STA312Exam2TakeHomePortion

Rickey Huang

3/27/2022

Question 1

Permutation Tests (using the prostate data from the faraway package):

```
library(faraway)
prostate <- faraway::prostate
```

Part a

Conduct a 5% significance permutation test to determine whether $y = lpsa$ is correlated with $x = lcp$.

The following code helps us to conduct a permutation test on the varaible lcp

```
# Set seed to fix the randomization
set.seed(2022)
lm1 <- lm(lpsa~lcp, data = prostate)
# Compute the original F statistic
forg <- summary(lm1)$fstat
# Initialize the p-value
pval=0
# Create the for loop
for (i in 1:4000){
  # Fit the model with the permuting
  lmnew <- lm(lpsa~sample(lcp), data = prostate)
  # Find out whether the F stistic is bigger
  if(summary(lmnew)$fstat > forg){
    # if bigger, add it to the p value
    pval=pval+1/4000
  }
}
# return the p-value
pval
```

```
## [1] 0
```

From the p-value the permutation test gave us which is $0 < 0.05$, we are more than 95% confident that *lpsa* is correlated to *lcp*.

Part b

Conduct a 5% significance permutation test to determine whether $y = lpsa$ is correlated with lcp, controlling for lcavol.

```
# Set seed to fix the randomization
set.seed(2022)
```

```r
lm2 <- lm(lpsa~lcp+lcavol, data = prostate)
# Compute the original F statistic
forg2 <- summary(lm2)$fstat
# Initialize the p-value
pval2=0
# Create the for loop
for (i in 1:4000){
  # Fit the model with the permuting
  lmnew2 <- lm(lpsa~sample(lcp)+lcavol, data = prostate)
  # Find out whether the F stistic is bigger
  if(summary(lmnew2)$fstat > forg2){
    # if bigger, add it to the p value
    pval2 <- pval2+1/4000
  }
}
# return the p-value
pval2
```

```
## [1] 0.307
```

From the p-value the permutation test gave us which is $0.307 > 0.05$, we are 95% confident that *lpsa* is not correlated to *lcp*, controlling for *lcavol*.

Part c

Conduct a 5% significance permutation test to determine whether $y = lpsa$ is jointly correlated with lcp and lcavol.

```r
# Set seed to fix the randomization
set.seed(2022)
lm3 <- lm(lpsa~lcp+lcavol, data = prostate)
# Compute the original F statistic
forg3 <- summary(lm2)$fstat
# Initialize the p-value
pval3=0
# Create the for loop
for (i in 1:4000){
  # Fit the model with the permuting
  lmnew3 <- lm(lpsa~sample(lcp)+sample(lcavol), data = prostate)
  # Find out whether the F stistic is bigger
  if(summary(lmnew3)$fstat > forg3){
    # if bigger, add it to the p value
    pval3 <- pval3+1/4000
  }
}
# return the p-value
pval3
```

```
## [1] 0
```

From the p-value the permutation test gave us which is $0 < 0.05$, we are more than 95% confident that *lpsa* is correlated to *lcp* and *lcavol* jointly.

Question 2

Autocorrelation (using the airpass data from the faraway package):

```
airpass <- faraway::airpass
```

Part a

Make a linear model to predict the next value of pass using the current value of pass and the three previous values.

```
# Find the number of data in the dataset
nn <- dim(airpass)[1]
# Store the response variable and explanatory varaibles
y <- airpass$pass[seq(5,nn)]
x1 <- airpass$pass[seq(4,nn-1)]
x2 <- airpass$pass[seq(3,nn-2)]
x3 <- airpass$pass[seq(2,nn-3)]
x4 <- airpass$pass[seq(1,nn-4)]
# Fit the linear model
lmauto <- lm(y~x1+x2+x3+x4)
# Return the model
summary(lmauto)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -81.394 -20.323  -6.093  19.226  78.981
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.495835   7.082015   1.764 0.079918 .
## x1           1.322642   0.086490  15.292  < 2e-16 ***
## x2          -0.509401   0.146236  -3.483 0.000668 ***
## x3           0.005615   0.146893   0.038 0.969565
## x4           0.145008   0.088235   1.643 0.102623
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31.38 on 135 degrees of freedom
## Multiple R-squared:  0.932,  Adjusted R-squared:   0.93
## F-statistic: 462.8 on 4 and 135 DF,  p-value: < 2.2e-16
```

Let y be the next value of pass, set x_1 be the current value of pass, x_2 be the previous value of pass, x_3 be the second previous value of pass, and x_4 be the third previous value of pass. From the code above, I got the model

$$\hat{y} = 12.495835 + 1.322642 \cdot x_1 - 0.509401 \cdot x_2 + 0.005615 \cdot x_3 + 0.145008 \cdot x_4 \tag{1}$$

Part b

Make a 99% prediction interval for the value of pass when year $= 60.08333$ *using that model.*

```r
# Create the input
x <- t(cbind(1, airpass$pass[132], airpass$pass[131], airpass$pass[130], airpass$pass[129]))
# Find the predicted value
betas <- lmauto$coefficients
betas
```

```
##  (Intercept)           x1           x2           x3           x4
## 12.495835098  1.322641641 -0.509401361  0.005614992  0.145007982
```

```r
pred <-t(x)%*%betas
pred
```

```
##          [,1]
## [1,] 433.1864
```

```r
# Find the tstar with 99% CL
X <- model.matrix(lmauto)
n <- dim(X)[1]
p <- dim(X)[2]
tstar <- qt(0.995,n-p)
# Find the sigmahat
SSE <- (t(lmauto$residuals)%*%lmauto$residuals)[1]
sigmasqhat=SSE/(n-p)
sigmahat=sigmasqhat^.5
# Find the prediction interval
ME=sigmahat*tstar*sqrt(1+t(x)%*%solve(t(X)%*%X)%*%x)
predint <- pred + c(-1,1)*ME
predint
```

```
## [1] 348.4243 517.9485
```

Checking from the dataset, we know that the row has $year = 60.08333$ is 133, Then we use the pass value from rows 132, 131, 130 and 129 to predict the pass value when $year = 60.08333$. From the code above, our prediction for the value of pass is 433.1864, and the prediction interval is $(348.4243, 517.9485)$.

Question 3

Boot-strapping (Undergraduate only) (using the prostate data from the faraway pack-age):

```r
prostate <- faraway::prostate
```

Part a

Make a 90% *boot-strapped confidence interval for the coefficient on lcp in the model* $lpsa \sim lcavol + lcp$.

```r
# Set seed to fix the randomization
set.seed(2022)
lm4 <- lm(lpsa~lcavol+lcp, data = prostate)
# Find the residuals and fitted value of the original model
resids <- lm4$residuals
fit <- lm4$fitted
# Construct a vector with 4000 entries
betalcps <- numeric(4000)
# Construct a for loop to find the boot-strapped interval
for (i in 1:4000){
  # Randomize the residuals to create 4000 random response variables with errors
```

```
  fitboot <- fit + sample(resids,rep = TRUE)
  # Fit new models with 4000 error terms
  lmboot <- lm(fitboot ~ lcavol + lcp, data = prostate)
  # Store the betas for the variable sex
  betalcps[i] <- lmboot$coeff[3]
}
# Find the 0.025 and 0.975 quantile of the betas to construct a boot-strapped confidence interval
cbind(quantile(betalcps,0.05),quantile(betalcps,0.95))
```

```
##            [,1]      [,2]
## 5% -0.04648054 0.2012478
```

I used the above code to construct a 90% boot-strapped confidence interval for the coefficient on *lcp* in the model. From the result R gave me, the 90% boot-strapped confidence interval is $(-0.04648054, 0.2012478)$.

Part b

Make a 90% *boot-strapped prediction interval for an individual with* $lcp = -1$ *and* $lcavol = 2$ *with the model* $lpsa \sim lcavol + lcp$.

```
# Set seed to fix the randomization
set.seed(2022)
lm5 <- lm(lpsa~lcavol+lcp, data = prostate)
# Find the residuals and fitted value of the original model
resids5 <- lm5$residuals
fit5 <- lm5$fitted
# Construct a vector with 4000 entries
lpsas <- numeric(4000)
# Create the input
x0 <- cbind(1,2,-1)
# Construct a for loop to find the boot-strapped interval
for (i in 1:4000){
  # Randomize the residuals to create 4000 random response variables with errors
  fitboot5 <- fit5 + sample(resids5,rep = TRUE)
  # Fit new models with 4000 error terms
  lmboot5 <- lm(fitboot5 ~ lcavol + lcp, data = prostate)
  # Use the new models make predictions
  lpsas[i] <- x0%*%lmboot5$coefficients
}
# Find the 0.025 and 0.975 quantile of the betas to construct a boot-strapped confidence interval
cbind(quantile(lpsas,0.05),quantile(lpsas,0.95))
```

```
##        [,1]     [,2]
## 5% 2.620092 3.065774
```

I used the above code to construct a 90% boot-strapped prediction interval for an individual with $lcp = -1$, and $lcavol = 2$. From the result R gave me, the 90% boot-strapped prediction interval for this individual is $(2.620092, 3.065774)$.