COMPLETED: Task 07, dplyr

Patrick McKenzie

28 February 2017

Resources

- R Studio's Data Wrangling Cheat Sheet. List of all R Studio cheat sheets here
- R for Data Science: Chapter 5, 9-12
- Wickham (2014) Tidy Data
- The dplyr vignette
- Regular expressions guide - there are many quickstart guides and cheat sheets on the web. I think this one is pretty good.

Notes on built-in datasets

It makes sense to practice **dplyr** using large-ish data sets, since **dplyr** is designed handle big(ish) data. Because those data sets make the package file size considerably bigger, they are distributed in packages of related data sets. To use the data sets, simply install and load them as you would with a regular pacakge. So, for instance:

```
install.packages("nycflights13")
library(nycflights)
```

We will be using two data packages: **nycflight13** and **babynames**. **nycflights13** contains the five distinct data sets:

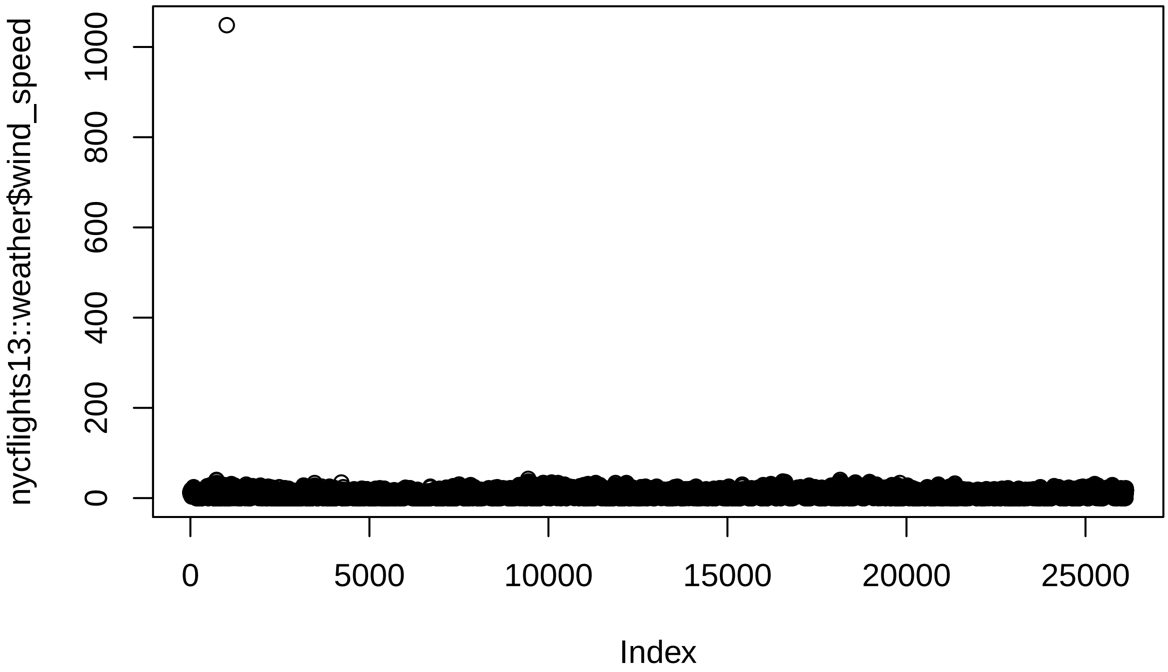
- airlines
- airports
- flights
- planes
- weather

Tasks

All tasks should be done, to the extent possible, using tidyverse syntax including piping and functions.

- Using the `nycflights13::weather`:
  - Determine whether there are any clear outliers in wind speed (`wind_speed`) that should be rejected. If so, filter those bad point(s) and proceed.
  - What direction has the highest median speed at each airport? Make a table and a plot of median wind speed by direction, for each airport. *Optional fun challenge: If you like, this is a rare opportunity to make use of `coord_polar()`.*

```
plot(nycflights13::weather$wind_speed) #Shows one super-high value
```



```
sum(is.na(nycflights13::weather$wind_speed)) #Shows that we have three NA wind speed values
```

```
## [1] 3
```

```
filtered.weather <- nycflights13::weather %>%
  filter(!(wind_speed > 50 | is.na(wind_speed))) #This removes our outlier and NA values
summarised_wind <- filtered.weather %>%
  group_by(origin,wind_dir) %>%
  summarise(median_wind_speed = median(wind_speed)) #This makes a data frame of median wind speeds by a
head(summarised_wind)
```

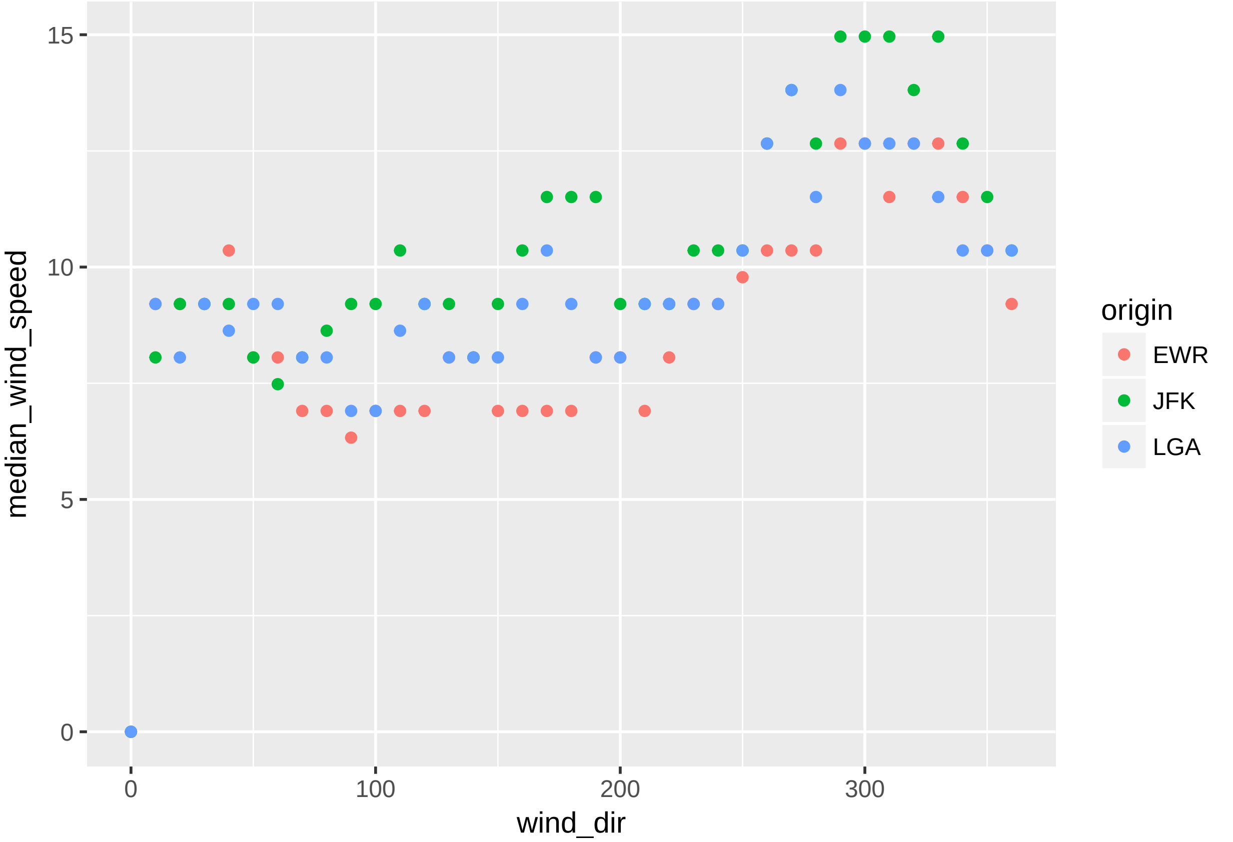
```
## Source: local data frame [6 x 3]
## Groups: origin [1]
##
##   origin wind_dir median_wind_speed
##    <chr>    <dbl>             <dbl>
## 1    EWR        0           0.00000
## 2    EWR       10           9.20624
## 3    EWR       20           9.20624
## 4    EWR       30           9.20624
## 5    EWR       40          10.35702
## 6    EWR       50           8.05546
```

```
highest_med_speeds <- summarised_wind %>%
  group_by(origin) %>%
  filter(median_wind_speed == max(median_wind_speed)) #This selects the direction for each airport at w
highest_med_speeds #Here are the directions with the highest median wind speeds at each airport
```

```
## Source: local data frame [10 x 3]
## Groups: origin [3]
##
##    origin wind_dir median_wind_speed
##     <chr>    <dbl>             <dbl>
## 1     EWR      290          12.65858
## 2     EWR      300          12.65858
## 3     EWR      320          12.65858
## 4     EWR      330          12.65858
## 5     JFK      290          14.96014
## 6     JFK      300          14.96014
## 7     JFK      310          14.96014
## 8     JFK      330          14.96014
## 9     LGA      270          13.80936
## 10    LGA      290          13.80936
```

```
summarised_wind %>%
  ggplot(aes(x = wind_dir, y = median_wind_speed,color = origin)) + geom_point()
```

```
## Warning: Removed 3 rows containing missing values (geom_point).
```



- Using `nycflights13::flights` and `nycflights13::airlines`:
  - Make a table with two columns: airline name (not carrier code) and median distance flown from JFK airport. The table should be arranged in order of decreasing mean flight distance. Hint: use a `_join` function to join `flights` and `airlines`.

```
table <- (nycflights13::flights[,c("carrier","distance")] %>%
  left_join(nycflights13::airlines,by = "carrier"))[,-1] #Makes a tbl of airlines and distances
head(table)
```

```
## # A tibble: 6 × 2
##   distance                   name
##      <dbl>                  <chr>
## 1     1400  United Air Lines Inc.
## 2     1416  United Air Lines Inc.
## 3     1089 American Airlines Inc.
## 4     1576        JetBlue Airways
## 5      762   Delta Air Lines Inc.
## 6      719  United Air Lines Inc.
```

```
summ_table <- table %>%
  group_by(name) %>%
  summarise(median.distance = median(distance),mean.distance = mean(distance)) %>%
  arrange(desc(mean.distance))
summ_table
```

```
## # A tibble: 16 × 3
##                           name median.distance mean.distance
##                          <chr>           <dbl>         <dbl>
## 1       Hawaiian Airlines Inc.            4983     4983.0000
## 2               Virgin America            2475     2499.4822
## 3         Alaska Airlines Inc.            2402     2402.0000
## 4       Frontier Airlines Inc.            1620     1620.0000
## 5        United Air Lines Inc.            1400     1529.1149
## 6       American Airlines Inc.            1096     1340.2360
## 7         Delta Air Lines Inc.            1020     1236.9012
## 8              JetBlue Airways            1023     1068.6215
## 9       Southwest Airlines Co.             748      996.2691
## 10 AirTran Airways Corporation             762      664.8294
## 11                   Envoy Air             502      569.5327
## 12    ExpressJet Airlines Inc.             533      562.9917
## 13           US Airways Inc.               529      553.4563
## 14          Endeavor Air Inc.              509      530.2358
## 15       SkyWest Airlines Inc.             419      500.8125
## 16          Mesa Airlines Inc.             229      375.0333
```

- Make a *wide-format* data frame that displays the number of flights that leave Newark ("EWR") airport each month, from each airline

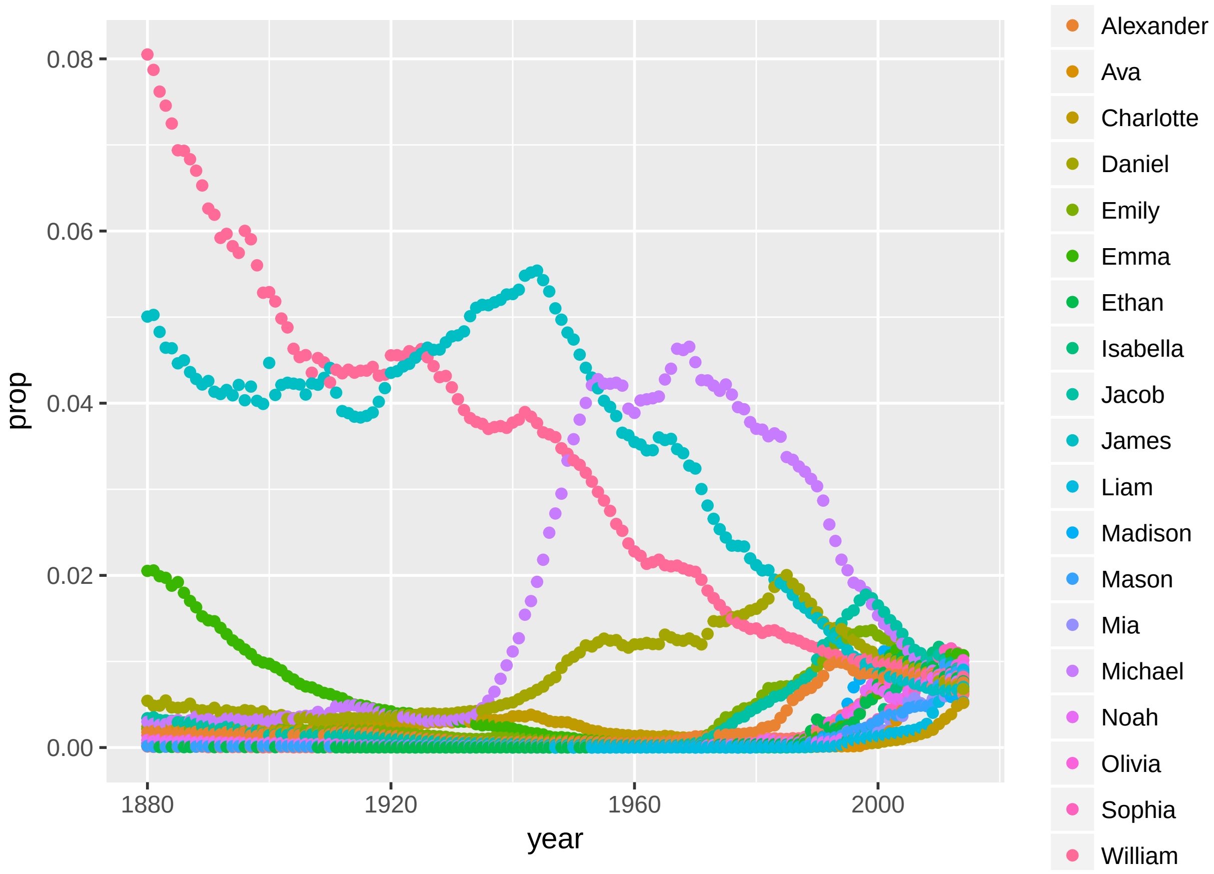
```
numflights_table <- nycflights13::flights %>%
  group_by(month,origin) %>%
  summarise(number.flights = length(flight)) %>%
  spread(month,number.flights)
numflights_table
```

```
## # A tibble: 3 × 13
##   origin   `1`   `2`   `3`   `4`   `5`   `6`   `7`   `8`   `9`  `10`  `11`
## *  <chr> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
## 1    EWR  9893  9107 10420 10531 10592 10175 10475 10359  9550 10104  9707
## 2    JFK  9161  8421  9697  9218  9397  9472 10023  9983  8908  9143  8710
## 3    LGA  7950  7423  8717  8581  8807  8596  8927  8985  9116  9642  8851
## # ... with 1 more variables: `12` <int>
```

- Using the **babynames** dataset:
    - Identify the ten most common male and female names in 2014. Make a plot of their frequency (`prop`) since 1880. (This may require two separate piped statements).
    - Make a single table of the 26th through 29th most common girls names in the year 1896, 1942, and 2016

```
common_names <- babynames[babynames$year == 2014,] %>% #Selects rows from 2014
  group_by(sex) %>% #Groups by sex so that we get the top group from each in the next line
  top_n(10, n) #This gives us a data frame of top 10 names from 2014.
```

```
babynames %>%
  filter(paste0(sex,name) %in% paste0(common_names$sex,common_names$name)) %>% #selects names for each
  ggplot(aes(x = year, y = prop, color = name)) + geom_point() #Plots frequencies since 1880.
```



```
#########################################################
filtered_df <- babynames %>%
  filter(year %in% c("1896", "1942","2014")) %>%
  filter(sex == "F") #This pulls out girls' names from the three years

girlsnames <- integer(0)
for (i in c("1896","1942","2014")) {
  girlsnames <- rbind(girlsnames,(filtered_df %>%
                  filter(year == i))[26:29,]) #this selects rows 26-29 from each year
}
girlsnames
```

```
## # A tibble: 12 × 5
##     year   sex    name     n        prop
## *  <dbl> <chr>   <chr> <int>       <dbl>
## 1   1896     F  Martha  2022 0.008023969
## 2   1896     F  Esther  1964 0.007793805
## 3   1896     F Frances  1964 0.007793805
## 4   1896     F   Edith  1932 0.007666819
## 5   1942     F   Helen 10014 0.007202575
## 6   1942     F Marilyn  9904 0.007123458
## 7   1942     F   Diane  9550 0.006868843
## 8   1942     F   Martha  9513 0.006842231
## 9   2014     F Brooklyn  6767 0.003490782
## 10  2014     F     Lily  6727 0.003470148
## 11  2014     F   Hannah  6512 0.003359240
## 12  2014     F    Layla  6428 0.003315908
```

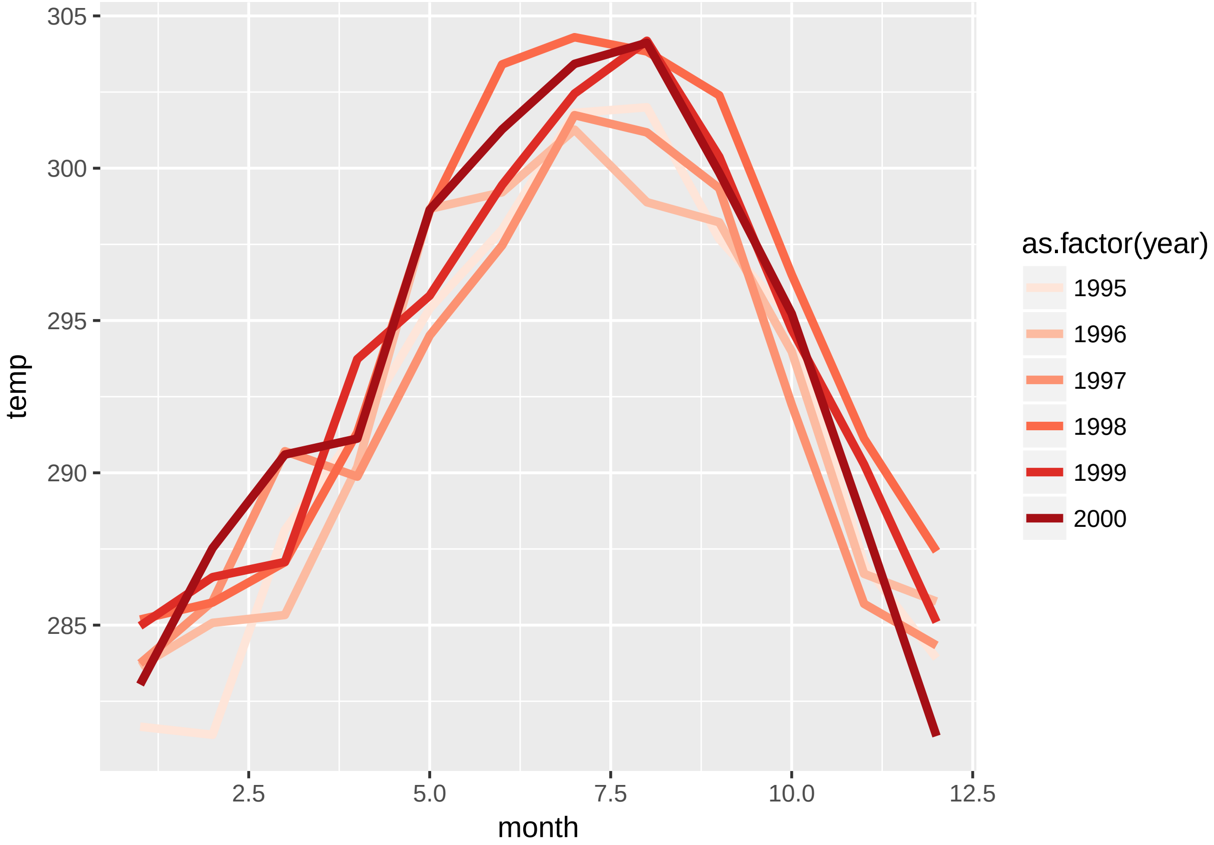
- Write task that involves some of the functions on the Data Wrangling Cheat Sheet and execute it. You may either use your own data or data packages (e.g., the ones listed here).

Plot average temperature per year for the area within five degrees of latitude/longitude of Knoxville.

```
atmos_data <- nasaweather::atmos
Kville.lon.lat <- ggmap::geocode("Knoxville")
```

```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Knoxville&sensor=fal
```

```
Kville_area_data <- atmos_data %>%
  filter(long > (Kville.lon.lat$lon-5) & long < (Kville.lon.lat$lon+5) &
           lat > (Kville.lon.lat$lat-5) & lat < (Kville.lon.lat$lat+5))
Kville_area_data %>%
  group_by(year,month) %>%
  summarise(temp = mean(temp)) %>%
  ggplot(aes(x=month,y=temp,color=as.factor(year))) + geom_line(size = 1.5) + scale_colour_manual(value
```



Optional challenge

Using regular expressions, make a plot of the change in frequency of some letter pattern in names. For instance: how has the frequency of female names ending in “leigh” changed over time relative to names ending in “lee”?