

# Data Visualisation with R

Richard White

# Topics

- Data manipulation
  - tidy data and the pipe (%>%) operator
  - loading data
  - selecting, filtering and creating new columns
- ggplot2
  - aesthetics
  - geoms
  - shapes and colours
  - facets
  - themes

# Tidyverse

Tidyverse

Packages    Blog    Learn    Help    Contribute

The tidyverse is an opinionated **collection of R packages** designed for data science. All packages share an underlying design philosophy, grammar, and data structures.

Install the complete tidyverse with:

```
install.packages("tidyverse")
```

<https://www.tidyverse.org/>

# Tidy data

- Each variable forms a column
- Each observation forms a row
- Each type of observational unit forms a table

name	description	sample_1	sample_2	sample_3
slc35a5	ENSDARG000000000001	36.28229146	33.4323	34.1081598
ccdc80	ENSDARG000000000002	75.50639034	113.155	102.324479
nrf1	ENSDARG000000000018	305.9479712	281.174	274.760176

# Tidy data

- Each variable forms a column
- Each observation forms a row
- Each type of observational unit forms a table

name	description	sample	normalised_count
slc35a5	ENSDARG000000000001	sample_1	36.2823
ccdc80	ENSDARG000000000002	sample_1	75.5064
nrf1	ENSDARG000000000018	sample_1	305.948
slc35a5	ENSDARG000000000001	sample_2	33.4323
ccdc80	ENSDARG000000000002	sample_2	113.155
nrf1	ENSDARG000000000018	sample_2	281.174
slc35a5	ENSDARG000000000001	sample_3	34.1082
ccdc80	ENSDARG000000000002	sample_3	102.324
nrf1	ENSDARG000000000018	sample_3	274.76

# Tidy data

- Each variable forms a column
- Each observation forms a row
- Each type of observational unit forms a table

name	description	sample	count	normalised_count
slc35a5	ENSDARG000000000001	sample_1	35	36.2823
ccdc80	ENSDARG000000000002	sample_1	75	75.5064
nrf1	ENSDARG000000000018	sample_1	300	305.948
slc35a5	ENSDARG000000000001	sample_2	30	33.4323
ccdc80	ENSDARG000000000002	sample_2	115	113.155
nrf1	ENSDARG000000000018	sample_2	283	281.174
slc35a5	ENSDARG000000000001	sample_3	31	34.1082
ccdc80	ENSDARG000000000002	sample_3	107	102.324
nrf1	ENSDARG000000000018	sample_3	276	274.76

# pipe

- `%>%`
- Equivalent of the Unix `|`
  - `cut -f1,2,4 data.txt | head`
  - `select(data, c(1,2,4)) %>% head()`
- Allows sending the results of one function into the next
- Makes code easier to read
  - `head(select(data, c(1,2,4)), 6)`
  - `select(data, c(1,2,4)) %>% head(6)`

%>%

```
eat(                                ingredients %>%
  slice(                                mix() %>%
  bake(                                pour(into=baking_form) %>%
  put(                                put(into=oven) %>%
  pour(                                bake(time=30) %>%
    mix(ingredients),                slice(pieces=6) %>%
    into=baking_form),              eat(1)
    into=oven),
    time=30),
    pieces=6),
1)
```

# readr

- `read_tsv`
  - reads in tab-delimited data and tries to guess the type of each column
    - character, integer etc.
- `read_csv`
  - same for comma-separated files



# read\_tsv

```
> read_tsv(file.path('rvis', 'Rvis_test_data.tsv'))
Parsed with column specification:
cols(
  A = col_character(),
  B = col_character(),
  C = col_double(),
  D = col_double(),
  E = col_integer(),
  F = col_logical()
)
# A tibble: 50 × 6
   A     B     C     D     E F
   <chr> <chr> <dbl> <dbl> <int> <lgl>
 1 b     E     4.80  29.9   58 FALSE
 2 b     D     0.819 29.5   69 FALSE
 3 d     E     2.47  36.3   21 TRUE 
 4 b     D     8.71  47.1   62 TRUE 
 5 d     D     8.95  28.7   82 FALSE
 6 d     C     3.64  14.3   46 TRUE 
 7 a     C     0.532 35.5   80 FALSE
 8 a     D     6.06  13.4   69 FALSE
 9 a     C     5.60  16.1   35 FALSE
10 c    E     4.87  44.4   26 FALSE
# ... with 40 more rows
```

# factors

```
> samples <- read_tsv('rvis/Rvis_test_samples.tsv')
Parsed with column specification:
cols(
  sample_name = col_character(),
  genotype = col_character()
)
> head(samples)
# A tibble: 6 × 2
  sample_name genotype
  <chr>       <chr>
1 sample_1     wt
2 sample_2     wt
3 sample_3     wt
4 sample_4     wt
5 sample_5     het
6 sample_6     het
> samples$genotype
[1] "wt"  "wt"  "wt"  "wt"  "het" "het" "het" "het" "hom" "hom" "hom"
[12] "hom"
> factor(samples$genotype)
[1] wt  wt  wt  wt  het  het  het  het  hom  hom  hom  hom
Levels: het hom wt
> factor(samples$genotype, levels = c('wt', 'het', 'hom'))
[1] wt  wt  wt  wt  het  het  het  het  hom  hom  hom  hom
Levels: wt  het  hom
> samples$genotype <- factor(samples$genotype, levels = c('wt', 'het', 'hom'))
```

# wide vs long data

id	x	y	z
1	a	c	e
2	b	d	f

id name val

1	x	a
1	y	c
1	z	e
2	x	b
2	y	d
2	z	f

# `pivot_wider()` and `pivot_longer()`

wide

id	x	y	z
1	a	c	e
2	b	d	f

Garrick Aden-Buie and Mara Averick

<https://github.com/batpigandme/tidyexplain/blob/pivot/images/tidyr-longer-wider.gif>

# Tidy data

name	description	sample_1	sample_2	sample_3
slc35a5	ENSDARG000000000001	36.28229146	33.4323	34.1081598
ccdc80	ENSDARG000000000002	75.50639034	113.155	102.324479
nrf1	ENSDARG000000000018	305.9479712	281.174	274.760176

name	description	sample	normalised_count
slc35a5	ENSDARG000000000001	sample_1	36.2823
ccdc80	ENSDARG000000000002	sample_1	75.5064
nrf1	ENSDARG000000000018	sample_1	305.948
slc35a5	ENSDARG000000000001	sample_2	33.4323
ccdc80	ENSDARG000000000002	sample_2	113.155
nrf1	ENSDARG000000000018	sample_2	281.174
slc35a5	ENSDARG000000000001	sample_3	34.1082
ccdc80	ENSDARG000000000002	sample_3	102.324
nrf1	ENSDARG000000000018	sample_3	274.76

# Data manipulation

- `select()`: pick variables
- `filter()`: pick rows
- `mutate()`: add new variables that are a function of existing ones
- `arrange()`: sort rows
- `summarise()`: reduce multiple values down to a single summary (mean, min, max, etc.)



# select()

- Choose variables from a table
  - use column names explicitly: `select(data, GeneID)`
  - or positions: `select(data, c(1,5,9))`
  - column names can be used like they are positions  
e.g. `select(data, GeneID:Name)`
  - or search functions
    - `starts_with()` `select(data, starts_with('uninf'))`
    - `ends_with()` `select(data, ends_with('count'))`
    - `contains()` `select(data, contains('3dpf'))`
    - `matches()` `select(data, matches('3dpf.*count'))`

# select()

```
> head(iris, 4)
  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
> summary(iris)
  Sepal.Length     Sepal.Width     Petal.Length     Petal.Width      Species
  Min.    :4.300   Min.    :2.000   Min.    :1.000   Min.    :0.100   setosa    :50
  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300   versicolor:50
  Median  :5.800   Median  :3.000   Median  :4.350   Median  :1.300   virginica :50
  Mean    :5.843   Mean    :3.057   Mean    :3.758   Mean    :1.199
  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
  Max.    :7.900   Max.    :4.400   Max.    :6.900   Max.    :2.500
> select(iris, Sepal.Length, Sepal.Width) %>% head(4)
  Sepal.Length Sepal.Width
1          5.1        3.5
2          4.9        3.0
3          4.7        3.2
4          4.6        3.1
> select(iris, starts_with('Petal')) %>% head(4)
  Petal.Length Petal.Width
1          1.4        0.2
2          1.4        0.2
3          1.3        0.2
4          1.5        0.2
```

# select()

```
> select(iris, ends_with('Width')) %>% head(4)
  Sepal.Width Petal.Width
1          3.5        0.2
2          3.0        0.2
3          3.2        0.2
4          3.1        0.2
> select(iris, -Species) %>% head(4)
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1          5.1        3.5        1.4        0.2
2          4.9        3.0        1.4        0.2
3          4.7        3.2        1.3        0.2
4          4.6        3.1        1.5        0.2
> vars <- c("Petal.Length", "Petal.Width")
> select(iris, one_of(vars)) %>% head(4)
  Petal.Length Petal.Width
1          1.4        0.2
2          1.4        0.2
3          1.3        0.2
4          1.5        0.2
```

# filter()

- choose rows where conditions are true
  - check equality with ==
  - Also <, >, <=, >=
  - combine conditions with & (AND), | (OR)
  - ! (NOT)
  - e.g.  
`adjustedp < 0.05 & log2fc > 2`

# filter()

```
> filter(iris, Species == "virginica") %>% dim()
[1] 50  5
> filter(iris, Species == "virginica") %>% head(4)
  Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
1          6.3       3.3        6.0       2.5 virginica
2          5.8       2.7        5.1       1.9 virginica
3          7.1       3.0        5.9       2.1 virginica
4          6.3       2.9        5.6       1.8 virginica
> filter(iris, Species == "virginica" | Species == "versicolor") %>% dim()
[1] 100  5
> filter(iris, Sepal.Length > 6) %>% head(4)
  Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
1          7.0       3.2        4.7       1.4 versicolor
2          6.4       3.2        4.5       1.5 versicolor
3          6.9       3.1        4.9       1.5 versicolor
4          6.5       2.8        4.6       1.5 versicolor
> filter(iris, Sepal.Length > 6) %>% dim()
[1] 61  5
> filter(iris, Sepal.Length > 6 & Sepal.Width < 3) %>% head(4)
  Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
1          6.5       2.8        4.6       1.5 versicolor
2          6.6       2.9        4.6       1.3 versicolor
3          6.1       2.9        4.7       1.4 versicolor
4          6.2       2.2        4.5       1.5 versicolor
> filter(iris, Sepal.Length > 6 & Sepal.Width < 3) %>% dim()
[1] 25  5
```

# mutate()

```
> mutate(iris, exp = 10^Petal.Width) %>% head(4)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species      exp
1          5.1        3.5         1.4        0.2  setosa 1.584893
2          4.9        3.0         1.4        0.2  setosa 1.584893
3          4.7        3.2         1.3        0.2  setosa 1.584893
4          4.6        3.1         1.5        0.2  setosa 1.584893
> mutate(iris, l2fc = log2(Sepal.Length)) %>% head(4)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species      l2fc
1          5.1        3.5         1.4        0.2  setosa 2.350497
2          4.9        3.0         1.4        0.2  setosa 2.292782
3          4.7        3.2         1.3        0.2  setosa 2.232661
4          4.6        3.1         1.5        0.2  setosa 2.201634
> mutate(iris, cumsum = cumsum(Petal.Length)) %>% head(4)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species     cumsum
1          5.1        3.5         1.4        0.2  setosa       1.4
2          4.9        3.0         1.4        0.2  setosa       2.8
3          4.7        3.2         1.3        0.2  setosa       4.1
4          4.6        3.1         1.5        0.2  setosa       5.6
> mutate(iris, tenfold = Petal.Length * 10, tenth = tenfold / 100) %>% head(4)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species    tenfold   tenth
1          5.1        3.5         1.4        0.2  setosa       14  0.14
2          4.9        3.0         1.4        0.2  setosa       14  0.14
3          4.7        3.2         1.3        0.2  setosa       13  0.13
4          4.6        3.1         1.5        0.2  setosa       15  0.15
```

# arrange()

```
> arrange(iris, Sepal.Length) %>% head(4)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          4.3       3.0        1.1       0.1   setosa
2          4.4       2.9        1.4       0.2   setosa
3          4.4       3.0        1.3       0.2   setosa
4          4.4       3.2        1.3       0.2   setosa

> arrange(iris, desc(Sepal.Length)) %>% head(4)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          7.9       3.8        6.4       2.0 virginica
2          7.7       3.8        6.7       2.2 virginica
3          7.7       2.6        6.9       2.3 virginica
4          7.7       2.8        6.7       2.0 virginica

> arrange(iris, Sepal.Length, Petal.Length) %>% head(4)
  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          4.3       3.0        1.1       0.1   setosa
2          4.4       3.0        1.3       0.2   setosa
3          4.4       3.2        1.3       0.2   setosa
4          4.4       2.9        1.4       0.2   setosa
```

# Exercises

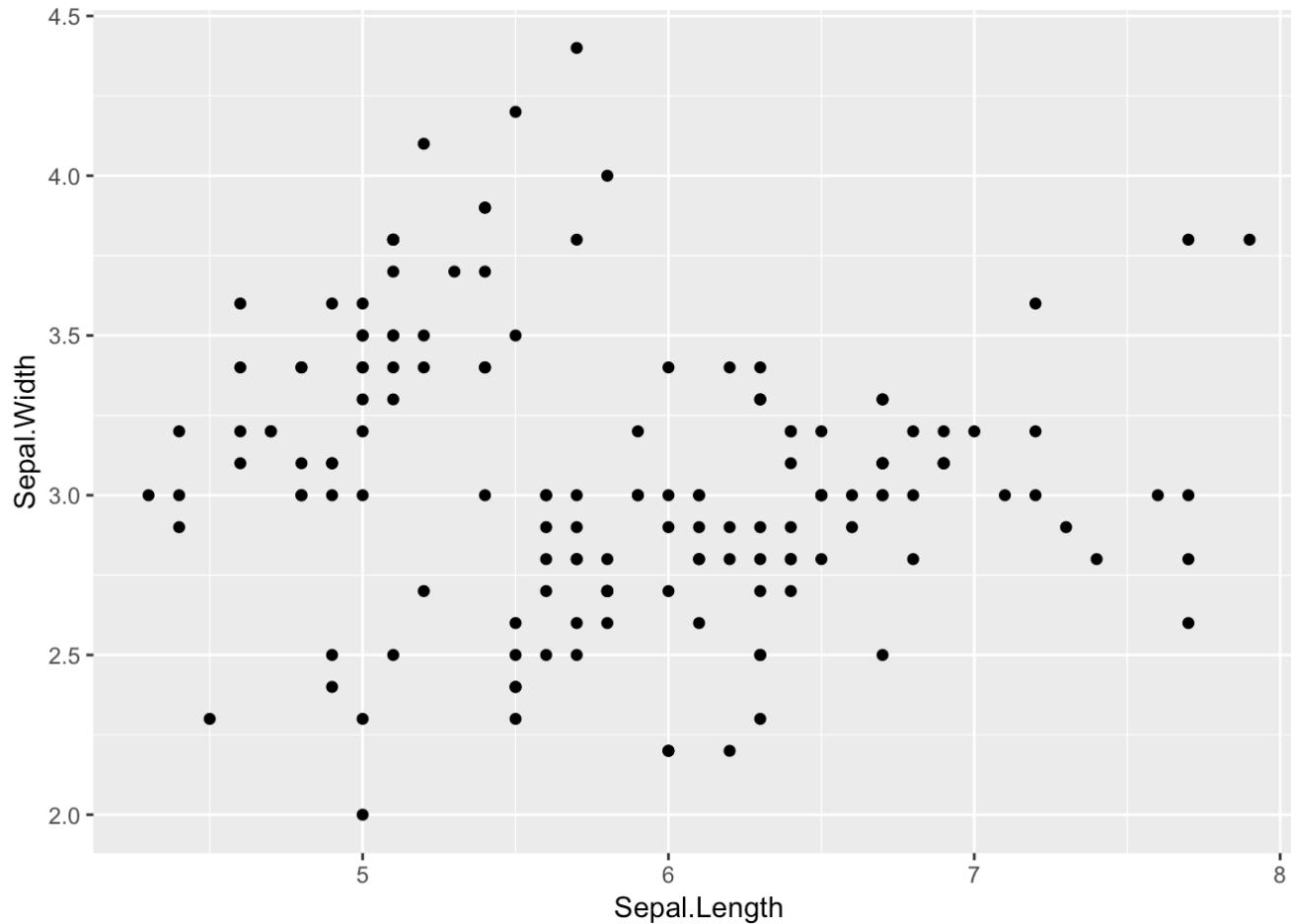
# ggplot2

- Grammar of Graphics
  - Leland Wilkinson (2005)
- Components of a plot
  1. data
  2. aesthetics
    - attributes of a plot that variables in the data are mapped to
    - x, y, colour, shape, length, size, linetype
  3. theme



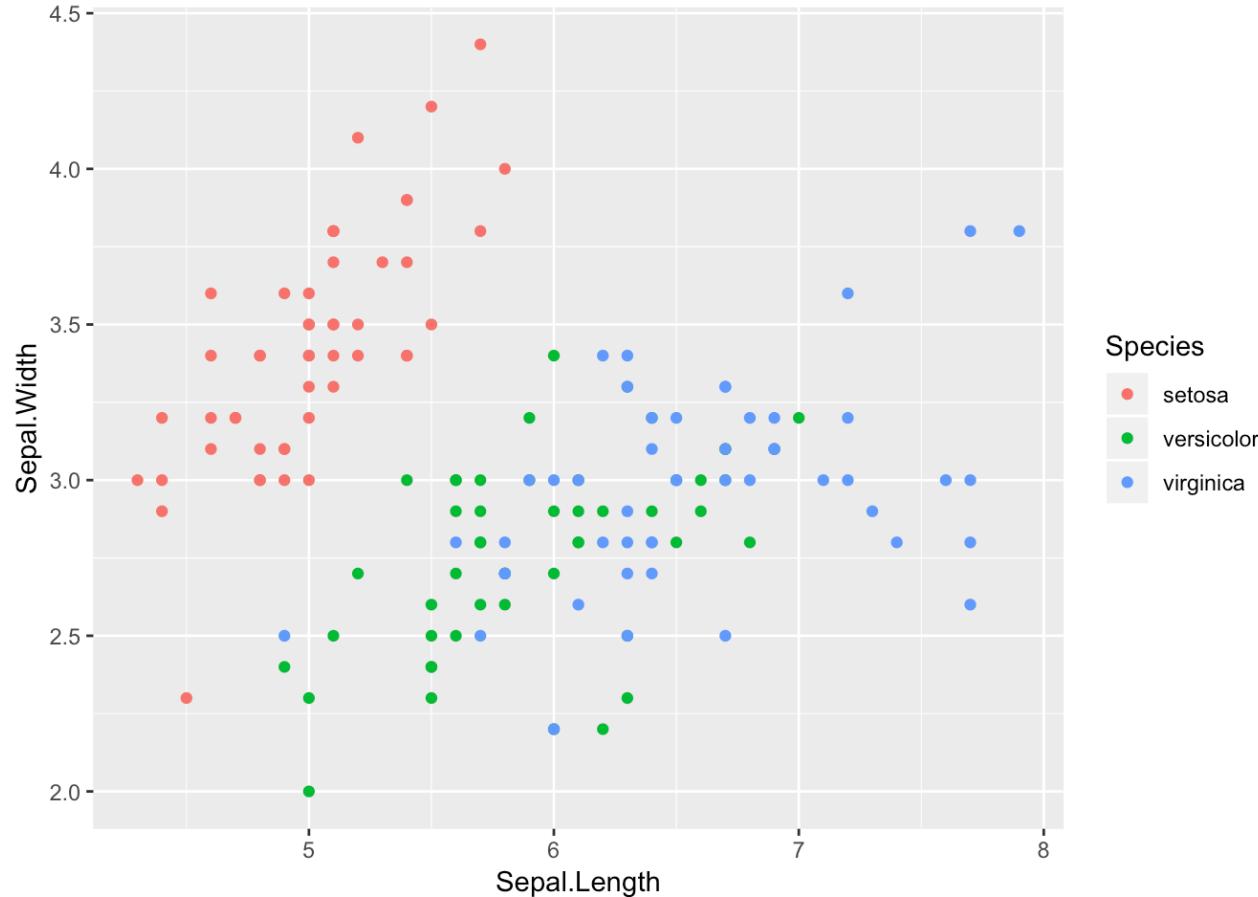
# scatterplot

```
ggplot(data = iris) +  
  geom_point( aes(x = Sepal.Length, y = Sepal.Width))
```



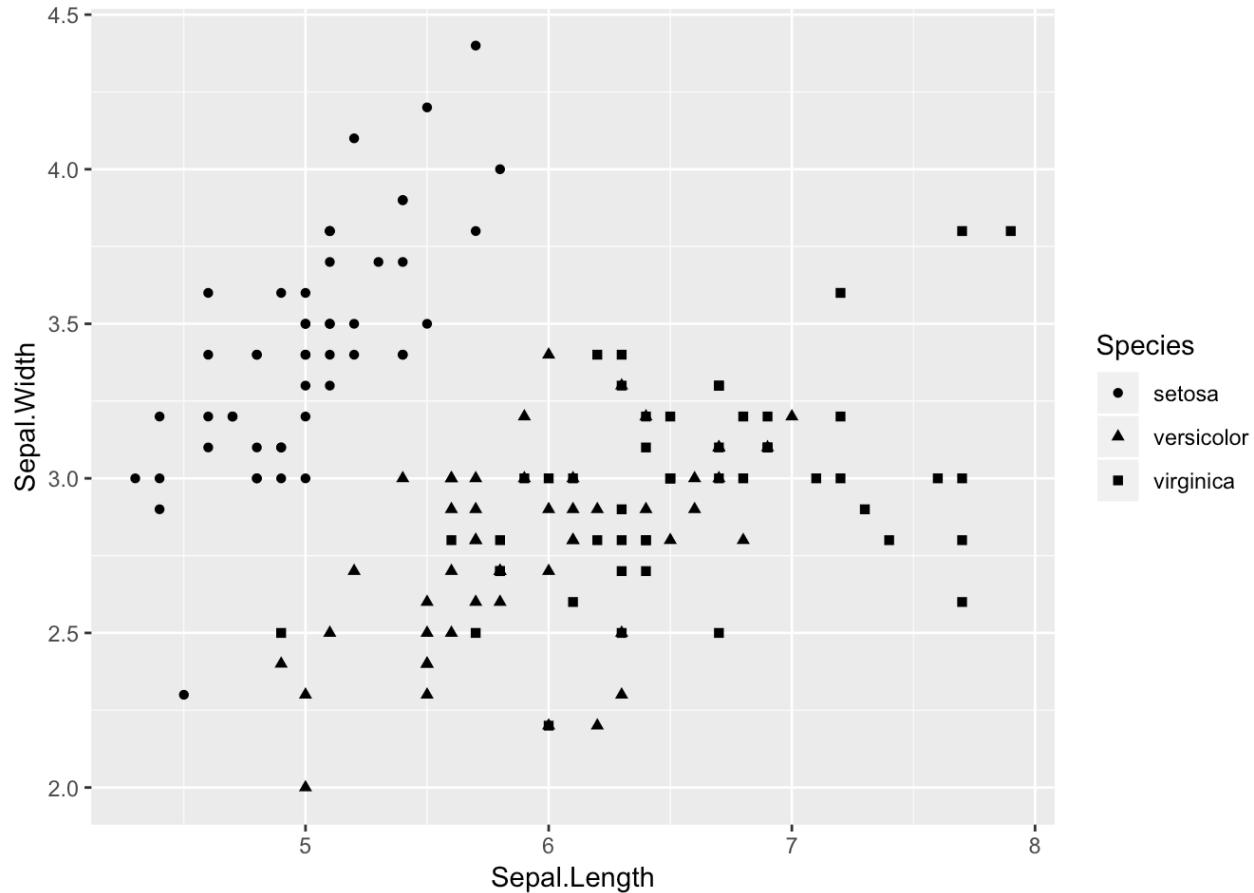
# Map categorical variable to colour

```
ggplot(data = iris) +  
  geom_point( aes(x = Sepal.Length, y = Sepal.Width,  
                  colour = Species))
```



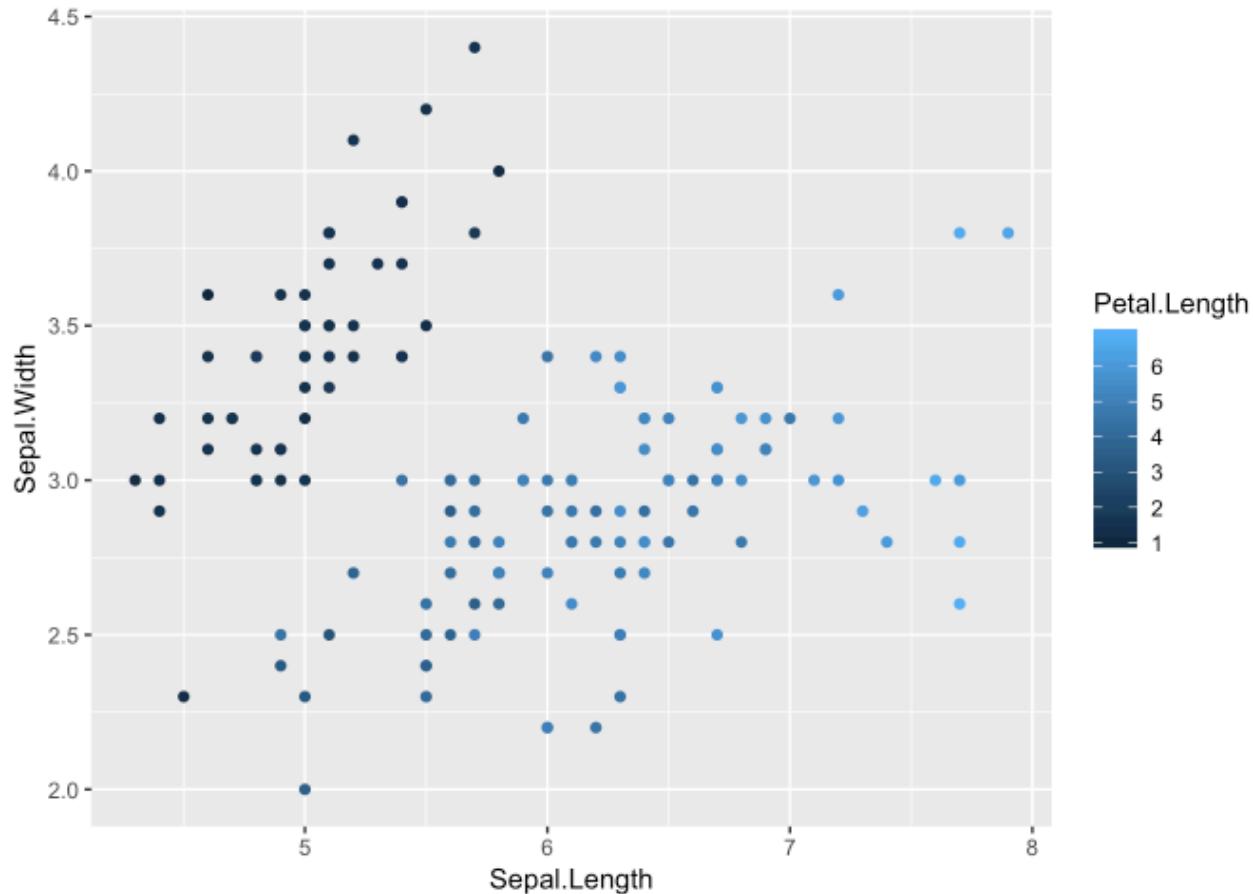
# Map categorical variable to shape

```
ggplot(data = iris) +  
  geom_point( aes(x = Sepal.Length, y = Sepal.Width,  
                  shape = Species))
```



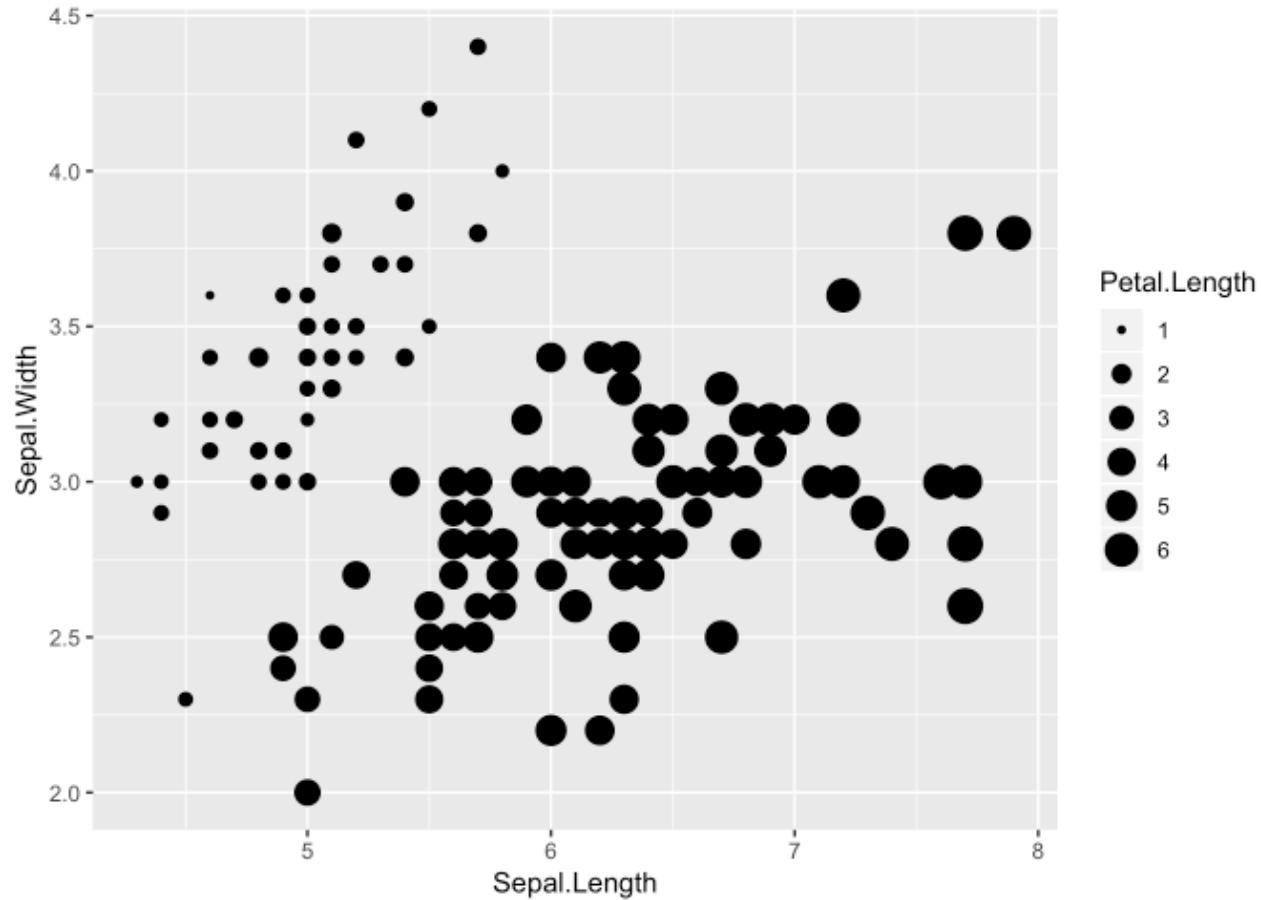
# Map continuous variable to colour

```
ggplot(data = iris) +  
  geom_point( aes(x = Sepal.Length, y = Sepal.Width,  
                  colour = Petal.Length))
```



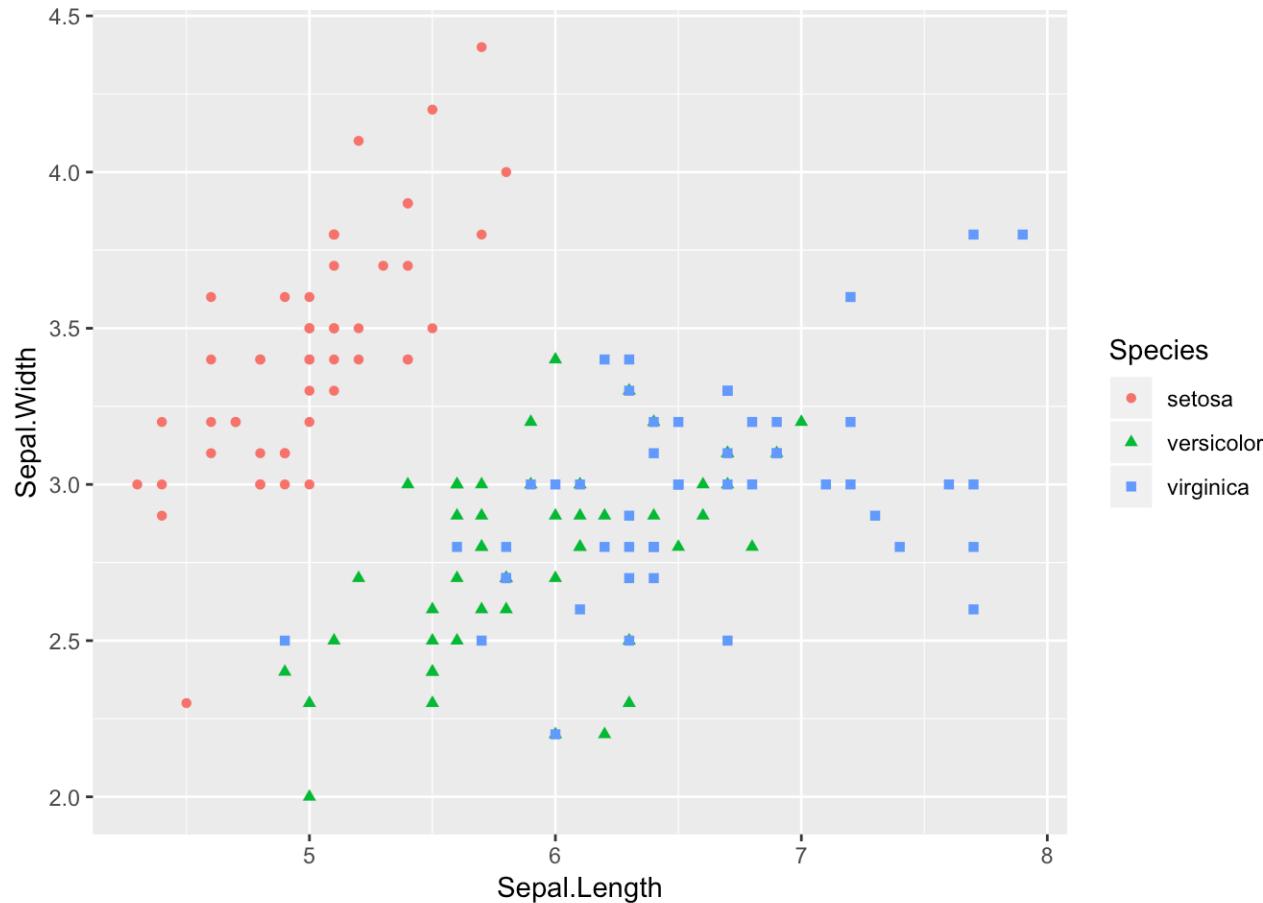
# Map continuous variable to size

```
ggplot(data = iris) +  
  geom_point( aes(x = Sepal.Length, y = Sepal.Width,  
                  size = Petal.Length))
```



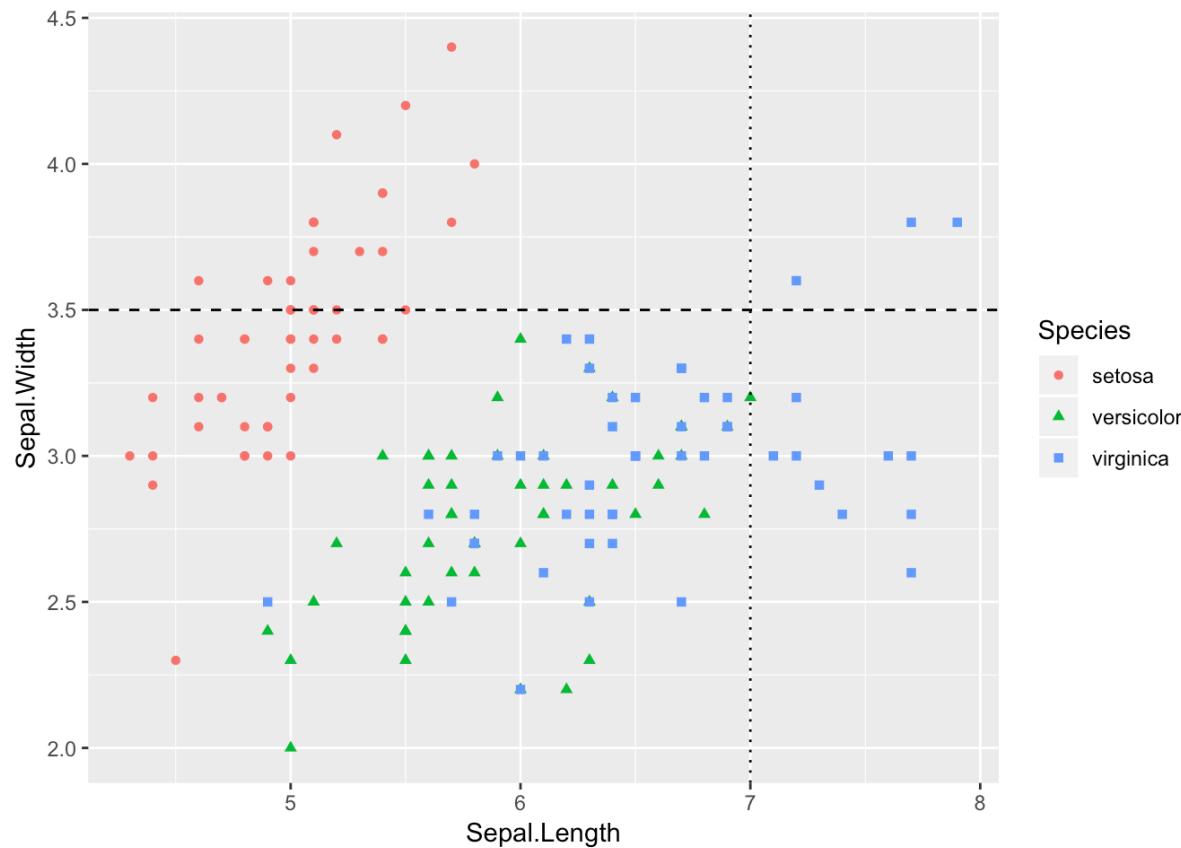
# Map variable to colour and shape

```
ggplot(data = iris) +  
  geom_point( aes(x = Sepal.Length, y = Sepal.Width,  
                  shape = Species, colour = Species))
```



# Add extra geoms

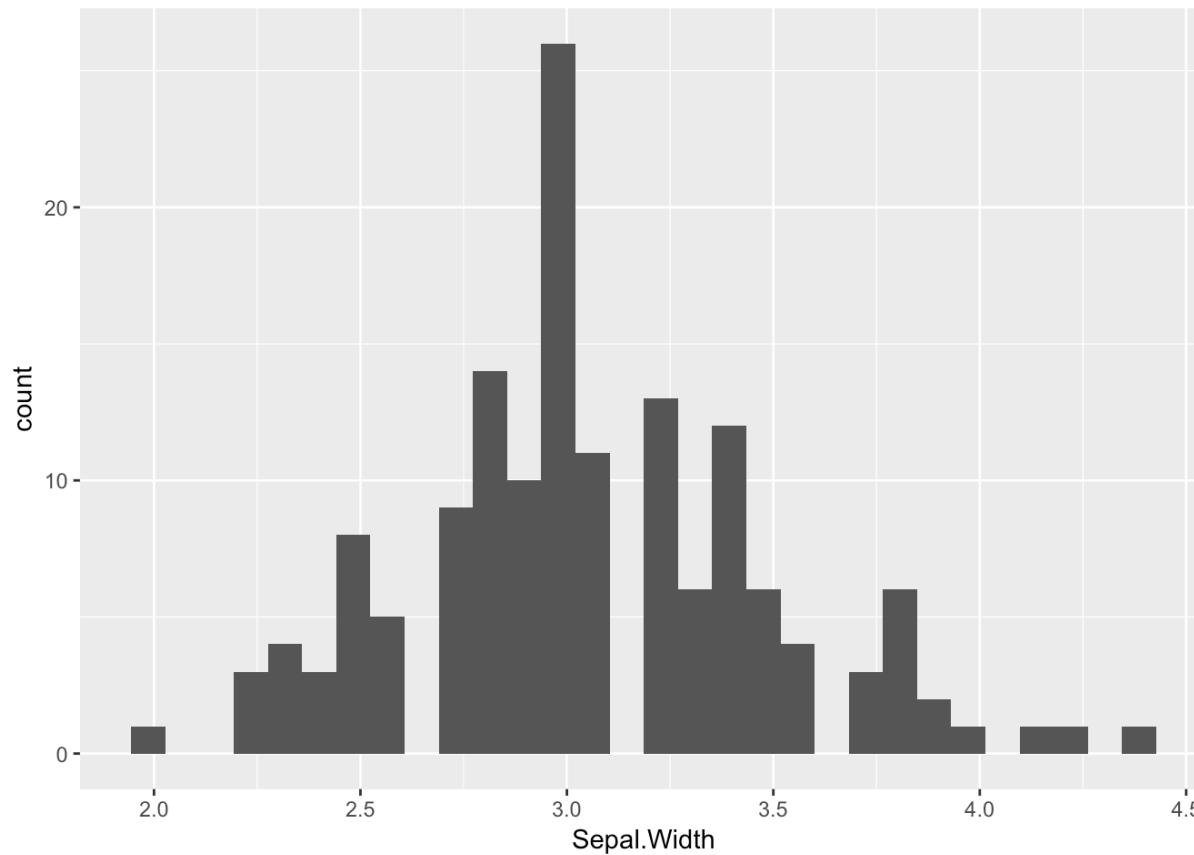
```
ggplot(data = iris) +  
  geom_point( aes(x = Sepal.Length, y = Sepal.Width,  
                  shape = Species, colour = Species)) +  
  geom_hline(yintercept = 3.5, linetype = 'dashed') +  
  geom_vline(xintercept = 7, linetype = 'dotted')
```



# geom\_histogram

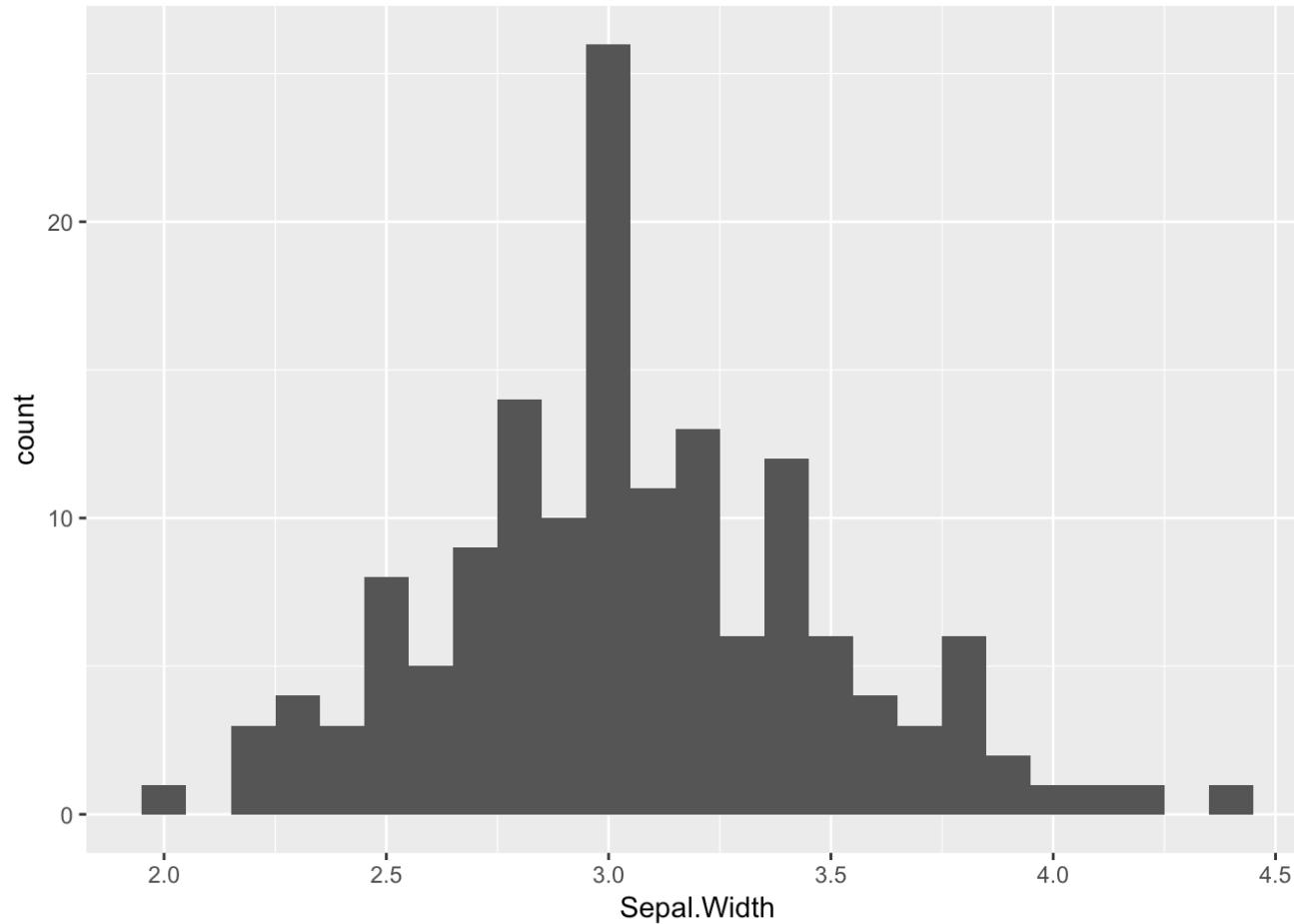
```
ggplot(data = iris) +  
  geom_histogram( aes(x = Sepal.Width) )
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



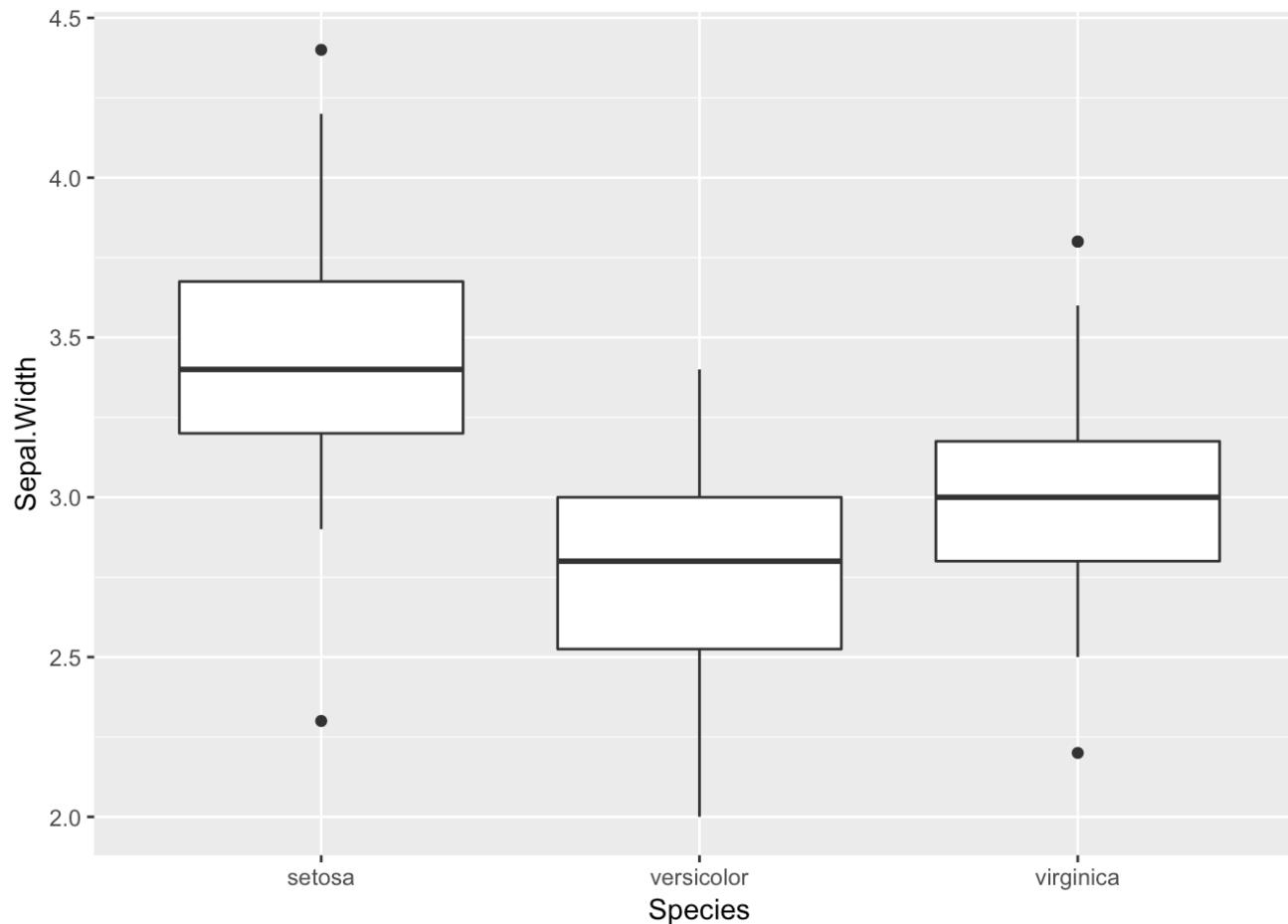
# setting the binwidth of a histogram

```
ggplot(data = iris) +  
  geom_histogram( aes(x = Sepal.Width), binwidth = 0.1 )
```



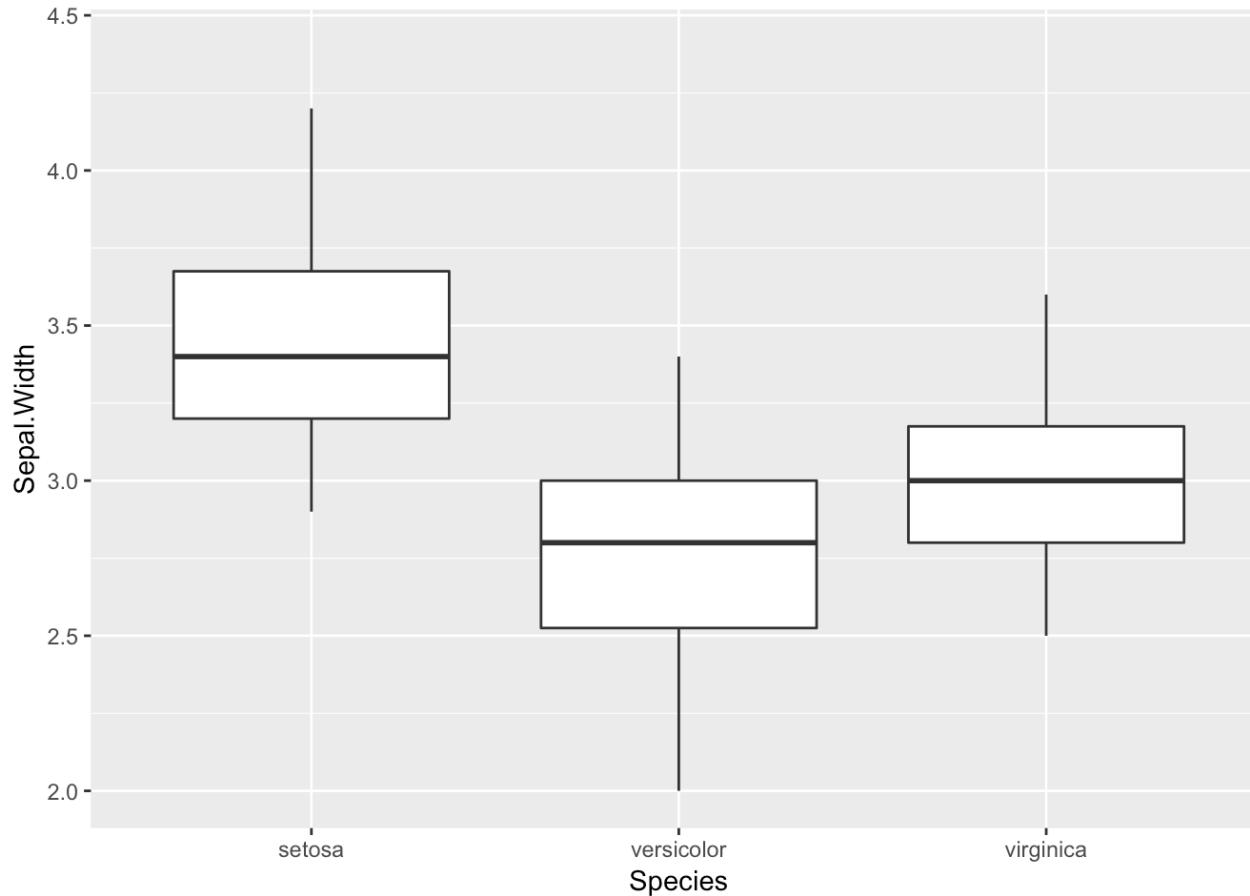
# Boxplots

```
ggplot(data = iris) +  
  geom_boxplot( aes(y = Sepal.Width, x = Species) )
```



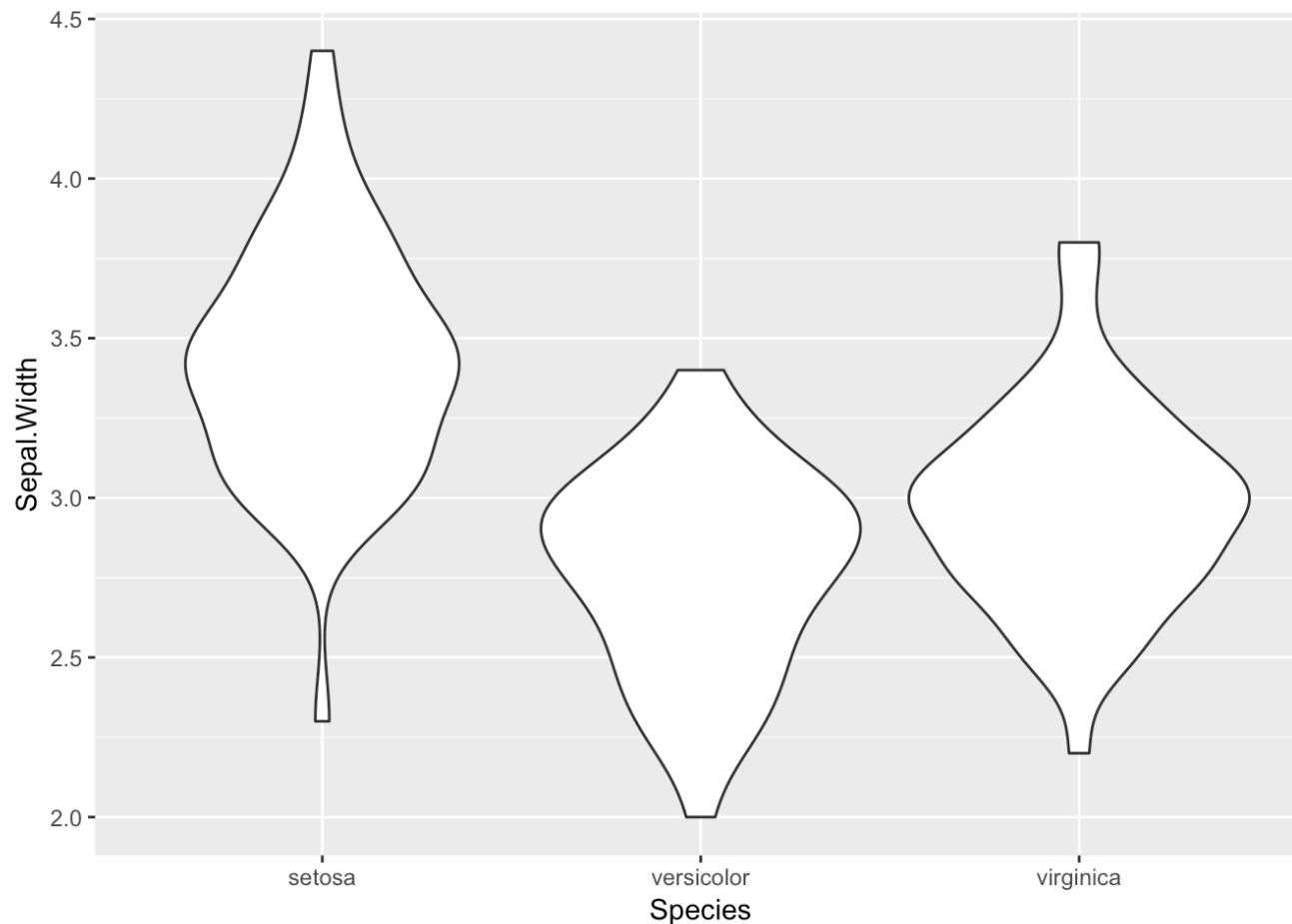
# Remove outliers

```
ggplot(data = iris) +  
  geom_boxplot( aes(y = Sepal.Width, x = Species),  
    outlier.shape = NA)
```



# Violin plot

```
ggplot(data = iris) +  
  geom_violin( aes(y = Sepal.Width, x = Species) )
```



# Colours

colours()

grDevices::colors

coral3	deeppink4	gray27	gray87	grey39	grey99	lightpink1	mistyrose1	pink4	slategray1	yellowgreen
coral2	deeppink3	gray26	gray86	grey38	grey98	lightpink	mistyrose	pink3	slategray4	yellow4
coral1	deeppink2	gray25	gray85	grey37	grey97	lightgrey	mintcream	pink2	slateblue4	yellow3
coral	deeppink1	gray24	gray84	grey36	grey96	lightgreen	midnightblue	pink1	slateblue3	yellow2
chocolate4	darkviolet	gray22	gray82	grey35	grey95	lightgray	mediumvioletred	beru	slateblue2	yellow1
chocolate3	darkturquoise	gray21	gray81	grey34	grey94	lightgoldenrod4	mediumturquoise	peachpuff4	skyblue3	yellow
chocolate2	darkslategray4	gray20	gray80	grey33	grey93	lightgoldenrod3	mediumspringgreen	peachpuff3	skyblue2	whitesmoke
chocolate1	darkslategray3	gray19	gray79	grey32	grey92	lightgoldenrod2	mediumsteelblue	peachpuff2	skyblue1	wheat4
chocolate	darkslategray4	gray19	gray78	grey31	grey91	lightgoldenrod1	mediumseagreen	peachpuff1	skyblue	wheat3
chartreuse4	darkslategray3	gray18	gray77	grey30	grey90	lightgoldenrod	mediumpurple4	peachpuff	papayawhip	wheat2
chartreuse3	darkslategray2	gray17	gray76	grey29	grey89	lightgoldenrod	mediumpurple3	peachpuff1	skyblue1	wheat1
chartreuse2	darkslategray1	gray16	gray75	grey28	grey88	lightcyan4	mediumpurple2	peachpuff	skyblue	wheat
chartreuse1	darkslategray	gray15	gray74	grey27	grey87	lightcyan3	mediumpurple1	palevioletred4	sienna4	violetred4
chartreuse	darkslateblue	gray14	gray73	grey26	grey86	lightcyan2	mediumpurple2	palevioletred3	sienna3	violetred3
cadetblue4	darkseagreen4	gray13	gray72	grey25	grey85	lightcyan1	mediumpurple3	palevioletred2	sienna3	violetred1
cadetblue3	darkseagreen3	gray12	gray71	grey24	grey84	lightcyan	mediumorchid4	palevioletred1	sienna1	violetred
cadetblue2	darkseagreen2	gray11	gray70	grey23	grey83	lightcoral	mediumorchid3	palevioletred	sienna	tomato4
cadetblue1	darkseagreen1	gray10	gray69	grey22	grey82	lightblue4	mediumorchid2	paleturquoise4	seashell4	tomato3
cadetblue	darkseagreen	gray9	gray68	grey21	grey81	lightblue3	mediumorchid1	paleturquoise3	seashell3	tomato2
burlywood4	darksalmon	gray8	gray67	grey20	grey80	lightblue2	mediumorchid	paleturquoise2	seashell2	tomato1
burlywood3	darkred	gray7	gray66	grey19	grey79	lightblue1	mediumblue	paleturquoise1	seashell1	thistle4
burlywood2	darkorchid4	gray6	gray65	grey18	grey78	lightblue	mediumaquamarine	paleturquoise1	seashell	thistle3
burlywood1	darkorchid3	gray5	gray64	grey17	grey77	lemonchiffon4	maroon4	paleturquoise	seagreen	thistle2
brown4	darkorchid1	gray3	gray63	grey16	grey76	lemonchiffon3	maroon3	palegreen4	seagreen4	thistle1
brown3	darkorchid	gray2	gray62	grey15	grey75	lemonchiffon2	maroon2	palegreen3	seagreen3	tan4
brown2	darkorange4	gray1	gray61	grey14	grey74	lemonchiffon1	maroon1	palegreen2	seagreen2	tan3
brown1	darkorange3	gray0	gray60	grey13	grey73	lemonchiffon	magenta4	palegreen1	seagreen1	tan2
brown	darkorange2	gray	gray59	grey12	grey72	lawngreen	magenta3	palegreen	seagreen	tan1
blueviolet	darkorange1	goldenrod4	gray58	grey11	grey71	lavenderblush4	maroon	palegoldenrod	sandybrown	steelblue4
blue4	darkorange	goldenrod3	gray57	grey10	grey70	lavenderblush3	magenta2	orchid4	salmon4	steelblue3
blue3	darkolivegreen4	goldenrod2	gray56	grey9	grey69	lavenderblush2	magenta1	orchid3	salmon3	steelblue2
blue2	darkolivegreen3	goldenrod1	gray55	grey8	grey68	lavenderblush1	magenta	orchid2	salmon2	steelblue1
blue1	darkolivegreen2	goldenrod	gray54	grey7	grey67	linen	lavenderblush	orchid1	salmon1	thistle4
blue	darkolivegreen1	gold4	gray53	grey6	grey66	lavender	limegreen	orchid	salmon	thistle3
blanchedalmond	darkolivegreen	gold3	gray52	grey5	grey65	lavenderblush4	lavender	orangered4	saddlebrown	thistle2
black	darkmagenta	gold2	gray51	grey4	grey64	lavenderblush3	khaki4	orangered3	royalblue4	thistle1
bisque4	darkkhaki	gold1	gray50	grey3	grey63	lavenderblush2	khaki3	orangered2	royalblue3	tan4
bisque3	darkgray	gold	gray49	grey2	grey62	lavenderblush1	khaki2	orangered1	royalblue2	tan3
bisque2	darkgreen	ghostwhite	gray48	grey1	grey61	khaki	lightsteelblue4	orange4	royalblue1	tan2
bisque1	darkgray	gainsboro	gray47	grey0	grey60	lavender	lightyellow4	orange3	rosybrown4	tan1
bisque	darkgoldenrod4	forestgreen	gray46	grey	grey59	lavenderblush	khaki4	orange3	rosybrown3	steelblue4
beige	darkgoldenrod3	floralwhite	gray45	green4	grey58	lavenderblush	lightyellow3	orange2	rosybrown2	steelblue3
azure4	darkgoldenrod2	firebrick4	gray44	green3	grey57	lavender	ivory4	orange2	rosybrown1	steelblue2
azure3	darkgoldenrod1	firebrick3	gray43	green2	grey56	lavenderblush	lightyellow2	orange1	rosybrown	steelblue1
azure2	darkgoldenrod	firebrick2	gray42	green1	grey55	lavender	ivory3	orange1	rosybrown2	springgreen4
azure1	darkcyan	firebrick1	gray41	green	grey54	lavenderblush	ivory2	orange1	rosybrown3	springgreen3
azure	darkblue	firebrick	gray40	gray100	grey53	lavender	ivory1	orange1	rosybrown1	springgreen2
aquamarine4	cyan4	dodgerblue4	gray39	gray99	grey51	indianred4	lightsategrey	olivedrab4	red4	snow4
aquamarine3	cyan3	dodgerblue3	gray38	gray98	grey50	indianred4	lightslategray	olivedrab3	red3	snow3
aquamarine2	cyan2	dodgerblue2	gray37	gray97	grey49	indianred3	lightslateblue	olivedrab2	red2	snow2
aquamarine1	cyan1	dodgerblue1	gray36	gray96	grey48	indianred3	lightskyblue4	olivedrab1	red1	snow1
aquamarine	cyan	dodgerblue	gray35	gray95	grey47	indianred1	lightskyblue3	olivedrab	oldlace	snow
antiquewhite4	cornsilk4	dimgrey	gray34	gray94	grey46	indianred1	lightskyblue2	navajowhite4	purple4	plum4
antiquewhite3	cornsilk3	dimgrey	gray33	gray93	grey45	hotpink4	lightskyblue1	navajowhite3	purple3	plum3
antiquewhite2	cornsilk2	deepskyblue4	gray32	gray92	grey44	hotpink2	lightskyblue	navajowhite2	purple2	plum2
antiquewhite1	cornsilk1	deepskyblue3	gray31	gray91	grey43	hotpink1	lightskyblue	navajowhite1	purple1	plum1
antiquewhite	cornsilk	deepskyblue2	gray30	gray90	grey42	hotpink1	lightskyblue	navajowhite	purple	plum
aliceblue	cornflowerblue	deepskyblue1	gray29	gray89	grey41	honeydew4	lightskyblue	navajowhite	powderblue	slategray4
white	coral4	deepskyblue	gray28	gray88	grey40	honeydew3	lightskyblue	navajowhite	plum4	slategray2

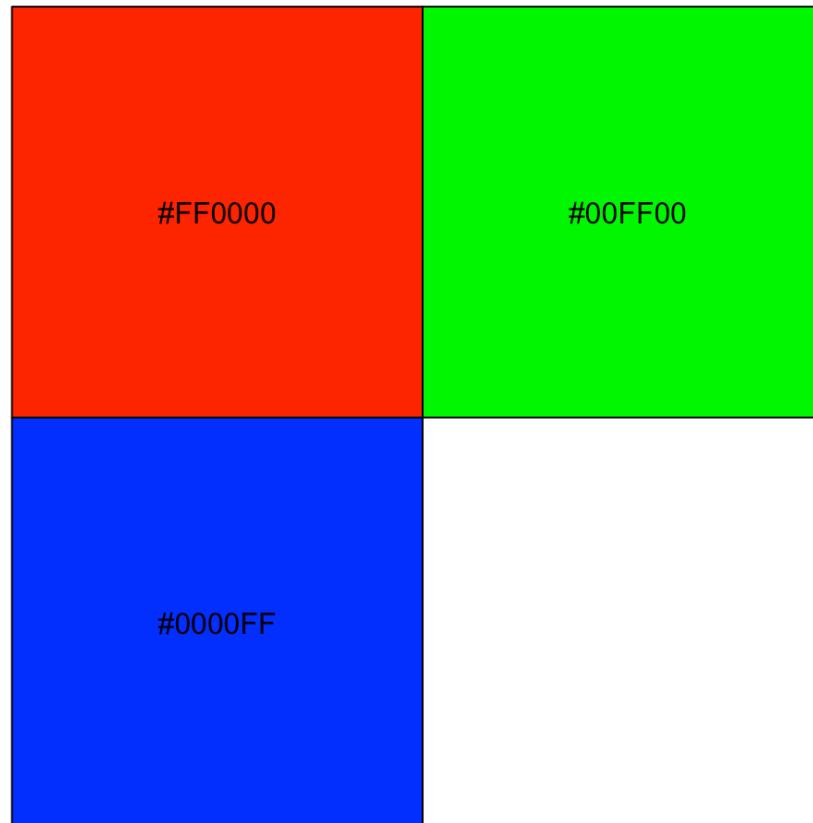
# specify colours by name

```
library(scales)  
show_col(c('red', 'blue', 'gold', 'green', 'firebrick', 'steelblue',  
'hotpink', 'royalblue1', 'mediumvioletred'))
```



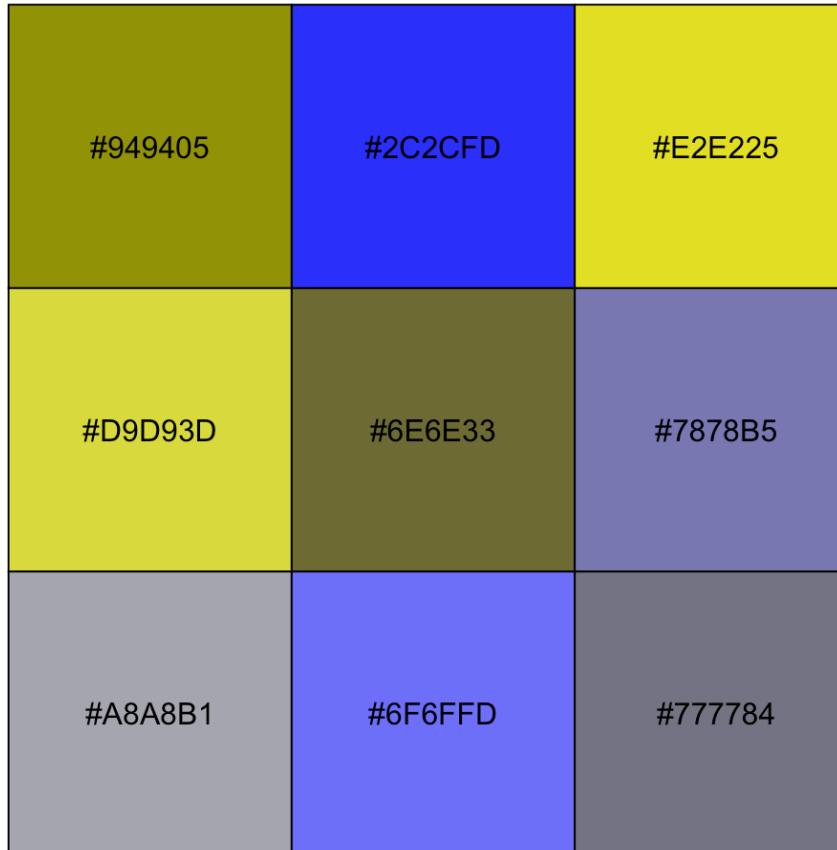
# specify colours as RGB values

```
rgb_colours <- c(rgb(255,0,0, maxColorValue = 255),  
                  rgb(0,255,0, maxColorValue = 255),  
                  rgb(0,0,255, maxColorValue = 255))  
show_col(rgb_colours)
```



# check how colour-blind friendly your palette is

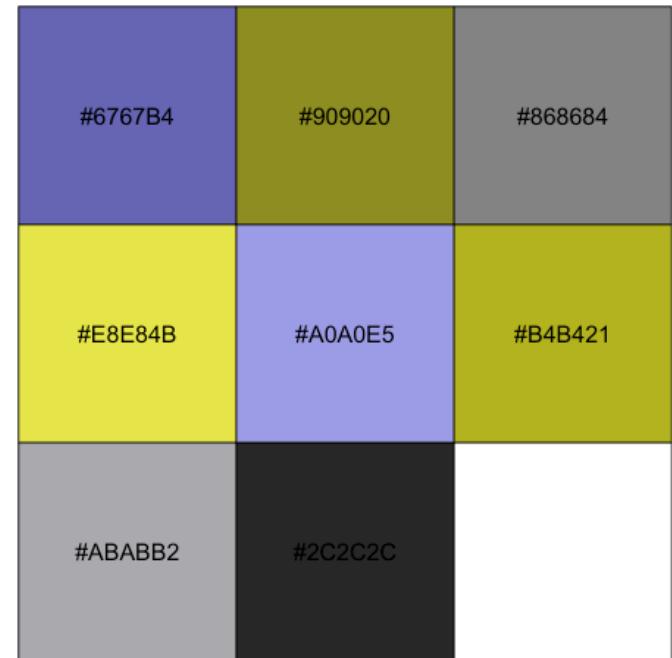
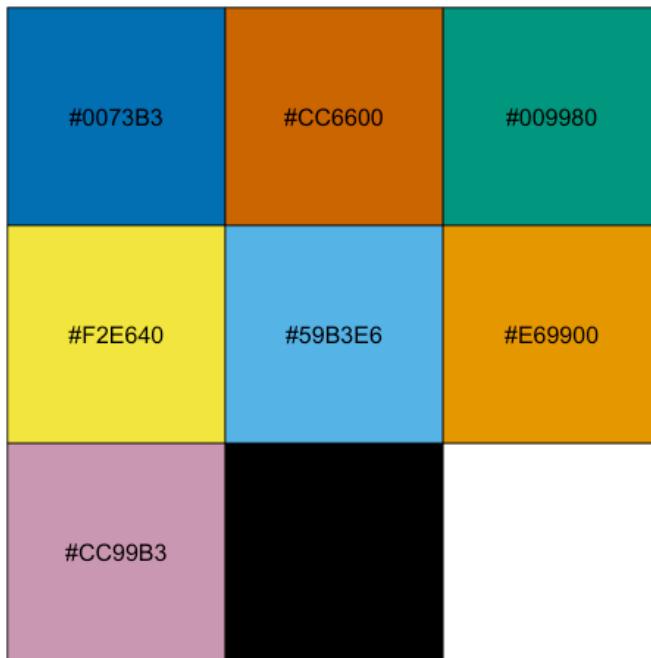
```
library(dichromat)
show_col(dichromat(c('red', 'blue', 'gold', 'green', 'firebrick', 'steelblue',
    'hotpink', 'royalblue1', 'mediumvioletred')))
```



# colour-blind friendly palette

```
colour_blind_palette <-  
  c( 'blue' = rgb(0,0.45,0.7),  
    'vermillion' = rgb(0.8, 0.4, 0),  
    'blue_green' = rgb(0, 0.6, 0.5),  
    'yellow' = rgb(0.95, 0.9, 0.25),  
    'sky_blue' = rgb(0.35, 0.7, 0.9),  
    'orange' = rgb(0.9, 0.6, 0),  
    'purple' = rgb(0.8, 0.6, 0.7),  
    'black' = rgb(0, 0, 0) )  
show_col(colour_blind_palette)
```

```
show_col(dichromat(colour_blind_palette))
```



<https://jfly.uni-koeln.de/color/#pallet>

<https://doi.org/10.1038/nmeth.1618>

# viridis: perceptually uniform colour scales



# viridis: perceptually uniform colour scales

```
library(viridis)  
show_col(viridis(16))
```

#440154FF	#481A6CFF	#472F7DFF	#414487FF
#39568CFF	#31688EFF	#2A788EFF	#23888EFF
#1F988BFF	#22A884FF	#35B779FF	#54C568FF
#7AD151FF	#A5DB36FF	#D2E21BFF	#FDE725FF

# viridis-plasma

```
show_col(plasma(16))
```

#0D0887FF	#330597FF	#5002A2FF	#6A00A8FF
#8405A7FF	#9C179EFF	#B12A90FF	#C33D80FF
#D35171FF	#E16462FF	#ED7953FF	#F68F44FF
#FCA636FF	#FEC029FF	#F9DC24FF	#F0F921FF

# ColorBrewer

Number of data classes: 7

Nature of your data:  
 sequential  diverging  qualitative

Pick a color scheme:

Only show:  
 colorblind safe  
 print friendly  
 photocopy safe

Context:  
 roads  
 cities  
 borders

Background:  
 solid color  
 terrain

color transparency

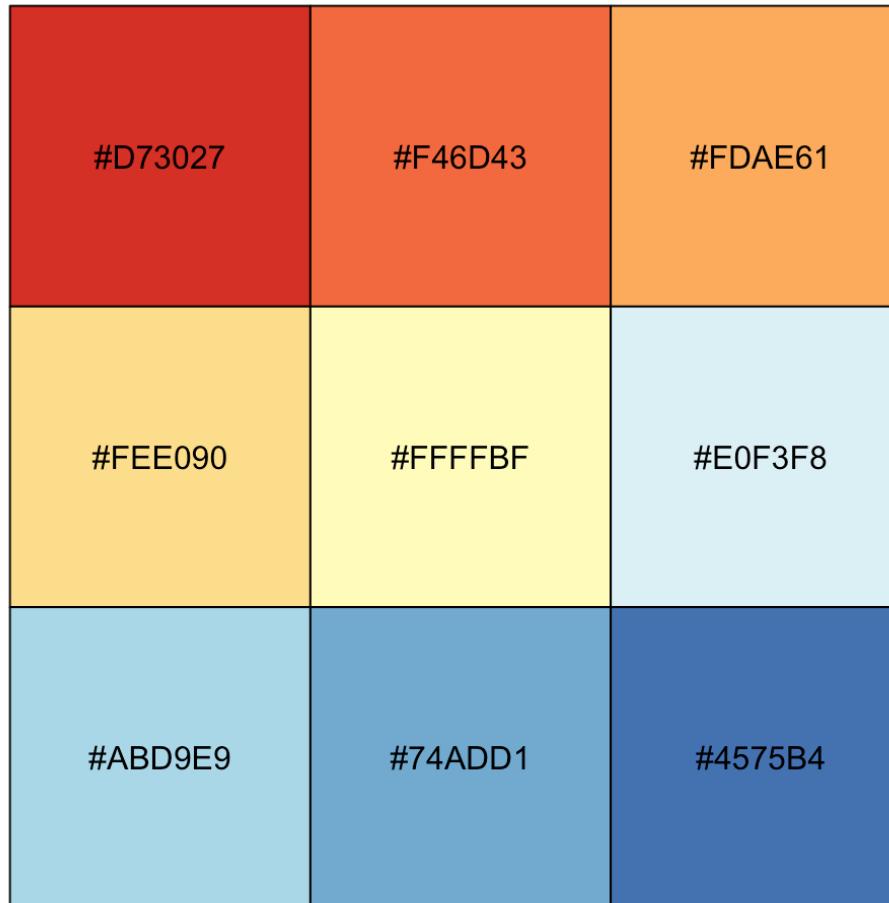
7-class RdYlBu

EXPORT

Color	Hex Code
#d73027	RdYlBu class 1
#fc8d59	
#fee090	
#ffffbf	
#e0f3f8	
#91bfdb	
#4575b4	

# ColorBrewer colour schemes

```
show_col(brewer_pal(type = "div", palette = 'RdYlBu', direction = 1)(9))
```



# Shapes

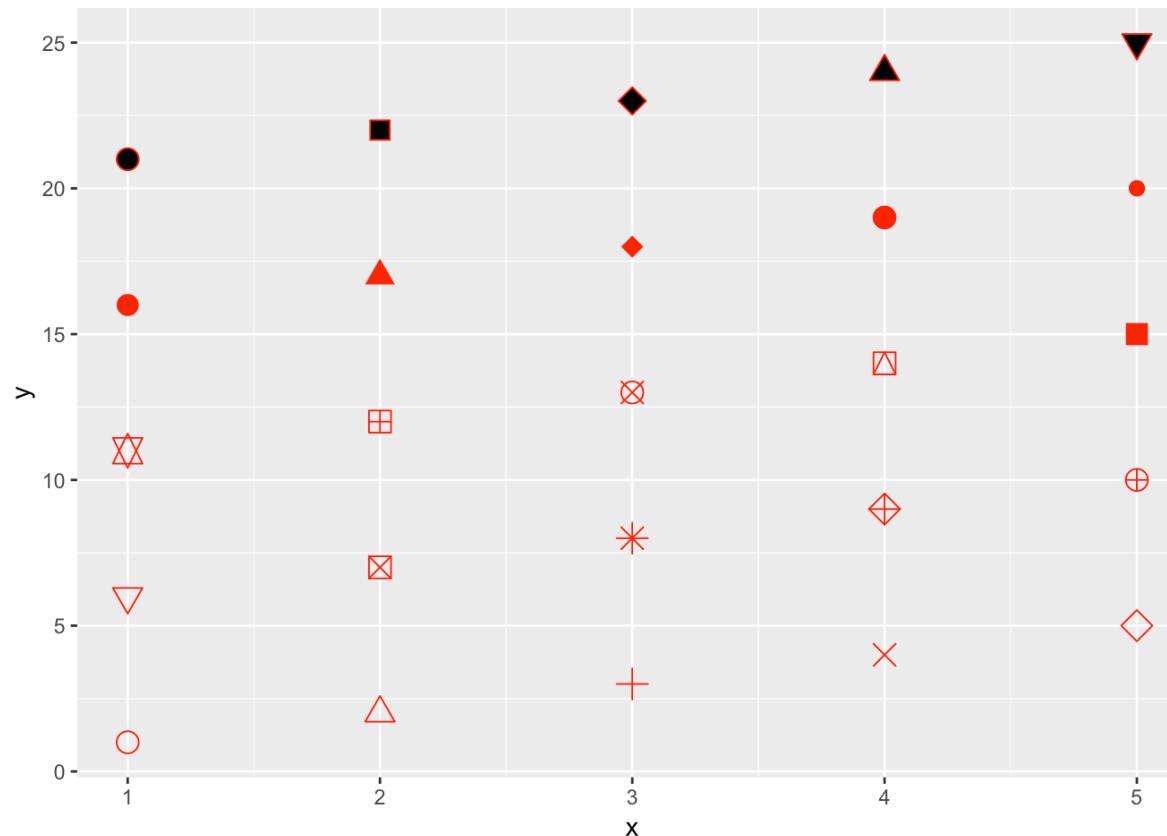
- There are 25 available shapes for plotting that are identified by numbers

□ 0	× 4	⊕ 10	■ 15	■ 22
○ 1	▽ 6	△△ 11	● 16	● 21
△ 2	◻ 7	田 12	▲ 17	▲ 24
◇ 5	* 8	⊗ 13	◆ 18	◆ 23
+ 3	◊ 9	▢ 14	● 19	● 20

- 0-14 are hollow. The border colour is determined by the colour aesthetic
- 15-20 are solid. The border colour is determined by the colour aesthetic
- 21-24 are filled shape that have a border colour and a fill colour

# Shapes

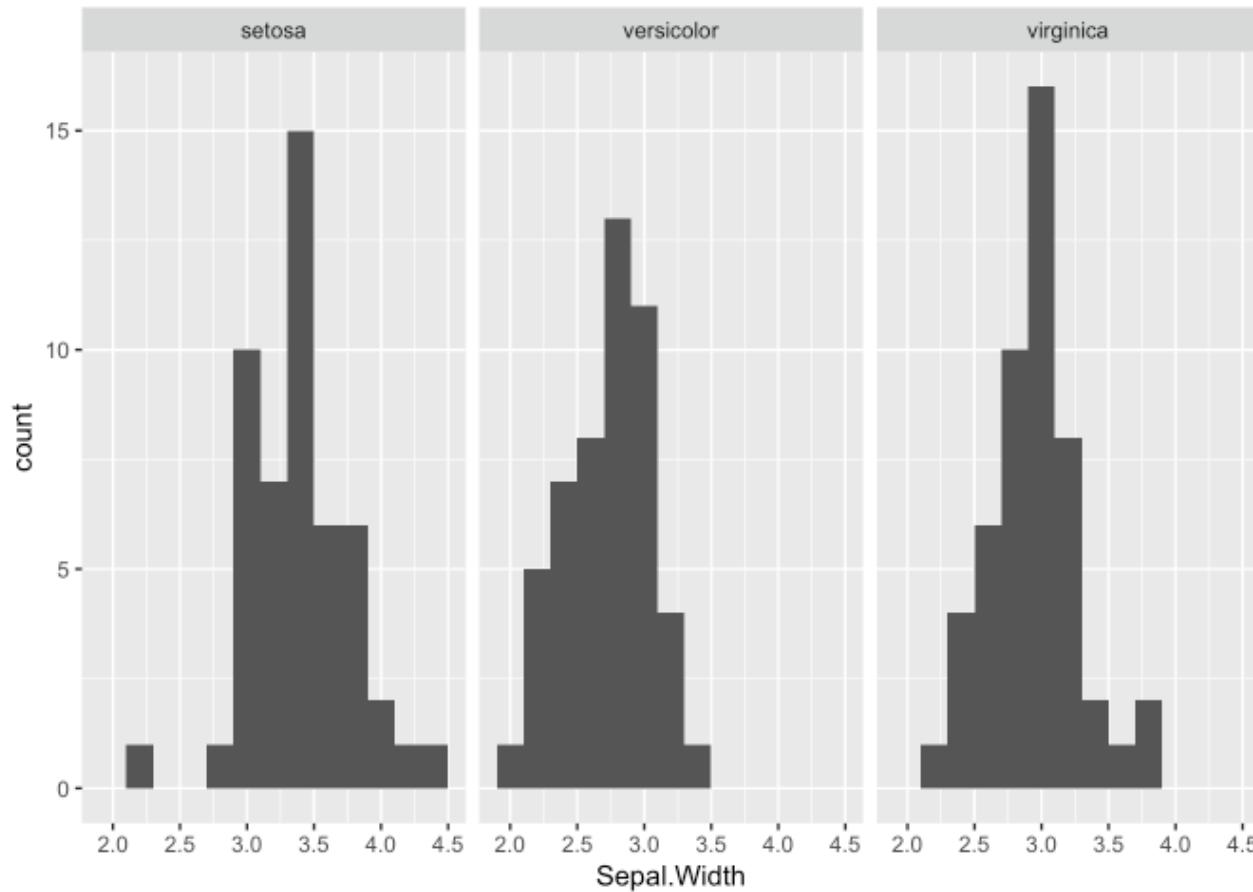
```
df2 <- data.frame(x = 1:5 , y = 1:25, z = 1:25)
ggplot(df2, aes(x, y)) +
  geom_point(aes(shape = z), size = 4,
             colour = "Red", fill = "Black") +
  scale_shape_identity()
```



[https://ggplot2.tidyverse.org/reference/aes\\_linetype\\_size\\_shape.html](https://ggplot2.tidyverse.org/reference/aes_linetype_size_shape.html)

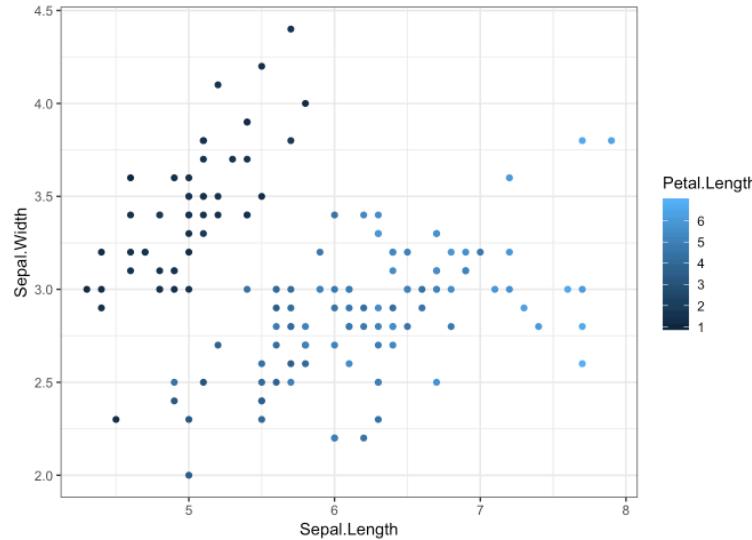
# Facets

```
ggplot(data = iris) +  
  geom_histogram( aes(x = Sepal.Width), binwidth = 0.2 ) +  
  facet_wrap(~ Species)
```

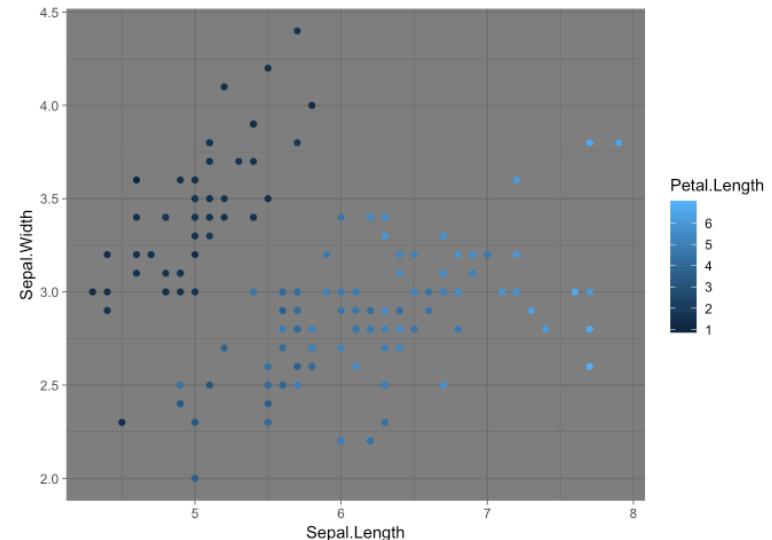


# Themes

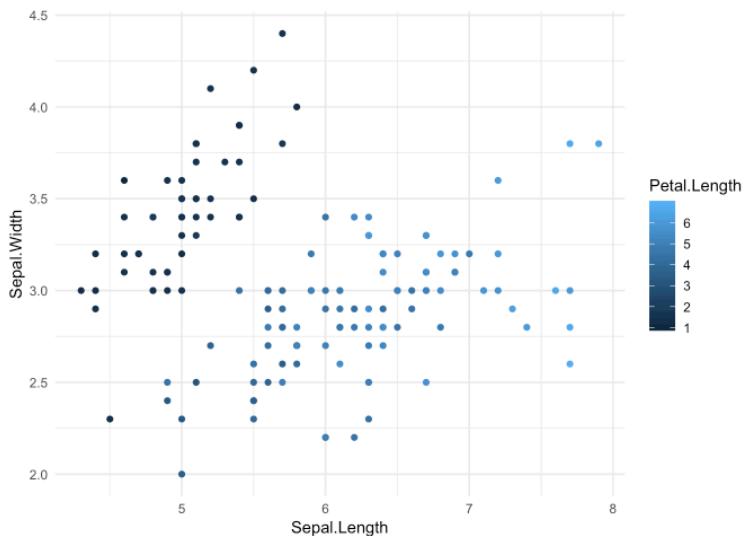
theme\_bw()



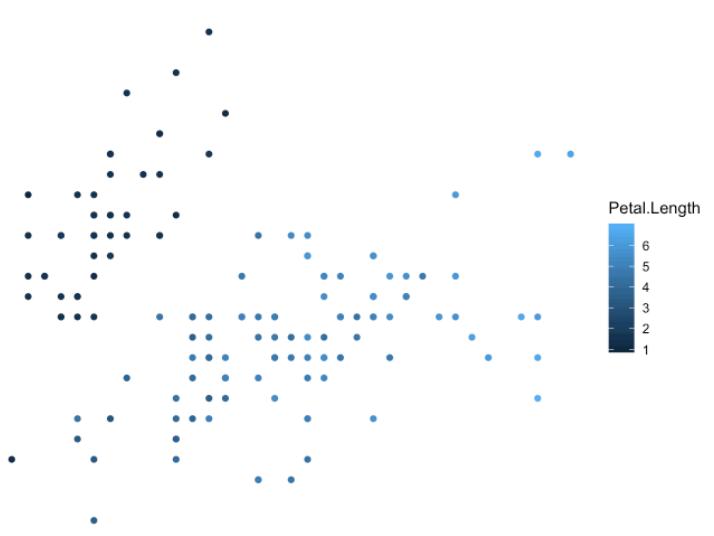
theme\_dark()



theme\_light()



theme\_void()



# Customising themes

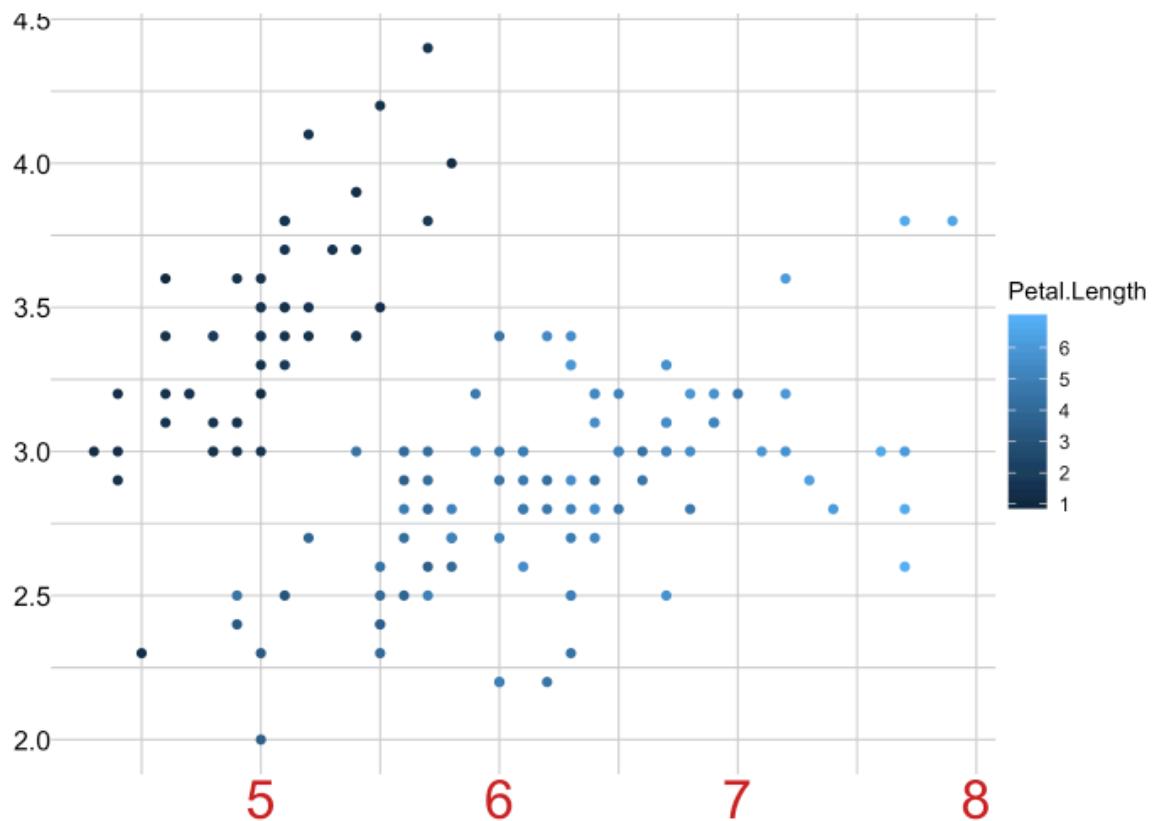
```
theme(line, rect, text, title, aspect.ratio, axis.title, axis.title.x,
axis.title.x.top, axis.title.x.bottom, axis.title.y, axis.title.y.left,
axis.title.y.right, axis.text, axis.text.x, axis.text.x.top,
axis.text.x.bottom, axis.text.y, axis.text.y.left, axis.text.y.right,
axis.ticks, axis.ticks.x, axis.ticks.x.top, axis.ticks.x.bottom,
axis.ticks.y, axis.ticks.y.left, axis.ticks.y.right, axis.ticks.length,
axis.ticks.length.x, axis.ticks.length.x.top, axis.ticks.length.x.bottom,
axis.ticks.length.y, axis.ticks.length.y.left, axis.ticks.length.y.right,
axis.line, axis.line.x, axis.line.x.top, axis.line.x.bottom, axis.line.y,
axis.line.y.left, axis.line.y.right, legend.background, legend.margin,
legend.spacing, legend.spacing.x, legend.spacing.y, legend.key,
legend.key.size, legend.key.height, legend.key.width, legend.text,
legend.text.align, legend.title, legend.title.align, legend.position,
legend.direction, legend.justification, legend.box, legend.box.just,
legend.box.margin, legend.box.background, legend.box.spacing,
panel.background, panel.border, panel.spacing, panel.spacing.x,
panel.spacing.y, panel.grid, panel.grid.major, panel.grid.minor,
panel.grid.major.x, panel.grid.major.y, panel.grid.minor.x,
panel.grid.minor.y, panel.on top, plot.background, plot.title,
plot.subtitle, plot.caption, plot.tag, plot.tag.position, plot.margin,
strip.background, strip.background.x, strip.background.y,
strip.placement, strip.text, strip.text.x, strip.text.y,
strip.switch.pad.grid, strip.switch.pad.wrap, ..., complete = FALSE,
validate = TRUE)
```

# Theme elements

- element\_text
  - font\_family, colour, size, angle
- element\_line
  - colour, size, linetype
- element\_rect: borders and backgrounds
  - fill, colour, size, linetype

# Customising themes

```
ggplot(data = iris) +  
  geom_point( aes(x = Sepal.Length, y = Sepal.Width,  
                  colour = Petal.Length)) +  
  theme_void() +  
  theme(axis.text = element_text(colour = "black", size = 12),  
        axis.text.x = element_text(colour = "firebrick3", size = 24),  
        panel.grid = element_line(colour = "grey80"))
```



# Further Reading

<https://ggplot2.tidyverse.org/reference/>

ggplot2 part of the tidyverse  
3.2.1

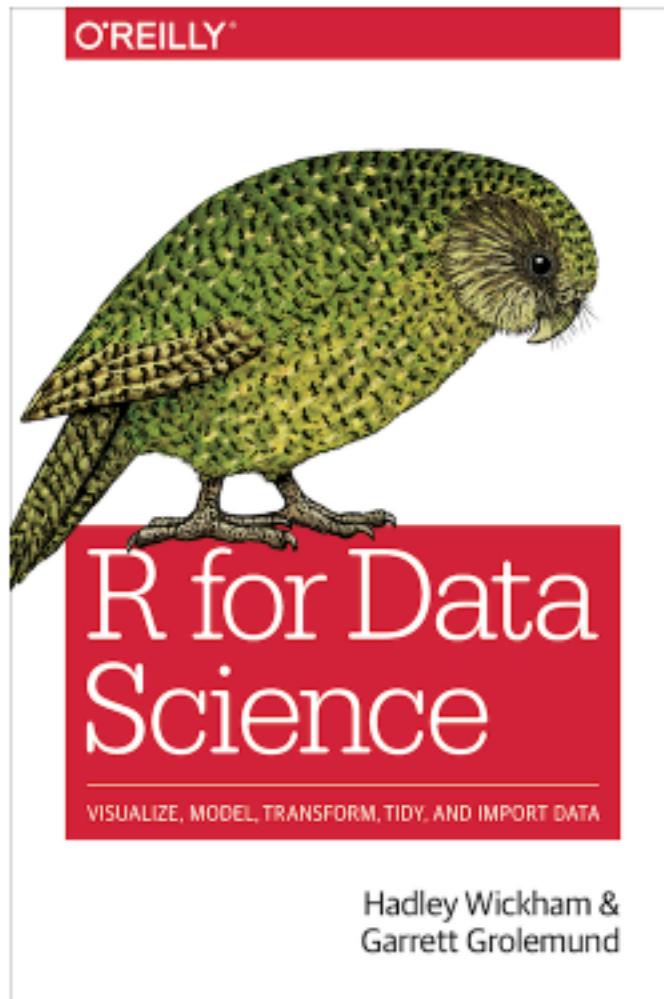
## Layer: geoms

A layer combines data, aesthetic mapping, a geom (geometric object), a stat (statistical transformation), and position adjustment. Typically, you will create layers using a `geom_` function, overriding the default position and

	<code>geom_abline()</code> <code>geom_hline()</code> <code>geom_vline()</code>	Reference lines: horizontal, vertical, and diagonal
	<code>geom_bar()</code> <code>geom_col()</code> <code>stat_count()</code>	Bar charts
	<code>geom_bin2d()</code> <code>stat_bin_2d()</code>	Heatmap of 2d bin counts
	<code>geom_blank()</code>	Draw nothing
	<code>geom_boxplot()</code> <code>stat_boxplot()</code>	A box and whiskers plot (in the style of Tukey)
	<code>geom_contour()</code> <code>stat_contour()</code>	2d contours of a 3d surface
	<code>geom_count()</code> <code>stat_sum()</code>	Count overlapping points
	<code>geom_density()</code> <code>stat_density()</code>	Smoothed density estimates

# Further Reading

- <https://r4ds.had.co.nz/>



# Exercises