NFL Fatigue Analysis in Defensive Linemen

Sarah Deussing & Imogen Meers

2024-11-11

Motivation

Our analysis will study fatigue in defensive linemen using their speed, acceleration, distance traveled, and orientation. We will be calculating a player load and analyze the defensive player's effectiveness. To do so, we will combine this movement data with play outcome data - if the player sacked the quarterback, had a solo tackle, etc.

The outcome of our analysis will be to predict likelihood of "mistakes" and positive plays, based on player fatigue and other game-time variables and to identify if specific players are more fatigue-resistant.

There has been published research on long-term fatigue and "burnout" in NFL players over the course of their careers, which focuses on long-term health and physiological effects. There has also been research on game day fatigue in European Football to determine limitations on performance based on fatigue, in order to create suitable, enhancing interventions (Alghannam). However, there is no player specific or NFL game day based fatigue analysis, which is what will set our project apart.

Research

We wanted to find a measure of fatigue that was derived from tracking data. From an study on *Monitoring Fatigue During Intermittent Exercise With Accelerometer-Derived Metrics* and Catapult's own description of Player Load, we found this idea of "Dynamic Load", that is a sum of triaxial acceleration across a set time period. We do not have acceleration in three plains but we do have an acceleration in the direction of play so we decided to derive a "Load per Play" value that will be acceleration summed per player per play.

As an extension to this, we will also derive a "Fatigue Index" based on the research by Beato et al., that will be a value of acceleration/speed for each value in time, i.

Problem framing

We will be creating two derived variables:

- Load by player by play
- Fatigue index by player by play

Perform a time-series analysis over the length of a game/period to see trend and seasonality of fatigue and the change in fatigue at specific events, e.g. "mistakes".

Define mistake as penalties that will be grouped based on type: - Formation - Illegal Shift, Illegal Formation, Defensive Offside

- Tackle
    - Defensive Holding, Horse Collar Tackle, Low Block, Illegal Use of Hands, Roughing the Passer

Use player demographic data, time, tracking and load data to predict probability of "mistake" at a given time.

Use player demographic data, time, tracking and load data to predict positive defensive outcomes such as sacks, pressure and fumbles.

Identify fatigue-resistant players and cluster to see if they have similar demographic characteristics.

Data overview

The dataset is a part of the NFL Big Data Bowl 2025, and it is taken from: https://www.kaggle.com/competitions/nfl-big-data-bowl-2025/data (https://www.kaggle.com/competitions/nfl-big-data-bowl-2025/data)

We are going to use four main parts of the dataset: player_play, players, plays, and the tracking data for each week.

The player_play data contains information about the action of the play, including its outcome or any penalties. Each row corresponds to one player's individual actions for each play within a game.

The players data contains information about each player, including position, height, weight, and college. We will use this data to filter to only defensive linemen within all data sets.

The plays data contains information about the quarter, time, and contents of each play.

The tracking data has the x-coordinate, y-coordinate, direction, orientation, speed, and acceleration for each player involved in each play in a game. We will use this movement data to calculate player fatigue. This tracking data is available for all 9 weeks of the season.

The datasets can be joined on: gameId, playId, and nflId.

Contribution

Imogen Meers & Sarah Deussing

Bibliography

Alghannam, Abdullah F. "Metabolic limitations of performance and fatigue in football." Asian journal of sports medicine vol. 3,2 (2012): 65-73. doi:10.5812/asjsm.34699 (doi:10.5812/asjsm.34699)

Beato, Marco et al. "Monitoring Fatigue During Intermittent Exercise With Accelerometer-Derived Metrics." Frontiers in physiology vol. 10 780. 26 Jun. 2019, doi:10.3389/fphys.2019.00780 (doi:10.3389/fphys.2019.00780)

Finster-Rowen, Alex. "What is Player Load?" Catapult Sports. Catapult Sports, n.d. Web.

Implementation

```
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(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.1
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
library(tidyr)
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.4.1
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

Data

Play Data for Each Player

```
play <- read.csv('player_play.csv')
player_data <- read.csv('players.csv')
```

Filter to only defensive linemen. - DE (Defensive End) - DT (Defensive Tackle) - NT (Nose Tackle)

```
def_line_players <- player_data[player_data$position == 'DE' | player_data$position == 'DT' |
                          player_data$position == 'NT',]

def_line <- play %>%
  left_join(def_line_players[, c("nflId", "position")], by = "nflId")

def_line <- def_line %>%
  select(-c('hadRushAttempt', 'rushingYards', 'hadDropback', 'passingYards', 'sackYardsAsOffens
e',
            'hadPassReception', 'receivingYards', 'wasTargettedReceiver', 'yardageGainedAfterThe
Catch',
            'fumbles', 'fumbleLost', 'fumbleOutOfBounds', 'wasInitialPassRusher', 'wasRunningRou
te',
            'routeRan', 'blockedPlayerNFLId1', 'blockedPlayerNFLId2', 'blockedPlayerNFLId3'))
```

Variables for defensive linemen: - assistedTackle: required an assist to make a tackle - forcedFumbleAsDefense: forced a fumble by the opposing team - halfSackYardsAsDefense: yards conceded by the offense because of a half-sack - passDefensed: a passing play was stopped by the player - quarterbackHit: player recorded a QB hit - sackYardsAsDefense: yards conceded by the offense because of a sack by the player - safetyAsDefense: player forced a safety on this play - soloTackle: player recorded a solo tackle on this play - tackleAssist: player was awarded an assisted tackle - tackleForALoss: player recorded a tackle behind the line of scrimmage - tackleForALossYardage: yards conceded by the offense because of a tackle - hadInterception: player intercepted a pass - interceptionYards: yards returned by the player on an intercepted pass - fumbleRecoveries: number of fumbles recovered by the player - fumbleRecoveryYards: yards returned by the player on a fumble recovery - penaltyNames - causedPressure: player pressured the QB - timeToPressureAsPassRusher: time elapsed from snap to player reaching a pressure pro. >= to 0.75 - inMotionAtBallSnap: player was in motion at snap - shiftSinceLineset: player shifted since the lineset - pressureAllowedAsBlocker: any pass rushers that the blocker had a true matchup against recorded a pressure - timeToPressureAllowedAsBlocker: time from snap to pass rusher who blocker had true matchup against achieving a pressure prob. > 0.75 - pff_defensiveCoverageAssignment: defensive coverage assignment given to the player

```
summary(def_line)
```

```
##      gameId              playId         nflId         teamAbbr
##  Min.   :2.022e+09   Min.   :  54   Min.   :25511   Length:354727
##  1st Qu.:2.022e+09   1st Qu.: 996   1st Qu.:43426   Class :character
##  Median :2.022e+09   Median :2017   Median :46457   Mode  :character
##  Mean   :2.022e+09   Mean   :2024   Mean   :47437
##  3rd Qu.:2.022e+09   3rd Qu.:3022   3rd Qu.:52590
##  Max.   :2.022e+09   Max.   :5120   Max.   :55241
##
##  assistedTackle     forcedFumbleAsDefense halfSackYardsAsDefense
##  Min.   :0.000000   Min.   :0.0000000     Min.   :-18.000000
##  1st Qu.:0.000000   1st Qu.:0.0000000     1st Qu.:  0.000000
##  Median :0.000000   Median :0.0000000     Median :  0.000000
##  Mean   :0.004138   Mean   :0.0005131     Mean   : -0.002154
##  3rd Qu.:0.000000   3rd Qu.:0.0000000     3rd Qu.:  0.000000
##  Max.   :1.000000   Max.   :1.0000000     Max.   :  0.000000
##
##   passDefensed      quarterbackHit     sackYardsAsDefense  safetyAsDefense
##  Min.   :0.000000   Min.   :0.000000   Min.   :-17.00000   Min.   :0.00e+00
##  1st Qu.:0.000000   1st Qu.:0.000000   1st Qu.:  0.00000   1st Qu.:0.00e+00
##  Median :0.000000   Median :0.000000   Median :  0.00000   Median :0.00e+00
##  Mean   :0.003093   Mean   :0.003986   Mean   : -0.01036   Mean   :1.13e-05
##  3rd Qu.:0.000000   3rd Qu.:0.000000   3rd Qu.:  0.00000   3rd Qu.:0.00e+00
##  Max.   :1.000000   Max.   :1.000000   Max.   :  0.00000   Max.   :1.00e+00
##
##    soloTackle       tackleAssist      tackleForALoss    tackleForALossYardage
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   : 0.0000
##  1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.: 0.0000
##  Median :0.00000   Median :0.00000   Median :0.00000   Median : 0.0000
##  Mean   :0.02483   Mean   :0.01544   Mean   :0.00327   Mean   : 0.0128
##  3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.: 0.0000
##  Max.   :1.00000   Max.   :1.00000   Max.   :1.00000   Max.   :17.0000
##
##  hadInterception     interceptionYards  fumbleRecoveries    fumbleRecoveryYards
##  Min.   :0.0000000   Min.   :-6.00000   Min.   :0.0000000   Min.   :-15.00000
##  1st Qu.:0.0000000   1st Qu.: 0.00000   1st Qu.:0.0000000   1st Qu.:  0.00000
##  Median :0.0000000   Median : 0.00000   Median :0.0000000   Median :  0.00000
##  Mean   :0.0005441   Mean   : 0.00708   Mean   :0.0007076   Mean   :  0.00127
##  3rd Qu.:0.0000000   3rd Qu.: 0.00000   3rd Qu.:0.0000000   3rd Qu.:  0.00000
##  Max.   :1.0000000   Max.   :99.00000   Max.   :2.0000000   Max.   : 68.00000
##
##   penaltyYards       penaltyNames       causedPressure
##  Min.   : 0.000000   Length:354727      Mode :logical
##  1st Qu.: 0.000000   Class :character   FALSE:350420
##  Median : 0.000000   Mode  :character   TRUE :4307
##  Mean   : 0.006614
##  3rd Qu.: 0.000000
##  Max.   :20.000000
##
##  timeToPressureAsPassRusher getOffTimeAsPassRusher inMotionAtBallSnap
##  Min.   : 0.8               Min.   :0.00           Mode :logical
##  1st Qu.: 2.2               1st Qu.:0.80           FALSE:103276
##  Median : 2.7               Median :0.96           TRUE :4572
##  Mean   : 2.9               Mean   :1.01           NA's :246879
##  3rd Qu.: 3.2               3rd Qu.:1.17
##  Max.   :11.6               Max.   :2.00
##  NA's   :350399             NA's   :306695
##  shiftSinceLineset motionSinceLineset pressureAllowedAsBlocker
##  Mode :logical     Mode :logical      Min.   :0.00
##  FALSE:172421      FALSE:84416        1st Qu.:0.00
##  TRUE :3757        TRUE :5822         Median :0.00
##  NA's :178549      NA's :264489       Mean   :0.08
##                                       3rd Qu.:0.00
##                                       Max.   :1.00
##                                       NA's   :301683
##  timeToPressureAllowedAsBlocker pff_defensiveCoverageAssignment
##  Min.   : 0.8                   Length:354727
##  1st Qu.: 2.3                   Class :character
##  Median : 2.7                   Mode  :character
##  Mean   : 2.9
##  3rd Qu.: 3.3
##  Max.   :11.6
##  NA's   :350647
##  pff_primaryDefensiveCoverageMatchupNflId
##  Min.   :29550
##  1st Qu.:44841
##  Median :47791
##  Mean   :47938
##  3rd Qu.:52608
##  Max.   :55168
##  NA's   :311243
##  pff_secondaryDefensiveCoverageMatchupNflId   position
##  Min.   :30842                              Length:354727
##  1st Qu.:44860                              Class :character
##  Median :46705                              Mode  :character
##  Mean   :47983
##  3rd Qu.:52645
##  Max.   :55157
##  NA's   :352340
```

Tracking Data

```
week1 <- read.csv('tracking_week_1.csv')
week2 <- read.csv('tracking_week_2.csv')
week3 <- read.csv('tracking_week_3.csv')
weeks <- rbind(week1, week2, week3)
#save(weeks, file = "weeks123.RData")
```

The following variables from the tracking data can be used to calculate fatigue: - s (speed), a (acceleration), dis (distance traveled), dir (direction), o (orientation)

Join Data

```
data <- inner_join(def_line, weeks, by = c("gameId", "playId", "nflId"))
#save(data, file = "joinedData.RData")
```

```
# load("joinedData.RData")
# colnames(data)
# head(data)
```

Exploratory Analysis/Visualizations

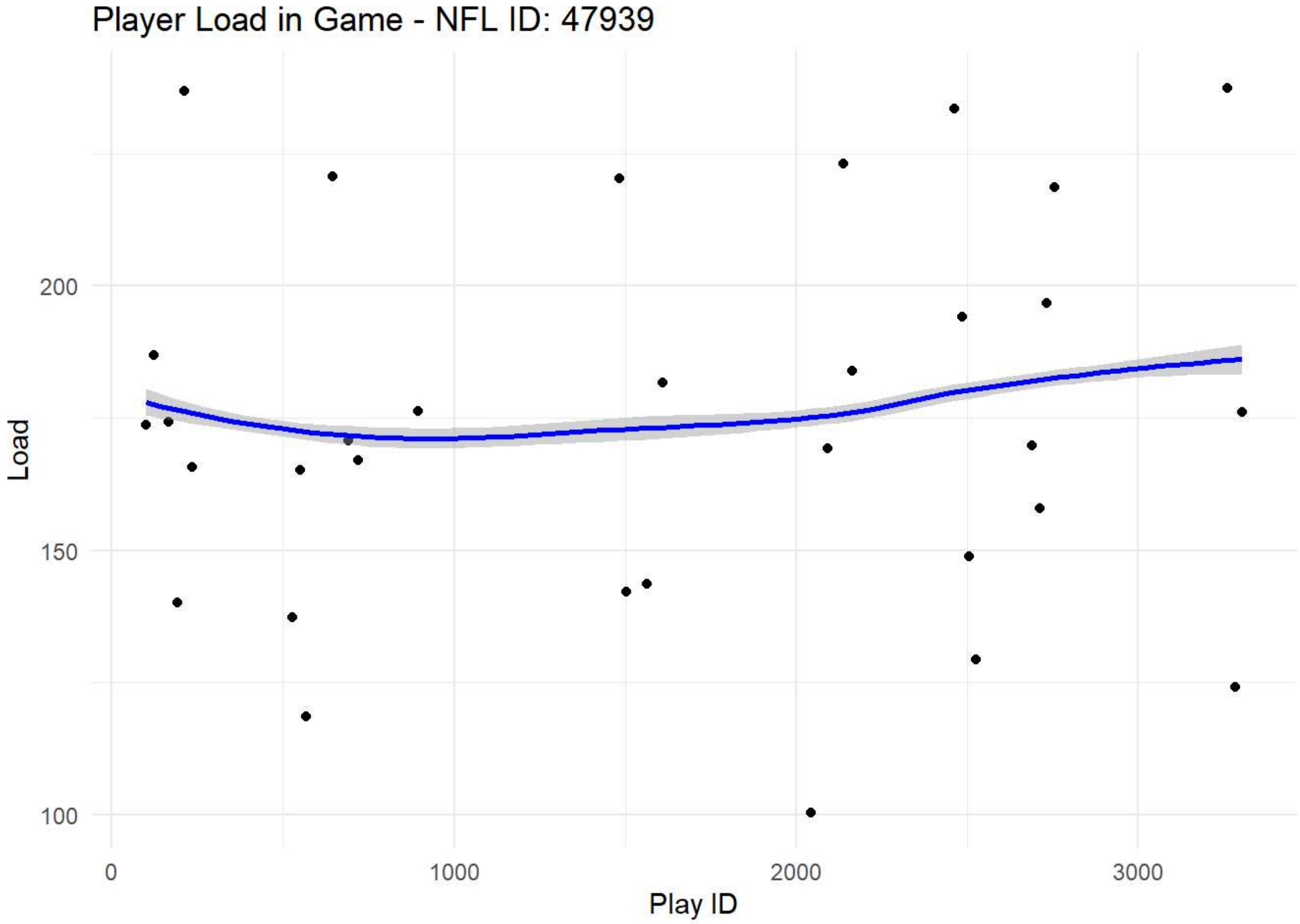
Our first derived variable is player load. We will calculate load on a per-play level. Player Load = SUM(Acceleration)

```
data <- data %>%
  group_by(gameId, playId, nflId) %>%
  mutate(load = sum(a, na.rm = TRUE)) %>% ungroup()
```

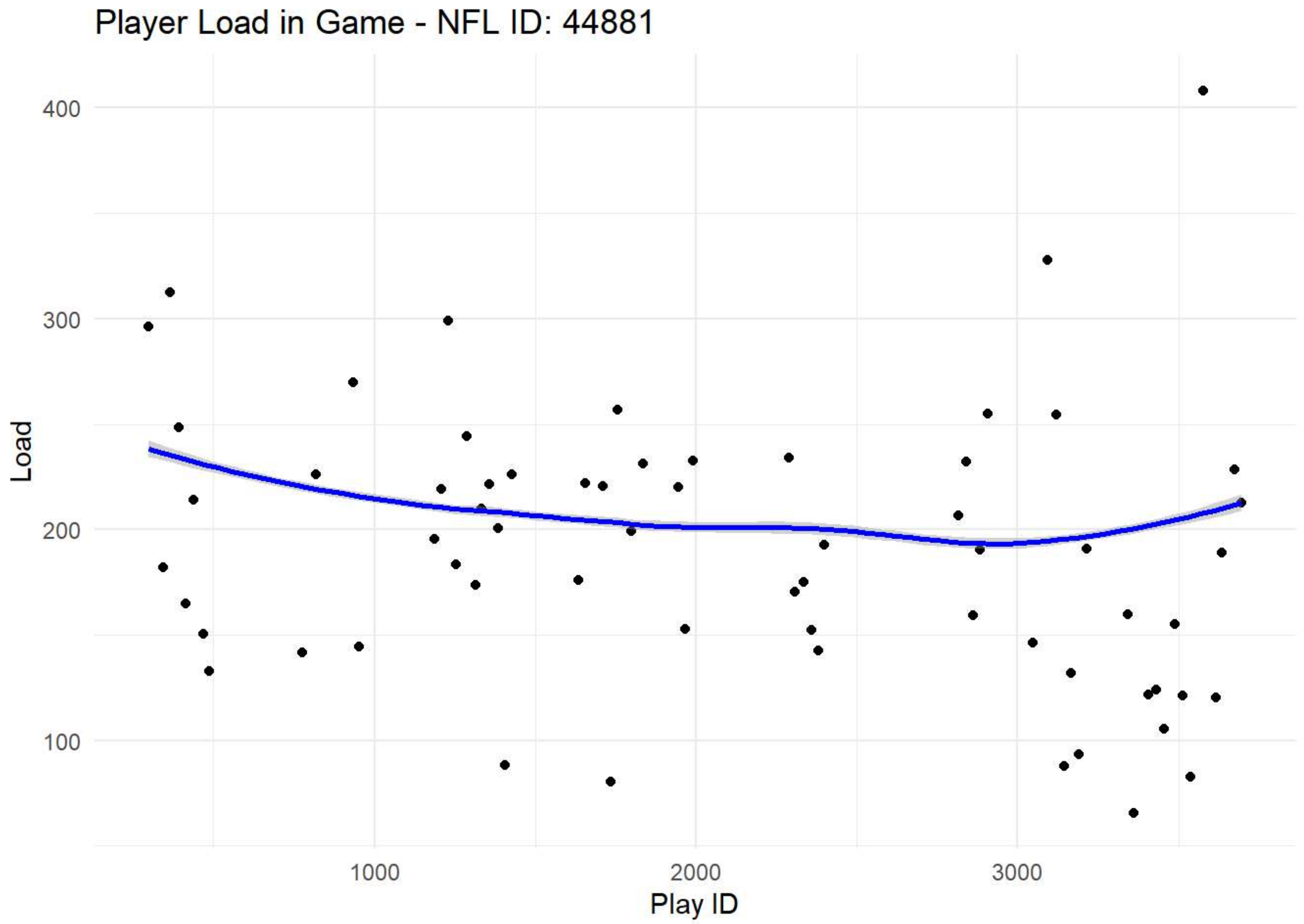
Load Per Player

```
game <- data[data$gameId == 2022090800,]

# select 10 defensive linemen at random
set.seed(456)
ran_nflIds <- sample(game$nflId, 10)

for (id in ran_nflIds) {
  player_data <- game %>% filter(nflId == id)
  load_plot <- ggplot(player_data, aes(x = playId, y = load)) +
    geom_point() +
    geom_smooth(method = "loess", color = "blue") +
    labs(x = "Play ID", y = "Load", title = paste0("Player Load in Game - NFL ID: ", id)) +
    theme_minimal()

  print(load_plot)
}
```

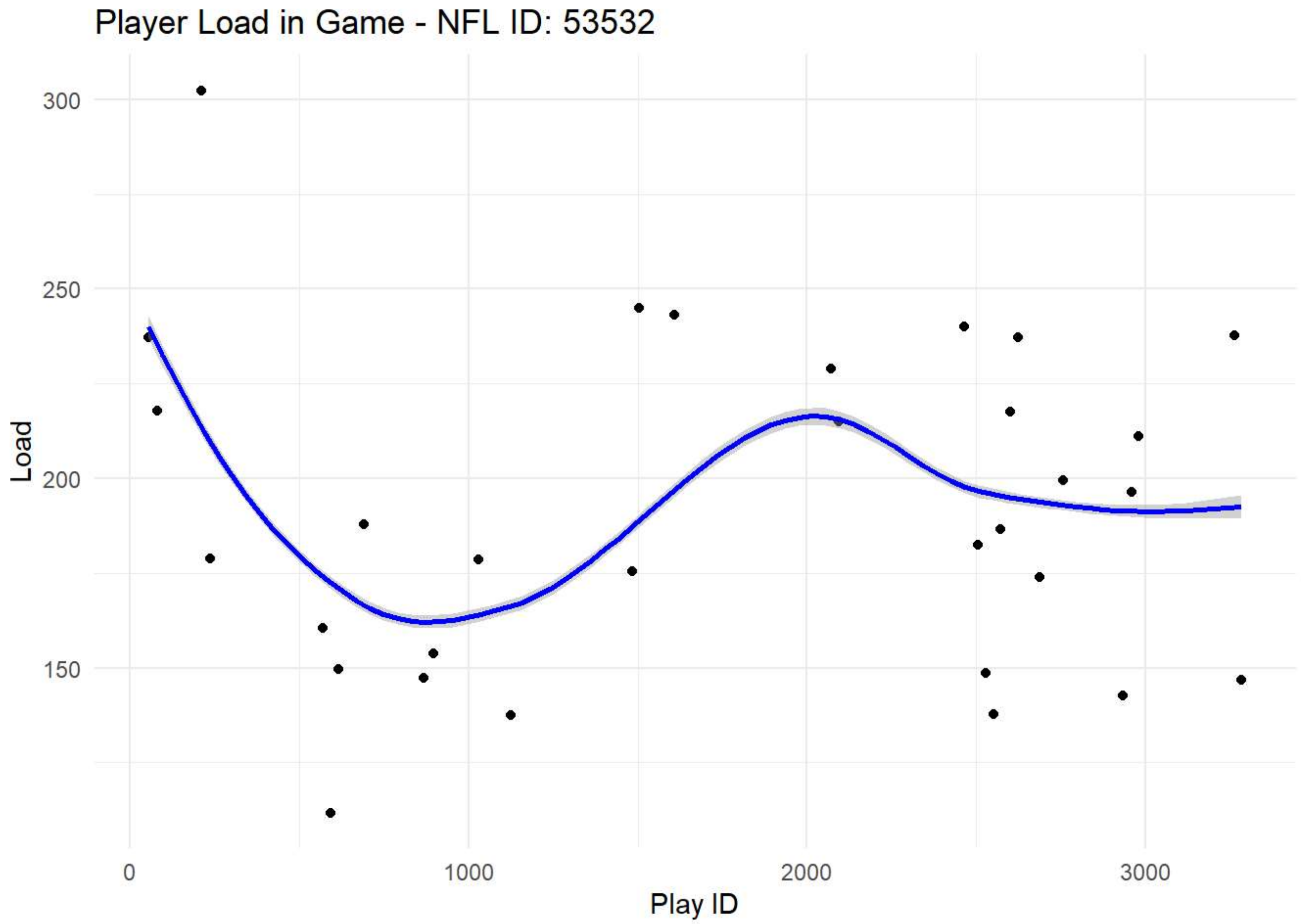
```
## `geom_smooth()` using formula = 'y ~ x'
```



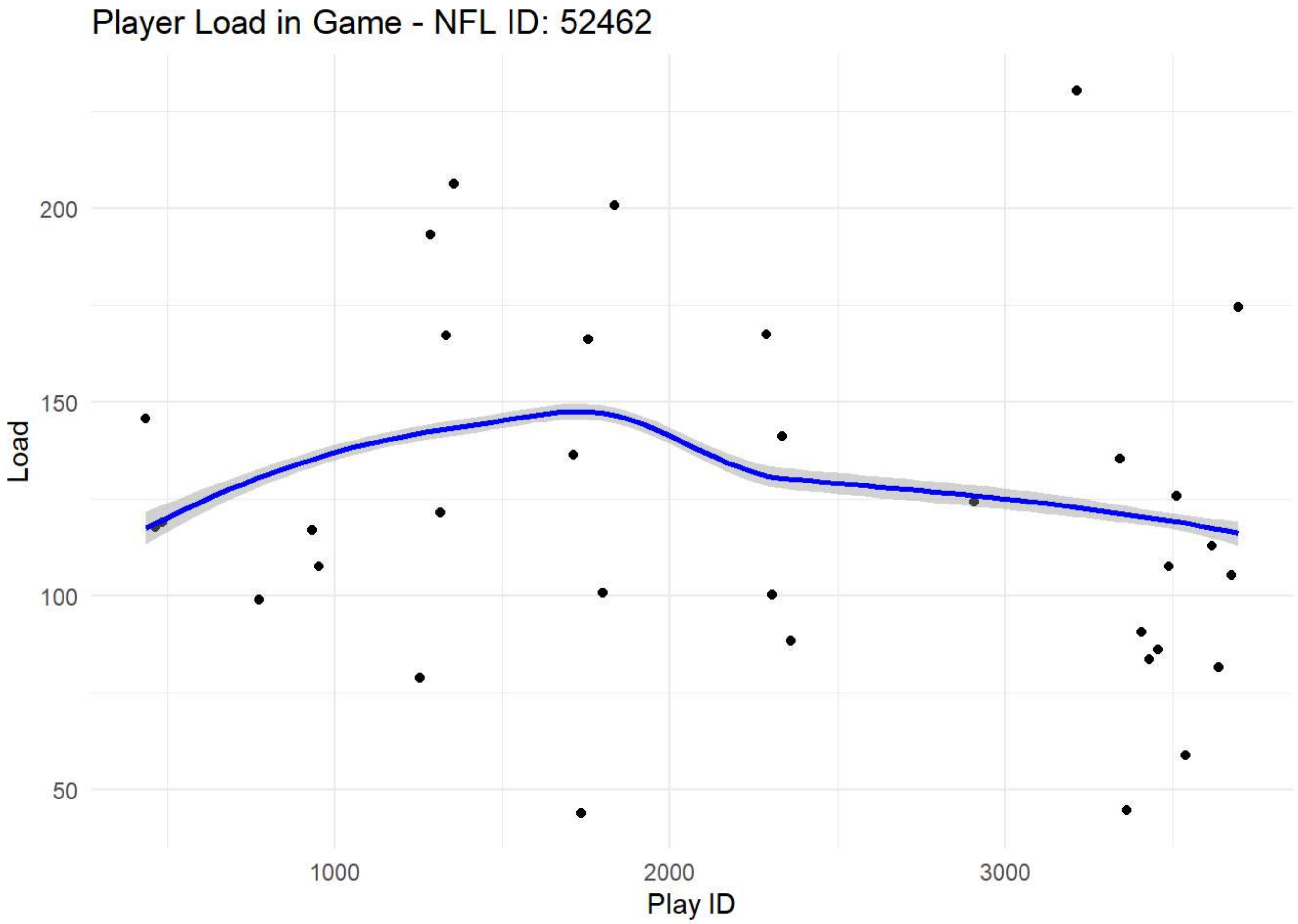
```
## `geom_smooth()` using formula = 'y ~ x'
```



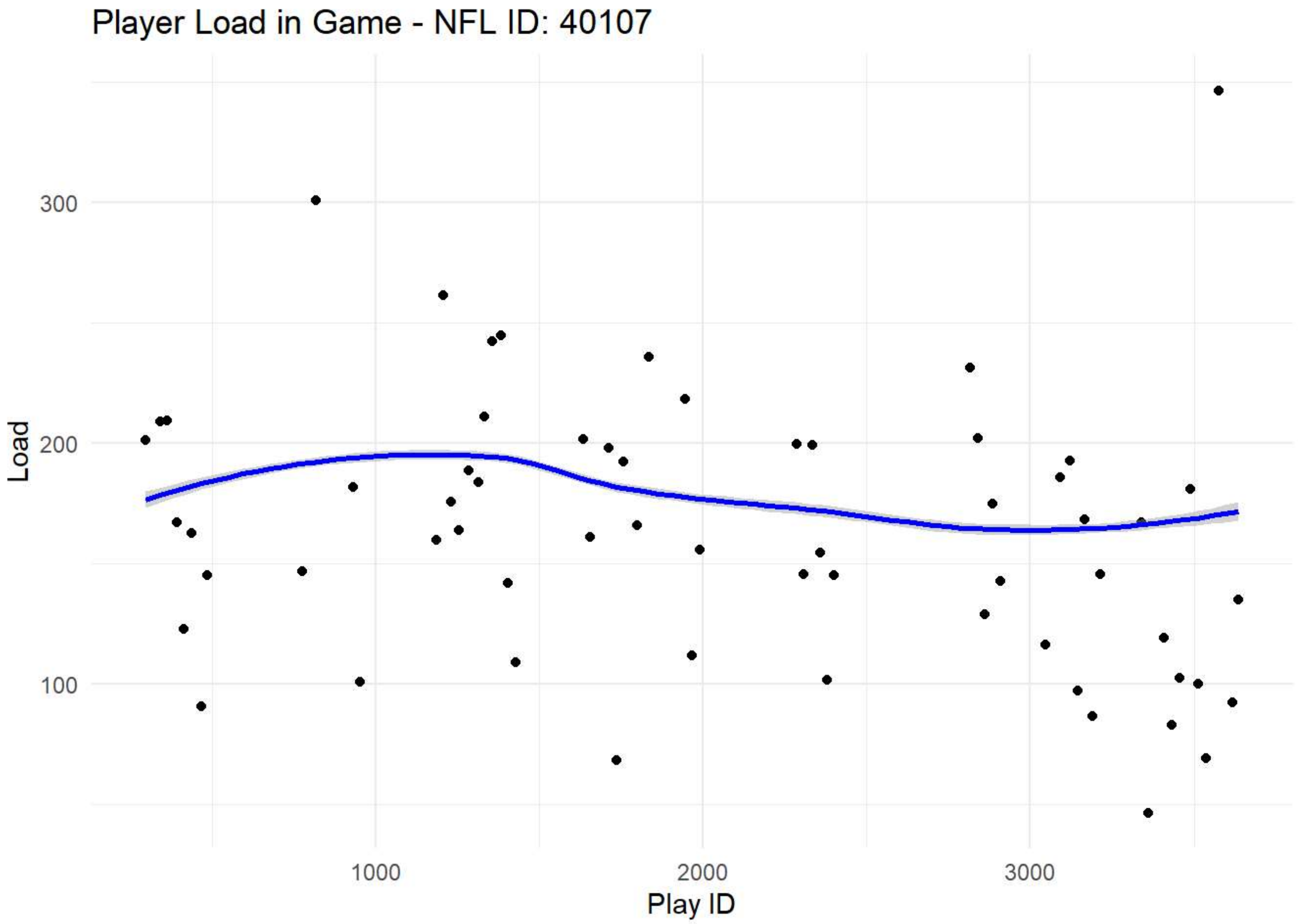
```
## `geom_smooth()` using formula = 'y ~ x'
```



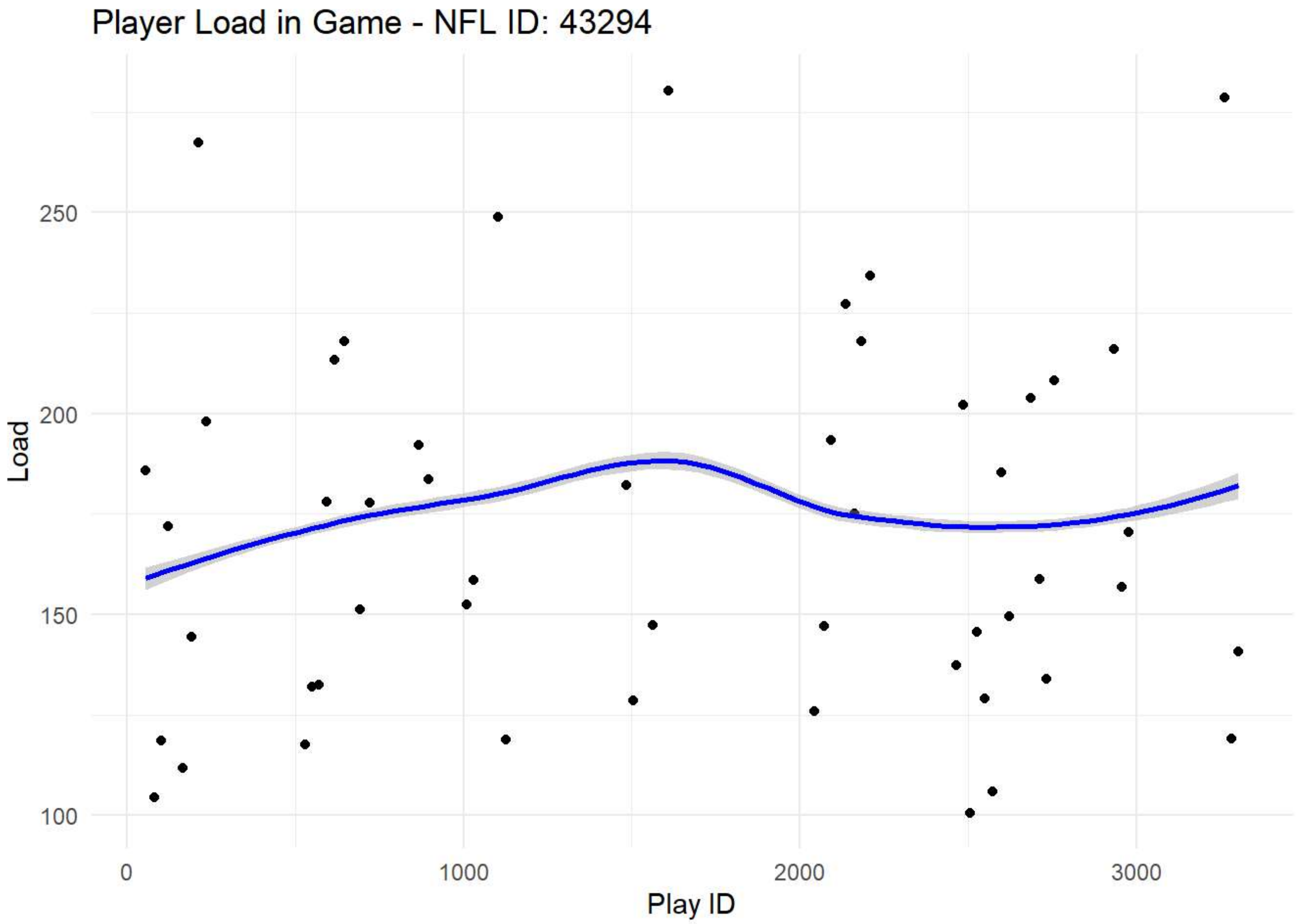
```
## `geom_smooth()` using formula = 'y ~ x'
```



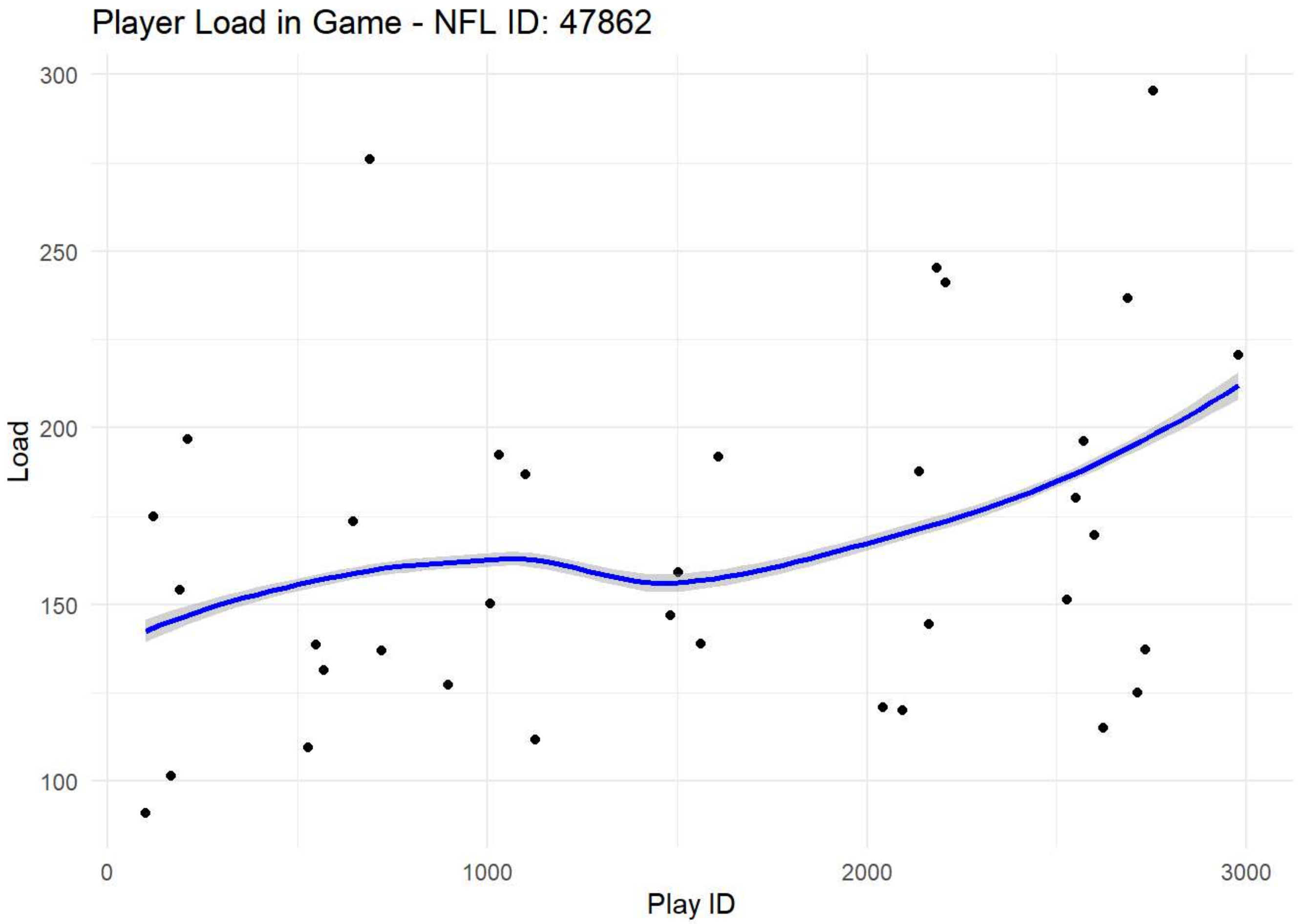
```
## `geom_smooth()` using formula = 'y ~ x'
```



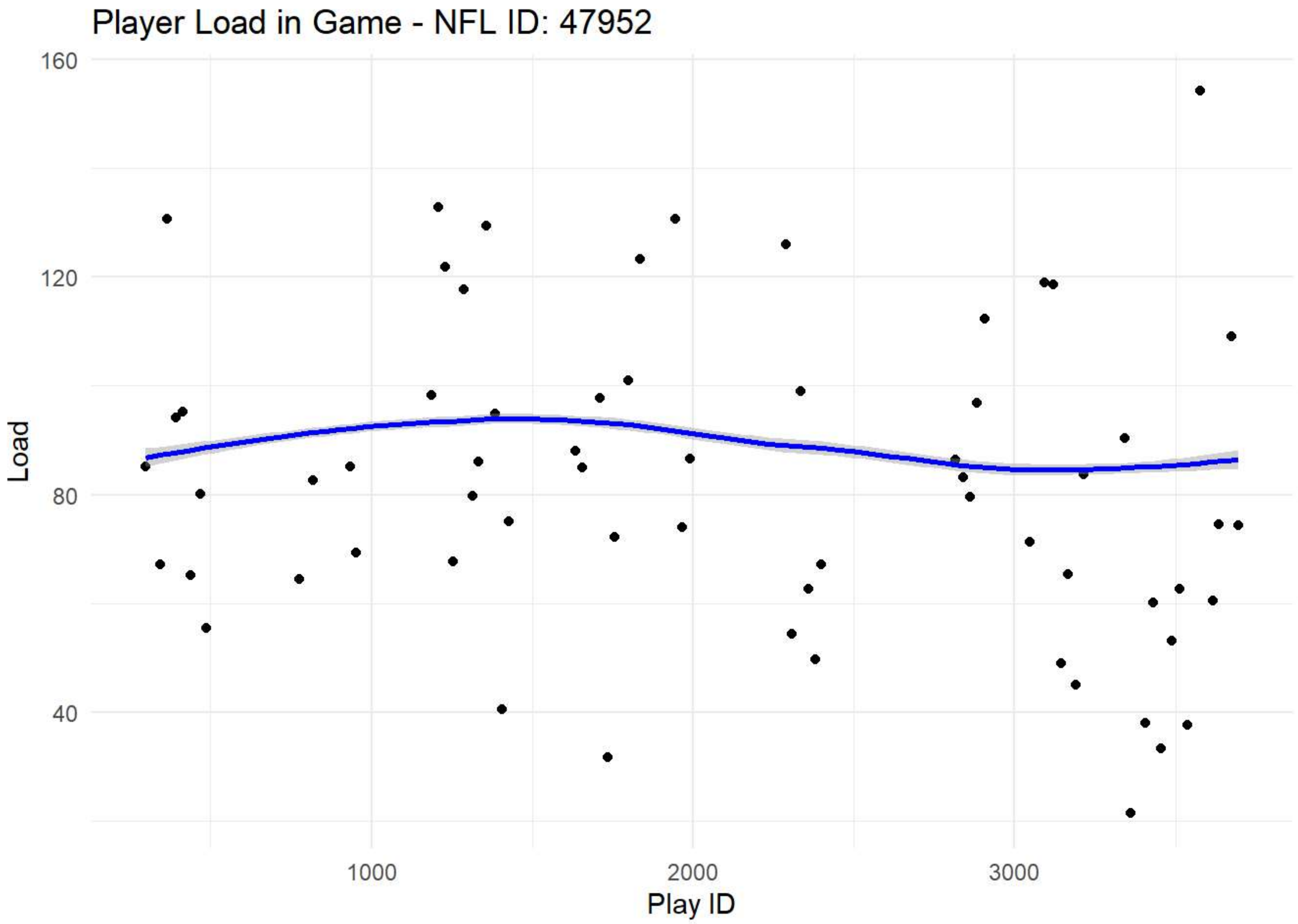
```
## `geom_smooth()` using formula = 'y ~ x'
```



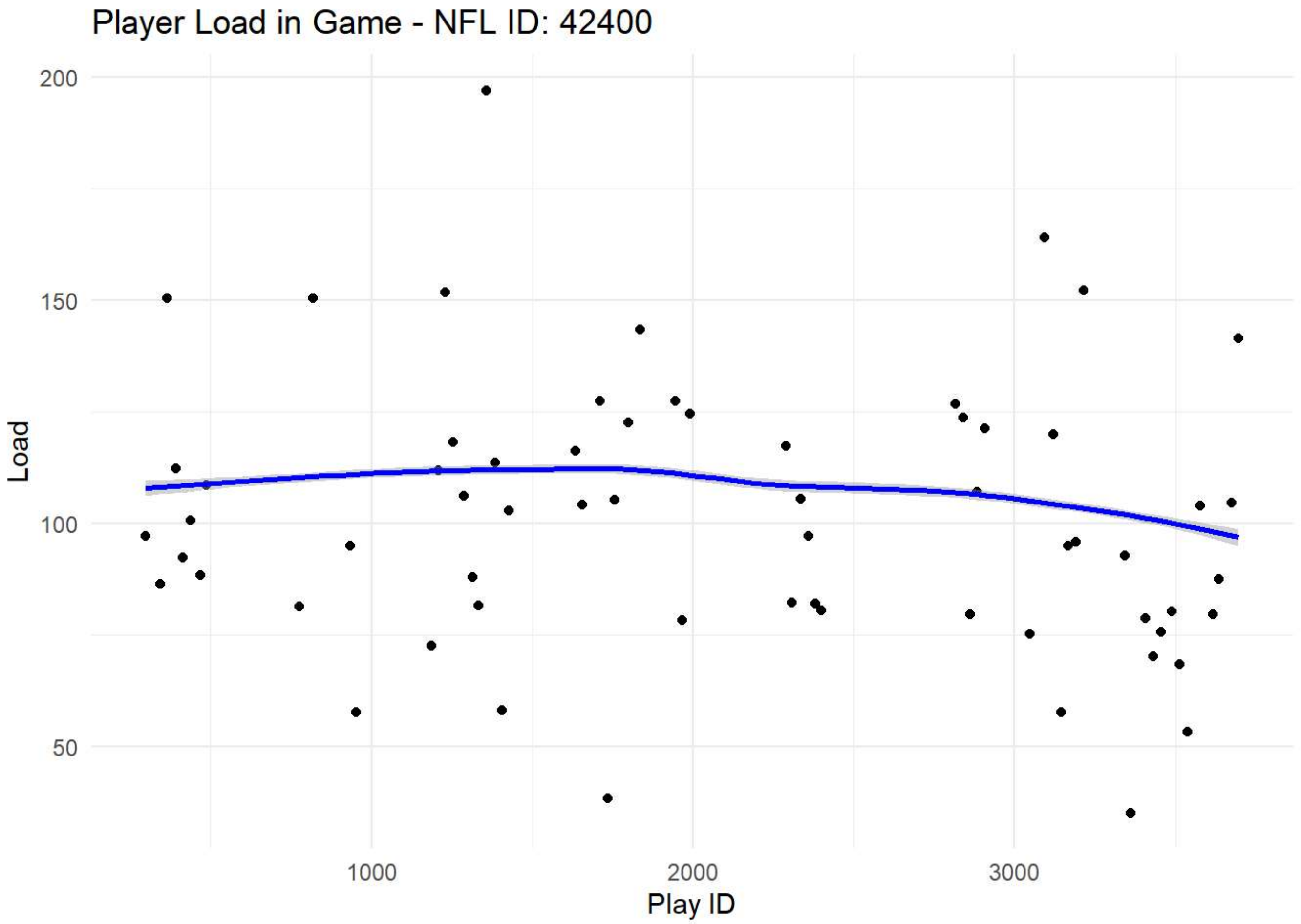
```
## `geom_smooth()` using formula = 'y ~ x'
```



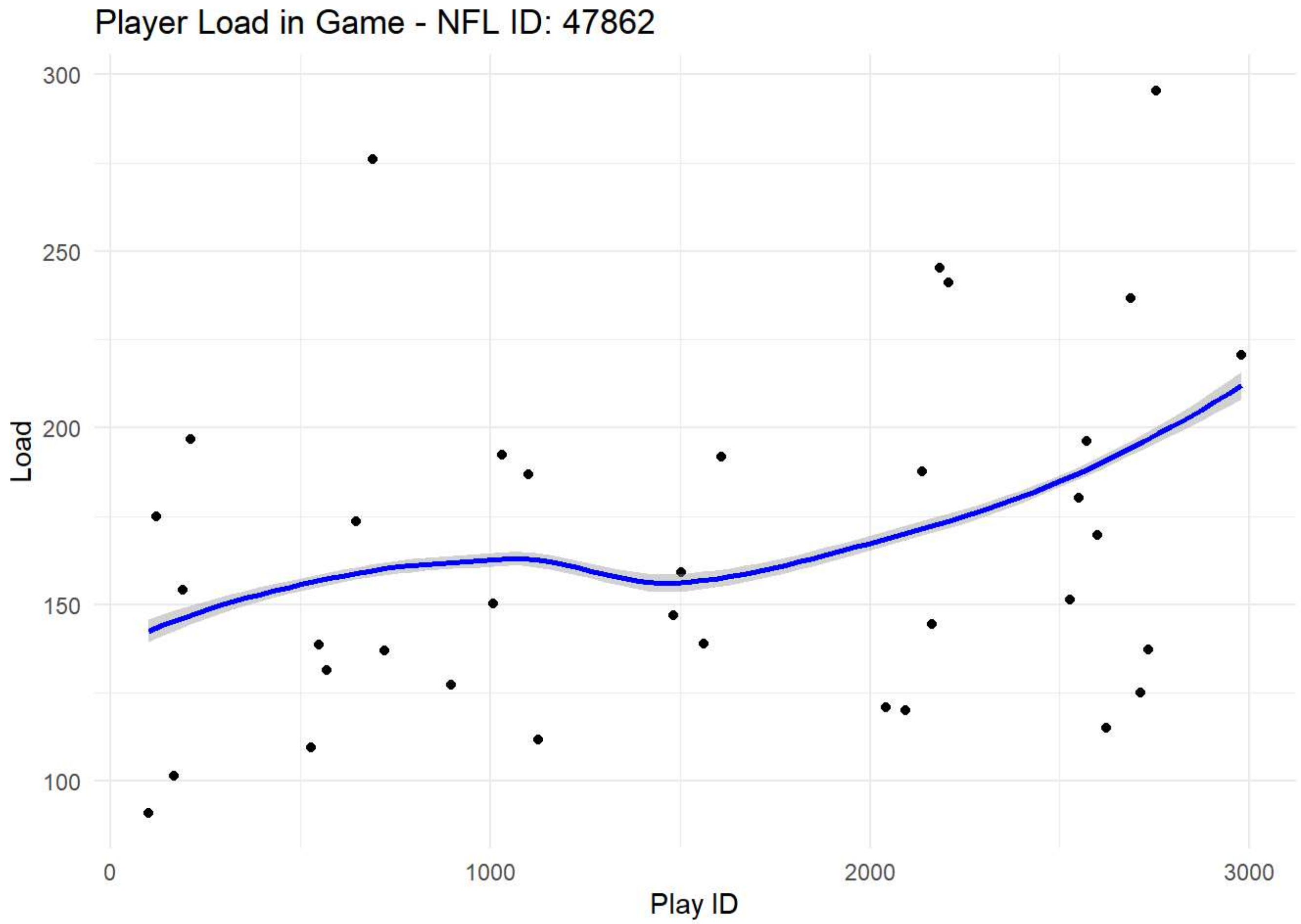
```
## `geom_smooth()` using formula = 'y ~ x'
```



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


Player Load in Game - NFL ID: 42400

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



We hypothesized that total load would decrease later in the game because a player would be more fatigues. For most of the linemen, we don’t see a decline in load throughout the game. These players either remain at a almost steady load value or increase slightly towards the end of the game. This trend could be the result of a break at halftime or a higher-intensity end to a game.

In addition to calculating total load using the sum of accelerations, we can also look at the change in the fatigue analysis of a player. This measure will incorporate both player speed and acceleration during the play.

With our fatigue analysis, we will also be looking at the outcome of the play. We have divided penalties into two main categories: tackle penalties (which happen during the play) and formation penalties (which happen before the play).

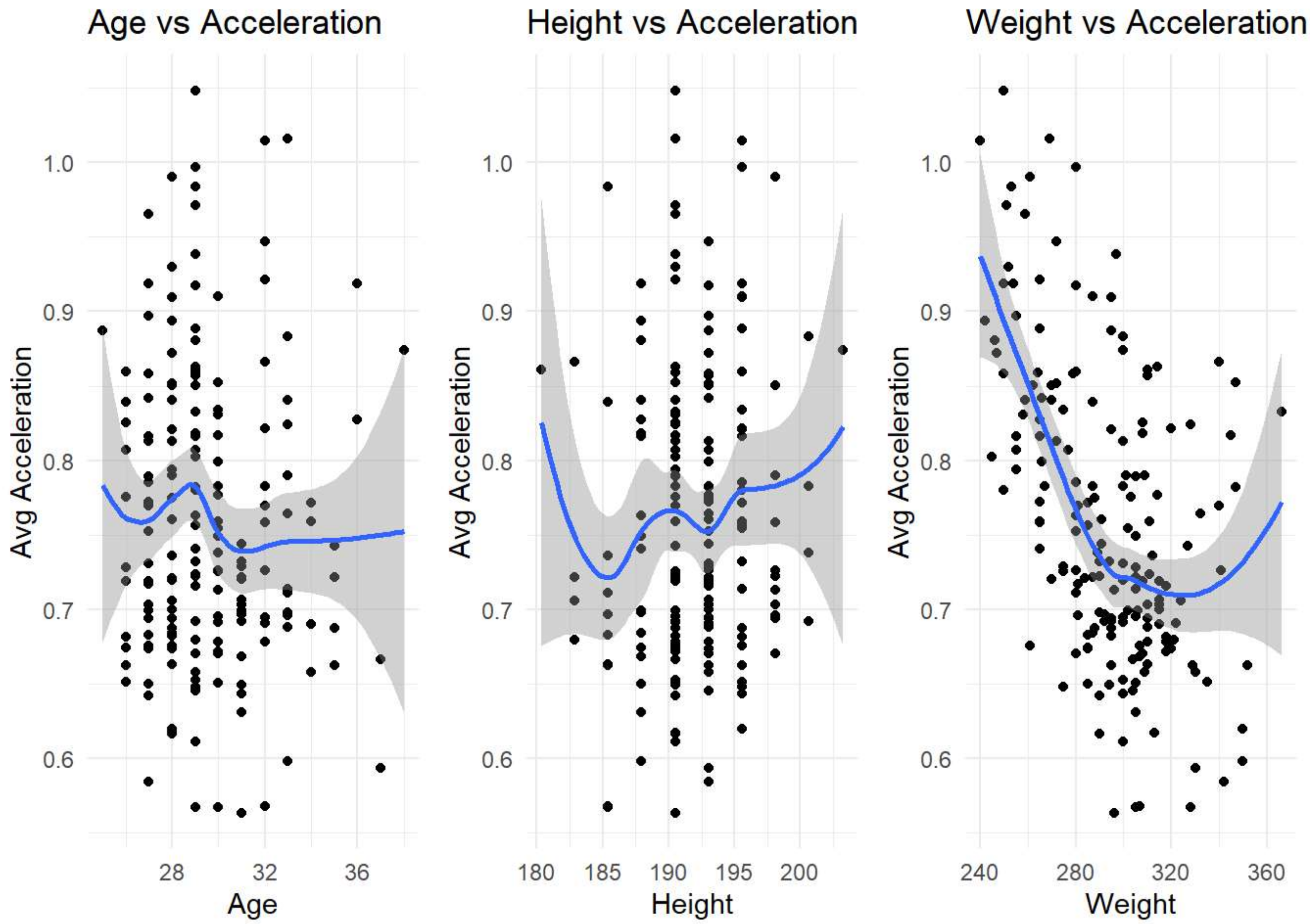
Effect of Demographics on Median Acceleration

```
max_acc <- data %>% group_by(nflId) %>% mutate(max_acc = mean(a)) %>% select(nflId, max_acc) %>% distinct()
demographic <- max_acc %>% inner_join(def_line_players)
```

```
## Joining with `by = join_by(nflId)`
```

```
demographic$birthDate <- as.Date(demographic$birthDate)



# Function to convert feet-inches to centimeters
convert_to_cm <- function(height)
  { height_parts <- strsplit(height, "-")[[1]]
  feet <- as.numeric(height_parts[1])
  inches <- as.numeric(height_parts[2])
  total_inches <- feet * 12 + inches
  total_cm <- total_inches * 2.54
  return(total_cm) } # Apply the conversion to the height column

demographic$height <- sapply(demographic$height, convert_to_cm)

demographic <- demographic %>% na.omit() %>% mutate(age = as.integer(round(as.Date(now()) - birthDate)/365))

# Scatter plot for age vs x
plot_age <- ggplot(demographic, aes(x = age, y = max_acc)) +
  geom_point() +
  labs(title = "Age vs Acceleration", x = "Age", y = "Avg Acceleration") +
  theme_minimal() +geom_smooth()

# Scatter plot for height vs x
plot_height <- ggplot(demographic, aes(x = height, y = max_acc)) +
  geom_point() +
  labs(title = "Height vs Acceleration", x = "Height", y = "Avg Acceleration") +
  theme_minimal() +geom_smooth()

# Scatter plot for weight vs x
plot_weight <- ggplot(demographic, aes(x = weight, y = max_acc)) +
  geom_point() +
  labs(title = "Weight vs Acceleration", x = "Weight", y = "Avg Acceleration") +
  theme_minimal() +geom_smooth()

grid.arrange(plot_age, plot_height, plot_weight, nrow = 1)
```

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



As one of the points of our analysis is to look for key factors that affect fatigue-resistant, we plotted the relationship between age, height and weight and median acceleration. As acceleration will be included in our derivation for “fatigue”, we thought it would be interesting to see these relationships.

We would have thought as players get older and larger that acceleration decreases. However this was not the case, with only weight showing a marginally negative correlation. This probably means that other factors such as match-up and time played/time since last TO are more important factors.

Penalties Throughout a Game

```
tackle_pen <- c("Horse Collar Tackle", "Defensive Holding", "Low Block", "Illegal Use of Hands",
"Roughing the Passer")
formation_pen <-  c("Illegal Shift", "Illegal Formation", "Defensive Offside")

play_details <- read.csv("plays.csv") %>% select(gameId, playId, quarter, gameClock)
head(play_details)
```

```
##       gameId playId quarter gameClock
## 1 2022102302   2655       3     01:54
## 2 2022091809   3698       4     02:13
## 3 2022103004   3146       4     02:00
## 4 2022110610    348       1     09:28
## 5 2022102700   2799       3     02:16
## 6 2022100205   2314       3     14:15
```

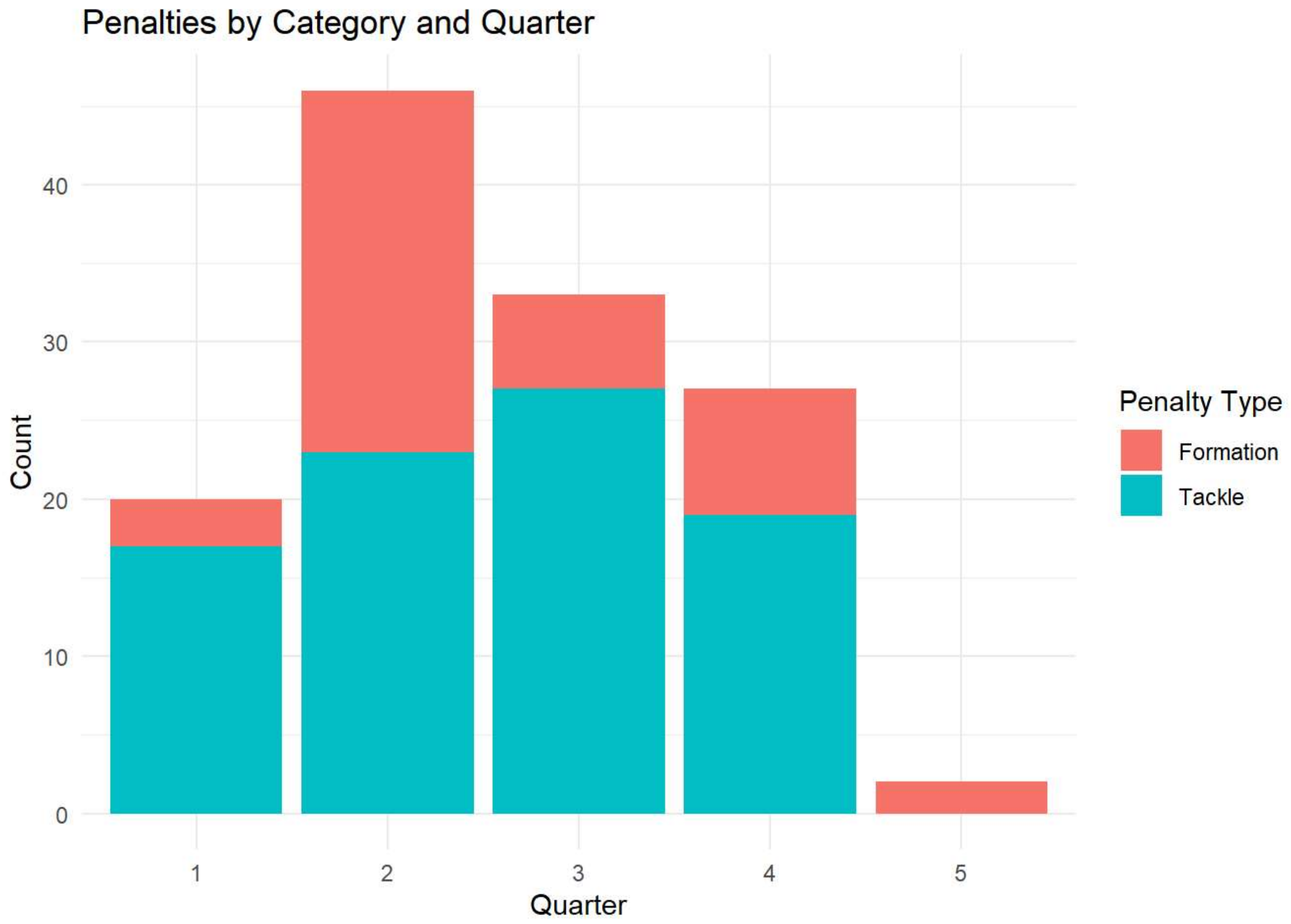
```
# penalties <- play %>% select(gameId, playId, nflId, penaltyNames) %>% full_join(play_details)
%>% mutate(penaltyType = ifelse(penaltyNames %in% tackle_pen, "Tackle", ifelse(penaltyNames %in%
formation_pen, "Formation", NA)), totalClock = (quarter-1) *  ms("15:00") +(ms("15:00")- ms(game
Clock)))


penalties <- play %>% select(gameId, playId, nflId, penaltyNames) %>% full_join(play_details) %
>% mutate(penaltyType = ifelse(penaltyNames %in% tackle_pen, "Tackle", ifelse(penaltyNames %in%
formation_pen, "Formation", NA)))
```

```
## Joining with `by = join_by(gameId, playId)`
```

```
plot_data <- penalties %>% select(quarter, penaltyType) %>% na.omit() %>% group_by(quarter, pena
ltyType) %>% summarise(count = n()) %>% ungroup()
```

```
## `summarise()` has grouped output by 'quarter'. You can override using the
## `.groups` argument.
```

```
stacked_bar <- ggplot(plot_data, aes(x = as.factor(quarter), y = count, fill = penaltyType)) +
  geom_bar(stat = "identity") +
  labs(title = "Penalties by Category and Quarter", x = "Quarter", y = "Count", fill = "Penalty
Type") +
  theme_minimal()

stacked_bar
```



```
# plot_data <- plot_data %>% na.omit()
# plot_data <- plot_data %>% pivot_wider(names_from = penaltyType, values_from = count, values_fill = list(count = 0))
#
# plot_data <- penalties %>% select(gameId, playId, penaltyType) %>% distinct() %>% arrange(gameId)
# unique(plot_data$playId)
#
# ggplot(data = plot_data) + geom_density(aes(x = totalClock, y = count, color = penaltyType))
# tail(penalties)
# unique(penalties$totalClock)
# ms("15:00")- ms(penalties$gameClock)
```

This doesn’t show exactly what we thought. There is an increase in both types of penalties from the first to the second quarter, and an increase in formation penalties from the third to fourth quarter. We would expect some type of reset after half time, so we will also look at fatigue by quarter. This is just a plot for week 1 data so it’s not totally representative of the season or the “NFL” in general.