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.

```
library(tidyverse)
library(openintro)
library(ggplot2)
library(ggridges)
library(knitr)
library(dsbox) # devtools::install_github("rstudio-education/dsbox")
```

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.

```
cat("The dataset has", nrow(edibnb), "rows")
```

```
## The dataset has 13245 rows
```

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

##Each row represents an observation of the airbnb listing.

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

```
names(edibnb)
```

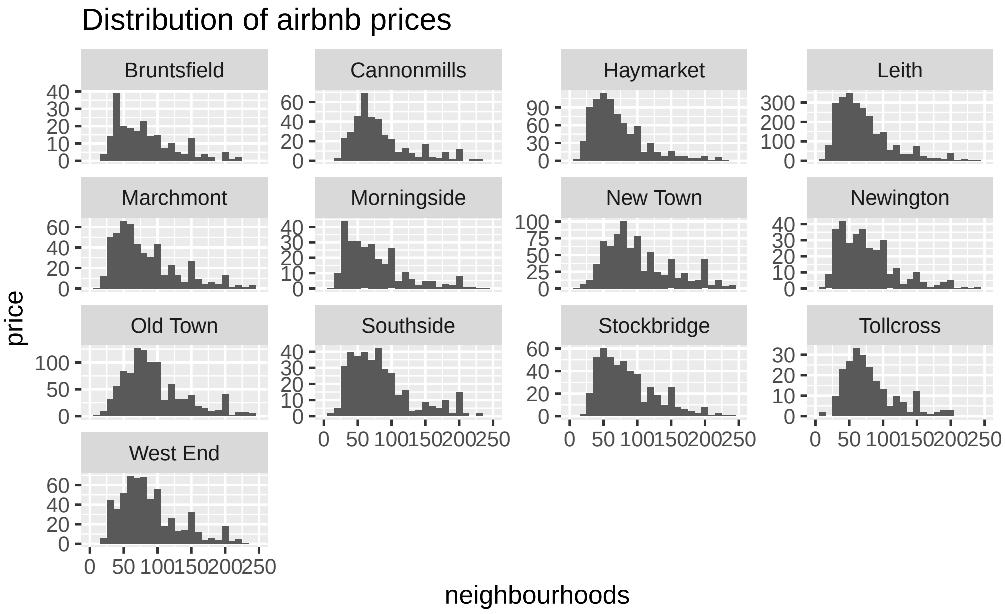
```
##  [1] "id"                   "price"                "neighbourhood"
##  [4] "accommodates"         "bathrooms"            "bedrooms"
##  [7] "beds"                 "review_scores_rating" "number_of_reviews"
## [10] "listing_url"
```

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 = na.omit(edibnb), mapping = aes(x = price)) +
  geom_histogram(binwidth = 10) +
  xlim(0, 250) +
  facet_wrap(~neighbourhood,scales = "free_y", nrow =4)+
    labs(title = "Distribution of airbnb prices",
         x = "neighbourhoods" ,
         y= "price")
```



##The number of observations as per neighbourhood is 13. If it was less, stacking the facets on top of each other in a column would make more sense to see the distribution. For exploratory data analysis: we can easily and rapidly compare patterns in different parts of the data and see whether they are the same or different In this case, wrapping them around would be more suitable since there are lots of observations. I used a range of 0-250 for x and chose to "free y" because we can get a better sense of distribution values.

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.

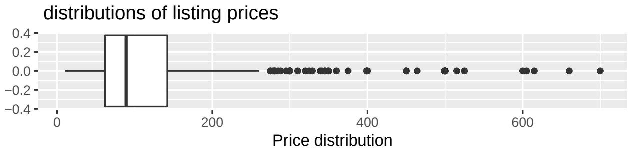
4. Use a single pipeline to identity the neighborhoods with the top five median listing prices. Then, in another pipeline filter the data for these five neighborhoods and make ridge plots of the distributions of listing prices in these five neighborhoods. In a third pipeline calculate the minimum, mean, 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.)

```
#neighbourhood with top median listing price
top_5 <- edibnb %>%
  #filter(!is.na(neighbourhood)) %>%
  group_by(neighbourhood) %>% #<<
  summarise(median_price = median(price, na.rm=T)) %>%
  arrange(desc(median_price)) %>%
  slice(1:5) #<<
top_5
```

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

##The top 5 neighbourhoods with highest median prices are listed above.

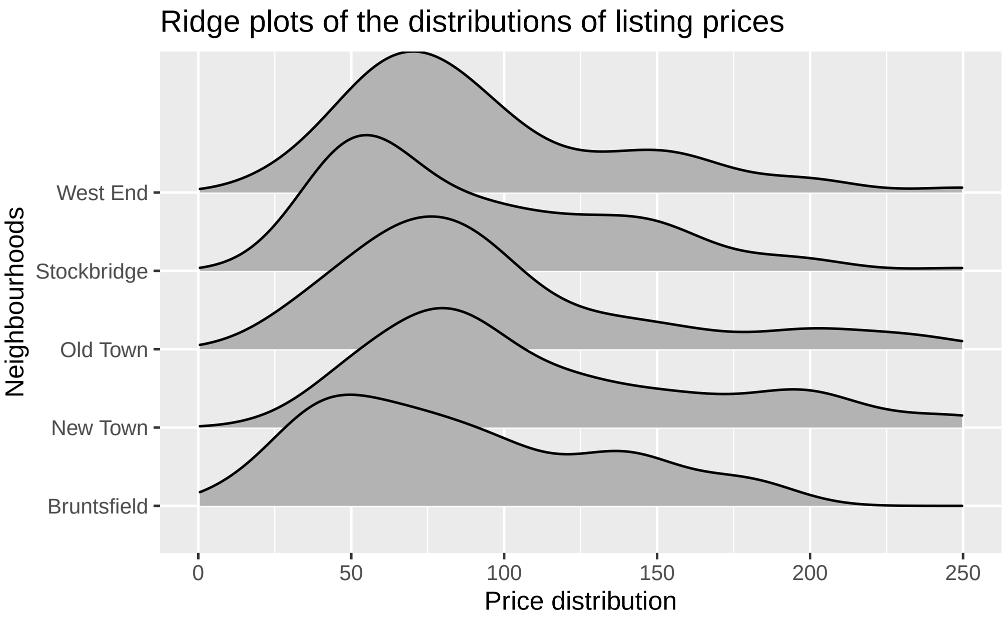
```
edibnb %>%
  filter(neighbourhood == top_5$neighbourhood) %>%
  ggplot(aes(x = price)) +
  geom_boxplot() +
  labs(title = " distributions of listing prices",
       x = "Price distribution")
```



##At first,I used box plot to find the five-number summary of the dataset to locate the range to observe the distribution of prices.I saw that the outliers are prices above 250, thus I set 250 as the lim in x-axis to study the ridgeplots

```
edibnb %>%
  filter(neighbourhood == top_5$neighbourhood) %>%
  ggplot(aes(x = price, y = neighbourhood)) +
  xlim(0, 250)+
  geom_density_ridges() +
  labs(title ="Ridge plots of the distributions of listing prices",
       x = "Price distribution", y = "Neighbourhoods")
```

```
## Picking joint bandwidth of 15.6
```



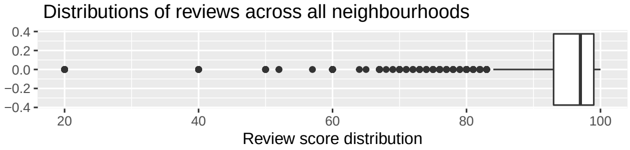
##Here we can see that the distribution is not symmetrical, but it is right skewed. If the distribution is right skewed, then the mean price of airbnb would be lower than the median price, which is proven by the summary statistics finding also..

```
edibnb %>%
  filter(neighbourhood == top_5$neighbourhood) %>%
  group_by(neighbourhood) %>% #<<
    summarise(
      min_price = min(price,na.rm=T),
      mean_price = mean(price,na.rm=T),
      median_price = median(price,na.rm=T),
      max_price = max(price,na.rm=T),
      std_dev = sd(price,na.rm=T),
      iqr_price = IQR(price,na.rm =T)
    )
```

```
## # A tibble: 5 x 7
##   neighbourhood min_price mean_price median_price max_price std_dev iqr_price
##   <chr>             <dbl>      <dbl>        <dbl>     <dbl>   <dbl>     <dbl>
## 1 Bruntsfield          10      109.            80       660   112.       88
## 2 New Town             25      131.           100       605    93.3      87.2
## 3 Old Town             20      124.            89       615   104.       75
## 4 Stockbridge          29       96.5           80       450    62.8      73
## 5 West End             19      114.            80       700    95.6      70
```

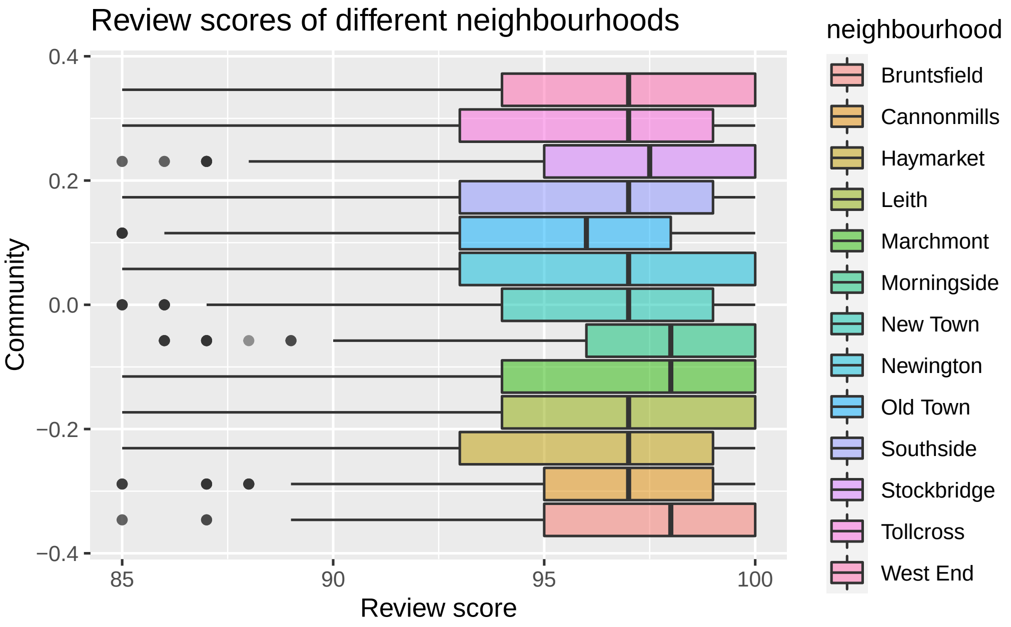
5. Create a visualization that will help you compare the distribution of review scores (`review_scores_rating`) across neighborhoods. You get to decide what type of visualization to create and there is more than one correct answer! In your answer, include a brief interpretation of how Airbnb guests rate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

```
edibnb %>%
  filter(!is.na(review_scores_rating)) %>%
  ggplot(aes(x = review_scores_rating)) +
  geom_boxplot() +
  labs(title = " Distributions of reviews across all neighbourhoods",
       x = "Review score distribution")
```



##At first,I used box plot to find the five-number summary of the dataset to locate the range to observe the distribution of review scores .I saw that the outliers are scores below 85, thus I set 85 as the lower lim in x-axis to study the box plots

```
edibnb %>%
  filter(!is.na(neighbourhood)) %>%
  ggplot(aes(x = review_scores_rating,
                fill = neighbourhood)) + #<<
   xlim(85, 100)+
   geom_boxplot(adjust = 2,
             alpha = 0.5) + #<<
  labs(
    x = "Review score",
    y = "Community",
    title = "Review scores of different neighbourhoods",
  )
```



##Here we can see that the distribution is not symmetrical, but it is `highly` *left skewed. This is proven by the summary statistics below finding also. Note that in a skewed left distribution like this, the bulk of the revies are well rated close to 100, with only a few observations that are rated bad in that range. This shows that the airbnb listings across the communities were generally highly rated*

```
library(e1071)
edibnb %>%
  group_by(neighbourhood) %>% #<<
    summarise(
      skewness_reviews = skewness(review_scores_rating,na.rm =T)
    )
```

```
## # A tibble: 14 x 2
##    neighbourhood skewness_reviews
##    <chr>                    <dbl>
##  1 Bruntsfield              -5.30
##  2 Cannonmills              -2.97
##  3 Haymarket                -2.08
##  4 Leith                    -4.31
##  5 Marchmont                -3.44
##  6 Morningside              -2.24
##  7 New Town                 -5.23
##  8 Newington                -3.61
##  9 Old Town                 -3.60
## 10 Southside                -3.39
## 11 Stockbridge              -3.13
## 12 Tollcross                -2.46
## 13 West End                 -4.40
## 14 <NA>                     -3.73
```