Final Project

Leo Genders

2023-10-17

Task 0

- Complete the following tasks and ensure your completed code to each question is copied into this .Rmd file into the appropriate R chunk, and that your final file is knitted into a .pdf document as already set up in this document. Leave the labels and instructions for each task. Include your name and date under the section above in quotations.
- Important: To avoid point reductions:
    - Do not include code that you don't need outside of the assignment. This includes not printing out any complete data frames or data sets and variable values that are not explicitly asked for. Print the output asked for so that it can be verified for correctness.
    - Ensure your code requested to generate an answer is executable and not in comments. You should comment to explain your code and your answer. You may also copy the question into your R chunk and comment it out to improve readability.
    - Proof read your submission to make sure you have included all files.
    - Ensure sure your assessment has been checked for readability. Add comments where necessary to improve readability.
    - Knit this assessment from a .Rmd file to .pdf file.
    - Include required libraries and links to datasets where necessary.
    - Always refer to file locations, paths, or directories relatively by putting a datasets in a data folder in your working directory and using "data/. . . .csv" inside any read data lines of code. This means there should be no setwd() commands inside your .Rmd file.
    - Do not alter any data sets at all (e.g. remove extraneous rows or columns, change certain values manually, or rename files) prior to uploading in R.

Resources Required

- Spend a few minutes looking through the documents on Canvas and https://www.realtor.com/research/data/ before you begin. For this Final Project, you may use historical data at the state level, metro level, county level or zip level under the inventory - monthly category. Data and a data dictionary are provided to you on Canvas.
- Choose one of the following:
    - RDC_Inventory_Core_Metrics_County_History.csv,
    - RDC_Inventory_Core_Metrics_Metro_History.csv,
    - RDC_Inventory_Core_Metrics_State_History.csv,
    - RDC_Inventory_Core_Metrics_Zip_History.csv

Overall Goal

- Your goal is to examine 6 variables, including potential subgroups based on location or date, and 4 relationships between 2 variables (visually and through an appropriate hypothesis test). In preparing the data, you can create subsets of the data set to make it easier to work with based on what you need. Read all the instructions below before making decisions on how to slice down the data set. Take note that some of the variables are measuring a similar thing, so be sure to choose unique variables to have the most accurate results.

Task 1 (25 points)

- Take steps to read in and prepare the data, ensuring that there are no extra lines that should not be in the dataset and that the variables you choose are in the correct data types.
- Save a new smaller data object that only includes the variables you choose to work with throughout the project.
- If you want to eliminate rows by subsetting a smaller dataset after eliminating columns in the step above, do so here and provide the rationale based on what you want to examine.
- Code any missing values appropriately and rename variables where necessary with explanation in comments of what the new named variable represents.
- Comment on how you cleaned the data and your rationale for doing so.

```r
# Task 1: Read and prepare the data. Choosing to examine the state
# level.
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(dplyr)
library(descr)
state.data <- read.csv("RDC_Inventory_Core_Metrics_State_History.csv")

# Review the state-level data.
summary(state.data)
```

```
##  month_date_yyyymm    state              state_id          median_listing_price
##  Min.   :201607    Length:3775        Length:3775        Min.   :134450
##  1st Qu.:201801    Class :character   Class :character   1st Qu.:230029
##  Median :201907    Mode  :character   Mode  :character   Median :290358
##  Mean   :201915                                          Mean   :325053
##  3rd Qu.:202102                                          3rd Qu.:375000
##  Max.   :202208                                          Max.   :879500
##
```

```
##  median_listing_price_mm median_listing_price_yy active_listing_count
##  Min.   :-0.0981         Min.   :-0.2142         Min.   :     1
##  1st Qu.:-0.0060         1st Qu.: 0.0372         1st Qu.:  5404
##  Median : 0.0009         Median : 0.0704         Median : 11895
##  Mean   : 0.0066         Mean   : 0.0795         Mean   : 20158
##  3rd Qu.: 0.0187         3rd Qu.: 0.1135         3rd Qu.: 24603
##  Max.   : 0.2090         Max.   : 0.3986         Max.   :163956
##  NA’s   :613             NA’s   :613
##  active_listing_count_mm active_listing_count_yy median_days_on_market
##  Min.   :-0.3977         Min.   :-0.7079         Min.   :  7.00
##  1st Qu.:-0.0624         1st Qu.:-0.2822         1st Qu.: 49.00
##  Median :-0.0100         Median :-0.1215         Median : 64.00
##  Mean   :-0.0060         Mean   :-0.1457         Mean   : 66.37
##  3rd Qu.: 0.0382         3rd Qu.:-0.0220         3rd Qu.: 81.00
##  Max.   : 0.6177         Max.   : 1.3855         Max.   :210.00
##  NA’s   :613             NA’s   :613
##  median_days_on_market_mm median_days_on_market_yy new_listing_count
##  Min.   :-0.7097          Min.   :-0.7064          Min.   :    0
##  1st Qu.:-0.0678          1st Qu.:-0.1679          1st Qu.: 2150
##  Median : 0.0270          Median :-0.0769          Median : 5500
##  Mean   : 0.0039          Mean   :-0.0960          Mean   : 8797
##  3rd Qu.: 0.0909          3rd Qu.:-0.0164          3rd Qu.:11206
##  Max.   : 0.8519          Max.   : 0.6923          Max.   :52876
##  NA’s   :613              NA’s   :613
##  new_listing_count_mm new_listing_count_yy price_increased_count
##  Min.   :-0.6874      Min.   :-0.7743      Min.   :    0.0
##  1st Qu.:-0.1011      1st Qu.:-0.0727      1st Qu.:  112.0
##  Median :-0.0103      Median :-0.0032      Median :  340.0
##  Mean   : 0.0150      Mean   : 0.0002      Mean   :  811.1
##  3rd Qu.: 0.1204      3rd Qu.: 0.0611      3rd Qu.:  848.0
##  Max.   : 2.3884      Max.   : 2.8965      Max.   :10460.0
##  NA’s   :613          NA’s   :613
##  price_increased_count_mm price_increased_count_yy price_reduced_count
##  Min.   :-0.9231          Min.   :-0.9540          Min.   :    0
##  1st Qu.:-0.1625          1st Qu.:-0.3125          1st Qu.: 1056
##  Median :-0.0117          Median :-0.0454          Median : 2892
##  Mean   : 0.0623          Mean   : 0.1377          Mean   : 5555
##  3rd Qu.: 0.1714          3rd Qu.: 0.3143          3rd Qu.: 6608
##  Max.   :19.5000          Max.   : 6.7090          Max.   :59600
##  NA’s   :613              NA’s   :613
##  price_reduced_count_mm price_reduced_count_yy pending_listing_count
##  Min.   :-0.7548        Min.   :-0.8341        Min.   :    0
##  1st Qu.:-0.1045        1st Qu.:-0.3331        1st Qu.: 1668
##  Median : 0.0133        Median :-0.0850        Median : 4838
##  Mean   : 0.0236        Mean   :-0.0709        Mean   : 9219
##  3rd Qu.: 0.1494        3rd Qu.: 0.0789        3rd Qu.:11526
##  Max.   : 2.8261        Max.   : 3.9203        Max.   :84759
##  NA’s   :613            NA’s   :613            NA’s   :21
##  pending_listing_count_mm pending_listing_count_yy
##  Min.   :-1.0000          Min.   :-0.9858
##  1st Qu.:-0.0699          1st Qu.:-0.0861
##  Median :-0.0175          Median : 0.0459
##  Mean   : 0.0486          Mean   : 0.5859
##  3rd Qu.: 0.0861          3rd Qu.: 0.3232
##  Max.   :52.0903          Max.   :81.0000
##  NA's   :633              NA's   :643
##  median_listing_price_per_square_foot median_listing_price_per_square_foot_mm
##  Min.   : 78.0                        Min.   :-0.2090
##  1st Qu.:118.0                        1st Qu.:-0.0015
##  Median :146.0                        Median : 0.0049
##  Mean   :174.1                        Mean   : 0.0073
##  3rd Qu.:189.0                        3rd Qu.: 0.0147
##  Max.   :695.0                        Max.   : 0.2279
##                                       NA's   :613
##  median_listing_price_per_square_foot_yy median_square_feet
##  Min.   :-0.1432                         Min.   : 990
##  1st Qu.: 0.0432                         1st Qu.:1796
##  Median : 0.0726                         Median :1936
##  Mean   : 0.0894                         Mean   :1925
##  3rd Qu.: 0.1260                         3rd Qu.:2052
##  Max.   : 0.5369                         Max.   :2808
##  NA's   :613
##  median_square_feet_mm median_square_feet_yy average_listing_price
##  Min.   :-0.1124       Min.   :-0.3077       Min.   : 207337
##  1st Qu.:-0.0077       1st Qu.:-0.0245       1st Qu.: 312809
##  Median :-0.0013       Median :-0.0006       Median : 413261
##  Mean   :-0.0004       Mean   :-0.0063       Mean   : 517178
##  3rd Qu.: 0.0062       3rd Qu.: 0.0166       3rd Qu.: 620527
##  Max.   : 0.1921       Max.   : 0.2827       Max.   :1707319
##  NA's   :613           NA's   :613
##  average_listing_price_mm average_listing_price_yy total_listing_count
##  Min.   :-0.3395          Min.   :-0.3019          Min.   :     1
##  1st Qu.:-0.0070          1st Qu.: 0.0294          1st Qu.:  8253
##  Median : 0.0032          Median : 0.0652          Median : 18441
##  Mean   : 0.0062          Mean   : 0.0820          Mean   : 29331
##  3rd Qu.: 0.0181          3rd Qu.: 0.1164          3rd Qu.: 35516
##  Max.   : 0.5063          Max.   : 0.7601          Max.   :218268
##  NA's   :613              NA's   :613
##  total_listing_count_mm total_listing_count_yy pending_ratio
##  Min.   :-0.3557        Min.   :-0.4981        Min.   :0.0000
##  1st Qu.:-0.0437        1st Qu.:-0.1736        1st Qu.:0.1802
##  Median :-0.0002        Median :-0.0847        Median :0.3810
##  Mean   :-0.0028        Mean   :-0.0818        Mean   :0.5635
##  3rd Qu.: 0.0416        3rd Qu.:-0.0019        3rd Qu.:0.7967
##  Max.   : 0.7868        Max.   : 0.8937        Max.   :2.9593
##  NA's   :613            NA's   :613            NA's   :21
##  pending_ratio_mm  pending_ratio_yy   quality_flag
##  Min.   :-0.8885   Min.   :-1.3269   Min.   :0.0000
##  1st Qu.:-0.0247   1st Qu.:-0.0037   1st Qu.:0.0000
##  Median :-0.0005   Median : 0.0518   Median :0.0000
##  Mean   : 0.0064   Mean   : 0.1649   Mean   :0.0199
##  3rd Qu.: 0.0390   3rd Qu.: 0.2710   3rd Qu.:0.0000
##  Max.   : 1.2009   Max.   : 2.5016   Max.   :1.0000
##  NA's   :633       NA's   :640       NA's   :612
```

```
# Create a subset. Rationale: Create a subset for the states I was a
# US Army Recruiting Company Commander and experienced buying and
# selling a home within the region.  Additionally, the VA loan was a
# specific military benefit referenced during my tenure in Recruiting
# Command and enticed potential prospects to consider a career in the
# US Army. I want to examine the condition of real estate within this
# region that I served and lived in as I recruited for the Army from
# August 2020 to August 2022 and will name it the Tri-state region
# for OH, WV, KY states.  I want to examine active listings, median
# days on the market, new listings, pending listings, median square
# feet, and total listings as my six variables.  I will select only
# these relevant columns for my variables from the data set as well
# as remove the state_id because it is redundant and longer than
# state to type.
tristate.region <- state.data %>%
    filter(state == "ohio" | state == "west virginia" | state == "kentucky") %>%
    filter(month_date_yyyymm > 202007 & month_date_yyyymm < 202209) %>%
    mutate(state, state = as.factor(state)) %>%
    mutate(state_id, state_id = as.factor(state_id)) %>%
    dplyr::select("month_date_yyyymm", "state", "active_listing_count",
        "active_listing_count_mm", "active_listing_count_yy", "median_days_on_market",
        "median_listing_price", "median_listing_price_mm", "median_listing_price_yy",
        "new_listing_count", "new_listing_count_mm", "new_listing_count_yy",
        "pending_listing_count", "pending_listing_count_mm", "pending_listing_count_yy",
        "median_square_feet", "median_square_feet_mm", "median_square_feet_yy",
        "total_listing_count", "total_listing_count_mm", "total_listing_count_yy")
summary(tristate.region)
```

```
##  month_date_yyyymm           state    active_listing_count
##  Min.   :202008    kentucky      :25   Min.   : 2182
##  1st Qu.:202102    ohio          :25   1st Qu.: 3454
##  Median :202108    west virginia:25   Median : 6215
##  Mean   :202119                        Mean   : 7620
##  3rd Qu.:202202                        3rd Qu.:10328
##  Max.   :202208                        Max.   :17325
##  active_listing_count_mm active_listing_count_yy median_days_on_market
##  Min.   :-0.207100       Min.   :-0.57350        Min.   : 29.00
##  1st Qu.:-0.068400       1st Qu.:-0.47165        1st Qu.: 39.00
##  Median :-0.005800       Median :-0.29470        Median : 49.00
##  Mean   :-0.001237       Mean   :-0.26479        Mean   : 52.27
##  3rd Qu.: 0.064250       3rd Qu.:-0.07115        3rd Qu.: 60.50
##  Max.   : 0.232600       Max.   : 0.21540        Max.   :100.00
##  median_listing_price median_listing_price_mm median_listing_price_yy
##  Min.   :159450       Min.   :-0.057100       Min.   :-0.07450
##  1st Qu.:179975       1st Qu.:-0.014550       1st Qu.: 0.00280
##  Median :217000       Median : 0.000000       Median : 0.06170
##  Mean   :212110       Mean   : 0.006181       Mean   : 0.06144
##  3rd Qu.:238200       3rd Qu.: 0.022950       3rd Qu.: 0.10230
##  Max.   :284900       Max.   : 0.099300       Max.   : 0.27160
##  new_listing_count new_listing_count_mm new_listing_count_yy
##  Min.   :  916     Min.   :-0.301000    Min.   :-0.28430
##  1st Qu.: 1870     1st Qu.:-0.092900    1st Qu.:-0.06985
##  Median : 5296     Median : 0.016600    Median : 0.01820
##  Mean   : 6975     Mean   : 0.008804    Mean   : 0.01000
##  3rd Qu.:11110     3rd Qu.: 0.067750    3rd Qu.: 0.08280
##  Max.   :18252     Max.   : 0.434900    Max.   : 0.37160
##  pending_listing_count pending_listing_count_mm pending_listing_count_yy
##  Min.   : 2120         Min.   :-0.183400        Min.   :-0.25080        
##  1st Qu.: 2938         1st Qu.:-0.051900        1st Qu.:-0.07565        
##  Median : 7928         Median :-0.011900        Median :-0.01290        
##  Mean   :10240         Mean   :-0.005677        Mean   : 0.09003        
##  3rd Qu.:17331         3rd Qu.: 0.052500        3rd Qu.: 0.24300        
##  Max.   :24761         Max.   : 0.151400        Max.   : 0.66310        
##  median_square_feet median_square_feet_mm median_square_feet_yy
##  Min.   :1583       Min.   :-0.027100     Min.   :-0.09140     
##  1st Qu.:1717       1st Qu.:-0.011250     1st Qu.:-0.04940     
##  Median :1751       Median :-0.002400     Median :-0.03540     
##  Mean   :1768       Mean   :-0.002277     Mean   :-0.02964     
##  3rd Qu.:1824       3rd Qu.: 0.007550     3rd Qu.:-0.01085     
##  Max.   :2004       Max.   : 0.026400     Max.   : 0.03830     
##  total_listing_count total_listing_count_mm total_listing_count_yy
##  Min.   : 4395       Min.   :-0.174300      Min.   :-0.32810      
##  1st Qu.: 6268       1st Qu.:-0.033600      1st Qu.:-0.23110      
##  Median :14318       Median : 0.000200      Median :-0.16580      
##  Mean   :17867       Mean   :-0.006048      Mean   :-0.16379      
##  3rd Qu.:27764       3rd Qu.: 0.039900      3rd Qu.:-0.08215      
##  Max.   :41544       Max.   : 0.128900      Max.   : 0.01890      
```

```
# output indicates no issues with NA or missing values for smaller
# subset and n>30, state changed to factor data type from character,
# also removed columns not being considered for this project
```

Task 2 (30 points)

- Examine at least 6 variables from the data set including all measures of central tendency and spread that we covered in the course for continuous variables, and frequency and relative frequency for categorical variables. Make your output clear by numbering your variables 1 - 6 and use the summarize function where appropriate to have R output most of the summary measures at once.
- Make sure your R generated output is visible and provide rationale and insights on what you took away from this step in comments.
- Depending on how you sliced the data, you may only have one categorical variable out of the 6.

```
# create variable1 for active_listing_count and measure central
# tendency, spread, frequency, and relative frequency

# variable1 - Active Listings
tristate.region %>%
    summarise(mean.active.listings = mean(x = active_listing_count), sd.active.listings = sd(x = active_
        var.active.listings = var(x = active_listing_count), median.active.listings = median(x = active_
        iqr.active.listings = IQR(x = active_listing_count), quant.active.listings = quantile(x = trista
        mode.active.listings = names(x = sort(table(active_listing_count),
            decreasing = TRUE))[1])
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
##   mean.active.listings sd.active.listings var.active.listings
## 1             7620.173           4731.856            22390458
## 2             7620.173           4731.856            22390458
## 3             7620.173           4731.856            22390458
## 4             7620.173           4731.856            22390458
## 5             7620.173           4731.856            22390458
##   median.active.listings iqr.active.listings quant.active.listings
## 1                   6215              6873.5                2182.0
## 2                   6215              6873.5                3454.5
## 3                   6215              6873.5                6215.0
## 4                   6215              6873.5               10328.0
## 5                   6215              6873.5               17325.0
##   mode.active.listings
## 1                 2878
## 2                 2878
## 3                 2878
## 4                 2878
## 5                 2878
```

```
# mean for active listings for tristate region is 7620.18 sd for
# active listings for tristate region is 4731.86 variance for active
# listings for tristate region is 22390458 IQR for active listings
# for tristate region is 6873.5 quantile for active listings for
# tristate regsion is 25% = 3454.5; 75% = 10328.0 median for active
# listings for tristate region is 6215 mode for active listings for
# tristate region is 2878 active_listing_count is numeric and
# therefore does not have a B index for mode spread

# variable2 - Median Days on the Market
tristate.region %>%
    summarise(mean.med.mkt.days = mean(x = median_days_on_market), sd.med.mkt.days = sd(x = median_days
        var.med.mkt.days = var(x = median_days_on_market), median.med.mkt.days = median(x = median_days
        iqr.med.mkt.days = IQR(x = median_days_on_market), quant.med.mkt.days = quantile(x = tristate.r
        mode.med.mkt.days = names(x = sort(table(median_days_on_market),
            decreasing = TRUE))[1])
```

```
## Warning: Returning more (or less) than 1 row per ‘summarise()‘ group was deprecated in
## dplyr 1.1.0.
## i Please use ‘reframe()‘ instead.
## i When switching from ‘summarise()‘ to ‘reframe()‘, remember that ‘reframe()‘
##   always returns an ungrouped data frame and adjust accordingly.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
##   mean.med.mkt.days sd.med.mkt.days var.med.mkt.days median.med.mkt.days
## 1          52.26667        16.89981         285.6036                  49
## 2          52.26667        16.89981         285.6036                  49
## 3          52.26667        16.89981         285.6036                  49
## 4          52.26667        16.89981         285.6036                  49
## 5          52.26667        16.89981         285.6036                   49
##   iqr.med.mkt.days quant.med.mkt.days mode.med.mkt.days
## 1             21.5               29.0                37
## 2             21.5               39.0                37
## 3             21.5               49.0                37
## 4             21.5               60.5                37
## 5             21.5              100.0                37
```

```
# mean for median days on the market for tristate region is
# 52.26667\t sd for median days on the market for tristate region is
# 16.89981 variance for median days on the market for tristate region
# is 285.6036\t IQR for median days on the market for tristate region
# is 21.5 quantile for median days on the market for tristate regsion
# is 25% = 39.0; 75% = 60.5 median for median days on the market for
# tristate region is 49 mode for median days on the market for
# tristate region is 37 median_days_on_market is numeric and
# therefore does not have a B index for mode spread

# variable3 - New Listings
tristate.region %>%
    summarise(mean.new.listings = mean(x = new_listing_count), sd.new.listings = sd(x = new_listing_cou
        var.new.listings = var(x = new_listing_count), median.new.listings = median(x = new_listing_cou
        iqr.new.listings = IQR(x = new_listing_count), quant.new.listings = quantile(x = tristate.regio
        mode.new.listings = names(x = sort(table(new_listing_count), decreasing = TRUE))[1])
```

```
## Warning: Returning more (or less) than 1 row per ‘summarise()‘ group was deprecated in
## dplyr 1.1.0.
## i Please use ‘reframe()‘ instead.
## i When switching from ‘summarise()‘ to ‘reframe()‘, remember that ‘reframe()‘
##   always returns an ungrouped data frame and adjust accordingly.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
##   mean.new.listings sd.new.listings var.new.listings median.new.listings
## 1          6975.147        5621.405         31600191                5296
## 2          6975.147        5621.405         31600191                5296
## 3          6975.147        5621.405         31600191                5296
## 4          6975.147        5621.405         31600191                5296
## 5          6975.147        5621.405         31600191                5296
##   iqr.new.listings quant.new.listings mode.new.listings
## 1             9240                916              1204
## 2             9240               1870              1204
## 3             9240               5296              1204
## 4             9240              11110              1204
## 5             9240              18252              1204
```

```
# mean for new listings for tristate region is 6975.147 sd for new
# listings for tristate region 5621.405\t variance for new listings
# for tristate region is 31600191\t IQR for new listings for tristate
# region is 9240 quantile for new listings for tristate regsion is
# 25% = 1870; 75% = 11110 median for new listings for tristate region
# is 5296\t mode for new listings for tristate region is 1204
# new_listing_count is numeric and therefore does not have a B index
# for mode spread

# variable4 - Pending Listings
tristate.region %>%
    summarise(mean.pending.listings = mean(x = pending_listing_count),
        sd.pending.listings = sd(x = pending_listing_count), var.pending.listings = var(x = pending_lis
        median.pending.listings = median(x = pending_listing_count), iqr.pending.listings = IQR(x = pen
        quant.pending.listings = quantile(x = tristate.region$pending_listing_count),
        mode.pending.listings = names(x = sort(table(pending_listing_count),
            decreasing = TRUE))[1])
```

```
## Warning: Returning more (or less) than 1 row per `summarise()` group was deprecated in
## dplyr 1.1.0.
## i Please use `reframe()` instead.
## i When switching from `summarise()` to `reframe()`, remember that `reframe()`
##   always returns an ungrouped data frame and adjust accordingly.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
##   mean.pending.listings sd.pending.listings var.pending.listings
## 1              10239.71            7510.275             56404233
## 2              10239.71            7510.275             56404233
## 3              10239.71            7510.275             56404233
## 4              10239.71            7510.275             56404233
## 5              10239.71            7510.275             56404233
##   median.pending.listings iqr.pending.listings quant.pending.listings
## 1                    7928                14393                   2120
## 2                    7928                14393                   2938
## 3                    7928                14393                   7928
## 4                    7928                14393                  17331
## 5                    7928                14393                  24761
##   mode.pending.listings
## 1                  2120
## 2                  2120
## 3                  2120
## 4                  2120
## 5                  2120
```

```
# mean for pending listings for tristate region is 10239.71 sd for
# pending listings for tristate region 7510.275\t variance for
# pending listings for tristate region is 31600191\t IQR for pending
# listings for tristate region is 9240 quantile for pending listings
# for tristate regsion is 25% = 1870; 75% = 11110 median for pending
# listings for tristate region is 5296\t mode for pending listings
# for tristate region is 1204 pending_listing_count is numeric and
# therefore does not have a B index for mode spread

# variable5 - Median Square Feet
tristate.region %>%
    summarise(mean.med.sqft = mean(x = median_square_feet), sd.med.sqft = sd(x = median_square_feet),
        var.med.sqft = var(x = median_square_feet), median.med.sqft = median(x = median_square_feet),
        iqr.med.sqft = IQR(x = median_square_feet), quant.med.sqft = quantile(x = tristate.region$media
        mode.med.sqft = names(x = sort(table(median_square_feet), decreasing = TRUE))[1])
```

```
## Warning: Returning more (or less) than 1 row per ‘summarise()‘ group was deprecated in
## dplyr 1.1.0.
## i Please use ‘reframe()‘ instead.
## i When switching from ‘summarise()‘ to ‘reframe()‘, remember that ‘reframe()‘
##   always returns an ungrouped data frame and adjust accordingly.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
##   mean.med.sqft sd.med.sqft var.med.sqft median.med.sqft iqr.med.sqft
## 1       1768.16    82.39638     6789.163            1751        106.5
## 2       1768.16    82.39638     6789.163            1751        106.5
## 3       1768.16    82.39638     6789.163            1751        106.5
## 4       1768.16    82.39638     6789.163            1751        106.5
## 5       1768.16    82.39638     6789.163            1751        106.5
##   quant.med.sqft mode.med.sqft
## 1         1583.0          1732
## 2         1717.0          1732
## 3         1751.0          1732
## 4         1823.5          1732
## 5         2004.0          1732
```

```
# mean for median sqft for tristate region is 1768.16\t sd for median
# sqft for tristate region 82.39638\t\t variance for median sqft for
# tristate region is 6789.163\t IQR for median sqft for tristate
# region is 106.5 quantile for median sqft for tristate regsion is
# 25% = 1717.0; 75% = 1823.5 median for median sqft for tristate
# region is 1751\t\t mode for median sqft for tristate region is 1732
# median_square_feet is numeric and therefore does not have a B index
# for mode spread

# variable6 - Total Listings
tristate.region %>%
    summarise(mean.total.listings = mean(x = total_listing_count), sd.total.listings = sd(x = total_lis
        var.total.listings = var(x = total_listing_count), median.total.listings = median(x = total_lis
        iqr.total.listings = IQR(x = total_listing_count), quant.total.listings = quantile(x = tristate
        mode.total.listings = names(x = sort(table(total_listing_count),
            decreasing = TRUE))[1])
```

```
## Warning: Returning more (or less) than 1 row per ‘summarise()‘ group was deprecated in
## dplyr 1.1.0.
## i Please use ‘reframe()‘ instead.
## i When switching from ‘summarise()‘ to ‘reframe()‘, remember that ‘reframe()‘
##   always returns an ungrouped data frame and adjust accordingly.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
##   mean.total.listings sd.total.listings var.total.listings
## 1            17866.56          12078.49          145890007
## 2            17866.56          12078.49          145890007
## 3            17866.56          12078.49          145890007
## 4            17866.56          12078.49          145890007
## 5            17866.56          12078.49          145890007
##   median.total.listings iqr.total.listings quant.total.listings
## 1                 14318            21496.5                4395.0
## 2                 14318            21496.5                6267.5
## 3                 14318            21496.5               14318.0
## 4                 14318            21496.5               27764.0
## 5                 14318            21496.5               41544.0
##   mode.total.listings
## 1                4395
## 2                4395
## 3                4395
## 4                4395
## 5                4395
```

```
# mean for median sqft for tristate region is 1768.16\t sd for median
# sqft for tristate region 12078.49\t variance for median sqft for
# tristate region is 145890007\t\t IQR for median sqft for tristate
# region is 21496.5\t quantile for median sqft for tristate regsion
# is 25% = 6267.5; 75% = 27764.0 median for median sqft for tristate
# region is 14318\t mode for median sqft for tristate region is 4395
# total_listing_count is numeric and therefore does not have a B
# index for mode spread
```

Task 3 (30 points)

- Use at least 3 grouping functions to group the data in order to see data at different levels. You may group categorical variables further to see different patterns based on location or even groups of dates. Make your grouping output visible and comment on the insights you took away from this step.

```
# Review Active Listings and group by State
state.review.active <- tristate.region %>%
    group_by(state) %>%
    summarise(mean.active.listings = mean(active_listing_count))
state.review.active
```

```
## # A tibble: 3 x 2
##   state         mean.active.listings
##   <fct>                        <dbl>
## 1 kentucky                     6168.
## 2 ohio                        13470.
## 3 west virginia                3223.
```

```
# Insights: West Virginia has the lowest mean active listings (m =
# 3222.68), followed by Kentucky (m = 6167.96) and lastly Ohio with
# the greatest (m = 13469.88).  Interestingly, Kentucky mean active
# listings are almost double West Virginia and Ohio is more than
# double Kentucky's mean average listings. There are many more mean
# active listings in Ohio than both West Virginia and Kentucky
# combined.


# Review New Listings and group by State
state.review.new <- tristate.region %>%
    group_by(state) %>%
    summarise(mean.new.listings = mean(new_listing_count))
state.review.new
```

```
## # A tibble: 3 x 2
##   state         mean.new.listings
##   <fct>                     <dbl>
## 1 kentucky                  5139.
## 2 ohio                     14163.
## 3 west virginia             1623.
```

```
# Insights: West Virginia has fewer new listings on average (m =
# 1623.04) than the state does have active listings, indicating there
# are less new homes listed than those already active. Kentucky also
# has fewer new listings on average (m = 5139.36) than the state does
# have active listings, also indicating there are less new homes
# listed than those already active. Ohio is different on mean new
# listings (m = 14163.04) which is greater than the mean of active
# listings for the state. From the new listing groups it appears Ohio
# has many more homes available for sale than both West Virginia and
# Kentucky.

# Review Median Days on the Market by State
state.review.days <- tristate.region %>%
    group_by(state) %>%
    summarise(mean.median.days = mean(median_days_on_market))
state.review.days
```

```
## # A tibble: 3 x 2
##   state         mean.median.days
##   <fct>                    <dbl>
## 1 kentucky                  47.4
## 2 ohio                      42.0
## 3 west virginia             67.3
```

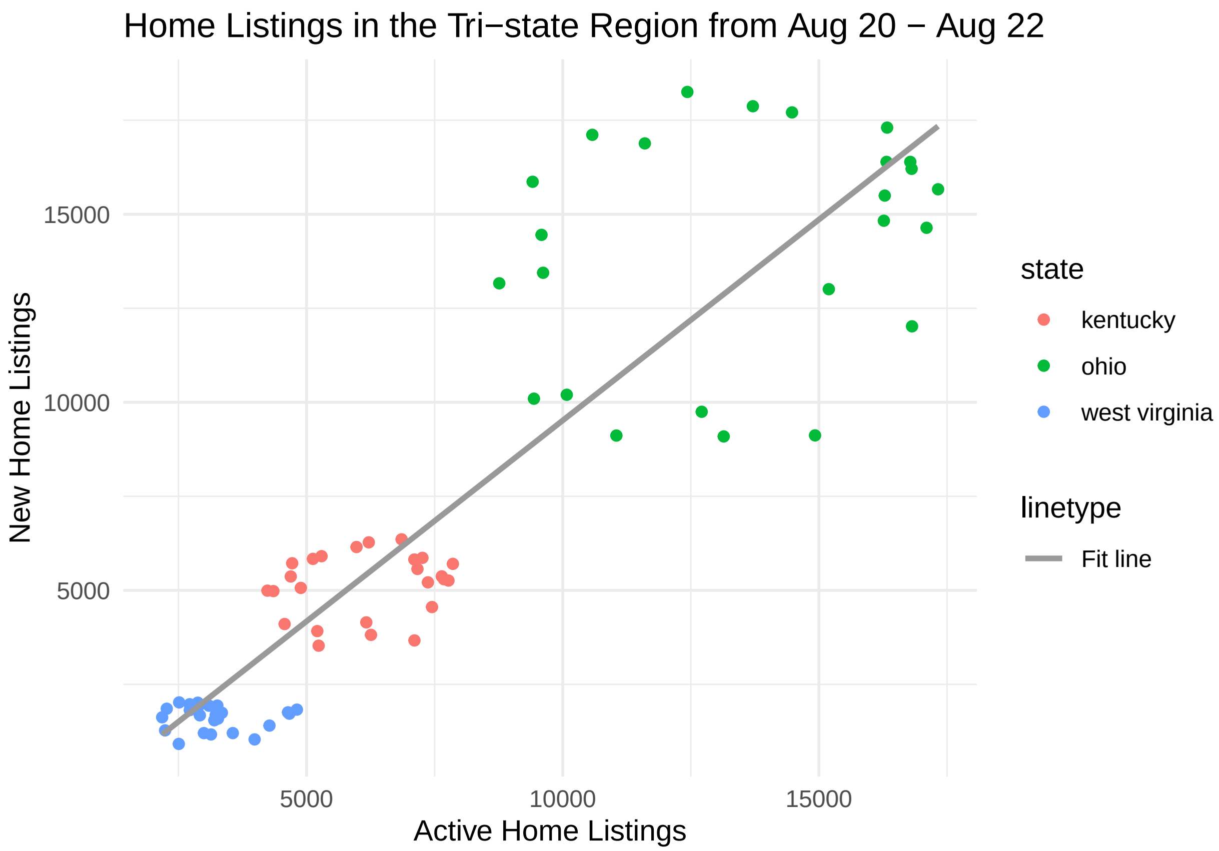
```
# Insights: These results further support that Ohio has a much more
# volatile housing market as the average median days a house is on
# the market (m = 42.04) is less than Kentucky (m = 47.44) and West
# Virginia (m = 67.32). Comparatively, Kentucky has a faster moving
# market than West Virginia with nearly double the amount of listings
# and only approximately 20 days faster.
```

Task 4 (40 points)

- Create at least 4 well-formatted, appropriate graphs depicting relationships between 2 variables (used or created in the step above) of your choosing. Use good code-formatting practices and ensure your graphs have good use of parameters including color, labels, legends, or titles - where appropriate. Include a sentence or two after each graph in comments that explains what the graph shows and what you learned from the graph.

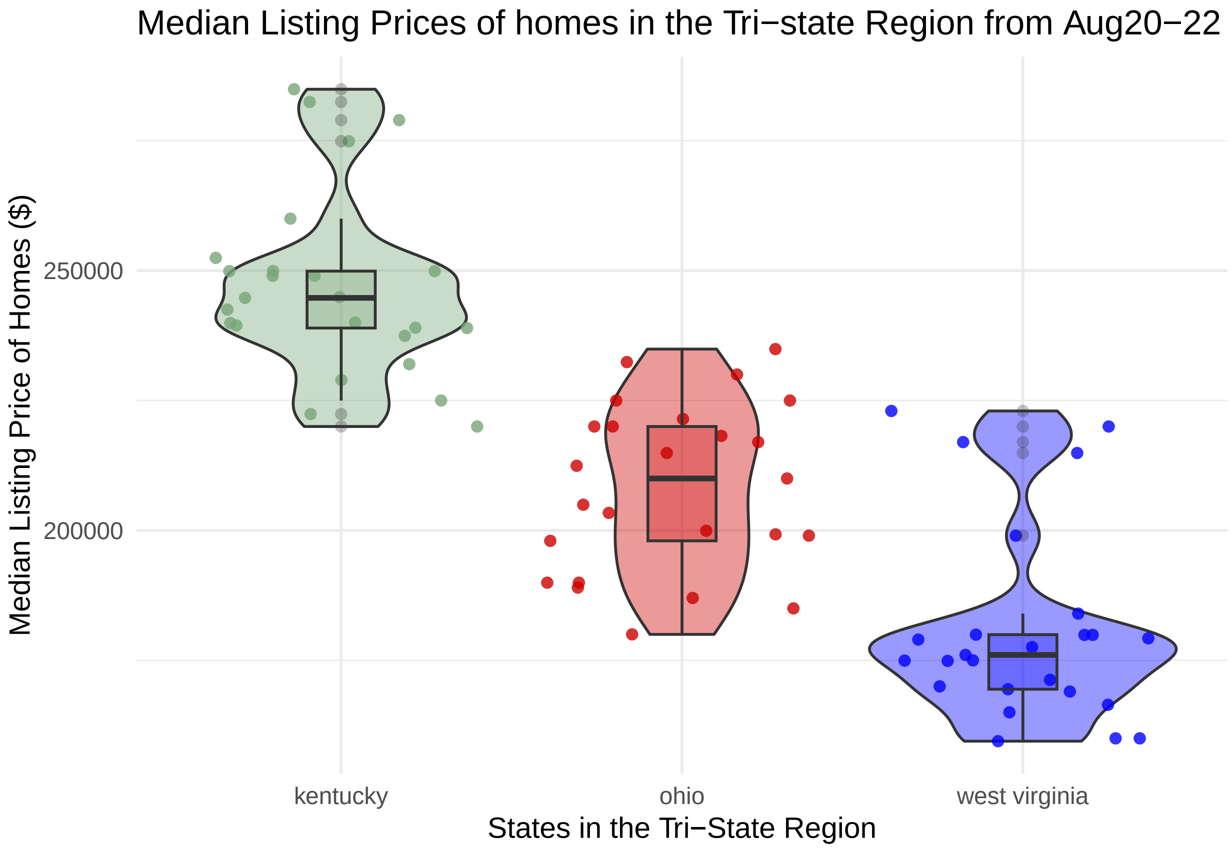
```r
# Compare by State, the new home listings and active home listings
tristate.region %>%
    ggplot(aes(y = new_listing_count, x = active_listing_count, color = state)) +
    geom_point() + geom_smooth(method = "lm", se = FALSE, aes(linetype = "Fit line"),
    color = "gray60") + theme_minimal() + labs(y = "New Home Listings",
    x = "Active Home Listings", title = "Home Listings in the Tri-state Region from Aug 20 - Aug 22")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

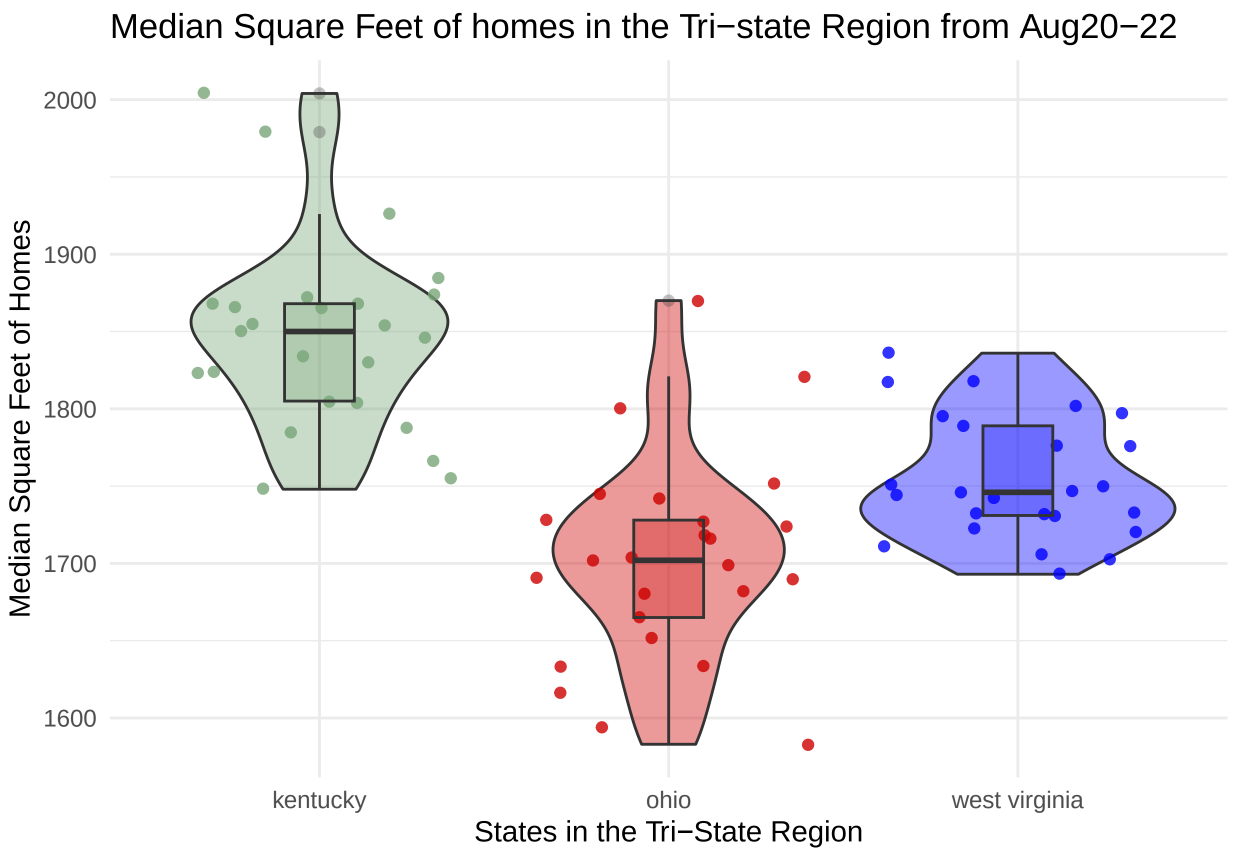


```r
# Insights: Ohio has more listings, followed by Kentucky, and then
# West Virginia.

# Compare by State, the median listing price of homes
tristate.region %>%
    ggplot(aes(x = state, y = median_listing_price, fill = state)) + geom_violin(aes(fill = state),
    alpha = 0.4) + geom_jitter(aes(color = state), alpha = 0.8) + geom_boxplot(width = 0.2,
    alpha = 0.3) + theme_minimal() + labs(y = "Median Listing Price of Homes ($)",
    x = "States in the Tri-State Region", title = "Median Listing Prices of homes in the Tri-state Regi
    scale_fill_manual(values = c("#78a678", "red3", "blue"), guide = FALSE) +
    scale_color_manual(values = c("#78A678", "red3", "blue"), guide = FALSE)
```

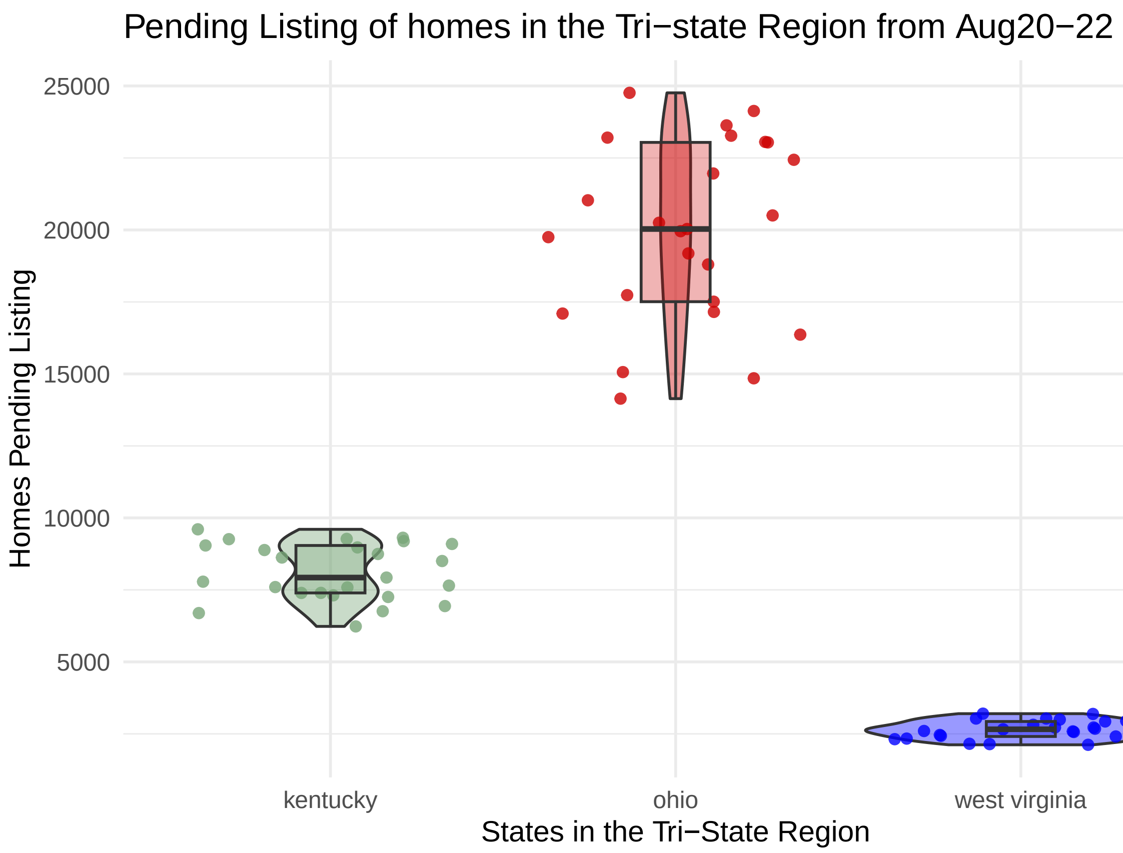
```
## Warning: The `guide` argument in `scale_*()` cannot be `FALSE`. This was deprecated in
## ggplot2 3.3.4.
## i Please use "none" instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



```
# Insights: Homes in Kentucky are the most expensive followed by Ohio
# and then West Virginia. West Virginia homes do have data with
# prices at or higher than the Ohio mean prices, though the majority
# of the data points are below the lower range of the Ohio homes.
# Kentucky's higher priced homes may impact the amount of days on the
# market and overall listings.

# Compare by State, the median square feet
tristate.region %>%
    ggplot(aes(x = state, y = median_square_feet, fill = state)) + geom_violin(aes(fill = state),
    alpha = 0.4) + geom_jitter(aes(color = state), alpha = 0.8) + geom_boxplot(width = 0.2,
    alpha = 0.3) + theme_minimal() + labs(y = "Median Square Feet of Homes",
    x = "States in the Tri-State Region", title = "Median Square Feet of homes in the Tri-state Region
    scale_fill_manual(values = c("#78a678", "red3", "blue"), guide = FALSE) +
    scale_color_manual(values = c("#78A678", "red3", "blue"), guide = FALSE)
```



```
# Insights: Homes in Kentucky are listed the highest and have the
# largest median square feet per home, but interestingly, the West
# Virginia homes are larger than that of the Ohio homes by
# approximately 50 square feet in mean median square feet per home.
# Based on the visuals from this section, the homes in Ohio are the
# smallest, but are on the market for the least amount of time, and
# cost less than Kentucky homes but more than West Virginia homes.
# Kentucky homes spend less time on the market than West Virginia
# homes, but cost more and are larger.

# Compare by State, the pending listing of homes
tristate.region %>%
    ggplot(aes(x = state, y = pending_listing_count, fill = state)) + geom_violin(aes(fill = state),
    alpha = 0.4) + geom_jitter(aes(color = state), alpha = 0.8) + geom_boxplot(width = 0.2,
    alpha = 0.3) + theme_minimal() + labs(y = "Homes Pending Listing",
    x = "States in the Tri-State Region", title = "Pending Listing of homes in the Tri-state Region fro
    scale_fill_manual(values = c("#78a678", "red3", "blue"), guide = FALSE) +
    scale_color_manual(values = c("#78A678", "red3", "blue"), guide = FALSE)
```



```
# Insights: Ohio has a much larger pool of homes pending listing than
# Kentucky and West Virginia by greater than 10,000 homes on average
# than Kentucky and nearly 20,000 more than West Virginia. This
# visual indicates there is a much larger inventory of homes in Ohio
# than the other two states in the tristate region. There are on
# average, less than 5,000 homes pending listing in West Virginia
# which may indicate a highly competitive buying market with reduced
# options and a more challenging seller's market in Ohio with the
# increased options for buyers to choose from.
```

Task 5 (20 points)

- Based on the graphs and statistics you choose from Tasks 2-4 above, make predictions in comments about what you would find when you compare variables.

```
# Based on the graphs and statistics from Tasks 2-4 I have the
# following predictions about what I will find when I compare
# variables.

# Prediction 1: There is a relationship between the median square
# feet of a home and the median listing price. I predict that as
# median square feet of a home increases, the price will increase.

# Prediction 2: There is no relationship between active listings and
# median listing price.

# Prediction 3: There is a relationship between median days on the
# market and median listing price. I predict that as median days
# increases, the price will decrease.

# Prediction 4: There is a relationship between total listings and
# median listing price. I predict that as the total listings
# increases, the price will decrease.
```

Task 6 (60 points)

- Select the appropriate test to test at least 4 relationships between a price variable as a Y variable (median or average, percentage or otherwise) and at least one other variable. Do not to choose another price variable in making predictions of price.
- Go through the correct steps for hypothesis testing based on the test you selected including listing your conclusions you made and whether they differ from your predictions above in Step 5 after examining relationships visually.
- You may choose to use ANOVA, simple, or multiple regression to evaluate 4 relationships with regards to Y to satisfy the requirements.

```
# Median Square Feet and Median Price (Simple Regression) Step 1 -
# Write the Null and Alternate Hypotheses H0: The slope of the line
# is equal to zero. HA: The slope of the line is not equal to zero.

# Step 2 - Compute the test statistic
med.price.by.med.sqft <- lm(formula = median_listing_price ~ median_square_feet,
    data = tristate.region, na.action = na.exclude)
summary(med.price.by.med.sqft)
```

```
##
## Call:
## lm(formula = median_listing_price ~ median_square_feet, data = tristate.region,
##     na.action = na.exclude)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -52195 -20669   2453  19603  50737
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -178442.90   67194.22  -2.656  0.00971 **
## median_square_feet     220.88      37.96   5.819 1.47e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26910 on 73 degrees of freedom
## Multiple R-squared:  0.3168, Adjusted R-squared:  0.3075
## F-statistic: 33.86 on 1 and 73 DF,  p-value: 1.469e-07
```

```r
# Add confidence interval calculation for interpretation of results
# portion
ci.med.price.by.med.sqft <- confint(object = med.price.by.med.sqft)
ci.med.price.by.med.sqft
```

```
##                            2.5 %      97.5 %
## (Intercept)         -312360.7891 -44525.0181
## median_square_feet      145.2235    296.5386
```

```r
# Step 3 - Calculate the probability that your test statistic is at
# least as big as it is if there is no relationship (i.e. the null is
# true) p-value of 1.47e-07 for the slope in the output and
# significant at the *** level

# Steps 4 and 5 - Interpret the probability and write a conclusion.
# The median square feet of a home is a statistically significant
# predictor of the median listing price for a home in the tristate
# region (b = 220.88; p<.001) in this sample. For every 1sqft
# increase in median square feet of a home, the predicted median
# listing price increases by 220.88 dollars.The value of the slope in
# the sample is 220.88, and the value of the slope is likely between
# 145.22 and 296.54 in the population that the sample came from (95%
# CI: 145.22-296.54).  With every 1sqft increase in median square
# feet of a home, the median listing price is between 145.22 and
# 296.54 more dollars expensive. These results suggest that homes
# with a larger median square feet value are more expensive in the
# tristate region. The Adjusted R-Squared value of .3075 indicates
# that this explains about 31% of the variance in median listing
# price. This aligns with my prediction above that as median square
# feet increases in a home so does the median listing price.



# Active Listings and Median Price (Simple Regression) Step 1 - Write
# the Null and Alternate Hypotheses H0: The slope of the line is
# equal to zero. HA: The slope of the line is not equal to zero.

# Step 2 - Compute the test statistic
med.price.by.active.listings <- lm(formula = median_listing_price ~ active_listing_count,
    data = tristate.region, na.action = na.exclude)
summary(med.price.by.active.listings)
```

```
##
## Call:
## lm(formula = median_listing_price ~ active_listing_count, data = tristate.region,
##     na.action = na.exclude)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -47820 -28680    855  27667  73176
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)          2.039e+05  7.071e+03  28.836   <2e-16 ***
## active_listing_count 1.079e+00  7.897e-01   1.366    0.176
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32150 on 73 degrees of freedom
## Multiple R-squared:  0.02494,    Adjusted R-squared:  0.01158
## F-statistic: 1.867 on 1 and 73 DF,  p-value: 0.176
```

```
# Add confidence interval calculation for interpretation of results
# portion
ci.med.price.by.active.listings <- confint(object = med.price.by.active.listings)
ci.med.price.by.active.listings
```

```
##                              2.5 %       97.5 %
## (Intercept)           1.897954e+05 2.179784e+05
## active_listing_count -4.947755e-01 2.653051e+00
```

```
# Step 3 - Calculate the probability that your test statistic is at
# least as big as it is if there is no relationship (i.e. the null is
# true) p-value of .176 for the slope in the output and not
# significant

# Steps 4 and 5 - Interpret the probability and write a conclusion.
# The active home listings is not a statistically significant
# predictor of the median listing price for a home in the tristate
# region (b = 1.079; p>.05) in this sample. Further, the Adjusted
# R-squared value is .01158 indicating that this only explains about
# 1% of the variance in median listing price. This aligns with my
# prediction above that there is not statistically significant
# relationship between active listings and the median listing price.



# Median Days on the Market and Median Price (Simple Regression) Step
# 1 - Write the Null and Alternate Hypotheses H0: The slope of the
# line is equal to zero. HA: The slope of the line is not equal to
# zero.

# Step 2 - Compute the test statistic
med.price.by.mkt.days <- lm(formula = median_listing_price ~ median_days_on_market,
    data = tristate.region, na.action = na.exclude)
summary(med.price.by.mkt.days)
```

```
##
## Call:
## lm(formula = median_listing_price ~ median_days_on_market, data = tristate.region,
##     na.action = na.exclude)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -40909 -19472  -2943  18852  58574
##
```

```
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           278474.2     9195.6   30.28  < 2e-16 ***
## median_days_on_market  -1269.7      167.5   -7.58 8.64e-11 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 24350 on 73 degrees of freedom
## Multiple R-squared:  0.4404, Adjusted R-squared:  0.4328
## F-statistic: 57.46 on 1 and 73 DF,  p-value: 8.639e-11
```

```
# Add confidence interval calculation for interpretation of results
# portion
ci.med.price.by.mkt.days <- confint(object = med.price.by.mkt.days)
ci.med.price.by.mkt.days
```

```
##                            2.5 %      97.5 %
## (Intercept)           260147.431 296800.8971
## median_days_on_market  -1603.563   -935.8775
```

```
# Step 3 - Calculate the probability that your test statistic is at
# least as big as it is if there is no relationship (i.e. the null is
# true) p-value of 8.64e-11 for the slope in the output and
# significant at the *** level

# Steps 4 and 5 - Interpret the probability and write a conclusion.
# The median days a home is on the market is a statistically
# significant predictor of the median listing price for a home in the
# tristate region (b = -1269.7; p<.001) in this sample. For every
# additional day the median days a home is on the market increases,
# the predicted median listing price decreases by -1269.7 dollars.The
# value of the slope in the sample is -1269.7, and the value of the
# slope is likely between -1603.56 and -935.88 in the population that
# the sample came from (95% CI: -1603.56 to -935.88).  With every
# additional days the median days a home is on the market, the median
# listing price is between -1603.56 and -935.88 dollars less
# expensive. These results suggest that homes lose value the longer
# they stay on the market in the tristate region. The Adjusted
# R-Squared value of .4328 indicates that this explains about 43% of
# the variance in median listing price. This aligns with my
# prediction above that as median days on the market increases, the
# median listing price will decrease.



# Total Listings and Median Price (Simple Regression) Step 1 - Write
# the Null and Alternate Hypotheses H0: The slope of the line is
# equal to zero. HA: The slope of the line is not equal to zero.

# Step 2 - Compute the test statistic
med.price.by.total.listings <- lm(formula = median_listing_price ~ total_listing_count,
    data = tristate.region, na.action = na.exclude)
summary(med.price.by.total.listings)
```

```
##
## Call:
## lm(formula = median_listing_price ~ total_listing_count, data = tristate.region,
##     na.action = na.exclude)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -46244 -27858  -2437  28057  74468
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         2.029e+05  6.616e+03  30.664   <2e-16 ***
## total_listing_count 5.174e-01  3.074e-01   1.683   0.0966 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31940 on 73 degrees of freedom
## Multiple R-squared:  0.03736,    Adjusted R-squared:  0.02417
## F-statistic: 2.833 on 1 and 73 DF,  p-value: 0.09661
```

```
# Add confidence interval calculation for interpretation of results
# portion
ci.med.price.by.total.listings <- confint(object = med.price.by.total.listings)
ci.med.price.by.total.listings
```

```
##                             2.5 %       97.5 %
## (Intercept)          1.896806e+05 2.160506e+05
## total_listing_count -9.523475e-02 1.130075e+00
```

```
# Step 3 - Calculate the probability that your test statistic is at
# least as big as it is if there is no relationship (i.e. the null is
# true) p-value of .0966 for the slope in the output and not
# significant

# Steps 4 and 5 - Interpret the probability and write a conclusion.
# The total listings of homes is not a statistically significant
# predictor of the median listing price for a home in the tristate
# region (b = .5174; p>.05) in this sample. Further, the Adjusted
# R-squared value is .02417 indicating that this only explains about
# 2% of the variance in median listing price. This is counter to my
# prediction above that there is a statistically significant
# relationship between total listings and the median listing price.
# Instead, there is not a statistically significant relationship
# (p>.05) between total listings and median listing price. I expected
# the more homes listed would increase competition and encourage
# sellers to decrease their listing prices.
```

Task 7 (20 points)

- Describe the variables you found to be the strongest predictors of price. What limitations, biases, or confounding variables could affect the results?

```
# The variables that I found to be the strongest predictors of median
# listing price are median_square_feet and median_days_on_market.
# With these two variables, both are statistically significant at the
# p<.001 *** level. Further, median_square_feet has an Adjusted
# R-squared value of: .3075 and median_days_on_market has an Adjusted
# R-squared value of: .4328 which explain approximately 31% and 43%
# of the variance in median listing price, respectively.
# Additionally, there is a positive relationship between
# median_square_feet and median_listing_price such that as
# median_square_feet increases, so does median_listing_price. After
# completing the linear regression for median_square_feet and
# median_listing_price, I determined that each additional median
# square foot a home has, the model predicts with 95% confidence that
# the median_listing_price will be between 145.22 and 296.54 more
# dollars expensive.

# Regarding median_days_on_market and median_listing_price, there is
# a negative relationship between these two variables such that as
# median_days_on_market increases, the median_listing_price
# decreases. I determined that each additional median day a home is
# on the market, the model predicts with 95% confidence that the
# median_listing_price will be between -1603.56 and -935.88 less
# expensive.  The longer the home is listed the more the home price
# will decrease.

# Intuitively, these results make logical sense - the larger the
# home, the more expensive and the longer the home is on the market,
# the more the home loses value. These results are limited by other
# additional considerations such as credit scores, seasonality, and
# federal interest rates for financing options.  Additionally,
# examining the data at the state level does not provide an in-depth
# review or analysis on the individual zip codes, counties, or
# districts within each state. Confounding variables that could
# affect the results are reoccurring listings of the same home at
# different times in the year (adding and taking the same home off
# the market listing), Realtor performance, or changes in utilities
# expenses, property tax rates, sizes of the total lot and yard
# ratios, employment opportunities, annual income, localized location
# (what is around the neighborhood that would entice someone to
# leave/not live there).
```