# Draft Write-Up

Michael Li

April 12, 2021

```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v tibble  3.0.6     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(readr)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'

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

```
library(broom)
library(readxl)
library(gam)
```

```
## Loading required package: splines

## Loading required package: foreach

##
## Attaching package: 'foreach'

## The following objects are masked from 'package:purrr':
##
##     accumulate, when

## Loaded gam 1.16.1
```

```
library(mgcv)
```

```
## Loading required package: nlme

##
## Attaching package: 'nlme'

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

```
##
##     collapse
```

```
## This is mgcv 1.8-33. For overview type 'help("mgcv-package")'.
```

```
##
## Attaching package: 'mgcv'
```

```
## The following objects are masked from 'package:gam':
##
##     gam, gam.control, gam.fit, s
```

```r
players <- read_csv("data/players.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_character(),
##   career_AST = col_double(),
##   career_G = col_double(),
##   career_PTS = col_double(),
##   career_WS = col_double()
## )
## i Use `spec()` for the full column specifications.
```

```
## Warning: 2 parsing failures.
##  row       col expected actual               file
## 2274 career_WS a double      - 'data/players.csv'
## 4370 career_WS a double      - 'data/players.csv'
```

```r
salaries <- read_csv("data/salaries_1985to2018.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   league = col_character(),
##   player_id = col_character(),
##   salary = col_double(),
##   season = col_character(),
##   season_end = col_double(),
##   season_start = col_double(),
##   team = col_character()
## )
```

```r
salary_cap <- read_excel("data/salarycap.xlsx", col_names = c("season", "SalaryCap", "SalaryCap2019"))
```

```r
colnames(players)[1] <- "player_id"

teams <- salaries %>%
  group_by(player_id) %>%
  count(team) %>%
  mutate(years_with_team = max(n)) %>%
  subset(n == years_with_team) %>%
  slice(1) %>%
  select(player_id, team, years_with_team)

# df of aggregate salaries
agg_salaries <- salaries %>%
  group_by(player_id) %>%
  summarise(career_salary = sum(salary),
            career_start = min(season_start),
            career_end = max(season_end))

agg_salaries <- agg_salaries %>%
  merge(teams, by = "player_id")
```

```
df <- players %>%
  merge(agg_salaries, by = "player_id") %>%
  separate(col = birthDate, into = c("MonthDay", "birthYear"), sep = ", ") %>%
  separate(col = birthPlace, into = c("City", "birthPlace"), sep = ", ") %>%
  separate(col = draft_pick, into = c("draft_pick", "overall"), sep = "[thrdndst]") %>%
  separate(col = height, into = c("feet", "inches"), sep = "-") %>%
  mutate(height = as.double(feet) * 12 + as.double(inches)) %>%
  separate(col = position, into = c("primary_pos", "secondary_pos", "tertiary_pos", "quarternary_pos"),
           sep = " and ") %>%
  mutate(num_positions = if_else(is.na(primary_pos), 0, 1) +
           if_else(is.na(secondary_pos), 0, 1) +
           if_else(is.na(tertiary_pos), 0 , 1) +
           if_else(is.na(quarternary_pos), 0, 1)) %>%
  separate(col = weight, into = c("weight", "metric"), sep = "l") %>%
  select(-c(MonthDay, City, overall, draft_round, feet, inches, metric)) %>%
  mutate(years_played = career_end - career_start) %>%
  mutate(averageWS = career_WS / years_played)

#df$birthYear <- as.Date(df$birthYear, "%Y")
df$`career_FG%` <- as.double(df$`career_FG%`)
df$`career_FG3%` <- as.double(df$`career_FG3%`)
df$`career_FT%` <- as.double(df$`career_FT%`)
df$career_TRB <- as.double(df$career_TRB)
df$`career_eFG%` <- as.double(df$`career_eFG%`)
df$draft_year <- as.double(df$draft_year)
df$weight <- as.double(df$weight)
df$career_PER <- as.double(df$career_PER)
df$draft_pick <- as.integer(df$draft_pick)
#df$career_start <- as.Date(as.character(df$career_start), "%Y")
#df$career_end <- as.Date(as.character(df$career_end), "%Y")
df <- df %>%
  mutate(average_salary  = (career_salary / years_played)/1000000)
# %>%  # salary in millions
#   mutate(stand_salary = (average_salary - mean(average_salary, na.rm = TRUE)) / sd(average_salary, na
```

```
salary_cap_final <- salary_cap %>%
  separate(season, into = c("start", "end"), sep = "-")
salary_cap_final$start <- as.double(salary_cap_final$start)
salary_cap_final <- salary_cap_final %>%
  filter(start <= 2017) %>%
  mutate(SalaryCap = SalaryCap / 1000000) %>%
  select(start, SalaryCap)
```

```
avg_cap_finder <- function(start_year, end_year) {
  rel_cap <- salary_cap_final %>%
    filter(start >= start_year & start <= end_year - 1) %>%
    summarise(avg_cap = mean(SalaryCap))
  return(rel_cap$avg_cap[1])
}
avg_cap_finder <- Vectorize(avg_cap_finder)
df_final <- df %>%
  mutate(avg_cap = avg_cap_finder(career_start, career_end)) %>%
  mutate(sal_rel = (average_salary / avg_cap) * 100)
```

## Linear Model

```
# lm_sal <- lm(sal_rel ~ career_AST +
#              + `career_G` + `career_PER` + career_PTS + career_TRB + averageWS +
#                `career_eFG%` + draft_pick + primary_pos +
#                num_positions + draft_year,
#              data  = df_final)
#
# lm_sal_out <- tidy(lm_sal, conf.int = TRUE)
#
# lm_sal_out$term <- c(
#   "(Intercept)",
#   "APG", "Career Games", "PER", "PPG", "RPG",
#   "Win Shares", "eFG Percentage", "Draft Pick", "Primary Position = PG", "Primary Position = PF",
#   "Primary Position = SG", "Primary Position = SF", "Number of Positions", "Draft Year"
# )
#
# lm_sal_out <- lm_sal_out %>%
#   mutate(p_value = case_when(
#     p.value < 0.001 ~ "<0.001",
#     TRUE ~ as.character(round(p.value, digits = 3))
#   ))
#
# lm_sal_out <- lm_sal_out %>%
#   mutate(std.error = case_when(
#     std.error < 0.001 ~ "<0.001",
#     TRUE ~ as.character(round(std.error, digits = 3))
#   ))
#
# lm_sal_out <- lm_sal_out %>%
#   dplyr::select(term, estimate, std.error, statistic, p_value, conf.low, conf.high)
#
# knitr::kable(lm_sal_out, digits = 3, caption = "Average Standardized Salary OLS Model Output", col.na
#   kable_styling(latex_options = "HOLD_position")
# car::vif(lm_sal)
```

```
# lm_ws <- lm(averageWS ~ career_AST +
#              + `career_G` + `career_PER` + career_PTS + career_TRB +
#                `career_eFG%` + draft_pick + primary_pos +
#                num_positions + draft_year,
#              data  = df_final)
#
# lm_ws_out <- tidy(lm_ws, conf.int = TRUE)
```

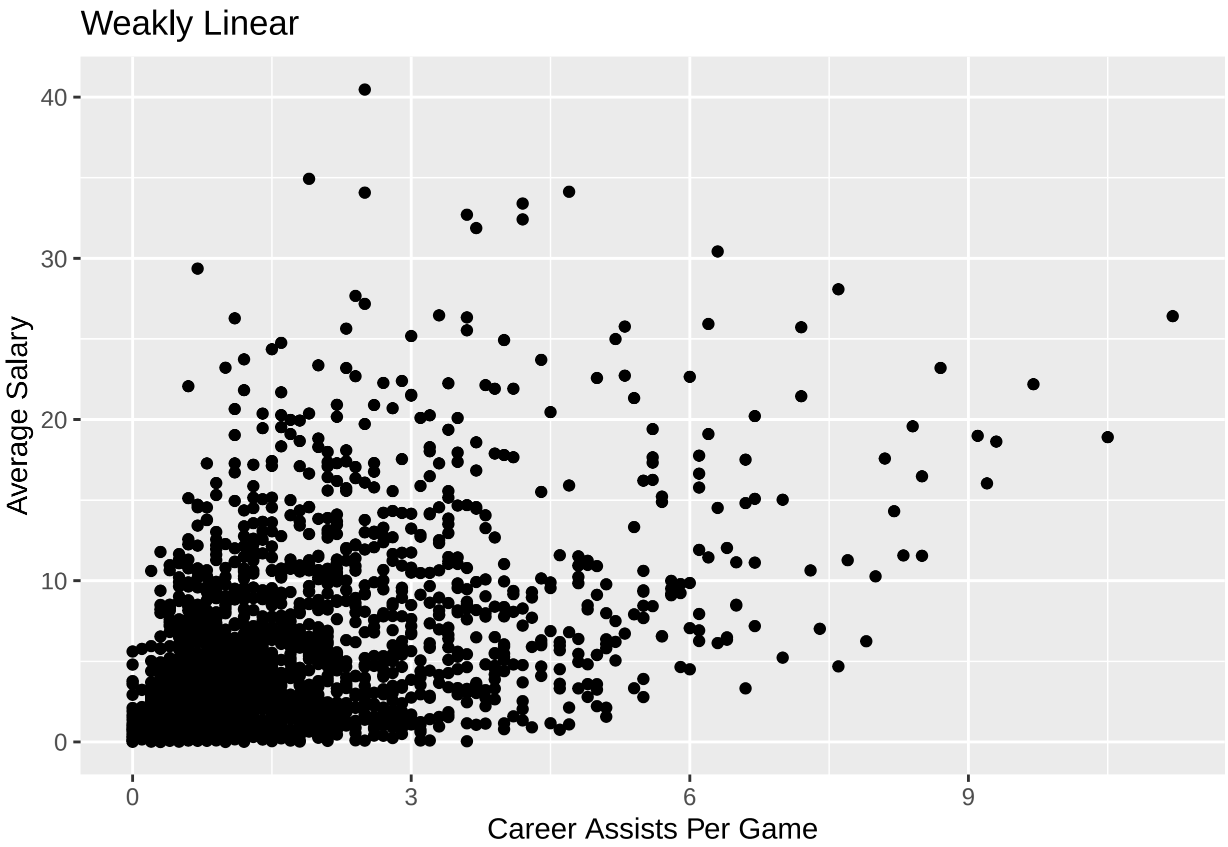
```
#
# lm_ws_out$term <- c(
#   "(Intercept)",
#   "APG", "Career Games", "PER", "PPG", "RPG",
#   "eFG Percentage", "Draft Pick", "Primary Position = PG", "Primary Position = PF",
#   "Primary Position = SG", "Primary Position = SF", "Number Of Positions", "Draft Year"
# )
#
# lm_ws_out <- lm_ws_out %>%
#   mutate(p_value = case_when(
#     p.value < 0.001 ~ "<0.001",
#     TRUE ~ as.character(round(p.value, digits = 3))
#   ))
#
# lm_ws_out <- lm_ws_out %>%
#   mutate(std.error = case_when(
#     std.error < 0.001 ~ "<0.001",
#     TRUE ~ as.character(round(std.error, digits = 3))
#   ))
#
# lm_ws_out <- lm_ws_out %>%
#   dplyr::select(term, estimate, std.error, statistic, p_value, conf.low, conf.high)
#
# knitr::kable(lm_ws_out, digits = 3, caption = "Average Win Shares OLS Model Output", col.names = c('T
#   kable_styling(latex_options = "HOLD_position")
# car::vif(lm_sal)
```

```
# temp_lm <- tibble(res = lm_sal$residuals,
#                   fitted = lm_sal$fitted.values)
# ggplot(data = temp_lm, aes(x = fitted, y = res)) +
#   geom_point() +
#   labs(x = "Fitted values", y = "Residuals",
#        title = "Salary Model: Highly non-constant variance") +
#   geom_hline(yintercept = 0, color = "red")
```

```
# temp_ws <- tibble(res = lm_ws$residuals,
#                   fitted = lm_ws$fitted.values)
# ggplot(data = temp_ws, aes(x = fitted, y = res)) +
#   geom_point() +
#   labs(x = "Fitted values", y = "Residuals",
#        title = "Win Shares Model: Highly non-constant variance") +
#   geom_hline(yintercept = 0, color = "red") +
#   ylim(c(-10, 75))
```

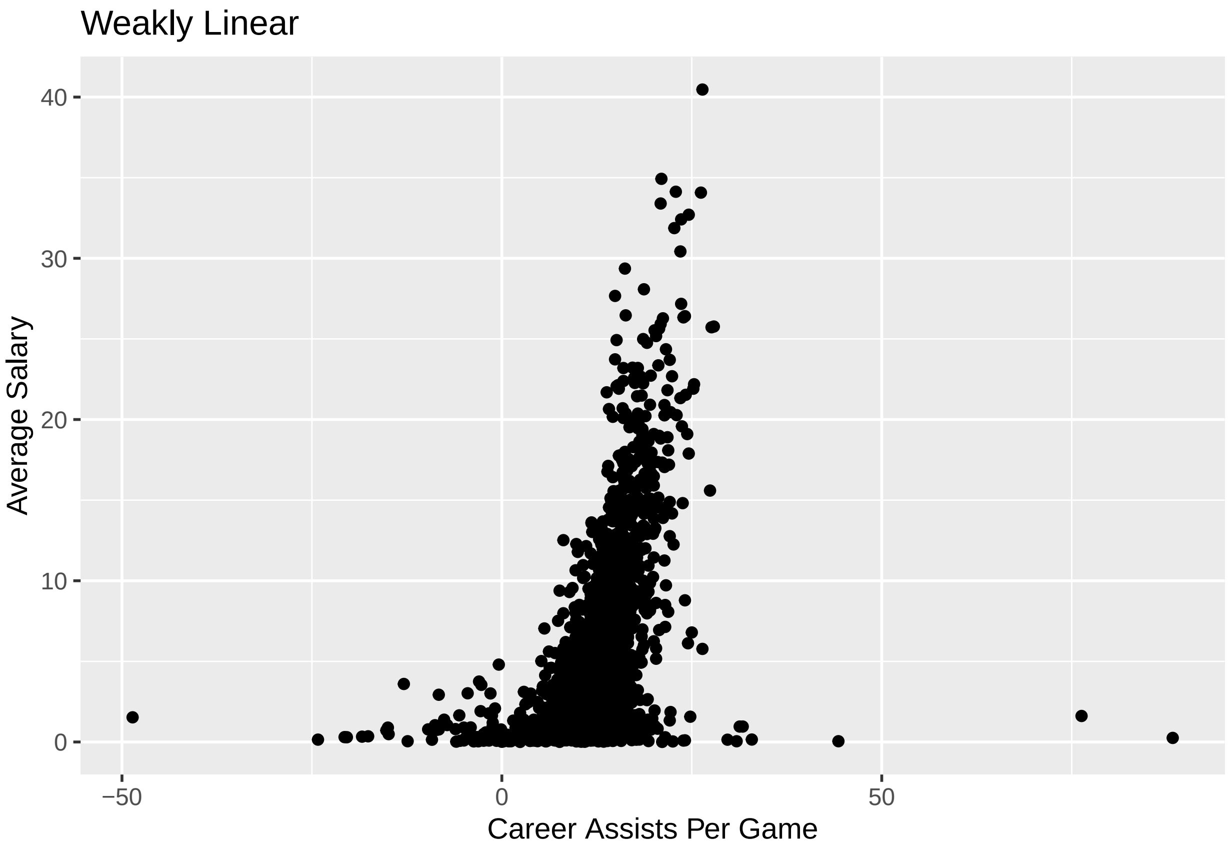
## EDA to investigate non-linearities

```
ggplot(data = df_final, aes(x = career_AST, y = sal_rel)) +
  geom_point() +
  labs(x = "Career Assists Per Game", y = "Average Salary",
       title = "Weakly Linear")
```

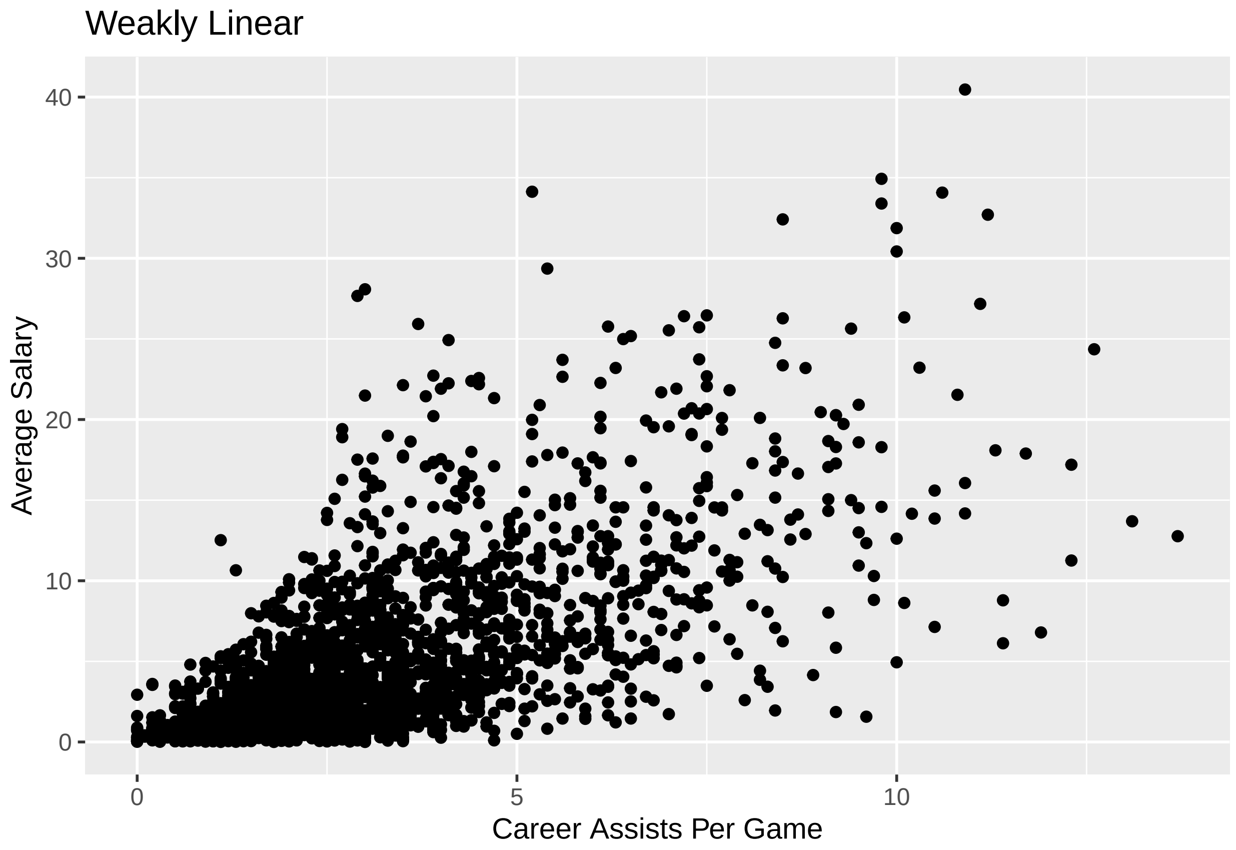


```
ggplot(data = df_final, aes(x = career_PER, y = sal_rel)) +
  geom_point() +
  labs(x = "Career Assists Per Game", y = "Average Salary",
       title = "Weakly Linear")
```

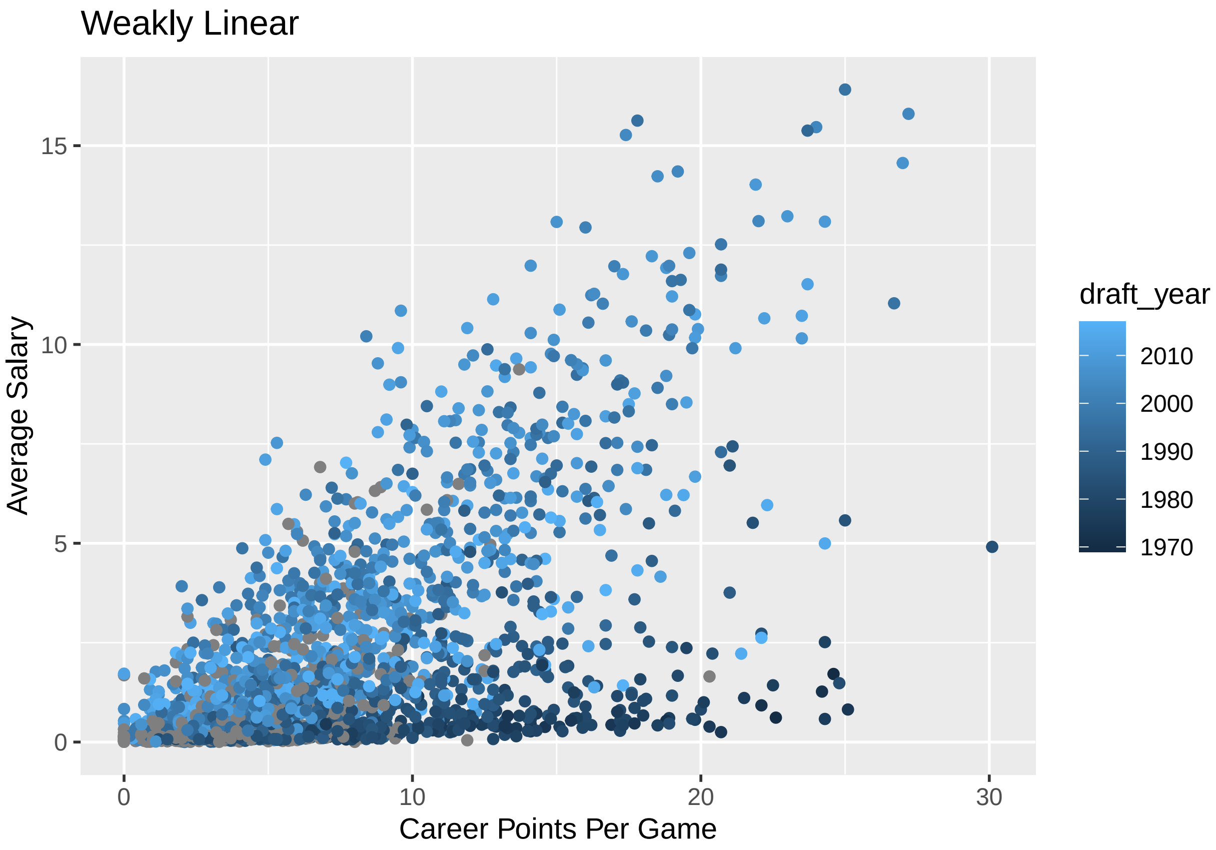
```
## Warning: Removed 2 rows containing missing values (geom_point).
```



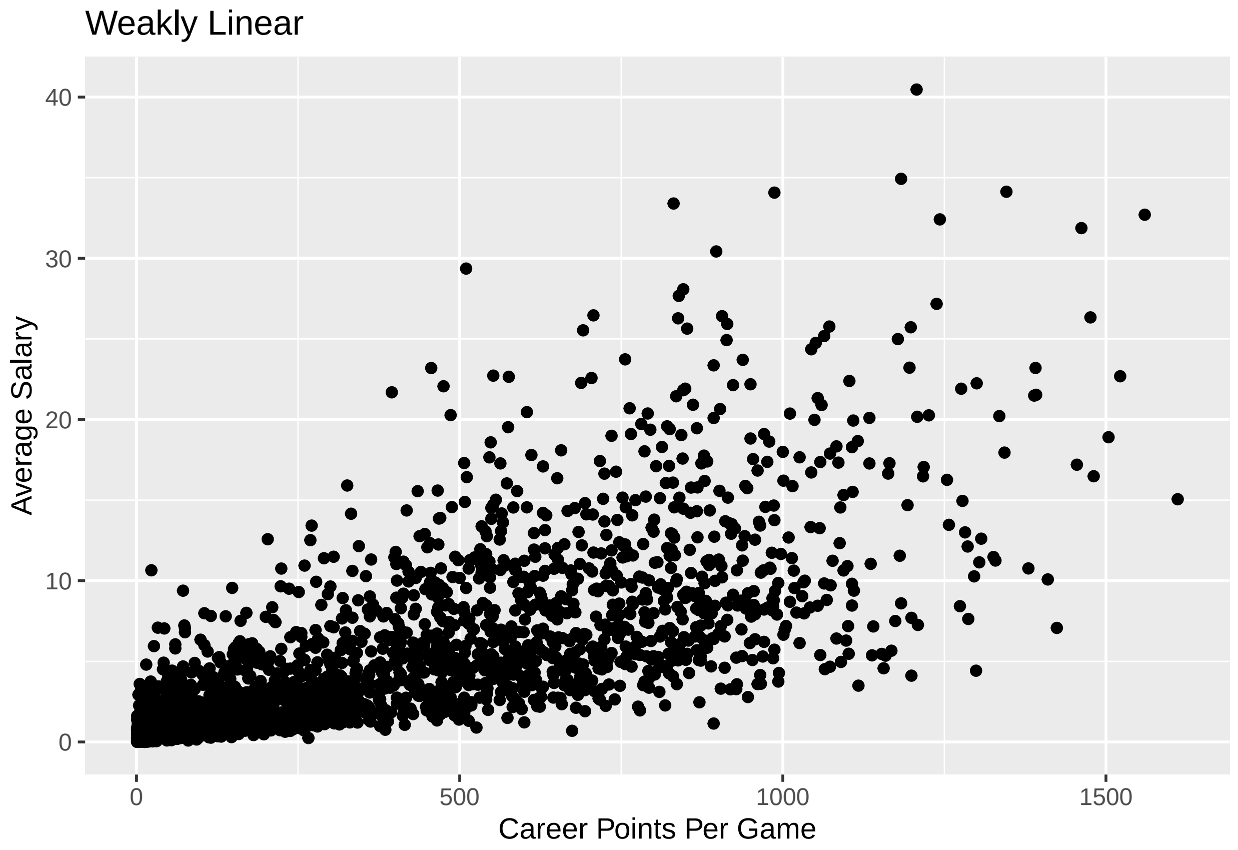
```r
ggplot(data = df_final, aes(x = career_TRB, y = sal_rel)) +
  geom_point() +
  labs(x = "Career Assists Per Game", y = "Average Salary",
       title = "Weakly Linear")
```



```
ggplot(data = df, aes(x = career_PTS, y = average_salary, colour = draft_year)) +
  geom_point() +
  labs(x = "Career Points Per Game", y = "Average Salary",
       title = "Weakly Linear")
```

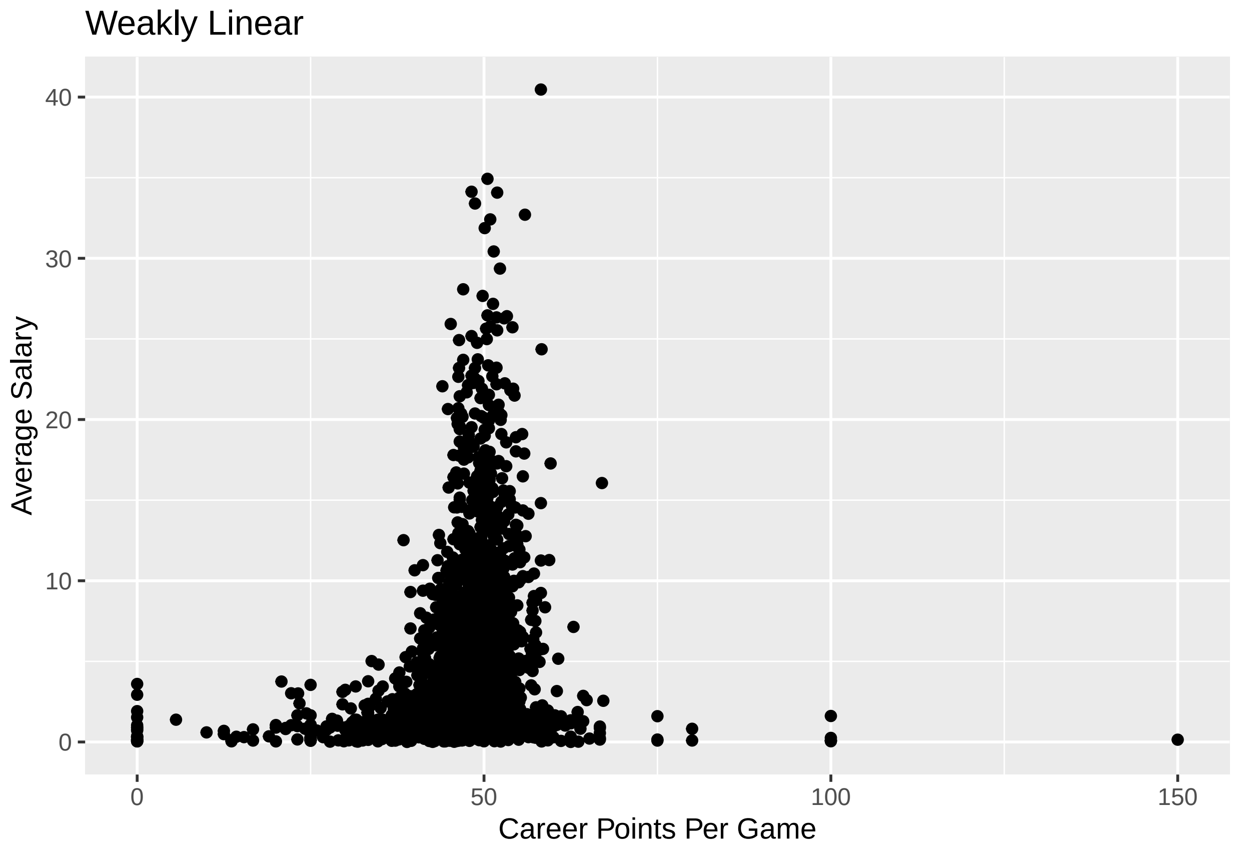


```r
ggplot(data = df_final, aes(x = career_G, y = sal_rel)) +
  geom_point() +
  labs(x = "Career Points Per Game", y = "Average Salary",
       title = "Weakly Linear")
```



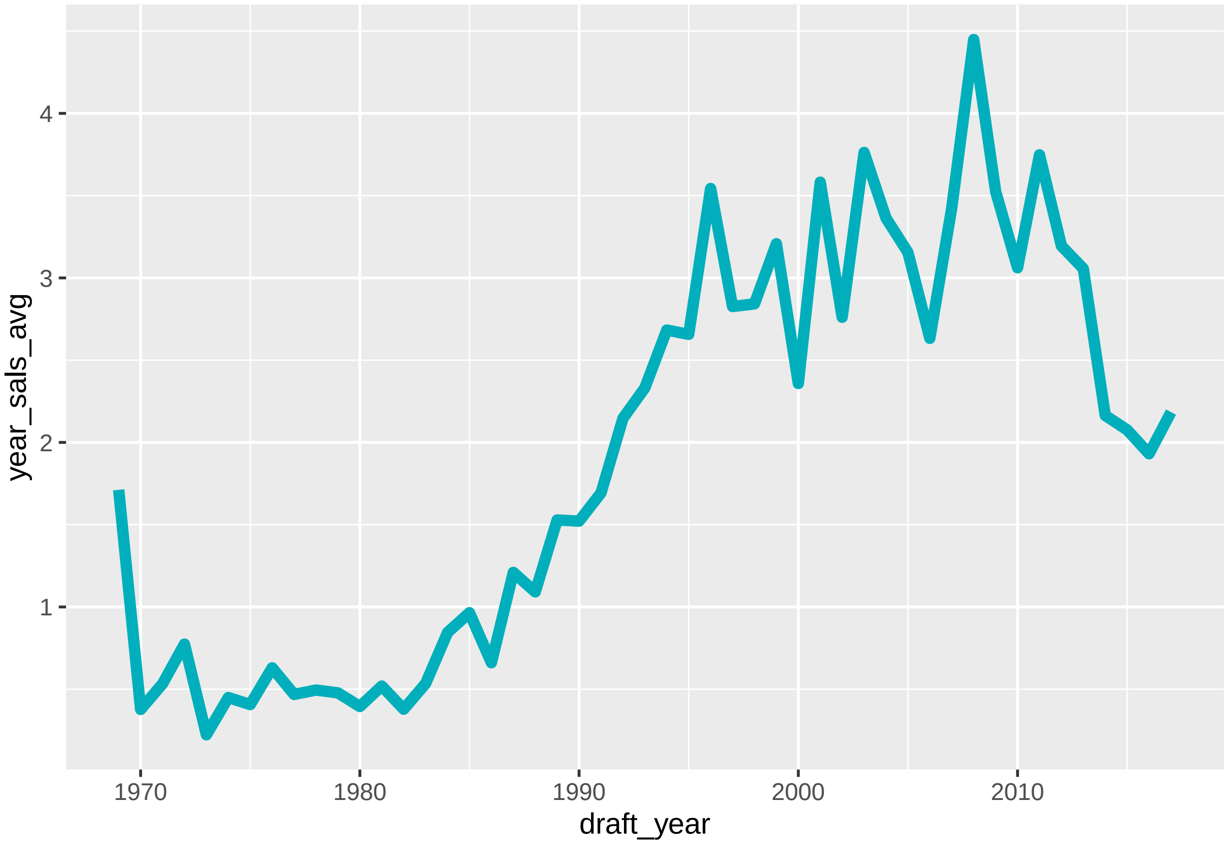
```
ggplot(data = df_final, aes(x = `career_eFG%`, y = sal_rel)) +
  geom_point() +
  labs(x = "Career Points Per Game", y = "Average Salary",
       title = "Weakly Linear")
```

```
## Warning: Removed 9 rows containing missing values (geom_point).
```



```
draft_year_sals <- df %>%
  group_by(draft_year) %>%
  mutate(year_sals_avg = mean(average_salary))
ggplot(data = draft_year_sals, aes(x = draft_year, y = year_sals_avg)) +
  geom_line(color = "#00AFBB", size = 2)
```

```
## Warning: Removed 524 row(s) containing missing values (geom_path).
```



```
gam_sal <- lm(sal_rel ~ bs(career_AST, df = 3, degree = 1) +
             + bs(`career_G`, df = 3, degree = 1) + bs(`career_PER`, df = 3, degree = 1) + bs(career_PTS
               bs(`career_eFG%`, df = 3, degree = 1) + bs(draft_pick, knots = c(14, 30), degree = 1) + p
               num_positions + draft_year, data = df_final)

gam_ws <- lm(averageWS ~ bs(career_AST, df = 3, degree = 1) +
             + bs(`career_G`, df = 3, degree = 1) + bs(`career_PER`, df = 3, degree = 1) + bs(career_PTS
               bs(`career_eFG%`, df = 3, degree = 1) + bs(draft_pick, knots = c(14, 30), degree = 1) + p
               num_positions + draft_year, data = df_final)
```

```
gam_sal_out <- tidy(gam_sal, conf.int = TRUE)
gam_sal_out$term <- c(
  "(Intercept)",
  "APG (B)", "APG (M)", "APG (T)", "Games Played (B)", "Games Played (M)",
  "Games Played (T)", "PER (B)", "PER (M)", "PER (T)", "PPG (B)", "PPG (M)",
  "PPG (T)", "RPG (B)", "RPG (M)", "RPG (T)", "Average Win Shares (B)",
  "Average Win Shares (M)", "Average Win Shares (T)", "Career eFG (B)",
  "Career eFG (M)", "Career eFG (T)", "Draft Pick (Lottery)", "Draft Pick (Late 1st)",
  "Draft Pick (2nd Round)", "Primary Position = PG", "Primary Position = PF",
  "Primary Position = SG", "Primary Position = SF", "Number of Positions",
  "Draft Year"
)

gam_sal_out <- gam_sal_out %>%
  mutate(p_value = case_when(
    p.value < 0.001 ~ "<0.001",
    TRUE ~ as.character(round(p.value, digits = 3))
  ))

gam_sal_out <- gam_sal_out %>%
  mutate(std.error = case_when(
    std.error < 0.001 ~ "<0.001",
    TRUE ~ as.character(round(std.error, digits = 3))
  ))

gam_sal_out <- gam_sal_out %>%
  dplyr::select(term, estimate, std.error, statistic, p_value, conf.low, conf.high)

knitr::kable(gam_sal_out, digits = 3, caption = "Relative Salary Score GAM Model Output", col.names = c
  kable_styling(latex_options = "HOLD_position")
```

Table 1: Relative Salary Score GAM Model Output

| Term | Estimate | Standard Error | Statistic | P-Value | CI (low) | CI (high) |
|---|---:|---|---:|---|---:|---:|
| (Intercept) | 17.230 | 14.476 | 1.190 | 0.234 | -11.161 | 45.621 |
| APG (B) | 0.080 | 0.459 | 0.175 | 0.861 | -0.819 | 0.980 |
| APG (M) | -0.428 | 0.488 | -0.876 | 0.381 | -1.385 | 0.530 |
| APG (T) | 3.955 | 1.014 | 3.899 | <0.001 | 1.966 | 5.945 |
| Games Played (B) | 0.241 | 0.445 | 0.542 | 0.588 | -0.631 | 1.113 |
| Games Played (M) | 3.417 | 0.496 | 6.895 | <0.001 | 2.445 | 4.388 |
| Games Played (T) | 8.687 | 0.704 | 12.343 | <0.001 | 7.307 | 10.067 |
| PER (B) | 1.348 | 2.376 | 0.567 | 0.571 | -3.312 | 6.009 |
| PER (M) | 1.318 | 2.34 | 0.563 | 0.573 | -3.272 | 5.908 |
| PER (T) | 17.871 | 4.995 | 3.578 | <0.001 | 8.075 | 27.668 |
| PPG (B) | 0.593 | 0.733 | 0.809 | 0.418 | -0.844 | 2.030 |
| PPG (M) | 0.539 | 0.783 | 0.689 | 0.491 | -0.996 | 2.075 |
| PPG (T) | 11.490 | 1.247 | 9.214 | <0.001 | 9.045 | 13.936 |
| RPG (B) | -0.900 | 0.598 | -1.504 | 0.133 | -2.073 | 0.274 |
| RPG (M) | -0.266 | 0.643 | -0.414 | 0.679 | -1.526 | 0.995 |
| RPG (T) | 6.324 | 1.052 | 6.013 | <0.001 | 4.261 | 8.386 |
| Average Win Shares (B) | -1.825 | 1.389 | -1.314 | 0.189 | -4.548 | 0.898 |
| Average Win Shares (M) | -2.557 | 1.393 | -1.835 | 0.067 | -5.290 | 0.176 |
| Average Win Shares (T) | -4.347 | 2.288 | -1.900 | 0.058 | -8.833 | 0.140 |
| Career eFG (B) | -0.518 | 1.175 | -0.441 | 0.659 | -2.823 | 1.786 |
| Career eFG (M) | -0.493 | 1.152 | -0.428 | 0.669 | -2.752 | 1.766 |
| Career eFG (T) | -8.196 | 3.199 | -2.562 | 0.01 | -14.470 | -1.923 |
| Draft Pick (Lottery) | -2.245 | 0.341 | -6.587 | <0.001 | -2.914 | -1.577 |
| Draft Pick (Late 1st) | -2.786 | 0.295 | -9.457 | <0.001 | -3.364 | -2.209 |
| Draft Pick (2nd Round) | -4.134 | 0.905 | -4.568 | <0.001 | -5.909 | -2.359 |
| Primary Position = PG | -0.940 | 0.405 | -2.321 | 0.02 | -1.734 | -0.146 |
| Primary Position = PF | -0.410 | 0.215 | -1.911 | 0.056 | -0.831 | 0.011 |
| Primary Position = SG | -0.290 | 0.322 | -0.903 | 0.366 | -0.921 | 0.340 |
| Primary Position = SF | -0.185 | 0.266 | -0.696 | 0.486 | -0.706 | 0.336 |
| Number of Positions | 0.054 | 0.135 | 0.398 | 0.691 | -0.211 | 0.319 |
| Draft Year | -0.006 | 0.007 | -0.833 | 0.405 | -0.020 | 0.008 |

```
summary(gam_ws)
```

```
##
```

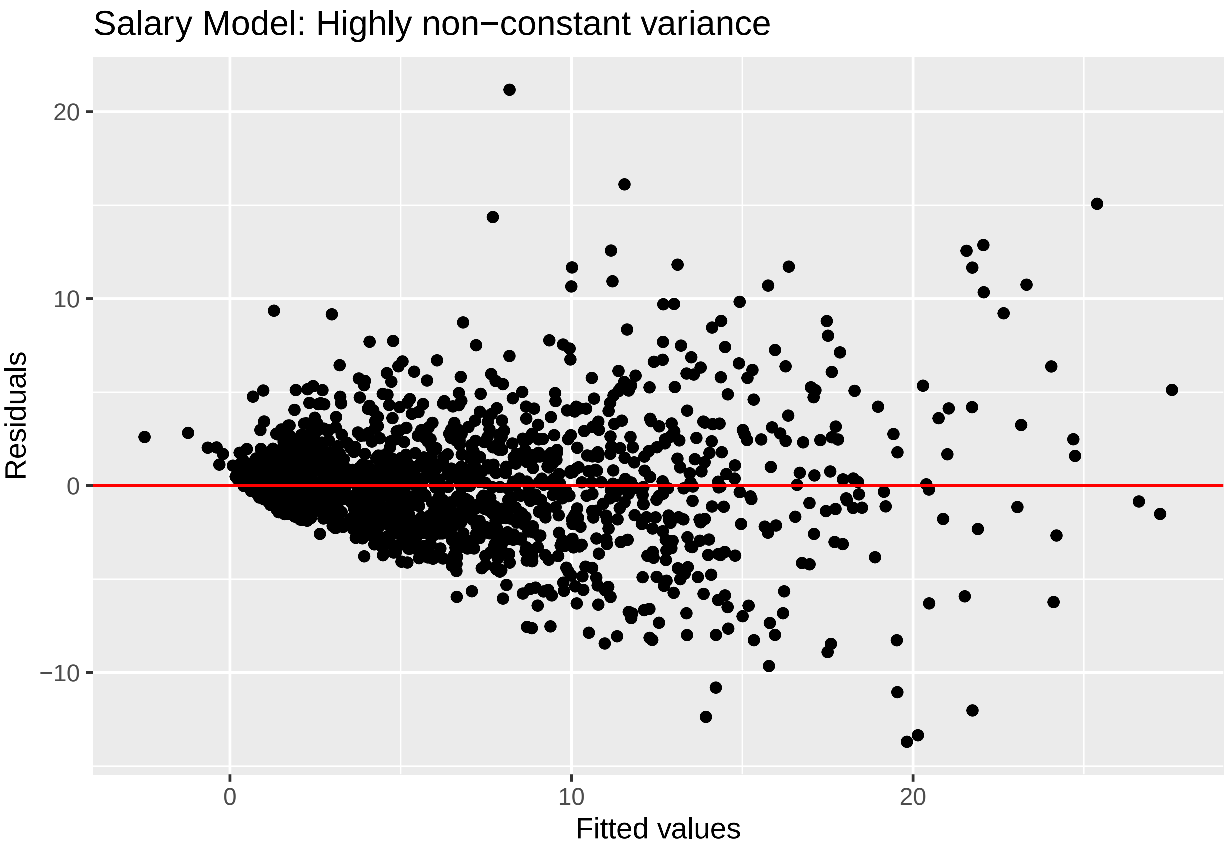
```
## Call:
## lm(formula = averageWS ~ bs(career_AST, df = 3, degree = 1) + 
##     +bs(career_G, df = 3, degree = 1) + bs(career_PER, df = 3, 
##     degree = 1) + bs(career_PTS, df = 3, degree = 1) + bs(career_TRB, 
##     df = 3, degree = 1) + bs(`career_eFG%`, df = 3, degree = 1) + 
##     bs(draft_pick, knots = c(14, 30), degree = 1) + primary_pos + 
##     num_positions + draft_year, data = df_final)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -11.791  -1.919  -0.315   1.071 133.971 
## 
## Coefficients:
##                                                  Estimate Std. Error t value
## (Intercept)                                     2.721e+02  2.806e+01   9.694
## bs(career_AST, df = 3, degree = 1)1             7.314e-01  9.102e-01   0.804
## bs(career_AST, df = 3, degree = 1)2             1.399e+00  9.634e-01   1.452
## bs(career_AST, df = 3, degree = 1)3             3.056e+00  2.000e+00   1.528
## bs(career_G, df = 3, degree = 1)1               2.212e-01  8.594e-01   0.257
## bs(career_G, df = 3, degree = 1)2              -2.080e-01  9.090e-01  -0.229
## bs(career_G, df = 3, degree = 1)3               1.840e+00  1.339e+00   1.374
## bs(career_PER, df = 3, degree = 1)1            -2.501e+00  4.717e+00  -0.530
## bs(career_PER, df = 3, degree = 1)2            -2.804e+00  4.645e+00  -0.604
## bs(career_PER, df = 3, degree = 1)3             1.404e+01  9.925e+00   1.414
## bs(career_PTS, df = 3, degree = 1)1            -5.909e-01  1.457e+00  -0.406
## bs(career_PTS, df = 3, degree = 1)2            -2.739e-01  1.556e+00  -0.176
## bs(career_PTS, df = 3, degree = 1)3             1.163e+01  2.466e+00   4.716
## bs(career_TRB, df = 3, degree = 1)1             1.923e+00  1.187e+00   1.621
## bs(career_TRB, df = 3, degree = 1)2             2.170e+00  1.268e+00   1.711
## bs(career_TRB, df = 3, degree = 1)3             1.022e+01  2.070e+00   4.939
## bs(`career_eFG%`, df = 3, degree = 1)1         -7.140e-01  2.330e+00  -0.306
## bs(`career_eFG%`, df = 3, degree = 1)2          4.900e-01  2.279e+00   0.215
## bs(`career_eFG%`, df = 3, degree = 1)3          8.521e+00  6.361e+00   1.340
## bs(draft_pick, knots = c(14, 30), degree = 1)1  2.762e+00  6.757e-01   4.087
## bs(draft_pick, knots = c(14, 30), degree = 1)2  2.747e+00  5.813e-01   4.725
## bs(draft_pick, knots = c(14, 30), degree = 1)3  7.475e+00  1.791e+00   4.174
## primary_posPoint Guard                         -3.653e-01  8.073e-01  -0.452
## primary_posPower Forward                       -3.962e-01  4.272e-01  -0.927
## primary_posShooting Guard                      -5.467e-04  6.403e-01  -0.001
## primary_posSmall Forward                       -1.434e-01  5.287e-01  -0.271
## num_positions                                  -4.701e-02  2.690e-01  -0.175
## draft_year                                     -1.368e-01  1.384e-02  -9.886
##                                                Pr(>|t|)    
## (Intercept)                                     < 2e-16 ***
## bs(career_AST, df = 3, degree = 1)1              0.4218    
## bs(career_AST, df = 3, degree = 1)2              0.1466    
## bs(career_AST, df = 3, degree = 1)3              0.1267    
## bs(career_G, df = 3, degree = 1)1                0.7969    
## bs(career_G, df = 3, degree = 1)2                0.8190    
## bs(career_G, df = 3, degree = 1)3                0.1696    
## bs(career_PER, df = 3, degree = 1)1              0.5961    
## bs(career_PER, df = 3, degree = 1)2              0.5461    
## bs(career_PER, df = 3, degree = 1)3              0.1575    
## bs(career_PTS, df = 3, degree = 1)1              0.6850    
## bs(career_PTS, df = 3, degree = 1)2              0.8603
## bs(career_PTS, df = 3, degree = 1)3            2.59e-06 ***
## bs(career_TRB, df = 3, degree = 1)1              0.1053
## bs(career_TRB, df = 3, degree = 1)2              0.0873 .
## bs(career_TRB, df = 3, degree = 1)3            8.56e-07 ***
## bs(`career_eFG%`, df = 3, degree = 1)1           0.7593
## bs(`career_eFG%`, df = 3, degree = 1)2           0.8298
## bs(`career_eFG%`, df = 3, degree = 1)3           0.1806
## bs(draft_pick, knots = c(14, 30), degree = 1)1 4.55e-05 ***
## bs(draft_pick, knots = c(14, 30), degree = 1)2 2.47e-06 ***
## bs(draft_pick, knots = c(14, 30), degree = 1)3 3.14e-05 ***
## primary_posPoint Guard                           0.6510
## primary_posPower Forward                         0.3538
## primary_posShooting Guard                        0.9993
## primary_posSmall Forward                         0.7862
## num_positions                                    0.8613
## draft_year                                      < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.776 on 1854 degrees of freedom
##   (526 observations deleted due to missingness)
## Multiple R-squared:  0.3361, Adjusted R-squared:  0.3264
## F-statistic: 34.76 on 27 and 1854 DF,  p-value: < 2.2e-16
```

```
gam_ws_out <- tidy(gam_ws, conf.int = TRUE)
gam_ws_out$term <- c(
  "(Intercept)",
  "APG (B)", "APG (M)", "APG (T)", "Games Played (B)", "Games Played (M)",
  "Games Played (T)", "PER (B)", "PER (M)", "PER (T)", "PPG (B)", "PPG (M)",
  "PPG (T)", "RPG (B)", "RPG (M)", "RPG (T)", "Career eFG (B)",
  "Career eFG (M)", "Career eFG (T)", "Draft Pick (Lottery)", "Draft Pick (Late 1st)",
  "Draft Pick (2nd Round)", "Primary Position = PG", "Primary Position = PF",
  "Primary Position = SG", "Primary Position = SF", "Number of Positions",
  "Draft Year"
)

gam_ws_out <- gam_ws_out %>%
  mutate(p_value = case_when(
    p.value < 0.001 ~ "<0.001",
    TRUE ~ as.character(round(p.value, digits = 3))
  ))

gam_ws_out <- gam_ws_out %>%
  mutate(std.error = case_when(
    std.error < 0.001 ~ "<0.001",
    TRUE ~ as.character(round(std.error, digits = 3))
  ))

gam_ws_out <- gam_ws_out %>%
  dplyr::select(term, estimate, std.error, statistic, p_value, conf.low, conf.high)

knitr::kable(gam_ws_out, digits = 3, caption = "Average Win Shares GAM Model Output", col.names = c('Te
  kable_styling(latex_options = "HOLD_position")
```

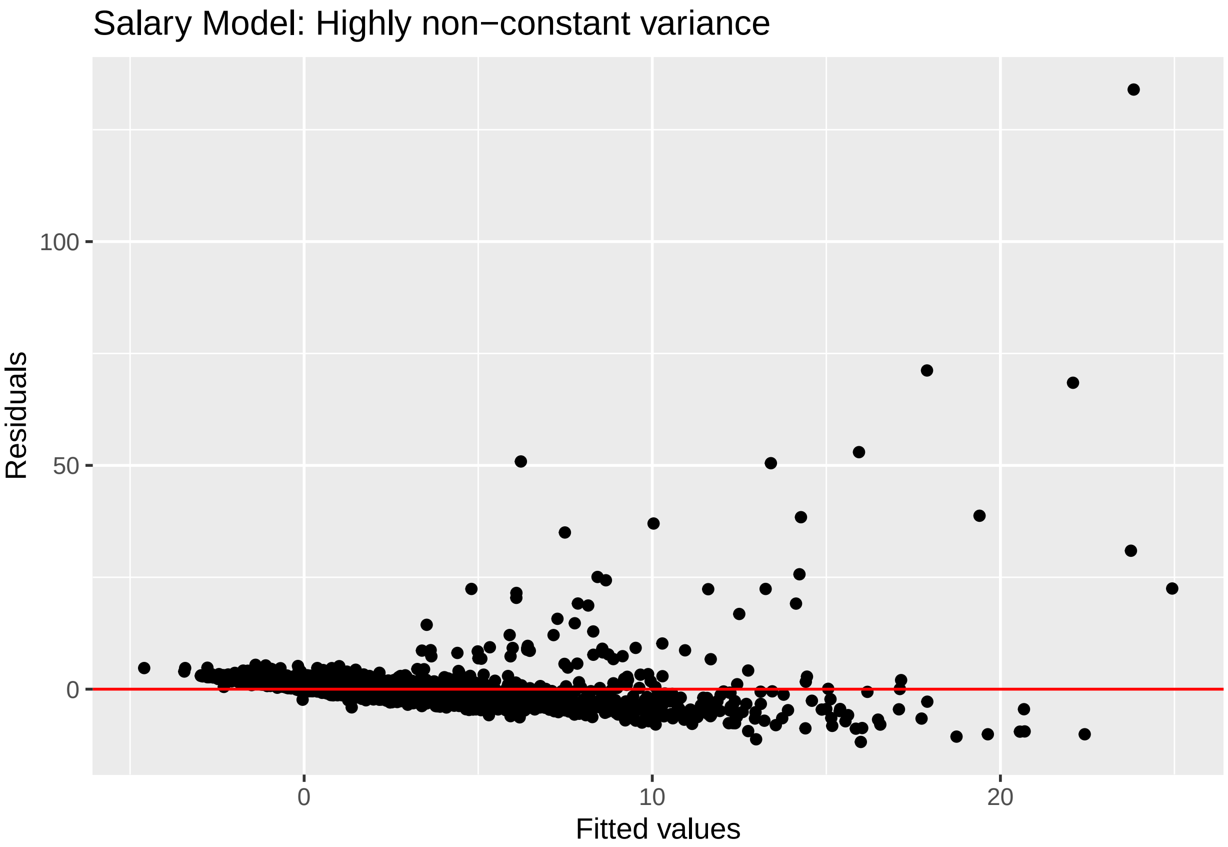
Table 2: Average Win Shares GAM Model Output

| Term | Estimate | Standard Error | Statistic | P-Value | CI (low) | CI (high) |
|---|---|---|---|---|---|---|
| (Intercept) | 272.067 | 28.065 | 9.694 | <0.001 | 217.025 | 327.109 |
| APG (B) | 0.731 | 0.91 | 0.804 | 0.422 | -1.054 | 2.516 |
| APG (M) | 1.399 | 0.963 | 1.452 | 0.147 | -0.490 | 3.289 |
| APG (T) | 3.056 | 2 | 1.528 | 0.127 | -0.866 | 6.978 |
| Games Played (B) | 0.221 | 0.859 | 0.257 | 0.797 | -1.464 | 1.907 |
| Games Played (M) | -0.208 | 0.909 | -0.229 | 0.819 | -1.991 | 1.575 |
| Games Played (T) | 1.840 | 1.339 | 1.374 | 0.17 | -0.786 | 4.466 |
| PER (B) | -2.501 | 4.717 | -0.530 | 0.596 | -11.752 | 6.751 |
| PER (M) | -2.804 | 4.645 | -0.604 | 0.546 | -11.914 | 6.305 |
| PER (T) | 14.036 | 9.925 | 1.414 | 0.157 | -5.430 | 33.502 |
| PPG (B) | -0.591 | 1.457 | -0.406 | 0.685 | -3.448 | 2.266 |
| PPG (M) | -0.274 | 1.556 | -0.176 | 0.86 | -3.326 | 2.778 |
| PPG (T) | 11.629 | 2.466 | 4.716 | <0.001 | 6.793 | 16.465 |
| RPG (B) | 1.923 | 1.187 | 1.621 | 0.105 | -0.404 | 4.250 |
| RPG (M) | 2.170 | 1.268 | 1.711 | 0.087 | -0.318 | 4.658 |
| RPG (T) | 10.222 | 2.07 | 4.939 | <0.001 | 6.163 | 14.281 |
| Career eFG (B) | -0.714 | 2.33 | -0.306 | 0.759 | -5.284 | 3.856 |
| Career eFG (M) | 0.490 | 2.279 | 0.215 | 0.83 | -3.980 | 4.960 |
| Career eFG (T) | 8.521 | 6.361 | 1.340 | 0.181 | -3.955 | 20.997 |
| Draft Pick (Lottery) | 2.762 | 0.676 | 4.087 | <0.001 | 1.437 | 4.087 |
| Draft Pick (Late 1st) | 2.747 | 0.581 | 4.725 | <0.001 | 1.607 | 3.887 |
| Draft Pick (2nd Round) | 7.475 | 1.791 | 4.174 | <0.001 | 3.962 | 10.987 |
| Primary Position = PG | -0.365 | 0.807 | -0.452 | 0.651 | -1.948 | 1.218 |
| Primary Position = PF | -0.396 | 0.427 | -0.927 | 0.354 | -1.234 | 0.442 |
| Primary Position = SG | -0.001 | 0.64 | -0.001 | 0.999 | -1.256 | 1.255 |
| Primary Position = SF | -0.143 | 0.529 | -0.271 | 0.786 | -1.180 | 0.894 |
| Number of Positions | -0.047 | 0.269 | -0.175 | 0.861 | -0.575 | 0.481 |
| Draft Year | -0.137 | 0.014 | -9.886 | <0.001 | -0.164 | -0.110 |

```r
temp_gam_sal <- tibble(res = gam_sal$residuals,
                       fitted = gam_sal$fitted.values)
ggplot(data = temp_gam_sal, aes(x = fitted, y = res)) +
  geom_point() +
  labs(x = "Fitted values", y = "Residuals",
       title = "Salary Model: Highly non-constant variance") +
  geom_hline(yintercept = 0, color = "red")
```

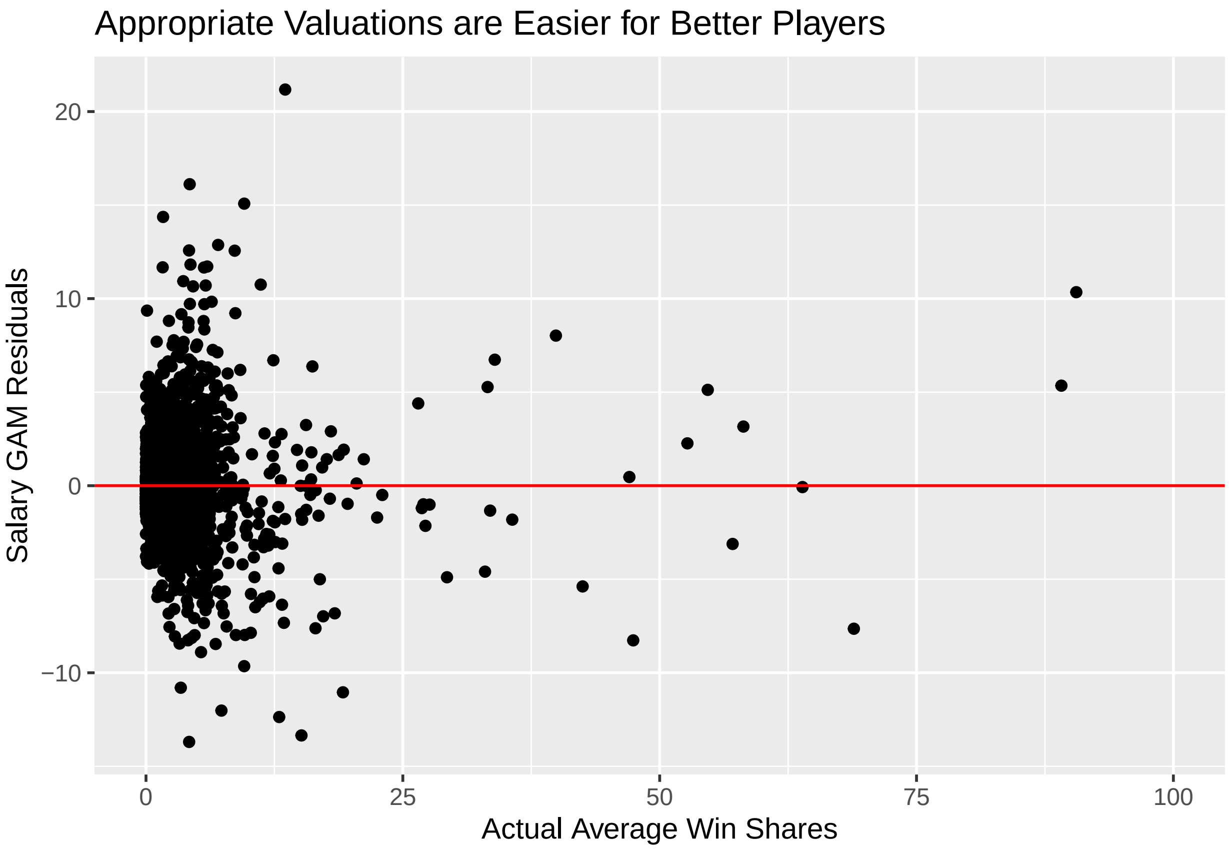


```
temp_gam_ws <- tibble(res = gam_ws$residuals,
                      fitted = gam_ws$fitted.values)
ggplot(data = temp_gam_ws, aes(x = fitted, y = res)) +
  geom_point() +
  labs(x = "Fitted values", y = "Residuals",
       title = "Salary Model: Highly non-constant variance") +
  geom_hline(yintercept = 0, color = "red")
```



```
# get rows with non NA WS values
nonNA_WS <- row.names(model.frame(gam_sal))
df_ws <- df_final[nonNA_WS,]

temp_gam_sal_ws <- tibble(ws = df_ws$averageWS,
                          res = gam_sal$residuals,
                          fitted = gam_sal$fitted.values)
ggplot(data = temp_gam_sal_ws, aes(x = ws, y = res)) +
  geom_point() +
  labs(x = "Actual Average Win Shares", y = "Salary GAM Residuals",
       title = "Appropriate Valuations are Easier for Better Players") +
  geom_hline(yintercept = 0, color = "red") +
  xlim(0, 100)
```

```
## Warning: Removed 167 rows containing missing values (geom_point).
```



```
# summary(gam_sal)
# summary(gam_ws)
# gam.check(gam_sal)
# plot(gam_sal)
# plot(gam_ws)
# plot.Gam(gam_sal, se = TRUE, col = "red")
```

$$\begin{aligned}
Salary\ Score_i = &\ \beta_0 + \beta_1 f_1(APG_i) + \beta_2 f_2(Games\ Played_i) + \beta_3 f_3(PER_i) \\
&+ \beta_4 f_4(PPG_i) + \beta_5 f_5(RPG_i) + \beta_6 f_6(Career\ eFG_i) \\
&+ \beta_7 f_7(Draft\ Pick_i) + \beta_8 I(Primary\ Position_i = PG) \\
&+ \beta_9 I(Primary\ Position_i = PF) + \beta_{10} I(Primary\ Position_i = SG) \\
&+ \beta 11 I(Primary\ Position_i = SF) + \beta_{12} Number\ of\ Positions_i \\
&+ \beta 13 Draft\ Year_i +\ \epsilon_i \\
&for\ i = 1,\ 2,\ ...,\ n
\end{aligned}$$

Each spline $f_j$ takes the following form for $j = 1,\ 2,\ ...\ ,\ 7$:

$f_j(x_{ij}) = b_0 + b_1(x_{ij}) + b_2(x_{ij}) + b3(x_{ij}) + \epsilon_{ij}$

where we have

$b_1(x_{ij}) = x_{ij}$

$b_2(x_{ij}) = (x_{ij}\ \ -\ \ \xi_2)_+$

$b_3(x_{ij}) = (x_{ij}\ \ -\ \ \xi_3)_+$

and

$$(x_{ij} - \xi_k)_+ = \begin{cases} x_{ij} - \xi_k & \text{if } x_{ij} \geq \xi_k \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{aligned}
Win\ Shares_i = &\ \beta_0 + \beta_1 f_1(APG_i) + \beta_2 f_2(Games\ Played_i) + \beta_3 f_3(PER_i) \\
&+ \beta_4 f_4(PPG_i) + \beta_5 f_5(RPG_i) + \beta_6 f_6(Draft\ Pick_i) \\
&+ \beta_7 I(Primary\ Position_i = PG) + \beta_8 I(Primary\ Position_i = PF) \\
&+ \beta_9 I(Primary\ Position_i = SG) + \beta 10 I(Primary\ Position_i = SF) \\
&+ \beta_{11} Number\ of\ Positions_i + \beta 12 Draft\ Year_i +\ \epsilon_i \\
&for\ i = 1,\ 2,\ ...,\ n
\end{aligned}$$

Each spline $f_j$ takes the following form for $j = 1,\ 2,\ ...\ ,\ 6$:

$f_j(x_{ij}) = b_0 + b_1(x_{ij}) + b_2(x_{ij}) + b3(x_{ij}) + \epsilon_{ij}$

where we have

$b_1(x_{ij}) = x_{ij}$

$b_2(x_{ij}) = (x_{ij}\ \ -\ \ \xi_2)_+$

$b_3(x_{ij}) = (x_{ij}\ \ -\ \ \xi_3)_+$

and

$$(x_{ij} - \xi_k)_+ = \begin{cases} x_{ij} - \xi_k & \text{if } x_{ij} \geq \xi_k \\ 0 & \text{otherwise} \end{cases}$$