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] 0.221
```

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

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

```
## [1] 0.981
```

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)

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

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

```r
#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 0.9496292 0.9359707 0.7459287 0.7191474 0.5891473
## rush_yds 0.9496292 1.0000000 0.9380959 0.7419185 0.7147011 0.5926756
## rush_tds 0.9359707 0.9380959 1.0000000 0.7098705 0.6833800 0.5808034
## rec_att  0.7459287 0.7419185 0.7098705 1.0000000 0.9468772 0.8241316
## rec_yds  0.7191474 0.7147011 0.6833800 0.9468772 1.0000000 0.8345562
## rec_tds  0.5891473 0.5926756 0.5808034 0.8241316 0.8345562 1.0000000
## fumbles  0.8420114 0.8411734 0.8374050 0.7340176 0.7112681 0.6037100
## fpts     0.9366076 0.9375703 0.9271668 0.8239334 0.8029470 0.6941779
##            fumbles      fpts
## rush_att 0.8420114 0.9366076
## rush_yds 0.8411734 0.9375703
## rush_tds 0.8374050 0.9271668
## rec_att  0.7340176 0.8239334
## rec_yds  0.7112681 0.8029470
## rec_tds  0.6037100 0.6941779
## fumbles  1.0000000 0.8452799
## fpts     0.8452799 1.0000000
```

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

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

```
##       rush_att rush_yds   rush_tds  rec_att  rec_yds     rec_tds
##  [1,] 64.17977 272.1317  2.4916960 14.64761 115.3910  0.93907962
##  [2,] 64.19481 272.0422  2.4905401 13.95754 114.7411 -1.18181206
##  [3,] 61.67695 269.5113 -0.1164215 13.13860 113.9438 -0.48072821
##  [4,] 62.45277 270.4907  1.0547751 13.36241 114.0781  0.06269497
##  [5,] 64.39493 272.3913  2.5204729 14.97935 115.6170  1.51931278
##  [6,] 62.33558 270.3034  1.0820162 13.80706 114.5993  0.22149474
##  [7,] 63.52590 271.4769  2.1229924 13.90111 114.4778 -0.80235041
##  [8,] 62.86720 270.8682  1.4495440 13.60525 114.3650  0.51480621
##  [9,] 62.59639 270.4516  0.6735651 14.10816 114.9072  0.32679435
## [10,] 63.66288 271.4893  1.6873580 15.03254 115.7078  0.90362549
## [11,] 63.98040 271.9412  2.1454489 15.66880 116.2275  1.21115095
## [12,] 63.46965 271.4334  1.6935831 15.30093 116.1063  1.65346890
## [13,] 63.56112 271.5668  2.4439144 14.32602 115.0611  1.11976504
## [14,] 62.50088 270.4840  1.2608615 14.09578 114.6428  0.45737864
## [15,] 63.50012 271.3771  1.9072988 15.35771 116.3601  1.93426760
## [16,] 63.80753 271.8093  1.8903399 13.78194 114.4357  0.36488600
## [17,] 64.18155 271.9508  2.3687503 14.23397 114.8397  0.49990970
## [18,] 61.75584 269.5532 -0.0334214 13.35663 114.1578 -0.80042030
## [19,] 65.14211 273.0939  3.8567849 15.07138 115.7074  0.99679233
## [20,] 65.44841 273.3182  3.6478232 17.23846 117.7524  2.33817870
## [21,] 65.06416 272.9767  3.4516699 16.71842 117.5755  2.99237136
## [22,] 63.86211 271.7740  2.1782632 13.22938 113.8061 -0.47507013
## [23,] 63.61276 271.4915  2.0407629 14.62611 115.3415  0.40981692
## [24,] 62.39919 270.4067  1.0885891 13.44011 114.2180 -0.39623692
## [25,] 64.63884 272.6194  3.1052449 15.67396 116.4728  1.92145974
## [26,] 62.01652 270.0108  0.1953633 13.63830 114.3346 -0.33187503
## [27,] 63.90900 271.8360  2.2546928 14.80286 115.5037  0.12937024
## [28,] 63.80452 271.7687  1.8801369 14.88320 115.6093  0.03521987
## [29,] 62.45907 270.3531  0.8215959 13.78891 114.5474  0.69994638
## [30,] 62.95288 270.9247  1.3149901 14.24286 114.9182 -0.58329747
##           fumbles     fpts
##  [1,]  0.96442847 51.92313
##  [2,]  1.98598371 51.56364
##  [3,] -0.95113740 49.41602
##  [4,]  0.04572841 50.31587
##  [5,]  1.34782002 52.14139
##  [6,]  0.92364037 50.29803
##  [7,]  0.87522161 51.14767
##  [8,]  0.60327676 50.69483
##  [9,]  0.18146587 50.37542
## [10,]  0.37671614 51.43545
## [11,]  2.13155078 51.86298
## [12,]  0.97178241 51.49876
## [13,]  1.32623988 51.51185
## [14,]  0.22245122 50.49153
## [15,]  0.98610522 51.58831
## [16,]  1.46975631 51.33461
## [17,]  1.01759145 51.63741
## [18,] -0.43182822 49.47107
## [19,]  1.90333861 52.85215
## [20,]  2.92345409 53.39567
## [21,]  2.74286722 53.16264
## [22,]  0.67013591 51.23107
## [23,]  1.25691560 51.37819
## [24,] -1.27020931 50.31592
## [25,]  1.43137572 52.61171
## [26,] -0.32432868 49.85869
## [27,]  0.58372437 51.67407
## [28,]  0.28431866 51.56049
## [29,] -0.66538734 50.32899
## [30,]  0.15546372 50.75323
```

Question 4

10 points

Use LaTeXto create the following expressions.

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

1.

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

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

2.

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

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

3.

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