

# PairProgramming\_Slicing

HDS

2024-08-09

## Pair Programming Slicing

Using dplyr to manipulate, structure and sort data frames

See Wickham et al, Chapter 3

<https://r4ds.hadley.nz/data-transform> (<https://r4ds.hadley.nz/data-transform>)

HD Sheets, August 9, 2024 checked 01/03/2025

Student Name:Ryan Waterman Teamate Name:Nicholas Perry

#Topics- by class of operations

These are tools to regroup data for various purposes, including plotting and generating summary tables, such as “cross tables” or “pivot tables” easily

Row Manipulations

-filter() -arrange() -distinct()

Column Manipulations

-mutate() -select() -rename() -relocate()

Grouping

-group\_by()

Summarize or calculate

-summarize()

#Libraries

```
library('tidyverse')
```

```
## — Attaching core tidyverse packages — tidyverse 2.0.0 —
## ✓ dplyr      1.1.4      ✓ readr      2.1.5
## ✓ forcats   1.0.0      ✓ stringr    1.5.1
## ✓ ggplot2   3.5.1      ✓ tibble     3.2.1
## ✓ lubridate 1.9.4      ✓ tidyr      1.3.1
## ✓ purrr     1.0.2
## — Conflicts — tidyverse_conflicts() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library('ggplot2')
```

#example data

we will use the rather standard mtcars dataset of car performance data created by MotorTrend (MT) in the 1970s

Not super exciting, but small enough to work with

Read that help menu!

```
data(mtcars)
head(mtcars)
```

```
##           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
## Valiant        18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

```
help(mtcars)
```

```
## starting httpd help server ... done
```

#Examples of filtering

Selecting specific rows, based on some condition on a variable

Filter on one variable

```
# 8 cylinder cars only

filter(mtcars,cyl==8)
```

```
##           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
## 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
## Chrysler Imperial 14.7   8  440.0 230 3.23 5.345 17.42  0  0    3    4
## Dodge Challenger 15.5   8  318.0 150 2.76 3.520 16.87  0  0    3    2
## AMC Javelin      15.2   8  304.0 150 3.15 3.435 17.30  0  0    3    2
## Camaro Z28       13.3   8  350.0 245 3.73 3.840 15.41  0  0    3    4
## Pontiac Firebird 19.2   8  400.0 175 3.08 3.845 17.05  0  0    3    2
## Ford Pantera L   15.8   8  351.0 264 4.22 3.170 14.50  0  1    5    4
## Maserati Bora     15.0   8  301.0 335 3.54 3.570 14.60  0  1    5    8
```

#filter on a condition

using a pipeline %>% to feed the dataframe into the filter

looking for cars with less than 75 hp (yeah, really...)

```
mtcars %>%filter(hp<75)
```

```
##           mpg cyl  disp hp drat   wt  qsec vs am gear carb
## Merc 240D   24.4   4 146.7 62 3.69 3.190 20.00  1  0    4    2
## 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
## Toyota Corolla 33.9  4  71.1 65 4.22 1.835 19.90  1  1    4    1
## Fiat X1-9   27.3   4  79.0 66 4.08 1.935 18.90  1  1    4    1
```

#combined conditions

Let's find 6 cylinder cars with mpg>25

we use an AND operation "&" to combine the conditions (cyl==6) with (mpg>20)

```
mtcars %>%filter( (cyl==6)&(mpg>20))
```

```
##           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
## Hornet 4 Drive 21.4   6 258 110 3.08 3.215 19.44  1  0    3    1
```

Hmm, lousy fuel economy in the 70s for 6 cylinders.

*#Question/Action*

filter the cars with 6 cylinders and a manual transmission the column am is 0 for automatic transmissions and 1 for manuals

```
mtcars %>% filter((cyl==6)&(am==1))
```

```
##           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
## Ferrari Dino 19.7   6 145 175 3.62 2.770 15.50  0  1    5    6
```

## arrange()

This is a sorting function

sort by increasing mpg

```
mtcars %>% arrange(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
##	Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
##	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
##	AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
##	Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
##	Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	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 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
##	Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
##	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
##	Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
##	Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
##	Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
##	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
##	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
##	Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2
##	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
##	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
##	Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
##	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
##	Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
##	Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	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 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
##	Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1

decreasing mpgh

Put desc() around the variable

```
mtcars %>% arrange(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
## Porsche 914-2	26.0	4	120.3	91	4.43	2.140	16.70	0	1	5	2
## Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
## Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
## Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
## Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
## Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
## Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2
## 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
## Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
## Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
## Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
## 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
## Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
## Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
## Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
## Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
## Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
## Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
## AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
## Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
## Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
## Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
## Camaro Z28	13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4
## 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

##sorting on multiple variables

just include them all in the arrange(), using desc() as needed

Lets sort by cylinder (descending) and qsec time (increasing, low qsec time means a fast car)

This looks a lot like the sorting options in Excel

```
mtcars %>% arrange(desc(cyl),qsec)
```

##	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
## Ford Pantera L	15.8	8	351.0	264	4.22	3.170	14.50	0	1	5	4
## Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
## 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
## Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
## Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
## Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
## AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2
## Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
## Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
## Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
## Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
## Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
## Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
## Ferrari Dino	19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6
## 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
## 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
## Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
## Valiant	18.1	6	225.0	105	2.76	3.460	20.22	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
## Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
## Volvo 142E	21.4	4	121.0	109	4.11	2.780	18.60	1	1	4	2
## Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
## Fiat X1-9	27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1
## Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
## Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
## Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
## Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
## Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2

### #Question/Action

sort the cars by transmission type, mpg and number of cylinders

```
mtcars %>% arrange(am,mpg,cyl)
```

```
##          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
## Merc 450SLC        15.2   8 275.8 180 3.07 3.780 18.00  0  0   3   3
## AMC Javelin        15.2   8 304.0 150 3.15 3.435 17.30  0  0   3   2
## Dodge Challenger   15.5   8 318.0 150 2.76 3.520 16.87  0  0   3   2
## 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 280C          17.8   6 167.6 123 3.92 3.440 18.90  1  0   4   4
## Valiant            18.1   6 225.0 105 2.76 3.460 20.22  1  0   3   1
## Hornet Sportabout  18.7   8 360.0 175 3.15 3.440 17.02  0  0   3   2
## Merc 280           19.2   6 167.6 123 3.92 3.440 18.30  1  0   4   4
## Pontiac Firebird   19.2   8 400.0 175 3.08 3.845 17.05  0  0   3   2
## 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
## Merc 230           22.8   4 140.8  95 3.92 3.150 22.90  1  0   4   2
## Merc 240D          24.4   4 146.7  62 3.69 3.190 20.00  1  0   4   2
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  1   5   8
## Ford Pantera L     15.8   8 351.0 264 4.22 3.170 14.50  0  1   5   4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  1   5   6
## 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
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  1   4   2
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  1   4   1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  1   5   2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  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 128            32.4   4  78.7  66 4.08 2.200 19.47  1  1   4   1
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  1   4   1
```

## distinct, finding unique instances

of cylinders

This allows use to determine how many distinct or unique values we have in a column

```
mtcars %>% distinct(cyl)
```

```
##          cyl
## Mazda RX4      6
## Datsun 710      4
## Hornet Sportabout 8
```

Let's look at combinations of cylinders and carbeurators

note the alternate form of the pipe |> is easier to type

```
mtcars |> distinct(cyl,carb)
```

```
##           cyl carb
## Mazda RX4      6   4
## Datsun 710      4   1
## Hornet 4 Drive  6   1
## Hornet Sportabout 8   2
## Duster 360      8   4
## Merc 240D       4   2
## Merc 450SE      8   3
## Ferrari Dino    6   6
## Maserati Bora   8   8
```

Okay, but I want it ordered by cyl

we just pipeline the arrange() after the distinct

```
mtcars |> distinct(cyl,carb) |> arrange(cyl,carb)
```

```
##           cyl carb
## Datsun 710      4   1
## Merc 240D       4   2
## Hornet 4 Drive  6   1
## Mazda RX4      6   4
## Ferrari Dino    6   6
## Hornet Sportabout 8   2
## Merc 450SE      8   3
## Duster 360      8   4
## Maserati Bora   8   8
```

That is much easier to read

### *#Question/Action*

find the unique combinations of gear and carb and am appearing in the data set

```
distinct_set <- mtcars |> distinct(gear, carb, am)
distinct_set
```



```
##           gear carb am
## Mazda RX4      4    4  1
## Datsun 710     4    1  1
## Hornet 4 Drive  3    1  0
## Hornet Sportabout 3    2  0
## Duster 360     3    4  0
## Merc 240D      4    2  0
## Merc 280       4    4  0
## Merc 450SE     3    3  0
## Honda Civic    4    2  1
## Porsche 914-2  5    2  1
## Ford Pantera L  5    4  1
## Ferrari Dino   5    6  1
## Maserati Bora  5    8  1
```

sort it by increasing am, decreasing gear, decreasing carb d

```
sorted_distinct_set <- distinct_set |> arrange(am, desc(gear), desc(carb))
sorted_distinct_set
```

```
##           gear carb am
## Merc 280      4    4  0
## Merc 240D     4    2  0
## Duster 360    3    4  0
## Merc 450SE    3    3  0
## Hornet Sportabout 3    2  0
## Hornet 4 Drive 3    1  0
## Maserati Bora  5    8  1
## Ferrari Dino   5    6  1
## Ford Pantera L  5    4  1
## Porsche 914-2  5    2  1
## Mazda RX4      4    4  1
## Honda Civic    4    2  1
## Datsun 710     4    1  1
```

Do this in two steps, get distinct working first, then arrange() it

## Column based Manipulation

#mutate

Alters columns, by adding new ones

Here is the power to weight (ptw) ratio

Note that here the altered version of mtcars is stored back into mtcars, so the change in mtcars is kep

```
#I accidentally ran this block and the next multiple times, so I was getting an error. Reloading
mtcars fixes it.
data(mtcars)
mtcars=mtcars |> mutate(ptw=hp/wt)
```

## #rename

renames a column

```
mtcars=mtcars |> rename(power2weight=ptw)

head(mtcars)
```

```
##           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
## Valiant        18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
##           power2weight
## Mazda RX4      41.98473
## Mazda RX4 Wag  38.26087
## Datsun 710     40.08621
## Hornet 4 Drive 34.21462
## Hornet Sportabout 50.87209
## Valiant        30.34682
```

## relocate

moves the location of a column

the default is to move variables to the front .before and .after can alter the location relative to other named columns

```
mtcars=mtcars |> relocate(power2weight, .after=hp)
head(mtcars)
```

```
##           mpg cyl disp  hp power2weight drat   wt  qsec vs am gear
## Mazda RX4      21.0   6  160 110      41.98473 3.90 2.620 16.46 0  1   4
## Mazda RX4 Wag  21.0   6  160 110      38.26087 3.90 2.875 17.02 0  1   4
## Datsun 710      22.8   4  108  93      40.08621 3.85 2.320 18.61 1  1   4
## Hornet 4 Drive  21.4   6  258 110      34.21462 3.08 3.215 19.44 1  0   3
## Hornet Sportabout 18.7   8  360 175      50.87209 3.15 3.440 17.02 0  0   3
## Valiant         18.1   6  225 105      30.34682 2.76 3.460 20.22 1  0   3
##               carb
## Mazda RX4          4
## Mazda RX4 Wag      4
## Datsun 710          1
## Hornet 4 Drive      1
## Hornet Sportabout   2
## Valiant             1
```

### #Question/Action

Create a new column called `specific_output`

`specific_output` is hp divided by displacement

Place this behind `power2weight` in the table

Use `head()` to show this has worked.

```
mtcars=mtcars |> mutate(specific_output=hp/displ)
mtcars=mtcars |> relocate(specific_output, .after=power2weight)
head(mtcars)
```

```
##           mpg cyl disp  hp power2weight specific_output drat   wt
## Mazda RX4      21.0   6  160 110      41.98473      0.6875000 3.90 2.620
## Mazda RX4 Wag  21.0   6  160 110      38.26087      0.6875000 3.90 2.875
## Datsun 710      22.8   4  108  93      40.08621      0.8611111 3.85 2.320
## Hornet 4 Drive  21.4   6  258 110      34.21462      0.4263566 3.08 3.215
## Hornet Sportabout 18.7   8  360 175      50.87209      0.4861111 3.15 3.440
## Valiant         18.1   6  225 105      30.34682      0.4666667 2.76 3.460
##               qsec vs am gear carb
## Mazda RX4      16.46 0  1   4   4
## Mazda RX4 Wag  17.02 0  1   4   4
## Datsun 710      18.61 1  1   4   1
## Hornet 4 Drive  19.44 1  0   3   1
## Hornet Sportabout 17.02 0  0   3   2
## Valiant         20.22 1  0   3   1
```

## Select

Just choices a set of columns

```
mtcars |> select(cyl,carb,gear)
```

##	cyl	carb	gear
## Mazda RX4	6	4	4
## Mazda RX4 Wag	6	4	4
## Datsun 710	4	1	4
## Hornet 4 Drive	6	1	3
## Hornet Sportabout	8	2	3
## Valiant	6	1	3
## Duster 360	8	4	3
## Merc 240D	4	2	4
## Merc 230	4	2	4
## Merc 280	6	4	4
## Merc 280C	6	4	4
## Merc 450SE	8	3	3
## Merc 450SL	8	3	3
## Merc 450SLC	8	3	3
## Cadillac Fleetwood	8	4	3
## Lincoln Continental	8	4	3
## Chrysler Imperial	8	4	3
## Fiat 128	4	1	4
## Honda Civic	4	2	4
## Toyota Corolla	4	1	4
## Toyota Corona	4	1	3
## Dodge Challenger	8	2	3
## AMC Javelin	8	2	3
## Camaro Z28	8	4	3
## Pontiac Firebird	8	2	3
## Fiat X1-9	4	1	4
## Porsche 914-2	4	2	5
## Lotus Europa	4	2	5
## Ford Pantera L	8	4	5
## Ferrari Dino	6	6	5
## Maserati Bora	8	8	5
## Volvo 142E	4	2	4

### *#Question/Action*

Select specific\_output, power2weight and qsec

```
mtcars |> select(specific_output, power2weight, qsec)
```

```
##           specific_output power2weight  qsec
## Mazda RX4           0.6875000      41.98473 16.46
## Mazda RX4 Wag       0.6875000      38.26087 17.02
## Datsun 710           0.8611111      40.08621 18.61
## Hornet 4 Drive      0.4263566      34.21462 19.44
## Hornet Sportabout   0.4861111      50.87209 17.02
## Valiant              0.4666667      30.34682 20.22
## Duster 360          0.6805556      68.62745 15.84
## Merc 240D           0.4226312      19.43574 20.00
## Merc 230            0.6747159      30.15873 22.90
## Merc 280            0.7338902      35.75581 18.30
## Merc 280C           0.7338902      35.75581 18.90
## Merc 450SE          0.6526468      44.22604 17.40
## Merc 450SL          0.6526468      48.25737 17.60
## Merc 450SLC         0.6526468      47.61905 18.00
## Cadillac Fleetwood  0.4343220      39.04762 17.98
## Lincoln Continental 0.4673913      39.63864 17.82
## Chrysler Imperial   0.5227273      43.03087 17.42
## Fiat 128             0.8386277      30.00000 19.47
## Honda Civic          0.6869221      32.19814 18.52
## Toyota Corolla       0.9142053      35.42234 19.90
## Toyota Corona        0.8076603      39.35091 20.01
## Dodge Challenger     0.4716981      42.61364 16.87
## AMC Javelin          0.4934211      43.66812 17.30
## Camaro Z28           0.7000000      63.80208 15.41
## Pontiac Firebird     0.4375000      45.51365 17.05
## Fiat X1-9            0.8354430      34.10853 18.90
## Porsche 914-2        0.7564422      42.52336 16.70
## Lotus Europa         1.1882229      74.68605 16.90
## Ford Pantera L       0.7521368      83.28076 14.50
## Ferrari Dino         1.2068966      63.17690 15.50
## Maserati Bora        1.1129568      93.83754 14.60
## Volvo 142E          0.9008264      39.20863 18.60
```

## #Groups and summarize

Summarize summarizes a specific statistic

We often want to group by one or more variables, then summarize

it looks like this

group\_by() by cylinders and then compute mean mpg using summarize

```
mtcars |>group_by(cyl) |> summarize(mean(hp))
```

```
## # A tibble: 3 × 2
##   cyl `mean(hp)`
##   <dbl>     <dbl>
## 1     4       82.6
## 2     6      122.
## 3     8      209.
```

we could then sort this by decreasing cylinders

```
mtcars |> group_by(cyl) |> summarize(mean(hp)) |> arrange(desc(cyl))
```

```
## # A tibble: 3 × 2
##   cyl `mean(hp)`
##   <dbl>     <dbl>
## 1     8     209.
## 2     6     122.
## 3     4     82.6
```

We can group by multiple variables

and compute multiple outputs in the summary

```
mtcars |> group_by(cyl,am) |> summarize (meanhp=mean(hp), meanpt2=mean(power2weight))|> arrange(d
esc(cyl))
```

```
## `summarise()` has grouped output by 'cyl'. You can override using the `.groups`
## argument.
```

```
## # A tibble: 6 × 4
## # Groups:   cyl [3]
##   cyl   am meanhp meanpt2
##   <dbl> <dbl> <dbl> <dbl>
## 1     8     0  194.    48.1
## 2     8     1  300.    88.6
## 3     6     0  115.    34.0
## 4     6     1  132.    47.8
## 5     4     0   84.7    29.6
## 6     4     1   81.9    41.0
```

This ability to combine group\_by and summarize allows us to generate all kinds of useful tables

Group\_by cyl and carbs, show the mean weight and mean mpg

```
mtcars |> group_by(cyl,carb) |> summarize (meanwt=mean(wt), meanmpg=mean(mpg))
```

```
## `summarise()` has grouped output by 'cyl'. You can override using the `.groups`
## argument.
```

```
## # A tibble: 9 × 4
## # Groups:   cyl [3]
##   cyl carb meanwt meanmpg
##   <dbl> <dbl> <dbl> <dbl>
## 1     4     1  2.15   27.6
## 2     4     2  2.40   25.9
## 3     6     1  3.34   19.8
## 4     6     4  3.09   19.8
## 5     6     6  2.77   19.7
## 6     8     2  3.56   17.2
## 7     8     3  3.86   16.3
## 8     8     4  4.43   13.2
## 9     8     8  3.57   15
```

### #Question/Action

Group by carb and cyl and find the mean power2weight ratio for each grouping

```
mtcars |> group_by(carb, cyl) |> summarize (mean_power2weight=mean(power2weight))
```

```
## `summarise()` has grouped output by 'carb'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 9 × 3
## # Groups:   carb [6]
##   carb cyl mean_power2weight
##   <dbl> <dbl> <dbl>
## 1     1     4           35.8
## 2     1     6           32.3
## 3     2     4           39.7
## 4     2     8           45.7
## 5     3     8           46.7
## 6     4     6           37.9
## 7     4     8           56.2
## 8     6     6           63.2
## 9     8     8           93.8
```

### #Counts

n=n() will give us the counts in the summary

```
mtcars |> group_by(cyl,am) |>summarize (n=n())
```

```
## `summarise()` has grouped output by 'cyl'. You can override using the `.groups`
## argument.
```

```
## # A tibble: 6 × 3
## # Groups:   cyl [3]
##   cyl    am    n
##   <dbl> <dbl> <int>
## 1     4     0     3
## 2     4     1     8
## 3     6     0     4
## 4     6     1     3
## 5     8     0    12
## 6     8     1     2
```

we can also ask for the count of a variable to find n as well

```
mtcars |> group_by(cyl,am) |> summarize(n=length(cyl)) |> arrange(desc(cyl))
```

```
## `summarise()` has grouped output by 'cyl'. You can override using the `.groups`
## argument.
```

```
## # A tibble: 6 × 3
## # Groups:   cyl [3]
##   cyl    am    n
##   <dbl> <dbl> <int>
## 1     8     0    12
## 2     8     1     2
## 3     6     0     4
## 4     6     1     3
## 5     4     0     3
## 6     4     1     8
```

## Question/Action

Make a table that shows both counts and mean horse power, arranging by carb and gear sort in descending carb and gear order

```
mtcars |> group_by(carb,gear) |> summarize(n=n(), mean_hp = mean(hp)) |> arrange(desc(carb), gear)
```

```
## `summarise()` has grouped output by 'carb'. You can override using the
## `.groups` argument.
```



```
## # A tibble: 11 × 4
## # Groups:   carb [6]
##   carb gear    n mean_hp
##   <dbl> <dbl> <int>   <dbl>
## 1     8     5     1    335
## 2     6     5     1    175
## 3     4     3     5    228
## 4     4     4     4    116.
## 5     4     5     1    264
## 6     3     3     3    180
## 7     2     3     4    162.
## 8     2     4     4     79.5
## 9     2     5     2    102
## 10    1     3     3    104
## 11    1     4     4     72.5
```

## finding specific rows

df |> slice\_head(n = 1) takes the first row from each group. df |> slice\_tail(n = 1) takes the last row in each group. df |> slice\_min(x, n = 1) takes the row with the smallest value of column x. df |> slice\_max(x, n = 1) takes the row with the largest value of column x. df |> slice\_sample(n = 1) takes one random row.

Find the max hp

```
mtcars |> slice_max(hp,n=1)
```

```
##           mpg cyl disp  hp power2weight specific_output drat   wt  qsec  vs
## Maserati Bora  15   8  301 335      93.83754      1.112957 3.54 3.57 14.6  0
##           am gear carb
## Maserati Bora   1   5   8
```

### #Question/Action

Find the min power2weight values for groupings by cylinder

```
mtcars |> group_by(cyl) |> slice_min(power2weight,n=1)
```

```
## # A tibble: 3 × 13
## # Groups:   cyl [3]
##   mpg   cyl  disp    hp  power2weight specific_output  drat   wt  qsec    vs
##   <dbl> <dbl> <dbl> <dbl>        <dbl>        <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  24.4     4  147.    62         19.4          0.423  3.69  3.19   20     1
## 2  18.1     6  225   105         30.3          0.467  2.76  3.46  20.2   1
## 3  10.4     8  472   205         39.0          0.434  2.93  5.25  18.0   0
## # i 3 more variables: am <dbl>, gear <dbl>, carb <dbl>
```