

```
# Study title: Injury and Illness in Non-Football Collegiate Athletics: a Single University, 8-Year Retrospective Analysis
# Author: Khang Tong
```

```
library(readxl)
library(ggplot2)
library(dplyr)
library(geepack)
library(emmeans)
```

```
options(digit = 3, stringsAsFactors = FALSE, width = 100)
```

```
pvalFormat <- function(p.values, method = 'none', replace = FALSE, math = TRUE){
  ## Formats p-values for reports, can report adjusted pvalues
  ## Inputs:
  ##   - p.value: numeric p-value
  ##   - method: pvalue adjustment, passed to p.adjust.methods
  ##   - replace: if TRUE, replaces p-values with their adjusted value
  ## Outputs:
  ##   - out: formatted p-value

  p.values <- suppressWarnings(as.numeric(p.values))
  out <- rep(NA, length(p.values))
  sig <- p.adjust(p.values, method)
  if(replace) p.values <- sig

  for(i in 1:length(p.values)){
    if(is.na(p.values[i])){out[i] <- NA}else{
      if(p.values[i] >= .001){
        out[i] <- paste('$', formatC(p.values[i], format = 'f', digits = 3), '$', sep = '')
      }

      if(p.values[i] < .001){
        out[i] <- '< $0.001$'
      }

      if(sig[i] > 0.01 & sig[i] <= 0.05){
        out[i] <- paste(out[i], '*', sep = '')
      }

      if(sig[i] > 0.001 & sig[i] <= 0.01) {
        out[i] <- paste(out[i], '**', sep = '')
      }

      if(sig[i] <= 0.001){
        out[i] <- paste(out[i], '***', sep = '')
      }
    }
  }

  out[is.na(out)] <- '-'
  return(out)
}
```

```
# Import injury data
df.injury <- read_excel("Injury_Data_Full.xlsx",
  col_types = c("numeric", "date", "text",
    "text", "text", "text", "text", "text",
    "text", "text", "text", "text", "text")) %>%
  janitor::clean_names(., "lower_camel")
```

```
#Import total team sizes
df.totals <- read_excel("TotalCount.xlsx") %>%
  data.frame(., check.names = FALSE)
```

```
# Import training hours
df.m.baseball <- read_excel("M_Baseball_Hrs.xlsx")
df.m.basketball <- read_excel("M_Basketball_Hrs.xlsx")
df.m.crew <- read_excel("M_Crew_Hrs.xlsx")
df.m.cross.country <- read_excel("M_Cross_Country_Hrs.xlsx")
df.m.fencing <- read_excel("M_Fencing_Hrs.xlsx")
df.m.golf <- read_excel("M_Golf_Hrs.xlsx")
df.m.soccer <- read_excel("M_Soccer_Hrs.xlsx")
df.m.swimming <- read_excel("M_Swimming_Hrs.xlsx")
df.m.tennis <- read_excel("M_Tennis_Hrs.xlsx")
df.m.volleyball <- read_excel("M_Volleyball_Hrs.xlsx")
df.m.water.polo <- read_excel("M_Water_Polo_Hrs.xlsx")
```

```
df.f.baseball <- read_excel("W_Baseball_Hrs.xlsx")
df.f.basketball <- read_excel("W_Basketball_Hrs.xlsx")
df.f.crew <- read_excel("W_Crew_Hrs.xlsx")
df.f.cross.country <- read_excel("W_Cross_Country_Hrs.xlsx")
df.f.fencing <- read_excel("W_Fencing_Hrs.xlsx")
df.f.soccer <- read_excel("W_Soccer_Hrs.xlsx")
df.f.swimming <- read_excel("W_Swimming_Hrs.xlsx")
df.f.tennis <- read_excel("W_Tennis_Hrs.xlsx")
df.f.volleyball <- read_excel("W_Volleyball_Hrs.xlsx")
df.f.water.polo <- read_excel("W_Water_Polo_Hrs.xlsx")
```

```
df.track = read_excel("hours.xlsx",
  sheet = "sprint_2022_hours")
colnames(df.track)[1] = "Week"
```

```
df.totals = df.totals %>%
  dplyr::select(-"2020-2021", -"2021-2022")
```

```
# Assign dual athlete injuries to the sport with typical season closest to injury date
dual.ids = unique(df.injury[grepl(";", df.injury$sport),"id"]) %>%
  data.frame() # 46 unique athletes
```

```

dual.ids = dual.ids[1]

# We don't have exact season dates so this was done manually. See dual-athlete-changes.txt for full documentation
df.injury[df.injury$id == dual.ids[1], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Basketball", "Basketball", "Track and Field")
df.injury[df.injury$id == dual.ids[2], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Crew", "Crew", "Cross Country", "Crew", "Crew")
df.injury[df.injury$id == dual.ids[3], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Volleyball", "Crew", "Crew", "Volleyball", "Volleyball")
df.injury[df.injury$id == dual.ids[4], c('id', 'sport', 'gender', 'injuryDate')]$sport[1] = "Water Polo"
df.injury[df.injury$id == dual.ids[5], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Track and Field", "Track and Field", "Softball")
df.injury[df.injury$id == dual.ids[6], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Cross Country", "Swimming/Diving", "Swimming/Diving")
df.injury[df.injury$id == dual.ids[7], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Crew", "Crew", "Track and Field")
df.injury[df.injury$id == dual.ids[8], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Cross Country", "Cross Country")
df.injury[df.injury$id == dual.ids[9], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Cross Country", "Track and Field", "Track and Field")
df.injury[df.injury$id == dual.ids[10], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Cross Country", "Cross Country", "Cross Country")
df.injury[df.injury$id == dual.ids[11], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Cross Country", "Cross Country", "Cross Country")
df.injury[df.injury$id == dual.ids[12], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Cross Country", "Track and Field", "Cross Country")
df.injury[df.injury$id == dual.ids[13], c('id', 'sport', 'gender', 'injuryDate')]$sport = rep("Cross Country", 4)
df.injury[df.injury$id == dual.ids[14], c('id', 'sport', 'gender', 'injuryDate')]$sport = rep("Cross Country", 2)
df.injury[df.injury$id == dual.ids[15], c('id', 'sport', 'gender', 'injuryDate')]$sport = rep("Cross Country", 4)
df.injury[df.injury$id == dual.ids[16], c('id', 'sport', 'gender', 'injuryDate')]$sport = "Track and Field"
df.injury[df.injury$id == dual.ids[17], c('id', 'sport', 'gender', 'injuryDate')]$sport = "Cross Country"
df.injury[df.injury$id == dual.ids[18], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Cross Country", "Track and Field", "Track and Field")
df.injury[df.injury$id == dual.ids[19], c('id', 'sport', 'gender', 'injuryDate')]$sport = "Cross Country"
df.injury[df.injury$id == dual.ids[20], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Cross Country", "Track and Field")
df.injury[df.injury$id == dual.ids[21], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Track and Field", "Cross Country", "Cross Country")
df.injury[df.injury$id == dual.ids[22], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Cross Country", "Cross Country", "Track and Field")
df.injury[df.injury$id == dual.ids[23], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Soccer", rep("Track and Field", 4))
df.injury[df.injury$id == dual.ids[24], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Track and Field", "Volleyball", "Volleyball")
df.injury[df.injury$id == dual.ids[25], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Volleyball", rep("Water Polo", 4))
df.injury[df.injury$id == dual.ids[26], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Swimming/Diving", "Swimming/Diving", "Water Polo")
df.injury[df.injury$id == dual.ids[27], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Soccer", "Soccer")
df.injury[df.injury$id == dual.ids[28], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Cross Country", "Track and Field", "Cross Country")
df.injury[df.injury$id == dual.ids[29], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Cross Country", "Cross Country", "Track and Field")
df.injury[df.injury$id == dual.ids[30], c('id', 'sport', 'gender', 'injuryDate')]$sport = rep("Cross Country", 3)
df.injury[df.injury$id == dual.ids[31], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Track and Field", "Cross Country", "Track and Field")
df.injury[df.injury$id == dual.ids[32], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Cross Country", "Track and Field", "Cross Country")
df.injury[df.injury$id == dual.ids[33], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Cross Country", "Track and Field")
df.injury[df.injury$id == dual.ids[34], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Cross Country", "Cross Country", "Track and Field")
df.injury[df.injury$id == dual.ids[35], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Cross Country", "Track and Field", "Cross Country")
df.injury[df.injury$id == dual.ids[36], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Cross Country", "Cross Country", "Cross Country")
df.injury[df.injury$id == dual.ids[37], c('id', 'sport', 'gender', 'injuryDate')]$sport = c(rep("Cross Country", 3), "Track and Field")
df.injury[df.injury$id == dual.ids[38], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Cross Country", "Cross Country")
df.injury[df.injury$id == dual.ids[39], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Cross Country", "Cross Country")
df.injury[df.injury$id == dual.ids[40], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Cross Country", "Track and Field", "Cross Country")
df.injury[df.injury$id == dual.ids[41], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Track and Field", "Track and Field", "Cross Country")
df.injury[df.injury$id == dual.ids[42], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Cross Country", "Cross Country")
df.injury[df.injury$id == dual.ids[43], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Cross Country", "Track and Field", "Track and Field")
df.injury[df.injury$id == dual.ids[44], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Track and Field", "Track and Field")
df.injury[df.injury$id == dual.ids[45], c('id', 'sport', 'gender', 'injuryDate')]$sport = "Track and Field"
df.injury[df.injury$id == dual.ids[46], c('id', 'sport', 'gender', 'injuryDate')]$sport = c("Soccer", "Soccer")

# Rename softball (female) to baseball for comparison with male baseball
df.injury$sport = ifelse(df.injury$sport=="Softball", "Baseball", df.injury$sport)

# Combine rowing with crew and call it crew
df.injury$sport = ifelse(df.injury$sport=="Rowing", "Crew", df.injury$sport)

# Keep only injuries from 2012-2013 to 2019-2020 season
df.injury = df.injury %>%
  filter(!academicYear %in% c("2011-2012", "2020-2021", "2021-2022", "2022-2023"))

# Calculate total number of athletes by sport and gender
df.totals$sum = rowSums(df.totals[,2:9])
df.totals$gender = substring(df.totals$`Player Count`, 1,1)
df.totals$sport = substring(df.totals$`Player Count`, 3)

# Number of athletes by year in long format
df.totals.long <- df.totals %>%
  tidyr::pivot_longer(
    cols = `2012-2013`:`2019-2020`,
    names_to = "academicYear",
    values_to = "n.athletes"
  )

# Number of injuries by sport and gender per season
df.injury.count.by.year = df.injury %>%
  group_by(sport, gender, academicYear) %>%
  summarise(n.injuries = n())

df.clinical.incidence.by.year = df.totals.long %>%
  dplyr::select(sport, gender, academicYear, n.athletes) %>%
  merge(., df.injury.count.by.year) %>%
  group_by(n.injuries) %>%
  mutate(incidence = n.injuries/n.athletes)

### Table 1 ###
df.clinical.incidence.by.year.overall.mean = df.clinical.incidence.by.year %>%
  group_by(sport, gender) %>%
  summarise(incidence.mean = mean(incidence), sd = sd(incidence), n.injuries = sum(n.injuries), n.athletes = sum(n.athletes))

tot.m = sum(subset(df.totals, gender=="M")$sum)
tot.f = sum(subset(df.totals, gender=="F")$sum)

df.injury.count.body.by.year = df.injury %>%
  group_by(sport, gender, injuryClassification, academicYear) %>%
  mutate(injuryClassification = factor(injuryClassification), sport = factor(sport), gender = factor(gender), academicYear = factor(academicYear))
  summarise(n.injuries = n())

```

```

df.clinical.incidence.body.by.year = df.totals.long %>%
  dplyr::select(sport, gender, academicYear, n.athletes) %>%
  mutate(academicYear = factor(academicYear), gender = factor(gender), sport = factor(sport)) %>%
  merge(., df.injury.count.body.by.year %>% ungroup()) %>% tidyr::complete(sport, gender, academicYear, injuryClassification), all=TRUE) %>%
  filter(!(sport == "Golf" & gender == "F")) %>% # UCSD does not have women's golf
  group_by(n.injuries) %>%
  mutate(n.injuries = ifelse(is.na(n.injuries), 0, n.injuries),
         incidence = n.injuries/n.athletes)

### Table 2 ###
df.clinical.incidence.body.by.year.overall.mean = df.clinical.incidence.body.by.year %>%
  group_by(sport, gender, injuryClassification) %>%
  summarise(incidence. = mean(incidence), sd = sd(incidence))

#### GEE Models ####

# Setting up the data for analysis

# Since our dataset contains data only when an injury was recorded, we inserted
# 0 injury count data so that the number of athletes in the dataset aligned with
# the known team sizes for each sport. Since we do not have information on
# athletes with no injuries between seasons (how many non-injured athletes are
# the same person across multiple seasons), all inserted 0 injury count data
# were assumed to be independent

# Number of zeros to create = total number of athletes per sport by gender per
# season - number of athletes with injuries per sport by gender per season

df.tot.ath = df.totals %>%
  tidyr::pivot_longer(
    cols = `2012-2013`:`2019-2020`,
    names_to = "season",
    values_to = "player.count"
  ) %>%
  arrange(sport) %>%
  arrange(desc(gender))

df.tot.inj = df.injury %>%
  mutate(sport = factor(sport),
         gender = factor(gender),
         academicYear = factor(academicYear)) %>%
  group_by(id, academicYear) %>%
  dplyr::distinct(id, academicYear, .keep_all = TRUE) %>%
  ungroup() %>%
  group_by(sport, gender, academicYear, .drop=F) %>%
  summarise(n.inj = n())

df.tot.inj = df.tot.inj[!grepl(";{1}", df.tot.inj$sport), ] %>%
  filter(!(sport == "Golf" & gender == "F")) %>%
  arrange(sport) %>%
  arrange(desc(gender))

sum(df.tot.ath$sport != df.tot.inj$sport) #Check

# There were 4 sport seasons where reported number of injured athletes exceeded the
# reported team size (by 1 or 2). For these cases, we did left the data as-is
# and did not add any zeroes.
df.tot = cbind(df.tot.ath, df.tot.inj[,c("n.inj")]) %>%
  mutate(diff = player.count - n.inj ,
         diff.analysis = ifelse((player.count - n.inj) < 0, 0, player.count - n.inj))

# Create necessary number of zeroes
add.zeros = function(data, sport, gender, year, n){
  rbind(data, matrix(rep(c(sport, gender, year, 0), n), byrow = T, nrow = n))
}
df.zeros = data.frame()
for (i in 1:dim(df.tot)[1]){
  df.zeros = add.zeros(df.zeros, df.tot$sport[i], df.tot$gender[i], df.tot$season[i], df.tot$diff.analysis[i])
}
z.id = 99999:(99999+dim(df.zeros)[1]-1) # Assign arbitrary ID to these added zeroes
df.zeros = cbind(z.id, df.zeros)
colnames(df.zeros) = c("id", "sport", "gender", "academicYear", "n.injuries")
df.zeros$id = as.numeric(df.zeros$id)
df.zeros$n.injuries = as.numeric(df.zeros$n.injuries)
sum(df.tot$diff.analysis)
dim(df.zeros) #Check

# Count of injury per subject
df.fit = df.injury %>%
  group_by(id, sport, gender, academicYear) %>%
  summarise(n.injuries = n()) %>%
  rbind(df.zeros) %>%
  ungroup() %>%
  mutate(gender =relevel(factor(gender), ref="M"))

## Repeat the above stratified by body region
df.tot.inj.body = df.injury %>%
  mutate(sport = factor(sport),
         gender = factor(gender),
         academicYear = factor(academicYear),
         injuryClassification = factor(injuryClassification)) %>%
  group_by(id, academicYear, injuryClassification) %>%
  dplyr::distinct(id, academicYear, .keep_all = TRUE) %>%
  ungroup() %>%
  group_by(sport, gender, academicYear, injuryClassification, .drop=F) %>%
  summarise(n.inj = n())

```

```

df.tot.inj.body = df.tot.inj.body[!grepl(";", df.tot.inj.body$sport), ] %>%
  filter(!(sport == "Golf" & gender == "F")) %>%
  arrange(sport) %>%
  arrange(desc(gender)) %>%
  arrange(injuryClassification)

sum(!as.vector(do.call("rbind", replicate(6, data.frame(df.tot.ath$sport), simplify = FALSE)))$df.tot.ath.sport == as.character(df.tot.inj.bo

df.tot.ath.6 = do.call("rbind", replicate(6, df.tot.ath, simplify = FALSE))
df.tot.body = cbind(df.tot.ath.6, df.tot.inj.body[,c("injuryClassification", "n.inj")]) %>%
  mutate(diff = player.count - n.inj,
         diff.analysis = ifelse((player.count - n.inj) < 0, 0, player.count - n.inj))

# Create necessary number of zeroes
add.zeros.body = function(data, sport, gender, injuryClassification, year, n){
  rbind(data, matrix(rep(c(sport, gender, injuryClassification, year, 0), n), byrow = T, nrow = n))
}
df.zeros.body = data.frame()
for (i in 1:dim(df.tot.body)[1]){
  df.zeros.body = add.zeros.body(df.zeros.body, df.tot.body$sport[i], df.tot.body$gender[i], as.character(df.tot.body$injuryClassification
})
z.id = 99999:(99999+dim(df.zeros.body)[1]-1)
df.zeros.body = cbind(z.id, df.zeros.body)
colnames(df.zeros.body) = c("id", "sport", "gender", "injuryClassification", "academicYear", "n.injuries")
df.zeros.body$id = as.numeric(df.zeros.body$id)
df.zeros.body$n.injuries = as.numeric(df.zeros.body$n.injuries)

# Count of injury per subject
df.fit.body = df.injury %>%
  group_by(id, sport, gender, injuryClassification, academicYear) %>%
  summarise(n.injuries = n()) %>%
  dplyr::select(id, sport, gender, injuryClassification, academicYear, n.injuries) %>%
  rbind(df.zeros.body) %>%
  ungroup() %>%
  mutate(gender = relevel(factor(gender), ref="M"))

### Supplementary table S1 ###
mod1 = geeglm(n.injuries ~ gender*injuryClassification + sport * injuryClassification + academicYear, id=id, data=df.fit.body, family=poisson)

# Add week of injury to injury data (relative to first week of season in the school year)
# Also add total training hours
create.week = function(data, Sport, Gender){
  df = df.injury %>%
    filter(sport == Sport & gender == Gender) %>%
    mutate(week = cut((lubridate::yday(as.Date(injuryDate)) - lubridate::yday(as.Date(data$...6[which(data[["Week"]]==1)]))) + 365) %>% 365,
           breaks = c(seq(-0.5, 363, by=7), Inf),
           labels = c(1:52)),
           week = as.numeric(week)) %>%
    rowwise() %>%
    mutate(total.hours = data[[which(data[["Week"]] == week), "Total Hours"]])
  return(df)
}
df.baseball.m = create.week(df.m.baseball, "Baseball", "M")
df.basketball.m = create.week(df.m.basketball, "Basketball", "M")
df.crew.m = create.week(df.m.crew, "Crew", "M")
df.cross.country.m = create.week(df.m.cross.country, "Cross Country", "M")
df.fencing.m = create.week(df.m.fencing, "Fencing", "M")
df.golf.m = create.week(df.m.golf, "Golf", "M")
df.soccer.m = create.week(df.m.soccer, "Soccer", "M")
df.swimming.m = create.week(df.m.swimming, "Swimming/Diving", "M")
df.tennis.m = create.week(df.m.tennis, "Tennis", "M")
df.volleyball.m = create.week(df.m.volleyball, "Volleyball", "M")
df.water.polo.m = create.week(df.m.water.polo, "Water Polo", "M")

df.baseball.f = create.week(df.f.baseball, "Baseball", "F")
df.basketball.f = create.week(df.f.basketball, "Basketball", "F")
df.crew.f = create.week(df.f.crew, "Crew", "F")
df.cross.country.f = create.week(df.f.cross.country, "Cross Country", "F")
df.fencing.f = create.week(df.f.fencing, "Fencing", "F")
df.soccer.f = create.week(df.f.soccer, "Soccer", "F")
df.swimming.f = create.week(df.f.swimming, "Swimming/Diving", "F")
df.tennis.f = create.week(df.f.tennis, "Tennis", "F")
df.volleyball.f = create.week(df.f.volleyball, "Volleyball", "F")
df.water.polo.f = create.week(df.f.water.polo, "Water Polo", "F")

df.track.f = create.week(df.track, "Track and Field", "F")
df.track.m = create.week(df.track, "Track and Field", "M")

df.injury.week = rbind(df.baseball.m, df.basketball.m, df.crew.m, df.cross.country.m, df.fencing.m, df.golf.m, df.soccer.m, df.swimming.m, df
df.training.week = rbind(df.m.baseball, df.m.basketball, df.m.crew, df.m.cross.country, df.m.fencing, df.m.golf, df.m.soccer, df.m.swimming,
df.f.baseball, df.f.basketball, df.f.crew, df.f.cross.country, df.f.fencing, df.f.soccer, df.f.swimming, df.f.tennis)
df.training.week = cbind(df.training.week, gender = c(rep("M", 52*12), rep("F", 52*11)), sport = c(rep("Baseball", 52), rep("Basketball",
df.injury.count.by.year.week = df.injury.week %>%
  ungroup() %>%
  mutate(week = as.factor(df.injury.week$week),
         gender = relevel(factor(gender), ref = "M")) %>%
  group_by(sport, gender, week, academicYear, .drop=F) %>%
  summarise(n.injuries = n())

df.injury.count.by.year.week.plt = df.injury.week %>%
  ungroup() %>%
  mutate(week = as.factor(df.injury.week$week),
         gender = relevel(factor(gender), ref = "M")) %>%
  group_by(sport, gender, week, .drop=F) %>%
  summarise(n.injuries = n())

df.clinical.incidence.by.year.week.plt= df.totals.long %>%

```

```

dplyr::select(sport, gender, n.athletes) %>%
merge(., df.injury.count.by.year.week.plt) %>%
mutate(incidence = n.injuries/n.athletes,
       gender = relevel(factor(gender), ref = "M"))

df.clinical.incidence.by.year.overall.week = df.clinical.incidence.by.year.week.plt %>%
  group_by(week, gender) %>%
  summarise(n.athletes = sum(n.athletes), n.injuries = sum(n.injuries)) %>%
  mutate(incidence = n.injuries/n.athletes)

### Figure 4 ###
ggplot(data=df.clinical.incidence.by.year.overall.week %>% mutate(gender = relevel(factor(gender), ref="F")), aes(x=as.numeric(week), y=incid
  geom_point(aes(color = gender))+
  geom_line(aes(color = gender)) +
  scale_color_manual(values=c("#69b3a2", "#404080")) +
  labs(x = "Week") +
  theme_bw()
ggsave("incidence-week.pdf", width = 7, height=3.5)

## Create dataset for analyzing at weekly level
df.tot.inj.week = df.injury.week %>%
  ungroup() %>%
  mutate(sport = factor(sport),
         gender = factor(gender),
         academicYear = factor(academicYear),
         week = factor(week)) %>%
  group_by(id, academicYear, week) %>%
  dplyr::distinct(id, academicYear, week, .keep_all = TRUE) %>%
  ungroup() %>%
  group_by(sport, gender, academicYear, week, total.hours, .drop=F) %>%
  summarise(n.inj = n())

df.tot.inj.week = df.tot.inj.week[!grepl(";", df.tot.inj.week$sport), ] %>%
  filter(!(sport == "Golf" & gender == "F")) %>%
  arrange(sport) %>%
  arrange(desc(gender)) %>%
  arrange(week)

sum(!as.vector(do.call("rbind", replicate(52, data.frame(df.tot.ath$sport), simplify = FALSE))))df.tot.ath == as.character(df.tot.inj.week$sp

df.tot.ath.52 = do.call("rbind", replicate(52, df.tot.ath, simplify = FALSE))
df.tot.week = cbind(df.tot.ath.52, df.tot.inj.week[,c("week", "n.inj")]) %>%
  mutate(diff = player.count - n.inj ,
         diff.analysis = ifelse(player.count - n.inj < 0, 0, player.count - n.inj))

# Create necessary number of zeroes
add.zeroes.week = function(data, sport, gender, week, year, n){
  rbind(data, matrix(rep(c(sport, gender, week, year, 0), n), byrow = T, nrow = n))
}
df.zeroes.week = data.frame()
for (i in 1:dim(df.tot.week)[1]){
  df.zeroes.week = add.zeroes.week(df.zeroes.week, df.tot.week$sport[i], df.tot.week$gender[i], df.tot.week$week[i], df.tot.week$season[i], d
}
z.id = 99999:(99999+dim(df.zeroes.week)[1]-1)
df.zeroes.week = cbind(z.id, df.zeroes.week)
colnames(df.zeroes.week) = c("id", "sport", "gender", "week", "academicYear", "n.injuries")
df.zeroes.week$id = as.numeric(df.zeroes.week$id)
df.zeroes.week$n.injuries = as.numeric(df.zeroes.week$n.injuries)
df.zeroes.week$week = as.numeric(df.zeroes.week$week)

# Count of injury per subject
df.fit.week = df.injury.week %>%
  group_by(id, sport, gender, week, academicYear) %>%
  summarise(n.injuries = n()) %>%
  rbind(df.zeroes.week) %>%
  ungroup() %>%
  mutate(gender =relevel(factor(gender), ref="M"))

df.fit.week$total.hours = rep(NA, dim(df.fit.week)[1])
for (i in 1:dim(df.fit.week)[1]){
  df.fit.week$total.hours[i] = df.training.week$`Total Hours`[which(df.training.week$Week == df.fit.week$week[i] & df.training.week$gender
}]

### Supplementary table S5 ###
mod2 = geeglm(n.injuries ~ week + gender + sport + academicYear, id=id, data=df.fit.week, family=poisson(link='log'))

### Supplementary table S6 ###
mod3 = geeglm(n.injuries ~ total.hours + gender + sport + academicYear, id=id, data=df.fit.week %>% subset(sport != "Track and Field"), fami

# Pairwise comparisons
comparisons = c("Baseball v Basketball", "Baseball v Crew", "Baseball v Cross Country", "Baseball v Fencing", "Baseball v Golf", "Baseball v
"Basketball v Crew", "Basketball v Cross Country", "Basketball v Fencing", "Basketball v Golf", "Basketball v Soccer", "Baske
"Crew v Cross Country", "Crew v Fencing", "Crew v Golf", "Crew v Soccer", "Crew v Swimming/Diving", "Crew v Tennis", "Crew v
"Cross Country v Fencing", "Cross Country v Golf", "Cross Country v Soccer", "Cross Country v Swimming/Diving", "Cross Countr
"Fencing v Golf", "Fencing v Soccer", "Fencing v Swimming/Diving", "Fencing v Tennis", "Fencing v Track", "Fencing v Volleyba
"Golf v Soccer", "Golf v Swimming/Diving", "Golf v Tennis", "Golf v Track", "Golf v Volleyball", "Golf v Water Polo",
"Soccer v Swimming/Diving", "Soccer v Tennis", "Soccer v Track", "Soccer v Volleyball", "Soccer v Water Polo",
"Swimming/Diving v Tennis", "Swimming/Diving v Track", "Swimming/Diving v Volleyball", "Swimming/Diving v Water Polo",
"Tennis v Track", "Tennis v Volleyball", "Tennis v Water Polo",
"Track v Volleyball", "Track v Water Polo",
"Volleyball v Water Polo")

sports = c("baseball", "basketball", "crew", "xc", "fencing", "golf", "soccer", "swimming", "tennis", "track", "volleyball", "waterpolo")
sports.level = c("Baseball", "Basketball", "Crew", "Cross Country", "Fencing", "Golf", "Soccer", "Swimming/Diving", "Tennis", "Track and Fiel

```

```

for (i in 1:length(sports)){
  nam = paste0("fit.pairwise.sport.", sports[i])
  dat = df.fit %>% mutate(sport = relevel(factor(sport), ref=sports.level[i]))
  fit = geeglm(n.injuries ~ sport, id=id, data=dat, family=poisson(link="log"))
  assign(nam, fit)
}

adj.bonf.sport = qnorm(0.025/66, lower.tail=F)
adj.bonf.gender.body = qnorm(0.025/6, lower.tail=F)

pairwise.pvals = c(summary(fit.pairwise.sport.baseball)$coefficients[2:12,4],
  summary(fit.pairwise.sport.basketball)$coefficients[3:12,4],
  summary(fit.pairwise.sport.crew)$coefficients[4:12,4],
  summary(fit.pairwise.sport.xc)$coefficients[5:12,4],
  summary(fit.pairwise.sport.fencing)$coefficients[6:12,4],
  summary(fit.pairwise.sport.golf)$coefficients[7:12,4],
  summary(fit.pairwise.sport.soccer)$coefficients[8:12,4],
  summary(fit.pairwise.sport.swimming)$coefficients[9:12,4],
  summary(fit.pairwise.sport.tennis)$coefficients[10:12,4],
  summary(fit.pairwise.sport.track)$coefficients[11:12,4],
  summary(fit.pairwise.sport.volleyball)$coefficients[12,4])

diffs = c(exp(summary(fit.pairwise.sport.baseball)$coefficients[2:12,1])*100-100,
  exp(summary(fit.pairwise.sport.basketball)$coefficients[3:12,1])*100-100,
  exp(summary(fit.pairwise.sport.crew)$coefficients[4:12,1])*100-100,
  exp(summary(fit.pairwise.sport.xc)$coefficients[5:12,1])*100-100,
  exp(summary(fit.pairwise.sport.fencing)$coefficients[6:12,1])*100-100,
  exp(summary(fit.pairwise.sport.golf)$coefficients[7:12,1])*100-100,
  exp(summary(fit.pairwise.sport.soccer)$coefficients[8:12,1])*100-100,
  exp(summary(fit.pairwise.sport.swimming)$coefficients[9:12,1])*100-100,
  exp(summary(fit.pairwise.sport.tennis)$coefficients[10:12,1])*100-100,
  exp(summary(fit.pairwise.sport.track)$coefficients[11:12,1])*100-100,
  exp(summary(fit.pairwise.sport.volleyball)$coefficients[12,1])*100-100)

diffs.ci.lwr = c(exp(summary(fit.pairwise.sport.baseball)$coefficients[2:12,1] - adj.bonf.sport*summary(fit.pairwise.sport.baseball)$coeffici
  exp(summary(fit.pairwise.sport.basketball)$coefficients[3:12,1] - adj.bonf.sport*summary(fit.pairwise.sport.basketball)$coef
  exp(summary(fit.pairwise.sport.crew)$coefficients[4:12,1] - adj.bonf.sport*summary(fit.pairwise.sport.crew)$coefficients[4:1
  exp(summary(fit.pairwise.sport.xc)$coefficients[5:12,1] - adj.bonf.sport*summary(fit.pairwise.sport.xc)$coefficients[5:12,2]
  exp(summary(fit.pairwise.sport.fencing)$coefficients[6:12,1] - adj.bonf.sport*summary(fit.pairwise.sport.fencing)$coefficien
  exp(summary(fit.pairwise.sport.golf)$coefficients[7:12,1] - adj.bonf.sport*summary(fit.pairwise.sport.golf)$coefficients[7:1
  exp(summary(fit.pairwise.sport.soccer)$coefficients[8:12,1] - adj.bonf.sport*summary(fit.pairwise.sport.soccer)$coefficients
  exp(summary(fit.pairwise.sport.swimming)$coefficients[9:12,1] - adj.bonf.sport*summary(fit.pairwise.sport.swimming)$coeffici
  exp(summary(fit.pairwise.sport.tennis)$coefficients[10:12,1] - adj.bonf.sport*summary(fit.pairwise.sport.tennis)$coefficient
  exp(summary(fit.pairwise.sport.track)$coefficients[11:12,1] - adj.bonf.sport*summary(fit.pairwise.sport.track)$coefficients[
  exp(summary(fit.pairwise.sport.volleyball)$coefficients[12,1] - adj.bonf.sport*summary(fit.pairwise.sport.volleyball)$coeffi

diffs.ci.upr = c(exp(summary(fit.pairwise.sport.baseball)$coefficients[2:12,1] + adj.bonf.sport*summary(fit.pairwise.sport.baseball)$coeffici
  exp(summary(fit.pairwise.sport.basketball)$coefficients[3:12,1] + adj.bonf.sport*summary(fit.pairwise.sport.basketball)$coef
  exp(summary(fit.pairwise.sport.crew)$coefficients[4:12,1] + adj.bonf.sport*summary(fit.pairwise.sport.crew)$coefficients[4:1
  exp(summary(fit.pairwise.sport.xc)$coefficients[5:12,1] + adj.bonf.sport*summary(fit.pairwise.sport.xc)$coefficients[5:12,2]
  exp(summary(fit.pairwise.sport.fencing)$coefficients[6:12,1] + adj.bonf.sport*summary(fit.pairwise.sport.fencing)$coefficien
  exp(summary(fit.pairwise.sport.golf)$coefficients[7:12,1] + adj.bonf.sport*summary(fit.pairwise.sport.golf)$coefficients[7:1
  exp(summary(fit.pairwise.sport.soccer)$coefficients[8:12,1] + adj.bonf.sport*summary(fit.pairwise.sport.soccer)$coefficients
  exp(summary(fit.pairwise.sport.swimming)$coefficients[9:12,1] + adj.bonf.sport*summary(fit.pairwise.sport.swimming)$coeffici
  exp(summary(fit.pairwise.sport.tennis)$coefficients[10:12,1] + adj.bonf.sport*summary(fit.pairwise.sport.tennis)$coefficient
  exp(summary(fit.pairwise.sport.track)$coefficients[11:12,1] + adj.bonf.sport*summary(fit.pairwise.sport.track)$coefficients[
  exp(summary(fit.pairwise.sport.volleyball)$coefficients[12,1] + adj.bonf.sport*summary(fit.pairwise.sport.volleyball)$coeffi

diffs.ci = paste0("($", diffs.ci.lwr %>% round(2), "$", "$", diffs.ci.upr %>% round(2), "$)")

sports.comparisons.tbl = data.frame(comparisons, pairwise.pvals) %>%
  mutate(pairwise.pvals.bonferroni = ifelse(pairwise.pvals*66>1, 1, pairwise.pvals*66),
    pairwise.pvals.bonferroni.holm = p.adjust(pairwise.pvals, method="holm")) %>%
  mutate(pairwise.pvals = pvalFormat(pairwise.pvals),
    pairwise.pvals.bonferroni = pvalFormat(pairwise.pvals.bonferroni),
    pairwise.pvals.bonferroni.holm = pvalFormat(pairwise.pvals.bonferroni.holm),
    diff = paste0("$", diffs %>% round(2), "$"),
    ci = diffs.ci)

rownames(sports.comparisons.tbl) = sports.comparisons.tbl[,1]
colnames(sports.comparisons.tbl) = c("Comparisons", "Unadjusted P-value", "P-value (Bonferroni)", "P-value (Bonferroni-Holm)", "Difference (%)

bodies = c("head", "le", "misc", "spine", "trunk", "ue")
bodies.level = c("Head", "Lower Extremity", "Miscellaneous", "Spine", "Trunk", "Upper Extremity")

for (i in 1:length(bodies)){
  nam = paste0("fit.pairwise.body.", bodies[i])
  dat = df.fit.body %>% subset(injuryClassification==bodies.level[i])
  fit = geeglm(n.injuries ~ gender, id=id, data=dat, family=poisson(link="log"))
  assign(nam, fit)
}

diffs.body = c(summary(contrast(regrid(emmeans(fit.pairwise.body.head, pairwise~gender)), "revpairwise"))$estimate*100,
  summary(contrast(regrid(emmeans(fit.pairwise.body.le, pairwise~gender)), "revpairwise"))$estimate*100,
  summary(contrast(regrid(emmeans(fit.pairwise.body.misc, pairwise~gender)), "revpairwise"))$estimate*100,
  summary(contrast(regrid(emmeans(fit.pairwise.body.spine, pairwise~gender)), "revpairwise"))$estimate*100,
  summary(contrast(regrid(emmeans(fit.pairwise.body.trunk, pairwise~gender)), "revpairwise"))$estimate*100,
  summary(contrast(regrid(emmeans(fit.pairwise.body.ue, pairwise~gender)), "revpairwise"))$estimate*100)

diffs.body.ci.lwr = c((summary(contrast(regrid(emmeans(fit.pairwise.body.head, pairwise~gender)), "revpairwise"))$estimate- adj.bonf.gender.b
  (summary(contrast(regrid(emmeans(fit.pairwise.body.le, pairwise~gender)), "revpairwise"))$estimate- adj.bonf.gender.bod
  (summary(contrast(regrid(emmeans(fit.pairwise.body.misc, pairwise~gender)), "revpairwise"))$estimate- adj.bonf.gender.b
  (summary(contrast(regrid(emmeans(fit.pairwise.body.spine, pairwise~gender)), "revpairwise"))$estimate- adj.bonf.gender.
  (summary(contrast(regrid(emmeans(fit.pairwise.body.trunk, pairwise~gender)), "revpairwise"))$estimate- adj.bonf.gender.
  (summary(contrast(regrid(emmeans(fit.pairwise.body.ue, pairwise~gender)), "revpairwise"))$estimate- adj.bonf.gender.bod

diffs.body.ci.upr = c((summary(contrast(regrid(emmeans(fit.pairwise.body.head, pairwise~gender)), "revpairwise"))$estimate+ adj.bonf.gender.b
  (summary(contrast(regrid(emmeans(fit.pairwise.body.le, pairwise~gender)), "revpairwise"))$estimate+ adj.bonf.gender.bod

```

```

(summary(contrast(regrid(emmeans(fit.pairwise.body.misc, pairwise~gender)), "revpairwise"))$estimate+ adj.bonf.gender.b
(summary(contrast(regrid(emmeans(fit.pairwise.body.spine, pairwise~gender)), "revpairwise"))$estimate+ adj.bonf.gender.
(summary(contrast(regrid(emmeans(fit.pairwise.body.trunk, pairwise~gender)), "revpairwise"))$estimate+ adj.bonf.gender.
(summary(contrast(regrid(emmeans(fit.pairwise.body.ue, pairwise~gender)), "revpairwise"))$estimate+ adj.bonf.gender.bod

diffs.body.ci = paste0("$", diffs.body.ci.lwr %>% round(2), "$, $", diffs.body.ci.upr %>% round(2), "$")

body.comparisons.tbl = data.frame(pairwise.pvals = c(summary(fit.pairwise.body.head)$coefficients[2,4],
summary(fit.pairwise.body.le)$coefficients[2,4],
summary(fit.pairwise.body.misc)$coefficients[2,4],
summary(fit.pairwise.body.spine)$coefficients[2,4],
summary(fit.pairwise.body.trunk)$coefficients[2,4],
summary(fit.pairwise.body.ue)$coefficients[2,4])) %>%
mutate(pairwise.pvals.bonferroni = ifelse(pairwise.pvals*6>1, 1, pairwise.pvals*6),
pairwise.pvals.bonferroni.holm = p.adjust(pairwise.pvals, method="holm")) %>%
mutate(pairwise.pvals = pvalFormat(pairwise.pvals),
pairwise.pvals.bonferroni = pvalFormat(pairwise.pvals.bonferroni),
pairwise.pvals.bonferroni.holm = pvalFormat(pairwise.pvals.bonferroni.holm),
diff = paste0("$", diffs.body %>% round(2), "$"),
ci = diffs.body.ci)
rownames(body.comparisons.tbl) = c("Head", "Lower Extremity", "Miscellaneous", "Spine", "Trunk", "Upper Extremity")
colnames(body.comparisons.tbl) = c("Unadjusted P-value", "P-value\n(Bonferroni)", "P-value\n(Bonferroni-Holm)", "Difference\n(%)", "95% CI\nA

```