GRAU EN ENGINYERIA DE DADES

104365
TEMA - 7. Sistemes Avançats I

Departament de Matemàtiques

7. Advanced systems of visualization (I)

7.1 Multiple variables and dimensions

7.2 Networks

7.3 3D Data

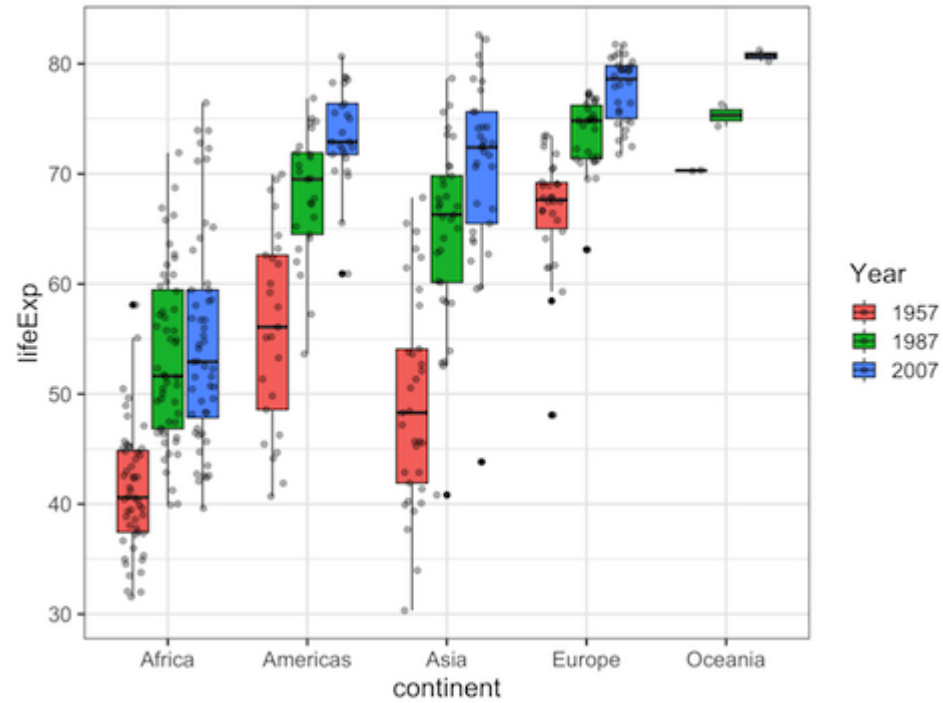
7.4 Vector Fields

7.1 Multiple variables and dimensions. Contents

7.1.1 Multiple variables and dimensions. Introduction

There are **large datasets**, **containing much more information than can be shown in a plot.**

- Some datasets can be shown **in a single figure panel by grouping variables**.



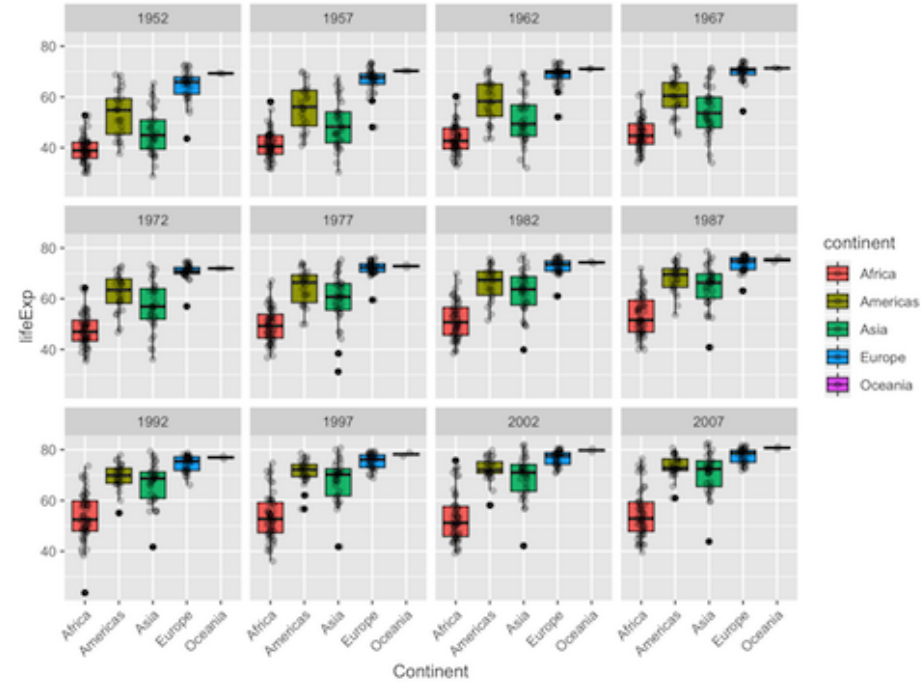
7.1.1 Multiple variables and dimensions. Introduction

There are large datasets, containing much more information than can be shown in a plot.

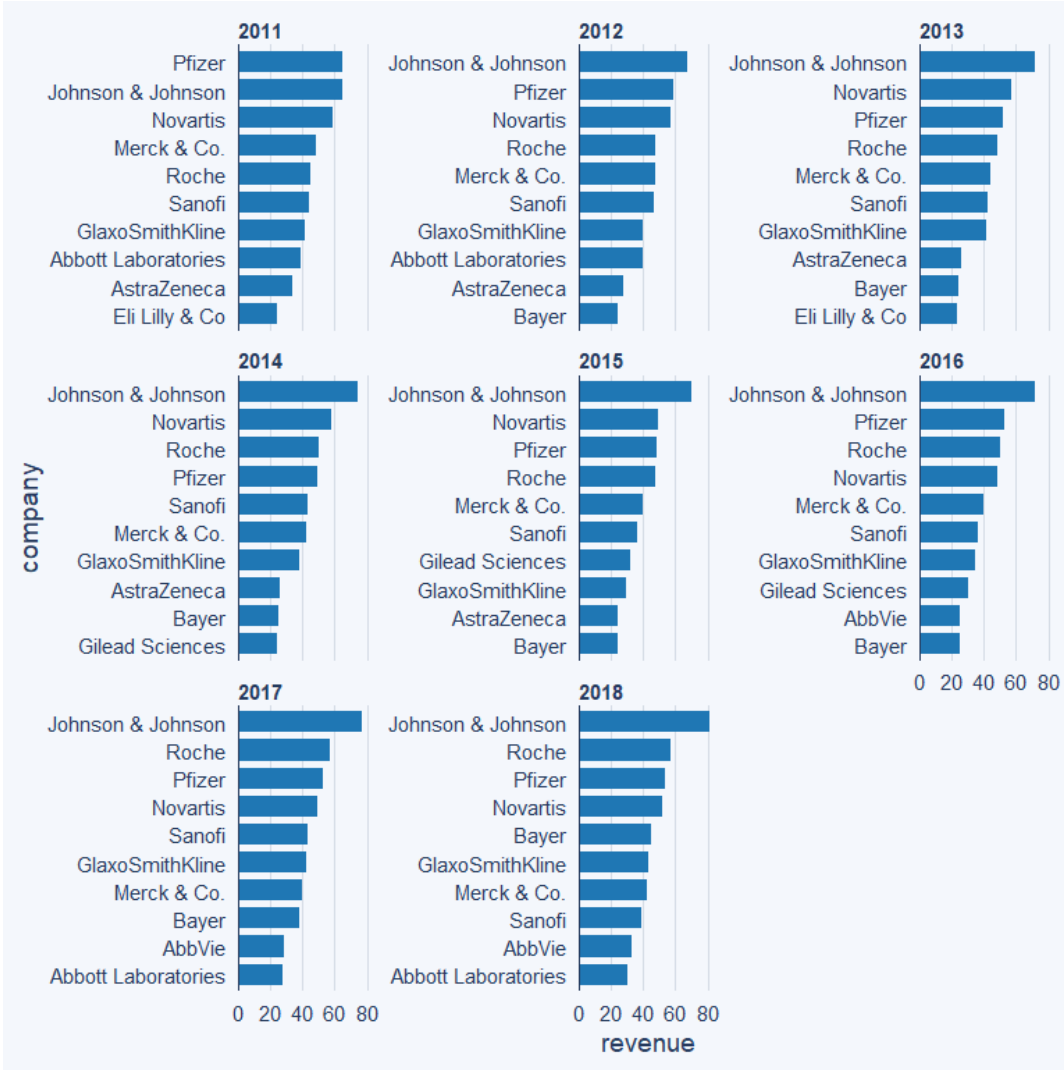
- However, **for more complex datasets, it can be helpful to create multi-panel figures**.

These are figures that consist of *multiple figure panels where* ***each panel shows some subset of the data.***

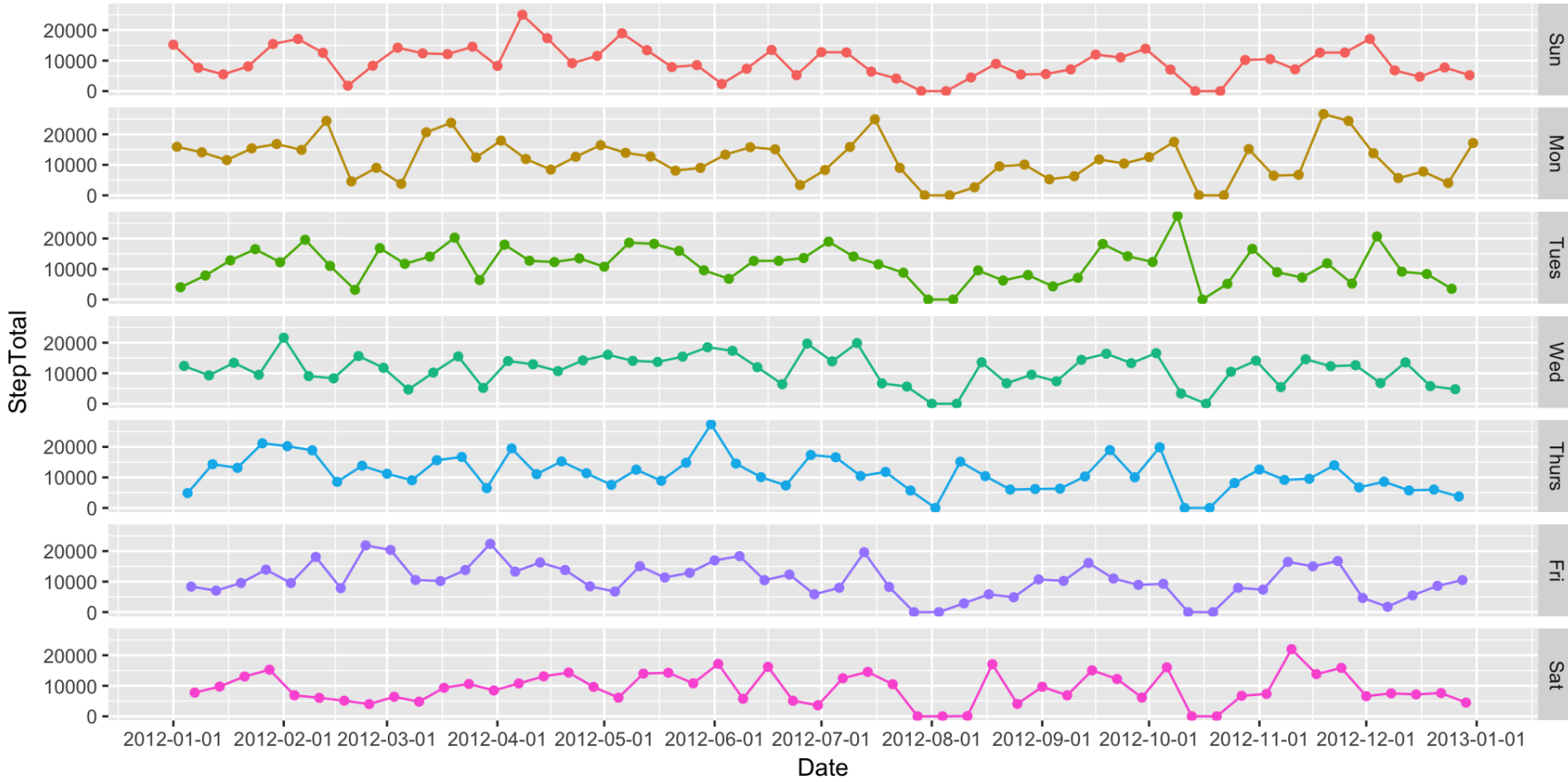
Exemple: Facets en R



7.1.1 Multiple variables and dimensions. Introduction



7.1.1 Multiple variables and dimensions. Introduction



Time Series data (one row for each weekday)

7.1.2 Visualizing many distributions at once

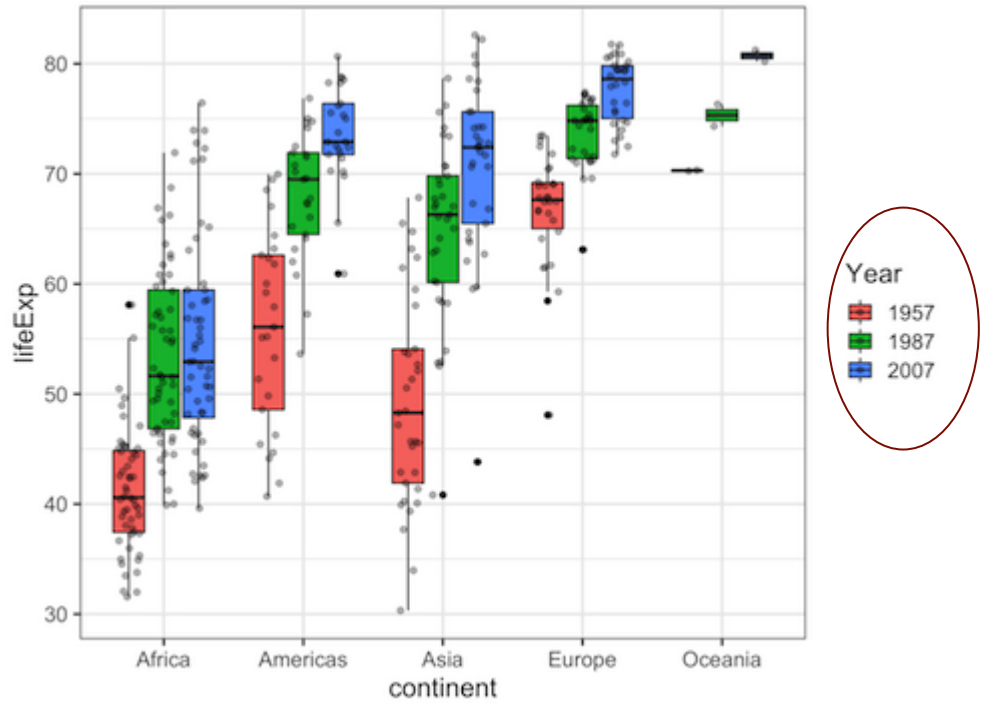
There are many scenarios in which we want **to visualize multiple distributions at the same time.**

For this, it is helpful to think in terms of the response variable and one or more grouping variables.

- The **response variable** is the variable *whose distributions we want to show.*
- The **grouping variables** define *subsets of the data with distinct distributions of the response variable.*

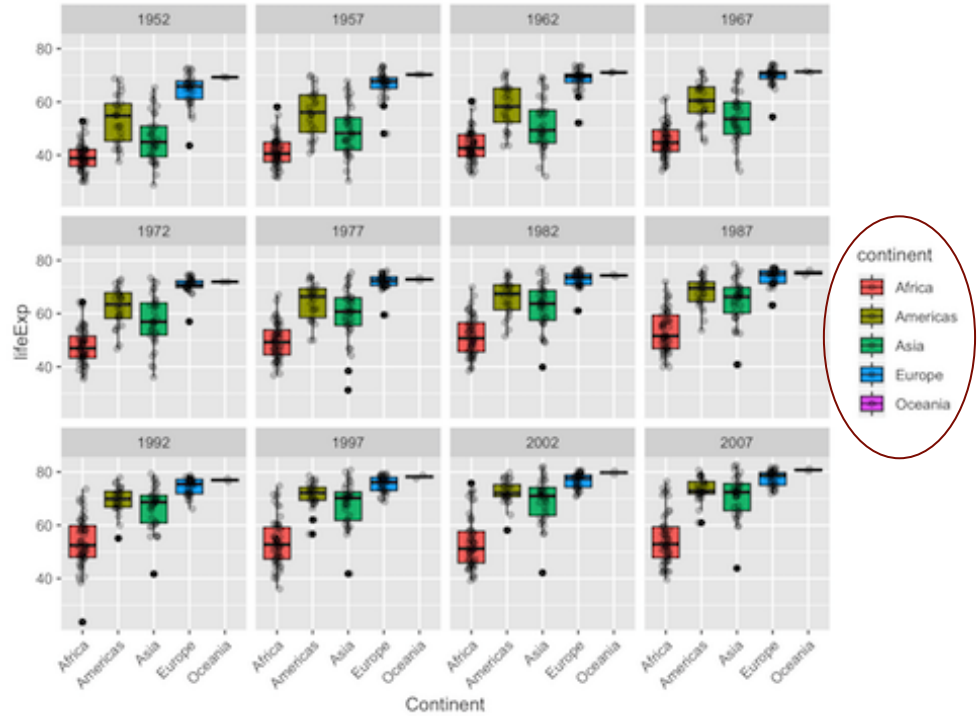
7.1.2 Visualizing many distributions at once

- The **response variable:** is the variable whose distributions we want to show (***lifeExp***).
- The **grouping variables:** define subsets of the data with distinct distributions of the response variable ***(continent / year)***



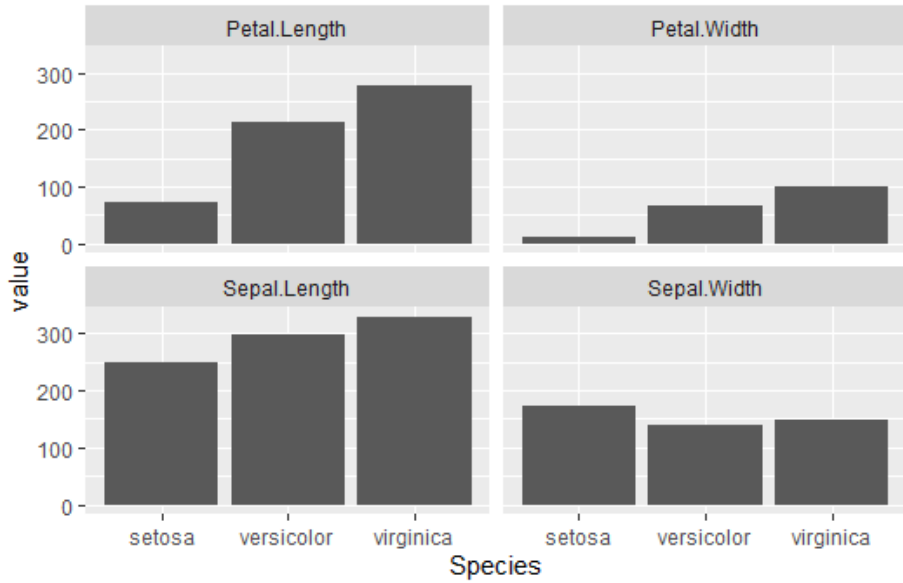
7.1.2 Visualizing many distributions at once

- The **response variable:** is the variable whose distributions we want to show (***lifeExp***).
- The **grouping variables:** define subsets of the data with distinct distributions of the response variable ***(continent / year)***



7.1.2 Visualizing many distributions at once

- The **response variable:** is the variable whose distributions we want to show.
- The **grouping variables:** define subsets of the data with distinct distributions of the response variable.



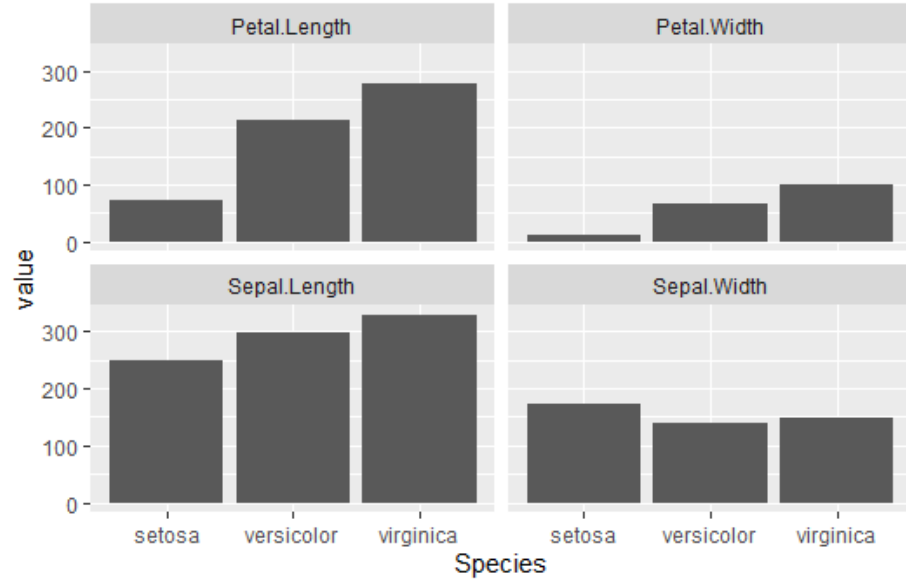
Seminar 4

```
>iris_long<-gather(iris, metric, value, -Species)
>ggplot(iris_long)+aes(Species, value)+geom_bar(stat='identity')+
facet_wrap(~ metric)
```

7.1.2 Visualizing many distributions at once

- The **response variable:** is the variable whose distributions we want to show (***value of the metric in cm– length or width***).

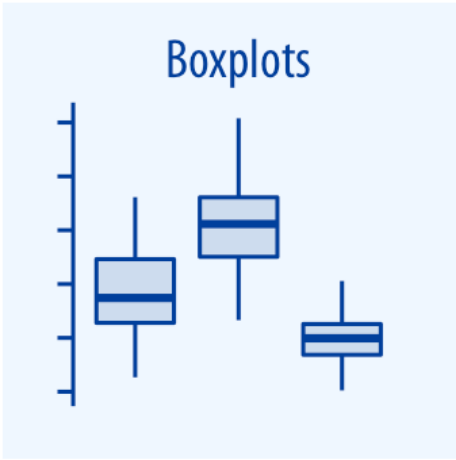
- The **grouping variables:** define subsets of the data with distinct distributions of the response variable ***(species / petal/sepal)***

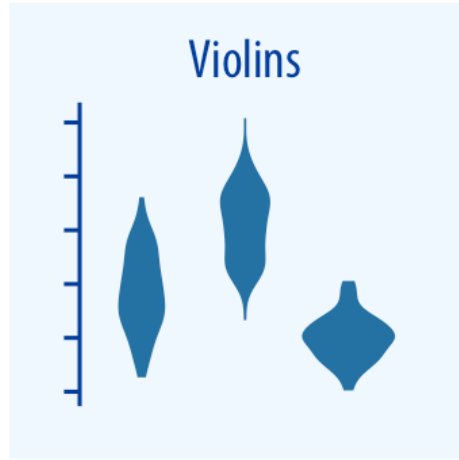


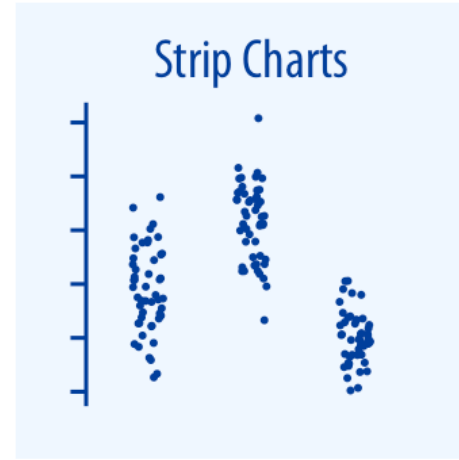
Seminar 4

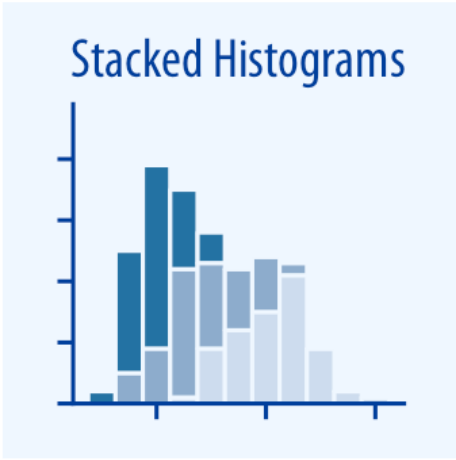
```
>iris_long<-gather(iris, metric, value, -Species)
>ggplot(iris_long)+aes(Species, value)+geom_bar(stat='identity')+
facet_wrap(~ metric)
```

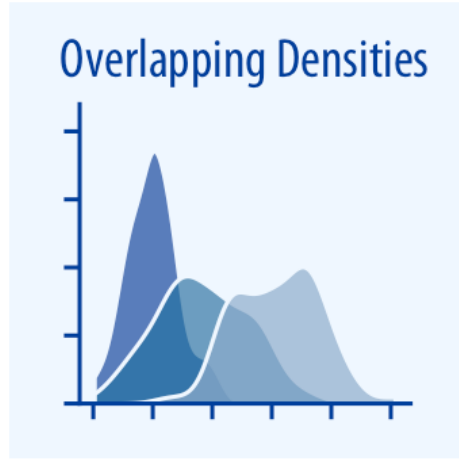
7.1.2 Visualizing many distributions at once

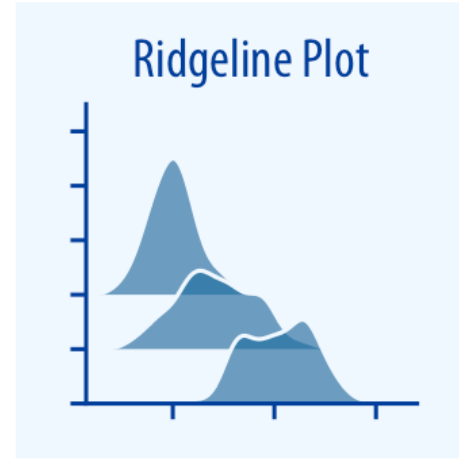




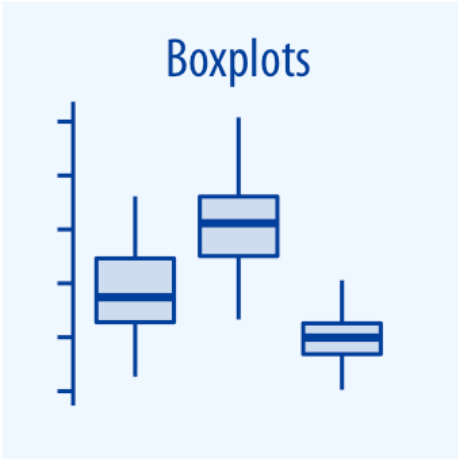






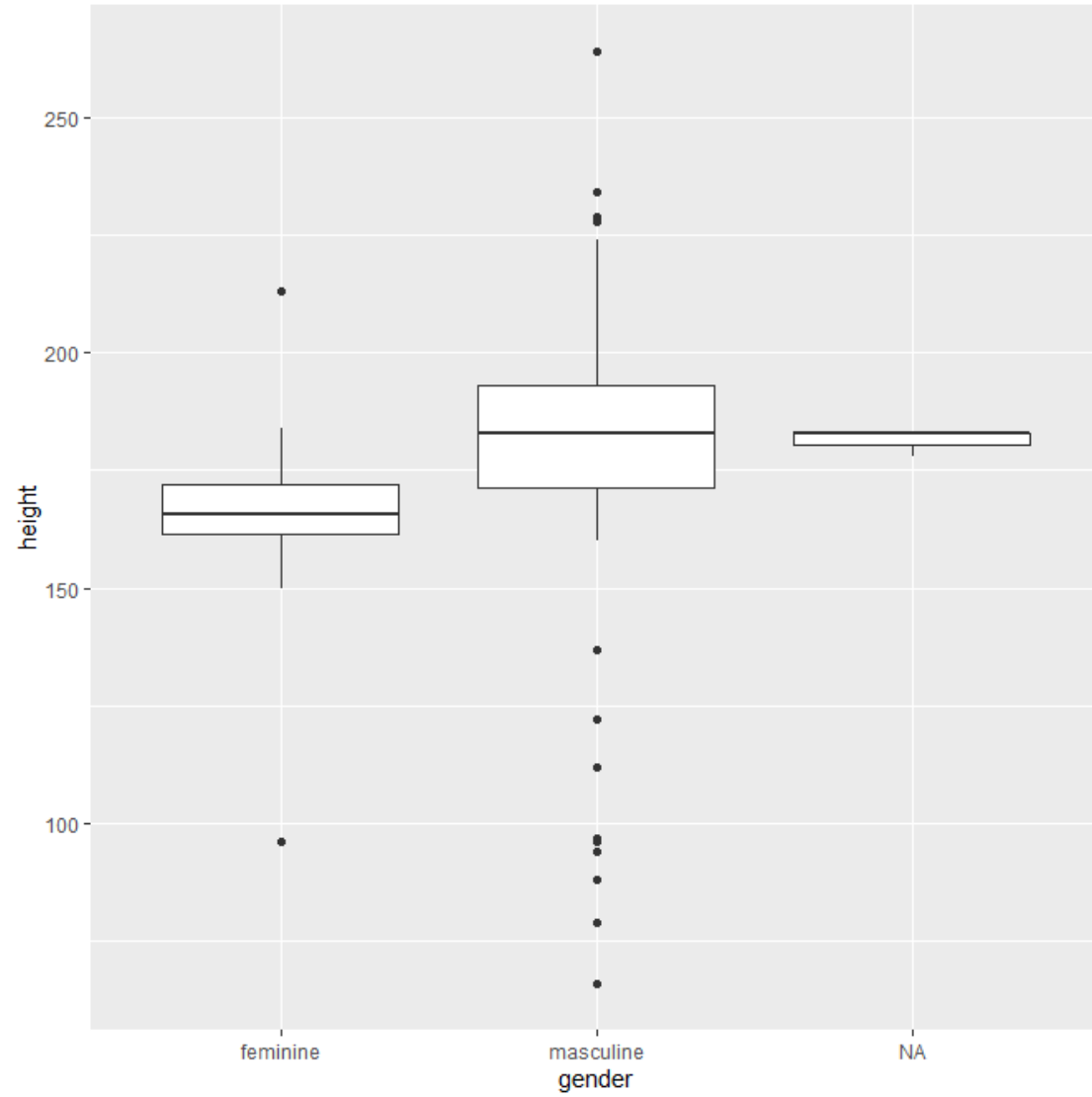


7.1.2 Visualizing many distributions at once



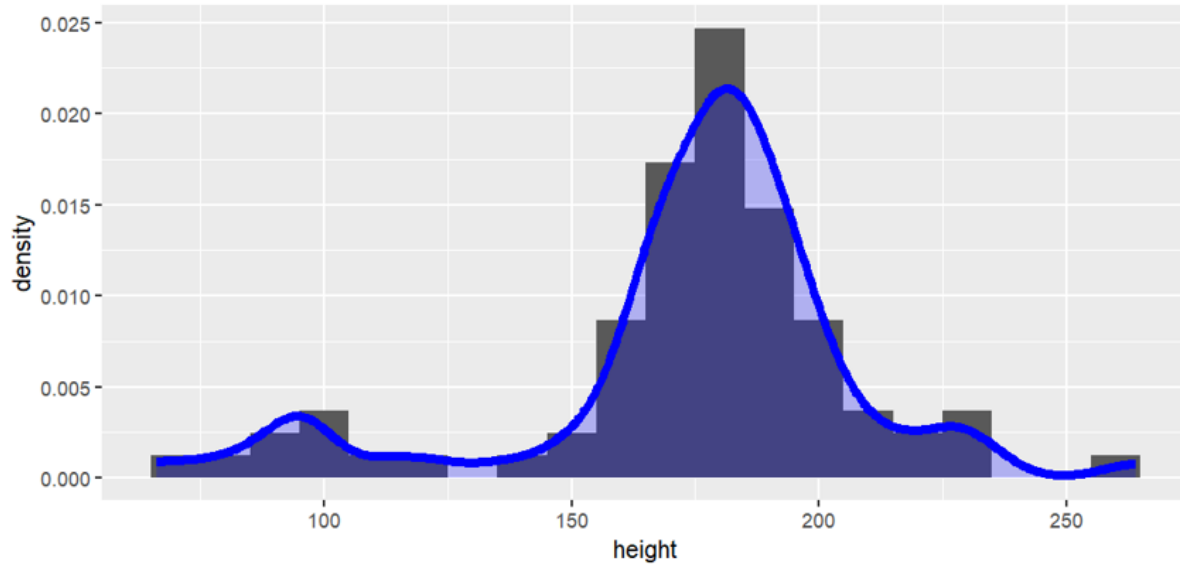
Seminar 3

```
>ggplot(starwars,aes(x=gender,
y=height))+geom_boxplot()
```



7.1.2 Visualizing many distributions at once

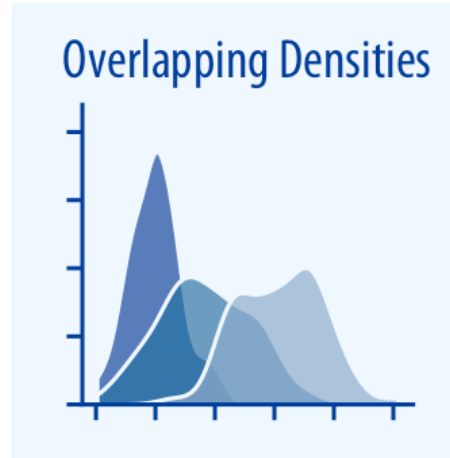
Una altra opció és adjuntar ambdues gràfiques en una fent servir una transparència. Per això podeu posar **aes(y=..density..)** en el geom_histogram.




```
>ggplot(starwars,aes(x=height))+geom_histogram(binwidth=10, aes(y=..density..))+geom_density(lwd = 2, colour = 'blue', fill = 'blue', alpha = 0.25)
```

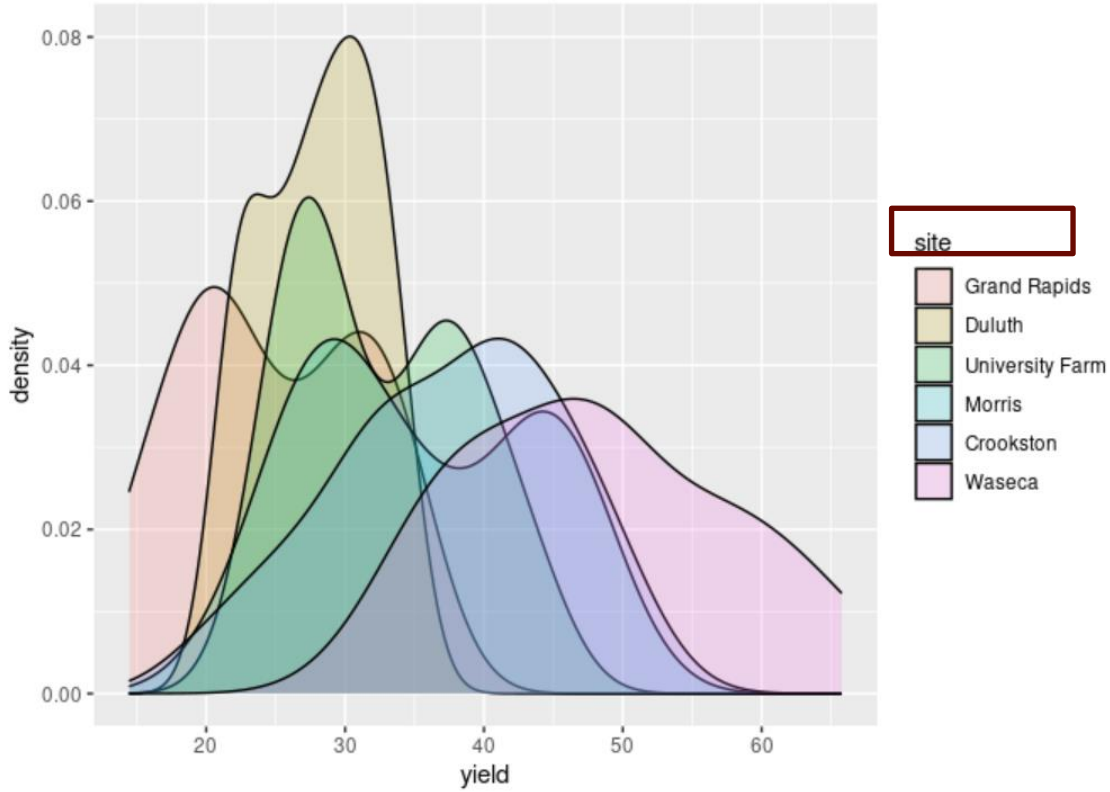
Nota: lwd només marca el gruix de la línia de geom_density

Seminar 3
Overlapping histogram and density



7.1.2 Visualizing many distributions at once

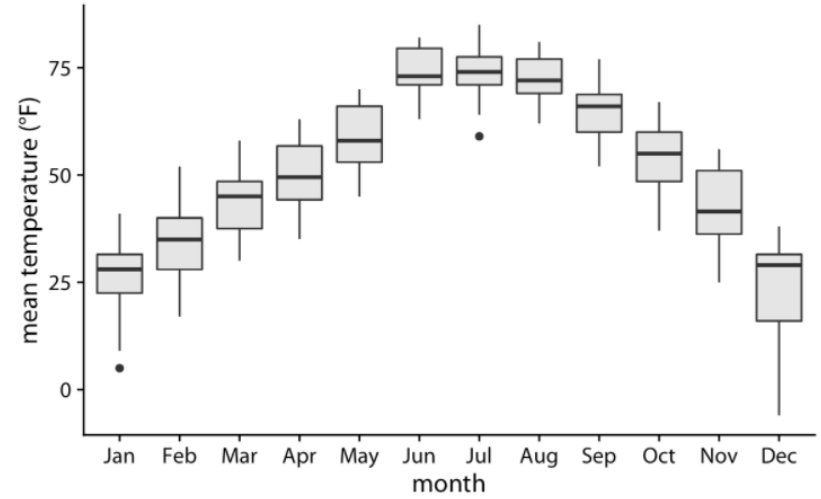
```
ggplot(barley) + geom_density(aes(x = yield, fill = site), alpha = 0.2)
```



https://homepage.divms.uiowa.edu/~luke/classes/STAT4580/histdens.html

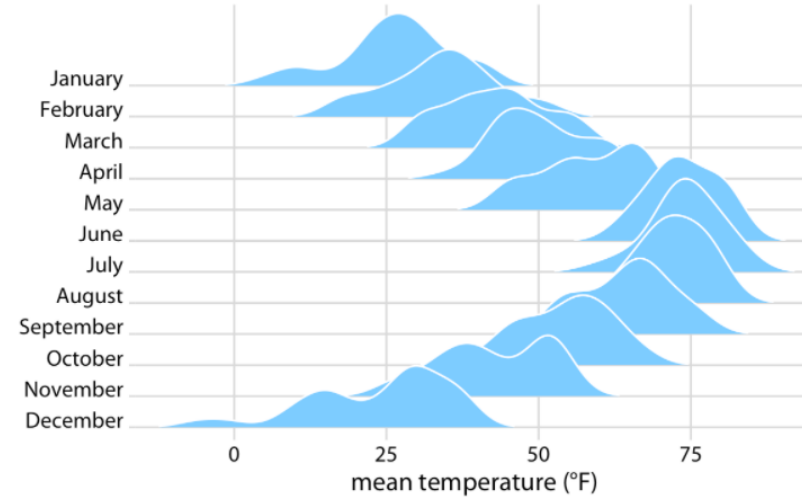
7.1.2 Visualizing many distributions at once

- **Along the vertical axis**

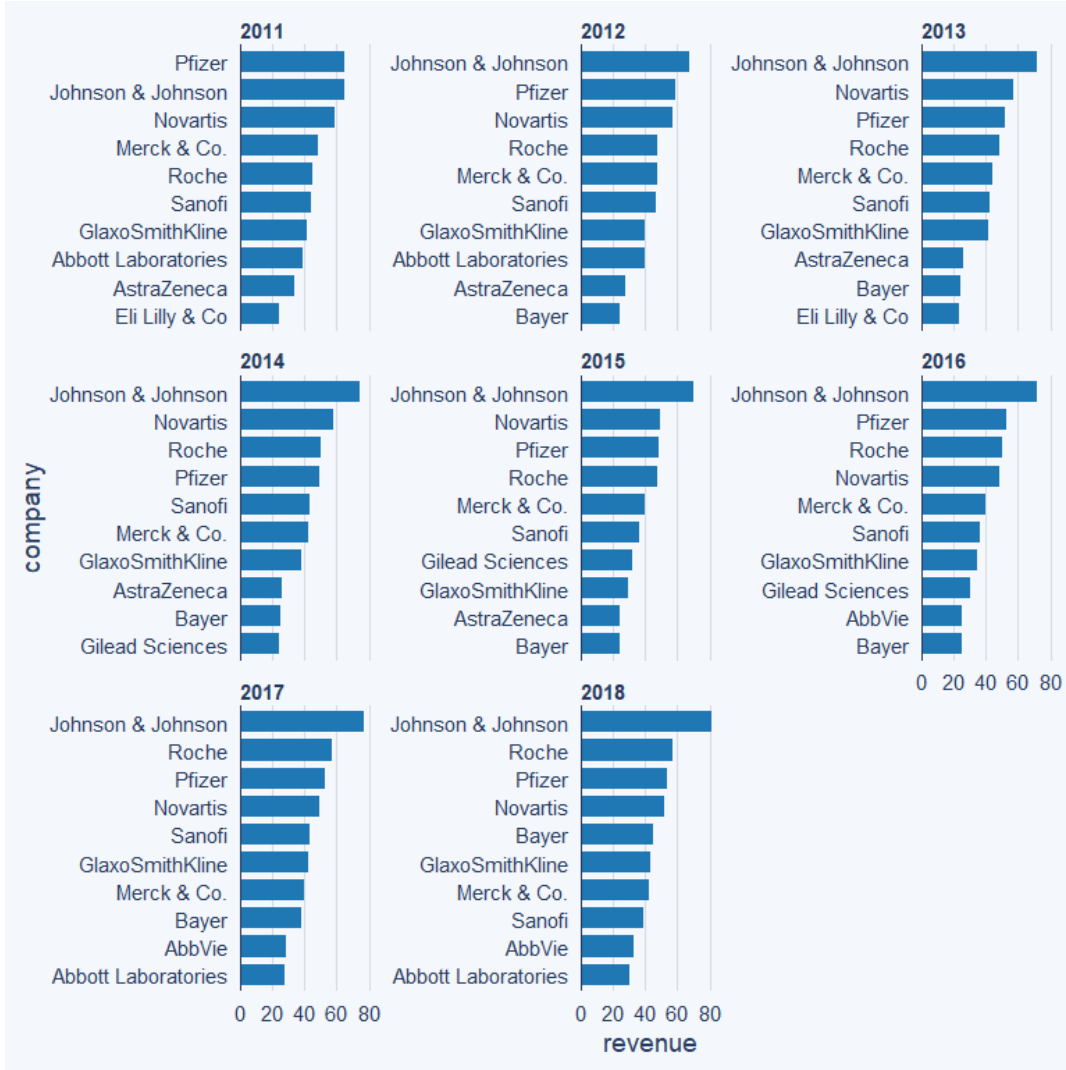


Claus Wilke

- **Along the horizontal axis**

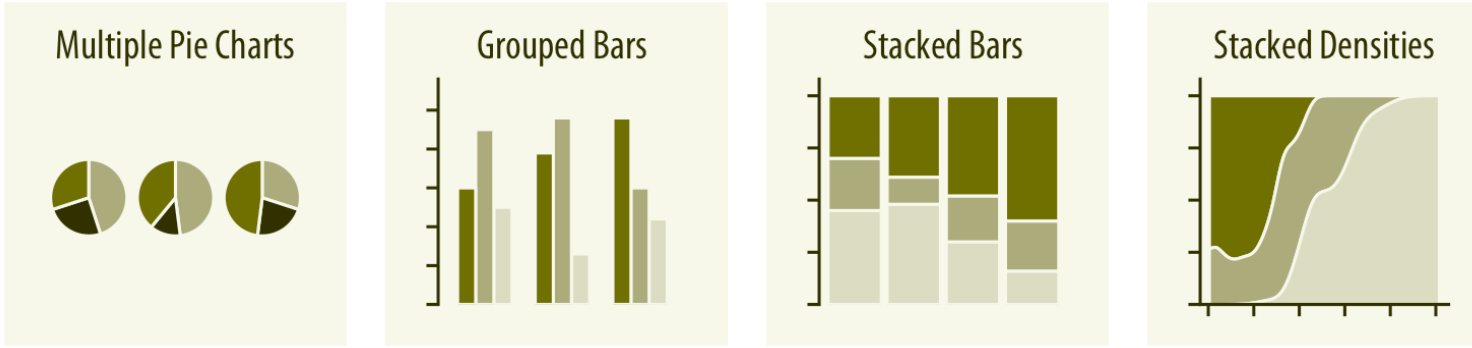


7.1.3 Visualizing many proportions at once



7.1.3 Visualizing many proportions at once

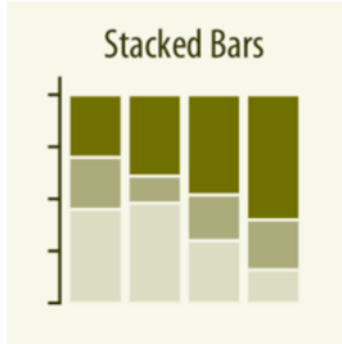
- **Pie charts** tend to be **space-inefficient and often obscure relationships.** ***(Example seminari 3)***
- **Grouped bars** work well as long as **the number of conditions compared is moderate.**
- **Stacked bars** can work **for large numbers of conditions**.
- **Stacked densities** are appropriate when **the proportions change along a continuous variable.**



Claus Wilke

7.1.3 Visualizing many proportions at once

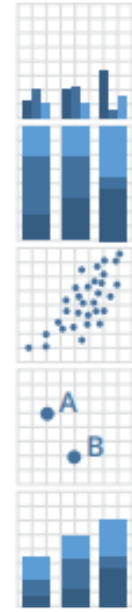
- **Stacked bars** can work **for large numbers of conditions**.



Claus Wilke

Position Adjustments

Position adjustments determine how to arrange geoms that would otherwise occupy the same space.



s <- ggplot(mpg, aes(fl, fill = drv))

s + geom_bar(position = "dodge")
Arrange elements side by side

s + geom_bar(position = "fill")
Stack elements on top of one another, normalize height

e + geom_point(position = "jitter")
Add random noise to X and Y position of each element to avoid overplotting

e + geom_label(position = "nudge")
Nudge labels away from points

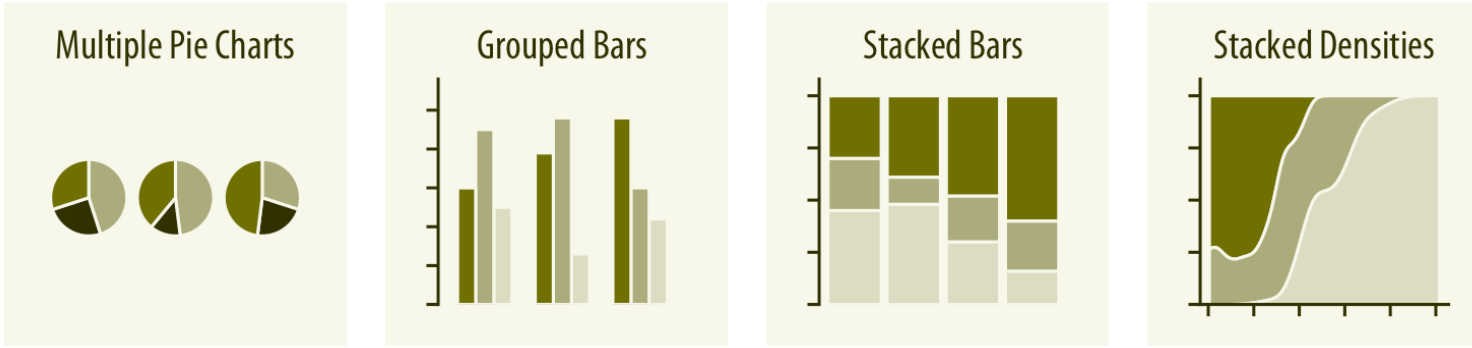
s + geom_bar(position = "stack")
Stack elements on top of one another

Each position adjustment can be recast as a function with manual **width** and **height** arguments

s + geom_bar(position = position_dodge(width = 1))

7.1.3 Visualizing many proportions at once

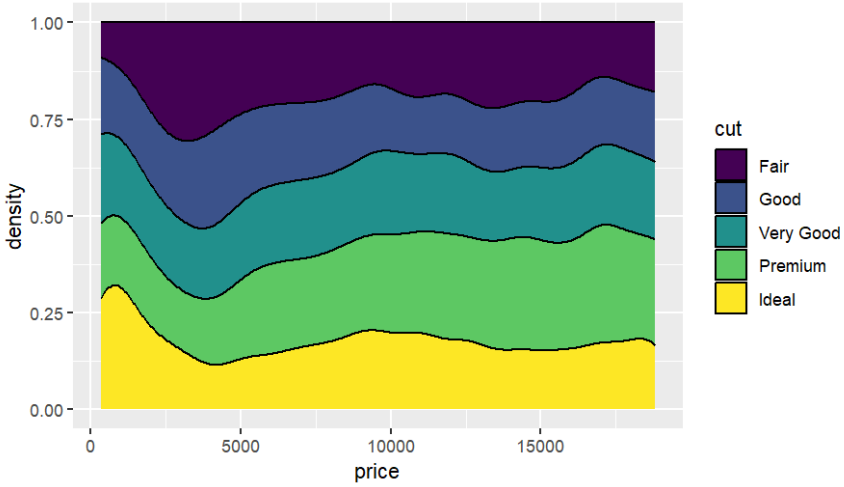
- **Pie charts** tend to be **space-inefficient and often obscure relationships.** ***(Example seminari 3)***
- **Grouped bars** work well as long as **the number of conditions compared is moderate.**
- **Stacked bars** can work **for large numbers of conditions**.
- **Stacked densities** are appropriate when **the proportions change along a continuous variable.**



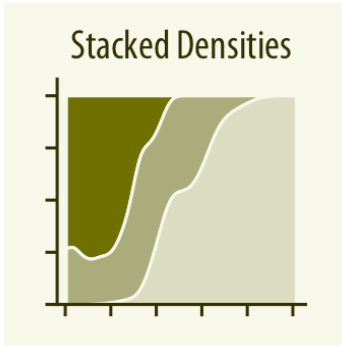
Claus Wilke

7.1.3 Visualizing many proportions at once

- **Stacked densities** are appropriate when **the proportions change along a continuous variable.**
- Stacking is a process where a chart is broken up across more than one categoric variables which make up the whole.



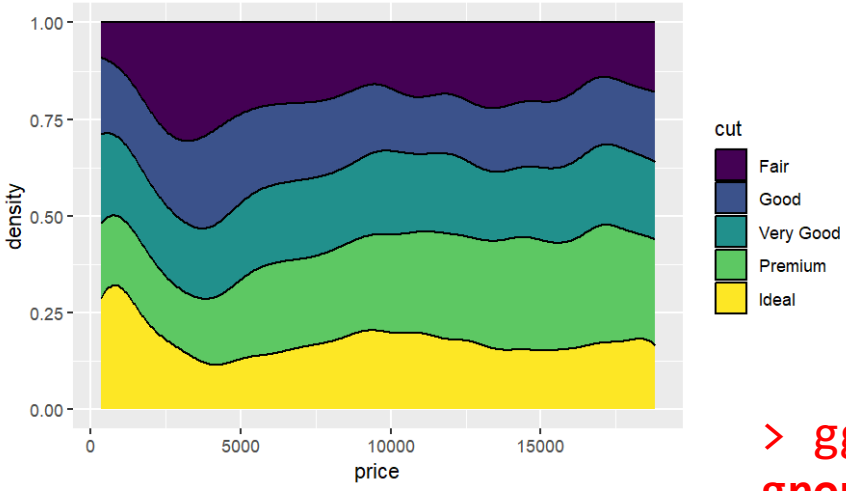
Diamonds contains information on price, cut characteristics, color, carats, etc... of nearly 54,000 diamonds



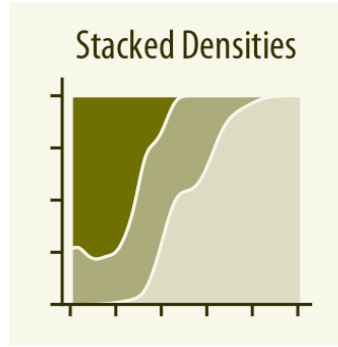
Claus Wilke

7.1.3 Visualizing many proportions at once

- **Stacked densities** are appropriate when **the proportions change along a continuous variable.**
- Stacking is a process where a chart is broken up across more than one categoric variables which make up the whole.



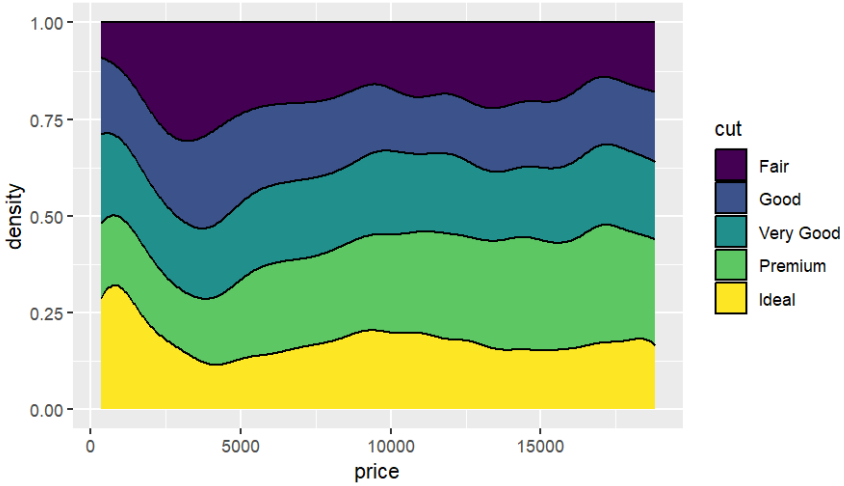
Diamonds contains information on price, cut characteristics, color, carats, etc... of nearly 54,000 diamonds



Claus Wilke

```
> ggplot(data=diamonds, aes(x=price,
group=cut, fill=cut)) +
geom_density(adjust=1.5, position="fill")
```

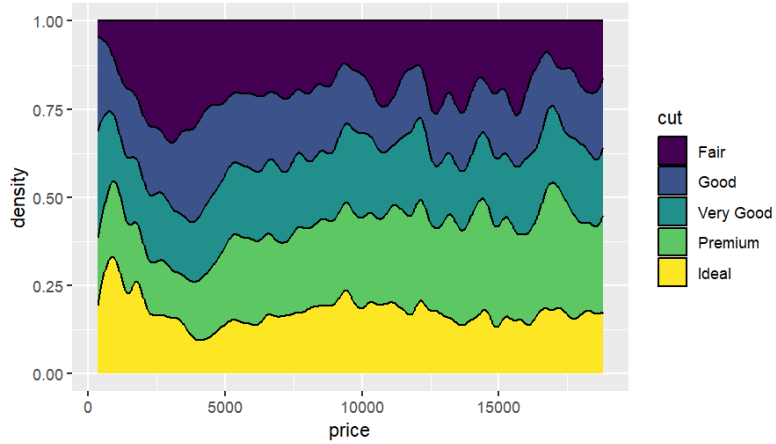
7.1.3 Visualizing many proportions at once



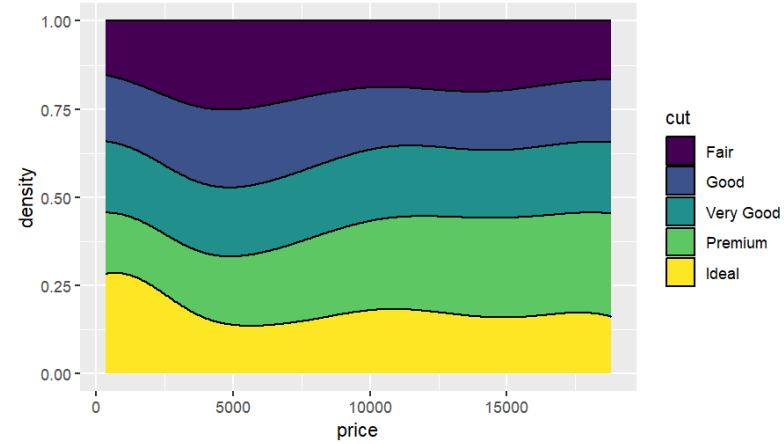
It contains information on price, cut characteristics, color, carats, etc... of nearly 54,000 diamonds

```
> ggplot(data=diamonds, aes(x=price, group=cut, fill=cut)) +
geom_density(adjust=1.5, position="fill")
```

A multiplicative bandwidth adjustment. This makes it possible to adjust the bandwidth while still using a bandwidth estimator



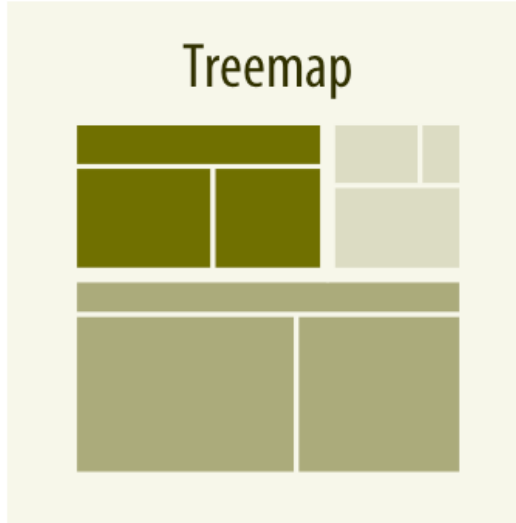
adjust=0.5



adjust=4

7.1.3 Visualizing many proportions at once

When proportions are specified **according to multiple grouping variables:**



Claus Wilke

Mosaic plots, treemaps, or parallel sets are useful visualization approaches

7.1.3 Visualizing many proportions at once

Whenever we have categories that overlap, it is best to show clearly how they relate to each other

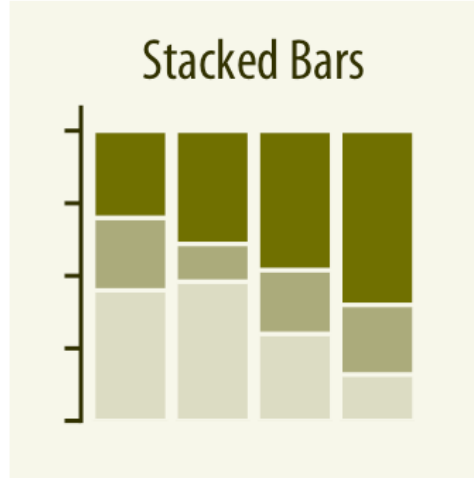
Mosaic Plot: assumes that **each level of one grouping variable** can be **combined with each level of another grouping variable**.

Claus Wilke

7.1.3 Visualizing many proportions at once

Whenever we have categories that overlap, it is best to show clearly how they relate to each other

- **Mosaic plots** looks similar to the stacked bar plot. However, in a mosaic plot **both the heights and the width of individual shaded areas vary**



Claus Wilke

7.1.3 Visualizing many proportions at once

Whenever we have categories that overlap, it is best to show clearly how they relate to each other

- **Mosaic plots example:**

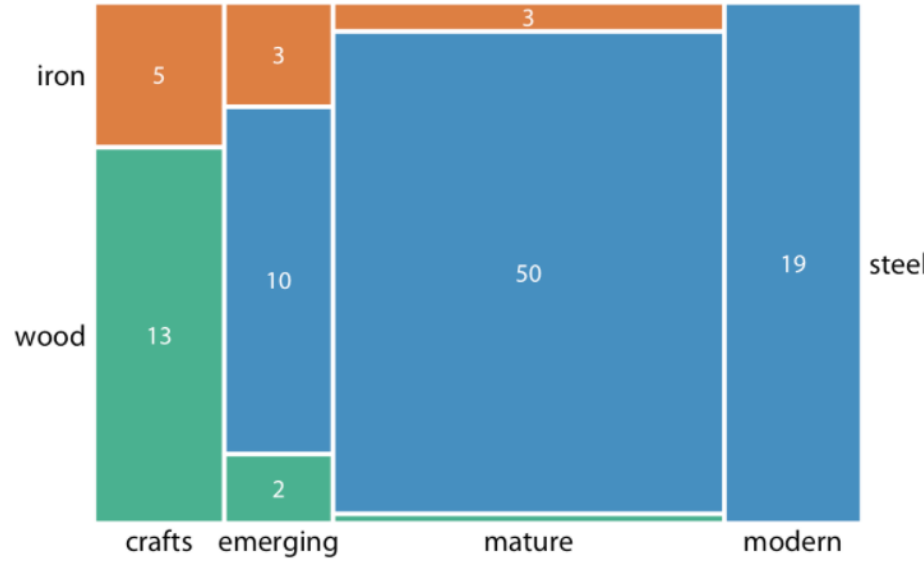


Figure 11.3: Breakdown of bridges in Pittsburgh by construction material (steel, wood, iron) and by era of construction (crafts, emerging, mature, modern), shown as a mosaic plot. The widths of each rectangle are proportional to the number of bridges constructed in that era, and the heights are proportional to the number of bridges constructed from that material. Numbers represent the counts of bridges within each category. Data source: Yoram Reich and Steven J. Fenves, via the UCI Machine Learning Repository (Dua and Karra Taniskidou 2017)

Claus Wilke

7.1.3 Visualizing many proportions at once

- **Mosaic plots assume that every level of one grouping variable can be combined with every level of another grouping variable**, whereas treemaps do not make such an assumption.

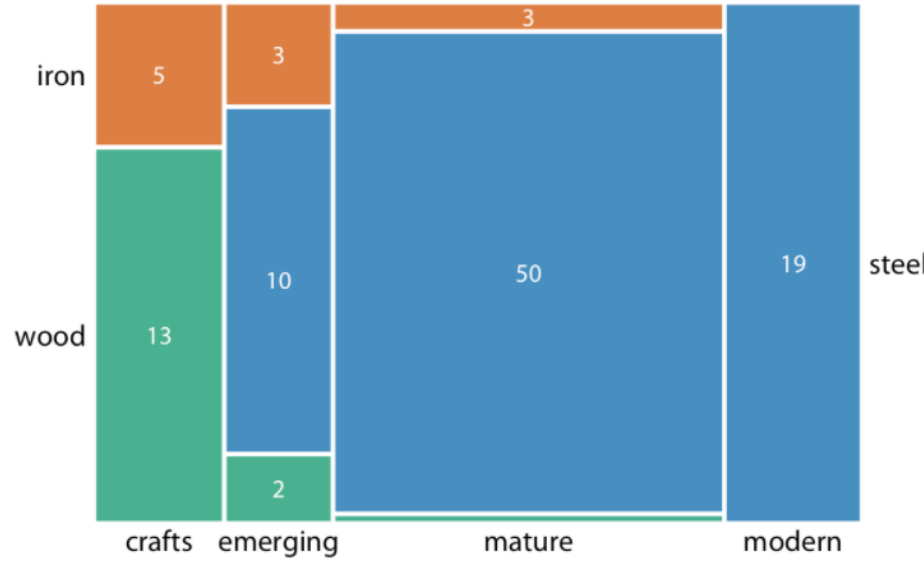


Figure 11.3: Breakdown of bridges in Pittsburgh by construction material (steel, wood, iron) and by era of construction (crafts, emerging, mature, modern), shown as a mosaic plot. The widths of each rectangle are proportional to the number of bridges constructed in that era, and the heights are proportional to the number of bridges constructed from that material. Numbers represent the counts of bridges within each category. Data source: Yoram Reich and Steven J. Fenves, via the UCI Machine Learning Repository (Dua and Karra Taniskidou 2017)

7.1.3 Visualizing many proportions at once

- **Mosaic plots**

We begin by **placing one categorical variable along the *x* axis** (here, *era* of bridge construction) **and subdivide the *x* axis by the relative proportions that make up the categories**.

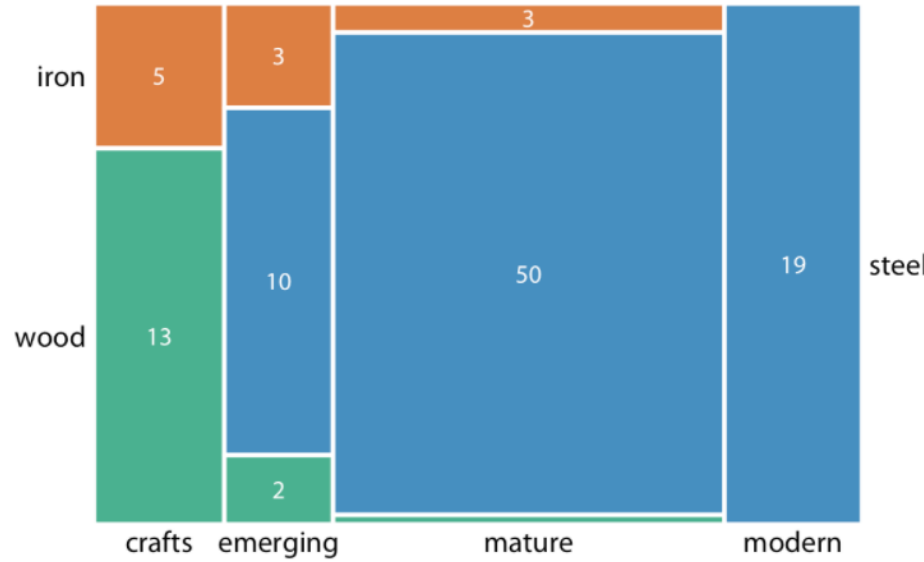
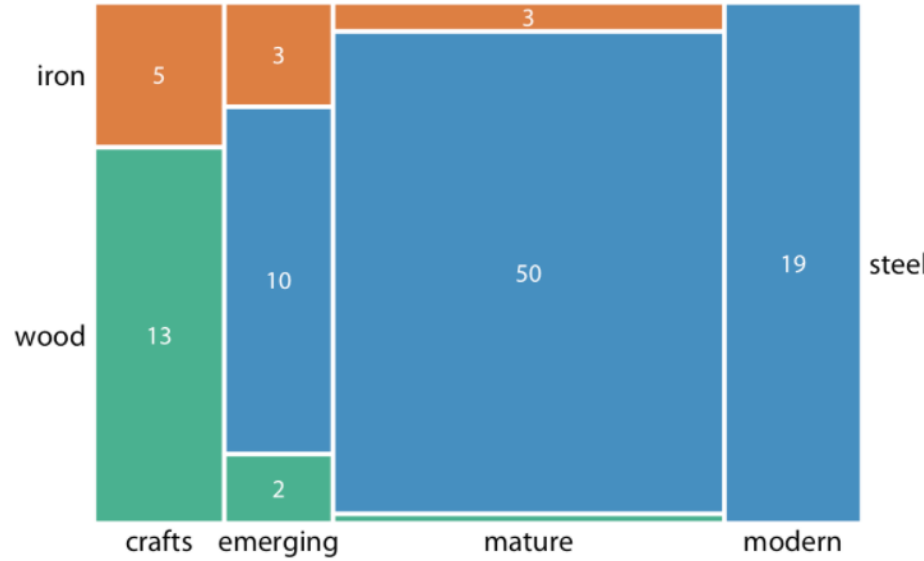


Figure 11.3: Breakdown of bridges in Pittsburgh by construction material (steel, wood, iron) and by era of construction (crafts, emerging, mature, modern), shown as a mosaic plot. The widths of each rectangle are proportional to the number of bridges constructed in that era, and the heights are proportional to the number of bridges constructed from that material. Numbers represent the counts of bridges within each category. Data source: Yoram Reich and Steven J. Fenves, via the UCI Machine Learning Repository (Dua and Karra Taniskidou 2017)

7.1.3 Visualizing many proportions at once

- **Mosaic plots**

We begin by **placing one categorical variable along the *x* axis** (here, era of bridge construction) **and subdivide the *x* axis by the relative proportions that make up the categories**.



We then place the other categorical variable along the *y* axis (here, *building material*) and, **within each category along the *x* axis, subdivide the *y* axis by the relative proportions that make up the categories of the *y* variable.**

Figure 11.3: Breakdown of bridges in Pittsburgh by construction material (steel, wood, iron) and by era of construction (crafts, emerging, mature, modern), shown as a mosaic plot. The widths of each rectangle are proportional to the number of bridges constructed in that era, and the heights are proportional to the number of bridges constructed from that material. Numbers represent the counts of bridges within each category. Data source: Yoram Reich and Steven J. Fenves, via the UCI Machine Learning Repository (Dua and Karra Taniskidou 2017)

7.1.3 Visualizing many proportions at once

- Mosaic plots assume that every level of one grouping variable can be combined with every level of another grouping variable, whereas treemaps do not make such an assumption.

The result is a set of rectangles whose areas are proportional to the number of cases representing each possible combination of the two categorical variables.

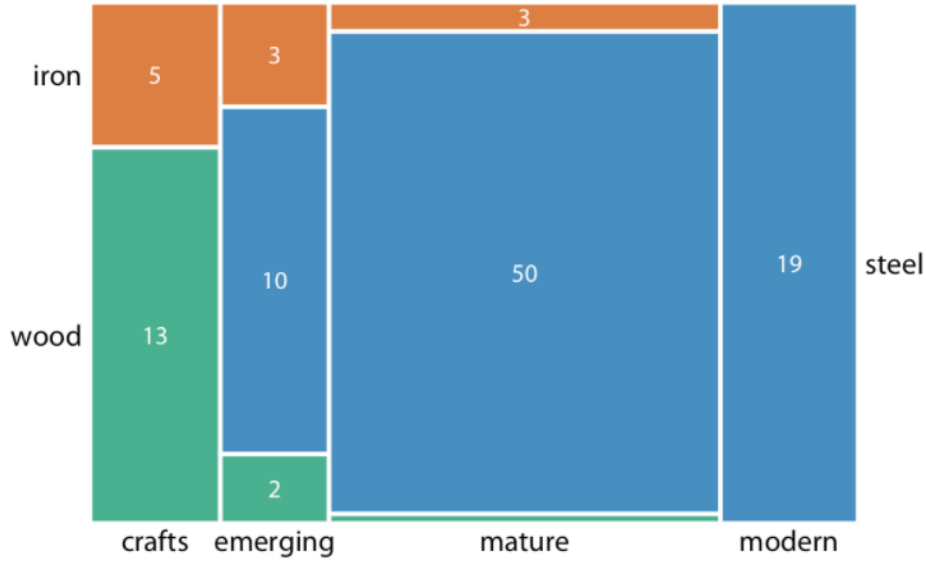


Figure 11.3: Breakdown of bridges in Pittsburgh by construction material (steel, wood, iron) and by era of construction (crafts, emerging, mature, modern), shown as a mosaic plot. The widths of each rectangle are proportional to the number of bridges constructed in that era, and the heights are proportional to the number of bridges constructed from that material. Numbers represent the counts of bridges within each category. Data source: Yoram Reich and Steven J. Fenves, via the UCI Machine Learning Repository (Dua and Karra Taniskidou 2017)

7.1.2 Visualizing many proportions at once

- **Treemaps** work well **even if the subdivisions of one group are entirely distinct from the subdivisions of another**.

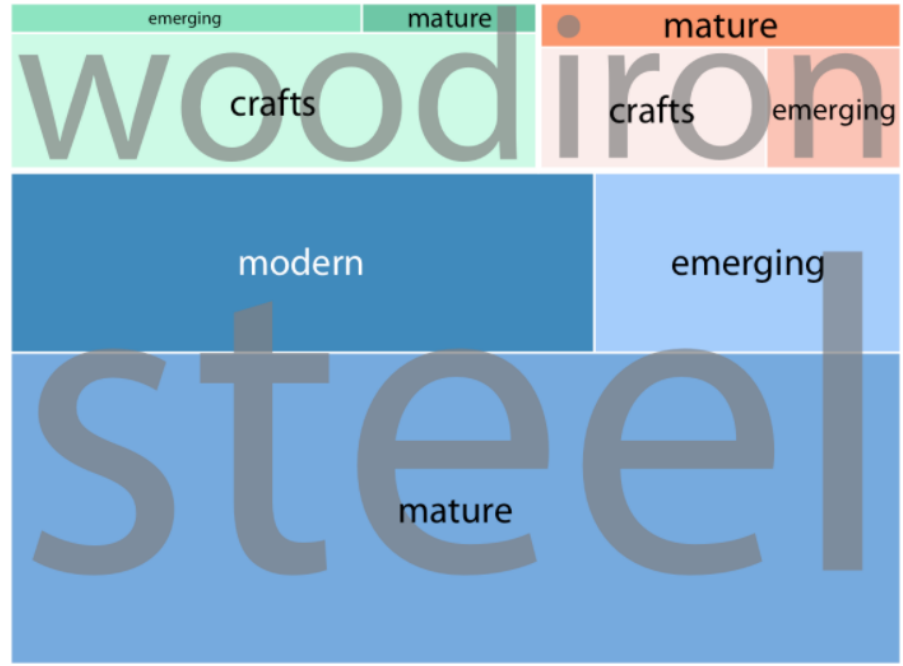


Figure 11.4: Breakdown of bridges in Pittsburgh by construction material (steel, wood, iron) and by era of construction (crafts, emerging, mature, modern), shown as a treemap. The area of each rectangle is proportional to the number of bridges of that type. Data source: Yoram Reich and Steven J. Fenves, via the UCI Machine Learning Repository (Dua and Karra Taniskidou 2017)

7.1.3 Visualizing many proportions at once

- Treemaps work well even if the subdivisions of one group are entirely distinct from the subdivisions of another.

In a treemap, **we recursively nest rectangles inside each other.**

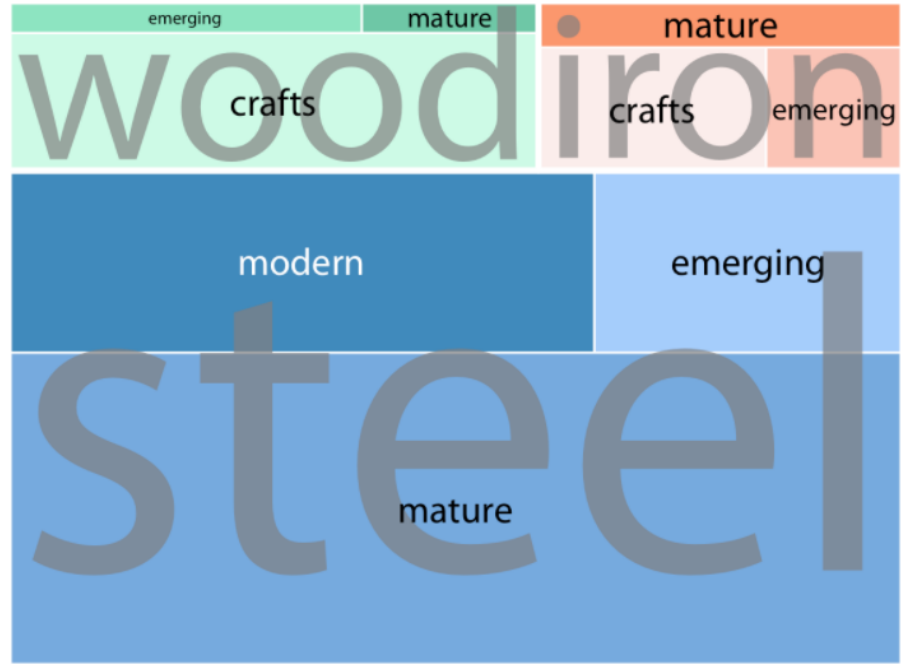


Figure 11.4: Breakdown of bridges in Pittsburgh by construction material (steel, wood, iron) and by era of construction (crafts, emerging, mature, modern), shown as a treemap. The area of each rectangle is proportional to the number of bridges of that type. Data source: Yoram Reich and Steven J. Fenves, via the UCI Machine Learning Repository (Dua and Karra Taniskidou 2017)

7.1.3 Visualizing many proportions at once

- Treemaps work well even if the subdivisions of one group are entirely distinct from the subdivisions of another.

For example, in the case of the Pittsburgh bridges, we can **first subdivide the total area into three parts representing the three building materials** *wood, iron, and steel*.

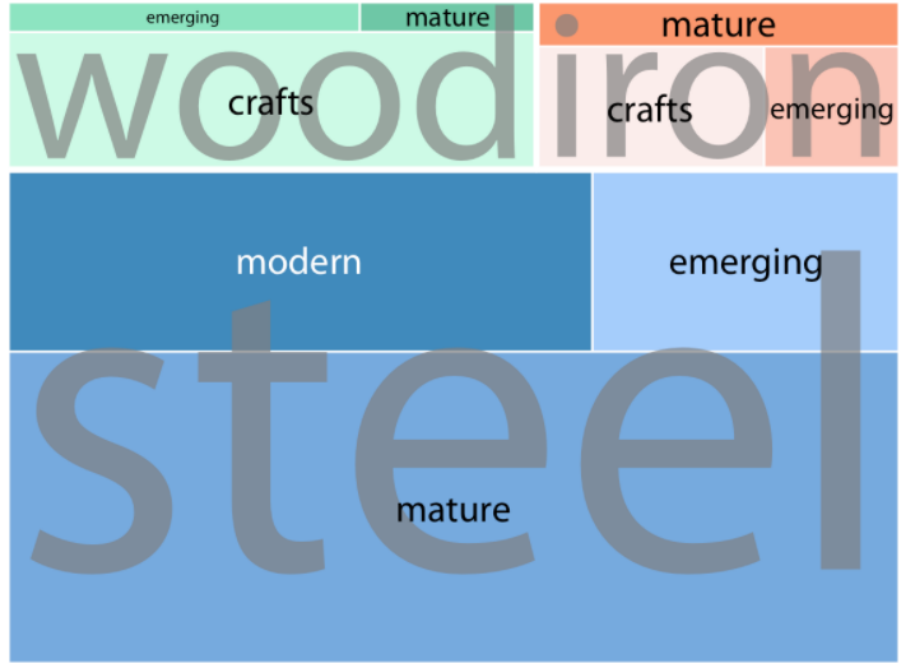
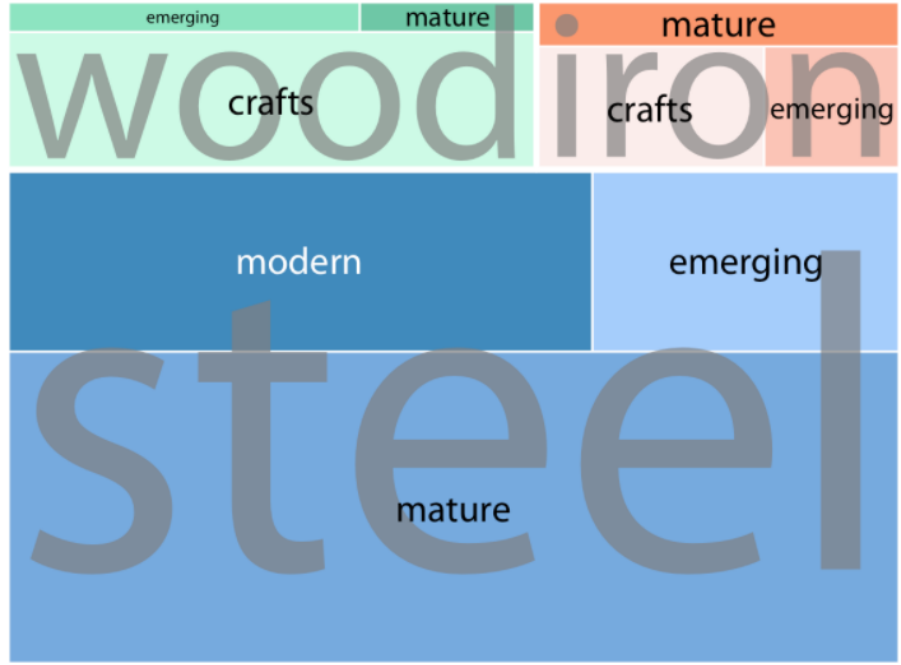


Figure 11.4: Breakdown of bridges in Pittsburgh by construction material (steel, wood, iron) and by era of construction (crafts, emerging, mature, modern), shown as a treemap. The area of each rectangle is proportional to the number of bridges of that type. Data source: Yoram Reich and Steven J. Fenves, via the UCI Machine Learning Repository (Dua and Karra Taniskidou 2017)

7.1.3 Visualizing many proportions at once

- Treemaps work well even if the subdivisions of one group are entirely distinct from the subdivisions of another.

For example, in the case of the Pittsburgh bridges, we can first subdivide the total area into three parts representing the three building materials wood, iron, and steel.



Then, we subdivide each of those areas further to represent the construction eras represented for each building material.

Figure 11.4: Breakdown of bridges in Pittsburgh by construction material (steel, wood, iron) and by era of construction (crafts, emerging, mature, modern), shown as a treemap. The area of each rectangle is proportional to the number of bridges of that type. Data source: Yoram Reich and Steven J. Fenves, via the UCI Machine Learning Repository (Dua and Karra Taniskidou 2017)

7.1.3 Visualizing many proportions at once

- **Mosaic plots** **assume that every level of one grouping variable can be combined with every level of another grouping variable**, whereas treemaps do not make such an assumption. (width/height)

- **Treemaps** work well **even if the subdivisions of one group are entirely distinct from the subdivisions of another**. (area)

- **Parallel sets** work better than either mosaic plots or treemaps **when there are more than two grouping variables**. (next class)

7.1.3 Visualizing many proportions at once

Remember: **A treemap is a rectangular plot divided into tiles**, **each of which represents a single observation**. **It is a nice way of displaying hierarchical data** by using nested rectangles.

The relative area of each tile expresses a continuous variable.

- treemapify provides ggplot2 geoms for drawing treemaps
- Install the release version of treemapify from CRAN:

```
>install.packages("treemapify")
> library(treemapify)
```

- geom_treemap () – A 'ggplot2' geom to draw a treemap

Material of interest (extra):

https://cran.r-project.org/web/packages/treemapify/treemapify.pdf
https://cran.r-project.org/web/packages/treemapify/vignettes/introduction-to-treemapify.html

7.1.3 Visualizing many proportions at once

One simple example that you can test (first remember to install the package and load the library):

1. Let's read the .csv dataframe:
   https://raw.githubusercontent.com/selva86/datasets/master/proglanguages.csv

```
> Proglangs <-
read.csv("https://raw.githubusercontent.com/selva86/datasets/master/proglanguages.csv")
```

This dataframe contains a hierarchical list of programming languages:

```
'data.frame':          40 obs. of  4 variables:
 $ id    : chr  "Java (general)" "PHP (general)" "dotNet (general)" "Python (general)" ...
 $ value : int  423 253 220 219 185 121 91 89 83 79 ...
 $ parent: chr  "Java" "PHP" "dotNet" "Python" ...
 $ rank  : num  40 39 38 37 36 35 34 33 32 31 ...
```

7.1.3 Visualizing many proportions at once

This dataframe contains a hierarchical list of programming languages:

```
'data.frame':          40 obs. of  4 variables:
 $ id    : chr  "Java (general)" "PHP (general)" "dotNet (general)" "Python (general)" ...
 $ value : int  423 253 220 219 185 121 91 89 83 79 ...
 $ parent: chr  "Java" "PHP" "dotNet" "Python" ...
 $ rank  : num  40 39 38 37 36 35 34 33 32 31 ...
```

2. In order to create a treemap, the data must be converted to desired format using treemapify(). The important requirement is, you need to identify in your data:
   - One numerical continuous variable each describes the area of the tiles (‘value’)
   - One variable for fill color (‘parent’)
   - One variable that has the tile’s label (‘id’)
   - The parent group (‘parent’)

7.1.3 Visualizing many proportions at once

2. In order to create a treemap, the data must be converted to desired format using treemapify(). The important requirement is, you need to identify in your data:
   - One variable each describes the area of the tiles ('value')
   - One variable for fill color ('parent')
   - One variable that has the tile's label ('id')
   - The parent group ('parent')

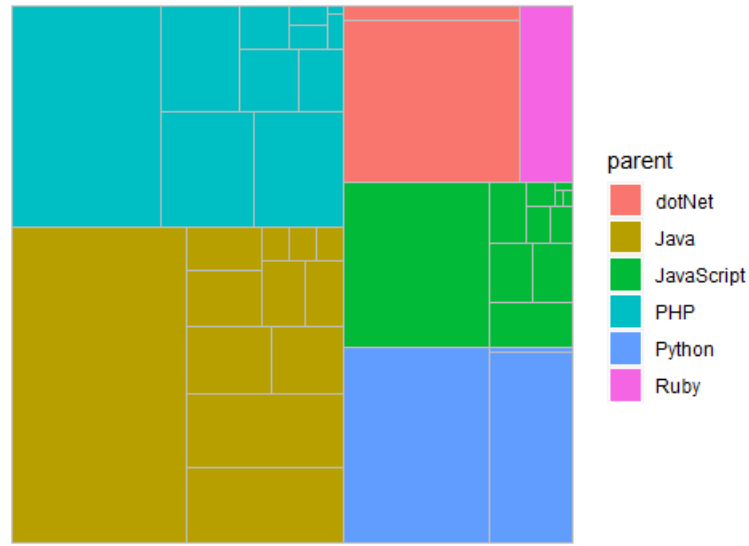
   And map them by using aes:

```
> ggplot(Proglangs, aes(area=value, fill=parent,
label=id, subgroup=parent)) + geom_treemap()
```

3. Use geom_treemap

7.1.3 Visualizing many proportions at once

```
> ggplot(Proglangs, aes(area=value, fill=parent,
label=id, subgroup=parent)) + geom_treemap()
```

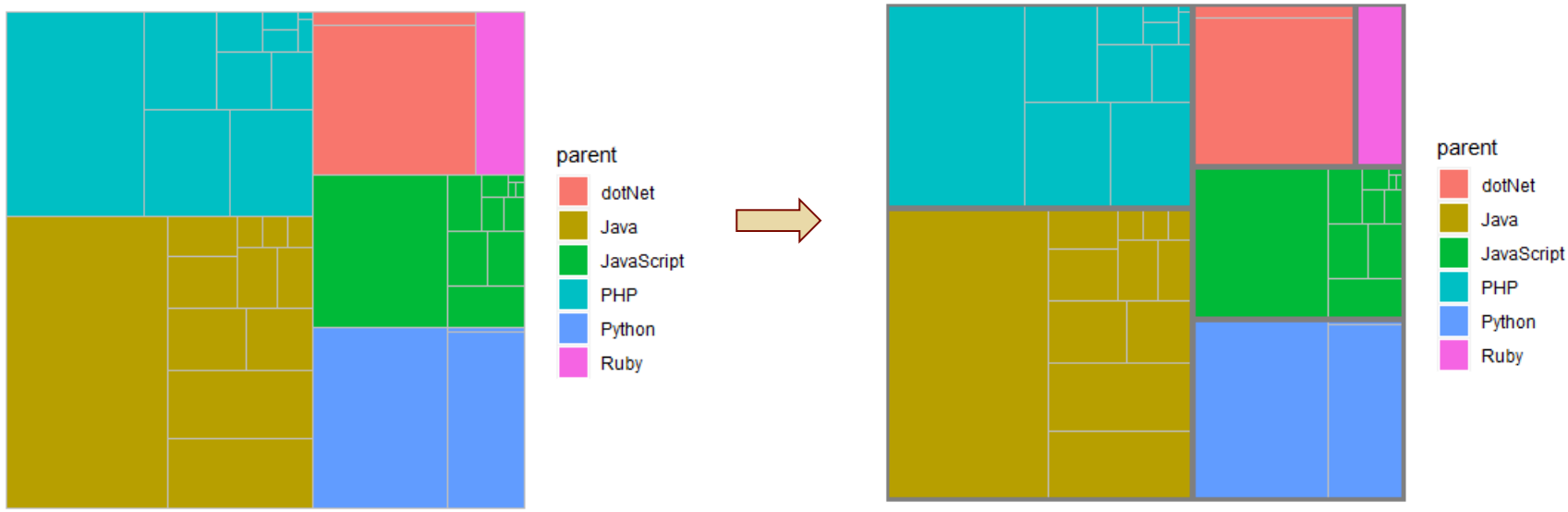


Remember that you can assign this to a variable and add the other optional layers afterwards

4. You can now add the main group bordering

```
> ggplot(Proglangs, aes(area=value, fill=parent,
label=id, subgroup=parent)) + geom_treemap()+
geom_treemap_subgroup_border()
```

7.1.3 Visualizing many proportions at once



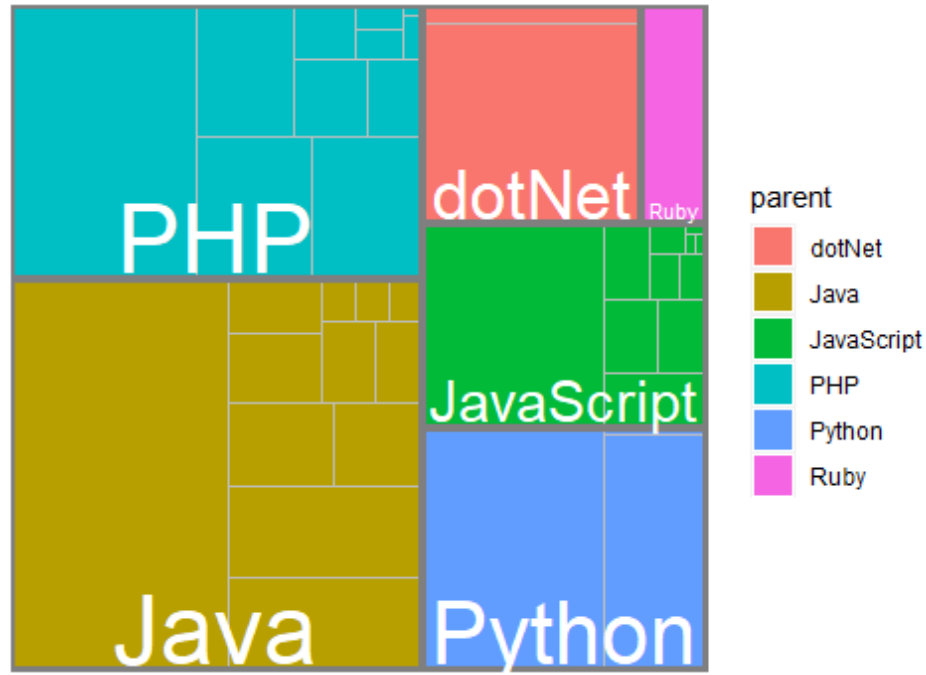
Thanks to:

`geom_treemap_subgroup_border()`

7.1.3 Visualizing many proportions at once

5. We can add subgroup heading in white

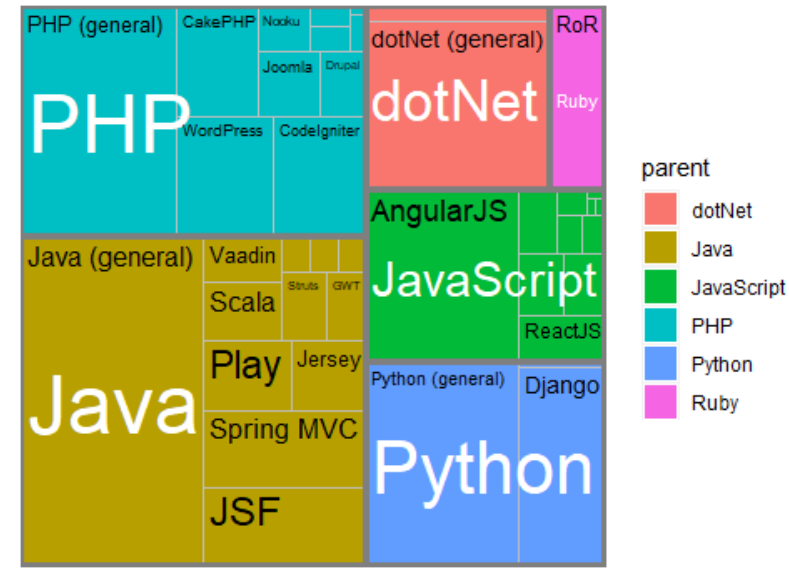
```
> ggplot(Proglangs, aes(area=value, fill=parent, label=id, subgroup=parent)) + geom_treemap()+ geom_treemap_subgroup_border()
+geom_treemap_subgroup_text(color='white')
```



7.1.3 Visualizing many proportions at once

6. Add all other text in black

```
> ggplot(Proglangs, aes(area=value,
fill=parent,label=id,subgroup=parent))+geom_treemap()+
geom_treemap_subgroup_border()+
+geom_treemap_subgroup_text(color="white",place="left")
+geom_treemap_text (aes(label=id))
```



More examples:
https://cran.r-project.org/web/packages/treemapify/vignettes/introduction-to-treemapify.html

7.1.4 Visualizing many relations at once

Many datasets contain **two or more quantitative variables,** and we may be **interested in how these variables relate to each other.**

Scatterplot matrix (SPLOM) uses **multiple scatterplots to determine the correlation** (if any) between a series of variables.

These scatterplots are then ***organized into a matrix*, making it easy to look at all the potential correlations in one place.**

7.1.4 Correlation between series variables

Bubble plot

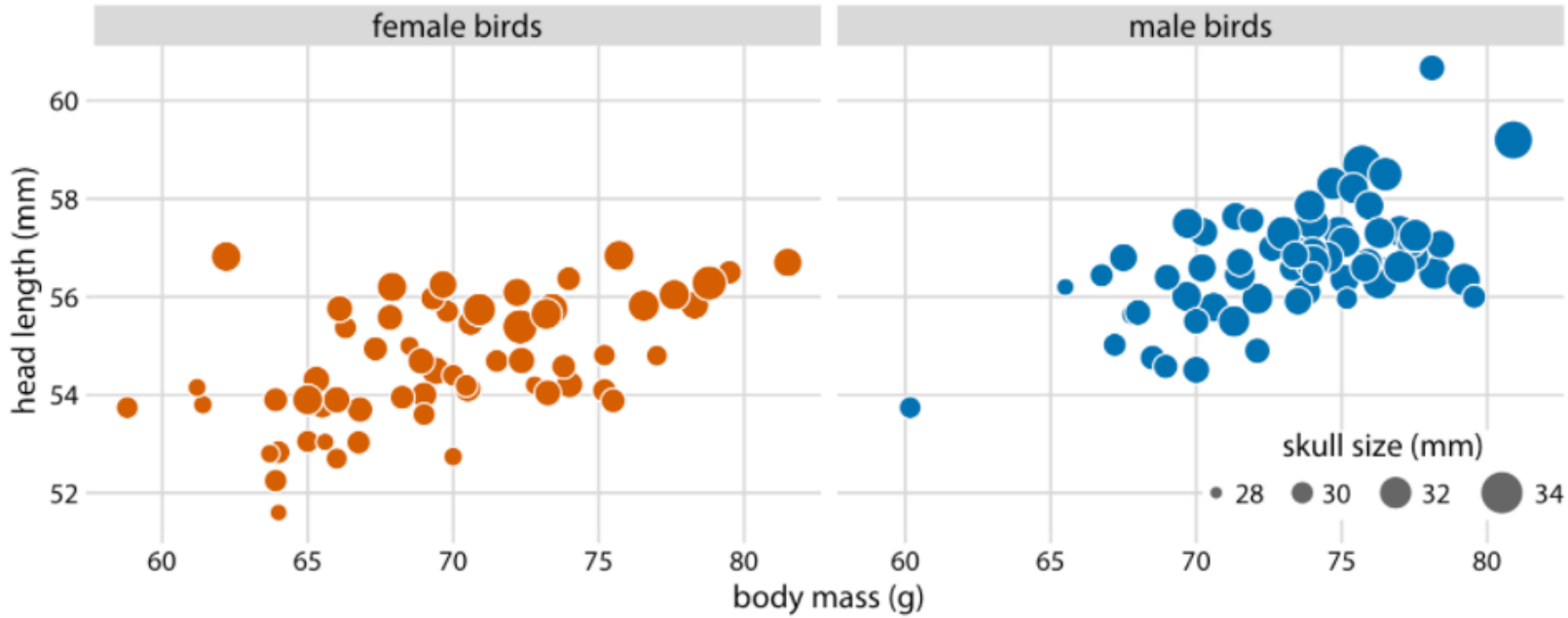


Figure 12.3: Head length versus body mass for 123 blue jays. The birds' sex is indicated by color, and the birds' skull size by symbol size. Head-length measurements include the length of the bill while skull-size measurements do not. Head length and skull size tend to be correlated, but there are some birds with unusually long or short bills given their skull size. Data source: Keith Tarvin, Oberlin College

Claus Wilke

7.1.4 Correlation between series variables

Scatterplot matrix (SPLOM) can provide answers to the following questions:

- Are there **pair wise relationships** between the variables?
- If there are relationships, what is **the nature of these relationships**?
- Are there **outliers** in the data?
- Is there **clustering by groups** in the data?

7.1.4 Scatterplot Matrix

More SPLOM examples

Thanks for your attention!