W271 Assignment 1

Due 11:59pm Pacific Time, Sunday February 2, 2020

Hersh Solanki

Instructions (Please Read Carefully):

- **Late submissions will not be accepted**
- No page limit, but be reasonable
- Do not modify fontsize, margin or line_spacing settings
- This assignment needs to be completed individually; this is not a group project
- Submission is by pushing to your student fork of the course repository
- Submit two files:
  1. A pdf file that details your answers (knit to pdf, do not knit to html then save as pdf). Include all R code used to produce the answers. Do not suppress the code in your pdf file
  2. The R markdown (Rmd) file used to produce the pdf file

  The assignment will not be graded unless **both** files are submitted
- Use the following file-naming convensation:
  - StudentFirstNameLastName_HWNumber.fileExtension
  - For example, if the student's name is Kyle Cartman for assignment 1, name your files follows:
    * KyleCartman_assignment1.Rmd
    * KyleCartman_assignment1.pdf
- Although it sounds obvious, please write your name on page 1 of your pdf and Rmd files
- Answers should clearly explain your reasoning; do not simply 'output dump' the results of code without explanation
- For statistical methods that we cover in this course, use the R libraries and functions that are covered in this course. If you use libraries and functions for statistical modeling that we have not covered, you must provide an explanation of why such libraries and functions are used and reference the library documentation. For data wrangling and data visualization, you are free to use other libraries, such as dplyr, ggplot2, etc.
- For mathematical formulae, type them in your R markdown file. Do not e.g. write them on a piece of paper, snap a photo, and use the image file.
- Incorrectly following submission instructions results in deduction of grades
- Students are expected to act with regard to UC Berkeley Academic Integrity

1. Confidence Intervals (2 points)

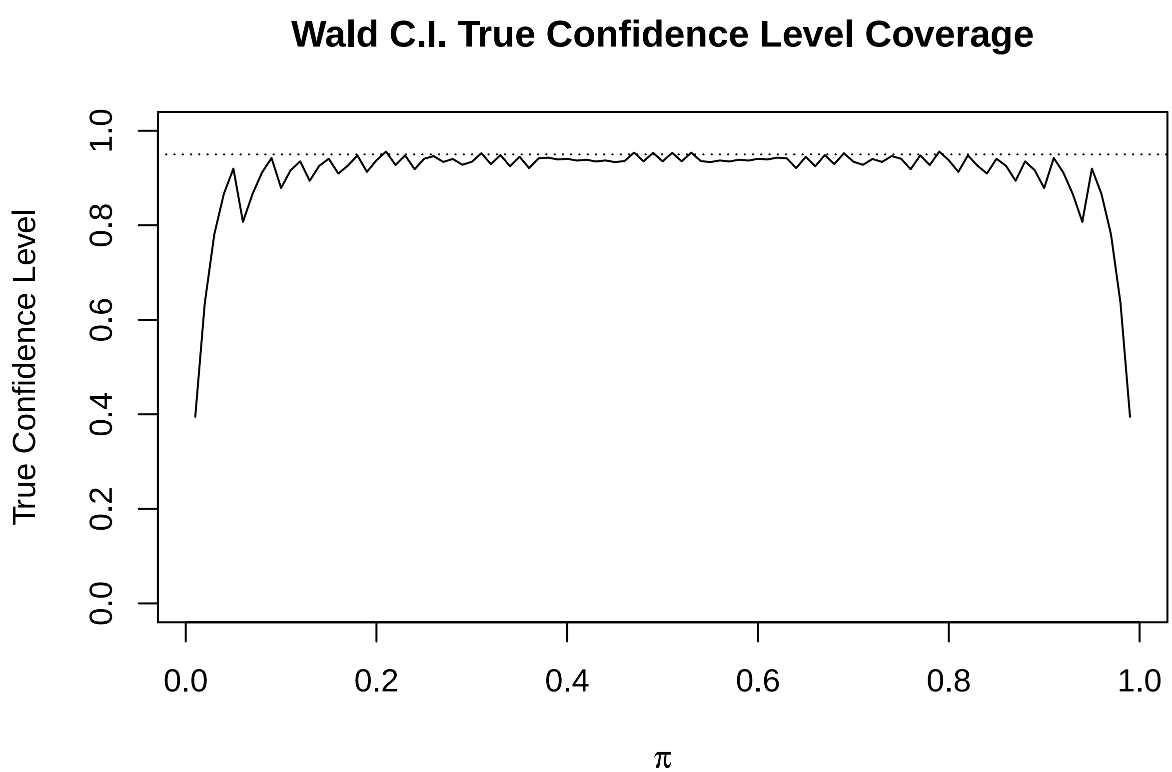
A Wald confidence interval for a binary response probability does not always have the stated confidence level, $1-\alpha$, where α (the probability of rejecting the null hypothesis when it is true) is often set to 0.05%. This was demonstrated with code in the week 1 live session file.

Question 1.1: Use the code from the week 1 live session file and: (1) redo the exercise for `n=50, n=100, n=500`, (2) plot the graphs, and (3) describe what you have observed from the results. Use the same `pi.seq` as in the live session code.

```
one.one = function(n) {

pi = 0.6
alpha = 0.05
n = n
w = 0:n

wald.CI.true.coverage = function(pi, alpha=0.05, n) {

    w = 0:n

    pi.hat = w/n
    pmf = dbinom(x=w, size=n, prob=pi)

    var.wald = pi.hat*(1-pi.hat)/n
    wald.CI_lower.bound = pi.hat - qnorm(p = 1-alpha/2)*sqrt(var.wald)
    wald.CI_upper.bound = pi.hat + qnorm(p = 1-alpha/2)*sqrt(var.wald)

    covered.pi = ifelse(test = pi>wald.CI_lower.bound, yes = ifelse(test = pi<wald.CI_upper.bou

    wald.CI.true.coverage = sum(covered.pi*pmf)

    wald.df = data.frame(w, pi.hat, round(data.frame(pmf, wald.CI_lower.bound,wald.CI_upper.bou

    return(wald.df)
  }

  wald.df = wald.CI.true.coverage(pi=0.6, alpha=0.05, n=n)
  wald.CI.true.coverage.level = sum(wald.df$covered.pi*wald.df$pmf)

  # Let's compute the ture coverage for a sequence of pi
  pi.seq = seq(0.01,0.99, by=0.01)
  wald.CI.true.matrix = matrix(data=NA,nrow=length(pi.seq),ncol=2)
  counter=1
  for (pi in pi.seq) {
      wald.df2 = wald.CI.true.coverage(pi=pi, alpha=0.05, n=n)
      #print(paste('True Coverage is', sum(wald.df2$covered.pi*wald.df2$pmf)))
      wald.CI.true.matrix[counter,] = c(pi,sum(wald.df2$covered.pi*wald.df2$pmf))
```

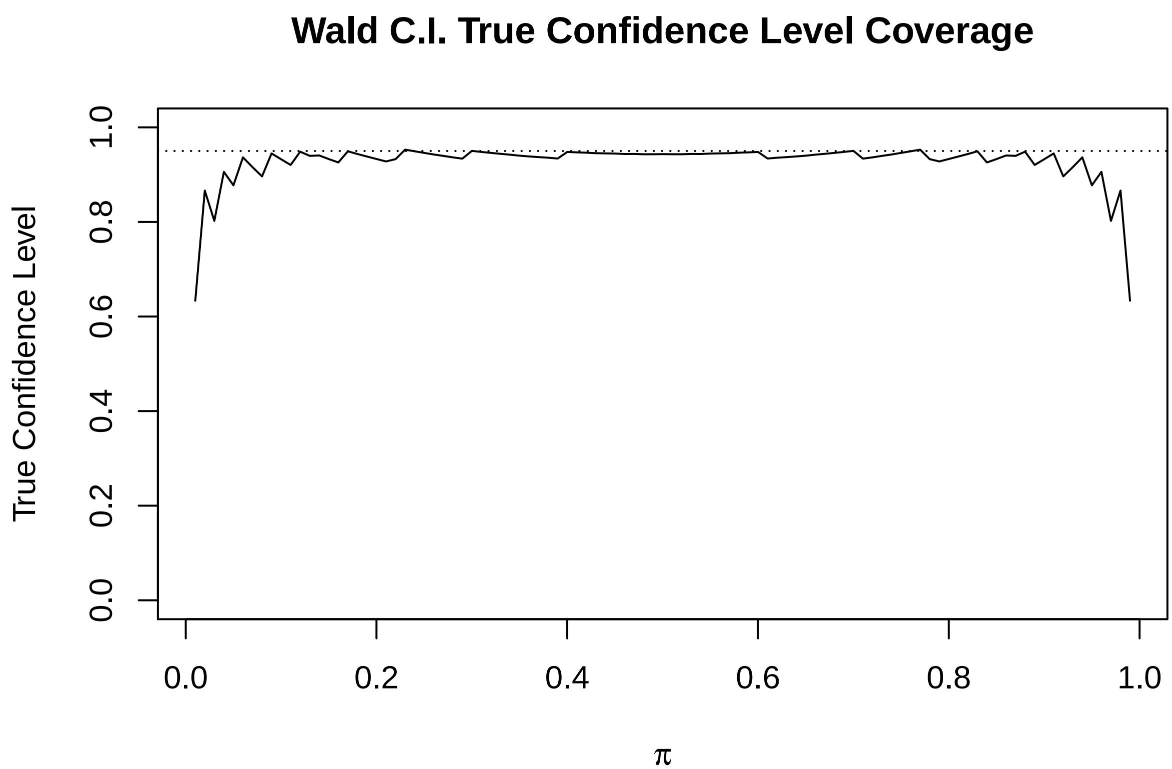
```r
      counter = counter+1
  }
  str(wald.CI.true.matrix)
  wald.CI.true.matrix[1:5,]

  # Plot the true coverage level (for given n and alpha)
  plot(x=wald.CI.true.matrix[,1],
       y=wald.CI.true.matrix[,2],
       ylim=c(0,1),
       main = "Wald C.I. True Confidence Level Coverage", xlab=expression(pi),
       ylab="True Confidence Level",
       type="l")
  abline(h=1-alpha, lty="dotted")
}

one.one(50)
```

```
##  num [1:99, 1:2] 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1 ...
```



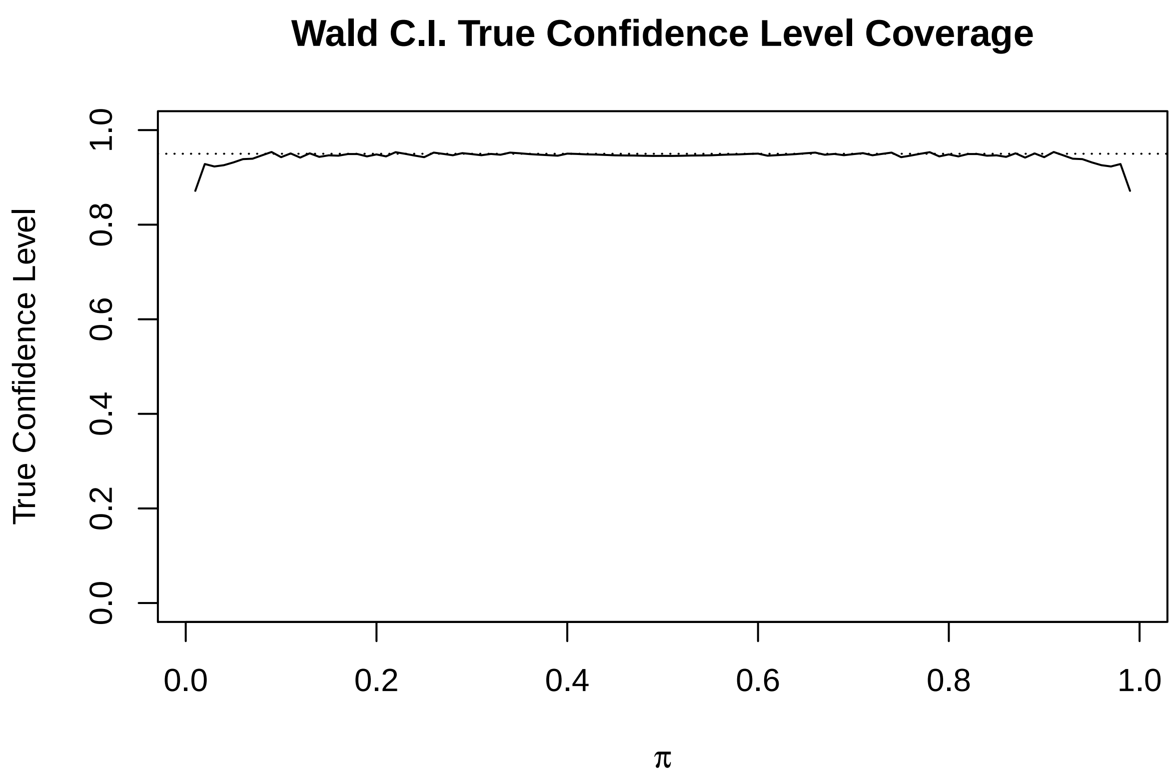

```r
one.one(100)
```

```
##  num [1:99, 1:2] 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1 ...
```




```r
one.one(500)
```

```
##  num [1:99, 1:2] 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1 ...
```



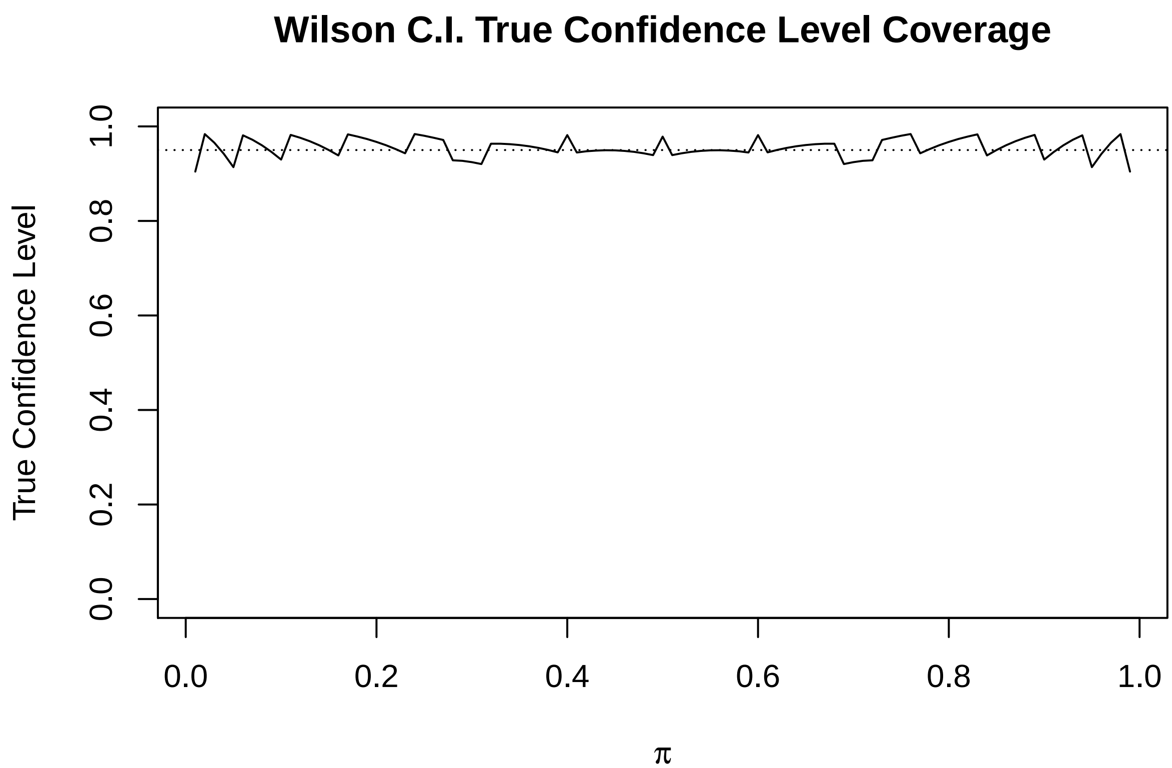
Question 1.2: (1) Modify the code for the Wilson Interval. (2) Do the exercise for `n=10`, `n=50`,

`n=100`, `n=500`. (3) Plot the graphs. (4) Describe what you have observed from the results and compare the Wald and Wilson intervals based on your results. Use the same `pi.seq` as in the live session code.

```
two.two = function(n) {

pi = 0.6
alpha = 0.05
n = n
w = 0:n

wilson.CI.true.coverage = function(pi, alpha=0.05, n) {

    w = 0:n

    pi.hat = w/n
    pmf = dbinom(x=w, size=n, prob=pi)

    p.tilde <- (w + qnorm(p = 1-alpha/2)^2 / 2) / (n + qnorm(p = 1-alpha/2)^2)

    wilson.CI_lower.bound = round(p.tilde - qnorm(p = 1-alpha/2) * sqrt(n) / (n + qnorm(p = 1-a
1-alpha/2)^2/(4*n)), 4)

    wilson.CI_upper.bound = round(p.tilde + qnorm(p = 1-alpha/2) * sqrt(n) / (n + qnorm(p = 1-a
1-alpha/2)^2/(4*n)), 4)

    # + ((qnorm(p = 1-alpha/2))^2/4*n)
    covered.pi = ifelse(test = pi>wilson.CI_lower.bound, yes = ifelse(test = pi<wilson.CI_uppe

    wilson.CI.true.coverage = sum(covered.pi*pmf)

    wilson.df = data.frame(w, pi.hat, round(data.frame(pmf, wilson.CI_lower.bound,wilson.CI_up

    return(wilson.df)
  }

  wilson.df = wilson.CI.true.coverage(pi=0.6, alpha=0.05, n=n)
  wilson.CI.true.coverage.level = sum(wilson.df$covered.pi*wilson.df$pmf)

  # Let's compute the ture coverage for a sequence of pi
  pi.seq = seq(0.01,0.99, by=0.01)
  wilson.CI.true.matrix = matrix(data=NA,nrow=length(pi.seq),ncol=2)
  counter=1
  for (pi in pi.seq) {
      wilson.df2 = wilson.CI.true.coverage(pi=pi, alpha=0.05, n=n)
      #print(paste('True Coverage is', sum(wald.df2$covered.pi*wald.df2$pmf)))
      wilson.CI.true.matrix[counter,] = c(pi,sum(wilson.df2$covered.pi*wilson.df2$pmf))
      counter = counter+1
```

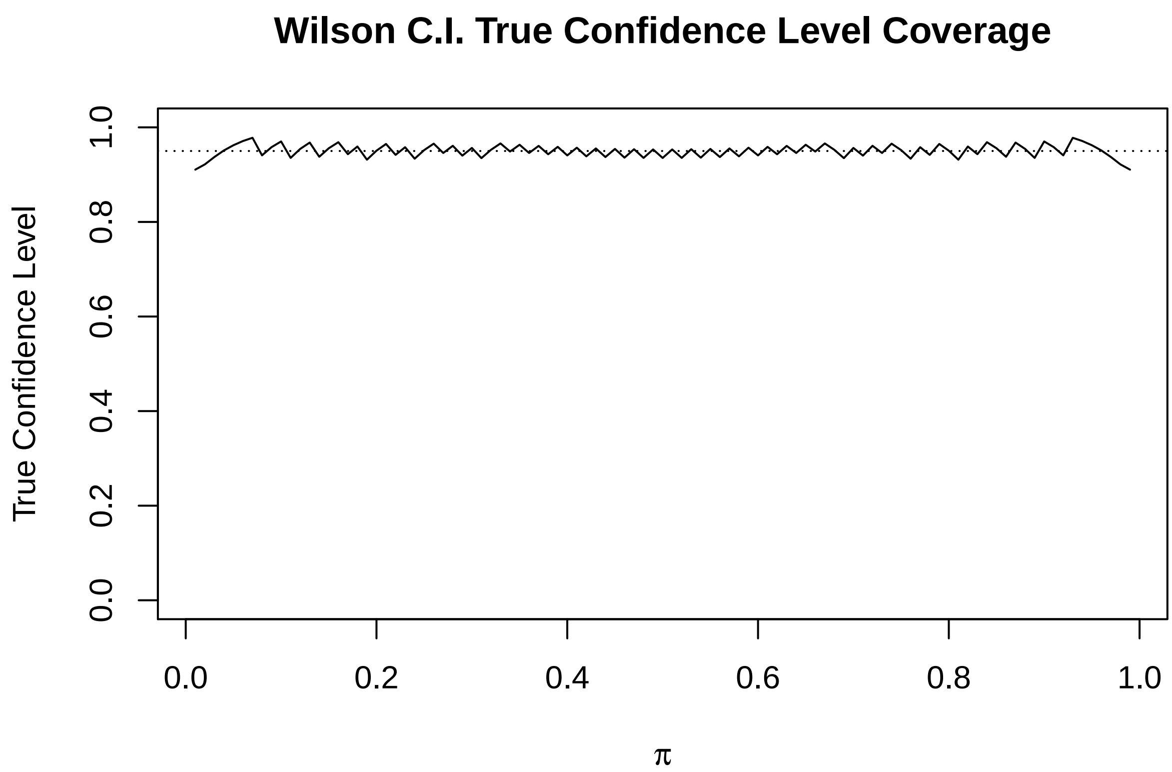
```
  }
  str(wilson.CI.true.matrix)
  wilson.CI.true.matrix[1:5,]

  # Plot the true coverage level (for given n and alpha)
  plot(x=wilson.CI.true.matrix[,1],
       y=wilson.CI.true.matrix[,2],
       ylim=c(0,1),
       main = "Wilson C.I. True Confidence Level Coverage", xlab=expression(pi),
       ylab="True Confidence Level",
       type="l")
  abline(h=1-alpha, lty="dotted")
}

two.two(10)
```

```
##  num [1:99, 1:2] 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1 ...
```



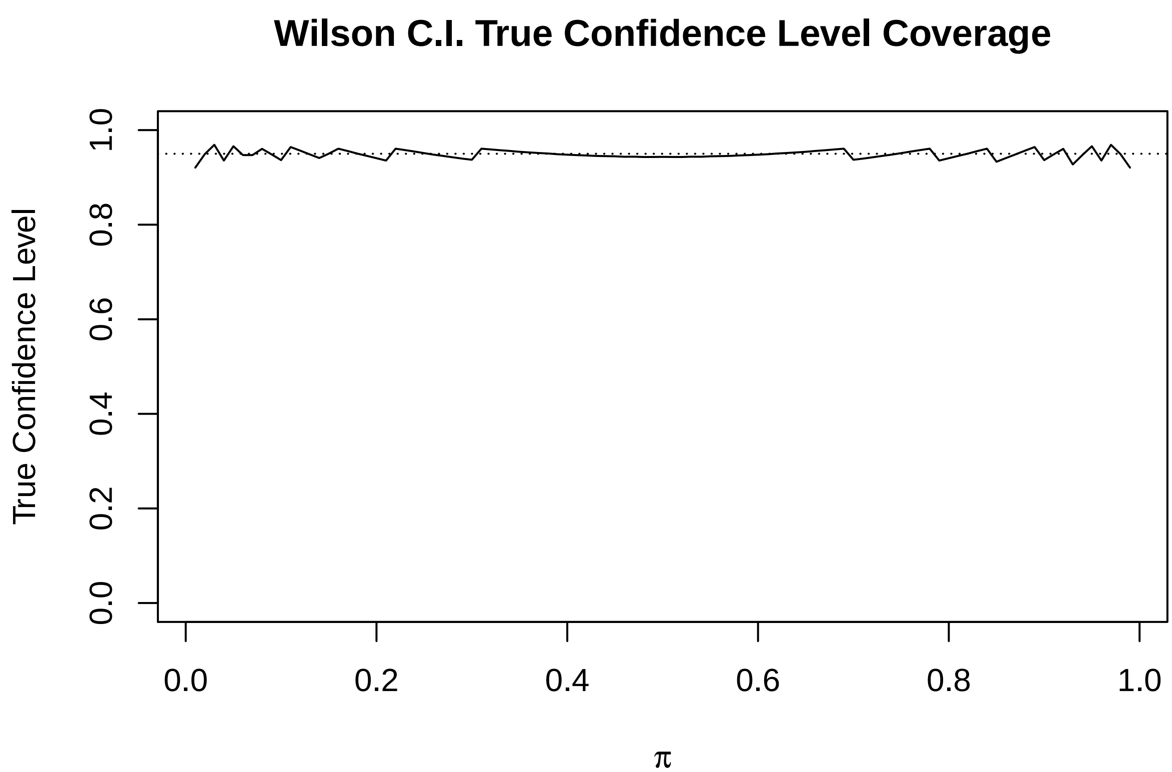

```
two.two(50)
```

```
##  num [1:99, 1:2] 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1 ...
```



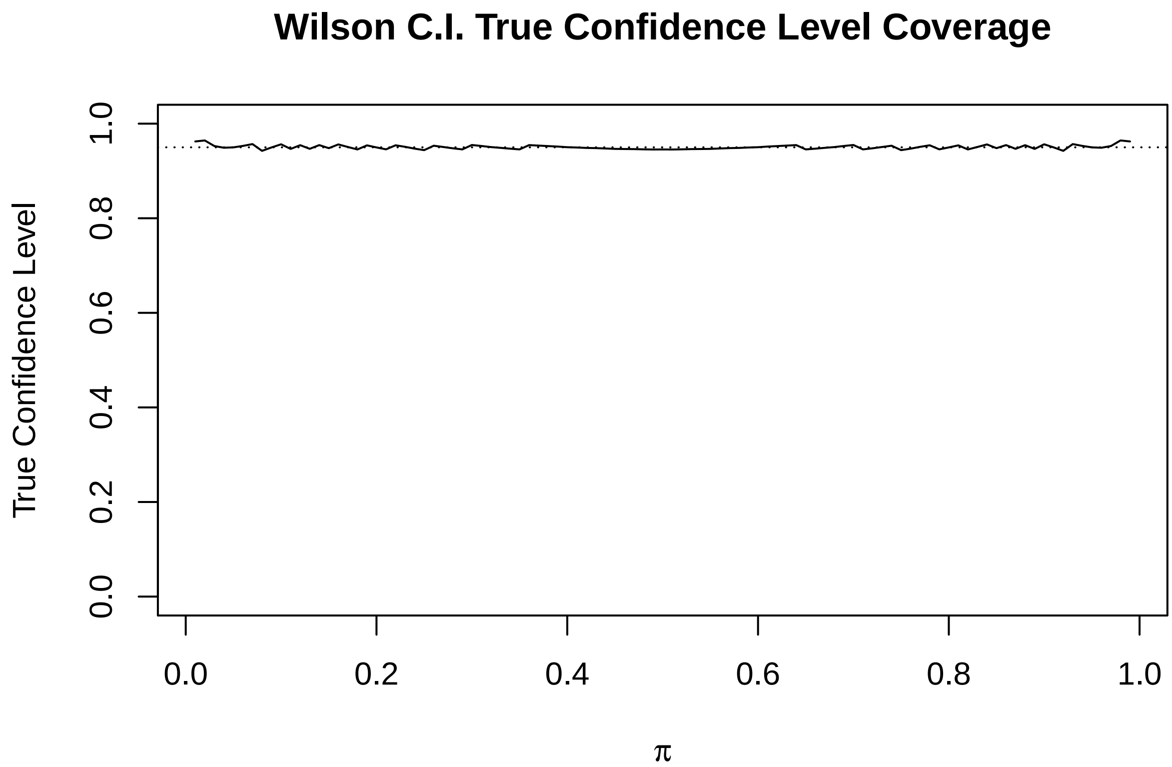

```r
two.two(100)
```

```
##  num [1:99, 1:2] 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1 ...
```



```r
two.two(500)
```

```
##  num [1:99, 1:2] 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09 0.1 ...
```



2: Binary Logistic Regression (2 points)

Do Exercise 8 a, b, c, and d on page 131 of Bilder and Loughin's textbook. Please write down each of the questions. The dataset for this question is stored in the file *"placekick.BW.csv"* which is provided to you.

In general, all the R codes and datasets used in Bilder and Loughin's book are provided on the book's website: chrisbilder.com

For **question 8b**, in addition to answering the question, re-estimate the model in part (a) using "*Sun*" as the base level category for *Weather*.

```
library(car)
placekick <- read.csv("placekick.BW.csv", header=TRUE, sep=",")
head(placekick)
```

```
##     GameNum Kicker Good Distance Weather Wind15 Temperature Grass Pressure
## 1 2002-0101 Bryant    Y       29     Sun      0        Nice     1        N
## 2 2002-0101 Bryant    Y       33     Sun      0        Nice     1        N
## 3 2002-0101 Cortez    N       25     Sun      0        Nice     1        N
## 4 2002-0101 Cortez    Y       23     Sun      0        Nice     1        N
## 5 2002-0101 Cortez    N       48     Sun      0        Nice     1        N
## 6 2002-0101 Cortez    Y       33     Sun      0        Nice     1        N
##   Ice
## 1   0
## 2   0
## 3   0
## 4   0
## 5   0
## 6   0
```

Continuing Exercise 7, use the Distance, Weather, Wind15, Temperature, Grass, Pressure, and Ice explanatory variables as linear terms in a new logistic regression model and complete the following:

(a) Estimate the model and properly define the indicator variables used within it.

```
model.2.a <- glm(Good ~ Distance + Weather + Wind15 + Temperature + Grass + Pressure + Ice, fa
summary(model.2.a)
```

```
##
## Call:
## glm(formula = Good ~ Distance + Weather + Wind15 + Temperature +
##     Grass + Pressure + Ice, family = binomial(link = logit),
##     data = placekick)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6804   0.2599   0.4360   0.7148   1.8698
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)      5.740185   0.369597  15.531   <2e-16 ***
## Distance        -0.109600   0.007188 -15.249   <2e-16 ***
## WeatherInside   -0.083030   0.214711  -0.387   0.6990
## WeatherSnowRain -0.444193   0.217852  -2.039   0.0415 *
## WeatherSun      -0.247582   0.139642  -1.773   0.0762 .
## Wind15          -0.243777   0.175527  -1.389   0.1649
## TemperatureHot   0.250013   0.247540   1.010   0.3125
## TemperatureNice  0.234932   0.181461   1.295   0.1954
## Grass           -0.328435   0.160050  -2.052   0.0402 *
## PressureY        0.270174   0.262809   1.028   0.3039
## Ice             -0.876133   0.451251  -1.942   0.0522 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2104.0  on 2002  degrees of freedom
## Residual deviance: 1791.3  on 1992  degrees of freedom
## AIC: 1813.3
##
## Number of Fisher Scoring iterations: 5
```

```r
model.2.a$coefficients
```

```
##     (Intercept)        Distance   WeatherInside WeatherSnowRain
##      5.74018455     -0.10959961     -0.08302951     -0.44419298
##      WeatherSun          Wind15  TemperatureHot TemperatureNice
##     -0.24758206     -0.24377683      0.25001316      0.23493183
##           Grass       PressureY             Ice
##     -0.32843455      0.27017353     -0.87613251
```

The indicator variables are Weather(4 levels) and Temperature (3 levels). Other onces include Wind15, Grass, Pressure, and Ice (2 levels).

The model equation is: Y = 5.74018455(Intercept) - 0.10959961(Distance) - 0.08302951(WeatherInside) - 0.44419298(WeatherSnowRain) - 0.24758206(WeatherSun) - 0.24377683(Wind15) + 0.25001316(TemperatureHot) + 0.23493183(TemperatureNice) - 0.32843455(Grass) + 0.27017353(PressureY) - 0.87613251(Ice).

(b) The authors use "Sun" as the base level category for Weather, which is not the default level that R uses. Describe how "Sun" can be specified as the base level in R.

```r
placekick$Weather <- relevel(placekick$Weather, ref = "Sun")

model.2.b <- glm(Good ~ Distance + Weather + Wind15 + Temperature + Grass + Pressure + Ice, fa
summary(model.2.b)
```

```
##
## Call:
## glm(formula = Good ~ Distance + Weather + Wind15 + Temperature +
##     Grass + Pressure + Ice, family = binomial(link = logit),
```

```
##     data = placekick)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6804   0.2599   0.4360   0.7148   1.8698
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)       5.492602   0.370141  14.839   <2e-16 ***
## Distance         -0.109600   0.007188 -15.249   <2e-16 ***
## WeatherClouds     0.247582   0.139642   1.773   0.0762 .
## WeatherInside     0.164553   0.215062   0.765   0.4442
## WeatherSnowRain -0.196611   0.219015  -0.898   0.3693
## Wind15           -0.243777   0.175527  -1.389   0.1649
## TemperatureHot    0.250013   0.247540   1.010   0.3125
## TemperatureNice   0.234932   0.181461   1.295   0.1954
## Grass            -0.328435   0.160050  -2.052   0.0402 *
## PressureY         0.270174   0.262809   1.028   0.3039
## Ice              -0.876133   0.451251  -1.942   0.0522 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2104.0  on 2002  degrees of freedom
## Residual deviance: 1791.3  on 1992  degrees of freedom
## AIC: 1813.3
##
## Number of Fisher Scoring iterations: 5
```

```r
model.2.b$coefficients
```

```
##     (Intercept)        Distance   WeatherClouds   WeatherInside
##       5.4926025      -0.1095996       0.2475821       0.1645526
## WeatherSnowRain          Wind15  TemperatureHot TemperatureNice
##      -0.1966109      -0.2437768       0.2500132       0.2349318
##           Grass       PressureY             Ice
##      -0.3284346       0.2701735      -0.8761325
```

In order to change the base categoery, we use relevel, and specificy what we want it to be. In the case above, we are using sun.

The model equation is: Y = 5.4926025(Intercept) - 0.1095996(Distance) - 0.2475821(WeatherClouds) - 0.1645526(WeatherInside) - 0.1966109(WeatherSnowRain) - 0.2437768(Wind15) + 0.2500132(TemperatureHot) + 0.2349318(TemperatureNice) - 0.3284346(Grass) + 0.2701735(PressureY) - 0.8761325(Ice).

(c) Perform LRTs for all explanatory variables to evaluate their importance within the model. Discuss the results.

```r
Anova(model.2.a, test = "LR")
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: Good
##             LR Chisq Df Pr(>Chisq)
## Distance     294.341  1  < 2e-16 ***
## Weather        5.670  3  0.12884
## Wind15         1.898  1  0.16833
## Temperature    1.723  2  0.42254
## Grass          4.314  1  0.03781 *
## Pressure       1.088  1  0.29682
## Ice            3.698  1  0.05448 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Distance is the most significant, with a p value of 2e-16. The presence of grass is important as well, with a p balue of 0.037, which is significant at the 5% level. Ice is significant at the 10% level.

(d) Estimate an appropriate odds ratio for distance, and compute the corresponding confidence interval. Interpret the odds ratio.

```r
print("Odds Ratio")
```

```
## [1] "Odds Ratio"
```

```r
exp(model.2.a$coefficients[2])
```

```
##  Distance
## 0.8961929
```

```r
print("Confidence Interval")
```

```
## [1] "Confidence Interval"
```

```r
beta.ci <- confint(object = model.2.a, parm = "Distance", level = 0.95)
as.numeric(exp(beta.ci))
```

```
## [1] 0.8834276 0.9086871
```

```r
print("-----------------------")
```

```
## [1] "-----------------------"
```

```r
# Alternatively, we can do it only on the significant variables. For this, we get:
model.2.d <- model.2.b <- glm(Good ~ Distance + Grass + Ice, family = binomial(link = logit), d
print("Odds Ratio")
```

```
## [1] "Odds Ratio"
```

```r
exp(model.2.d$coefficients[2])
```

```
##  Distance
```

```
## 0.8981461
```

```r
print("Confidence Interval")
```

```
## [1] "Confidence Interval"
```

```r
beta.ci <- confint(object = model.2.d, parm = "Distance", level = 0.95)
as.numeric(exp(beta.ci))
```

```
## [1] 0.8855381 0.9104894
```

Interpretation - odds of success changes by 0.896 times for every 1 year decrease in distance of the kick.

3: Binary Logistic Regression (2 points)

The dataset *"admissions.csv"* contains a small sample of graduate school admission data from a university. The variables are specificed below:

1. admit - the depenent variable that takes two values: $0, 1$ where 1 denotes *admitted* and 0 denotes *not admitted*
2. gre - GRE score
3. gpa - College GPA
4. rank - rank in college major

Suppose you are hired by the University's Admission Committee and are charged to analyze this data to quantify the effect of GRE, GPA, and college rank on admission probability. We will conduct this analysis by answering the follwing questions:

Question 3.1: Examine the data and conduct EDA

```
library(psych)

admissions <- read.csv("admissions.csv", header=TRUE, sep=",")
admissions$X <- NULL

# Basic EDA
print("DESCRIBE")
```

```
## [1] "DESCRIBE"
```

```
describe(admissions)
```

```
##       vars   n   mean     sd median trimmed    mad    min max  range  skew
## admit    1 400   0.32   0.47    0.0    0.27   0.00   0.00   1   1.00  0.78
## gre      2 400 587.70 115.52  580.0  589.06 118.61 220.00 800 580.00 -0.14
## gpa      3 400   3.39   0.38    3.4    3.40   0.40   2.26   4   1.74 -0.21
## rank     4 400   2.48   0.94    2.0    2.48   1.48   1.00   4   3.00  0.10
##       kurtosis   se
## admit    -1.39 0.02
## gre      -0.36 5.78
## gpa      -0.60 0.02
## rank     -0.91 0.05
```

```
print("SUMMARY")
```

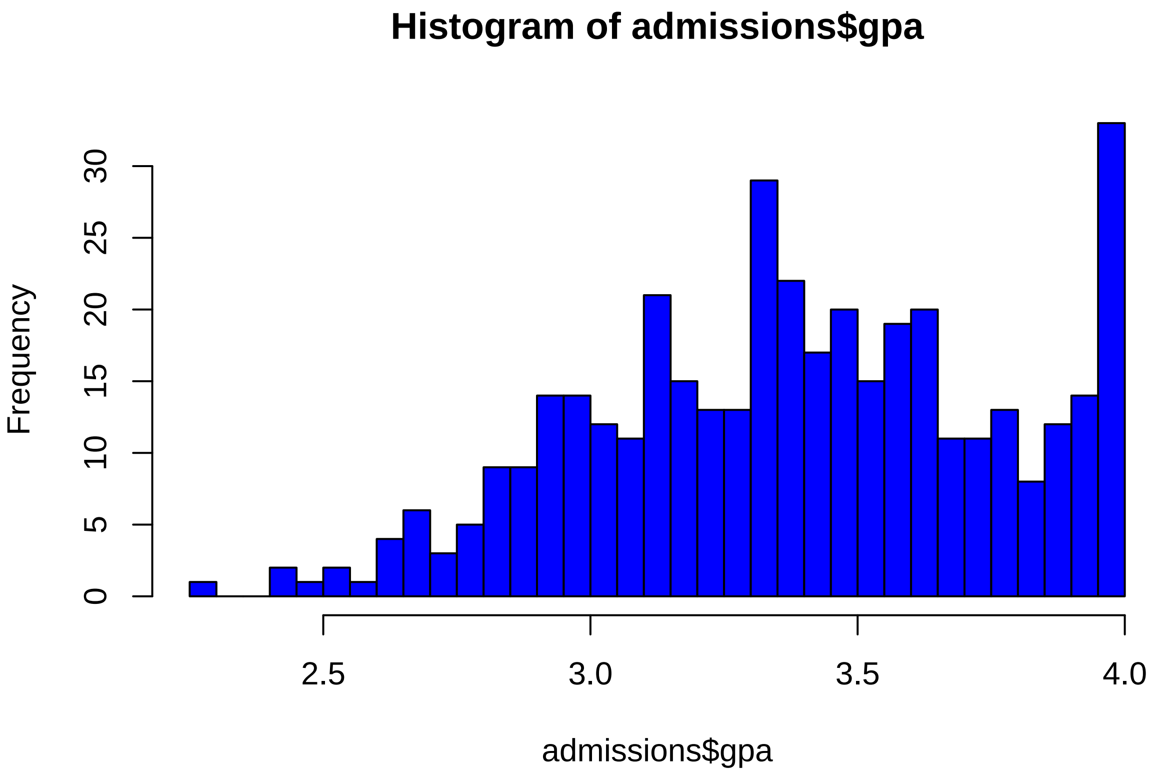
```
## [1] "SUMMARY"
```

```
summary(admissions)
```

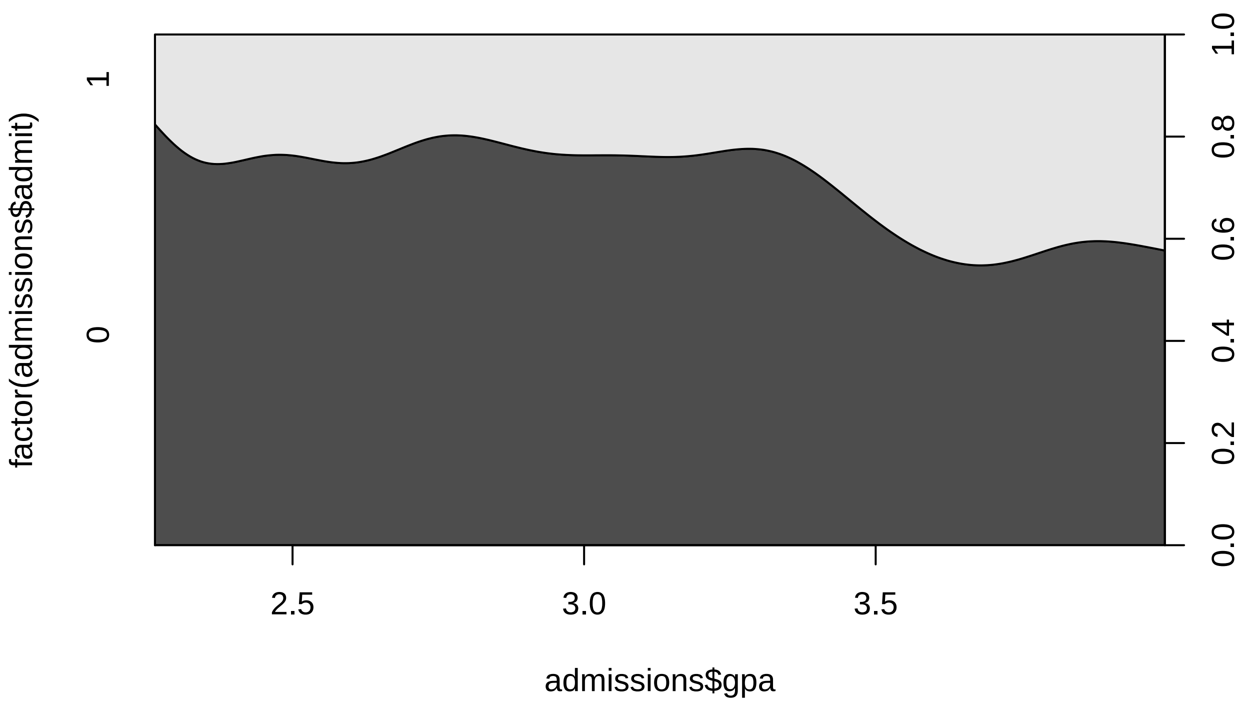
```
##      admit             gre             gpa             rank      
##  Min.   :0.0000   Min.   :220.0   Min.   :2.260   Min.   :1.000  
##  1st Qu.:0.0000   1st Qu.:520.0   1st Qu.:3.130   1st Qu.:2.000  
##  Median :0.0000   Median :580.0   Median :3.395   Median :2.000  
```

```
## Mean   :0.3175   Mean   :587.7   Mean   :3.390   Mean   :2.485
## 3rd Qu.:1.0000   3rd Qu.:660.0   3rd Qu.:3.670   3rd Qu.:3.000
## Max.   :1.0000   Max.   :800.0   Max.   :4.000   Max.   :4.000
```

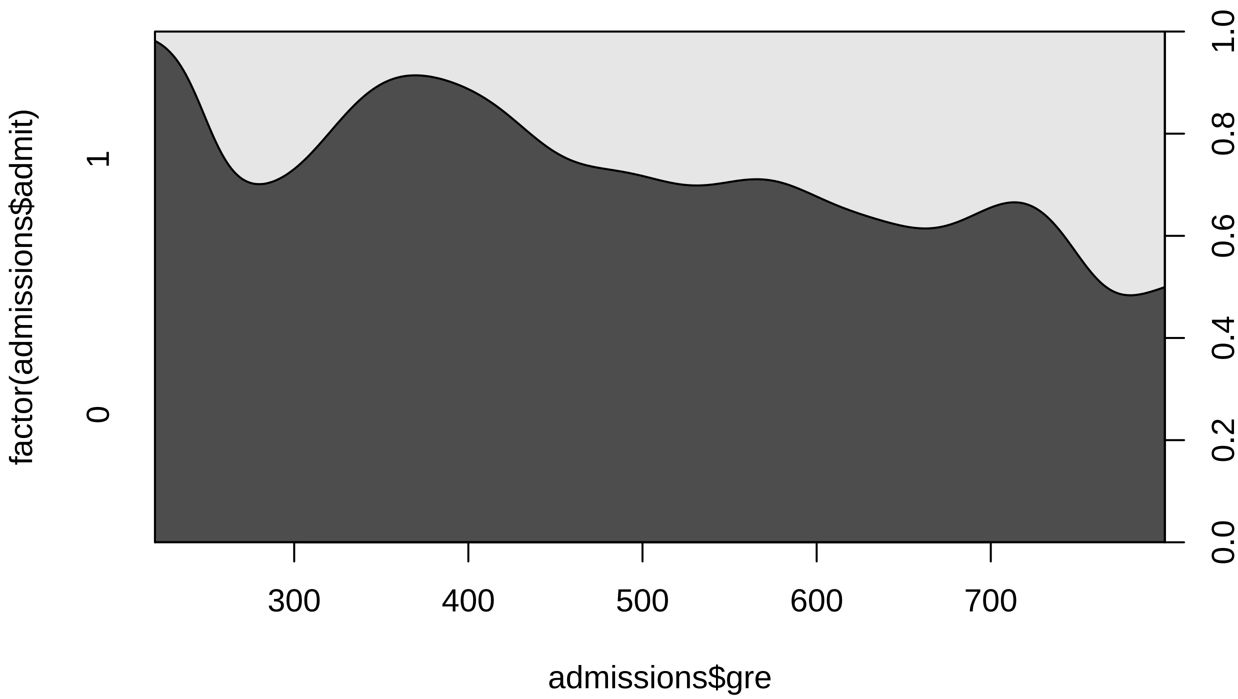
```r
hist(admissions$gpa, breaks = 40, col="blue")
```



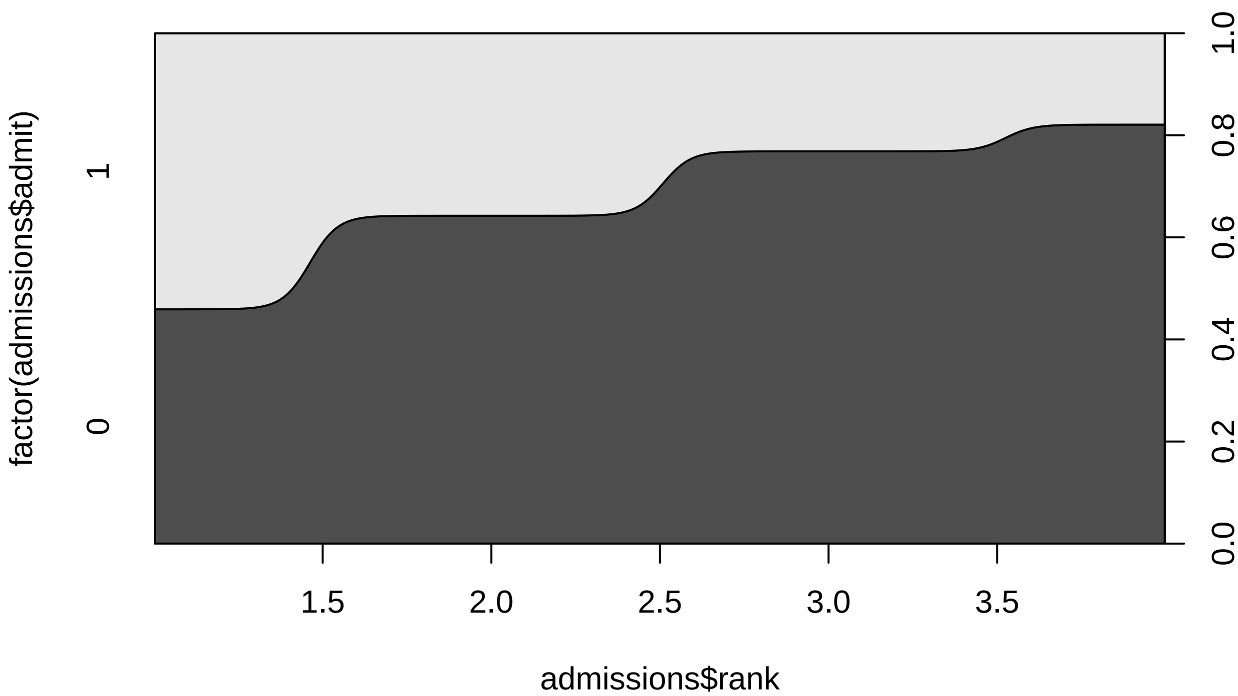

```r
cdplot(factor(admissions$admit) ~ admissions$gpa)
```



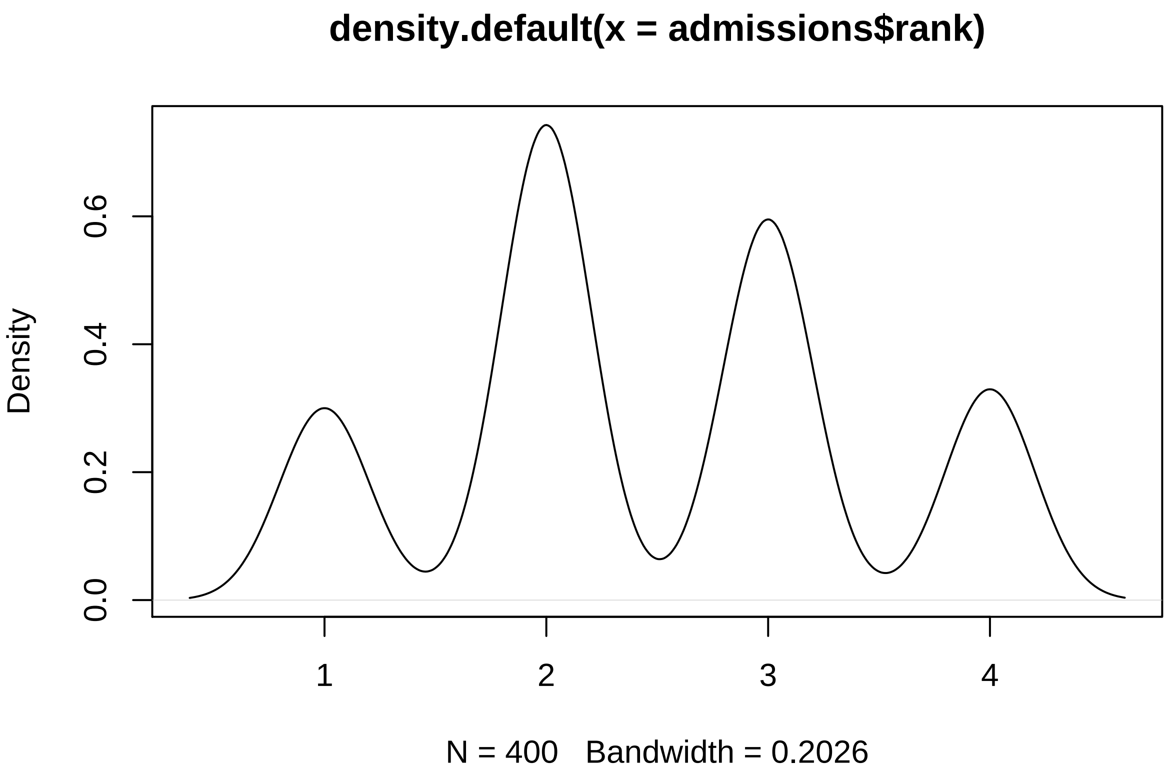
```r
cdplot(factor(admissions$admit) ~ admissions$gre)
```




```r
cdplot(factor(admissions$admit) ~ admissions$rank)
```




```r
plot(density(admissions$rank))
```



Biggest point is that there is no missing data. The means/SD can be seen in describe. It seems as if rank is the biggest predictor of admission, with a significant cliff at every rank level.

Question 3.2: Estimate a binary logistic regression using the following set of explanatory variables: *gre*, *gpa*, *rank*, gre^2, gpa^2, and $gre \times gpa$, where $gre \times gpa$ denotes the interaction between *gre* and *gpa* variables

Here, I make rank a factor. This is because it is not a continous variable.

```
model.3.2 <- glm(formula = admit ~ gre + gpa + factor(rank) + I(gre^2) + I(gpa^2) + gpa:gre, fa
summary(model.3.2)
```

```
##
## Call:
## glm(formula = admit ~ gre + gpa + factor(rank) + I(gre^2) + I(gpa^2) +
##     gpa:gre, family = binomial(link = logit), data = admissions)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5502  -0.8754  -0.6297   1.1187   2.1888
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -7.325e+00  9.065e+00  -0.808 0.419012
## gre            1.860e-02  1.184e-02   1.571 0.116136
## gpa           -1.777e-01  4.952e+00  -0.036 0.971371
## factor(rank)2 -7.130e-01  3.202e-01  -2.227 0.025958 *
```

```
## factor(rank)3 -1.341e+00  3.474e-01  -3.861 0.000113 ***
## factor(rank)4 -1.595e+00  4.221e-01  -3.780 0.000157 ***
## I(gre^2)       3.070e-06  8.216e-06   0.374 0.708624
## I(gpa^2)       6.699e-01  7.625e-01   0.878 0.379690
## gre:gpa       -5.888e-03  3.196e-03  -1.842 0.065475 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 499.98  on 399  degrees of freedom
## Residual deviance: 454.90  on 391  degrees of freedom
## AIC: 472.9
##
## Number of Fisher Scoring iterations: 4
```

```r
model.3.2$coefficients
```

```
##   (Intercept)           gre           gpa factor(rank)2 factor(rank)3
## -7.325485e+00  1.860245e-02 -1.777052e-01 -7.130421e-01 -1.341372e+00
## factor(rank)4      I(gre^2)      I(gpa^2)       gre:gpa
## -1.595493e+00  3.070427e-06  6.698514e-01 -5.887872e-03
```

The equation is: Y = -7.325485e+00(Intercept) + 1.860245e-02(gre) - 1.777052e-01(gpa) - 7.130421e-01(rank2) - 1.341372e+00(rank3) - 1.595493e+00(rank4) + 3.070427e-06(gre^2) + 6.698514e-01(gpa^2) - 5.887872e-03(gre:gpa).

Question 3.3: Test the hypothesis that GRE has no effect on admission using the likelihood ratio test

```r
library(car)

Anova(model.3.2, test = "LR")
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: admit
##              LR Chisq Df Pr(>Chisq)
## gre            0.3687  1    0.54373
## gpa            0.0238  1    0.87749
## factor(rank)  21.8244  3  7.095e-05 ***
## I(gre^2)       0.1383  1    0.70994
## I(gpa^2)       0.7620  1    0.38269
## gre:gpa        3.4119  1    0.06473 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
model.3.3 <- glm(formula = admit ~ gpa + factor(rank)  + I(gpa^2), family = binomial(link = log
anova(model.3.2, model.3.3, test = "Chisq")
```

```
## Analysis of Deviance Table
```

```
##
## Model 1: admit ~ gre + gpa + factor(rank) + I(gre^2) + I(gpa^2) + gpa:gre
## Model 2: admit ~ gpa + factor(rank) + I(gpa^2)
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1       391     454.90
## 2       394     462.74 -3  -7.8421  0.04939 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Based on these two models, we can see that GRE does have an effect on admission. Using the Anova test, the interaction of gre:gpa have an affect at the 10% level. Using the anova test to further understand the difference, we see signifiance at the 5% level.

Question 3.4: What is the estimated effect of college GPA on admission?

```r
exp(model.3.2$coefficients)
```

```
##   (Intercept)           gre           gpa factor(rank)2 factor(rank)3
##  0.0006585402  1.0187765503  0.8371892064  0.4901508247  0.2614866848
## factor(rank)4      I(gre^2)      I(gpa^2)       gre:gpa
##  0.2028086036  1.0000030704  1.9539469935  0.9941294278
```

```r
# Looking at the result based on gre = 700, rank = 2
effect <- function(gpa) {
  predict.data <- data.frame(gpa =gpa, gre = 700, rank = 2)
  linear.pred <- predict(object = model.3.2, newdata = predict.data, type = "link", se = TRUE)
  pi.hat <- exp(linear.pred$fit) / (1 + exp(linear.pred$fit))
  return (pi.hat)
}
effect(2.0)
```

```
##         1
## 0.6384562
```

```r
effect(2.3)
```

```
##         1
## 0.5356926
```

```r
effect(2.7)
```

```
##         1
## 0.4410333
```

```r
effect(3.0)
```

```
##         1
## 0.4058162
```

```r
effect(3.3)
```

```
##         1
## 0.4001035
```

```r
effect(3.7)
```

```
##         1
## 0.4380362
```

```r
effect(4.0)
```

```
##         1
## 0.5021141
```

```r
c <- .3

# Using log odds here
admission <- exp(c*model.3.2$coefficients['gpa'] + c * 6 * model.3.2$coefficients['I(gpa^2)'])

admission
```

```
##      gpa
## 3.165848
```

We can see a 0.3 increase in GPA increases the odds of admission by 3.16x

Question 3.5: Construct the confidence interval for the admission probability for the students with $GPA = 3.3$, $GRE = 720$, and $rank = 1$

```r
alpha = 0.05
predict.data <- data.frame(gpa = (3.3), gre = 720, rank = 1)
predict(object = model.3.2, newdata = predict.data, type = "response")
```

```
##         1
## 0.5935494
```

```r
linear.pred <- predict(object = model.3.2, newdata = predict.data, type = "link", se = TRUE)
linear.pred$fit
```

```
##        1
## 0.378658
```

```r
pi.hat <- exp(linear.pred$fit) / (1 + exp(linear.pred$fit))

CI.lin.pred <- linear.pred$fit + qnorm(p = c(alpha/2, 1-alpha/2)) * linear.pred$se

CI.pi <- exp(CI.lin.pred)/(1+exp(CI.lin.pred))

data.frame(predict.data, pi.hat, lower = CI.pi[1], upper= CI.pi[2])
```

```
##   gpa gre rank    pi.hat     lower     upper
## 1 3.3 720    1 0.5935494 0.4344916 0.7351409
```

The expected probability of admission is 0.5692897 with an interval of [0.4366982, 0.6926379]

4. Binary Logistic Regression (2 points)

Load the `Mroz` data set that comes with the *car* library (this data set is used in the week 2 live session file).

```
library(car)
head(Mroz)
```

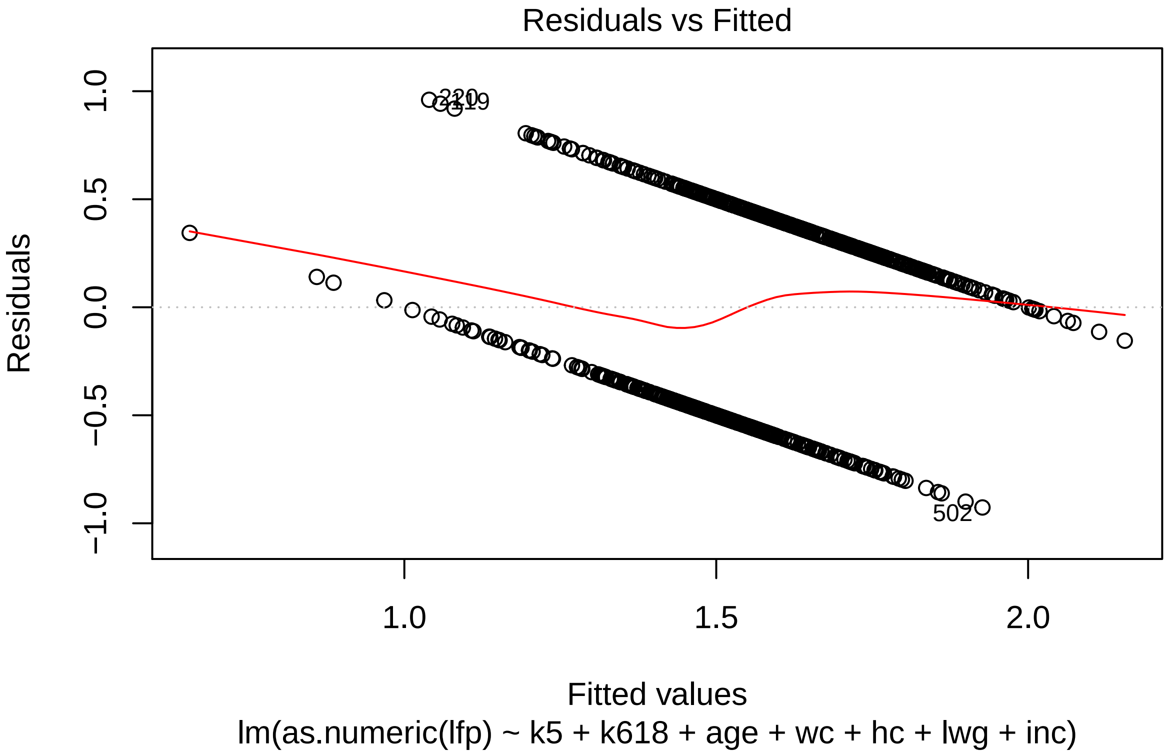
```
##   lfp k5 k618 age  wc hc       lwg    inc
## 1 yes  1    0  32  no no 1.2101647 10.910
## 2 yes  0    2  30  no no 0.3285041 19.500
## 3 yes  1    3  35  no no 1.5141279 12.040
## 4 yes  0    3  34  no no 0.0921151  6.800
## 5 yes  1    2  31 yes no 1.5242802 20.100
## 6 yes  0    0  54  no no 1.5564855  9.859
```

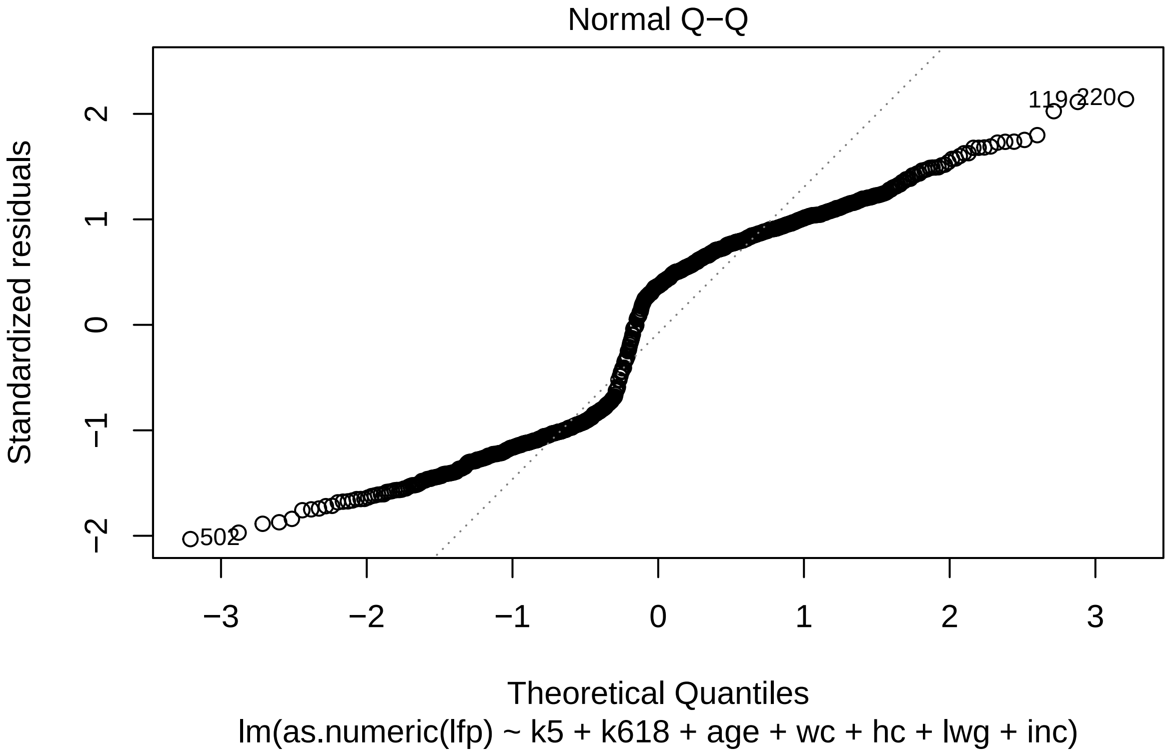
Question 4.1: Estimate a linear probability model using the same specification as in the binary logistic regression model estimated in the week 2 live session. Interpret the model results. Conduct model diagnostics. Test the CLM model assumptions.

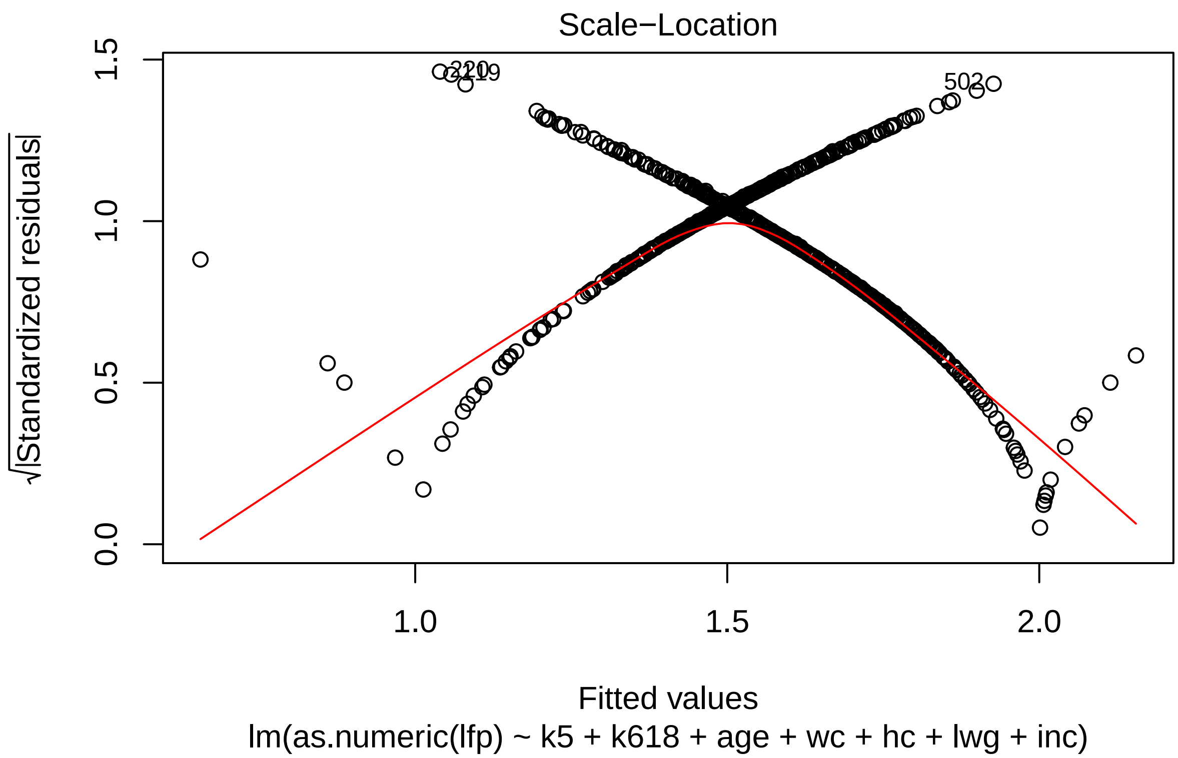
```
library(car)
# REMEMBER HERE YOU ARE CHANGING FROM FACTOR TO NUMBER
model.4.1 <- lm(formula = as.numeric(lfp) ~ k5 + k618 + age + wc + hc + lwg + inc, data = Mroz)
summary(model.4.1)$coefficients
```

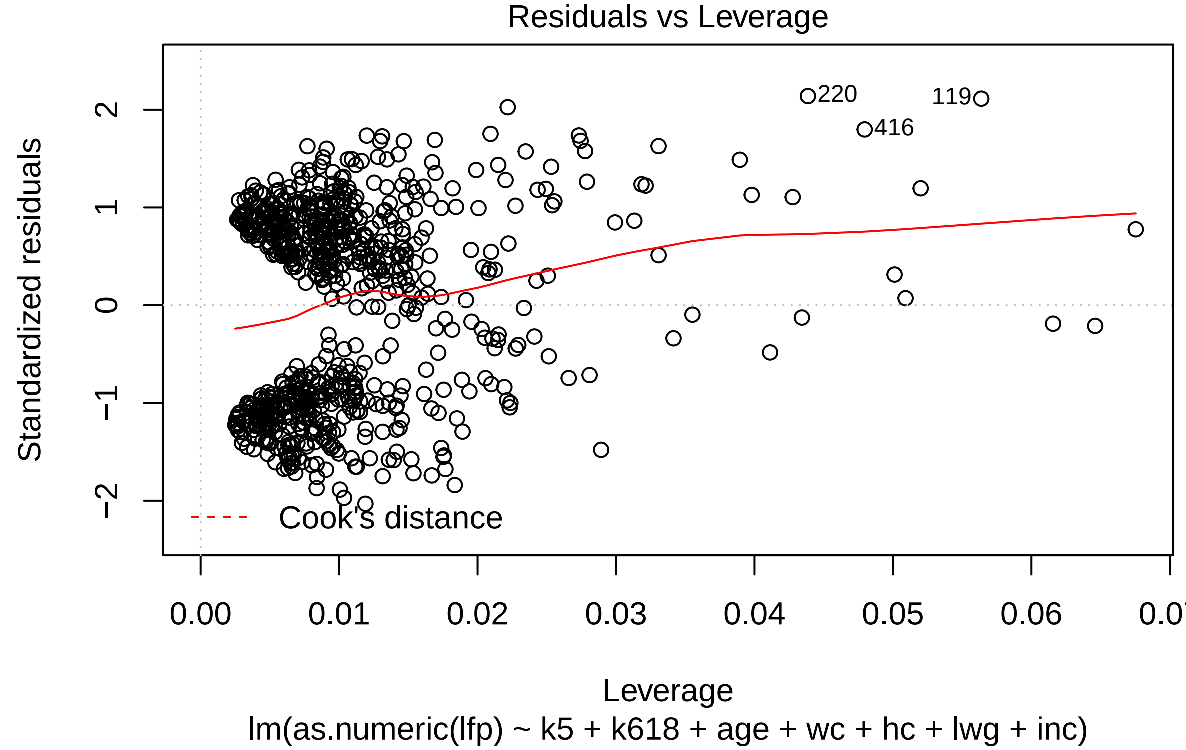
```
##                 Estimate  Std. Error    t value     Pr(>|t|)
## (Intercept)  2.143547836 0.127052665 16.8713331 2.488880e-54
## k5          -0.294835968 0.035902719 -8.2120790 9.577673e-16
## k618        -0.011215027 0.013962739 -0.8032111 4.221089e-01
## age         -0.012741098 0.002537729 -5.0206691 6.447882e-07
## wcyes        0.163679033 0.045828365  3.5715661 3.776538e-04
## hcyes        0.018951039 0.042532966  0.4455612 6.560437e-01
## lwg          0.122740218 0.030191492  4.0653909 5.305225e-05
## inc         -0.006760342 0.001570773 -4.3038299 1.902808e-05
```

```
# summary(model.4.1)
plot(model.4.1)
```


Residuals vs Fitted
Residuals
1.0
0.5
0.0
−0.5
−1.0
220
119
502
1.0
1.5
2.0
Fitted values
lm(as.numeric(lfp) ~ k5 + k618 + age + wc + hc + lwg + inc)


Normal Q−Q
Standardized residuals
2
1
0
−1
−2
119
220
502
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(as.numeric(lfp) ~ k5 + k618 + age + wc + hc + lwg + inc)






```
# Binomial Mass Function
```

The equation of the model is: 1.143547836(Intercept) - 0.294835968(k5) - 0.011215027(k618) - 0.012741098(age) + 0.163679033(wcyes) + 0.018951039(hcyes) + 0.122740218(lwg) - 0.006760342(inc). Starting with the residuals, we can see there is a clear pattern in them (based of the 0 residusal line). This breaks the

hetroskedastic assumption of linear regression. Next, the QQ plot should not be S shaped, and should be more of a straight line. For the scale v location, you want to see a horizational line with points that are equally spaced. This is clealy not the case for the hyperbolic results we seeFor the lev v residual plot, we can see there are a decent amount of influential cases which can be concerning. Thus, all 4 graphs show how linear regression is a terrible way to model the Mroz dataset.

Question 4.2: Estimate a binary logistic regression with `lfp`, which is a binary variable recoding the participation of the females in the sample, as the dependent variable. The set of explanatory variables includes `age`, `inc`, `wc`, `hc`, `lwg`, `totalKids`, and a quadratic term of `age`, called `age_squared`, where `totalKids` is the total number of children up to age 18 and is equal to the sum of `k5` and `k618`.

```
totalKids <- Mroz$k5 + Mroz$k618
age_squared <- I(Mroz$age^2)

model.4.2 <- glm(formula = lfp ~ age + inc + hc + wc + lwg + age_squared + totalKids, family =
summary(model.4.2)
```

```
##
## Call:
## glm(formula = lfp ~ age + inc + hc + wc + lwg + age_squared +
##     totalKids, family = binomial(link = logit), data = Mroz)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8342  -1.1669   0.6773   1.0079   2.0614
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.294073   2.281551  -2.320 0.020320 *
## age          0.318014   0.109463   2.905 0.003670 **
## inc         -0.034561   0.007922  -4.363 1.28e-05 ***
## hcyes        0.098260   0.198970   0.494 0.621417
## wcyes        0.666013   0.218074   3.054 0.002258 **
## lwg          0.549976   0.145506   3.780 0.000157 ***
## age_squared -0.004114   0.001272  -3.233 0.001224 **
## totalKids   -0.222490   0.063849  -3.485 0.000493 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1029.75  on 752  degrees of freedom
## Residual deviance:  952.02  on 745  degrees of freedom
## AIC: 968.02
##
## Number of Fisher Scoring iterations: 4
```

```r
model.4.2$coefficients
```

```
##   (Intercept)           age           inc         hcyes         wcyes 
## -5.294072819   0.318013831 -0.034561469  0.098259837  0.666013385 
##           lwg   age_squared     totalKids 
##  0.549976370 -0.004113978 -0.222489591
```

log likelihood = -5.150511297 + 0.311895142(age) - 0.033434758(inc) +0.713378272(wcyes) + 0.550747255(lwg) - 0.004051356(age_squared) - 0.221626269(totalKids)

Question 4.3: Is the age effect statistically significant?

```r
Anova(model.4.2, test = "LR")
```

```
## Analysis of Deviance Table (Type II tests)
## 
## Response: lfp
##             LR Chisq Df Pr(>Chisq)    
## age           8.6144  1  0.0033351 ** 
## inc          21.0740  1  4.419e-06 ***
## hc            0.2439  1  0.6213914    
## wc            9.5398  1  0.0020107 ** 
## lwg          15.0213  1  0.0001063 ***
## age_squared  10.7487  1  0.0010435 ** 
## totalKids    12.4267  1  0.0004232 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The age effect, as well as age squared, is significant at the .001% level.

Question 4.4: What is the effect of a decrease in age by 5 years on the odds of labor force participation for a female who was 45 years of age.

```r
c = -5
Age = 45

OR.change <- exp(c*(coef(model.4.2)[['age']] + coef(model.4.2)[['age_squared']]*(2*Age + c)))
OR.change
```

```
## [1] 1.171602
```

The odds of labor participation go up by 1.17x

Question 4.5: Estimate the profile likelihood confidence interval of the probability of labor force participation for females who were 40 years old, had income equal to 20, did not attend college, had log wage equal to 1, and did not have children.

```r
alpha = 0.05

model.4.5 <- glm(formula = lfp ~ age + inc + wc + lwg + age_squared + totalKids, family = binor
predict.data <- data.frame(age = 40, inc = 20, wc = "no", lwg = 1, age_squared = 1600, totalKic
predict(object = model.4.5, newdata = predict.data, type =
"response")
```

```
##        1
## 0.673746
```

```
linear.pred <- predict(object = model.4.5, newdata = predict.data, type = "link", se = TRUE)
pi.hat <- exp(linear.pred$fit) / (1 + exp(linear.pred$fit))

CI.lin.pred <- linear.pred$fit + qnorm(p = c(alpha/2, 1-alpha/2)) * linear.pred$se

CI.pi <- exp(CI.lin.pred)/(1+exp(CI.lin.pred))

data.frame(predict.data, pi.hat, lower = CI.pi[1], upper= CI.pi[2])
```

```
##   age inc wc lwg age_squared totalKids   pi.hat     lower     upper
## 1  40  20 no   1        1600         0 0.673746 0.5942122 0.7443968
```

The p hat is 0.673746, with a CI of [0.5942122, 0.7443968]

5: Maximum Likelihood (2 points)

Question 18 a and b of Chapter 3 (page 192,193)

For the wheat kernel data (*wheat.csv*), consider a model to estimate the kernel condition using the density explanatory variable as a linear term.

```
library(nnet)

wheat <- read.csv('wheat.csv')
head(wheat)
```

```
##   class  density hardness    size  weight moisture    type
## 1   hrw 1.349253 60.32952 2.30274 24.6480 12.01538 Healthy
## 2   hrw 1.287440 56.08972 2.72573 33.2985 12.17396 Healthy
## 3   hrw 1.233985 43.98743 2.51246 31.7580 11.87949 Healthy
## 4   hrw 1.336534 53.81704 2.27164 32.7060 12.11407 Healthy
## 5   hrw 1.259040 44.39327 2.35478 26.0700 12.06487 Healthy
## 6   hrw 1.300258 48.12066 2.49132 33.2985 12.18577 Healthy
```

```
levels(wheat$type)
```

```
## [1] "Healthy" "Scab"    "Sprout"
```

```
wheat.model <- multinom(formula = type ~ density, data = wheat)
```

```
## # weights:  9 (4 variable)
## initial  value 302.118379
## iter  10 value 229.769334
## iter  20 value 229.712304
## final  value 229.712290
## converged
```

We can see for log (scab/healthy), the equation is 29.37827 - 24.56215(Density). For log (Sprout/healthy), it is 19.12165 - 15.47633. When I say scab, I mean the predicted prob of success of scab.

Question 5.1 Write an R function that computes the log-likelihood function for the multinomial regression model. Evaluate the function at the parameter estimates produced by multinom(), and verify that your computed value is the same as that produced by logLik() (use the object saved from multinom() within this function).

```
# Calculate log_likelihood by hand
logL <- function(beta, x, Y) {

  beta.1.3 <- exp(beta[1] + beta[3] * x)
  beta.2.4 <- exp(beta[2] + beta[4] * x)

  pi.h <- 1/(1 + beta.1.3 + beta.2.4)
  Y.h <- ifelse(Y == "Healthy", 1, 0)

  pi.sc <- beta.1.3/(1 + beta.1.3 + beta.2.4)
```

```r
  Y.sc <- ifelse(Y == "Scab", 1, 0)

  pi.sp <- beta.2.4/(1 + beta.2.4 + beta.1.3)
  Y.sp <- ifelse(Y == "Sprout", 1, 0)


  sum(Y.h *log(pi.h) + Y.sc*log(pi.sc) + Y.sp * log(pi.sp))

}

logL(beta = summary(wheat.model)$coefficients, x = wheat$density, Y = wheat$type)
```

```
## [1] -229.7123
```

```r
logLik(wheat.model)
```

```
## 'log Lik.' -229.7123 (df=4)
```

Confirmed, the two values are the same (-229.7123).

Question 5.2 Maximize the log-likelihood function using optim() to obtain the MLEs and the estimated covariance matrix. Compare your answers to what is obtained by multinom(). Note that to obtain starting values for optim(), one approach is to estimate separate logistic regression models for $log\left(\frac{\pi_2}{\pi_1}\right)$ and $log\left(\frac{\pi_3}{\pi_1}\right)$. These models are estimated only for those observations that have the corresponding responses (e.g., a $Y = 1$ or $Y = 2$ for $log\left(\frac{\pi_2}{\pi_1}\right)$).

```r
# Using page 72 as reference from the book
mod.fit.optim <- optim(summary(wheat.model)$coefficients, logL, x = wheat$density, Y = wheat$ty
mod.fit.optim$value
```

```
## [1] -229.7123
```

```r
mod.fit.optim$hessian
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] -39.36826  27.48339 -45.58971  31.12966
## [2,]  27.48339 -57.76752  31.12966 -68.86774
## [3,] -45.58971  31.12966 -53.14665  35.51096
## [4,]  31.12966 -68.86774  35.51096 -82.62965
```

The optimized log likelihood and Hessian matrix can be seen above with the use of the optim function.