COMPSCIX 415.2 Homework 5/Midterm

Vishnu Vardhan

3/4/2018

Contents

Link to Github

1. RStudio and R Markdown (3 points)

Q.1

Use markdown headers in your document to clearly separate each midterm question and add a table of contents to your document.

Answer

Assignment submitted in the requested format.

2. The tidyverse packages (3 points)

Q.1

Can you name which package is associated with each task below?

- Plotting
- Data munging/wrangling
- Reshaping (speading and gathering) data
- Importing/exporting data

Answer

- Plotting - ggplot
- Data munging/wrangling - dplry
- Reshaping - tidyr, a part of the tidyverse universe
- Importing / exporting - This is spread accross the utils and the base packages

Q.2

Now can you name two functions that you've used from each package that you listed above for these tasks?

- Plotting
- Data munging/wrangling
- Reshaping data
- Importing/exporting data (note that readRDS and saveRDS are base R functions) -

Answer

- Plotting - ggplot
- geom_point
- geom_boxplot
- Data munging/wrangling - dplry
- select

- filter
- Reshaping - tidyr, a part of the tidyverse universe
- spread
- gather
- Importing / exporting - This is spread accross the utils and the base packages
- read.csv
- saveRDS

3. R Basics (1.5 points)

Q.1

Fix this code with the fewest number of changes possible so it works:

My_data.name___is.too00ooLong! <- c(1 , 2 , 3)

Answer

```
My_data.name___is.too00ooLong <- c( 1 , 2 , 3 )
My_data.name___is.too00ooLong
```

```
## [1] 1 2 3
```

Q.2

Fix this code so it works: my_string <- C('has', 'an', 'error', 'in', 'it)

Answer

```
my_string <- c('has', 'an', 'error', 'in', 'it')
my_string
```

```
## [1] "has"   "an"    "error" "in"    "it"
```

Q.3

Look at the code below and comment on what happened to the values in the vector.

my_vector <- c(1, 2, '3', '4', 5) my_vector

[1] "1" "2" "3" "4" "5"

Answer

Rephrasing the help in my own words.

c is a generic function that combines all its arguments into a vector or a list, coercing all the elements to a common type.

The output type is determined from the highest type in the following hierarchy:

NULL < raw < logical < double < complex < character < list < expression

In this case, because character is larger than any of the numeric options, the entire vector is a vector of characters.

4. Data import/export (3 points)

Q.1

Download the rail_trail.txt file from Canvas (in the Midterm Exam section here) and successfully import it into R. Prove that it was imported successfully by including your import code and taking a glimpse of the result.

Answer

```
rail_trail <- read.delim('rail_trail.txt', header = TRUE, sep = "|")
glimpse(rail_trail)
```

```
## Observations: 90
## Variables: 10
## $ hightemp   <int> 83, 73, 74, 95, 44, 69, 66, 66, 80, 79, 78, 65, 41,...
## $ lowtemp    <int> 50, 49, 52, 61, 52, 54, 39, 38, 55, 45, 55, 48, 49,...
## $ avgtemp    <dbl> 66.5, 61.0, 63.0, 78.0, 48.0, 61.5, 52.5, 52.0, 67....
## $ spring     <int> 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, ...
## $ summer     <int> 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, ...
## $ fall       <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, ...
## $ cloudcover <dbl> 7.6, 6.3, 7.5, 2.6, 10.0, 6.6, 2.4, 0.0, 3.8, 4.1, ...
## $ precip     <dbl> 0.00, 0.29, 0.32, 0.00, 0.14, 0.02, 0.00, 0.00, 0.0...
## $ volume     <int> 501, 419, 397, 385, 200, 375, 417, 629, 533, 547, 4...
## $ weekday    <int> 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, ...
```

Q.2

Export the file into an R-specific format and name it "rail_trail.rds". Make sure you define the path correctly so that you know where it gets saved. Then reload the file. Include your export and import code and take another glimpse.

Answer

```
saveRDS(rail_trail, "rail_trail.rds")
new_rail_trail <- readRDS("rail_trail.rds")
glimpse(new_rail_trail)
```

```
## Observations: 90
## Variables: 10
## $ hightemp   <int> 83, 73, 74, 95, 44, 69, 66, 66, 80, 79, 78, 65, 41,...
## $ lowtemp    <int> 50, 49, 52, 61, 52, 54, 39, 38, 55, 45, 55, 48, 49,...
## $ avgtemp    <dbl> 66.5, 61.0, 63.0, 78.0, 48.0, 61.5, 52.5, 52.0, 67....
## $ spring     <int> 0, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 0, 1, 0, ...
## $ summer     <int> 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, ...
## $ fall       <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, ...
## $ cloudcover <dbl> 7.6, 6.3, 7.5, 2.6, 10.0, 6.6, 2.4, 0.0, 3.8, 4.1, ...
## $ precip     <dbl> 0.00, 0.29, 0.32, 0.00, 0.14, 0.02, 0.00, 0.00, 0.0...
## $ volume     <int> 501, 419, 397, 385, 200, 375, 417, 629, 533, 547, 4...
## $ weekday    <int> 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, ...
```

5. Visualization (6 points)

Q.1

Critique this graphic: give only three examples of what is wrong with this graphic. Be concise.

Answer

1. The diagram uses ‘area of a circle’ to distinguish between the sizes of each segment. Bar graphs would be better, since they are single dimensional and easier to understand.
2. These are two seperate charts, but they look like one. The first chart is a chart with three ranges (<45, 45 to 64, and >64), the second chart is a men vs women chart. This simple difference is not easily visible with how it is layed out currently.
3. With the way the data is currently layed out it is not clear that yes/no data points are proportions. Though it says proportions in the title, it should visually be represented (and the numbers should show %)

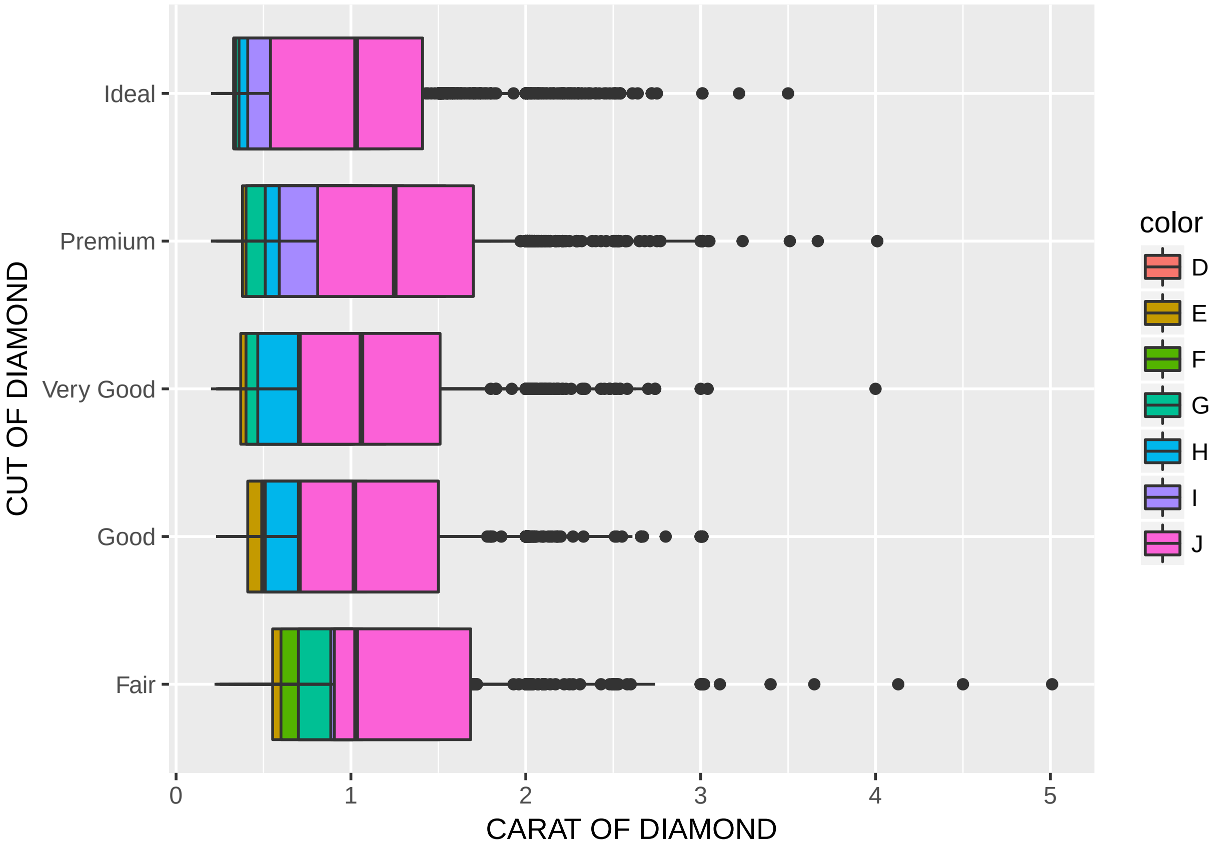
Q.2

Reproduce this graphic using the diamonds data set.

Answer

I was not able to get an exact replica, since it was not clear to me, what was used to re-order the boxplots. I used the lower of the inter-quartile range to get as close as i could.

```
ggplot(data = diamonds) +
  geom_boxplot(mapping = aes(x = reorder(cut, carat, function(x){quantile(x)[2] * -1}),
                             y = carat, fill = color),
               position = "identity") +
  labs(x = "CUT OF DIAMOND", y = "CARAT OF DIAMOND") +
  coord_flip()
```



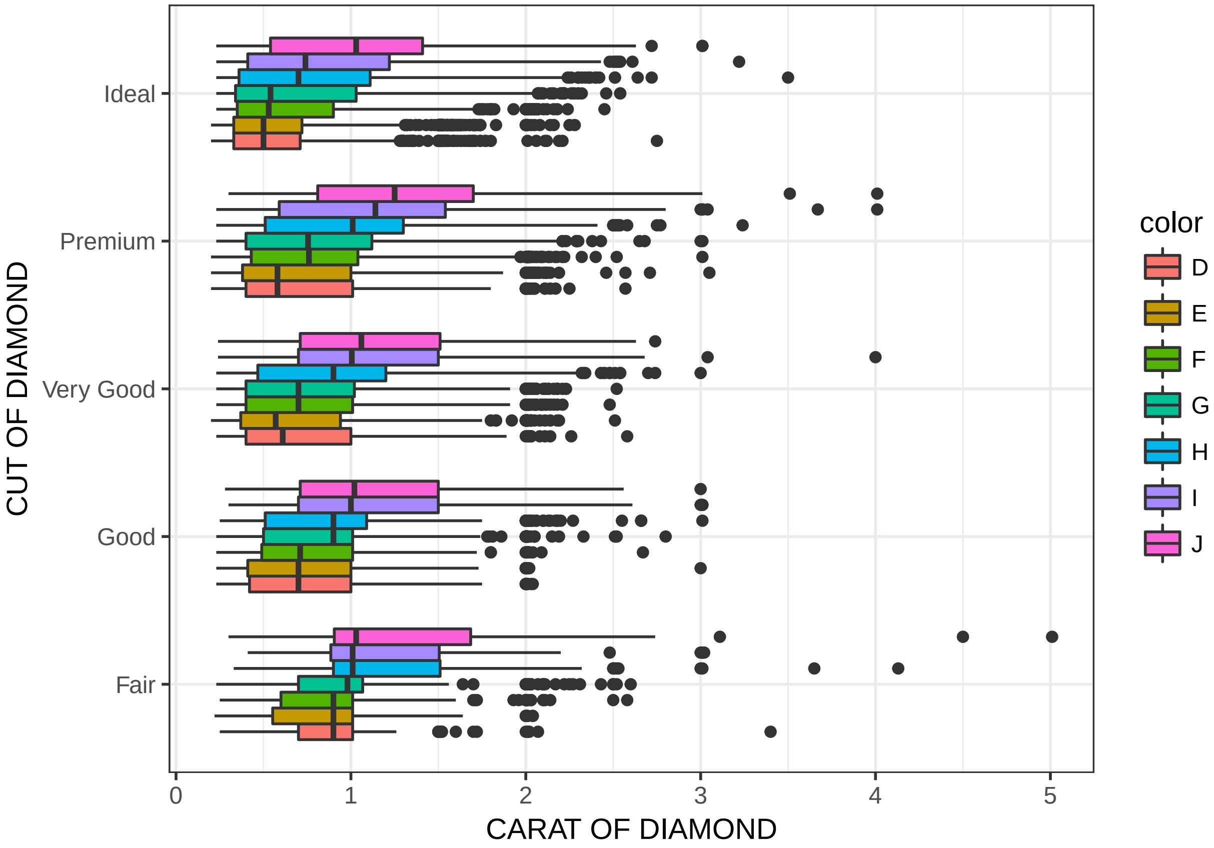
Q.3

The previous graphic is not very useful. We can make it much more useful by changing one thing about it. Make the change and plot it again.

Answer

I changed the position of the boxplots. So, that all the boxplots are visible, versus being overlapped.

```
ggplot(data = diamonds) +
  geom_boxplot(mapping = aes(x = reorder(cut, carat, function(x){quantile(x)[2] * -1}),
                             y = carat, fill = color)) +
  labs(x = "CUT OF DIAMOND", y = "CARAT OF DIAMOND") +
  coord_flip() +
  theme_bw()
```



6. Data munging and wrangling (6 points)

Q.1

Is this data “tidy”? If yes, leave it alone and go to the next problem. If no, make it tidy. Note: this data set is called table2 and is available in the tidyverse package. It should be ready for you to use after you've loaded the tidyverse package.

```
table2
```

```
## # A tibble: 12 x 4
##        country  year       type      count
##          <chr> <int>      <chr>      <int>
##  1 Afghanistan  1999      cases        745
##  2 Afghanistan  1999 population   19987071
##  3 Afghanistan  2000      cases       2666
##  4 Afghanistan  2000 population   20595360
##  5      Brazil  1999      cases      37737
##  6      Brazil  1999 population  172006362
##  7      Brazil  2000      cases      80488
##  8      Brazil  2000 population  174504898
##  9       China  1999      cases     212258
## 10       China  1999 population 1272915272
## 11       China  2000      cases     213766
## 12       China  2000 population 1280428583
```

Answer

This data is not tidy.

Take the example of Afghanistan. For one observation in the year 1999, we have two rows, one for cases and one for population. To make this data tidy, there needs to be one observation per row, which we can achieve with a "spread".

```r
table2 %>% spread(type,count)
```

```
## # A tibble: 6 x 4
##       country  year  cases population
## *       <chr> <int>  <int>      <int>
## 1 Afghanistan  1999    745   19987071
## 2 Afghanistan  2000   2666   20595360
## 3      Brazil  1999  37737  172006362
## 4      Brazil  2000  80488  174504898
## 5       China  1999 212258 1272915272
## 6       China  2000 213766 1280428583
```

Q.2

Create a new column in the diamonds data set called price_per_carat that shows the price of each diamond per carat (hint: divide). Only show me the code, not the output.

Answer

```r
diamonds %>% mutate(price_per_carat = price / carat)
```

Q.3

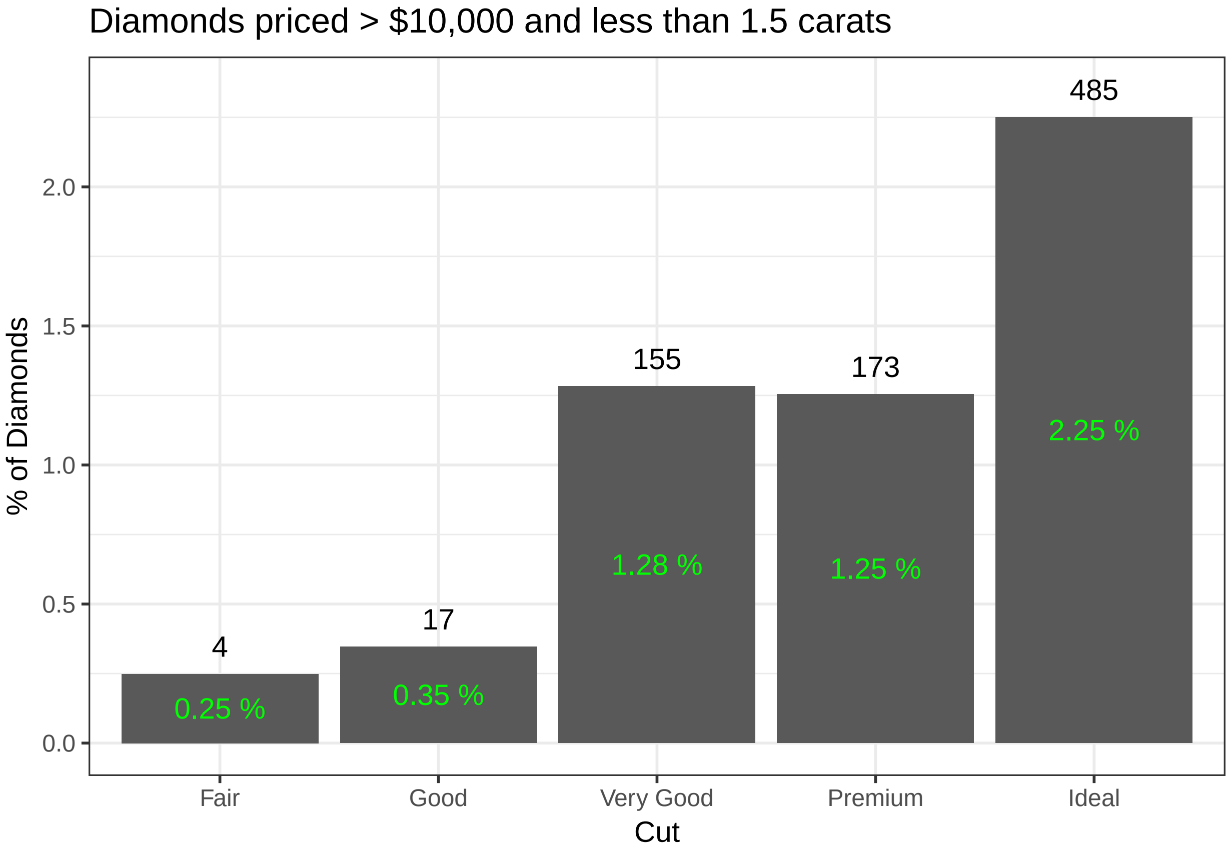
For each cut of diamond in the diamonds data set, how many diamonds, and what proportion, have a price > 10000 and a carat < 1.5? There are several ways to get to an answer, but your solution must use the data wrangling verbs from the tidyverse in order to get credit.

- Do the results make sense? Why?
- Do we need to be wary of any of these numbers? Why?

Answer

```r
diamond_target <- diamonds %>%
  mutate (target_segment = (price > 10000 & carat < 1.5)) %>%
  group_by(cut) %>%
  summarise( target_propotion = (sum(target_segment)/length(target_segment))*100,
             target_count     = sum(target_segment))

ggplot(data = diamond_target, mapping = aes(x = cut, y = target_propotion)) +
  geom_bar(stat = "identity") +
  geom_text(aes(y= target_propotion + 0.1,label = target_count)) +
  geom_text(aes(y= target_propotion/2,
                label = paste("paste(",round(target_propotion,2),",\" %\")")),
            parse = TRUE, color = "green") +
  labs(title = "Diamonds priced > $10,000 and less than 1.5 carats", x = "Cut",
     y = "% of Diamonds") +
  theme_bw()
```



As illustrated in the table above, there are 485 ideal diamonds, and they comprise 2.25% of all ideal diamonds. This makes sense, since as the diamon is more ideal, small diamonds are more expensive. Similarly, most fair diamonds won't have the same price as any of the others.

It is interesting that very-good and premium diamonds are the same. Which implies that we are missing some other parameter, likely clarity, colour or some such variable.

ps: I referred to this stack overflow thread to get the geom_text labels right. https://stackoverflow.com/questions/4408414/cannot-concatenate-more-than-3-elements-in-an-expression-for-ggplot2s-geom-text

7. EDA (6 points)

Take a look at the txhousing data set that is included with the ggplot2 package and answer these questions:

Q.1

During what time period is this data from?

Answer

The data is from Jan/2000 to July/2015

```r
txhousing %>% arrange(year,month)
```

```
## # A tibble: 8,602 x 9
##                     city  year month sales    volume median listings
##                    <chr> <int> <int> <dbl>     <dbl>  <dbl>    <dbl>
##  1               Abilene  2000     1    72   5380000  71400      701
##  2              Amarillo  2000     1   102   8860000  80000      972
##  3             Arlington  2000     1   241  26220683  94000     1417
##  4                Austin  2000     1  1025 173053635 133700     3084
##  5              Bay Area  2000     1   244  29322659 100700     1766
##  6              Beaumont  2000     1    97  10100000  82100      876
##  7       Brazoria County  2000     1    55   5245000  74400      512
##  8           Brownsville  2000     1    NA        NA     NA      400
##  9 Bryan-College Station  2000     1    61   5615000  77900      498
## 10         Collin County  2000     1   464  94788821 158700     2844
## # ... with 8,592 more rows, and 2 more variables: inventory <dbl>,
## #   date <dbl>
```

```r
txhousing %>% arrange(desc(year), desc(month))
```

```
## # A tibble: 8,602 x 9
##                     city  year month sales     volume median listings
##                    <chr> <int> <int> <dbl>      <dbl>  <dbl>    <dbl>
##  1               Abilene  2015     7   268   45845730 148700      986
##  2              Amarillo  2015     7   354   62261916 149700     1247
##  3             Arlington  2015     7   605  125495239 178900      752
##  4                Austin  2015     7  3466 1150381553 264600     7913
##  5              Bay Area  2015     7   849  197368370 200800     2144
##  6              Beaumont  2015     7   318   52882965 139300     1561
##  7       Brazoria County  2015     7    NA         NA     NA       NA
##  8           Brownsville  2015     7    NA         NA     NA       NA
##  9 Bryan-College Station  2015     7   414   90432362 190700      894
## 10         Collin County  2015     7  1861  613669702 292600     2809
## # ... with 8,592 more rows, and 2 more variables: inventory <dbl>,
## #   date <dbl>
```

Q.2

How many cities are represented?

Answer

46 Cities are represented

```r
txhousing %>% select(city) %>% unique() %>% count()
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1    46
```

Q.3

Which city, month and year had the highest number of sales?

Answer

Houston, in July/2015 had sales volume of $ 2.568 B

```
txhousing %>% arrange(desc(volume)) %>% top_n(1,volume)
```

```
## # A tibble: 1 x 9
##      city  year month sales     volume median listings inventory   date
##     <chr> <int> <int> <dbl>      <dbl>  <dbl>    <dbl>     <dbl>  <dbl>
## 1 Houston  2015     7  8945 2568156780 217600    23875       3.4 2015.5
```

Q.4

What kind of relationship do you think exists between the number of listings and the number of sales? Check your assumption and show your work.

Answer

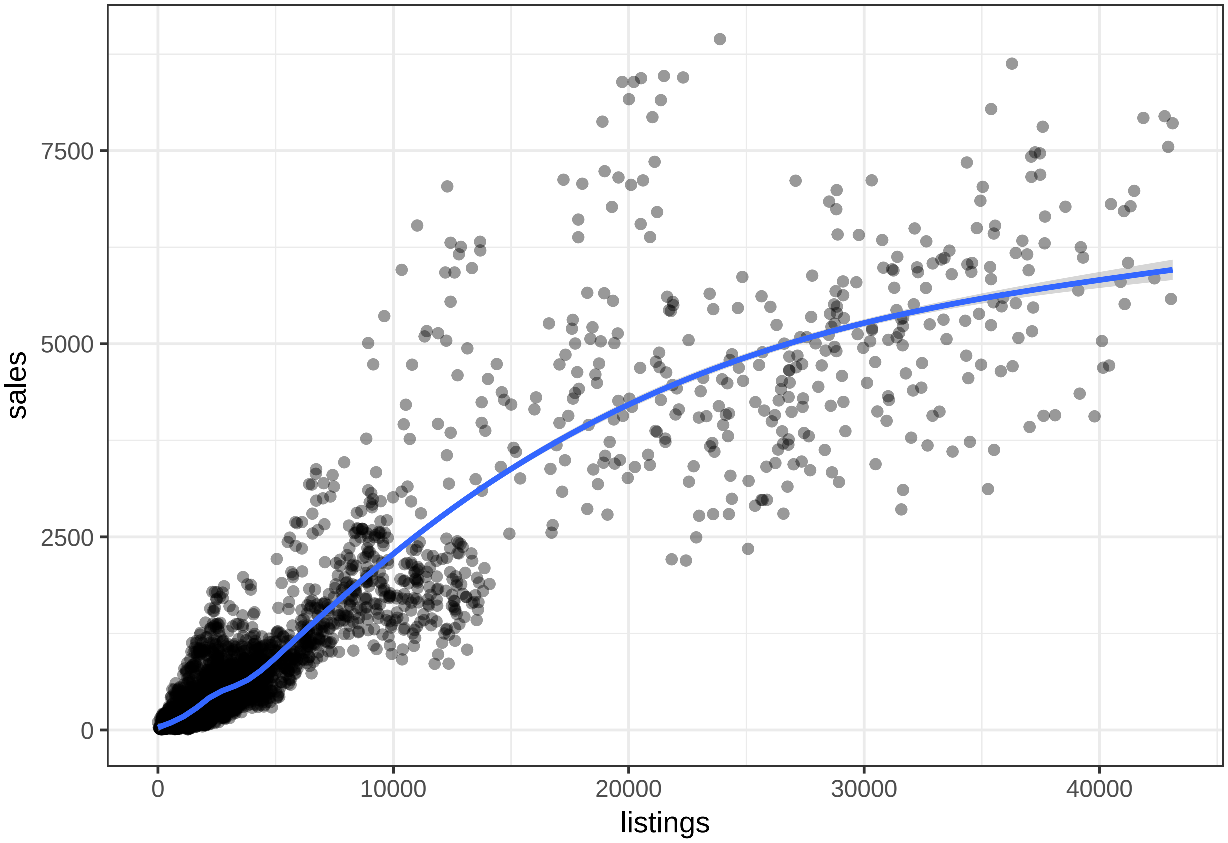
My assumption was the number of sales would increase as the listings increase, but clearly the law of diminishing returns apply.

```
ggplot(data = txhousing,mapping = aes(x=listings, y = sales)) +
  geom_point(alpha=0.4) +
  geom_smooth() +
  theme_bw()
```

```
## `geom_smooth()` using method = 'gam'

## Warning: Removed 1426 rows containing non-finite values (stat_smooth).

## Warning: Removed 1426 rows containing missing values (geom_point).
```



Q.5

What proportion of sales is missing for each city?

Answer

Proportion of sales missing is:

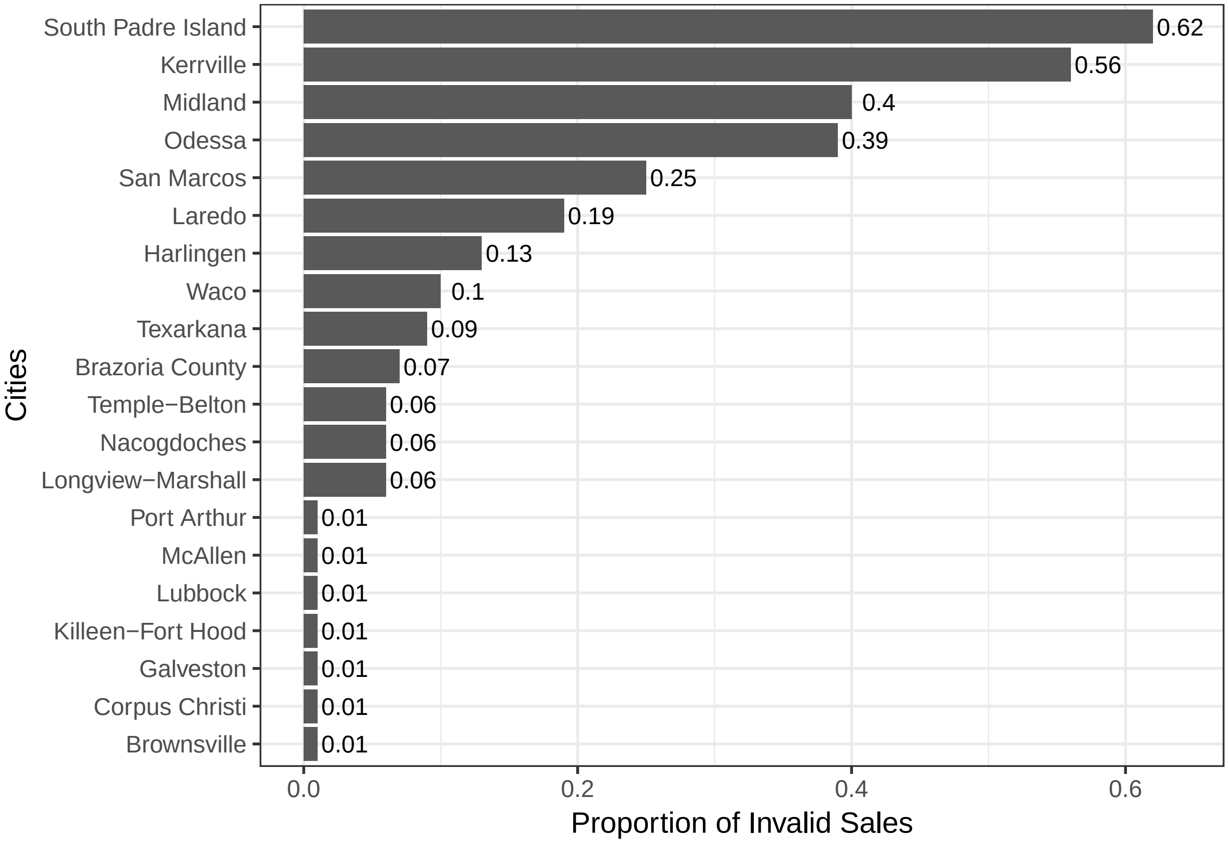
```r
t <- txhousing %>%
  mutate(valid_sales = !is.na(sales)) %>%
  group_by(city) %>%
  summarize(proportion = round(1 - sum(valid_sales)/length(valid_sales),2)) %>%
  arrange(desc(proportion))
t
```

```
## # A tibble: 46 x 2
##                  city proportion
##                 <chr>      <dbl>
##   1 South Padre Island       0.62
##   2          Kerrville       0.56
##   3            Midland       0.40
##   4             Odessa       0.39
##   5         San Marcos       0.25
##   6             Laredo       0.19
##   7          Harlingen       0.13
##   8               Waco       0.10
##  9          Texarkana       0.09
## 10    Brazoria County       0.07
## # ... with 36 more rows
```

Only plotting cities with invalid sales.

```
t <- t %>% filter(proportion > 0)

ggplot(data = t,
       mapping = aes(x=reorder(city,proportion,min),
                     y=proportion))+
  geom_col()  +
  geom_text(aes(label=round(proportion,2)), nudge_y = 0.02, size = 3) +
  labs(x = "Cities", y = "Proportion of Invalid Sales") +
  coord_flip() +
  theme_bw()
```



Q.6

Looking at only the cities and months with greater than 500 sales:

- Are the distributions of the median sales price (column name median), when grouped by city, different? The same? Show your work.
- Any cities that stand out that you'd want to investigate further?
- Why might we want to filter out all cities and months with sales less than 500?

Answer

Cities with sales less than 500, are very small in value (1/8), but very large by volume (6/7), skewing our results, and it makes sense to eliminate them.

```
txhousing %>% group_by(sales < 500) %>% summarise(sum(volume))
```

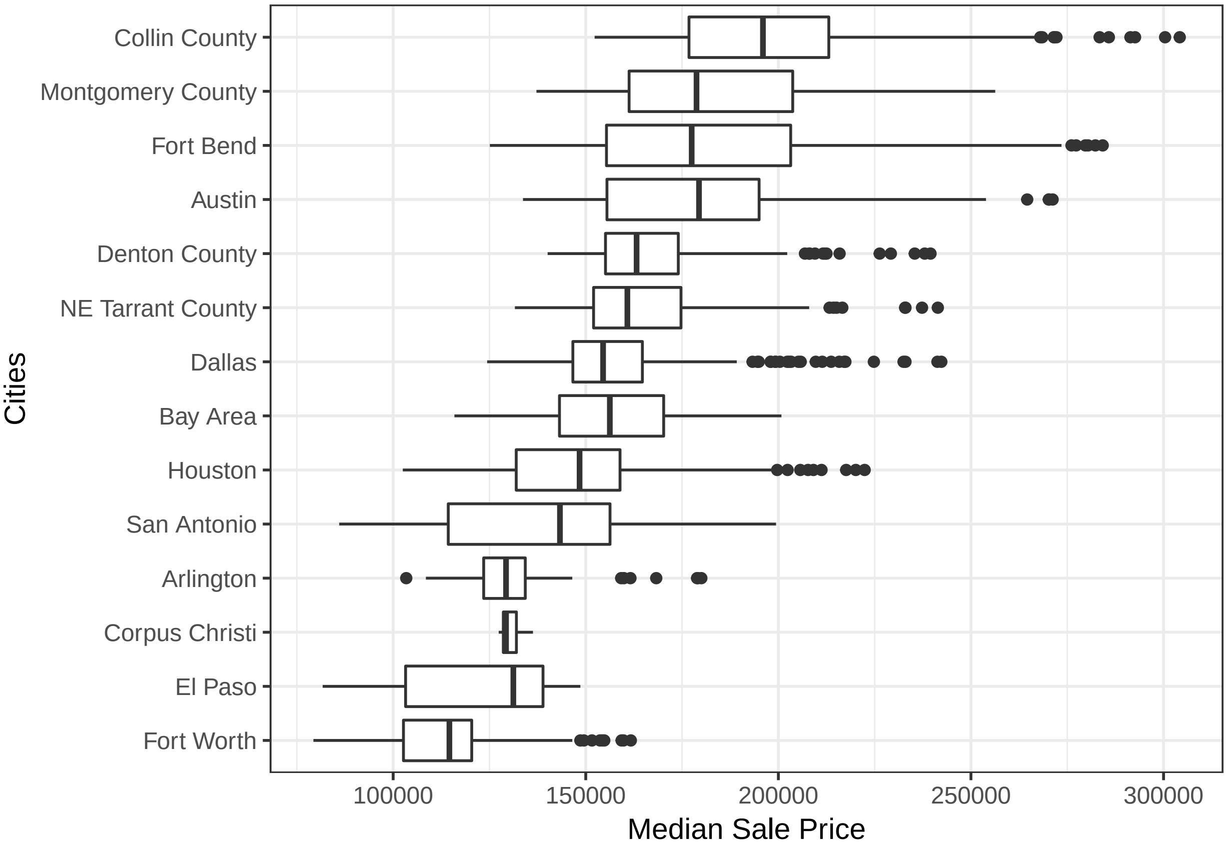
```
## # A tibble: 3 x 2
##   `sales < 500` `sum(volume)`
##           <lgl>         <dbl>
## 1         FALSE  719783334758
## 2          TRUE  138718824595
## 3            NA            NA
```

```
txhousing %>% group_by(sales < 500) %>% summarise(n())
```

```
## # A tibble: 3 x 2
##   `sales < 500` `n()`
##           <lgl> <int>
## 1         FALSE  1889
## 2          TRUE  6145
## 3            NA   568
```

Looking at the distribution shows that the median when grouped by cities is clearly different.

```
tx_500 <- txhousing %>% filter(sales > 500)
ggplot(data = tx_500, mapping = aes(x=reorder(city,median,mean), y=median)) +
  geom_boxplot() +
  labs(x = "Cities", y = "Median Sale Price") +
  coord_flip() +
  theme_bw()
```



Some of the interesting elements of the interesting cities are

1. Those were the median price is fairly high compared to the IQR
   - San Antonio
   - El Paso
   - Austin
2. Cities with lots of outliers may also beg further study
   - Denton County
   - Dallas
   - Houston

8. Git and Github (1.5 points)

Q.1

To demonstrate your use of git and Github, at the top of your document put a hyperlink to your Github repository.

Answer

Git hub link added at the top of the document.

Q.2

Once you are finished with your midterm, commit your final changes with the comment “finished the midterm-woohoo” and push your R Markdown file and your html or pdf file to Github.

Answer

done