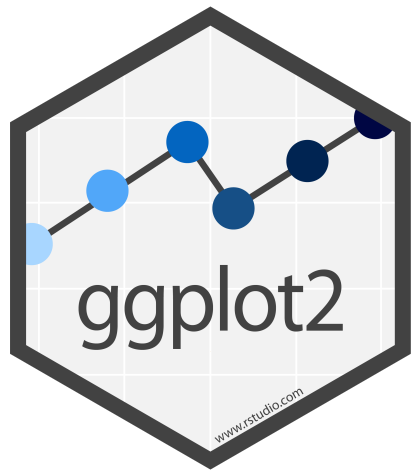
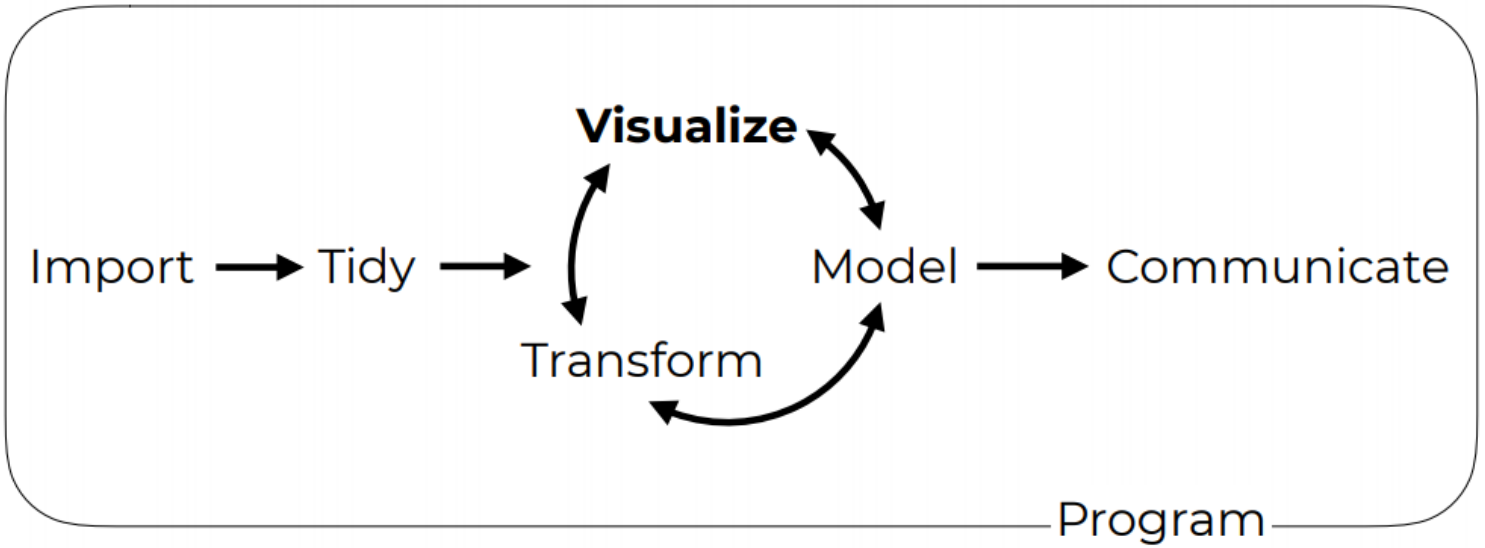
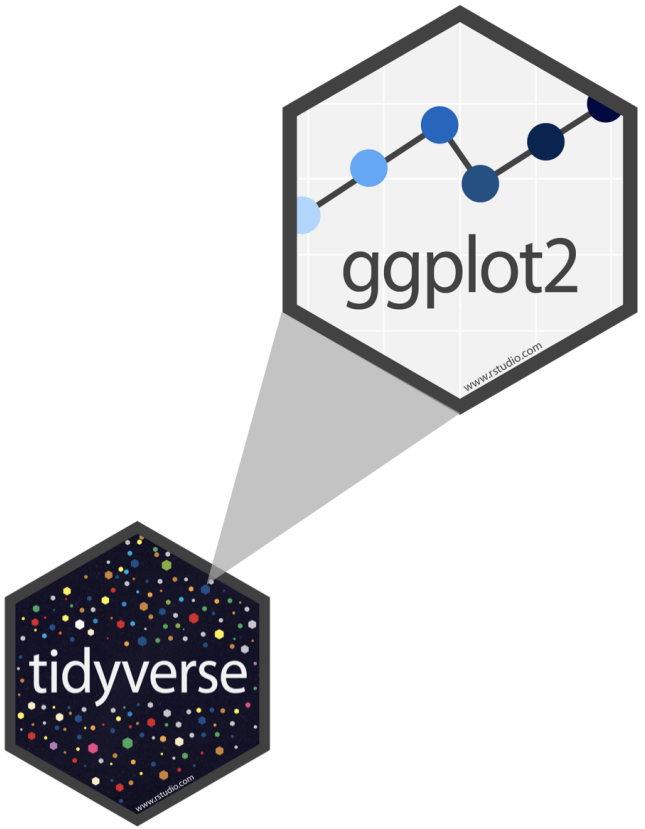
Data Visualization with




Visualize
Import
Tidy
Model
Communicate
Transform
Program

Grammar of Graphics

ggplot2 is a data visualization tool that follow grammar of graphics and using different functions.

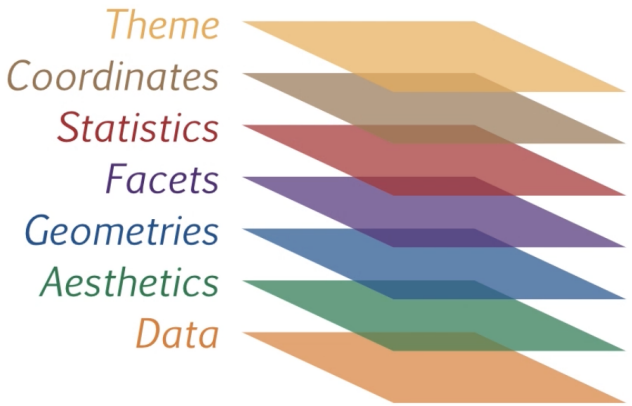


"The simple graph has brought more information to the data analyst's mind than any other device." - John Tuckey

ggplot2

- **data** maps to
- aesthetics in
- layers

ggplot2 layers



Setup

02-visualization.Rmd file

look around the file

```
library(ggplot2)
library(dplyr)
library(readr)
bechdel <- read_csv("./data/bechdel.csv")
```

y...	imdb	title	test	clean_test	bina...	budget
<int>	<chr>	<chr>	<chr>	<ord>	<chr>	<int>
2013	tt1711425	21 & Over	notalk	notalk	FAIL	13000000
2012	tt1343727	Dredd 3D	ok-disagree	ok	PASS	45000000
2013	tt2024544	12 Years a Slave	notalk-disagree	notalk	FAIL	20000000
2013	tt1272878	2 Guns	notalk	notalk	FAIL	61000000

4 rows | 1-7 of 15 columns

About the data

The raw data behind the story "The Dollar-And-Cents Case Against Hollywood's Exclusion of Women".

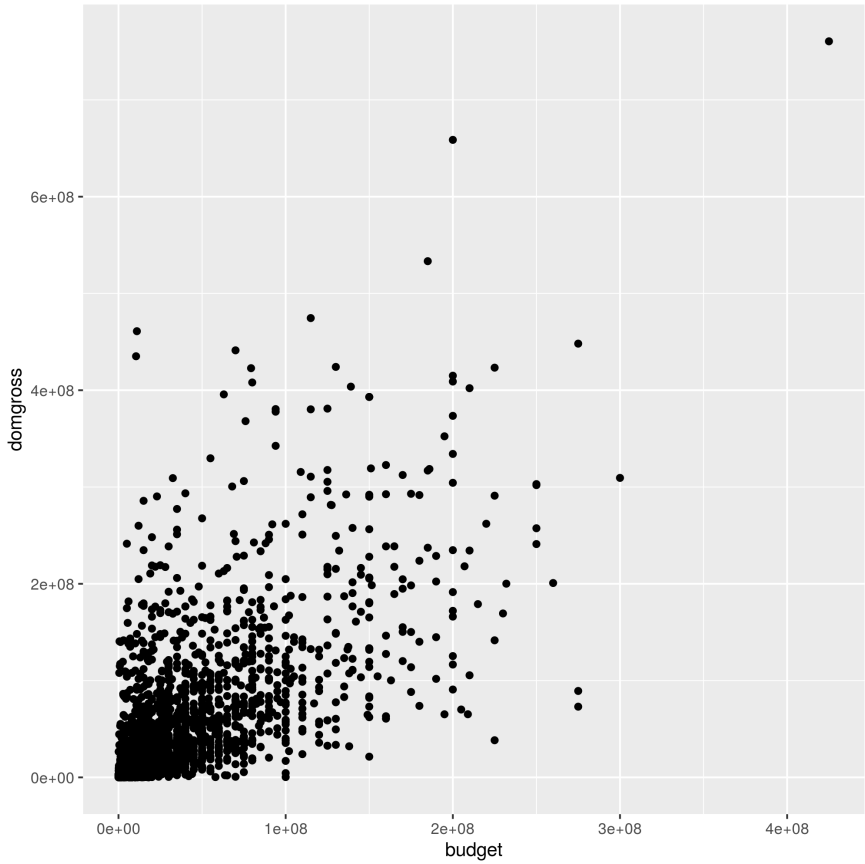
- year :Year of release
- imdb :Text to construct IMDB url. Ex: https://www.imdb.com/title/tt1711425
- title :Movie title
- test :bechdel test result (detailed, with discrepancies indicated)
- clean_test : bechdel test result (detailed): **ok** = passes test, dubious, **men** = women only talk about men, **notalk** = women don't talk to each other, **nowomen** = fewer than two women
- binary :Bechdel Test PASS vs FAIL binary
- budget :Film budget
- **domgross** :Domestic (US) gross
- **intgross** :Total International (i.e., worldwide) gross
- **code** :Bechdel Code
- **budget_2013** :Budget in 2013 inflation adjusted dollars
- **domgross_2013** :Domestic gross (US) in 2013 inflation adjusted dollars
- **intgross_2013** :Total International (i.e., worldwide) gross in 2013 inflation adjusted dollars
- **period_code**

Your Turn 01

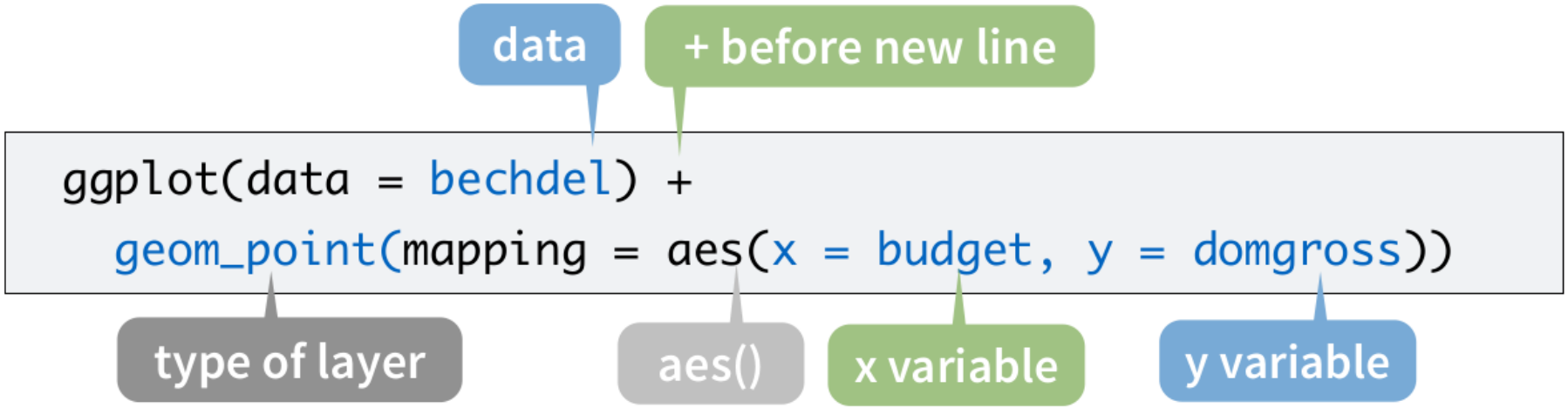
01:00

Run this code to make a graph

```
ggplot(data = bechdel) +
 geom_point(mapping = aes(x = budget,
                          y =
domgross))
```



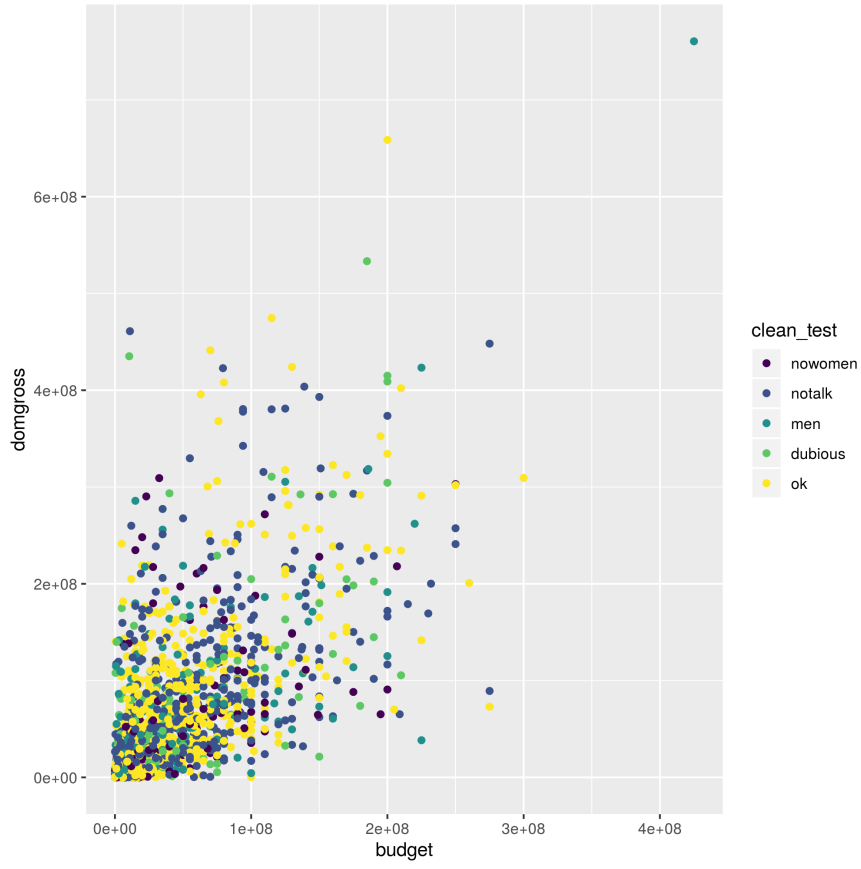
ggplot2 Template



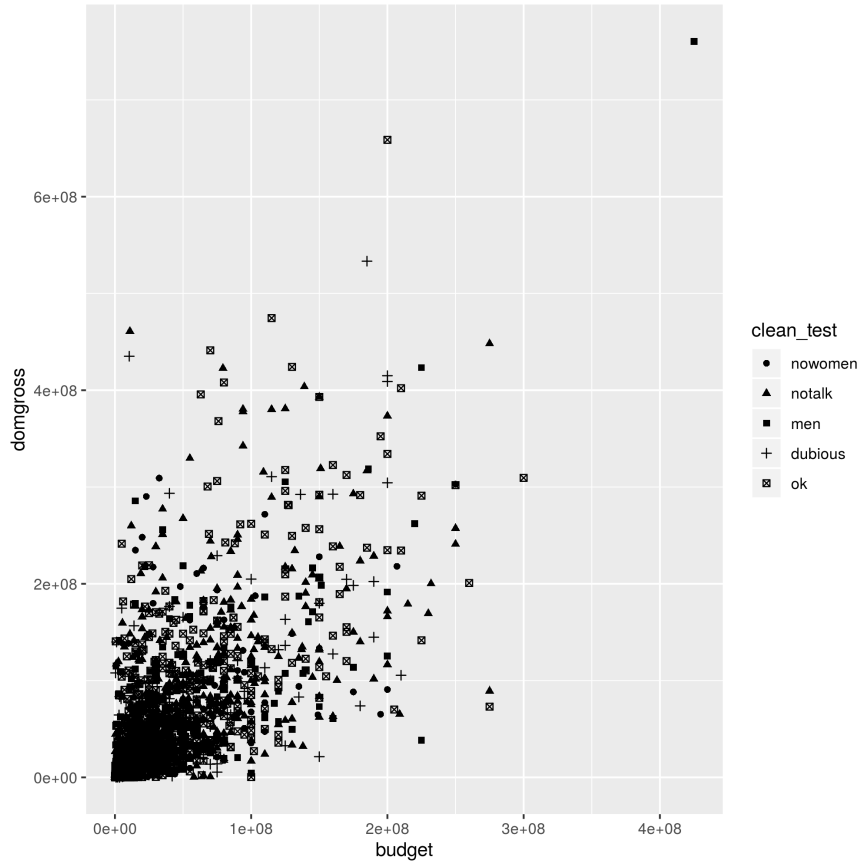
Aesthetics

- something we can see in graph
- X and Y axis, **color**, **size**, ***shape***

```
ggplot(data = bechdel) +
 geom_point(mapping = aes(x = budget,
                          y = domgross,
                          color =
clean_test))
```



```
ggplot(bechdel) +
  geom_point(aes(x = budget,
                 y = domgross,
                 shape = clean_test))
```



Your turn 2

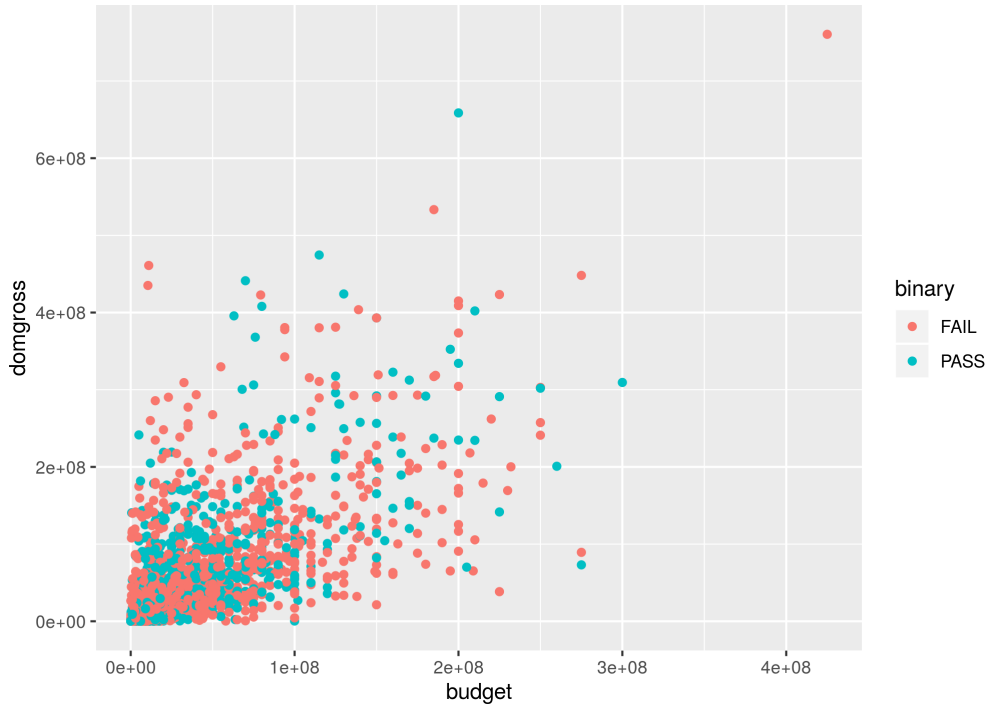
05:00

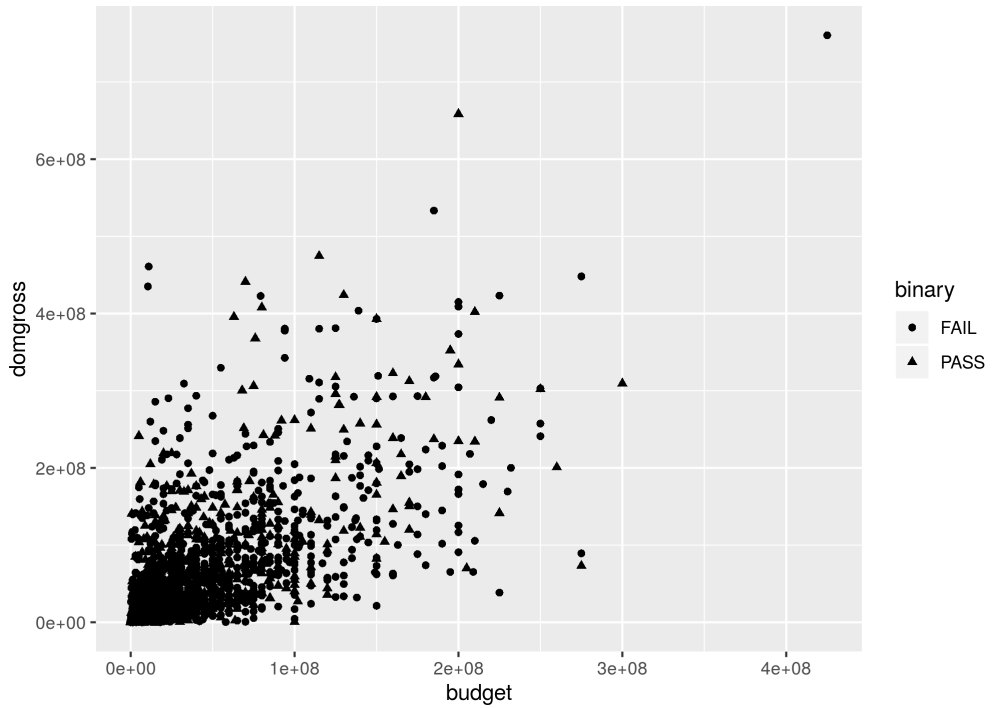
- Experiment adding **color**, **size** , **alpha** , and shape aesthetics to your graph
- How do aesthetics behave different when mapped to discrete and continous variable?
- What happens when you use more than one aesthetic?

```
# color aesthetic
ggplot(data = bechdel) +
  geom_point(mapping = aes(x = fill me up , y = fill me up, color = ....))
```

```
# shape aesthetic
ggplot(data = bechdel) +
  geom_point(mapping = aes(x = fill me up , y = fill me up, shape = ....))
```

Your Answer 2

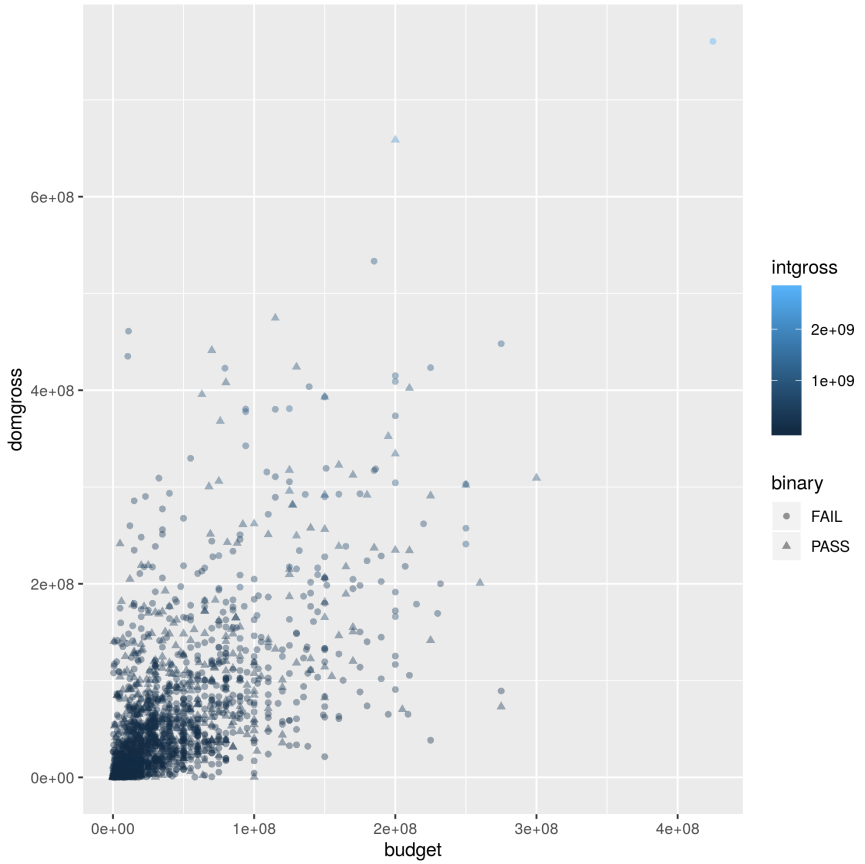




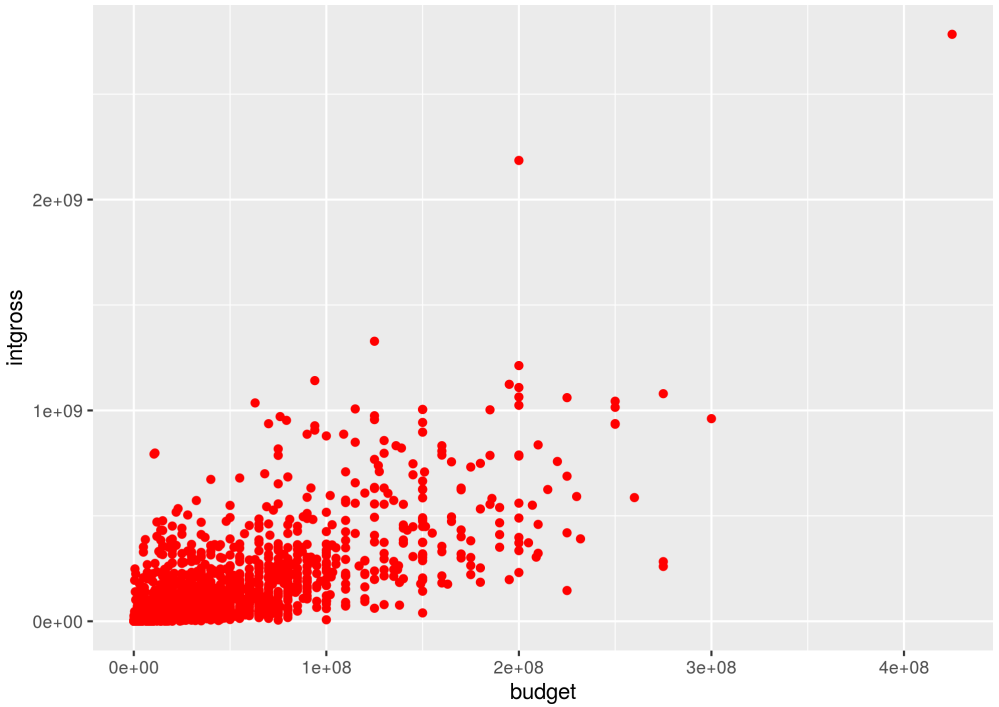

```
ggplot(bechdel) +
  geom_point(mapping = aes(
    x = budget,
    y = domgross,
    shape = binary
  ))
```

What happens when you use more than one aesthetic?

```
ggplot(bechdel) +
  geom_point(
    mapping =
      aes(
        x = budget,
        y = domgross,
        shape = binary,
        color = intgross
      ),
    alpha = 0.4
  )
```



```
ggplot(data = bechdel) +
  geom_point(mapping = aes(x = budget,
                           y = intgross),
             color ="red")
```



Geoms

- functions that define the geometric or visual object of the graphs.
- used to define different type of plots like bar charts, boxplots, line charts and many more.

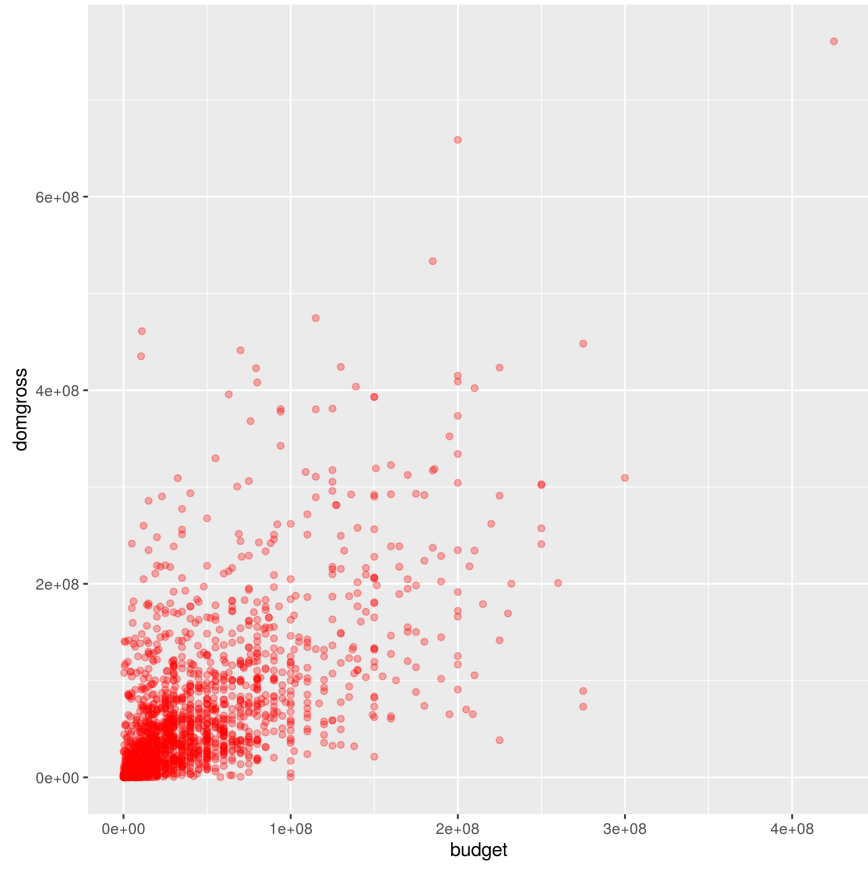
```
ggplot(data = <DATA>) +
 <GEOM_FUNCTION>(mapping =
                      aes(<MAPPINGS>))
```

Geom	Description
geom_point()	creates scatterplot
geom_boxplot()	creates boxplot
geom_histogram()	creates histogram
geom_density()	creates density plot
geom_bar()	creates barplot
geom_line()	creates lineplot

ScatterPlot

- Scatterplot is used to define the relationship between two variable
- **geom_point()** geom is used for scatterplot.

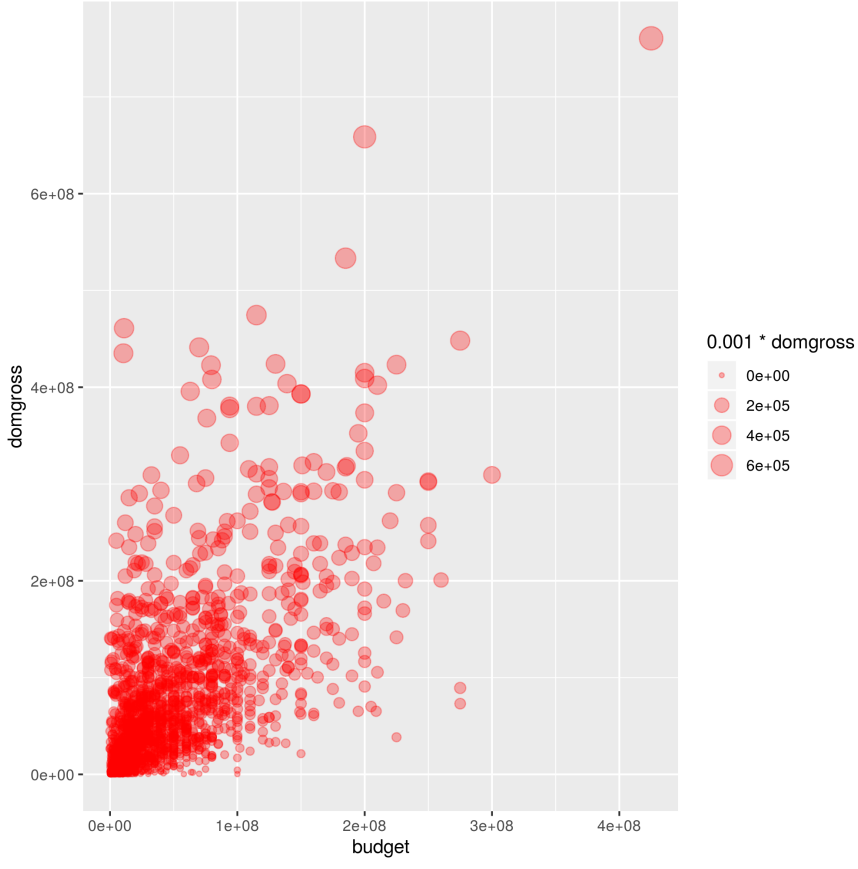
```
ggplot(data = bechdel) +
  geom_point(aes(
    x = budget,
    y = domgross),
    alpha = 0.3,color="red")
```



Bubble Chart

- bubblechart is same like scatterplot with points being bubbles
- bubblechart is defined using **geom_point()** geoms

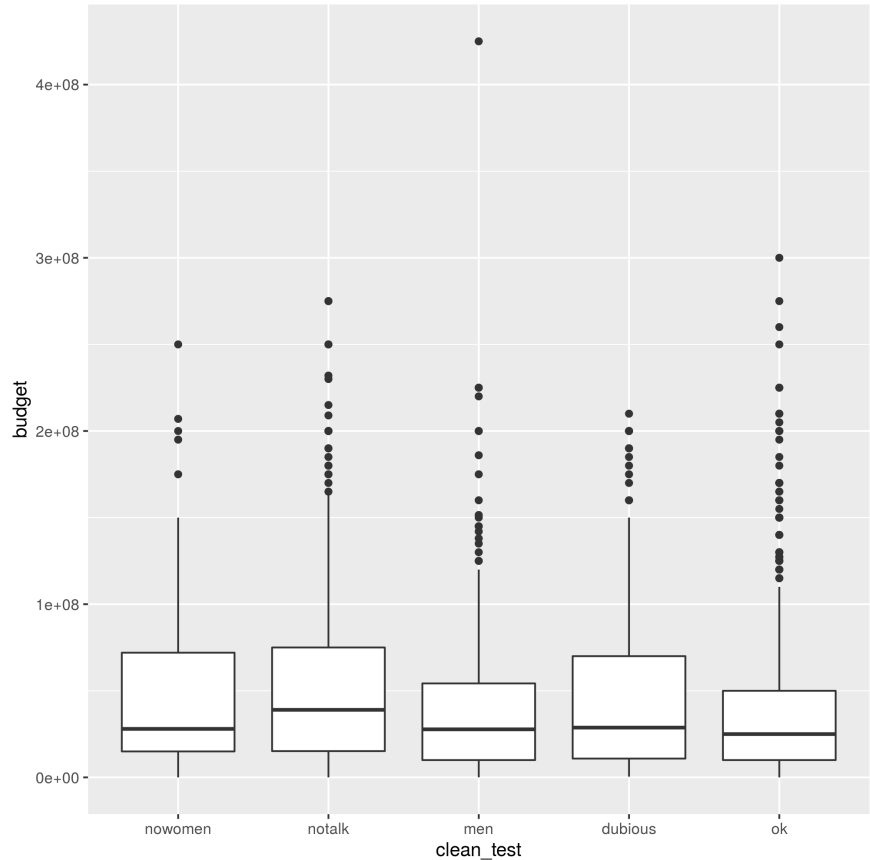
```
ggplot(data = bechdel) +
  geom_point(aes(
    x = budget,
    y = domgross,
    size = .001*domgross
  ),
    alpha = 0.3,color="red")
```



BoxPlot

- boxplot can be defined using **geom_boxplot()** geoms
- used to define **quartiles** and find outliers.

```
ggplot(data = bechdel) +
  geom_boxplot(aes(x = clean_test,
                   y =  budget))
```



Your Turn 3

02:00

- scatterplot of the domgross and intgross
- boxplot of the domgross VS binary

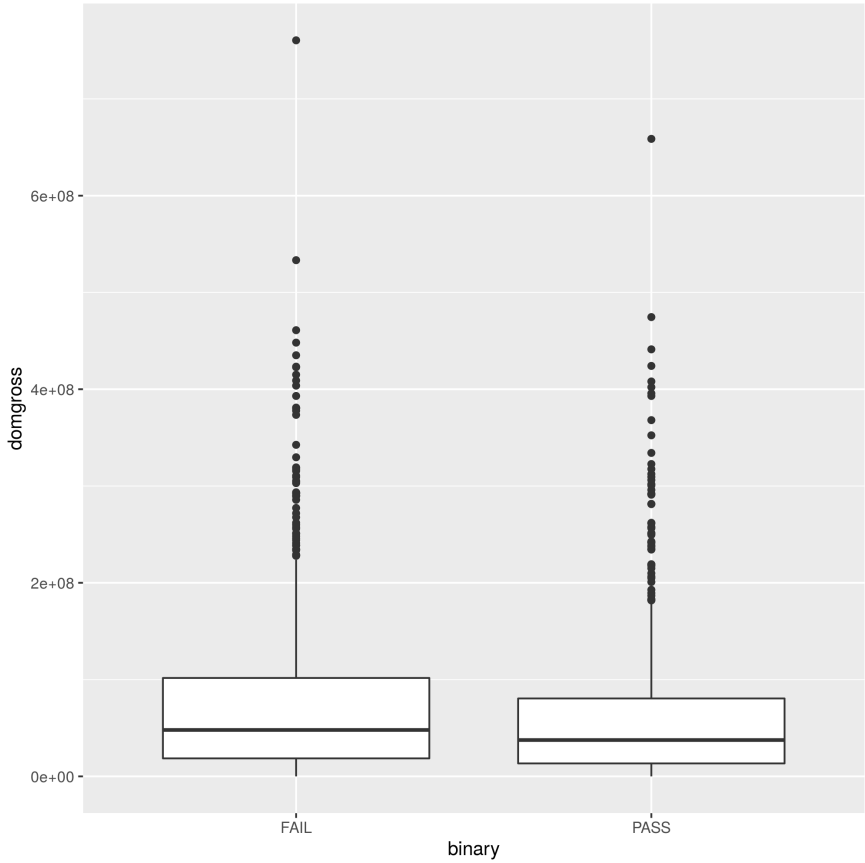
```
ggplot(data = bechdel) +
  geom_point(aes(x = domgross,
                 y = <FILL IT  UP>))
```

```
ggplot(data = bechdel) +
  geom_boxplot(aes(x = binary,
                   y = <FILL IT  UP>))
```

Your Answer 3

- boxplot of the domgross VS binary

```
ggplot(data = bechdel) +
  geom_boxplot(aes(x = binary,
                   y = domgross))
```



Histogram

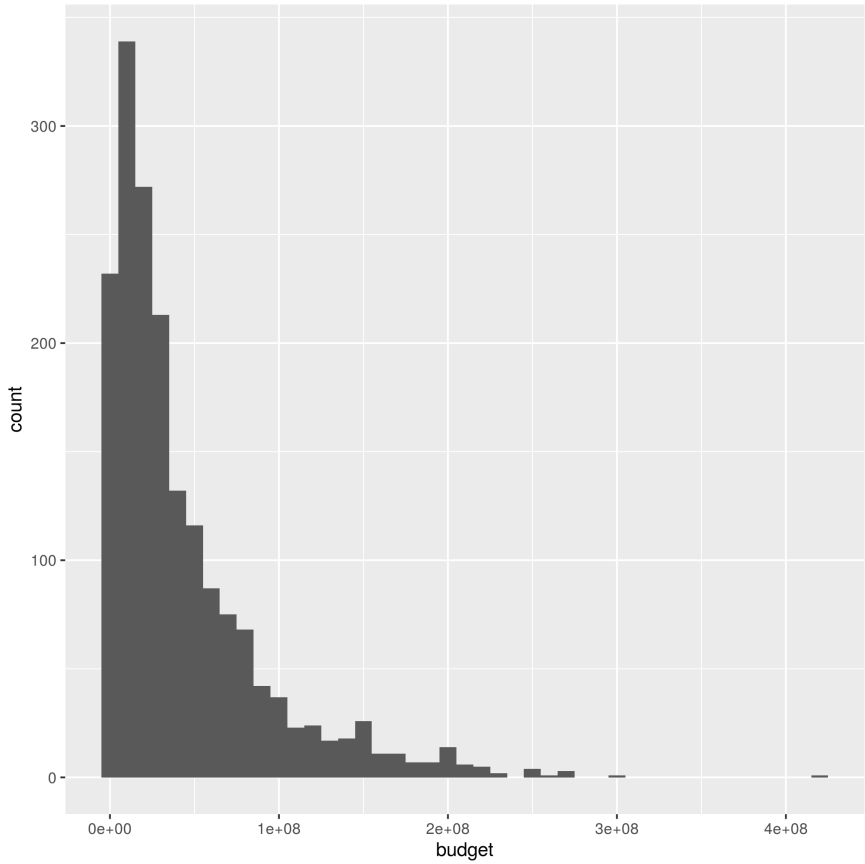
- shows the distribution of data
- **geom_histogram()** geom is used.

```
ggplot(data = bechdel) +
  geom_histogram(aes(x = budget ))
```

Histogram

- binwidth gives width to bins.

```
ggplot(data = bechdel) +
  geom_histogram(aes(x = budget),
    binwidth = 10000000
  )
```



Your Turn 4

02:00

- Make the Histogarm of **intgross** shown below.

Your Answer 4

```
ggplot(data = bechdel) +
  geom_histogram(mapping = aes(x = intgross))
```

Density Plot

- shows the distribution of data
- **geom_density()** geom is used for density plot

```
ggplot(data = bechdel) +
  geom_density(
    mapping =
      aes(x = budget)
  )
```

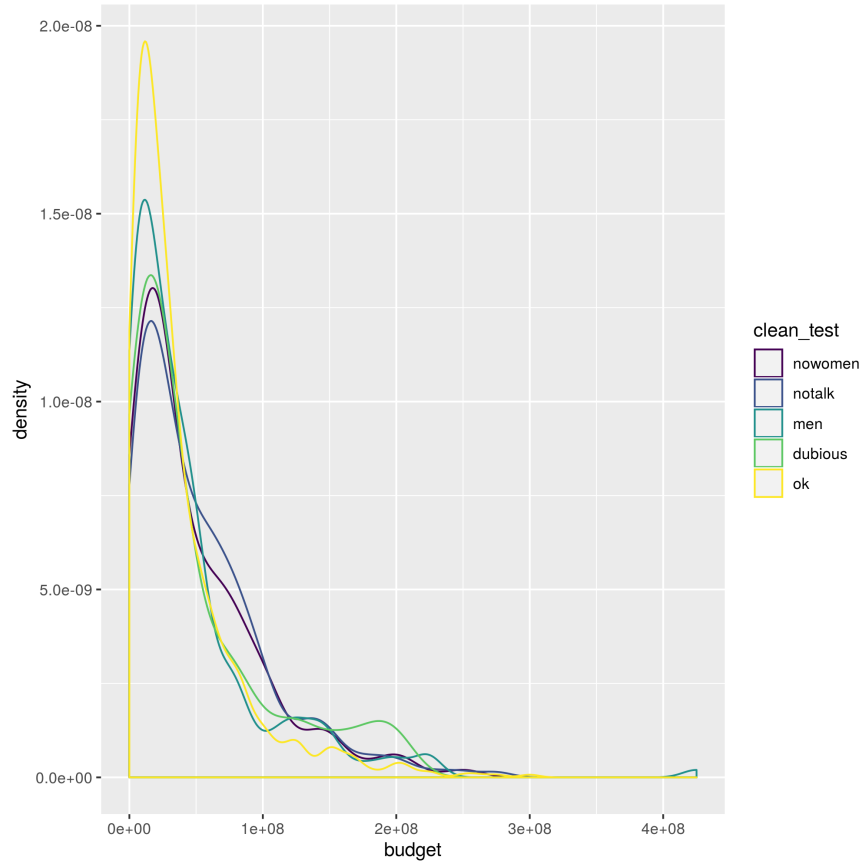
Your Turn 5

02:00

- Make the Density of **budget** colored by **clean_test** as shown below

Your Answer 5

```
ggplot(data = bechdel) +
  geom_density(aes(
    x = budget,
    color = clean_test
  ))
```



Bar Plot

- visualize the count of the values
- **geom_bar()** geom is used for barplot.

```
ggplot(data = bechdel) +
  geom_bar(aes(x = clean_test))
```

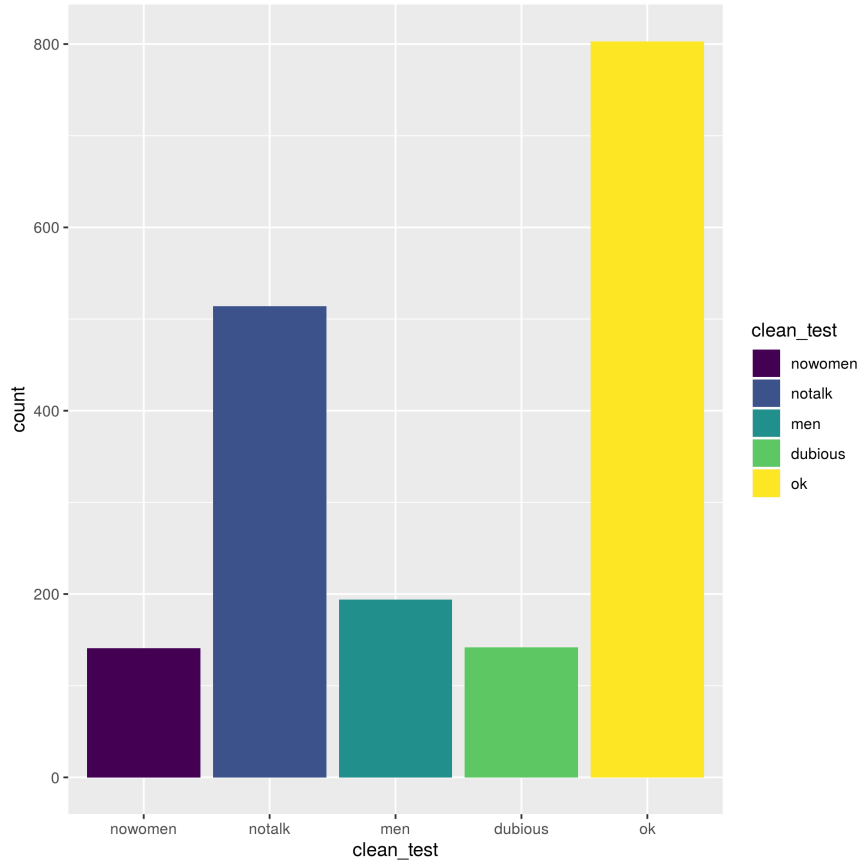
Your Turn 6

02:00

- Make the bar chart of clean_test and colored by clean_test as shown below.

Your Answer 6

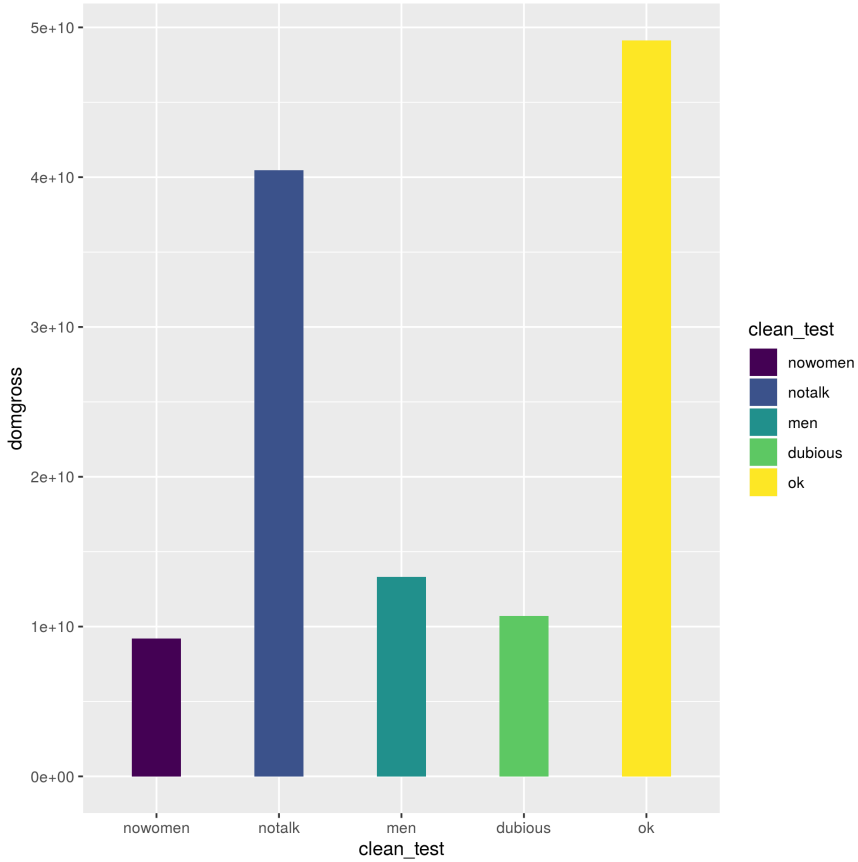
```
ggplot(data = bechdel) +
  geom_bar(aes(
    x = clean_test,
    fill = clean_test
  ))
```



Coordinate scales

- Arrange the width of the bar

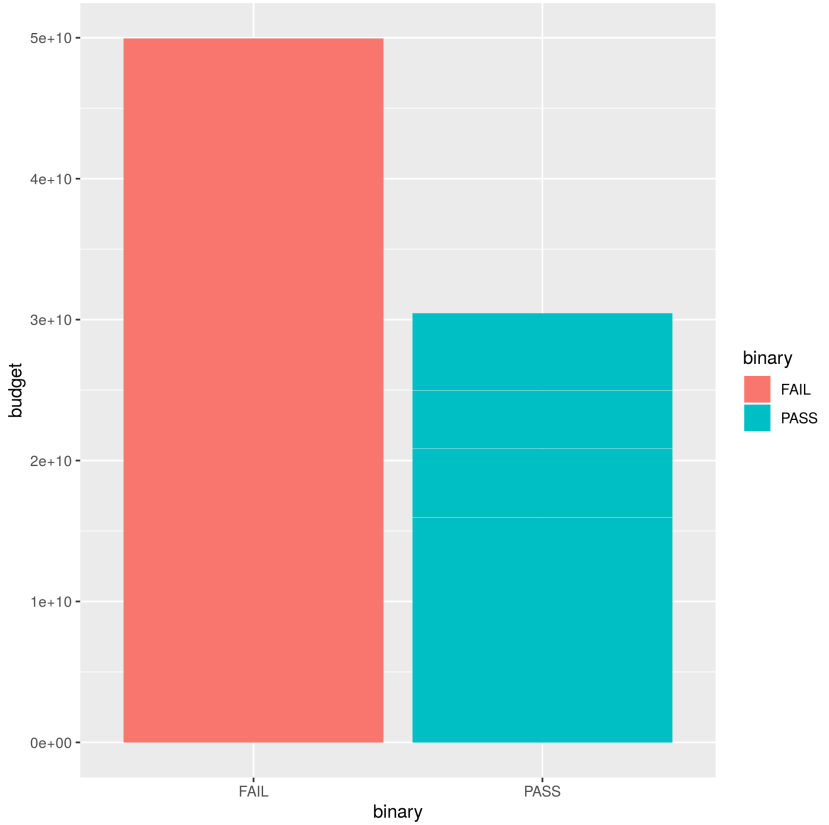
```
ggplot(bechdel, aes(
  x = clean_test,
  y = domgross
)) +
  geom_col(aes(fill = clean_test),
    width = 0.4
  )
```

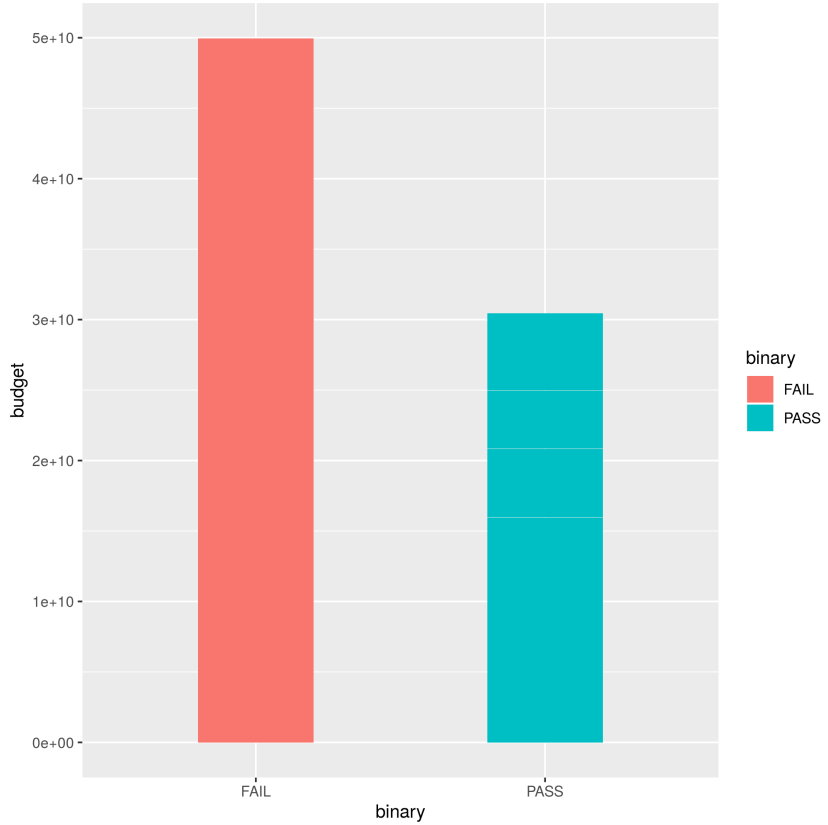


Your turn 7

02:00

- Add a width adjustment to this plot





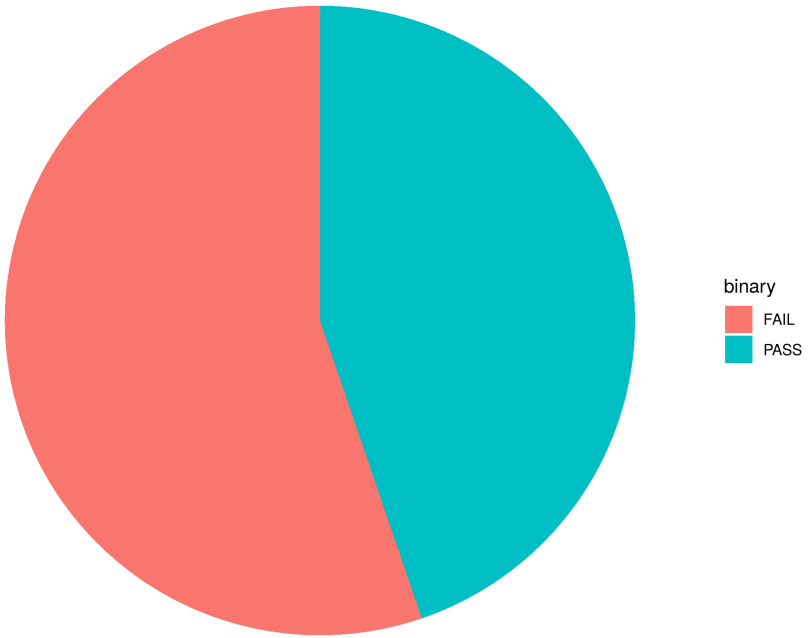
Your Answer 7

```
ggplot(bechdel, aes(x = binary ,
                    y = budget)) +
  geom_col(aes(fill = binary),width = 0.4)
```

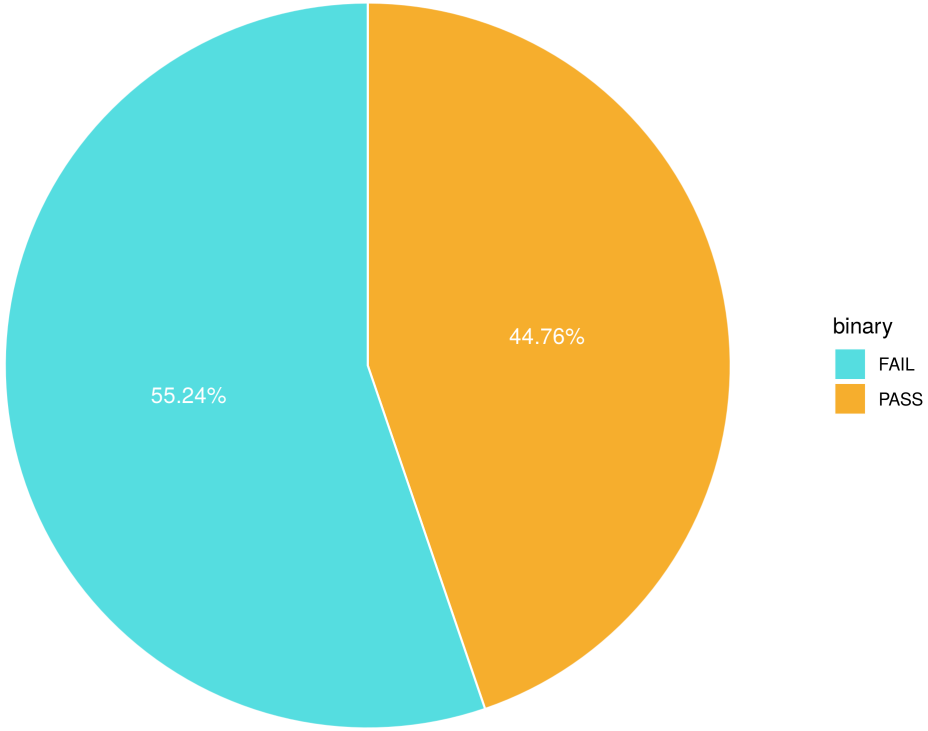
Pie Chart

- pie chart is created rotating the barchart
- pie chart round shaped

```
ggplot(data = bechdel, aes(
  x = factor(1),
  fill = binary
)) +
  geom_bar(width = 1) +
  coord_polar("y")
```

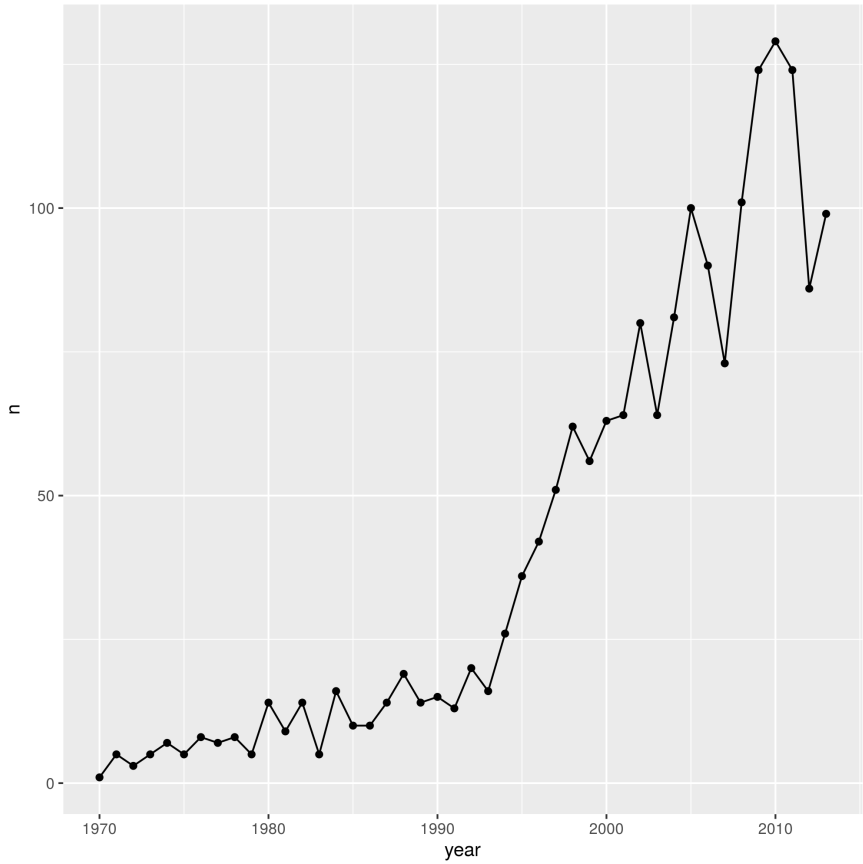


```r
# create new dataframe
binary_data <- bechdel %>%
  count(binary) %>%
  mutate(percent = (n / sum(n)) * 100)
ggplot(binary_data, aes(x = "", y = percent, fill = binary)) +
  geom_bar(width = 1, stat = "identity", color = "white") +
  # convert barplot to polar coordinates
  coord_polar("y", start = 0) +
  # add labels
  geom_text(aes(label = paste0(round(percent, 2), "%")),
    position = position_stack(vjust = 0.5), color = "white"
  ) +
  # add color scale manually
  scale_fill_manual(values = c("#55DDE0", "#F6AE2D")) +
  theme_void()
```

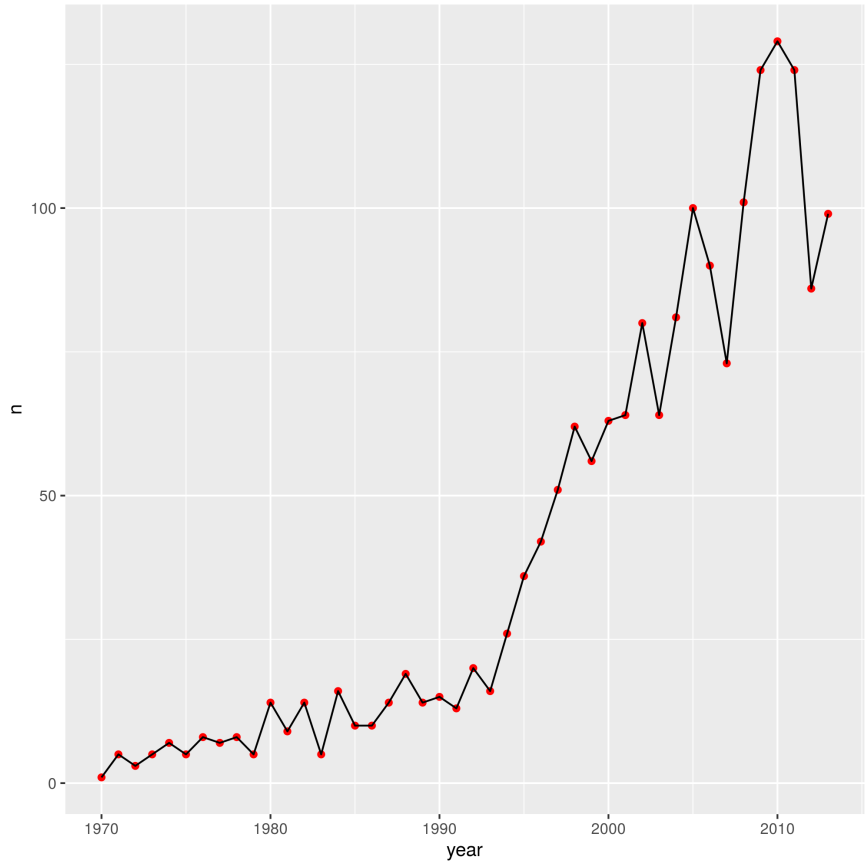

44.76%
55.24%
binary
FAIL
PASS

Lineplot

```
bechdel %>%
  count(year) %>%
  ggplot() +
  geom_point(mapping = aes(x = year, y
= n)) +
  geom_line(mapping = aes(x = year, y
= n))
```



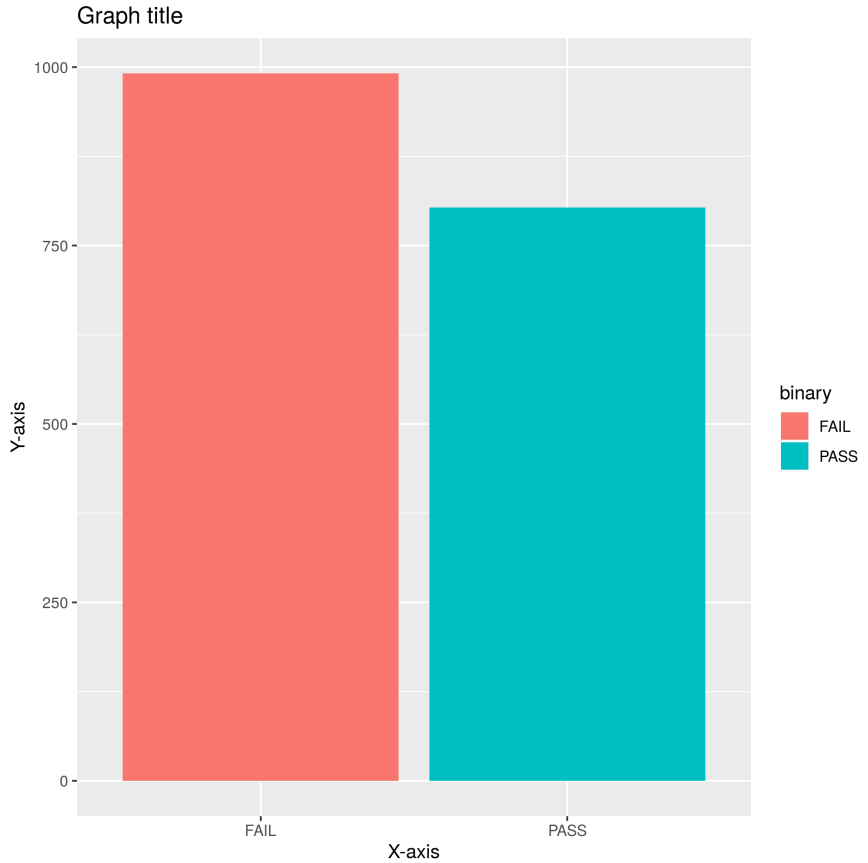
```
bechdel %>%
  count(year) %>%
  ggplot() +
  geom_point(
    mapping = aes(x = year, y = n),
    color = "red"
  ) +
  geom_line(
    mapping = aes(x = year, y = n)
  )
```



Title , Axis and caption

- **labs()** is used to add title ,caption etc

```
ggplot(
  data = bechdel,
  aes(x = binary)
) +
  geom_bar(aes(fill = binary)) +
  labs(
    title = "Graph title",
    x = "X-axis",
    y = "Y-axis",
    caption = "data source: xyz"
  )
```

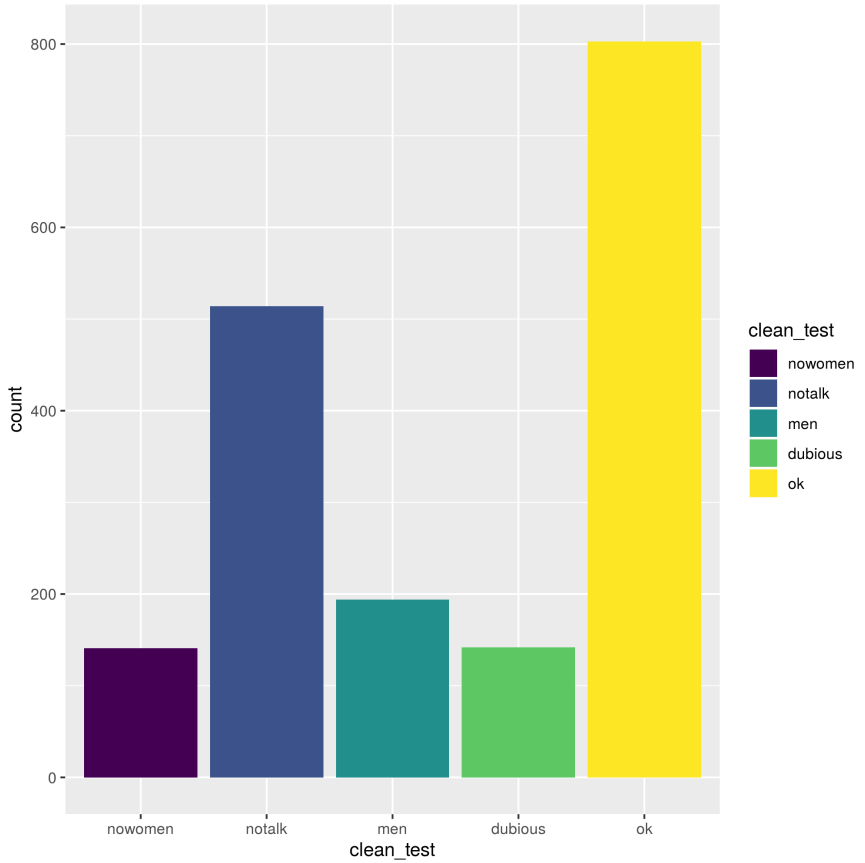


Your Turn 8

02:00

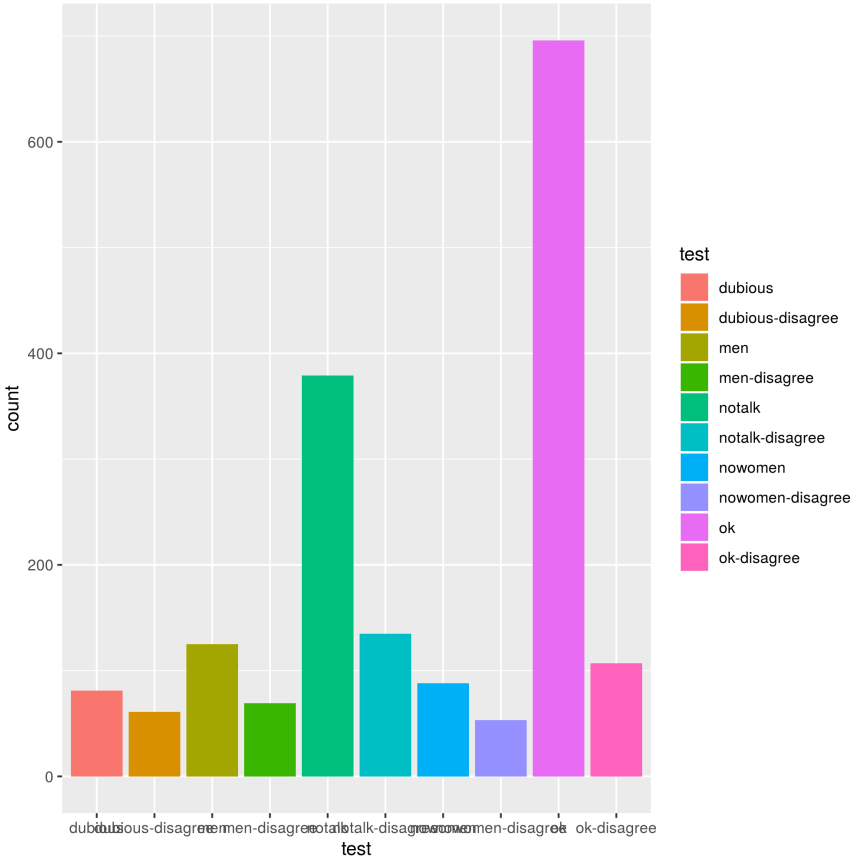
- Add Title,caption to below plot

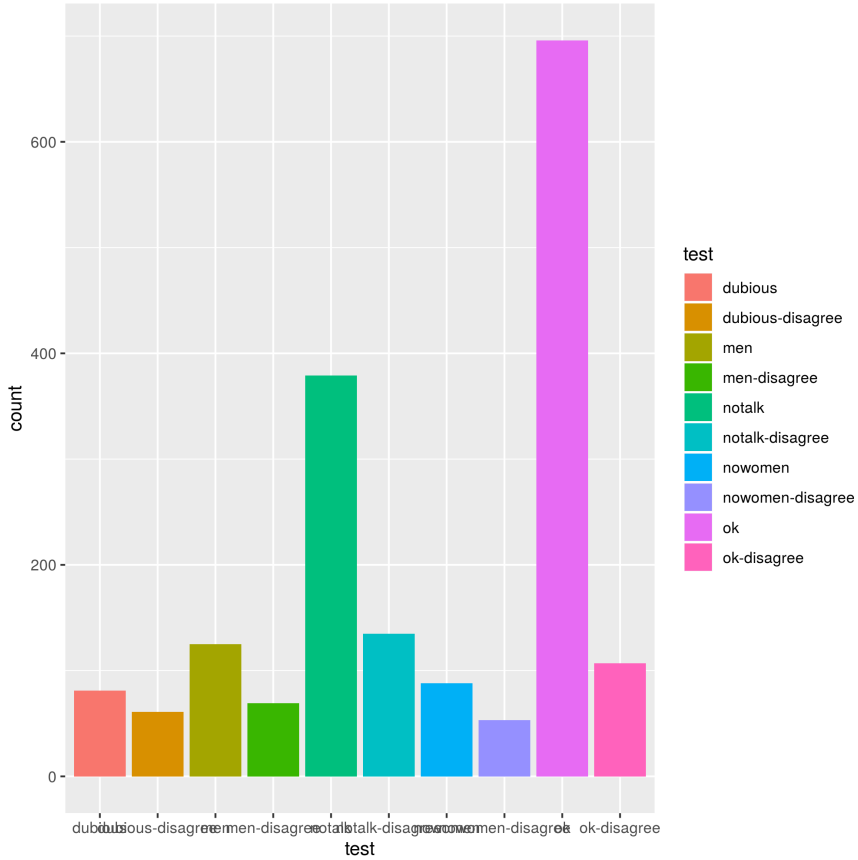
```
ggplot(data = bechdel,aes(x =
clean_test))+
  geom_bar(aes(fill = clean_test))
```



```
ggplot(data = bechdel,aes(x = clean_test))+
  geom_bar(aes(fill = clean_test)) +
    labs(
    title = "Count of the film based on Clean Test ",
    caption = "Data Source: FiveThirtyEight")
```

Change axis tick mark labels





```
ggplot(data = bechdel,aes(x = test))+
  geom_bar(aes(fill = test)) +
    theme(axis.text.x = element_text(angle = 90))
```

Flip the plot

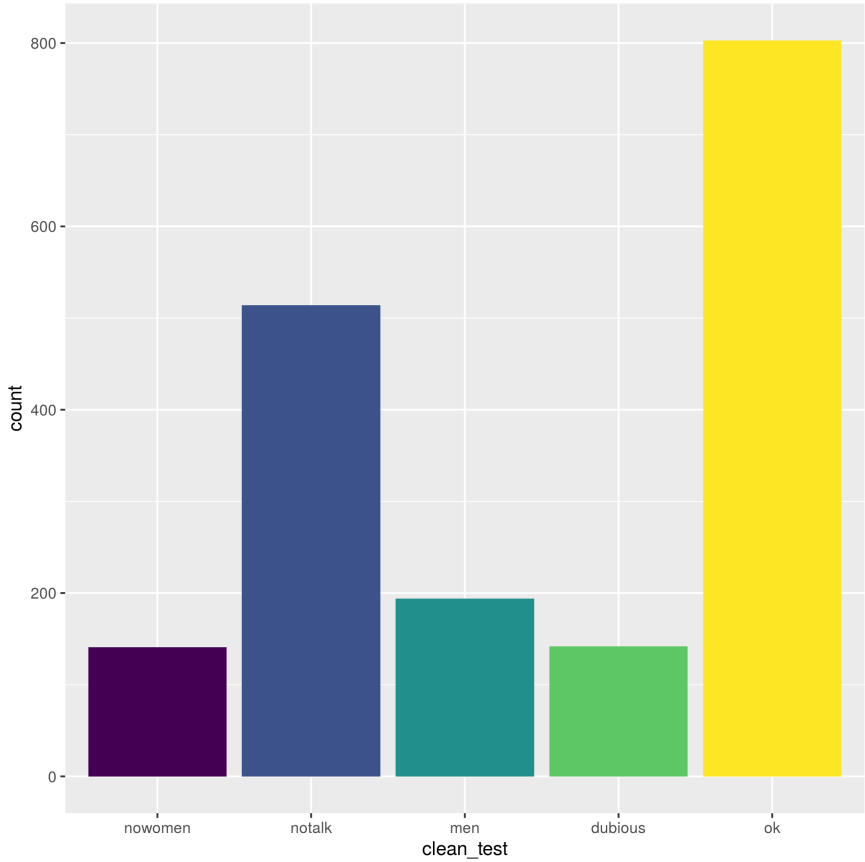
01:00

```
ggplot(data = bechdel,aes(x = test))+
  geom_bar(aes(fill = test)) +coord_flip()
```

Legend

- **Remove the legend**

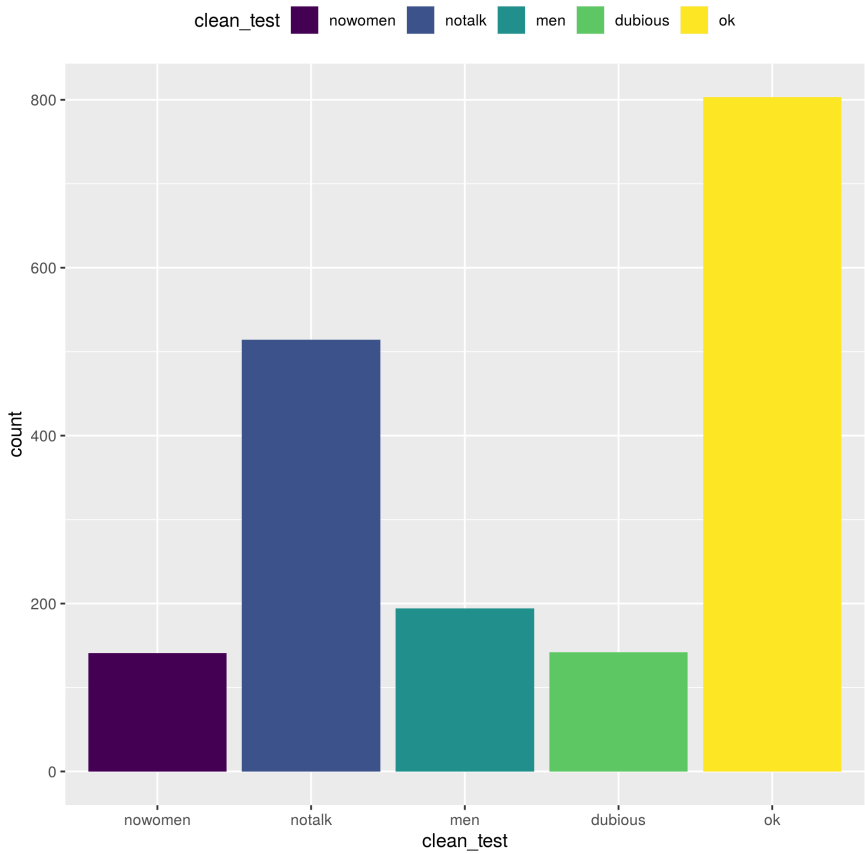
```
ggplot(data = bechdel) +
  geom_bar(mapping = aes(
    x = clean_test,
    fill = clean_test
  )) +
  guides(fill = FALSE)
```



Legend Position

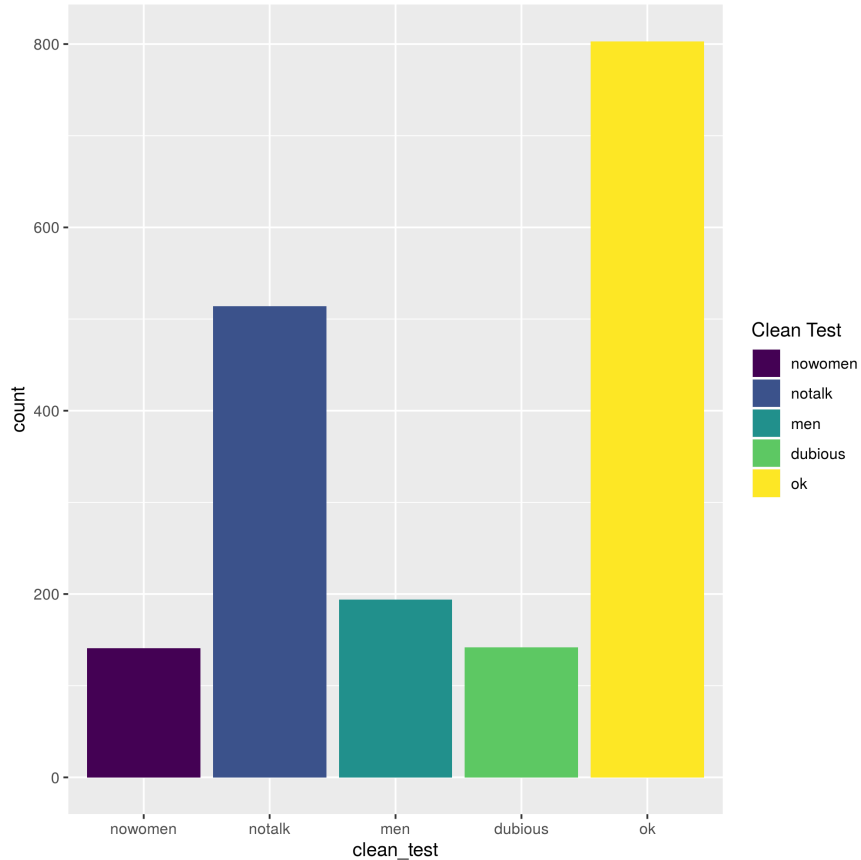
top,**bottom**,left,right

```
ggplot(data = bechdel) +
  geom_bar(aes(
    x = clean_test,
    fill = clean_test
  )) +
  theme(legend.position = "top")
```



Change legend title

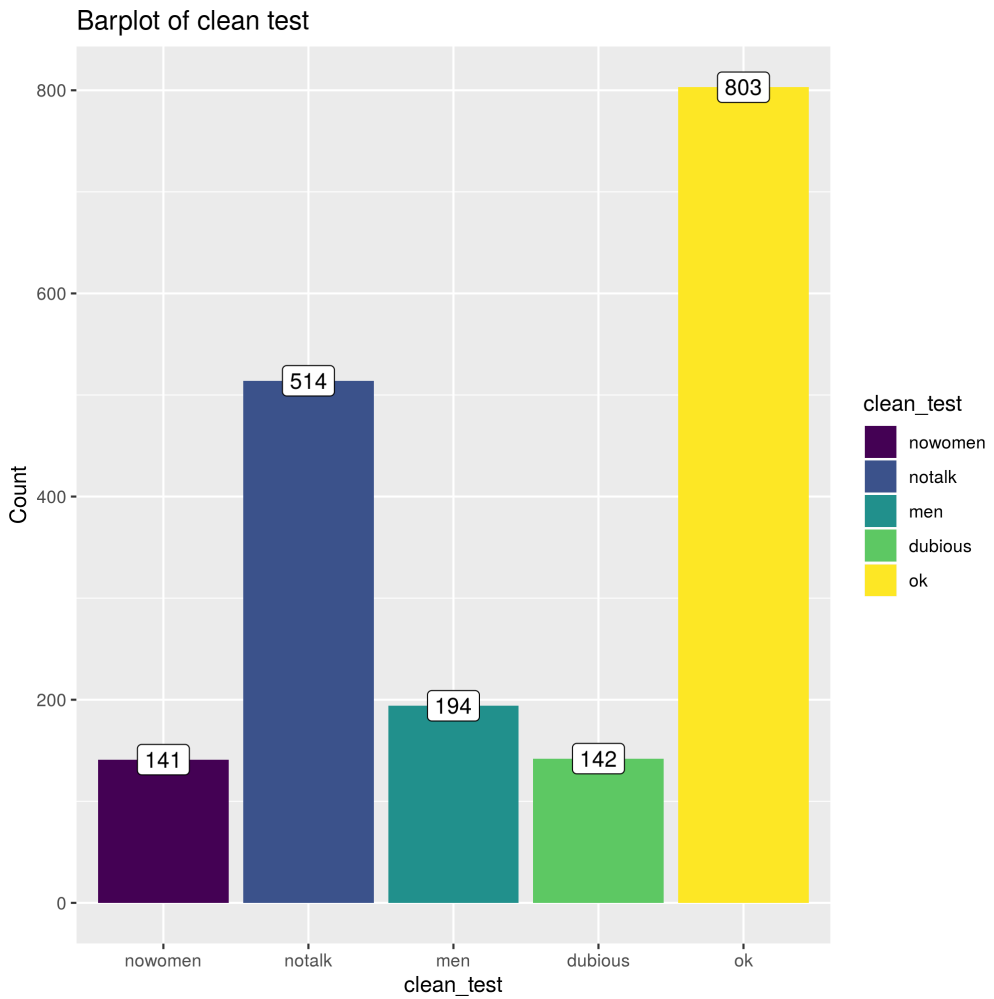
```
ggplot(data = bechdel) +
  geom_bar(
    mapping =
      aes(
        x = clean_test,
        fill = clean_test
      )
  ) +
  labs(fill = "Clean Test")
```



Labels

- **geom_text()** adds text directly to the plot.
- **geom_label()** create a a rectangle behind the text.

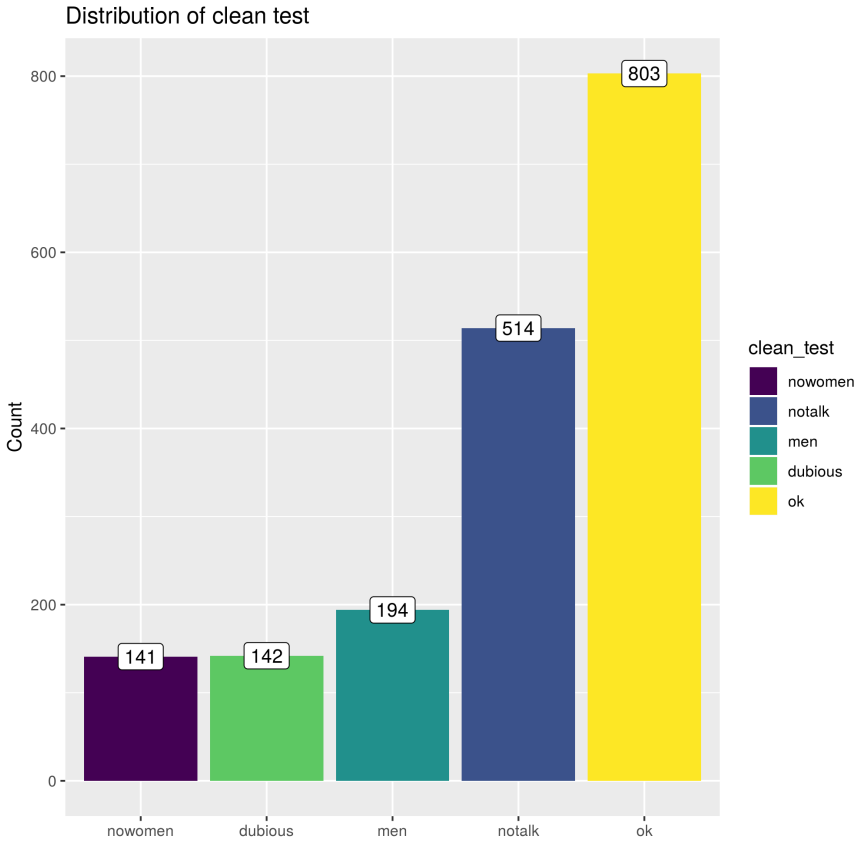
```
bechdel %>%
  count(clean_test) %>%
  ggplot() +
  geom_col(aes(x = clean_test, y = n,
fill = clean_test)) +
  geom_label(aes(x = clean_test, y =
n, label = n)) +
  labs(title = "Barplot of clean
test", y = "Count")
```


Barplot of clean test
Count
800
600
400
200
0
803
514
194
142
141
nowomen
notalk
men
dubious
ok
clean_test
clean_test
nowomen
notalk
men
dubious
ok

Arrange the barplot

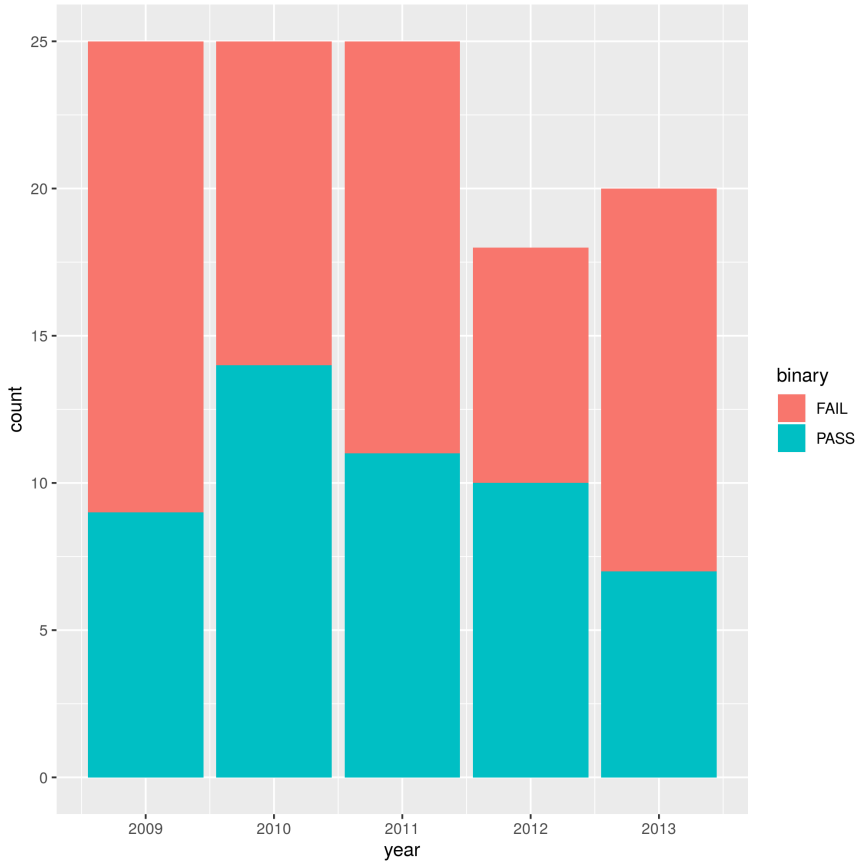
- **reorder()** arrange the values

```
bechdel %>%
  count(clean_test) %>%
  ggplot() +
  geom_col(aes(x = reorder(clean_test,
n), y = n, fill = clean_test)) +
  geom_label(aes(x = clean_test, y =
n, label = n)) +
  labs(title = "Distribution of clean
test", y = "Count", x = "")
```

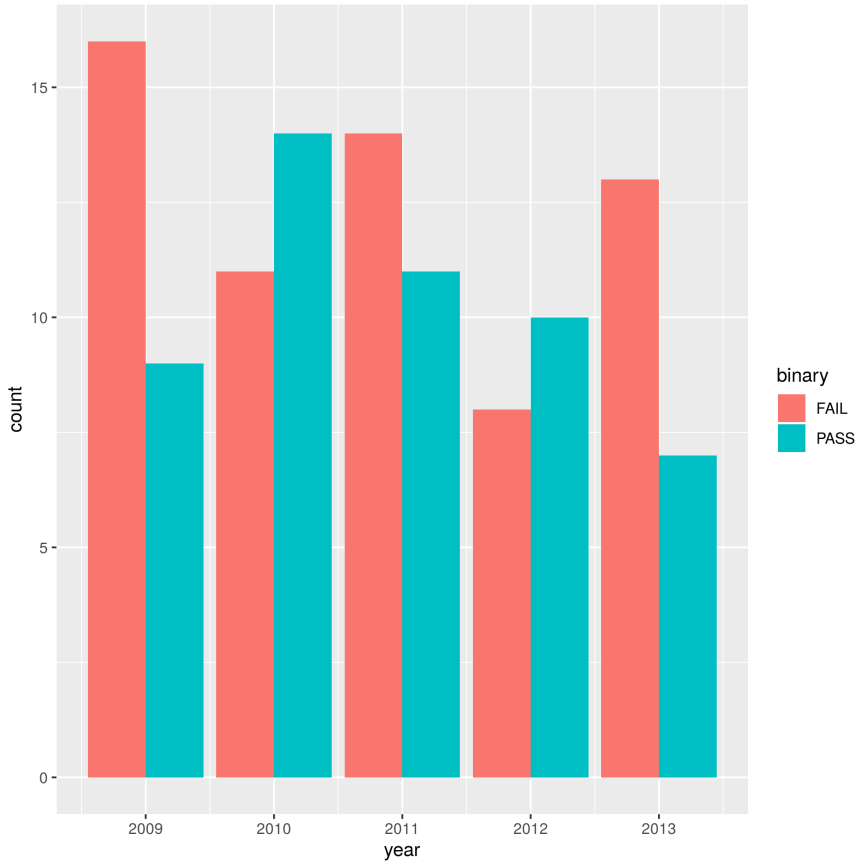


Position

```
bechdel %>%
  filter(
    year ==
      c(2013, 2012, 2011, 2010, 2009)
  ) %>%
  ggplot(aes(x = year)) +
  geom_bar(aes(fill = binary))
```



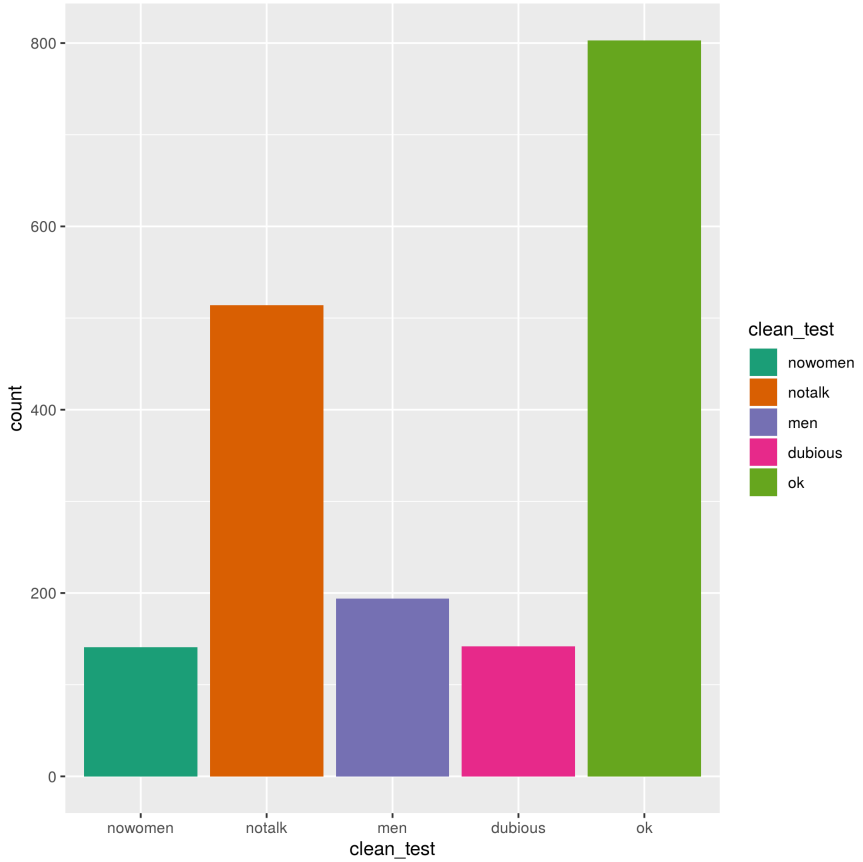
```
bechdel %>%
  filter(year == c(2013, 2012, 2011,
2010, 2009)) %>%
  ggplot(aes(x = year)) +
  geom_bar(aes(fill = binary),
position = "dodge")
```



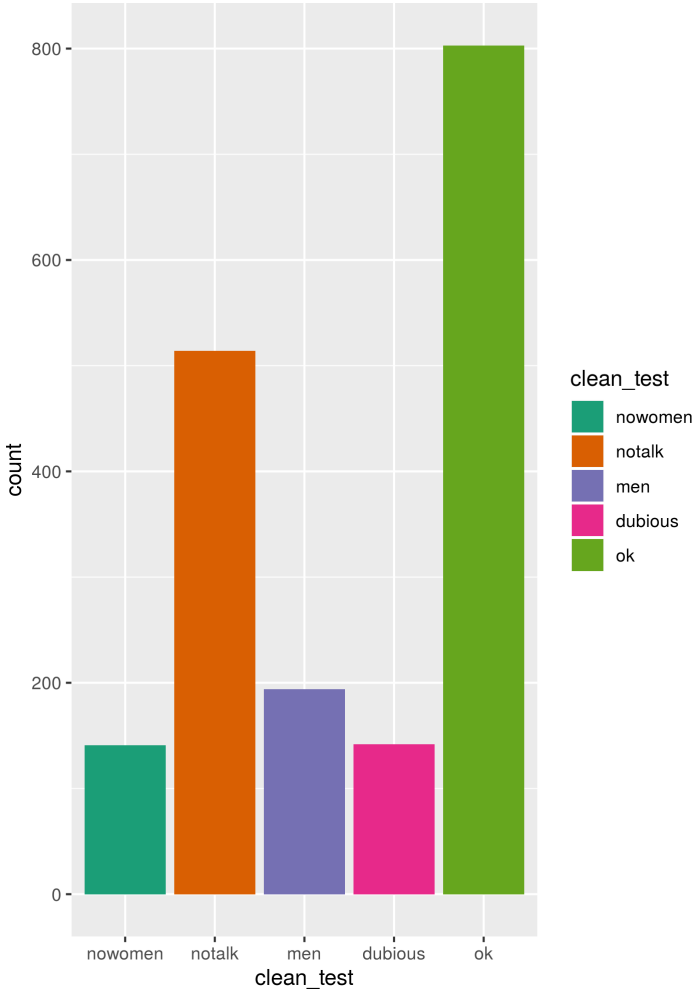
Scales

- assign different color palette
- Enter **?scale_color_brewer**

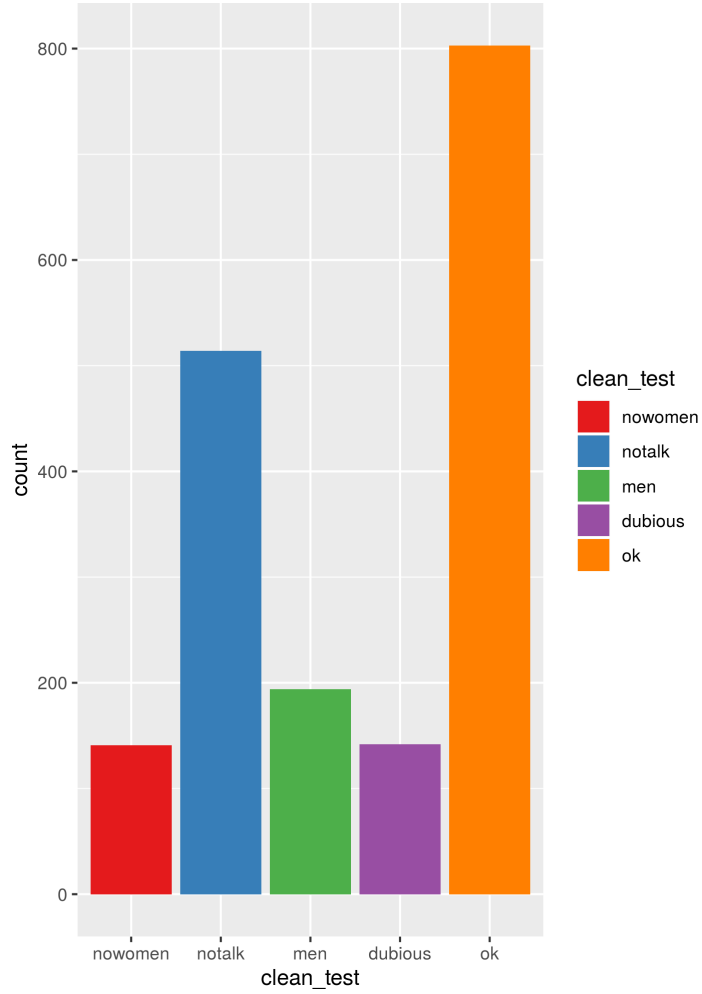
```
ggplot(data = bechdel, aes(x =
clean_test)) +
  geom_bar(aes(fill = clean_test)) +
  scale_fill_brewer(palette = "Dark2")
```



- Palette:Dark2



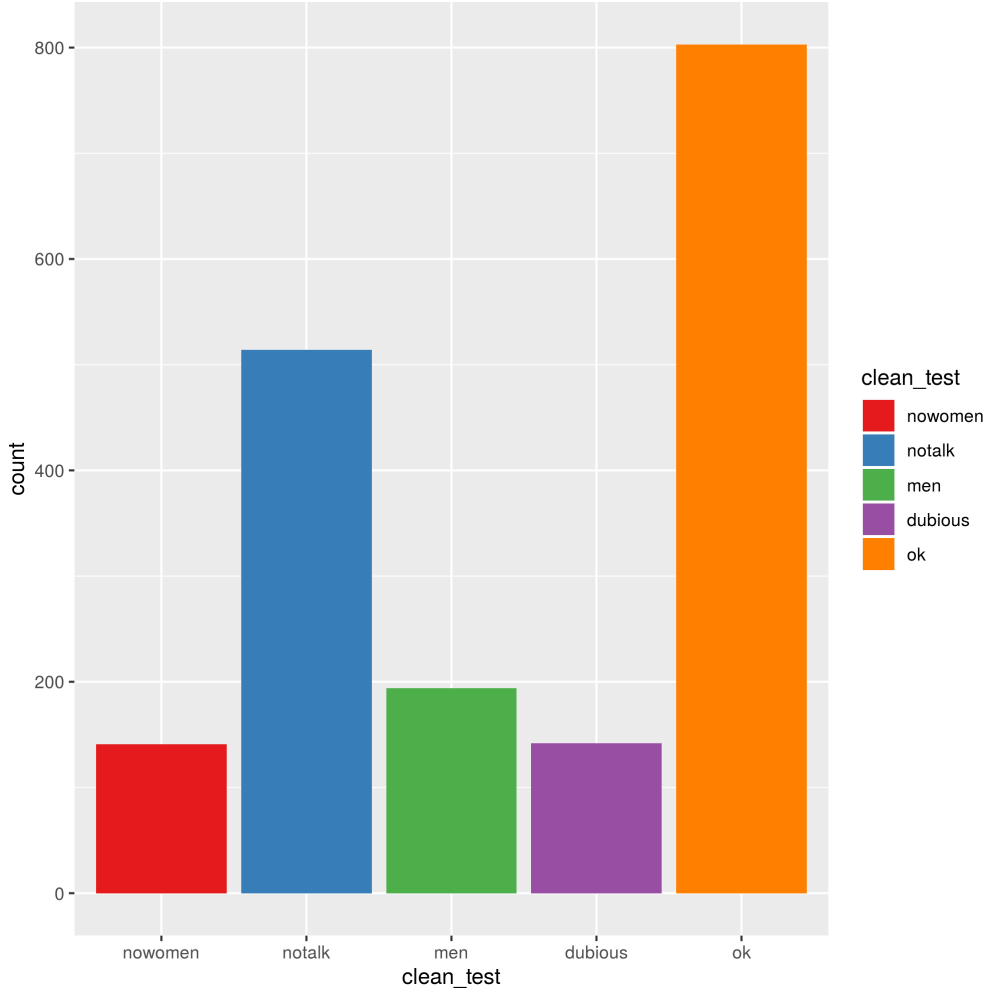
- Palette:Set1



Your turn 10

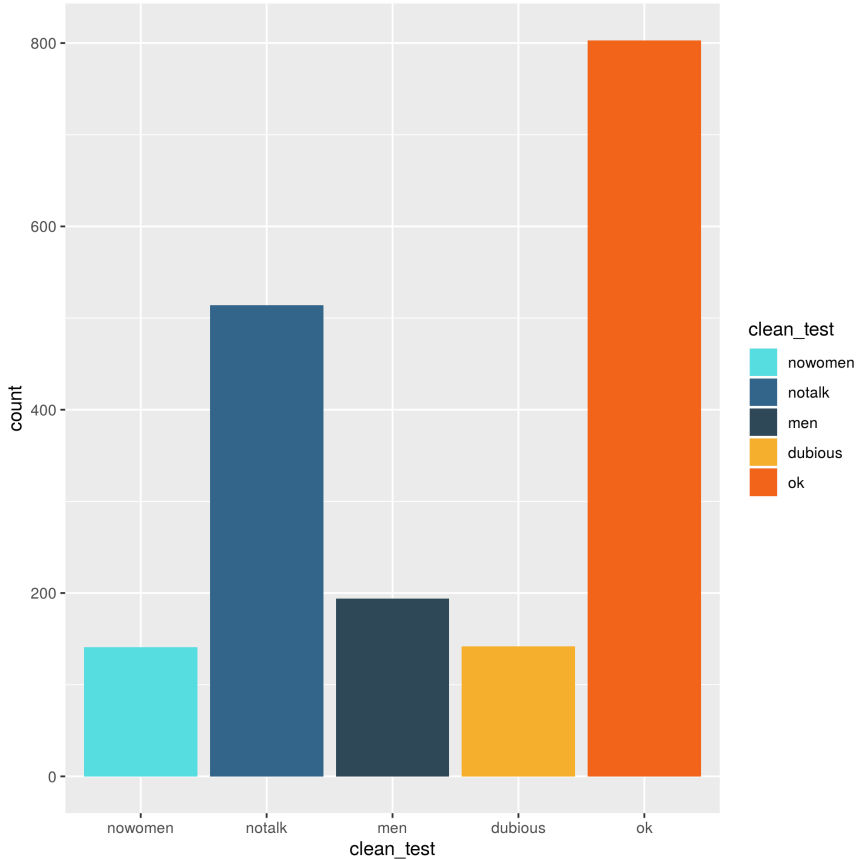
02:00

- Pratice with different color palette



Add color manually

```
ggplot(data = bechdel, aes(x =
clean_test)) +
  geom_bar(aes(fill = clean_test)) +
  scale_fill_manual(
    values = c(
      "#55DDE0", "#33658A",
      "#2F4858", "#F6AE2D",
      "#F26419"
    )
  )
```



Facets

- Layout panel for the graphs
- **facet_wrap()** & **facet_grid()**

Facet_wrap()

```
ggplot(data = bechdel) +
 geom_point(mapping = aes(x = budget, y = domgross, color = clean_test)) +
 facet_wrap(~ clean_test)
```

Facet_grid()

```
ggplot(data = bechdel) +
  geom_point(mapping = aes(x = budget, y = domgross, color = clean_test)) +
  facet_grid(decade_code ~ clean_test)
```

Themes

theme_bw()

```
ggplot(data = bechdel) +
 geom_point(mapping = aes(x = budget, y = intgross, color = clean_test)) +
    theme_bw()
```

theme_classic()

```
ggplot(data = bechdel) +
 geom_point(mapping = aes(x = budget, y = intgross, color = clean_test))
```

Your turn 11

02:00

- Try different theme for this graph

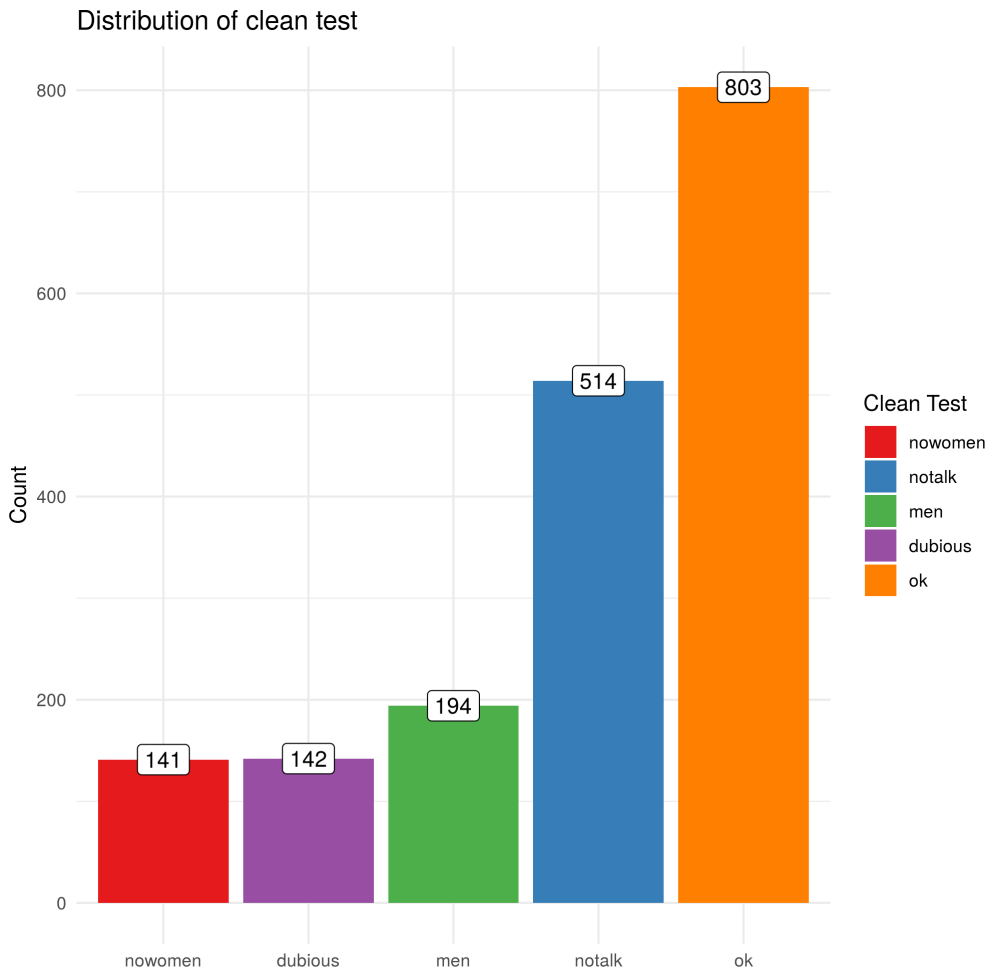
```
ggplot(data = bechdel, aes(x = clean_test)) +
  geom_bar(aes(fill = clean_test))
```

Saving plots

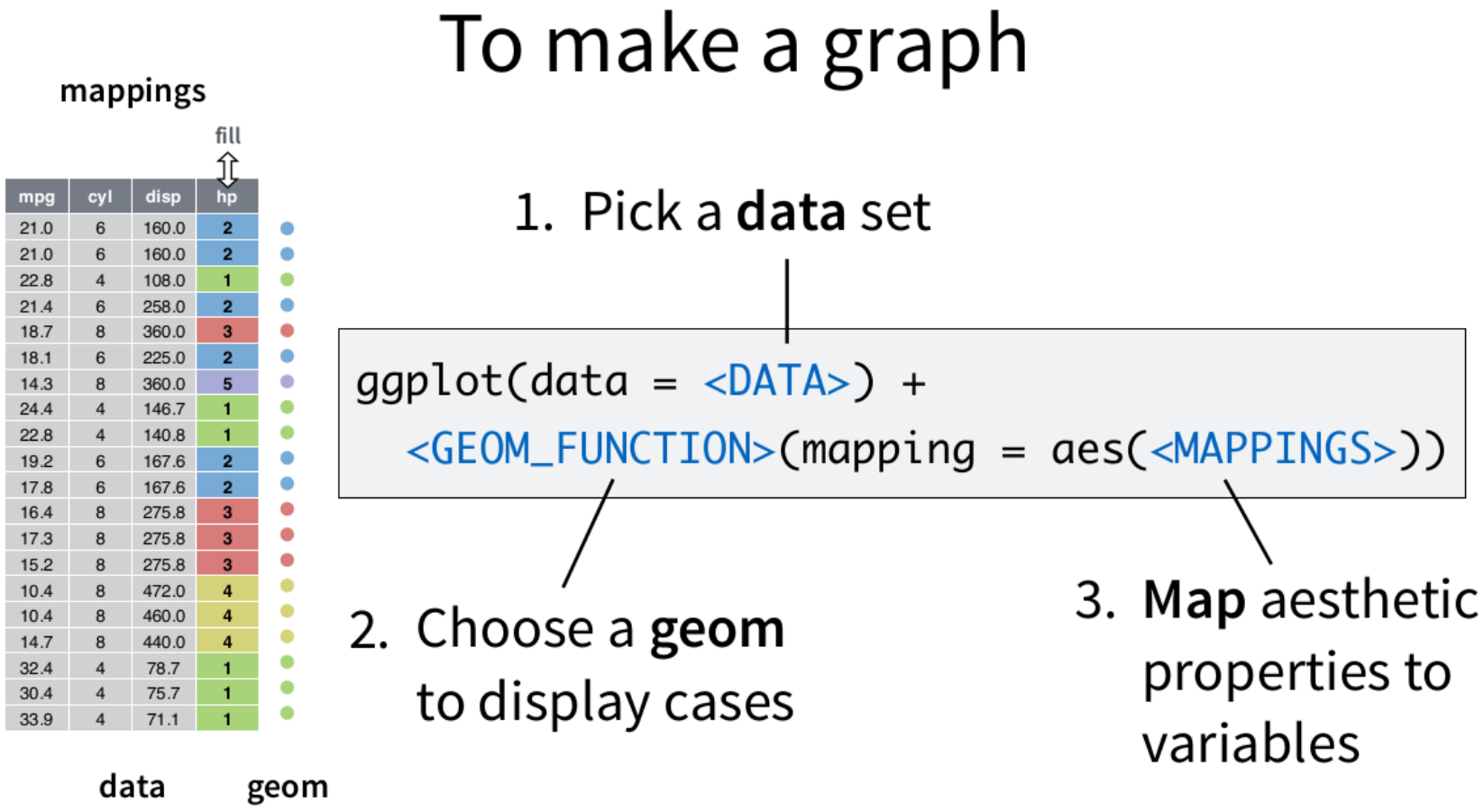
- **ggsave()** is used to save plot.

```
# assign the ggplot graph to myplot variable

myplot <- bechdel %>%
  count(clean_test) %>%
  ggplot() +
  # plot barplot
  geom_col(aes(x = reorder(clean_test, n), y = n, fill = clean_test)) +
  # adding labels to plot
  geom_label(aes(x = clean_test, y = n, label = n)) +
  # adding title and edit xaxis and y axis
  labs(title = "Distribution of clean test", y = "Count", x = "", fill = "Clean
Test") +
  scale_fill_brewer(palette = "Set1") +
  # use theme minimal
  theme_minimal()
```


Distribution of clean test
Count
800
600
400
200
0
803
514
194
141
142
nowomen
dubious
men
notalk
ok
Clean Test
nowomen
notalk
men
dubious
ok

ggplot template



Data Visualization with ggplot2