

KIT306/606 Tutorial 3

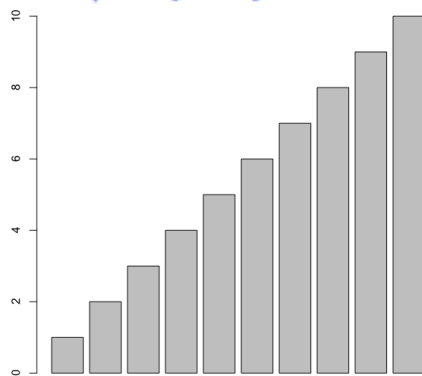
Student ID	Name

The following tutorial work should be completed by tutorial 4 (week5).

● Bar chart

We will use barplot function, which enables creating a bar plot with vertical or horizontal bars. Let's try to make a simple barplot with 1 to 10 data.

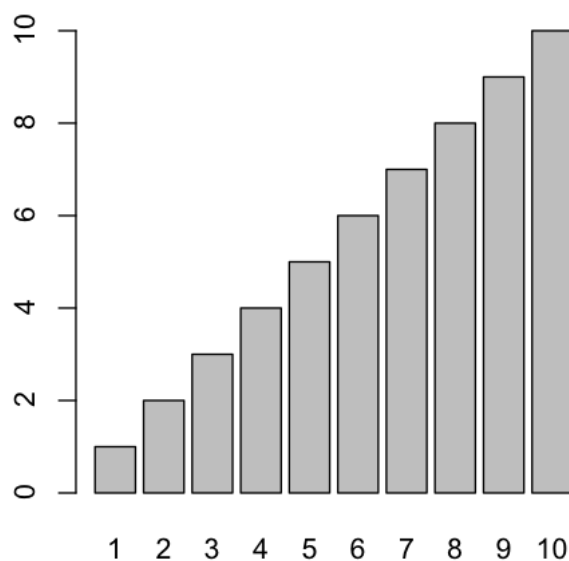
```
> barplot(1:10)
```



You can define the barplot with categories' names.

```
> barplot(1:10, names.arg=1:10)
```

- `names.arg`: a vector of names to be plotted below each bar or group of bars. If this argument is omitted, then the names are taken from the `names` attribute of `height` if this is a vector, or the column names if it is a matrix



Let's engage with dataset from csv file.

dog.csv file can be downloaded from MyLo website.

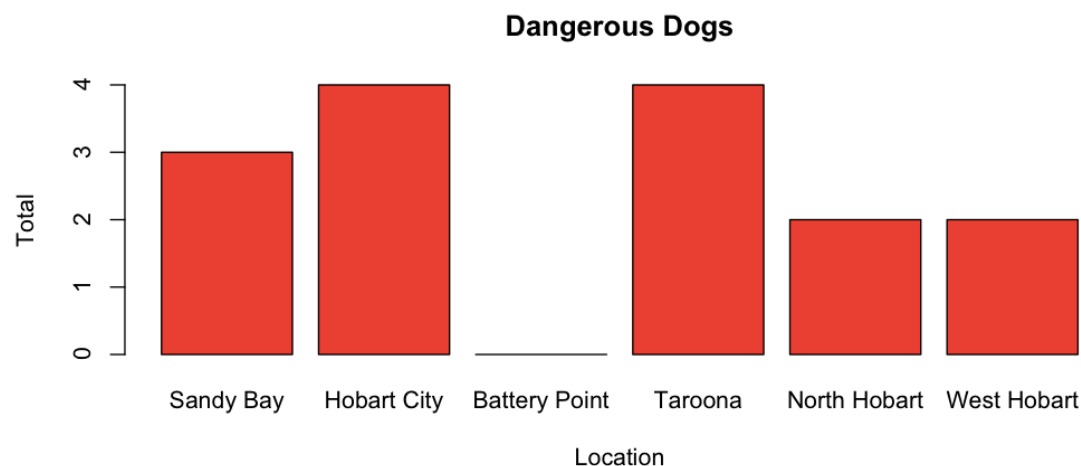
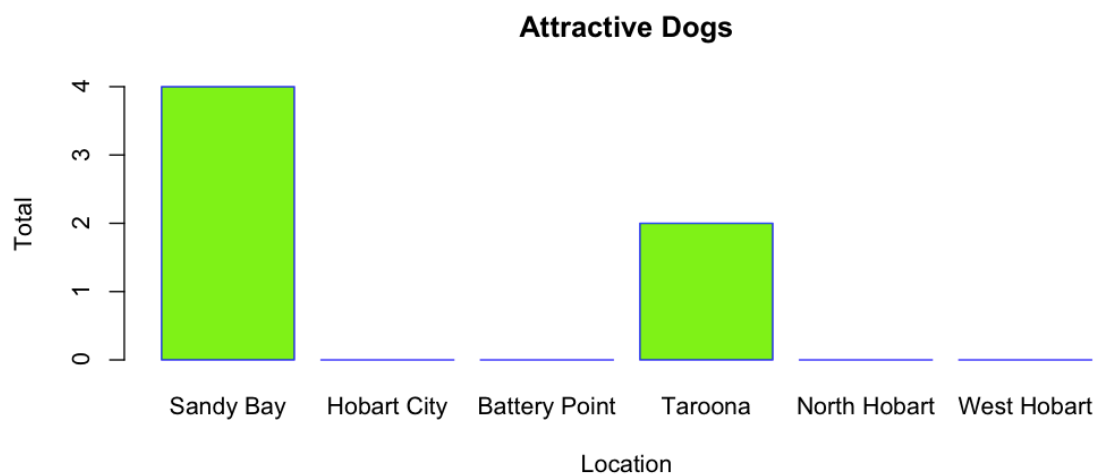
The file represents the number of three types of dogs (Attractive dogs, Dangerous dogs, and Others) in different locations.

```
> dogcsv<-read.csv("~/Desktop/R/dog.csv",head=T)
> dogcsv
```

	ID	Location	Attractive.Dogs	Dangerous.Dogs	Others
1	1	Sandy Bay	4	3	10
2	2	Hobart City	0	4	5
3	3	Battery Point	0	0	1
4	4	Taroona	2	4	5
5	5	North Hobart	0	2	2
6	6	West Hobart	0	2	4

Let's make the barplot for each type. Enter the following codes in R and check the result.

```
> barplot(dogcsv$Attractive.Dogs,names.arg=dogcsv$Location,main="Attractive
Dogs",xlab="Location",ylab="Total",border="blue",col="green")
> barplot(dogcsv$Dangerous.Dogs,names.arg=dogcsv$Location,main="Dangerous
Dogs",xlab="Location",ylab="Total",border="Black",col="red")
> barplot(dogcsv$Others,names.arg=dogcsv$Location,main="Other
Dogs",xlab="Location",ylab="Total",border="Black",col="grey")
```



● Mosaic plot

The function `mosaicplot()` enables plotting a mosaic.

Mosaicplot is suitable to analyzing and visualizing the data with multiple categorical variables.

Before we get started, let's enable the stats library to use Titanic dataset that contains 4 different categorical.

```
> require(stats)
```

The dataset can be seen with the following command.

```
> Titanic
```

```
, , Age = Child, Survived = No
```

	Sex	
Class	Male	Female
1st	0	0
2nd	0	0
3rd	35	17
Crew	0	0

```
, , Age = Adult, Survived = No
```

	Sex	
Class	Male	Female
1st	118	4
2nd	154	13
3rd	387	89
Crew	670	3

```
, , Age = Child, Survived = Yes
```

	Sex	
Class	Male	Female
1st	5	1
2nd	11	13
3rd	13	14
Crew	0	0

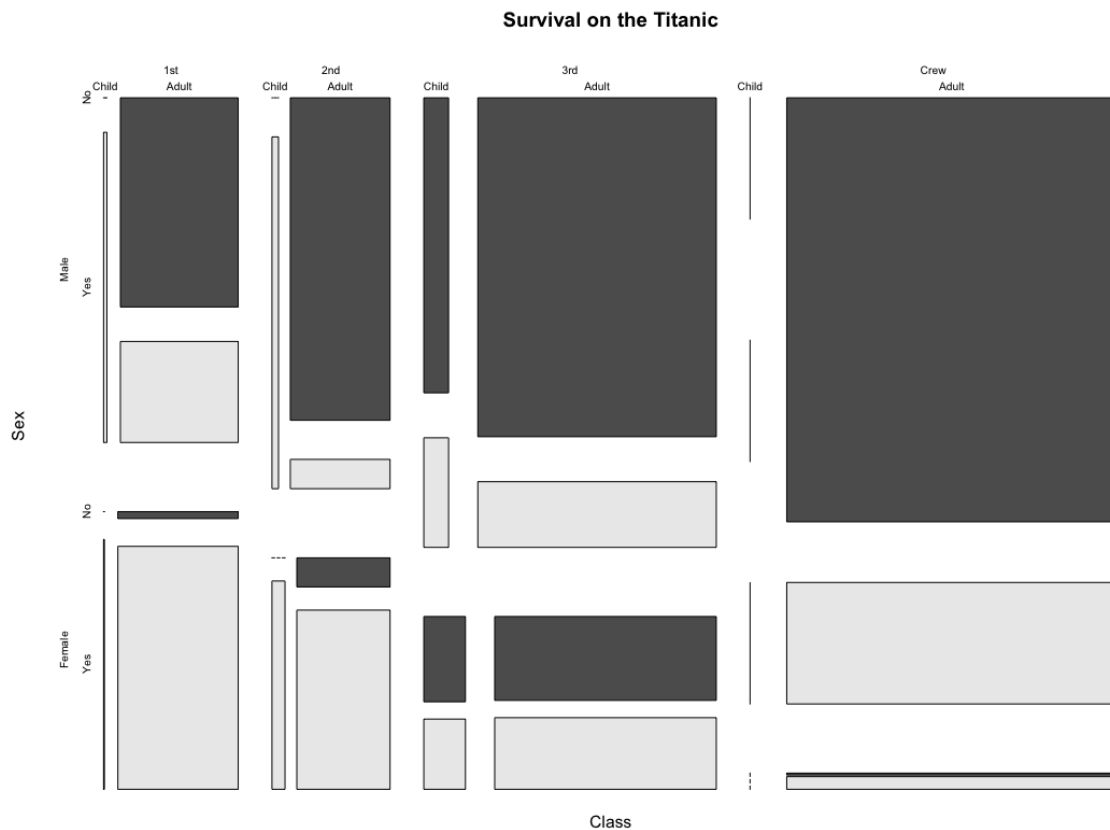
```
, , Age = Adult, Survived = Yes
```

	Sex	
Class	Male	Female
1st	57	140
2nd	14	80
3rd	75	76
Crew	192	20

The data set 'Titanic' provides information on the fate of passengers on the fatal maiden voyage of the ocean liner 'Titanic', summarized according to economic status (class), sex, age and survival. If you want to have a look at the description (or any other detailed information) of this dataset 'Titanic', you can put the following command in R console.

```
> help(Titanic)
```

Then, you can draw mosaic plot with the above dataset.
(Please zoom the pdf file if you want to have a look at the graph properly)

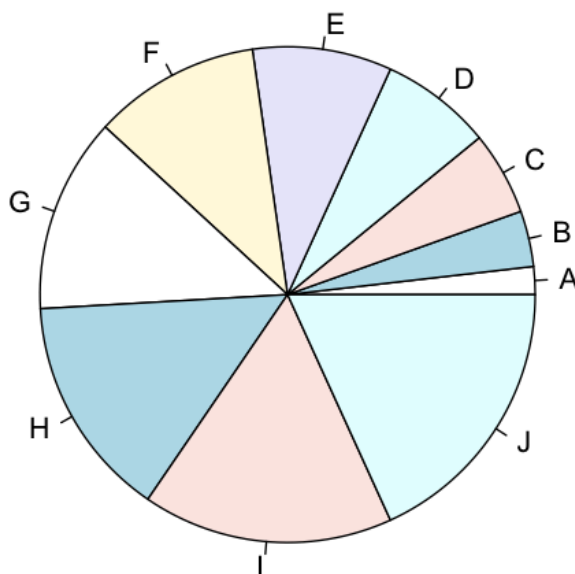


● Pie chart

The function `pie()` enables drawing a pie chart.

Let's define the `x` variable as 1 to 10 dataset. Then, you can draw pie chart with letters label. LETTERS shows the 26 upper-case letters of the Roman alphabet.

```
> x=1:10
> pie(x, labels=LETTERS[1:10])
```



You can decorate the chart or change the angle of the pie chart with the following commands.

```
> pie(x,labels=LETTERS[1:10],main="pie chart",clockwise=T)
> pie(x,labels=LETTERS[1:10],main="pie chart",init.angle=45)
> pie(x,labels=LETTERS[1:10],main="pie chart",init.angle=90)
```

- main: an overall title for the plot
- clockwise: logical indicating if slices are drawn clockwise or counter clockwise (i.e., mathematically positive direction), the latter is default
- angle: the slope of shading lines, given as an angle in degrees (counter-clockwise)
- More Arguments can be found by using **help(pie)** command.

Now, as you understand the basic idea of drawing pie chart, it is time to draw the pie chart with the csv dataset. In page 2, we defined dogcsv variable (reading from dog.csv file).

In order to draw the chart, we should use round function for get percentile value. The percentile value for each location category can be seen with calling dog_per variable as follows:

```
> dog_per<-round(dogcsv$0thers/sum(dogcsv$0thers)*100)
> dog_per
[1] 37 19 4 19 7 15
```

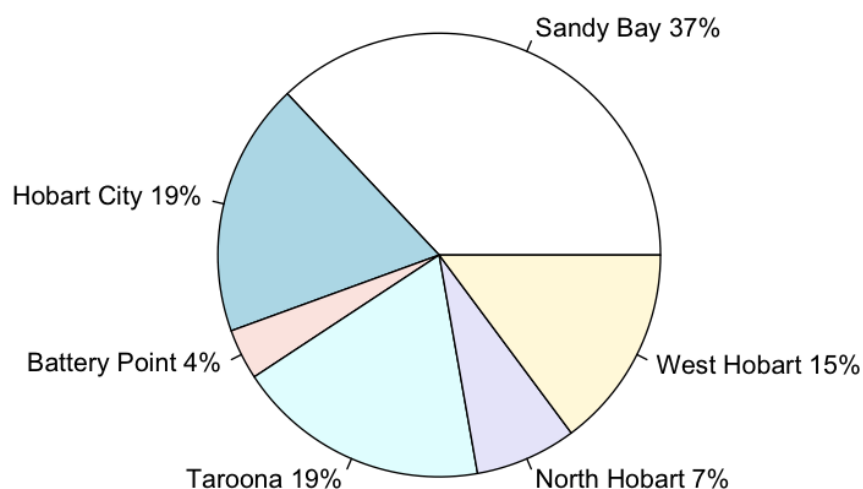
Labels can be added with paste function. The following command makes the label in this structure: Location name (dogcsv\$Location), percentile for each location (dog_per), %, and those will be separated in " " space.

```
> labels<-paste(dogcsv$Location,dog_per,"%",sep=" ")
```

After calculating percentile and defining labels, you are now ready to draw the pie chart.

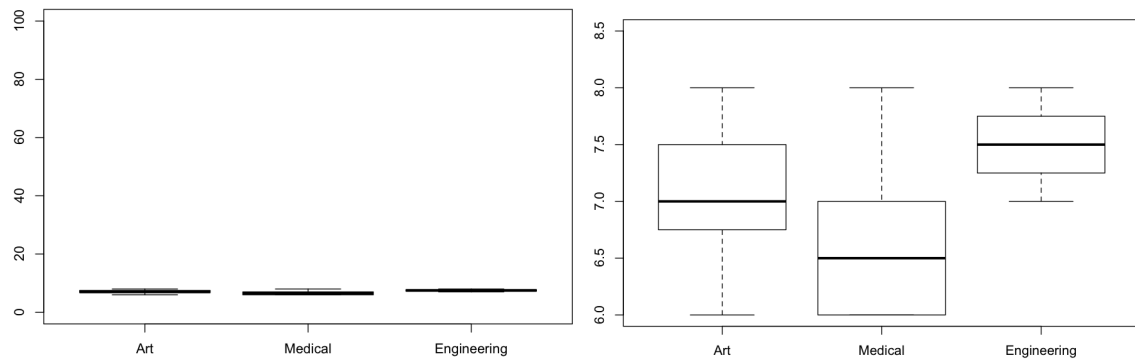
```
> pie(dogcsv$0thers,labels=labels, main="other dogs")
> pie(dogcsv$0thers,labels=labels, main="other dogs",col=rainbow(length(labels)))
```

other dogs



Tutorial Questions

1. Sleep.csv shows the students' sleeping hour while they are taking different courses (Art, Medical, and Engineering). Download sleep.csv file and define it as a variable sleep.
2. Draw two boxplots (as follows) that contain the following three variables (sleep\$Art, sleep\$Medical, and sleep\$Engineering). Then, explain which one is better and the reason why you think.



3. Draw the bar chart that shows different sleeping hours while students are taking Art Course.

