Chapter Sixteen: Data Visualisation

Simona Simona

4/12/2020

In this chapter, we shall looks the following topics:
- Bar chart
- Histogram
- Box plots
- Scatter plots
- Violin plots

Data vusualisation in one of the most important inventions of the 21th century in statistical presentations. Following the saying that pictures speak a thousand words, there has been significant developments in data visualisations in the past few decade. The R programming language has contributed a great deal to this development. It is flexible enough to permit a wide range of shapes and colours. It provides an easy way of getting a feel about the graph you want to create with a few lines of code. But also because of it's extensibility nature, R allows for an integration with more sophisticated packages to be used to produce publication-worth visualisations. The past decade has seen enormous improvements that allow for interative online aplications and maps to be created in RStudio through through packages like RShiny.

In this chapter, we shall explore a cocktail of visualisation types that are common in social science research. We shall also provide a snnipet into more sophisticated visualisations like correlation plots, regression plots, diverging bars and maps. Its important to remember that data visualisation is a field on it's own and it is not possible to exhaust all the possible graphics you may want to do for your research. You may want to look at Thomas Rahlf's *Data visualisation with R* and Hadley Wickham's *Elegant Graphics for Data Analysis* for a much more detailed treatment of the subject.

R comes with graphics capabilities with relatively easy to write codes but the graphs may not be as fancy as those that are produced in the ggplot2 package. Hardley Wickarm has has made a very signficant contribution to R by producing the whole tidyverse echo system of R packages that are very sophiticated but also intuitive and relatively easier to write. I will show you how to produce a few common graphs using base R before I introduce the *ggplot2* package that we shall use throughout the chapter. I often times find myself relying on base R functionality just to have an impression of the graph before creating it in ggplot2. There are a few more graphics packages that we will not use in this chapter.

As usual, we start by reading in the data. We are going to use the Titanic dataset. I guessing I don't really to explain the story of the Titanic. I have never needed to explain anywhere because most people have watched the movie *Titanic* so it needs less introduction. But in case there are some people who live under the rock, the Titanic was a British passenger cruise liner that sunk in the North Atlantic Ocean in 1912 after hitting on an ice berg on her first sea voyage from Southampton UK. The data we are using is about passengers on the ship. The data is in .csv format. First we need the set the working directory. Again, I will keep on repeating this, make sure that the name of the folder which contains your data file is the last item in the file path. In my case 'Data' is the file.

```
setwd("~/Documents/Publications/Books/Quantitative/Data")
```

Now that you have set the working directory, you can read in your data. This time you don't need to provide the path because R now knows where your data is.

```
titanic <- read.csv("titanic.csv")
```

After reading in the data, we can use the names() function to check the variable names of the data:

```
names(titanic)
```

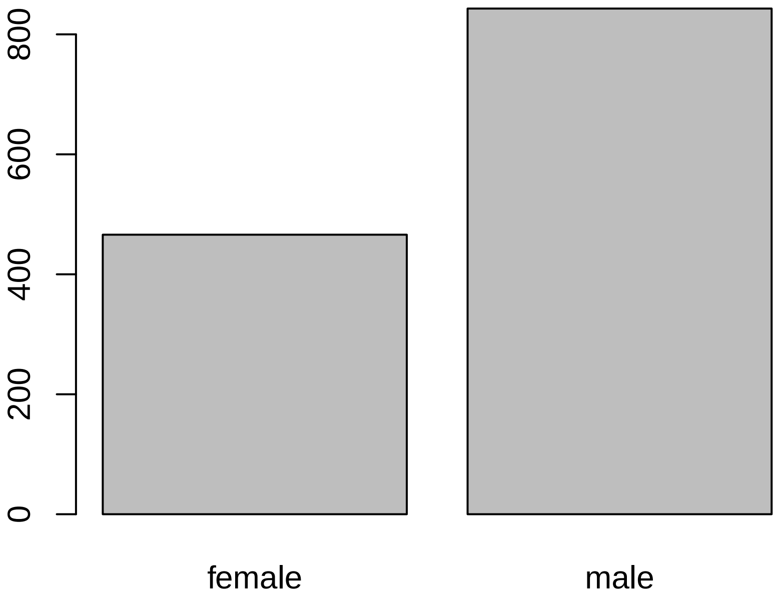
```
##  [1] "pclass"    "survived"  "name"      "sex"       "age"       "sibsp"
##  [7] "parch"     "ticket"    "fare"      "cabin"     "embarked"  "boat"
## [13] "body"      "home.dest"
```

We can see now that we have 14 variables in our dataset and if we want to see the spreadsheet of the data, we can use the View() function. We have already explained these variables in details in the previous chapters. Here, we will jump straight into our graphics. I will not analyse the graphs just yet as I will do that when we move to the ggplot2 package.

Bar chart

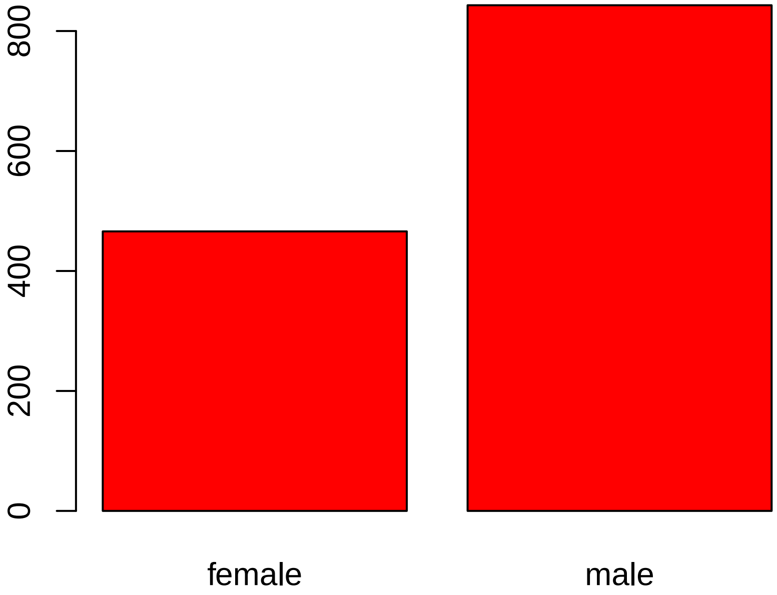
To get the barchar, first create an object where you put the distribution of the variable you want to plot and you use the table() function for the distribution. In my case, the object is tab and I am putting the variable *sex* into it. Of course, you haven't forgotten that everytime you call the variable, you first write the name of the data followed by a dollar sign. My data frame object is titanic. After that, I use the barplot() function with my object tab as the argument to create the graph.

```
tab <- table(titanic$sex)
barplot(tab)
```



We can see that the variable sex has two categories (Male and Female) and are represented by the bars accordingly. I can play around with this graph in terms of providing different calours for my bars. If for example, I want to use the colour res, I simply supply the colour name to the barplot function as an argument: col = 'red'. If you use the full word colour, it will not work here!

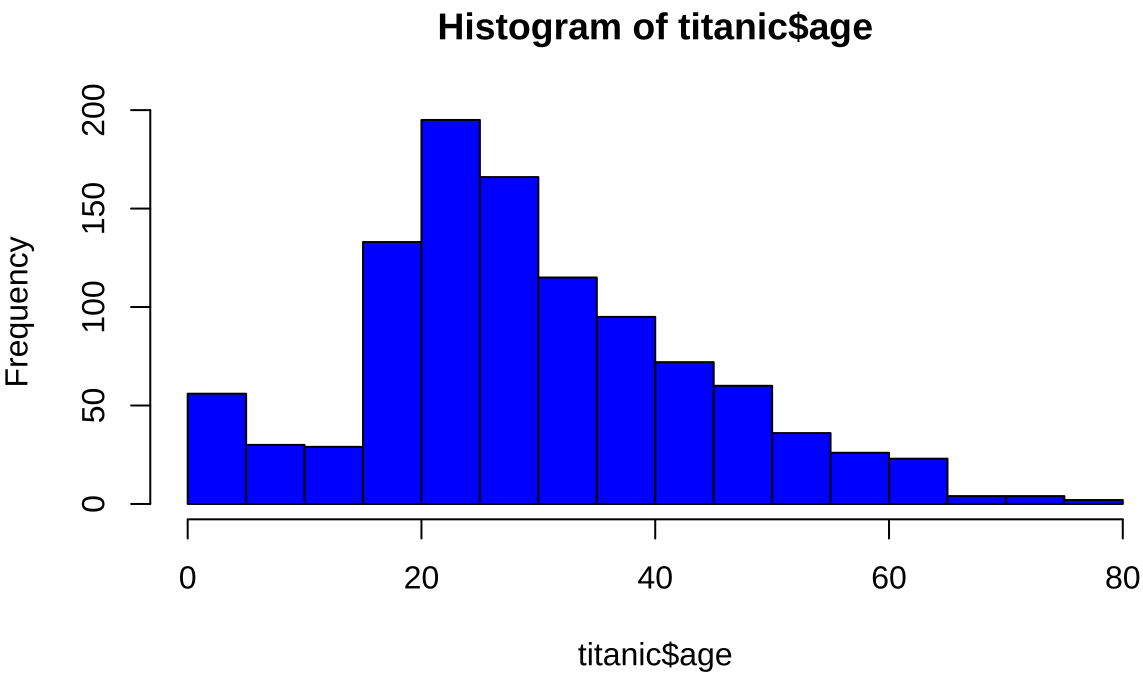
```r
barplot(tab, col = "red")
```



Histogram

You can plot a histogram directly by using the hist() function and again, if you want to change the colour you can supply the col argument just like we did in the barplot() function. As we shall explain shortly, a histogram is used on interval/ratio variables and in our case we shall use age.

```r
hist(titanic$age, col = "blue")
```

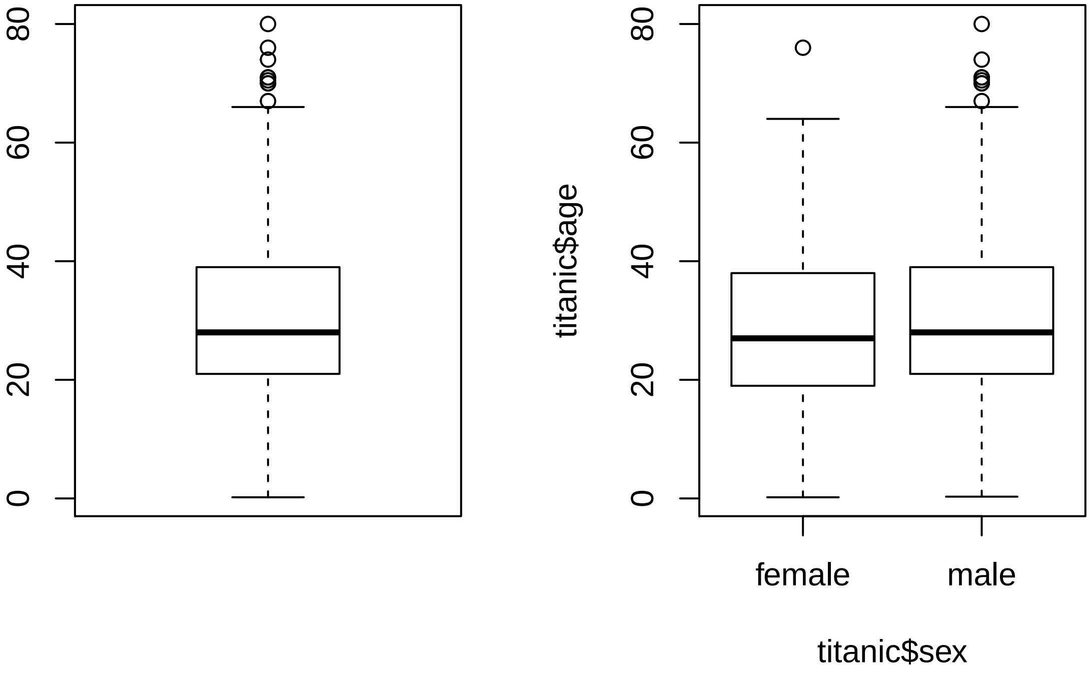


Box plot

For the box plot, we need to use either one variabe, an interval/ratio one or two variables: an interval/ratio and a categorical variable. We shall plot them side by side here using age and sex:

```
## named list()
```

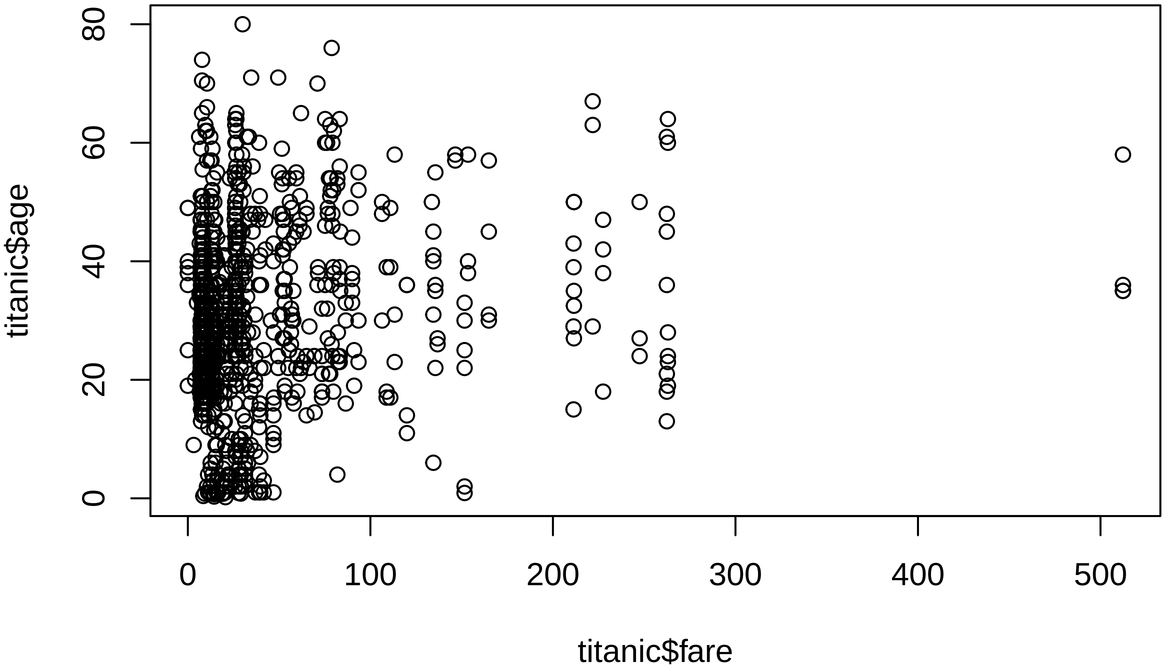
```r
boxplot(titanic$age) # box plot of age
boxplot(titanic$age ~ titanic$sex) # boxplot of age by sex
```



Scatter plot

For a scatter plot, we use two interval/ratio variables. In this case we use age and fare. To see if there is a likely relationship between age and the amount of money paid (in British pounds). The plot() function with two variables specified, separated by the symbol tilde as arguments will produce a scatter plot for us:

```
plot(titanic$age ~ titanic$fare)
```



It's kind of difficult to see what is going on there but that's none of business for now. What matters is that we have ploted the graph successfully.

These are the most basic graphs you are likely to encounter as you begin your journey in quantitative social science or data science. Let's now turn to the ggplot2 package, which you are going to use if you are considering to take data graphics seriously. We will start by discussing what makes ggplot2 unique and how it differs from other graphics packages then we dive into grapphing different types of graphics. Here we are not only going to plot data on graphs but also take time to explain the results of our graphics.

ggplot2 package: The Basics

The difference between the plots that we have just done in base R and those produced by the ggplot2 package is not only with the elegancy of graphics but also the approach to building the graphics. ggplot2 is based on the layered grammar of graphics. What that means is that any graphics is not viewed as a continuous whole but a combination of discrete elements that can be arranged in many different ways. The layered grammar of graphics provides an underlying theory behind graphics so that when you understand it, you will be able to build any graph that is appropriate for your data.

We are not going to bother with the theory here because it sounds very complicated when the practical side is pretty straightforward. What is important is to know that a ggplot2 graph is composed of layers which we build one on top of the other to create the graph we want. At the base of a ggplot2 graph is the **data** and the **aestheic mappings** which are basically how your variables are mapped on the x and y coordinates. This is where you specify which of your variables should be on the x-axis and which one should be on the y-axis. If you are ploting one variable, you only use the x-axis. The base of the graph is included in the ggplot() function[1] Geometric objects (**geoms**) begin the the component of layers and are added with the plus sign. These are basically the visual representations of the plot. In other words, the type of plot you want to visualise. These are designed in terms of **geom_**and what follows the underscore is the specific geometric type you want to produce. As we shall see, there are many different types of geoms. Different agurments can be included in the geom_x() function to adjust the default specifications of the function.

There are may other layers that can as we shall see below but we can begin with this basic understanding. I have arranged the remaining part of the chapter in terms of the number of variables involved in the analysis begining with univariate visualisations. I will be explaining the differences in terms of type of variables and the interpretations appropriate for the variable(s) in question.

Univariate data visualisation

We must know by now that uni means one. Univariate data visualisation is the type of visualisation that is based on a single variable. But varibles are themselves differently measured and every level of measurement has appropriate types of visualisation commerate to it. It is very important to wrap your head around this so that you don't apply graphics on 'wrong' variables. You will commonly use barcharts on categorical variables[2] and histogram and box plots on interval/ratio variables. We will discuss these three under this section

Barplot

Barplots and used to visualise the distribution of categorical variables. Iam assuming you still have the titanic data loaded so we will go straight and build the graph using the ggplot() function and then we will explain what is happening afterwards. We need to load the package first for us to access the the function. The ggplot2 package is loaded like any other package we have used so far in the book using the library() function and if it is not in your computer already, go ahead and install it as per instructions you are already familiar with. However, my advise is that you load the package as part of the tidyverse echo system which allows you access to all tidyverse packages. You do this by loading the tidyverse package instead and you can use the ggplot2 package perfectly well after that.

```r
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------ tidyverse 1.3.0 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.4
## v tidyr   1.0.2     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
```

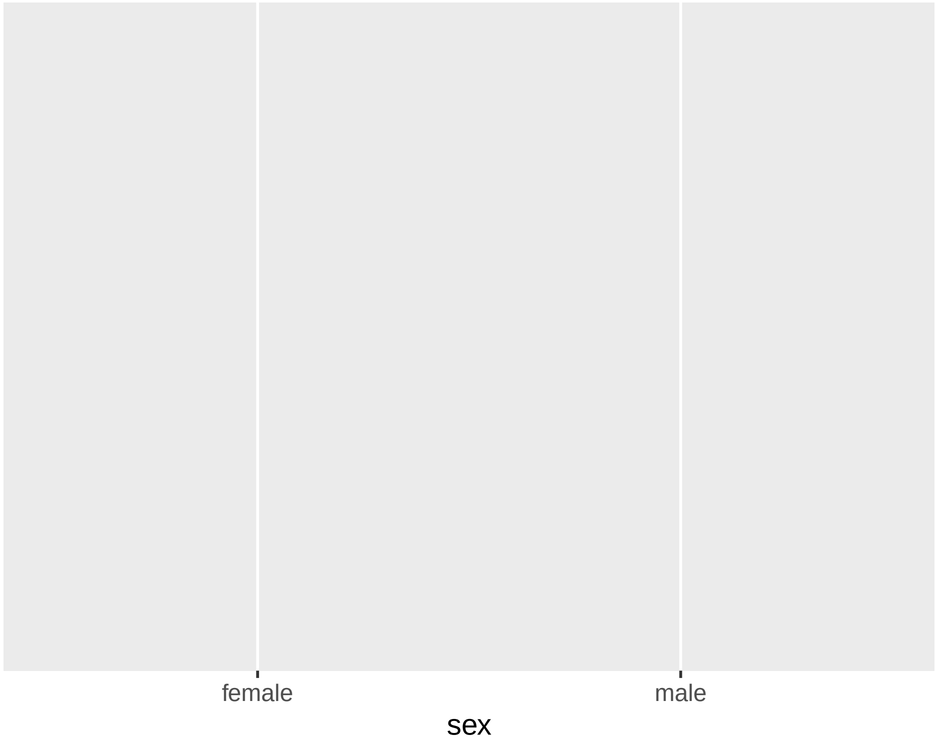
```
## -- Conflicts --------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
```

[1] Not to be confused with the ggplot2 package from which the function comes

[2] Categorical variables are nominal and ordinal level variables

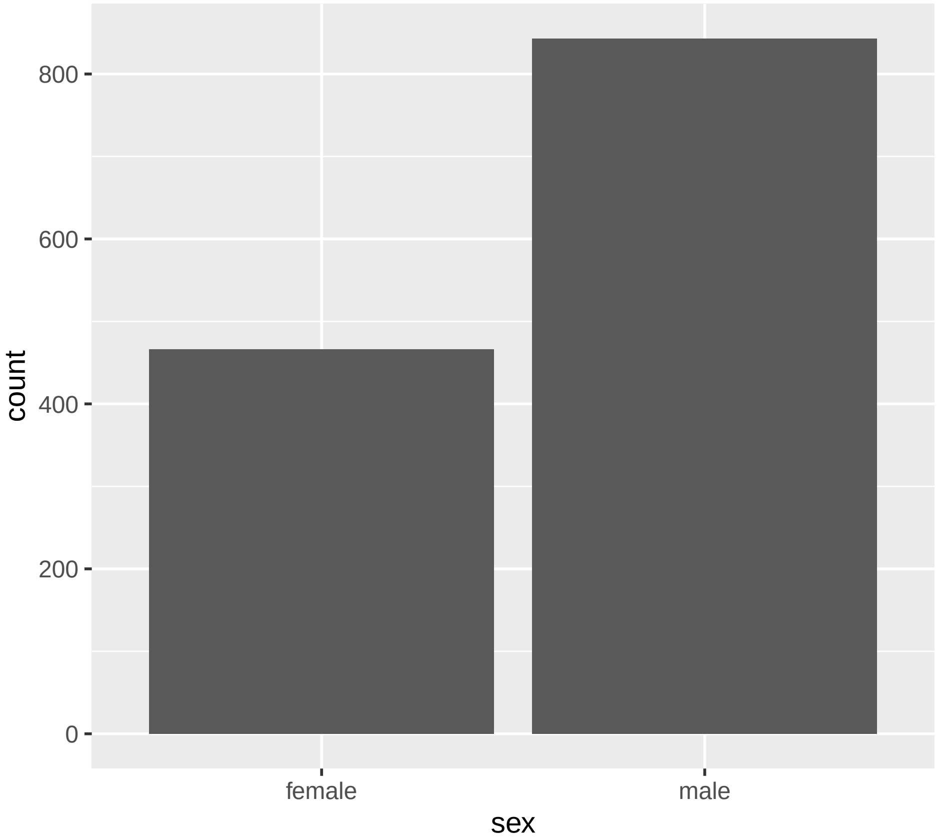
```
## x dplyr::lag()    masks stats::lag()
```

```
ggplot(titanic, aes(x = sex))
```



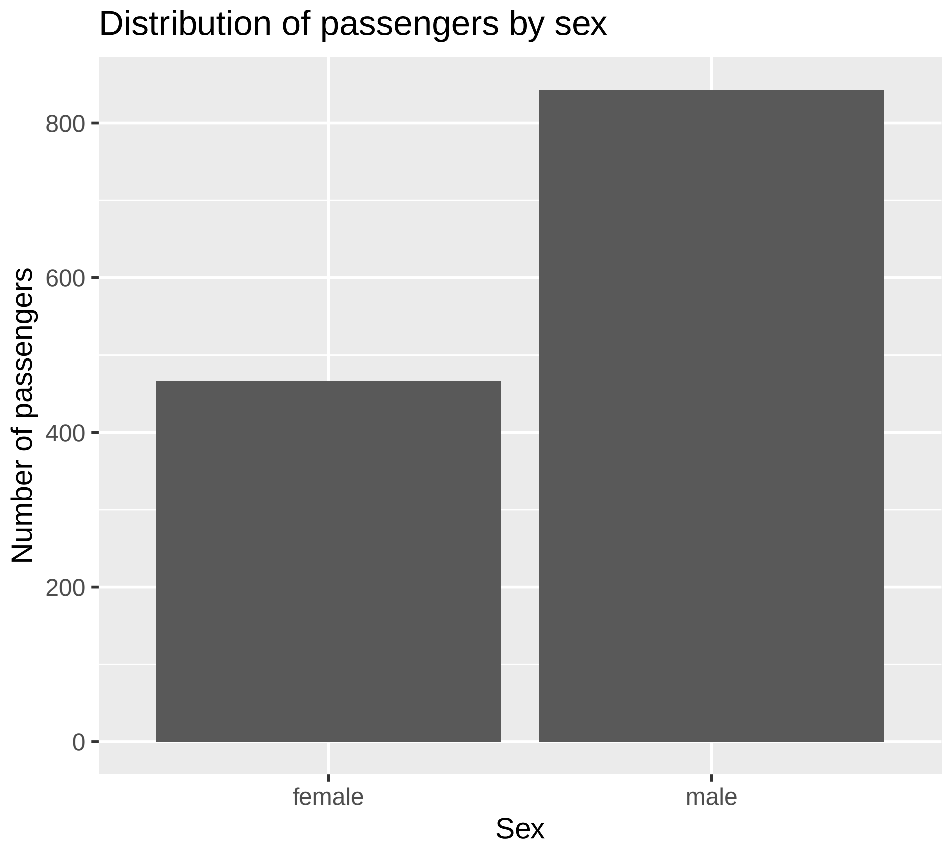
We have set the base of our graph and that is why you are seeing the shaded area where our graph will be positioned with two categories in the variable sex. Arguments ing the ggplot() function begin with the data, which in this case is titanic, followed by the aesthetic mapping including the sex variables on the x-axis. We are plotting the graphs in bits because we want to demostrate the layered principles underlying the ggplot2 package. We are now going to add the geom_bar() function because the geometric object we want is a barchart:

```
ggplot(titanic, aes(x = sex))+
  geom_bar()
```



Now need to provide the axis labels and title to our plot so that we can properly interpret it. The x-axis looks like it has already been labelled just fine. But we will include in the description so that we can know how to do it. The labels are added by the labs() function which is added as a new layer to our graph

```
ggplot(titanic, aes(x = sex))+
  geom_bar()+
  labs(x = "Sex", y = "Number of passengers", title = "Distribution of passengers by sex")
```



We can interpret our graph by saying the bar chart represents the distribution of passengers by sex. In terms of gender representation there were more male than female passengers in the Titanic. In other words, there were almost twice as many male passengers as there were female passengers in the Titanic. It is important tobe stylish in the maner that you interpret the graphs. You don't want to just be regurgitating the information that is already given in the graph. To be sure about the actual numbers in categories, its batter you run the table() function first before you graph the bar chart.

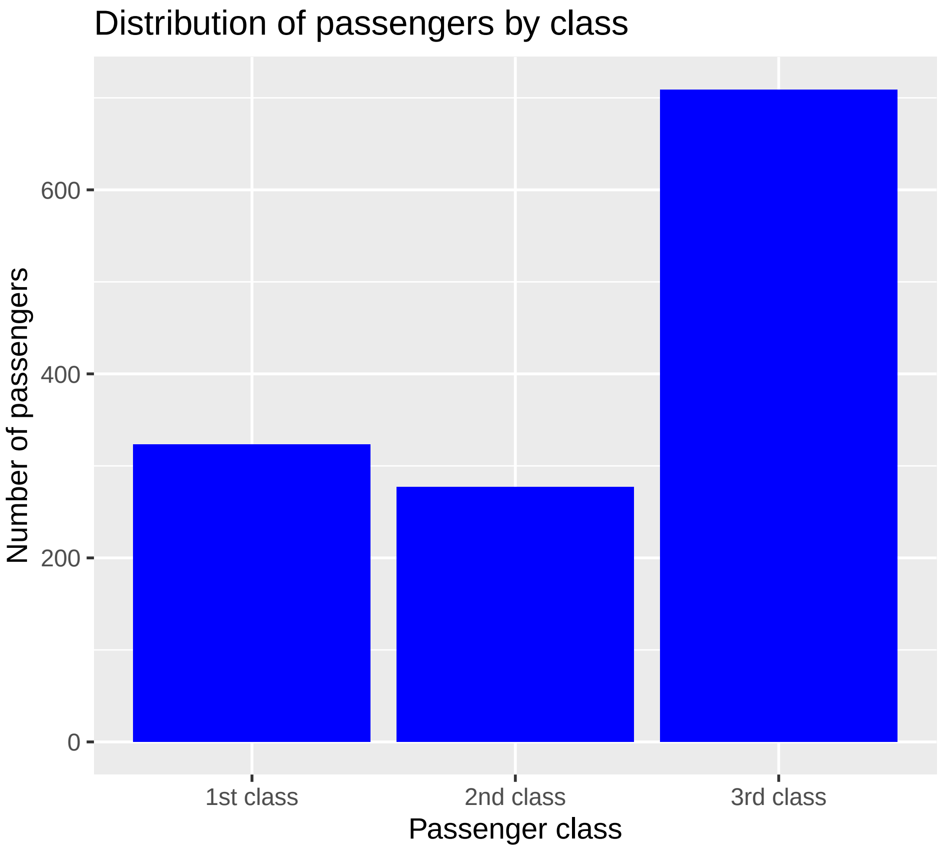
We will also use a barchart on an ordinal variable and for this we use the passenger class (pclass) variables which has three categories()

```
table(titanic$pclass)
```

```
##
## 1st class 2nd class 3rd class
##       323       277       709
```

There were 323 first class, 277 2nd clas and 709 third class passengers on the Titanic. Which means the classwith most passengers was the third class. To plot this distribution, we follow the steps we did in the previous bar chart and label it accordingly. Notice that our x-axis now goes to pclass. I have also included the fill argument in the geom_bar() function to change the colours of the bars to blue.

```
ggplot(titanic, aes(x = pclass))+
  geom_bar(fill = "blue")+
  labs(x = "Passenger class", y = "Number of passengers", title = "Distribution of passengers b
```

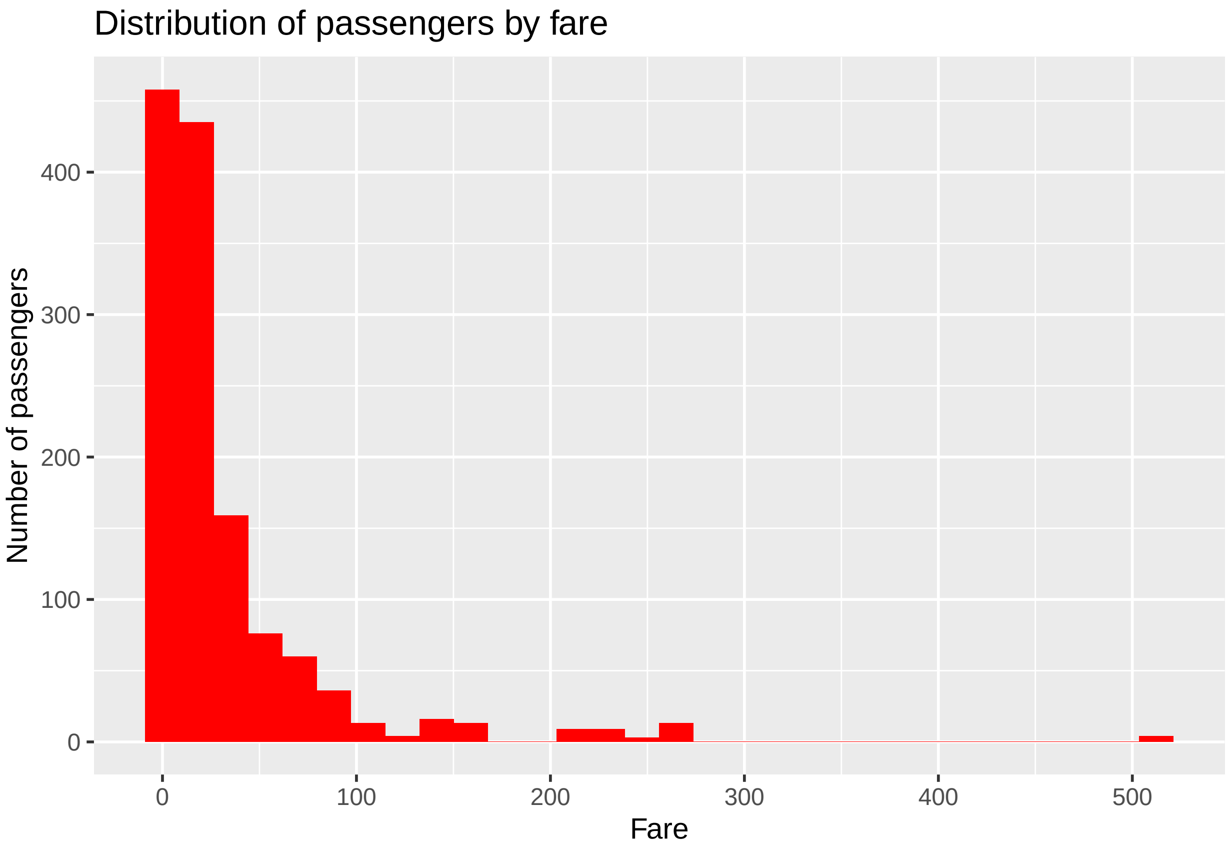


Histogram

As we have stated alreay, histogram isused to study the distribution of an interval/ratio variable. We use the fare variable for this. Again the procedure is the same only that thus time we are using the geom_histogram() because the geometric object we are interested in is the histogram. I am using the red colour for the bins.

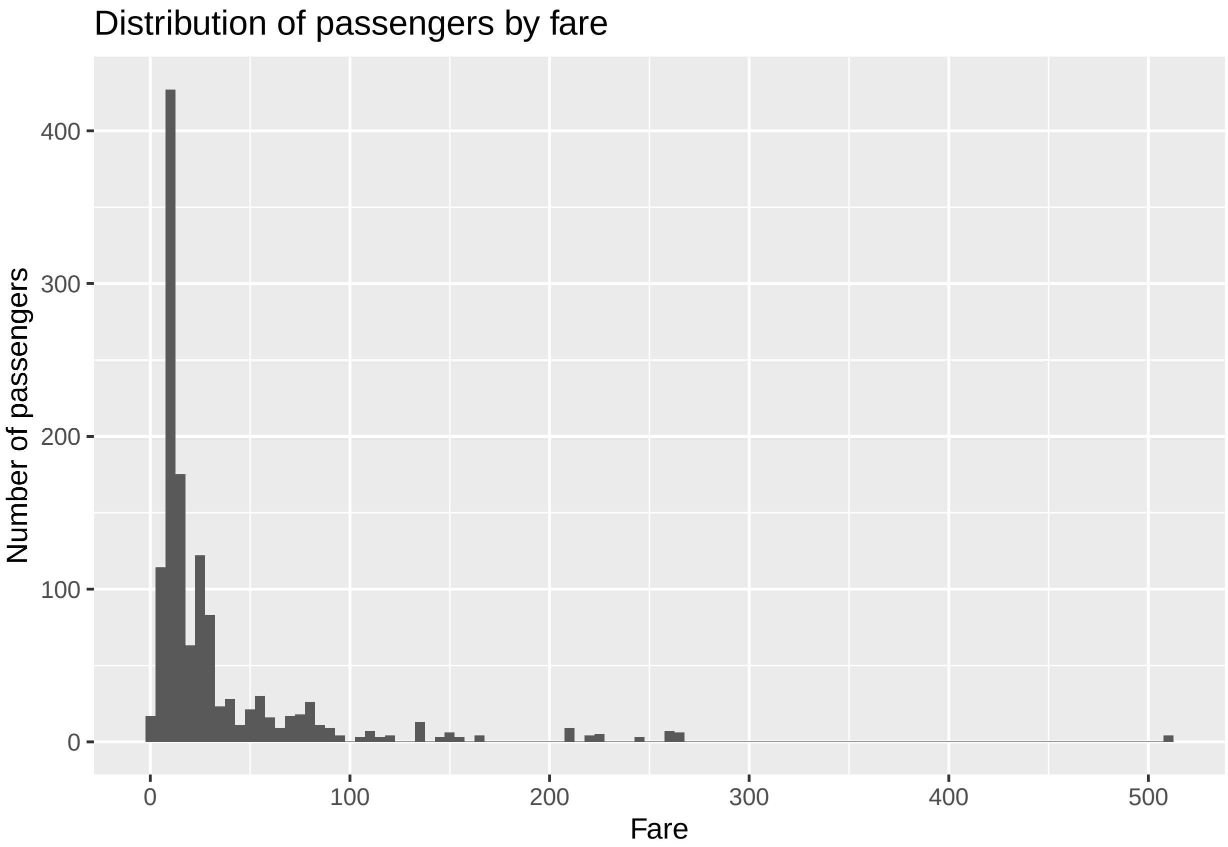
```
ggplot(titanic, aes(x = fare))+
  geom_histogram(fill = "red")+
  labs(x = "Fare", y = "Number of passengers", title = "Distribution of passengers by fare")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



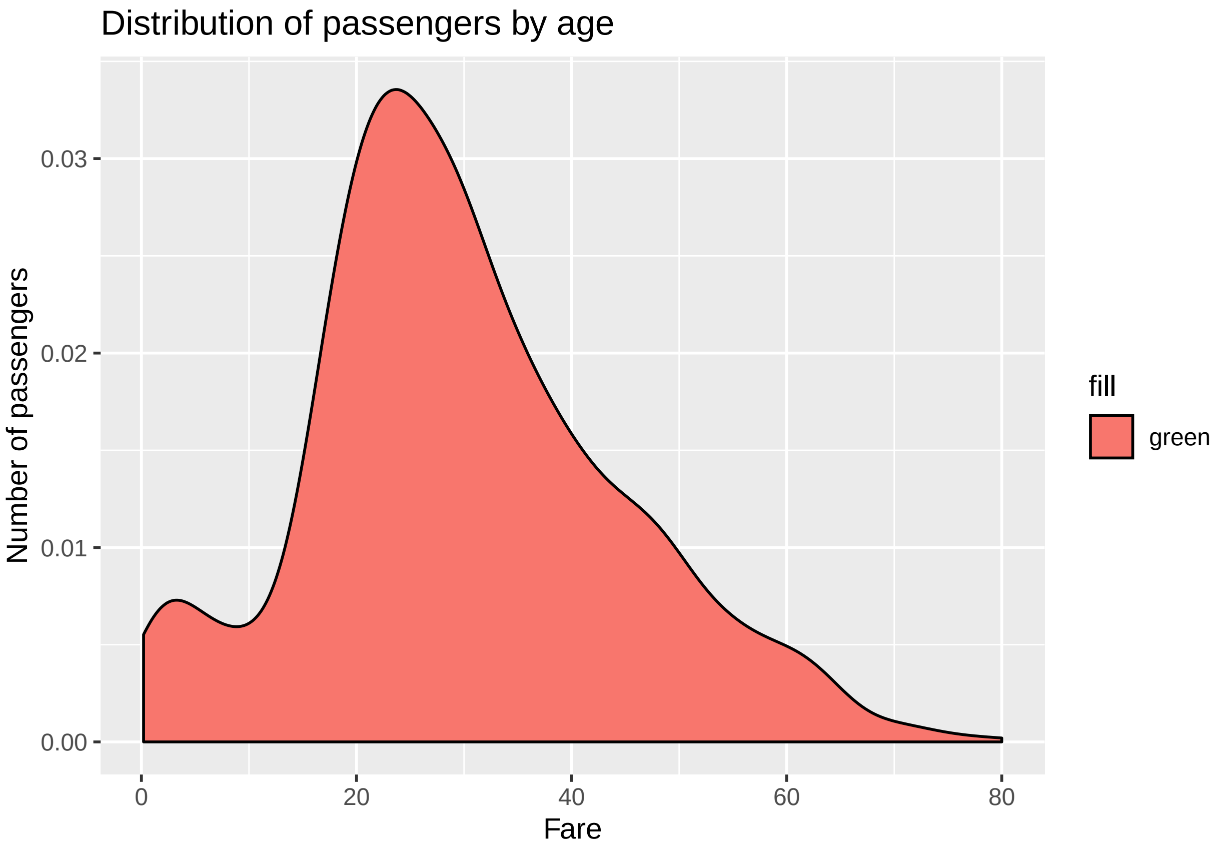
We can also change the size of the bins by supplying the binwidth argument. Lets change the binwidth to 1.

```
ggplot(titanic, aes(x = fare))+
  geom_histogram(binwidth = 5)+
  labs(x = "Fare", y = "Number of passengers", title = "Distribution of passengers by fare")
```



We can also create a histogram using smoothed density

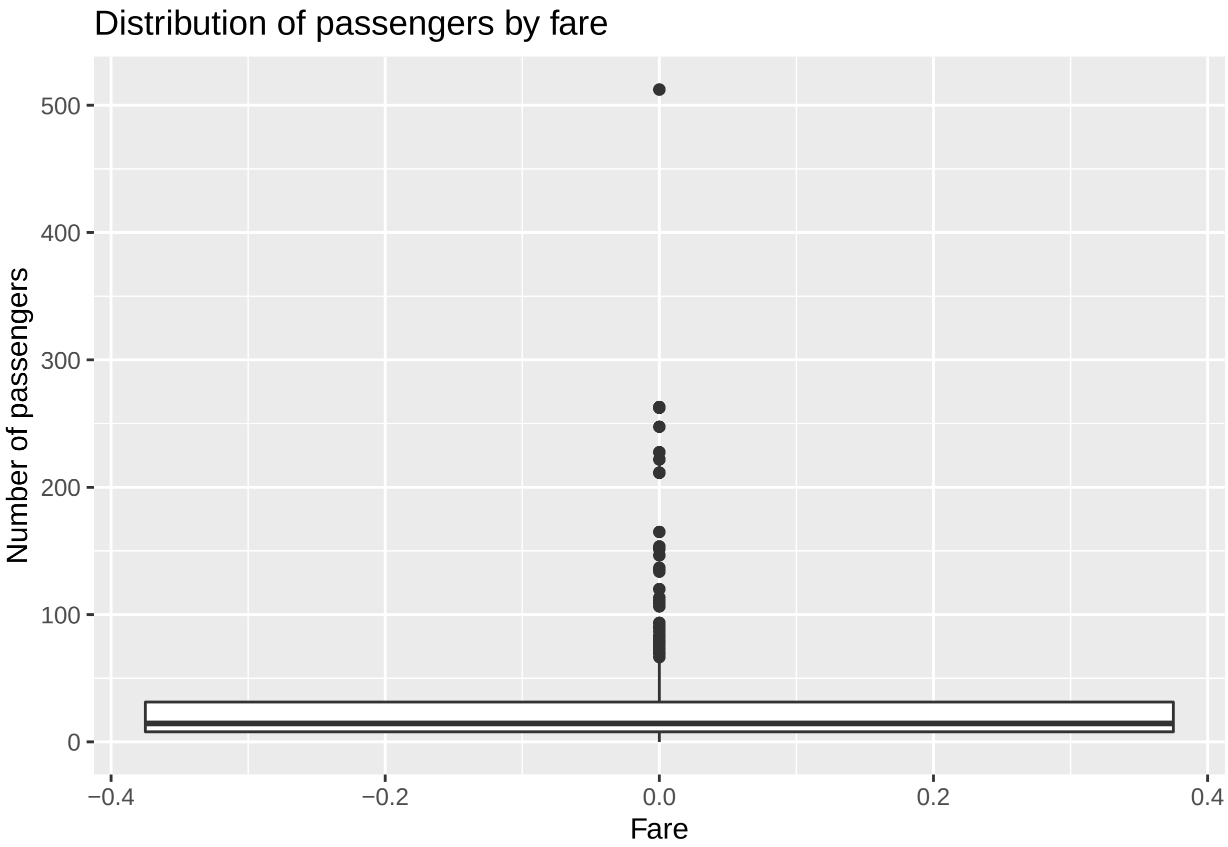
```
ggplot(titanic, aes(x = age, fill = "green"))+
  geom_density(binwidth = 5)+
  labs(x = "Fare", y = "Number of passengers", title = "Distribution of passengers by age")
```



Boxplot

We use fare for the box plot and we can see that there are quite a lot of outliers and it is even difficult for us to interpret it

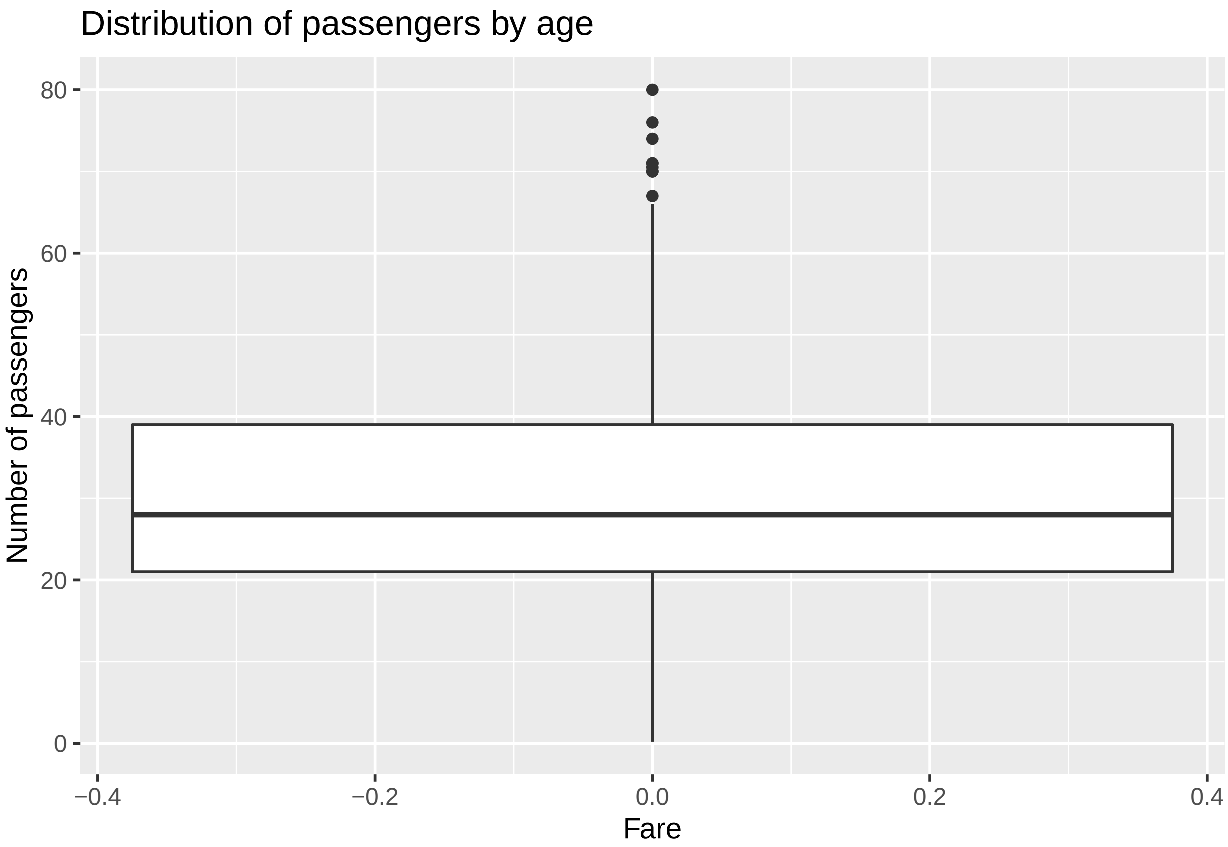
```
ggplot(titanic, aes(y = fare))+
  geom_boxplot()+
  labs(x = "Fare", y = "Number of passengers", title = "Distribution of passengers by fare")
```



We will use age for this and provide an interpretation

```
ggplot(titanic, aes(y = age))+
  geom_boxplot()+
  labs(x = "Fare", y = "Number of passengers", title = "Distribution of passengers by age")
```

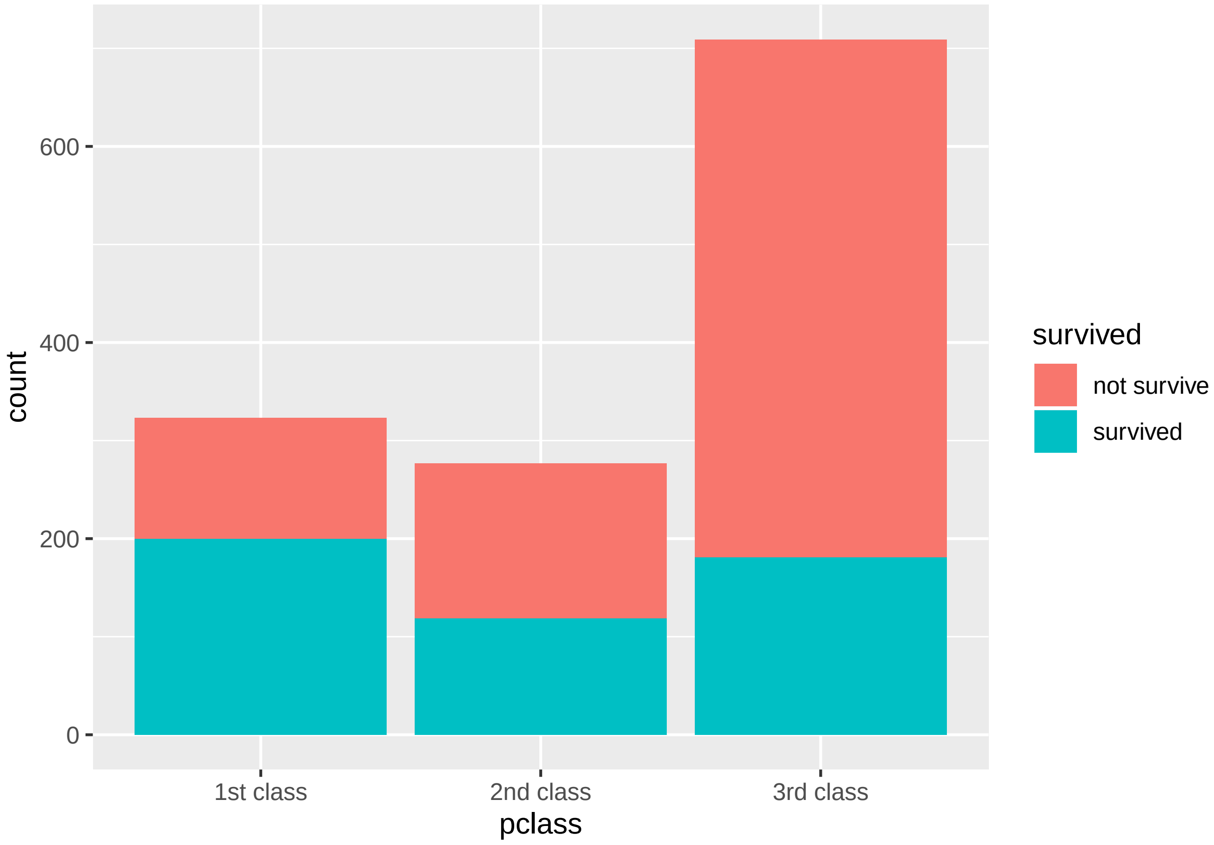
```
## Warning: Removed 263 rows containing non-finite values (stat_boxplot).
```



Bivariate data visualisation

Bivariate means two and in this case, we will combine two variables in one plot and we will use stacked bar charts, scatter plots and box plots

```
ggplot(titanic, aes(x = pclass, fill = survived))+
  geom_bar()
```


count
600
400
200
0
1st class
2nd class
3rd class
pclass
survived
not survive
survived