Lecture 10: Grouped Summaries

STAT 450, Fall 2021

Today we discuss the `dplyr` functions `summarise()` and `group_by()` which can be used to compute summary statistics across different groups.

```
library(tidyverse)
library(nycflights13)
```

Using the `mpg` data frame, the following code gives the count, mean city mileage, and mean highway mileage for each category of `class` (car type):

```
mpg %>%
  group_by(class) %>%
  summarise(
    count = n(),
    hwy_mean = mean(hwy),
    city_mean = mean(cty)
  )
```

```
## # A tibble: 7 x 4
##   class      count hwy_mean city_mean
##   <chr>      <int>    <dbl>     <dbl>
## 1 2seater        5     24.8      15.4
## 2 compact       47     28.3      20.1
## 3 midsize       41     27.3      18.8
## 4 minivan       11     22.4      15.8
## 5 pickup        33     16.9      13
## 6 subcompact    35     28.1      20.4
## 7 suv           62     18.1      13.5
```

Using the `flights` data frame, the following code gives the count and average departure delay (in minutes) on each date:

```
flights %>%
  group_by(year, month, day) %>%
  summarise(
    count = n(),
    delay = mean(dep_delay, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'year', 'month'. You can override using the `.gro
```

```
## # A tibble: 365 x 5
## # Groups:   year, month [12]
##     year month   day count delay
##    <int> <int> <int> <int> <dbl>
##  1  2013     1     1   842 11.5
##  2  2013     1     2   943 13.9
##  3  2013     1     3   914 11.0
##  4  2013     1     4   915  8.95
##  5  2013     1     5   720  5.73
##  6  2013     1     6   832  7.15
##  7  2013     1     7   933  5.42
##  8  2013     1     8   899  2.55
##  9  2013     1     9   902  2.28
## 10  2013     1    10   932  2.84
## # ... with 355 more rows
```

The argument `na.rm = TRUE` specifies to remove the missing data (`NA` values) when computing the grouped means.

To see the entire data set use `View()`, which will open the data set in the RStudio viewer.

```
flights %>%
  group_by(year, month, day) %>%
  summarise(
    count = n(),
    delay = mean(dep_delay, na.rm = TRUE)) %>%
  View()
```

Exercises:

1. Using the `mpg` data frame, for each car manufacturer, compute the count, average city miles per gallon, and average highway miles per gallon.
2. For each carrier, compute the number of flights and median departure delay. Which carrier has the longest median departure delay?
3. For each carrier, compute the proportion of flights that were delayed more than 5 minutes (i.e., the departure delay was greater than 5 minutes). Which carrier has the smallest proportion of delays?

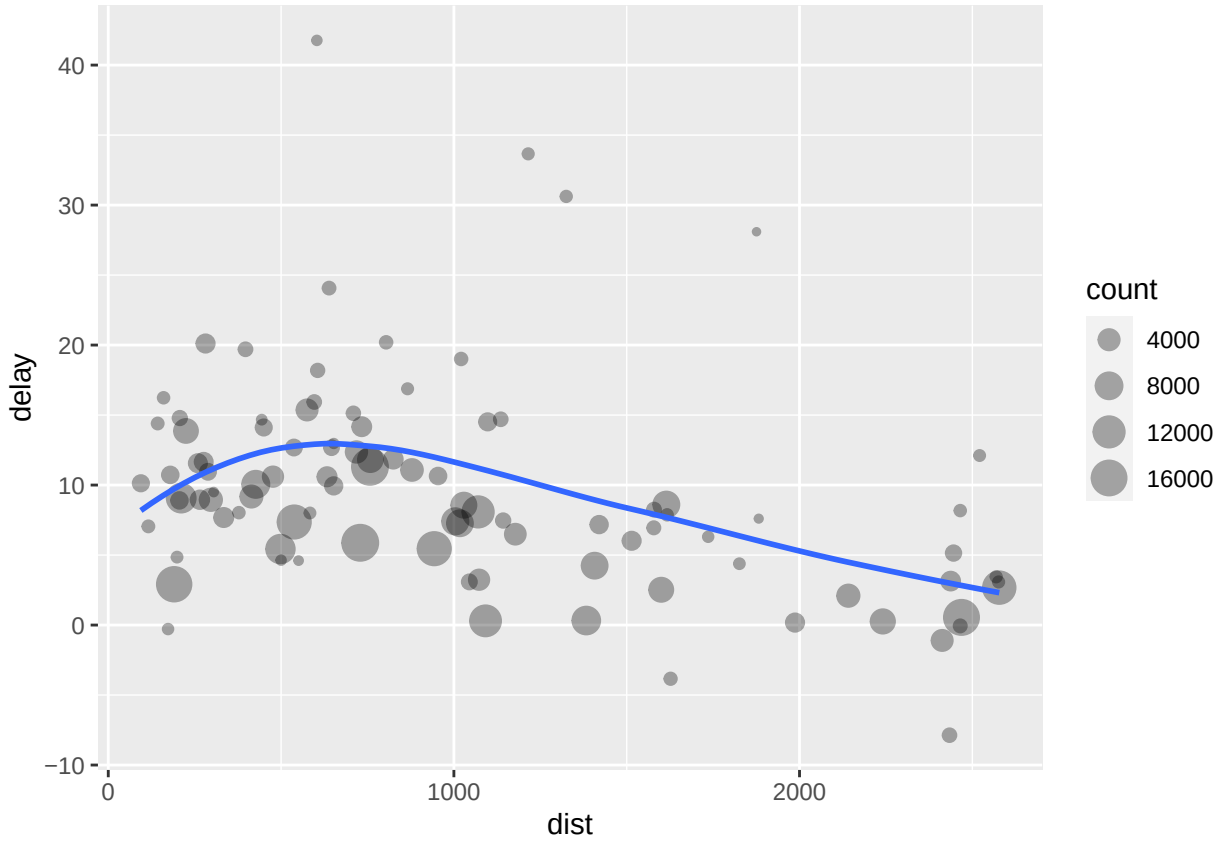
Suppose we want to explore the relationship between the distance and average arrival delay for each destination. Using what we know about `dplyr`, we might write code like this:

```
delays <- flights %>%
  group_by(dest) %>%
  summarise(
    count = n(),
    dist = mean(distance, na.rm = TRUE),
    delay = mean(arr_delay, na.rm = TRUE)
  ) %>%
  filter(count > 20, dest != "HNL")
delays
```

```
## # A tibble: 96 x 4
##    dest  count  dist delay
##    <chr> <int> <dbl> <dbl>
##  1 ABQ     254 1826   4.38
##  2 ACK     265  199   4.85
##  3 ALB     439  143  14.4
##  4 ATL   17215  757. 11.3
##  5 AUS    2439 1514.  6.02
##  6 AVL     275  584.  8.00
##  7 BDL     443  116   7.05
##  8 BGR     375  378   8.03
##  9 BHM     297  866. 16.9
## 10 BNA    6333  758. 11.8
## # ... with 86 more rows
```

Next, use `ggplot2` to visualize the relationship:

```
ggplot(data = delays, aes(x = dist, y = delay)) +
  geom_point(aes(size = count), alpha = 1/3) +
  geom_smooth(se = FALSE)
```



It looks like delays increase with distance up to 750 miles and then decrease.