STAT 33B Homework 3

Yuanrui Zhu (3034615728)

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:

```
vehicles = readRDS("2020.09_cl_vehicles.rds")
# Q1 number of advertisements
nrow(vehicles)
```

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

```
# Q2 columns that aren't factors
lapply(vehicles, is.factor)
```

```
## $title
## [1] FALSE
##
## $text
## [1] FALSE
##
## $latitude
## [1] FALSE
##
## $longitude
## [1] FALSE
```

```
##
## $city_text
## [1] TRUE
##
## $date_posted
## [1] FALSE
##
## $date_updated
## [1] FALSE
##
## $price
## [1] FALSE
##
## $vin
## [1] FALSE
##
## $condition
## [1] TRUE
##
## $drive
## [1] TRUE
##
## $fuel
## [1] TRUE
##
## $odometer
## [1] FALSE
##
## $transmission
## [1] TRUE
##
## $type
## [1] TRUE
##
## $year
## [1] FALSE
##
## $make
## [1] FALSE
##
## $model
## [1] FALSE
##
## $fname
## [1] FALSE
##
## $craigslist
## [1] TRUE
##
## $place
## [1] FALSE
##
## $city
## [1] FALSE
```

```
##
## $state
## [1] FALSE
##
## $county
## [1] FALSE
```

```
# from the result we can see that "city text", "condition", "drive", "fuel", "transmission",
# "type", "craigslist" are factors
summary(vehicles)
```

```
##     title               text              latitude        longitude
##  Length:14990       Length:14990       Min.   :-25.07   Min.   :-157.9
##  Class :character   Class :character   1st Qu.: 37.36   1st Qu.:-122.3
##  Mode  :character   Mode  :character   Median : 37.64   Median :-122.0
##                                        Mean   : 37.56   Mean   :-119.7
##                                        3rd Qu.: 37.98   3rd Qu.:-121.8
##                                        Max.   : 48.94   Max.   : 117.1
##                                        NA's   :355      NA's   :355
##                             city_text      date_posted
##  TOUCHLESS DELIVERY TO YOUR HOME:  913   Min.   :2020-09-03 20:35:03
##  redwood city                   :  359   1st Qu.:2020-09-19 10:36:46
##  san jose west                  :  344   Median :2020-09-24 11:03:35
##  san mateo                      :  284   Mean   :2020-09-22 16:12:12
##  santa rosa                     :  279   3rd Qu.:2020-09-26 09:12:14
##  (Other)                        :12232   Max.   :2020-09-28 23:13:59
##  NA's                           :  579
##   date_updated                     price            vin
##  Min.   :2020-09-04 00:14:20   Min.   :     0   Length:14990
##  1st Qu.:2020-09-22 20:49:51   1st Qu.:  8900   Class :character
##  Median :2020-09-25 19:43:35   Median : 16000   Mode  :character
##  Mean   :2020-09-24 12:54:43   Mean   : 19572
##  3rd Qu.:2020-09-27 14:31:02   3rd Qu.: 25000
##  Max.   :2020-09-28 23:44:31   Max.   :539995
##  NA's   :8723                  NA's   :928
##      condition     drive            fuel          odometer
##  excellent:5388   4wd :4229   diesel  :  645   Min.   :      0
##  fair     : 203   fwd :5234   electric:  265   1st Qu.:  28900
##  good     :3111   rwd :2883   gas     :12480   Median :  56980
##  like new : 916   NA's:2644   hybrid  :  716   Mean   :  74573
##  new      : 158               other   :  882   3rd Qu.: 100903
##  salvage  :  28               NA's    :    2   Max.   :9999999
##  NA's     :5186                                NA's   :1631
##     transmission          type           year          make
##  automatic:12935   sedan    :3967   Min.   :1921   Length:14990
##  manual   : 1107   suv      :2970   1st Qu.:2009   Class :character
##  other    :  869   hatchback:1201   Median :2015   Mode  :character
##  NA's     :   79   truck    :1005   Mean   :2011
##                    coupe    : 980   3rd Qu.:2017
##                    (Other)  :3447   Max.   :2021
##                    NA's     :1420   NA's   :1
##     model              fname               craigslist      place
##  Length:14990       Length:14990       sfbay_eby:2997   Length:14990
##  Class :character   Class :character   sfbay_nby:2998   Class :character
```

```
##  Mode  :character   Mode  :character   sfbay_pen:3000   Mode  :character
##                                        sfbay_sby:2998
##                                        sfbay_sfc:2997
##
##
##      city              state              county
##  Length:14990       Length:14990       Length:14990
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
```

```
str(vehicles)
```

```
## 'data.frame':    14990 obs. of  24 variables:
##  $ title       : chr  "1963 Valiant convertable slant 6/auto. - $4,750 (oakland west)" " 1966 Chevrol
##  $ text        : chr  "QR Code Link to This Post\n            \n        \n1963 *** Plymouth Valiant (
##  $ latitude    : num  NA 37.9 37.9 37.9 NA ...
##  $ longitude   : num  NA -122 -122 -122 NA ...
##  $ city_text   : Factor w/ 2015 levels " Acura of Fremont :  ) (Credit Challenge Call Now!!",..: 383
##  $ date_posted : POSIXlt, format: "2020-09-24 10:40:17" "2020-09-24 15:15:07" ...
##  $ date_updated: POSIXlt, format: "2020-09-24 10:40:18" NA ...
##  $ price       : num  4750 93000 28000 84000 4495 ...
##  $ vin         : chr  "1432552546" "136176Z147141" "WBSHD9317MBK05527" "WP0AA2991TS321164" ...
##  $ condition   : Factor w/ 6 levels "excellent","fair",..: 2 1 1 1 1 1 3 3 1 1 ...
##  $ drive       : Factor w/ 3 levels "4wd","fwd","rwd": 3 3 3 1 2 3 1 3 2 2 ...
##  $ fuel        : Factor w/ 5 levels "diesel","electric",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ odometer    : num  63000 1762 88462 78087 131000 ...
##  $ transmission: Factor w/ 3 levels "automatic","manual",..: 1 2 2 1 1 1 1 1 2 2 ...
##  $ type        : Factor w/ 13 levels "bus","convertible",..: 2 3 9 3 13 10 10 9 3 4 ...
##  $ year        : int  1963 1966 1991 1996 1998 1999 1999 2001 2002 2002 ...
##  $ make        : chr  "plymouth" "chevrolet" "bmw" "porsche" ...
##  $ model       : chr  "valiant" "chevelle" "m5" "911 carrera s" ...
##  $ fname       : chr  "data_vehicles//sfbay_eby/_eby_ctd_d_1963-valiant-convertable-slant-6-auto_720
##  $ craigslist  : Factor w/ 5 levels "sfbay_eby","sfbay_nby",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ place       : chr  NA "Walnut Creek" "Walnut Creek" "Walnut Creek" ...
##  $ city        : chr  NA "Walnut Creek" "Walnut Creek" "Walnut Creek" ...
##  $ state       : chr  NA "CA" "CA" "CA" ...
##  $ county      : chr  NA "Contra Costa" "Contra Costa" "Contra Costa" ...
```

```
# from the summary we can see that "city text", "condition", "drive", "fuel",
# "transmission", "type", "craigslist", "make", "model", "place", "city",
# "state", "country", "year" are categorical
# as a result, "make", "model", "place", "city", "state", "country", "year" are categorical but not fac
# convert the above to factors
cols = c("make", "model", "place", "city", "state", "county")
vehicles[cols] = lapply(vehicles[cols], as.factor)

#Q3 missing values
colSums(is.na(vehicles)) / nrow(vehicles)
```

```
##        title         text     latitude    longitude    city_text  date_posted
```

```
## 0.000000e+00 0.000000e+00 2.368245e-02 2.368245e-02 3.862575e-02 0.000000e+00
## date_updated        price          vin    condition        drive         fuel
## 5.819213e-01 6.190794e-02 3.610407e-01 3.459640e-01 1.763843e-01 1.334223e-04
##     odometer transmission         type         year         make        model
## 1.088059e-01 5.270180e-03 9.472982e-02 6.671114e-05 3.308873e-02 3.308873e-02
##        fname   craigslist        place         city        state       county
## 0.000000e+00 0.000000e+00 1.466311e-01 2.197465e-01 2.448299e-02 2.448299e-02
```

```
# from the table I created, I can see that "date_updated", "vin", "condition",
#"city" all have a lot of missing values,
# with "a lot" being defined as more than 20% of the total data missing
```

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.

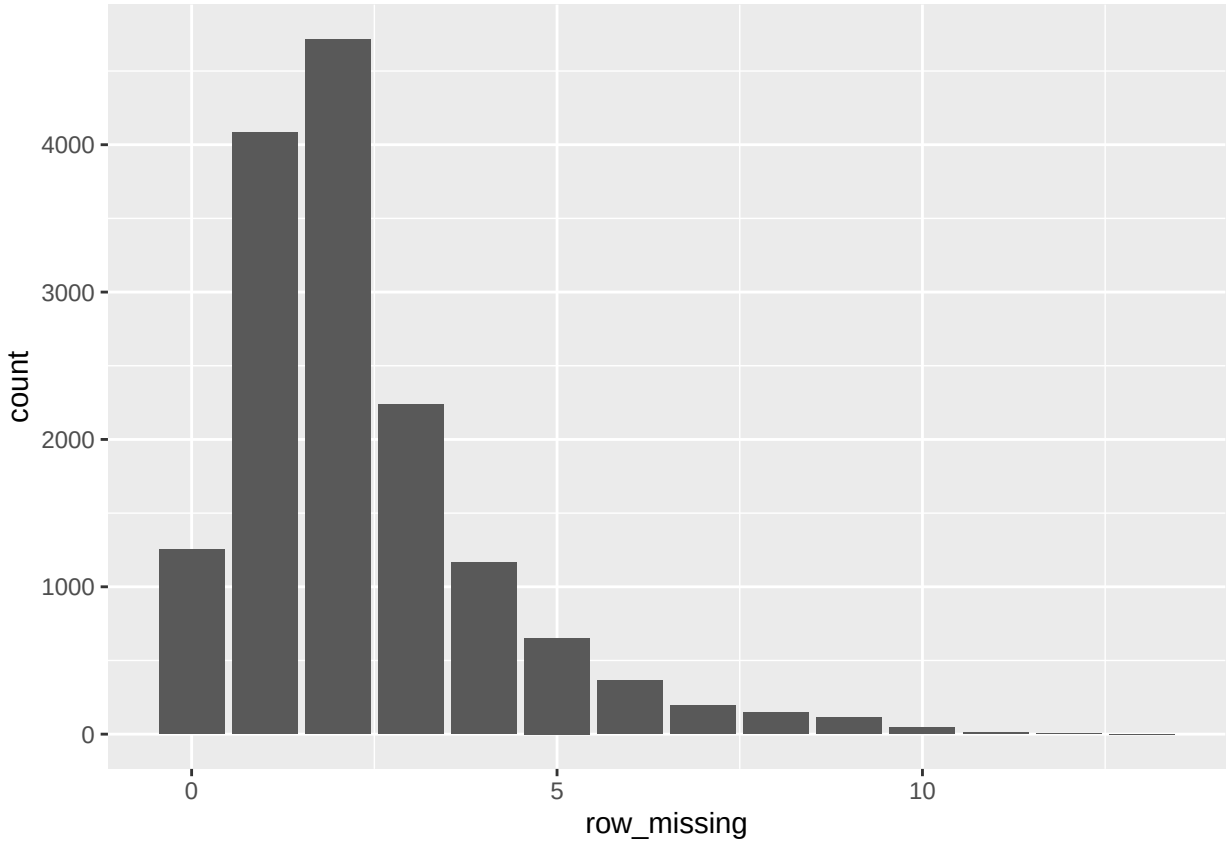
   Hint: You can create a data frame with the `data.frame()` function.

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:

```
#Q1
row_missing = rowSums(is.na(vehicles))

#Q2
#install.packages("ggplot2")
library(ggplot2)
ggplot(data.frame(row_missing), aes(x = row_missing)) + geom_bar()
```




```
#Q3
# from the bar chart we can see the majority of missing values lies around 2 and 3,
# so in terms of the total number of columns (number of elements within a row),
# a row tend to have only a few 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.

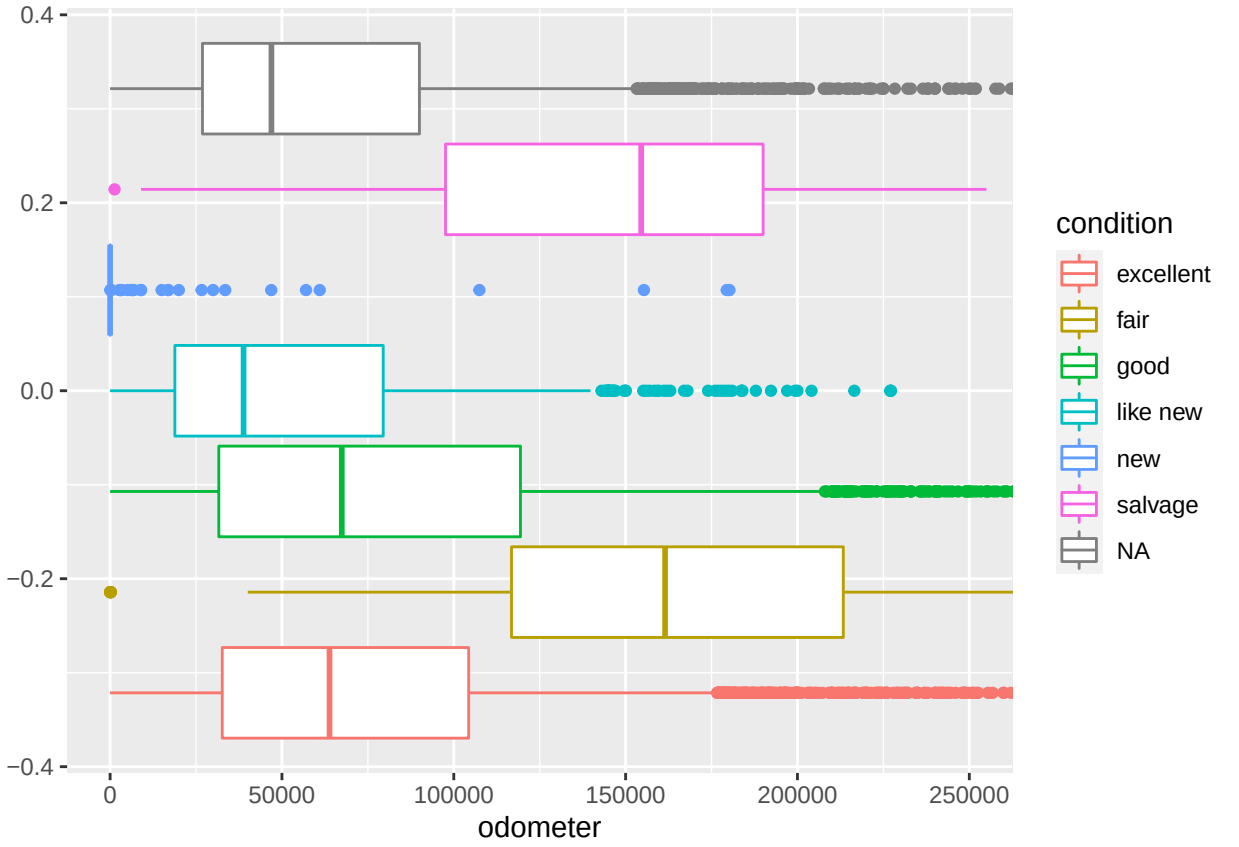
Comment briefly on the distribution of odometer readings for the various conditions.

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:

```
ggplot(vehicles, aes(x = odometer, color = condition)) + geom_boxplot(coef = 1) + coord_cartesian(xlim =
```

```
## Warning: Removed 1631 rows containing non-finite values (stat_boxplot).
```




```
# fair and salvage cars has a higher overall odometer in terms of median,
# max and span, while good, like new and excellent cars has lower overall odometer.
# New cars, not surprisingly, has minimal overall odometer (close to zero).
```

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:

```
#number of advertisements for vehicles in Berkeley is 73
nrow(subset(vehicles, city == "Berkeley"))
```

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

```r
#the number of cars of each type
summary(subset(vehicles, city == "Berkeley")['type'])
```

```
##           type
##  sedan      :28
##  suv        :12
##  hatchback  :11
##  van        : 3
##  convertible: 2
##  (Other)    : 7
##  NA's       :10
```

```r
#we can see that "sedan" is the most common

berkeley = subset(vehicles, city == "Berkeley")
tapply(berkeley$price, berkeley$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
```

```r
#we can see that pickup has the highest median, and bus has the lowest median
```

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.

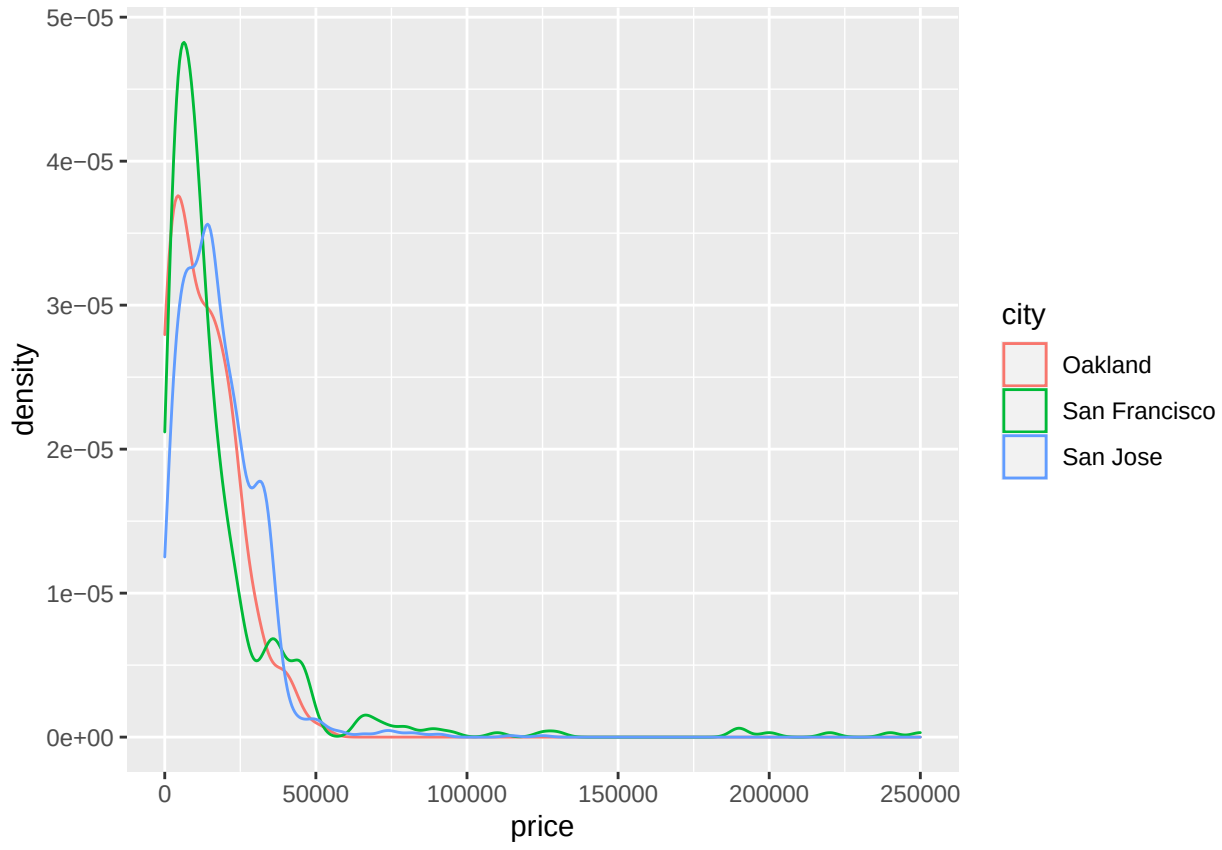
   Hint 3: The PDFLaTeX program that RMarkdown uses to knit PDFs only supports ASCII characters. Many of the advertisments contain non-ASCII characters. If you get a knit error like `! Package inputenc Error: Unicode character`, you probably printed an ad with non-ASCII characters.

   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:

```
three_city = subset(vehicles, city == "Oakland"| city == "San Jose"| city == "San Francisco")
ggplot(three_city, aes(x = price, color = city)) + geom_density()
```

```
## Warning: Removed 117 rows containing non-finite values (stat_density).
```




```
#the overall price of San Francisco and Oakland are lower
# (as density for smaller prices are higher),
# and the price for San Jose are higher (shown by the small climax around 40000)
# As the graph shows, San Francisco has some extreme prices
# (demonstrated by small bulges for extreme high prices)

high_price = subset(three_city, price > 1e5)
high_price['title']
```

```
##                                                                                  title
## 7764      2013 Rolls Royce Ghost EWB - 650 Score? WE CARRY CONTRACTS - $129,995 (mountain view)
## 7765  2014 Aston Martin Vanquish Cpe - 650 Score? WE CARRY CONTRACTS - $109,999 (mountain view)
## 7766   2014 Ferrari 458 Italia Spider - 650 Score WE CARRY CONTRACTS - $190,000 (mountain view)
## 7767             2014 Ferrari F12 Cpe - 650 Score WE CARRY CONTRACTS - $219,995 (mountain view)
## 7768         2015 Ferrari 458 Spider - 650 Score? WE CARRY CONTRACTS - $199,999 (mountain view)
## 7769      2016 Rolls Royce Wraith Cpe - 650 Score WE CARRY CONTRACTS - $189,999 (mountain view)
## 7770             2018 Ferrari 488 GTB - 650 Score WE CARRY CONTRACTS - $239,999 (mountain view)
## 7771             2019 Ferrari 488 GTB - 650 Score WE CARRY CONTRACTS - $249,995 (mountain view)
## 10150         2014 Ford Mustang Shelby GT500 Super Snake Convertible Signature Editi - $125,000
```

```
## 11807                                    2015 911 Turbo S Cabriolet - $114,500 (san jose north)
## 14849                  An exquisite Aston Martin Vanquish 2014 - $124,995 (richmond / seacliff)
```

```
high_price['text']
```

```
##
## 7764
## 7765
## 7766
## 7767
## 7768
## 7769
## 7770
## 7771
## 10150 QR Code Link to This Post\n            \n        \n2014 Ford Mustang Shelby Super Snake Conver
## 11807
## 14849
Vanquish Car 2014\nThis gorgeous 2014 Aston Martin Vanquish Coupe is for sale. Skyfall Silver with White
```

```
# the extreme high prices are reasonable as the cars are luxury cars
# (such as Ferrari, Rolls Royce, Aston Martin etc).
# Moreover, the cars are all in good conditions, which makes it reasonable for high prices.
```