Commercial Aviation Final Project

Tiffany Tse, Justine Huynh, Quan Nguyen

11/12/2019

Project Overview

Question/Problem Statement

What factors significantly impact consumer demand of airlines? Why do we ask this question? To help airlines become aware of areas they can improve to ensure passengers have best flight experience possible.

Data

Our data, airlineData, comes from the UK airline quality assessment website,Skytrax. The initial data frame, provided by NYU business and web intern, Fred Zeng, consist of 3898 observations and 19 variables with travelers' reviews from April 2013-July 2019. Customer's satisfaction is measured on aircraft models, seat type, travel type, entertainment, food, ground service, seat comfort, service, and value ratings.

Motivation

We were interested on this project because we have always been avid plane spotters since we were young, especially Tiffany and Quan. Since a young age, we have always dreamed of soaring through the air under a sky full of stars. While we cannot grow wings to fly, we can definitely ride airplanes, which leads to the next question: what determines how amazing an airline can be? We are also interested in applying the practice of this mentally active hobby to help major U.S. airlines, which is often negatively rated for top reasons: bad customer service, delay, and cancellations.

Hypotheses, Task, Methods, Challenges

We were intereted in the following hypotheses: 1) Seat Comfort, Entertainment, Ground Service, Cabin Service, and Food positively influence customer satisfaction of the airline. 2) Travel Type and Seat Type have no direct effect on overall quality of the airline.

The journey to answer these burning questions are mainly explanatory and will involve regression as well as plots, specifically ggplot(). At first, we experienced challenges in finding a relevant data set with sufficient info to answer the questions we raised and self-learned web scraping in R.Some other challenges included using Lasso for variable selection, checking for non-constant variance, extraordinary residual vs. fitted plots, and generally confusing user input. For instance, when asked about what model aircraft they flew on, some passengers did not know and subsequently guessed their aircraft. Others mispelled the aircraft's name.

Analysis

Data Pre-processing

```r
# clean international airline data

# check which observations are NA in Recommended column and replace with "no"
# get levels and add "no"
levels <- levels(airlineData$Recommended)
levels[length(levels) + 1] <- "no"
```

```r
# refactor Recommended to include "no" as a factor level
airlineData$Recommended <- factor(airlineData$Recommended, levels = levels)
airlineData$Recommended[is.na(airlineData$Recommended)] <- "no"

# clean AircraftModel column by first checking the column's levels
aircraft_models <- levels(airlineData$AircraftModel)
# replace NAs in AircraftModel column with level "Unknown"
airlineData$AircraftModel[is.na(airlineData$AircraftModel)] <- "Unknown"

# check whether there are unnecessary strings
str(airlineData)
```

```
## 'data.frame':    3898 obs. of  19 variables:
##  $ AirName            : Factor w/ 10 levels "ANA All Nippon Airways",..: 10 10 10 10 10 10 10 10 10 1
##  $ AircraftModel      : Factor w/ 368 levels " A330-300 /  Boeing B777-300ER",..: 367 367 367 45 367
##  $ Comments           : Factor w/ 1 level " |  Jakarta to Tokyo Haneda. Best airline experience I've
##  $ DateFlown          : Factor w/ 62 levels "Apr-13","Apr-15",..: 47 47 11 47 47 47 47 47 47 47 ...
##  $ EntertainmentRating: int  3 5 4 0 2 2 3 2 0 1 ...
##  $ FoodRating         : int  1 5 5 3 1 1 3 3 1 1 ...
##  $ GroundServiceRating: int  4 4 5 3 3 3 3 2 1 1 ...
##  $ OverallScore       : int  1 10 10 5 3 6 7 7 1 2 ...
##  $ Recommended        : Factor w/ 2 levels "yes","no": 2 1 1 2 2 1 1 2 2 2 ...
##  $ ReviewDate         : Factor w/ 1679 levels "10th April 2011",..: 1253 1150 1042 990 990 773 657 2
##  $ ReviewTitle        : Factor w/ 2764 levels " \"crew extremely polite and helpful\"",..: 2645 2687
##  $ ReviewrCountry     : Factor w/ 48 levels " (Argentina) ",..: 45 45 45 45 45 45 45 45 45 45 ...
##  $ Route              : Factor w/ 2012 levels " Frankfurt to Tokyo",..: 746 1696 54 1413 1679 1676 1
##  $ SeatComfortRating  : int  3 4 5 3 4 2 4 2 1 1 ...
##  $ SeatType           : Factor w/ 4 levels "Business Class",..: 2 2 2 2 2 2 2 1 4 2 ...
##  $ ServiceRating      : int  1 5 5 2 1 2 4 5 1 2 ...
##  $ TravelType         : Factor w/ 4 levels "Business","Couple Leisure",..: 1 3 3 2 2 2 4 2 1 2 ...
##  $ ValueRating        : int  1 5 5 3 2 2 3 3 1 1 ...
##  $ WifiRating         : int  0 0 0 0 0 1 4 2 0 0 ...
```

```r
# separate feature and response data
OverallScore <- airlineData$OverallScore
indx_overallScore <- which(names(airlineData) == "OverallScore") # locate response column

# data frame of all the predictors w/o the response (OverallScore)
#x_vars <- airlineData[, -indx_overallScore]
```

Getting rid of missing data

```r
# our strategy: 1) remove variables/columns with a lot (> 50%) NAs (missing values)
# 2) remove observations/rows with NAs

# determining insignificant columns
# count the number of NAs in each column
(how_many_nas <- sapply(airlineData, function(x) sum(is.na(x))))
```

```
##             AirName       AircraftModel            Comments
##                   0                   0                3878
##           DateFlown EntertainmentRating          FoodRating
##                 803                   0                   0
## GroundServiceRating        OverallScore         Recommended
##                    0                  14                   0
##           ReviewDate         ReviewTitle      ReviewrCountry
##                    0                   0                   0
##                Route   SeatComfortRating            SeatType
##                  802                   0                   1
##        ServiceRating          TravelType         ValueRating
##                    0                 800                   0
##           WifiRating
##                    0
```

```
# we noticed Comments, DateFlown, Route, TravelType, OverallScore, SeatType variables contain NAs

# more than 1949 NAs
indx_remove <- which(how_many_nas > dim(airlineData)[1]/2)
names(airlineData[indx_remove]) # variables to remove
```

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

```
# remove Comments column as it's the only column w/majority of NAs
# and this column does not provide useful info
airlinesRefined <- airlineData[, -indx_remove] # df w/o Comments column
```

We have removed the Comments column from our data frame due to an exorbitant amount of NAs.

Determining rating columns with numerous zeros (equal to NAs)

```
# realized the zeros in rating columns are equivalent to NAs

# display zeros per variable
cols_many_zeros <- sapply(airlinesRefined, function(x) sum(x == 0))
(indx_cols_many_zeros <- which(cols_many_zeros > dim(airlinesRefined)[1]/2))
```

```
## WifiRating
##         18
```

```
airlinesRefined <- airlinesRefined[, -indx_cols_many_zeros] # df w/o Comments & WiFi variables
str(airlinesRefined)
```

```
## 'data.frame':    3898 obs. of  17 variables:
##  $ AirName            : Factor w/ 10 levels "ANA All Nippon Airways",..: 10 10 10 10 10 10 10 10 10
##  $ AircraftModel      : Factor w/ 368 levels " A330-300 /  Boeing B777-300ER",..: 367 367 367 45 367
##  $ DateFlown          : Factor w/ 62 levels "Apr-13","Apr-15",..: 47 47 11 47 47 47 47 47 47 47 ...
##  $ EntertainmentRating: int  3 5 4 0 2 2 3 2 0 1 ...
##  $ FoodRating         : int  1 5 5 3 1 1 3 3 1 1 ...
##  $ GroundServiceRating: int  4 4 5 3 3 3 3 2 1 1 ...
##  $ OverallScore       : int  1 10 10 5 3 6 7 7 1 2 ...
##  $ Recommended        : Factor w/ 2 levels "yes","no": 2 1 1 2 2 1 1 2 2 2 ...
##  $ ReviewDate         : Factor w/ 1679 levels "10th April 2011",..: 1253 1150 1042 990 990 773 657 2
##  $ ReviewTitle        : Factor w/ 2764 levels " \"crew extremely polite and helpful\"",..: 2645 2687
##  $ ReviewrCountry     : Factor w/ 48 levels " (Argentina) ",..: 45 45 45 45 45 45 45 45 45 45 ...
##  $ Route              : Factor w/ 2012 levels " Frankfurt to Tokyo",..: 746 1696 54 1413 1679 1676 1
##  $ SeatComfortRating  : int  3 4 5 3 4 2 4 2 1 1 ...
##  $ SeatType           : Factor w/ 4 levels "Business Class",..: 2 2 2 2 2 2 2 1 4 2 ...
##  $ ServiceRating      : int  1 5 5 2 1 2 4 5 1 2 ...
##  $ TravelType         : Factor w/ 4 levels "Business","Couple Leisure",..: 1 3 3 2 2 2 4 2 1 2 ...
##  $ ValueRating        : int  1 5 5 3 2 2 3 3 1 1 ...
```

Since more than half of the travelers between the time period April 2013 - July 2019 did not provide a rating for the WiFi field in their reviews (zeros are equivalent to NAs), we've decided to remove the WiFi rating column from our data frame.

Checking and removing unnecessary observations

```
nas_per_row <- apply(airlinesRefined, 1, function(x) sum(is.na(x)))
#summary(nas_per_row) # max. NAs in rows is 4

indx_remove_rows <- which(nas_per_row > dim(airlinesRefined)[2]/2)
# none of the reviewers left more than 1/2 the questions as NAs or blanks
# travelers (passengers) mainly completed most of the fields in their reviews

#which.max(nas_per_row) # stops at first occurrence...so instead,
most_NA_reviews <- which(nas_per_row == max(nas_per_row)) # obs. with most NAs
no_NA_reviews <- which(nas_per_row == min(nas_per_row)) # obs. with no NAs

df_NA_reviews <- airlinesRefined[most_NA_reviews, ]
# reviewers did not provide AircraftModel, DateFlown, GroundServiceRating, OverallScore, Route,
# and TravelType
# fair to omit these rows

airlines_no_NA <- na.omit(airlinesRefined) # remove all observations with NAs

# double check there are no remaining missing data
sum(is.na(airlines_no_NA))
```

```
## [1] 0
```

```
#str(airlines_no_NA)
```

We have eliminated all observations with NAs in our data frame. The updated data frame is called airlinesRefined.

Cleaning AircraftModel column

We do not want the observations with "Various" as aircraft model, so we will need to clean that column. This applies to the observations (reviews) where more than one aircraft model were provided, suggesting that they were clearly unsure of which aircraft they flew on (unreliable info). Ironically, some reviewers either mistakenly or unknowingly put the flight # as the aircraft model of their flight. One reviewer went so far as to putting the 2nd generation BMW X5 mid-size luxury crossover SUV as the aircraft model.

```
# do not want observations with Various as an aircraft model
# index of observation containing Various as an aircraft model
various_model <- which(airlines_no_NA$AircraftModel == "Various")
airlines_no_NA <- airlines_no_NA[-various_model,] # updated df

# remove AircraftModel observations containing specified special characters/words
no_chars <- airlines_no_NA[!grepl("/|and|,|&|then", airlines_no_NA$AircraftModel),] # updated df

no_chars$AircraftModel <- str_replace_all(no_chars$AircraftModel, "320 neo", "320neo")

# sub changes only first occurrence within string
no_chars$AircraftModel <- sub("Boeing |Boeingv", "B", no_chars$AircraftModel)
no_chars$AircraftModel <- sub("Airbus |Airbus|Airbus A|A ", "A", no_chars$AircraftModel)
```

```r
no_chars$AircraftModel <- sub("CRJ-|CRJ ", "CRJ", no_chars$AircraftModel)
no_chars$AircraftModel <- sub(" .*|\\-.*", "", no_chars$AircraftModel)

# merge aircraft models under same family into one (parent) group
# Airbus-manufactured planes
no_chars$AircraftModel <- str_replace_all(no_chars$AircraftModel, "A319|A320neo|A321", "A320")
no_chars$AircraftModel <- str_replace_all(no_chars$AircraftModel, "B330|A333", "A330")
no_chars$AircraftModel <- str_replace_all(no_chars$AircraftModel, "A343", "A340")
no_chars$AircraftModel <- str_replace_all(no_chars$AircraftModel, "A359", "A350")
no_chars$AircraftModel <- str_replace_all(no_chars$AircraftModel, "A388", "A380")
no_chars$AircraftModel <- str_replace_all(no_chars$AircraftModel, "A388", "A380")

#Boeing-manufactured planes
no_chars$AircraftModel <- str_replace_all(no_chars$AircraftModel, "B744|B744C", "B747")
no_chars$AircraftModel <- str_replace_all(no_chars$AircraftModel, "B773|B773ER|B77W|777W|77L|B700LR", "
no_chars$AircraftModel <- str_replace_all(no_chars$AircraftModel, "Dreamliner|B788|B789", "B787")

#Bombardier-manufactured planes
no_chars$AircraftModel <- str_replace_all(no_chars$AircraftModel, "CRJ9|CRJ900|CRJ1000|Q400", "Bombardi

# removing incorrect/insufficient info for AircraftModel (mispelling or flight # given instead)
a_model <- which(no_chars$AircraftModel == "A")
b_model <- which(no_chars$AircraftModel == "Beoing")
b_model2 <- which(no_chars$AircraftModel == "737")
b_model3 <- which(no_chars$AircraftModel == "787")
b_model4 <- which(no_chars$AircraftModel == "777")
b_model5 <- which(no_chars$AircraftModel == "B747C")
b_model6 <- which(no_chars$AircraftModel == "B777ER")
c_model <- which(no_chars$AircraftModel == "Bombardier00")
flight1 <- which(no_chars$AircraftModel == "BR0051")
flight2 <- which(no_chars$AircraftModel == "HU7989")
flight3 <- which(no_chars$AircraftModel == "OZ751")
car <- which(no_chars$AircraftModel == "E70")

no_chars <- no_chars[-c(a_model, b_model, b_model2, b_model3, b_model4, b_model5, b_model6, c_model, fli

no_chars$AircraftModel <- factor(no_chars$AircraftModel)
#levels(no_chars$AircraftModel)
str(no_chars)
```

```
## 'data.frame':    2768 obs. of  17 variables:
##  $ AirName            : Factor w/ 10 levels "ANA All Nippon Airways",..: 10 10 10 10 10 10 10 10 10
##  $ AircraftModel      : Factor w/ 15 levels "A300","A320",..: 15 15 15 3 15 11 6 15 15 15 ...
##  $ DateFlown          : Factor w/ 62 levels "Apr-13","Apr-15",..: 47 47 11 47 47 47 47 47 47 30 ...
##  $ EntertainmentRating: int  3 5 4 0 2 2 3 0 1 4 ...
##  $ FoodRating         : int  1 5 5 3 1 1 3 1 1 4 ...
##  $ GroundServiceRating: int  4 4 5 3 3 3 3 1 1 4 ...
##  $ OverallScore       : int  1 10 10 5 3 6 7 1 2 9 ...
##  $ Recommended        : Factor w/ 2 levels "yes","no": 2 1 1 2 2 1 1 2 2 1 ...
##  $ ReviewDate         : Factor w/ 1679 levels "10th April 2011",..: 1253 1150 1042 990 990 773 657 2
##  $ ReviewTitle        : Factor w/ 2764 levels " \"crew extremely polite and helpful\"",..: 2645 2687
##  $ ReviewrCountry     : Factor w/ 48 levels " (Argentina) ",..: 45 45 45 45 45 45 45 45 45 21 ...
##  $ Route              : Factor w/ 2012 levels " Frankfurt to Tokyo",..: 746 1696 54 1413 1679 1676 1
##  $ SeatComfortRating  : int  3 4 5 3 4 2 4 1 1 4 ...
##  $ SeatType           : Factor w/ 4 levels "Business Class",..: 2 2 2 2 2 2 2 4 2 2 ...
##  $ ServiceRating      : int  1 5 5 2 1 2 4 1 2 5 ...
##  $ TravelType         : Factor w/ 4 levels "Business","Couple Leisure",..: 1 3 3 2 2 2 4 1 2 4 ...
##  $ ValueRating        : int  1 5 5 3 2 2 3 1 1 5 ...
##  - attr(*, "na.action")= 'omit' Named int  920 1079 1085 1086 1087 1088 1089 1090 1091 1092 ...
##   ..- attr(*, "names")= chr  "920" "1079" "1085" "1086" ...
```

Some people (like Justine, Tiffany, and Quan) are not proficient in distinguishing aircraft models. However, unlike Justine, Tiffany, and Quan, these people are completely honest about their lack of aircraft model expertise. Instead of guessing and writing down a random aircraft model, these people wrote “Unknown”, so the programmers could easily group these reviews into another factor level called “Unknown” instead of wonder how trustworthy these passengers were. Had the passengers simply guessed the aircraft model, our regression model may have had different results for every aircraft model. We would like to have a moment of silence to commmend these honorable travellers who were unafraid to reveal their true expertise in aircraft models.

Getting rid of variables that are not “variable”

Some columns need to be eliminated because they contain very similar values. For instance, if there were a WiFi rating column who consistently had 3s in the observations, then we can safely assume that WiFi rating does not significantly affect the overall score of the airlines.

Our strategy is to check the coefficient of variation (CV) AKA standard deviation divided by the mean.

```
# strategy is to check coefficient of variation (CV) aka sd/mean
# CV measures dispersion of the variable
# comparable across dif. variables as they are now measured on same scale
# large value suggests variable varies, value near 0 suggests little variation

# first convert to numeric data to run calculations
numericData <- sapply(no_chars, as.numeric)

columnMeans <- colMeans(numericData)
sds <- apply(numericData, 2, sd)
coef.variation <- sds/columnMeans

# let's look how variable our variables are
summary(coef.variation)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.3300  0.3811  0.4258  0.4531  0.5512  0.5774 
```

```
# keep all the variables whose variability factor > 0.05
# eliminate the variables whose variability is close to nonexistent (zero)
(indxVariablesToKeep <- which(coef.variation > 0.05))
```

```
##             AirName       AircraftModel           DateFlown 
##                   1                   2                   3 
## EntertainmentRating          FoodRating GroundServiceRating 
##                   4                   5                   6 
##        OverallScore         Recommended          ReviewDate 
##                   7                   8                   9 
##         ReviewTitle      ReviewrCountry               Route 
##                  10                  11                  12 
##   SeatComfortRating            SeatType       ServiceRating 
##                  13                  14                  15 
##          TravelType         ValueRating 
##                  16                   17
```

```r
# we have 17 variables in our model now

# re-refine our airlineData
airlinesCleaned <- no_chars[, indxVariablesToKeep]
str(airlinesCleaned)
```

```
## 'data.frame':    2768 obs. of  17 variables:
##  $ AirName            : Factor w/ 10 levels "ANA All Nippon Airways",..: 10 10 10 10 10 10 10 10 10 1
##  $ AircraftModel      : Factor w/ 15 levels "A300","A320",..: 15 15 15 3 15 11 6 15 15 15 ...
##  $ DateFlown          : Factor w/ 62 levels "Apr-13","Apr-15",..: 47 47 11 47 47 47 47 47 47 30 ...
##  $ EntertainmentRating: int  3 5 4 0 2 2 3 0 1 4 ...
##  $ FoodRating         : int  1 5 5 3 1 1 3 1 1 4 ...
##  $ GroundServiceRating: int  4 4 5 3 3 3 3 1 1 4 ...
##  $ OverallScore       : int  1 10 10 5 3 6 7 1 2 9 ...
##  $ Recommended        : Factor w/ 2 levels "yes","no": 2 1 1 2 2 1 1 2 2 1 ...
##  $ ReviewDate         : Factor w/ 1679 levels "10th April 2011",..: 1253 1150 1042 990 990 773 657 2
##  $ ReviewTitle        : Factor w/ 2764 levels " \"crew extremely polite and helpful\"",..: 2645 2687
##  $ ReviewrCountry     : Factor w/ 48 levels " (Argentina) ",..: 45 45 45 45 45 45 45 45 45 21 ...
##  $ Route              : Factor w/ 2012 levels " Frankfurt to Tokyo",..: 746 1696 54 1413 1679 1676 1
##  $ SeatComfortRating  : int  3 4 5 3 4 2 4 1 1 4 ...
##  $ SeatType           : Factor w/ 4 levels "Business Class",..: 2 2 2 2 2 2 2 4 2 2 ...
##  $ ServiceRating      : int  1 5 5 2 1 2 4 1 2 5 ...
##  $ TravelType         : Factor w/ 4 levels "Business","Couple Leisure",..: 1 3 3 2 2 2 4 1 2 4 ...
##  $ ValueRating        : int  1 5 5 3 2 2 3 1 1 5 ...
```

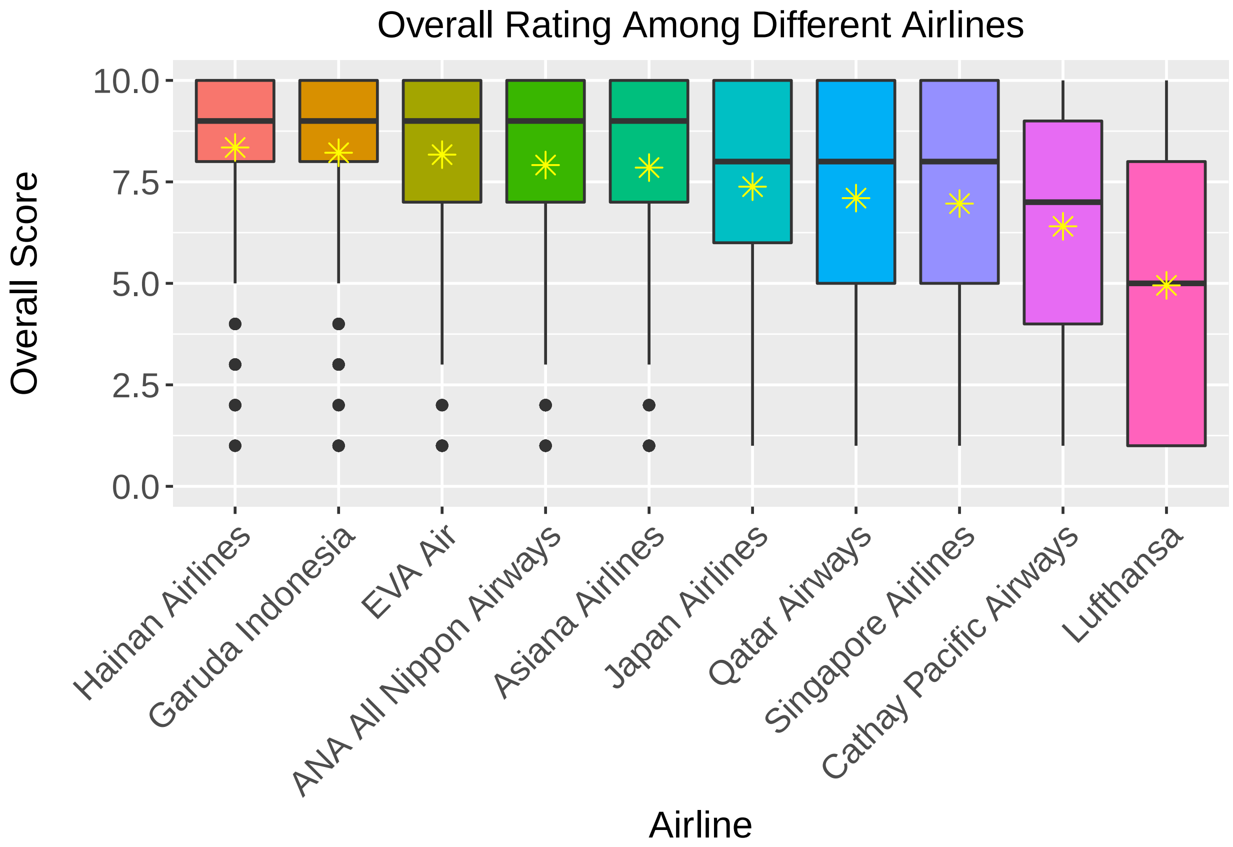
```r
# separate response from all predictors in the data
OverallScore <- airlinesCleaned$OverallScore
indx_overallScore <- which(names(airlinesCleaned) == "OverallScore") # locate response column
airlinesCleaned <- airlinesCleaned[, -indx_overallScore]
```

No variables were removed at this step besides Comments and WiFi Rating in previous steps.

Graphs (Descriptive Stats/Exploratory Analysis)!

Mean Overall Scores Among Different Airlines

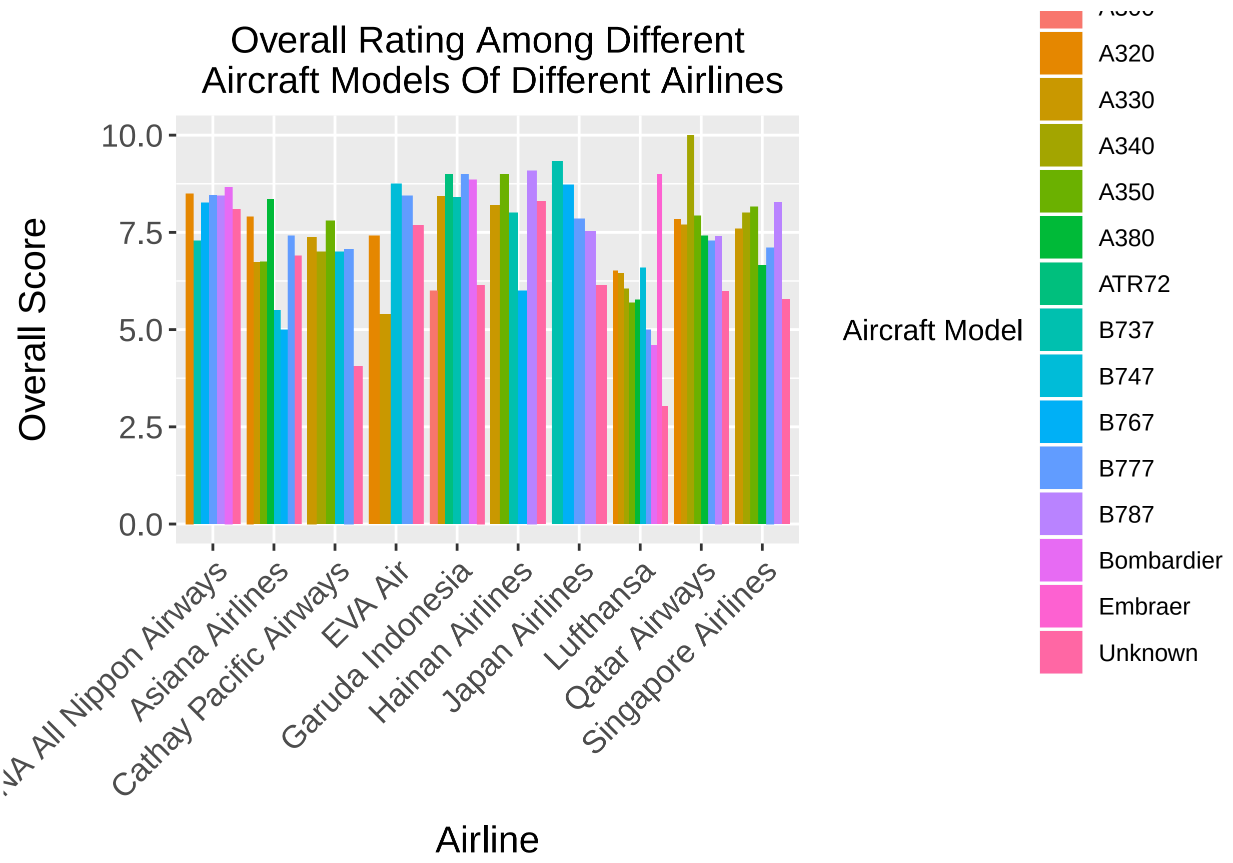
```r
meanOverallScores <- airlineData %>%
  filter(!is.na(OverallScore)) %>%
  group_by(AirName) %>%
  summarize(meanScore = mean(OverallScore))


airlineData %>%
  group_by(AirName) %>%
  filter(!is.na(OverallScore)) %>%
  ggplot() +
  geom_boxplot(aes(x = reorder(AirName, -OverallScore), y = OverallScore, fill = reorder(AirName, -Overa
  geom_point(aes(x = AirName, y = meanOverallScores$meanScore), data = meanOverallScores, color = "yell
  coord_cartesian(ylim = c(0,10)) +
  xlab("Airline") +
  ylab("Overall Score\n") +
  labs(title = "Overall Rating Among Different Airlines") +
  theme(legend.position = "none") +
  #guides(fill = guide_legend(title = "Airline", title.position = "top")) +
  theme(axis.text = element_text(hjust = 1, size = 13)) +
  theme(axis.title = element_text(size = 14)) +
  theme(axis.text.x = element_text(angle = 45)) +
  theme(plot.title = element_text(hjust = 0.5, size = 14))
```



Overall, the top contenders for highest average overall score are Hainan Airlines (8.35), Garuda Indonesia (8.22), followed by EVA Air (8.16) and ANA All Nippon Airways (7.91). Lufthansa has consisently relatively lower ratings, with an average overall score of 4.95.

Overall Scores By Aircraft Model Among Different Airlines

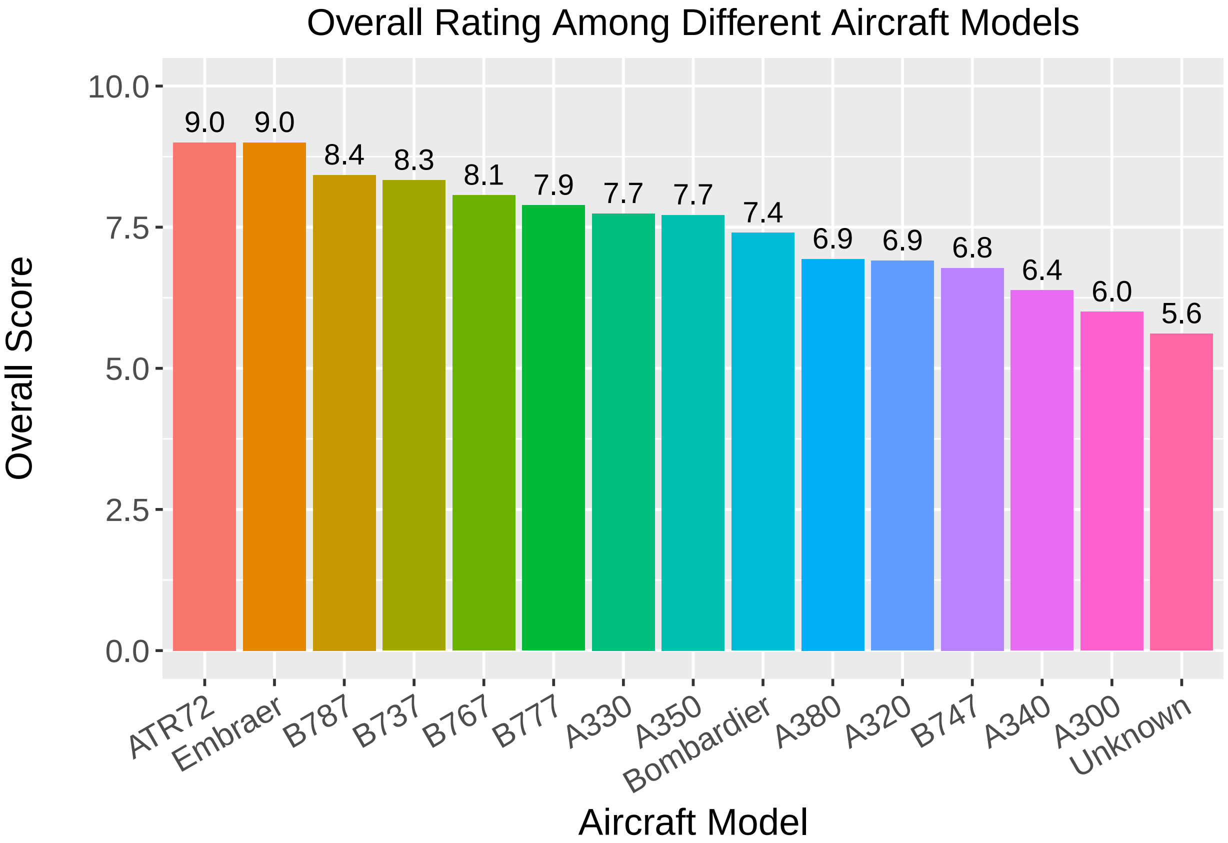
```
meanOverallScoresByModelAndAirname <-
  no_chars %>%
  group_by(AirName, AircraftModel) %>%
  filter(!is.na(AircraftModel)) %>%
  summarize(meanOverallScore = mean(OverallScore))


no_chars %>%
  group_by(AirName) %>%
  #group_by(AircraftModel) %>%
  ggplot() +
  geom_bar(aes(x = AirName, y = meanOverallScore, group = factor(AircraftModel), fill = AircraftModel),
  xlab("Airline") +
  ylab("Overall Score\n") +
  labs(title = "Overall Rating Among Different\n Aircraft Models Of Different Airlines") +
  guides(fill = guide_legend(title = "Aircraft Model", title.position = "left")) +
  theme(axis.text = element_text(hjust = 1, size = 12)) +
  theme(axis.title = element_text(size = 14)) +
  theme(axis.text.x = element_text(angle = 45)) +
  theme(plot.title = element_text(hjust = 0.5, size = 14))
```



According to this dense graph, Lufthansa tends to have the lowest rated aircraft models. Qatar Airways, on the other hand, appears to have the best average overall rating among its different aircraft models. But what about the overall rating of different aircraft models in general?

Overall Scores Of Different Aircraft Models

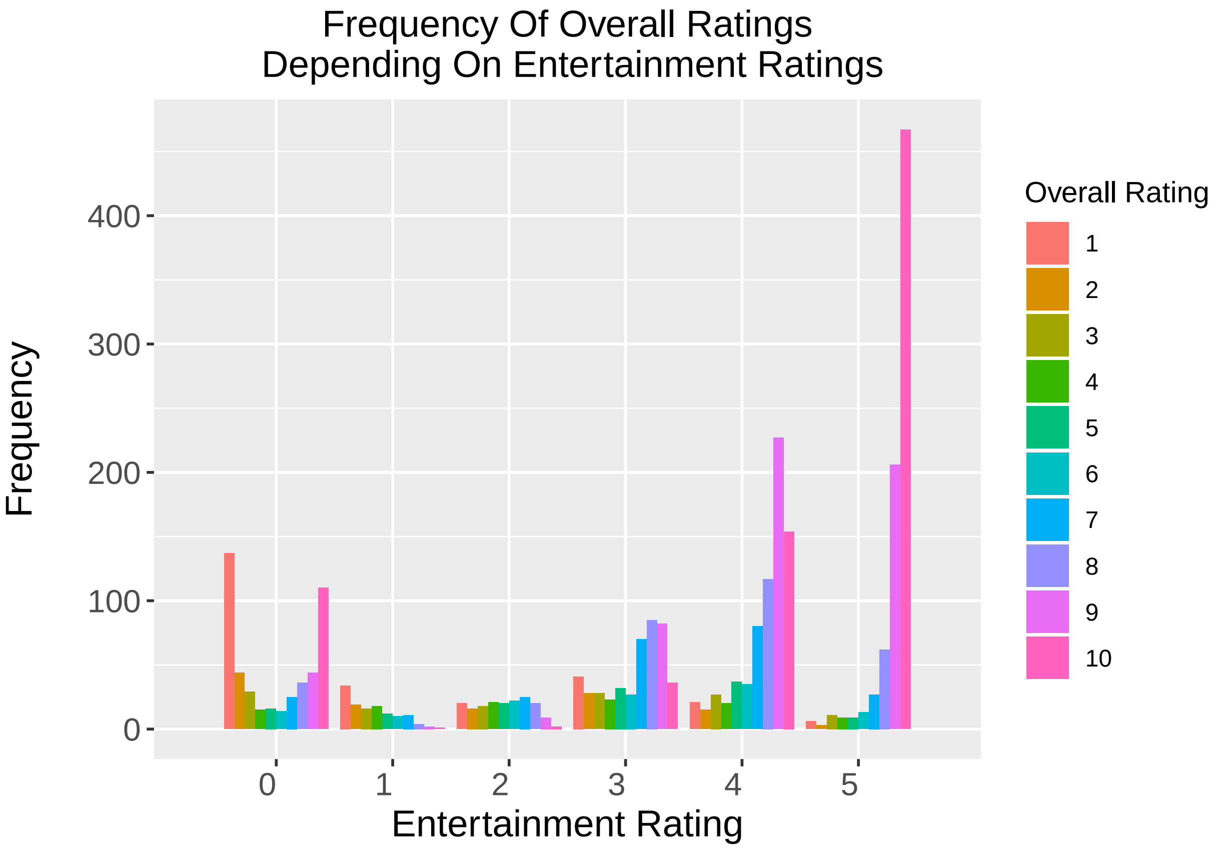
```
meanOverallScoresByModel <-
  no_chars %>%
  group_by(AircraftModel) %>%
  filter(!is.na(AircraftModel)) %>%
  summarize(meanOverallScore = mean(OverallScore))

no_chars %>%
  group_by(AircraftModel) %>%
  #group_by(AircraftModel) %>%
  ggplot() +
  geom_bar(aes(x = reorder(AircraftModel, -meanOverallScore), y = meanOverallScore, fill = reorder(Airc
  geom_text(aes(x = AircraftModel, y = meanOverallScore, label = sprintf("%0.1f", round(meanOverallScor
  coord_cartesian(ylim = c(0,10)) +
  xlab("Aircraft Model") +
  ylab("Overall Score\n") +
  labs(title = "Overall Rating Among Different Aircraft Models") +
  theme(legend.position = "none") +
  #guides(fill = guide_legend(title = "Aircraft Model", title.position = "left")) +
  theme(axis.text = element_text(hjust = 1, size = 12)) +
  theme(axis.title = element_text(size = 14)) +
  theme(axis.text.x = element_text(angle = 30)) +
  theme(plot.title = element_text(hjust = 0.5, size = 14))
```



Both aircraft models ATR72 and Embraer have a highest overall score of 9.0 rating while A300 has the lowest overall rating of 6.0. However, most of the aircraft models are around 7.0 and above.

Frequency Of Overall Ratings Depending On Entertainment Ratings

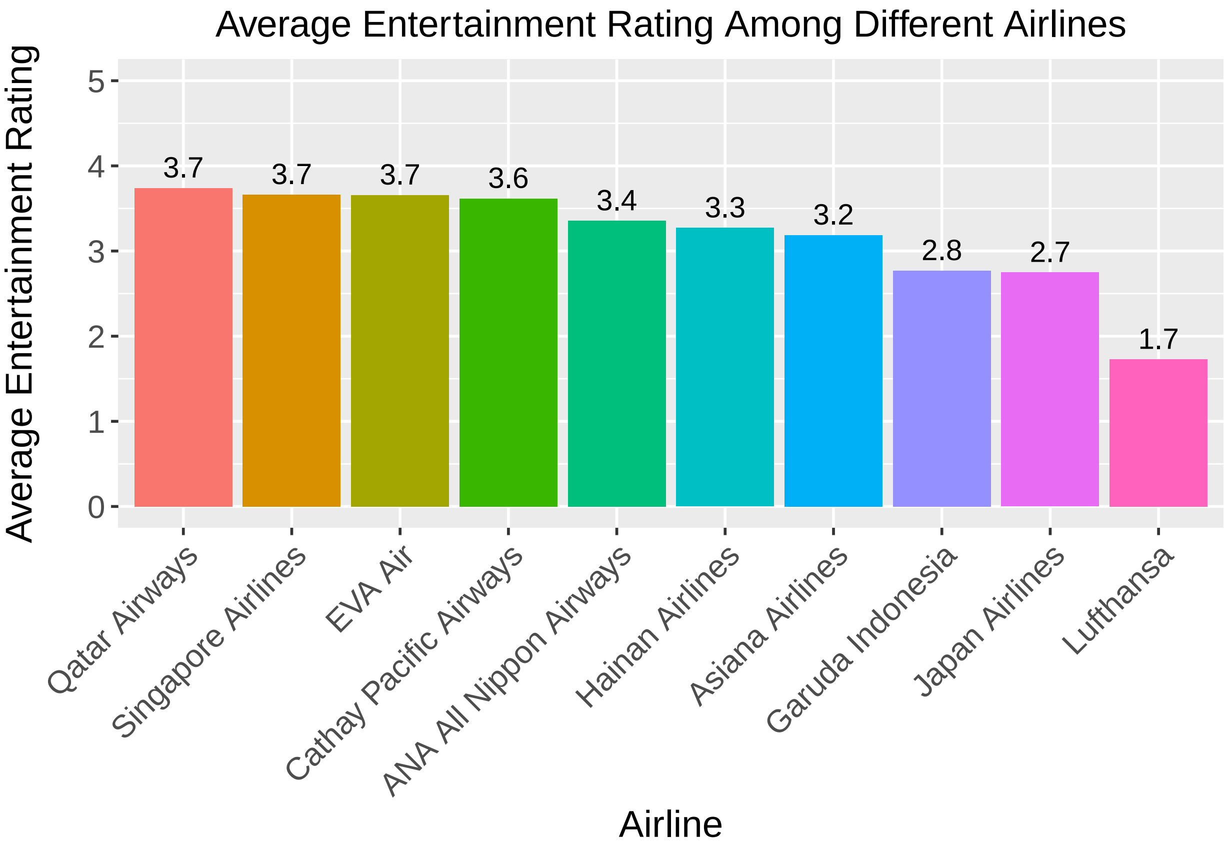
```
no_chars %>%
  group_by(EntertainmentRating, OverallScore) %>%
ggplot() +
  geom_bar(aes(x = EntertainmentRating, group = factor(OverallScore), fill = as.factor(OverallScore)),
  scale_x_discrete(limits = c(0, 1, 2, 3, 4, 5), breaks = c(0, 1, 2, 3, 4, 5)) +
  #scale_color_gradientn(colors = rainbow(10)) +
  #scale_fill_hue() +
  xlab("Entertainment Rating") +
  ylab("Frequency\n") +
  labs(title = "Frequency Of Overall Ratings\n Depending On Entertainment Ratings") +
  guides(fill = guide_legend(title = "Overall Rating", title.position = "top")) +
  theme(axis.text = element_text(hjust = 1, size = 12)) +
  theme(axis.title = element_text(size = 14)) +
  #theme(axis.text.x = element_text(angle = 45)) +
  theme(plot.title = element_text(hjust = 0.5, size = 14))
```



Interestingly, it appears that even when airlines did not provide any sort of entertainment, a little more than 100 passengers still gave the airline a perfect score of 10. This does not mean that an airline does not need to care about its entertainment services. As the entertainment rate increased, so does the number of people who gave high overall ratings.

Average Entertainment Ratings Among Different Airlines

```
airlineData %>%
  group_by(AirName) %>%
  filter(!is.na(EntertainmentRating)) %>%
  #group_by(SeatType) %>%
  summarize(meanEntertainmentRating = mean(EntertainmentRating)) %>%
  ggplot() +
  geom_bar(aes(x = reorder(AirName, -meanEntertainmentRating), y = meanEntertainmentRating, fill = reor
  geom_text(aes(x = AirName, y = meanEntertainmentRating, label = sprintf("%0.1f", round(meanEntertainm
  coord_cartesian(ylim = c(0,5)) +
  xlab("Airline") +
  ylab("Average Entertainment Rating\n") +
  theme(legend.position = "none") +
  #guides(fill = guide_legend(title = "Airline Name", title.position = "top")) +
  labs(title = "Average Entertainment Rating Among Different Airlines") +
  #guides(fill = guide_legend(title = "Airline", title.position = "top")) +
  theme(axis.text = element_text(hjust = 1, size = 12)) +
  theme(axis.text.x = element_text(size = 12, angle = 45)) +
  theme(axis.title = element_text(size = 14)) +
  theme(plot.title = element_text(hjust = 0.5, size = 14)) +
  theme(legend.title = element_text(hjust = 0.5, size = 13))
```

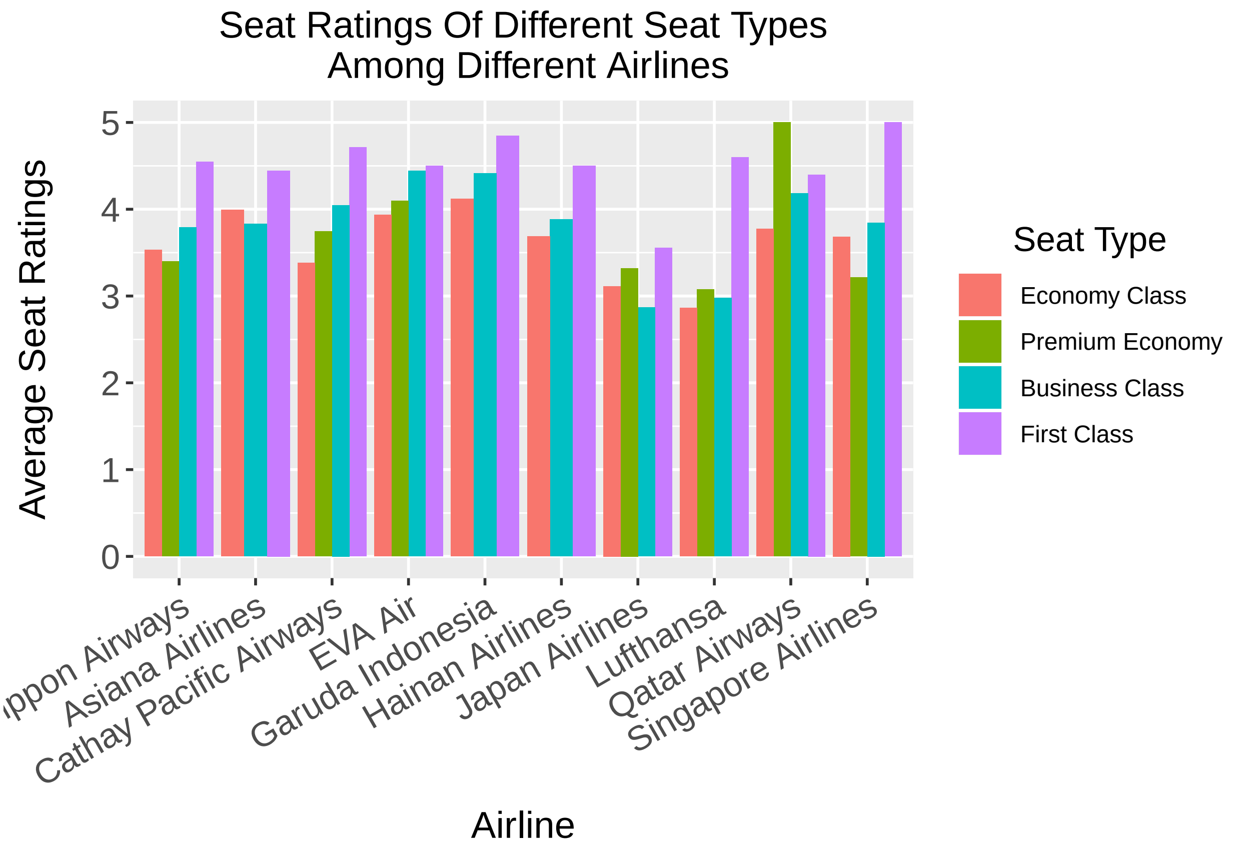


Qatar Airways, Singapore Airlines, and EVA Air both tie for highest average entertainment quality of 3.7, with Cathay Pacific Airways coming in a close second with an average entertainment rating of 3.6. Lufthansa has the lowest average entertainment, as low as 1.7.

Average Seat Comfort Ratings Per Seat Type Among Different Airlines

```
airlineNames <- airlineData$AirName # Major Group
seatType <- airlineData$SeatType# SubGroup
seatRating <- airlineData$SeatComfortRating

seatsData <- data.frame(airlineNames, seatType, seatRating)

meanSeatRatings <- seatsData %>%
  filter(!is.na(seatType)) %>%
  mutate(seatType = fct_relevel(seatType, "Economy Class", "Premium Economy", "Business Class", "First
  group_by(airlineNames, seatType) %>%
  summarize(meanSeatRating = mean(seatRating))

seatsData %>%
    group_by(airlineNames) %>%
    mutate(seatType = fct_relevel(seatType, "Economy Class", "Premium Economy", "Business Class", "Firs
    group_by(seatType) %>%
    filter(!is.na(seatType)) %>%
    #summarize(meanSeatRating = mean(seatRating), group = factor(seatType)) %>%
  ggplot() +
  geom_bar(aes(x = airlineNames, y = meanSeatRating, fill = seatType), data = meanSeatRatings, stat = "
  #geom_hline(aes(x = airlineNames, yintercept = mean(seatRating), color = "magenta"), size = 7) +
   xlab("Airline") +
  ylab("Average Seat Ratings\n") +
  guides(fill = guide_legend(title = "Airline Name", title.position = "top")) +
  labs(title = "Seat Ratings Of Different Seat Types\n Among Different Airlines") +
  guides(fill = guide_legend(title = "Seat Type", title.position = "top")) +
  theme(axis.text = element_text(hjust = 1, size = 13)) +
  theme(axis.text.x = element_text(size = 13, angle = 30)) +
  theme(axis.title = element_text(size = 14)) +
  theme(plot.title = element_text(hjust = 0.5, size = 14)) +
  theme(legend.title = element_text(hjust = 0.5, size = 13))
```

```
## Warning: Factor `seatType` contains implicit NA, consider using
## `forcats::fct_explicit_na`
```

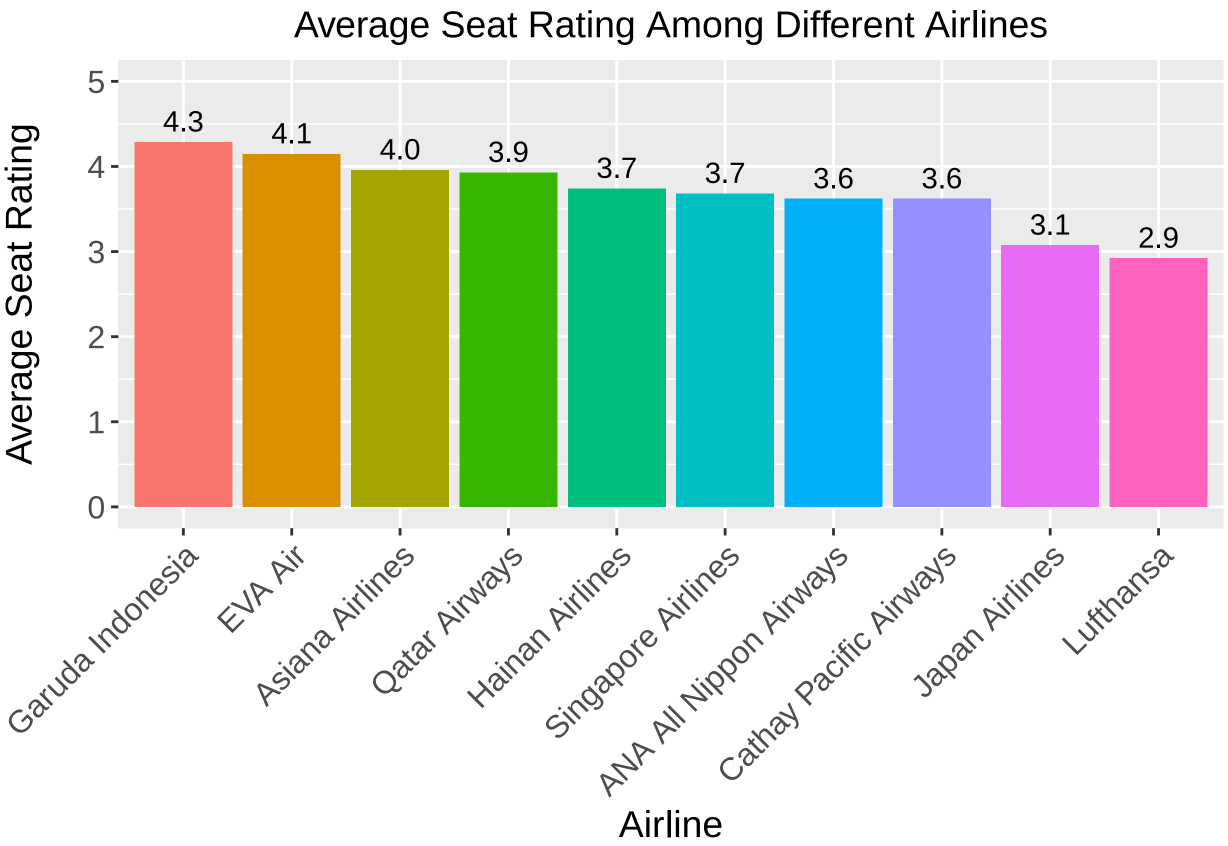


Japan Airlines tend to have the lowest average comfort rating for all their seats. Lufthansa has the same fate; its saving grace is its First Class seats, which ranked is relatively comfortable. Interestingly, Qatar Airways's Premium Economy seats ranked significantly higher than all of the other types of seats (even First Class seats) in Qator Airways. But what about the average seat ratings in general for every airline?

Average Seat Comfort Ratings Among Different Airlines

```
airlineData %>%
  group_by(AirName) %>%
  filter(!is.na(SeatType)) %>%
  #group_by(SeatType) %>%
  summarize(meanSeatComfortRating = mean(SeatComfortRating)) %>%
  ggplot() +
  geom_bar(aes(x = reorder(AirName, -meanSeatComfortRating), y = meanSeatComfortRating, fill = reorder(A
  geom_text(aes(x = AirName, y = meanSeatComfortRating, label = sprintf("%0.1f", round(meanSeatComfortRa
  coord_cartesian(ylim = c(0,5)) +
  xlab("Airline") +
  ylab("Average Seat Rating\n") +
  theme(legend.position = "none") +
  #guides(fill = guide_legend(title = "Airline Name", title.position = "top")) +
  labs(title = "Average Seat Rating Among Different Airlines") +
  guides(fill = guide_legend(title = "Airline", title.position = "top")) +
  theme(axis.text = element_text(hjust = 1, size = 12)) +
  theme(axis.text.x = element_text(size = 12, angle = 45)) +
  theme(axis.title = element_text(size = 14)) +
  theme(plot.title = element_text(hjust = 0.5, size = 14)) +
  theme(legend.title = element_text(hjust = 0.5, size = 13))
```

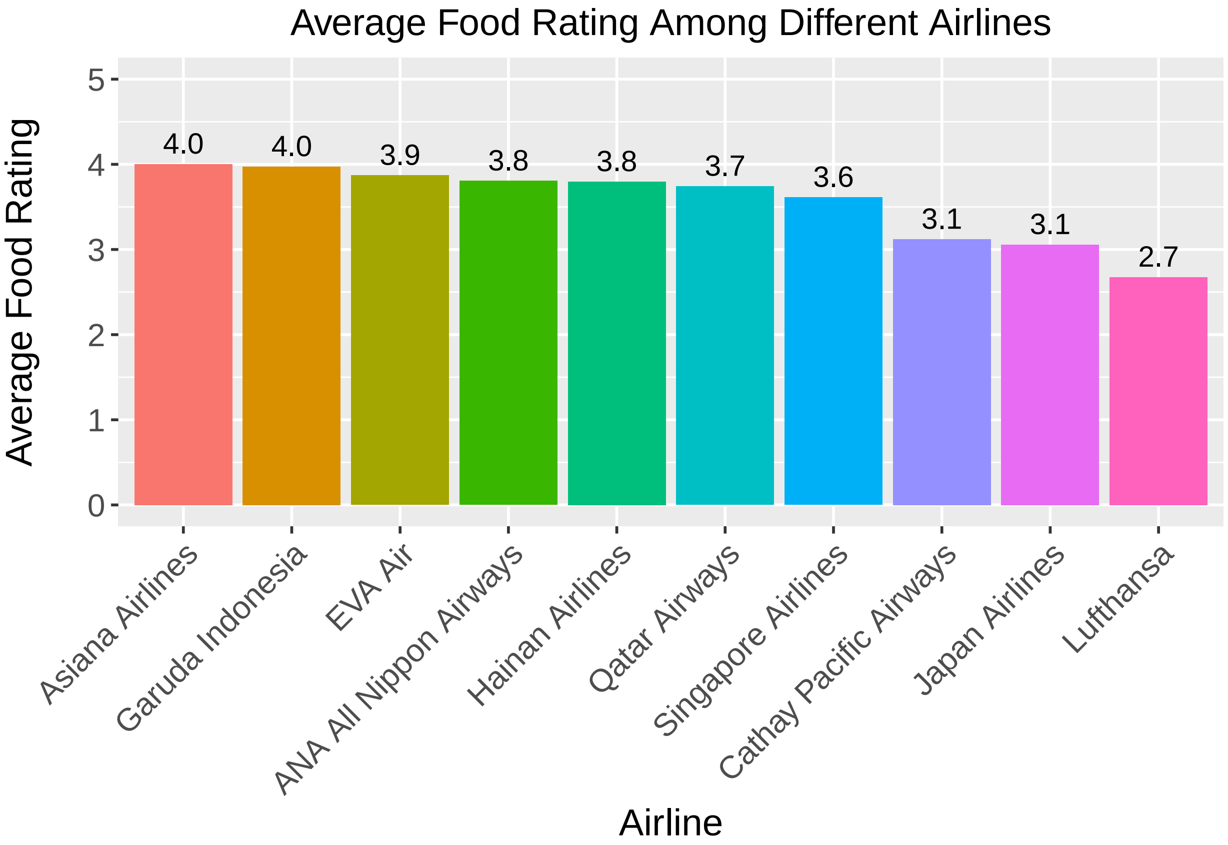
```
## Warning: Ignoring unknown parameters: ymax
```



Garuda Indonesia appears to have the most comfortable seats, with an average rating of 4.2. EVA Air comes in a close second with an average seat comfort rating of 4.1; Qatar Airways and Asiana Airlines come in a very close third place with an average seat rating of 3.9. Both Japan Airlines (average seat rating of 3.0) and Lufthansa (average seat rating of 2.9) rank lowest in the average seat comfort level.

Average Food Ratings Among Airlines

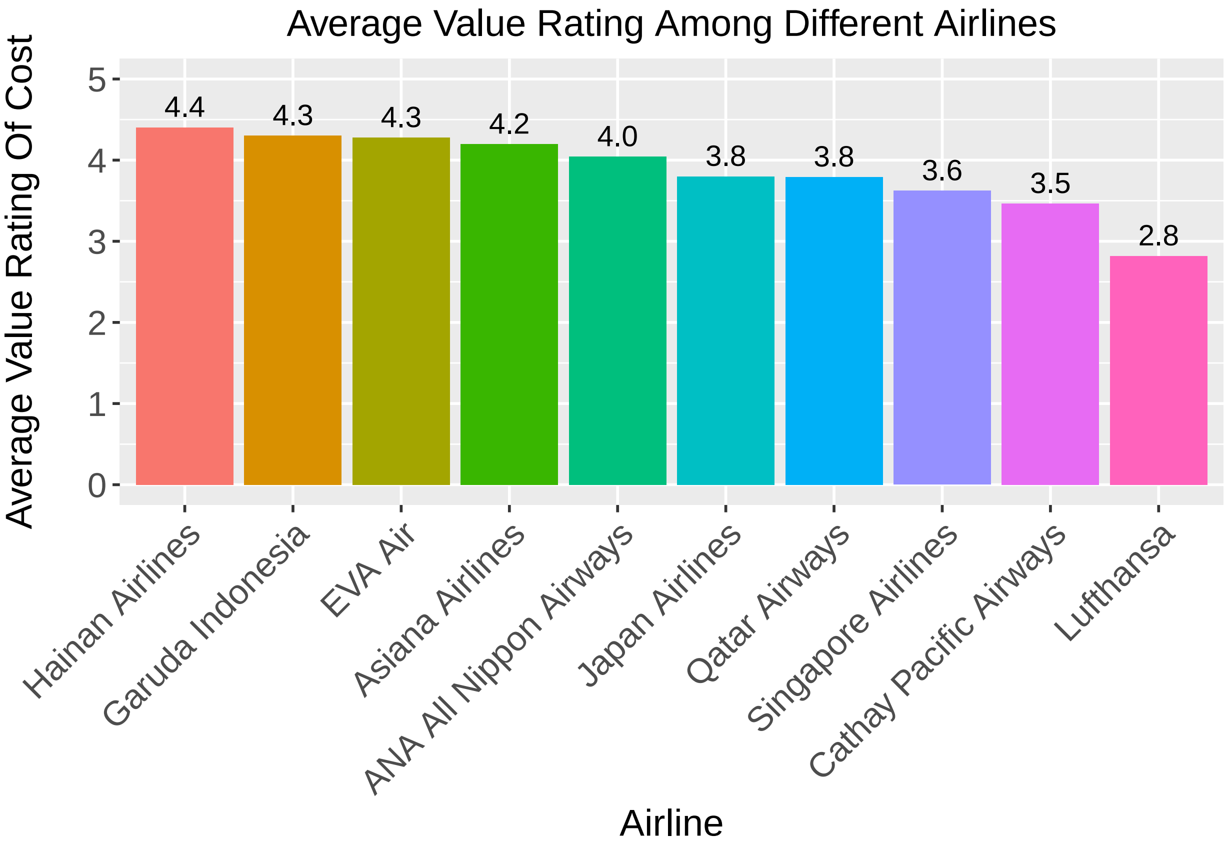
```
airlineData %>%
  group_by(AirName) %>%
  summarize(meanFoodRating = mean(FoodRating)) %>%
  ggplot() +
  geom_bar(aes(x = reorder(AirName, -meanFoodRating), y = meanFoodRating, fill = reorder(AirName, -meanF
  geom_text(aes(x = AirName, y = meanFoodRating, label = sprintf("%0.1f", round(meanFoodRating, digits =
  coord_cartesian(ylim = c(0, 5)) +
  xlab("Airline") +
  ylab("Average Food Rating\n") +
  theme(legend.position = "none") +
  #guides(fill = guide_legend(title = "Airline Name", title.position = "top")) +
  labs(title = "Average Food Rating Among Different Airlines") +
  #guides(fill = guide_legend(title = "Airline", title.position = "top")) +
  theme(axis.text = element_text(hjust = 1, size = 12)) +
  theme(axis.text.x = element_text(size = 12, angle = 45)) +
  theme(axis.title = element_text(size = 14)) +
  theme(plot.title = element_text(hjust = 0.5, size = 14)) +
  theme(legend.title = element_text(hjust = 0.5, size = 13))
```



Asiana Airlines and Garuda Indonesia tie for best average food, with a close second from EVA Air. Lufthansa has the lowest overall food quality of 2.7, unfortunately.

Average Value Ratings Among Different Airlines

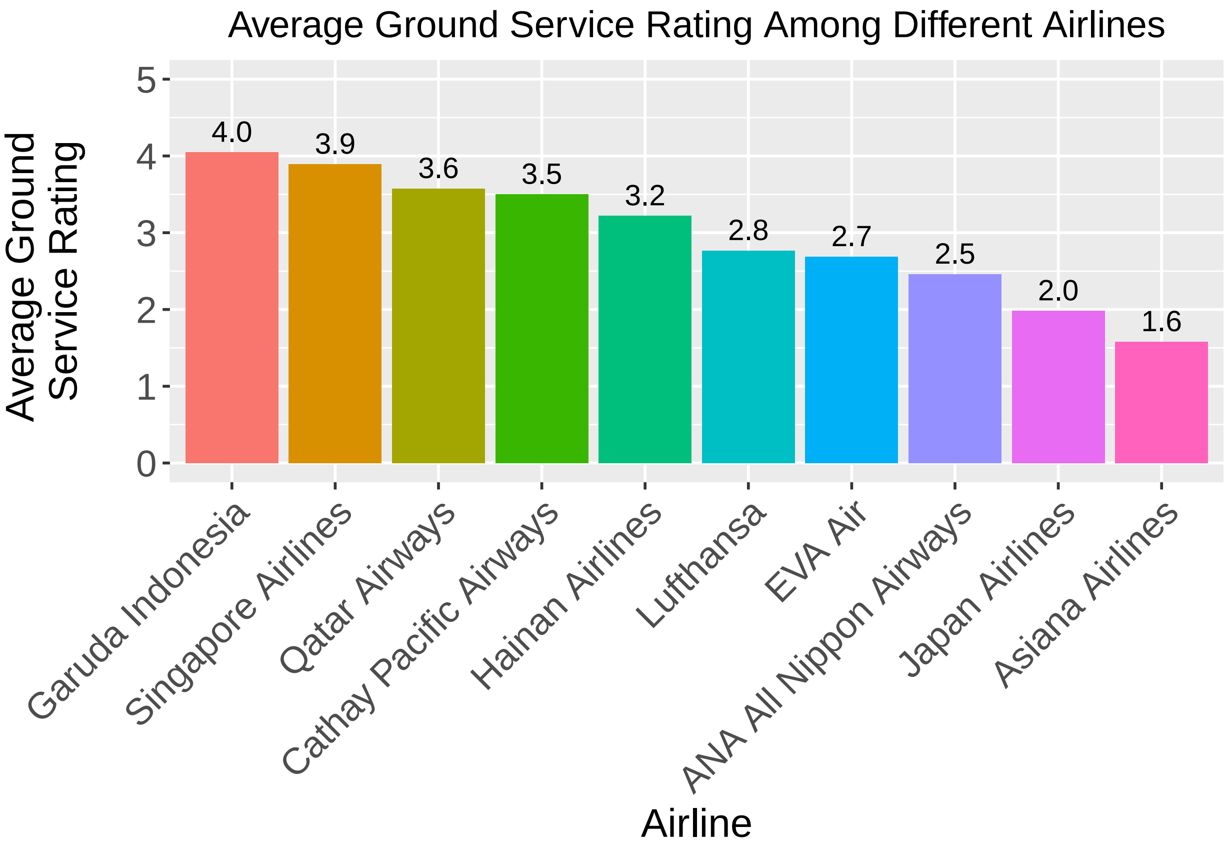
```
airlineData %>%
  group_by(AirName) %>%
  filter(!is.na(ValueRating)) %>%
  summarize(meanValueRating = mean(ValueRating)) %>%
  ggplot() +
  geom_bar(aes(x = reorder(AirName, -meanValueRating), y = meanValueRating, fill = reorder(AirName, -mea
  geom_text(aes(x = AirName, y = meanValueRating, label = sprintf("%0.1f", round(meanValueRating, digits
  coord_cartesian(ylim = c(0,5)) +
  xlab("Airline") +
  ylab("Average Value Rating Of Cost\n") +
  theme(legend.position = "none") +
  #guides(fill = guide_legend(title = "Airline Name", title.position = "top")) +
  labs(title = "Average Value Rating Among Different Airlines") +
  theme(axis.text = element_text(hjust = 1, size = 13)) +
  theme(axis.text.x = element_text(size = 13, angle = 45)) +
  theme(axis.title = element_text(size = 14)) +
  theme(plot.title = element_text(hjust = 0.5, size = 14)) +
  theme(legend.title = element_text(hjust = 0.5, size = 13))
```



Hainan Airlines and Garuda Indonesia are the top two airlines whose overall scores were well-correlated with its ticket price. Lufthansa has the lowest worth-it value for its ticket price with a rating of 2.8. In other words, Lufthansa's ticket price does not indicate the overall quality.

Average Ground Service Ratings Among Different Airlines

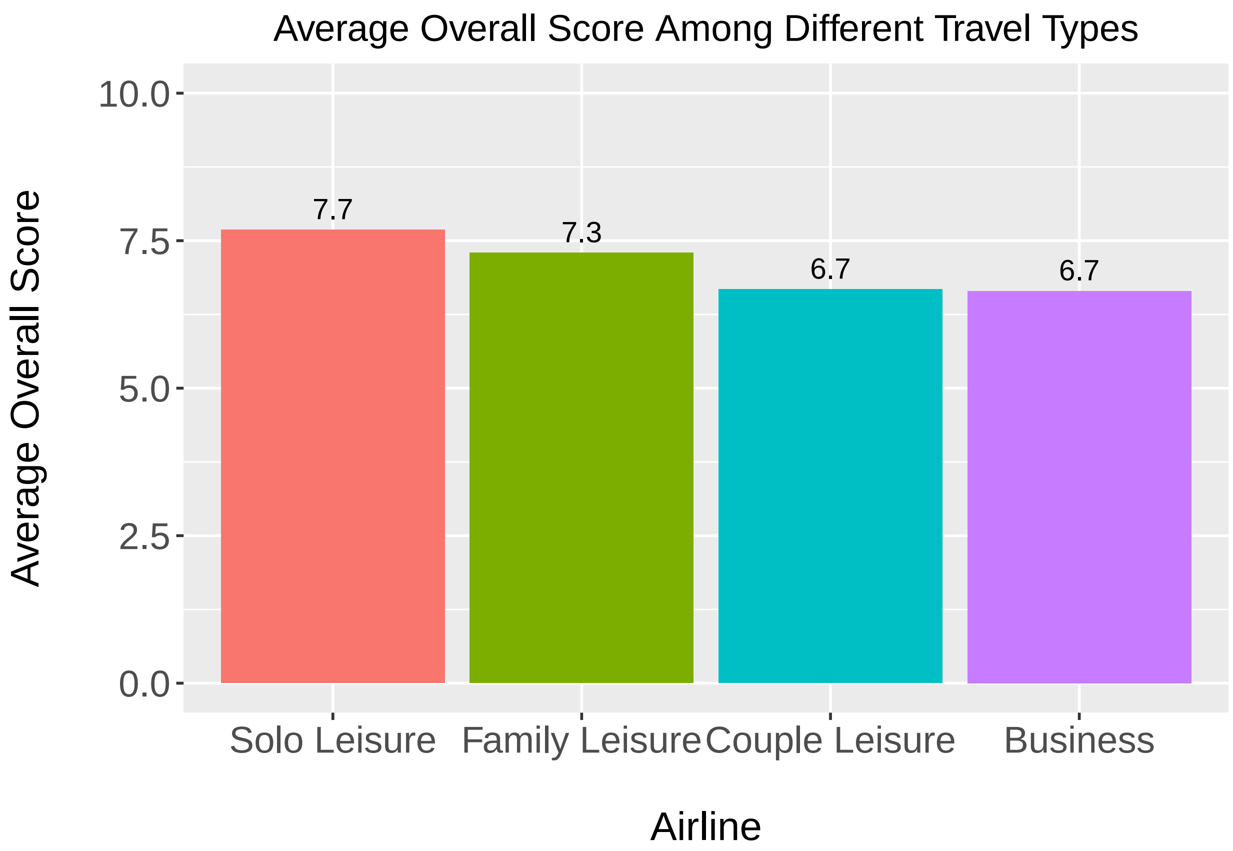
```
airlineData %>%
  group_by(AirName) %>%
  filter(!is.na(GroundServiceRating)) %>%
  summarize(meanGroundServiceRating = mean(GroundServiceRating)) %>%
  ggplot() +
  geom_bar(aes(x = reorder(AirName, -meanGroundServiceRating), y = meanGroundServiceRating, fill = reor
  geom_text(aes(x = AirName, y = meanGroundServiceRating, label = sprintf("%0.1f", round(meanGroundServi
  coord_cartesian(ylim = c(0,5)) +
  xlab("Airline") +
  ylab("Average Ground \nService Rating\n") +
  theme(legend.position = "none") +
  #guides(fill = guide_legend(title = "Airline Name", title.position = "top")) +
  labs(title = "Average Ground Service Rating Among Different Airlines") +
  theme(axis.text = element_text(hjust = 1, size = 14)) +
  theme(axis.text.x = element_text(size = 14, angle = 45)) +
  theme(axis.title = element_text(size = 15)) +
  theme(plot.title = element_text(hjust = 0.5, size = 14)) +
  theme(legend.title = element_text(hjust = 0.5, size = 13))
```



Garuda Indonesia has the highest average ground service rating, with a rating of 4.0. Singapore Airlines comes a close second with an average ground service rating of 3.9. Asiana Airlines has the lowest average ground service rating of 1.6.

Average Overall Scores Among Different Travel Types

```
airlineData %>%
  filter(!is.na(TravelType)) %>%
  group_by(TravelType) %>%
  summarize(meanOverallScore = mean(OverallScore)) %>%
  ggplot() +
  geom_bar(aes(x = reorder(TravelType, -meanOverallScore), y = meanOverallScore, fill = reorder(TravelTy
  geom_text(aes(x = TravelType, y = meanOverallScore, label = sprintf("%0.1f", round(meanOverallScore, 
  coord_cartesian(ylim = c(0,10)) +
  xlab("\nAirline") +
  ylab("Average Overall Score\n") +
  theme(legend.position = "none") +
  #guides(fill = guide_legend(title = "Airline Name", title.position = "top")) +
  labs(title = "Average Overall Score Among Different Travel Types") +
  theme(axis.text = element_text(size = 14)) +
  theme(axis.text.x = element_text(size = 13.5)) +
  theme(axis.title = element_text(size = 15)) +
  theme(plot.title = element_text(hjust = 0.5, size = 14)) +
  theme(legend.title = element_text(hjust = 0.5, size = 13))
```



It appears that passengers who fly solo enjoy the ride more than those who do not fly solo. Family Leisure comes in a close second with an average overall score of 7.3. Surprinslgy, couples flying together placed 3rd in average overall score–it even tied with people who flew for business!

Linear Regression Model

```
# run a linear regression model
# 6 most important predictors for travelers' overall rating of their flight experience
reg <- lm(OverallScore ~ EntertainmentRating + FoodRating + GroundServiceRating + SeatComfortRating
          + ServiceRating + ValueRating, data = airlinesCleaned)
(summary.data <- summary(reg))
```

```
##
## Call:
## lm(formula = OverallScore ~ EntertainmentRating + FoodRating +
##     GroundServiceRating + SeatComfortRating + ServiceRating +
##     ValueRating, data = airlinesCleaned)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.3032 -0.5969  0.0860  0.5666  7.5607
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -1.57988    0.07395 -21.365  < 2e-16 ***
## EntertainmentRating  0.04679    0.01460   3.206  0.00136 **
## FoodRating           0.23059    0.02369   9.732  < 2e-16 ***
## GroundServiceRating  0.43895    0.02167  20.256  < 2e-16 ***
## SeatComfortRating    0.24532    0.02592   9.466  < 2e-16 ***
## ServiceRating        0.33235    0.02730  12.175  < 2e-16 ***
## ValueRating          1.00479    0.02807  35.795  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.133 on 2761 degrees of freedom
## Multiple R-squared:  0.8615, Adjusted R-squared:  0.8612
## F-statistic:  2862 on 6 and 2761 DF,  p-value: < 2.2e-16
```

```r
(significance_table <- as.data.frame(summary.data$coefficients))
```

```
##                        Estimate Std. Error    t value      Pr(>|t|)
## (Intercept)         -1.57988230 0.07394840 -21.364658  7.531250e-94
## EntertainmentRating  0.04678732 0.01459525   3.205654  1.362934e-03
## FoodRating           0.23059381 0.02369368   9.732294  4.941022e-22
## GroundServiceRating  0.43895139 0.02167063  20.255589  3.635278e-85
## SeatComfortRating    0.24531681 0.02591563   9.465980  6.019872e-21
## ServiceRating        0.33235123 0.02729674  12.175491  2.943501e-33
## ValueRating          1.00478508 0.02807081  35.794665 7.554702e-231
```

```r
# assuming 95% confidence, extract variables that are statistically significant at 0.05 significance
# observe p-value
sig_level <- 0.05
# locate variables with significant p-values
(sig_indx <- which(summary.data$coefficients[2:nrow(summary.data$coefficients), 4] < sig_level))
```

```
## EntertainmentRating          FoodRating GroundServiceRating
##                   1                   2                   3
##   SeatComfortRating       ServiceRating         ValueRating
##                   4                   5                   6
```

```r
# surprisingly, all variables are statistically significant!

# recall an estimate is not useful if it is large in value but also high in uncertainty (variance)
# observe t-value
threshold <- 2.5
# locate variables with extreme t-values
(t_indx <- which(summary.data$coefficients[2:(nrow(summary.data$coefficients)), 3] > threshold))
```

```
## EntertainmentRating          FoodRating GroundServiceRating
##                   1                   2                   3
##   SeatComfortRating       ServiceRating         ValueRating
##                   4                   5                   6
```

```r
# relevant variables of an airline passenger's review that are most relevant in determining
# its expected overall score
relevant_vars <- names(t_indx)
relevant_vars_table <- filter(significance_table, significance_table[,4] < sig_level)
```

```
(rel_ordered <- relevant_vars_table[order(-relevant_vars_table$Estimate),])
```

```
##      Estimate Std. Error    t value      Pr(>|t|)
## 7  1.00478508 0.02807081  35.794665 7.554702e-231
## 4  0.43895139 0.02167063  20.255589  3.635278e-85
## 6  0.33235123 0.02729674  12.175491  2.943501e-33
## 5  0.24531681 0.02591563   9.465980  6.019872e-21
## 3  0.23059381 0.02369368   9.732294  4.941022e-22
## 2  0.04678732 0.01459525   3.205654  1.362934e-03
## 1 -1.57988230 0.07394840 -21.364658  7.531250e-94
```

```
# based on list of relevant variables, goal is to find 3 greatest estimates
# since those will likely have greatest impact on overall airline rating
(top_three_pos <- rel_ordered[1:3,]) # value, ground service, in-flight service
```

```
##    Estimate Std. Error  t value      Pr(>|t|)
## 7 1.0047851 0.02807081 35.79467 7.554702e-231
## 4 0.4389514 0.02167063 20.25559  3.635278e-85
## 6 0.3323512 0.02729674 12.17549  2.943501e-33
```

```
(top_three_neg <- rel_ordered[4:6,]) # seat comfort, food, entertainment
```

```
##     Estimate Std. Error  t value     Pr(>|t|)
## 5 0.24531681 0.02591563 9.465980 6.019872e-21
## 3 0.23059381 0.02369368 9.732294 4.941022e-22
## 2 0.04678732 0.01459525 3.205654 1.362934e-03
```

We performed a linear regression analysis with Overall Score (defined as customer satisfaction) as the dependent variable with six independent variables: Entertainment, Food, Ground Service, Seat Comfort, Cabin Service, and Value Ratings. The verall variance explained by the six predictors was 86.15% (R^2 = 0.8615) and all factors appeared to be statistically significant at $p < 0.001$ level except Entertainment factor. Among the five factors that are scientifically found to be positively related to customer satisfaction, the Value factor holds the highest standardized coefficients, which means this flight experience aspect of passengers is most important factor associated with customer satisfaction. Our hypothesis that "Entertainment positively influences customer satisfaction" was rejected. Interestingly, many review titles consisting the word `value` were also observed to have no entertainment provided in contrast to other rating columns.

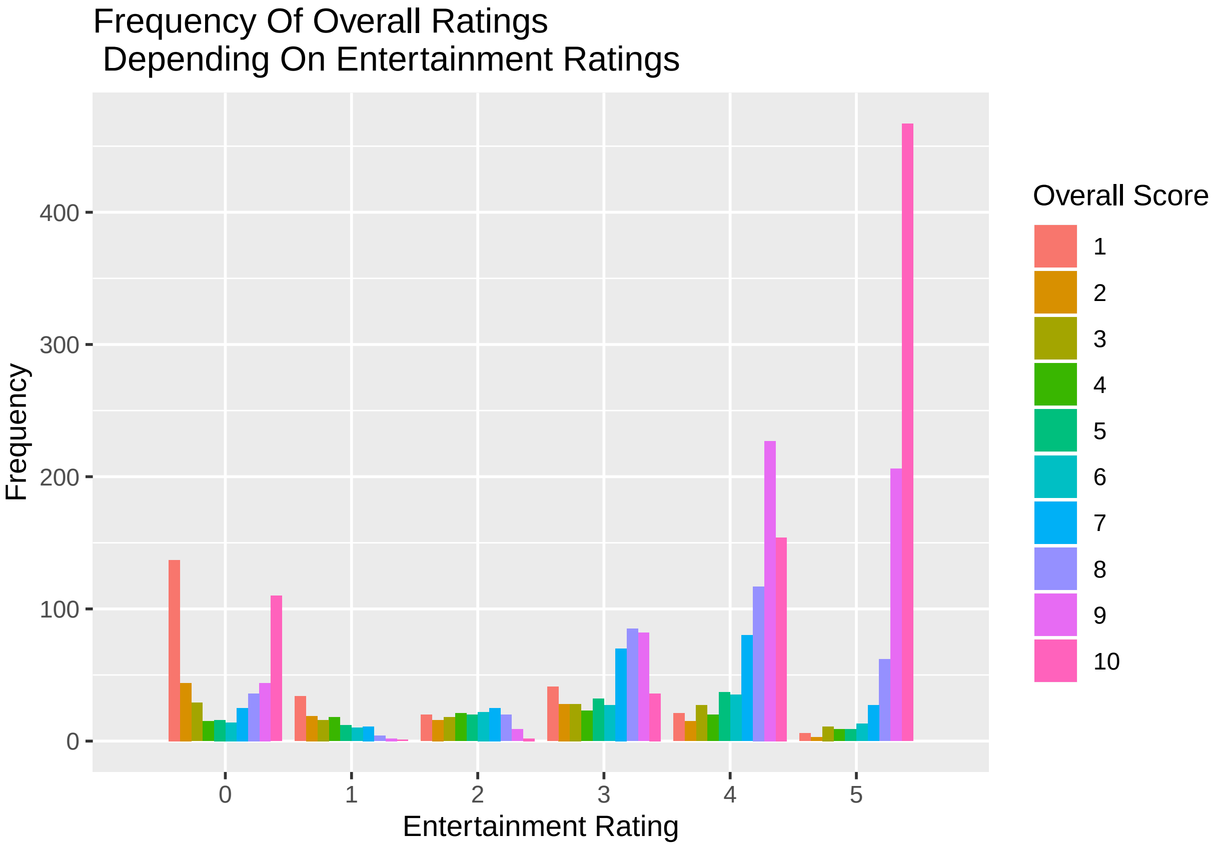
Checking sample size per airline

```
ana <- filter(airlinesCleaned, AirName == "ANA All Nippon Airways")   # 205 obs.
asiana <- filter(airlinesCleaned, AirName == "Asiana Airlines")       # 142  obs.
cathay <- filter(airlinesCleaned, AirName == "Cathay Pacific Airways") # 347 obs.
eva <- filter(airlinesCleaned, AirName == "EVA Air")                  # 234 obs.
garuda <- filter(airlinesCleaned, AirName == "Garuda Indonesia")      # 378 obs.
hainan <- filter(airlinesCleaned, AirName == "Hainan Airlines")       # 274 obs.
jal <- filter(airlinesCleaned, AirName == "Japan Airlines")           # 156 obs.
luf <- filter(airlinesCleaned, AirName == "Lufthansa")                # 356  obs.
qatar <- filter(airlinesCleaned, AirName == "Qatar Airways")        # 313  obs.
sia <- filter(airlinesCleaned, AirName == "Singapore Airlines")     # 363 obs.
```

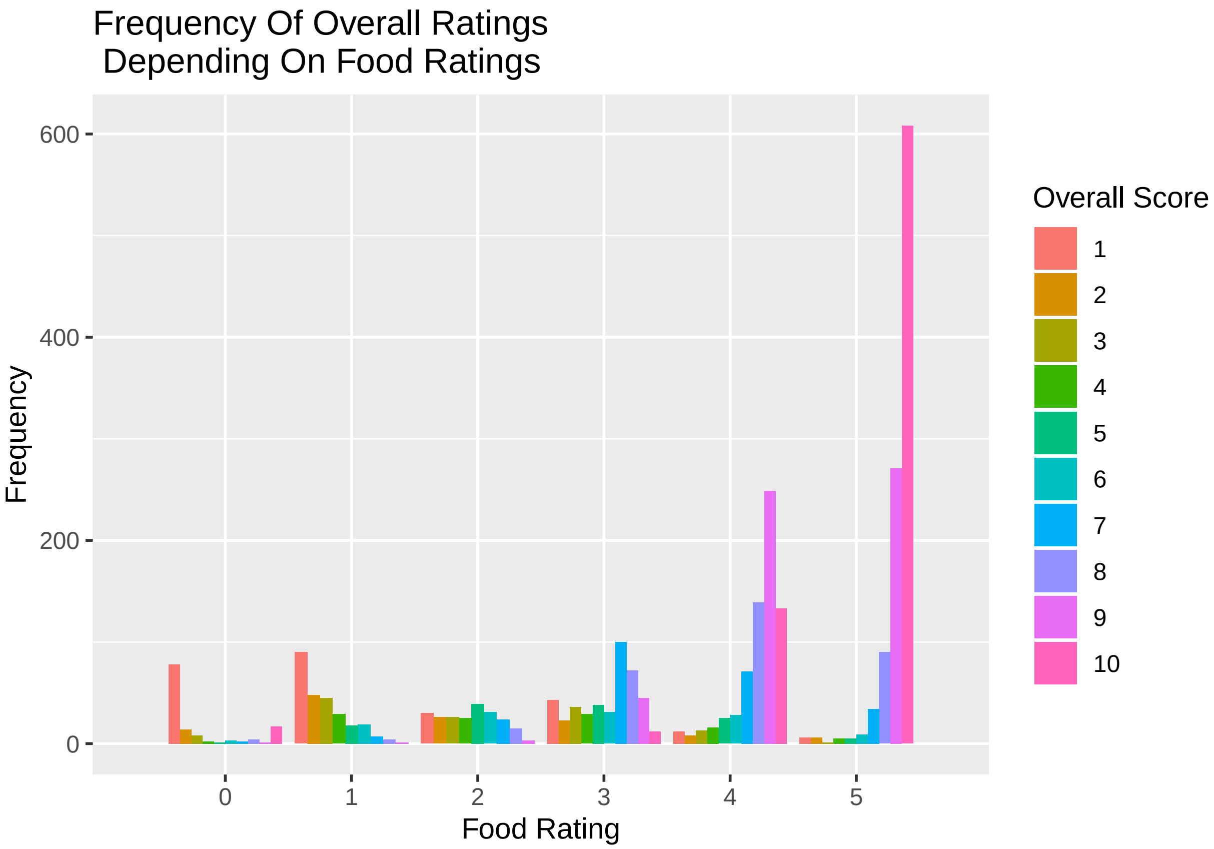
Fortunately, data does not seem to be skewed due to slight differences in sample size per airline.

Weighted bar plots of each predictor against the response

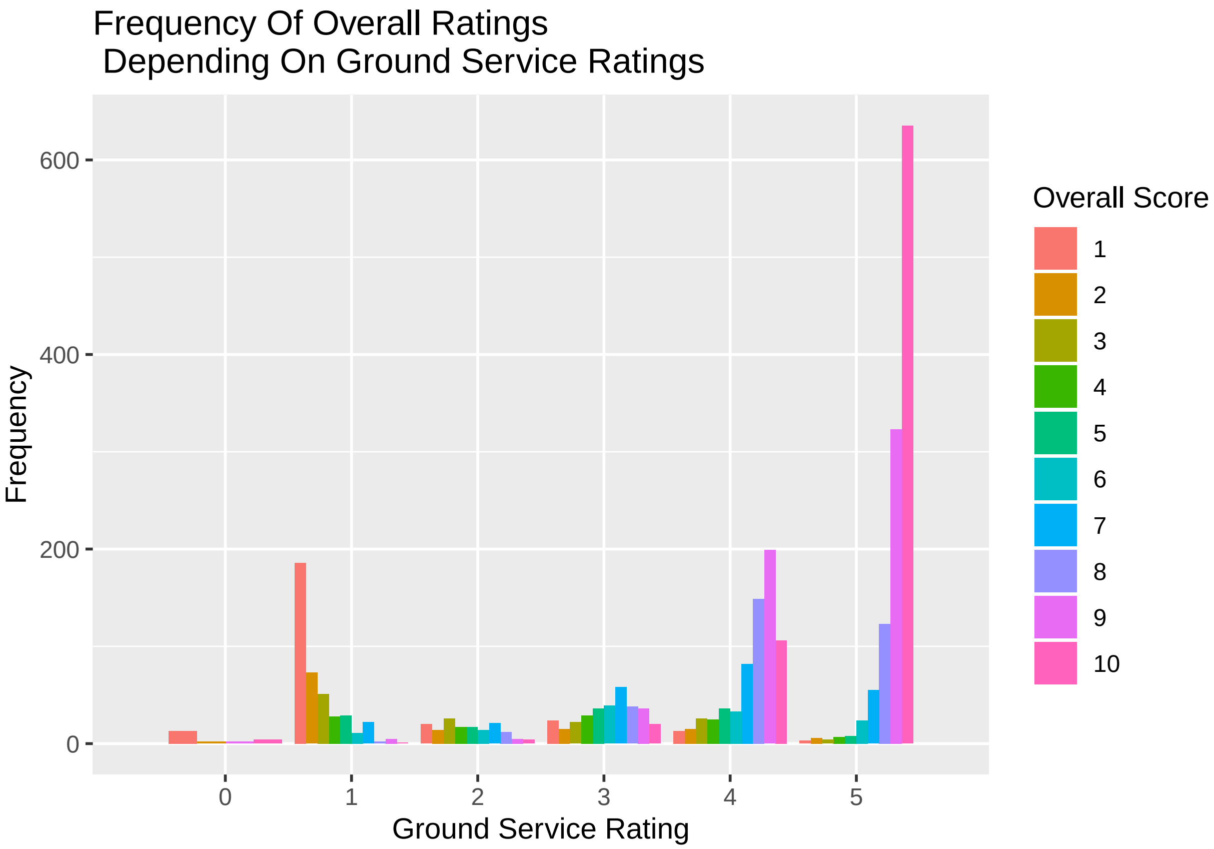
```r
# plots to better understand the relationship between each predictor and response
# ggplot equivalent of par(mfrow)
# require(gridExtra)

ggplot(no_chars) + geom_bar(aes(x = EntertainmentRating, group = factor(OverallScore),
                                fill = as.factor(OverallScore)), position = "dodge") +
    scale_x_discrete(limits = c(0, 1, 2, 3, 4, 5), breaks = c(0, 1, 2, 3, 4, 5)) +
  guides(fill = guide_legend(title = "Overall Score", title.position = "top")) +
  xlab("Entertainment Rating") + ylab("Frequency") +
  labs(title = "Frequency Of Overall Ratings\n Depending On Entertainment Ratings")
```



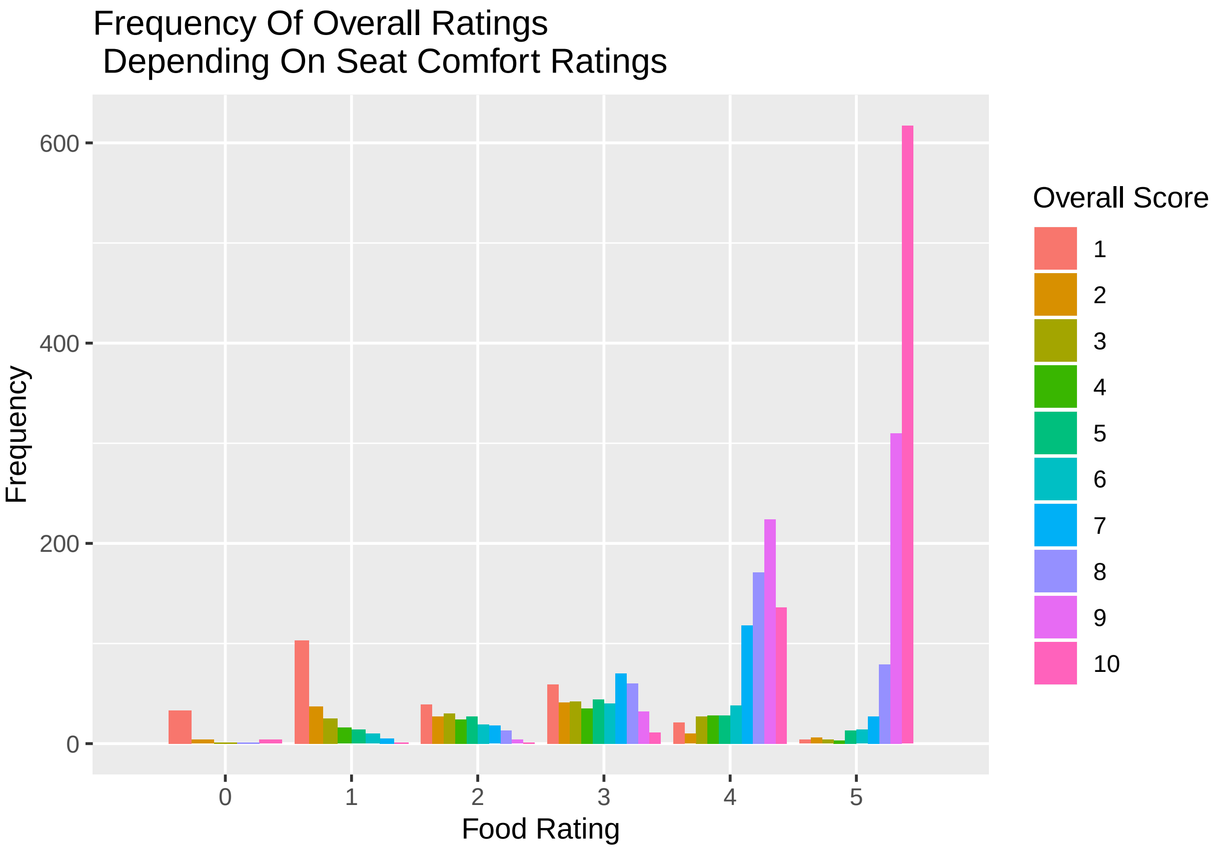
```r
ggplot(no_chars) + geom_bar(aes(x = FoodRating, group = factor(OverallScore),
                                fill = as.factor(OverallScore)), position = "dodge") +
  scale_x_discrete(limits = c(0, 1, 2, 3, 4, 5), breaks = c(0, 1, 2, 3, 4, 5)) +
  guides(fill = guide_legend(title = "Overall Score", title.position = "top")) +
  xlab("Food Rating") + ylab("Frequency") +
  labs(title = "Frequency Of Overall Ratings\n Depending On Food Ratings")
```



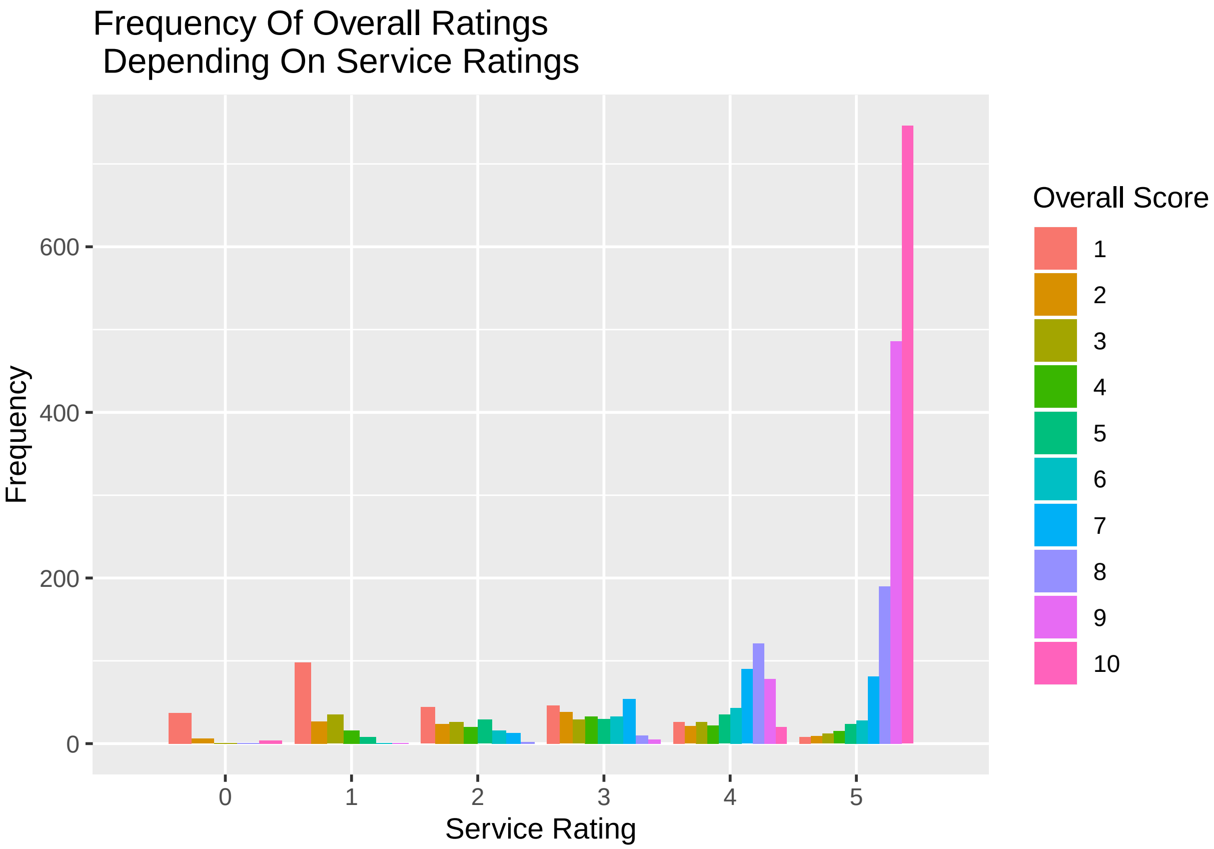
```r
ggplot(no_chars) + geom_bar(aes(x = GroundServiceRating, group = factor(OverallScore),
                               fill = as.factor(OverallScore)), position = "dodge") +
  scale_x_discrete(limits = c(0, 1, 2, 3, 4, 5), breaks = c(0, 1, 2, 3, 4, 5)) +
  guides(fill = guide_legend(title = "Overall Score", title.position = "top")) +
  xlab("Ground Service Rating") + ylab("Frequency") +
  labs(title = "Frequency Of Overall Ratings\n Depending On Ground Service Ratings")
```



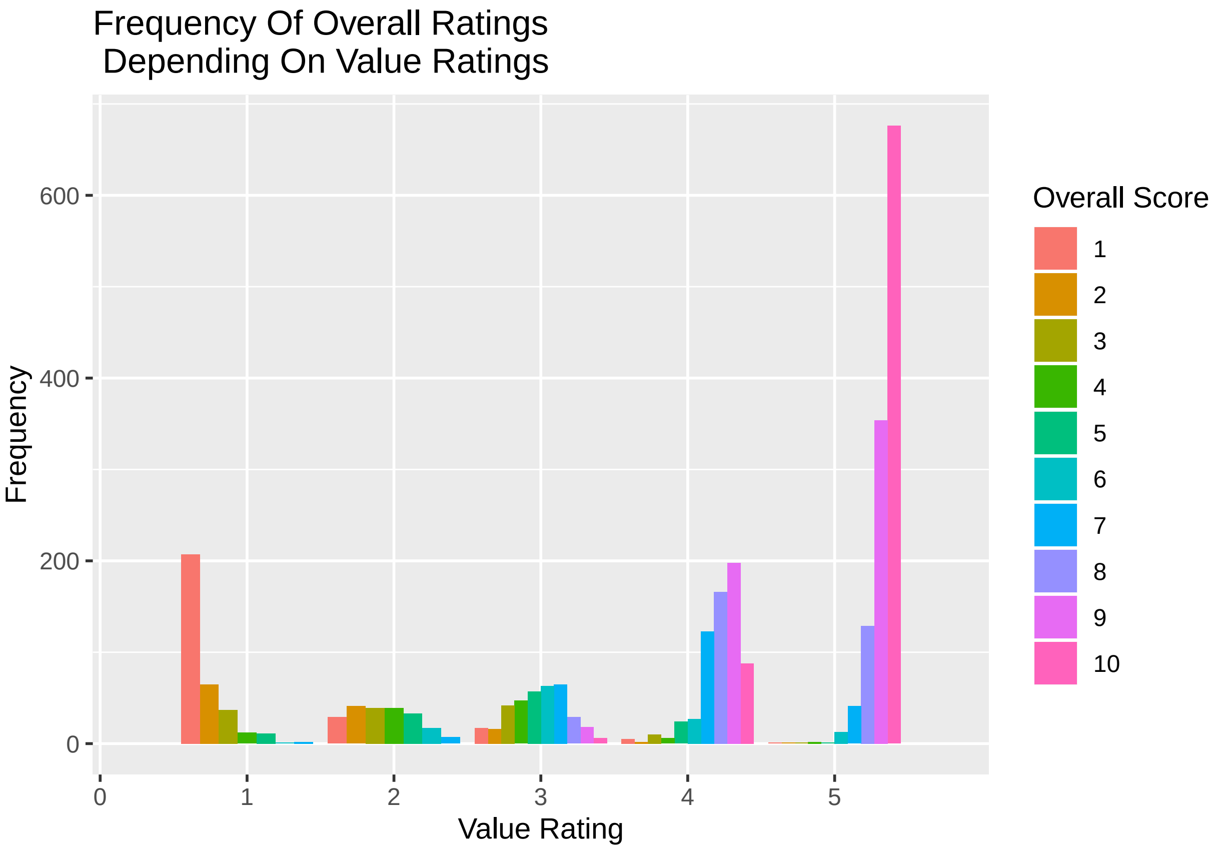
```
ggplot(no_chars) + geom_bar(aes(x = SeatComfortRating, group = factor(OverallScore),
                                fill = as.factor(OverallScore)), position = "dodge") +
  scale_x_discrete(limits = c(0, 1, 2, 3, 4, 5), breaks = c(0, 1, 2, 3, 4, 5)) +
  guides(fill = guide_legend(title = "Overall Score", title.position = "top")) +
  xlab("Food Rating") + ylab("Frequency") +
  labs(title = "Frequency Of Overall Ratings\n Depending On Seat Comfort Ratings")
```



```r
ggplot(no_chars) + geom_bar(aes(x = ServiceRating, group = factor(OverallScore),
                                fill = as.factor(OverallScore)), position = "dodge") +
  scale_x_discrete(limits = c(0, 1, 2, 3, 4, 5), breaks = c(0, 1, 2, 3, 4, 5)) +
  guides(fill = guide_legend(title = "Overall Score", title.position = "top")) +
  xlab("Service Rating") + ylab("Frequency") +
  labs(title = "Frequency Of Overall Ratings\n Depending On Service Ratings")
```



```
ggplot(no_chars) + geom_bar(aes(x = ValueRating, group = factor(OverallScore),
                                fill = as.factor(OverallScore)), position = "dodge") +
  scale_x_discrete(limits = c(0, 1, 2, 3, 4, 5), breaks = c(0, 1, 2, 3, 4, 5)) +
  guides(fill = guide_legend(title = "Overall Score", title.position = "top")) +
  xlab("Value Rating") + ylab("Frequency") +
  labs(title = "Frequency Of Overall Ratings\n Depending On Value Ratings")
```



```
#grid.arrange(plot1, plot2, plot3, plot4, plot5, plot6, ncol = 2)
#ggsave("Aviation_Final_Project.pdf", arrangeGrob(plot1, plot2, plot3, plot4, plot5, plot6))
```

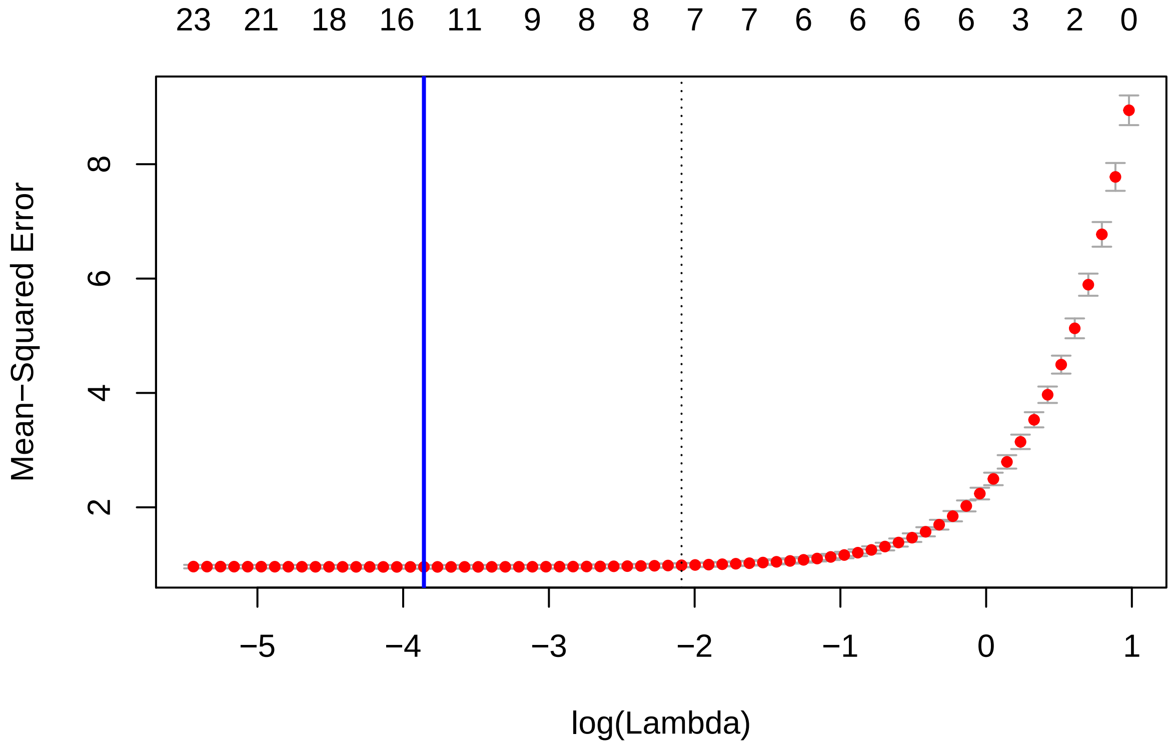
Lasso

```
x <- model.matrix(OverallScore ~ EntertainmentRating + GroundServiceRating + SeatComfortRating + Servic
#x <- model.matrix(OverallScore ~. , data = no_chars)[, -1]
y <- no_chars$OverallScore
grid.lambda <- 10^seq(10, -2, length = 100)
lasso.model <- glmnet(x, y, alpha = 1, lambda = grid.lambda)
train <- sample(1:nrow(x), nrow(x) / 2)
test <- (-train)
y.train <- y[train]
y.test <- y[test]


#Now, fit a Lasso regression model to the training data
lasso.model.train <- glmnet(x[train, ], y.train, alpha = 1, lambda = grid.lambda)


#Perform cross validation on the training set to select the best lambda
set.seed(1) #for reproducability
cv.out <- cv.glmnet(x[train, ], y.train, alpha = 1)
plot(cv.out)

#Find the best lambda value
```

```r
best.lambda <- cv.out$lambda.min
best.lambda
```

```
## [1] 0.02111966
```

```r
plot(cv.out)
abline(v = log(best.lambda), col = "blue", lwd = 2)
```



```r
#Calculate the MSPE of the model on the test set
lasso.pred <- predict(lasso.model.train, s = best.lambda, newx = x[test,])
mspe.lasso <- mean((lasso.pred - y.test)^2)
mspe.lasso
```

```
## [1] 1.110282
```

```r
#Fit the final model to the entire data set using the chosen lambda
final.model <- glmnet(x, y, alpha = 1, lambda = best.lambda)
(Coef.Lasso <- coef(final.model)[1:31,])
```

```
##                    (Intercept)           EntertainmentRating
##                     1.44072758                    0.03624755
##            GroundServiceRating             SeatComfortRating
##                     0.32266344                    0.19265314
##                  ServiceRating                   ValueRating
##                     0.23251317                    0.68182029
##                     FoodRating        AirNameAsiana Airlines
##                     0.19091879                   -0.01916740
## AirNameCathay Pacific Airways                AirNameEVA Air
##                     0.02471714                    0.00000000
##        AirNameGaruda Indonesia        AirNameHainan Airlines
##                     0.00000000                    0.00000000
##          AirNameJapan Airlines              AirNameLufthansa
##                     0.00000000                   -0.04557654
##           AirNameQatar Airways     AirNameSingapore Airlines
##                     0.00000000                    0.00000000
##              AircraftModelA320             AircraftModelA330
##                     0.08531816                    0.00000000
##              AircraftModelA340             AircraftModelA350
##                     0.00000000                    0.00000000
##              AircraftModelA380            AircraftModelATR72
##                     0.00000000                    0.00000000
##              AircraftModelB737             AircraftModelB747
##                     0.03302586                    0.00000000
##              AircraftModelB767             AircraftModelB777
##                     0.00000000                    0.00000000
##              AircraftModelB787       AircraftModelBombardier
##                     0.00000000                    0.00000000
##           AircraftModelEmbraer          AircraftModelUnknown
##                     0.00000000                   -0.08233218
##                  Recommendedno
##                    -2.05687741
```

```
pred <- predict(final.model, s = best.lambda, newx = x[test,])
final <- cbind(y[test], pred)
head(final)
```

```
##               1
## 1   1 2.384126
## 5   3 2.899689
## 10  2 1.190848
## 12  7 7.783455
## 14  6 6.489019
## 15 10 9.090913
```

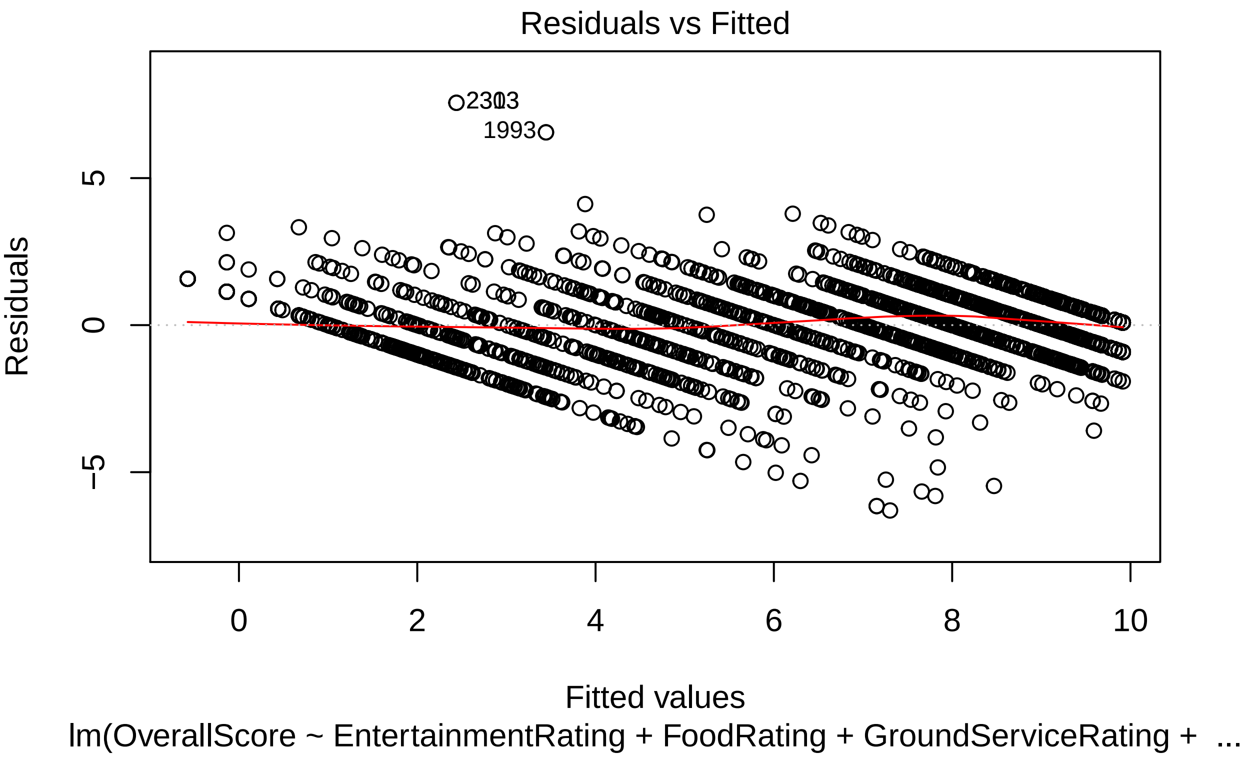
Note that we did not include DateFlown, ReviewDate, ReviewTitle, ReviewrCountry, or Route into our Lasso because doing so would have made Lasso given us an MSPE value for all the factors of each of the aforementioned columns, something we do not need. We already know that the date or route of the flight does not affect overall score.

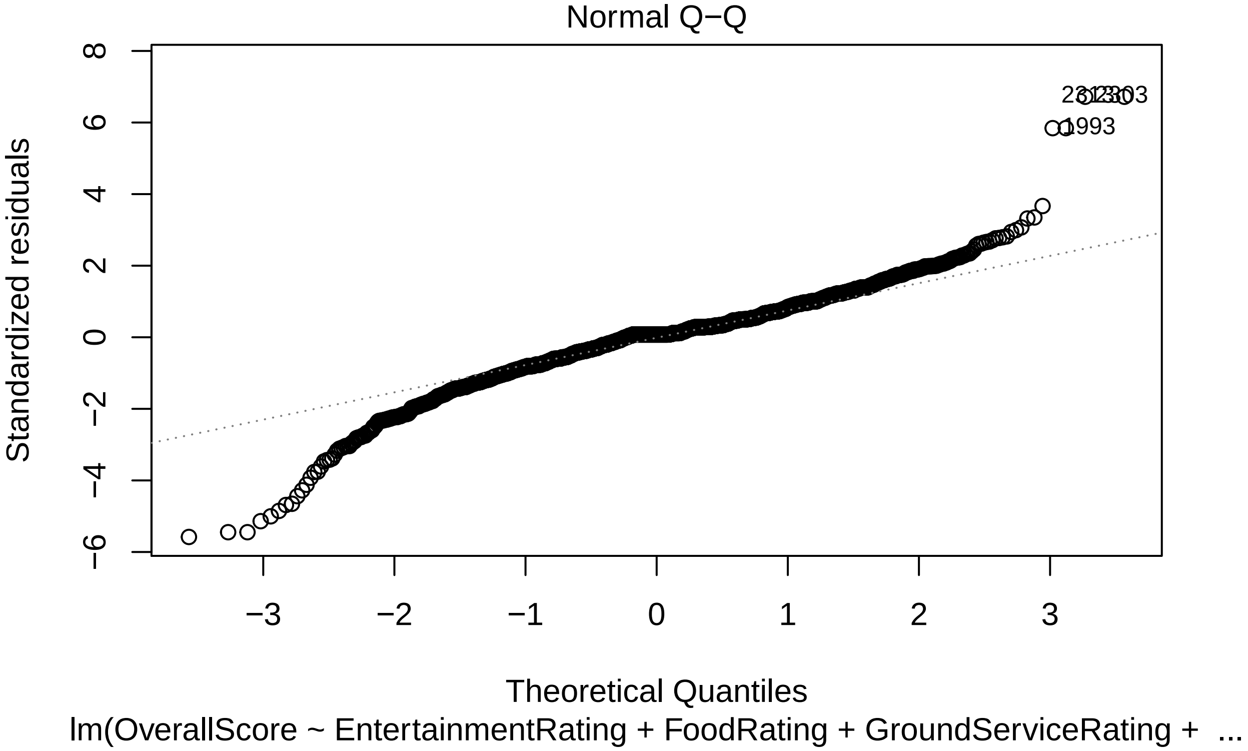
Lasso selected our variables! We see that the variables selected (with legal, positive MSPE) include the ratings such as: Entertainment Rating, Ground Service Rating, Seat Comfort Rating, Service Rating, Value Rating, and Food Rating.

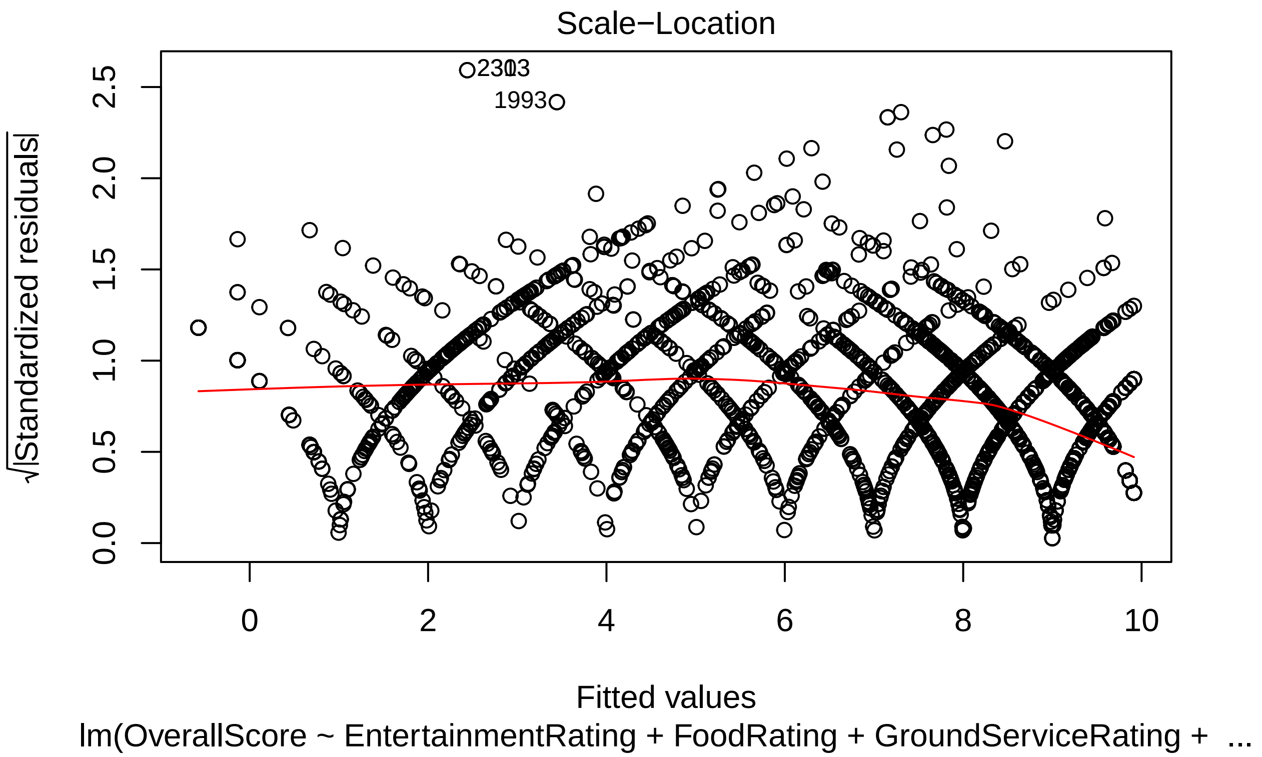
Diagnostics for Multiple Linear Regression

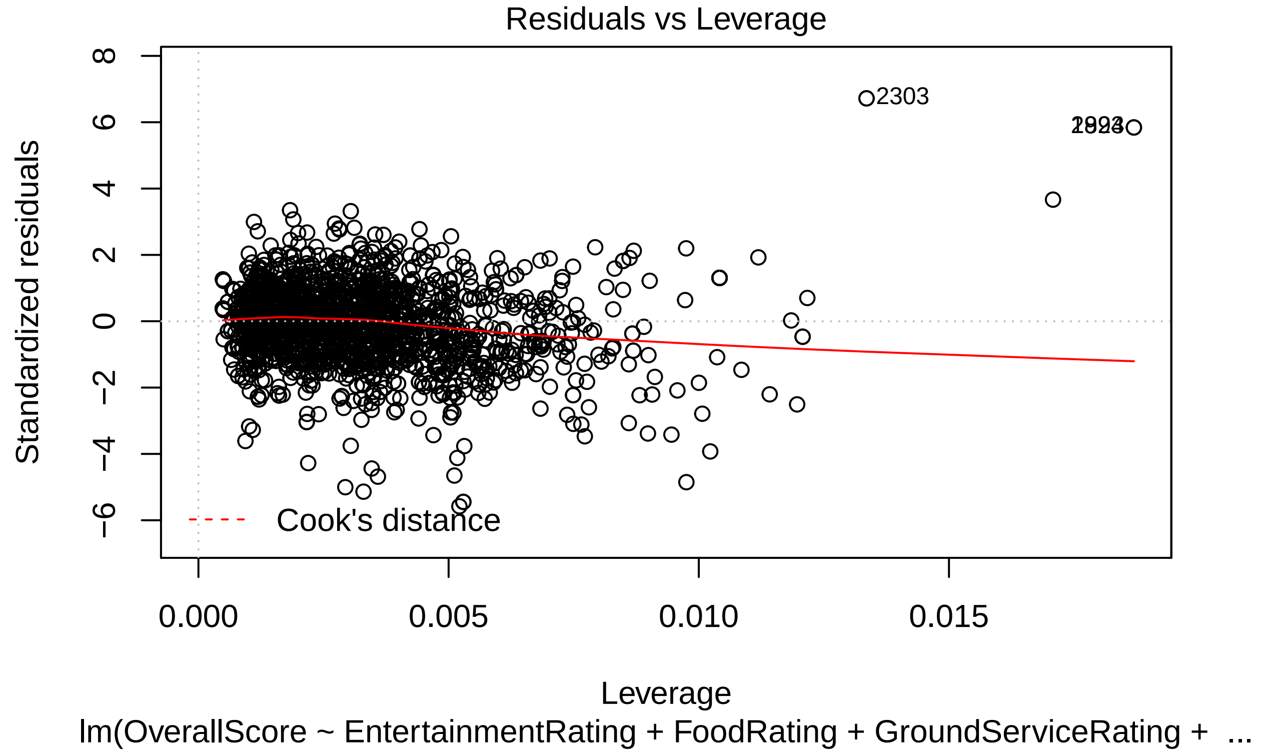
Homoscedasticity And Normality

```
# can do this with residual plots (plot residuals against fitted values)
#plot(fitted(reg), residuals(reg), xlab = "Fitted", ylab = "Residuals")
#abline(h=0)
plot(reg)
```

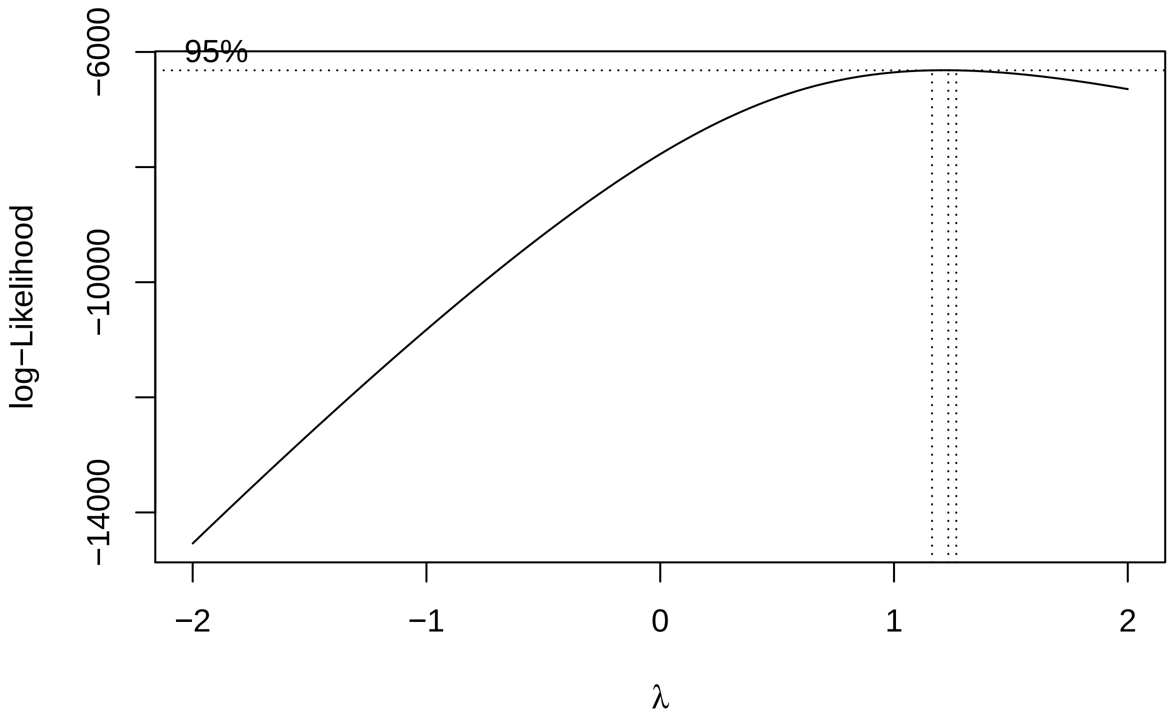

Residuals vs Fitted
2303
1993
Residuals
5
0
−5
0
2
4
6
8
10
Fitted values
lm(OverallScore ~ EntertainmentRating + FoodRating + GroundServiceRating + ...


Normal Q−Q
2313303
1993
Standardized residuals
8
6
4
2
0
−2
−4
−6
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(OverallScore ~ EntertainmentRating + FoodRating + GroundServiceRating + ...





```r
# transformation needed on y
# use Box-Cox method

# gives likelihood view rather than SSE. Identify lambda that maximizes the log-likehood/smallest SSE.
power_trans <- boxcox(reg)
```



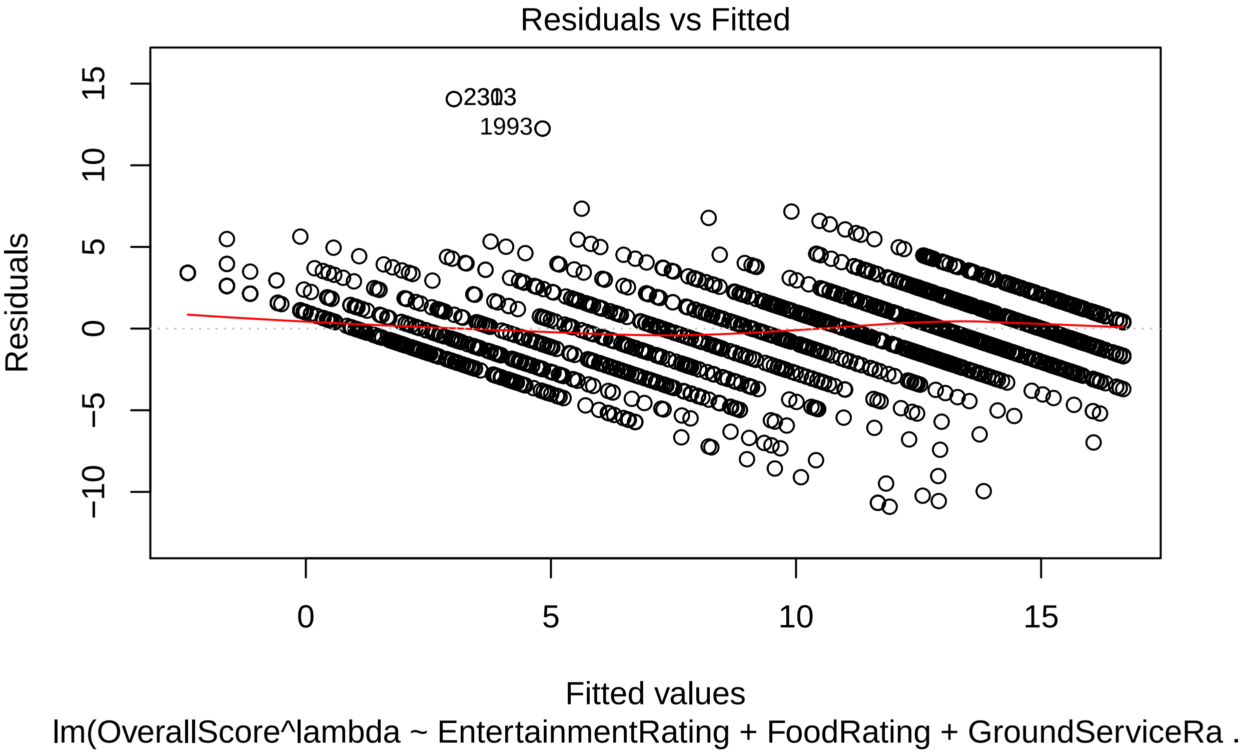
```r
which.max(power_trans$y)
```

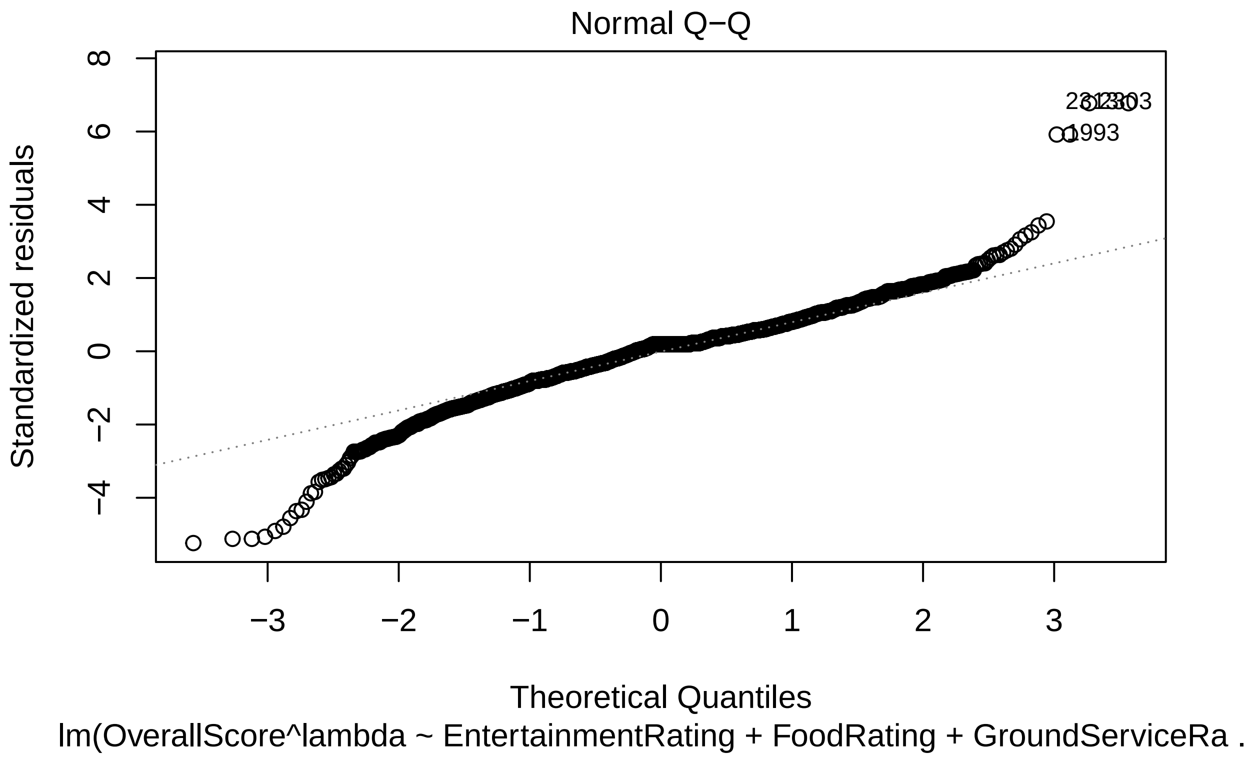
```
## [1] 81
```

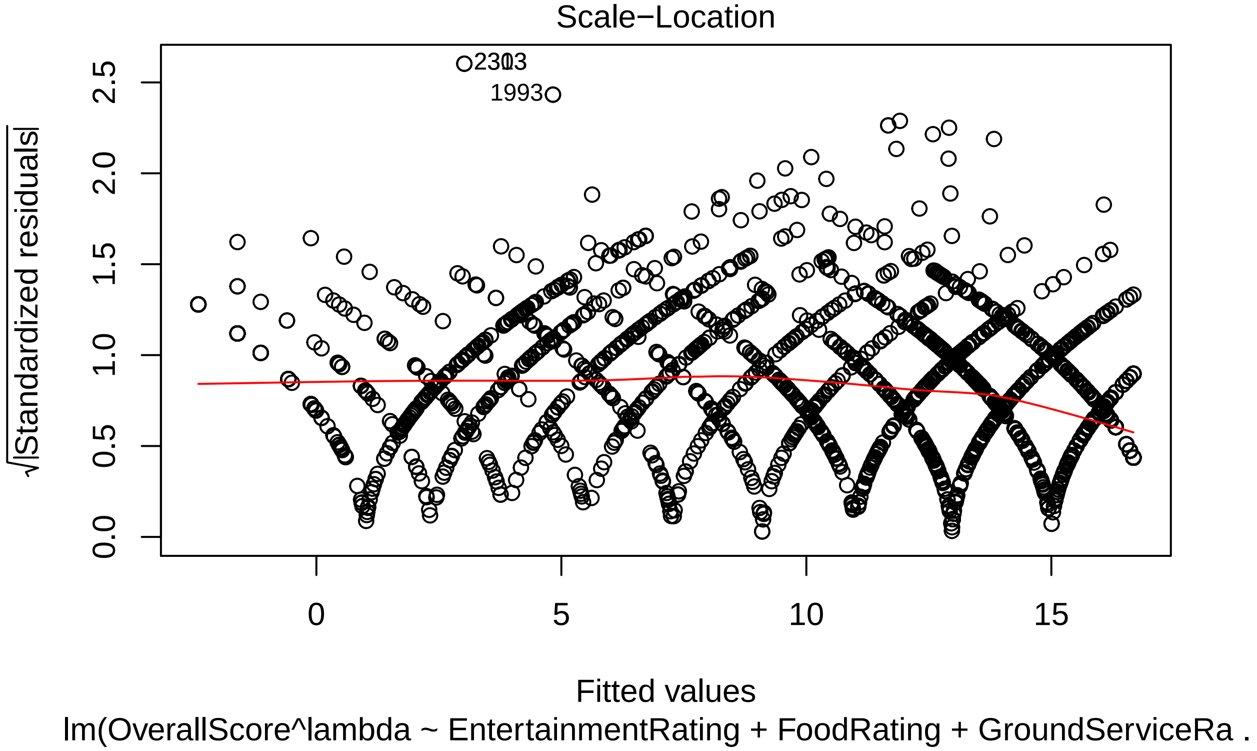
```r
lambda <- power_trans$x[which.max(power_trans$y)]

# maximum likelihood estimator near 1.23
# refit model with transformation
trans_model <- lm(OverallScore^lambda ~  EntertainmentRating + FoodRating + GroundServiceRating +
                    SeatComfortRating + ServiceRating + ValueRating, data = airlinesCleaned)
summary(trans_model) # goodness of fit slightly decreased
```

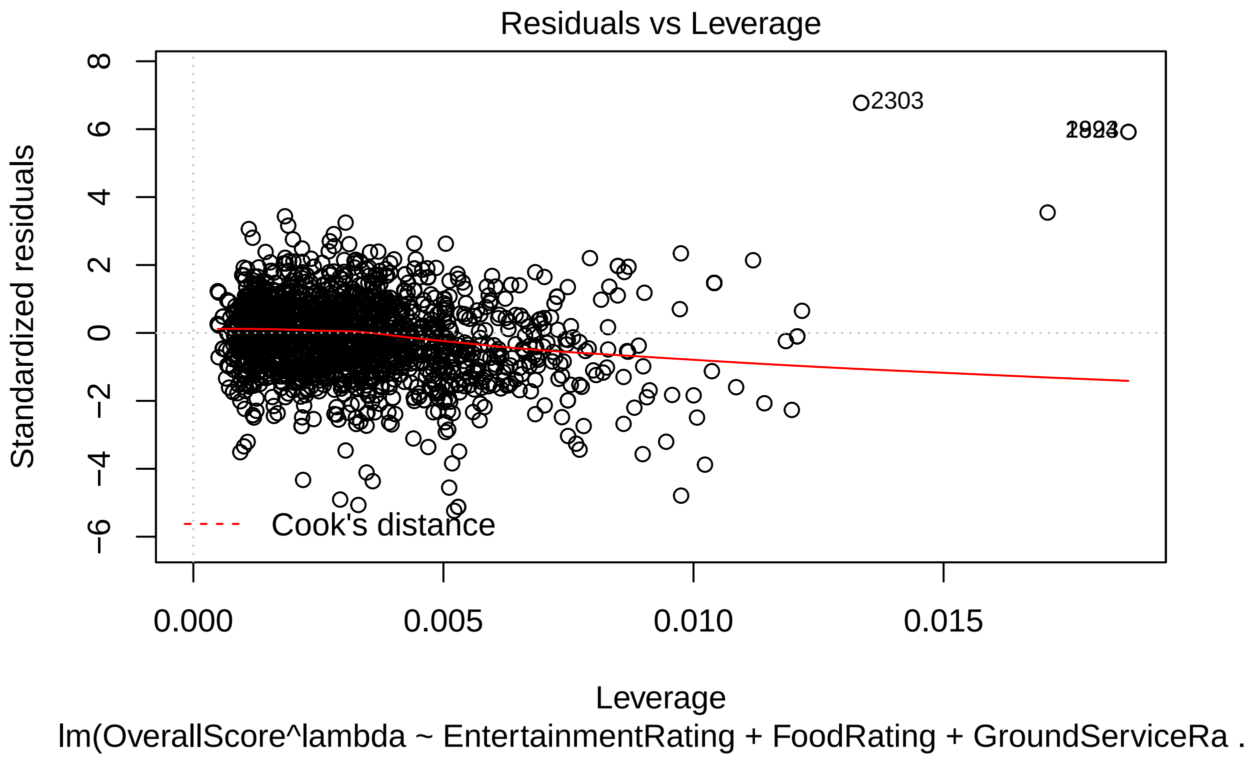
```
##
## Call:
## lm(formula = OverallScore^lambda ~ EntertainmentRating + FoodRating +
##     GroundServiceRating + SeatComfortRating + ServiceRating +
##     ValueRating, data = airlinesCleaned)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.9097  -1.1488   0.3976   1.1119  14.0541
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -4.21824    0.13635 -30.937  < 2e-16 ***
## EntertainmentRating  0.07245    0.02691   2.692  0.00714 **
## FoodRating           0.46000    0.04369  10.529  < 2e-16 ***
## GroundServiceRating  0.79893    0.03996  19.995  < 2e-16 ***
## SeatComfortRating    0.47400    0.04778   9.919  < 2e-16 ***
## ServiceRating        0.56402    0.05033  11.206  < 2e-16 ***
## ValueRating          1.80943    0.05176  34.959  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.089 on 2761 degrees of freedom
## Multiple R-squared:  0.8581, Adjusted R-squared:  0.8578
## F-statistic:  2783 on 6 and 2761 DF,  p-value: < 2.2e-16
```

```
plot(trans_model)
```




Normal Q−Q
Standardized residuals
8
6
4
2
0
−2
−4
−3
−2
−1
0
1
2
3
2313
2303
1993
Theoretical Quantiles
lm(OverallScore^lambda ~ EntertainmentRating + FoodRating + GroundServiceRa .


Scale−Location
√|Standardized residuals|
2.5
2.0
1.5
1.0
0.5
0.0
0
5
10
15
2313
2303
1993
Fitted values
lm(OverallScore^lambda ~ EntertainmentRating + FoodRating + GroundServiceRa .



```r
shapiro.test(residuals(reg))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(reg)
## W = 0.94412, p-value < 2.2e-16
```

Our residuals vs fitted plots look extraordinary. Fortunately, despite the odd jumps between supposed lines composed of residual points due to discrete nature of our predictors and response, we seem to have constant variance as indicated by the flat red line and a rough amount of equal number of residuals both above and below the red line.

The central values in the qq-plot above look good, but non-normality exists on extreme values of the residuals. This suggests some sort of squashing transformation on y ($\sqrt{}$ or log). Since the p-value < 2.2e-16 is significantly less than 0.05 and is statistically significant, there is strong evidence against the null hypothesis and so, we reject the null hypothesis that the residuals are normal at 5% significance level.

Leverage

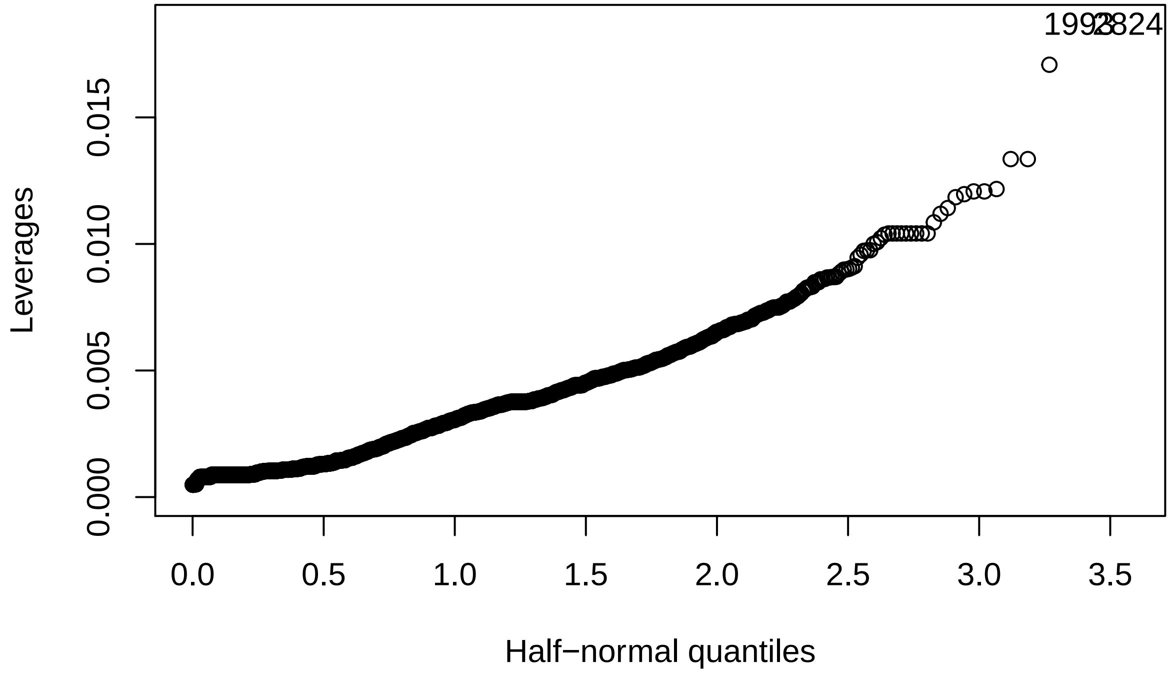
```r
# check for large leverage points
hatv <- hatvalues(reg)
head(hatv)
```

```
##            1            2            3            4            5
## 0.0056468065 0.0012971511 0.0007932376 0.0029615587 0.0047690995
##            6
## 0.0018371413
```

```r
sum(hatv) # verify sum of leverages is 7 - # of predictors in our model
```

```
## [1] 7
```

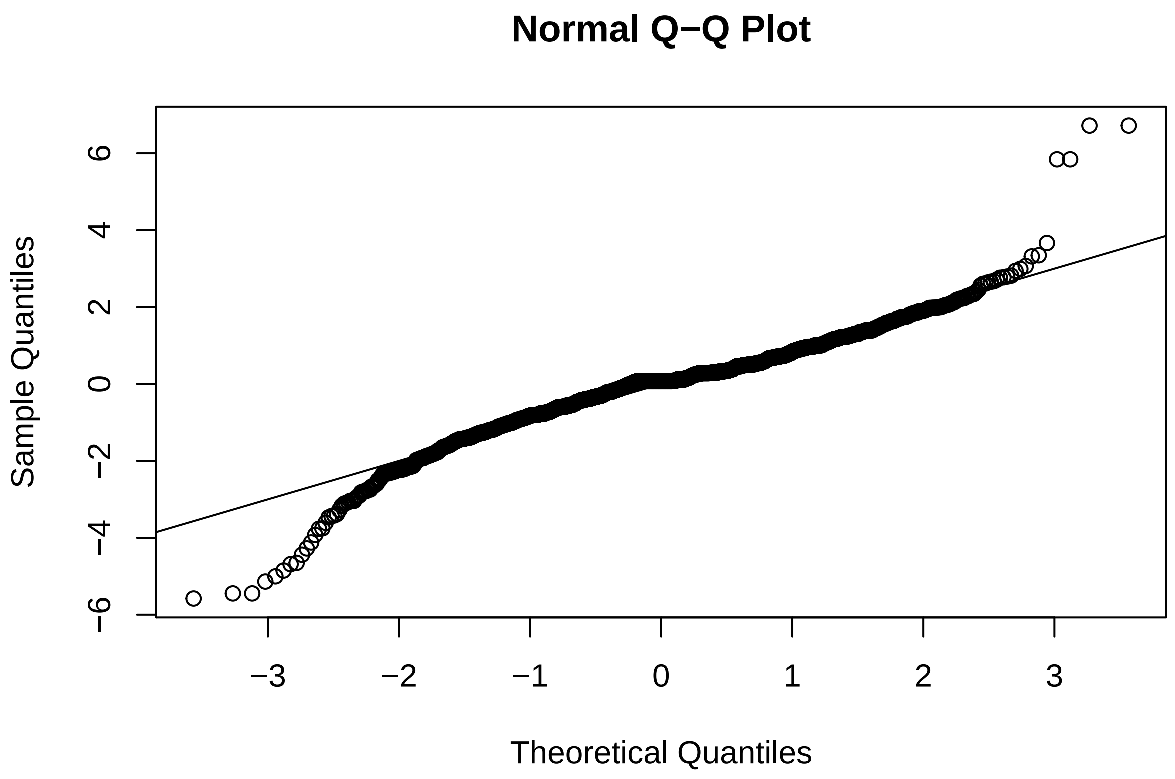
```
# half-normal plot is a good way to identify unusually large values of the leverage
# these plots are designed for assessment of positive data
review_id <- row.names(airlinesCleaned)
halfnorm(hatv, labs = review_id, ylab = "Leverages")
```



```
index <- sort(hatv, index = TRUE, decreasing = TRUE)$ix # extract the index
head(hatv[index]) # first 6 obs. with largest leverage
```

```
##       1993       2824       1424       2303       2313       3662
## 0.01869621 0.01869621 0.01707991 0.01335281 0.01335281 0.01216937
```

```
leverage <- c("1993", "2824", "1424", "2303", "2313", "3662")
levSubset <- subset(airlinesCleaned, rownames(airlinesCleaned) %in% leverage)

# leverages can also be used in scaling residuals (standardized residuals)
# if model assumptions are correct, var(ri) = 1 and corr(ri, rj) tend to be small

# standardized to have equal variance
qqnorm(rstandard(reg))
abline(0, 1) # intercept and slope
```



```
# due to standardization of residuals, points are expected to approx. follow y=x line if
# normality holds
```

There exists a few leverage points, both equal tied in terms of extremity. For these most extreme points, their Entertainment ratings and Food Ratings were 0. Interestingly, these passengers still recommended the airlines to other people.

Outliers

```
# check for outliers

# compute studentized residuals for airlinesCleaned data
stud_res <- rstudent(reg)
summary(stud_res)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -5.609948 -0.528115  0.075903 -0.000129  0.500531  6.774311
```

```
# pick out the largest residual
stud_res[which.max(abs(stud_res))]
```

```
##     2303
## 6.774311
```

```
# largest six residuals
index <- sort(abs(stud_res), index = TRUE, decreasing = TRUE)$ix # extract the index
head(stud_res[index]) # first 6 outliers
```

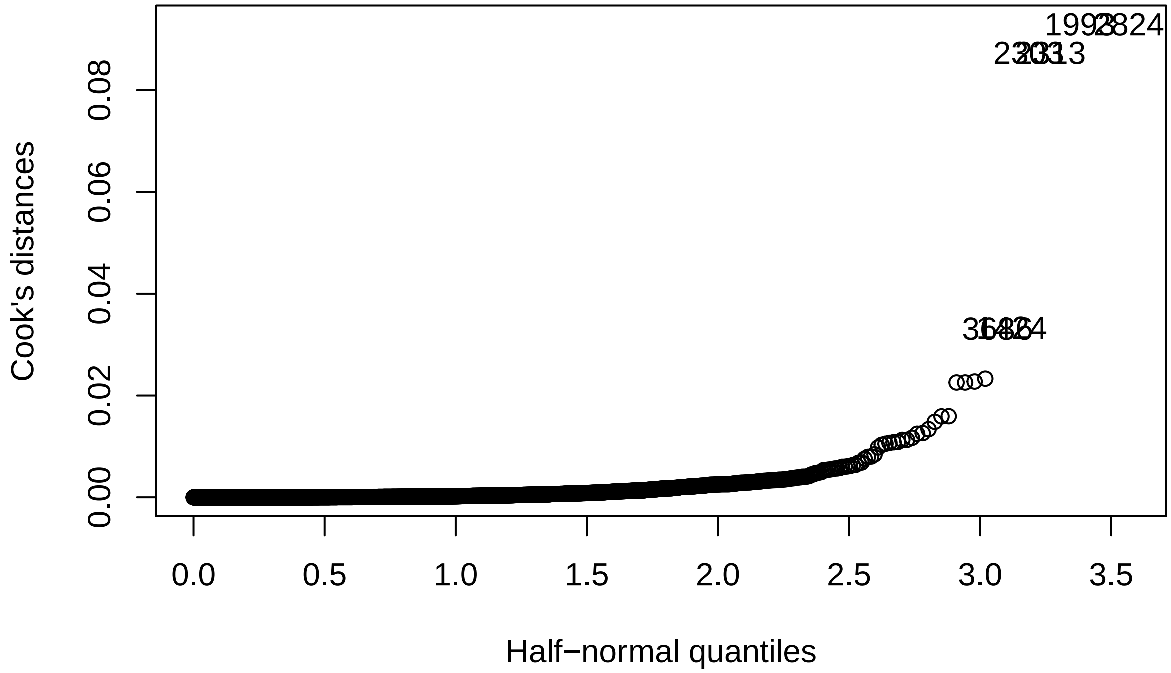
```
##      2303      2313      1993      2824      1341       404
##  6.774311  6.774311  5.878096  5.878096 -5.609948 -5.475435
```

```
outliers <- c("2303", "2313", "1993", "2824", "1341", "404")
outlierSubset <- subset(airlinesCleaned, rownames(airlinesCleaned) %in% outliers)
```

Recall outliers (in terms of responses) are observations where the error (distance between the truth and fitted value, or y-difference) are large in magnitude. Studentized residuals are the quotient resulting from division of a residual by the best estimate of standard deviation of residuals. To check for outliers, we look at residuals more extreme than 3. The largest residual, review # 2303 (an Asiana Airlines' flight in April 2019 from Chicago to Seoul), is significantly large for a standard normal scale. This requires further investigation and check on whether this observation is also an influential point.

Influential Points

```
# identify influential obs. with half-normal plot of Cook's distance

cook <- cooks.distance(reg)
halfnorm(cook, 6, labs = review_id, ylab = "Cook's distances")
```



```
# largest six Cook's distances
index <- sort(cook, index = TRUE, decreasing = TRUE)$ix # extract the index
head(cook[index]) # first 6 obs. with largest Cook's distance
```

```
##       1993       2824       2303       2313       1424       3686
## 0.09291369 0.09291369 0.08730488 0.08730488 0.03336360 0.03312069
```

```
influential <- c("1993", "2824", "2303", "2313", "1424", "3686")
(influentialSubset <- subset(airlinesCleaned, rownames(airlinesCleaned) %in% influential))
```

```
##                      AirName AircraftModel DateFlown EntertainmentRating
## 1424               Lufthansa       Unknown    Jun-18                   0
## 1993 ANA All Nippon Airways       Unknown    Aug-17                   0
## 2303         Asiana Airlines       Unknown    Apr-19                   0
## 2313         Asiana Airlines       Unknown    Apr-19                   0
## 2824 Cathay Pacific Airways       Unknown    Nov-17                   0
## 3686       Garuda Indonesia       Unknown    Aug-17                   4
##      FoodRating GroundServiceRating Recommended        ReviewDate
## 1424          0                   1         yes    20th June 2018
## 1993          0                   0         yes  17th August 2017
## 2303          0                   0         yes   12th April 2019
## 2313          0                   0         yes   12th April 2019
## 2824          0                   0         yes 17th October 2018
## 3686          0                   5          no  12th August 2017
##                           ReviewTitle    ReviewrCountry
## 1424      "operated by Deutsche Bahn"        (Germany)
## 1993 "highlight the customer service"  (United States)
## 2303     "such an amazing experience"  (United States)
## 2313     "such an amazing experience"  (United States)
## 2824   "kind and very helpful manner"         (Canada)
## 3686       "requested a special meal"      (Indonesia)
##                                     Route SeatComfortRating        SeatType
## 1424                          QKL to FRA                 0   Economy Class
## 1993 Kuala Lumpur to Chicago via Narita                  0 Premium Economy
## 2303                    Chicago to Seoul                 0   Economy Class
## 2313                    Chicago to Seoul                 0   Economy Class
## 2824                    Dubai to Bahrain                 0   Economy Class
## 3686               Denpasar to Melbourne                 4   Economy Class
##      ServiceRating     TravelType ValueRating
## 1424             0       Business           5
## 1993             0 Family Leisure           5
## 2303             0 Couple Leisure           4
## 2313             0 Couple Leisure           4
## 2824             0   Solo Leisure           5
## 3686             5 Family Leisure           5
```

Recall influential points are observations that are both an outlier and a leverage point. Its removal from the data would cause a large (significant) change in the fit. This can be checked with Cook statistics as these popular influence diagnostics reduce the information to a single value for each case.

The largest six values are identified in the plot above. Since reviews 1993, 2303, 2313, 2824 were flagged as an outlier and 1424 has high/noticeable leverage, these may be influential points that can be further investigated. Let's compare the model fit by observing how it changes with/without the aforementioned reviews.

Final Fit

```
summary.data # full model fit
```

```
##
## Call:
## lm(formula = OverallScore ~ EntertainmentRating + FoodRating +
##     GroundServiceRating + SeatComfortRating + ServiceRating +
##     ValueRating, data = airlinesCleaned)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.3032 -0.5969  0.0860  0.5666  7.5607
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -1.57988    0.07395 -21.365  < 2e-16 ***
## EntertainmentRating  0.04679    0.01460   3.206  0.00136 **
## FoodRating           0.23059    0.02369   9.732  < 2e-16 ***
## GroundServiceRating  0.43895    0.02167  20.256  < 2e-16 ***
## SeatComfortRating    0.24532    0.02592   9.466  < 2e-16 ***
## ServiceRating        0.33235    0.02730  12.175  < 2e-16 ***
## ValueRating          1.00479    0.02807  35.795  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.133 on 2761 degrees of freedom
## Multiple R-squared:  0.8615, Adjusted R-squared:  0.8612
## F-statistic:  2862 on 6 and 2761 DF,  p-value: < 2.2e-16
```

```
reg <- lm(OverallScore ~ EntertainmentRating + FoodRating + GroundServiceRating + SeatComfortRating +
            ServiceRating + ValueRating, data = airlinesCleaned)

#model1_c <- lm(OverallScore ~ EntertainmentRating + FoodRating + GroundServiceRating + SeatComfortRati
#summary(model1_c) # model fit w/o most extreme Cook's distance obs. (reviews 1993 & 2824)

model1_c2 <- lm(OverallScore ~ EntertainmentRating + FoodRating + GroundServiceRating +
                  SeatComfortRating + ServiceRating + ValueRating, data = airlinesCleaned,
                subset = (!(rownames(airlinesCleaned) %in% c("1993", "2824", "2303", "2313"))))
summary(model1_c2) # model fit w/o 4 most extreme influential points
```

```
##
## Call:
## lm(formula = OverallScore ~ EntertainmentRating + FoodRating +
##     GroundServiceRating + SeatComfortRating + ServiceRating +
##     ValueRating, data = airlinesCleaned, subset = (!(rownames(airlinesCleaned) %in%
##     c("1993", "2824", "2303", "2313"))))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1425 -0.5575  0.0681  0.5839  4.6056
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -1.69594    0.07227 -23.468  < 2e-16 ***
## EntertainmentRating   0.04968    0.01416   3.509 0.000457 ***
## FoodRating            0.24229    0.02300  10.533  < 2e-16 ***
## GroundServiceRating   0.46614    0.02112  22.067  < 2e-16 ***
## SeatComfortRating     0.27619    0.02525  10.939  < 2e-16 ***
## ServiceRating         0.36642    0.02661  13.772  < 2e-16 ***
## ValueRating           0.92484    0.02791  33.139  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.099 on 2757 degrees of freedom
## Multiple R-squared:  0.8697, Adjusted R-squared:  0.8694
## F-statistic:  3066 on 6 and 2757 DF,  p-value: < 2.2e-16
```

The coefficients of the newly fitted regression model changed after removing the four most extreme influential points with the variable ValueRating changing the most (-8%), SeatComfortRating and ServiceRating coming in second with the most change (~3%), GroundServiceRating (2.7%), FoodRating (1.17%), and EntertainmentRating (0.29%) with the least change. That is, the estimated coefficients in the original model are determined by each of the removed observations. We do not like our estimates to be so sensitive to the presence of these 4 reviews. Compared to the full model fit, the goodness of fit (Rˆ2) of our final model also slightly increased.

We trust the model because intuitively it makes sense. If the people at the terminals and before boarding are kind, then of course the overall quality increases.

Discussion

This model answered our questions on what factors contribute most to the overall quality of the airlines. It can be deduced that the top 10 global airlines are good at multiple areas instead of just one. We know that some of the most influential factors to improve overall quality of an airline are Ground Service, Seat Comfort, Cabin Service, Value (with respect to ticket price), and Food. Among them, Value has greatest impact on customer satisfaction. Thus, we suggest airlines to focus its most essential tangible and intangible resources on value for money. With cabin service coming in 2nd, it is another key factor to create a good image in the service industry and must be constantly managed to maintain the image of the company.

Some improvements to our data would be to include seat type and aircraft type int our regression model and observe its effects on overall quality of his or her flight. Furthermore, since we dealt with discrete/count values in our data, our choice of linear regression was a mediocre solution to analyze our data. A Poisson model may have been a better solution. Another limitation is that this study was only a surface understanding of the factors that impact flyers' impressions as each flight aspect merits an investigation on its own. For instance, research into the cost and benefit of improving seat comfort would be highly beneficial for an airline looking to improve sales through seat comfort.

In the future, we hope to include text of the review rather than the review title to perform frequency/semantic network analysis to better understand customer's thoughts for sustainable marketing decisions against competitors. Because the study focused on airlines with strong presence on Skytrax, we can consider social media data reviews - known around the world - as a better alternative to understand consumer trends. We are interested in employing advanced ML techniques to help predict customer flight ratings and sales with unused scraped data about routes, seat & traveler type, and date flown (possibly weather data).