Bios 6301: Assignment 3

Rui Wang

October 01, 2016

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)

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)

```
set.seed(100)
n <- 100
pvals <- numeric(n)
# iterate 1000 times, saving each p value
for (i in 1:1000) {
  # Bernoulli distribution
  treat <- rbinom(n,1,0.5)
```

```
  # add 5 to the outcome if the treat is 1
  outcome <- rnorm(n, 60, 20) + treat * 5
  x <- data.frame(treat, outcome)
  pvals[i] <- t.test(outcome ~ treat, dat = x, var.equal = TRUE)$p.value

}
power_100 <- mean(pvals <= 0.05 )*100
power_100
```

```
## [1] 23.6
```

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

```
set.seed(1000)
n <- 1000
pvals <- numeric(n)
# iterate 1000 times, saving each p value
for (i in 1:1000) {
  # Bernoulli distribution
  treat <- rbinom(n,1,0.5)
  # add 5 to the outcome if the treat is 1
  outcome <- rnorm(n, 60, 20) + treat * 5
  x <- data.frame(treat, outcome)
  pvals[i] <- t.test(outcome ~ treat, dat = x, var.equal = TRUE)$p.value

}
power_1000 <- mean(pvals <= 0.05 )*100
power_1000
```

```
## [1] 96.8
```

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)

```
setwd("/Users/ruiwang/Dropbox/Biostatistics/6301/homework")
fb <- data.frame(read.csv("proj_rb15.csv"))
# remove the first two columns
fb <- fb[,c(-1,-2)]
# show the correlation matrix of this data set
cor.fb <- cor(fb)
cor.fb
```

```
##           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
```

```
var.fb<-var(fb)
var.fb
```

```
##             rush_att    rush_yds    rush_tds    rec_att     rec_yds
## rush_att  6328.46094  27979.7864 192.0590190  905.61735  7114.11126
## rush_yds 27979.78642 124314.0880 855.7254243 3988.44050 31286.96872
## rush_tds   192.05902    855.7254   6.1647756   26.68256   209.50699
## rec_att    905.61735   3988.4405  26.6825607  218.89892  1777.36140
## rec_yds   7114.11126  31286.9687 209.5069913 1777.36140 14593.66553
## rec_tds     30.99774    138.5786   0.9576186    8.09766    67.17920
## fumbles     67.34980    298.2894   2.0867780   10.87760    86.02767
## fpts      4712.19384  20925.1864 145.0569358  763.34248  6074.88603
##              rec_tds     fumbles        fpts
## rush_att  30.9977423  67.3497975  4712.19384
## rush_yds 138.5786082 298.2893881 20925.18645
## rush_tds   0.9576186   2.0867780   145.05694
## rec_att    8.0976598  10.8775990   763.34248
## rec_yds   67.1791959  86.0276733  6074.88603
## rec_tds    0.4261237   0.3896289    28.07858
## fumbles    0.3896289   0.9715215    51.32110
## fpts      28.0785773  51.3211018  3635.46115
```

```
mean.fb<-colMeans(fb)
mean.fb
```

```
##    rush_att    rush_yds    rush_tds     rec_att     rec_yds     rec_tds
##  63.4651282 271.3948718   1.8323077  14.4671795 115.1815385   0.5400000
##     fumbles        fpts
##   0.7912821  51.2610256
```

1. 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)

```
library(MASS)
loops <- 1e4
keep.1 <- 0
```

```
set.seed(1)
for (i in seq(loops)) {
    fb.sim <- as.data.frame(mvrnorm(n=30, mu = mean.fb, Sigma=var.fb))
    keep.1 <- keep.1 + cor(fb.sim)/loops
}
# a similar correlation
keep.1
```

```
##           rush_att  rush_yds  rush_tds   rec_att   rec_yds   rec_tds
## rush_att 1.0000000 0.9974500 0.9713472 0.7643291 0.7350341 0.5911347
## rush_yds 0.9974500 1.0000000 0.9767243 0.7593050 0.7291431 0.5962737
## rush_tds 0.9713472 0.9767243 1.0000000 0.7205943 0.6926162 0.5849183
## rec_att  0.7643291 0.7593050 0.7205943 1.0000000 0.9942779 0.8347768
## rec_yds  0.7350341 0.7291431 0.6926162 0.9942779 1.0000000 0.8483749
## rec_tds  0.5911347 0.5962737 0.5849183 0.8347768 0.8483749 1.0000000
## fumbles  0.8545518 0.8538644 0.8478191 0.7410831 0.7175978 0.6001970
## fpts     0.9817474 0.9836956 0.9677966 0.8520135 0.8300580 0.7081649
##            fumbles      fpts
## rush_att 0.8545518 0.9817474
## rush_yds 0.8538644 0.9836956
## rush_tds 0.8478191 0.9677966
## rec_att  0.7410831 0.8520135
## rec_yds  0.7175978 0.8300580
## rec_tds  0.6001970 0.7081649
## fumbles  1.0000000 0.8593798
## fpts     0.8593798 1.0000000
```

```
cor(fb)
```

```
##           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
```

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

```r
# set empirical equals TRUE value, we can have the exact correlation structure
fb.sim <- mvrnorm(n=30, mu = mean.fb, Sigma=var.fb, empirical = TRUE)
cor(fb.sim)
```

```
##           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
cor.fb
```

```
##           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

10 points

Use LaTeXto create the following expressions.

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

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

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

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

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

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

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

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

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

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

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