

# R Tutorial

May 14<sup>th</sup>, 2024

Kith Pradhan, Ph.D.

Assistant professor

Department of Epidemiology & Population Health (Biostatistics)

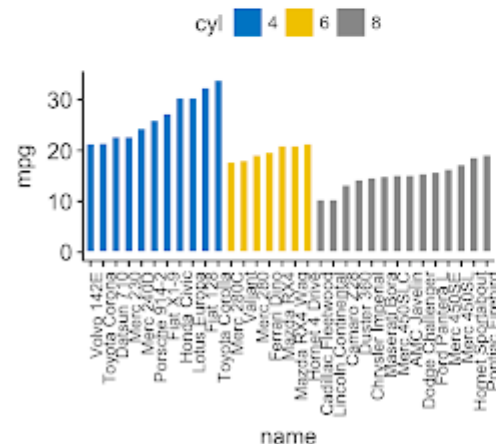
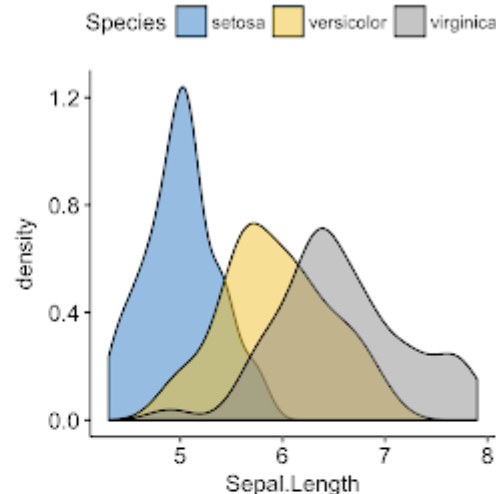
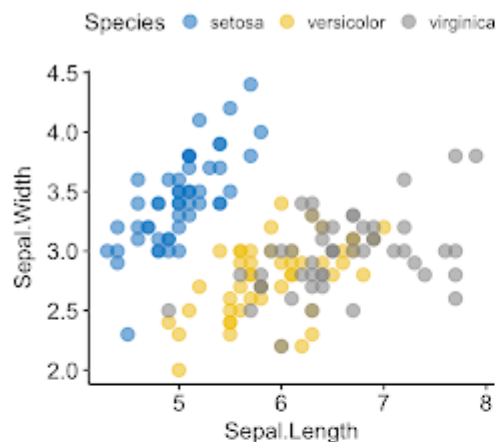
[kith.pradhan@einsteinmed.org](mailto:kith.pradhan@einsteinmed.org)



# ggplot2

# ggplot2

- A graphics package completely separate from base R
- Many cool features and beautiful plots
- Tricky syntax

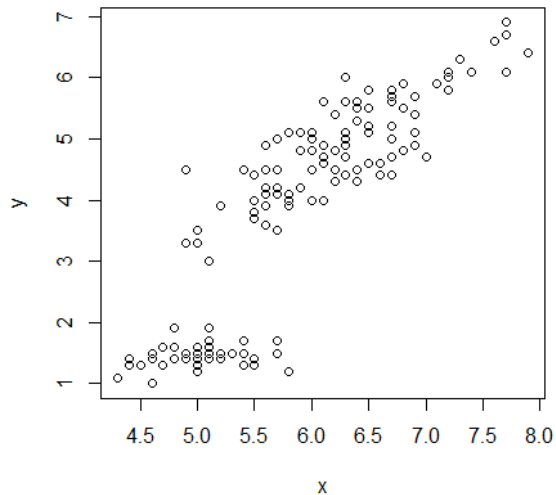


Species	length	mean	sd
setosa	50	5.01	0.352
versicolor	50	5.94	0.516
virginica	50	6.59	0.636

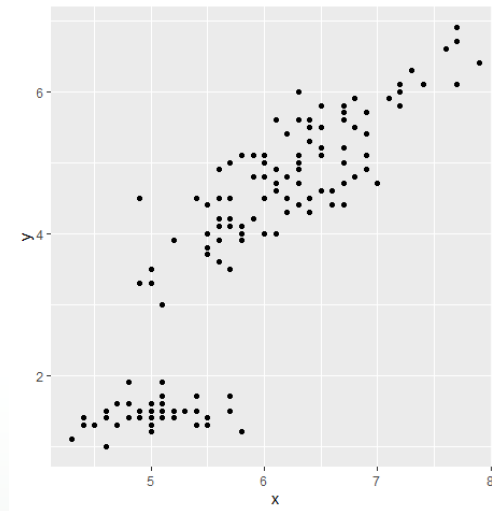
iris data set gives the measurements in cm of the variables sepal length and width and petal length and width, respectively, for 50 flowers from each of 3 species of iris. The species are *Iris setosa*, *versicolor*, and *virginica*.

# Base graphics vs ggplot2

- `plot(x, y)`



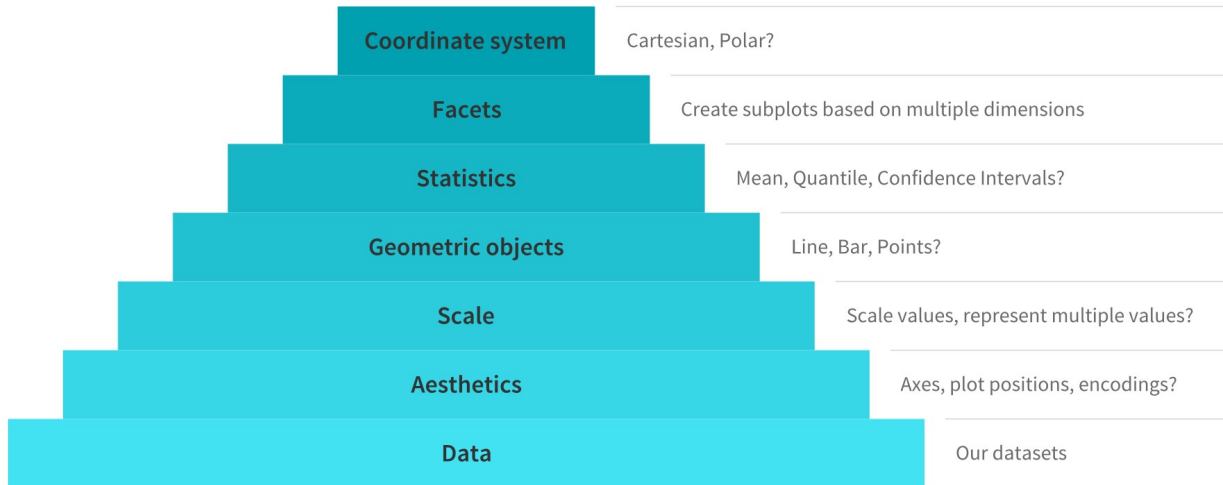
- `dat = data.frame(x=x, y=y)`
- `ggplot(dat, aes(x=x, y=y)) + geom_point()`





# ggplot2

## Major Components of the Grammar of Graphics



- Based on a “grammar of graphics”
  - The idea that any visualization can be generalized into a framework of layered components.
- The three most important concepts:
  - 1. Data: what you’re trying to plot
  - 2. Geom: the type of plot
  - 3. Aesthetic: the mapping from data to the various features of a plot

# ggplot2

## 1. data

- To use ggplot, you organize your data into a single data.frame
- To keep things simple, think of it like an excel matrix
  - 1 subject per row
  - The columns contain the info about the subjects

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.0	180	3.07	4.070	17.40	0	0	3	2

# ggplot2

## 2. geom

- A “geom” is the type of plot you want to show



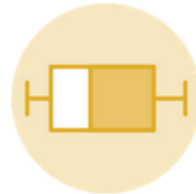
Violin



Density



Histogram



Boxplot



Ridgeline



Scatter



Heatmap



Correlogram



Bubble



Connected scatter

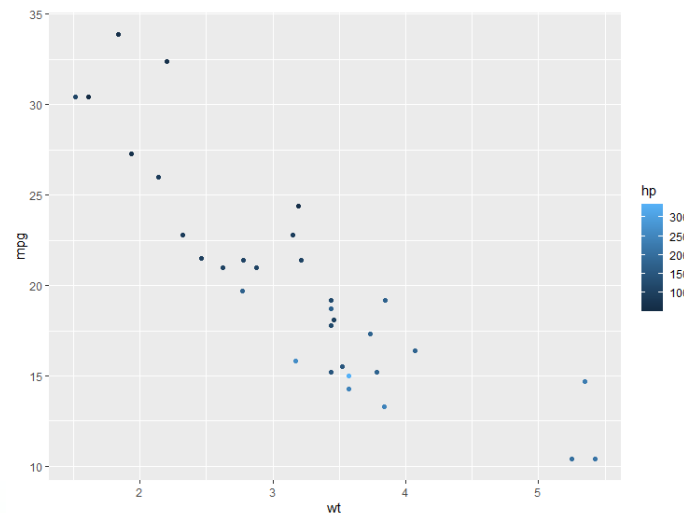


Density 2d

# ggplot2

## 2. geom

- Each geom is defined by certain attributes
- `geom_point` (simple scatter plot) has 7
  - X (horizontal axis)
  - Y (vertical axis)
  - Alpha (the transparency of the points)
  - Color (the color of the point's outline)
  - Fill (the color of the points inside)
  - Shape (what the point looks like)
  - Size (how big the point is)



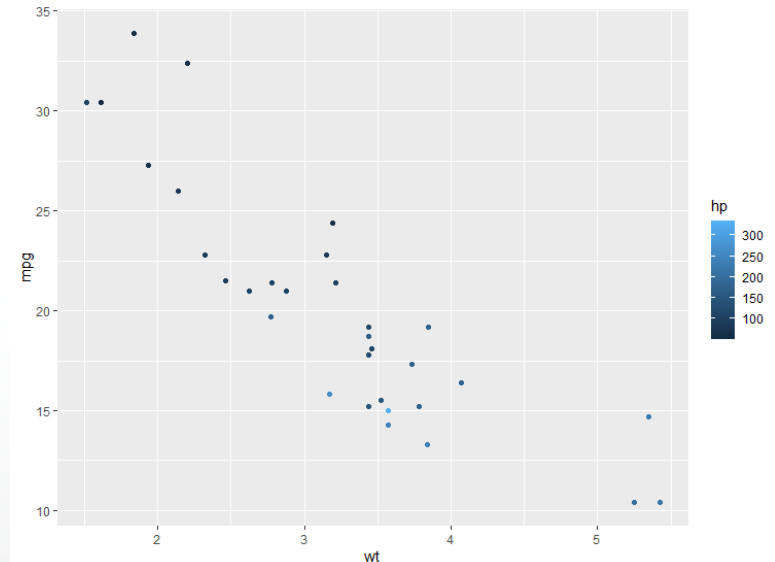


# ggplot2

## 3. aesthetic

- To work with ggplot, you link your “data” to the “geom” through an “aesthetic”
  - wt → x-axis
  - mpg → y-axis
  - hp → color

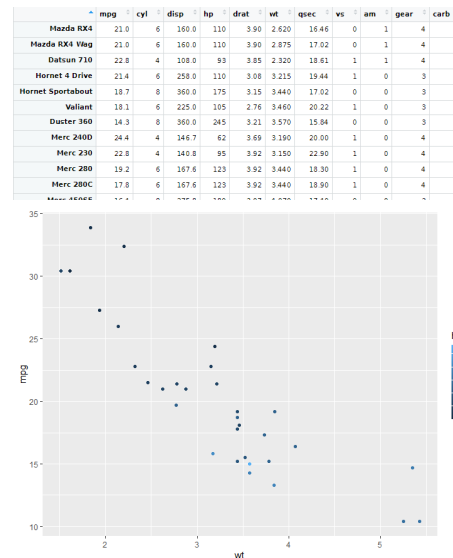
	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Merc 450SE	16.4	8	275.0	180	3.07	4.070	17.16	0	0	3	2



# ggplot2

## 3. aesthetic

- To work with ggplot, you link your “data” to the “geom” through an “aesthetic”
  - wt → x-axis
  - mpg → y-axis
  - hp → color



R:

```
ggplot(mtcars, aes(x=wt, y=mpg, color=hp)) + geom_point()
```

# Ggplot2: code

- `ggplot(mtcars, aes(x=wt, y=mpg, color=hp)) + geom_point()`

Data

Aesthetic

Geom

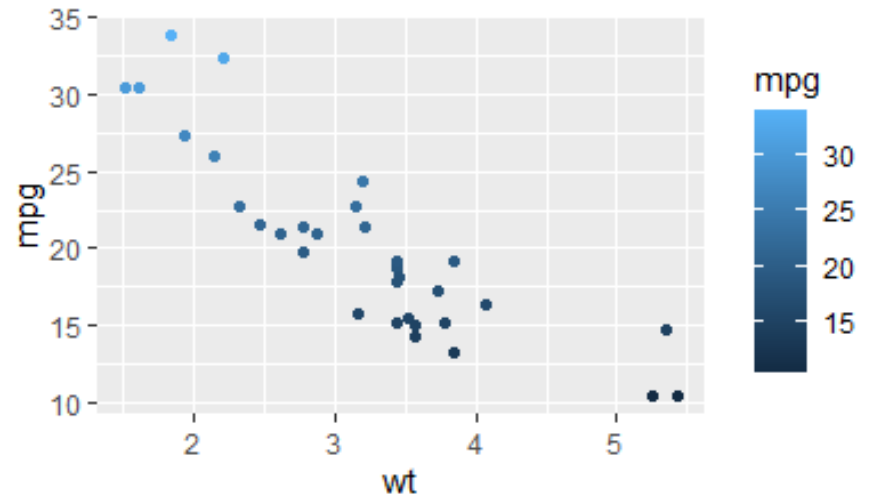
- You can put it all in one line, or you can break it up into multiple lines, building as you go

```
g1 = ggplot(mtcars)
g2 = g1 + aes(x=wt, y=mpg, color=mpg)
g3 = g2 + geom_point()
g3
```

# Ggplot2: code

- The '+' here doesn't mean mathematical addition. It's an overloaded operator that makes coding a bit easier to type out and read.
- In a sense, you're adding information and layers to an increasingly complex plot
- It's sometimes easier to write and debug your code when you break it down like this.

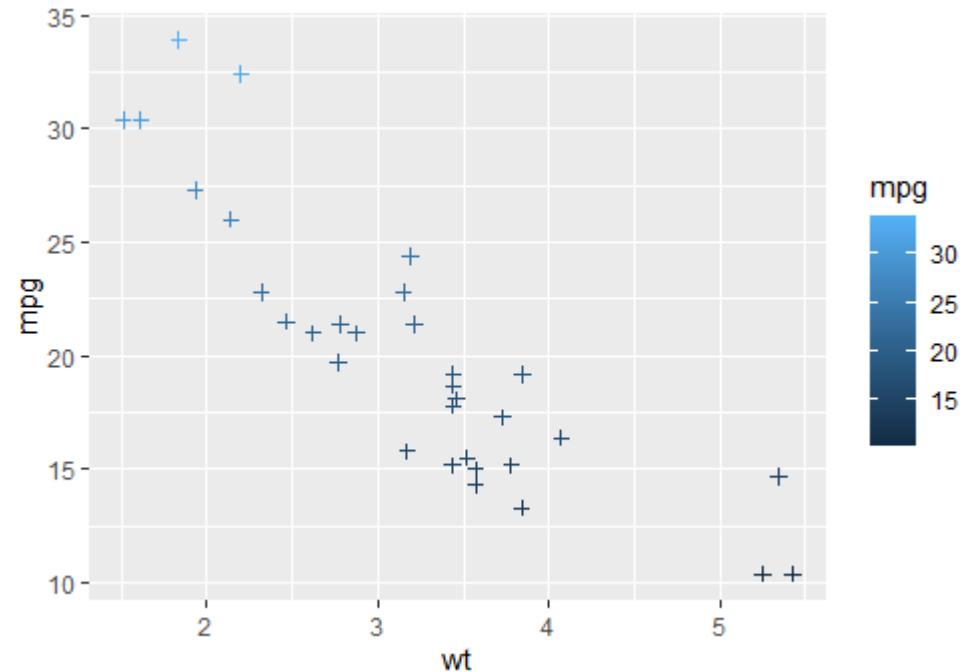
```
g1 = ggplot(mtcars)
g2 = g1 + aes(x=wt, y=mpg, color=mpg)
g3 = g2 + geom_point()
g3
```



# Ggplot2: code

- ggplot has pretty good defaults, so you don't have to specify every single attribute of a geom.
- If you want to change an attribute manually (instead of being dependent on your data), you can do it right in the geom's function.

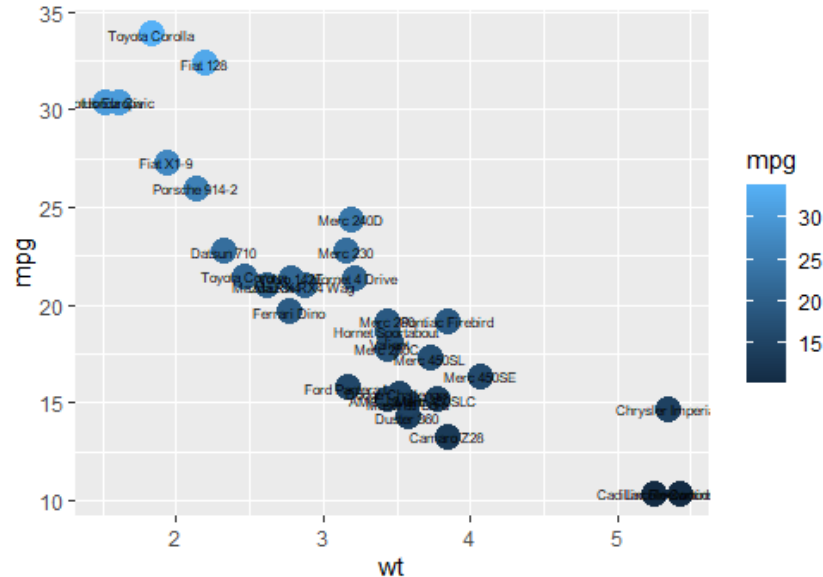
```
g1 = ggplot(mtcars)
g2 = g1 + aes(x=wt, y=mpg, color=mpg)
g3 = g2 + geom_point(shape=3)
g3
```



# ggplot2: code

- You can combine multiple geom's in the same plot.
- Be sure to update the aesthetics as you build your plot

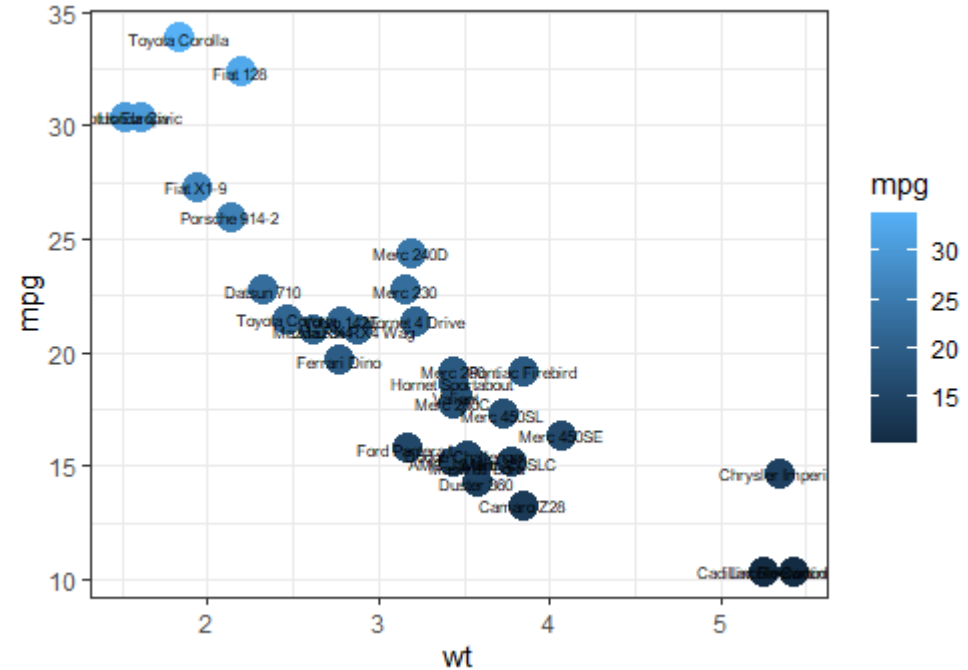
```
g1 = ggplot(mtcars)
g2 = g1 + aes(x=wt, y=mpg, color=mpg, label=carname)
g3 = g2 + geom_point(shape=16, size=5)
g4 = g3 + geom_text(color="black", size=2)
g4
```



# Ggplot: code

- You can change the look of the plot with a “theme”

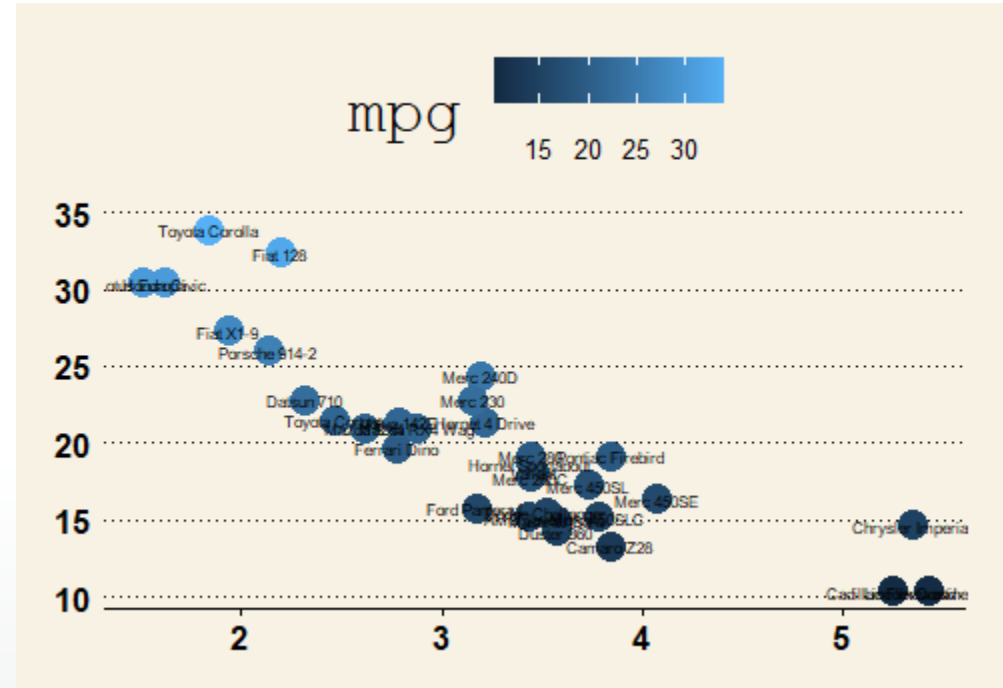
`g4 + theme_bw()`



# Ggplot: code

- You can change the look of the plot with a “theme”

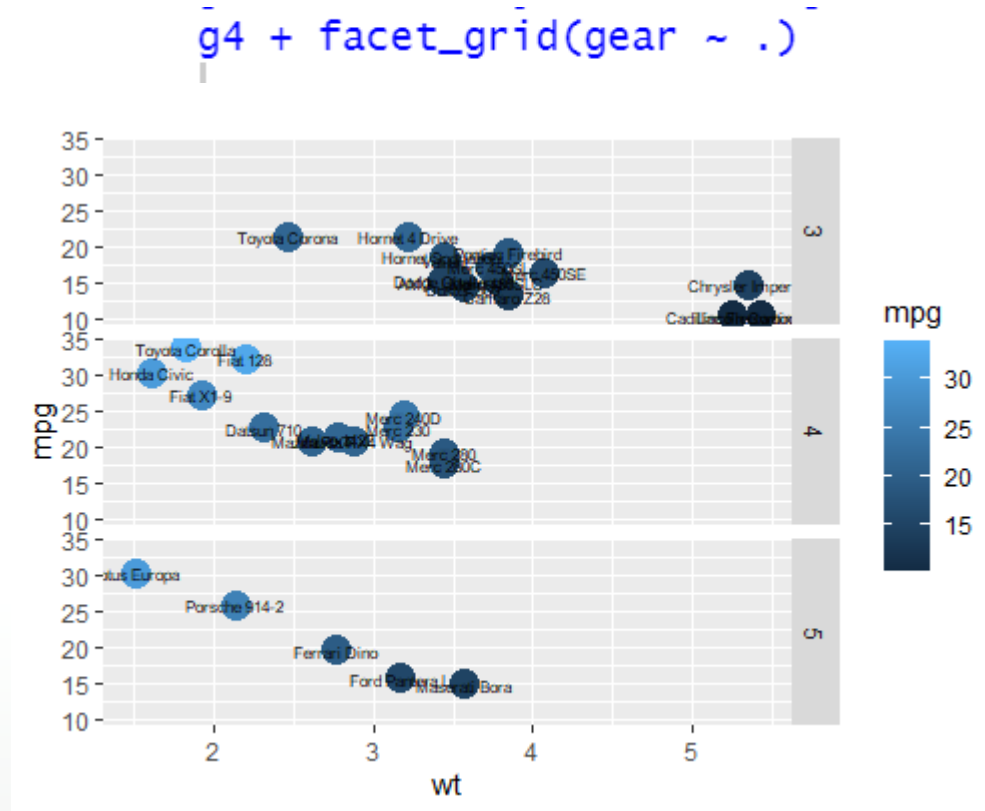
```
g4 + theme_wsj()
```



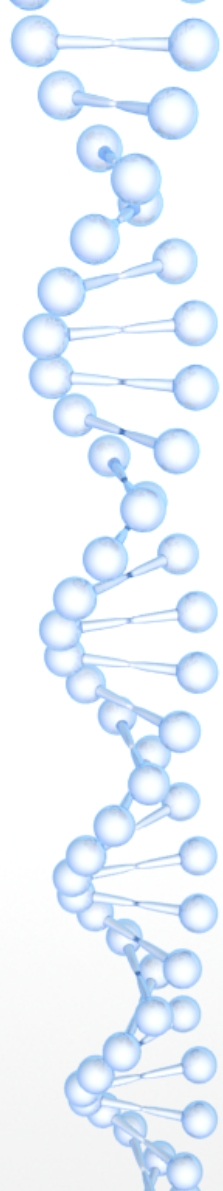


# Ggplot: code

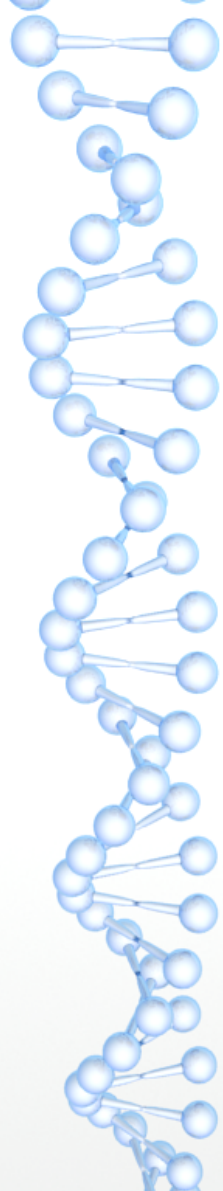
- You can break the plot up into parts based on a grouping variable with “facet\_grid”
- The variable before the ‘~’ will separate the graph by rows



- 

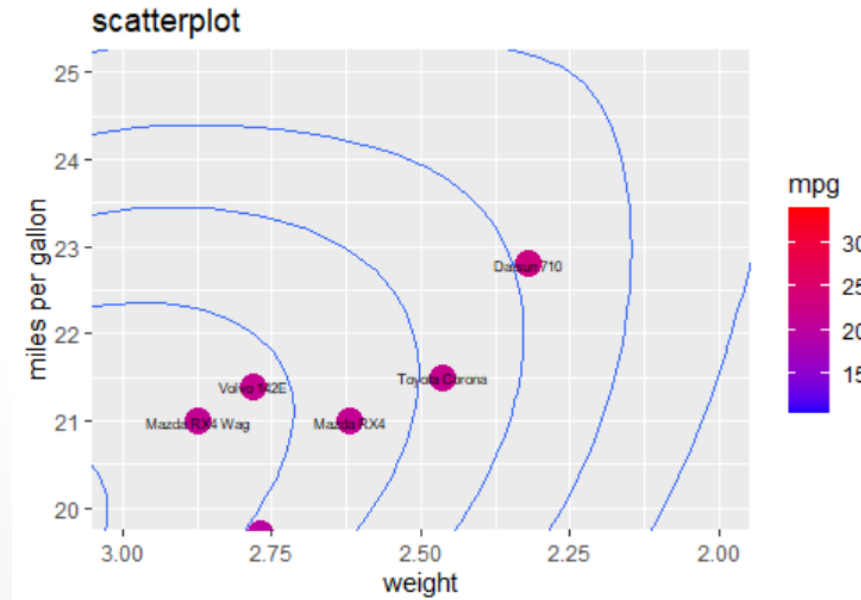


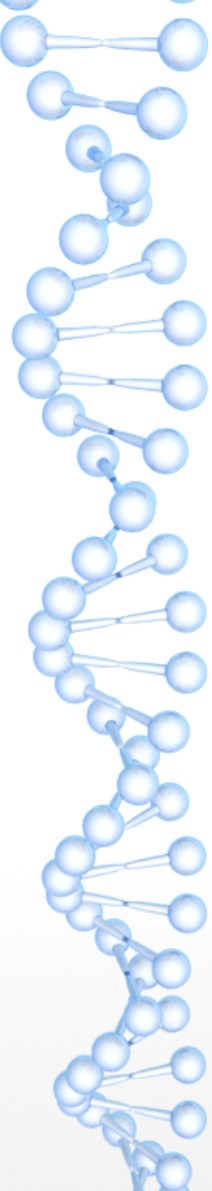
- 



# ggplot2: demo

```
ggplot(mtcars, aes(x=wt, y=mpg, color=mpg, label=carname)) +  
  geom_point(shape=16, size=5) +  
  geom_text(color="black", size=2) +  
  stat_density2d() +  
  scale_x_reverse() +  
  scale_color_gradient(low="blue", high="red") +  
  xlab("weight") + ylab("miles per gallon") + ggtitle("scatterplot") +  
  coord_cartesian( xlim = c(3,2), ylim = c(20, 25)) +  
  theme()
```

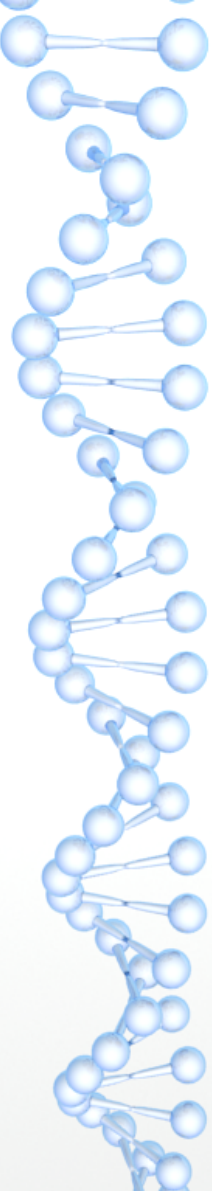




## ggplot2: gotcha

- You can write your R commands in a long single line but it's easier to read when it's broken up to span multiple lines.
- If a line is a valid R statement, R will process it as such, so be sure R knows you haven't finished your line
  - End the line with a +
  - Keep an open parenthesis

1	1 + 2 + 3 + 4 + 5	<-- Good
2		
3	1 + 2	
4	+ 3 + 4 + 5	<--Bad
5		
6	1 + 2 +	
7	3 + 4 + 5	<-- Good
8		
9	(1 + 2	
10	+ 3 + 4 + 5)	<-- Good
11		



# Ggplot2: resources

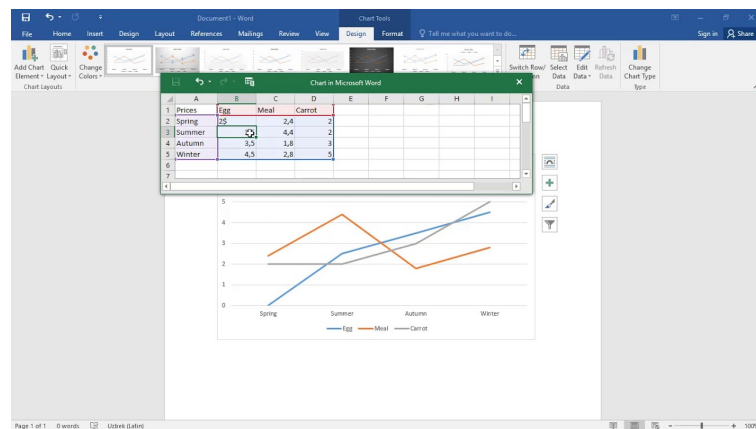
- <https://www.datanovia.com/en/blog/ggplot-examples-best-reference/>
- <https://rstudio.com/wp-content/uploads/2015/03/ggplot2-cheatsheet.pdf>
- <https://www.kaggle.com/raenish/cheatsheet-70-ggplot-charts>
- Google
  - R ggplot boxplot
  - Then click on the images tab!



# knitr

# knitr: reproducible research

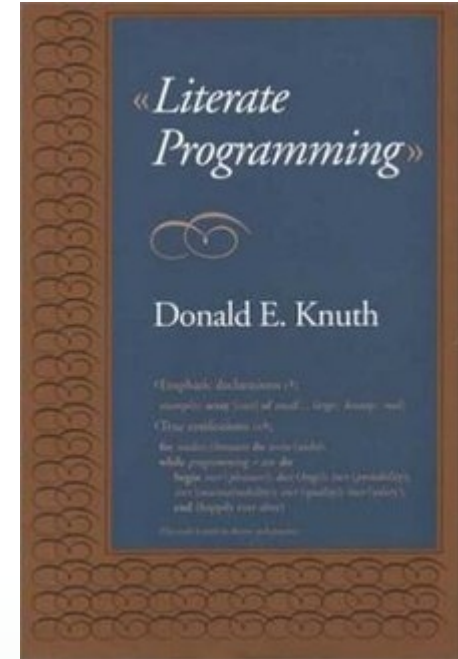
- How do you report the results from an analysis?
  - Open up microsoft Word.
  - Start writing about what you did
  - copy/paste results from the R console to the word doc
  - Add your plots to the document
  - Email your collaborators
- It's never this simple in the real world
  - Data from new patients just came in, need to rerun.
  - Maybe the plot will look better in blue
  - Run a wilcoxon test instead of t-test
  - Mistakes happen
- It gets difficult to keep track after just a few rounds of changes. It's even harder with multiple collaborators.
  - The code changes
  - The data changes
  - The results change





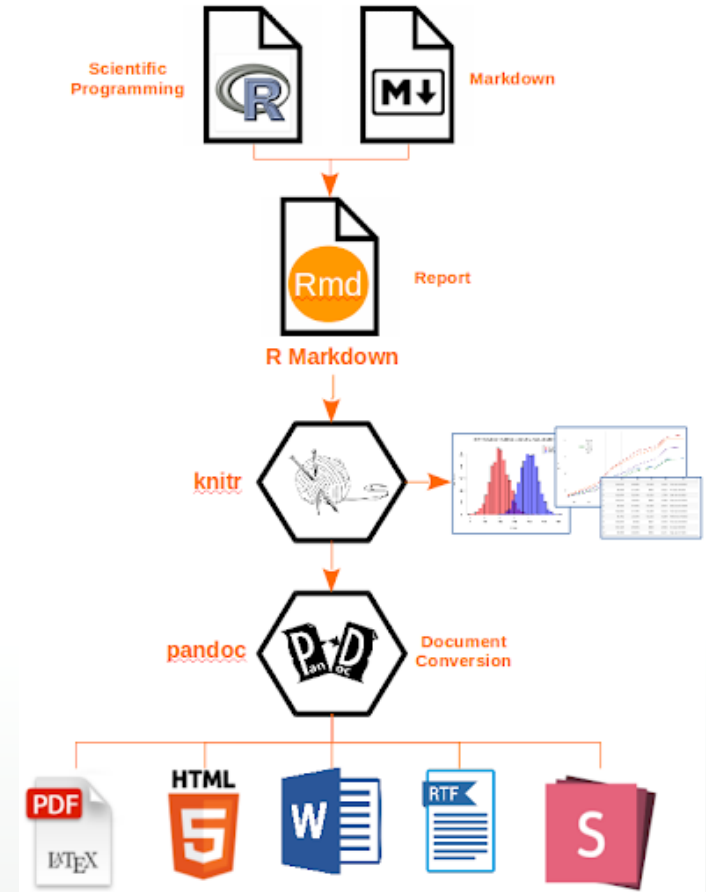
# knitr: literate programming

- We need a way to combine the reporting of results, with the code and data that was used to generate those results.
  - A single document that has it all
  - A way to see what data was used, all the processing steps involved, and where the results came from
  - Needs to be easy to use



# knitr and Rstudio

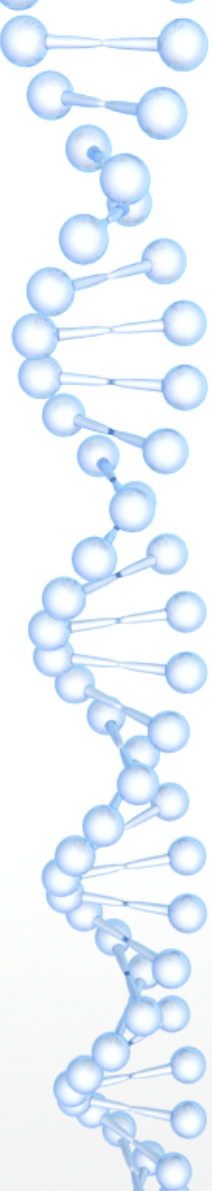
- Rstudio has a solution.
- You write a single “.Rmd” file that contains your
  - R code
  - Results and plots
  - Write up
  - Data
- With the click of a button a report is generated for you automatically



# knitr: a simple .Rmd file

```
1 ---
2 title: "Untitled"
3 output: html_document
4 ---
5
6 ```{r setup, include=FALSE}
7 knitr::opts_chunk$set(echo = TRUE)
8 ```
9
10 ## R Markdown
11
12 This is an R Markdown document. Markdown is a simple formatting syntax for authoring
13 HTML, PDF, and MS Word documents. For more details on using R Markdown see
14 <http://rmarkdown.rstudio.com>.
15
16 When you click the Knit button a document will be generated that includes both
17 content as well as the output of any embedded R code chunks within the document. You
18 can embed an R code chunk like this:
19
20 ```{r cars}
21 summary(cars)
22 ```
23
24 ## Including Plots
25
26 You can also embed plots, for example:
27
28 ```{r pressure, echo=FALSE}
29 plot(pressure)
30 ```
31
32 Note that the `echo = FALSE` parameter was added to the code chunk to prevent
33 printing of the R code that generated the plot.
```

# knitr: a simple .Rmd file



```
1 ---
2 title: "Untitled"
3 output: html_document
4 ---
5
6 ```{r setup, include=FALSE}
7 knitr::opts_chunk$set(echo = TRUE)
8 ```
9
10 ## R Markdown
11
12 This is an R Markdown document. Markdown is a simple formatting syntax for authoring
13 HTML, PDF, and MS Word documents. For more details on using R Markdown see
14 <http://rmarkdown.rstudio.com>.
15
16 When you click the Knit button a document will be generated that includes both
17 content as well as the output of any embedded R code chunks within the document. You
18 can embed an R code chunk like this:
19
20 ```{r cars}
21 summary(cars)
22 ```
23
24 ## Including Plots
25
26 You can also embed plots, for example:
27
28 ```{r pressure, echo=FALSE}
29 plot(pressure)
30 ```
31
32 Note that the `echo = FALSE` parameter was added to the code chunk to prevent
33 printing of the R code that generated the plot.
```

- The header lies between two --- sequences
  - title: “Untitled”
    - Be sure to keep the quotations “stuff”
  - output: html\_document
    - There are many output formats available.
      - pdf\_document
      - slidy\_presentation
      - powerpoint\_presentation
      - <https://rmarkdown.rstudio.com/lesson-9.html>
- Each output format has it's own parameters

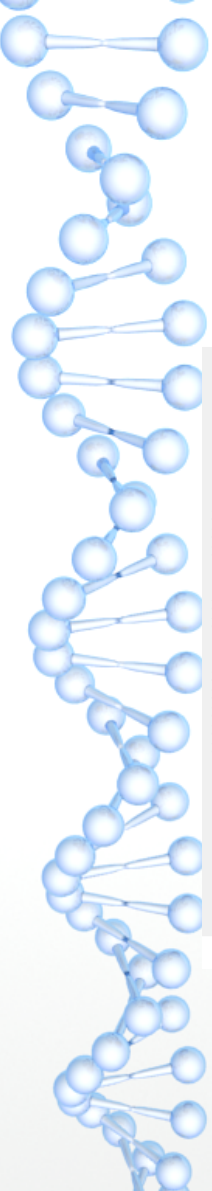
```
---
title: "test3"
output:
  html_document:
    toc: true
    toc_float: true
---
```

# knitr: a simple .Rmd file

```
1 ---
2 title: "Untitled"
3 output: html_document
4 ---
5
6 ```{r setup, include=FALSE}
7 knitr::opts_chunk$set(echo = TRUE)
8 ```
9
10 ## R Markdown
11
12 This is an R Markdown document. Markdown is a simple formatting syntax for authoring
13 HTML, PDF, and MS Word documents. For more details on using R Markdown see
14 <http://rmarkdown.rstudio.com>.
15
16 When you click the Knit button a document will be generated that includes both
17 content as well as the output of any embedded R code chunks within the document. You
18 can embed an R code chunk like this:
19
20 ```{r cars}
21 summary(cars)
22 ```
23
24 ## Including Plots
25
26 You can also embed plots, for example:
27
28 ```{r pressure, echo=FALSE}
29 plot(pressure)
30 ```
31
32 Note that the `echo = FALSE` parameter was added to the code chunk to prevent
33 printing of the R code that generated the plot.
```

- R code is placed between two ``` sequences
  - You specify the language of the code chunk in brackets
  - You can name the code chunk, or leave it blank
  - options for displaying code are specified after a comma
    - include = FALSE prevents code and results from appearing in the finished file. R Markdown still runs the code in the chunk, and the results can be used by other chunks.
    - echo = FALSE prevents code, but not the results from appearing in the finished file. This is a useful way to embed figures.
    - message = FALSE prevents messages that are generated by code from appearing in the finished file.
    - warning = FALSE prevents warnings that are generated by code from appearing in the finished.
    - fig.cap = "..." adds a caption to graphical results.

# knitr: a simple .Rmd file



```
1 ---
2 title: "Untitled"
3 output: html_document
4 ---
5
6 {r setup, include=FALSE}
7 knitr::opts_chunk$set(echo = TRUE)
8
9
10 ## R Markdown
11
12 This is an R Markdown document. Markdown is a simple formatting syntax for authoring
13 HTML, PDF, and MS Word documents. For more details on using R Markdown see
14 <http://rmarkdown.rstudio.com>.
15
16 When you click the Knit button a document will be generated that includes both
17 content as well as the output of any embedded R code chunks within the document. You
18 can embed an R code chunk like this:
19
20 {r cars}
21 summary(cars)
22
23 ## Including Plots
24
25 You can also embed plots, for example:
26
27 {r pressure, echo=FALSE}
28 plot(pressure)
29
30 Note that the `echo = FALSE` parameter was added to the code chunk to prevent
31 printing of the R code that generated the plot.
```

- Everything else is Rmarkdown
  - A line starting with `##` appears as a header
  - Txt between two sets of `*` appear as italic
  - Txt between two sets of `**` appears as bold
  - You can add inline code here without having to specify a code chunk
    - The sum of 3 and 4 is ``r 3+4``
  - There's so much you can do
    - <https://rmarkdown.rstudio.com/lesson-8.html>

# knitr: a simple .Rmd file

```
1 ---
2 title: "Untitled"
3 output: html_document
4 ---
5
6 {r setup, include=FALSE}
7 knitr::opts_chunk$set(echo = TRUE)
8
9
10 ## R Markdown
11
12 This is an R Markdown document. Markdown is a simple formatting syntax for authoring
13 HTML, PDF, and MS Word documents. For more details on using R Markdown see
14 <http://rmarkdown.rstudio.com>.
15
16 When you click the Knit button a document will be generated that includes both
17 content as well as the output of any embedded R code chunks within the document. You
18 can embed an R code chunk like this:
19
20 {r cars}
21 summary(cars)
22
23
24 ## Including Plots
25
26 You can also embed plots, for example:
27
28 {r pressure, echo=FALSE}
29 plot(pressure)
30
31 Note that the `echo = FALSE` parameter was added to the code chunk to prevent
32 printing of the R code that generated the plot.
```

## Untitled

### R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

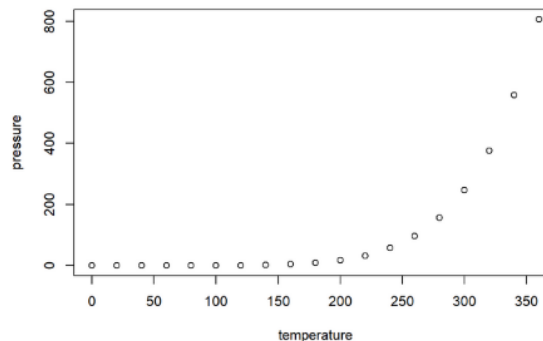
When you click the Knit button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed      dist
##  Min.   : 4.0   Min.   : 2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  2nd Qu.:19.0   2nd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

### Including Plots

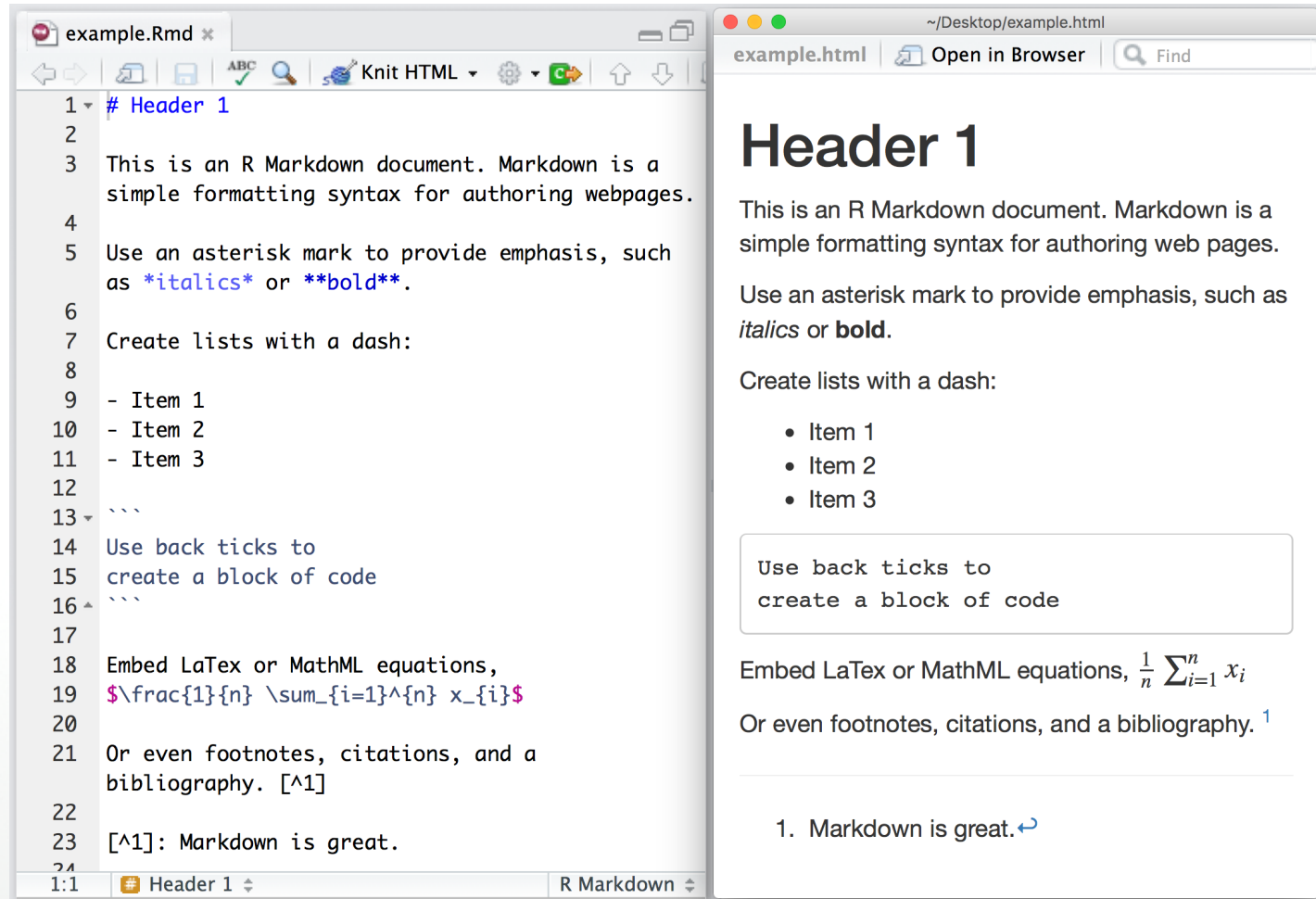
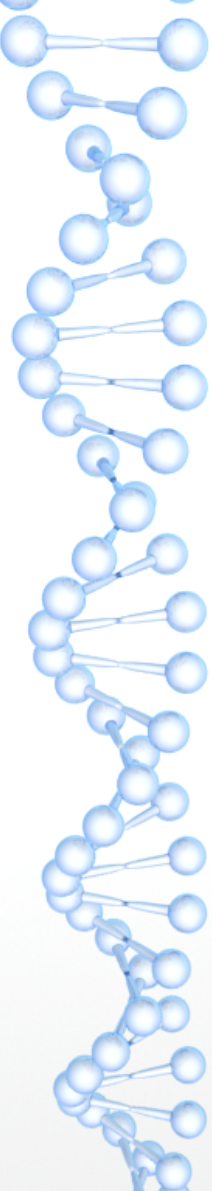
You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.



# knitr: Rmarkdown



The image displays a side-by-side comparison of an R Markdown source file and its rendered HTML output.

**Left Panel (Source File: example.Rmd):**

```
1 # Header 1
2
3 This is an R Markdown document. Markdown is a
  simple formatting syntax for authoring webpages.
4
5 Use an asterisk mark to provide emphasis, such
  as italics or bold.
6
7 Create lists with a dash:
8
9 - Item 1
10 - Item 2
11 - Item 3
12
13 ```
14 Use back ticks to
15 create a block of code
16 ```
17
18 Embed LaTeX or MathML equations,
19 
$$\frac{1}{n} \sum_{i=1}^n x_i$$

20
21 Or even footnotes, citations, and a
  bibliography. [^1]
22
23 [^1]: Markdown is great.
24
25 1:1
```

**Right Panel (Rendered HTML: example.html):**

## Header 1

This is an R Markdown document. Markdown is a simple formatting syntax for authoring web pages.

Use an asterisk mark to provide emphasis, such as *italics* or **bold**.

Create lists with a dash:

- Item 1
- Item 2
- Item 3

Use back ticks to create a block of code

Embed LaTeX or MathML equations, 
$$\frac{1}{n} \sum_{i=1}^n x_i$$

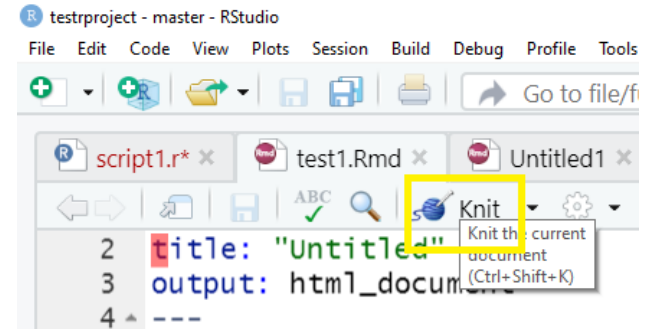
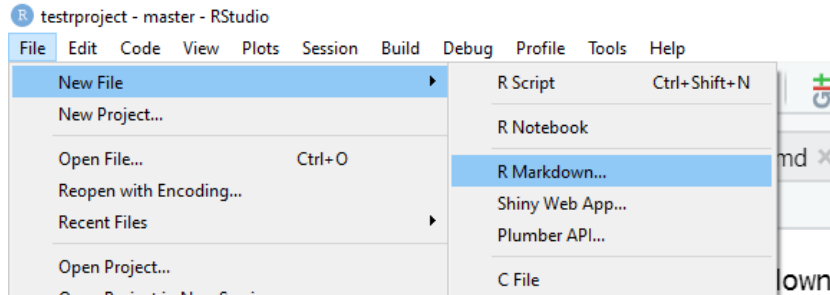
Or even footnotes, citations, and a bibliography. <sup>1</sup>

---

1. Markdown is great.↩



# knitr: Demo



- File → New File → R Markdown
  - Choose title and output format. You might have to install extras if you want to save as pdf or Word document.
- It will produce a very simple .Rmd file. A good starting point where you do your own analysis.
- Click the “Knit” button (looks like a ball of yarn)
  - Choose where you want to save the output report.



# knitr: references

- Some tutorials on RMarkdown
  - <https://ourcodingclub.github.io/tutorials/rmarkdown/>
  - <https://bookdown.org/yihui/rmarkdown/markdown-syntax.html>
  - <https://rmarkdown.rstudio.com/lesson-1.html>
- Rmarkdown cheatsheet
  - [https://rstudio.com/wp-content/uploads/2016/03/rmarkdown-cheatsheet-2.0.pdf?\\_ga=2.128839753.473261408.1598550734-1967726035.1567620807](https://rstudio.com/wp-content/uploads/2016/03/rmarkdown-cheatsheet-2.0.pdf?_ga=2.128839753.473261408.1598550734-1967726035.1567620807)