GRAU EN ENGINYERIA DE DADES

104365 Visualització de Dades

Seminari 2. Comparacions i Distribucions

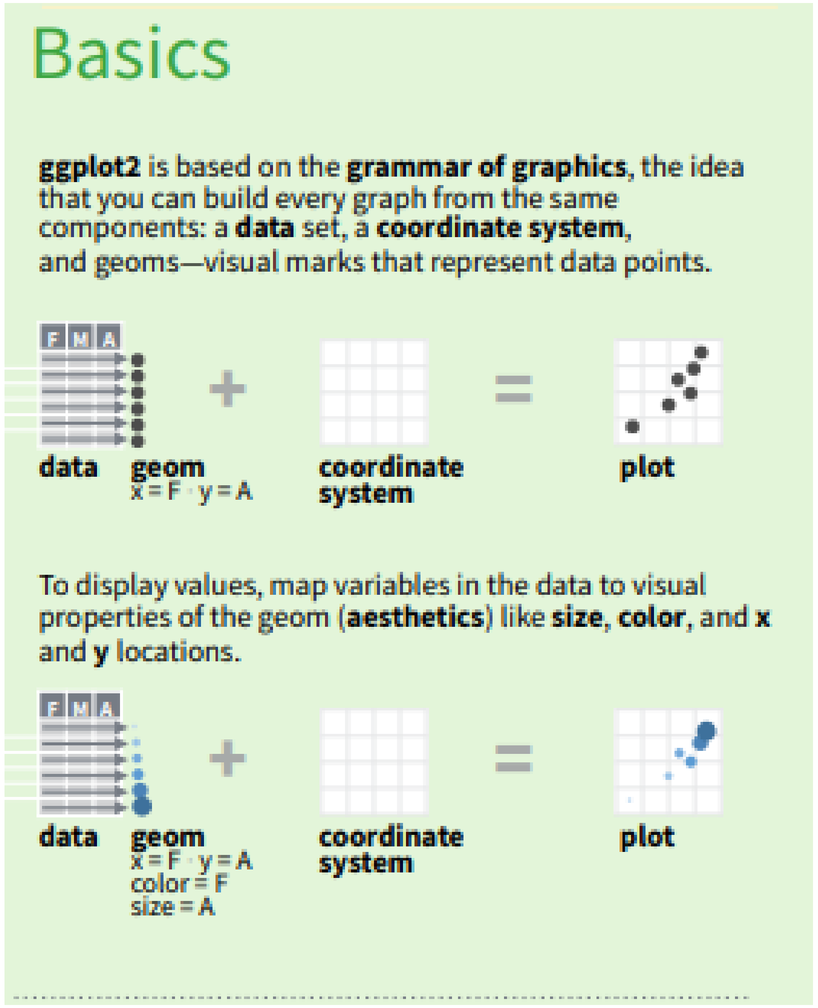
Departament de Matemàtiques

Escola d'Enginyeria UAB

UAB

Index

1.1. Quick summary: ggplot2 Basics



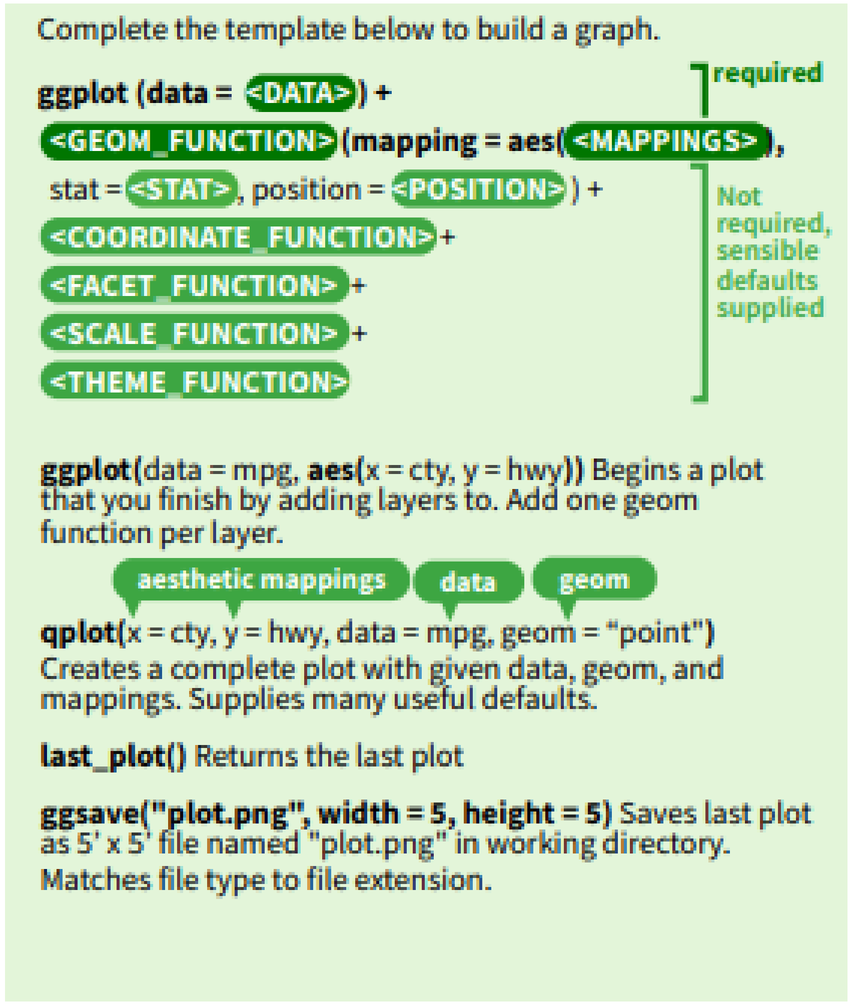
Basics

ggplot2 is based on the **grammar of graphics**, the idea that you can build every graph from the same components: a **data** set, a **coordinate system**, and geoms—visual marks that represent data points.

F M A
+
=
data
geom
x = F · y = A
coordinate system
plot

To display values, map variables in the data to visual properties of the geom (**aesthetics**) like **size**, **color**, and **x** and **y** locations.

F M A
+
=
data
geom
x = F · y = A
color = F
size = A
coordinate system
plot



Complete the template below to build a graph.

```
ggplot (data = <DATA>) +
  <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>),
  stat = <STAT>, position = <POSITION>) +
  <COORDINATE_FUNCTION> +
  <FACET_FUNCTION> +
  <SCALE_FUNCTION> +
  <THEME_FUNCTION>
```

required
Not required, sensible defaults supplied

ggplot(data = mpg, **aes**(x = cty, y = hwy)) Begins a plot that you finish by adding layers to. Add one geom function per layer.

aesthetic mappings
data
geom

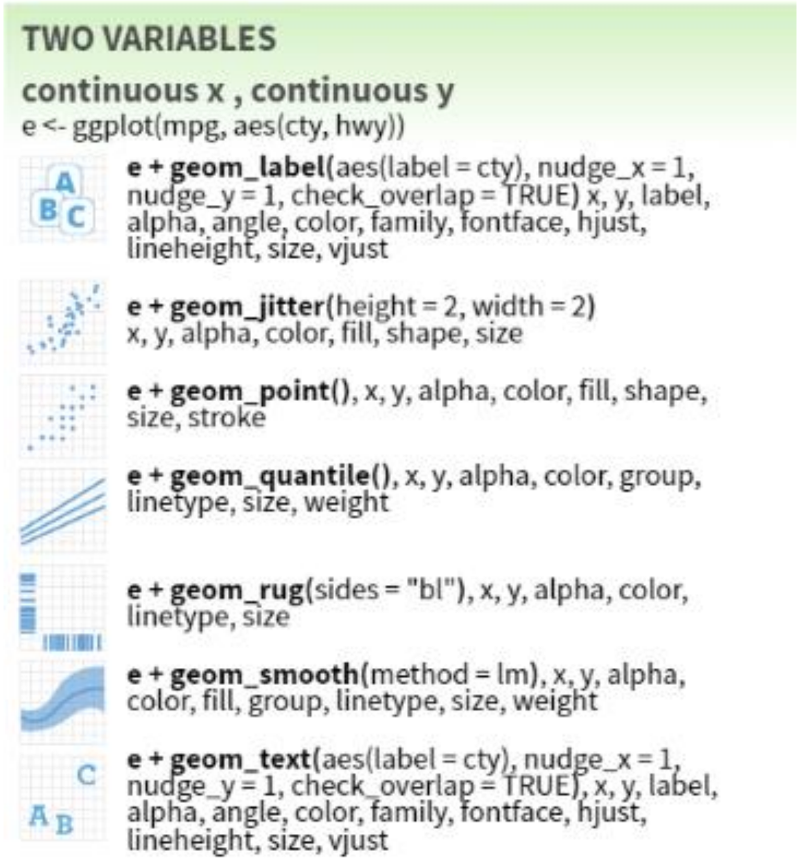
qplot(x = cty, y = hwy, data = mpg, geom = "point") Creates a complete plot with given data, geom, and mappings. Supplies many useful defaults.

last_plot() Returns the last plot

ggsave("plot.png", width = 5, height = 5) Saves last plot as 5' x 5' file named "plot.png" in working directory. Matches file type to file extension.

https://github.com/rstudio/cheatsheets/blob/master/data-visualization-2.1.pdf

1.1. Quick summary: **ggplot2 Basics**



TWO VARIABLES

continuous x , continuous y
e <- ggplot(mpg, aes(cty, hwy))

e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE) x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust

e + geom_jitter(height = 2, width = 2) x, y, alpha, color, fill, shape, size

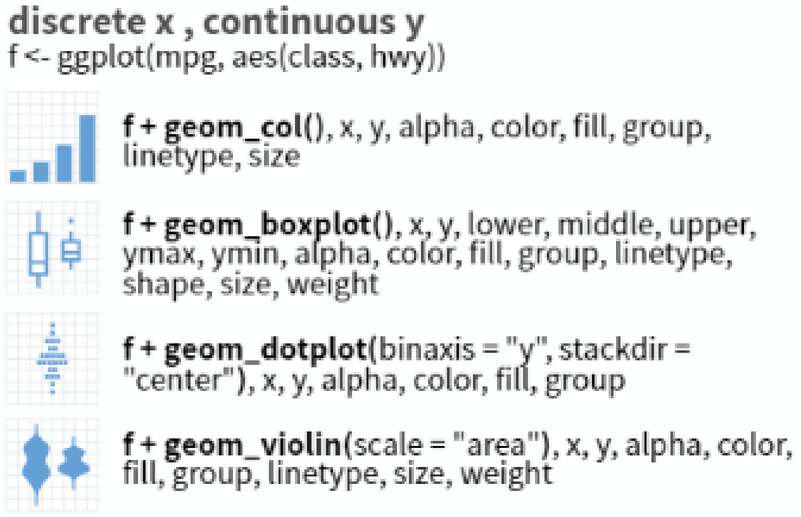
e + geom_point(), x, y, alpha, color, fill, shape, size, stroke

e + geom_quantile(), x, y, alpha, color, group, linetype, size, weight

e + geom_rug(sides = "bl"), x, y, alpha, color, linetype, size

e + geom_smooth(method = lm), x, y, alpha, color, fill, group, linetype, size, weight

e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE), x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, vjust



discrete x , continuous y
f <- ggplot(mpg, aes(class, hwy))

f + geom_col(), x, y, alpha, color, fill, group, linetype, size

f + geom_boxplot(), x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

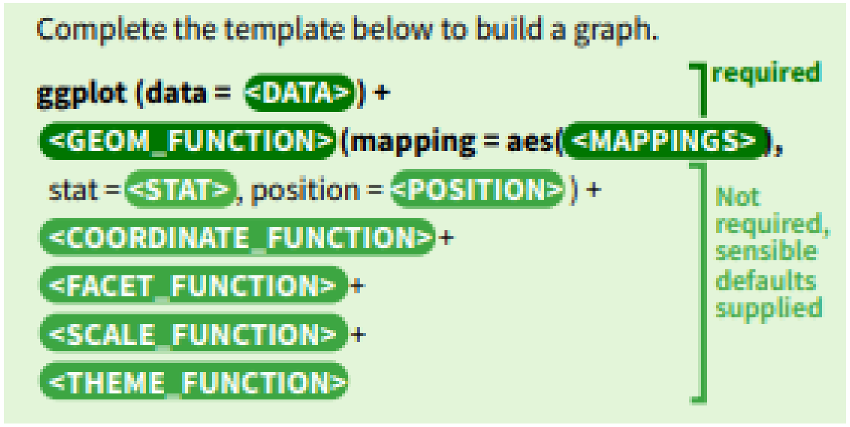
f + geom_dotplot(binaxis = "y", stackdir = "center"), x, y, alpha, color, fill, group

f + geom_violin(scale = "area"), x, y, alpha, color, fill, group, linetype, size, weight

https://github.com/rstudio/cheatsheets/blob/master/data-visualization-2.1.pdf

1.2. Quick summary: **Type of variables**

- **Numerical variables:**
  - **Continuous**
  - **Discrete**



Complete the template below to build a graph.

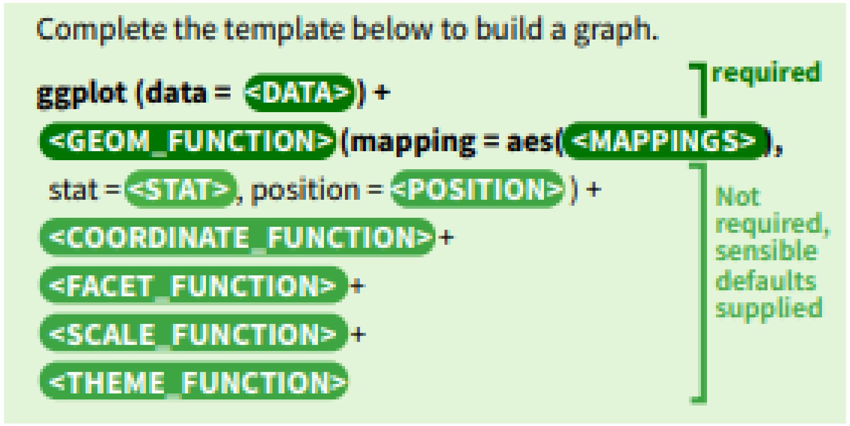
```
ggplot (data = <DATA>) +
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>),
 stat = <STAT>, position = <POSITION>) +
<COORDINATE_FUNCTION> +
<FACET_FUNCTION> +
<SCALE_FUNCTION> +
<THEME_FUNCTION>
```

required
Not required, sensible defaults supplied

! In R the term 'category' and 'enumerated type' are used for factors.

1.2. Quick summary: **Type of variables**

- **Numerical variables:**
  - **Continuous**
  - **Discrete**


Complete the template below to build a graph.

```
ggplot (data = <DATA>) +
<GEOM_FUNCTION>(mapping = aes(<MAPPINGS>),
 stat = <STAT>, position = <POSITION>) +
<COORDINATE_FUNCTION> +
<FACET_FUNCTION> +
<SCALE_FUNCTION> +
<THEME_FUNCTION>
```

aes	Discreta	Contínua
Color (*color*)	Arco iris de colors	Gradient de colors
Forma (*shape*)	Diferent formes	NO APLICA
Talla (*size*)	Escala discreta de talles	Mapeig lineal entre àrea i el valor
Transparència (*alpha*)	NO APLICA	Mapeig lineal a la transparència

Last day

Escola d'Enginyeria UAB

UAB

2. Factorial variables of R: nominal/ordinal

- **For us, the numerical variables:**
    - **Categoric (nominals)**
    - **Ordinal**

!!! **Factorial variables** in R:

- **nominal variables**. Do not imply any order. Example: i) sex.
- **ordinal variables**. They imply order or gradation. Examples: i) The height of a person could be categorized (as short, medium, tall). ii) the example of the cylinders of a car from last day (three groups of cars: 4 cylinders, 6 cylinders, 8 cylinders).
- !! We also saw it for logical variables (such as 'vs') – nominal V/S

!! The last day, we used **factor to categorize variables. However, if we want to use categoric variable with order. In R, `ordered` `allow` us this**

https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/factor

Escola d'Enginyeria UAB

UAB

2. Factorial variables of R: nominal/ordinal

!!! **Factorial variables** in R. You can use:

- (Factor) **nominal variables**. Do not imply any order. Example: i) sex, ii) a logical variable (we can add labels).
- (Ordered) **ordinal variables**. They imply order or gradation. Examples: i) The height of a person could be categorized (as short, medium, tall). ii) the example of the cylinders of a car from last day (three groups of cars: 4 cylinders, 6 cylinders, 8 cylinders).

https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/factor

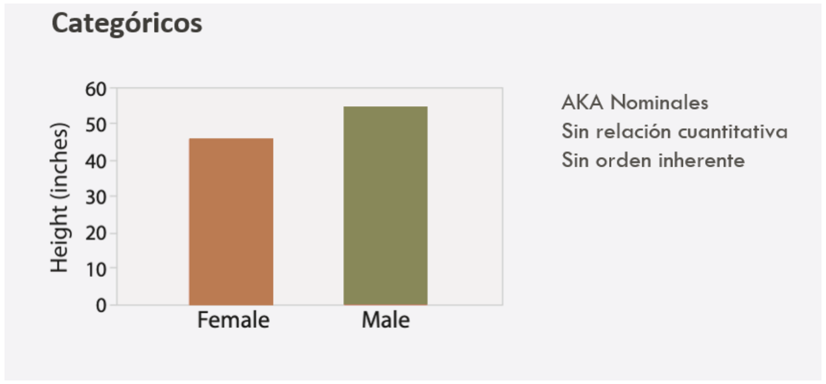
```
➢ mtcars2 <- within(mtcars, { vs <- factor(vs, labels = c("V", "S"))
am <- factor(am, labels = c("automatic", "manual"))
cyl <- ordered(cyl)
gear <- ordered(gear)
carb <- ordered(carb) })
```

Escola d'Enginyeria UAB

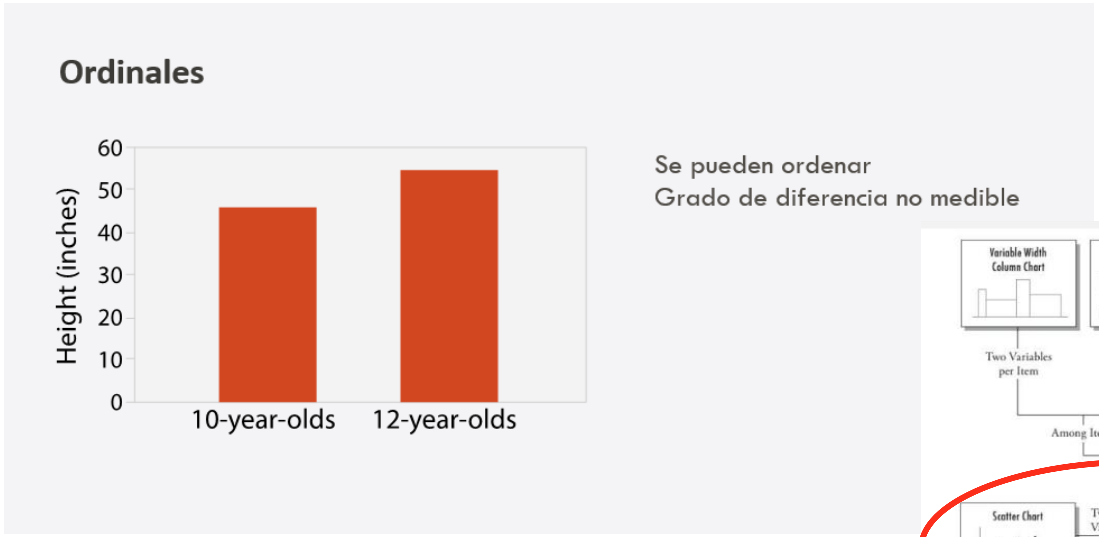
UAB

3. Comparisons

- **Numerical variables:**
  - **Categoric**
  - **Ordinal**

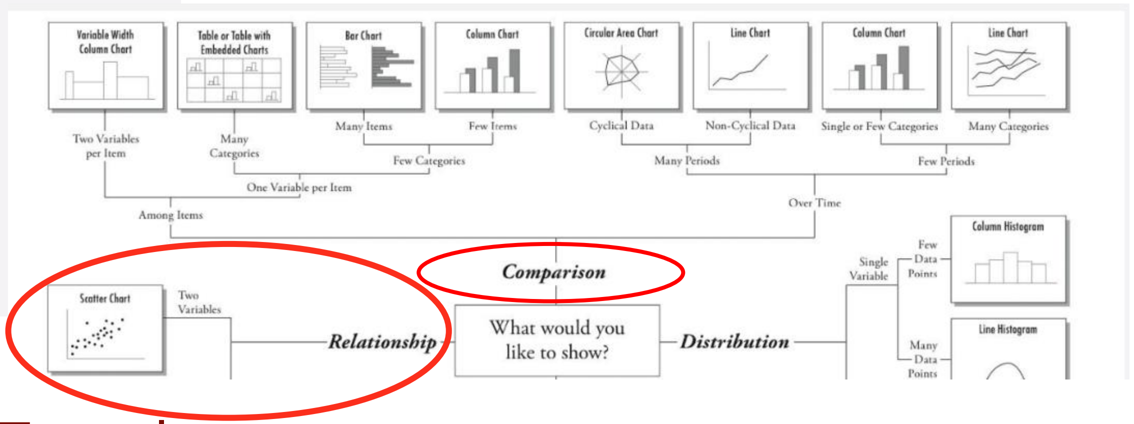


AKA Nominales
Sin relación cuantitativa
Sin orden inherente



Se pueden ordenar
Grado de diferencia no medible

You saw it with Guillermo



To explore

3. Comparisons: Bar charts

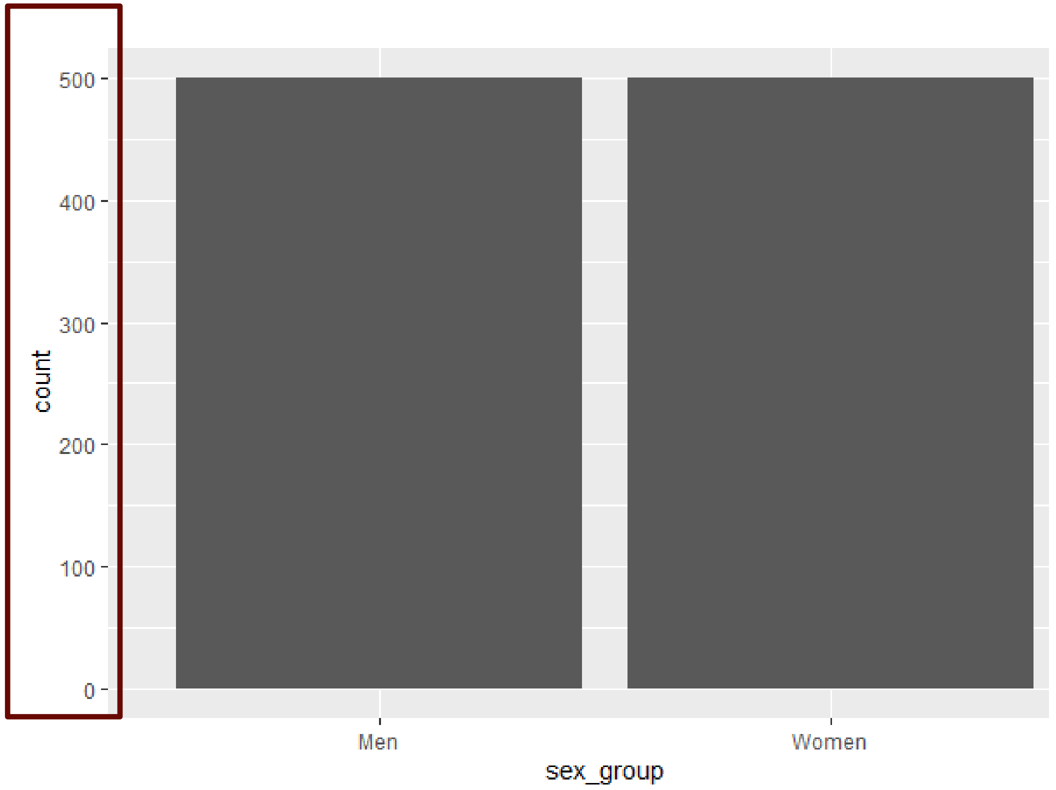
- **Two types of bar charts:**
  - **`geom_bar()`:** It makes the **height** of the bar **proportional to the number of cases in each group**.
    - It does a default statistical transformation. It uses **`stat_count()`** by default: it counts the number of cases at each x position.
  - **`geom_col()`:** It represents **values in the data as the heights of the bars** (it just treats the data as is)
    - It uses **stat_identity()**: it leaves the data as is.

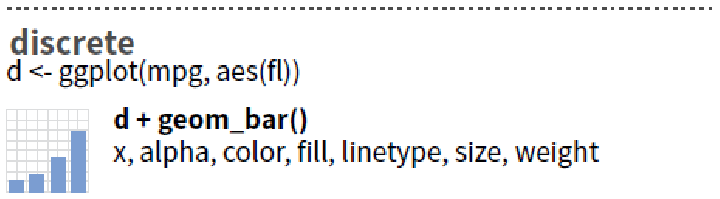
Useful for one discrete variable.

If we have two variables → useful when one is discrete.

3. Comparisons: Bar charts – ggplot2 Example

```
> ggplot(data=data_example) + aes(sex_group) + geom_bar()
```



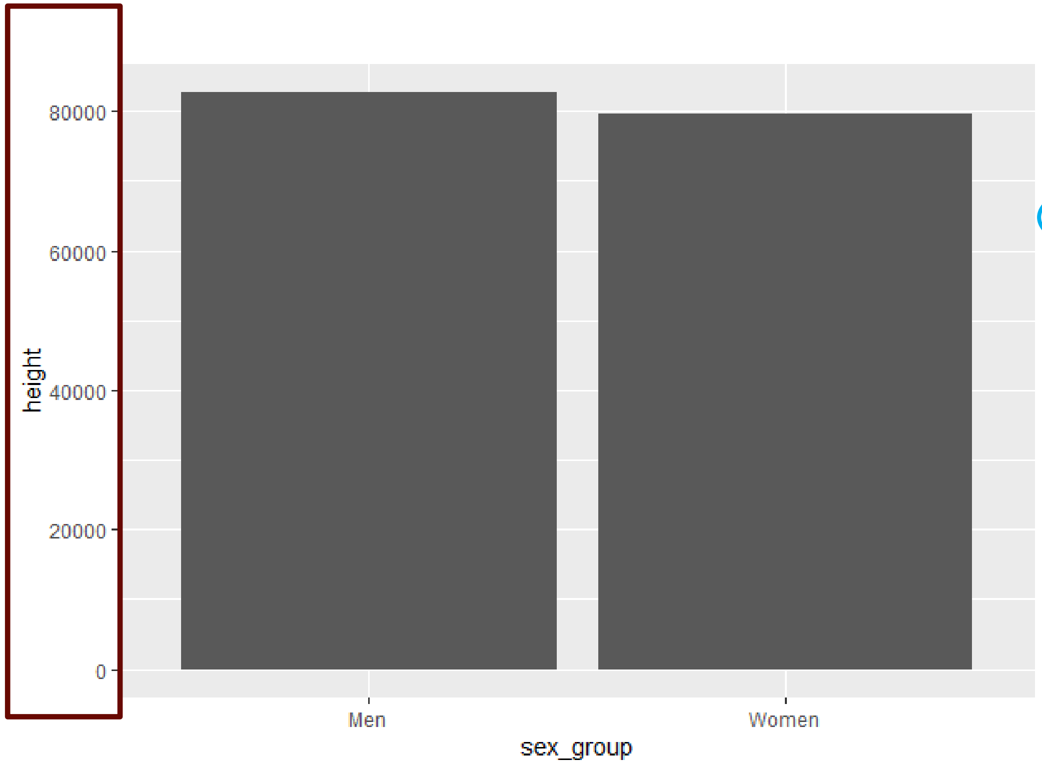


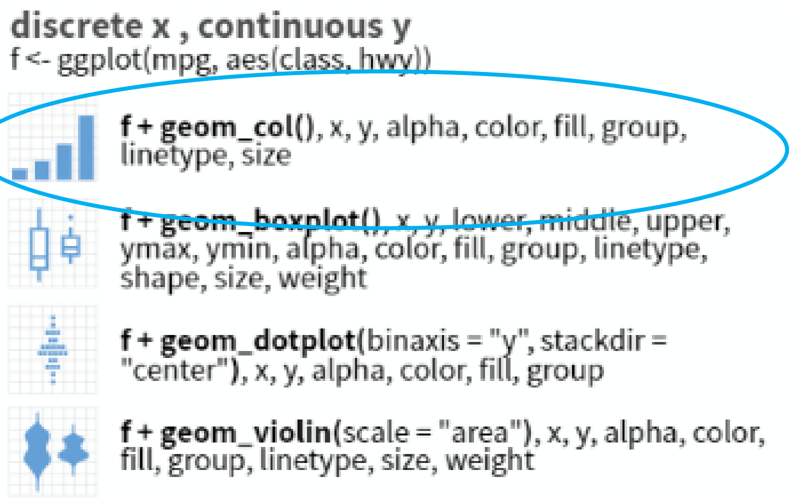
```
d <- ggplot(mpg, aes(fl))
d + geom_bar()
x, alpha, color, fill, linetype, size, weight
```

Data_example contains the heights of **two groups with the same number of men & women**

3. Comparisons: Bar charts – ggplot2 Example

```
> ggplot(data=data_example) + aes(x=sex_group, y=height) + geom_col()
```





discrete x , continuous y
f <- ggplot(mpg, aes(class, hwy))

f + geom_col(), x, y, alpha, color, fill, group, linetype, size

f + geom_boxplot(), x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

f + geom_dotplot(binaxis = "y", stackdir = "center"), x, y, alpha, color, fill, group

f + geom_violin(scale = "area"), x, y, alpha, color, fill, group, linetype, size, weight

Data_example contains the heights of two groups with the same number of men & women

Escola d'Enginyeria UAB

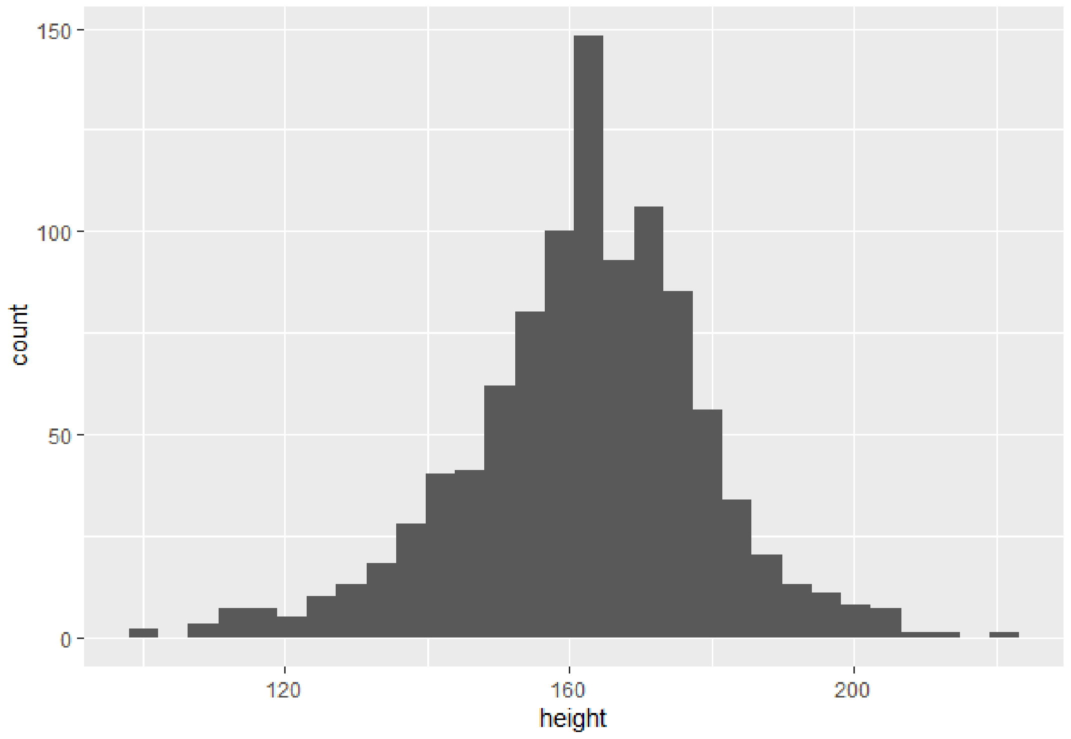
UAB

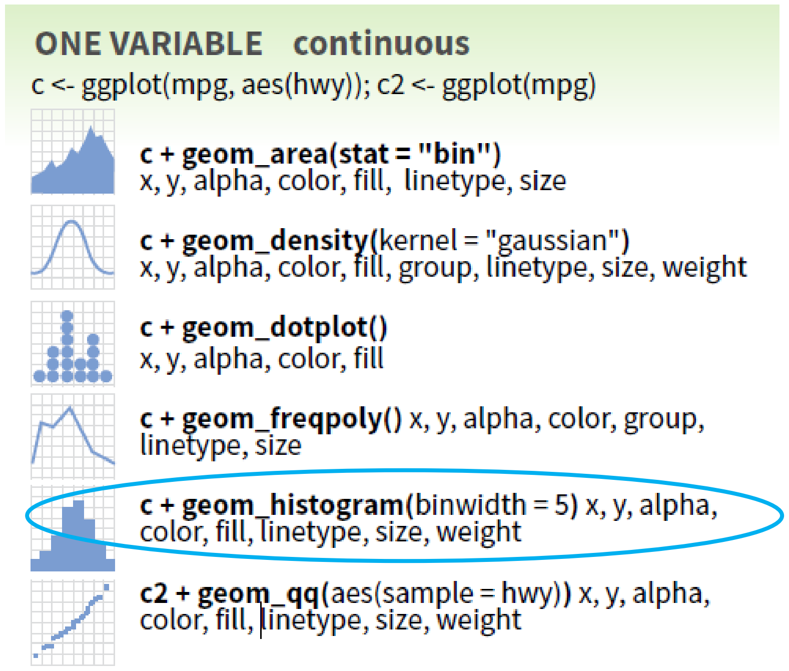
4.1. Distributions: Histograms

- **To visualize the distribution of a single continuous variable by dividing the x axis into bins and counting the number of observations in each bin**
    - **`geom_histogram():`** Display the **counts with bars**. Default `bindwith=30`, we need to specify this argument
    - **`geom_freqpoly():`** Display the **counts with lines**. They are *more suitable when you want to compare the distribution across the levels of a categorical variable*

4.1. Distributions: Histograms – ggplot2 Example

```
> ggplot(data=data_example) + aes(height) + geom_histogram()
```




ONE VARIABLE continuous

c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)

c + geom_area(stat = "bin")
x, y, alpha, color, fill, linetype, size

c + geom_density(kernel = "gaussian")
x, y, alpha, color, fill, group, linetype, size, weight

c + geom_dotplot()
x, y, alpha, color, fill

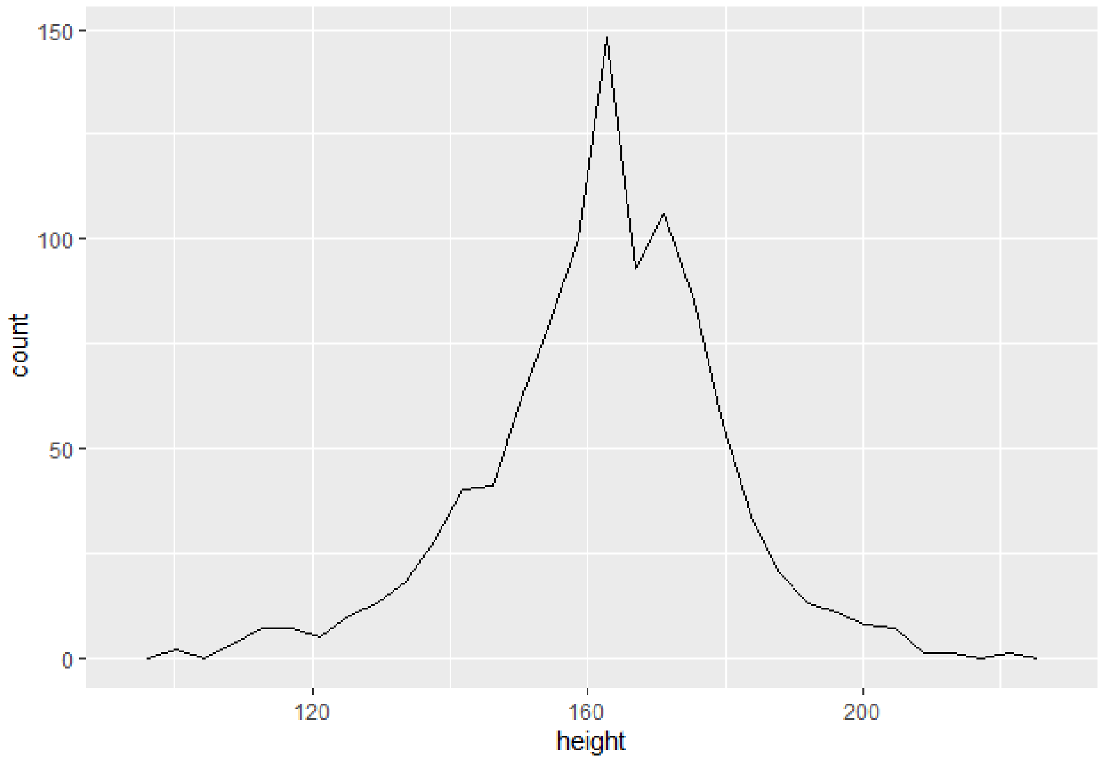
c + geom_freqpoly() x, y, alpha, color, group, linetype, size

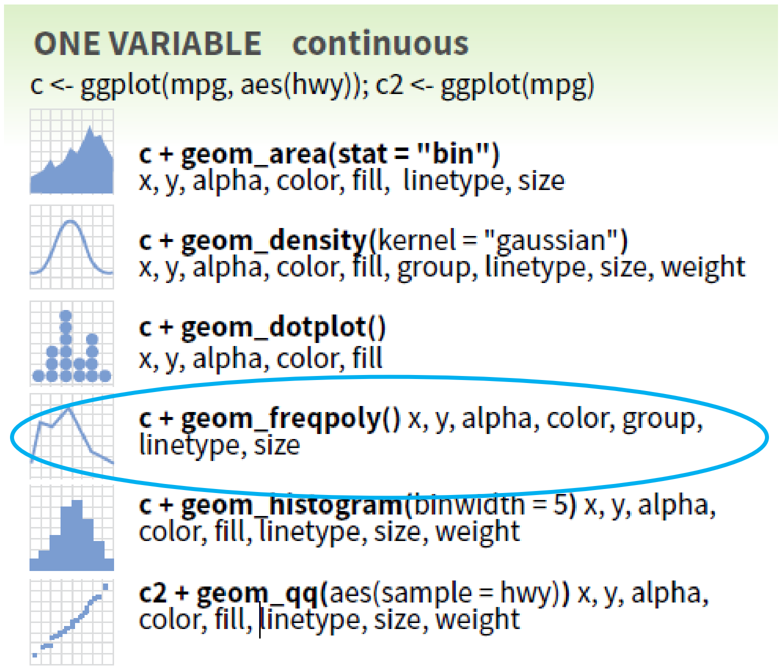
c + geom_histogram(binwidth = 5) x, y, alpha, color, fill, linetype, size, weight

c2 + geom_qq(aes(sample = hwy)) x, y, alpha, color, fill, linetype, size, weight

4.1. Distributions: Poligon de freqüencies – ggplot2 Example

```
> ggplot(data=data_example) + aes(height) + geom_freqpoly()
```





ONE VARIABLE continuous

c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)

c + geom_area(stat = "bin")
x, y, alpha, color, fill, linetype, size

c + geom_density(kernel = "gaussian")
x, y, alpha, color, fill, group, linetype, size, weight

c + geom_dotplot()
x, y, alpha, color, fill

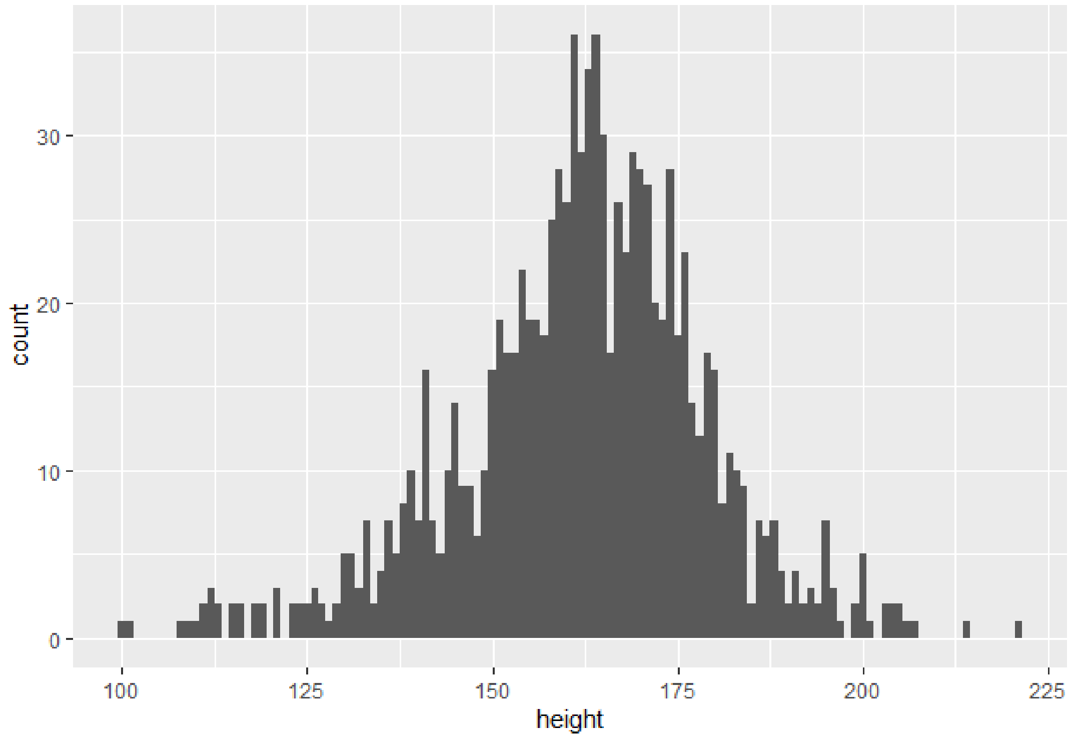
c + geom_freqpoly() x, y, alpha, color, group, linetype, size

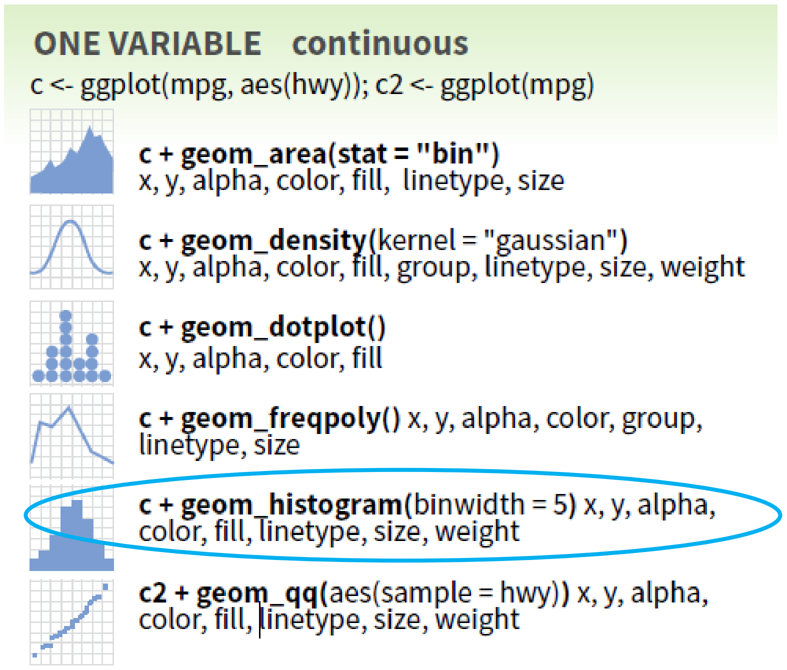
c + geom_histogram(binwidth = 5) x, y, alpha, color, fill, linetype, size, weight

c2 + geom_qq(aes(sample = hwy)) x, y, alpha, color, fill, linetype, size, weight

4.1. Distributions: Histograms – ggplot2 Example

```
> ggplot(data=data_example) + aes(height) + geom_histogram(binwidth=1)
```





ONE VARIABLE continuous

c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)

c + geom_area(stat = "bin")
x, y, alpha, color, fill, linetype, size

c + geom_density(kernel = "gaussian")
x, y, alpha, color, fill, group, linetype, size, weight

c + geom_dotplot()
x, y, alpha, color, fill

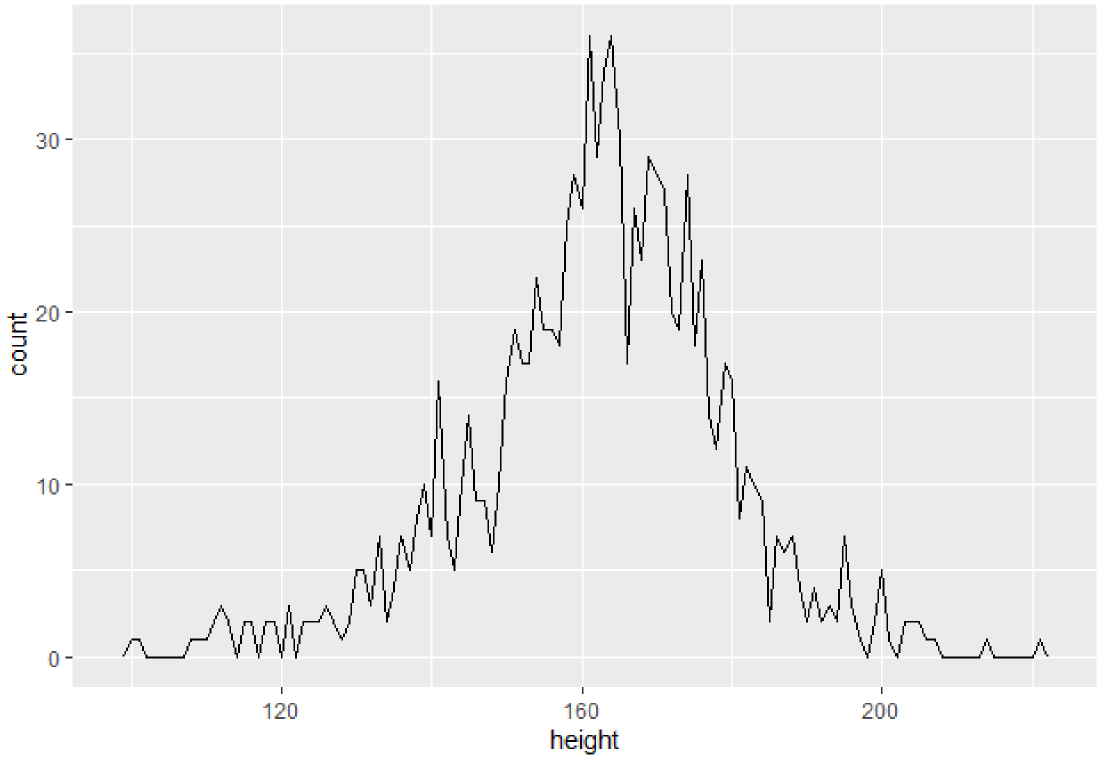
c + geom_freqpoly() x, y, alpha, color, group, linetype, size

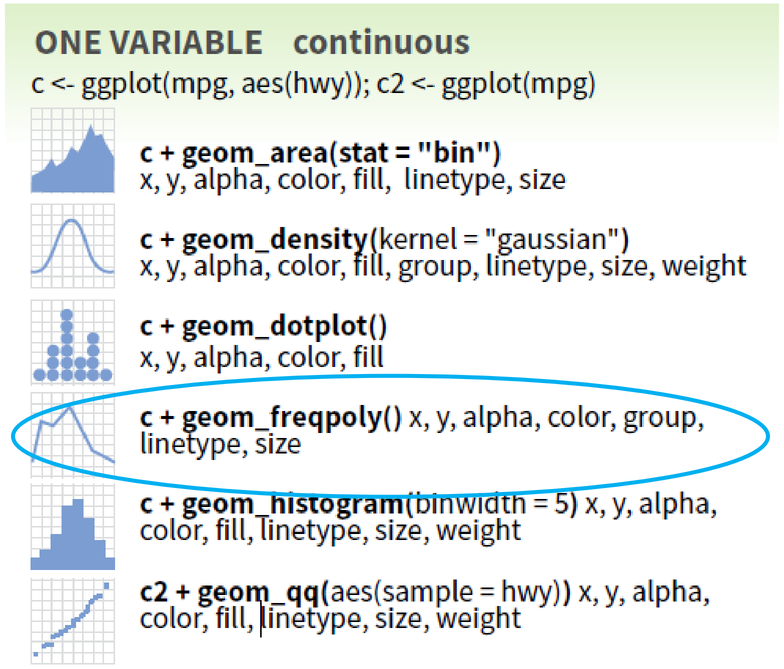
c + geom_histogram(binwidth = 5) x, y, alpha, color, fill, linetype, size, weight

c2 + geom_qq(aes(sample = hwy)) x, y, alpha, color, fill, linetype, size, weight

4.1. Distributions: Poligon de freqüencies – ggplot2 Example

```
> ggplot(data=data_example) + aes(height) + geom_freqpoly(binwidth=1)
```





ONE VARIABLE continuous

c <- ggplot(mpg, aes(hwy)); c2 <- ggplot(mpg)

c + geom_area(stat = "bin")
x, y, alpha, color, fill, linetype, size

c + geom_density(kernel = "gaussian")
x, y, alpha, color, fill, group, linetype, size, weight

c + geom_dotplot()
x, y, alpha, color, fill

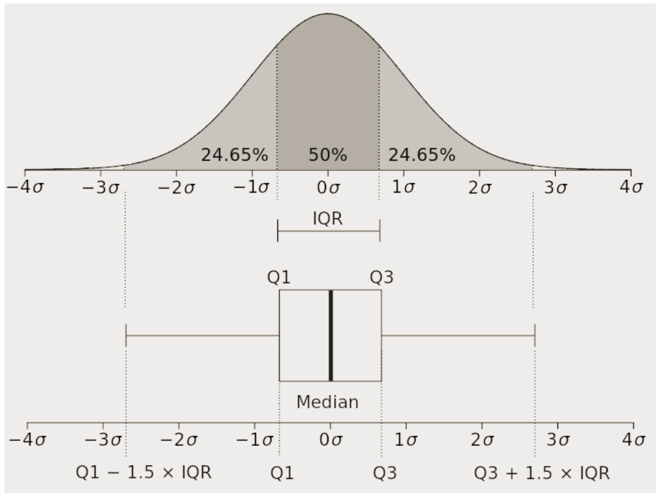
c + geom_freqpoly() x, y, alpha, color, group, linetype, size

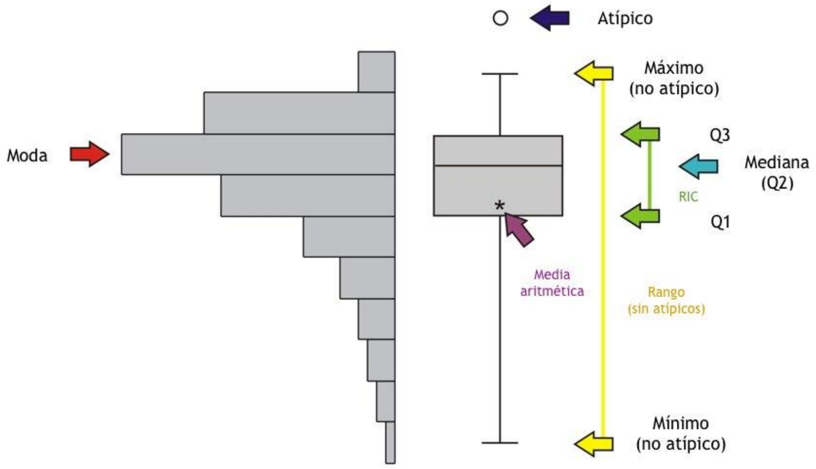
c + geom_histogram(binwidth = 5) x, y, alpha, color, fill, linetype, size, weight

c2 + geom_qq(aes(sample = hwy)) x, y, alpha, color, fill, linetype, size, weight

4.2. Distributions: Boxplots

- Boxplots are a measure of **how well distributed is the data in a data set. It divides the data set into three quartiles.**



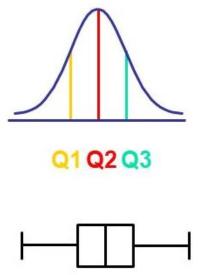


Useful for two variables, one discrete and one continuous.
Boxplots can be created for individual variables or for variables by group. Useful in comparing the distribution of data across datasets.

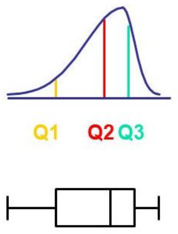
[ggplot2 box plot : Quick start guide - R software and data visualization - Easy Guides - Wiki - STHDA]

4.2. Distributions: Boxplots - Distributions

- Boxplots are a measure of **how well distributed is the data in a data set.** It divides the data set into three quartiles.

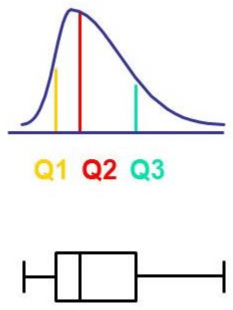


Symmetric distribution (normal)
– bell shape
Most of the cases have central values



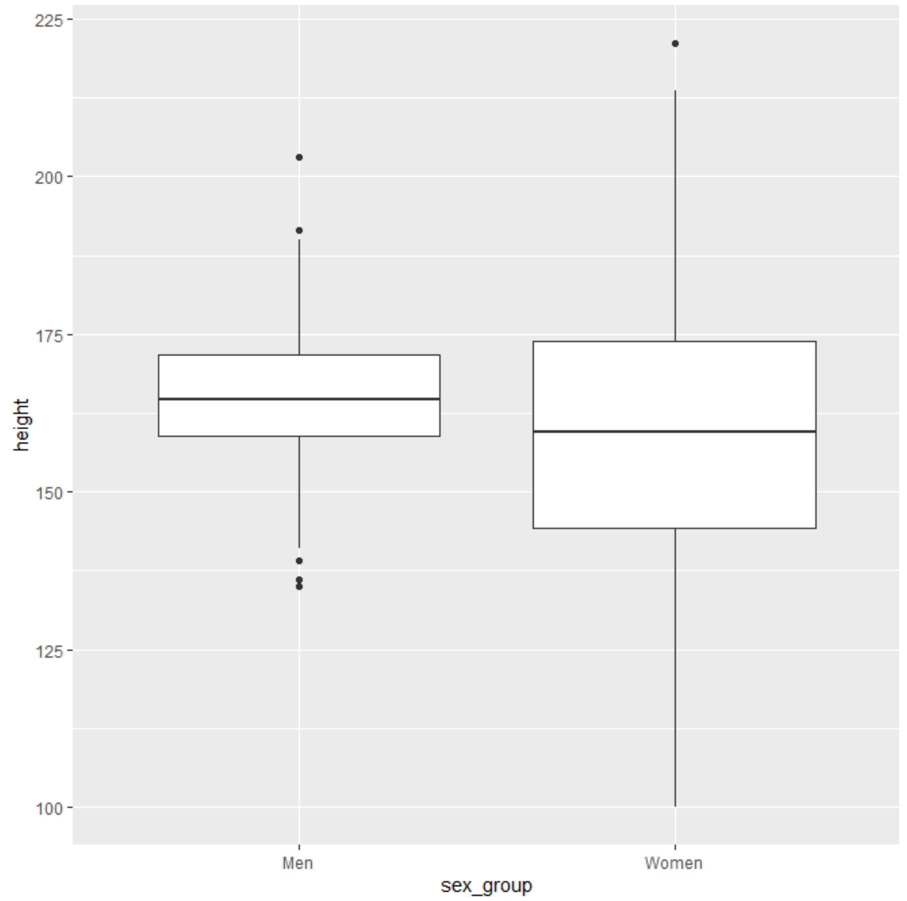
Asymmetric negative/left distribution
Most of the cases have *high values*

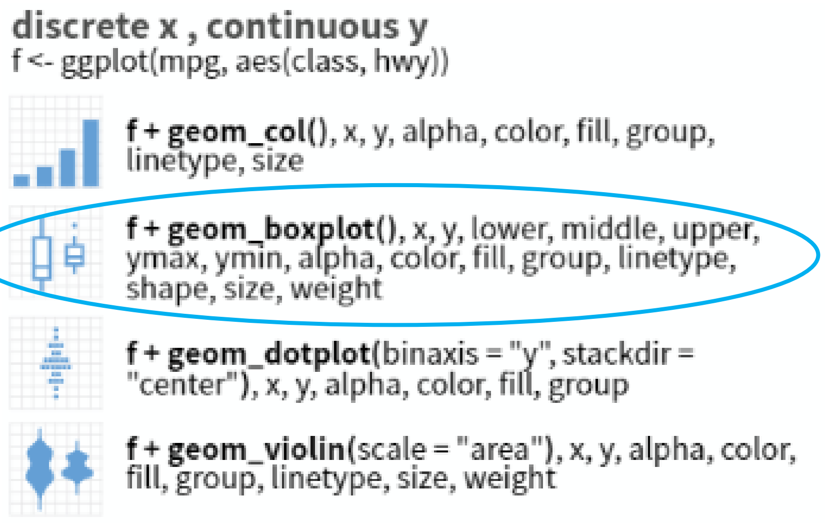
Asymmetric positive/right distribution
Most of the cases have *high values*



3.2. Boxplots – ggplot2 Example

```
> ggplot(data=data_example) + aes(height) + geom_boxplot()
```





discrete x , continuous y
f <- ggplot(mpg, aes(class, hwy))

f + **geom_col()**, x, y, alpha, color, fill, group, linetype, size

f + **geom_boxplot()**, x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight

f + **geom_dotplot**(binaxis = "y", stackdir = "center"), x, y, alpha, color, fill, group

f + **geom_violin**(scale = "area"), x, y, alpha, color, fill, group, linetype, size, weight

Let’s do some exercises