

Data Frames

with Tidyverse

Kristin Scott, CMB PhD Candidate

Dataframes

- A data structure
- 2-dimensional table of rows and columns (like an excel spreadsheet)
- One of the most common data structures; flexible and intuitive
- Data types
 - Character (chr; e.g. “hello world!”)
 - Factor (fctr, categorical; e.g. up/down, +/-, A/T/U/G/C)
 - Numeric (integer (int), double (dbl))
 - date/time
 - null/missing values (**NULL** or **NA**)

df [18x4]

Column
names
(header)

Row index

Data

	date	instructor	topic	HW Due
1	01/17/22	NA	NA	NA
2	01/19/22	David	Intro/syllabus	NA
3	01/24/22	Erin	R/Rstudio	NA
4	01/26/22	Erin	R/Rstudio	NA
5	01/31/22	Brooke	Rmarkdown	Assignment 1
6	02/02/22	Brooke	Rmarkdown	NA
7	02/07/22	Jessica	Biological data/genome data	NA
8	02/09/22	Jessica	Biological data/genome data	Assignment 2
9	02/14/22	volunteer	Student vignettes	Assignment 3
10	02/16/22	volunteer	Student vignettes	NA
11	02/21/22	Kristin	Tidyverse	Assignment 4
12	02/23/22	Kristin	Tidyverse	NA
13	02/28/22	David	CMD1	Assignment 5
14	03/02/22	David	CMD1	NA
15	03/07/22	David	CMD2	Assignment 6
16	03/09/22	David	CMD2	NA
17	03/14/22	NA	NA	NA
18	03/16/22	NA	NA	NA

What data types is each column?

- character (“hello world!”)
- factor ([ordered categorical data](#))
- numeric ([integer, double](#))
- dates and times
- null/missing values ([NA](#))

chromosome	genomic_coordinate	nucleotide	strand	upstream_sequence	seq_length	sequence_frequency
3	25136245	A	+	AGTCATGCTG	10	0.002
4	8692145	T	-	TGCTGGTC	8	0.134
2	69314589	G	+	GCTAGTCAAAC	11	0.894

Wide format

Multiple data points (observations) per row

observations			
group	Replicate 1	Replicate 2	Replicate 3
control	12	11	14
treatment	29	21	27

variables		
replicate	control	treatment
Replicate 1	12	29
Replicate 2	11	21
Replicate 3	14	27

“Tidy” data:
→ Every column
is a variable
→ Every row
is an observation

Long format

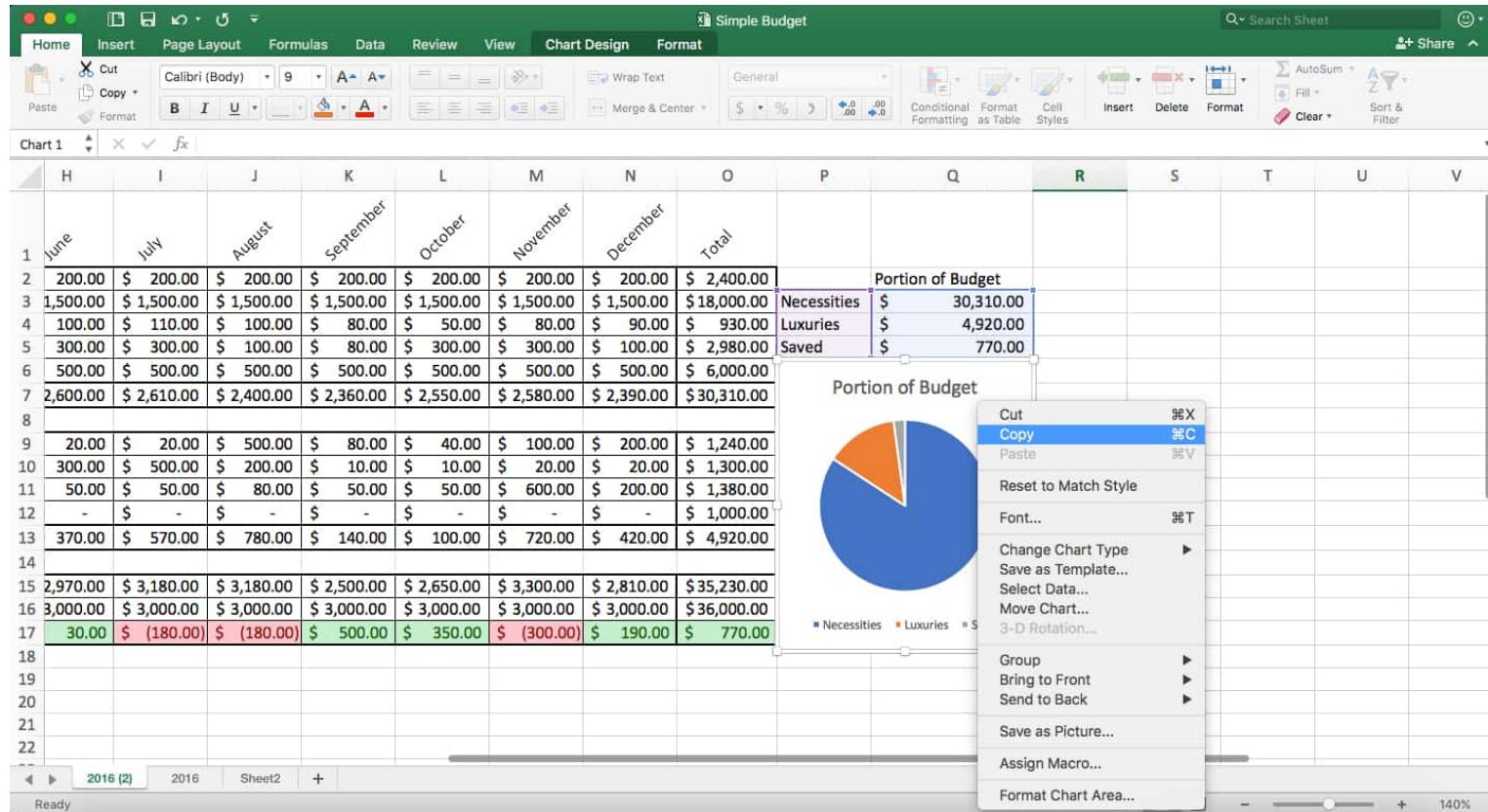
One data point (observation) per row

group	replicate	observation
control	1	12
control	2	11
control	3	14
treatment	1	29
treatment	2	21
treatment	3	27

All information for each
observation is encoded by a
row values, and only ONE
observation is listed per row

Storing 2D Dataframes

Excel spreadsheet



Storing 2D Dataframes

Delimited text file

Tab separated (TSV)

```
date    instructor  topic   HW Due
01/17/22  NA     NA      NA
01/19/22  David   Intro/syllabus  NA
01/24/22  Erin    R/Rstudio    NA
01/26/22  Erin    R/Rstudio    NA
01/31/22  Brooke  Rmarkdown   Assignment 1
02/02/22  Brooke  Rmarkdown   NA
02/07/22  Jessica Biological data/genome data NA
02/09/22  Jessica Biological data/genome data Assignment 2
02/14/22  volunteer Student vignettes Assignment 3
02/16/22  volunteer Student vignettes NA
02/21/22  Kristin Tidyverse Assignment 4
02/23/22  Kristin Tidyverse NA
02/28/22  David   CMD1     Assignment 5
03/02/22  David   CMD1     NA
03/07/22  David   CMD2     Assignment 6
03/09/22  David   CMD2     NA
03/14/22  NA     NA      NA
03/16/22  NA     NA      NA
```

Comma separated (CSV)

```
date,instructor,topic,HW Due
01/17/22,NA,NA,NA
01/19/22,David,Intro/syllabus,NA
01/24/22,Erin,R/Rstudio,NA
01/26/22,Erin,R/Rstudio,NA
01/31/22,Brooke,Rmarkdown,Assignment 1
02/02/22,Brooke,Rmarkdown,NA
02/07/22,Jessica,Biological data/genome data,NA
02/09/22,Jessica,Biological data/genome data,Assignment 2
02/14/22,volunteer,Student vignettes,Assignment 3
02/16/22,volunteer,Student vignettes,NA
02/21/22,Kristin,Tidyverse,Assignment 4
02/23/22,Kristin,Tidyverse,NA
02/28/22,David,CMD1,Assignment 5
03/02/22,David,CMD1,NA
03/07/22,David,CMD2,Assignment 6
03/09/22,David,CMD2,NA
03/14/22,NA,NA,NA
03/16/22,NA,NA,NA
```

* separated

```
date*instructor*topic*HW Due
01/17/22*NA*NA*NA
01/19/22*David*Intro/syllabus*NA
01/24/22*Erin*R/Rstudio*NA
01/26/22*Erin*R/Rstudio*NA
01/31/22*Brooke*Rmarkdown*Assignment 1
02/02/22*Brooke*Rmarkdown*NA
02/07/22*Jessica*Biological data/genome data*NA
02/09/22*Jessica*Biological data/genome data*Assignment 2
02/14/22*volunteer*Student vignettes*Assignment 3
02/16/22*volunteer*Student vignettes*NA
02/21/22*Kristin*Tidyverse*Assignment 4
02/23/22*Kristin*Tidyverse*NA
02/28/22*David*CMD1*Assignment 5
03/02/22*David*CMD1*NA
03/07/22*David*CMD2*Assignment 6
03/09/22*David*CMD2*NA
03/14/22*NA*NA*NA
03/16/22*NA*NA*NA
```

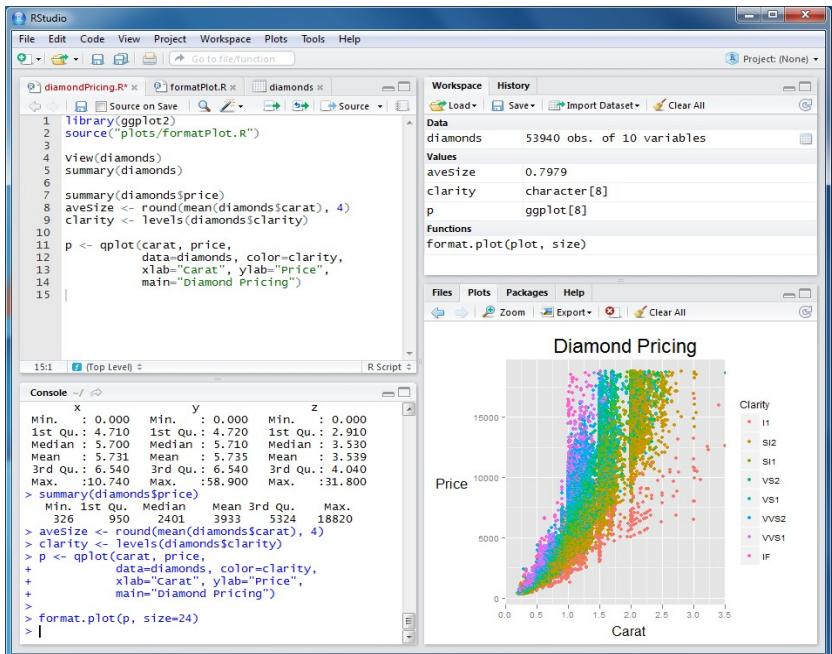
Manipulating Dataframes

- Add/remove/modifying/retrieving data
- Large data frames can be difficult to compute
- Efficiency?
- Accuracy?
- Reproducible?
- Transparent?



R is a programming language for **statistical computing** and **graphics** supported by the R Core Team and the R Foundation for Statistical Computing.

- **Created by statisticians** Ross Ihaka and Robert Gentleman
- R is used among data miners and statisticians **for data analysis** and developing statistical software



Data Frames

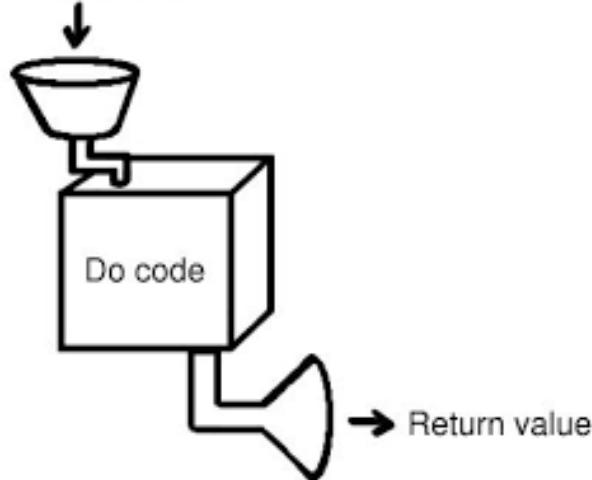
with



The tidyverse is a **collection of R packages** introduced by Hadley Wickham and his team that share an underlying design philosophy, grammar, and data structures of tidy data.

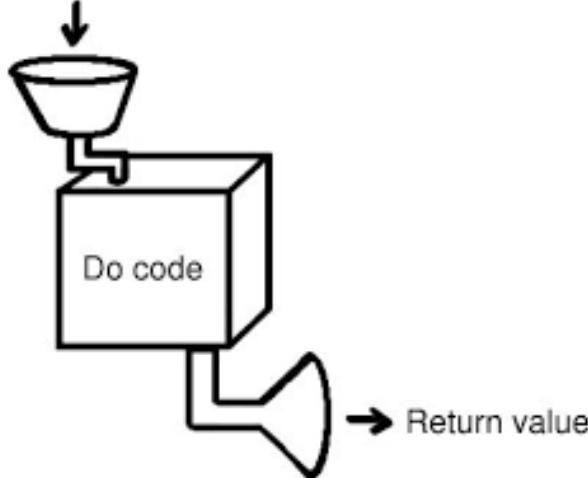


Input variables



Functions

Input variables



Functions

```
> function(argument1, argument2, ....)  
[1] output
```

```
> ls <- c(3,4,5,6)  
> mean(ls)  
[1] 4.5
```



```
df <- dataframe  
> function(arg1 = df, arg2, arg3, ....)  
[1] df # modified df with function applied
```

```
new_df <- df
```



```
df <- dataframe  
> function(arg1 = df, arg2, arg3, ....)  
[1] df # modified df with function applied
```

```
new_df <- df %>%  
  function(arg2, arg3, ...) %>%  
  function(arg2, arg3, ...) %>%  
  function(arg2, arg3, ...)
```

Install tidyverse:

```
# In your R console, run:  
install.packages("tidyverse")
```

Import tidyverse into
your session



The image shows a screenshot of RStudio. At the top, there's a toolbar with various icons. Below it is a tab bar with 'dataframes.rmd' selected. The main area contains R code. Lines 10 through 15 show the setup of an R Markdown chunk to include the 'tidyverse' package. Line 16 shows the command to import the package. The output pane at the bottom shows the results of running this code in the console. It includes messages about S3 method overwriting, package attachments, and conflicts.

```
9  
10 ```{r setup, include=FALSE}  
11 knitr::opts_chunk$set(echo = TRUE)  
12  
13 library("tidyverse")  
14  
15 ```  
16  
17  
16:1 (Top Level) R Markdown
```

Console Terminal × R Markdown × Jobs ×

```
~/  
> library("tidyverse")  
Registered S3 methods overwritten by 'dbplyr':  
  method      from  
  print.tbl_lazy  
  print.tbl_sql  
— Attaching packages ————— tidyverse 1.3.1 —  
✓ ggplot2 3.3.3    ✓ purrr  0.3.4  
✓ tibble  3.1.6    ✓ dplyr   1.0.7  
✓ tidyr   1.2.0    ✓ stringr 1.4.0  
✓ readr   1.4.0    ✓ forcats 0.5.1  
— Conflicts ————— tidyverse_conflicts() —  
x dplyr::filter() masks stats::filter()  
x dplyr::lag()   masks stats::lag()  
>  
>
```

Tidyverse and
associated packages
are imported

Manipulations we will work on:

- Mutate, select, rename, filter, merge, arrange, gather, spread
- Other functions: grouping & summarizing, custom functions, reading/writing data
- Datatype conversions

If time permit:

- If time REALLY permit: Data visualization with ggplot2

mutate (adding columns)

```
mutate( df, new_column = <formula> )
```

```
new_df <- df %>%  
  mutate( new_column = <formula> )
```

```
new_df <- df %>%  
  mutate( fold_change = treatment / control )
```

> df

replicate	control	treatment
1	12	29
2	11	21
3	14	27

> new_df

replicate	control	treatment	fold_change
1	12	29	2.4
2	11	21	1.91
3	14	27	1.92

mutate (adding columns)

```
mutate( df, new_column = <formula> )
```

```
new_df <- df %>%  
  mutate( new_column = <formula> )
```

```
new_df <- df %>%  
  mutate( fold_change = treatment / control )
```

```
new_df <- df %>%  
  mutate( fold_change = treatment / control ) %>%  
  mutate( log2FC = log2(fold_change) )
```

> df

replicate	control	treatment
1	12	29
2	11	21
3	14	27

> new_df

replicate	control	treatment	fold_change	log2FC
1	12	29	2.4	1.2
2	11	21	1.91	0.93
3	14	27	1.92	0.94

select (selecting/removing columns)

> df

```
select(df, column_name, -column_name)
```

```
new_df <- df %>%  
  select(column_name, -column_name)
```

```
new_df <- df %>%  
  select(replicate, control, treatment,  
         fold_change)
```

OR...

```
new_df <- df %>%  
  select(-log2FC)
```

replicate	control	treatment	fold_change	log2FC
1	12	29	2.4	1.2
2	11	21	1.91	0.93
3	14	27	1.92	0.94

> new_df

replicate	control	treatment	fold_change
1	12	29	2.4
2	11	21	1.91
3	14	27	1.92

select() can also be used to re-order columns

rename (renaming columns)

```
dplyr::rename(df, new_column_name=old_column_name) > df
```

```
new_df <- df %>%  
  dplyr::rename(new_name = old_name)
```

```
new_df <- df %>%  
  dplyr::rename(FC = fold_change)
```

replicate	control	treamtment	fold_change
1	12	29	2.4
2	11	21	1.91
3	14	27	1.92

> new_df

replicate	control	treamtment	FC
1	12	29	2.4
2	11	21	1.91
3	14	27	1.92

filter (select rows based on value)

```
filter(df, column_name <operation> <formula>)
```

> df

```
new_df <- df %>%  
  filter( column_name <operation> <formula> )
```

Operations:

==, !=, >, <
&, |

gene	RPKM control	RPKM treatment	fold change	log2FC	pvalue
gene1	15	14	0.93	-0.10	0.95
gene2	2	5	2.50	1.32	0.68
gene3	4	7	1.75	0.81	0.45
gene4	6	10	1.67	0.74	0.79
gene5	32	10	0.31	-1.68	0.02
gene6	45	56	1.24	0.32	0.55
gene7	12	1	0.08	-3.58	0.001
gene8	65	57	0.88	-0.19	0.37
gene9	12	12	1.00	0.00	1
gene10	19	70	3.68	1.88	0.011

```
new_df <- df %>%  
  filter(log2FC > 1 | log2FC < -1)
```

> df

gene	RPKM control	RPKM treatment	fold change	log2FC	pvalue
gene1	15	14	0.93	-0.10	0.95
gene2	2	5	2.50	1.32	0.68
gene3	4	7	1.75	0.81	0.45
gene4	6	10	1.67	0.74	0.79
gene5	32	10	0.31	-1.68	0.02
gene6	45	56	1.24	0.32	0.55
gene7	12	1	0.08	-3.58	0.001
gene8	65	57	0.88	-0.19	0.37
gene9	12	12	1.00	0.00	1
gene10	19	70	3.68	1.88	0.011

> new_df

gene	RPKM control	RPKM treatment	fold change	log2FC	pvalue
gene2	2	5	2.50	1.32	0.68
gene5	32	10	0.31	-1.68	0.02
gene7	12	1	0.08	-3.58	0.001
gene10	19	70	3.68	1.88	0.011

> df

gene	RPKM control	RPKM treatment	fold change	log2FC	pvalue
gene1	15	14	0.93	-0.10	0.95
gene2	2	5	2.50	1.32	0.68
gene3	4	7	1.75	0.81	0.45
gene4	6	10	1.67	0.74	0.79
gene5	32	10	0.31	-1.68	0.02
gene6	45	56	1.24	0.32	0.55
gene7	12	1	0.08	-3.58	0.001
gene8	65	57	0.88	-0.19	0.37
gene9	12	12	1.00	0.00	1
gene10	19	70	3.68	1.88	0.011

```
new_df <- df %>%
  filter(log2FC > 1 | log2FC < -1) %>%
  filter(pvalue < 0.05)
```

OR...

```
new_df <- df %>%
  filter( (log2FC > 1 | log2FC < -1) & pvalue < 0.05)
```

> new_df

gene	RPKM control	RPKM treatment	fold change	log2FC	pvalue
gene5	32	10	0.31	-1.68	0.02
gene7	12	1	0.08	-3.58	0.001
gene10	19	70	3.68	1.88	0.011

```
new_df <- df %>%
  filter( gene == "gene1" )
```

> new_df

gene	RPKM control	RPKM treatment	fold change	log2FC	pvalue
gene1	15	14	0.93	-0.10	0.95

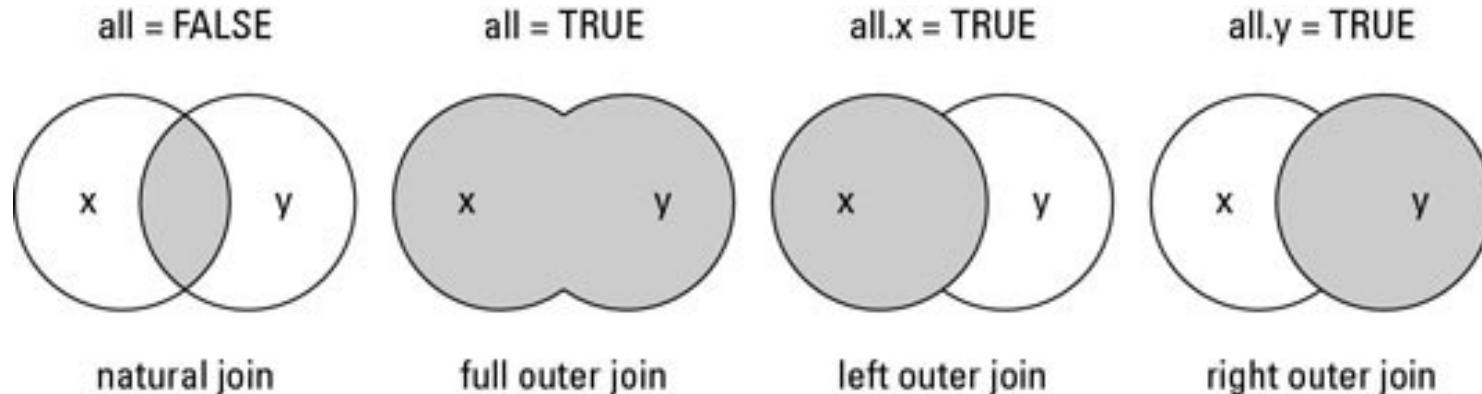
```
new_df <- df %>%
  filter( gene != "gene1" )
```

> new_df

gene	RPKM control	RPKM treatment	fold change	log2FC	pvalue
gene2	2	5	2.50	1.32	0.68
gene3	4	7	1.75	0.81	0.45
gene4	6	10	1.67	0.74	0.79
gene5	32	10	0.31	-1.68	0.02
gene6	45	56	1.24	0.32	0.55
gene7	12	1	0.08	-3.58	0.001
gene8	65	57	0.88	-0.19	0.37
gene9	12	12	1.00	0.00	1
gene10	19	70	3.68	1.88	0.011

merge dataframes

```
merge(x = df.x, y = df.y, by = c("column_name"), all.x = TRUE)  
      all.y = TRUE  
      all = TRUE  
      all = FALSE
```



```

new_df <- df %>%
  filter(log2FC > 1 | log2FC < -1) %>%
  filter(pvalue < 0.05)

```

> new_df

gene	RPKM control	RPKM treatment	fold change	log2FC	pvalue
gene5	32	10	0.31	-1.68	0.02
gene7	12	1	0.08	-3.58	0.001
gene10	19	70	3.68	1.88	0.011

> df2

gene	symbol	description
gene1	A1BG	alpha-1-B glycoprotein
gene2	NAT2	N-acetyltransferase 2
gene3	ADA	adenosine deaminase
gene4	CDH2	cadherin 2
gene5	GAGE12F	G antigen 12F
gene6	MED6	mediator complex subunit 6
gene7	NAALAD2	N-acetylated acidic dipeptidase 2
gene8	DUXB	double homeobox B
gene9	DDTL	D-dopachrome tautomerase like
gene10	ZBTB33	zinc finger and BTB domain containing 33

```

new_df <- df %>%
  filter(log2FC > 1 | log2FC < -1) %>%
  filter(pvalue < 0.05) %>%
  merge(y=df2, by= "gene", all.x=TRUE)

```

> new_df

gene	RPKM control	RPKM treatment	fold change	log2FC	pvalue	symbol	description
gene 5	32	10	0.31	-1.68	0.02	GAGE1 2F	G antigen 12F
gene 7	12	1	0.08	-3.58	0.001	NAALA D2	N-acetylated acidic dipeptidase 2
gene 10	19	70	3.68	1.88	0.011	ZBTB3 3	zinc finger and BTB domain containing 33

arrange

```
arrange(df, column_name)
```

```
arrange(df, desc(column_name) )
```

```
arrange(df, -column_name)
```

```
new_df <- df %>%
  arrange(column_name )
```

```
new_df <- df %>%
  filter(log2FC > 1 | log2FC < -1) %>%
  filter(pvalue < 0.05) %>%
  merge(y=df2, by= "gene", all.x=TRUE)
```

> new_df

gene	RPKM control	RPKM treatment	fold change	log2FC	pvalue	symbol	description
gene 5	32	10	0.31	-1.68	0.02	GAGE1 2F	G antigen 12F
gene 7	12	1	0.08	-3.58	0.001	NAALA D2	N-acetylated acidic dipeptidase 2
gene 10	19	70	3.68	1.88	0.011	ZBTB3 3	zinc finger and BTB domain containing 33

arrange

```
arrange(df, column_name)
```

```
arrange(df, desc(column_name) )
```

```
arrange(df, -column_name)
```

```
new_df <- df %>%
    arrange(column_name )
```

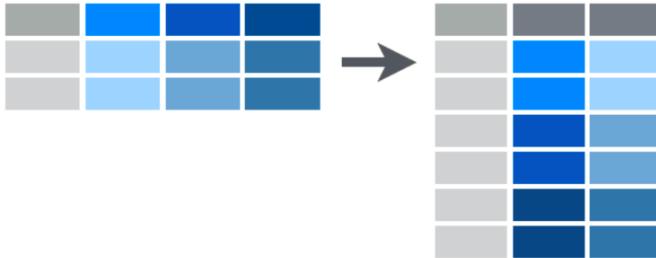
```
new_df <- df %>%
    filter(log2FC > 1 | log2FC < -1) %>%
    filter(pvalue < 0.05) %>%
    merge(y=df2, by= "gene", all.x=TRUE) %>%
    arrange(pvalue)
```

> new_df

gene	RPKM control	RPKM treatment	fold change	log2FC	pvalue	symbol	description
gene 7	12	1	0.08	-3.58	0.001	NAALA D2	N-acetylated acidic dipeptidase 2
gene 10	19	70	3.68	1.88	0.011	ZBTB3 3	zinc finger and BTB domain containing 33
gene 5	32	10	0.31	-1.68	0.02	GAGE1 2F	G antigen 12F

gather (melt)

Wide form → long form



```
new_df <- df %>%
  gather(key = <column_name>,
         value = <column_name>,
         <column_range>)
```

```
df_long <- df_wide %>%
  gather(key= "group",
         value = "observation", 2:3)
```

> df_long

replicate	group	observation
1	control	12
2	control	11
3	control	14
1	treamtment	29
2	treamtment	21
3	treamtment	27

> df_wide

replicate	control	treamtment
1	12	29
2	11	21
3	14	27

spread (pivot)

Long form → wide form



```
df_wide <- df_long %>%  
  spread(key = <column_name>,  
         value = <column_name>)
```

> df_long

replicate	group	observation
1	control	12
2	control	11
3	control	14
1	treamtment	29
2	treamtment	21
3	treamtment	27

```
df_wide <- df_long %>%  
  spread(key = "group",  
         value = "observation")
```

> df_wide

replicate	control	treamtment
1	12	29
2	11	21
3	14	27

group_by / summarize

```
new_df <- df %>%
  group_by(column1, column2, ...) %>%
  summarize(<formulas>)
```

> df_long

replicate	group	observation	
1	control	12	
2		11	
3		14	
1	treatment	29	
2		21	
3		27	

```
new_df <- df_long %>%
  group_by(group) %>%
  summarize(n=n(),
            mean = mean(observation),
            sd = sd(observation) )
```

> new_df

group	n	mean	sd
control	3	12.33333	1.527525
treatment	3	25.66667	4.163332

Data type conversions

replicate	group	observation
1	control	12
2	control	11
3	control	14
1	treatment	29
2	treatment	21
3	treatment	27

=

replicate	group	observation
<int>	<chr>	<int>
1	1 control	12
2	2 control	11
3	3 control	14
4	1 treatment	29
5	2 treatment	21
6	3 treatment	27

```
# to convert data types:  
df$column <- as.<type>(df$column)
```

```
#options:  
as.character(df$column)  
as.factor(df$column);  
  factor(df$column, levels=c(...))  
as.integer(df$column)  
as.double(df$column)
```

```
# Examples  
df$group <- as.factor(df$group)
```

replicate	group	observation
<int>	<fctr>	<int>
1	1 control	12
2	2 control	11
3	3 control	14
4	1 treatment	29
5	2 treatment	21
6	3 treatment	27

Importing/Exporting Data

```
# Importing dataframe from text file  
df <- read.delim( <path_to_data>, sep = <delimiter> )
```

```
# Exporting dataframe to text file  
write.table(df, <path_to_destination>, sep = <delimiter>, row.name=FALSE)
```

```
# Example  
df <- read.delim("./project/DGE.tsv", sep = "\t")  
  
new_df <- df %>% filter( log2FC > 1 | log2FC < -1) & pvalue < 0.05  
  
write.table(new_df, "./DGE_analysis.csv", sep = ",", row.name = FALSE)
```

Practical

R datasets:

- Iris
- ToothGrowth

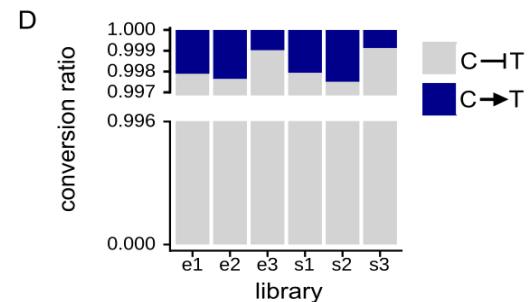
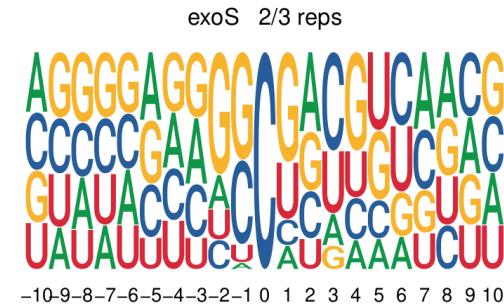
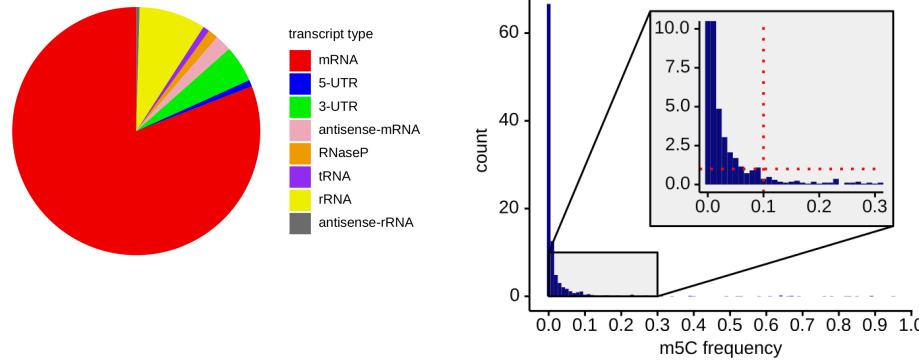
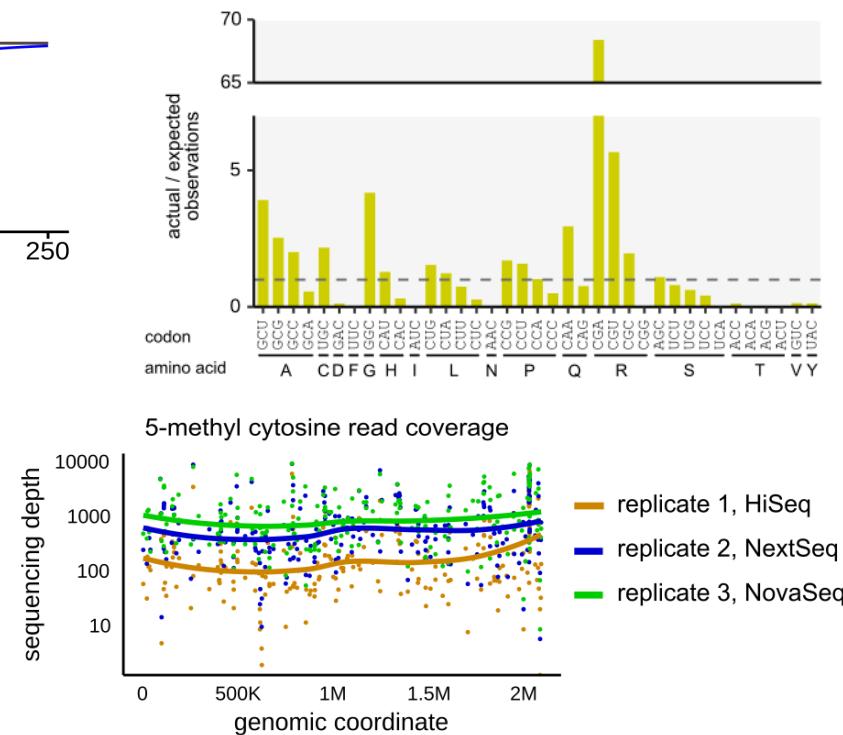
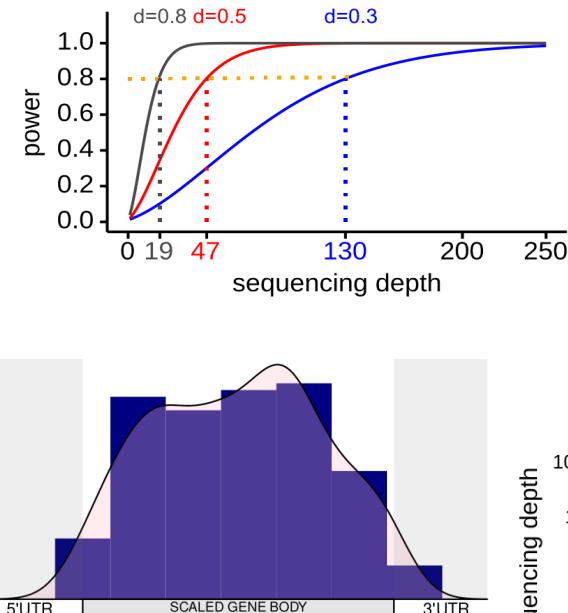
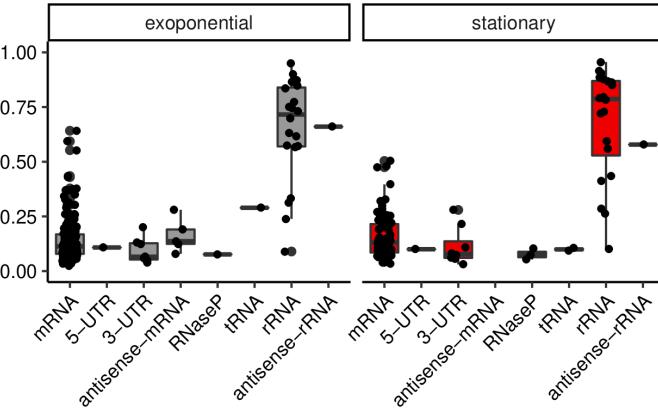
Import data:

- differential gene expression
- control_vs_treatment

Data Visualization with

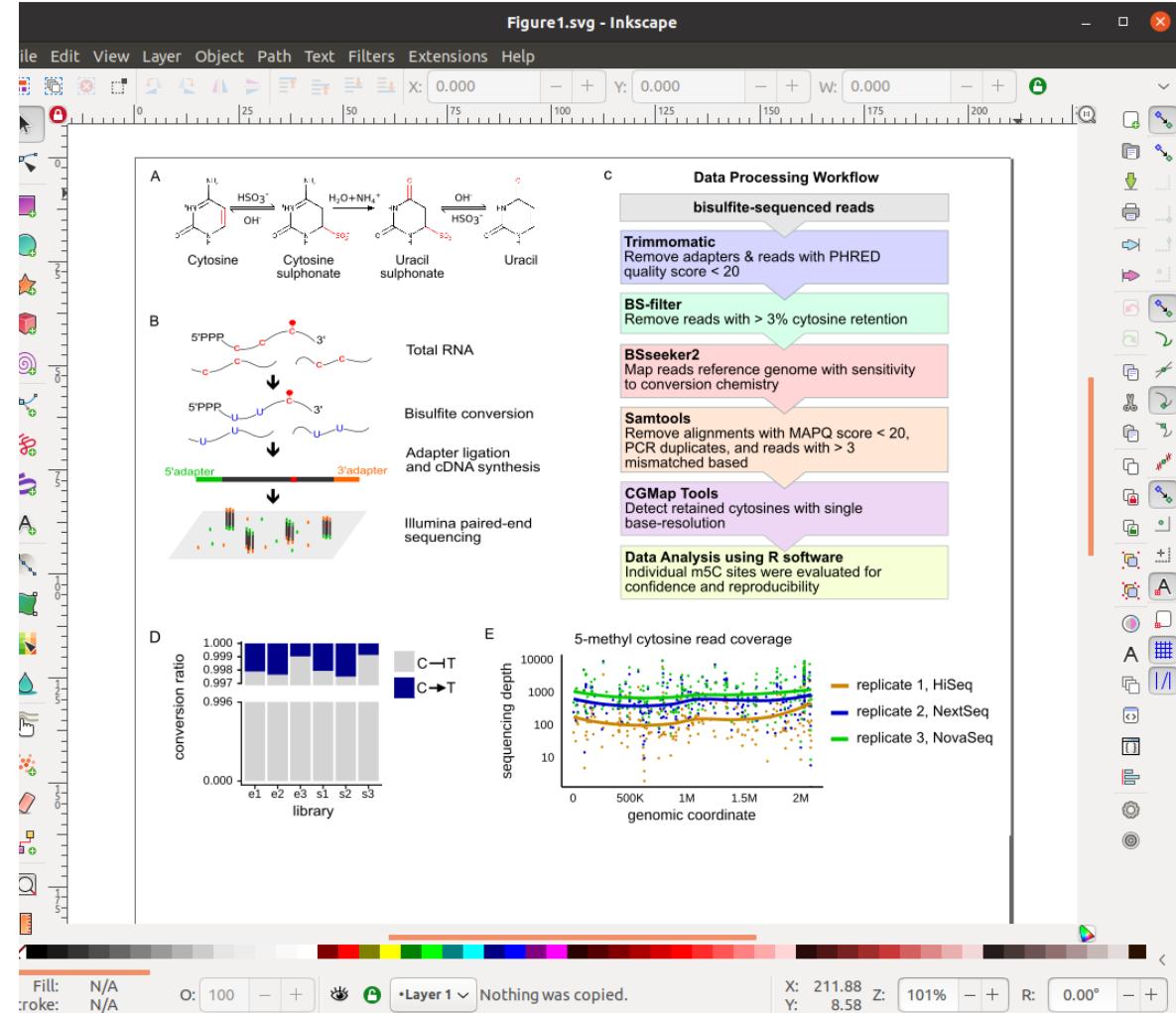


Some of my own ggplots





INKSCAPE
Draw Freely.



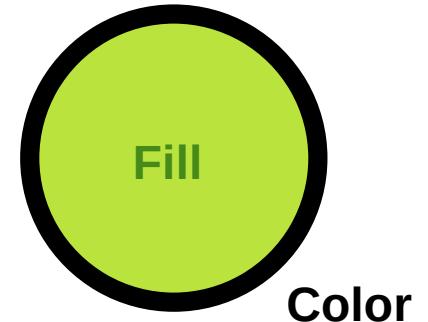
ggplot2

- Building graphs layer-by-layer (**piping operator: +**)
- **Input:** dataframe (tidy or long format)
- **Output:** ggplot

```
ggplot(data, aes(x=..., y=..., color = ..., fill = ...)) +
  geom_point() +
  theme_classic() +
  labs(x = "...", y = "...", title = "...")
  theme(<args>)
```

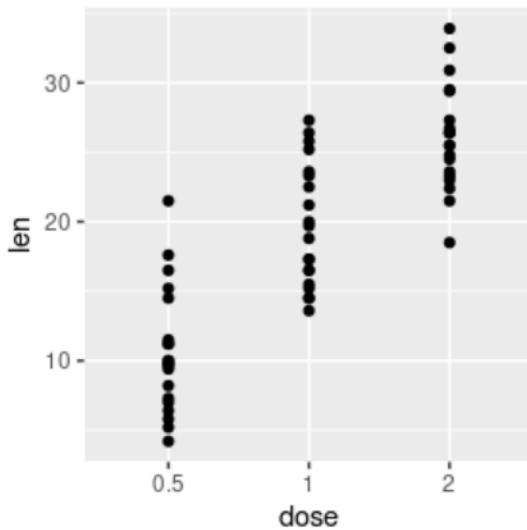
```
ggplot(df, aes(x=..., y=..., color = ..., fill = ...))
```

```
ggplot(data = df,  
       mapping = aes(x = col_name,  
                      y = col_name,  
                      color = col_name,  
                      fill = col_name)  
)
```



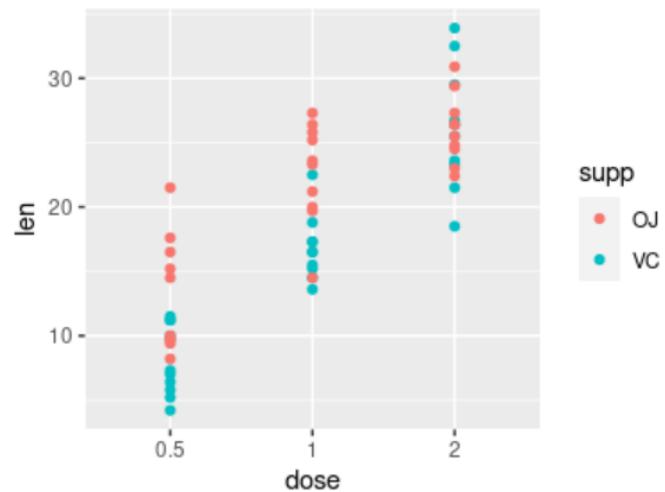
No color assignment

```
ggplot(ToothGrowth_df, aes( x = dose, y = len)) +  
  geom_point()  
...
```



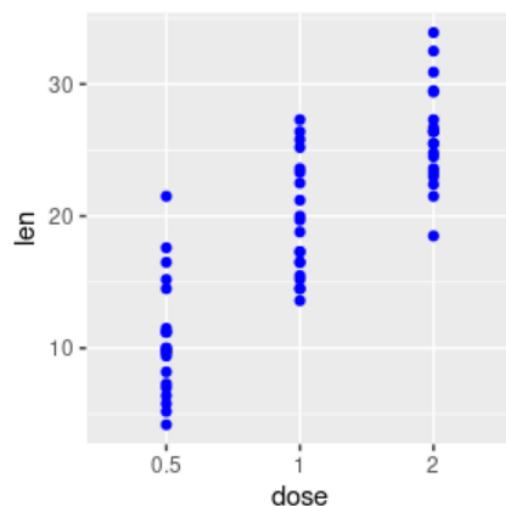
Color by group:
Color assigned in **aes()**

```
ggplot(ToothGrowth_df, aes( x = dose, y = len, color = supp)) +  
  geom_point()  
...
```



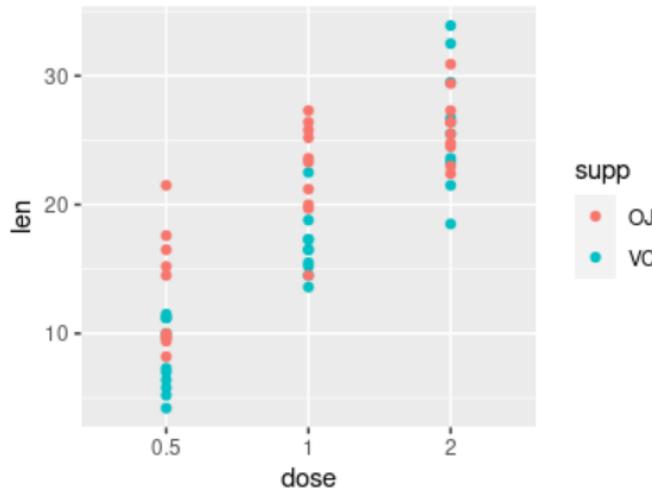
Color all data points:
Color assigned in **geom()**

```
ggplot(ToothGrowth_df, aes( x = dose, y = len)) +  
  geom_point( color = "blue")  
...
```



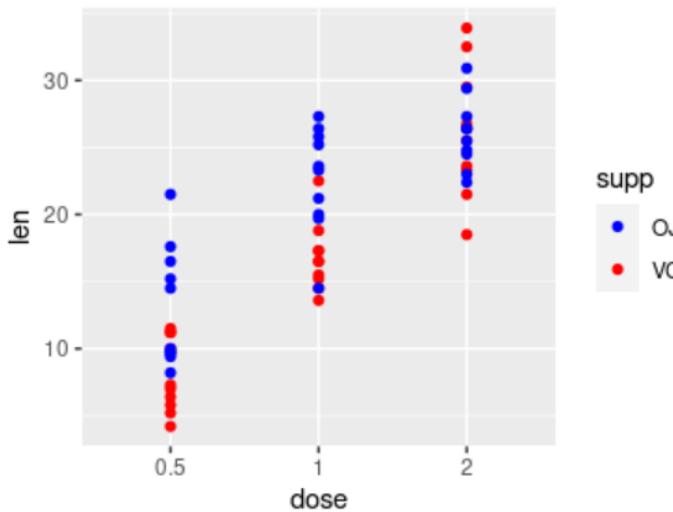
Color assigned in aes()

```
ggplot(ToothGrowth_df, aes( x = dose, y = len, color = supp)) +  
  geom_point()  
...
```



Customize colors in scale_color_manual(values = c("..."))

```
ggplot(ToothGrowth_df, aes( x = dose, y = len, color = supp)) +  
  geom_point() +  
  scale_color_manual(values = c("blue", "red"))  
...
```

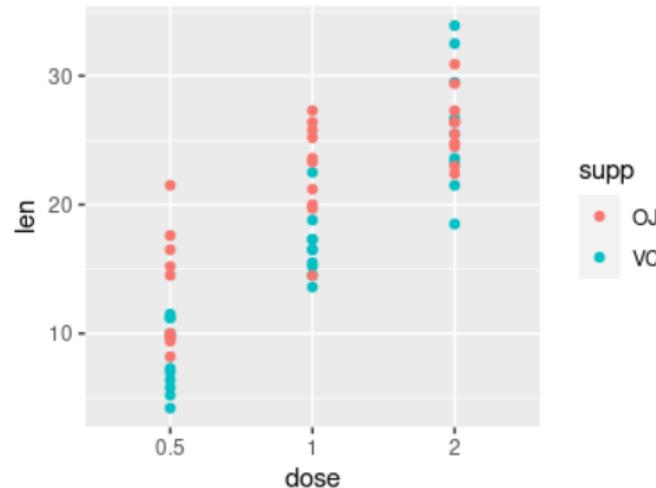


Also see **scale_fill_manual(values = c("..."))**
Number of values \geq number of unique "supp" values
Order of "supp" values = factor level

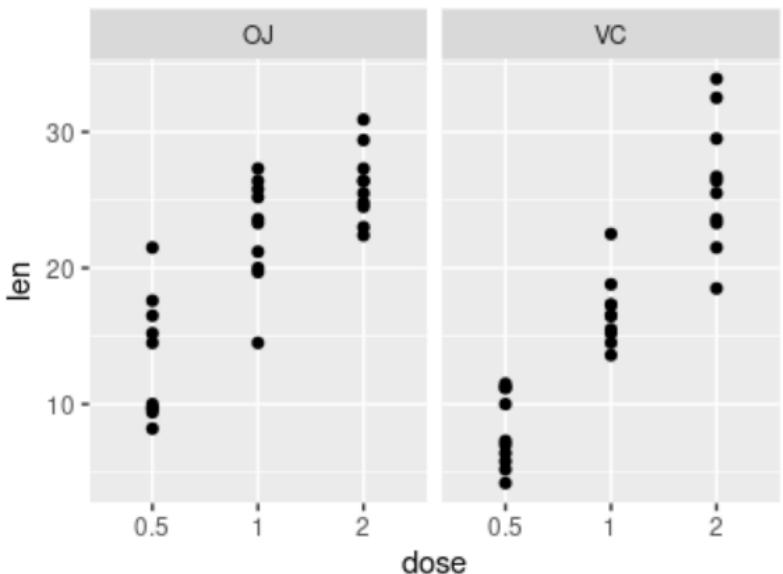
facet_wrap(~ col_name)

Color assigned in aes()

```
ggplot(ToothGrowth_df, aes(x = dose, y = len, color = supp)) +  
  geom_point()  
...
```



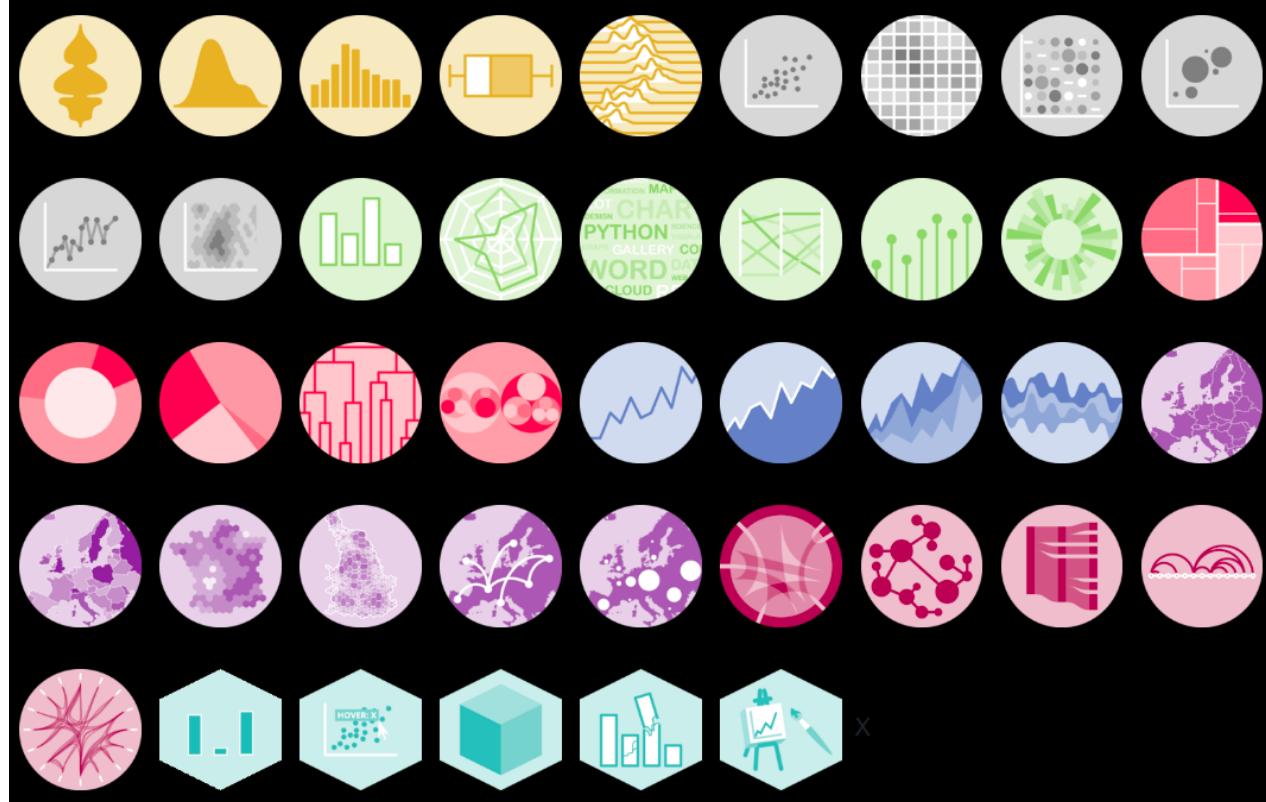
```
ggplot(ToothGrowth_df, aes(x = dose, y = len)) +  
  geom_point() +  
  facet_wrap(~ supp)
```



Also see **facet_grid(col_name ~ col_name)**

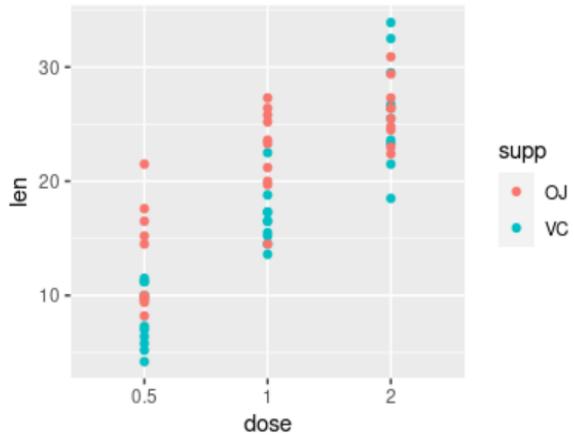
```
ggplot(df, aes(x=..., y=..., color = ..., fill = ...)) +  
  geom_point()
```

[https://www.r-graph-
gallery.com/](https://www.r-graph-gallery.com/)



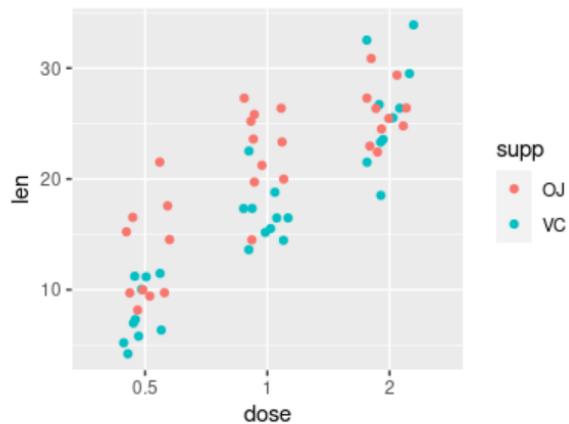
geom_point()

```
ggplot(ToothGrowth_df, aes( x = dose, y = len, color = supp)) +  
  geom_point()  
...
```



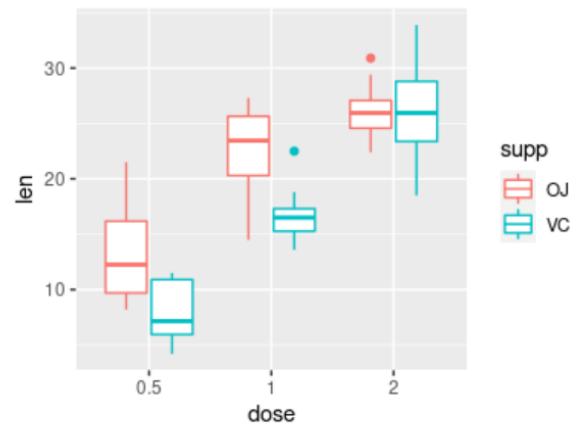
geom_jitter()

```
ggplot(ToothGrowth_df, aes( x = dose, y = len, color = supp)) +  
  geom_jitter(position = position_jitter(.2))  
...
```

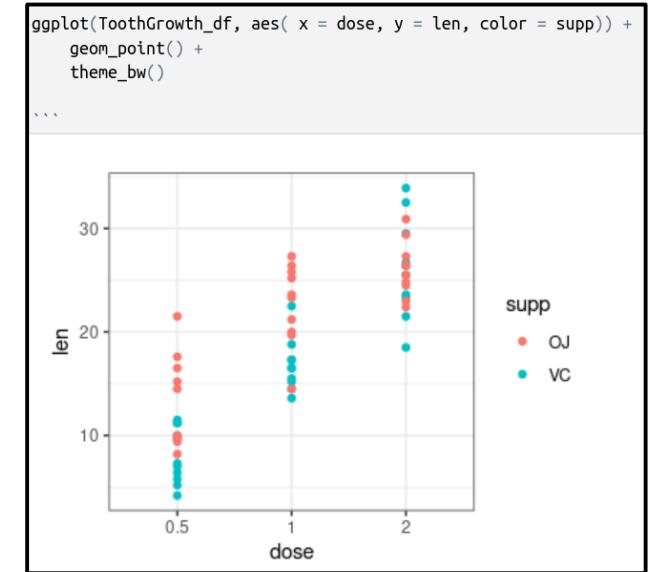
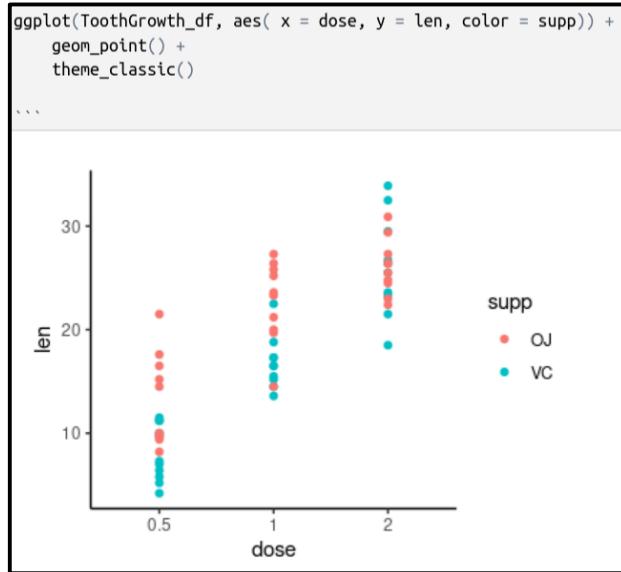


geom_boxplot()

```
ggplot(ToothGrowth_df, aes( x = dose, y = len, color = supp)) +  
  geom_boxplot()  
...
```



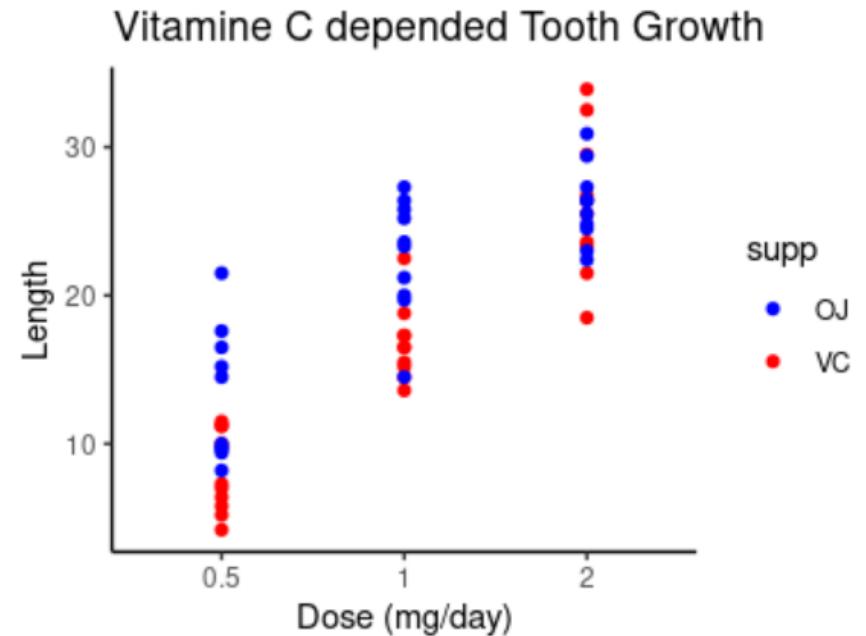
```
ggplot(df, aes(x=..., y=..., color = ..., fill = ...)) +  
  geom_point() +  
  theme_classic()
```



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

```
ggplot(df, aes(x=..., y=..., color = ..., fill = ...)) +  
  geom_point() +  
  theme_classic() +  
  labs(x = "...", y = "...", title = "...")
```

```
labs(x="Dose (mg/day)",  
     y = "Length",  
     title = "Vitamine C depended Tooth Growth")
```



```
ggplot(data, aes(x=..., y=..., color = ..., fill = ...)) +
  geom_point() +
  theme_classic() +
  labs(x = “...”, y = “...” , title = “...”) +
  theme(<args>)
```

All text on graph can be customized using **theme()**

```
theme(  
  plot.title = element_text(hjust = 0.5, size = 14),  
  axis.text = element_text(size = 10, color="black"),  
  axis.ticks = element_line(color="black" ),  
  axis.title = element_text( size = 10, color="black")  
)
```

axis customization

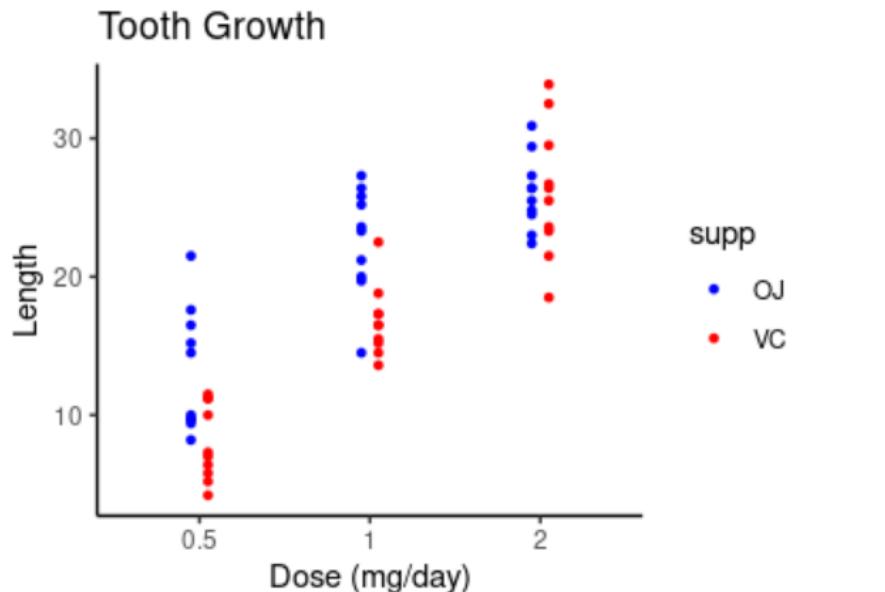
```
theme(  
  axis.text.x = element_text(size = 10, color="black"),  
  axis.text.y = element_text(size = 10, color="black"),  
  axis.title.x = element_text( size = 10, color="black"),  
  axis.title.y = element_text( size = 10, color="black")  
)
```

legend customization

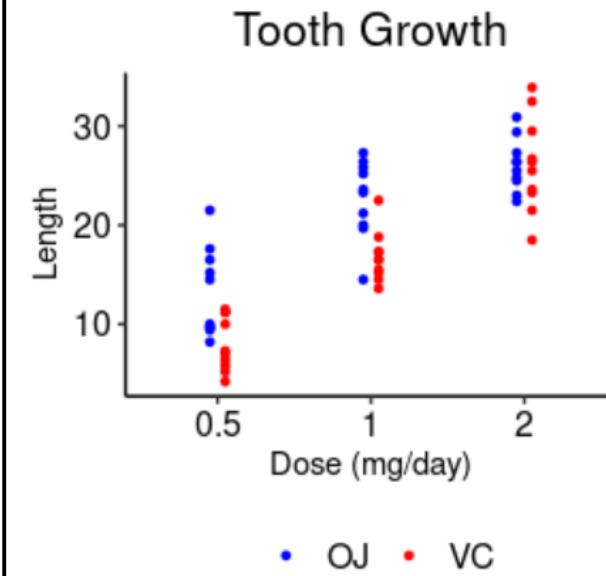
<http://www.sthda.com/english/wiki/ggplot2-legend-easy-steps-to-change-the-position-and-the-appearance-of-a-graph-legend-in-r-software>

```
theme(legend.position = c(5,5) ,  
      legend.title = element_text(size=10),  
      legend.text = element_text(size=10)  
)
```

```
ggplot(ToothGrowth_df, aes( x = dose, y = len, color = supp)) +  
  geom_jitter(size = 1, position = position_dodge(0.2)) +  
  theme_classic() +  
  scale_color_manual(values=c("blue", "red")) +  
  labs(x = "Dose (mg/day)", y = "Length",  
       title = "Tooth Growth")  
...
```



```
theme(plot.title = element_text(hjust = 0.5, color="black", size = 16),  
      axis.text = element_text(color="black", size=12),  
      legend.position = "bottom",  
      legend.title = element_blank(),  
      legend.text = element_text(size=12)  
)  
...
```



Save your graph

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

```
ggsave(plot = <plot>,    #default is last ggplot  
       filename = "plot.pdf"  
       path = "./path/to/destination/",  
       device = "pdf",  
       width = <inches>,  
       height = <inches>,  
       dpi = <dpi>    #quality, dots per inch
```

Save your graph

```
ggsave("plot.pdf",  
       path = "./figures",  
       device = "pdf",  
       width = 4,  
       height = 3,  
       dpi = 200
```

Other useful options:

- Change axis break points and limits

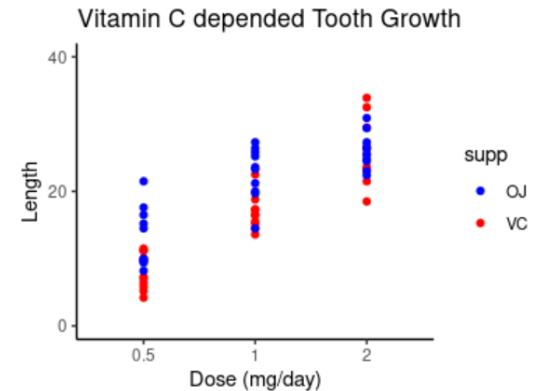
```
scale_x_continuous(limits=c(0,100), breaks = (0,25,50,100) )
```

Also see **scale_y_continuous()**

The x axis will have limits with a lower bound of 0 and upper bound of 100

The x axis tick marks will include 0, 25, 50, 75, and 100.

```
scale_y_continuous(limits = c(0,40), breaks =c(0,20,40))
```



- Add text, shapes and lines to graphs

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

```
annotate("text", x = 1, y = 25, label = "Some text")
```

Will add text at coordinate positions 1,25 (x,y coordinates) that will say "Some text"

```
annotate("text", x = 1, y = 25, label = "Some text")
```

