Bios 6301: Assignment 3

Rui Wang

October 01, 2016

Grade: 47/50

JC Grading -3 Hi Rui, just a heads up that this was the 2015 assignment. Fortunately, this assignment was similar to 2016 (though the data was slightly different). This may not always be the case for other assignments.

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)

```r
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"))
fb <- data.frame(read.csv("proj_wr16.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 0.9906030 0.88608205 0.19706851 0.14473723 0.13548999
## rush_yds 0.9906030 1.0000000 0.91252627 0.18745520 0.13765791 0.12772327
## rush_tds 0.8860820 0.9125263 1.00000000 0.06914613 0.03114206 0.03163468
```

```
## rec_att  0.1970685 0.1874552 0.06914613 1.00000000 0.99002712 0.96757796
## rec_yds  0.1447372 0.1376579 0.03114206 0.99002712 1.00000000 0.98209522
## rec_tds  0.1354900 0.1277233 0.03163468 0.96757796 0.98209522 1.00000000
## fumbles  0.1844220 0.1881021 0.10845675 0.43577978 0.40349289 0.35852435
## fpts     0.1766540 0.1698501 0.06567865 0.98754942 0.99760259 0.99058639
##            fumbles       fpts
## rush_att 0.1844220 0.17665405
## rush_yds 0.1881021 0.16985010
## rush_tds 0.1084568 0.06567865
## rec_att  0.4357798 0.98754942
## rec_yds  0.4034929 0.99760259
## rec_tds  0.3585244 0.99058639
## fumbles  1.0000000 0.38269698
## fpts     0.3826970 1.00000000
```

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

```
##             rush_att   rush_yds    rush_tds      rec_att      rec_yds
## rush_att   5.3301775  32.187375 0.202350270 1.256726e+01 1.240286e+02
## rush_yds  32.1873748 198.075240 1.270338571 7.287276e+01 7.190963e+02
## rush_tds   0.2023503   1.270339 0.009784036 1.889207e-01 1.143345e+00
## rec_att   12.5672625  72.872757 0.188920688 7.629661e+02 1.015010e+04
## rec_yds  124.0286200 719.096342 1.143344727 1.015010e+04 1.377659e+05
## rec_tds    0.8143230   4.679547 0.008145937 6.957564e+01 9.489494e+02
## fumbles    0.1779604   1.106493 0.004483896 5.031061e+00 6.259601e+01
## fpts      21.3954750 125.403136 0.340808761 1.430998e+03 1.942477e+04
##               rec_tds      fumbles         fpts
## rush_att 8.143230e-01  0.177960412 2.139547e+01
## rush_yds 4.679547e+00  1.106492875 1.254031e+02
## rush_tds 8.145937e-03  0.004483896 3.408088e-01
## rec_att  6.957564e+01  5.031061456 1.430998e+03
## rec_yds  9.489494e+02 62.596006870 1.942477e+04
## rec_tds  6.776998e+00  0.390101180 1.352815e+02
## fumbles  3.901012e-01  0.174694759 8.391169e+00
## fpts     1.352815e+02  8.391169098 2.752042e+03
```

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

```
##     rush_att     rush_yds     rush_tds      rec_att      rec_yds
##   0.67901235   3.80288066   0.01152263  28.78971193 377.25020576
##      rec_tds      fumbles         fpts
##   2.34238683   0.32674897  51.58724280
```

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
```

```r
keep.1
```

```
##           rush_att  rush_yds   rush_tds    rec_att    rec_yds    rec_tds
## rush_att 1.0000000 0.9903162 0.88313477 0.18952925 0.13757736 0.12798676
## rush_yds 0.9903162 1.0000000 0.91005848 0.17987168 0.13040422 0.12021200
## rush_tds 0.8831348 0.9100585 1.00000000 0.06384226 0.02598521 0.02626808
## rec_att  0.1895293 0.1798717 0.06384226 1.00000000 0.98961588 0.96631027
## rec_yds  0.1375774 0.1304042 0.02598521 0.98961588 1.00000000 0.98145769
## rec_tds  0.1279868 0.1202120 0.02626808 0.96631027 0.98145769 1.00000000
## fumbles  0.1781950 0.1818948 0.10351025 0.43132799 0.39908030 0.35408860
## fpts     0.1689594 0.1620685 0.05997181 0.98703276 0.99750983 0.99024148
##            fumbles       fpts
## rush_att 0.1781950 0.16895942
## rush_yds 0.1818948 0.16206847
## rush_tds 0.1035103 0.05997181
## rec_att  0.4313280 0.98703276
## rec_yds  0.3990803 0.99750983
## rec_tds  0.3540886 0.99024148
## fumbles  1.0000000 0.37828188
## fpts     0.3782819 1.00000000
```

```r
cor(fb)
```

```
##           rush_att  rush_yds   rush_tds    rec_att    rec_yds    rec_tds
## rush_att 1.0000000 0.9906030 0.88608205 0.19706851 0.14473723 0.13548999
## rush_yds 0.9906030 1.0000000 0.91252627 0.18745520 0.13765791 0.12772327
## rush_tds 0.8860820 0.9125263 1.00000000 0.06914613 0.03114206 0.03163468
## rec_att  0.1970685 0.1874552 0.06914613 1.00000000 0.99002712 0.96757796
## rec_yds  0.1447372 0.1376579 0.03114206 0.99002712 1.00000000 0.98209522
## rec_tds  0.1354900 0.1277233 0.03163468 0.96757796 0.98209522 1.00000000
## fumbles  0.1844220 0.1881021 0.10845675 0.43577978 0.40349289 0.35852435
## fpts     0.1766540 0.1698501 0.06567865 0.98754942 0.99760259 0.99058639
##            fumbles       fpts
## rush_att 0.1844220 0.17665405
## rush_yds 0.1881021 0.16985010
## rush_tds 0.1084568 0.06567865
## rec_att  0.4357798 0.98754942
## rec_yds  0.4034929 0.99760259
## rec_tds  0.3585244 0.99058639
## fumbles  1.0000000 0.38269698
## fpts     0.3826970 1.00000000
```

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 0.9906030 0.88608205 0.19706851 0.14473723 0.13548999
## rush_yds 0.9906030 1.0000000 0.91252627 0.18745520 0.13765791 0.12772327
## rush_tds 0.8860820 0.9125263 1.00000000 0.06914613 0.03114206 0.03163468
## rec_att  0.1970685 0.1874552 0.06914613 1.00000000 0.99002712 0.96757796
## rec_yds  0.1447372 0.1376579 0.03114206 0.99002712 1.00000000 0.98209522
## rec_tds  0.1354900 0.1277233 0.03163468 0.96757796 0.98209522 1.00000000
```

```
## fumbles  0.1844220 0.1881021 0.10845675 0.43577978 0.40349289 0.35852435
## fpts     0.1766540 0.1698501 0.06567865 0.98754942 0.99760259 0.99058639
##            fumbles       fpts
## rush_att 0.1844220 0.17665405
## rush_yds 0.1881021 0.16985010
## rush_tds 0.1084568 0.06567865
## rec_att  0.4357798 0.98754942
## rec_yds  0.4034929 0.99760259
## rec_tds  0.3585244 0.99058639
## fumbles  1.0000000 0.38269698
## fpts     0.3826970 1.00000000
```

```
cor.fb
```

```
##           rush_att  rush_yds   rush_tds    rec_att    rec_yds    rec_tds
## rush_att 1.0000000 0.9906030 0.88608205 0.19706851 0.14473723 0.13548999
## rush_yds 0.9906030 1.0000000 0.91252627 0.18745520 0.13765791 0.12772327
## rush_tds 0.8860820 0.9125263 1.00000000 0.06914613 0.03114206 0.03163468
## rec_att  0.1970685 0.1874552 0.06914613 1.00000000 0.99002712 0.96757796
## rec_yds  0.1447372 0.1376579 0.03114206 0.99002712 1.00000000 0.98209522
## rec_tds  0.1354900 0.1277233 0.03163468 0.96757796 0.98209522 1.00000000
## fumbles  0.1844220 0.1881021 0.10845675 0.43577978 0.40349289 0.35852435
## fpts     0.1766540 0.1698501 0.06567865 0.98754942 0.99760259 0.99058639
##            fumbles       fpts
## rush_att 0.1844220 0.17665405
## rush_yds 0.1881021 0.16985010
## rush_tds 0.1084568 0.06567865
## rec_att  0.4357798 0.98754942
## rec_yds  0.4034929 0.99760259
## rec_tds  0.3585244 0.99058639
## fumbles  1.0000000 0.38269698
## fpts     0.3826970 1.00000000
```

Question 4

10 points

Use LaTeXto create the following expressions.

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

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

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

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

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)

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

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