Introduction to R/Tidyverse

Matthew Galbraith
Linda Crnic Institute for Down Syndrome

Data Science for Developing Scholars in Down Syndrome Research (DS3) 2025

INCLUDE
INvestigation of Co-occurring conditions across the Lifespan to Understand Down syndromE

Linda Crnic Institute for Down Syndrome
UNIVERSITY OF COLORADO **ANSCHUTZ MEDICAL CAMPUS**

GLOBAL
DOWN SYNDROME FOUNDATION®

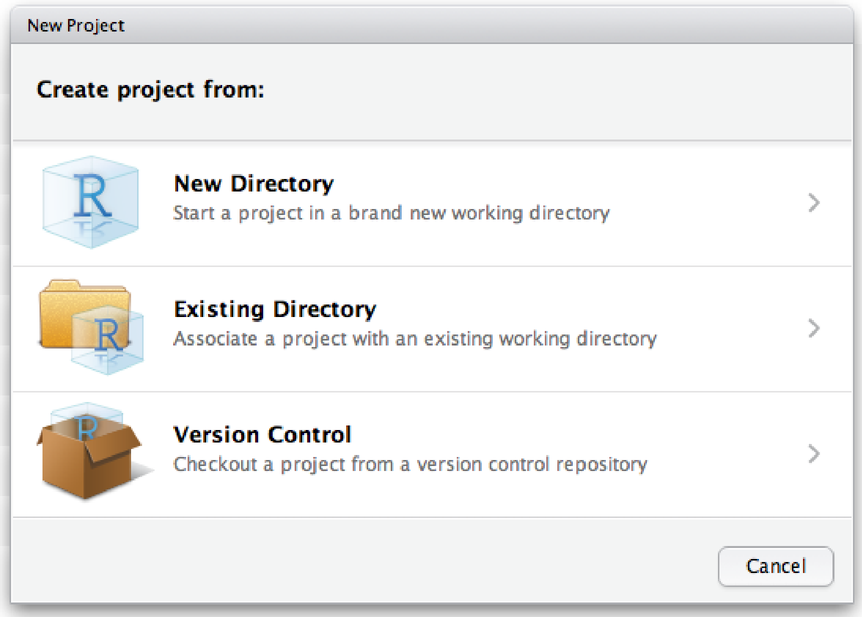
Links for this session

https://support.rstudio.com/hc/en-us/articles/200526207-Using-RStudio-Projects
https://r4ds.had.co.nz/workflow-projects.html
https://github.com/DS3-2025/Rproject_template

https://tidyverse.tidyverse.org/
https://r4ds.hadley.nz/
https://github.com/DS3-2025/tidy_data_exercise

https://ggplot2.tidyverse.org/
https://ggplot2-book.org/

RStudio Projects


New Project

Create project from:

New Directory
Start a project in a brand new working directory

Existing Directory
Associate a project with an existing working directory

Version Control
Checkout a project from a version control repository

Cancel

- Open existing projects via .Rproj file
- Automatically sets your working directory
- Self-contained set of directories, scripts, and data files (very important for multiple projects)

Organizing your Rstudio Projects

- Only **/data** and R scripts are required - everything else can be recreated (incl. earlier versions)
- Treat **/data** directory as read-only
- Analysis outputs go to **/results** or **/plots** (with version info)
- R workspace and large intermediate files stored in **/rdata**
- Additional directories added as needed, eg /Archive
- Compatible with manual or other version control

```
Project_directory
├ /data
├ /results
├ /plots
├ /rdata
├ analysis_script.R
├ helper_functions.R
├ project.Rproj
```

https://support.rstudio.com/hc/en-us/articles/200526207-Using-RStudio-Projects
https://r4ds.had.co.nz/workflow-projects.html
https://github.com/DS3-2025/Rproject_template

Reproducible data analysis: Package management using renv

renv

- `install.packages("renv")`
- `renv::init()` to initialize a new project-local environment with a private R library
- `renv::install()` to install packages after initialization
- `renv::snapshot()` to save the state of your project to renv.lock
- `renv::restore()` to restore the state of your project from renv.lock

```
Project_directory
├ /data
├ /results
├ /plots
├ /rdata
├ /renv
├ renv.lock
├ helper_functions.R
├ analysis_script.R
├ project.Rproj
```

Allows for fully self-contained R projects
(Usually) takes care of installing packages

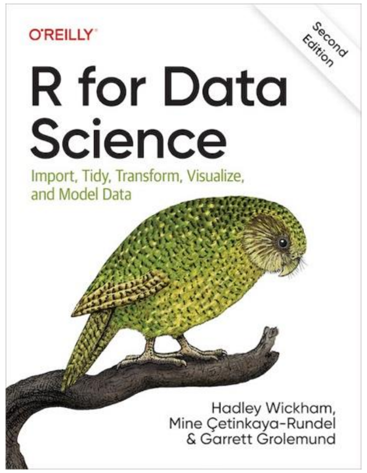
https://rstudio.github.io/renv/articles/renv.html
https://rstudio.github.io/renv/articles/bioconductor.html

Into the Tidyverse

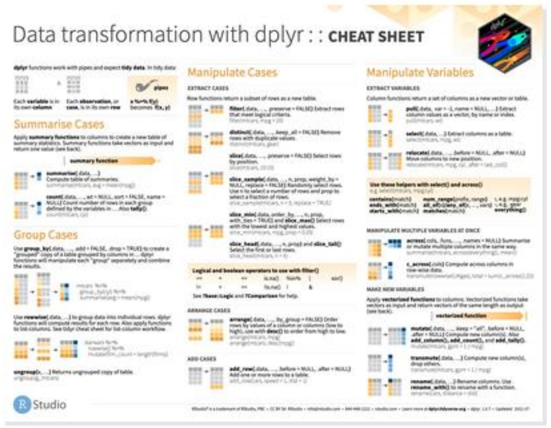
tidyverse

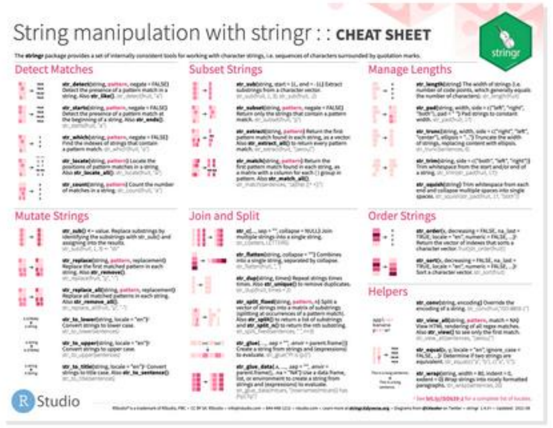
https://tidyverse.tidyverse.org/
Base R: `install.packages("tidyverse")`
renv: `renv::install("tidyverse")`

Core tidyverse packages:
ggplot2, for data visualisation.
dplyr, for data manipulation.
tidyr, for data tidying.
readr, for data import.
purrr, for functional programming.
tibble, for tibbles, a modern re-imagining of data frames.
stringr, for strings.
forcats, for factors.
lubridate, for date/times.


O'REILLY
Second Edition
R for Data Science
Import, Tidy, Transform, Visualize, and Model Data
Hadley Wickham, Mine Çetinkaya-Rundel & Garrett Grolemund

https://r4ds.hadley.nz/


Data transformation with dplyr : : CHEAT SHEET


String manipulation with stringr : : CHEAT SHEET

https://posit.co/resources/cheatsheets/

Into the Tidyverse: Pipes

Pipe operator `%>%`

- Avoids nesting
- Minimizes need to create objects and functions
- Structure sequences of operations left-to-right or **top-to-bottom**
- Easy to inspect and add steps anywhere

- `x %>% f` is equivalent to `f(x)`
- `x %>% f(y)` is equivalent to `f(x, y)`
- `x %>% f %>% g %>% h` is equivalent to `h(g(f(x)))`

- Keyboard shortcut in RStudio: cmd/ctrl+shift+m

- Note: R 4.1.0 introduced a native pipe operator `|>` with some minor differences
  https://www.tidyverse.org/blog/2023/04/base-vs-magrittr-pipe/

https://r4ds.had.co.nz/pipes.html

Into the Tidyverse: syntax

Base R "dollar sign" syntax

Example - summary statistics:

one continuous variable:

```
mean(mtcars$mpg)
```

one categorical variable:

```
table(mtcars$cyl)
```

two categorical variables:

```
table(mtcars$cyl, mtcars$am)
```

one continuous, one categorical:

```
mean(mtcars$mpg[mtcars$cyl==4])
mean(mtcars$mpg[mtcars$cyl==6])
mean(mtcars$mpg[mtcars$cyl==8])
```

Tidyverse syntax

Example - summary statistics:

one continuous variable:

```
mtcars %>% dplyr::summarize(mean(mpg))
```

one categorical variable:

```
mtcars %>%
    dplyr::group_by(cyl)%>%
    dplyr::summarize(n())
```

two categorical variables:

```
mtcars %>%
    dplyr::group_by(cyl, am) %>%
    dplyr::summarize(n())
```

one continuous, one categorical:

```
mtcars %>%
    dplyr::group_by(cyl)%>%
    dplyr::summarize(mean(mpg))
```

https://rstudio.github.io/cheatsheets/base-r.pdf
https://rstudio.github.io/cheatsheets/syntax.pdf

Into the Tidyverse

Z-score calculation with base R:

```
x <- sweep(sweep(t(dat), 1,
apply(t(dat),1,mean,na.rm=T), FUN = "-"),
1, apply(t(dat),1,sd,na.rm=T), FUN = "/")
```

- hard to decipher (learning barrier)
- have to enter target object name in several places

Z-score calc with tidyverse + scale():

```
zscores <- dat |>
        select(LabID, Analyte, Value) |>
        pivot_wider() |>
        scale()
```

- Somewhat easier to decipher, but not obvious that this calculates Z-scores, even looking at `?scale` defaults (center = TRUE, scale = TRUE)

Manual Z-score calc with tidyverse:

```
zscores <- dat |>
        select(LabID, Analyte, Value) |>
        group_by(Analyte) |>
        mutate(
         zscore = (Value - mean(Value, na.rm = TRUE)) / sd(Value, na.rm = TRUE)
        ) |>
        ungroup()
```

- Naming of new variable
- Easier to see how calculation was performed
- Easy to keep both original and transformed values for comparison

Into the Tidyverse

Z-score calculation with base R:

```
x <- sweep(sweep(t(dat), 1,
apply(t(dat),1,mean,na.rm=T), FUN = "-"),
1, apply(t(dat),1,sd,na.rm=T), FUN = "/")
```

- hard to decipher (learning barrier)
- have to enter target object name in several places

Even more verbose tidyverse version:

```
zscores <- dat |>
       select(LabID, Analyte, Value) |>
       group_by(Analyte) |>
       mutate(
        mean = mean(Value, na.rm = TRUE),
        sd = sd(Value, na.rm = TRUE),
        zscore = (Value - mean) / sd
       ) |>
       ungroup()
```

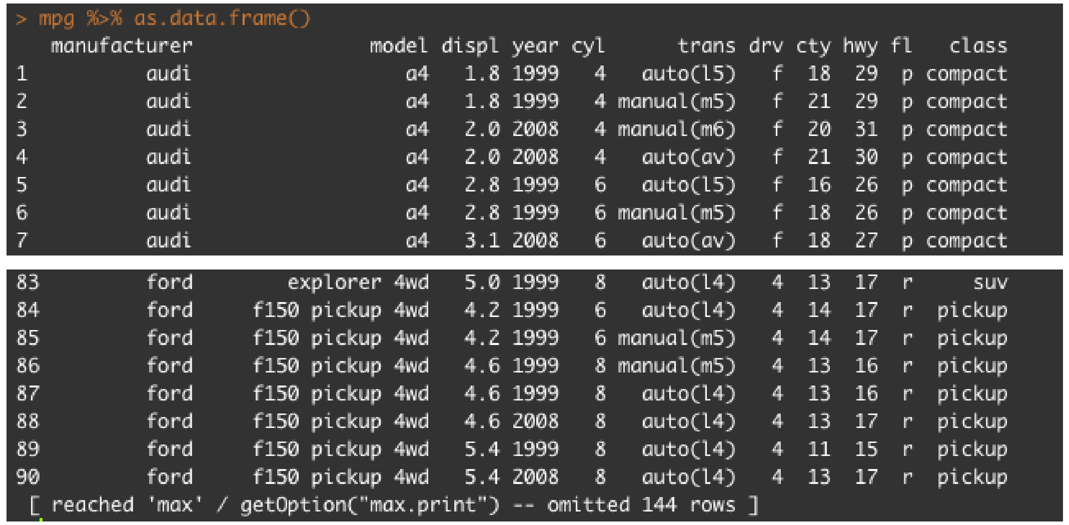
- Naming of new variable
- Easier to see how calculation was performed
- Easy to keep both original and transformed values for comparison

Into the Tidyverse: important packages

tibble

Tibbles = enhanced data frames

- Easier preview of data
- Concise summary information including data types



```
> mpg %>% as.data.frame()
   manufacturer                model displ year cyl      trans drv cty hwy fl     class
1          audi                   a4   1.8 1999   4   auto(l5)   f  18  29  p   compact
2          audi                   a4   1.8 1999   4 manual(m5)   f  21  29  p   compact
3          audi                   a4   2.0 2008   4 manual(m6)   f  20  31  p   compact
4          audi                   a4   2.0 2008   4   auto(av)   f  21  30  p   compact
5          audi                   a4   2.8 1999   6   auto(l5)   f  16  26  p   compact
6          audi                   a4   2.8 1999   6 manual(m5)   f  18  26  p   compact
7          audi                   a4   3.1 2008   6   auto(av)   f  18  27  p   compact

83         ford         explorer 4wd   5.0 1999   8   auto(l4)   4  13  17  r       suv
84         ford      f150 pickup 4wd   4.2 1999   6   auto(l4)   4  14  17  r    pickup
85         ford      f150 pickup 4wd   4.2 1999   6 manual(m5)   4  14  17  r    pickup
86         ford      f150 pickup 4wd   4.6 1999   8 manual(m5)   4  13  16  r    pickup
87         ford      f150 pickup 4wd   4.6 1999   8   auto(l4)   4  13  16  r    pickup
88         ford      f150 pickup 4wd   4.6 2008   8   auto(l4)   4  13  17  r    pickup
89         ford      f150 pickup 4wd   5.4 1999   8   auto(l4)   4  11  15  r    pickup
90         ford      f150 pickup 4wd   5.4 2008   8   auto(l4)   4  13  17  r    pickup
 [ reached 'max' / getOption("max.print") -- omitted 144 rows ]
```

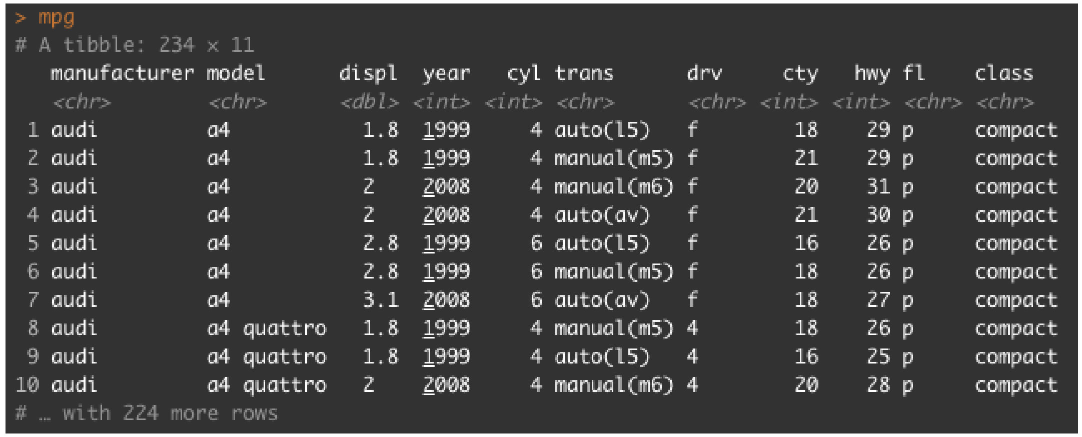
vs.

readr

readxl

Importing delimited data

- Easy reading in of data from .txt, .csv, .tsv, .xlsx
- Guessing of column types
- Will not convert strings
- Imported as tibble



```
> mpg
# A tibble: 234 x 11
   manufacturer model      displ  year   cyl trans      drv     cty   hwy fl    class  
   <chr>        <chr>      <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>  
 1 audi         a4           1.8  1999     4 auto(l5)   f        18    29 p     compact
 2 audi         a4           1.8  1999     4 manual(m5) f        21    29 p     compact
 3 audi         a4           2    2008     4 manual(m6) f        20    31 p     compact
 4 audi         a4           2    2008     4 auto(av)   f        21    30 p     compact
 5 audi         a4           2.8  1999     6 auto(l5)   f        16    26 p     compact
 6 audi         a4           2.8  1999     6 manual(m5) f        18    26 p     compact
 7 audi         a4           3.1  2008     6 auto(av)   f        18    27 p     compact
 8 audi         a4 quattro   1.8  1999     4 manual(m5) 4        18    26 p     compact
 9 audi         a4 quattro   1.8  1999     4 auto(l5)   4        16    25 p     compact
10 audi         a4 quattro   2    2008     4 manual(m6) 4        20    28 p     compact
# … with 224 more rows
```

Into the Tidyverse: important packages

dplyr

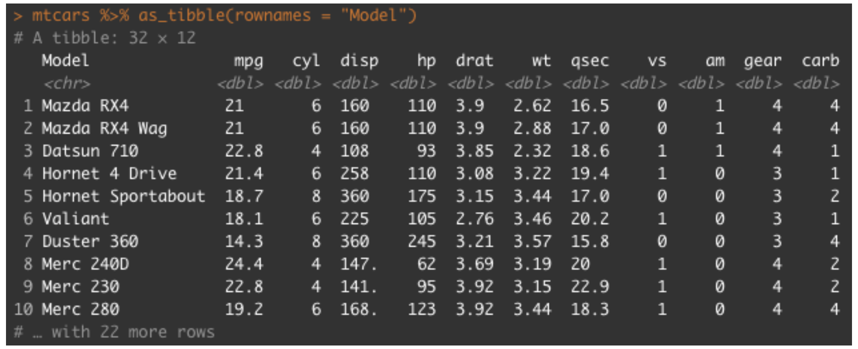
Data manipulation

- `mutate()` adds new variables that are functions of existing variables.
- `select()` picks variables based on their names.
- `filter()` picks rows based on their values.
- `summarize()` reduces multiple values down to a single summary.
- `arrange()` changes the ordering of the rows.
- `group_by()` perform group-wise operations.

tidyr

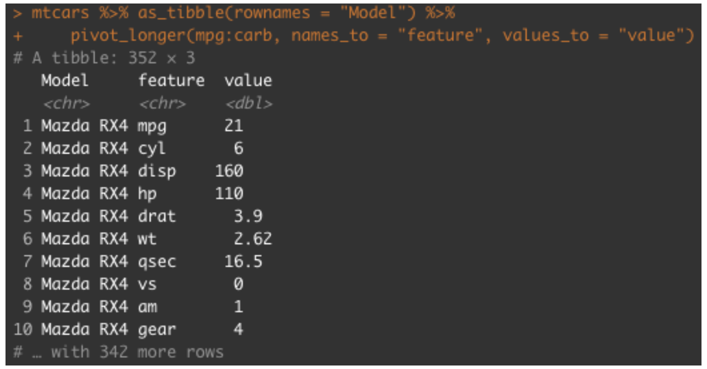
Reshaping data

- Conversion to/from Tidy data where each column is a variable and each row is an observation.
- `pivot_longer()` converts to Tidy/long format.
- `pivot_wider()` converts to wide format.
- `tibble::column_to_rownames(var = “id_col”)` converts to data frame (required for some functions).



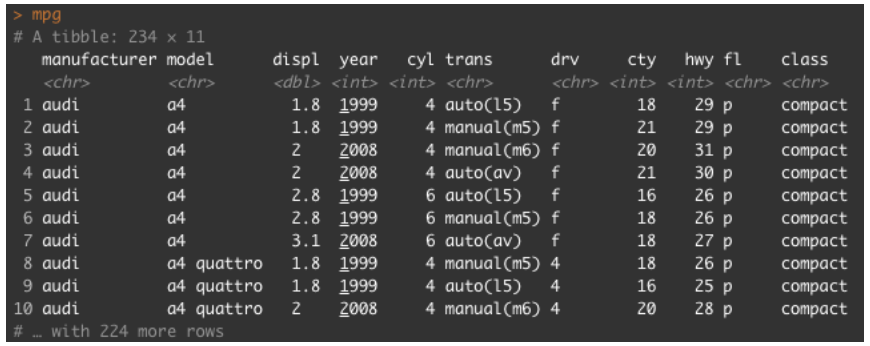
```
> mtcars %>% as_tibble(rownames = "Model")
# A tibble: 32 × 12
   Model               mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
   <chr>             <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
 1 Mazda RX4          21       6  160    110  3.9   2.62  16.5     0     1     4     4
 2 Mazda RX4 Wag      21       6  160    110  3.9   2.88  17.0     0     1     4     4
 3 Datsun 710         22.8     4  108     93  3.85  2.32  18.6     1     1     4     1
 4 Hornet 4 Drive     21.4     6  258    110  3.08  3.22  19.4     1     0     3     1
 5 Hornet Sportabout  18.7     8  360    175  3.15  3.44  17.0     0     0     3     2
 6 Valiant            18.1     6  225    105  2.76  3.46  20.2     1     0     3     1
 7 Duster 360         14.3     8  360    245  3.21  3.57  15.8     0     0     3     4
 8 Merc 240D          24.4     4  147.    62  3.69  3.19  20       1     0     4     2
 9 Merc 230           22.8     4  141.    95  3.92  3.15  22.9     1     0     4     2
10 Merc 280           19.2     6  168.   123  3.92  3.44  18.3     1     0     4     4
# … with 22 more rows
```

vs.



```
> mtcars %>% as_tibble(rownames = "Model") %>%
+     pivot_longer(mpg:carb, names_to = "feature", values_to = "value")
# A tibble: 352 × 3
   Model     feature  value
   <chr>     <chr>    <dbl>
 1 Mazda RX4 mpg      21
 2 Mazda RX4 cyl       6
 3 Mazda RX4 disp    160
 4 Mazda RX4 hp      110
 5 Mazda RX4 drat      3.9
 6 Mazda RX4 wt        2.62
 7 Mazda RX4 qsec     16.5
 8 Mazda RX4 vs        0
 9 Mazda RX4 am        1
10 Mazda RX4 gear      4
# … with 342 more rows
```

+ add additional variables



```
> mpg
# A tibble: 234 × 11
   manufacturer model      displ  year   cyl trans      drv     cty   hwy fl    class
   <chr>        <chr>      <dbl> <int> <int> <chr>      <chr> <int> <int> <chr> <chr>
 1 audi         a4           1.8  1999     4 auto(l5)   f        18    29 p     compact
 2 audi         a4           1.8  1999     4 manual(m5) f        21    29 p     compact
 3 audi         a4           2    2008     4 manual(m6) f        20    31 p     compact
 4 audi         a4           2    2008     4 auto(av)   f        21    30 p     compact
 5 audi         a4           2.8  1999     6 auto(l5)   f        16    26 p     compact
 6 audi         a4           2.8  1999     6 manual(m5) f        18    26 p     compact
 7 audi         a4           3.1  2008     6 auto(av)   f        18    27 p     compact
 8 audi         a4 quattro   1.8  1999     4 manual(m5) 4        18    26 p     compact
 9 audi         a4 quattro   1.8  1999     4 auto(l5)   4        16    25 p     compact
10 audi         a4 quattro   2    2008     4 manual(m6) 4        20    28 p     compact
# … with 224 more rows
```

Into the Tidyverse: important packages

stringr

Character string manipulations

- `str_detect(x, pattern)` looks for match to the pattern; commonly used with `dplyr::filter()`
- `str_extract(x, pattern)` extracts the text of the match; commonly used with `dplyr::mutate()`
- `str_replace(x, pattern, replacement)` replaces the matches with new text; commonly used with `dplyr::mutate()`

forcats

Managing factors

- R uses factors to handle categorical variables
- Often important to control the ordering of factors eg for plotting or modelling
- `fct_relevel()` changes the order of a factor as specified by a character vector
- `fct_inorder()` changes the order of a factor as specified by current order ; commonly used with `dplyr::arrange()`

Into the Tidyverse: important packages

purrr

Functional programming tools for iterating with functions and vectors

- map() family of functions to replace for loops
- see the Iteration chapter of R for Data Science to learn more

https://github.com/rstudio/cheatsheets/blob/master/purrr.pdf

Apply functions with purrr : : CHEAT SHEET

Map Functions

Function Shortcuts

Work with Lists

Filter

Index

Modify

List-Columns

Reshape

Combine

Reduce

RStudio

broom

Summarizes key statistical model information in tidy format

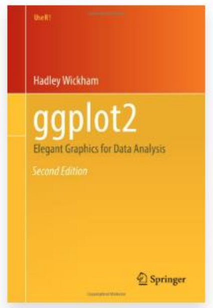
- `tidy()` summarizes information about model components.
- `glance()` reports information about the entire model.
- `augment()` adds information about observations to a dataset (eg residuals).
- Works with 100+ model objects.
- Plays well with the `nest/unnest` functions in `tidyr` and the `map` functions in `purr`

https://broom.tidymodels.org/articles/broom_and_dplyr.html

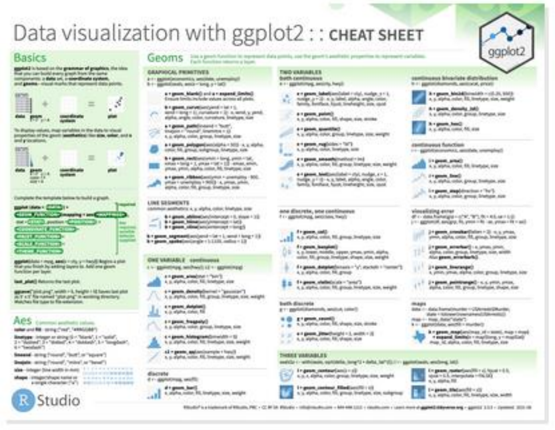
Into the Tidyverse: Visualization using ggplot2

Publication-quality data visualization

- Implements a “grammar of graphics”
- Start by defining the data to be plotted (“aesthetics”):
  `ggplot(aes(x, y, color, fill, shape, alpha, linetype))`
- Then add layers (“geoms”) to specify how data is plotted, eg:
  `+ geom_point()`
- Add additional geom layers, eg:
  `+ geom_boxplot()`
- Can split into separate plots, eg male vs. female, by “faceting”:
  `+ facet_wrap(~ Sex)`
- Finally add title and modify theme:
  ```
  + labs(title = “Plot title”, subtitle = “plot details”
  + theme(aspect.ratio = 1)
  ```



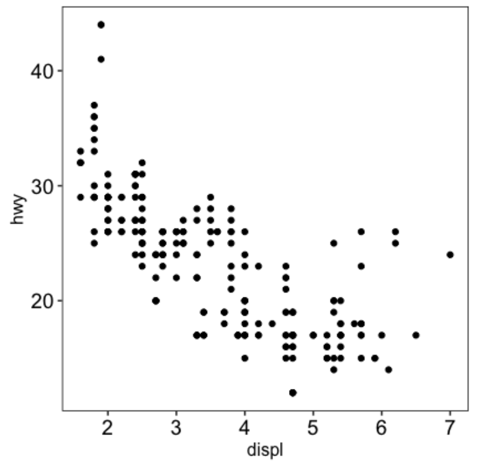
https://ggplot2-book.org/



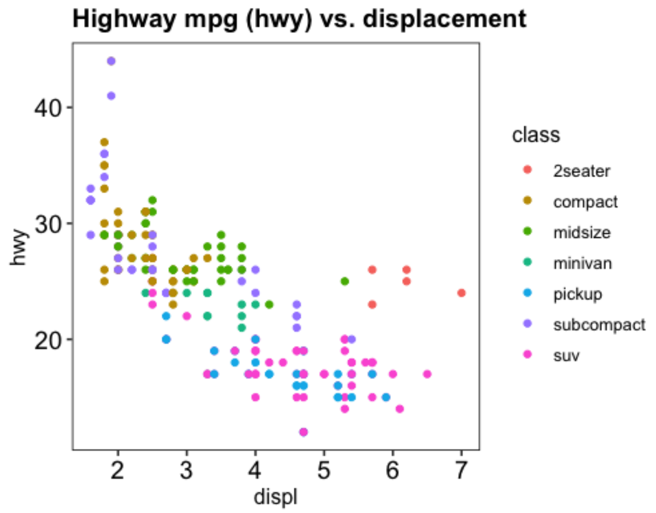
https://raw.githubusercontent.com/rstudio/cheatsheets/main/data-visualization.pdf
https://www.data-to-viz.com/caveats.html
https://r-graph-gallery.com/index.html

Into the Tidyverse: Visualization using ggplot2

```
mpg %>%
    ggplot(aes(displ, hwy)) +
    geom_point()
```



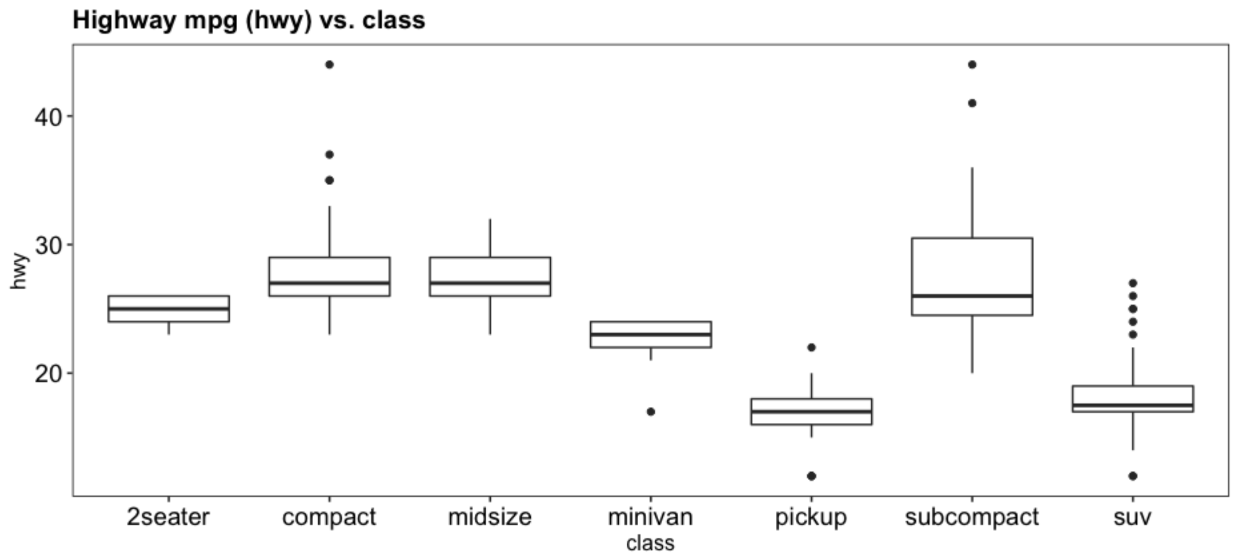
```
mpg %>%
    ggplot(aes(displ, hwy, color = class)) +
    geom_point() +
    theme(aspect.ratio = 1) +
    labs(title = "Highway mpg (hwy) vs. displacement")
```



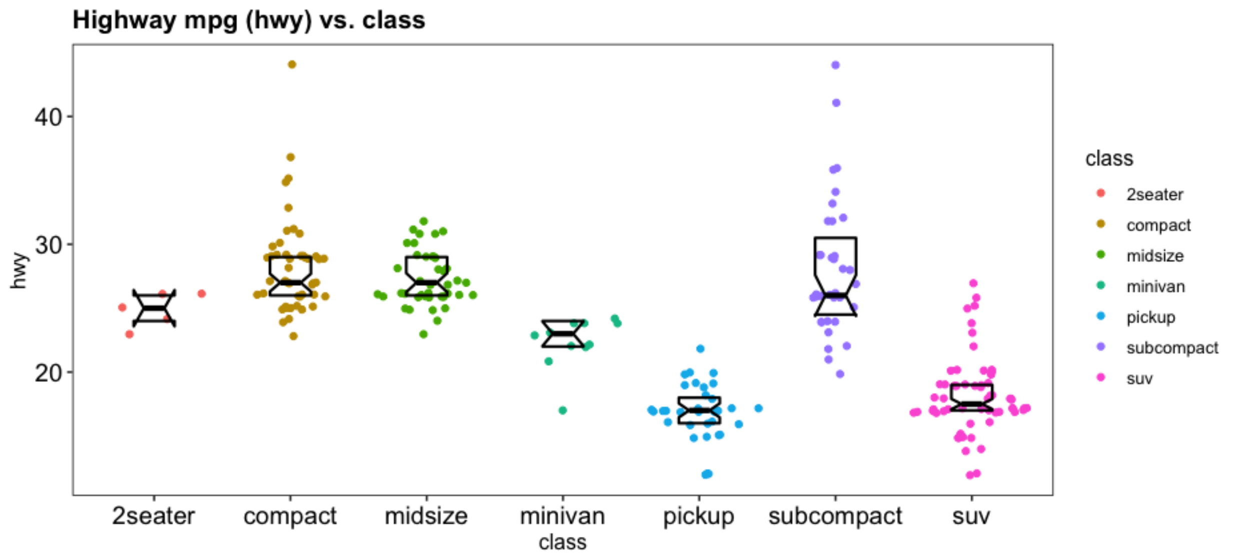
Into the Tidyverse: Visualization using ggplot2

ggplot2

```
mpg %>%
    ggplot(aes(class, hwy)) +
    geom_boxplot() +
    labs(title = "Highway mpg (hwy) vs. class")
```



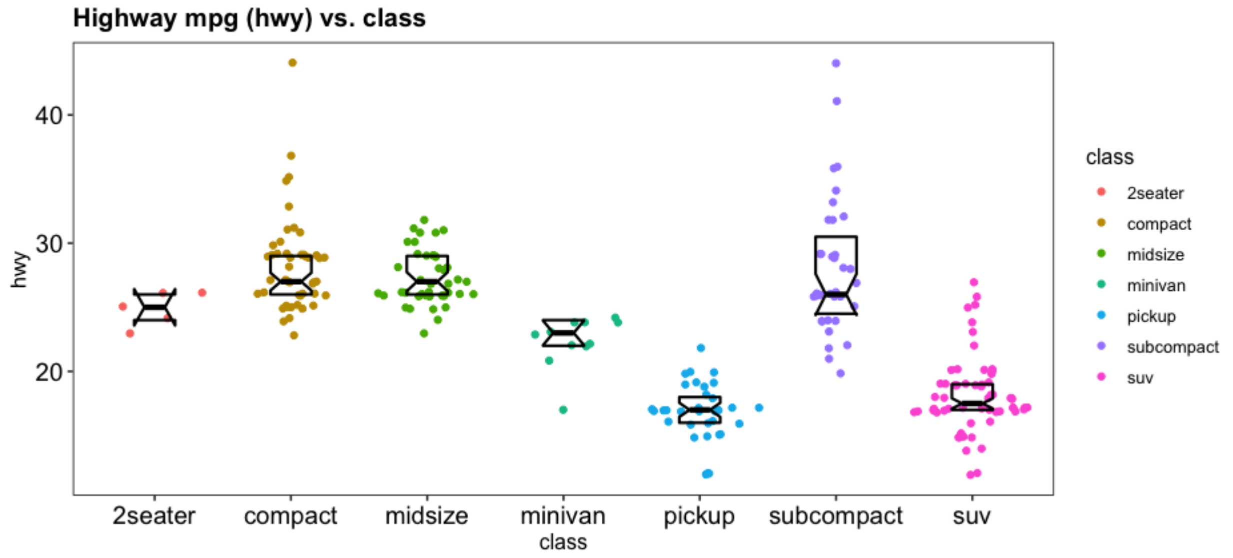
```
mpg %>%
    ggplot(aes(class, hwy, color = class)) +
    ggforce::geom_sina() +
    geom_boxplot(
       notch=TRUE, varwidth=FALSE,
       outlier.shape=NA, coef=FALSE,
       width=0.3, color="black",
       fill="transparent", size=0.75
                ) +
    labs(title = "Highway mpg (hwy) vs. class")
```

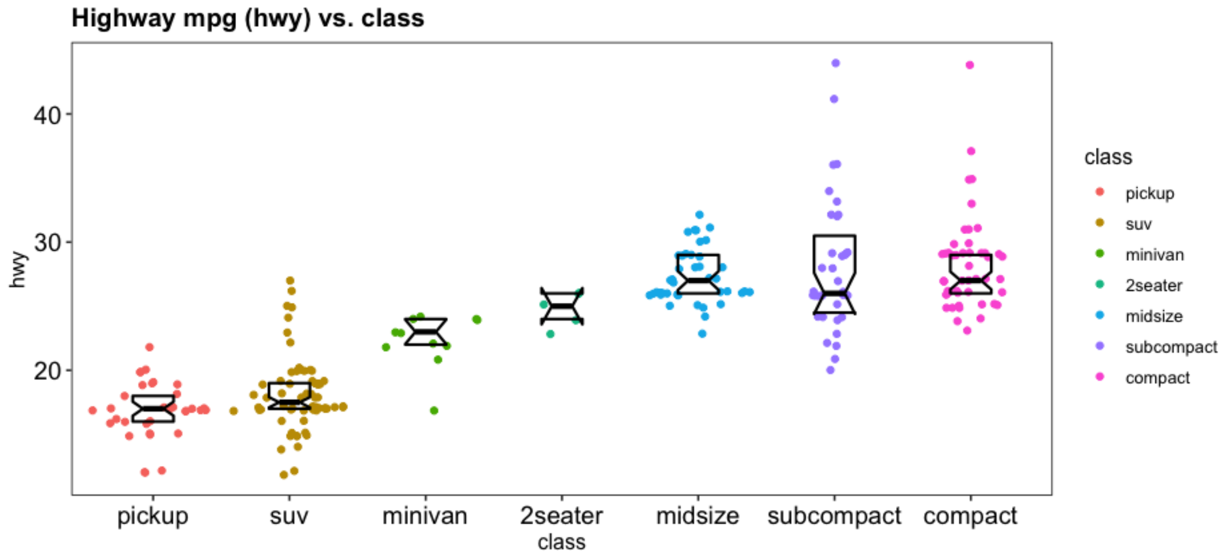


Into the Tidyverse: Visualization using ggplot2

ggplot2

```
mpg %>%
    group_by(class) %>%
    mutate(mean = mean(hwy)) %>%
    ungroup() %>%
    arrange(mean) %>%
    mutate(class = fct_inorder(class)) %>%
    ggplot(aes(class, hwy, color = class)) +
    ggforce::geom_sina() +
    geom_boxplot(
        notch=TRUE, varwidth=FALSE, outlier.shape=NA, coef=FALSE, width=0.3, color="black", fill="transparent", size=0.75
    ) +
    labs(title = "Highway mpg (hwy) vs. class")
```

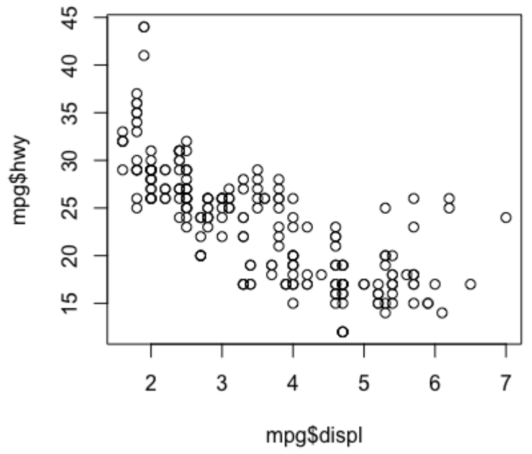




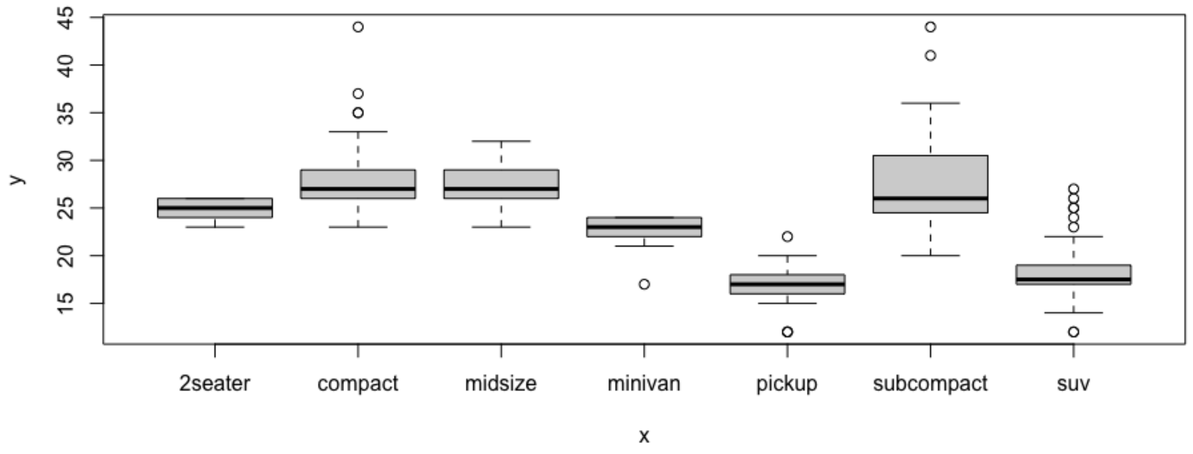
Into the Tidyverse: Visualization using ggplot2

Equivalent plots using base R

```
plot(mpg$displ, mpg$hwy)
```



```
plot(as.factor(mpg$class), mpg$hwy)
```


y
x
2seater
compact
midsize
minivan
pickup
subcompact
suv