UNIVERSIDADE DE SÃO PAULO - USP
ESCOLA DE ARTES, CIÊNCIAS E HUMANIDADES - EACH

BÁRBARA ALBUQUERQUE nUSP 9037585
DÉBORA ATANES BUSS nUSP 9276860
GABRIEL BARBOSA nUSP 7163395

ASSOCIATION OF TENNIS PROFESSIONALS
MATCHES: YEAR 2016

SÃO PAULO
2017

1. O DataSet

O DataSet consiste em 2941 partidas de tênis disputadas em 2016 através de 68 torneios, sendo 66 da Association of Tennis Professionals (ATP), 1 da International Tennis Federation (ITF) e 1 nos jogos olímpicos.

1.1. Origem

O DataSet foi encontrado na plataforma Kaggle[1] que foi criada para competições de modelagem preditiva e analítica, em que empresas e pesquisadores postam dados para que estatísticos e mineradores de dados possam competir para produzir os melhores modelos para prever e descrever os dados.

1.2. Fonte

Os dados do DataSet foram extraídos e estão disponíveis no site da ATP[2].

2. Sobre o Dataset - Tênis 101

Uma partida de tênis é disputada em uma quadra retangular por dois jogadores (simples) ou quatro (duplas). Os jogadores ficam em lados opostos de uma rede e usam uma raquete de cordas para bater uma bola de um lado para o outro.Cada jogador tem no máximo um quique de bola depois dela ter sido atingido por seu oponente para retorná-la pela rede e dentro dos limites da quadra. Uma vez que um jogador não faz nenhuma dessas três coisas, seu oponente ganha um ponto.

2.1. Games

Os games são disputados por pontos que são contados em 0, 15, 30 e 40. Em caso de empate 40 a 40, ganha quem fizer dois pontos seguidos.

2.2. Sets

Um jogo de tênis é divido em três ou cinco sets. Cada set possui seis games. Para ganhar um set é preciso ter uma diferença de dois games de vantagem sobre o adversário.

2.3. Tie break

Quando um set fica empatado em seis games à seis (6x6) a partida vai para o *tie break*, onde cada jogada vale um ponto. O primeiro a fazer sete pontos ou mais com dois pontos de diferença vence o set.

2.4. Saque

[1] Disponível em: <https://www.kaggle.com/gmadevs/atp-matches-dataset>.
[2] Disponível em: <http://www.atpworldtour.com>.

Diferente do vôlei, no tênis os jogadores sacam durante um game completo. O saque, também chamado de serviço, é onde o jogador tem duas chances para acertar a jogad. Caso erre o primeiro saque, pode sacar de novo. Caso erre o segundo saque a jogada é considerada dupla falta e o adversário sai na frente do placar no game.

2.5. Break Point

É o ponto em que, se vencido pelo receptor, resulta em uma quebra de serviço. Surge quando a pontuação é 30-40 ou 40-Adv.

Um duplo break point(ou dois break points) surge com a pontuação de 15-40.

Um triplo break point(ou três break points) surge em 00-40.

2.6. Superficies

Há quatro tipos de superfícies em que uma partida pode ser disputada, cada superfície causa uma diferença na velocidade e no quique da bola de tênis. As quatro superfícies são: Clay (Saibro), Hard(Piso duro), Grass(Grama) e Carpet(Carpete).

Figura 1 - Tipos de superfícies
Fonte : Imagens do Google.

Como pode ser visualizado na Figura 2, em termos de velocidade da bola e altura do quique o saibro é a mais lenta e com quique mais alto, enquanto a grama é a mais rápida e a com quique mais baixa (além de algumas quadras de carpete). Os pisos duros são intermediários.

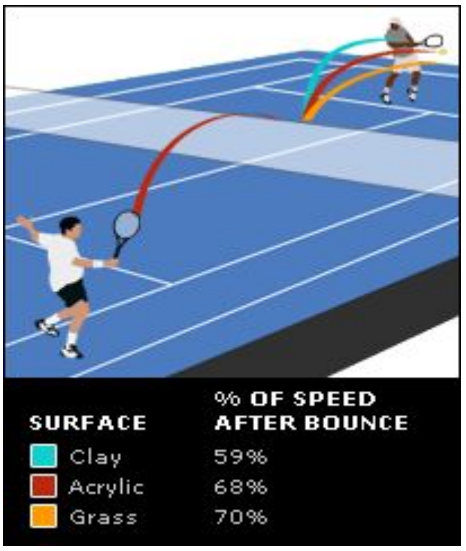

Figura 2 - Velocidade da bola após quicar
Fonte: BBC[3]

3. Variáveis

O DataSet é composto por 49 variáveis, sendo 31 quantitativas (27 discretas e 4 contínuas) e 18 qualitativas (10 nominais e 8 ordinais). Os dados são brutos.

3.1. Classificação das Variáveis

A classificação das variáveis do dataset está disposta na tabela 1.

Variável	Significado	Tipo	Subtipo
tourney_id	ID do torneio	Qualitativa	Ordinal
tourney_name	Nome do torneio	Qualitativa	Nominal
surface	Superfície em que o torneio é disputado	Qualitativa	Nominal
draw_size	Tamanho da chave do torneio	Quantitativa	Discreta
tourney_level	Nível do torneio	Qualitativa	Ordinal
tourney_date	Data do torneio	Qualitativa	Ordinal
match_num	Número da partida	Quantitativa	Discreta
winner_id	ID do vencedor da partida	Qualitativa	Ordinal
winner_seed	Número do vencedor da partida na chave do torneio	Qualitativa	Ordinal
winner_entry	Forma de entrada do vencedor da partida no torneio	Qualitativa	Nominal
winner_name	Nome do vencedor da partida	Qualitativa	Nominal
winner_hand	Mão dominante do vencedor da partida	Qualitativa	Nominal

[3] Disponível em: <http://news.bbc.co.uk/sport2/hi/tennis/skills/5274588.stm>.

winner_ht	Altura do vencedor da partida	Quantitativa	Contínua
winner_ioc	Nacionalidade do vencedor da partida	Qualitativa	Nominal
winner_age	Idade do vencedor da partida	Quantitativa	Contínua
winner_rank	Ranking do vencedor da partida	Quantitativa	Discreta
winner_rank_points	Pontos acumulados no ranking pelo vencedor da partida	Quantitativa	Discreta
loser_id	ID do perdedor da partida	Qualitativa	Ordinal
loser_seed	Número do perdedor da partida na chave do torneio	Qualitativa	Nominal
loser_entry	Forma de entrada do perdedor da partida no torneio	Qualitativa	Ordinal
loser_name	Nome do perdedor da partida	Qualitativa	Nominal
loser_hand	Mão dominante do perdedor da partida	Qualitativa	Nominal
loser_ht	Altura do perdedor da partida	Quantitativa	Contínua
loser_ioc	Nacionalidade do perdedor da partida	Qualitativa	Nominal
loser_age	Idade do perdedor da partida	Quantitativa	Contínua
loser_rank	Ranking do perdedor da partida	Quantitativa	Discreta
loser_rank_points	Pontos acumulados no ranking pelo perdedor da partida	Quantitativa	Discreta
score	Placar da partida	Quantitativa	Discreta
best_of	Quantidade máxima de sets disputados na partida	Quantitativa	Discreta
round	Rodada em que a partida foi disputada	Qualitativa	Ordinal
minutes	Duração da partida	Quantitativa	Discreta
w_ace	Número de aces do vencedor	Quantitativa	Discreta
w_df	Número de faltas duplas do vencedor	Quantitativa	Discreta
w_svpt	Número de pontos vencidos pelo vencedor	Quantitativa	Discreta
w_1stIn	Pontos em que o vencedor acertou o primeiro saque	Quantitativa	Discreta
w_1stWon	Pontos ganhos com o primeiro saque do vencedor	Quantitativa	Discreta
w_2ndWon	Pontos ganhos com o segundo saque do vencedor	Quantitativa	Discreta
w_SvGms	Número de games de saque jogados pelo vencedor	Quantitativa	Discreta
w_bpSaved	Número de break points salvos pelo vencedor	Quantitativa	Discreta
w_bpFaced	Número de break points enfrentados pelo vencedor	Quantitativa	Discreta

l_ace	Número de aces do perdedor	Quantitativa	Discreta
l_df	Número de faltas duplas do perdedor	Quantitativa	Discreta
l_svpt	Número de pontos vencidos pelo perdedor	Quantitativa	Discreta
l_1stIn	Pontos em que o perdedor acertou o primeiro saque	Quantitativa	Discreta
l_1stWon	Pontos ganhos com o primeiro saque do perdedor	Quantitativa	Discreta
l_2ndWon	Pontos ganhos com o segundo saque do perdedor	Quantitativa	Discreta
l_SvGms	Número de games de saque jogados pelo perdedor	Quantitativa	Discreta
l_bpSaved	Número de break points salvos pelo perdedor	Quantitativa	Discreta
l_bpFaced	Número de break points enfrentados pelo perdedor	Quantitativa	Discreta

Tabela 1 - Classificação das variáveis do DataSet
Fonte: Criação própria.

4. Conteúdo do Banco (Quantidade de Dados e Esparsidade)

O Banco de dados possuía inicialmente **3004** linhas. A fim de entender melhor o quão completo estava nosso banco, aplicamos uma função chamada ***missmap( )*** da biblioteca ***"Amelia"*** em nosso *dataset* que cria um mapeamento apontando dados faltantes. O resultado pode ser conferido na figura a seguir.

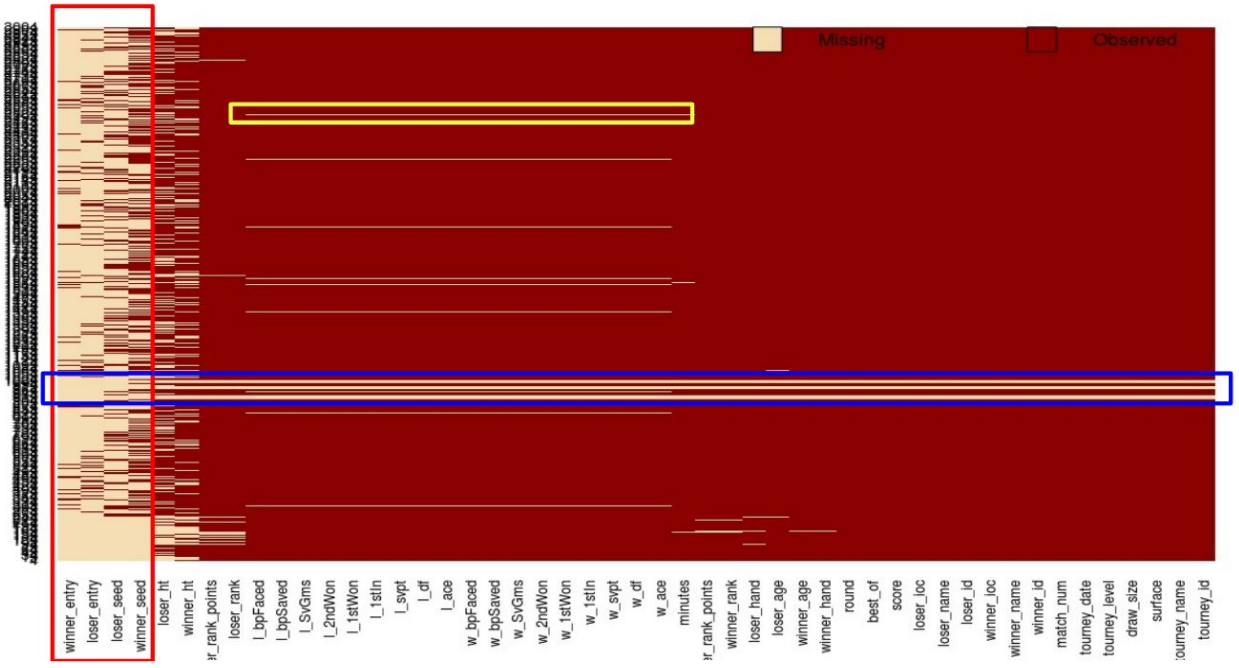

Figura 3 - Dados faltantes no dataset - "Missingness Map"
Fonte: Criação Própria.

Dados existentes no banco foram coloridos de vermelho enquanto dados faltantes foram coloridos de rosa claro. Em azul, destacamos linhas completamente em branco que existiam no banco. Essas linhas serviam de divisória entre um campeonato e outro, separando suas partidas, ignorando-as ficamos com um total de 2941 linhas, ou seja, **2941 partidas de tênis**. Em amarelo, são exemplos de dados faltantes porém recuperáveis através de pesquisas. Em vermelho destacamos quatro variáveis (winner_entry, loser_entry, winner_seed e loser_seed) que apresentam uma esparsidade muito grande, tão grande que as tornam descartáveis para nosso propósito.

Com o dataset dessa forma, se retirarmos as linhas que contém no mínimo um dado faltante ficamos com um total de **zero linhas**. Retirando as quatro variáveis que apresentam maior grau de esparsidade (já destacadas) do dataset e aplicando o mesmo filtro de retirar linhas com no mínimo um dado faltante ficamos com um total de 1508 linhas.

5. Descrição das Variáveis

5.1. winner_age e loser_age

Na figura 4 está disposta a distribuição das variáveis *winner_age* (idade do vencedor) e *loser_age* (idade do perdedor). Analisando-se a figura observa-se que não existe diferença na distribuição entre as idades dos jogadores vencedores e perdedores. Também nota-se que a maior parte dos jogadores se concentra na faixa etária de 25 à 30 anos.

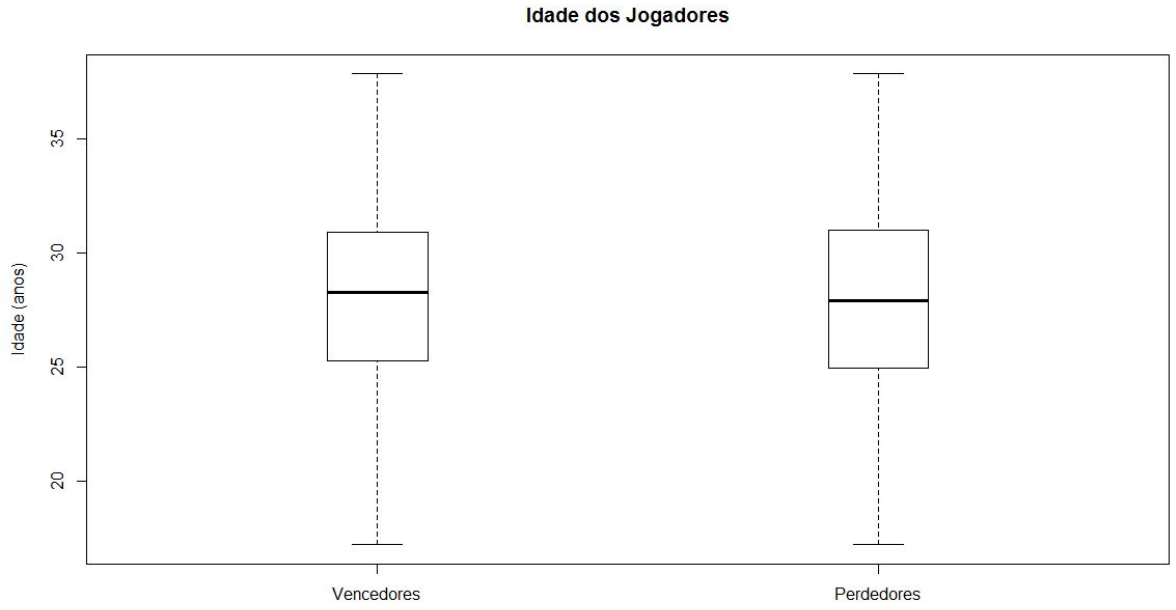

Figura 4 - Boxplot da Idade dos Jogadores
Fonte: Criação Própria.

5.2. winner_age e loser_age

Na figura 5 está disposta a função densidade das variáveis *winner_age* (idade do vencedor) e *loser_age* (idade do perdedor). A função é semelhante a função da distribuição normal. Analisando-se a figura observa-se que a maior parte dos jogadores se concentra na faixa etária de 25 à 30 anos.

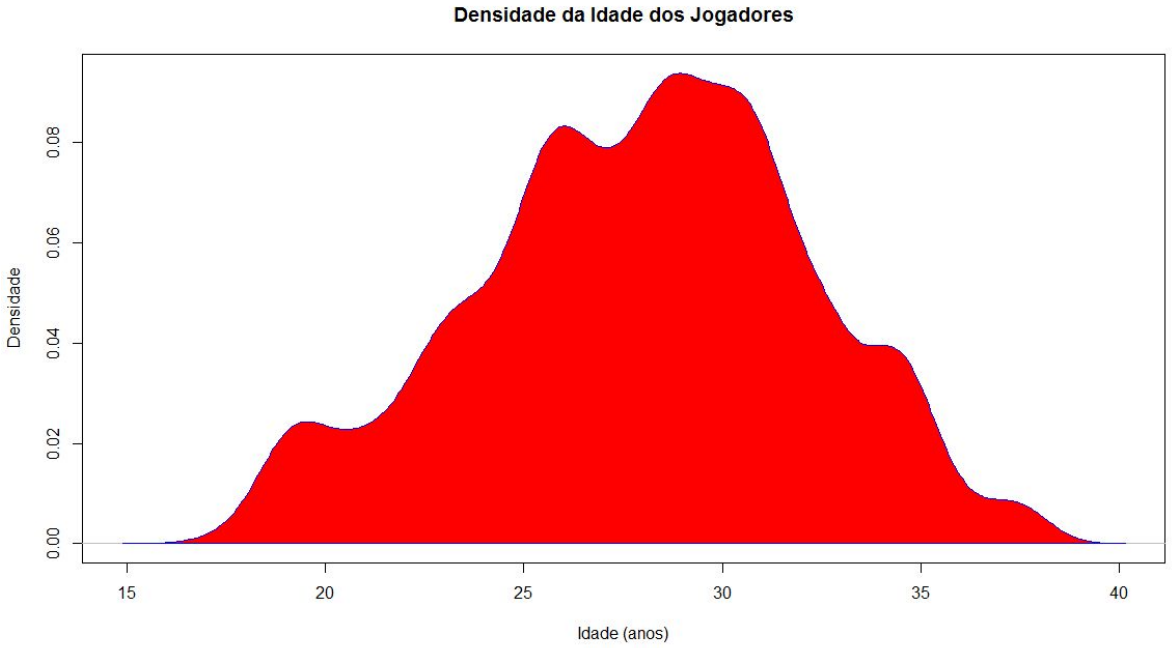

Figura 5 - Densidade da Idade dos Jogadores
Fonte: Criação Própria.

5.3. winner_ht e loser_ht

Na Figura 6 está disposta a distribuição das variáveis *winner_ht* (altura do vencedor) e *loser_ht* (altura do perdedor). Analisando-se a figura observa-se que existe uma leve diferença na distribuição entre as alturas dos jogadores vencedores e perdedores.

Os jogadores vencedores são ligeiramente mais altos com uma mediana de 188 centímetros enquanto que os perdedores apresentam uma mediana de 185 centímetros. Também nota-se que a maior parte dos jogadores apresentam altura entre 183 e 190 centímetros.

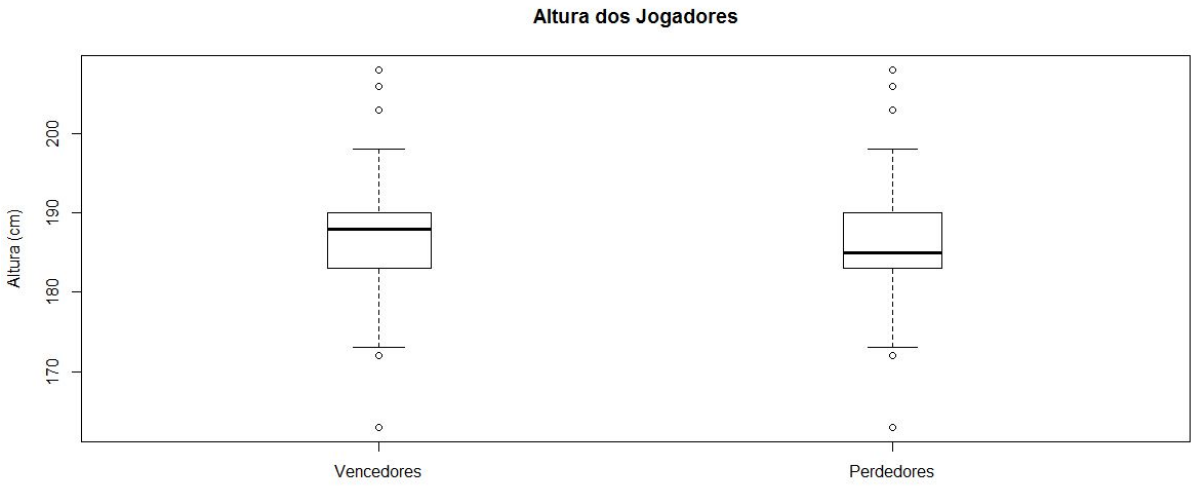

Figura 6 - Boxplot da Altura dos Jogadores
Fonte: Criação Própria.

5.4. winner_ht e loser_ht

Na Figura 7 está disposta a função densidade das variáveis *winner_ht* (altura do vencedor) e *loser_ht* (altura do perdedor). A função é semelhante a função da distribuição normal. Analisando-se a figura observa-se que a maior parte dos jogadores apresentam altura entre 183 e 190 centímetros.

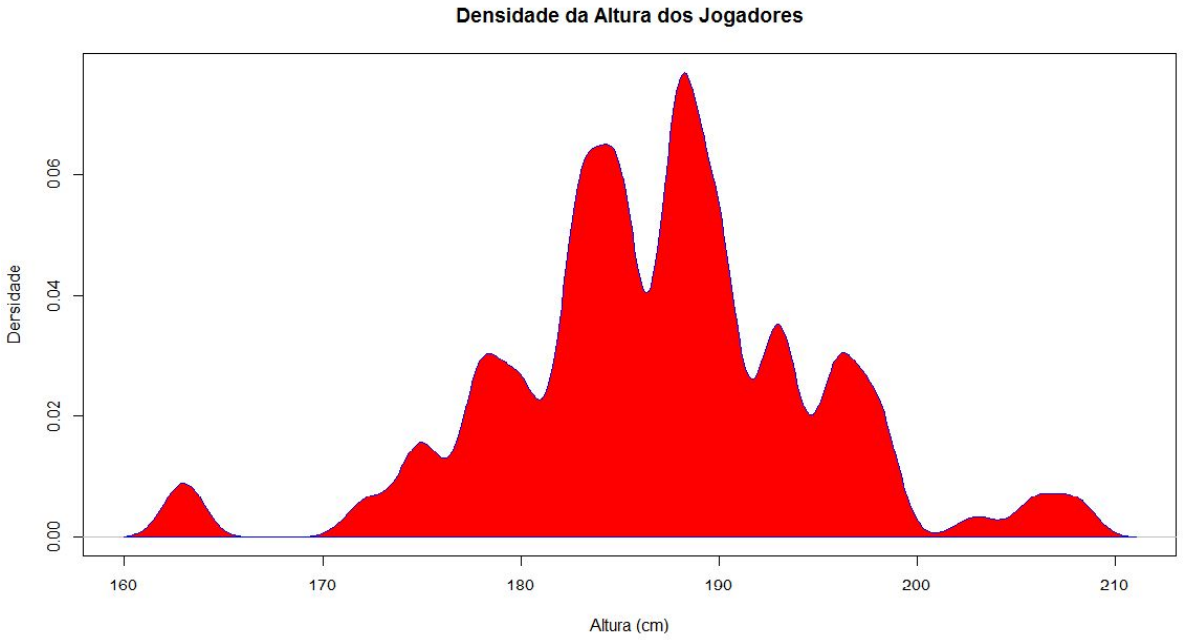

Figura 7 - Densidade da Altura dos Jogadores
Fonte: Criação Própria.

5.5. w_ace, winner_ht , l_ace e loser_ht

Na figura 8 está disposta o boxplot da distribuição das variáveis *w_ace* (número de aces do vencedor) e *l_ace* (número de aces do perdedor) pelas

variáveis *winner_ht* (altura do vencedor), *loser_ht* (altura do perdedor). Analisando-se a figura observa-se que jogadores mais altos tendem a acertar um maior número de aces que jogadores mais baixos.

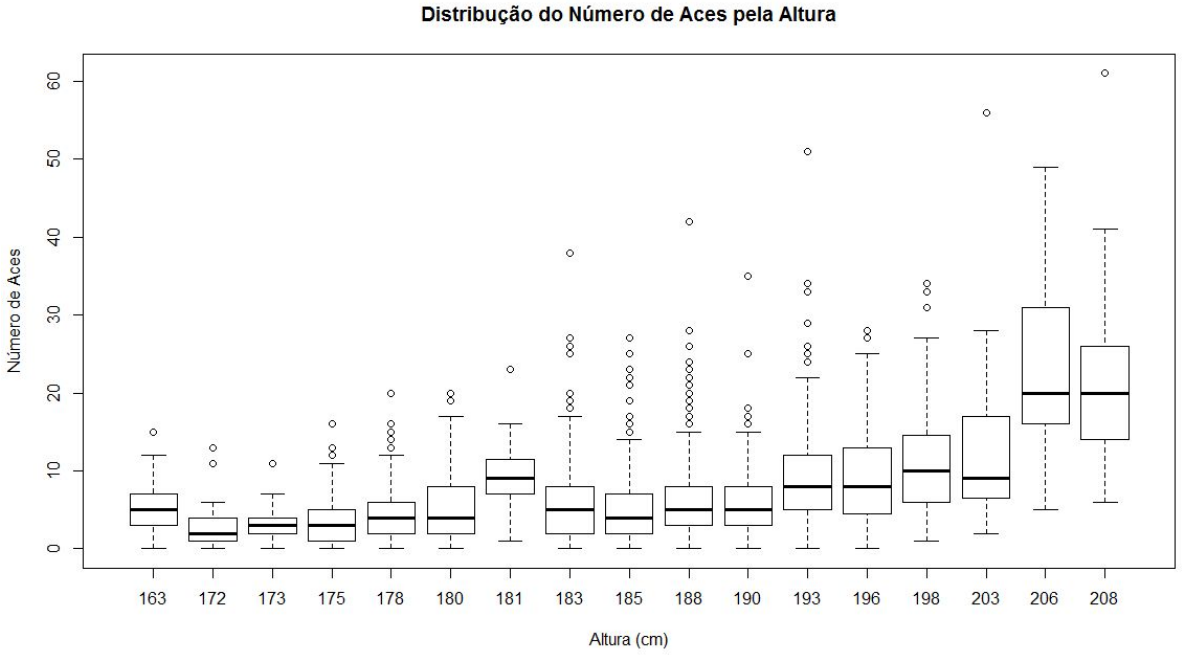

Figura 8 - Boxplot Distribuição do Número de Aces pela altura
Fonte: Criação Própria.

5.6. surface

Devido ao fato de que o elemento comum a uma partida de tênis é que a mesma é disputada em uma quadra, variando apenas a superfície da mesma, nos concentramos em tentar dispor as demais variáveis segundo esse elemento comum, para que assim tenhamos uma maior compreensão do DataSet.

Na figura 9 está disposta a distribuição do número de torneios pela variável *surface* (superfície). Analisando-se a figura observa-se que a maior parte dos torneios é disputado nas superfícies *Hard* (piso duro) e *Clay* (Saibro). Também destaca-se o fato de que apenas um torneio é disputado na superfície *Carpet* (Carpete).

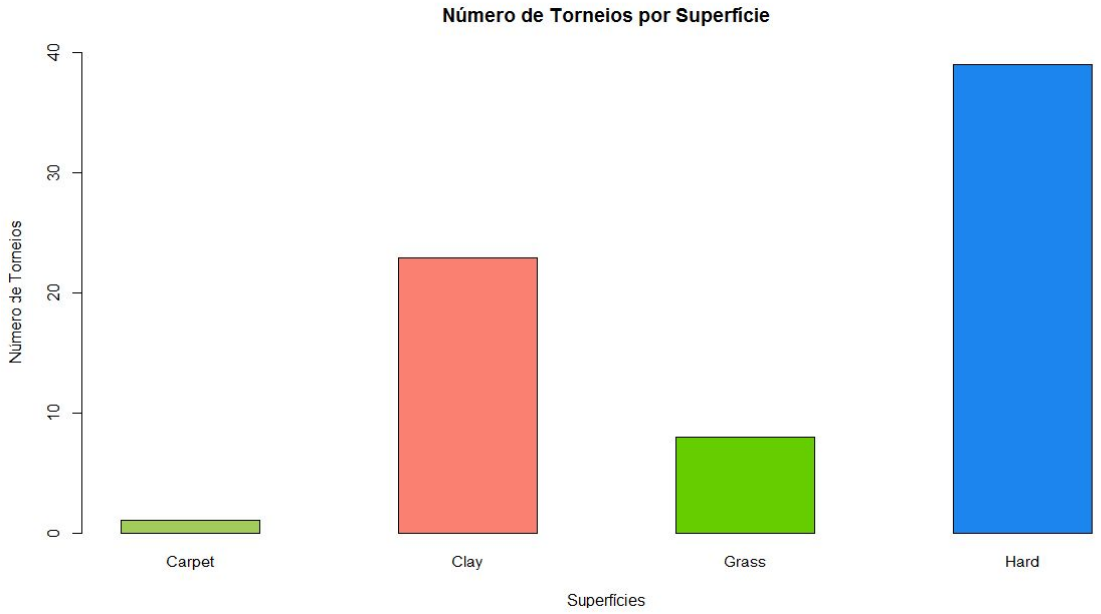

Figura 9 - Número de Torneios por Superfície
Fonte: Criação Própria.

Na figura 10 está disposta a distribuição do número de partidas pela variável surface. Analisando-se a figura observa-se que a maior parte dos jogos é disputado nas superfícies *Hard* (piso duro) e *Clay* (Saibro). Também destaca-se o fato de que apenas 11 partidas foram disputadas na superfície *Carpet* (Carpete), o que nos leva a considerar a remoção das partidas disputadas nesta superfície pela baixa amostragem da mesma.

Figura 10 - Número de Partidas por Superfície
Fonte: Criação Própria.

5.6.1. w_ace, l_ace e surface

Na figura 11 está disposta a distribuição das médias das variáveis *w_ace* (número de aces do vencedor) e *l_ace* (número de aces do perdedor) pela variável *surface* (superfície). Analisando-se a figura observa-se que é mais fácil acertar aces na superfície *Grass* (Grama) do que nas demais superfícies.

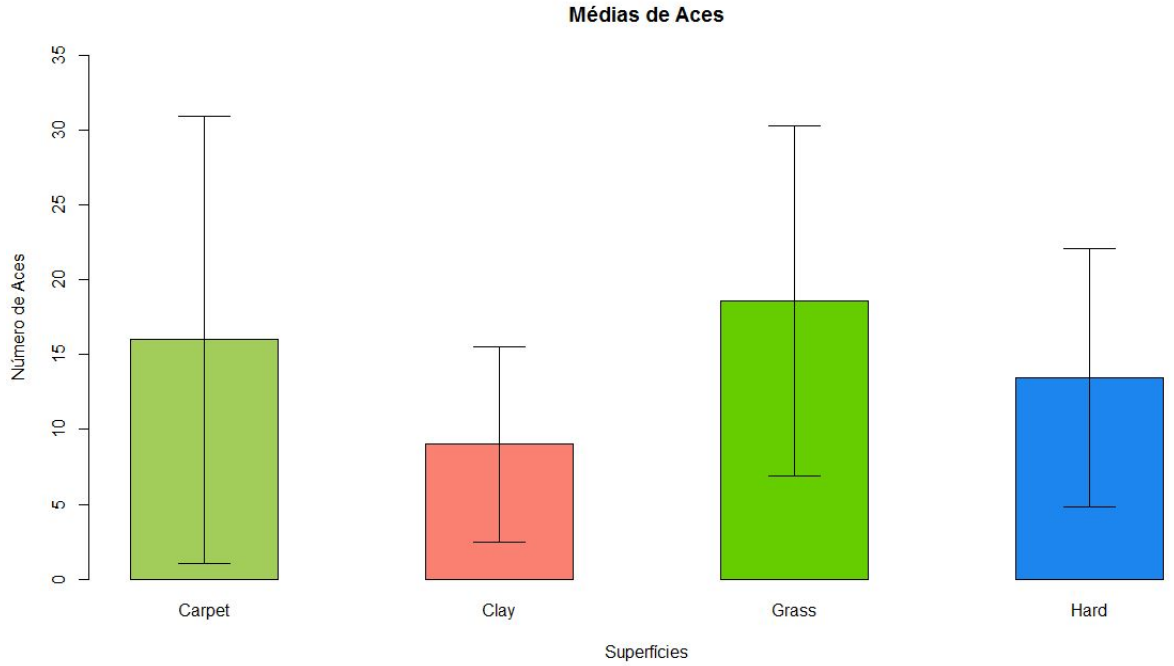

Figura 11 - Médias de Aces por Superfície
Fonte: Criação Própria.

5.6.2. w_df, l_df e surface

Na figura 12 está disposta a distribuição das médias das variáveis *w_df* (número de duplas faltas do vencedor) e *l_df* (número de duplas faltas do perdedor) pela variável *surface* (superfície). Analisando-se a figura observa-se que não existem diferenças significativas.

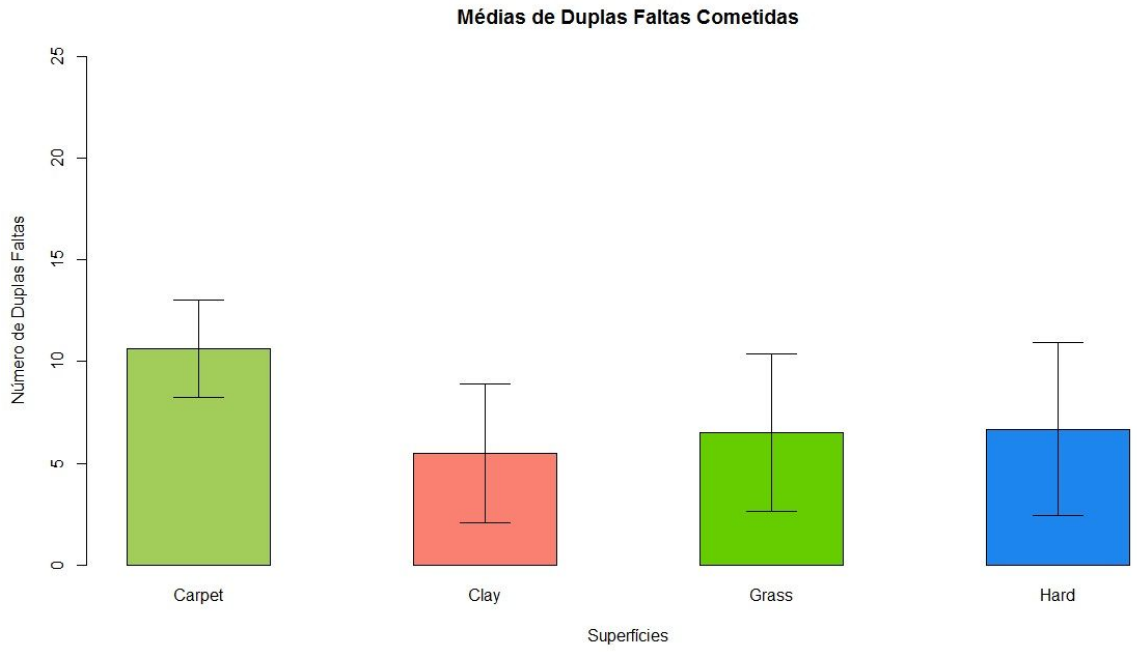

Figura 12 - Médias de Duplas Faltas Cometidas por Superfície
Fonte: Criação Própria.

5.6.3. w_bpFaced, l_bpFaced e Surface

Na figura 13 está disposta a distribuição das médias das variáveis *w_bpFaced* (número de break points enfrentados pelo vencedor) e *l_bpFaced* (número break points enfrentados pelo perdedor) pela variável *surface* (superfície). Analisando-se a figura observa-se que os jogadore enfrentam mais break points na superfície *Clay* (Saibro) do que nas demais superfícies.

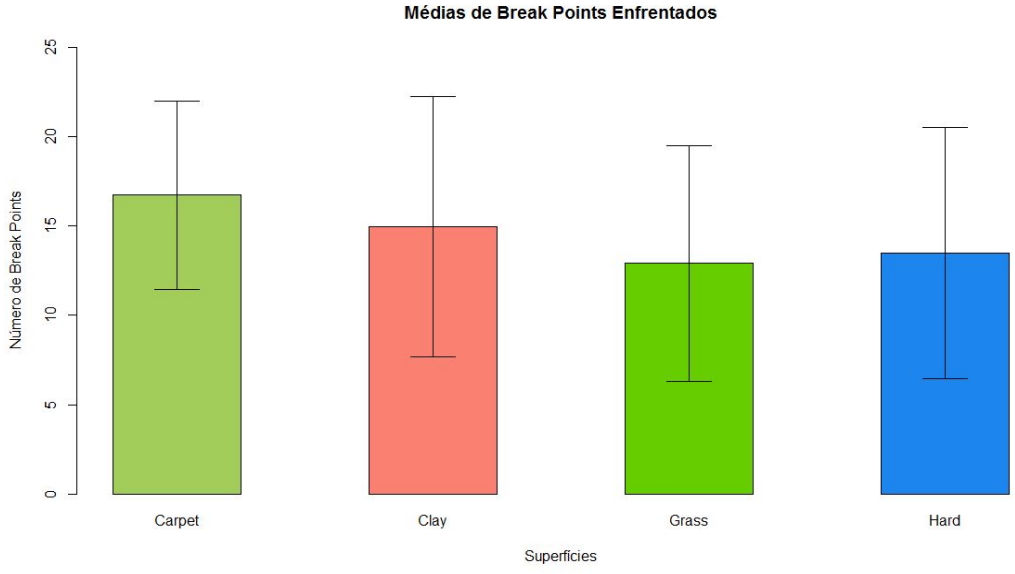

Figura 13 - Médias de Duplas Break points Enfrentados por Superfície
Fonte: Criação Própria.

5.6.4. w_bpSaved, l_bpSaved e surface

Na figura 14 está disposta a distribuição das médias das variáveis *w_bpSaved* (número de break points salvos pelo vencedor) e (número de break points salvos pelo perdedor) pela variável *surface* (superfície). Analisando-se a figura observa-se que não existem diferenças significativas.

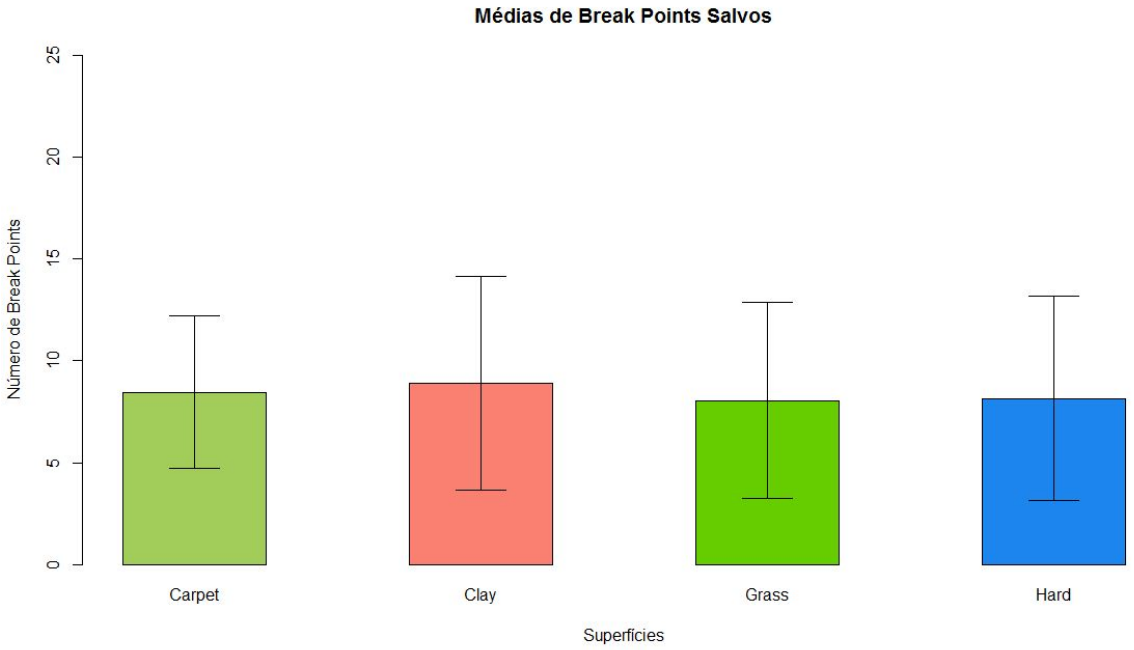

Figura 14 - Médias de Duplas Break points Salvos por Superfície
Fonte: Criação Própria.

5.6.5. w_1stWon, l_1stWon e surface

Na figura 15 está disposta a distribuição das médias das variáveis *w_1stWon* (número de pontos ganhos com o 1º saque do vencedor) e *l_1stWon* (número de pontos ganhos com o 1º saque do perdedor) pela variável *surface* (superfície). Analisando-se a figura observa-se que é mais fácil ganhar pontos com o primeiro saque na superfície *Grass* (Grama) do que nas demais superfícies.

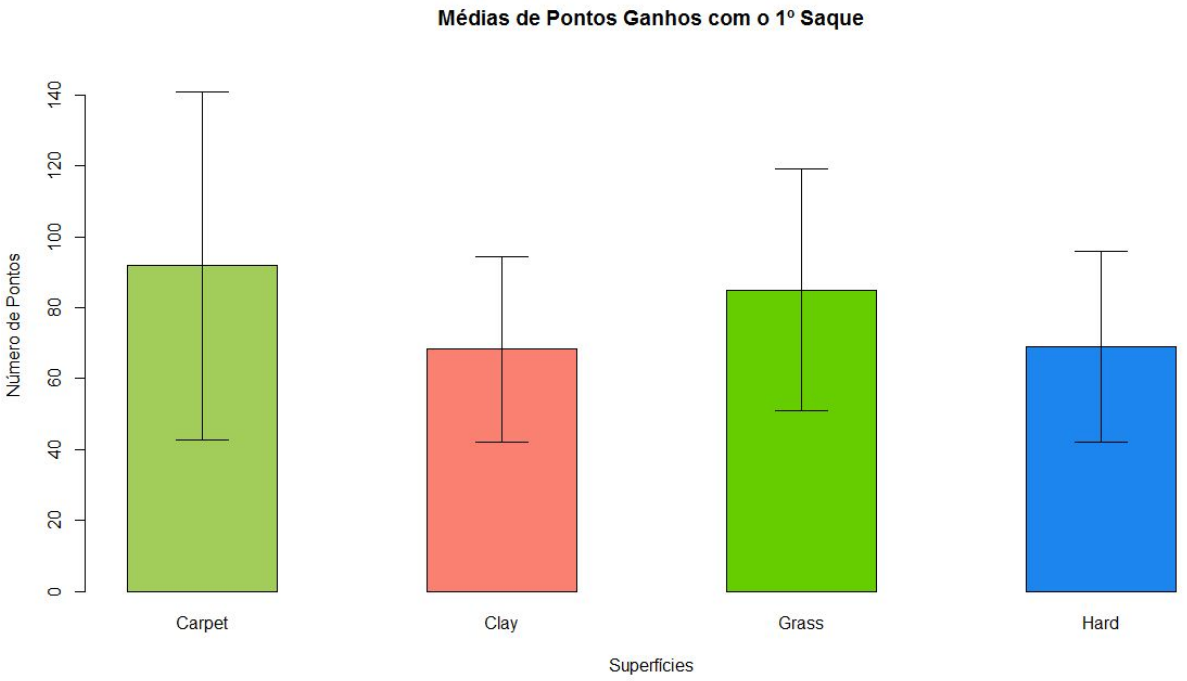

Figura 15 - Médias de Pontos Ganhos com o 1º Saque por Superfície
Fonte: Criação Própria.

5.6.6. w_2ndWon, l_2ndWon e surface

Na figura 16 está disposta a distribuição das médias das variáveis *w_2ndWon* (número de pontos ganhos com o 2º saque do vencedor) e *l_2ndWon* (número de pontos ganhos com o 2º saque do perdedor) pela variável *surface* (superfície). Analisando-se a figura observa-se que não existem diferenças significativas.

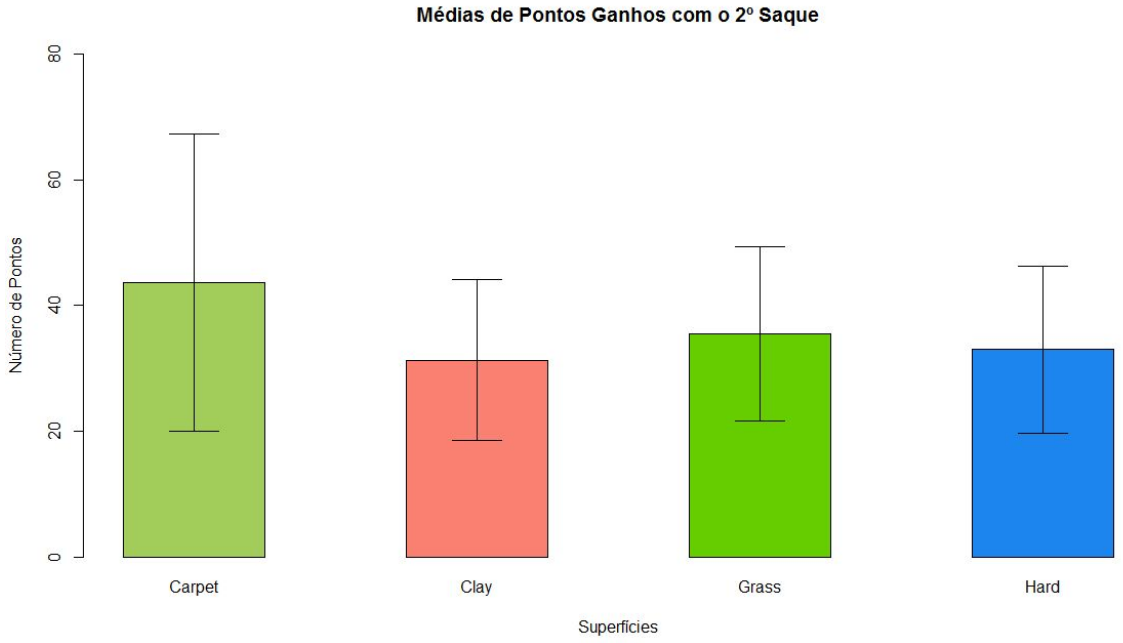

Figura 16 - Médias de Pontos Ganhos com o 2º Saque por Superfície
Fonte: Criação Própria.

5.6.7. w_1stIn, l_1stIn, w_2stIn, l_2stIn e surface

Na figura 17 está disposta a distribuição das médias das variáveis *w_1stWon* (número de pontos ganhos com o 1º saque do vencedor), *l_1stWon* (número de pontos ganhos com o 1º saque do perdedor), *w_2ndWon* (número de pontos ganhos com o 2º saque do vencedor) e *l_2ndWon* (número de pontos ganhos com o 2º saque do perdedor) pela variável *surface* (superfície). Analisando-se a figura observa-se que as médias de pontos ganhos com o 2º saque é cerca de metade as médias de pontos ganhos com o 1º saque .

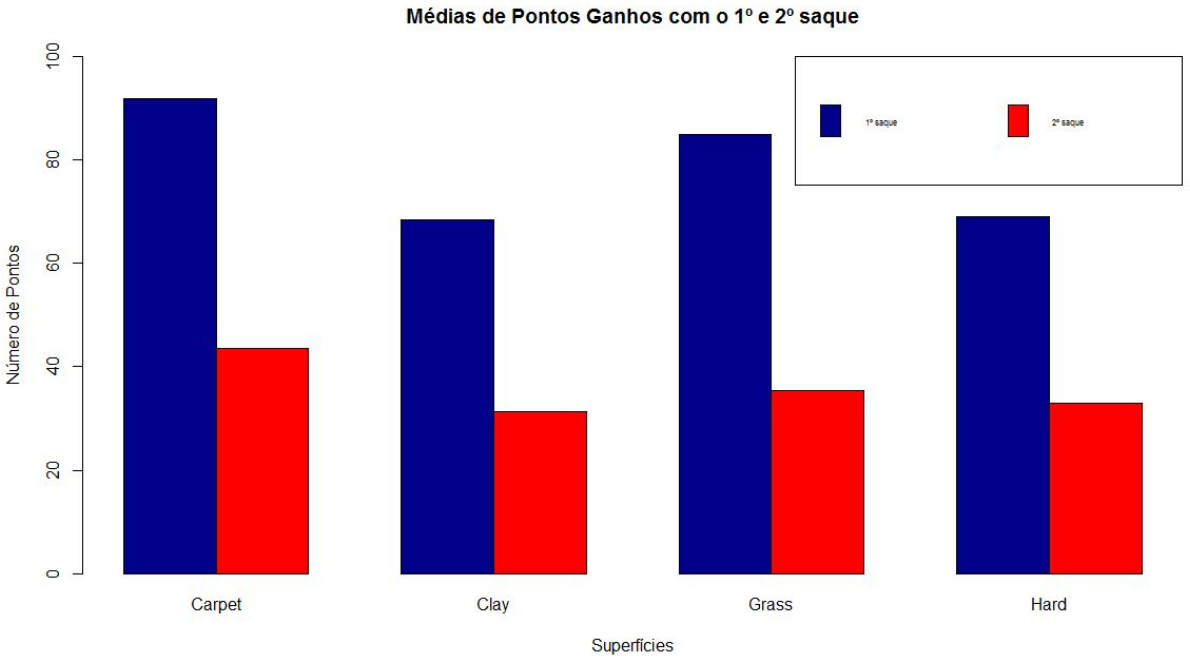

Figura 17 - Médias de Pontos Ganhos com o 1º Saque e 2º Saque por Superfície
Fonte: Criação Própria.

5.6.8. w_svpt, l_svpt e surface

Na figura 18 está disposta a distribuição das médias das variáveis *w_svpt* (número de pontos ganhos do vencedor) e *l_svpt* (número de pontos ganhos do perdedor) pela variável *surface* (superfície). Analisando-se a figura observa-se que não existem diferenças significativas.

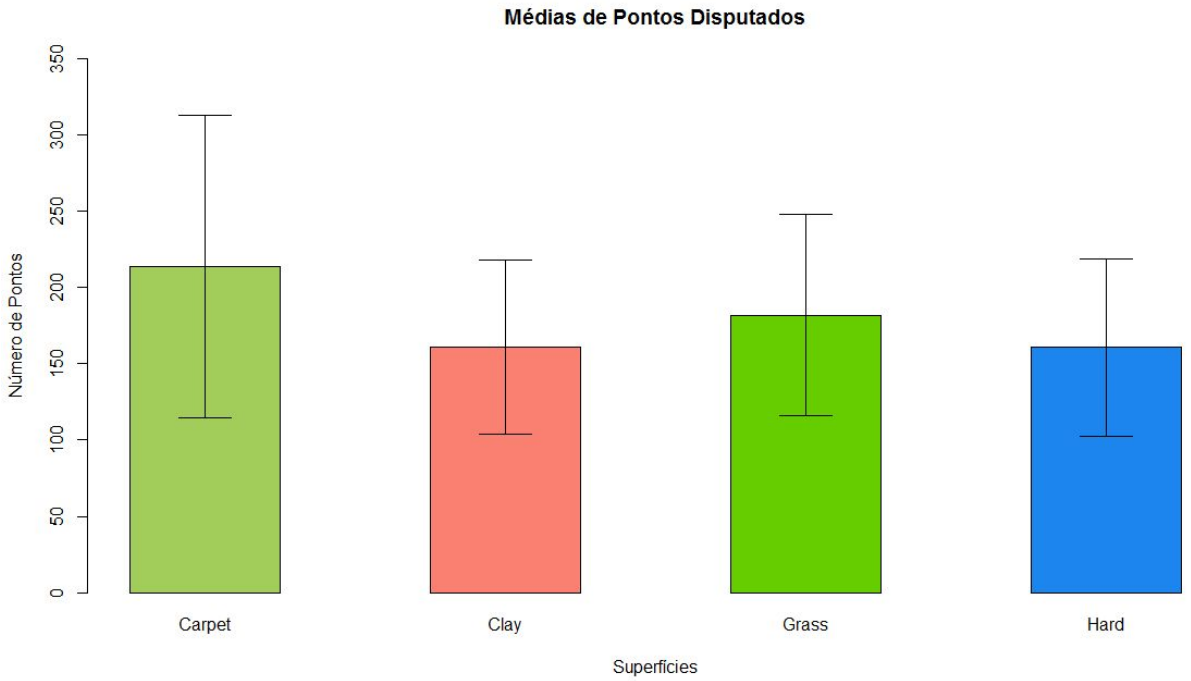

Figura 18 - Médias de Pontos Disputados por Superfície
Fonte: Criação Própria.

5.7. minute e surface

Na figura 19 está disposta a distribuição da variável *minute* (duração da partida) pela variável *surface* (superfície). Analisando-se a figura observa-se que não existem diferenças significativas.

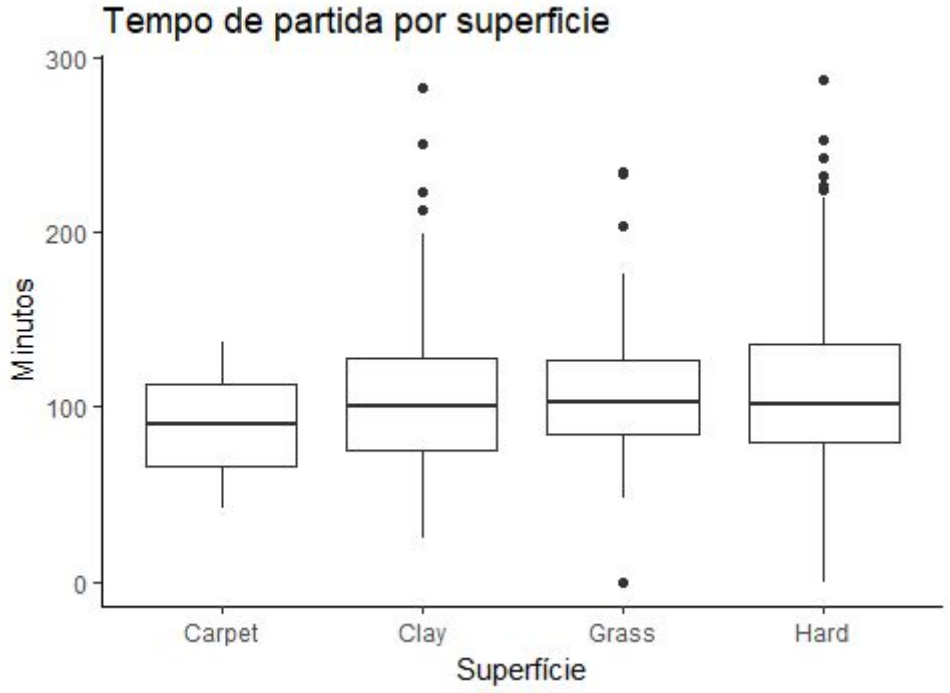

Figura 19 - Tempo de Partida por Superfície
Fonte: Criação Própria.

6. Pergunta

Com esse dataset nos propomos a tentar responder a seguinte pergunta: Uma vez que diferentes superfícies afetam uma partida de tênis de formas diferentes (de tal forma dos jogadores inclusive se especializarem seu jogo em certa superfície), é possível ***agrupar*** partidas de modo que se possa ***identificar a superfície em que as mesmas foram jogadas***?

7. Apêndice: Scripts

7.1. Missingness Map 1

```
library(Amelia)

# Ler Dados
raw.data = read_csv("atp_matches_2016.csv")

# Plotando gráfico de missing data
missmap(raw.data)

#Pegando dimensão do data frame sem as linhas que possuem no
mínimo um valor faltando
clean.data = na.omit(raw.data)
dim(clean.data)
```

7.2. Missingness Map 2 (slides)

```
library(Amelia)

# Ler Dados
raw.data = read_csv("atp_matches_2016.csv")

# Retirando Colunas Descartáveis
excluir = c("looser_entry", "winner_entry", "loser_seed",
"winner_seed")
clean.data.frame = raw.data[,!(names(raw.data)%in%excluir)]

# Plotando gráfico de missing data
missmap(clean.data.frame)

#Pegando dimensão do data frame sem as linhas que possuem no mínimo
um valor faltando
clean.data = na.omit(clean.data.frame)
dim(clean.data)
```

7.3. Boxplot da Idade dos Jogadores

```
# Ler Dados
atp_matches_2016 = read_csv("atp_matches_2016.csv")

# Plotando gráfico
boxplot(atp_matches_2016$winner_age, atp_matches_2016$loser_age,
        main = "Idade dos Jogadores",
        ylab= "Idade (anos)",
        names = c("Vencedores", "Perdedores"),
        boxwex = 0.2,
        varwidth=TRUE,
        na.rm = TRUE)
```

7.4. Densidade da Idade dos Jogadores

```
# Ler Dados
atp_matches_2016 = read_csv("atp_matches_2016.csv")

# Plotando gráfico
d <- density(atp_matches_2016$winner_age, na.rm = TRUE)

plot(d, main = "Densidade da Idade dos Jogadores",
     xlab = "Idade (anos)", ylab = "Densidade")

polygon(d, col = "RED", border = "BLUE")
```

7.5. Boxplot da Altura dos Jogadores

```
# Ler Dados
atp_matches_2016 = read_csv("atp_matches_2016.csv")

# Plotando gráfico
boxplot(atp_matches_2016$winner_ht,
        atp_matches_2016$loser_ht,
        main="Altura dos Jogadores",
        ylab= "Altura (cm)",
        names = c("Vencedores", "Perdedores"),
        boxwex = 0.2,
        varwidth=TRUE,
        na.rm = TRUE)
```

7.6. Densidade da Altura dos Jogadores

```
# Ler Dados
atp_matches_2016 = read_csv("atp_matches_2016.csv")

# Plotando gráfico
d<-density(atp_matches_2016$winner_ht, na.rm = TRUE)

plot(d, main = "Densidade da Altura dos Jogadores",
     xlab = "Altura (cm)", ylab = "Densidade")

polygon(d, col = "RED", border = "BLUE")
```

7.7. Boxplot da Distribuição do Número de Aces pela Altura

```
# Ler Dados
atp_matches_2016 = read_csv("atp_matches_2016.csv")

# Plotando gráfico
df <- stack(atp_matches_2016, select=c("winner_ht","loser_ht"))
df2 <- stack(atp_matches_2016, select=c("w_ace","l_ace"))

boxplot(df2$values~df$values,
        main="Distribução do Número de Aces pela Altura",
        ylab="Número de Aces",
        xlab="Altura (cm)",
        na.rm=TRUE)
```

7.8. Número de Torneios por Superfície

```
# Ler Dados
atp_matches_2016 = read_csv("atp_matches_2016.csv")

for(i in 2691:length(atp_matches_2016$tourney_name))
     atp_matches_2016[i, 2] <- "Davis Cup"

count.tourney_surfaces = tapply(atp_matches_2016$tourney_name,
                         atp_matches_2016$surface,
                         FUN = function(x) length(unique(x)))

barplot(count.tourney_surfaces,
        main = "Número de Torneios por Superfície",
        col = c("darkolivegreen3", "salmon", "chartreuse3",
                "dodgerblue2"),
        names.arg = c("Carpet", "Clay","Grass","Hard"),
        ylim = c(0,40),
        ylab = "Número de Torneios",
        xlab = "Superfícies",
        space = 1)
```

7.9. Número de Partidas por Superfície

```
# Ler Dados
atp_matches_2016 = read_csv("atp_matches_2016.csv")

# Plotando gráfico
count.matches_surfaces = tapply(atp_matches_2016$surface,
                                atp_matches_2016$surface,
                                FUN = function(x) length(x))
barplot(count.matches_surfaces,
        main = "Número de Jogos por Superfí-cie",
        col = c("darkolivegreen3", "salmon", "chartreuse3",
                "dodgerblue2"),
        names.arg = c("Carpet", "Clay","Grass","Hard"),
        ylim = c(0,2000),
        ylab = "Número de Jogos",
        xlab = "Superfí-cies",
        space = 1)
```

7.10. Médias de Aces por Superfície

```
# Ler Dados
atp_matches_2016 = read_csv("atp_matches_2016.csv")

# Plotando gráfico

mean.surfaces=tapply(atp_matches_2016$w_ace+atp_matches_2016$l_ace,
                     atp_matches_2016$surface, mean, na.rm=TRUE)
sd.surfaces=tapply(atp_matches_2016$w_ace+atp_matches_2016$l_ace,
                   atp_matches_2016$surface, sd, na.rm=TRUE)

mids = barplot(mean.surfaces, main = "Médias de Aces",
               col = c("darkolivegreen3", "salmon", "chartreuse3",
                       "dodgerblue2"),
               names.arg = c("Carpet", "Clay","Grass","Hard"),
               ylim = c(0,35),
               ylab = "Número de Aces",
               xlab = "Superfícies",
               space = 1)

arrows(mids, mean.surfaces-sd.surfaces, mids,
       mean.surfaces+sd.surfaces, angle = 90, code=3)
```

7.11. Médias de Duplas Faltas Cometidas por Superfície

```
# Ler Dados
atp_matches_2016 = read_csv("atp_matches_2016.csv")

# Plotando gráfico

mean.surfaces=tapply(atp_matches_2016$w_df+atp_matches_2016$l_df,
                     atp_matches_2016$surface, mean, na.rm=TRUE)
sd.surfaces=tapply(atp_matches_2016$w_df+atp_matches_2016$l_df,
                   atp_matches_2016$surface, sd, na.rm=TRUE)

mids=barplot(mean.surfaces,
             main = "Médias de Duplas Faltas Cometidas",
             col = c("darkolivegreen3", "salmon", "chartreuse3",
                     "dodgerblue2"),
             names.arg = c("Carpet", "Clay","Grass","Hard"),
             ylim = c(0,25),
             ylab = "Número de Duplas Faltas",
             xlab = "Superfícies", space = 1)

arrows(mids, mean.surfaces-sd.surfaces, mids,
       mean.surfaces+sd.surfaces, angle = 90, code=3)
```

7.12. Médias de Duplas Break points Enfrentados por Superfície

```
# Ler Dados
atp_matches_2016 = read_csv("atp_matches_2016.csv")

# Plotando gráfico
mean.surfaces=tapply(atp_matches_2016$w_bpFaced+atp_matches_2016$l_bpFaced,
                     atp_matches_2016$surface, mean, na.rm=TRUE)
sd.surfaces=tapply(atp_matches_2016$w_bpFaced+atp_matches_2016$l_bpFaced,
                   atp_matches_2016$surface, sd, na.rm=TRUE)
mids=barplot(mean.surfaces,
             main = "Médias de Break Points Enfrentados",
             col = c("darkolivegreen3", "salmon", "chartreuse3",
                     "dodgerblue2"),
             names.arg = c("Carpet", "Clay","Grass","Hard"),
             ylim = c(0,25),
             ylab = "Número de Break Points",
             xlab = "Superfí-cies",
             space = 1)
arrows(mids, mean.surfaces-sd.surfaces,
       mids, mean.surfaces+sd.surfaces,
       angle = 90, code=3)
```

7.13. Médias de Duplas Break points Salvos por Superfície

```
# Ler Dados
atp_matches_2016 = read_csv("atp_matches_2016.csv")

# Plotando gráfico
mean.surfaces=tapply(atp_matches_2016$w_bpSaved+atp_matches_2016$l_bpSaved,
                     atp_matches_2016$surface, mean, na.rm=TRUE)
sd.surfaces=tapply(atp_matches_2016$w_bpSaved+atp_matches_2016$l_bpSaved,
                   atp_matches_2016$surface, sd, na.rm=TRUE)

mids = barplot(mean.surfaces, main = "Médias de Break Points Salvos",
               col = c("darkolivegreen3","salmon","chartreuse3","dodgerblue2"),
               names.arg = c("Carpet", "Clay","Grass","Hard"),
               ylim = c(0,25),
               ylab = "Número de Break Points",
               xlab = "Superfí-cies",
               space = 1)
arrows(mids, mean.surfaces-sd.surfaces,
       mids, mean.surfaces+sd.surfaces,
       angle = 90, code=3)
```

7.14. Médias de Pontos Ganhos com o 1º Saque por Superfície

```
# Ler Dados
atp_matches_2016 = read_csv("atp_matches_2016.csv")

# Plotando gráfico
mean.surfaces=tapply(atp_matches_2016$w_1stWon+atp_matches_2016$l_1stWon,
                     atp_matches_2016$surface, mean, na.rm=TRUE)

sd.surfaces=tapply(atp_matches_2016$w_1stWon+atp_matches_2016$l_1stWon,
                   atp_matches_2016$surface, sd, na.rm=TRUE)

mids = barplot(mean.surfaces,
               main = "Médias de Pontos Ganhos com o 1º Saque",
               col = c("darkolivegreen3", "salmon", "chartreuse3",
                       "dodgerblue2"),
               names.arg = c("Carpet", "Clay","Grass","Hard"),
               ylim = c(0,150),
               ylab = "Número de Pontos",
               xlab = "Superfícies",
               space = 1)
arrows(mids, mean.surfaces-sd.surfaces,
       mids, mean.surfaces+sd.surfaces,
       angle = 90, code=3)
```

7.15. Médias de Pontos Ganhos com o 2º Saque por Superfície

```
# Ler Dados
atp_matches_2016 = read_csv("atp_matches_2016.csv")

# Plotando gráfico
mean.surfaces=tapply(atp_matches_2016$w_2ndWon+atp_matches_2016$l_2ndWon,
                     atp_matches_2016$surface, mean, na.rm=TRUE)
sd.surfaces=tapply(atp_matches_2016$w_2ndWon+atp_matches_2016$l_2ndWon,
                   atp_matches_2016$surface, sd, na.rm=TRUE)

mids = barplot(mean.surfaces,
               main = "Médias de Pontos Ganhos com o 2° Saque",
               col = c("darkolivegreen3", "salmon", "chartreuse3",
                       "dodgerblue2"),
               names.arg = c("Carpet", "Clay","Grass","Hard"),
               ylim = c(0,80),
               ylab = "Número de Pontos",
               xlab = "Superfícies",
               space = 1)
arrows(mids, mean.surfaces-sd.surfaces,
      mids, mean.surfaces+sd.surfaces,
      angle = 90, code=3)
```

7.16. Médias de Pontos Ganhos com o 1º Saque e 2º Saque por Superfície

```
# Ler Dados
atp_matches_2016 = read_csv("atp_matches_2016.csv")

# Plotando gráfico
mean1.surfaces=tapply(atp_matches_2016$w_1stWon+atp_matches_2016$l_1stWon,
                      atp_matches_2016$surface, mean, na.rm=TRUE)
mean2.surfaces=tapply(atp_matches_2016$w_2ndWon+atp_matches_2016$l_2ndWon,
                      atp_matches_2016$surface, mean, na.rm=TRUE)
df <- stack(mean1.surfaces)
df2 <- stack(mean2.surfaces)
df3 <- rbind(df$values,df2$values)
barplot(df3, main="Médias de Pontos Ganhos com o 1° e 2° saque",
        xlab="Superfícies",
       ylab="Número de Pontos",
       legend = rownames(df3),
       col = c("darkblue","red"),
       names = c("Carpet","Clay","Grass","Hard"),
       ylim = c(0, 100),
       beside=TRUE)

legend("topright",  fill = c("darkblue", "red"),
       legend = c("1° saque", "2° saque"),
       cex = 0.52, pt.cex = 50, ncol = 2)
```

7.17. Médias de Pontos Disputados por Superfície

```
# Ler Dados
atp_matches_2016 = read_csv("atp_matches_2016.csv")

# Plotando gráfico
mean.surfaces=tapply(atp_matches_2016$w_svpt+atp_matches_2016$l_svpt,
                     atp_matches_2016$surface, mean, na.rm=TRUE)
sd.surfaces=tapply(atp_matches_2016$w_svpt+atp_matches_2016$l_svpt,
                   atp_matches_2016$surface, sd, na.rm=TRUE)
mids = barplot(mean.surfaces, main = "Médias de Pontos Disputados",
               col = c("darkolivegreen3", "salmon", "chartreuse3",
                       "dodgerblue2"),
               names.arg = c("Carpet", "Clay","Grass","Hard"),
               ylim = c(0,350),
               ylab = "Número de Pontos",
               xlab = "Superfí-cies",
               space = 1)
arrows(mids, mean.surfaces-sd.surfaces,
       mids, mean.surfaces+sd.surfaces,
       angle = 90, code=3)
```

7.18. Tempo de Partida por Superfície

```
library(ggplot2)
attach(atp_matches_2016_1_)
# gera media e desvio padrao do tempo por superficie
bd <- atp_matches_2016_1_
# pega todas as superficies e minutos menos as vazias
vetorSurface <- (unique( na.exclude(bd$surface)))
vetorMinutes <- (unique( na.exclude(bd$minutes)))
# salva em um subset
novaBase <- subset(bd, surface == vetorSurface)
```

```
# imprime o grafico
geraGrafico <- function(banco, titulo, eixoX, eixoY, tipo){
    grafico <- ggplot(banco, aes(x=surface, y=minutes))
    grafico <- grafico + labs(title = titulo, x= eixoX,y= eixoY)
    grafico <- grafico + theme_classic()
  if (tipo == 1)
    grafico <- grafico + geom_boxplot()
  else if (tipo == 2)
    grafico <- grafico + geom_line()
  return (grafico)
}
print(geraGrafico(novaBase,
                  "Tempo de partida por superficie",
                  "Superfície",
                  "Minutos", 1))
baseCarpet <- subset(bd, surface == vetorSurface[4])
agg <- mean(baseCarpet$minutes)
print(geraGrafico(agg, "Média de tempo de partida por superficie",
                  "Superfície","Minutos", 2))
detach(atp_matches_2016_1_)
```