Getting Started with ggplot 2 - Part 1

Tom Blackwood

06/11/2020

Table of Contents

## Introduction

The [First Tutorial](https://brightspace.uhi.ac.uk/d2l/le/content/220648/Home) requested the students to practice using *R-Studio’s Markdown features* to create a document that can be used for **Assignment Part 1**.

This document will show:

1. How to import library items
2. Creating a plot from imported library items
3. Importing a new set of library items and creating a plot from them
4. using the facet\_wrap() feature of ggplot2 to create a series of plots

## Exercise 1

*Exercise 1* requires that we create a markdown document and import the library ggplot2 like so:

library(ggplot2)

This imports the plotting library to allow the data to be presented.

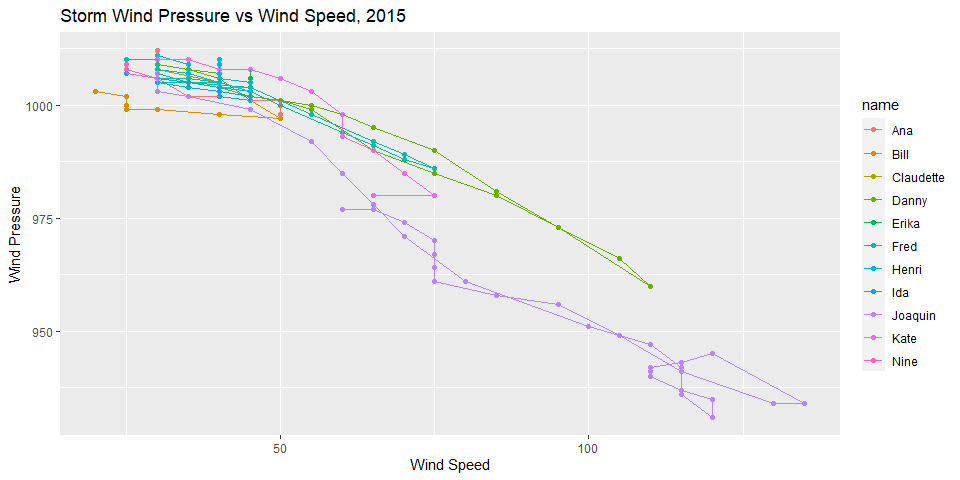
## Exercise 2

In *Exercise 2*, storm data from dplyr is imported to practice with the tools available from ggplot2. The storms data will be called and a particular year will be filtered out. Then the data for the **wind** and **pressure** will be displayed on the x and y axis appropriately. The plot will be shown as a scatter plot with lines joining to show the path in which the pressure vs wind took over time.

The chunk of code from the exercise provided to run is:

library(dplyr)  
  
storms = filter(storms, year == 2015)  
p <- ggplot(storms, aes(x = wind, y = pressure, colour = name)) +  
 geom\_point() +  
 geom\_path()  
p + ggtitle("Storm Wind Pressure vs Wind Speed, 2015") +   
 xlab("Wind Speed") + ylab("Wind Pressure")

The *storms* data-set is imported with dplyr, and filtering the data for the year of our lord 2015 provides us with the data-set to plot. This clarifies data to fewer storms from the vast amount of data available, and allows this plot to be produced.



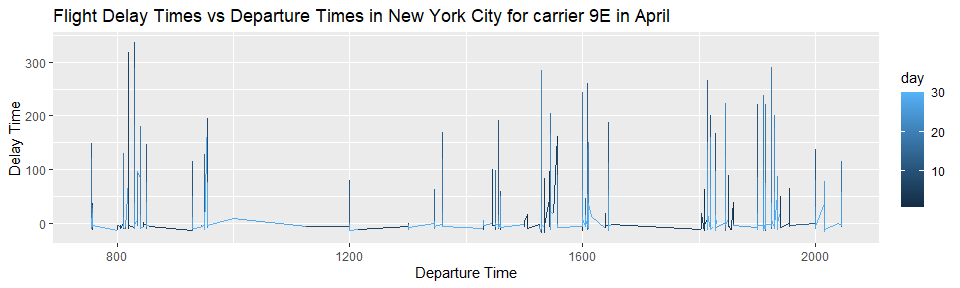
Wind pressure with respect to speed for each storm during 2015

## Exercise 3

In this exercise, the data-set for *New York City Flights* is requested from the nycflights13. A suitable data-set compiled was *Departure Delay* versus *Scheduled Departure Time*, to determine at what time delays most frequently occurred.

library(nycflights13)  
  
p <- ggplot(filter(flights, month==4 & carrier=="9E"), aes(x=sched\_dep\_time, y=dep\_delay, colour=day)) +   
 geom\_line()  
p + ggtitle("Flight Delay Times vs Departure Times for carrier 9E in New York City") +   
 xlab("Departure Time") + ylab("Delay Time")

Which produced the following plot:

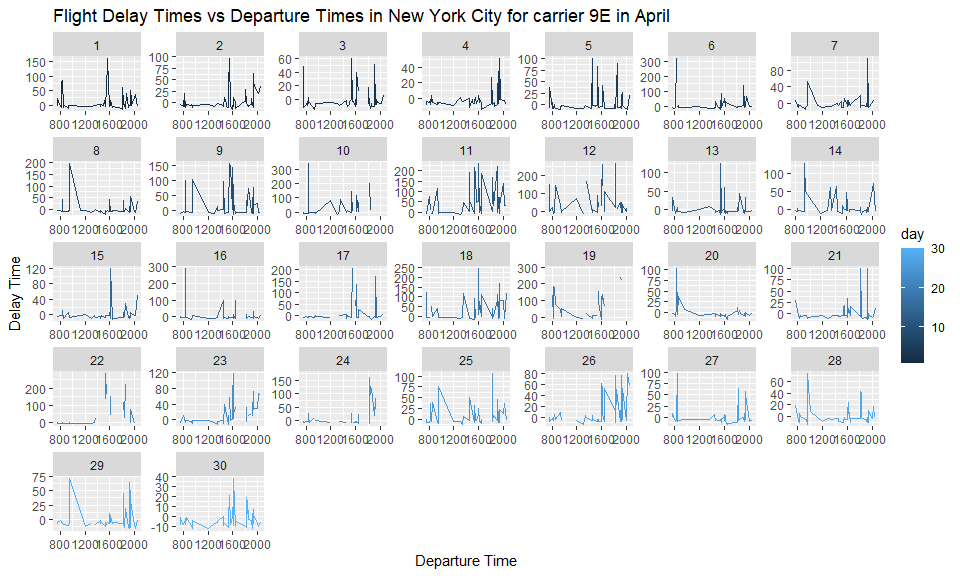


Flight for carrier 9E’s departure delays with respect to departure time and day in the month of April

## Exercise 4

*Exercise 4* requests an example of the use of the facet\_wrap() function in a chart. The chart from *Exercise 3* would be a prime example to separate the days display when delays occur and at what times. The following code was written:

p <- ggplot(filter(flights, month==4 & carrier=="9E"), aes(x=sched\_dep\_time, y=dep\_delay, colour=day)) +   
 geom\_line() +  
 facet\_wrap(vars(day), scales = "free", ncol=7)   
p + ggtitle("Flight Delay Times vs Departure Times in New York City for carrier 9E in April") +   
 xlab("Departure Time") + ylab("Delay Time")



Flight for carrier 9E’s departure delays with respect to departure time and day in the month of April