Bios 6301: Assignment 3

Brooklyn Stanley

Due Thursday, 08 October, 1:00 PM

50 points total.

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(7)
powerTest <- function(n, trials, alpha = 0.05) {
  powerTally <- rep(0, trials)
  for (i in 1:trials) {
    treatment <- rbinom(n,1,0.5)
    outcome <- rnorm(n,60,20)
    for (patient in 1:n){
      if (treatment[patient] == 1) {
        outcome[patient] <- outcome[patient] + 5
      }
    }
    pval <- summary(lm(outcome ~ treatment))$coefficients[2,4]
    if (pval <= alpha) powerTally[i] <- 1
  }
  return(sum(powerTally)/trials)
}
powerTest(100,1000)
```

```
## [1] 0.222
```

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

```r
powerTest(1000,1000)
```

```
## [1] 0.986
```

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)

```r
football <- read.csv('proj_rb15.csv')
football <- football[ ,3:9]
(corFootball <- cor(football))
```

```
##           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
##            fumbles
## rush_att 0.8589364
## rush_yds 0.8583243
## rush_tds 0.8526904
## rec_att  0.7459076
## rec_yds  0.7224865
## rec_tds  0.6055598
## fumbles  1.0000000
```

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)

```r
library(MASS)
varFootball <- var(football)
meanFootball <- colMeans(football)

corAvg <- 0
sims <- 10000
for(i in 1:sims) {
  dataSim <- mvrnorm(30, mu=meanFootball, Sigma=varFootball)
  corAvg <- corAvg + cor(dataSim)/sims
}
corAvg # The simulated correlation matrix
```

```
##           rush_att  rush_yds  rush_tds   rec_att   rec_yds   rec_tds
## rush_att 1.0000000 0.9974580 0.9713314 0.7647592 0.7351435 0.5912361
## rush_yds 0.9974580 1.0000000 0.9766644 0.7595856 0.7291006 0.5962068
## rush_tds 0.9713314 0.9766644 1.0000000 0.7207256 0.6924619 0.5846423
## rec_att  0.7647592 0.7595856 0.7207256 1.0000000 0.9942162 0.8335852
## rec_yds  0.7351435 0.7291006 0.6924619 0.9942162 1.0000000 0.8472704
## rec_tds  0.5912361 0.5962068 0.5846423 0.8335852 0.8472704 1.0000000
## fumbles  0.8552712 0.8546177 0.8487268 0.7400539 0.7162023 0.5983954
##            fumbles
## rush_att 0.8552712
## rush_yds 0.8546177
## rush_tds 0.8487268
## rec_att  0.7400539
## rec_yds  0.7162023
## rec_tds  0.5983954
## fumbles  1.0000000
```

```
corFootball # The actual 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
##            fumbles
## rush_att 0.8589364
## rush_yds 0.8583243
## rush_tds 0.8526904
## rec_att  0.7459076
## rec_yds  0.7224865
## rec_tds  0.6055598
## fumbles  1.0000000
```

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

```
exact <- mvrnorm(30, mu=meanFootball, Sigma=varFootball, empirical=T)
```

Question 4

10 points

Use LaTeXto create the following expressions.

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

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

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

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

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

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

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

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

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