workshop04

February 3, 2019

In [2]:
```
library(plotly)
library(tidyverse)
library(GGally)
library(gcookbook)
library(gridExtra)

# Plot size deppening on your screen resolution to 5 x 3
options(repr.plot.width=5, repr.plot.height=3)
```


In [ ]:

1 Welcome to Workshop 4

There are very few options to plot single categorical variable. The basic choices are bar charts and pie charts. Pie charts are not favoured by many people. Quote from Edward Tufte's book "The Visual Display of Quantitative Information": "A table is nearly always better than a dumb pie chart; the only worse design than a pie chart is several of them, for then the viewer is asked to compare quantities located in spatial disarray both within and between charts [...] Given their low density and failure to order numbers along a visual dimension, pie charts should never be used." See also the forum http://www.edwardtufte.com.

1.0.1 Exercise 1

Aims: * Learn how to visualise categorical variables * Learn how to interprete mosaic plots

The dataset "Titanic" available in R, is a multidimensional table. To see it, type `Titanic`.

In [9]:
```
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
```


Now convert the table to a data frame called `titantic.df`:

```
In [10]: titantic.df <- data.frame(Titanic)
```

```
In [19]: titantic.df
```

Class	Sex	Age	Survived	Freq
1st	Male	Child	No	0
2nd	Male	Child	No	0
3rd	Male	Child	No	35
Crew	Male	Child	No	0
1st	Female	Child	No	0
2nd	Female	Child	No	0
3rd	Female	Child	No	17
Crew	Female	Child	No	0
1st	Male	Adult	No	118
2nd	Male	Adult	No	154
3rd	Male	Adult	No	387
Crew	Male	Adult	No	670
1st	Female	Adult	No	4
2nd	Female	Adult	No	13
3rd	Female	Adult	No	89
Crew	Female	Adult	No	3
1st	Male	Child	Yes	5
2nd	Male	Child	Yes	11
3rd	Male	Child	Yes	13
Crew	Male	Child	Yes	0
1st	Female	Child	Yes	1
2nd	Female	Child	Yes	13
3rd	Female	Child	Yes	14
Crew	Female	Child	Yes	0
1st	Male	Adult	Yes	57
2nd	Male	Adult	Yes	14
3rd	Male	Adult	Yes	75
Crew	Male	Adult	Yes	192
1st	Female	Adult	Yes	140
2nd	Female	Adult	Yes	80
3rd	Female	Adult	Yes	76
Crew	Female	Adult	Yes	20

Have a careful look at it the first few rows and relate them to the table from above. A table can be multidimensional. When it is converted to a data frame, the same information is provided by having more columns. Note that four of the columns refer to variables, and the fifth to frequency (Freq)

1. Question:

In the data frame titantic.df, the rows don't correspond to individuals on the Titanic. What do they correspond to?

2. Question:

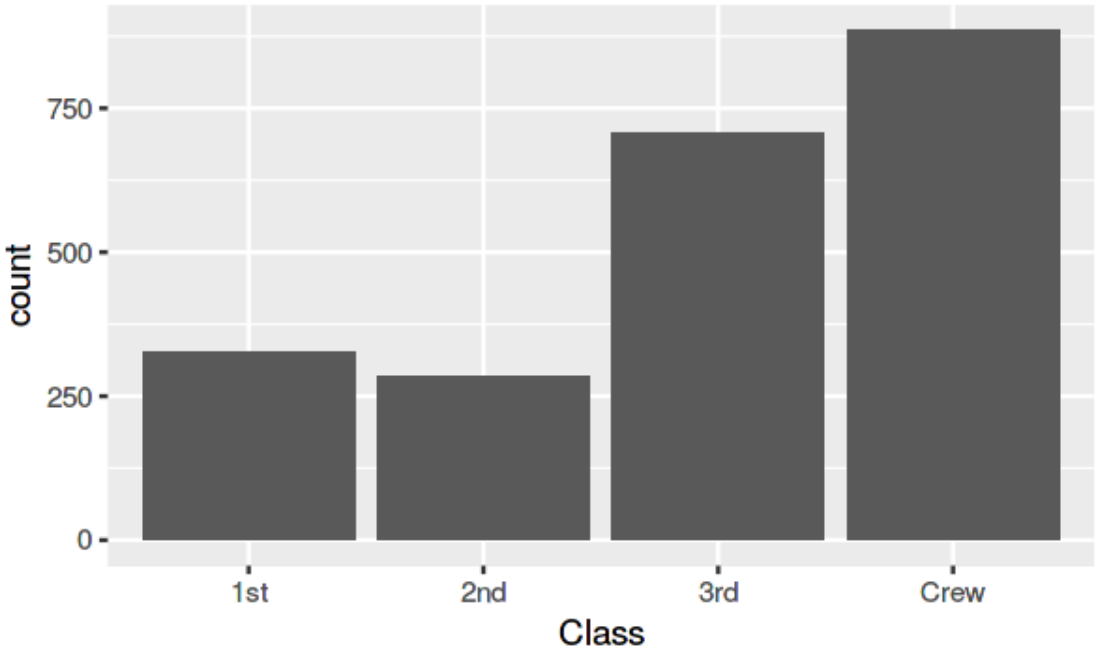
What are the variables that we might draw bar charts for?

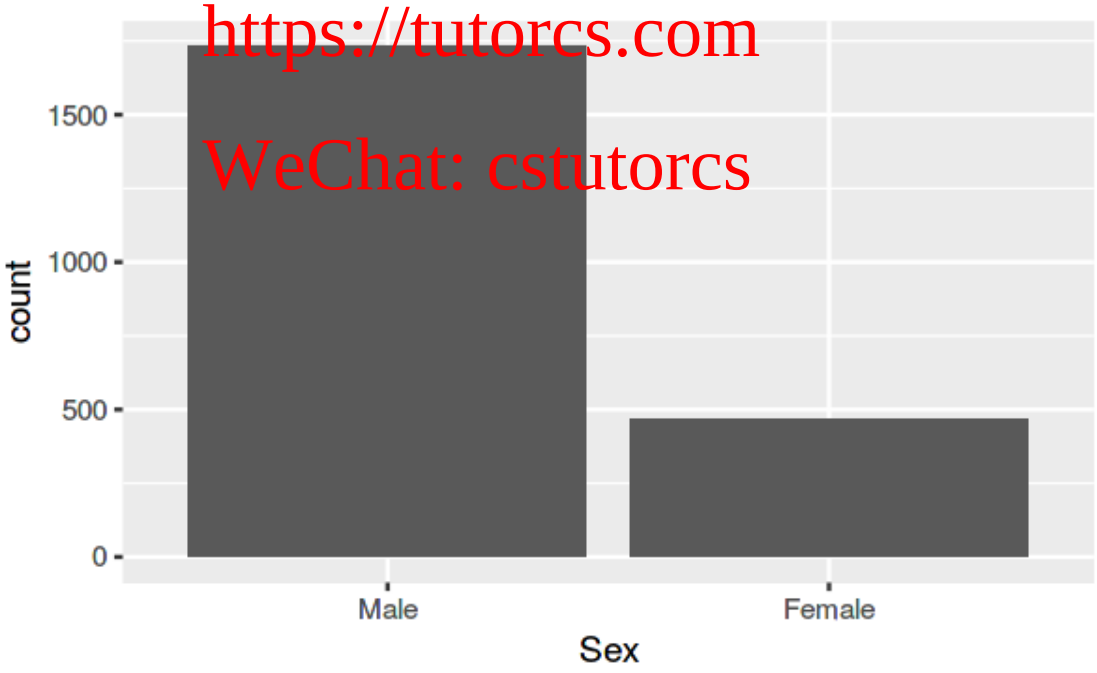
Each row of the data frame is a category, for example: row 31 refers to Third class Female Adults who Survived. And there were 76 of these. So how do we create bar charts using data like this?

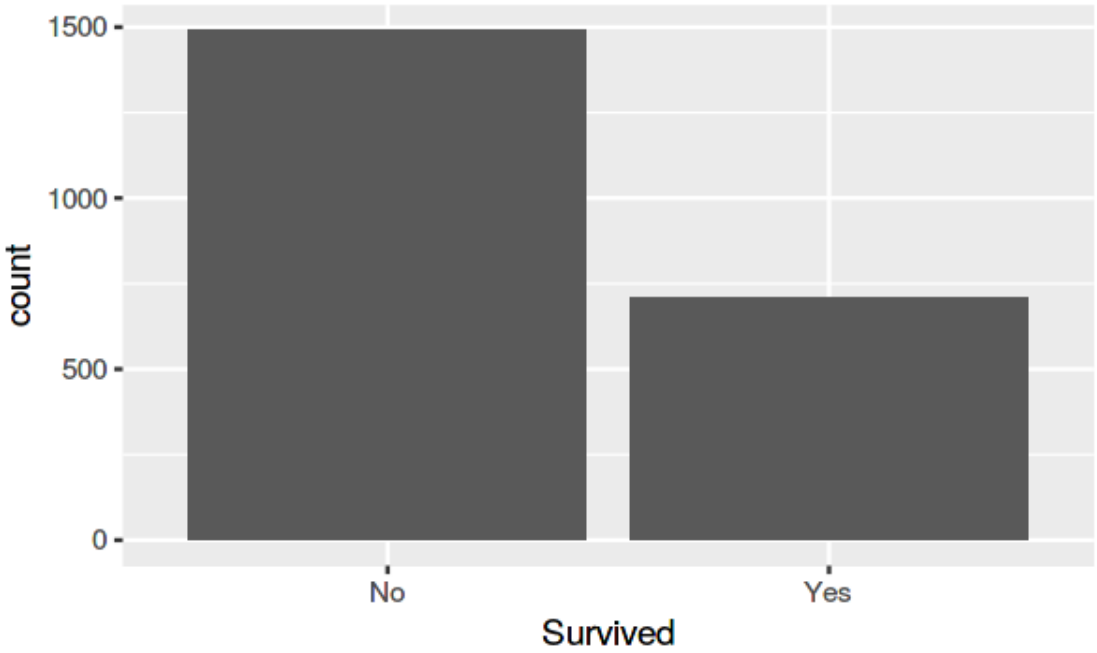
Exercise: Create bar charts for each of the variables: Class, Sex, Age and Survived. The frequencies indicate how much weight to give to each line of the data frame, and so the aesthetic to use in this case is "weight". We will produce bar charts for the four variables listed above. We start

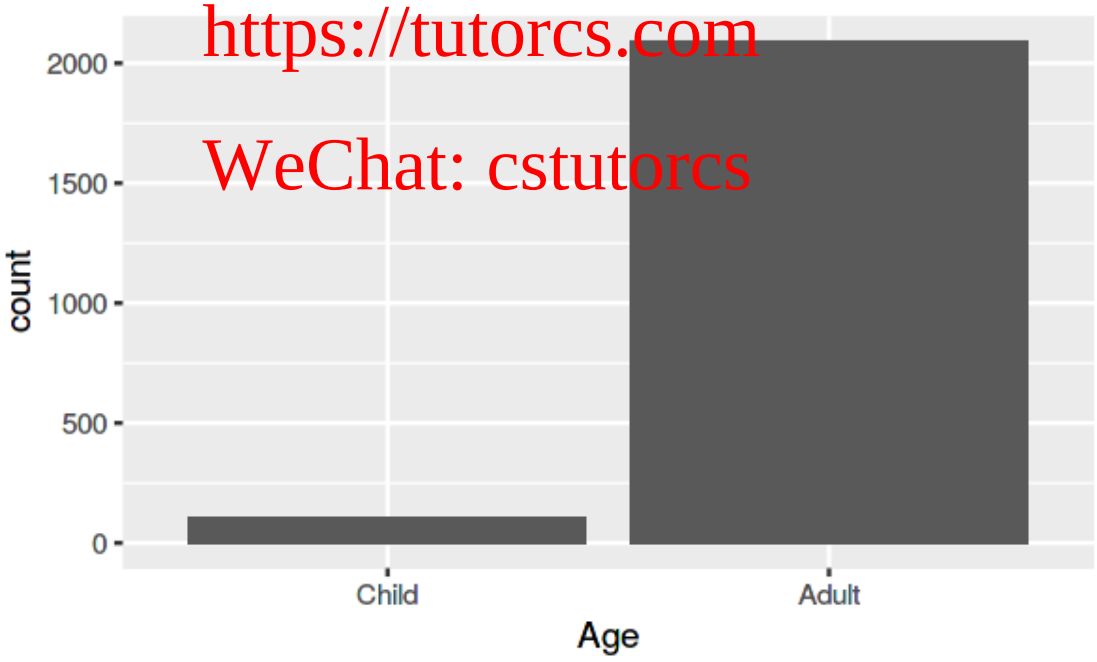
by writing the common part of the plot command:

```
In [55]: pBase <- ggplot(data=titantic.df,
                         aes(weights = Freq))
         pBase
         pClass <- ggplot(data=titantic.df,
                         aes(weights = Freq,x=Class))+ geom_bar()
         pClass
         pSex <- ggplot(data=titantic.df,
                         aes(weights = Freq,x=Sex))+ geom_bar()
         pSex
         pSurvived <- ggplot(data=titantic.df,
                         aes(weights = Freq,x=Survived))+ geom_bar()
         pSurvived
         pAge <- ggplot(data=titantic.df,
                         aes(weights = Freq,x=Age))+ geom_bar()
         pAge
```









This means when we plot a bar chart, each row of data doesn't only add one to the height of the appropriate column, but it adds whatever is in the Freq column. Thus the contribution from a row is weighted by the Freq column. This is obviously what we want in this case since for example, Freq = 35 tells us that there were 35 male children in the 3rd class who didn't survive. create each

bar chart now by appending with `aes(x = VAR ) + geom_bar()` and store it in corresponding variables:

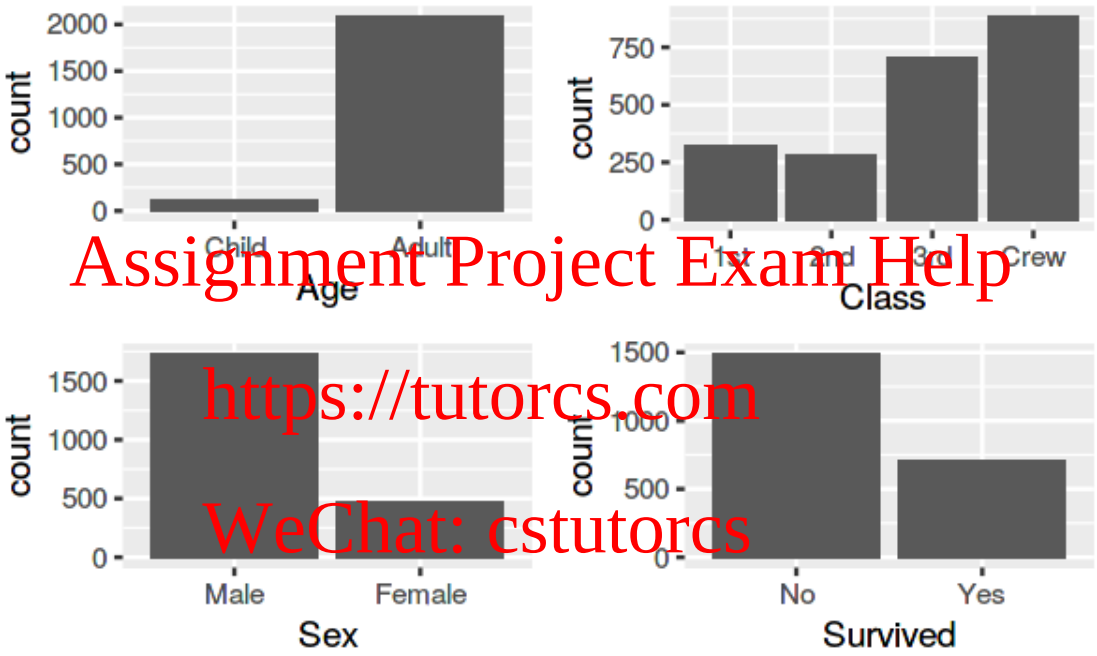
```
In [ ]:
```

```
In [56]: ls()
```

1. 'pAge' 2. 'pBase' 3. 'pClass' 4. 'pSex' 5. 'pSurvived' 6. 'titantic.df' 7. 'Var1'

Let's say we want to plot all of them side by side in a grid, a command for this would be `grid.arange(Var1,Var2,.. nrow=2)` with the last bit representing the number of rows:

```
In [68]: grid.arrange(pAge,pClass,pSex,pSurvived)
```



As each graph has a separate scale, comparison is a bit difficult. Redo the procedure but this time pass `ylim(0,2250)` to be the base object. Later we will learn how to do this all at once:

```
In [69]: ylim(0,2250)
```

```
<ScaleContinuousPosition>
 Range:
 Limits:    0 -- 2.25e+03
```

1.0.2 Excercise 2: Mosaic Plots

A plot that is useful in representing categorical variables is a mosaic plot. A favourite command is `mosaic` in the package `vcd`. We will use mosaic from vcd (visualisation of categorical data). The input is a cross-tabulated frequency table. For example, this is exactly what the dataset Titanic is.

Note: This is the first time we are not using a ggplot package for visualisation

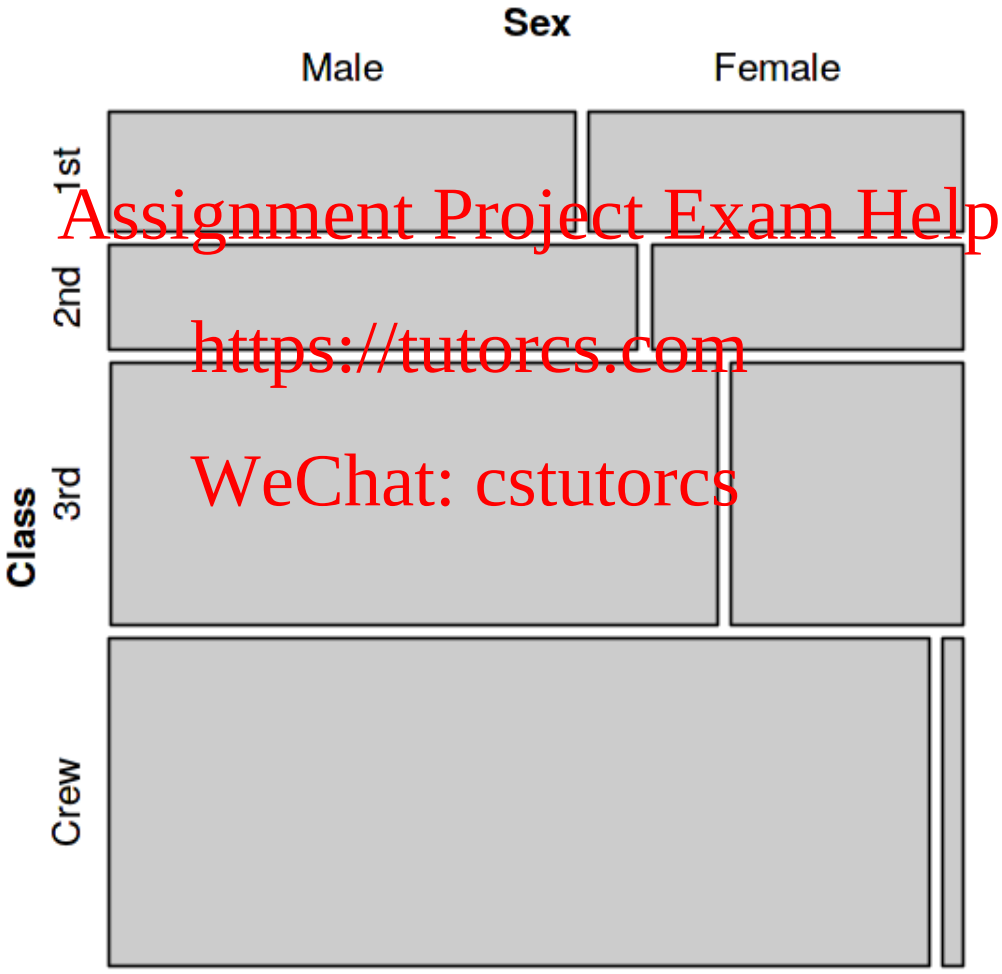
Load the library `vcd` and create your first mosaic plot with `mosaic(Freq~Class+Sex, data = titantic.df)`

```
In [70]: #lets increase the screen size for the plots
         options(repr.plot.width=5, repr.plot.height=5)
```

```
In [71]: library(vcd)
```

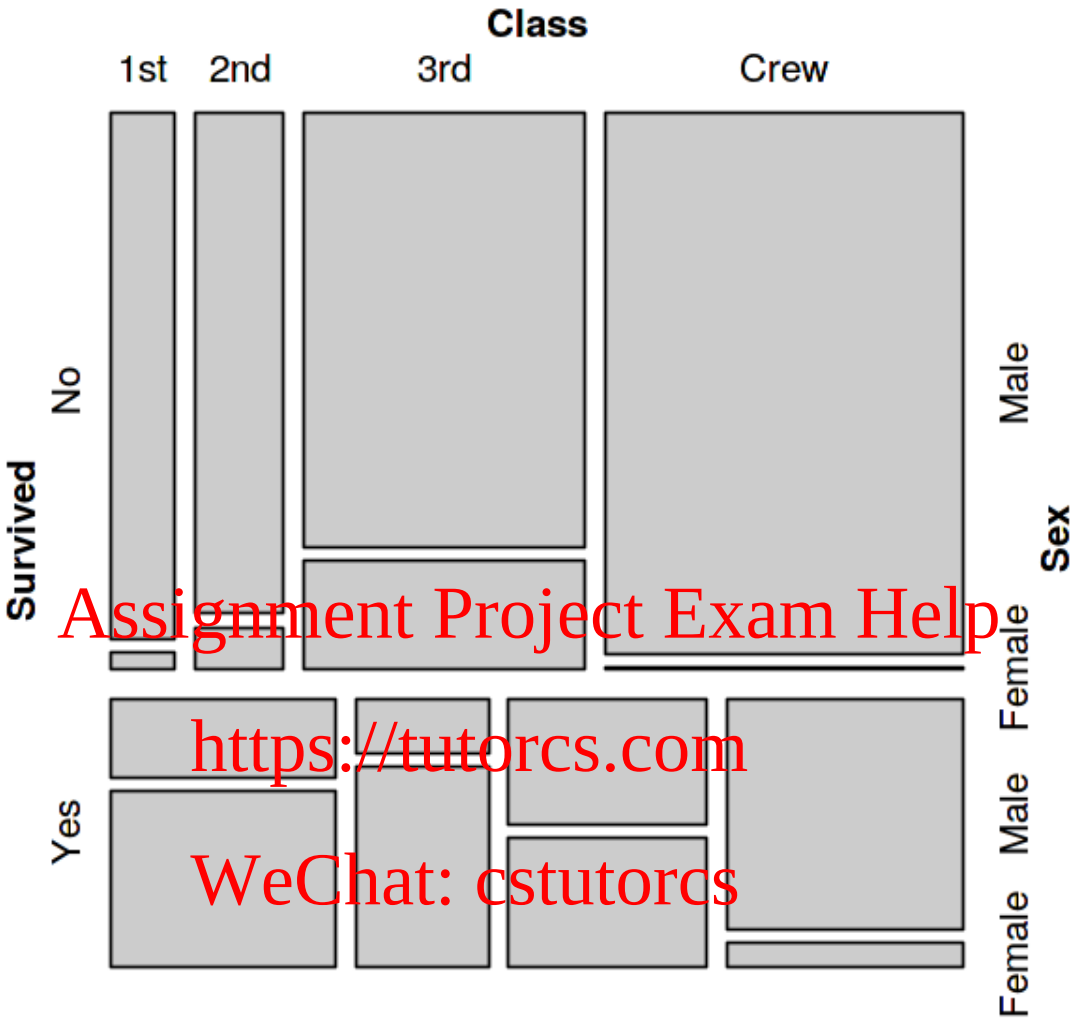
```
Loading required package: grid
```

```
In [72]: mosaic(Freq~Class+Sex, data = titantic.df)
```



Now, try out `Freq~Survived+Class+Sex`:

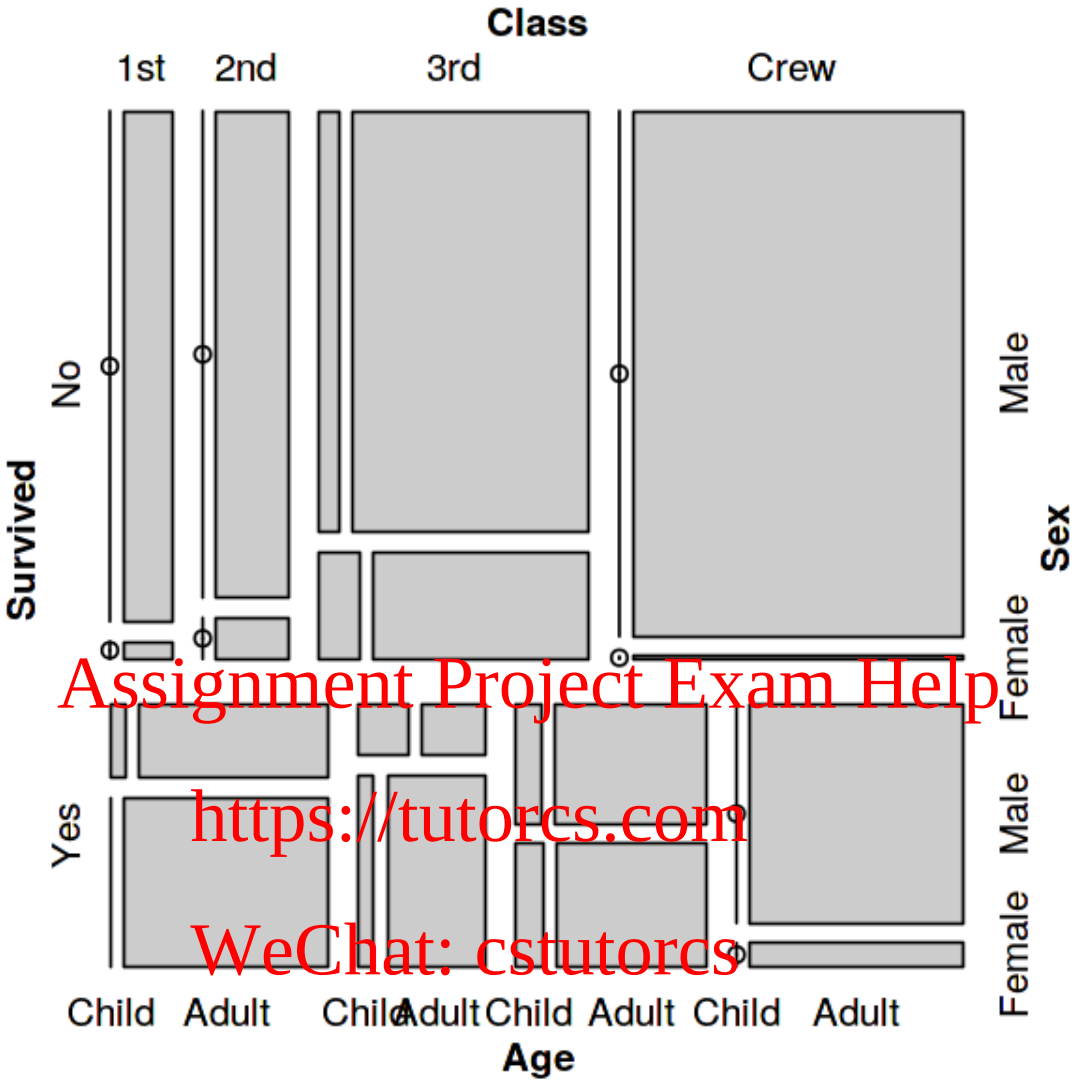
```
In [74]: mosaic(Freq~Survived+Class+Sex, data = titantic.df)
```



Make sure you can interpret these mosaic plots. For example, * consider third-class survivors. What proportion of this was female? * consider third class non-survivors. What proportion of this was female?

Are these the questions you want to answer? If not, arrange the variables differently. Try a few and decide which are the most useful. Try also including all four variables. That is, include Age as well. Compare it to the previous bar charts.

```
In [75]: mosaic(Freq~Survived+Class+Sex+Age, data = titantic.df)
```



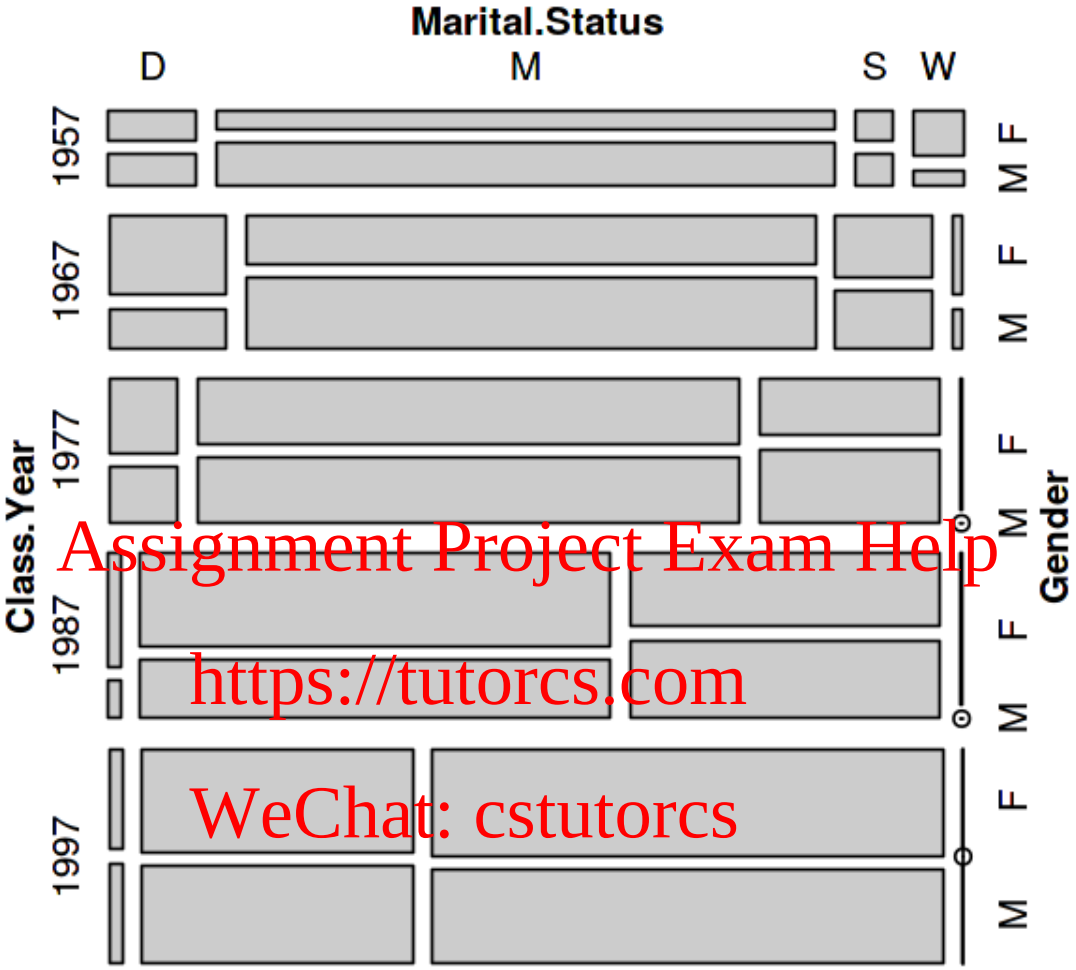
The other possible use is, if every row represents one observations. Load the csv `contribution.csv` and plot `~Class.Year+Marital.Status+Gender`

```
In [90]: contribution<-read.table(file = "contribution.csv",
                                  header = TRUE,
                                  sep = ",")
         contribution
```

Gender	Class.Year	Marital.Status	Major	Next.Degree	Give04	Give03	Give02
M	1957	M	History	LLB	2500	2500	1400
M	1957	M	Physics	MS	5000	5000	5000
F	1957	M	Music	NONE	5000	5000	5000
M	1957	M	History	NONE	0	5100	200
M	1957	M	Biology	MD	1000	1000	1000
F	1957	M	Mathematics	NONE	0	0	0
F	1957	S	History	MA	0	0	0
F	1957	M	Music	NONE	100	100	100
M	1957	M	Spanish	NONE	100	100	100
M	1957	M	English	NONE	0	0	0
F	1957	M	Mathematics	TC	0	0	0
F	1957	M	Psychology	NONE	0	0	0
M	1957	M	Economics-Business	NONE	0	0	0
M	1957	M	Biology	PHD	1500	1500	1500
F	1957	M	English	MLS	1500	1500	1500
F	1957	M	Comparative Literature	NONE	0	0	0
M	1957	D	Mathematics	PHD	0	0	0
M	1957	M	Psychology	JD	158	157	156
F	1957	W	Sociology	ME	1500	1500	1000
M	1957	M	Philosophy-Religion	STM	100	100	100
F	1957	M	Sociology	NONE	0	150	50
F	1957	M	English	MA	500	500	400
M	1957	M	Economics-Business	NONE	150	75	0
M	1957	M	Economics-Business	MBA	0	50	0
F	1957	W	Education	NONE	50	50	50
M	1957	M	Physical Education	MA	0	50	0
F	1957	M	Education	NONE	360	357	335
M	1957	M	Economics-Business	NONE	0	0	0
F	1957	W	English	NONE	0	0	0
M	1957	M	English	MA	0	0	0
M	1997	S	Physics	BSE2	0	160	157
F	1997	M	Spanish	NONE	0	0	0
M	1997	M	Music	MA	0	0	0
F	1997	S	English	MA	0	25	10
M	1997	S	Economics	JD	0	0	0
F	1997	M	German	NONE	0	0	0
M	1997	S	Political Science	JD	50	25	25
M	1997	S	History	NONE	0	0	0
F	1997	S	History	NONE	0	0	40
M	1997	M	General Science-Physics	NONE	0	0	0
M	1997	M	Economics	NONE	0	0	0
M	1997	M	Chinese	NONE	158	50	21
F	1997	S	Anthropology	NDA	0	0	0
M	1997	M	Anthropology	NONE	50	50	25
M	1997	M	Music	NONE	30	25	10
M	1997	S	Economics	JD	50	50	50
M	1997	S	History	NONE	0	0	0
M	1997	S	History	JD	50	50	50
M	1997	M	Sociology	ME	158	157	25
F	1997	M	Political Science	NDA	250	200	200
M	1997	S	Political Science	NDA	5	0	0

```
In [84]: contribution.df <- data.frame(contribution)
```

```
In [124]: mosaic(~Class.Year+Marital.Status+Gender, data = contribution.df)
```



If we want to combine categorical and continous variable for plotting, one possible solution is to discretize continous variables. For this there is a nice function called `discretize_get_bins` and `discretize_df`:

```
In [97]: #library(funModeling)
         install.packages('funModeling',lib='.', verbose=TRUE)
         library(funModeling,lib.loc='.')
```

```
system (cmd0): /usr/lib/R/bin/R CMD INSTALL
also installing the dependencies pander, entropy
```

```
foundpkgs: pander, entropy, funModeling, /tmp/Rtmp2mtby9/downloaded_packages/pander_0.6.3.tar.gz
files: /tmp/Rtmp2mtby9/downloaded_packages/pander_0.6.3.tar.gz,
        /tmp/Rtmp2mtby9/downloaded_packages/entropy_1.2.1.tar.gz,
        /tmp/Rtmp2mtby9/downloaded_packages/funModeling_1.6.8.tar.gz
1): succeeded '/usr/lib/R/bin/R CMD INSTALL -l '/srv/home/whtam4' /tmp/Rtmp2mtby9/downloaded_pac
2): succeeded '/usr/lib/R/bin/R CMD INSTALL -l '/srv/home/whtam4' /tmp/Rtmp2mtby9/downloaded_pac
3): succeeded '/usr/lib/R/bin/R CMD INSTALL -l '/srv/home/whtam4' /tmp/Rtmp2mtby9/downloaded_pac
Loading required package: Hmisc
Loading required package: lattice
Loading required package: survival
Loading required package: Formula

Attaching package: Hmisc

The following objects are masked from package:dplyr:

    src, summarize

The following object is masked from package:plotly:

    subplot

The following objects are masked from package:base:

    format.pval, units

funModeling v.1.6.8 :)
Examples and tutorials at livebook.datascienceheroes.com


Attaching package: funModeling

The following object is masked from package:GGally:

    range01
```



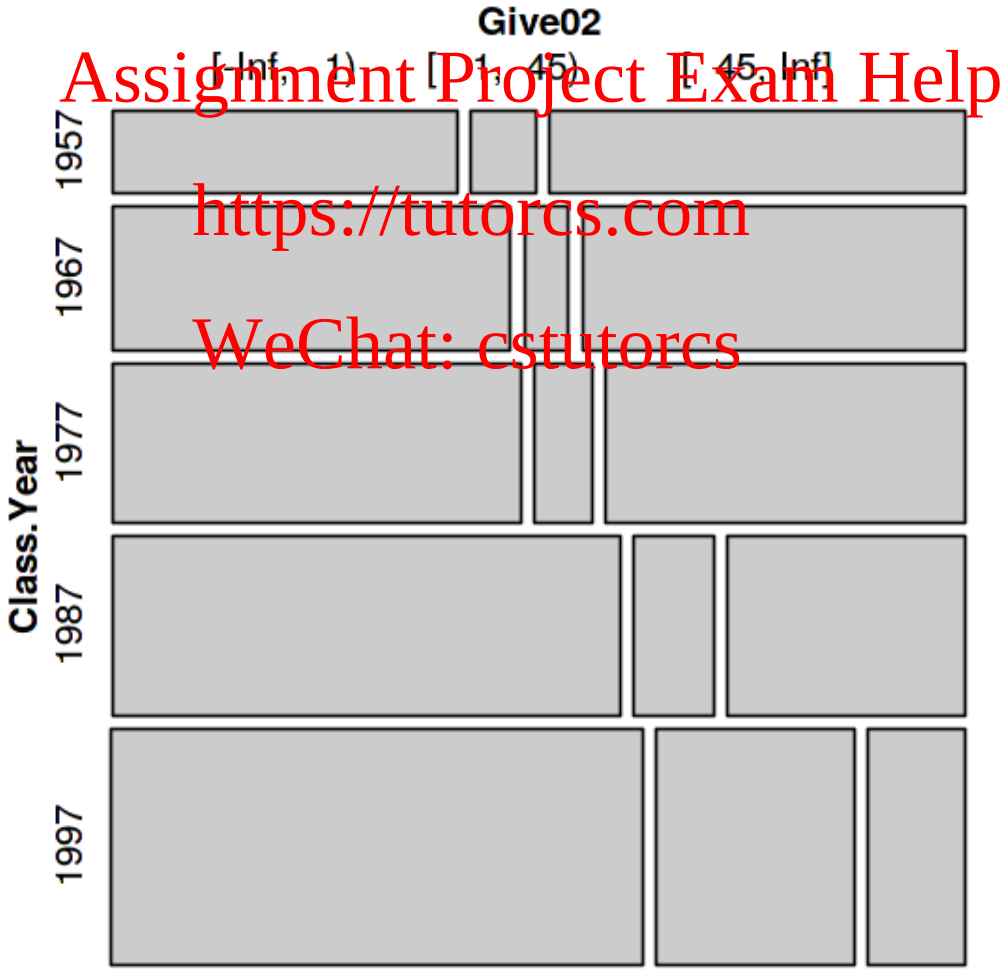
```
In [101]: d.bins <- discretize_get_bins(data = contribution.df,
                                input = c('Give04','Give03','Give02','Give01','Give00'),
                                n_bins=3)
          d.bins
           alumniDisc.df=discretize_df(data=contribution.df, data_bins=d.bins, stringsAsFactors=
          head(alumniDisc.df)
          mosaic(~Class.Year+Give02, data = alumniDisc.df)

[1] "Variables processed: Give04, Give03, Give02, Give01, Give00"
```

variable	cuts
Give04	5 | 35 | Inf
Give03	5 | 55 | Inf
Give02	1 | 45 | Inf
Give01	5 | 51 | Inf
Give00	5 | 41 | Inf

```
[1] "Variables processed: Give04, Give03, Give02, Give01, Give00"
```

Gender	Class.Year	Marital.Status	Major	Next.Degree	Give04	Give03	Give02	Give0
M	1957	M	History	LLB	[35, Inf]	[55, Inf]	[45, Inf]	[51, I
M	1957	M	Physics	MS	[35, Inf]	[55, Inf]	[45, Inf]	[51, I
F	1957	M	Music	NONE	[35, Inf]	[55, Inf]	[45, Inf]	[51, I
M	1957	M	History	NONE	[-Inf, 5)	[55, Inf]	[45, Inf]	[51, I
M	1957	M	Biology	MD	[35, Inf]	[55, Inf]	[45, Inf]	[51, I
F	1957	M	Mathematics	NONE	[-Inf, 5)	[-Inf, 5)	[-Inf, 1)	[-Inf,



1.0.3 Excercise 3: Experiment

Try to discretize and create mosaic plots out of the datassets `countries` and `mtcars` we have used so far. What do you find? Discuss this with your table.

```
In [108]: countries <-read.table(file = "CountryMatching.csv",
                                 header = TRUE,
                                 sep = ",")
          countries
```

Name	RegionNumber	RegionName
Algeria	1	Africa
Angola	1	Africa
Benin	1	Africa
Botswana	1	Africa
Burkina Faso	1	Africa
Burundi	1	Africa
Cameroon	1	Africa
Cape Verde	1	Africa
Central African Republic	1	Africa
Chad	1	Africa
Comoros	1	Africa
Congo, Dem. Rep.	1	Africa
Congo, Rep.	1	Africa
Cote d'Ivoire	1	Africa
Djibouti	1	Africa
Egypt, Arab Rep.	1	Africa
Equatorial Guinea	1	Africa
Eritrea	1	Africa
Ethiopia	1	Africa
Gabon	1	Africa
Gambia, The	1	Africa
Ghana	1	Africa
Guinea	1	Africa
Guinea-Bissau	1	Africa
Kenya	1	Africa
Lesotho	1	Africa
Liberia	1	Africa
Libya	1	Africa
Madagascar	1	Africa
Malawi	1	Africa
Lithuania	5	FSU
Moldova	5	FSU
Russian Federation	5	FSU
Tajikistan	5	FSU
Turkmenistan	5	FSU
Ukraine	5	FSU
Uzbekistan	5	FSU
Bahrain	6	Middle East
Iran, Islamic Rep.	6	Middle East
Iraq	6	Middle East
Israel	6	Middle East
Jordan	6	Middle East
Kuwait	6	Middle East
Lebanon	6	Middle East
Oman	6	Middle East
Qatar	6	Middle East
Saudi Arabia	6	Middle East
Syrian Arab Republic	6	Middle East
Turkey	6	Middle East
United Arab Emirates	6	Middle East
Yemen, Rep.	6	Middle East

```
In [ ]: countries.df <- data.frame (countries)
```

```
In [ ]: mosaic(~Name+RegionName, data = countries.df)
```