

5. Data Visualization

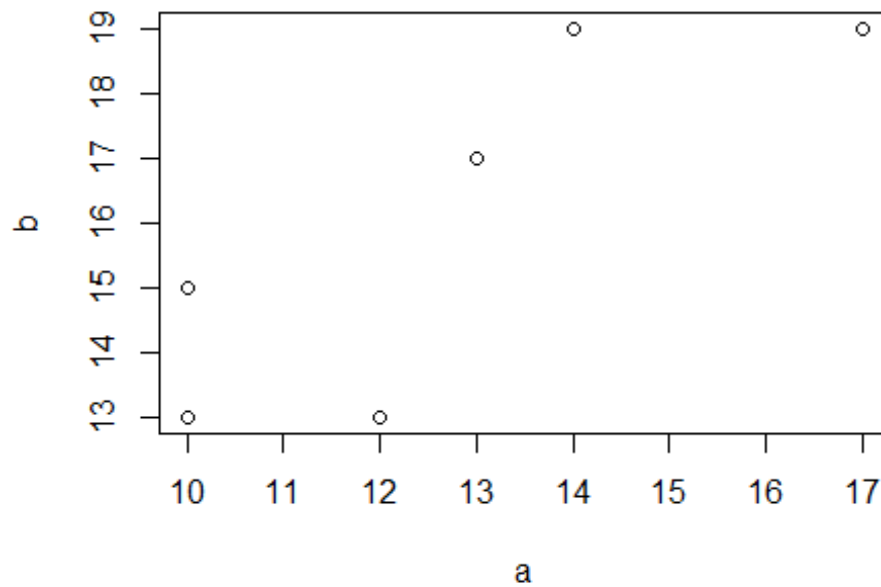
5.1 Plots Introduction

We will discuss three main types of plots in this section.

- Scatter Plot
- Histogram
- Box Plot (also called Box-and-Whisker Plot)

The first plot

```
a <- c(10,13,14,10,12,17)
b <- c(15,17,19,13,13,19)
plot(a,b)
```



Scatter Plot and customization

Download **nycflights13** package and plot a scatter chart with x axis as arrival delay and y axis as departure delay.

```
# installed.packages()
# install.packages("nycflights13")
```

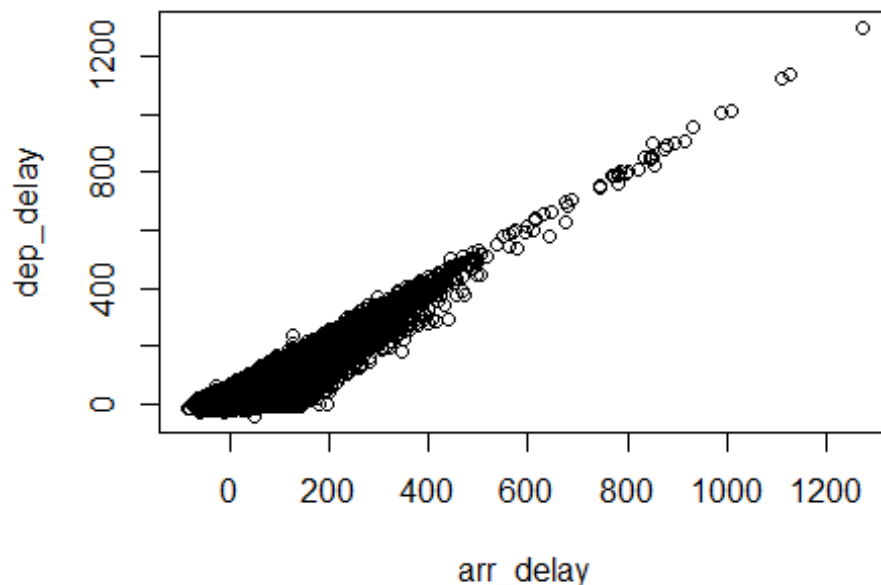
```

library(nycflights13)
nycflights13::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 <dtm>

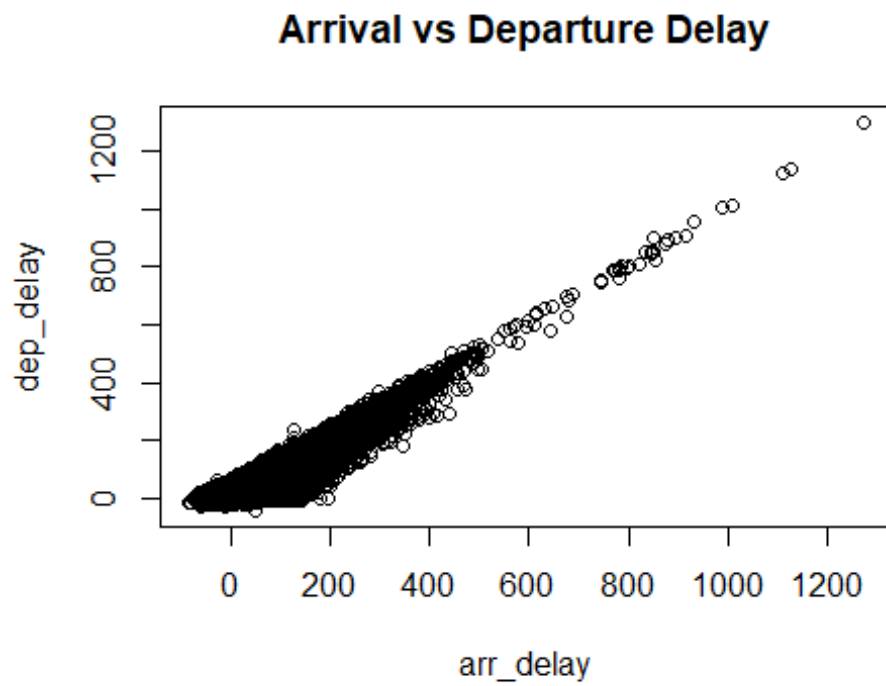
attach(nycflights13::flights)
plot(arr_delay, dep_delay)

```



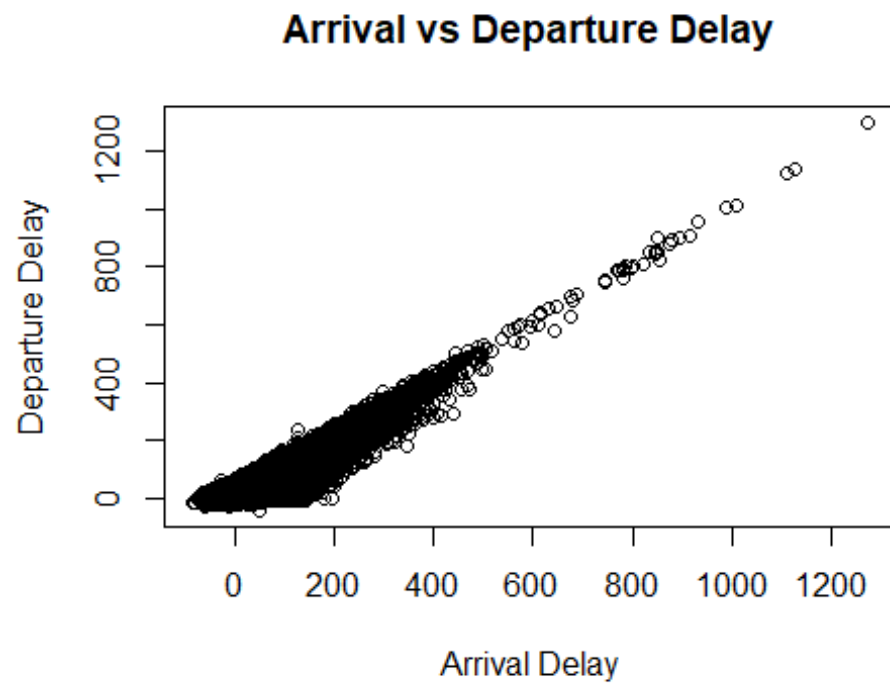
Adding header to the plot

```
plot(arr_delay, dep_delay, main="Arrival vs Departure Delay")
```



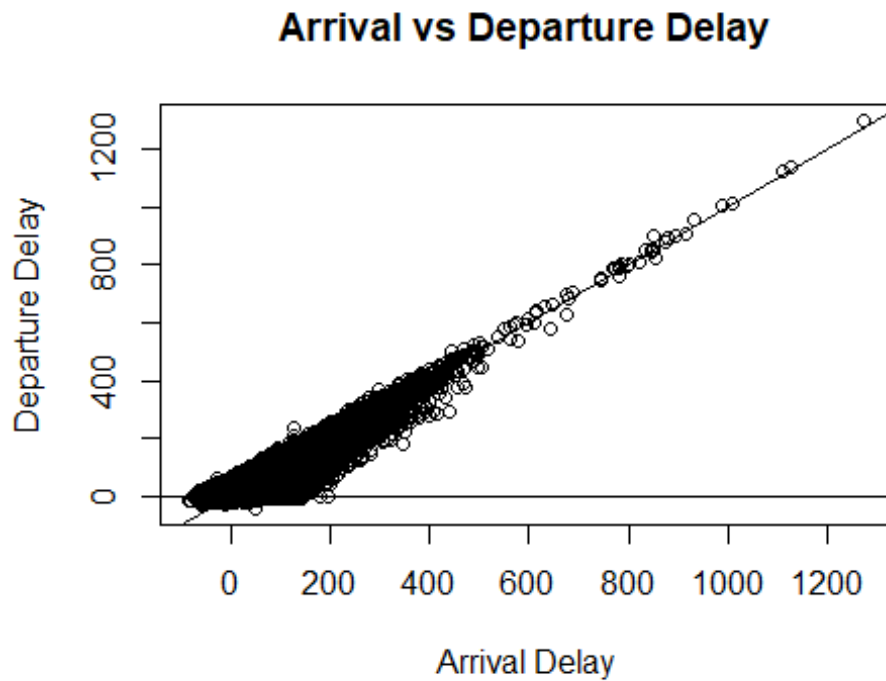
Adding x and y axis labels

```
plot(arr_delay,dep_delay, main="Arrival vs Departure Delay",  
      xlab="Arrival Delay", ylab="Departure Delay")
```



Plotting a line on top of the plot

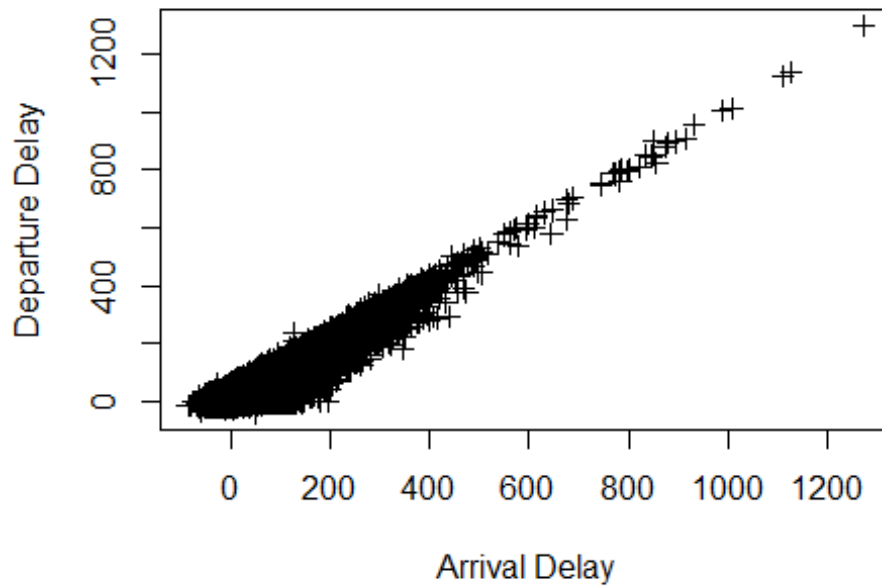
```
plot(arr_delay, dep_delay, main="Arrival vs Departure Delay",  
      xlab="Arrival Delay", ylab="Departure Delay")  
abline(0,1)  
abline(0,0)
```



Changing Plot Character (pch=) 1. Circle 2. Triangle 3. Plus 4. Cross 5. Diamond, 6. Reverse triangle 7. Box and crossed

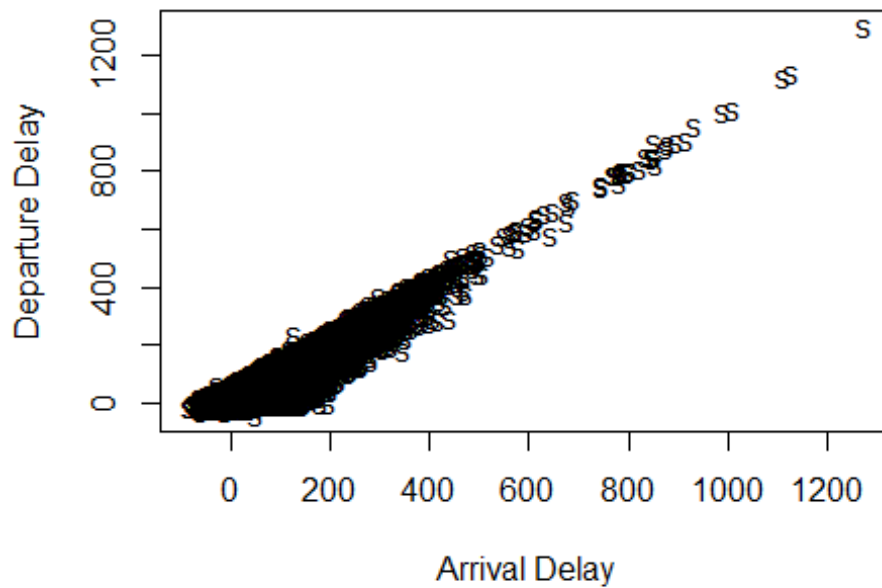
```
plot(arr_delay, dep_delay, main="Arrival vs Departure Delay",  
      xlab="Arrival Delay", ylab="Departure Delay", pch=3)
```

Arrival vs Departure Delay



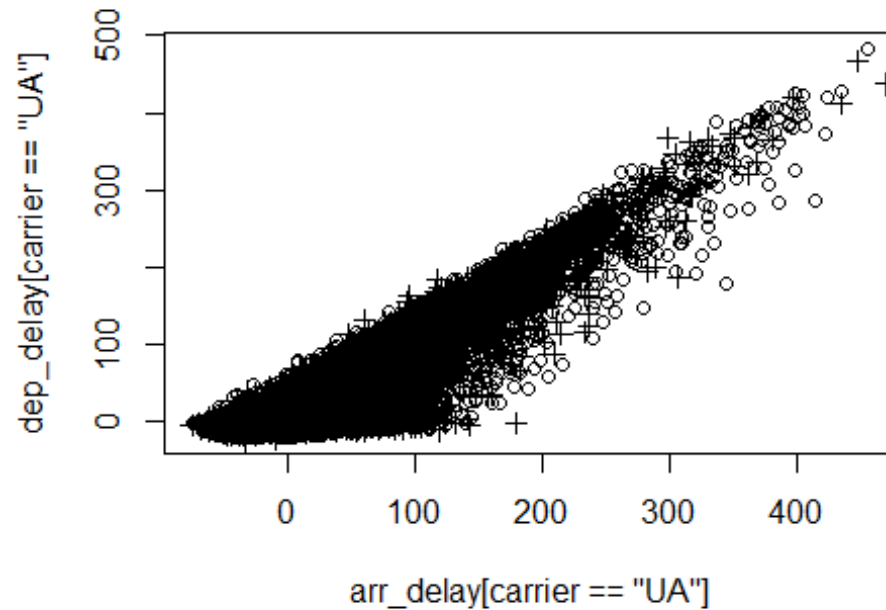
```
plot(arr_delay, dep_delay, main="Arrival vs Departure Delay",  
      xlab="Arrival Delay", ylab="Departure Delay", pch="s")
```

Arrival vs Departure Delay

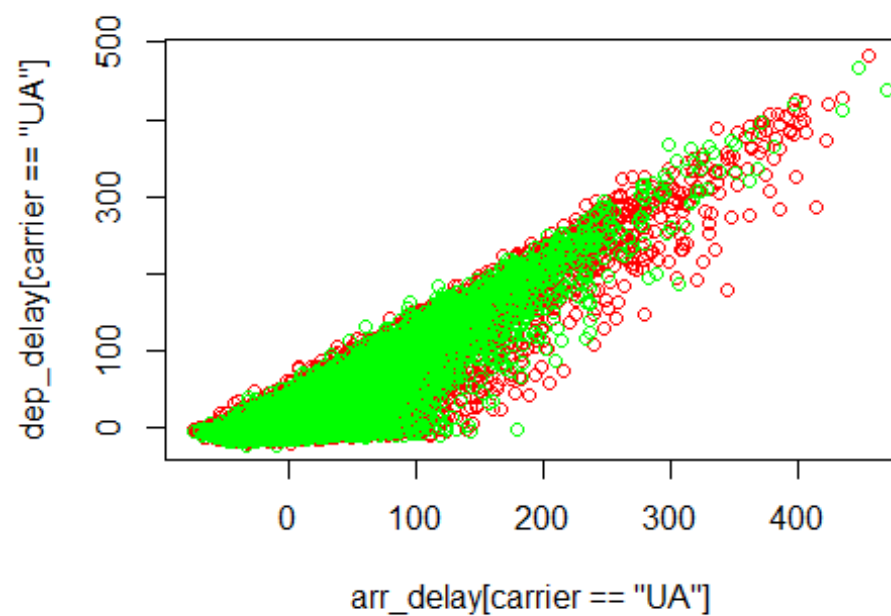


Filtering data

```
plot(arr_delay[carrier=="UA"],dep_delay[carrier=="UA"])  
points(arr_delay[carrier=="AA"],dep_delay[carrier=="AA"], pch=3)
```

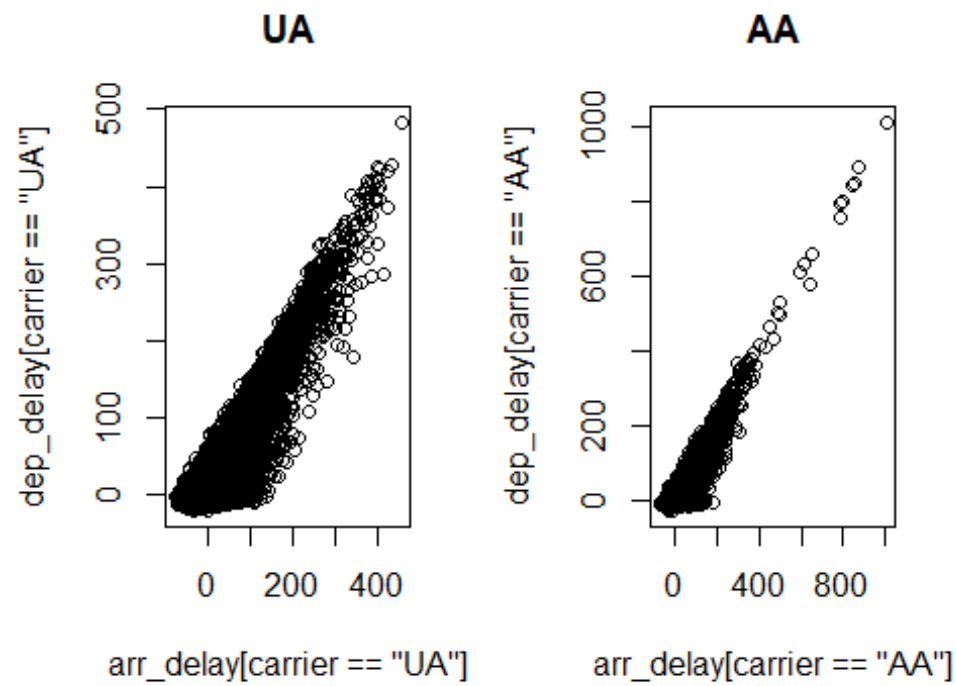


```
plot(arr_delay[carrier=="UA"],dep_delay[carrier=="UA"], col="Red")  
points(arr_delay[carrier=="AA"],dep_delay[carrier=="AA"], col="green")
```



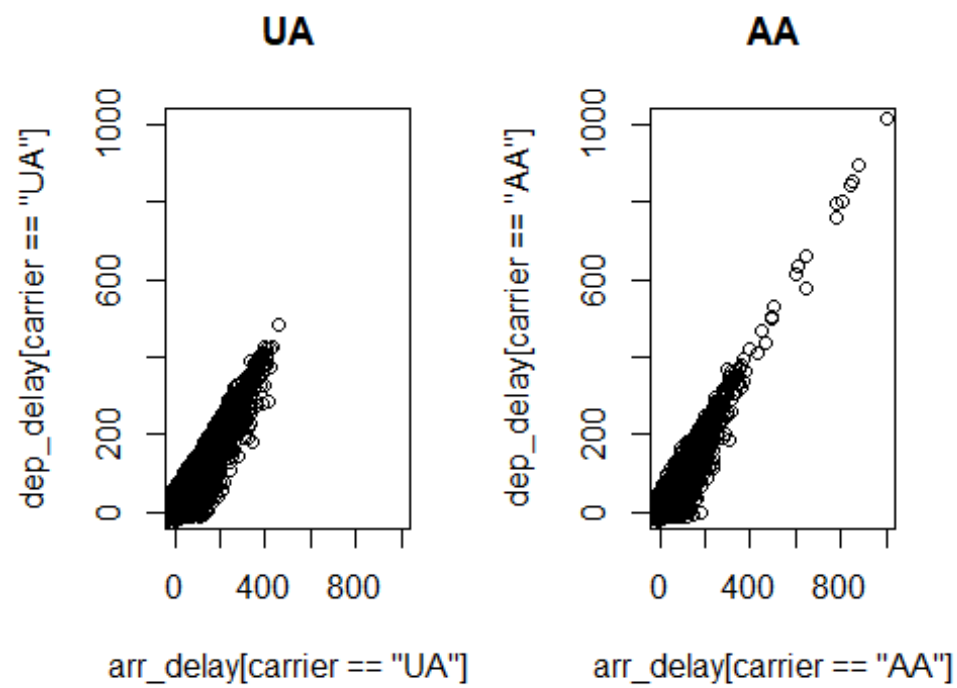
Partition - Multi Frame Rows

```
par(mfrow=c(1,2))  
plot(arr_delay[carrier=="UA"],dep_delay[carrier=="UA"], main="UA")  
plot(arr_delay[carrier=="AA"],dep_delay[carrier=="AA"], main="AA")
```

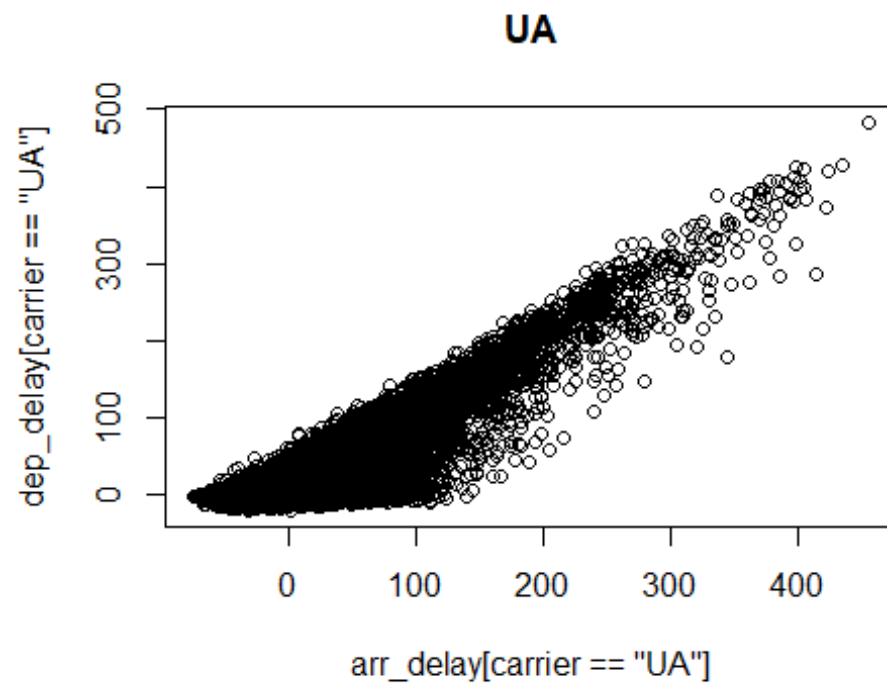
Equal scale for x and y axis.

```
par(mfrow=c(1,2))
plot(arr_delay[carrier=="UA"],dep_delay[carrier=="UA"], main="UA",
      xlim=c(0,1000), ylim=c(0,1000))
plot(arr_delay[carrier=="AA"],dep_delay[carrier=="AA"], main="AA",
      xlim=c(0,1000), ylim=c(0,1000))
```



Restoring the frame for a single plot

```
par(mfrow=c(1,1))  
plot(arr_delay[carrier=="UA"],dep_delay[carrier=="UA"], main="UA")
```



Adding text and lines

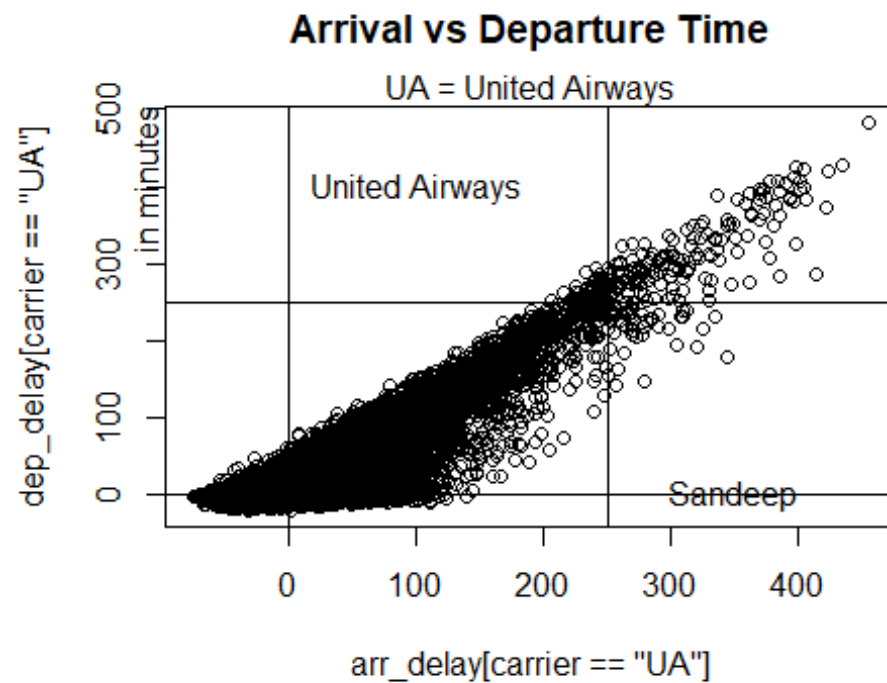
```
plot(arr_delay[carrier=="UA"],dep_delay[carrier=="UA"],
      main="Arrival vs Departure Time")

text(x=100, y=400, label="United Airways")
text(x=400, y=0, adj=1, label="Sandeep")

mtext(text="UA = United Airways", side=3)
mtext(text="in minutes", side=2, adj=1)

abline(h=250)
abline(v=250)

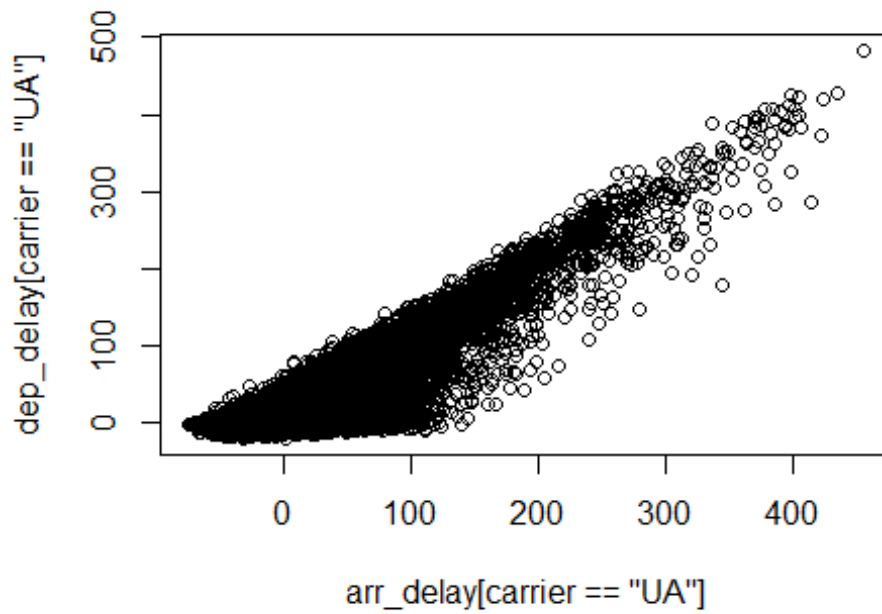
abline(h=0)
abline(v=0)
```



Changing the size and color of points using 'cex' and 'col' argument

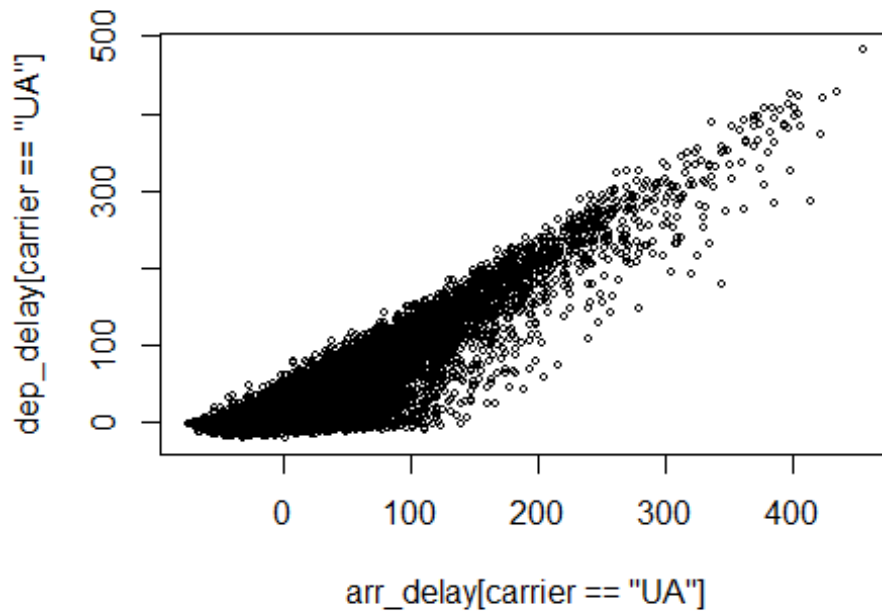
```
# Normal size point  
plot(arr_delay[carrier=="UA"], dep_delay[carrier=="UA"],  
      main="Arrival vs Departure Time")
```

Arrival vs Departure Time



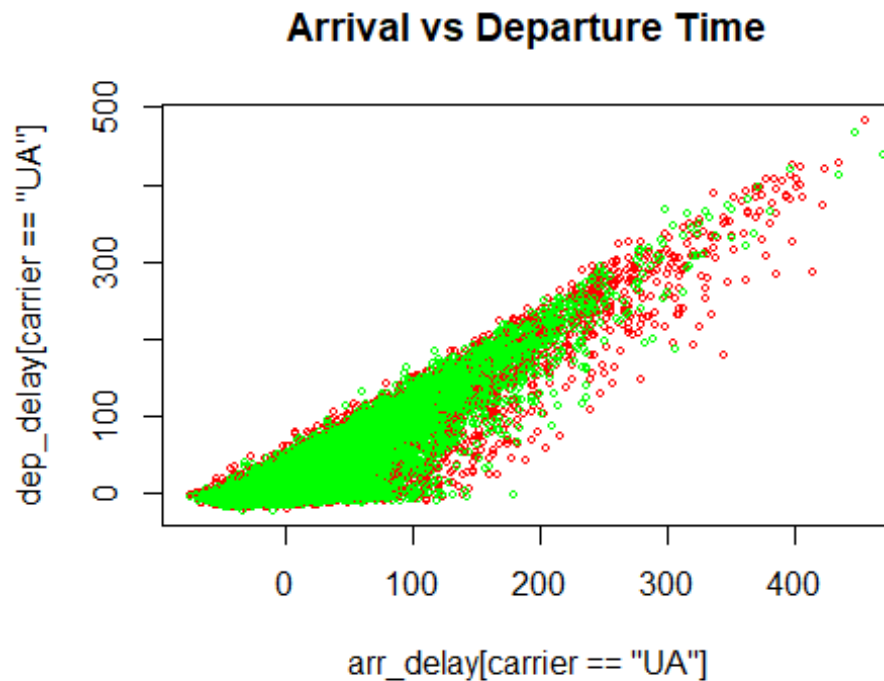
```
# Half size point  
plot(arr_delay[carrier=="UA"],dep_delay[carrier=="UA"],  
      main="Arrival vs Departure Time", cex=0.5)
```

Arrival vs Departure Time



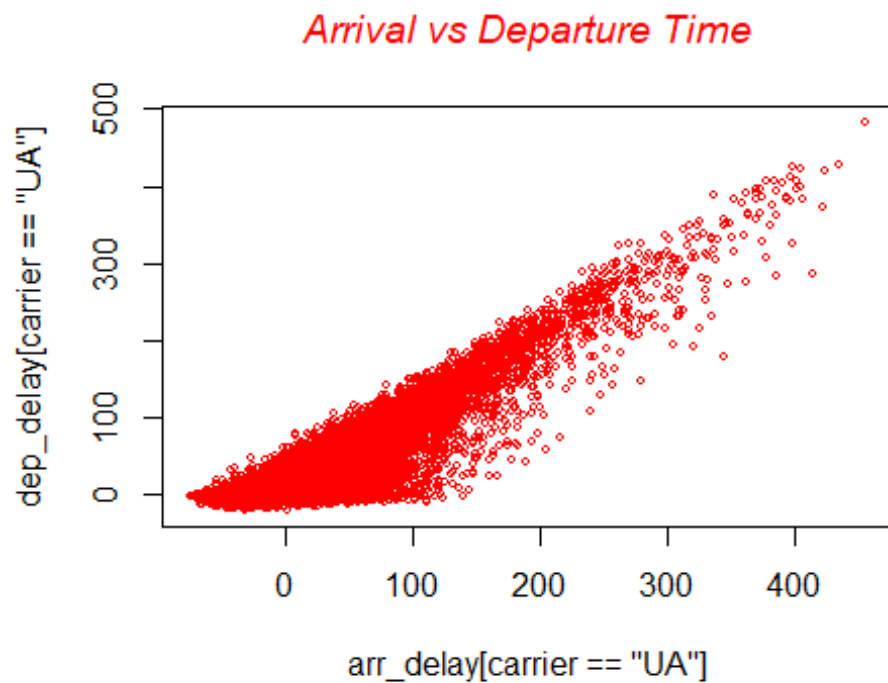
```
# Color to red
plot(arr_delay[carrier=="UA"],dep_delay[carrier=="UA"],
     main="Arrival vs Departure Time", cex=0.5, col="red")

# add green points on top of the UA plot for AA
points(arr_delay[carrier=="AA"],dep_delay[carrier=="AA"], cex=0.5,
       col="green" )
```



Change the color and font of labels

```
plot(arr_delay[carrier=="UA"],dep_delay[carrier=="UA"],
     main="Arrival vs Departure Time", cex=0.5, col="red",
     col.main="red", font.main=3)
```

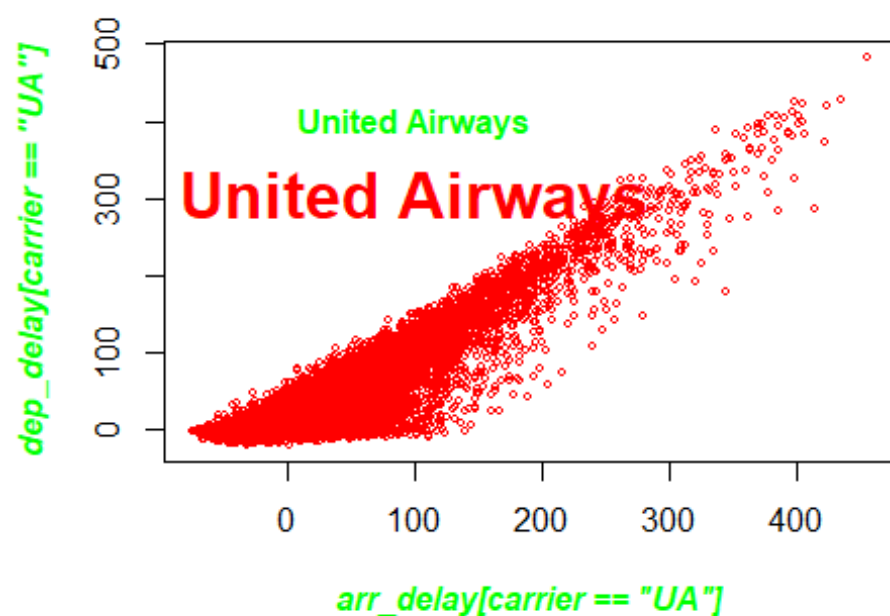


Change the font and color of text as well

```
plot(arr_delay[carrier=="UA"],dep_delay[carrier=="UA"],
      main="Arrival vs Departure Time", cex=0.5, col="red",
      col.main="red", font.main=3, col.lab="green", font.lab=4)
text(100,400, "United Airways", col="green", font=2)

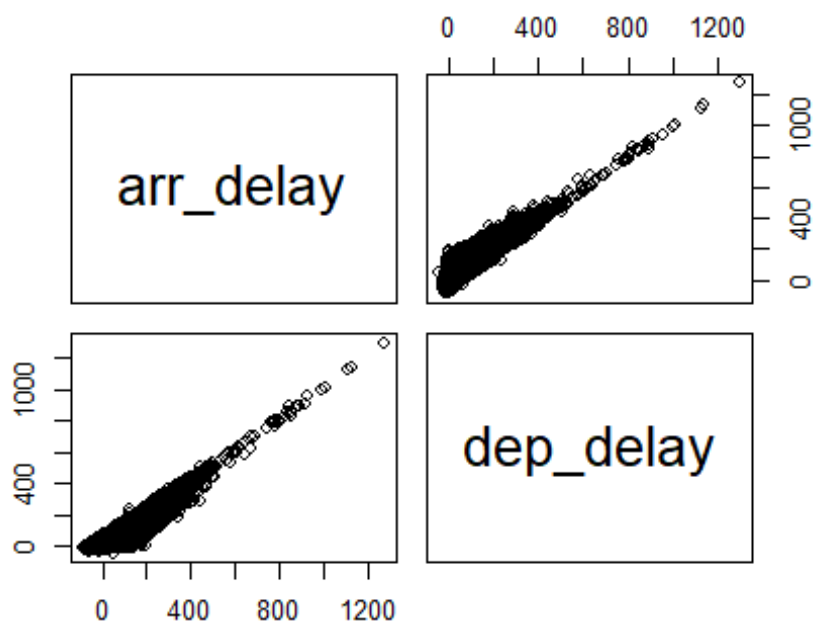
text(100,300, "United Airways", col="red", font=2, cex=2)
```

Arrival vs Departure Time

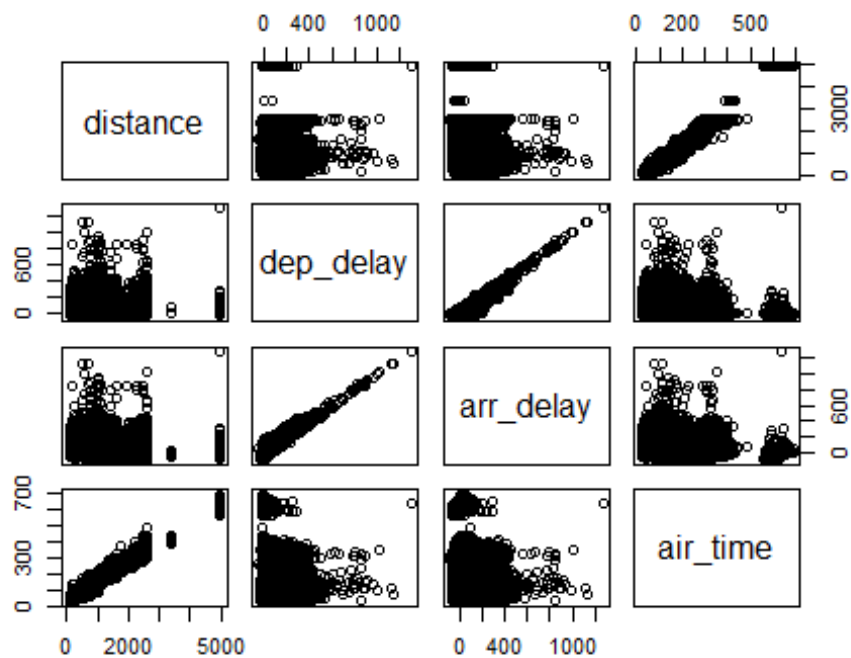


Pairs of Scatter plots

```
pairs(~ arr_delay + dep_delay)
```

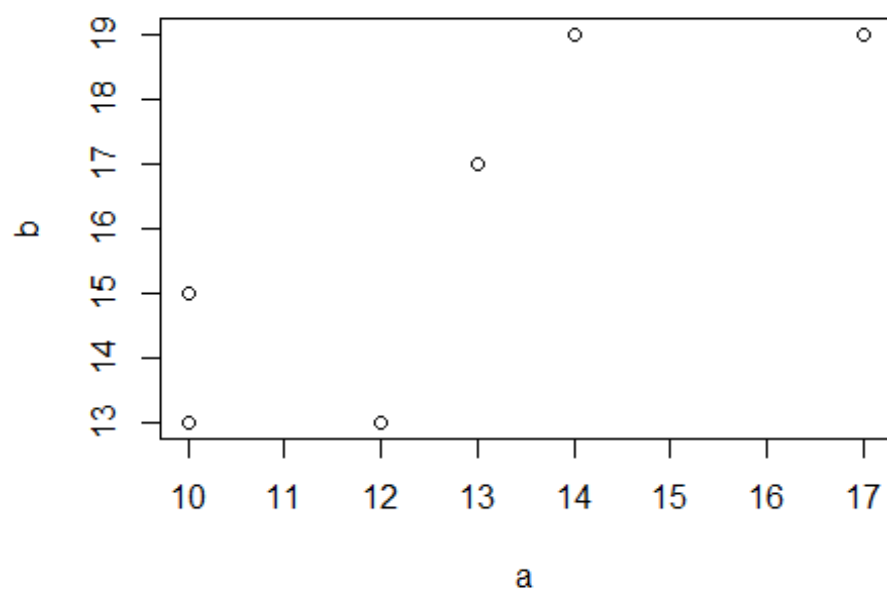



```
pairs(~ distance + dep_delay + arr_delay + air_time)
```

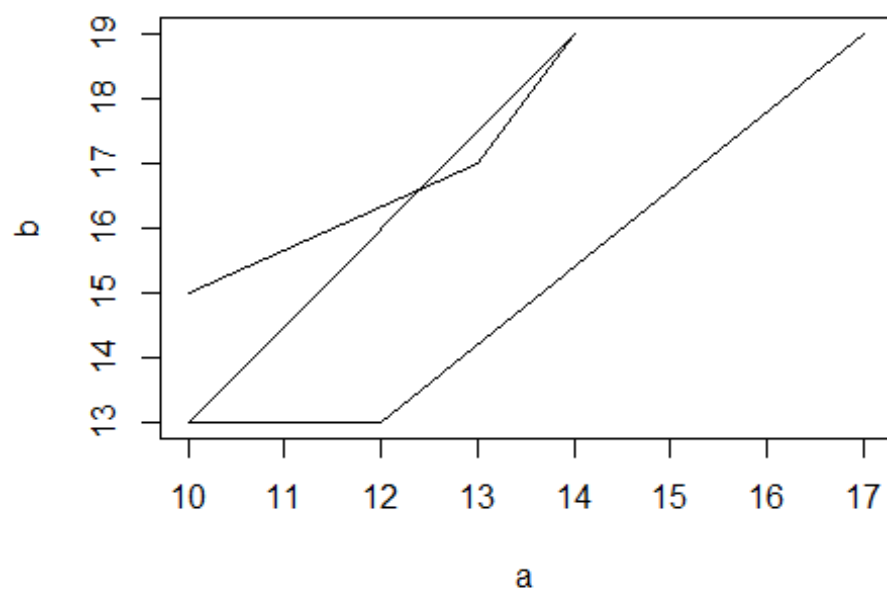


Points or line plot

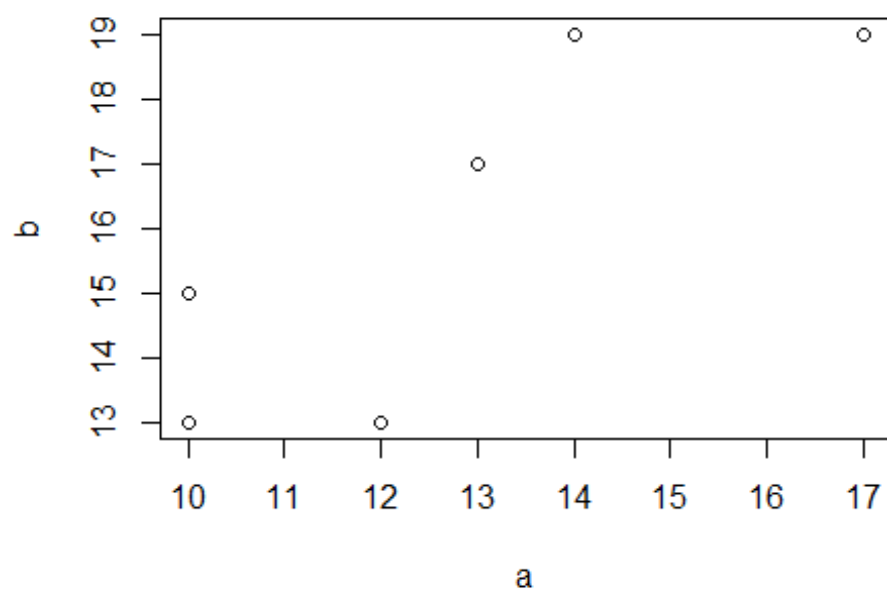
```
a <- c(10,13,14,10,12,17)
b <- c(15,17,19,13,13,19)
plot(a,b)
```



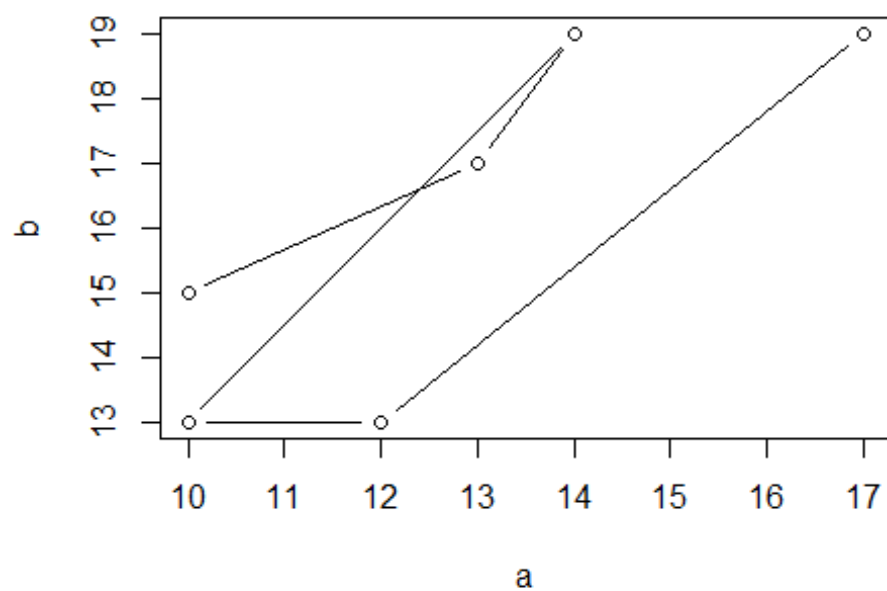
```
plot(a,b,type = "l")
```



```
plot(a,b,type = "p")
```



```
plot(a,b,type = "b")
```

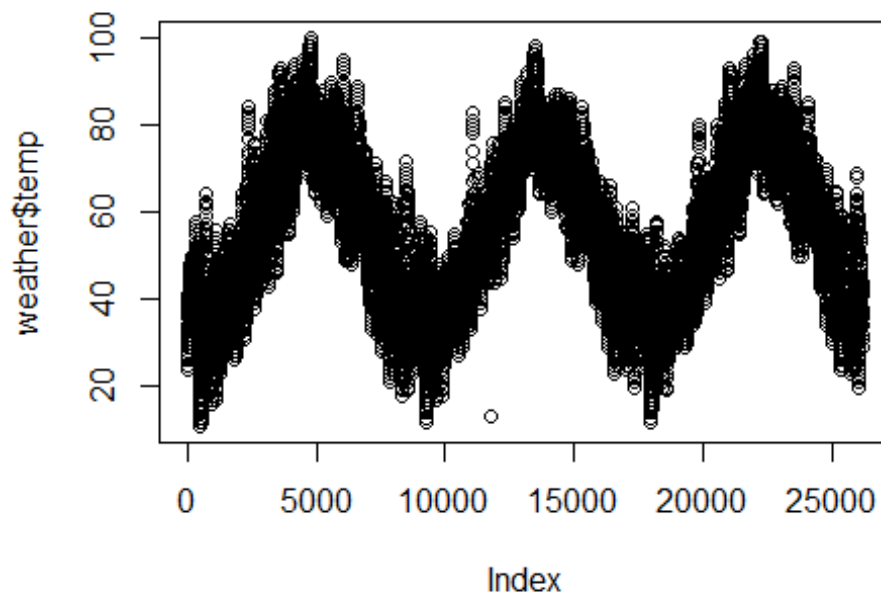


Time series

weather

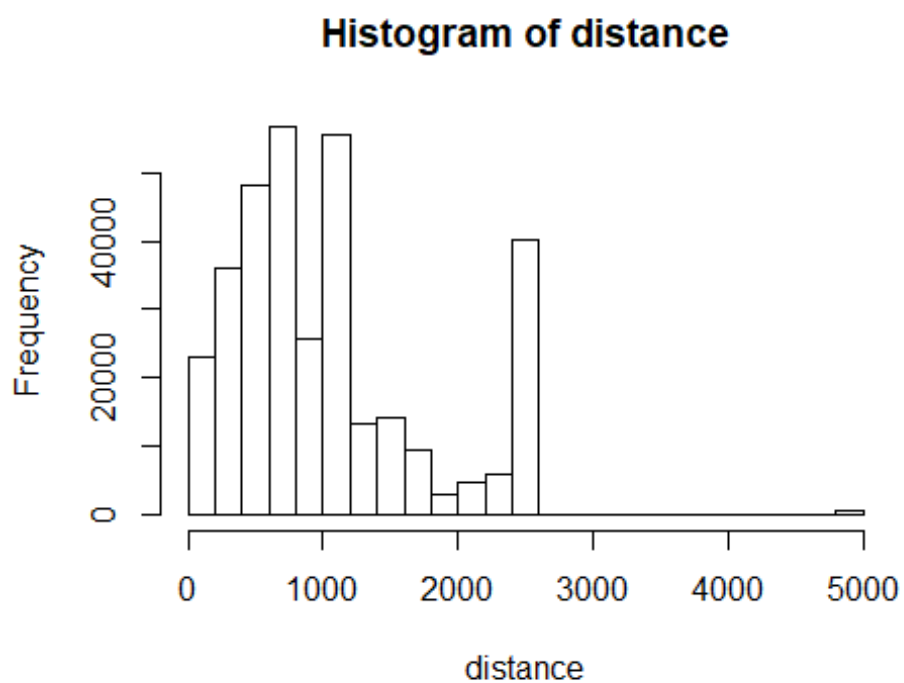
```
## # A tibble: 26,130 x 15
##   origin year month   day hour  temp  dewp humid wind_dir wind_speed
##   <chr>  <dbl> <dbl> <int> <int> <dbl> <dbl> <dbl>   <dbl>     <dbl>
## 1    EWR  2013     1     1     0 37.04 21.92 53.97    230    10.35702
## 2    EWR  2013     1     1     1 37.04 21.92 53.97    230    13.80936
## 3    EWR  2013     1     1     2 37.94 21.92 52.09    230    12.65858
## 4    EWR  2013     1     1     3 37.94 23.00 54.51    230    13.80936
## 5    EWR  2013     1     1     4 37.94 24.08 57.04    240    14.96014
## 6    EWR  2013     1     1     6 39.02 26.06 59.37    270    10.35702
## 7    EWR  2013     1     1     7 39.02 26.96 61.63    250     8.05546
## 8    EWR  2013     1     1     8 39.02 28.04 64.43    240    11.50780
## 9    EWR  2013     1     1     9 39.92 28.04 62.21    250    12.65858
## 10   EWR  2013     1     1    10 39.02 28.04 64.43    260    12.65858
## # ... with 26,120 more rows, and 5 more variables: wind_gust <dbl>,
## #   precip <dbl>, pressure <dbl>, visib <dbl>, time_hour <dtm>
```

```
plot(weather$temp)
```

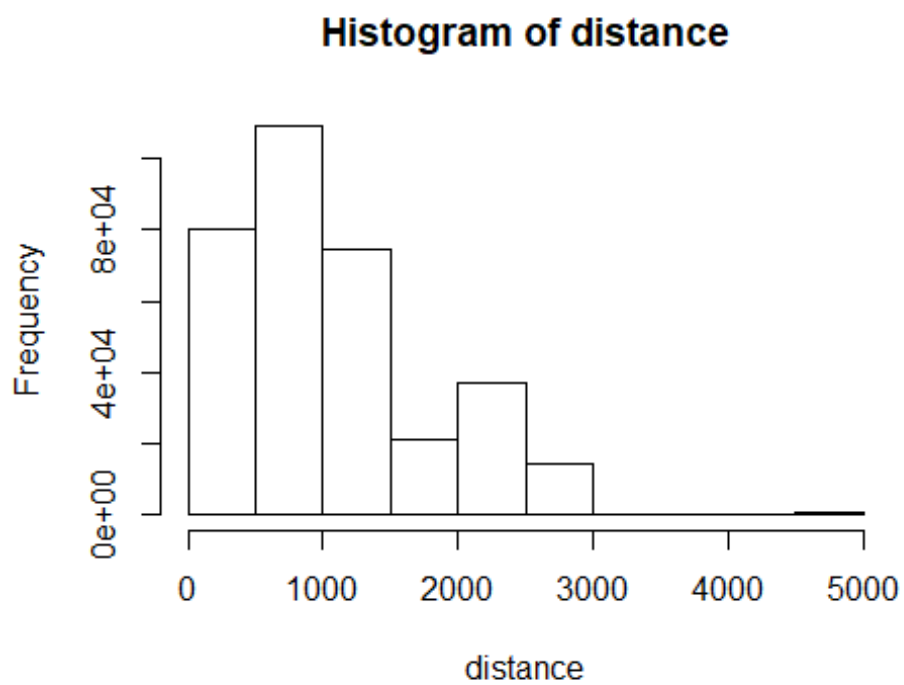


Histogram

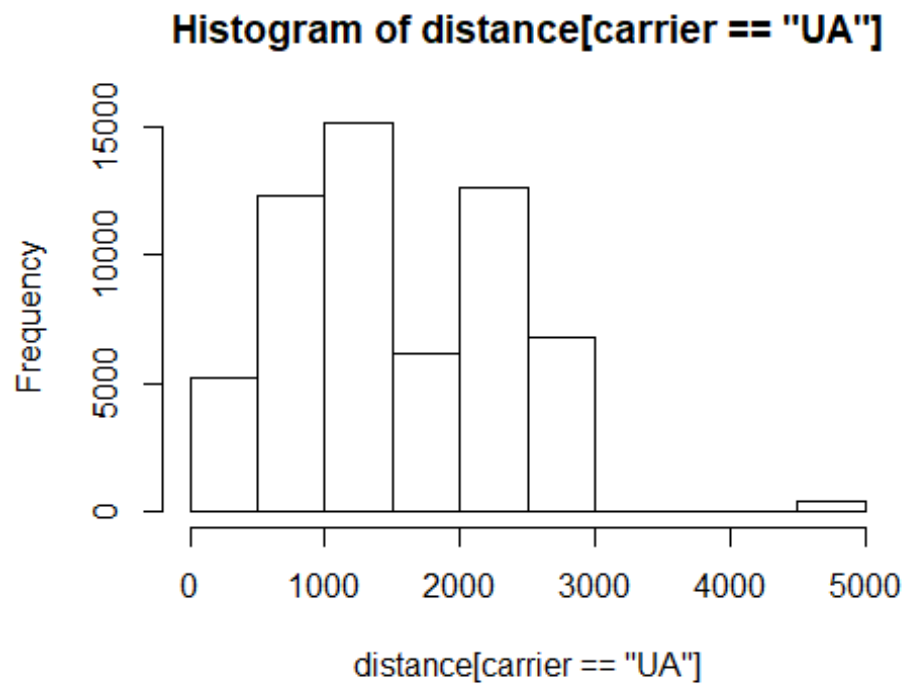
```
hist(distance)
```



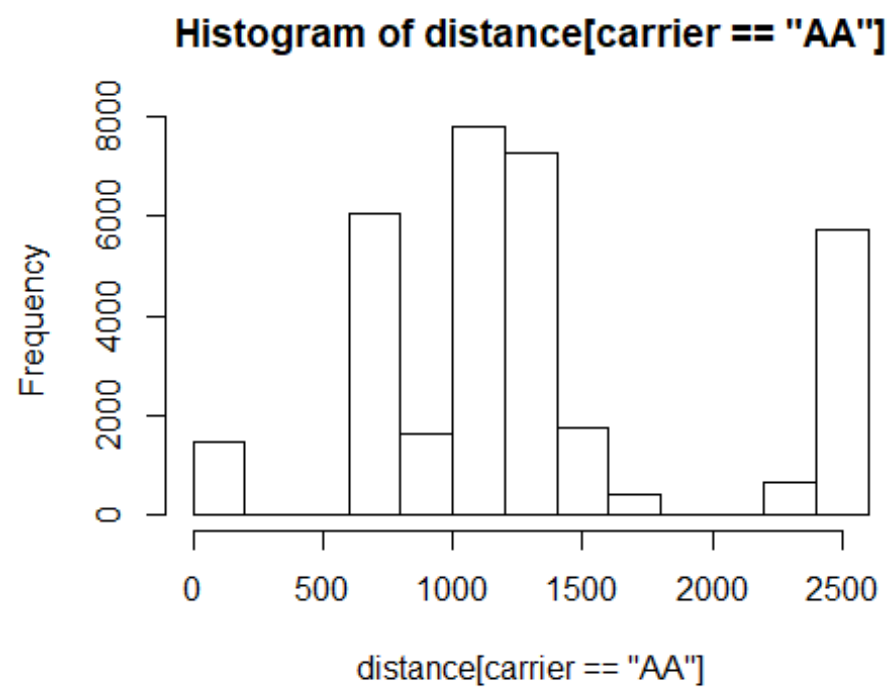
```
hist(distance, breaks = 10)
```



```
# Filter by airline  
hist(distance[carrier == "UA"])
```

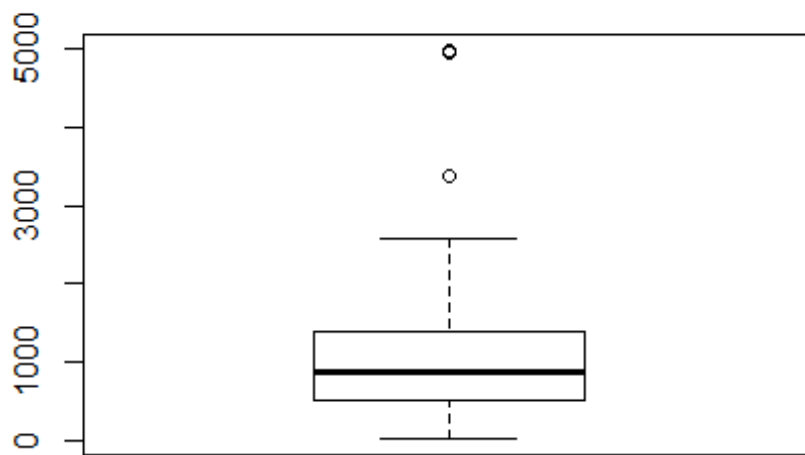


```
hist(distance[carrier == "AA"])
```

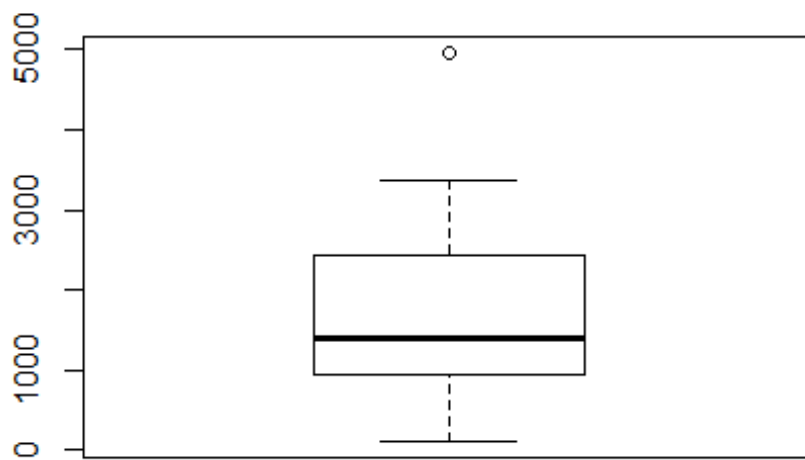


Box and Whisker Plot

```
boxplot(distance)
```

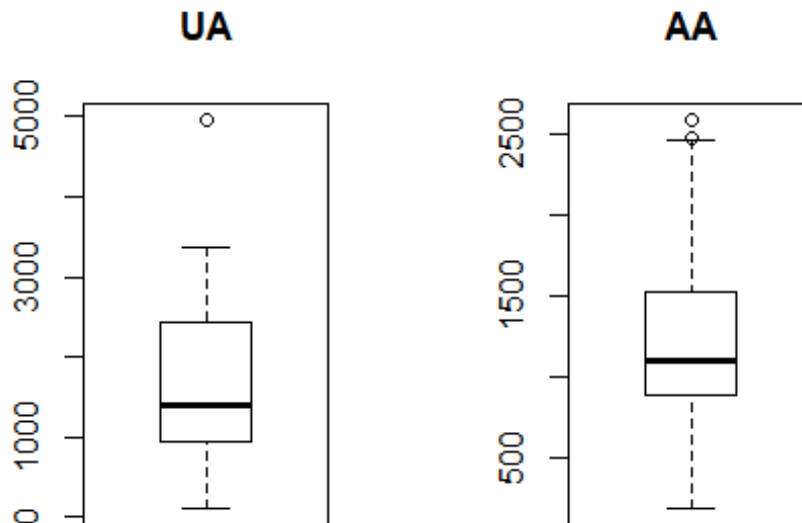


```
#Filter by airline  
boxplot(distance[carrier == "UA"])
```



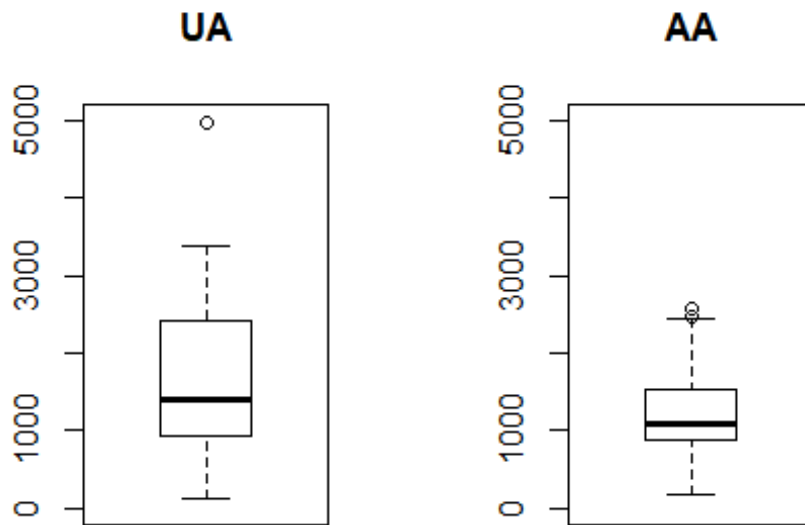
Two box plots side by side

```
par(mfrow=c(1,2))  
boxplot(distance[carrier == "UA"], main="UA")  
boxplot(distance[carrier == "AA"], main="AA")
```



Two box plots side by side with equal scale

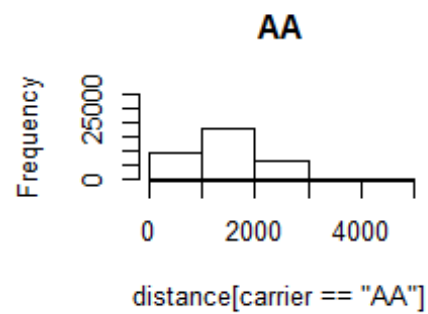
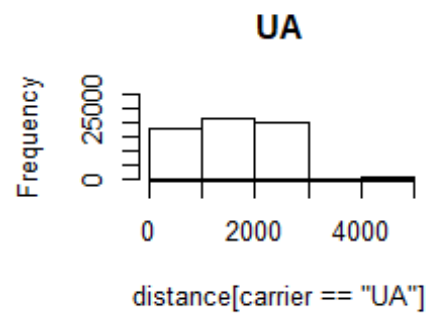
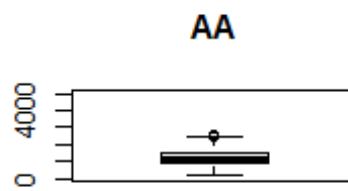
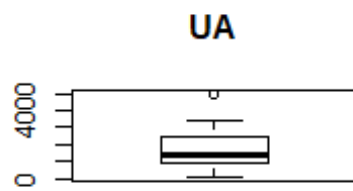
```
par(mfrow=c(1,2))  
boxplot(distance[carrier == "UA"], main="UA", ylim = c(0,5000))  
boxplot(distance[carrier == "AA"], main="AA", ylim = c(0,5000))
```



Two histograms and two box plots

```
par(mfrow=c(2,2))
boxplot(distance[carrier == "UA"], main="UA", ylim = c(0,5000))
boxplot(distance[carrier == "AA"], main="AA", ylim = c(0,5000))

hist(distance[carrier == "UA"], main="UA",
      breaks = c(0, 1000, 2000, 3000, 4000, 5000),
      xlim = c(0,5000), ylim = c(0,30000))
hist(distance[carrier == "AA"], main="AA",
      breaks = c(0, 1000, 2000, 3000, 4000, 5000),
      xlim = c(0,5000), ylim = c(0,30000))
```



Box plots of distance by carrier

```
par(mfrow=c(1,1))
boxplot(distance ~ carrier)
```

