

2) A partir das duas (2) variáveis adotadas para análise:

a) Desenvolva uma breve análise exploratória/descritiva das mesmas

## Problema

Como dito anteriormente, estaremos analisando a relação entre as variáveis “chutes\_man” e “gols\_man” que serão referentes ao quantidade de chutes e gols do time mandante (equipe que joga em casa) respectivamente.

Importanto pacotes utilizados na análise

```
library(ggplot2)
library(car)
library(ggpubr)
library(dplyr)
library(knitr)
library(DataExplorer)
```

Conhecendo nossa base de dados, temos 36 variáveis:

```
dados <- read.csv("brasileirao_serie_a.csv")
colnames(dados)

## [1] "ano_campeonato"      "data"
## [3] "horario"              "rodada"
## [5] "estadio"              "arbitro"
## [7] "publico"              "publico_max"
## [9] "time_man"             "time_vis"
## [11] "tecnico_man"          "tecnico_vis"
## [13] "colocacao_man"        "colocacao_vis"
## [15] "valor_equipe_titular_man" "valor_equipe_titular_vis"
## [17] "idade_media_titular_man" "idade_media_titular_vis"
## [19] "gols_man"              "gols_vis"
## [21] "gols_1_tempo_man"     "gols_1_tempo_vis"
## [23] "escanteios_man"       "escanteios_vis"
## [25] "faltas_man"           "faltas_vis"
## [27] "chutes_bola_parada_man" "chutes_bola_parada_vis"
## [29] "defesas_man"          "defesas_vis"
## [31] "impedimentos_man"     "impedimentos_vis"
## [33] "chutes_man"           "chutes_vis"
## [35] "chutes_fora_man"      "chutes_fora_vis"
```

Selecionando da base de dados apenas as colunas que serão analisadas:

```
dados_filtrados <- dados %>%
  select(time = time_man, chutes = chutes_man, gols = gols_man) %>%
  filter(!is.na(chutes))

kable(head(dados_filtrados), caption="Primeiras linhas da base de dados")
```

Table 1: Primeiras linhas da base de dados

time	chutes	gols
Santos FC	3	1
CearÃ SC	5	0
AtlÃtico-PR	4	1
CearÃ SC	4	0
EC Bahia	6	1
AtlÃtico-MG	4	3

## Visualizar a relaão entre chutes e gols de cada time mandante

Média de gols por time:

```
media_de_gols_por_time <- dados_filtrados %>%
  group_by(time) %>%
  summarise_at(vars(gols),
    list(media_gols=mean))
knitr::kable(media_de_gols_por_time, digits=2, caption="Média de gols por time")
```

Table 2: Média de gols por time

time	media_gols
AmÃrica-MG	0.90
Athletico-PR	1.15
AtlÃtico-GO	1.26
AtlÃtico-MG	1.84
AtlÃtico-PR	2.42
AvaÃ FC	0.25
Botafogo	0.95
CearÃ SC	1.16
Chapecoense	1.15
Corinthians	1.29
Coritiba FC	0.68
Cruzeiro	1.00
CSA	0.50
EC Bahia	1.30
EC VitÃria	1.00
Flamengo	1.97
Fluminense	1.30
Fortaleza	1.28
GoiÃs EC	1.19
GrÃmio	1.89
Internacional	1.74
Palmeiras	2.00
ParanÃ	0.70
RB Bragantino	1.79
SÃo Paulo	1.28
Santos FC	1.74
Sport Recife	0.97

time	media_gols
Vasco da Gama	1.19

O time Atlético-PR tem a maior média de gols.

Média de chutes por time:

```
media_de_chutes_por_time <- dados_filtrados %>%
  group_by(time) %>%
  summarise_at(vars(chutes),
    list(media_chutes=mean))
knitr::kable(media_de_chutes_por_time, digits=2, caption="Média de chutes por time")
```

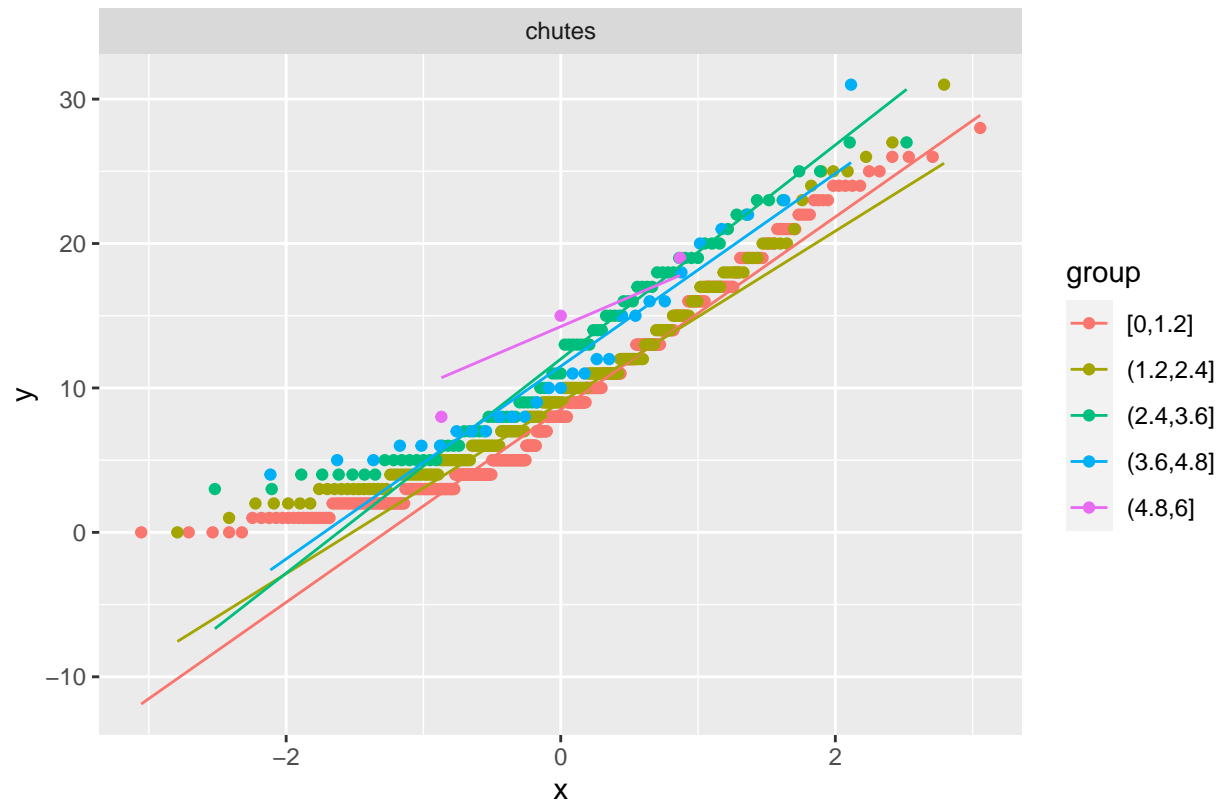
Table 3: Média de chutes por time

time	media_chutes
América-MG	3.00
Athletico-PR	10.19
Atlético-GO	12.47
Atlético-MG	10.92
Atlético-PR	6.42
Avaí FC	8.00
Botafogo	10.18
Ceará SC	10.41
Chapecoense	5.75
Corinthians	9.97
Coritiba FC	10.63
Cruzeiro	7.20
CSA	6.50
EC Bahia	9.54
EC Vitória	3.64
Flamengo	12.43
Fluminense	8.95
Fortaleza	10.24
Goiás EC	10.74
Grêmio	11.43
Internacional	9.56
Palmeiras	11.47
Paraná	3.00
RB Bragantino	16.16
São Paulo	10.67
Santos FC	10.33
Sport Recife	8.14
Vasco da Gama	8.75

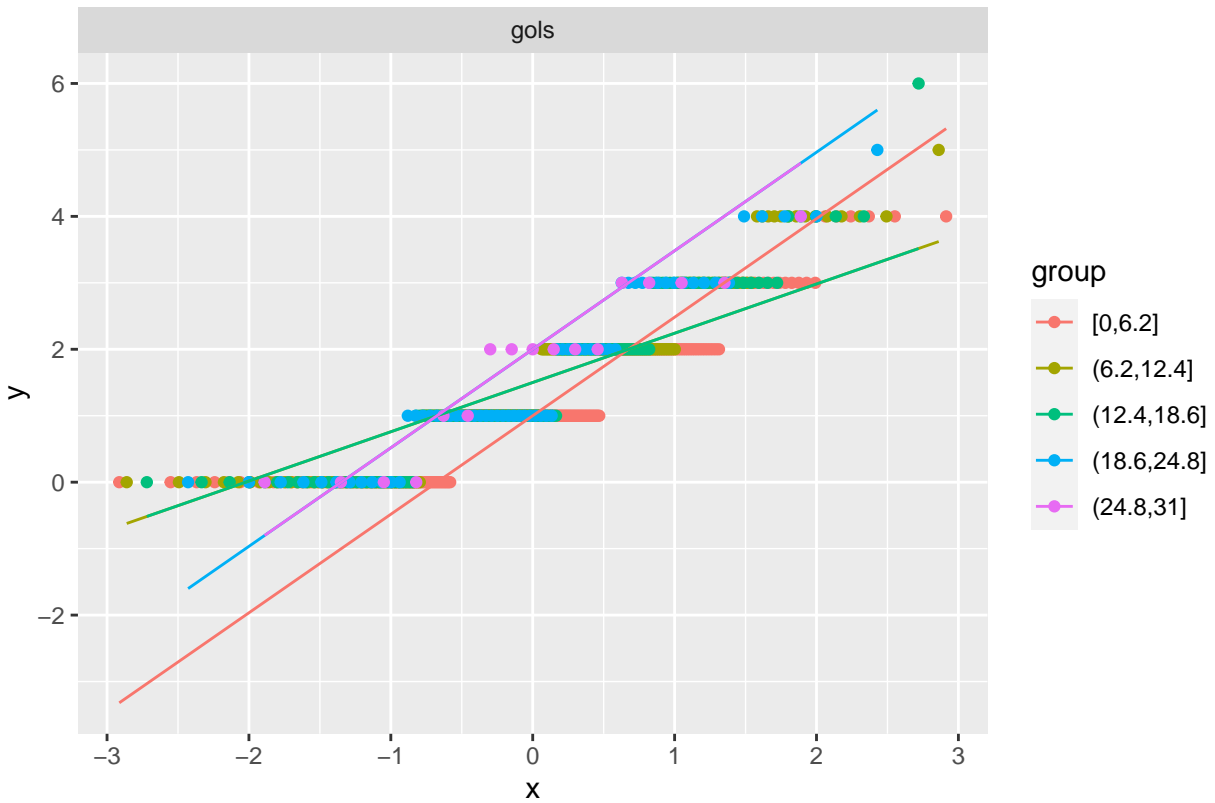
O time RB Bragantino tem a maior média de chutes

Plot quantil-quantil pra verificarmos a validade de cada distribuição das variáveis:

```
plot_qq(dados_filtrados, by="goals")
```

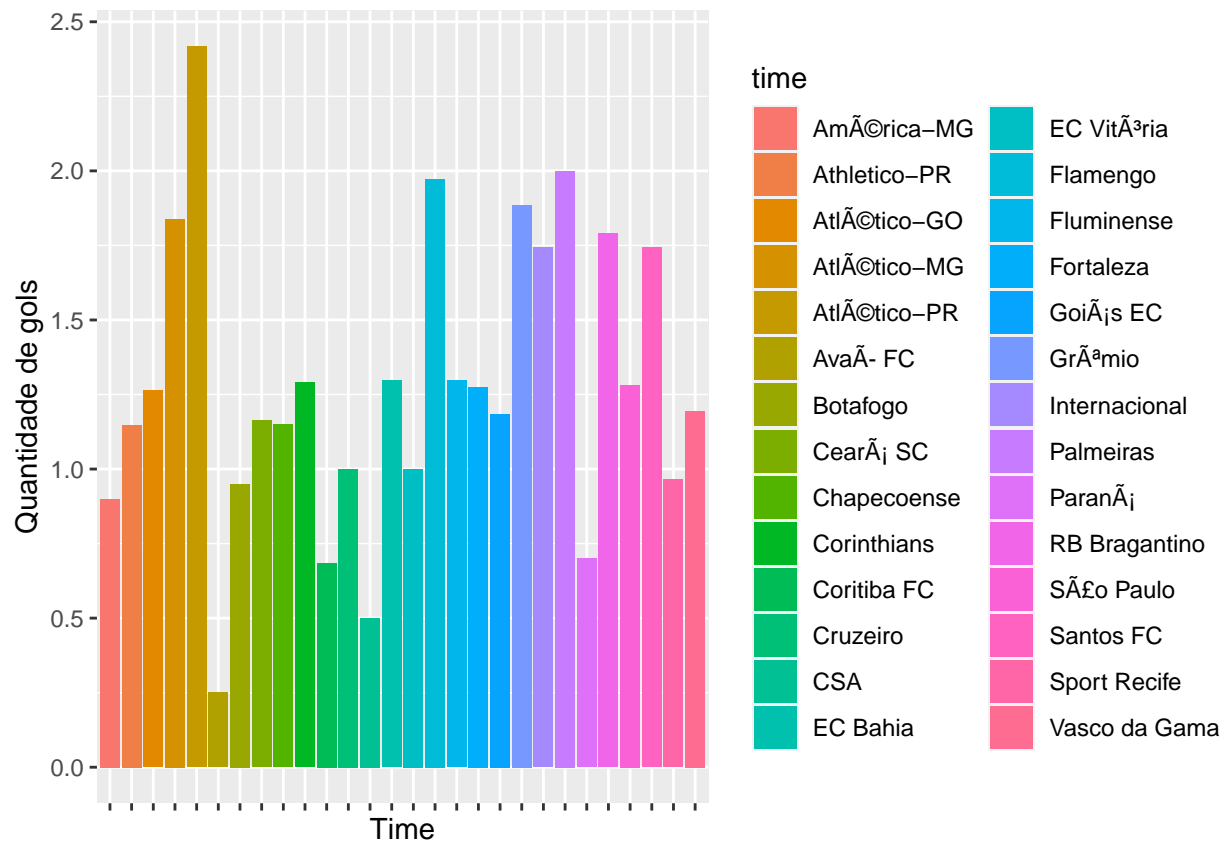


```
plot_qq(dados_filtrados, by="chutes")
```



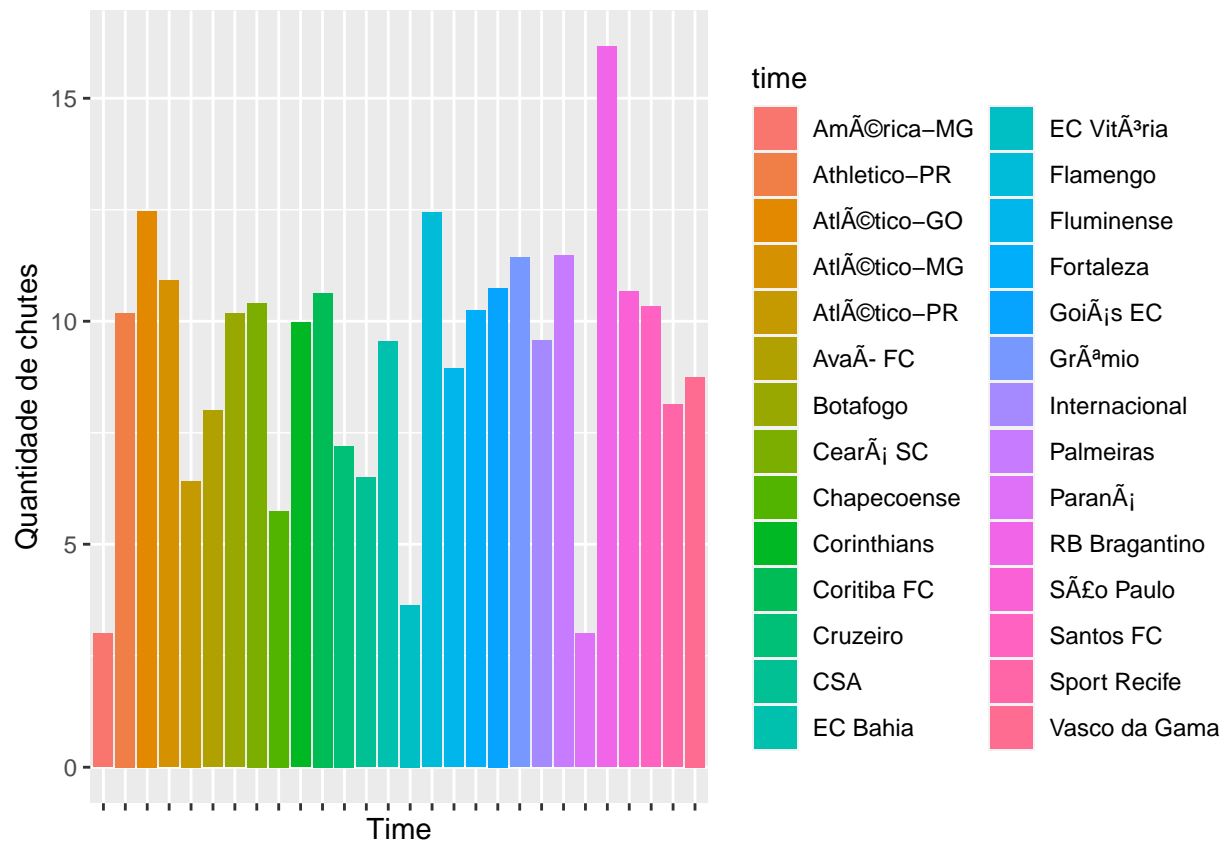
Visualizando por meio do gráfico de barra o número de gols de cada time

```
ggplot(as.data.frame(media_de_gols_por_time),
  aes(x=time, y=media_gols, fill = time))+
geom_col(position="dodge")+labs(x="Time", y="Quantidade de gols")+
theme(axis.text.x = element_blank())
```



Visualizando por meio do gráfico de barra o número de chutes de cada time

```
ggplot(as.data.frame(media_de_chutes_por_time), aes(x=time, y=media_chutes, fill = time))+
  geom_col(position="dodge")+
  labs(x="Time", y="Quantidade de chutes")+
  theme(axis.text.x = element_blank())
```



b) Desenvolva e **interprete de forma prática** uma análise de correlação.

```
#Calculando coeficiente de correlação de pearson
cor(dados_filtrados$chutes,dados_filtrados$gols)
```

```
## [1] 0.1723087
```

```
#Teste estatístico do grau de correlação
cor.test(dados_filtrados$chutes,dados_filtrados$gols)
```

```
##
## Pearson's product-moment correlation
##
## data: dados_filtrados$chutes and dados_filtrados$gols
## t = 4.7937, df = 751, p-value = 1.974e-06
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
## 0.1021200 0.2407902
## sample estimates:
## cor
## 0.1723087
```

```
#Outros tipos de correlação
```

```
cor(dados_filtrados$chutes,dados_filtrados$gols, method="kendall")
```

```
## [1] 0.1412668
```

```
cor(dados_filtrados$chutes,dados_filtrados$gols, method="spearman")
```

```
## [1] 0.1814503
```

- c) Desenvolva e **interprete de forma prática** uma análise de regressão linear simples, incluindo a análise de resíduos e previsões para alguns valores estabelecidos para a variável independente,  $X = x$ .

```
# Modelo de Regressao Linear Simples (MRLS)
```

```
mod <- lm(chutes ~ gols, data = dados_filtrados)
```

```
# Coeficientes Estimados
```

```
mod
```

```
##
```

```
## Call:
```

```
## lm(formula = chutes ~ gols, data = dados_filtrados)
```

```
##
```

```
## Coefficients:
```

```
## (Intercept)      gols  
##      8.511      0.977
```

```
# Inferências
```

```
summary(mod)
```

```
##
```

```
## Call:
```

```
## lm(formula = chutes ~ gols, data = dados_filtrados)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max  
## -10.465  -5.465  -1.442   4.489  20.535
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)  
## (Intercept)   8.5111     0.3598  23.655 < 2e-16 ***  
## gols          0.9770     0.2038   4.794 1.97e-06 ***
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## Residual standard error: 6.178 on 751 degrees of freedom
```

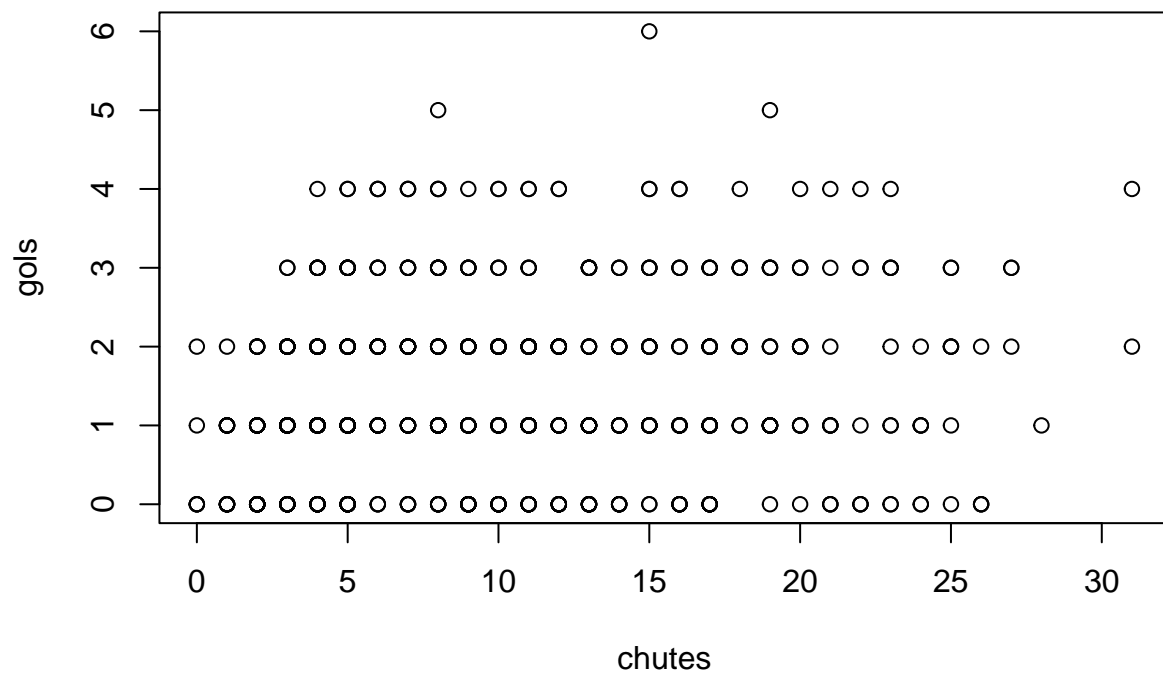
```
## Multiple R-squared:  0.02969,    Adjusted R-squared:  0.0284
```

```
## F-statistic: 22.98 on 1 and 751 DF,  p-value: 1.974e-06
```

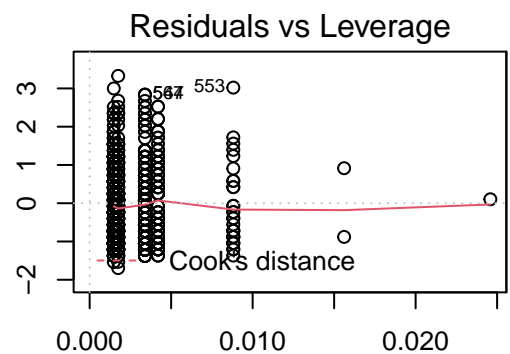
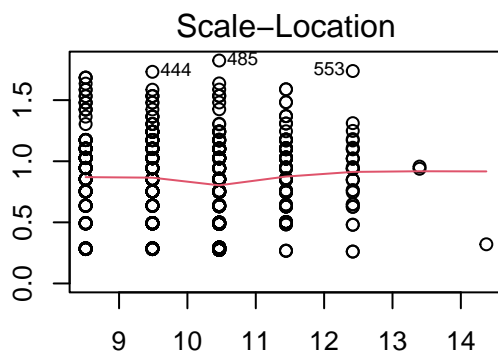
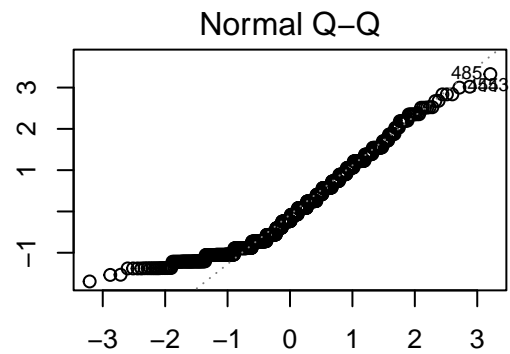
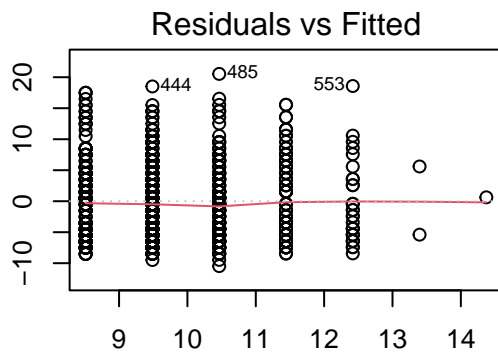
```
#Gráfico de relação
```

```
plot(dados_filtrados$chutes, dados_filtrados$gols, xlab="chutes", ylab="gols")
```





```
## Analise grafica baseada no modelo estimado
par(mfrow=c(2,2), mar=c(3,3,3,3))
plot(mod)
```



```
# Normalidade dos residuos:
shapiro.test(mod$residuals)
```

```
##
## Shapiro-Wilk normality test
##
## data:  mod$residuals
## W = 0.93874, p-value < 2.2e-16
```

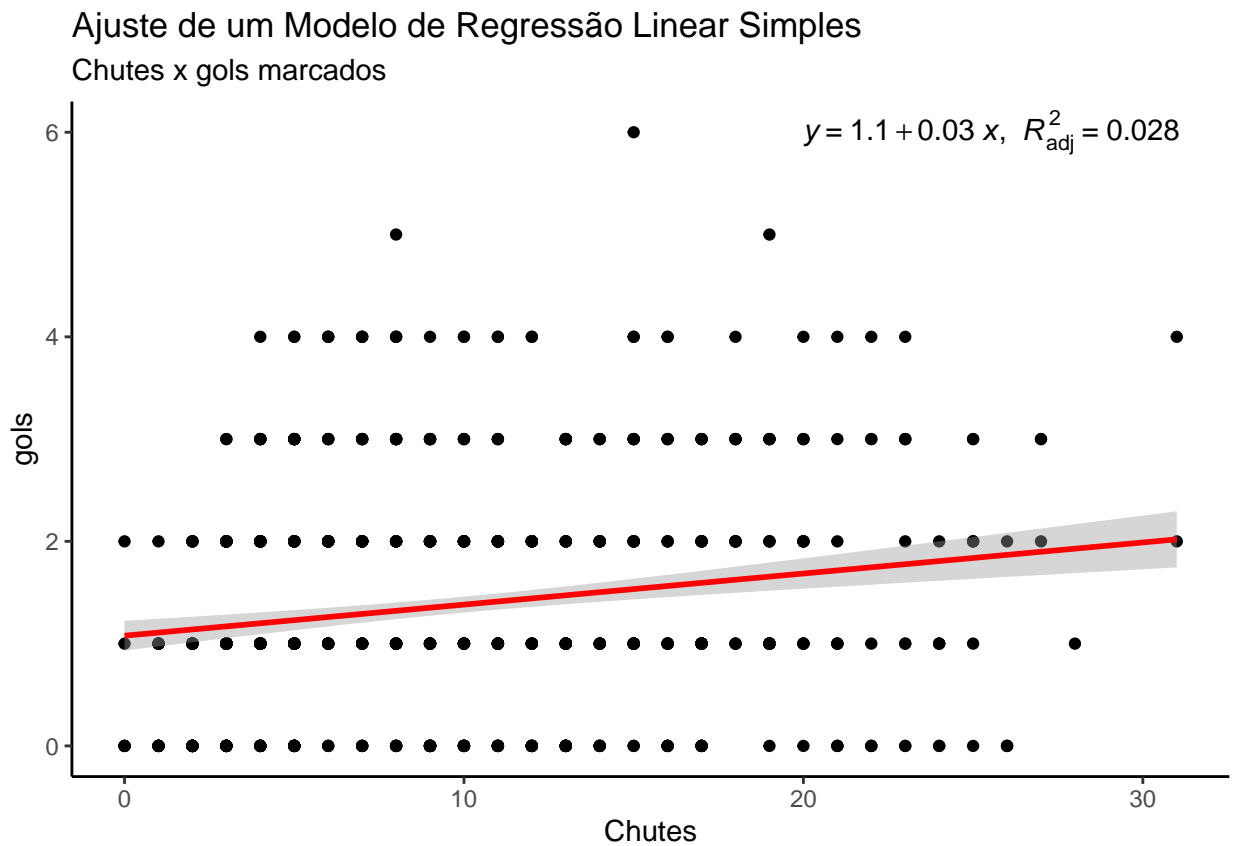
```
# Outliers nos residuos:
summary(rstandard(mod))
```

```
##      Min.    1st Qu.    Median      Mean   3rd Qu.      Max.
## -1.695425 -0.885386 -0.233911  0.000022  0.727833  3.326820
```

```
# Independencia dos residuos
durbinWatsonTest(mod)
```

```
## lag Autocorrelation D-W Statistic p-value
## 1      0.5278336      0.9413638      0
## Alternative hypothesis: rho != 0
```

```
dados_filtrados %>% ggplot(aes(x = chutes, y = gols)) +
  theme_classic() +
  geom_point() +
  geom_smooth(method = "lm", col = "red") +
  stat_regline_equation(aes(label = paste(..eq.label.., ..adj.rr.label..,
                                          sep = "*plain(\",\")~~")),
                      label.x = 20, label.y = 6) +
  labs(x='Chutes', y='gols',
       title='Ajuste de um Modelo de Regressão Linear Simples',
       subtitle = 'Chutes x gols marcados')
```



```
# Prevendo a quantidade de chutes para um time com 5 gols
df_chutes <- data.frame(gols = c(5))
predict(mod, df_chutes)
```

```
##          1
## 13.39601
```