In [ ]:
```python
import pandas as pd
import numpy as np
from pandasgui import show
from datetime import datetime
import matplotlib.pyplot as plt
import seaborn as sns
import lifelines
from lifelines import KaplanMeierFitter
from lifelines.plotting import add_at_risk_counts
from sksurv.nonparametric import kaplan_meier_estimator
import scipy
import scipy.stats as ss
import statsmodels
import statsmodels.api as sm
plt.rcParams['font.family'] = "Times New Roman"
from sklearn import metrics

import schemdraw
from schemdraw import flow



%matplotlib inline
```

```
c:\Users\myleg\miniconda3\envs\cudapy_310\lib\site-packages\numpy\_distributor_init.py:30: UserWarning: loaded more than 1 DLL from .libs:
c:\Users\myleg\miniconda3\envs\cudapy_310\lib\site-packages\numpy\.libs\libopenblas.FB5AE2TYXYH2IJRDKGDGQ3XBKLKTF43H.gfortran-win_amd64.dll
c:\Users\myleg\miniconda3\envs\cudapy_310\lib\site-packages\numpy\.libs\libopenblas64__v0.3.21-gcc_10_3_0.dll
  warnings.warn("loaded more than 1 DLL from .libs:"
```

In [ ]:
```python
print(lifelines.__version__)
print(scipy.__version__)
print(statsmodels.__version__)
```

```
0.27.8
1.10.1
0.14.0
```

In [ ]:
```python
raw_df = pd.read_excel(r"C:\Users\myleg\OneDrive\바탕 화면\Thesis works\Long term outcome of bypass for ICAD\Long_term_result_Ischemic_disease_and_bypass.xlsx")
```

In [ ]:
```python
raw_df.head()
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	4y_mRS	4y_mRS_grade	5y_mRS	5y_mRS_grade	last_mRS	last_mRS_grade	last_fu_date	op_date_to_last_fu_date_days	op_date_to_last_fu_date_weeks	op_date_to_last_f
0	NaN	NaN	NaN	0: male\n1: female	NaN	NaN	NaN	0: no\n1: yes	NaN	NaN	...	NaN	0: good (0-2)\n1: moderate (3)\n2: poor (4-6)	NaN	0: good (0-2)\n1: moderate (3)\n2: poor (4-6)	NaN	0: good (0-2)\n1: moderate (3)\n2: poor (4-6)	NaN	NaN	NaN	
1	20181008.0	1.0	40.0	0	Rt. ICA occlusion	EIAB, Rt. (STA-M4)	NaN	0	NaN	NaN	...	1.0	0	1.0	0	1.0	0	20231108.0	NaN	NaN	
2	20180830.0	2.0	49.0	1	Lt. M1 and Rt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 2 months expired due to extensiv...	0	NaN	NaN	...	NaN	NaN	NaN	NaN	6.0	2	20181024.0	NaN	NaN	
3	20181224.0	3.0	59.0	0	Rt. ICA stenosis	EIAB, Rt. (STA-M4) and EDAS, Rt.	restenosis of previously stented lesion (6 mon...	0	NaN	NaN	...	3.0	1	3.0	1	3.0	1	20230908.0	NaN	NaN	
4	20181031.0	4.0	59.0	0	Lt. M1 occlusion	EIAB, Lt. (STA parietal-M4)	NaN	0	NaN	NaN	...	1.0	0	1.0	0	1.0	0	20231113.0	NaN	NaN	

5 rows × 264 columns

In [ ]:
```python
raw_df.drop(0, axis=0, inplace=True)
```

In [ ]:
```python
raw_df = raw_df[raw_df.op_date.notna()]
```

In [ ]:
```python
raw_df.head()
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	4y_mRS	4y_mRS_grade	5y_mRS	5y_mRS_grade	last_mRS	last_mRS_grade	last_fu_date	op_date_to_last_fu_date_days	op_date_to_last_fu_date_weeks	op_date_to_last_fu_date
1	20181008.0	1.0	40.0	0	Rt. ICA occlusion	EIAB, Rt. (STA-M4)	NaN	0	NaN	NaN	...	1.0	0	1.0	0	1.0	0	20231108.0	NaN	NaN	
2	20180830.0	2.0	49.0	1	Lt. M1 and Rt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 2 months expired due to extensiv...	0	NaN	NaN	...	NaN	NaN	NaN	NaN	6.0	2	20181024.0	NaN	NaN	
3	20181224.0	3.0	59.0	0	Rt. ICA stenosis	EIAB, Rt. (STA-M4) and EDAS, Rt.	restenosis of previously stented lesion (6 mon...	0	NaN	NaN	...	3.0	1	3.0	1	3.0	1	20230908.0	NaN	NaN	
4	20181031.0	4.0	59.0	0	Lt. M1 occlusion	EIAB, Lt. (STA parietal-M4)	NaN	0	NaN	NaN	...	1.0	0	1.0	0	1.0	0	20231113.0	NaN	NaN	
5	20180829.0	5.0	59.0	0	Rt. ICA and M1 stenosis	EIAB, Rt. (STA-M4)	aborted angioplasty at main lesion	0	NaN	NaN	...	2.0	0	2.0	0	2.0	0	20230901.0	NaN	NaN	

5 rows × 264 columns

In [ ]:
```python
raw_df.shape
```

Out[ ]:
```
(109, 264)
```

In [ ]:
```python
raw_df.dtypes
```

Out[ ]:
```
op_date                          float64
id                               float64
age                              float64
sex                               object
diagnosis                         object
                                  ...
last_mRS_grade                    object
last_fu_date                     float64
op_date_to_last_fu_date_days     float64
op_date_to_last_fu_date_weeks    float64
op_date_to_last_fu_date_months   float64
Length: 264, dtype: object
```

In [ ]:
```python
raw_df.columns
```

Out[ ]:
```
Index(['op_date', 'id', 'age', 'sex', 'diagnosis', 'op_name', 'remarks',
       'ipsilateral_postop_stroke', 'ipsilateral_postop_stroke_date',
       'op_date_to_ipsilateral_postop_stroke_date_days',
       ...
       '4y_mRS', '4y_mRS_grade', '5y_mRS', '5y_mRS_grade', 'last_mRS',
       'last_mRS_grade', 'last_fu_date', 'op_date_to_last_fu_date_days',
       'op_date_to_last_fu_date_weeks', 'op_date_to_last_fu_date_months'],
      dtype='object', length=264)
```

In [ ]:
```python
col_list = raw_df.columns.values.tolist()
col_list
```

Out[ ]:
```
['op_date',
 'id',
 'age',
 'sex',
 'diagnosis',
 'op_name',
 'remarks',
 'ipsilateral_postop_stroke',
 'ipsilateral_postop_stroke_date',
 'op_date_to_ipsilateral_postop_stroke_date_days',
 'ipsilateral_symptomatic_postop_stroke',
 'ipsilateral_symptomatic_postop_stroke_date',
 'op_date_to_ipsilateral_symptomatic_postop_stroke_date_days',
 'ipsilateral_asymptomatic_postop_stroke',
 'ipsilateral_asymptomatic_postop_stroke_date',
 'op_date_to_ipsilateral_asymptomatic_postop_stroke_date_days',
 'any_postop_stroke',
 'any_postop_stroke_date',
 'op_date_to_any_postop_stroke_date_days',
 'any_symptomatic_postop_stroke',
 'any_symptomatic_postop_stroke_date',
 'op_date_to_any_symptomatic_postop_stroke_date_days',
 'any_asymptomatic_postop_stroke',
 'any_asymptomatic_postop_stroke_date',
 'op_date_to_any_asymptomatic_postop_stroke_date_days',
 'any_postop_complication',
 'any_postop_complication_date',
 'op_date_to_any_postop_complication_date_days',
 'any_symptomatic_postop_complication',
 'any_symptomatic_postop_complication_date',
 'op_date_to_any_symptomatic_postop_complication_date_days',
 'any_asymptomatic_postop_complication',
 'any_asymptomatic_postop_complication_date',
 'any_postop_30_days_complication',
 'any_postop_30_days_complication_date',
 'op_date_to_any_postop_30_days_complication_date_days',
 'any_symptomatic_postop_30_days_complication',
 'any_symptomatic_postop_30_days_complication_date',
 'op_date_to_any_symptomatic_postop_30_days_complication_date_days',
 'any_asymptomatic_postop_30_days_complication',
 'any_asymptomatic_postop_30_days_complication_date',
 'op_date_to_any_asymptomatic_postop_30_days_complication_date_days',
 'any_postop_1_year_complication',
 'any_postop_1_year_complication_date',
 'op_date_to_any_postop_1_year_complication_date_days',
 'any_symptomatic_postop_1_year_complication',
 'any_symptomatic_postop_1_year_complication_date',
 'op_date_to_any_symptomatic_postop_1_year_complication_date_days',
 'any_asymptomatic_postop_1_year_complication',
 'any_asymptomatic_postop_1_year_complication_date',
 'op_date_to_any_asymptomatic_postop_1_year_complication_date_days',
 'previous_EVT',
 'BMI',
 'HTN',
 'DM',
 'HbA1c_level',
 'DL',
```

```
'LDL_level',
'TG_level',
'HDL_level',
'CAD',
'arrhythmia',
'FHx',
'smoking',
'prev_stroke',
'RNF213',
'presentation',
'presentation_remarks',
'etiology',
'TIA',
'multiple_TIAs',
'infarction',
'multiple_infarction',
'hemorrhage',
'multiple_hemorrhage',
'first_event_date',
'last_event_date',
'first_event_to_last_event_days',
'first_event_to_last_event_weeks',
'first_event_to_last_event_months',
'first_event_to_treatment_days',
'first_event_to_treatment_weeks',
'first_event_to_treatment_months',
'last_event_to_treatment_days',
'last_event_to_treatment_weeks',
'last_event_to_treatment_months',
'initial_mRS',
'initial_mRS_grade',
'initial_NIHSS',
'initial_NIHSS_grade',
'preop_mRS',
'preop_mRS_grade',
'preop_NIHSS',
'preop_NIHSS_grade',
'initial_preop_mRS_diff',
'initial_preop_NIHSS_diff',
'diffusion_infarction_pattern',
'initial_infarction_volume',
'preop_infarction_volume',
'infarction_volume_diff',
'lesion_stenosis_occlusion',
'degree_stenosis',
'length_stenosis',
'ICA_lesion_type',
'MCA_lesion_type',
'Mori_type',
'single_ICAD',
'multiple_ICAD_bilateral',
'main_lesion_site',
'other_lesion_site',
'n_intracranial_lesion_over_stenosis_50',
'collateral_status',
'preop_image_hemodynamic_insufficiency',
'pre_SPECT_R_ACA_perfusion',
'pre_SPECT_R_MCA_perfusion',
'pre_SPECT_R_PCA_perfusion',
'pre_SPECT_L_ACA_perfusion',
'pre_SPECT_L_MCA_perfusion',
'pre_SPECT_L_PCA_perfusion',
'pre_SPECT_R_ACA_reserve',
'pre_SPECT_R_MCA_reserve',
'pre_SPECT_R_PCA_reserve',
'pre_SPECT_L_ACA_reserve',
'pre_SPECT_L_MCA_reserve',
'pre_SPECT_L_PCA_reserve',
'postop_image_hemodynamic_insufficiency',
'postop_hemodynamic_insufficiency_grade',
'post_SPECT_R_ACA_perfusion',
'post_SPECT_R_MCA_perfusion',
'post_SPECT_R_PCA_perfusion',
'post_SPECT_L_ACA_perfusion',
'post_SPECT_L_MCA_perfusion',
'post_SPECT_L_PCA_perfusion',
'post_SPECT_R_ACA_reserve',
'post_SPECT_R_MCA_reserve',
'post_SPECT_R_PCA_reserve',
'post_SPECT_L_ACA_reserve',
'post_SPECT_L_MCA_reserve',
'post_SPECT_L_PCA_reserve',
'3y_image_hemodynamic_insufficiency',
'3y_hemodynamic_insufficiency_grade',
'3y_SPECT_R_ACA_perfusion',
'3y_SPECT_R_MCA_perfusion',
'3y_SPECT_R_PCA_perfusion',
'3y_SPECT_L_ACA_perfusion',
'3y_SPECT_L_MCA_perfusion',
'3y_SPECT_L_PCA_perfusion',
'3y_SPECT_R_ACA_reserve',
'3y_SPECT_R_MCA_reserve',
'3y_SPECT_R_PCA_reserve',
'3y_SPECT_L_ACA_reserve',
'3y_SPECT_L_MCA_reserve',
'3y_SPECT_L_PCA_reserve',
'bypass_type',
'initial_bypass_patency',
'6m_bypass_patency',
'1y_bypass_patency',
'2y_bypass_patency',
'3y_bypass_patency',
'4y_bypass_patency',
'5y_bypass_patency',
'last_bypass_patency',
'initial_STAb_diameter',
'1w_STAb_diameter',
'6m_STAb_diameter',
'1y_STAb_diameter',
'2y_STAb_diameter',
'3y_STAb_diameter',
'4y_STAb_diameter',
'5y_STAb_diameter',
'last_STAb_diameter',
'initial_STAb_diameter_second',
'1w_STAb_diameter_second',
'6m_STAb_diameter_second',
'1y_STAb_diameter_second',
'2y_STAb_diameter_second',
'3y_STAb_diameter_second',
'4y_STAb_diameter_second',
'5y_STAb_diameter_second',
'last_STAb_diameter_second',
'initial_STAb_diameter_indirect',
'1w_STAb_diameter_indirect',
'6m_STAb_diameter_indirect',
'1y_STAb_diameter_indirect',
'2y_STAb_diameter_indirect',
'3y_STAb_diameter_indirect',
'4y_STAb_diameter_indirect',
'5y_STAb_diameter_indirect',
'last_STAb_diameter_indirect',
'initial_STAs_diameter',
'1w_STAs_diameter',
'6m_STAs_diameter',
'1y_STAs_diameter',
'2y_STAs_diameter',
'3y_STAs_diameter',
'4y_STAs_diameter',
'5y_STAs_diameter',
'last_STAs_diameter',
'30_days_stroke',
'30_days_complication',
'30_days_event_date',
'6_mo_stroke',
'6_mo_stroke_date',
'1_year_stroke',
'1_year_stroke_date',
'2_year_stroke',
'2_year_stroke_date',
'3_year_stroke',
'3_year_stroke_date',
'4_year_stroke',
'4_year_stroke_date',
'5_year_stroke',
'5_year_stroke_date',
'5_year_complication',
'5_year_complication_date',
'last_fu_stroke',
'first_fu_stroke_date',
'last_fu_stroke_date',
'last_fu_event',
'last_fu_event_date',
'op_date_to_ipsilateral_postop_stroke_days',
'op_date_to_ipsilateral_symptomatic_postop_stroke_days',
'op_date_to_ipsilateral_asymptomatic_postop_stroke_days',
'op_date_to_any_postop_stroke_days',
'op_date_to_any_symptomatic_postop_stroke_days',
'op_date_to_any_asymptomatic_postop_stroke_days',
'op_date_to_any_postop_complication_days',
'op_date_to_any_symptomatic_postop_complication_days',
'op_date_to_any_asymptomatic_postop_complication_days',
'op_date_to_any_postop_30_days_complication_days',
'op_date_to_any_symptomatic_postop_30_days_complication_days',
'op_date_to_any_asymptomatic_postop_30_days_complication_days',
'op_date_to_any_postop_1_year_complication_days',
'op_date_to_any_symptomatic_postop_1_year_complication_days',
'op_date_to_any_asymptomatic_postop_1_year_complication_days',
'op_date_to_first_fu_stroke_days',
'op_date_to_first_fu_stroke_weeks',
'op_date_to_first_fu_stroke_months',
'op_date_to_last_fu_stroke_days',
'op_date_to_last_fu_stroke_weeks',
'op_date_to_last_fu_stroke_months',
'op_date_to_last_fu_event_days',
'op_date_to_last_fu_event_weeks',
'op_date_to_last_fu_event_months',
'postop_mRS',
'postop_mRS_grade',
'6m_mRS',
'6m_mRS_grade',
'1y_mRS',
'1y_mRS_grade',
'2y_mRS',
'2y_mRS_grade',
'3y_mRS',
'3y_mRS_grade',
'4y_mRS',
'4y_mRS_grade',
'5y_mRS',
'5y_mRS_grade',
'last_mRS',
'last_mRS_grade',
'last_fu_date',
'op_date_to_last_fu_date_days',
'op_date_to_last_fu_date_weeks',
'op_date_to_last_fu_date_months']
```

In [ ]:
```python
# Selecting date columns
def hasdate(x):
    if 'date' in x:
        if 'date_' in x:
            return False
        else:
            return True

    else:
        return False

def hasdays(x):
    if '_days' in x:
        if '_days_' in x:
            return False
        else:
            return True
    else:
        return False

def hasweeks(x):
    if 'weeks' in x:
        return True
    else:
        return False

def hasmonths(x):
    if 'months' in x:
        return True
    else:
        return False
```

In [ ]:
```python
date_cols = [x for x in col_list if hasdate(x)]
days_cols = [x for x in col_list if hasdays(x)]
weeks_cols = [x for x in col_list if hasweeks(x)]
months_cols = [x for x in col_list if hasmonths(x)]
```

In [ ]:
```python
date_cols
```

Out[ ]:
```
['op_date',
 'ipsilateral_postop_stroke_date',
 'ipsilateral_symptomatic_postop_stroke_date',
 'ipsilateral_asymptomatic_postop_stroke_date',
 'any_postop_stroke_date',
 'any_symptomatic_postop_stroke_date',
 'any_asymptomatic_postop_stroke_date',
 'any_postop_complication_date',
 'any_symptomatic_postop_complication_date',
 'any_asymptomatic_postop_complication_date',
 'any_postop_30_days_complication_date',
 'any_symptomatic_postop_30_days_complication_date',
 'any_asymptomatic_postop_30_days_complication_date',
 'any_postop_1_year_complication_date',
 'any_symptomatic_postop_1_year_complication_date',
 'any_asymptomatic_postop_1_year_complication_date',
 'first_event_date',
 'last_event_date',
 '30_days_event_date',
 '6_mo_stroke_date',
 '1_year_stroke_date',
 '2_year_stroke_date',
 '3_year_stroke_date',
 '4_year_stroke_date',
 '5_year_stroke_date',
 '5_year_complication_date',
 'first_fu_stroke_date',
 'last_fu_stroke_date',
 'last_fu_event_date',
 'last_fu_date']
```

In [ ]:
```python
raw_df[date_cols].dtypes
```

Out[ ]:
```
op_date                                          float64
ipsilateral_postop_stroke_date                   float64
ipsilateral_symptomatic_postop_stroke_date       float64
ipsilateral_asymptomatic_postop_stroke_date      float64
any_postop_stroke_date                           float64
any_symptomatic_postop_stroke_date               float64
any_asymptomatic_postop_stroke_date              float64
any_postop_complication_date                     float64
any_symptomatic_postop_complication_date         float64
any_asymptomatic_postop_complication_date        float64
any_postop_30_days_complication_date             float64
any_symptomatic_postop_30_days_complication_date float64
any_asymptomatic_postop_30_days_complication_date float64
any_postop_1_year_complication_date              float64
any_symptomatic_postop_1_year_complication_date  float64
any_asymptomatic_postop_1_year_complication_date float64
first_event_date                                 float64
last_event_date                                  float64
30_days_event_date                               float64
6_mo_stroke_date                                 float64
1_year_stroke_date                               float64
2_year_stroke_date                               float64
3_year_stroke_date                               float64
4_year_stroke_date                               float64
5_year_stroke_date                               float64
5_year_complication_date                         float64
first_fu_stroke_date                             float64
last_fu_stroke_date                              float64
last_fu_event_date                               float64
last_fu_date                                     float64
dtype: object
```

In [ ]:
```python
date_cols_df = raw_df[date_cols].astype('Int64').astype('str').applymap(lambda x: datetime.strptime(x, '%Y%m%d') if len(x)==8 else None)
date_cols_df
```

Out[ ]:

	op_date	ipsilateral_postop_stroke_date	ipsilateral_symptomatic_postop_stroke_date	ipsilateral_asymptomatic_postop_stroke_date	any_postop_stroke_date	any_symptomatic_postop_stroke_date	any_asymptomatic_postop_stroke_date	any_postop_complication_date	any_symptomatic_postop_complication_date	any_asymptomatic_postop_complication_
1	2018-10-08	NaT	NaT	NaT	NaT	NaT	NaT	NaT	NaT	
2	2018-08-30	NaT	NaT	NaT	2018-10-24	2018-10-24	NaT	2018-10-24	2018-10-24	
3	2018-12-24	NaT	NaT	NaT	NaT	NaT	NaT	NaT	NaT	
4	2018-10-31	NaT	NaT	NaT	NaT	NaT	NaT	NaT	NaT	
5	2018-08-29	NaT	NaT	NaT	NaT	NaT	NaT	NaT	NaT	
...	...	...	...	...	...	...	...	...	...	
105	2007-10-15	NaT	NaT	NaT	NaT	NaT	NaT	NaT	NaT	
106	2007-10-23	NaT	NaT	NaT	NaT	NaT	NaT	NaT	NaT	
107	2007-10-24	NaT	NaT	NaT	NaT	NaT	NaT	NaT	NaT	
108	2010-04-05	NaT	NaT	NaT	2010-04-08	2010-04-08	NaT	2010-04-08	2010-04-08	
109	2017-10-19	2017-10-22	2017-10-22	NaT	2017-10-22	2017-10-22	NaT	2017-10-22	2017-10-22	

109 rows × 30 columns

In [ ]:
```python
def diff_days(s1, s2):
    return (s2 - s1) / np.timedelta64(1, 'D')

def diff_weeks(s1, s2):
    return (s2 - s1) / np.timedelta64(1, 'W')

def diff_months(s1, s2):
    return diff_weeks(s1, s2) / 4
```

In [ ]:
```python
diff_days(date_cols_df['first_event_date'], date_cols_df['last_event_date'])
```

Out[ ]:
```
1      479.0
2      169.0
3      158.0
4      129.0
5       32.0
       ...
105     16.0
106    822.0
107    665.0
108      7.0
109     43.0
Length: 109, dtype: float64
```

In [ ]:
```python
diff_weeks(date_cols_df['first_event_date'], date_cols_df['last_event_date'])
```

Out[ ]:
```
1      68.428571
2      24.142857
3      22.571429
4      18.428571
5       4.571429
          ...
105     2.285714
106    117.428571
107     95.000000
108      1.000000
109      6.142857
Length: 109, dtype: float64
```

In [ ]:
```python
diff_months(date_cols_df['first_event_date'], date_cols_df['last_event_date'])
```

Out[ ]:
```
1      17.107143
2       6.035714
3       5.642857
4       4.607143
5       1.142857
          ...
105     0.571429
106    29.357143
107    23.750000
108     0.250000
109     1.535714
Length: 109, dtype: float64
```

In [ ]:
```python
( date_cols_df['last_event_date'] - date_cols_df['first_event_date']) / np.timedelta64(1, 'D')
```

Out[ ]:
```
1      479.0
2      169.0
3      158.0
4      129.0
5       32.0
       ...
105     16.0
106    822.0
107    665.0
108      7.0
109     43.0
Length: 109, dtype: float64
```

In [ ]:
```
( date_cols_df['last_event_date'] - date_cols_df['first_event_date']) / np.timedelta64(1, 'W')
```

Out[ ]:
```
1       68.428571
2       24.142857
3       22.571429
4       18.428571
5        4.571429
           ...
105      2.285714
106    117.428571
107     95.000000
108      1.000000
109      6.142857
Length: 109, dtype: float64
```

In [ ]:
```
(( date_cols_df['last_event_date'] - date_cols_df['first_event_date']) / np.timedelta64(1, 'W'))/4
```

Out[ ]:
```
1      17.107143
2       6.035714
3       5.642857
4       4.607143
5       1.142857
           ...
105     0.571429
106    29.357143
107    23.750000
108     0.250000
109     1.535714
Length: 109, dtype: float64
```

In [ ]:
```
date_cols
```

Out[ ]:
```
['op_date',
 'ipsilateral_postop_stroke_date',
 'ipsilateral_symptomatic_postop_stroke_date',
 'ipsilateral_asymptomatic_postop_stroke_date',
 'any_postop_stroke_date',
 'any_symptomatic_postop_stroke_date',
 'any_asymptomatic_postop_stroke_date',
 'any_postop_complication_date',
 'any_symptomatic_postop_complication_date',
 'any_asymptomatic_postop_complication_date',
 'any_postop_30_days_complication_date',
 'any_symptomatic_postop_30_days_complication_date',
 'any_asymptomatic_postop_30_days_complication_date',
 'any_postop_1_year_complication_date',
 'any_symptomatic_postop_1_year_complication_date',
 'any_asymptomatic_postop_1_year_complication_date',
 'first_event_date',
 'last_event_date',
 '30_days_event_date',
 '6_mo_stroke_date',
 '1_year_stroke_date',
 '2_year_stroke_date',
 '3_year_stroke_date',
 '4_year_stroke_date',
 '5_year_stroke_date',
 '5_year_complication_date',
 'first_fu_stroke_date',
 'last_fu_stroke_date',
 'last_fu_event_date',
 'last_fu_date']
```

In [ ]:

In [ ]:
```
# Create copy of raw data
df = raw_df
```

In [ ]:
```
df[date_cols] = raw_df[date_cols].astype('Int64').astype('str').applymap(lambda x: datetime.strptime(x, '%Y%m%d') if len(x)==8 else None)
```

In [ ]:
```
days_cols
```

Out[ ]:
```
['op_date_to_ipsilateral_postop_stroke_date_days',
 'op_date_to_ipsilateral_symptomatic_postop_stroke_date_days',
 'op_date_to_ipsilateral_asymptomatic_postop_stroke_date_days',
 'op_date_to_any_postop_stroke_date_days',
 'op_date_to_any_symptomatic_postop_stroke_date_days',
 'op_date_to_any_asymptomatic_postop_stroke_date_days',
 'op_date_to_any_postop_complication_date_days',
 'op_date_to_any_symptomatic_postop_complication_date_days',
 'op_date_to_any_postop_1_year_complication_date_days',
 'op_date_to_any_symptomatic_postop_1_year_complication_date_days',
 'op_date_to_any_asymptomatic_postop_1_year_complication_date_days',
 'first_event_to_last_event_days',
 'first_event_to_treatment_days',
 'last_event_to_treatment_days',
 'op_date_to_ipsilateral_postop_stroke_days',
 'op_date_to_ipsilateral_symptomatic_postop_stroke_days',
 'op_date_to_ipsilateral_asymptomatic_postop_stroke_days',
 'op_date_to_any_postop_stroke_days',
 'op_date_to_any_symptomatic_postop_stroke_days',
 'op_date_to_any_asymptomatic_postop_stroke_days',
 'op_date_to_any_postop_complication_days',
 'op_date_to_any_symptomatic_postop_complication_days',
 'op_date_to_any_asymptomatic_postop_complication_days',
 'op_date_to_any_postop_1_year_complication_days',
 'op_date_to_any_symptomatic_postop_1_year_complication_days',
 'op_date_to_any_asymptomatic_postop_1_year_complication_days',
 'op_date_to_first_fu_stroke_days',
 'op_date_to_last_fu_stroke_days',
 'op_date_to_last_fu_event_days',
 'op_date_to_last_fu_date_days']
```

In [ ]:
```
weeks_cols
```

Out[ ]:
```
['first_event_to_last_event_weeks',
 'first_event_to_treatment_weeks',
 'last_event_to_treatment_weeks',
 'op_date_to_first_fu_stroke_weeks',
 'op_date_to_last_fu_stroke_weeks',
 'op_date_to_last_fu_event_weeks',
 'op_date_to_last_fu_date_weeks']
```

In [ ]:
```
months_cols
```

Out[ ]:
```
['first_event_to_last_event_months',
 'first_event_to_treatment_months',
 'last_event_to_treatment_months',
 'op_date_to_first_fu_stroke_months',
 'op_date_to_last_fu_stroke_months',
 'op_date_to_last_fu_event_months',
 'op_date_to_last_fu_date_months']
```

In [ ]:
```
# Calculate number of days, weeks, and months between dates
df[diff_days(date_cols_df['ipsilateral_postop_stroke_date'], date_cols_df['op_date'])].notnull()]
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	4y_mRS	4y_mRS_grade	5y_mRS	5y_mRS_grade	last_mRS	last_mRS_grade	last_fu_date	op_date_to_last_fu_date_days	op_date_to_last_fu_date_weeks	op_date_to_last_fu_
17	2015-01-12	17.0	62.0	0	Lt. ICA occlusion	EIAB, Lt (STA-M4 bypass)	postoperative 5 months focal infarction at Lt....	1	2015-06-11	NaN	...	2.0	0	2.0	0	2.0	0	2023-07-21	NaN	NaN	
22	2015-11-17	22.0	63.0	1	Bilateral M1 and ICA stenosis	EIAB, Lt. (double barrel STA-M4 bypass)	immediate postoperative bypass failure lead to...	1	2015-11-17	NaN	...	NaN	NaN	NaN	NaN	3.0	1	2016-11-28	NaN	NaN	
28	2014-05-17	28.0	74.0	0	Lt. ICA occlusion	STA-M3 bypass, Lt.	postoperative 1 day ipsilateral MCA territory ...	1	2014-05-18	NaN	...	6.0	2	NaN	NaN	6.0	2	2018-07-16	NaN	NaN	
37	2013-06-19	37.0	56.0	0	ICA stenosis, bilateral (s/p STA-M2 bypass, Rt...	STA-M4 bypass, Lt.	postoperative 3 years ipsilateral MCA territor...	1	2017-10-25	NaN	...	4.0	2	4.0	2	4.0	2	2020-11-02	NaN	NaN	
39	2013-05-08	39.0	49.0	0	Lt. ICA occlusion (s/p stent insertion d/t Lt....	OA-M4 bypass, Lt.	angioplasty and stenting at Lt. V4\nimmediate ...	1	2013-05-21	NaN	...	2.0	0	2.0	0	2.0	0	2022-10-05	NaN	NaN	
60	2010-11-10	60.0	72.0	0	Rt. distal ICA and M1 stenosis	EIAB, Rt. (STA-M4)	2 vessel disease (high perioperative risk grou...	1	2010-11-13	NaN	...	4.0	2	4.0	2	4.0	2	2020-12-24	NaN	NaN	
101	2007-08-06	101.0	60.0	0	Lt. distal ICA occlusion and Rt. distal ICA oc...	EIAB, Lt. (STA-M4)	postoperative 3 months asymptomatic ipsilatera...	1	2008-01-03	NaN	...	2.0	0	2.0	0	2.0	0	2016-07-05	NaN	NaN	
109	2017-10-19	109.0	58.0	0	Bilateral ICA occlusion	EIAB, Lt. (STA-M4)	Hemodynamic insufficiency (Rt. MCA territory >...	1	2017-10-22	NaN	...	NaN	NaN	NaN	NaN	5.0	2	2019-04-03	NaN	NaN	

8 rows × 264 columns

In [ ]:
```
df['op_date_to_ipsilateral_postop_stroke_date_days'] = diff_days(date_cols_df['op_date'],date_cols_df['ipsilateral_postop_stroke_date'])
df['op_date_to_ipsilateral_symptomatic_postop_stroke_date_days'] = diff_days(date_cols_df['op_date'], date_cols_df['ipsilateral_symptomatic_postop_stroke_date'])
df['op_date_to_ipsilateral_asymptomatic_postop_stroke_date_days']= diff_days(date_cols_df['op_date'], date_cols_df['ipsilateral_asymptomatic_postop_stroke_date'])
df['op_date_to_any_postop_stroke_date_days']= diff_days(date_cols_df['op_date'], date_cols_df['any_postop_stroke_date'])
df['op_date_to_any_symptomatic_postop_stroke_date_days']= diff_days(date_cols_df['op_date'], date_cols_df['any_symptomatic_postop_stroke_date'])
df['op_date_to_any_asymptomatic_postop_stroke_date_days']= diff_days(date_cols_df['op_date'], date_cols_df['any_asymptomatic_postop_stroke_date'])
df['op_date_to_any_postop_complication_date_days']= diff_days(date_cols_df['op_date'], date_cols_df['any_postop_complication_date'])
df['op_date_to_any_symptomatic_postop_complication_date_days'] = diff_days(date_cols_df['op_date'], date_cols_df['any_symptomatic_postop_complication_date'])
df['op_date_to_any_postop_30_days_complication_date_days']= diff_days(date_cols_df['op_date'], date_cols_df['any_postop_30_days_complication_date'])
df['op_date_to_any_symptomatic_postop_30_days_complication_date_days'] = diff_days(date_cols_df['op_date'], date_cols_df['any_symptomatic_postop_30_days_complication_date'])
df['op_date_to_any_asymptomatic_postop_30_days_complication_date_days'] = diff_days(date_cols_df['op_date'], date_cols_df['any_asymptomatic_postop_30_days_complication_date'])
df['op_date_to_any_postop_1_year_complication_date_days']= diff_days(date_cols_df['op_date'], date_cols_df['any_postop_1_year_complication_date'])
df['op_date_to_any_symptomatic_postop_1_year_complication_date_days'] = diff_days(date_cols_df['op_date'], date_cols_df['any_symptomatic_postop_1_year_complication_date'])
df['op_date_to_any_asymptomatic_postop_1_year_complication_date_days'] = diff_days(date_cols_df['op_date'], date_cols_df['any_asymptomatic_postop_1_year_complication_date'])
```

In [ ]:
```
df['first_event_to_last_event_days'] = diff_days(date_cols_df['first_event_date'], date_cols_df['last_event_date'])
df['first_event_to_last_event_weeks'] = diff_weeks(date_cols_df['first_event_date'], date_cols_df['last_event_date'])
df['first_event_to_last_event_months'] = diff_months(date_cols_df['first_event_date'], date_cols_df['last_event_date'])
```

In [ ]:
```
df['first_event_to_treatment_days'] = diff_days(date_cols_df['first_event_date'], date_cols_df['op_date'])
df['first_event_to_treatment_weeks'] = diff_weeks(date_cols_df['first_event_date'], date_cols_df['op_date'])
df['first_event_to_treatment_months'] = diff_months(date_cols_df['first_event_date'], date_cols_df['op_date'])
```

In [ ]:
```
df['last_event_to_treatment_days'] = diff_days(date_cols_df['last_event_date'], date_cols_df['op_date'])
df['last_event_to_treatment_weeks'] = diff_weeks(date_cols_df['last_event_date'], date_cols_df['op_date'])
df['last_event_to_treatment_months'] = diff_months(date_cols_df['last_event_date'], date_cols_df['op_date'])
```

In [ ]:
```
df['op_date_to_first_fu_stroke_days'] = diff_days(date_cols_df['op_date'], date_cols_df['first_fu_stroke_date'])
df['op_date_to_first_fu_stroke_weeks'] = diff_weeks(date_cols_df['op_date'], date_cols_df['first_fu_stroke_date'])
df['op_date_to_first_fu_stroke_months'] = diff_months(date_cols_df['op_date'], date_cols_df['first_fu_stroke_date'])
```

In [ ]:
```
df['op_date_to_last_fu_stroke_days'] = diff_days(date_cols_df['op_date'], date_cols_df['last_fu_stroke_date'])
df['op_date_to_last_fu_stroke_weeks'] = diff_weeks(date_cols_df['op_date'], date_cols_df['last_fu_stroke_date'])
df['op_date_to_last_fu_stroke_months'] = diff_months(date_cols_df['op_date'], date_cols_df['last_fu_stroke_date'])
```

In [ ]:
```
df['op_date_to_last_fu_event_days'] = diff_days(date_cols_df['op_date'], date_cols_df['last_fu_event_date'])
df['op_date_to_last_fu_event_weeks'] = diff_weeks(date_cols_df['op_date'], date_cols_df['last_fu_event_date'])
df['op_date_to_last_fu_event_months'] = diff_months(date_cols_df['op_date'], date_cols_df['last_fu_event_date'])
```

In [ ]:
```
df['op_date_to_last_fu_date_days'] = diff_days(date_cols_df['op_date'], date_cols_df['last_fu_date'])
df['op_date_to_last_fu_date_weeks'] = diff_weeks(date_cols_df['op_date'], date_cols_df['last_fu_date'])
df['op_date_to_last_fu_date_months'] = diff_months(date_cols_df['op_date'], date_cols_df['last_fu_date'])
```

In [ ]:
```
df[days_cols]
```

Out[ ]:

	op_date_to_ipsilateral_postop_stroke_date_days	op_date_to_ipsilateral_symptomatic_postop_stroke_date_days	op_date_to_ipsilateral_asymptomatic_postop_stroke_date_days	op_date_to_any_postop_stroke_date_days	op_date_to_any_symptomatic_postop_stroke_date_days	op_date_to_any_asymptomatic_postop_stroke_date_days	op_date_to_any_postop
1	NaN	NaN	NaN	NaN	NaN	NaN	
2	NaN	NaN	NaN	55.0	55.0	NaN	
3	NaN	NaN	NaN	NaN	NaN	NaN	
4	NaN	NaN	NaN	NaN	NaN	NaN	
5	NaN	NaN	NaN	NaN	NaN	NaN	
...	...	...	...	...	...	...	
105	NaN	NaN	NaN	NaN	NaN	NaN	
106	NaN	NaN	NaN	NaN	NaN	NaN	
107	NaN	NaN	NaN	NaN	NaN	NaN	
108	NaN	NaN	NaN	3.0	3.0	NaN	
109	3.0	3.0	NaN	3.0	3.0	NaN	

109 rows × 30 columns

In [ ]:
```
df[weeks_cols] = df[weeks_cols].applymap(lambda x: round(x, 0))
```

In [ ]:
```
df[months_cols] = df[months_cols].applymap(lambda x: round(x, 0))
```

In [ ]:

In [ ]:
```
# Check preprocessed raw data
df
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	4y_mRS	4y_mRS_grade	5y_mRS	5y_mRS_grade	last_mRS	last_mRS_grade	last_fu_date	op_date_to_last_fu_date_days	op_date_to_last_fu_date_weeks	op_date_to_last_fu_d
1	2018-10-08	1.0	40.0	0	Rt. ICA occlusion	EIAB, Rt. (STA-M4)	NaN	0	NaT	NaN	...	1.0	0	1.0	0	1.0	0	2023-11-08	1857.0	265.0	
2	2018-08-30	2.0	49.0	1	Lt. M1 and Rt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 2 months expired due to extensiv...	0	NaT	NaN	...	NaN	NaN	NaN	NaN	6.0	2	2018-10-24	55.0	8.0	
3	2018-12-24	3.0	59.0	0	Rt. ICA stenosis	EIAB, Rt. (STA-M4) and EDAS, Rt.	restenosis of previously stented lesion (6 mon...	0	NaT	NaN	...	3.0	1	3.0	1	3.0	1	2023-09-08	1719.0	246.0	
4	2018-10-31	4.0	59.0	0	Lt. M1 occlusion	EIAB, Lt. (STA parietal-M4)	NaN	0	NaT	NaN	...	1.0	0	1.0	0	1.0	0	2023-11-13	1839.0	263.0	
5	2018-08-29	5.0	59.0	0	Rt. ICA and M1 stenosis	EIAB, Rt. (STA-M4)	aborted angioplasty at main lesion	0	NaT	NaN	...	2.0	0	2.0	0	2.0	0	2023-09-01	1829.0	261.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
105	2007-10-15	105.0	56.0	0	Rt. distal ICA stenosis and Lt. distal ICA ste...	EIAB, Rt. (STA-M4)	NaN	0	NaT	NaN	...	0.0	0	0.0	0	0.0	0	2015-12-10	2978.0	425.0	
106	2007-10-23	106.0	52.0	1	Rt. distal ICA occlusion, Lt. proximal ICA ste...	EIAB, Rt. (STA-M4)	NaN	0	NaT	NaN	...	1.0	0	1.0	0	1.0	0	2013-06-30	2077.0	297.0	
107	2007-10-24	107.0	64.0	1	Lt. distal ICA occlusion	EIAB, Lt. (STA-M4)	NaN	0	NaT	NaN	...	0.0	0	0.0	0	0.0	0	2013-11-04	2203.0	315.0	
108	2010-04-05	108.0	67.0	0	Lt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 3 days contralateral BDZ infarct...	0	NaT	NaN	...	3.0	1	3.0	1	3.0	1	2019-10-05	3470.0	496.0	
109	2017-10-19	109.0	58.0	0	Bilateral ICA occlusion	EIAB, Lt. (STA-M4)	Hemodynamic insufficiency (Rt. MCA territory >...	1	2017-10-22	3.0	...	NaN	NaN	NaN	NaN	5.0	2	2019-04-03	531.0	76.0	

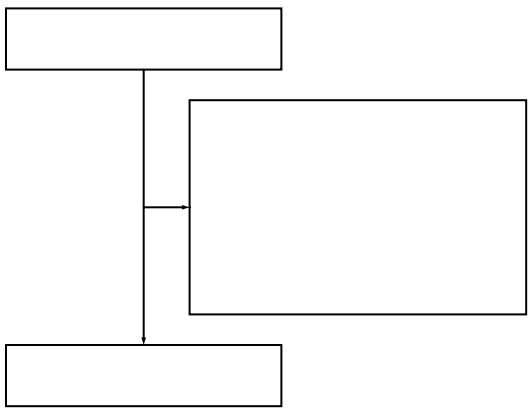
109 rows × 264 columns

In [ ]:

In [ ]:

Demographics and Clinical presentation

In [ ]:
```
with schemdraw.Drawing() as d:
    d.config(font="Times New Roman")
    d += flow.Box(w=9).anchor('S').label("145 cases of \n symptomatic steno-occlusive arterial disease\n treated with bypass surgery")
    d += flow.Arrow().down(d.unit*3)
    d += flow.Box(w=9).label("109 cases of symptomatic steno-occlusive disease\n due to atherosclerosis were included ")
    d += flow.Arrow().right(d.unit/2).at((0, -4.5))
    d += flow.Box(w=11, h=7).label("    36 Excluded:\n         9 Follow-up loss (less than 5 years)\n         1 Asymptomatic\n         1 Invasive fungal sinusitis\n         4 Iatrogenic causes\n         3 ICA injury during surgery\n           2 Tumor resection\n           1 CEA\n           1 Coil
```



In [ ]:
```
# age
df.age.median(), df.age.quantile([.25, .75])
```

Out[ ]:
```
(57.0,
 0.25    50.0
 0.75    63.0
 Name: age, dtype: float64)
```

In [ ]:
```
# sex
print(df.sex.value_counts())
print(df.sex.value_counts().values/df.sex.value_counts().sum())
```

```
0    80
1    29
Name: sex, dtype: int64
[0.73394495 0.26605505]
```

In [ ]:
```
# BMI
df.BMI.mean(), df.BMI.std()
```

Out[ ]:
```
(25.49669724770641, 2.947100287051121)
```

In [ ]:
```
df.BMI.median(), df.BMI.quantile([.25, .75])
```

Out[ ]:
```
(25.47,
 0.25    23.81
 0.75    26.77
 Name: BMI, dtype: object)
```

In [ ]:
```
# HTN
print(df.HTN.value_counts())
print(df.HTN.value_counts()/df.HTN.value_counts().sum()*100)
```

```
1    85
0    24
Name: HTN, dtype: int64
1    77.981651
0    22.018349
Name: HTN, dtype: float64
```

In [ ]:
```
# DM
print(df.DM.value_counts())
print(df.DM.value_counts()/df.DM.value_counts().sum()*100)
```

```
0    58
1    51
Name: DM, dtype: int64
0    53.211009
1    46.788991
Name: DM, dtype: float64
```

In [ ]:
```
# HbA1c
df.HbA1c_level.notnull().sum(), df.HbA1c_level.mean(), df.HbA1c_level.std()
```

Out[ ]:
```
(61, 7.359016393442623, 1.8729991145379938)
```

In [ ]:
```
# DL
print(df.DL.value_counts())
print(df.DL.value_counts()/df.DL.value_counts().sum()*100)
```

```
0    59
1    50
Name: DL, dtype: int64
0    54.12844
1    45.87156
Name: DL, dtype: float64
```

In [ ]:
```
# LDL, TG, HDL
print(df.LDL_level.notnull().sum(), df.LDL_level.mean(), df.LDL_level.std())
print(df.TG_level.notnull().sum(), df.TG_level.mean(), df.TG_level.std())
print(df.HDL_level.notnull().sum(), df.HDL_level.mean(), df.HDL_level.std())
```

```
49 114.93877551020408 43.27307099651454
49 162.75510204081633 95.64594664095043
49 42.59183673469388 11.935448545102489
```

In [ ]:
```
# CAD
print(df.CAD.value_counts())
print(df.CAD.value_counts()/df.CAD.value_counts().sum()*100)
```

```
0    84
1    25
Name: CAD, dtype: int64
0    77.06422
1    22.93578
Name: CAD, dtype: float64
```

In [ ]:
```
# arrhythmia
print(df.arrhythmia.value_counts())
print(df.arrhythmia.value_counts()/df.arrhythmia.value_counts().sum()*100)
```

```
0    106
1      3
Name: arrhythmia, dtype: int64
0    97.247706
1     2.752294
Name: arrhythmia, dtype: float64
```

In [ ]:
```
# Family history
print(df.FHx.value_counts())
print(df.FHx.value_counts()/df.FHx.value_counts().sum()*100)
```

```
0    71
1    38
Name: FHx, dtype: int64
0    65.137615
1    34.862385
Name: FHx, dtype: float64
```

In [ ]:
```
# smoking
print(df.smoking.value_counts().sort_index())
print(df.smoking.value_counts().sort_index()/df.smoking.value_counts().sum()*100)
```

```
0    42
1    17
2    50
Name: smoking, dtype: int64
0    38.53211
1    15.59633
2    45.87156
Name: smoking, dtype: float64
```

In [ ]:
```
# previous stroke history
print(df.prev_stroke.value_counts())
print(df.prev_stroke.value_counts()/df.prev_stroke.value_counts().sum()*100)
```

```
0    94
1    15
Name: prev_stroke, dtype: int64
0    86.238532
1    13.761468
Name: prev_stroke, dtype: float64
```

In [ ]:
```
# RNF213 (0: no mutation, 1: heterozygote, 2: homozygote, 3: unknown)
print(df.RNF213.value_counts().sort_index())
print(df.RNF213.value_counts().sort_index()/df.RNF213.value_counts().sum()*100)
```

```
0      6
1      1
3    102
Name: RNF213, dtype: int64
0     5.504587
1     0.917431
3    93.577982
Name: RNF213, dtype: float64
```

In [ ]:

In [ ]:
```
# Previous endovascular treatment at the main lesion (0: no, 1: yes (aborted), 2: yes (restenosis or occlusion), 3: yes (stable))
print(df.previous_EVT.value_counts().sort_index())
print()
print(df.previous_EVT.value_counts().sort_index() / df.shape[0] * 100)
```

```
0    99
1     4
2     5
3     1
Name: previous_EVT, dtype: int64

0    90.825688
1     3.669725
2     4.587156
3     0.917431
Name: previous_EVT, dtype: float64
```

In [ ]:
```
# presentation (0: aymptomatic, 1: TIA, 2: infarction, 3: hemorrhage, 4: other symptoms)
print(df.presentation.value_counts().sort_index())
print(df.presentation.value_counts().sort_index()/df.presentation.value_counts().sum()*100)
```

```
1    27
2    80
3     2
Name: presentation, dtype: int64
1    24.770642
2    73.394495
3     1.834862
Name: presentation, dtype: float64
```

In [ ]:
```
# etiology
print(df.etiology.value_counts().sort_index())
print(df.etiology.value_counts().sort_index()/df.etiology.value_counts().sum()*100)
```

```
ICAD    109
Name: etiology, dtype: int64
ICAD    100.0
Name: etiology, dtype: float64
```

In [ ]:
```
# TIA
print(df.TIA.value_counts().sort_index())
print(df.TIA.value_counts().sort_index()/df.TIA.value_counts().sum()*100)
```

```
0      9
1    100
Name: TIA, dtype: int64
0     8.256881
1    91.743119
Name: TIA, dtype: float64
```

In [ ]:
```
# multiple TIAs
print(df.multiple_TIAs.value_counts().sort_index())
print(df.multiple_TIAs.value_counts().sort_index()/df.multiple_TIAs.value_counts().sum()*100)
```

```
0    12
1    97
Name: multiple_TIAs, dtype: int64
0    11.009174
1    88.990826
Name: multiple_TIAs, dtype: float64
```

In [ ]:
```
# infarction
print(df.infarction.value_counts().sort_index())
print(df.infarction.value_counts().sort_index()/df.infarction.value_counts().sum()*100)
```

```
0    30
1    79
Name: infarction, dtype: int64
0    27.522936
1    72.477064
Name: infarction, dtype: float64
```

In [ ]:
```
# multiple infarction
print(df.multiple_infarction.value_counts().sort_index())
print(df.multiple_infarction.value_counts().sort_index()/df.multiple_infarction.value_counts().sum()*100)
```

```
0    84
1    25
Name: multiple_infarction, dtype: int64
0    77.06422
1    22.93578
Name: multiple_infarction, dtype: float64
```

In [ ]:
```
# number of days between first event date to treatment date
print(df.first_event_to_treatment_days.mean(), df.first_event_to_treatment_days.std(), df.first_event_to_treatment_days.min(), df.first_event_to_treatment_days.max())

# number of weeks between first event date to treatment date
print(df.first_event_to_treatment_weeks.mean(), df.first_event_to_treatment_weeks.std(), df.first_event_to_treatment_weeks.min(), df.first_event_to_treatment_weeks.max())

# number of months between first event date to treatment date
print(df.first_event_to_treatment_months.mean(), df.first_event_to_treatment_months.std(), df.first_event_to_treatment_months.min(), df.first_event_to_treatment_months.max())
```

```
209.14678899082568 296.3728203190252 5.0 2110.0
29.880733944954127 42.29257109501275 1.0 301.0
7.4495412844036695 10.588678648838693 0.0 75.0
```

In [ ]:
```
# number of months between first event date to treatment date
print(df.first_event_to_treatment_months.mean(), df.first_event_to_treatment_months.std(), df.first_event_to_treatment_months.min(), df.first_event_to_treatment_months.max())
```

```
7.4495412844036695 10.588678648838693 0.0 75.0
```

In [ ]:
```
# number of days between last event date to treatment date
print(df.last_event_to_treatment_days.mean(), df.last_event_to_treatment_days.std(), df.last_event_to_treatment_days.min(), df.last_event_to_treatment_days.max())

# number of weeks between last event date to treatment date
print(df.last_event_to_treatment_weeks.mean(), df.last_event_to_treatment_weeks.std(), df.last_event_to_treatment_weeks.min(), df.last_event_to_treatment_weeks.max())

# number of months between last event date to treatment date
print(df.last_event_to_treatment_months.mean(), df.last_event_to_treatment_months.std(), df.last_event_to_treatment_months.min(), df.last_event_to_treatment_months.max())
```

```
50.8348623853211 51.919366571755944 0.0 315.0
7.293577981651376 7.400328761833112 0.0 45.0
1.7339449541284404 1.913123303584919 0.0 11.0
```

In [ ]:
```
# number of days between last event date to treatment date
print(df.last_event_to_treatment_days.median(),  np.quantile(df.last_event_to_treatment_days, 0.75) - np.quantile(df.last_event_to_treatment_days, 0.25))
print(np.quantile(df.last_event_to_treatment_days, 0.25), np.quantile(df.last_event_to_treatment_days, 0.75))
# number of weeks between last event date to treatment date
print(df.last_event_to_treatment_weeks.median(), np.quantile(df.last_event_to_treatment_weeks, 0.75) - np.quantile(df.last_event_to_treatment_weeks, 0.25))

# number of months between last event date to treatment date
print(df.last_event_to_treatment_months.median(), np.quantile(df.last_event_to_treatment_months, 0.75) - np.quantile(df.last_event_to_treatment_months, 0.25))
```

```
37.0 59.0
13.0 72.0
5.0 8.0
1.0 3.0
```

In [ ]:

In [ ]:
```
# initial and preoperative mRS score
initial_preop_mRS_df = pd.merge(df.initial_mRS.value_counts().sort_index()/df.initial_mRS.value_counts().sum()*100, df.preop_mRS.value_counts().sort_index()/df.preop_mRS.value_counts().sum()*100, left_index=True, right_index=True).T
initial_preop_mRS_df
```

Out[ ]:

	1.0	2.0	3.0	4.0	5.0
initial_mRS	22.018349	38.532110	31.192661	7.339450	0.917431
preop_mRS	21.100917	39.449541	26.605505	11.926606	0.917431

In [ ]:
```
df.preop_mRS.median(), df.preop_mRS.quantile([0.25, 0.75])
```

Out[ ]:
```
(2.0,
 0.25    2.0
 0.75    3.0
 Name: preop_mRS, dtype: float64)
```

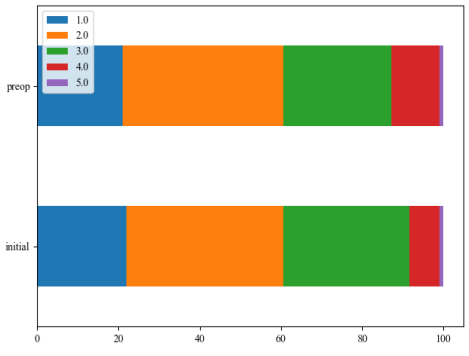
In [ ]:
```
df.preop_mRS.value_counts().sort_index()
```

Out[ ]:
```
1.0    23
2.0    43
3.0    29
4.0    13
5.0     1
Name: preop_mRS, dtype: int64
```

In [ ]:
```
df.preop_mRS.value_counts().sort_index()/df.preop_mRS.value_counts().sum()*100
```

Out[ ]:
```
1.0    21.100917
2.0    39.449541
3.0    26.605505
4.0    11.926606
5.0     0.917431
Name: preop_mRS, dtype: float64
```

In [ ]:
```
ax = initial_preop_mRS_df.plot(kind='barh', stacked=True)
ax.set_yticklabels(['initial', 'preop'])
ax.legend()
plt.tight_layout()
plt.show()
```

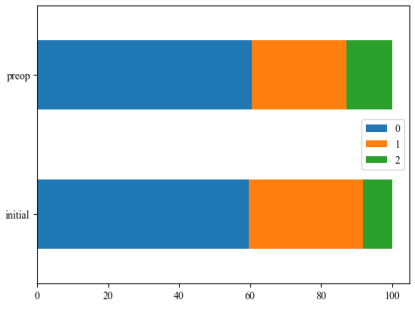


In [ ]:
```
# initial and preoperative mRS grade
initial_preop_mRS_grade_df = pd.merge(df.initial_mRS_grade.value_counts().sort_index()/df.initial_mRS_grade.value_counts().sum()*100, df.preop_mRS_grade.value_counts().sort_index()/df.preop_mRS_grade.value_counts().sum()*100, left_index=True, right_index=True).T
initial_preop_mRS_grade_df
```

Out[ ]:

	0	1	2
initial_mRS_grade	59.633028	32.110092	8.256881
preop_mRS_grade	60.550459	26.605505	12.844037

In [ ]:
```
ax = initial_preop_mRS_grade_df.plot(kind='barh', stacked=True)
ax.set_yticklabels(['initial', 'preop'])
plt.show()
```



In [ ]:
```
# initial and preoperative NIHSS
df.preop_NIHSS.median(), df.preop_NIHSS.quantile([0.25, 0.75]), df.preop_NIHSS.min(), df.preop_NIHSS.max()
```

Out[ ]:
```
(3.0,
 0.25    1.0
 0.75    6.0
 Name: preop_NIHSS, dtype: float64,
 0.0,
 16.0)
```

In [ ]:
```
# initial and preoperative NIHSS grade
initial_preop_NIHSS_grade_df = pd.merge(df.initial_NIHSS_grade.value_counts().sort_index()/df.initial_NIHSS_grade.value_counts().sum()*100, df.preop_NIHSS_grade.value_counts().sort_index()/df.preop_NIHSS_grade.value_counts().sum()*100, left_index=True, right_index=True).T
initial_preop_NIHSS_grade_df
```

Out[ ]:

	0	1	2
initial_NIHSS_grade	63.302752	35.779817	0.917431
preop_NIHSS_grade	61.467890	34.862385	3.669725

In [ ]:
```
df.preop_NIHSS_grade.value_counts().sort_index()
```

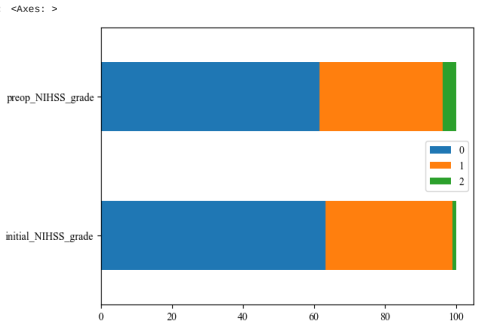
Out[ ]:
```
0    67
1    38
2     4
Name: preop_NIHSS_grade, dtype: int64
```

In [ ]:
```
df.preop_NIHSS_grade.value_counts().sort_index()/df.preop_NIHSS_grade.value_counts().sum()*100
```

Out[ ]:
```
0    61.467890
1    34.862385
2     3.669725
Name: preop_NIHSS_grade, dtype: float64
```

In [ ]:
```
initial_preop_NIHSS_grade_df.plot(kind='barh', stacked=True)
```

Out[ ]:
```
<Axes: >
```



In [ ]:

In [ ]:

Radiological findings

In [ ]:
```
# Diffusion MRI infarction pattern
df.diffusion_infarction_pattern.value_counts()
```

Out[ ]:
```
5      32
4      26
8      19
9       9
3       7
1       6
7       2
6       2
2       2
1,4     1
1,5     1
11      1
10      1
Name: diffusion_infarction_pattern, dtype: int64
```

In [ ]:
```
df.diffusion_infarction_pattern.str.get_dummies(sep=',')
```

Out[ ]:

	1	10	11	2	3	4	5	6	7	8	9
1	0	0	0	0	0	0	0	0	0	1	0
2	1	0	0	0	0	1	0	0	0	0	0
3	0	0	0	0	0	0	1	0	0	0	0
4	0	0	0	0	0	0	1	0	0	0	0
5	0	0	0	0	0	0	1	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...
105	0	0	0	0	1	0	0	0	0	0	0

106	0	0	0	0	0	1	0	0	0	0	0
107	0	0	0	0	0	1	0	0	0	0	0
108	0	0	0	0	0	0	1	0	0	0	0
109	0	0	0	0	0	1	0	0	0	0	0

109 rows × 11 columns

In [ ]:
```
infarct_pattern_list = ['infarct_lobar', 'infarct_ICH', 'infarct_IVH', 'infarct_small_cortical', 'infarct_cortical_BDZ', 'infarct_internal_BDZ', 'infarct_cortical_internal_BDZ', 'infarct_large_deep',
  'infarct_small_deep','infarct_old', 'infarct_no_infarct']
```

In [ ]:
```
df[['infarct_lobar', 'infarct_ICH', 'infarct_IVH', 'infarct_small_cortical', 'infarct_cortical_BDZ', 'infarct_internal_BDZ', 'infarct_cortical_internal_BDZ', 'infarct_large_deep',
  'infarct_small_deep','infarct_old', 'infarct_no_infarct']] = df.diffusion_infarction_pattern.str.get_dummies(sep=',')
```

In [ ]:
```
df[infarct_pattern_list].sum()
```

Out[ ]:
```
infarct_lobar                     8
infarct_ICH                       1
infarct_IVH                       1
infarct_small_cortical            2
infarct_cortical_BDZ              7
infarct_internal_BDZ             27
infarct_cortical_internal_BDZ    33
infarct_large_deep                2
infarct_small_deep                2
infarct_old                      19
infarct_no_infarct                9
dtype: int64
```

In [ ]:
```
df[infarct_pattern_list].sum().sum()
```

Out[ ]:
```
111
```

In [ ]:
```
df[infarct_pattern_list].sum()/df.shape[0]*100
```

Out[ ]:
```
infarct_lobar                     7.339450
infarct_ICH                       0.917431
infarct_IVH                       0.917431
infarct_small_cortical            1.834862
infarct_cortical_BDZ              6.422018
infarct_internal_BDZ             24.770642
infarct_cortical_internal_BDZ    30.275229
infarct_large_deep                1.834862
infarct_small_deep                1.834862
infarct_old                      17.431193
infarct_no_infarct                8.256881
dtype: float64
```

In [ ]:

In [ ]:
```
# Side of Main lesion
print(df.main_lesion_site.str.split('.').str[0].value_counts())
print()
print(df.main_lesion_site.str.split('.').str[0].value_counts()/df.main_lesion_site.str.split('.').str[0].value_counts().sum()*100)
```

```
Lt    59
Rt    50
Name: main_lesion_site, dtype: int64

Lt    54.12844
Rt    45.87156
Name: main_lesion_site, dtype: float64
```

In [ ]:
```
# Site of Main lesion
print(df.main_lesion_site.str.split('.').str[1].value_counts())
print()
print(df.main_lesion_site.str.split('.').str[1].value_counts()/df.main_lesion_site.str.split('.').str[1].value_counts().sum()*100)
```

```
pICA    35
dICA    32
M1t     22
M1o     18
M2st     2
Name: main_lesion_site, dtype: int64

pICA    32.110092
dICA    29.357798
M1t     20.183486
M1o     16.513761
M2st     1.834862
Name: main_lesion_site, dtype: float64
```

In [ ]:
```
# ICA lesion type
df.ICA_lesion_type.value_counts()
```

Out[ ]:
```
proximal         35
communicating    17
cavernous         8
ophthalmic        4
terminal          2
petrous           1
Name: ICA_lesion_type, dtype: int64
```

In [ ]:
```
df.ICA_lesion_type.value_counts()/df.ICA_lesion_type.value_counts().sum()*100
```

Out[ ]:
```
proximal         52.238806
communicating    25.373134
cavernous        11.940299
ophthalmic        5.970149
terminal          2.985075
petrous           1.492537
Name: ICA_lesion_type, dtype: float64
```

In [ ]:
```
# MCA lesion type
print(df.MCA_lesion_type.value_counts().sort_index())
print()
print(df.MCA_lesion_type.value_counts().sum())
```

```
0    18
1    22
3     2
Name: MCA_lesion_type, dtype: int64

42
```

In [ ]:
```
df.MCA_lesion_type.value_counts().sort_index()/df.MCA_lesion_type.value_counts().sort_index().sum()*100
```

Out[ ]:
```
0    42.857143
1    52.380952
3     4.761905
Name: MCA_lesion_type, dtype: float64
```

In [ ]:

In [ ]:
```
# Stenosis or Occlusion (0:occlusion, 1:stenosis)
df.lesion_stenosis_occlusion.value_counts()
```

Out[ ]:
```
0    94
1    15
Name: lesion_stenosis_occlusion, dtype: int64
```

In [ ]:
```
pd.Series([15,9,0,9,6,94,58,35,23,36])/109*100
```

Out[ ]:
```
0    13.761468
1     8.256881
2     0.000000
3     8.256881
4     5.504587
5    86.238532
6    53.211009
7    32.110092
8    21.100917
9    33.027523
dtype: float64
```

In [ ]:
```
# Stenosis or occlusion and main lesion site (0:occlusion, 1:stenosis)
df.groupby(['lesion_stenosis_occlusion','ICA_lesion_type'])['ICA_lesion_type'].count()
```

Out[ ]:
```
lesion_stenosis_occlusion  ICA_lesion_type
0                          cavernous          8
                           communicating     13
                           ophthalmic         1
                           petrous            1
                           proximal          35
1                          communicating      4
                           ophthalmic         3
                           terminal           2
Name: ICA_lesion_type, dtype: int64
```

In [ ]:
```
# MCA lesion type (0: M1 origin, 1: M1 trunk, 2: M1 bifurcation,3: M2 post-bifurcation)
df.groupby(['lesion_stenosis_occlusion','MCA_lesion_type'])['MCA_lesion_type'].count()
```

Out[ ]:
```
lesion_stenosis_occlusion  MCA_lesion_type
0                          0                15
                           1                20
                           3                 1
1                          0                 3
                           1                 2
                           3                 1
Name: MCA_lesion_type, dtype: int64
```

In [ ]:
```
# Degree of stenosis among stenosis lesions
d_stenosis_s = df[df.lesion_stenosis_occlusion==1]['degree_stenosis']
d_stenosis_s
```

Out[ ]:
```
3       80.1
4       74.1
5       79.5
11        79
13        85
22        90
35      74.1
43      69.4
46      60.3
57        90
70        74
95     75.35
100    71.23
103    56.32
105    73.02
Name: degree_stenosis, dtype: object
```

In [ ]:
```
d_stenosis_s.median(), d_stenosis_s.std(), d_stenosis_s.min(), d_stenosis_s.max()
```

Out[ ]:
```
(74.1, 9.360938750238368, 56.32, 90)
```

In [ ]:
```
# Stenosis Mori type
df.Mori_type.value_counts().sort_index()
```

Out[ ]:
```
A    2
B    4
C    9
Name: Mori_type, dtype: int64
```

In [ ]:
```
df.Mori_type.value_counts().sort_index()/df.Mori_type.value_counts().sort_index().sum()*100
```

Out[ ]:
```
A    13.333333
B    26.666667
C    60.000000
Name: Mori_type, dtype: float64
```

In [ ]:

In [ ]:
```
# Single lesion or Multiple lesions
df.single_ICAD.value_counts().sort_index()
```

Out[ ]:
```
0    13
1    96
Name: single_ICAD, dtype: int64
```

In [ ]:
```
df.single_ICAD.value_counts().sort_index()/df.single_ICAD.value_counts().sort_index().sum()*100
```

Out[ ]:
```
0    11.926606
1    88.073394
Name: single_ICAD, dtype: float64
```

In [ ]:
```
# Bilaterality of multiple lesions
df.multiple_ICAD_bilateral.value_counts()
```

Out[ ]:
```
1    90
0    19
Name: multiple_ICAD_bilateral, dtype: int64
```

In [ ]:
```
df.multiple_ICAD_bilateral.value_counts()/109*100
```

Out[ ]:
```
1    82.568807
0    17.431193
Name: multiple_ICAD_bilateral, dtype: float64
```

In [ ]:
```
# Number of lesions with stenosis over 50% in multiple steno-occlusive lesions
df[df.single_ICAD==1]['n_intracranial_lesion_over_stenosis_50'].value_counts()/df[df.single_ICAD==1]['n_intracranial_lesion_over_stenosis_50'].value_counts().sum()*100
```

Out[ ]:
```
1.0    43.750000
2.0    41.666667
3.0    11.458333
4.0     3.125000
Name: n_intracranial_lesion_over_stenosis_50, dtype: float64
```

In [ ]:
```
df[df.single_ICAD==1]['n_intracranial_lesion_over_stenosis_50'].value_counts()
```

Out[ ]:
```
1.0    42
2.0    40
3.0    11
4.0     3
Name: n_intracranial_lesion_over_stenosis_50, dtype: int64
```

In [ ]:

In [ ]:
```
# Type of collaterals (0: poor collateral, 1: Willisian, 2: leptomeningeal, 3: basal collateral, 4: transdural)
df.collateral_status.str.get_dummies(sep=',')
```

Out[ ]:

	1	2	3	4
1	1	1	0	0
2	1	1	0	1
3	0	1	0	1
4	0	1	0	0
5	0	1	0	0
...	...	...	...	...
105	0	1	1	0
106	1	1	1	0
107	1	1	0	0
108	1	1	0	1
109	1	1	0	1

109 rows × 4 columns

In [ ]:
```
collateral_status_list = ['collateral_Willisian', 'collateral_leptomeningeal', 'collateral_basal', 'collateral_transdural']
```

In [ ]:
```
df[collateral_status_list] = df.collateral_status.str.get_dummies(sep=',')
```

In [ ]:
```
df[collateral_status_list].sum()
```

Out[ ]:
```
collateral_Willisian          82
collateral_leptomeningeal    109
collateral_basal              32
collateral_transdural         40
dtype: int64
```

In [ ]:
```
df[collateral_status_list].sum()/df.shape[0]*100
```

Out[ ]:
```
collateral_Willisian          75.229358
collateral_leptomeningeal    100.000000
collateral_basal              29.357798
collateral_transdural         36.697248
dtype: float64
```

In [ ]:

In [ ]:

Hemodynamic evaluation

In [ ]:
```
# Preoperative hemodynamic status evaluation modality (0: perfusion SPECT, 1: diamox SPECT, 2: MR perfusion, 3: CT perfusion, 4: PET, 5: no w/u)
df.preop_image_hemodynamic_insufficiency.value_counts().sort_index()
```

Out[ ]:
```
0     5
1    93
2    10
5     1
Name: preop_image_hemodynamic_insufficiency, dtype: int64
```

In [ ]:
```
df.preop_image_hemodynamic_insufficiency.value_counts().sort_index()/df.preop_image_hemodynamic_insufficiency.value_counts().sort_index().sum()*100
```

Out[ ]:
```
0     4.587156
1    85.321101
2     9.174312
5     0.917431
Name: preop_image_hemodynamic_insufficiency, dtype: float64
```

In [ ]:
```
df.preop_image_hemodynamic_insufficiency.value_counts().sort_index().sum()
```

Out[ ]:
```
109
```

In [ ]:
```
# Postoperative (within 1 month) hemodynamic status evaluation modality (0: perfusion SPECT, 1: diamox SPECT, 2: MR perfusion, 3: CT perfusion, 4: PET, 5: no w/u)
df.postop_image_hemodynamic_insufficiency.value_counts().sort_index()
```

Out[ ]:
```
0     2
1    40
2     4
3    63
Name: postop_image_hemodynamic_insufficiency, dtype: int64
```

In [ ]:
```
df.postop_image_hemodynamic_insufficiency.value_counts().sort_index()/df.postop_image_hemodynamic_insufficiency.value_counts().sort_index().sum()*100
```

Out[ ]:
```
0     1.834862
1    36.697248
2     3.669725
3    57.798165
Name: postop_image_hemodynamic_insufficiency, dtype: float64
```

In [ ]:

In [ ]:
```
# Postoperative (within 1 month) hemodynamic status change (0: worsen, 1: no significant change, 2: slight/focal improvement, 3: moderate/lobar improvement, 4: significant/territorial improvement, 5: no f/u)
df.postop_hemodynamic_insufficiency_grade.value_counts().sort_index()
```

Out[ ]:
```
1     5
2    31
3    53
4    20
Name: postop_hemodynamic_insufficiency_grade, dtype: int64
```

In [ ]:
```
df.postop_hemodynamic_insufficiency_grade.value_counts().sort_index()/df.postop_hemodynamic_insufficiency_grade.value_counts().sort_index().sum()*100
```

Out[ ]:
```
1     4.587156
2    28.440367
3    48.623853
4    18.348624
Name: postop_hemodynamic_insufficiency_grade, dtype: float64
```

In [ ]:
```
# Postoperative 3 years hemodynamic status evaluation modality (0: perfusion SPECT, 1: diamox SPECT, 2: MR perfusion, 3: CT perfusion, 4: PET, 5: no w/u)
df['3y_image_hemodynamic_insufficiency'].value_counts()
```

Out[ ]:
```
1    38
2    27
3    26
5    17
Name: 3y_image_hemodynamic_insufficiency, dtype: int64
```

In [ ]:
```
df['3y_image_hemodynamic_insufficiency'].value_counts().sum()
```

Out[ ]:
```
108
```

In [ ]:
```
# Postoperative 3 years hemodynamic status change (0: worsen, 1: no significant change, 2: slight/focal improvement, 3: moderate/lobar improvement, 4: significant/territorial improvement, 5: no f/u)
df['3y_image_hemodynamic_insufficiency'].value_counts()/df['3y_image_hemodynamic_insufficiency'].value_counts().sum()*100
```

Out[ ]:
```
1    35.185185
2    25.000000
3    24.074074
5    15.740741
Name: 3y_image_hemodynamic_insufficiency, dtype: float64
```

In [ ]:
```
df['3y_hemodynamic_insufficiency_grade'].value_counts().sort_index()/df['3y_hemodynamic_insufficiency_grade'].value_counts().sort_index().sum()*100
```

Out[ ]:
```
0     3.703704
1     0.925926
2     2.777778
3    24.074074
4    52.777778
5    15.740741
Name: 3y_hemodynamic_insufficiency_grade, dtype: float64
```

In [ ]:

In [ ]:

Bypass outcomes

In [ ]:
```
# Bypass type (0: single-barrel, 1: double_barrel, 2: interposition (high flow), 3: interposition (low flow), 4: direct + indirect)
df.bypass_type.value_counts().sort_index()
```

Out[ ]:
```
0    97
1    10
4     2
Name: bypass_type, dtype: int64
```

In [ ]:
```
df.bypass_type.value_counts().sort_index()/df.bypass_type.value_counts().sort_index().sum()*100
```

Out[ ]:
```
0    88.990826
1     9.174312
4     1.834862
Name: bypass_type, dtype: float64
```

In [ ]:

In [ ]:
```
# Initial bypass patency
df.initial_bypass_patency.value_counts().sort_index()
```

Out[ ]:
```
0     1
1     5
2    91
3     9
4     2
5     1
Name: initial_bypass_patency, dtype: int64
```

In [ ]:
```
df.initial_bypass_patency.value_counts().sort_index().sum()
```

Out[ ]:
```
109
```

In [ ]:
```
df.initial_bypass_patency.value_counts().sort_index()/df.initial_bypass_patency.value_counts().sort_index().sum()*100
```

Out[ ]:
```
0     0.917431
1     4.587156
2    83.486239
3     8.256881
4     1.834862
5     0.917431
Name: initial_bypass_patency, dtype: float64
```

In [ ]:

In [ ]:
```
# 1 year bypass patency (0: failure, 1: poor, 2: good, 3: good/good, 4: good/poor, 5: poor/poor)
df['1y_bypass_patency'].value_counts().sort_index()
```

Out[ ]:
```
0     1
1     7
2    84
3    10
4     1
Name: 1y_bypass_patency, dtype: int64
```

In [ ]:
```
df['1y_bypass_patency'].value_counts().sort_index().sum()
```

Out[ ]:
```
103
```

In [ ]:
```
df['1y_bypass_patency'].value_counts().sort_index()/df['1y_bypass_patency'].value_counts().sort_index().sum()*100
```

Out[ ]:
```
0     0.970874
1     6.796117
2    81.553398
3     9.708738
4     0.970874
Name: 1y_bypass_patency, dtype: float64
```

In [ ]:

In [ ]:
```
df[df['1y_bypass_patency'].isnull()]
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	infarct_internal_BDZ	infarct_cortical_internal_BDZ	infarct_large_deep	infarct_small_deep	infarct_old	infarct_no_infarct	collateral_Willisian	collateral_leptomeningeal	collateral_l
2	2018-08-30	2.0	49.0	1	Lt. M1 and Rt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 2 months expired due to extensiv...	0	NaT	NaN	...	1	0	0	0	0	0	1	1	
22	2015-11-17	22.0	63.0	1	Bilateral M1 and ICA stenosis	EIAB, Lt. (double barrel STA-M4 bypass)	immediate postoperative bypass failure lead to...	1	2015-11-17	0.0	...	0	1	0	0	0	0	1	1	
28	2014-05-17	28.0	74.0	0	Lt. ICA occlusion	STA-M3 bypass, Lt.	postoperative 1 day ipsilateral MCA territory ...	1	2014-05-18	1.0	...	1	0	0	0	0	0	1	1	
60	2010-11-10	60.0	72.0	0	Rt. distal ICA and M1 stenosis	EIAB, Rt. (STA-M4)	2 vessel disease (high perioperative risk grou...	1	2010-11-13	3.0	...	0	0	0	0	0	0	0	1	
70	2009-08-17	70.0	45.0	1	Lt. M1 and A1 stenosis and Rt. ICA stenosis	EIAB, Lt (STA-M4)	postoperative 2 days contralateral PCA-MCA ter...	0	NaT	NaN	...	0	0	0	0	1	0	1	1	
109	2017-10-19	109.0	58.0	0	Bilateral ICA occlusion	EIAB, Lt. (STA-M4)	Hemodynamic insufficiency (Rt. MCA territory >...	1	2017-10-22	3.0	...	1	0	0	0	0	0	1	1	

6 rows × 279 columns

In [ ]:

In [ ]:
```
# 3 year bypass patency (0: failure, 1: poor, 2: good, 3: good/good, 4: good/poor, 5: poor/poor)
df['3y_bypass_patency'].value_counts().sort_index()
```

Out[ ]:
```
0     1
1     8
2    83
3    10
4     1
Name: 3y_bypass_patency, dtype: int64
```

In [ ]:
```
df['3y_bypass_patency'].value_counts().sort_index().sum()
```

Out[ ]:
```
103
```

In [ ]:
```
df['3y_bypass_patency'].value_counts().sort_index()/df['3y_bypass_patency'].value_counts().sort_index().sum()*100
```

Out[ ]:
```
0     0.970874
1     7.766990
2    80.582524
3     9.708738
4     0.970874
Name: 3y_bypass_patency, dtype: float64
```

In [ ]:

In [ ]:

In [ ]:
```
# 5 year bypass patency
df['5y_bypass_patency'].value_counts().sort_index()
```

Out[ ]:
```
0     1
1     8
2    83
3    10
4     1
Name: 5y_bypass_patency, dtype: int64
```

In [ ]:
```
df['5y_bypass_patency'].value_counts().sort_index().sum()
```

Out[ ]:
```
103
```

In [ ]:
```
df['5y_bypass_patency'].value_counts().sort_index()/df['5y_bypass_patency'].value_counts().sort_index().sum()*100
```

Out[ ]:
```
0     0.970874
1     7.766990
2    80.582524
3     9.708738
4     0.970874
Name: 5y_bypass_patency, dtype: float64
```

In [ ]:

In [ ]:

Clinical outcomes

In [ ]:
```
# Mean follow-up period (days)
df.op_date_to_last_fu_date_days.mean(), df.op_date_to_last_fu_date_days.std(), df.op_date_to_last_fu_date_days.min(), df.op_date_to_last_fu_date_days.max()
```

Out[ ]:
```
(3314.5229357798166, 1135.8719166539656, 55.0, 5566.0)
```

In [ ]:
```
# Mean follow-up period (weeks)
df.op_date_to_last_fu_date_weeks.mean(), df.op_date_to_last_fu_date_weeks.std(), df.op_date_to_last_fu_date_weeks.min(), df.op_date_to_last_fu_date_weeks.max()
```

Out[ ]:
```
(473.4678899082569, 162.24259207007697, 8.0, 795.0)
```

In [ ]:
```
# Mean follow-up period (months)
df.op_date_to_last_fu_date_months.mean(), df.op_date_to_last_fu_date_months.std(), df.op_date_to_last_fu_date_months.min(), df.op_date_to_last_fu_date_months.max()
```

Out[ ]:
```
(118.35779816513761, 40.647200016496924, 2.0, 199.0)
```

In [ ]:
```
# any stroke within 30 days
df[df.any_postop_stroke==1][(diff_days(df[df.any_postop_stroke==1]['op_date'], df[df.any_postop_stroke==1]['any_postop_stroke_date'])<31)]
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	infarct_internal_BDZ	infarct_cortical_internal_BDZ	infarct_large_deep	infarct_small_deep	infarct_old	infarct_no_infarct	collateral_Willisian	collateral_leptomeningeal	collateral_
22	2015-11-17	22.0	63.0	1	Bilateral M1 and ICA stenosis	EIAB, Lt. (double barrel STA-M4 bypass)	immediate postoperative bypass failure lead to...	1	2015-11-17	0.0	...	0	1	0	0	0	0	1	1	
28	2014-05-17	28.0	74.0	0	Lt. ICA occlusion	STA-M3 bypass, Lt.	postoperative 1 day ipsilateral MCA territory ...	1	2014-05-18	1.0	...	1	0	0	0	0	0	1	1	
39	2013-05-08	39.0	49.0	0	Lt. ICA occlusion (s/p stent insertion d/t Lt....	OA-M4 bypass, Lt.	angioplasty and stenting at Lt. V4\nimmediate ...	1	2013-05-21	13.0	...	0	0	0	0	1	0	1	1	
60	2010-11-10	60.0	72.0	0	Rt. distal ICA and M1 stenosis	EIAB, Rt. (STA-M4)	2 vessel disease (high perioperative risk grou...	1	2010-11-13	3.0	...	0	0	0	0	0	0	0	1	

70	2009-08-17	70.0	45.0	1	Lt. M1 and A1 stenosis and Rt. ICA stenosis	EIAB, Lt (STA-M4)	postoperative 2 days contralateral PCA-MCA ter...	0	NaT	NaN	...	0	0	0	0	1	0	1	1
108	2010-04-05	108.0	67.0	0	Lt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 3 days contralateral BDZ infarct...	0	NaT	NaN	...	0	1	0	0	0	0	1	1
109	2017-10-19	109.0	58.0	0	Bilateral ICA occlusion	EIAB, Lt. (STA-M4)	Hemodynamic insufficiency (Rt. MCA territory >...	1	2017-10-22	3.0	...	1	0	0	0	0	0	1	1

7 rows × 279 columns

In [ ]:

In [ ]:
```
# Any stroke within 2 years
df[df.any_postop_stroke==1][diff_days(df[df.any_postop_stroke==1]['op_date'], df[df.any_postop_stroke==1]['any_postop_stroke_date'])<365*2+1]
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	infarct_internal_BDZ	infarct_cortical_internal_BDZ	infarct_large_deep	infarct_small_deep	infarct_old	infarct_no_infarct	collateral_Willisian	collateral_leptomeningeal	collateral_
2	2018-08-30	2.0	49.0	1	Lt. M1 and Rt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 2 months expired due to extensiv...	0	NaT	NaN	...	1	0	0	0	0	0	1	1
17	2015-01-12	17.0	62.0	0	Lt. ICA occlusion	EIAB, Lt (STA-M4 bypass)	postoperative 5 months focal infarction at Lt....	1	2015-06-11	150.0	...	0	1	0	0	0	0	1	1
22	2015-11-17	22.0	63.0	1	Bilateral M1 and ICA stenosis	EIAB, Lt. (double barrel STA-M4 bypass)	immediate postoperative bypass failure lead to...	1	2015-11-17	0.0	...	0	1	0	0	0	0	1	1
28	2014-05-17	28.0	74.0	0	Lt. ICA occlusion	STA-M3 bypass, Lt.	postoperative 1 day ipsilateral MCA territory ...	1	2014-05-18	1.0	...	1	0	0	0	0	0	1	1
39	2013-05-08	39.0	49.0	0	Lt. ICA occlusion (s/p stent insertion d/t Lt....	OA-M4 bypass, Lt.	angioplasty and stenting at Lt. V4\nimmediate ...	1	2013-05-21	13.0	...	0	0	0	0	1	0	1	1
46	2011-03-02	46.0	56.0	0	Rt. ICA stenosis	STA-MCA bypass, Rt.	postoperative 1.5 years ipsilateral ACA territ...	0	NaT	NaN	...	0	1	0	0	0	0	1	1
53	2011-03-28	53.0	68.0	0	Lt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 5 months contralateral multifoca...	0	NaT	NaN	...	0	0	0	0	1	0	1	1
60	2010-11-10	60.0	72.0	0	Rt. distal ICA and M1 stenosis	EIAB, Rt. (STA-M4)	2 vessel disease (high perioperative risk grou...	1	2010-11-13	3.0	...	0	0	0	0	0	0	0	1
70	2009-08-17	70.0	45.0	1	Lt. M1 and A1 stenosis and Rt. ICA stenosis	EIAB, Lt (STA-M4)	postoperative 2 days contralateral PCA-MCA ter...	0	NaT	NaN	...	0	0	0	0	1	0	1	1
101	2007-08-06	101.0	60.0	0	Lt. distal ICA occlusion and Rt. distal ICA oc...	EIAB, Lt. (STA-M4)	postoperative 3 months asymptomatic ipsilatera...	1	2008-01-03	150.0	...	0	1	0	0	0	0	1	1
108	2010-04-05	108.0	67.0	0	Lt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 3 days contralateral BDZ infarct...	0	NaT	NaN	...	0	1	0	0	0	0	1	1
109	2017-10-19	109.0	58.0	0	Bilateral ICA occlusion	EIAB, Lt. (STA-M4)	Hemodynamic insufficiency (Rt. MCA territory >...	1	2017-10-22	3.0	...	1	0	0	0	0	0	1	1

12 rows × 279 columns

In [ ]:
```
# Any stroke within 5 years
df[df.any_postop_stroke==1][diff_days(df[df.any_postop_stroke==1]['op_date'], df[df.any_postop_stroke==1]['any_postop_stroke_date'])<365*5+1]
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	infarct_internal_BDZ	infarct_cortical_internal_BDZ	infarct_large_deep	infarct_small_deep	infarct_old	infarct_no_infarct	collateral_Willisian	collateral_leptomeningeal	collateral_
2	2018-08-30	2.0	49.0	1	Lt. M1 and Rt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 2 months expired due to extensiv...	0	NaT	NaN	...	1	0	0	0	0	0	1	1
17	2015-01-12	17.0	62.0	0	Lt. ICA occlusion	EIAB, Lt (STA-M4 bypass)	postoperative 5 months focal infarction at Lt....	1	2015-06-11	150.0	...	0	1	0	0	0	0	1	1
22	2015-11-17	22.0	63.0	1	Bilateral M1 and ICA stenosis	EIAB, Lt. (double barrel STA-M4 bypass)	immediate postoperative bypass failure lead to...	1	2015-11-17	0.0	...	0	1	0	0	0	0	1	1
28	2014-05-17	28.0	74.0	0	Lt. ICA occlusion	STA-M3 bypass, Lt.	postoperative 1 day ipsilateral MCA territory ...	1	2014-05-18	1.0	...	1	0	0	0	0	0	1	1
36	2013-03-15	36.0	55.0	0	ICA occlusion, bilateral	STA-M4 bypass, Rt.	postoperative 3 years contralateral MCA territ...	0	NaT	NaN	...	1	0	0	0	0	0	1	1
37	2013-06-19	37.0	56.0	0	ICA stenosis, bilateral (s/p STA-M2 bypass, Rt...	STA-M4 bypass, Lt.	postoperative 3 years ipsilateral MCA territor...	1	2017-10-25	1589.0	...	1	0	0	0	0	0	1	1
39	2013-05-08	39.0	49.0	0	Lt. ICA occlusion (s/p stent insertion d/t Lt....	OA-M4 bypass, Lt.	angioplasty and stenting at Lt. V4\nimmediate ...	1	2013-05-21	13.0	...	0	0	0	0	1	0	1	1
46	2011-03-02	46.0	56.0	0	Rt. ICA stenosis	STA-MCA bypass, Rt.	postoperative 1.5 years ipsilateral ACA territ...	0	NaT	NaN	...	0	1	0	0	0	0	1	1
53	2011-03-28	53.0	68.0	0	Lt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 5 months contralateral multifoca...	0	NaT	NaN	...	0	0	0	0	1	0	1	1
60	2010-11-10	60.0	72.0	0	Rt. distal ICA and M1 stenosis	EIAB, Rt. (STA-M4)	2 vessel disease (high perioperative risk grou...	1	2010-11-13	3.0	...	0	0	0	0	0	0	0	1
70	2009-08-17	70.0	45.0	1	Lt. M1 and A1 stenosis and Rt. ICA stenosis	EIAB, Lt (STA-M4)	postoperative 2 days contralateral PCA-MCA ter...	0	NaT	NaN	...	0	0	0	0	1	0	1	1
101	2007-08-06	101.0	60.0	0	Lt. distal ICA occlusion and Rt. distal ICA oc...	EIAB, Lt. (STA-M4)	postoperative 3 months asymptomatic ipsilatera...	1	2008-01-03	150.0	...	0	1	0	0	0	0	1	1
108	2010-04-05	108.0	67.0	0	Lt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 3 days contralateral BDZ infarct...	0	NaT	NaN	...	0	1	0	0	0	0	1	1
109	2017-10-19	109.0	58.0	0	Bilateral ICA occlusion	EIAB, Lt. (STA-M4)	Hemodynamic insufficiency (Rt. MCA territory >...	1	2017-10-22	3.0	...	1	0	0	0	0	0	1	1

14 rows × 279 columns

In [ ]:

In [ ]:
```
# Ipsilateral stroke within postoperative 30 days
df[df.ipsilateral_postop_stroke==1][(diff_days(df[df.ipsilateral_postop_stroke==1]['op_date'], df[df.ipsilateral_postop_stroke==1]['ipsilateral_postop_stroke_date'])<31)]
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	infarct_internal_BDZ	infarct_cortical_internal_BDZ	infarct_large_deep	infarct_small_deep	infarct_old	infarct_no_infarct	collateral_Willisian	collateral_leptomeningeal	collateral_
22	2015-11-17	22.0	63.0	1	Bilateral M1 and ICA stenosis	EIAB, Lt. (double barrel STA-M4 bypass)	immediate postoperative bypass failure lead to...	1	2015-11-17	0.0	...	0	1	0	0	0	0	1	1
28	2014-05-17	28.0	74.0	0	Lt. ICA occlusion	STA-M3 bypass, Lt.	postoperative 1 day ipsilateral MCA territory ...	1	2014-05-18	1.0	...	1	0	0	0	0	0	1	1
39	2013-05-08	39.0	49.0	0	Lt. ICA occlusion (s/p stent insertion d/t Lt....	OA-M4 bypass, Lt.	angioplasty and stenting at Lt. V4\nimmediate ...	1	2013-05-21	13.0	...	0	0	0	0	1	0	1	1
60	2010-11-10	60.0	72.0	0	Rt. distal ICA and M1 stenosis	EIAB, Rt. (STA-M4)	2 vessel disease (high perioperative risk grou...	1	2010-11-13	3.0	...	0	0	0	0	0	0	0	1
109	2017-10-19	109.0	58.0	0	Bilateral ICA occlusion	EIAB, Lt. (STA-M4)	Hemodynamic insufficiency (Rt. MCA territory >...	1	2017-10-22	3.0	...	1	0	0	0	0	0	1	1

5 rows × 279 columns

In [ ]:
```
5/109*100
```

Out[ ]:
```
4.587155963302752
```

In [ ]:

In [ ]:
```
# Ipsilateral stroke between postoperative 30 days to 2 years
df[df.ipsilateral_postop_stroke==1][(diff_days(df[df.ipsilateral_postop_stroke==1]['op_date'], df[df.ipsilateral_postop_stroke==1]['ipsilateral_postop_stroke_date'])>30)&(diff_days(df[df.ipsilateral_postop_stroke==1]['op_date'], df[df.ipsilateral_postop_stroke==1]['ipsilateral_postop_stroke_date'])<
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	infarct_internal_BDZ	infarct_cortical_internal_BDZ	infarct_large_deep	infarct_small_deep	infarct_old	infarct_no_infarct	collateral_Willisian	collateral_leptomeningeal	collateral_b
17	2015-01-12	17.0	62.0	0	Lt. ICA occlusion	EIAB, Lt (STA-M4 bypass)	postoperative 5 months focal infarction at Lt....	1	2015-06-11	150.0	...	0	1	0	0	0	0	1	1	
101	2007-08-06	101.0	60.0	0	Lt. distal ICA occlusion and Rt. distal ICA oc...	EIAB, Lt. (STA-M4)	postoperative 3 months asymptomatic ipsilatera...	1	2008-01-03	150.0	...	0	1	0	0	0	0	1	1	

2 rows × 279 columns

In [ ]:
```
2/109*100
```

Out[ ]:
```
1.834862385321101
```

In [ ]:

In [ ]:
```
# Ipsilateral stroke between postoperative 30 days to 5 years
df[df.ipsilateral_postop_stroke==1][(diff_days(df[df.ipsilateral_postop_stroke==1]['op_date'], df[df.ipsilateral_postop_stroke==1]['ipsilateral_postop_stroke_date'])>30)&(diff_days(df[df.ipsilateral_postop_stroke==1]['op_date'], df[df.ipsilateral_postop_stroke==1]['ipsilateral_postop_stroke_date'])<
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	infarct_internal_BDZ	infarct_cortical_internal_BDZ	infarct_large_deep	infarct_small_deep	infarct_old	infarct_no_infarct	collateral_Willisian	collateral_leptomeningeal	collateral_b
17	2015-01-12	17.0	62.0	0	Lt. ICA occlusion	EIAB, Lt (STA-M4 bypass)	postoperative 5 months focal infarction at Lt....	1	2015-06-11	150.0	...	0	1	0	0	0	0	1	1	
37	2013-06-19	37.0	56.0	0	ICA stenosis, bilateral (s/p STA-M2 bypass, Rt...	STA-M4 bypass, Lt.	postoperative 3 years ipsilateral MCA territor...	1	2017-10-25	1589.0	...	1	0	0	0	0	0	1	1	
101	2007-08-06	101.0	60.0	0	Lt. distal ICA occlusion and Rt. distal ICA oc...	EIAB, Lt. (STA-M4)	postoperative 3 months asymptomatic ipsilatera...	1	2008-01-03	150.0	...	0	1	0	0	0	0	1	1	

3 rows × 279 columns

In [ ]:
```
3/109*100
```

Out[ ]:
```
2.7522935779816518
```

In [ ]:
```
# Postoperative ipsilateral stroke
df.ipsilateral_postop_stroke.sum()
```

Out[ ]:
```
8
```

In [ ]:
```
df.ipsilateral_postop_stroke.sum() / df.shape[0] * 100
```

Out[ ]:
```
7.339449541284404
```

In [ ]:

In [ ]:
```
# Postoperative ipsilateral symptomatic stroke
df.ipsilateral_symptomatic_postop_stroke.sum()
```

Out[ ]:
```
7
```

In [ ]:
```
df.ipsilateral_symptomatic_postop_stroke.sum() / df.shape[0] * 100
```

Out[ ]:
```
6.422018348623854
```

In [ ]:

In [ ]:
```
# Postoperative ipsilateral asymptomatic stroke
df.ipsilateral_asymptomatic_postop_stroke.sum()
```

Out[ ]:
```
1
```

In [ ]:
```
df.ipsilateral_asymptomatic_postop_stroke.sum() / df.shape[0] * 100
```

Out[ ]:
```
0.9174311926605505
```

In [ ]:
```
7/109*100, 3/109*100, 15/109*100, 14/109*100
```

Out[ ]:
```
(6.422018348623854, 2.7522935779816518, 13.761467889908257, 12.844036697247708)
```

In [ ]:

In [ ]:
```
# Postoperative any stroke
df.any_postop_stroke.sum()
```

Out[ ]:
```
15
```

In [ ]:
```
df.any_postop_stroke.sum() / df.shape[0] * 100
```

Out[ ]:
```
13.761467889908257
```

In [ ]:
```
df[df.any_postop_stroke==1]['remarks']
```

Out[ ]:
```
2      postoperative 2 months expired due to extensiv...
17     postoperative 5 months focal infarction at Lt....
22     immediate postoperative bypass failure lead to...
28     postoperative 1 day ipsilateral MCA territory ...
36     postoperative 3 years contralateral MCA territ...
37     postoperative 3 years ipsilateral MCA territor...
39     angioplasty and stenting at Lt. V4\nimmediate ...
46     postoperative 1.5 years ipsilateral ACA territ...
53     postoperative 5 months contralateral multifoca...
60     2 vessel disease (high perioperative risk grou...
64     postoperative 10 years contralateral focal MCA...
70     postoperative 2 days contralateral PCA-MCA ter...
101    postoperative 3 months asymptomatic ipsilatera...
108    postoperative 3 days contralateral BDZ infarct...
109    Hemodynamic insufficiency (Rt. MCA territory >...
Name: remarks, dtype: object
```

In [ ]:

In [ ]:
```
# Postoperative any symptomatic stroke
df.any_symptomatic_postop_stroke.sum()
```

Out[ ]:
```
14
```

In [ ]:
```
df.any_symptomatic_postop_stroke.sum()/ df.shape[0] * 100
```

Out[ ]:
```
12.844036697247708
```

In [ ]:

In [ ]:
```
# Postoperative any asymptomatic stroke
df.any_asymptomatic_postop_stroke.sum()
```

Out[ ]:
```
1
```

In [ ]:
```
df.any_asymptomatic_postop_stroke.sum() / df.shape[0] * 100
```

Out[ ]:
```
0.9174311926605505
```

In [ ]:

In [ ]:
```
# Postoperative any complication
df.any_postop_complication.sum()
```

Out[ ]:
```
16
```

In [ ]:
```
df[df.any_postop_complication==1]
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	infarct_internal_BDZ	infarct_cortical_internal_BDZ	infarct_large_deep	infarct_small_deep	infarct_old	infarct_no_infarct	collateral_Willisian	collateral_leptomeningeal	collateral_
2	2018-08-30	2.0	49.0	1	Lt. M1 and Rt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 2 months expired due to extensiv...	0	NaT	NaN	...	1	0	0	0	0	0	1	1	
17	2015-01-12	17.0	62.0	0	Lt. ICA occlusion	EIAB, Lt (STA-M4 bypass)	postoperative 5 months focal infarction at Lt....	1	2015-06-11	150.0	...	0	1	0	0	0	0	1	1	
22	2015-11-17	22.0	63.0	1	Bilateral M1 and ICA stenosis	EIAB, Lt. (double barrel STA-M4 bypass)	immediate postoperative bypass failure lead to...	1	2015-11-17	0.0	...	0	1	0	0	0	0	1	1	
23	2015-12-29	23.0	56.0	0	Lt. ICA occlusion	EIAB, Lt. (STA-M4 bypass)	transient postoperative hyperperfusion syndrom...	0	NaT	NaN	...	0	0	0	0	0	0	1	1	
28	2014-05-17	28.0	74.0	0	Lt. ICA occlusion	STA-M3 bypass, Lt.	postoperative 1 day ipsilateral MCA territory ...	1	2014-05-18	1.0	...	1	0	0	0	0	0	1	1	
39	2013-05-08	39.0	49.0	0	Lt. ICA occlusion (s/p stent insertion d/t Lt....	OA-M4 bypass, Lt.	angioplasty and stenting at Lt. V4\nimmediate ...	1	2013-05-21	13.0	...	0	0	0	0	1	0	1	1	
44	2011-02-19	44.0	58.0	0	Lt. M1 occlusion	STA-MCA bypass, Lt.	delayed wound healing	0	NaT	NaN	...	0	1	0	0	0	0	1	1	
60	2010-11-10	60.0	72.0	0	Rt. distal ICA and M1 stenosis	EIAB, Rt. (STA-M4)	2 vessel disease (high perioperative risk grou...	1	2010-11-13	3.0	...	0	0	0	0	0	0	0	1	
62	2010-11-19	62.0	40.0	0	Lt. M1 stenosis	EIAB, Lt. (STA-M4)	postoperative 1 month ipsilateral subacute sub...	0	NaT	NaN	...	1	0	0	0	0	0	1	1	
70	2009-08-17	70.0	45.0	1	Lt. M1 and A1 stenosis and Rt. ICA stenosis	EIAB, Lt (STA-M4)	postoperative 2 days contralateral PCA-MCA ter...	0	NaT	NaN	...	0	0	0	0	1	0	1	1	
74	2009-12-09	74.0	67.0	0	Rt. ICA occlusion	EIAB, Rt. (STA-M4)	postoperative 1 month ipsilateral subacute sub...	0	NaT	NaN	...	0	0	0	0	1	0	1	1	

75	2009-12-23	75.0	66.0	0	Lt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 9 days ipsilateral subacute subd...	0	NaT	NaN	...	0	0	0	0	0	1	1	1
78	2008-02-27	78.0	28.0	0	Lt. M1 stenosis	EIAB, Lt. (STA-M4)	delayed wound healing	0	NaT	NaN	...	0	0	0	0	1	0	1	1
101	2007-08-06	101.0	60.0	0	Lt. distal ICA occlusion and Rt. distal ICA oc...	EIAB, Lt. (STA-M4)	postoperative 3 months asymptomatic ipsilatera...	1	2008-01-03	150.0	...	0	1	0	0	0	0	1	1
108	2010-04-05	108.0	67.0	0	Lt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 3 days contralateral BDZ infarct...	0	NaT	NaN	...	0	1	0	0	0	0	1	1
109	2017-10-19	109.0	58.0	0	Bilateral ICA occlusion	EIAB, Lt. (STA-M4)	Hemodynamic insufficiency (Rt. MCA territory >...	1	2017-10-22	3.0	...	1	0	0	0	0	0	1	1

16 rows × 279 columns

In [ ]:
```
df.any_postop_complication.sum() / df.shape[0] * 100
```

Out[ ]:
```
14.678899082568808
```

In [ ]:

In [ ]:
```
# Postoperative any symptomatic complication
df.any_symptomatic_postop_complication.sum() / df.shape[0] * 100
```

Out[ ]:
```
11.009174311926607
```

In [ ]:

In [ ]:
```
# Postoperative any asymptomatic complication
df.any_asymptomatic_postop_complication.sum() / df.shape[0] * 100
```

Out[ ]:
```
2.7522935779816518
```

In [ ]:

In [ ]:
```
# Postoperative 30 days any complication
df.any_postop_30_days_complication.sum() / df.shape[0] * 100
```

Out[ ]:
```
11.926605504587156
```

In [ ]:

In [ ]:
```
# Postoperative 30 days any symptomatic complication
df.any_symptomatic_postop_30_days_complication.sum() / df.shape[0] * 100
```

Out[ ]:
```
9.174311926605505
```

In [ ]:

In [ ]:
```
# Postoperative 30 days any asymptomatic complication
df.any_asymptomatic_postop_30_days_complication.sum() / df.shape[0] * 100
```

Out[ ]:
```
2.7522935779816518
```

In [ ]:

In [ ]:
```
# Postoperative 1 year any complication
df.any_postop_1_year_complication.sum() / df.shape[0] * 100
```

Out[ ]:
```
13.761467889908257
```

In [ ]:

In [ ]:
```
# Postoperative 1 year any symptomatic complication
df.any_symptomatic_postop_1_year_complication.sum() / df.shape[0] * 100
```

Out[ ]:
```
11.009174311926607
```

In [ ]:

In [ ]:
```
# Postoperative 1 year any asymptomatic complication
df.any_asymptomatic_postop_1_year_complication.sum() / df.shape[0] * 100
```

Out[ ]:
```
2.7522935779816518
```

In [ ]:

In [ ]:
```
df[df.any_postop_stroke==1]
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	infarct_internal_BDZ	infarct_cortical_internal_BDZ	infarct_large_deep	infarct_small_deep	infarct_old	infarct_no_infarct	collateral_Willisian	collateral_leptomeningeal	collateral_
2	2018-08-30	2.0	49.0	1	Lt. M1 and Rt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 2 months expired due to extensiv...	0	NaT	NaN	...	1	0	0	0	0	0	1	1	
17	2015-01-12	17.0	62.0	0	Lt. ICA occlusion	EIAB, Lt (STA-M4 bypass)	postoperative 5 months focal infarction at Lt....	1	2015-06-11	150.0	...	0	1	0	0	0	0	1	1	
22	2015-11-17	22.0	63.0	1	Bilateral M1 and ICA stenosis	EIAB, Lt. (double barrel STA-M4 bypass)	immediate postoperative bypass failure lead to...	1	2015-11-17	0.0	...	0	1	0	0	0	0	1	1	
28	2014-05-17	28.0	74.0	0	Lt. ICA occlusion	STA-M3 bypass, Lt.	postoperative 1 day ipsilateral MCA territory ...	1	2014-05-18	1.0	...	1	0	0	0	0	0	1	1	
36	2013-03-15	36.0	55.0	0	ICA occlusion, bilateral	STA-M4 bypass, Rt.	postoperative 3 years contralateral MCA territ...	0	NaT	NaN	...	1	0	0	0	0	0	1	1	
37	2013-06-19	37.0	56.0	0	ICA stenosis, bilateral (s/p STA-M2 bypass, Rt...	STA-M4 bypass, Lt.	postoperative 3 years ipsilateral MCA territor...	1	2017-10-25	1589.0	...	1	0	0	0	0	0	1	1	
39	2013-05-08	39.0	49.0	0	Lt. ICA occlusion (s/p stent insertion d/t Lt....	OA-M4 bypass, Lt.	angioplasty and stenting at Lt. V4\nimmediate ...	1	2013-05-21	13.0	...	0	0	0	0	1	0	1	1	
46	2011-03-02	46.0	56.0	0	Rt. ICA stenosis	STA-MCA bypass, Rt.	postoperative 1.5 years ipsilateral ACA territ...	0	NaT	NaN	...	0	1	0	0	0	0	1	1	
53	2011-03-28	53.0	68.0	0	Lt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 5 months contralateral multifoca...	0	NaT	NaN	...	0	0	0	0	1	0	1	1	
60	2010-11-10	60.0	72.0	0	Rt. distal ICA and M1 stenosis	EIAB, Rt. (STA-M4)	2 vessel disease (high perioperative risk grou...	1	2010-11-13	3.0	...	0	0	0	0	0	0	0	1	
64	2010-12-11	64.0	68.0	1	Lt. M1 stenosis	EIAB, Lt. (STA-M4)	postoperative 10 years contralateral focal MCA...	0	NaT	NaN	...	0	1	0	0	0	0	1	1	
70	2009-08-17	70.0	45.0	1	Lt. M1 and A1 stenosis and Rt. ICA stenosis	EIAB, Lt (STA-M4)	postoperative 2 days contralateral PCA-MCA ter...	0	NaT	NaN	...	0	0	0	0	1	0	1	1	
101	2007-08-06	101.0	60.0	0	Lt. distal ICA occlusion and Rt. distal ICA oc...	EIAB, Lt. (STA-M4)	postoperative 3 months asymptomatic ipsilatera...	1	2008-01-03	150.0	...	0	1	0	0	0	0	1	1	
108	2010-04-05	108.0	67.0	0	Lt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 3 days contralateral BDZ infarct...	0	NaT	NaN	...	0	1	0	0	0	0	1	1	
109	2017-10-19	109.0	58.0	0	Bilateral ICA occlusion	EIAB, Lt. (STA-M4)	Hemodynamic insufficiency (Rt. MCA territory >...	1	2017-10-22	3.0	...	1	0	0	0	0	0	1	1	

15 rows × 279 columns

In [ ]:

In [ ]:

In [ ]:

In [ ]:
```
# df['op_date_to_ipsilateral_postop_stroke_date_days']
# df['op_date_to_ipsilateral_symptomatic_postop_stroke_date_days']
# df['op_date_to_ipsilateral_asymptomatic_postop_stroke_date_days']
# df['op_date_to_any_postop_stroke_date_days']
# df['op_date_to_any_symptomatic_postop_stroke_date_days']
# df['op_date_to_any_asymptomatic_postop_stroke_date_days']
# df['op_date_to_any_postop_complication_date_days']
# df['op_date_to_any_symptomatic_postop_complication_date_days']
# df['op_date_to_any_postop_30_days_complication_date_days']
# df['op_date_to_any_symptomatic_postop_30_days_complication_date_days']
# df['op_date_to_any_asymptomatic_postop_30_days_complication_date_days']
# df['op_date_to_any_postop_1_year_complication_date_days']
# df['op_date_to_any_symptomatic_postop_1_year_complication_date_days']
# df['op_date_to_any_asymptomatic_postop_1_year_complication_date_days']
```

In [ ]:
```
df['op_date_to_ipsilateral_postop_stroke_date_days'].fillna(df['op_date_to_last_fu_date_days'])
```

Out[ ]:
```
1      1857.0
2        55.0
3      1719.0
4      1839.0
5      1829.0
         ...
105    2978.0
106    2077.0
107    2203.0
108    3470.0
109       3.0
Name: op_date_to_ipsilateral_postop_stroke_date_days, Length: 109, dtype: float64
```

In [ ]:
```python
df['op_date_to_last_fu_date_days']
```

Out[ ]:
```
1      1857.0
2        55.0
3      1719.0
4      1839.0
5      1829.0
        ...
105    2978.0
106    2077.0
107    2203.0
108    3470.0
109     531.0
Name: op_date_to_last_fu_date_days, Length: 109, dtype: float64
```

In [ ]:
```python
df['op_date_to_last_fu_date_days']
```

Out[ ]:
```
1      1857.0
2        55.0
3      1719.0
4      1839.0
5      1829.0
        ...
105    2978.0
106    2077.0
107    2203.0
108    3470.0
109     531.0
Name: op_date_to_last_fu_date_days, Length: 109, dtype: float64
```

In [ ]:

In [ ]:
```python
df['op_date_to_last_fu_date_days'].min(), df['op_date_to_last_fu_date_days'].max()
```

Out[ ]:
```
(55.0, 5566.0)
```

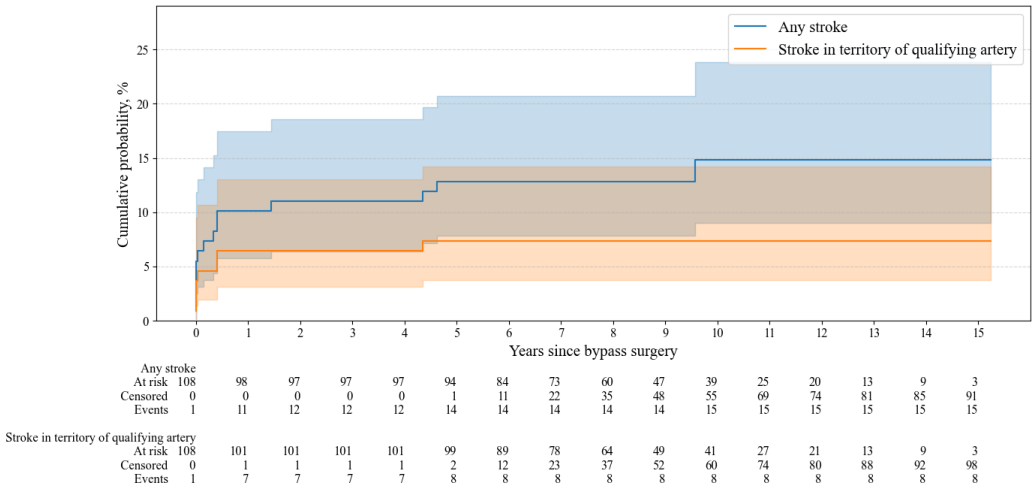
In [ ]:
```python
df['op_date_to_any_postop_stroke_date_days'].fillna(df['op_date_to_last_fu_date_days'])/365
```

Out[ ]:
```
1      5.087671
2      0.150685
3      4.709589
4      5.038356
5      5.010959
         ...
105    8.158904
106    5.690411
107    6.035616
108    0.008219
109    0.008219
Name: op_date_to_any_postop_stroke_date_days, Length: 109, dtype: float64
```

In [ ]:

In [ ]:
```python
# Postoperative any infarction
ax = plt.figure(figsize=(14, 7))

kmf1 = KaplanMeierFitter()
kmf1.fit(durations=df['op_date_to_any_postop_stroke_date_days'].fillna(df['op_date_to_last_fu_date_days']), event_observed=df['any_postop_stroke'], label="Any stroke")
ax = kmf1.plot_cumulative_density()

# Postoperative ipsilateral infarction
kmf2 = KaplanMeierFitter()
kmf2.fit(durations=df['op_date_to_ipsilateral_postop_stroke_date_days'].fillna(df['op_date_to_last_fu_date_days']), event_observed=df['ipsilateral_postop_stroke'], label='Stroke in territory of qualifying artery')
ax = kmf2.plot_cumulative_density()

# # Postoperative any infarction (years)
# kmf3 = KaplanMeierFitter()
# kmf3.fit((df['op_date_to_any_postop_stroke_date_days'].fillna(df['op_date_to_last_fu_date_days'])/365).round(), event_observed=df['any_postop_stroke'], label='Any stroke')


# # Postoperative ipsilateral infarction (years)
# kmf4 = KaplanMeierFitter()
# kmf4.fit((df['op_date_to_ipsilateral_postop_stroke_date_days'].fillna(df['op_date_to_last_fu_date_days'])/365).round(), event_observed=df['ipsilateral_postop_stroke'], label='Stroke in territory of qualifying artery')


ax.set_xticks(365* np.arange(0, 16))
ax.set_xticklabels(np.arange(0, 16), fontsize=12)
ax.set_yticks(np.arange(0, 0.3, 0.05))
ax.set_yticklabels(5 * np.arange(0, 6, 1), fontsize=12)

ax.set_xlabel("Years since bypass surgery", fontsize=15)
ax.set_ylabel("Cumulative probability, %", fontsize=15)
ax.set_ylim(0, 0.29)

ax.grid(True, axis='y', linestyle='--', alpha=0.5)
ax.legend(loc='best')

plt.legend(fontsize=15)

add_at_risk_counts(kmf1, kmf2, ax=ax, fontsize=12)
plt.tight_layout()
# plt.savefig(r"C:\Users\myleg\OneDrive\바탕 화면\Thesis works\Long term outcome of bypass for ICAD\stroke_cumulative_density.pdf")

plt.show()
```

```
c:\Users\myleg\miniconda3\envs\cudapy_310\lib\site-packages\lifelines\utils\__init__.py:1187: UserWarning: Attempting to convert an unexpected datatype 'object' to float. Suggestion: 1) use `lifelines.utils.datetimes_to_durations` to do conversions or 2) manually convert to floats/booleans.
  warnings.warn(warning_text, UserWarning)
c:\Users\myleg\miniconda3\envs\cudapy_310\lib\site-packages\lifelines\utils\__init__.py:1187: UserWarning: Attempting to convert an unexpected datatype 'object' to float. Suggestion: 1) use `lifelines.utils.datetimes_to_durations` to do conversions or 2) manually convert to floats/booleans.
  warnings.warn(warning_text, UserWarning)
```



In [ ]:

In [ ]:

Analysis

In [ ]:
```python
# Follow-up period after bypass surgery
df['op_date_to_last_fu_date_months'].median(), df['op_date_to_last_fu_date_months'].quantile([0.25, 0.75])
```

Out[ ]:
```
(116.0,
 0.25     92.0
 0.75    143.0
 Name: op_date_to_last_fu_date_months, dtype: float64)
```

In [ ]:

In [ ]:
```python
df.lesion_stenosis_occlusion.value_counts()
```

Out[ ]:
```
0    94
1    15
Name: lesion_stenosis_occlusion, dtype: int64
```

In [ ]:
```python
df[df.lesion_stenosis_occlusion==1]
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_days	...	infarct_internal_BDZ	infarct_cortical_internal_BDZ	infarct_large_deep	infarct_small_deep	infarct_old	infarct_no_infarct	collateral_Willisian	collateral_leptomeningeal	collateral_l
3	2018-12-24	3.0	59.0	0	Rt. ICA stenosis	EIAB, Rt. (STA-M4) and EDAS, Rt.	restenosis of previously stented lesion (6 mon...	0	NaT	NaN	...	0	1	0	0	0	0	0	1	
4	2018-10-31	4.0	59.0	0	Lt. M1 occlusion	EIAB, Lt. (STA parietal-M4)	NaN	0	NaT	NaN	...	0	1	0	0	0	0	0	1	
5	2018-08-29	5.0	59.0	0	Rt. ICA and M1 stenosis	EIAB, Rt. (STA-M4)	aborted angioplasty at main lesion	0	NaT	NaN	...	0	1	0	0	0	0	0	1	
11	2017-03-13	11.0	56.0	1	Lt. ICA stenosis (s/p stent insertion d/t Lt. ...	EIAB, Lt. (STA-M4)	two times of angioplasty and stenting at main ...	0	NaT	NaN	...	0	1	0	0	0	0	1	1	
13	2017-06-26	13.0	60.0	1	Rt. ICA stenosis	EIAB, Rt. (STA-M4)	NaN	0	NaT	NaN	...	0	0	0	0	0	1	1	1	
22	2015-11-17	22.0	63.0	1	Bilateral M1 and ICA stenosis	EIAB, Lt. (double barrel STA-M4 bypass)	immediate postoperative bypass failure lead to...	1	2015-11-17	0.0	...	0	1	0	0	0	0	1	1	
35	2013-02-25	35.0	57.0	0	Rt. ICA stenosis (s/p balloon angioplasty d/t ...	STA-MCA(M4) bypass, Rt.	NaN	0	NaT	NaN	...	0	0	0	0	1	0	1	1	
43	2012-08-14	43.0	58.0	1	Lt. M1 stenosis (s/p Stent insertion d/t Lt. M...	EIAB, Lt. (STA-M4) (double barrel)	twice angioplasty and stenting at main lesion	0	NaT	NaN	...	0	1	0	0	0	0	1	1	

46	2011-03-02	46.0	56.0	0	Rt. ICA stenosis	STA-MCA bypass, Rt.	postoperative 1.5 years ipsilateral ACA territ...	0	NaT	NaN	...	0	1	0	0	0	0	1	1
57	2010-09-24	57.0	63.0	0	Rt. ICA occlusion	EIAB, Rt. (STA-M4)	NaN	0	NaT	NaN	...	0	1	0	0	0	0	1	1
70	2009-08-17	70.0	45.0	1	Lt. M1 and A1 stenosis and Rt. ICA stenosis	EIAB, Lt (STA-M4)	postoperative 2 days contralateral PCA-MCA ter...	0	NaT	NaN	...	0	0	0	0	1	0	1	1
95	2007-01-22	95.0	68.0	1	Lt. distal ICA stenosis	EIAB, Lt. (STA-M4)	NaN	0	NaT	NaN	...	1	0	0	0	0	0	1	1
100	2007-07-03	100.0	51.0	0	Lt. M2 stenosis, Rt. M1 stenosis, Rt. distal I...	EIAB, Lt. (STA-M4)	NaN	0	NaT	NaN	...	1	0	0	0	0	0	0	1
103	2007-08-29	103.0	58.0	0	Lt. M1 stenosis, Lt. distal ICA stenosis, Rt. ...	EIAB, Lt. (STA-M4)	NaN	0	NaT	NaN	...	1	0	0	0	0	0	0	1
105	2007-10-15	105.0	56.0	0	Rt. distal ICA stenosis and Lt. distal ICA ste...	EIAB, Rt. (STA-M4)	NaN	0	NaT	NaN	...	0	0	0	0	0	0	0	1

15 rows × 279 columns

In [ ]:
```
df[(df.lesion_stenosis_occlusion==1)&(df.any_postop_stroke==1)]
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	infarct_internal_BDZ	infarct_cortical_internal_BDZ	infarct_large_deep	infarct_small_deep	infarct_old	infarct_no_infarct	collateral_Willisian	collateral_leptomeningeal	collateral_bas
22	2015-11-17	22.0	63.0	1	Bilateral M1 and ICA stenosis	EIAB, Lt. (double barrel STA-M4 bypass)	immediate postoperative bypass failure lead to...	1	2015-11-17	0.0	...	0	1	0	0	0	0	1	1	
46	2011-03-02	46.0	56.0	0	Rt. ICA stenosis	STA-MCA bypass, Rt.	postoperative 1.5 years ipsilateral ACA territ...	0	NaT	NaN	...	0	1	0	0	0	0	1	1	
70	2009-08-17	70.0	45.0	1	Lt. M1 and A1 stenosis and Rt. ICA stenosis	EIAB, Lt (STA-M4)	postoperative 2 days contralateral PCA-MCA ter...	0	NaT	NaN	...	0	0	0	0	1	0	1	1	

3 rows × 279 columns

In [ ]:
```
pd.crosstab(df.lesion_stenosis_occlusion, df.any_postop_stroke)
```

Out[ ]:

any_postop_stroke	0	1
lesion_stenosis_occlusion		
0	82	12
1	12	3

In [ ]:
```
f, p = ss.fisher_exact(pd.crosstab(df.lesion_stenosis_occlusion, df.any_postop_stroke))
```

In [ ]:
```
print(f, p)
```

```
1.7083333333333333 0.4310822658836879
```

In [ ]:

In [ ]:

In [ ]:
```
df[df.any_postop_1_year_complication==1]['remarks']
```

Out[ ]:
```
2      postoperative 2 months expired due to extensiv...
17     postoperative 5 months focal infarction at Lt....
22     immediate postoperative bypass failure lead to...
23     transient postoperative hyperperfusion syndrom...
28     postoperative 1 day ipsilateral MCA territory ...
39     angioplasty and stenting at Lt. V4\nimmediate ...
44                                 delayed wound healing
60     2 vessel disease (high perioperative risk grou...
62     postoperative 1 month ipsilateral subacute sub...
70     postoperative 2 days contralateral PCA-MCA ter...
74     postoperative 1 month ipsilateral subacute sub...
75     postoperative 9 days ipsilateral subacute subd...
78                                 delayed wound healing
108    postoperative 3 days contralateral BDZ infarct...
109    Hemodynamic insufficiency (Rt. MCA territory >...
Name: remarks, dtype: object
```

In [ ]:
```
df[df.any_symptomatic_postop_1_year_complication==1]['remarks']
```

Out[ ]:
```
2      postoperative 2 months expired due to extensiv...
17     postoperative 5 months focal infarction at Lt....
22     immediate postoperative bypass failure lead to...
23     transient postoperative hyperperfusion syndrom...
28     postoperative 1 day ipsilateral MCA territory ...
39     angioplasty and stenting at Lt. V4\nimmediate ...
60     2 vessel disease (high perioperative risk grou...
62     postoperative 1 month ipsilateral subacute sub...
70     postoperative 2 days contralateral PCA-MCA ter...
74     postoperative 1 month ipsilateral subacute sub...
75     postoperative 9 days ipsilateral subacute subd...
108    postoperative 3 days contralateral BDZ infarct...
109    Hemodynamic insufficiency (Rt. MCA territory >...
Name: remarks, dtype: object
```

In [ ]:
```
df[df.any_postop_30_days_complication==1]['remarks']
```

Out[ ]:
```
22     immediate postoperative bypass failure lead to...
23     transient postoperative hyperperfusion syndrom...
28     postoperative 1 day ipsilateral MCA territory ...
39     angioplasty and stenting at Lt. V4\nimmediate ...
44                                 delayed wound healing
60     2 vessel disease (high perioperative risk grou...
62     postoperative 1 month ipsilateral subacute sub...
70     postoperative 2 days contralateral PCA-MCA ter...
74     postoperative 1 month ipsilateral subacute sub...
75     postoperative 9 days ipsilateral subacute subd...
78                                 delayed wound healing
108    postoperative 3 days contralateral BDZ infarct...
109    Hemodynamic insufficiency (Rt. MCA territory >...
Name: remarks, dtype: object
```

In [ ]:

In [ ]:
```
# Cases with decrease in mRS score on last F/U after bypass surgery
(df['last_mRS'] - df['preop_mRS'] < 0).sum()
```

Out[ ]: 72

In [ ]:
```
# Cases with no change mRS score on last F/U after bypass surgery
(df['last_mRS'] - df['preop_mRS'] == 0).sum()
```

Out[ ]: 28

In [ ]:
```
# Cases with increase in mRS score on last F/U after bypass surgery
(df['last_mRS'] - df['preop_mRS'] > 0).sum()
```

Out[ ]: 9

In [ ]:

Poor outcome group analysis

In [ ]:
```
# Defining poor outcome group (Increase in mRS score on last F/U compare to preoperative mRS score, any postoperative stroke, any symptomatic complication within 1 year)
df['poor_outcome'] = 0
df.loc[(df['last_mRS'] - df['preop_mRS'] > 0)|(df['any_postop_stroke']==1)|(df['any_symptomatic_postop_1_year_complication']==1), 'poor_outcome'] = 1

# Exclude cases with transient problem from poor outcome group
df.loc[[23, 62, 75], 'poor_outcome'] = 0
```

In [ ]:
```
df[df['poor_outcome']==1]
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	infarct_cortical_internal_BDZ	infarct_large_deep	infarct_small_deep	infarct_old	infarct_no_infarct	collateral_Willisian	collateral_leptomeningeal	collateral_basal	collateral_trans
2	2018-08-30	2.0	49.0	1	Lt. M1 and Rt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 2 months expired due to extensiv...	0	NaT	NaN	...	0	0	0	0	0	1	1	0	
17	2015-01-12	17.0	62.0	0	Lt. ICA occlusion	EIAB, Lt (STA-M4 bypass)	postoperative 5 months focal infarction at Lt....	1	2015-06-11	150.0	...	1	0	0	0	0	1	1	0	
22	2015-11-17	22.0	63.0	1	Bilateral M1 and ICA stenosis	EIAB, Lt. (double barrel STA-M4 bypass)	immediate postoperative bypass failure lead to...	1	2015-11-17	0.0	...	1	0	0	0	0	1	1	0	
28	2014-05-17	28.0	74.0	0	Lt. ICA occlusion	STA-M3 bypass, Lt.	postoperative 1 day ipsilateral MCA territory ...	1	2014-05-18	1.0	...	0	0	0	0	0	1	1	1	
36	2013-03-15	36.0	55.0	0	ICA occlusion, bilateral	STA-M4 bypass, Rt.	postoperative 3 years contralateral MCA territ...	0	NaT	NaN	...	0	0	0	0	0	1	1	0	
37	2013-06-19	37.0	56.0	0	ICA stenosis, bilateral (s/p STA-M2 bypass, Rt...	STA-M4 bypass, Lt.	postoperative 3 years ipsilateral MCA territor...	1	2017-10-25	1589.0	...	0	0	0	0	0	1	1	0	
39	2013-05-08	39.0	49.0	0	Lt. ICA occlusion (s/p stent insertion d/t Lt....	OA-M4 bypass, Lt.	angioplasty and stenting at Lt. V4\nimmediate ...	1	2013-05-21	13.0	...	0	0	0	1	0	1	1	0	
46	2011-03-02	46.0	56.0	0	Rt. ICA stenosis	STA-MCA bypass, Rt.	postoperative 1.5 years ipsilateral ACA territ...	0	NaT	NaN	...	1	0	0	0	0	1	1	0	

53	2011-03-28	53.0	68.0	0	Lt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 5 months contralateral multifoca...	0	NaT	NaN	...	0	0	0	1	0	1	1	0
60	2010-11-10	60.0	72.0	0	Rt. distal ICA and M1 stenosis	EIAB, Rt. (STA-M4)	2 vessel disease (high perioperative risk grou...	1	2010-11-13	3.0	...	0	0	0	0	0	0	1	0
64	2010-12-11	64.0	68.0	1	Lt. M1 stenosis	EIAB, Lt. (STA-M4)	postoperative 10 years contralateral focal MCA...	0	NaT	NaN	...	1	0	0	0	0	1	1	0
70	2009-08-17	70.0	45.0	1	Lt. M1 and A1 stenosis and Rt. ICA stenosis	EIAB, Lt (STA-M4)	postoperative 2 days contralateral PCA-MCA ter...	0	NaT	NaN	...	0	0	0	1	0	1	1	0
101	2007-08-06	101.0	60.0	0	Lt. distal ICA occlusion and Rt. distal ICA oc...	EIAB, Lt. (STA-M4)	postoperative 3 months asymptomatic ipsilatera...	1	2008-01-03	150.0	...	1	0	0	0	0	1	1	0
108	2010-04-05	108.0	67.0	0	Lt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 3 days contralateral BDZ infarct...	0	NaT	NaN	...	1	0	0	0	0	1	1	0
109	2017-10-19	109.0	58.0	0	Bilateral ICA occlusion	EIAB, Lt. (STA-M4)	Hemodynamic insufficiency (Rt. MCA territory >...	1	2017-10-22	3.0	...	0	0	0	0	0	1	1	0

15 rows × 280 columns

In [ ]:
```
df[df['any_asymptomatic_postop_1_year_complication']==1]
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	infarct_cortical_internal_BDZ	infarct_large_deep	infarct_small_deep	infarct_old	infarct_no_infarct	collateral_Willisian	collateral_leptomeningeal	collateral_basal	collateral_transdur
44	2011-02-19	44.0	58.0	0	Lt. M1 occlusion	STA-MCA bypass, Lt.	delayed wound healing	0	NaT	NaN	...	1	0	0	0	0	1	1	0	
74	2009-12-09	74.0	67.0	0	Rt. ICA occlusion	EIAB, Rt. (STA-M4)	postoperative 1 month ipsilateral subacute sub...	0	NaT	NaN	...	0	0	0	1	0	1	1	0	
78	2008-02-27	78.0	28.0	0	Lt. M1 stenosis	EIAB, Lt. (STA-M4)	delayed wound healing	0	NaT	NaN	...	0	0	0	1	0	1	1	1	

3 rows × 280 columns

In [ ]:

In [ ]:
```
df[df.poor_outcome==1][['sex', 'age']]
```

Out[ ]:

	sex	age
2	1	49.0
17	0	62.0
22	1	63.0
28	0	74.0
36	0	55.0
37	0	56.0
39	0	49.0
46	0	56.0
53	0	68.0
60	0	72.0
64	1	68.0
70	1	45.0
101	0	60.0
108	0	67.0
109	0	58.0

In [ ]:
```
df[df.poor_outcome==1][['preop_mRS', 'last_mRS']]
```

Out[ ]:

	preop_mRS	last_mRS
2	4.0	6.0
17	3.0	2.0
22	3.0	3.0
28	3.0	6.0
36	2.0	4.0
37	2.0	4.0
39	3.0	2.0
46	3.0	4.0
53	2.0	3.0
60	3.0	4.0
64	4.0	3.0
70	1.0	3.0
101	3.0	2.0
108	4.0	3.0
109	2.0	5.0

In [ ]:
```
df[df.poor_outcome==1]['first_event_to_treatment_days']
```

Out[ ]:
```
2     178.0
17     78.0
22     46.0
28    105.0
36    186.0
37    282.0
39    371.0
46    119.0
53    146.0
60    253.0
64      6.0
70    319.0
101    64.0
108    81.0
109   106.0
Name: first_event_to_treatment_days, dtype: float64
```

In [ ]:
```
df[df.poor_outcome==1]['last_event_to_treatment_days']
```

Out[ ]:
```
2       9.0
17     37.0
22     13.0
28      6.0
36     26.0
37    122.0
39     73.0
46     68.0
53    111.0
60      1.0
64      1.0
70     30.0
101     1.0
108    74.0
109    63.0
Name: last_event_to_treatment_days, dtype: float64
```

In [ ]:
```
df[df.poor_outcome==1]['op_date_to_any_postop_stroke_days']
```

Out[ ]:
```
2     NaN
17    NaN
22    NaN
28    NaN
36    NaN
37    NaN
39    NaN
46    NaN
53    NaN
60    NaN
64    NaN
70    NaN
101   NaN
108   NaN
109   NaN
Name: op_date_to_any_postop_stroke_days, dtype: float64
```

In [ ]:
```
df[df.poor_outcome==1]['op_date_to_last_fu_date_days']
```

Out[ ]:
```
2       55.0
17    3112.0
22     377.0
28    1521.0
36    2789.0
37    2693.0
39    3437.0
46    4418.0
53    2968.0
60    3697.0
64    4169.0
70    3252.0
101   3256.0
108   3470.0
109    531.0
Name: op_date_to_last_fu_date_days, dtype: float64
```

In [ ]:
```
df['op_date_to_any_postop_stroke_days'] = diff_days(df['op_date'], df['any_postop_stroke_date'])
```

In [ ]:
```
df[df.poor_outcome==1]['op_date_to_any_postop_stroke_days']
```

Out[ ]:
```
2        55.0
17      150.0
22        0.0
28        1.0
36     1685.0
37     1589.0
39       13.0
46      528.0
53      120.0
60        3.0
64     3492.0
70        2.0
101     150.0
108       3.0
109       3.0
Name: op_date_to_any_postop_stroke_days, dtype: float64
```

In [ ]:
```
df[df.poor_outcome==1]['op_date_to_last_fu_date_days']
```

Out[ ]:
```
2        55.0
17     3112.0
22      377.0
28     1521.0
36     2789.0
37     2693.0
39     3437.0
46     4418.0
53     2968.0
60     3697.0
64     4169.0
70     3252.0
101    3256.0
108    3470.0
109     531.0
Name: op_date_to_last_fu_date_days, dtype: float64
```

In [ ]:
```
df[df.poor_outcome==1]['main_lesion_site']
```

Out[ ]:
```
2       Lt. M1o
17     Lt. pICA
22     Lt. dICA
28     Lt. pICA
36     Rt. pICA
37     Lt. pICA
39     Lt. pICA
46     Rt. dICA
53     Lt. pICA
60      Rt. M1t
64      Lt. M1t
70      Lt. M1o
101    Lt. dICA
108    Lt. dICA
109    Rt. pICA
Name: main_lesion_site, dtype: object
```

In [ ]:
```
# 0: aymptomatic, 1: TIA, 2: infarction, 3: hemorrhage, 4: other symptoms
df[df.poor_outcome==1]['presentation']
```

Out[ ]:
```
2      2
17     2
22     2
28     2
36     2
37     2
39     1
46     2
53     1
60     2
64     2
70     1
101    2
108    2
109    2
Name: presentation, dtype: object
```

In [ ]:
```
df[df.poor_outcome==1]['diffusion_infarction_pattern']
```

Out[ ]:
```
2      1,4
17       5
22       5
28       4
36       4
37       4
39       8
46       5
53       8
60       3
64       5
70       8
101      5
108      5
109      4
Name: diffusion_infarction_pattern, dtype: object
```

In [ ]:
```
df[df.poor_outcome==1]['lesion_stenosis_occlusion']
```

Out[ ]:
```
2      0
17     0
22     1
28     0
36     0
37     0
39     0
46     1
53     0
60     0
64     0
70     1
101    0
108    0
109    0
Name: lesion_stenosis_occlusion, dtype: object
```

In [ ]:
```
df[df.poor_outcome==1]['last_bypass_patency']
```

Out[ ]:
```
2      2
17     1
22     4
28     2
36     2
37     1
39     1
46     2
53     2
60     0
64     2
70     2
101    2
108    2
109    2
Name: last_bypass_patency, dtype: object
```

In [ ]:

Bypass patency analysis

In [ ]:
```
# Bypass types (0: single-barrel, 1: double_barrel, 2: interposition (high flow), 3: interposition (low flow), 4: direct + indirect)
df['bypass_type'].value_counts().sort_index()
```

Out[ ]:
```
0    97
1    10
4     2
Name: bypass_type, dtype: int64
```

In [ ]:
```
# Last bypass patency (0: failure, 1: poor, 2: good, 3: good/good, 4: good/poor, 5: poor/poor)
df['last_bypass_patency'].value_counts().sort_index()
```

Out[ ]:
```
0     2
1     8
2    87
3    10
4     2
Name: last_bypass_patency, dtype: int64
```

In [ ]:
```
df['last_bypass_patency'].value_counts().sort_index()/109*100
```

Out[ ]:
```
0     1.834862
1     7.339450
2    79.816514
3     9.174312
4     1.834862
Name: last_bypass_patency, dtype: float64
```

In [ ]:

In [ ]:
```
# Second direct anastomosis of double barrel bypass cases
db2 = df[df['bypass_type']==1][['initial_STAb_diameter_second', '1y_STAb_diameter_second', '3y_STAb_diameter_second', 'last_STAb_diameter_second']]
db2.rename(columns={'initial_STAb_diameter_second':'initial_STAb_diameter', '1y_STAb_diameter_second': '1y_STAb_diameter', '3y_STAb_diameter_second': '3y_STAb_diameter', 'last_STAb_diameter_second':'last_STAb_diameter'}, inplace=True)
db2
```

Out[ ]:

	initial_STAb_diameter	1y_STAb_diameter	3y_STAb_diameter	last_STAb_diameter
22	0.61	NaN	NaN	1.16
40	1.18	2.12	2.28	2.32
43	0.78	1.3	1.35	1.41
49	0.93	1.89	1.9	1.94
52	1.03	1.05	1.52	1.37
83	1.12	2.31	2.33	2.33
84	1.12	2.02	2.04	2.29
98	0.95	1.25	1.31	1.54
102	1.43	1.7	2.01	2.3
104	1.19	2.85	3.05	3.39

In [ ]:
```
# Every direct anastomosis (Single barrel + Single barrel with indirect + Double barrel)
overall_bypass_graft_diameter = pd.concat([df[(df['bypass_type']==0)|(df['bypass_type']==1)|(df['bypass_type']==4)][['initial_STAb_diameter', '1w_STAb_diameter','1y_STAb_diameter', '3y_STAb_diameter', 'last_STAb_diameter']], db2])
overall_bypass_graft_diameter.reset_index(drop=True, inplace=True)
```

In [ ]:
```
overall_bypass_graft_diameter
```

Out[ ]:

	initial_STAb_diameter	1w_STAb_diameter	1y_STAb_diameter	3y_STAb_diameter	last_STAb_diameter
0	1.1	1.58	2.36	2.4	2.41
1	0.77	1.92	NaN	NaN	1.92
2	1.16	1.89	2.07	2.15	2.15
3	1.06	2.02	2.42	2.48	2.44
4	1.13	1.77	2.5	2.5	2.55
...	...	...	...	...	...
114	1.12	NaN	2.31	2.33	2.33

115	1.12	NaN	2.02	2.04	2.29
116	0.95	NaN	1.25	1.31	1.54
117	1.43	NaN	1.7	2.01	2.3
118	1.19	NaN	2.85	3.05	3.39

119 rows × 5 columns

In [ ]:
```
# Initial bypass patency of single direct anastomosis cases (0: failure, 1: poor, 2: good, 3: good/good, 4: good/poor, 5: poor/poor)
df[(df['bypass_type']==0)|(df['bypass_type']==4)]['initial_bypass_patency'].value_counts().sort_index()
```

Out[ ]:
```
0     1
1     5
2    91
3     2
Name: initial_bypass_patency, dtype: int64
```

In [ ]:
```
# Failure, poor, good
1/99*100, 5/99*100, 93/99*100
```

Out[ ]:
```
(1.0101010101010102, 5.05050505050505, 93.93939393939394)
```

In [ ]:

In [ ]:

In [ ]:
```
# Postoperative 1 year single direct anastomosis cases (0: failure, 1: poor, 2: good, 3: good/good, 4: good/poor, 5: poor/poor)
df[(df['bypass_type']==0)|(df['bypass_type']==4)]['1y_bypass_patency'].value_counts().sort_index().sum()
```

Out[ ]:
```
94
```

In [ ]:
```
df[(df['bypass_type']==0)|(df['bypass_type']==4)]['1y_bypass_patency'].value_counts().sort_index()
```

Out[ ]:
```
0     1
1     7
2    84
3     2
Name: 1y_bypass_patency, dtype: int64
```

In [ ]:
```
1/94*100, 7/94*100, 86/94*100
```

Out[ ]:
```
(1.0638297872340425, 7.446808510638298, 91.48936170212765)
```

In [ ]:
```
df[df['bypass_type']==0][df[df['bypass_type']==0]['1y_bypass_patency'].isnull()==1]
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	infarct_cortical_internal_BDZ	infarct_large_deep	infarct_small_deep	infarct_old	infarct_no_infarct	collateral_Willisian	collateral_leptomeningeal	collateral_basal	collateral_trans
2	2018-08-30	2.0	49.0	1	Lt. M1 and Rt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 2 months expired due to extensiv...	0	NaT	NaN	...	0	0	0	0	0	1	1	0	
28	2014-05-17	28.0	74.0	0	Lt. ICA occlusion	STA-M3 bypass, Lt.	postoperative 1 day ipsilateral MCA territory ...	1	2014-05-18	1.0	...	0	0	0	0	0	1	1	1	
60	2010-11-10	60.0	72.0	0	Rt. distal ICA and M1 stenosis	EIAB, Rt. (STA-M4)	2 vessel disease (high perioperative risk grou...	1	2010-11-13	3.0	...	0	0	0	0	0	0	1	0	
70	2009-08-17	70.0	45.0	1	Lt. M1 and A1 stenosis and Rt. ICA stenosis	EIAB, Lt (STA-M4)	postoperative 2 days contralateral PCA-MCA ter...	0	NaT	NaN	...	0	0	0	1	0	1	1	0	
109	2017-10-19	109.0	58.0	0	Bilateral ICA occlusion	EIAB, Lt. (STA-M4)	Hemodynamic insufficiency (Rt. MCA territory >...	1	2017-10-22	3.0	...	0	0	0	0	0	1	1	0	

5 rows × 280 columns

In [ ]:

In [ ]:
```
# Postoperative 3 year single direct anastomosis cases (0: failure, 1: poor, 2: good, 3: good/good, 4: good/poor, 5: poor/poor)
df[(df['bypass_type']==0)|(df['bypass_type']==4)]['3y_bypass_patency'].value_counts().sort_index().sum()
```

Out[ ]:
```
94
```

In [ ]:
```
df[(df['bypass_type']==0)|(df['bypass_type']==4)]['3y_bypass_patency'].value_counts().sort_index()
```

Out[ ]:
```
0     1
1     8
2    83
3     2
Name: 3y_bypass_patency, dtype: int64
```

In [ ]:
```
1/94*100, 8/94*100, 85/94*100
```

Out[ ]:
```
(1.0638297872340425, 8.51063829787234, 90.42553191489363)
```

In [ ]:
```
df[df['bypass_type']==0][df[df['bypass_type']==0]['3y_bypass_patency'].isnull()==1]
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	infarct_cortical_internal_BDZ	infarct_large_deep	infarct_small_deep	infarct_old	infarct_no_infarct	collateral_Willisian	collateral_leptomeningeal	collateral_basal	collateral_trans
2	2018-08-30	2.0	49.0	1	Lt. M1 and Rt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 2 months expired due to extensiv...	0	NaT	NaN	...	0	0	0	0	0	1	1	0	
28	2014-05-17	28.0	74.0	0	Lt. ICA occlusion	STA-M3 bypass, Lt.	postoperative 1 day ipsilateral MCA territory ...	1	2014-05-18	1.0	...	0	0	0	0	0	1	1	1	
60	2010-11-10	60.0	72.0	0	Rt. distal ICA and M1 stenosis	EIAB, Rt. (STA-M4)	2 vessel disease (high perioperative risk grou...	1	2010-11-13	3.0	...	0	0	0	0	0	0	1	0	
70	2009-08-17	70.0	45.0	1	Lt. M1 and A1 stenosis and Rt. ICA stenosis	EIAB, Lt (STA-M4)	postoperative 2 days contralateral PCA-MCA ter...	0	NaT	NaN	...	0	0	0	1	0	1	1	0	
109	2017-10-19	109.0	58.0	0	Bilateral ICA occlusion	EIAB, Lt. (STA-M4)	Hemodynamic insufficiency (Rt. MCA territory >...	1	2017-10-22	3.0	...	0	0	0	0	0	1	1	0	

5 rows × 280 columns

In [ ]:

In [ ]:
```
# Postoperative 5 year single direct anastomosis cases (0: failure, 1: poor, 2: good, 3: good/good, 4: good/poor, 5: poor/poor)
df[(df['bypass_type']==0)|(df['bypass_type']==4)]['5y_bypass_patency'].value_counts().sort_index().sum()
```

Out[ ]:
```
94
```

In [ ]:
```
df[(df['bypass_type']==0)|(df['bypass_type']==4)]['5y_bypass_patency'].value_counts().sort_index()
```

Out[ ]:
```
0     1
1     8
2    83
3     2
Name: 5y_bypass_patency, dtype: int64
```

In [ ]:
```
1/94*100, 8/94*100, 85/94*100
```

Out[ ]:
```
(1.0638297872340425, 8.51063829787234, 90.42553191489363)
```

In [ ]:
```
df[df['bypass_type']==0][df[df['bypass_type']==0]['5y_bypass_patency'].isnull()==1]
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	infarct_cortical_internal_BDZ	infarct_large_deep	infarct_small_deep	infarct_old	infarct_no_infarct	collateral_Willisian	collateral_leptomeningeal	collateral_basal	collateral_trans
2	2018-08-30	2.0	49.0	1	Lt. M1 and Rt. ICA occlusion	EIAB, Lt. (STA-M4)	postoperative 2 months expired due to extensiv...	0	NaT	NaN	...	0	0	0	0	0	1	1	0	
28	2014-05-17	28.0	74.0	0	Lt. ICA occlusion	STA-M3 bypass, Lt.	postoperative 1 day ipsilateral MCA territory ...	1	2014-05-18	1.0	...	0	0	0	0	0	1	1	1	
60	2010-11-10	60.0	72.0	0	Rt. distal ICA and M1 stenosis	EIAB, Rt. (STA-M4)	2 vessel disease (high perioperative risk grou...	1	2010-11-13	3.0	...	0	0	0	0	0	0	1	0	
70	2009-08-17	70.0	45.0	1	Lt. M1 and A1 stenosis and Rt. ICA stenosis	EIAB, Lt (STA-M4)	postoperative 2 days contralateral PCA-MCA ter...	0	NaT	NaN	...	0	0	0	1	0	1	1	0	
109	2017-10-19	109.0	58.0	0	Bilateral ICA occlusion	EIAB, Lt. (STA-M4)	Hemodynamic insufficiency (Rt. MCA territory >...	1	2017-10-22	3.0	...	0	0	0	0	0	1	1	0	

5 rows × 280 columns

In [ ]:

In [ ]:
```
# Initial bypass patency of double barrel anastomosis cases (0: failure, 1: poor, 2: good, 3: good/good, 4: good/poor, 5: poor/poor)
df[df['bypass_type']==1]['initial_bypass_patency'].value_counts().sort_index()
```

Out[ ]:
```
3    7
4    2
5    1
Name: initial_bypass_patency, dtype: int64
```

In [ ]:
```
df[df['bypass_type']==1][df[df['bypass_type']==1]['initial_bypass_patency'].isnull()==1]
```

Out[ ]:

op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	infarct_cortical_internal_BDZ	infarct_large_deep	infarct_small_deep	infarct_old	infarct_no_infarct	collateral_Willisian	collateral_leptomeningeal	collateral_basal	collateral_transdural	poor_

0 rows × 280 columns

In [ ]:
```
df[df['bypass_type']==1][df[df['bypass_type']==1]['initial_bypass_patency']==4]
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	infarct_cortical_internal_BDZ	infarct_large_deep	infarct_small_deep	infarct_old	infarct_no_infarct	collateral_Willisian	collateral_leptomeningeal	collateral_basal	collateral_transd
22	2015-11-17	22.0	63.0	1	Bilateral M1 and ICA stenosis	EIAB, Lt. (double barrel STA-M4 bypass)	immediate postoperative bypass failure lead to...	1	2015-11-17	0.0	...	1	0	0	0	0	1	1	0	

102	2007-08-22	102.0	57.0	0	Rt. distal ICA occlusion	EIAB, Rt. (STA-M4 double barrel bypass)	NaN	0	NaT	NaN	...	1	0	0	0	0	1	1	0				

2 rows × 280 columns

In [ ]:
```
df[df['bypass_type']==1][df[df['bypass_type']==1]['initial_bypass_patency']==5]
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	infarct_cortical_internal_BDZ	infarct_large_deep	infarct_small_deep	infarct_old	infarct_no_infarct	collateral_Willisian	collateral_leptomeningeal	collateral_basal	collateral_transdural
52	2011-08-04	52.0	62.0	1	Bilateral MCA occlusion and Rt. PCA stenosis	EIAB, Rt. (STA-M4)	NaN	0	NaT	NaN	...	0	0	0	1	0	1	1	1	0

1 rows × 280 columns

In [ ]:

In [ ]:
```
# Postoperative 1 year bypass patency of double barrel anastomosis cases (0: failure, 1: poor, 2: good, 3: good/good, 4: good/poor, 5: poor/poor)
df[df['bypass_type']==1]['1y_bypass_patency'].value_counts().sort_index()
```

Out[ ]:
```
3    8
4    1
Name: 1y_bypass_patency, dtype: int64
```

In [ ]:
```
df[df['bypass_type']==1][df[df['bypass_type']==1]['1y_bypass_patency']==4]
```

Out[ ]:

	op_date	id	age	sex	diagnosis	op_name	remarks	ipsilateral_postop_stroke	ipsilateral_postop_stroke_date	op_date_to_ipsilateral_postop_stroke_date_days	...	infarct_cortical_internal_BDZ	infarct_large_deep	infarct_small_deep	infarct_old	infarct_no_infarct	collateral_Willisian	collateral_leptomeningeal	collateral_basal	collateral_transdural
52	2011-08-04	52.0	62.0	1	Bilateral MCA occlusion and Rt. PCA stenosis	EIAB, Rt. (STA-M4)	NaN	0	NaT	NaN	...	0	0	0	1	0	1	1	1	0

1 rows × 280 columns

In [ ]:

In [ ]:
```
# Postoperative 3 year bypass patency of double barrel anastomosis cases (0: failure, 1: poor, 2: good, 3: good/good, 4: good/poor, 5: poor/poor)
df[df['bypass_type']==1]['3y_bypass_patency'].value_counts().sort_index()
```

Out[ ]:
```
3    8
4    1
Name: 3y_bypass_patency, dtype: int64
```

In [ ]:
```
df[df['bypass_type']==1]['5y_bypass_patency'].value_counts().sort_index()
```

Out[ ]:
```
3    8
4    1
Name: 5y_bypass_patency, dtype: int64
```

In [ ]:

In [ ]:
```
# Overall direct anastomosis graft diameter at preoperative
overall_bypass_graft_diameter['initial_STAb_diameter'].median(), overall_bypass_graft_diameter['initial_STAb_diameter'].quantile([0.25, 0.75])
```

Out[ ]:
```
(1.1,
 0.25     1.0
 0.75    1.22
 Name: initial_STAb_diameter, dtype: object)
```

In [ ]:
```
# Overall direct anastomosis graft diameter at postoperative 1 week
overall_bypass_graft_diameter['1w_STAb_diameter'].median(), overall_bypass_graft_diameter['1w_STAb_diameter'].quantile([0.25, 0.75])
```

Out[ ]:
```
(1.8,
 0.25    1.58
 0.75    2.02
 Name: 1w_STAb_diameter, dtype: object)
```

In [ ]:
```
# Overall direct anastomosis graft diameter at postoperative 1 year
overall_bypass_graft_diameter['1y_STAb_diameter'].median(), overall_bypass_graft_diameter['1y_STAb_diameter'].quantile([0.25, 0.75])
```

Out[ ]:
```
(2.27,
 0.25    1.955
 0.75    2.555
 Name: 1y_STAb_diameter, dtype: object)
```

In [ ]:
```
# Overall direct anastomosis graft diameter at postoperative 3 year
overall_bypass_graft_diameter['3y_STAb_diameter'].median(), overall_bypass_graft_diameter['3y_STAb_diameter'].quantile([0.25, 0.75])
```

Out[ ]:
```
(2.45,
 0.25    2.1375
 0.75      2.75
 Name: 3y_STAb_diameter, dtype: object)
```

In [ ]:
```
# Overall direct anastomosis graft diameter at postoperative 5 year
overall_bypass_graft_diameter['last_STAb_diameter'].median(), overall_bypass_graft_diameter['last_STAb_diameter'].quantile([0.25, 0.75])
```

Out[ ]:
```
(2.5,
 0.25    2.11
 0.75    2.84
 Name: last_STAb_diameter, dtype: object)
```

In [ ]:

In [ ]:
```
# Single direct anastomosis graft diameter at preoperative
df[(df['bypass_type']==0)|(df['bypass_type']==4)]['initial_STAb_diameter'].median(), df[(df['bypass_type']==0)|(df['bypass_type']==4)]['initial_STAb_diameter'].quantile([0.25, 0.75])
```

Out[ ]:
```
(1.09,
 0.25     1.01
 0.75    1.265
 Name: initial_STAb_diameter, dtype: object)
```

In [ ]:
```
# Single direct anastomosis graft diameter at postoperative 1 week
df[(df['bypass_type']==0)|(df['bypass_type']==4)]['1w_STAb_diameter'].median(), df[(df['bypass_type']==0)|(df['bypass_type']==4)]['1w_STAb_diameter'].quantile([0.25, 0.75])
```

Out[ ]:
```
(1.79,
 0.25    1.575
 0.75     2.02
 Name: 1w_STAb_diameter, dtype: object)
```

In [ ]:
```
# Single direct anastomosis graft diameter at postoperative 1 year
df[(df['bypass_type']==0)|(df['bypass_type']==4)]['1y_STAb_diameter'].median(), df[(df['bypass_type']==0)|(df['bypass_type']==4)]['1y_STAb_diameter'].quantile([0.25, 0.75])
```

Out[ ]:
```
(2.31,
 0.25    2.04
 0.75    2.56
 Name: 1y_STAb_diameter, dtype: object)
```

In [ ]:
```
# Single direct anastomosis graft diameter at postoperative 3 year
df[(df['bypass_type']==0)|(df['bypass_type']==4)]['3y_STAb_diameter'].median(), df[(df['bypass_type']==0)|(df['bypass_type']==4)]['3y_STAb_diameter'].quantile([0.25, 0.75])
```

Out[ ]:
```
(2.455,
 0.25    2.2275
 0.75    2.7575
 Name: 3y_STAb_diameter, dtype: object)
```

In [ ]:
```
# Single direct anastomosis graft diameter at postoperative 5 year
df[(df['bypass_type']==0)|(df['bypass_type']==4)]['5y_STAb_diameter'].median(), df[(df['bypass_type']==0)|(df['bypass_type']==4)]['5y_STAb_diameter'].quantile([0.25, 0.75])
```

Out[ ]:
```
(2.5549999999999997,
 0.25     2.245
 0.75    2.8425
 Name: 5y_STAb_diameter, dtype: object)
```

In [ ]:

In [ ]:
```
# Double barrel direct anastomosis graft diameter at preoperative
print(df[(df['bypass_type']==1)]['initial_STAb_diameter'].median(), df[(df['bypass_type']==1)]['initial_STAb_diameter'].quantile([0.25, 0.75]))
print(df[(df['bypass_type']==1)]['initial_STAb_diameter_second'].median(), df[(df['bypass_type']==1)]['initial_STAb_diameter_second'].quantile([0.25, 0.75]))
```

```
1.145 0.25    0.9925
0.75     1.215
Name: initial_STAb_diameter, dtype: object
1.0750000000000002 0.25    0.935
0.75    1.165
Name: initial_STAb_diameter_second, dtype: object
```

In [ ]:
```
# Double barrel direct anastomosis graft diameter at postoperative 1 week
print(df[(df['bypass_type']==1)]['1w_STAb_diameter'].median(), df[(df['bypass_type']==1)]['1w_STAb_diameter'].quantile([0.25, 0.75]))
print(df[(df['bypass_type']==1)]['1w_STAb_diameter_second'].median(), df[(df['bypass_type']==1)]['1w_STAb_diameter_second'].quantile([0.25, 0.75]))
```

```
1.85 0.25    1.7025
0.75    1.9925
Name: 1w_STAb_diameter, dtype: object
1.63 0.25    1.1625
0.75    1.8375
Name: 1w_STAb_diameter_second, dtype: object
```

In [ ]:
```
# Double barrel direct anastomosis graft diameter at postoperative 1 year
print(df[(df['bypass_type']==1)]['1y_STAb_diameter'].median(), df[(df['bypass_type']==1)]['1y_STAb_diameter'].quantile([0.25, 0.75]))
print(df[(df['bypass_type']==1)]['1y_STAb_diameter_second'].median(), df[(df['bypass_type']==1)]['1y_STAb_diameter_second'].quantile([0.25, 0.75]))
```

```
2.4 0.25    2.08
0.75    2.64
Name: 1y_STAb_diameter, dtype: object
1.89 0.25     1.3
0.75    2.12
Name: 1y_STAb_diameter_second, dtype: object
```

In [ ]:
```
# Double barrel direct anastomosis graft diameter at postoperative 3 year
print(df[(df['bypass_type']==1)]['3y_STAb_diameter'].median(), df[(df['bypass_type']==1)]['3y_STAb_diameter'].quantile([0.25, 0.75]))
print(df[(df['bypass_type']==1)]['3y_STAb_diameter_second'].median(), df[(df['bypass_type']==1)]['3y_STAb_diameter_second'].quantile([0.25, 0.75]))
```

```
2.45 0.25    2.19
0.75    2.65
Name: 3y_STAb_diameter, dtype: object
2.01 0.25    1.52
0.75    2.28
Name: 3y_STAb_diameter_second, dtype: object
```

In [ ]:
```
# Double barrel direct anastomosis graft diameter at postoperative 5 year
print(df[(df['bypass_type']==1)]['5y_STAb_diameter'].median(), df[(df['bypass_type']==1)]['5y_STAb_diameter'].quantile([0.25, 0.75]))
print(df[(df['bypass_type']==1)]['5y_STAb_diameter_second'].median(), df[(df['bypass_type']==1)]['5y_STAb_diameter_second'].quantile([0.25, 0.75]))
```

```
2.49 0.25     2.1
0.75    2.65
Name: 5y_STAb_diameter, dtype: object
2.13 0.25    1.42
0.75    2.32
Name: 5y_STAb_diameter_second, dtype: object
```

In [ ]:

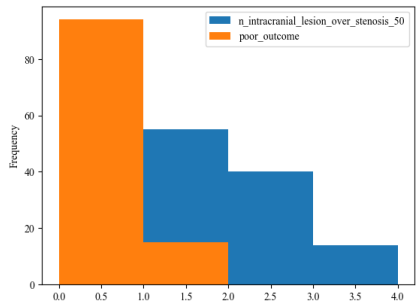
In [ ]:

Re-analysis of the important features from machine learning models for predicting poor outcome using traditional statistical test

Univariate analysis

In [ ]:
```
# Number of lesions with stenosis over 50% (Cochran-Armitage test for trend)
df[['n_intracranial_lesion_over_stenosis_50', 'poor_outcome']].plot(kind='hist', bins=4)
```

Out[ ]: <Axes: ylabel='Frequency'>



In [ ]:
```
pd.crosstab(df['poor_outcome'], df['n_intracranial_lesion_over_stenosis_50'])
```

Out[ ]:

n_intracranial_lesion_over_stenosis_50	1.0	2.0	3.0	4.0
poor_outcome				
0	54	28	11	1
1	1	12	0	2

In [ ]:
```
table = sm.stats.Table(pd.crosstab(df['poor_outcome'], df['n_intracranial_lesion_over_stenosis_50']))
print(table.test_ordinal_association())
```

```
null_mean   10.191780821917808
null_sd     2.83759317985763
pvalue      0.0019084830991753226
statistic   19.0
zscore      3.1041162773460447
```

In [ ]:

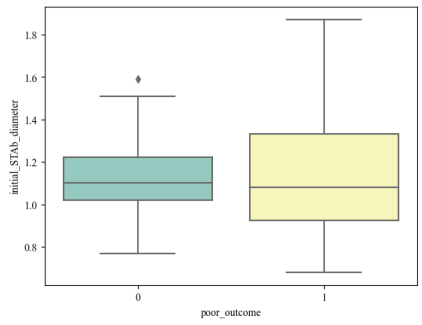
In [ ]:
```
# Basal collateral
ss.fisher_exact(pd.crosstab(df.collateral_basal, df.poor_outcome))
```

Out[ ]: SignificanceResult(statistic=0.14516129032258066, pvalue=0.06284919511635127)

In [ ]:

In [ ]:
```
# Preoperative diameter of STA branch (Welch's t test Result)
sns.boxplot(x='poor_outcome', y='initial_STAb_diameter', data=df, palette = 'Set3')
```

Out[ ]: <Axes: xlabel='poor_outcome', ylabel='initial_STAb_diameter'>



In [ ]:
```
print(ss.shapiro(df.initial_STAb_diameter[df.poor_outcome==0]))
print(ss.shapiro(df.initial_STAb_diameter[df.poor_outcome==1]))
```

```
ShapiroResult(statistic=0.9786454439163208, pvalue=0.12718097865581512)
ShapiroResult(statistic=0.9291804432868958, pvalue=0.2652782201766968)
```

In [ ]:
```
ss.levene(df.initial_STAb_diameter[df.poor_outcome==0], df.initial_STAb_diameter[df.poor_outcome==1])
```

Out[ ]: LeveneResult(statistic=13.913684340325473, pvalue=0.0003081506966668781)

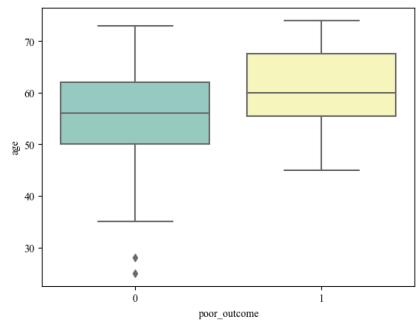
In [ ]:
```
ss.ttest_ind(df.initial_STAb_diameter[df.poor_outcome==0].astype('float'), df.initial_STAb_diameter[df.poor_outcome==1].astype('float'), equal_var=False)
```

Out[ ]: Ttest_indResult(statistic=-0.43335490453997894, pvalue=0.670897049715387)

In [ ]:

In [ ]:
```
# Age (Student's t-test)
sns.boxplot(x='poor_outcome', y='age', data=df, palette = 'Set3')
```

Out[ ]: <Axes: xlabel='poor_outcome', ylabel='age'>



In [ ]:
```
print(ss.shapiro(df.age[df.poor_outcome==0]))
print(ss.shapiro(df.age[df.poor_outcome==1]))
```

```
ShapiroResult(statistic=0.9708372950553894, pvalue=0.033716801553964615)
ShapiroResult(statistic=0.9680704474449158, pvalue=0.828521728515625)
```

In [ ]:
```
ss.levene(df.age[df.poor_outcome==0], df.age[df.poor_outcome==1])
```

Out[ ]: LeveneResult(statistic=0.013650166332831733, pvalue=0.9072106721045875)

In [ ]:
```
ss.ttest_ind(df.age[df.poor_outcome==0], df.age[df.poor_outcome==1], equal_var=True)
```

Out[ ]: Ttest_indResult(statistic=-1.7430859300353019, pvalue=0.08419124610463075)

In [ ]:

In [ ]:

In [ ]:
```
# Coronary artery disease (Fisher's exact test)
ss.fisher_exact(pd.crosstab(df.CAD, df.poor_outcome))
```

Out[ ]: SignificanceResult(statistic=3.6944444444444446, pvalue=0.04097406895023492)

In [ ]:

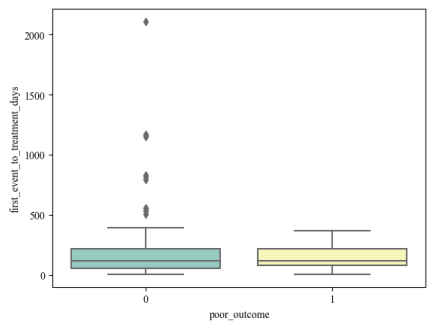
In [ ]:
```
# Willisian collateral (Fisher's exact test)
ss.fisher_exact(pd.crosstab(df.collateral_Willisian, df.poor_outcome))
```

Out[ ]: SignificanceResult(statistic=5.352941176470588, pvalue=0.10962664263707014)

In [ ]:

In [ ]:
```
# Days between the first event date and treatment date (Student's t-test)
sns.boxplot(x='poor_outcome', y='first_event_to_treatment_days', data=df, palette = 'Set3')
```

Out[ ]: <Axes: xlabel='poor_outcome', ylabel='first_event_to_treatment_days'>



In [ ]:
```
print(ss.shapiro(df.first_event_to_treatment_days[df.poor_outcome==0]))
print(ss.shapiro(df.first_event_to_treatment_days[df.poor_outcome==1]))
```

```
ShapiroResult(statistic=0.5925015211105347, pvalue=9.362892981811064e-15)
ShapiroResult(statistic=0.9354932308197021, pvalue=0.3290626108646393)
```

In [ ]:
```
ss.levene(df.first_event_to_treatment_days[df.poor_outcome==0], df.first_event_to_treatment_days[df.poor_outcome==1])
```

Out[ ]: LeveneResult(statistic=1.043513979427488, pvalue=0.30930968789227775)

In [ ]:
```
ss.ttest_ind(df.first_event_to_treatment_days[df.poor_outcome==0], df.first_event_to_treatment_days[df.poor_outcome==1], equal_var=True)
```

Out[ ]:
```
Ttest_indResult(statistic=0.7463476021020128, pvalue=0.4570941478563255)
```

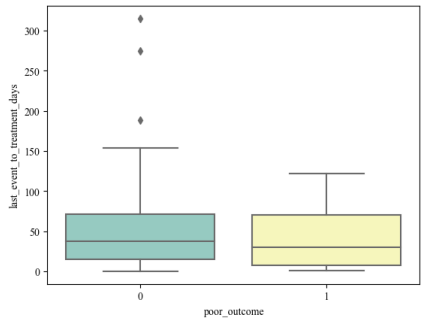
In [ ]:

In [ ]:

In [ ]:

In [ ]:
```
# Days between the last event date and treatment date (Student's t-test)
sns.boxplot(x='poor_outcome', y='last_event_to_treatment_days', data=df, palette = 'Set3')
```

Out[ ]:
```
<Axes: xlabel='poor_outcome', ylabel='last_event_to_treatment_days'>
```



In [ ]:
```
print(ss.shapiro(df.last_event_to_treatment_days[df.poor_outcome==0]))
print(ss.shapiro(df.last_event_to_treatment_days[df.poor_outcome==1]))
```

```
ShapiroResult(statistic=0.7650874853134155, pvalue=5.922615176068291e-11)
ShapiroResult(statistic=0.8828034996986389, pvalue=0.05225284397602081)
```

In [ ]:
```
ss.levene(df.last_event_to_treatment_days[df.poor_outcome==0], df.last_event_to_treatment_days[df.poor_outcome==1])
```

Out[ ]:
```
LeveneResult(statistic=0.024297610066936474, pvalue=0.8764237498312805)
```

In [ ]:
```
ss.ttest_ind(df.last_event_to_treatment_days[df.poor_outcome==0], df.last_event_to_treatment_days[df.poor_outcome==1], equal_var=True)
```

Out[ ]:
```
Ttest_indResult(statistic=0.6812120687092129, pvalue=0.4972089770655974)
```

In [ ]:

In [ ]:

Multivariate analysis - Binary logistic regression

In [ ]:
```
df['initial_STAb_diameter'] = df['initial_STAb_diameter'].astype('float')
df['CAD'] = df['CAD'].astype('int')
```

In [ ]:
```
X = df[['n_intracranial_lesion_over_stenosis_50', 'age', 'initial_STAb_diameter', 'CAD', 'collateral_basal', 'collateral_Willisian', 'first_event_to_treatment_days','last_event_to_treatment_days']]
y = df['poor_outcome']

X = sm.add_constant(X)
model = sm.Logit(y, X)
results = model.fit()

print(results.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.260845
         Iterations 9
                           Logit Regression Results
==============================================================================
Dep. Variable:           poor_outcome   No. Observations:                  109
Model:                          Logit   Df Residuals:                      100
Method:                           MLE   Df Model:                            8
Date:                Tue, 19 Dec 2023   Pseudo R-squ.:                  0.3489
Time:                        06:31:15   Log-Likelihood:                -28.432
converged:                       True   LL-Null:                       -43.666
Covariance Type:            nonrobust   LLR p-value:                 0.0001746
==============================================================================================
                                             coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------
const                                    -14.0913      4.850     -2.906      0.004     -23.596      -4.586
n_intracranial_lesion_over_stenosis_50     1.7272      0.533      3.240      0.001      0.682       2.772
age                                        0.1129      0.052      2.192      0.028      0.012       0.214
initial_STAb_diameter                      0.1834      1.433      0.128      0.898     -2.625       2.992
CAD                                        1.7544      0.752      2.332      0.020      0.280       3.229
collateral_basal                          -2.7346      1.259     -2.172      0.030     -5.203      -0.266
collateral_Willisian                       2.5032      1.313      1.907      0.057     -0.070       5.076
first_event_to_treatment_days             -0.0006      0.002     -0.280      0.779     -0.005       0.003
last_event_to_treatment_days              -0.0022      0.009     -0.247      0.805     -0.019       0.015
==============================================================================================
```

In [ ]:
```
confidenceInterval = results.conf_int()
confidenceInterval['OddsRatio'] = results.params
confidenceInterval.columns = ['2.5%', '97.5%', 'OddsRatio']
np.exp(confidenceInterval)
```

Out[ ]:

	2.5%	97.5%	OddsRatio
const	5.652027e-11	0.010192	7.589646e-07
n_intracranial_lesion_over_stenosis_50	1.978389e+00	15.991415	5.624699e+00
age	1.012010e+00	1.238451	1.119520e+00
initial_STAb_diameter	7.246262e-02	19.915851	1.201314e+00
CAD	1.322921e+00	25.251263	5.779743e+00
collateral_basal	5.501459e-03	0.766119	6.492130e-02
collateral_Willisian	9.326887e-01	160.149280	1.222168e+01
first_event_to_treatment_days	9.954602e-01	1.003417	9.994307e-01
last_event_to_treatment_days	9.808058e-01	1.015161	9.978356e-01

In [ ]:

In [ ]:

In [ ]:
```
X = df[['n_intracranial_lesion_over_stenosis_50', 'age', 'CAD', 'collateral_basal', 'collateral_Willisian']]
y = df['poor_outcome']

X = sm.add_constant(X)
model = sm.Logit(y, X)
results = model.fit()

print(results.summary())
```

```
Optimization terminated successfully.
         Current function value: 0.262203
         Iterations 8
                           Logit Regression Results
==============================================================================
Dep. Variable:           poor_outcome   No. Observations:                  109
Model:                          Logit   Df Residuals:                      103
Method:                           MLE   Df Model:                            5
Date:                Tue, 19 Dec 2023   Pseudo R-squ.:                  0.3455
Time:                        06:31:15   Log-Likelihood:                -28.580
converged:                       True   LL-Null:                       -43.666
Covariance Type:            nonrobust   LLR p-value:                 1.364e-05
==============================================================================================
                                             coef    std err          z      P>|z|      [0.025      0.975]
----------------------------------------------------------------------------------------------
const                                    -14.4742      4.296     -3.370      0.001     -22.893      -6.055
n_intracranial_lesion_over_stenosis_50     1.7532      0.535      3.276      0.001      0.704       2.802
age                                        0.1178      0.050      2.377      0.017      0.021       0.215
CAD                                        1.7459      0.747      2.338      0.019      0.282       3.210
collateral_basal                          -2.7624      1.227     -2.251      0.024      -5.167      -0.357
collateral_Willisian                       2.5625      1.290      1.986      0.047      0.034       5.092
==============================================================================================
```

In [ ]:
```
confidenceInterval = results.conf_int()
confidenceInterval['OddsRatio'] = results.params
confidenceInterval.columns = ['2.5%', '97.5%', 'OddsRatio']
np.exp(confidenceInterval)
```

Out[ ]:

	2.5%	97.5%	OddsRatio
const	1.141763e-10	0.002346	5.175160e-07
n_intracranial_lesion_over_stenosis_50	2.022464e+00	16.477579	5.772808e+00
age	1.020894e+00	1.239822	1.125046e+00
CAD	1.325789e+00	24.772946	5.730943e+00
collateral_basal	5.699071e-03	0.699470	6.313740e-02
collateral_Willisian	1.034103e+00	162.641071	1.296872e+01

In [ ]:
```
pred_y = results.predict(X)
pred_y
```

Out[ ]:
```
1      0.124931
2      0.291910
3      0.017700
4      0.017700
5      0.003112
         ...
105    0.000138
106    0.176253
107    0.294853
108    0.374917
109    0.171989
Length: 109, dtype: float64
```

In [ ]:
```
def cut_off(y, threshold):
    Y = y.copy()
    Y[Y>=threshold] = 1
    Y[Y<threshold] = 0
    return Y.astype(int)
```

In [ ]:

In [ ]:
```
# Cutoff value of 0.2
pred_Y = cut_off(pred_y, 0.2)
pred_Y
```

Out[ ]:
```
1      0
2      1
3      0
4      0
5      0
      ..
105    0
106    0
107    1
108    1
109    0
Length: 109, dtype: int32
```

In [ ]:
```
cfmat = metrics.confusion_matrix(y, pred_Y)
```

In [ ]:
```
cfmat
```

Out[ ]:
```
array([[80, 14],
       [ 3, 12]], dtype=int64)
```

In [ ]:
```
accuracy1=(cfmat[0,0]+cfmat[1,1])/sum(sum(cfmat))
print ('Accuracy : ', accuracy1)

sensitivity1 = cfmat[1,1]/(cfmat[1,0]+cfmat[1,1])
print('Sensitivity : ', sensitivity1 )

specificity1 = cfmat[0,0]/(cfmat[0,0]+cfmat[0,0])
print('Specificity : ', specificity1)
```

```
Accuracy :  0.8440366972477065
Sensitivity :  0.8
Specificity :  0.5
```

In [ ]:

In [ ]:
```
# Cutoff value of 0.3
pred_Y = cut_off(pred_y, 0.3)
pred_Y
```

Out[ ]:
```
1      0
2      0
3      0
4      0
5      0
      ..
105    0
106    0
107    0
108    1
109    0
Length: 109, dtype: int32
```

In [ ]:
```
cfmat = metrics.confusion_matrix(y, pred_Y)
```

In [ ]:
```
cfmat
```

Out[ ]:
```
array([[86,  8],
       [ 5, 10]], dtype=int64)
```

In [ ]:
```
accuracy1=(cfmat[0,0]+cfmat[1,1])/sum(sum(cfmat))
print ('Accuracy : ', accuracy1)

sensitivity1 = cfmat[1,1]/(cfmat[1,0]+cfmat[1,1])
print('Sensitivity : ', sensitivity1 )

specificity1 = cfmat[0,0]/(cfmat[0,0]+cfmat[0,0])
print('Specificity : ', specificity1)
```

```
Accuracy :  0.8807339449541285
Sensitivity :  0.6666666666666666
Specificity :  0.5
```

In [ ]:

In [ ]:
```
# Cutoff value of 0.4
pred_Y = cut_off(pred_y, 0.4)
pred_Y
```

Out[ ]:
```
1      0
2      0
3      0
4      0
5      0
      ..
105    0
106    0
107    0
108    0
109    0
Length: 109, dtype: int32
```

In [ ]:
```
cfmat = metrics.confusion_matrix(y, pred_Y)
```

In [ ]:
```
cfmat
```

Out[ ]:
```
array([[89,  5],
       [ 8,  7]], dtype=int64)
```

In [ ]:
```
accuracy1=(cfmat[0,0]+cfmat[1,1])/sum(sum(cfmat))
print ('Accuracy : ', accuracy1)

sensitivity1 = cfmat[1,1]/(cfmat[1,0]+cfmat[1,1])
print('Sensitivity : ', sensitivity1 )

specificity1 = cfmat[0,0]/(cfmat[0,0]+cfmat[0,0])
print('Specificity : ', specificity1)
```

```
Accuracy :  0.8807339449541285
Sensitivity :  0.4666666666666667
Specificity :  0.5
```

In [ ]:

In [ ]:
```
# Cutoff value of 0.5
pred_Y = cut_off(pred_y, 0.5)
pred_Y
```

Out[ ]:
```
1      0
2      0
3      0
4      0
5      0
      ..
105    0
106    0
107    0
108    0
109    0
Length: 109, dtype: int32
```

In [ ]:
```
cfmat = metrics.confusion_matrix(y, pred_Y)
```

In [ ]:
```
cfmat
```

Out[ ]:
```
array([[91,  3],
       [11,  4]], dtype=int64)
```

In [ ]:
```
accuracy1=(cfmat[0,0]+cfmat[1,1])/sum(sum(cfmat))
print ('Accuracy : ', accuracy1)

sensitivity1 = cfmat[1,1]/(cfmat[1,0]+cfmat[1,1])
print('Sensitivity : ', sensitivity1 )

specificity1 = cfmat[0,0]/(cfmat[0,0]+cfmat[0,0])
print('Specificity : ', specificity1)
```

```
Accuracy :  0.8715596330275229
Sensitivity :  0.2666666666666666
Specificity :  0.5
```

In [ ]:

In [ ]:

Subgroup analysis for the poor outcome

In [ ]:
```
# Association between poor outcome and Age group
df['age']
```

Out[ ]:
```
1      40.0
2      49.0
3      59.0
4      59.0
5      59.0
       ...
105    56.0
106    52.0
107    64.0
108    67.0
109    58.0
Name: age, Length: 109, dtype: float64
```

In [ ]:

In [ ]:
```
# Association between poor outcome and Age group
df.loc[:, 'age_group'] = df.age.apply(lambda x: 0 if x<50 else (2 if x>60 else 1))
```

In [ ]:
```
df.age_group
```

Out[ ]:
```
1      0
2      0
3      1
4      1
5      1
      ..
105    1
106    1
107    2
108    2
109    1
Name: age_group, Length: 109, dtype: int64
```

In [ ]:
```
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['age_group'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
2.282031529310629
0.3194943258930848
[[21.55963303  3.44036697]
 [43.98165138  7.01834862]
 [28.4587156   4.5412844 ]]
```

In [ ]:
```
print(df[df.age_group==0].poor_outcome.sum())
print(len(df[df.age_group==0].poor_outcome))
print(df[df.age_group==0].poor_outcome.sum()/len(df[df.age_group==0].poor_outcome)*100)
```

```
3
25
12.0
```

In [ ]:
```
print(df[df.age_group==1].poor_outcome.sum())
print(len(df[df.age_group==1].poor_outcome))
print(df[df.age_group==1].poor_outcome.sum()/len(df[df.age_group==1].poor_outcome)*100)
```

```
5
51
9.803921568627452
```

In [ ]:
```
print(df[df.age_group==2].poor_outcome.sum())
print(len(df[df.age_group==2].poor_outcome))
print(df[df.age_group==2].poor_outcome.sum()/len(df[df.age_group==2].poor_outcome)*100)
```

```
7
33
21.21212121212121
```

In [ ]:

In [ ]:
```
# Association between poor outcome and old age group (Fisher's exact test)
df.loc[:, 'old_age'] = df.age.apply(lambda x: 1 if x>60 else 0)
```

In [ ]:
```
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['old_age'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
1.404997695318245
0.23588858255611328
[[65.5412844 10.4587156]
 [28.4587156  4.5412844]]
```

In [ ]:
```
ss.fisher_exact(pd.crosstab(df['old_age'], df['poor_outcome']))
```

Out[ ]:
```
SignificanceResult(statistic=2.2884615384615383, pvalue=0.1446121411020166)
```

In [ ]:
```
print(df[df.old_age==0].poor_outcome.sum())
print(len(df[df.old_age==0].poor_outcome))
print(df[df.old_age==0].poor_outcome.sum()/len(df[df.old_age==0].poor_outcome)*100)
```

```
8
76
10.526315789473683
```

In [ ]:
```
print(df[df.old_age==1].poor_outcome.sum())
print(len(df[df.old_age==1].poor_outcome))
print(df[df.old_age==1].poor_outcome.sum()/len(df[df.old_age==1].poor_outcome)*100)
```

```
7
33
21.21212121212121
```

In [ ]:

In [ ]:
```
# Association between poor outcome and Sex
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['sex'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
0.0
1.0
[[68.99082569 11.00917431]
 [25.00917431  3.99082569]]
```

In [ ]:
```
print(df[df.sex==0].poor_outcome.sum())
print(len(df[df.sex==0].poor_outcome))
print(df[df.sex==0].poor_outcome.sum()/len(df[df.sex==0].poor_outcome)*100)
```

```
11
80
13.750000000000002
```

In [ ]:
```
print(df[df.sex==1].poor_outcome.sum())
print(len(df[df.sex==1].poor_outcome))
print(df[df.sex==1].poor_outcome.sum()/len(df[df.sex==1].poor_outcome)*100)
```

```
4
29
13.793103448275861
```

In [ ]:

In [ ]:
```
# Association between poor outcome and previous EVT
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['previous_EVT'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
7.080783365570599
0.06936702200370615
[[85.37614679 13.62385321]
 [ 3.44954128  0.55045872]
 [ 4.31192661  0.68807339]
 [ 0.86238532  0.13761468]]
```

In [ ]:
```
print(df[df.previous_EVT==0].poor_outcome.sum())
print(len(df[df.previous_EVT==0].poor_outcome))
print(df[df.previous_EVT==0].poor_outcome.sum()/len(df[df.previous_EVT==0].poor_outcome)*100)
```

```
12
99
12.121212121212121
```

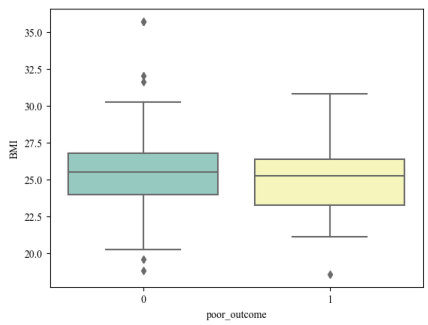
In [ ]:
```
print(df[df.previous_EVT==1].poor_outcome.sum())
print(len(df[df.previous_EVT==1].poor_outcome))
print(df[df.previous_EVT==1].poor_outcome.sum()/len(df[df.previous_EVT==1].poor_outcome)*100)
```

```
1
4
25.0
```

In [ ]:

In [ ]:
```
# Association between poor outcome and BMI (Mann-Whitney U test)
sns.boxplot(x='poor_outcome', y='BMI', data=df, palette = 'Set3')
```

Out[ ]:
```
<Axes: xlabel='poor_outcome', ylabel='BMI'>
```



In [ ]:
```
df.BMI = df.BMI.astype('float')
```

In [ ]:
```
print(ss.shapiro(df.BMI[df.poor_outcome==0]))
print(ss.shapiro(df.BMI[df.poor_outcome==1]))
```

```
ShapiroResult(statistic=0.9547408223152161, pvalue=0.0025514115113765)
ShapiroResult(statistic=0.9824836850166321, pvalue=0.9836516380310059)
```

In [ ]:
```
ss.mannwhitneyu(df.BMI[df.poor_outcome==0], df.BMI[df.poor_outcome==1])
```

Out[ ]:
```
MannwhitneyuResult(statistic=797.5, pvalue=0.4183703834410861)
```

In [ ]:

In [ ]:
```
df.loc[:, 'BMI_obesity'] = df.BMI.apply(lambda x: 0 if x<25 else 1)
```

In [ ]:
```
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['BMI_obesity'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
0.0
1.0
[[39.66972477  6.33027523]
 [54.33027523  8.66972477]]
```

In [ ]:
```
print(df[df.BMI_obesity==0].poor_outcome.sum())
print(len(df[df.BMI_obesity==0].poor_outcome))
print(df[df.BMI_obesity==0].poor_outcome.sum()/len(df[df.BMI_obesity==0].poor_outcome)*100)
```

```
6
46
13.043478260869565
```

In [ ]:
```
print(df[df.BMI_obesity==1].poor_outcome.sum())
print(len(df[df.BMI_obesity==1].poor_outcome))
print(df[df.BMI_obesity==1].poor_outcome.sum()/len(df[df.BMI_obesity==1].poor_outcome)*100)
```

```
9
63
14.285714285714285
```

In [ ]:

In [ ]:
```
# Association between poor outcome and Hypertension (Fisher's exact test)
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['HTN'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
1.4631954005006258
0.22642238270625 02
[[20.69724771  3.30275229]
 [73.30275229 11.69724771]]
```

In [ ]:
```
ss.fisher_exact(pd.crosstab(df['HTN'], df['poor_outcome']))
```

Out[ ]:
```
SignificanceResult(statistic=4.535211267605634, pvalue=0.18269985298721847)
```

In [ ]:
```
print(df[df.HTN==0].poor_outcome.sum())
print(len(df[df.HTN==0].poor_outcome))
print(df[df.HTN==0].poor_outcome.sum()/len(df[df.HTN==0].poor_outcome)*100)
```

```
1
24
4.166666666666666
```

In [ ]:
```
print(df[df.HTN==1].poor_outcome.sum())
print(len(df[df.HTN==1].poor_outcome))
print(df[df.HTN==1].poor_outcome.sum()/len(df[df.HTN==1].poor_outcome)*100)
```

```
14
85
16.470588235294116
```

In [ ]:

In [ ]:

In [ ]:
```
# Association between poor outcome and DM
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['DM'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
1.9122459707776471
0.16671398589428987
[[50.01834862  7.98165138]
 [43.98165138  7.01834862]]
```

In [ ]:
```
print(df[df.DM==0].poor_outcome.sum())
print(len(df[df.DM==0].poor_outcome))
print(df[df.DM==0].poor_outcome.sum()/len(df[df.DM==0].poor_outcome)*100)
```

```
5
58
8.620689655172415
```

In [ ]:
```
print(df[df.DM==1].poor_outcome.sum())
print(len(df[df.DM==1].poor_outcome))
print(df[df.DM==1].poor_outcome.sum()/len(df[df.DM==1].poor_outcome)*100)
```

```
10
51
19.607843137254903
```

In [ ]:

In [ ]:
```
# Association between poor outcome and Dyslipidemia
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['DL'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
0.8163469768000958
0.3662503934589181
[[50.88073394  8.11926606]
 [43.11926606  6.88073394]]
```

In [ ]:
```
print(df[df.DL==0].poor_outcome.sum())
print(len(df[df.DL==0].poor_outcome))
print(df[df.DL==0].poor_outcome.sum()/len(df[df.DL==0].poor_outcome)*100)
```

```
6
59
10.16949152542373
```

In [ ]:
```
print(df[df.DL==1].poor_outcome.sum())
print(len(df[df.DL==1].poor_outcome))
print(df[df.DL==1].poor_outcome.sum()/len(df[df.DL==1].poor_outcome)*100)
```

```
9
50
18.0
```

In [ ]:

In [ ]:
```
# Association between poor outcome and CAD
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['CAD'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
4.094300996285039
0.04302801846336615
[[72.44036697 11.55963303]
 [21.55963303  3.44036697]]
```

In [ ]:
```
ss.fisher_exact(pd.crosstab(df['CAD'], df['poor_outcome']))
```

Out[ ]:
```
SignificanceResult(statistic=3.6944444444444446, pvalue=0.04097406895023492)
```

In [ ]:
```
print(df[df.CAD==0].poor_outcome.sum())
print(len(df[df.CAD==0].poor_outcome))
print(df[df.CAD==0].poor_outcome.sum()/len(df[df.CAD==0].poor_outcome)*100)
```

```
8
84
9.523809523809524
```

In [ ]:
```
print(df[df.CAD==1].poor_outcome.sum())
print(len(df[df.CAD==1].poor_outcome))
print(df[df.CAD==1].poor_outcome.sum()/len(df[df.CAD==1].poor_outcome)*100)
```

```
7
25
28.000000000000004
```

In [ ]:

In [ ]:
```
# Association between poor outcome and arrhythmia
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['arrhythmia'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
0.021939537891966624
0.8822480565240814
[[91.41284404 14.58715596]
 [ 2.58715596  0.41284404]]
```

In [ ]:
```
ss.fisher_exact(pd.crosstab(df['arrhythmia'], df['poor_outcome']))
```

Out[ ]:
```
SignificanceResult(statistic=3.2857142857142856, pvalue=0.3614945649585107)
```

In [ ]:
```
print(df[df.arrhythmia==0].poor_outcome.sum())
print(len(df[df.arrhythmia==0].poor_outcome))
print(df[df.arrhythmia==0].poor_outcome.sum()/len(df[df.arrhythmia==0].poor_outcome)*100)
```

```
14
106
13.20754716981132
```

In [ ]:
```
print(df[df.arrhythmia==1].poor_outcome.sum())
print(len(df[df.arrhythmia==1].poor_outcome))
print(df[df.arrhythmia==1].poor_outcome.sum()/len(df[df.arrhythmia==1].poor_outcome)*100)
```

```
1
3
33.33333333333333
```

In [ ]:

In [ ]:
```
# Association between poor outcome and family history of stroke
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['FHx'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
0.18109218018074874
0.6704363134646918
[[61.2293578  9.7706422]
 [32.7706422  5.2293578]]
```

In [ ]:
```
print(df[df.FHx==0].poor_outcome.sum())
print(len(df[df.FHx==0].poor_outcome))
print(df[df.FHx==0].poor_outcome.sum()/len(df[df.FHx==0].poor_outcome)*100)
```

```
11
71
15.492957746478872
```

In [ ]:

In [ ]:
```
print(df[df.FHx==1].poor_outcome.sum())
print(len(df[df.FHx==1].poor_outcome))
print(df[df.FHx==1].poor_outcome.sum()/len(df[df.FHx==1].poor_outcome)*100)
```

```
4
38
10.526315789473683
```

In [ ]:

In [ ]:
```
# Association between poor outcome and smoking
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['smoking'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
0.06923666487871741
0.9659740268602116
[[36.22018349  5.77981651]
 [14.66055046  2.33944954]
 [43.11926606  6.88073394]]
```

In [ ]:
```
print(df[df.smoking==0].poor_outcome.sum())
print(len(df[df.smoking==0].poor_outcome))
print(df[df.smoking==0].poor_outcome.sum()/len(df[df.smoking==0].poor_outcome)*100)
```

```
6
42
14.285714285714285
```

In [ ]:
```
print(df[df.smoking==1].poor_outcome.sum())
print(len(df[df.smoking==1].poor_outcome))
print(df[df.smoking==1].poor_outcome.sum()/len(df[df.smoking==1].poor_outcome)*100)
```

```
2
17
11.76470588235294
```

In [ ]:
```
print(df[df.smoking==2].poor_outcome.sum())
print(len(df[df.smoking==2].poor_outcome))
print(df[df.smoking==2].poor_outcome.sum()/len(df[df.smoking==2].poor_outcome)*100)
```

```
7
50
14.000000000000002
```

In [ ]:

In [ ]:
```
# Association between poor outcome and previous stroke
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['prev_stroke'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
3.8647136713445005
0.04931160704258262 5
[[81.06422018 12.93577982]
 [12.93577982  2.06422018]]
```

In [ ]:
```
ss.fisher_exact(pd.crosstab(df['prev_stroke'], df['poor_outcome']))
```

Out[ ]:
```
SignificanceResult(statistic=4.2, pvalue=0.03275563873712087)
```

In [ ]:
```
print(df[df.prev_stroke==0].poor_outcome.sum())
print(len(df[df.prev_stroke==0].poor_outcome))
print(df[df.prev_stroke==0].poor_outcome.sum()/len(df[df.prev_stroke==0].poor_outcome)*100)
```

```
10
94
10.638297872340425
```

In [ ]:
```
print(df[df.prev_stroke==1].poor_outcome.sum())
print(len(df[df.prev_stroke==1].poor_outcome))
print(df[df.prev_stroke==1].poor_outcome.sum()/len(df[df.prev_stroke==1].poor_outcome)*100)
```

```
5
15
33.33333333333333
```

In [ ]:

In [ ]:
```
# Association between poor outcome and presentation (TIA vs infarction) (Fisher's exact test)
tia_infarction_df = df[(df['presentation']==1)|(df['presentation']==2)]
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(tia_infarction_df['presentation'], tia_infarction_df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
0.033339262949543752 4
0.855004772801992
[[23.21495327  3.78504673]
 [68.78504673 11.21495327]]
```

In [ ]:
```
ss.fisher_exact(pd.crosstab(tia_infarction_df['presentation'], tia_infarction_df['poor_outcome']))
```

Out[ ]:
```
SignificanceResult(statistic=1.411764705882353, pvalue=0.7561447870075224)
```

In [ ]:
```
print(tia_infarction_df[tia_infarction_df.presentation==1].poor_outcome.sum())
print(len(tia_infarction_df[tia_infarction_df.presentation==1].poor_outcome))
print(tia_infarction_df[tia_infarction_df.presentation==1].poor_outcome.sum()/len(tia_infarction_df[tia_infarction_df.presentation==1].poor_outcome)*100)
```

```
3
27
11.11111111111111
```

In [ ]:
```
print(tia_infarction_df[tia_infarction_df.presentation==2].poor_outcome.sum())
print(len(tia_infarction_df[tia_infarction_df.presentation==2].poor_outcome))
print(tia_infarction_df[tia_infarction_df.presentation==2].poor_outcome.sum()/len(tia_infarction_df[tia_infarction_df.presentation==2].poor_outcome)*100)
```

```
12
80
15.0
```

In [ ]:

In [ ]:
```
# Association between poor outcome and presentation multiple infarction
infarction_df = df[df['presentation']==2]
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(infarction_df['multiple_infarction'], infarction_df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
1.3975044563279857
0.2371418532091719
[[46.75  8.25]
 [21.25  3.75]]
```

In [ ]:
```
ss.fisher_exact(pd.crosstab(infarction_df['multiple_infarction'], infarction_df['poor_outcome']))
```

Out[ ]:
```
SignificanceResult(statistic=2.5789473684210527, pvalue=0.17700503575371482)
```

In [ ]:
```
print(infarction_df[infarction_df.multiple_infarction==0].poor_outcome.sum())
print(len(infarction_df[infarction_df.multiple_infarction==0].poor_outcome))
print(infarction_df[infarction_df.multiple_infarction==0].poor_outcome.sum()/len(infarction_df[infarction_df.multiple_infarction==0].poor_outcome)*100)
```

```
6
55
10.909090909090908
```

In [ ]:
```
print(infarction_df[infarction_df.multiple_infarction==1].poor_outcome.sum())
print(len(infarction_df[infarction_df.multiple_infarction==1].poor_outcome))
print(infarction_df[infarction_df.multiple_infarction==1].poor_outcome.sum()/len(infarction_df[infarction_df.multiple_infarction==1].poor_outcome)*100)
```

```
6
25
24.0
```

In [ ]:

In [ ]:
```
# Association between poor outcome and preoperative mRS grade
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['preop_mRS_grade'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
5.452452202562253
0.06546588568337018
[[56.91743119  9.08256881]
 [25.00917431  3.99082569]
 [12.0733945   1.9266055 ]]
```

In [ ]:
```
print(df[df.preop_mRS_grade==0].poor_outcome.sum())
print(len(df[df.preop_mRS_grade==0].poor_outcome))
print(df[df.preop_mRS_grade==0].poor_outcome.sum()/len(df[df.preop_mRS_grade==0].poor_outcome)*100)
```

```
5
66
7.575757575757576
```

In [ ]:
```
print(df[df.preop_mRS_grade==1].poor_outcome.sum())
print(len(df[df.preop_mRS_grade==1].poor_outcome))
print(df[df.preop_mRS_grade==1].poor_outcome.sum()/len(df[df.preop_mRS_grade==1].poor_outcome)*100)
```

```
7
29
24.137931034482758
```

In [ ]:
```
print(df[df.preop_mRS_grade==2].poor_outcome.sum())
print(len(df[df.preop_mRS_grade==2].poor_outcome))
print(df[df.preop_mRS_grade==2].poor_outcome.sum()/len(df[df.preop_mRS_grade==2].poor_outcome)*100)
```

```
3
14
21.428571428571427
```

In [ ]:

In [ ]:
```
df.loc[:, 'preop_mRS_good'] = df['preop_mRS_grade'].apply(lambda x: 1 if x==0 else 0)
```

In [ ]:
```
df.preop_mRS_good.sum()
```

Out[ ]:
```
66
```

In [ ]:
```
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['preop_mRS_good'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
4.153718593655506
0.0415431961209197
[[37.08256881  5.91743119]
 [56.91743119  9.08256881]]
```

In [ ]:
```
print(df[df.preop_mRS_good==0].poor_outcome.sum())
print(len(df[df.preop_mRS_good==0].poor_outcome))
print(df[df.preop_mRS_good==0].poor_outcome.sum()/len(df[df.preop_mRS_good==0].poor_outcome)*100)
```

```
10
43
23.25581395348837
```

In [ ]:
```
print(df[df.preop_mRS_good==1].poor_outcome.sum())
print(len(df[df.preop_mRS_good==1].poor_outcome))
print(df[df.preop_mRS_good==1].poor_outcome.sum()/len(df[df.preop_mRS_good==1].poor_outcome)*100)
```

```
5
66
7.575757575757576
```

In [ ]:

In [ ]:
```
# Association between poor outcome and preoperative NIHSS grade
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['preop_NIHSS_grade'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
7.9248215807858795
0.019017212470883556
[[57.77981651  9.22018349]
 [32.7706422   5.2293578 ]
 [ 3.44954128  0.55045872]]
```

In [ ]:
```
print(df[df.preop_NIHSS_grade==0].poor_outcome.sum())
print(len(df[df.preop_NIHSS_grade==0].poor_outcome))
print(df[df.preop_NIHSS_grade==0].poor_outcome.sum()/len(df[df.preop_NIHSS_grade==0].poor_outcome)*100)
```

```
5
67
7.462686567164178
```

In [ ]:
```
print(df[df.preop_NIHSS_grade==1].poor_outcome.sum())
print(len(df[df.preop_NIHSS_grade==1].poor_outcome))
print(df[df.preop_NIHSS_grade==1].poor_outcome.sum()/len(df[df.preop_NIHSS_grade==1].poor_outcome)*100)
```

```
10
38
26.31578947368421
```

In [ ]:
```
print(df[df.preop_NIHSS_grade==2].poor_outcome.sum())
print(len(df[df.preop_NIHSS_grade==2].poor_outcome))
print(df[df.preop_NIHSS_grade==2].poor_outcome.sum()/len(df[df.preop_NIHSS_grade==2].poor_outcome)*100)
```

```
0
4
0.0
```

In [ ]:
```
print(df[df.preop_NIHSS_grade==3].poor_outcome.sum())
print(len(df[df.preop_NIHSS_grade==3].poor_outcome))
print(df[df.preop_NIHSS_grade==3].poor_outcome.sum()/len(df[df.preop_NIHSS_grade==3].poor_outcome)*100)
```

```
0
0
nan
```

In [ ]:
```
# Association between poor outcome and preoperative NIHSS good grade
df.loc[:, 'preop_NIHSS_good'] = df['preop_NIHSS_grade'].apply(lambda x: 1 if x==0 else 0)
```

In [ ]:
```
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['preop_NIHSS_good'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
4.51715516893748
0.033556584503584154
[[36.22018349  5.77981651]
 [57.77981651  9.22018349]]
```

In [ ]:
```
print(df[df.preop_NIHSS_good==1].poor_outcome.sum())
print(len(df[df.preop_NIHSS_good==1].poor_outcome))
print(df[df.preop_NIHSS_good==1].poor_outcome.sum()/len(df[df.preop_NIHSS_good==1].poor_outcome)*100)
```

```
5
67
7.462686567164178
```

In [ ]:
```
print(df[df.preop_NIHSS_good==0].poor_outcome.sum())
print(len(df[df.preop_NIHSS_good==0].poor_outcome))
print(df[df.preop_NIHSS_good==0].poor_outcome.sum()/len(df[df.preop_NIHSS_good==0].poor_outcome)*100)
```

```
10
42
23.809523809523807
```

In [ ]:

In [ ]:
```
# Association between poor outcome and stenotic or occluded lesion (Fisher's exact test)
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['lesion_stenosis_occlusion'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
0.12370164981640767
0.7250539201849798
[[81.06422018 12.93577982]
 [12.93577982  2.06422018]]
```

In [ ]:
```
ss.fisher_exact(pd.crosstab(df['lesion_stenosis_occlusion'], df['poor_outcome']))
```

Out[ ]:
```
SignificanceResult(statistic=1.7083333333333333, pvalue=0.4310822658836879)
```

In [ ]:
```
print(df[df.lesion_stenosis_occlusion==0].poor_outcome.sum())
print(len(df[df.lesion_stenosis_occlusion==0].poor_outcome))
print(df[df.lesion_stenosis_occlusion==0].poor_outcome.sum()/len(df[df.lesion_stenosis_occlusion==0].poor_outcome)*100)
```

```
12
94
12.76595744680851
```

In [ ]:
```
print(df[df.lesion_stenosis_occlusion==1].poor_outcome.sum())
print(len(df[df.lesion_stenosis_occlusion==1].poor_outcome))
print(df[df.lesion_stenosis_occlusion==1].poor_outcome.sum()/len(df[df.lesion_stenosis_occlusion==1].poor_outcome)*100)
```

```
3
15
20.0
```

In [ ]:

In [ ]:
```
# Association between poor outcome and single ICAD lesion (Fisher's exact test)
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['single_ICAD'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
1.2227718960265501
0.2688173515035899
[[11.21100917  1.78899083]
 [82.78899083 13.21100917]]
```

In [ ]:
```
ss.fisher_exact(pd.crosstab(df['single_ICAD'], df['poor_outcome']))
```

Out[ ]:
```
SignificanceResult(statistic=inf, pvalue=0.20765186480402245)
```

In [ ]:
```
print(df[df.single_ICAD==0].poor_outcome.sum())
print(len(df[df.single_ICAD==0].poor_outcome))
print(df[df.single_ICAD==0].poor_outcome.sum()/len(df[df.single_ICAD==0].poor_outcome)*100)
```

```
0
13
0.0
```

In [ ]:
```
print(df[df.single_ICAD==1].poor_outcome.sum())
print(len(df[df.single_ICAD==1].poor_outcome))
print(df[df.single_ICAD==1].poor_outcome.sum()/len(df[df.single_ICAD==1].poor_outcome)*100)
```

```
15
96
15.625
```

In [ ]:

In [ ]:
```
# Association between poor outcome and multiple bilateral ICAD lesions (Fisher's exact test)
multiple_ICAD_df = df[df['single_ICAD']==1]
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(multiple_ICAD_df['multiple_ICAD_bilateral'], multiple_ICAD_df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
0.25810699588477365
0.6114236667183403
[[ 5.0625  0.9375]
 [75.9375 14.0625]]
```

In [ ]:
```
ss.fisher_exact(pd.crosstab(multiple_ICAD_df['multiple_ICAD_bilateral'], multiple_ICAD_df['poor_outcome']))
```

Out[ ]:
```
SignificanceResult(statistic=inf, pvalue=0.5854326701837103)
```

In [ ]:
```
print(multiple_ICAD_df[multiple_ICAD_df.multiple_ICAD_bilateral==0].poor_outcome.sum())
print(len(multiple_ICAD_df[multiple_ICAD_df.multiple_ICAD_bilateral==0].poor_outcome))
print(multiple_ICAD_df[multiple_ICAD_df.multiple_ICAD_bilateral==0].poor_outcome.sum()/len(multiple_ICAD_df[multiple_ICAD_df.multiple_ICAD_bilateral==0].poor_outcome)*100)
```

```
0
6
0.0
```

In [ ]:
```
print(multiple_ICAD_df[multiple_ICAD_df.multiple_ICAD_bilateral==1].poor_outcome.sum())
print(len(multiple_ICAD_df[multiple_ICAD_df.multiple_ICAD_bilateral==1].poor_outcome))
print(multiple_ICAD_df[multiple_ICAD_df.multiple_ICAD_bilateral==1].poor_outcome.sum()/len(multiple_ICAD_df[multiple_ICAD_df.multiple_ICAD_bilateral==1].poor_outcome)*100)
```

```
15
90
16.666666666666664
```

In [ ]:

In [ ]:
```
# Association between poor outcome and Willisian collateral (Fisher's exact test)
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['collateral_Willisian'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
2.036404457129678
0.15357227399747062
[[23.28440367  3.71559633]
 [70.71559633 11.28440367]]
```

In [ ]:
```
ss.fisher_exact(pd.crosstab(df['collateral_Willisian'], df['poor_outcome']))
```

Out[ ]:
```
SignificanceResult(statistic=5.352941176470588, pvalue=0.10962664263707014)
```

In [ ]:
```
print(df[df.collateral_Willisian==0].poor_outcome.sum())
print(len(df[df.collateral_Willisian==0].poor_outcome))
print(df[df.collateral_Willisian==0].poor_outcome.sum()/len(df[df.collateral_Willisian==0].poor_outcome)*100)
```

```
1
27
3.7037037037037033
```

In [ ]:
```
print(df[df.collateral_Willisian==1].poor_outcome.sum())
print(len(df[df.collateral_Willisian==1].poor_outcome))
print(df[df.collateral_Willisian==1].poor_outcome.sum()/len(df[df.collateral_Willisian==1].poor_outcome)*100)
```

```
14
82
17.073170731707318
```

In [ ]:

In [ ]:
```
# Association between poor outcome and basal collateral (Fisher's exact test)
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['collateral_basal'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
3.1427809391406463
0.07626367212417438
[[66.40366972 10.59633028]
 [27.59633028  4.40366972]]
```

In [ ]:
```
ss.fisher_exact(pd.crosstab(df['collateral_basal'], df['poor_outcome']))
```

Out[ ]:
```
SignificanceResult(statistic=0.14516129032258066, pvalue=0.06284919511635127)
```

In [ ]:
```
print(df[df.collateral_basal==0].poor_outcome.sum())
print(len(df[df.collateral_basal==0].poor_outcome))
print(df[df.collateral_basal==0].poor_outcome.sum()/len(df[df.collateral_basal==0].poor_outcome)*100)
```

```
14
77
18.181818181818183
```

In [ ]:
```
print(df[df.collateral_basal==1].poor_outcome.sum())
print(len(df[df.collateral_basal==1].poor_outcome))
print(df[df.collateral_basal==1].poor_outcome.sum()/len(df[df.collateral_basal==1].poor_outcome)*100)
```

```
1
32
3.125
```

In [ ]:

In [ ]:
```
# Association between poor outcome and transdural collateral
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['collateral_transdural'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
1.3250028908418137
0.2496961580763468 2
[[59.50458716  9.49541284]
 [34.49541284  5.50458716]]
```

In [ ]:
```
print(df[df.collateral_transdural==0].poor_outcome.sum())
print(len(df[df.collateral_transdural==0].poor_outcome))
print(df[df.collateral_transdural==0].poor_outcome.sum()/len(df[df.collateral_transdural==0].poor_outcome)*100)
```

```
7
69
10.144927536231885
```

In [ ]:
```
print(df[df.collateral_transdural==1].poor_outcome.sum())
print(len(df[df.collateral_transdural==1].poor_outcome))
print(df[df.collateral_transdural==1].poor_outcome.sum()/len(df[df.collateral_transdural==1].poor_outcome)*100)
```

```
8
40
20.0
```

In [ ]:

In [ ]:
```
# Association between Basal collateral and MCA occlusion
df_mcao = df[(df['MCA_lesion_type'].notnull())&(df['lesion_stenosis_occlusion']==0)]
df_mcao.index
```

Out[ ]:
```
Int64Index([  2,  10,  12,  15,  19,  21,  24,  31,  32,  33,  34,  44,  48,
            50,  52,  58,  59,  60,  61,  63,  64,  66,  67,  68,  76,  78,
            79,  80,  81,  82,  83,  84,  85,  86,  96, 104],
           dtype='int64')
```

In [ ]:
```
df.loc[df_mcao.index, 'MCAO'] = 1
```

In [ ]:
```
df['MCAO'].fillna(0, inplace=True)
```

In [ ]:
```
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['MCAO'], df['collateral_basal']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
9.607919837131092
0.0019373999787372724
[[51.56880734 21.43119266]
 [25.43119266 10.56880734]]
```

In [ ]:

In [ ]:
```
# Association between Willisian collateral and ICA occlusion
df_icao = df[(df['ICA_lesion_type'].notnull())&(df['lesion_stenosis_occlusion']==0)]
df_mcao.index
```

Out[ ]:
```
Int64Index([  2,  10,  12,  15,  19,  21,  24,  31,  32,  33,  34,  44,  48,
            50,  52,  58,  59,  60,  61,  63,  64,  66,  67,  68,  76,  78,
            79,  80,  81,  82,  83,  84,  85,  86,  96, 104],
           dtype='int64')
```

In [ ]:
```
df.loc[df_icao.index, 'ICAO'] = 1
```

In [ ]:
```
df['ICAO'].fillna(0, inplace=True)
```

In [ ]:
```
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['ICAO'], df['collateral_Willisian']))
```

In [ ]:
```
pd.crosstab(df['ICAO'], df['collateral_Willisian'])
```

Out[ ]:

collateral_Willisian	0	1
ICAO		
0.0	23	28
1.0	4	54

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
19.251808860634245
1.1456178351163557e-05
[[12.63302752 38.36697248]
 [14.36697248 43.63302752]]
```

In [ ]:

In [ ]:
```
# Association between poor outcome and MCA occlusion
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['MCAO'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
0.73899339626715
0.3899835200820135 4
[[62.95412844 10.04587156]
 [31.04587156  4.95412844]]
```

In [ ]:
```
print(df[df.MCAO==0].poor_outcome.sum())
print(len(df[df.MCAO==0].poor_outcome))
print(df[df.MCAO==0].poor_outcome.sum()/len(df[df.MCAO==0].poor_outcome)*100)
```

```
12
73
16.43835616438356
```

In [ ]:
```
print(df[df.MCAO==1].poor_outcome.sum())
print(len(df[df.MCAO==1].poor_outcome))
print(df[df.MCAO==1].poor_outcome.sum()/len(df[df.MCAO==1].poor_outcome)*100)
```

```
3
36
8.333333333333332
```

In [ ]:

In [ ]:
```
# Association between poor outcome and ICA occlusion
chi, p, dof, expected = ss.chi2_contingency(pd.crosstab(df['ICAO'], df['poor_outcome']))
```

In [ ]:
```
print(chi)
print(p)
print(expected)
```

```
0.08342690096336919
0.7727059992468828
[[43.98165138  7.01834862]
 [50.01834862  7.98165138]]
```

In [ ]:
```
print(df[df.ICAO==0].poor_outcome.sum())
print(len(df[df.ICAO==0].poor_outcome))
print(df[df.ICAO==0].poor_outcome.sum()/len(df[df.ICAO==0].poor_outcome)*100)
```

```
6
51
11.76470588235294
```

In [ ]:
```
print(df[df.ICAO==1].poor_outcome.sum())
print(len(df[df.ICAO==1].poor_outcome))
print(df[df.ICAO==1].poor_outcome.sum()/len(df[df.ICAO==1].poor_outcome)*100)
```

```
9
58
15.517241379310345
```

In [ ]: