Ch3

Michael Rose

```
flights <- nycflights13::flights
print(flights)
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   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>
```

5 key dplyr functions:

filter() - pick observations by their values
arrange() - reorder the rows
select() - pick variables by their names
mutate() - create new variables with functions of existing variables
summarize() - collapse many values down to a single summary

all can be used in conjunction with group_by() which changes the scope of each function from operating

filter

```
(jan1 <- filter(flights, month == 1, day == 1))
```

```
## # A tibble: 842 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
```

```
## # ... with 832 more rows, and 12 more variables: sched_arr_time <int>,
## #   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>
```

```
(dec25 <- filter(flights, month == 12, day == 25))
```

```
## # A tibble: 719 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013    12    25      456            500        -4      649
##  2  2013    12    25      524            515         9      805
##  3  2013    12    25      542            540         2      832
##  4  2013    12    25      546            550        -4     1022
##  5  2013    12    25      556            600        -4      730
##  6  2013    12    25      557            600        -3      743
##  7  2013    12    25      557            600        -3      818
##  8  2013    12    25      559            600        -1      855
##  9  2013    12    25      559            600        -1      849
## 10  2013    12    25      600            600         0      850
## # ... with 709 more rows, and 12 more variables: sched_arr_time <int>,
## #   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>
```

Comparisons

```
near(sqrt(2)^2, 2)
```

```
## [1] TRUE
```

```
near(1/49 * 49, 1)
```

```
## [1] TRUE
```

```
(nov_or_dec <- filter(flights, month == 11 | month == 12))
```

```
## # A tibble: 55,403 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013    11     1        5           2359         6      352
##  2  2013    11     1       35           2250       105      123
##  3  2013    11     1      455            500        -5      641
##  4  2013    11     1      539            545        -6      856
##  5  2013    11     1      542            545        -3      831
##  6  2013    11     1      549            600       -11      912
##  7  2013    11     1      550            600       -10      705
##  8  2013    11     1      554            600        -6      659
##  9  2013    11     1      554            600        -6      826
## 10  2013    11     1      554            600        -6      749
## # ... with 55,393 more rows, and 12 more variables: sched_arr_time <int>,
## #   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>
```

```
# nov or dec 2
(nov_or_dec_2 <- filter(flights, month %in% c(11,12)))
```

```
## # A tibble: 55,403 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013    11     1        5           2359         6      352
##  2  2013    11     1       35           2250       105      123
##  3  2013    11     1      455            500        -5      641
##  4  2013    11     1      539            545        -6      856
##  5  2013    11     1      542            545        -3      831
##  6  2013    11     1      549            600       -11      912
##  7  2013    11     1      550            600       -10      705
##  8  2013    11     1      554            600        -6      659
##  9  2013    11     1      554            600        -6      826
## 10  2013    11     1      554            600        -6      749
## # ... with 55,393 more rows, and 12 more variables: sched_arr_time <int>,
## #   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>
```

```
# flights that weren't delayed by more than 2 hours

(low_delay <- filter(flights, !(arr_delay > 120 | dep_delay > 120)))
```

```
## # A tibble: 316,050 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 316,040 more rows, and 12 more variables: sched_arr_time <int>,
## #   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>
```

```
(low_delay2 <- filter(flights, !(arr_delay <= 120 | dep_delay <= 120)))
```

```
## # A tibble: 8,335 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      848           1835       853     1001
##  2  2013     1     1      957            733       144     1056
##  3  2013     1     1     1114            900       134     1447
##  4  2013     1     1     1815           1325       290     2120
##  5  2013     1     1     1842           1422       260     1958
##  6  2013     1     1     1856           1645       131     2212
##  7  2013     1     1     1934           1725       129     2126
##  8  2013     1     1     1938           1703       155     2109
##  9  2013     1     1     1942           1705       157     2124
## 10  2013     1     1     2006           1630       216     2230
```

```
## # ... with 8,325 more rows, and 12 more variables: sched_arr_time <int>,
## #   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>
```

Missing Values

```
# filter only includes ros where the condition is true, it excludes both FALSE and NA values. To preser
df <- tibble(x = c(1, NA, 3))
(df_no_na <- filter(df, x>1))
```

```
## # A tibble: 1 x 1
##       x
##   <dbl>
## 1     3
```

```
(df_na <- filter(df, is.na(x) | x > 1))
```

```
## # A tibble: 2 x 1
##       x
##   <dbl>
## 1    NA
## 2     3
```

```
# check carriers
print(unique(flights$carrier))
```

```
##  [1] "UA" "AA" "B6" "DL" "EV" "MQ" "US" "WN" "VX" "FL" "AS" "9E" "F9" "HA"
## [15] "YV" "OO"
```

```
# more than 2 hours late
(morethan2hours <- filter(flights, !between(arr_delay, 0, 120)))
```

```
## # A tibble: 198,967 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      544            545        -1     1004
##  2  2013     1     1      554            600        -6      812
##  3  2013     1     1      557            600        -3      709
##  4  2013     1     1      557            600        -3      838
##  5  2013     1     1      558            600        -2      849
##  6  2013     1     1      558            600        -2      853
##  7  2013     1     1      558            600        -2      923
##  8  2013     1     1      559            559         0      702
##  9  2013     1     1      559            600        -1      854
## 10  2013     1     1      600            600         0      851
## # ... with 198,957 more rows, and 12 more variables: sched_arr_time <int>,
## #   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>
```

```
 # flew to Houston or Iowa
(flewToHouston <- filter(flights, dest == "IAH" | dest == "HOU"))
```

```
## # A tibble: 9,313 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
```

```
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      623            627        -4      933
##  4  2013     1     1      728            732        -4     1041
##  5  2013     1     1      739            739         0     1104
##  6  2013     1     1      908            908         0     1228
##  7  2013     1     1     1028           1026         2     1350
##  8  2013     1     1     1044           1045        -1     1352
##  9  2013     1     1     1114            900       134     1447
## 10  2013     1     1     1205           1200         5     1503
## # ... with 9,303 more rows, and 12 more variables: sched_arr_time <int>,
## #   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>
```

```
# Were on carriers United Airlines, American Airlines or Delta Airlines
(unitedAmericanDelta <- filter(flights, carrier %in% c("UA", "AA", "DL")))
```

```
## # A tibble: 139,504 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      554            600        -6      812
##  5  2013     1     1      554            558        -4      740
##  6  2013     1     1      558            600        -2      753
##  7  2013     1     1      558            600        -2      924
##  8  2013     1     1      558            600        -2      923
##  9  2013     1     1      559            600        -1      941
## 10  2013     1     1      559            600        -1      854
## # ... with 139,494 more rows, and 12 more variables: sched_arr_time <int>,
## #   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>
```

```
# Departed in Summer
(departInSummer <- filter(flights, month %in% c(7, 8, 9)))
```

```
## # A tibble: 86,326 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     7     1        1           2029       212      236
##  2  2013     7     1        2           2359         3      344
##  3  2013     7     1       29           2245       104      151
##  4  2013     7     1       43           2130       193      322
##  5  2013     7     1       44           2150       174      300
##  6  2013     7     1       46           2051       235      304
##  7  2013     7     1       48           2001       287      308
##  8  2013     7     1       58           2155       183      335
##  9  2013     7     1      100           2146       194      327
## 10  2013     7     1      100           2245       135      337
## # ... with 86,316 more rows, and 12 more variables: sched_arr_time <int>,
## #   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>
```

```
# Arrived greater than 2 hours late but still departed on time
(arrLateDepartOnTime <- filter(flights, !is.na(dep_delay), arr_delay > 120, dep_delay <= 0))
```

```
## # A tibble: 29 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1    27     1419           1420        -1     1754
##  2  2013    10     7     1350           1350         0     1736
##  3  2013    10     7     1357           1359        -2     1858
##  4  2013    10    16      657            700        -3     1258
##  5  2013    11     1      658            700        -2     1329
##  6  2013     3    18     1844           1847        -3       39
##  7  2013     4    17     1635           1640        -5     2049
##  8  2013     4    18      558            600        -2     1149
##  9  2013     4    18      655            700        -5     1213
## 10  2013     5    22     1827           1830        -3     2217
## # ... with 19 more rows, and 12 more variables: sched_arr_time <int>,
## #   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>
```

```
# delayed by at least an hour, but made up over 30 minutes in flight
(delayedMadeUpInFlight <- filter(flights, !is.na(dep_delay), dep_delay >= 60, dep_delay - arr_delay > 3
```

```
## # A tibble: 1,844 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1     2205           1720       285       46
##  2  2013     1     1     2326           2130       116      131
##  3  2013     1     3     1503           1221       162     1803
##  4  2013     1     3     1839           1700        99     2056
##  5  2013     1     3     1850           1745        65     2148
##  6  2013     1     3     1941           1759       102     2246
##  7  2013     1     3     1950           1845        65     2228
##  8  2013     1     3     2015           1915        60     2135
##  9  2013     1     3     2257           2000       177       45
## 10  2013     1     4     1917           1700       137     2135
## # ... with 1,834 more rows, and 12 more variables: sched_arr_time <int>,
## #   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>
```

```
# departed between midnight and 6 am
(midAndSix <- filter(flights, between(dep_time, 0, 600)))
```

```
## # A tibble: 9,344 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
```

```
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 9,334 more rows, and 12 more variables: sched_arr_time <int>,
## #   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>
```

```
# print missing dep_times
print(sum(is.na(flights$dep_time)))
```

```
## [1] 8255
```

Arrange

```
# arrange works similarly to filter, except that instead of selecting rows, it changes their order
# if more than one column is passed to the function, it gives orderwise precendence to the columns
arrange(flights, year, month, day)
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   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>
```

```
# descending
arrange(flights, desc(arr_delay))
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     9      641            900      1301     1242
##  2  2013     6    15     1432           1935      1137     1607
##  3  2013     1    10     1121           1635      1126     1239
##  4  2013     9    20     1139           1845      1014     1457
##  5  2013     7    22      845           1600      1005     1044
##  6  2013     4    10     1100           1900       960     1342
##  7  2013     3    17     2321            810       911      135
##  8  2013     7    22     2257            759       898      121
##  9  2013    12     5      756           1700       896     1058
## 10  2013     5     3     1133           2055       878     1250
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
```

```
## #   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>
```

```
# missing values are always sorted at the end
df <- tibble(x = c(5, 2, NA))
arrange(df, x)
```

```
## # A tibble: 3 x 1
##       x
##   <dbl>
## 1     2
## 2     5
## 3    NA
```

```
arrange(df, desc(x))
```

```
## # A tibble: 3 x 1
##       x
##   <dbl>
## 1     5
## 2     2
## 3    NA
```

```
arrange(df, desc(is.na(x)), x)
```

```
## # A tibble: 3 x 1
##       x
##   <dbl>
## 1    NA
## 2     2
## 3     5
```

```
# most delayed flights
arrange(flights, desc(dep_delay))
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     9      641            900      1301     1242
##  2  2013     6    15     1432           1935      1137     1607
##  3  2013     1    10     1121           1635      1126     1239
##  4  2013     9    20     1139           1845      1014     1457
##  5  2013     7    22      845           1600      1005     1044
##  6  2013     4    10     1100           1900       960     1342
##  7  2013     3    17     2321            810       911      135
##  8  2013     6    27      959           1900       899     1236
##  9  2013     7    22     2257            759       898      121
## 10  2013    12     5      756           1700       896     1058
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   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>
```

```
# earliest leaving flights
arrange(flights, dep_delay)
```

```
## # A tibble: 336,776 x 19
```

```
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013    12     7     2040           2123       -43       40
##  2  2013     2     3     2022           2055       -33     2240
##  3  2013    11    10     1408           1440       -32     1549
##  4  2013     1    11     1900           1930       -30     2233
##  5  2013     1    29     1703           1730       -27     1947
##  6  2013     8     9      729            755       -26     1002
##  7  2013    10    23     1907           1932       -25     2143
##  8  2013     3    30     2030           2055       -25     2213
##  9  2013     3     2     1431           1455       -24     1601
## 10  2013     5     5      934            958       -24     1225
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   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>
```

```
# fastest flight times
arrange(flights, air_time)
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1    16     1355           1315        40     1442
##  2  2013     4    13      537            527        10      622
##  3  2013    12     6      922            851        31     1021
##  4  2013     2     3     2153           2129        24     2247
##  5  2013     2     5     1303           1315       -12     1342
##  6  2013     2    12     2123           2130        -7     2211
##  7  2013     3     2     1450           1500       -10     1547
##  8  2013     3     8     2026           1935        51     2131
##  9  2013     3    18     1456           1329        87     1533
## 10  2013     3    19     2226           2145        41     2305
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   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>
```

```
# longest flight distance
arrange(flights, desc(distance))
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      857            900        -3     1516
##  2  2013     1     2      909            900         9     1525
##  3  2013     1     3      914            900        14     1504
##  4  2013     1     4      900            900         0     1516
##  5  2013     1     5      858            900        -2     1519
##  6  2013     1     6     1019            900        79     1558
##  7  2013     1     7     1042            900       102     1620
##  8  2013     1     8      901            900         1     1504
##  9  2013     1     9      641            900      1301     1242
## 10  2013     1    10      859            900        -1     1449
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   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>
```

```
# shortest flight distance
arrange(flights, distance)
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     7    27       NA            106        NA       NA
##  2  2013     1     3     2127           2129        -2     2222
##  3  2013     1     4     1240           1200        40     1333
##  4  2013     1     4     1829           1615       134     1937
##  5  2013     1     4     2128           2129        -1     2218
##  6  2013     1     5     1155           1200        -5     1241
##  7  2013     1     6     2125           2129        -4     2224
##  8  2013     1     7     2124           2129        -5     2212
##  9  2013     1     8     2127           2130        -3     2304
## 10  2013     1     9     2126           2129        -3     2217
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   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>
```

Select

```
# select columns by name
byname <- select(flights, year, month, day)

# select all columns between year and day
bybetween <- select(flights, year:day)

# select all columns except those from year to day
bynotbetween <- select(flights, -(year:day))

names(byname)
```

```
## [1] "year"  "month" "day"
```

```
names(bybetween)
```

```
## [1] "year"  "month" "day"
```

```
names(bynotbetween)
```

```
##  [1] "dep_time"       "sched_dep_time" "dep_delay"      "arr_time"
##  [5] "sched_arr_time" "arr_delay"      "carrier"        "flight"
##  [9] "tailnum"        "origin"         "dest"           "air_time"
## [13] "distance"       "hour"           "minute"         "time_hour"
```

```
# helper functions:
# starts_with("abc") matches names that begin with abc
# ends_with("xyz") matches names that end with xyz
# contains("ijk") matches names that contain "ijk"
# matches("(.)\\1") selects variables that match a regular expression. This one matches vars that conta
# num_range("x", 1:3) matches x1,x2,x3
```

```
# renames all vars mentioned, keeps those not mentioned. select would drop all vars not mentioned
rename(flights, tail_num = tailnum)
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tail_num <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

```
names(flights)
```

```
##  [1] "year"           "month"          "day"            "dep_time"
##  [5] "sched_dep_time" "dep_delay"      "arr_time"       "sched_arr_time"
##  [9] "arr_delay"      "carrier"        "flight"         "tailnum"
## [13] "origin"         "dest"           "air_time"       "distance"
## [17] "hour"           "minute"         "time_hour"
```

```
# select in conjunction with everything() helped moves a handful of variables to the start of the dataf
select(flights, time_hour, air_time, everything())
```

```
## # A tibble: 336,776 x 19
##    time_hour           air_time  year month   day dep_time sched_dep_time
##    <dttm>                 <dbl> <int> <int> <int>    <int>          <int>
##  1 2013-01-01 05:00:00      227  2013     1     1      517            515
##  2 2013-01-01 05:00:00      227  2013     1     1      533            529
##  3 2013-01-01 05:00:00      160  2013     1     1      542            540
##  4 2013-01-01 05:00:00      183  2013     1     1      544            545
##  5 2013-01-01 06:00:00      116  2013     1     1      554            600
##  6 2013-01-01 05:00:00      150  2013     1     1      554            558
##  7 2013-01-01 06:00:00      158  2013     1     1      555            600
##  8 2013-01-01 06:00:00       53  2013     1     1      557            600
##  9 2013-01-01 06:00:00      140  2013     1     1      557            600
## 10 2013-01-01 06:00:00      138  2013     1     1      558            600
## # ... with 336,766 more rows, and 12 more variables: dep_delay <dbl>,
## #   arr_time <int>, sched_arr_time <int>, arr_delay <dbl>, carrier <chr>,
## #   flight <int>, tailnum <chr>, origin <chr>, dest <chr>, distance <dbl>,
## #   hour <dbl>, minute <dbl>
```

```
# some ways to select dep_time, dep_delay, arr_time, arr_delay
select(flights, dep_time, dep_delay, arr_time, arr_delay)
```

```
## # A tibble: 336,776 x 4
##    dep_time dep_delay arr_time arr_delay
```

```
##       <int>     <dbl>    <int>     <dbl>
##  1      517         2      830        11
##  2      533         4      850        20
##  3      542         2      923        33
##  4      544        -1     1004       -18
##  5      554        -6      812       -25
##  6      554        -4      740        12
##  7      555        -5      913        19
##  8      557        -3      709       -14
##  9      557        -3      838        -8
## 10      558        -2      753         8
## # ... with 336,766 more rows
```

```
select(flights, starts_with("dep"), starts_with("arr"))
```

```
## # A tibble: 336,776 x 4
##    dep_time dep_delay arr_time arr_delay
##       <int>     <dbl>    <int>     <dbl>
##  1      517         2      830        11
##  2      533         4      850        20
##  3      542         2      923        33
##  4      544        -1     1004       -18
##  5      554        -6      812       -25
##  6      554        -4      740        12
##  7      555        -5      913        19
##  8      557        -3      709       -14
##  9      557        -3      838        -8
## 10      558        -2      753         8
## # ... with 336,766 more rows
```

```
select(flights, matches("^(dep|arr)_(time|delay)$"))
```

```
## # A tibble: 336,776 x 4
##    dep_time dep_delay arr_time arr_delay
##       <int>     <dbl>    <int>     <dbl>
##  1      517         2      830        11
##  2      533         4      850        20
##  3      542         2      923        33
##  4      544        -1     1004       -18
##  5      554        -6      812       -25
##  6      554        -4      740        12
##  7      555        -5      913        19
##  8      557        -3      709       -14
##  9      557        -3      838        -8
## 10      558        -2      753         8
## # ... with 336,766 more rows
```

```
# send duplicate parameters into select
select(flights, dep_time, dep_time, dep_time)
```

```
## # A tibble: 336,776 x 1
##    dep_time
##       <int>
##  1      517
##  2      533
##  3      542
```

```
##  4      544
##  5      554
##  6      554
##  7      555
##  8      557
##  9      557
## 10      558
## # ... with 336,766 more rows
```

```
# one of
vars <- c("year", "month", "day", "dep_delay", "arr_delay")

select(flights, one_of(vars))
```

```
## # A tibble: 336,776 x 5
##     year month   day dep_delay arr_delay
##    <int> <int> <int>     <dbl>     <dbl>
##  1  2013     1     1         2        11
##  2  2013     1     1         4        20
##  3  2013     1     1         2        33
##  4  2013     1     1        -1       -18
##  5  2013     1     1        -6       -25
##  6  2013     1     1        -4        12
##  7  2013     1     1        -5        19
##  8  2013     1     1        -3       -14
##  9  2013     1     1        -3        -8
## 10  2013     1     1        -2         8
## # ... with 336,766 more rows
```

```
select(flights, contains("TIME"))
```

```
## # A tibble: 336,776 x 6
##    dep_time sched_dep_time arr_time sched_arr_time air_time
##       <int>          <int>    <int>          <int>    <dbl>
##  1      517            515      830            819      227
##  2      533            529      850            830      227
##  3      542            540      923            850      160
##  4      544            545     1004           1022      183
##  5      554            600      812            837      116
##  6      554            558      740            728      150
##  7      555            600      913            854      158
##  8      557            600      709            723       53
##  9      557            600      838            846      140
## 10      558            600      753            745      138
## # ... with 336,766 more rows, and 1 more variable: time_hour <dttm>
```

Mutate

```
flights_sml <- select(flights, year:day, ends_with("delay"), distance, air_time)

# create new variables with mutate!
mutate(flights_sml, gain = arr_delay - dep_delay, speed = distance / air_time * 60)
```

```
## # A tibble: 336,776 x 9
##     year month   day dep_delay arr_delay distance air_time  gain speed
```

```
##    <int> <int> <int>     <dbl>     <dbl>    <dbl>    <dbl> <dbl> <dbl>
##  1  2013     1     1         2        11     1400      227     9  370.
##  2  2013     1     1         4        20     1416      227    16  374.
##  3  2013     1     1         2        33     1089      160    31  408.
##  4  2013     1     1        -1       -18     1576      183   -17  517.
##  5  2013     1     1        -6       -25      762      116   -19  394.
##  6  2013     1     1        -4        12      719      150    16  288.
##  7  2013     1     1        -5        19     1065      158    24  404.
##  8  2013     1     1        -3       -14      229       53   -11  259.
##  9  2013     1     1        -3        -8      944      140    -5  405.
## 10  2013     1     1        -2         8      733      138    10  319.
## # ... with 336,766 more rows
```

```
mutate(flights_sml, gain = arr_delay - dep_delay, hours = air_time / 60, gains_per_hour = gain / hours)
```

```
## # A tibble: 336,776 x 10
##     year month   day dep_delay arr_delay distance air_time  gain hours
##    <int> <int> <int>     <dbl>     <dbl>    <dbl>    <dbl> <dbl> <dbl>
##  1  2013     1     1         2        11     1400      227     9 3.78
##  2  2013     1     1         4        20     1416      227    16 3.78
##  3  2013     1     1         2        33     1089      160    31 2.67
##  4  2013     1     1        -1       -18     1576      183   -17 3.05
##  5  2013     1     1        -6       -25      762      116   -19 1.93
##  6  2013     1     1        -4        12      719      150    16 2.5
##  7  2013     1     1        -5        19     1065      158    24 2.63
##  8  2013     1     1        -3       -14      229       53   -11 0.883
##  9  2013     1     1        -3        -8      944      140    -5 2.33
## 10  2013     1     1        -2         8      733      138    10 2.3
## # ... with 336,766 more rows, and 1 more variable: gains_per_hour <dbl>
```

```
# if you only want to keep new variables, use transmute()
transmute(flights, gain = arr_delay - dep_delay, hours = air_time / 60, gains_per_hour = gain / hours)
```

```
## # A tibble: 336,776 x 3
##     gain hours gains_per_hour
##    <dbl> <dbl>          <dbl>
##  1     9 3.78            2.38
##  2    16 3.78            4.23
##  3    31 2.67           11.6
##  4   -17 3.05           -5.57
##  5   -19 1.93           -9.83
##  6    16 2.5             6.4
##  7    24 2.63            9.11
##  8   -11 0.883         -12.5
##  9    -5 2.33           -2.14
## 10    10 2.3             4.35
## # ... with 336,766 more rows
```

```
# any function that is vectorized - i.e. takes a vector and returns a vector can be used in conjunction

# for arithmetic operators, if one is parameter is shorter than another, it will automatically be exten

# modular arithmetic
transmute(flights, dep_time, hour = dep_time%/% 100, minute = dep_time%% 100)
```

```
## # A tibble: 336,776 x 3
```

```
##    dep_time  hour minute
##       <int> <dbl>  <dbl>
##  1      517     5     17
##  2      533     5     33
##  3      542     5     42
##  4      544     5     44
##  5      554     5     54
##  6      554     5     54
##  7      555     5     55
##  8      557     5     57
##  9      557     5     57
## 10      558     5     58
## # ... with 336,766 more rows
```

```
# offsets
(x <- 1:10)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
lag(x)
```

```
##  [1] NA  1  2  3  4  5  6  7  8  9
```

```
lead(x)
```

```
##  [1]  2  3  4  5  6  7  8  9 10 NA
```

```
# cumulative and rolling aggregates
x
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
cumsum(x)
```

```
##  [1]  1  3  6 10 15 21 28 36 45 55
```

```
cummean(x)
```

```
##  [1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5
```

```
cumprod(x)
```

```
##  [1]       1       2       6      24     120     720    5040   40320
##  [9]  362880 3628800
```

```
# ranking functions
y <- c(1, 2, 2, NA, 3, 4)
min_rank(y)
```

```
## [1]  1  2  2 NA  4  5
```

```
min_rank(desc(y))
```

```
## [1]  5  3  3 NA  2  1
```

```
row_number(y)
```

```
## [1]  1  2  3 NA  4  5
```

```
dense_rank(y)
```

```
## [1]  1  2  2 NA  3  4
```

```
percent_rank(y)
```

```
## [1] 0.00 0.25 0.25   NA 0.75 1.00
```

```
cume_dist(y)
```

```
## [1] 0.2 0.6 0.6  NA 0.8 1.0
```

```
# mutate dep_time and sched_dep_time
mutate(flights, dep_time_min = dep_time%/%100 * 60 + dep_time%% 100, sched_dep_time_min = sched_dep_tim
  select(dep_time, dep_time_min, sched_dep_time, sched_dep_time_min)
```

```
## # A tibble: 336,776 x 4
##    dep_time dep_time_min sched_dep_time sched_dep_time_min
##       <int>        <dbl>          <int>              <dbl>
##  1      517          317            515                315
##  2      533          333            529                329
##  3      542          342            540                340
##  4      544          344            545                345
##  5      554          354            600                360
##  6      554          354            558                358
##  7      555          355            600                360
##  8      557          357            600                360
##  9      557          357            600                360
## 10      558          358            600                360
## # ... with 336,766 more rows
```

```
# Done using a function
time2mins <- function(x) {
  x %/% 100 * 60 + x %% 100
}

mutate(flights, dep_time_min = time2mins(dep_time), sched_dep_time_min = time2mins(sched_dep_time)) %>%
  select(dep_time, dep_time_min, sched_dep_time, sched_dep_time_min)
```

```
## # A tibble: 336,776 x 4
##    dep_time dep_time_min sched_dep_time sched_dep_time_min
##       <int>        <dbl>          <int>              <dbl>
##  1      517          317            515                315
##  2      533          333            529                329
##  3      542          342            540                340
##  4      544          344            545                345
##  5      554          354            600                360
##  6      554          354            558                358
##  7      555          355            600                360
##  8      557          357            600                360
##  9      557          357            600                360
## 10      558          358            600                360
## # ... with 336,766 more rows
```

```
mutate(flights, arrdep = arr_time - dep_time) %>%
select(air_time, arrdep)
```

```
## # A tibble: 336,776 x 2
##    air_time arrdep
##       <dbl>  <int>
##  1      227    313
```

```
##  2      227    317
##  3      160    381
##  4      183    460
##  5      116    258
##  6      150    186
##  7      158    358
##  8       53    152
##  9      140    281
## 10      138    195
## # ... with 336,766 more rows
```

```
# find 10 most delayed flights using a ranking function
head(min_rank(desc(flights$dep_delay)))
```

```
## [1] 114150 103893 114150 144947 258934 209494
```

```
1:3 + 1:10
```

```
## Warning in 1:3 + 1:10: longer object length is not a multiple of shorter
## object length
```

```
##  [1]  2  4  6  5  7  9  8 10 12 11
```

Grouped Summaries with summarize()

```
# not terribly useful without group_by()
summarize(flights, delay = mean(dep_delay, na.rm = TRUE))
```

```
## # A tibble: 1 x 1
##   delay
##   <dbl>
## 1  12.6
```

```
# with group_by
by_day <- group_by(flights, year, month, day)
summarize(by_day, delay = mean(dep_delay, na.rm = TRUE))
```

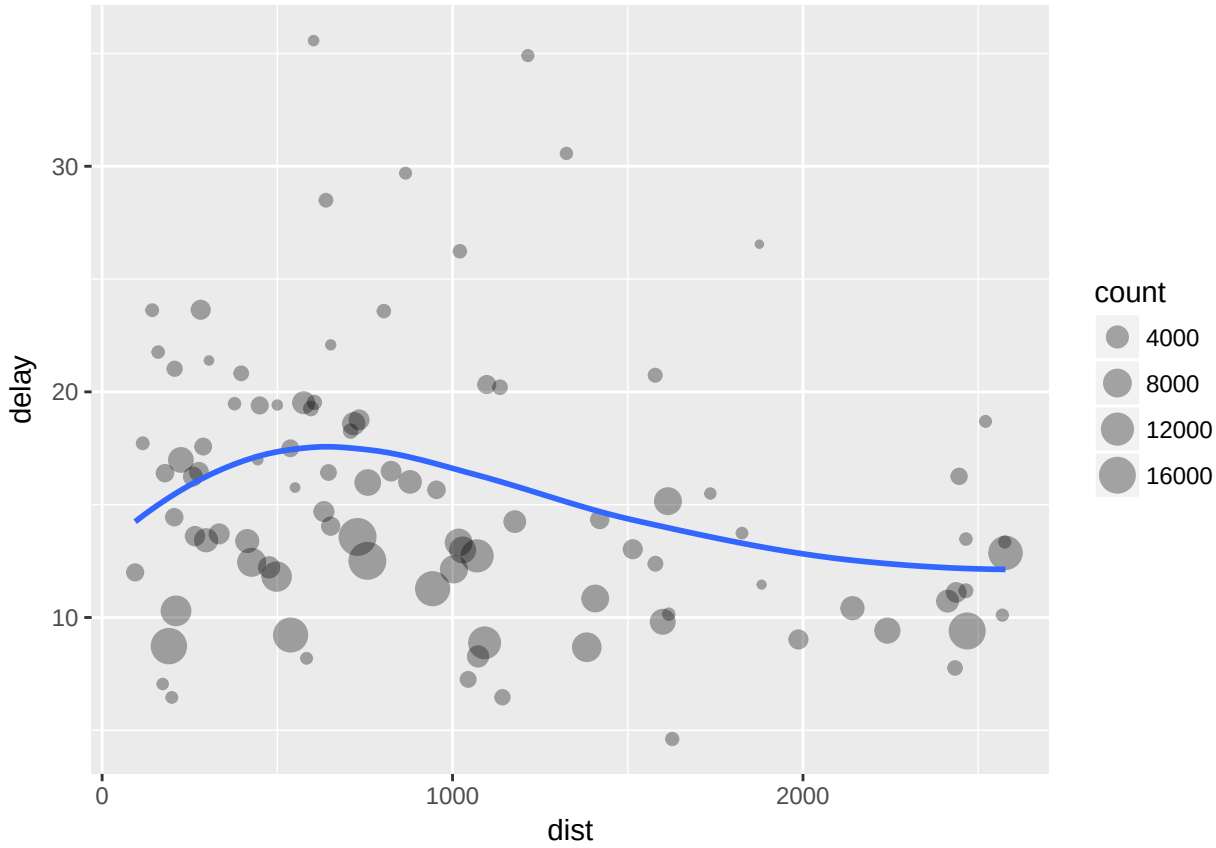
```
## # A tibble: 365 x 4
## # Groups:   year, month [?]
##     year month   day delay
##    <int> <int> <int> <dbl>
##  1  2013     1     1 11.5
##  2  2013     1     2 13.9
##  3  2013     1     3 11.0
##  4  2013     1     4  8.95
##  5  2013     1     5  5.73
##  6  2013     1     6  7.15
##  7  2013     1     7  5.42
##  8  2013     1     8  2.55
##  9  2013     1     9  2.28
## 10  2013     1    10  2.84
## # ... with 355 more rows
```

```
# this provides a group summary. In the case above, a group summary for the mean departure delay time f
```

The Pipe

```
# we want to explore the relationship between distance and average delay for each location.

# The hard way:

# group flights by destination
# summarize to compute distance, average delay, and number of flights
# filter to remove noisy points and honolulu airport which is almost twice as far away as the next clos
# plot and use local regression

by_dest <- group_by(flights, dest)
delay <- summarize(by_dest,
                   count = n(),
                   dist = mean(distance, na.rm = TRUE),
                   delay = mean(dep_delay, na.rm = TRUE))
delay <- filter(delay, count > 20, dest != "HNL")

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

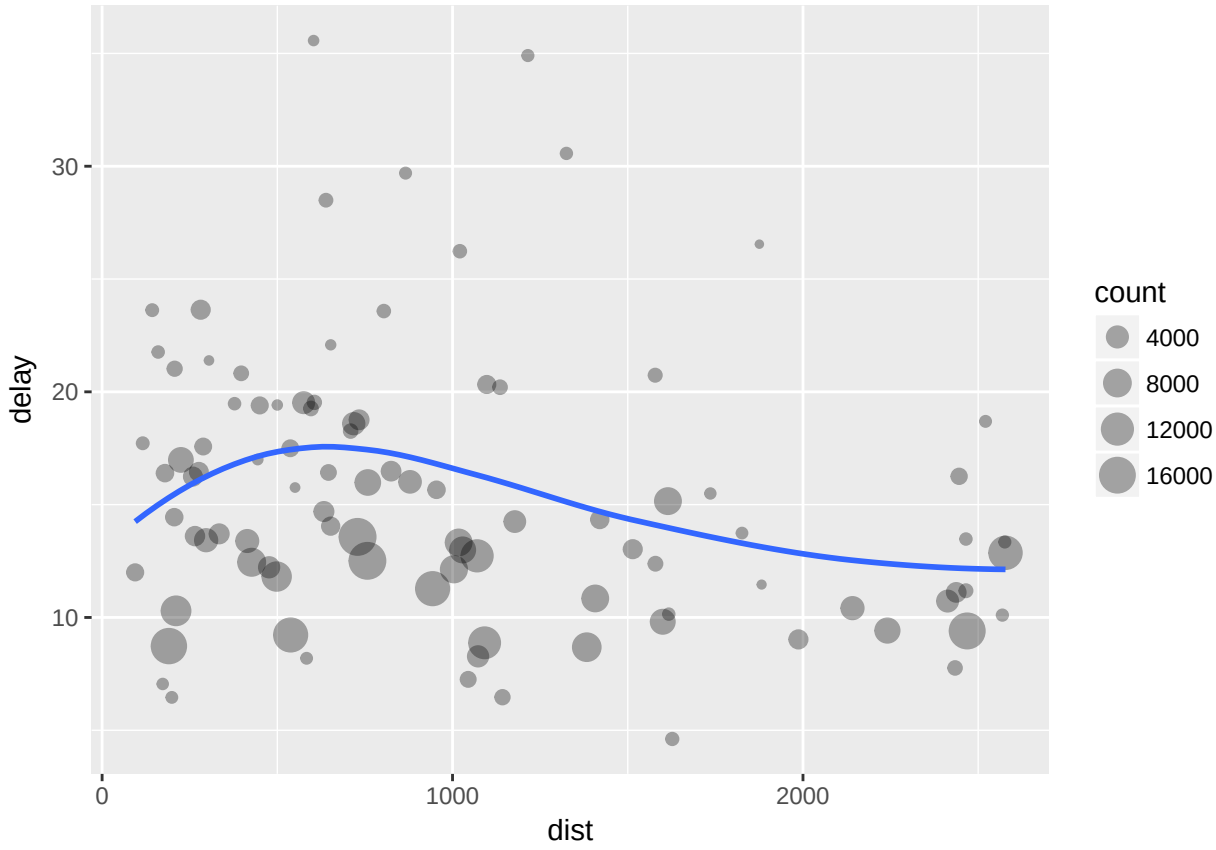
```
## `geom_smooth()` using method = 'loess'
```




```
# The pipe way:

delays <- flights %>%
  group_by(dest) %>%
    summarize(count = n(),
```

```
              dist = mean(distance, na.rm = TRUE),
              delay = mean(dep_delay, na.rm = TRUE)) %>%
      filter(count > 20, dest != "HNL")

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

```
## `geom_smooth()` using method = 'loess'
```



Missing Values

```
# not including na.rm = TRUE leads to lots of NA values for the mean. If there are any missing values i
flights %>%
  group_by(year, month, day) %>%
    summarize(mean = mean(dep_delay))
```

```
## # A tibble: 365 x 4
## # Groups:   year, month [?]
##     year month   day  mean
##    <int> <int> <int> <dbl>
##  1  2013     1     1    NA
##  2  2013     1     2    NA
##  3  2013     1     3    NA
##  4  2013     1     4    NA
##  5  2013     1     5    NA
```

```
##  6  2013     1     6    NA
##  7  2013     1     7    NA
##  8  2013     1     8    NA
##  9  2013     1     9    NA
## 10  2013     1    10    NA
## # ... with 355 more rows
```

```
# including na.rm
flights %>%
  group_by(year, month, day) %>%
    summarize(mean = mean(dep_delay, na.rm = TRUE))
```

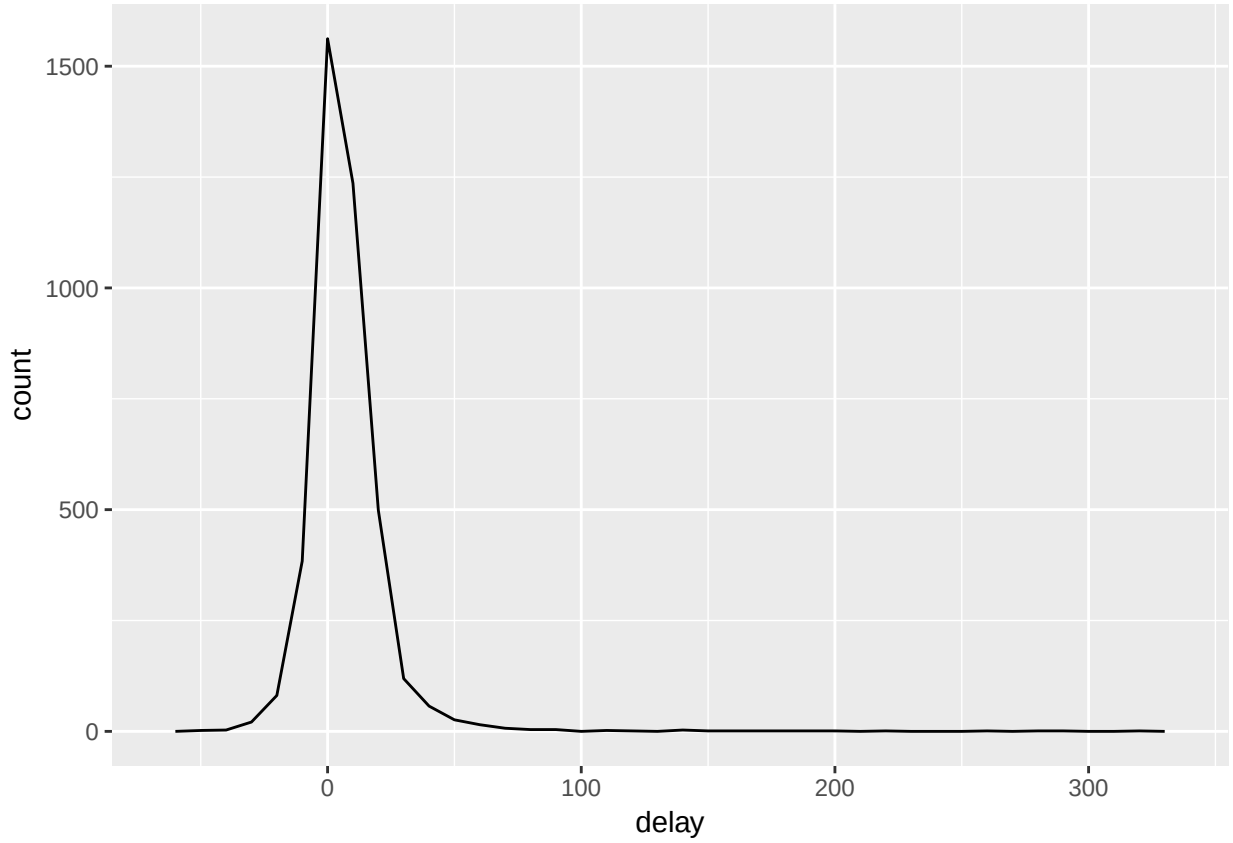
```
## # A tibble: 365 x 4
## # Groups:   year, month [?]
##     year month   day  mean
##    <int> <int> <int> <dbl>
##  1  2013     1     1 11.5
##  2  2013     1     2 13.9
##  3  2013     1     3 11.0
##  4  2013     1     4  8.95
##  5  2013     1     5  5.73
##  6  2013     1     6  7.15
##  7  2013     1     7  5.42
##  8  2013     1     8  2.55
##  9  2013     1     9  2.28
## 10  2013     1    10  2.84
## # ... with 355 more rows
```

```
not_cancelled <- flights %>%
  filter(!is.na(dep_delay), !is.na(arr_delay))

not_cancelled %>%
  group_by(year, month, day) %>%
  summarize(mean = mean(dep_delay))
```

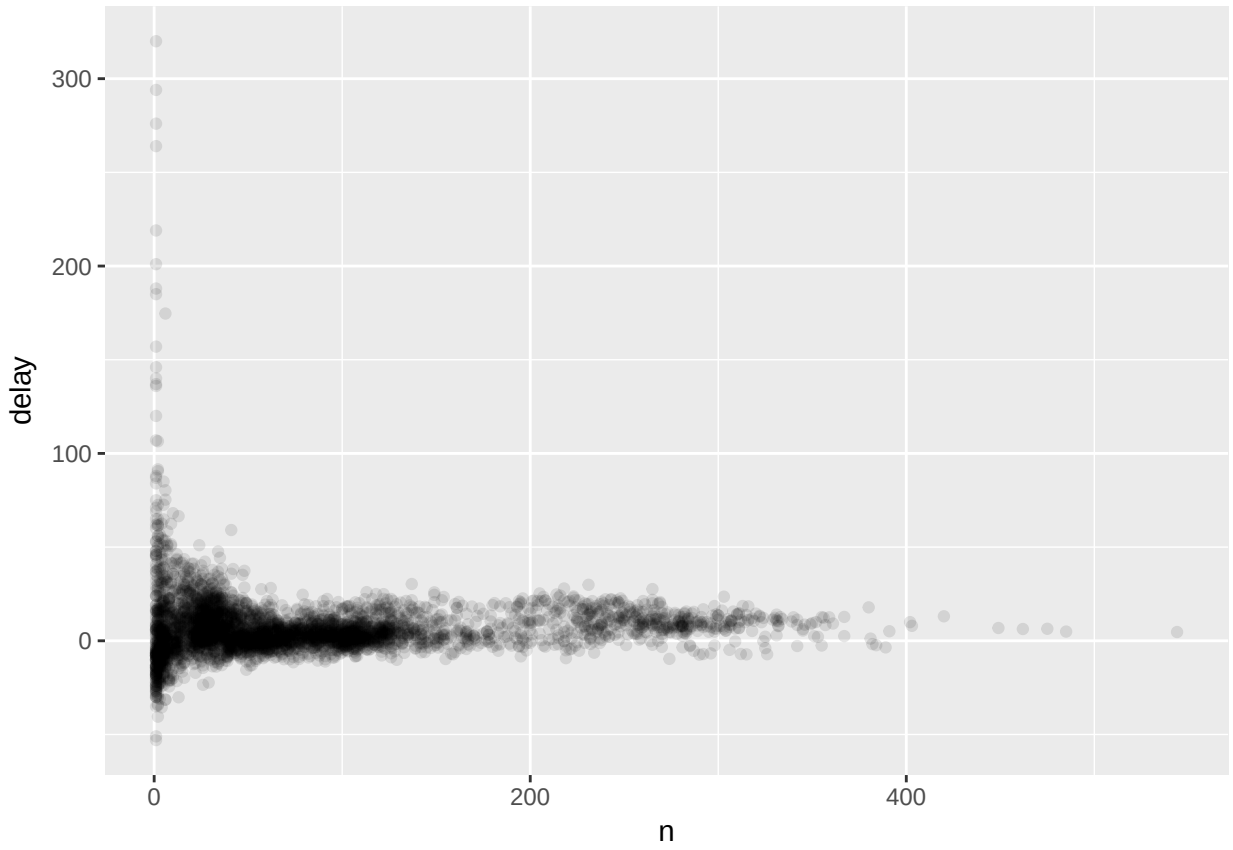
```
## # A tibble: 365 x 4
## # Groups:   year, month [?]
##     year month   day  mean
##    <int> <int> <int> <dbl>
##  1  2013     1     1 11.4
##  2  2013     1     2 13.7
##  3  2013     1     3 10.9
##  4  2013     1     4  8.97
##  5  2013     1     5  5.73
##  6  2013     1     6  7.15
##  7  2013     1     7  5.42
##  8  2013     1     8  2.56
##  9  2013     1     9  2.30
## 10  2013     1    10  2.84
## # ... with 355 more rows
```

Counts

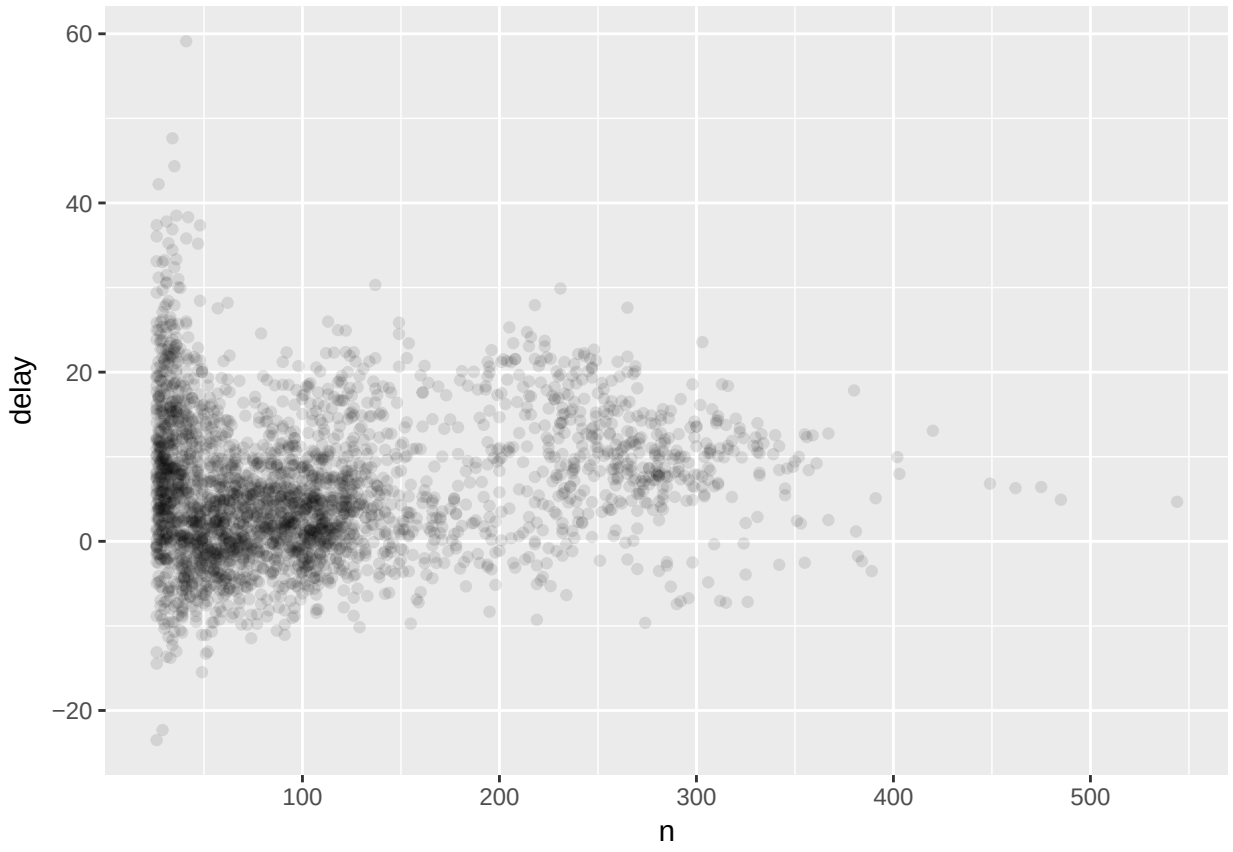
```r
#whenever doing an aggregation, check count or number of na to make sure we don't draw conclusions on a
planes <- nycflights13::planes
weather <- nycflights13::weather
airports <- nycflights13::airports


delays <- not_cancelled %>%
  group_by(tailnum) %>%
    summarize(
      delay = mean(arr_delay)
    )

ggplot(data = delays, mapping = aes(x = delay)) +
  geom_freqpoly(binwidth = 10)
```




```r
delays <- not_cancelled %>%
  group_by(tailnum) %>%
    summarize(
      delay = mean(arr_delay, na.rm = TRUE),
      n = n()
    )

ggplot(data = delays, mapping = aes(x = n, y = delay)) +
  geom_point(alpha = 1/10)
```



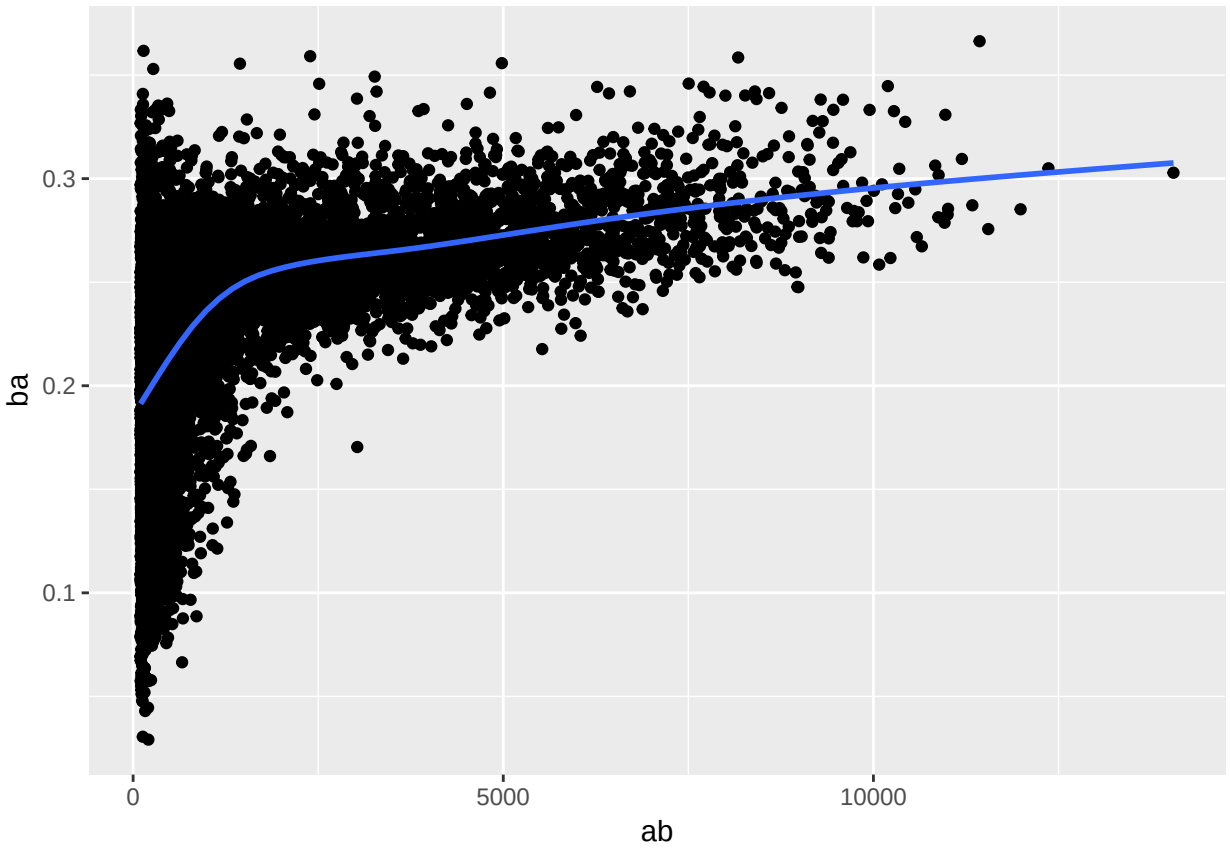

```
delays %>%
  filter(n > 25) %>%
    ggplot(mapping = aes(x = n, y = delay)) +
      geom_point(alpha = 1/10)
```



Counts with baseball

```
batting <- as_tibble(Lahman::Batting)

batters <- batting %>%
  group_by(playerID) %>%
    summarize(
      ba = sum(H, na.rm = TRUE) / sum(AB, na.rm = TRUE),
      ab = sum(AB, na.rm = TRUE)
    )

batters %>%
  filter(ab > 100) %>%
  ggplot(mapping = aes(x = ab, y = ba)) +
    geom_point() +
    geom_smooth(se = FALSE)
```

```
## `geom_smooth()` using method = 'gam'
```



Useful Summary Functions

```
median(x)
# measures of spread
sd(x) - standard deviation
IQR(x) - Interquartile Range
mad(x) - median absolute deviation

# measures of rank
min(x)
quantile(x, 0.25) - x greater than 25% of the values, or less than 75% of the values
max(x)

# measures of position
first(x) - similar to x[1]
nth(x, 2) - similar to x[n]
last(x) - similar to x[length(x)]

# counts
n() - takes no arguments, returns the size of the current group
sum(!is.na(x)) - sum of non missing values
n_distinct(x) - number of unique values
count(x, wt = y) - a simple count varible with optional weight variable
```

```
# combine aggregation with logical subsetting
not_cancelled %>%
```

```
  group_by(year, month, day) %>%
  summarize(
    # average delay
    avg_delay1 = mean(arr_delay),
    # average positive delay
    avg_delay2 = mean(arr_delay[arr_delay > 0])
  )
```

```
## # A tibble: 365 x 5
## # Groups:   year, month [?]
##     year month   day avg_delay1 avg_delay2
##    <int> <int> <int>      <dbl>      <dbl>
##  1  2013     1     1     12.7         32.5
##  2  2013     1     2     12.7         32.0
##  3  2013     1     3      5.73        27.7
##  4  2013     1     4     -1.93        28.3
##  5  2013     1     5     -1.53        22.6
##  6  2013     1     6      4.24        24.4
##  7  2013     1     7     -4.95        27.8
##  8  2013     1     8     -3.23        20.8
##  9  2013     1     9     -0.264       25.6
## 10  2013     1    10     -5.90        27.3
## # ... with 355 more rows
```

```
# why is distance to some destinations more variable than others?
not_cancelled %>%
  group_by(dest) %>%
  summarize(distance_sd = sd(distance)) %>%
  arrange(desc(distance_sd))
```

```
## # A tibble: 104 x 2
##    dest  distance_sd
##    <chr>       <dbl>
##  1 EGE         10.5
##  2 SAN         10.4
##  3 SFO         10.2
##  4 HNL         10.0
##  5 SEA          9.98
##  6 LAS          9.91
##  7 PDX          9.87
##  8 PHX          9.86
##  9 LAX          9.66
## 10 IND          9.46
## # ... with 94 more rows
```

```
# When do the first and last flights leave each day?
not_cancelled %>%
  group_by(year, month, day) %>%
  summarize(
    first = min(dep_time),
    last = max(dep_time)
  )
```

```
## # A tibble: 365 x 5
## # Groups:   year, month [?]
```

```
##     year month   day first  last
##    <int> <int> <int> <dbl> <dbl>
##  1  2013     1     1   517  2356
##  2  2013     1     2    42  2354
##  3  2013     1     3    32  2349
##  4  2013     1     4    25  2358
##  5  2013     1     5    14  2357
##  6  2013     1     6    16  2355
##  7  2013     1     7    49  2359
##  8  2013     1     8   454  2351
##  9  2013     1     9     2  2252
## 10  2013     1    10     3  2320
## # ... with 355 more rows
```

```
# alternatively
not_cancelled %>%
  group_by(year, month, day) %>%
  summarize(
    first(dep_time),
    last(dep_time)
  )
```

```
## # A tibble: 365 x 5
## # Groups:   year, month [?]
##     year month   day `first(dep_time)` `last(dep_time)`
##    <int> <int> <int>             <int>            <int>
##  1  2013     1     1               517             2356
##  2  2013     1     2                42             2354
##  3  2013     1     3                32             2349
##  4  2013     1     4                25             2358
##  5  2013     1     5                14             2357
##  6  2013     1     6                16             2355
##  7  2013     1     7                49             2359
##  8  2013     1     8               454             2351
##  9  2013     1     9                 2             2252
## 10  2013     1    10                 3             2320
## # ... with 355 more rows
```

```
# alternatively again
not_cancelled %>%
  group_by(year, month, day) %>%
  mutate(r = min_rank(desc(dep_time))) %>%
  filter(r %in% range(r))
```

```
## # A tibble: 770 x 20
## # Groups:   year, month, day [365]
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1     2356           2359        -3      425
##  3  2013     1     2       42           2359        43      518
##  4  2013     1     2     2354           2359        -5      413
##  5  2013     1     3       32           2359        33      504
##  6  2013     1     3     2349           2359       -10      434
##  7  2013     1     4       25           2359        26      505
##  8  2013     1     4     2358           2359        -1      429
```

```
##  9  2013     1     4     2358           2359        -1      436
## 10  2013     1     5       14           2359        15      503
## # ... with 760 more rows, and 13 more variables: sched_arr_time <int>,
## #   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>, r <int>
```

```
# Which destinations have the most carriers?
not_cancelled %>%
  group_by(dest) %>%
  summarize(carriers = n_distinct(carrier)) %>%
  arrange(desc(carriers))
```

```
## # A tibble: 104 x 2
##    dest  carriers
##    <chr>    <int>
##  1 ATL          7
##  2 BOS          7
##  3 CLT          7
##  4 ORD          7
##  5 TPA          7
##  6 AUS          6
##  7 DCA          6
##  8 DTW          6
##  9 IAD          6
## 10 MSP          6
## # ... with 94 more rows
```

```
not_cancelled %>%
    count(tailnum, wt = distance)
```

```
## # A tibble: 4,037 x 2
##    tailnum      n
##    <chr>    <dbl>
##  1 D942DN    3418
##  2 N0EGMQ  239143
##  3 N10156  109664
##  4 N102UW   25722
##  5 N103US   24619
##  6 N104UW   24616
##  7 N10575  139903
##  8 N105UW   23618
##  9 N107US   21677
## 10 N108UW   32070
## # ... with 4,027 more rows
```

```
# How many flights left before 5 am? These usually represent delayed flights from the previous day

not_cancelled %>%
  group_by(year, month, day) %>%
  summarize(n_early = sum(dep_time < 500))
```

```
## # A tibble: 365 x 4
## # Groups:   year, month [?]
##     year month   day n_early
##    <int> <int> <int>   <int>
```

```
##  1  2013     1     1       0
##  2  2013     1     2       3
##  3  2013     1     3       4
##  4  2013     1     4       3
##  5  2013     1     5       3
##  6  2013     1     6       2
##  7  2013     1     7       2
##  8  2013     1     8       1
##  9  2013     1     9       3
## 10  2013     1    10       3
## # ... with 355 more rows
```

```
# What proportionn of flights are delayed by more than an hour?

not_cancelled %>%
  group_by(year, month, day) %>%
  summarize(hour_perc = mean(arr_delay > 60))
```

```
## # A tibble: 365 x 4
## # Groups:   year, month [?]
##     year month   day hour_perc
##    <int> <int> <int>     <dbl>
##  1  2013     1     1    0.0722
##  2  2013     1     2    0.0851
##  3  2013     1     3    0.0567
##  4  2013     1     4    0.0396
##  5  2013     1     5    0.0349
##  6  2013     1     6    0.0470
##  7  2013     1     7    0.0333
##  8  2013     1     8    0.0213
##  9  2013     1     9    0.0202
## 10  2013     1    10    0.0183
## # ... with 355 more rows
```

```
# Grouping by multiple variables
# each iteration peels off a layer

daily <- group_by(flights, year, month, day)
(per_day <- summarize(daily, flights = n()))
```

```
## # A tibble: 365 x 4
## # Groups:   year, month [?]
##     year month   day flights
##    <int> <int> <int>   <int>
##  1  2013     1     1     842
##  2  2013     1     2     943
##  3  2013     1     3     914
##  4  2013     1     4     915
##  5  2013     1     5     720
##  6  2013     1     6     832
##  7  2013     1     7     933
##  8  2013     1     8     899
##  9  2013     1     9     902
## 10  2013     1    10     932
## # ... with 355 more rows
```

```
(per_month <- summarize(per_day, flights = sum(flights)))
```

```
## # A tibble: 12 x 3
## # Groups:   year [?]
##     year month flights
##    <int> <int>   <int>
##  1  2013     1   27004
##  2  2013     2   24951
##  3  2013     3   28834
##  4  2013     4   28330
##  5  2013     5   28796
##  6  2013     6   28243
##  7  2013     7   29425
##  8  2013     8   29327
##  9  2013     9   27574
## 10  2013    10   28889
## 11  2013    11   27268
## 12  2013    12   28135
```

```
(per_year <- summarize(per_month, flights = sum(flights)))
```

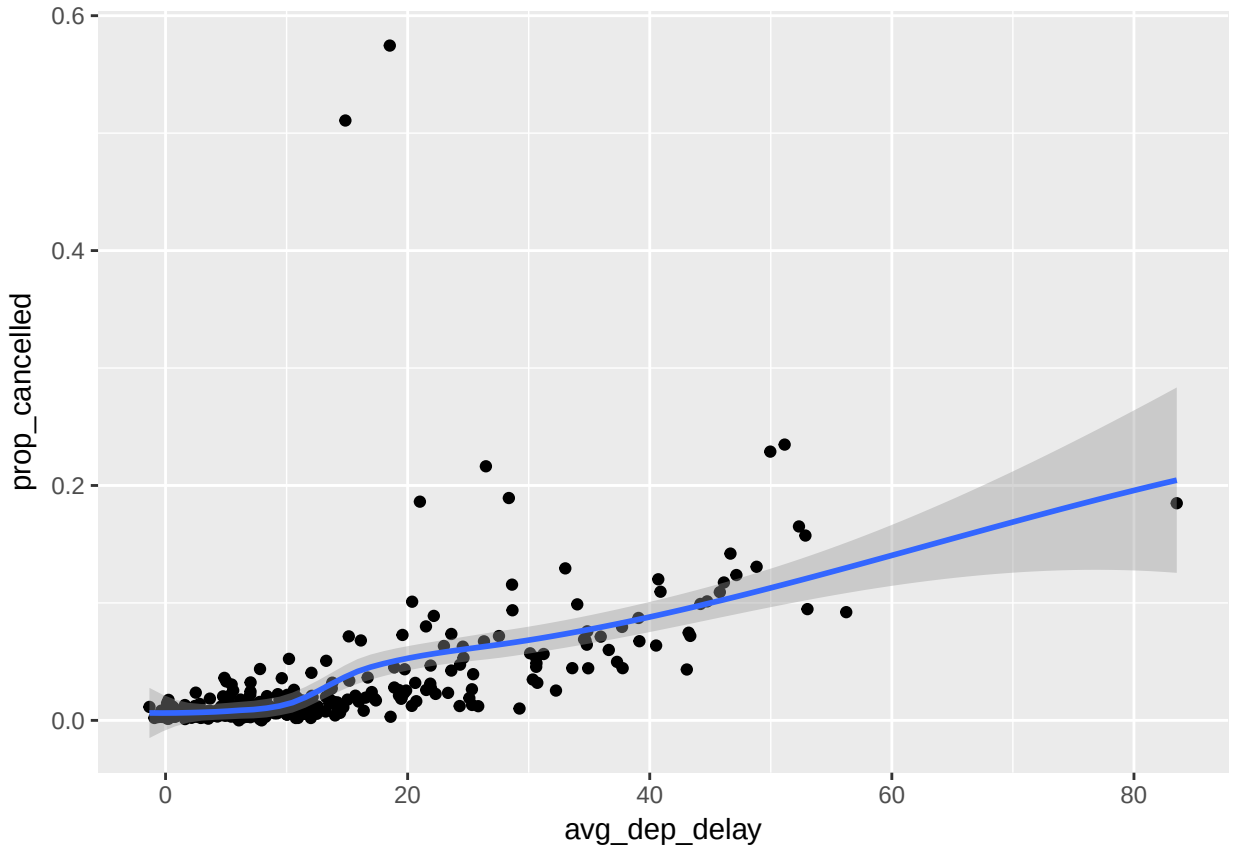
```
## # A tibble: 1 x 2
##    year flights
##   <int>   <int>
## 1  2013  336776
```

```
# to ungroup data

daily %>%
  ungroup() %>%  # no longer grouped by date
  summarize(flights = n())
```

```
## # A tibble: 1 x 1
##   flights
##     <int>
## 1  336776
```

```
# number of cancelled flights per day
cancelled_delayed <- flights %>%
  mutate(cancelled = (is.na(arr_delay) | is.na(dep_delay))) %>%
  group_by(year, month, day) %>%
  summarize(prop_cancelled = mean(cancelled),
            avg_dep_delay = mean(dep_delay, na.rm = TRUE))

ggplot(data = cancelled_delayed, aes(y = prop_cancelled, x = avg_dep_delay)) +
  geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'loess'
```




```
# which carrier has the worst delays?

flights %>%
  group_by(carrier) %>%
  summarise(avg_delay = mean(arr_delay, na.rm = TRUE)) %>%
  arrange(desc(avg_delay))
```

```
## # A tibble: 16 x 2
##    carrier avg_delay
##    <chr>       <dbl>
##  1 F9         21.9
##  2 FL         20.1
##  3 EV         15.8
##  4 YV         15.6
##  5 OO         11.9
##  6 MQ         10.8
##  7 WN          9.65
##  8 B6          9.46
##  9 9E          7.38
## 10 UA          3.56
## 11 US          2.13
## 12 VX          1.76
## 13 DL          1.64
## 14 AA          0.364
## 15 HA         -6.92
## 16 AS         -9.93
```

```
# num of flights before first delay of at least 60 minutes

flights %>%
  arrange(tailnum, year, month, day) %>% # order columns
  group_by(tailnum) %>% # group by tail number of plane
  mutate(delay_gt1hr = dep_delay > 60) %>% # create variable with all values > 60 min dep delay
  mutate(before_delay = cumsum(delay_gt1hr)) %>% # create cumulative sum of flights before the first de
  filter(before_delay < 1) %>% # remove planes which have first flight delayed
  count(sort = TRUE) # show sum of each planes flights in order from most to least
```

```
## # A tibble: 3,755 x 2
## # Groups:   tailnum [3,755]
##    tailnum     n
##    <chr>   <int>
##  1 N954UW    206
##  2 N952UW    163
##  3 N957UW    142
##  4 N5FAAA    117
##  5 N38727     99
##  6 N3742C     98
##  7 N5EWAA     98
##  8 N705TW     97
##  9 N765US     97
## 10 N635JB     94
## # ... with 3,745 more rows
```

Grouped Mutates(and Filters)

```
# grouping is most useful in conjunction with summarize, but you can also do convenient operations with

# find the worst members of each group
flights_sml %>%
  group_by(year, month, day) %>%
  filter(rank(desc(arr_delay)) < 10)
```

```
## # A tibble: 3,306 x 7
## # Groups:   year, month, day [365]
##     year month   day dep_delay arr_delay distance air_time
##    <int> <int> <int>     <dbl>     <dbl>    <dbl>    <dbl>
##  1  2013     1     1       853       851      184       41
##  2  2013     1     1       290       338     1134      213
##  3  2013     1     1       260       263      266       46
##  4  2013     1     1       157       174      213       60
##  5  2013     1     1       216       222      708      121
##  6  2013     1     1       255       250      589      115
##  7  2013     1     1       285       246     1085      146
##  8  2013     1     1       192       191      199       44
##  9  2013     1     1       379       456     1092      222
## 10  2013     1     2       224       207      550       94
## # ... with 3,296 more rows
```

```
# find all groups bigger than a threshold:
```

```r
popular_dests <- flights %>%
  group_by(dest) %>%
  filter(n() > 365)

# standardize to compute group metrics
popular_dests %>%
  filter(arr_delay > 0) %>%
  mutate(prop_delay = arr_delay / sum(arr_delay)) %>%
  select(year:day, dest, arr_delay, prop_delay)
```

```
## # A tibble: 131,106 x 6
## # Groups:   dest [77]
##     year month   day dest  arr_delay prop_delay
##    <int> <int> <int> <chr>     <dbl>      <dbl>
##  1  2013     1     1 IAH          11  0.000111
##  2  2013     1     1 IAH          20  0.000201
##  3  2013     1     1 MIA          33  0.000235
##  4  2013     1     1 ORD          12  0.0000424
##  5  2013     1     1 FLL          19  0.0000938
##  6  2013     1     1 ORD           8  0.0000283
##  7  2013     1     1 LAX           7  0.0000344
##  8  2013     1     1 DFW          31  0.000282
##  9  2013     1     1 ATL          12  0.0000400
## 10  2013     1     1 DTW          16  0.000116
## # ... with 131,096 more rows
```

```r
# which plane has the worst on-time record?

flights %>%
  group_by(tailnum) %>% # make groups of tailnums
  summarize(arr_delay = mean(arr_delay, na.rm = TRUE)) %>% # let arr_delay be mean arr_delay
  ungroup() %>% # no longer grouped by date, so mean across all days
  filter(rank(desc(arr_delay)) <= 1) # remove all arr_delay less than or equal to 1 and rank planes by
```

```
## # A tibble: 1 x 2
##   tailnum arr_delay
##   <chr>       <dbl>
## 1 N844MH        320
```

```r
# What time of the day should you fly if you want to avoid delays as much as possible?
# generally it seems that earlier is better

flights %>%
  group_by(hour) %>%
  summarize(arr_delay = mean(arr_delay, na.rm = TRUE)) %>%
  ungroup() %>%
  arrange(arr_delay)
```

```
## # A tibble: 20 x 2
##     hour arr_delay
##    <dbl>     <dbl>
##  1     7    -5.30
##  2     5    -4.80
##  3     6    -3.38
##  4     9    -1.45
```

```
##  5     8    -1.11
##  6    10     0.954
##  7    11     1.48
##  8    12     3.49
##  9    13     6.54
## 10    14     9.20
## 11    23    11.8
## 12    15    12.3
## 13    16    12.6
## 14    18    14.8
## 15    22    16.0
## 16    17    16.0
## 17    19    16.7
## 18    20    16.7
## 19    21    18.4
## 20     1   NaN
```

```
# For each destination, compute the total minutes of delay. For each flight, compute the proportion of

flights %>%
  filter(!is.na(arr_delay), arr_delay > 0) %>%
  group_by(dest) %>%
  mutate(total_delay = sum(arr_delay),
         prop_delay = arr_delay / sum(arr_delay))
```

```
## # A tibble: 133,004 x 21
## # Groups:   dest [103]
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      554            558        -4      740
##  5  2013     1     1      555            600        -5      913
##  6  2013     1     1      558            600        -2      753
##  7  2013     1     1      558            600        -2      924
##  8  2013     1     1      559            600        -1      941
##  9  2013     1     1      600            600         0      837
## 10  2013     1     1      602            605        -3      821
## # ... with 132,994 more rows, and 14 more variables: sched_arr_time <int>,
## #   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>, total_delay <dbl>, prop_delay <dbl>
```

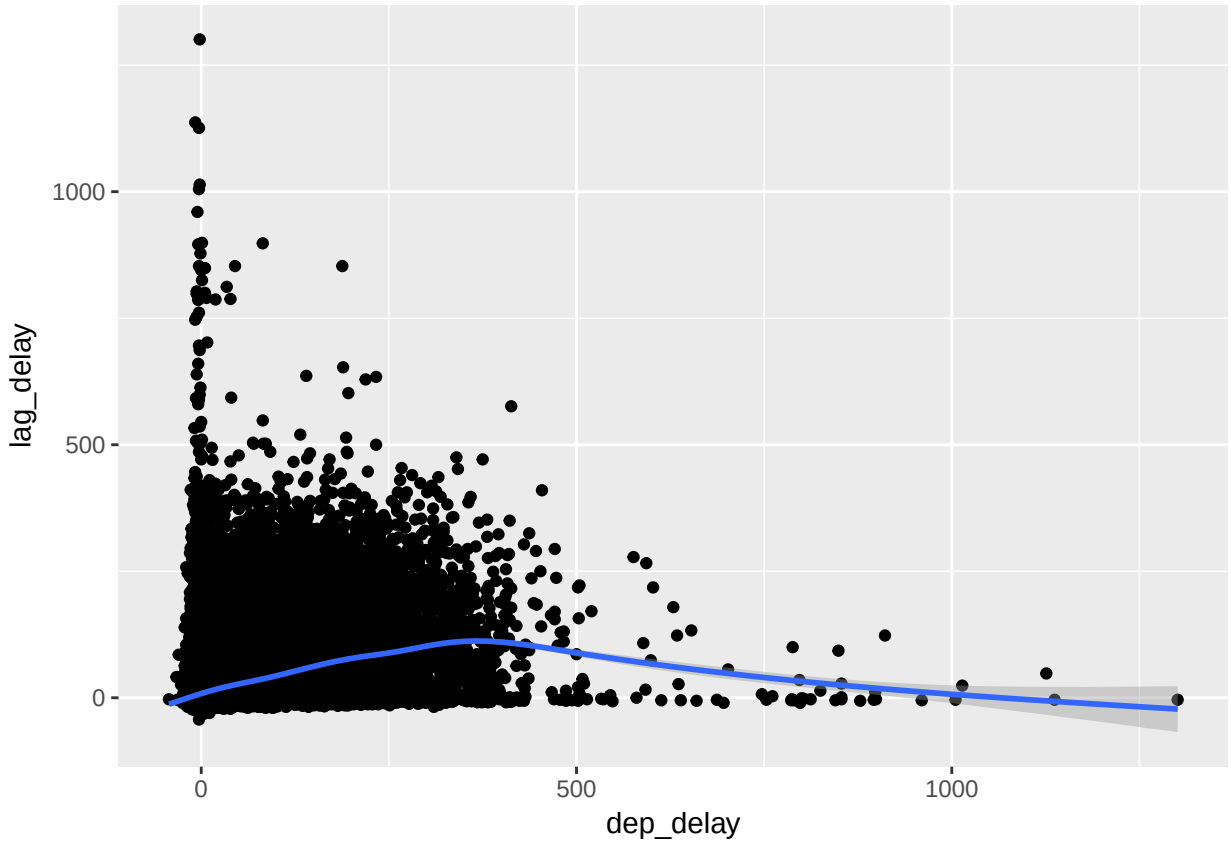
```
# alternatively, considering the delay as relative to the minimum delay for any flight to that destinat

flights %>%
  filter(!is.na(arr_delay), arr_delay > 0) %>%
  group_by(dest) %>%
  mutate(total_delay = sum(arr_delay - min(arr_delay)),
         prop_delay = arr_delay / sum(arr_delay))
```

```
## # A tibble: 133,004 x 21
## # Groups:   dest [103]
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
```

```
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      554            558        -4      740
##  5  2013     1     1      555            600        -5      913
##  6  2013     1     1      558            600        -2      753
##  7  2013     1     1      558            600        -2      924
##  8  2013     1     1      559            600        -1      941
##  9  2013     1     1      600            600         0      837
## 10  2013     1     1      602            605        -3      821
## # ... with 132,994 more rows, and 14 more variables: sched_arr_time <int>,
## #   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>, total_delay <dbl>, prop_delay <dbl>
```

```r
# Using lag(), explore how the delay of a flight is related to the delay of the immediately preceding f

flights %>%
  group_by(year, month, day) %>%
  filter(!is.na(dep_delay)) %>%
  mutate(lag_delay = lag(dep_delay)) %>%
  filter(!is.na(lag_delay)) %>%
  ggplot(aes(x = dep_delay, y = lag_delay)) +
    geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'gam'
```



```r
# Can you find flights that are suspiciously fast? Compute the air time of a flight relative to the sho

flights %>%
  filter(!is.na(air_time)) %>%
  group_by(dest) %>%
  mutate(med_time = median(air_time),
         fast = (air_time - med_time) / med_time) %>%
  arrange(fast) %>%
  select(air_time, med_time, fast, dep_time, sched_dep_time, arr_time, sched_arr_time) %>%
  head(15)
```

```
## Adding missing grouping variables: `dest`
```

```
## # A tibble: 15 x 8
## # Groups:   dest [9]
##    dest  air_time med_time   fast dep_time sched_dep_time arr_time
##    <chr>    <dbl>    <dbl>  <dbl>    <int>          <int>    <int>
##  1 BOS         21       38 -0.447     1450           1500     1547
##  2 ATL         65      112 -0.420     1709           1700     1923
##  3 GSP         55       92 -0.402     2040           2025     2225
##  4 BOS         23       38 -0.395     1954           2000     2131
##  5 BNA         70      113 -0.381     1914           1910     2045
##  6 MSP         93      149 -0.376     1558           1513     1745
##  7 CVG         62       95 -0.347     1359           1343     1523
##  8 PIT         40       61 -0.344     1557           1610     1723
##  9 PHL         21       32 -0.344     2153           2129     2247
## 10 PHL         21       32 -0.344     2123           2130     2211
## 11 PHL         21       32 -0.344     2026           1935     2131
## 12 RIC         35       53 -0.340     1812           1639     1942
## 13 BOS         26       38 -0.316     1711           1700     1827
## 14 BOS         26       38 -0.316     1200           1200     1254
## 15 PHL         22       32 -0.312     2125           2129     2224
## # ... with 1 more variable: sched_arr_time <int>
```

```r
# with a z-score

flights %>%
  filter(!is.na(air_time)) %>%
  group_by(dest) %>%
  mutate(air_time_mean = mean(air_time),
         air_time_sd = sd(air_time),
         z_score = (air_time - air_time_mean) / air_time_sd) %>%
  arrange(z_score) %>%
  select(z_score, air_time_mean, air_time_sd, air_time, dep_time, sched_dep_time, arr_time, sched_arr_t
```

```
## Adding missing grouping variables: `dest`
```

```
## # A tibble: 327,346 x 9
## # Groups:   dest [104]
##    dest  z_score air_time_mean air_time_sd air_time dep_time
##    <chr>   <dbl>         <dbl>       <dbl>    <dbl>    <int>
##  1 MSP     -4.90         151.        11.8        93     1558
##  2 ATL     -4.88         113.         9.81       65     1709
##  3 GSP     -4.72          93.4        8.13       55     2040
##  4 BNA     -4.05         114.        11.0        70     1914
##  5 CVG     -3.98          96.0        8.52       62     1359
```

```
##  6 BOS     -3.63          39.0        4.95       21     1450
##  7 PBI     -3.57         145.        11.3       105     1559
##  8 RIC     -3.50          54.0        5.44       35     1812
##  9 BUF     -3.39          55.7        5.23       38     2307
## 10 SEA     -3.37         328.        15.6       275     1533
## # ... with 327,336 more rows, and 3 more variables: sched_dep_time <int>,
## #   arr_time <int>, sched_arr_time <int>
```

```
# find all destinations that are flown by at least 2 carriers and use that information to rank the carr

flights %>%
  group_by(dest, carrier) %>%
  count(carrier) %>%
  group_by(carrier) %>%
  count(sort = TRUE)
```

```
## # A tibble: 16 x 2
## # Groups:   carrier [16]
##    carrier    nn
##    <chr>   <int>
##  1 EV         61
##  2 9E         49
##  3 UA         47
##  4 B6         42
##  5 DL         40
##  6 MQ         20
##  7 AA         19
##  8 WN         11
##  9 US          6
## 10 OO          5
## 11 VX          5
## 12 FL          3
## 13 YV          3
## 14 AS          1
## 15 F9          1
## 16 HA          1
```

```
filter(airlines, carrier == "EV")
```

```
## # A tibble: 1 x 2
##   carrier name
##   <chr>   <chr>
## 1 EV      ExpressJet Airlines Inc.
```