Par

Week 1

Tomás

5 de diciembre de 2019

# Principles of Analytic Graphics

Principios:

* 1: Muestra comparaciones para poder entender mejor los datos
* 2: Demuestra causalidad, explicaciones, estructuras sistematicas, que nos permite identificar el por que de estas diferencias
* 3: Busca muchas variables o datos, esto permite asemejar las comparaciones al mundo real
* 4: Integra evidencia, ocupa muchas formas de mostrar la evidencia para que tu analisis sea lo mas rico posible. Numeros, imagenes, diagramas, graficos.
* 5: Describe y documenta tu evidencia
* 6: El contenido es lo importante, un analisis que no tiene un contenido solido e interesante solamente va a ser una perdida de tiempo

# Exploratory Graphs (part 1)

Por que usamos graficos?

* Entender las propiedades de la data
* Encontrar patrones
* Sugerir modelos
* Realizar analsiis
* Comunicar resultados

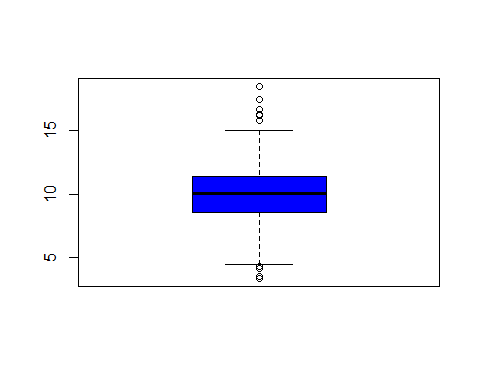
pollution <- read.csv("avgpm25.csv", colClasses = c("numeric","character","factor","numeric","numeric"))  
head(pollution)

## pm25 fips region longitude latitude  
## 1 9.771185 01003 east -87.74826 30.59278  
## 2 9.993817 01027 east -85.84286 33.26581  
## 3 10.688618 01033 east -87.72596 34.73148  
## 4 11.337424 01049 east -85.79892 34.45913  
## 5 12.119764 01055 east -86.03212 34.01860  
## 6 10.827805 01069 east -85.35039 31.18973

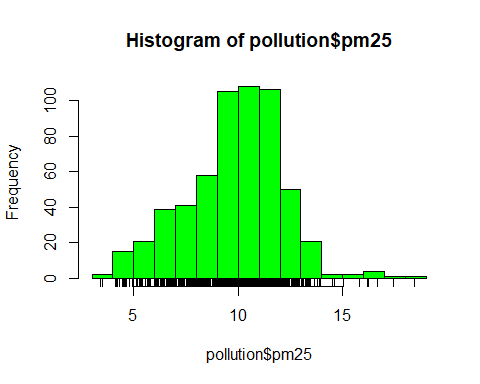
summary(pollution$pm25)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 3.383 8.549 10.047 9.836 11.356 18.441

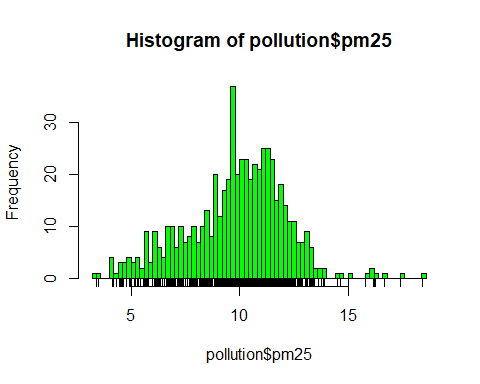
boxplot(pollution$pm25, col = "blue") # Boxplot



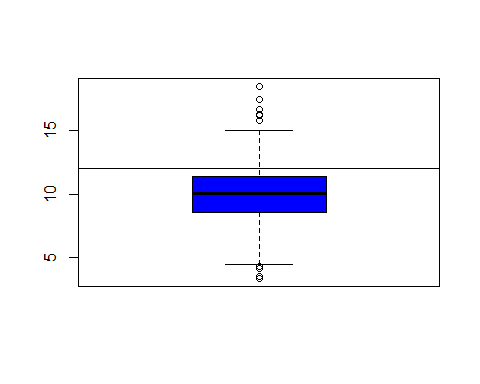
hist(pollution$pm25, col = "green") # Histograma  
rug(pollution$pm25) #Entrega una "Alfombra"" que situa las observaciones en el grafico



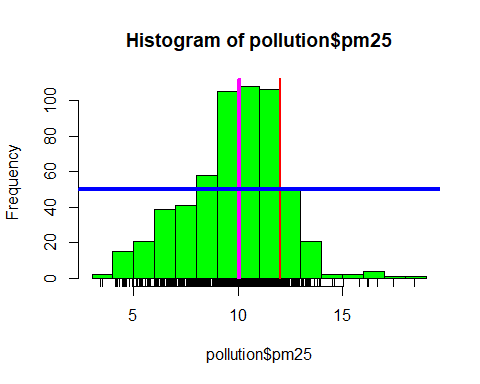
hist(pollution$pm25, col = "green", breaks = 100) # Breaks permite entregar la cantidad de columnas que quiero  
rug(pollution$pm25)



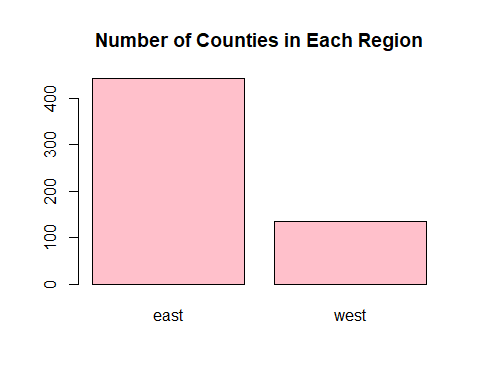
boxplot(pollution$pm25, col = "blue") # Boxplot  
abline(h = 12) # Crea una linea horizontal en el numero que le indico (En el cual me interesa)



hist(pollution$pm25, col = "green")  
abline(v = 12, lwd = 2, col = "red") #Crea una linea pero ahora vertical, v es la posicion, lwd el ancho o grosor  
abline(v = median(pollution$pm25), col = "magenta", lwd = 4)  
abline(h = 50, lwd = 4, col = "blue") # h= horizontal, v = vertical  
rug(pollution$pm25)

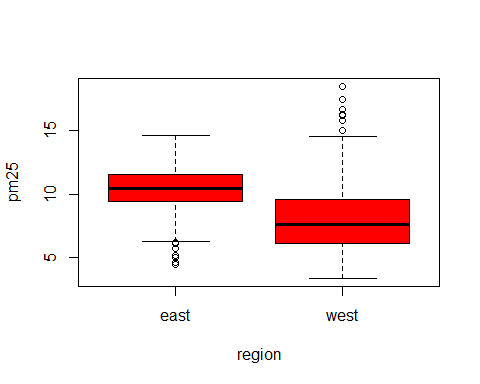


barplot(table(pollution$region), col = "pink", main = "Number of Counties in Each Region") # main permite darle titulo al grafico (En general se utiliza para graficar variables categoricas "factors")

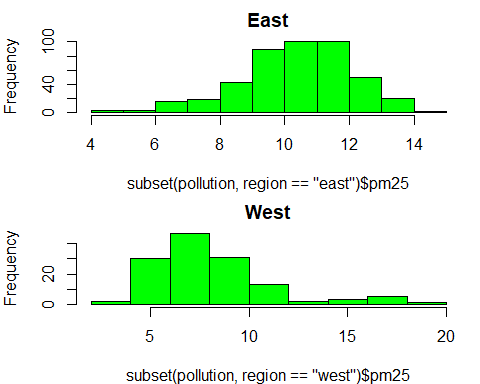


# Exploratory Graphs (part2)

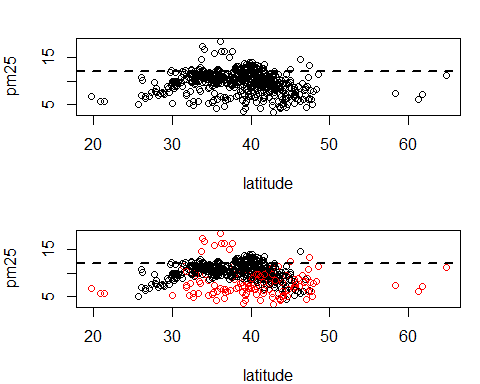
boxplot(pm25 ~ region, data = pollution, col = "red") # data permite utilizar las variables por su nombre y poder utilizar la cola de chancho para graficar mas de una variable a la vez



par(mfrow = c(2,1), mar = c(4,4,2,1)) # Permite graficar en una foto dos graficos, mfrow = Las dimensiones de los graficos en este caso 2 linea con 1 grafico, mar = Me da los margenes de los histogramas para que queden mejor posicionados  
hist(subset(pollution, region == "east")$pm25, col = "green", main = "East") #Creo un histograma solo con las observaciones east  
hist(subset(pollution, region == "west")$pm25, col = "green", main = "West") #Creo un histograma solo con las observaciones west



with(pollution, plot(latitude,pm25)) #Plot es un grafico de puntos, (with permite utilizar los nombres de las variables de pollution y no tener que llamar a pollution$pm25)  
abline(h = 12, lwd = 2, lty = 2) # lty nos permite puntear la linea  
with(pollution, plot(latitude,pm25, col = region)) # col permite diferencia por region  
abline(h = 12, lwd = 2, lty = 2)

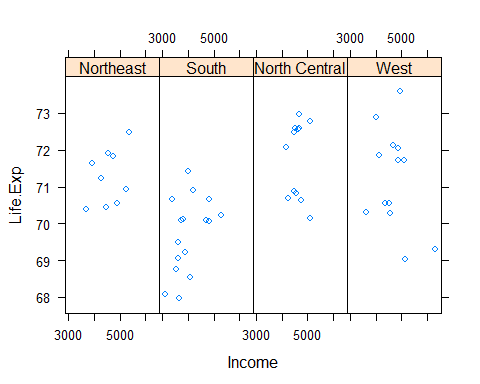


# Plotting Systems in R

#Basic Plot  
library(datasets)  
data(cars)  
with(cars, plot(speed,dist))  
#The Lattice System  
library(lattice)



state <- data.frame(state.x77, region = state.region)  
xyplot(Life.Exp ~ Income | region, data = state, layout = c(4,1))



## ggplot2 System  
library(ggplot2)

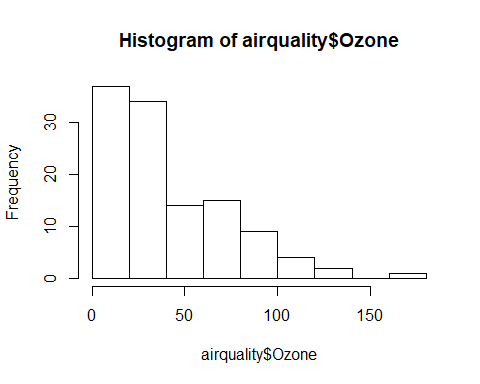
## Warning: package 'ggplot2' was built under R version 3.6.1

data(mpg)  
qplot(displ, hwy, data = mpg)

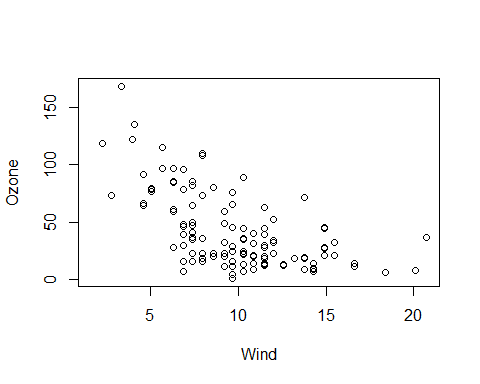


# Base Plotting System (part 1)

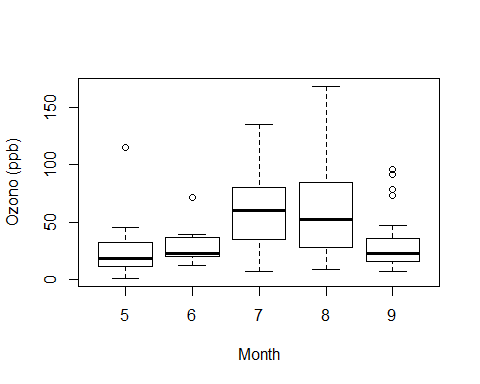
library(datasets)  
hist(airquality$Ozone)



with(airquality, plot(Wind,Ozone))



airquality <- transform(airquality, Month = (factor(Month))) # Convierte los meses en factores  
boxplot(Ozone ~ Month, airquality, xlab = "Month", ylab = "Ozono (ppb)") # Grafica el Ozono en cada mes un boxplot separados



Los graficos pueden tener una infinidad de parametros pero es importante recordar estos:

* pch: Cambia el simbolo que utiliza, default son circulos. Puedo entregarle numeros o caracteres (“H”, “A”, 1)
* lty: El tipo de linea que se utiliza, puede ser punteada etc…
* lwd: Permite controlar el ancho de las lineas que se utilizan
* col: Es el color
* xlab: Nombre de la variable x
* ylan: Nombre de la variable y
* main: Nombre del grafico

La funcion par() permite setear los parametros globales de cualquier grafico, una vez usado sirve para todos los graficos y se puede sobreescribir:

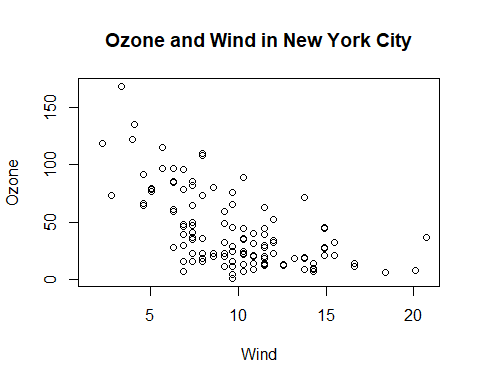
* las: La orientacion de la etiqueta en los ejes
* bg: El color de fondo
* mar: Los margenes
* oma: El margen exterior
* mfrow: Numero de graficos por (fila,columna) (Los graficos se llenan por filas)
* mfcol: Numero de graficos por (fila,columna) (Los graficos se llenan por columnas)

# Base Plotting System (part 2)

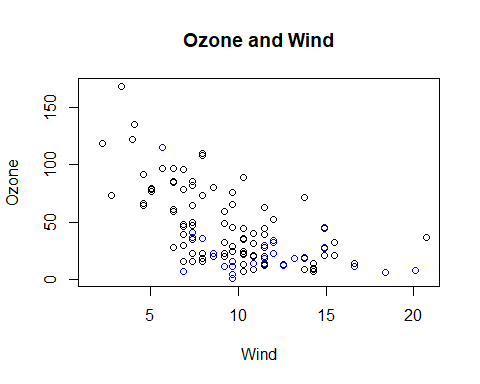
Funciones importantes del Base Plotting:

* plot: Permite realizar un grafico de puntos, sin embargo, dependiendo del tipo de variables puede ser otra cosa
* lines: Conecta los puntos de un grafico de puntos
* points: Agrega puntos a un graficos
* text: Agrega textos de los graficos
* title: Cambia los titulos de los ejes, del grafico, subtitulos
* mtext: Agrega texto a los margenes
* axis: Agrega marcas de ejes o etiquetas

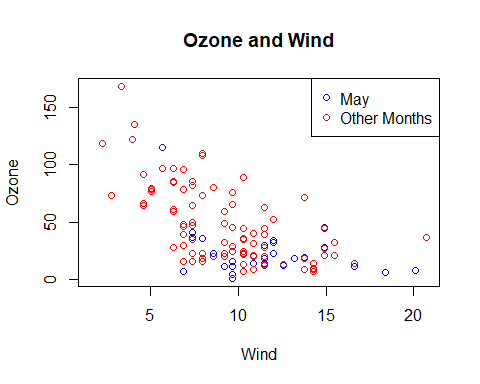
library(datasets)  
with(airquality, plot(Wind,Ozone))  
title(main = "Ozone and Wind in New York City") #Agrega titulo



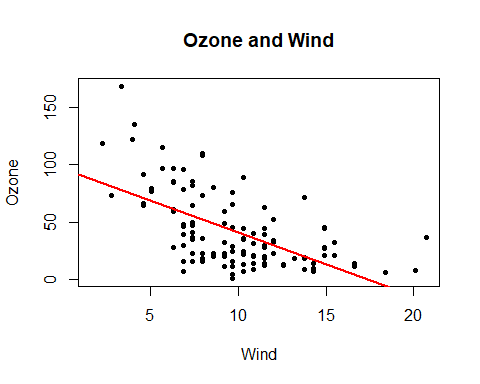
with(airquality, plot(Wind,Ozone, main = "Ozone and Wind")) #Incluye titulo  
with(subset(airquality, Month == 5), points(Wind,Ozone, col = "blue")) #Incluyo los puntos del mes 5 en azul



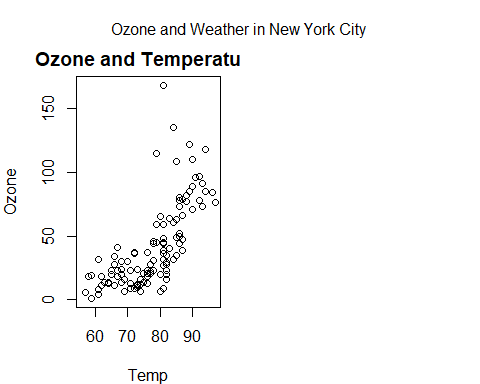
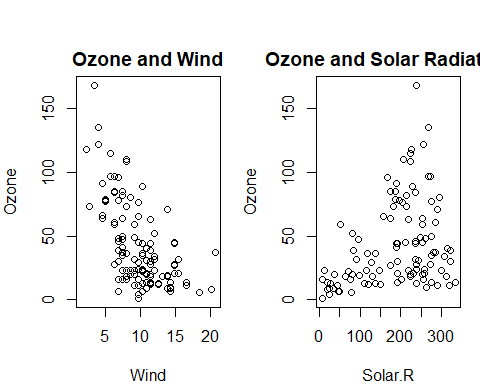
with(airquality, plot(Wind,Ozone, main = "Ozone and Wind", type = "n")) #Ploteo el grafico, type = "n" significa que no agrega nada dentro del grafico para despues crear los puntos por separado  
with(subset(airquality, Month == 5), points(Wind,Ozone,col="blue")) # Agrego los puntos azules al mes 5  
with(subset(airquality, Month != 5), points(Wind,Ozone,col="red")) #Agrego los puntos rojos a los otro meses  
legend("topright", pch = 1, col = c("blue","red"), legend = c("May", "Other Months")) #Creo la leyenda, pch = tipo de simbolo (circulo, tringualo), col = color. legend = titulos de las leyendas



with(airquality, plot(Wind,Ozone, main = "Ozone and Wind", pch =20))  
model <- lm(Ozone ~ Wind, airquality) #Creo un modelo de regresion lineal que se ajuste a los datos  
abline(model, lwd = 2, col = "red") #Agrego una linea que es igual al modelo de color rojo



par(mfrow = c(1,2))  
with(airquality, {  
 plot(Wind,Ozone, main = "Ozone and Wind")  
 plot(Solar.R, Ozone, main = "Ozone and Solar Radiation") #Con la funcion par puedo poner dos o mas graficos  
})  
par(mfrow = c(1,2), mar = c(4,4,2,1), oma = c(0,0,2,0))  
with(airquality, {  
 plot(Wind,Ozone, main = "Ozone and Wind")  
 plot(Solar.R, Ozone, main = "Ozone and Solar Radiation") #Con la funcion par puedo poner dos o mas graficos  
 plot(Temp, Ozone, main = "Ozone and Temperature")  
 mtext("Ozone and Weather in New York City", outer = TRUE) #mtext permite poner un texto general al ploteo, si uso la funcion main es para cada uno de los graficos.  
})



# Graphics Devices in R (part 1)

library(datasets)  
pdf(file = "myplot.pdf") #Crea un archivo pdf  
with(faithful, plot(eruptions,waiting))# Plotea  
title(main = "Old Faithgul Geyser data")# Agrega titulo  
dev.off()# Cierra el pdf y lo guarda, grafic devices

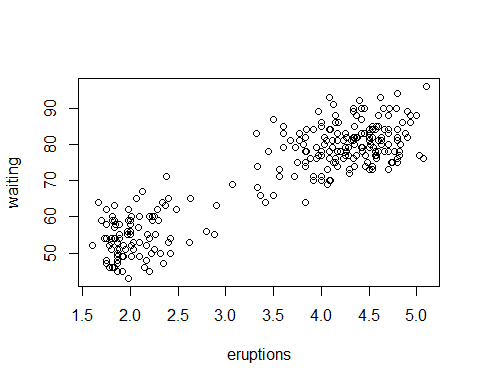
## png   
## 2

# Graphics Devices in R (part 2)

Existen dos tipos basic para guardar archivos: Vector devices and Bitmap Devices

* Vector: Grafico de lineas, sin imagenes o fotos. No sufre la calidad de la imagen (Si los graficos tienen demasiados puntos no es conveniente usar este tipo)
* + pdf: Clasico  
  + svg: Graficos para web (Aguanta animaciones)  
  + win.metafile: Predecesor del pdf  
  + postscript
* Bitmap:
* + png: Bueno para guardar graficos que tienen demasiados puntos  
  + jpeg: Para fotos  
  + tiff: Formato antiguo  
  + bmp: un formato nativo de windows usado para los iconos

with(faithful, plot(eruptions,waiting))



dev.copy(png, file = "eruptions.png") #Esto es mas usado, copia el grafico que se abre en windows y lo convierte en pdf, asi se puede ver que se esta incluyendo en el pdf

## png   
## 3

dev.off()

## png   
## 2