Data manipulation with dplyr

group 6: Finn Womack, Lorne Curran, Martin Leandro Uranga Priore, Chenyang Duo, Emerson Webb, Jiarui Xu

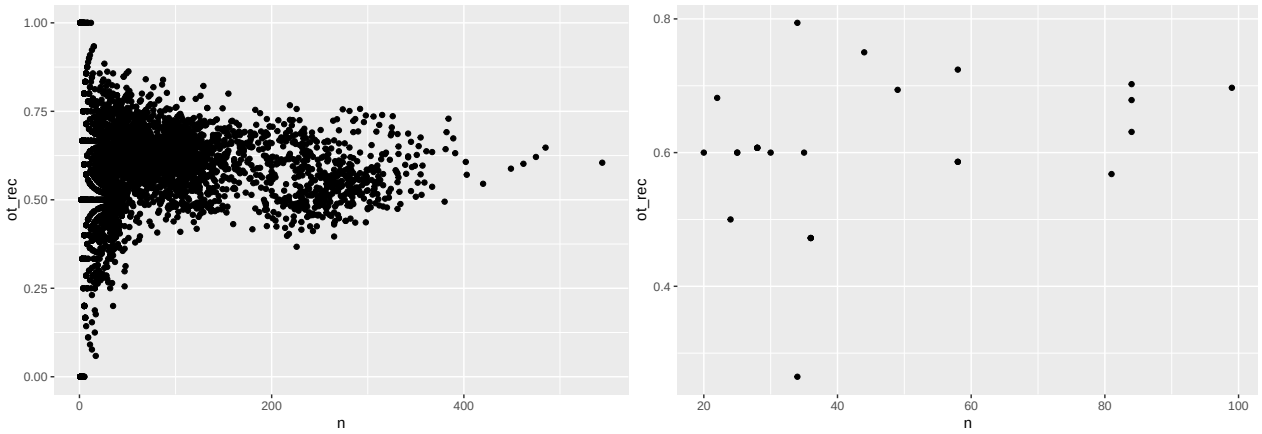
5/10/2019

5.7.1.2 Which plane (tailnum) has the worst on-time record?

There's no unequivocal definition of worst record here. If it's binary: either at the gate by the scheduled time or not, we can deliver a proportion of on-time:

```
## # A tibble: 3 x 3
##   tailnum ot_rec     n
##   <chr>    <dbl> <int>
## 1 N121DE       0     2
## 2 N136DL       0     1
## 3 N143DA       0     1
```

Those with no on-time flights have company, but a whole lot of tailnums have a piddly count of flights. We can look at the whole spread, then drill down to a meaningful breakpoint:



It looks like there's some losers among those with 20 or more flights in 2013. We'll choose the worst of those.

```
## # A tibble: 4 x 3
##   tailnum ot_rec     n
##   <chr>    <dbl> <int>
## 1 N988AT   0.2      35
## 2 N983AT   0.25     32
## 3 N980AT   0.255    47
## 4 N969AT   0.265    34
```

Tailnum N988AT arrived on time for only 20% of its 35 flights.

If on-time record is referring to average number of minutes late:

```
## # A tibble: 2 x 3
##   tailnum ot_rec     n
##   <chr>    <dbl> <int>
## 1 N844MH     320     1
## 2 N911DA     294     1
```

We have a "winner" here (N844MH), but then it only made 1 flight.

5.7.1.4 For each destination, compute the total minutes of delay. For each flight, compute the proportion of the total delay for its destination.

This takes the sum of all delays for each destination (including negatives). Total delay per destination:

```
## # A tibble: 104 x 2
##    dest  total_delay
##    <chr>       <dbl>
##  1 ABQ          1113
##  2 ACK          1281
##  3 ALB          6018
##  4 ANC           -20
##  5 ATL        190260
##  6 AUS         14514
##  7 AVL          2089
##  8 BDL          2904
##  9 BGR          2874
## 10 BHM          4540
## # ... with 94 more rows
```

For each flight, its proportion of the total delay for its destination (including negatives and so can be a negative ratio):

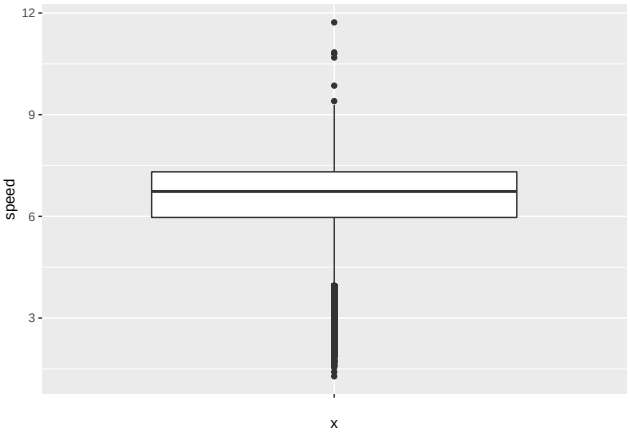
```
## # A tibble: 327,346 x 6
## # Groups:   dest [104]
##    month   day flight delay_prop dest  total_delay
##    <int> <int>  <int>      <dbl> <chr>       <dbl>
##  1     1     1      1  0.000684  LAX          8768
##  2     1     1      1  0.000302  FLL         96153
##  3     1     1      3  0.000125  FLL         96153
##  4     1     1      3  0.000798  LAX          8768
##  5     1     1      4 -0.000122  BUF         40883
##  6     1     1      4 -0.000315  MCO         76185
##  7     1     1      6 -0.0440    SLC           432
##  8     1     1      7  0.00445   SEA         -4270
##  9     1     1      8  0.000660  BUF         40883
## 10     1     1      9  0.0000525 MCO         76185
## # ... with 327,336 more rows
```

Or we can introduce a minor change by assuming that the delay cannot be negative.

```
## # A tibble: 133,004 x 7
## # Groups:   dest [103]
##    month   day dep_time flight delay_prop dest  total_delay
##    <int> <int>    <int>  <int>      <dbl> <chr>       <dbl>
##  1     1     1      856      1  0.0000295 LAX        203226
##  2     1     1     1153      1  0.000143  FLL        202605
##  3     1     1      805      3  0.0000592 FLL        202605
##  4     1     1     1155      3  0.0000344 LAX        203226
##  5     1     1     1527      8  0.000353  BUF         76478
##  6     1     1     1751      9  0.0000194 MCO        206119
##  7     1     1     2229     11  0.000242  FLL        202605
##  8     1     1     1607     12  0.000701  SYR         28547
##  9     1     1     1344     15  0.00254   HNL          8254
## 10     1     1      933     17  0.000207  FLL        202605
## # ... with 132,994 more rows
```

5.7.1.6 Look at each destination. Can you find flights that are suspiciously fast? (i.e. flights that represent a potential data entry error). Compute the air time a flight relative to the shortest flight to that destination. Which flights were most delayed in the air?

We calculate speed:



Lot's of slow ones, but a few on the fast side. We list them by using the 1st quartile + 1.5*IQR rule of thumb.

```
## # A tibble: 6 x 8
##    year month   day flight speed distance air_time   out
##   <int> <int> <int>  <int> <dbl>    <dbl>    <dbl> <dbl>
## 1  2013     5    25   1499 11.7       762       65  9.33
## 2  2013     7     2   4667 10.8      1008       93  9.33
## 3  2013     5    13   4292 10.8       594       55  9.33
## 4  2013     3    23   3805 10.7       748       70  9.33
## 5  2013     1    12   1902  9.86     1035      105  9.33
## 6  2013    11    17    315  9.4      1598      170  9.33
```

The question asks us to compute the air time of a flight relative to the shortest flight to that destination. There's 3 possible origins we don't think we should conflate.

```
## # A tibble: 336,776 x 7
## # Groups:   dest, origin [224]
##    dest  origin month   day flight air_time air_diff
##    <chr> <chr>  <int> <int>  <int>    <dbl>    <dbl>
##  1 ABQ   JFK        4    22   1505      256       44
##  2 ABQ   JFK        4    23   1505      274       62
##  3 ABQ   JFK        4    24   1505      278       66
##  4 ABQ   JFK        4    25   1505      274       62
##  5 ABQ   JFK        4    26   1505      258       46
##  6 ABQ   JFK        4    27   1505      246       34
##  7 ABQ   JFK        4    28   1505      243       31
##  8 ABQ   JFK        4    29   1505      239       27
##  9 ABQ   JFK        4    30   1505      236       24
## 10 ABQ   JFK        5     1   1505      246       34
## # ... with 336,766 more rows
```

The slowest per destination, per origin:

```
## # A tibble: 247 x 7
## # Groups:   dest, origin [223]
##    dest  origin month   day flight air_time max_air
##    <chr> <chr>  <int> <int>  <int>    <dbl>   <dbl>
```

```
##  1 ABQ   JFK       12    10     65       318      106
##  2 ACK   JFK        6    29   1491       141      106
##  3 ALB   EWR        5     5   4117        50       26
##  4 ANC   EWR        8     3    887       434       46
##  5 ATL   EWR        7     6    926       176       88
##  6 ATL   JFK        3    11     95       172       83
##  7 ATL   LGA        6    19   2247       175      110
##  8 AUS   EWR        4     2   1178       301      127
##  9 AUS   JFK       12    20   1295       301      126
## 10 AVL   EWR       12    30   4175       119       43
## # ... with 237 more rows
```

13.4.6. 1 Compute the average delay by destination, then join on the airports data frame so you can show the spatial distribution of delays. (See R4DS for code to help here.)