Homework 1

Warm up

Before we introduce the data, let's warm up with some simple exercises.

- Update the YAML, changing the author name to your name, and **knit** the document.

Packages

We'll use the **tidyverse** package for much of the data wrangling and visualization and the data lives in the **dsbox** package.

Data

The data can be found in the **dsbox** package, and it's called `edibnb`. Since the dataset is distributed with the package, we don't need to load it separately; it becomes available to us when we load the package.

You can view the dataset as a spreadsheet using the `View()` function. Note that you should not put this function in your R Markdown document, but instead type it directly in the Console, as it pops open a new window (and the concept of popping open a window in a static document doesn't really make sense…). When you run this in the console, you'll see the following **data viewer** window pop up.

```
View(edibnb)
```

You can find out more about the dataset by inspecting its documentation, which you can access by running `?edibnb` in the Console or using the Help menu in RStudio to search for `edibnb`.

Exercises

Hint: The Markdown Quick Reference sheet has an example of inline R code that might be helpful. You can access it from the Help menu in RStudio.

1. How many observations (rows) does the dataset have? Instead of hard coding the number in your answer, use inline code.

```
summary(edibnb)
```

```
##        id              price        neighbourhood       accommodates
##  Min.   :   15420   Min.   :  0.00   Length:13245       Min.   : 1.000
##  1st Qu.:13279107   1st Qu.: 49.00   Class :character   1st Qu.: 2.000
##  Median :20171841   Median : 75.00   Mode  :character   Median : 3.000
##  Mean   :20077242   Mean   : 97.21                      Mean   : 3.541
##  3rd Qu.:27397925   3rd Qu.:110.00                      3rd Qu.: 4.000
##  Max.   :36066014   Max.   :999.00                      Max.   :19.000
##                     NA's   :199
##    bathrooms        bedrooms           beds        review_scores_rating
##  Min.   :0.000   Min.   : 0.000   Min.   : 0.000   Min.   : 20.00
##  1st Qu.:1.000   1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.: 93.00
##  Median :1.000   Median : 1.000   Median : 2.000   Median : 97.00
##  Mean   :1.226   Mean   : 1.583   Mean   : 2.032   Mean   : 95.02
##  3rd Qu.:1.000   3rd Qu.: 2.000   3rd Qu.: 3.000   3rd Qu.: 99.00
##  Max.   :9.000   Max.   :13.000   Max.   :30.000   Max.   :100.00
##  NA's   :12      NA's   :4        NA's   :15       NA's   :2177
##  number_of_reviews listing_url
##  Min.   :  0.00    Length:13245
##  1st Qu.:  2.00    Class :character
##  Median : 12.00    Mode  :character
##  Mean   : 37.73
##  3rd Qu.: 45.00
##  Max.   :773.00
##
```

Answer: There are 13,245 observations in the dataset.

2. Run `View(edibnb)` in your Console to view the data in the data viewer. What does each row in the dataset represent?

Answer: Each row represents a unique Airbnb listing in Edinburgh, along with some information about it (i.e. price and number of bedrooms).

Each column represents a variable. We can get a list of the variables in the data frame using the `names()` function.

```
names(edibnb)
```

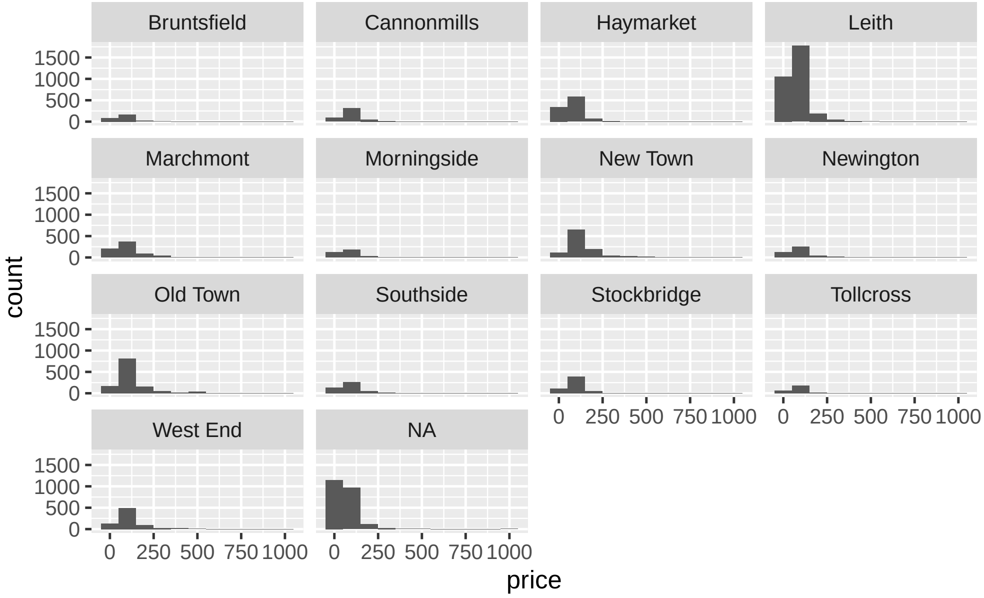
```
##  [1] "id"                   "price"                "neighbourhood"
##  [4] "accommodates"         "bathrooms"            "bedrooms"
##  [7] "beds"                 "review_scores_rating" "number_of_reviews"
## [10] "listing_url"
```

You can find descriptions of each of the variables in the help file for the dataset, which you can access by running `?edibnb` in your Console.

Note: The plot will give a warning about some observations with non-finite values for price being removed. Don't worry about the warning, it simply means that 199 listings in the data didn't have prices available, so they can't be plotted.

3. Create a faceted histogram where each facet represents a neighbourhood and displays the distribution of Airbnb prices in that neighbourhood. Think critically about whether it makes more sense to stack the facets on top of each other in a column, lay them out in a row, or wrap them around. Along with your visualization, include your reasoning for the layout you chose for your facets.

```
ggplot(data = edibnb, mapping = aes(x = price)) +
  geom_histogram(binwidth = 100) +
  facet_wrap(~neighbourhood)
```



I chose this layout for my facets because since it is designed so each of the individual neighborhoods can be viewed all at once, this will allow us to get an idea of how the prices are distributed in every one of them. This also works because there are many different neighborhoods in Edinburgh, and some listings do not have a specified neighborhood.

Let's de-construct this code:

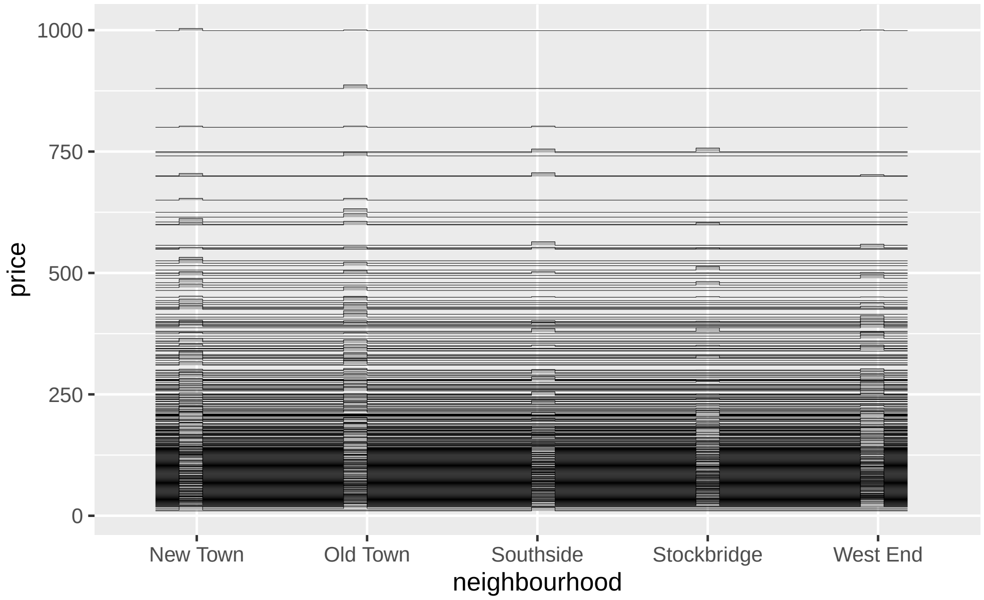
- `ggplot()` is the function we are using to build our plot, in layers.
- In the first layer we always define the data frame as the first argument. Then, we define the mappings between the variables in the dataset and the **aes**thetics of the plot (e.g. x and y coordinates, colours, etc.).
- In the next layer we represent the data with **geom**etric shapes, in this case with a histogram. You should decide what makes a reasonable bin width for the histogram by trying out a few options.
- In the final layer we facet the data by neighborhood.

4. Use a single pipeline to identity the neighborhoods with the top five median listing prices. Then, in another pipeline filter the data for these five neighborhoods and make ridge plots of the distributions of listing prices in these five neighborhoods. In a third pipeline calculate the minimum, mean, median, standard deviation, IQR, and maximum listing price in each of these neighborhoods. Use the visualization and the summary statistics to describe the distribution of listing prices in the neighborhoods. (Your answer will include three pipelines, one of which ends in a visualization, and a narrative.)

```r
# Get neighborhoods with top 5 median prices by using
# group_by(), summarise(), median(), and top_n() functions
data1 <- edibnb %>% group_by(neighbourhood) %>% summarise(
  median_price = median(price, na.rm = TRUE)) %>%
top_n(n = 5, median_price)
data1
```

```
## # A tibble: 7 x 2
##   neighbourhood median_price
##   <chr>                <dbl>
## 1 Bruntsfield             80
## 2 Marchmont               80
## 3 New Town               100
## 4 Old Town                90
## 5 Southside               80
## 6 Stockbridge             85
## 7 West End                90
```

```r
# Filter the data based on the neighborhoods we got
# and then make a ridge plot
data2 <- edibnb %>%
filter(neighbourhood %in% c("New Town", "Old Town","West End", "Stockbridge", "Southside"))
ggplot(data = data2, mapping = aes(x = neighbourhood, y = price, group = price)) +
ggridges::geom_ridgeline(size = 0.1, stat = "binline")
```



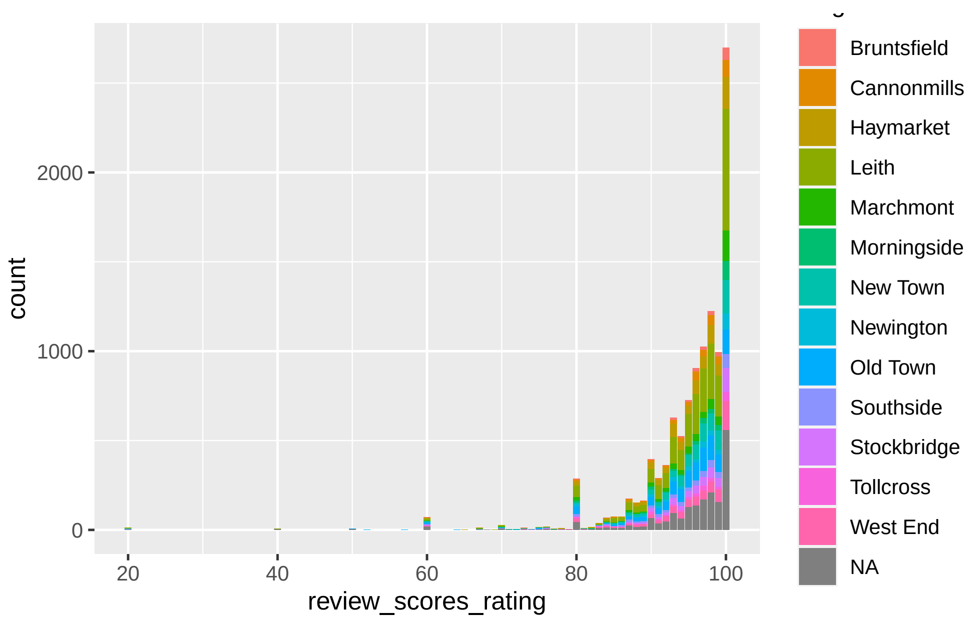
```
# Group by neighborhood and get minimum, mean, maximum prices,
# as well as median, standard deviation, and IQR
edibnb %>% group_by(neighbourhood) %>% summarise(min(price, na.rm = TRUE), mean(price, na.rm = TRUE),
                   median(price, na.rm = TRUE), sd(price, na.rm = TRUE)
                    ,IQR(price, na.rm = TRUE), max(price, na.rm = TRUE)) %>%
filter(neighbourhood %in% c("New Town", "Old Town","West End", "Stockbridge", "Southside"))
```

```
## # A tibble: 5 x 7
##   neighbourhood min(price, na.rm = TRU~1 mean(~2 media~3 sd(pr~4 IQR(p~5 max(p~6
##   <chr>                            <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 New Town                            12    136.     100   109.     86.5     999
## 2 Old Town                            15    128.      90   110.     76       999
## 3 Southside                           10    107.      80    99.3    70       800
## 4 Stockbridge                         21    104.      85    77.6    66       750
## 5 West End                            19    116.      90    93.3    80       999
## # ... with abbreviated variable names 1: `min(price, na.rm = TRUE)`,
## #   2: `mean(price, na.rm = TRUE)`, 3: `median(price, na.rm = TRUE)`,
## #   4: `sd(price, na.rm = TRUE)`, 5: `IQR(price, na.rm = TRUE)`,
## #   6: `max(price, na.rm = TRUE)`
```

New Town has the highest median price and it shares its maximum price with Old Town and West End. Southside has the lowest median price, while Stockbridge has the lowest maximum and mean prices.

5. Create a visualization that will help you compare the distribution of review scores (`review_scores_rating`) across neighborhoods. You get to decide what type of visualization to create and there is more than one correct answer! In your answer, include a brief interpretation of how Airbnb guests rate properties in general and how the neighborhoods compare to each other in terms of their ratings.

```
ggplot(data = edibnb, mapping = aes(x = review_scores_rating)) +
  geom_bar(mapping = aes(fill = neighbourhood), binwidth = 100)
```



Starting at a score of approximately 81, the total amount of reviews (when all of the neighborhoods are combined together) containing a particular score appears to (for the most part) increase. Also starting from that point on, the neighborhood of Leith appears to always make up the largest portion of the total amount of reviews containing the corresponding score.