Bios 6301: Assignment 3

Hannah Weeks

Due Thursday, 08 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 of 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.

```
#Function to be used for determining p-values
power.sim <- function(sampleSize, alpha = .05){
  #Create vector of random 1s and 0s to assign subjects to treatment or control, respectively
  assignment <- rbinom(n = sampleSize, size = 1, prob = .5)
  #Define outcome
  outcome <- rnorm(n = sampleSize, mean = 60, sd = 20)
  #Define treatment effect
  treatmentOutcome <- outcome + 5
  #If subject in treatment group, update outcome to include treatment effect
  outcomeEffects <- treatmentOutcome*assignment + outcome*(1-assignment)
  #Create linear model comparing outcome effects based on treatment group
  model <- lm(outcomeEffects ~ assignment)
  #Extract p-values
  pval <- summary(model)$coefficients[2,4]
  return(pval)
}
```

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 for replication of results
set.seed(5)

nSims <- 1000
alpha <- .05
sig.exp <- rep(NA, nSims)

#Run simulation 1000 times for n = 100
for(i in 1:nSims){
  #Enter T (or F) for that exp being significant if p-value is less than alpha (or not)
  sig.exp[i] <- (power.sim(sampleSize = 100) < alpha)
}
#Determine average number of significant experiments
power <- mean(sig.exp)
power
```

```
## [1] 0.212
```

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

```
#Set seed for replication of results
set.seed(5)

nSims <- 1000
alpha <- .05
sig.exp <- rep(NA, nSims)

#Run simulation 1000 times for n = 1000
for(i in 1:nSims){
  #Enter T (or F) for that exp being significant if p-value is less than alpha (or not)
  sig.exp[i] <- (power.sim(sampleSize = 1000) < alpha)
}
#Determine average number of significant experiments
power <- mean(sig.exp)
power
```

```
## [1] 0.971
```

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.

```r
setwd("~/Documents/BIOS 6301/Homework")
rb <- read.csv("proj_rb15.csv")
#See what first two columns are
head(rb)
```

```
##        PlayerName Team rush_att rush_yds rush_tds rec_att rec_yds rec_tds
## 1  Jamaal Charles   KC    232.4   1116.5      9.2    52.0   436.4     4.5
## 2  Marshawn Lynch  SEA    278.1   1308.6      9.8    37.8   326.0     2.2
## 3      Eddie Lacy   GB    282.6   1264.5      9.2    45.7   368.2     2.4
## 4 Adrian Peterson  MIN    283.7   1325.2      9.8    43.6   329.8     1.5
## 5    Le'Veon Bell  PIT    243.9   1075.9      7.0    66.4   567.6     2.4
## 6      Matt Forte  CHI    252.6   1062.8      6.5    70.3   585.3     2.6
##   fumbles  fpts
## 1     3.6 229.8
## 2     2.9 229.5
## 3     2.6 227.5
## 4     3.4 226.4
## 5     1.4 218.1
## 6     2.2 214.6
```

```r
names(rb)
```

```
##  [1] "PlayerName" "Team"       "rush_att"   "rush_yds"   "rush_tds"
##  [6] "rec_att"    "rec_yds"    "rec_tds"    "fumbles"    "fpts"
```

```r
#Remove those columns
rb$PlayerName <- NULL
rb$Team <- NULL

#Check to make sure those columns are gone
head(rb)
```

```
##   rush_att rush_yds rush_tds rec_att rec_yds rec_tds fumbles  fpts
## 1    232.4   1116.5      9.2    52.0   436.4     4.5     3.6 229.8
## 2    278.1   1308.6      9.8    37.8   326.0     2.2     2.9 229.5
## 3    282.6   1264.5      9.2    45.7   368.2     2.4     2.6 227.5
## 4    283.7   1325.2      9.8    43.6   329.8     1.5     3.4 226.4
## 5    243.9   1075.9      7.0    66.4   567.6     2.4     1.4 218.1
## 6    252.6   1062.8      6.5    70.3   585.3     2.6     2.2 214.6
```

```r
names(rb)
```

```
## [1] "rush_att" "rush_yds" "rush_tds" "rec_att"  "rec_yds"  "rec_tds"
## [7] "fumbles"  "fpts"
```

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

```r
(cor.rb <- cor(rb))
```

```
##           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)

```
library(MASS)

#Generate data set with 30 rows
rb.sim <- mvrnorm(30, mu = colMeans(rb), Sigma = var(rb))
#Correlation matrix for single iteration of simulated data
cor.sim <- cor(rb.sim)
```

Now repeat the above procedure 10,000 times to obtain a mean correlation matrix:

```
corMatrix <- 0
nsims <- 10000
#Create 10,000 similar correlation matricies and add them together
for(i in seq(nsims)){
  rb.sim <- mvrnorm(30, mu = colMeans(rb), Sigma = var(rb))
  corMatrix <- corMatrix + cor(rb.sim)
}
#Create average correlation matrix
meanMatrix <- corMatrix/nsims
meanMatrix
```

```
##           rush_att  rush_yds  rush_tds   rec_att   rec_yds   rec_tds
## rush_att 1.0000000 0.9974588 0.9713632 0.7634651 0.7338562 0.5896594
## rush_yds 0.9974588 1.0000000 0.9766988 0.7584346 0.7279654 0.5947524
## rush_tds 0.9713632 0.9766988 1.0000000 0.7189759 0.6907402 0.5827179
## rec_att  0.7634651 0.7584346 0.7189759 1.0000000 0.9941889 0.8330550
## rec_yds  0.7338562 0.7279654 0.6907402 0.9941889 1.0000000 0.8468645
## rec_tds  0.5896594 0.5947524 0.5827179 0.8330550 0.8468645 1.0000000
## fumbles  0.8551341 0.8544435 0.8484544 0.7393659 0.7155675 0.5976773
## fpts     0.9818841 0.9838224 0.9677442 0.8509213 0.8287245 0.7062353
##            fumbles      fpts
## rush_att 0.8551341 0.9818841
```

```
## rush_yds 0.8544435 0.9838224
## rush_tds 0.8484544 0.9677442
## rec_att  0.7393659 0.8509213
## rec_yds  0.7155675 0.8287245
## rec_tds  0.5976773 0.7062353
## fumbles  1.0000000 0.8596303
## fpts     0.8596303 1.0000000
```

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

To obtain a data set with the *exact* correlation structure, we are set `empirical=TRUE` in the `mvrnorm()` function. Becuase we are specifying mu and sigma to align with the empirical correlation matrix, it isn't necessary to run this 10,000 times. A single run will produce the desired result:

```
rb.sim <- mvrnorm(30, mu = colMeans(rb), Sigma = var(rb), empirical=TRUE)
exactMatrix <- cor(rb.sim)
exactMatrix
```

```
##           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)

$$P(B) = \sum_j P(B|A_j)P(A_j),$$

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

$$P(B) = \sum_j P(B|A_j)P(A_j),$$

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

2. Hint: `\zeta` (3 points)

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

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

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

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

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