GOAT of Tennis

Grant Weaver

2023-02-25

Who is the GOAT of men's singles tennis?

In other words, who is the greatest male singles player in the open era? This is the question that we will answer in this document. Only on-court achievements will be included, thus no points will be given for style or excitement factors.

Data Source

The dataset comes from JeffSackmann's repository on github. Ultimate Tennis Statistics Website holds a lot of the data as well. Many thanks to them, for without them this project wouldn't be possible!

Figure 1: .

Packages

```
# Packages needed
library(tidyverse)
library(tibble)
library(data.table)
library(dplyr)
library(purrr)
library(ggthemes)
library(lubridate)
```

Reading in the Data

```
# Below imports the data
# Using the read_csv function imports the data as a tibble and thus is now tidy data
Atp_singles_1968 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1968.csv")
Atp_singles_1969 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1969.csv")
Atp_singles_1970 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1970.csv")
Atp_singles_1971 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1971.csv")
Atp_singles_1972 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1972.csv")
Atp_singles_1973 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1973.csv")
Atp_singles_1974 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1974.csv")
Atp_singles_1975 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1975.csv")
```

```
Atp_singles_1976 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1976.csv")
Atp_singles_1977 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1977.csv")
Atp_singles_1978 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1978.csv")
Atp_singles_1979 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1979.csv")
Atp_singles_1980 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1980.csv")
Atp_singles_1981 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1981.csv")
Atp_singles_1982 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1982.csv")
Atp_singles_1983 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1983.csv")
Atp_singles_1984 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1984.csv")
Atp_singles_1985 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1985.csv")
Atp_singles_1986 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1986.csv")
Atp_singles_1987 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1987.csv")
Atp_singles_1988 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1988.csv")
Atp_singles_1989 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1989.csv")
Atp_singles_1990 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1990.csv")
Atp_singles_1991 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1991.csv")
Atp_singles_1992 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1992.csv")
Atp_singles_1993 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1993.csv")
Atp_singles_1994 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1994.csv")
Atp_singles_1995 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1995.csv")
Atp_singles_1996 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1996.csv")
Atp_singles_1997 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1997.csv")
Atp_singles_1998 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1998.csv")
Atp_singles_1999 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_1999.csv")
Atp_singles_2000 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2000.csv")
Atp_singles_2001 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2001.csv")
Atp_singles_2002 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2002.csv")
Atp_singles_2003 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2003.csv")
Atp_singles_2004 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2004.csv")
Atp_singles_2005 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2005.csv")
Atp_singles_2006 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2006.csv")
Atp_singles_2007 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2007.csv")
Atp_singles_2008 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2008.csv")
Atp_singles_2009 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2009.csv")
Atp_singles_2010 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2010.csv")
Atp_singles_2011 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2011.csv")
Atp_singles_2012 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2012.csv")
Atp_singles_2013 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2013.csv")
Atp_singles_2014 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2014.csv")
Atp_singles_2015 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2015.csv")
Atp_singles_2016 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2016.csv")
Atp_singles_2017 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2017.csv")
Atp_singles_2018 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2018.csv")
Atp_singles_2019 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2019.csv")
Atp_singles_2020 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2020.csv")
Atp_singles_2021 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2021.csv")
Atp_singles_2022 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2022.csv")
Atp_singles_2022 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2022.csv")
Atp_singles_2023 <- read_csv("/Users/grantweaver/Desktop/Stats/R/Tennis/Data/atp_matches_2023.csv")
```

Exploring the Variable Names

```
# --- "Variable Names" Then a description of the variable. 49 variable names (columns) -------
# "tourney_id" Is the id for the tournament, just a number. However, the first four digits are the year
# "tourney_name" The name of the tournament. "US Open", Roland Garros
# "surface" Grass, Clay, Carpet, Hard. Also unknown or N/A
# "draw_size" Number = number of players, thus 128 = 128 player draw
# "tourney_level" (F = tour finals or Nxt Gen, G = Grand Slam, M = Masters, D = Davis Cup, A = others)
# "tourney_date" (year/month/date | 2000/06/21) Also the date is the first day that the tournament star
# "match_num" Is the number of the match. Thus the last match_num is the final.
# "winner_id" Each player has their own specific winner_id, with Federer = 103819
# "winner_seed" The seeding the player was during the tournament
# "winner_entry" Blank = normal; WC = wildcard, PR = Protected Ranking, Q = qualifer, LL = lucky loser,
# SE = Special Exemption, ALT = Alternate, Alt = Alternate
# "winner_name" first_name last_name , "Roger Federer"
# "winner_hand" R = Right Hander, L = Left Handed, U = Unkown
# "winner_ht" 211 =2.11 meters = 6ft 11inches
# "winner_ioc" The three letters that represent their country. USA = United States of America. RSA = Ru
# "winner_age" 20.9 = 20 years old and
# "loser_id" id of the player
# "loser_seed" Seed of the loser
# "loser_entry" Entry of the loser, WC, PR for example
# "loser_name" Name of the loser
#"loser_hand" Does the player play right or left handed
#"loser_ht" Height of the loser in meters
# "loser_ioc" Country ioc
# "loser_age" Age of the loser
# "score" The score of the match
# "best_of" 3 = 3sets, 5 = 5sets, 1 = 1set?(Denver)
# "round" F = Final, SF = semifinal, QF = quartfinal, R16 = round of 16,
# R32 = round of 32, R64 = round of 64, R128 = round of 128, RR = round robin
# "minutes" Returns in minutes how long the match was.
# "w_ace"  Returns the number of aces the winner hit.
# "w_df"   Returns the number of double faults the winner hit.
# "w_svpt" Number of points won on serve in total
# "w_1stIn" Number of points
# "w_1stWon" Number of first serve points won
# "w_2ndWon" Number of 2nd serve points won
# "w_SvGms" Number of service games won during the match
# "w_bpSaved" Number of break points saved on serve for the winner
# "w_bpFaced" Number of break points faced on serve for the winner
# "l_ace" Number of aces hit by winner
# "l_df" Number of double faults hit by loser
# "l_svpt" Number of points won on serve in the match
# "l_1stIn" Number of of first serve's in
# "l_1stWon" Number of first serve points won
#"l_2ndWon" Number of second serve points won
# "l_SvGms" Number of service games won for the loser
# "l_bpSaved" Number of break points saved as the loser
# "l_bpFaced" Number of break points faced as the loser
# "winner_rank" The Atp ranking of the player at the start of the tournament. Integer
# "winner_rank_points" How many ranking points the player has at the start of the tournament.
# "loser_rank" Same as winner rank
# "loser_rank_points" Same as winner_rank_points
```

Combining the data sets into one master frame

For right now, we have a lot of datasets, but we want to combine them. Here we begin that process.

```
Atp_together_68_69 <- full_join(Atp_singles_1968, Atp_singles_1969)
Atp_together_68_70 <- full_join(Atp_together_68_69, Atp_singles_1970)
Atp_together_68_71 <- full_join(Atp_together_68_70, Atp_singles_1971)
Atp_together_68_72 <- full_join(Atp_together_68_71, Atp_singles_1972)
Atp_together_68_73 <- full_join(Atp_together_68_72, Atp_singles_1973)
Atp_together_68_74 <- full_join(Atp_together_68_73, Atp_singles_1974)
Atp_together_68_75 <- full_join(Atp_together_68_74, Atp_singles_1975)
Atp_together_68_76 <- full_join(Atp_together_68_75, Atp_singles_1976)
Atp_together_68_77 <- full_join(Atp_together_68_76, Atp_singles_1977)
Atp_together_68_78 <- full_join(Atp_together_68_77, Atp_singles_1978)
Atp_together_68_79 <- full_join(Atp_together_68_78, Atp_singles_1979)
Atp_together_68_80 <- full_join(Atp_together_68_79, Atp_singles_1980)
Atp_together_68_81 <- full_join(Atp_together_68_80, Atp_singles_1981)
Atp_together_68_82 <- full_join(Atp_together_68_81, Atp_singles_1982)
Atp_together_68_83 <- full_join(Atp_together_68_82, Atp_singles_1983)
Atp_together_68_84 <- full_join(Atp_together_68_83, Atp_singles_1984)
Atp_together_68_85 <- full_join(Atp_together_68_84, Atp_singles_1985)
Atp_together_68_86 <- full_join(Atp_together_68_85, Atp_singles_1986)
Atp_together_68_87 <- full_join(Atp_together_68_86, Atp_singles_1987)
Atp_together_68_88 <- full_join(Atp_together_68_87, Atp_singles_1988)
Atp_together_68_89 <- full_join(Atp_together_68_88, Atp_singles_1989)
Atp_together_68_90 <- full_join(Atp_together_68_89, Atp_singles_1990)
Atp_together_68_91 <- full_join(Atp_together_68_90, Atp_singles_1991)
Atp_together_68_92 <- full_join(Atp_together_68_91, Atp_singles_1992)
Atp_together_68_93 <- full_join(Atp_together_68_92, Atp_singles_1993)
Atp_together_68_94 <- full_join(Atp_together_68_93, Atp_singles_1994)
Atp_together_68_95 <- full_join(Atp_together_68_94, Atp_singles_1995)
Atp_together_68_96 <- full_join(Atp_together_68_95, Atp_singles_1996)
Atp_together_68_97 <- full_join(Atp_together_68_96, Atp_singles_1997)
Atp_together_68_98 <- full_join(Atp_together_68_97, Atp_singles_1998)
Atp_together_68_99 <- full_join(Atp_together_68_98, Atp_singles_1999)
Atp_together_68_00 <- full_join(Atp_together_68_99, Atp_singles_2000)
Atp_together_68_01 <- full_join(Atp_together_68_00, Atp_singles_2001)
Atp_together_68_02 <- full_join(Atp_together_68_01, Atp_singles_2002)
Atp_together_68_03 <- full_join(Atp_together_68_02, Atp_singles_2003)
Atp_together_68_04 <- full_join(Atp_together_68_03, Atp_singles_2004)
Atp_together_68_05 <- full_join(Atp_together_68_04, Atp_singles_2005)
Atp_together_68_06 <- full_join(Atp_together_68_05, Atp_singles_2006)
Atp_together_68_07 <- full_join(Atp_together_68_06, Atp_singles_2007)
Atp_together_68_08 <- full_join(Atp_together_68_07, Atp_singles_2008)
Atp_together_68_09 <- full_join(Atp_together_68_08, Atp_singles_2009)
Atp_together_68_10 <- full_join(Atp_together_68_09, Atp_singles_2010)
Atp_together_68_11 <- full_join(Atp_together_68_10, Atp_singles_2011)
Atp_together_68_12 <- full_join(Atp_together_68_11, Atp_singles_2012)
Atp_together_68_13 <- full_join(Atp_together_68_12, Atp_singles_2013)
Atp_together_68_14 <- full_join(Atp_together_68_13, Atp_singles_2014)
Atp_together_68_15 <- full_join(Atp_together_68_14, Atp_singles_2015)
Atp_together_68_16 <- full_join(Atp_together_68_15, Atp_singles_2016)
Atp_together_68_17 <- full_join(Atp_together_68_16, Atp_singles_2017)
Atp_together_68_18 <- full_join(Atp_together_68_17, Atp_singles_2018)
```

```
Atp_together_68_19 <- full_join(Atp_together_68_18, Atp_singles_2019)
Atp_together_68_20 <- full_join(Atp_together_68_19, Atp_singles_2020)
Atp_together_68_21 <- full_join(Atp_together_68_20, Atp_singles_2021)
```

Data Join Problems

Data from 2022 and 2023 can't currently join, thus we need to clean the data to make sure the data can be joined. Can't join winner_seed variable because it's double and other is character. The code below fixes it for 2022.

```
#Exploring the winner_seed variable
# We see that WC, Q, LL are included in the winner_seed when in fact,
#they should be on winner_entry
unique(Atp_singles_2022$winner_seed)
```

```
##  [1] NA   "1"  "2"  "3"  "6"  "7"  "8"  "4"  "5"  "25" "17" "16" "19" "31" "WC"
## [16] "Q"  "23" "14" "10" "18" "28" "32" "11" "15" "20" "26" "27" "24" "9"  "13"
## [31] "12" "33" "29" "30" "21" "22" "LL"
```

```
#This fixes the WC problem
Atp_singles_2022 <- Atp_singles_2022 %>%
  mutate(winner_entry = ifelse(is.na(winner_entry),"WC",winner_entry)) %>%
  mutate(winner_seed=ifelse(winner_seed=="WC",NA,winner_seed))

# This fixes the Q problem
Atp_singles_2022 <- Atp_singles_2022 %>%
  mutate(winner_entry = ifelse(is.na(winner_entry),"Q",winner_entry)) %>%
  mutate(winner_seed=ifelse(winner_seed=="Q",NA,winner_seed))

# This fixes the LL problem
Atp_singles_2022 <- Atp_singles_2022 %>%
  mutate(winner_entry = ifelse(is.na(winner_entry),"LL",winner_entry)) %>%
  mutate(winner_seed=ifelse(winner_seed=="LL",NA,winner_seed))

# This shows that only numbers and NA are now in the winner_seed
# Thus we don't have to worry about WC, LL, Q anymore
unique(Atp_singles_2022$winner_seed)
```

```
##  [1] NA   "1"  "2"  "3"  "6"  "7"  "8"  "4"  "5"  "25" "17" "16" "19" "31" "23"
## [16] "14" "10" "18" "28" "32" "11" "15" "20" "26" "27" "24" "9"  "13" "12" "33"
## [31] "29" "30" "21" "22"
```

```
# Need to change the winner_seed to numeric
Atp_singles_2022 <- Atp_singles_2022 %>%
  mutate(winner_seed = as.numeric(winner_seed))

# Here we look at the loser_seed
# LL, Q, WC are included when they shouldn't be
unique(Atp_singles_2022$loser_seed)
```

```
##  [1] NA   "2"  "3"  "6"  "7"  "4"  "8"  "5"  "1"  "LL" "Q"  "WC" "12" "30" "21"
## [16] "22" "29" "10" "26" "13" "25" "16" "31" "23" "18" "28" "15" "24" "19" "32"
## [31] "20" "27" "17" "14" "11" "9"  "33"
```

```
#This fixes the WC problem in loser_seed
Atp_singles_2022 <- Atp_singles_2022 %>%
  mutate(loser_entry = ifelse(is.na(loser_entry),"WC",loser_entry)) %>%
  mutate(loser_seed=ifelse(loser_seed=="WC",NA,loser_seed))

# This fixes the Q problem in loser_seed
Atp_singles_2022 <- Atp_singles_2022 %>%
  mutate(loser_entry = ifelse(is.na(loser_entry),"Q",loser_entry)) %>%
  mutate(loser_seed=ifelse(loser_seed=="Q",NA,loser_seed))

# This fixes the LL problem in loser_seed
Atp_singles_2022 <- Atp_singles_2022 %>%
  mutate(loser_entry = ifelse(is.na(loser_entry),"LL",loser_entry)) %>%
  mutate(loser_seed=ifelse(loser_seed=="LL",NA,loser_seed))

# We see that there are just numbers and NA, thus it worked!
unique(Atp_singles_2022$loser_seed)
```

```
##  [1] NA   "2"  "3"  "6"  "7"  "4"  "8"  "5"  "1"  "12" "30" "21" "22" "29" "10"
## [16] "26" "13" "25" "16" "31" "23" "18" "28" "15" "24" "19" "32" "20" "27" "17"
## [31] "14" "11" "9"  "33"
```

```
# This changes the loser_seed to numeric
Atp_singles_2022 <- Atp_singles_2022 %>%
  mutate(loser_seed = as.numeric(loser_seed))

# The data is successfully joined!
Atp_together_68_22 <- full_join(Atp_together_68_21, Atp_singles_2022)
Atp_together_68_22 %>%
  filter(tourney_level == "G") %>%
  filter(round == "F") %>%
  select (tourney_name, winner_name, tourney_date) %>%
  arrange(tourney_date) %>%
  tail()
```

```
## # A tibble: 6 x 3
##   tourney_name    winner_name     tourney_date
##   <chr>           <chr>                  <dbl>
## 1 Wimbledon       Novak Djokovic      20210628
## 2 Us Open         Daniil Medvedev     20210830
## 3 Australian Open Rafael Nadal        20220117
## 4 Roland Garros   Rafael Nadal        20220523
## 5 Wimbledon       Novak Djokovic      20220627
## 6 Us Open         Carlos Alcaraz      20220829
```

Correcting the 2023 data

The 2023 data has the same problem as the 2022, thus I did the same steps.

```
# We have the same problem in joining the tables as the 2022 data with the 2023
#Exploring the winner_seed variable
# We see that NA, WC, Q, LL are included in the winner_seed when in fact, they should be on winner_entr
unique(Atp_singles_2023$winner_seed)
```

```
##  [1] "3"  NA   "1"  "5"  "15" "16" "2"  "9"  "14" "7"  "18" "8"  "6"  "4"  "Q"
## [16] "31" "WC" "10" "20" "29" "32" "11" "28" "25" "LL" "22" "27" "23" "12" "24"
## [31] "30"
```

```
#This fixes the WC problem
Atp_singles_2023 <- Atp_singles_2023 %>%
  mutate(winner_entry = ifelse(is.na(winner_entry),"WC",winner_entry)) %>%
  mutate(winner_seed=ifelse(winner_seed=="WC",NA,winner_seed))

# This fixes the Q problem
Atp_singles_2023 <- Atp_singles_2023 %>%
  mutate(winner_entry = ifelse(is.na(winner_entry),"Q",winner_entry)) %>%
  mutate(winner_seed=ifelse(winner_seed=="Q",NA,winner_seed))

# This fixes the LL problem
Atp_singles_2023 <- Atp_singles_2023 %>%
  mutate(winner_entry = ifelse(is.na(winner_entry),"LL",winner_entry)) %>%
  mutate(winner_seed=ifelse(winner_seed=="LL",NA,winner_seed))

# We see now that only NA and numbers are in the winner_seed
unique(Atp_singles_2023$winner_seed)
```

```
##  [1] "3"  NA   "1"  "5"  "15" "16" "2"  "9"  "14" "7"  "18" "8"  "6"  "4"  "31"
## [16] "10" "20" "29" "32" "11" "28" "25" "22" "27" "23" "12" "24" "30"
```

```
# This changes the winner_seed variable to numeric
Atp_singles_2023 <- Atp_singles_2023 %>%
  mutate(winner_seed = as.numeric(winner_seed))

# Trying to join the tables together, but now it doesn't work because of loser_seed
# Thus we need to fix looser_seed

# We see that there are, Q, LL and WC which should be in loser_entry not seed
unique(Atp_singles_2023$loser_seed)
```

```
##  [1] "5"  NA   "2"  "8"  "3"  "12" "16" "9"  "13" "10" "14" "4"  "7"  "11" "18"
## [16] "6"  "17" "1"  "Q"  "LL" "WC" "21" "26" "32" "23" "30" "20" "28" "25" "27"
## [31] "31" "15" "22" "24" "29"
```

```
#This fixes the WC problem in loser_seed
Atp_singles_2023 <- Atp_singles_2023 %>%
  mutate(loser_entry = ifelse(is.na(loser_entry),"WC",loser_entry)) %>%
  mutate(loser_seed=ifelse(loser_seed=="WC",NA,loser_seed))

# This fixes the Q problem
Atp_singles_2023 <- Atp_singles_2023 %>%
```

```r
  mutate(loser_entry = ifelse(is.na(loser_entry),"Q",loser_entry)) %>%
  mutate(loser_seed=ifelse(loser_seed=="Q",NA,loser_seed))

# This fixes the LL problem in loser_seed
Atp_singles_2023 <- Atp_singles_2023 %>%
  mutate(loser_entry = ifelse(is.na(loser_entry),"LL",loser_entry)) %>%
  mutate(loser_seed=ifelse(loser_seed=="LL",NA,loser_seed))

# We see that there are just numbers and NA, thus it worked!
unique(Atp_singles_2023$loser_seed)
```

```
##  [1] "5"  NA   "2"  "8"  "3"  "12" "16" "9"  "13" "10" "14" "4"  "7"  "11" "18"
## [16] "6"  "17" "1"  "21" "26" "32" "23" "30" "20" "28" "25" "27" "31" "15" "22"
## [31] "24" "29"
```

```r
# This changes the loser_seed to numeric
Atp_singles_2023 <- Atp_singles_2023 %>%
  mutate(loser_seed = as.numeric(loser_seed))

# The data is successfuly joined!
Atp_together_68_23 <- full_join(Atp_together_68_22, Atp_singles_2023)
Atp_together_68_23 %>%
  filter(tourney_level == "G") %>%
  filter(round == "F") %>%
  select (tourney_name, winner_name, tourney_date) %>%
  arrange(tourney_date) %>%
  tail()
```

```
## # A tibble: 6 x 3
##   tourney_name    winner_name     tourney_date
##   <chr>           <chr>                  <dbl>
## 1 Us Open         Daniil Medvedev     20210830
## 2 Australian Open Rafael Nadal        20220117
## 3 Roland Garros   Rafael Nadal        20220523
## 4 Wimbledon       Novak Djokovic      20220627
## 5 Us Open         Carlos Alcaraz      20220829
## 6 Australian Open Novak Djokovic      20230116
```

Cleaning the Data

Cleaning the Grandslams

```r
tennis_mf <- Atp_together_68_23

# This function returns all the unique GS names
tennis_mf %>%
  filter(round == "F", tourney_level == "G") %>%
  select(tourney_name) %>%
  unique()
```

```
## # A tibble: 7 x 1
##   tourney_name
##   <chr>
## 1 Roland Garros
## 2 Wimbledon
## 3 US Open
## 4 Australian Chps.
## 5 Australian Open
## 6 Australian Open-2
## 7 Us Open
```

```
# This fixes the US Open
tennis_mf <- tennis_mf %>%
    mutate(tourney_name = str_replace_all(tourney_name, fixed("Us Open"), "US Open"))

#Fixes part of the Australian Open
tennis_mf <- tennis_mf %>%
    mutate(tourney_name = str_replace_all(tourney_name, fixed("Australian Chps."), "Australian Open"))

# This fixes the other part of the Australian Open
tennis_mf<- tennis_mf %>%
    mutate(tourney_name = str_replace_all(tourney_name, fixed("Australian Open-2"), "Australian Open"))

# This shows that the above functions worked
# We now just have 4 names for the grand slams
tennis_mf %>%
  filter(round == "F", tourney_level == "G") %>%
  select(tourney_name) %>%
  unique()
```

```
## # A tibble: 4 x 1
##   tourney_name
##   <chr>
## 1 Roland Garros
## 2 Wimbledon
## 3 US Open
## 4 Australian Open
```

Cleaning the M100s Tournaments

```
# Now we need to see are there different names for Masters tournaments
# We find out that there are repeat names that need to be changed
tennis_mf %>%
  filter(round == "F", tourney_level == "M") %>%
  select(tourney_name) %>%
  unique() %>%
  print(n=Inf) # This prints all the infinite rows, because tibble just prints the first 10 rows
```

```
## # A tibble: 23 x 1
##    tourney_name
##    <chr>
##  1 Rome
##  2 Cincinnati
##  3 Toronto
##  4 Montreal / Toronto
##  5 Masters
##  6 Delray Beach
##  7 Boca West
##  8 Masters Dec
##  9 Key Biscayne
## 10 Indian Wells
## 11 Indian Wells Masters
## 12 Miami Masters
## 13 Monte Carlo Masters
## 14 Hamburg Masters
## 15 Rome Masters
## 16 Canada Masters
## 17 Cincinnati Masters
## 18 Stockholm Masters
## 19 Paris Masters
## 20 Essen Masters
## 21 Stuttgart Masters
## 22 Madrid Masters
## 23 Shanghai Masters
```

```
# Fixes the Rome Masters
tennis_mf <- tennis_mf %>%
    mutate(tourney_name = str_replace_all(tourney_name, fixed("Rome"), "Rome Masters")) %>%
    mutate(tourney_name = str_replace_all(tourney_name, fixed("Rome Masters Masters"), "Rome Masters" )

# This fixes the Cincy tournament
tennis_mf <- tennis_mf %>%
    mutate(tourney_name = str_replace_all(tourney_name, fixed("Cincinnati"), "Cincinnati Masters")) %>%
    mutate(tourney_name = str_replace_all(tourney_name, fixed("Cincinnati Masters Masters"), "Cincinnat

# This fixes the Canada Masters 1000 tournament names
tennis_mf <- tennis_mf %>%
    mutate(tourney_name = str_replace_all(tourney_name, fixed("Montreal / Toronto"), "Canada Masters"))
    mutate(tourney_name = str_replace_all(tourney_name, fixed("Canada Masters Masters"), "Canada Master
    mutate(tourney_name = str_replace_all(tourney_name, fixed("Toronto"), "Canada Masters"))

# This fixes the Indian Wells tournament
tennis_mf <- tennis_mf %>%
    mutate(tourney_name = str_replace_all(tourney_name, fixed("Indian Wells"), "Indian Wells Masters"))
    mutate(tourney_name = str_replace_all(tourney_name, fixed("Indian Wells Masters Masters"), "Indian 

# This fixes the Key Biscayne and turns it into the Miami Masters tournament
tennis_mf <- tennis_mf %>%
    mutate(tourney_name = str_replace_all(tourney_name, fixed("Key Biscayne"), "Miami Masters")) %>%
    mutate(tourney_name = str_replace_all(tourney_name, fixed("Miami Masters Masters"), "Miami Masters"
```

Cleaning the Winner Entry

```
# We see that there are two spellings for Alternate: ALT and Alt, thus we need
# to make sure there is only one spelling of Alternate
unique(tennis_mf$winner_entry)
```

```
## [1] NA    "Q"   "LL"  "WC"  "PR"  "SE"  "ALT" "Alt"
```

```
tennis_mf <- tennis_mf %>%
  mutate(tourney_name = str_replace_all(winner_entry, fixed("Alt"), "ALT"))

# Here we see that it worked
unique(tennis_mf$winner_entry)
```

```
## [1] NA    "Q"   "LL"  "WC"  "PR"  "SE"  "ALT" "Alt"
```

Data Analysis

Grand Slam Performance

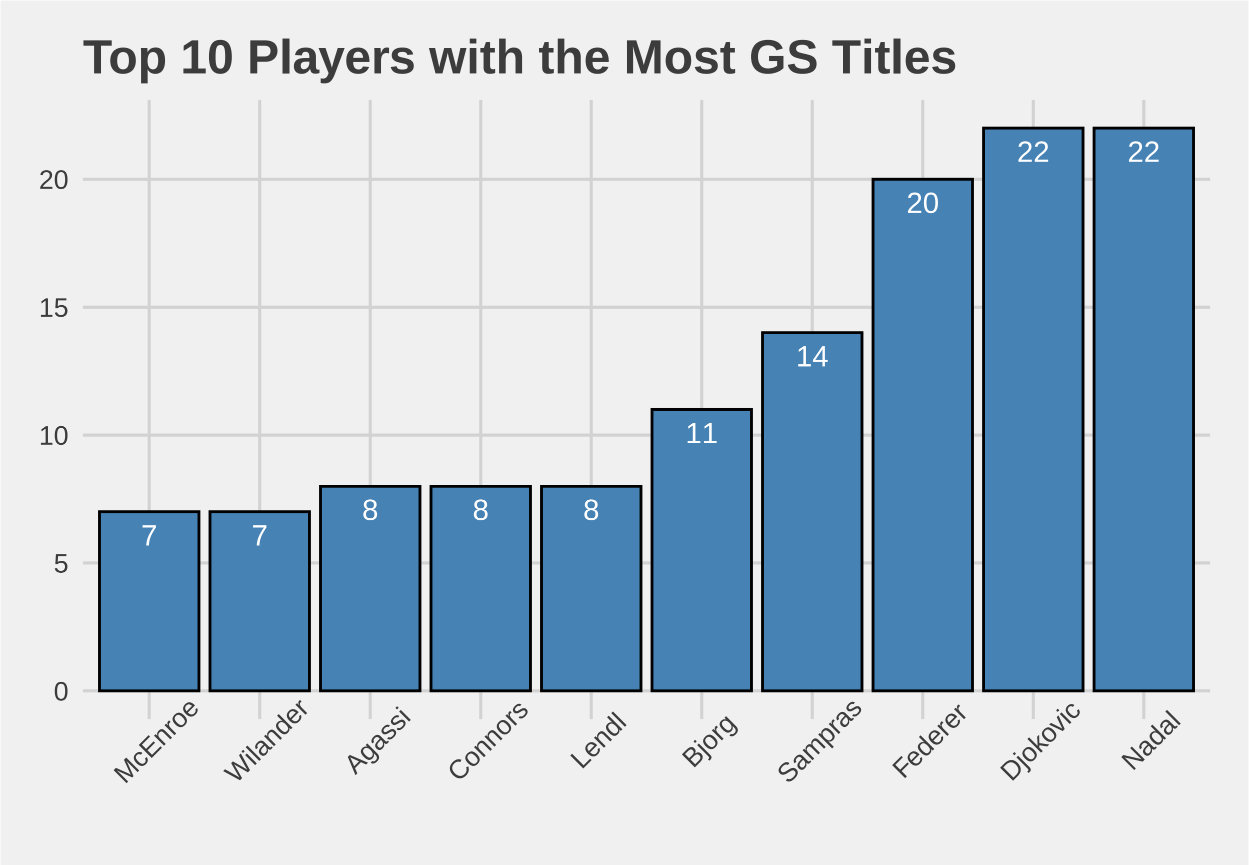
Likely the most common indicator to determine greatness is the performance in the four grandslams (Australian Open, Roland Garros, Wimbledon and US Open).

```
# Here will produce who won the most grandslams
# This variable counts the most grandslams
gs_count <- tennis_mf %>%
  filter(tourney_level == "G", round == "F") %>%
  count(winner_name) %>%
  arrange(desc(n)) %>%
  head(10)

# I would like it to just inlcude the last name of the players
# For there isn't enough space on the graph for first and last name
gs_count10 <- gs_count %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Novak Djokovic"), "Djokovic")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Rafael Nadal"), "Nadal")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Roger Federer"), "Federer")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Pete Sampras"), "Sampras")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Bjorn Borg"), "Bjorg")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Andre Agassi"), "Agassi")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Ivan Lendl"), "Lendl")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Jimmy Connors"), "Connors")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("John McEnroe"), "McEnroe")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Mats Wilander"), "Wilander"))



ggplot(gs_count10, aes(reorder(winner_name, n), n)) +
  geom_bar(stat = "identity", color = "black", fill = "steelblue") +
  ggtitle("Top 10 Players with the Most GS Titles") +
  theme_fivethirtyeight() +
  theme(axis.text.x = element_text(angle = 45)) + # This makes the names slanted
  geom_text(aes(label=n), vjust = 1.6, color = "white") # This includes the number inside the chart
```



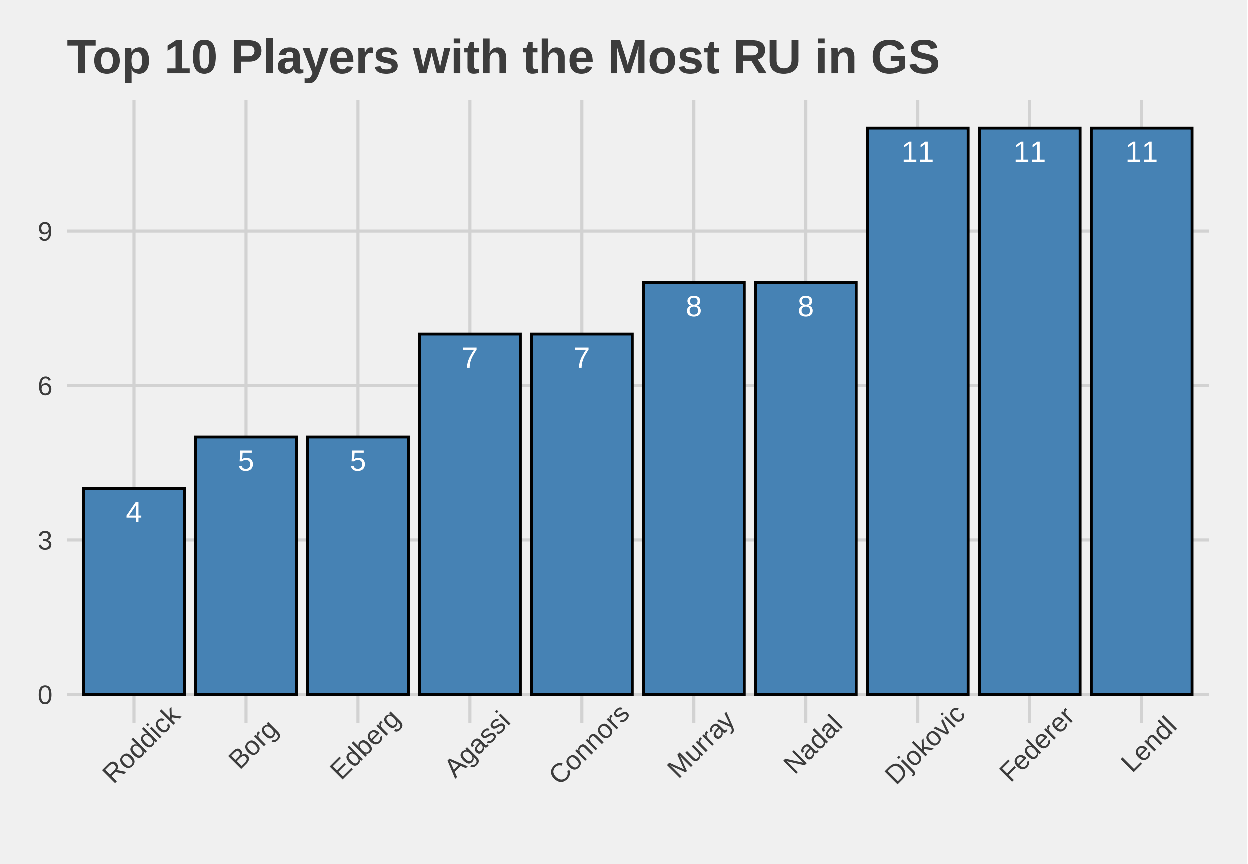
We see from the graph that Nadal, Federer and Djokovic are head and shoulders above the competition when it comes to GS won.

Runner Up at GS Making a finals apperance at a grandslam is also an accomplishment, thus, let's explore the players with the most RU at GS

```
ru <- tennis_mf %>%
  filter(round == "F", tourney_level == "G") %>%
  count(loser_name) %>%
  arrange(desc(n)) %>%
  head(10)

# Only includes the last name
ru_short <- ru %>%
  mutate(loser_name = str_replace_all(loser_name, fixed("Ivan Lendl"), "Lendl")) %>%
  mutate(loser_name = str_replace_all(loser_name, fixed("Novak Djokovic"), "Djokovic")) %>%
  mutate(loser_name = str_replace_all(loser_name, fixed("Roger Federer"), "Federer")) %>%
  mutate(loser_name = str_replace_all(loser_name, fixed("Andy Murray"), "Murray")) %>%
  mutate(loser_name = str_replace_all(loser_name, fixed("Rafael Nadal"), "Nadal")) %>%
  mutate(loser_name = str_replace_all(loser_name, fixed("Andre Agassi"), "Agassi")) %>%
  mutate(loser_name = str_replace_all(loser_name, fixed("Jimmy Connors"), "Connors")) %>%
  mutate(loser_name = str_replace_all(loser_name, fixed("Bjorn Borg"), "Borg")) %>%
  mutate(loser_name = str_replace_all(loser_name, fixed("Stefan Edberg"), "Edberg")) %>%
  mutate(loser_name = str_replace_all(loser_name, fixed("Andy Roddick"), "Roddick"))

# Creates the plot
ggplot(ru_short, aes(reorder(loser_name, n), n)) +
  geom_bar(stat = "identity", color = "black", fill = "steelblue") +
  theme_fivethirtyeight() +
  ggtitle("Top 10 Players with the Most RU in GS") +
  theme(axis.text.x = element_text(angle = 45)) + # This makes the names slanted so it's easier to read
  geom_text(aes(label=n), vjust = 1.6, color = "white")
```



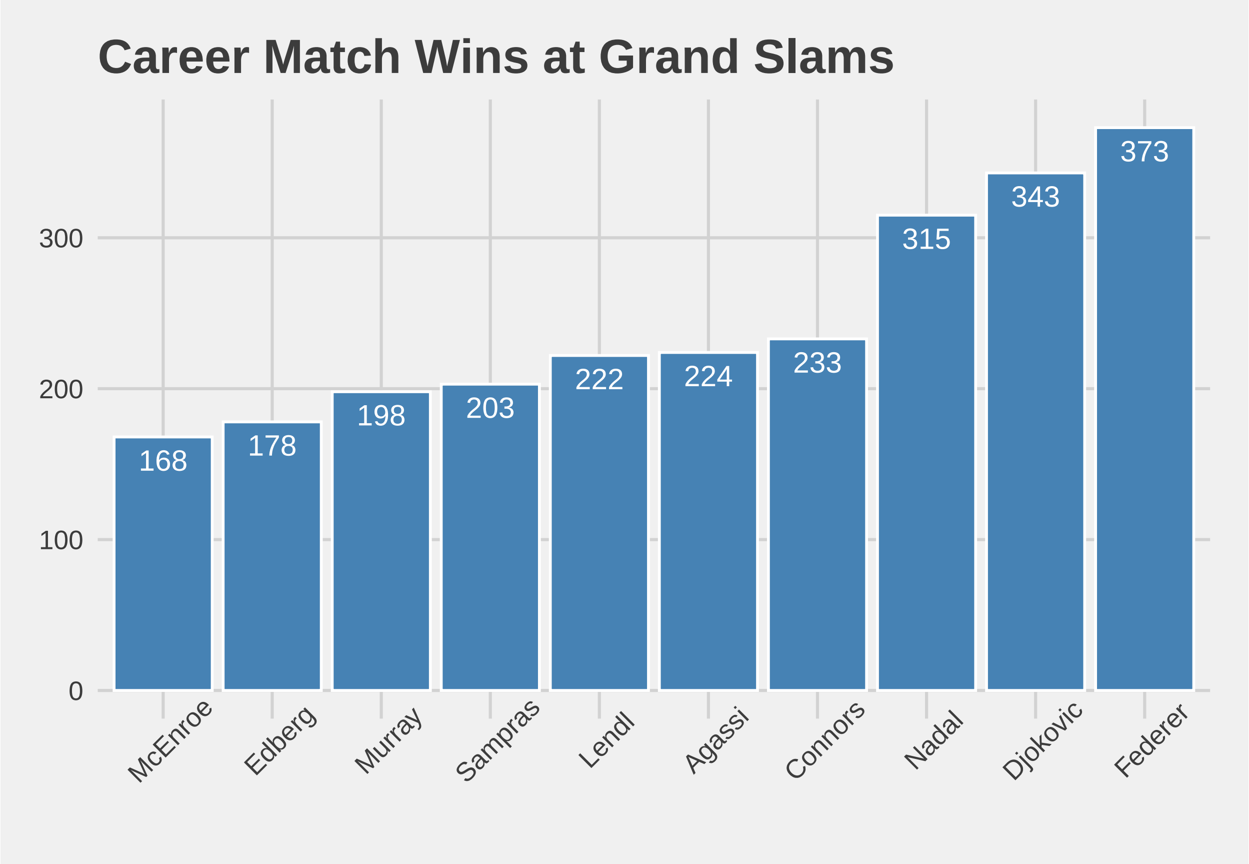
An interesting thing to note from this graph is that Sampras who has 14 GS, isn't even in the top 10 in RU in GS. He has 14 GS and only 4 RU appearances. Roddick Saldy is on the other end, only winning one GS and 4RU for he ran into Federer every time.

Career Wins at GS

```
gs_all_win <-tennis_mf %>%
  filter(tourney_level == "G") %>%
  count(winner_name) %>%
  arrange(desc(n)) %>%
  head(10)

# Only includes the last name
gs_all_win <- gs_all_win %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Roger Federer"), "Federer")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Novak Djokovic"), "Djokovic")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Rafael Nadal"), "Nadal")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Jimmy Connors"), "Connors")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Andre Agassi"), "Agassi")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Ivan Lendl"), "Lendl")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Pete Sampras"), "Sampras")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Andy Murray"), "Murray")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Stefan Edberg"), "Edberg")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("John McEnroe"), "McEnroe"))

# Creates the graph
ggplot(gs_all_win, aes(reorder(winner_name, n), n)) +
  geom_bar(stat = "identity", color = "white", fill = "steelblue") +
  theme_fivethirtyeight() +
  ggtitle("Career Match Wins at Grand Slams") +
  geom_text(aes(label=n), vjust = 1.6, color = "white") +
  theme(axis.text.x = element_text(angle = 45))
```



We see that Federer, Djokovic and Nadal (Big 3) have the most wins at the grandslams. They have a considerable lead over the player in 4th position, Connors. It's also interesting to note that Connors has more wins at GS than Sampras yet less GS. This is because Connors played more years on the tour while Sampras retired relatively young.

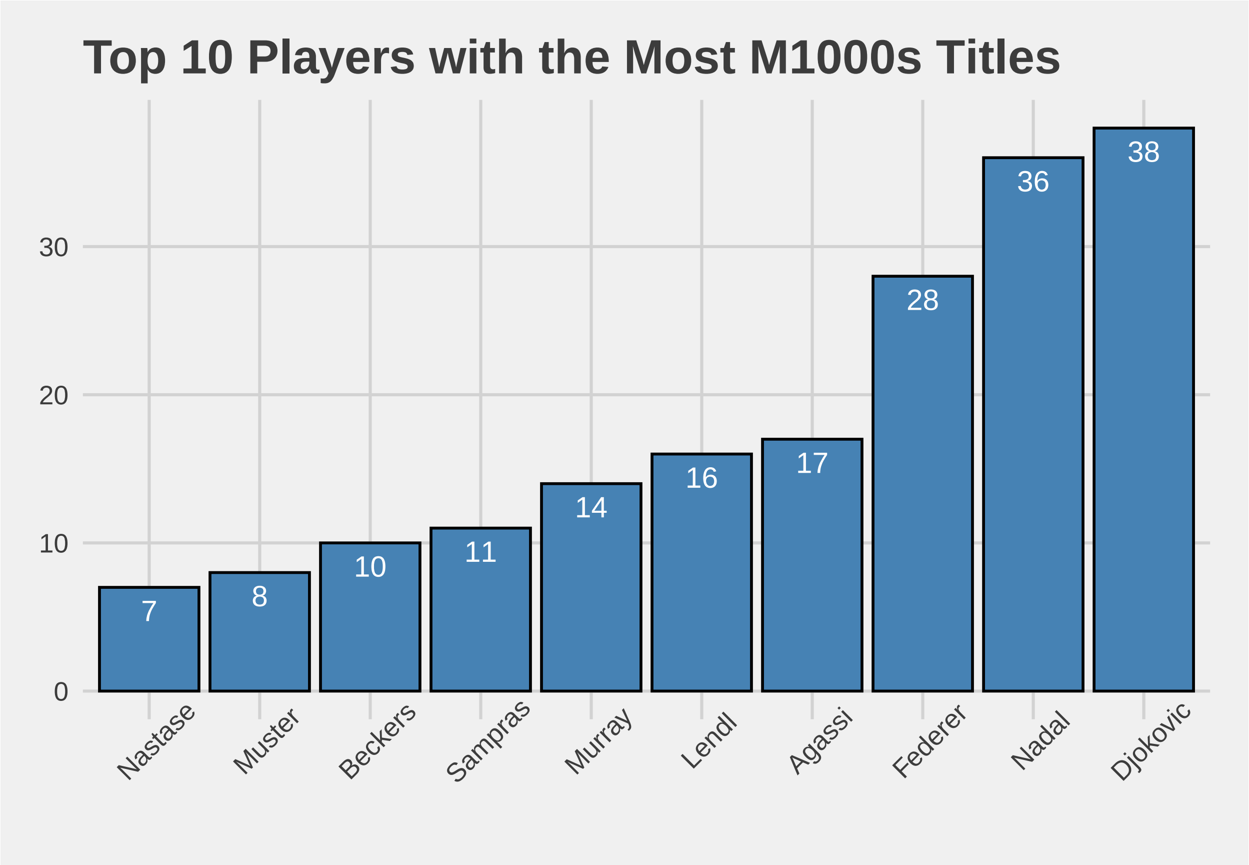
Master 1000s Performance

After the Grand Slams, the biggest tournaments are the Masters 1000s

```
# This gets the top 10 players with masters 1000s titles
ms_count <- tennis_mf %>%
  filter(round == "F", tourney_level == "M") %>%
  select(winner_name) %>%
  count(winner_name) %>%
  arrange(desc(n)) %>%
  head(10)

# This cuts out the first name
ms_count_10 <- ms_count %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Novak Djokovic"), "Djokovic")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Rafael Nadal"), "Nadal")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Roger Federer"), "Federer")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Andre Agassi"), "Agassi")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Ivan Lendl"), "Lendl")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Andy Murray"), "Murray")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Pete Sampras"), "Sampras")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Boris Becker"), "Beckers")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Thomas Muster"), "Muster")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Ilie Nastase"), "Nastase"))

# Now this makes the graph
# I want to add some colors to the graph
# Trying to add a different color for every name was too much
# I like the outline black and the color steel blue. Looks good to me
ms_graph <- ggplot(ms_count_10, aes(reorder(winner_name, n), n)) +
    geom_bar(stat = "identity",  color = "black", fill = "steelblue") +
    theme_fivethirtyeight() +
    ggtitle("Top 10 Players with the Most M1000s Titles") +
    theme(axis.text.x = element_text(angle = 45)) +
    geom_text(aes(label=n), vjust = 1.6, color = "white")
ms_graph
```



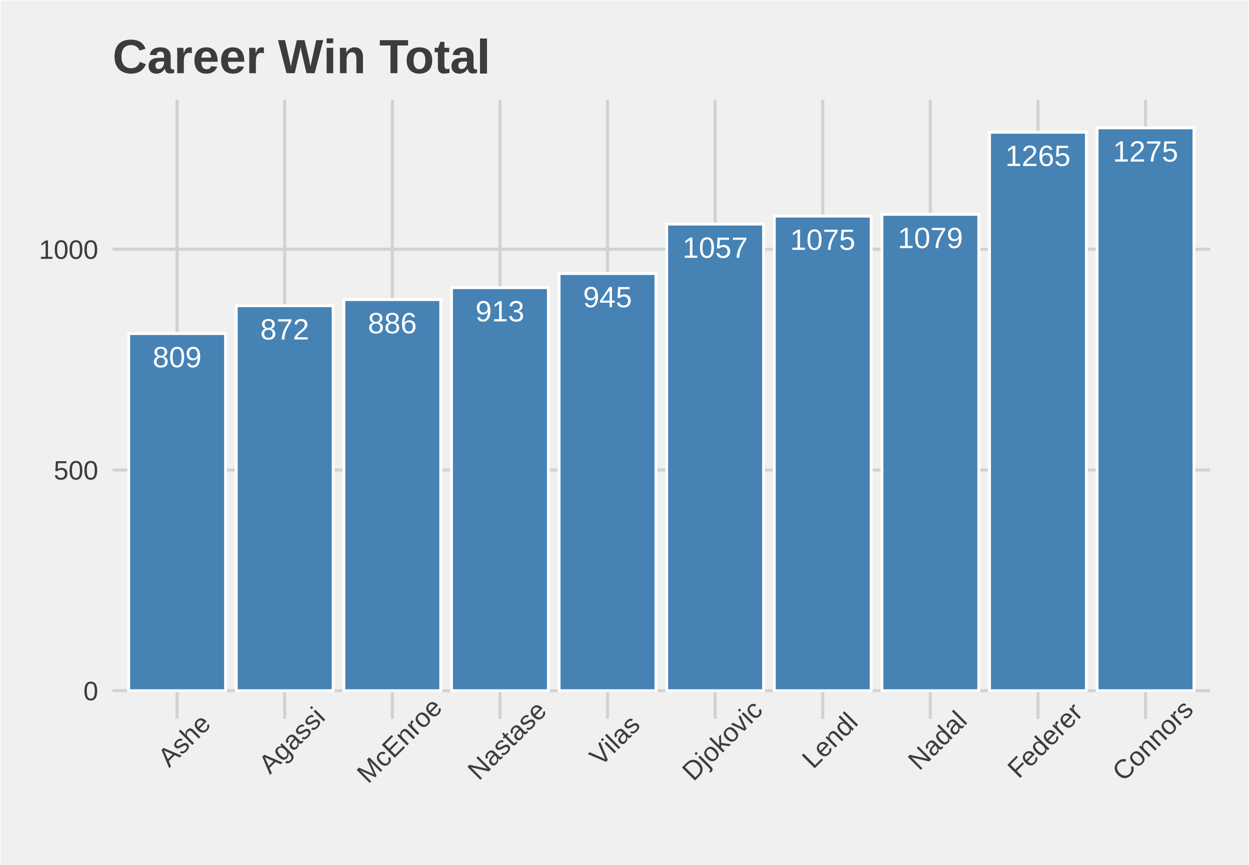
We see that Djokovic has the most Masters 1000s Titles

Career Match Wins

```
# This creates a _ with the top 10 matches won of all time
all_time_wins <- tennis_mf %>%
  count(winner_name) %>%
  arrange(desc(n)) %>%
  head(10)

# Now only including the last name of the player
all_time_wins_short <- all_time_wins %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Jimmy Connors"), "Connors")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Roger Federer"), "Federer")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Rafael Nadal"), "Nadal")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Ivan Lendl"), "Lendl")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Novak Djokovic"), "Djokovic")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Guillermo Vilas"), "Vilas")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Ilie Nastase"), "Nastase")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("John McEnroe"), "McEnroe")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Andre Agassi"), "Agassi")) %>%
  mutate(winner_name = str_replace_all(winner_name, fixed("Arthur Ashe"), "Ashe"))

# Now making the graph
ggplot(all_time_wins_short, aes(reorder(winner_name, n), n)) +
  geom_bar(stat = "identity", color = "white", fill = "steelblue") +
  theme_fivethirtyeight() +
  ggtitle("Career Win Total") +
  geom_text(aes(label=n), vjust = 1.6, color = "white") +
  theme(axis.text.x = element_text(angle = 45))
```



Here we see that Connors has the most match wins of all time followed by Federer, Nadal and then Lendl.

Conclusion

I will continue this analysis to include rankings as well.

- Djokovic is the GOAT
  - Is currently tied with Nadal for most Grand Slam Titles
  - Has the most Master 1000s Titles
  - Is playing great tennis as of 2023, thus likely to acquire more big titles