hands-on (2a part)

# Ajustant els eixos

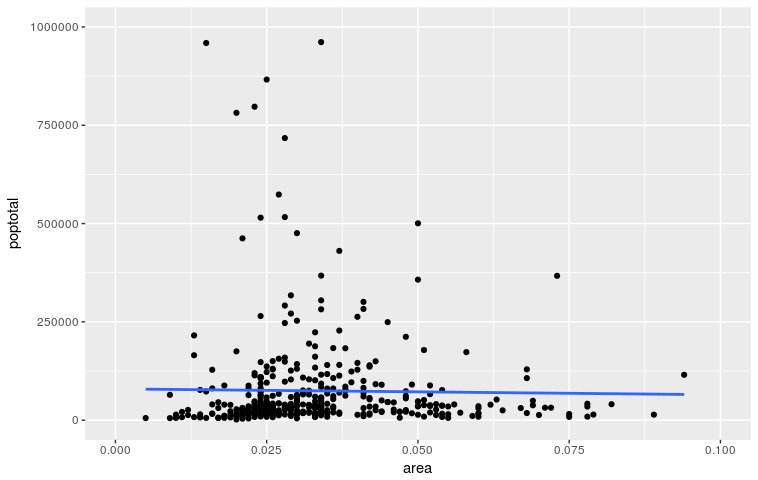
Els eixos *X i Y* es poden ajustar eliminant els punts que no estiguin dins d’un límit que nosaltres porposem utilitzant les funcions **xlim()** i **ylim().** Aquestes funcions el que fan servir és un limit superior i inferior per ajustar el mínim i màxim que volem que ens mostri el gràfic. Veiem-ho:

library(ggplot2) # important! Carreguem la llibreria que ens permet

midwest <- read.csv("http://goo.gl/G1K41K") # llegim la base de dades

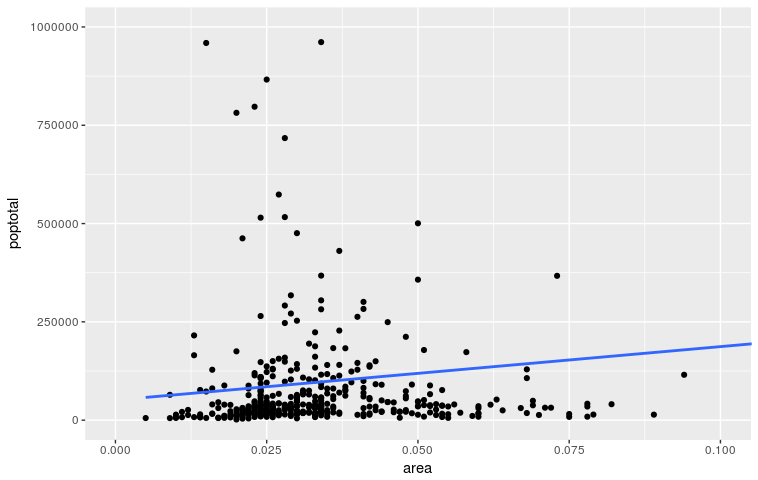
g <- ggplot(midwest, aes(x=area, y=poptotal)) + geom\_point() + geom\_smooth(method="lm", se =FALSE) # Fem el plot igual que abans

g + xlim(c(0, 0.1)) + ylim(c(0, 1000000)) # eliminem els punts que quedin fora dels límits següents



1. **Grafiqueu ara el mateix gràfic sense posar les funcions xlim i ylim. Podem veure que la línia de correlació que tenim en aquest gràfic no és la mateixa que la del gràfic anterior. Per què? (Fixeu-vos amb el *warning message* que us dona el R)**

Ara utilitzeu la funció **coord\_cartesian** i esbrineu quina diferència hi ha amb el **xlim** i **ylim**.



## Afegint labels als eixos

Per afegir text als eixos el més fàcil és utilitzar la funció **labs()** amb l’ajuda d’alguns arguments com **title, subtitle, y, x**...

g <- ggplot(midwest, aes(x=area, y=poptotal)) + geom\_point() + geom\_smooth(method="lm", se =FALSE) # Fem el plot igual que abans

g + labs(title = "Scatterplot", subtitle = "Using midwest dataset", x = "Area", y = "Total population")

A més, podem retocar el format (font, mida, color...) a través de la funció **theme().**L’últim dia vam veure que amb la funció theme() podem canviar l’estil del gràfic (fons, tipus de grid...) però també permet canviar el format juntament amb la funció **element\_text().** La idea és que s’ha de crear primer el tipus d’element amb element text i li passarem com a argument a la funció theme.

g <- ggplot(midwest, aes(x= area, y=poptotal)) + geom\_point() + geom\_smooth(method="lm") + labs(title = "Scatterplot", subtitle = "Using midwest dataset", x = "Area", y = "Total population")

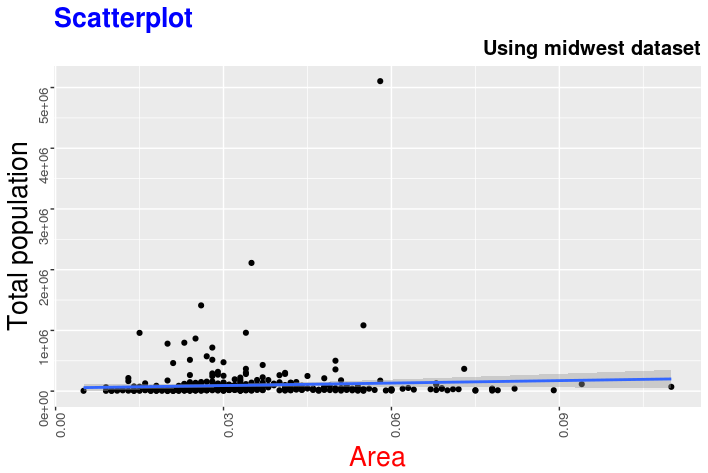
title\_format <- element\_text(size=20, face=”bold”, color = “red”)

x\_format <- element\_text(size = 10, angle = 30, family = “American Typewriter”, vjust = 0.5)

g + theme(plot.title=title\_format, axis.text.x=x\_format)

#Altres paràmetres que es poden modificar són: plot.title, plot.subtitle, plot.caption, axis.title.x, axis.title.y, axis.text.x o axis.text.y

1. **Intenteu fer un gràfic semblant al que hi ha a continuació jugant amb els diferents paramètres que hem vist abans i la funció theme().**



g + theme(plot.title=element\_text(size=20, face="bold", color="blue", hjust=0),

+ plot.subtitle=element\_text(size=15, face="bold",hjust=1),

+ axis.title.x=element\_text(size=20, color = "red"),

+ axis.title.y=element\_text(size=20),

+ axis.text.x=element\_text(size=10, angle = 90), +

axis.text.y=element\_text(size=10, angle = 90))

# scatterplots (geom\_smooth)

Anem a veure diferents tipus de gràfics. Bàsicament veurem 4 gràfics: scatterplots (amb el qual hem fet la primera part del tutorial), el boxplot, gràfic de barres i histogrames.

Lo primer que farem serà una correlació, com hem fet fins ara, però amb una nova data base. A priori, una correlació permet la visualització de dues variables. En aquest exemple anirem veient com també serveix per veure 3 o, fins i tot, 4 variables en un sol gràfic.

library(ggplot2)

mpg <- read.csv("http://goo.gl/uEeRGu")

theme\_set(theme\_bw()) # amb aquesta comana posem com a predeterminat el tema *bw*

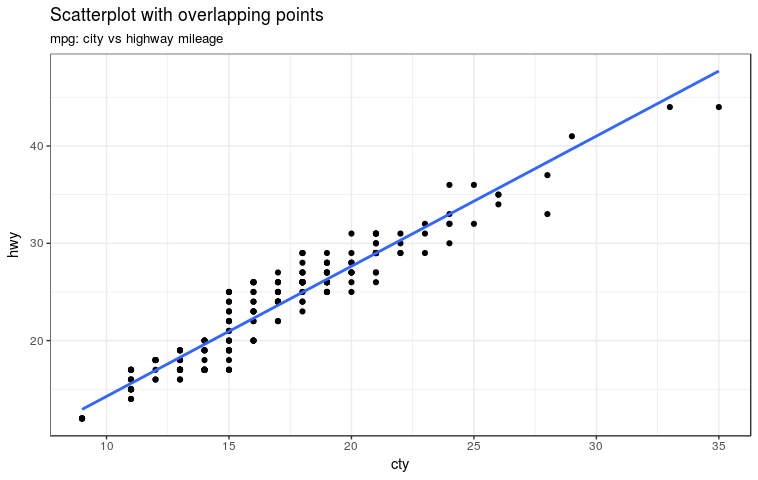
g <- ggplot(mpg, aes(cty, hwy))

# Scatterplot

g + geom\_point() +

geom\_smooth(method="lm", se=F) +

labs(subtitle="mpg: city vs highway mileage", y="hwy", x="cty", title="Scatterplot with overlapping points")



Aquí podem veure l’alta correlació que hi ha entre el kilometratge de les ciutats i el kilometratge de les carreteres. Però què passa? Si ens fixem amb el número de punts, no apareixen tots els que hi ha a la data base.

Com podem manejar-ho? El que fem és utilitzar un marge per graficar tots els punts possibles (encara que no sigui exactament la seva posició al gràfic).

1. **Utilitzeu la funció geom\_count() per fer el plot i descobriu què aporta respecte al geom\_point()**

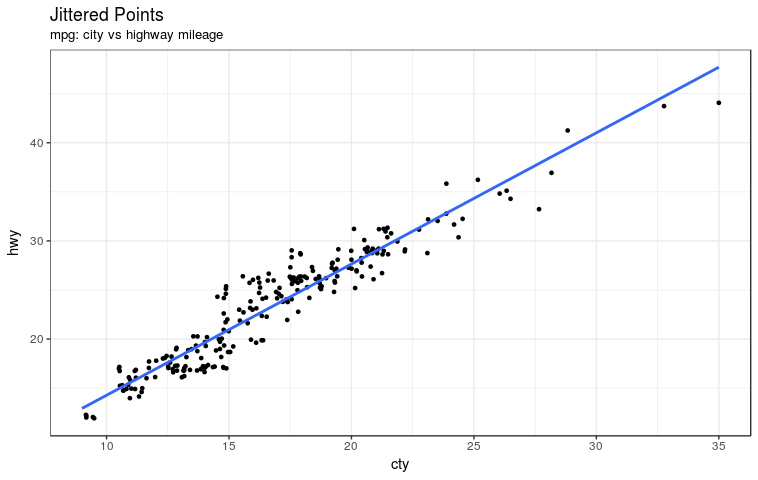
Una altra manera de visualitzar les dades que tenen valors repetits és amb **geom\_jitter().** Observeu quina diferència hi ha respecte geom\_point i geom\_count.

g <- ggplot(mpg, aes(cty, hwy))

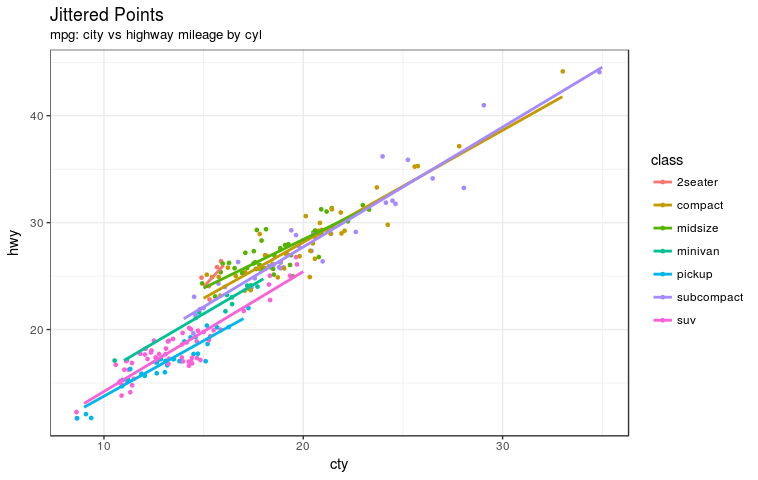
g <- g + geom\_jitter(width = .5, size=1) +

labs(subtitle="mpg: city vs highway mileage", y="hwy", x="cty", title="Jittered Points")

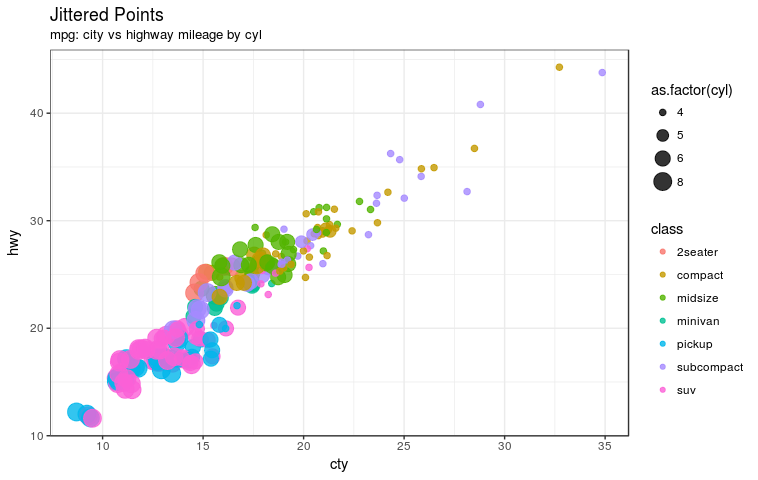
g + geom\_smooth(method="lm", se=F)



1. **Hem vist com crear una correlació per tota la mostra. Anem a crear ara una línia de correlació per cada classe de cotxe (class). Utilitzeu la funció aes() i el paràmetre color en la funció ggplot que hem creat abans per fer-ho.**



1. **Finalment, i per afegir l’última dimensió, anem a reflectir la cilindrada (cyl) en la mida dels punts. Per fer-ho, igual que abans, utilitzem la funció aes() i la variable size en la part de “geom\_jitter” del gràfic anterior**



# boxplots (geom\_boxplot)

Anem a pel segon tipus de gràfic. Els boxplot són una eina excel·lent per estudiar la distribució d’una variable a través de diferents grups. Els boxplots et permeten veure la mediana i el rang d’aquesta variable per cadascun dels grups d’interès.

library(ggplot2)

theme\_set(theme\_classic()) #Canviem el tipus de tema a “clàssic”

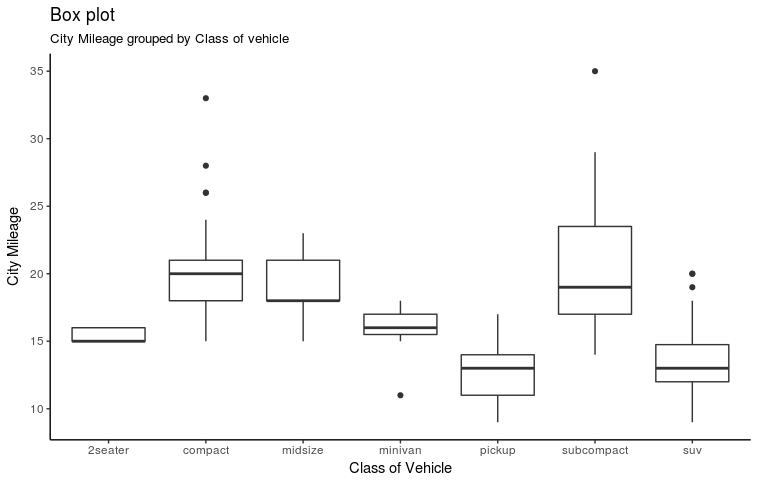
# Plot

g <- ggplot(mpg, aes(class, cty))

g + geom\_boxplot() +

labs(title="Box plot", subtitle="City Mileage grouped by Class of vehicle", x="Class of Vehicle",

y="City Mileage")



Anem a pintar cada cada “caixa” d’un color. Ho farem amb una de les paletes que hem vist a l’altre tutorial (Set1). Per fer-ho simplement utilitzarem el paràmetre fill i la variable class...

library(ggplot2)

theme\_set(theme\_classic()) #Canviem el tipus de tema a “clàssic”

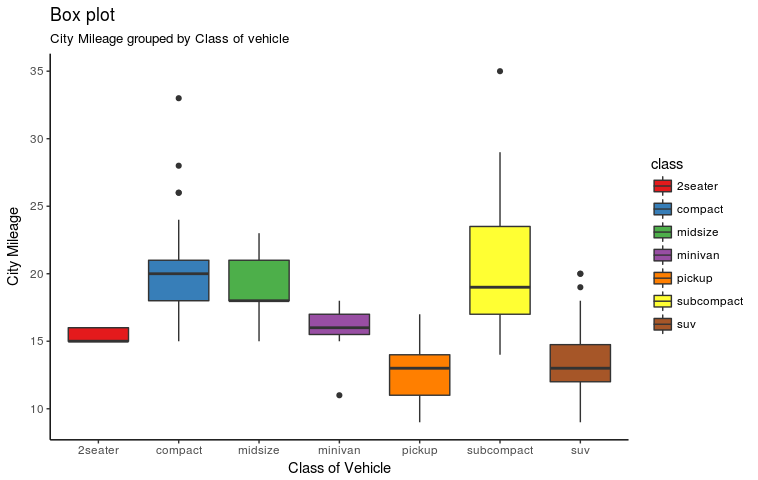
# Plot

g <- ggplot(mpg, aes(class, cty))

g + geom\_boxplot(aes(fill = class)) +

scale\_fill\_brewer(palette = "Set1") +

labs(title="Box plot", subtitle="City Mileage grouped by Class of vehicle", x="Class of Vehicle",+ y="City Mileage")



Ara graficarem els punts corresponents per cada grup. Per fer-ho utilitzarem ela funció geom\_jitter() que hem vist a l’apartat anterior.

library(ggplot2)

theme\_set(theme\_classic()) #Canviem el tipus de tema a “clàssic”

# Plot

g <- ggplot(mpg, aes(class, cty))

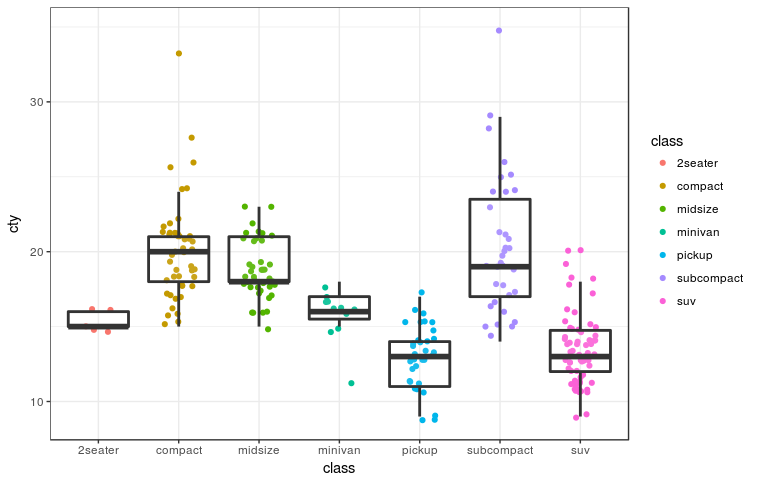
g + geom\_boxplot(aes(fill = class)) +

scale\_fill\_brewer(palette = "Set1") +

geom\_jitter(width = 0.1) +

labs(title="Box plot", subtitle="City Mileage grouped by Class of vehicle", x="Class of Vehicle",+ y="City Mileage")

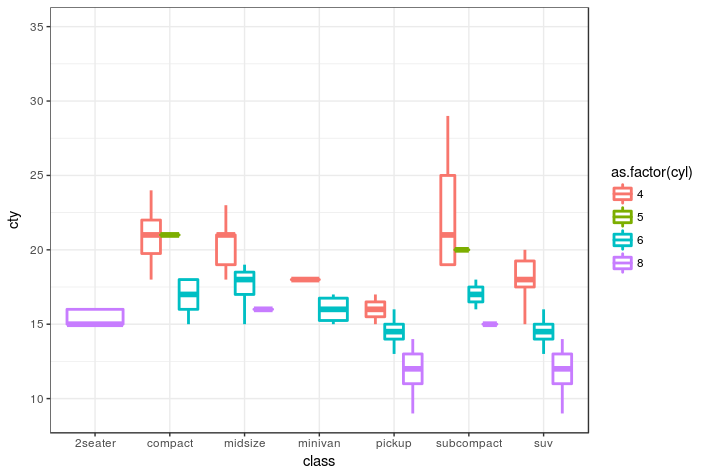
1. **Toca els paràmetres que creguis que són necessaris per obtenir un gràfic com el següent.**



*g <- ggplot(mpg, aes(class, cty))*

*g + scale\_fill\_brewer(palette = "Set1") + geom\_jitter(aes(color = class), width = 0.2) + geom\_boxplot(alpha = 0.1, size = 1, outlier.alpha = 0) + theme\_bw()*

1. **Finalment, en aquest tipus de plot també podem subclassificar dins de cada grup. És a dir, una “caixa” per cada grup. Intenteu crear un gràfic com el següent:**



g <- ggplot(mpg, aes(class, cty))

g + scale\_fill\_brewer(palette = "Set1") + geom\_boxplot(aes(color = as.factor(cyl)), size = 1, outlier.alpha = 0) + theme\_bw()

# DIagrama de barres (i histogrames)

Anem a dibuixar un altre tipus de diagrames: diagrama de barres. Per a aquest tipus de gràfics s’utilitza la funció **geom\_bar()**. Veurem que hi ha dos tipus de gràfic diferents. Podem graficar variables categòriques i variables quantitatives.

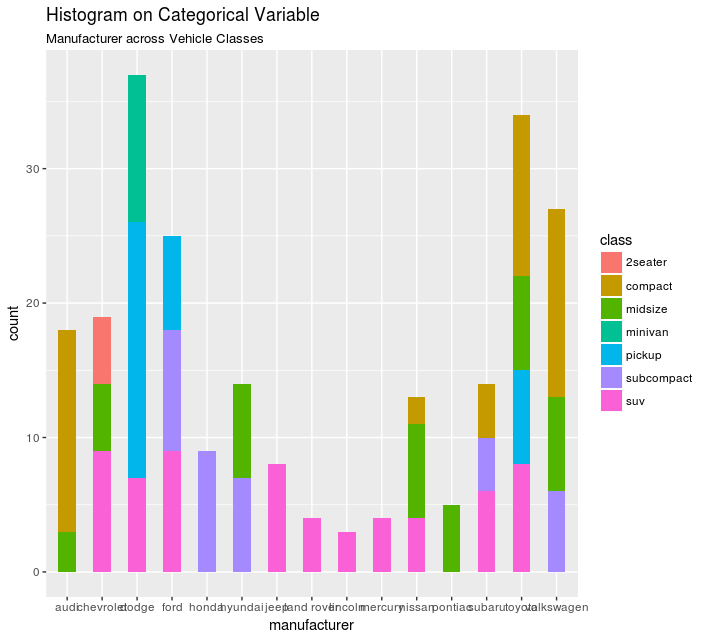
g <- ggplot(mpg, aes(manufacturer))

g + geom\_bar(aes(fill=class), width = 0.5) +

labs(title="Histogram on Categorical Variable",

subtitle="Manufacturer across Vehicle Classes")

1. **Si us fixeu, en aquest gràfic no hem posat cap ‘y’. Mireu el gràfic i la base de dades i esbrineu què hi ha a l’eix ‘y’.**

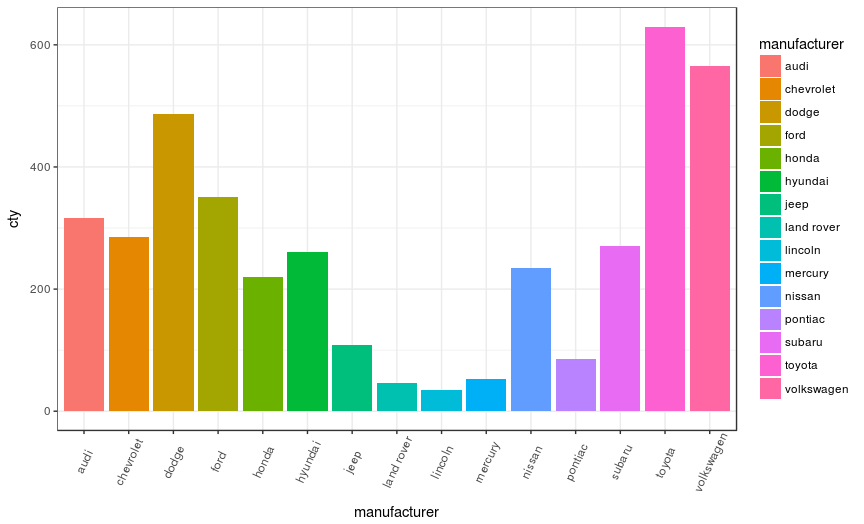


1. **Després de dibuixar aquest histograma, anem a utilitzar el diagrama de barres per veure com es reparteixen el kilometratge (cty) entre les diferents marques (eix X). (Utilitzeu l’argument stat = ‘identity’)**

g <- ggplot(mpg, aes(x = manufacturer, y = cty))

g + geom\_bar(aes(fill = manufacturer), width = 0.5, stat = "identity") +

labs(title="Bar diagram of cty between manufacturers") + theme(axis.text.x = element\_text(angle = 65, vjust = 0.5)) + theme\_bw()



# spagHeTti plot

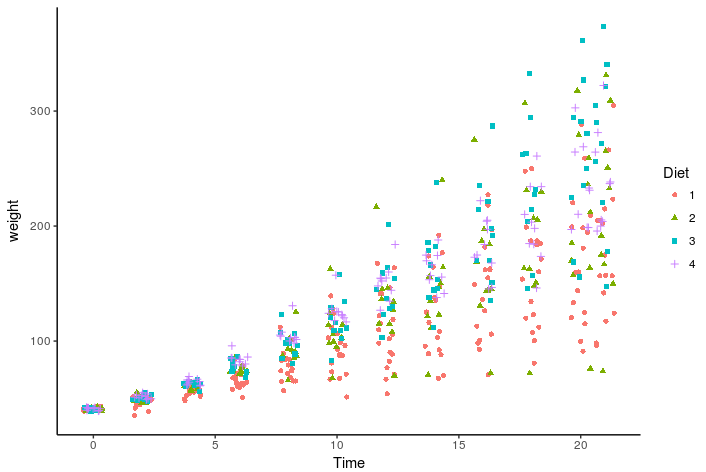
Anem a crear un spaghetti plot. Aquest tipus de gràfic s’utilitza quan tens informació longitudinal de les dades, és a dir, dades que es van repetint periòdicament. Una bona base de dades per crear aquest tipus de gràfiques seria una relacionada amb la meteorologia o amb l’evolució de subjectes al llarg del temps.

1. **ChickWeight serà la dataset que utilitzarem. Inspeccioneu aquesta base de dades i esbrineu com evolucionen les dades**

datasets:: ChickWeight #Aquest serà el nostre dataset

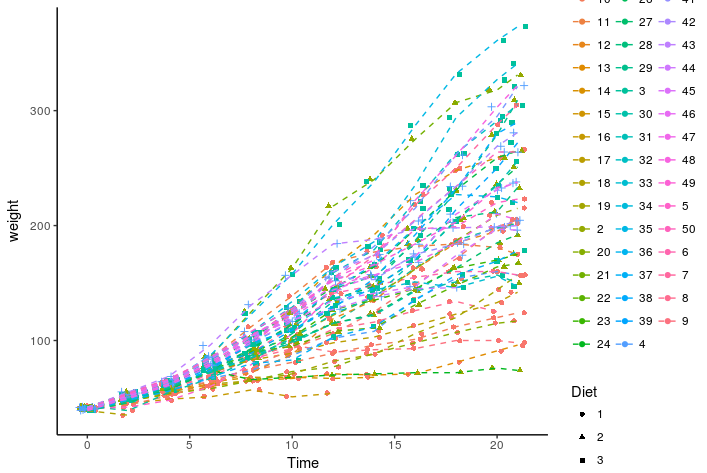
ggplot(ChickWeight, aes(Time,weight)) + geom\_jitter() # veiem com són les dades

1. **Modifiqueu la línia anterior i grafiqueu un gràfic amb punt per cada seguiment on cada dieta tingui una forma (shape) i un color (col) diferent.**

****

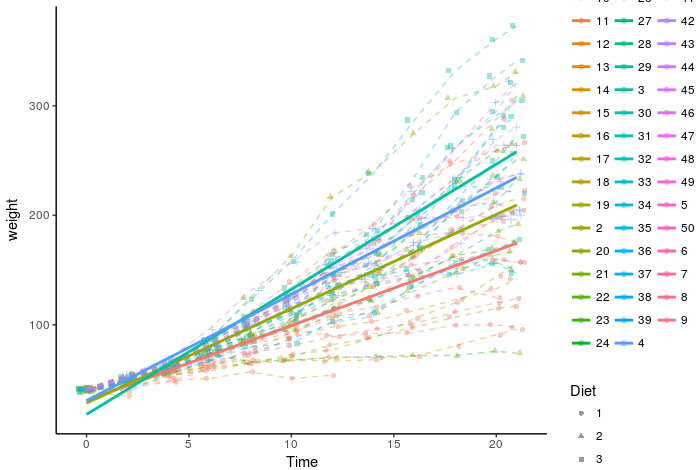
1. **A més, uniu tots els punts corresponents al mateix individu amb una línia de punts que també sigui del mateix color per cada subjecte amb geom\_line**

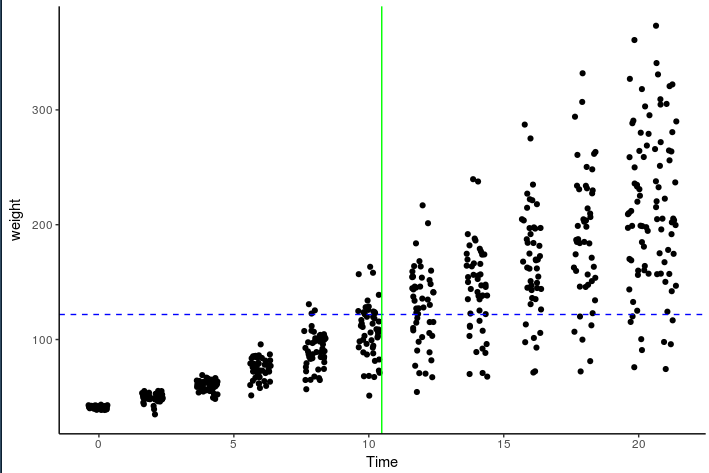
ggplot(ChickWeight, aes(Time,weight)) + geom\_jitter(aes(color = Diet, shape = Diet)) + geom\_line(aes(color = Chick), linetype = 2)

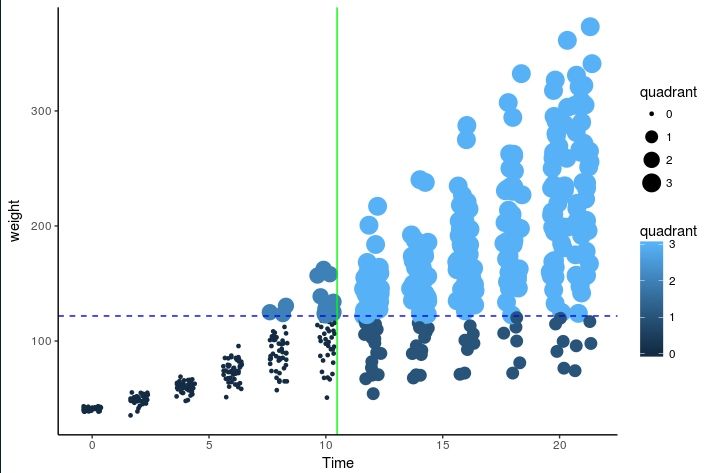
****

1. **Ara, ajusteu un línia recta per cada dieta. Si podeu, feu que els punts i les línies individuals siguin semi-transparents (feu servir l’argument “alpha” ).**

ggplot(ChickWeight, aes(Time,weight)) + geom\_jitter(aes(color = Diet, shape = Diet)) + geom\_line(aes(color = Chick), linetype = 2) + geom\_smooth(aes(x = Time, y = weight, color = Diet), method = "lm", se = FALSE)

****

1. **Afegiu dues línies (una horitzontal de punts de color blau i una vertical continua de color taronja) per dividir la mostra en 4 quadrants. La línia horitzontal ha de ser la mitja de pes de tota la mostra i la línia vertical ha de ser la mitja de seguiment dels individus que segueixen la dieta 1. (mireu ?geom\_hline i ?geom\_vline)**
2. **Finalment, feu que cada quadrant tingui una mida, forma i color diferent. (Ajuda: Creeu una columna nova a n’aquest dataset que assigni a cada punt el quadrant al qual pertany. Nota2: Per què es compleixin dues condicions a l’hora d’assignar un valor heu d’aplicar el símbol “&”: condició1 & condició2)**

****

# exportar el gràfic

Hi ha moltes maneres d’exportar un gràfic. En aquest tutorial bàsicament n’ensenyarem dos diferents. La primera i més senzilla, com una imatge (png, tif o jpg) amb la comana **ggsave()**.

# primer fem el gràfic

gg <- ggplot(midwest, aes(x=area, y=poptotal)) +

geom\_point(aes(col=state, size=popdensity)) +

geom\_smooth(method="lm", se=F)

# i després el guardem...

ggsave(“gràfic\_test.png",width = 9, height = 5, units ="in", dpi = 300)

La següent manera de fer-ho és exportant-ho a un power point.

#Carreguem les llibreries necessaries

library(ReporteRs)

library(ReporteRsjars)

plantilla <- "//dspau.santpau.es/U/CarpUMNeurologia/BaseUdM/Export-analisis/Resultados/ReportTemplate.pptx"

doc <- pptx(title="Report", template=plantilla)

#Ara és quan creem el gràfic

gg <- ggplot(midwest, aes(x=area, y=poptotal)) +

geom\_point(aes(col=state, size=popdensity)) +

geom\_smooth(method="lm", se=F) + xlim(c(0, 0.1)) + ylim(c(0, 500000)) +

labs(title="Area Vs Population", y="Population", x="Area ")

#Afegim una diapositiva amb el nostre gràfic

doc <- addSlide(doc,"Title and Content")

doc <- addTitle(doc,"El nostre gràfic", level=3)

doc <- addPlot(doc, fun=function() print(gg),vector.graphic =TRUE )