

Supplemental material 2 - Variables transformation and models diagnostics

Lucas Krüger

2024-02-08

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##   filter, lag
```

```
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
library(lmerTest)
```

```
## Loading required package: lme4
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'lmerTest'
```

```
## The following object is masked from 'package:lme4':
##
##   lmer
```

```
## The following object is masked from 'package:stats':
##
##   step
```

```

library(sjPlot)
library(patchwork)

dfS<-readRDS("C:/GiantSuperModel/WindUse/GiantsOnIce/GiantPetrels_processed_trips.Rds")

daydf<-plyr::ddply(dfS, c("daystamp", "tripID", "Ring", "Trip", "beh.state", "head.tails"), summarise,
  MCD=max(ColDist), # maximum colony distance for the day, trip, and state
  MTD=max(total_dist), #maximum trip distance
  Tdur=mean(as.numeric(return-departure)), #trip duration # does not vary between states or days if the trip
  has more than one day
  Tdir=mean(direction), # trip direction, does not vary between states or days if the trip has more than on
  e day
  speed=mean(speed.ms), # mean bird ground speed
  dist=sum(dist), #accumulated distance covered in one day per trip per state
  abs.dir=mean(abs.dir), # mean direction between consecutive points
  windspeed=mean(windspeed),
  head.tail=mean(na.omit(headtailN)),
  SIC=mean(na.omit(SIC)), CHL=mean(na.omit(CHL)),
  N=length(ColDist),
  perc=(N/max(n_locs))*100 # percentage of fixes

body<-read.csv("C:/GiantSuperModel/body_measures.csv") # body measures

body$sexo<-ifelse(body$bill.mm>90, "M", "F")

ids<-na.omit(merge(daydf, body, by="Ring", all=F))

# variable transformation
ids$speedZ<-scale(sqrt(ids$speed))

ids$distT<-sqrt((log10(ids$dist+1))) # consecutive transformations to reduce over-dispersion

#subset
transit<-subset(ids, beh.state=="Transit")

forag<-subset(ids, beh.state!="Transit")

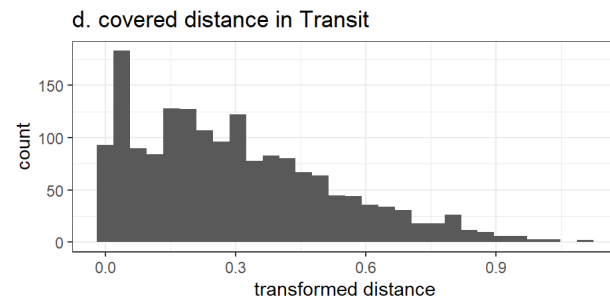
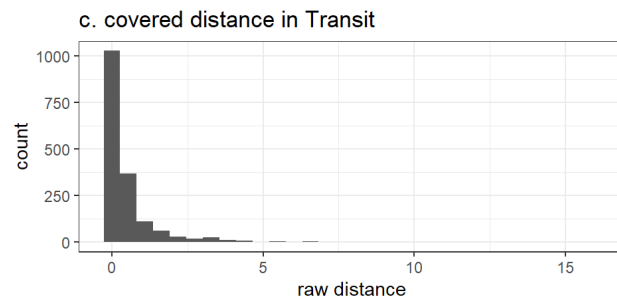
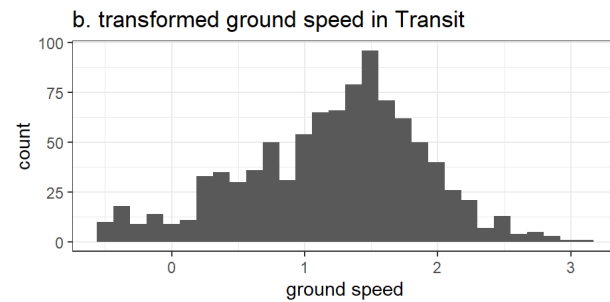
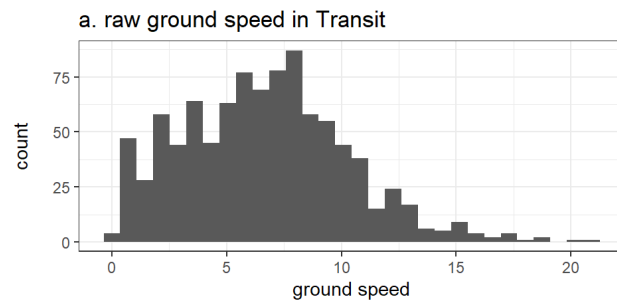
```

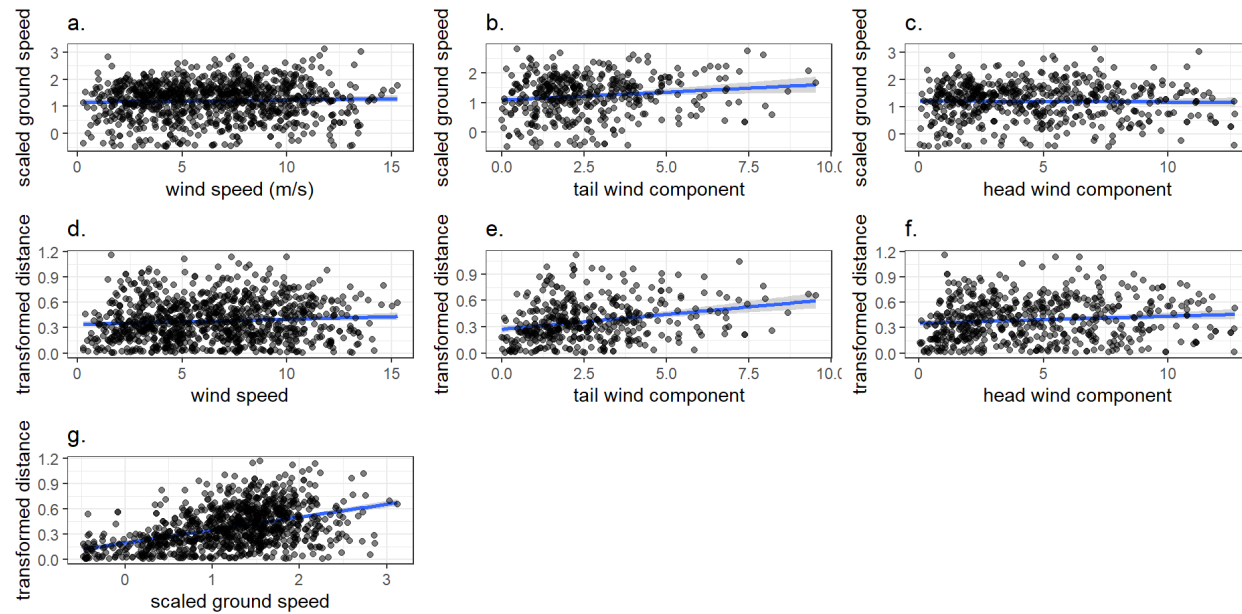
first, transform variables and do some histogram and scatterplots

```
### some diagnostics
```

```
(ggplot(subset(ids,beh.state=="Transit"),aes(speed))+geom_histogram()+  
  theme_bw()+xlab("ground speed")+  
  ggtitle(label="a. raw ground speed in Transit")+  
  
  ggplot(subset(ids,beh.state=="Transit"),aes(speedZ))+geom_histogram()+  
  theme_bw()+xlab("ground speed")+  
  ggtitle(label="b. transformed ground speed in Transit")+  
  
  ggplot(subset(ids,beh.state!="Transit"),aes(dist))+geom_histogram() +  
  theme_bw()+xlab("raw distance")+  
  ggtitle(label="c. covered distance in Transit")+  
  
  ggplot(subset(ids,beh.state!="Transit"),aes(distT))+geom_histogram() +  
  theme_bw()+xlab("transformed distance")+  
  ggtitle(label="d. covered distance in Transit"))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.  
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```





dispersion seems ok for the analysis after transformations,
however, a posteriori diagnostics will be conducted anyway

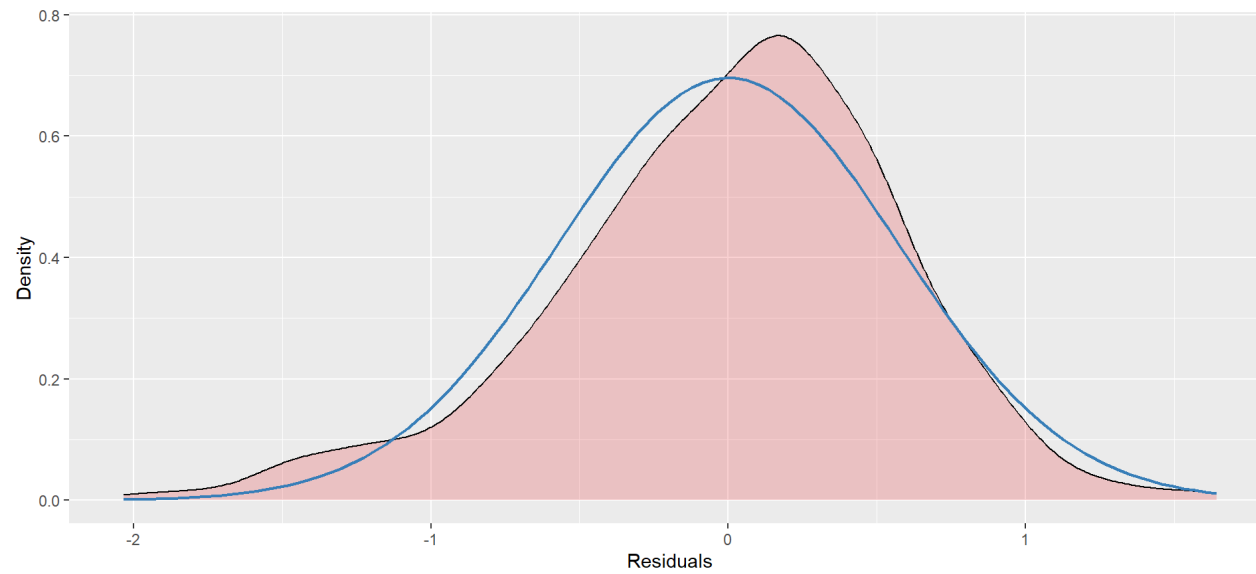
```
lmm1<-lmer(speedZ~(windspeed)+(windspeed|Ring),data=transit)
```

```
plot_model(lmm1,type="diag",grid=F)[3] # normality
```

```
## [[1]]
```

Non-normality of residuals

Distribution should look like normal curve



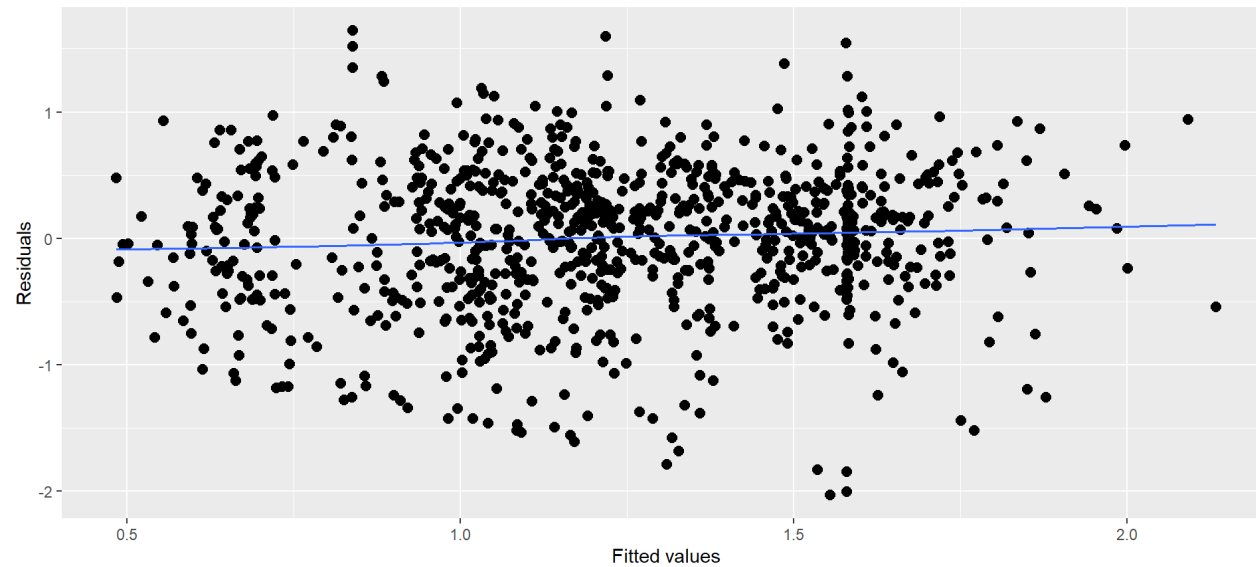
```
plot_model(lmm1,type="diag",grid=F)[4] # dispersion
```

```
## [[1]]
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

Homoscedasticity (constant variance of residuals)

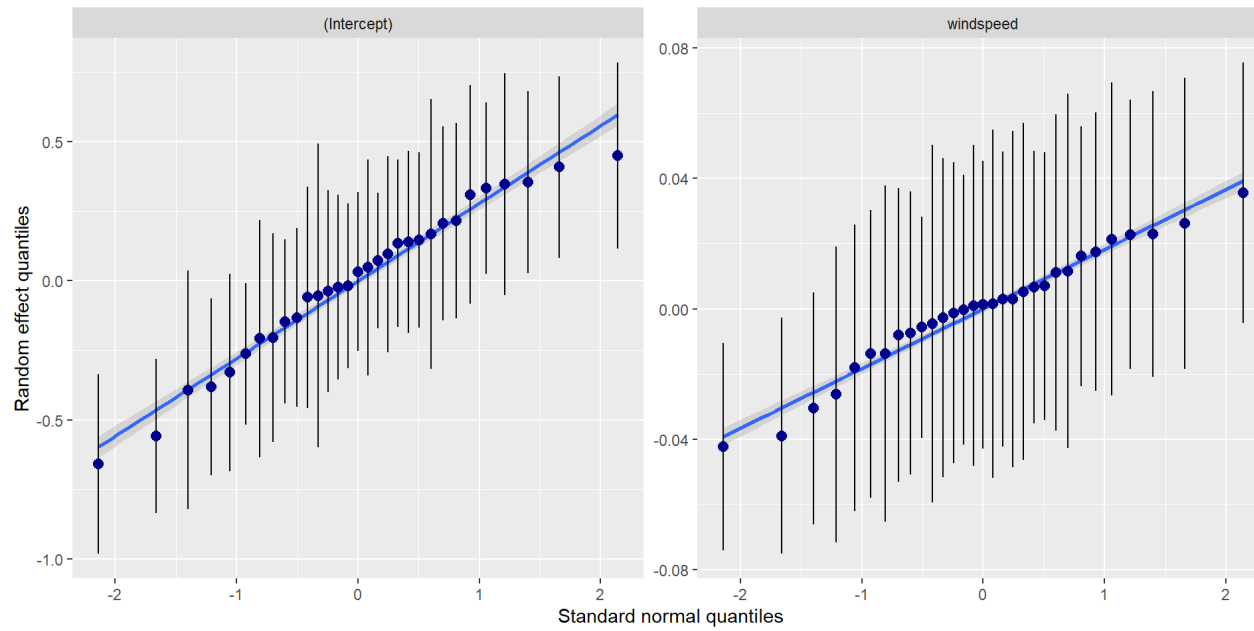
Amount and distance of points scattered above/below line is equal or randomly spread



```
plot_model(lmm1,type="diag")[2]      # random effect dispersion
```

```
## [[1]]  
## [[1]]$Ring
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
# all forward plots follows the same order
```

```
lmm2<-lmer(speedZ~head.tail+(head.tail|Ring),data=subset(transit,head.tails=="Tail"))
```

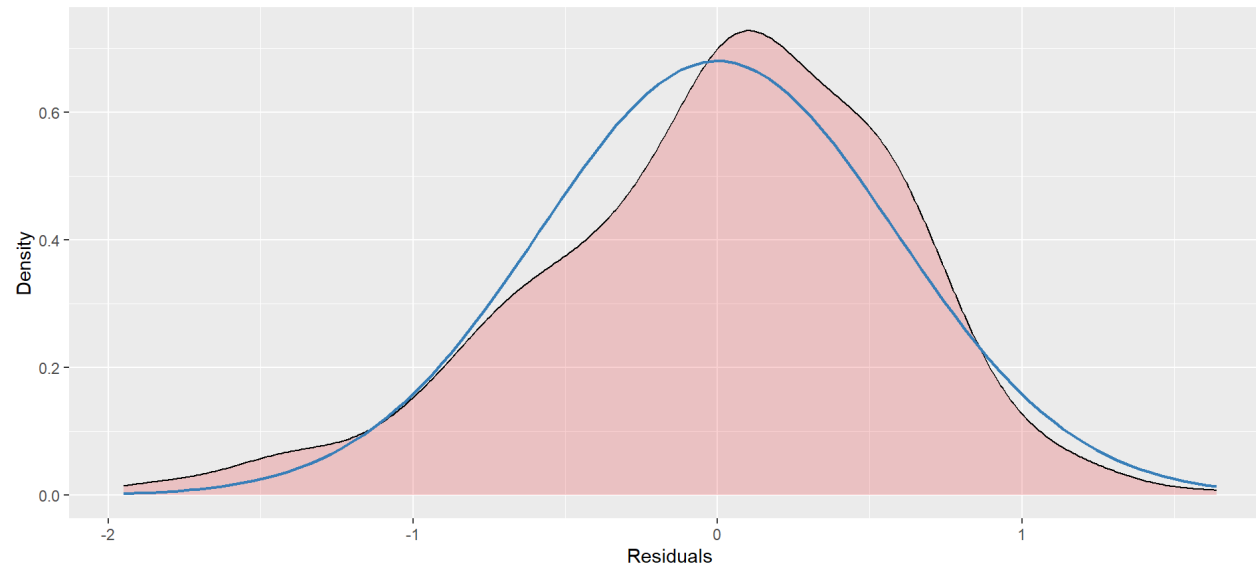
```
## boundary (singular) fit: see help('isSingular')
```

```
(plot_model(lmm2,type="diag")[3])
```

```
## [[1]]
```

Non-normality of residuals

Distribution should look like normal curve



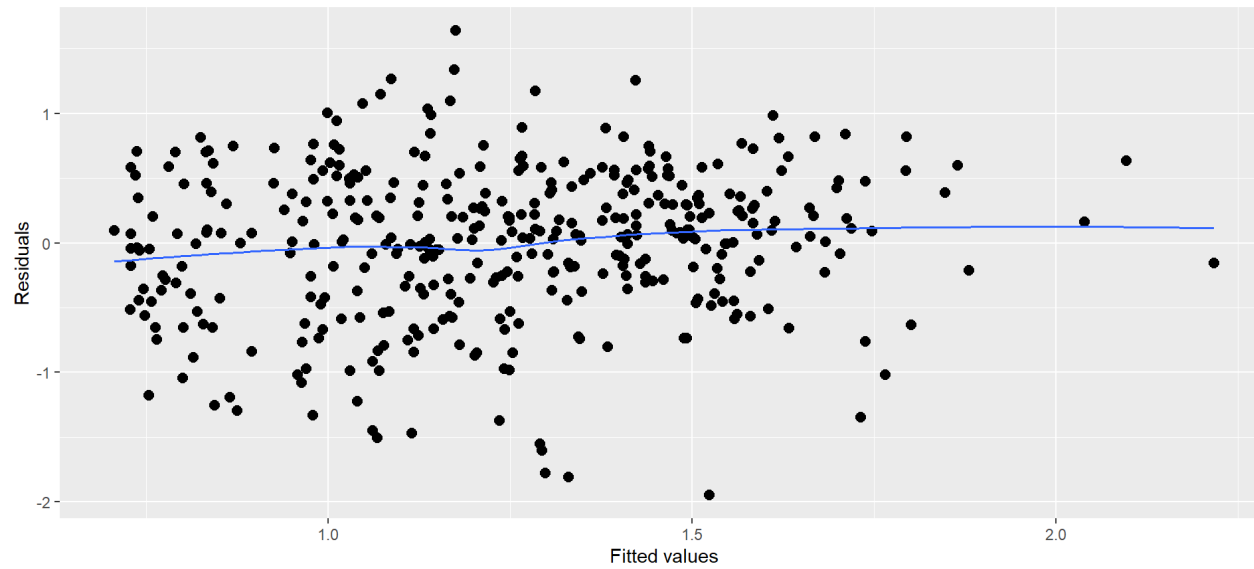
```
(plot_model(lmm2,type="diag"))[4])
```

```
## [[1]]
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

Homoscedasticity (constant variance of residuals)

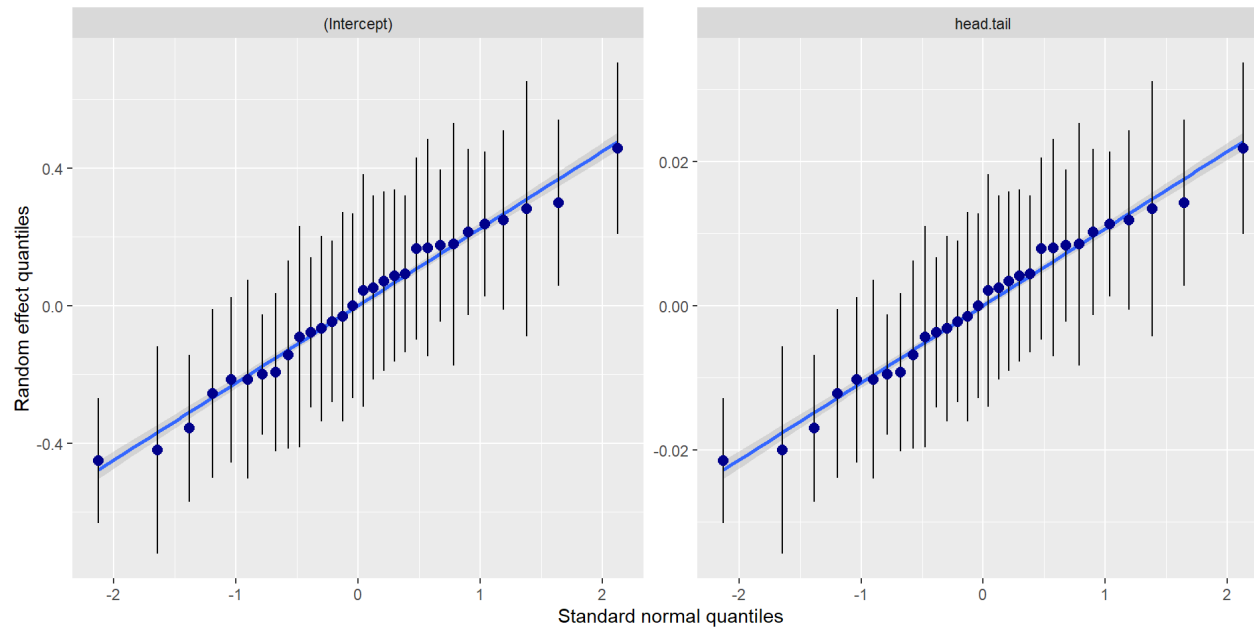
Amount and distance of points scattered above/below line is equal or randomly spread



```
(plot_model(lmm2,type="diag")[2])
```

```
## [[1]]  
## [[1]]$Ring
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



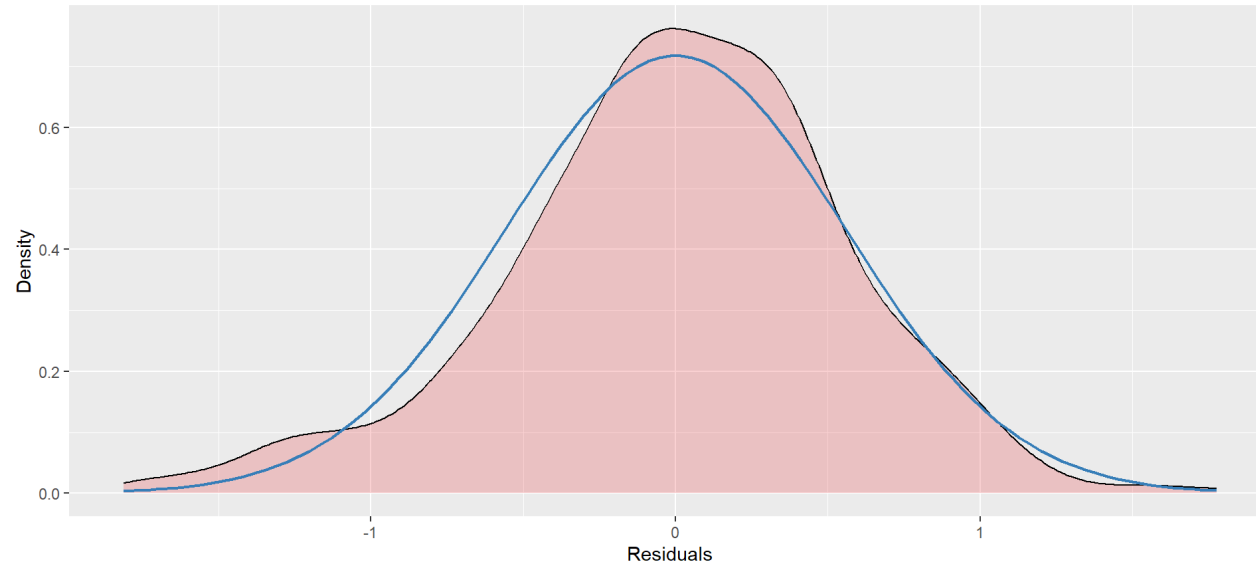
```
lmm3<-lmer(speedZ~head.tail+(head.tail|Ring),data=subset(transit,head.tails=="Head"))
```

```
(plot_model(lmm3,type="diag")[3])
```

```
## [[1]]
```

Non-normality of residuals

Distribution should look like normal curve



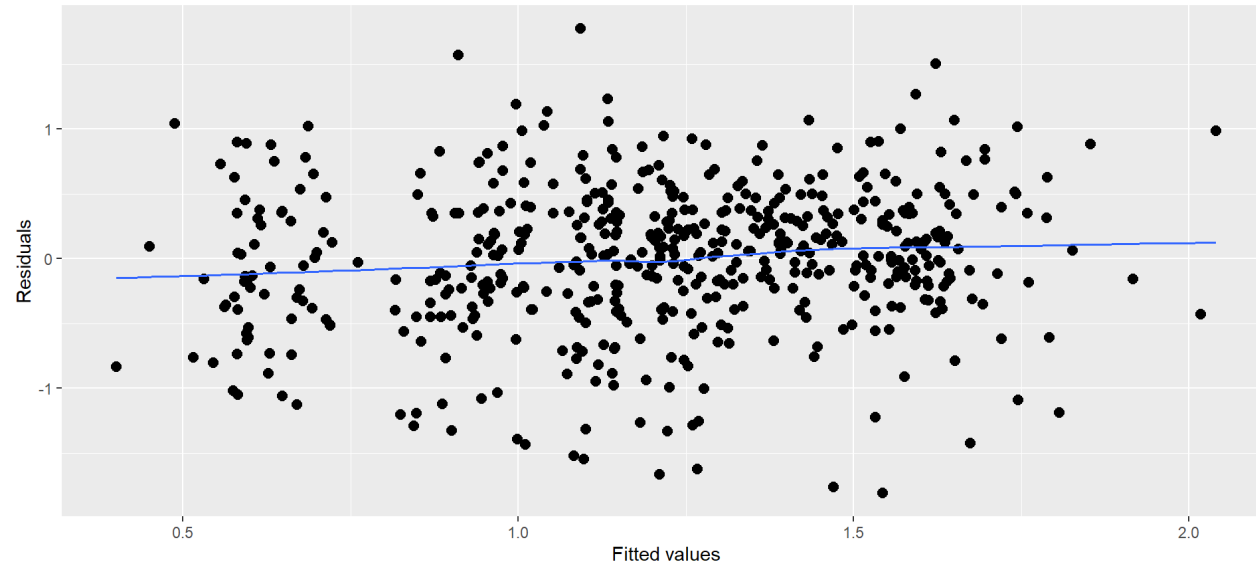
```
(plot_model(lmm3,type="diag"))[4])
```

```
## [[1]]
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

Homoscedasticity (constant variance of residuals)

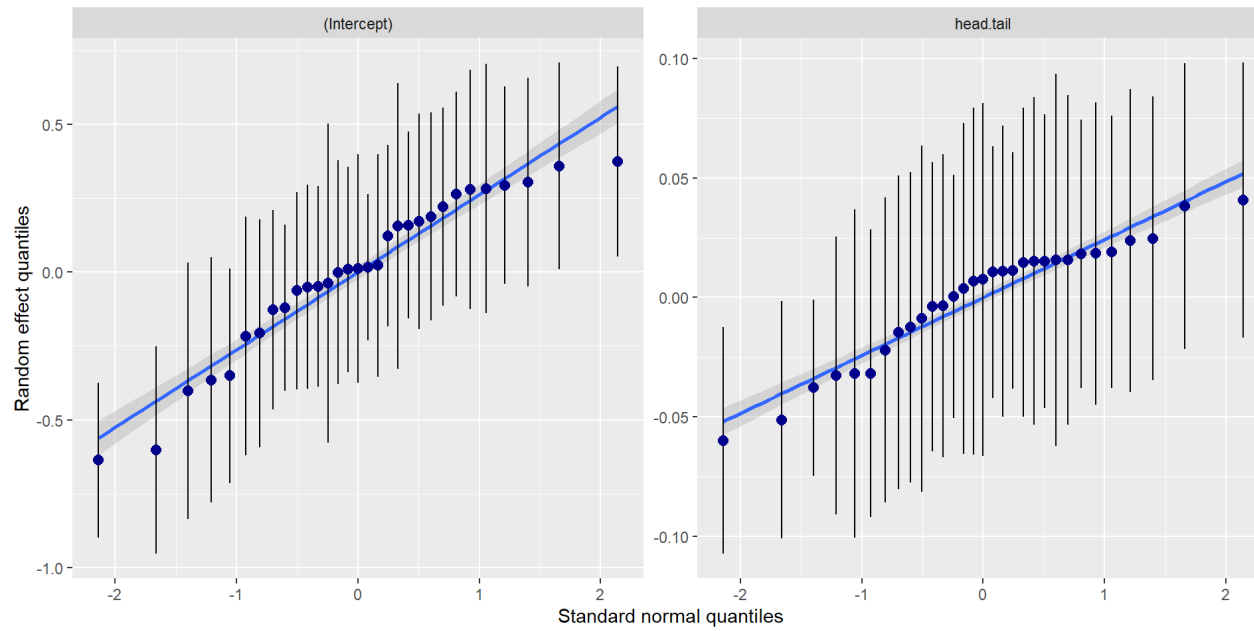
Amount and distance of points scattered above/below line is equal or randomly spread



```
(plot_model(lmm3,type="diag")[2])
```

```
## [[1]]  
## [[1]]$Ring
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



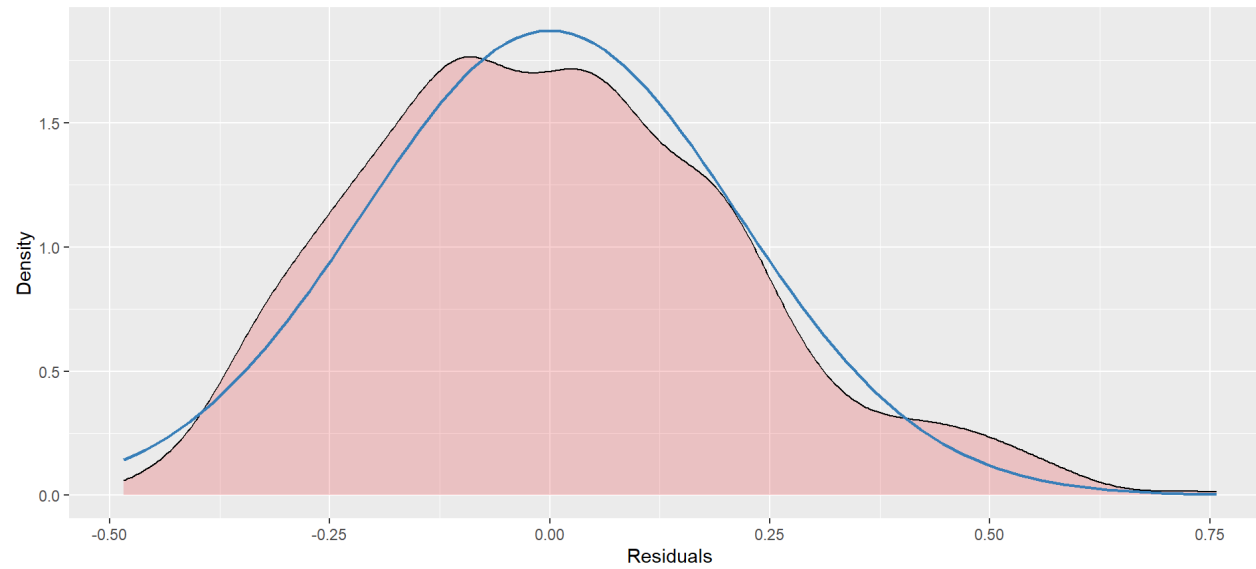
```
lmm4<-lmer(distT~windspeed+(windspeed|Ring),data=transit)
```

```
(plot_model(lmm4,type="diag")[3])
```

```
## [[1]]
```

Non-normality of residuals

Distribution should look like normal curve



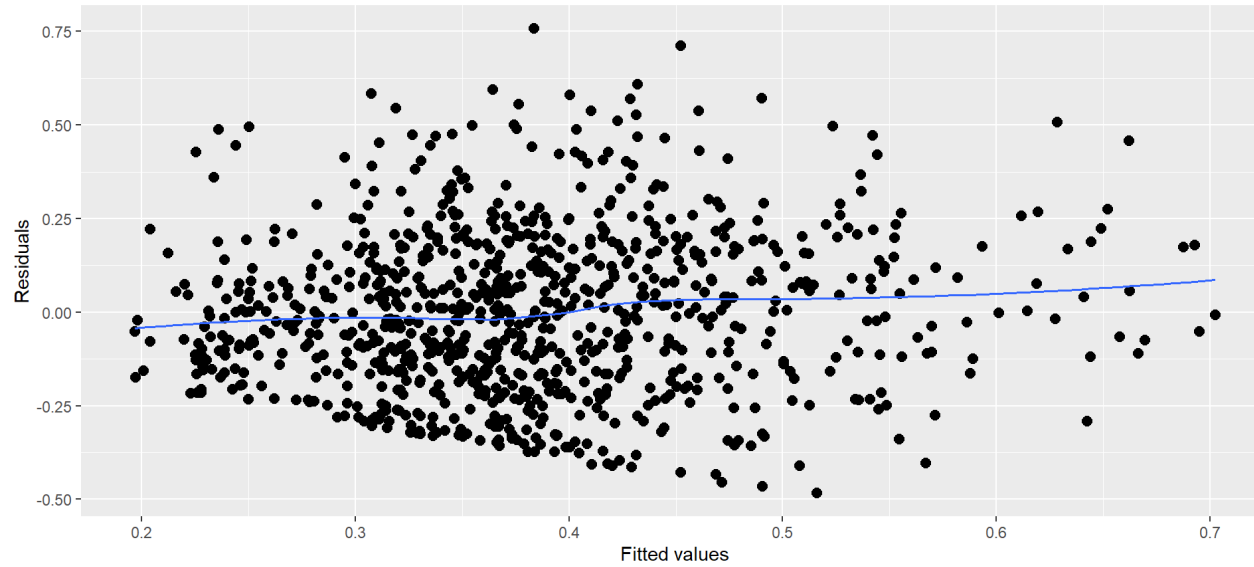
```
(plot_model(lmm4,type="diag")[4])
```

```
## [[1]]
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

Homoscedasticity (constant variance of residuals)

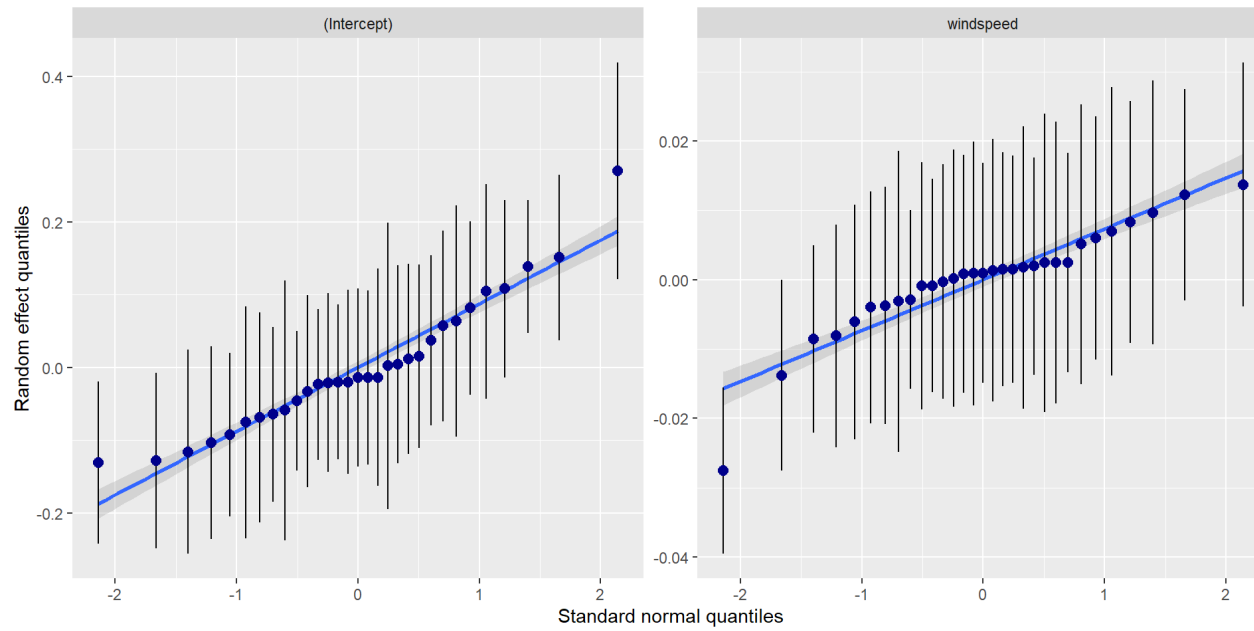
Amount and distance of points scattered above/below line is equal or randomly spread



```
(plot_model(lmm4,type="diag")[2])
```

```
## [[1]]  
## [[1]]$Ring
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



```
lmm5<-lmer(distT~head.tail+(head.tail|Ring),data=subset(transit,head.tails=="Tail"))
```

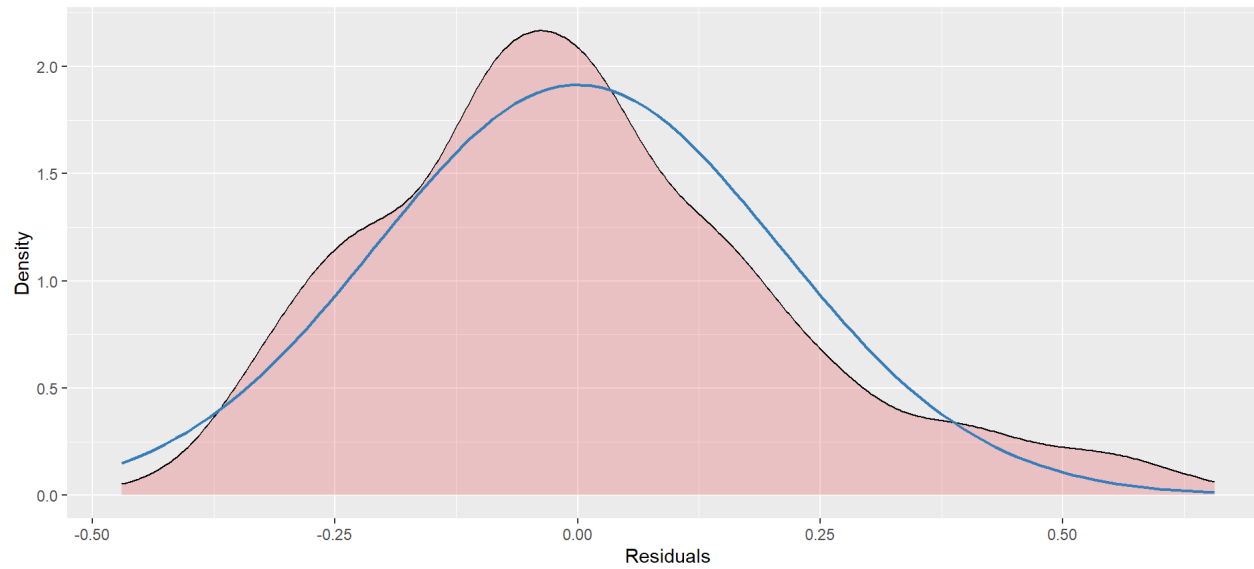
```
## boundary (singular) fit: see help('isSingular')
```

```
(plot_model(lmm5,type="diag")[3])
```

```
## [[1]]
```

Non-normality of residuals

Distribution should look like normal curve



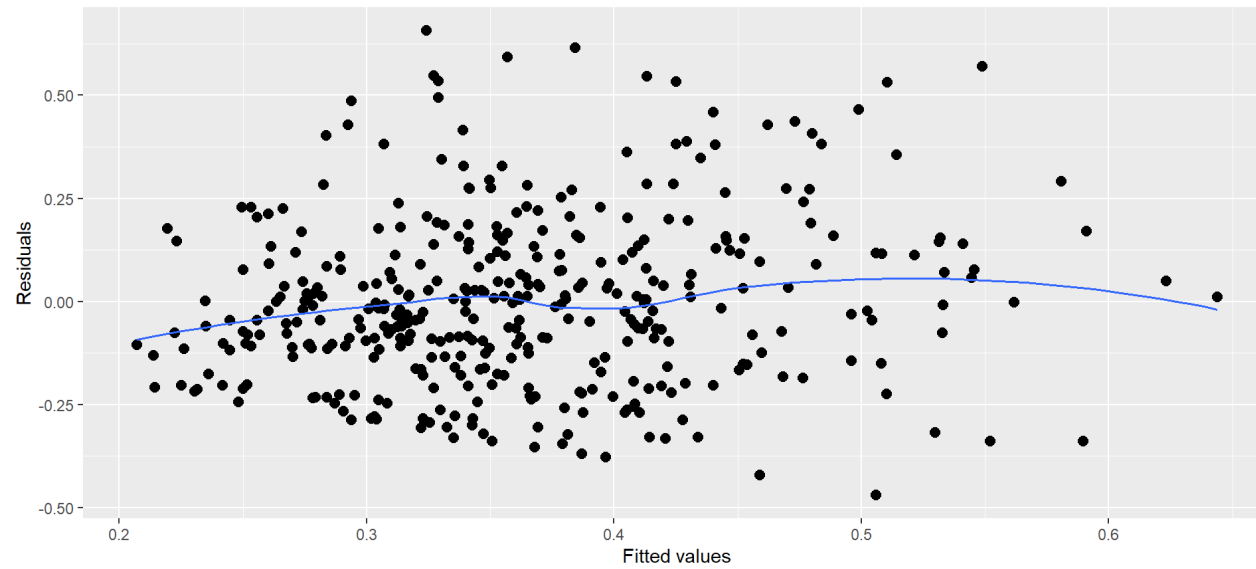
```
(plot_model(lmm5,type="diag"))[4])
```

```
## [[1]]
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

Homoscedasticity (constant variance of residuals)

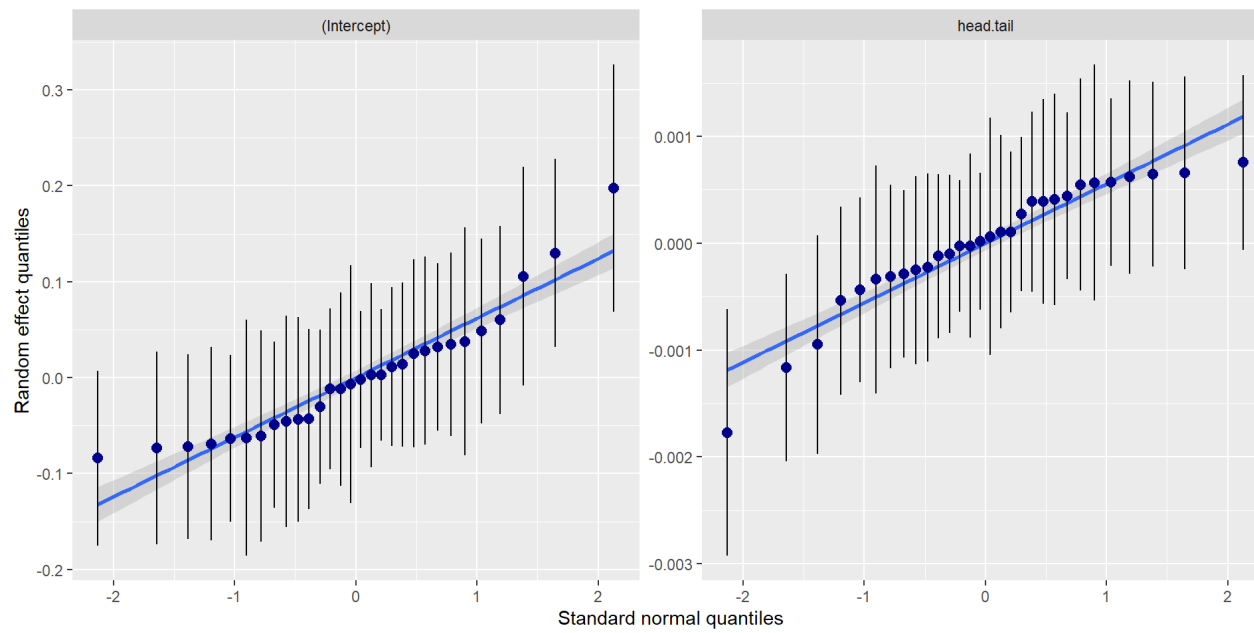
Amount and distance of points scattered above/below line is equal or randomly spread



```
(plot_model(lmm5,type="diag")[2])
```

```
## [[1]]  
## [[1]]$Ring
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



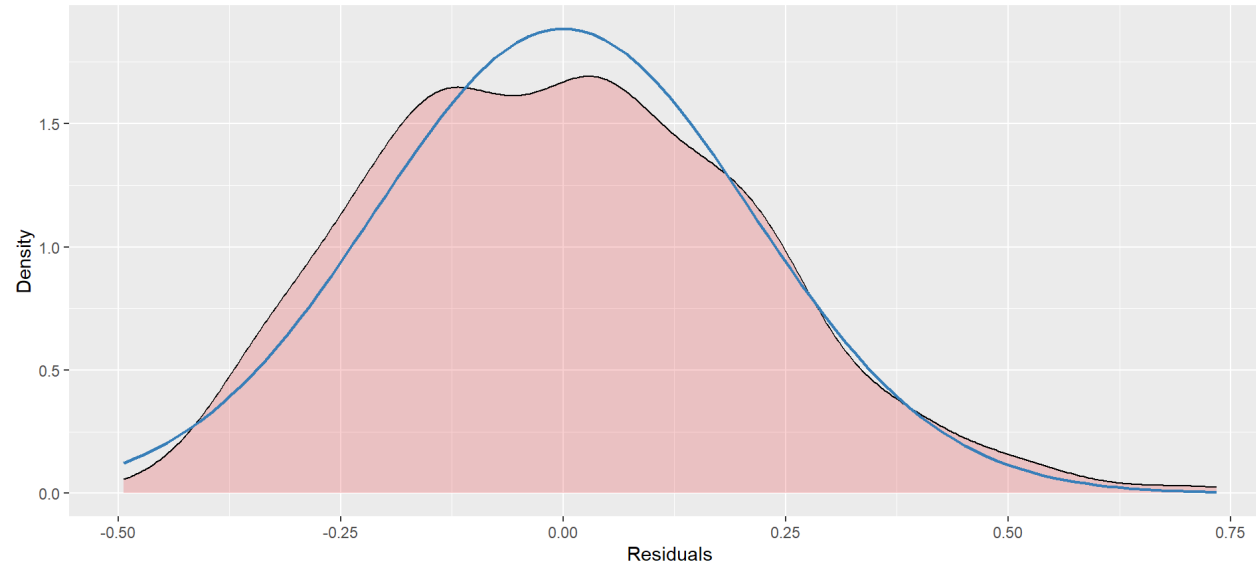
```
lmm6<-lmer(distT~head.tail+(head.tail|Ring),data=subset(transit,head.tails=="Head"))
```

```
(plot_model(lmm6,type="diag")[3])
```

```
## [[1]]
```

Non-normality of residuals

Distribution should look like normal curve



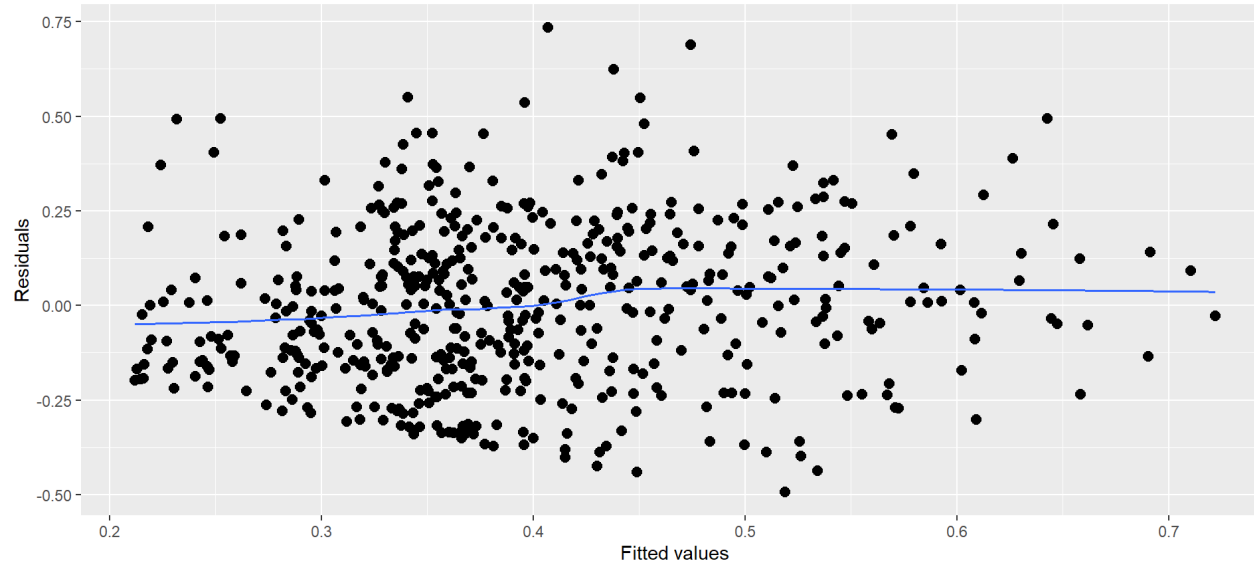
```
(plot_model(lmm6,type="diag"))[4])
```

```
## [[1]]
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

Homoscedasticity (constant variance of residuals)

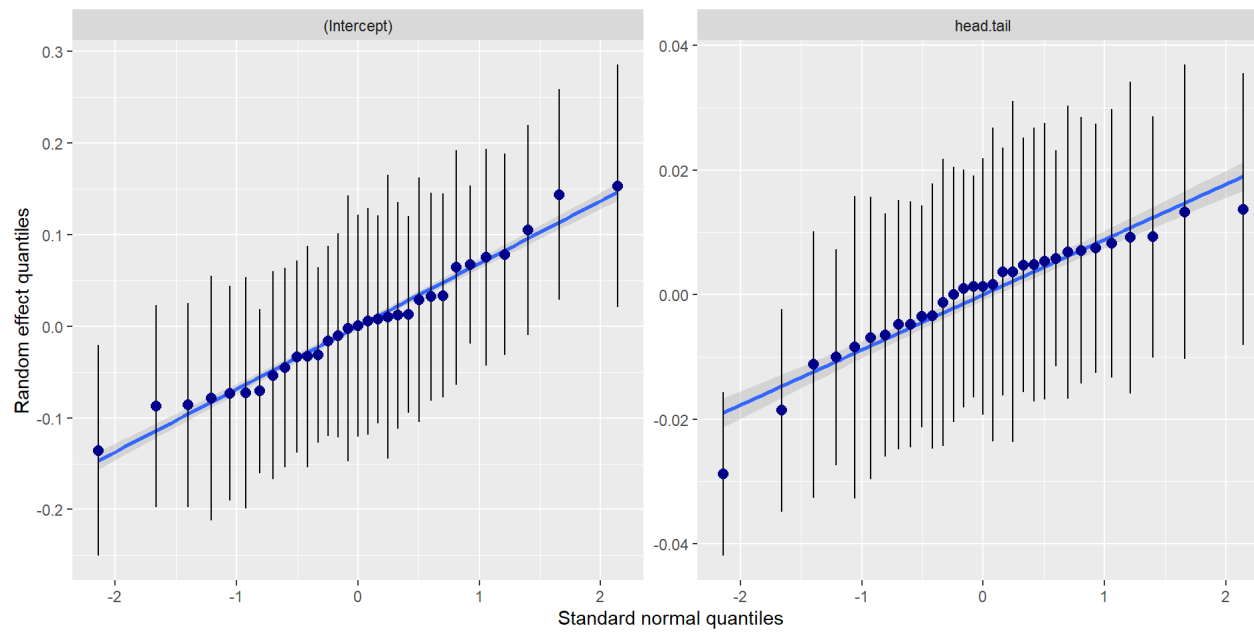
Amount and distance of points scattered above/below line is equal or randomly spread



```
(plot_model(lmm6,type="diag")[2])
```

```
## [[1]]  
## [[1]]$Ring
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



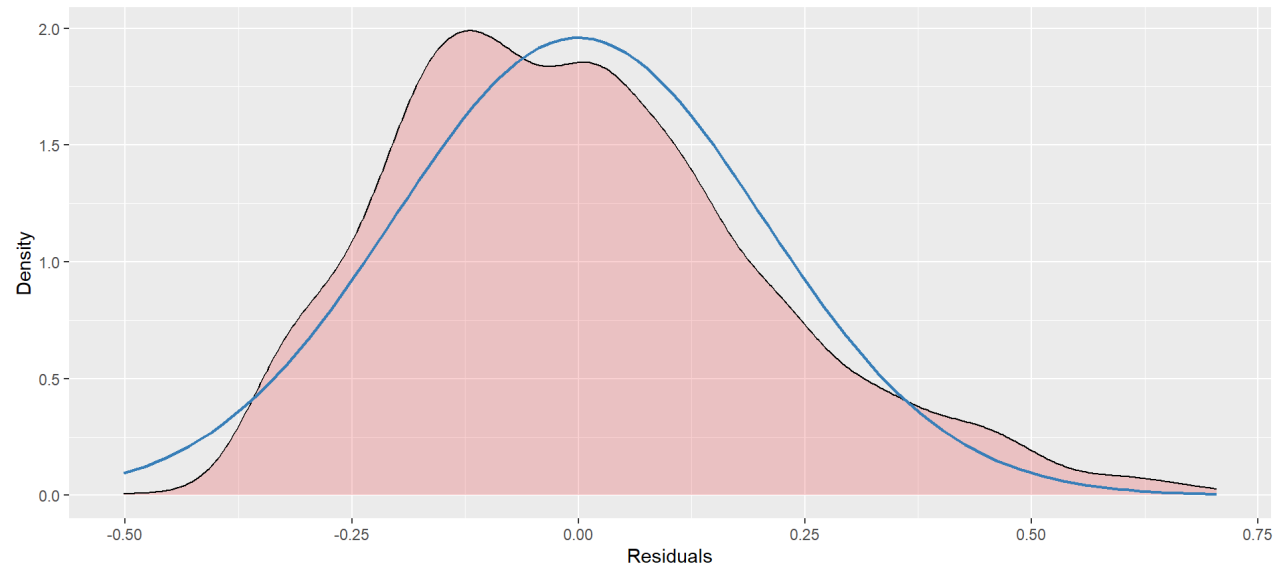
```
lmm8<-lmer(distT~CHL+(CHL|Ring),data=forag)
```

```
(plot_model(lmm8,type="diag")[3])
```

```
## [[1]]
```

Non-normality of residuals

Distribution should look like normal curve



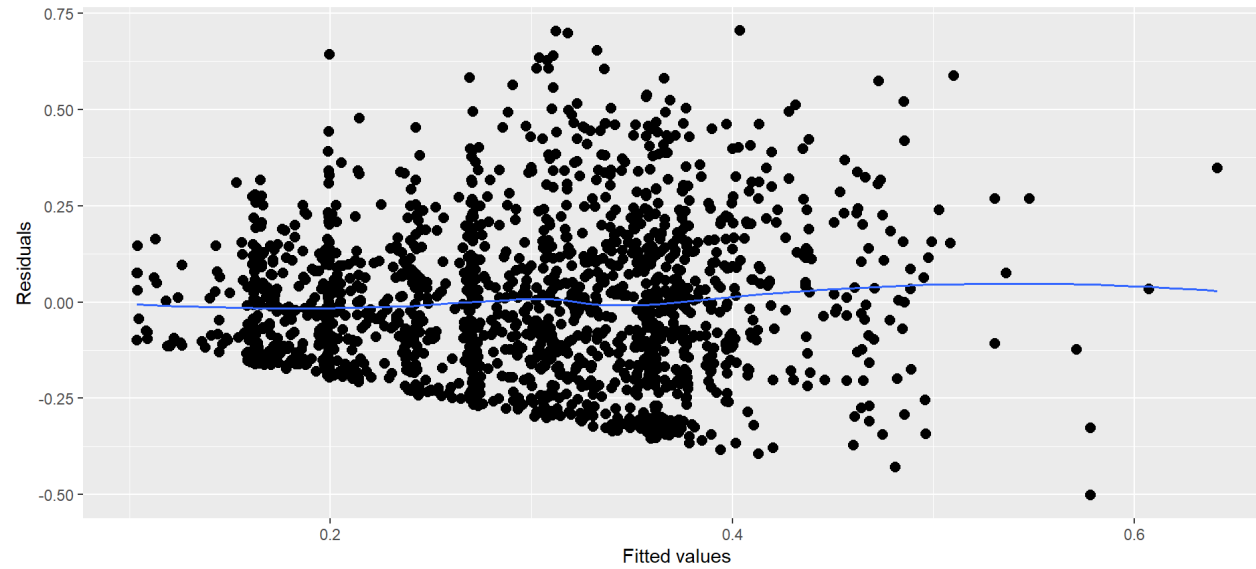
```
(plot_model(lmm8,type="diag")[4]) # some dispersion
```

```
## [[1]]
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

Homoscedasticity (constant variance of residuals)

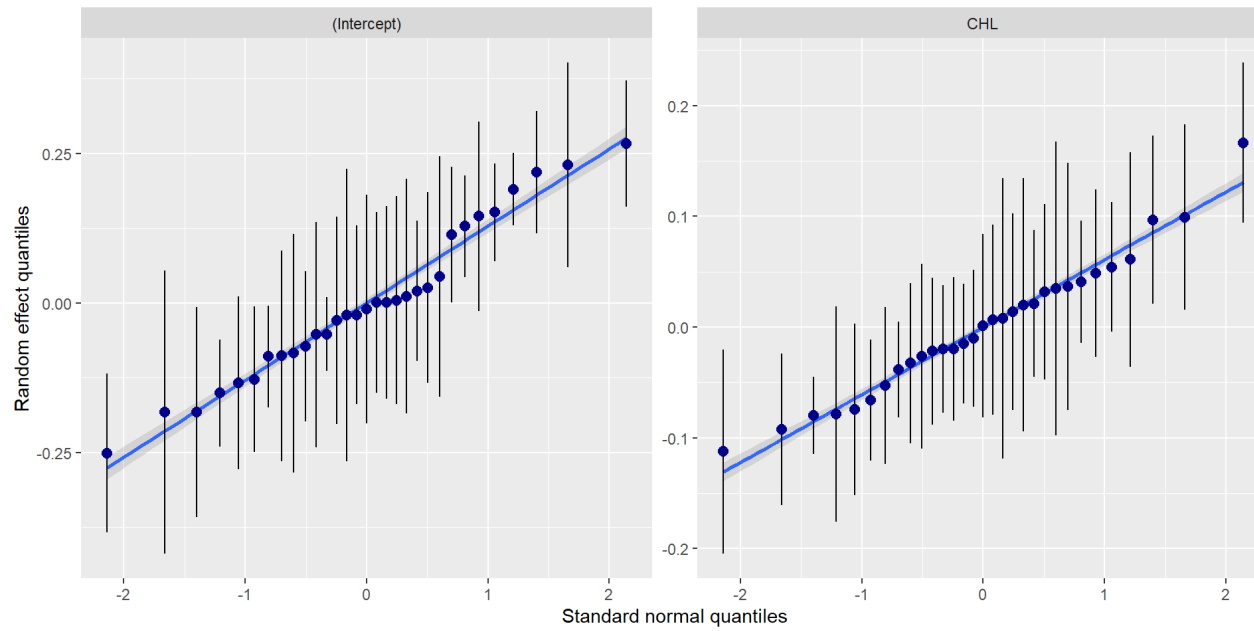
Amount and distance of points scattered above/below line is equal or randomly spread



```
(plot_model(lmm8,type="diag")[2])
```

```
## [[1]]  
## [[1]]$Ring
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



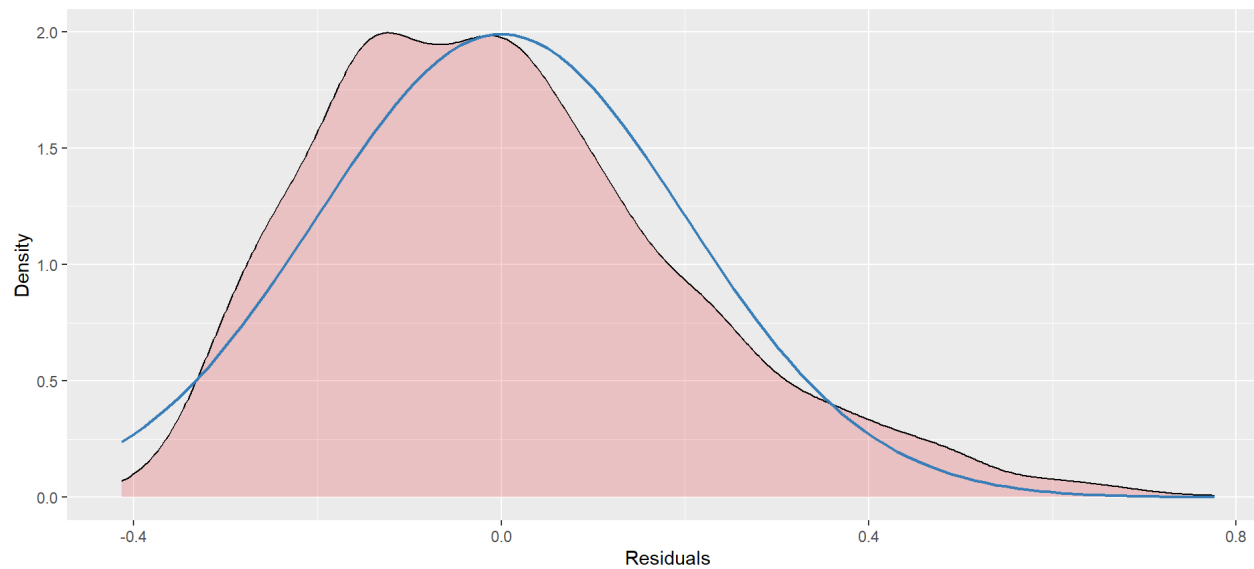
```
lmm9<-lmer(distT~SIC+(SIC|Ring),data=forag)
```

```
(plot_model(lmm9,type="diag")[3])
```

```
## [[1]]
```

Non-normality of residuals

Distribution should look like normal curve



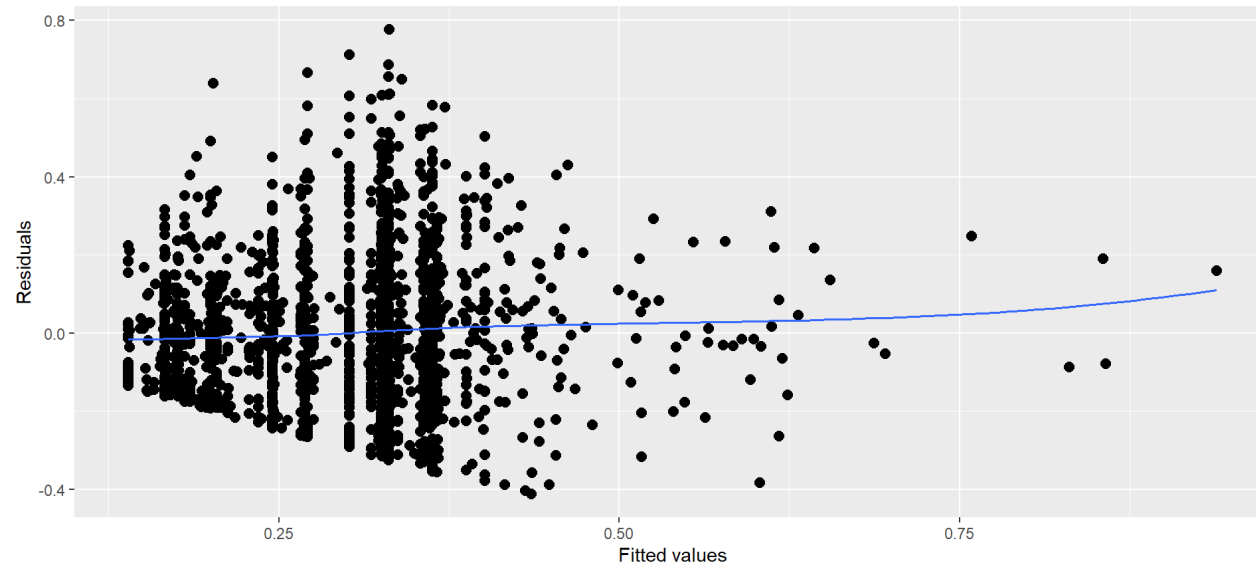
```
(plot_model(lmm9,type="diag")[4]) # some dispersion
```

```
## [[1]]
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

Homoscedasticity (constant variance of residuals)

Amount and distance of points scattered above/below line is equal or randomly spread



```
(plot_model(lmm9,type="diag")[2])
```

```
## [[1]]
```

```
## [[1]]$Ring
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

