Reproducible research

using tools from the tidyverse

Henrik Skov Midtiby, October 28th, 2025

SDU

Reproducible research

using tools from the tidyverse

Scientific Methods

Henrik Skov Midtiby, University of Southern Denmark

October 2025

SDU

Lecture outline

Why reproducible research?

Some papers are not reproducible!

A 2016 poll of 1,500 scientists reported that 70% of them had failed to reproduce at least one other scientist's experiment (50% had failed to reproduce one of their own experiments).

Source: Nature Video (28 May 2016). "Is There a Reproducibility Crisis in Science?". Scientific American. Retrieved 15 August 2019.

SDU

It is not even uncommon …

A 2016 study in the journal Science found that one-third of 18 experimental studies from two top-tier economics journals (American Economic Review and the Quarterly Journal of Economics) failed to successfully replicate.

Source:

My experience

I use reproducible methods in my daily work and when I work on the analysis part of a paper.

Look at the document with statistics from the midterm evaluation in EMAIP and RobtekMat 2023.

/home/hemi/Nextcloud/Work/06 Papers/2023-11-08 DUT Omkring BTC/midtvejsevaluering-kodet.Rmd

How can you benefit from using reproducible research?

It becomes much easier to look back at an old analysis and actually redo it from scratch.

It also allows you to modify the analysis, if an error is discovered or a new thing should be investigated.

Why is data visualisation important?

A good visualisation will show you things that you did not expect, or raise new questions about the data.

A good visualisation might also hint that you're asking the wrong question, or you need to collect different data.

Visualisations can surprise you, but don't scale particularly well because they require a human to interpret them.

Anscombe’s quartet – four data sets

Mean x value of 9.

Sample variance of x: $s_x^2 = 11$

Mean of y 7.50 to 2 decimal places

Sample variance of y: $s_y^2 = 4.125 \pm 0.003$

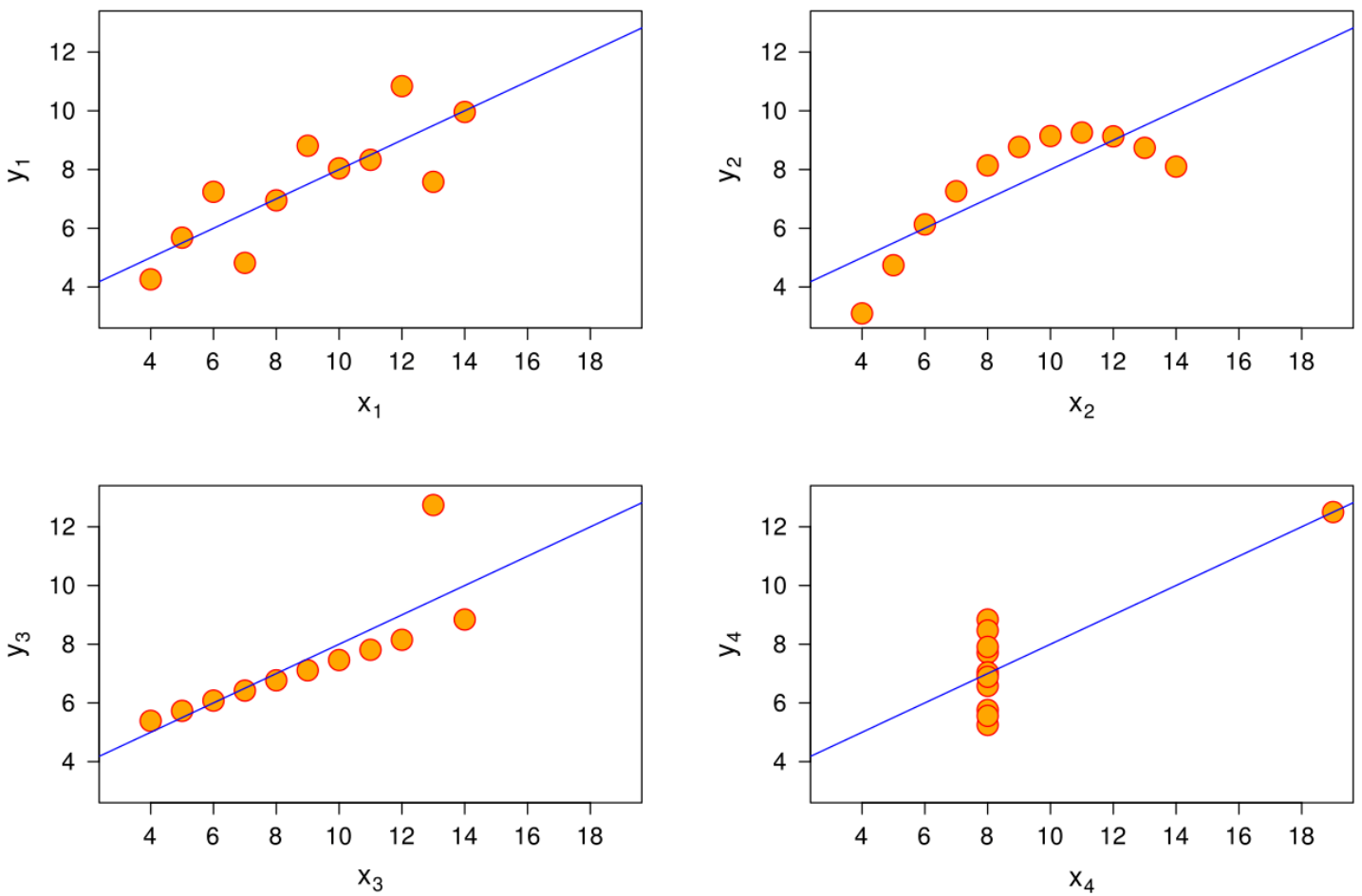
Correlation between x and y: 0.816 to 3 decimal places

Linear regression line: y = 3.00 + 0.500x to 2 and 3 decimal places, respectively

Coefficient of determination of the linear regression:

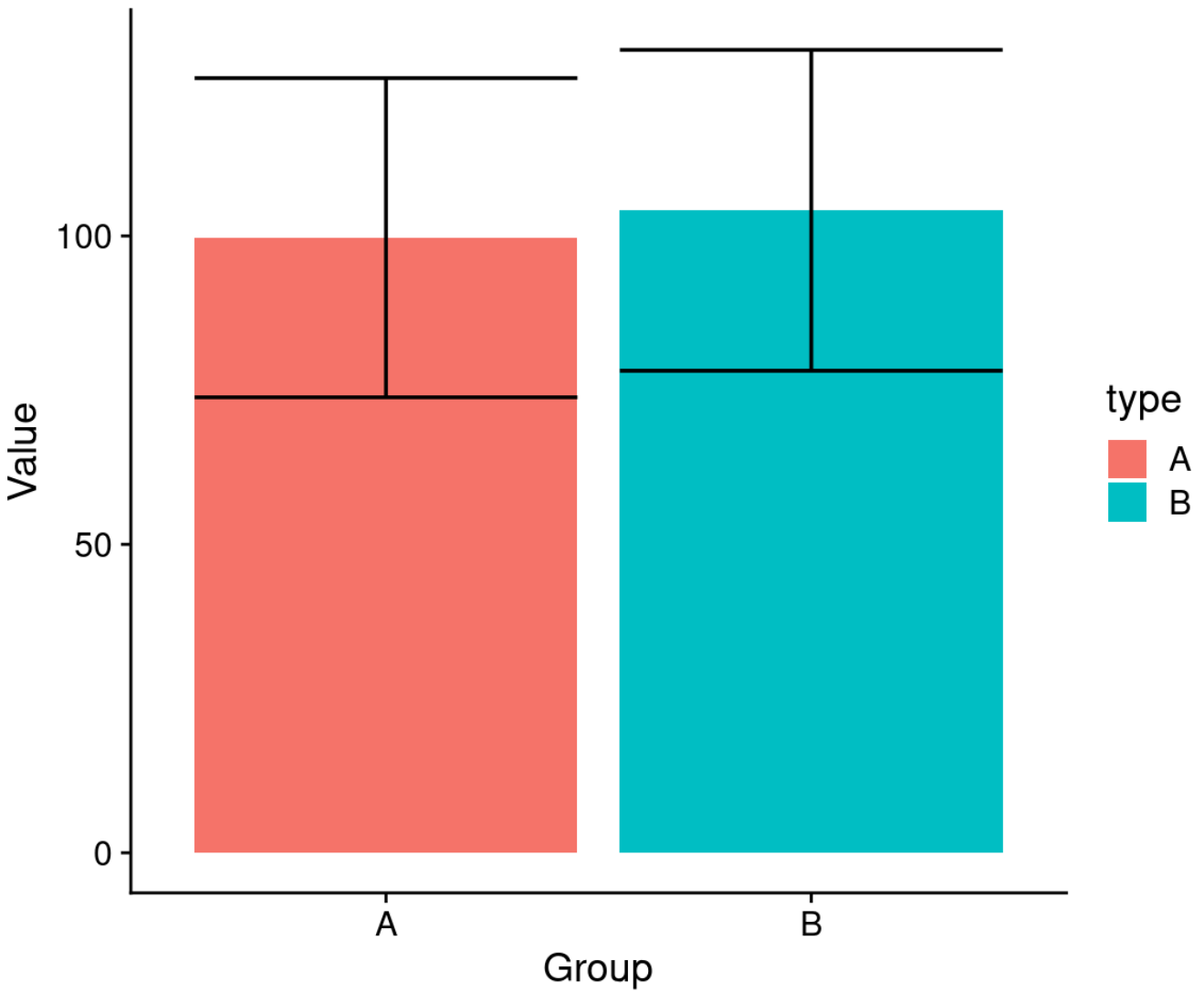
$R^2 = 0.67$ to 2 decimal places

Anscombe's quartet

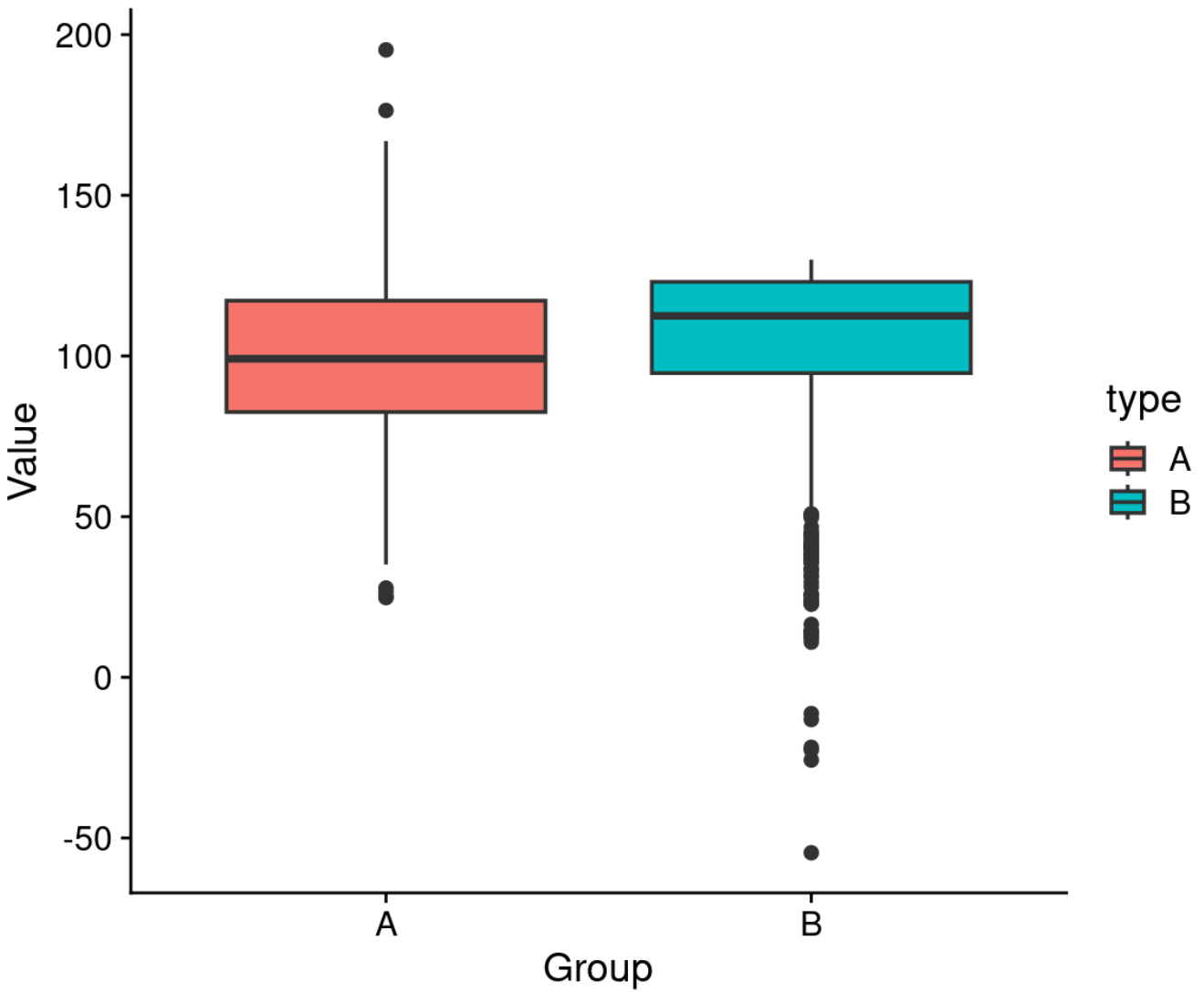


Not all data visualizations work equally well

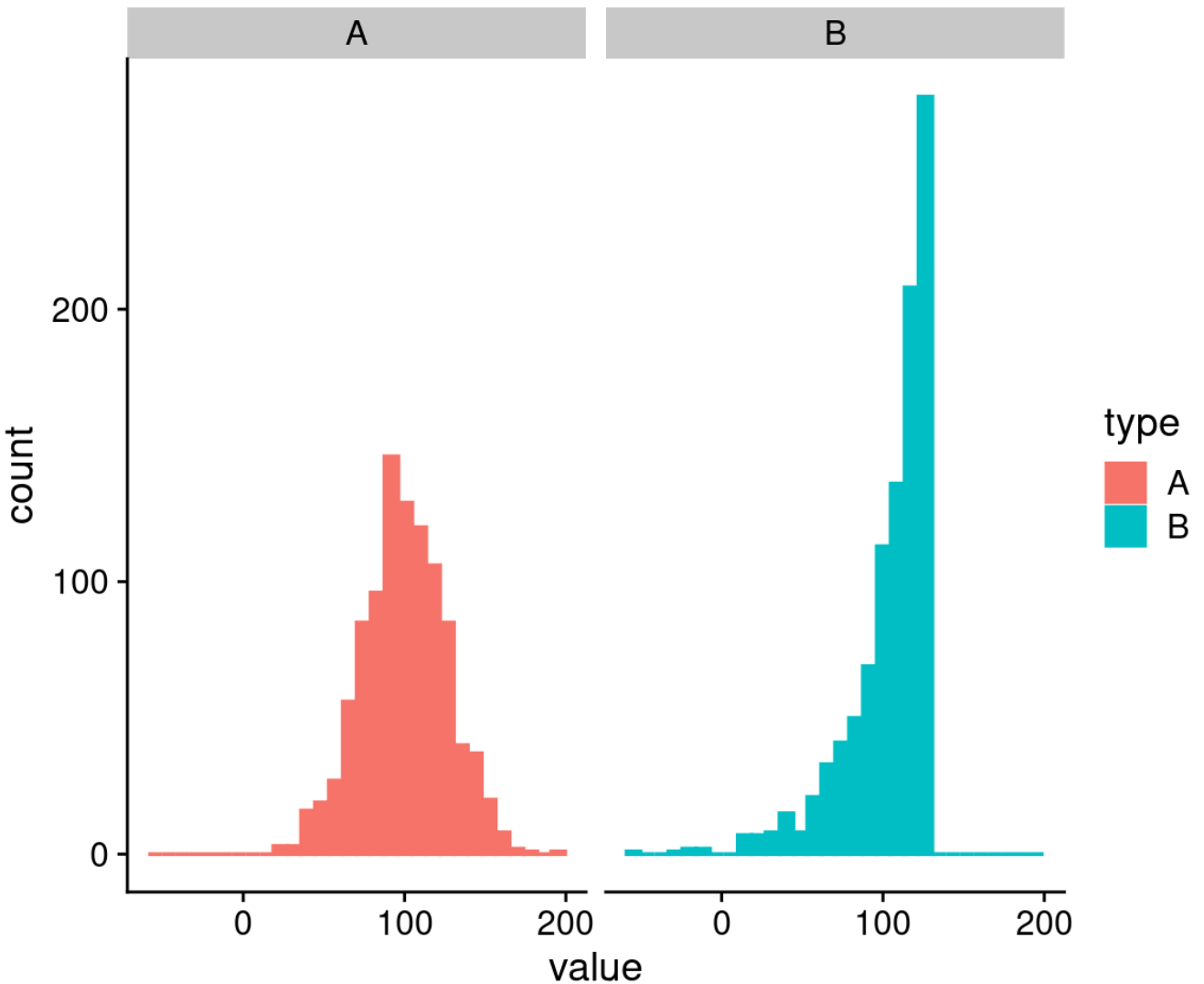
Barplot with errorbars



Boxplot

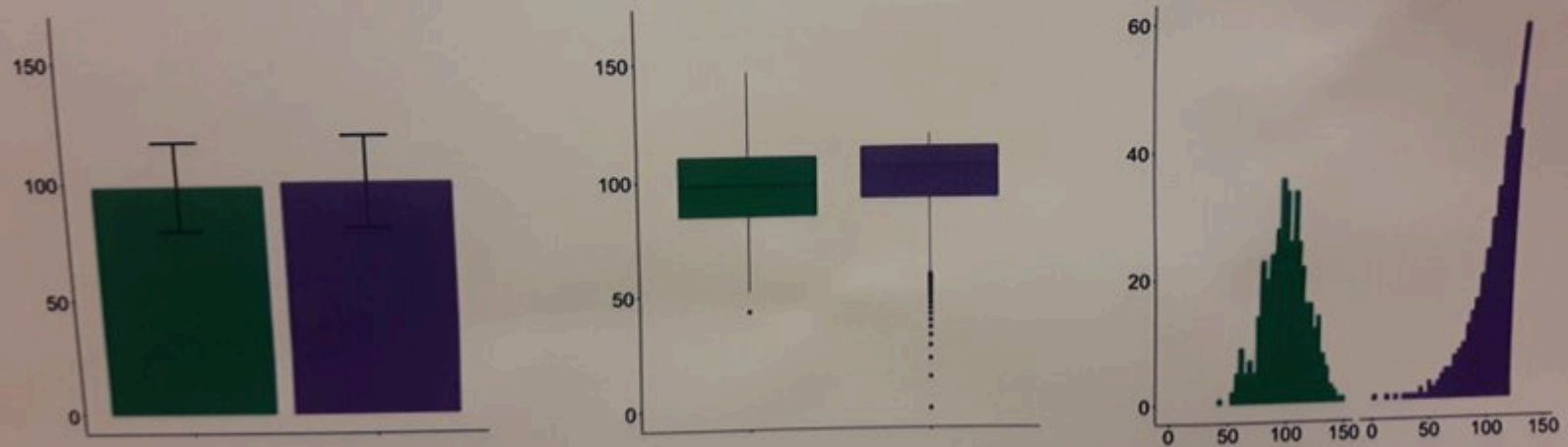


Histogram

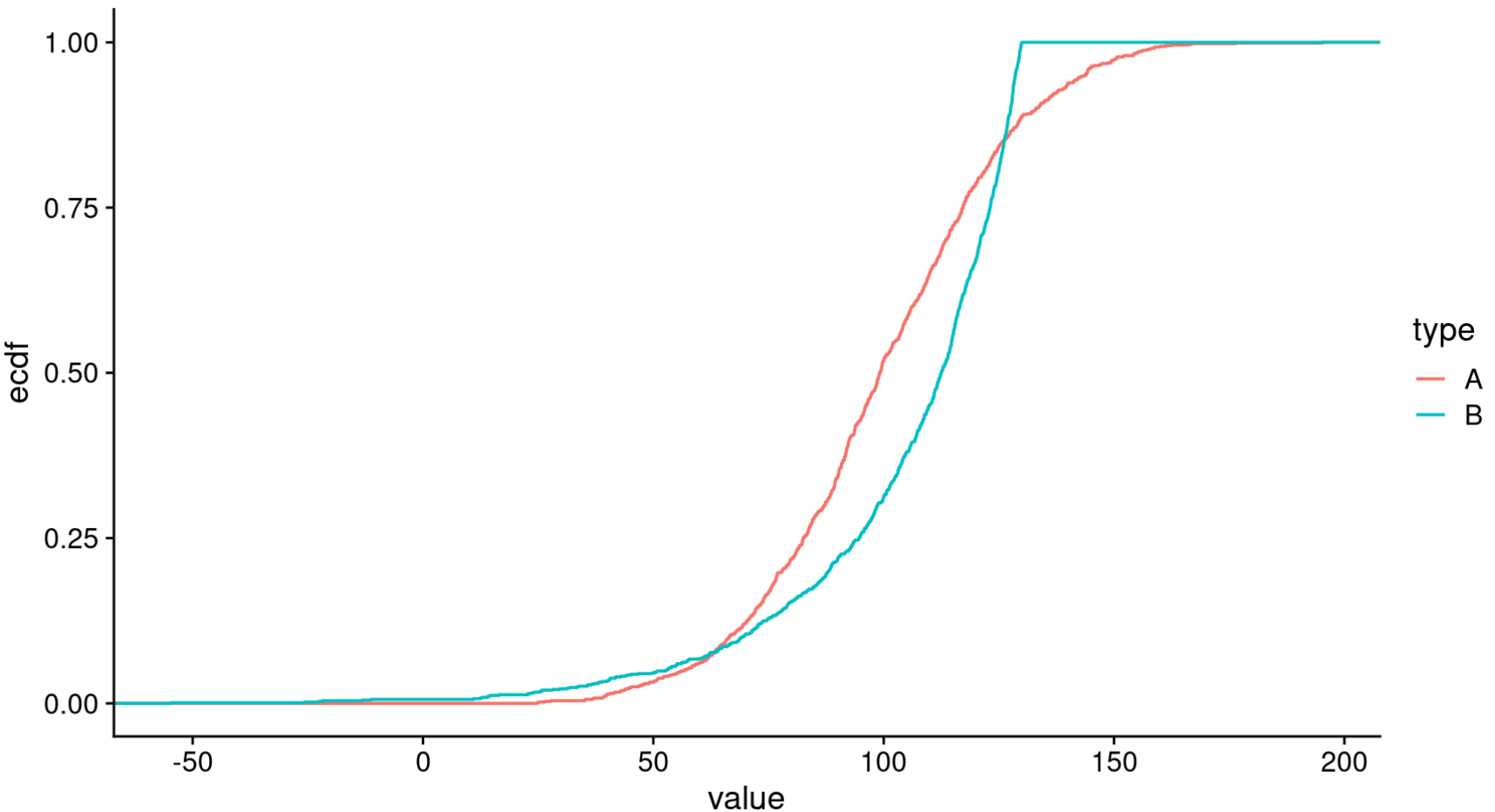


Friends don't let friends make bar plots.

These look the same! Wait a minute… Oooh!



Empirical Cumulative Density Function (ECDF)



SDU

RMarkdown

With RMarkdown you can *format* and **emphasize** your text and combine it with source code in eg. R or python.

–

RMarkdown is a tool for literate programming.

–

Read more about RMarkdown here:

- Chapter 1 R Markdown

Demonstration of ggplot

SDU

Demonstration of ggplot

```
1 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
```

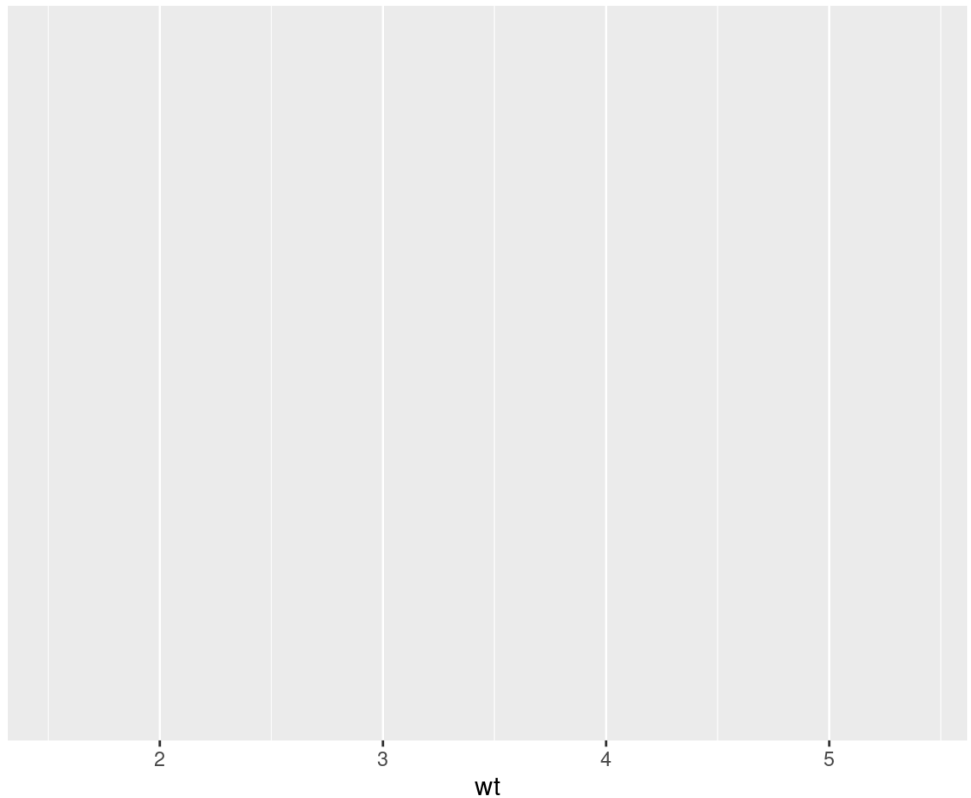
SDU

Demonstration of ggplot

```
mtcars %>%
  ggplot()
```

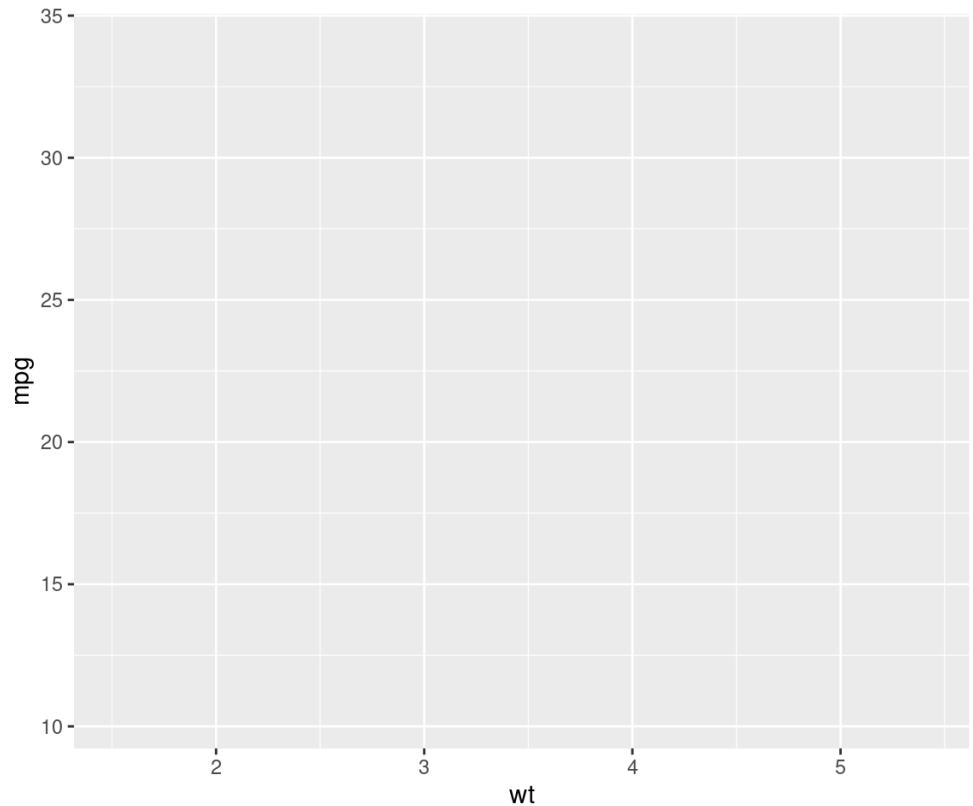
Demonstration of ggplot

```
1 mtcars %>%
2   ggplot() +
3   aes(x = wt)
```



Demonstration of ggplot

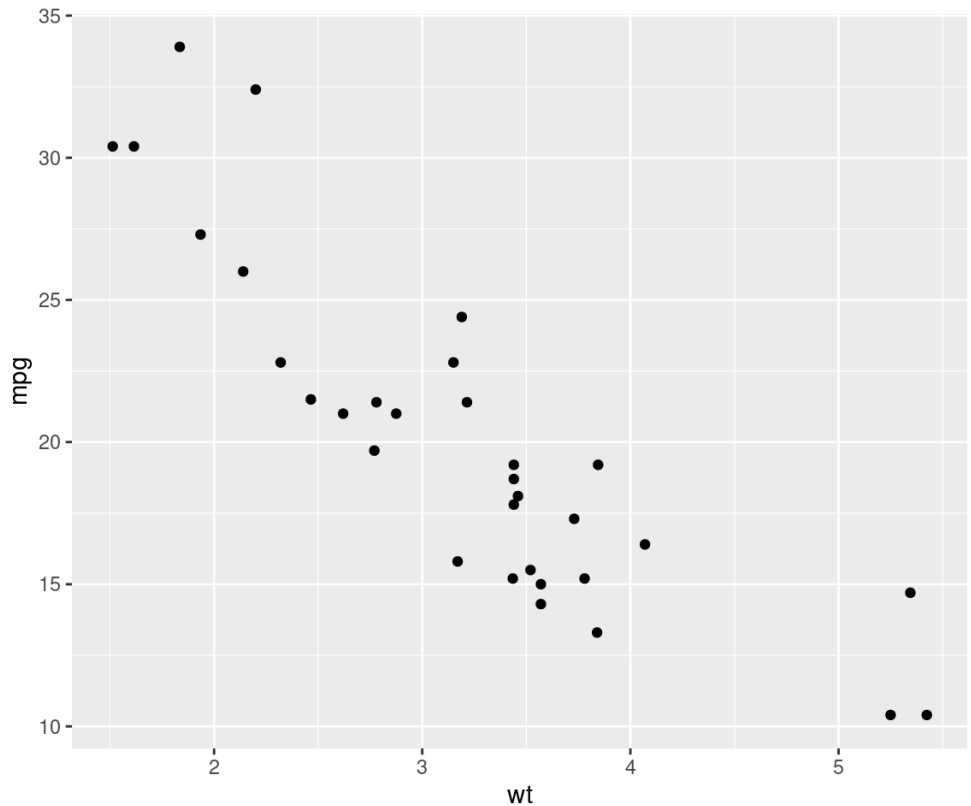
```
1 mtcars %>%
2   ggplot() +
3   aes(x = wt) +
4   aes(y = mpg)
```



SDU

Demonstration of ggplot

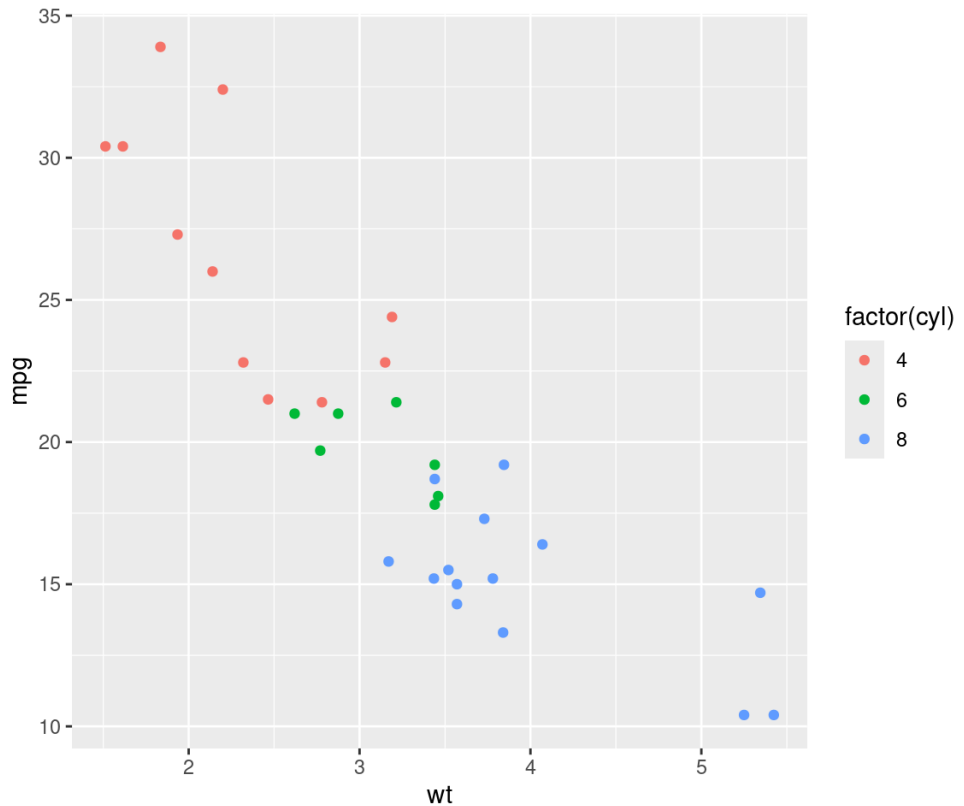
```
mtcars %>%
  ggplot() +
  aes(x = wt) +
  aes(y = mpg) +
  geom_point()
```



SDU

Demonstration of ggplot

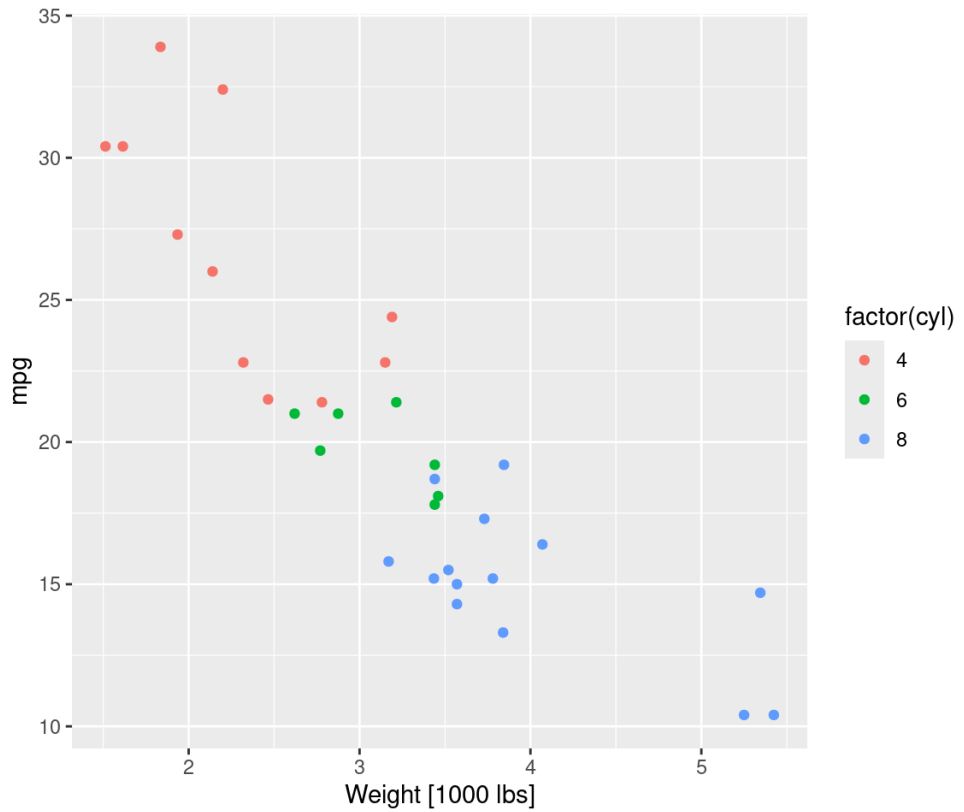
```
mtcars %>%
  ggplot() +
  aes(x = wt) +
  aes(y = mpg) +
  geom_point() +
  aes(color = factor(cyl))
```



SDU

Demonstration of ggplot

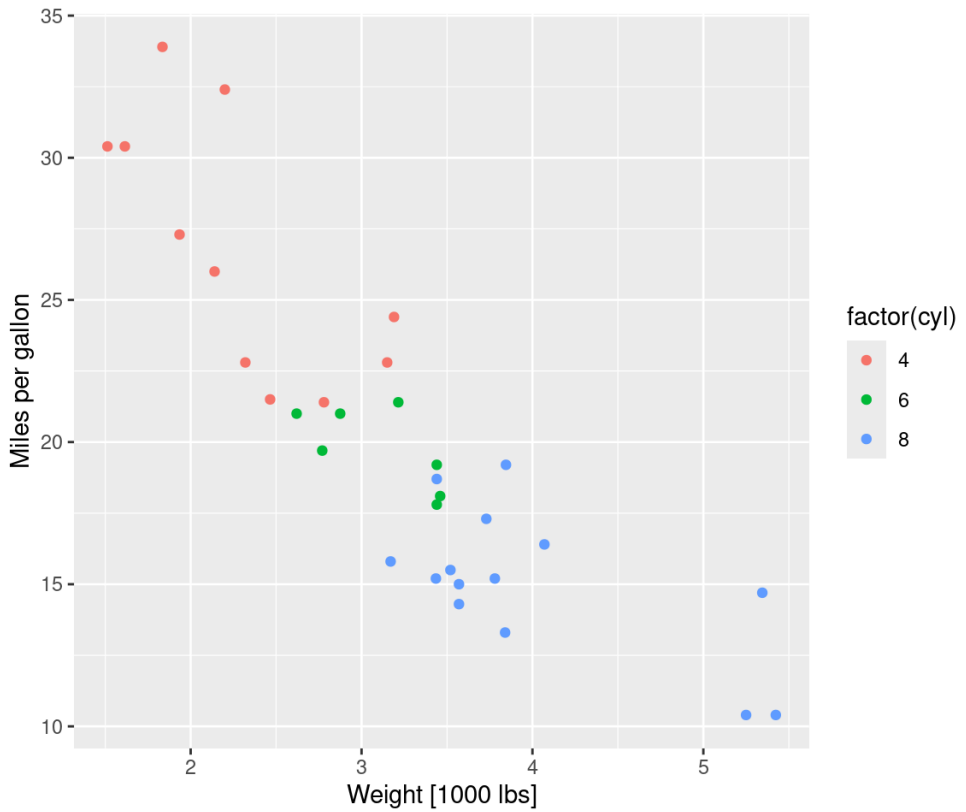
```
mtcars %>%
  ggplot() +
  aes(x = wt) +
  aes(y = mpg) +
  geom_point() +
  aes(color = factor(cyl)) +
  labs(x = 'Weight [1000 lbs]')
```



SDU

Demonstration of ggplot

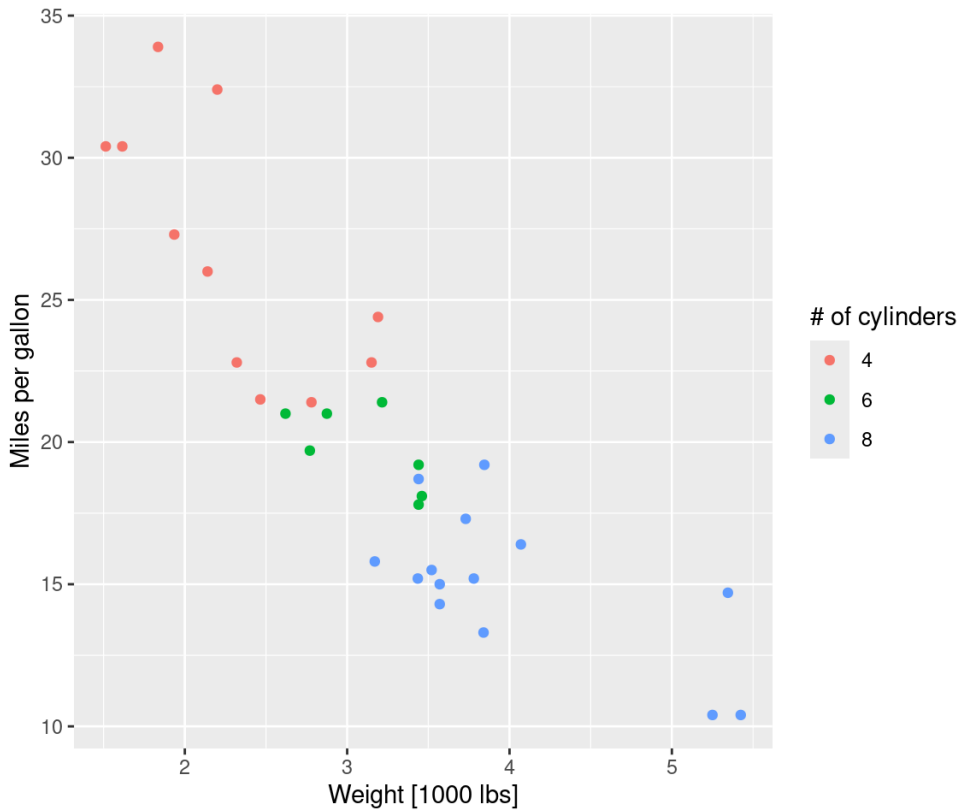
```
mtcars %>%
  ggplot() +
  aes(x = wt) +
  aes(y = mpg) +
  geom_point() +
  aes(color = factor(cyl)) +
  labs(x = 'Weight [1000 lbs]') +
  labs(y = "Miles per gallon")
```



SDU

Demonstration of ggplot

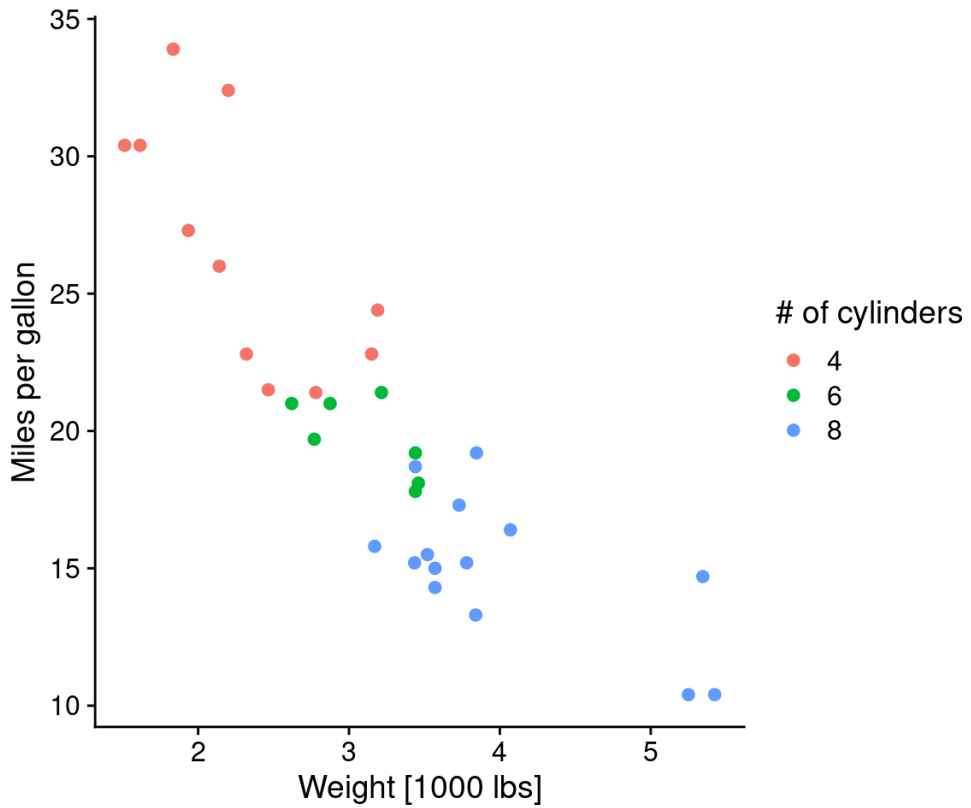
```
mtcars %>%
  ggplot() +
  aes(x = wt) +
  aes(y = mpg) +
  geom_point() +
  aes(color = factor(cyl)) +
  labs(x = 'Weight [1000 lbs]') +
  labs(y = "Miles per gallon") +
  labs(color = "# of cylinders")
```



SDU

Demonstration of ggplot

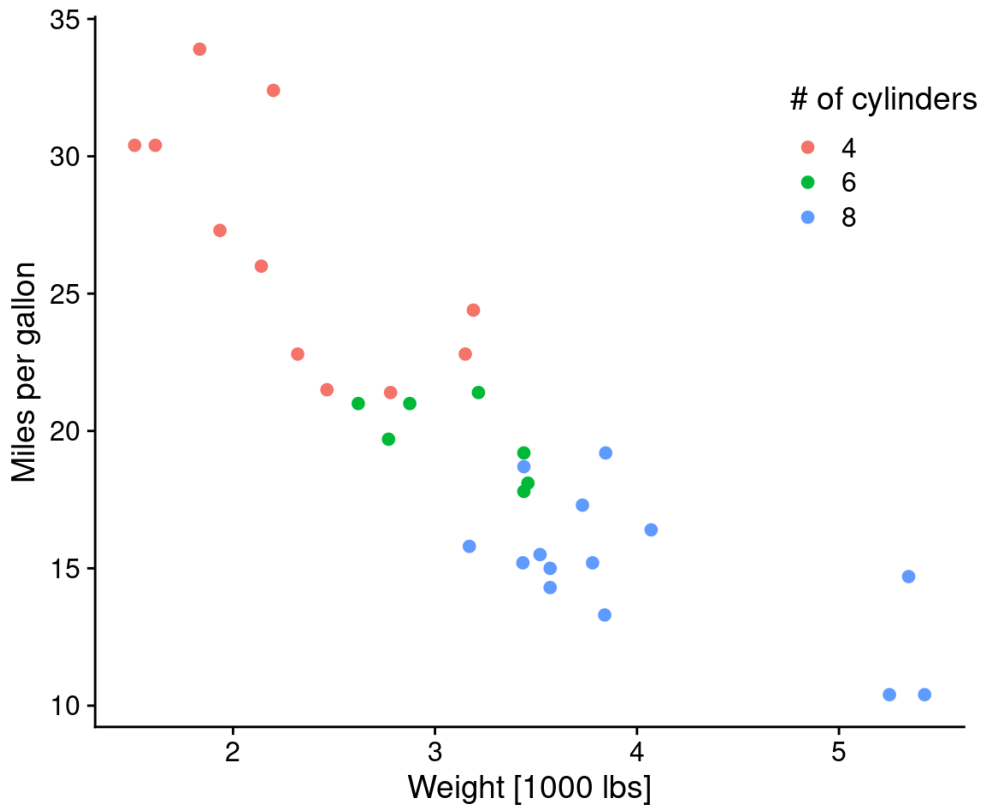
```
mtcars %>%
  ggplot() +
  aes(x = wt) +
  aes(y = mpg) +
  geom_point() +
  aes(color = factor(cyl)) +
  labs(x = 'Weight [1000 lbs]') +
  labs(y = "Miles per gallon") +
  labs(color = "# of cylinders") +
  theme_cowplot()
```



SDU

Demonstration of ggplot

```
mtcars %>%
  ggplot() +
  aes(x = wt) +
  aes(y = mpg) +
  geom_point() +
  aes(color = factor(cyl)) +
  labs(x = 'Weight [1000 lbs]') +
  labs(y = "Miles per gallon") +
  labs(color = "# of cylinders") +
  theme_cowplot() +
  theme(legend.position = c(0.8, 0.8))
```



SDU

```r
cars
```

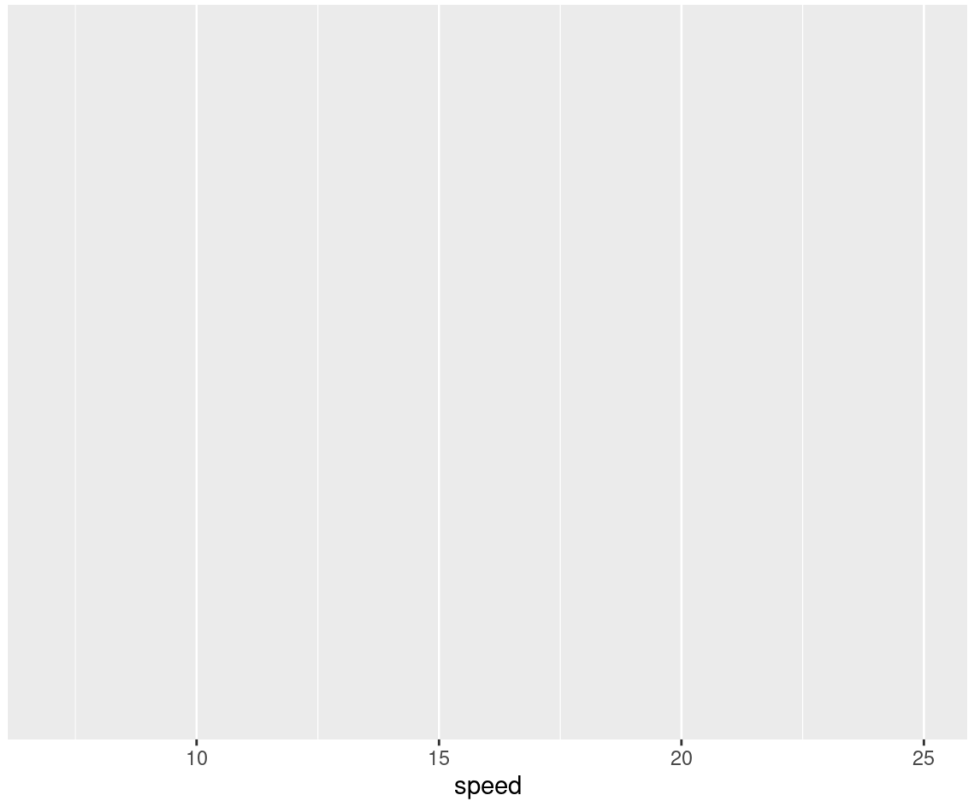
```
   speed dist
1      4    2
2      4   10
3      7    4
4      7   22
5      8   16
6      9   10
7     10   18
8     10   26
9     10   34
10    11   17
11    11   28
12    12   14
13    12   20
14    12   24
15    12   28
16    13   26
17    13   34
18    13   34
19    13   46
20    14   26
21    14   36
22    14   60
23    14   80
```

```r
cars %>%
  filter(speed > 4)
```

```
   speed dist
1      7    4
2      7   22
3      8   16
4      9   10
5     10   18
6     10   26
7     10   34
8     11   17
9     11   28
10    12   14
11    12   20
12    12   24
13    12   28
14    13   26
15    13   34
16    13   34
17    13   46
18    14   26
19    14   36
20    14   60
21    14   80
22    15   20
23    15   26
```

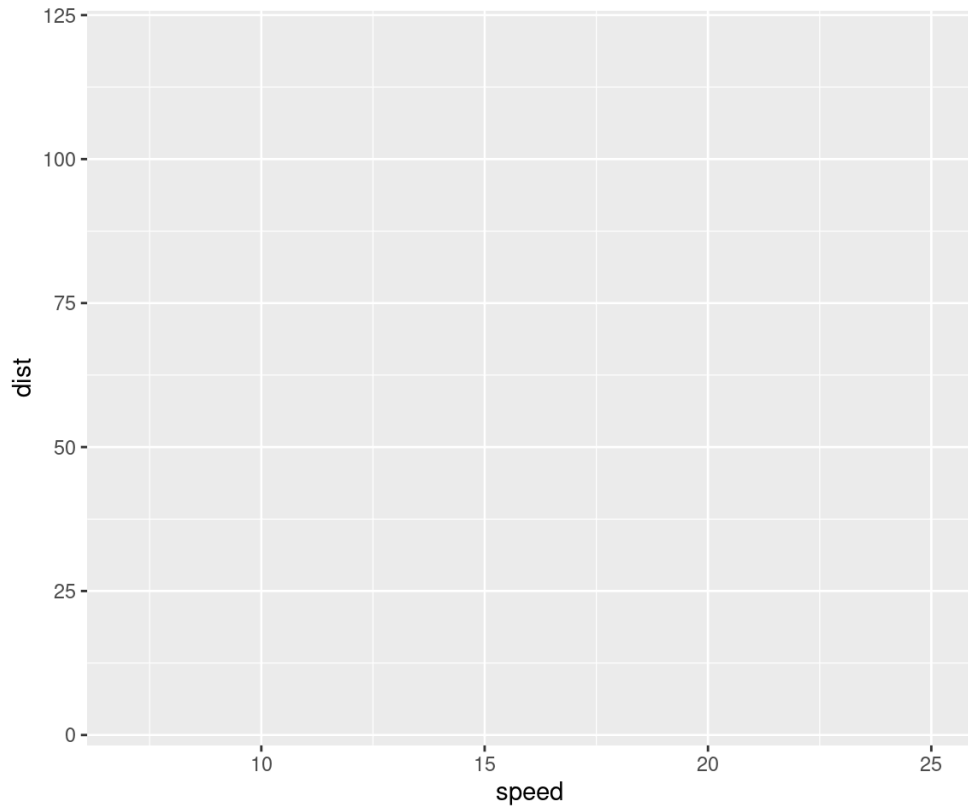
```r
cars %>%
  filter(speed > 4) %>%
  ggplot()
```

```r
cars %>%
  filter(speed > 4) %>%
  ggplot() +
  aes(x = speed)
```



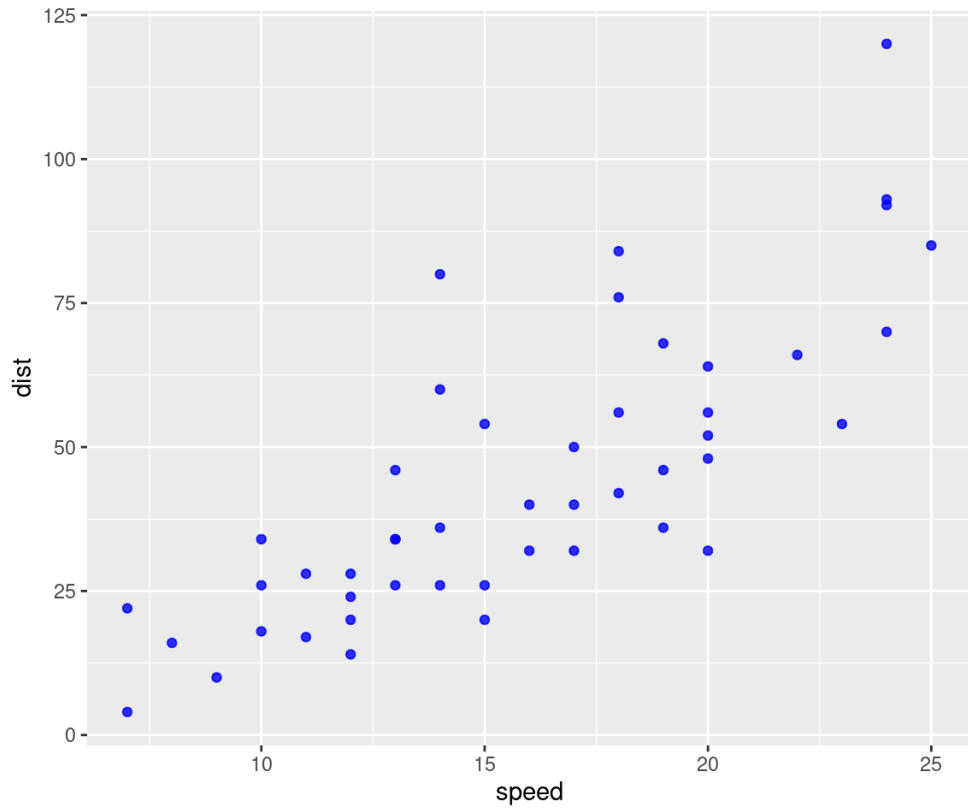
SDU

```r
cars %>%
  filter(speed > 4) %>%
  ggplot() +
  aes(x = speed) +
  aes(y = dist)
```



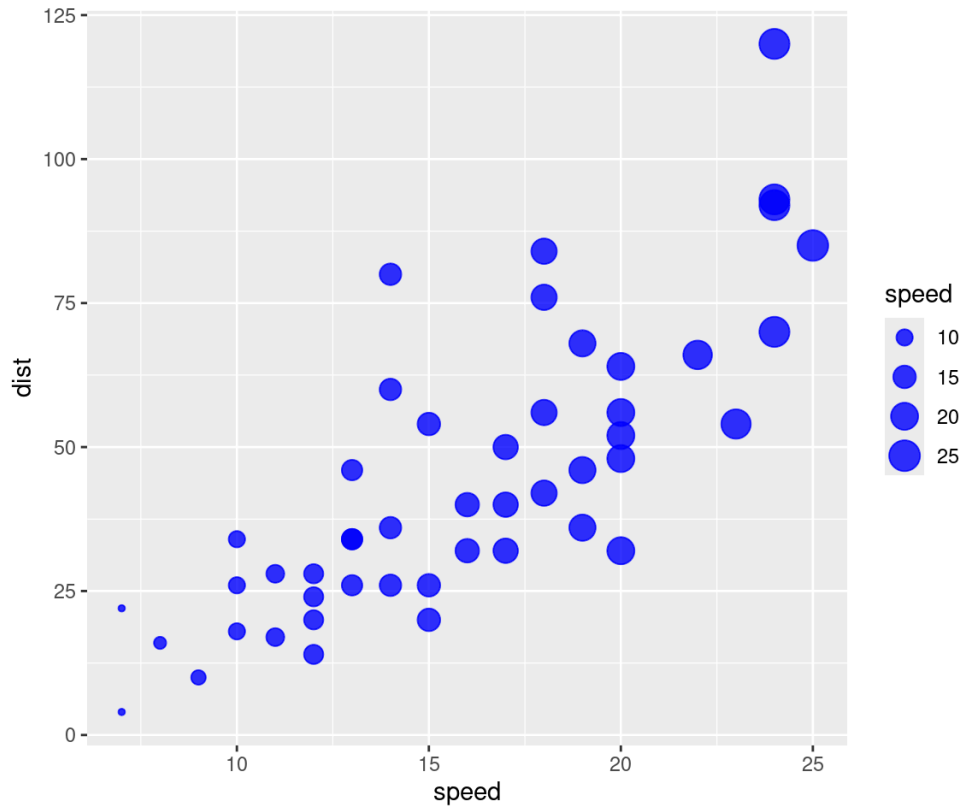
SDU

```r
cars %>%
  filter(speed > 4) %>%
  ggplot() +
  aes(x = speed) +
  aes(y = dist) +
  geom_point(
    alpha = .8,
    color = "blue"
    )
```



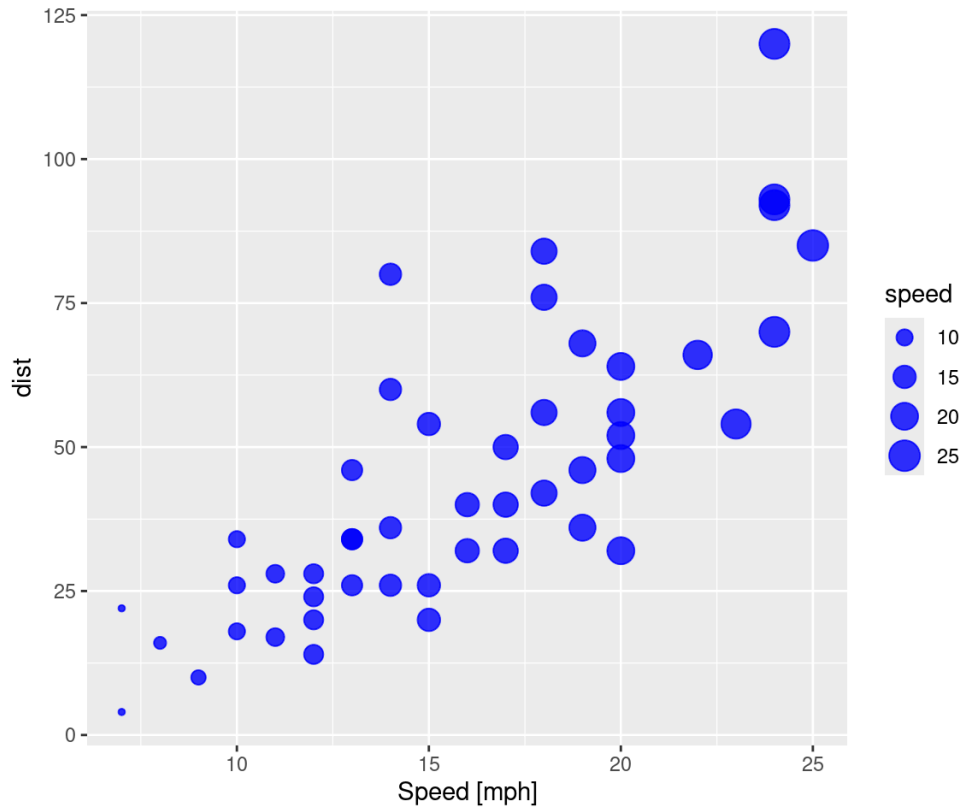
SDU

```r
cars %>%
  filter(speed > 4) %>%
  ggplot() +
  aes(x = speed) +
  aes(y = dist) +
  geom_point(
    alpha = .8,
    color = "blue"
    ) +
  aes(size = speed)
```



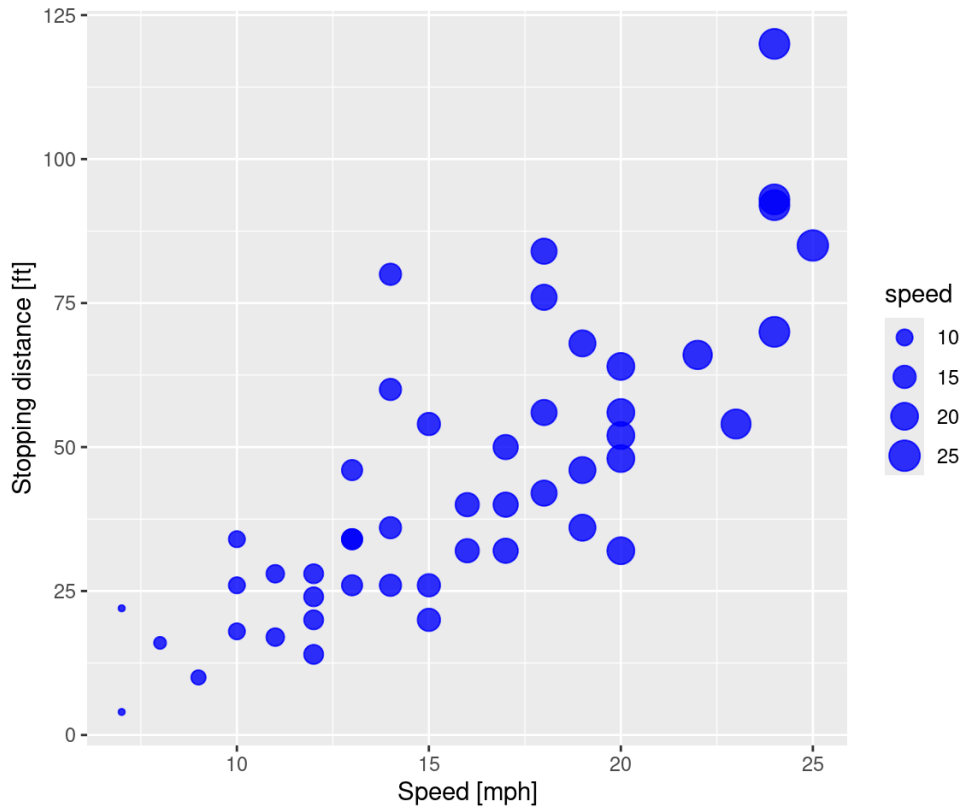
SDU

```r
cars %>%
  filter(speed > 4) %>%
  ggplot() +
  aes(x = speed) +
  aes(y = dist) +
  geom_point(
    alpha = .8,
    color = "blue"
    ) +
  aes(size = speed) +
  labs(x = "Speed [mph]")
```



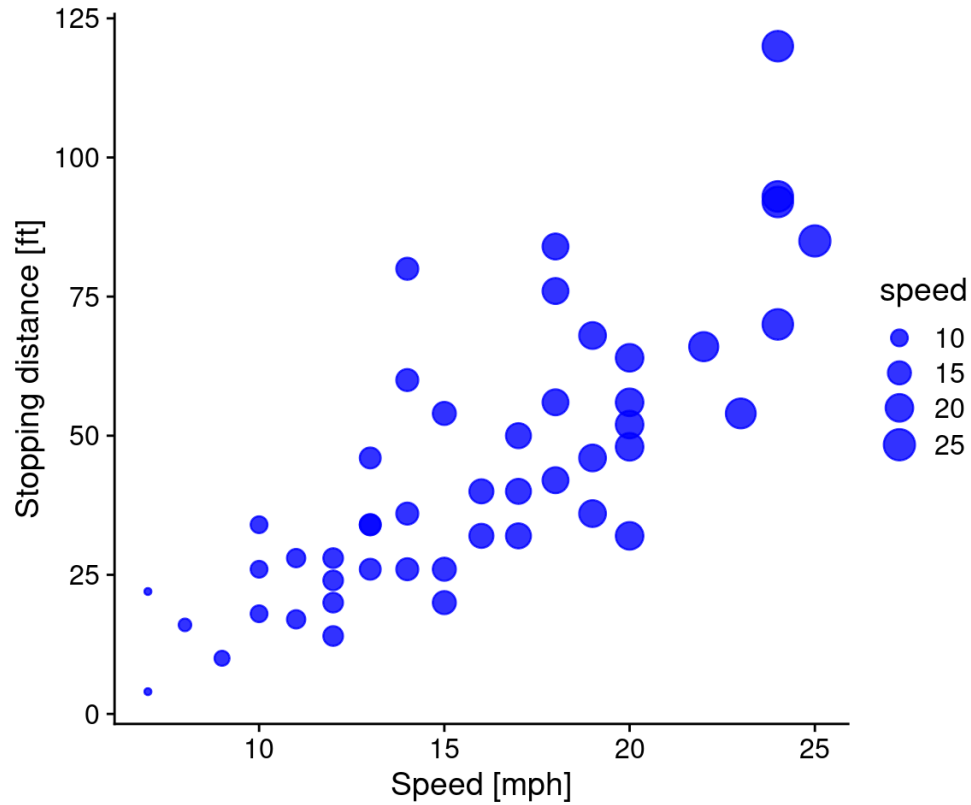
SDU

```r
cars %>%
  filter(speed > 4) %>%
  ggplot() +
  aes(x = speed) +
  aes(y = dist) +
  geom_point(
    alpha = .8,
    color = "blue"
    ) +
  aes(size = speed) +
  labs(x = "Speed [mph]") +
  labs(y = "Stopping distance [ft]")
```



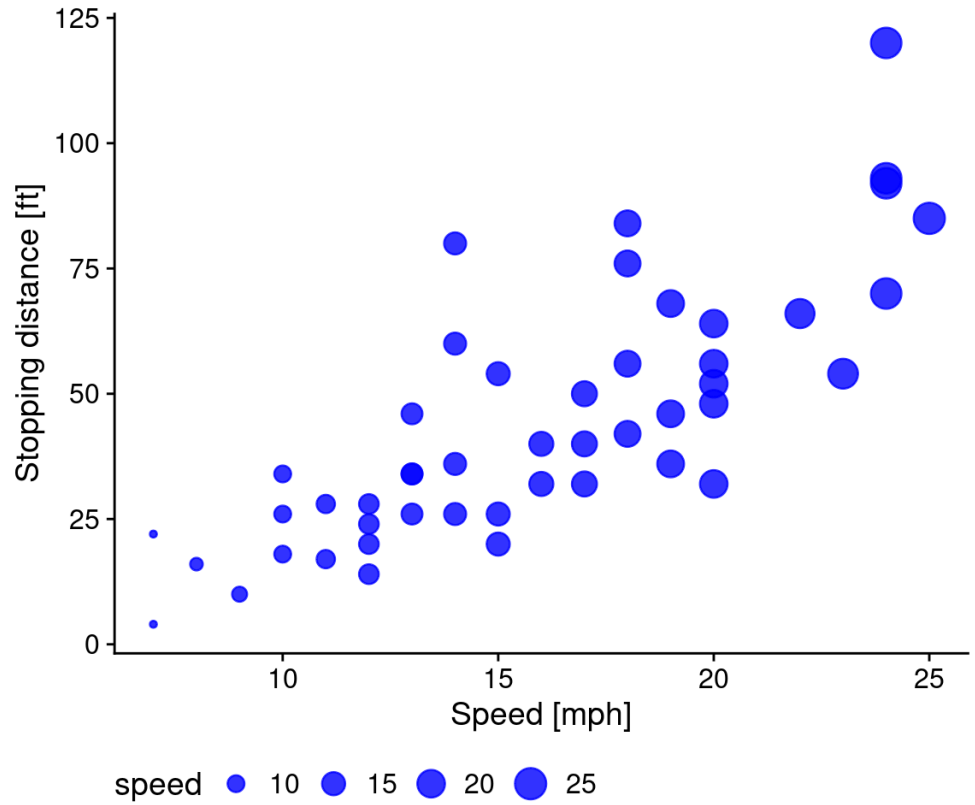
SDU

```r
cars %>%
  filter(speed > 4) %>%
  ggplot() +
  aes(x = speed) +
  aes(y = dist) +
  geom_point(
    alpha = .8,
    color = "blue"
    ) +
  aes(size = speed) +
  labs(x = "Speed [mph]") +
  labs(y = "Stopping distance [ft]") +
  theme_cowplot()
```



SDU

```r
cars %>%
  filter(speed > 4) %>%
  ggplot() +
  aes(x = speed) +
  aes(y = dist) +
  geom_point(
    alpha = .8,
    color = "blue"
    ) +
  aes(size = speed) +
  labs(x = "Speed [mph]") +
  labs(y = "Stopping distance [ft]") +
  theme_cowplot() +
  theme(legend.position="bottom")
```



SDU

Demonstration of data wrangling

```
1 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	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
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
Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
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

```r
mtcars %>%
  rownames_to_column()
```

	rowname	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
1	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
2	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
3	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
4	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
5	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2
6	Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
7	Duster 360	14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4
8	Merc 240D	24.4	4	146.7	62	3.69	3.190	20.00	1	0	4	2
9	Merc 230	22.8	4	140.8	95	3.92	3.150	22.90	1	0	4	2
10	Merc 280	19.2	6	167.6	123	3.92	3.440	18.30	1	0	4	4
11	Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
12	Merc 450SE	16.4	8	275.8	180	3.07	4.070	17.40	0	0	3	3
13	Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
14	Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
15	Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4
16	Lincoln Continental	10.4	8	460.0	215	3.00	5.424	17.82	0	0	3	4
17	Chrysler Imperial	14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4
18	Fiat 128	32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1
19	Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
20	Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
21	Toyota Corona	21.5	4	120.1	97	3.70	2.465	20.01	1	0	3	1
22	Dodge Challenger	15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2
23	AMC Javelin	15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2

```r
mtcars %>%
  rownames_to_column() %>%
  select(rowname, mpg, cyl, disp, hp)
```

```
               rowname  mpg cyl  disp  hp
1            Mazda RX4 21.0   6 160.0 110
2        Mazda RX4 Wag 21.0   6 160.0 110
3           Datsun 710 22.8   4 108.0  93
4       Hornet 4 Drive 21.4   6 258.0 110
5    Hornet Sportabout 18.7   8 360.0 175
6              Valiant 18.1   6 225.0 105
7           Duster 360 14.3   8 360.0 245
8            Merc 240D 24.4   4 146.7  62
9             Merc 230 22.8   4 140.8  95
10            Merc 280 19.2   6 167.6 123
11           Merc 280C 17.8   6 167.6 123
12          Merc 450SE 16.4   8 275.8 180
13          Merc 450SL 17.3   8 275.8 180
14         Merc 450SLC 15.2   8 275.8 180
15  Cadillac Fleetwood 10.4   8 472.0 205
16 Lincoln Continental 10.4   8 460.0 215
17   Chrysler Imperial 14.7   8 440.0 230
18            Fiat 128 32.4   4  78.7  66
19         Honda Civic 30.4   4  75.7  52
20      Toyota Corolla 33.9   4  71.1  65
21       Toyota Corona 21.5   4 120.1  97
22    Dodge Challenger 15.5   8 318.0 150
23         AMC Javelin 15.2   8 304.0 150
```

```r
mtcars %>%
  rownames_to_column() %>%
  select(rowname, mpg, cyl, disp, hp) %>%
  mutate(hp_per_cyl = hp / cyl)
```

	rowname	mpg	cyl	disp	hp	hp_per_cyl
1	Mazda RX4	21.0	6	160.0	110	18.33333
2	Mazda RX4 Wag	21.0	6	160.0	110	18.33333
3	Datsun 710	22.8	4	108.0	93	23.25000
4	Hornet 4 Drive	21.4	6	258.0	110	18.33333
5	Hornet Sportabout	18.7	8	360.0	175	21.87500
6	Valiant	18.1	6	225.0	105	17.50000
7	Duster 360	14.3	8	360.0	245	30.62500
8	Merc 240D	24.4	4	146.7	62	15.50000
9	Merc 230	22.8	4	140.8	95	23.75000
10	Merc 280	19.2	6	167.6	123	20.50000
11	Merc 280C	17.8	6	167.6	123	20.50000
12	Merc 450SE	16.4	8	275.8	180	22.50000
13	Merc 450SL	17.3	8	275.8	180	22.50000
14	Merc 450SLC	15.2	8	275.8	180	22.50000
15	Cadillac Fleetwood	10.4	8	472.0	205	25.62500
16	Lincoln Continental	10.4	8	460.0	215	26.87500
17	Chrysler Imperial	14.7	8	440.0	230	28.75000
18	Fiat 128	32.4	4	78.7	66	16.50000
19	Honda Civic	30.4	4	75.7	52	13.00000
20	Toyota Corolla	33.9	4	71.1	65	16.25000
21	Toyota Corona	21.5	4	120.1	97	24.25000
22	Dodge Challenger	15.5	8	318.0	150	18.75000
23	AMC Javelin	15.2	8	304.0	150	18.75000

```r
mtcars %>%
  rownames_to_column() %>%
  select(rowname, mpg, cyl, disp, hp) %>%
  mutate(hp_per_cyl = hp / cyl) %>%
  arrange(desc(hp_per_cyl))
```

	rowname	mpg	cyl	disp	hp	hp_per_cyl
1	Maserati Bora	15.0	8	301.0	335	41.87500
2	Ford Pantera L	15.8	8	351.0	264	33.00000
3	Duster 360	14.3	8	360.0	245	30.62500
4	Camaro Z28	13.3	8	350.0	245	30.62500
5	Ferrari Dino	19.7	6	145.0	175	29.16667
6	Chrysler Imperial	14.7	8	440.0	230	28.75000
7	Lotus Europa	30.4	4	95.1	113	28.25000
8	Volvo 142E	21.4	4	121.0	109	27.25000
9	Lincoln Continental	10.4	8	460.0	215	26.87500
10	Cadillac Fleetwood	10.4	8	472.0	205	25.62500
11	Toyota Corona	21.5	4	120.1	97	24.25000
12	Merc 230	22.8	4	140.8	95	23.75000
13	Datsun 710	22.8	4	108.0	93	23.25000
14	Porsche 914-2	26.0	4	120.3	91	22.75000
15	Merc 450SE	16.4	8	275.8	180	22.50000
16	Merc 450SL	17.3	8	275.8	180	22.50000
17	Merc 450SLC	15.2	8	275.8	180	22.50000
18	Hornet Sportabout	18.7	8	360.0	175	21.87500
19	Pontiac Firebird	19.2	8	400.0	175	21.87500
20	Merc 280	19.2	6	167.6	123	20.50000
21	Merc 280C	17.8	6	167.6	123	20.50000
22	Dodge Challenger	15.5	8	318.0	150	18.75000
23	AMC Javelin	15.2	8	304.0	150	18.75000

```r
mtcars %>%
  rownames_to_column() %>%
  select(rowname, mpg, cyl, disp, hp) %>%
  mutate(hp_per_cyl = hp / cyl) %>%
  arrange(desc(hp_per_cyl)) %>%
  filter(hp_per_cyl > 20)
```

	rowname	mpg	cyl	disp	hp	hp_per_cyl
1	Maserati Bora	15.0	8	301.0	335	41.87500
2	Ford Pantera L	15.8	8	351.0	264	33.00000
3	Duster 360	14.3	8	360.0	245	30.62500
4	Camaro Z28	13.3	8	350.0	245	30.62500
5	Ferrari Dino	19.7	6	145.0	175	29.16667
6	Chrysler Imperial	14.7	8	440.0	230	28.75000
7	Lotus Europa	30.4	4	95.1	113	28.25000
8	Volvo 142E	21.4	4	121.0	109	27.25000
9	Lincoln Continental	10.4	8	460.0	215	26.87500
10	Cadillac Fleetwood	10.4	8	472.0	205	25.62500
11	Toyota Corona	21.5	4	120.1	97	24.25000
12	Merc 230	22.8	4	140.8	95	23.75000
13	Datsun 710	22.8	4	108.0	93	23.25000
14	Porsche 914-2	26.0	4	120.3	91	22.75000
15	Merc 450SE	16.4	8	275.8	180	22.50000
16	Merc 450SL	17.3	8	275.8	180	22.50000
17	Merc 450SLC	15.2	8	275.8	180	22.50000
18	Hornet Sportabout	18.7	8	360.0	175	21.87500
19	Pontiac Firebird	19.2	8	400.0	175	21.87500
20	Merc 280	19.2	6	167.6	123	20.50000
21	Merc 280C	17.8	6	167.6	123	20.50000

```r
mtcars %>%
  rownames_to_column() %>%
  select(rowname, mpg, cyl, disp, hp) %>%
  mutate(hp_per_cyl = hp / cyl) %>%
  arrange(desc(hp_per_cyl)) %>%
  filter(hp_per_cyl > 20) %>%
  slice(1:5)
```

```
          rowname  mpg cyl disp  hp hp_per_cyl
1  Maserati Bora 15.0   8  301 335   41.87500
2 Ford Pantera L 15.8   8  351 264   33.00000
3     Duster 360 14.3   8  360 245   30.62500
4     Camaro Z28 13.3   8  350 245   30.62500
5   Ferrari Dino 19.7   6  145 175   29.16667
```

SDU

Load data from a file

```r
cases <- readr::read_delim('cases.txt',
                           delim='\t')

cases %>%
  head()
```

```
# A tibble: 6 × 4
  timestamp           hospitalized critical respirator
  <dttm>                     <dbl>    <dbl>      <dbl>
1 2020-03-17 08:00:00           82       18          0
2 2020-03-18 08:00:00          129       24          0
3 2020-03-19 08:00:00          153       30         27
4 2020-03-20 08:00:00          186       37         32
5 2020-03-21 08:00:00          206       42         33
6 2020-03-22 08:00:00          232       46         40
```

SDU

Tidy data

Data shapes

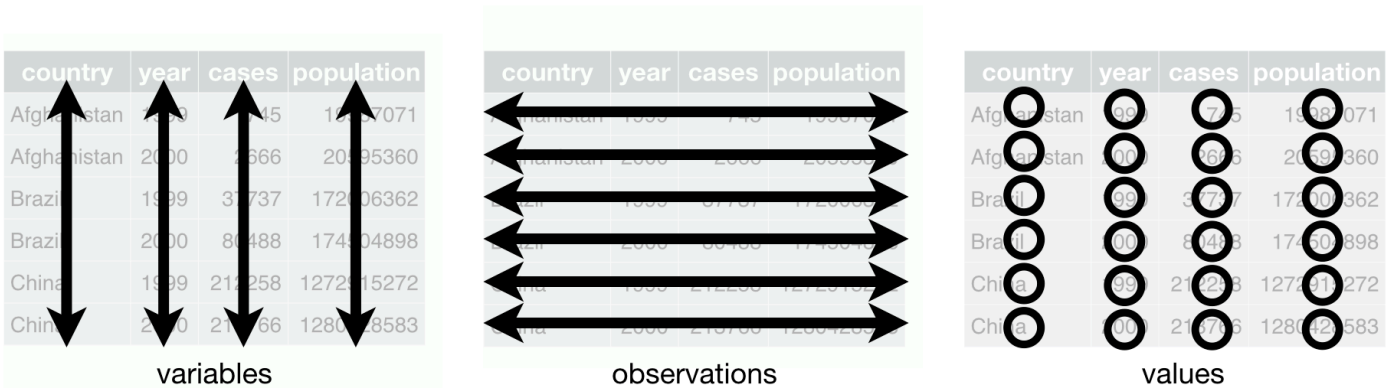
For `ggplot()` to work,
your data needs to be in a **tidy** format

This doesn't mean that it's clean—
it refers to the *structure* of the data

All the packages in the **tidyverse** work best with
tidy data; that why it's called that!

Tidy data

- Each variable has its own column
- Each observation has its own row
- Each value has its own cell



From chapter 12 of *R for Data Science*

SDU

Untidy data example

Real world data is often untidy, like this:

	A	B	C	D
1	Number of incidents			
2				
3	Office	2015	2016	2017
4	Utah County	134	145	**167**
5	Salt Lake County	**302**	334	331
6	Davis County	254	288	299
7	Juab County	78	**82**	87
8				
9	**bold = needs verification**			
10	yellow = compiled from different source			
11				

Tidy data example

Here's the tidy version of that same data:

	A	B	C	D	E
1	Office	Year	Incidents	Needs Verification	Different Source
2	Utah County	2015	134	FALSE	FALSE
3	Salt Lake County	2015	302	TRUE	FALSE
4	Davis County	2015	254	FALSE	FALSE
5	Juab County	2015	78	FALSE	FALSE
6	Utah County	2016	145	FALSE	TRUE
7	Salt Lake County	2016	334	FALSE	FALSE
8	Davis County	2016	288	FALSE	FALSE
9	Juab County	2016	82	TRUE	TRUE
10	Utah County	2017	167	TRUE	FALSE
11	Salt Lake County	2017	331	FALSE	FALSE
12	Davis County	2017	299	FALSE	TRUE
13	Juab County	2017	87	FALSE	FALSE

This is plottable!

Wide vs. long

Tidy data is also called “long” data

`wide`

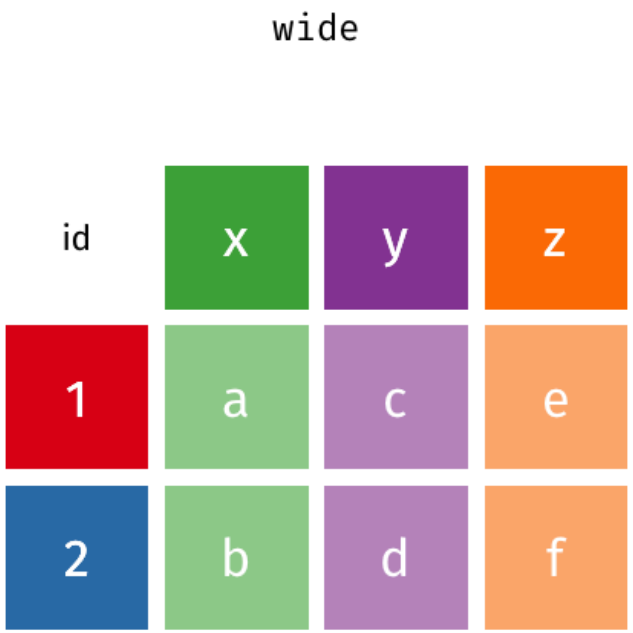
id	x	y	z
1	a	c	e
2	b	d	f

`long`

id	key	val
1	x	a
2	x	b
1	y	c
2	y	d
1	z	e
2	z	f

Moving from wide to long

Nowadays, `gather()` is called `pivot_longer()` and `spread()` is called `pivot_wider()`



SDU

Example of comparing two datasets

Vision based localization

Build a map from two images of the same scene.

Then estimate the position of the camera that acquired a third image of the same scene.

We would like to compare two different feature detector for this SIFT and ORB.

For a set of images we have the number of matches with the map and the number of inliers.

Load the data

```
readr::read_delim("data/codrone-results/experiment_01.csv",
                  delim="\t",
                  skip = 10,
                  col_names = c("file", "mapmatches", "inliers")) %>%
  mutate(featuredetector = "SIFT") -> data1
```

```
Rows: 96 Columns: 3
── Column specification ────────────────────────────────────────────────────────
Delimiter: "\t"
chr (1): file
dbl (2): mapmatches, inliers

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
data3 <- readr::read_delim("data/codrone-results/experiment_03.csv",
                           delim="\t",
                           skip = 10,
                           col_names = c("file", "mapmatches", "inliers")) %>%
  mutate(featuredetector = "ORB") -> data3
```

```
Rows: 96 Columns: 3
── Column specification ────────────────────────────────────────────────────────
Delimiter: "\t"
chr (1): file
dbl (2): mapmatches, inliers

ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

SDU

Provide an overview of the loaded data

```
full_data <- rbind(data1, data3)
summary(full_data)
```

```
    file            mapmatches       inliers       featuredetector
Length:192         Min.   : 45.0   Min.   :  0.00   Length:192
Class :character   1st Qu.: 56.0   1st Qu.:  0.00   Class :character
Mode  :character   Median :114.0   Median :  0.00   Mode  :character
                   Mean   :101.8   Mean   : 14.76
                   3rd Qu.:142.0   3rd Qu.: 16.25
                   Max.   :202.0   Max.   :202.00
```

Number of feature matches with map

```
1 full_data
```

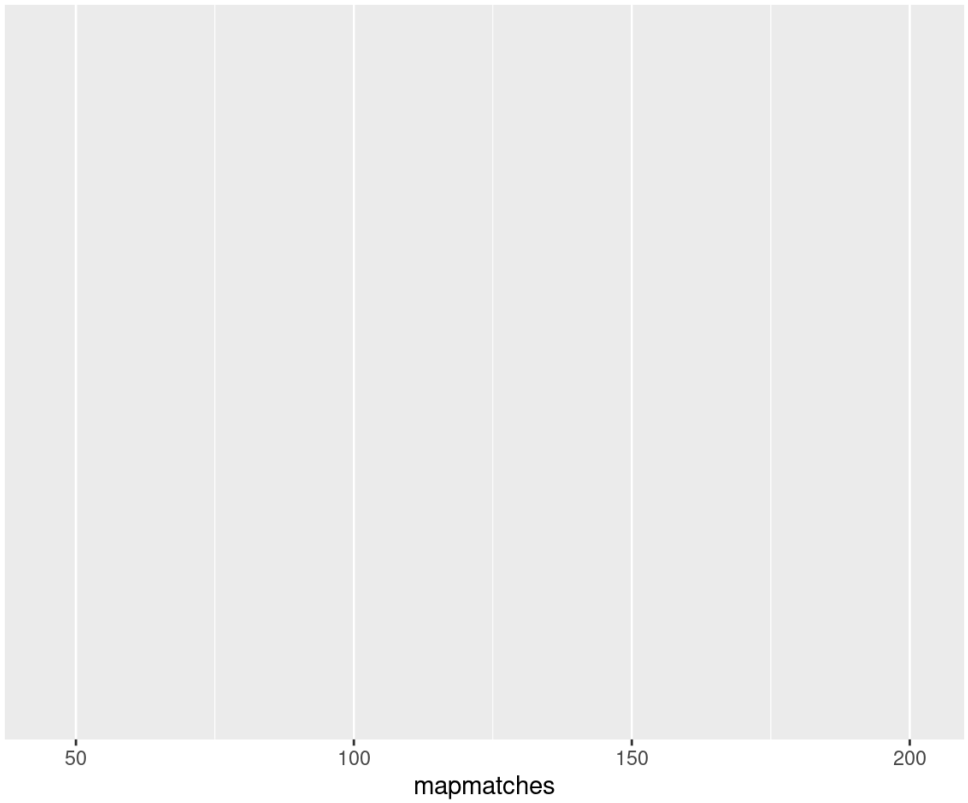
```
# A tibble: 192 × 4
   file                                        mapmatches inliers
featuredetector
   <chr>                                            <dbl>   <dbl>
<chr>
 1 input_campus_flight_reduced/frame-002300.…         116       0
SIFT
 2 input_campus_flight_reduced/frame-002400.…         131       0
SIFT
 3 input_campus_flight_reduced/frame-002500.…         131       0
SIFT
 4 input_campus_flight_reduced/frame-002600.…         132       0
SIFT
 5 input_campus_flight_reduced/frame-002700.…         145       0
SIFT
 6 input_campus_flight_reduced/frame-002800.…         140       0
SIFT
 7 input_campus_flight_reduced/frame-002900.…         144       0
SIFT
 8 input_campus_flight_reduced/frame-003000.…         147       0
SIFT
 9 input_campus_flight_reduced/frame-003100.…         148       0
SIFT
10 input_campus_flight_reduced/frame-003200.…         145       0
```

Number of feature matches with map

```
full_data %>%
  ggplot()
```

Number of feature matches with map

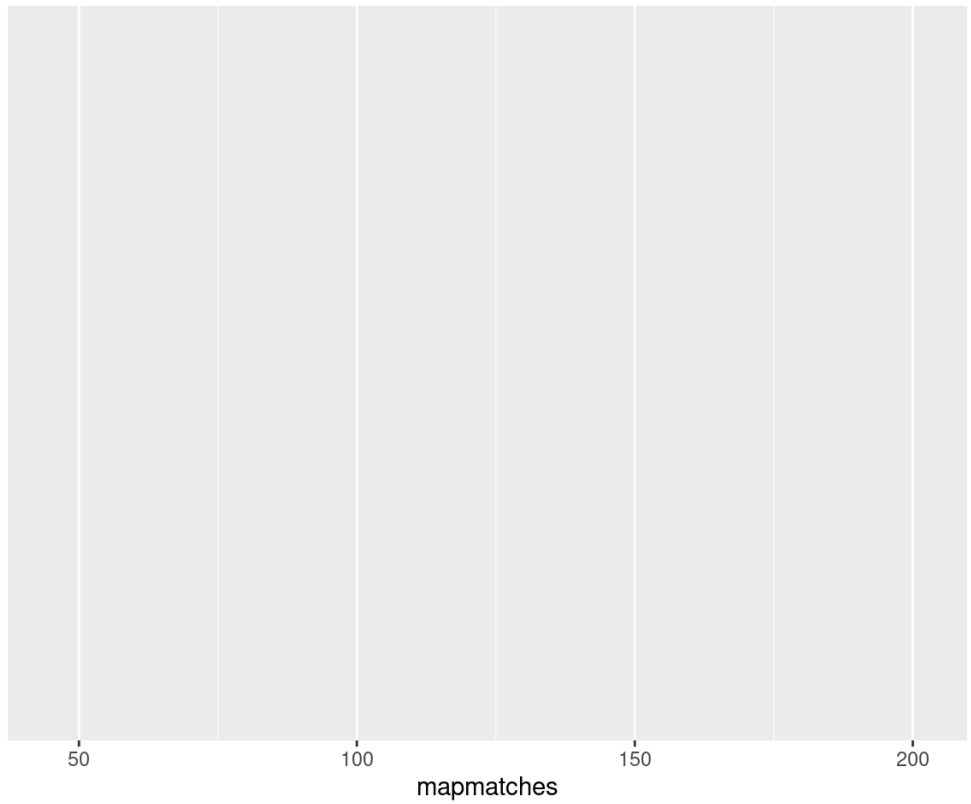
```
full_data %>%
  ggplot() +
  aes(x = mapmatches)
```



SDU

Number of feature matches with map

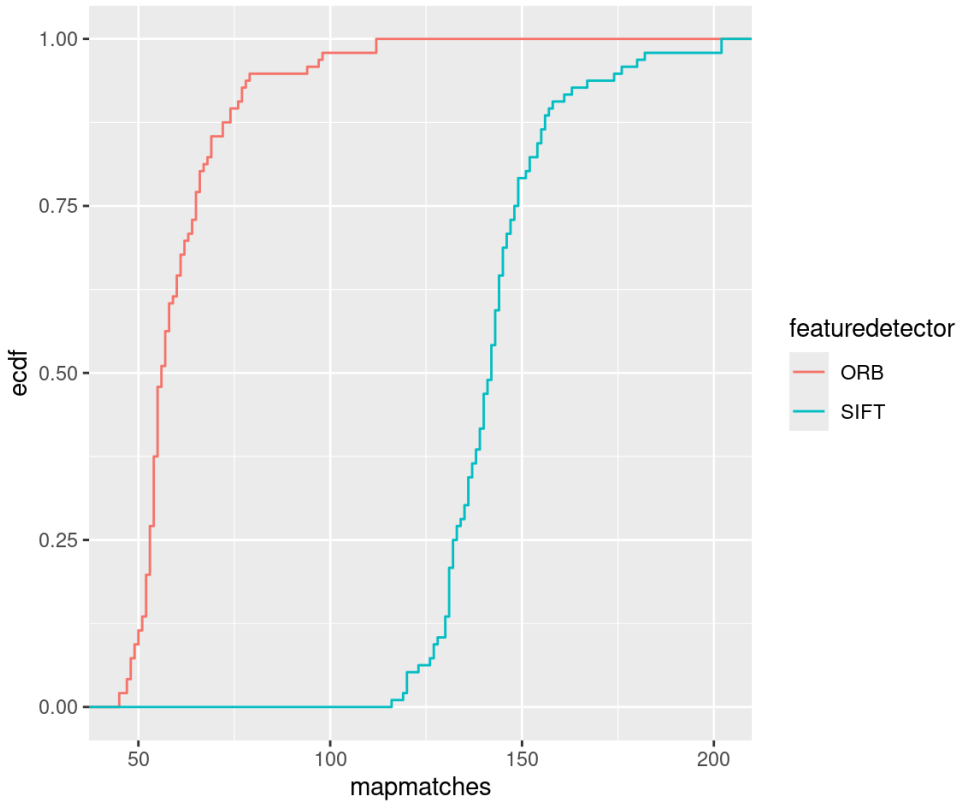
```
full_data %>%
  ggplot() +
  aes(x = mapmatches) +
  aes(color = featuredetector)
```



SDU

Number of feature matches with map

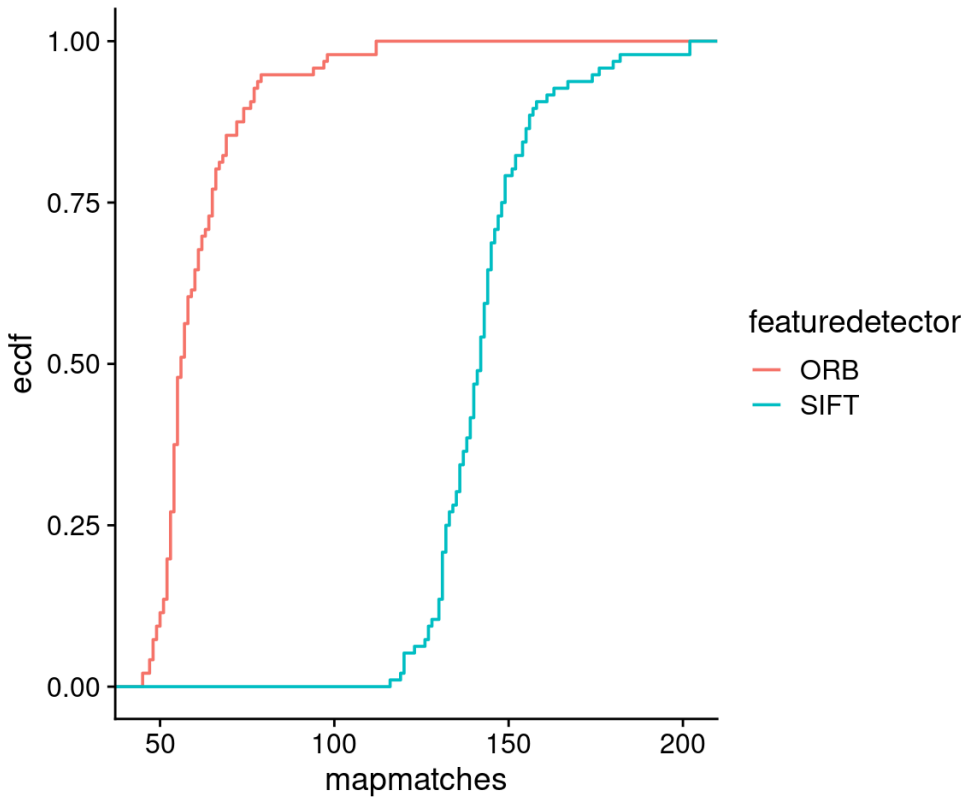
```
full_data %>%
  ggplot() +
  aes(x = mapmatches) +
  aes(color = featuredetector) +
  stat_ecdf()
```



SDU

Number of feature matches with map

```
full_data %>%
  ggplot() +
  aes(x = mapmatches) +
  aes(color = featuredetector) +
  stat_ecdf() +
  theme_cowplot()
```



SDU

Number of inliers with the map

```
1 full_data
```

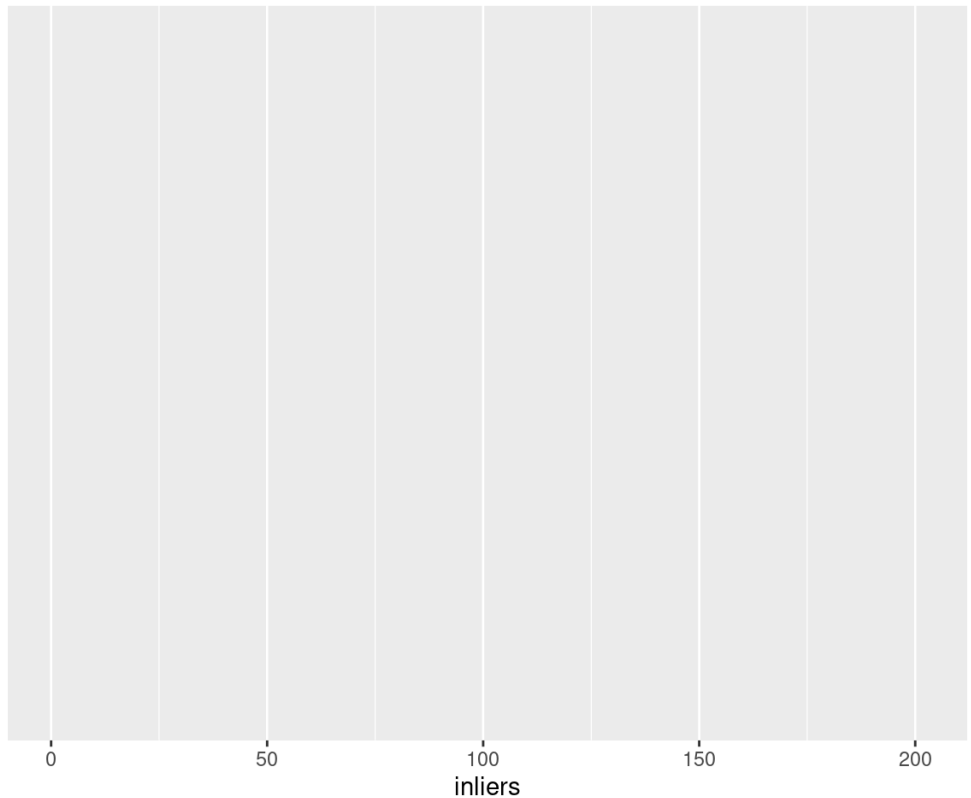
```
# A tibble: 192 × 4
   file                                          mapmatches inliers
featuredetector
   <chr>                                              <dbl>   <dbl>
<chr>
 1 input_campus_flight_reduced/frame-002300.…         116       0
SIFT
 2 input_campus_flight_reduced/frame-002400.…         131       0
SIFT
 3 input_campus_flight_reduced/frame-002500.…         131       0
SIFT
 4 input_campus_flight_reduced/frame-002600.…         132       0
SIFT
 5 input_campus_flight_reduced/frame-002700.…         145       0
SIFT
 6 input_campus_flight_reduced/frame-002800.…         140       0
SIFT
 7 input_campus_flight_reduced/frame-002900.…         144       0
SIFT
 8 input_campus_flight_reduced/frame-003000.…         147       0
SIFT
 9 input_campus_flight_reduced/frame-003100.…         148       0
SIFT
10 input_campus_flight_reduced/frame-003200.…         145       0
```

Number of inliers with the map

```
full_data %>%
  ggplot()
```

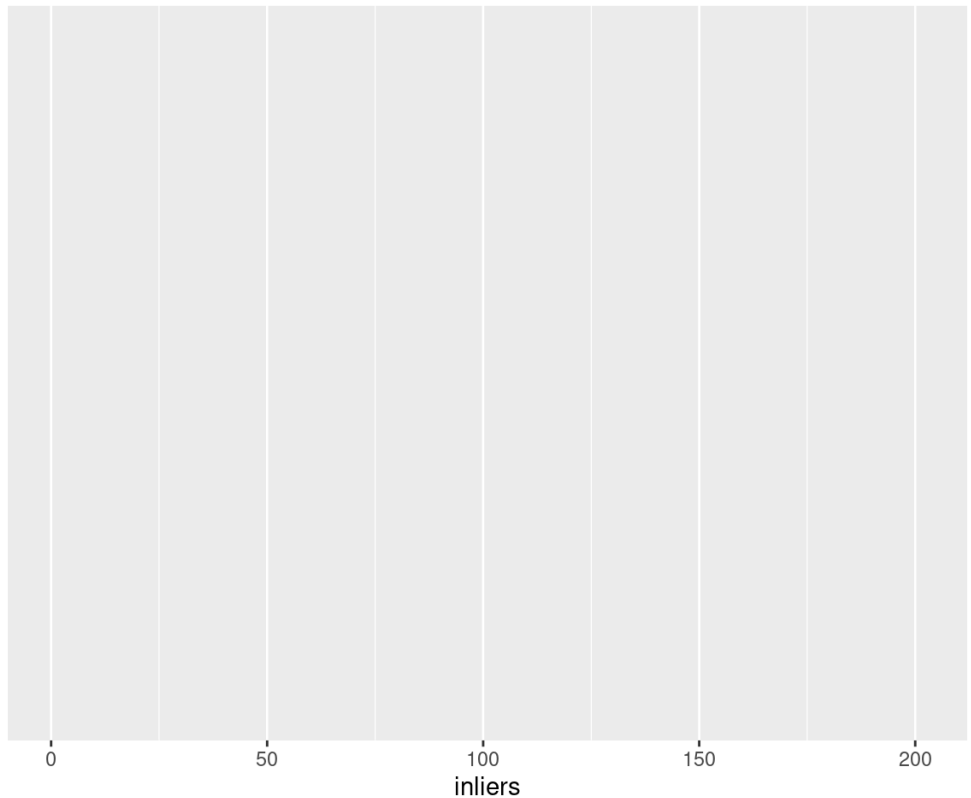
Number of inliers with the map

```
full_data %>%
  ggplot() +
  aes(x = inliers)
```



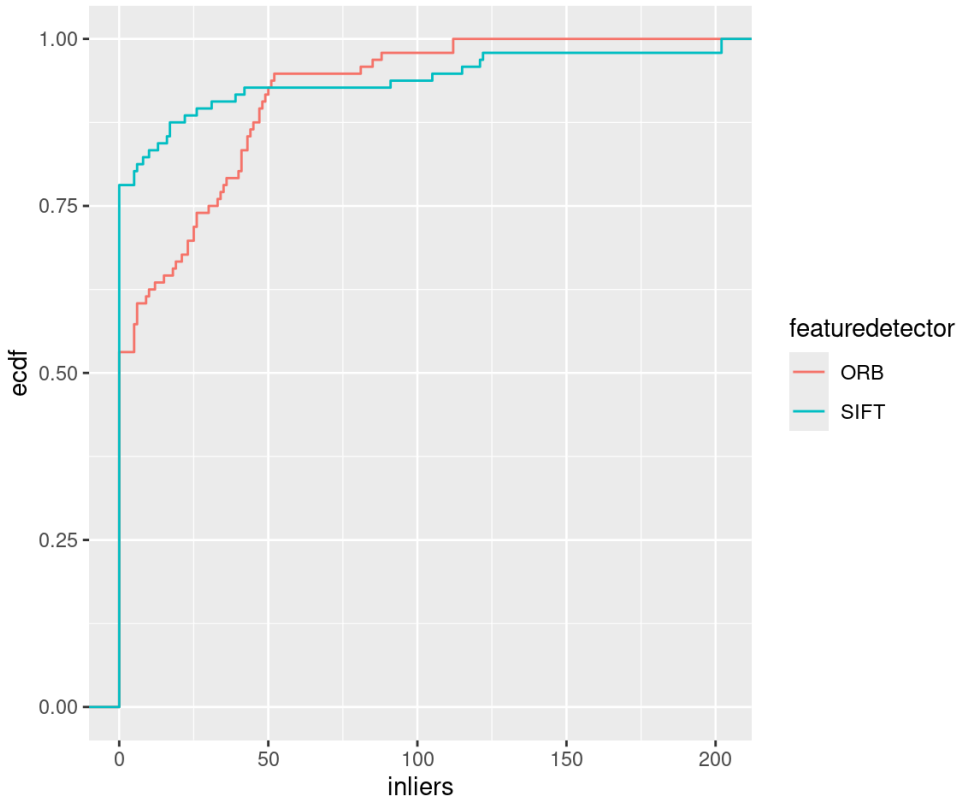
Number of inliers with the map

```
full_data %>%
  ggplot() +
  aes(x = inliers) +
  aes(color = featuredetector)
```



Number of inliers with the map

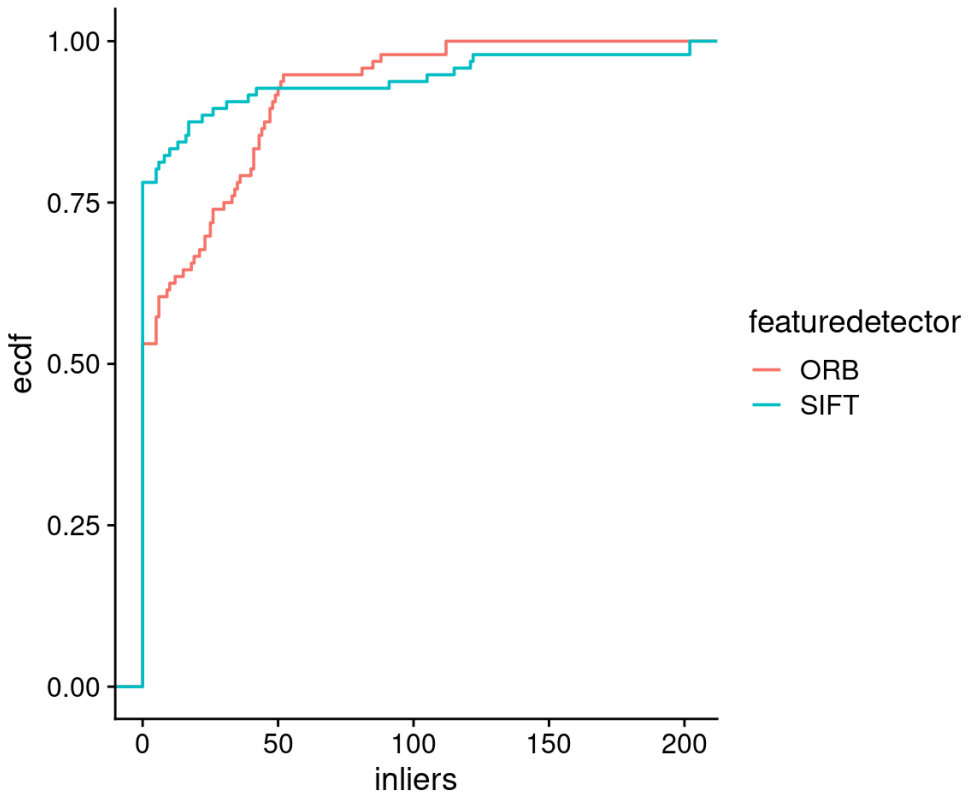
```
full_data %>%
  ggplot() +
  aes(x = inliers) +
  aes(color = featuredetector) +
  stat_ecdf()
```



SDU

Number of inliers with the map

```
full_data %>%
  ggplot() +
  aes(x = inliers) +
  aes(color = featuredetector) +
  stat_ecdf() +
  theme_cowplot()
```



SDU

Assignment

Your goal is to use a reproducible workflow, which covers the following

1. Loading data (Own data or Tidy Tuesday)
2. Visualizing data
3. Commenting on data
4. Modelling data (statistical test or fit a model)
5. Commenting on results
6. Generate a pdf with the results

Due date: Thursday in two weeks (2025-11-13).