HW1 Part 2

August 4, 2021

1 HW1 Part 2

1.1 Instructions:

For the rest of the semester, we will be using Vocareum to work on and submit your homework assignments. Vocareum is a cloud platform for programming classes. It provides an infrastructure that allows us to move the educational aspects like assignments, exams, quizzes, etc, to the cloud. The merit of this platform is that all of you will be working in the same coding environment. This way we can eliminate many issues we might encounter when working on an individual basis, such as those with library installations and RStudio malfunctions. Some you you might be new to this platform, here we provided a few things to keep in mind to get you started, please try to read through them.

Things to keep in mind:

Even though we are moving from your local envrionment to the cloud, **our expectations from your homework will stay the same**. Same goes for the rubrics.

Vocareum has its own cloud based file system, the data files you will be using for the assignments will be stored in the cloud with path **"../resource/asnlib/publicdata/FILENAME.csv"**. You will be able to import them with the same method as you do in RStudio, simply substitute the path name to the one specified in the instructions. You won't be able to modify these data files.

You will be able to find the data files on Canvas/EdX if you would like to explore them offline.

For coding questions, you will be graded on the R code as well as the output in your submission.

For interpretations or short response questions, please type the answers in the notebook's markdown cells. To change a code cell to a markdown cell, click on the cell, and in the dropdown menu above, switch the type of the cell block from "code" to "markdown". **Adding print statements to code cells for short response/interpretation questions is also fine, as long as we can clearly see the output of your response**.

You don't need to, but if you would like to learn more about how to format your markdown cells, visit the following site: https://www.earthdatascience.org/courses/intro-to-earth-data-science/file-formats/use-text-files/format-text-with-markdown-jupyter-notebook/. Jupyter notebook also support LaTeX.

Feel free to add as many additional cells as you need. But please keep your solution to a question directly under that question to aviod confusions.

You may delete the #SOLUTION BEGINS/ENDS HERE comments from the cell blocks, they are just pointers that indicates where to put you solutions.

When you have finished the assignment, remember to rerun your notebook to check if it runs correctly. You can do so by going to **Kernel-> Restart & Run All**. You may lose points if your solutions does not run successfully.

Click the "Submit" button on the top right corner to turn in your assignment. Your assignment will enter the next phase for peer review.

**You are allowed a total of 2 submissions for this assignment. So make sure that you submit your responses carefully. You will be able to come back and resubmit your assignment as long as it is before the start of the peer review period.

Please remember to finish the peer reviews after you have submitted your assignment. You are responsible for grading the work of three of your peers thoroughly, and in addherence to the rubrics. And you will be held accountable for peer grading. **There will be a 30% penalty to your grade if you fail to complete one or more peer reviews in proper fashion.**

This is the first time we implement this homework system in the MGT6203, feel free to address your questions, concerns, and provide any feedback on Piazza. We will continuously try to improve going forward.

Good Luck!

2 About Package Installation:

Most of the packages (if not all) that you will need to complete this assignment are already installed in this environment. An easy way to check this is to run the command: library(PackageName). If this command runs successfully then the package was already installed and has been successfully attached to the code. If the command gave an error saying the Package was not found then follow the steps below to successfully install the package and attach it to the code:

Use *installed.packages()* command to return a table of the packages that are preinstalled in the environment.

To attach a preinstalled library in Vocareum, simply use *library(PackageName)*

To install a package that does not come with the provided environment, please use the following syntax:

install.packages("PackageName", lib="../work/")

To attach a library you just installed, use the following syntax:

library(PackageName, lib.loc="../work/")

Make sure the file location is the same as the above code snippets *("../work/")*

2.1 Q1. Use the *"airbnb_data.csv"* provided and answer the following questions on Linear Regression:

Instruction: The file "airbnb_data.csv" can be accessed at the path: ("../resource/asnlib/publicdata/airbnb_data.csv")

a) Remove 'id'columns ('room_id', 'survey_id', 'host_id') and 'city' from your dataset, and fit a multiple linear regression model using price as the response variable and all others as predictor variables. (Note: Do not fit a model using id columns and city as predictors). Which variables are statistically significant at a 95% confidence interval. (4 point)

```
In [20]: # SOLUTION BEGINS HERE
         #Read RAW data
         data_raw = read.csv("../resource/asnlib/publicdata/airbnb_data.csv", header = TRUE)

         #Select subset of columns excluding id columns & city
```

```
data <- data_raw[,c('room_type', 'reviews', 'overall_satisfaction','accommodates','bed
head(data)
# Run the model
model = lm(price ~ ., data=data)
summary(model)

# SOLUTION ENDS HERE
```

A data.frame: 6 Œ 6

room_type <fct>	reviews <int>	overall_satisfaction <dbl>	accommodates <int>	bedrooms <int>	price <int>
Shared room	0	0.0	4	1	67
Shared room	32	5.0	4	1	76
Shared room	4	4.5	2	1	45
Shared room	24	4.5	6	1	26
Shared room	152	4.5	6	1	26
Shared room	20	4.5	4	1	26

```
Call:
lm(formula = price ~ ., data = data)

Residuals:
   Min     1Q Median     3Q    Max
-367.8  -49.2    3.2   38.6 4032.7

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)          -23.36172   21.88618  -1.067  0.28609
room_typePrivate room  -0.93115   13.21827  -0.070  0.94386
room_typeShared room  -76.66780   59.90939  -1.280  0.20099
reviews                 0.01090    0.09982   0.109  0.91310
overall_satisfaction  -10.48160    3.47320  -3.018  0.00262 **
accommodates           23.00721    5.23952   4.391 1.27e-05 ***
bedrooms               85.64533   11.45983   7.474 1.95e-13 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 167.1 on 847 degrees of freedom
Multiple R-squared:  0.3228,Adjusted R-squared:  0.318
F-statistic:  67.3 on 6 and 847 DF,  p-value: < 2.2e-16
```

Based on the model output above, the most significant independent variables are [overall_satisfaction, accommodates,bedrooms]. Since these variables have a Pr(>|t|) or less than 0.05 which means there is a less than 5% chance that the T value can be equal to greater than the current value

In [11]:
```
# Listing all the unique room type and aggregating rows counts
aggregate(data$room_type, by = list(data$room_type), FUN = length)
```

A data.frame: 3 Œ 2

Group.1	x
<fct>	<int>
Entire home/apt	512
Private room	334
Shared room	8

b) Interpret the coefficients for predictors: 'room_type', 'bedrooms'. (4 points)

When running lm() on th above dataset, R automatically converts room_type which is of type Factor to dummy variables

room_typePrivate room, Coefficient: -0.93115 room_typeShared room, Coefficient: -76.66780

The Base case here is Entire home/apt. If all other variables are kept the same with a Private room price drops by -0.93115 over Entire home/apt Shared room price drops by -76.66780 over Entire home/apt

Bedrooms is an integer column and one of the significant factors in estimating the price. Each Bedroom adds 85.64533 to the property price

c) Predict the price (nearest dollar) for a listing with the following factors: 'bedrooms' = 1, 'accommodates' = 2, 'reviews' = 70, 'overall_satisfaction' = 4, and 'room_type' = 'Private room'. (4 points)

bedrooms	accomodates	reviews	overall_satisfaction	room_type
1	2	70	4	Private room

In [22]:
```
# SOLUTION BEGINS HERE
# Predict Price for the given dataset

test<-data.frame(bedrooms = 1,accommodates = 2, reviews = 70, overall_satisfaction= 4
predict(model, test)
# SOLUTION ENDS HERE
```

66.20348

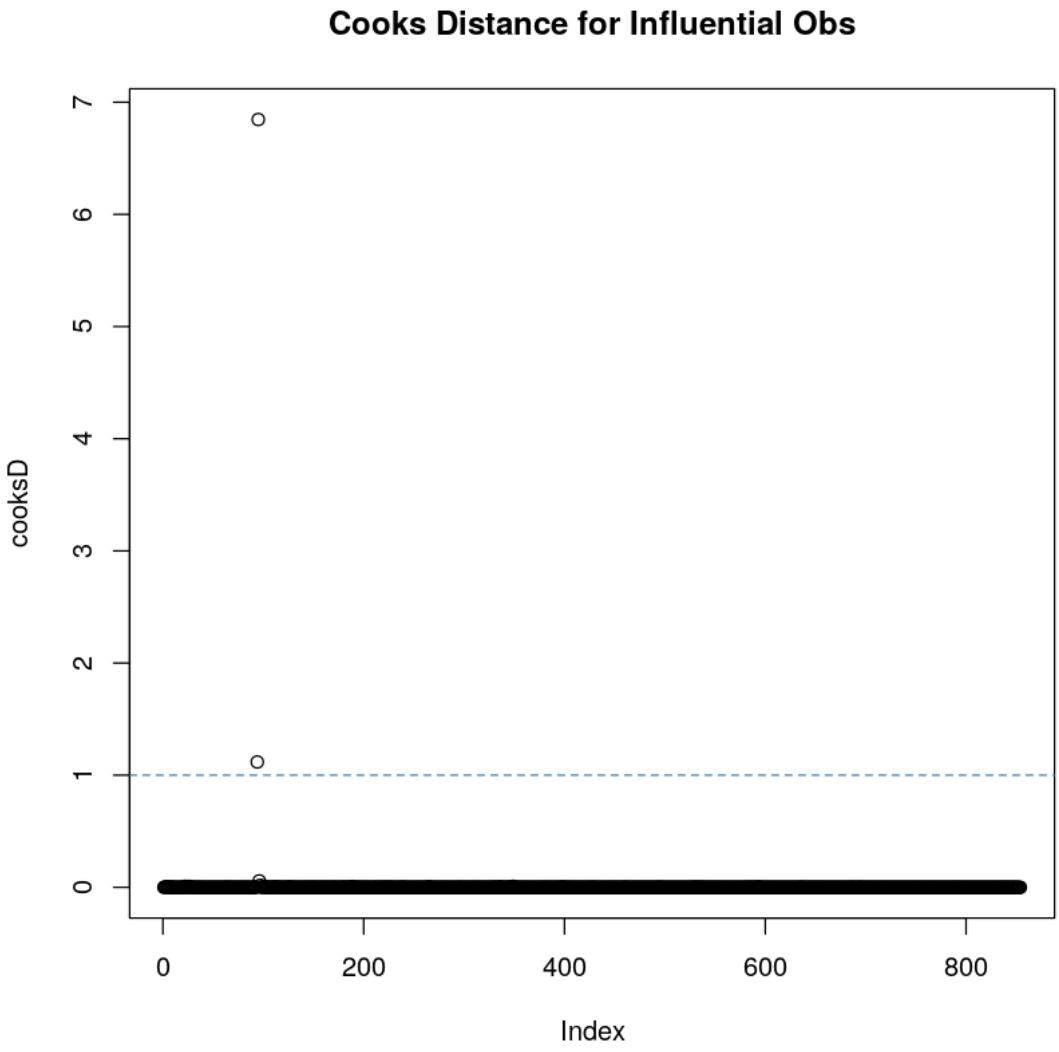
1: 66.2031622509052

d) Identify outliers using Cook's distance approach. Remove points having Cook's distance > 1. Rerun the model after the removal of these points and print the summary. (4 points)

In [13]:
```
# SOLUTION BEGINS HERE
# Estimate cook's distance for all data points and plot the graph
cooksD <- cooks.distance(model)
n <- nrow(data)
plot(cooksD, main = "Cooks Distance for Influential Obs")
abline(h = 1, lty = 2, col = "steelblue") # add cutoff line

# Identify all points where distance > 1
influential_obs <- as.numeric(names(cooksD)[(cooksD > 1)])
```

```r
#Remove influential points
data_new <- data[-influential_obs, ]

#Run the Model with influential points removed
model2 = lm(price ~ ., data=data_new)
summary(model2)
# SOLUTION ENDS HERE
```

```
Call:
lm(formula = price ~ ., data = data_new)

Residuals:
    Min      1Q  Median      3Q     Max
-190.95  -32.43   -7.09   20.35  876.26

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)             75.01310    9.09152   8.251 6.01e-16 ***
room_typePrivate room -32.28201    5.38034  -6.000 2.92e-09 ***
room_typeShared room  -91.69951   24.28958  -3.775 0.000171 ***
reviews                -0.05915    0.04047  -1.462 0.144202
overall_satisfaction   -6.78957    1.41118  -4.811 1.78e-06 ***
accommodates           11.90698    2.14267   5.557 3.68e-08 ***
bedrooms               35.93177    4.87968   7.364 4.25e-13 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 67.73 on 845 degrees of freedom
Multiple R-squared:  0.4249,Adjusted R-squared:  0.4208
F-statistic:   104 on 6 and 845 DF,  p-value: < 2.2e-16
```



2.2 Q2. Use the *"direct_marketing.csv"* provided and answer the following questions on Linear Regression:

Create indicator variables for the 'History' column. Considering the base case as None (i.e., create Low, Medium and High variables with 1 denoting the positive case and 0 the negative) and few additional variables LowSalary, MediumSalary and HighSalary based on the customer history type i.e., MediumSalary = Medium*Salary etc.

Instruction: The dataset "direct_marketing.csv" can be accessed at the path: ("../resource/asnlib/publicdata/direct_marketing.csv")

a) Fit a multiple linear regression model using 'AmountSpent' as the response variable and the indicator variables along with their salary variables as the predictors: (4 points)

$$AmountSpent = \beta_0 + \beta_1 Salary + \beta_2 Low + \beta_3 Medium + \beta_4 High + \beta_5 LowSalary + \beta_6 MediumSalary + \beta_7 HighSalar$$

In [14]:
```
# SOLUTION BEGINS HERE
data = read.csv("../resource/asnlib/publicdata/direct_marketing.csv", header = TRUE)

data$Low <- 0
data$Medium <- 0
data$High <- 0
data$LowSalary <- 0
data$MediumSalary <- 0
data$HighSalary <- 0

data$Low[data$History == "Low"] <- 1
data$Medium[data$History == "Medium"] <- 1
data$High[data$History == "High"] <- 1

data$LowSalary <- data$Low*data$Salary
data$MediumSalary <- data$Medium*data$Salary
data$HighSalary <- data$High*data$Salary

head(data)

model = lm(AmountSpent ~ Salary+Low+Medium+High+LowSalary+MediumSalary+HighSalary, dat
summary(model)
# SOLUTION ENDS HERE
```

A data.frame: 6 Œ 16

Age <fct>	Gender <fct>	OwnHome <fct>	Married <fct>	Location <fct>	Salary <int>	Children <int>	History <fct>	C
Old	Female	Own	Single	Far	47500	0	High	6
Middle	Male	Rent	Single	Close	63600	0	High	6
Young	Female	Rent	Single	Close	13500	0	Low	1
Middle	Male	Own	Married	Close	85600	1	High	1
Middle	Female	Own	Single	Close	68400	0	High	1
Young	Male	Own	Married	Close	30400	0	Low	6

```
Call:
lm(formula = AmountSpent ~ Salary + Low + Medium + High + LowSalary +
    MediumSalary + HighSalary, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-214.33  -25.47   -6.46   20.64  352.50

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.9622199  6.3880253   0.307 0.758777
```

```
Salary        0.0023641  0.0001071  22.083  < 2e-16 ***
Low          25.4466733  8.9203292   2.853 0.004426 **
Medium       79.2984388 12.8982169   6.148 1.14e-09 ***
High         72.6735221 15.2270169   4.773 2.09e-06 ***
LowSalary    -0.0021069  0.0001890 -11.150  < 2e-16 ***
MediumSalary -0.0021153  0.0002182  -9.693  < 2e-16 ***
HighSalary   -0.0006408  0.0001926  -3.328 0.000908 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 55.79 on 992 degrees of freedom
Multiple R-squared:  0.6654,Adjusted R-squared:  0.6631
F-statistic: 281.9 on 7 and 992 DF,  p-value: < 2.2e-16
```

b) What is the amount spent by a customer for each historic type (None, Low, Medium, and High) provided their salary is $10,000 based on the model constructed in part a? (4 points)

```
In [15]: # SOLUTION BEGINS HERE
         #History = None
         test<-data.frame(Salary = 10000,Low = 0, Medium = 0,High = 0,LowSalary = 0, MediumSala
         predict(model, test)

         #History = Low
         test<-data.frame(Salary = 10000,Low = 1, Medium = 0,High = 0,LowSalary = 10000, Mediu
         predict(model, test)

         #History = Medium
         test<-data.frame(Salary = 10000,Low = 0, Medium = 1,High = 0,LowSalary = 0, MediumSala
         predict(model, test)

         #History = High
         test<-data.frame(Salary = 10000,Low = 0, Medium = 0,High = 1,LowSalary = 0, MediumSala
         predict(model, test)
         # SOLUTION ENDS HERE
```

1: 25.603473247411
1: 29.9815735873686
1: 83.749086769995
1: 91.8687402713128

Use the *"airbnb_data.csv"* provided and answer the following questions (part c and part d) on Linear Regression. DO NOT remove outliers from the dataset: Perform Log transformation for the variables price and overall_satisfaction, make necessary transformations suggested in the class.

c) Fit all four models i.e., linear-linear, linear-log, log-linear and log-log regression models using price as the response variable and overall_satisfaction as the predictor. (Note: Because

overall_satisfaction contains '0' values, you will need to use log(x+1) transformations instead of log(x) transformations) (6 points)

In [16]:
```
# SOLUTION BEGINS HERE
data = read.csv("../resource/asnlib/publicdata/airbnb_data.csv", header = TRUE)
data$overall_satisfaction_log <- log(data$overall_satisfaction+1)
data$price_log <- log(data$price)
print('##################Linear-Linear model below ######################################')
model_linear_linear <- lm(price ~ overall_satisfaction, data=data)
summary(model_linear_linear)

print('##################Linear-Log model below ########################################')

model_linear_log <- lm(price ~ overall_satisfaction_log, data=data)
summary(model_linear_log)

print('##################Log-Linear model below ########################################')

model_log_linear <- lm(price_log ~ overall_satisfaction, data=data)
summary(model_log_linear)

print('##################Log-Log model below ###########################################')

model_log_log <- lm(price_log ~ overall_satisfaction_log, data=data)
summary(model_log_log)
print('################################################################################')
# SOLUTION ENDS HERE
```

```
[1] "##################Linear-Linear model below ###########################################"



Call:
lm(formula = price ~ overall_satisfaction, data = data)

Residuals:
   Min     1Q Median     3Q    Max
-167.0  -51.3  -24.2   16.8 4805.0

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           194.967     17.698  11.016  < 2e-16 ***
overall_satisfaction  -16.353      3.903  -4.189 3.09e-05 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 200.4 on 852 degrees of freedom
Multiple R-squared:  0.02018,Adjusted R-squared:  0.01903
```

```
F-statistic: 17.55 on 1 and 852 DF,  p-value: 3.088e-05


[1] "##################Linear-Log model below ##############################################"


Call:
lm(formula = price ~ overall_satisfaction_log, data = data)

Residuals:
   Min     1Q Median     3Q    Max
-168.5  -50.7  -24.7   16.3 4803.5

Coefficients:
                         Estimate Std. Error t value Pr(>|t|)
(Intercept)                196.46      17.76  11.062  < 2e-16 ***
overall_satisfaction_log   -46.20      10.84  -4.263 2.24e-05 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 200.4 on 852 degrees of freedom
Multiple R-squared:  0.02089,Adjusted R-squared:  0.01974
F-statistic: 18.18 on 1 and 852 DF,  p-value: 2.239e-05


[1] "##################Log-Linear model below ##############################################"


Call:
lm(formula = price_log ~ overall_satisfaction, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-1.6234 -0.3525 -0.0432  0.3302  3.7220

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)           4.79515    0.05083  94.339  < 2e-16 ***
overall_satisfaction -0.04401    0.01121  -3.926 9.33e-05 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.5757 on 852 degrees of freedom
Multiple R-squared:  0.01777,Adjusted R-squared:  0.01662
F-statistic: 15.41 on 1 and 852 DF,  p-value: 9.331e-05
```

```
[1] "#################Log-Log model below ###################################################"



Call:
lm(formula = price_log ~ overall_satisfaction_log, data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-1.6030 -0.3551 -0.0327  0.3298  3.7132

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)                4.80396    0.05098  94.228  < 2e-16 ***
overall_satisfaction_log -0.12750    0.03111  -4.099 4.55e-05 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.5752 on 852 degrees of freedom
Multiple R-squared:  0.01934,Adjusted R-squared:  0.01819
F-statistic:  16.8 on 1 and 852 DF,  p-value: 4.547e-05



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

d) Which of the four models has the best R^2? Do you have any comments on the choice of the independent variables? (2 points)

```
In [17]: # SOLUTION BEGINS HERE
         summary(model_linear_linear)$r.squared
         summary(model_linear_log)$r.squared
         summary(model_log_linear)$r.squared
         summary(model_log_log)$r.squared
         # SOLUTION ENDS HERE
```

0.0201842827776218
0.020887386741353
0.0177702713103193
0.019338613785989

Based on the above output you can see that linear-log has the best R^2. we can see that log transformation of the X variable "overall_satisfaction" gives a slightly better R^2

2.3 Q3. The attached "*titanic_data.csv*" file was obtained from the following source: http://math.ucdenver.edu/RTutorial/

It has been cleaned to remove all rows which contain missing values. We will perform a logistic regression on this cleaned dataset.

The dataset contains the following columns:

Column Name	Description	Data Type
Name	Passenger Name	factor
PClass	Passenger Class (1st, 2nd, 3rd)	factor
Age	Passenger Age	number
Sex	Passenger Sex - female, male	factor
Survived	1 if passenger survived, 0 if not	number

After converting the survived variable to be a factor with two levels, 0 and 1, perform a logistic regression on the dataset using 'survived' as the response and 'Sex' as the explanatory variable.

Instruction: The file "titanic_data.csv" can be accessed at the path: ("../resource/asnlib/publicdata/titanic_data.csv")

a. Display the model summary. (2 points)

```
In [18]: # SOLUTION BEGINS HERE
         data = read.csv("../resource/asnlib/publicdata/titanic_data.csv", header = TRUE)
         data['Survived'] <- as.factor(data['Survived'])
         head(data)

         model <- glm(Survived ~ Sex, data = data, family=binomial(link="logit"))
         summary(model)

         # SOLUTION ENDS HERE
```

A data.frame: 6 Œ 5

	Name <fct>	PClass <fct>	Age <dbl>	Sex <fct>	Survived <fct>
	Allen, Miss Elisabeth Walton	1st	29.00	female	1
	Allison, Miss Helen Loraine	1st	2.00	female	0
	Allison, Mr Hudson Joshua Creighton	1st	30.00	male	0
	Allison, Mrs Hudson JC (Bessie Waldo Daniels)	1st	25.00	female	0
	Allison, Master Hudson Trevor	1st	0.92	male	1
	Anderson, Mr Harry	1st	47.00	male	1

```
Call:
glm(formula = Survived ~ Sex, family = binomial(link = "logit"),
    data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.6735  -0.6776  -0.6776   0.7524   1.7800
```

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.1172     0.1367   8.171 3.05e-16 ***
Sexmale      -2.4718     0.1783 -13.861  < 2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1025.57  on 755  degrees of freedom
Residual deviance:  796.64  on 754  degrees of freedom
AIC: 800.64

Number of Fisher Scoring iterations: 4
```

b. What does the value of the intercept coefficient represent in this model? (2 points)

As you can see from the output above, R converts variable Sex to a dummy variable SexMale. So the intercept coefficient represents log odds of survival log(p/1-p) for female passengers.

c. Determine the probability of survival for females. (2 points)

log(p/1-p) = b_0 log(p/1-p) = 1.1172 p/1-p = $e^{1.1172}$ p=(1-p)*3.0563 p=3.0563/4.0563 p=0.75
The probability of survival for a female passenger is 0.75

d. Determine the probability of survival for males. (2 points)

log(p/1-p) = b_0+b_1X log(p/1-p) = 1.1172 - 2.4718 p/1-p = $e^{-1.3546}$ p=(1-p)*0.258 p=0.258/1.258 p=0.20
The probability of survival for a male passenger is 0.20

In [ ]: