Exercise 2: Reporting, Data Wrangling and Graphing

Yaqi Shi

07/09/2024

- Quick R
- Rstudio cheatsheet
- Rstudio for beginners

Part 1: Analyze NYC flight delays.

Install the "nycflits13" package. The data comes from the US Bureau of Transportation Statistics. Using the data, complete the following tasks:

1. Find all flights that had an arrival delay of >4 hours, return the first 5 row. (Note: `arr_delay` is in mins)
2. Find all flight names that flew from JFK to IAH, i.e. return only unique values of "flight" variable after filtering. Hint: `unique()` would help.
3. Find how many flights were operated by UA.
4. Find how many unique flights were operated by UA.
5. Sort flights that have the most delayed flights. Show the first 5 row.
6. Generate a scatter plot with x-axis `dist` and y-axis `delay`, where each dot is a unique flights and destination, `dist` is the average distance of each destination `dest`, and `delay` is the average delay time `arr_delay`, with the size of `dot` equals to the count of delay records.

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## 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 errors
```

```
library(nycflights13)
head(flights)
```

```
## # A tibble: 6 x 19
##    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
```

```
## 1  2013     1     1      517            515         2      830            819
## 2  2013     1     1      533            529         4      850            830
## 3  2013     1     1      542            540         2      923            850
## 4  2013     1     1      544            545        -1     1004           1022
## 5  2013     1     1      554            600        -6      812            837
## 6  2013     1     1      554            558        -4      740            728
## # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

Solution

1. Find all flights that had an arrival delay of >4 hours, i.e. return the first 5 row. (Note: `arr_delay` is in mins)

```
flights %>% filter(arr_delay > 240) %>% head(5)
```

```
## # A tibble: 5 x 19
##    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
## 1  2013     1     1      848           1835       853     1001           1950
## 2  2013     1     1     1815           1325       290     2120           1542
## 3  2013     1     1     1842           1422       260     1958           1535
## 4  2013     1     1     2115           1700       255     2330           1920
## 5  2013     1     1     2205           1720       285       46           2040
## # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

2. Find all flight names that flew from JFK to IAH, i.e. return only unique values of "flight" variable after filtering. Hint: `unique()` would help.

```
df <- flights %>% filter(origin == "JFK" & dest == "IAH")
unique(df$flight)
```

```
## [1]  211 1901  523
```

3. Find how many flights were operated by UA.

```
nrow(filter(flights, carrier %in% c("UA")))
```

```
## [1] 58665
```

4. Find how many unique flights were operated by UA.

```
df <- filter(flights, carrier %in% c("UA"))
length(unique(df$flight))
```

```
## [1] 1285
```

5. Sort flights that have the most delayed flights. Show the first 5 row.

```r
flights %>% arrange(desc(dep_delay)) %>% head(5)
```

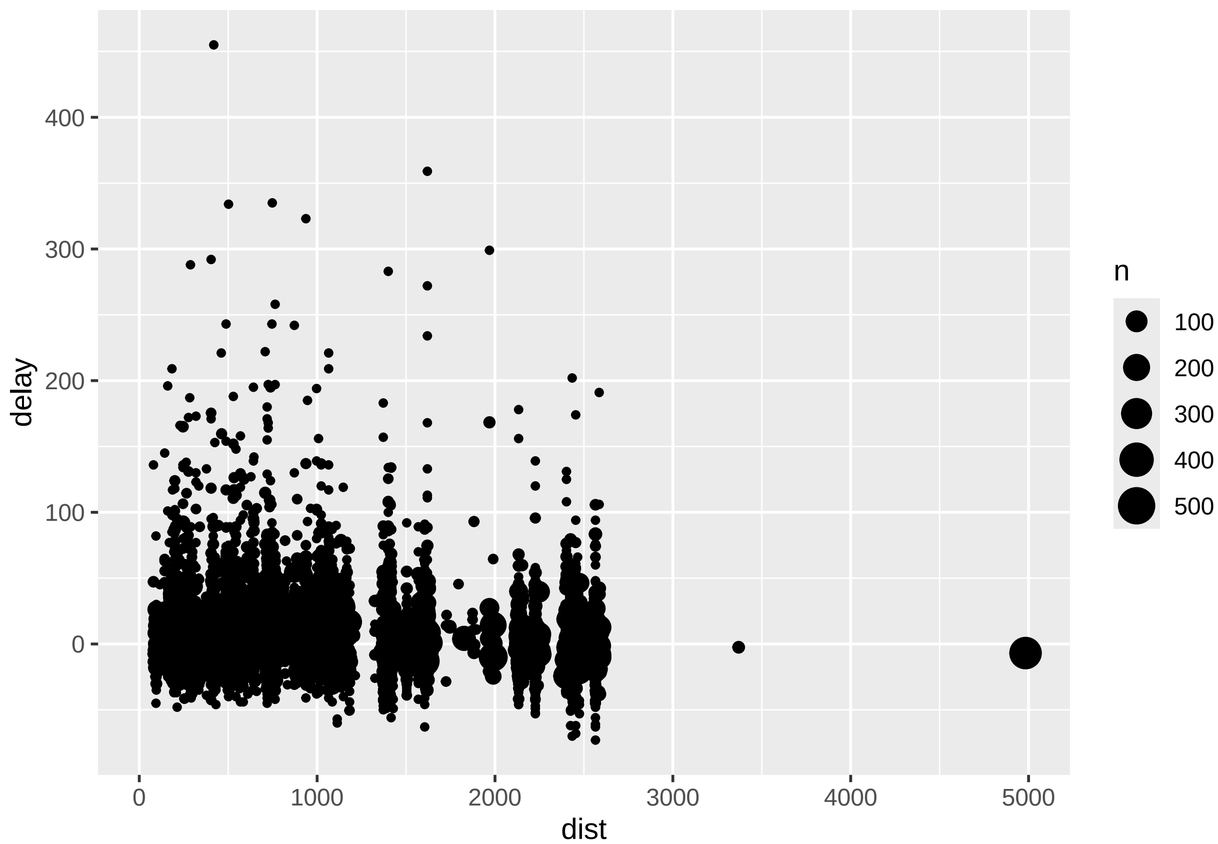
```
## # A tibble: 5 x 19
##    year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int> <int>    <int>          <int>     <dbl>    <int>          <int>
## 1  2013     1     9      641            900      1301     1242           1530
## 2  2013     6    15     1432           1935      1137     1607           2120
## 3  2013     1    10     1121           1635      1126     1239           1810
## 4  2013     9    20     1139           1845      1014     1457           2210
## 5  2013     7    22      845           1600      1005     1044           1815
## # i 11 more variables: arr_delay <dbl>, carrier <chr>, flight <int>,
## #   tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>,
## #   hour <dbl>, minute <dbl>, time_hour <dttm>
```

6. Generate a scatter plot with x-axis `dist` and y-axis `delay`, where each dot is a unique flights and destination, `dist` is the average distance of each destination `dest`, and `delay` is the average delay time `arr_delay`, with the size of `dot` equals to the count of delay records.

```r
flights %>%
  group_by(flight, dest) %>%
  summarise(delay = mean(arr_delay), dist = mean(distance), n = n()) %>%
  ggplot() +
  geom_point(aes(x = dist, y = delay, size = n))
```

```
## `summarise()` has grouped output by 'flight'. You can override using the
## `.groups` argument.
```

```
## Warning: Removed 2824 rows containing missing values or values outside the scale range
## (`geom_point()`).
```



delay
400
300
200
100
0
0
1000
2000
3000
4000
5000
dist
n
100
200
300
400
500

Part 2: LaTeX.

1. Finish the Markdown tutorial: https://www.markdowntutorial.com/
2. (Tossing for a head, C&B Example 1.5.4) Suppose we do an experiment that consists of tossing a coin until a head appears. Let $p =$ probability of a head on any given toss, and define a random variable $X =$ number of tosses required to get a head. **Use Rmarkdown to type the the solution.**

(i) What is $P(X = x)$?
(ii) For any positive integer x, calculate $P(X \leq x)$.
(iii) Calculate the cdf $F_X(x)$.
(iv) What is $\lim_{x\to\infty} F_X(x)$?

Solution:

(i)

$$P(X = x) = (1-p)^{x-1}p$$

(ii)

$$P(X \leq x) = \sum_{i=1}^{x} P(X = i) = \sum_{i=1}^{x}(1-p)^{i-1}p$$

(iii)

$$\begin{aligned} F_X(x) &= P(X \leq x) \\ &= \frac{1-(1-p)^x}{1-(1-p)}p \\ &= 1-(1-p)^x, \quad x = 1, 2, \ldots. \end{aligned}$$

(iv)

$$\lim_{x\to\infty} F_X(x) = \lim_{x\to\infty} 1-(1-p)^x = 1$$