Data wrangling

El Mex

Contents

1 Basic functions from `dplyr`.

Needed packages

```
library(dplyr)
library(ggplot2)
library(nycflights13)
```

```
# explore the dataframe called ‘flights‘
glimpse(flights)
```

```
## Rows: 336,776
## Columns: 19
## $ year           <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013...
## $ month          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ day            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ dep_time       <int> 517, 533, 542, 544, 554, 554, 555, 557, 557, 558, 55...
## $ sched_dep_time <int> 515, 529, 540, 545, 600, 558, 600, 600, 600, 600, 60...
## $ dep_delay      <dbl> 2, 4, 2, -1, -6, -4, -5, -3, -3, -2, -2, -2, -2, -2,...
## $ arr_time       <int> 830, 850, 923, 1004, 812, 740, 913, 709, 838, 753, 8...
## $ sched_arr_time <int> 819, 830, 850, 1022, 837, 728, 854, 723, 846, 745, 8...
## $ arr_delay      <dbl> 11, 20, 33, -18, -25, 12, 19, -14, -8, 8, -2, -3, 7,...
## $ carrier        <chr> "UA", "UA", "AA", "B6", "DL", "UA", "B6", "EV", "B6"...
## $ flight         <int> 1545, 1714, 1141, 725, 461, 1696, 507, 5708, 79, 301...
## $ tailnum        <chr> "N14228", "N24211", "N619AA", "N804JB", "N668DN", "N...
## $ origin         <chr> "EWR", "LGA", "JFK", "JFK", "LGA", "EWR", "EWR", "LG...
## $ dest           <chr> "IAH", "IAH", "MIA", "BQN", "ATL", "ORD", "FLL", "IA...
## $ air_time       <dbl> 227, 227, 160, 183, 116, 150, 158, 53, 140, 138, 149...
## $ distance       <dbl> 1400, 1416, 1089, 1576, 762, 719, 1065, 229, 944, 73...
## $ hour           <dbl> 5, 5, 5, 5, 6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 5, 6, 6...
## $ minute         <dbl> 15, 29, 40, 45, 0, 58, 0, 0, 0, 0, 0, 0, 0, 0, 0, 59...
## $ time_hour      <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-0...
```

1.1 filter rows

```
# flights from New York City to Portland, Oregon (destination code: "PDX")
portland_flights <- flights %>%
  filter(dest == "PDX")


# see the first 6 rows
head(portland_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     1739           1740        -1     2051           2112
## 2  2013     1     1     1805           1757         8     2117           2119
## 3  2013     1     1     2052           2029        23     2349           2350
## 4  2013     1     2      804            805        -1     1039           1110
## 5  2013     1     2     1552           1550         2     1853           1922
## 6  2013     1     2     1727           1720         7     2042           2040
## # ... 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 <dttm>
```

Let's begin with an exercise

```
# filter all rows from JFK that were heading to Burlington ("BTV") or Seattle ("SEA") &
# in the months of October, November, or December.
btv_sea_flights_fall <- flights %>%
  filter(origin == "JFK" & (dest == "BTV" | dest == "SEA") & month >= 10)

# see the first 6 rows
head(btv_sea_flights_fall)
```

```
## # 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    10     1      729            735        -6     1049           1040
## 2  2013    10     1      853            900        -7     1217           1157
## 3  2013    10     1      916            925        -9     1016           1033
## 4  2013    10     1     1216           1221        -5     1326           1328
## 5  2013    10     1     1452           1459        -7     1602           1622
## 6  2013    10     1     1459           1500        -1     1817           1829
## # ... 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 <dttm>
```

We could use a comma instead of "&" to get the same results

```
btv_sea_flights_fall <- flights %>%
  filter(origin == "JFK", (dest == "BTV" | dest == "SEA"), month >= 10)
```

Using `!` "not" operator to pick rows that don't match a criteria

```
# filtering rows to flights that didn't go to Burlington, "BTV" or Seattle, "SEA"
not_BTV_SEA <- flights %>%
  filter(!(dest == "BTV" | dest == "SEA"))

head(not_BTV_SEA)
```

```
## # 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
## # ... 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 <dttm>
```

Warning: note the parentheses around the (dest == "BTV" | dest == "SEA"). Let's say we put it this way:

`flights %>% filter(!dest == "BTV" | dest == "SEA")`

This would give us all flights not headed to "BTV" or those headed to "SEA", clearly a different result.

Now say we have a larger number of airports we want to filter for, say "SEA", "SFO", "PDX", "BTV", and "BDL". Using the `|` (or) operator isn't quite practical

```
many_airports <- flights %>%
  filter(dest == "SEA" | dest == "SFO" | dest == "PDX" |
         dest == "BTV" | dest == "BDL")
```

A better idea is to use the `%in%` operator

```
many_airports <- flights %>%
  filter(dest %in% c("SEA", "SFO", "PDX", "BTV", "BDL"))

head(many_airports)
```

```
## # 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      558            600        -2      923            937
## 2  2013     1     1      611            600        11      945            931
## 3  2013     1     1      655            700        -5     1037           1045
## 4  2013     1     1      724            725        -1     1020           1030
## 5  2013     1     1      729            730        -1     1049           1115
## 6  2013     1     1      734            737        -3     1047           1113
## # ... 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 <dttm>
```

1.2 summarize variables

Calculate summary stats of the `temp` variable in the `weather` data frame

```
# create variable 'summary_temp' & get both, mean and standard deviation of 'temp'
summary_temp <- weather %>%
  summarize(mean = mean(temp), std_dev = sd(temp))

# check it out
head(summary_temp)
```

```
## # A tibble: 1 x 2
##    mean std_dev
##   <dbl>   <dbl>
## 1    NA      NA
```

Why did we get both `NA` as result? it's because we have some `NA` (missing values) on the the dataset. Let's use `na.rm = TRUE` to ignore any `NA`

```
summary_temp <- weather %>%
  summarize(mean = mean(temp, na.rm = TRUE),
          std_dev = sd(temp, na.rm = TRUE))

# check it out
head(summary_temp)
```

```
## # A tibble: 1 x 2
##    mean std_dev
##   <dbl>   <dbl>
## 1  55.3    17.8
```

1.3 group_by rows

Instead of a single mean temperature for the whole year, you would like one for each of the 12 months separately

```r
summary_monthly_temp <- weather %>%
  group_by(month) %>%
  summarize(mean = mean(temp, na.rm = TRUE),
            std_dev = sd(temp, na.rm = TRUE))

#check it out
summary_monthly_temp
```

```
## # A tibble: 12 x 3
##    month  mean std_dev
##    <int> <dbl>   <dbl>
##  1     1  35.6   10.2
##  2     2  34.3    6.98
##  3     3  39.9    6.25
##  4     4  51.7    8.79
##  5     5  61.8    9.68
##  6     6  72.2    7.55
##  7     7  80.1    7.12
##  8     8  74.5    5.19
##  9     9  67.4    8.47
## 10    10  60.1    8.85
## 11    11  45.0   10.4
## 12    12  38.4    9.98
```

Let's use the `n()` counting summary function (it counts rows). Suppose we'd like to count how many flights departed each of the three airports in New York City

```r
by_origin <- flights %>%
  group_by(origin) %>%
  summarize(count = n())

#check it out
by_origin
```

```
## # A tibble: 3 x 2
##   origin  count
##   <chr>   <int>
## 1 EWR    120835
## 2 JFK    111279
## 3 LGA    104662
```

Say you want to know the number of flights leaving each of the three New York City airports for each month (grouping by more than one variable)

```r
by_origin_monthly <- flights %>%
  group_by(origin, month) %>%
  summarize(count = n())

# check it out
head(by_origin_monthly)
```

```
## # A tibble: 6 x 3
## # Groups:   origin [1]
##   origin month count
##   <chr>  <int> <int>
## 1 EWR        1  9893
## 2 EWR        2  9107
## 3 EWR        3 10420
## 4 EWR        4 10531
## 5 EWR        5 10592
## 6 EWR        6 10175
```

When grouping by more than two variables, remember to include all variables at the same time in the same `group_by()`. Otherwise, look at this:

```r
by_origin_monthly_incorrect <- flights %>%
  group_by(origin) %>%
  group_by(month) %>%
  summarize(count = n())

# check it out
by_origin_monthly_incorrect
```

```
## # A tibble: 12 x 2
##    month count
##    <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
```

Here `group_by(month)` overwrote the grouping structure meta-data of the earlier `group_by(origin)`

1.4 `mutate` existing variables

Convert temperatures from °F to °C with the formula: temp in F - 32 / 1.8

```r
# use `temp` variable for calculations
weather <- weather %>%
  mutate(temp_in_C = (temp - 32) / 1.8)

# check it out
head(weather)
```

```
## # A tibble: 6 x 16
##   origin  year month   day  hour  temp  dewp humid wind_dir wind_speed wind_gust
##   <chr>  <int> <int> <int> <int> <dbl> <dbl> <dbl>    <dbl>      <dbl>     <dbl>
## 1 EWR     2013     1     1     1  39.0  26.1  59.4      270      10.4         NA
## 2 EWR     2013     1     1     2  39.0  27.0  61.6      250       8.06        NA
## 3 EWR     2013     1     1     3  39.0  28.0  64.4      240      11.5         NA
## 4 EWR     2013     1     1     4  39.9  28.0  62.2      250      12.7         NA
## 5 EWR     2013     1     1     5  39.0  28.0  64.4      260      12.7         NA
## 6 EWR     2013     1     1     6  37.9  28.0  67.2      240      11.5         NA
## # ... with 5 more variables: precip <dbl>, pressure <dbl>, visib <dbl>,
## #   time_hour <dttm>, temp_in_C <dbl>
```

Let's now compute monthly average temperatures in both °F and °C using the `group_by()` and `summarize()`

```
summary_monthly_temp <- weather %>%
  group_by(month) %>%
  summarize(mean_temp_in_F = mean(temp, na.rm = TRUE),
            mean_temp_in_C = mean(temp_in_C, na.rm = TRUE))

# check it out
summary_monthly_temp
```

```
## # A tibble: 12 x 3
##    month mean_temp_in_F mean_temp_in_C
##    <int>          <dbl>          <dbl>
##  1     1           35.6           2.02
##  2     2           34.3           1.26
##  3     3           39.9           4.38
##  4     4           51.7          11.0
##  5     5           61.8          16.6
##  6     6           72.2          22.3
##  7     7           80.1          26.7
##  8     8           74.5          23.6
##  9     9           67.4          19.7
## 10    10           60.1          15.6
## 11    11           45.0           7.22
## 12    12           38.4           3.58
```

Passengers are frustrated when their flight departs late, but aren't as annoyed if pilots can make up some time during the flight. This is known in the airline industry as gain. Let's create the variable "gain"

```
flights <- flights %>%
  mutate(gain = dep_delay - arr_delay)

# check it out
head(flights)
```

```
## # A tibble: 6 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
## # ... with 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 <dttm>, gain <dbl>
```

The flight in the first row departed 2 minutes late but arrived 11 minutes late, so its "gained time in the air" is a loss of 9 minutes, hence its gain is 2 - 11 = -9. On the other hand, the flight in the fourth row departed a minute early (dep_delay of -1) but arrived 18 minutes early (arr_delay of -18), so its "gained time in the air" is 17 minutes, hence its gain is +17

Let's look at some summary stats of `gain`

```r
gain_summary <- flights %>%
  summarize(
    min = min(gain, na.rm = TRUE),
    q1 = quantile(gain, 0.25, na.rm = TRUE),
    median = quantile(gain, 0.5, na.rm = TRUE),
    q3 = quantile(gain, 0.75, na.rm = TRUE),
    max = max(gain, na.rm = TRUE),
    mean = mean(gain, na.rm = TRUE),
    sd = sd(gain, na.rm = TRUE),
    missing = sum(is.na(gain))
  )

gain_summary
```

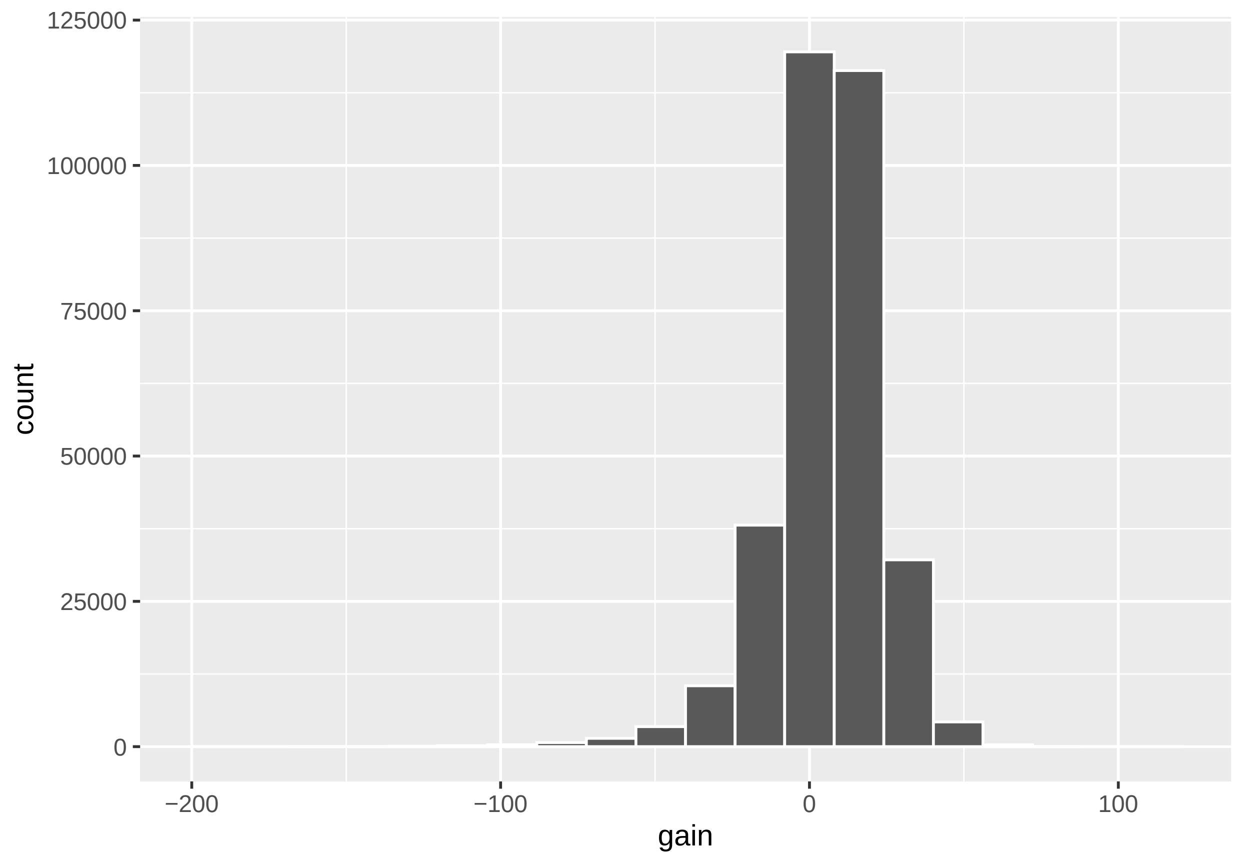
```
## # A tibble: 1 x 8
##     min    q1 median    q3   max  mean    sd missing
##   <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>   <int>
## 1  -196    -3      7    17   109  5.66  18.0    9430
```

This code was a bit long to type, we'll see a succint way to do it by using `skim()` **from** `skimr` **package later on**

We can visualize gain as it is a numerical variable. Let's make an histogram

```r
ggplot(data = flights, mapping = aes(x = gain)) +
  geom_histogram(color = "white", bins = 20)
```

```
## Warning: Removed 9430 rows containing non-finite values (stat_bin).
```



Finally, we can create multiple new variables at once in the same `mutate()` code

```r
flights <- flights %>%
  mutate(
    gain = dep_delay - arr_delay,
    hours = air_time / 60,
    gain_per_hour = gain / hours
  )

head(flights)
```

```
## # A tibble: 6 x 22
##    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
## # ... with 14 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>, gain <dbl>, hours <dbl>,
## #   gain_per_hour <dbl>
```

1.5 arrange and sort rows

Suppose we are interested in determining the most frequent destination airports for all domestic flights departing from New York City in 2013

```
freq_dest <- flights %>%
  group_by(dest) %>%
  summarize(num_flights = n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
#check it out
head(freq_dest)
```

```
## # A tibble: 6 x 2
##   dest  num_flights
##   <chr>       <int>
## 1 ABQ           254
## 2 ACK           265
## 3 ALB           439
## 4 ANC             8
## 5 ATL         17215
## 6 AUS          2439
```

Observe that by default the rows are sorted in alphabetical order. Say instead we would like to see it sorted from the most to the least number of flights (`num_flights`) instead

```
freq_dest_default <- freq_dest %>%
  arrange(num_flights)

# check it out
head(freq_dest_default, 10)
```

```
## # A tibble: 10 x 2
##    dest  num_flights
##    <chr>       <int>
##  1 LEX             1
##  2 LGA             1
##  3 ANC             8
##  4 SBN            10
##  5 HDN            15
##  6 MTJ            15
##  7 EYW            17
##  8 PSP            19
##  9 JAC            25
## 10 BZN            36
```

As we see, "ascending" order is default. To switch the ordering to be in "descending" order, we use the `desc()`

```
freq_dest_desc <- freq_dest %>%
  arrange(desc(num_flights))

head(freq_dest_desc, 10)
```

```
## # A tibble: 10 x 2
##    dest  num_flights
##    <chr>       <int>
##  1 ORD         17283
##  2 ATL         17215
##  3 LAX         16174
##  4 BOS         15508
##  5 MCO         14082
##  6 CLT         14064
##  7 SFO         13331
##  8 FLL         12055
##  9 MIA         11728
## 10 DCA          9705
```

1.6 join data frames

In both the flights and airlines data frames, the key variable we want to join/merge/match the rows by has the same name: carrier. Let's use the inner_join() function to join the two data frames

```
flights_joined <- flights %>%
  inner_join(airlines, by = "carrier")

# check both data frames
glimpse(flights)
```

```
## Rows: 336,776
## Columns: 22
## $ year           <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013...
## $ month          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ day            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ dep_time       <int> 517, 533, 542, 544, 554, 554, 555, 557, 557, 558, 55...
## $ sched_dep_time <int> 515, 529, 540, 545, 600, 558, 600, 600, 600, 600, 60...
## $ dep_delay      <dbl> 2, 4, 2, -1, -6, -4, -5, -3, -3, -2, -2, -2, -2, -2,...
## $ arr_time       <int> 830, 850, 923, 1004, 812, 740, 913, 709, 838, 753, 8...
## $ sched_arr_time <int> 819, 830, 850, 1022, 837, 728, 854, 723, 846, 745, 8...
## $ arr_delay      <dbl> 11, 20, 33, -18, -25, 12, 19, -14, -8, 8, -2, -3, 7,...
## $ carrier        <chr> "UA", "UA", "AA", "B6", "DL", "UA", "B6", "EV", "B6"...
## $ flight         <int> 1545, 1714, 1141, 725, 461, 1696, 507, 5708, 79, 301...
## $ tailnum        <chr> "N14228", "N24211", "N619AA", "N804JB", "N668DN", "N...
## $ origin         <chr> "EWR", "LGA", "JFK", "JFK", "LGA", "EWR", "EWR", "LG...
## $ dest           <chr> "IAH", "IAH", "MIA", "BQN", "ATL", "ORD", "FLL", "IA...
## $ air_time       <dbl> 227, 227, 160, 183, 116, 150, 158, 53, 140, 138, 149...
## $ distance       <dbl> 1400, 1416, 1089, 1576, 762, 719, 1065, 229, 944, 73...
## $ hour           <dbl> 5, 5, 5, 5, 6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 5, 6, 6...
## $ minute         <dbl> 15, 29, 40, 45, 0, 58, 0, 0, 0, 0, 0, 0, 0, 0, 0, 59...
## $ time_hour      <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-0...
## $ gain           <dbl> -9, -16, -31, 17, 19, -16, -24, 11, 5, -10, 0, 1, -9...
## $ hours          <dbl> 3.7833333, 3.7833333, 2.6666667, 3.0500000, 1.933333...
## $ gain_per_hour  <dbl> -2.3788546, -4.2290749, -11.6250000, 5.5737705, 9.82...
```

```
glimpse(flights_joined)
```

```
## Rows: 336,776
## Columns: 23
## $ year           <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013...
## $ month          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ day            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ dep_time       <int> 517, 533, 542, 544, 554, 554, 555, 557, 557, 558, 55...
## $ sched_dep_time <int> 515, 529, 540, 545, 600, 558, 600, 600, 600, 600, 60...
## $ dep_delay      <dbl> 2, 4, 2, -1, -6, -4, -5, -3, -3, -2, -2, -2, -2, -2,...
## $ arr_time       <int> 830, 850, 923, 1004, 812, 740, 913, 709, 838, 753, 8...
## $ sched_arr_time <int> 819, 830, 850, 1022, 837, 728, 854, 723, 846, 745, 8...
## $ arr_delay      <dbl> 11, 20, 33, -18, -25, 12, 19, -14, -8, 8, -2, -3, 7,...
## $ carrier        <chr> "UA", "UA", "AA", "B6", "DL", "UA", "B6", "EV", "B6"...
## $ flight         <int> 1545, 1714, 1141, 725, 461, 1696, 507, 5708, 79, 301...
## $ tailnum        <chr> "N14228", "N24211", "N619AA", "N804JB", "N668DN", "N...
## $ origin         <chr> "EWR", "LGA", "JFK", "JFK", "LGA", "EWR", "EWR", "LG...
## $ dest           <chr> "IAH", "IAH", "MIA", "BQN", "ATL", "ORD", "FLL", "IA...
## $ air_time       <dbl> 227, 227, 160, 183, 116, 150, 158, 53, 140, 138, 149...
## $ distance       <dbl> 1400, 1416, 1089, 1576, 762, 719, 1065, 229, 944, 73...
## $ hour           <dbl> 5, 5, 5, 5, 6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 5, 6, 6...
## $ minute         <dbl> 15, 29, 40, 45, 0, 58, 0, 0, 0, 0, 0, 0, 0, 0, 0, 59...
## $ time_hour      <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-0...
## $ gain           <dbl> -9, -16, -31, 17, 19, -16, -24, 11, 5, -10, 0, 1, -9...
## $ hours          <dbl> 3.7833333, 3.7833333, 2.6666667, 3.0500000, 1.933333...
## $ gain_per_hour  <dbl> -2.3788546, -4.2290749, -11.6250000, 5.5737705, 9.82...
## $ name           <chr> "United Air Lines Inc.", "United Air Lines Inc.", "A...
```

```
head(flights_joined, 10)
```

```
## # A tibble: 10 x 23
##     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 15 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>, gain <dbl>, hours <dbl>,
## #   gain_per_hour <dbl>, name <chr>
```

In airports the airport code is in "faa", whereas in flights the airport codes are in "origin" and "dest". In order to join these two data frames by airport code, our `inner_join()` operation will use the `by = c("dest" = "faa")`

```
flights_with_airport_names <- flights %>%
  inner_join(airports, by = c("dest" = "faa"))

# explore it
glimpse(flights_with_airport_names)
```

```
## Rows: 329,174
## Columns: 29
## $ year           <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013...
## $ month          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ day            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ dep_time       <int> 517, 533, 542, 554, 554, 555, 557, 557, 558, 558, 55...
## $ sched_dep_time <int> 515, 529, 540, 600, 558, 600, 600, 600, 600, 600, 60...
## $ dep_delay      <dbl> 2, 4, 2, -6, -4, -5, -3, -3, -2, -2, -2, -2, -2, -1,...
## $ arr_time       <int> 830, 850, 923, 812, 740, 913, 709, 838, 753, 849, 85...
## $ sched_arr_time <int> 819, 830, 850, 837, 728, 854, 723, 846, 745, 851, 85...
## $ arr_delay      <dbl> 11, 20, 33, -25, 12, 19, -14, -8, 8, -2, -3, 7, -14,...
## $ carrier        <chr> "UA", "UA", "AA", "DL", "UA", "B6", "EV", "B6", "AA"...
## $ flight         <int> 1545, 1714, 1141, 461, 1696, 507, 5708, 79, 301, 49,...
## $ tailnum        <chr> "N14228", "N24211", "N619AA", "N668DN", "N39463", "N...
## $ origin         <chr> "EWR", "LGA", "JFK", "LGA", "EWR", "EWR", "LGA", "JF...
## $ dest           <chr> "IAH", "IAH", "MIA", "ATL", "ORD", "FLL", "IAD", "MC...
## $ air_time       <dbl> 227, 227, 160, 116, 150, 158, 53, 140, 138, 149, 158...
## $ distance       <dbl> 1400, 1416, 1089, 762, 719, 1065, 229, 944, 733, 102...
## $ hour           <dbl> 5, 5, 5, 6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 5, 6, 6, 6...
## $ minute         <dbl> 15, 29, 40, 0, 58, 0, 0, 0, 0, 0, 0, 0, 0, 0, 59, 0,...
## $ time_hour      <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-0...
## $ gain           <dbl> -9, -16, -31, 19, -16, -24, 11, 5, -10, 0, 1, -9, 12...
## $ hours          <dbl> 3.7833333, 3.7833333, 2.6666667, 1.9333333, 2.500000...
## $ gain_per_hour  <dbl> -2.3788546, -4.2290749, -11.6250000, 9.8275862, -6.4...
## $ name           <chr> "George Bush Intercontinental", "George Bush Interco...
## $ lat            <dbl> 29.98443, 29.98443, 25.79325, 33.63672, 41.97860, 26...
## $ lon            <dbl> -95.34144, -95.34144, -80.29056, -84.42807, -87.9048...
## $ alt            <dbl> 97, 97, 8, 1026, 668, 9, 313, 96, 668, 19, 26, 126, ...
## $ tz             <dbl> -6, -6, -5, -5, -6, -5, -5, -5, -6, -5, -5, -8, -8, ...
## $ dst            <chr> "A", "A", "A", "A", "A", "A", "A", "A", "A", "A", "A...
## $ tzone          <chr> "America/Chicago", "America/Chicago", "America/New_Y...
```

```
head(flights_with_airport_names)
```

```
## # A tibble: 6 x 29
##    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      554            600        -6      812            837
## 5  2013     1     1      554            558        -4      740            728
## 6  2013     1     1      555            600        -5      913            854
## # ... with 21 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>, gain <dbl>, hours <dbl>,
## #   gain_per_hour <dbl>, name <chr>, lat <dbl>, lon <dbl>, alt <dbl>, tz <dbl>,
## #   dst <chr>, tzone <chr>
```

Let's construct the chain of pipe operators %>% that computes the number of flights from NYC to each destination, but also includes information about each destination airport

```
named_dests <- flights %>%
  group_by(dest) %>%
  summarize(num_flights = n()) %>%
  arrange(desc(num_flights)) %>%
  inner_join(airports, by = c("dest" = "faa")) %>%
  rename(airport_name = name)
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
#explore it
head(named_dests)
```

```
## # A tibble: 6 x 9
##   dest  num_flights airport_name           lat    lon   alt    tz dst   tzone
##   <chr>       <int> <chr>                <dbl>  <dbl> <dbl> <dbl> <chr> <chr>
## 1 ORD         17283 Chicago Ohare Intl    42.0  -87.9   668    -6 A     America/~
## 2 ATL         17215 Hartsfield Jackson~   33.6  -84.4  1026    -5 A     America/~
## 3 LAX         16174 Los Angeles Intl      33.9 -118.    126    -8 A     America/~
## 4 BOS         15508 General Edward Law~   42.4  -71.0    19    -5 A     America/~
## 5 MCO         14082 Orlando Intl          28.4  -81.3    96    -5 A     America/~
## 6 CLT         14064 Charlotte Douglas ~   35.2  -80.9   748    -5 A     America/~
```

Say instead we want to join two data frames by multiple key variables. For example, we see that in order to join the `flights` and `weather` data frames, we need more than one key variable: `year`, `month`, `day`, `hour`, and `origin`. This is because the combination of these 5 variables act to uniquely identify each observational unit in the weather data frame: hourly weather recordings at each of the 3 NYC airports

```
flights_weather_joined <-  flights %>%
  inner_join(weather, by = c("year", "month", "day", "hour", "origin"))

# explore it
glimpse(flights_weather_joined)
```

```
## Rows: 335,220
## Columns: 33
## $ year           <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013...
## $ month          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ day            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ dep_time       <int> 517, 533, 542, 544, 554, 554, 555, 557, 557, 558, 55...
## $ sched_dep_time <int> 515, 529, 540, 545, 600, 558, 600, 600, 600, 600, 60...
## $ dep_delay      <dbl> 2, 4, 2, -1, -6, -4, -5, -3, -3, -2, -2, -2, -2, -2,...
## $ arr_time       <int> 830, 850, 923, 1004, 812, 740, 913, 709, 838, 753, 8...
## $ sched_arr_time <int> 819, 830, 850, 1022, 837, 728, 854, 723, 846, 745, 8...
## $ arr_delay      <dbl> 11, 20, 33, -18, -25, 12, 19, -14, -8, 8, -2, -3, 7,...
## $ carrier        <chr> "UA", "UA", "AA", "B6", "DL", "UA", "B6", "EV", "B6"...
## $ flight         <int> 1545, 1714, 1141, 725, 461, 1696, 507, 5708, 79, 301...
## $ tailnum        <chr> "N14228", "N24211", "N619AA", "N804JB", "N668DN", "N...
## $ origin         <chr> "EWR", "LGA", "JFK", "JFK", "LGA", "EWR", "EWR", "LG...
## $ dest           <chr> "IAH", "IAH", "MIA", "BQN", "ATL", "ORD", "FLL", "IA...
## $ air_time       <dbl> 227, 227, 160, 183, 116, 150, 158, 53, 140, 138, 149...
## $ distance       <dbl> 1400, 1416, 1089, 1576, 762, 719, 1065, 229, 944, 73...
## $ hour           <dbl> 5, 5, 5, 5, 6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 5, 6, 6...
## $ minute         <dbl> 15, 29, 40, 45, 0, 58, 0, 0, 0, 0, 0, 0, 0, 0, 0, 59...
## $ time_hour.x    <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-0...
## $ gain           <dbl> -9, -16, -31, 17, 19, -16, -24, 11, 5, -10, 0, 1, -9...
## $ hours          <dbl> 3.7833333, 3.7833333, 2.6666667, 3.0500000, 1.933333...
## $ gain_per_hour  <dbl> -2.3788546, -4.2290749, -11.6250000, 5.5737705, 9.82...
## $ temp           <dbl> 39.02, 39.92, 39.02, 39.02, 39.92, 39.02, 37.94, 39....
## $ dewp           <dbl> 28.04, 24.98, 26.96, 26.96, 24.98, 28.04, 28.04, 24....
## $ humid          <dbl> 64.43, 54.81, 61.63, 61.63, 54.81, 64.43, 67.21, 54....
## $ wind_dir       <dbl> 260, 250, 260, 260, 260, 260, 240, 260, 260, 260, 26...
## $ wind_speed     <dbl> 12.65858, 14.96014, 14.96014, 14.96014, 16.11092, 12...
## $ wind_gust      <dbl> NA, 21.86482, NA, NA, 23.01560, NA, NA, 23.01560, NA...
## $ precip         <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0...
## $ pressure       <dbl> 1011.9, 1011.4, 1012.1, 1012.1, 1011.7, 1011.9, 1012...
## $ visib          <dbl> 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, ...
## $ time_hour.y    <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-0...
## $ temp_in_C      <dbl> 3.9, 4.4, 3.9, 3.9, 4.4, 3.9, 3.3, 4.4, 3.3, 4.4, 3....
```

```
head(flights_weather_joined)
```

```
## # A tibble: 6 x 33
##    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
## # ... with 25 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.x <dttm>, gain <dbl>, hours <dbl>,
## #   gain_per_hour <dbl>, temp <dbl>, dewp <dbl>, humid <dbl>, wind_dir <dbl>,
## #   wind_speed <dbl>, wind_gust <dbl>, precip <dbl>, pressure <dbl>,
## #   visib <dbl>, time_hour.y <dttm>, temp_in_C <dbl>
```

1.7 select variables/columns

flights data frame has 19 variables. Say you only need two of these 19 variables, say `carrier` and `flight`

```
just_two <- flights %>%
  select(carrier, flight)
```

```
head(just_two)
```

```
## # A tibble: 6 x 2
##   carrier flight
##   <chr>    <int>
## 1 UA        1545
## 2 UA        1714
## 3 AA        1141
## 4 B6         725
## 5 DL         461
## 6 UA        1696
```

Let's say instead you want to drop, or de-select, certain variables. For example, consider the variable `year` in the `flights` data frame. This variable isn't quite a "variable" because it is always 2013 and hence doesn't change. Remove this variable from the data frame by using the "-" sign

```
flights_no_year <- flights %>%
  select(-year)

head(flights_no_year)
```

```
## # A tibble: 6 x 21
##   month   day dep_time sched_dep_time dep_delay arr_time sched_arr_time
##   <int> <int>    <int>          <int>     <dbl>    <int>          <int>
## 1     1     1      517            515         2      830            819
## 2     1     1      533            529         4      850            830
## 3     1     1      542            540         2      923            850
## 4     1     1      544            545        -1     1004           1022
## 5     1     1      554            600        -6      812            837
## 6     1     1      554            558        -4      740            728
## # ... with 14 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>, gain <dbl>, hours <dbl>,
## #   gain_per_hour <dbl>
```

Another way of selecting columns/variables is by specifying a range of columns

```
flight_arr_times <- flights %>%
  select(month:day, arr_time:sched_arr_time)

head(flight_arr_times)
```

```
## # A tibble: 6 x 4
##   month   day arr_time sched_arr_time
##   <int> <int>    <int>          <int>
## 1     1     1      830            819
## 2     1     1      850            830
## 3     1     1      923            850
## 4     1     1     1004           1022
## 5     1     1      812            837
## 6     1     1      740            728
```

The `select()` function can also be used to reorder columns when used with the `everything()` helper function. Suppose we want the `hour`, `minute`, and `time_hour` variables to appear immediately after the `year`, `month`, and `day` variables, while not discarding the rest of the variables

```r
flights_reorder <- flights %>%
  select(year, month, day, hour, minute, time_hour, everything())

glimpse(flights_reorder)
```

```
## Rows: 336,776
## Columns: 22
## $ year           <int> 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013, 2013...
## $ month          <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ day            <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1...
## $ hour           <dbl> 5, 5, 5, 5, 6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 5, 6, 6...
## $ minute         <dbl> 15, 29, 40, 45, 0, 58, 0, 0, 0, 0, 0, 0, 0, 0, 0, 59...
## $ time_hour      <dttm> 2013-01-01 05:00:00, 2013-01-01 05:00:00, 2013-01-0...
## $ dep_time       <int> 517, 533, 542, 544, 554, 554, 555, 557, 557, 558, 55...
## $ sched_dep_time <int> 515, 529, 540, 545, 600, 558, 600, 600, 600, 600, 60...
## $ dep_delay      <dbl> 2, 4, 2, -1, -6, -4, -5, -3, -3, -2, -2, -2, -2, -2,...
## $ arr_time       <int> 830, 850, 923, 1004, 812, 740, 913, 709, 838, 753, 8...
## $ sched_arr_time <int> 819, 830, 850, 1022, 837, 728, 854, 723, 846, 745, 8...
## $ arr_delay      <dbl> 11, 20, 33, -18, -25, 12, 19, -14, -8, 8, -2, -3, 7,...
## $ carrier        <chr> "UA", "UA", "AA", "B6", "DL", "UA", "B6", "EV", "B6"...
## $ flight         <int> 1545, 1714, 1141, 725, 461, 1696, 507, 5708, 79, 301...
## $ tailnum        <chr> "N14228", "N24211", "N619AA", "N804JB", "N668DN", "N...
## $ origin         <chr> "EWR", "LGA", "JFK", "JFK", "LGA", "EWR", "EWR", "LG...
## $ dest           <chr> "IAH", "IAH", "MIA", "BQN", "ATL", "ORD", "FLL", "IA...
## $ air_time       <dbl> 227, 227, 160, 183, 116, 150, 158, 53, 140, 138, 149...
## $ distance       <dbl> 1400, 1416, 1089, 1576, 762, 719, 1065, 229, 944, 73...
## $ gain           <dbl> -9, -16, -31, 17, 19, -16, -24, 11, 5, -10, 0, 1, -9...
## $ hours          <dbl> 3.7833333, 3.7833333, 2.6666667, 3.0500000, 1.933333...
## $ gain_per_hour  <dbl> -2.3788546, -4.2290749, -11.6250000, 5.5737705, 9.82...
```

Lastly, the helper functions `starts_with()`, `ends_with()`, and `contains()` can be used to select variables/columns that match those conditions

```r
a_var <- flights %>%
  select(starts_with("a"))

head(a_var)
```

```
## # A tibble: 6 x 3
##   arr_time arr_delay air_time
##      <int>     <dbl>    <dbl>
## 1      830        11      227
## 2      850        20      227
## 3      923        33      160
## 4     1004       -18      183
## 5      812       -25      116
## 6      740        12      150
```

```r
delay_var <- flights %>%
 select(ends_with("delay"))

head(delay_var)
```

```
## # A tibble: 6 x 2
##   dep_delay arr_delay
##       <dbl>     <dbl>
## 1         2        11
## 2         4        20
## 3         2        33
## 4        -1       -18
## 5        -6       -25
## 6        -4        12
```

```r
time_var <- flights %>%
  select(contains("time"))

head(time_var)
```

```
## # A tibble: 6 x 6
##   dep_time sched_dep_time arr_time sched_arr_time air_time time_hour
##      <int>          <int>    <int>          <int>    <dbl> <dttm>
## 1      517            515      830            819      227 2013-01-01 05:00:00
## 2      533            529      850            830      227 2013-01-01 05:00:00
## 3      542            540      923            850      160 2013-01-01 05:00:00
## 4      544            545     1004           1022      183 2013-01-01 05:00:00
## 5      554            600      812            837      116 2013-01-01 06:00:00
## 6      554            558      740            728      150 2013-01-01 05:00:00
```

1.8 rename variables

rename changes the name of variables. In flights, change dep_time and arr_time to be departure_time and arrival_time

```r
flights_time_new <- flights %>%
  select(dep_time, arr_time) %>%
  rename(departure_time = dep_time, arrival_time = arr_time)

glimpse(flights_time_new)
```

```
## Rows: 336,776
## Columns: 2
## $ departure_time <int> 517, 533, 542, 544, 554, 554, 555, 557, 557, 558, 55...
## $ arrival_time   <int> 830, 850, 923, 1004, 812, 740, 913, 709, 838, 753, 8...
```

1.9 top_n values of a variable

return a data frame of the top 10 destination airports

```r
top_10 <- named_dests %>%
  top_n(n = 10, wt = num_flights)

top_10
```

```
## # A tibble: 10 x 9
##    dest  num_flights airport_name          lat    lon   alt    tz dst   tzone
##    <chr>       <int> <chr>               <dbl>  <dbl> <dbl> <dbl> <chr> <chr>
##  1 ORD         17283 Chicago Ohare Intl   42.0  -87.9   668    -6 A     America~
##  2 ATL         17215 Hartsfield Jackson~  33.6  -84.4  1026    -5 A     America~
##  3 LAX         16174 Los Angeles Intl     33.9 -118.    126    -8 A     America~
##  4 BOS         15508 General Edward Law~  42.4  -71.0    19    -5 A     America~
##  5 MCO         14082 Orlando Intl         28.4  -81.3    96    -5 A     America~
##  6 CLT         14064 Charlotte Douglas ~  35.2  -80.9   748    -5 A     America~
##  7 SFO         13331 San Francisco Intl   37.6 -122.     13    -8 A     America~
##  8 FLL         12055 Fort Lauderdale Ho~  26.1  -80.2     9    -5 A     America~
##  9 MIA         11728 Miami Intl           25.8  -80.3     8    -5 A     America~
## 10 DCA          9705 Ronald Reagan Wash~  38.9  -77.0    15    -5 A     America~
```