

# Data Science and R: tidyverse

Chang-Ye Tu

- 1 Data Wrangling: `dplyr`
  - Choosing columns: `select`, `rename`
  - Choosing rows: `filter`, `between`, `slice`, `slice_sample`, `slice_max`, `distinct`
  - Adding new variables: `mutate`, `transmute`, `rownames_to_column`
  - Grouping and counting: `group_by`, `summarize`, `tally`, `count`, `group_size`, `n_groups`, `ungroup`
  - Creating data frames: `tibble`
  - Viewing more output: `print`, `View`
  - Plotting: `ggplot2`
- 2 Data Visualization: `ggplot2`
- 3 Data Importing and Tidying: `readr`, `lubridate`, `stringr`

## Data Wrangling: `dplyr`

# Prerequisites

- loading `tidyverse`

```
library(tidyverse)
```

- loading dataset `nycflights13`

```
library(nycflights13)
```

- print the sub dataset `flights` from `nycflights13`

```
flights
# # A tibble: 336,776 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
# 7  2013     1     1     555             600        -5       913           854
# 8  2013     1     1     557             600        -3       709           723
# 9  2013     1     1     557             600        -3       838           846
# 10 2013     1     1     558             600        -2       753           745
# # ... with 336,766 more rows, and 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 <dtm>
```

## Choosing columns: `select`, `rename`

- pick columns: `select()`

```
flights %>% select(carrier, flight)
# # A tibble: 336,776 x 2
#   carrier flight
#   <chr>    <int>
# 1 UA      1545
# 2 UA      1714
# 3 AA      1141
# 4 B6       725
# 5 DL       461
# 6 UA      1696
# 7 B6       507
# 8 EV      5708
# 9 B6        79
# 10 AA      301
# # ... with 336,766 more rows
```

- hide columns: `select()` with minus signs

```
flights %>% select(-month, -day)
# # A tibble: 336,776 x 17
#   year dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay
#   <int>   <int>         <int>     <dbl>   <int>         <int>     <dbl>
# 1  2013     517           515         2     830           819         11
# 2  2013     533           529         4     850           830         20
# 3  2013     542           540         2     923           850         33
# 4  2013     544           545        -1    1004          1022        -18
# 5  2013     554           600        -6     812           837        -25
# 6  2013     554           558        -4     740           728         12
# 7  2013     555           600        -5     913           854         19
# 8  2013     557           600        -3     709           723        -14
# 9  2013     557           600        -3     838           846         -8
#10  2013     558           600        -2     753           745          8
# # ... with 336,766 more rows, and 10 more variables: carrier <chr>,
# #   flight <int>, tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>,
# #   distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dtm>
```

- hide a range of columns: `range`

```
flights %>% select(-(dep_time:arr_delay))
```

- hide any column with a matching name: `contains()`

```
flights %>% select(-contains("time"))
```

- pick columns using a character vector of column names

```
cols <- c("carrier", "flight", "tailnum")
flights %>% select(all_of(cols))
```

```
# # A tibble: 336,776 x 3
#   carrier flight tailnum
#   <chr>    <int> <chr>
# 1 UA      1545 N14228
# 2 UA      1714 N24211
# 3 AA      1141 N619AA
# 4 B6        725 N804JB
# 5 DL        461 N668DN
# 6 UA      1696 N39463
# 7 B6        507 N516JB
# 8 EV      5708 N829AS
# 9 B6         79 N593JB
# 10 AA       301 N3ALAA
# # ... with 336,766 more rows
```

- `all_of()` vs `any_of()`

```
cols <- c("carrier", "flight", "tailnum_ttt")
flights %>% select(any_of(cols))

# # A tibble: 336,776 x 2
#   carrier flight
#   <chr>    <int>
# 1 UA      1545
# 2 UA      1714
# 3 AA      1141
# 4 B6        725
# 5 DL        461
# 6 UA      1696
# 7 B6        507
# 8 EV      5708
# 9 B6         79
# 10 AA       301
# # ... with 336,766 more rows
```



- rename columns with that all columns not mentioned are dropped: `select()`

```
flights %>% select(tail = tailnum)
# # A tibble: 336,776 x 1
#   tail
#   <chr>
# 1 N14228
# 2 N24211
# 3 N619AA
# 4 N804JB
# 5 N668DN
# 6 N39463
# 7 N516JB
# 8 N829AS
# 9 N593JB
# 10 N3ALAA
# # ... with 336,766 more rows
```

- rename columns with that all columns not mentioned are kept: `rename()`

```
flights %>% rename(tail = tailnum)
# # A tibble: 336,776 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
# 7  2013     1     1     555           600          -5       913           854
# 8  2013     1     1     557           600          -3       709           723
# 9  2013     1     1     557           600          -3       838           846
#10  2013     1     1     558           600          -2       753           745
# # ... with 336,766 more rows, and 11 more variables: arr_delay <dbl>,
# #   carrier <chr>, flight <int>, tail <chr>, origin <chr>, dest <chr>,
# #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dtm>
```

## Choosing rows: `filter`, `between`, `slice`, `slice_sample`, `slice_max`, `distinct`

- `filter()` supports the use of multiple conditions

```
flights %>% filter(dep_time >= 600, dep_time <= 605)

# # A tibble: 2,460 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     600             600           0     851           858
# 2  2013     1     1     600             600           0     837           825
# 3  2013     1     1     601             600           1     844           850
# 4  2013     1     1     602             610          -8     812           820
# 5  2013     1     1     602             605          -3     821           805
# 6  2013     1     2     600             600           0     814           749
# 7  2013     1     2     600             605          -5     751           818
# 8  2013     1     2     600             600           0     819           815
# 9  2013     1     2     600             600           0     846           846
#10  2013     1     2     600             600           0     737           725
# # ... with 2,450 more rows, and 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 <dtm>
```

- if numeric values fall in a range: use `between()`

```
flights %>% filter(between(dep_time, 600, 605))
```

- `is.na()` is useful when filtering

```
flights %>% filter(!is.na(dep_time))
```

- filter rows by position: `slice()`

```
flights %>% slice(1000:1005)
```

```
# # 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     2     809             810        -1      950           948
# 2  2013     1     2     810             800         10     1008          1014
# 3  2013     1     2     811             815         -4     1100          1056
# 4  2013     1     2     811             815         -4     1126          1131
# 5  2013     1     2     811             820         -9      944           955
# 6  2013     1     2     815             815          0     1109          1128
# # ... with 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 <dtm>
```

- keep the first five rows within each group: `group_by()` then `slice(1:5)`

```
flights %>%
  group_by(month, day) %>%
  slice(1:5)

# # A tibble: 1,825 x 19
# # Groups:   month, day [365]
#   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     2      42            2359          43     518           442
# 7  2013     1     2     126            2250         156     233          2359
# 8  2013     1     2     458             500          -2     703           650
# 9  2013     1     2     512             515          -3     809           819
# 10 2013     1     2     535             540          -5     831           850
# # ... with 1,815 more rows, and 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 <dtm>
```

- sample three rows from each group: `group_by()` then `slice_sample(n = 3)`

```
flights %>%
  group_by(month, day) %>%
  slice_sample(n = 3)

# # A tibble: 1,095 x 19
# # Groups:   month, day [365]
#   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    1424           1349          35    1701         1556
# 2  2013     1     1    1744           1720          24    2052         2025
# 3  2013     1     1    1518           1516           2    1837         1832
# 4  2013     1     2     702           700           2    1054         1008
# 5  2013     1     2    2028           2004          24    2202         2133
# 6  2013     1     2    1848           1840           8    2333         2151
# 7  2013     1     3    1111           1055          16    1420         1405
# 8  2013     1     3    1519           1459          20    1855         1805
# 9  2013     1     3    1427           1420           7    1724         1713
# 10 2013     1     4     758           805          -7     909         919
# # ... with 1,085 more rows, and 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 <dtm>
```

- keep three rows from each group with the top `dep_delay`: `group_by()` then `slice_max(dep_delay, n = 3)`

```
flights %>%
  group_by(month, day) %>%
  slice_max(dep_delay, n = 3)

# # A tibble: 1,108 x 19
# # Groups:   month, day [365]
#   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    2343             1724        379      314         1938
# 3  2013     1     1    1815             1325        290     2120         1542
# 4  2013     1     2    2131             1512        379     2340         1741
# 5  2013     1     2    1607             1030        337     2003         1355
# 6  2013     1     2    1412              838        334     1710         1147
# 7  2013     1     3    2056             1605        291     2239         1754
# 8  2013     1     3    2008             1540        268     2339         1909
# 9  2013     1     3    2012             1600        252     2314         1857
# 10 2013     1     4    2123             1635        288     2332         1856
# # ... with 1,098 more rows, and 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 <dtm>
```

- sort by `dep_delay` within each group: `group_by()`, `slice_max()`, then `arrange(desc(dep_delay))`

```
flights %>%
  group_by(month, day) %>%
  slice_max(dep_delay, n = 3) %>%
  arrange(desc(dep_delay))

# # A tibble: 1,108 x 19
# # Groups:   month, day [365]
#   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
# 6  2013     4    10    1100            1900         960      1342          2211
# 7  2013     3    17    2321             810         911       135          1020
# 8  2013     6    27     959            1900         899      1236          2226
# 9  2013     7    22    2257             759         898       121          1026
# 10 2013    12     5     756            1700         896      1058          2020
# # ... with 1,098 more rows, and 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 <dtm>
```



- unique rows can be identified using `unique()` from base R

```
flights %>%  
  select(origin, dest) %>%  
  unique()  
  
# # A tibble: 224 x 2  
#   origin dest  
#   <chr> <chr>  
# 1 EWR    IAH  
# 2 LGA    IAH  
# 3 JFK    MIA  
# 4 JFK    BQN  
# 5 LGA    ATL  
# 6 EWR    ORD  
# 7 EWR    FLL  
# 8 LGA    IAD  
# 9 JFK    MCO  
# 10 LGA    ORD  
# # ... with 214 more rows
```

- **tidyverse** provides an alternative that is more efficient: `distinct`

```
flights %>%  
  select(origin, dest) %>%  
  distinct()  
  
# you don't have to include the parentheses if there are no arguments
```

## Adding new variables: `mutate`, `transmute`, `rownames_to_column`

- create new variables and keep all existing variables: `mutate()`

```
flights %>% mutate(speed = distance / air_time * 60)

# # A tibble: 336,776 x 20
#   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
# 7  2013     1     1     555             600          -5       913           854
# 8  2013     1     1     557             600          -3       709           723
# 9  2013     1     1     557             600          -3       838           846
# 10 2013     1     1     558             600          -2       753           745
# # ... with 336,766 more rows, and 12 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 <dtm>,
# #   speed <dbl>
```

- create new variables only: `transmute()`

```
flights %>% transmute(speed = distance / air_time * 60)
# # A tibble: 336,776 x 1
#   speed
#   <dbl>
# 1  370.
# 2  374.
# 3  408.
# 4  517.
# 5  394.
# 6  288.
# 7  404.
# 8  259.
# 9  405.
# 10 319.
# # ... with 336,766 more rows
```

- `rownames_to_column()` turns row names into an explicit variable

```
mtcars %>%
  rownames_to_column("model") %>%
  head()

#           model mpg cyl disp  hp drat   wt  qsec vs am gear carb
# 1      Mazda RX4 21.0   6  160 110 3.90 2.620 16.46 0  1    4    4
# 2      Mazda RX4 Wag 21.0   6  160 110 3.90 2.875 17.02 0  1    4    4
# 3      Datsun 710 22.8   4  108  93 3.85 2.320 18.61 1  1    4    1
# 4    Hornet 4 Drive 21.4   6  258 110 3.08 3.215 19.44 1  0    3    1
# 5 Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02 0  0    3    2
# 6      Valiant 18.1   6  225 105 2.76 3.460 20.22 1  0    3    1

# tidyverse does not print row names for local data frames
mtcars %>% as_tibble()

# # A tibble: 32 x 11
#       mpg   cyl  disp    hp  drat    wt   qsec    vs  am  gear  carb
#   <dbl> <dbl> <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
# 1  21       6   160    110   3.9   2.62  16.5     0     1     4     4
# 2  21       6   160    110   3.9   2.88  17.0     0     1     4     4
# 3 22.8      4   108     93   3.85   2.32  18.6     1     1     4     1
# 4 21.4      6   258    110   3.08   3.22  19.4     1     0     3     1
# 5 18.7      8   360    175   3.15   3.44  17.0     0     0     3     2
# 6 18.1      6   225    105   2.76   3.46  20.2     1     0     3     1
# 7 14.3      8   360    245   3.21   3.57  15.8     0     0     3     4
# 8 24.4      4   147.     62   3.69   3.19   20      1     0     4     2
# 9 22.8      4   141.     95   3.92   3.15  22.9     1     0     4     2
# 10 19.2      6   168.    123   3.92   3.44  18.3     1     0     4     4
# # ... with 22 more rows
```

## Grouping and counting: `group_by`, `summarize`, `tally`, `count`, `group_size`, `n_groups`, `ungroup`

- count the number of rows in each group: `group_by()` then `summarize(cnt = n())`

```
flights %>%  
  group_by(month) %>%  
  summarize(cnt = n())
```

```
# # A tibble: 12 x 2
```

```
#   month    cnt
```

```
#   <int> <int>
```

```
# 1     1 27004
```

```
# 2     2 24951
```

```
# 3     3 28834
```

```
# 4     4 28330
```

```
# 5     5 28796
```

```
# 6     6 28243
```

```
# 7     7 29425
```

```
# 8     8 29327
```

```
# 9     9 27574
```

```
# 10    10 28889
```

```
# 11    11 27268
```

```
# 12    12 28135
```

- count the number of rows in each group: `group_by()` then `tally()`

```
flights %>%  
  group_by(month) %>%  
  tally()  
  
# # A tibble: 12 x 2  
#   month     n  
#   <int> <int>  
# 1     1 27004  
# 2     2 24951  
# 3     3 28834  
# 4     4 28330  
# 5     5 28796  
# 6     6 28243  
# 7     7 29425  
# 8     8 29327  
# 9     9 27574  
# 10    10 28889  
# 11    11 27268  
# 12    12 28135
```

- count the number of rows in each group: `count()`

```
flights %>% count(month)
```

```
# # A tibble: 12 x 2
```

```
#   month     n
```

```
#   <int> <int>
```

```
# 1     1 27004
```

```
# 2     2 24951
```

```
# 3     3 28834
```

```
# 4     4 28330
```

```
# 5     5 28796
```

```
# 6     6 28243
```

```
# 7     7 29425
```

```
# 8     8 29327
```

```
# 9     9 27574
```

```
# 10    10 28889
```

```
# 11    11 27268
```

```
# 12    12 28135
```

- you can sort by the count: `arrange(desc(cnt))`

```
flights %>%  
  group_by(month) %>%  
  summarize(cnt = n()) %>%  
  arrange(desc(cnt))  
  
# # A tibble: 12 x 2  
#   month    cnt  
#   <int> <int>  
# 1     7 29425  
# 2     8 29327  
# 3    10 28889  
# 4     3 28834  
# 5     5 28796  
# 6     4 28330  
# 7     6 28243  
# 8    12 28135  
# 9     9 27574  
# 10    11 27268  
# 11     1 27004  
# 12     2 24951
```



- `tally()` and `count()` have a `sort` parameter for this purpose

```
flights %>%  
  group_by(month) %>%  
  tally(sort = TRUE)  
flights %>% count(month, sort = TRUE)
```

- you can sum over a specific variable instead of simply counting rows

```
flights %>%  
  group_by(month) %>%  
  summarize(dist = sum(distance))
```

```
# # A tibble: 12 x 2
```

```
#   month      dist
```

```
#   <int>    <dbl>
```

```
# 1      1 27188805
```

```
# 2      2 24975509
```

```
# 3      3 29179636
```

```
# 4      4 29427294
```

```
# 5      5 29974128
```

```
# 6      6 29856388
```

```
# 7      7 31149199
```

```
# 8      8 31149334
```

```
# 9      9 28711426
```

```
# 10     10 30012086
```

```
# 11     11 28639718
```

```
# 12     12 29954084
```

- `tally()` and `count()` have a `wt` parameter for this purpose

```
flights %>%  
  group_by(month) %>%  
  tally(wt = distance)  
flights %>% count(month, wt = distance)
```

- return the counts as a vector: `group_size()`

```
flights %>%  
  group_by(month) %>%  
  group_size()  
# [1] 27004 24951 28834 28330 28796 28243 29425 29327 27574 28889 27268 28135
```

- report the number of groups: `n_groups()`

```
flights %>%  
  group_by(month) %>%  
  n_groups()  
# [1] 12
```

- group by two variables, `summarize()`, `arrange()` (output is possibly confusing)

```
flights %>%
  group_by(month, day) %>%
  summarize(cnt = n()) %>%
  arrange(desc(cnt)) %>%
  print(n = 10)

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

# # A tibble: 365 x 3
# # Groups:   month [12]
#   month    day    cnt
#   <int> <int> <int>
# 1     11     27  1014
# 2      7     11  1006
# 3      7      8  1004
# 4      7     10  1004
# 5     12      2  1004
# 6      7     18  1003
# 7      7     25  1003
# 8      7     12  1002
# 9      7      9  1001
# 10     7     17  1001
# # ... with 355 more rows
```

- arrange across all groups: `ungroup()` before arranging

```
flights %>%
  group_by(month, day) %>%
  summarize(cnt = n()) %>%
  ungroup() %>%
  arrange(desc(cnt))

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

# # A tibble: 365 x 3
#   month   day   cnt
#   <int> <int> <int>
# 1     11    27  1014
# 2      7    11  1006
# 3      7     8  1004
# 4      7    10  1004
# 5     12     2  1004
# 6      7    18  1003
# 7      7    25  1003
# 8      7    12  1002
# 9      7     9  1001
# 10     7    17  1001
# # ... with 355 more rows
```

# Creating data frames: `tibble`

- `tibble()` is a better way than `data.frame()` for creating data frames. Benefits of `tibble()`:
  - You can use previously defined columns to compute new columns.
  - It never coerces column types.
  - It never munges column names.
  - It never adds row names.
  - It only recycles length 1 input.
  - It returns a local data frame (a `tbl_df`).
- `tibble()`, `data.frame()` examples

```
tibble(a = 1:6, b = a * 2, c = "string", "d+e" = 1) %>% glimpse()
# Rows: 6
# Columns: 4
# $ a      <int> 1, 2, 3, 4, 5, 6
# $ b      <dbl> 2, 4, 6, 8, 10, 12
# $ c      <chr> "string", "string", "string", "string", "string", "string"
# $ `d+e`  <dbl> 1, 1, 1, 1, 1, 1

data.frame(a = 1:6, c = "string", "d+e" = 1) %>% glimpse()
# Rows: 6
# Columns: 3
# $ a      <int> 1, 2, 3, 4, 5, 6
# $ c      <chr> "string", "string", "string", "string", "string", "string"
# $ d.e    <dbl> 1, 1, 1, 1, 1, 1
```

## Viewing more output: `print`, `View`

- specify that you want to see more rows

```
flights %>% print(n = 15)
# # A tibble: 336,776 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
# 7 2013     1     1     555           600          -5     913           854
# 8 2013     1     1     557           600          -3     709           723
# 9 2013     1     1     557           600          -3     838           846
#10 2013     1     1     558           600          -2     753           745
#11 2013     1     1     558           600          -2     849           851
#12 2013     1     1     558           600          -2     853           856
#13 2013     1     1     558           600          -2     924           917
#14 2013     1     1     558           600          -2     923           937
#15 2013     1     1     559           600          -1     941           910
# # ... with 336,761 more rows, and 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 <dtm>
```

- specify that you want to see ALL rows (don't run this!)

```
flights %>% print(n = Inf)
```

- specify that you want to see all columns

```
flights %>% print(width = Inf)

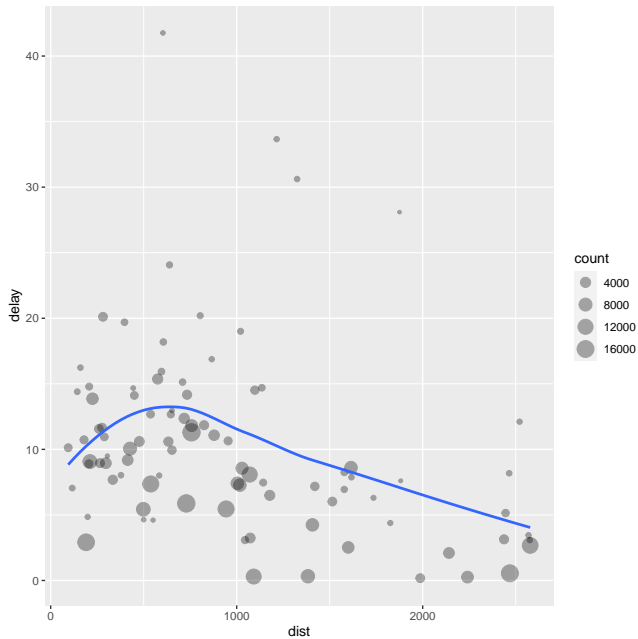
# # A tibble: 336,776 x 19
#   year month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
#   <int> <int> <int>   <int>         <int>         <dbl>   <int>         <int>
# 1 2013     1     1     517             515           2       830           819
# 2 2013     1     1     533             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
# 7 2013     1     1     555             600          -5       913           854
# 8 2013     1     1     557             600          -3       709           723
# 9 2013     1     1     557             600          -3       838           846
# 10 2013     1     1     558             600          -2       753           745
#   arr_delay carrier flight tailnum origin dest air_time distance hour minute
#   <dbl> <chr>   <int> <chr>   <chr> <chr>   <dbl>   <dbl> <dbl> <dbl>
# 1      11 UA      1545 N14228 EWR   IAH      227    1400     5     15
# 2      20 UA      1714 N24211 LGA   IAH      227    1416     5     29
# 3      33 AA      1141 N619AA JFK   MIA      160    1089     5     40
# 4     -18 B6       725 N804JB JFK   BQN      183    1576     5     45
# 5     -25 DL       461 N668DN LGA   ATL      116     762     6      0
# 6      12 UA      1696 N39463 EWR   ORD      150     719     5     58
# 7      19 B6       507 N516JB EWR   FLL      158    1065     6      0
# 8     -14 EV      5708 N829AS LGA   IAD       53     229     6      0
# 9      -8 B6       79 N593JB JFK   MCO      140     944     6      0
# 10     8 AA       301 N3ALAA LGA   ORD      138     733     6      0
#   time_hour
#   <dtm>
# 1 2013-01-01 05:00:00
# 2 2013-01-01 05:00:00
# 3 2013-01-01 05:00:00
# 4 2013-01-01 05:00:00
# 5 2013-01-01 06:00:00
# 6 2013-01-01 05:00:00
# 7 2013-01-01 06:00:00
# 8 2013-01-01 06:00:00
# 9 2013-01-01 06:00:00
# 10 2013-01-01 06:00:00
# # ... with 336,766 more rows
```

## Plotting: ggplot2

```
library(ggplot2)

flights %>%
  group_by(dest) %>%
  summarize(
    count = n(),
    dist = mean(distance, na.rm = TRUE),
    delay = mean(arr_delay, na.rm = TRUE)
  ) %>%
  filter(delay > 0, count > 20, dest != "HNL") %>%
  ggplot(mapping = aes(x = dist, y = delay)) +
    geom_point(aes(size = count), alpha = 1 / 3) +
    geom_smooth(se = FALSE)
```





## Data Visualization: `ggplot2`

## Data Importing and Tidying: `readr`, `lubridate`, `stringr`