Part 6 - Section 2: Getting start

Dr. Nguyen Quang Huy

May 16, 2020

Getting start

In this section, you'll learn how to quickly produce useful graphics with *ggplot2* including:

- The *gapminder* dataset in *dslabs* package.
- The three key components of every plot: data, aesthetics and geoms.
- How to add additional variables to a plot with aesthetics.
- How to display additional categorical variables in a plot using facetting.
- The different geoms that you can use to create different types of plots.
- How to modify the axes.
- Things you can do with a plot object other than display it.

Gapminder data

```
library(dslabs)
head(gapminder)
```

```
##                country year infant_mortality life_expectancy
## 1              Albania 1960           115.40            62.87
## 2              Algeria 1960           148.20            47.50
## 3               Angola 1960           208.00            35.98
## 4  Antigua and Barbuda 1960               NA            62.97
## 5            Argentina 1960            59.87            65.39
## 6              Armenia 1960               NA            66.86
##   population          gdp continent          region
## 1    1636054           NA    Europe Southern Europe
## 2   11124892  13828152297    Africa Northern Africa
## 3    5270844           NA    Africa   Middle Africa
## 4      54681           NA  Americas       Caribbean
## 5   20619075 108322326649  Americas   South America
```

Gapminder data

Health and income outcomes for 184 countries from 1960 to 2010

- **country**: The name of countries.
- **Year**: from 1960 to 2016 (many NA values from 2010 to 2016)
- **infant_mortality**: numbers of infant deaths per 1000.
- **life_expectancy**: life expectancy in years.
- **fertility**: the average number of children per woman.
- **population**: country population.
- **gpd** GDP of country in years
- **continent**: 5 continents
- **region**: 22 geographical regions.

Gapminder data

- List several functions that you could use to get more information about the dataset ?

Gapminder data

- List several functions that you could use to get more information about the dataset ?

```
str(gapminder) #head(gapminder), view(gapminder)
```

Gapminder data

- List several functions that you could use to get more information about the dataset ?

```
str(gapminder) #head(gapminder), view(gapminder)
```

- GDP per capita is often considered an indicator of a country's standard of living, which is calculated by dividing the GDP of a country by its population. Add **GDP_per_capita** variable to *gapminder* dataset.

Answer:

```
library(tidyverse)
dat<-mutate(gapminder,GDP_per_capita = gdp/population)
```

Gapminder data

- List names of 3 countries have the highest *GDP_per_capita* and 3 countries have the lowest *GDP_per_capita* in 2010 ?

Gapminder data

- List names of 3 countries have the highest *GDP_per_capita* and 3 countries have the lowest *GDP_per_capita* in 2010 ?

Answer:

```
dat2010<-filter(dat,year==2010)
ind<-order(dat2010$GDP_per_capita)[1:3]
as.character(dat$country[ind])
```

```
## [1] "Congo, Dem. Rep." "Burundi"          "Guinea-Bissau"
```

```
ind<-order(dat2010$GDP_per_capita,decreasing = TRUE)[1:3]
as.character(dat$country[ind])
```

```
## [1] "Luxembourg" "Japan"      "Norway"
```

Key components

Every ggplot2 plot has three key components:

- 1. **Data.**
- 2. **Aesthetic mappings (aes)**: how variables in the data are mapped to aesthetic attributes.
- 3. At least one **geometric object (geoms)** represent what you actually see on the plot: points, lines, polygons, etc.

We want to analyze the relationship between *GDP_per_capita* and *fertility* in 2010.

```
ggplot(dat2010,aes(x=fertility,y=GDP_per_capita))+
  geom_point()
```

In this code: 1. **Data**: dat2010;

Key components

Every ggplot2 plot has three key components:

- 1. **Data.**
- 2. **Aesthetic mappings (aes)**: how variables in the data are mapped to aesthetic attributes.
- 3. At least one **geometric object (geoms)** represent what you actually see on the plot: points, lines, polygons, etc.

We want to analyze the relationship between *GDP_per_capita* and *fertility* in 2010.

```
ggplot(dat2010,aes(x=fertility,y=GDP_per_capita))+
  geom_point()
```

In this code: 1. **Data**: dat2010; 2. **Mappings**: *fertility* $\rightarrow$ x position and *GDP_per_capita* $\rightarrow$ y position;

Key components

Every ggplot2 plot has three key components:

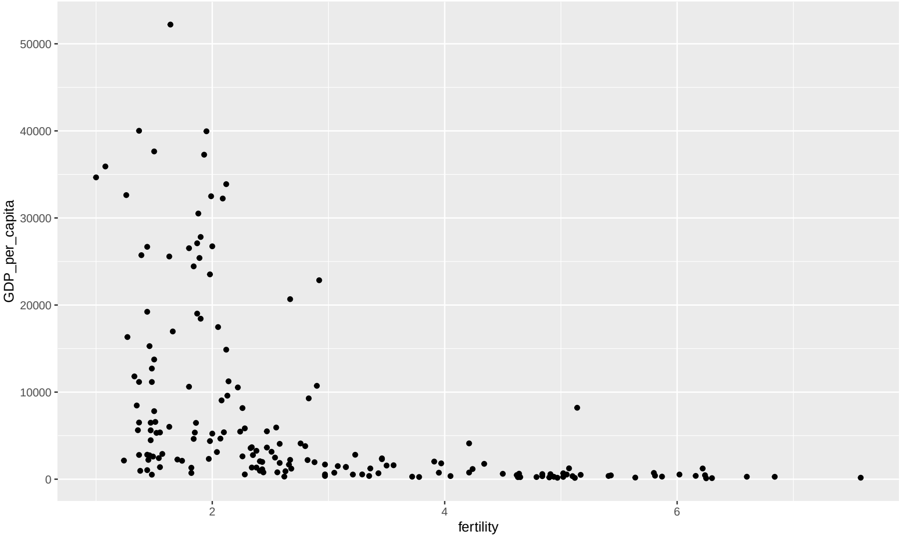
- 1. **Data.**
- 2. **Aesthetic mappings (aes)**: how variables in the data are mapped to aesthetic attributes.
- 3. At least one **geometric object (geoms)** represent what you actually see on the plot: points, lines, polygons, etc.

We want to analyze the relationship between *GDP_per_capita* and *fertility* in 2010.

```
ggplot(dat2010,aes(x=fertility,y=GDP_per_capita))+
  geom_point()
```

In this code: 1. **Data**: dat2010; 2. **Mappings**: *fertility* $\rightarrow$ x position and *GDP_per_capita* $\rightarrow$ y position; 3. **Geometric object**: points.

Key components



Key components

- What conclusion can be drawn from the plot of *fertility* and *GDP_per_capita*?

Key components

- What conclusion can be drawn from the plot of *fertility* and *GDP_per_capita*?
- There are also some interesting outliers: some coutries with high fertility have higher GDP per capita than average. Which countries do you think they are?

Key components

- What conclusion can be drawn from the plot of *fertility* and *GDP_per_capita*?
- There are also some interesting outliers: some coutries with high fertility have higher GDP per capita than average. Which countries do you think they are?
- Using *ggplot2* to describe the relationship between *infant_mortality* and *life_expectancy* ?

Key components

- What conclusion can be drawn from the plot of *fertility* and *GDP_per_capita*?
- There are also some interesting outliers: some coutries with high fertility have higher GDP per capita than average. Which countries do you think they are?
- Using *ggplot2* to describe the relationship between *infant_mortality* and *life_expectancy* ?

Answer

```
ggplot(dat2010,aes(x=infant_mortality,y=life_expectancy))+
  geom_point()
```

There is a negative linear relationship between *infant_mortality* and *life_expectancy*.

Key components

Describe the **data**, **aesthetic mappings** and **geometric objects** used for each of the following plots. You should predict what the plot will look like before running the code.

- 1. ggplot(mpg, aes(cty, hwy)) + geom_point()
- 2. ggplot(diamonds, aes(carat, price)) + geom_point()
- 3. ggplot(economics, aes(date, unemploy)) + geom_line()
- 4. ggplot(dat2010, aes(GDP_per_capita)) + geom_histogram()

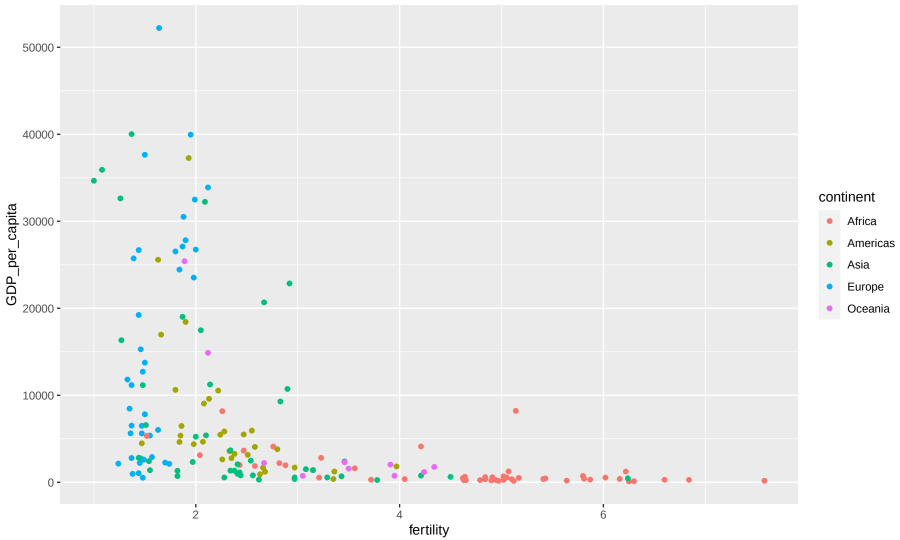
Aesthetic attributes

We can use other aesthetics like color, shape and size to add variables to a plot

```
ggplot(dat2010,aes(fertility,GDP_per_capita,color=continent))+
  geom_point()
```

- Continent variable is added using *aes()* in the same way as **x** and **y**.
- *ggplot2* converts values "Africa", "Americas", "Asia", "Europe" and "Oceania" into colors with a **scale**
- **Scale** is also responsible for creating a guide, an axis or legend, that allows you to read the plot, converting aesthetic values back into data values.
- **Scale** will be discussed in section 6

Aesthetic attributes



Aesthetic attributes

If you want to set an aesthetic to a fixed value, without scaling it, do so in the individual layer outside of **aes()**. Try the following codes

Code 1

```
ggplot(dat2010,aes(fertility,GDP_per_capita,color="blue"))+
  geom_point()
```

Code 2

```
ggplot(dat2010,aes(fertility,GDP_per_capita,color="blue"))+
  geom_point(aes(color=continents))
```

Code 3

```
ggplot(dat2010,aes(fertility,GDP_per_capita))+
  geom_point(color="blue")
```

Aesthetic attributes

- Different types of aesthetic attributes work better with different types of variables.
- Colour and shape work well with categorical variables.
- Size works well for continuous variables.
- If there is a lot of data it can be hard to distinguish different groups (using facetting as an alternative solution).
- It's difficult to see the simultaneous relationships among colour and shape and size.
- Instead of trying to make one very complex plot that shows everything at once, you should create a series of simple plots that tell your story

Aesthetic attributes

- 1. Mapping size and shape to **continent** (a categorical variable). What happens when you map size to **continent** ?
- 2. Mapping color, size and shape to **population** (a continuous variable).
  - What happens when you map color to **population** ?
  - What happens when you map shape to **population** ?

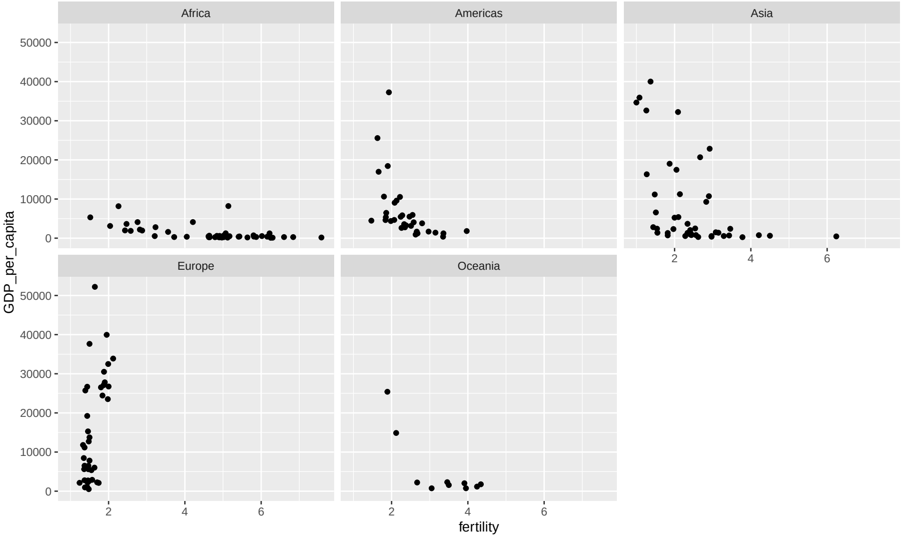
Facetting

Facetting is a technique for displaying additional categorical variables.

- Facetting creates tables of graphics by splitting the data into subsets and displaying the same graph for each subset.
- There are two types of facetting: grid and wrapped.
    - To facet a plot using wrapped, you add a facetting specification with *facet_wrap(∼ **categorical variable**)*
    - To facet a plot using grid, you add a facetting specification with *facet_grid(∼ **categorical variable**)*
    - The difference between grid and wrapped facetting will be discussed in section 7

```
ggplot(dat2010,aes(fertility,GDP_per_capita))+
  geom_point()+facet_wrap(~continent)
```

Facetting



Africa
Americas
Asia
Europe
Oceania
GDP_per_capita
50000
40000
30000
20000
10000
0
2
4
6
fertility

Facetting

- Facetting is an alternative to using aesthetics (like colour, shape or size) to differentiate groups.
- Using facetting or aesthetics based on the relative positions of the subsets.
  - With facetting, there is no overlap between the groups.
  - Facetting is good if the groups overlap a lot, but it does make small differences harder to see.
  - When using aesthetics to differentiate groups, the groups are close together and may overlap
  - Small differences are easier to see.

Facetting

- 1. What happen when you try to facet by a continuous variable like *population*?
- 2. Read the documentation for *facet_wrap()*. What arguments can you use to control how many rows and columns appear in the output? How to apply these arguments to **dat2010** data?
- 3. What does the scales argument to *facet_wrap()* do? When might you use it?

Facetting

ncol and *nrow* arguments in *facet_wrap()*.

```
ggplot(dat2010,aes(fertility,GDP_per_capita))+
  geom_point()+facet_wrap(~continent,ncol=5,nrow=1)
```

```
ggplot(dat2010,aes(fertility,GDP_per_capita))+
  geom_point()+facet_wrap(~continent,ncol=1,nrow=5)
```

The default value of *scales* argument is "fixed". It can be "free", "free_x", or "free_y".

```
ggplot(dat2010,aes(fertility,GDP_per_capita))+
  geom_point()+facet_wrap(~continent,scales = "free_x")
```

```
ggplot(dat2010,aes(fertility,GDP_per_capita))+
  geom_point()+facet_wrap(~continent,scales = "free_y")
```

Practice

- 1. Read the documentation for **mpg** data and answer the following questions
  - How many observations ?
  - How many variables? What is the meaning of each variables?
- 2. Vietnam uses fuel consumption (fuel consumed over 100km) rather than fuel economy (distance travelled in miles per gallon). Converting *cty* and *hwy* into variables *vn_cty* and *vn_hwy* - fuel consumption ratings in L/100 km.
- 3. Remove the specification of drive train, including "quattro","2wd", 4wd", and "awd", from *model* variable.
- 4. Which manufacturer has the most the models in this dataset?

Practice

1.

```
# ? mpg
str(mpg)
```

```
## tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
##  $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ..
##  $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
##  $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.
##  $ year        : int [1:234] 1999 1999 2008 2008 1999 1999
##  $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manua
##  $ drv         : chr [1:234] "f" "f" "f" "f" ...
##  $ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20
##  $ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28
##  $ fl          : chr [1:234] "p" "p" "p" "p" ...
##  $ class       : chr [1:234] "compact" "compact" "compact"
```

Practice

2. Fuel consumption ratings in L/100 km.

Practice

2. Fuel consumption ratings in L/100 km.

```
dat<-mutate(mpg,vn_cty = 100/((cty * 1.6093)/3.7854),
               vn_hwy = 100/((hwy * 1.6093)/3.7854))
mean(dat$vn_cty)
```

```
## [1] 14.84167
```

```
mean(dat$vn_hwy)
```

```
## [1] 10.73374
```

Practice

2. Fuel consumption ratings in L/100 km.

```
dat<-mutate(mpg,vn_cty = 100/((cty * 1.6093)/3.7854),
              vn_hwy = 100/((hwy * 1.6093)/3.7854))
mean(dat$vn_cty)
```

```
## [1] 14.84167
```

```
mean(dat$vn_hwy)
```

```
## [1] 10.73374
```

3. Remove the specification of drive train

Practice

2. Fuel consumption ratings in L/100 km.

```
dat<-mutate(mpg,vn_cty = 100/((cty * 1.6093)/3.7854),
            vn_hwy = 100/((hwy * 1.6093)/3.7854))
mean(dat$vn_cty)
```

```
## [1] 14.84167
```

```
mean(dat$vn_hwy)
```

```
## [1] 10.73374
```

3. Remove the specification of drive train

```
dat$model<-str_replace(dat$model," 2wd","")
dat$model<-str_replace(dat$model," 4wd","")
dat$model<-str_replace(dat$model," awd","")
dat$model<-str_replace(dat$model," quattro","")
```

Practice

4. Which manufacturer has the most the models in this dataset?

Practice

4. Which manufacturer has the most the models in this dataset?

```
dat1<-group_by(dat,model)%>%
  summarize(manufacturer=manufacturer)%>%
  group_by(manufacturer)%>%
  summarize(number_model=length(manufacturer))%>%
  arrange(desc(number_model))
head(dat1,3)
```

```
## # A tibble: 3 x 2
##   manufacturer number_model
##   <chr>               <int>
## 1 dodge                  37
## 2 toyota                 34
## 3 volkswagen             27
```

Practice

- 5. Describe the relationship between *vn_cty* and *vn_hwy* using *ggplot2*. What remarks can be drawn from the plot?
- 6. Describe the relationship between *displ* and *vn_cty* using *ggplot2*. What remarks can be drawn from the plot? Is there any outlier?
- 7. Add *class* variable in the plot using color aesthetic. What remarks can be drawn from the plot? Can you explain the outliers?
- 8. Use facetting to explore the three-way relationship between fuel economy, engine size, and number of cylinders. How does facetting by number of cylinders change your assessment of the relationship between engine size and fuel economy?

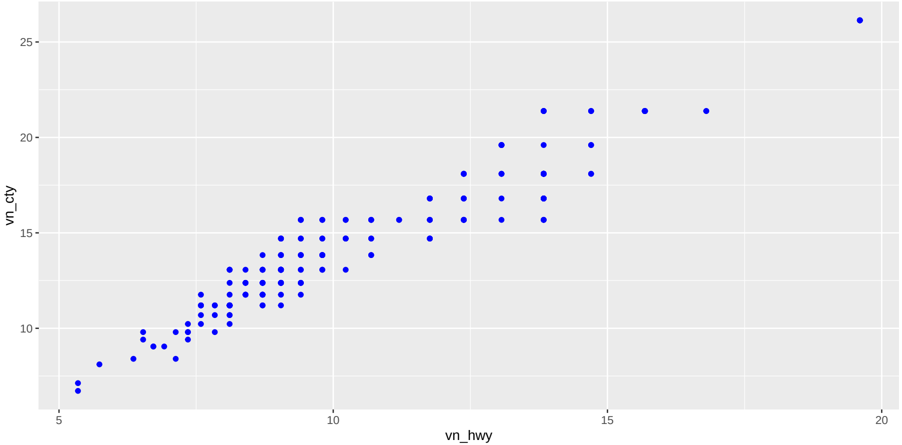
Practice

5. The relationship between *vn_cty* and *vn_hwy*

Practice

5. The relationship between *vn_cty* and *vn_hwy*

```
dat%>%ggplot(aes(vn_hwy,vn_cty))+geom_point(color="blue")
```



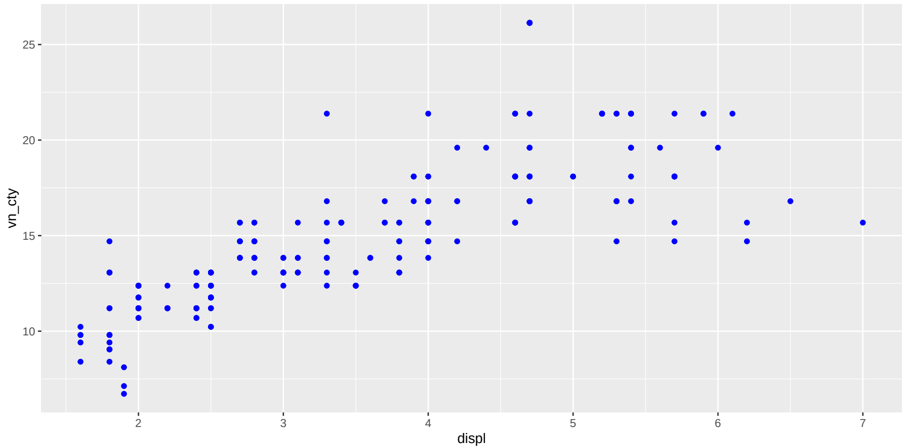
Practice

6. The relationship between *vn_cty* and *displ*

Practice

6. The relationship between *vn_cty* and *displ*

```
dat%>%ggplot(aes(displ, vn_cty))+geom_point(color="blue")
```



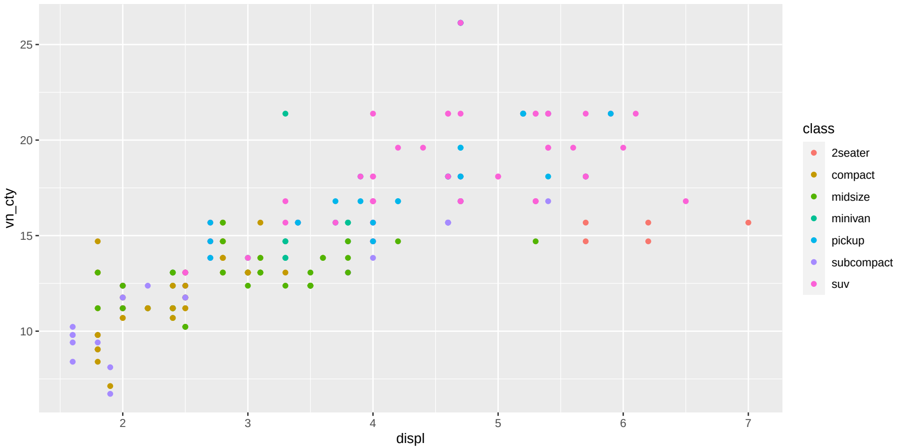
Practice

7. Add *class* variable

Practice

7. Add *class* variable

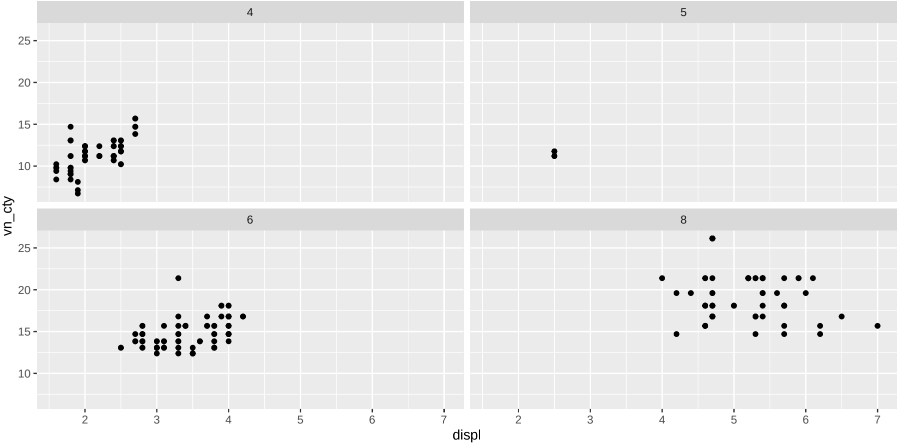
```
dat%>%ggplot(aes(displ, vn_cty))+geom_point(aes(color=class))
```



Practice

8. Fuel economy, engine size, and number of cylinders

```
dat%>%ggplot(aes(displ, vn_cty))+geom_point()+facet_wrap(~cyl)
```



Other geometric objects

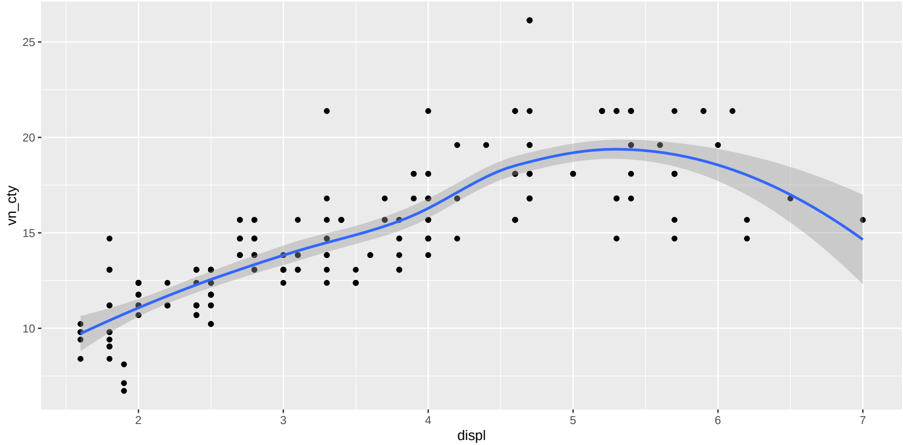
By substituting *geom_point()* for a different geom function, we'll get a different type of plot

- *geom_smooth()* fits a smoother to the data and displays the smooth and its standard error.
- *geom_boxplot()* produces a box-and-whisker plot to summarise the distribution.
- *geom_histogram()* and *geom_freqpoly()* show the distribution of continuous variables
- *geom_bar()* shows the distribution of categorical variables.
- *geom_path()* and *geom_line()* draw lines between the data points that change over time.

Geom_smooth()

Add a smoothed line to the plot to see the dominant pattern.

```
dat%>%ggplot(aes(displ, vn_cty))+geom_point()+geom_smooth()
```



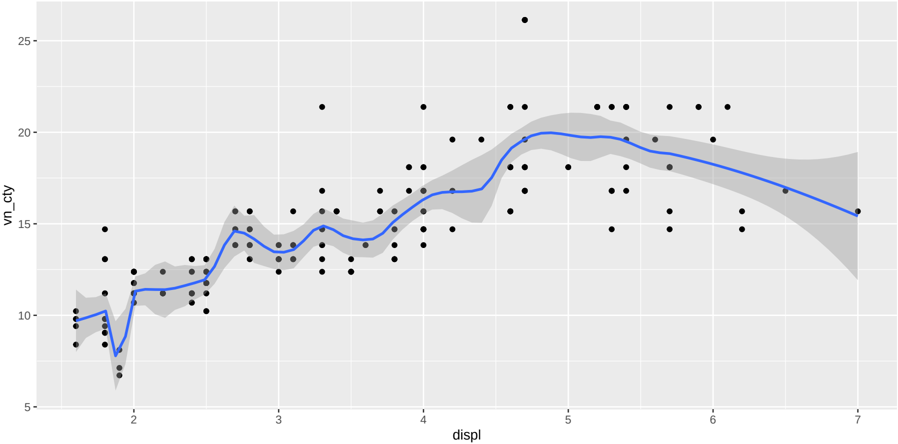
Geom_smooth()

The most important argument to *geom_smooth()* is the *method*, which allows you to choose which type of model is used to fit the smooth curve:

- The default method is "loess" which uses a smooth local regression when the number of data points is less than 1000. The wiggliness of the line is controlled by the *span* parameter, which ranges from 0 to 1.
- In *loess* method, span parameter, often denoted by α, is the proportion of data points are used in each local regression.
- Each subset of the data can be fit by a line or a polynomial of degree 2.

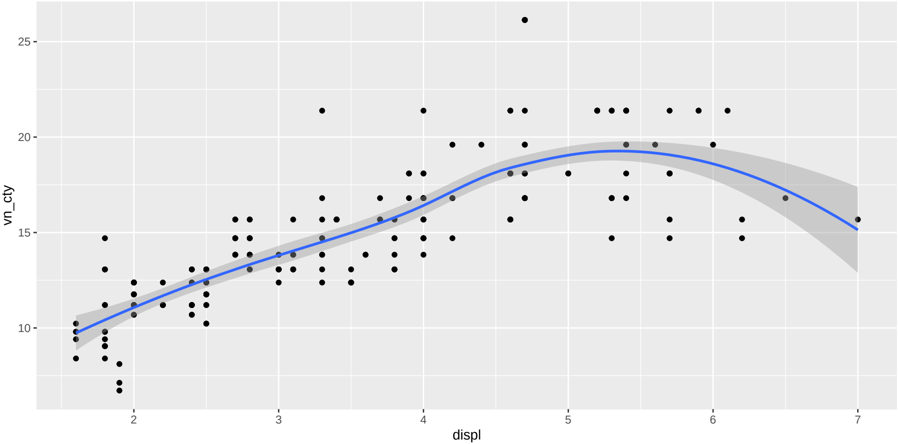
Geom_smooth()

```
dat%>%ggplot(aes(displ, vn_cty))+geom_point()+
  geom_smooth(span=0.2)
```



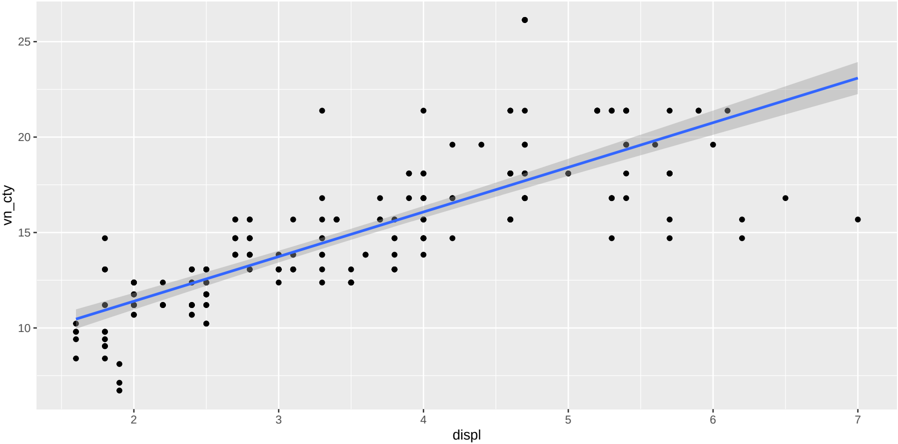
Geom_smooth()

```
dat%>%ggplot(aes(displ, vn_cty))+geom_point()+
  geom_smooth(span=0.8)
```



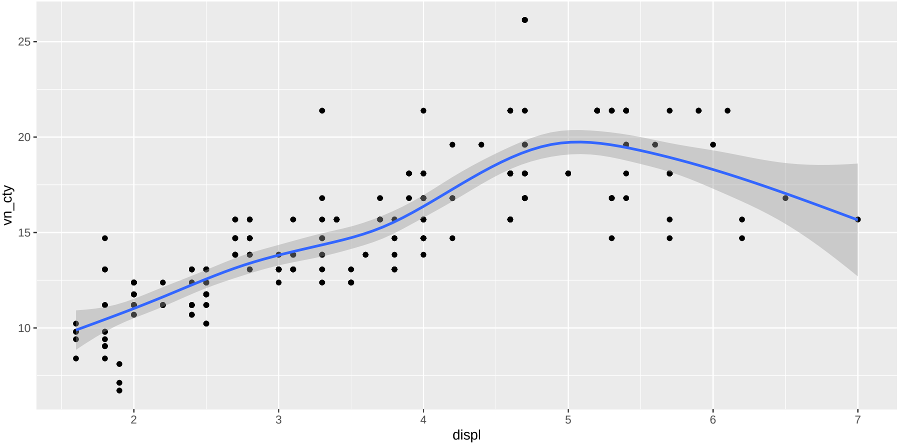
Geom_smooth()

```
dat%>%ggplot(aes(displ, vn_cty))+geom_point()+
  geom_smooth(method="lm")
```



Geom_smooth()

```
dat%>%ggplot(aes(displ, vn_cty))+geom_point()+
  geom_smooth(method="gam")
```

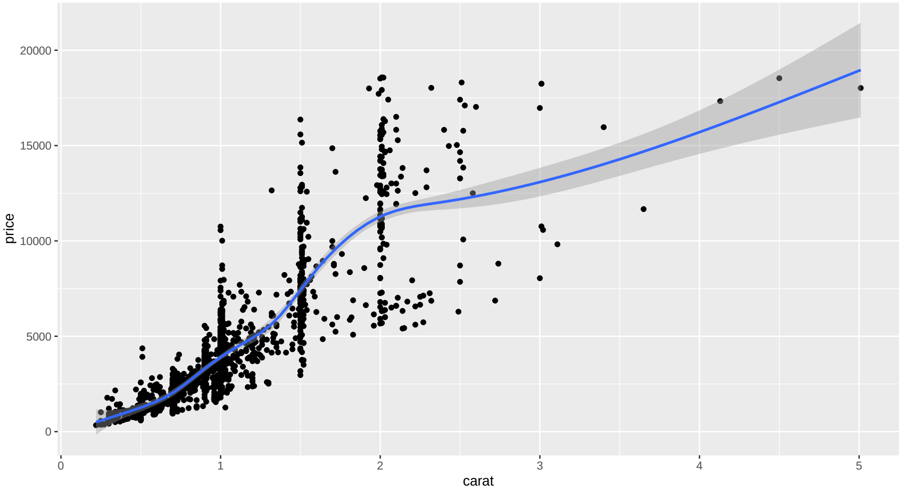


Geom_smooth()

- *loess* gives a better appearance, but is $O(n^2)$ in memory, so does not work for larger datasets.
- GAM, stands for "Generalized Additive Models", is used when when the number of data points is more than 1000.
- When there is only one predictor, GAM is a smoothing spline.

```
dat1<-diamonds%>%filter(cut=="Fair")%>%
  select(price,carat) #1610 data points
dat1%>%ggplot(dat1,aes(carat,price))+
  geom_point()+geom_smooth() # GAM will be used
```

Geom_smooth()



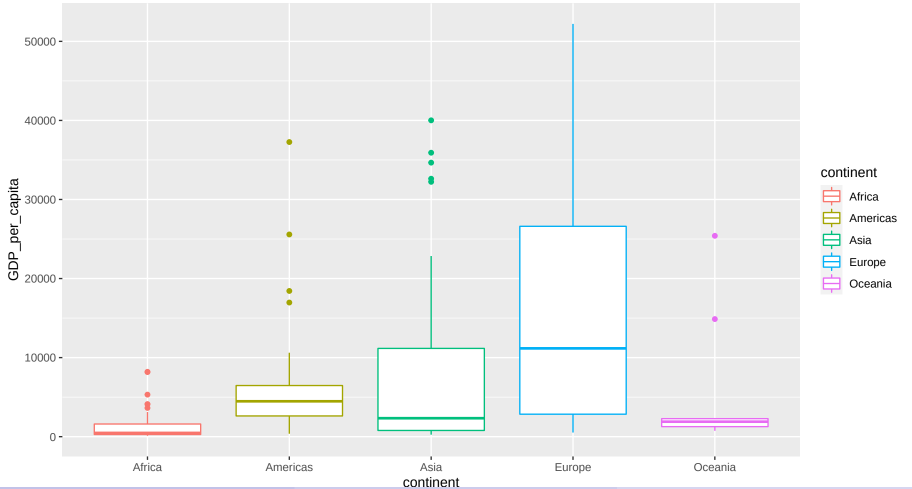
Boxplot

- Boxplot (or box-and-whisker plot) is used to explain how the values of the continuous variables vary with the levels of the categorical variable.
- Boxplot summarise the distribution based on a five-number summary: the minimum, the maximum, the sample median, and the first and third quartiles, while outliers may be plotted as individual points.
- We use *geom_boxplot()* to create a boxplot.
- Beside boxplot, there are two techniques that are useful
  - Jittering plot (*geom_jitter()*).
  - Violin plot (*geom_violin*)

Using boxplot to show how the *GDP_per_capita* variable varies with the levels of *continent* variable?

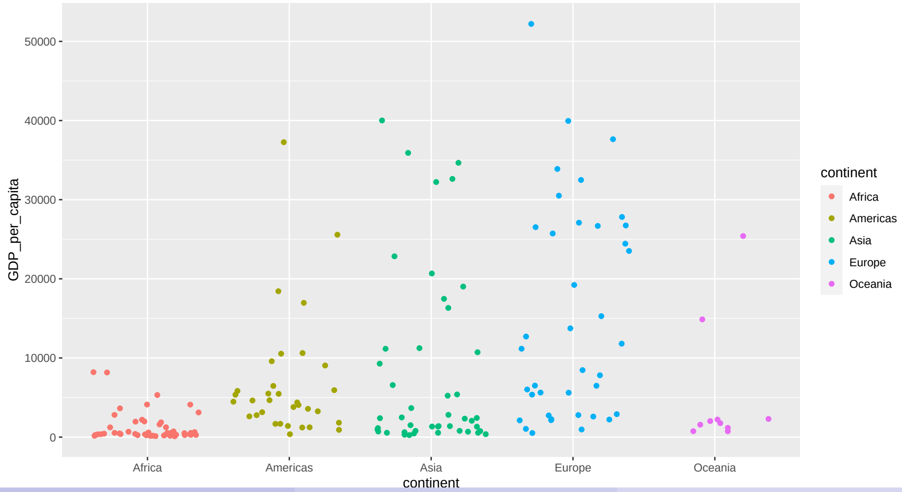
Boxplot

```
dat2010%>%ggplot(aes(continent,GDP_per_capita,color=continent)
geom_boxplot()
```



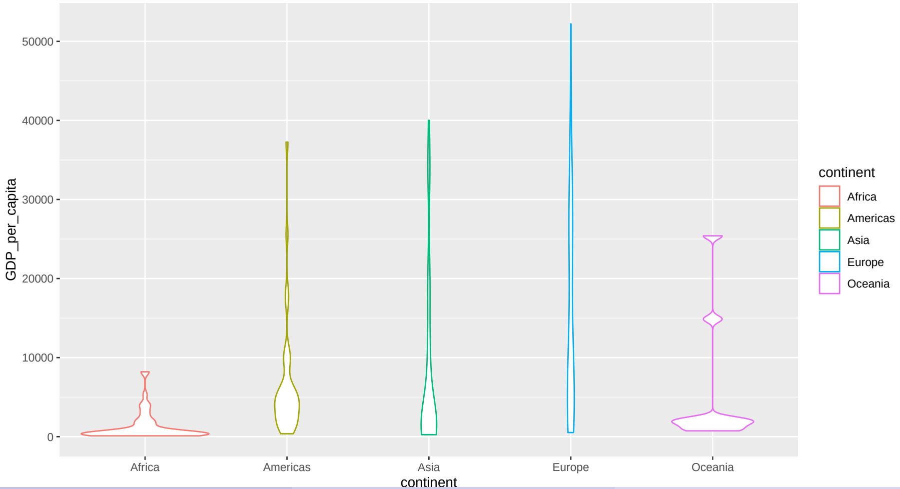
Boxplot

```
dat2010%>%ggplot(aes(continent,GDP_per_capita,color=continent)
geom_jitter()
```



Boxplot

```
dat2010%>%ggplot(aes(continent,GDP_per_capita,color=continent)
geom_violin()
```



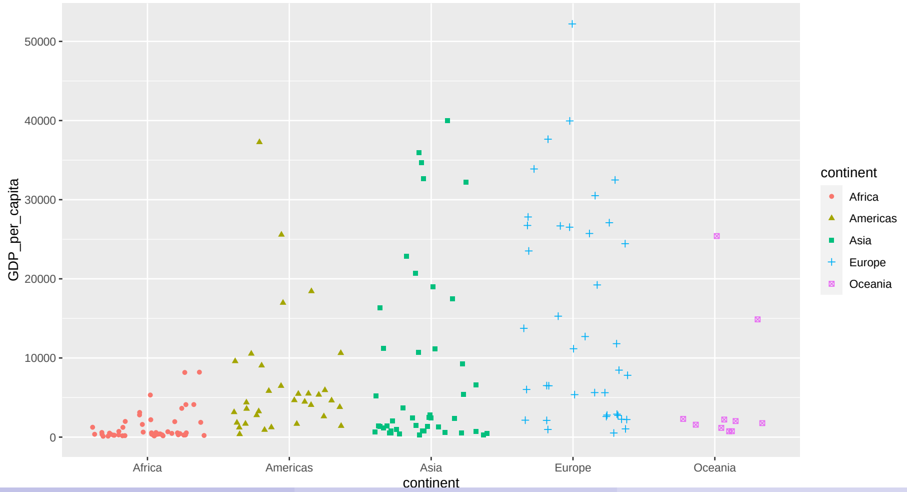
Boxplot

Each method has its strengths and weaknesses.

- Boxplots summarise the distribution with only five numbers.
- Jittered plots show every point but only work with relatively small datasets.
- Violin plots give the richest display but it can be hard to interpret.
- *geom_jitter()* offers the same control over aesthetics as *geom_point()*: size, colour, and shape
- We can control the outline colour and the internal fill colour of *geom_boxplot()* and *geom_violin()*

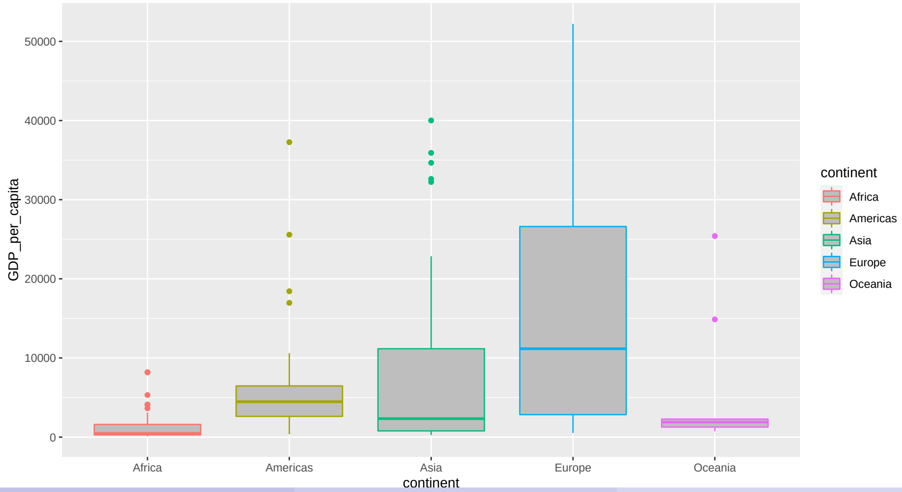
Boxplot

```
dat2010%>%ggplot(aes(continent,GDP_per_capita,color=continent)
geom_jitter(aes(shape = continent))
```



Boxplot

```
dat2010%>%ggplot(aes(continent,GDP_per_capita,color=continent)
geom_boxplot(fill = "gray")
```

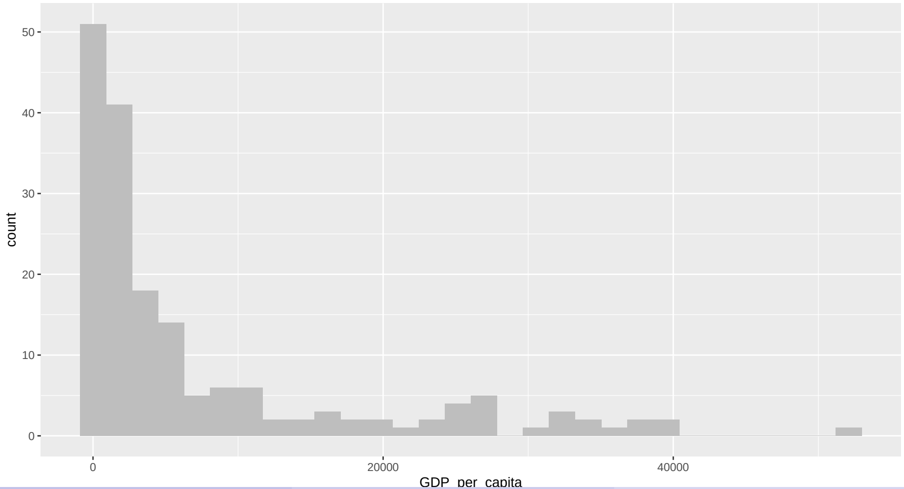


Histograms and frequency

- Histograms and frequency polygons show the distribution of a single numeric variable
- Both histograms and frequency polygons work in the same way:
  - They bin the data.
  - Then count the number of observations in each bin
  - Histograms (*geom_histogram()*) use bars
  - Frequency (*geom_freqpoly()*) polygons use lines.
- We can control the width of the bins with the binwidth argument (the default splits the data into 30 bins)

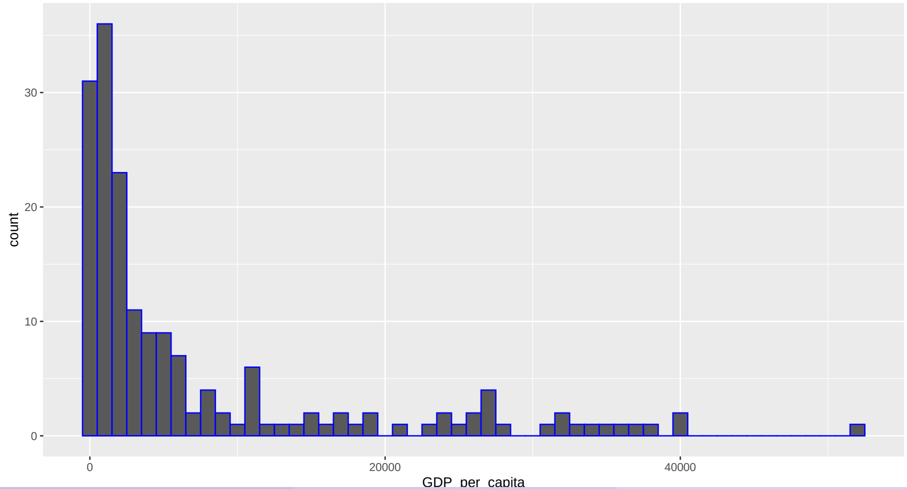
Histograms and frequency

```
dat2010%>%ggplot(aes(GDP_per_capita))+
  geom_histogram(fill = "gray")# 30 bins
```



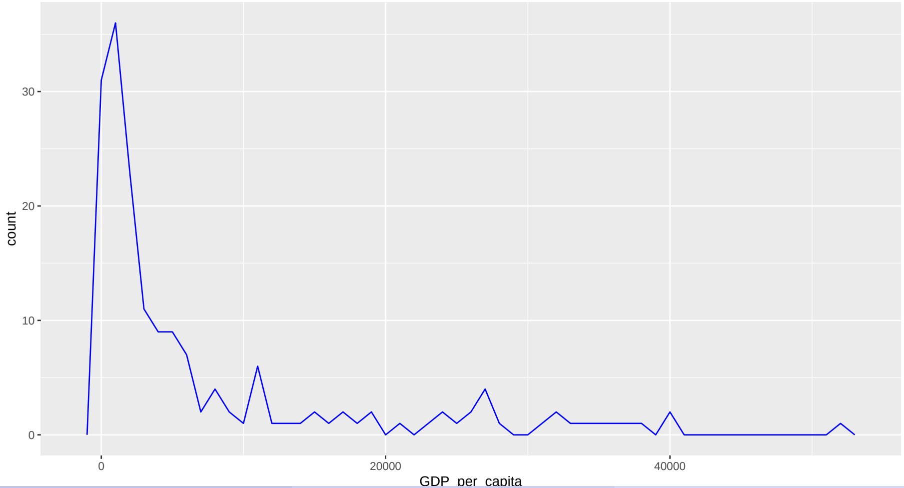
Histograms and frequency

```
dat2010%>%ggplot(aes(GDP_per_capita))+
  geom_histogram(color = "blue",binwidth = 1000)
```



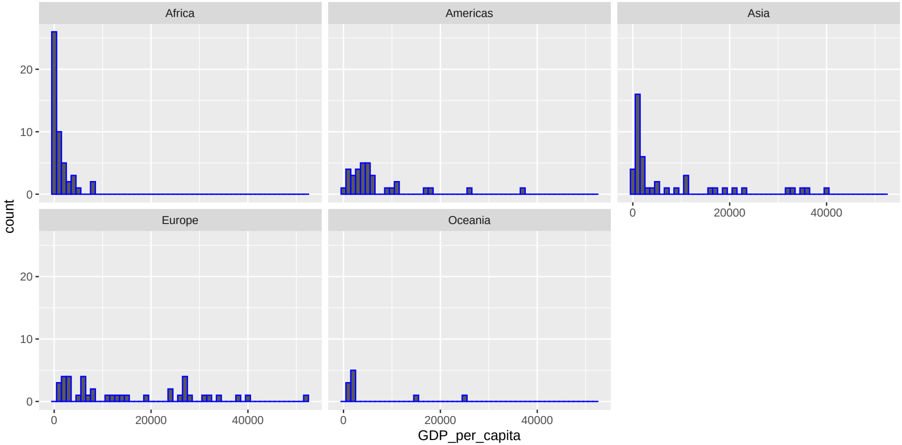
Histograms and frequency

```
dat2010%>%ggplot(aes(GDP_per_capita))+
  geom_freqpoly(color = "blue",binwidth = 1000)
```



Histograms and frequency

```
dat2010%>%ggplot(aes(GDP_per_capita))+
  geom_histogram(color = "blue",binwidth = 1000)+
  facet_wrap(~continent)
```



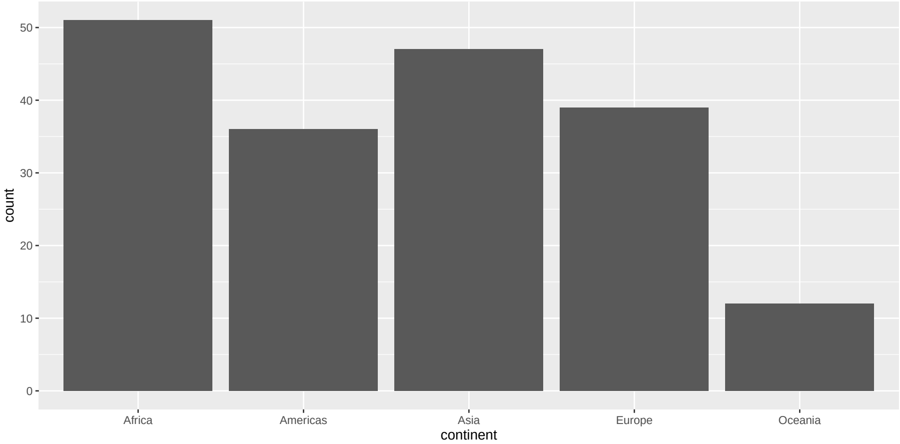
Bar charts

- Histograms summarise continuous data and bar charts summarise discrete data.
- By default, bar charts is used to visualized unsummarised data but it is also used for presummarised data

```
dat2010%>%ggplot(aes(continent))+geom_bar()
```

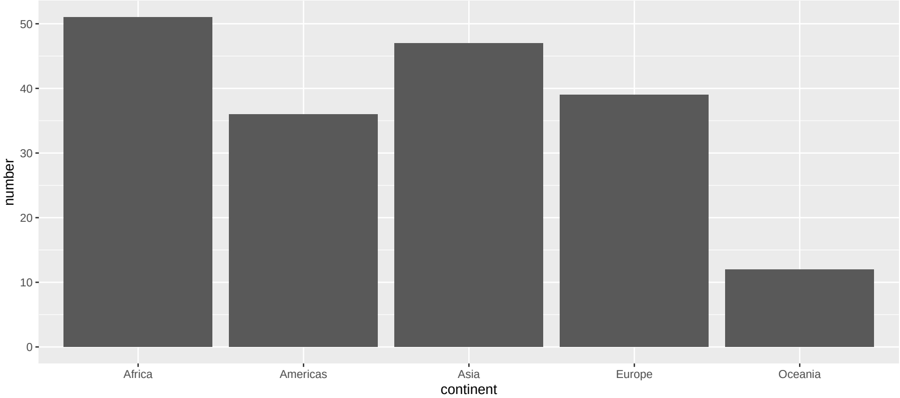
Bar charts

```
dat2010%>%ggplot(aes(continent))+geom_bar()
```



Bar charts

```
dat2010%>%group_by(continent)%>% #summarised data
  summarise(number=length(continent))%>%
  ggplot(aes(continent,number))+geom_bar(stat="identity")
```



Line and path plots

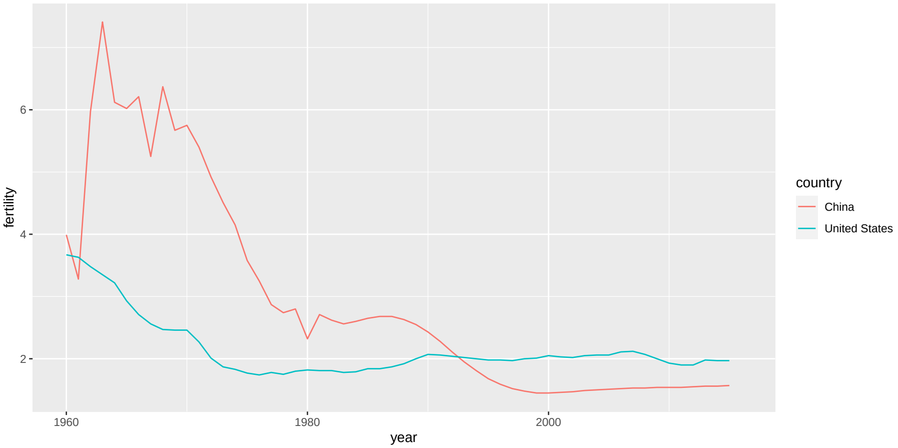
- Line and path plots are used for time series data.
- Line plots join the points from left to right. They often have time on the x-axis, showing how a single variable has changed over time.
- Path plots join them in the order that they appear in the dataset. Path plots could show how two variables have simultaneously changed over time.

We create the following time series

```
datUsCh<-filter(gapminder,country %in% c("United States","Chi
  select(year,country,fertility,population)
```

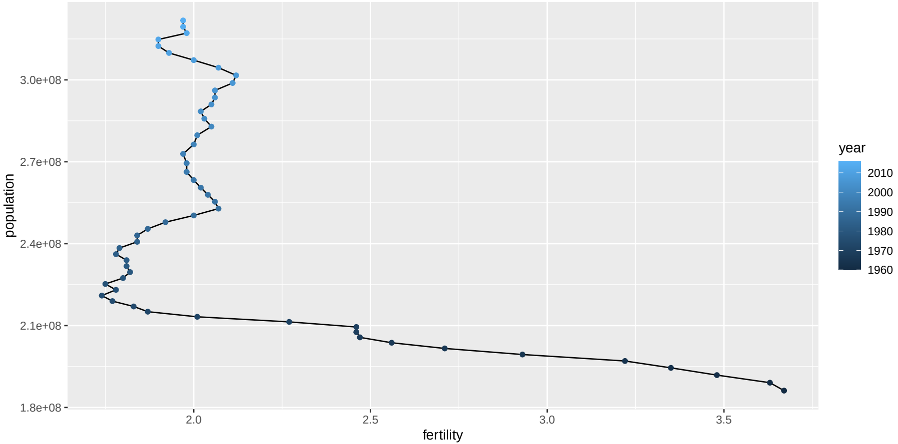
Line and path plots

```
datUsCh%>%ggplot(aes(year,fertility))+
  geom_line(aes(color=country))
```



Line and path plots

```
datUsCh%>%filter(country == "United States")%>%
  ggplot(aes(fertility,population))+geom_path()+
  geom_point(aes(color = year))
```



The Axes

1. *xlab()* and *ylab()* modify the x-axis and y-axis label

```
dat2010%>%ggplot(aes(fertility,GDP_per_capita))+
  geom_point(aes(color=continent))+
  xlab("Average number of children per woman") +
  ylab("GDP per capita in USD")
```

2. *xlim()* and *ylim()* modify the limits of axes

```
dat2010%>%
  ggplot(aes(continent,GDP_per_capita,color=continent))+
  geom_boxplot()+
  xlim("Africa","Americas","Asia","Oceania")+
  ylim(0,40000)
```

ggplot2 output

1. you can save a plot to a variable and manipulate it:

```
p<-dat2010%>%ggplot(aes(fertility,GDP_per_capita))+
  geom_point(aes(color=continent))+
  xlab("Average number of children per woman") +
  ylab("GDP per capita in USD")
p+geom_smooth()
```

2. *ggsave()* function saves plot to disk

```
setwd("C:/Users/AD/Desktop")
ggsave("myplot.png",width = 20, height = 20, units = "cm")
# we can choose .png or .pdf extension
```

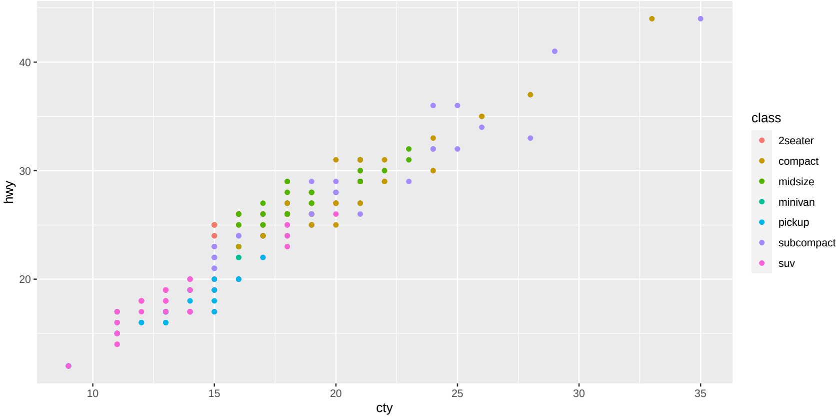
Practice

We back to **mpg** and **diamond** dataset.

1. What's the problem when creating this plot *ggplot(mpg, aes(cty, hwy))+geom_point()*? Is there other geom which is more effective at this problem?
2. *ggplot(mpg, aes(class, hwy)) + geom_boxplot()* creates a plot where the ordering of class on x-axis is alphabetical which is not useful. How could you change the order to be more informative?
3. What does function *reorder()* do in *ggplot(mpg,aes(reorder(class, hwy), hwy))+geom_boxplot()*? Read the documetation.
4. List out several ways to visualise a 2d categorical distribution. Try them out by visualising the distribution of (*trans* and *class*) and (*cyl* and *trans*).

Practice

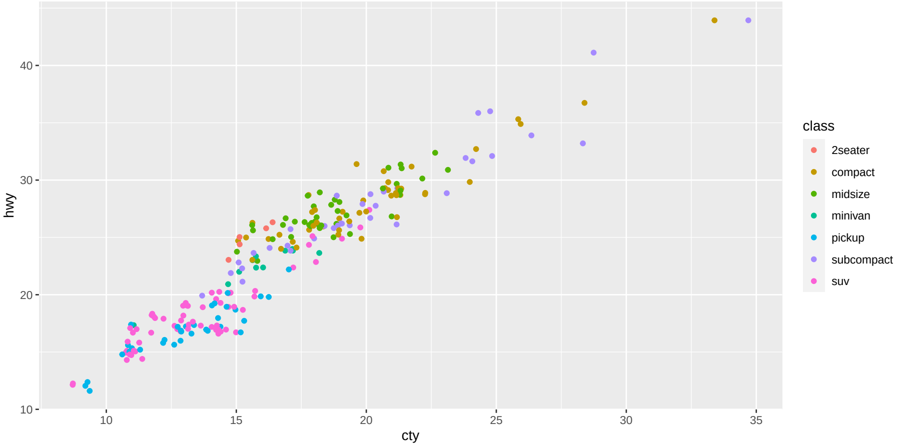
1. There is overplotting, graph doesn't show all the availble data points in the dataset.



Practice

1. Using jittering plots to avoid overplotting

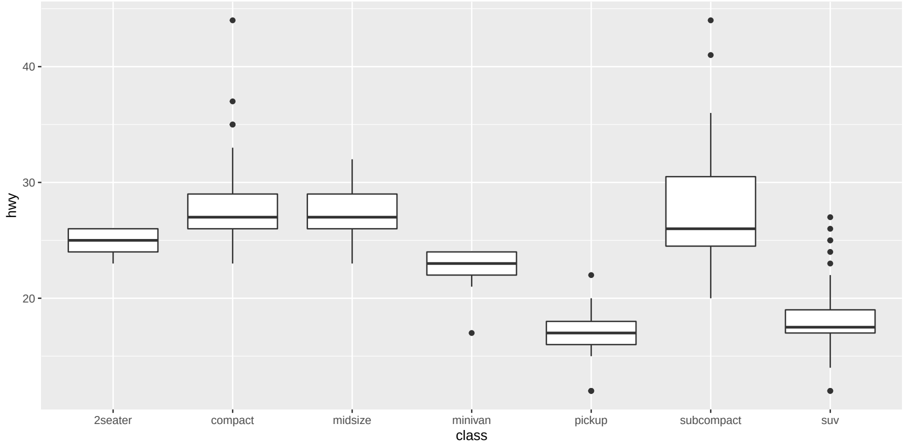
```
ggplot(mpg, aes(cty, hwy))+geom_jitter(aes(color=class))
```



Practice

2. The order of class on x-axis is alphabetical

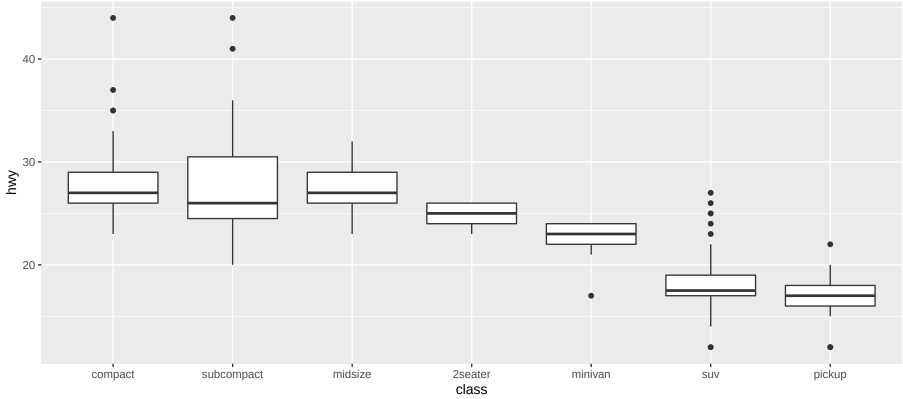
```
ggplot(mpg, aes(class, hwy)) + geom_boxplot()
```



Practice

2. The order of class on x-axis is alphabetical

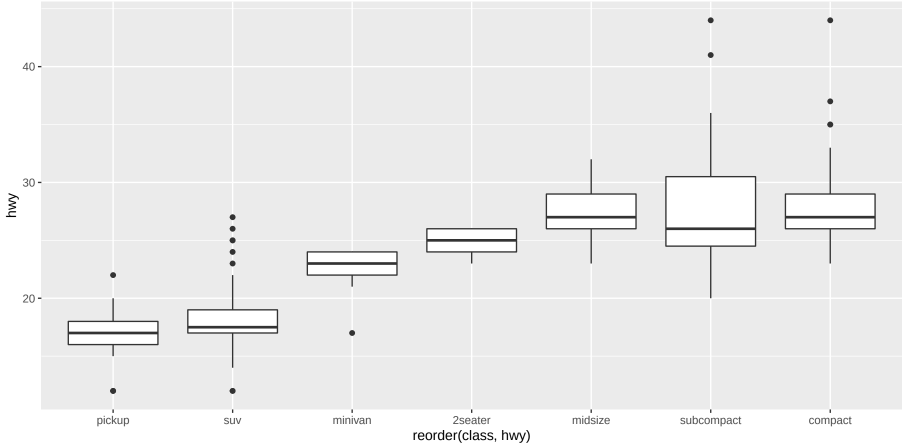
```
ggplot(mpg, aes(class, hwy)) + geom_boxplot() +xlim("compact",
```



Practice

3.

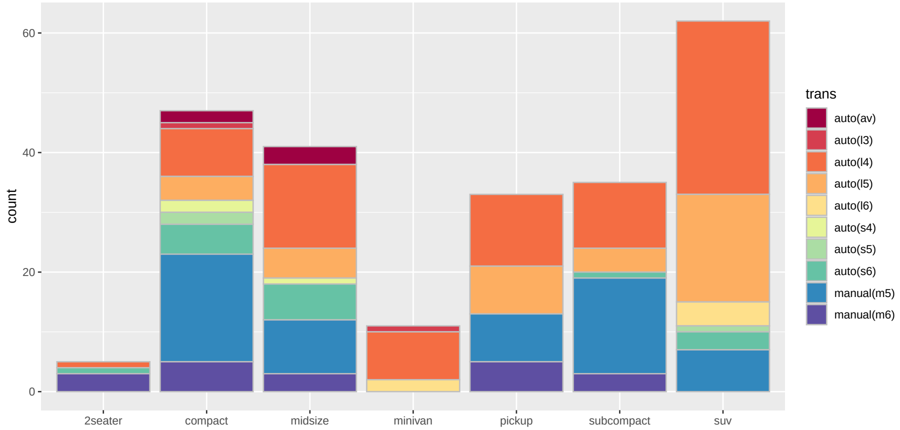
```
ggplot(mpg,aes(reorder(class, hwy), hwy))+geom_boxplot()
```



Practice

4. Visualise *trans* and *class* using *geom_bar()*

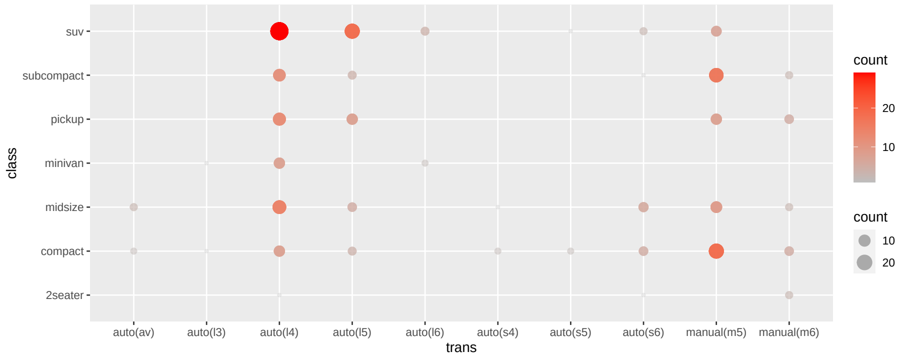
```
ggplot(mpg,aes(class))+geom_bar(aes(fill=trans),color="gray")+
  scale_fill_brewer(palette = "Spectral")
```



Practice

4. Visualise *trans* and *class* using *geom_point()*

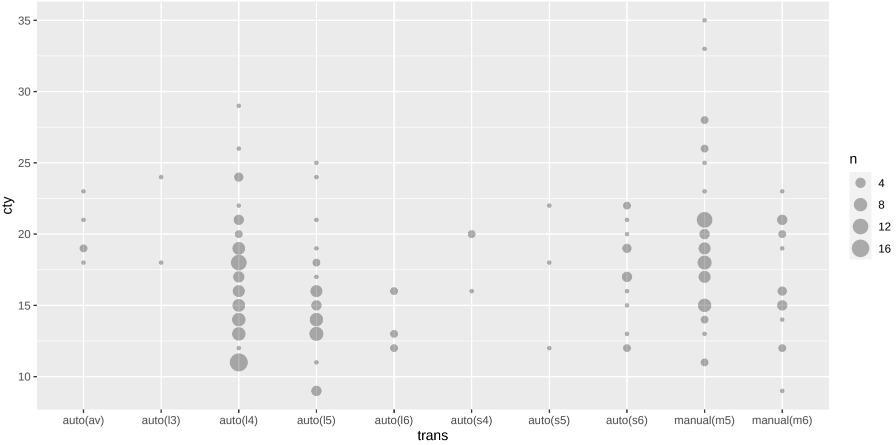
```
mpg%>%group_by(trans,class)%>%mutate(count=length(model))%>%
  ungroup%>%ggplot(aes(trans,class))+
  geom_point(aes(color=count,size=count),alpha=0.3)+
  scale_colour_gradient(low="gray",high="red")
```



Practice

4. Visualise *trans* and *cty* using *geom_count()*

```
mpg%>%ggplot(aes(trans,cty))+geom_count(alpha=0.3)
```



Practice

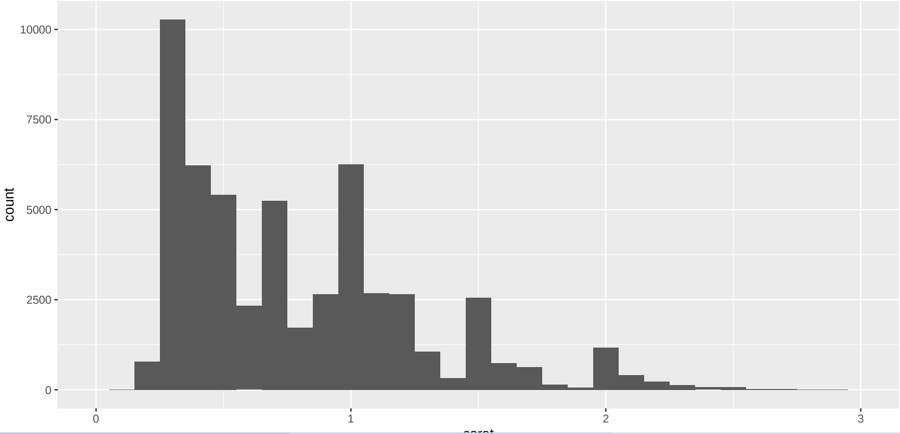
Read the documentation for **diamond** data

5. Explore the distribution of the carat variable in the diamonds dataset. What binwidth reveals the most interesting patterns?
6. Explore the distribution of the carat variable varied by cut ?
7. Explore the distribution of the price variable in the diamonds data. How does the distribution varied by cut?
8. Visualize the relation between price and carat using *geom_point()*. What does parameter *alpha* do in *geom_point()*?

Practice

5. Distribution of the carat variable in the diamonds dataset.

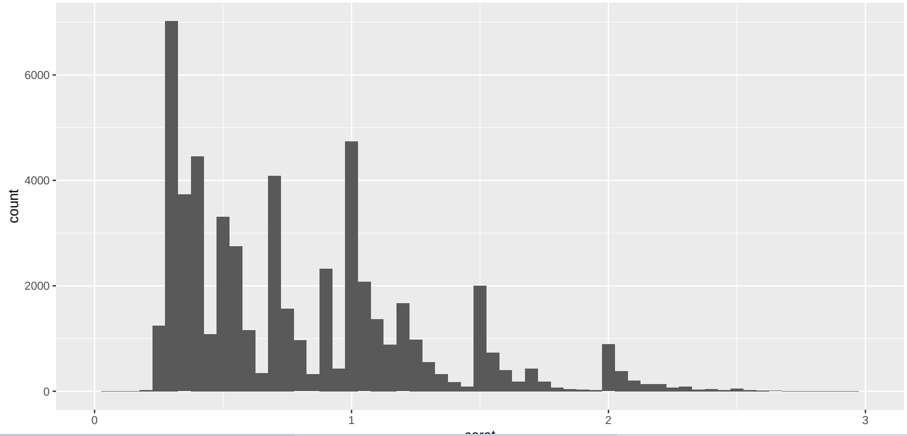
```
diamonds%>%ggplot(aes(carat))+geom_histogram(binwidth=0.1)+
  xlim(0,3)
```



Practice

5. Distribution of the *carat* variable in the *diamonds* dataset.

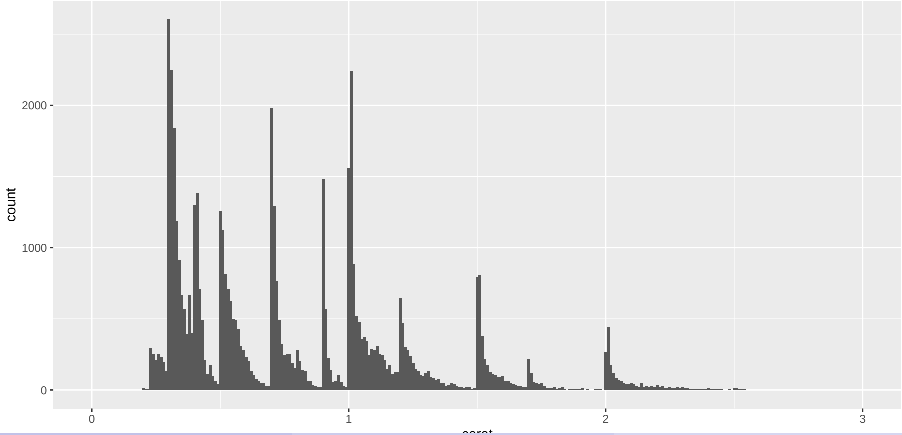
```
diamonds%>%ggplot(aes(carat))+geom_histogram(binwidth=0.05)+
  xlim(0,3)
```



Practice

5. Distribution of the *carat* variable in the diamonds dataset.

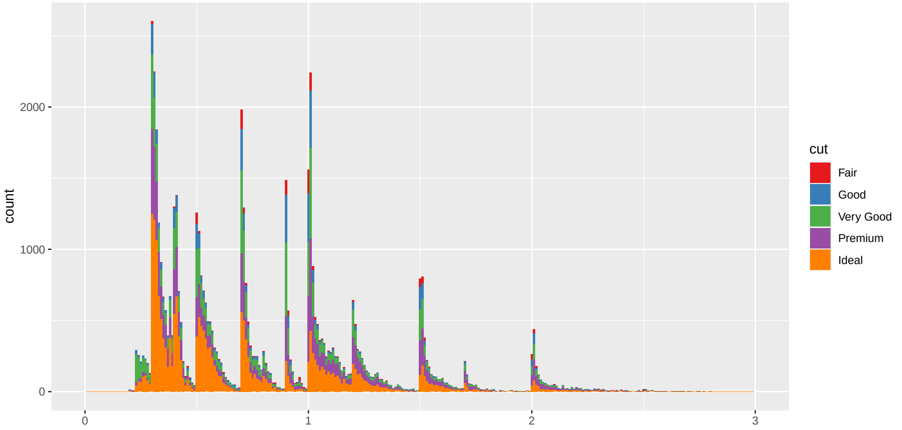
```
diamonds%>%ggplot(aes(carat))+geom_histogram(binwidth=0.01)+
  xlim(0,3)
```



Practice

6. Distribution of the *carat* varied by *cut*.

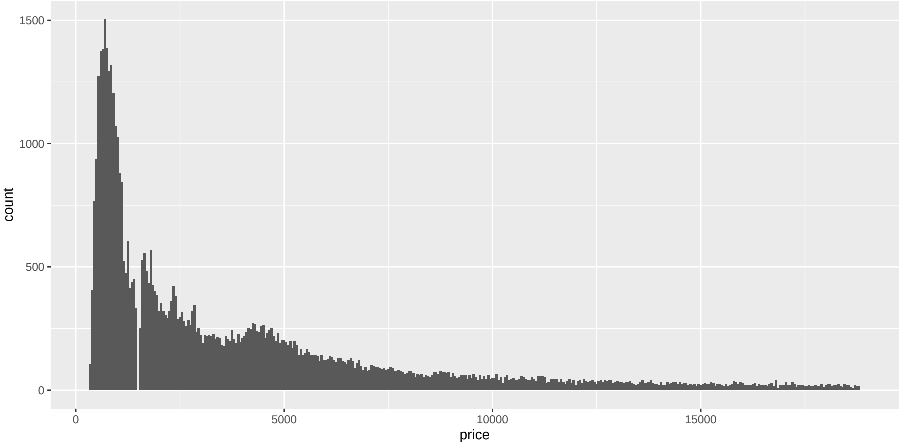
```
diamonds%>%ggplot(aes(carat,fill=cut))+geom_histogram(binwidth
  xlim(0,3)+scale_fill_brewer(palette = "Set1")
```



Practice

7. Distribution of the *price* variable in the diamonds dataset.

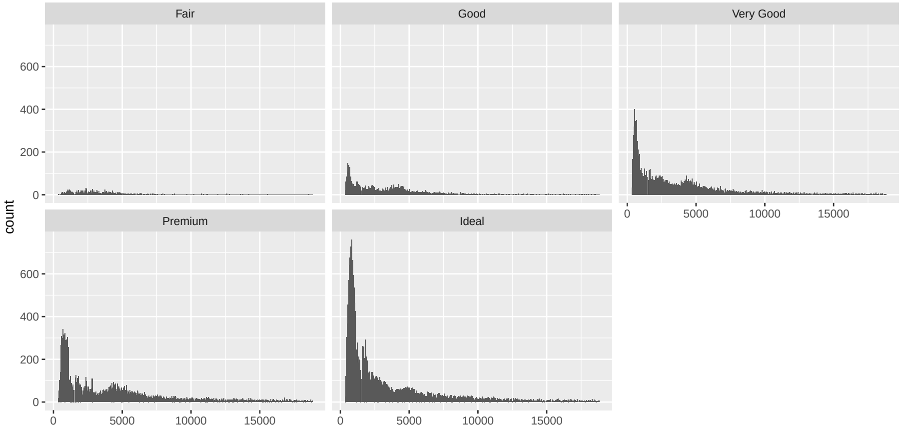
```
diamonds%>%ggplot(aes(price))+geom_histogram(binwidth=50)
```



Practice

7. Distribution of the *price* variable varied by cut.

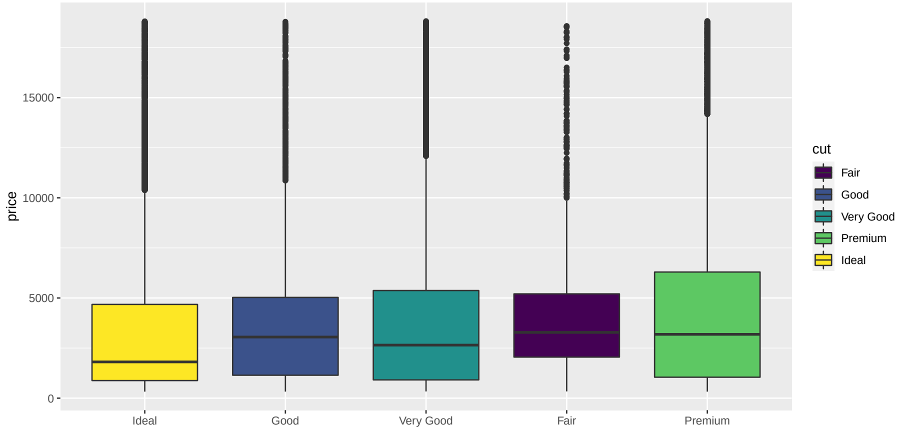
```
diamonds%>%ggplot(aes(price))+geom_histogram(binwidth=50)+
  facet_wrap(~cut)
```



Practice

7. Distribution of the *price* variable varied by cut.

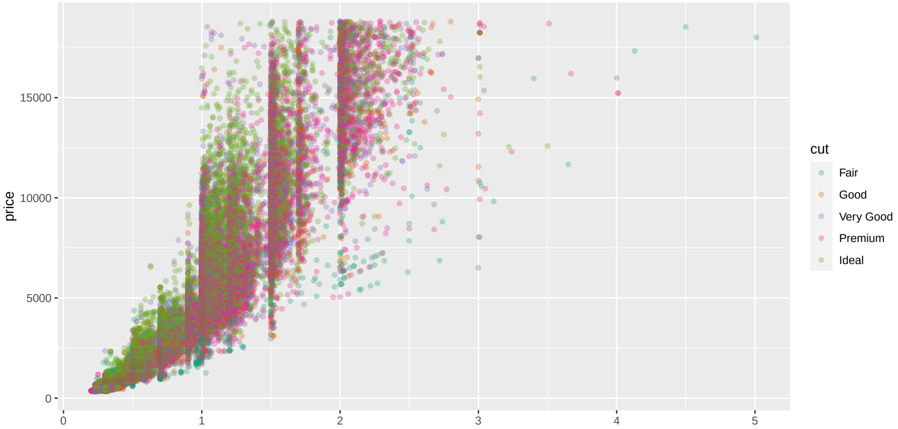
```
diamonds%>%ggplot(aes(reorder(cut, price),price,fill=cut))+
  geom_boxplot()
```



Practice

8. The relation between price and carat

```
diamonds%>%ggplot(aes(carat,price,color=cut))+
  geom_point(alpha=0.3)+scale_color_brewer(palette = "Dark2")
```



Practice

End of Section 2