m565A4

Neeraj Namani

2023-10-04

Model Selection and ANOVA

1. (20) In this exercise, we will generate simulated data, and will then use this data to perform best subset selection.

(a) Use the rnorm() function to generate a predictor X of length $n = 100$, as well as a noise vector ϵ of length $n = 100$.

```
n <- 100
X<-rnorm(n, mean = 0, sd = 1)
e <- rnorm(n, mean = 0, sd = 1)
head(X)
```

```
## [1]  1.42486421 -0.32712554  1.20815516  0.42840910 -0.04305384  0.25643704
```

```
head(e)
```

```
## [1]  0.5250801 -1.2040714  0.4931406  2.2896245 -1.1747705  0.7421085
```

(b) Generate a response vector Y of length $n = 100$ according to the model $Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$, where $\beta_0, \beta_1, \beta_2$, and β_3 are constants of your choice.

```
beta_0 <- 2
beta_1 <- 1.5
beta_2 <- -0.5
beta_3 <- 0.2
Y <- beta_0 + beta_1 * X + beta_2 * X^2 + beta_3 * X^3 + e
head(Y)
```

```
## [1] 4.2258201 0.2447335 3.9282479 4.8561966 0.7597059 3.0972567
```

(c) Use the regsubsets() function to perform best subset selection in order to choose the best model containing the predictors $X, X^2, \ldots, X^{10}$.

Note that you will need to use the data.frame() function to create a single data set containing both X and all the powers in consideration, and Y. Example: for X^3 use, in the data.frame, $X3 = X^3$.

Use regsubsets() with nbest $=$ 3,really.big $=$ T, nvmax $=$ 10

```
mydata <- data.frame(X = X, X2 = X^2, X3 = X^3, X4 = X^4, X5 = X^5, X6 = X^6,
X7 = X^7, X8 = X^8, X9 = X^9, X10 = X^10, Y = Y)
library(leaps)
best <- regsubsets(Y ~ ., data = mydata, nbest = 3, really.big = TRUE, nvmax
= 10)
summary(best)
```

```
## Subset selection object
## Call: regsubsets.formula(Y ~ ., data = mydata, nbest = 3, really.big = TRUE,
##     nvmax = 10)
## 10 Variables  (and intercept)
##     Forced in Forced out
## X       FALSE      FALSE
## X2      FALSE      FALSE
## X3      FALSE      FALSE
## X4      FALSE      FALSE
## X5      FALSE      FALSE
## X6      FALSE      FALSE
## X7      FALSE      FALSE
## X8      FALSE      FALSE
## X9      FALSE      FALSE
## X10     FALSE      FALSE
## 3 subsets of each size up to 10
## Selection Algorithm: exhaustive
##           X   X2  X3  X4  X5  X6  X7  X8  X9  X10
## 1  ( 1 )  "*" " " " " " " " " " " " " " " " " " "
## 1  ( 2 )  " " " " "*" " " " " " " " " " " " " " "
## 1  ( 3 )  " " " " " " " " "*" " " " " " " " " " "
## 2  ( 1 )  "*" "*" " " " " " " " " " " " " " " " "
## 2  ( 2 )  "*" " " "*" " " " " " " " " " " " " " "
## 2  ( 3 )  "*" " " " " " " "*" " " " " " " " " " "
## 3  ( 1 )  "*" " " " " "*" " " " " " " " " "*" " "
## 3  ( 2 )  "*" " " " " "*" " " " " "*" " " " " " "
## 3  ( 3 )  "*" " " " " "*" " " " " " " " " " " "*"
## 4  ( 1 )  "*" " " "*" "*" " " " " " " "*" " " " "
## 4  ( 2 )  "*" " " "*" "*" " " " " " " " " " " "*"
## 4  ( 3 )  "*" " " "*" "*" " " "*" " " " " " " " "
## 5  ( 1 )  "*" " " " " "*" "*" " " "*" " " " " "*"
## 5  ( 2 )  "*" " " "*" "*" " " " " " " " " "*" "*"
## 5  ( 3 )  "*" " " "*" "*" " " " " "*" " " " " "*"
## 6  ( 1 )  "*" " " "*" "*" "*" " " "*" " " "*" " "
## 6  ( 2 )  "*" " " " " "*" "*" " " "*" " " "*" "*"
## 6  ( 3 )  "*" " " " " "*" "*" " " "*" "*" "*" " "
## 7  ( 1 )  "*" " " "*" "*" "*" "*" "*" " " "*" " "
## 7  ( 2 )  "*" " " "*" "*" "*" " " "*" "*" "*" " "
## 7  ( 3 )  "*" " " "*" "*" "*" " " "*" " " "*" "*"
## 8  ( 1 )  "*" " " "*" " " "*" "*" "*" "*" "*" "*"
## 8  ( 2 )  "*" " " "*" "*" "*" " " "*" "*" "*" "*"
## 8  ( 3 )  "*" "*" "*" "*" "*" "*" "*" " " "*" " "
## 9  ( 1 )  "*" " " "*" "*" "*" "*" "*" "*" "*" "*"
## 9  ( 2 )  "*" "*" "*" " " "*" "*" "*" "*" "*" "*"
## 9  ( 3 )  "*" "*" "*" "*" "*" " " "*" "*" "*" "*"
## 10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
```

Before proceeding, follow structure in file modelselection Example: Heights. The goal is to find the best model obtained according to Cp, BIC, and adjusted R^2.

Report the coefficients of the best model obtained.

See the results:

```
sbest<-summary(best)
sbest
```

```
## Subset selection object
## Call: regsubsets.formula(Y ~ ., data = mydata, nbest = 3, really.big = TRUE,
##     nvmax = 10)
## 10 Variables  (and intercept)
##     Forced in Forced out
## X       FALSE      FALSE
## X2      FALSE      FALSE
## X3      FALSE      FALSE
## X4      FALSE      FALSE
## X5      FALSE      FALSE
## X6      FALSE      FALSE
## X7      FALSE      FALSE
## X8      FALSE      FALSE
## X9      FALSE      FALSE
## X10     FALSE      FALSE
## 3 subsets of each size up to 10
## Selection Algorithm: exhaustive
##            X   X2  X3  X4  X5  X6  X7  X8  X9  X10
## 1  ( 1 )  "*" " " " " " " " " " " " " " " " " " "
## 1  ( 2 )  " " " " "*" " " " " " " " " " " " " " "
## 1  ( 3 )  " " " " " " " " "*" " " " " " " " " " "
## 2  ( 1 )  "*" "*" " " " " " " " " " " " " " " " "
## 2  ( 2 )  "*" " " "*" " " " " " " " " " " " " " "
## 2  ( 3 )  "*" " " " " " " "*" " " " " " " " " " "
## 3  ( 1 )  "*" " " " " "*" " " " " " " " " "*" " "
## 3  ( 2 )  "*" " " " " "*" " " " " "*" " " " " " "
## 3  ( 3 )  "*" " " " " "*" " " " " " " " " " " "*"
## 4  ( 1 )  "*" " " "*" "*" " " " " " " "*" " " " "
## 4  ( 2 )  "*" " " "*" "*" " " " " " " " " " " "*"
## 4  ( 3 )  "*" " " "*" "*" " " "*" " " " " " " " "
## 5  ( 1 )  "*" " " " " "*" "*" " " "*" " " " " "*"
## 5  ( 2 )  "*" " " "*" "*" " " " " " " " " "*" "*"
## 5  ( 3 )  "*" " " "*" "*" " " " " "*" " " " " "*"
## 6  ( 1 )  "*" " " "*" "*" "*" " " "*" " " "*" " "
## 6  ( 2 )  "*" " " " " "*" "*" " " "*" " " "*" "*"
## 6  ( 3 )  "*" " " " " "*" "*" " " "*" "*" "*" " "
## 7  ( 1 )  "*" " " "*" "*" "*" "*" "*" " " "*" " "
## 7  ( 2 )  "*" " " "*" "*" "*" " " "*" "*" "*" " "
## 7  ( 3 )  "*" " " "*" "*" "*" " " "*" " " "*" "*"
## 8  ( 1 )  "*" " " "*" " " "*" "*" "*" "*" "*" "*"
## 8  ( 2 )  "*" " " "*" "*" "*" " " "*" "*" "*" "*"
## 8  ( 3 )  "*" "*" "*" "*" "*" "*" "*" " " "*" " "
## 9  ( 1 )  "*" " " "*" "*" "*" "*" "*" "*" "*" "*"
## 9  ( 2 )  "*" "*" "*" " " "*" "*" "*" "*" "*" "*"
## 9  ( 3 )  "*" "*" "*" "*" "*" " " "*" "*" "*" "*"
## 10  ( 1 ) "*" "*" "*" "*" "*" "*" "*" "*" "*" "*"
```

See what is inside sbest:

```
names(sbest)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

To see the models and their Cp: (you can do the same with adjr2 and bic)

```
cbind(sbest$which ,sbest$cp)
```

```
##    (Intercept) X X2 X3 X4 X5 X6 X7 X8 X9 X10
## 1             1 1  0  0  0  0  0  0  0  0   0  82.217093
## 1             1 0  0  1  0  0  0  0  0  0   0 205.710622
## 1             1 0  0  0  0  1  0  0  0  0   0 363.893920
## 2             1 1  1  0  0  0  0  0  0  0   0  58.039585
## 2             1 1  0  1  0  0  0  0  0  0   0  63.053376
## 2             1 1  0  0  0  1  0  0  0  0   0  69.361446
## 3             1 1  0  0  1  0  0  0  0  1   0   2.011045
## 3             1 1  0  0  1  0  0  1  0  0   0   2.518557
## 3             1 1  0  0  1  0  0  0  0  0   1   3.560569
## 4             1 1  0  1  1  0  0  0  1  0   0   1.352302
## 4             1 1  0  1  1  0  0  0  0  0   1   1.490505
## 4             1 1  0  1  1  0  1  0  0  0   0   1.634107
## 5             1 1  0  0  1  1  0  1  0  0   1   2.725378
## 5             1 1  0  1  1  0  0  0  0  1   1   2.910338
## 5             1 1  0  1  1  0  0  1  0  0   1   2.953437
## 6             1 1  0  1  1  1  0  1  0  1   0   3.997196
## 6             1 1  0  0  1  1  0  1  0  1   1   4.499197
## 6             1 1  0  0  1  1  0  1  1  1   0   4.521410
## 7             1 1  0  1  1  1  1  1  0  1   0   5.876923
## 7             1 1  0  1  1  1  0  1  1  1   0   5.887372
## 7             1 1  0  1  1  1  0  1  0  1   1   5.921573
## 8             1 1  0  1  0  1  1  1  1  1   1   7.390675
## 8             1 1  0  1  1  1  0  1  1  1   1   7.808436
## 8             1 1  1  1  1  1  1  1  0  1   0   7.858416
## 9             1 1  0  1  1  1  1  1  1  1   1   9.297465
## 9             1 1  1  1  0  1  1  1  1  1   1   9.375591
## 9             1 1  1  1  1  1  0  1  1  1   1   9.665630
## 10            1 1  1  1  1  1  1  1  1  1   1  11.000000
```

```
cbind(sbest$which, sbest$adjr2)
```

```
##    (Intercept) X X2 X3 X4 X5 X6 X7 X8 X9 X10
## 1             1 1  0  0  0  0  0  0  0  0   0 0.7387371
## 1             1 0  0  1  0  0  0  0  0  0   0 0.5576979
## 1             1 0  0  0  0  1  0  0  0  0   0 0.3258042
## 2             1 1  1  0  0  0  0  0  0  0   0 0.7748150
## 2             1 1  0  1  0  0  0  0  0  0   0 0.7673891
## 2             1 1  0  0  0  1  0  0  0  0   0 0.7580463
## 3             1 1  0  0  1  0  0  0  0  1   0 0.8593104
## 3             1 1  0  0  1  0  0  1  0  0   0 0.8585509
## 3             1 1  0  0  1  0  0  0  0  0   1 0.8569915
## 4             1 1  0  1  1  0  0  0  1  0   0 0.8618502
## 4             1 1  0  1  1  0  0  0  0  0   1 0.8616412
## 4             1 1  0  1  1  0  1  0  0  0   0 0.8614240
## 5             1 1  0  0  1  1  0  1  0  0   1 0.8613387
## 5             1 1  0  1  1  0  0  0  0  1   1 0.8610560
## 5             1 1  0  1  1  0  0  1  0  0   1 0.8609901
## 6             1 1  0  1  1  1  0  1  0  1   0 0.8609726
## 6             1 1  0  0  1  1  0  1  0  1   1 0.8601971
## 6             1 1  0  0  1  1  0  1  1  1   0 0.8601628
## 7             1 1  0  1  1  1  1  1  0  1   0 0.8596492
## 7             1 1  0  1  1  1  0  1  1  1   0 0.8596329
## 7             1 1  0  1  1  1  0  1  0  1   1 0.8595795
## 8             1 1  0  1  0  1  1  1  1  1   1 0.8588746
## 8            1 1  0  1  1  1  0  1  1  1   1 0.8582150
## 8            1 1  1  1  1  1  1  1  0  1   0 0.8581361
## 9            1 1  0  1  1  1  1  1  1  1   1 0.8574553
## 9            1 1  1  1  0  1  1  1  1  1   1 0.8573306
## 9            1 1  1  1  1  1  0  1  1  1   1 0.8568676
## 10           1 1  1  1  1  1  1  1  1  1   1 0.8563339
```

```
cbind(sbest$which, sbest$bic)
```

```
##    (Intercept) X X2 X3 X4 X5 X6 X7 X8 X9 X10
## 1            1 1  0  0  0  0  0  0  0  0   0 -126.02771
## 1            1 0  0  1  0  0  0  0  0  0   0  -73.38112
## 1            1 0  0  0  0  1  0  0  0  0   0  -31.22837
## 2            1 1  1  0  0  0  0  0  0  0   0 -137.30869
## 2            1 1  0  1  0  0  0  0  0  0   0 -134.06421
## 2            1 1  0  0  0  1  0  0  0  0   0 -130.12626
## 3            1 1  0  0  1  0  0  0  0  1   0 -180.77638
## 3            1 1  0  0  1  0  0  1  0  0   0 -180.23799
## 3            1 1  0  0  1  0  0  0  0  0   1 -179.14158
## 4            1 1  0  1  1  0  0  0  1  0   0 -179.04009
## 4            1 1  0  1  1  0  0  0  0  0   1 -178.88892
## 4            1 1  0  1  1  0  1  0  0  0   0 -178.73209
## 5            1 1  0  0  1  1  0  1  0  0   1 -175.12356
## 5            1 1  0  1  1  0  0  0  0  1   1 -174.91990
## 5            1 1  0  1  1  0  0  1  0  0   1 -174.87250
## 6            1 1  0  1  1  1  0  1  0  1   0 -171.32425
## 6            1 1  0  0  1  1  0  1  0  1   1 -170.76800
## 6            1 1  0  0  1  1  0  1  1  1   0 -170.74346
## 7            1 1  0  1  1  1  1  1  0  1   0 -166.85281
## 7            1 1  0  1  1  1  0  1  1  1   0 -166.84118
## 7            1 1  0  1  1  1  0  1  0  1   1 -166.80314
## 8            1 1  0  1  0  1  1  1  1  1   1 -162.79012
## 8            1 1  0  1  1  1  0  1  1  1   1 -162.32387
## 8            1 1  1  1  1  1  1  1  0  1   0 -162.26823
## 9            1 1  0  1  1  1  1  1  1  1   1 -158.28928
## 9            1 1  1  1  0  1  1  1  1  1   1 -158.20183
## 9            1 1  1  1  1  1  0  1  1  1   1 -157.87784
## 10           1 1  1  1  1  1  1  1  1  1   1 -154.01778
```

Follow the notes in modelselection Example: Heights to see how to select the best model according to the various metrics.

```
mybestmodel<-function(Xnames,Yname,dataset,p,crit="bic"){
if(crit=="Cp"){
n<-dim(dataset)[1]
fullMSE=summary(lm(as.formula(paste(Yname,"~.")),data=dataset))$sigma^2
}
varsel<-lapply(0:p, function(x) combn(p,x))
modcrit<-numeric(p); form<-character(p)
for(k in 1:p){
s<-dim(varsel[[k+1]])[2]
tempform<-character(s); tempcrit<-numeric(s)
for(j in 1:s){
temp <- Xnames[varsel[[k+1]][,j]]
tempform[j]<- ifelse(length(temp)>1,
paste(temp, collapse = " + "), temp)
```

```r
tempform[j] <- paste(Yname, tempform[j],sep='~')
tempmod<-lm(as.formula(tempform[j]),data=dataset)
if(crit=="aic"){
tempcrit[j] <- AIC(tempmod)
}
if(crit=="bic"){
tempcrit[j] <- BIC(tempmod)
}
if(crit=="r2"){
tempcrit[j] <- summary(tempmod)$adj
}
if(crit=="Cp"){
tempcrit[j]<-sum(tempmod$res^2)/fullMSE+2*(k+1)-n
}
}
# best model of size k
if(crit %in% c("aic", "bic")){
best<-which.min(tempcrit)
}
if(crit == "r2"){
best<-which.max(tempcrit)
}
if(crit=="Cp"){
best<-which.min(abs(tempcrit[j]-(k+1)))
}
form[k]<-tempform[best]
modcrit[k]<-tempcrit[best]
}
if(crit %in% c("aic", "bic")){
out<-form[which.min(modcrit)]
}
if(crit == "r2"){
out<-form[which.max(modcrit)]
}
if(crit=="Cp"){
out<-form[which.min(abs(modcrit[-p]-(2:p)))]
}
return(out)
}
```

```r
suppressWarnings({
p<-length(names(mydata))-1
Xnames<-names(mydata)[-1]
Yname<-"Y"
dataset<-mydata
bicform<-mybestmodel(Xnames, Yname, mydata, p, crit="bic")
bicform})
```

```
## [1] "Y~X3 + X4 + X5 + X7"
```

```r
Criteria<-function(model){
out<-data.frame(`p+1`
=length(model$coef),
R2adj=summary(model)$adj,
AIC=AIC(model),
BIC=BIC(model))
return(out)
}
suppressWarnings({
modbic<-lm(as.formula(bicform), data=mydata)
aicform<-mybestmodel(Xnames, Yname, mydata, p, crit="aic")
modaic<-lm(as.formula(aicform), data=mydata)
cpform<-mybestmodel(Xnames, Yname, mydata, p, crit="Cp")
modCp<-lm(as.formula(cpform), data=mydata)
r2form<-mybestmodel(Xnames, Yname, mydata, p, crit="r2")
modr2<-lm(as.formula(r2form), data=mydata)})
```

Enter here the best model for each metric: bic, aic, cp, adjR2.

```r
suppressWarnings({# Assuming 'mydata' is your data frame, properly formatted
p <- length(names(mydata)) - 1  # Number of predictors
Xnames <- names(mydata)[-1]  # Names of predictor variables
Yname <- "Y"  # Name of the response variable
dataset <- mydata  # The dataset

# Initialize a list to store the best model formulas
best_models <- list()

# Find the best model according to BIC
best_models$bic <- mybestmodel(Xnames, Yname, dataset, p, crit = "bic")

# Find the best model according to AIC
best_models$aic <- mybestmodel(Xnames, Yname, dataset, p, crit = "aic")

# Find the best model according to Cp
best_models$cp <- mybestmodel(Xnames, Yname, dataset, p, crit = "Cp")

# Find the best model according to adjusted R^2
best_models$adjR2 <- mybestmodel(Xnames, Yname, dataset, p, crit = "r2")

# Display the best models
best_models

})
```

```
## $bic
## [1] "Y~X3 + X4 + X5 + X7"
##
## $aic
## [1] "Y~X3 + X5 + X6 + X7 + X9 + X10"
##
## $cp
## [1] "Y~X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10"
##
## $adjR2
## [1] "Y~X3 + X5 + X6 + X7 + X9 + X10"
```

(d) Now find the best model using backward stepwise selection.

```r
#backwards
modback<-step(lm(Y~.,data=mydata),trace=0, direction = "backward")
modback
```

```
##
## Call:
## lm(formula = Y ~ X + X5 + X6 + X7 + X8, data = mydata)
##
## Coefficients:
## (Intercept)            X           X5           X6           X7           X8
##    1.695463     1.709011     0.055870    -0.038078    -0.006072     0.003415
```

(e) Display the adjR^2, BIC and AIC for all the model, that is, the model selected by Cp, by AIC, by BIC, by adj R-square and the backwards selection.

```r
rbind(bicm=Criteria(modbic),
aicm=Criteria(modaic),
adjr2m=Criteria(modr2),
cpm=Criteria(modCp),
bsel=Criteria(modback)
)
```

```
##        p.1     R2adj      AIC      BIC
## bicm     5 0.8184585 311.1615 326.7925
## aicm     7 0.8279829 307.6447 328.4861
## adjr2m   7 0.8279829 307.6447 328.4861
## cpm     10 0.8243252 312.4698 341.1267
## bsel     6 0.8607495 285.5823 303.8185
```

(f) Write down the model equation for each of the models.

Equation for each model:

Bic: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \epsilon$

Aic: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5 + \beta_6 X_6 + \epsilon$

Cp: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5 + \beta_6 X_6 + \beta_7 X_7 + \beta_8 X_8 + \beta_9 X_9 + \beta_10X_10 + \epsilon$

AdjR2: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5 + \beta_6 X_6 + \beta_7 X_7 + \epsilon$

backwards selection: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \beta_5 X_5 + \beta_6 X_6 + \beta_7 X_7 + \beta_8 X_8 + \epsilon$

Comment on your observations comparing the variables selected by the models vs the equations for the response variable Y. Do they all contain X_1, X_2 and X_3? Do they contain other variables? Comment on your findings.

Ans. The models BIC, AIC, Cp and AdjR2 contain X_1, X_2, X_3 along with other variables of X. Each model selection has led to different model complexity ranging from 4 to 10 predictors. The models with more predictors, such as C_p model (10 predictors) and the backward selection model (8 predictors) likely include additional variables beyond X_1, X_2, X_3. The adjusted R^2 value is highest for the backward model suggesting that it might provide the best fit among the models.

(g) Now use cross validation to select the final model.

```r
cv.lm <- function(data, formulae, nfolds = 5) {
data <- na.omit(data) # remove missing values
formulae <- sapply(formulae, as.formula)
n <- nrow(data)
fold.labels <- sample(rep(1:nfolds, length.out = n))
mses <- matrix(NA, nrow = nfolds, ncol = length(formulae))
colnames <- as.character(formulae)
for (fold in 1:nfolds) {
test.rows <- which(fold.labels == fold)
train <- data[-test.rows, ]
test <- data[test.rows, ]
for (form in 1:length(formulae)) {
current.model <- lm(formula = formulae[[form]], data = train)
predictions <- predict(current.model, newdata = test)
test.responses <- eval(formulae[[form]][[2]], envir = test)
test.errors <- test.responses - predictions
mses[fold, form] <- mean(test.errors^2) } }
return(colMeans(mses))
}
formulae<-c(formula(modbic),
formula(modaic),
formula(modr2),
formula(modCp),
formula(modback))
mse<-cv.lm(data=mydata, formulae, nfolds = 5)
print(mse)
```

```
## [1]    349.2566 126248.7799 126248.7799 163381.7989    359.1320
```

2. (20) The perils of post-selection inference, and data splitting to the rescue.

A.

(a) Generate a 1000 x 101 array, where all the entries are IID standard Gaussian variables. We'll call the first column the response variable Y, and the others the predictors $X_1, \ldots, X_{100}$.

```r
mydata<-as.data.frame(matrix(rnorm(1000*101),ncol=101))
names(mydata)<-c("Y",paste("X",1:100,sep=""))
mydata[1:3,1:5]
```

```
##            Y         X1         X2          X3         X4
## 1  0.4029399 -0.3259896  0.3833298 -0.04113137 -1.0038915
## 2 -0.8860570  1.2897669 -1.6179002 -0.56815805  0.8456529
## 3 -0.1198037 -0.7993374 -0.4437829  0.13590486  0.4517901
```

By design, there is no true relationship between the response and the predictors (but all the usual linear-Gaussian-modeling assumptions hold).

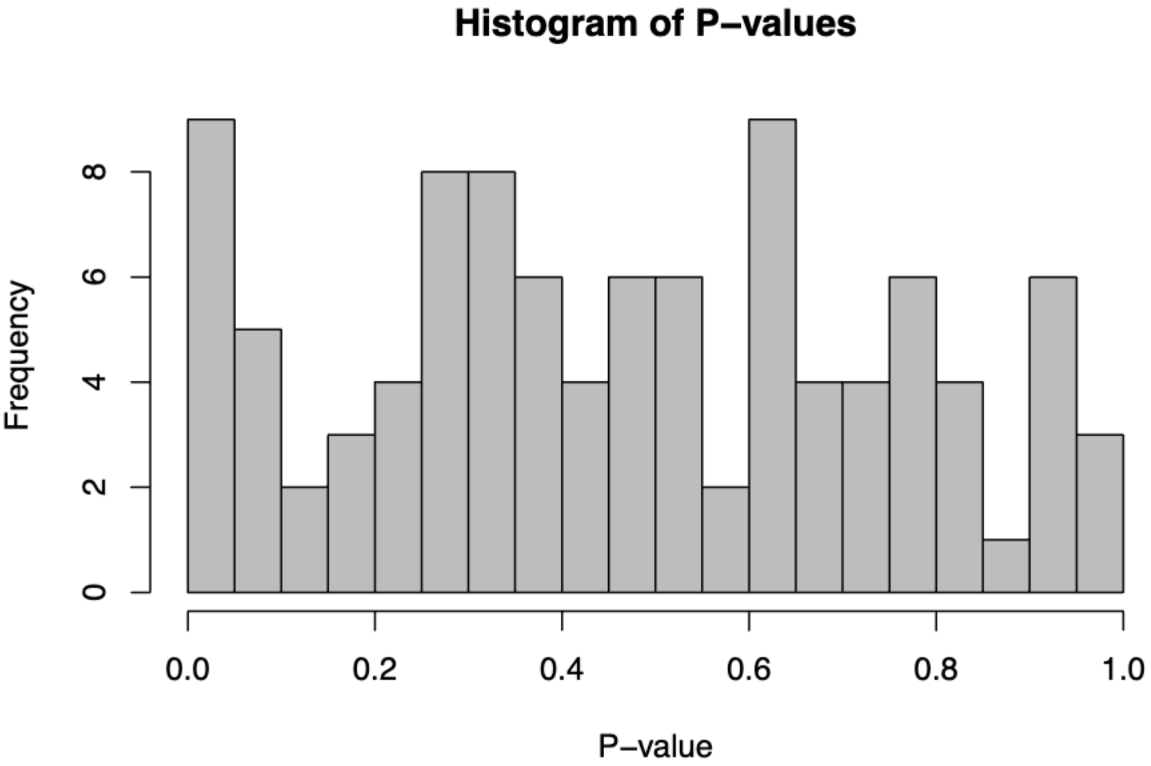
(b) Estimate the model $Y = \beta_0 + \beta_1 X_1 + \beta_{50} X_{50} + \epsilon$ Extract the p-value for the **F test** of the whole model.

```r
getpvalue <- function(model, data) {
    model <- lm(model, data = data)
    sum <- summary(model)
    pvalue <- pf(sum$fstatistic[1], sum$fstatistic[2], sum$fstatistic[3], lower.tail=FALSE)
    return (pvalue)
}

cat("The Extracted value of p is ", getpvalue(Y ~ X1 + X50, mydata))
```

```
## The Extracted value of p is  0.3186972
```

Repeat the simulation (steps a and b), estimation and testing 100 times, and plot the histogram of the p-values. What does it look like? What should it look like?

```
# Function to generate data and get p-value
getpvalue <- function() {
  # Generate data
  mydata <- as.data.frame(matrix(rnorm(1000 * 101), ncol = 101))
  names(mydata) <- c("Y", paste("X", 1:100, sep = ""))

  # Fit model and get summary
  model <- lm(Y ~ X1 + X50, data = mydata)
  sum <- summary(model)

  # Calculate p-value
  pvalue <- pf(sum$fstatistic[1], sum$fstatistic[2], sum$fstatistic[3], lower.tail = FALSE)

  return(pvalue)
}

# Repeat simulation 100 times
p_values <- replicate(100, getpvalue())

# Plot histogram of p-values
hist(p_values, main = "Histogram of P-values", xlab = "P-value", breaks = 20, col = "grey", border = "bl
```



B.

(c) From step (a), use the step function to select a linear model by backward stepwise selection. Extract the p-value for the **F-test** of the selected model.

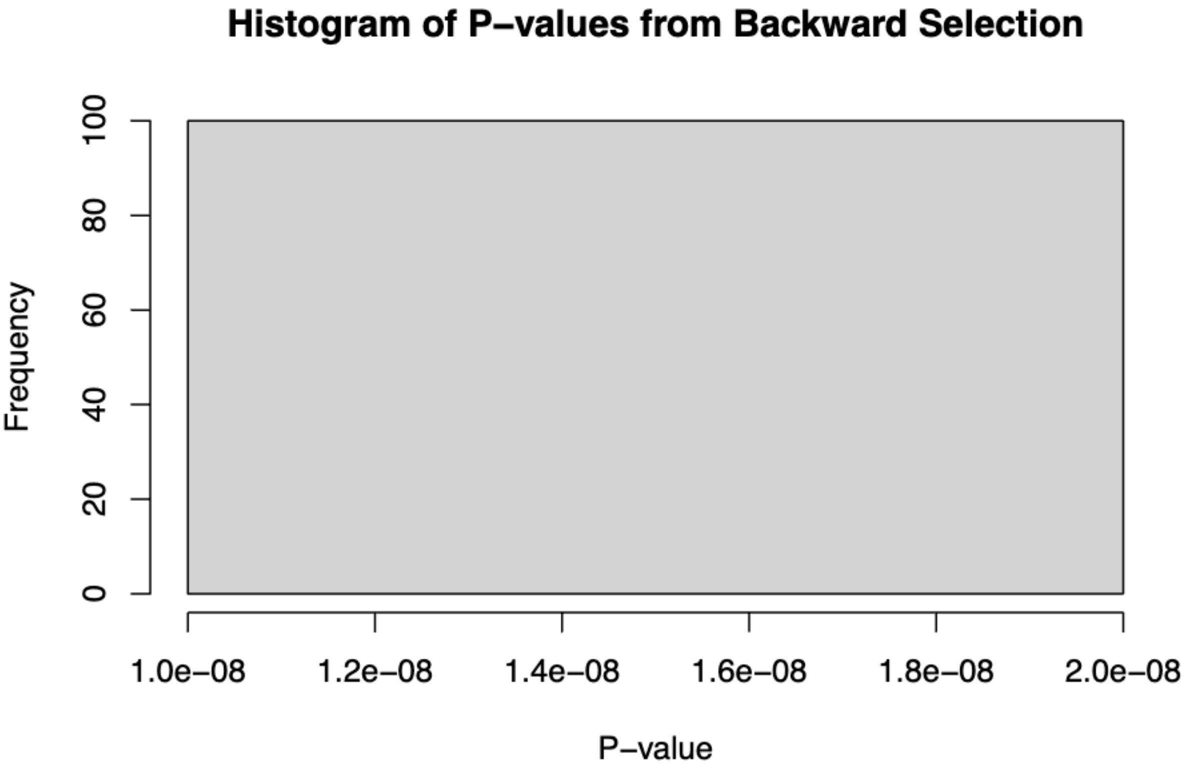
```
#final_model <- step(lm(Y~., mydata2), direction = "backward", trace = 0)
model_backward <- function(model, data){
final_model <- step(model, direction = "backward", trace = 0)
```

```r
sum<-summary(final_model)
pvalue<-pf(sum$fstatistic[1], sum$fstatistic[2], sum$fstatistic[3],
lower.tail=FALSE)
return (pvalue)
}
model <- lm(Y~., data = mydata)
cat("The Extracted p-value is ",model_backward(model, mydata))
```

```
## The Extracted p-value is  1.277748e-08
```

Repeat (a) and (c) 100 times and plot the histogram of p-values. Explain what's going on.

```r
library(parallel)

get_p_value_backward <- function(mydata) {
  f_model <- lm(Y ~ ., data = mydata)
  selected_model <- step(f_model, direction = "backward", trace = 0)
  sum <- summary(selected_model)
  pvalue <- pf(sum$fstatistic[1], sum$fstatistic[2], sum$fstatistic[3], lower.tail = FALSE)

  return (pvalue)
}

# Assuming 'mydata' is already defined and 'Y' is the dependent variable in 'mydata'
# Detect the number of available cores
no_cores <- detectCores() - 1

# Use parallel processing to run the replications
p_values <- mclapply(1:100, function(x) get_p_value_backward(mydata), mc.cores = no_cores)

# Convert the list to a numeric vector
p_values_vector <- unlist(p_values)

# Plot histogram of p-values
hist(p_values_vector, main="Histogram of P-values from Backward Selection", xlab="P-value", breaks=20)
```



3. (15) Practicing one-way ANOVA. We will use the crab_dataset

```
library(MASS)     # install MASS if you need to
head(crabs)
```

```
##   sp sex index   FL  RW   CL   CW  BD
## 1  B   M     1  8.1 6.7 16.1 19.0 7.0
## 2  B   M     2  8.8 7.7 18.1 20.8 7.4
## 3  B   M     3  9.2 7.8 19.0 22.4 7.7
## 4  B   M     4  9.6 7.9 20.1 23.1 8.2
## 5  B   M     5  9.8 8.0 20.3 23.0 8.2
## 6  B   M     6 10.8 9.0 23.0 26.5 9.8
```

Notice that there are two categorical variables:

- sp for species as “B” or “O” for blue or orange and
- sex for “M” males and “F” females.

Conduct a one-way ANOVA to determine if species variable sp have different means for the continuous variable CL or the carapace length (mm).

First we check the assumption of homogeneity of the variance among the groups.

```
aggregate(CL ~ sp , data = crabs, FUN = sd)
```

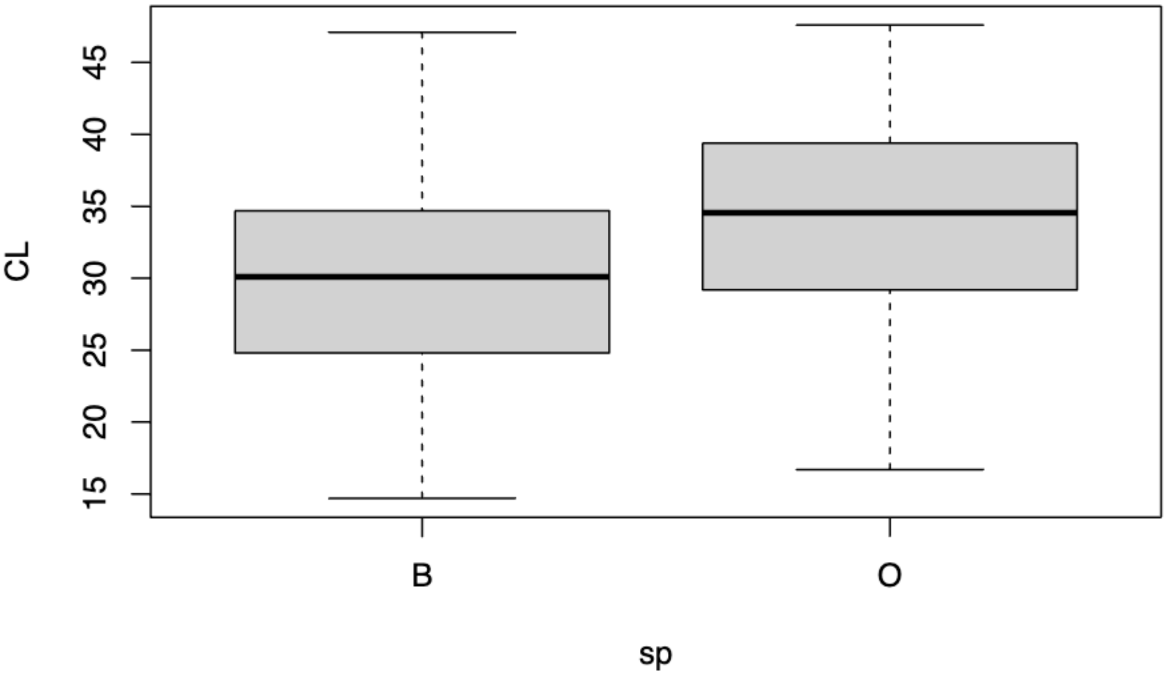
```
##   sp       CL
## 1  B 6.902703
## 2  O 6.764262
```

A general rule of thumb for equal variances is to compare the smallest and largest sample standard deviations. If the ratio of these two sample standard deviations falls within 0.5 to 2, then it may be that the assumption is not violated.

In this case, the sd are very close. The homogeneity of the variance assumption checks.

a) Make a boxplot of CL vs sp.

```
boxplot(CL~sp, data = crabs)
```



What do you notice?

The box plot compares two groups, labeled B and O, on a continuous variable CL.

Central Tendency: The median of group B is lower than the median of group O, as indicated by the line within the box.

Variability: Both groups have a similar range of variability, as suggested by the height of the boxes and the length of the whiskers.

Outliers: There are no outliers indicated in either group, as there are no points plotted beyond the whiskers.

Symmetry: Both groups show a fairly symmetric distribution of CL, with the box being relatively centered around the median.

Interquartile Range: The interquartile range (IQR), represented by the height of the box, seems slightly larger for group O than for group B, which indicates more variability in the middle 50% of the data for group O.

b) Write the hypothesis for this ANOVA analysis.

Ans) For a one-way ANOVA with CL as the dependent variable and species (sp) as the independent variable, the hypotheses would be:

Null Hypothesis (H_0): The means of CL are the same across all species. There are no significant differences among the species' mean CL. Mathematically, it can be stated as: $\mu_1 = \mu_2 = \ldots = \mu_k$ (where μ represents the mean CL for each species and k is the number of species). Alternative Hypothesis (H_1): At least one species has a mean CL that is significantly different from the others.

c) run the one way ANOVA test for factor sp.

```
anova_result <- aov(CL ~ sp, data = crabs)
summary(anova_result)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## sp            1    838   838.5   17.95 3.47e-05 ***
## Residuals   198   9247    46.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Conclusion:

The p-value is 3.47e-05 which is less than the significance level, Reject the null hypothesis and we can conclude that there is a significant difference in mean `CL` among at least some of the species.

Two way ANOVA with sp and sex.

```r
anova_result_2way <- aov(CL ~ sp * sex, data = crabs)
summary(anova_result_2way)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)    
## sp            1    838   838.5  18.585 2.57e-05 ***
## sex           1    111   111.2   2.464   0.1181    
## sp:sex        1    293   293.1   6.496   0.0116 *  
## Residuals   196   8843    45.1                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Means per group:

```r
aggregate(CL ~ sp + sex, data = crabs, FUN = mean)
```

```
##   sp sex     CL
## 1  B   F 28.102
## 2  O   F 34.618
## 3  B   M 32.014
## 4  O   M 33.688
```

Checking the assumption of constant variance in the groups:

```r
aggregate(CL ~ sp + sex, data = crabs, FUN = sd)
```

```
##   sp sex       CL
## 1  B   F 5.919649
## 2  O   F 5.837168
## 3  B   M 7.308676
## 4  O   M 7.611207
```

A general rule of thumb for equal variances is to compare the smallest and largest sample standard deviations. If the ratio of these two sample standard deviations falls within 0.5 to 2, then it may be that the assumption is not violated.

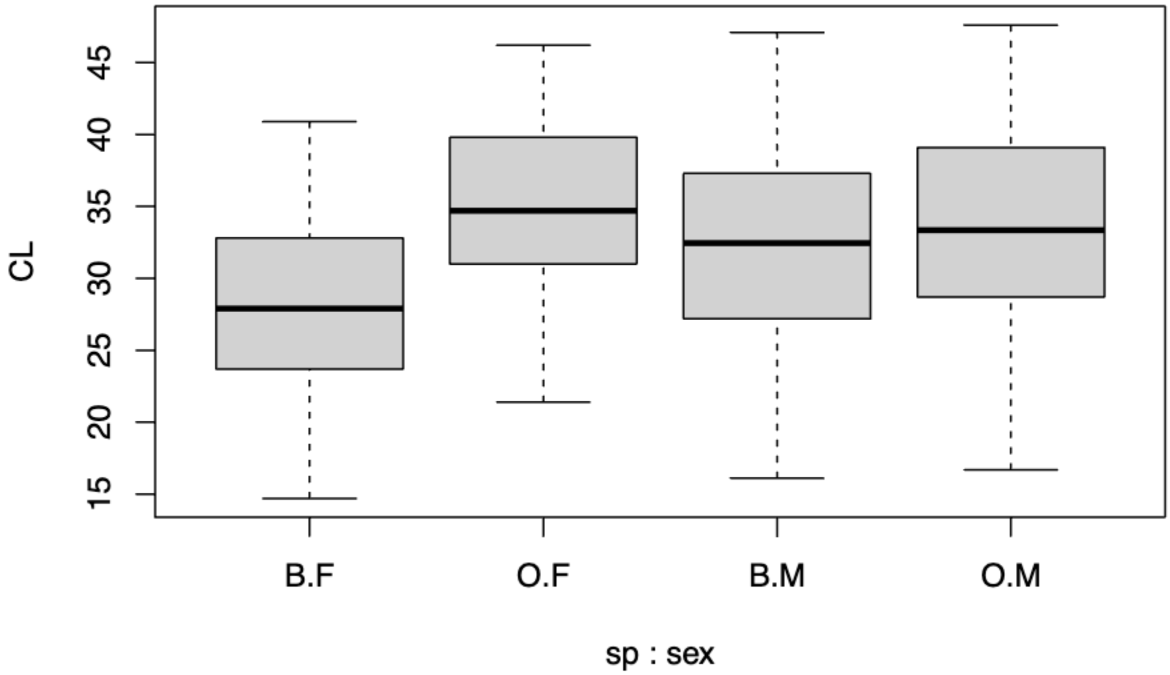
Variance check:

The Variance check for species alone is $Ratio = SmallestSD/LargestSD = 6.764262/6.902703 = 0.98(approx)$
The Variance check for species and sex combination is $Ratio = SmallestSD/LargestSD = 5.837168/7.611207 = 0.77$

Both the ratios fall within the range of 0.5 to 2, which suggests that the assumption is not violated

d) Make a boxplot of CL vs sp+sex.

```r
boxplot(CL~sp+sex,data=crabs)
```



e) Write the hypothesis for this two way ANOVA analysis.

1. Main Effect of Species (sp):

Null Hypothesis (H0): The means of CL are the same across all species. Alternative Hypothesis (H1): At least one species has a mean CL that is significantly different from the others.

2. Main Effect of Sex (sex):

Null Hypothesis (H0): The means of CL are the same for all sexes. Alternative Hypothesis (H1): The mean CL is significantly different between sexes.

3. Interaction Effect of Species and Sex (sp:sex):

Null Hypothesis (H0): There is no interaction effect between species and sex on the mean CL. In other words, the effect of species on CL is the same for all sexes, and vice versa. Alternative Hypothesis (H1): The effect of species on CL differs by sex, or the effect of sex on CL differs by species.

f) run the one way ANOVA test for factors sp and sex.

```
# One-way ANOVA for the factor 'sp'
anova_result_sp <- aov(CL ~ sp, data = crabs)
summary(anova_result_sp)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)    
## sp            1    838   838.5   17.95 3.47e-05 ***
## Residuals   198   9247    46.7                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# One-way ANOVA for the factor 'sex'
anova_result_sp <- aov(CL ~ sex, data = crabs)
summary(anova_result_sp)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## sex           1    111  111.15   2.207  0.139
## Residuals   198   9974   50.37               
```

Conclusion:

For sp: The p-value for sp is 3.47e-05, which indicates that the value is lower than 0.05, so we can reject the null hypothesis and conclude that their is a statistically significant effect of species on carapace length CL.

For sex: The p-value for sex is 0.139, which indicates that the value is higher than 0.05, so we can't reject the null hypothesis and conclude that their is not a statistically significant effect of sex on carapace length CL.

4. (15) Repeat with dataset sat.act from package psych.

```
library(psych)
str(sat.act)
```

```
## 'data.frame':    700 obs. of  6 variables:
##  $ gender   : int  2 2 2 1 1 1 2 1 2 2 ...
##  $ education: int  3 3 3 4 2 5 5 3 4 5 ...
##  $ age      : int  19 23 20 27 33 26 30 19 23 40 ...
##  $ ACT      : int  24 35 21 26 31 28 36 22 22 35 ...
##  $ SATV     : int  500 600 480 550 600 640 610 520 400 730 ...
##  $ SATQ     : int  500 500 470 520 550 640 500 560 600 800 ...
```

Note that gender is coded as male = 1, female = 2. Education is coded as 0,1,2,3,4,5 but both are categorical variables. Recode the variables so that they are treated as factors. Make appropriate boxplot for each situation. Also, use aggregate() as shown in the example above, to observe the mean corresponding to each group.

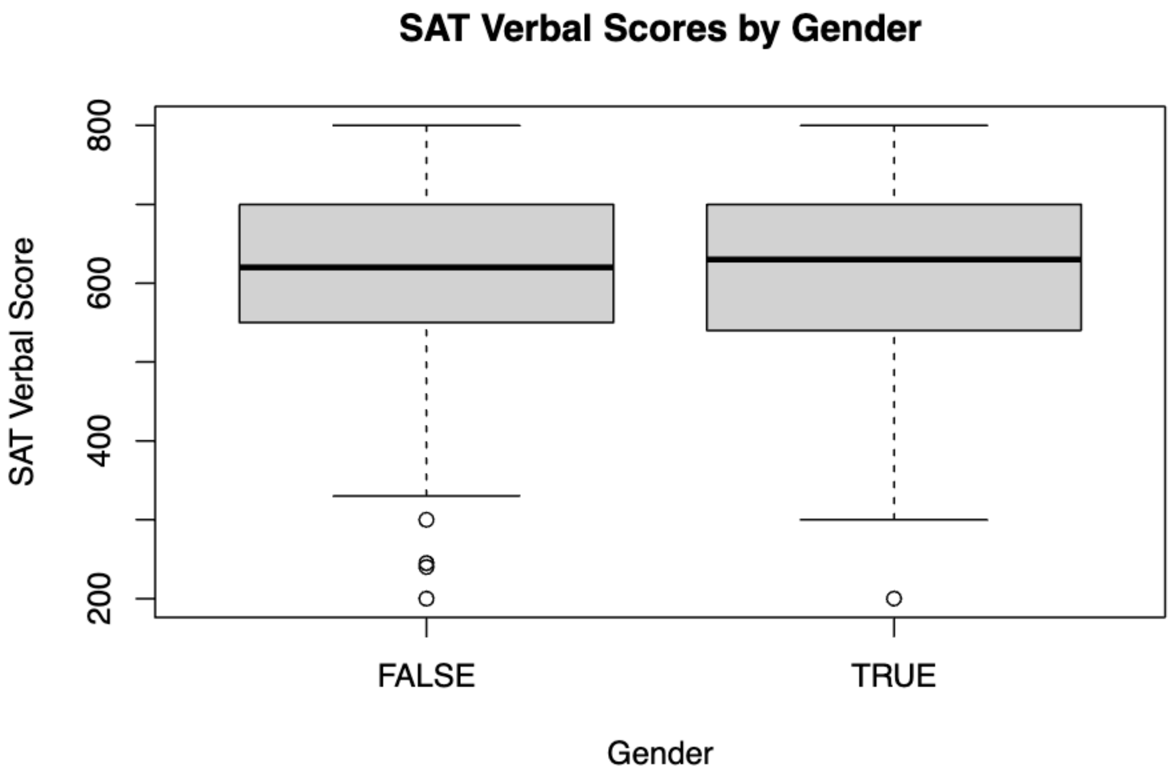
```
names(sat.act)
```

```
## [1] "gender"    "education" "age"       "ACT"       "SATV"      "SATQ"
```

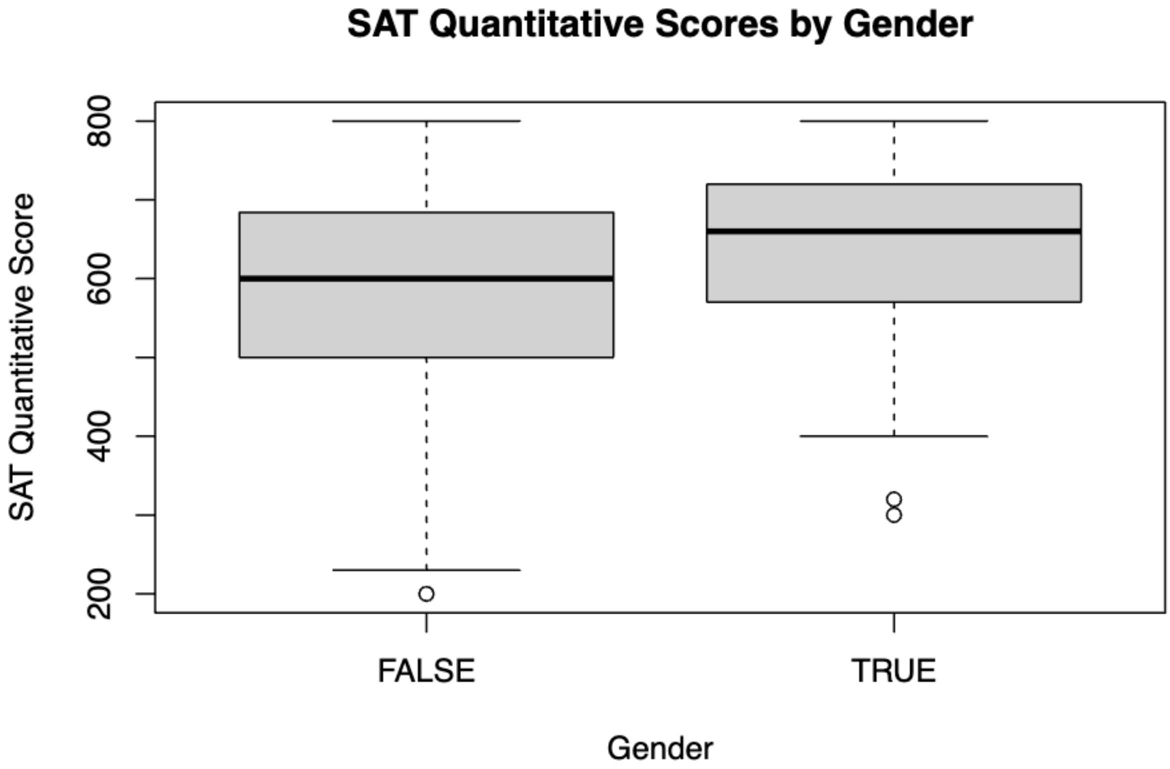
```
head(sat.act)
```

```
##       gender education age ACT SATV SATQ
## 29442      2         3  19  24  500  500
## 29457      2         3  23  35  600  500
## 29498      2         3  20  21  480  470
## 29503      1         4  27  26  550  520
## 29504      1         2  33  31  600  550
## 29518      1         5  26  28  640  640
```

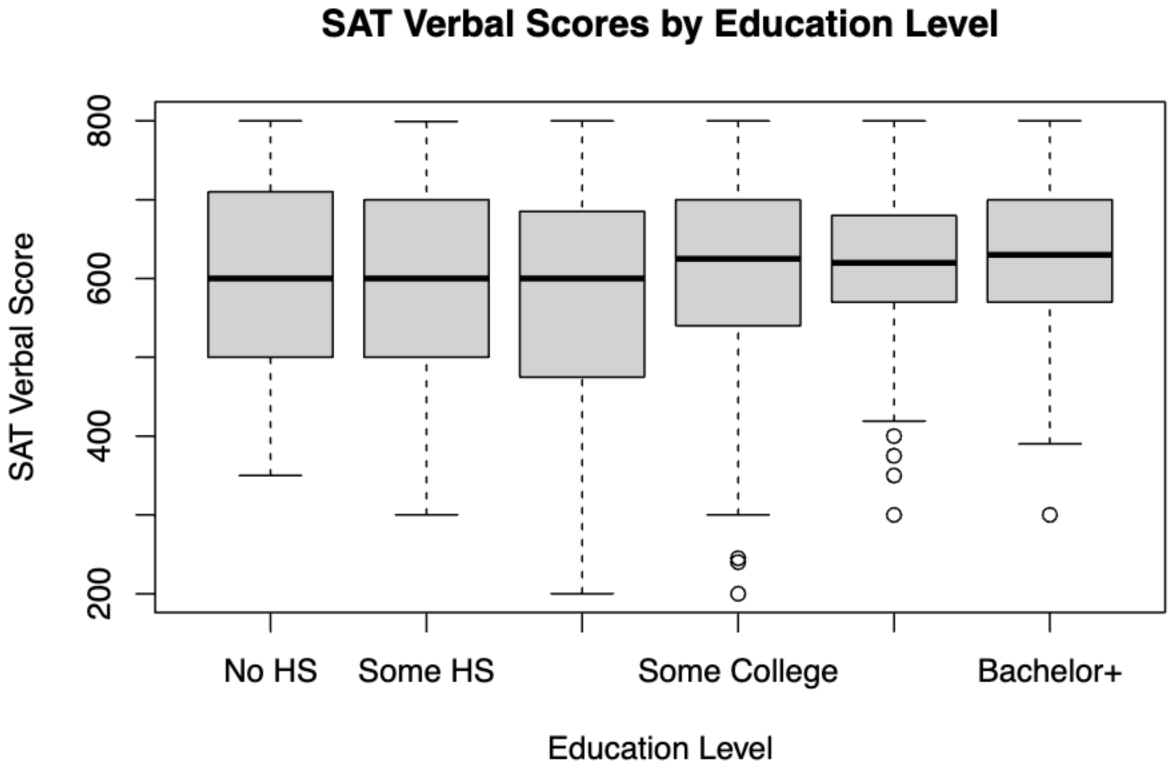
```
sat.act$male <- sat.act$gender == 1

sat.act$ed <- factor(sat.act$education, levels = c(0, 1, 2, 3, 4, 5), labels = c("No HS", "Some HS", "H

# Boxplot for SAT Verbal scores by gender
boxplot(SATV ~ male, data = sat.act, main = "SAT Verbal Scores by Gender", xlab = "Gender", ylab = "SAT
```



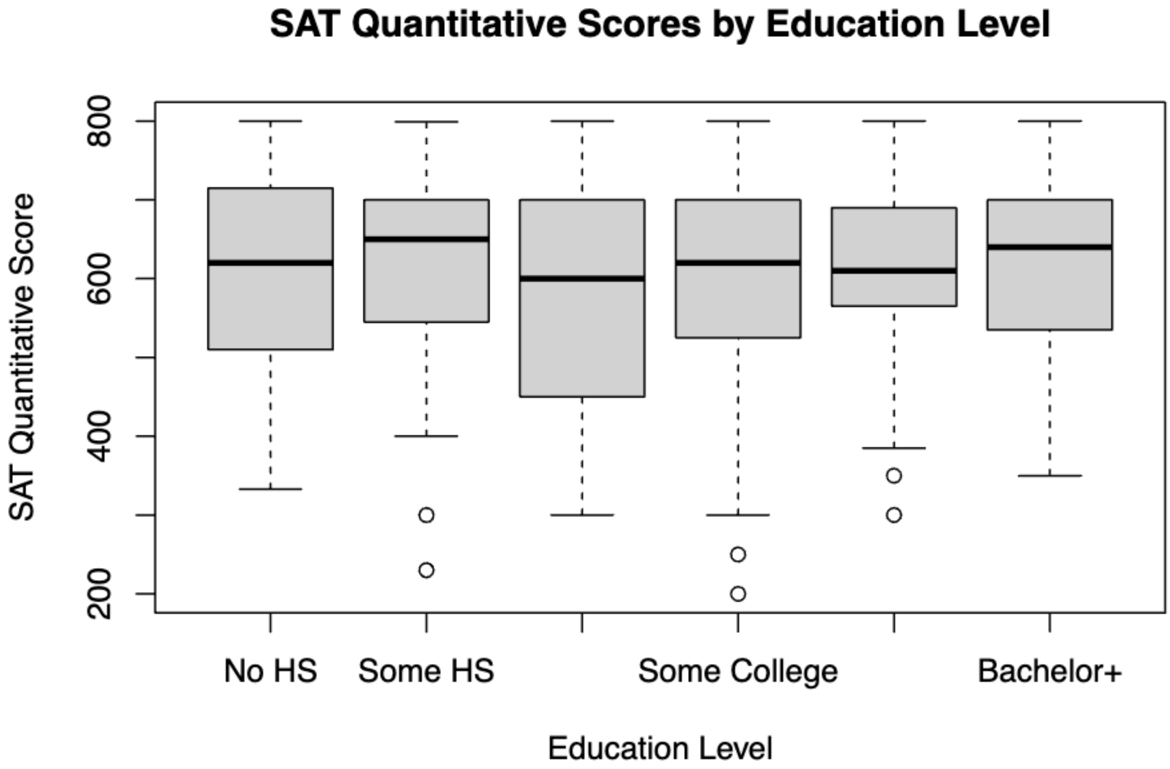
```
# Boxplot for SAT Quantitative scores by gender
boxplot(SATQ ~ male, data = sat.act, main = "SAT Quantitative Scores by Gender", xlab = "Gender", ylab =
```



```
# Boxplot for SAT Verbal scores by education level
boxplot(SATV ~ ed, data = sat.act, main = "SAT Verbal Scores by Education Level", xlab = "Education Leve
```



```r
# Boxplot for SAT Quantitative scores by education level
boxplot(SATQ ~ ed, data = sat.act, main = "SAT Quantitative Scores by Education Level", xlab = "Educati
```



```r
# Mean SAT Verbal scores by gender
aggregate(SATV ~ male, data = sat.act, FUN = mean)
```

```
##    male     SATV
## 1 FALSE 610.6645
## 2  TRUE 615.1134
```

```
# Mean SAT Quantitative scores by gender
aggregate(SATQ ~ male, data = sat.act, FUN = mean)
```

```
##    male     SATQ
## 1 FALSE 595.9955
## 2  TRUE 635.8735
```

```
# Mean SAT Verbal scores by education level
aggregate(SATV ~ ed, data = sat.act, FUN = mean)
```

```
##              ed     SATV
## 1         No HS 616.5088
## 2       Some HS 599.6667
## 3       HS Grad 576.0227
## 4  Some College 612.1345
## 5     Associate 616.9493
## 6     Bachelor+ 621.3972
```

```
# Mean SAT Quantitative scores by education level
aggregate(SATQ ~ ed, data = sat.act, FUN = mean)
```

```
##              ed     SATQ
## 1         No HS 620.4286
## 2       Some HS 607.2558
## 3       HS Grad 577.2093
## 4  Some College 606.0743
## 5     Associate 611.8540
## 6     Bachelor+ 623.6331
```

a) Use anova to investigate if the mean ACT is the same for males and females.

Hypothesis:

Null Hypothesis: The mean ACT score is the same for males and females. There is no effect of gender on ACT Scores

$H_0 : \mu_{male} = \mu_{female}$

Alternative Hypothesis: The mean ACT score is no the same for males and females. There is an effect of gender on ACT Scores

$H_1 : \mu_{male} \neq \mu_{female}$

Test:

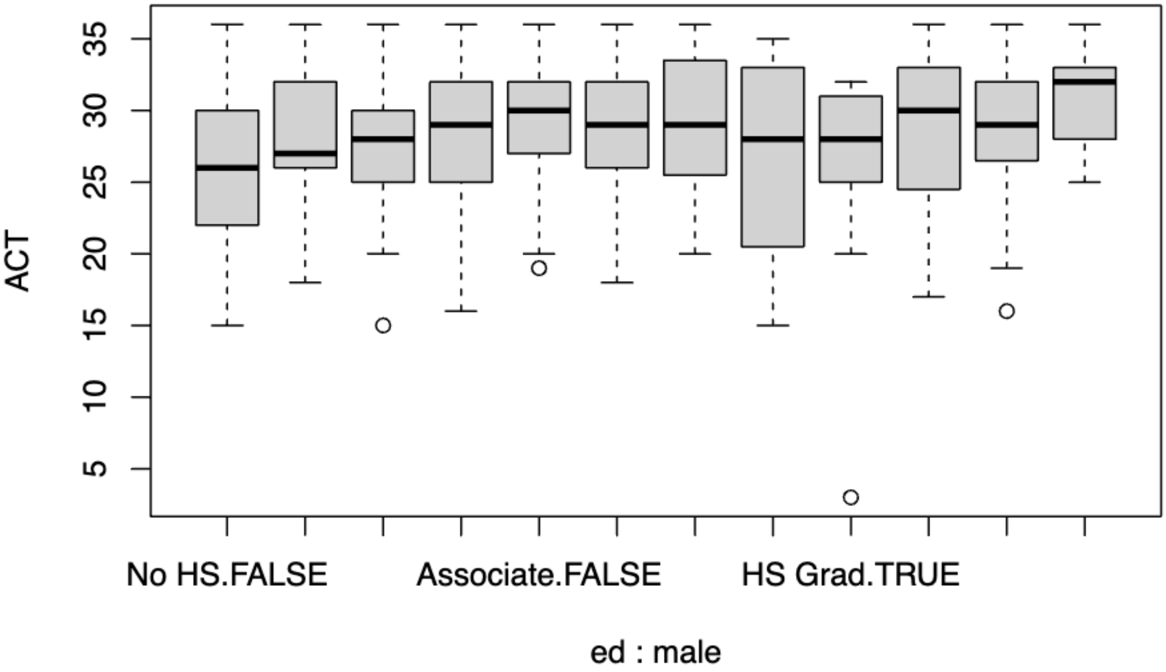
```
# One-way ANOVA for ACT scores by gender
anova_result <- aov(ACT ~ male, data = sat.act)
summary(anova_result)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## male          1     22   21.67   0.931  0.335
## Residuals   698  16242   23.27
```

Conclusion:

According to the summary of the model, the p_value is 0.335 which is higher than the significance level 0.05. We fail to reject the null hypothesis and we can conclude that there is an effect of gender on ACT Scores

```r
boxplot(ACT ~ ed + male, data = sat.act)
```



b) Use anova to investigate if the mean ACT is the same across factors of gender and education.

Hypothesis:

1. Main effect on Gender:

Null Hypothesis ($H0_1$): The mean ACT score is the same for all genders. Alternative Hypothesis ($H1_1$): The mean ACT score is different for at least one gender

2. Main effect of Education:

Null Hypothesis ($H0_2$): The mean ACT Score is the same for all education levels Alternative Hypothesis ($H1_2$): The mean ACT score is different for at least one education level.

3. Interaction Effect of Gender and Education :

Null Hypothesis ($H0_3$): There is no interaction effect between gender and education on the ACT scores. Alternative Hypothesis ($H1_3$): There is an interaction effect between gender and education on the ACT scores. Test:

```r
# Two-way ANOVA for ACT scores by gender and education
anova_result <- aov(ACT ~ male + ed + male:ed, data = sat.act)
summary(anova_result)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)    
## male          1     22   21.67   0.961 0.327401    
## ed            5    489   97.84   4.336 0.000682 ***
## male:ed       5    230   46.01   2.039 0.071254 .  
## Residuals   688  15523   22.56                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Conclusion:

The p-value for Gender (male) is 0.327401. This value is higher than the significance level 0.05. We fail to reject the null hypothesis and conclude that the mean ACT score is different for at least one gender.

The p-value for Education (ed) is 0.000682. This value is lower than the significance level 0.05. We reject the null hypothesis and conclude that the mean ACT Score is the same for all education levels.

The p-value for interaction term (male:ed) is 0.071254. This value is higher than the significance level 0.05. We fail to reject the null hypothesis and conclude that their is an interaction effect between gender and education on the ACT scores.