BIOS6301: Assignment 3

Elizabeth Sigworth

due October 10, 2016

Due Tuesday, 11 October, 1:00 PM

50 points total.

$5^{n=day}$ points taken off for each day late.

This assigment includes turning in the first two assignments. All three should include knitr files (named `homework1.rmd`, `homework2.rmd`, `homework3.rmd`) along with valid PDF output files. Inside each file, clearly indicate which parts of your responses go with which problems (you may use the original homework document as a template). Add your name as `author` to the file's metadata section. Raw R code/output or word processor files are not acceptable.

Failure to properly name files or include author name may result in 5 points taken off.

Question 1

10 points

1. Use GitHub to turn in the first three homework assignments. Make sure the teacher (couthcommander) and TA (trippcm) are collaborators. (5 points)
2. Commit each assignment individually. This means your repository should have at least three commits. (5 points)

Question 2

15 points

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). 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.

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

```
set.seed(872)
n <-100 ##Set number of patients
pvalues.1 <- NULL
for (i in 1:1000) {
  simul <- data.frame(Treatment=rbinom(100,1,.5))
  simul$Outcome <- rnorm(100,60,20)
  for (j in 1:n) {
```

```
    if (simul$Treatment[j]==1) {
      simul$Outcome[j] <- simul$Outcome[j] + 5
    }
  }
  mod <- lm(simul$Outcome~simul$Treatment)
  p.val <- coef(summary(mod))[2,4]
  pvalues.1 <- c(pvalues.1,p.val <=0.05)
}
power.1 <- mean(pvalues.1)
power.1
```

```
## [1] 0.228
```

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

```
n <- 1000
set.seed(5424)
pvalues.2 <- NULL
for (i in 1:1000) {
  simul <- data.frame(Treatment=rbinom(1000,1,.5))
  simul$Outcome <- rnorm(1000,60,20)
  for (j in 1:n) {
    if (simul$Treatment[j]==1) {
      simul$Outcome[j] <- simul$Outcome[j] + 5
    }
  }
  mod <- lm(simul$Outcome~simul$Treatment)
  p.val <- coef(summary(mod))[2,4]
  pvalues.2 <- c(pvalues.2,p.val <=0.05)
}
power.2 <- mean(pvalues.2)
power.2
```

```
## [1] 0.97
```

When the sample size is 100 patients, the power of the study is 22.8%, but when the sample size is 1000 patients, the power of the study is 97%.

Question 3

15 points

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("/var/folders/kp/zlsf12h14y92__lp668ljv2m0000gn/T//RtmpKa1QwJ/data2c845e674976", s
proj_rb15 <- proj_rb15[,3:ncol(proj_rb15)]
```

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

```r
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
```

2. 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)
rho.proj <- cor(proj_rb15)
vcov.proj <- var(proj_rb15)
means.proj <- colMeans(proj_rb15)

average.cor <- 0
for(i in 1:10000) {
  proj.sim <- mvrnorm(30, mu=means.proj, Sigma=vcov.proj)
  rho.sim <- cor(proj.sim)
  average.cor <- average.cor + rho.sim/10000
}
average.cor # similar correlation in the mean correlation matrix
```

```
##           rush_att  rush_yds  rush_tds   rec_att   rec_yds   rec_tds
## rush_att 1.0000000 0.9974374 0.9710432 0.7623527 0.7326401 0.5879537
## rush_yds 0.9974374 1.0000000 0.9764304 0.7572944 0.7267313 0.5932030
## rush_tds 0.9710432 0.9764304 1.0000000 0.7181594 0.6898969 0.5819064
## rec_att  0.7623527 0.7572944 0.7181594 1.0000000 0.9941804 0.8334221
## rec_yds  0.7326401 0.7267313 0.6898969 0.9941804 1.0000000 0.8473904
## rec_tds  0.5879537 0.5932030 0.5819064 0.8334221 0.8473904 1.0000000
## fumbles  0.8537527 0.8531723 0.8476081 0.7385830 0.7147849 0.5970675
## fpts     0.9816180 0.9835811 0.9675043 0.8506045 0.8283821 0.7058570
##            fumbles      fpts
## rush_att 0.8537527 0.9816180
## rush_yds 0.8531723 0.9835811
## rush_tds 0.8476081 0.9675043
## rec_att  0.7385830 0.8506045
## rec_yds  0.7147849 0.8283821
```

```
## rec_tds  0.5970675 0.7058570
## fumbles  1.0000000 0.8586232
## fpts     0.8586232 1.0000000
```

```
cor(proj_rb15) # original correlation matrix
```

```
##           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
```

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

```
exact.avg.cor <- 0
for(i in 1:10000) {
  proj.sim <- mvrnorm(30, mu=means.proj, Sigma=vcov.proj, empirical=TRUE)
  rho.sim <- cor(proj.sim)
  exact.avg.cor <- exact.avg.cor + rho.sim/10000
}
exact.avg.cor # exact correlation structure as original
```

```
##           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
```

```r
cor(proj_rb15) # original correlation
```

```
##           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

10 points

Use LaTeXto create the following expressions.

1. Hint: `\Rightarrow` (4 points)

$$\begin{aligned} P(B) &= \sum_j P(B|A_j)P(A_j), \\ &\Rightarrow \quad P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_j (B|A_j)P(A_j)} \end{aligned}$$

2. Hint: `\zeta` (3 points)
   $\hat{f}(\zeta) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i x \zeta} dx$

3. Hint: `\partial` (3 points)

$$\mathbf{J} = \frac{d\mathbf{f}}{d\mathbf{x}} = \left[\frac{\partial \mathbf{f}}{\partial x_1} \cdots \frac{\partial \mathbf{f}}{\partial x_n} \right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}$$