Week 2

Tomás

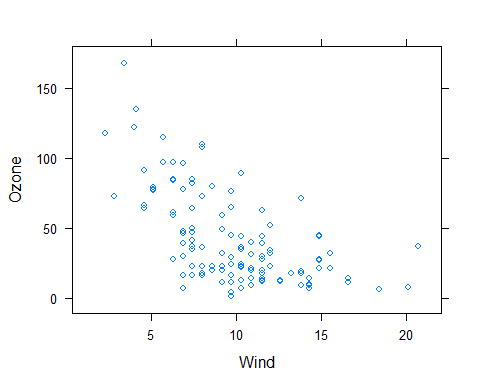
6 de diciembre de 2019

# Lattice Plotting System (part 1)

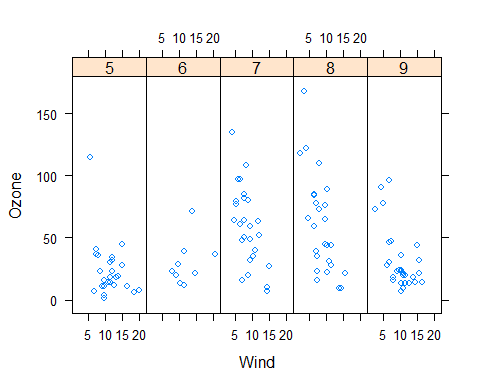
Las funciones mas imporatnte de Lattices son:

* xyplot: Genera graficos de dispersion o puntos
* bwplot: boxplot
* histogram: Histogramas
* stripplot: Diagrama de cajas pero usa puntos
* dotplot: Traza puntos “lineas”
* splom: Matriz de graficas de dispersion (Es como par en el base plot)
* levelplot, contourplot: Graficar datos de imagenes

library(datasets)  
library(lattice)  
xyplot(Ozone ~ Wind, data = airquality) # xyplot(y~x, data)

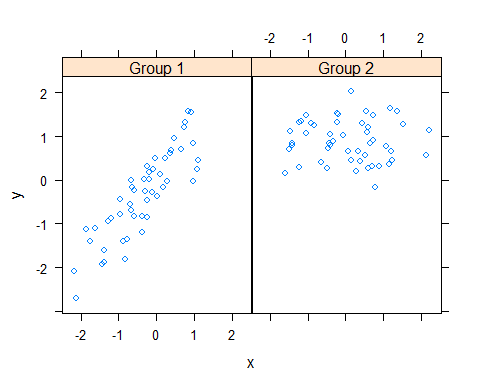


airquality <- transform(airquality, Month = factor(Month))  
xyplot(Ozone ~ Wind | Month, data = airquality, layout = c(5,1)) # La barra permite utilizar categorias para que me las grafique por separado, layout = en que tipo de matriz quiero mostrar los graficos (5 columnas por 1 fila)

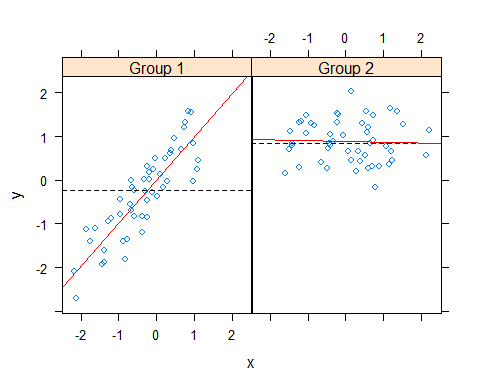


# Lattice Plotting System (part 2)

set.seed(10)  
x <- rnorm(100)  
f <- rep(0:1, each = 50)  
y <- x+f-f\*x+rnorm(100, sd = 0.5)  
f <- factor(f,labels = c("Group 1", "Group 2"))  
xyplot(y ~ x | f, layout = c(2,1)) #Plot normal



xyplot( y ~ x | f, panel = function(x,y,...){ # La funcion panel me permite editar el plot  
 panel.xyplot(x,y,...) # Llamo a la funcion xyplot para crear los puntos  
 panel.abline(h = median(y), lty = 2) #Creo una linea horizontal igual a la media (lty me da las lineas punteadas)  
 panel.lmline(x , y, col = 2) #Una simple regresion lineal para los datos  
})



# ggplot2 (part 2)

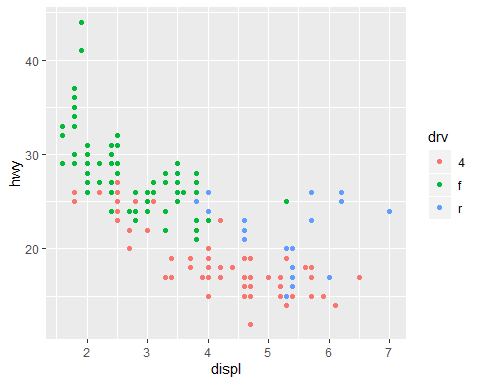
library(ggplot2)

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

qplot(displ,hwy, data = mpg) # qplot(x,y,data)

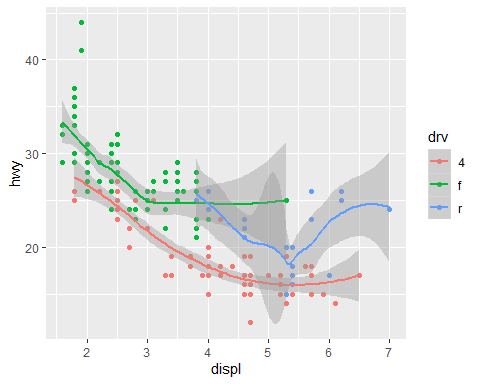


qplot(displ,hwy, data = mpg, color = drv) # Puedo destacar subgrupos (factores) con color = factor



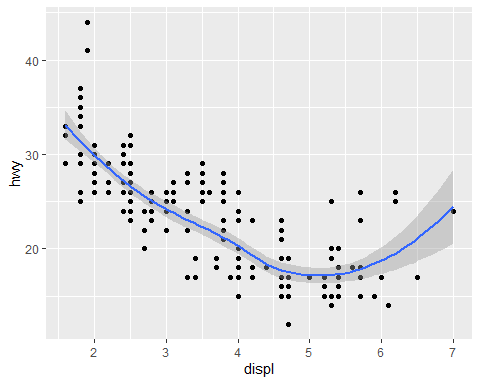
qplot(displ,hwy, data = mpg, color = drv, geom = c("point", "smooth")) #Puedo crear un suavizamiento de los datos por tipo

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



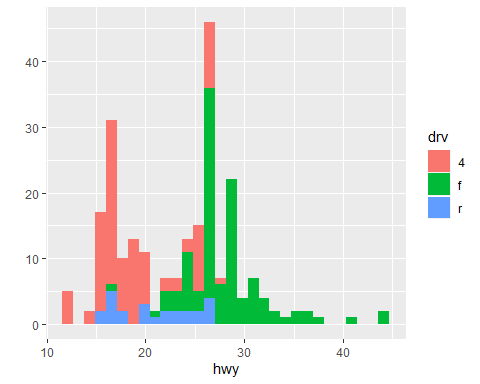
qplot(displ,hwy, data = mpg, geom = c("point", "smooth")) #Aqui lo puedo tener de manera general, la zona gris me indica el 95% de confiabilidad

## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'

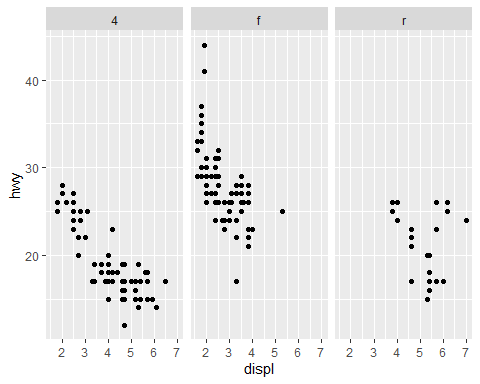


qplot(hwy, data = mpg, fill = drv) #Puedo crear un histograma especificando solamente 1 variables qplot(x,data), aqui los separo por tipos como antes

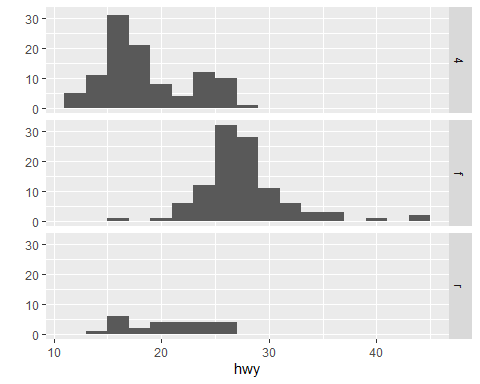
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



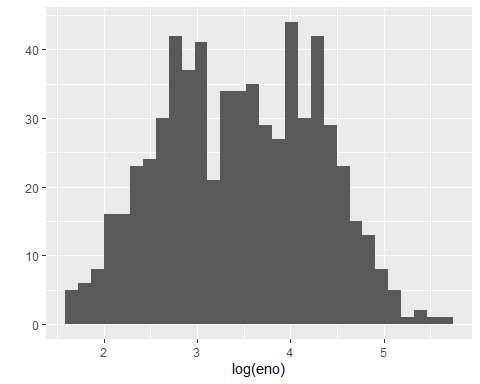
qplot(displ, hwy, data = mpg, facets = .~drv) # Facets permite generar ventanas con varios graficos (El argumento se separa por ~, si pongo la variable categorica al lado derecho obtengo solo filas, al derecho solo columnas)



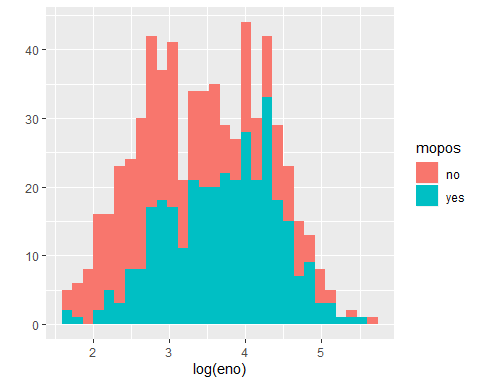
qplot(hwy, data = mpg, facets = drv~., binwidth = 2)



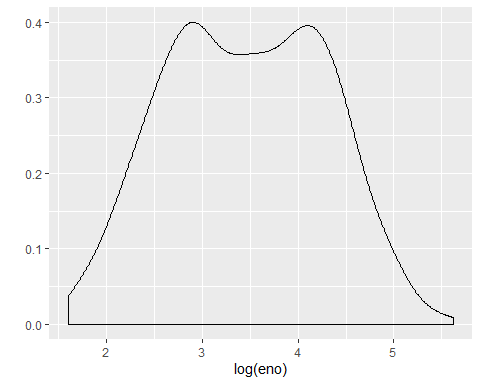
maacs <- read.csv("maacs.csv")  
qplot(log(eno), data = maacs, na.rm =TRUE, bins = 30)



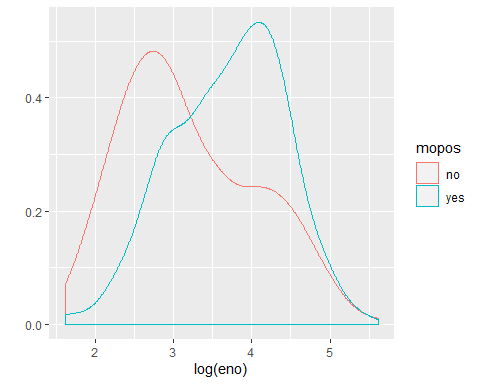
qplot(log(eno), data = maacs, na.rm =TRUE, bins = 30, fill = mopos)



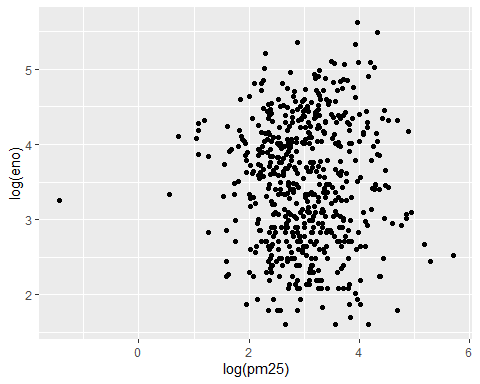
qplot(log(eno), data = maacs , geom = "density", na.rm = TRUE) # Crea un grafico de densidad



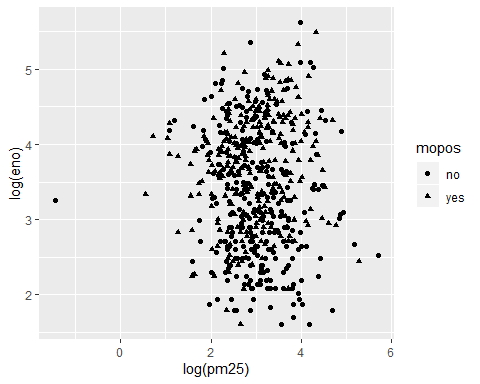
qplot(log(eno), data = maacs , geom = "density", na.rm = TRUE, color = mopos) #Crea el grafico de densidad por tipo



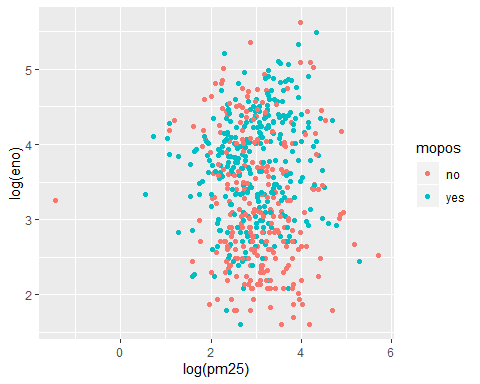
qplot(log(pm25), log(eno), data = maacs, na.rm = TRUE) #Grafico de dispersion entre pm25 y eno



qplot(log(pm25), log(eno), data = maacs, na.rm = TRUE, shape = mopos) #Shapes permite cambiar las figuras

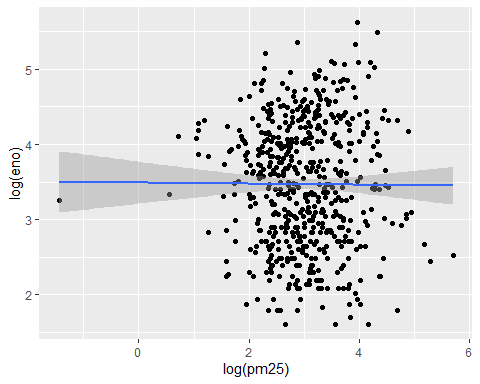


qplot(log(pm25), log(eno), data = maacs, na.rm = TRUE, color = mopos)

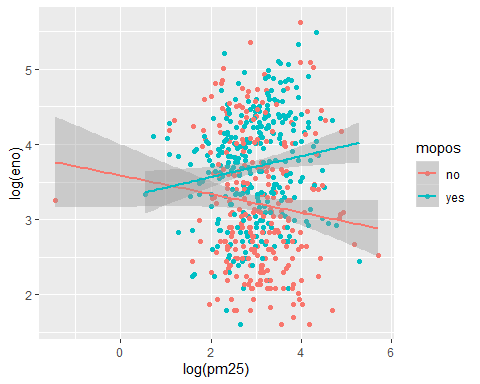


qplot(log(pm25),log(eno), data = maacs, na.rm = TRUE) + geom\_smooth(method = "lm") #Agrego una regresion lineal

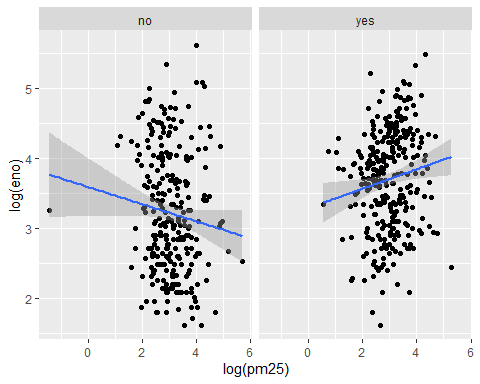
## Warning: Removed 184 rows containing non-finite values (stat\_smooth).



qplot(log(pm25),log(eno), data = maacs, color = mopos, na.rm = TRUE) + geom\_smooth(method = "lm", na.rm = TRUE) #La misma regresion pero por grupos



qplot(log(pm25), log(eno), data = maacs, facets = .~mopos, na.rm = TRUE) + geom\_smooth(method = "lm", na.rm = TRUE)



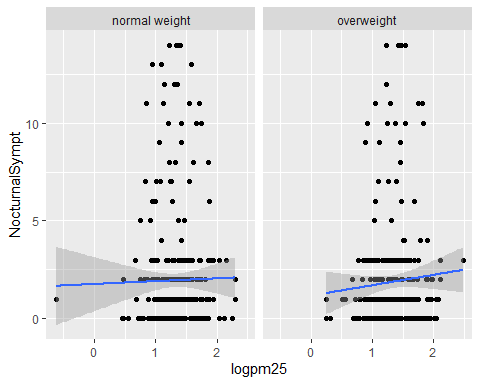
# ggplot2 (part 3)

Los componentes basicos de ggplot2 son:

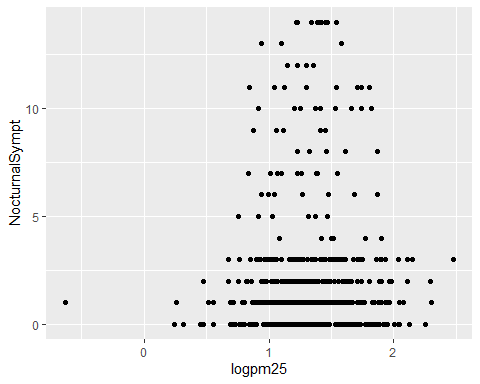
* dataframe: Datos
* aesthetic mappings: Como los datos son mapeados, color, tamaño, posicion
* geoms: figuras geometricas a usar
* facets: Para crear mas de un grafico
* stats: Para realizar transformaciones como suavizamientos, quantiles, regresiones
* scales: Que escala usa aesthetic maps ( male = red, female = blue)
* coordinate system

library(ggplot2)  
qplot(logpm25, NocturnalSympt, data = maacs, facets = .~bmicat, method = "lm", na.rm = TRUE, geom = c("point", "smooth"))

## Warning: Ignoring unknown parameters: method



# Manera mas lenta paso a paso  
g <- ggplot(maacs[,c(6,7,8)], aes(logpm25, NocturnalSympt), na.rm = TRUE) #Creo el grafico  
g + geom\_point(na.rm = TRUE) #Le agrego los puntos

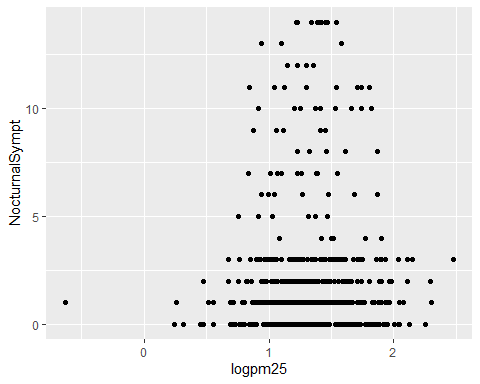


# ggplot2 (part 4)

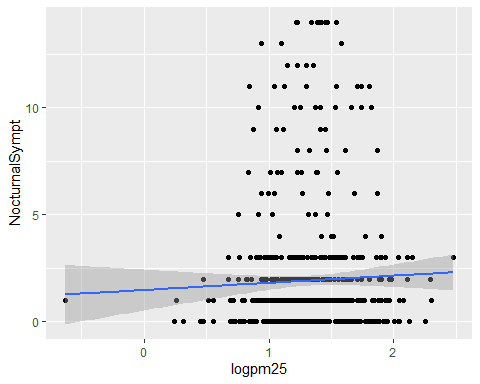
Algunas cosas extras que siempre son utiles:

* xlab()
* ylab()
* labs()
* ggtitle()
* geom(): Existen muchas variaciones de este no solo “points”, “smooth”
* themes(): Permite editar colores, position de la legenda entre otras cosas (El “tema”)
* theme\_gray(): El tema default (fondo gris)
* theme\_bw(): Fondo blanco y negro

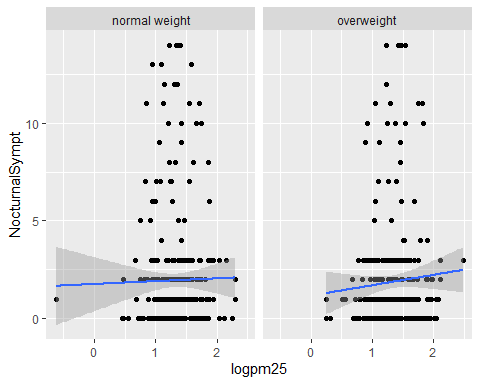
g <- ggplot(maacs[,c(6,7,8)], aes(logpm25, NocturnalSympt), na.rm = TRUE) #Creo el grafico  
g + geom\_point(na.rm = TRUE) #Le agrego los puntos



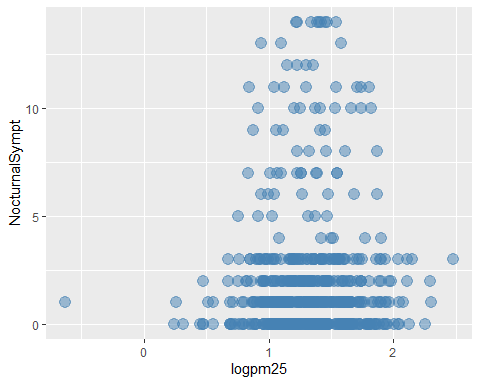
g + geom\_point(na.rm = TRUE) + geom\_smooth(method = "lm", na.rm = TRUE) #Creo una regresion lineal



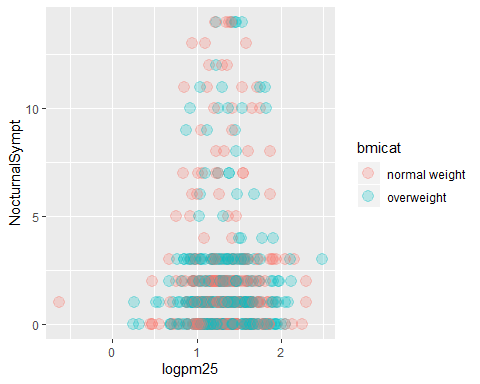
g + geom\_point(na.rm = TRUE) + geom\_smooth(method = "lm", na.rm = TRUE) + facet\_grid(.~bmicat) #Agrego mas de un plot



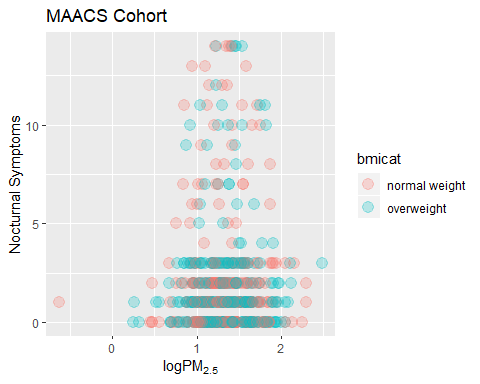
g + geom\_point(color = "steelblue", size = 4, alpha = 1/2, na.rm = TRUE) # Cambio de color, tamaño, transparencia



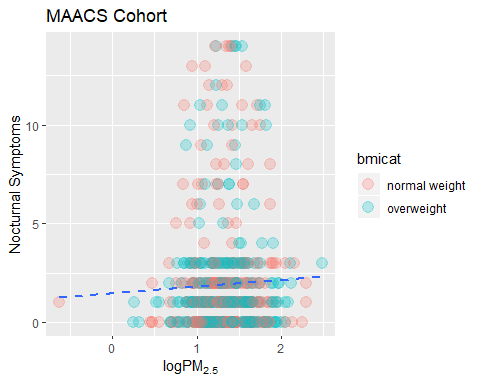
g + geom\_point(aes(color = bmicat), size = 4, alpha = 1/4, na.rm = TRUE) # Cambio de color por categoria



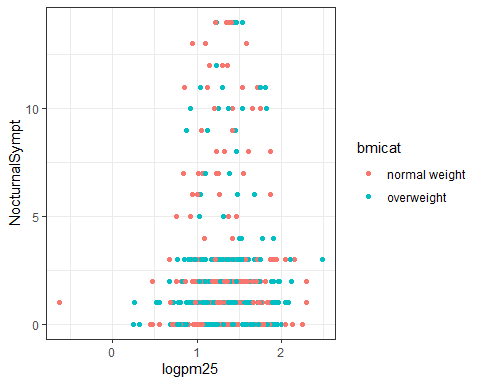
g + geom\_point(aes(color = bmicat), size = 4, alpha = 1/4, na.rm = TRUE) + labs(title = "MAACS Cohort") + labs(x = expression("log"\*PM[2.5]), y = "Nocturnal Symptoms") #Le entrego titulo y nombre a los ejes



g + geom\_point(aes(color = bmicat), size = 4, alpha = 1/4, na.rm = TRUE) + labs(title = "MAACS Cohort") + labs(x = expression("log"\*PM[2.5]), y = "Nocturnal Symptoms") + geom\_smooth(size = 1, linetype = 2, method = "lm", se = FALSE, na.rm = TRUE) # Agrego una regresion lineal, (se = FALSE, desactiva el intervalo de confiaza)

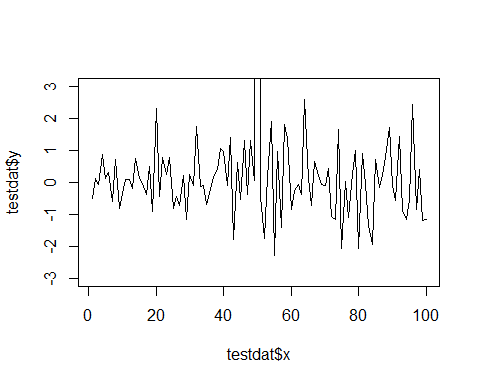


g + geom\_point(aes(color = bmicat), na.rm = TRUE) + theme\_bw(base\_family = "") #Cambio el aspecto del grafico



# ggplot2 (part 5)

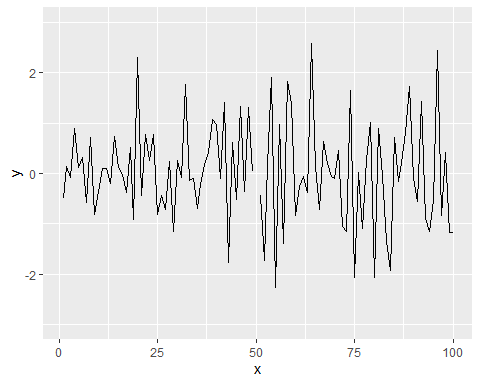
set.seed(100)  
testdat <- data.frame(x = 1:100, y = rnorm(100))  
testdat[50,2] <- 100  
plot(testdat$x,testdat$y, type = "l", ylim = c(-3,3)) #Existe un dato que se escapa de el grafico



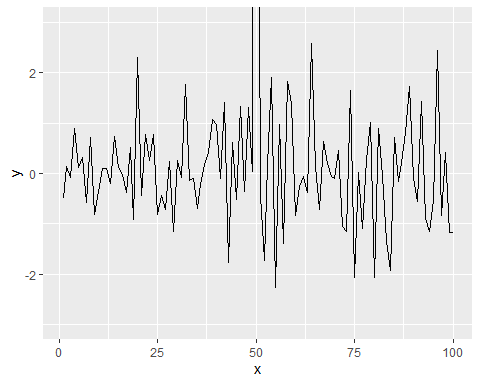
g <- ggplot(testdat, aes(x = x, y = y))  
g + geom\_line() #El valor extremo hace que todo mi grafico se vea demasiado grande



g + geom\_line() + ylim(-3,3) #solo me muestra los datos que estan entre -3 y 3, oculta el resto



g + geom\_line() + coord\_cartesian(ylim = c(-3,3)) #coord\_cartesian permite limitar el plano y no esconder datos



cutpoints <- quantile(maacs$logno2\_new, seq(0,1,length = 4), na.rm = TRUE) #Me crea 4 intervalos para una variable continua poder hacerla categorica  
maacs$logno2\_new <- cut(maacs$logno2\_new, cutpoints) #Corta la variable y la transforma en factores con los puntos dados  
levels(maacs$logno2\_new) #Niveles creados

## [1] "(-0.629,1.18]" "(1.18,1.44]" "(1.44,2.48]"

# Grafico Final

## Matriz de plots

g <- ggplot(maacs, aes(logpm25, NocturnalSymptoms), na.rm = TRUE) g + geom\_point(alpha = 1/3) + facet\_wrap(bmicat ~ logno2\_new, nrow = 2, ncol = 3) + geom\_smooth(methods = “lm”, se = FALSE , col = “steelblue”) + theme\_bw(base\_family = “Avenir”, base\_size = 10) + labs(x = expression(“log” \* PM[2.5])) + labs (y = “Nocturnal Symptoms”) + labs(title = “MAACS Cohort”)

