Second Half

April 6, 2022

[1]:
```
install.packages("qqplotr")
```

```
Installing package into ‘/home/jupyter/R/x86_64-pc-linux-gnu-library/4.1’
(as ‘lib’ is unspecified)
```

[2]:
```
install.packages("leaps")
```

```
Installing package into ‘/home/jupyter/R/x86_64-pc-linux-gnu-library/4.1’
(as ‘lib’ is unspecified)
```

[1]:
```
library(ggplot2)
library(qqplotr)
library(tidyverse)
library(repr)
library(tidymodels)
library(stringr)
library(leaps)
```

```
Attaching package: ‘qqplotr’


The following objects are masked from ‘package:ggplot2’:

    stat_qq_line, StatQqLine


Warning message in system("timedatectl", intern = TRUE):
"running command 'timedatectl' had status 1"
  Attaching packages                      tidyverse
1.3.1

  tibble  3.1.6      dplyr   1.0.8
  tidyr   1.2.0      stringr 1.4.0
  readr   2.1.2      forcats 0.5.1
  purrr   0.3.4
```

```
── Conflicts ──────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter()         masks stats::filter()
✖ dplyr::lag()            masks stats::lag()
✖ qqplotr::stat_qq_line() masks ggplot2::stat_qq_line()

── Attaching packages ─────────────────────────────── tidymodels 0.2.0 ──

✔ broom        0.7.12      ✔ rsample      0.1.1
✔ dials        0.1.0       ✔ tune         0.2.0
✔ infer        1.0.0       ✔ workflows    0.2.6
✔ modeldata    0.1.1       ✔ workflowsets 0.2.1
✔ parsnip      0.2.1       ✔ yardstick    0.0.9
✔ recipes      0.2.0

── Conflicts ─────────────────────────────────────── tidymodels_conflicts() ──
✖ scales::discard()       masks purrr::discard()
✖ dplyr::filter()         masks stats::filter()
✖ recipes::fixed()        masks stringr::fixed()
✖ dplyr::lag()            masks stats::lag()
✖ yardstick::spec()       masks readr::spec()
✖ qqplotr::stat_qq_line() masks ggplot2::stat_qq_line()
✖ recipes::step()         masks stats::step()
• Use suppressPackageStartupMessages() to eliminate package startup
messages
```

[2]:
```
data <- read.csv(url("https://drive.google.com/uc?
↪export=download&id=1_MECmUXZuuILYeEOfonSGqodW6qVdhsS"))
```

[3]:
```
data <- mutate(data, str_replace(data$Age, " \\s*\\([^\\)]+\\)", ""))
data <- mutate(data, Age = str_replace(data$Age, " \\s*\\([^\\)]+\\)", ""))
data <- mutate(data, Current.Rank = str_replace(data$Current.Rank, "␣
 ↪\\s*\\([^\\)]+\\)", ""))
data <- mutate(data, Best.Rank = str_replace(data$Best.Rank, "␣
 ↪\\s*\\([^\\)]+\\)", ""))

money <- c(data$Prize.Money)
money <- money %>%
        lapply(gsub, pattern="$", fixed=TRUE, replacement="") %>%
        lapply(gsub, pattern="US", fixed=TRUE, replacement="") %>%
        lapply(gsub, pattern="all-time leader in earnings", fixed=TRUE,␣
 ↪replacement="") %>%
        lapply(gsub, pattern="All-time leader in earnings", fixed=TRUE,␣
 ↪replacement="") %>%
        lapply(gsub, pattern="all-time in earnings", fixed=TRUE,␣
 ↪replacement="") %>%
        lapply(gsub, pattern="11th", fixed=TRUE, replacement="") %>%
        lapply(gsub, pattern="24th", fixed=TRUE, replacement="") %>%
        lapply(gsub, pattern="10th", fixed=TRUE, replacement="") %>%
        lapply(gsub, pattern="14th", fixed=TRUE, replacement="") %>%
        lapply(gsub, pattern="2nd", fixed=TRUE, replacement="") %>%
        lapply(gsub, pattern="27th", fixed=TRUE, replacement="") %>%
        lapply(gsub, pattern="15th", fixed=TRUE, replacement="") %>%
        lapply(gsub, pattern="30th", fixed=TRUE, replacement="") %>%
        lapply(gsub, pattern="4th", fixed=TRUE, replacement="") %>%
        lapply(gsub, pattern="28th", fixed=TRUE, replacement="") %>%
        lapply(gsub, pattern="6th", fixed=TRUE, replacement="") %>%
        lapply(gsub, pattern="33rd", fixed=TRUE, replacement="") %>%
        lapply(gsub, pattern="26th", fixed=TRUE, replacement="") %>%
        lapply(gsub, pattern="24th", fixed=TRUE, replacement="") %>%
        lapply(gsub, pattern="48th", fixed=TRUE, replacement="") %>%
        lapply(gsub, pattern="41st", fixed=TRUE, replacement="") %>%
        lapply(gsub, pattern="24th", fixed=TRUE, replacement="") %>%
        lapply(gsub, pattern="15th", fixed=TRUE, replacement="")

data_selected <- data %>%
            mutate(data, Prize.Money = money) %>%
            select(Age, Name,Country, Current.Rank, Best.Rank, Prize.
 ↪Money,Seasons) %>%
            mutate(Prize.Money = gsub(",","", Prize.Money))

tidy_data <- data_selected %>%
        filter(Prize.Money != "") %>%
        mutate(Prize.Money = as.numeric(Prize.Money)) %>%
        mutate(Age = as.numeric(Age)) %>%
        mutate(Current.Rank = as.numeric(Current.Rank)) %>%
```

```
        mutate(Best.Rank = as.numeric(Best.Rank)) %>%
        mutate(Seasons = as.numeric(Seasons))

tidy_data <- drop_na(tidy_data)
```

```
Warning message in mask$eval_all_mutate(quo):
"NAs introduced by coercion"
```

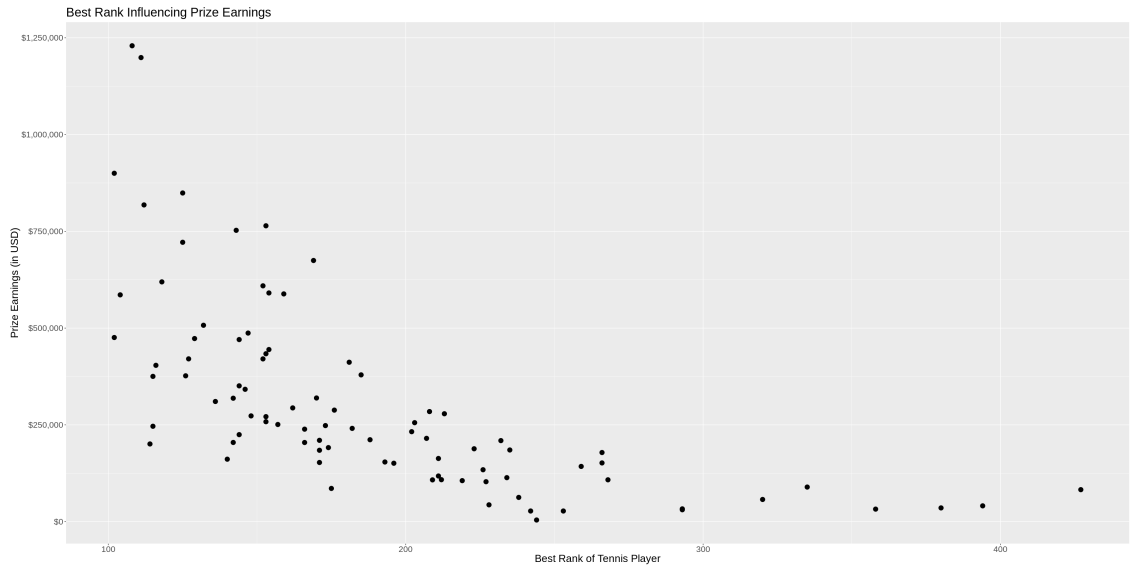
[4]:
```
# Find average prize money for each country's players
table1 <- tidy_data %>%
    group_by(Country) %>%
    summarize(avg_award_in_USD = mean(Prize.Money))

avg_award_in_USD <- table1$avg_award_in_USD

# count each country's number of players and then bind the data with the␣
 ↪average prize money column from above
final_table <- tidy_data %>%
    group_by(Country) %>%
    summarize(n = n()) %>%
    bind_cols(avg_award_in_USD) %>%
    mutate(avg_award_in_USD = ...3) %>%
    select(-...3)

# Find top 10 country with the most players
top_10 <- final_table %>%
        arrange(n) %>%
        tail(10)

# Plot the number of players for each top 10 country
top_10_graph <- ggplot(top_10,aes(x = Country, y = n)) +
        geom_bar(stat = "identity") +
        labs(x = "Country", y = "Number of People in Top 500 Tennis Players") +
        ggtitle("Top 10 Countries with most People in Top 500 Tennis Players") +
        coord_flip()

top_10_names <- pull(top_10, Country)
```

```
New names:
* `` -> …3
```

[5]:
```
# Start working on Top 10
top_10_data <- tidy_data %>%
                filter(Country == "United States"| Country == "United Kingdom"␣
 ↪| Country == "Spain" |
```

```
                    Country == "Russian Federation" | Country == "Japan" |␣
 ↪Country == "Italy" |
                    Country == "Germany" | Country == "France" | Country ==␣
 ↪"Australia"| Country == "Argentina") %>%
                select(-Name)

# averaged Best Rank
top_10_mean_money_over_rank <- top_10_data %>%
    group_by(Best.Rank) %>%
    summarize(mean_Prize_Money = mean(Prize.Money))

top_10_data <- filter(top_10_data, Prize.Money != 61544007 & Prize.Money !=␣
 ↪119601561)
```

[6]:
```
# Start working on bottom ranked players
top_10_data_bottom <- filter(top_10_data, Best.Rank > 100)

options(repr.plot.width = 30, repr.plot.height = 15)

bestRank_over_money_plot <- ggplot(top_10_data_bottom, aes(x = Best.Rank, y =␣
 ↪Prize.Money)) +
        geom_point(size = 4) +
        labs(x = "Best Rank of Tennis Player", y = "Prize Earnings (in USD)") +
        scale_y_continuous(labels = dollar_format()) +
        ggtitle("Best Rank Influencing Prize Earnings") +
        theme(text = element_text(size = 20))

bestRank_over_money_plot
```

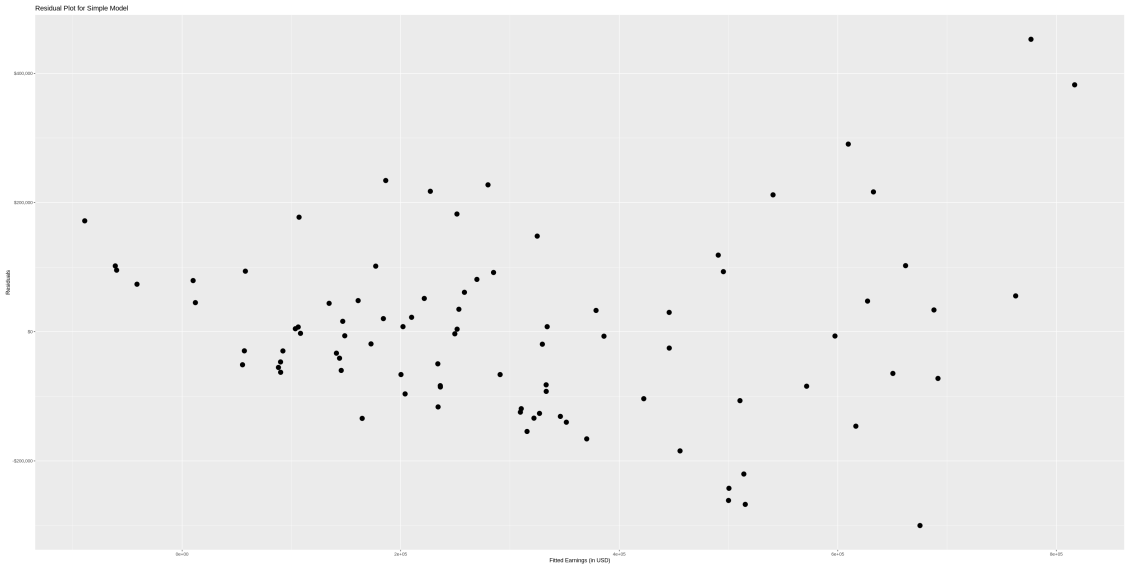


Now start model fitting

[13]:
```
#After removing those two outliers
# Prize Money = Age + Seasons + Best Rank + Best Rank^2 + Age * Best Rank
simple_model <- lm(Prize.Money~Age+Best.Rank+Seasons, data = top_10_data_bottom)
simple_model_summary <- summary(simple_model)
simple_residual_plot <- ggplot(simple_model,  aes(x = fitted.values(simple_model), y = residuals(simple_model))) +
       geom_point(size = 4) +
       labs(x = "Fitted Earnings (in USD)", y = "Residuals") +
       scale_y_continuous(labels = dollar_format()) +
       ggtitle("Residual Plot for Simple Model")
       theme(text = element_text(size = 20))

simple_normal_plot <- ggplot(simple_model, mapping = aes(sample =residuals(simple_model))) +
       stat_qq_point(size = 2) +
       ggtitle("Normal Plot for Simple Model") +
       theme(text = element_text(size = 20))

AIC_simple <- AIC(simple_model)
BIC_simple <- BIC(simple_model)



simple_model_summary
simple_residual_plot
simple_normal_plot
AIC_simple
BIC_simple
```

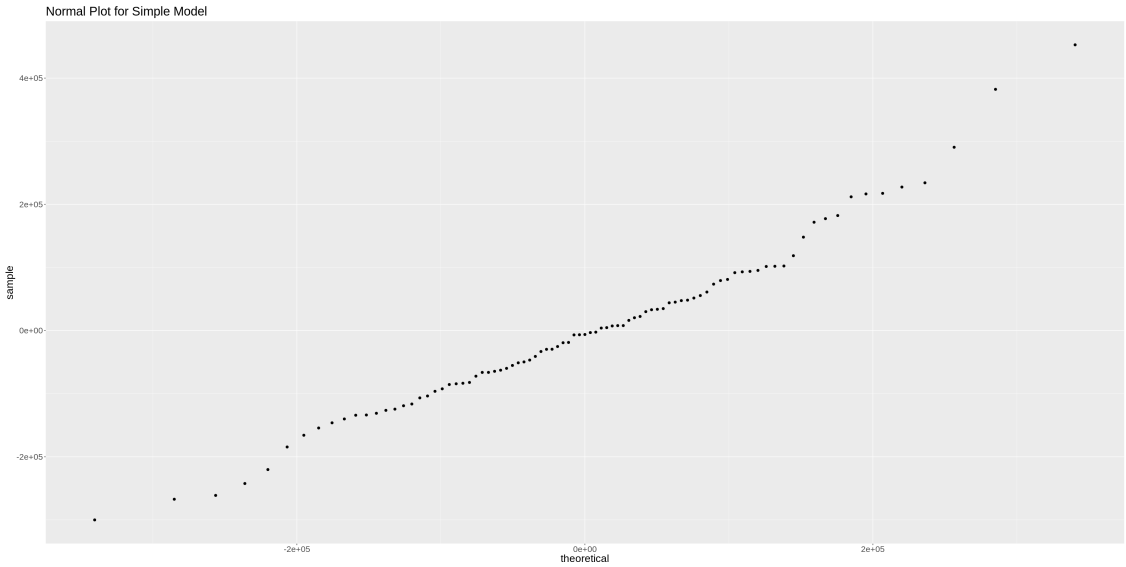
```
List of 1
 $ text:List of 11
  ..$ family       : NULL
  ..$ face         : NULL
  ..$ colour       : NULL
  ..$ size         : num 20
  ..$ hjust        : NULL
  ..$ vjust        : NULL
  ..$ angle        : NULL
  ..$ lineheight   : NULL
  ..$ margin       : NULL
  ..$ debug        : NULL
  ..$ inherit.blank: logi FALSE
  ..- attr(*, "class")= chr [1:2] "element_text" "element"
 - attr(*, "class")= chr [1:2] "theme" "gg"
 - attr(*, "complete")= logi FALSE
```

```
 - attr(*, "validate")= logi TRUE


Call:
lm(formula = Prize.Money ~ Age + Best.Rank + Seasons, data = top_10_data_bottom)

Residuals:
    Min      1Q  Median      3Q     Max
-299998  -84323   -6246   73573  452715

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -37818.6   140511.6  -0.269  0.78847
Age          10626.2     4532.7   2.344  0.02139 *
Best.Rank     -847.1      274.3  -3.088  0.00272 **
Seasons      97885.4    11671.4   8.387  9.4e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 137400 on 85 degrees of freedom
Multiple R-squared:  0.7114,        Adjusted R-squared:  0.7012
F-statistic: 69.83 on 3 and 85 DF,  p-value: < 2.2e-16
```



2364.3308991404

2376.77408098906



[8]:
```
# regsusbets model selection
data_without_country <- select(top_10_data_bottom, -Country)
s <- regsubsets(Prize.Money~., data = data_without_country, method =␣
 ↪"exhaustive")
model_selection_stats <- summary(s)

model_selection_stats
model_selection_stats$adjr2
model_selection_stats$cp
model_selection_stats$rsq
```

```
Subset selection object
Call: regsubsets.formula(Prize.Money ~ ., data = data_without_country,
    method = "exhaustive")
4 Variables  (and intercept)
             Forced in Forced out
Age              FALSE      FALSE
Current.Rank     FALSE      FALSE
Best.Rank        FALSE      FALSE
Seasons          FALSE      FALSE
1 subsets of each size up to 4
Selection Algorithm: exhaustive
         Age Current.Rank Best.Rank Seasons
1  ( 1 ) " " " "          " "       "*"
2  ( 1 ) " " " "          "*"       "*"
3  ( 1 ) "*" "*"          " "       "*"
4  ( 1 ) "*" "*"          "*"       "*"
```

1. 0.638500093004398 2. 0.685564414609355 3. 0.709958766603309 4. 0.712151456511671

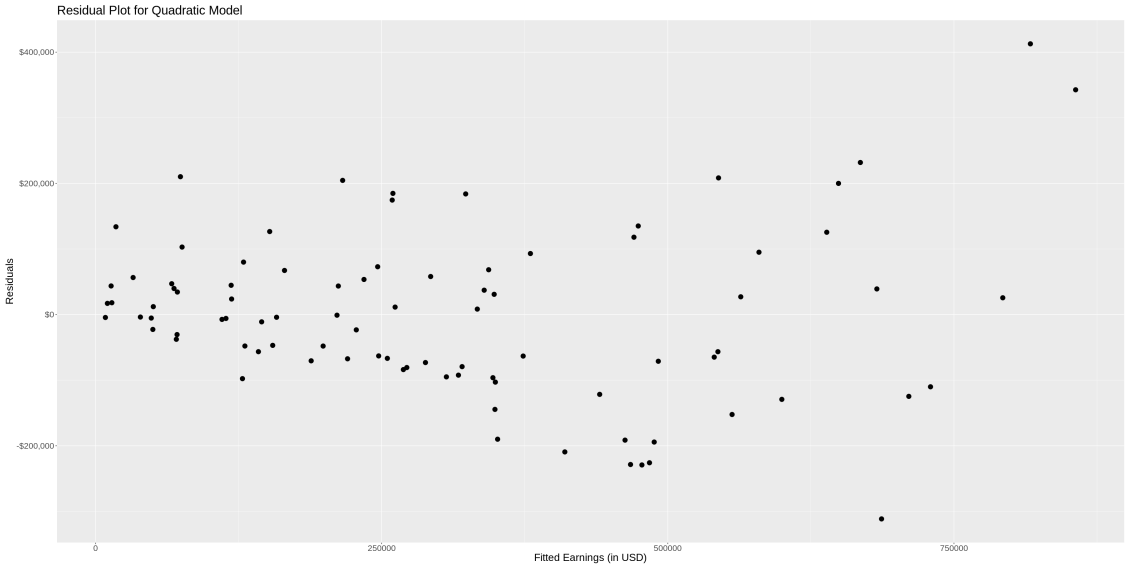
1. 24.2605560114378 2. 10.9433634643072 3. 4.64748857142759 4. 5

1. 0.642608046492985 2. 0.692710677913688 3. 0.719846535923651 4. 0.725235481215686

[9]:
```
# Prize Money = Age + Seasons + Best Rank + Best Rank^2
quadratic_model <- lm(Prize.Money~Age+I(Best.Rank^2)+Best.Rank+Seasons, data =␣
 ↪top_10_data_bottom)
quadratic_model_summary <- summary(quadratic_model)
quadratic_model_plot <- ggplot(quadratic_model, aes(y =␣
 ↪residuals(quadratic_model), x = fitted.values(quadratic_model))) +
       geom_point(size = 4) +
       labs(x = "Fitted Earnings (in USD)", y = "Residuals") +
       scale_y_continuous(labels = dollar_format()) +
       ggtitle("Residual Plot for Quadratic Model") +
       theme(text = element_text(size = 20))

quadratic_normal_plot <- ggplot(quadratic_model, mapping = aes(sample =␣
 ↪residuals(quadratic_model))) +
       stat_qq_point(size = 2) +
       ggtitle("Normal Plot for Quadratic Model") +
       theme(text = element_text(size = 20))

AIC_quadratic <- AIC(quadratic_model)
BIC_quadratic <- BIC(quadratic_model)


quadratic_model_summary
quadratic_model_plot
quadratic_normal_plot
AIC_quadratic
BIC_quadratic
```

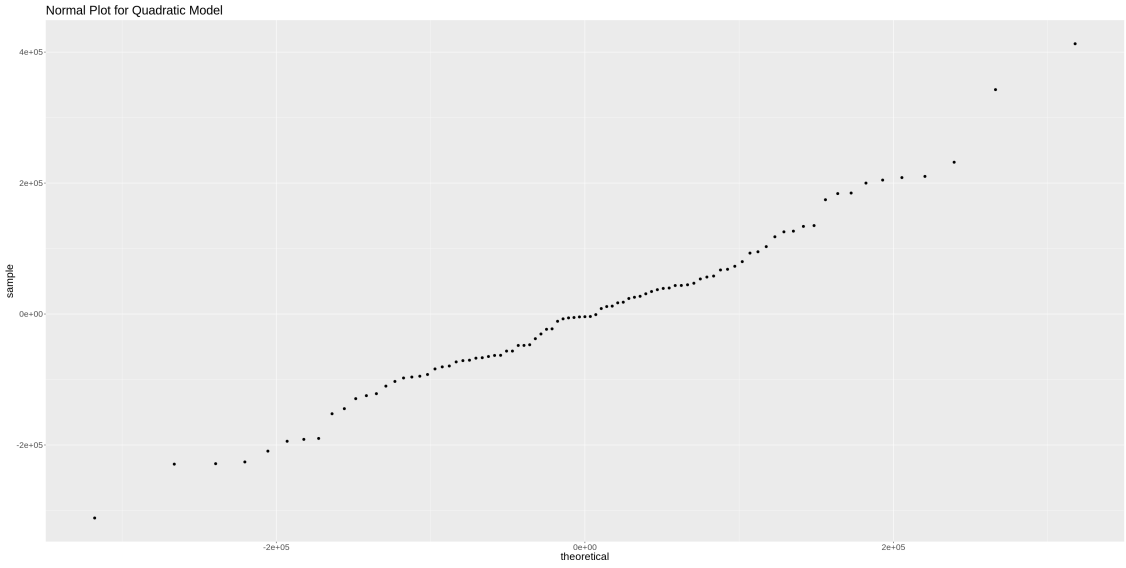
```
Call:
lm(formula = Prize.Money ~ Age + I(Best.Rank^2) + Best.Rank +
    Seasons, data = top_10_data_bottom)

Residuals:
    Min      1Q  Median      3Q     Max
-311490  -73125   -4095   58015  412741

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)    380518.622 177498.668   2.144 0.034941 *
Age             12137.211   4277.587   2.837 0.005700 **
I(Best.Rank^2)      7.975      2.264   3.523 0.000693 ***
Best.Rank       -4751.660   1137.807  -4.176 7.21e-05 ***
Seasons         82048.205  11845.007   6.927 8.09e-10 ***
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 129000 on 84 degrees of freedom
Multiple R-squared:  0.7485,        Adjusted R-squared:  0.7366
F-statistic: 62.51 on 4 and 84 DF,  p-value: < 2.2e-16
```



2354.06512262697

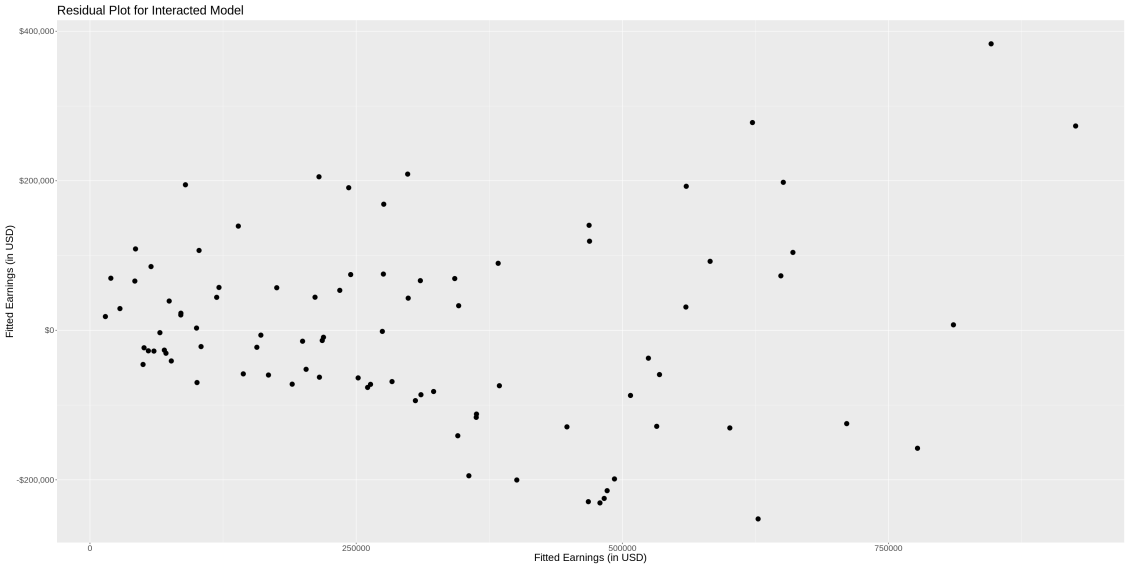
2368.99694084537



[12]:
```
#Prize Money = Age + Seasons + Best Rank
interacted_model <- lm(Prize.Money~Age*Best.Rank+I(Best.Rank^2)+Seasons, data =␣
 ↪top_10_data_bottom)
interacted_model_summary <- summary(interacted_model)
interacted_model_plot <- interacted_model %>%
        ggplot(aes(y = residuals(interacted_model), x = fitted.
 ↪values(interacted_model))) +
        geom_point(size = 4) +
        labs(x = "Fitted Earnings (in USD)", y = "Fitted Earnings (in USD)") +
        scale_y_continuous(labels = dollar_format()) +
        ggtitle("Residual Plot for Interacted Model") +
        theme(text = element_text(size = 20))

interacted_normal_plot <- ggplot(mapping = aes(sample =␣
 ↪residuals(interacted_model))) +
        stat_qq_point(size = 2) +
        ggtitle("Normal Plot for Interacted Model") +
        theme(text = element_text(size = 20))

AIC_interacted<- AIC(interacted_model)
BIC_interacted<- BIC(interacted_model)


interacted_model_summary
interacted_model_plot
interacted_normal_plot
AIC_interacted
BIC_interacted
```

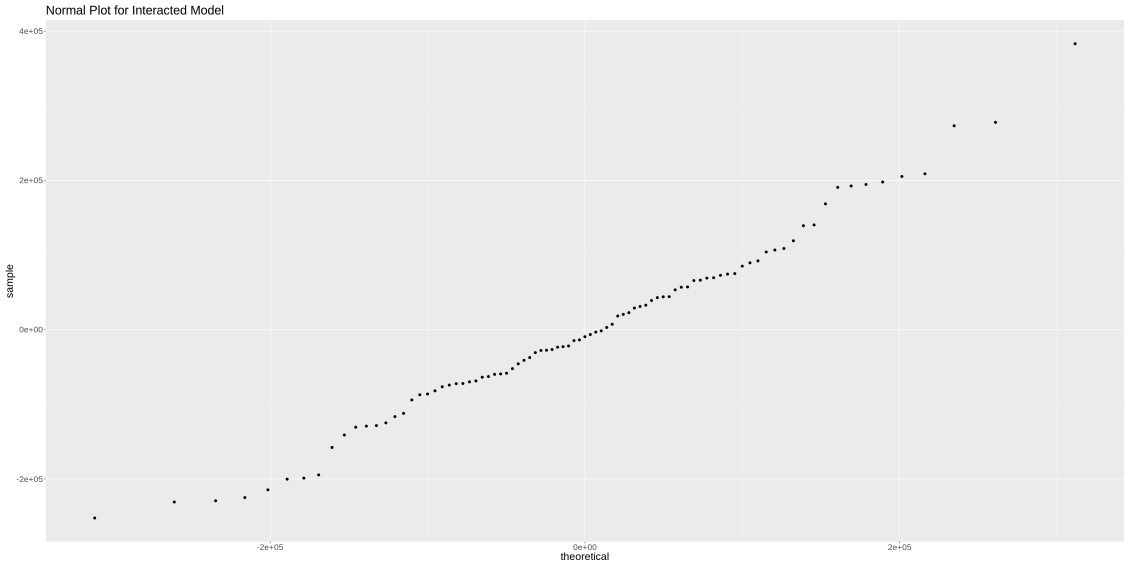
```
Call:
lm(formula = Prize.Money ~ Age * Best.Rank + I(Best.Rank^2) +
    Seasons, data = top_10_data_bottom)

Residuals:
    Min      1Q  Median      3Q     Max
-252263  -72278   -9306   69681  383184

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)    -3.553e+05  4.404e+05  -0.807   0.4222
Age             3.807e+04  1.486e+04   2.563   0.0122 *
Best.Rank      -1.036e+02  2.789e+03  -0.037   0.9705
I(Best.Rank^2)  4.798e+00  2.834e+00   1.693   0.0942 .
Seasons         7.905e+04  1.180e+04   6.699 2.34e-09 ***
Age:Best.Rank  -1.419e+02  7.792e+01  -1.821   0.0722 .
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 127300 on 83 degrees of freedom
Multiple R-squared:  0.7582,        Adjusted R-squared:  0.7436
F-statistic: 52.05 on 5 and 83 DF,  p-value: < 2.2e-16
```



2352.57941968382

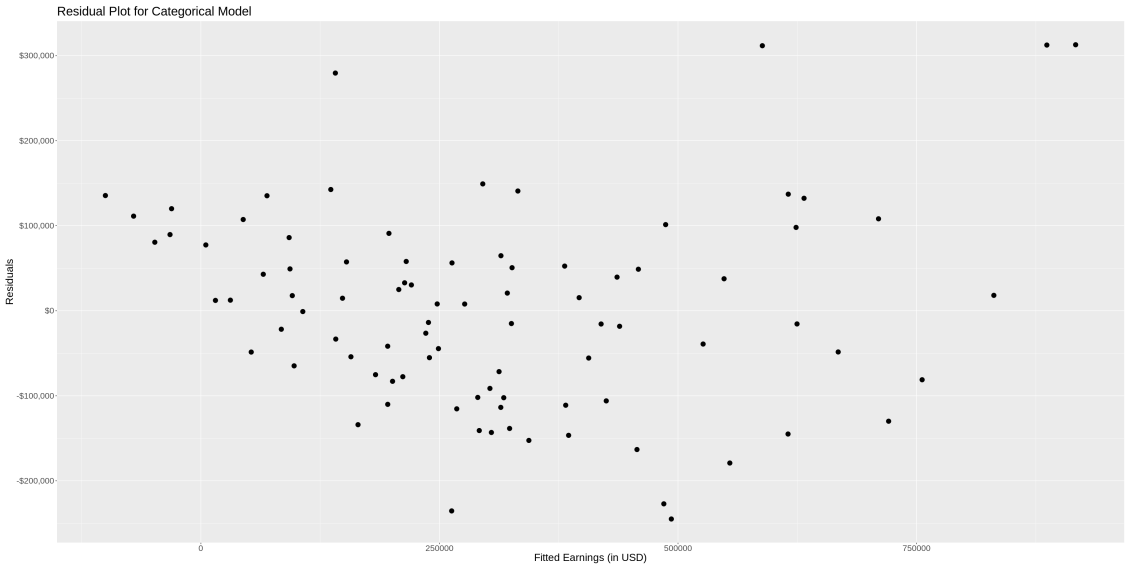
2369.99987427195



[11]:
```
# Categorical model
categorical_model <- lm(Prize.Money~Age+Current.Rank+Best.Rank+Seasons+Country,␣
 ↪data = top_10_data_bottom)
categorical_model_summary <- summary(categorical_model)
categorical_model_plot <- ggplot(categorical_model, aes(y =␣
 ↪residuals(categorical_model), x = fitted.values(categorical_model))) +
        geom_point(size = 4) +
        labs(x = "Fitted Earnings (in USD)", y = "Residuals") +
        scale_y_continuous(labels = dollar_format()) +
        ggtitle("Residual Plot for Categorical Model") +
        theme(text = element_text(size = 20))

categorical_normal_plot <- ggplot(categorical_model, mapping = aes(sample =␣
 ↪residuals(categorical_model))) +
        stat_qq_point(size = 2) +
        ggtitle("Normal Plot for Categorical Model") +
        theme(text = element_text(size = 20))

AIC_categorical <- AIC(categorical_model)
BIC_categorical <- BIC(categorical_model)

categorical_model_summary
categorical_model_plot
categorical_normal_plot
AIC_categorical
BIC_categorical
```

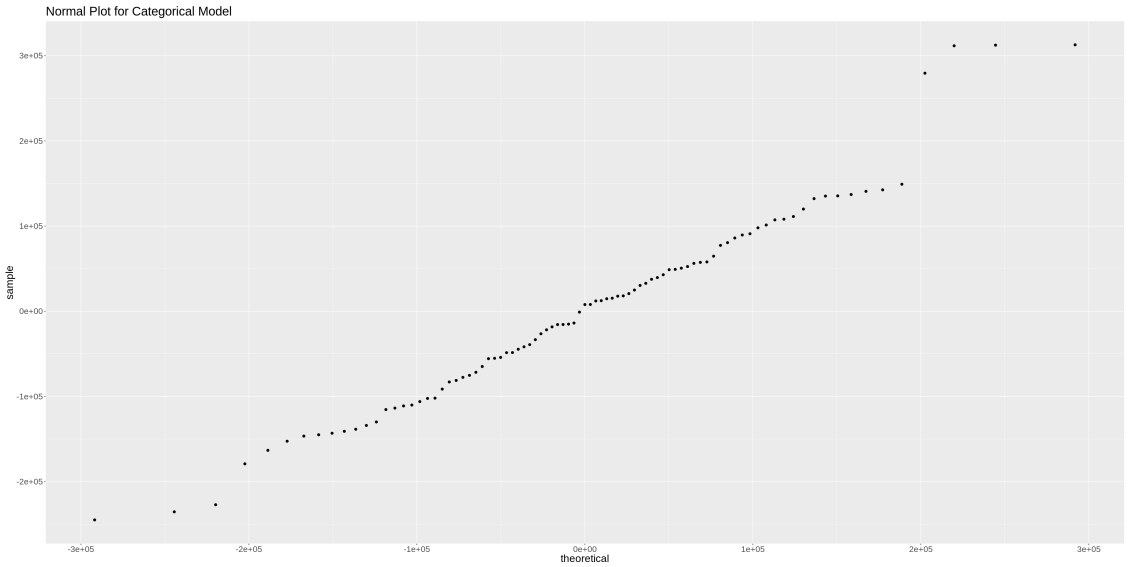
```
Call:
lm(formula = Prize.Money ~ Age + Current.Rank + Best.Rank + Seasons +
    Country, data = top_10_data_bottom)

Residuals:
    Min      1Q  Median      3Q     Max
-244891  -81142    7905   64714  312806

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)             -179947.6   142462.8  -1.263  0.21046
Age                       14581.4     4657.5   3.131  0.00249 **
Current.Rank               -390.7      183.2  -2.132  0.03626 *
Best.Rank                  -464.9      338.1  -1.375  0.17318
Seasons                   99700.1    11769.1   8.471 1.51e-12 ***
CountryAustralia         231543.5    67948.0   3.408  0.00106 **
CountryFrance             65873.4    61653.4   1.068  0.28875
CountryGermany            57256.9    67199.6   0.852  0.39690
CountryItaly              28994.5    64151.7   0.452  0.65260
```

```
CountryJapan                123138.7    89336.2   1.378  0.17219
CountryRussian Federation    56352.0    72512.0   0.777  0.43952
CountrySpain                110052.5    65642.1   1.677  0.09779 .
CountryUnited Kingdom       187232.6    92063.0   2.034  0.04551 *
CountryUnited States         27060.0    62492.5   0.433  0.66625
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 125400 on 75 degrees of freedom
Multiple R-squared:  0.7878,        Adjusted R-squared:  0.7511
F-statistic: 21.42 on 13 and 75 DF,  p-value: < 2.2e-16
```



2356.93470315161

2394.26424869759


Normal Plot for Categorical Model
sample
theoretical
3e+05
2e+05
1e+05
0e+00
-1e+05
-2e+05
-3e+05
-2e+05
-1e+05
0e+00
1e+05
2e+05
3e+05

[ ]: