Bios 6301: Assignment 3

Nick Strayer

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)

✓

1. 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. 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.

Code to set up the test:

```
powerTest <- function(num_patients = 100){
  #Set the treatment group options
  treatmentGroups <- c(0,1)

  #Set the treatments
  treatments <- sample(treatmentGroups, num_patients, replace = T)

  #Draw outcome values
  outcome_pre    <- rnorm(num_patients, mean = 60, sd = 20)

  #Adjust based on treatment.
  outcome_treat  <- ifelse(treatments==1, outcome_pre + 5, outcome_pre)

  #grab pvalue from the lm.
  p = summary(lm(outcome_treat ~ treatments))$coefficients[2,4]
  return(p)
```

```
}

runTests <- function(num_tests = 1000, num_patients = 100, alpha = 0.05){
  #Run the tests a bunch of times and aggregate the results
  tests <- replicate(num_tests, powerTest(num_patients))

  #Return the proportion of values below or equal to our alpha.
  return(sum(tests <= alpha)/num_tests)
}
```

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

```
runTests(num_tests = 1000, num_patients = 100)
```

```
## [1] 0.238
```

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

```
runTests(num_tests = 1000, num_patients = 1000)
```

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

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.

```
d <- read.csv("/Users/Nick/Dropbox/vandy/computing/homeworks/Bios6301_homeworks/data/proj_rb15.csv")[,c
```

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

```
cor(d)
```

```
##           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)

```
#install.packages("MASS")
library(MASS)
```

```
## Warning: package 'MASS' was built under R version 3.1.3
```

```
#define a function to mess with the correlation matrix a "bit".
perturb = function(val) return(val + rnorm(1, 0, 0.05))

cor_aggregate <- matrix(0, 8,8) #Initialize a matrix to hold the sums of correlation matrix results.
n <- 10000
for(i in 1:n){
  new_cor <- apply(cor(d),c(1, 2), FUN = perturb) #make a similar correlation matrix
  fake_approx <- mvrnorm(30, colMeans(d), new_cor, 8,8) #Generate data with that correlation matrix
  cor_aggregate = cor_aggregate + cor(fake_approx) #add to aggregate matrix
}

cor_aggregate/n #Find average
```

```
##           rush_att  rush_yds  rush_tds   rec_att   rec_yds   rec_tds
## rush_att 1.0000000 0.9498002 0.9360941 0.7465103 0.7193619 0.5886865
## rush_yds 0.9498002 1.0000000 0.9389720 0.7417179 0.7146230 0.5921434
## rush_tds 0.9360941 0.9389720 1.0000000 0.7111754 0.6841330 0.5810997
## rec_att  0.7465103 0.7417179 0.7111754 1.0000000 0.9471199 0.8241183
## rec_yds  0.7193619 0.7146230 0.6841330 0.9471199 1.0000000 0.8369931
## rec_tds  0.5886865 0.5921434 0.5810997 0.8241183 0.8369931 1.0000000
## fumbles  0.8419198 0.8409048 0.8377025 0.7343766 0.7121826 0.6028009
## fpts     0.9372646 0.9373631 0.9272245 0.8244610 0.8039450 0.6947859
##            fumbles      fpts
## rush_att 0.8419198 0.9372646
## rush_yds 0.8409048 0.9373631
## rush_tds 0.8377025 0.9272245
## rec_att  0.7343766 0.8244610
## rec_yds  0.7121826 0.8039450
## rec_tds  0.6028009 0.6947859
## fumbles  1.0000000 0.8449661
## fpts     0.8449661 1.0000000
```

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

```
(fake_exact <- mvrnorm(30, colMeans(d), cor(d), 8,8))
```

```
##       rush_att rush_yds   rush_tds  rec_att  rec_yds     rec_tds
##  [1,] 63.18187 271.0748  1.4150900 15.09753 115.9885  1.80273593
##  [2,] 63.86696 271.7404  1.9787485 15.01685 115.7875  1.29848372
##  [3,] 64.32737 272.2559  2.6707974 14.94146 115.5805  0.06605702
##  [4,] 62.25823 270.2880  0.7321786 12.48870 113.0936 -0.53145530
##  [5,] 63.27115 271.2322  1.7375308 13.85131 114.6062 -0.49379718
```

```
##  [6,] 64.73956 272.6284  2.8791632 15.75869 116.3573  1.12407155
##  [7,] 62.78538 270.6226  0.7008350 14.71447 115.4668  0.70909738
##  [8,] 64.26572 272.2016  2.4894702 14.71940 115.3987  0.76498855
##  [9,] 62.87755 270.7890  1.6023634 13.99160 114.7119 -0.19658860
## [10,] 63.92711 271.9647  2.5188664 16.08245 116.8430  2.03971949
## [11,] 63.65067 271.6895  2.1949288 13.78301 114.6180  0.85751089
## [12,] 65.12945 273.1360  3.6440632 15.69326 116.2250  1.56212960
## [13,] 62.78659 270.7496  1.3230679 14.81091 115.4729  2.01853545
## [14,] 64.66511 272.6627  3.1915342 15.34056 116.0681  1.25924371
## [15,] 64.20547 272.1101  2.4000901 14.53169 115.1135  0.17053645
## [16,] 64.78680 272.6948  3.3544422 15.07607 115.8347  0.93384134
## [17,] 62.84277 270.6426  0.8644373 13.61505 114.3898 -0.33889075
## [18,] 63.23288 271.1340  1.4948586 13.25224 113.8540 -1.13805348
## [19,] 63.62029 271.5463  1.7134077 14.89462 115.6719  1.18001837
## [20,] 65.53225 273.5082  3.9902951 15.97115 116.6776  2.22009018
## [21,] 63.59891 271.3892  1.9208687 13.60596 114.4165 -0.05062720
## [22,] 64.16960 272.0009  2.3524761 15.02979 115.5115  1.06853037
## [23,] 62.72015 270.6221  1.1743291 14.81815 115.6124  0.35014619
## [24,] 62.44286 270.3371  0.7688826 13.37036 114.1486 -1.13027986
## [25,] 62.85100 270.8258  0.8176035 14.85164 115.7145  1.26243173
## [26,] 63.65062 271.5439  1.9785527 15.46271 116.0551  0.63874241
## [27,] 61.52259 269.4228  0.3841083 13.12963 113.9579  0.07378159
## [28,] 61.40641 269.4888 -0.1359632 12.11958 112.6809 -1.66304752
## [29,] 63.06915 271.0278  1.8752860 13.90746 114.6946  0.83689576
## [30,] 62.56939 270.5163  0.9369186 14.08911 114.8948 -0.49484781
##           fumbles     fpts
##  [1,]  0.10731366 51.23912
##  [2,]  0.55162462 51.68036
##  [3,]  1.37762497 52.00187
##  [4,]  0.25469457 49.86685
##  [5,] -0.15753441 50.98697
##  [6,]  2.50374898 52.45345
##  [7,]  0.06976427 50.61434
##  [8,]  1.42624592 51.92998
##  [9,]  0.57250332 50.71059
## [10,]  1.32344261 52.17244
## [11,]  0.26393169 51.44864
## [12,]  2.25260466 52.95484
## [13,]  0.18824514 50.92643
## [14,]  1.51927120 52.54448
## [15,]  2.08387211 51.73545
## [16,]  1.66803139 52.52333
## [17,] -0.26156736 50.40418
## [18,] -0.07181786 50.68002
## [19,]  1.19609790 51.44812
## [20,]  3.26117586 53.35907
## [21,]  1.29899178 51.07159
## [22,]  0.48796851 51.85898
## [23,]  0.31462126 50.73636
## [24,] -0.49958413 50.10591
## [25,]  0.94460859 50.82522
## [26,]  1.70820142 51.54318
## [27,]  0.09388851 49.50071
## [28,] -1.39762587 49.09103
```

```
## [29,]  0.52720509 50.98595
## [30,]  0.13091317 50.43129
```

Question 4

10 points

Use LaTeXto create the following expressions.

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

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

1.

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

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

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

2.

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

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

3.

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