Homework 3

Molly Olson

October 3, 2015

Question 2

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 a 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 grouop, and extract the p-value (hint: see assignment 1). 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.

```r
#nset seed
set.seed(1)
```

```r
#the function takes the number of patients. It is one procedure through and will be used to repeat the p
treatment <- function(sampleSize){

  #treatment groups
  treatmentGroups <- c(0,1)
  # assign treatments for n patients, we can use sample because
  #there is equal probability of being in either treatment group
  treatmentPerPatient <- sample(treatmentGroups, sampleSize, replace=TRUE)
  #treatments

  #outcome should be normal with mu=60 and Sigma=20
  outcome <- rnorm(sampleSize, mean = 60, sd = 20)

  #if the patient is in the treatment group, i=1, then add 5 to it, otherwise do nothing
  treatmentOutcome <- ifelse(treatmentPerPatient==1, outcome + 5, outcome)
  #treatmentOutcome

  #craeate a linear model for the outcome by treatment group
  linMod <- lm(treatmentOutcome ~ treatmentPerPatient)
  #summary(linMod)
  #return the pvalue
  p <- coef(summary(linMod))[2,4]
  return(p)
}
```

```r
getPower <- function(simulations, sampleSize, alpha){
  #repeat procedure (simulations) number of times
  #allPvalues will have all of the pvalues for each simulation
  allPvalues <- replicate(simulations, treatment(sampleSize))
  #power is the percentage of times the pvalue is less than alpha
  power <- sum(allPvalues <= alpha)/simulations
  return(power)
}
```

1. Find the power when the sample size is 100 patients

```r
getPower(1000, 100, 0.05)
```

```
## [1] 0.281
```

2. Find the power when the sample size is 1000 patients

```r
getPower(1000,1000,0.05)
```

```
## [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.

```r
setwd("~/Documents/Vanderbilt_2015_Fall/Statistical_Computing/football-values-master/2015")
library(MASS)

rb_data <- read.csv("proj_rb15.csv",header=TRUE)

newrb <- rb_data[c(-1,-2)]
```

1. Show the correlation matrix of this dataset.

```r
rho <- cor(newrb)
```

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.

```r
rho.rb <- cor(newrb) ##Correlation matrix
vcov.rb <- var(newrb) ##Variance matrix
means.rb <- colMeans(newrb) ##returns the mean for columns
```

```r
##simultate using mvnorm, 30 values, mu and sigma the same as rb data
rb.sim <- mvrnorm(30, mu = means.rb, Sigma = vcov.rb)
##convert to data frame
rb.sim <- as.data.frame(rb.sim)
rb.sim
```

```
##      rush_att   rush_yds    rush_tds    rec_att    rec_yds     rec_tds
## 1   81.603049  368.20818  2.66539424  32.647163  270.23197  1.42706012
## 2  204.971736  874.49662  5.96802566  41.104803  322.63345  2.10178580
## 3   55.814312  237.61319  1.22411164  37.059668  282.57144  0.81138594
## 4  156.923723  701.54404  4.25525394  37.444179  310.72406  1.24264068
## 5   36.618366  135.05783  1.08421204  10.736366   97.41280  0.71921027
## 6   26.838136  124.76106  0.96911520 -17.714416 -174.42288 -1.01937631
## 7   29.386349   74.50493  0.59860193  21.107729  169.85787  0.93222480
## 8   69.155922  275.53265  1.51347198  11.381829  106.35415  0.39447559
## 9  157.458662  678.43864  5.06371696  38.934384  313.73355  1.27714827
## 10 -67.153927 -302.97420 -2.26820519  -5.062380  -31.89264  0.02039508
## 11  -9.685195  -51.53835  0.01080570  -2.309514  -19.20674  0.13736237
## 12  54.242222  235.22428  1.32124232   7.700035   73.87487  0.27246688
## 13 133.316430  543.53860  2.92430358  23.748318  194.30598  0.96597997
## 14  45.310490  195.79482  1.50839540   7.996124   62.29252 -0.31527714
## 15 -46.254292 -179.53808 -2.27905817  -2.267460  -22.62423 -0.25843567
## 16  80.631714  318.69144  1.21519874  39.430006  310.39324  0.96120800
## 17 -78.789728 -361.70752 -2.22086077  -7.113977  -59.70555  0.11883696
## 18 132.076053  578.83766  3.83092068  18.983166  162.92013  0.82634254
## 19   7.874279   21.22121  0.50333821   7.718814   42.36561 -0.16990814
## 20  85.114436  314.94189  2.81024580   8.371183   61.31674  0.28101007
## 21 112.052798  487.37362  3.77392811  18.101022  145.01281  0.81954927
## 22 135.930618  616.95055  4.18890506  27.202361  206.27046  0.20386051
## 23  49.576277  226.16747  0.04011639  16.572201  112.81670  0.09415518
## 24 154.196972  645.83187  3.04988065  32.588940  255.64857  1.29190755
## 25  76.692247  309.63108  2.04744083  18.652667  141.42449  0.45210674
## 26  59.775270  288.15646  1.84833271  19.559118  146.59329  0.50264473
## 27 196.897781  869.43481  5.31502567  37.118997  278.99445  1.11128397
## 28 126.857498  554.27988  2.68916155  36.858273  304.15759  1.71436520
## 29  95.440260  440.70423  2.20788363  28.913109  236.38547  0.58732804
## 30   8.220845   40.38183  0.27298665  10.982371   55.02183 -0.01452906
##       fumbles       fpts
## 1   1.0040622  86.267371
## 2   2.2004696 163.514013
## 3   1.4656123  61.283711
## 4   1.2869233 131.857394
## 5   0.7051101  32.885669
## 6  -0.3109560  -4.597646
## 7   0.5078943  32.623525
## 8   0.6780489  48.259274
## 9   1.5096920 134.371997
## 10 -0.1424627 -46.693626
## 11  0.7231372  -7.480020
## 12  1.3711491  37.332899
## 13  1.6327265  93.811057
## 14  0.2495200  32.402069
## 15 -0.1938751 -35.124862
## 16  0.5078816  74.990534
## 17 -0.1611034 -54.990897
## 18  0.5360741 101.327195
## 19 -0.8864040  10.349110
## 20  0.4301827  55.163541
## 21  1.7693964  86.950303
## 22  1.7167957 105.286981
## 23  0.9172106  33.349347
## 24  0.7112733 114.731165
## 25  0.5414861  59.509590
## 26  0.9664182  55.701433
## 27  0.6300027 152.065361
## 28  1.5472803 108.825051
## 29  1.1722519  82.257080
## 30  0.6531031   9.519046
```

```
rho.sim <- cor(rb.sim)
rho.sim ## this is the similar correlation matrix
```

```
##           rush_att  rush_yds  rush_tds   rec_att   rec_yds   rec_tds
## rush_att 1.0000000 0.9975454 0.9659727 0.8027501 0.7973140 0.7149788
## rush_yds 0.9975454 1.0000000 0.9648480 0.8009759 0.7944074 0.6999228
## rush_tds 0.9659727 0.9648480 1.0000000 0.7273136 0.7219083 0.6579178
## rec_att  0.8027501 0.8009759 0.7273136 1.0000000 0.9954492 0.8566808
## rec_yds  0.7973140 0.7944074 0.7219083 0.9954492 1.0000000 0.8798934
## rec_tds  0.7149788 0.6999228 0.6579178 0.8566808 0.8798934 1.0000000
## fumbles  0.6863435 0.6912500 0.6691106 0.6810339 0.6934187 0.6776780
## fpts     0.9862491 0.9855233 0.9546242 0.8795748 0.8770637 0.7908361
##            fumbles      fpts
## rush_att 0.6863435 0.9862491
## rush_yds 0.6912500 0.9855233
## rush_tds 0.6691106 0.9546242
## rec_att  0.6810339 0.8795748
## rec_yds  0.6934187 0.8770637
## rec_tds  0.6776780 0.7908361
## fumbles  1.0000000 0.7147066
## fpts     0.7147066 1.0000000
```

```
rho.rb ## this is the original correlation matrix
```

```
##           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
```

```
average <-0
n <- 10000
for(i in seq(n)){
    rb.sim <- mvrnorm(30, mu = means.rb, Sigma=vcov.rb) #create several datasets using mvrnorm
    average <- average + cor(rb.sim)/n #just keep adding the average to build up the average
    }
average ## this is the mean correlation matrix with similar correlation structure
```

```
##           rush_att  rush_yds  rush_tds   rec_att   rec_yds   rec_tds
## rush_att 1.0000000 0.9974706 0.9714270 0.7650199 0.7356050 0.5914271
## rush_yds 0.9974706 1.0000000 0.9767655 0.7601052 0.7298445 0.5966978
## rush_tds 0.9714270 0.9767655 1.0000000 0.7216207 0.6935825 0.5855420
## rec_att  0.7650199 0.7601052 0.7216207 1.0000000 0.9942286 0.8340451
## rec_yds  0.7356050 0.7298445 0.6935825 0.9942286 1.0000000 0.8476956
## rec_tds  0.5914271 0.5966978 0.5855420 0.8340451 0.8476956 1.0000000
## fumbles  0.8547969 0.8542501 0.8485418 0.7405059 0.7168070 0.5996222
## fpts     0.9818244 0.9837858 0.9679489 0.8523715 0.8303573 0.7081600
##            fumbles      fpts
## rush_att 0.8547969 0.9818244
## rush_yds 0.8542501 0.9837858
## rush_tds 0.8485418 0.9679489
## rec_att  0.7405059 0.8523715
## rec_yds  0.7168070 0.8303573
## rec_tds  0.5996222 0.7081600
## fumbles  1.0000000 0.8594595
## fpts     0.8594595 1.0000000
```

```
rho.rb ##this is the original data correlation matrix
```

```
##           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

1.
$$\begin{aligned} P(B) &= \sum_j P(B|A_j)P(A_j), \\ \Rightarrow P(A_i|B) &= \frac{P(B|A_i)P(A_i)}{\sum_j (B|A_j)P(A_j)} \end{aligned} \tag{1}$$

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

3.
$$\mathbf{J} = \frac{d\mathbf{f}}{d\mathbf{x}} = \begin{bmatrix} \frac{\partial \mathbf{f}}{\partial x_1} & \cdots & \frac{\partial \mathbf{f}}{\partial x_n} \end{bmatrix} \begin{bmatrix} \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \tag{3}$$