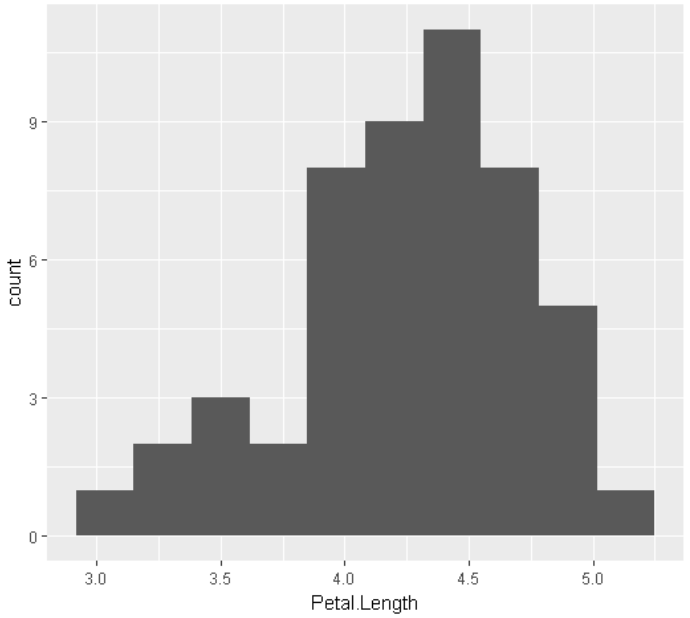
RenR 480/711

Exploratory graphics

Histogram

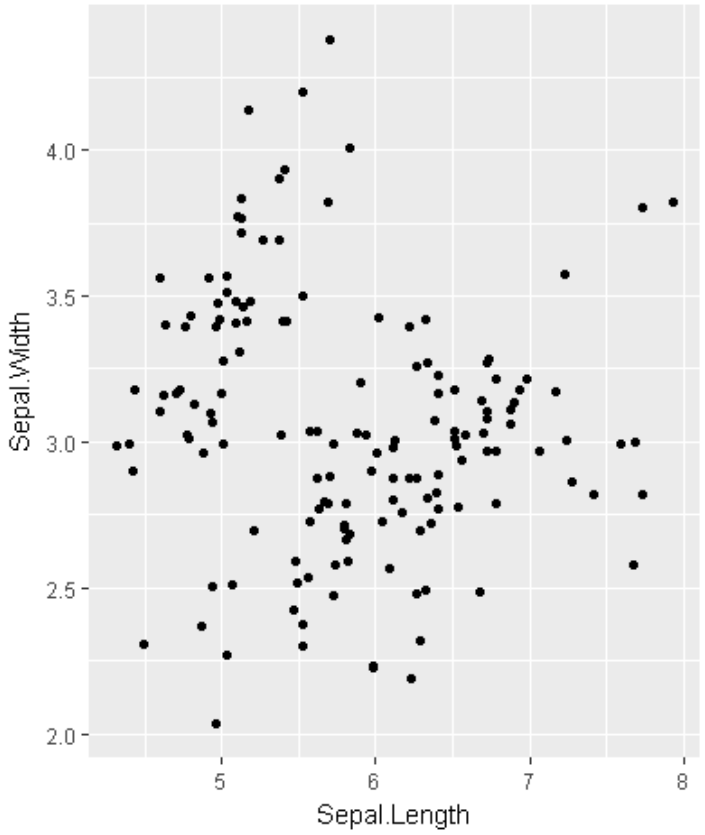
By Cliff from Arlington, Virginia, USA - Blue Flag (Iris versicolor)Uploaded by Jacopo Werther, CC BY 2.0, https://commons.wikimedia.org/w/index.php?curid=25492191

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	5.4	3.7	1.5	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa
13	4.8	3.0	1.4	0.1	setosa
14	4.3	3.0	1.1	0.1	setosa
15	5.8	4.0	1.2	0.2	setosa
16	5.7	4.4	1.5	0.4	setosa
17	5.4	3.9	1.3	0.4	setosa
18	5.1	3.5	1.4	0.3	setosa
19	5.7	3.8	1.7	0.3	setosa



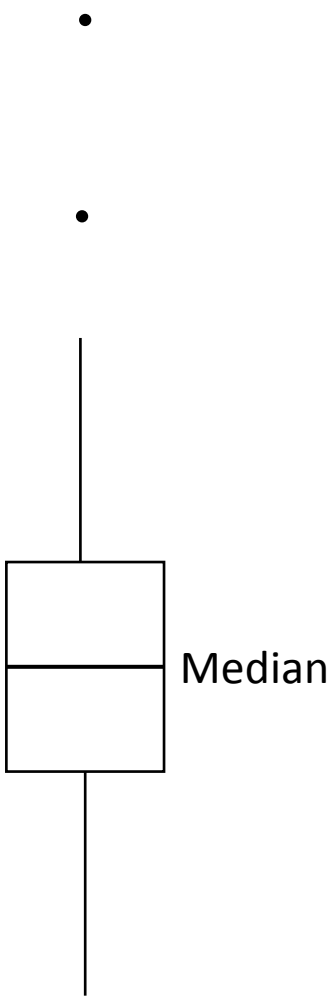
Histograms show the distribution

Scatter plot

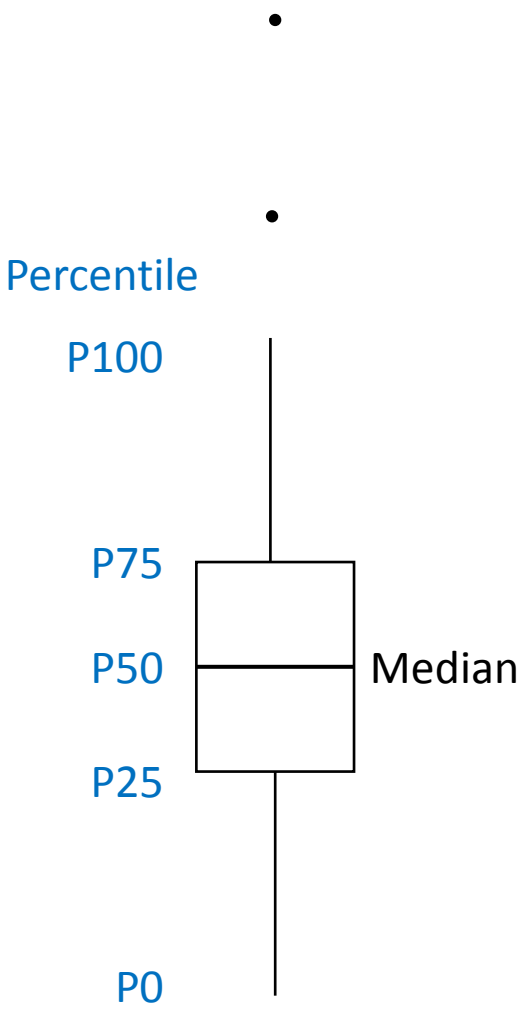


Scatter plots show relationships between variables

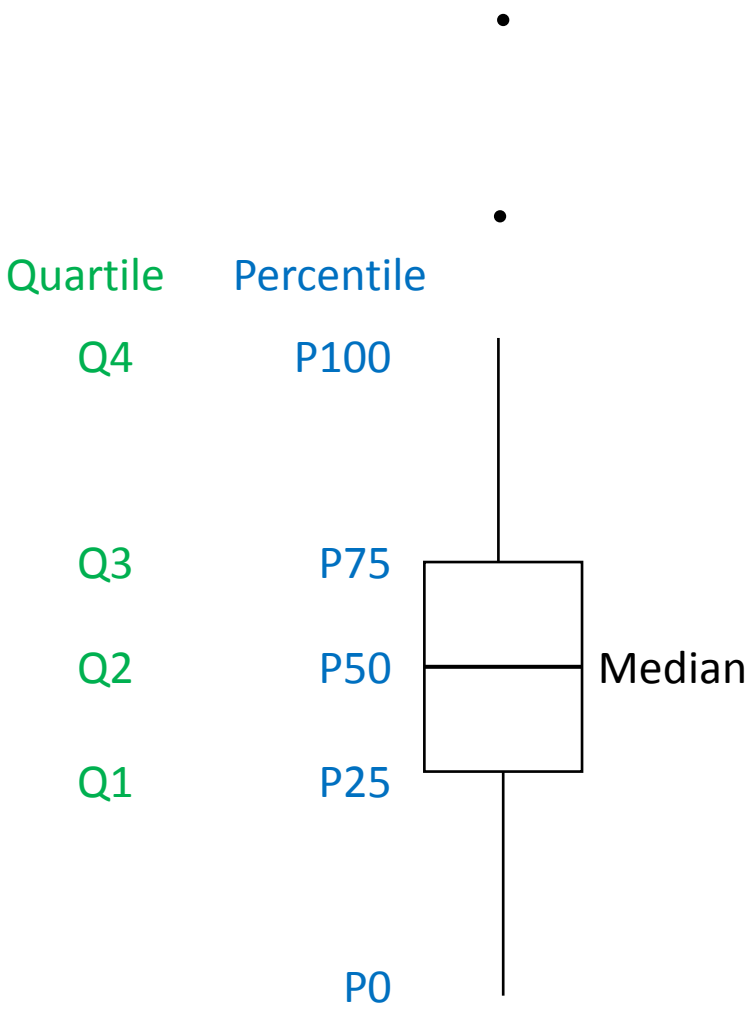
Boxplot



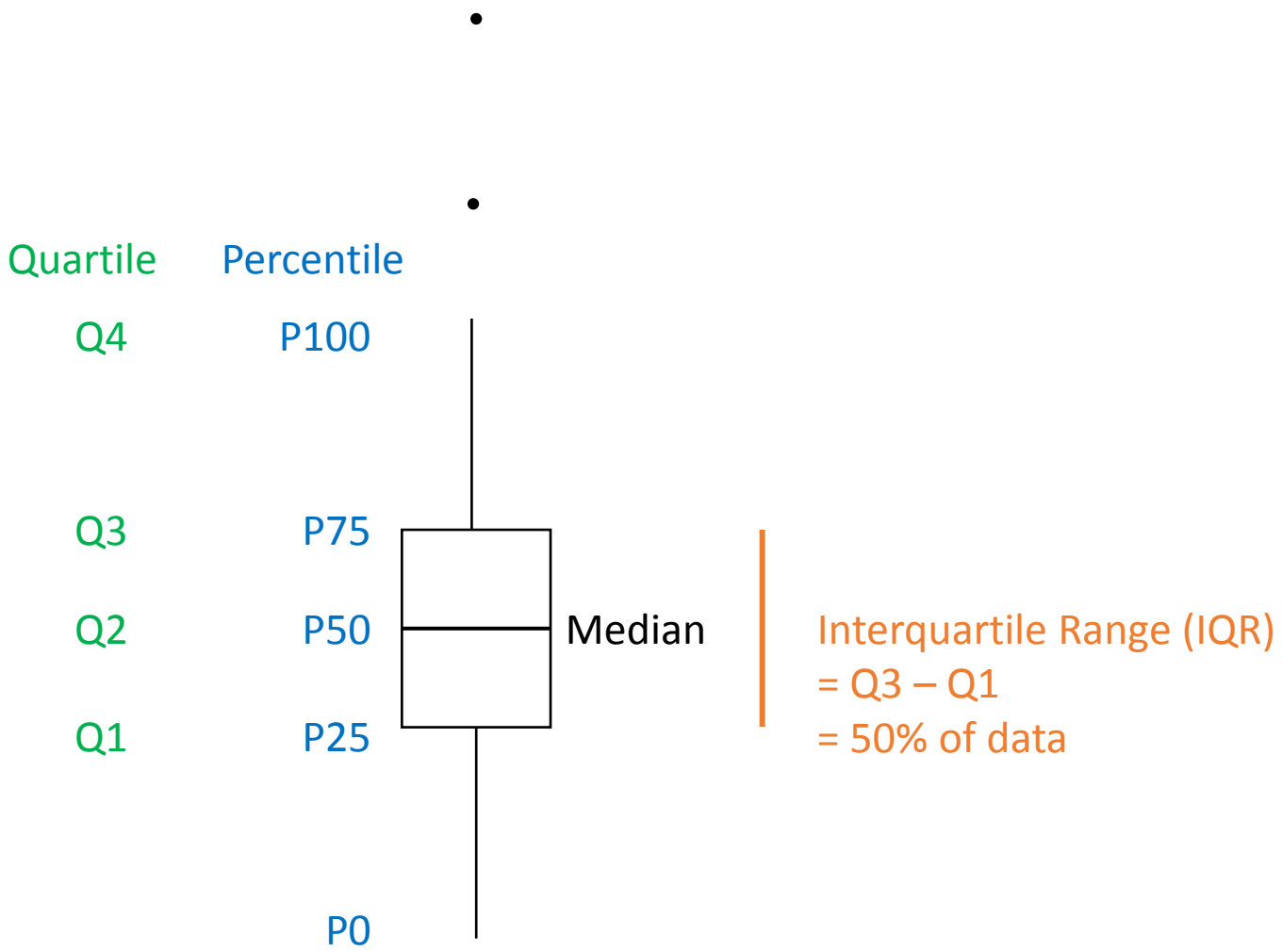
Boxplot



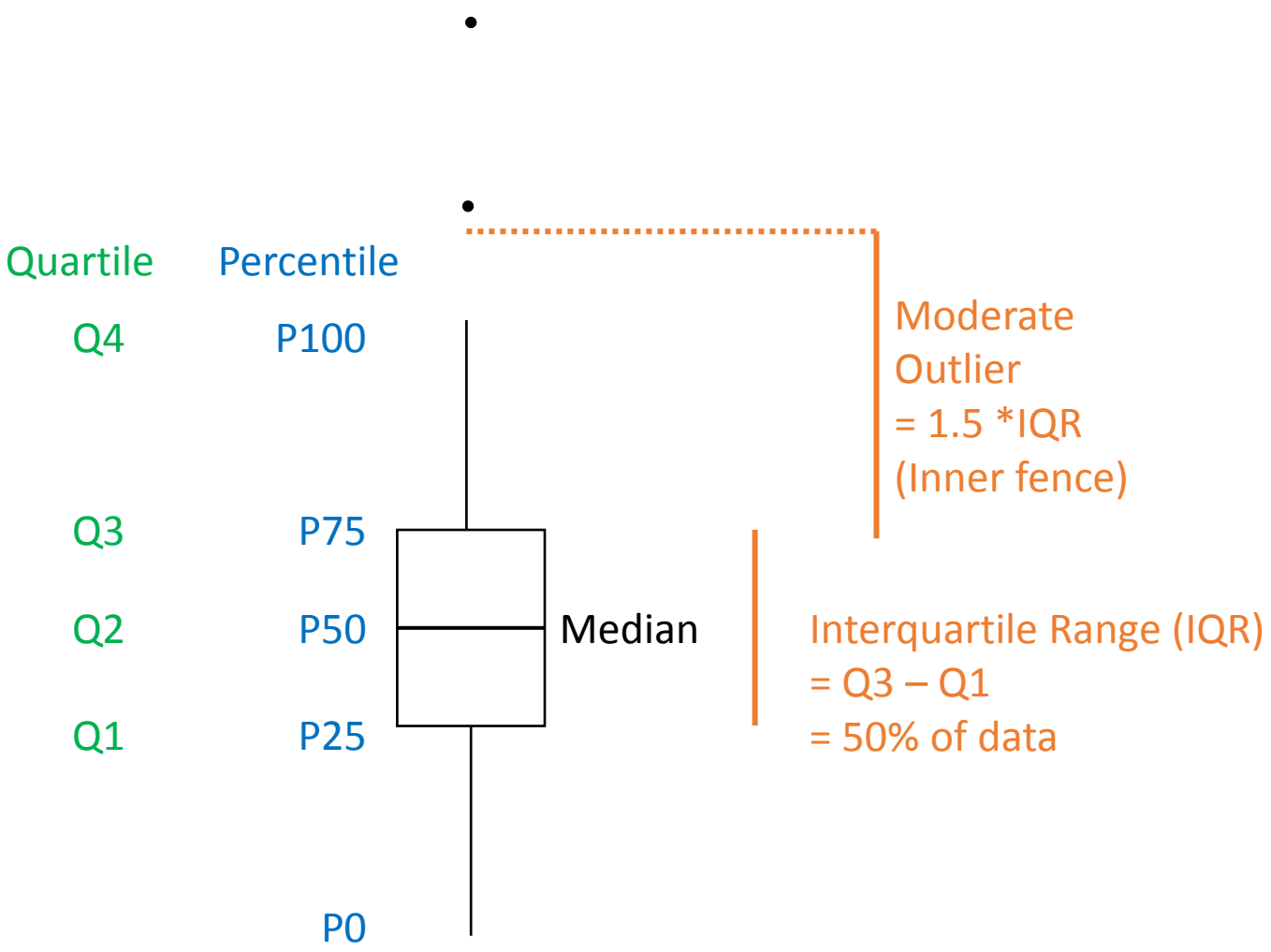
Boxplot



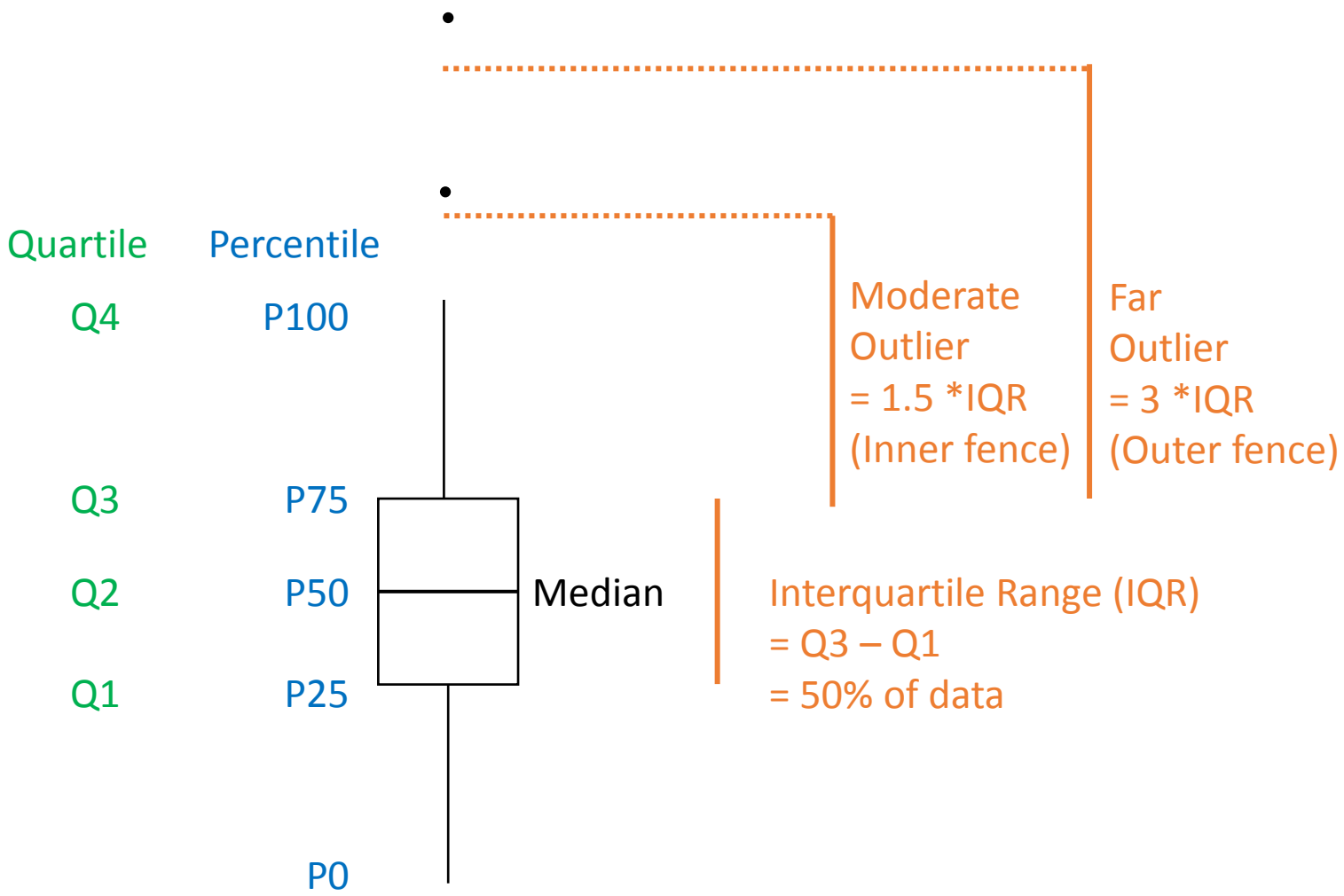
Boxplot


Quartile
Percentile
Q4
P100
Q3
P75
Q2
P50
Median
Q1
P25
P0
Interquartile Range (IQR)
= Q3 – Q1
= 50% of data

Boxplot


Quartile
Percentile
Q4
P100
Q3
P75
Q2
P50
Median
Q1
P25
P0
Moderate
Outlier
= 1.5 *IQR
(Inner fence)
Interquartile Range (IQR)
= Q3 – Q1
= 50% of data

Boxplot

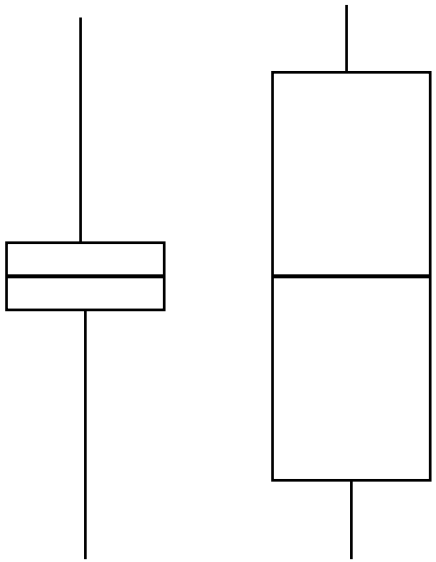

Quartile
Percentile
Q4
P100
Q3
P75
Q2
P50
Median
Q1
P25
P0
Moderate Outlier = 1.5 *IQR (Inner fence)
Far Outlier = 3 *IQR (Outer fence)
Interquartile Range (IQR) = Q3 – Q1 = 50% of data

Boxplot

Measurement of:

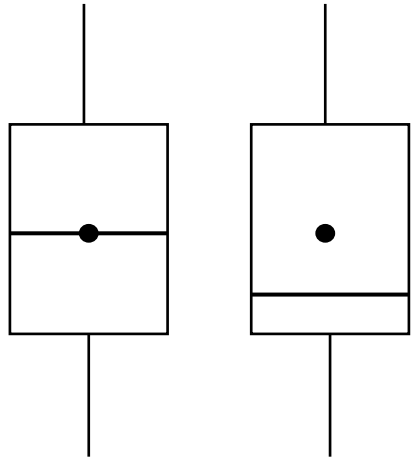
- Center of the data (central tendency)
- Spread
- Shape

Kurtosis

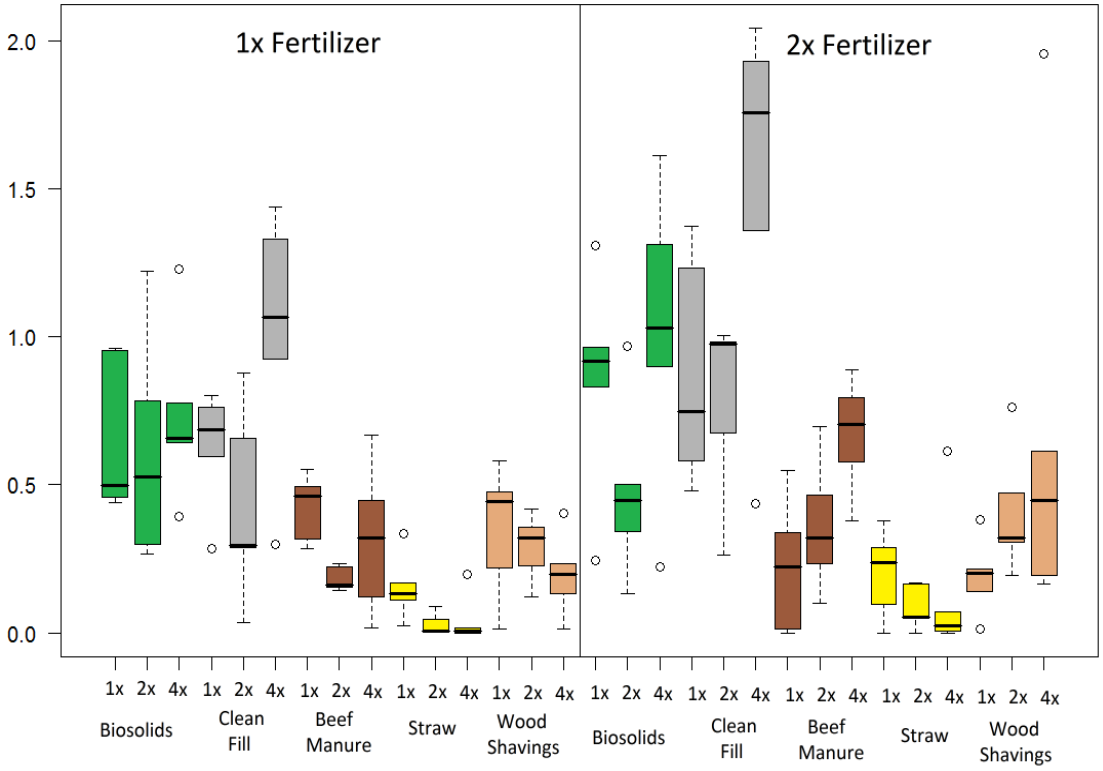


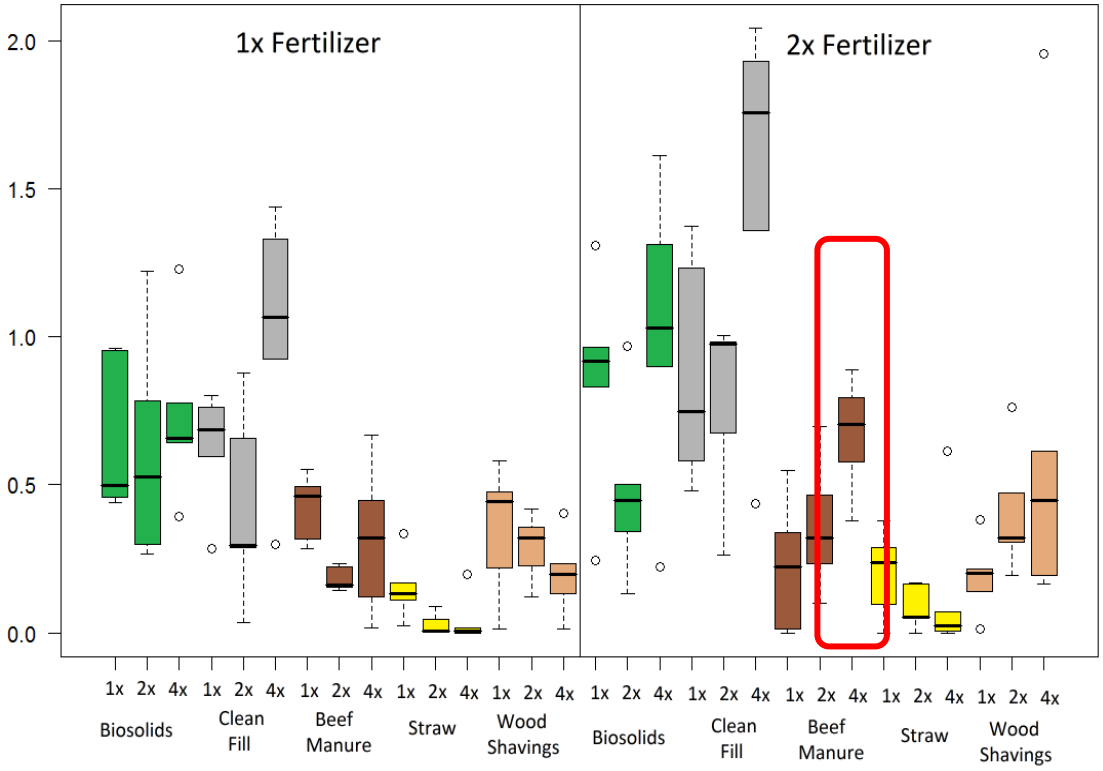
Narrow Wide

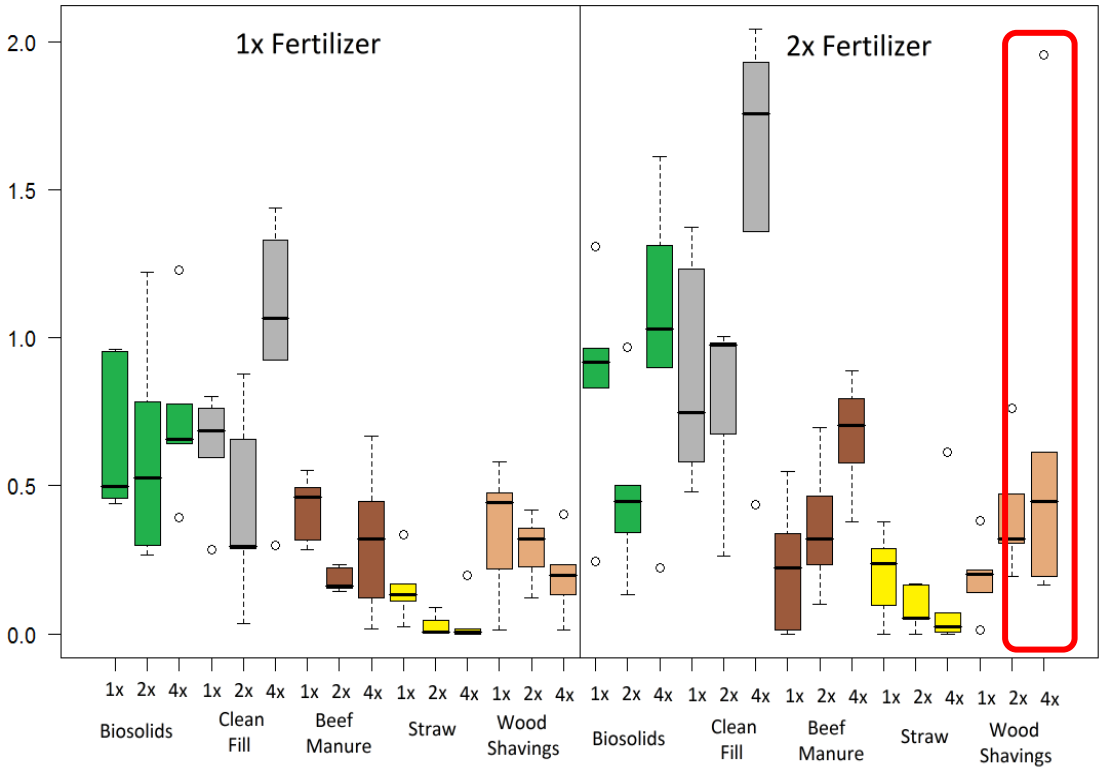
Symmetry



Symmetric Asymmetric


1x Fertilizer
2x Fertilizer
2.0
1.5
1.0
0.5
0.0
1x 2x 4x
Biosolids
Clean Fill
Beef Manure
Straw
Wood Shavings


1x Fertilizer
2x Fertilizer
2.0
1.5
1.0
0.5
0.0
1x 2x 4x
Biosolids
Clean Fill
Beef Manure
Straw
Wood Shavings


1x Fertilizer
2x Fertilizer
2.0
1.5
1.0
0.5
0.0
1x 2x 4x
Biosolids
Clean Fill
Beef Manure
Straw
Wood Shavings

Exploratory graphics in R

GGplot2 package

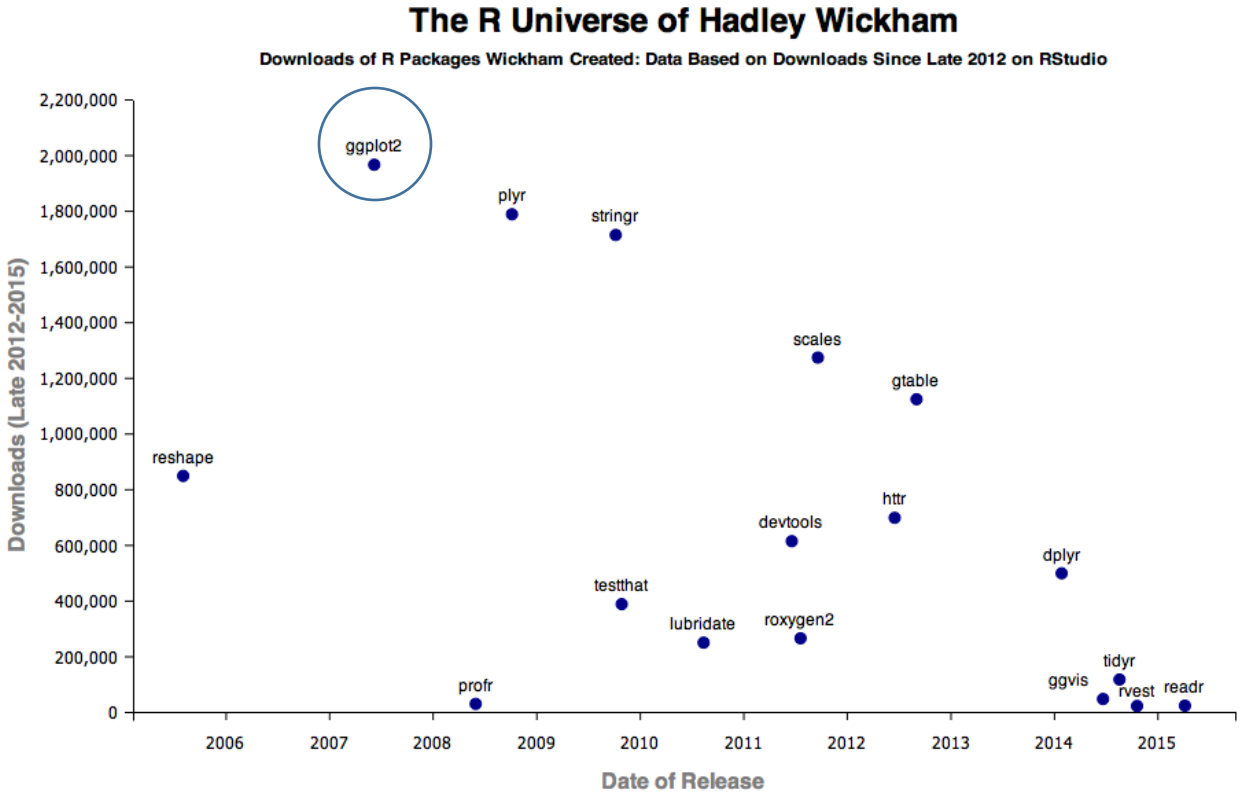
Introduction: ggplot2

- Developed by Hadley Wickham
  - NZ data scientist; produces several popular R packages
  - Responsible for the “tidy-verse” to clean up the R syntax ecosystem

https://priceonomics.com/hadley-wickham-the-man-who-revolutionized-r/

Introduction: ggplot2

- Developed by Hadley Wickham
  - NZ data scientist; produces several popular R packages
  - Responsible for the “tidy-verse” to clean up the R syntax ecc



https://priceonomics.com/hadley-wickham-the-man-who-revolutionized-r/

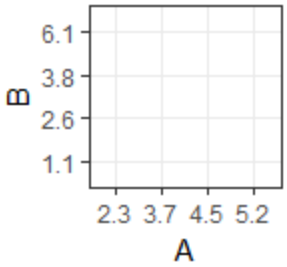
Introduction: ggplot2

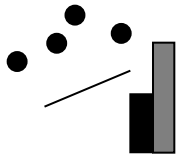
- “GG” stands for “grammar of graphics”
- A package offering efficient coding to produce beautiful graphs

General ggplot2 syntax

Data	+ Coordinate System	+ Data Visualization ("Geom")	= Plot

A	B	C	D
5.2	6.1	1.5	X
4.5	3.8	2.0	X
3.7	2.6	1.4	Y
2.3	1.1	0.8	Z





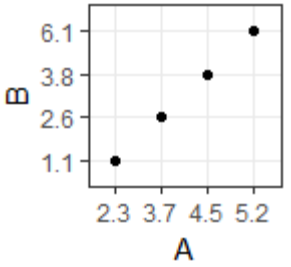


fig1 = **ggplot(data=*data*, mapping = aes(*mappings*)) +**

geom_*function*(stat = *stat* , position = *position* **) +**

coordinate_*function*() +

facet_*function*() +

scale_*function*() +

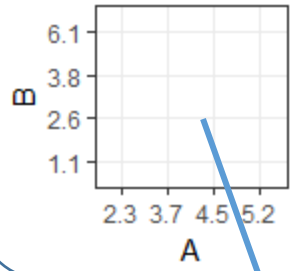
theme_*function*()

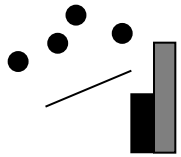
fig1

General ggplot2 syntax

Data + Coordinate System + Data Visualization ("Geom") = Plot

A	B	C	D
5.2	6.1	1.5	X
4.5	3.8	2.0	X
3.7	2.6	1.4	Y
2.3	1.1	0.8	Z





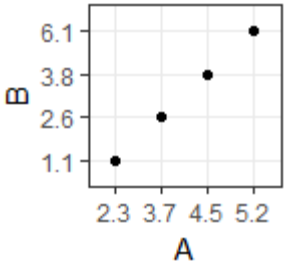


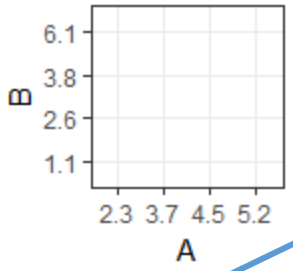
fig1 = **ggplot(data=*data*, mapping = aes(*mappings*)) +**
geom_*function*(stat = *stat* , position = *position* **) +**
coordinate_*function*() +
facet_*function*() +
scale_*function*() +
theme_*function*()

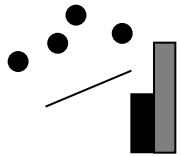
fig1

General ggplot2 syntax

Data + Coordinate System + Data Visualization ("Geom") = Plot

A	B	C	D
5.2	6.1	1.5	X
4.5	3.8	2.0	X
3.7	2.6	1.4	Y
2.3	1.1	0.8	Z





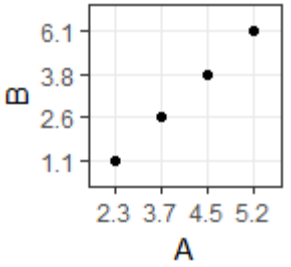


fig1 = **ggplot(data=*data*, mapping = aes(*mappings*)) +**

geom_*function*(stat = *stat* , position = *position* **) +**

coordinate_*function*() +

facet_*function*() +

scale_*function*() +

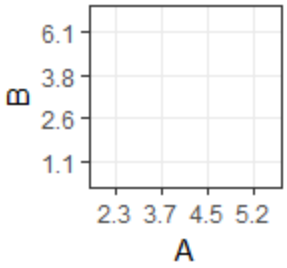
theme_*function*()

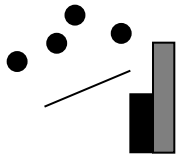
fig1

General ggplot2 syntax

Data + Coordinate System + Data Visualization ("Geom") = Plot

A	B	C	D
5.2	6.1	1.5	X
4.5	3.8	2.0	X
3.7	2.6	1.4	Y
2.3	1.1	0.8	Z





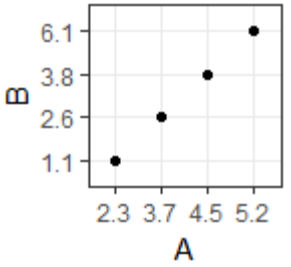


fig1 = **ggplot(data=*data*, mapping = aes(*mappings*)) +**

geom_*function*(stat = *stat* , position = *position* **) +**

coordinate_*function*() +

facet_*function*() +

scale_*function*() +

theme_*function*()

fig1

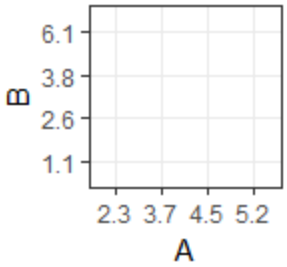
Required

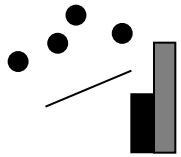
Optional

General ggplot2 syntax

Data + Coordinate System + Data Visualization ("Geom") = Plot

A	B	C	D
5.2	6.1	1.5	X
4.5	3.8	2.0	X
3.7	2.6	1.4	Y
2.3	1.1	0.8	Z





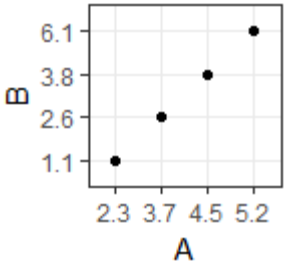


fig1 = **ggplot(data=*data*, mapping = aes(*mappings*)) +**

geom_*function*(stat = *stat* , position = *position* **) +**

coordinate_*function*() +

facet_*function*() +

scale_*function*() +

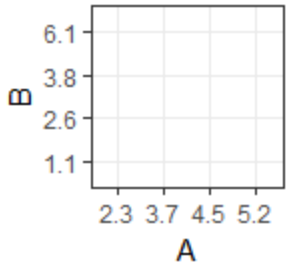
theme_*function*()

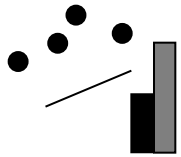
fig1

General ggplot2 syntax

Data	+ Coordinate System	+ Data Visualization ("Geom")	= Plot

A	B	C	D
5.2	6.1	1.5	X
4.5	3.8	2.0	X
3.7	2.6	1.4	Y
2.3	1.1	0.8	Z





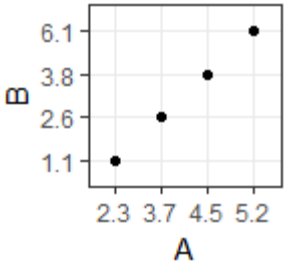


fig1 = **ggplot(data=*data*, mapping = aes(*mappings*)*) +**

geom_*function*(stat = *stat*, position = *position* **) +**

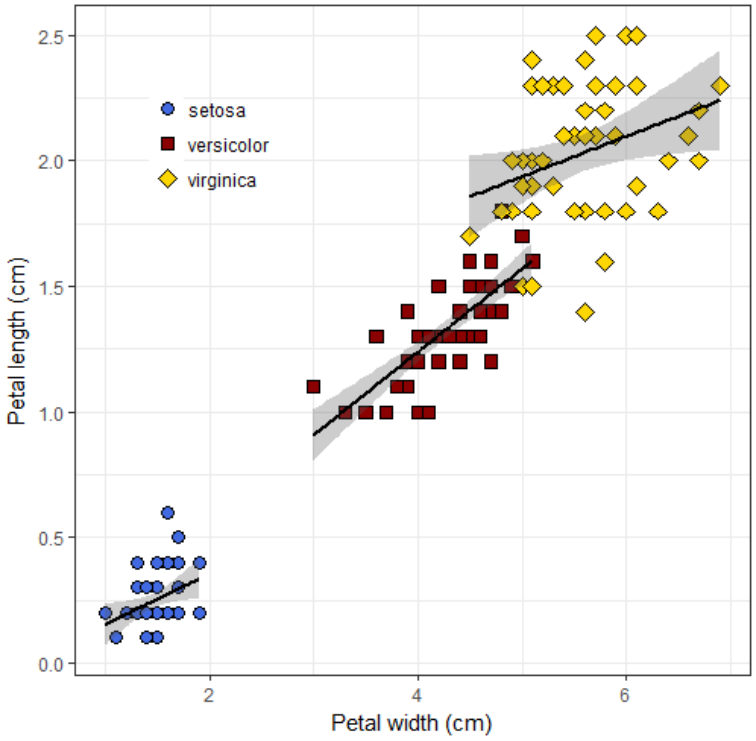
coordinate_*function*() +

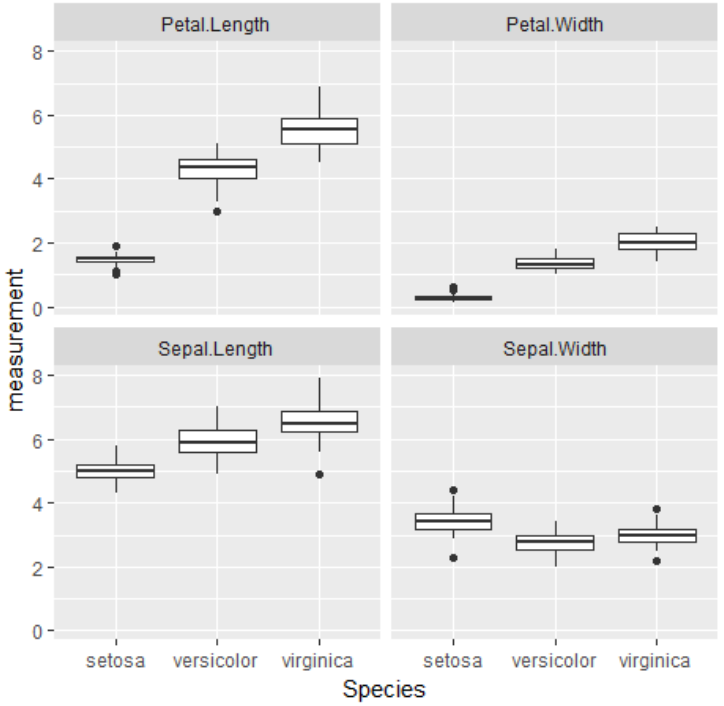
facet_*function*() +

scale_*function*() +

theme_*function*()

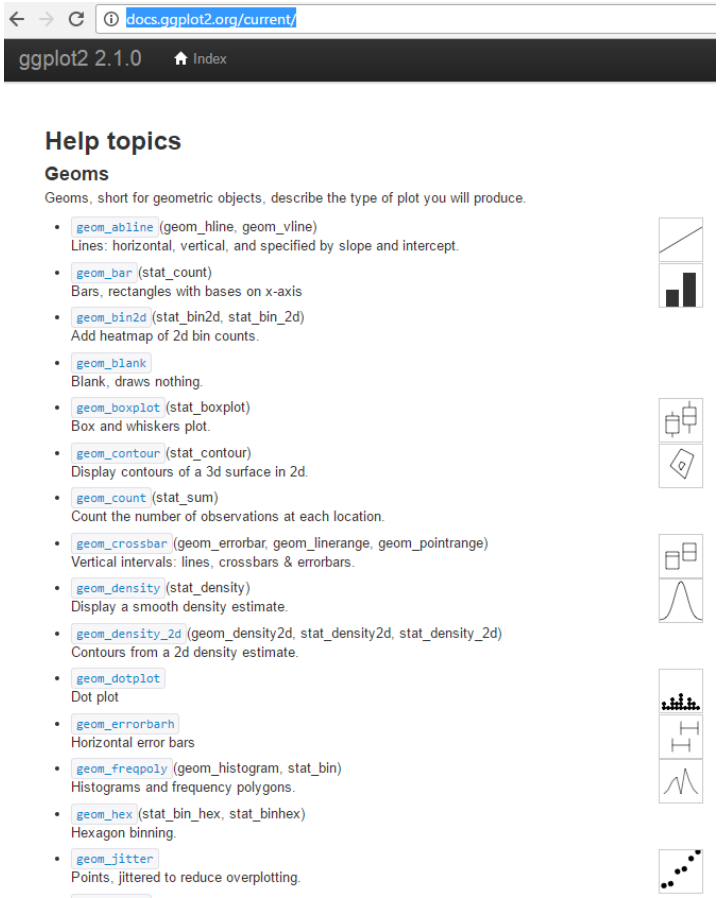
fig1


setosa
versicolor
virginica
Petal length (cm)
Petal width (cm)
0.0
0.5
1.0
1.5
2.0
2.5
2
4
6

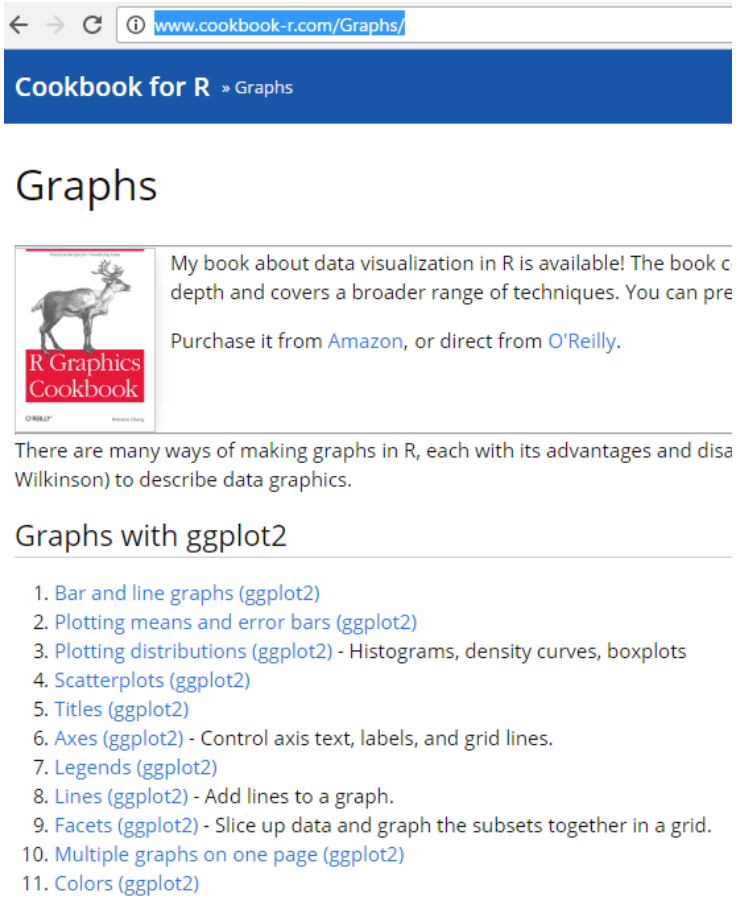

Petal.Length
Petal.Width
Sepal.Length
Sepal.Width
measurement
0
2
4
6
8
setosa
versicolor
virginica
Species

Other resources:

Ggplot2 website:
http://docs.ggplot2.org/current/



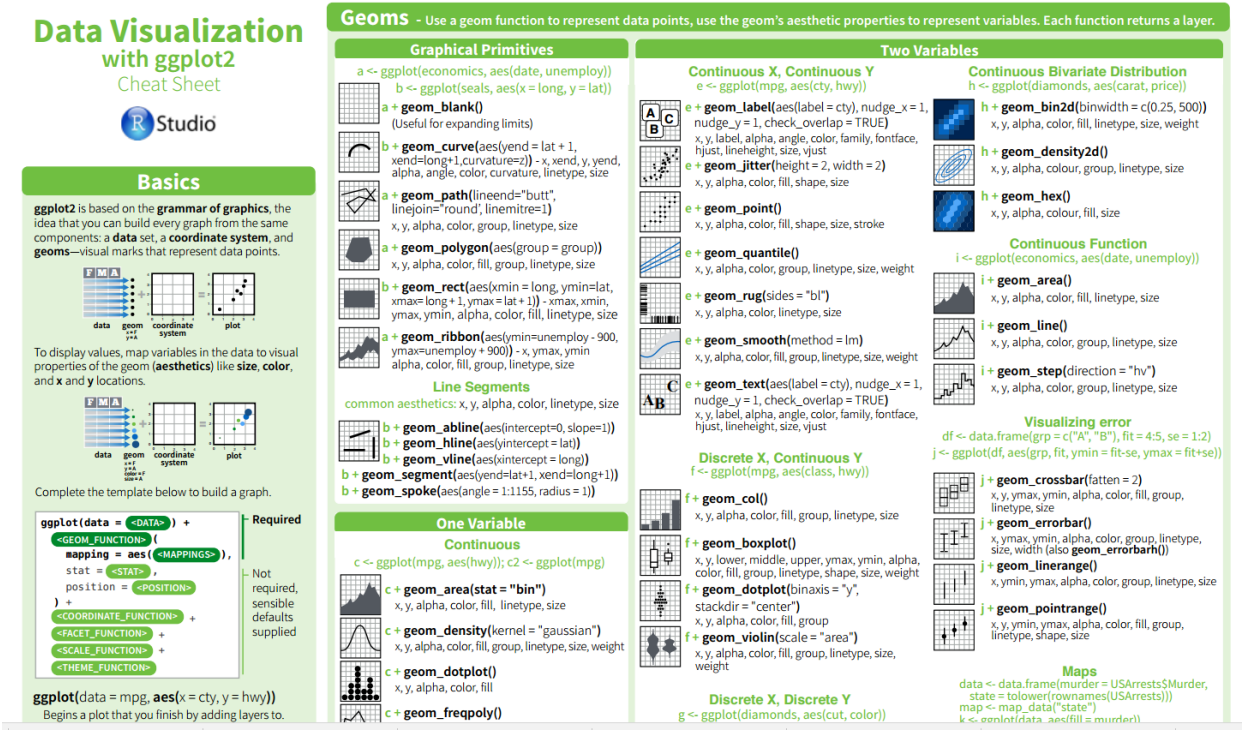
R Cookbook (graphs):
http://www.cookbook-r.com/Graphs/



Other resources:

ggplot2 cheatsheet

https://www.rstudio.com/wp-content/uploads/2016/11/ggplot2-cheatsheet-2.1.pdf



Data Visualization with ggplot2 Cheat Sheet

RStudio

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.

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

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.

Geoms - Use a geom function to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.

Graphical Primitives

a <- ggplot(economics, aes(date, unemploy))
b <- ggplot(seals, aes(x = long, y = lat))

a + **geom_blank()**
(Useful for expanding limits)

b + **geom_curve**(aes(yend = lat + 1, xend=long+1,curvature=z)) - x, xend, y, yend, alpha, angle, color, curvature, linetype, size

a + **geom_path**(lineend="butt", linejoin="round", linemitre=1)
x, y, alpha, color, group, linetype, size

a + **geom_polygon**(aes(group = group))
x, y, alpha, color, fill, group, linetype, size

b + **geom_rect**(aes(xmin = long, ymin=lat, xmax= long + 1, ymax = lat + 1)) - xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size

a + **geom_ribbon**(aes(ymin=unemploy - 900, ymax=unemploy + 900)) - x, ymax, ymin
alpha, color, fill, group, linetype, size

Line Segments

common aesthetics: x, y, alpha, color, linetype, size

b + **geom_abline**(aes(intercept=0, slope=1))
b + **geom_hline**(aes(yintercept = lat))
b + **geom_vline**(aes(xintercept = long))
b + **geom_segment**(aes(yend=lat+1, xend=long+1))
b + **geom_spoke**(aes(angle = 1:1155, radius = 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**()

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

Discrete X, Discrete Y

g <- ggplot(diamonds, aes(cut, color))

Continuous Bivariate Distribution

h <- ggplot(diamonds, aes(carat, price))

h + **geom_bin2d**(binwidth = c(0.25, 500))
x, y, alpha, color, fill, linetype, size, weight

h + **geom_density2d**()
x, y, alpha, colour, group, linetype, size

h + **geom_hex**()
x, y, alpha, colour, fill, size

Continuous Function

i <- ggplot(economics, aes(date, unemploy))

i + **geom_area**()
x, y, alpha, color, fill, linetype, size

i + **geom_line**()
x, y, alpha, color, group, linetype, size

i + **geom_step**(direction = "hv")
x, y, alpha, color, group, linetype, size

Visualizing error

df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2)
j <- ggplot(df, aes(grp, fit, ymin = fit-se, ymax = fit+se))

j + **geom_crossbar**(fatten = 2)
x, y, ymax, ymin, alpha, color, fill, group, linetype, size

j + **geom_errorbar**()
x, ymax, ymin, alpha, color, group, linetype, size, width (also **geom_errorbarh()**)

j + **geom_linerange**()
x, ymin, ymax, alpha, color, group, linetype, size

j + **geom_pointrange**()
x, y, ymin, ymax, alpha, color, fill, group, linetype, shape, size

Maps

data <- data.frame(murder = USArrests$Murder, state = tolower(rownames(USArrests)))
map <- map_data("state")
k <- ggplot(data, aes(fill = murder))