Bios 6301: Assignment 3

Lingjun Fu

07 October, 2015

Due Thursday, 08 October, 1:00 PM

50 points total.

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

This assigment includes turning in the first two assignments. All three should include knitr files (named `homework1.rmd`, `homework2.rmd`, `homework3.rmd`) along with valid PDF output files. Inside each file, clearly indicate which parts of your responses go with which problems (you may use the original homework document as a template). Add your name as `author` to the file's metadata section. Raw R code/output or word processor files are not acceptable.

Failure to properly name files or include author name may result in 5 points taken off.

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)

2. Commit each assignment individually. This means your repository should have at least three commits. (5 points)

Please see my GitHub repsitory for details.

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.

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

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

```
##step 1
alpha <- 0.05
sample <- 100
times <- 1000
count <- 0
for(i in 1:times){
    set.seed(i)
    group <- c(rbinom(sample, 1, 0.5))
    outcome <- c(rnorm(sample, 60, 20))
    x <- data.frame(group, outcome)
    for(j in 1:sample){
        if(x[j,1] == 1){
            x[j,2] = x[j,2] + 5
        }
    }
    mod <- lm(outcome ~ group, dat=x)
    mod.c <- coef(summary(mod))
    p <- 2*pt(mod.c[2,3], df = 98, lower.tail = FALSE)
    if(p <= alpha){
        count = count + 1
    }
}
power100 <- count/times
print(paste0("the power when the sample size is 100 patients: ", power100))
```

```
## [1] "the power when the sample size is 100 patients: 0.237"
```

```
#############

##step 2
alpha <- 0.05
sample <- 1000
times <- 1000
count <- 0
for(i in 1:times){
    set.seed(i)
    group <- c(rbinom(sample, 1, 0.5))
    outcome <- c(rnorm(sample, 60, 20))
    x <- data.frame(group, outcome)
    for(j in 1:sample){
        if(x[j,1] == 1){
            x[j,2] = x[j,2] + 5
        }
    }
    mod <- lm(outcome ~ group, dat=x)
    mod.c <- coef(summary(mod))
    p <- 2*pt(mod.c[2,3], df = 998, lower.tail = FALSE)
    if(p <= alpha){
        count = count + 1
    }
}
power1000 <- count/times
print(paste0("the power when the sample size is 1000 patients: ", power1000))
```

```
## [1] "the power when the sample size is 1000 patients: 0.977"
```

The above is the R code. We can see that power increases significantly when we increase sample size from 100 to 1000.

Question 3

15 points

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.

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

```
library(RCurl)
```

```
## Loading required package: bitops
```

```
my_url<-"https://raw.githubusercontent.com/couthcommander/football-values/master/2015/proj_rb15.csv"
download.file(my_url, destfile="football.csv",method="curl")
football=read.csv("football.csv")
football <- football[,-(1:2)]
##step 1
corr <- cor(football)
corr
```

```
##           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
```

```r
##step 2
library(MASS)
times <- 10000
corr2 <- matrix(data = 0, nrow = 8, ncol = 8, byrow = FALSE)
for(i in 1:times){
    mean <- rep(0, 8)
    for(i in 1:8){
        mean[i] = mean(football[,i])
    }
    r2 <- mvrnorm(n = 30, mean, corr, tol = 1e-6, empirical = FALSE, EISPACK = FALSE)
    corr2 = corr2 + cor(r2)
}
mean_corr <- corr2/times
mean_corr # can use corr-mean_corr to compare the difference
```

```
##             rush_att  rush_yds  rush_tds   rec_att   rec_yds   rec_tds
## rush_att 1.0000000 0.9974523 0.9711989 0.7650333 0.7355767 0.5903583
## rush_yds 0.9974523 1.0000000 0.9765317 0.7603085 0.7299774 0.5957052
## rush_tds 0.9711989 0.9765317 1.0000000 0.7214188 0.6933456 0.5841919
## rec_att  0.7650333 0.7603085 0.7214188 1.0000000 0.9942207 0.8341093
## rec_yds  0.7355767 0.7299774 0.6933456 0.9942207 1.0000000 0.8477129
## rec_tds  0.5903583 0.5957052 0.5841919 0.8341093 0.8477129 1.0000000
## fumbles  0.8546867 0.8540099 0.8477369 0.7404500 0.7169456 0.5982858
## fpts     0.9817308 0.9837508 0.9677611 0.8526135 0.8305464 0.7074210
##              fumbles      fpts
## rush_att 0.8546867 0.9817308
## rush_yds 0.8540099 0.9837508
## rush_tds 0.8477369 0.9677611
## rec_att  0.7404500 0.8526135
## rec_yds  0.7169456 0.8305464
## rec_tds  0.5982858 0.7074210
## fumbles  1.0000000 0.8591420
## fpts     0.8591420 1.0000000
```

```r
##step 3
r3 <- mvrnorm(n = 30, mean, corr, tol = 1e-6, empirical = TRUE, EISPACK = FALSE)
r3 # cor(r3)-corr is very close to zero matrix
```

```
##       rush_att rush_yds    rush_tds  rec_att  rec_yds      rec_tds
##  [1,] 63.37275 271.2530  1.43006204 13.45018 114.1158 -0.423270712
##  [2,] 64.65352 272.6673  3.12471322 15.77105 116.5157  1.958941595
##  [3,] 63.72226 271.5667  2.05794685 14.90989 115.6085  1.378705722
##  [4,] 63.08657 270.9035  1.62921516 14.71883 115.6316 -0.100349503
##  [5,] 63.75997 271.6960  1.91346291 15.43095 116.1089  1.004507305
##  [6,] 65.15040 273.0105  3.03148889 15.22801 116.0012  0.225190155
##  [7,] 64.54117 272.4491  2.77978203 15.15751 115.8323  0.794395670
##  [8,] 64.83597 272.8337  3.57136010 14.96184 115.6067  1.491299058
##  [9,] 62.80002 270.7096  1.04001273 12.88721 113.5877 -0.773794993
## [10,] 61.40559 269.3660  0.01834375 12.56661 113.4175 -0.885428507
## [11,] 63.54436 271.5056  2.06924987 14.52442 115.3695  2.093515885
## [12,] 64.97369 272.8729  3.26754026 15.26233 115.9851  1.429149167
## [13,] 62.95832 270.9925  1.31055308 13.87309 114.5244 -0.193265727
## [14,] 62.98983 270.9836  1.29640902 14.34986 115.2293  0.985797729
## [15,] 64.56238 272.4093  2.72115769 15.78608 116.3358  1.624807008
## [16,] 63.88627 271.7979  2.39114006 14.38317 115.0818 -0.032273053
## [17,] 63.77785 271.7327  2.11664102 14.61119 115.0823 -0.004243528
## [18,] 61.25055 269.2055 -0.18571191 12.85420 113.6282 -1.020334539
## [19,] 62.17441 270.1484  0.72934269 12.97767 113.7662 -0.641181997
## [20,] 63.26048 271.2453  1.53424409 13.86984 114.4464  0.116416879
## [21,] 62.60934 270.4894  0.89159291 13.16936 113.8473 -0.737964482
## [22,] 64.73992 272.7546  3.46328058 15.57277 116.3651  1.779092507
## [23,] 62.68580 270.6995  0.84125702 15.54508 116.3498  1.716788303
## [24,] 63.88440 271.9022  2.14957441 14.82855 115.5020  1.063498681
## [25,] 63.65540 271.4939  1.82025488 16.45441 117.2598  2.355872445
## [26,] 63.26394 271.2526  1.77658782 14.28411 114.7922 -0.657545518
## [27,] 64.19702 272.0452  2.63883148 15.00484 115.6854  0.223727783
## [28,] 62.84446 270.7375  1.21651808 13.59309 114.4076 -0.028600224
## [29,] 62.18078 269.9556  0.16329594 13.85191 114.5962  0.505291297
## [30,] 63.18642 271.1666  2.16108410 14.13735 114.7658  0.951255595
##           fumbles     fpts
##  [1,]  0.98486563 50.79208
##  [2,]  1.79075199 52.64679
##  [3,]  0.98128531 51.54938
##  [4,]  0.25302558 50.98857
##  [5,]  0.57591319 51.67181
##  [6,]  2.31085272 52.59818
##  [7,]  1.32534562 52.24153
##  [8,]  1.68591583 52.65017
##  [9,]  0.30411490 50.27692
## [10,] -1.37831036 49.25269
## [11,]  1.11468161 51.51201
## [12,]  2.05601423 52.66153
## [13,]  0.86374843 50.71191
## [14,]  0.41744829 50.94279
## [15,]  1.32755730 52.36420
## [16,]  0.60090975 51.59139
## [17,]  1.60820537 51.44439
## [18,] -1.28056589 49.13968
## [19,]  0.04045773 49.92627
## [20,]  0.40284457 50.93178
## [21,] -1.11201087 50.20933
## [22,]  1.84544799 52.74074
## [23,]  0.73506877 50.92551
## [24,]  0.15178399 51.75367
## [25,]  1.88367097 51.81241
## [26,]  0.57257585 51.02462
## [27,]  2.32854207 51.86967
## [28,]  0.30558959 50.54624
## [29,] -0.54467629 49.93288
## [30,]  1.58740768 51.12162
```

See the above R code.
In step 1, we show the correlation matrix of the original data set.
In step 2, we generate 10,000 data sets each of which has 30 rows and similar correlation structure. Then we calculate the mean correlation matrix of them. We see that it's close to the original one.
In step 3, we generate one data set which has exactly the same correlation structure as the original one.

Question 4

10 points

1. Formula 1:

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

2. Formula 2:

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

3. Formula 3:

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

Use LaTeXto create the following expressions.

1. Hint: `\Rightarrow` (4 points)

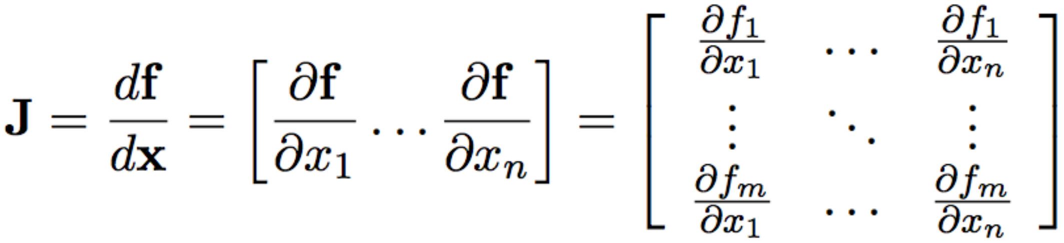
$$P(B) = \sum_j P(B|A_j)P(A_j),$$

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

2. Hint: `\zeta` (3 points)

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

3. Hint: `\partial` (3 points)



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