Intro to Statistical Computing HW 3

Sarah Lotspeich

11 October 2016

Grade: 45/50

JC Grading -3 Hi Sarah, just a heads up that this was the 2015 assignment. Fortunately, this assignment was similar to 2016 (though the data was slightly different). This may not always be the case for other assignments.

Question 1

Use GitHub to turn in the first three homework assignments. Make sure the teacher (couthcommander) and TA (trippcm) are collaborators. (5 points)

Commit each assignment individually. This means your repository should have at least three commits. (5 points)

Question 2

Write a simulation to calculate the power for the following study design.

The study has two variables, treatment group and outcome.

```r
treatment <- NULL
outcome <- NULL
n <- 100 #sample size
```

There are two treatment groups (0, 1) and they should be assigned randomly with equal probability.

```r
treatment <- sample(x = seq(0,1),size = n, replace=TRUE)
```

The outcome should be a random normal variable with a mean of 60 and standard deviation of 20.

```r
outcome <- rnorm(n, 60,20)
studyDesign <- data.frame(cbind(treatment,outcome))
```

If a patient is in the treatment group, add 5 to the outcome. 5 is the true treatment effect.

```r
studyDesign[studyDesign[,1]==1,2] <- studyDesign[studyDesign[,1]==1,2]+5
```

Create a linear of model for the outcome by the treatment group, and extract the p-value (hint: see assigment1).

```r
mod1 <- lm(outcome~treatment, data=studyDesign)
pVal <- coef(summary(mod1))[2,4] #extract the p-value for treatment from the model
```

Test if the p-value is less than or equal to the alpha level, which should be set to 0.05.

```r
pVal <= 0.05
```

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

Repeat this procedure 1000 times. The power is calculated by finding the percentage of times the p-value is less than or equal to the alpha level. Use the set.seed command so that the professor can reproduce your results.

Find the power when the sample size is 100 patients. (10 points)

```
set.seed(1000)
runExperiment <- function(n)
{
  treatment <- sample(x = seq(0,1),size = n, replace=TRUE)
  outcome <- rnorm(n, 60,20)
  studyDesign <- data.frame(cbind(treatment,outcome))
  studyDesign[studyDesign[,1]==1,2] <- studyDesign[studyDesign[,1]==1,2]+5
  mod1 <- lm(outcome~treatment, data=studyDesign)
  pVal <- coef(summary(mod1))[2,4] #extract the p-value for treatment from the model
  return(pVal < 0.05) #return boolean TRUE if p-val <= 0.05, FALSE otherwise
}

n <- 100 #sample size
N <- 1000 #run experiment N times
pValues <- NULL
for (i in 1:N)
{
  pValues[i] <- runExperiment(n)
}

power <- mean(pValues)
```

JC Grading -1 Please print out/report result of power.

Find the power when the sample size is 1000 patients. (5 points)

```
set.seed(1000)
n <- 1000 #sample size
N <- 1000 #run experiment N times
pValues <- NULL
for (i in 1:N)
{
  pValues[i] <- runExperiment(n)
}

power <- mean(pValues)
```

JC Grading -1 Please print out/report result of power.

Question 3

Obtain a copy of the football-values lecture. Save the 2015/proj_rb15.csv file in your working directory. Read in the data set and remove the first two columns.

```
proj_rb15 <- read.csv("https://raw.githubusercontent.com/couthcommander/football-values/master/2015/pro
    head = TRUE)
proj_rb15$PlayerName <- NULL  #remove first column
proj_rb15$Team <- NULL  #remove second column
```

Show the correlation matrix of this data set. (3 points)

```
cor(proj_rb15)
```

```
##            rush_att  rush_yds  rush_tds   rec_att   rec_yds   rec_tds
## rush_att 1.0000000 0.9975511 0.9723599 0.7694384 0.7402687 0.5969159
## rush_yds 0.9975511 1.0000000 0.9774974 0.7645768 0.7345496 0.6020994
## rush_tds 0.9723599 0.9774974 1.0000000 0.7263519 0.6984860 0.5908348
## rec_att  0.7694384 0.7645768 0.7263519 1.0000000 0.9944243 0.8384359
## rec_yds  0.7402687 0.7345496 0.6984860 0.9944243 1.0000000 0.8518924
## rec_tds  0.5969159 0.6020994 0.5908348 0.8384359 0.8518924 1.0000000
## fumbles  0.8589364 0.8583243 0.8526904 0.7459076 0.7224865 0.6055598
## fpts     0.9824135 0.9843044 0.9689472 0.8556928 0.8340195 0.7133908
##            fumbles      fpts
## rush_att 0.8589364 0.9824135
## rush_yds 0.8583243 0.9843044
## rush_tds 0.8526904 0.9689472
## rec_att  0.7459076 0.8556928
## rec_yds  0.7224865 0.8340195
## rec_tds  0.6055598 0.7133908
## fumbles  1.0000000 0.8635550
## fpts     0.8635550 1.0000000
```

Generate a data set with 30 rows that has a similar correlation structure. Repeat the procedure 10,000 times and return the mean correlation matrix. (10 points)

```r
library(MASS)
#start with summary statistics
rho.fb <- cor(proj_rb15)
vcov.fb <- var(proj_rb15)
means.fb <- colMeans(proj_rb15)

#generate a data set with 30 rows with similar correlation structure
fb.sim <- mvrnorm(30, mu = means.fb, Sigma = vcov.fb)

#repeat the procedure 10,000 times and return mean correlation matrix
avgCor <- 0
loops <- 10000
for (i in seq(loops))
{
  fb.sim <- mvrnorm(30, mu = means.fb, Sigma = vcov.fb)
  avgCor <- avgCor + cor(fb.sim)/loops
}
avgCor
```

```
##           rush_att  rush_yds  rush_tds   rec_att   rec_yds   rec_tds
## rush_att 1.0000000 0.9974622 0.9715708 0.7643381 0.7348544 0.5911409
## rush_yds 0.9974622 1.0000000 0.9768387 0.7593608 0.7290245 0.5961377
## rush_tds 0.9715708 0.9768387 1.0000000 0.7210280 0.6929934 0.5845818
## rec_att  0.7643381 0.7593608 0.7210280 1.0000000 0.9941969 0.8340694
## rec_yds  0.7348544 0.7290245 0.6929934 0.9941969 1.0000000 0.8477925
## rec_tds  0.5911409 0.5961377 0.5845818 0.8340694 0.8477925 1.0000000
## fumbles  0.8548809 0.8541038 0.8481028 0.7397535 0.7160232 0.5980345
## fpts     0.9818100 0.9837319 0.9679676 0.8519626 0.8299069 0.7077571
##            fumbles      fpts
## rush_att 0.8548809 0.9818100
## rush_yds 0.8541038 0.9837319
## rush_tds 0.8481028 0.9679676
## rec_att  0.7397535 0.8519626
## rec_yds  0.7160232 0.8299069
## rec_tds  0.5980345 0.7077571
## fumbles  1.0000000 0.8591682
## fpts     0.8591682 1.0000000
```

```
rho.fb
```

```
##           rush_att  rush_yds  rush_tds   rec_att   rec_yds   rec_tds
## rush_att 1.0000000 0.9975511 0.9723599 0.7694384 0.7402687 0.5969159
## rush_yds 0.9975511 1.0000000 0.9774974 0.7645768 0.7345496 0.6020994
## rush_tds 0.9723599 0.9774974 1.0000000 0.7263519 0.6984860 0.5908348
## rec_att  0.7694384 0.7645768 0.7263519 1.0000000 0.9944243 0.8384359
## rec_yds  0.7402687 0.7345496 0.6984860 0.9944243 1.0000000 0.8518924
## rec_tds  0.5969159 0.6020994 0.5908348 0.8384359 0.8518924 1.0000000
## fumbles  0.8589364 0.8583243 0.8526904 0.7459076 0.7224865 0.6055598
## fpts     0.9824135 0.9843044 0.9689472 0.8556928 0.8340195 0.7133908
##            fumbles      fpts
## rush_att 0.8589364 0.9824135
## rush_yds 0.8583243 0.9843044
## rush_tds 0.8526904 0.9689472
## rec_att  0.7459076 0.8556928
## rec_yds  0.7224865 0.8340195
## rec_tds  0.6055598 0.7133908
## fumbles  1.0000000 0.8635550
## fpts     0.8635550 1.0000000
```

Generate a data set with 30 rows that has the exact correlation structure as the original data set. (2 points)

```
fb.sim <- mvrnorm(30, mu = means.fb, Sigma = vcov.fb, empirical=TRUE)
cor(fb.sim)
```

```
##           rush_att  rush_yds  rush_tds   rec_att   rec_yds   rec_tds
## rush_att 1.0000000 0.9975511 0.9723599 0.7694384 0.7402687 0.5969159
## rush_yds 0.9975511 1.0000000 0.9774974 0.7645768 0.7345496 0.6020994
## rush_tds 0.9723599 0.9774974 1.0000000 0.7263519 0.6984860 0.5908348
## rec_att  0.7694384 0.7645768 0.7263519 1.0000000 0.9944243 0.8384359
## rec_yds  0.7402687 0.7345496 0.6984860 0.9944243 1.0000000 0.8518924
## rec_tds  0.5969159 0.6020994 0.5908348 0.8384359 0.8518924 1.0000000
## fumbles  0.8589364 0.8583243 0.8526904 0.7459076 0.7224865 0.6055598
## fpts     0.9824135 0.9843044 0.9689472 0.8556928 0.8340195 0.7133908
##            fumbles      fpts
## rush_att 0.8589364 0.9824135
## rush_yds 0.8583243 0.9843044
## rush_tds 0.8526904 0.9689472
## rec_att  0.7459076 0.8556928
## rec_yds  0.7224865 0.8340195
## rec_tds  0.6055598 0.7133908
## fumbles  1.0000000 0.8635550
## fpts     0.8635550 1.0000000
```

```
rho.fb
```

```
##           rush_att  rush_yds  rush_tds   rec_att   rec_yds   rec_tds
## rush_att 1.0000000 0.9975511 0.9723599 0.7694384 0.7402687 0.5969159
## rush_yds 0.9975511 1.0000000 0.9774974 0.7645768 0.7345496 0.6020994
## rush_tds 0.9723599 0.9774974 1.0000000 0.7263519 0.6984860 0.5908348
## rec_att  0.7694384 0.7645768 0.7263519 1.0000000 0.9944243 0.8384359
## rec_yds  0.7402687 0.7345496 0.6984860 0.9944243 1.0000000 0.8518924
## rec_tds  0.5969159 0.6020994 0.5908348 0.8384359 0.8518924 1.0000000
## fumbles  0.8589364 0.8583243 0.8526904 0.7459076 0.7224865 0.6055598
## fpts     0.9824135 0.9843044 0.9689472 0.8556928 0.8340195 0.7133908
##            fumbles      fpts
## rush_att 0.8589364 0.9824135
## rush_yds 0.8583243 0.9843044
## rush_tds 0.8526904 0.9689472
## rec_att  0.7459076 0.8556928
## rec_yds  0.7224865 0.8340195
## rec_tds  0.6055598 0.7133908
## fumbles  1.0000000 0.8635550
## fpts     0.8635550 1.0000000
```

Question 4

Use LaTeXto create the following expressions.

1.

$$P(B) = \underset{j}{\Sigma} P(B|A_i)P(A_i),$$
$$\rightarrow P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\Sigma_j P(B|A_j)P(A_j)}$$

2.

$$\hat{f}(\zeta) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \zeta} dx$$

3.

$$\mathbf{J} = \frac{d\mathbf{f}}{d\mathbf{x}} = [\frac{\partial \mathbf{f}}{\partial x_1} \cdots \frac{\partial \mathbf{f}}{\partial x_n}] = \begin{bmatrix} \frac{\partial \mathbf{f}_1}{\partial x_1} & \cdots & \frac{\partial \mathbf{f}_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{f}_m}{\partial x_1} & \cdots & \frac{\partial \mathbf{f}_m}{\partial x_n} \end{bmatrix}$$