STAT 6289-11 Causal Inference
Homework 1

Eunice Wu

February 3, 2022

Problem 1

a) (2pt) Suppose X and Y have joint density $p(X, Y)$. How is $\mathbb{E}[Y \mid X]$ defined? (Write it out in terms of an integral and density function).

$\mathbb{E}[Y \mid X]=\int_Y y \cdot f_{Y \mid X}(y \mid x)\, dy=\int_Y y \cdot \frac{p(X, Y)}{f_X(x)}\, dy=\frac{1}{f_Y(x)} \int_Y y \cdot p(X, Y)\, dy$

b) (2pt) If X and Y were independent, what does $\mathbb{E}[X \mid Y]$ reduce to? (Show why this is.).

$\mathbb{E}[Y \mid X]=\int_Y y \cdot \frac{p(X, Y)}{f_X(x)}\, dy=\int_Y y \cdot \frac{f_X(x) \cdot f_Y(y)}{f_X(x)}\, dy=\int_Y y \cdot f_Y(y)\, dy=\mathbb{E}[Y]$

c) (4pt) Draw random variables $X_1, X_2, \ldots, X_N$, all independently from the marginal density of X. Let $\bar{X}=\frac{1}{N} \sum_i^N X_i$. Is $\bar{X}$ unbiased for $\mathbb{E}[X]$? Prove it. (Do not just cite a theorem!)

The mean $\bar{X}$ of a random sample is an unbiased estimate of the population moment $\mu=\mathbb{E}[X]$, as $\mathbb{E}(\bar{x})=\mathbb{E}\left(\sum \frac{X_i}{n}\right)=\frac{1}{n} \sum \mathbb{E}\left(X_i\right)=\frac{n}{n} \mu=\mu$

d) (4pt) Derive the variance of $\bar{X}$. What happens to it as $N \rightarrow \infty$?

$\operatorname{Var}(\bar{X})=\operatorname{Var}\left(\sum \frac{X_i}{n}\right)=\frac{1}{n^2} \sum \operatorname{Var}\left(X_i\right)=\frac{n}{n^2} \sigma^2=\frac{\sigma^2}{n}$ When $N \rightarrow \infty$, $\operatorname{Var}(\bar{X}) \rightarrow \infty$

Problem 2

Consider random variables $Y \in \mathbb{R}$ and $X \in \mathbb{R}^{\mathbb{P}}$, drawn from joint density $p(X, Y)$. You collect a sample of draws from this distribution, $\{(Y_1, X_i), \ldots, (Y_N, X_N)\}$. Let $\mathbf{X}$ be a $N \times(1+P)$ matrix, with row i equal to $\left[1 X_i^{\top}\right]$ (i.e., there is an intercept and then a column for each "covariate"). Consider an OLS model, $Y=\mathbf{X} \beta+\epsilon$, where $\mathbb{E}[\epsilon \mid X]=0$.

a) (4pt) Using matrix notation at each step, derive the ordinary least squares estimator for β :

$$\hat{\beta}=\underset{\beta \in \mathbb{R}^{P+1}}{\operatorname{argmin}}(\mathbf{Y}-\mathbf{X} \beta)^{\top}(\mathbf{Y}-\mathbf{X} \beta)$$

$$\left[\begin{array}{c} Y_1 \\ Y_2 \\ \vdots \\ \vdots \\ Y_N \end{array}\right]_{N \times 1}=\left[\begin{array}{ccccc} 1 & X_{11} & X_{21} & \ldots & X_{p1} \\ 1 & X_{12} & X_{22} & \ldots & X_{p2} \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ \vdots & \vdots & \vdots & \ldots & \vdots \\ 1 & X_{1n} & X_{2n} & \ldots & X_{pN} \end{array}\right]_{N \times(p+1)}\left[\begin{array}{c} \beta_0 \\ \beta_2 \\ \vdots \\ \beta_p \end{array}\right]_{(p+1) \times 1}+\left[\begin{array}{c} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \vdots \\ \varepsilon_N \end{array}\right]_{N \times 1}$$

$$\begin{aligned}
\varepsilon^T\varepsilon &= (\mathbf{Y}-\mathbf{X}\hat{\beta})^T(\mathbf{Y}-\mathbf{X}\hat{\beta}) \\
&= \mathbf{Y}^T\mathbf{Y}-\hat{\beta}^T\mathbf{X}^T\mathbf{Y}-\mathbf{Y}^T\mathbf{X}\hat{\beta}+\hat{\beta}^T\mathbf{X}^T\mathbf{X}\hat{\beta} \\
&= \mathbf{Y}^T\mathbf{Y}-2\hat{\beta}^T\mathbf{X}^T\mathbf{Y}+\hat{\beta}^T\mathbf{X}'\mathbf{X}\hat{\beta}
\end{aligned}$$

$$\frac{\partial\varepsilon^T\varepsilon}{\partial\hat{\beta}} = -2\mathbf{X}'\mathbf{Y}+2\mathbf{X}'\mathbf{X}\hat{\beta}=0 \quad \frac{\partial^2\varepsilon^T\varepsilon}{\partial\hat{\beta}^2} = 2\mathbf{X}^T\mathbf{X} > 0$$

$$\begin{aligned}
\Rightarrow & \left(\mathbf{X}^T\mathbf{X}\right)\hat{\beta} = \mathbf{X}^T\mathbf{Y} \\
& \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\left(\mathbf{X}^T\mathbf{X}\right)\hat{\beta} = \left(\mathbf{X}^T X\right)^{-1}\mathbf{X}^T\mathbf{Y} \\
& I\hat{\beta} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y} \\
& \hat{\beta} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y}
\end{aligned}$$

b) (5pt) Show R code that would achieve the following:
i. Construct a matrix X to represent $\mathbf{X}$ in the above, with $N = 100$, one column of ones, and two columns of randomly drawn numbers (from any distribution you like).

```
N = 100
X <- cbind(matrix(1:1, ncol = 1, nrow = N),  matrix(rpois(N, 5), ncol = 1),
           matrix(rpois(N, 10), ncol = 1))
head(X)
```

```
##      [,1] [,2] [,3]
## [1,]    1    6   13
## [2,]    1    5   14
## [3,]    1    4    4
## [4,]    1    7    8
## [5,]    1    1   12
## [6,]    1    4   13
```

ii. Using $\beta = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}^\top$, compute vector Y equal to $X\beta+\epsilon$, where ϵ is drawn from a standard normal distribution.

```
beta = c(1,2,3)
epsilon = matrix(rnorm(N, mean=0, sd=1), ncol = 1)
beta
```

```
## [1] 1 2 3
```

```
head(epsilon)
```

```
##            [,1]
## [1,]  0.3260207
## [2,]  0.0932062
## [3,] -0.2524867
## [4,] -0.2877227
## [5,]  0.6339740
## [6,] -0.6630345
```

```
Y = X %*% beta + epsilon
head(Y)
```

```
##          [,1]
## [1,] 52.32602
## [2,] 53.09321
## [3,] 20.74751
## [4,] 38.71228
## [5,] 39.63397
## [6,] 47.33697
```

iii. Compute $\left(X^{\top}X\right)^{-1}\left(X^{\top}Y\right)$. Use the solve function in R.

```
beta_est = solve(t(X) %*% X)  %*% t(X)  %*% Y
beta_est

##           [,1]
## [1,] 0.6751192
## [2,] 1.9567181
## [3,] 3.0473170
```

iv. Compare the result to the coefficients obtained using lm with the data you have constructed.

```
coeff_lm <- lm(Y ~ X[,2]+X[,3])
summary(coeff_lm)

##
## Call:
## lm(formula = Y ~ X[, 2] + X[, 3])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.15919 -0.54278  0.04758  0.52232  2.39986
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.67512    0.37629   1.794   0.0759 .
## X[, 2]       1.95672    0.04659  42.001   <2e-16 ***
## X[, 3]       3.04732    0.03161  96.401   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9595 on 97 degrees of freedom
## Multiple R-squared:  0.9922,Adjusted R-squared:  0.992
## F-statistic:  6136 on 2 and 97 DF,  p-value: < 2.2e-16
```

c) (5pt) Prove the unbiasedness of $\hat{\beta}$ for β.

With Matrix notation:
$\hat{\beta} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{Y} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T(\mathbf{X}\beta+\varepsilon) = \beta + \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\varepsilon$
And $\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{X} = I$.
So $\mathbb{E}[\hat{\beta}] = \mathbb{E}[\beta] + \mathbb{E}\left[\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\varepsilon\right] = \beta + \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbb{E}[\varepsilon]$
$\mathbb{E}[\varepsilon] = 0$. So
$\mathbb{E}(\hat{\beta}) = \mathbb{E}(\beta) + \mathbb{E}\left[\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\varepsilon\right] = \beta + \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbb{E}\left[\mathbf{X}^T\varepsilon\right]$
As $\mathbb{E}\left(\mathbf{X}^T\varepsilon\right) = 0$, $\mathbb{E}(\hat{\beta}) = \beta$

d) (5pt) Compute the variance, with $\mathbf{X}$ taken as fixed (not random), i.e. $\mathbb{V}[\hat{\beta} \mid \mathbf{X}]$, again sticking with matrix notation.

You may assume $\mathbb{E}\left[\epsilon\epsilon^{\top} \mid \mathbf{X}\right] = \sigma^2 I_N$, where I_N is the $N \times N$ identity matrix.

$$\begin{aligned}
\mathbb{E}\left[(\hat{\beta}-\beta)(\hat{\beta}-\beta)^T\right] &= \mathbb{E}\left[\left(\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\varepsilon\right)\left(\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\varepsilon\right)^T\right] \\
&= \mathbb{E}\left[\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\varepsilon\varepsilon^T\mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\right] \\
&= \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbb{E}\left[\varepsilon\varepsilon^T\right]\mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1} \\
&= \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\left(\sigma^2 I\right)\mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1} \\
&= \sigma^2 I\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1} \\
&= \sigma^2\left(\mathbf{X}^T\mathbf{X}\right)^{-1}
\end{aligned}$$

Therefore,

$$\operatorname{Var}(\hat{\beta}) = \mathbb{E}\left[(\hat{\beta}-\beta)(\hat{\beta}-\beta)^T\right] = \begin{bmatrix} \operatorname{var}\left(\hat{\beta}_1\right) & \operatorname{cov}\left(\hat{\beta}_1,\hat{\beta}_2\right) & \ldots & \operatorname{cov}\left(\hat{\beta}_1,\hat{\beta}_k\right) \\ \operatorname{cov}\left(\hat{\beta}_2,\hat{\beta}_1\right) & \operatorname{var}\left(\hat{\beta}_2\right) & \ldots & \operatorname{cov}\left(\hat{\beta}_2,\hat{\beta}_k\right) \\ \vdots & \vdots & \vdots & \vdots \\ \operatorname{cov}\left(\hat{\beta}_k,\hat{\beta}_1\right) & \operatorname{cov}\left(\hat{\beta}_k,\hat{\beta}_2\right) & \ldots & \operatorname{var}\left(\hat{\beta}_k\right) \end{bmatrix}$$

e) (2pt) What meaning would you give to the matrix $\mathbb{E}\left[\epsilon\epsilon^{\top} \mid \mathbf{X}\right]$? Give an intuitive explanation of what the assumption that this matrix equals $\sigma^2 I$ implies.

The variance of ε_i is the same for all X_i and ε_i's are independent.

Problem 3

Which of the following statements are true or false? For credit, explain your choice briefly
a) (3pt) If there is perfect collinearity, the OLS estimator will give biased and inconsistent estimates.

Not true. The coefficients on the collinear terms might have infinite variance if the collinearity is extreme while the estimator remains unbiased.

b) (3pt) A very large p-value for the estimated coefficient for an explanatory variable provides strong evidence that the variable has zero effect on the outcome.

Not true. If the p-value is larger than the confidence level during F-test, we rejected the null hypothesis and it can be concluded that there is at least one coefficient is not zero.

c) (3pt) If an estimator is unbiased it is also consistent.

Not true.

d) (3pt) You have a model $Y = X^{\top}\beta + \epsilon$, and you fit it by OLS. If the OLS residuals are uncorrelated with X, our estimate of β are unbiased.

True. $\mathbb{E}(\hat{\beta}) = \beta + \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbb{E}\left[\mathbf{X}^T\varepsilon\right]$ If $\mathbb{E}\left(\mathbf{X}^T\varepsilon\right) = \mathbb{E}(\varepsilon) = 0, \mathbb{E}(\hat{\beta}) = \beta$

Problem 4

Suppose $X_1 \sim N(5, 2)$ and $X_2 \sim \exp(\lambda = 1)$ (where $\exp$ indicates the exponential distribution). In R, construct two vectors, X1 with 10000 draws from the same distribution as X_1, and X2 with 10000 draws from the same distribution as X_2.

We will take sub-samples from these two variables to evaluate the coverage probability of 95% confidence intervals using different types of data and different sample sizes.

(a) (4pt) Describe the distribution of X1 and X2 using histograms. Mark the true (population) expectation on your plot using a line.

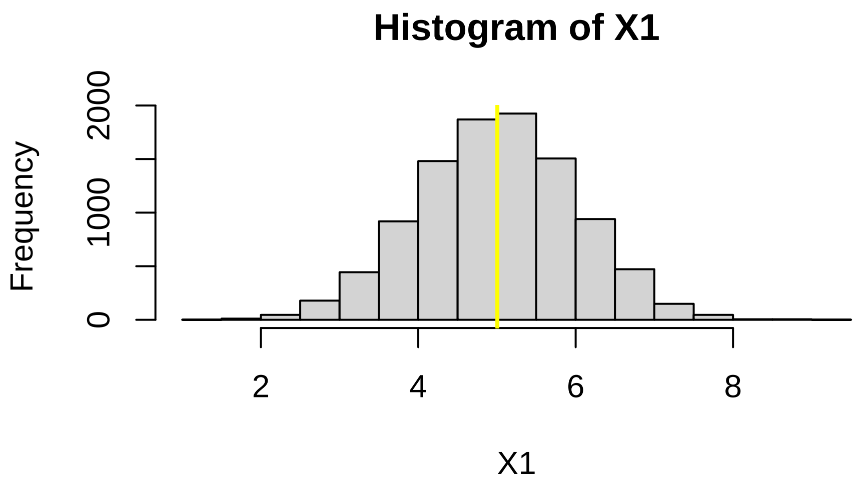
```
X1 <- matrix(rnorm(10000, mean = 5, sd = 1),ncol = 1)
X2 <- matrix(rpois(10000, 1),ncol = 1)
head(X1)
```

```
##          [,1]
## [1,] 5.257544
## [2,] 4.818645
## [3,] 6.778251
## [4,] 3.542059
## [5,] 4.783530
## [6,] 5.105978
```

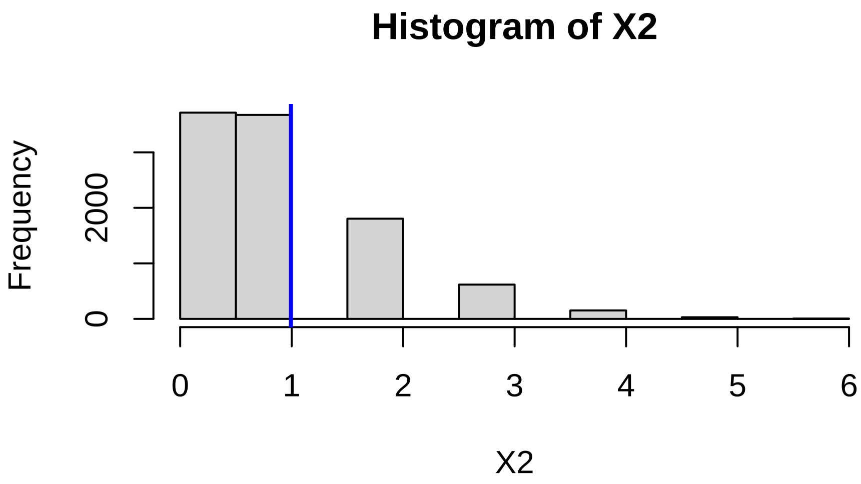
```
head(X2)
```

```
##      [,1]
## [1,]    1
## [2,]    0
## [3,]    0
## [4,]    2
## [5,]    2
## [6,]    3
```

```
hist(X1)
abline(v = mean(X1), col = "yellow", lwd = 2)
```



```
hist(X2)
abline(v = mean(X2), col = "blue", lwd = 2)
```



(b) (4pts) Consider for a moment the random variables $\bar{X}_1$ and $\bar{X}_2$, representing sample means. Using math (not R), give solutions for:
- $\mathbb{E}\left[\bar{X}_1\right]$?
- $\mathbb{E}\left[\bar{X}_2\right]$?
- Var $\left(\bar{X}_1\right)$?
- Var $\left(\bar{X}_2\right)$?

$\mathbb{E}\left[\bar{X}_1\right] = 5$
$\mathbb{E}\left[\bar{X}_2\right] = 1$
$\text{Var}\left(\bar{X}_1\right) = 2$
$\text{Var}\left(\bar{X}_2\right) = 1$

(c) (4pts). Now, for X_1, draw 5000 sub-samples each of size $N = 6$. Get the sample mean and compute the 95% confidence interval each time. What portion of the confidence intervals you computed include the true expectation? Repeat this for X_2.

```
X1_sub_sample <- matrix(sample(X1, size = 5000 *6, replace = TRUE),6, 5000)

head(X1_sub_sample,5:6)

##          [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
## [1,] 3.085447 5.183113 5.758012 4.273961 5.115787 5.731389
## [2,] 5.329938 3.545789 5.154577 5.053304 3.081665 4.560495
## [3,] 4.433019 7.033668 5.493081 3.634116 5.217159 6.808585
## [4,] 5.897117 5.747834 4.940110 5.033851 3.941513 4.967928
## [5,] 5.856407 5.832355 4.256276 4.470993 5.094903 4.124827

X1_sub_sample_mean = apply(X1_sub_sample, 2, mean)

#head(X1_sub_sample_mean)

X1_sub_sample_ci = apply(X1_sub_sample, 2, function(x) ci(x))

head(X1_sub_sample_ci,5:6)

##                 [,1]     [,2]      [,3]      [,4]      [,5]      [,6]
## Estimate   5.0559698 5.506918 5.1162317 4.7669467 4.7436136 5.0104831
## CI lower   3.8917212 4.311981 4.5757271 3.8715399 3.6432567 3.8567714
## CI upper   6.2202184 6.701856 5.6567364 5.6623536 5.8439705 6.1641948
## Std. Error 0.4529125 0.464851 0.2102655 0.3483285 0.4280575 0.4488134
```

```
sum(apply(X1_sub_sample_ci, 2, function(x){ifelse(x[2] < 5 & x[3] > 5,
                                                  TRUE, FALSE)}))
```

```
## [1] 4752
```

```
X2_sub_sample <- matrix(sample(X2, size = 5000 *6, replace = TRUE),6, 5000)

head(X2_sub_sample,5:6)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    0    1    2    0    1    1
## [2,]    1    2    1    0    1    2
## [3,]    2    0    1    1    0    0
## [4,]    0    2    0    3    0    1
## [5,]    1    1    1    0    0    2
```

```
X2_sub_sample_mean = apply(X2_sub_sample, 2, mean)

X2_sub_sample_ci = apply(X2_sub_sample, 2, function(x) ci(x))

head(X2_sub_sample_ci,5:6)
```

```
##                  [,1]       [,2]      [,3]       [,4]        [,5]      [,6]
## Estimate   0.83333333 1.00000000 1.0000000  0.8333333  0.50000000 1.1666667
## CI lower   0.04334688 0.06135623 0.3362786 -0.3935044 -0.07479957 0.3766802
## CI upper   1.62331978 1.93864377 1.6637214  2.0601710  1.07479957 1.9566531
## Std. Error 0.30731815 0.36514837 0.2581989  0.4772607  0.22360680 0.3073181
```

```
sum(apply(X2_sub_sample_ci, 2, function(x){ifelse(x[2] < 1 & x[3] > 1,
                                                  TRUE, FALSE)}))
```

```
## [1] 4652
```

(d) (4pt) Repeat (c) for samples of size 6, 20, 50, and 500. Report the coverage probability for each of your eight simulations in a table. How do your results change? What differences do you see between X1 and X2?

```
#X1 sampling

X1_size_6 = sum(apply(apply(matrix(sample(X1, size = 5000 *6, replace = TRUE),
                                   6,5000), 2, function(x) ci(x)), 2,
                      function(x){ifelse(x[2] <5 & x[3] > 5, TRUE, FALSE)}))

X1_size_20 = sum(apply(apply(matrix(sample(X1, size = 5000 *20, replace = TRUE),
                                   20,5000), 2, function(x) ci(x)), 2,
                      function(x){ifelse(x[2] < 5 & x[3] > 5, TRUE, FALSE)}))
X1_size_50 = sum(apply(apply(matrix(sample(X1, size = 5000 *50, replace = TRUE),
                                   50,5000), 2, function(x) ci(x)), 2,
                      function(x){ifelse(x[2] < 5 & x[3] > 5, TRUE, FALSE)}))
```

```
X1_size_500 = sum(apply(apply(matrix(sample(X1, size = 5000 *500, replace = TRUE),
                                   500,5000), 2, function(x) ci(x)), 2,
                      function(x){ifelse(x[2] < 5 & x[3] > 5, TRUE, FALSE)}))
X1_sampling <- data.frame("size = 6" = c(6, X1_size_6),
                          "size = 20" = c(20, X1_size_20),
                          "size = 50" = c(50, X1_size_50),
                          "size = 500" = c(500, X1_size_500))

X1_sampling
```

```
##   size...6 size...20 size...50 size...500
## 1        6        20        50        500
## 2     4763      4744      4750       4793
```

```
#X2 sampling

X2_size_6 = sum(apply(apply(matrix(sample(X2, size = 5000 *6, replace = TRUE),
                                   6,5000), 2, function(x) ci(x)), 2,
                      function(x){ifelse(x[2] < 1 & x[3] > 1, TRUE, FALSE)}))

X2_size_20 = sum(apply(apply(matrix(sample(X2, size = 5000 *20, replace = TRUE),
                                   20,5000), 2, function(x) ci(x)), 2,
                      function(x){ifelse(x[2] < 1 & x[3] > 1, TRUE, FALSE)}))
X2_size_50 = sum(apply(apply(matrix(sample(X2, size = 5000 *50, replace = TRUE),
                                   50,5000), 2, function(x) ci(x)), 2,
                      function(x){ifelse(x[2] < 1 & x[3] > 1, TRUE, FALSE)}))
X2_size_500 = sum(apply(apply(matrix(sample(X2, size = 5000 *500, replace = TRUE),
                                   500,5000), 2, function(x) ci(x)), 2,
                      function(x){ifelse(x[2] < 1 & x[3] > 1, TRUE, FALSE)}))
X2_sampling <- data.frame("size = 6" = c(6, X2_size_6),
                          "size = 20" = c(20, X2_size_20),
                          "size = 50" = c(50, X2_size_50),
                          "size = 500" = c(500, X2_size_500))

X2_sampling
```

```
##   size...6 size...20 size...50 size...500
## 1        6        20        50        500
## 2     4625      4669      4742       4727
```

(e)(4pt) Explain your findings in parts (c) and (d).

The probability of confidence interval in each sampling to include population mean is converging to 95%.

Problem 5

We're going to use the mtcars dataset that can be found in the R package "datasets". Import the dataset by running "library(datasets); data(mtcars)". Use "?mtcars" in R to see a description of the data.
(a). (5pt) Fit a logistic regression model with the variable am as the response and mpg and hp as predictors. What are the estimated regression coefficients from this model? How do we interpret them here?

```r
logistic_regression <- glm(data = mtcars, am ~ hp + mpg, family = "binomial")
logistic_regression
```

```
##
## Call:  glm(formula = am ~ hp + mpg, family = "binomial", data = mtcars)
##
## Coefficients:
## (Intercept)           hp          mpg
##   -33.60517      0.05504      1.25961
##
## Degrees of Freedom: 31 Total (i.e. Null);  29 Residual
## Null Deviance:     43.23
## Residual Deviance: 19.23  AIC: 25.23
```

```r
summary(logistic_regression)
```

```
##
## Call:
## glm(formula = am ~ hp + mpg, family = "binomial", data = mtcars)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.41460  -0.42809  -0.07021   0.16041   1.66500
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -33.60517   15.07672  -2.229   0.0258 *
## hp            0.05504    0.02692   2.045   0.0409 *
## mpg           1.25961    0.56747   2.220   0.0264 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 43.230  on 31  degrees of freedom
## Residual deviance: 19.233  on 29  degrees of freedom
## AIC: 25.233
##
## Number of Fisher Scoring iterations: 7
```

As the p-value is less than 0.05, there is a statistically significant association between the response variable (am) and the predictors (mpg and hp).
$\ln \frac{p}{1-p} = -33.60517 + 0.05504 \cdot \mathrm{hp} + 1.25961 \cdot \mathrm{mpg}$

(b). (5pt) What is the predicted probability that a car is automatic if it has $\mathrm{hp} = 180$ and $\mathrm{mpg} = 20$?

$$\frac{p}{1-p} = \exp(-33.60517) \cdot \exp(0.05504 \cdot 180) \cdot \exp(1.25961 \cdot 20) = 4.4559 \quad \Rightarrow p = 0.816712$$

(c). (5pt) Randomly split the data into a 80% train set and a 20% test set. Fit a logistic model on the training set and predict the transmission type on the test set. What is the prediction accuracy of transmission type on the test set?

(Hint: if the probability of being 1 is greater than 0.5 then set the transmission type equal to 1 , otherwise, set it to 0)

```
split <- sample.split(mtcars, SplitRatio = 0.8)
training_set <- subset(mtcars, split == "TRUE")
test_set <- subset(mtcars, split == "FALSE")

logistric_training_set <- glm(data = training_set, am ~ hp + mpg, family = "binomial")

res <- predict(logistric_training_set, training_set, type = "response")

confmatrix <- table(Actual_value = training_set$am, Predicted_value = res >0.5)
confmatrix
```

```
##              Predicted_value
## Actual_value FALSE TRUE
##            0    12    2
##            1     2    7
```

```
(confmatrix[[1,1]] + confmatrix[[2,2]]) / sum(confmatrix)
```

```
## [1] 0.826087
```

Problem 6

We will work on the "Smarket" data, which is part of the ISLR library in R. Take a look at the dataset by running "library(ISLR); summary(Smarket)". This data set consists of percentage returns for the S&P 500 stock index over 1,250 days, from the beginning of 2001 until the end of 2005. For each date, we have recorded the percentage returns for each of the five previous trading days, Lag1 through Lag5. We have also recorded Volume (the number of shares traded on the previous day, in billions), Today (the percentage return on the date in question) and Direction (whether the market was Up or Down on this date).
(a). (5pt) Fit a logistic regression model to predict "Direction" using Lag1 through Lag5 and Volume. What are the estimated regression coefficients from this model? How do we interpret them here?

```
library(ISLR)
data(Smarket)

Smarket<- Smarket %>%
      mutate(Direction = ifelse(Direction == "Up",1,0))
```

```
log_reg <- glm(data = Smarket, Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
                 Volume, family = "binomial")
summary(log_reg)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = "binomial", data = Smarket)
##
```

```
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.446  -1.203   1.065   1.145   1.326
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.126000   0.240736  -0.523    0.601
## Lag1        -0.073074   0.050167  -1.457    0.145
## Lag2        -0.042301   0.050086  -0.845    0.398
## Lag3         0.011085   0.049939   0.222    0.824
## Lag4         0.009359   0.049974   0.187    0.851
## Lag5         0.010313   0.049511   0.208    0.835
## Volume       0.135441   0.158360   0.855    0.392
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1731.2  on 1249  degrees of freedom
## Residual deviance: 1727.6  on 1243  degrees of freedom
## AIC: 1741.6
##
## Number of Fisher Scoring iterations: 3
```

As all the p-values are larger than 0.05, there is no significant evidence in this sample to conclude that a non-zero correlation exists.
We can drop the predictors with the larger p-values.
$\ln \frac{p}{1-p} = -0.126000 + (-0.073074) \cdot \text{Lag1} + (-0.042301) \cdot \text{Lag2} + 0.135441 \cdot \text{Volume}$

(b) (5pt) Predict the probability that the market will go up, given values of the predictors in this dataset.

```
test_stock <- predict(log_reg, Smarket, type = "response")
summary(test_stock)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4084  0.5020  0.5180  0.5184  0.5338  0.6486
```

It is more likely to go up according to this logistic regression result.

(c) (5pt) Predict whether the market will go up or down on the values of the predictors in this dataset. What is the prediction accuracy (this is your training accuracy)?

```
confmatrix_stock <- table(Actual_value = Smarket$Direction,
                          Predicted_value = test_stock >0.5)
confmatrix_stock
```

```
##             Predicted_value
## Actual_value FALSE TRUE
##            0   145  457
##            1   141  507
```

```
(confmatrix_stock[[1,1]] + confmatrix_stock[[2,2]]) / sum(confmatrix_stock)
```

```
## [1] 0.5216
```

(d) (5pt) Now we fit the model using the past data, and then examine how well it predicts future data, which is more like the reality. To implement this strategy, first create the training data corresponding to the observations from 2001 through 2004 and testing data from the observations in 2005 . Fit a logistic regression model using the training data and use the model to predict for the testing data. What is the prediction accuracy (this is your testing accuracy)?

```
Smarket_2004 <- Smarket %>%  filter(Year < 2005)
Smarket_2005 <- Smarket %>%  filter(Year > 2004)

log_reg_2004 <- glm(data = Smarket_2004, Direction ~
                      Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume,
                    family = "binomial")

test_stock_2005 <- predict(log_reg_2004, Smarket_2005, type = "response")

confmatrix_stock <- table(Actual_value = Smarket_2005$Direction,
                          Predicted_value = test_stock_2005 >0.5)
confmatrix_stock
```

```
##              Predicted_value
## Actual_value FALSE TRUE
##            0    77   34
##            1    97   44
```

```
(confmatrix_stock[[1,1]] + confmatrix_stock[[2,2]]) / sum(confmatrix_stock)
```

```
## [1] 0.4801587
```