What Makes the Best Tennis Player?

Steven Herrera and Ethan Shen

11/09/2018

Our Packages

```
library(tidyverse)
```

Introduction

In the sport of tennis, statisticians are often asked to analyze an individual player's statistics on how well they are performing and what they could possibly do to perform better at tournaments. But often, information about how well other players are doing is left out of the picture when determining what a tennis player should work on, especially given that the sport is continuously changing as new techniques and racquets are being introduced. Using datasets from 2010-2017 of the Men's ATP World Tour, we will attempt to solve the question "What makes a good tennis player?" using multiple linear regression models, filtering for each unique surface (Hard, Clay, and Grass). In our dataset, we will use the average winning statistics for players that have won tournaments in the tour, throughout these years. We are interested in solving what will give us the largest number of tournament wins, as that is what determines the "best player." We will look at many variables and hope to see if we can use it to predict what stats the best tennis player should have, in order to win the most number of tournaments wins.

Data

The data sets that we will be using for our analysis is not the same as the ones we initially received, as multiple data set manipulation techniques were used in Excel and in R.

1) In order to obtain the dataset in our RMD, we will first need the data to fit nicely into a CSV file. In excel, we used the `Text to Columns` function, and inserted the file here:

```
atp <- read_csv("files/atp.csv")
```

2) There are a lot of things interesting about this dataset. The first thing that we did was filter for events that are normal tournament-style matches (not ATP Final events or Davis Cup, which both occur past November 13, 2017, coded as `20171113` in the dataset under the `tourney_date` variable).

```
atp1 <- atp %>%
  filter(tourney_date < 20171113)
```

3) Because we are interested in winners, we will only look at the observations where the match was a **final**.

```
winners2017 <- atp1 %>%
  filter(round == "F")
```

4) Because we are looking in the time period from 2010-2017, we will do the same thing for the rest of the years:

```
atp2016 <- read_csv("files/atp2016.csv")
atp2015 <- read_csv("files/atp2015.csv")
atp2014 <- read_csv("files/atp2014.csv")
```

```
atp2013 <- read_csv("files/atp2013.csv")
atp2012 <- read_csv("files/atp2012.csv")
atp2011 <- read_csv("files/atp2011.csv")
atp2010 <- read_csv("files/atp2010.csv")

winners2016 <- atp2016 %>%
  filter(tourney_date < 20161114) %>%
  filter(round == "F")

winners2015 <- atp2015 %>%
  filter(tourney_date < 20151115) %>%
  filter(round == "F")

winners2014 <- atp2014 %>%
  filter(tourney_date < 20141109) %>%
  filter(round == "F")

winners2013 <- atp2013 %>%
  filter(tourney_date < 20131104) %>%
  filter(round == "F")

winners2012 <- atp2012 %>%
  filter(tourney_date < 20121105) %>%
  filter(round == "F")

winners2011 <- atp2011 %>%
  filter(tourney_date < 20111114) %>%
  filter(round == "F")

winners2010 <- atp2010 %>%
  filter(tourney_date < 20101121) %>%
  filter(round == "F")
```

```
winners <- rbind(winners2017, winners2016, winners2015, winners2014, winners2013,
                 winners2012, winners2011, winners2010)
```

5) We want to look at the statistics from each game based on the type of surface. Thus, we created three new data sets, each one filtered for a specific surface. For each data set, we first filtered out the NA's, which are in the original data set because those statistics were not recorded at the tournament. Then, we grouped by the winner's name and found the mean of each of the variables for that specific player. We also removed the variables with the statistics of the loser.

```
hard <- winners %>%
  filter(surface == "Hard",
         best_of == 3,
         !is.na(w_ace),
         !is.na(w_df),
         !is.na(w_svpt),
         !is.na(w_1stIn),
         !is.na(w_1stWon),
         !is.na(w_2ndWon),
         !is.na(w_SvGms),
         !is.na(w_bpSaved),
         !is.na(w_bpFaced)) %>%
```

```
  group_by(winner_name) %>%
  mutate(mean_w_ace = mean(w_ace),
         mean_w_df = mean(w_df),
         mean_w_svpt = mean(w_svpt),
         mean_w_1stIn = mean(w_1stIn),
         mean_w_1stWon = mean(w_1stWon),
         mean_w_2ndWon = mean(w_2ndWon),
         mean_w_SvGms = mean(w_SvGms),
         mean_w_bpSaved = mean(w_bpSaved),
         mean_w_bpFaced = mean(w_bpFaced),
         mean_minutes = mean(minutes),
         num = n())

myvars <- names(hard) %in% c("loser_id", "loser_seed", "loser_entry", "loser_name",
                             "loser_hand", "loser_ht", "loser_ioc", "loser_age",
                             "loser_rank", "loser_rank_points", "l_ace", "l_df",
                             "l_svpt", "l_1stIn", "l_1stWon", "l_2ndWon", "l_SvGms",
                             "l_bpSaved", "l_bpFaced")
hard <- hard[!myvars]
```

```
clay <- winners %>%
  filter(surface == "Clay",
         best_of == 3,
         !is.na(w_ace),
         !is.na(w_df),
         !is.na(w_svpt),
         !is.na(w_1stIn),
         !is.na(w_1stWon),
         !is.na(w_2ndWon),
         !is.na(w_SvGms),
         !is.na(w_bpSaved),
         !is.na(w_bpFaced)) %>%
  group_by(winner_name) %>%
  mutate(mean_w_ace = mean(w_ace),
         mean_w_df = mean(w_df),
         mean_w_svpt = mean(w_svpt),
         mean_w_1stIn = mean(w_1stIn),
         mean_w_1stWon = mean(w_1stWon),
         mean_w_2ndWon = mean(w_2ndWon),
         mean_w_SvGms = mean(w_SvGms),
         mean_w_bpSaved = mean(w_bpSaved),
         mean_w_bpFaced = mean(w_bpFaced),
         mean_minutes = mean(minutes),
         num = n())

myvars <- names(clay) %in% c("loser_id", "loser_seed", "loser_entry", "loser_name",
                             "loser_hand", "loser_ht", "loser_ioc", "loser_age",
                             "loser_rank", "loser_rank_points", "l_ace", "l_df",
                             "l_svpt", "l_1stIn", "l_1stWon", "l_2ndWon", "l_SvGms",
                             "l_bpSaved", "l_bpFaced")
clay <- clay[!myvars]
```

```
grass <- winners %>%
  filter(surface == "Grass",
```

```
        best_of == 3,
        !is.na(w_ace),
        !is.na(w_df),
        !is.na(w_svpt),
        !is.na(w_1stIn),
        !is.na(w_1stWon),
        !is.na(w_2ndWon),
        !is.na(w_SvGms),
        !is.na(w_bpSaved),
        !is.na(w_bpFaced)) %>%
  group_by(winner_name) %>%
  mutate(mean_w_ace = mean(w_ace),
        mean_w_df = mean(w_df),
        mean_w_svpt = mean(w_svpt),
        mean_w_1stIn = mean(w_1stIn),
        mean_w_1stWon = mean(w_1stWon),
        mean_w_2ndWon = mean(w_2ndWon),
        mean_w_SvGms = mean(w_SvGms),
        mean_w_bpSaved = mean(w_bpSaved),
        mean_w_bpFaced = mean(w_bpFaced),
        mean_minutes = mean(minutes),
        num = n())

myvars <- names(grass) %in% c("loser_id", "loser_seed", "loser_entry", "loser_name",
                              "loser_hand", "loser_ht", "loser_ioc", "loser_age",
                              "loser_rank", "loser_rank_points", "l_ace", "l_df",
                              "l_svpt", "l_1stIn", "l_1stWon", "l_2ndWon", "l_SvGms",
                              "l_bpSaved", "l_bpFaced")
grass <- grass[!myvars]
```

Now, we have three data sets, each one with the statistics of a player's performance on a specific surface.

```
glimpse(hard)
```

```
## Observations: 270
## Variables: 41
## $ tourney_id         <chr> "2017-M020", "2017-0891", "2017-0451", "201...
## $ tourney_name       <chr> "Brisbane", "Chennai", "Doha", "Auckland", ...
## $ surface            <chr> "Hard", "Hard", "Hard", "Hard", "Hard", "Ha...
## $ draw_size          <int> 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,...
## $ tourney_level      <chr> "A", "A", "A", "A", "A", "A", "A", "A", "A"...
## $ tourney_date       <int> 20170102, 20170102, 20170102, 20170109, 201...
## $ match_num          <int> 300, 300, 300, 300, 300, 300, 300, 300, 300...
## $ winner_id          <int> 105777, 105138, 104925, 106058, 100644, 105...
## $ winner_seed        <int> 7, 2, 2, 4, 4, 3, NA, 6, 2, NA, 1, 9, 4, 2,...
## $ winner_entry       <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ winner_name        <chr> "Grigor Dimitrov", "Roberto Bautista Agut",...
## $ winner_hand        <chr> "R", "R", "R", "R", "R", "R", "R", "R", "R"...
## $ winner_ht          <int> 188, 183, 188, 185, 198, 188, 183, 188, 188...
## $ winner_ioc         <chr> "BUL", "ESP", "SRB", "USA", "GER", "BUL", "...
## $ winner_age         <dbl> 25.63450, 28.72005, 29.61807, 24.29295, 19....
## $ winner_rank        <int> 17, 14, 2, 23, 21, 13, 62, 14, 11, 40, 1, 1...
## $ winner_rank_points <int> 2035, 2350, 11780, 1710, 1735, 2765, 746, 2...
## $ score              <chr> "6-2 2-6 6-3", "6-3 6-4", "6-3 5-7 6-4", "6...
```

```
## $ best_of            <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3...
## $ round              <chr> "F", "F", "F", "F", "F", "F", "F", "F", "F"...
## $ minutes            <int> 108, 73, 174, 116, 88, 98, 76, 115, 69, 94,...
## $ w_ace              <int> 7, 4, 2, 13, 8, 10, 9, 10, 7, 19, 2, 2, 5, ...
## $ w_df               <int> 2, 1, 3, 3, 4, 9, 0, 0, 0, 2, 4, 1, 1, 1, 2...
## $ w_svpt             <int> 77, 49, 110, 78, 72, 85, 66, 81, 46, 75, 45...
## $ w_1stIn            <int> 52, 38, 79, 54, 46, 58, 37, 55, 33, 48, 25,...
## $ w_1stWon           <int> 41, 32, 54, 43, 37, 41, 25, 44, 30, 39, 20,...
## $ w_2ndWon           <int> 12, 9, 19, 11, 11, 8, 18, 11, 10, 17, 12, 1...
## $ w_SvGms            <int> 13, 10, 16, 15, 11, 11, 9, 14, 10, 11, 9, 1...
## $ w_bpSaved          <int> 5, 0, 5, 1, 0, 6, 12, 6, 0, 6, 1, 2, 4, 0, ...
## $ w_bpFaced          <int> 7, 0, 7, 4, 1, 9, 12, 8, 0, 6, 3, 3, 4, 1, ...
## $ mean_w_ace         <dbl> 6.600000, 2.500000, 4.285714, 11.000000, 5....
## $ mean_w_df          <dbl> 3.8000000, 1.0000000, 1.9285714, 3.0000000,...
## $ mean_w_svpt        <dbl> 85.00000, 43.25000, 68.39286, 75.00000, 72....
## $ mean_w_1stIn       <dbl> 55.20000, 28.25000, 44.64286, 46.50000, 47....
## $ mean_w_1stWon      <dbl> 40.00000, 22.25000, 33.50000, 37.00000, 36....
## $ mean_w_2ndWon      <dbl> 16.00000, 9.00000, 13.42857, 15.50000, 13.2...
## $ mean_w_SvGms       <dbl> 13.00000, 8.50000, 11.32143, 14.50000, 11.5...
## $ mean_w_bpSaved     <dbl> 5.000000, 0.000000, 2.571429, 1.000000, 2.7...
## $ mean_w_bpFaced     <dbl> 7.000000, 1.000000, 4.000000, 4.000000, 3.7...
## $ mean_minutes       <dbl> 119.80000, 68.50000, NA, 117.00000, 92.0000...
## $ num                <int> 5, 4, 28, 2, 4, 5, 1, 10, 10, 6, 18, 22, 22...
```

There are 270 observations in the Hard Court-only dataset, along with 41 variables.

```
glimpse(clay)
```

```
## Observations: 172
## Variables: 41
## $ tourney_id         <chr> "2017-0506", "2017-6932", "2017-0533", "201...
## $ tourney_name       <chr> "Buenos Aires", "Rio De Janeiro", "Sao Paul...
## $ surface            <chr> "Clay", "Clay", "Clay", "Clay", "Clay", "Cl...
## $ draw_size          <int> 32, 32, 32, 32, 32, 64, 64, 32, 32, 32, 32,...
## $ tourney_level      <chr> "A", "A", "A", "A", "A", "M", "A", "A", "A"...
## $ tourney_date       <int> 20170213, 20170220, 20170227, 20170410, 201...
## $ match_num          <int> 300, 300, 270, 300, 300, 300, 300, 300, 300...
## $ winner_id          <int> 105238, 106233, 104655, 105449, 106432, 104...
## $ winner_seed        <int> NA, 2, 3, 4, NA, 4, 3, 1, 1, 2, 3, 4, 16, 1...
## $ winner_entry       <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,...
## $ winner_name        <chr> "Alexandr Dolgopolov", "Dominic Thiem", "Pa...
## $ winner_hand        <chr> "R", "R", "R", "R", "R", "L", "L", "R", "R"...
## $ winner_ht          <int> 180, 185, 180, 188, NA, 185, 185, 185, 188,...
## $ winner_ioc         <chr> "UKR", "AUT", "URU", "USA", "CRO", "ESP", "...
## $ winner_age         <dbl> 28.26831, 23.46612, 31.15674, 27.29363, 20....
## $ winner_rank        <int> 66, 8, 33, 29, 79, 7, 5, 14, 21, 8, 20, 5, ...
## $ winner_rank_points <int> 715, 3375, 1085, 1380, 675, 3735, 4235, 251...
## $ score              <chr> "7-6(4) 6-4", "7-5 6-4", "6-7(3) 6-4 6-4", ...
## $ best_of            <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3...
## $ round              <chr> "F", "F", "F", "F", "F", "F", "F", "F", "F"...
## $ minutes            <int> 100, 94, 179, 144, 158, 76, 90, 64, 82, 120...
## $ w_ace              <int> 4, 7, 13, 9, 10, 5, 1, 5, 1, 9, 8, 4, 7, 3,...
## $ w_df               <int> 0, 1, 12, 8, 2, 1, 0, 2, 0, 1, 1, 4, 2, 2, ...
## $ w_svpt             <int> 75, 61, 112, 107, 118, 41, 53, 48, 58, 79, ...
## $ w_1stIn            <int> 44, 33, 73, 53, 72, 31, 35, 26, 43, 51, 34,...
```

```
## $ w_1stWon           <int> 37, 22, 54, 42, 47, 25, 25, 22, 31, 42, 22,...
## $ w_2ndWon           <int> 17, 20, 20, 27, 28, 7, 13, 10, 12, 14, 17, ...
## $ w_SvGms            <int> 11, 11, 16, 16, 18, 8, 9, 8, 10, 11, 10, 11...
## $ w_bpSaved          <int> 3, 1, 6, 6, 6, 0, 1, 0, 0, 7, 2, 5, 0, 5, 2...
## $ w_bpFaced          <int> 3, 3, 7, 9, 9, 0, 1, 1, 1, 7, 3, 6, 0, 8, 2...
## $ mean_w_ace         <dbl> 5.000000, 5.666667, 7.000000, 9.000000, 10....
## $ mean_w_df          <dbl> 0.500000, 3.000000, 4.000000, 8.000000, 2.0...
## $ mean_w_svpt        <dbl> 74.50000, 78.83333, 76.50000, 107.00000, 11...
## $ mean_w_1stIn       <dbl> 38.00000, 51.50000, 51.50000, 53.00000, 72....
## $ mean_w_1stWon      <dbl> 32.50000, 38.50000, 38.50000, 42.00000, 47....
## $ mean_w_2ndWon      <dbl> 20.50000, 14.83333, 14.50000, 27.00000, 28....
## $ mean_w_SvGms       <dbl> 12.500000, 12.833333, 12.500000, 16.000000,...
## $ mean_w_bpSaved     <dbl> 2.500000, 4.166667, 2.500000, 6.000000, 6.0...
## $ mean_w_bpFaced     <dbl> 3.500000, 6.333333, 3.500000, 9.000000, 9.0...
## $ mean_minutes       <dbl> 117.00000, NA, 112.00000, 144.00000, 158.00...
## $ num                <int> 2, 6, 6, 1, 1, 22, 22, 1, 1, 2, 2, 22, 2, 5...
```

There are 172 observations in the Clay Court-only dataset, along with 41 variables.

```
glimpse(grass)
```

```
## Observations: 43
## Variables: 41
## $ tourney_id         <chr> "2017-M010", "2017-0321", "2017-0500", "201...
## $ tourney_name       <chr> "'S-Hertogenbosch", "Stuttgart", "Halle", "...
## $ surface            <chr> "Grass", "Grass", "Grass", "Grass", "Grass"...
## $ draw_size          <int> 32, 32, 32, 32, 32, 32, 32, 32, 32, 32, 32,...
## $ tourney_level      <chr> "A", "A", "A", "A", "A", "A", "A", "A", "A"...
## $ tourney_date       <int> 20170612, 20170612, 20170619, 20170619, 201...
## $ match_num          <int> 300, 300, 300, 300, 300, 300, 300, 300, 300...
## $ winner_id          <int> 104180, 106298, 103819, 103852, 105216, 104...
## $ winner_seed        <int> 4, 4, 1, NA, NA, 1, 1, 8, 3, NA, 1, 6, NA, ...
## $ winner_entry       <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, "PR", N...
## $ winner_name        <chr> "Gilles Muller", "Lucas Pouille", "Roger Fe...
## $ winner_hand        <chr> "L", "R", "R", "L", "R", "R", "R", "R", "R"...
## $ winner_ht          <int> 193, 185, 185, 188, 173, 188, 206, 190, NA,...
## $ winner_ioc         <chr> "LUX", "FRA", "SUI", "ESP", "JPN", "SRB", "...
## $ winner_age         <dbl> 34.09446, 23.29911, 35.86311, 35.74538, 28....
## $ winner_rank        <int> 28, 16, 5, 32, 66, 4, 21, 49, 7, 192, 2, 38...
## $ winner_rank_points <int> 1425, 2365, 4765, 1220, 725, 5805, 1885, 93...
## $ score              <chr> "7-6(5) 7-6(4)", "4-6 7-6(5) 6-4", "6-1 6-3...
## $ best_of            <int> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3...
## $ round              <chr> "F", "F", "F", "F", "F", "F", "F", "F", "F"...
## $ minutes            <int> 92, 125, 53, 151, 89, 76, 74, 76, 131, 115,...
## $ w_ace              <int> 22, 29, 4, 19, 5, 1, 17, 7, 12, 3, 6, 10, 2...
## $ w_df               <int> 1, 4, 0, 0, 3, 1, 0, 1, 6, 0, 1, 3, 2, 0, 0...
## $ w_svpt             <int> 65, 105, 42, 105, 66, 58, 61, 66, 104, 98, ...
## $ w_1stIn            <int> 49, 58, 28, 66, 44, 42, 41, 47, 50, 63, 62,...
## $ w_1stWon           <int> 46, 45, 26, 51, 33, 32, 37, 34, 45, 44, 52,...
## $ w_2ndWon           <int> 12, 30, 7, 27, 8, 8, 12, 10, 28, 19, 10, 13...
## $ w_SvGms            <int> 12, 16, 8, 17, 10, 9, 11, 10, 16, 15, 15, 1...
## $ w_bpSaved          <int> 0, 6, 1, 2, 3, 3, 0, 3, 5, 2, 0, 0, 3, 4, 0...
## $ w_bpFaced          <int> 0, 7, 1, 3, 5, 3, 0, 4, 5, 4, 1, 1, 3, 4, 0...
## $ mean_w_ace         <dbl> 22.00000, 29.00000, 8.50000, 16.33333, 5.00...
## $ mean_w_df          <dbl> 1.0000000, 4.0000000, 0.7500000, 3.0000000,...
```

```
## $ mean_w_svpt        <dbl> 65.00000, 105.00000, 71.75000, 109.33333, 6...
## $ mean_w_1stIn       <dbl> 49.00000, 58.00000, 47.75000, 65.66667, 44....
## $ mean_w_1stWon      <dbl> 46.00000, 45.00000, 39.50000, 48.66667, 33....
## $ mean_w_2ndWon      <dbl> 12.00000, 30.00000, 14.00000, 28.00000, 8.0...
## $ mean_w_SvGms       <dbl> 12.00, 16.00, 11.75, 16.00, 10.00, 9.00, 11...
## $ mean_w_bpSaved     <dbl> 0.0000000, 6.0000000, 2.5000000, 4.0000000,...
## $ mean_w_bpFaced     <dbl> 0.0000000, 7.0000000, 3.0000000, 5.0000000,...
## $ mean_minutes       <dbl> 92.00000, 125.00000, NA, 150.66667, 89.0000...
## $ num                <int> 1, 1, 4, 3, 1, 1, 3, 4, 1, 1, 4, 1, 4, 4, 4...
```

There are 43 observations in the Grass Court-only dataset, along with 41 variables.

Analysis

1) Our response variable is going to be `num`, a variable that we created that represents the number of tournaments wins for a tennis player. The variable type is an integer. One interest thing that we initially noticed is that the number of tournament wins varied greatly within each group type: Hard, Clay, and Grass, which just means that for some tournaments, being a consistent tournament winner is more likely in a surface type over another.

```
winners %>%
  filter(!is.na(w_ace),
         !is.na(w_df),
         !is.na(w_svpt),
         !is.na(w_1stIn),
         !is.na(w_1stWon),
         !is.na(w_2ndWon),
         !is.na(w_SvGms),
         !is.na(w_bpSaved),
         !is.na(w_bpFaced),
         best_of == 3) %>%
  group_by(surface) %>%
  mutate(num = n()) %>%
  count(num) %>%
  select(surface, num)
```

```
## # A tibble: 3 x 2
## # Groups:   surface [3]
##   surface   num
##   <chr>   <int>
## 1 Clay      172
## 2 Grass      43
## 3 Hard      270
```

2) The explanatory variables that we wish to understand are the ones that can be changed by a player. The variables include:

- `mean_w_ace` : the average amount of aces played.
- `mean_w_df` : the average amount of double faults.
- `mean_w_svpt` : the average amount of service points won.
- `mean_w_1stIn` : the average first-service percentage.
- `mean_w_1stWon` : the average winning percentage of the point, given that it was a first serve.

- `mean_w_2ndWon` : the average winnning percentage of the point, given that it was a first serve.
- `mean_w_SvGms` : the average amount of service games.
- `mean_w_bpSaved` : the average amount of break points saved.
- `mean_w_bpFaced` : the average amount of break points faced.
- `mean_minutes` : the average time it takes to win the match, in minutes.
- We understand that all of these variables are numerical, so differences among groups isn't a concern, since we already established that the "best tennis player" differs greatly by surface type and we would not be testing whether that is different.

3) The hypotheses tests that we will be using are nested F-tests to determine which variables best determine the amount of tournament wins from 2010-2017. We will begin with using a backwards selection method, and then determine what variables best explain our response variable by interpreting the variables of interest in our model. We will also consider interaction variables, and use nested F-tests to determine if those interactions are significant.

4) The proposed methods we will be using in our model include:

- Exploratory Data Analysis, in which we check for potential issues or multicollinearity.
- Backwards Selection Modelling
- ANOVA Nested F-tests
- Assumptions (and appropriately adjusting our model to meet the assumptions)
- Conclusion with detailed analysis of our model, which includes how well our model predicts and examples of how this information can be used to consult tennis players.

Reference

We used the following data sets: `atp_matches_2017`, `atp_matches_2016`,`atp_matches_2015`, `atp_matches_2014`, `atp_matches_2013`, `atp_matches_2012`, `atp_matches_2011`, `atp_matches_2010`, all created by JeffSackMann on GitHub.

Here is a link to the data sets: https://github.com/JeffSackmann/tennis_atp.