HW1 Part 2

July 31, 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 will be able to come back and resubmit your assignment as long as it is before the start of the peer review period.

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 points)

```
In [1]: # SOLUTION BEGINS HERE
        data = read.csv("../resource/asnlib/publicdata/airbnb_data.csv")
        head(data)

        drops = c("room_id", "survey_id", "host_id", "city")
        data = data[,!names(data) %in% drops]
        head(data)
        str(data)

        model = lm(price~., data=data)
        summary(model)
        # SOLUTION ENDS HERE
```

A data.frame: 6 Œ 10

room_id <int>	survey_id <int>	host_id <int>	room_type <fct>	city <fct>	reviews <int>	overall_satis <dbl>
15771735	1498	101992409	Shared room	Asheville	0	0.0
18284194	1498	126414164	Shared room	Asheville	32	5.0
18091012	1498	122380971	Shared room	Asheville	4	4.5
12286328	1498	746673	Shared room	Asheville	24	4.5
156926	1498	746673	Shared room	Asheville	152	4.5
12989718	1498	746673	Shared room	Asheville	20	4.5

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

```
'data.frame':        854 obs. of  6 variables:
 $ room_type           : Factor w/ 3 levels "Entire home/apt",..: 3 3 3 3 3 3 3 3 1 1 ...
 $ reviews             : int  0 32 4 24 152 20 52 14 3 30 ...
 $ overall_satisfaction: num  0 5 4.5 4.5 4.5 4.5 4.5 4.5 5 5 ...
 $ accommodates        : int  4 4 2 6 6 4 5 2 6 5 ...
 $ bedrooms            : int  1 1 1 1 1 1 1 1 3 2 ...
 $ price               : int  67 76 45 26 26 26 26 20 46 150 ...
```

```
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
```

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

**price=-23.36-0.93(Private_room)-76.67(Shared_room)+0.01(reviews)-10.48(overall_satisfaction)+23(accomodates)+85.64(bedrooms)

**Room_type: With "Entire house" as a reference, a "Private room" will cost -0.93 units less, and a "Shared room" will cost -76.67 units less, keeping all else constant.

**Bedrooms: A unit increase in number of bedrooms will lead to 85.64 units increase in price, keeping all else constant.

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 [3]:
```
# SOLUTION BEGINS HERE
data_pred = data.frame(bedrooms = c(1),
                       accommodates = c(2),
                       reviews = c(70),
                       overall_satisfaction = c(4),
                       room_type = c("Private room"))

data_pred$bedrooms = as.integer(data_pred$bedrooms)
data_pred$accommodates = as.integer(data_pred$accommodates)
data_pred$reviews = as.integer(data_pred$reviews)
data_pred$overall_satisfaction = as.double(data_pred$overall_satisfaction)
str(data_pred)
head(data_pred)

predict(model, data_pred)
# SOLUTION ENDS HERE
```

```
'data.frame':        1 obs. of  5 variables:
 $ bedrooms            : int 1
 $ accommodates        : int 2
 $ reviews             : int 70
 $ overall_satisfaction: num 4
 $ room_type           : Factor w/ 1 level "Private room": 1
```

A data.frame: 1 Œ 5

bedrooms <int>	accommodates <int>	reviews <int>	overall_satisfaction <dbl>	room_type <fct>
1	2	70	4	Private room

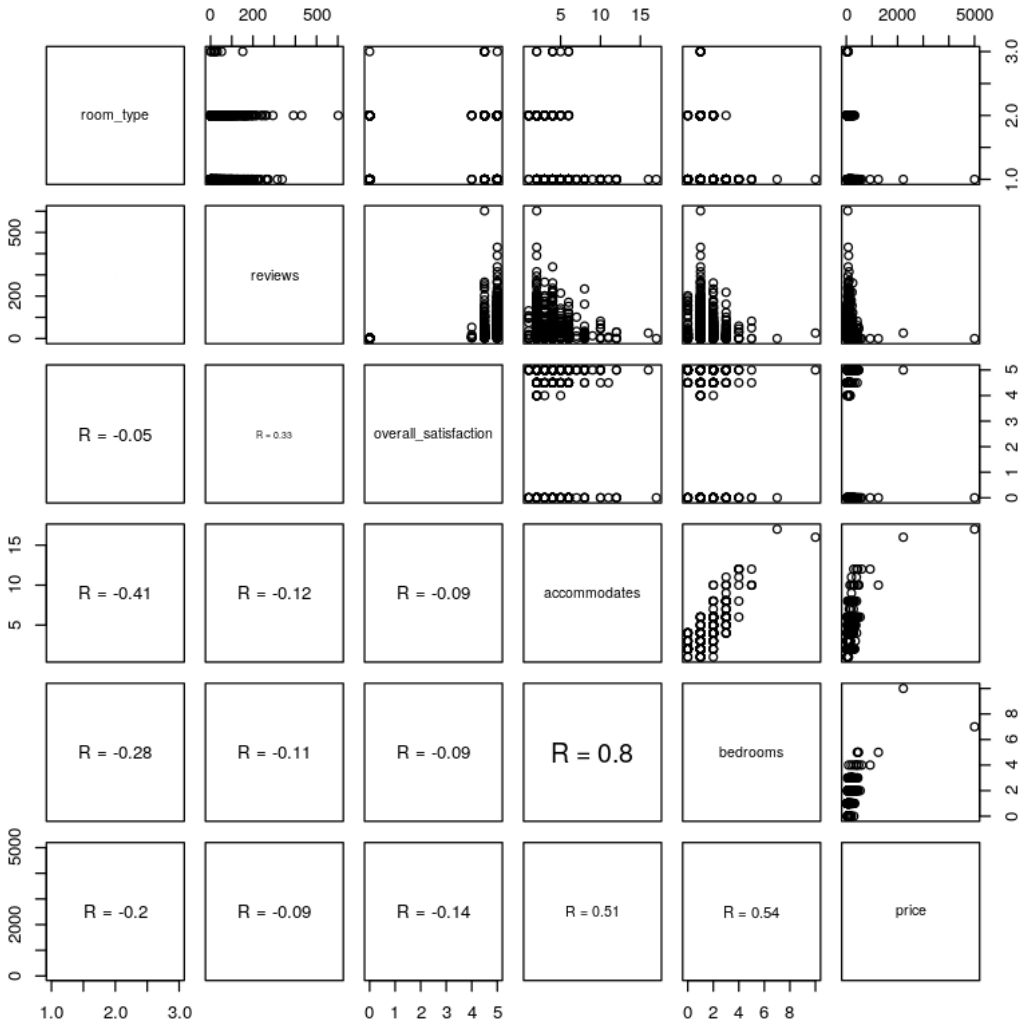
1: 66.2031622509052

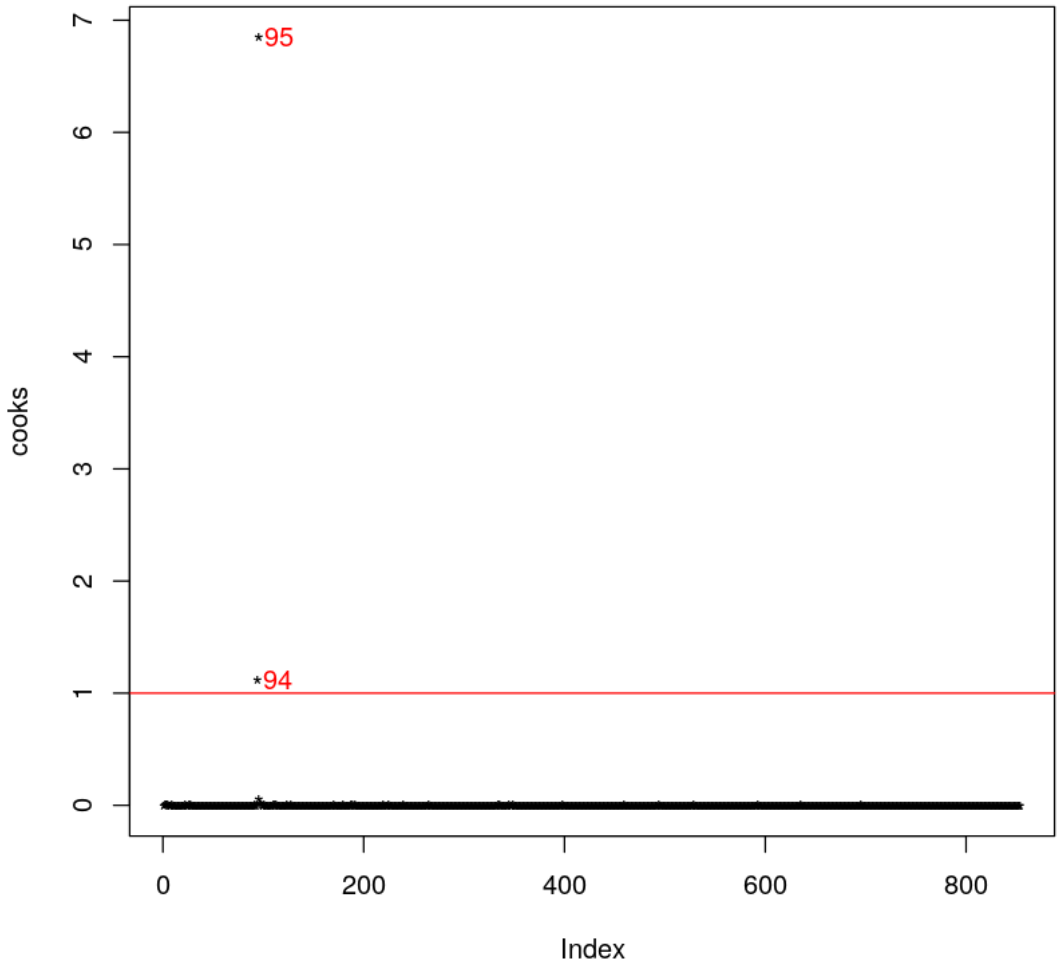
d) Identify outliers using Cook's distance approach. Remove points having Cook's distance > 1. Rerun the model after the removal of these points and print the summary. (4 points)

In [10]:
```
# SOLUTION BEGINS HERE
# visualize outliers
lower.panel <- function(x, y){
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    r <- round(cor(x, y), digits=2)
    txt <- paste0("R = ", r)
    cex.cor <- 0.8/strwidth(txt)
    text(0.5, 0.5, txt, cex = cex.cor * r)
}
pairs(data,
      lower.panel = lower.panel)

# install.packages("tidyverse", lib="../work/")
# library("tidyverse", lib.loc="../work/")

# identifying outliers using cooks
cooks = cooks.distance(model)
plot(cooks, pch="*")
abline(h=1,col="red")
text(x=1:length(cooks)+20, y=cooks, label=ifelse(cooks>1, names(cooks), ""), col="red
```


room_type
reviews
overall_satisfaction
accommodates
bedrooms
price
R = -0.05
R = 0.33
R = -0.41
R = -0.12
R = -0.09
R = -0.28
R = -0.11
R = -0.09
R = 0.8
R = -0.2
R = -0.09
R = -0.14
R = 0.51
R = 0.54




```
In [19]: as.numeric(names(cooks[cooks>1]))
```

1. 94 2. 95

```
In [4]: # removing outlier
        outlier = as.numeric(names(cooks[cooks>1]))
        data_clean = data[-outlier,]

        # model without outlier
        model_clean = lm(price~., data_clean)
        summary(model_clean)
        predict(model_clean, data_pred)
        # SOLUTION ENDS HERE
```

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

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
```

1: 285.503591258805

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)

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

```
In [5]: # SOLUTION BEGINS HERE
        data = read.csv("../resource/asnlib/publicdata/direct_marketing.csv")
        str(data)
        head(data)
```

```r
data = within(data, History <- relevel(History, ref = 4))
str(data)

model = lm(AmountSpent~Salary + History + Salary*History, data)
summary(model)
# AmountSpent=1.96+0.002Salary+25.45Low+79.30Medium+72.67High-0.002LowSalary-0.002Medi
# SOLUTION ENDS HERE
```

```
'data.frame':        1000 obs. of  10 variables:
 $ Age        : Factor w/ 3 levels "Middle","Old",..: 2 1 3 1 1 3 1 1 1 2 ...
 $ Gender     : Factor w/ 2 levels "Female","Male": 1 2 1 2 1 2 1 2 1 2 ...
 $ OwnHome    : Factor w/ 2 levels "Own","Rent": 1 2 2 1 1 1 2 1 1 1 ...
 $ Married    : Factor w/ 2 levels "Married","Single": 2 2 2 1 2 1 2 2 1 1 ...
 $ Location   : Factor w/ 2 levels "Close","Far": 2 1 1 1 1 1 1 1 1 2 ...
 $ Salary     : int  47500 63600 13500 85600 68400 30400 48100 68400 51900 80700 ...
 $ Children   : int  0 0 0 1 0 0 0 0 3 0 ...
 $ History    : Factor w/ 4 levels "High","Low","Medium",..: 1 1 2 1 1 2 3 1 2 4 ...
 $ Catalogs   : int  6 6 18 18 12 6 12 18 6 18 ...
 $ AmountSpent: num  75.5 131.8 29.6 243.6 130.4 ...
```

A data.frame: 6 Œ 10

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

```
'data.frame':        1000 obs. of  10 variables:
 $ Age        : Factor w/ 3 levels "Middle","Old",..: 2 1 3 1 1 3 1 1 1 2 ...
 $ Gender     : Factor w/ 2 levels "Female","Male": 1 2 1 2 1 2 1 2 1 2 ...
 $ OwnHome    : Factor w/ 2 levels "Own","Rent": 1 2 2 1 1 1 2 1 1 1 ...
 $ Married    : Factor w/ 2 levels "Married","Single": 2 2 2 1 2 1 2 2 1 1 ...
 $ Location   : Factor w/ 2 levels "Close","Far": 2 1 1 1 1 1 1 1 1 2 ...
 $ Salary     : int  47500 63600 13500 85600 68400 30400 48100 68400 51900 80700 ...
 $ Children   : int  0 0 0 1 0 0 0 0 3 0 ...
 $ History    : Factor w/ 4 levels "None","High",..: 2 2 3 2 2 3 4 2 3 1 ...
 $ Catalogs   : int  6 6 18 18 12 6 12 18 6 18 ...
 $ AmountSpent: num  75.5 131.8 29.6 243.6 130.4 ...
```

```
Call:
lm(formula = AmountSpent ~ Salary + History + Salary * History,
    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 ***
HistoryHigh          72.6735221 15.2270169   4.773 2.09e-06 ***
HistoryLow           25.4466733  8.9203292   2.853 0.004426 **
HistoryMedium        79.2984388 12.8982169   6.148 1.14e-09 ***
Salary:HistoryHigh   -0.0006408  0.0001926  -3.328 0.000908 ***
Salary:HistoryLow    -0.0021069  0.0001890 -11.150  < 2e-16 ***
Salary:HistoryMedium -0.0021153  0.0002182  -9.693  < 2e-16 ***
---
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 [6]: # SOLUTION BEGINS HERE
        data_none = data.frame(Salary = c(10000),
                               History = c("None"))
        data_low = data.frame(Salary = c(10000),
                              History = c("Low"))
        data_medium = data.frame(Salary = c(10000),
                                 History = c("Medium"))
        data_high = data.frame(Salary = c(10000),
                               History = c("High"))
        print(paste0("None: ", predict(model, data_none)))
        print(paste0("Low: ", predict(model, data_low)))
        print(paste0("Medium: ", predict(model, data_medium)))
        print(paste0("High: ", predict(model, data_high)))
        # SOLUTION ENDS HERE

[1] "None: 25.603473247411"
[1] "Low: 29.9815735873687"
[1] "Medium: 83.7490867699951"
[1] "High: 91.8687402713129"
```

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 [7]: # SOLUTION BEGINS HERE
        data = read.csv("../resource/asnlib/publicdata/airbnb_data.csv")
        str(data)

        # linear-linear
        model_linear_linear = lm(price~overall_satisfaction, data)
        summary(model_linear_linear)

        # linear-log
        model_linear_log = lm(price~log(overall_satisfaction+1), data)
        summary(model_linear_log)

        # log-linear
        model_log_linear = lm(log(price)~overall_satisfaction, data)
        summary(model_log_linear)

        # log-log
        model_log_log = lm(log(price+1)~log(overall_satisfaction+1), data)
        summary(model_log_log)

        # SOLUTION ENDS HERE
```

```
'data.frame':        854 obs. of  10 variables:
 $ room_id            : int  15771735 18284194 18091012 12286328 156926 12989718 6765734 16185
 $ survey_id          : int  1498 1498 1498 1498 1498 1498 1498 1498 1498 1498 ...
 $ host_id            : int  101992409 126414164 122380971 746673 746673 746673 746673 4857905
 $ room_type          : Factor w/ 3 levels "Entire home/apt",..: 3 3 3 3 3 3 3 3 1 1 ...
 $ city               : Factor w/ 1 level "Asheville": 1 1 1 1 1 1 1 1 1 1 ...
 $ reviews            : int  0 32 4 24 152 20 52 14 3 30 ...
 $ overall_satisfaction: num  0 5 4.5 4.5 4.5 4.5 4.5 4.5 5 5 ...
 $ accommodates       : int  4 4 2 6 6 4 5 2 6 5 ...
 $ bedrooms           : int  1 1 1 1 1 1 1 1 3 2 ...
 $ price              : int  67 76 45 26 26 26 26 20 46 150 ...
```

```
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




Call:
lm(formula = price ~ log(overall_satisfaction + 1), 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 ***
log(overall_satisfaction + 1)  -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




Call:
lm(formula = log(price) ~ 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
```

```
Call:
lm(formula = log(price + 1) ~ log(overall_satisfaction + 1),
    data = data)

Residuals:
    Min      1Q  Median      3Q     Max
-1.5657 -0.3522 -0.0338  0.3255  3.7030

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                    4.81441    0.05048  95.364  < 2e-16 ***
log(overall_satisfaction + 1) -0.12684    0.03080  -4.118  4.2e-05 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 0.5696 on 852 degrees of freedom
Multiple R-squared:  0.01951,Adjusted R-squared:  0.01836
F-statistic: 16.96 on 1 and 852 DF,  p-value: 4.196e-05
```

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)

linear-log model has the best R-squared: 0.02089, Adjusted R-squared: 0.01974. In business sense, overall satisfaction might not be that of a significant or logical indicator on price even if it is shown in the model summary. Using multiple variables such as location and room type will probably provide more information on price variability.

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 [8]: # SOLUTION BEGINS HERE
        data = read.csv("../resource/asnlib/publicdata/titanic_data.csv")
        str(data)
        head(data)

        model = glm(Survived~Sex, data, family = "binomial")
        summary(model)
        # SOLUTION ENDS HERE
```

```
'data.frame':	756 obs. of  5 variables:
 $ Name    : Factor w/ 753 levels "Abbing, Mr Anthony",..: 22 25 26 27 24 31 44 45 49 53 ...
 $ PClass  : Factor w/ 3 levels "1st","2nd","3rd": 1 1 1 1 1 1 1 1 1 1 ...
 $ Age     : num  29 2 30 25 0.92 47 63 39 58 71 ...
 $ Sex     : Factor w/ 2 levels "female","male": 1 1 2 1 2 2 1 2 1 2 ...
 $ Survived: int  1 0 0 0 1 1 1 0 1 0 ...
```

A data.frame: 6 Œ 5

	Name <fct>	PClass <fct>	Age <dbl>	Sex <fct>	Survived <int>
	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", 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)

The intercept value of 1.1172 is the log of odds of survival for females since they are the reference group.

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

log(p/(1-p)) = 1.1172 - 2.4718(0) p/(1-p) = exp(1.1172) = 3.05628461455 p = 3.05628461455 - 3.05628461455p 4.05628461455p = 3.05628461455 p = 0.75346897591 Probability of survival for females ~ 0.75

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

log(p/(1-p)) = 1.1172 - 2.4718(1) = -1.3546 p/(1-p) = exp(-1.3546) = 0.258050494 p = 0.258050494 - 0.258050494p 1.258050494p = 0.258050494 p = 0.20511934555 Probability of survival for males ~ 0.21