Advanced Topics in R

February 15, 2023

Outline

- **Morning**
  - Data manipulation/Data cleaning
  - ggplot2
- **Afternoon**
  - ggplot2 (continued)
  - Statistical tests
  - Regression

Confirm you have these packages installed in your R studio:

ggplot2
doBy
reshape2

To check:

1) Open R studio
2) Run the commands:

```
library(ggplot2)
library(doBy)
library(reshape2)
```

3) If no errors, then you're OK

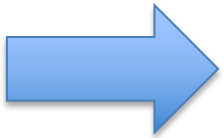
cribioinfo@uic.edu

Data Manipulation

Data manipulation/Data cleaning for complex data

"Messy" data

Patient ID	Cohort	Age	Gender	Disease	Genotype
TMA-26MS-1-KP	UIC	82.5	F	1	Mutant
TMA-26MS-2-KP	UIC	55.6	M	1	WT
TMA-53ML-3-KP	UIC		Male	0	MT
TMA-53ML-4-KP	Uchicago	73.7	F	.	
TMA-33MJ-5-KP	UIC	73.1	F		
TMA-33MJ-6-KP	UIC	55.2	F	.	Wildtype
TMA-53LV-7-KP	UIC	55.9	Female	1	Wild type
TMA-53LV-8-KP	UIC	68.6	M	1	mt
TMA-3NC-9-KP	Uchicago	46.2	F	0	Mutant



"Cleaned" data

Patient ID	Cohort	Age	Gender	Disease	Genotype
TMA.26MS.1	UIC	82.5	F	1	MT
TMA.26MS.2	UIC	55.6	M	1	WT
TMA.53ML.3	UIC	64.3	M	0	MT
TMA.53ML.4	Uchicago	73.7	F		
TMA.33MJ.5	UIC	73.1	F		
TMA.33MJ.6	UIC	55.2	F		WT
TMA.53LV.7	UIC	55.9	F	1	WT
TMA.53LV.8	UIC	68.6	M	1	MT
TMA.3NC.9	Uchicago	46.2	F	0	MT

https://uic-ric.github.io/workshop-data/advanced_R/Messy_data.xlsx

What to fix:

(1) Patient ID: remove "-KP" at the end; replace "-" with "."

(2) Age: Change missing values with the average age in all patients.

(3) Gender: only "F", "M" are allowed

(4) Disease: change "." or "" to NA (missing value)

(5) Genotype: only "MT", "WT", or NA (missing value) are allowed

https://github.com/uic-ric/uic-ric.github.io/wiki/Providing-sample-information

Data manipulation – apply functions

- Apply a function to various parts of a matrix, data frame, vector or list.
  - "apply" functions are like a loop
  - Fewer lines in coding; faster running speed

- apply() - operate on selected dimensions of arrays

  apply() will execute a function on every row or every column of a matrix

- lapply(), sapply() - operate on each element of a vector or list
- tapply() - operate on elements based on a grouping variable
- mapply() - multivariate extension of sapply

“apply” function

- apply(X, MARGIN, FUN, ...)
  - X is the matrix
  - MARGIN indicates by row (1) or by column (2)
  - FUN is the function to be applied, could be name of defined function or function code

e.g. `apply(gene.exp, 1, mean)`, this will calculate the average expression for each gene (row)

cribioinfo@uic.edu

Comparison with “for” loop

- “apply” function

```
> mean.gene.exp <- apply(gene.exp, 1, mean)
```

- For loop

```
> N.genes <- nrow(gene.exp)
> mean.gene.exp <- vector()
> for (i in 1:N.genes)
+ {
+ mean.gene.exp[i] <- mean(as.numeric(gene.exp[i, ]))
+ }
>
```

- Apply is more compact, and executes *much* faster in R

Exercise 1.1: apply function vs for loop

Test the speed of apply function and for loop

Tools to combine datasets

- Combine datasets
  - rbind() , cbind() - return a new dataset with added rows or columns.

- Merged datasets
  - merge() - merge two data frames (datasets) horizontally, in most cases, join two data frames by one or more common key(s).

cribioinfo@uic.edu

Data combination: rbind

- We have two tables for clinical data. In the table, each row represents one patient. We want to combine the patient clinical data row by row.
  - Number of columns must match
  - R *may not* check if the column names match (depends on the class of the objects)

cohort1

ID	Age	Sex	Race
1	55	F	Asian
2	50	M	Black
3	62	M	White
4	52	F	Black

cohort2

ID	Age	Sex	Race
5	50	F	White
6	42	M	White
7	63	F	White
8	52	F	White
9	70	F	Black
10	59	M	Black

cohorts

ID	Age	Sex	Race
1	55	F	Asian
2	50	M	Black
3	62	M	White
4	52	F	Black
5	50	F	White
6	42	M	White
7	63	F	White
8	52	F	White
9	70	F	Black
10	59	M	Black

```
> cohorts <- rbind(cohort1, cohort2)
```

Data combination: cbind

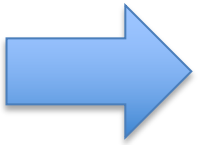
- We have two tables for clinical data. We want to combine the patient clinical data column by column.
  - Number of rows must match
  - R *will* not check if the row names match: **The subject IDs must be in the same order in the two tables.**

df.part1

ID	Age	Sex
1	55	F
2	50	M
3	62	M
4	52	F
5	50	F
6	42	M

df.part2

ID	Race	Vital.Status
1	Asian	Alive
2	Black	Alive
3	White	Dead
4	Black	Alive
5	White	Dead
6	White	Alive



df.all

ID	Age	Sex	Race	Vital.Status
1	55	F	Asian	Alive
2	50	M	Black	Alive
3	62	M	White	Dead
4	52	F	Black	Alive
5	50	F	White	Dead
6	42	M	White	Alive

```
> df.all <- cbind(df.part1, df.part2[, 2:3])
```


cribioinfo@uic.edu

Merge - intersection

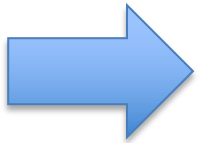
- Merge two datasets (intersection): only the rows that matched are included.
- The subject IDs don't need to be in the same order.

df.h

ID	Sex	Height
1	F	4
2	M	3.5
3	M	4.7
4	F	7.7
5	F	5

df.w

ID	Sex	Weight
6	M	2.1
9	F	0.8
4	F	1.1
5	F	2.1
2	M	1.5
7	F	2.6
8	F	1.5
3	M	5.4
10	M	1.9



df.hw

ID	Sex.x	Height	Sex.y	Weight
2	M	3.5	M	1.5
3	M	4.7	M	5.4
4	F	7.7	F	1.1
5	F	5	F	2.1

```
> df.hw <- merge(df.h, df.w, by="ID")
```

Merge - union

Merge two datasets (union): all the rows are included

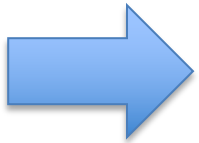
```
> df.hw <- merge(df.h, df.w, by="ID", all=T)
```

df.h

ID	Sex	Height
1	F	4
2	M	3.5
3	M	4.7
4	F	7.7
5	F	5

df.w

ID	Sex	Weight
6	M	2.1
9	F	0.8
4	F	1.1
5	F	2.1
2	M	1.5
7	F	2.6
8	F	1.5
3	M	5.4
10	M	1.9



df.hw

ID	Sex.x	Height	Sex.y	Weight
1	F	4	<NA>	NA
2	M	3.5	M	1.5
3	M	4.7	M	5.4
4	F	7.7	F	1.1
5	F	5	F	2.1
6	<NA>	NA	M	2.1
7	<NA>	NA	F	2.6
8	<NA>	NA	F	1.5
9	<NA>	NA	F	0.8
10	<NA>	NA	M	1.9

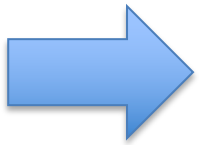
A warning on merge()

- merge() will make a new row for all possible matches

ID	Age	Sex
1	55	F
2	50	M
3	62	M
4	52	F

ID	Age	Sex
1	55	F
2	50	M
3	62	M
4	52	F

merge by "Sex"



Sex	ID.x	Age.x	ID.y	Age.y
F	1	55	1	55
F	1	55	4	52
F	4	52	1	55
F	4	52	4	52
M	2	50	2	50
M	2	50	3	62
M	3	62	2	50
M	3	62	3	62

- If you merge on a column with many possible matches, you can end up with a result that is much larger than either starting table!

Add a new column to a data frame

Create a new column in an existing data frame based on existing values

```
> patients$bmi <- patients$weight/patients$height^2
```

Order of precedence for mathematical operations (^ before /)
https://stat.ethz.ch/R-manual/R-devel/library/base/html/Syntax.html

```
> patients$bmi <- patients$weight/(patients$height^2)
```

Create age category based on numeric age number

```
> patients$agecat[patients$age > 75] <- "Elder"
> patients$agecat[patients$age > 45 & patients$age <= 75] <- "Middle Aged"
> patients$agecat[patients$age <= 45] <- "Young"
> patients$agecat <- as.factor(patients$agecat)
```

Use cut to create age category based on numeric age number

```
> patients$agecat <- cut(patients$age, breaks=c(0,45,75,Inf),
    labels=c("Young","Middle Aged","Elder"))
```

Data subsetting

Suppose we have a patient clinical data. We want to select old female patients

- We can subset a data based on [] selection

```
> patients[patients$gender == "F" & patients$age >= 60, ]
```

- Use subset function: no need to type data frame name repeatedly

```
> subset(patients, gender == "F" & age >=60 )
```

For a gene expression matrix, only keep those genes with average expression > 5 across all samples

- Use apply function in subset

```
> subset(gene.exp, apply(gene.exp, 1, mean) > 5)
```

Exercise 1.2: combining and sub-setting data

(1) Use cbind/rbind/merge functions to operate on tables

(2) Add a new column

(3) Use subset function to select young female patients

cribioinfo@uic.edu

String manipulation: sub

Find and **sub**stitue string for the first match:

- `sub(pattern, replacement, x, ignore.case = FALSE, perl = FALSE, fixed = FALSE, useBytes = FALSE)`
  - pattern: string to be matched as a regular expression (or character string for fixed = TRUE)
  - replacement: string for replacement for matched pattern
  - x: string or string vector

- Substitute string for the first match: sub

```
> input.file <- "gene.exp.txt"
> output.file <- sub(".txt", ".heatmap.pdf", input.file, fixed=T)
> output.file
[1] "gene.exp.heatmap.pdf"
```

Regular expression

Regular expression: a sequence of characters that define a search pattern

Expression	
[]	Character class. match one or more character(s) within the bracket
[0-9]	matches any of the following 0,1,2,3,4,5,6,7,8,9
[ATCG]	matches ANY character from “ATCG”
[a-z]	match any a to z character
^	match at the beginning of string
$	match at the end of string
+	match 1 or more times
?	match 1 or 0 times
*	match 0 or more times
.	ANY character
\\	Escape, e.g. “\\.” will literally be character “.”

String manipulation: sub and gsub

sub only replaces the first match, **gsub** replaces all matches.

The fields of each sample name are delimited with ".", and we want to replace it with "-" for all occurrences.

- Substitute string for all matches: gsub

```
> head(sampleIDs)
[1] "TCGA.D3.A3MR.06A.11R.A21D.07" "TCGA.D3.A8GI.06A.11R.A37K.07"
[3] "TCGA.D9.A3Z1.06A.11R.A239.07" "TCGA.EE.A2GM.06B.11R.A18S.07"
[5] "TCGA.QB.A6FS.06A.11R.A311.07" "TCGA.D3.A2JH.06A.11R.A18T.07"
> fixed.sampleIDs <- gsub(".", "-", sampleIDs, fixed=T)
> head(fixed.sampleIDs)
[1] "TCGA-D3-A3MR-06A-11R-A21D-07" "TCGA-D3-A8GI-06A-11R-A37K-07"
[3] "TCGA-D9-A3Z1-06A-11R-A239-07" "TCGA-EE-A2GM-06B-11R-A18S-07"
[5] "TCGA-QB-A6FS-06A-11R-A311-07" "TCGA-D3-A2JH-06A-11R-A18T-07"
```

String manipulation: grep

grep (globally search a regular expression and print)

```
grep(pattern, x, ignore.case = FALSE, perl = FALSE, value = FALSE, fixed = FALSE, …)
```

- pattern: string to be matched as a regular expression (or character string for fixed = TRUE)
- x: string or string vector
- value: if set to TRUE will return matching values. Default is to return index of matching values.

Find all samples with sample name containing "06A"

```
> head(sampleIDs)
[1] "TCGA.D3.A3MR.06A.11R.A21D.07" "TCGA.D3.A8GI.06A.11R.A37K.07"
[3] "TCGA.D9.A3Z1.06A.11R.A239.07" "TCGA.EE.A2GM.06B.11R.A18S.07"
> sampleIDs[grep("06A", sampleIDs)]
[1] "TCGA.D3.A3MR.06A.11R.A21D.07" "TCGA.D3.A8GI.06A.11R.A37K.07"
[3] "TCGA.D9.A3Z1.06A.11R.A239.07"
> grep("06A", sampleIDs, value=T)
[1] "TCGA.D3.A3MR.06A.11R.A21D.07" "TCGA.D3.A8GI.06A.11R.A37K.07"
[3] "TCGA.D9.A3Z1.06A.11R.A239.07"
```

Exercise 1.3: String substitution and grep

sub only replaces the first match

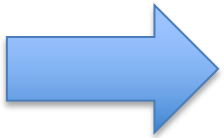
gsub replaces all matches

grep globally search a regular expression and print

Data manipulation/Data cleaning for complex data

"Messy" data

Patient ID	Cohort	Age	Gender	Disease	Genotype
TMA-26MS-1-KP	UIC	82.5	F	1	Mutant
TMA-26MS-2-KP	UIC	55.6	M	1	WT
TMA-53ML-3-KP	UIC		Male	0	MT
TMA-53ML-4-KP	Uchicago	73.7	F	.	
TMA-33MJ-5-KP	UIC	73.1	F		
TMA-33MJ-6-KP	UIC	55.2	F	.	Wildtype
TMA-53LV-7-KP	UIC	55.9	Female	1	Wild type
TMA-53LV-8-KP	UIC	68.6	M	1	mt
TMA-3NC-9-KP	Uchicago	46.2	F	0	Mutant



"Cleaned" data

Patient ID	Cohort	Age	Gender	Disease	Genotype
TMA.26MS.1	UIC	82.5	F	1	MT
TMA.26MS.2	UIC	55.6	M	1	WT
TMA.53ML.3	UIC	64.3	M	0	MT
TMA.53ML.4	Uchicago	73.7	F		
TMA.33MJ.5	UIC	73.1	F		
TMA.33MJ.6	UIC	55.2	F		WT
TMA.53LV.7	UIC	55.9	F	1	WT
TMA.53LV.8	UIC	68.6	M	1	MT
TMA.3NC.9	Uchicago	46.2	F	0	MT

What to fix:

(1) Patient ID: remove "-KP" at the end; replace "-" with "."

(2) Age: Change missing values with the average age in all patients.

(3) Gender: only "F", "M" are allowed

(4) Disease: change "." or "" to NA (missing value)

(5) Genotype: only "MT", "WT", or NA (missing value) are allowed

https://github.com/uic-ric/uic-ric.github.io/wiki/Providing-sample-information

Exercise 1.4: Clean messy tables

Convert a messy table to a cleaned data frame

BREAK

cribioinfo@uic.edu

Introduction to ggplot2

- Understand basic concepts and terminology for ggplot2 – R package
- Generate basic plots using ggplot2
  - scatter plot
  - box plot
  - histogram

Different Plotting system in R

- Base Graphics plotting system (Base R)
  - R's pre-installed system (i.e. it comes with R and no extra packages are installed).
  - Functions include plot, barplot, hist
  - Different plotting functions may have different parameters.
  - Quick and easy way to visualize data
- ggplot2 - Plots are build up in layers using **grammar of graphics**. Often compared to assembling a sandwich.

Aesthetic mapping for categorical variables

- Best options
  - Qualitatively different colors (red vs. green vs. blue)
  - Labels (monotone)
- Better options
  - Sequential colors (red vs. orange vs. yellow)
  - Shape outlines (monotone)
  - Line types (dashed, double dashed, etc.)
- Lowest efficiency
  - Different shapes (same color)
  - Hatching (monotone)
  - Line width

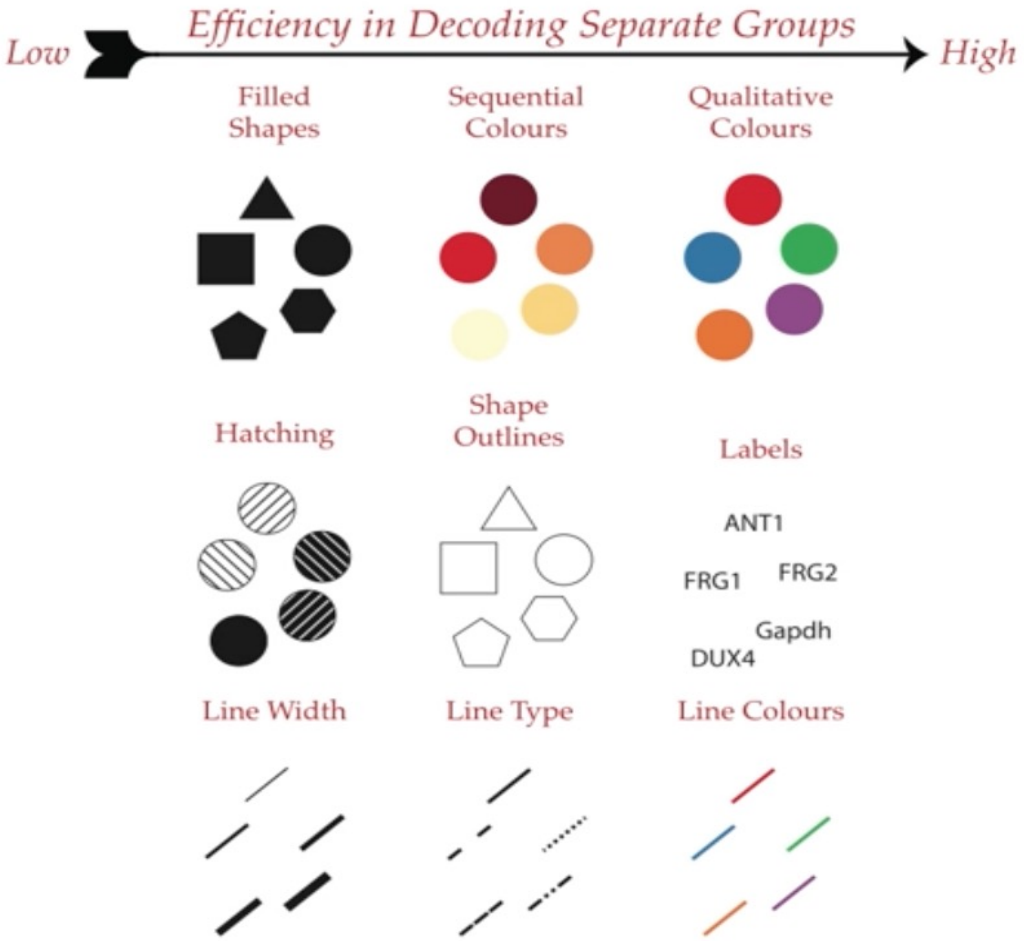


Image Source: https://rstudio-pubs-static.s3.amazonaws.com/294962_40c4eb3b399d494b9f063e05847fa933.html

cribioinfo@uic.edu

Aesthetic mappings for continuous variables

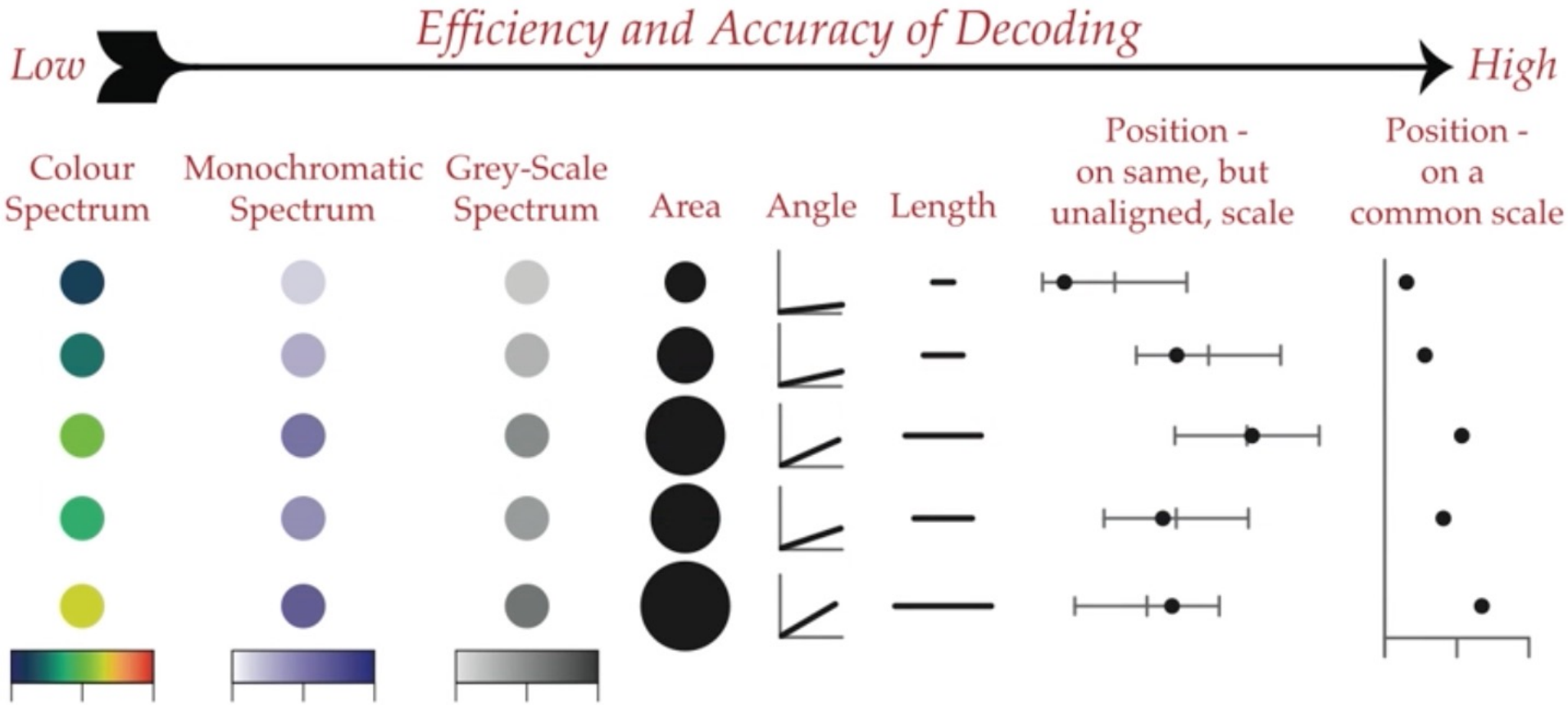


Image Source: https://rstudio-pubs-static.s3.amazonaws.com/294962_40c4eb3b399d494b9f063e05847fa933.html

cribioinfo@uic.edu

ggplot2 – grammar of graphics plot

Element	Description	Example
Data	The dataset being plotted. Data must be stored as R data frame	mtcars or any data frame with variables of interest
Aesthetics	Describes visual character tics that represents data	axes(x-axis, y-axis), color fills, labels, legends, shape, size, transparency
Geometrics	Describes type of geometric objects that represent data	points, lines, bars, areas
Facets	Describe how to break data into subsets and displayed as multiple small graphs	Columns, Rows
Statistics	Describes statistical transformation to understand the data. Data summary.	binning and counting observations to create a histogram
Coordinates	Describes how data coordinates are mapped to 2-D space.	Log transformation, reverse scale
Themes	All non-data ink	Panel background, legend background

Template of ggplot2 commands

Define data and (optional) global aesthetics (mapping)

```
ggplot(data = <DATA>, mapping = aes(<MAPPINGS>)) +
  geom_<FUNCTION>(mapping = aes(<MAPPINGS>),
                  stat = <STAT> , position = <POSITION>) +
  coord_<FUNCTION>() +
  stat_<FUNCTION>() +
  facet_<FUNCTION>() +
  scale_<FUNCTION>() +
  theme_<FUNCTION>()
```

Geometrics layer with (optional) parameters

(optional) functions to modify coordinates, statistics, facets, scales, or themes of the plot

ggplot2 elements/functions are combined (+) to create the final plot

cribioinfo@uic.edu

Example data: `mtcars` data frame

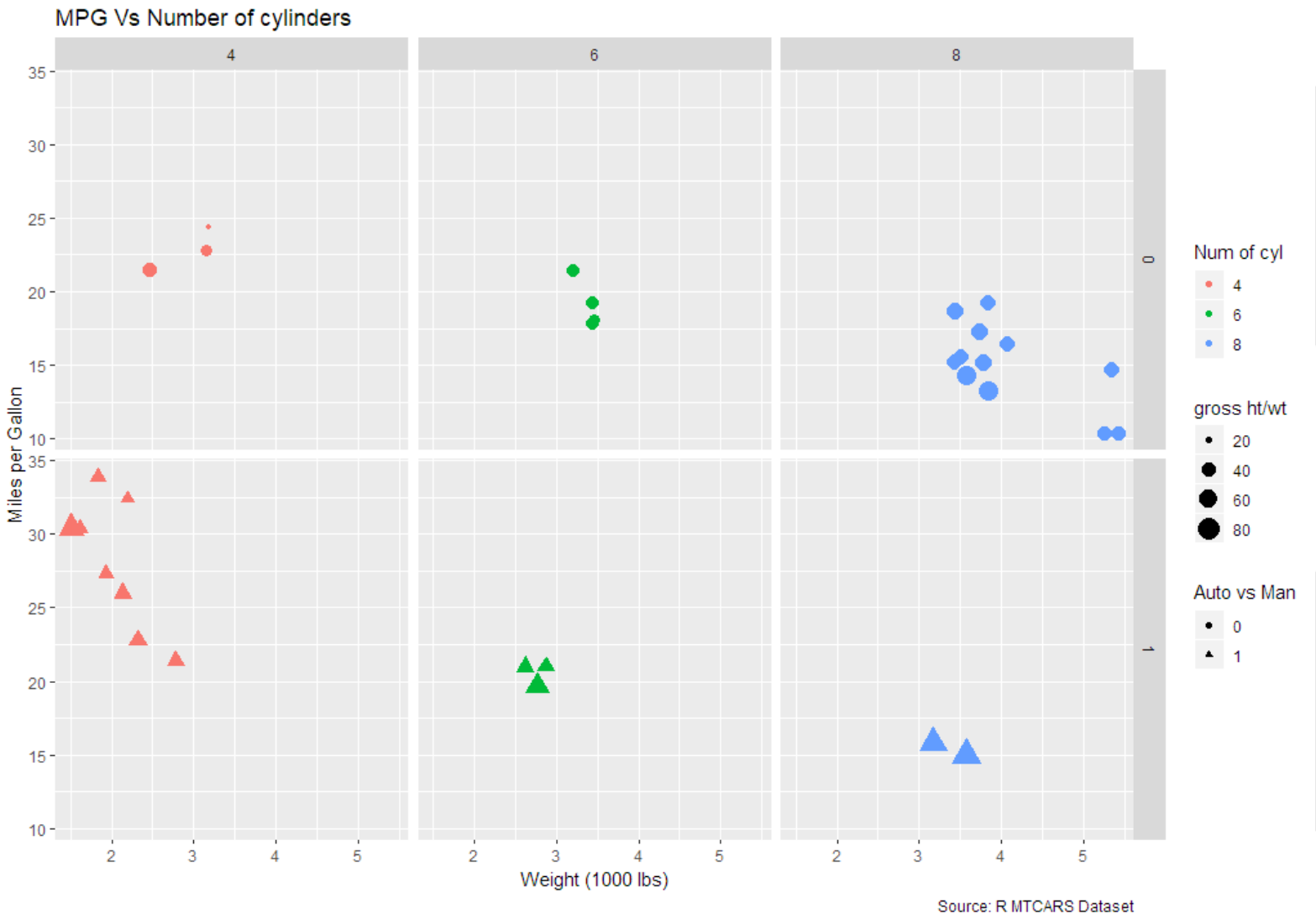
	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21	6	160	110	3.9	2.62	16.46	0	1	4	4
Mazda RX4 Wag	21	6	160	110	3.9	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.32	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.44	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.46	20.22	1	0	3	1

A data frame with 32 observations on 11 (numeric) variables.
Built in to R as a commonly used test data set.

The data was extracted from the 1974 Motor Trend US magazine and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models).

mpg	Miles/(US) gallon
cyl	Number of cylinders
disp	Displacement (cu.in.)
hp	Gross horsepower
drat	Rear axle ratio
wt	Weight (1000 lbs)
qsec	1/4 mile time
vs	Engine (0 = V-shaped, 1 = straight)
am	Transmission (0 = automatic, 1 = manual)
gear	Number of forward gears
carb	Number of carburetors

ggplot2 – grammar of graphics plot – Scatter plot

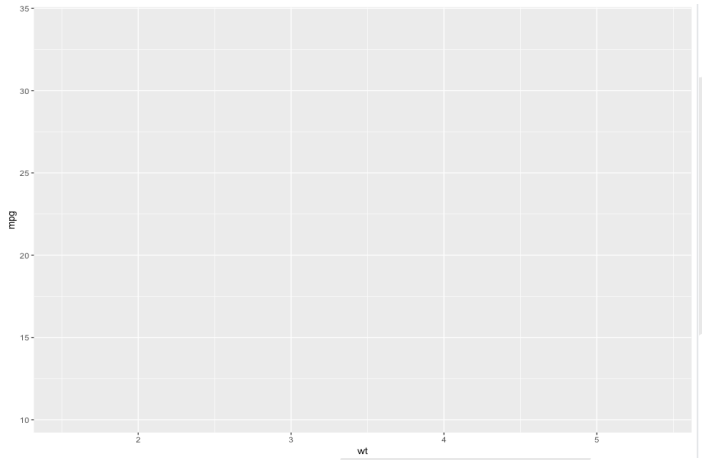


cribioinfo@uic.edu

ggplot2 – Scatter Plot

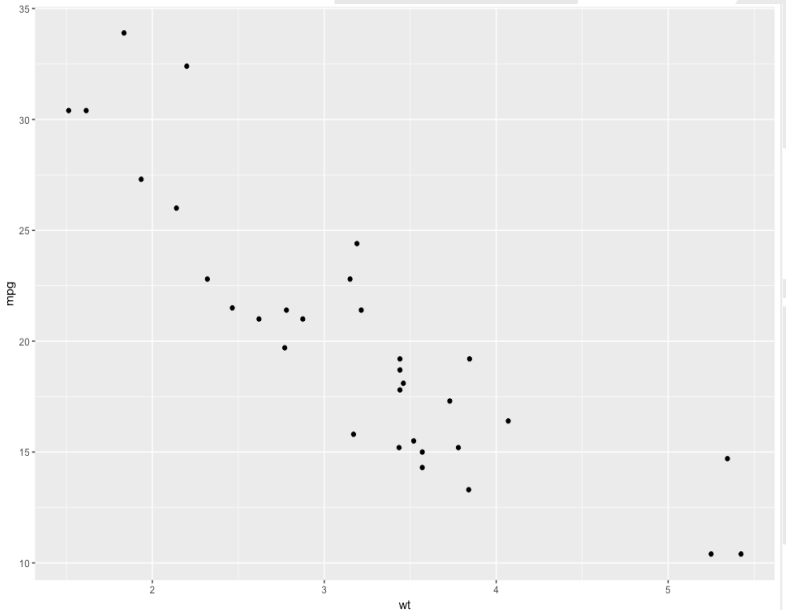
- Create a frame

```
ggplot(mtcars, aes(x = wt, y = mpg))
```



- Add a scatter plot

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

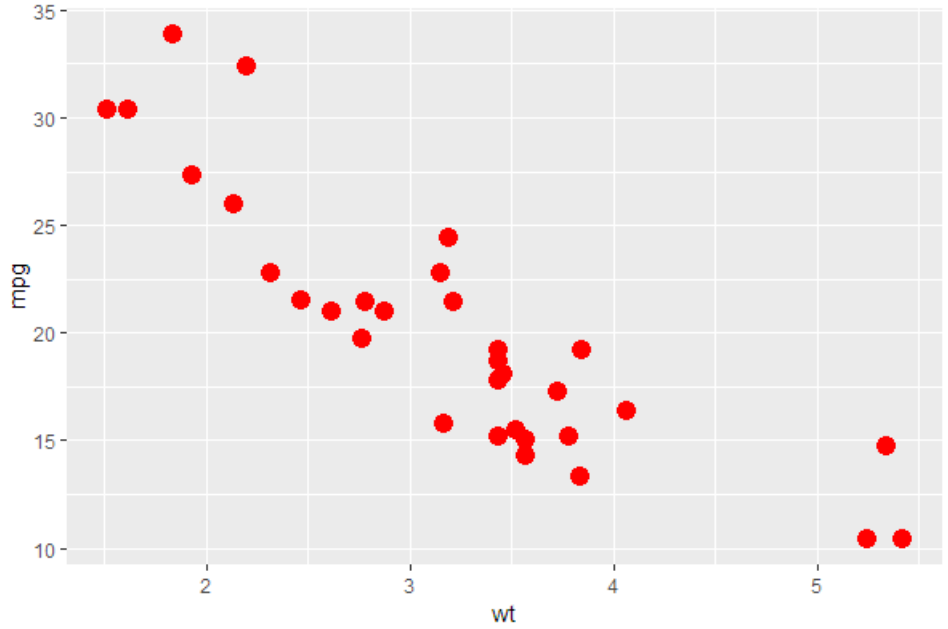


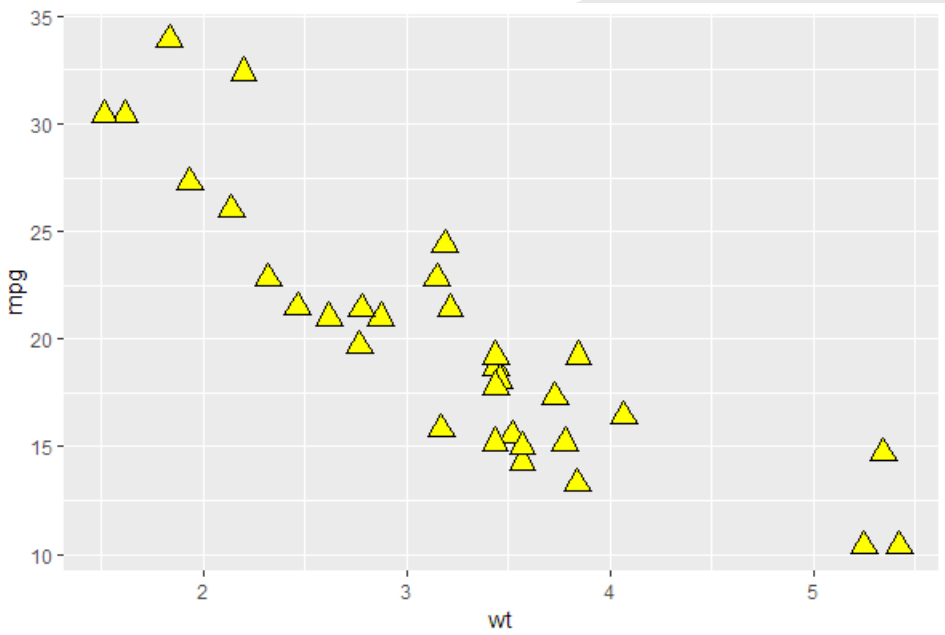
cribioinfo@uic.edu

ggplot2 – Scatter Plot – Set point size/shape/color

- Add shape (generic), size, color

```
ggplot(mtcars, aes(wt, mpg)) +
    geom_point(size=4, color="red")
```






```
ggplot(mtcars, aes(x=wt, y=mpg)) +
  geom_point(shape=24, size=4,
  fill="yellow")
```

http://www.cookbook-r.com/Graphs/Shapes_and_line_types

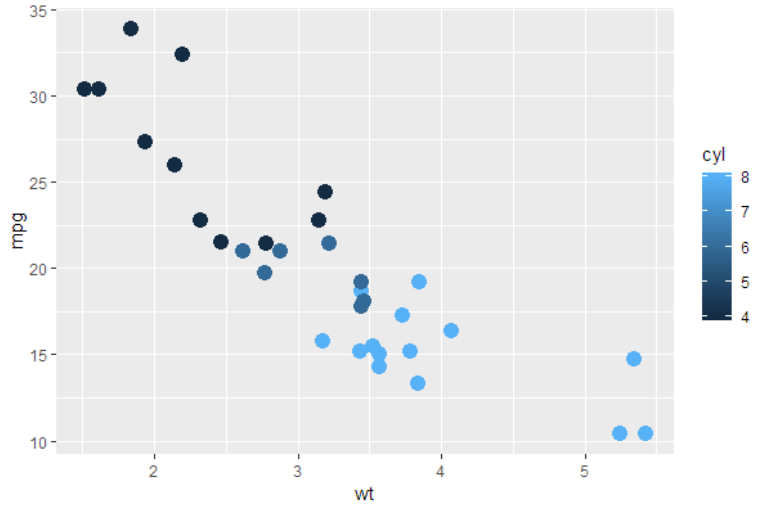
cribioinfo@uic.edu

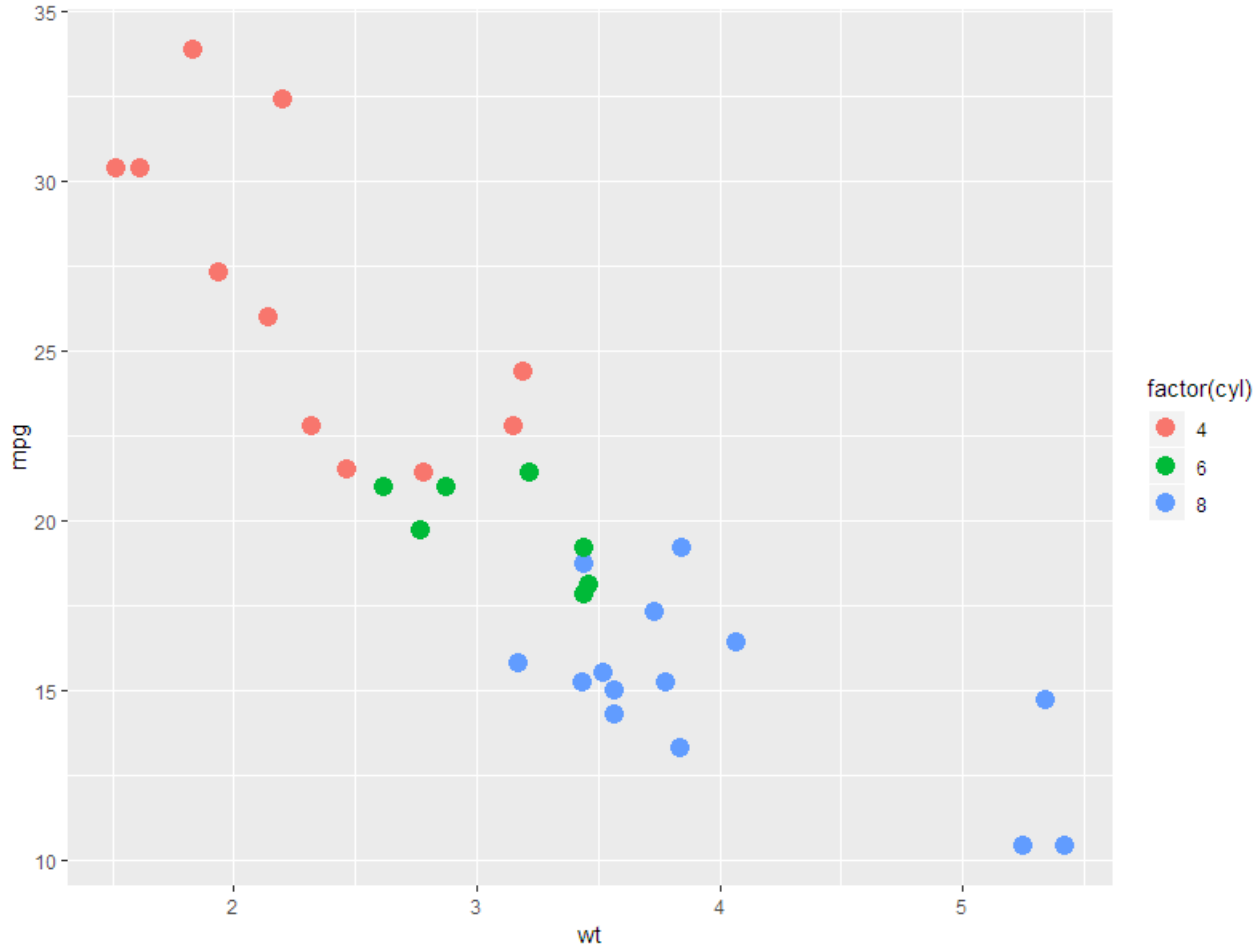
ggplot2 – Scatter Plot – variable colors

- Plot with a variable color

```
ggplot(mtcars, aes(x=wt, y=mpg, color=factor(cyl))) +
  geom_point(size=4)
```

Tell ggplot2 to use the values of the column “cyl” to color the points.
We need to set as **factor** because “cyl” is a numeric vector and ggplot2 would set colors on a continuous rather than discrete scale.





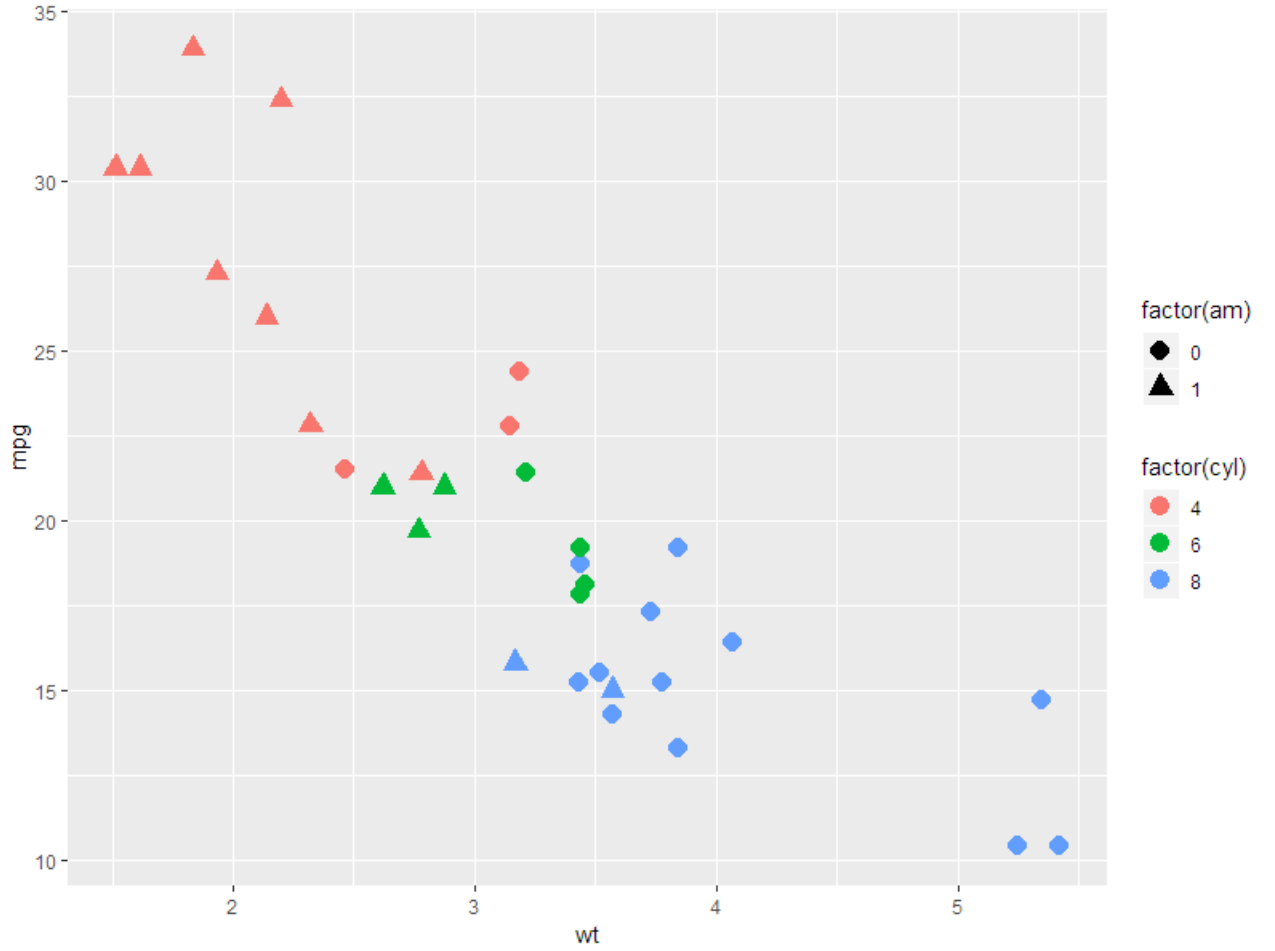
cribioinfo@uic.edu

ggplot2 –Scatter Plot – variable color and shapes

```
ggplot(mtcars, aes(wt, mpg, color=factor(cyl),
    shape = factor(am))) + geom_point(size=4)
```

Generate a plot of mtcars where

- “x” is the column “wt”
- “y” is the column “mpg”
- Color determined by value of “cyl”
- **Point shapes are determined by value of “am” (automatic vs. manual transmission)**
- Plot as scatter plot with point size of 4.



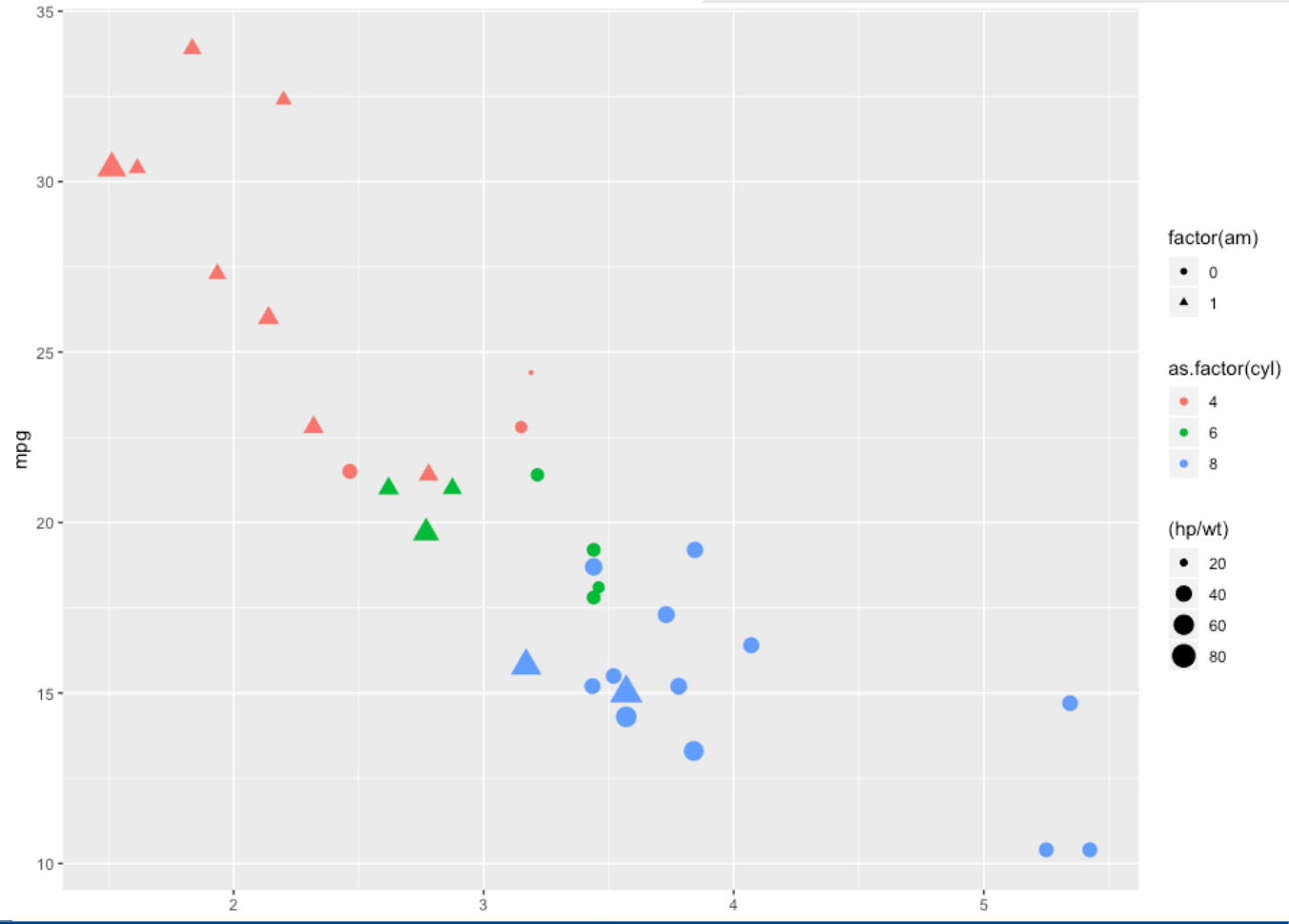
cribioinfo@uic.edu

ggplot2 – Scatter Plot – Variable color, shape and size

```
ggplot(mtcars, aes(wt, mpg, color=as.factor(cyl),
   shape = factor(am), size = (hp/wt))) +  geom_point()
```

Generate a plot of mtcars where

- "x" is the column "wt"
- "y" is the column "mpg"
- Color determined by value of "cyl"
- Point shapes are determined by value of "am" (automatic vs. manual transmission)
- **Point sizes are determined by ratio of "hp" to "wt"**
- Plot as scatter plot.



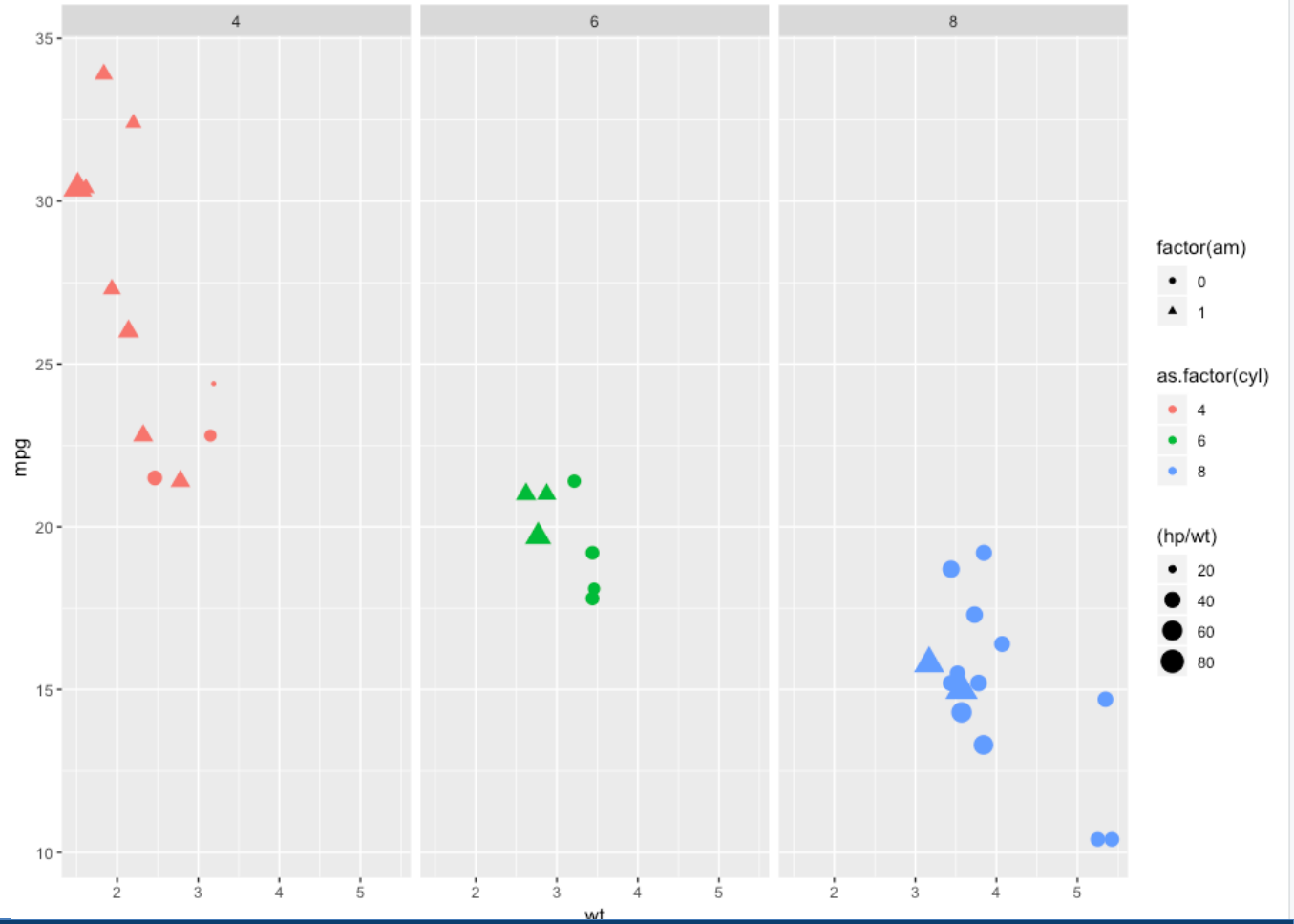
cribioinfo@uic.edu

ggplot2 – Scatter Plot – Faceting the plot

```
ggplot(mtcars, aes(wt, mpg, color=as.factor(cyl),
   shape = factor(am),size = (hp/wt))) +  geom_point() +
   facet_grid(~cyl)
```

Generate a plot of mtcars where

- "x" is the column "wt"
- "y" is the column "mpg"
- Color determined by value of "cyl"
- Point shapes are determined by value of "am" (automatic vs. manual transmission)
- Point sizes are determined by ratio of "hp" to "wt"
- Plot as scatter plot.
- **Split (facet) the plot based on value of "cyl"**



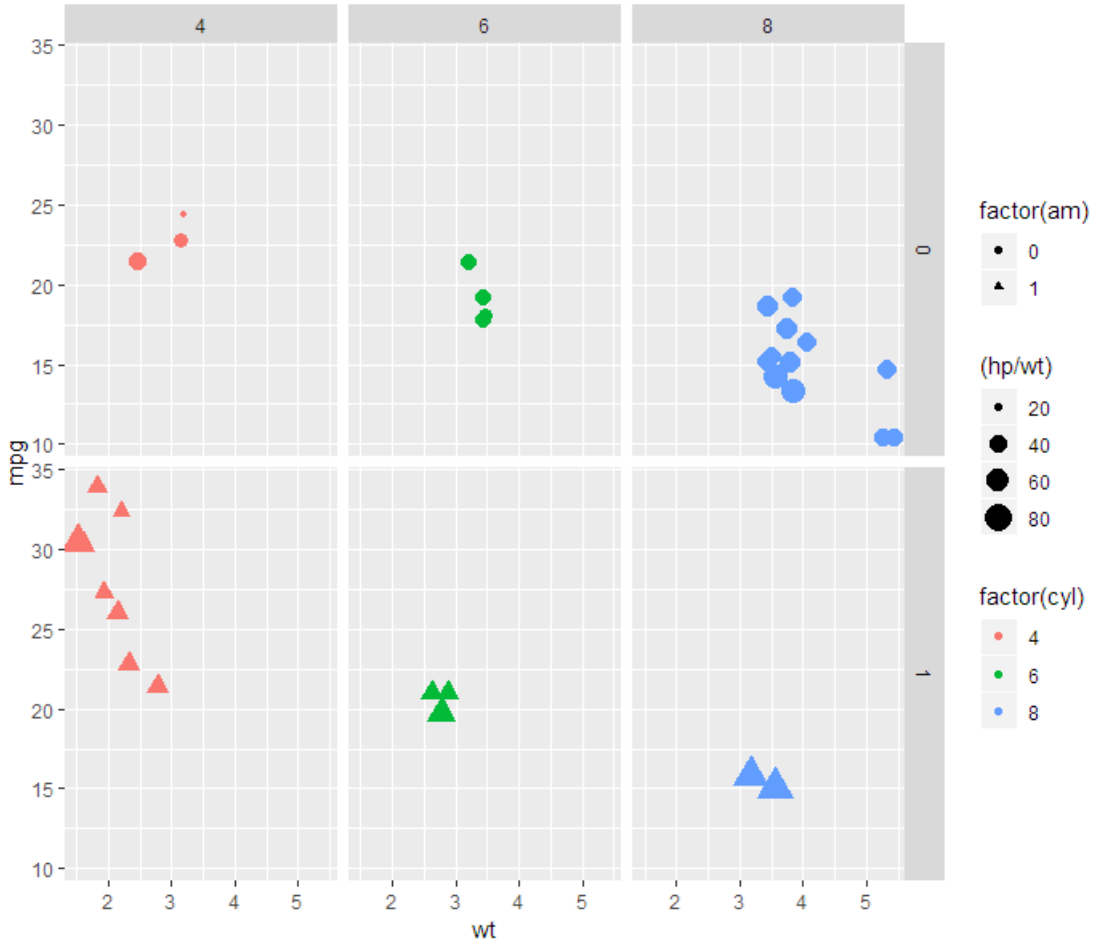
cribioinfo@uic.edu

ggplot2 – Scatter Plot – Faceting the plot

```
ggplot(mtcars, aes(wt, mpg, color=factor(cyl),
   shape=factor(am), size = (hp/wt))) + geom_point() +
   facet_grid(am ~ cyl)
```

Generate a plot of mtcars where

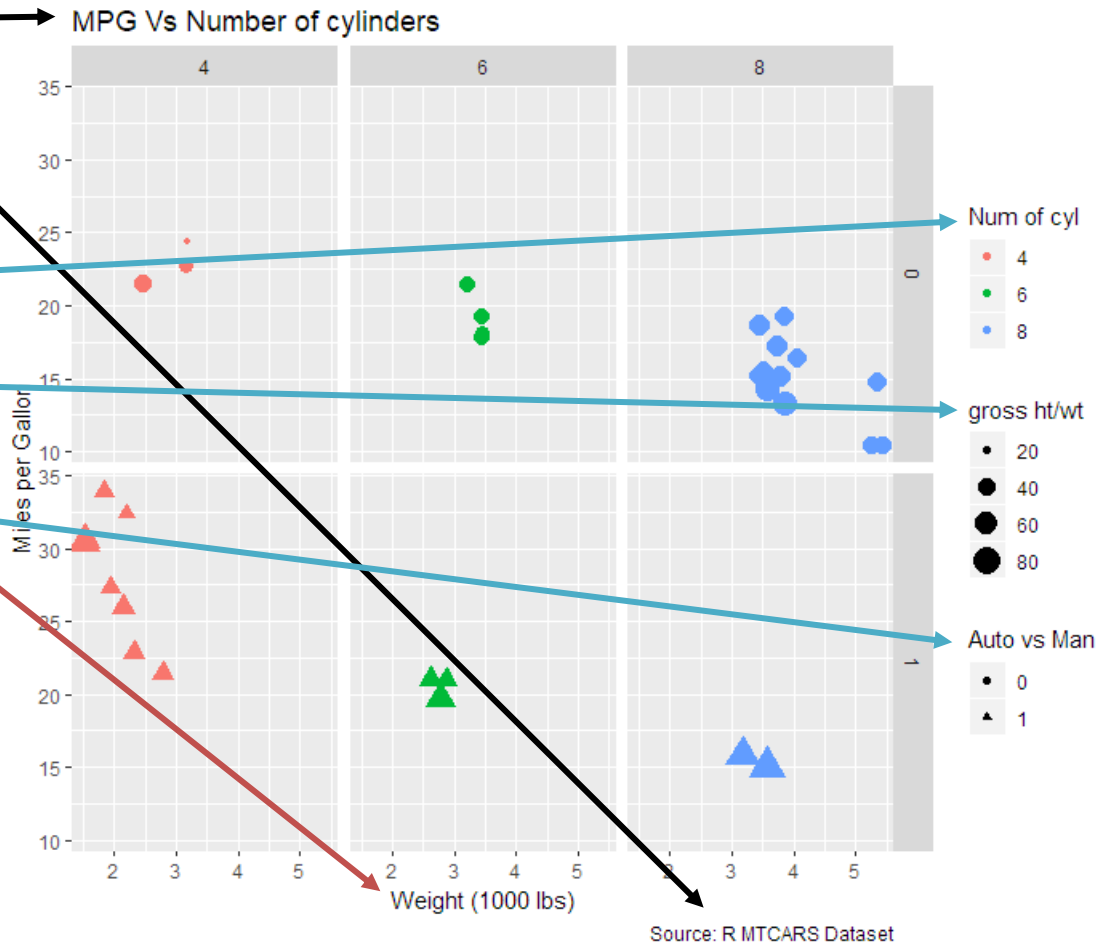
- "x" is the column "wt"
- "y" is the column "mpg"
- Color determined by value of "cyl"
- Point shapes are determined by value of "am" (automatic vs. manual transmission)
- Point sizes are determined by ratio of "hp" to "wt"
- Plot as scatter plot.
- **Split (facet) the plot based on value of "am" and "cyl" (rows by columns)**



cribioinfo@uic.edu

ggplot2 – Scatter Plot – Update labels

```
ggplot(mtcars, aes(wt, mpg, color=factor(cyl),
   shape=factor(am), size = (hp/wt))) + geom_point() +  facet_grid(am ~ cyl) +
   labs(title="MPG Vs Number of cylinders",
      caption="Source: R MTCARS Dataset",
      x="Weight (1000 lbs)",
      y="Miles per Gallon",
      color="Num of cyl",
      size="gross hp/wt",
      shape="Auto vs Man")
```



cribioinfo@uic.edu

ggplot2 – grammar of graphics plot – Scatter Plot

- A very long command…

```
ggplot(mtcars, aes(wt, mpg, colour=factor(cyl), shape=factor(am),
   size=(hp/wt))) +  geom_point() + facet_grid(am ~ cyl) +
   labs(title="MPG Vs Number of cylinders", y="Miles per Gallon",
   x="Weight (1000Ibs)", caption="Source: R MTCARS Dataset", shape="Auto vs Man",
   colour="Num of cyl", size="gross hp/wt").
```

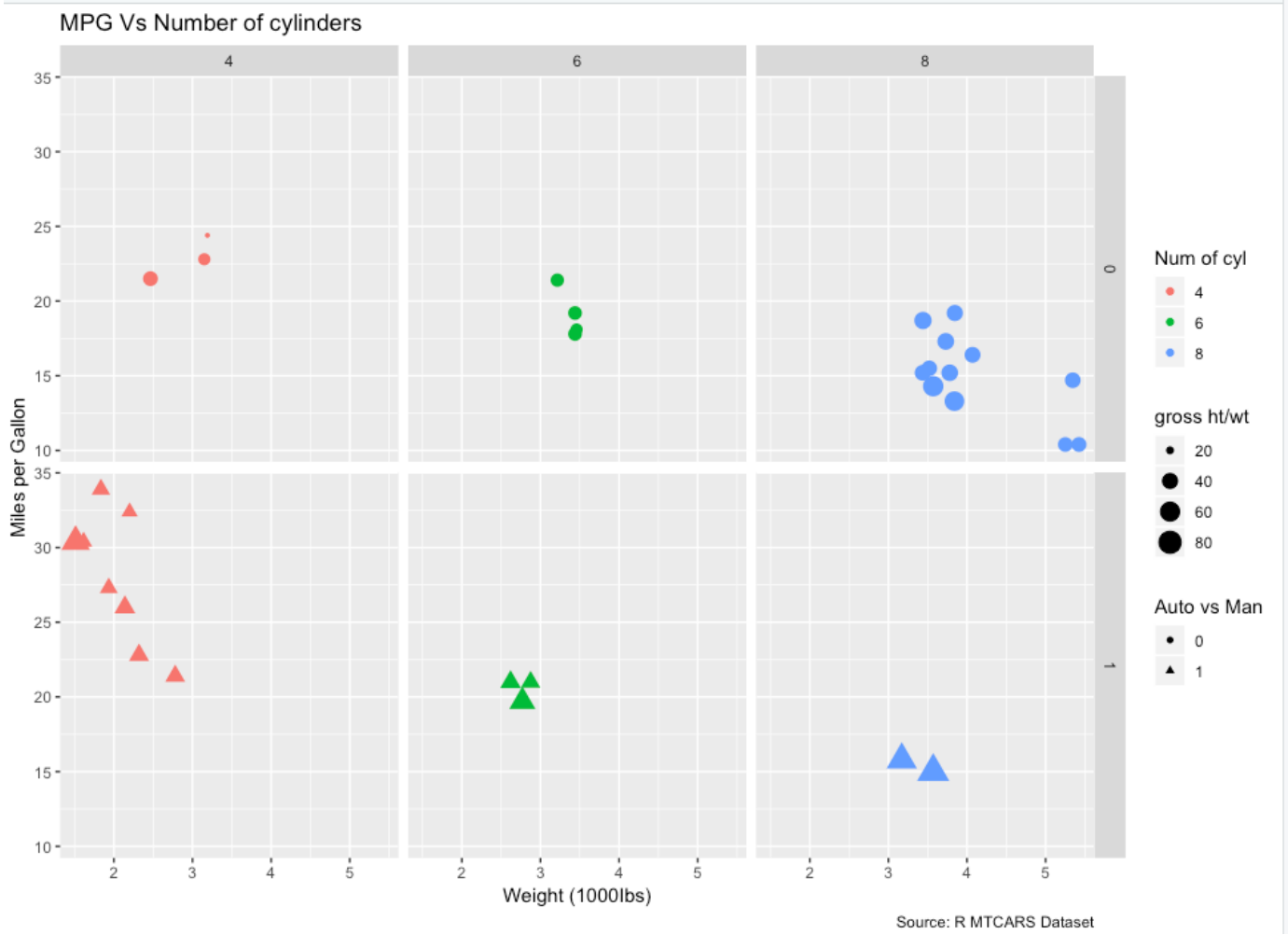
- We can start storing some of the commands in variables

```
basePlot <- ggplot(mtcars, aes(wt, mpg, colour=factor(cyl), shape=factor(am),
   size=(hp/wt)))

basePlotTitle <- labs(title="MPG Vs Number of cylinders", y="Miles per Gallon",
   x="Weight (1000Ibs)", caption="Source: R MTCARS Dataset", shape="Auto vs Man",
   colour="Num of cyl", size="gross hp/wt")

basePlot + geom_point() + facet_grid(am ~ cyl) + basePlotTitle
```

ggplot2 – grammar of graphics plot – Scatter Plot

```
basePlot + geom_point() + facet_grid(am ~ cyl) +
  basePlotTitle
```

- Same plot as the original long command, but possible to save parts as variables then combine in the end.
- Can also allow you to change parts without having to retype the entire command.

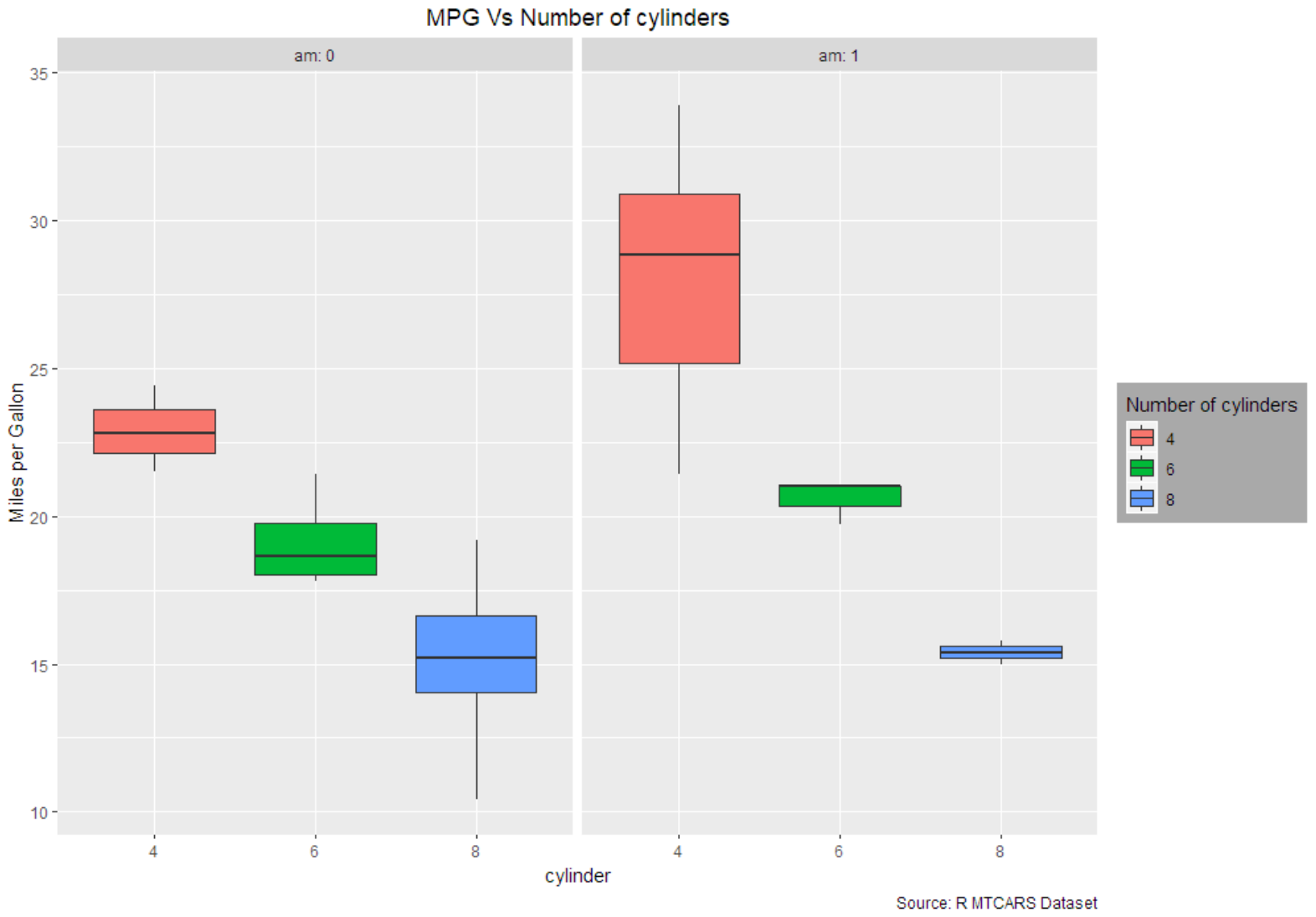


cribioinfo@uic.edu

Exercise 2.1: ggplot2 - scatter plots

Make scatter plots using different ggplot2 functions

ggplot2 – grammar of graphics plot – box plot

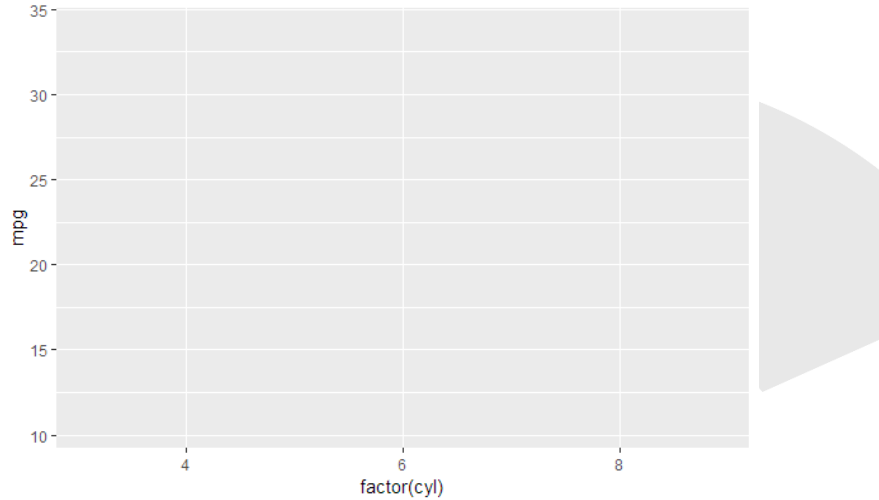


cribioinfo@uic.edu

ggplot2 – grammar of graphics plot – Box Plot

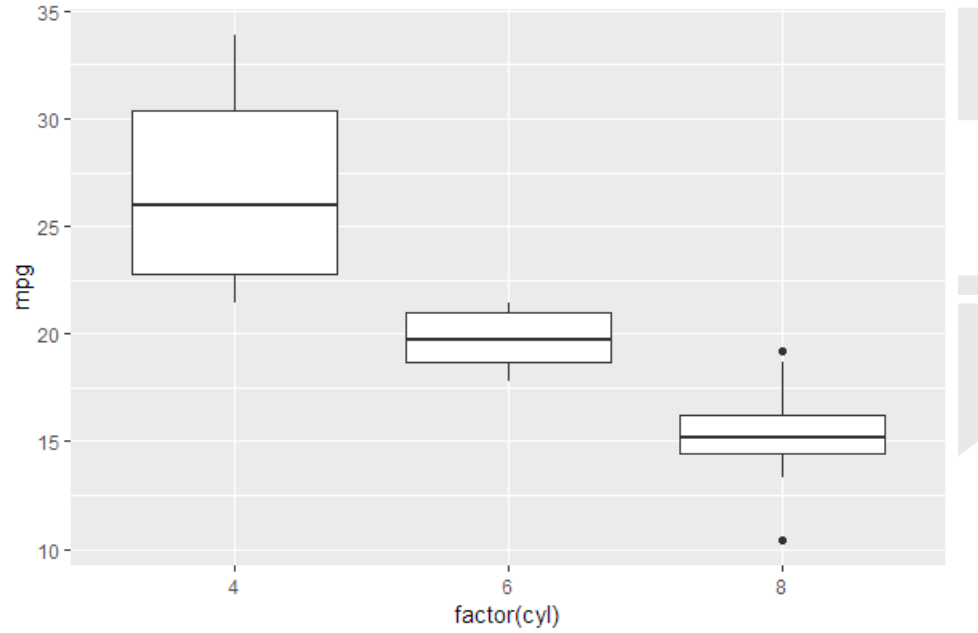
- Create a frame

```
ggplot(mtcars, aes(x=factor(cyl), y=mpg))
```



- Add a box plot

```
ggplot(mtcars, aes(x=factor(cyl), y=mpg))
 + geom_boxplot()
```

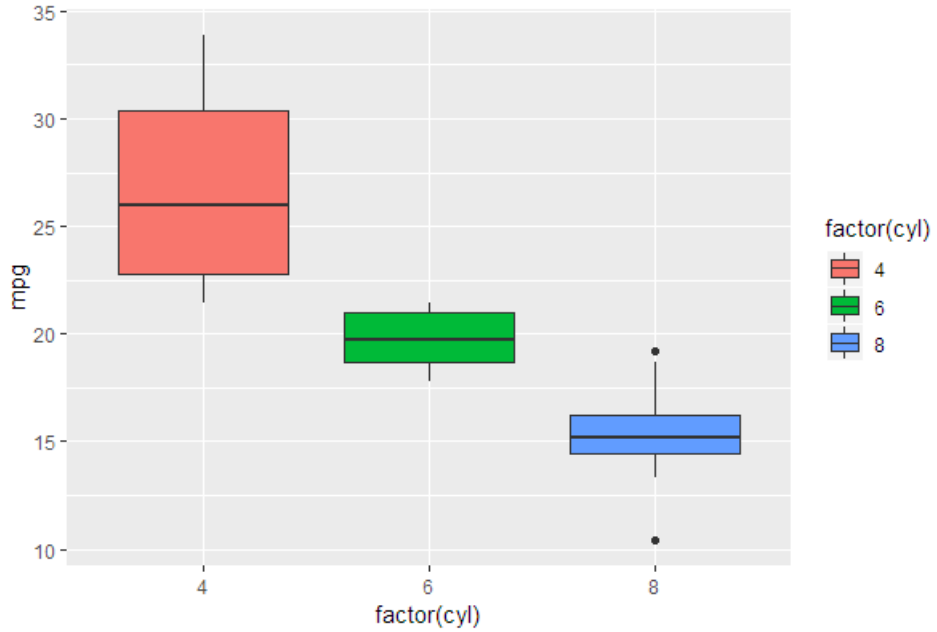


cribioinfo@uic.edu

ggplot2 – box plot – variable colors

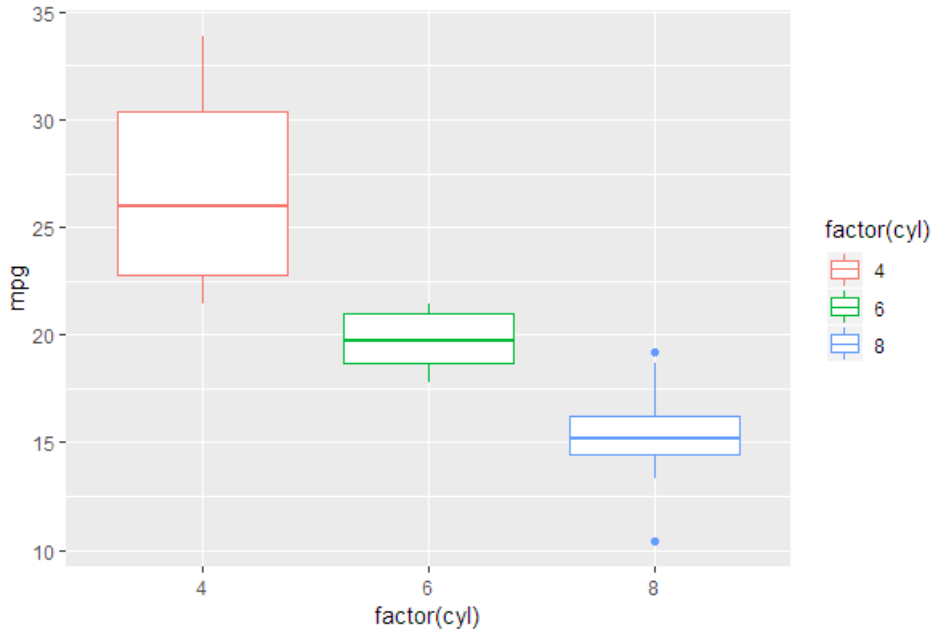
- Set fill color

```
ggplot(mtcars, aes(x=factor(cyl),
   y=mpg)) +
geom_boxplot(aes(fill=factor(cyl)))
```



- Set line color

```
ggplot(mtcars, aes(x=factor(cyl),
   y=mpg)) +
geom_boxplot(aes(color=factor(cyl)))
```

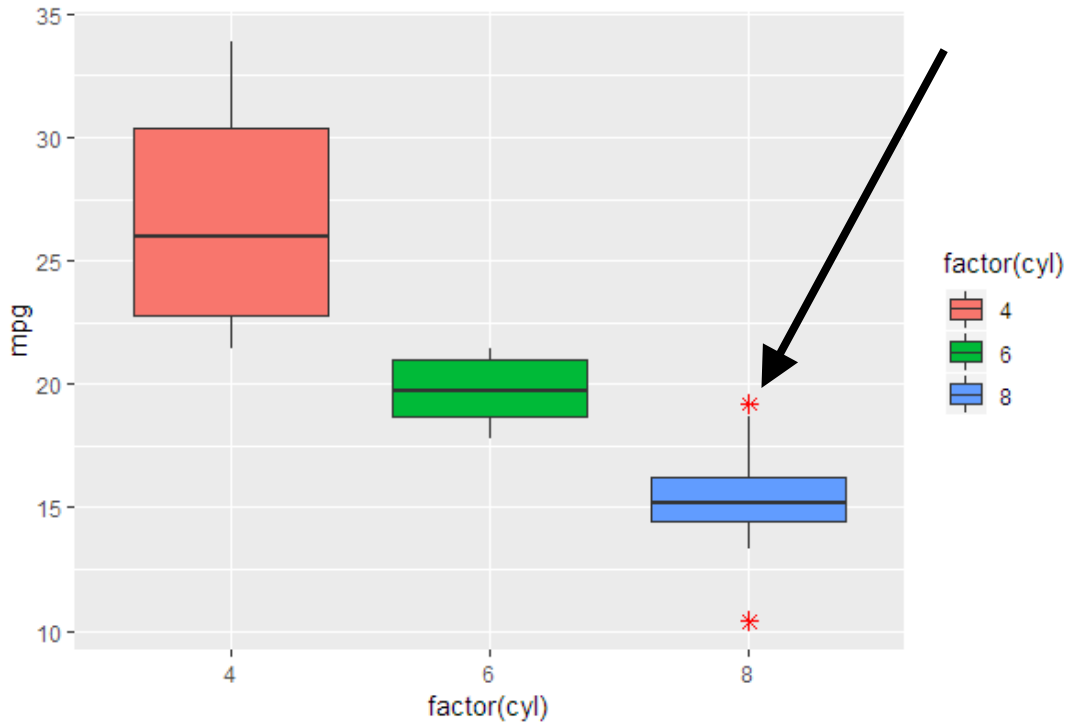


cribioinfo@uic.edu

ggplot2 – box plots – variable colors

- Custom outlier color and shape

```
ggplot(mtcars, aes(x=factor(cyl), y=mpg)) +
  geom_boxplot(aes(fill=factor(cyl)), outlier.fill="red",
  outlier.colour="red", outlier.shape=8, outlier.size=2)
```



cribioinfo@uic.edu

ggplot2 – box plot – Plot using variables

```
boxParam <- geom_boxplot( aes(fill =   as.factor(cyl)),
  outlier.fill="red", outlier.colour="red", outlier.shape=8,
outlier.size=2)

boxTitle <- labs(title="MPG Vs Number of cylinders",
  y="Miles per Gallon", x="cylinder",
  caption="Source: R MTCARS Dataset", fill="Number of cylinders")

boxTheme <- theme(plot.title = element_text(hjust = 0.5),
  legend.background = element_rect(fill = "darkgray"))

ggplot(mtcars, aes(x=as.factor(cyl), y=mpg)) + boxParam +
  boxTitle + boxTheme
```


cribioinfo@uic.edu

ggplot2 – box plot – Plot using variables

```
ggplot(mtcars, aes(x=as.factor(cyl), y=mpg)) + boxParam +
   boxTitle + boxTheme
```

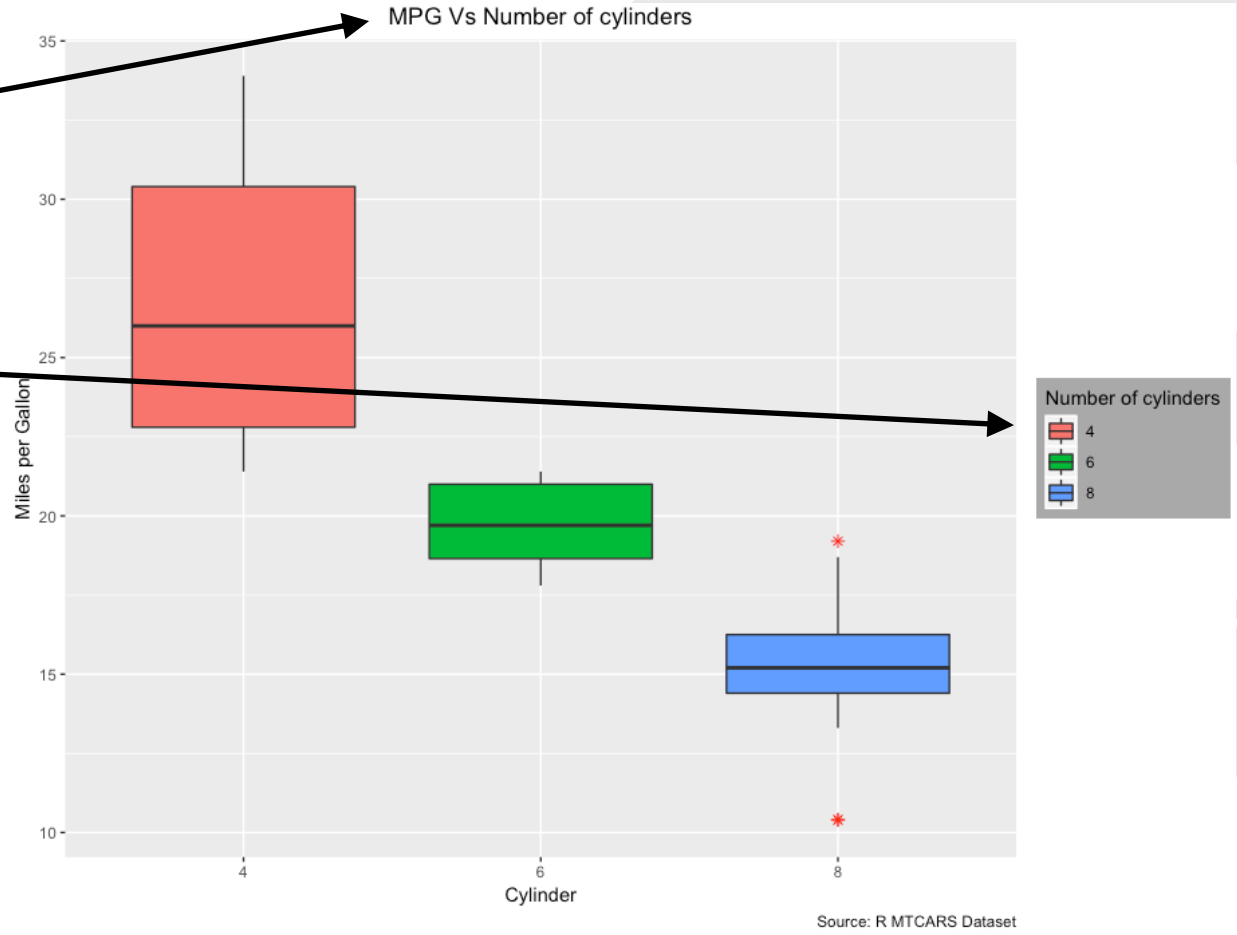
Theme items

- Center the title

```
plot.title = element_text(hjust = 0.5)
```

- Set the background color of the legend

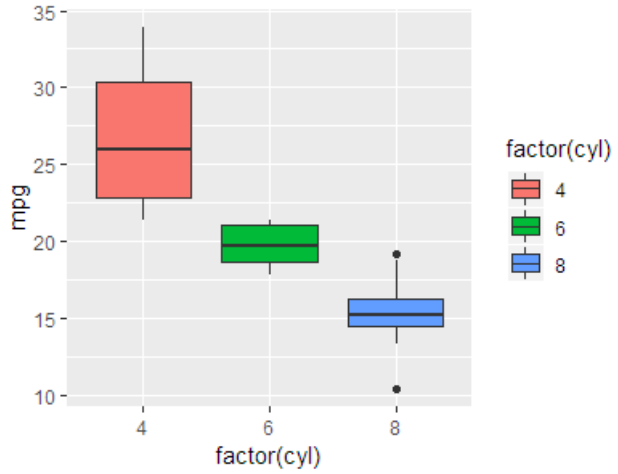
```
legend.background =
    element_rect(fill = "darkgray"))
```



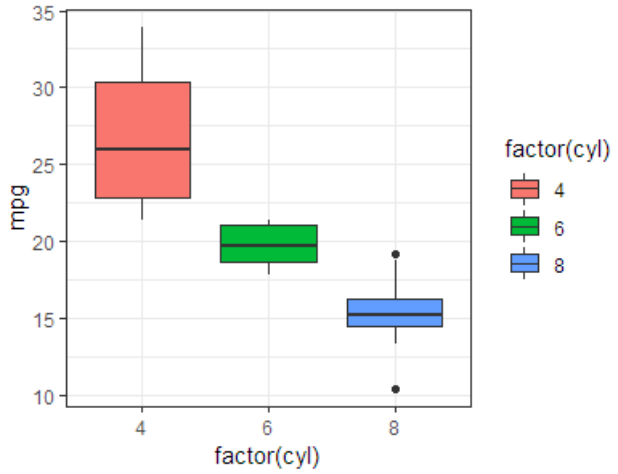
cribioinfo@uic.edu

ggplot2 – Built-in Themes

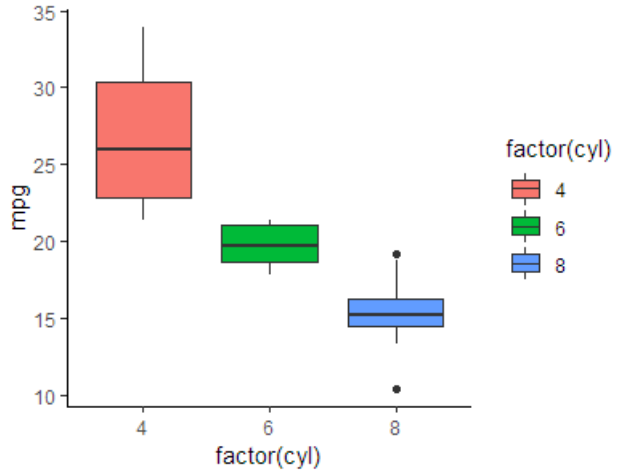
theme_gray() [default]



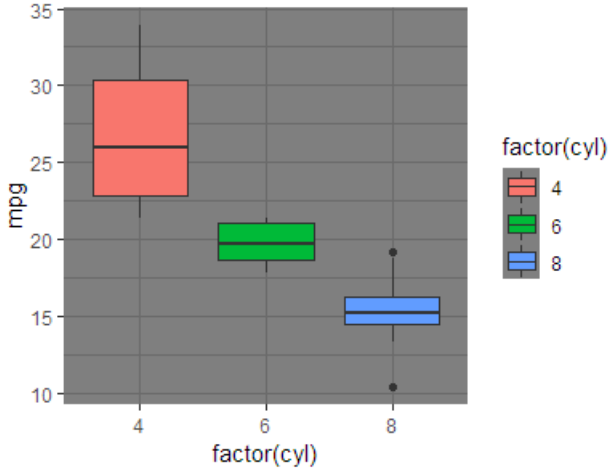
theme_bw()



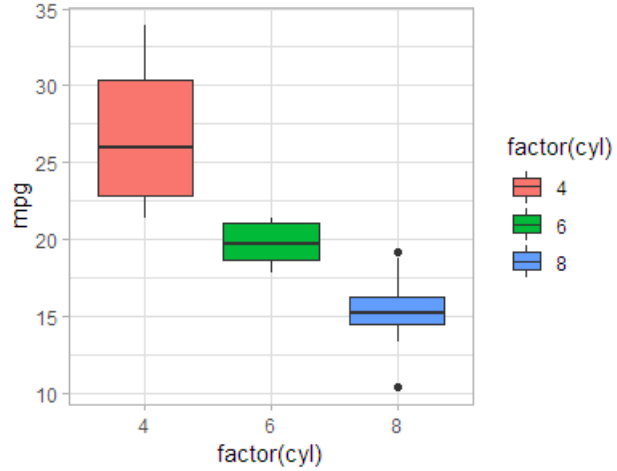
theme_classic()



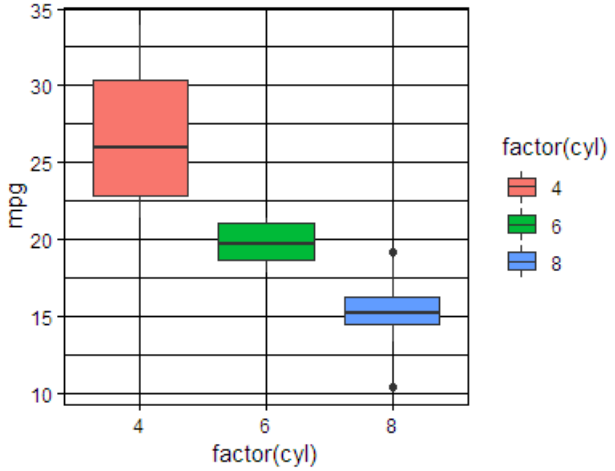
theme_dark()



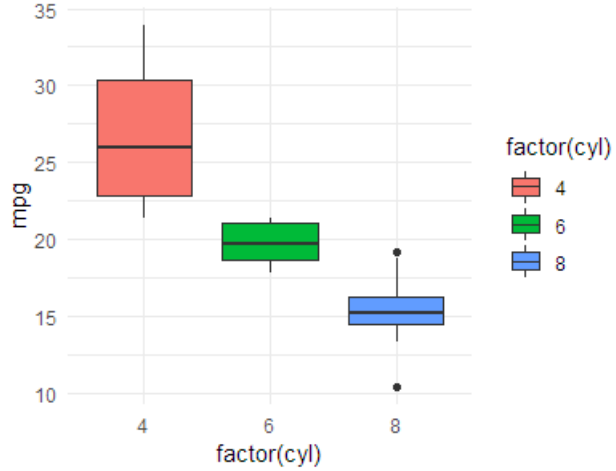
theme_light()



theme_linedraw()



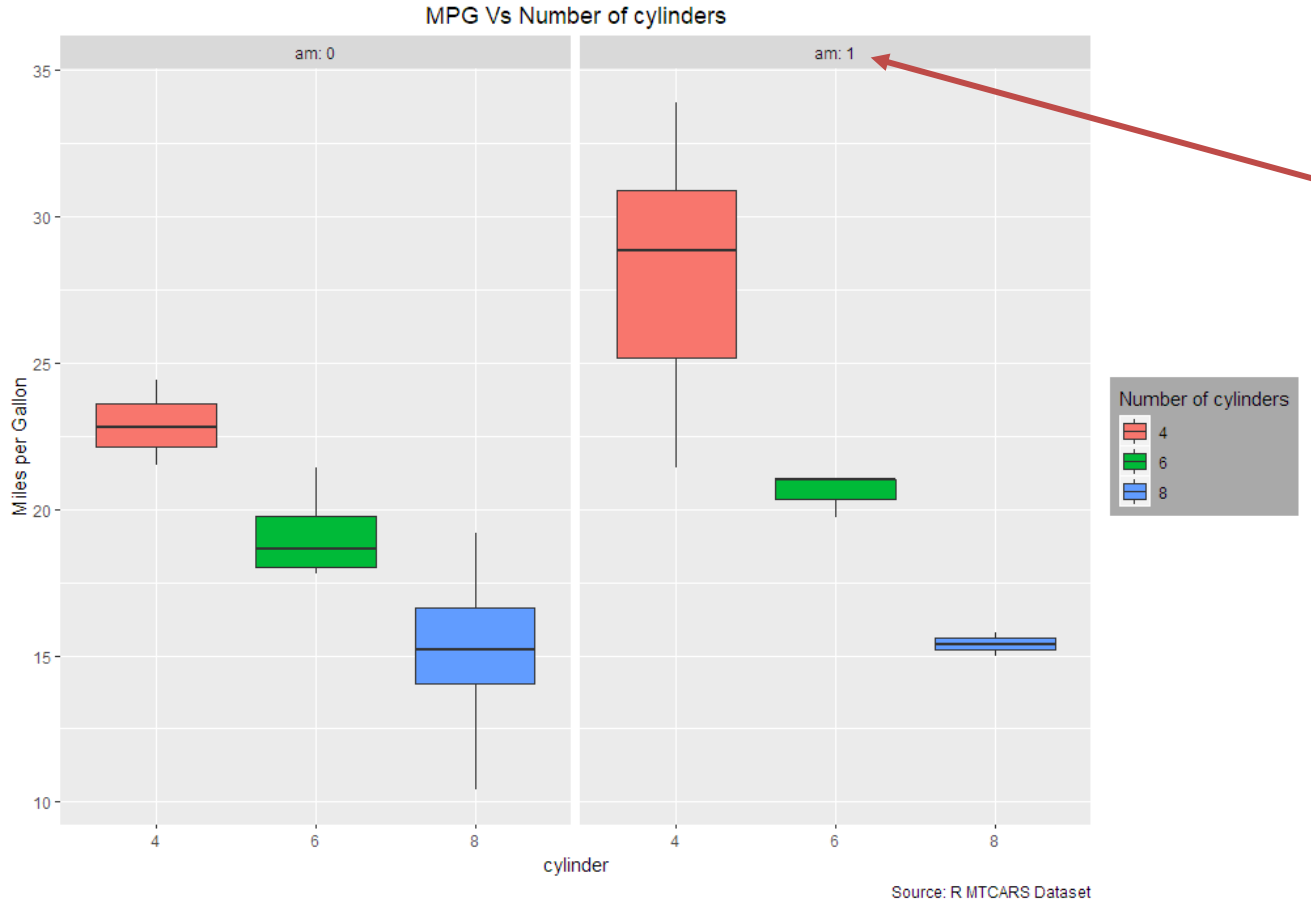
theme_minimal()



cribioinfo@uic.edu

ggplot2 – box plot – Add a facet

```
ggplot(mtcars, aes(x=as.factor(cyl), y=mpg)) + boxParam +
  boxTitle + boxTheme + facet_grid(~ am, labeller=label_both)
```



`labeller` parameter changes how the facets are labelled.

`label_value` (default) – Use values
`label_both` – Use column name and value

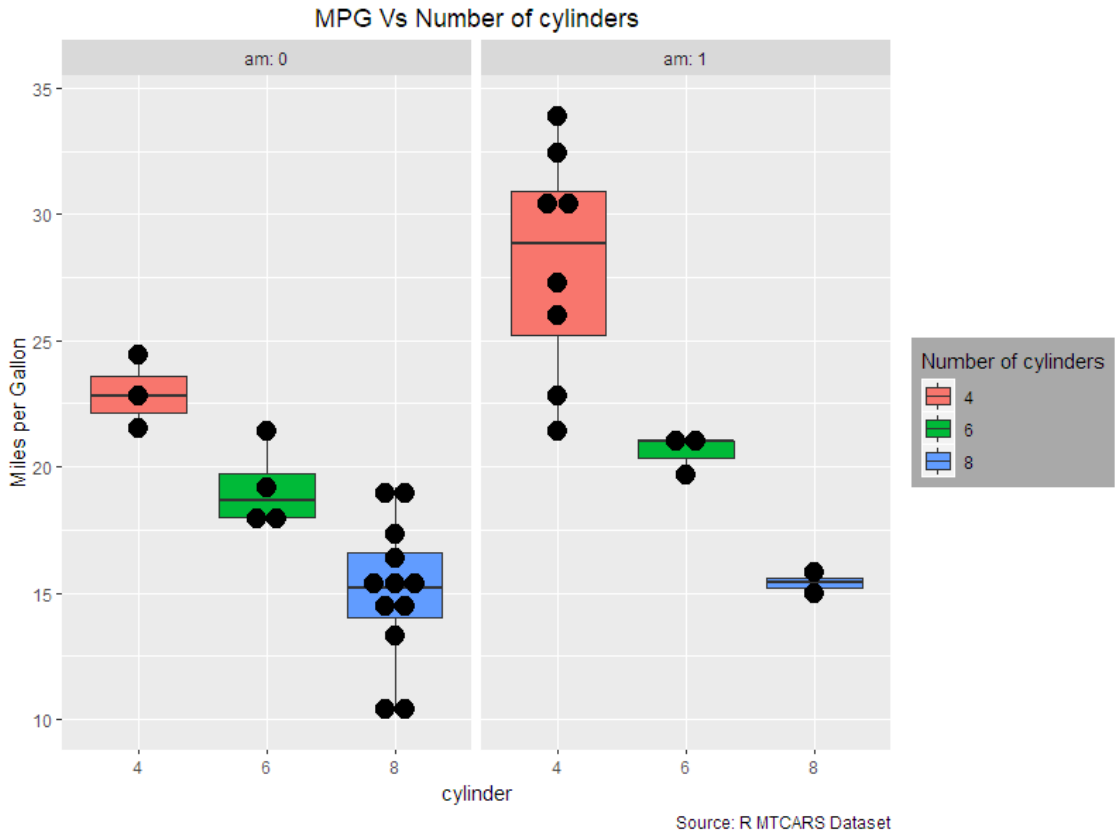
Additional label_* functions are available and it is possible to create custom labeller functions.

cribioinfo@uic.edu

ggplot2 – box plot – Overlay dotplot

```
ggplot(mtcars, aes(x=as.factor(cyl), y=mpg)) + boxParam +
  boxTitle + boxTheme + facet_grid(~ am, labeller=label_both) +
  geom_dotplot(binaxis="y", stackdir="center")
```

Added an additional geometric layer by adding a single command.



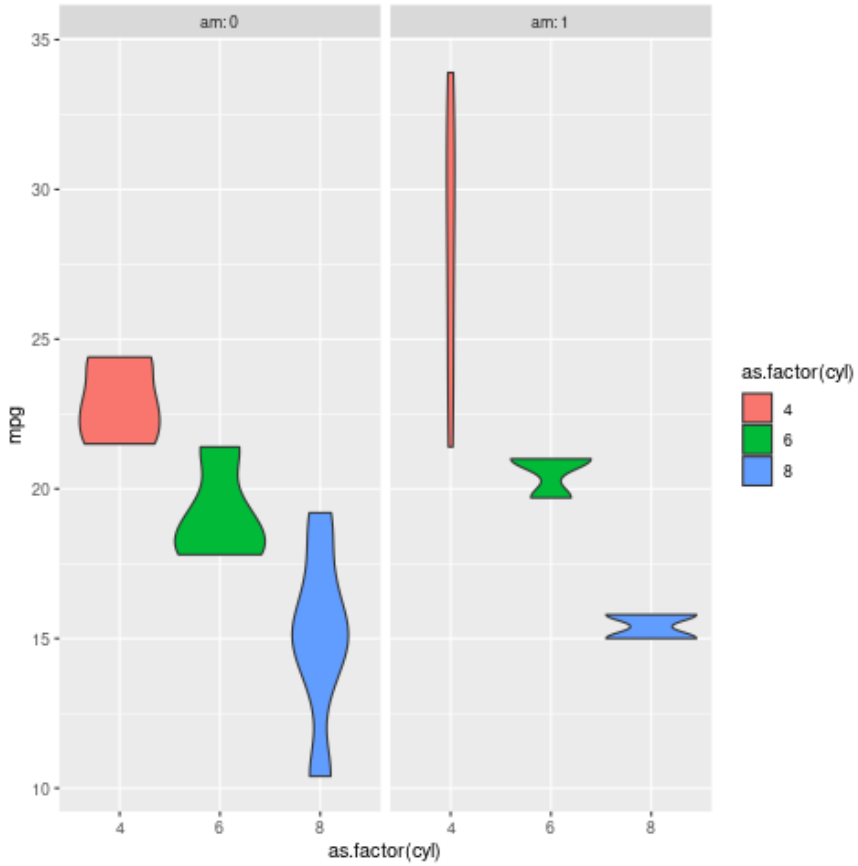
`binaxis="y"` – bin along the "y" axis, default is "x"
`stackdir="center"` - center the dot stacks. Default is to align to bottom or left (based on binaxis)

NOTE: The dotplot geometrics layer inherited the "x" and "y" aesthetics mapping from the ggplot() command, but did not inherit the "fill" aesthetic mapping as it was included in the geom_boxplot() command.

ggplot2 – violin plot

```
ggplot(mtcars, aes(x=as.factor(cyl), y=mpg)) +
  geom_violin(aes(fill=as.factor(cyl))) +
  facet_grid(~ am, labeller=label_both)
```

Plot a violin plot rather than a box plot



cribioinfo@uic.edu

Exercise 2.2: ggplot2 - box plots and violin plots

Make box plots using different ggplot2 functionalities

cribioinfo@uic.edu

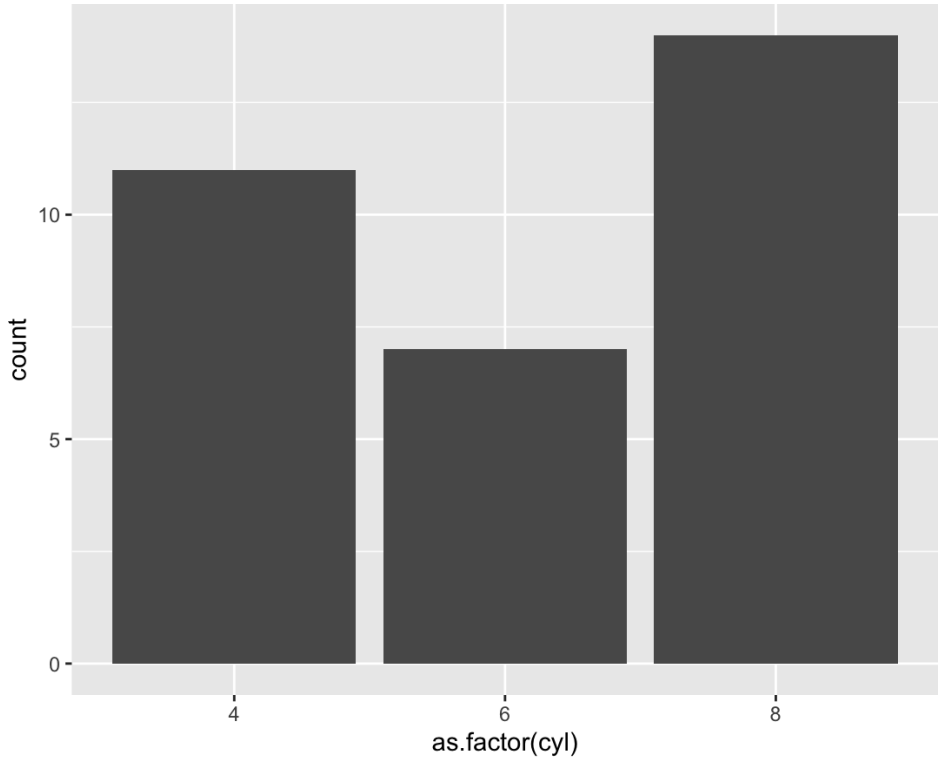
LUNCH

ggplot2 – barplot

`geom_bar()`: plot frequency of items in a vector

```
ggplot(mtcars,aes(x=as.factor(cyl))) +
  geom_bar()
```

Number of cars with 4, 6, or 8 cylinders in the data frame



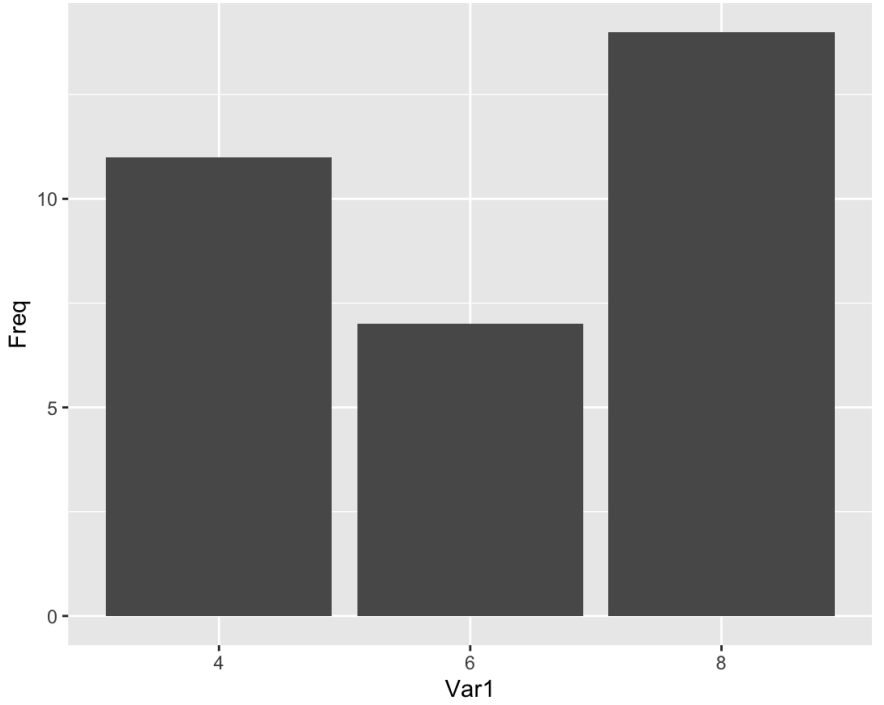
cribioinfo@uic.edu

ggplot2 – barplot

- `geom_col()`: plot bars directly on values in data frame

```
cyl.table = data.frame(table(mtcars$cyl))
cyl.table
  Var1 Freq
1    4   11
2    6    7
3    8   14
```

Number of cars with 4, 6, or 8 cylinders in the data frame



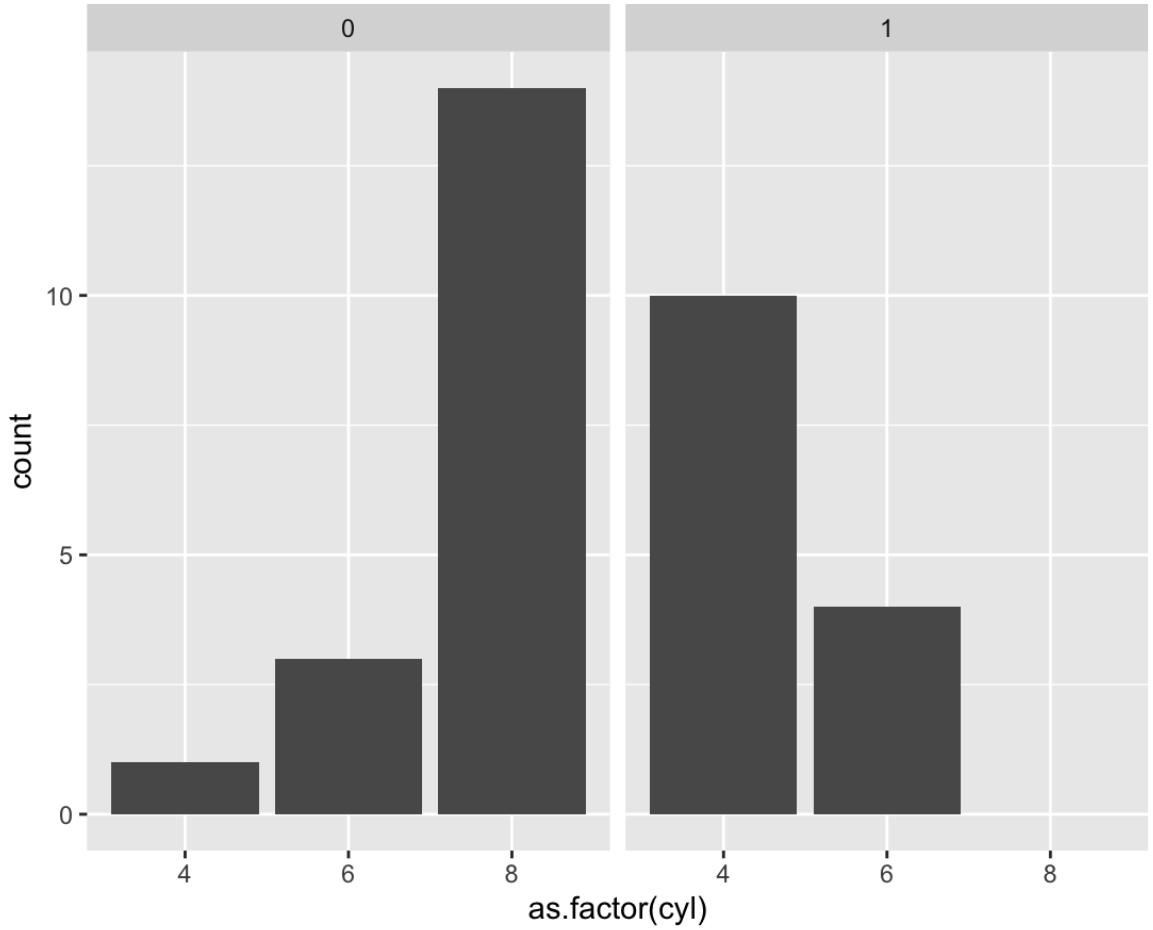

```
ggplot(cyl.table,aes(x=Var1,y=Freq))+
 geom_col()
```


cribioinfo@uic.edu

ggplot2 – barplot – Facet the plot

```
ggplot(mtcars,aes(x=as.factor(cyl))) +
  geom_bar() + facet_grid(~vs)
```

Number of cars with 4, 6, or 8 cylinders for V-shared vs straight engines

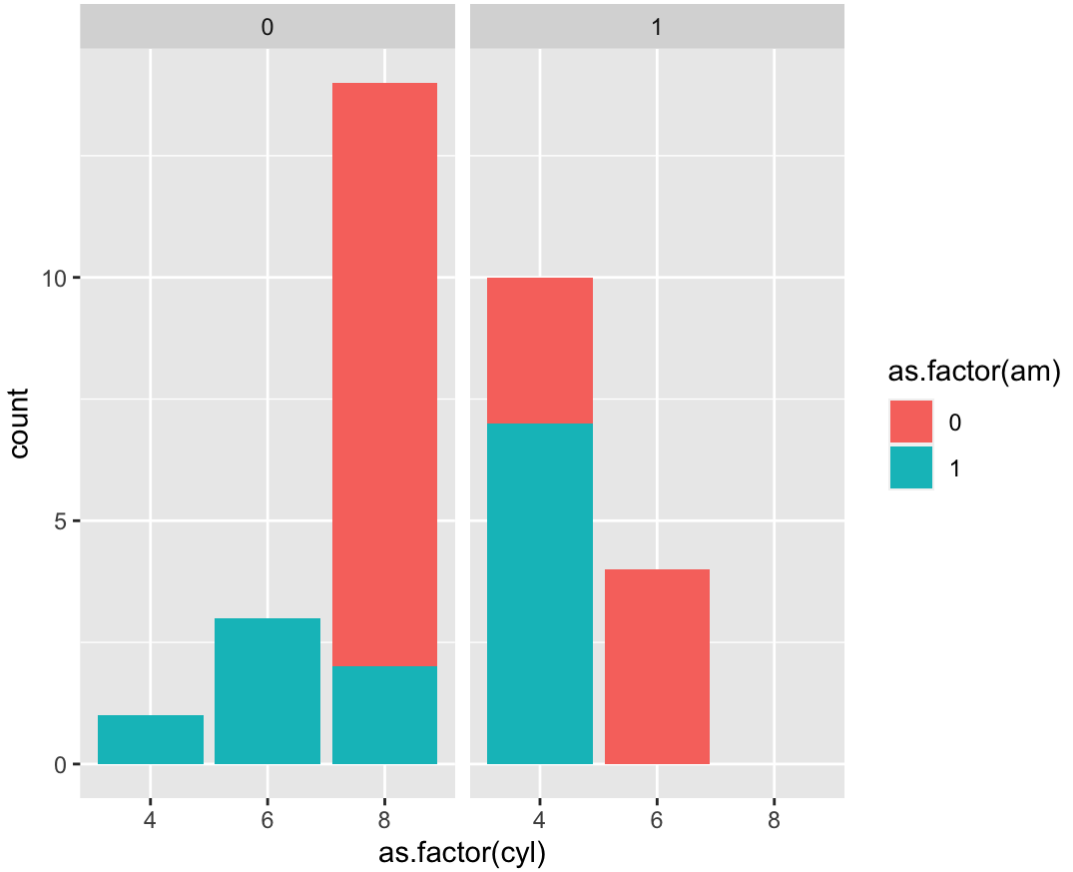


cribioinfo@uic.edu

ggplot2 – barplot – Split by another factor with color

```
ggplot(mtcars,aes(x=as.factor(cyl))) +
  geom_bar(aes(fill=as.factor(am))) + facet_grid(~vs)
```

Also with coloring for automatic or manual transmissions



Note: default is stacked barplot. To put side-by-side, use:

```
geom_bar(..., position='dodge')
```

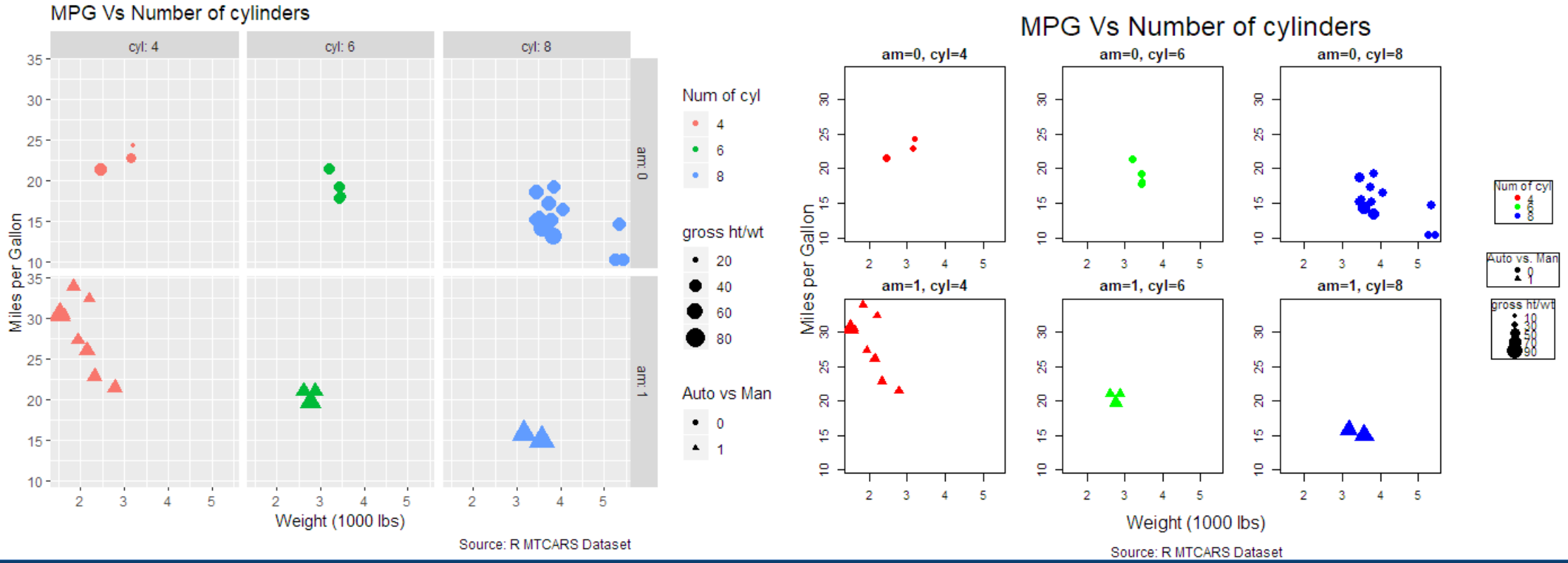

cribioinfo@uic.edu

Exercise 2.3: ggplot2 - barplot

Make barplots using different ggplot2 functionalities

ggplot2 vs. Basic R plots

- ggplot2 can allow one to quick generate plots and create scales (colors, sizes, shapes) automatically.
  - Ggplot2 version: 3 lines of code (383 characters)
  - Basic R version: 39 lines of code (2344 characters)
- ggplot2 also allows one save components to quickly reuse/reformat plots without having to re-type all of the commands.



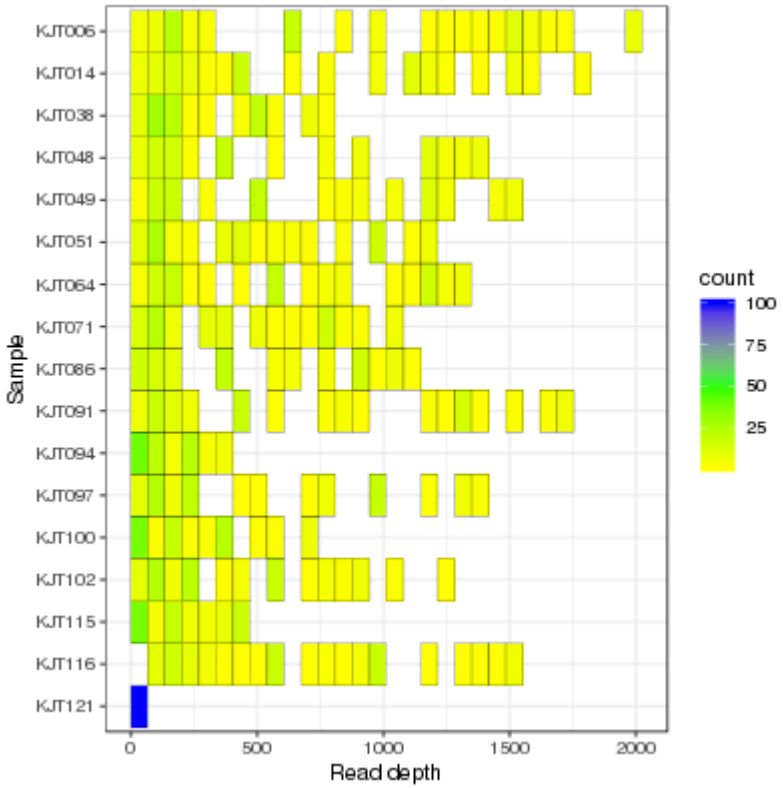
cribioinfo@uic.edu

Exercise 2.4: ggplot2 vs built-in R plots (Take home exercise)

Compare ggplot2 and built-in R plot functions
(on mtcars data set)

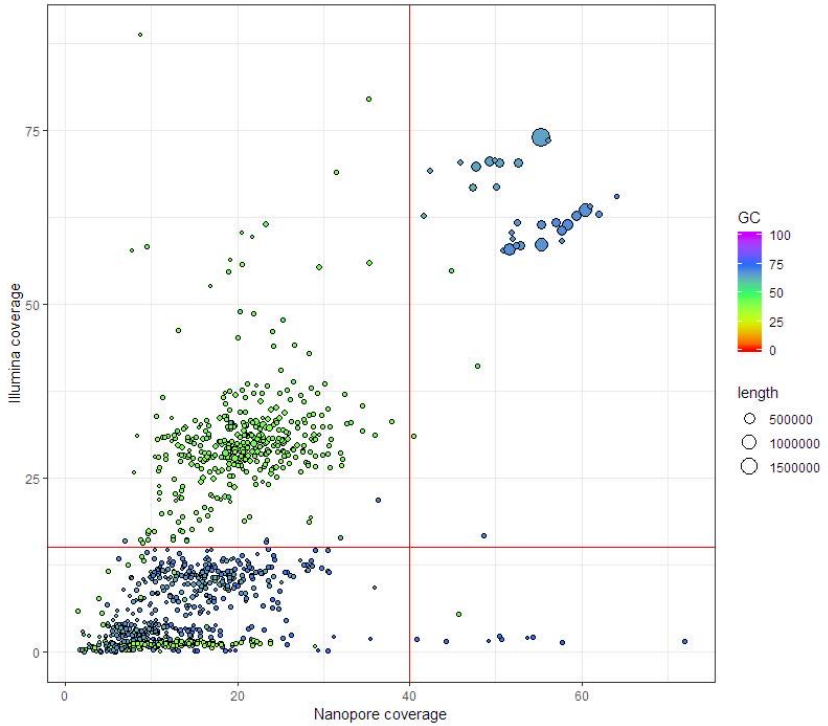
cribioinfo@uic.edu

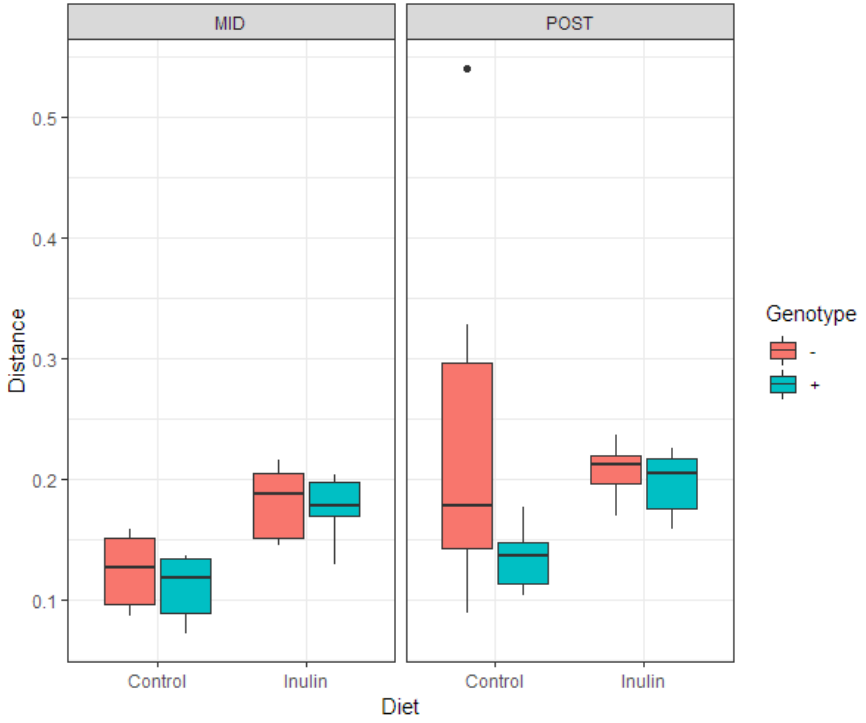
More ggplot2 examples



2D bin plot
(geom_bin2d)

Scatter plot with variable color and size points with fixed vertical and horizontal lines

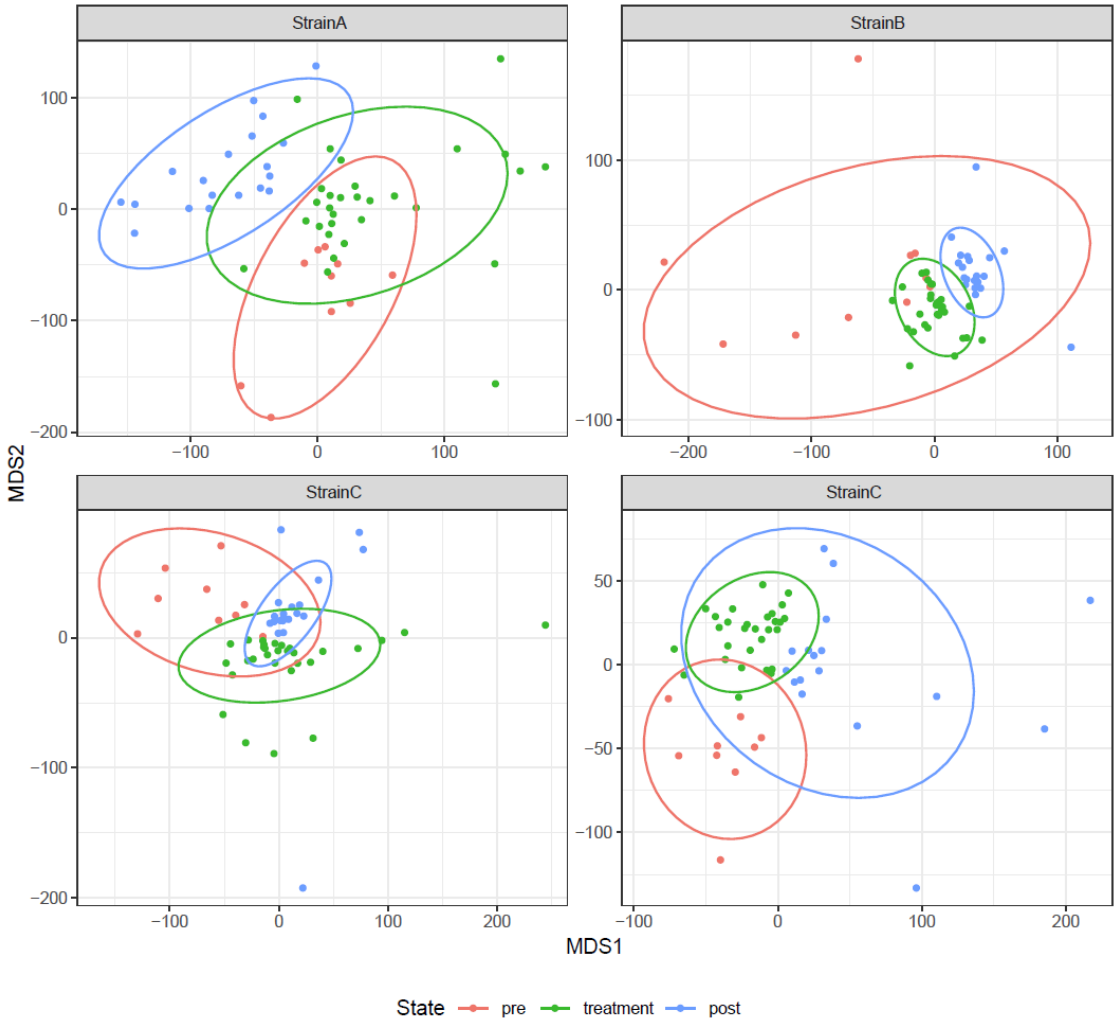




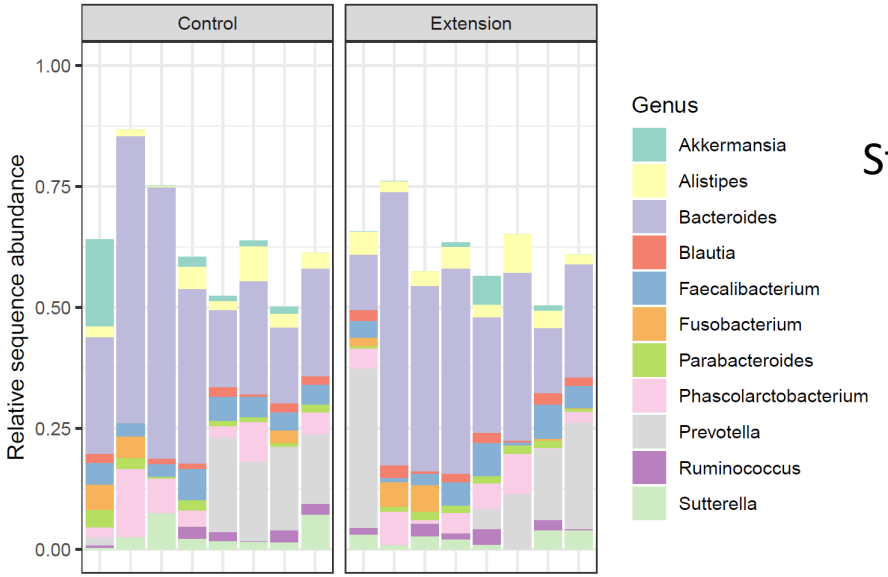
Box plot with variable color and facet

cribioinfo@uic.edu

More ggplot2 examples

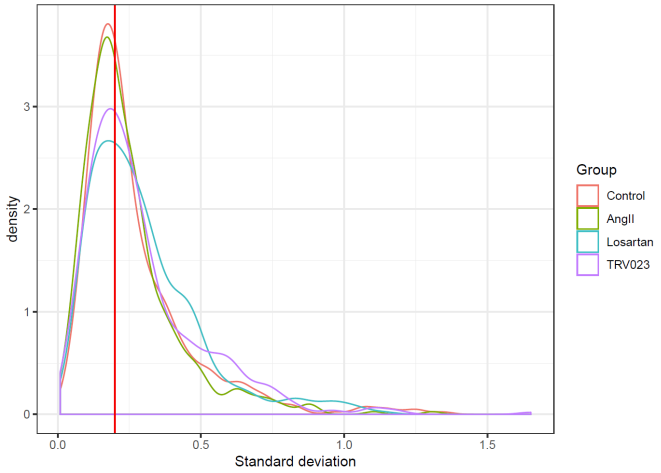


Scatter plot with variable color and facet
Confidence ellipses automatically generated using stat_ellipse



Stacked barchart with facet

Density plots with variable colors and fixed vertical line



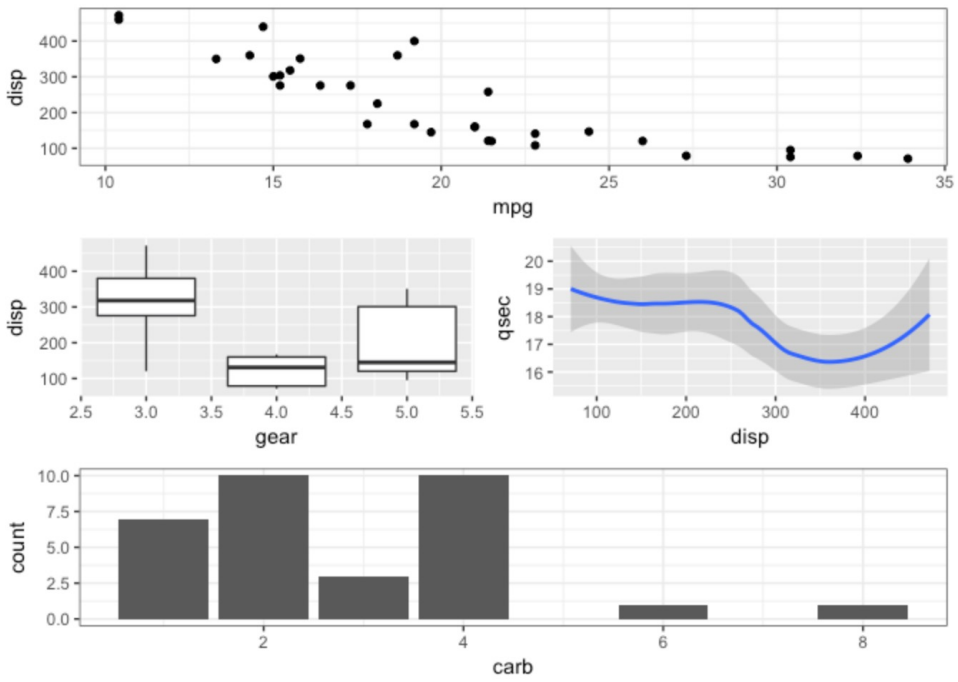
cribioinfo@uic.edu

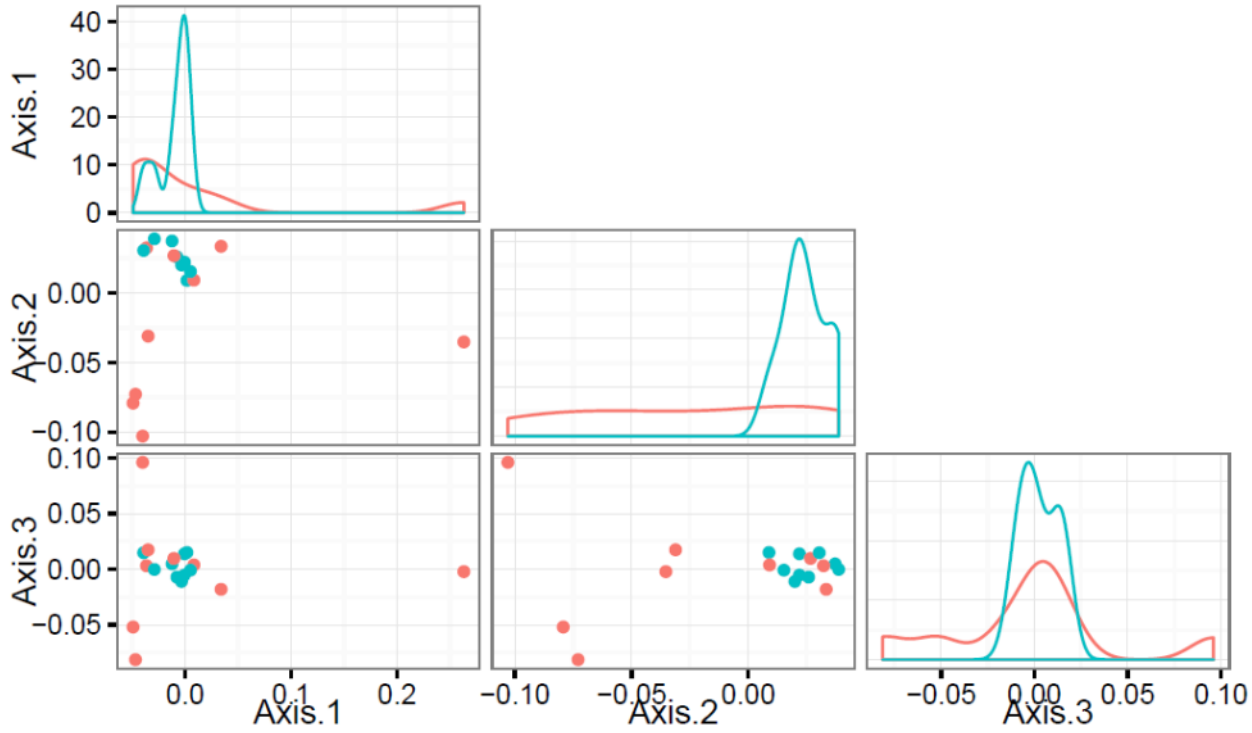
ggplot2 – Resources

- Reference: https://ggplot2.tidyverse.org/reference/index.html
- Cheatsheet: https://github.com/rstudio/cheatsheets/raw/master/data-visualization-2.1.pdf
- R Graph Gallery (https://www.r-graph-gallery.com/)
  - https://www.r-graph-gallery.com/portfolio/ggplot2-package/
- Online ggplot2 user community
  - http://groups.google.com/group/ggplot2
  - http://stackoverflow.com/tags/ggplot2
- Book: ggplot2 by Hadley Wickham
  - http://moderngraphics11.pbworks.com/f/ggplot2-Book09hWickham.pdf

Extensions to ggplot2

- There are a number of “extensions” to ggplot2 that add additional features or allow one to generate specialized plots
- https://mode.com/blog/r-ggplot-extension-packages/





cribioinfo@uic.edu

Statistics

- Statistics test
- Regression

Reminder: formula (~) in R

- R uses formulas with ~ in a variety of applications

```
outcome ~ factor1 + factor2 + factor1:factor2
outcome ~ factor1 * factor2
```

  - Left-hand side is dependent variable (measurement)
  - Right-hand side lists independent variables (experimental factors)
    - Interaction terms can be included with `:`
    - Complete model with `*`
  - R automatically builds out linear models based on all levels of factors
- Data needs to be in long format
  - `outcome`, `factor1`, and `factor2` are column names from a data frame

Descriptive statistics

- Descriptive statistics:
  - General overview of the data by describing, presenting, summarizing the data
  - **Mean, standard devation, max, min, median, 75th percentile, etc.**
- Computing in R:
  - Run summary statistics over all rows or columns: **apply**
  - Look into sub-groups: **summaryBy** (doBy package)
- Data set used: baby's birth weight. For example:
  - Mean values of all columns – use apply
  - Mean birth weight based on Mother's smoking status – use summaryBy

bwt	gestation	parity	age	height	weight	smoke
120	284	0	27	62	100	0
113	282	0	33	64	135	0
128	279	0	28	64	115	1
108	282	0	23	67	125	1

Descriptive statistics for multiple variables

- Summarize the mean and standard deviation for each variable using “apply” function:

```
> bw_data <- read.table("birth_weight.txt", header=T)
> apply(bw_data, 2, mean)
       bwt   gestation      parity         age      height      weight       smoke
119.4625213 279.1013629   0.2623509  27.2282794  64.0494037 128.4787053   0.3909710

> apply(bw_data, 2, sd)
       bwt  gestation     parity        age     height     weight      smoke
18.3286714 16.0103051  0.4400999  5.8178387  2.5261015 20.7342822  0.4881759
```


cribioinfo@uic.edu

Groupwise summary statistics

“summaryBy” function from “doBy” package

- summaryBy(formula, data, FUN = mean, ...)
  - formula: a formula object indicating what variable to summarize and grouping variable, e.g. `bwt ~ smoke`
  - data: A data frame
  - FUN: A list of functions to be applied, e.g. `c(mean, sd, median)`

“table” function: build a contingency table of the counts

- table(..., dnn, ...)
  - … : list of factors
  - dnn: list of dimension names

Groupwise summary statistics

```
> head(bw_data)
  bwt gestation parity age height weight smoke
1 120       284      0  27     62    100     0
2 113       282      0  33     64    135     0
3 128       279      0  28     64    115     1
4 108       282      0  23     67    125     1
5 136       286      0  25     62     93     0
6 138       244      0  33     62    178     0

> library(doBy)
> summaryBy(bwt~smoke, data=bw_data, FUN=c(mean, sd, min, median, max))
  smoke bwt.mean   bwt.sd bwt.min bwt.median bwt.max
1     0 123.0853 17.42370      55        123     176
2     1 113.8192 18.29501      58        115     163
```

Build a contingency table of the counts with table()

```
> table(bw_data$parity, bw_data$smoke, dnn=c("parity", "smoke"))
      smoke
parity   0   1
     0 525 341
     1 190 118
```

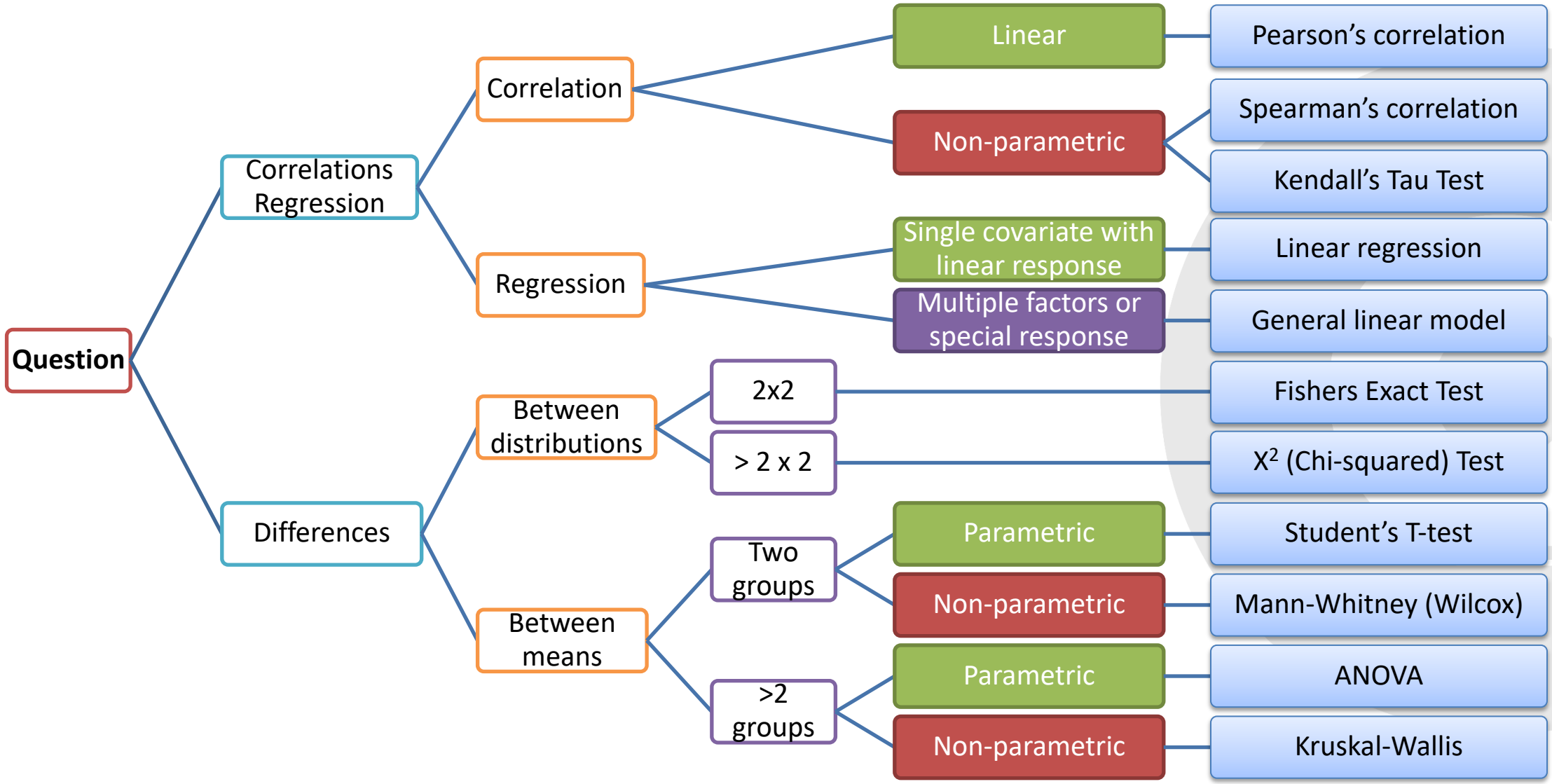

cribioinfo@uic.edu

Exercise 3.1: Descriptive statistics

Descriptive statistics for baby's birth weight dataset

cribioinfo@uic.edu

Introduction to different tests



cribioinfo@uic.edu

Tests for difference in means

Test	Number of groups	Assumption
T test	2	normal distribution
ANOVA	>=3	normal distribution
Wilcoxon Rank-Sum Test	2	None (Nonparametric)
Kruskal-Wallis test	>=3	None (Nonparametric)

Non-parametric tests generally have less power

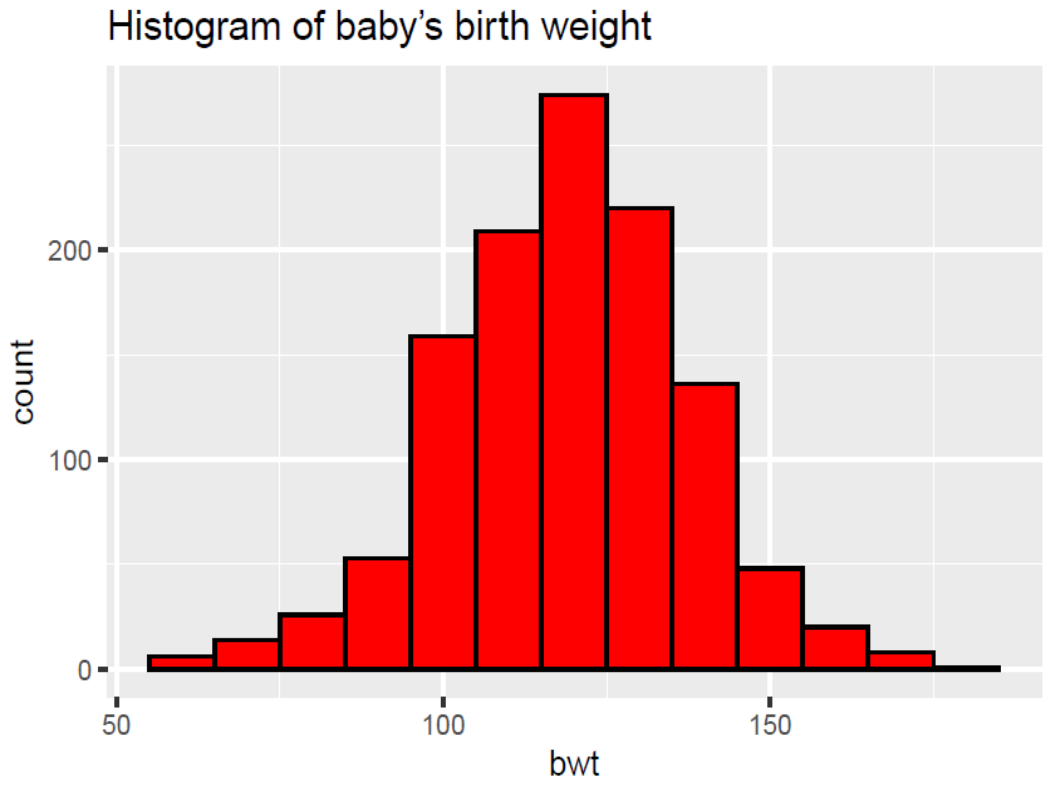
cribioinfo@uic.edu

ANOVA assumption: Normality

Use Shapiro test to check normality.

- Refer to both the p-value AND the W statistic
  - W varies from 0 to 1. 1 = best match to normal distribution.
- NOT: For large sample sizes, even W slightly less than 1 may be statistically significant, but the difference may be small enough that normality assumption is still OK

```
> shapiro.test(sample(bw_data$bwt, 20))

        Shapiro-Wilk normality test
data:  sample(bw_data$bwt, 20)
W = 0.95517, p-value = 0.4524
```



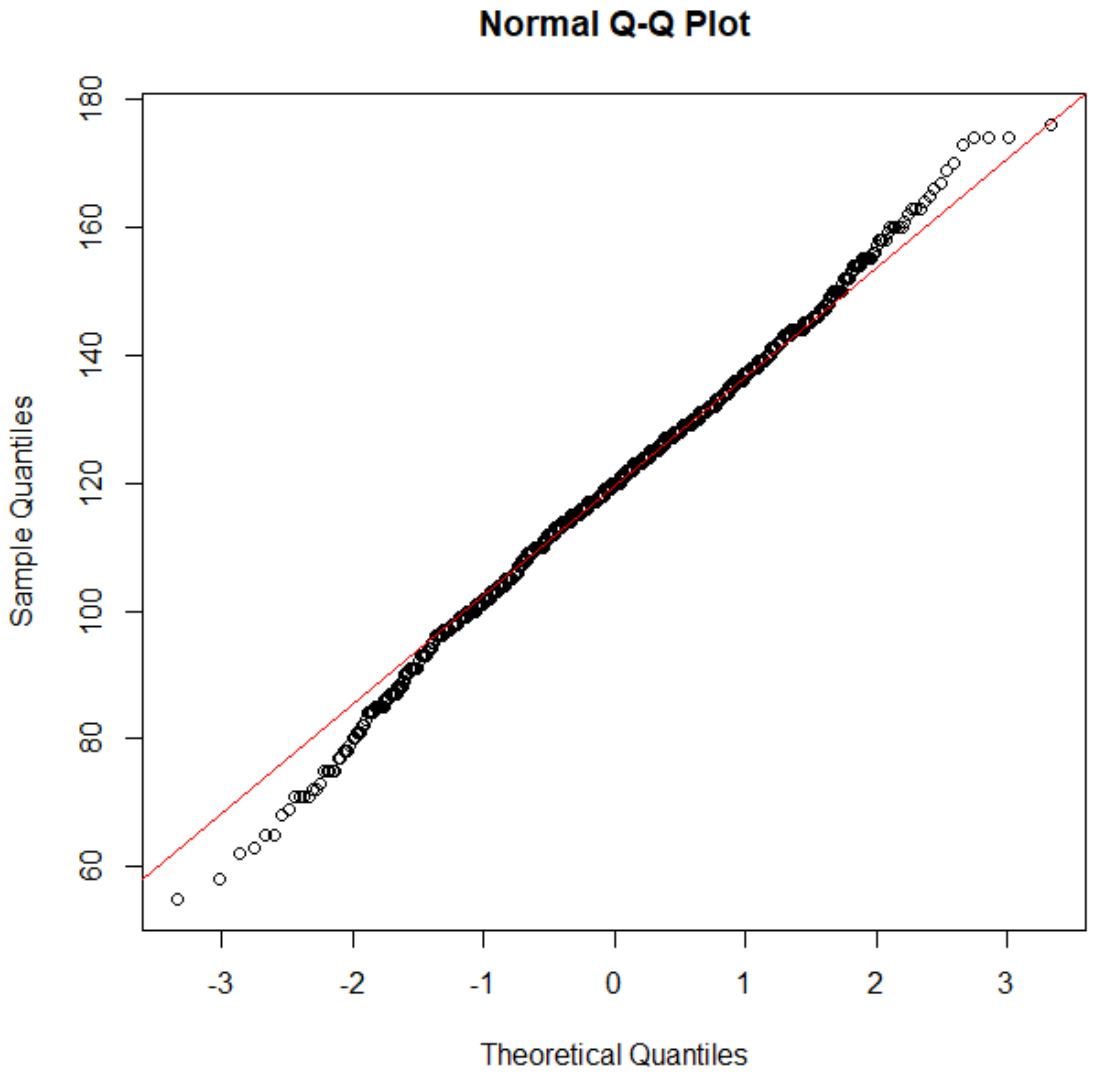
p-value is large and W close to 1 means we can't reject the hypothesis that baby's birth weights are normally distributed: **normality assumption is safe**

cribioinfo@uic.edu

ANOVA assumption: Normality

- The quantile-quantile (Q-Q) plot is a graphical technique for determining if two data sets come from populations with a common distribution.
- If the data is normally distributed, the points in the QQ-normal plot lie on a straight diagonal line.

```
> qqnorm(bw_data$bwt)
# qqline shows a line for a "theoretical"
# normal distribution
> qqline(bw_data$bwt, col="red")
```



cribioinfo@uic.edu

One way ANOVA

One way ANOVA: test for any differences between multiple groups

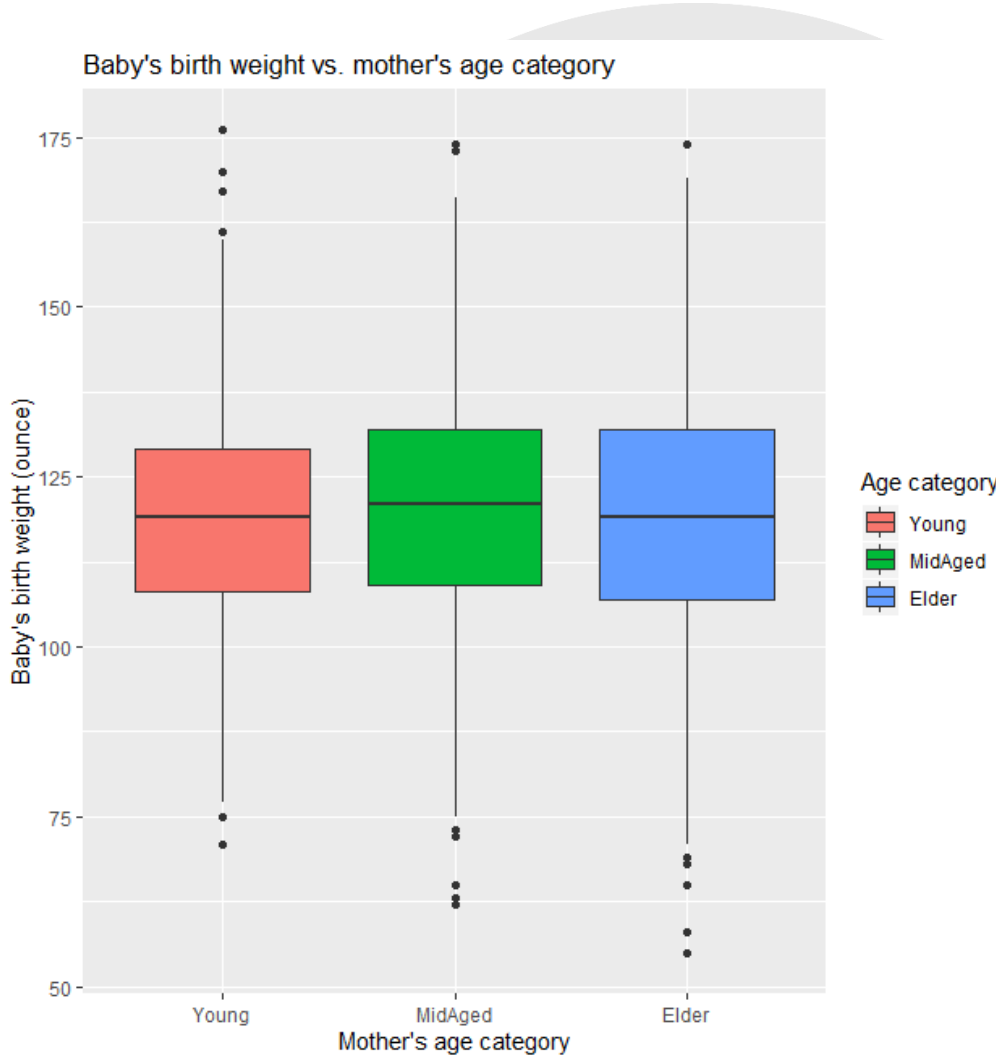
```
> head(bw_data)
  bwt gestation parity age height weight smoke   agecat
1 120       284      0  27     62    100     0 MidAged
2 113       282      0  33     64    135     0   Elder
3 128       279      0  28     64    115     1 MidAged
4 108       282      0  23     67    125     1   Young
5 136       286      0  25     62     93     0 MidAged
```

`aov()` : Fit ANOVA model based on formula

`summary()` : gives us p-values and other ANOVA statistics.

```
> summary(aov(bwt ~ agecat, data=bw_data))
              Df Sum Sq Mean Sq F value Pr(>F)
agecat         2    728   364.1   1.084  0.339
Residuals   1171 393330   335.9
```

p-value > 0.05 means the baby's birth weight has no significant difference among mother's three age categories.



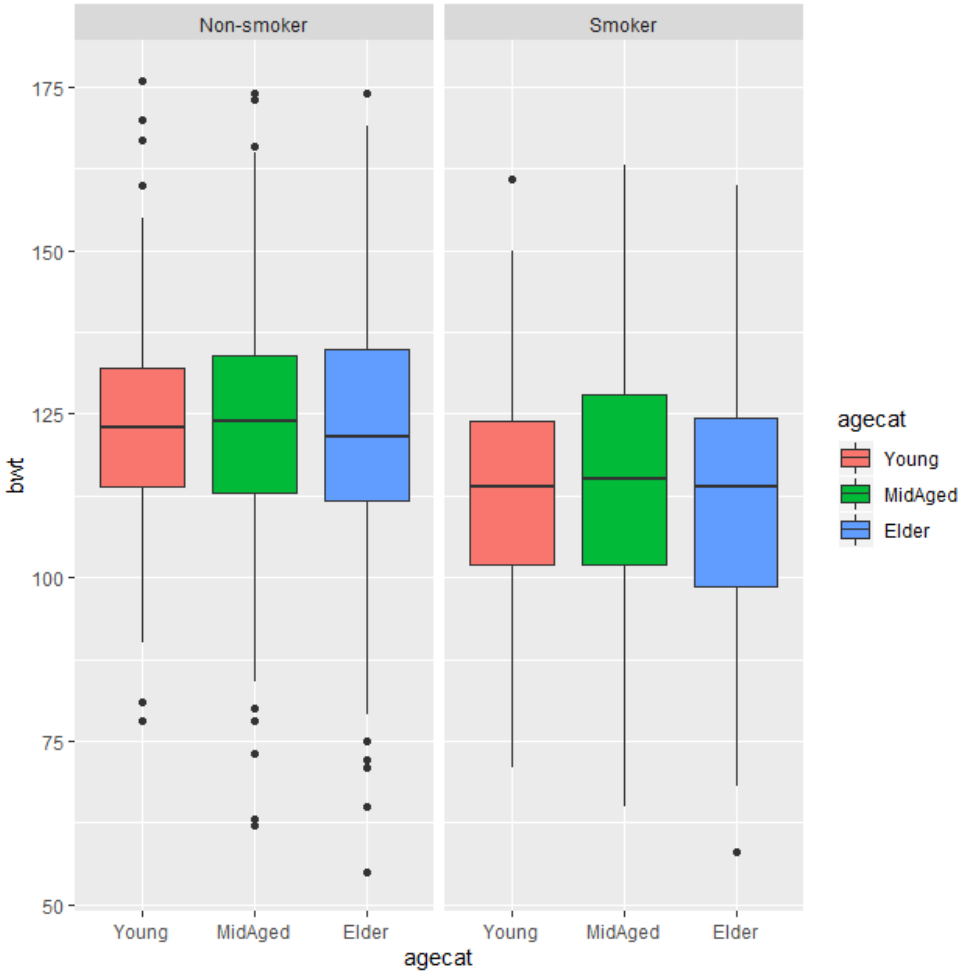
cribioinfo@uic.edu

Two-way ANOVA

Two-way ANOVA: test for differences between groups across two **factors** simultaneously

- E.g., Birth weight vs *age category* (factor 1) AND *smoking status* (factor 2) simultaneously
- We can also test for **interactions** between these factors
- Increases power in our analysis: we can model more sources of variability

```
> summary(aov(bwt ~ agecat + smoke, data=bw_data))
             Df  Sum Sq Mean Sq F value  Pr(>F)
agecat        2     728     364   1.153   0.316
smoke         1   23894   23894  75.671 <2e-16 ***
Residuals  1170  369436     316
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05
'.' 0.1 ' ' 1
```

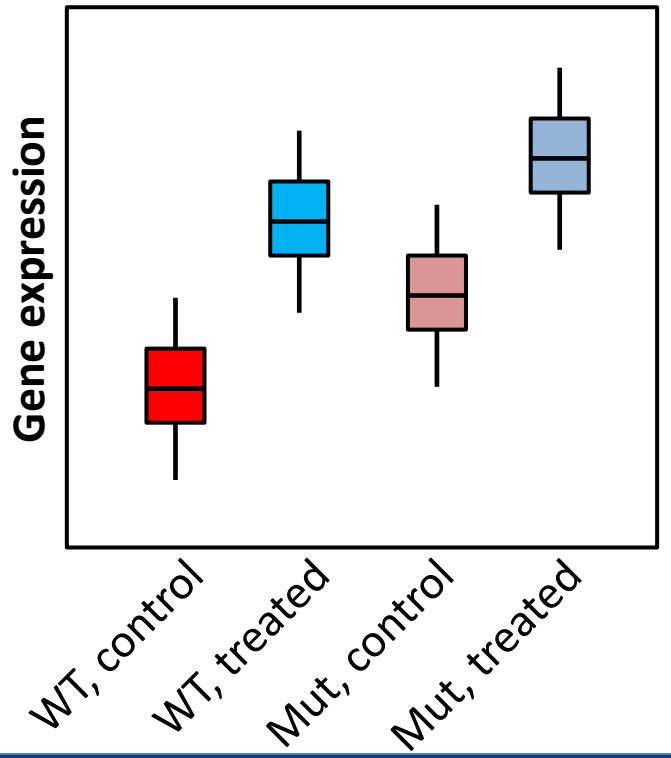


cribioinfo@uic.edu

ANOVA with interaction term(s)

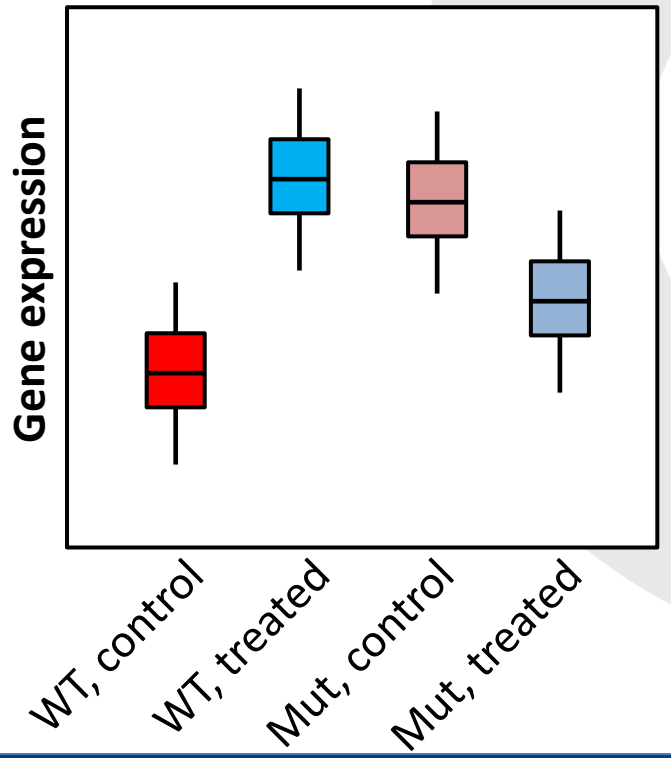
- Interaction terms capture effects where the effect of one factor depends on the *level* of another factor
- For example, modelling `gene_expression ~ treatment * genotype`
  - The effect of treatment depends on genotype, e.g., treatment makes the gene go up in WT, but go down in mutant

Example 1: no interaction



Expression increases with treatment and in mutant, effects are additive. The effects of treatment and genotype are independent

Example 2: interaction



Expression increases with treatment for WT, but decreases for Mut. Effects are dependent: to know the effect of treatment, we need to know the genotype.

cribioinfo@uic.edu

ANOVA with interaction term(s)

- Two way ANOVA with an interaction term

```
> summary(aov(bwt ~ agecat + smoke + agecat:smoke, data=bw_data))
              Df Sum Sq Mean Sq F value Pr(>F)
agecat         2    207     104   0.328 0.7202
smoke          1  23844   23844  75.599 <2e-16 ***
agecat:smoke   2   1620     810   2.568 0.0771 .
Residuals   1168 368387     315
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interaction term

- Alternatively, use full model formula:

```
> summary(aov(bwt ~ agecat*smoke, data=bw_data))
```

same as agecat + smoke + agecat:smoke

Full model: all factors and interaction term(s)

cribioinfo@uic.edu

Exercise 3.2: Check normality and ANOVA

Check normality

One way ANOVA

Two way ANOVA

Two way ANOVA with an interaction term

cribioinfo@uic.edu

BREAK

Please complete our workshop survey TODAY:

http://go.uic.edu/RICWorkshopSurvey

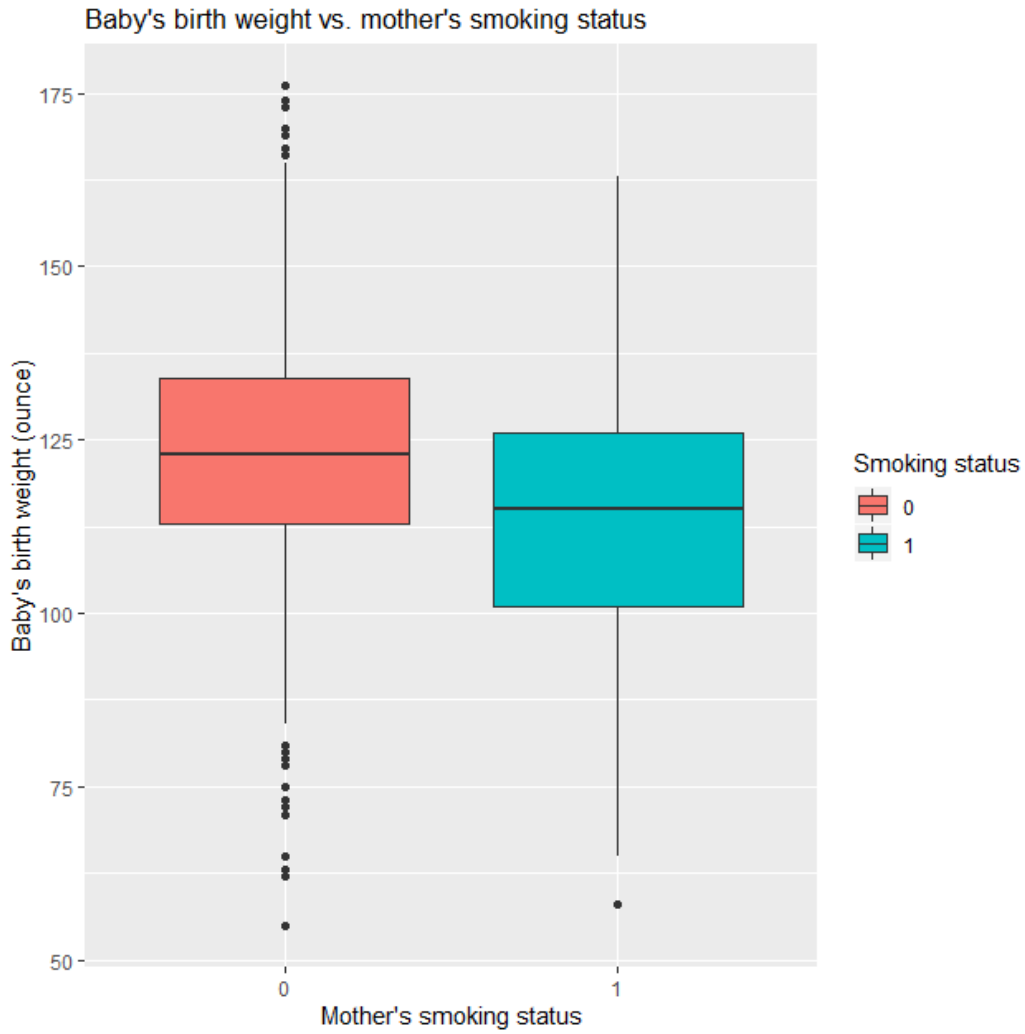
Wilcoxon rank sum test

- Wilcoxon rank sum test
  - Two groups
  - Nonparametric test:

    Do not assume normality (normal distribution)

```
> wilcox.test(bwt ~ smoke, data=bw_data)

        Wilcoxon rank sum test with continuity correction

data:  bwt by smoke

W = 212970, p-value < 2.2e-16

alternative hypothesis: true location shift is not equal to 0
```

p-value < 0.05 means the baby's birth weight is significant different between smoking mothers and non-smoking mothers.



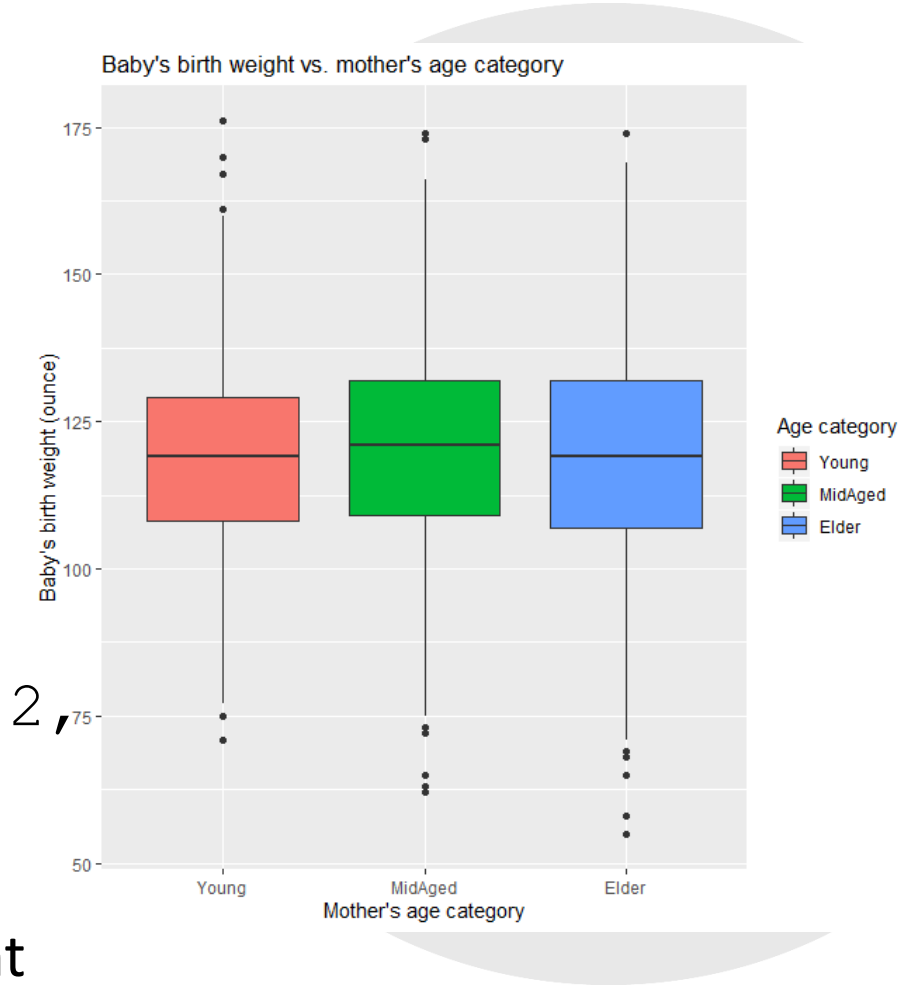
cribioinfo@uic.edu

Kruskal-Wallis test

- Kruskal-Wallis test
  - More than two groups
  - Non-parametric test:

    Do not assume normality (normal distribution)

```
> kruskal.test(bwt ~ agecat, data=bw_data)

        Kruskal-Wallis rank sum test

data:  bwt by agecat

Kruskal-Wallis chi-squared = 2.6045, df = 2,
p-value = 0.2719
```



p-value > 0.05 means the baby's birth weight has no significant difference among mother's three age categories.

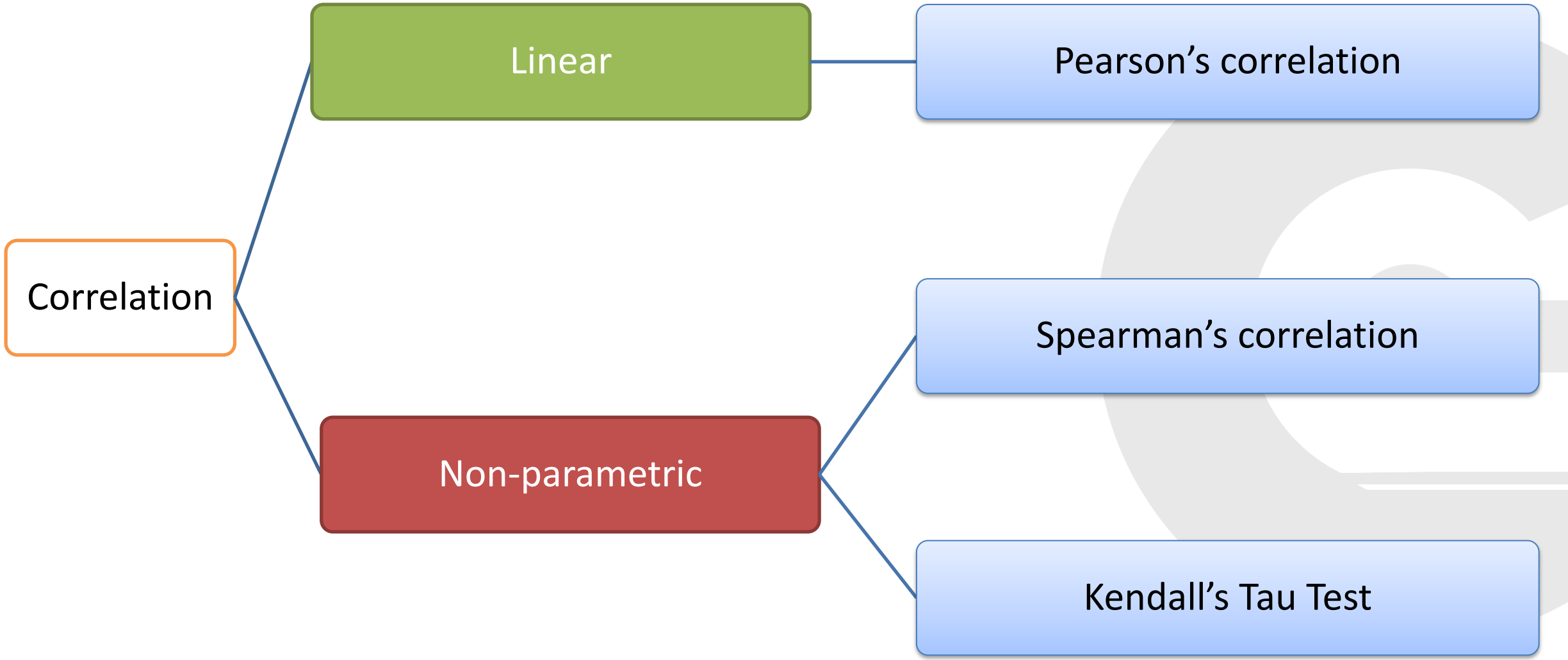
cribioinfo@uic.edu

Exercise 3.3: Wilcoxon rank sum test and Kruskal-Wallis test

Wilcoxon rank sum test

Kruskal-Wallis test

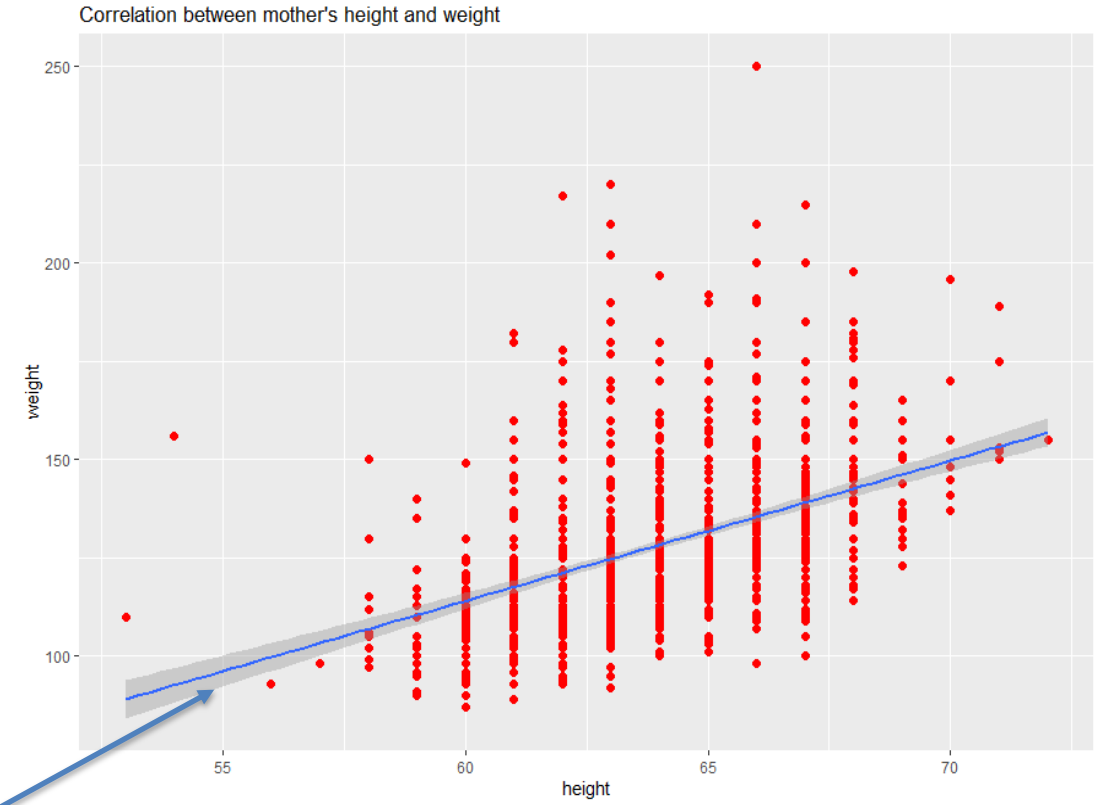
Correlation test



cribioinfo@uic.edu

Correlation test

- Correlation test: measure the strength of association between two variables and the direction of the relationship.
  - The value of the correlation coefficient varies between +1 and -1.
  - a + sign indicates a positive relationship.
  - a – sign indicates a negative relationship.



geom_smooth() in ggplot: add a smoothing line with confidence bands on the smooth , which helps to see what the trends look like

cribioinfo@uic.edu

Correlation test: Pearson

- Pearson correlation: linear relationship for two variables which are continuous in nature

```
> cor.test(bw_data$height, bw_data$weight)

        Pearson's product-moment correlation

data:  bw_data$height and bw_data$weight
t = 16.552, df = 1172, p-value < 2.2e-16
alternative hypothesis: true correlation is
not equal to 0
95 percent confidence interval:
 0.3877305 0.4805332
sample estimates:
      cor
0.4352874
```

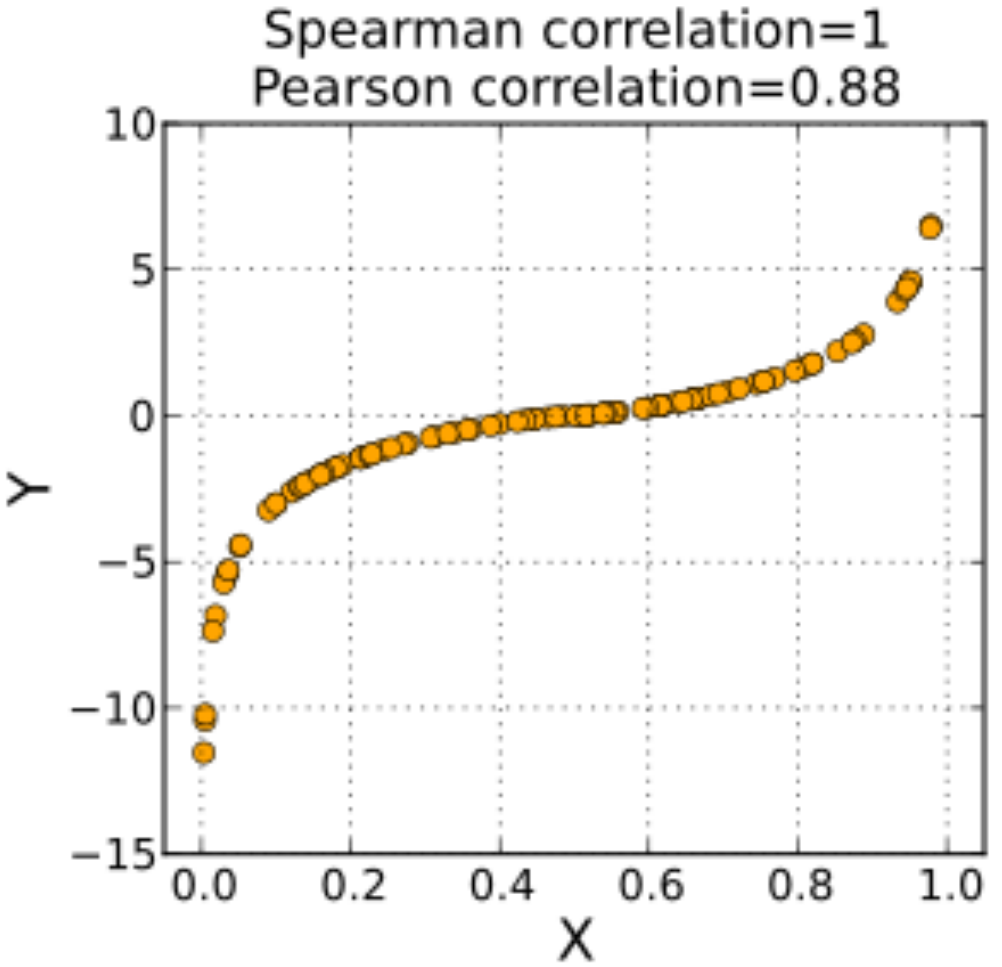
p-value < 0.05 means there is **significant correlation** between mother's height and weight.

correlation coefficient [-1, 1]: direction and degree of correlation

cribioinfo@uic.edu

Correlation test: Spearman

- Spearman correlation: measure of rank correlation
  - Intuitive definition: compute Pearson correlation on *rank-ordered* data
- A Spearman correlation of 1 results when the two variables being compared are **monotonically** related, even if their relationship is not linear.
  - *Monotonic: values increase and decrease together*
  - Same idea applies for Kendall



cribioinfo@uic.edu

Correlation test: Spearman

```
> cor.test(bw_data$height, bw_data$weight,
method="spearman")

        Spearman's rank correlation rho

data:  bw_data$height and bw_data$weight
S = 134713533, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.5004735
```


cribioinfo@uic.edu

Correlation test: Kendall

Kendall correlation: measure the ordinal association between two measured quantities. Calculation is based on concordant and discordant pairs.

- Concordant: $x_i > x_j$ and $y_i > y_j$ for any two pairs of data points (x_i, y_i) and (x_j, y_j)
- Discordant: $x_i > x_j$ and $y_i < y_j$
- Kendall's Tau $= \frac{|concordant| - |discordant|}{n(n-1)/2}$

```
> cor.test(bw_data$height, bw_data$weight, method="kendall")
        Kendall's rank correlation tau
data:  bw_data$height and bw_data$weight
z = 18.02, p-value < 2.2e-16
alternative hypothesis: true tau is not equal to 0
sample estimates:
      tau
0.3743995
```


cribioinfo@uic.edu

Kendall vs Spearman

- Both are non-parametric; Kendall is even "more" non-parametric (if that makes sense)
  - Spearman still computes differences (of rank values)
  - Kendall just compares higher/lower
  - Kendall *tends* to give coefficients closer to 0
- Spearman accuracy can be affected by ties, Kendall not affected
- Compute time: Kendall will scale quadratically with data set size. **Avoid using for large data sets.**

Run time in seconds

Data set size	Spearman	Kendall
500	<0.001	0.085
1000	0.001	0.34
2000	0.003	1.4
4000	0.004	5.5
8000	0.008	22

Exercise 3.4: Correlation test

Pearson correlation

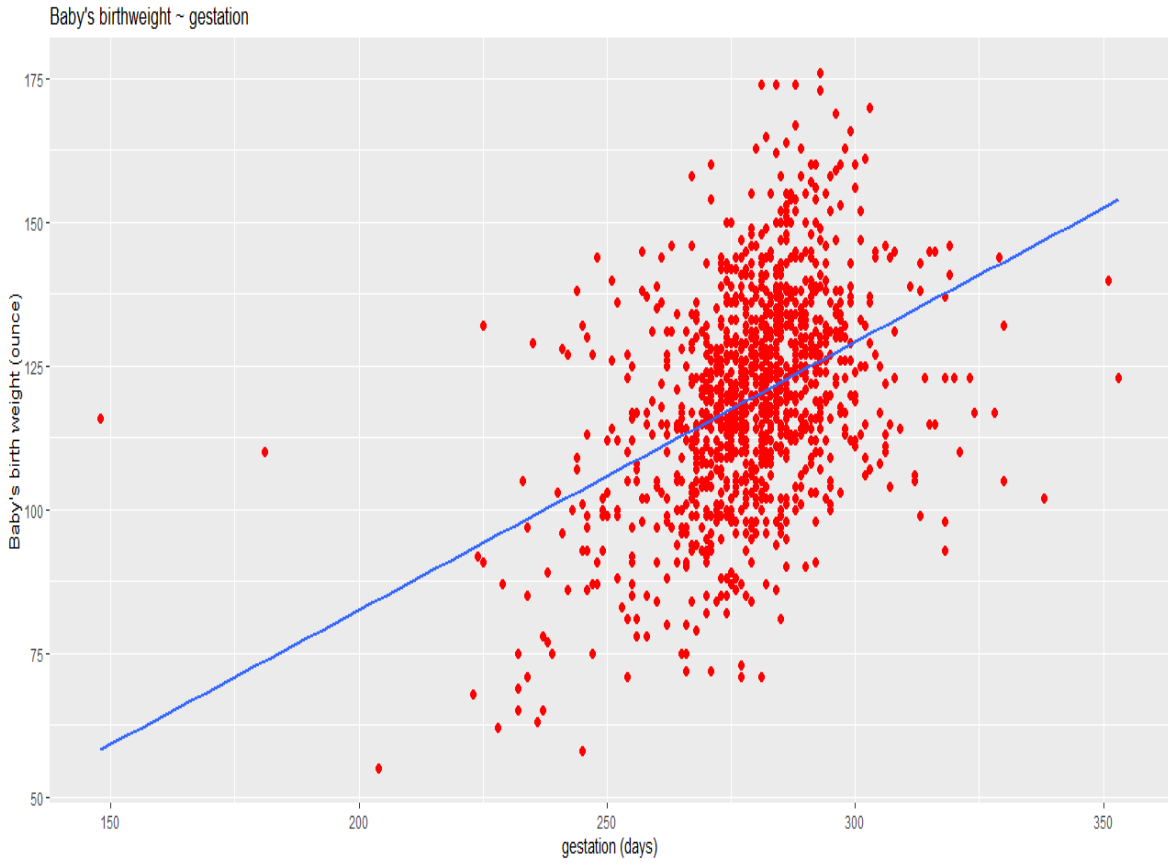
Spearman correlation

Kendall correlation

cribioinfo@uic.edu

Linear regression

- Linear Regression
  - Examine the relationship between quantitative variables x and y via a mathematical equation
  - Predict the value of a response variable (y) from the value of explanatory variables (x_1, x_2,…x_k).
  - Analyze the specific relationship between the explanatory variable and the response variable.



cribioinfo@uic.edu

Linear regression

```
> model <- lm(bwt ~ gestation, data=bw_data) # fit the regression model
> summary(model) # show regression coefficients table
Call:
lm(formula = bwt ~ gestation, data = bw_data)

Residuals:
    Min      1Q  Median      3Q     Max
-49.348 -11.065   0.218  10.101  57.704

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -10.75414    8.53693   -1.26    0.208
gestation     0.46656    0.03054   15.28   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.74 on 1172 degrees of freedom
Multiple R-squared:  0.1661,    Adjusted R-squared:  0.1654
F-statistic: 233.4 on 1 and 1172 DF,  p-value: < 2.2e-16
```

p-value tests if the regression coefficient is equal to zero.

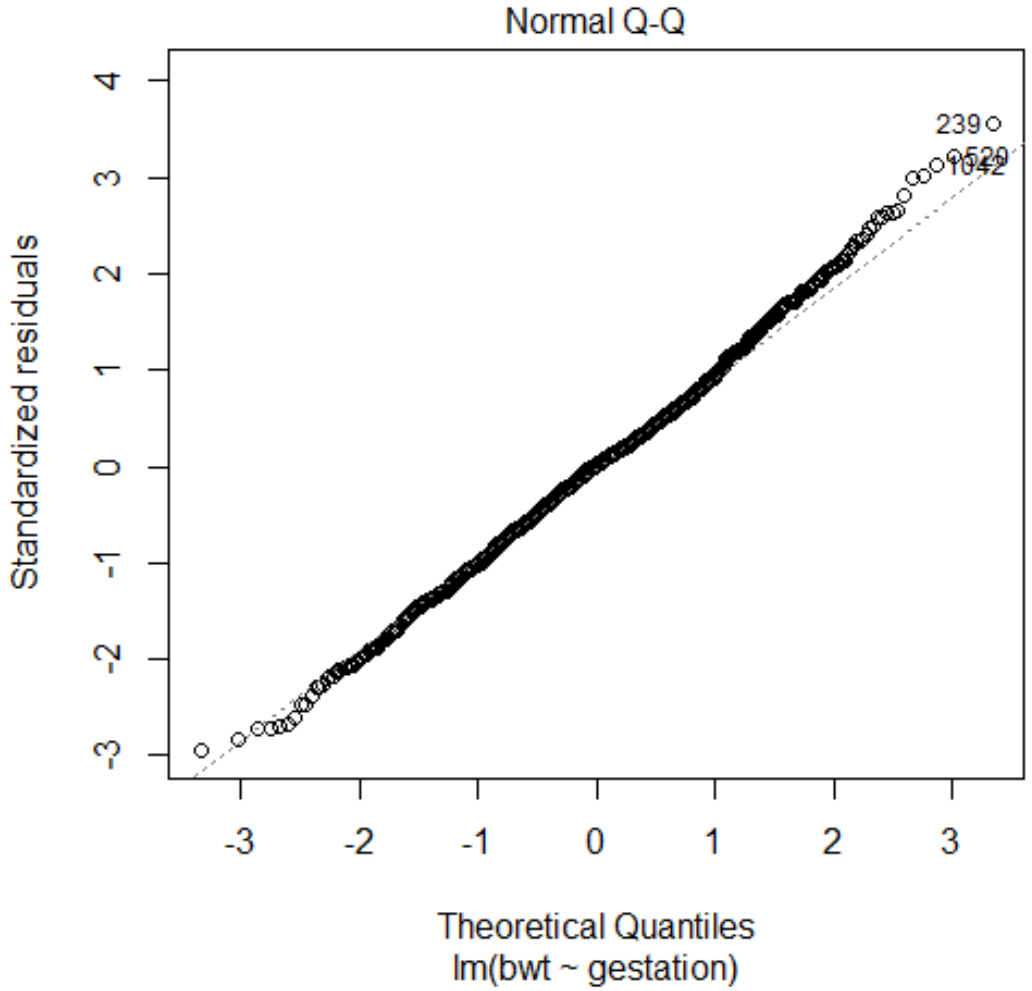
percentage of variation explained by the independent variable

cribioinfo@uic.edu

Q-Q plot

- Linear model fits and p-values assume residuals are normally distributed: check with Q-Q plot

```
# There will be four plots from
# plot(model). The 2nd one is Q-Q plot.
> plot(model, 2)
```



The Q-Q plot show that residuals are normally distributed, which means that the data meets normality assumptions of linear regression.

General linear model for multiple factors

General linear model: predict the value of the dependent variable based on the values of two or more independent variables (either numeric or categorical).

```
> model <- lm(bwt ~ gestation + weight + factor(smoke), data=bw_data)
> summary(model) # display results
Call:
lm(formula = bwt ~ gestation + weight + factor(smoke), data = bw_data)
Residuals:
    Min      1Q  Median      3Q     Max
-51.920 -10.759  -0.279   9.743  51.354
Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    -17.62648    8.69128  -2.028   0.0428 *
gestation        0.44809    0.02936  15.261  < 2e-16 ***
weight           0.11818    0.02267   5.213  2.2e-07 ***
factor(smoke)1  -8.07789    0.96444  -8.376  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.07 on 1170 degrees of freedom
Multiple R-squared:  0.2335,    Adjusted R-squared:  0.2315
F-statistic: 118.8 on 3 and 1170 DF,  p-value: < 2.2e-16
```


cribioinfo@uic.edu

Exercise 3.5: linear regression

Simple linear model

Scatter plot for mother's bwt vs gestation

Q-Q plot

General linear model for multiple factors

Association test of categorical variables

Association test of categorical variables, i.e., whether the variables are independent or related.

- Fisher's Exact Test (introduced in previous R workshop)
  - Usually for test of association in a 2 × 2 table.
  - Most useful when the total sample size and the expected values are small.

- Chi-squared test
  - A test for association in a $m \times n$ table.
  - Not accurate when the sample size is small
    - Parametric test: assumptions break down when there less than ~5 counts for any cell in the table
    - In this case, you can use a permutation test to compute fair p-values

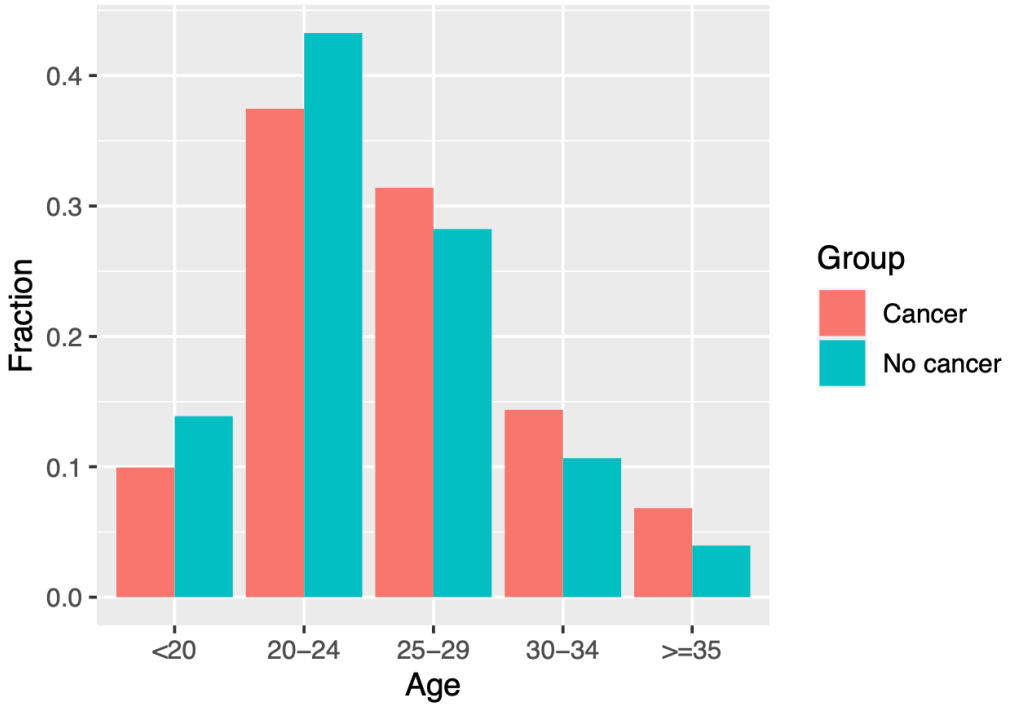
Chi-squared test

Is there a significant association between women's age at first birth and breast cancer status?

		women's age at first birth				
		<20	20-24	25-29	30-34	>=35
breast cancer status	Cancer	320	1206	1011	463	220
	No cancer	1422	4432	2893	1092	406

```
> chisq.test(age_cancer)

        Pearson's Chi-squared test

data:  age_cancer
X-squared = 130.34, df = 4, p-value < 2.2e-16
```

Post-hoc option: follow-up with Fisher's Exact test for individual group comparisons.



p-value < 0.05 means there is **significant association** between women's age at first birth and breast cancer.

cribioinfo@uic.edu

Exercise 3.6: Chi-squared test

Chi-squared test

cribioinfo@uic.edu

Summary

- Data manipulation/Data cleaning
- ggplot2
  - scatter plot
  - boxplot
  - barplot
- Statistics
  - Statistical tests
  - Linear regression

THANK YOU!

Please complete our workshop survey TODAY:

http://go.uic.edu/RICWorkshopSurvey

Advanced Data Import/Export

- Importing/writing data to Excel Spreadsheet
- Reading data from SPSS, SAS, or Stata

Excel spreadsheets

- Two packages are available
  - `readxl` – Import data from Excel using embedded C library
    - No additional dependencies required
    - Used by “Import Dataset” feature in Rstudio
    - Data are returned as “tibble” a data.frame like object.
    - More information at https://readxl.tidyverse.org/
  - `openxlsx` – Read, write, and edit Excel spreadsheets
    - Can write data frames to Excel file with formatting and styles (colors, font size, borders)
    - Can load a workbook, edit, and then save
    - When using read functions, data are returned as data.frame
    - More information at https://ycphs.github.io/openxlsx/

Reading an Excel spreadsheet - readxl

- Reading the first sheet

```
> library(readxl)
> my_sheet <- read_excel("my_spreadsheet.xlsx", sheet=1)
```

- Read the second sheet

```
> my_sheet <- read_excel("my_spreadsheet.xlsx", sheet=2)
```

- Read the sheet named "Sheet1"

```
> my_sheet <- read_excel("my_spreadsheet.xlsx", sheet="Sheet1")
```

- Retrieve a list of sheets in a excel file.

```
> sheet_names <- excel_sheets("my_spreadsheet.xlsx")
```

Reading an Excel spreadsheet - openxlsx

- Reading the first sheet

```
> library(openxlsx)
> my_sheet <- read.xlsx("my_spreadsheet.xlsx", 1)
```

- Read the second sheet

```
> my_sheet <- read.xlsx("my_spreadsheet.xlsx", 2)
```

- Read the sheet named "Sheet1"

```
> my_sheet <- read.xlsx("my_spreadsheet.xlsx",
                        sheet="Sheet1")
```

- Retrieve a list of sheets in a excel file.

```
> wb <- loadWorkbook("my_spreadsheet.xlsx")
> sheet_names <- names(wb)
```

Writing a data.frame to an Excel spreadsheet - openxlsx

- Load the library and create a workbook with two sheets

```
> library(openxlsx)
> wb <- createWorkbook()
> addWorksheet(wb, “Data Frame 1”)
> addWorksheet(wb, “Data Frame 2”)
```

- Write the data for each sheet to the workbook

```
> writeData(wb, sheet=“Data Frame 1”, x=data_frame_1)
> writeData(wb, sheet=“Data Frame 2”, x=data_frame_2)
```

- Save the workbook to an XLSX file

```
> saveWorkbook(wb,“my_new_spreadsheet.xlsx”)
```

Exercise 3.7: Read/Write data from Excel spreadsheet

Read using readxl

Read/Write using openxlsx

Reading data from SPSS, SAS or Stata files

- `haven` library enables R to read and write various data formats used by other statistical packages using an embedded C library (no additional dependencies)
  - SAS: `read_sas()` reads .sas7bdat + .sas7bcat files and `read_xpt()` reads SAS transport files (version 5 and version 8).
  - SPSS: `read_sav()` reads .sav files and `read_por()` reads the older .por files. `write_sav()` writes .sav files.
  - Stata: `read_dta()` reads .dta files (up to version 15). `write_dta()` writes .dta files (versions 8-15).
- Like `readxl`, data are returned as “tibble” a data.frame like object.
- More information available at https://haven.tidyverse.org/

Examples of reading SPSS, SAS or Stata files

- Read SPSS file

```
> library(haven)
> my_table <- read_sav("mtcars.sas7bdat")
```

- Read SAS file

```
> my_table <- read_sas("mtcars.sav")
```

- Read Stata file

```
> my_table <- read_dta("mtcars.dta")
```

THANK YOU!

Please complete our workshop survey TODAY:

http://go.uic.edu/RICWorkshopSurvey