Homework 1

9/16/2020

Question 1

Use the airbnb_data.csv provided and answer the following questions (Q1a. b. c. and d.) on Linear Regression.

Importing the data:

```
airbnb <- read.csv("airbnb_data.csv", header = TRUE)
summary(airbnb)
```

```
##     room_id           survey_id       host_id           room_type
##  Min.   :   67870   Min.   :1498   Min.   :    62667   Length:854
##  1st Qu.: 6413734   1st Qu.:1498   1st Qu.:  6453926   Class :character
##  Median :13329838   Median :1498   Median : 22920130   Mode  :character
##  Mean   :11672573   Mean   :1498   Mean   : 37877449
##  3rd Qu.:16856088   3rd Qu.:1498   3rd Qu.: 58634762
##  Max.   :19912932   Max.   :1498   Max.   :141036151
##      city              reviews       overall_satisfaction  accommodates
##  Length:854         Min.   :  0.00   Min.   :0.00         Min.   : 1.000
##  Class :character   1st Qu.:  8.00   1st Qu.:4.50         1st Qu.: 2.000
##  Mode  :character   Median : 28.00   Median :5.00         Median : 3.000
##                     Mean   : 49.11   Mean   :4.18         Mean   : 3.412
##                     3rd Qu.: 65.00   3rd Qu.:5.00         3rd Qu.: 4.000
##                     Max.   :602.00   Max.   :5.00         Max.   :17.000
##     bedrooms          price
##  Min.   : 0.000   Min.   :  20.0
##  1st Qu.: 1.000   1st Qu.:  70.0
##  Median : 1.000   Median :  95.0
##  Mean   : 1.352   Mean   : 126.6
##  3rd Qu.: 2.000   3rd Qu.: 139.0
##  Max.   :10.000   Max.   :5000.0
```

a) Fit a multiple linear regression model using price as the response variable and all others as predictor variables (Note: remove 'id' columns). Which variables are statistically significant in determining the price?

```
# Removing the id columns
model_data = select(airbnb, -c("room_id", "survey_id", "host_id", "city"))

# Fitting & outputting the model
airbnb_model <- lm(price ~ room_type + reviews + overall_satisfaction + accommodates +
    bedrooms, data = model_data)
summary(airbnb_model)
```

```
##
## Call:
## lm(formula = price ~ room_type + reviews + overall_satisfaction +
##     accommodates + bedrooms, data = model_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 p value, the room type (room_typePrivate room and room_typeShared room columns in the data set) and reviews are not statistically significant as their p value is greater than 0.05. The remaining predictors (overall_satisfaction, accomodates and bedrooms) are the statisically significant predictors of price.

b) Interpret the coefficients for predictors: room type(Shared Room), bedrooms?

Room Type (Shared Room): if the Airbnb property is a shared room, the price will decrease by $76.68
Bedrooms: the addition of one extra bedroom will increase the price by $85.65

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'.

```
listing = data.frame(bedrooms = 1, accommodates = 2, reviews = 870, overall_satisfaction = 4,
    room_type = "Private room")
predict(airbnb_model, newdata = listing)
```

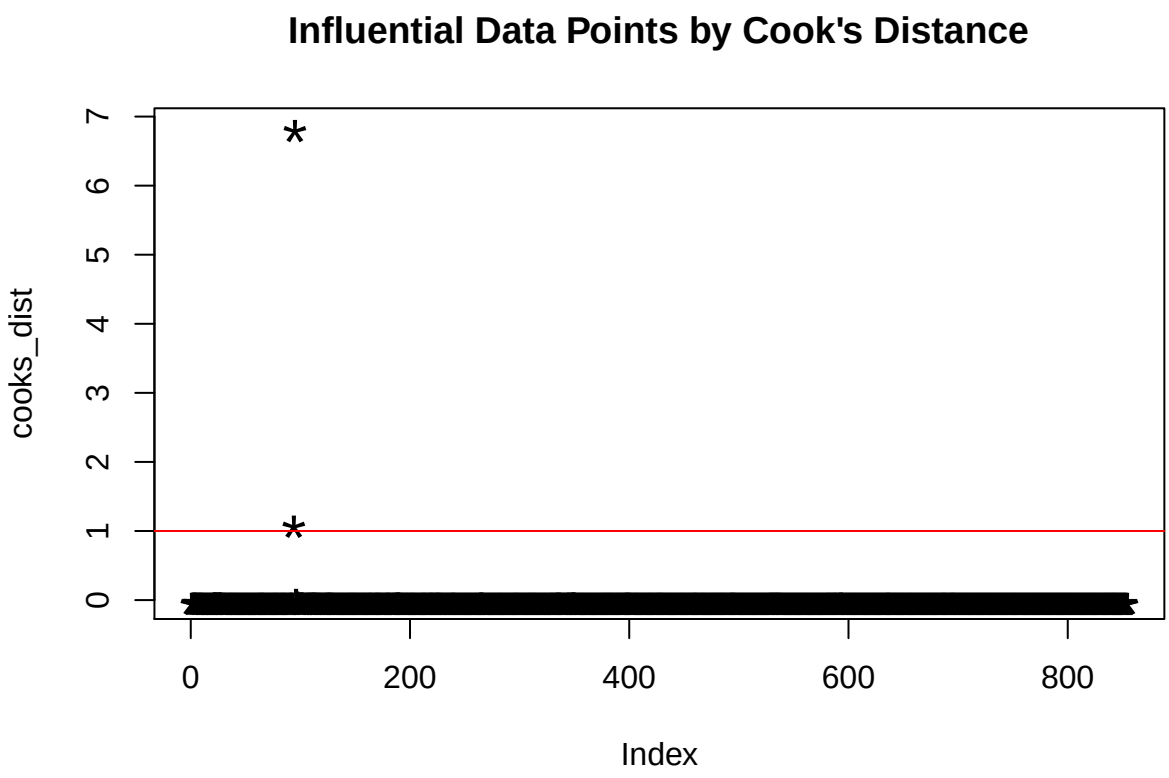
```
##        1
## 74.91967
```

Based on the specs of the given listing and the model fitted in part a, the price of this listing is $75.

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.

```r
# Calculate Cook's distance
cooks_dist <- cooks.distance(airbnb_model)

# Graphing Cook's distance
sample_size <- nrow(model_data)
plot(cooks_dist, pch = "*", cex = 2, main = "Influential Data Points by Cook's Distance")
abline(h = 1, col = "red")  # add cutoff line
```




```r
# Identifying whic points have Cook's distance > 1 and removing them from the
# data set
influential <- as.numeric(names(cooks_dist)[(cooks_dist > 1)])
model_data2 <- model_data[-influential, ]

# Creating a new model based on the data without the influential points
airbnb_model2 <- lm(price ~ room_type + reviews + overall_satisfaction + accommodates +
    bedrooms, data = model_data2)
summary(airbnb_model2)
```

```
##
## Call:
## lm(formula = price ~ room_type + reviews + overall_satisfaction +
##     accommodates + bedrooms, data = model_data2)
##
## 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
```

After removing the influential points, the room type is now statistically significant. The R squared and adj. R squared value of this model is better than the previous model that includes the influential points.

Question 2

Use the direct_marketing.csv provided and answer the following questions (Q2.a and Q2.b) on Linear Regression.

Importing the data:

```
dir_market <- read.csv("direct_marketing.csv", header = TRUE)
summary(dir_market)
```

```
##      Age               Gender            OwnHome            Married
##  Length:1000        Length:1000        Length:1000        Length:1000
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    Location             Salary          Children       History
##  Length:1000        Min.   : 10100   Min.   :0.000   Length:1000
##  Class :character   1st Qu.: 29975   1st Qu.:0.000   Class :character
##  Mode  :character   Median : 53700   Median :1.000   Mode  :character
##                     Mean   : 56104   Mean   :0.934
##                     3rd Qu.: 77025   3rd Qu.:2.000
##                     Max.   :168800   Max.   :3.000
##     Catalogs      AmountSpent
##  Min.   : 6.00   Min.   :  3.80
##  1st Qu.: 6.00   1st Qu.: 48.83
##  Median :12.00   Median : 96.20
##  Mean   :14.68   Mean   :121.68
##  3rd Qu.:18.00   3rd Qu.:168.85
##  Max.   :24.00   Max.   :621.70
```

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.

```
# Creating indicator variables for History column
dir_market <- dir_market %>% mutate(history_low = ifelse(History == "Low", 1, 0)) %>%
    mutate(history_med = ifelse(History == "Medium", 1, 0)) %>% mutate(history_high = ifelse(History ==
    "High", 1, 0))

# Creating interaction terms
dir_market <- dir_market %>% mutate(LowSalary = history_low * Salary) %>% mutate(MediumSalary = history_
    Salary) %>% mutate(HighSalary = history_high * Salary)

summary(dir_market)
```

```
##       Age              Gender            OwnHome            Married
##  Length:1000        Length:1000        Length:1000        Length:1000
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    Location             Salary          Children       History
##  Length:1000        Min.   : 10100   Min.   :0.000   Length:1000
##  Class :character   1st Qu.: 29975   1st Qu.:0.000   Class :character
##  Mode  :character   Median : 53700   Median :1.000   Mode  :character
##                     Mean   : 56104   Mean   :0.934
##                     3rd Qu.: 77025   3rd Qu.:2.000
##                     Max.   :168800   Max.   :3.000
##     Catalogs      AmountSpent      history_low    history_med
##  Min.   : 6.00   Min.   :  3.80   Min.   :0.00   Min.   :0.000
##  1st Qu.: 6.00   1st Qu.: 48.83   1st Qu.:0.00   1st Qu.:0.000
##  Median :12.00   Median : 96.20   Median :0.00   Median :0.000
##  Mean   :14.68   Mean   :121.68   Mean   :0.23   Mean   :0.212
##  3rd Qu.:18.00   3rd Qu.:168.85   3rd Qu.:0.00   3rd Qu.:0.000
##  Max.   :24.00   Max.   :621.70   Max.   :1.00   Max.   :1.000
##   history_high     LowSalary       MediumSalary      HighSalary
##  Min.   :0.000   Min.   :     0   Min.   :     0   Min.   :     0
##  1st Qu.:0.000   1st Qu.:     0   1st Qu.:     0   1st Qu.:     0
##  Median :0.000   Median :     0   Median :     0   Median :     0
##  Mean   :0.255   Mean   :  7420   Mean   : 11739   Mean   : 21305
##  3rd Qu.:1.000   3rd Qu.:     0   3rd Qu.:     0   3rd Qu.: 44550
##  Max.   :1.000   Max.   :118000   Max.   :140700   Max.   :168800
```

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.

```
# Creating the model
dm_model <- lm(AmountSpent ~ history_low + history_med + history_high + LowSalary +
    MediumSalary + HighSalary, data = dir_market)

summary(dm_model)
```

```
##
## Call:
## lm(formula = AmountSpent ~ history_low + history_med + history_high +
##     LowSalary + MediumSalary + HighSalary, data = dir_market)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -214.33  -35.19   -7.49   25.17  374.41
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.240e+02  3.912e+00  31.694  < 2e-16 ***
## history_low  -9.658e+01  8.548e+00 -11.299  < 2e-16 ***
## history_med  -4.273e+01  1.423e+01  -3.004  0.00274 **
## history_high -4.935e+01  1.732e+01  -2.850  0.00447 **
## LowSalary     2.573e-04  1.901e-04   1.354  0.17620
## MediumSalary  2.488e-04  2.321e-04   1.072  0.28397
## HighSalary    1.723e-03  1.954e-04   8.820  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 68.1 on 993 degrees of freedom
## Multiple R-squared:  0.501,  Adjusted R-squared:  0.498
## F-statistic: 166.1 on 6 and 993 DF,  p-value: < 2.2e-16
```

b) What is the amount spent by a customer for each historic type provided their salary is $10,000 based on the model constructed in part a?

```
none_sal = data.frame(history_low = 0, history_med = 0, history_high = 0, LowSalary = 0,
    MediumSalary = 0, HighSalary = 0)
low_sal = data.frame(history_low = 1, history_med = 0, history_high = 0, LowSalary = 10000,
    MediumSalary = 0, HighSalary = 0)
med_sal = data.frame(history_low = 0, history_med = 1, history_high = 0, LowSalary = 0,
    MediumSalary = 10000, HighSalary = 0)
high_sal = data.frame(history_low = 0, history_med = 0, history_high = 1, LowSalary = 0,
    MediumSalary = 0, HighSalary = 10000)

none_amountspent = predict(dm_model, newdata = none_sal)
low_amountspent = predict(dm_model, newdata = low_sal)
med_amountspent = predict(dm_model, newdata = med_sal)
high_amountspent = predict(dm_model, newdata = high_sal)

cat(none_amountspent, "\n", low_amountspent, "\n", med_amountspent, "\n", high_amountspent)
```

```
## 123.9901
##  29.98157
##  83.74909
##  91.86874
```

Given a salary of $10,000, a customer with no history will spend $124, a customer with low history will spend $30, a customer with medium history will spend $84 and a customer with high history will spend $92.

Use the airbnb_data.csv Preview the document provided and answer the following questions (Q2.c and Q2.d) on Linear Regression. 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.

```
# Generating the four lin/log models. Since overall_statisfaction has values of
# 0, I use the function log1p instead of log to add 1 to every data point
lin_lin <- lm(price ~ overall_satisfaction, data = airbnb)
lin_log <- lm(price ~ log1p(overall_satisfaction), data = airbnb)
log_lin <- lm(log(price) ~ overall_satisfaction, data = airbnb)
log_log <- lm(log(price) ~ log1p(overall_satisfaction), data = airbnb)
```

d) Which of the four models has the best R2? Do you have any comments on the choice of the dependent variable?

```
cat("R Squared Values:", "\n", "Linear-Linear:", summary(lin_lin)$r.squared, "\n",
    "Linear-Log:", summary(lin_log)$r.squared, "\n", "Log-Linear", summary(log_lin)$r.squared,
    "\n", "Log-Log", summary(log_log)$r.squared, "\n")
```

```
## R Squared Values:
##  Linear-Linear: 0.02018428
##  Linear-Log: 0.02088739
##  Log-Linear 0.01777027
##  Log-Log 0.01933861
```

```
summary(lin_log)
```

```
##
## Call:
## lm(formula = price ~ log1p(overall_satisfaction), data = airbnb)
##
## 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 ***
## log1p(overall_satisfaction)    -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
```

The model with the best R squared value is the linear-log model. However, the R squared value for all 4 models is very low. Even though overall satisfaction appears to be statistically significant, the low R squared value is an indication that overall satisfaction does not explain much of the variation in the price and another dependant variable should replace overall satisfaction as a predictor or be addded to the model.

Question 3

The attached titanic_data.csv has been cleaned to remove all rows which contain missing values. We will perform logistic regression on this cleaned dataset.

The dataset contains the following columns:

'Name' - Passenger Name - factor

'PClass' - Passenger Class (1st, 2nd, 3rd) - factor

'Age' - Passenger Age - number

'Sex' - Passenger Sex – female, male

'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 logistic regression on the dataset using 'Survived' as the response and 'Sex' as the explanatory variable.

a) Display the model summary.

```
# Importing the data
titanic <- read.csv("titanic_data.csv", header = TRUE)

# Converting Survived to a factor and Sex to an indicator variable
titanic$Survived <- as.factor(titanic$Survived)
titanic <- titanic %>% mutate(Sex = ifelse(Sex == "female", 1, 0))

# Fitting the model
titanic_model <- glm(Survived ~ Sex, data = titanic, family = "binomial")
summary(titanic_model)
```

```
##
## Call:
## glm(formula = Survived ~ Sex, family = "binomial", data = titanic)
##
## 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)  -1.3545     0.1145  -11.83   <2e-16 ***
## Sex           2.4718     0.1783   13.86   <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?

The intercept coefficient of -1.3545 is the log odds for the base case of Sex = 0 or Sex = male. It is used to calculate the probability that Survived = 1, given that a person is male in p(x) = exp(b0+b1x)/[1 + exp(b0+b1x)]

c) Determine the probability of survival for females.

```
b0 = coef(titanic_model)[1]
b1 = coef(titanic_model)[2]
x = 1
p_female = exp(b0 + b1 * x)/(1 + exp(b0 + b1 * x))

print(p_female)
```

```
## (Intercept)
##   0.7534722
```

The probability of survival for females is 75.35%.

d) Determine the probability of survival for males.

```
b0 = coef(titanic_model)[1]
b1 = coef(titanic_model)[2]
x = 0
p_male = exp(b0 + b1 * x)/(1 + exp(b0 + b1 * x))

print(p_male)
```

```
## (Intercept)
##   0.2051282
```

The probability of survival for males is 20.51%.