Bios 6301: Assignment 3

Andrea Perreault

Grade: 50/50

Due Tuesday, 11 October, 1:00 PM

50 points total.

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

QUESTION 1

10 points

1. Use GitHub to turn in the first three homework assignments. Make sure the teacher (couthcommander) and TA (chipmanj) 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)

```
patients1 = 100
groups = c(0,1)
mean = 60
sd = 20
alpha = 0.05
p_vals = numeric(1000)
significant1 = 0
set.seed(100)

for (i in seq(p_vals)) {
  treatment <- sample(groups, patients1, replace = TRUE)
  outcome <- rnorm(100, mean, sd)
  data <- data.frame(cbind(treatment, outcome))
  data$outcome[data$treatment == 1] <- data$outcome[data$treatment == 1] + 5
  model <- lm(outcome ~ treatment, data = data)
  pval <- summary(model)$coefficients[2,4]
  p_vals[i] <- pval
  if (pval < alpha) {
    significant1 = significant1 + 1
    #print("TRUE")
```

```
  } else {
    #print("FALSE")
  }
}

significant1
```

```
## [1] 236
```

```
power = significant1 / length(p_vals)
power
```

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

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

```
patients2 = 1000
significant2 = 0

for (i in seq(p_vals)) {
  treatment <- sample(groups, patients2, replace = TRUE)
  outcome <- rnorm(1000, mean, sd)
  data <- data.frame(cbind(treatment, outcome))
  data$outcome[data$treatment == 1] <- data$outcome[data$treatment == 1] + 5
  model <- lm(outcome ~ treatment, data = data)
  pval <- summary(model)$coefficients[2,4]
  p_vals[i] <- pval
  if (pval < alpha) {
    significant2 = significant2 + 1
    #print("TRUE")
  } else {
    #print("FALSE")
  }
}

significant2
```

```
## [1] 977
```

```
power = significant2 / length(p_vals)
power
```

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

QUESTION 3

15 points

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.

```
wr <- read.csv("proj_wr16.csv", header = TRUE, sep = ",")
head(wr)
```

```
##          PlayerName Team rush_att rush_yds rush_tds rec_att rec_yds
## 1     Antonio Brown  PIT      3.1     17.0        0   123.6  1648.8
## 2       Julio Jones  ATL      0.3      1.6        0   116.6  1623.5
## 3 Odell Beckham Jr.  NYG      0.8      4.8        0    98.0  1439.5
## 4   DeAndre Hopkins  HOU      0.0      0.0        0   100.0  1423.2
```

```
## 5        Dez Bryant  DAL      0.0      0.0        0    85.2  1195.1
## 6        A.J. Green  CIN      0.0      0.1        0    87.4  1255.3
##   rec_tds fumbles  fpts
## 1    10.8     1.1 229.1
## 2     8.8     0.8 214.0
## 3    11.1     0.1 210.6
## 4     9.6     0.1 199.5
## 5    10.1     0.1 179.6
## 6     9.3     0.9 179.3
```

```
wr[,1] <- NULL
wr[,1] <- NULL
head(wr)
```

```
##   rush_att rush_yds rush_tds rec_att rec_yds rec_tds fumbles  fpts
## 1      3.1     17.0        0   123.6  1648.8    10.8     1.1 229.1
## 2      0.3      1.6        0   116.6  1623.5     8.8     0.8 214.0
## 3      0.8      4.8        0    98.0  1439.5    11.1     0.1 210.6
## 4      0.0      0.0        0   100.0  1423.2     9.6     0.1 199.5
## 5      0.0      0.0        0    85.2  1195.1    10.1     0.1 179.6
## 6      0.0      0.1        0    87.4  1255.3     9.3     0.9 179.3
```

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

```
cor.wr <- cor(wr)
cor.wr
```

```
##           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
```

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)

```
cor.wr <- cor(wr)
cov.wr <- var(wr)
means.wr <- colMeans(wr)
library(MASS)

wr.sim1 <- mvrnorm(30, mu = means.wr, Sigma = cov.wr, empirical = FALSE)
cor.sim1 <- cor(wr.sim1)
cor.sim1; cor.wr
```

```
##           rush_att   rush_yds     rush_tds    rec_att      rec_yds
## rush_att 1.0000000 0.99290632  0.935544127 0.22364409  0.127641953
## rush_yds 0.9929063 1.00000000  0.950188823 0.23981390  0.147151297
## rush_tds 0.9355441 0.95018882  1.000000000 0.08630121 -0.005992955
## rec_att  0.2236441 0.23981390  0.086301208 1.00000000  0.986196632
## rec_yds  0.1276420 0.14715130 -0.005992955 0.98619663  1.000000000
## rec_tds  0.0701388 0.09057443 -0.051751157 0.94460087  0.972330023
## fumbles  0.3334827 0.32202147  0.201813874 0.31786050  0.257946846
## fpts     0.1484916 0.16906699  0.019656748 0.98222187  0.996966455
##              rec_tds   fumbles       fpts
## rush_att  0.07013880 0.3334827 0.14849159
## rush_yds  0.09057443 0.3220215 0.16906699
## rush_tds -0.05175116 0.2018139 0.01965675
## rec_att   0.94460087 0.3178605 0.98222187
## rec_yds   0.97233002 0.2579468 0.99696646
## rec_tds   1.00000000 0.1463710 0.98474257
## fumbles   0.14637101 1.0000000 0.22294317
## fpts      0.98474257 0.2229432 1.00000000
```

```
##           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
```

```
matrix.wr <- 0
sims <- 10000

for (i in seq(sims)) {
  wr.sim1 <- mvrnorm(30, mu = means.wr, Sigma = cov.wr, empirical = FALSE)
  matrix.wr <- matrix.wr + cor(wr.sim1)
}

matrix.mean <- matrix.wr/sims
matrix.mean
```

```
##           rush_att  rush_yds   rush_tds    rec_att    rec_yds    rec_tds
## rush_att 1.0000000 0.9902050 0.88256004 0.19054782 0.13914595 0.13063719
## rush_yds 0.9902050 1.0000000 0.90979727 0.18112159 0.13223593 0.12305166
## rush_tds 0.8825600 0.9097973 1.00000000 0.06502123 0.02786383 0.02880958
## rec_att  0.1905478 0.1811216 0.06502123 1.00000000 0.98969002 0.96647989
## rec_yds  0.1391460 0.1322359 0.02786383 0.98969002 1.00000000 0.98144194
```

```
## rec_tds  0.1306372 0.1230517 0.02880958 0.96647989 0.98144194 1.00000000
## fumbles  0.1793620 0.1834118 0.10630959 0.42873601 0.39690304 0.35283248
## fpts     0.1707709 0.1641268 0.06197964 0.98712580 0.99750890 0.99024243
##            fumbles       fpts
## rush_att 0.1793620 0.17077091
## rush_yds 0.1834118 0.16412677
## rush_tds 0.1063096 0.06197964
## rec_att  0.4287360 0.98712580
## rec_yds  0.3969030 0.99750890
## rec_tds  0.3528325 0.99024243
## fumbles  1.0000000 0.37643041
## fpts     0.3764304 1.00000000
```

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

```
wr.sim2 <- mvrnorm(30, mu = means.wr, Sigma = cov.wr, empirical = TRUE)
cor.sim2 <- cor(wr.sim2)
cor.sim2; cor.wr
```

```
##           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
```

```
##           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
```

QUESTION 4

10 points

Use LaTeXto create the following expressions.

1. Equation 1 (4 points)

$$P(B) = \sum_j P(B|A_j)P(A_j), \Rightarrow P(A_i|B) = \frac{P(B|A_i)P(A_i)}{\sum_j P(B|A_j)P(A_j)} \tag{1}$$

2. Equation 2 (3 points)

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

3. Equation 3 (3 points)

$$\mathbf{J} = \frac{d\mathbf{f}}{d\mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial x_1} & \cdots & \frac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \cdots & \frac{\partial f}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial fm}{\partial x_1} & \cdots & \frac{\partial fm}{\partial x_n} \end{bmatrix} \tag{3}$$