Analyzing Traits and Tendencies in Tennis Players

Caleb Skinner

Table of contents

Abstract

I assess and quantify the underlying tendencies and traits of tennis players using dimension reduction techniques. With Grand Slam tennis data, I conduct principle component analysis to reduce 67 variables into a few interpretable principle components. These principle components follow reasonable interpretations and describe the strategies and traits of famous tennis players well. From these results, I suggest more advanced and thorough methods for analyzing tennis players. These findings should lead to more predictive and causal research on the strategies that tennis players employ.

Introduction

Tennis is a competitive individual sport. Players possess drastically different skill sets and employ various strategies. Six foot 11 American giant John Isner and 5 foot 7 Argentinean speedster Diego Schwartzman may play the same game with the same rules, but their style and objectives differ completely. Many of these strategies and characteristics are noticeable intuitively, but it can be difficult to quantify. My objective is to identify the underlying traits and tendencies that differentiate players. Is it possible to quantify the similarities between players in different contexts?

These questions possess enormous strategic and predictive implications. Players and coaches who understand the overarching traits and tendencies of their opponent can develop and adjust strategies beyond simple intuition or stereotypes. Moreover, identifying these traits and tendencies can play a significant role in the development of models predicting player performance.

Moreover, research on players' underlying traits and tendencies can greatly enhance our understanding of how players react to other events. I developed this project to assist my research on the momentum effect in tennis. Without research on players' underlying traits and tendencies, we have to assume that all players react to potential catalysts for momentum in the same way.

I acquired point-level data of Grand Slam tennis matches from 2011-2023 from Jeff Sackmann. I transformed the data into a player-level and summarized many important statistics for each player. I included players who played four matches or more. From there, I standardized each of the variables to give each variable equal weight in the analysis. I conducted principle component analysis to reduce the dimensions of the data.

The first few principle components explain much of the variance in the data. The first principle component measures the serve strength of the player and is also highly associated with the sex and height of the player. The second principal component measures the players' overall dominance. The third principle component measures a players' carelessness, and the fourth principle component measures a players' propensity to play long points. I assessed the principle components on a few famous tennis players and the results are convincing.

Overall, the analysis served to identify several important traits and tendencies within players. This is an encouraging first result from a simple approach. In the future, more advanced methods should be taken, and more nuanced questions should be asked.

Data

Each year, there are hundreds of professional tennis tournaments awarding millions of dollars in prize money. Yet, the pinnacle of the tennis calendar is falls on four Grand Slam events. These four tournaments – the Australian Open (held in January), French Open (May), Wimbledon (July), and US Open (September) – offer players the most ranking points, publicity, prize money, and prestige. Players always have incentive to exert effort towards victory, but the incentives are never higher than in the Grand Slams. Accordingly, players structure their complex schedules of training, recovery, and competition in order to maximize their performance at these four events. Grand Slam events are relatively rare and valuable, so all players should be have incentive to put forth substantial effort. Moreover, Grand Slam events possess significantly more resources and content than smaller tournaments. The Grand Slam events each partner with tracking services like IBM or Infosys. This allows for accurate and detailed information of each point.

I acquired point-by-point tennis data from Jeff Sackmann's github page titled "Grand Slam Point-by-Point Data, 2011-present." He scraped the data "from the four Grand Slam websites shortly after each event." I used data from both men's and women's singles matches from all four Grand Slams. In 2018, the Australian Open and French Open shifted their partnership from IBM to Infosys. Infosys tracks points differently and has fewer features. This complicates the integration of the data, so I chose to only include matches from the Australian Open and French Open from 2011-2017. In total, this includes 7917 matches and over 1.4 million points.

IBM tracks several useful calculations for each point. Some of the important variables include elapsed time, serve speed, rally length, distance run, serve depth, and return depth. Until 2016, many courts were not equipped to measure many of these important variables. For this reason, I reduced the scope fo the data set to tournaments from 2016-2023. This reduces the data set to 4070 matches and 750,000 points. Even now, not all courts have been equipped to calculate all of these complex measurements. In general, the most recent tournaments and later rounds have much more complete measuring systems.

Transformations

The original data is on the point-level, but, in order to evaluate the tendencies of players, I needed to convert the data into the player-level. In essence, I duplicated each point and shifted them into the perspective of both players. This left me with 1.5 million points. Then, I grouped by each player and created derived variables that summarized the players' relevant statistics. Overall, this produced 698 players and 67 variables.

Many of these derived variables are simple means of point-level variables. For instance, **distance_run** and **rally_count** reflect the players' average distance run and rally count per point. Many of the other derived variables are players' rates and frequencies over the data set. For example, **net_approach_rate** and **backhand_winner_rate** are proportions from 0 to 1.

Unfortunately, this summary method loses a lot of the point-level information. Means and proportions are helpful, but they cannot speak to the distribution or circumstances of the events.

Filtering

Some of these variables describe the occurrence of rare events. For example, **convert_break_pt_rate** describes players' conversion rate of break points. Players often only have a few break points per match. A fluke match could lead to a significant outlier. I hoped to reduce the effect of extreme values from these summaries by removing all players with fewer than four matches. I also removed players that did not play any matches on courts equipped with a serve location measuring system. Overall, this reduced our data set to 460 players. Now, a few abnormal matches will not greatly affect our analysis.

Unfortunately, many of the players removed from the data are weaker players. The median ranking of the 698 players is 99, while the median ranking of the 460 players is 79.1. Strong players are likely to advance far in tournaments and reach the four match threshold. Weaker players may only make one or two Grand Slam tournaments in their career.

Descriptive Statistics of Variables

Below is a table that summarizes the variables included in the analysis.

Variable	Lower Quartile	Mean	Upper Quartile	Standard Deviation
player_ht	174.00	180.41	185.00	9.12
player_hand	1.00	0.89	1.00	0.32
player_age	23.67	26.71	29.36	4.19

Variable	Lower Quartile	Mean	Upper Quartile	Standard Deviation
player_sex	0.00	0.51	1.00	0.50
seconds_per_point	40.34	47.62	47.04	15.44
points_per_match	142.84	184.70	225.30	45.89
ranking	49.53	83.96	105.22	58.62
serve_rate	0.49	0.50	0.51	0.01
first_serve_rate	0.59	0.62	0.65	0.05
point_win_rate	0.48	0.49	0.50	0.02
win_rate_serve	0.56	0.60	0.63	0.05
win_rate_return	0.35	0.38	0.42	0.05
game_win_rate	0.45	0.48	0.51	0.05
hold_rate	0.64	0.71	0.79	0.10
break_rate	0.18	0.24	0.31	0.09
set_win_rate	0.35	0.44	0.54	0.14
match_win_rate	0.29	0.42	0.55	0.19
ace_rate	0.03	0.06	0.08	0.04
ace_allow_rate	0.04	0.07	0.09	0.03
winner_rate	0.13	0.15	0.17	0.03
winner_allow_rate	0.14	0.16	0.18	0.03
forehand_winner_rate	0.07	0.08	0.09	0.02
backhand_winner_rate	0.03	0.04	0.04	0.01
forehand_winner_allow_rate	0.07	0.08	0.09	0.02
backhand_winner_allow_rate	0.03	0.04	0.05	0.01
double_fault_rate	0.03	0.05	0.05	0.02
unforced_error_rate	0.15	0.17	0.19	0.03
net_approach_rate	0.07	0.09	0.11	0.03
net_win_rate	0.64	0.67	0.71	0.05
break_chances_per_game	0.47	0.59	0.71	0.16

Variable	Lower Quartile	Mean	Upper Quartile	Standard Deviation
convert_break_pt_rate	0.36	0.40	0.45	0.07
save_break_pt_rate	0.53	0.57	0.62	0.07
distance_run	11.97	14.72	17.14	3.60
rally_count	2.89	3.33	3.70	0.58
first_serve_speed	97.57	105.51	113.66	9.76
second_serve_speed	82.11	87.53	92.87	7.80
first_serve_wide_rate	0.22	0.27	0.32	0.07
first_serve_body_wide_rate	0.15	0.18	0.21	0.05
first_serve_body_rate	0.04	0.08	0.10	0.05
first_serve_body_center_rate	0.12	0.15	0.19	0.05
first_serve_center_rate	0.26	0.32	0.37	0.08
first_serve_close_to_line_rate	0.34	0.38	0.41	0.05
second_serve_wide_rate	0.36	0.45	0.54	0.12
second_serve_body_wide_rate	0.23	0.32	0.37	0.13
second_serve_body_rate	0.06	0.14	0.17	0.14
second_serve_body_center_rate	0.18	0.27	0.33	0.14
second_serve_center_rate	0.44	0.53	0.61	0.15
second_serve_close_to_line_rate	0.56	0.64	0.71	0.12
deep_return_rate	0.37	0.40	0.42	0.05

Methodology

The player level data set contains high dimensional data with relatively few observations. I chose to use Principle Component Analysis to reduce the large number of variables into a small number of principle components.

Principle Component Analysis finds low-dimensional approximations for high-dimensional data. It is highly likely that this player-level data can be explained by a low-dimensional representation. Players have a few inherent characteristics and strategies that manifest in many visible outcomes. One visible outcome cannot alone measure the players' inward traits or strategies, but Principle Component Analysis can reduce the high dimensional data into understandable measurements of these traits and tendencies.

If the principle components are able to explain a large portion of the variance in interpretable manner, then this is good evidence that my assumption of a low-dimensional representation in the data is correct.

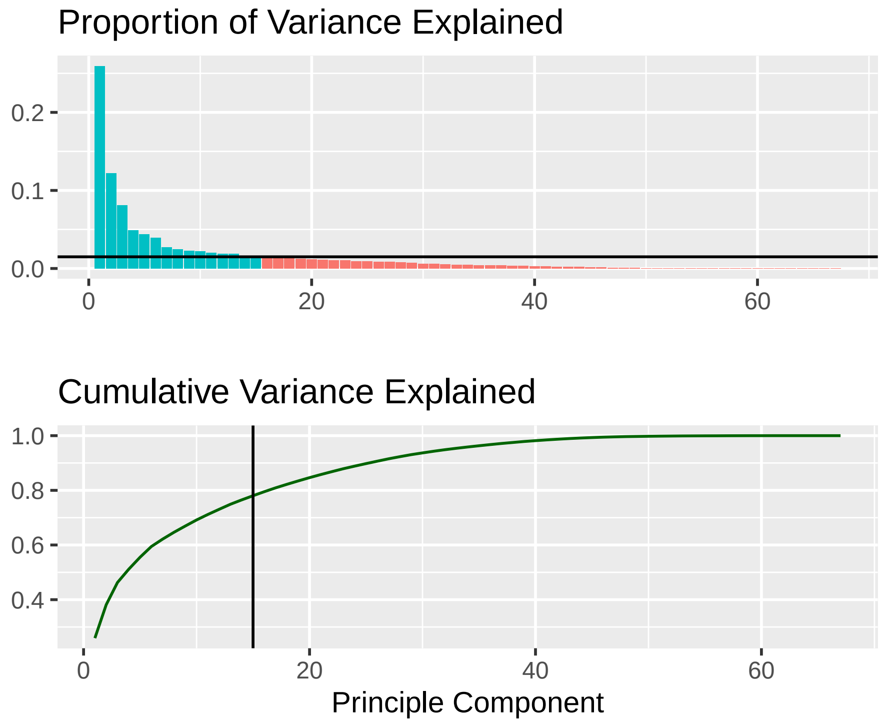
I chose to normalize each variable, because the variables contain a large discrepancy in their range of values. Without scaling the variables, variables like ranking (mean of 84) would dominate all of the proportions. I consider scaling the men and women separately and joining them together, but I am interested in the principle components' ability to recognize the inherent differences in the game. Men and women do not play against each other, so, in the future, it would be interesting to scale the players within their pool of competitors and rejoin the pools.

Results and Interpretations

Variance Explained

After performing the Principle Component Analysis, we can assess the proportion of variance in the original data that each principle component explains. There were 67 variables used in the analysis. If all the original variables were completely orthogonal, then each principle component would explain 1.49% (1/67) of the variance. For this reason, a reasonable cutoff is to include each of the principle components that exceeds this explanatory power.

A total of 15 principle components exceed 1.49% of variance explained. In total, they explain 78.02% of the total variancle. The first principle component explains the most variance by far (25.94%). The top three explain a combined 46.28% of the variance.



Interpreting Principle Components

The usefulness of Principle Component Analysis typically relies on the clarity and value of the principle components' interpretations. Many of the principle components in this analysis have strong and clear interpretations.

Principle Component 1: Serve Strength and Sex

There are two major differences in the men's and women's games. First, the men play best-of-five set matches and the women play best-of-three set matches. Thus, men typically play more points in a match than women. Second, men have a much stronger advantage on their serve than women. Their serves are much faster, they hit more aces, and they win a higher proportion of points on their serve.

This first principle component serves two purposes. First, it generally divides the men and women. Second, it measures a players' relative serving dominance. A player with a large positive value has a dominant serve and a player with a large negative value has a weak serve.

The table of the variable loadings on the first principle component display the variables that contribute the most to this component. **First_serve_speed**, **win_rate_serve**, **hold_rate**, **ace_rate**, **opp_break_chances_per_game**, and **second_serve_speed** are all factors of a strong serve. **Points_per_match** reflect the distinction between the rules of the mens and womens game.

Variable	PC1
first_serve_speed	0.22
win_rate_serve	0.21
opp_first_serve_speed	0.21
points_per_match	0.20
hold_rate	0.20
opp_second_serve_speed	0.20
ace_rate	0.19
opp_break_chances_per_game	-0.19
second_serve_speed	0.19

Principle Component 2: Dominance

The second principle component is more straight forward. This variable reflects the players' general dominance. Each of the variables with a high loading are some measure of a players' ability to win points (**point_win_rate**, **game_win_rate**, **set_win_rate**, etc.). Players with a large positive PC2 dominate their competition. There is also a slight emphasis on players that are dominant returners (**win_rate_return**, **break_rate**, **break_chances_per_game**).

Variable	PC2
point_win_rate	0.29
game_win_rate	0.28
set_win_rate	0.28
match_win_rate	0.28
winner_allow_rate	-0.23
winner_rate	0.21

Variable	PC2
win_rate_return	0.20
break_rate	0.20
forehand_winner_rate	0.19
break_chances_per_game	0.19

Principle Component 3: Carelessness

The third principle component represents a players' propensity to make mistakes. High rates of unforced errors and double faults both contribute to a positive PC3. Consequently, players with a positive PC3 play in shorter rallies and their opponents make fewer unforced errors and hit fewer winners. A positive PC3 is also generally associated with losing points and games.

Variable	PC3
unforced_error_rate	0.28
first_serve_rate	-0.24
opp_unforced_error_rate	-0.24
opp_net_approach_rate	-0.24
double_fault_rate	0.23
forehand_winner_allow_rate	-0.22
rally_count	-0.22
second_serve_close_to_line_rate	-0.20
point_win_rate	-0.19
game_win_rate	-0.19

Principle Component 4: Point Length and Opponent Aggressiveness

The fourth principle component represents a players' propensity to play long points. Players who play long points with a lot of running will have high values of PC4. Interestingly, the opponents' aggressiveness also factors into PC4. If the players' opponent hits deep returns or aims wide or close to the line, then this will contribute towards a negative PC4. Together, these two factors are related, because an aggressive opponent will shorten points and a passive opponent will lengthen them.

Variable	PC4
opp_distance_run	0.33
distance_run	0.31
opp_second_serve_wide_rate	-0.27
opp_deep_return_rate	-0.25
opp_first_serve_close_to_line_rate	-0.23
opp_second_serve_body_rate	0.22
seconds_per_point	0.18

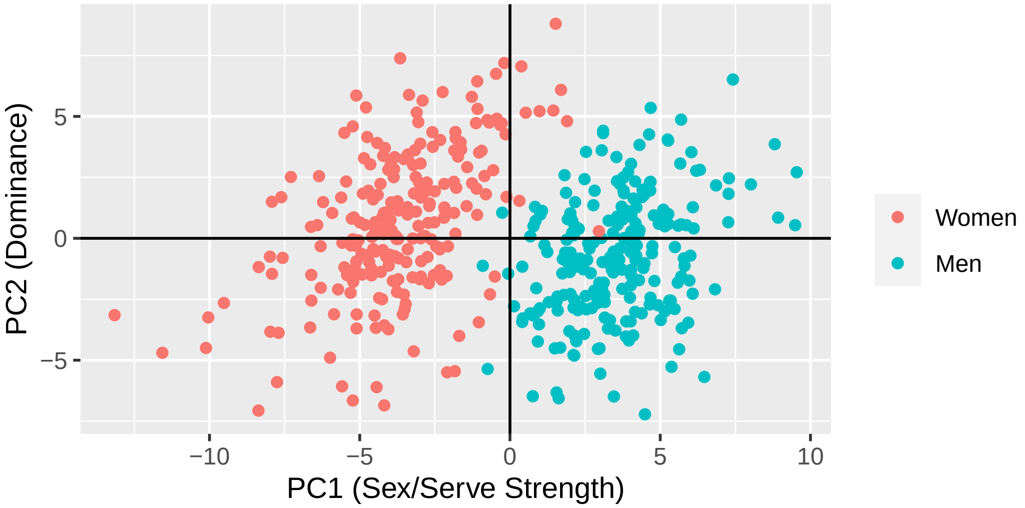
Additional Principle Components

- PC5 - Point Length and Serve Aggressiveness - a high distance run, low first serve percentage, and low second serve aggressiveness contribute to high PC5 values.
- PC6 - Serve Location - serving close to the lines on the first or second serve leads to positive PC6 values.
- PC7 - Opponent Serve Location - Opponent Serving close to the lines and double faulting leads to positive PC7 values.

Visualizing Data on Components

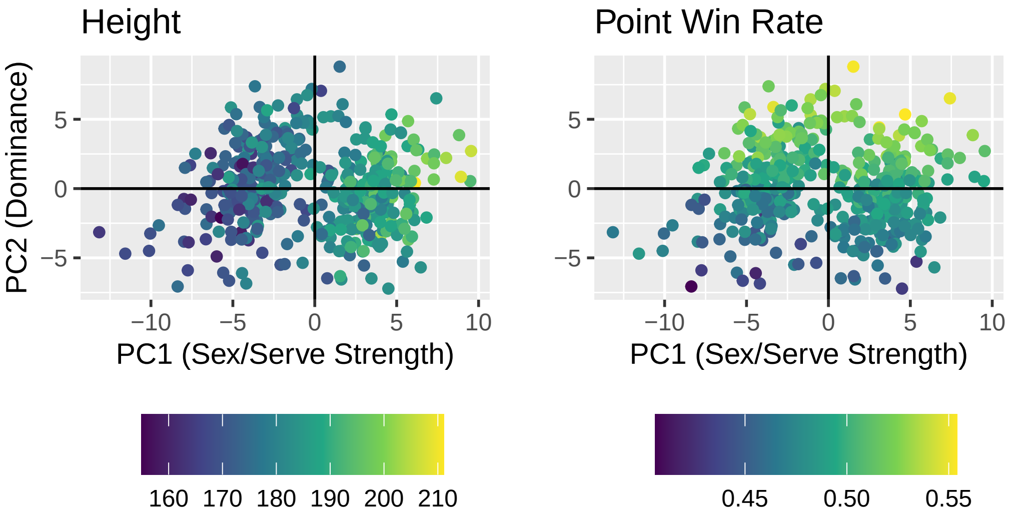
It can be helpful to visualize the observations on the principle components. PC1 provides a clear distinction between men and women. Men generally have positive PC1 values and women generally have negative PC1 values. Naomi Broady is the woman in the middle of the blue cloud. She is known for her big serve and overall mediocre level.

It is worth noting that within each sex, there is a general positive trend between PC1 and PC2. This makes sense, players with a dominant serve are generally dominant overall.



The scatterplot on the left visualizes another interesting find. The players height was not included in the analysis, but there is a clear association between the players' serve strength and height. Taller players generally have an advantage when serving, so this fits with expectations.

On the right, the association between PC2 and **point_win_rate** is displayed. In general, players who win more points have high PC2 values.



Case Studies

I assessed the seven principle components with 8 popular tennis players.

Overall, these players' results fit with our interpretations of the principle components.

- John Isner has a dominant serve but is only a moderately above average player. This fits with the first two principle components. He has a strong serve accuracy (PC6) and tends to try to shorten points (PC5).
- Serena Williams has an incredibly strong serve (PC1) and is one of the most dominant tennis players ever (PC2). She is aggressive (PC3) and is an accurate server (PC6).
- Rafael Nadal has a mediocre serve (PC1) but is fairly dominant (PC2). He is a very careful player that makes few errors (PC3).

Player	PC1	PC2	PC3	PC4	PC5	PC6	PC7
John Isner	9.54	2.71	-0.83	2.74	-5.61	2.14	0.09
Roger Federer	7.42	6.51	-2.62	-1.15	-2.33	0.99	1.02
Novak Djokovic	4.68	5.35	-4.94	-0.56	0.17	-0.42	-0.17
Rafael Nadal	3.10	4.41	-5.56	-1.72	-0.66	-2.23	-0.27
Serena Williams	1.52	8.80	0.28	-1.30	-1.44	1.99	0.68
Petra Kvitova	-0.46	6.75	2.57	-0.36	-2.25	0.85	0.27
Angelique Kerber	-5.45	2.33	-5.87	-0.83	1.07	0.52	0.55
Nuria Parrizas Diaz	-7.75	-5.90	-0.11	1.72	-1.60	-1.45	7.43

Overall, these results affirm my interpretations on the first three principle components but cast doubt on the fourth. I would have expected the signs to be flipped.

Conclusions and Future Work

This analysis provides substantial insight into the traits and tendencies of tennis players. The first seven principle components are understandable and have reasonable interpretations. They provide helpful avenues for distinguishing players.

This is a strong first exploratory step in the process of quantifying the tendencies of tennis players. I identified variables that quantify a players' serve strength, overall dominance, carelessness, point length, and serve location.

This work can be utilized for strategic gain by tennis coaches and players looking to analyze themselves or their future opponents. The results can also be used to predict future performance for players with certain characteristics.

The principle components provide measurement tools for other predictive or causal research. The analysis describes players with few dimensions and these dimensions can be reasonably tested in various circumstances.

Future Work

This initial analysis grossly reduces the explanation of the players' tendencies. I would suggest a future analysis that analyzes the players on a point level. This would help account for the covariance of many of the variables.

I would also consider standardizing the variables within the pool of mens and womens players separately. Men and women do not compete, so there is justifiable reason to standardize separately. In addition, in my analysis, much of the overall variance is found in the strategic differences between the mens and womens game. If I scale out these differences, then perhaps this will shed light on more nuanced differences in player tendencies.

These advanced methods will permit more nuanced research that can assess players' tendencies in different situations. Here are some ideas.

How do players' tendencies develop over time.

It would be interesting to evaluate players' tendencies and characteristics as they progress through their career. For several players, I have some data from 2011-2023. I could model standard aging patterns.

Do some players perform well against certain “types” of players?

High dimensional clustering methods can separate players into different player-types. It would be intriguing to assess the performance of players within certain types against each other. Perhaps certain types of players are positioned towards success against other types.

Do players change their approach as the encounter different opponents?

This report would involve an assessment of player tendencies on a match-level. I could research the overall variability of players’ tendencies over the course of different matches and assess if they appropriately vary their strategies. This would be an advanced study that would require a strong understanding of optimal tennis strategies.

Appendix

```r
library("tidyverse")
library("lubridate")
library("flextable")
library("corrr")
library("arrow")
library("here")
library("tidymodels")
library("patchwork")
knitr::opts_chunk$set(
  comment = "#", fig.height = 3,
  cache = FALSE,  collapse = TRUE,
  error = TRUE, echo = FALSE,
  message = FALSE
)
set_flextable_defaults(
  font.size = 10, theme_fun = theme_zebra,
  padding = 6,
  background.color = "#EFEFEF")
# read in data
gs <- read_parquet(here("data.parquet"))

# duplicate points from player 1's perspective
player1 <- gs %>%
  rename_with(~paste0("player", str_sub(.x, 3, -1)), starts_with("p1_")) %>%
  rename_with(~paste0("opp", str_sub(.x, 3, -1)), starts_with("p2_")) %>%
  rename("player" = "player1") %>%
  rename("opponent" = "player2") %>%
  rename("player_ranking" = "ranking1") %>%
  rename("opp_ranking" = "ranking2") %>%
  rename("player_ht" = "ht1") %>%
  rename("player_hand" = "hand1") %>%
  rename("player_age" = "age1") %>%
  rename("opp_ht" = "ht2") %>%
  rename("opp_hand" = "hand2") %>%
  rename("opp_age" = "age2") %>%
  rename("player_b365" = "b365_1") %>%
  rename("player_ps" = "ps_1") %>%
  rename("opp_b365" = "b365_2") %>%
  rename("opp_ps" = "ps_2") %>%
  mutate(
    status = case_when(
      status == "p1_break" ~ "player_break",
      status == "p2_break" ~ "opp_break",
      .default = "no_break")) %>%
  group_by(match_id) %>%
  mutate(
    time_since_last_pt = if_else(point_no == 1, 0, second(elapsed_time) - second(lag(elaps
  ungroup()

# duplicate points from player 2's perspective
player2 <- gs %>%
  rename_with(~paste0("player", str_sub(.x, 3, -1)), starts_with("p2_")) %>%
  rename_with(~paste0("opp", str_sub(.x, 3, -1)), starts_with("p1_")) %>%
  rename("player" = "player2") %>%
  rename("opponent" = "player1") %>%
  rename("player_ranking" = "ranking2") %>%
  rename("opp_ranking" = "ranking1") %>%
  rename("player_ht" = "ht2") %>%
  rename("player_hand" = "hand2") %>%
  rename("player_age" = "age2") %>%
  rename("opp_ht" = "ht1") %>%
  rename("opp_hand" = "hand1") %>%
  rename("opp_age" = "age1") %>%
  rename("player_b365" = "b365_2") %>%
  rename("player_ps" = "ps_2") %>%
  rename("opp_b365" = "b365_1") %>%
  rename("opp_ps" = "ps_1") %>%
  mutate(
    server = case_when(
      server == 1 ~ 2,
      server == 2 ~ 1,
      .default = NA),
    status = case_when(
      status == "p1_break" ~ "opp_break",
      status == "p2_break" ~ "player_break",
      .default = "no_break"),
    match_victor = case_when(
      match_victor == 1 ~ 2,
      match_victor == 2 ~ 1,
      .default = match_victor),
    set_victor = case_when(
      set_victor == 1 ~ 2,
      set_victor == 2 ~ 1,
      .default = set_victor),
    game_victor = case_when(
      game_victor == 1 ~ 2,
      game_victor == 2 ~ 1,
      .default = game_victor),
    point_victor = case_when(
      point_victor == 1 ~ 2,
      point_victor == 2 ~ 1,
      .default = point_victor),
    run = run * -1) %>%
  group_by(match_id) %>%
  mutate(
    time_since_last_pt = if_else(point_no == 1, 0, second(elapsed_time) - second(lag(elaps
  ungroup()

# combine the two players
players <- bind_rows(player1, player2) %>%
  filter(year != "2012", year != "2013", !str_detect(match_id, "2014-ausopen")) %>% # remo
  filter(year != "2011", year != "2014", !str_detect(match_id, "2015-ausopen")) %>% # remo
  filter(year != "2015", !str_detect(match_id, "2016-ausopen"))

# condense data into player-level
condensed <- players %>%
  mutate(
    player_first_serve_mph = if_else(server == 1&serve_no == 1, speed_mph, NA),
    player_second_serve_mph = if_else(server == 1&serve_no == 2, speed_mph, NA),
    opp_first_serve_mph = if_else(server == 2&serve_no == 1, speed_mph, NA),
    opp_second_serve_mph = if_else(server == 2&serve_no == 2, speed_mph, NA)
  ) %>%
  group_by(player) %>%
  summarise(
    player_ht = mean(player_ht, na.rm = TRUE),
    player_hand = player_hand[1],
    player_age = mean(player_age),
    player_sex = sex[1],
    matches = sum(match_pt),
    points = n(),
    seconds_per_point = mean(time_since_last_pt, na.rm = TRUE),
    service_games = sum(game_victor != 0 & server == 1 & tiebreak == 0),
    return_games = sum(game_victor != 0 & server == 2 & tiebreak == 0),
    points_per_match = points/matches,
    ranking = mean(player_ranking, na.rm = TRUE),
    opp_ranking = mean(opp_ranking, na.rm = TRUE),
    serve_rate = sum(server == 1)/points,
    first_serve_rate = sum(serve_no == 1&server == 1, na.rm = TRUE)/sum(server == 1, na.rm
    point_win_rate = sum(point_victor == 1)/points,
    win_rate_serve = sum(point_victor == 1&server == 1)/sum(server == 1),
    win_rate_return = sum(point_victor == 1&server == 2)/sum(server == 2),
    game_win_rate = sum(game_victor == 1)/sum(game_victor != 0),
    hold_rate = sum(game_victor == 1 & server == 1 & tiebreak == 0)/service_games,
    break_rate = sum(game_victor == 1 & server == 2 & tiebreak == 0)/return_games,
    set_win_rate = sum(set_victor == 1)/sum(set_victor != 0),
    match_win_rate = sum(match_victor == 1&match_pt == 1)/matches,
    ace_rate = sum(player_ace == 1)/sum(server == 1),
    ace_allow_rate = sum(opp_ace == 1)/sum(server == 2),
    winner_rate = sum(player_winner == 1)/sum(!is.na(player_winner), na.rm = TRUE),
    winner_allow_rate = sum(opp_winner == 1)/sum(!is.na(player_winner), na.rm = TRUE),
    forehand_winner_rate = sum(winner_shot_type == "F"&player_winner == 1, na.rm = TRUE)/s
    backhand_winner_rate = sum(winner_shot_type == "B"&player_winner == 1, na.rm = TRUE)/s
    forehand_winner_allow_rate = sum(winner_shot_type == "F"&opp_winner == 1, na.rm = TRUE
    backhand_winner_allow_rate = sum(winner_shot_type == "B"&opp_winner == 1, na.rm = TRUE
    double_fault_rate = sum(player_double_fault == 1&server == 1)/sum(server == 1),
    opp_double_fault_rate = sum(opp_double_fault == 1&server == 2)/sum(server == 2),
    unforced_error_rate = mean(player_unf_err, na.rm = TRUE),
    opp_unforced_error_rate = mean(opp_unf_err, na.rm = TRUE),
    net_approach_rate = mean(player_net_pt, na.rm = TRUE),
    opp_net_approach_rate = mean(opp_net_pt, na.rm = TRUE),
    net_win_rate = sum(player_net_pt_won == 1)/sum(player_net_pt),
    opp_net_win_rate = sum(opp_net_pt_won == 1)/sum(opp_net_pt),
    break_chances_per_game = sum(player_break_pt == 1)/return_games,
    opp_break_chances_per_game = sum(opp_break_pt == 1)/service_games,
    convert_break_pt_rate = sum(player_break_pt_won == 1)/sum(player_break_pt == 1),
    save_break_pt_rate = sum(opp_break_pt_missed == 1)/sum(opp_break_pt == 1),
    distance_run = mean(player_distance_run, na.rm = TRUE),
    opp_distance_run = mean(opp_distance_run, na.rm = TRUE),
    rally_count = mean(rally_count, na.rm = TRUE),
    first_serve_speed = mean(player_first_serve_mph, na.rm = TRUE),
    second_serve_speed = mean(player_second_serve_mph, na.rm = TRUE),
    opp_first_serve_speed = mean(opp_first_serve_mph, na.rm = TRUE),
    opp_second_serve_speed = mean(opp_second_serve_mph, na.rm = TRUE),
    first_serve_wide_rate = sum(serve_width == "W"&server == 1&serve_no==1, na.rm = TRUE)/
    first_serve_body_wide_rate = sum(serve_width == "BW"&server == 1&serve_no==1, na.rm =
    first_serve_body_rate = sum(serve_width == "B"&server == 1&serve_no==1, na.rm = TRUE)/
    first_serve_body_center_rate = sum(serve_width == "BC"&server == 1&serve_no==1, na.rm
    first_serve_center_rate = sum(serve_width == "C"&server == 1&serve_no==1, na.rm = TRUE
    first_serve_close_to_line_rate = sum(serve_depth == "CTL"&server == 1&serve_no == 1, n
    second_serve_wide_rate = sum(serve_width == "W"&server == 1&serve_no==1, na.rm = TRUE)
    second_serve_body_wide_rate = sum(serve_width == "BW"&server == 1&serve_no==1, na.rm =
    second_serve_body_rate = sum(serve_width == "B"&server == 1&serve_no==1, na.rm = TRUE)
    second_serve_body_center_rate = sum(serve_width == "BC"&server == 1&serve_no==1, na.rm
    second_serve_center_rate = sum(serve_width == "C"&server == 1&serve_no==1, na.rm = TRU
    second_serve_close_to_line_rate = sum(serve_depth == "CTL"&server == 1&serve_no == 1,
    opp_first_serve_wide_rate = sum(serve_width == "W"&server == 2&serve_no==1, na.rm = TR
    opp_first_serve_body_wide_rate = sum(serve_width == "BW"&server == 2&serve_no==1, na.r
    opp_first_serve_body_rate = sum(serve_width == "B"&server == 2&serve_no==1, na.rm = TR
    opp_first_serve_body_center_rate = sum(serve_width == "BC"&server == 2&serve_no==1, na
    opp_first_serve_center_rate = sum(serve_width == "C"&server == 2&serve_no==1, na.rm =
    opp_first_serve_close_to_line_rate = sum(serve_depth == "CTL"&server == 2&serve_no ==
    opp_second_serve_wide_rate = sum(serve_width == "W"&server == 2&serve_no==2, na.rm = T
    opp_second_serve_body_wide_rate = sum(serve_width == "BW"&server == 2&serve_no==2, na.
    opp_second_serve_body_rate = sum(serve_width == "B"&server == 2&serve_no==2, na.rm = T
    opp_second_serve_body_center_rate = sum(serve_width == "BC"&server == 2&serve_no==2, n
    opp_second_serve_center_rate = sum(serve_width == "C"&server == 2&serve_no==2, na.rm =
    opp_second_serve_close_to_line_rate = sum(serve_depth == "CTL"&server == 2&serve_no ==
    deep_return_rate = sum(return_depth == "D"&server==2, na.rm = TRUE)/sum(!is.na(return_
    opp_deep_return_rate = sum(return_depth == "D"&server==1, na.rm = TRUE)/sum(!is.na(ret
    tiebreak_win_rate = sum(tiebreak == 1&set_victor==1)/sum(tiebreak==1&set_victor!=0),
    tiebreak_pt_win_rate = sum(tiebreak == 1&point_victor==1)/sum(tiebreak==1)) %>%
  mutate(across(where(is.numeric), ~round(.x, digits = 3)))

# filter out players with inadequate data
reduced <- condensed %>% filter(matches > 3, !is.na(first_serve_wide_rate), !is.na(first_s
  mutate(
    player_ht = case_when(
      player == "Anna-Lena Friedsam" ~ 174,
      player == "Ben Shelton" ~ 193,
      player == "Catherine Bellis" ~ 168,
      player == "Caty McNally" ~ 180,
      player == "Cristina Bucsa" ~ 176,
      player == "Dalma Galfi" ~ 178,
      player == "Elena Gabriela Ruse" ~ 173,
      player == "Elisabetta Cocciaretto" ~ 166,
      player == "Elizaveta Kulichkova" ~ 176,
      player == "Greet Minnen" ~ 175,
      player == "Harmony Tan" ~ 170,
      player == "Jodie Burrage" ~ 175,
      player == "Kamilla Rakhimova" ~ 170,
      player == "Katie Volynets" ~ 170,
      player == "Linda Fruhvirtova" ~ 172,
      player == "Linda Noskova" ~ 179,
      player == "Lucia Bronzetti" ~ 170,
      player == "Mirra Andreeva" ~ 171,
      player == "Oceane Dodin" ~ 183,
      player == "Patricia Maria Tig" ~ 180,
      player == "Peyton Stearns" ~ 173,
      player == "Qinwen Zheng" ~ 178,
      player == "Rinky Hijikata" ~ 178,
      player == "Varvara Gracheva" ~ 178,
      player == "Viktoria Hruncakova" ~ 180,
      player == "Xiyu Wang" ~ 182,
      .default = player_ht),
    player_hand = case_when(
      player == "Cristina Bucsa" ~ "R",
      player == "Dalma Galfi" ~ "R",
      player == "Elena Gabriela Ruse" ~ "R",
      player == "Mirra Andreeva" ~ "R",
      player == "Nadia Podoroska" ~ "R",
      player == "Rinky Hijikata" ~ "R",
      player == "Yue Yuan" ~ "R",
      .default = player_hand),
    player_hand = case_when(
      player_hand == "R" ~ 1,
      player_hand == "L" ~ 0)) %>%
  select(-tiebreak_win_rate, -tiebreak_pt_win_rate)

# create means of variables
mean <- reduced %>% summarise(
  across(where(is.numeric), ~mean(.x, na.rm = TRUE))) %>%
  mutate(analysis = "mean")

# create lower quantile of variables
lower_quantile <- reduced %>% summarise(
  across(where(is.numeric), ~quantile(.x, .25, na.rm = TRUE))) %>%
  mutate(analysis = "lower_quantile")

# create upper quantile of variables
upper_quantile <- reduced %>% summarise(
  across(where(is.numeric), ~quantile(.x, .75, na.rm = TRUE))) %>%
  mutate(analysis = "upper_quantile")

# create standard deviation of variables
sd <- reduced %>% summarise(
  across(where(is.numeric), ~sd(.x, na.rm = TRUE))) %>%
  mutate(analysis = "sd")

# set seed
set.seed(1234)

# create pca and standardize variables - remove characterstic variables
reduced_pca <- reduced %>% select(-matches, -points, -service_games, -return_games,
                                  -player, -player_ht, -player_hand, -player_age, -player_
  prcomp(scale = TRUE)

# creating recipe of pca
mp_rec <- recipe(~., data = reduced) %>%
  step_normalize(all_numeric()) %>%
  step_pca(all_numeric(), id = "pca") %>%
  prep()

# creating loadings matrix
loadings_matrix <- tidy(reduced_pca, matrix = "loadings") %>%
  filter(PC %in% c(1:17)) %>%
  pivot_wider(names_from = PC, values_from = value) %>%
  rename_with(
    ~ paste0("PC", .x), .cols = everything()) %>%
  select(-PCcolumn) %>%
  as.matrix()

# creating data matrix
data_matrix <- reduced %>%
  select(-matches, -points, -service_games, -return_games, -player) %>%
  mutate(across(where(is.numeric), ~scale(.x))) %>% as.matrix()
# means, sd, and na
bind_rows(lower_quantile, mean, upper_quantile, sd) %>%
  select(-matches, -points, -service_games, -return_games) %>%
  relocate(analysis, .before = "player_ht") %>%
  mutate(across(where(is.numeric), ~round(.x, digits = 2))) %>% t() %>% as.data.frame() %>
  rownames_to_column() %>% as_tibble() %>%
  rename(
    "lower_quantile" = "V1",
    "mean" = "V2",
    "upper_quantile" = "V3",
    "sd" = "V4",
    "variable" = "rowname") %>% slice(-1) %>%
  transmute(
    Variable = variable,
    "Lower Quartile" = as.double(lower_quantile),
    Mean = as.double(mean),
    "Upper Quartile" = as.double(upper_quantile),
    "Standard Deviation" = as.double(sd)) %>%
  filter(!str_detect(Variable, "opp_")) %>%
  flextable() %>%
  width(j = 1, width = 2.1) %>%
  width(j = 3, width = .7) %>%
  width(j = c(2,4), width = 1.2) %>%
  width(j = c(5), width = 1.4) %>%
  align(align = "center", part = "all")
p1 <- tidy(reduced_pca, matrix = "eigenvalues") %>%
```

```r
  mutate(
    random = 1/n(),
    include = percent > random) %>%
  ggplot(aes(PC, percent)) +
  geom_col(aes(fill = include)) +
  labs(x = "", y = "", title = "Proportion of Variance Explained") +
  geom_hline(yintercept = 1/67, color = "black") +
  theme(legend.position = "none")

p2 <- tidy(reduced_pca, matrix = "eigenvalues") %>%
  ggplot(aes(PC, cumulative)) +
  geom_line(color = "darkgreen") +
  labs(x = "Principle Component", y = "", title = "Cumulative Variance Explained") +
  geom_vline(xintercept = 15, color = "black") +
  theme(legend.position = "none")

p1 / p2
tidy(reduced_pca, matrix = "loadings") %>%
  filter(PC %in% c(1)) %>%
  arrange(PC) %>%
  pivot_wider(names_from = PC, values_from = value) %>%
  rename("Variable" = "column",
         "PC1" = "1") %>%
  mutate(across(where(is.numeric), ~round(-.x, digits = 2))) %>%
  arrange(desc(abs(PC1))) %>%
  slice(1:9) %>%
  flextable() %>%
  align(align = "center", part = "all") %>%
  width(j = 1, width = 2.4)
tidy(reduced_pca, matrix = "loadings") %>%
  filter(PC %in% c(2)) %>%
  arrange(PC) %>%
  pivot_wider(names_from = PC, values_from = value) %>%
  rename("Variable" = "column",
         "PC2" = "2") %>%
  mutate(across(where(is.numeric), ~round(.x, digits = 2))) %>%
  arrange(desc(abs(PC2))) %>%
  slice(1:10) %>%
  flextable() %>%
  align(align = "center", part = "all") %>%
  width(j = 1, width = 2.4)
```

```
tidy(reduced_pca, matrix = "loadings") %>%
  filter(PC %in% c(3)) %>%
  arrange(PC) %>%
  pivot_wider(names_from = PC, values_from = value) %>%
  rename("Variable" = "column",
         "PC3" = "3") %>%
  mutate(across(where(is.numeric), ~round(-.x, digits = 2))) %>%
  arrange(desc(abs(PC3))) %>%
  slice(1:10) %>%
  flextable() %>%
  align(align = "center", part = "all") %>%
  width(j = 1, width = 2.4)
tidy(reduced_pca, matrix = "loadings") %>%
  filter(PC %in% c(4)) %>%
  arrange(PC) %>%
  pivot_wider(names_from = PC, values_from = value) %>%
  rename("Variable" = "column",
         "PC4" = "4") %>%
  mutate(across(where(is.numeric), ~round(-.x, digits = 2))) %>%
  arrange(desc(abs(PC4))) %>%
  slice(1:7) %>%
  flextable() %>%
  align(align = "center", part = "all") %>%
  width(j = 1, width = 2.4)
augment(reduced_pca) %>%
  mutate(sex = reduced$player_sex) %>%
  ggplot() +
  geom_jitter(aes(x = -.fittedPC1, y = .fittedPC2, color = factor(sex))) +
  labs(x = "PC1 (Sex/Serve Strength)", y = "PC2 (Dominance)", title = "") +
  geom_vline(xintercept = 0) +
  geom_hline(yintercept = 0) +
  scale_colour_discrete(name = "",
                        labels = c("Women", "Men"))
# augment(reduced_pca) %>%
#   mutate(sex = reduced$player_sex,
#          point_win_rate = reduced$point_win_rate,
#          player = reduced$player) %>%
#   relocate(player, .after = .rownames) %>%
#   relocate(sex, .after = player) %>%
#   filter(sex == 0) %>% arrange(.fittedPC1)
pp1 <- augment(reduced_pca) %>%
```

```r
  mutate(sex = reduced$player_sex,
         height = reduced$player_ht) %>%
  ggplot() +
  geom_jitter(aes(x = -.fittedPC1, y = .fittedPC2, color = height)) +
  labs(x = "PC1 (Sex/Serve Strength)", y = "PC2 (Dominance)", title = "Height") +
  geom_vline(xintercept = 0) +
  geom_hline(yintercept = 0) +
  theme(legend.position = "bottom",
        legend.key.size = unit(.8, "cm"),
        legend.title = element_blank()) +
  scale_color_continuous(name = "Height")

pp2 <- augment(reduced_pca) %>%
  mutate(sex = reduced$player_sex,
         point_win_rate = reduced$point_win_rate) %>%
  ggplot() +
  geom_jitter(aes(x = -.fittedPC1, y = .fittedPC2, color = point_win_rate)) +
  labs(x = "PC1 (Sex/Serve Strength)", y = "", title = "Point Win Rate") +
  geom_vline(xintercept = 0) +
  geom_hline(yintercept = 0) +
  theme(legend.position = "bottom",
        legend.key.size = unit(.8, "cm"),
        legend.title = element_blank()) +
  scale_color_continuous(name = "Point Win Rate")

pp1 + pp2
# creating data matrix
data_matrix <- reduced %>%
  select(-matches, -points, -service_games, -return_games, -player, -player_age, -player_h
  mutate(across(where(is.numeric), ~scale(.x))) %>% as.matrix()

# multiplying matrices and displaying case studies
(data_matrix %*% loadings_matrix) %>% as_tibble() %>%
  mutate(player = reduced$player) %>%
  relocate(player, .before = "PC1") %>%
  select(player:PC7) %>%
  filter(player %in% c("Nuria Parrizas Diaz", "John Isner", "Petra Kvitova", "Novak Djokov
  mutate(across(where(is.numeric), ~round(.x, digits = 2)),
         PC1 = -PC1,
         PC3 = -PC3,
         PC4 = -PC4,
         PC7 = -PC7) %>%
  rename(Player = player) %>%
  arrange(desc(PC1)) %>%
  flextable() %>%
  color(j = 2, color = "red", i = ~ PC1 < 0) %>%
  color(j = 3, color = "red", i = ~ PC2 < 0) %>%
  color(j = 4, color = "red", i = ~ PC3 < 0) %>%
  color(j = 5, color = "red", i = ~ PC4 < 0) %>%
  color(j = 6, color = "red", i = ~ PC5 < 0) %>%
  color(j = 7, color = "red", i = ~ PC6 < 0) %>%
  width(j = c(2:7), width = .7) %>%
  align(align = "center", part = "all")
```