Untitled

arroz+

11 de Janeiro de 2016

Grupo B2

Ana João Rodrigues Fonseca 2013170494

Gonçalo Daniel Tavares Duarte 2013155376

Ricardo Jorge Ferreira Margarido 2013145676

Contexto:

Um investidor da bolsa esporádico teve conhecimento de métodos estatísticos de apoio à decisão. Reuniu uma série de parâmetros que achava pertinentes e que geralmente usava para prever pontos de compra e venda de acções. Os dados foram organizados numa base de dados tendo sido adicionada uma variável dicotómica de compra (0) e venda (1). O investidor procurou posteriormente uma empresa de análise de dados com o intuito de implementar um método de previsão da compra/venda.

Variáveis:

As variáveis são genéricas não sendo facultada nenhuma descrição das mesmas conhecendose apenas o seu nível de mensuração (V1 e V2 - nominal; V3 e V4 - ordinal; V5 e V6 - quantitativa).

A última variável (R) na base de dados corresponde aos rótulos para os quais se pretende implementar o modelo estatístico de classificação. A base de dados é simulada.

```
library(knitr)
library(pscl)
```

```
## Warning: package 'pscl' was built under R version 3.2.3
```

```
## Loading required package: MASS
## Loading required package: lattice
## Classes and Methods for R developed in the
##
## Political Science Computational Laboratory
##
## Department of Political Science
##
## Stanford University
##
## Simon Jackman
##
## hurdle and zeroinfl functions by Achim Zeileis
```

```r
library(survey)
```

```
## Loading required package: grid
##
## Attaching package: 'survey'
##
## The following object is masked from 'package:graphics':
##
##     dotchart
```

```r
library(ResourceSelection)
```

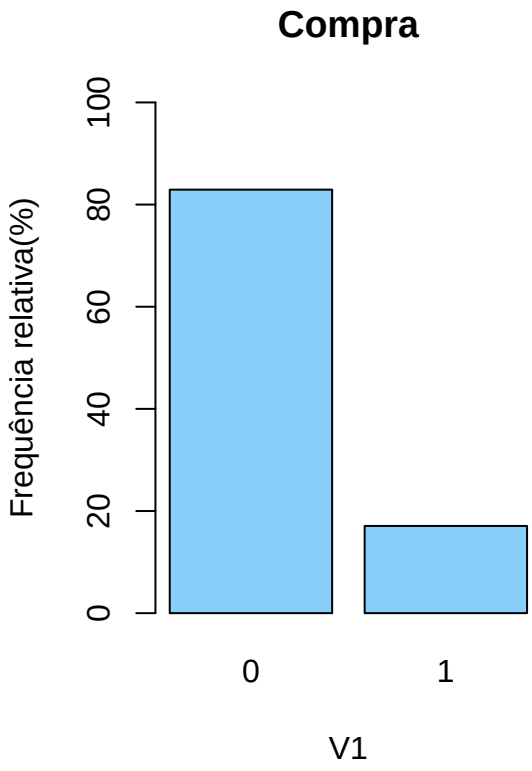
```
## ResourceSelection 0.2-5   2015-11-06
```

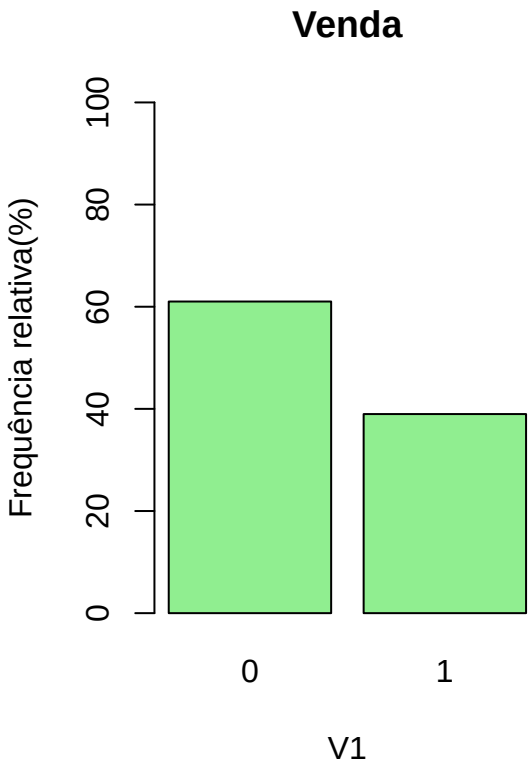
```r
library(car)
library(e1071)
library(rpart)
setwd("C:\\Users\\Gonçalo Duarte\\Desktop")
data<-read.csv2("grupo_B2.csv",sep=",", dec=".")
data=data[complete.cases(data),]
dataR=data$R
```

```r
compra <- data[data$R == '0',]
venda <- data[data$R == '1',]
```

V1

```r
dataR[dataR==0] <- 'Compra'
dataR[dataR==1] <- 'Venda'
par(mfrow=c(1,2))
V1c <- table(compra$V1)/nrow(compra)*100
V1v <- table(venda$V1)/nrow(venda)*100
V1compra <- barplot(V1c, col='lightskyblue', main='Compra', xlab='V1', ylab='Frequência relativa(%)', yl
V1venda <- barplot(V1v, col='lightgreen', main='Venda', xlab='V1', ylab='Frequência relativa(%)', ylim=
```






```
tablet1 <- table(data$R, data$V1)

chisq.test(tablet1)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  tablet1
## X-squared = 4.5355, df = 1, p-value = 0.0332
```

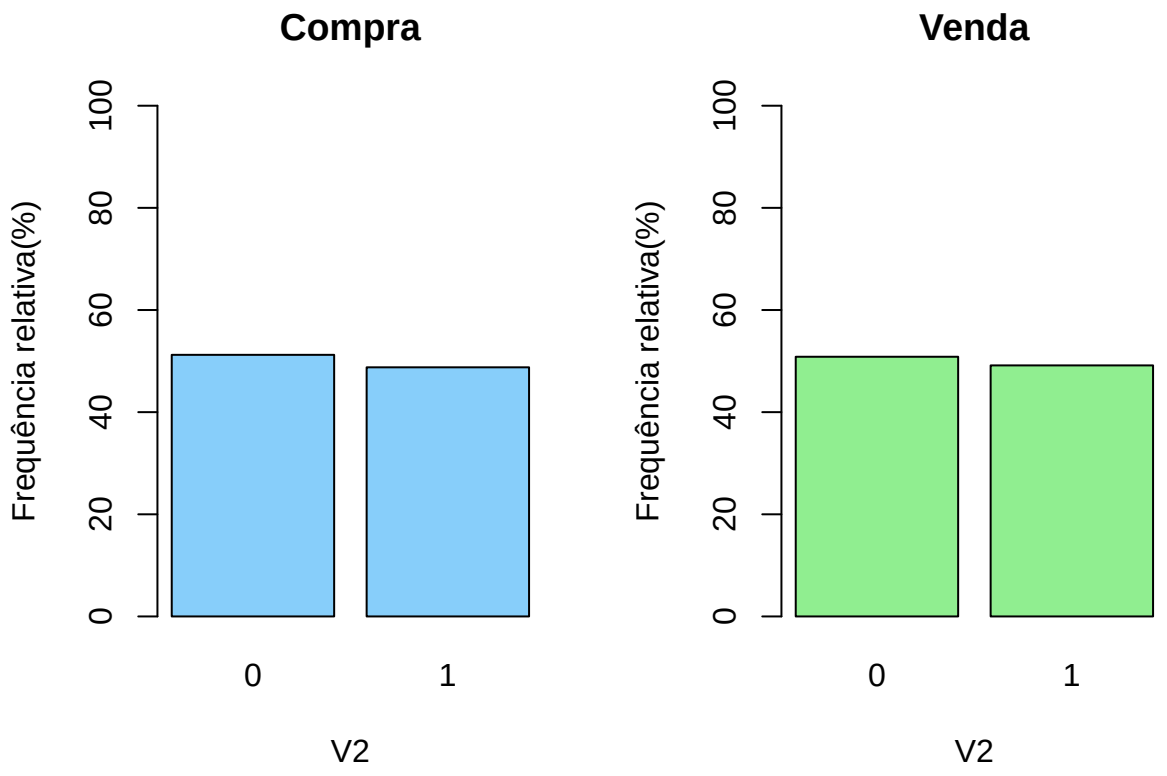
```
kable(tablet1)
```

	0	1
0	34	7
1	36	23

V1 p menor que alpha por isso rejeita-se a hipótese nula logo as variáveis não são independentes havendo por isso relação entre ambas.

V2

```
par(mfrow=c(1,2))
dataR[dataR==0] <- 'Compra'
dataR[dataR==1] <- 'Venda'
V2c <- table(compra$V2)/nrow(compra)*100
V2v <- table(venda$V2)/nrow(venda)*100
V2compra <- barplot(V2c, col='lightskyblue', main='Compra', xlab='V2', ylab='Frequência relativa(%)', yl
V2venda <- barplot(V2v, col='lightgreen', main='Venda', xlab='V2', ylab='Frequência relativa(%)', ylim=c
```




```
tablet2 <- table(data$V2, dataR)
kable(tablet2)
```

	Compra	Venda
0	21	30
1	20	29

```
chisq.test(tablet2)
```

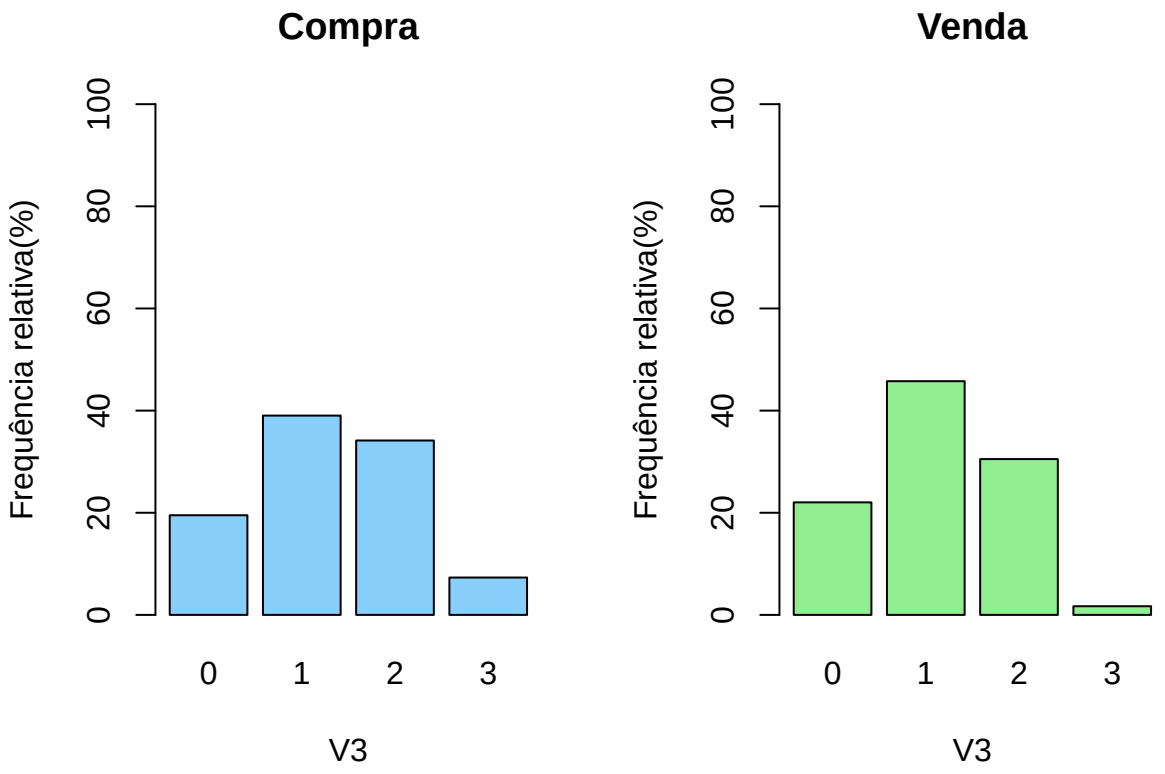
```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
```

```
## data:  tablet2
## X-squared = 0, df = 1, p-value = 1
```

V2 p maior que alpha, nao se rejeita a hipotese nula, as variaveis sao independentes e nao existe relacao entre as mesmas.

V3

```
par(mfrow=c(1,2))
dataR[dataR==0] <- 'Compra'
dataR[dataR==1] <- 'Venda'
V3c <- table(compra$V3)/nrow(compra)*100
V3v <- table(venda$V3)/nrow(venda)*100
V1compra <- barplot(V3c, col='lightskyblue', main='Compra', xlab='V3', ylab='Frequência relativa(%)', yl
V1venda <- barplot(V3v, col='lightgreen', main='Venda', xlab='V3', ylab='Frequência relativa(%)', ylim=
```




```
tablet3 <- table(data$V3, dataR)
kable(tablet3)
```

	Compra	Venda
0	8	13
1	16	27

	Compra	Venda
2	14	18
3	3	1

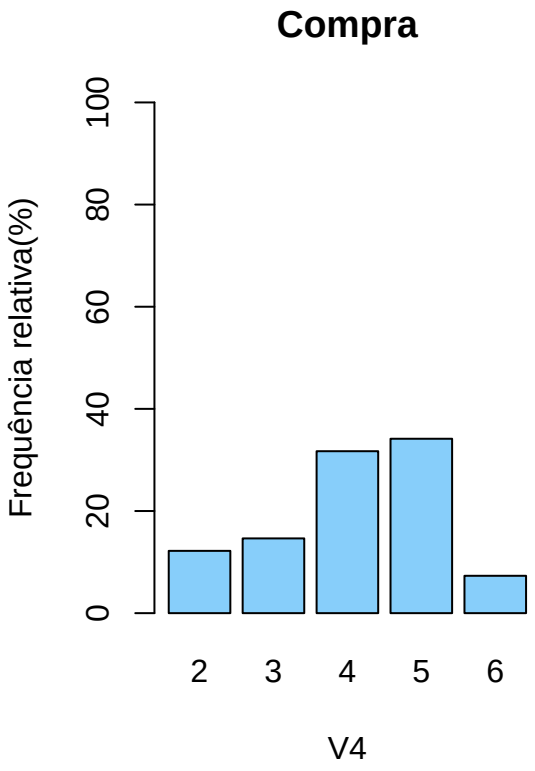
```
wilcox.test(V3 ~ R, data=data, paired=FALSE) #PORQUE?
```

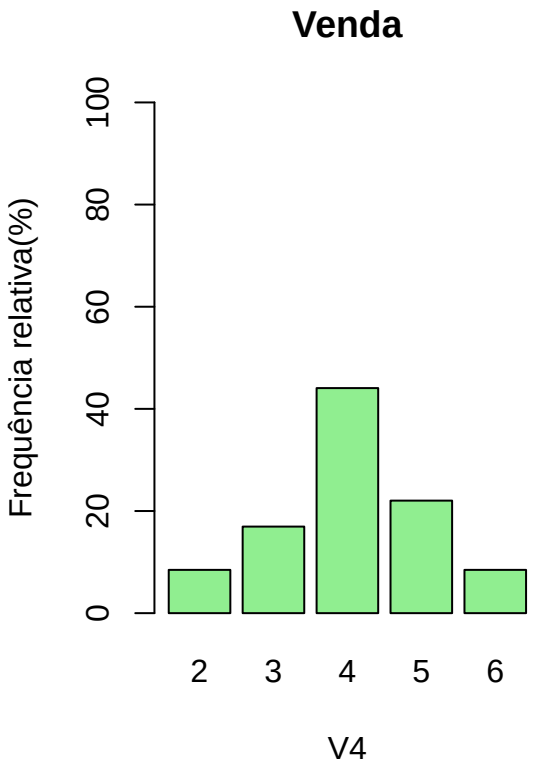
```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  V3 by R
## W = 1337.5, p-value = 0.3404
## alternative hypothesis: true location shift is not equal to 0
```

Verifica-se que o valor de p é maior que alpha não se podendo rejeitar a hipótese nula (a diferença das medianas ser 0) concluindo assim que não existe relação entre as variáveis em estudo.

V4

```
par(mfrow=c(1,2))
dataR[dataR==0] <- 'Compra'
dataR[dataR==1] <- 'Venda'
V4c <- table(compra$V4)/nrow(compra)*100
V4v <- table(venda$V4)/nrow(venda)*100
V1compra <- barplot(V4c, col='lightskyblue', main='Compra', xlab='V4', ylab='Frequência relativa(%)', y
V1venda <- barplot(V4v, col='lightgreen', main='Venda', xlab='V4', ylab='Frequência relativa(%)', ylim=
```






```
tablet4 <- table(data$V4, dataR)
kable(tablet4)
```

	Compra	Venda
2	5	5
3	6	10
4	13	26
5	14	13
6	3	5

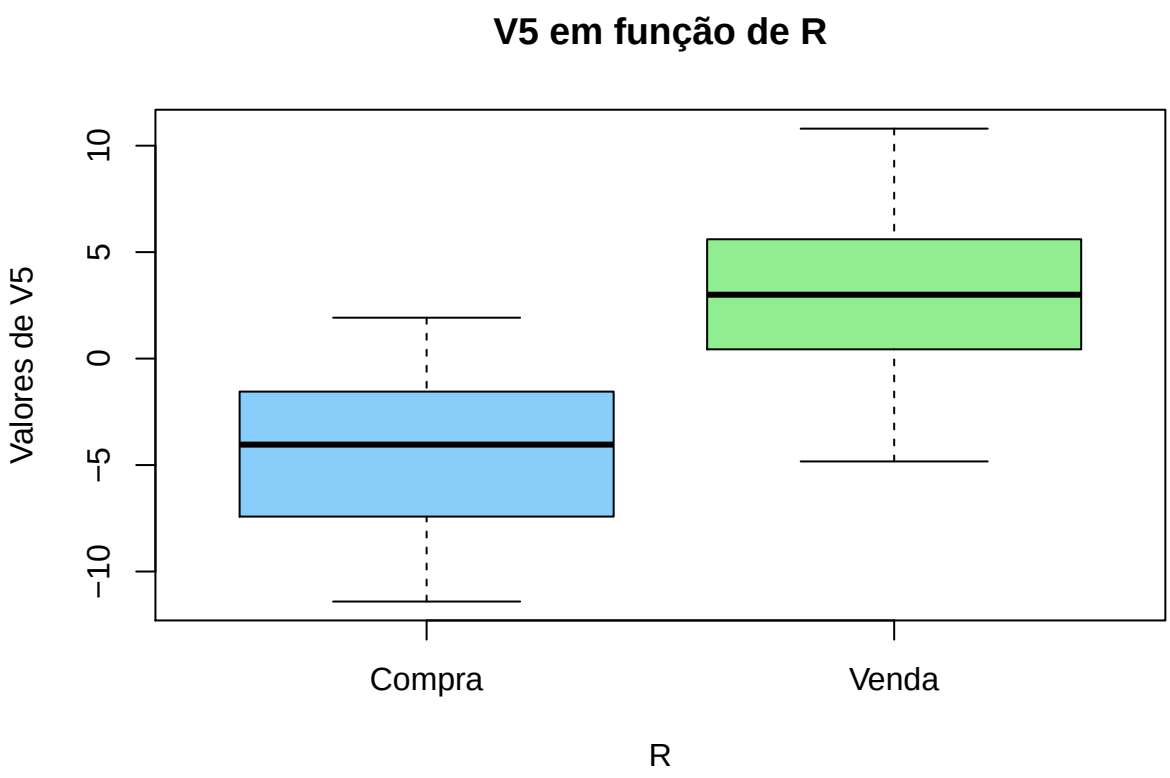
```
wilcox.test(V4 ~ R, data=data, paired=FALSE) #PORQUE?
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  V4 by R
## W = 1271, p-value = 0.655
## alternative hypothesis: true location shift is not equal to 0
```

Verifica-se que o valor de p é maior que alpha não se podendo rejeitar a hipótese nula (a diferença das medianas ser 0) concluindo assim que não existe relação entre as variáveis em estudo.

V5

```
boxplot (data$V5 ~ data$R, main='V5 em função de R', xlab='R', names=c('Compra', 'Venda'), ylab='Valores
```




```
V50 <- data$V5[data$R == 0]
V51 <- data$V5[data$R == 1]
shapiro.test(V50)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  V50
## W = 0.96982, p-value = 0.3401
```

```
shapiro.test(V51)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  V51
## W = 0.98285, p-value = 0.5715
```

Como p é maior que alpha em ambas as amostras não podemos rejeitar a hipótese nula, logo as amostras seguem uma distribuição normal.

```r
leveneTest(V5~factor(R),data=data)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.0662 0.7975
##       98
```

Uma vez que o valor de p é maior que alpha não se rejeita a hipótese nula, logo as variâncias são iguais.

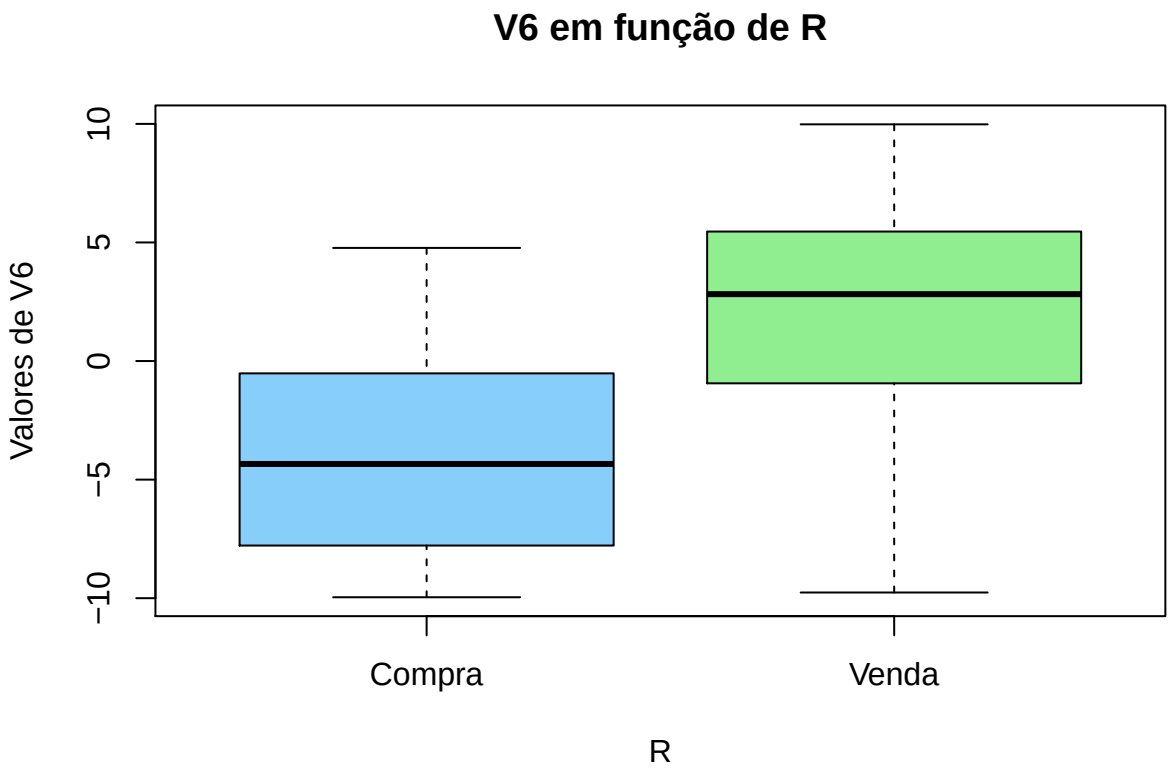
```r
t.test(V5~R, data = data, var.equal = TRUE, paired = FALSE)
```

```
##
##  Two Sample t-test
##
## data:  V5 by R
## t = -9.2698, df = 98, p-value = 4.655e-15
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -8.613180 -5.575642
## sample estimates:
## mean in group 0 mean in group 1
##       -4.263902        2.830508
```

Verifica-se que p é menor que alpha, logo rejeita-se a hipótese nula (as médias das amostras serem iguais), concluindo assim que existe relação entre V5 e R.

V6

```r
boxplot (data$V6 ~ data$R, main='V6 em função de R', xlab='R', names=c('Compra', 'Venda'), ylab='Valores
```




```
V60 <- data$V6[data$R == 0]
V61 <- data$V6[data$R == 1]
shapiro.test(V60)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  V60
## W = 0.95743, p-value = 0.1278
```

```
shapiro.test(V61)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  V61
## W = 0.95317, p-value = 0.02367
```

Uma vez que um dos valores de p é menor que alpha rejeita-se a hipótese nula logo não existe uma distribuição normal das variáveis. Uma vez que as amostras são independentes recorre-se a um teste de Wilcox.

```
wilcox.test(V6 ~ R, data=data, paired=FALSE)
```

```
##
```

```
##  Wilcoxon rank sum test with continuity correction
##
## data:  V6 by R
## W = 467, p-value = 1.991e-07
## alternative hypothesis: true location shift is not equal to 0
```

Verifica-se que o valor de p é menor que alpha rejeitando-se a hipótese nula (a diferença das medianas ser 0) concluindo assim que existe relação entre as variáveis em estudo.

Modelo Logístico

```
options(warn=-1)
ind=1:nrow(data)
testind=sample(ind,trunc(length(ind)*0.7))
trainSet=data[testind,]
testSet=data[-testind,]

logit2=glm(R~V1+V5+V6, data=trainSet, family=binomial(link=logit))
summary(logit2)
```

```
##
## Call:
## glm(formula = R ~ V1 + V5 + V6, family = binomial(link = logit),
##     data = trainSet)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.92056  -0.01280   0.00000   0.01186   1.47614
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   3.6990     1.7946   2.061   0.0393 *
## V1            5.7125    12.1319   0.471   0.6377
## V5            2.4854     1.1558   2.150   0.0315 *
## V6            1.3373     0.5919   2.260   0.0238 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 94.2216  on 69  degrees of freedom
## Residual deviance:  8.7662  on 66  degrees of freedom
## AIC: 16.766
##
## Number of Fisher Scoring iterations: 11
```

```
confint(logit2)
```

```
## Waiting for profiling to be done...
```

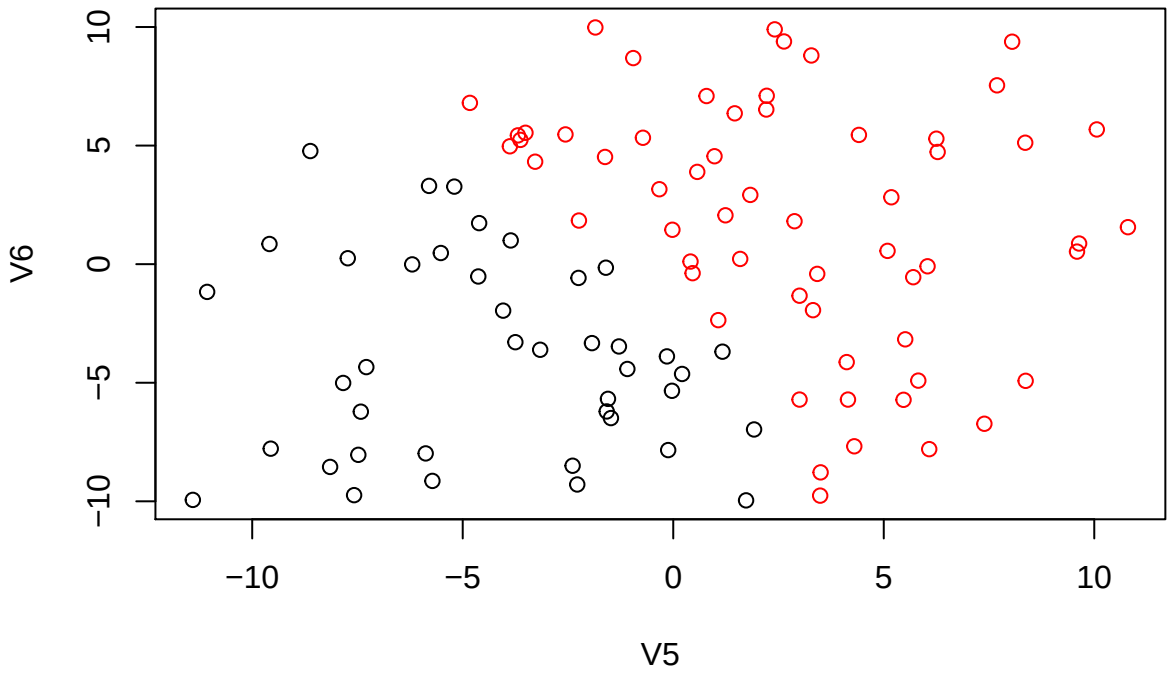
```
##                   2.5 %    97.5 %
## (Intercept)  1.2528554  9.498783
## V1          -2.3959223 28.422409
## V5           1.1015112  6.529871
## V6           0.5938013  3.351902
```

```
options(warn=-1)
hoslem.test(logit2$y,fitted(logit2),g=10)
```

```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  logit2$y, fitted(logit2)
## X-squared = 0.11461, df = 8, p-value = 1
```

Concluimos então tanto pelo valor de p como pelo facto de 0 estar ou não presente no intervalo de confiança que as variáveis a considerar para a elaboração do nosso modelo devem ser apenas V5 e V6.

```
options(warn=-1)
plot(data$V5,data$V6,col=as.factor(data$R),xlab="V5",ylab="V6")
```




```
ind=1:nrow(data)
testind=sample(ind,trunc(length(ind)*0.7))
trainSet=data[testind,]
```

```
testSet=data[-testind,]

logit1=glm(R~V5+V6, data=trainSet, family=binomial(link=logit))
summary(logit1)
```

```
##
## Call:
## glm(formula = R ~ V5 + V6, family = binomial(link = logit), data = trainSet)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.88809  -0.00451   0.00000   0.00587   1.86169
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   4.1609     1.9607   2.122   0.0338 *
## V5            2.7285     1.2627   2.161   0.0307 *
## V6            1.5596     0.6521   2.392   0.0168 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 94.222  on 69  degrees of freedom
## Residual deviance: 10.866  on 67  degrees of freedom
## AIC: 16.866
##
## Number of Fisher Scoring iterations: 10
```

```
hoslem.test(logit1$y,fitted(logit1),g=10)
```

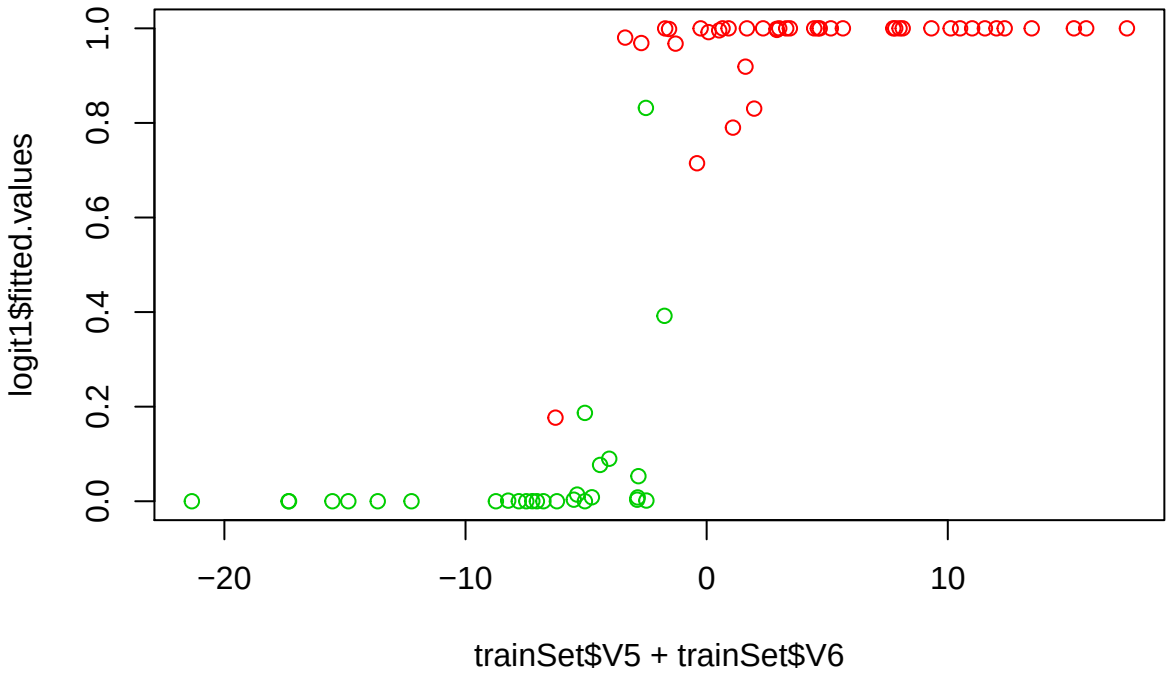
```
##
##  Hosmer and Lemeshow goodness of fit (GOF) test
##
## data:  logit1$y, fitted(logit1)
## X-squared = 0.065075, df = 8, p-value = 1
```

```
confint(logit1)
```

```
## Waiting for profiling to be done...
```

```
##                 2.5 %    97.5 %
## (Intercept) 1.5475065 10.488445
## V5          1.1732322  6.954447
## V6          0.6994388  3.670232
```

```
Color = trainSet$R == '0'
plot(trainSet$V5+trainSet$V6,logit1$fitted.values, col = Color + 2)
```




```
logit1_null <- glm(R ~ 1, data=trainSet, family=binomial)
LL_null <- logLik(logit1_null)
LL_k <- logLik(logit1)
R_Cox <- 1 - (exp(LL_null[1])/exp(LL_k[1]))^(2/length(trainSet))
R_Nag <- R_Cox/(1-(exp(LL_null[1]))^(2/length(trainSet)))
print(sprintf('R2 Cox = %s',R_Cox))
```

```
## [1] "R2 Cox = 0.999993263106366"
```

```
print(sprintf('R2 Naguelkerke = %s',R_Nag))
```

```
## [1] "R2 Naguelkerke = 0.999994689674862"
```

```
prob = predict(logit1,type = c('response'),trainSet)
confusion<-table(prob>0.3,trainSet$R)
rownames(confusion)[1] <- "Compra"
rownames(confusion)[2] <- "Venda"
kable(confusion, caption='Avaliação Treino', col.names=c('Compra','Venda'))
```

Table 5: Avaliação Treino

	Compra	Venda
Compra	26	1
Venda	2	41

```
exatidao1=(confusion[1,1]+confusion[2,2])/70
exatidao1
```

```
## [1] 0.9571429
```

```
logit1_null1 <- glm(R ~ 1, data=testSet, family=binomial)
LL_null1 <- logLik(logit1_null1)
LL_k1 <- logLik(logit1)
R_Cox1 <- 1 - (exp(LL_null1[1])/exp(LL_k1[1]))^(2/length(testSet))
R_Nag1 <- R_Cox1/(1-(exp(LL_null1[1]))^(2/length(testSet)))
print(sprintf('R2 Cox = %s',R_Cox1))
```

```
## [1] "R2 Cox = 0.986600286936681"
```

```
print(sprintf('R2 Naguelkerke = %s',R_Nag1))
```

```
## [1] "R2 Naguelkerke = 0.989407698138627"
```

```
prob = predict(logit1,type = c('response'),testSet)
confusion2<-table(prob>0.3,testSet$R)
rownames(confusion2)[1] <- "Compra"
rownames(confusion2)[2] <- "Venda"
kable(confusion2, col.names=c('Compra', 'Venda'),caption='Avaliação Teste')
```

Table 6: Avaliação Teste

	Compra	Venda
Compra	13	0
Venda	0	17

```
exatidao=(confusion2[1,1]+confusion2[2,2])/30
exatidao
```

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

```
sensibilidade=(confusion2[1,1]/(confusion2[1,1]+confusion2[2,1]))
sensibilidade
```

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

```
especificidade=(confusion2[2,2]/(confusion2[1,2]+confusion2[2,2]))
especificidade
```

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

```
svm1=svm(R ~ V5+V6 , data= trainSet, cost=100 , gamma=0.01 )
predsvm1= predict(svm1, newdata= trainSet, type=c('response'))
tablesvm1=table(predsvm1>0.5 , trainSet$R)
rownames(tablesvm1)[1] <- "Compra"
rownames(tablesvm1)[2] <- "Venda"
kable(tablesvm1, caption='Avaliação Treino', col.names=c('Compra', 'Venda'))
```

Table 7: Avaliação Treino

	Compra	Venda
Compra	15	0
Venda	13	42

```
exatidao3=(tablesvm1[1,1]+tablesvm1[2,2])/70
exatidao3
```

```
## [1] 0.8142857
```

```
predsvm= predict(svm1, newdata=testSet,type=c('response'))
tablesvm=table(predsvm>0.5 , testSet$R)
rownames(tablesvm)[1] <- "Compra"
rownames(tablesvm)[2] <- "Venda"
kable(tablesvm, caption='Avaliação Teste', col.names=c('Compra', 'Venda'))
```

Table 8: Avaliação Teste

	Compra	Venda
Compra	11	0
Venda	2	17

```
exatidao2=(tablesvm[1,1]+tablesvm[2,2])/30
exatidao2
```

```
## [1] 0.9333333
```

```
sensibilidade1=(tablesvm[1,1]/(tablesvm[1,1]+tablesvm[2,1]))
sensibilidade1
```

```
## [1] 0.8461538
```

```
especificidade1=(tablesvm[2,2]/(tablesvm[1,2]+tablesvm[2,2]))
especificidade1
```

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

```
plot(trainSet$V5+trainSet$V6, svm1$fitted.values, col = Color + 2)
```