MATH 254 - Winning in Tennis: Are There Hidden Factors?

Pandelis Margaronis Troy Pollock Tural Sadigov

October 17, 2022

This paper investigates the factors that contribute to winning in professional tennis. Using a data set with Association of Tennis Professional Matches from 2000, we conduct various tests. Firstly, we conduct a chi-square test with permutation to identify if the likelihood of an underdog win is independent of court surface. Then, we conduct a difference in means test to identify if the duration of a match is independent of court surface. Finally, we create a model to predict match duration in minutes with features such as court surface, winner rank, loser rank, winner ace, and loser ace. Some professionals may have higher success rates on specific court surfaces, but court surface does not, on average, impact the likelihood of an upset in tennis. Similarly, although tennis matches can range in length from under an hour to multiple hours, the duration of a tennis match is independent of court surface. The paper explores the creation of multiple models in an attempt to, as accurately as possible, predict the duration of a tennis match given certain variables.

Table of contents

1 Background and Significance

There are countless factors and statistics that influence the outcome of a sporting event; more specifically, factors such as court surface and ranking can play a role in a tennis match. Different players are said to have varying degrees of success when it comes to different court surfaces. For example, Rafael Nadal, a 22-time Grand Slam winner, has been most successful on clay throughout his career. On the other hand, his competitor and fellow multi-time Grand Slam Winner, Roger Federer, has found most of his success on grass. In another study, it was discovered that the #10 seed typically has about a 0.35 chance of beating the #2 seed in these tournaments (Ovaska and Sumell). This number increases when it comes to players who are not ranked as high, assuming the distance of eight spots remains constant. It also increases in Grand Slam tournaments. Moreover, Roger Federer's probability of winning a match on hard court is roughly 0.636; however, that number drops to 0.130 on clay ((Ovaska 2014)).

There is certainly an influence of court on player performance, as athletes have preferences. Many studies have backed up this claim with abundant evidence. Thus, it is interesting to consider how the court surface can affect the chance of an upset (a lower seed beating a higher seed). Additionally, one can look into whether court surface is related to the duration of a game, since tennis matches can take anywhere from under an hour to multiple hours. Finding connections between the aforementioned variables could play a big role in sports betting. From an economic standpoint, a longer tennis match brings in more money and attention.

The goal of this project is to study the role that court surface and ranking play in the outcome of different characteristics of a tennis match using Association of Tennis Professionals (ATP) match data. This paper sets out to discover any links between court surface and the likelihood of upsets, as well as duration of time.

2 Methods

a. *Data collection.*

This data was collected from tennis matches across various Association of Tennis Professionals (ATP) tournaments during the year 2000. Each observation (each row) in the data set represents information about a specific tennis match, and includes info about both the winner and the loser. Such information includes, but is not limited to, their respective rankings prior to the tournament, and also their performance stats from the match. More information regarding the match itself is also included, such as which round it was a part of (ex: semi-final round) and the duration of the match in minutes.

There are missing data points (N/A's), with some columns having a lot more missing data than others. Missing data is dropped where appropriate. The variables used in our tests and models do not have much missing data, and with missing data rows dropped, most rows remain to be used for testing.

For more information on the data, refer to the source: https://www.kaggle.com/datasets/gmadevs/atp-matches-dataset?resource=download

b. *Variable creation.*

Individual observations: Each row in our data set represents a tennis match, on a certain day and in an ATP tournament. Some examples of data included in each observation include, but are not limited to, winner rank, loser rank, winner ace, loser ace, minutes (duration), and surface that the match was played on.

Main Variables:

- Surface - A categorical variable with 4 levels (carpet, clay, grass, hard-court). It represents the surface that the match was played on.
- Minutes - A numerical variable that is continuous. Describes the length of the match in minutes.
- Winner Rank - A numerical variable that is discrete. Describes the rank of the winner prior to the tournament.
- Loser Rank - A numerical variable that is discrete. Describes the rank of the loser prior to the tournament.
- Winner Ace - A numerical variable that is discrete. Describes the number of aces that the winner had in the match.
- Loser Ace - A numerical variable that is discrete. Describes the number of aces that the loser had in the match.

For more information on other variables, refer to the cited data source. (Sackmann 2022)

c. *Analytic Methods.*

Likelihood of an Upset vs Surface:

- Chi-square testing is used to investigate if the likelihood of an upset is independent of the court surface. The testing will involve permutation testing to permute the court surfaces for every upset and identify if there are statistically significant differences among the different surfaces. This will be done by calculating the chi-squared statistics for each permutation conducted.

A linear model to predict duration of a match (in minutes) based off of court surface, and the competitors' ranks going into the match:

- A regression model was created in an attempt to predict the duration of a tennis match, in minutes, given the ranks of the two players participating in the match (as separate variables), the number of aces by the two players participating in the match (as separate variables), and the court surface that the match was played on.

Duration vs Surface:

- A hypothesis test, which tests the difference in means of duration of a tennis match (in minutes) on hard court surface and clay was conducted to observe if duration is independent of surface. These two surfaces were selected because they are the two surfaces in the data set that appear on the most observations.

3 Results

There are various results shown below from different tests that were conducted. The code utilizes different aspects from the tidyverse ((Wickham et al. 2019)), infer ((Couch et al. 2021)), and ggplot2 ((Wickham 2016)) packages.

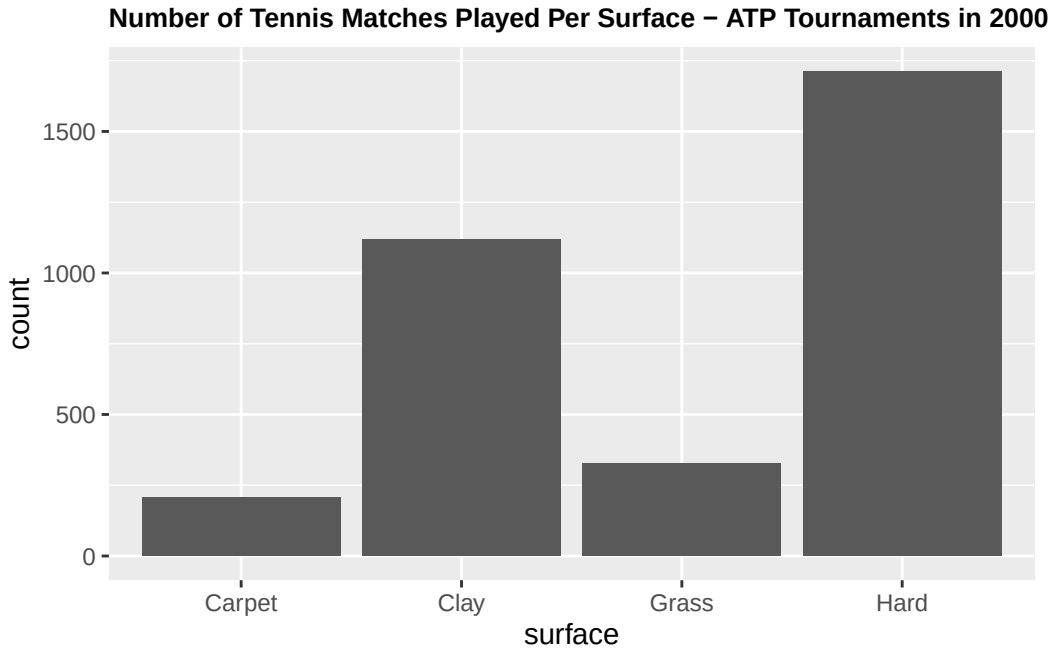
There are four different surfaces on which tennis matches were played in this data set. The surfaces consists of carpet, clay, grass, and hard-court.

Loading the data, and mutating a new column that informs us about upsets.

```
```{r}
library(tidyverse)
library(infer)
library(ggplot2)
new_data <- read_csv('https://raw.githubusercontent.com/turalsadigov/MATH_254/main/Datasets%
```
```

Displaying counts – grouping by surface

```{r}
new_data %>%
  group_by(surface) %>%
  ggplot(aes(x = surface)) +
  geom_bar() +
  ggtitle("Number of Tennis Matches Played Per Surface - ATP Tournaments in 2000") +
  theme(plot.title=element_text(size=10, face = 'bold'))
```



Mutate: adding a row to indicates if there was an upset (1) or not (0)

```{r}
new_data_2 <-
  new_data %>%
  mutate(upset = case_when(winner_rank > loser_rank ~ 1, winner_rank < loser_rank ~ 0))
new_data_2 %>%
  select(winner_rank, loser_rank, upset)
```

```
# A tibble: 3,364 x 3
   winner_rank loser_rank upset
         <dbl>      <dbl> <dbl>
 1         113         50     1
```

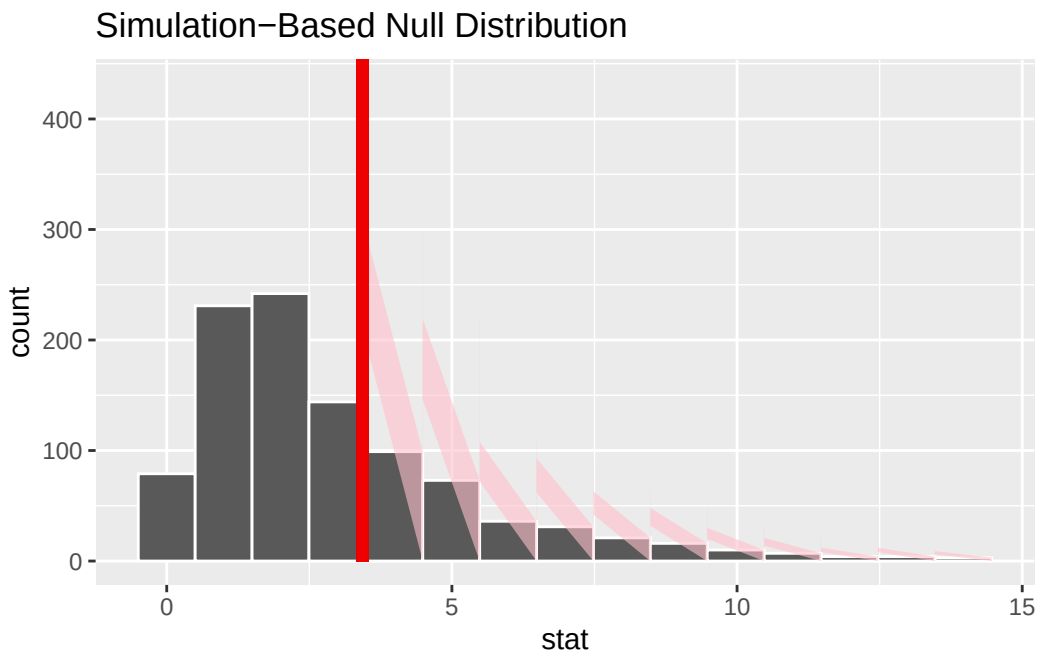
```
 2         352         139     1
 3         103         133     0
 4         107          95     1
 5          74         111     0
 6          92         102     0
 7         120         112     1
 8          79          91     0
 9          89          97     0
10         125         117     1
# ... with 3,354 more rows
```

The code below creates a data-frame to display details regarding the number of upsets per court surface and respective proportions. See Appendix for derivation of the results below.

```
```{r}
upsets_per_court_surface <- data.frame(
  surface = c('Carpet', 'Clay', 'Grass', 'Hard-Court'),
  Num_of_Matches = c(208, 1119, 326, 1711),
  Upsets_on_surface = c(65, 397, 122, 545),
  prop_of_upsets = c(0.3125, 0.3547811, 0.3742331, 0.3185272))
```
```

3.1 Investigating Upsets in Tennis Matches:

Is the likelihood of an upset, where a lower ranked player beats a higher ranked player, independent of court surface?

3.1.1 Checking Conditions for a Chi-Square Test for Independence:

The observations are independent of one another, and the observations were collected in no specific order; thus, independence and randomness are satisfied. Both variables being studied are categorical. The test checks for independence between the likelihood of an upset and court surface. The value of each cell's expected value is at least 5 in at least 80% of the cells, and no cell has an expected value of less than one. Since all conditions are met, we are able to conduct a chi-square test in which we test for independence between the following variables: upset and surface.

3.1.2 Hypotheses:

H_o: The likelihood of an upset is independent of the court surface in a tennis match.

H_a: The likelihood of an upset is dependent on the court surface in a tennis match.

α: 0.001

```
```{r}
set.seed(10)
# cleaning data for usage
new_data_2$upset<-factor(new_data_2$upset)
new_data_2$surface<-factor(new_data_2$surface)
# calculating the independent test statistic
observed_indep_stat <-
  new_data_2 %>%
  specify(surface ~ upset) %>%
  hypothesize(null = 'independence') %>%
  calculate(stat = 'Chisq')
# chi-square test with permutation:
null_dist <- new_data_2 %>%
  specify(surface ~ upset) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "Chisq")

null_dist %>%
  visualize() +
  shade_p_value(observed_indep_stat,
                direction = "greater")
null_dist %>%
  get_p_value(observed_indep_stat, direction = 'greater') # p-value: 0.311
```
```

```
# A tibble: 1 x 1
  p_value
    <dbl>
1   0.311
```



3.2 Investigating Duration of Tennis Matches:

Does court surface impact the duration of a tennis match?

We chose to explore this topic with a difference in means test between the mean duration of tennis matches on clay and the mean duration of tennis matches on hard-court. These two surfaces were chosen over carpet and grass because they had significantly more observations.

```{r}
surface_counts <- new_data %>%
  group_by(surface) %>%
  count() %>%
  arrange(rev(n))
```

Creating a data frame that selects surface and minutes and filters so that the two surfaces being studied are Clay and Hard Court.

```{r}
new_data_cp4 <-
  new_data %>%
  select(surface, minutes) %>%
  filter(surface == 'Clay' | surface == 'Hard')
```

3.2.1 Checking Conditions for a Difference in Means Test:

Both the clay and hard-court samples are independent. The sample sizes are n1 = 1711 and n2 = 1119 for hard court and clay respectively. Both of these values are greater than or equal to 30, so they are large enough to satisfy the condition that checks for (approximate) normality. Moreover, both samples are less than 10% of all matches played on the respective surface. As a result, the condition for sufficiently large samples checks out. Since all conditions are met, we are able to conduct a difference in means test to study whether court surface plays a role in the duration of a tennis match.

3.2.2 Hypothesis Test: Difference in Means

H_o: The mean duration (in minutes) of a tennis match on clay surface is equal to the mean duration (in minutes) of a tennis match on hard-court surface. That is, the difference in means is equal to 0. Thus, $\mu_1 - \mu_2 = 0$.

H_a: The mean duration (in minutes) of a tennis match on clay surface is NOT equal to the mean duration (in minutes) of a tennis match on hard-court surface. That is, the difference in means is not equal to 0. Thus, $\mu_1 - \mu_2 \neq 0$.

α: 0.001

3.2.3 Code to set up the difference in means test:

Calculating an observed difference in means between clay duration and hard-court duration.

```{r}
observed_stat <-
  new_data_cp4 %>%
  specify(minutes ~ surface) %>%
  calculate(stat = 'diff in means')
obs_stat_extracted = observed_stat$stat
```

Combining n1 + n2 into a general sample size, n.

```{r}
total_sample_size = 1711 + 1119
deg_freedom = total_sample_size - 2
```

P-value:

```{r}
2 * (1 - pt(obs_stat_extracted, deg_freedom))
```

```
[1] 0.3558173
```

3.3 Predicting Duration of Tennis Matches:

Assembling multiple linear regression models while utilizing a validation set approach.

3.3.1 Approach:

To conduct a validation set approach, we took our data set and conducted various splits. We split the data into two groups initially: training data and testing data. Then, the training data (or non-testing data) was split into a training component and a validation component (See Appendix).

4 Discussion/Conclusions

As a disclaimer, various interpretations of our results in this paper use ATP Tennis and professional tennis as interchangeable populations. Considering that ATP is a high-level professional organization, it is safe to expect similar results as those that were observed across non-ATP matches and tournaments. This is under the assumption that the sample falls under a general population of high-level professional tennis. It would not be realistic to generalize our results to amateur tennis, or other less-known intermediate organizations.

Predicting an upset is a difficult task, and every competition generally has both a favorite and an underdog. In tennis, the 'favorite' to win typically depends on rank and in many matches, court surface. Influential players like Roger Federer and Rafael Nadal have distinct preferences of court surface, and their record shows it. In important matches, for many pro tennis players, court surfaces could impact the outcome, and in some cases, may cause an upset. The objective of this study was to examine the relationship between court surface and upsets of tennis matches using data collected from tennis matches across various Association of Tennis Professionals (ATP).

In understanding that preferred court surface is an advantage for many pro tennis players, especially in uneven match ups, we ask: is there sufficient evidence to claim that an upset is independent of court surface? After conducting the chi-squared tests for each permutation of our data, we have confidence that an upset and court surface are indeed independent of each other. The p-value obtained from the chi-square test was approximately equal to 0.311. This value is much larger than the pre-declared significance level of 0.001. Thus, we fail to reject the

null hypothesis that the likelihood of an upset is independent of the court surface in a tennis match. We do not have conclusive evidence for the alternative hypothesis that the likelihood of an upset is dependent on the court surface in a tennis match.

We chose to explore where court surface impacts the duration of a tennis match with a difference in means test between the mean duration of tennis matches on clay and the mean duration of tennis matches on hard-court. The p-value that was generated from the difference in means test is approximately equal to 0.356. This p-value is larger than the significance level of 0.001. As a result, we fail to reject the null hypothesis that the mean duration (in minutes) of a tennis match on clay surface is equal to the mean duration (in minutes) of a tennis match on hard-court surface.

In an attempt to predict the duration of a tennis match created various linear regression models. Out of the models created, the model that performed the best was model 2. It had the greater R^2 value over the four models we created. Our chosen model used court surface, winner ace, loser ace, winner rank and loser rank to predict match duration. Of those predictors, those that proved to be statistically significant were the following: loser rank, winner ace, loser ace, and surface, for all levels except grass. Loser rank has a negative coefficient, meaning that, as the ranking of the loser in a tennis match gets worse, the duration of the match decreases. This makes sense; a worse player is easier to beat. Loser ace has a positive coefficient, meaning that match duration increases as the loser gets more aces. This makes sense, since an increase amount of aces for the loser gives them more points, as well as confidence. This would, on average, temporarily even the playing field by bringing their score closer to that of the winner and thus extending the length of the match. We were surprised that winner ace also had a positive coefficient, since one would expect that as the winner got more aces, the match would finish sooner. In terms of court surface, results varied greatly with different seed generators, something that is addressed below.

By changing the random seed used in our calculations, we get different results as to which model performs best. If we were to redo this process from scratch, we would use logistic regression. This way, we would split the data repeatedly with parallel programming, repeating the process where non-testing data is split into a training set and validation set. Bootstrap sampling would be utilized in this process. Overall, this new approach would ensure more stability in our results and less variability in which model is "ideal" for fitting the data.

Finally, we give credit to the knitr package ((Xie 2022)) for providing the resources needed to format and construct this paper, as well as USCLAP ((Ward and Tackett 2022)) for providing the project template.

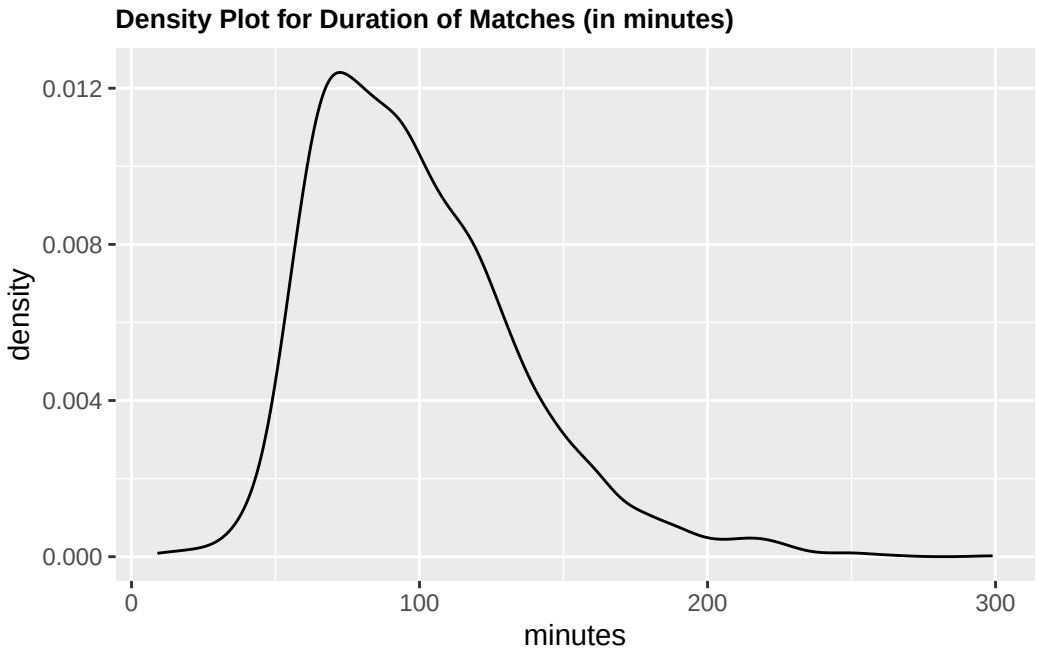
5 Appendix

The code below consists of various calculations that produced the needed results to create the "upsets_per_court_surface" data frame.

```{r}
total_upset_carpet <-
  new_data_2 %>%
  filter(upset == 1 & surface == "Carpet") %>%
  count()
prop_upset_carpet <- total_upset_carpet / 208

total_upset_clay <-
  new_data_2 %>%
  filter(upset == 1 & surface == "Clay") %>%
  count()
prop_upset_clay <- total_upset_clay / 1119


total_upset_grass <-
new_data_2 %>%
  filter(upset == 1 & surface == "Grass") %>%
  count()
prop_upset_grass <- total_upset_grass / 326


total_upset_hard <-
  new_data_2 %>%
  filter(upset == 1 & surface == "Hard") %>%
  count()
prop_upset_hard <- total_upset_hard / 1711
```

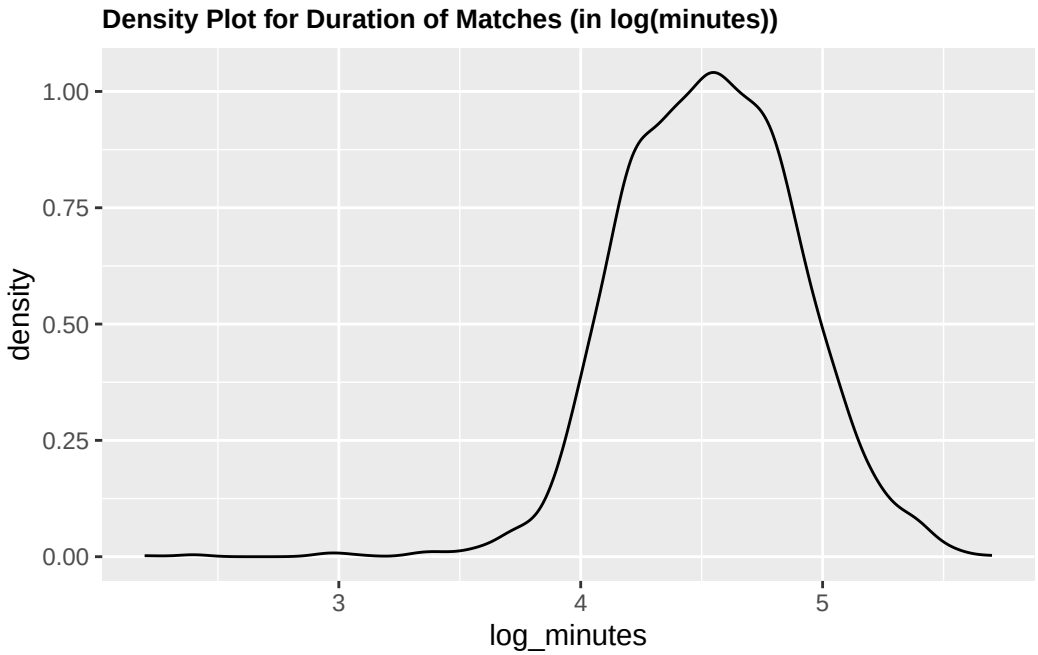
Density Plot for Duration of Matches (in minutes)

```{r}
new_data %>%
  select(minutes) %>%
  drop_na() %>%
  ggplot(aes(x = minutes)) +
  geom_density() +
  ggtitle("Density Plot for Duration of Matches (in minutes)") +
  theme(plot.title=element_text(size=10, face = 'bold'))
```



Density Plot for Duration of Matches (in log(minutes))

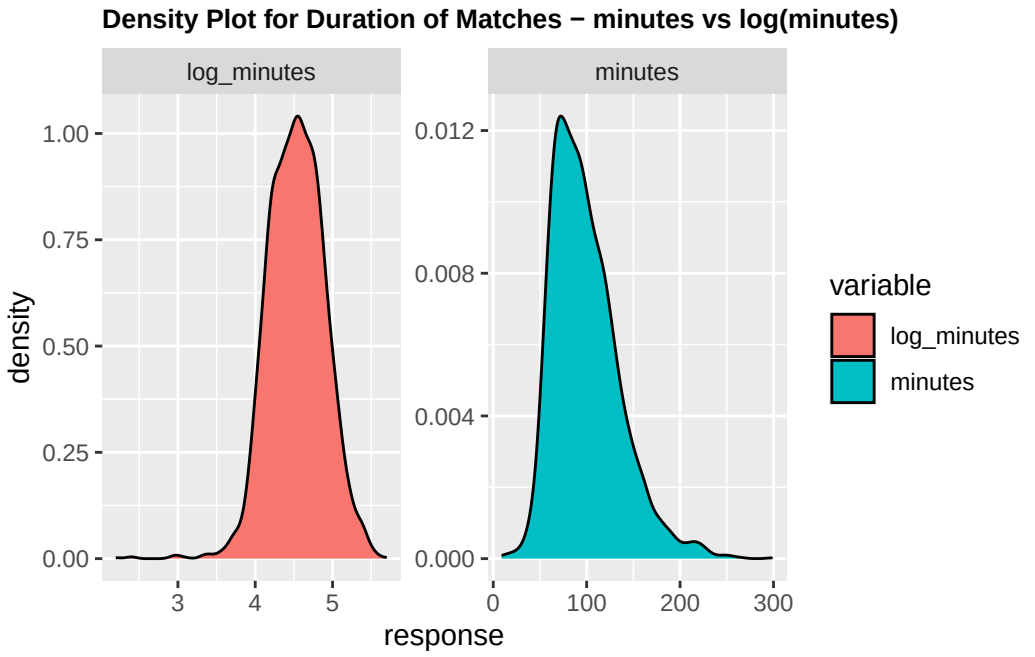
```{r}
new_data %>%
  select(minutes) %>%
  drop_na() %>%
  mutate(log_minutes = log(minutes)) %>%
  ggplot(aes(x = log_minutes)) +
  geom_density() +
  ggtitle("Density Plot for Duration of Matches (in log(minutes))") +
  theme(plot.title=element_text(size=10, face = 'bold'))
```



The following code creates a density plot for duration of matches. It shows a density plot for minutes vs log(minutes)). The code uses aspects of the reshape ((Wickham 2007)) and ggplot2 ((Wickham 2016)) packages.

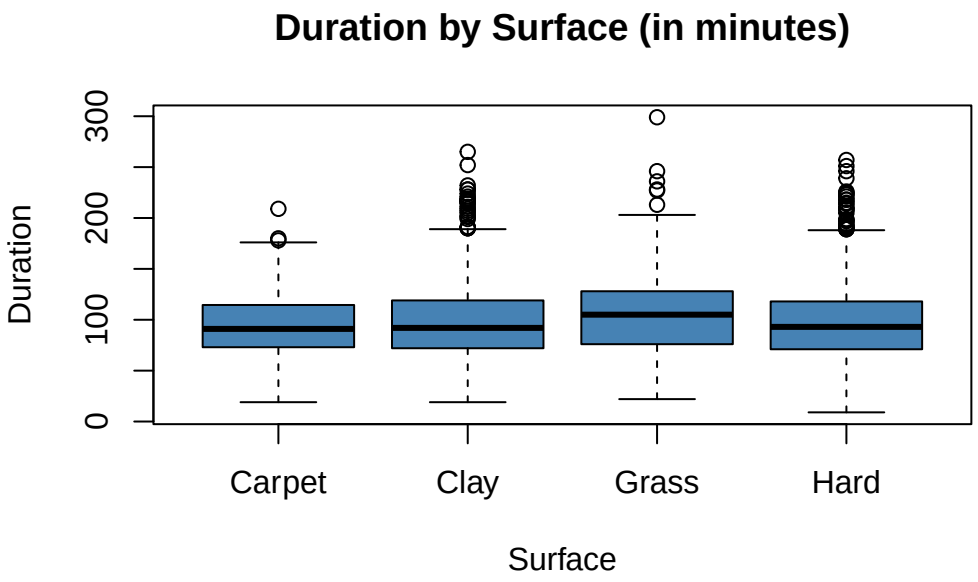
```{r}
library(reshape)
library(ggplot2)

new_data_minutes <-
  new_data %>%
  select(minutes) %>%
  drop_na() %>%
  mutate(log_minutes = log(minutes))
df_minutes <- data.frame(
  new_data_minutes %>%
  select(minutes, log_minutes))

df_minutes %>%
  mutate(row = row_number()) %>%
  pivot_longer(names_to = 'variable', values_to = 'response', cols = c(minutes, log_minutes)
  geom_density() +
  facet_wrap(~variable, scales = 'free') +
  ggtitle("Density Plot for Duration of Matches - minutes vs log(minutes)") +
```

```
  theme(plot.title=element_text(size=10, face = 'bold'))
```



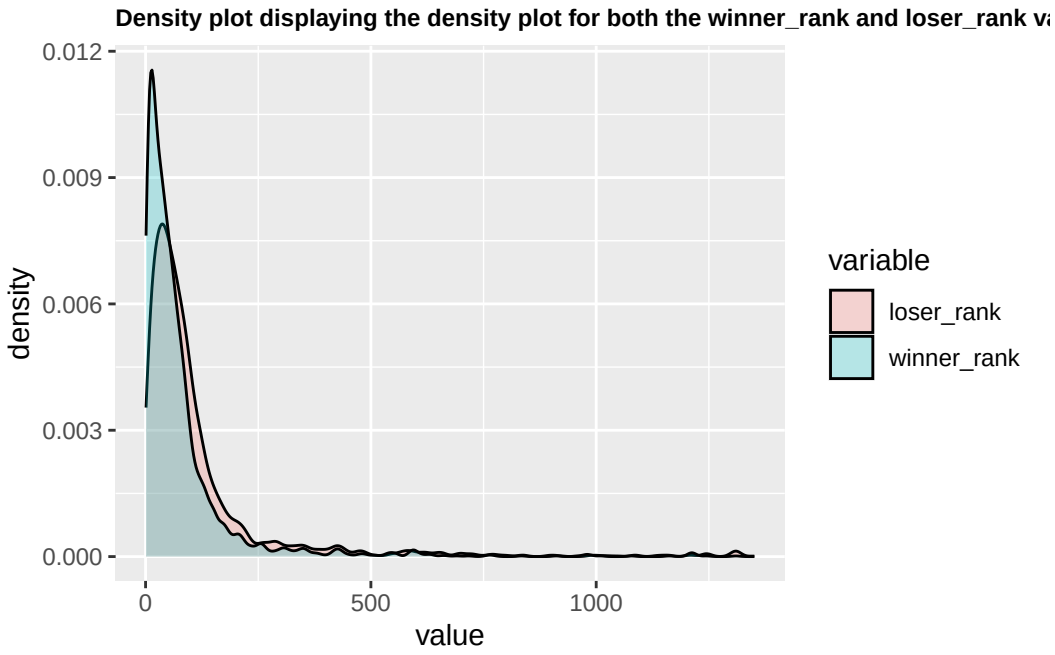
Side-by-side box plots displaying the distributions of minutes per game, for each surface.

```{r}
boxplot(new_data$minutes ~ new_data$surface, col='steelblue',
        main='Duration by Surface (in minutes)',
        xlab='Surface',
        ylab='Duration')
```



Density plot displaying the density plot for both the winner_rank and loser_rank variables.

```{r}
library(reshape)
library(ggplot2)

df_ranks <- data.frame(
  new_data_2 %>%
  select(loser_rank, winner_rank) %>%
    drop_na())
data_1 <- melt(df_ranks)
data_1 %>%
  ggplot(aes(x=value, fill=variable)) +
  geom_density(alpha=.25) +
  ggtitle("Density plot displaying the density plot for both the winner_rank and loser_rank v
  theme(plot.title=element_text(size=9, face = 'bold'))
```



5.1 Linear Regression Model: Predicting Minutes - The Duration of a Tennis Match

The code below utilizes aspects from the tidymodels ((Kuhn and Wickham 2020)) and vip ((Greenwell and Boehmke 2020)) libraries.

```{r}
library(tidymodels)
library(vip)
```

Mutating new variable: log of response variable (minutes)

```{r}
new_data11 <-
  new_data %>%
  mutate(log_minutes_plus_1 = log(minutes+1))
```

Relocating variables for better layout.

```{r}
df <-
new_data11 %>%
```

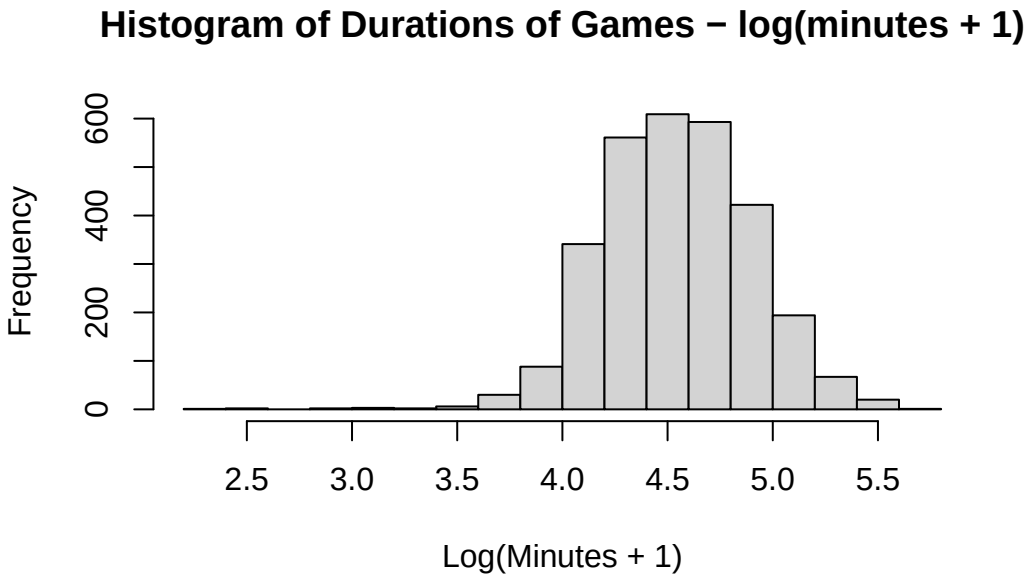
```
  relocate(minutes, .before = tourney_id) %>%
   relocate(log_minutes_plus_1, .after = minutes) %>%
   relocate(winner_rank, .after = log_minutes_plus_1) %>%
   relocate(loser_rank, .after = winner_rank) %>%
  relocate(surface, .after = loser_rank)
```

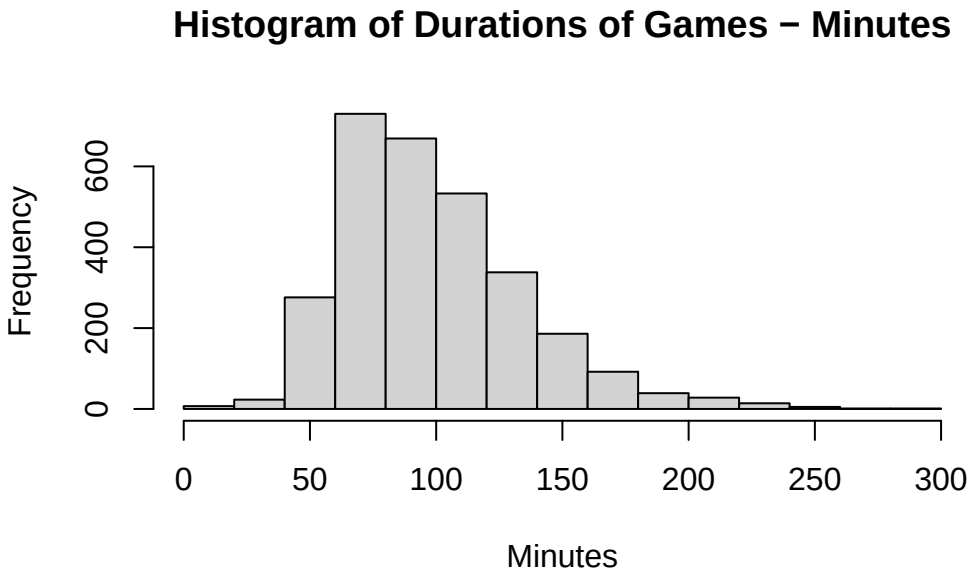
The code below splits the data into two groups: Testing Data and Non-testing (Training and Validation) Data. The code draws inspiration from the Math254 github page. (Sadigov 2022)

```{r}
set.seed(10)
df_split <- initial_split(data = df, prop = .80, strata = minutes)
df_training <- training(df_split)
df_testing <- testing(df_split)
df_validation_resample <- validation_split(data = df_training,
                                           prop = .80,
                                           strata = minutes)
```

Histograms of response variables.

```{r}
hist(df$log_minutes_plus_1,
      main='Histogram of Durations of Games - log(minutes + 1)',
      xlab='Log(Minutes + 1)',
      ylab='Frequency')
hist(df$minutes,
      main='Histogram of Durations of Games - Minutes',
      xlab='Minutes',
      ylab='Frequency')
```





Model Specifications:

```{r}
```

```
lm_specs <-
  linear_reg() %>%
  set_engine('lm') %>%
  set_mode('regression')
```

The code below fits various models, with varying predictors, to the training data. The code draws inspiration from the Math254 github page. (Sadigov 2022)

```{r}
set.seed(10)
model_1 <-
  lm_specs %>%
  fit_resamples(log_minutes_plus_1 ~ winner_rank + loser_rank + w_ace + l_ace + surface,
                df_validation_resample,
                metrics = metric_set(rsq, rmse, mae))
model_1

model_2 <-
  lm_specs %>%
  fit_resamples(minutes ~ winner_rank + loser_rank + w_ace + l_ace + surface,
                df_validation_resample,
                metrics = metric_set(rsq, rmse, mae))
model_2

model_3 <-
  lm_specs %>%
  fit_resamples(log_minutes_plus_1 ~ surface + w_ace + l_ace,
                df_validation_resample,
                metrics = metric_set(rsq, rmse, mae))
model_3

model_4 <-
  lm_specs %>%
  fit_resamples(minutes ~ surface + w_ace + l_ace,
                df_validation_resample,
                metrics = metric_set(rsq, rmse, mae))
model_4
```


```
# Resampling results
# Validation Set Split (0.8/0.2)  using stratification
# A tibble: 1 x 4
```

```
  splits             id         .metrics         .notes
  <list>             <chr>      <list>           <list>
1 <split [2150/540]> validation <tibble [3 x 4]> <tibble [0 x 3]>
# Resampling results
# Validation Set Split (0.8/0.2)  using stratification
# A tibble: 1 x 4
  splits             id         .metrics         .notes
  <list>             <chr>      <list>           <list>
1 <split [2150/540]> validation <tibble [3 x 4]> <tibble [0 x 3]>
# Resampling results
# Validation Set Split (0.8/0.2)  using stratification
# A tibble: 1 x 4
  splits             id         .metrics         .notes
  <list>             <chr>      <list>           <list>
1 <split [2150/540]> validation <tibble [3 x 4]> <tibble [0 x 3]>
# Resampling results
# Validation Set Split (0.8/0.2)  using stratification
# A tibble: 1 x 4
  splits             id         .metrics         .notes
  <list>             <chr>      <list>           <list>
1 <split [2150/540]> validation <tibble [3 x 4]> <tibble [0 x 3]>
```

Putting models together.

````
```{r}
fitted_models <- list(model_1, model_2, model_3, model_4)
fitted_models
```
````

```
[[1]]
# Resampling results
# Validation Set Split (0.8/0.2)  using stratification
# A tibble: 1 x 4
  splits             id         .metrics         .notes
  <list>             <chr>      <list>           <list>
1 <split [2150/540]> validation <tibble [3 x 4]> <tibble [0 x 3]>

[[2]]
# Resampling results
# Validation Set Split (0.8/0.2)  using stratification
# A tibble: 1 x 4
  splits             id         .metrics         .notes
```

```
  <list>              <chr>      <list>           <list>
1 <split [2150/540]> validation <tibble [3 x 4]> <tibble [0 x 3]>

[[3]]
# Resampling results
# Validation Set Split (0.8/0.2)  using stratification
# A tibble: 1 x 4
  splits              id         .metrics         .notes
  <list>              <chr>      <list>           <list>
1 <split [2150/540]> validation <tibble [3 x 4]> <tibble [0 x 3]>

[[4]]
# Resampling results
# Validation Set Split (0.8/0.2)  using stratification
# A tibble: 1 x 4
  splits              id         .metrics         .notes
  <list>              <chr>      <list>           <list>
1 <split [2150/540]> validation <tibble [3 x 4]> <tibble [0 x 3]>
```

Pull out metrics for comparison.

```{r}
validation_results <-
  bind_rows(fitted_models) %>%
  mutate(model_name = row_number())
validation_results
```

```
# A tibble: 4 x 5
  splits              id         .metrics         .notes           model_name
  <list>              <chr>      <list>           <list>                <int>
1 <split [2150/540]> validation <tibble [3 x 4]> <tibble [0 x 3]>          1
2 <split [2150/540]> validation <tibble [3 x 4]> <tibble [0 x 3]>          2
3 <split [2150/540]> validation <tibble [3 x 4]> <tibble [0 x 3]>          3
4 <split [2150/540]> validation <tibble [3 x 4]> <tibble [0 x 3]>          4
```

More Wrangling.

```{r}
set.seed(10)
validation_results %>%
  select(.metrics, model_name) %>%
```

```
  unnest(.metrics) %>%
  select(.metric, .estimate, model_name) %>%
  pivot_wider(names_from = .metric, values_from = .estimate)
```

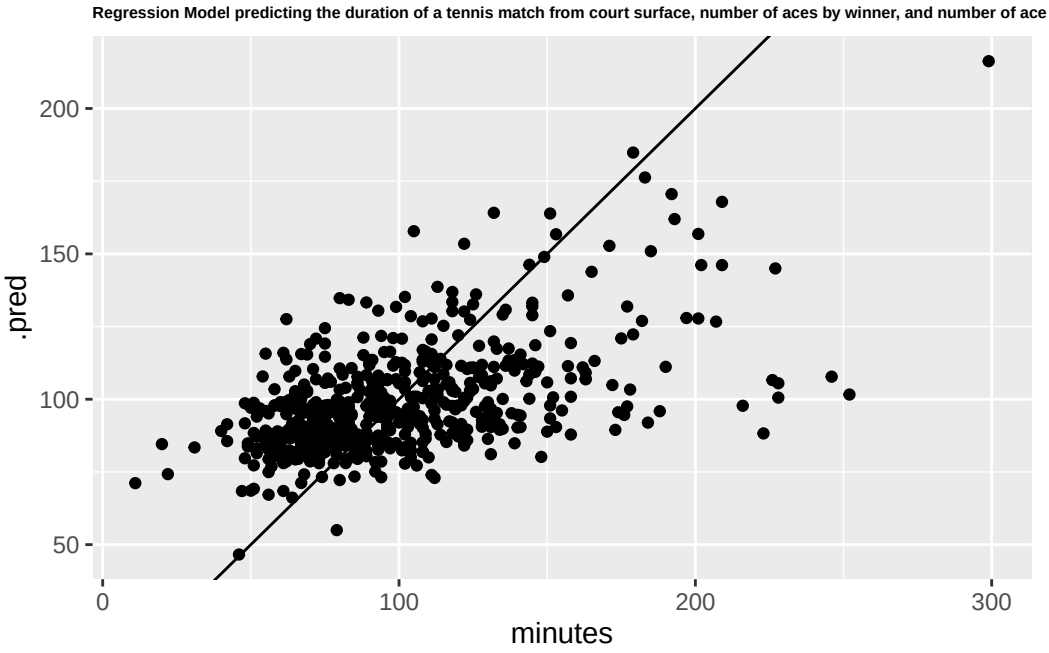
```
# A tibble: 4 x 4
  model_name   rsq   rmse    mae
       <int> <dbl>  <dbl>  <dbl>
1          1 0.236  0.314  0.250
2          2 0.269 30.7   24.5
3          3 0.221  0.316  0.251
4          4 0.251 30.8   24.5
```

We chose model 3. Final re-sample below.

```{r}
set.seed(10)
chosen_model_2 <-
  lm_specs %>%
  last_fit(minutes ~ surface + w_ace + l_ace + winner_rank + loser_rank,
           df_split,
           metrics = metric_set(rsq, rmse, mae))
chosen_model_2
collect_metrics(chosen_model_2)

collect_predictions(chosen_model_2) %>%
  rsq(truth = minutes, estimate = .pred)

collect_predictions(chosen_model_2) %>%
  ggplot(aes(x = minutes, y = .pred)) +
  geom_point() +
  geom_abline(slope = 1, intercept = 0) +
  ggtitle("Regression Model predicting the duration of a tennis match from court surface, nu
  theme(plot.title=element_text(size=6, face = 'bold'))
```

```
# Resampling results
# Manual resampling
# A tibble: 1 x 6
  splits             id               .metrics .notes   .predictions .workflow
  <list>             <chr>            <list>   <list>   <list>       <list>
1 <split [2690/674]> train/test split <tibble> <tibble> <tibble>     <workflow>
```

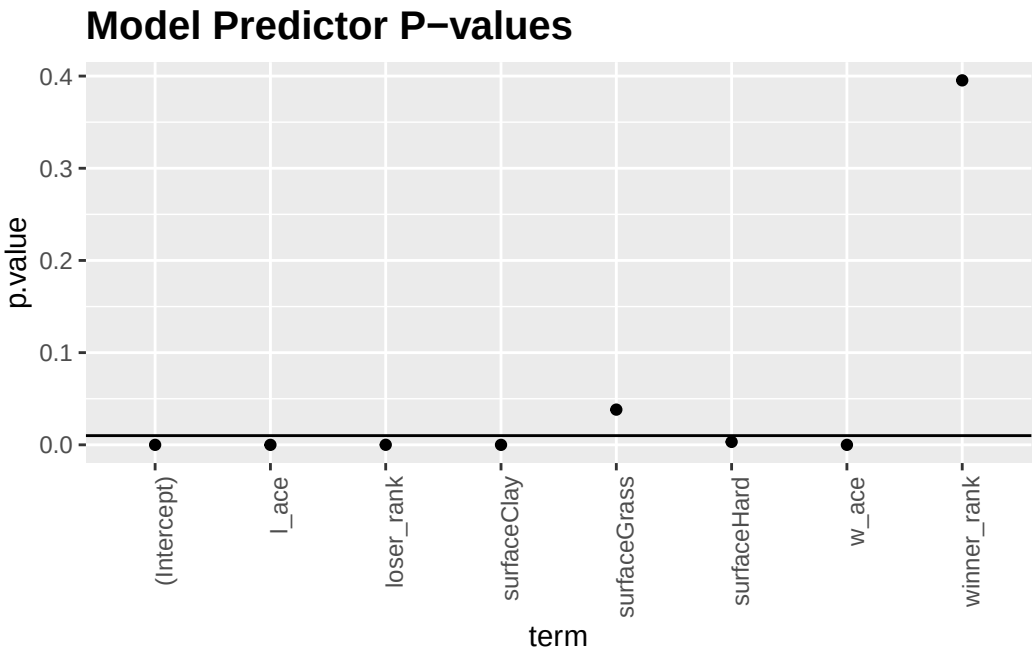
```
# A tibble: 3 x 4
  .metric .estimator .estimate .config
  <chr>   <chr>          <dbl> <chr>
1 rsq     standard       0.300 Preprocessor1_Model1
2 rmse    standard      32.4   Preprocessor1_Model1
3 mae     standard      24.7   Preprocessor1_Model1
# A tibble: 1 x 3
  .metric .estimator .estimate
  <chr>   <chr>          <dbl>
1 rsq     standard       0.300
```



Investigating the significance of a model. We used multiple testing, so we must adjust our a value.

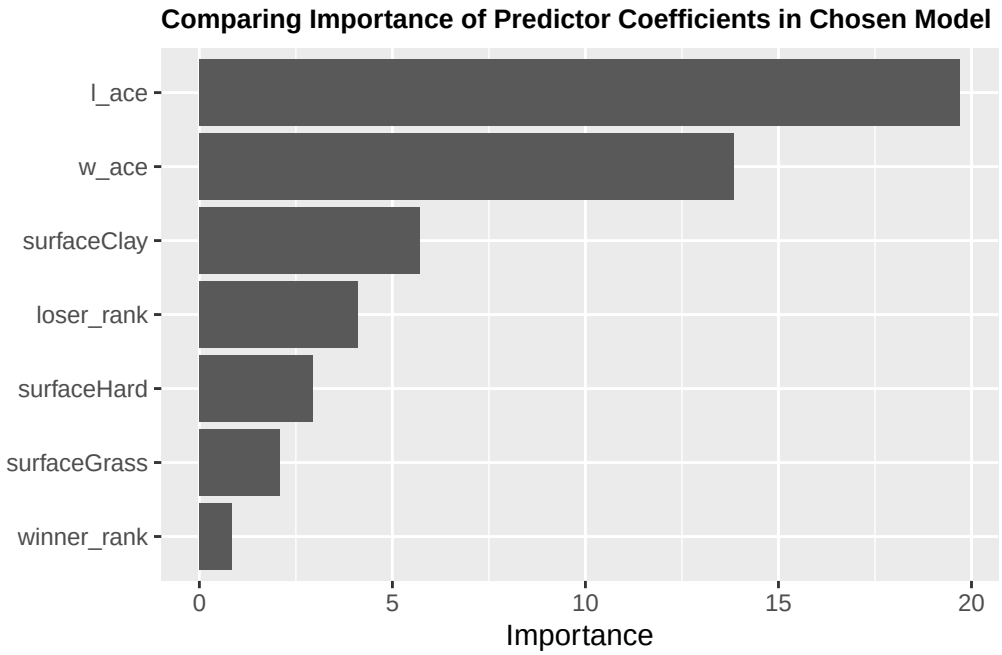
```{r}
a = 0.05/5
set.seed(10)
extract_workflow(chosen_model_2) %>%
  tidy() %>%
  mutate(sgnfc = if_else(p.value < a, 'Yes', 'No')) %>%
  arrange(p.value) %>%
  ggplot(aes(x = term, y = p.value)) +
```

```
  geom_point() +
  geom_hline(yintercept = a) +
  theme(axis.text.x = element_text(angle = 90, vjust = 1, hjust = 1)) +
  ggtitle("Model Predictor P-values") +
  theme(plot.title=element_text(size=15, face = 'bold'))
```



Bar graph displaying the importance of each variable as a predictor.

```{r}
model <- extract_fit_engine(chosen_model_2$.workflow[[1]])
vip(model) +
  ggtitle("Comparing Importance of Predictor Coefficients in Chosen Model") +
  theme(plot.title=element_text(size=10, face = 'bold'))
```



Saving the chosen model.

```{r}
save(chosen_model_2, file = 'Tennis_Duration_model.Rdata')
```

This paper was created using the platform provided by R ((R Core Team 2021)) and R-studio ((Team 2020)).

Couch, Simon P., Andrew P. Bray, Chester Ismay, Evgeni Chasnovski, Benjamin S. Baumer, and Mine Çetinkaya-Rundel. 2021. "{Infer}: An {r} Package for Tidyverse-Friendly Statistical Inference" 6: 3661. https://doi.org/10.21105/joss.03661.

Greenwell, Brandon M., and Bradley C. Boehmke. 2020. "Variable Importance Plots—an Introduction to the Vip Package" 12. https://doi.org/10.32614/RJ-2020-013.

Kuhn, Max, and Hadley Wickham. 2020. "Tidymodels: A Collection of Packages for Modeling and Machine Learning Using Tidyverse Principles." https://www.tidymodels.org.

Ovaska, et al, Tomi. 2014. "WHO HAS THE ADVANTAGE? AN ECONOMIC EXPLORATION OF WINNING IN MEN's PROFESSIONAL TENNIS." 59. http://www.jstor.org/stable/43664760.

R Core Team. 2021. "R: A Language and Environment for Statistical Computing." https://www.R-project.org/.

Sackmann, Jeff. 2022. "Jeff Sackmann Github Page." https://github.com/JeffSackmann/tennis_atp.

Sadigov, Tural. 2022. "MATH 254 GitHub Page." https://github.com/turalsadigov/MATH_254.

Team, RStudio. 2020. "RStudio: Integrated Development Environment for r." http://www.rstudio.com/.

Ward, Jennifer, and Maria Tackett. 2022. "USCLAP Competition." https://www.causeweb.org/usproc/usclap.

Wickham, Hadley. 2007. "Reshaping Data with the Reshape Package" 21. https://www.jstatsoft.org/v21/i12/.

———. 2016. "Ggplot2: Elegant Graphics for Data Analysis." https://ggplot2.tidyverse.org.

Wickham, Hadley, Mara Averick, Jennifer Bryan, Winston Chang, Lucy D'Agostino McGowan, Romain François, Garrett Grolemund, et al. 2019. "Welcome to the {Tidyverse}" 4: 1686. https://doi.org/10.21105/joss.01686.

Xie, Yihui. 2022. "Knitr: A General-Purpose Package for Dynamic Report Generation in r." https://yihui.org/knitr/.