AN INTRODUCTION TO tidyverse

Part 1

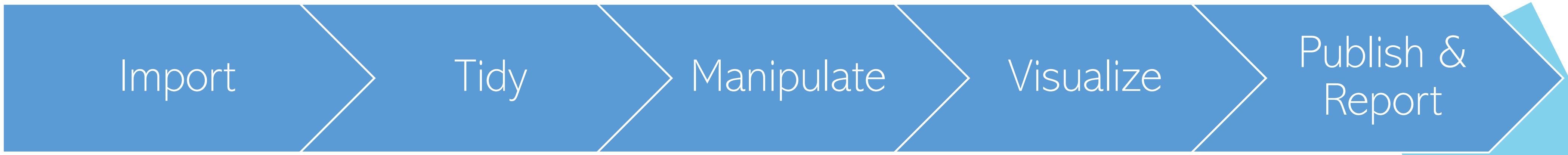
AGENDA

- Overview of tidyverse
- Data Import with readr
- Data Manipulation with dplyr
  - Basic Grammar
  - The Pipeline
  - group_by
  - case_when
- Exercises

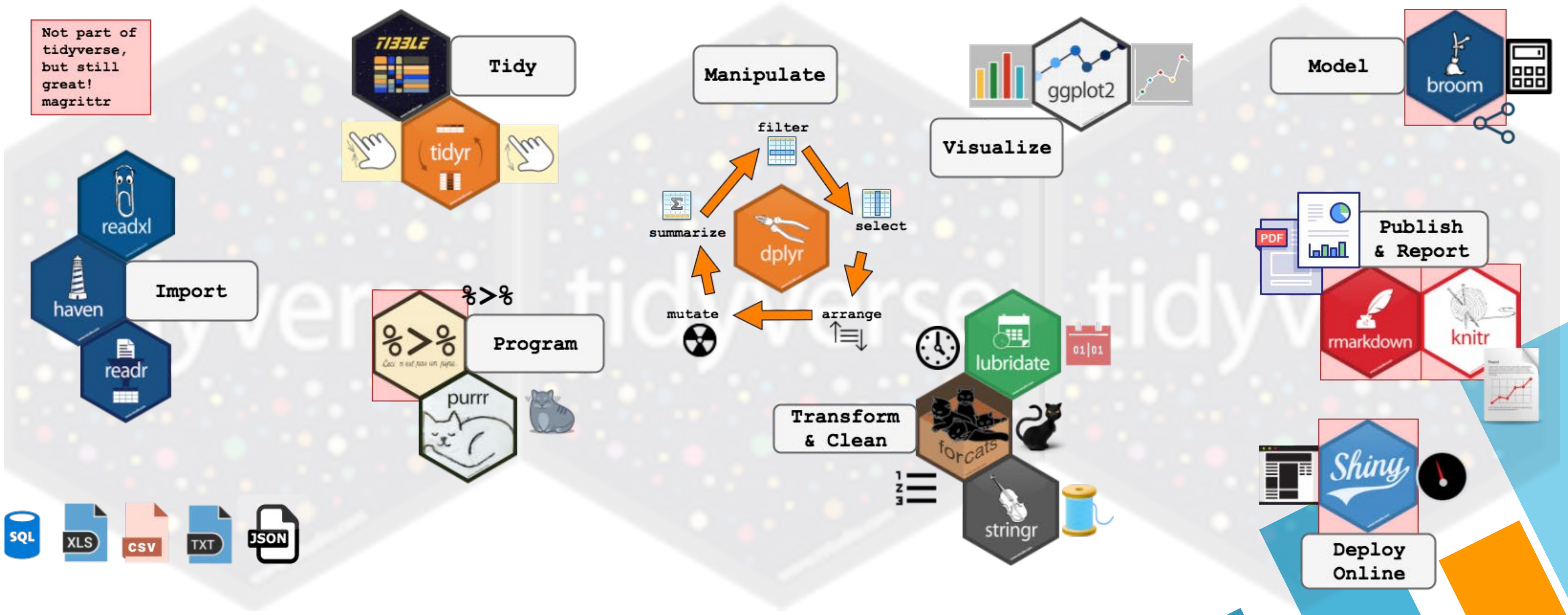
1.
OVERVIEW

WHAT IS TIDYVERSE?

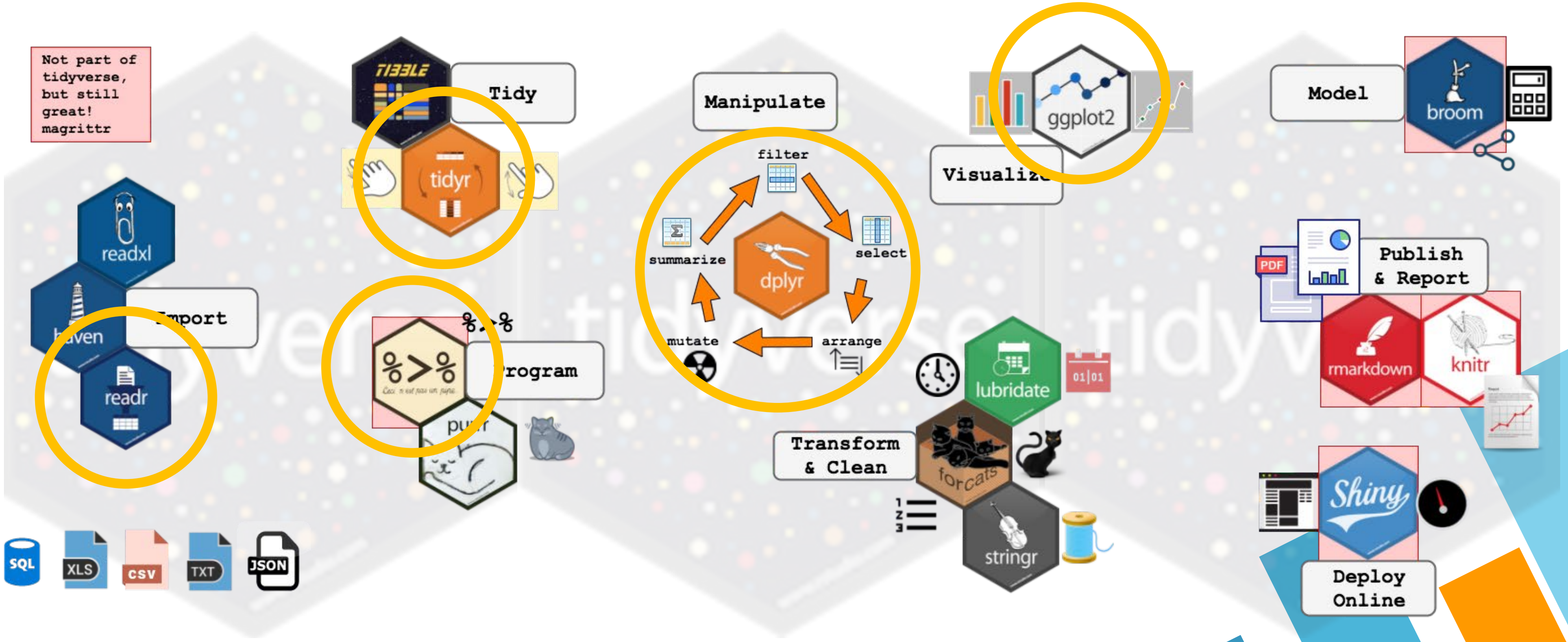
- Collection of R packages
- Covers most of the basic data analysis workflow


Import
Tidy
Manipulate
Visualize
Publish & Report

WHAT IS TIDYVERSE?



WHAT IS TIDYVERSE?

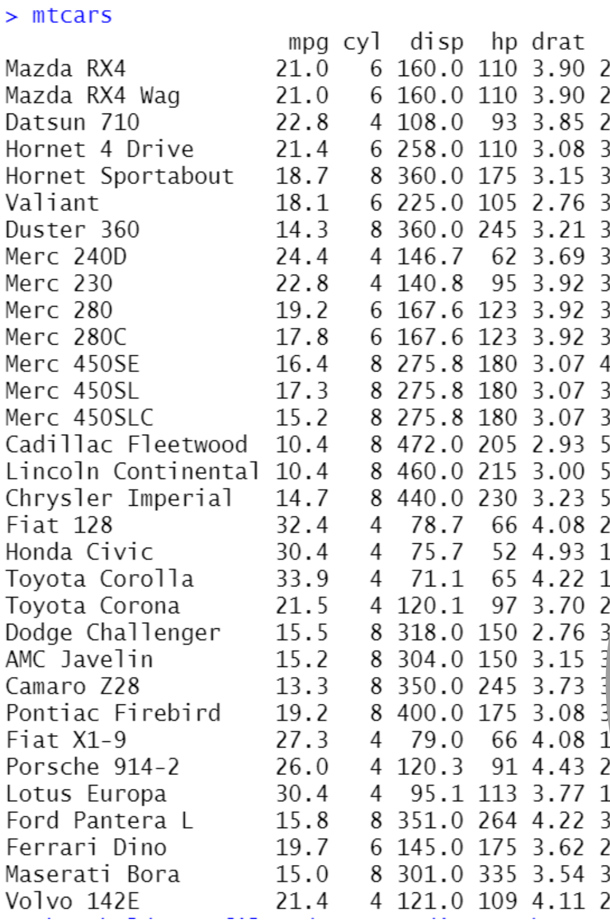


2.

MOTIVATION FOR

tidyverse
www.rstudio.com

TESTING THE CARS



```
> mtcars
                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1  0    4    4
Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0
Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00
Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.
Lincoln Continental 10.4   8 460.0 215 3.00 5.424 1
Chrysler Imperial   14.7   8 440.0 230 3.23 5.345
Fiat 128            32.4   4  78.7  66 4.08 2.20
Honda Civic         30.4   4  75.7  52 4.93 1.6
Toyota Corolla      33.9   4  71.1  65 4.22 1.
Toyota Corona       21.5   4 120.1  97 3.70 2
Dodge Challenger    15.5   8 318.0 150 2.76 3
AMC Javelin         15.2   8 304.0 150 3.15 3
Camaro Z28          13.3   8 350.0 245 3.73 3
Pontiac Firebird    19.2   8 400.0 175 3.08 3
Fiat X1-9           27.3   4  79.0  66 4.08 1
Porsche 914-2       26.0   4 120.3  91 4.43 2.
Lotus Europa        30.4   4  95.1 113 3.77 1.
Ford Pantera L      15.8   8 351.0 264 4.22 3.1
Ferrari Dino        19.7   6 145.0 175 3.62 2.77
Maserati Bora       15.0   8 301.0 335 3.54 3.570
Volvo 142E          21.4   4 121.0 109 4.11 2.780 18
```

“I need to look further into some cars from the ones we tested. Take out all the cars which have four carburetors, and only keep those whose horsepower per gear is higher than 50. Make sure cars with higher number of carburetors and lower miles/gallon come up first in the list. On that note, I need a report on the average displacement and ¼ mile time of these cars.”

3.

READ RECTANGULAR TEXT DATA WITH

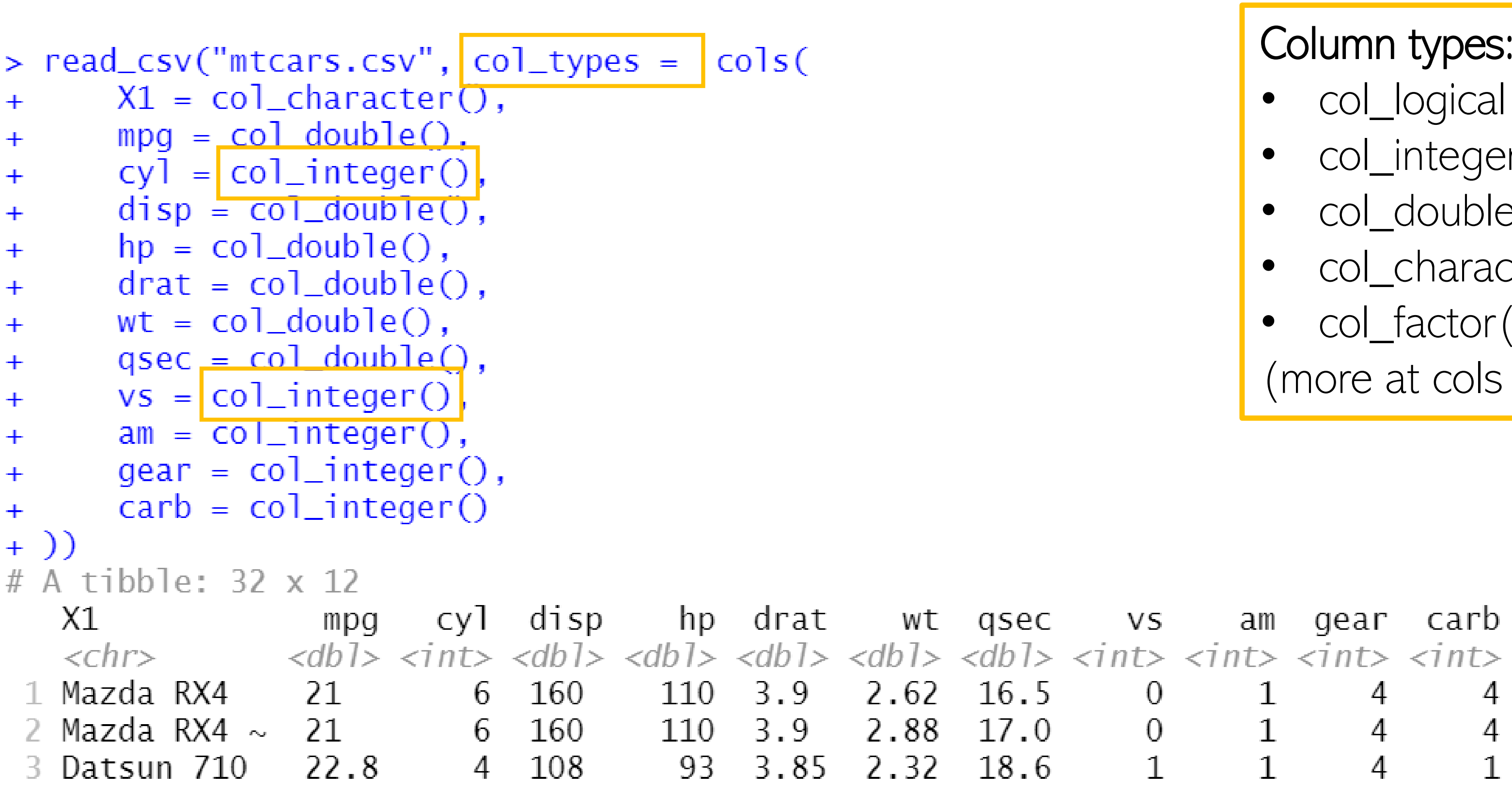
WHAT IS TIDYVERSE?

- read_csv, read_csv2
- Data type parsing

```
> read_csv("mtcars.csv")
Parsed with column specification:
cols(
  X1 = col_character(),
  mpg = col_double(),
  cyl = col_double(),
  disp = col_double(),
  hp = col_double(),
  drat = col_double(),
  wt = col_double(),
  qsec = col_double(),
  vs = col_double(),
  am = col_double(),
  gear = col_double(),
  carb = col_double()
)
# A tibble: 32 x 12
   X1             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 ~   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 D~   21.4     6  258    110  3.08  3.22  19.4     1     0     3     1
 5 Hornet Spo~   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
```

```
"","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
"Duster 360",14.3,8,360,245,3.21,3.57,15.84,0,0,3,4
"Merc 240D",24.4,4,146.7,62,3.69,3.19,20,1,0,4,2
"Merc 230",22.8,4,140.8,95,3.92,3.15,22.9,1,0,4,2
"Merc 280",19.2,6,167.6,123,3.92,3.44,18.3,1,0,4,4
"Merc 280C",17.8,6,167.6,123,3.92,3.44,18.9,1,0,4,4
"Merc 450SE",16.4,8,275.8,180,3.07,4.07,17.4,0,0,3,3
```

CHANGE COLUMN SPECIFICATION



```
> read_csv("mtcars.csv", col_types = cols(
+     X1 = col_character(),
+     mpg = col_double(),
+     cyl = col_integer(),
+     disp = col_double(),
+     hp = col_double(),
+     drat = col_double(),
+     wt = col_double(),
+     qsec = col_double(),
+     vs = col_integer(),
+     am = col_integer(),
+     gear = col_integer(),
+     carb = col_integer()
+ ))
# A tibble: 32 x 12
   X1              mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
   <chr>         <dbl> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <int> <int> <int> <int>
 1 Mazda RX4      21       6   160   110  3.9   2.62  16.5     0     1     4     4
 2 Mazda RX4 ~    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
```

Column types:

- col_logical()
- col_integer()
- col_double()
- col_character()
- col_factor(levels, ordered)

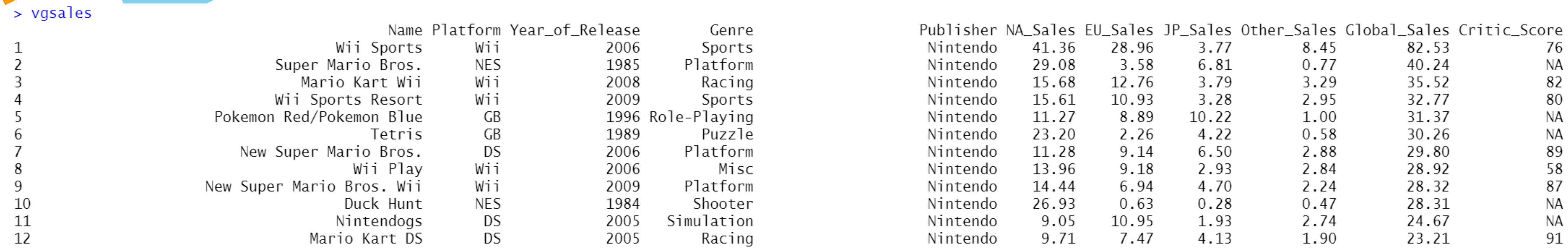
(more at cols {readr})

4.

TIBBLE

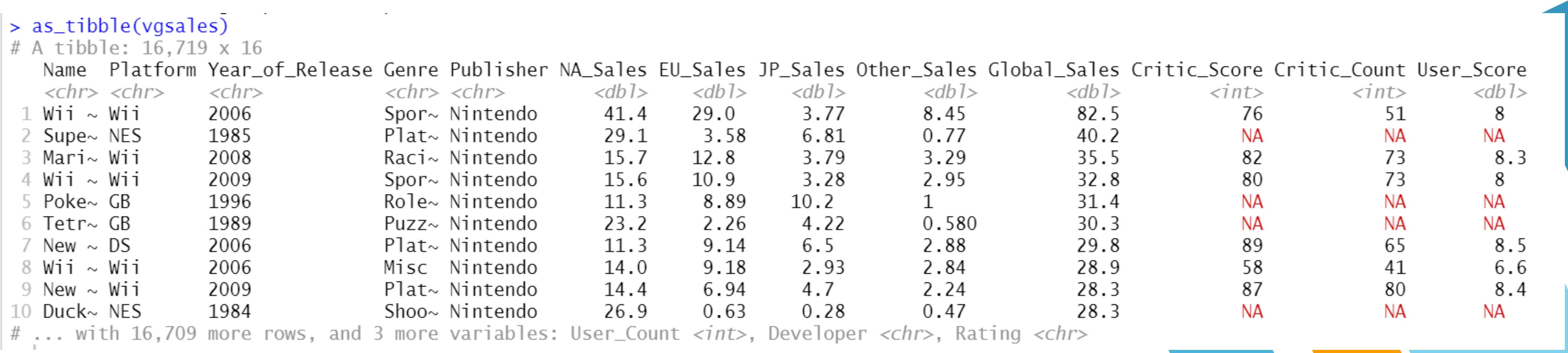
WHAT IS TIBBLE?

- A refined, more concise data frame
- Better print method
- No input type conversion

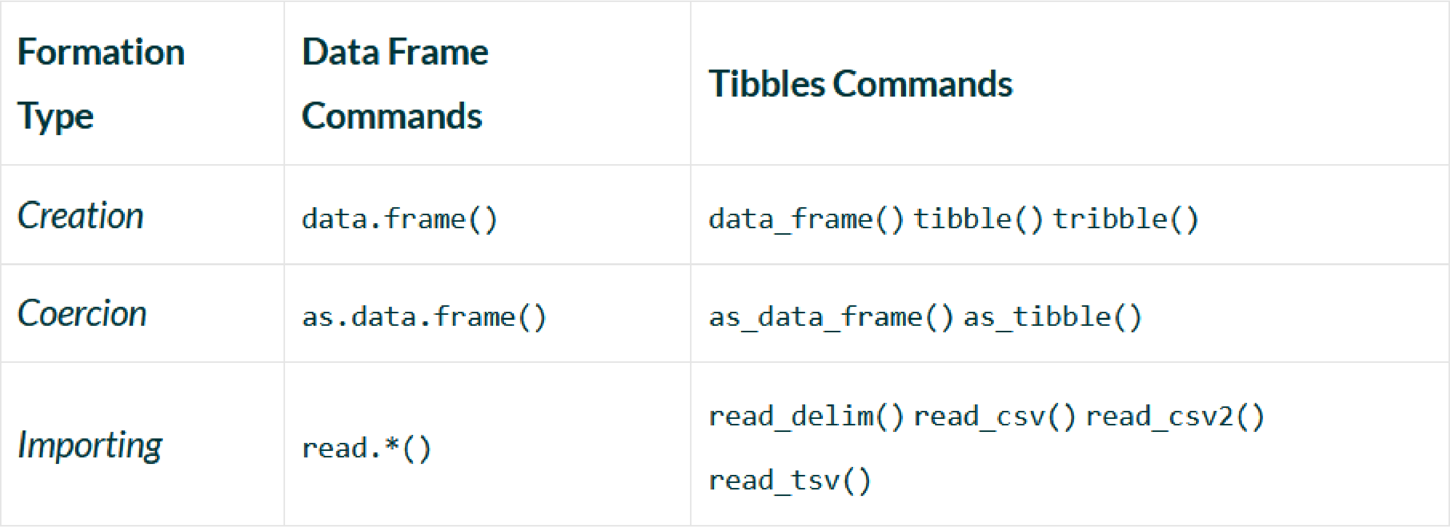


```
> vgsales
                           Name Platform Year_of_Release        Genre                   Publisher NA_Sales EU_Sales JP_Sales Other_Sales Global_Sales Critic_Score
1                    Wii Sports      Wii            2006       Sports                    Nintendo    41.36    28.96     3.77        8.45        82.53           76
2             Super Mario Bros.      NES            1985     Platform                    Nintendo    29.08     3.58     6.81        0.77        40.24           NA
3                Mario Kart Wii      Wii            2008       Racing                    Nintendo    15.68    12.76     3.79        3.29        35.52           82
4             Wii Sports Resort      Wii            2009       Sports                    Nintendo    15.61    10.93     3.28        2.95        32.77           80
5      Pokemon Red/Pokemon Blue       GB            1996 Role-Playing                    Nintendo    11.27     8.89    10.22        1.00        31.37           NA
6                        Tetris       GB            1989       Puzzle                    Nintendo    23.20     2.26     4.22        0.58        30.26           NA
7         New Super Mario Bros.       DS            2006     Platform                    Nintendo    11.28     9.14     6.50        2.88        29.80           89
8                      Wii Play      Wii            2006         Misc                    Nintendo    13.96     9.18     2.93        2.84        28.92           58
9     New Super Mario Bros. Wii      Wii            2009     Platform                    Nintendo    14.44     6.94     4.70        2.24        28.32           87
10                    Duck Hunt      NES            1984      Shooter                    Nintendo    26.93     0.63     0.28        0.47        28.31           NA
11                   Nintendogs       DS            2005   Simulation                    Nintendo     9.05    10.95     1.93        2.74        24.67           NA
12                Mario Kart DS       DS            2005       Racing                    Nintendo     9.71     7.47     4.13        1.90        23.21           91
```

TIBBLE vs DATA FRAME



```
> as_tibble(vgsales)
# A tibble: 16,719 x 16
   Name   Platform Year_of_Release Genre Publisher NA_Sales EU_Sales JP_Sales Other_Sales Global_Sales Critic_Score Critic_Count User_Score
   <chr>  <chr>    <chr>           <chr> <chr>        <dbl>    <dbl>    <dbl>       <dbl>        <dbl>        <int>        <int>      <dbl>
 1 Wii ~  Wii      2006            Spor~ Nintendo      41.4    29.0      3.77        8.45         82.5           76           51        8
 2 Supe~  NES      1985            Plat~ Nintendo      29.1     3.58     6.81        0.77         40.2           NA           NA       NA
 3 Mari~  Wii      2008            Raci~ Nintendo      15.7    12.8      3.79        3.29         35.5           82           73        8.3
 4 Wii ~  Wii      2009            Spor~ Nintendo      15.6    10.9      3.28        2.95         32.8           80           73        8
 5 Poke~  GB       1996            Role~ Nintendo      11.3     8.89    10.2         1            31.4           NA           NA       NA
 6 Tetr~  GB       1989            Puzz~ Nintendo      23.2     2.26     4.22        0.580        30.3           NA           NA       NA
 7 New ~  DS       2006            Plat~ Nintendo      11.3     9.14     6.5         2.88         29.8           89           65        8.5
 8 Wii ~  Wii      2006            Misc  Nintendo      14.0     9.18     2.93        2.84         28.9           58           41        6.6
 9 New ~  Wii      2009            Plat~ Nintendo      14.4     6.94     4.7         2.24         28.3           87           80        8.4
10 Duck~  NES      1984            Shoo~ Nintendo      26.9     0.63     0.28        0.47         28.3           NA           NA       NA
# ... with 16,709 more rows, and 3 more variables: User_Count <int>, Developer <chr>, Rating <chr>
```



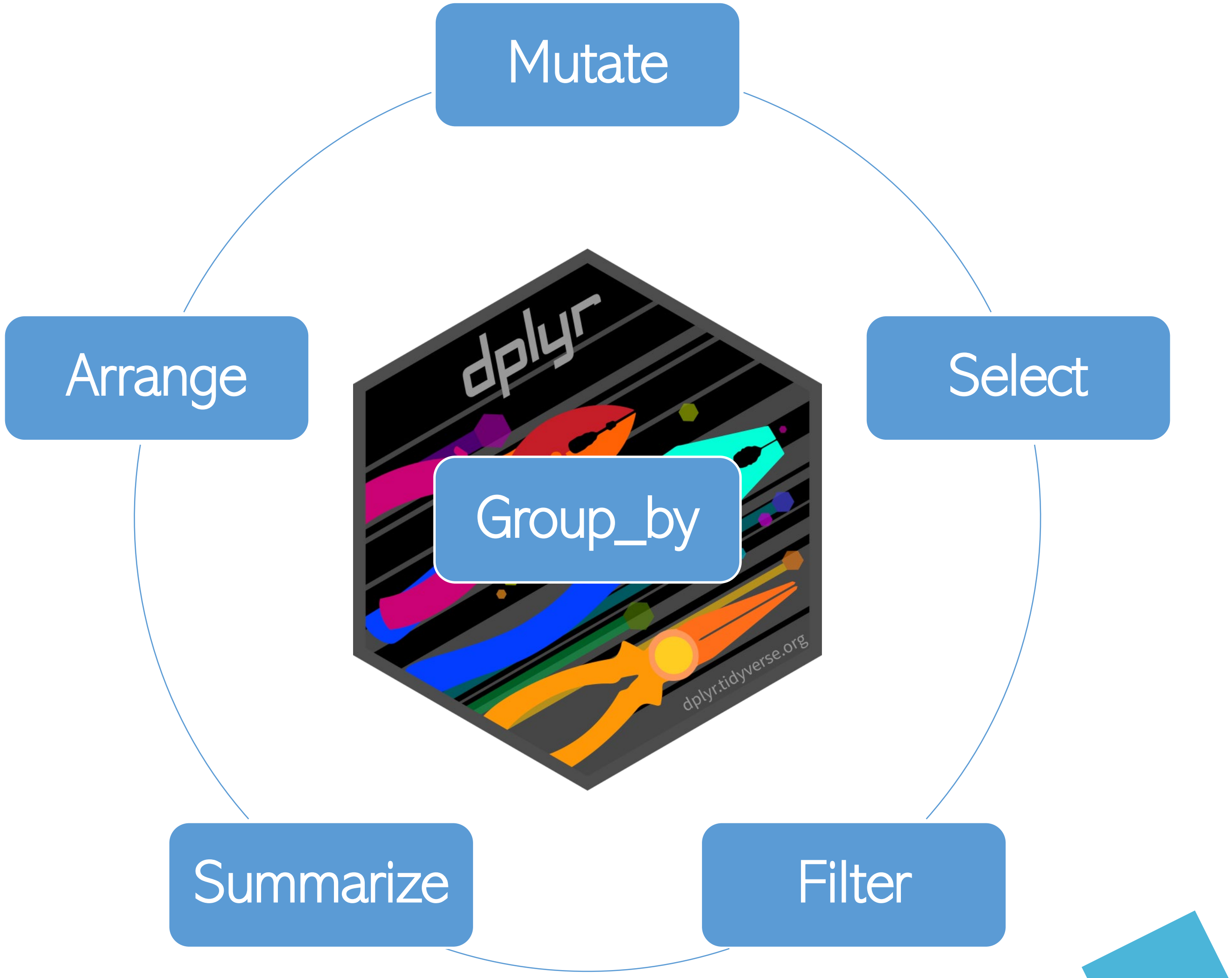
Formation Type	Data Frame Commands	Tibbles Commands
Creation	`data.frame()`	`data_frame()` `tibble()` `tribble()`
Coercion	`as.data.frame()`	`as_data_frame()` `as_tibble()`
Importing	`read.*()`	`read_delim()` `read_csv()` `read_csv2()` `read_tsv()`

More on TIBBLE

https://www.jumpingrivers.com/blog/the-trouble-with-tibbles/

4.

THE BASIC GRAMMAR OF DATA MANIPULATION



Mutate

Select

Filter

Summarize

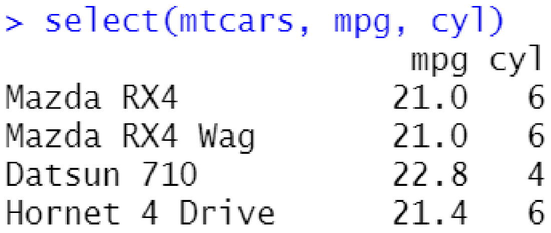
Arrange

Group_by

1 SELECT

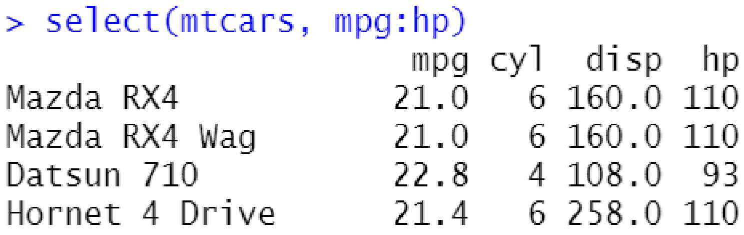
Extract subset of column(s) in a tibble

Select some columns



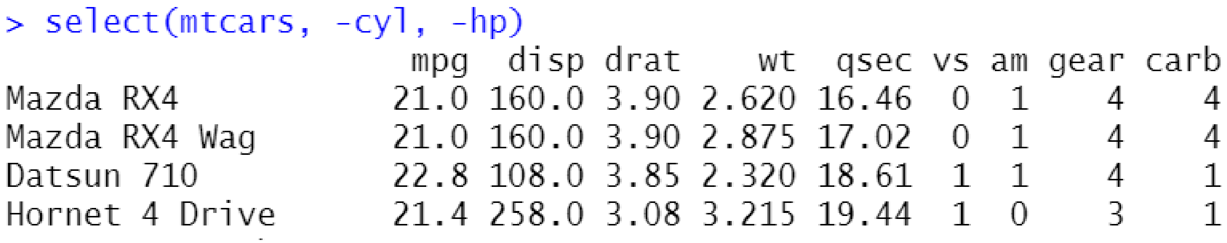
```
> select(mtcars, mpg, cyl)
                    mpg cyl
Mazda RX4          21.0   6
Mazda RX4 Wag      21.0   6
Datsun 710         22.8   4
Hornet 4 Drive     21.4   6
```

Select a range of columns



```
> select(mtcars, mpg:hp)
                    mpg cyl  disp  hp
Mazda RX4          21.0   6 160.0 110
Mazda RX4 Wag      21.0   6 160.0 110
Datsun 710         22.8   4 108.0  93
Hornet 4 Drive     21.4   6 258.0 110
```

Select all but some columns



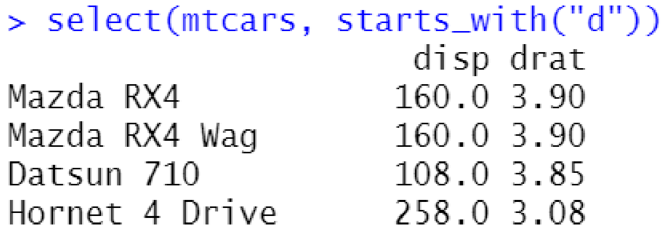
```
> select(mtcars, -cyl, -hp)
                    mpg  disp drat    wt  qsec vs am gear carb
Mazda RX4          21.0 160.0 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag      21.0 160.0 3.90 2.875 17.02  0  1    4    4
Datsun 710         22.8 108.0 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive     21.4 258.0 3.08 3.215 19.44  1  0    3    1
```

1

SELECT

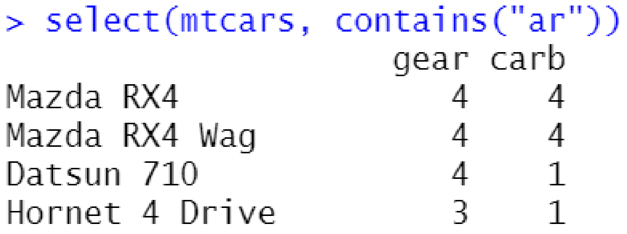
Extract subset of column(s) in a tibble

Select all columns with a certain prefix



```
> select(mtcars, starts_with("d"))
                     disp drat
Mazda RX4           160.0 3.90
Mazda RX4 Wag       160.0 3.90
Datsun 710          108.0 3.85
Hornet 4 Drive      258.0 3.08
```

Select all columns which contains a certain string



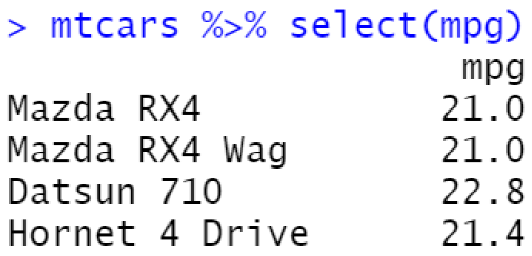
```
> select(mtcars, contains("ar"))
                    gear carb
Mazda RX4              4    4
Mazda RX4 Wag          4    4
Datsun 710             4    1
Hornet 4 Drive         3    1
```

1

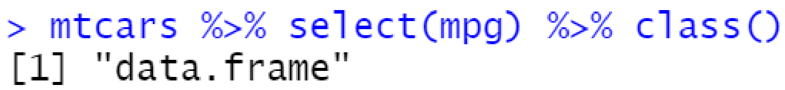
SELECT

Extract subset of column(s) in a tibble

SELECT: return a data frame/ tibble, even for a single column

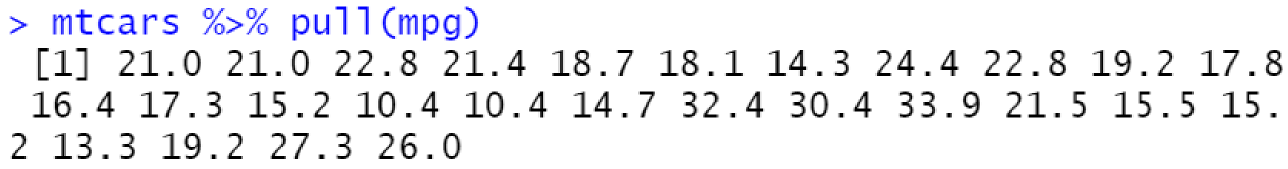


```
> mtcars %>% select(mpg)
                    mpg
Mazda RX4          21.0
Mazda RX4 Wag      21.0
Datsun 710         22.8
Hornet 4 Drive     21.4
```



```
> mtcars %>% select(mpg) %>% class()
[1] "data.frame"
```

PULL: get a vector of data

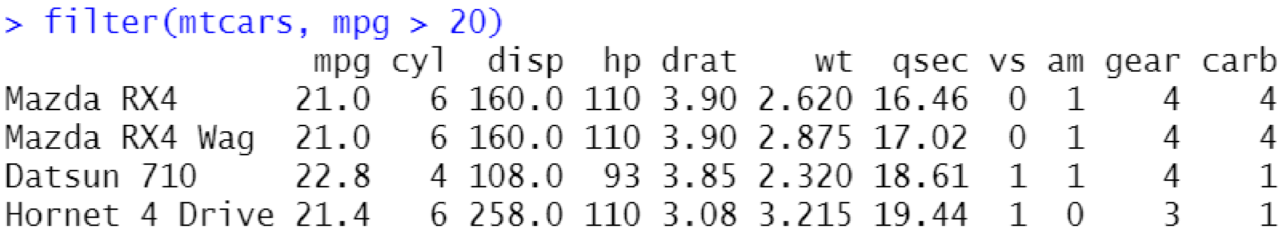


```
> mtcars %>% pull(mpg)
 [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8
 16.4 17.3 15.2 10.4 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.
2 13.3 19.2 27.3 26.0
```

2 FILTER

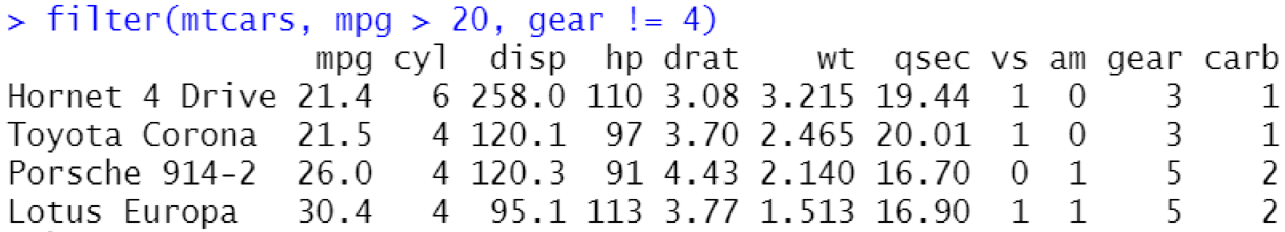
Subset rows using column values or conditions

Filter rows with a criterium



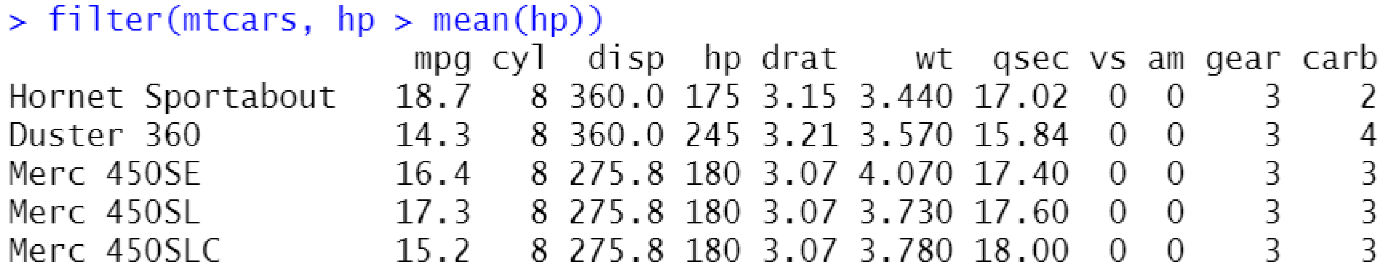
```
> filter(mtcars, mpg > 20)
                mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4      21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag  21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710     22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
```

Filter rows with multiple criteria



```
> filter(mtcars, mpg > 20, gear != 4)
                mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
Toyota Corona  21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
Porsche 914-2  26.0   4 120.3  91 4.43 2.140 16.70  0  1    5    2
Lotus Europa   30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
```

Filter rows with on-the-fly values

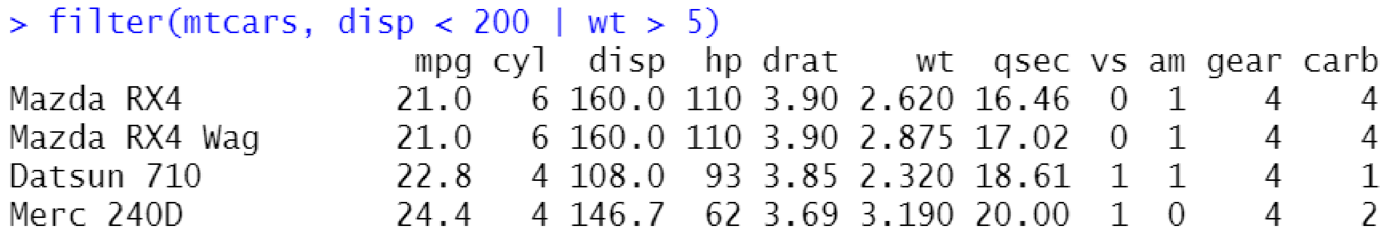


```
> filter(mtcars, hp > mean(hp))
                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3
Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3
```

2 FILTER

Subset rows using column values or conditions

Filter rows with logical expressions



```
> filter(mtcars, disp < 200 | wt > 5)
                   mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
```

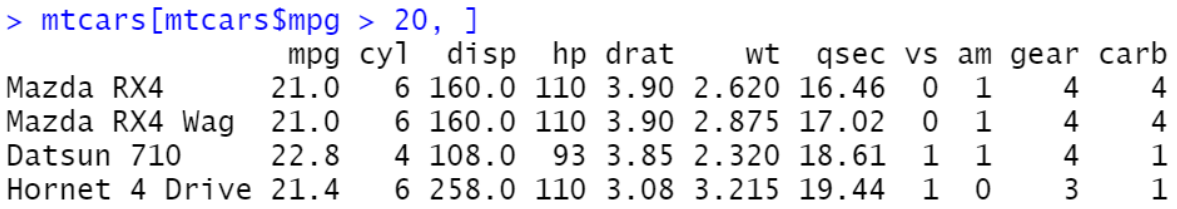
Operator	Description
+	Addition
–	Subtraction
*	Multiplication
/	Division
^	Exponent
%%	Modulus (Remainder from division)
%/%	Integer Division

Operator	Description
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	Equal to
!=	Not equal to

Operator	Description
!	NOT
&	AND
|	OR

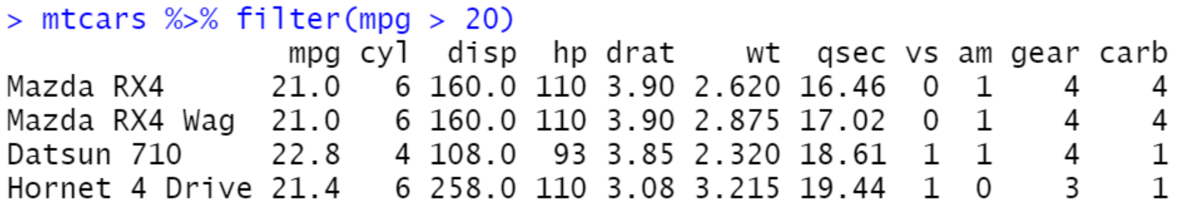
DATA MASKING

Base R: column has to be referenced by dataset



```
> mtcars[mtcars$mpg > 20, ]
                mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4      21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag  21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710     22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
```

Dplyr: Most cases reference not needed



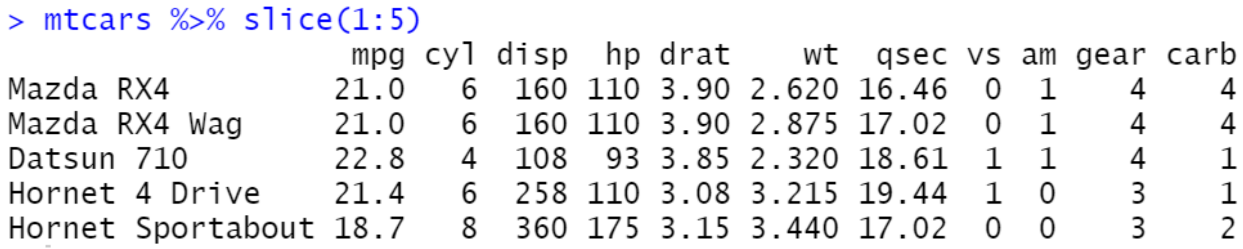
```
> mtcars %>% filter(mpg > 20)
                mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Mazda RX4      21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag  21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
Datsun 710     22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
```

2

SLICE

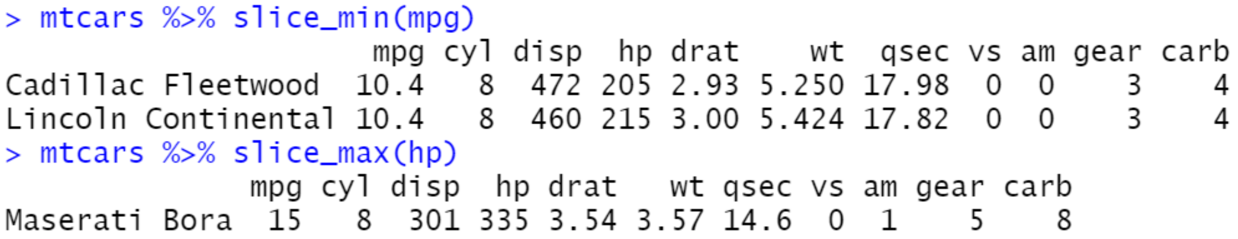
Subset rows using their position

Subset rows using their indices

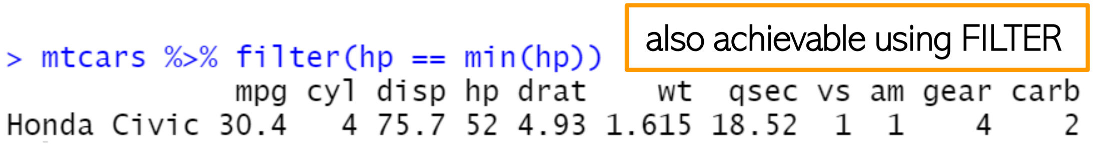


```
> mtcars %>% slice(1:5)
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    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 3.440 17.02  0  0    3    2
```

Slice rows with min/max value from a column



```
> mtcars %>% slice_min(mpg)
                     mpg cyl disp  hp drat    wt  qsec vs am gear carb
Cadillac Fleetwood  10.4   8  472 205 2.93 5.250 17.98  0  0    3    4
Lincoln Continental 10.4   8  460 215 3.00 5.424 17.82  0  0    3    4
> mtcars %>% slice_max(hp)
              mpg cyl disp  hp drat   wt qsec vs am gear carb
Maserati Bora  15   8  301 335 3.54 3.57 14.6  0  1    5    8
```



also achievable using FILTER

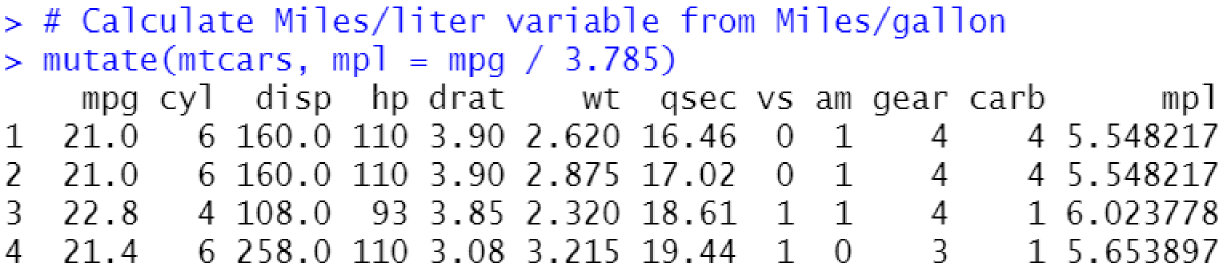
```
> mtcars %>% filter(hp == min(hp))
             mpg cyl disp hp drat    wt  qsec vs am gear carb
Honda Civic 30.4   4 75.7 52 4.93 1.615 18.52  1  1    4    2
```

3

MUTATE

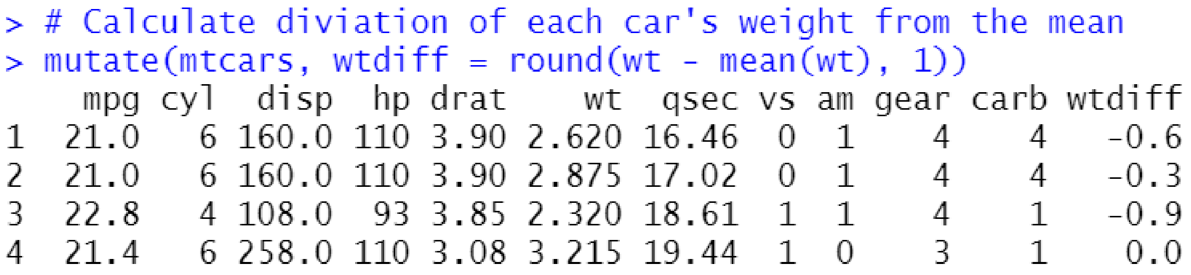
Create, modify and delete columns

Calculate a new variable from the existing one



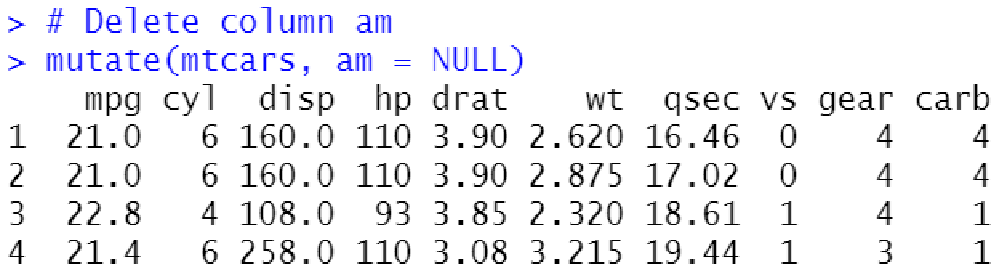
```
> # Calculate Miles/liter variable from Miles/gallon
> mutate(mtcars, mpl = mpg / 3.785)
   mpg cyl  disp  hp drat    wt  qsec vs am gear carb      mpl
1 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4 5.548217
2 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4 5.548217
3 22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1 6.023778
4 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1 5.653897
```

Using “one-the-fly” results



```
> # Calculate diviation of each car's weight from the mean
> mutate(mtcars, wtdiff = round(wt - mean(wt), 1))
   mpg cyl  disp  hp drat    wt  qsec vs am gear carb wtdiff
1 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4   -0.6
2 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4   -0.3
3 22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1   -0.9
4 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1    0.0
```

Another way to delete columns



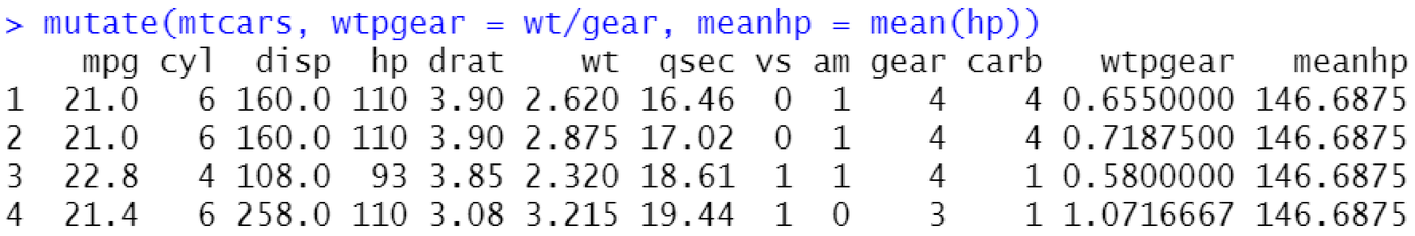
```
> # Delete column am
> mutate(mtcars, am = NULL)
   mpg cyl  disp  hp drat    wt  qsec vs gear carb
1 21.0   6 160.0 110 3.90 2.620 16.46  0    4    4
2 21.0   6 160.0 110 3.90 2.875 17.02  0    4    4
3 22.8   4 108.0  93 3.85 2.320 18.61  1    4    1
4 21.4   6 258.0 110 3.08 3.215 19.44  1    3    1
```

3

MUTATE

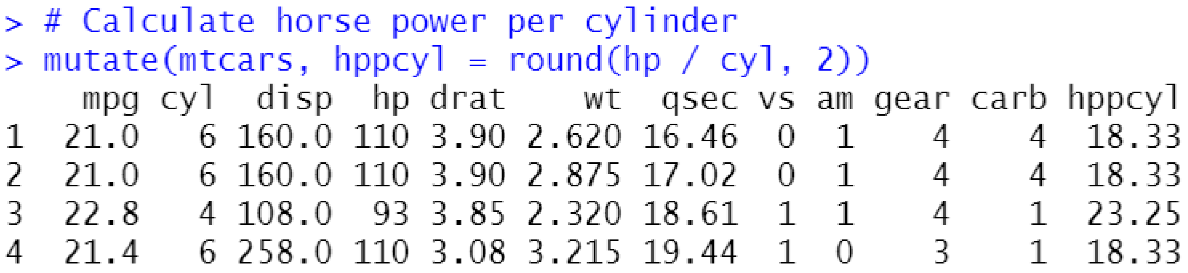
Create, modify and delete columns

Calculate multiple variables



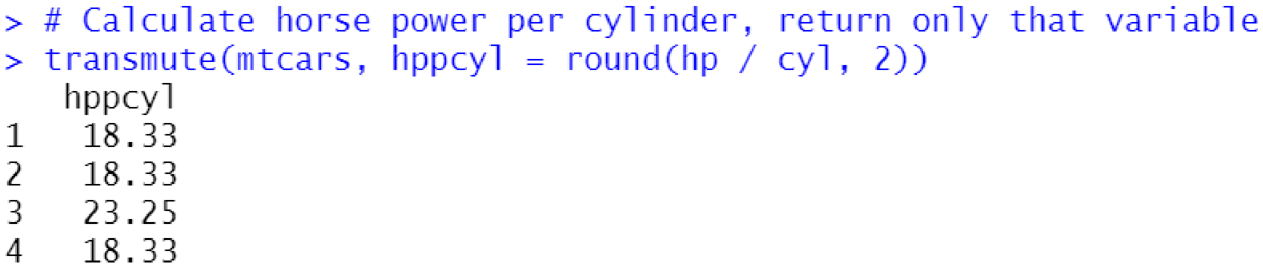
```
> mutate(mtcars, wtpgear = wt/gear, meanhp = mean(hp))
   mpg cyl  disp  hp drat    wt  qsec vs am gear carb   wtpgear   meanhp
1 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4 0.6550000 146.6875
2 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4 0.7187500 146.6875
3 22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1 0.5800000 146.6875
4 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1 1.0716667 146.6875
```

MUTATE: keep all columns



```
> # Calculate horse power per cylinder
> mutate(mtcars, hppcyl = round(hp / cyl, 2))
   mpg cyl  disp  hp drat    wt  qsec vs am gear carb hppcyl
1 21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4  18.33
2 21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4  18.33
3 22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1  23.25
4 21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1  18.33
```

TRANSMUTE: keep only calculated columns

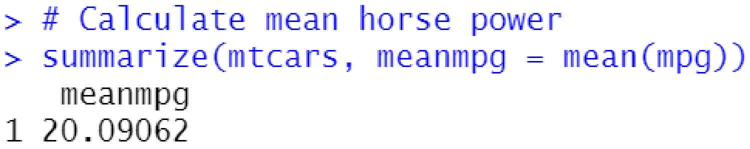


```
> # Calculate horse power per cylinder, return only that variable
> transmute(mtcars, hppcyl = round(hp / cyl, 2))
  hppcyl
1  18.33
2  18.33
3  23.25
4  18.33
```

4 SUMMARIZE

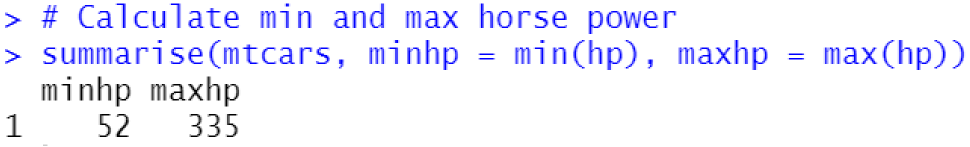
Summarize to
fewer rows

Summarize a column into a value



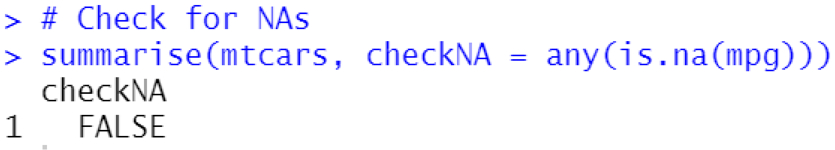
```
> # Calculate mean horse power
> summarize(mtcars, meanmpg = mean(mpg))
   meanmpg
1 20.09062
```

Multiple summarized columns



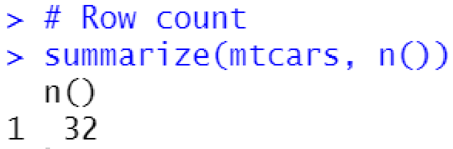
```
> # Calculate min and max horse power
> summarise(mtcars, minhp = min(hp), maxhp = max(hp))
  minhp maxhp
1    52   335
```

Check for NAs



```
> # Check for NAs
> summarise(mtcars, checkNA = any(is.na(mpg)))
  checkNA
1   FALSE
```

Row counts



```
> # Row count
> summarize(mtcars, n())
  n()
1  32
```

Useful functions

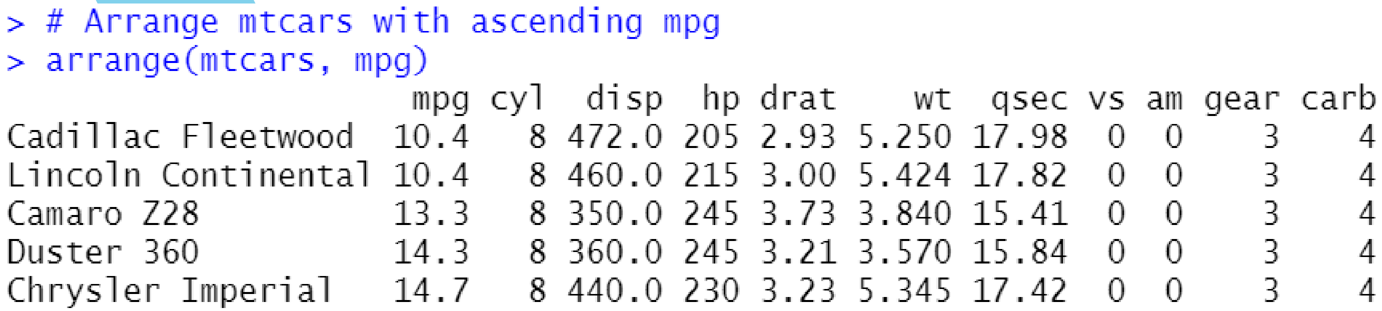
Center	mean(), median()
Spread	sd()
Range	min(), max(), quantile()
Position	first(), last()
Count	n()
Logical	any(), all()

5

ARRANGE

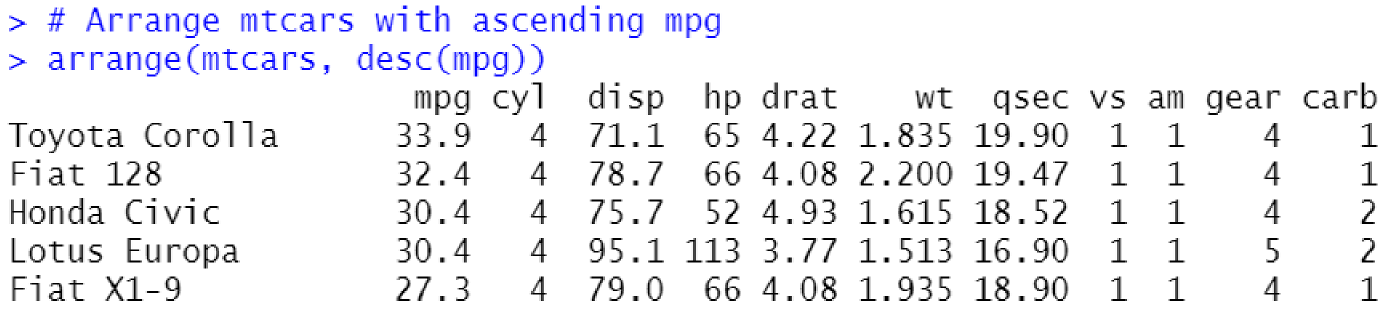
Arrange rows by column values

Sort tibble based on a column



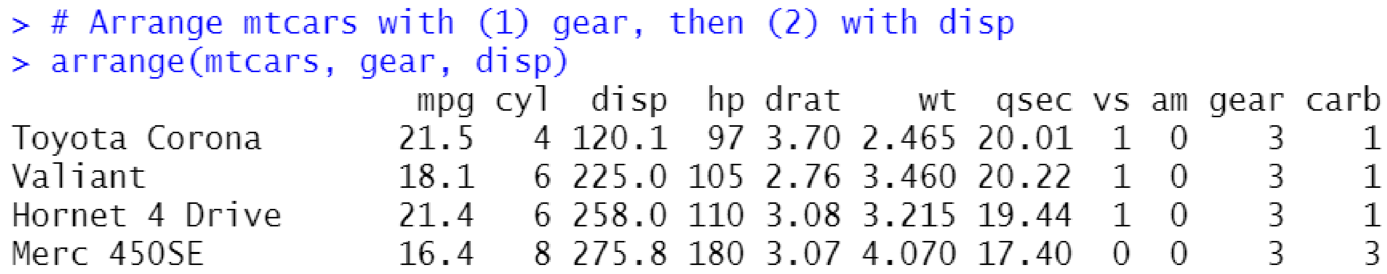
```
> # Arrange mtcars with ascending mpg
> arrange(mtcars, mpg)
                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0  0    3    4
Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0  0    3    4
Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0  0    3    4
Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0  0    3    4
```

With descending order



```
> # Arrange mtcars with ascending mpg
> arrange(mtcars, desc(mpg))
                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1    4    1
Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1    4    1
Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  1    4    2
Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  1    5    2
Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  1    4    1
```

Using multiple columns



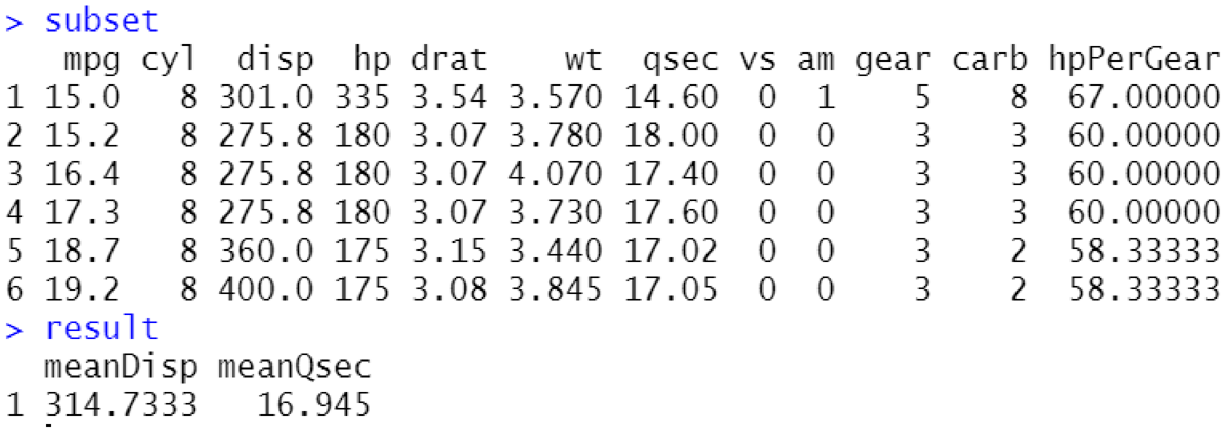
```
> # Arrange mtcars with (1) gear, then (2) with disp
> arrange(mtcars, gear, disp)
                     mpg cyl  disp  hp drat    wt  qsec vs am gear carb
Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1  0    3    1
Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3
```

EX: PUTTING ALL TOGETHER

“I need to look further into some cars from the ones we tested. Take out all the cars which have four carburetors, and only keep those whose horsepower per gear is higher than 50. Make sure cars with higher number of carburetors and lower miles/gallon come up first in the list. On that note, I need a report on the average displacement and ¼ mile time of these cars.”

EX: PUTTING ALL TOGETHER

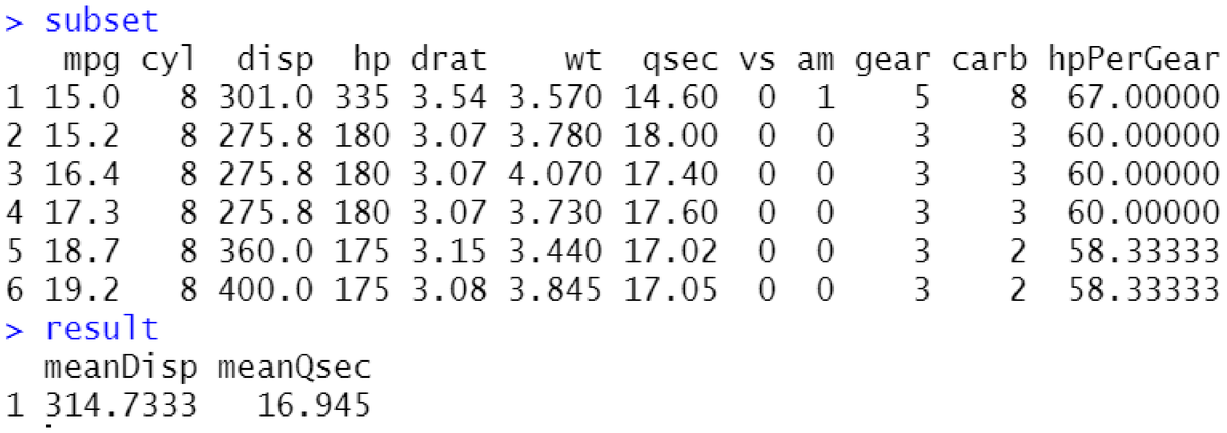
“I need to look further into some cars from the ones we tested. Take out all the cars which have four carburetors, and only keep those whose horsepower per gear is higher than 50. Make sure cars with higher number of carburetors and lower miles/gallon come up first in the list. On that note, I need a report on the average displacement and ¼ mile time of these cars.”



```
> subset
   mpg cyl  disp  hp drat    wt  qsec vs am gear carb hpPerGear
1 15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8  67.00000
2 15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3  60.00000
3 16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3  60.00000
4 17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3  60.00000
5 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2  58.33333
6 19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2  58.33333
> result
  meanDisp meanQsec
1 314.7333   16.945
```

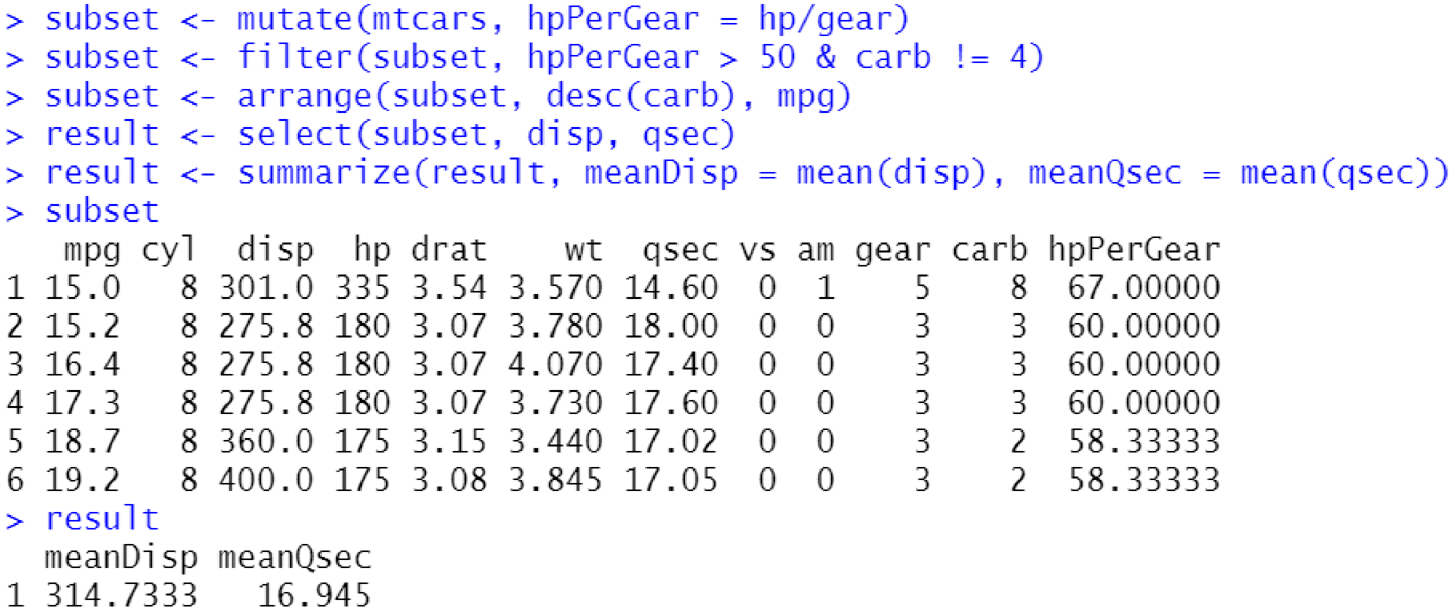
EX: PUTTING ALL TOGETHER

I need to look further into some cars from the ones we tested. **Take out** all the cars which **have four carburetors**, and only **keep** those whose **horsepower per gear is higher than 50**. Make sure cars with **higher number of carburetors** and **lower miles/gallon** come up first in the list. On that note, I need a report on the **average displacement and ¼ mile time** of these cars.



```
> subset
   mpg cyl  disp  hp drat    wt  qsec vs am gear carb hpPerGear
1 15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8  67.00000
2 15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3  60.00000
3 16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3  60.00000
4 17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3  60.00000
5 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2  58.33333
6 19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2  58.33333
> result
  meanDisp meanQsec
1 314.7333   16.945
```

EX: PUTTING ALL TOGETHER

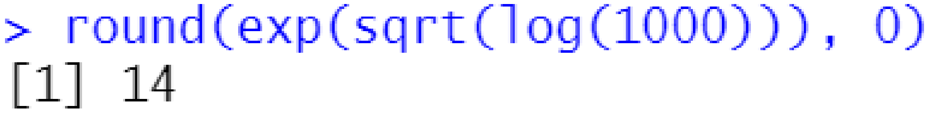


```
> subset <- mutate(mtcars, hpPerGear = hp/gear)
> subset <- filter(subset, hpPerGear > 50 & carb != 4)
> subset <- arrange(subset, desc(carb), mpg)
> result <- select(subset, disp, qsec)
> result <- summarize(result, meanDisp = mean(disp), meanQsec = mean(qsec))
> subset
   mpg cyl  disp  hp drat    wt  qsec vs am gear carb hpPerGear
1 15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8  67.00000
2 15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3  60.00000
3 16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3  60.00000
4 17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3  60.00000
5 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2  58.33333
6 19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2  58.33333
> result
  meanDisp meanQsec
1 314.7333   16.945
```

I need to look further into some cars from the ones we tested. **Take out** all the cars which **have four carburetors**, and only **keep** those whose **horsepower per gear is higher than 50**. Make sure cars with **higher number of carburetors** and **lower miles/gallon** come up first in the list. On that note, I need a report on the **average displacement and ¼ mile time** of these cars.

THE PIPE OPERATOR %>%

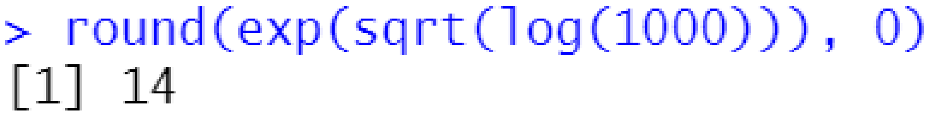
I want to calculate the rounded value **OF** the exponent **OF** the square root **OF** the logarithm OF 1000.



```
> round(exp(sqrt(log(1000))), 0)
[1] 14
```

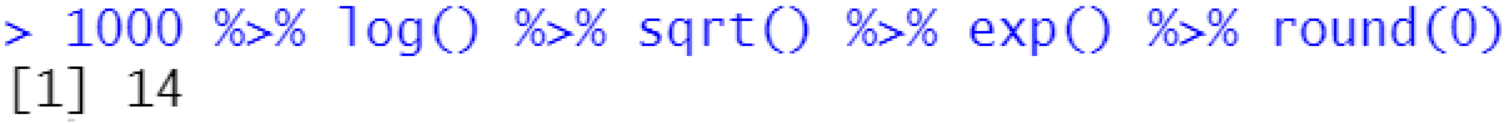
THE PIPE OPERATOR %>%

I want to calculate the rounded value OF the exponent OF the square root OF the logarithm OF 1000.



```
> round(exp(sqrt(log(1000))), 0)
[1] 14
```

I want to TAKE 1000 THEN calculate the logarithm THEN the square root THEN the exponent THEN round it up.



```
> 1000 %>% log() %>% sqrt() %>% exp() %>% round(0)
[1] 14
```

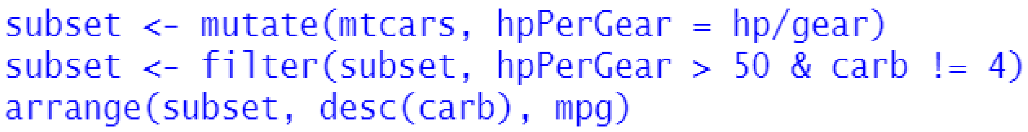
THE PIPE OPERATOR %>%

Advantages

- Left-to-right structured sequence of operations
- Avoid nesting functions
- Reducing needs for extra variables
- Easy to modify in any steps of the operation

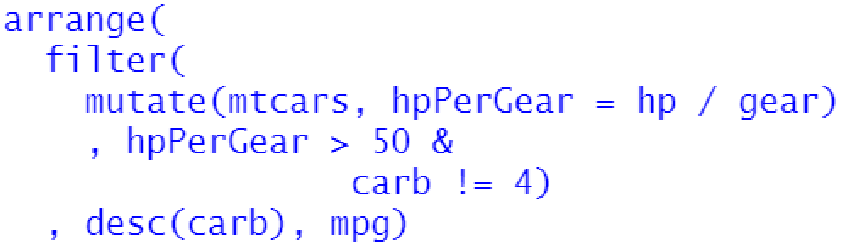
THE PIPE OPERATOR %>%

Traditional way

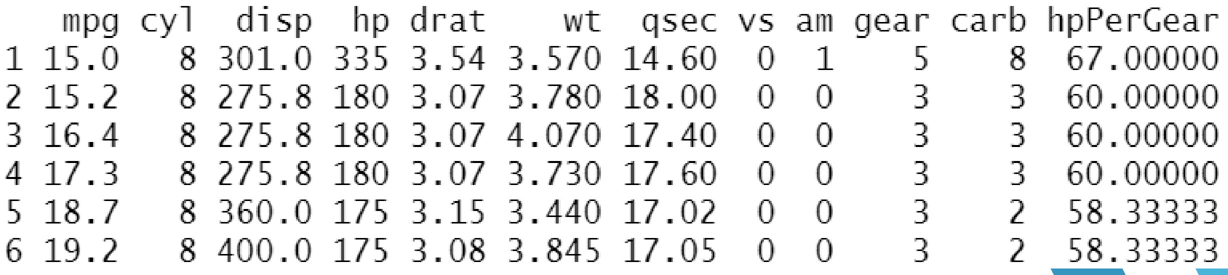


```
subset <- mutate(mtcars, hpPerGear = hp/gear)
subset <- filter(subset, hpPerGear > 50 & carb != 4)
arrange(subset, desc(carb), mpg)
```

or



```
arrange(
  filter(
    mutate(mtcars, hpPerGear = hp / gear)
    , hpPerGear > 50 &
                carb != 4)
  , desc(carb), mpg)
```



```
   mpg cyl  disp  hp drat    wt  qsec vs am gear carb hpPerGear
1 15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8  67.00000
2 15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3  60.00000
3 16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3  60.00000
4 17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3  60.00000
5 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2  58.33333
6 19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2  58.33333
```

THE PIPE OPERATOR %>%

Traditional way

```
subset <- mutate(mtcars, hpPerGear = hp/gear)
subset <- filter(subset, hpPerGear > 50 & carb != 4)
arrange(subset, desc(carb), mpg)
```

or

```
arrange(
  filter(
    mutate(mtcars, hpPerGear = hp / gear)
    , hpPerGear > 50 &
                 carb != 4)
  , desc(carb), mpg)
```

With pipe operator

```
mtcars %>%
  mutate(hpPerGear = hp/gear) %>%
  filter(hpPerGear > 50, carb != 4) %>%
  arrange(desc(carb), mpg)
```

```
   mpg cyl  disp  hp drat    wt  qsec vs am gear carb hpPerGear
1 15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8  67.00000
2 15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3  60.00000
3 16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3  60.00000
4 17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3  60.00000
5 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2  58.33333
6 19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2  58.33333
```

THE PIPE OPERATOR %>%

Traditional way

```
subset <- mutate(mtcars, hpPerGear = hp/gear)
subset <- filter(subset, hpPerGear > 50 & carb != 4)
arrange(subset, desc(carb), mpg)
```

or

```
arrange(
  filter(
    mutate(mtcars, hpPerGear = hp / gear)
    , hpPerGear > 50 &
                  carb != 4)
  , desc(carb), mpg)
```

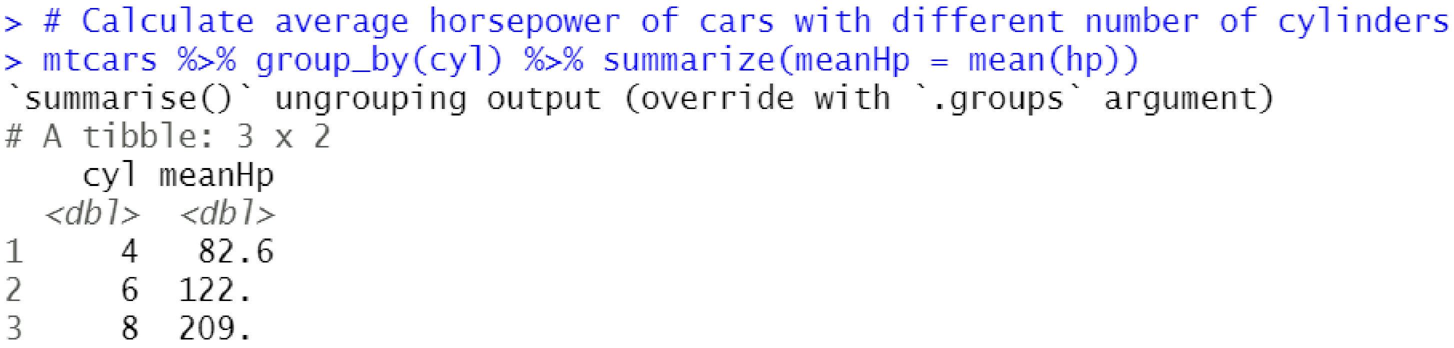
With pipe operator

```
mtcars %>%
  mutate(hpPerGear = hp/gear) %>%
  filter(hpPerGear > 50, carb != 4) %>%
  arrange(desc(carb), mpg)
```

```
   mpg cyl  disp  hp drat    wt  qsec vs am gear carb hpPerGear
1 15.0   8 301.0 335 3.54 3.570 14.60  0  1    5    8  67.00000
2 15.2   8 275.8 180 3.07 3.780 18.00  0  0    3    3  60.00000
3 16.4   8 275.8 180 3.07 4.070 17.40  0  0    3    3  60.00000
4 17.3   8 275.8 180 3.07 3.730 17.60  0  0    3    3  60.00000
5 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2  58.33333
6 19.2   8 400.0 175 3.08 3.845 17.05  0  0    3    2  58.33333
```

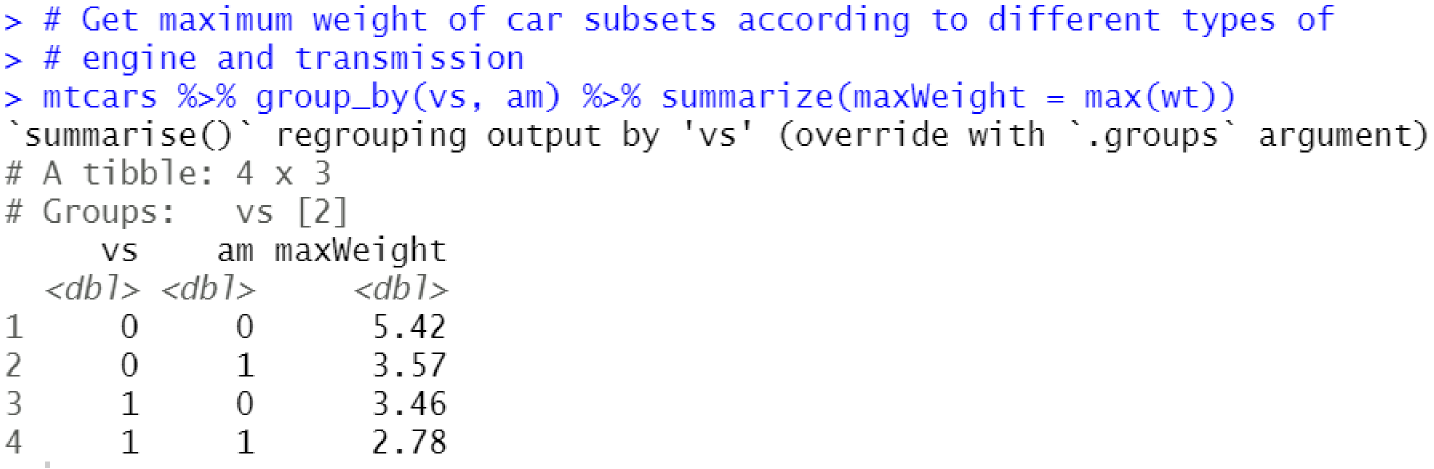
GROUP_BY

Summarize with groups in a column



```
> # Calculate average horsepower of cars with different number of cylinders
> mtcars %>% group_by(cyl) %>% summarize(meanHp = mean(hp))
`summarise()` ungrouping output (override with `.groups` argument)
# A tibble: 3 x 2
    cyl meanHp
  <dbl>  <dbl>
1     4   82.6
2     6  122. 
3     8  209. 
```

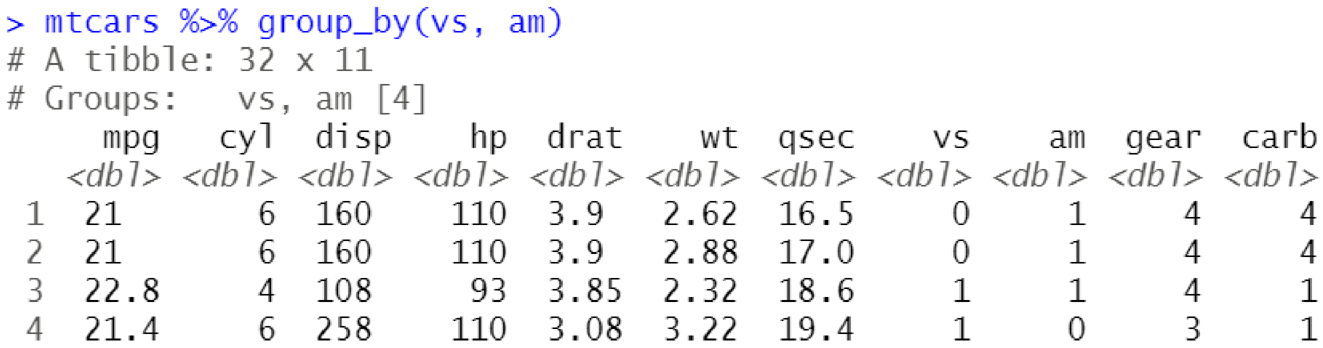
Groups with multiple columns



```
> # Get maximum weight of car subsets according to different types of
> # engine and transmission
> mtcars %>% group_by(vs, am) %>% summarize(maxWeight = max(wt))
`summarise()` regrouping output by 'vs' (override with `.groups` argument)
# A tibble: 4 x 3
# Groups:   vs [2]
     vs    am maxWeight
  <dbl> <dbl>     <dbl>
1     0     0      5.42
2     0     1      3.57
3     1     0      3.46
4     1     1      2.78
```

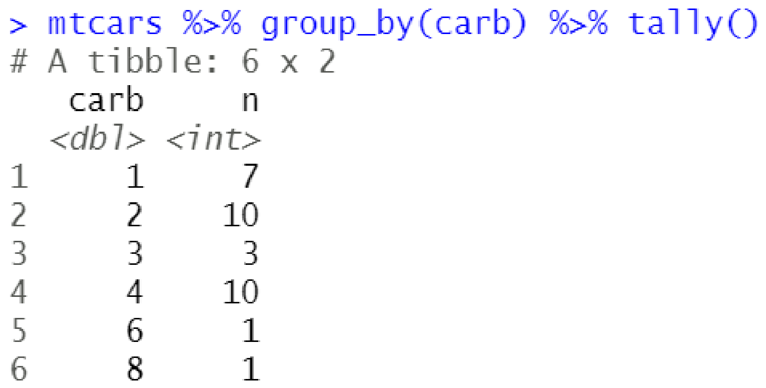
GROUP_BY

Group_by alone doesn't change how data look



```
> mtcars %>% group_by(vs, am)
# A tibble: 32 x 11
# Groups:   vs, am [4]
    mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
1  21       6   160   110  3.9   2.62  16.5     0     1     4     4
2  21       6   160   110  3.9   2.88  17.0     0     1     4     4
3  22.8     4   108    93  3.85  2.32  18.6     1     1     4     1
4  21.4     6   258   110  3.08  3.22  19.4     1     0     3     1
```

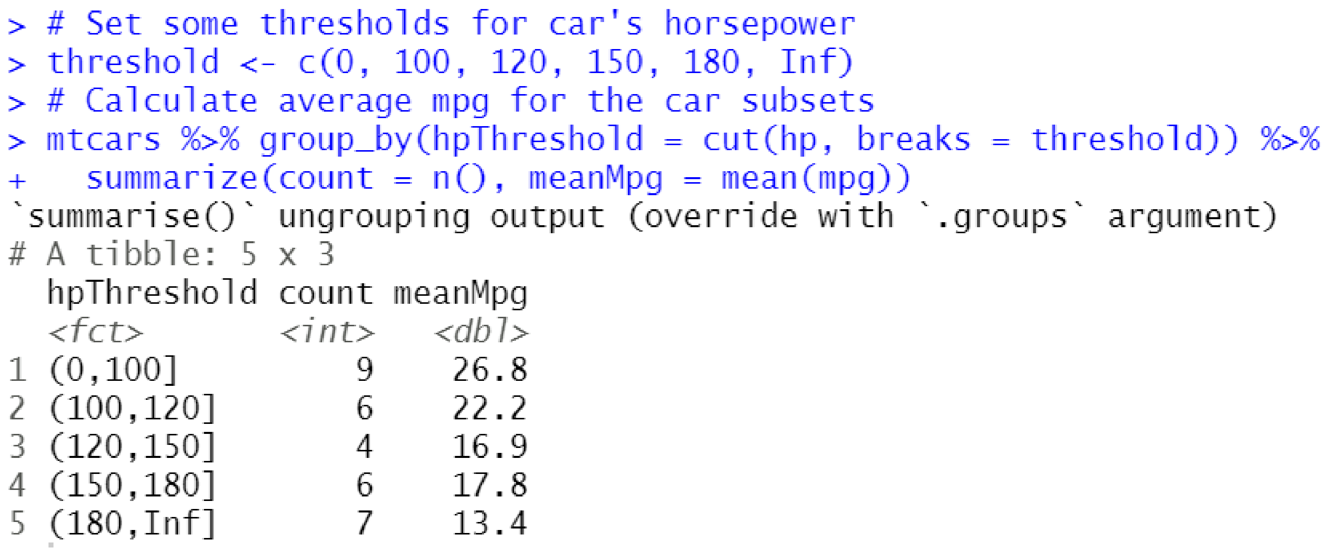
Count the number of rows in each group with tally()



```
> mtcars %>% group_by(carb) %>% tally()
# A tibble: 6 x 2
   carb     n
  <dbl> <int>
1     1     7
2     2    10
3     3     3
4     4    10
5     6     1
6     8     1
```

GROUP_BY

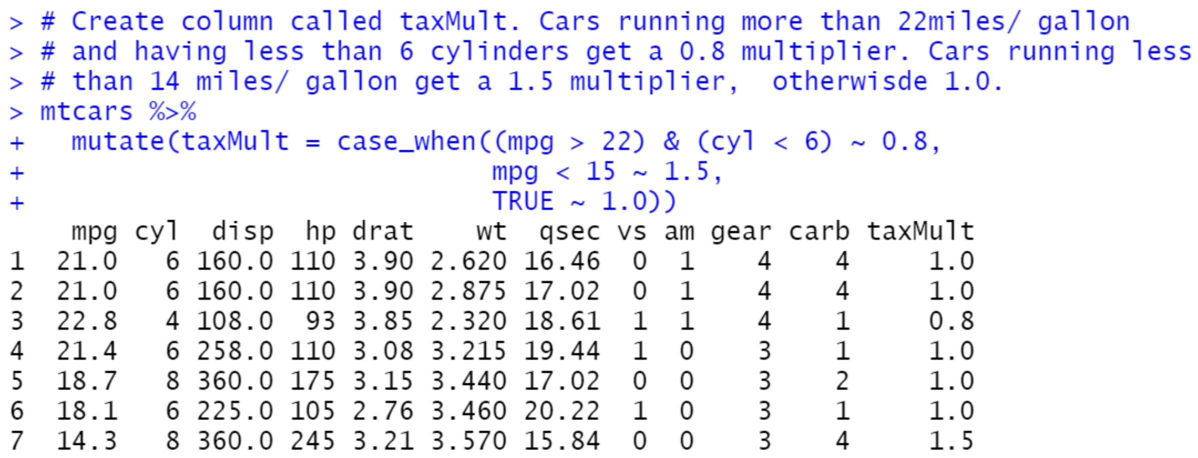
You can also effectively group numeric variables with cut()



```
> # Set some thresholds for car's horsepower
> threshold <- c(0, 100, 120, 150, 180, Inf)
> # Calculate average mpg for the car subsets
> mtcars %>% group_by(hpThreshold = cut(hp, breaks = threshold)) %>%
+   summarize(count = n(), meanMpg = mean(mpg))
`summarise()` ungrouping output (override with `.groups` argument)
# A tibble: 5 x 3
  hpThreshold count meanMpg
  <fct>       <int>   <dbl>
1 (0,100]         9    26.8
2 (100,120]       6    22.2
3 (120,150]       4    16.9
4 (150,180]       6    17.8
5 (180,Inf]       7    13.4
```

CASE_WHEN

- Vectorized IF statement
- Useful to label stuff based on conditions
- Condition can be formed using multiple columns



```
> # Create column called taxMult. Cars running more than 22miles/ gallon
> # and having less than 6 cylinders get a 0.8 multiplier. Cars running less
> # than 14 miles/ gallon get a 1.5 multiplier,  otherwisde 1.0.
> mtcars %>%
+   mutate(taxMult = case_when((mpg > 22) & (cyl < 6) ~ 0.8,
+                              mpg < 15 ~ 1.5,
+                              TRUE ~ 1.0))
   mpg cyl  disp  hp drat    wt  qsec vs am gear carb taxMult
1  21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4     1.0
2  21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4     1.0
3  22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1     0.8
4  21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1     1.0
5  18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2     1.0
6  18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1     1.0
7  14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4     1.5
```

EXERCISE