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:

```
set.seed(8)
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] 0.258
```

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

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

```
## [1] 0.975
```

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/homework/data/proj_rb15.csv")[,c(-1,-2)]
```

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 0.9494871 0.9353233 0.7468867 0.7186479 0.5890362
## rush_yds 0.9494871 1.0000000 0.9383563 0.7432888 0.7149627 0.5932075
## rush_tds 0.9353233 0.9383563 1.0000000 0.7105165 0.6835282 0.5805593
## rec_att  0.7468867 0.7432888 0.7105165 1.0000000 0.9468610 0.8248086
## rec_yds  0.7186479 0.7149627 0.6835282 0.9468610 1.0000000 0.8366219
## rec_tds  0.5890362 0.5932075 0.5805593 0.8248086 0.8366219 1.0000000
## fumbles  0.8422518 0.8403695 0.8374313 0.7340727 0.7112185 0.6038325
## fpts     0.9373294 0.9380371 0.9270160 0.8244209 0.8039230 0.6956220
##            fumbles      fpts
## rush_att 0.8422518 0.9373294
## rush_yds 0.8403695 0.9380371
## rush_tds 0.8374313 0.9270160
## rec_att  0.7340727 0.8244209
## rec_yds  0.7112185 0.8039230
## rec_tds  0.6038325 0.6956220
## fumbles  1.0000000 0.8445371
## fpts     0.8445371 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,] 64.28080 272.2125  2.2580988 14.92546 115.8485  0.93579466
##  [2,] 61.36621 269.2532 -0.2254936 12.23128 112.9395 -1.00473209
##  [3,] 64.42668 272.4676  3.0683437 14.31539 114.8593 -0.40967656
##  [4,] 62.65015 270.4971  1.1238276 13.66067 114.3924  0.24227138
##  [5,] 63.73929 271.7837  2.3488850 15.12542 115.8093  1.18131622
```

```
##  [6,] 62.37773 270.3491  0.8189804 12.63579 113.3117 -1.39524896
##  [7,] 62.97378 270.9076  1.1159613 14.23190 114.9387 -0.52867281
##  [8,] 62.57616 270.5359  0.9473763 13.30755 114.0013 -1.01580624
##  [9,] 63.23750 271.1391  1.5012577 14.31280 115.0486  0.09358632
## [10,] 63.56985 271.6141  2.3068725 14.94520 115.7094  1.72600947
## [11,] 64.93871 272.8864  3.2704177 15.94705 116.7153  2.14178120
## [12,] 65.04262 272.9175  2.9656441 15.76957 116.1900  1.14017517
## [13,] 63.88244 271.7641  2.2482904 15.32047 116.0437  1.19647722
## [14,] 61.83185 269.6800  0.3354090 14.07196 114.9302  0.65142800
## [15,] 64.31564 272.2241  2.6801615 15.69085 116.3659  1.12094552
## [16,] 63.85280 271.8035  2.2655493 13.28619 113.9784  0.31807037
## [17,] 64.28225 272.1537  2.2656787 14.75147 115.3714  0.66428421
## [18,] 63.31727 271.2146  1.2500282 14.54756 115.2570  0.67905228
## [19,] 63.77532 271.7838  2.3389132 14.39050 115.1397  1.02289644
## [20,] 63.96281 271.7753  1.9473895 14.80257 115.5468  0.87537783
## [21,] 64.63996 272.5400  3.1319971 14.91953 115.5139  1.31566764
## [22,] 62.36000 270.3421  0.6865766 14.21477 115.0452  0.30354205
## [23,] 64.73279 272.5108  3.0682244 16.10029 117.0072  1.03282899
## [24,] 62.48515 270.5232  0.7055583 14.77296 115.4957  1.36480481
## [25,] 62.87408 270.7794  1.5143151 12.82090 113.6651 -0.57803587
## [26,] 62.43134 270.3617  0.9440291 14.33129 115.1020  0.78983950
## [27,] 65.00003 273.0187  3.8321002 15.54729 116.2039  1.74179530
## [28,] 63.42774 271.2860  1.8591044 14.68573 115.2296 -0.15993558
## [29,] 62.18730 270.1477  0.6208093 13.02145 113.6932 -1.51033316
## [30,] 63.41558 271.3735  1.7749248 15.33154 116.0929  2.26449665
##           fumbles     fpts
##  [1,]  1.98794620 51.95857
##  [2,] -0.33097861 48.98503
##  [3,]  1.40987767 52.05027
##  [4,]  0.61559128 50.39193
##  [5,]  1.10901998 51.77139
##  [6,]  0.34708597 49.91853
##  [7,] -0.26395167 50.71144
##  [8,] -0.50078543 50.25273
##  [9,]  1.10268107 50.96489
## [10,]  1.44656572 51.66968
## [11,]  2.87358900 52.83321
## [12,]  2.10610019 52.62883
## [13,]  1.22019573 51.77993
## [14,] -0.43664329 49.88387
## [15,]  2.26986125 52.18640
## [16,]  0.04548931 51.37804
## [17,]  0.67318244 51.86455
## [18,]  0.32706001 51.05630
## [19,]  0.42018328 51.64337
## [20,]  1.44120569 51.57968
## [21,]  1.80451267 52.33239
## [22,] -0.55390307 50.36532
## [23,]  1.43749897 52.60274
## [24,]  0.06156344 50.61283
## [25,] -0.44872529 50.48662
## [26,] -0.11673793 50.47043
## [27,]  2.73592964 52.91906
## [28,]  0.68278081 51.16840
```

```
## [29,] -0.17905989 49.82674
## [30,]  0.45132635 51.53760
```

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.

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