STAT 33B Homework 3

Ming Fong (3035619833)

Oct 8, 2020

This homework is due **Oct 8, 2020** by 11:59pm PT.

Homeworks are graded for correctness.

As you work, write your answers in this notebook. Answer questions with complete sentences, and put code in code chunks. You can make as many new code chunks as you like.

Please do not delete the exercises already in this notebook, because it may interfere with our grading tools.

You need to submit your work in two places:

- Submit this Rmd file with your edits on bCourses.
- Knit and submit the generated PDF file on Gradescope.

If you have any last-minute trouble knitting, **DON'T PANIC**. Submit your Rmd file on time and follow up in office hours or on Piazza to sort out the PDF.

The Bay Area Vehicles Data Set

The Bay Area Vehicles Data Set is a collection of advertisements for vehicles for sale in the San Francisco Bay Area. The data set was collected from the website Craigslist on Sep 28, 2020.

The data set is available on the bCourse as `2020.09_cl_vehicles.rds`.

Each row is one advertisement. The columns are:

- `title`: title of advertisement
- `text`: full text of advertisement
- `latitude`: latitude of vehicle
- `longitude`: longitude of vehicle
- `city_text`: city listed in advertisement
- `date_posted`: date advertisement was posted
- `date_updated`: date advertisement was updated, if any
- `price`: price in US dollars
- `vin`: vehicle identification number (like a serial number)
- `condition`: condition, as a category
- `drive`: type of drivetrain
- `fuel`: type of fuel used
- `odometer`: odometer reading, in miles
- `transmission`: type of transmission
- `type`: type of vehicle (sedan, truck, van, etc.)
- `year`: year vehicle was manufactured
- `make`: brand of vehicle
- `model`: model of vehicle
- `craigslist`: craigslist region where advertisement was posted
- `place`: place name (like city, but also includes small towns) based on latitude/longitude

- `city`: city based on latitude/longitude
- `state`: state based on latitude/longitude
- `county`: county based on latitude/longitude

Many of the columns were programmatically extracted from the `title` and `text`, so there may be missing or incorrect values.

Exercise 1

Read the vehicles data set into R, then use R functions to answer the following:

1. How many advertisements are there?

2. Which columns are categorical but aren't factors? Convert these to factors.

   Hint 1: Remember that categorical features are usually qualitive and have a limited set of possible values.

   Hint 2: You can use subsetting and `lapply()` to convert many columns at once.

3. What percentage of each column is missing? Which columns have a lot of missing values?

   Hint 1: Call `is.na()` on each column, then use `colSums()`.

   Hint 2: Yes, the second question is a little bit vague. Think of it as the sort of casual question a supervisor might ask you in an industry job. Your answer should clarify how you interpreted "a lot of missing values".

YOUR ANSWER GOES HERE:

1. There are 14990 advertisements (rows).

```r
data = readRDS("data/2020.09_cl_vehicles.rds")
nrow(data)
```

```
## [1] 14990
```

2. Categorical features to be converted to factors are: `$make $model $place $city $state $county`

```r
data[, c("make", "model", "place", "city", "state", "county")] =
  lapply(data[, c("make", "model", "place", "city", "state", "county")],
  factor)
```

3. The columns with "a lot" (> 10%) of missing values are: `date_updated, vin, condition, drive, odometer, place, city`

```r
# Count of NA by column
na_count = colSums(sapply(data[, ], is.na))
# Percentages of NA by column
na_percentages = na_count / nrow(data) * 100
na_percentages
```

```
##        title         text     latitude    longitude    city_text  date_posted
##  0.000000000  0.000000000  2.368245497  2.368245497  3.862575050  0.000000000
## date_updated        price          vin    condition        drive         fuel
## 58.192128085  6.190793863 36.104069380 34.596397598 17.638425617  0.013342228
##     odometer transmission         type         year         make        model
## 10.880587058  0.527018012  9.472981988  0.006671114  3.308872582  3.308872582
##        fname   craigslist        place         city        state       county
##  0.000000000  0.000000000 14.663108739 21.974649767  2.448298866  2.448298866
```

```r
# We will use 10% of rows as the cutoff for "a lot"
# columns with TRUE have > 10% missing values
na_percentages > 10
```

```
##        title         text     latitude    longitude    city_text  date_posted
##        FALSE        FALSE        FALSE        FALSE        FALSE        FALSE
## date_updated        price          vin    condition        drive         fuel
##         TRUE        FALSE         TRUE         TRUE         TRUE        FALSE
##     odometer transmission         type         year         make        model
##         TRUE        FALSE        FALSE        FALSE        FALSE        FALSE
##        fname   craigslist        place         city        state       county
##        FALSE        FALSE         TRUE         TRUE        FALSE        FALSE
```

Exercise 2

1. Compute the number of missing values in each row.

   Hint 1: Call `is.na()` on each row.

   Hint 2: Some of the apply functions transpose the results. The `dim()` function is one way to check.

2. Use ggplot2 to make a bar plot of the numbers from from part 1. Make sure to put an appropriate title and labels on your plot.

   Hint: You can create a data frame with the `data.frame()` function.

3. When a row in this data set has missing values, does it tend to have a lot of missing values, or only a few?

YOUR ANSWER GOES HERE:

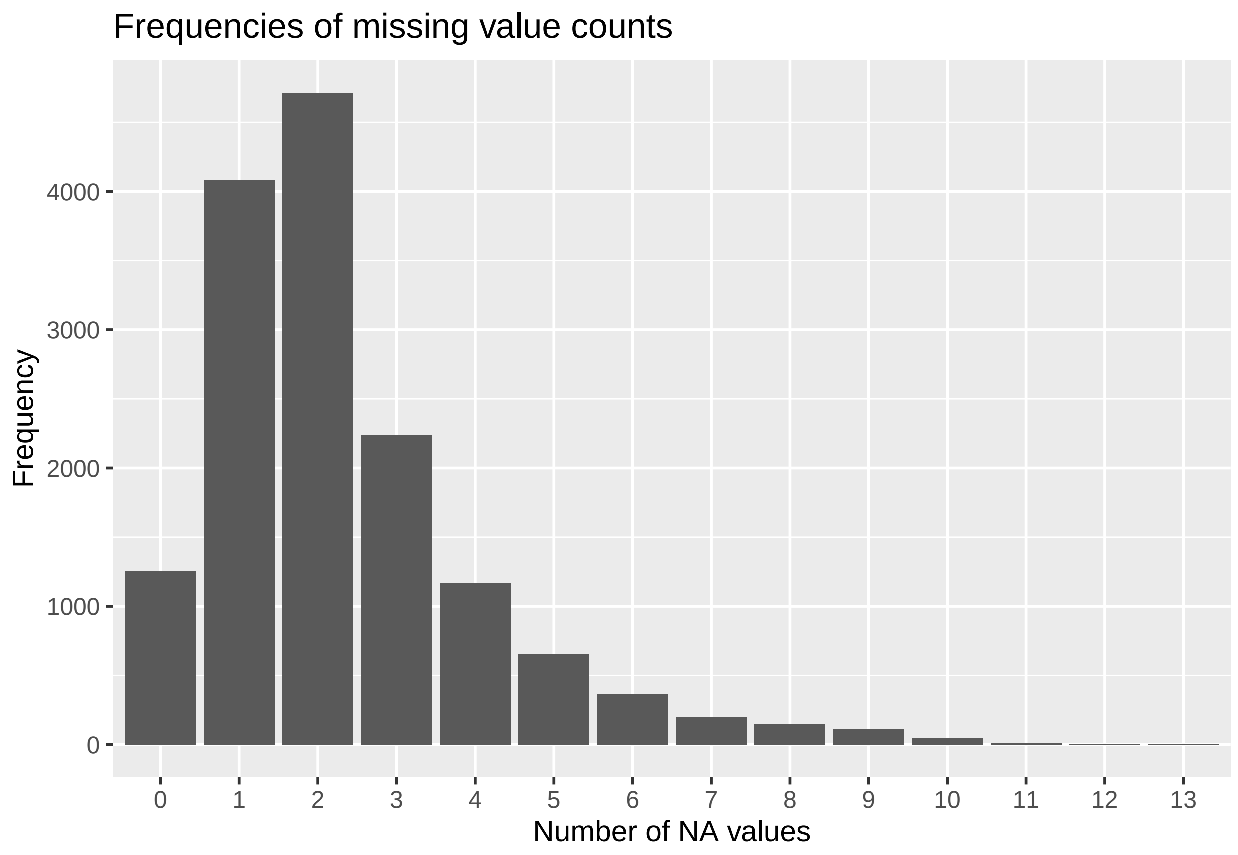
1.

```r
row_na_count = colSums((apply(data, 1, is.na)))
dim(row_na_count)
```

```
## NULL
```

2.

```r
library(ggplot2)
ggplot(data.frame(table(row_na_count)), aes(y = Freq, x = row_na_count)) +
geom_bar(stat = "identity") +
ggtitle("Frequencies of missing value counts") +
xlab("Number of NA values") + ylab("Frequency")
```



3. Rows tend to have fewer missing values. This can be seen in right-skewed distribution of the bar plot. The center of the distribution is around 2 missing values.

Exercise 3

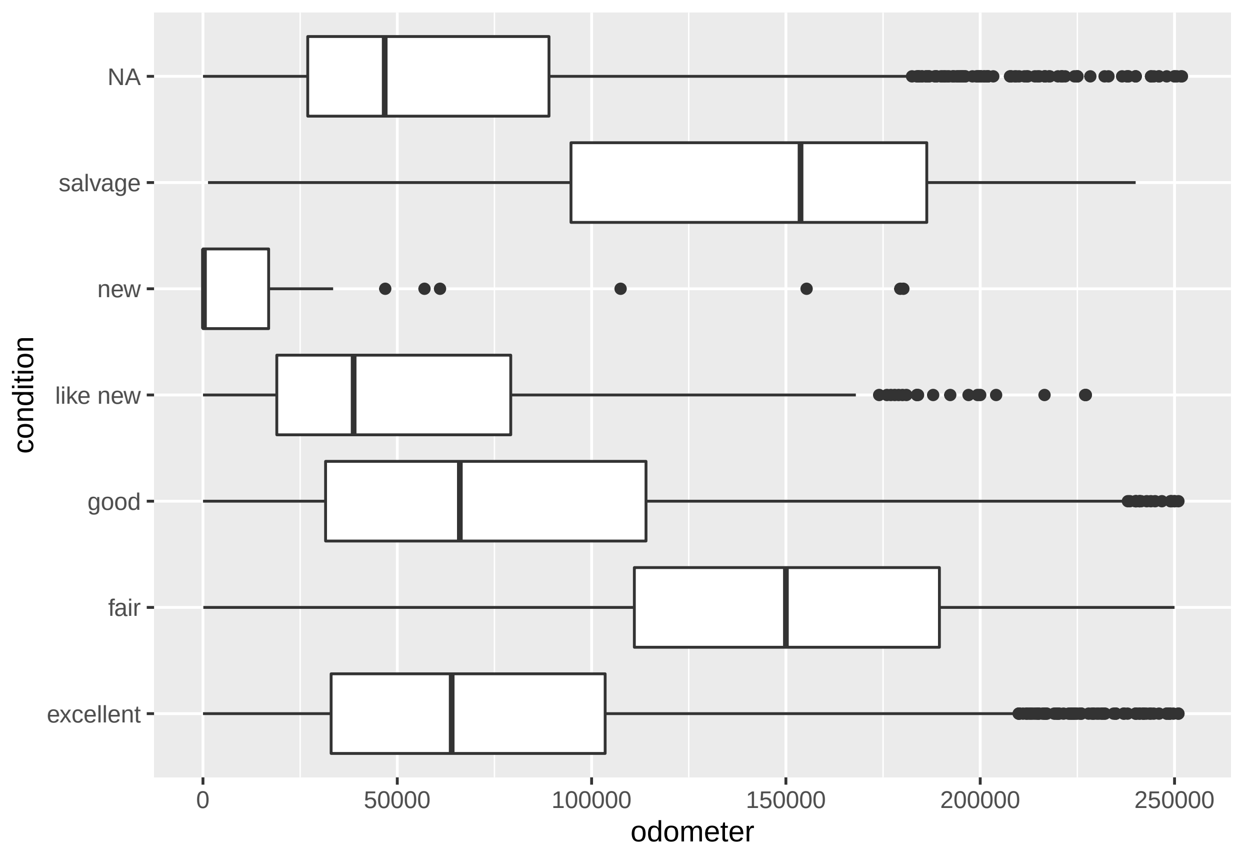
Make a box plot of `odometer` readings, broken down by the `condition` of the vehicle. Remove any extreme `odometer` values, so that it is easy to compare the boxes.

Comment briefly on the distribution of odometer readings for the various conditions.

Hint: There are several ways to identify extreme values. One way is to make a box plot. Another way is to find values above a certain percentile (`quantile()`), say 99%. Yet another way is to find values more than 2-3 standard deviations (`sd()`) from the mean. Each has trade-offs, but we won't focus on those in this class.

YOUR ANSWER GOES HERE:

```
# removing outliers
Q = quantile(data$odometer, probs = c(.01, .99), na.rm = TRUE)

ggplot(subset(data, data$odometer < Q[2] & data$odometer > Q[1]),
aes(x = odometer, y = condition)) +
geom_boxplot()
```



As the condition of the car gets worse, the milage on the odometer tends to increase. This makes sense because cars that are driven more are more likely to be in bad condition.

Exercise 4

Answer each question about advertisements for vehicles **in Berkeley**.

Hint: You might want to get started by taking a subset. Watch out for missing values.

1. How many advertisements are for vehicles in Berkeley?
2. How many of each `type` of car are there? Which type is the most common?
3. What's the median price (ignoring missing values) of each `type` of car? Which type has the highest median, and which has the lowest?

YOUR ANSWER GOES HERE:

1. There are 73 advertisements for vehicles in Berkeley. This result is the same using either `$place or $city`.

```
data_in_berkeley = subset(data, data$city == "Berkeley")
nrow(data_in_berkeley)
```

```
## [1] 73
```

2. The most common type of car is `sedan` with 28 advertisements.

```
table(data_in_berkeley$type)
```

```
##
```

```
##          bus convertible        coupe    hatchback     mini-van      offroad
##            0            2            2           11            0            0
##        other       pickup        sedan          suv        truck          van
##            0            2           28           12            1            3
##        wagon
##            2
```

```
# The mose common type of car
head(sort(table(data_in_berkeley$type), decreasing = TRUE), 1)
```

```
##
## sedan
##    28
```

3. The highest median price type is `coupe` at `39300` while the lowest is `wagon` at `13400`.

```
tapply(data_in_berkeley$price, data_in_berkeley$type, median, na.rm = TRUE)
```

```
##          bus convertible        coupe    hatchback     mini-van      offroad
##           NA        38150        39300        13791           NA           NA
##        other       pickup        sedan          suv        truck          van
##           NA        37200        15991        16993        17500        14900
##        wagon
##        13400
```

```
# Highest median price
head(sort(tapply(data_in_berkeley$price, data_in_berkeley$type, median, na.rm = TRUE), decreasing = TRU
```

```
## coupe
## 39300
```

```
# Lowest median price
tail(sort(tapply(data_in_berkeley$price, data_in_berkeley$type, median, na.rm = TRUE), decreasing = TRU
```

```
## wagon
## 13400
```

Exercise 5

1. Make a density plot of price. Use three separate lines for ads in San Francisco, San Jose, and Oakland (omit the other cities).

   Hint: You can use the `droplevels()` function to drop factor levels that aren't present.

2. How do price distributions of the three cities compare?

3. Based on the plot, which of these cities have ads with extreme/anomalous prices? Isolate one of these ads. Does the extreme price seem accurate, or is it a mistake? Use the original title and text of the ad as evidence.

   Hint 1: You can print the text of an ad in human-readable form with the `message()` function.

   Hint 2: You can use the `stringr` package's `str_wrap()` function to wrap long strings (e.g., the ad text) for printing in the notebook.

   Hint 3: The PDFLaTeX program that RMarkdown uses to knit PDFs only supports ASCII characters. Many of the advertisments contain non-ASCII characters. If you get a knit error like `! Package inputenc Error: Unicode character`, you probably printed an ad with non-ASCII characters.

To fix it, you can either comment out the line that prints the ad, or switch from PDFLaTeX to XeLaTeX or LuaLaTeX. See https://bookdown.org/yihui/rmarkdown-cookbook/latex-unicode.html for details about how to switch.

YOUR ANSWER GOES HERE:

1.

```
# Prepare data
three_cities_data = subset(data, city == c("San Francisco", "Oakland", "San Jose"))
```

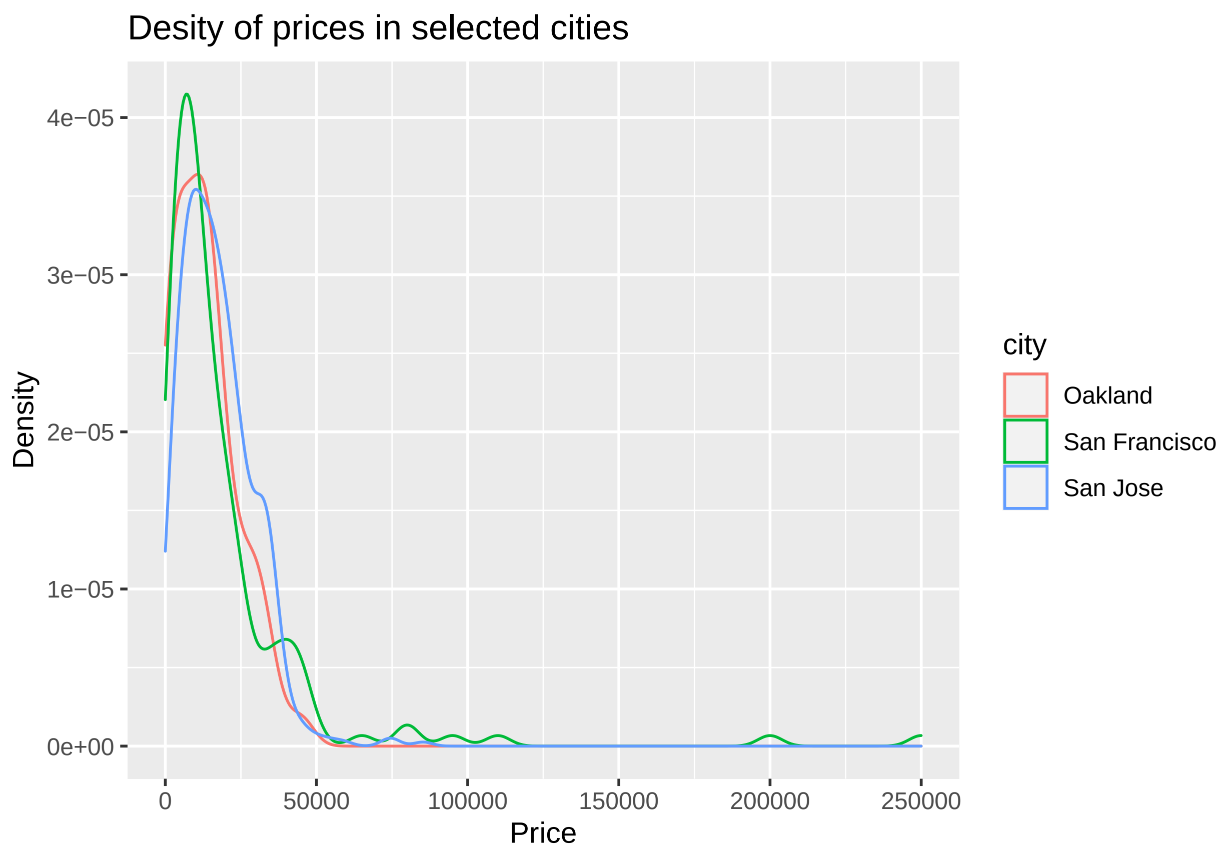
```
## Warning in `==.default`(city, c("San Francisco", "Oakland", "San Jose")): longer
## object length is not a multiple of shorter object length

## Warning in is.na(e1) | is.na(e2): longer object length is not a multiple of
## shorter object length
```

```
three_cities_data$city = droplevels(three_cities_data$city)
levels(three_cities_data$city)
```

```
## [1] "Oakland"       "San Francisco" "San Jose"
```

```
ggplot(three_cities_data, aes(x = price, color = city)) + geom_density() +
ggtitle("Desity of prices in selected cities") + xlab("Price") + ylab("Density")
```

```
## Warning: Removed 38 rows containing non-finite values (stat_density).
```



2. The price distrubutions are similar. San Francisco seems to have slightly higher prices.
3. San Francisco has some very extreme values for prices. The two most expensive ads are both Ferraris, so the price makes sense. The cars are simply very expensive.

```r
library(stringr)
extreme_value = subset(three_cities_data, three_cities_data$price > 240000)
extreme_value$title
```

```
## [1] "2019 Ferrari 488 GTB - 650 Score WE CARRY CONTRACTS - $249,995 (mountain view)"
```

```r
text <- str_c(extreme_value$text, collapse = "\n")
cat(str_wrap(text), "\n")
```

```
## QR Code Link to This Post Call Adam, 18052330098 for more details! Lease for
## $1,956+ tax for 60 months. This is a commercial lease example, a consumer lease
## is available at slightly different terms. OAC for California residents only.
## To change the terms or adjust miles, go to our lease calculator for this car.
## Amazon leasing services 1100 dealers and is not a dealership. We only re-market
## a very few, very select cars at lease termination.
```