Tema 3

Gráficos de datos

3.1 Introducción

El paquete de gráficos **ggplot2** permite definir gráficos de una manera eficiente y sencilla. En estos apuntes se presenta su uso básico con distintos ejemplos. Se recomienda consultar **??**esta página para ver ejemplos más completos. Asímismo, el manual de esta librería puede consultarse en R una vez instalada.

Para poder usar el paquete **ggplot2** cargaremos el paquete **tidyverse** que lo incluye. Tambiém cargaremos un par de paquetes que incluyen bases de datos que utilizaremos en este capítulo:

```
> library(tidyverse)
> library(UsingR)
> library(MASS)
```

3.2 Uso básico

Para definir un gráfico es necesario siempre empezar por la instrucción **ggplot**. Por ejemplo podemos utilizar la base de datos **cars** y representar la distancia necesaria de frenado en función de la velocidad (fig.3.1).

```
> ggplot(cars,aes(x=speed,y=dist)) + geom_point()
```

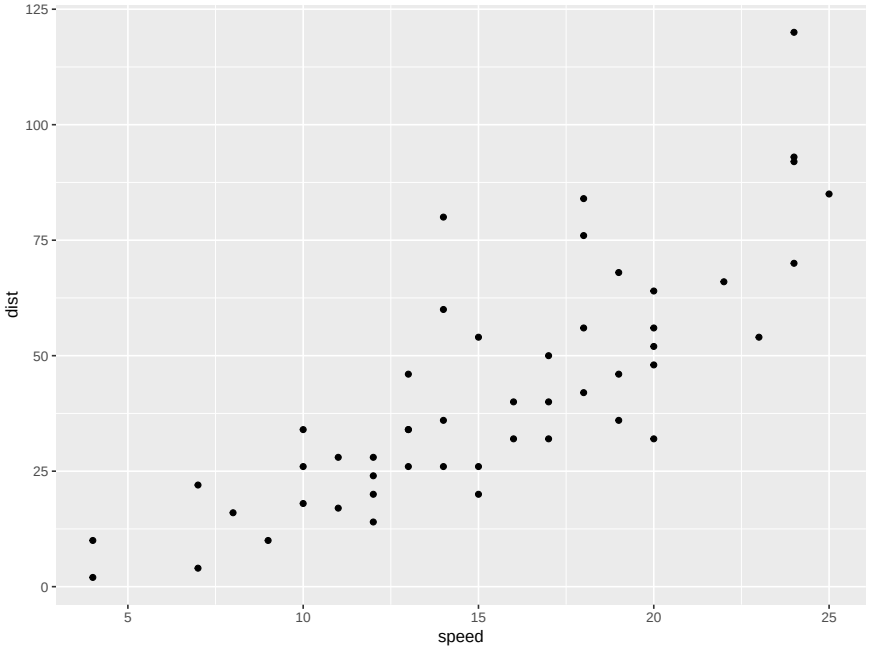

Figure 3.1: Distancia de frenado en función de la velocidad

En esta instrucción, se define la base de datos que utilizamos y la 'estética' básica de la gráfica, es decir qué variables vamos a representar. Una vez definida, se añade una 'geometria', en este caso los puntos que indican la distancia necesaria para frenar en cada caso respecto a la velocidad del coche.

De manera muy simple podemos añadir opciones a la gráfica. Por ejemplo, podemos etiquetar los ejes (fig.3.2):

```
> ggplot(cars,aes(x=speed,y=dist)) +
+         geom_point() +
+         ggtitle("Distancia de frenado")+
+         xlab("Velocidad")+
+         ylab("Distancia")
```

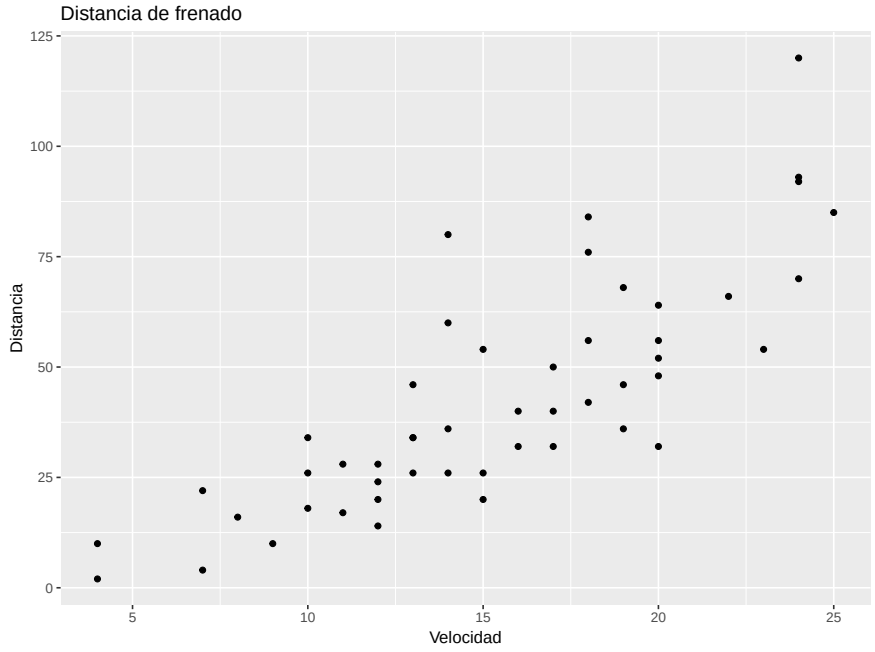

Figure 3.2: Distancia de frenado en función de la velocidad

La geometría admite sus propios parámetros. Por ejemplo, podemos hacer que los puntos sean rojos (color='red'), un poco más grandes (size=3) y que se representen por cuadrados (shpae=15)(fig.3.3):

```
> ggplot(cars,aes(x=speed,y=dist)) +
+         geom_point(color='red',size=3, shape=15) +
+         labs(title="Distancia de frenado",
+                         x="Velocidad",
+                         y="Distancia")
```

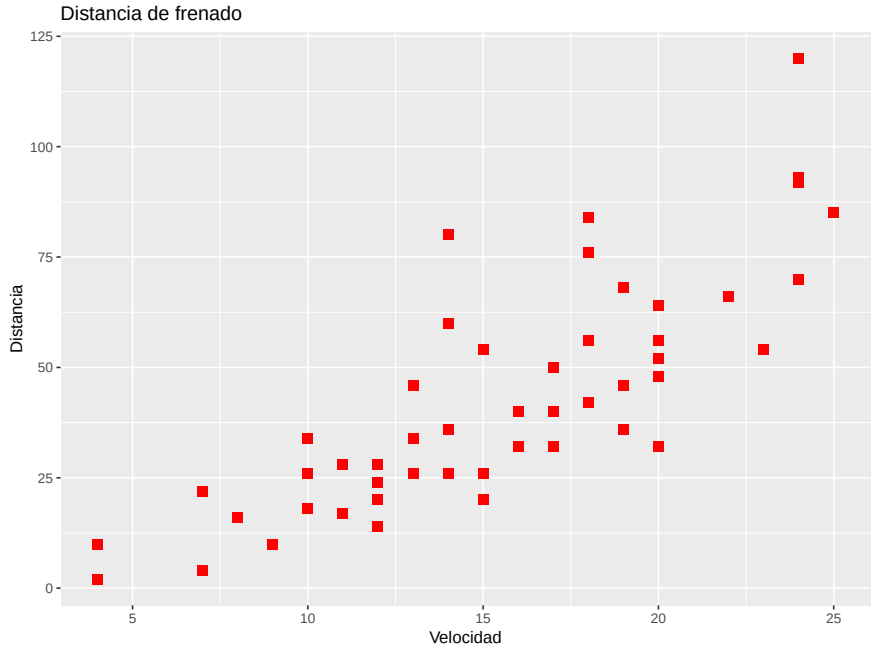

Figure 3.3: Distancia de frenado en función de la velocidad

Los gráficos en **ggplot2** pueden asignarse a objetos para ser modificados posteriormente. Por ejemplo (fig.3.4):

```
> p1 <- ggplot(cars,aes(x=speed,y=dist))
> p1 + geom_point()
```

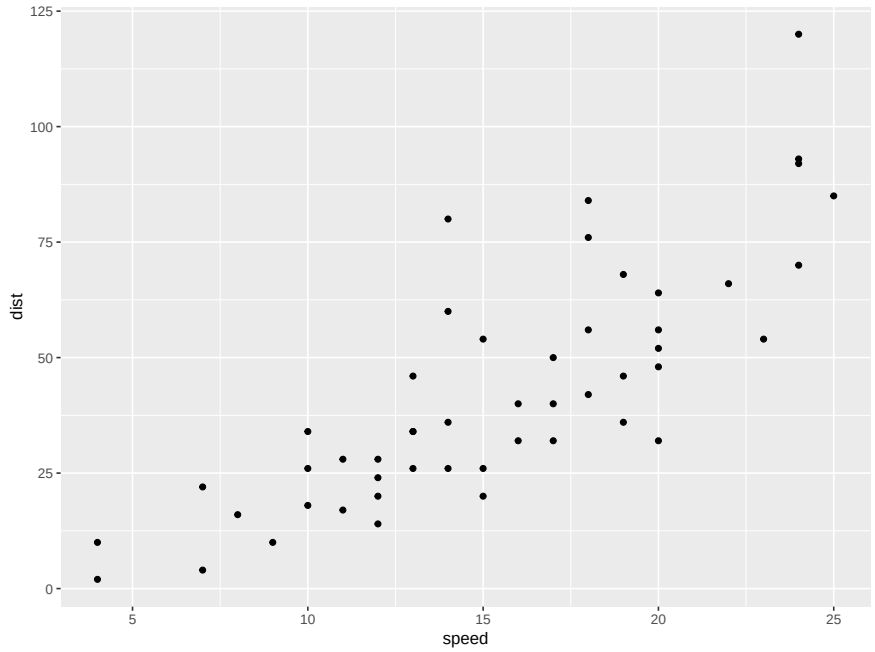

Figure 3.4: Distancia de frenado en función de la velocidad

Ahora podemos obtener una gráfica distinta aprovechando que tenemos el objeto **p1**. En este ejemplo utilizamos **geom_smooth** que permite representar una línea de tendencia. Para obtener la línea de regressión utilizamos method='lm' como opción. (fig.3.5):

```
> p1 + geom_point() + geom_smooth(method='lm')
```

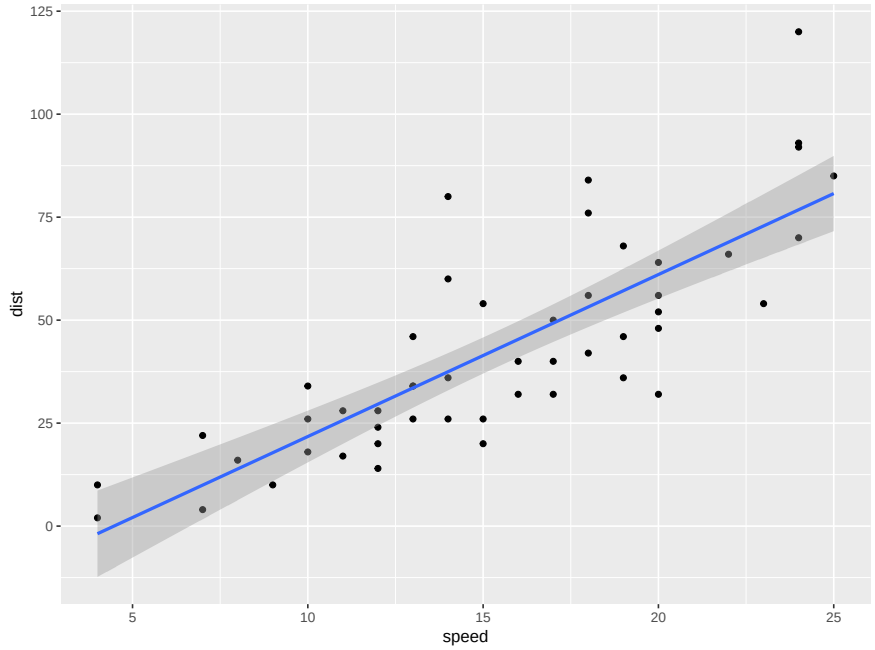

Figure 3.5: Distancia de frenado en función de la velocidad

3.3 Histogramas

Un histograma permite representar la distribución de valores de una variable contínua observados en una muestra. Por ejemplo, en los datos **fat** disponible en la libreria **UsingR** se recogen datos de distintas variables fisiológicas de una muestra de personas. Podemos obtener una gráfica de la distribución de grasa corporal mediante (fig.3.6):

```
> ggplot(fat,aes(x=body.fat)) +
+         geom_histogram(fill="white",color="black")
```

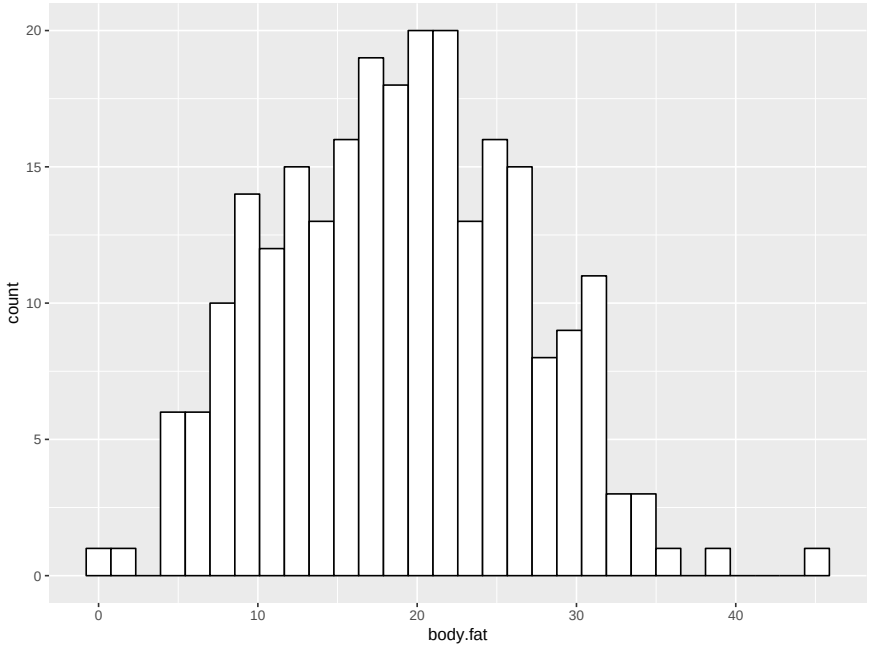

Figure 3.6: Distribución de la grasa corporal

Podemos observar que el resultado es un tanto confuso. Podemos mejorar la gráfica cambiando la anchura de los intervalos (fig.3.7):

```
> ggplot(fat,aes(x=body.fat)) +
+         geom_histogram(fill="white",color="black",binwidth=5)
```

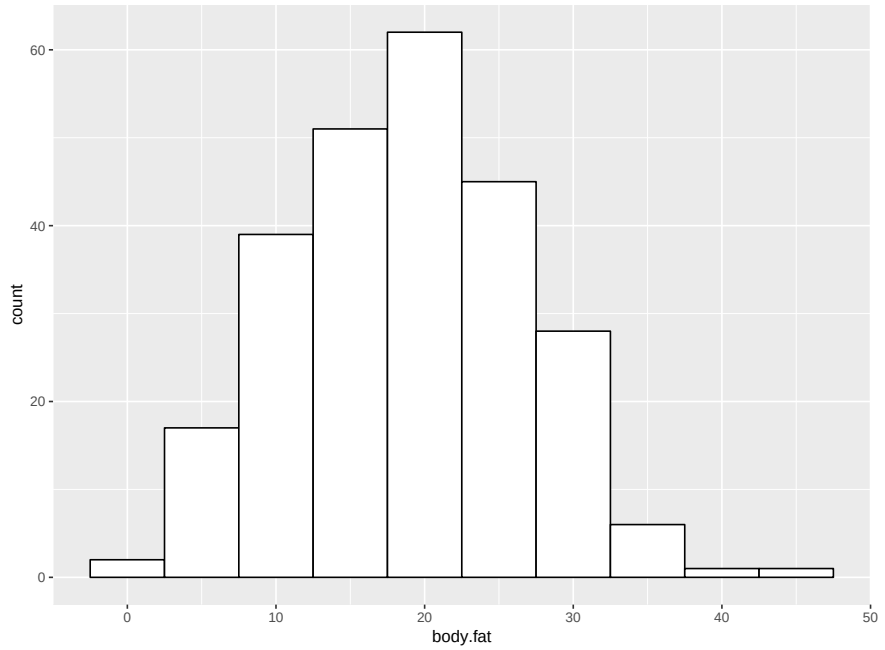

Figure 3.7: Distribución de la grasa corporal

Por defecto, el histograma representa la frecuencia absoluta. Para obtener el porcentage debemos especificarlo en la estética (fig.3.8):

```
> ggplot(fat,aes(x=body.fat)) +
+         geom_histogram(aes(y = (..count..)/sum(..count..)),
+                 fill="white",
+                 color="black",
+                 binwidth=5)
```

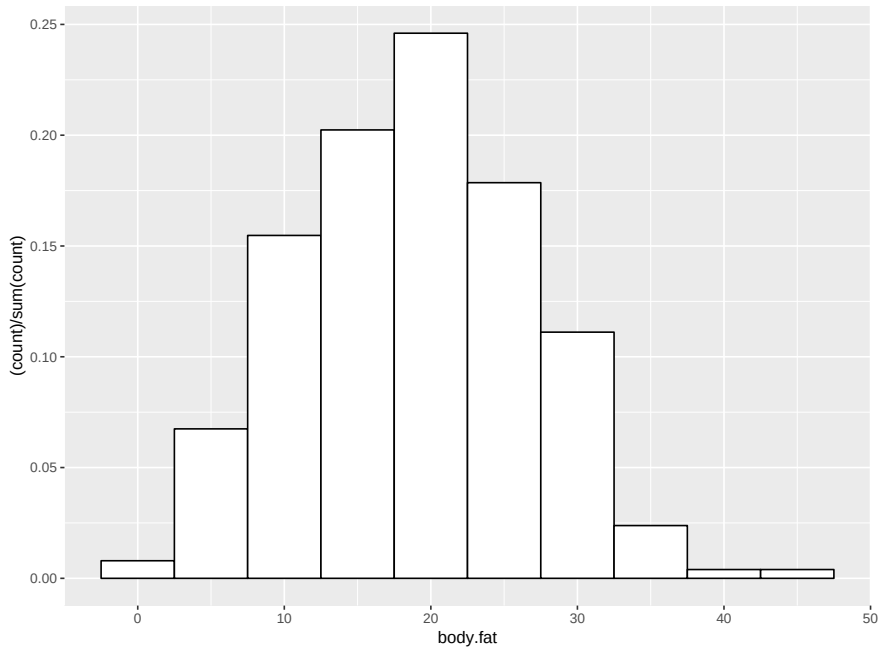

Figure 3.8: Distribución de la grasa corporal

3.4 Boxplot

El gráfico boxplot permite resumir de manera eficiente distintos índices estadísticos. Por ejemplo (fig.3.9):

```
> p <- ggplot(ToothGrowth,aes(supp,len))
> p + geom_boxplot(aes(fill=supp))
```

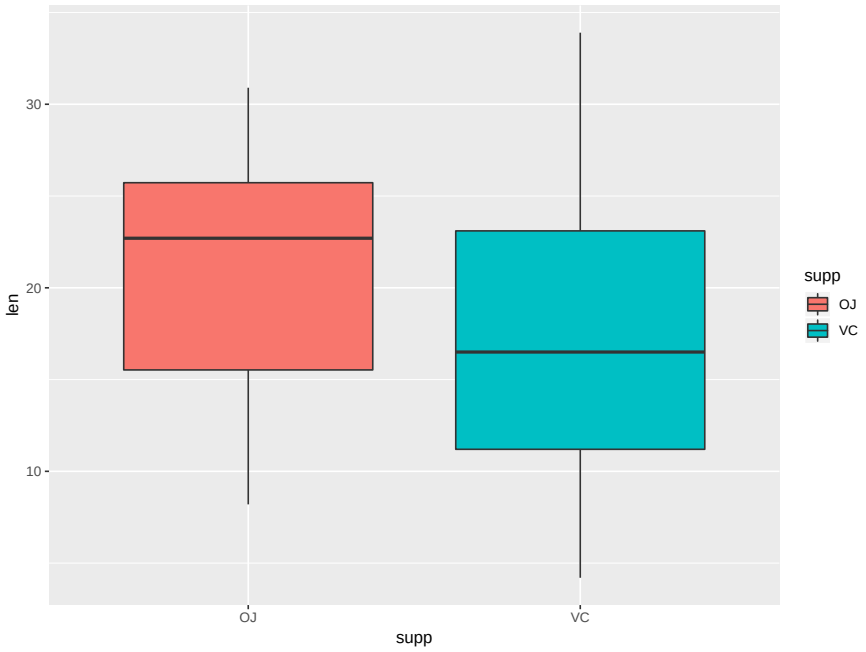

Figure 3.9: Longitud de los dientes en función del suplemento alimenticio

Los valores correspondientes al límite superior de la caja indican el percentil 75, el límite inferior corresponde al percentil 25 y la línia horizontoal a la mediana o percentil 50. Las línias verticales indican el valor máximo y mínimo obervado. Podemos superponer los puntos correspondientes a las obervaciones añadiendo (fig.3.10):

```
> p + geom_boxplot(aes(fill=supp))+ geom_point()
```

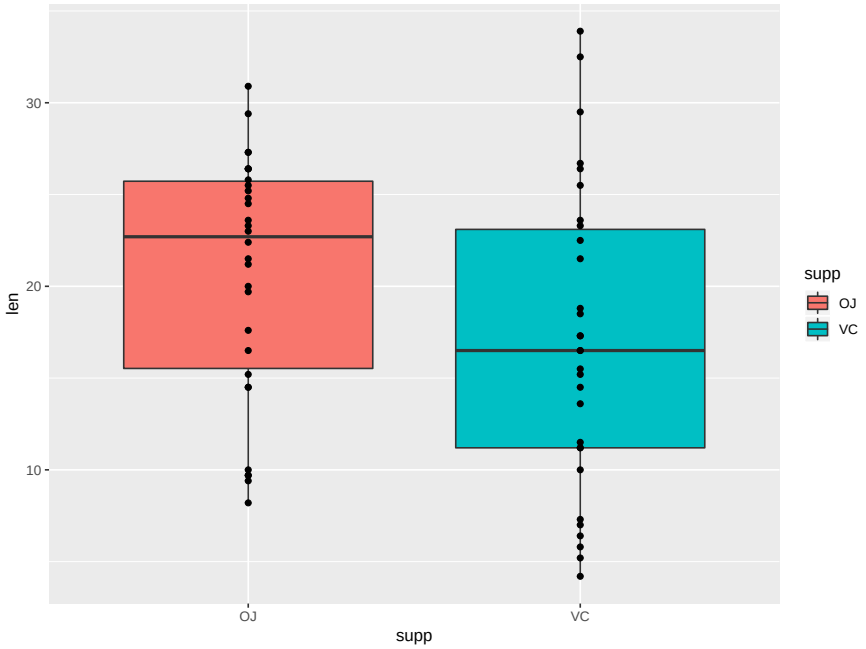

Figure 3.10: Longitud de los dientes en función del suplemento alimenticio

En muchos casos es interesante considerar distintos factores. En los datos de crecimeinto de los dientes podemos considerar el suplemento alimenticio y la dosis. Para ello, basta con añadir una instrucción a la estética del boxplot (fig.3.11):

```
> p + geom_boxplot(aes(fill=factor(dose)))
```

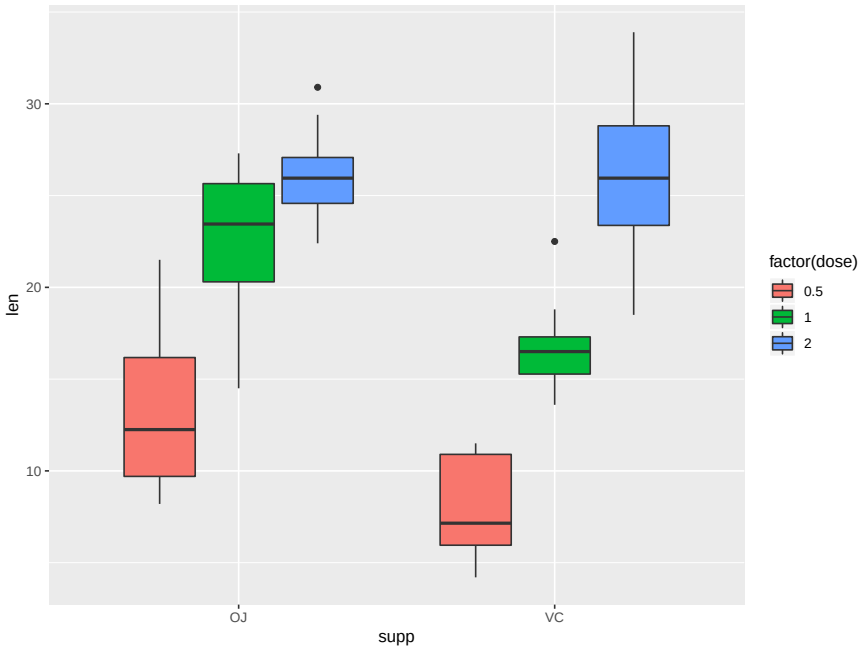

Figure 3.11: Longitud de los dientes en función del suplemento alimenticio

Evidentemente, podemos cambiar el orden de los factores si consideramos que es más interesante. Por ejemplo (fig.3.12):

```
> p <- ggplot(ToothGrowth,aes(factor(dose),len))
> p + geom_boxplot(aes(fill=factor(dose)))
```

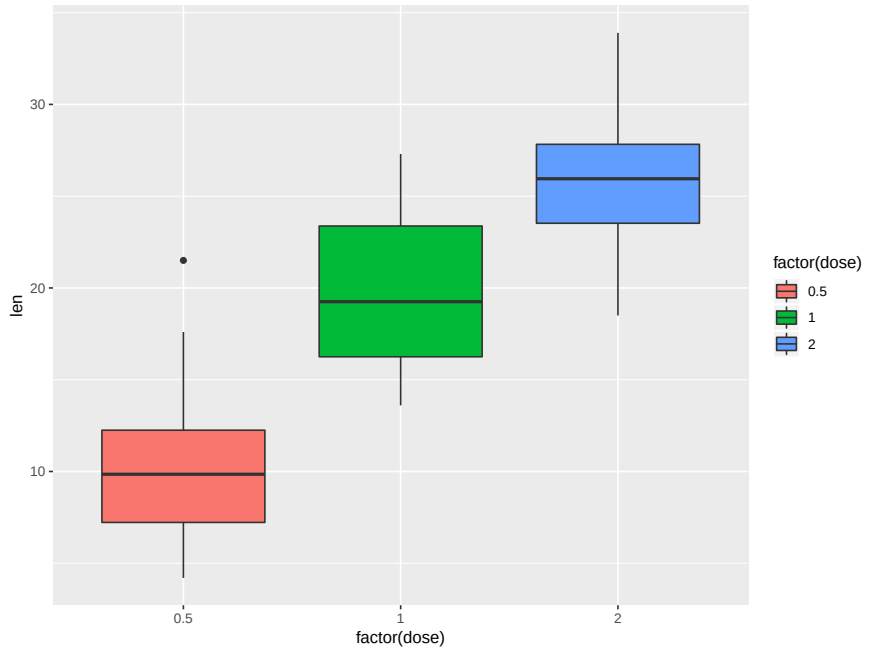

Figure 3.12: Longitud de los dientes en función del suplemento alimenticio

Ahora podemos subdividir por el suplemento alimenticio (fig.3.13):

```
> p + geom_boxplot(aes(fill=factor(supp)))
```

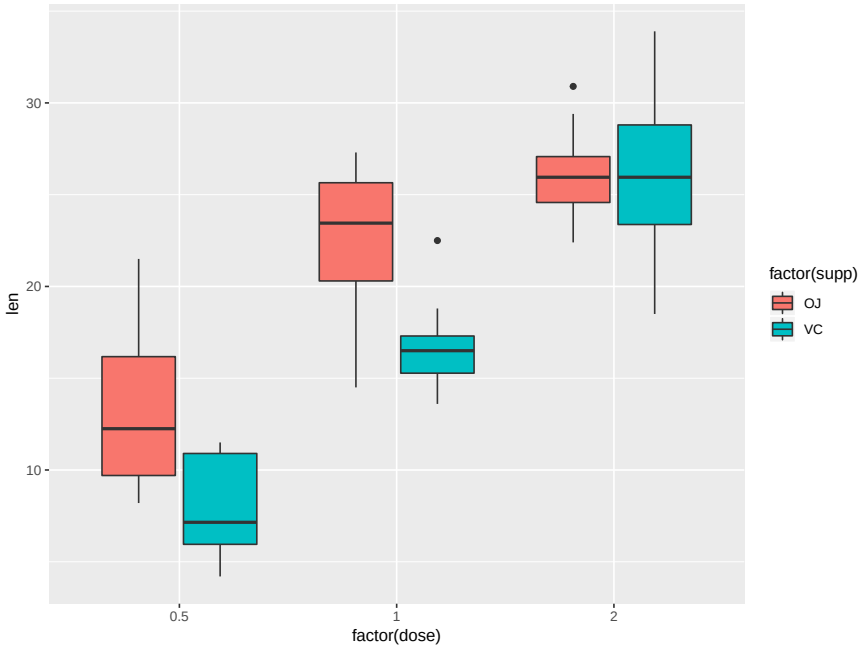

Figure 3.13: Longitud de los dientes en función del suplemento alimenticio

Podemos resaltar los **outliers** o datos extremos (fig.3.14):

```
> p + geom_boxplot(aes(fill=factor(supp)),
+          outlier.colour = "red", outlier.size = 3)
```

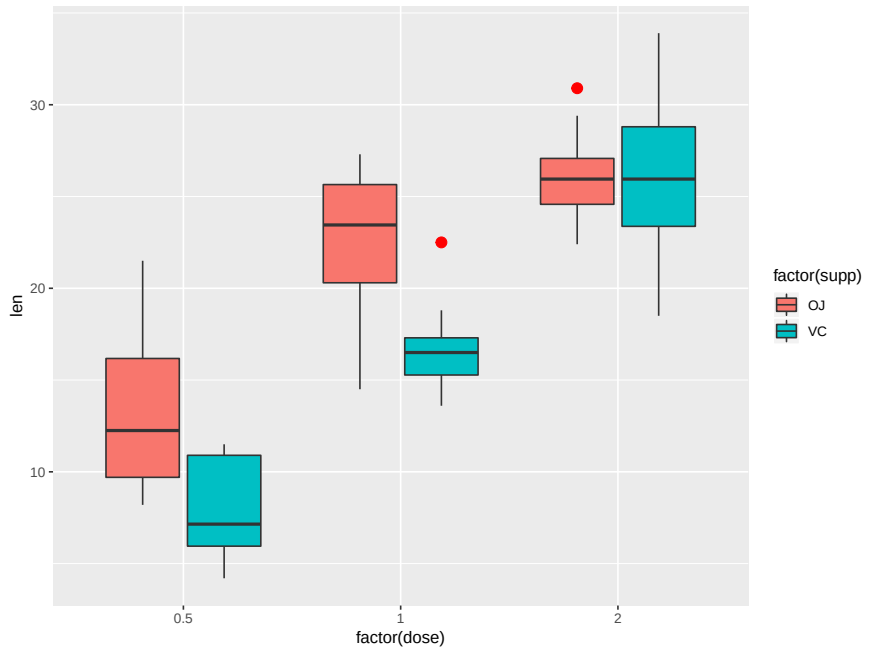

Figure 3.14: Longitud de los dientes en función del suplemento alimenticio

3.5 Gráficos por subgrupos

En muchos casos será necesario comparar resultados en distintos grupos. Algunos procedimientos, p.e. **geom_boxplot** ya incluyen esta posibilidad para dos grupos, pero podría ser necesario considerar más variables y subdividir las gráficas. El paquete **ggplot2** permite hacer esto mediante la instruccions **facets_grid** y sus variantes. Veamos como funciona con unos ejemplos. Para empezar, representaremos un boxplot separando las gráficas por suplemento alimenticio. Para ello, **facet_grid** especifica qué variable aparecerá en los subgrupos de filas(en este caso ninguna) y qué variable definira las columnas (en este caso el suplemento alimenticio) en una matriz de gráficos donde se mostrará el gráfico requerido para cada combinación de fila y comuna (fig.3.15):

```
> ggplot(ToothGrowth,aes(factor(dose),len)) +
+         geom_boxplot(aes(fill=factor(dose))) +
+         facet_grid(.~supp)
```

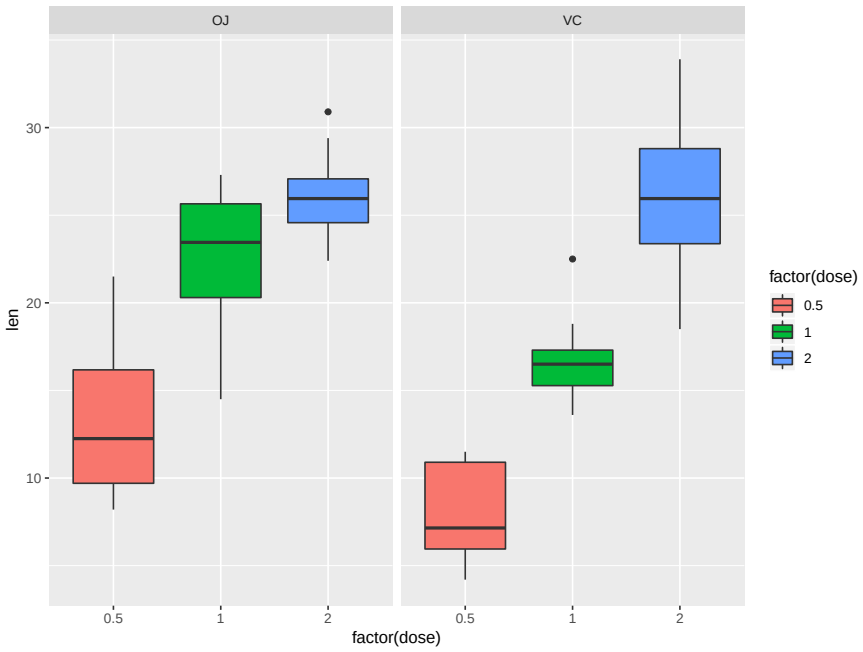

Figure 3.15: Uso de **facet_grid**

Para apreciar las posibilidades de esta opción, vamos a representar los histogramas de distribución de la longitud de dientes por dosis y suplemento alimenticio (fig.3.16):

```
> ggplot(ToothGrowth,aes(x=len)) +
+         geom_histogram(fill="white",color="black",binwidth=5) +
+         facet_grid(dose~supp)
```

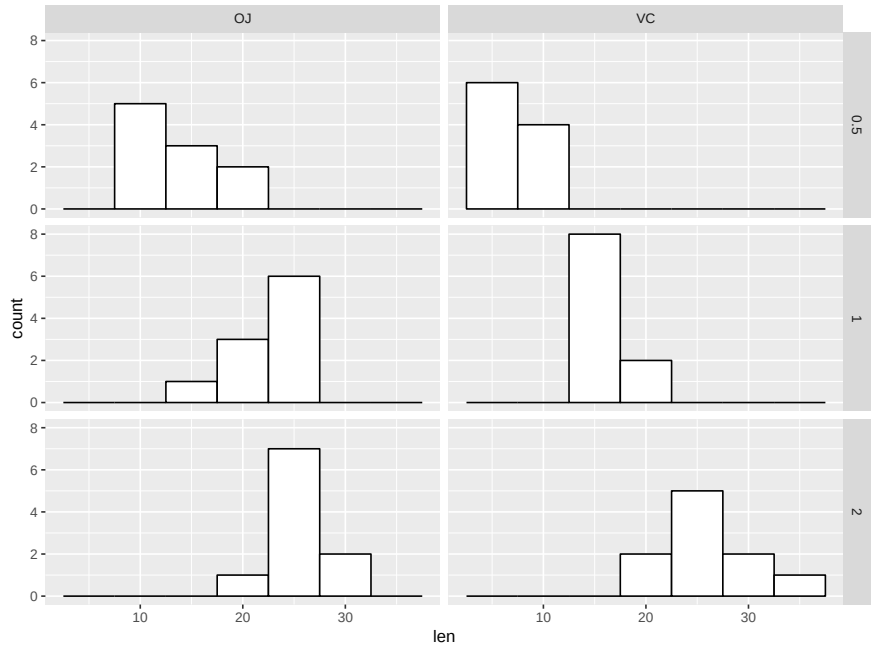

Figure 3.16: Uso de **facet_grid**

La opción de **facet_grid** puede utilizarse con cualquier tipo de gráficos (fig.3.17):

```
> p <- ggplot(mtcars,aes(x=wt,y=mpg))
> p + geom_point() +
+          facet_grid(.~cyl)
```

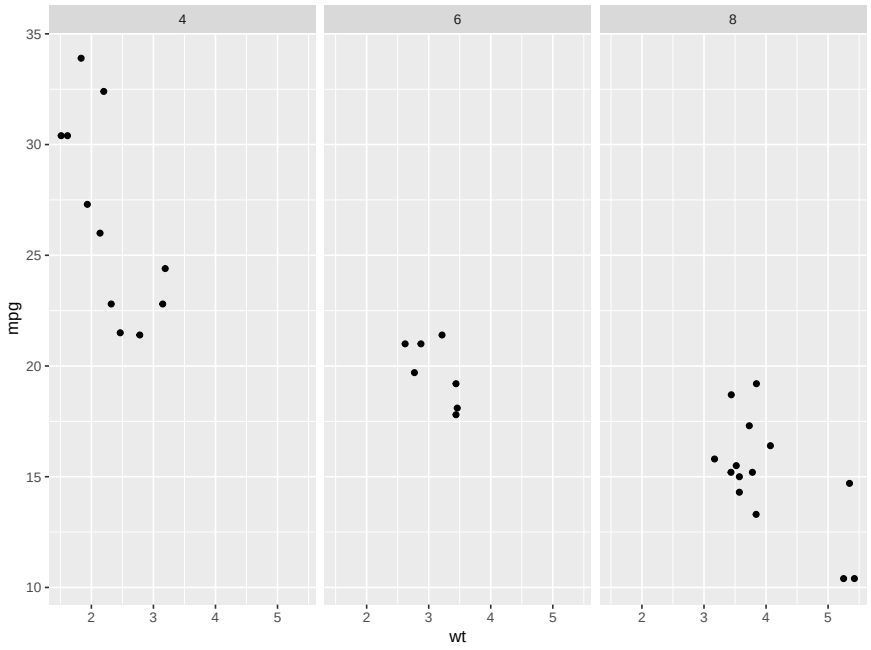

Figure 3.17: Uso de **facet_grid** con **geom_point**

Evidentemente, podemos utilizar más opciones. Por ejemplo, podemos incluir la recta de regresión en las observaciones de cada subgrupo (fig.3.18):

```
> p + geom_point() +
+          facet_grid(am~gear) +
+          geom_smooth(se=F,method="lm")
```

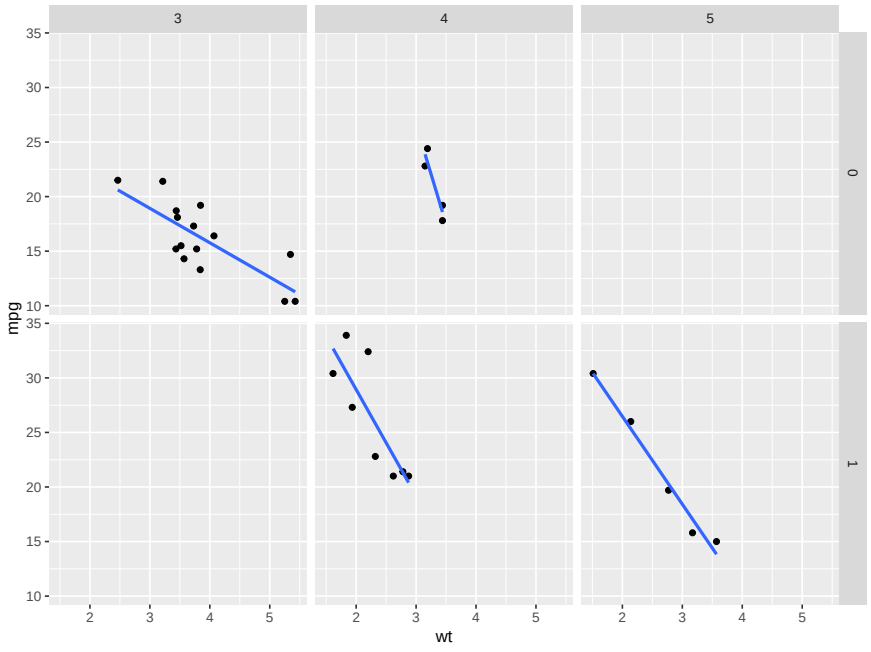

Figure 3.18: Uso de **facet_grid** con **geom_point**

3.6 Transformación de las escalas de los ejes

En algunos casos, será necesario transformas las escalas de los ejes para obtener unos resultados más claros. Veamos como puede hacerse. Consideremos los datos **mammals** que se encuentran en la libreria **MASS**:

```
> library(MASS)
> head(mammals)
```

```
                  body brain
Arctic fox        3.38  44.5
Owl monkey        0.48  15.5
Mountain beaver   1.35   8.1
Cow             465.00 423.0
Grey wolf        36.33 119.5
Goat             27.66 115.0
```

Si representamos el peso del cerebro respecto al peso del cuerpo, obtenemos (fig.3.19):

```
> p <- ggplot(mammals,aes(x=body,y=brain))
> p + geom_point()
```

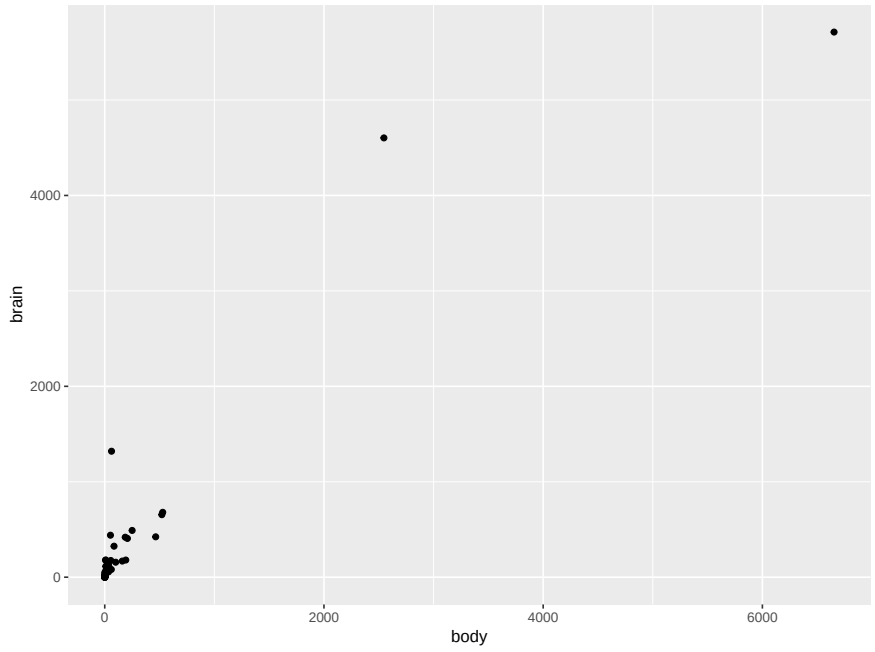

Figure 3.19: Uso de **facet_grid** con **geom_point**

Dado que existe una diferencia muy grande en el tamaño de los animales, puede ser más interesante representar estos datos en escala logarítmica (fig.3.20):

```
> p + geom_point()+coord_trans(x="log",y="log")
```

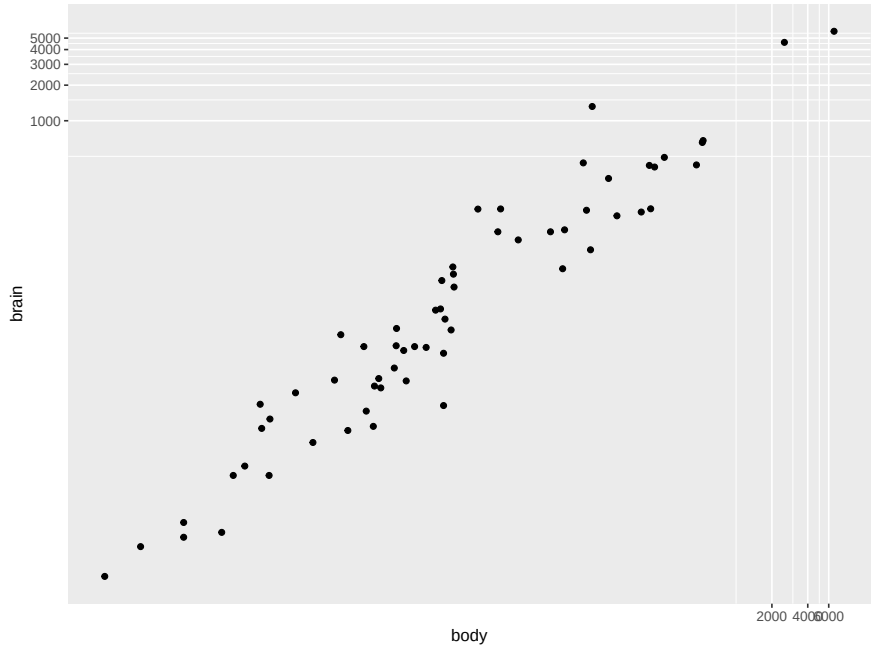

Figure 3.20: Uso de **facet_grid** con **geom_point**

3.7 Gráficos de barras

Los gráficos de barras son apropiados para representar las frecuencias absolutas de los valores de factores. Por ejemplo, la base de datos **birthwt** contiene información acerca del peso de recien nacidos en función de distintos características de la madre.

```
> head(birthwt)
```

```
   low age lwt race smoke ptl ht ui ftv  bwt
85   0  19 182    2     0   0  0  1   0 2523
86   0  33 155    3     0   0  0  0   3 2551
87   0  20 105    1     1   0  0  0   1 2557
88   0  21 108    1     1   0  0  1   2 2594
89   0  18 107    1     1   0  0  1   0 2600
91   0  21 124    3     0   0  0  0   0 2622
```

Podemos obtener una tabla de frecuencias que indique cuantos ninños nacen con poco peso en relación, por ejemplo, a la raza de la madre:

```
> with(birthwt,
+         table(race,low))
```

```
    low
race  0  1
   1 73 23
   2 15 11
   3 42 25
```

Podemos obtener la gráfica mediante el siguiente procedimiento (fig.3.21):

```
> ggplot(birthwt,aes(x=factor(low),fill=factor(race)))+
+         geom_bar(position=position_dodge())
```

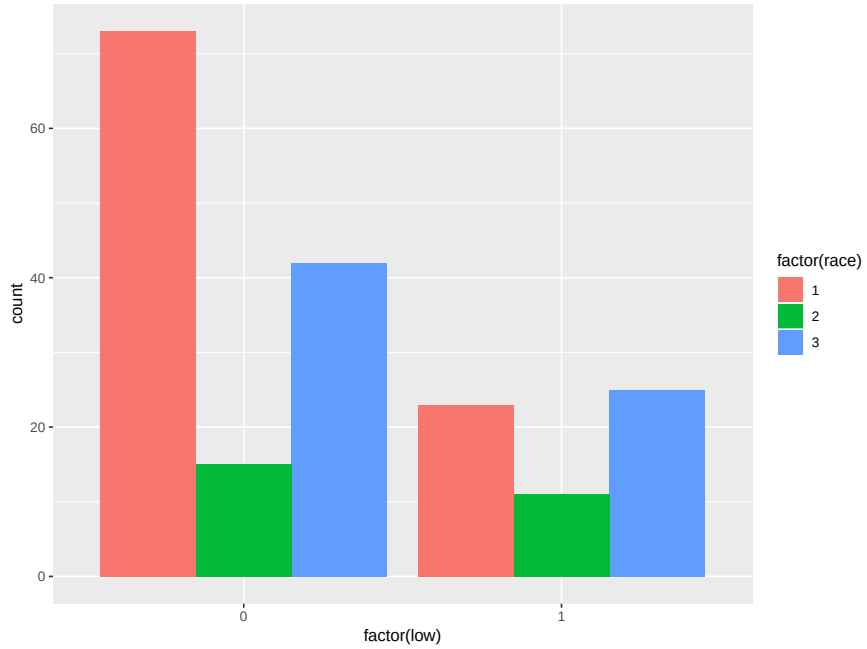

Figure 3.21: Uso de **geom_bar**

Podemos cambiar el color de las barras haciendo (fig.3.22):

```
> ggplot(birthwt,aes(x=factor(low),fill=factor(race)))+
+         geom_bar(position=position_dodge(),color="black") +
+         scale_fill_manual(values=c("#999999", "#E69F00", "#E70F00"))
```

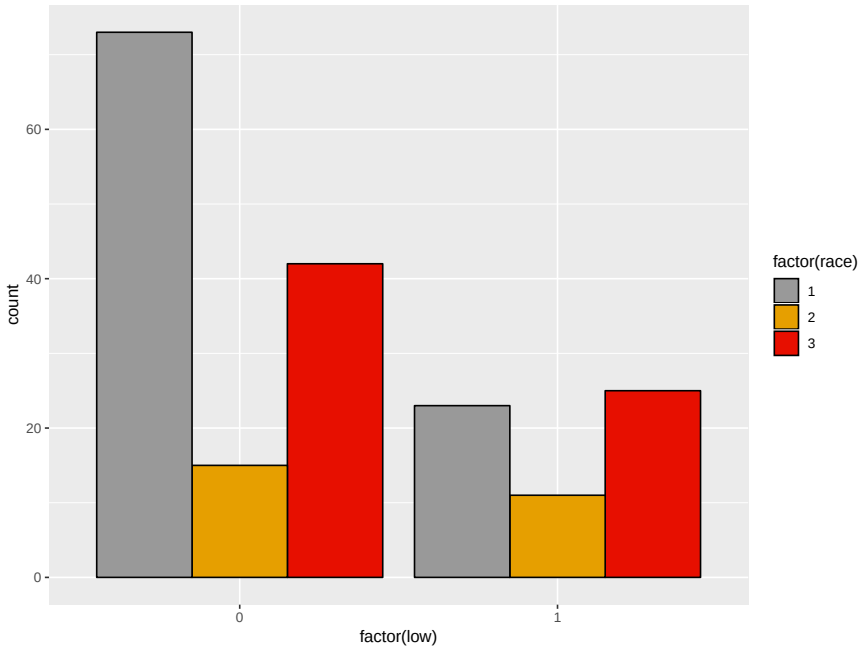

Figure 3.22: Uso de **geom_bar** y cambio de color

En general, es más interesante representar los porcentajes. En las tablas podemos hacer, por ejemplo, el procentage de bajo peso por raza:

```
> t <- with(birthwt,
+                          table(race,low))
> tp <- round(prop.table(t,1),1)
> tp
```

```
    low
race   0   1
   1 0.8 0.2
   2 0.6 0.4
   3 0.6 0.4
```

Para poder utilizar los resultados en **ggplot2**, debemos transformar la tabla a un **data.frame**:

```
> tp <- as.data.frame(round(prop.table(t,1),1))
> tp
```

```
  race low Freq
1    1   0  0.8
2    2   0  0.6
3    3   0  0.6
4    1   1  0.2
5    2   1  0.4
6    3   1  0.4
```

Ahora podemos representar estos porcentages (fig.3.23):

```
> ggplot(tp,aes(x=factor(race),fill=factor(low),Freq))+
+         geom_bar(stat="identity",
+                  position=position_dodge(),
+                  color="black")
```

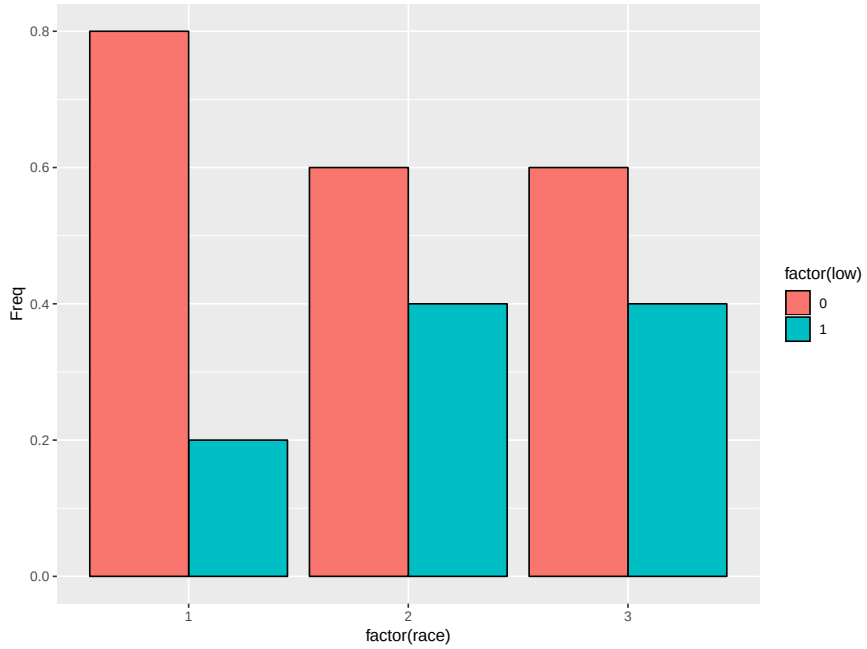

Figure 3.23: Gráfica de barras con porcentages por subgrupo

3.8 Gráficas de medias e intervalos de confianza

La obtención de gráficos de medias és algo complicada con **ggplot2**. Empezaremos, calulado los datos necesarios. En este caso, utilizamos los datos **ToothGrowth**:

```
> df <- ToothGrowth
> dfc <- df %>% group_by(supp,dose) %>%
+   summarise(m=mean(len),
+             lower=t.test(len)$conf.int[1],
+             upper=t.test(len)$conf.int[2])
> dfc
```

```
# A tibble: 6 x 5
# Groups:   supp [2]
  supp   dose     m lower upper
  <fct> <dbl> <dbl> <dbl> <dbl>
1 OJ      0.5 13.2  10.0  16.4
2 OJ      1   22.7  19.9  25.5
3 OJ      2   26.1  24.2  28.0
4 VC      0.5  7.98  6.02  9.94
5 VC      1   16.8  15.0  18.6
6 VC      2   26.1  22.7  29.6
```

![h]

```
> ggplot(dfc, aes(x=dose, y=m, colour=supp)) +
+     geom_errorbar(aes(ymin=lower, ymax=upper), width=.1,size=1) +
+     geom_line(size=1) +
+     geom_point(size=3)
```

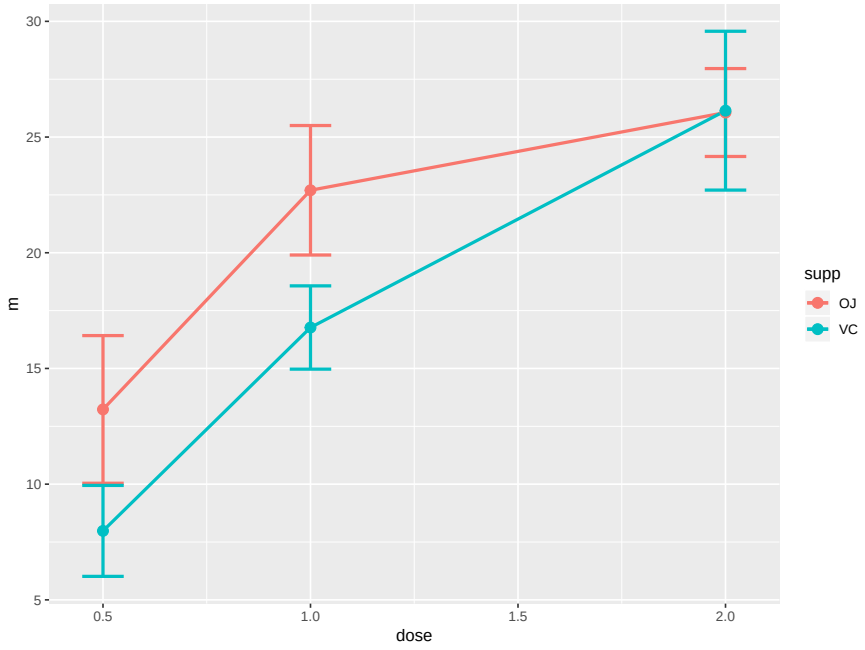

Figure 3.24: Media y su IC por subgrupos

Se puede apreciar que se ha creado un **data.frame** con la media de longitud por dosis y suplemento. Asímismo, se ha calculado el intervalo de confianza. Una vez calculados, podemos representarlos fácilmente (fig.3.24):

Un ejemplo que reune muchas de las opciones que pueden utilizarse seria (fig.3.25):

```
> pd <- position_dodge(.1)
> ggplot(dfc, aes(x=dose, y=m, colour=supp, group=supp)) +
+     geom_errorbar(aes(ymin=lower, ymax=upper),
+                   colour="black",
+                   width=.1,
+                   position=pd) +
+     geom_line(position=pd) +
+     geom_point(position=pd,
+                   size=3,
+                   shape=21,
+                   fill="white") +
+     xlab("Dose (mg)") +
+     ylab("Tooth length") +
+     scale_colour_hue(name="Supplement type", # Legend label, use darker colors
+                      breaks=c("OJ", "VC"),
+                      labels=c("Orange juice", "Ascorbic acid"),
+                      l=40) +  # Use darker colors, lightness=40
+     ggtitle("The Effect of Vitamin C on\nTooth Growth in Guinea Pigs") +
+     scale_y_continuous(limits=c(0, max(dfc$m + dfc$m-dfc$upper[2])), # Set y range
+                        breaks=0:20*4) + # Set tick every 4
+     theme_bw() +
+     theme(legend.justification=c(1,0),
+           legend.position=c(1,0)) # Position legend in bottom right
```

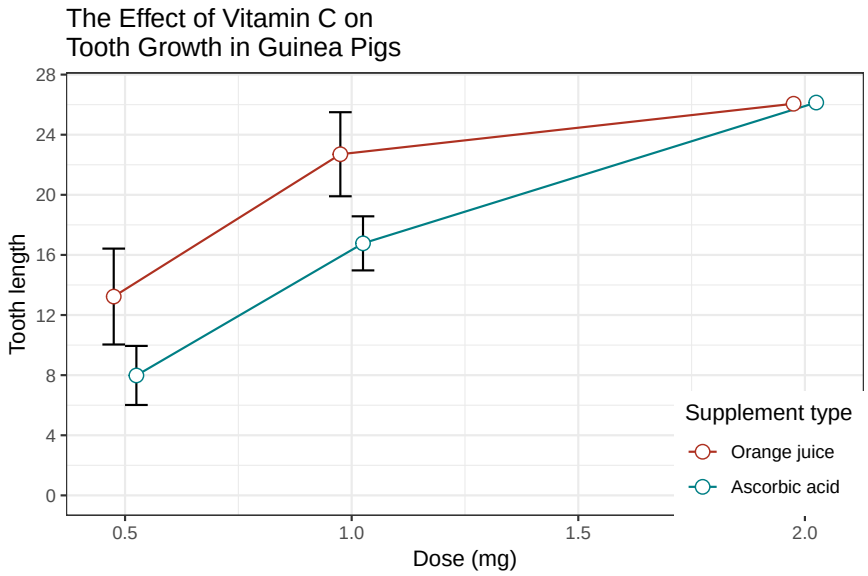

Figure 3.25: Media y su IC por subgrupos

En algunos casos, se prefiere representar las medias con barras. Podemos hacer lo siguiente. Primero, definiremos la dosis como un factor:

```
> dfc2 <- dfc
> dfc2$dose <- factor(dfc2$dose)
```

Ahora podemos obtener la gráfica haciendo(fig.3.26):

```
> ggplot(dfc2, aes(x=dose, y=m, fill=supp)) +
+     geom_bar(stat="identity",position=position_dodge(),color='black') +
+     geom_errorbar(aes( ymin=lower,ymax=upper),
+                   width=.2,                    # Width of the error bars
+                   position=position_dodge(.9))+
+     geom_point(aes(y=m),position=position_dodge(.9),size=4)
```

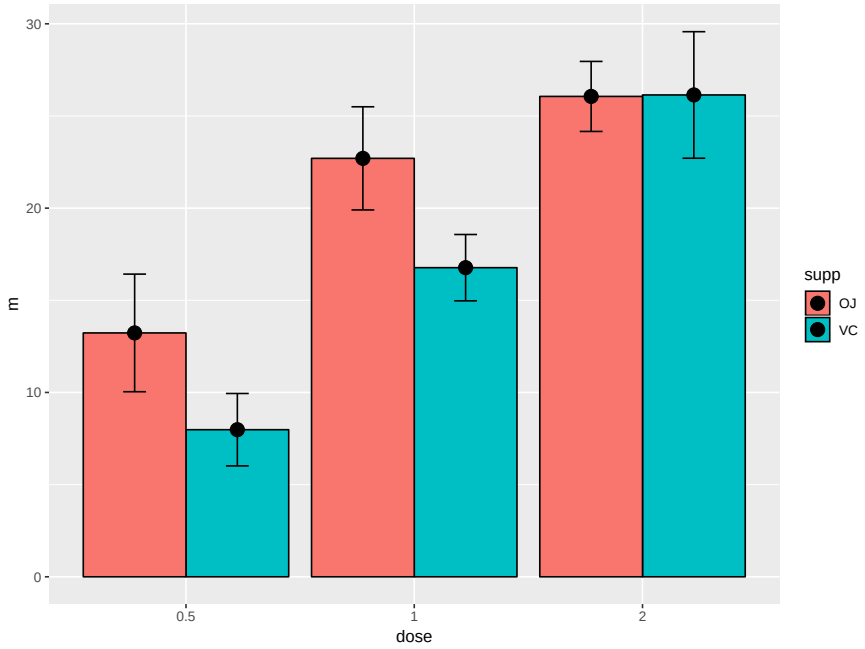

Figure 3.26: Media y su IC por subgrupos

Podemos acabar, con un ejemplo más elaborado (fig.3.27):

```
> ggplot(dfc2, aes(x=dose, y=m, fill=supp)) +
+     geom_bar(stat="identity",position=position_dodge(),
+              colour="black", # Use black outlines,
+              size=.3) +      # Thinner lines
+     geom_errorbar(aes(ymin=lower, ymax=upper),
+                   size=.3,    # Thinner lines
+                   width=.2,
+                   position=position_dodge(.9)) +
+     geom_point(aes(y=m),position=position_dodge(.9),size=4)+
+     xlab("Dose (mg)") +
+     ylab("Tooth length") +
+     scale_fill_hue(name="Supplement type", # Legend label, use darker colors
+                    breaks=c("OJ", "VC"),
+                    labels=c("Orange juice", "Ascorbic acid")) +
+     ggtitle("The Effect of Vitamin C on\nTooth Growth in Guinea Pigs") +
+     scale_y_continuous(breaks=0:20*4) +
+     theme_bw()
```

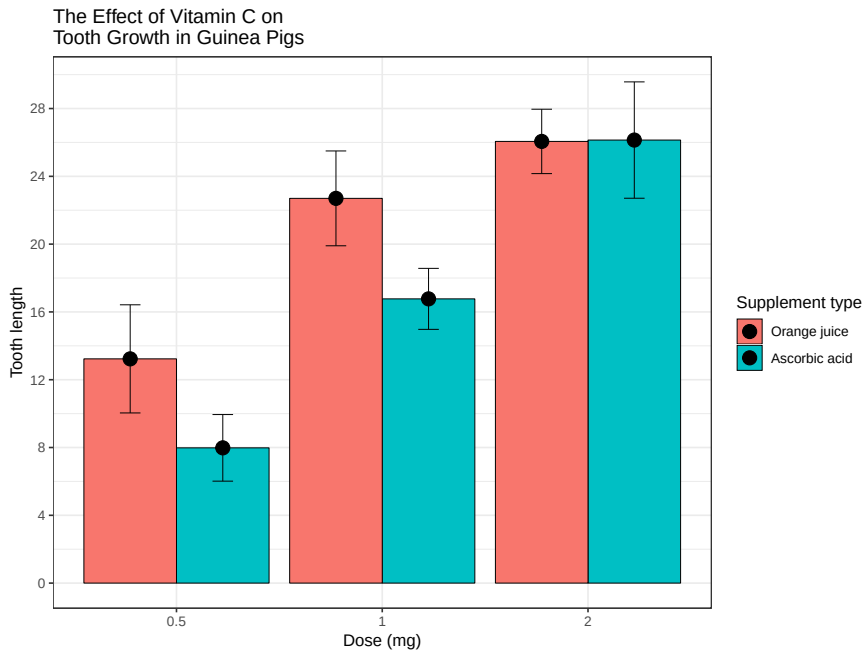

Figure 3.27: Media y su IC por subgrupos

Índice