Computing Assignment 3

Angela Huynh

October 11, 2016

Grade 47/50

Question 1

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

Commit each assignment individually. This means your repository should have at least three commits.

Question 2

Write a simulation to calculate the power for the following study design. The study has two variables, treatment group and outcome. There are two treatment groups (0, 1) and they should be assigned randomly with equal probability. The outcome should be a random normal variable with a mean of 60 and standard deviation of 20. If a patient is in the treatment group, add 5 to the outcome. 5 is the true treatment effect. Create a linear model for the outcome by the treatment group, and extract the p-value (hint: see assigment1, binomial). Test if the p-value is less than or equal to the alpha level, which should be set to 0.05.

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.Find the power when the sample size is 1000 patients.

```
set.seed(100)  # set seed to replicate outcome
n <- 1000  # number of times we repeat procedure
alpha <- 0.05
pvalue.1 <- vector()  # use this vector to find the power when n = 100
pvalue.2 <- vector()  # use this vector to find the power when n = 1000

for (i in (1:n)) {
    treatment <- rbinom(n = 100, size = 1, prob = 0.5)
    outcome <- rnorm(n = 100, mean = 60, sd = 20)
    for (a in (1:100)) {
        if (treatment[a] == 1) {
            outcome[a] <- outcome[a] + 5
        }
    }
    sim <- lm(outcome ~ treatment)
    pvalue <- coef(summary(sim))[2, 4]
    pvalue.1 <- c(pvalue.1, pvalue <= 0.05)
}

mean(pvalue.1)  # power when sample size is 100
```

```
## [1] 0.236
```

```r
for (i in (1:n)) {
    treatment1k <- rbinom(n = 1000, size = 1, prob = 0.5)
    outcome1k <- rnorm(n = 1000, mean = 60, sd = 20)
    for (a in (1:1000)) {
        if (treatment1k[a] == 1) {
            outcome1k[a] <- outcome1k[a] + 5
        }
    }
    sim1k <- lm(outcome1k ~ treatment1k)
    pvalue1k <- coef(summary(sim1k))[2, 4]
    pvalue.2 <- c(pvalue.2, pvalue1k <= 0.05)
}

mean(pvalue.2)  # power when sample size is 1000
```

```
## [1] 0.977
```

Question 3

Obtain a copy of the football-values lecture. Save the 2016/proj_wr16.csv file in your working directory. Read in the data set and remove the first two columns.

Show the correlation matrix of this data set.

```r
# setwd('~/Documents/Vanderbilt 1/Semester 1/BIOS 6301/Assignments')
# football.data <- read.csv('~/Documents/Vanderbilt 1/Semester 1/BIOS
# 6301/Assignments/proj_wr16.csv')
football.data <- read.csv("proj_wr16.csv")
# View(football.data)

# remove first two columns
fb.data <- football.data[-c(1:2)]

# correlation matrix of data set
rho.fb <- cor(fb.data)

vcov.fb <- var(fb.data)
means.fb <- colMeans(fb.data)
```

JC Grading -1 Please print the correlation matrix.

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.

```r
library(MASS)

loops <- 10000
avCor <- 0

for (i in (1:10000)) {
    fb.sim <- mvrnorm(30, mu = means.fb, Sigma = vcov.fb)
    avCor <- avCor + cor(fb.sim)/loops
}
avCor  # mean correlation matrix, similar structure
```

```
##           rush_att  rush_yds   rush_tds    rec_att    rec_yds    rec_tds
## rush_att 1.0000000 0.9902059 0.88256933 0.19056014 0.13915277 0.13064846
## rush_yds 0.9902059 1.0000000 0.90980378 0.18113123 0.13223953 0.12305955
## rush_tds 0.8825693 0.9098038 1.00000000 0.06502175 0.02785182 0.02879898
## rec_att  0.1905601 0.1811312 0.06502175 1.00000000 0.98969081 0.96648254
## rec_yds  0.1391528 0.1322395 0.02785182 0.98969081 1.00000000 0.98144258
## rec_tds  0.1306485 0.1230596 0.02879898 0.96648254 0.98144258 1.00000000
## fumbles  0.1793917 0.1834451 0.10634702 0.42873783 0.39690593 0.35283177
## fpts     0.1707787 0.1641311 0.06196741 0.98712743 0.99750896 0.99024287
##            fumbles       fpts
## rush_att 0.1793917 0.17077867
## rush_yds 0.1834451 0.16413110
## rush_tds 0.1063470 0.06196741
## rec_att  0.4287378 0.98712743
## rec_yds  0.3969059 0.99750896
## rec_tds  0.3528318 0.99024287
## fumbles  1.0000000 0.37643430
## fpts     0.3764343 1.00000000
```

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

```r
loops <- 10000
avCorExact <- 0

for (i in (1:10000)) {
    fb.sim <- mvrnorm(30, mu = means.fb, Sigma = vcov.fb, empirical = TRUE)
    avCorExact <- avCorExact + cor(fb.sim)/loops
}
avCorExact  # mean correlation matrix, exact structure
```

```
##           rush_att  rush_yds   rush_tds    rec_att    rec_yds    rec_tds
## rush_att 1.0000000 0.9906030 0.88608205 0.19706851 0.14473723 0.13548999
## rush_yds 0.9906030 1.0000000 0.91252627 0.18745520 0.13765791 0.12772327
## rush_tds 0.8860820 0.9125263 1.00000000 0.06914613 0.03114206 0.03163468
## rec_att  0.1970685 0.1874552 0.06914613 1.00000000 0.99002712 0.96757796
## rec_yds  0.1447372 0.1376579 0.03114206 0.99002712 1.00000000 0.98209522
## rec_tds  0.1354900 0.1277233 0.03163468 0.96757796 0.98209522 1.00000000
## fumbles  0.1844220 0.1881021 0.10845675 0.43577978 0.40349289 0.35852435
## fpts     0.1766540 0.1698501 0.06567865 0.98754942 0.99760259 0.99058639
##            fumbles       fpts
## rush_att 0.1844220 0.17665405
## rush_yds 0.1881021 0.16985010
## rush_tds 0.1084568 0.06567865
## rec_att  0.4357798 0.98754942
## rec_yds  0.4034929 0.99760259
## rec_tds  0.3585244 0.99058639
## fumbles  1.0000000 0.38269698
## fpts     0.3826970 1.00000000
```

JC Grading -2 Looking for a single generated dataset.

Question 4

Use LaTeXto create the following expressions.

1.

$$P(B) = \sum_j P(B|A_j)P(A_j), \Rightarrow P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_j P(B|A_j)P(A_j)}$$

2.

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

3.

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