Data Transformations: ncyflights13$flights_1

Monica Buczynski

08/05/2021

Note: The purpose of this document is to showcase a sample of skills covered in *R for Data Science* (chapter: Data Transformations) by Garrett Grolemund and Hadley Wickham. All scripts were taken from https://r4ds.had.co.nz/transform.html and https://jrnold.github.io/r4ds-exercise-solutions/index.html. The code for each exercise was studied carefully for understanding and then was retyped manually into R to maximize the learning experience; however, many of the original scripts were altered for further experimentation and presentation aesthetics.

The skills that I focused on include:

- Filter rows with *filter()*
- Arrange rows with *arrange()*
- Select columns with *select()*
- Add new variables with *mutate()*
- Grouped summaries with *summarise()*
- Grouped mutates (and filters)

```r
# View first row of data
head(flights, 1)
```

```
## # A tibble: 1 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
## # ... 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>
```

Find flights that arrived more than two hours late, but didn't leave late.

```r
head(filter(flights, dep_delay <= 0, arr_delay > 120))
```

```
## # 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    27     1419           1420        -1     1754           1550
## 2  2013    10     7     1350           1350         0     1736           1526
## 3  2013    10     7     1357           1359        -2     1858           1654
## 4  2013    10    16      657            700        -3     1258           1056
## 5  2013    11     1      658            700        -2     1329           1015
## 6  2013     3    18     1844           1847        -3       39           2219
## # ... 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>
```

Find flights that flew to Houston (IAH or HOU).

```r
head(filter(flights, dest == "IAH" | dest == "HOU")) # characters need quotation marks
```

```
## # 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      623            627        -4      933            932
## 4  2013     1     1      728            732        -4     1041           1038
## 5  2013     1     1      739            739         0     1104           1038
## 6  2013     1     1      908            908         0     1228           1219
## # ... 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>
```

Find flights that were operated by United, American, or Delta.

```
airlines # to lookup airline codes
```

```
## # A tibble: 16 x 2
##    carrier name
##    <chr>   <chr>
##  1 9E      Endeavor Air Inc.
##  2 AA      American Airlines Inc.
##  3 AS      Alaska Airlines Inc.
##  4 B6      JetBlue Airways
##  5 DL      Delta Air Lines Inc.
##  6 EV      ExpressJet Airlines Inc.
##  7 F9      Frontier Airlines Inc.
##  8 FL      AirTran Airways Corporation
##  9 HA      Hawaiian Airlines Inc.
## 10 MQ      Envoy Air
## 11 OO      SkyWest Airlines Inc.
## 12 UA      United Air Lines Inc.
## 13 US      US Airways Inc.
## 14 VX      Virgin America
## 15 WN      Southwest Airlines Co.
## 16 YV      Mesa Airlines Inc.
```

```
head(filter(flights, carrier == "UA" | carrier == "AA" | carrier == "DL"))
```

```
## # 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      554            600        -6      812            837
## 5  2013     1     1      554            558        -4      740            728
## 6  2013     1     1      558            600        -2      753            745
## # ... 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>
```

Find flights that do NOT have a greater delay than 120 minutes.

```
head(filter(flights, !(arr_delay > 120 | dep_delay >120)))
```

```
## # 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>
```

Find flights that do have a delay that is less than 120 minutes.

```r
head(filter(flights, arr_delay <= 120, dep_delay <= 120))
```

```
## # 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>
```

Find flights that had an arrival delay of two or more hours.

```r
head(filter(flights, arr_delay >=120))
```

```
## # 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      811            630       101     1047            830
## 2  2013     1     1      848           1835       853     1001           1950
## 3  2013     1     1      957            733       144     1056            853
## 4  2013     1     1     1114            900       134     1447           1222
## 5  2013     1     1     1505           1310       115     1638           1431
## 6  2013     1     1     1525           1340       105     1831           1626
## # ... 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>
```

Find flights that were delayed by at least an hour, but made up over 30 minutes in flight.

```r
head(filter(flights, dep_delay >= 60, dep_delay - arr_delay > 30))
```

```
## # 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     2205           1720       285       46           2040
## 2  2013     1     1     2326           2130       116      131             18
## 3  2013     1     3     1503           1221       162     1803           1555
## 4  2013     1     3     1839           1700        99     2056           1950
## 5  2013     1     3     1850           1745        65     2148           2120
## 6  2013     1     3     1941           1759       102     2246           2139
## # ... 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>
```

Find flights that departed between midnight and 6am (inclusive).

```
summary(flights$dep_time) # to find if midnight is denoted as 2400 or 0.
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##       1     907    1401    1349    1744    2400    8255
```

```
head(filter(flights, dep_time == 2400 | dep_time <= 6000))
```

```
## # 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>
```

Sort flights to find the most delayed flight in the dataset.

Flight HA 5 from JFK to HNL had a 1301 minute (21.68 hours) delay on January 9th, 2013.

```
slice_head(flights %>%
       select(dep_delay, carrier, flight, origin, dest, month, day, year) %>%
       arrange(desc(dep_delay)))
```

```
## # A tibble: 1 x 8
##   dep_delay carrier flight origin dest  month   day  year
##       <dbl> <chr>    <int> <chr>  <chr> <int> <int> <int>
## 1      1301 HA          51 JFK    HNL       1     9  2013
```

Find the flight that left the earliest in the dataset.

Flight B6 97 from JFK to DEN departed 43 minutes early than scheduled on December 7th, 2013.

```
slice_head(flights %>%
       select(dep_delay, carrier, flight, origin, dest, month, day, year) %>%
       arrange(dep_delay))
```

```
## # A tibble: 1 x 8
##   dep_delay carrier flight origin dest  month   day  year
##       <dbl> <chr>    <int> <chr>  <chr> <int> <int> <int>
## 1       -43 B6          97 JFK    DEN      12     7  2013
```

Sort flights to find the fastest (highest speed) flight in the dataset.

Flight DL 1499 had the fastest average ground speed of 703.38 miles/hour.

```
slice_head(flights %>%
       mutate(ground_speed = distance/(air_time/60)) %>% # create a new variable, ground_speed
       arrange(desc(distance/air_time)) %>%
       select(ground_speed, carrier, flight, origin, dest, month, day, year))
```

```
## # A tibble: 1 x 8
##   ground_speed carrier flight origin dest  month   day  year
##          <dbl> <chr>    <int> <chr>  <chr> <int> <int> <int>
## 1         703. DL        1499 LGA    ATL       5    25  2013
```

Which flights traveled the farthest?

Flight HA 51 from JFK to HNL is the longest #flight with a distance of 4,983 miles.

```
slice_head((flights %>%
       select(distance, carrier, flight, origin, dest) %>%
       arrange(desc(distance))))
```

```
## # A tibble: 1 x 5
##   distance carrier flight origin dest
##      <dbl> <chr>    <int> <chr>  <chr>
## 1     4983 HA          51 JFK    HNL
```

Which flight traveled the shortest?

Flight US 1632 from EWR to LGA is the shortest flight with a distance of 17 miles.

```
slice_head(flights %>%
       select(distance, carrier, flight, origin, dest) %>%
       arrange(distance))
```

```
## # A tibble: 1 x 5
##   distance carrier flight origin dest
##      <dbl> <chr>    <int> <chr>  <chr>
## 1       17 US        1632 EWR    LGA
```

For the flights that have a missing dep_time, what other variables are missing? What might these other missing rows represent?

Since arrive_time is also missing, these may be canceled flights.

```
head(filter(flights, is.na(dep_time)))
```

```
## # 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       NA           1630        NA       NA           1815
## 2  2013     1     1       NA           1935        NA       NA           2240
## 3  2013     1     1       NA           1500        NA       NA           1825
## 4  2013     1     1       NA            600        NA       NA            901
## 5  2013     1     2       NA           1540        NA       NA           1747
## 6  2013     1     2       NA           1620        NA       NA           1746
## # ... 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>
```

Identify flights which were not canceled.

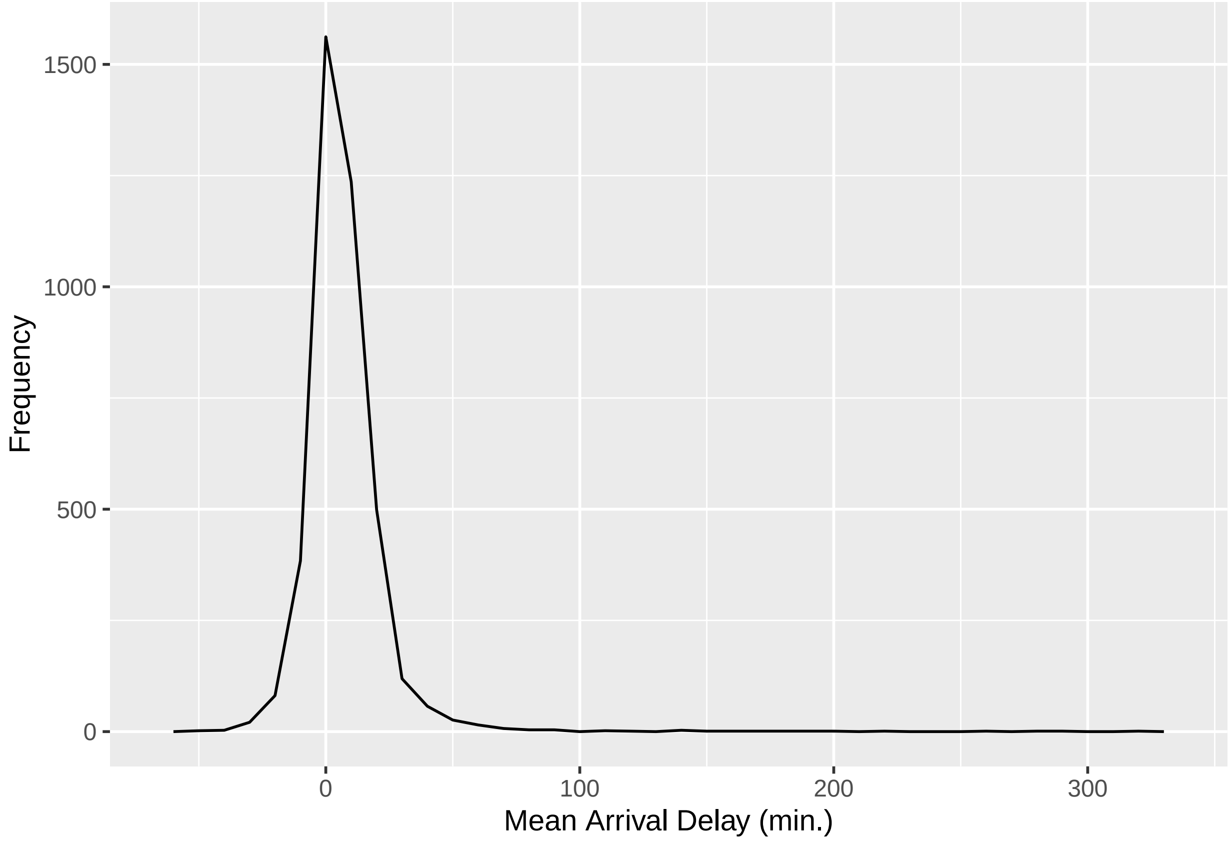
```
not_cancelled <- filter(flights, !is.na(dep_time)|!is.na(arr_time))
```

Planes (identified by their tail number) that have the highest average delays

```
head(delays <- not_cancelled %>%
  filter(!is.na(arr_delay)) %>%
  group_by(tailnum) %>%
  summarise(mean_arr_delay = mean(arr_delay)) %>%
  arrange(desc(mean_arr_delay)), 10)
```

```
## # A tibble: 10 x 2
##    tailnum mean_arr_delay
##    <chr>            <dbl>
##  1 N844MH             320
##  2 N911DA             294
##  3 N922EV             276
##  4 N587NW             264
##  5 N851NW             219
##  6 N928DN             201
##  7 N7715E             188
##  8 N654UA             185
##  9 N665MQ             175.
## 10 N427SW             157
```

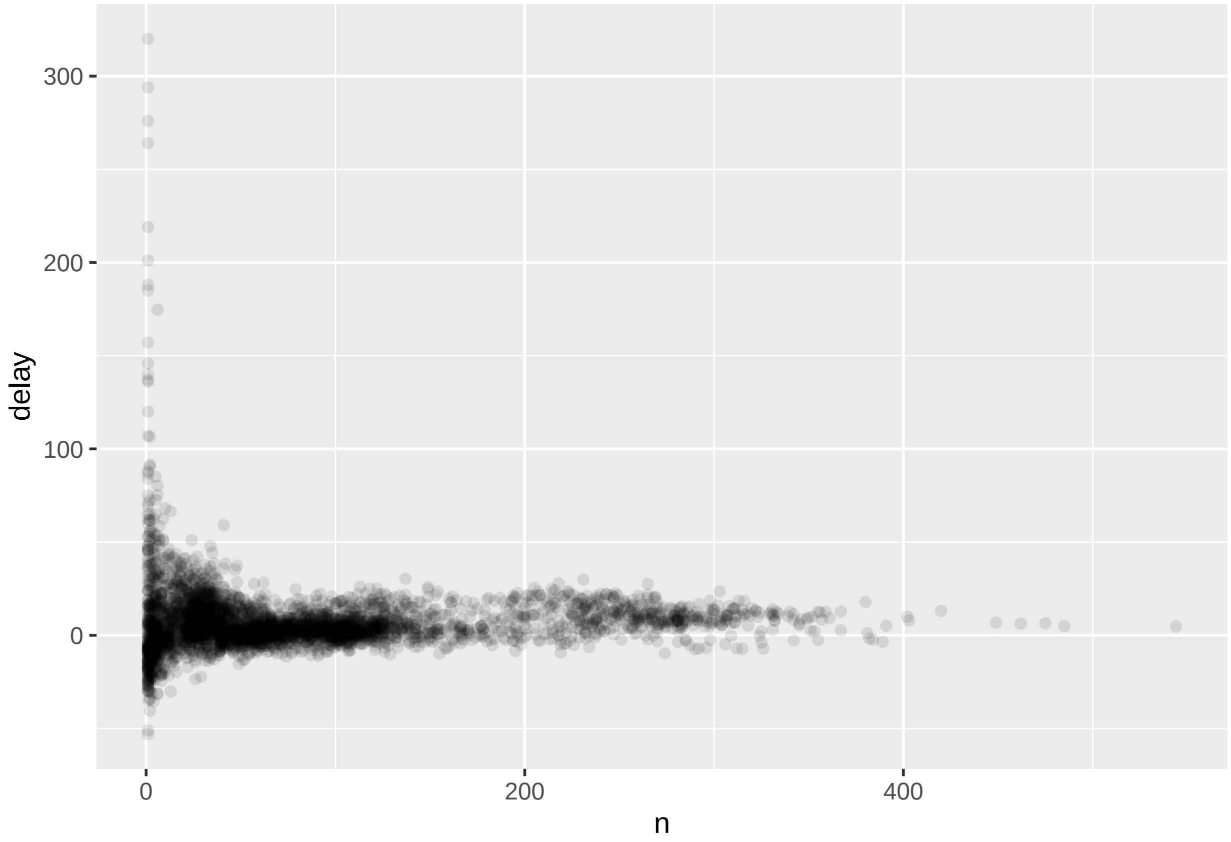
```
ggplot(data = delays, aes(x= mean_arr_delay)) +
  geom_freqpoly(binwidth = 10) +
  xlab("Mean Arrival Delay (min.)") + ylab("Frequency")
```



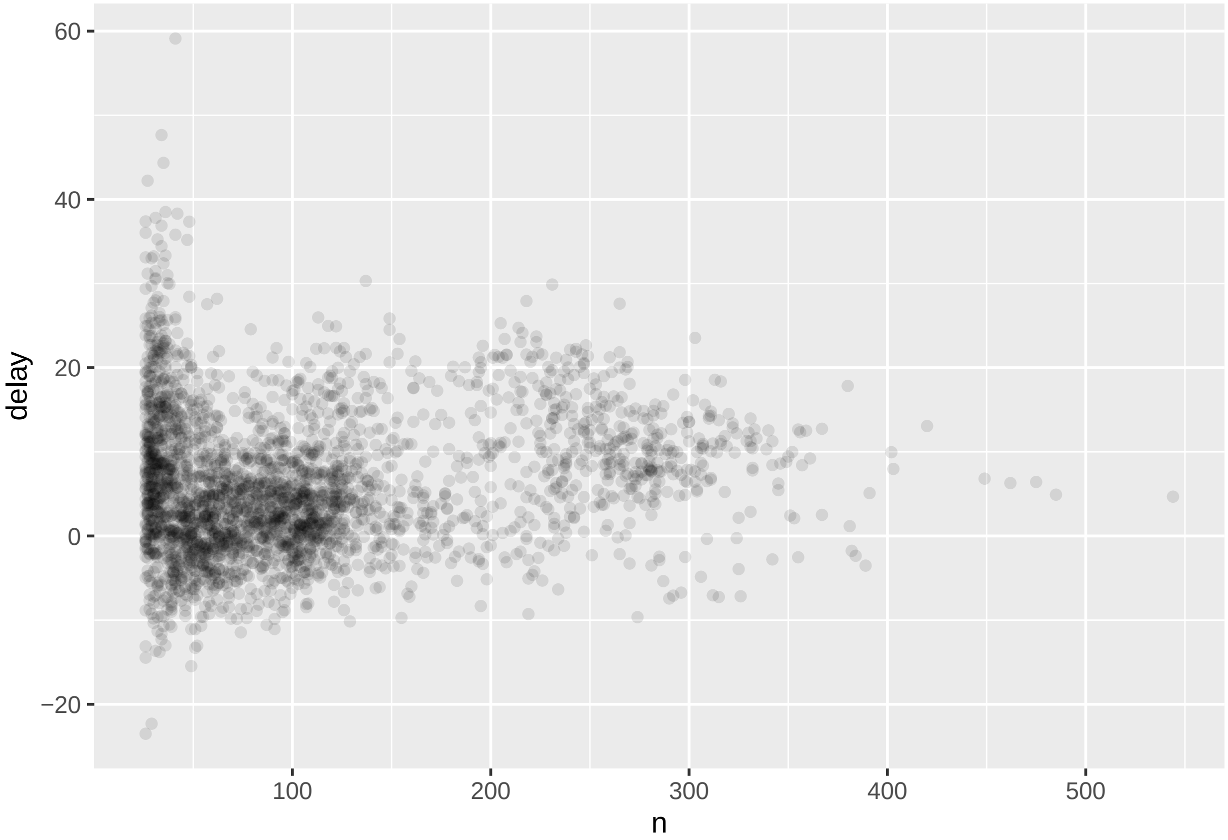
```
#scatterplot of number of flights vs. average delay

head(delays <- not_cancelled %>%
  filter(!is.na(arr_delay)) %>%
  group_by(tailnum) %>%
  summarise(delay = mean(arr_delay), n=n()))
```

```
## # A tibble: 6 x 3
##   tailnum delay     n
##   <chr>   <dbl> <int>
## 1 D942DN  31.5      4
## 2 N0EGMQ   9.98   352
## 3 N10156  12.7    145
## 4 N102UW   2.94    48
## 5 N103US  -6.93    46
## 6 N104UW   1.80    46
```

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



```
# useful to filter out the groups with the smallest numbers of observations,
#so you can see more of the pattern
#and less of the extreme variation in the smallest groups.

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



Using measures of location:mean(x), median(x)

```
head(not_cancelled %>%
       filter(!is.na(arr_delay)) %>%
       group_by(tailnum) %>%
       summarise(mean_arr_delay = mean(arr_delay[arr_delay > 0]),
                 mean_arr_early = mean(arr_delay[arr_delay < 0]))) %>%
  select(tailnum, mean_arr_delay, mean_arr_early, everything()) %>%
  arrange(desc(mean_arr_delay))
```

```
## # A tibble: 6 x 3
##   tailnum mean_arr_delay mean_arr_early
##   <chr>            <dbl>          <dbl>
## 1 N104UW           45.7           -15.5
## 2 D942DN           45.7           -11
## 3 N102UW           40.6           -12.6
## 4 N10156           39.3           -14.5
## 5 N0EGMQ           37.1           -12.7
## 6 N103US            8.31          -12.9
```

Using measures of spread: sd(x), IQR(x), mad(x)

```
head(not_cancelled %>%
  group_by(dest) %>%
  summarize(distance_sd=sd(distance)) %>%
  arrange(desc(distance_sd)))
```

```
## # A tibble: 6 x 2
##   dest  distance_sd
##   <chr>       <dbl>
## 1 EGE         10.5
## 2 SAN         10.3
## 3 SFO         10.2
## 4 HNL         10.0
## 5 SEA          9.98
## 6 LAS          9.91
```

Measures of rank: min(x), quantile(x, 0.25), max(x)

Question: When do the first and last flights leave each day?

```
head(not_cancelled %>%
  group_by(year, month, day) %>%
  summarise(
    first = min(dep_time),
    last = max (dep_time)
  ))
```

```
## # A tibble: 6 x 5
## # Groups:   year, month [1]
##    year month   day first  last
##   <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
```

```
# Counts: n() -> returns the size of the current group
# sum(!is.na(x)) -> count the number of non-missing values
# n_distinct(x) -> count the number of distinct (unique) values

head(not_cancelled %>%
  group_by(dest) %>%
  summarise(carriers =n_distinct(carrier)) %>%
  arrange(desc(carriers)))
```

```
## # A tibble: 6 x 2
##   dest  carriers
##   <chr>    <int>
## 1 ATL          7
## 2 BOS          7
## 3 CLT          7
## 4 ORD          7
## 5 TPA          7
## 6 AUS          6
```

```
head(not_cancelled %>%
  count(dest))
```

```
## # A tibble: 6 x 2
##   dest      n
##   <chr> <int>
## 1 ABQ     254
## 2 ACK     265
## 3 ALB     419
## 4 ANC       8
## 5 ATL   16898
## 6 AUS    2418
```

The total number of miles a plane flew:

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

```
## # A tibble: 6 x 2
##   tailnum      n
##   <chr>    <dbl>
## 1 D942DN    3418
## 2 N0EGMQ  240626
## 3 N10156  110389
## 4 N102UW   25722
## 5 N103US   24619
## 6 N104UW   25157
```

How many flights left before 5am?

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

```
## # A tibble: 6 x 3
## # Groups:   year [1]
##    year month n_early
##   <int> <int>   <int>
## 1  2013     1      75
## 2  2013     2      85
## 3  2013     3     147
## 4  2013     4     148
## 5  2013     5     120
## 6  2013     6     221
```

What proportion of flights are delayed by more than an hour in what month, the frequency, and which airlines?

```
not_cancelled %>%
  filter(!is.na(arr_delay))  %>%
  group_by(month) %>%
  summarise(hour_prop = mean(arr_delay > 60)) %>%
  arrange(desc(hour_prop))
```

```
## # A tibble: 12 x 2
##    month hour_prop
##    <int>     <dbl>
##  1     7    0.145
##  2     6    0.142
##  3    12    0.107
##  4     4    0.102
##  5     3    0.0837
##  6     8    0.0835
##  7     5    0.0795
##  8     1    0.0705
##  9     2    0.0689
## 10     9    0.0476
## 11    10    0.0451
## 12    11    0.0415
```

```
not_cancelled %>%
       filter(!is.na(arr_delay)) %>%
       group_by(carrier) %>%
      summarise(hour_prop = mean(arr_delay > 60), hour_freq = sum(arr_delay > 60)) %>%
                  arrange(desc(hour_prop))
```

```
## # A tibble: 16 x 3
##    carrier hour_prop hour_freq
##    <chr>       <dbl>     <int>
##  1 OO         0.138          4
##  2 YV         0.136         74
##  3 EV         0.133       6803
##  4 F9         0.128         87
##  5 FL         0.113        360
##  6 9E         0.106       1830
##  7 MQ         0.0928      2323
##  8 B6         0.0919      4965
##  9 WN         0.0883      1063
## 10 VX         0.0731       374
## 11 UA         0.0680      3931
## 12 AA         0.0648      2070
## 13 DL         0.0614      2927
## 14 US         0.0472       937
## 15 AS         0.0465        33
## 16 HA         0.0234         8
```

```
head(not_cancelled %>%
       filter(!is.na(arr_delay)) %>%
       group_by(carrier) %>%
       summarise(delay_freq = sum(arr_delay > 60)) %>%
       arrange(desc(delay_freq)))
```

```
## # A tibble: 6 x 2
##   carrier delay_freq
##   <chr>        <int>
## 1 EV            6803
## 2 B6            4965
## 3 UA            3931
## 4 DL            2927
## 5 MQ            2323
## 6 AA            2070
```

Rank airlines by the number of destinations that they fly to, considering only those airports that are flown to by two or more airlines.

```
flights %>%
  # find all airports with > 1 carrier
  group_by(dest) %>%
  mutate(n_carriers = n_distinct(carrier)) %>%
  # n_distinct is a faster and more concise equivalent of length(unique(x))
  #- counts the number of unique values

  filter(n_carriers > 1) %>%

  # rank carriers by number of destinations

  group_by(carrier) %>%
  summarize(n_dest = n_distinct(dest)) %>%
  arrange(desc(n_dest))
```

```
## # A tibble: 16 x 2
##    carrier n_dest
##    <chr>    <int>
##  1 EV          51
##  2 9E          48
##  3 UA          42
##  4 DL          39
##  5 B6          35
##  6 AA          19
##  7 MQ          19
##  8 WN          10
##  9 OO           5
## 10 US           5
## 11 VX           4
## 12 YV           3
## 13 FL           2
## 14 AS           1
## 15 F9           1
## 16 HA           1
```

```
# What airline does the "EV" carrier code correspond to?

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

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

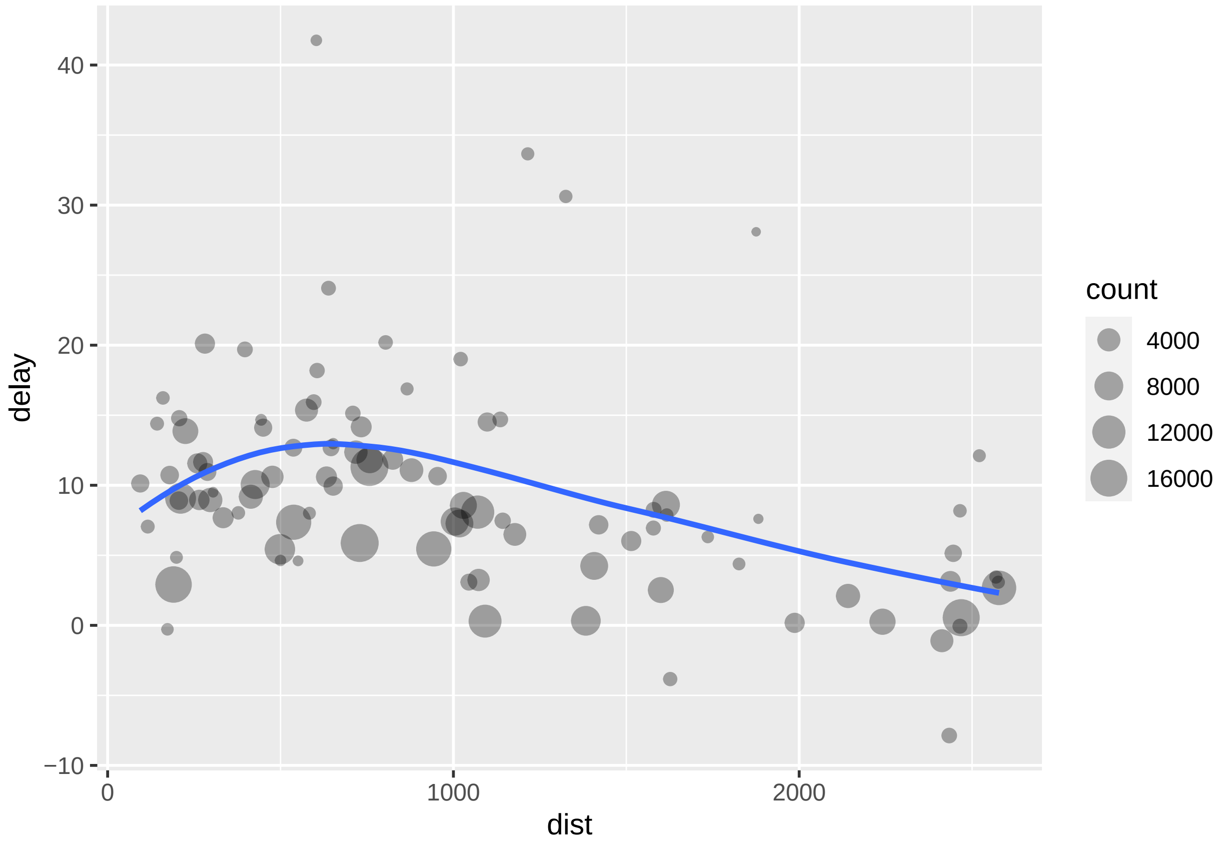
Find the 10 most delayed flights using a ranking function.

```
flights_delayed <- top_n(flights, 10, dep_delay)
select(flights_delayed, month, day, carrier, flight, dep_delay)
```

```
## # A tibble: 10 x 5
##    month   day carrier flight dep_delay
##    <int> <int> <chr>    <int>     <dbl>
##  1     1     9 HA          51      1301
##  2     1    10 MQ        3695      1126
##  3    12     5 AA         172       896
##  4     3    17 DL        2119       911
##  5     4    10 DL        2391       960
##  6     6    15 MQ        3535      1137
##  7     6    27 DL        2007       899
##  8     7    22 MQ        3075      1005
##  9     7    22 DL        2047       898
## 10     9    20 AA         177      1014
```

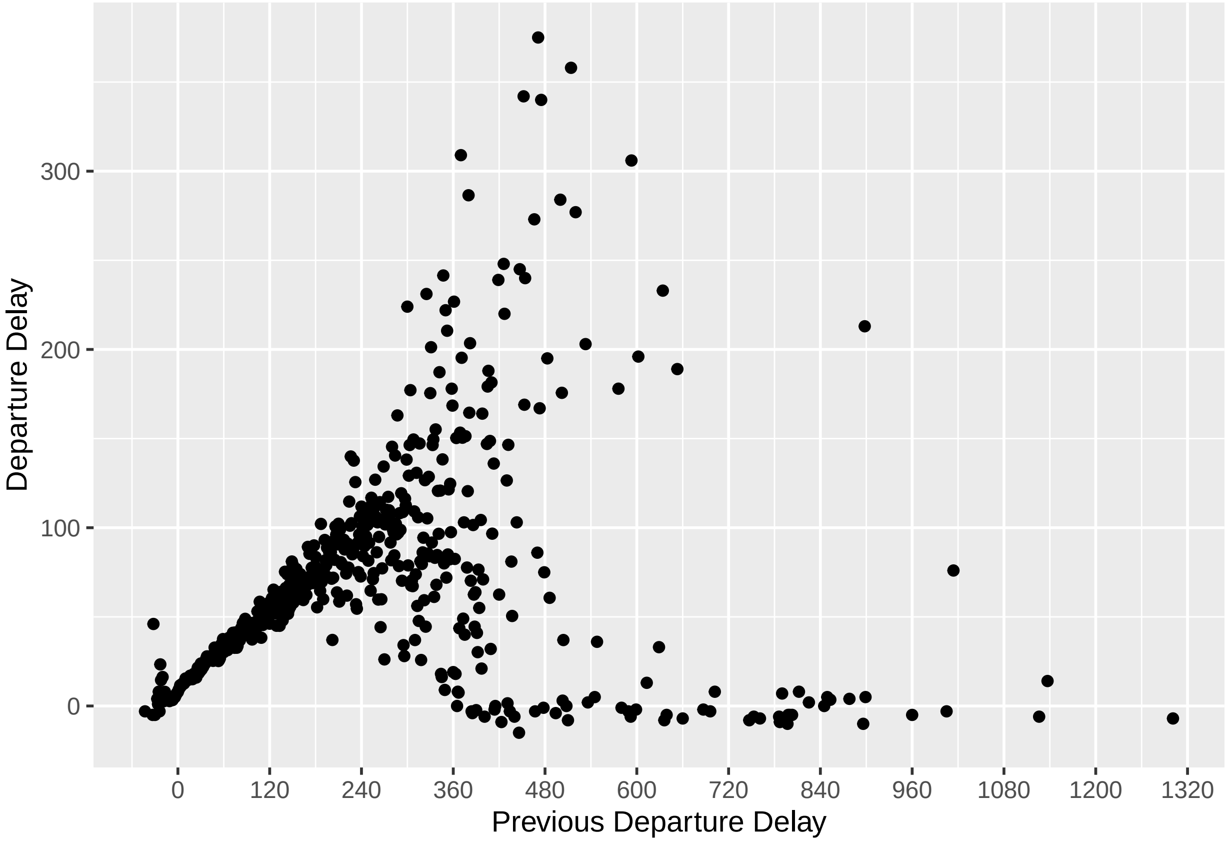
Show graphically the relationship between the distance and average delay for each location.

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

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

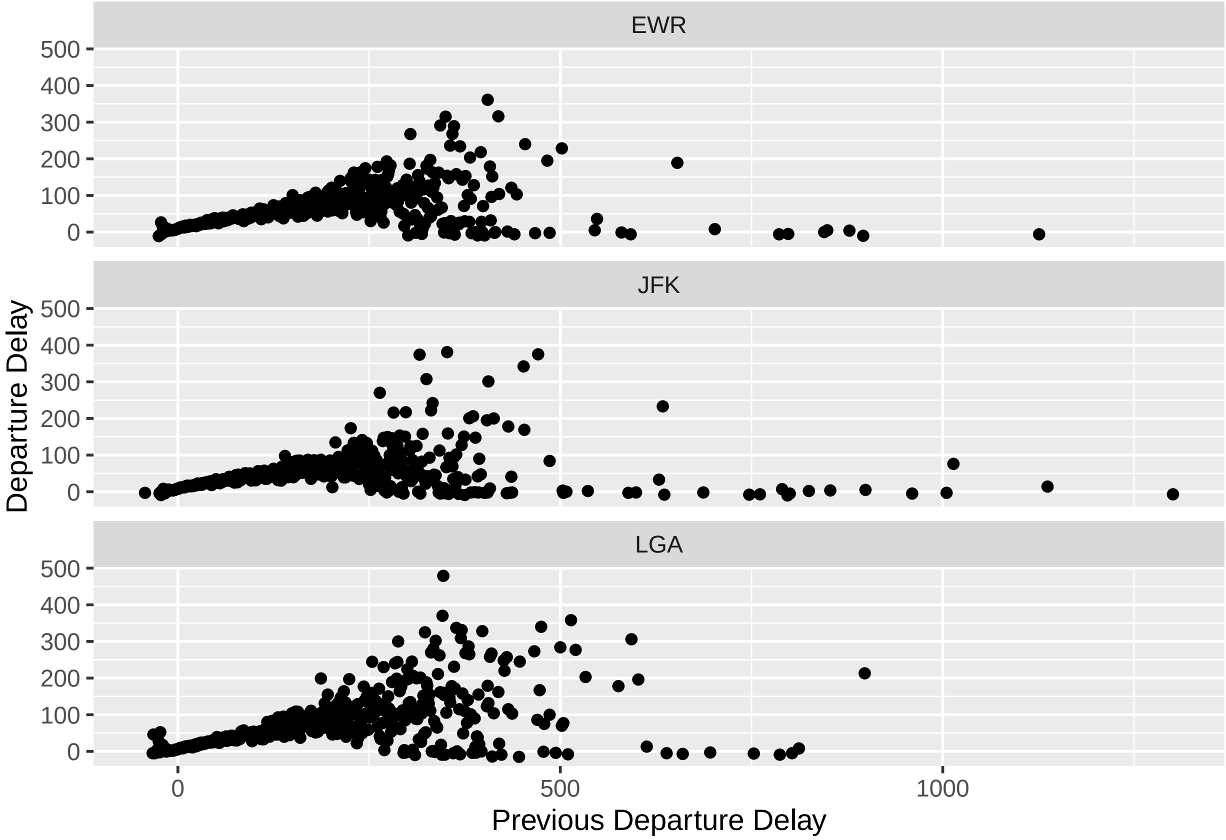


Delays are typically temporally correlated: even once the problem that caused the initial delay has been resolved, later flights are delayed to allow earlier flights to leave. Using lag() explore how the delay of a flight is related to the delay of the immediately preceding flight.

```
# This calculates the departure delay of the preceding flight from the same airport.

lagged_delays <- flights %>%
  arrange(origin, month, day, dep_time) %>%
  group_by(origin) %>%
  mutate(dep_delay_lag = lag(dep_delay)) %>%
  filter(!is.na(dep_delay), !is.na(dep_delay_lag))

# plots the relationship between the mean delay of a flight for all values of the previous flight.
# There seems to be an inverse "U" relationship
# between mean delay of a flight and the mean delay of the preceding flight.

lagged_delays %>%
  group_by(dep_delay_lag) %>%
  summarise(dep_delay_mean = mean(dep_delay)) %>%
  ggplot(aes(y= dep_delay_mean, x=dep_delay_lag)) +
  geom_point() +
  scale_x_continuous(breaks = seq(0, 1500, by = 120)) +
  labs(y = "Departure Delay", x = "Previous Departure Delay")
```



```
# The overall relationship looks similar in all three origin airports.

lagged_delays %>%
  group_by(origin, dep_delay_lag) %>%
  summarise(dep_delay_mean = mean(dep_delay)) %>%
  ggplot(aes(y = dep_delay_mean, x = dep_delay_lag)) +
  geom_point() +
  facet_wrap(~ origin, ncol=1) +
  labs(y = "Departure Delay", x = "Previous Departure Delay")
```


EWR
JFK
LGA
Departure Delay
Previous Departure Delay