Worksheet 1

Introduction to R for MATH414

You can run R on Google Colab or locally on your machine. We begin by installing a few packages that we will use. The first one is fastR2 which was created by Randall Pruim.

```
install.packages("fastR2",repos = "http://cran.us.r-project.org")
```

The following command loads the fastR2 package.

```
require(fastR2)
```

The following gives a preview of the dataset iris.

```
glimpse(iris)
```

These are some miscellaneous commands to try with datasets.

```
head(iris, n = 3)
tail(iris, n = 3)
iris[50:51, 3:5]
sample(iris, 6)
#' this requires mosaic package which was loaded with fastR2,
#' if you don't load fastR2, you should load  mosaic to use this.
```

One can access an entire column of the dataset by using $ or using with

```
iris$Sepal.Length
```

```
with(iris,Species)
```

Exercise 1: Try the commands above with a different dataset such as "mtcars".

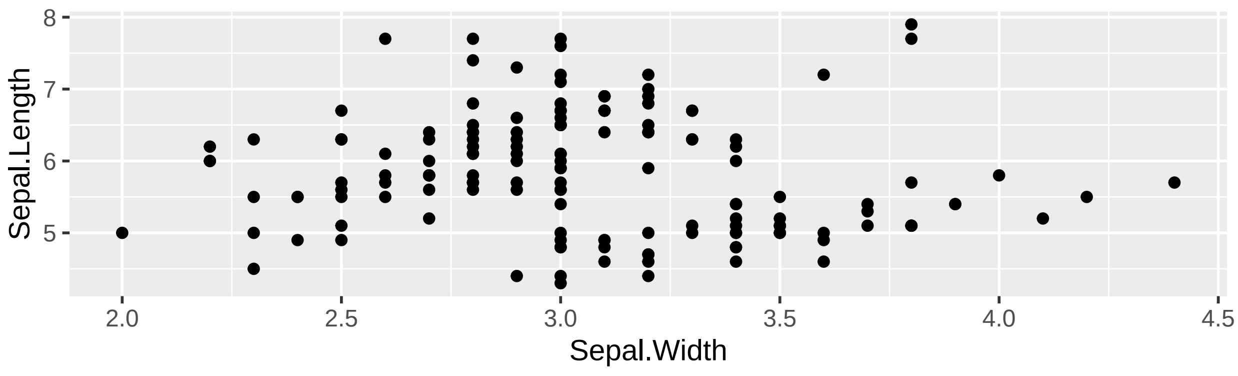
A Formula Template

We will make use of lattice, ggformula and mosaic packages to typeset most commands into a formula template which looks like "goal(formula, data = mydata)". One would need to specify what these goals, formulae, and data would be. One can alternatively use ggplot2 which has a different interface.

Scatterplot

The following command creates a scatterplot in R using ggformula.

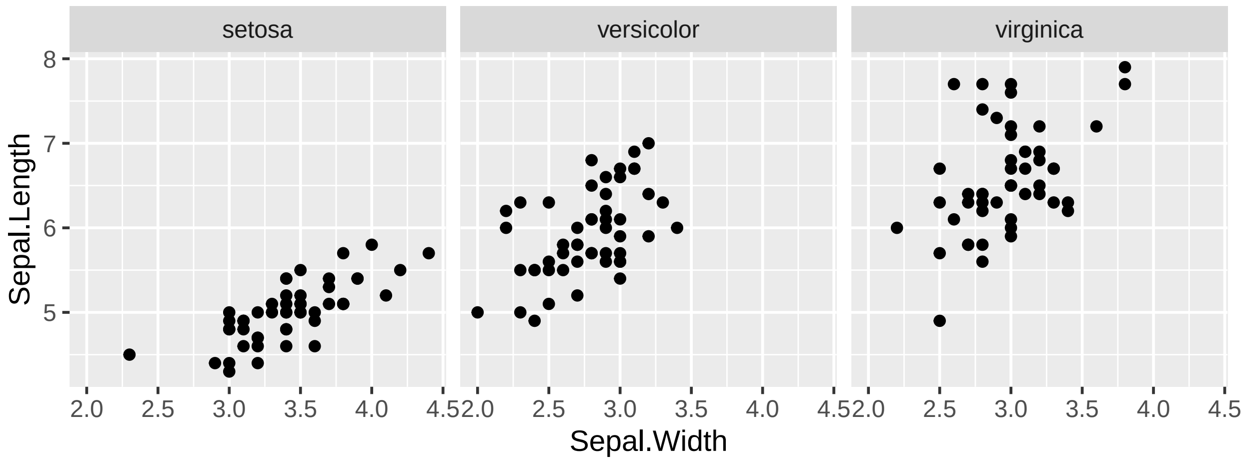
```
gf_point(Sepal.Length ~ Sepal.Width, data = iris)
```



The following are two alternative ways of creating scatterplots for each species of iris separately

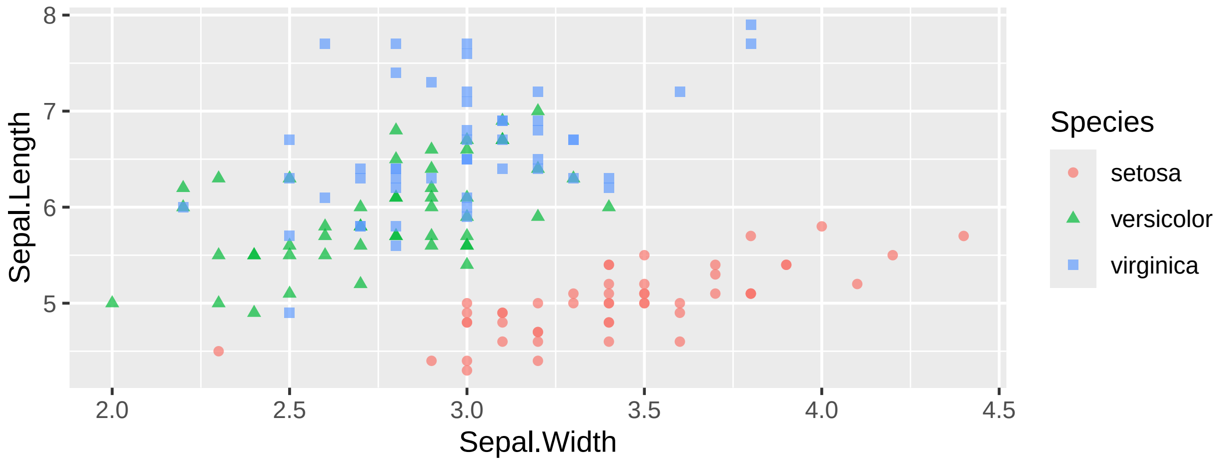
```
gf_point(Sepal.Length ~ Sepal.Width, data = iris) %>%
gf_facet_wrap( ~ Species)
```

```
gf_point(Sepal.Length ~ Sepal.Width | Species, data = iris)
```



You can get different colours for different species like this.

```
gf_point(Sepal.Length ~ Sepal.Width, data = iris,
color = ~ Species, shape = ~ Species, alpha = 0.7)
```



Exercise 2: Try the commands above with a different dataset such as "mtcars". In particular, draw a scatterplot for horsepower and weight. Use the ? command to find out what each column

stands for.

Tabulation and Histograms

Categorical data such as Species in iris dataset can be tabulated to note down its frequency as follows.

```
tally( ~ Species, data = iris)
```

```
## Species
##     setosa versicolor  virginica
##         50         50         50
```

On the other hand, quantitative variables like Sepal.length can be tabulated using frequency tables using the cut command.

```
tally( ~ cut(Sepal.Length, breaks = 2:10), data = iris)
```

```
## cut(Sepal.Length, breaks = 2:10)
##  (2,3]  (3,4]  (4,5]  (5,6]  (6,7]  (7,8]  (8,9] (9,10]
##      0      0     32     57     49     12      0      0
```

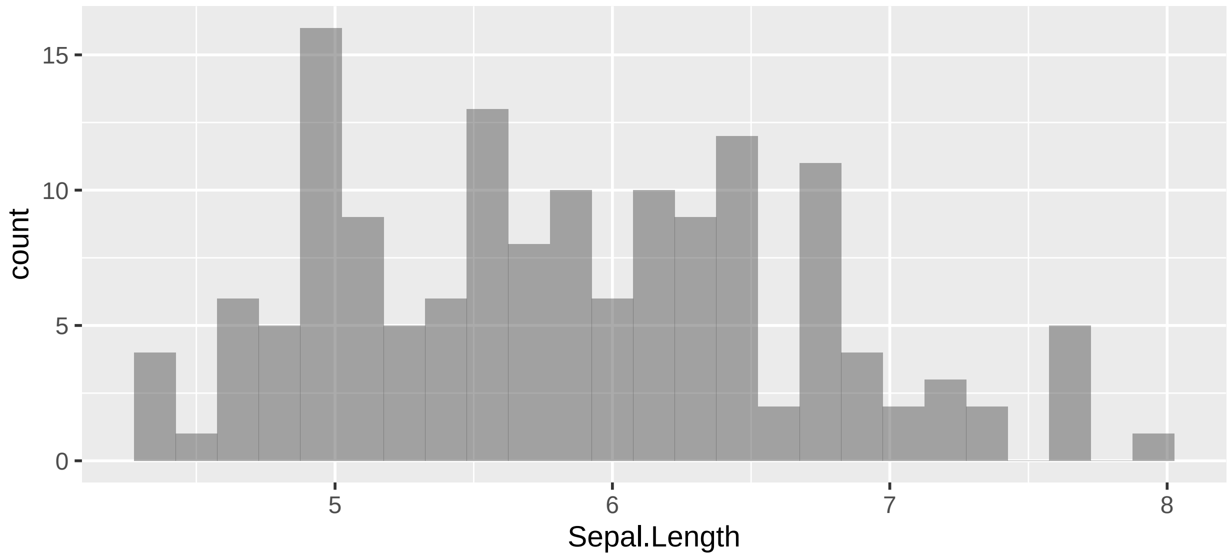
Alteratively, one can specify the number of bins

```
tally( ~ cut(Sepal.Length, 10), data = iris)
```

```
## cut(Sepal.Length, 10)
##  (4.3,4.66] (4.66,5.02] (5.02,5.38] (5.38,5.74]  (5.74,6.1]  (6.1,6.46]
##           9          23          14          27          22          20
## (6.46,6.82] (6.82,7.18] (7.18,7.54]  (7.54,7.9]
##          18           6           5           6
```

These frequency tables can be given the graphical form of a histogram as follows:

```
gf_histogram( ~ Sepal.Length, data = iris)
```



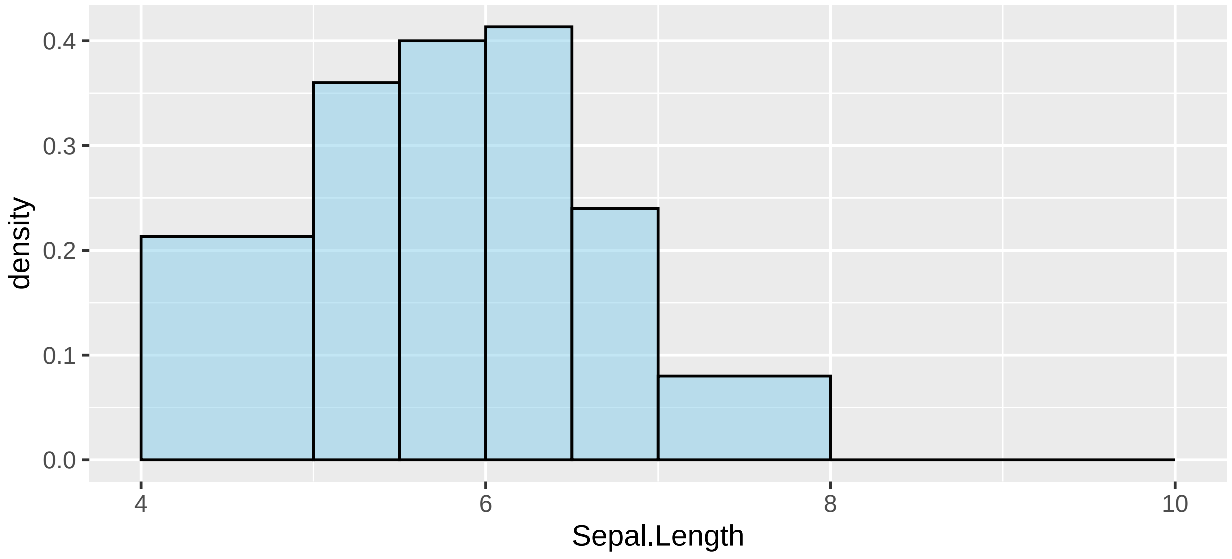
One can specify bins in the following ways:

```
# manually selecting width of bins
gf_histogram( ~ Sepal.Length, data = iris, binwidth = 0.5)
# also selecting the boundary of the bins
gf_histogram( ~ Sepal.Length, data = iris, binwidth = 0.5, boundary = 8)
```

```r
# manually selecting number of bins
gf_histogram( ~ Sepal.Length, data = iris, bins = 20)
```

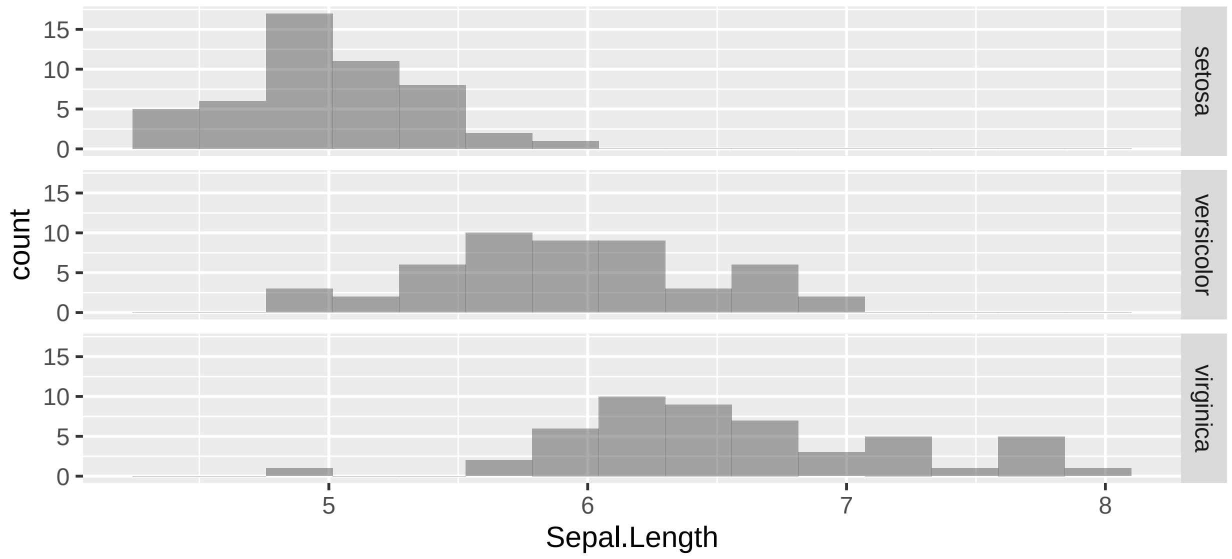
When the bins are not equally spaced, one should use density histograms.

```r
gf_dhistogram( ~ Sepal.Length, data = iris,
breaks = c(4, 5, 5.5, 6, 6.5, 7, 8, 10),
color = "black", fill = "skyblue")
```

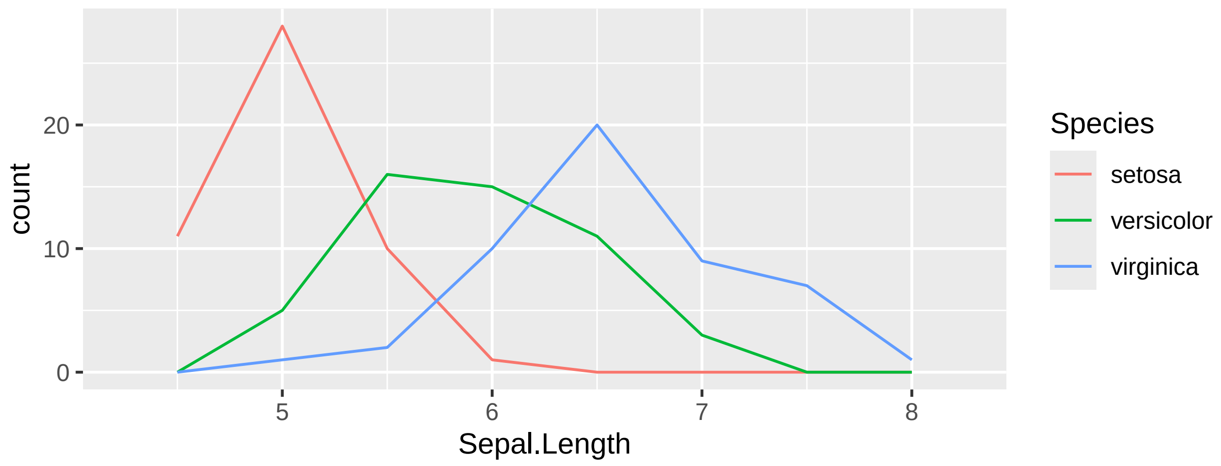


We might want to create three histograms, one for each species.

```r
gf_histogram( ~ Sepal.Length | Species ~ ., data = iris,
bins = 15)
```



```r
gf_freqpoly( ~ Sepal.Length, color = ~ Species, data = iris, binwidth = 0.5)
```



Exercise 3: Try the commands in this section with a different dataset such as "mtcars". In particular, draw histograms for horsepower, weight and condition it on transmission type. Use the ? command to find out what each column stands for.

Descriptive Statistics

We can find out mean, median, etc using R.

```
mean( ~ Sepal.Length, data = iris)
```

```
## [1] 5.843333
```

```
median( ~ Sepal.Length, data = iris )
```

```
## [1] 5.8
```

They can be grouped by species

```
mean(Sepal.Length ~ Species, data = iris)
```

```
##     setosa versicolor  virginica
##      5.006      5.936      6.588
```

```
df_stats(Sepal.Length ~ Species, data = iris, mean, median)
```

```
##       response    Species  mean median
## 1 Sepal.Length     setosa 5.006    5.0
## 2 Sepal.Length versicolor 5.936    5.9
## 3 Sepal.Length  virginica 6.588    6.5
```

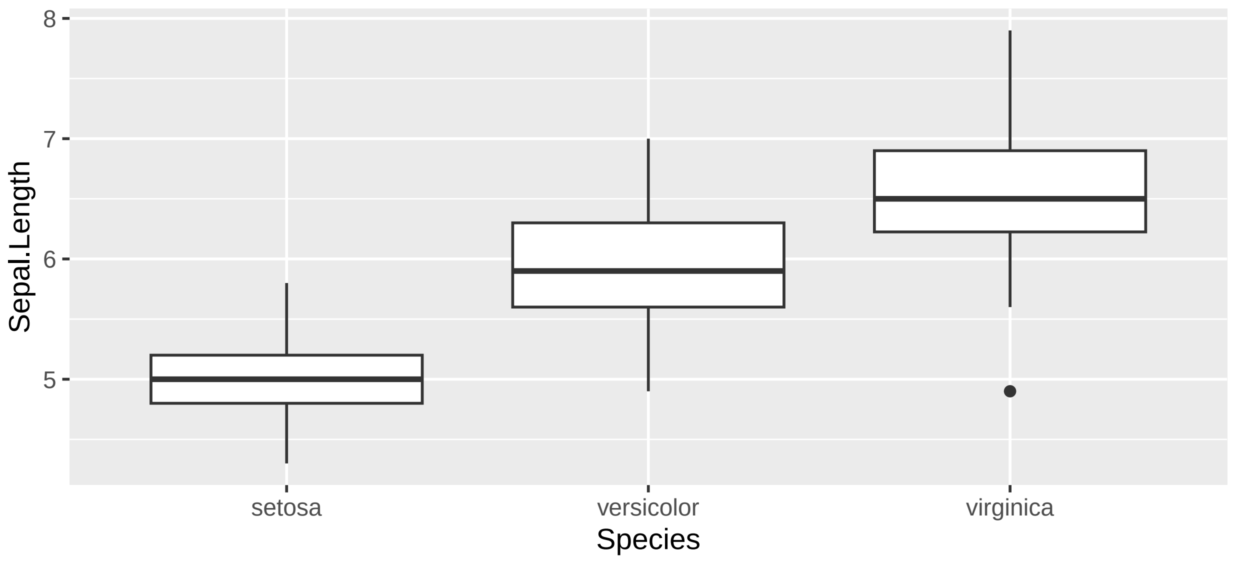
p-quantile is the x-value that returns x when p ratio of data is below x. In particular, median is the 0.5-quantile.

```
quantile(~ Sepal.Length | Species, data=iris)
```

```
##      Species  0%   25% 50% 75% 100%
## 1     setosa 4.3 4.800 5.0 5.2  5.8
## 2 versicolor 4.9 5.600 5.9 6.3  7.0
## 3  virginica 4.9 6.225 6.5 6.9  7.9
```

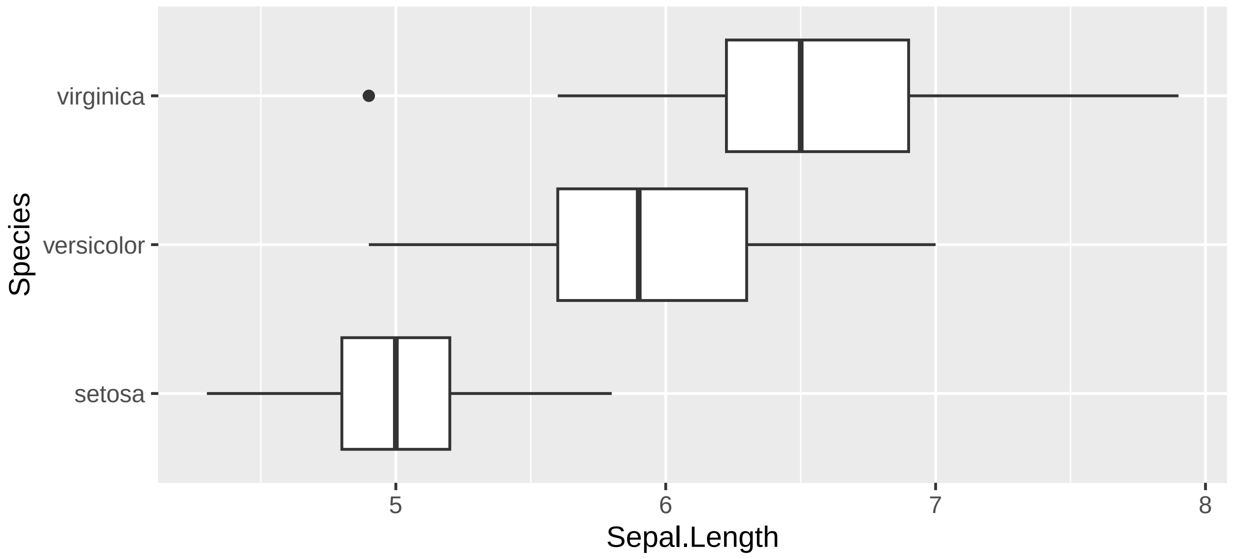
Quantiles are graphically represented by boxplots

```
gf_boxplot(Sepal.Length ~ Species, data = iris)
```



We can draw them horizontally too

```
gf_boxplot(Sepal.Length ~ Species, data = iris) %>%
gf_refine(coord_flip())
```



We can also get standard deviation and variance

```
var(Sepal.Length ~ Species, data = iris)
```

```
##     setosa versicolor  virginica
##  0.1242490  0.2664327  0.4043429
```

```
sd(Sepal.Length ~ Species, data = iris)
```

```
##     setosa versicolor  virginica
##  0.3524897  0.5161711  0.6358796
```

```
favstats(Sepal.Length ~ Species, data = iris)
```

```
##      Species min    Q1 median  Q3 max  mean        sd  n missing
## 1     setosa 4.3 4.800    5.0 5.2 5.8 5.006 0.3524897 50       0
## 2 versicolor 4.9 5.600    5.9 6.3 7.0 5.936 0.5161711 50       0
## 3  virginica 4.9 6.225    6.5 6.9 7.9 6.588 0.6358796 50       0
```

> **Exercise 4**: Try the commands in this section with a different dataset such as "mtcars". In particular, summarize the data and draw boxplots for horsepower, weight and condition it on transmission type. Use the ? command to find out what each column stands for.

Two way tables and mosaic plots

DeathPenalty is a dataset recording death sentences and race of defendant and that of the victim. These are categorical variables. We can tabulate the death penalties for black and white defendants and we can differentiate these by race of victims as follows

```
tally(death ~ defendant | victim, data = DeathPenalty)
```

```
## , , victim = Bl
##
##      defendant
## death  Bl  Wh
##   No   97   9
##   Yes   6   0
##
## , , victim = Wh
##
##      defendant
## death  Bl  Wh
##   No   52 132
##   Yes  11  19
```

An analysis of this data reveals that black people receive death penalty more often as compared to white people. This is not evident when we simply look at the death penalty convictions without differentiating the race of victims.

```
tally(death ~ defendant , data = DeathPenalty)
```

```
##      defendant
## death  Bl  Wh
##   No  149 141
##   Yes  17  19
```

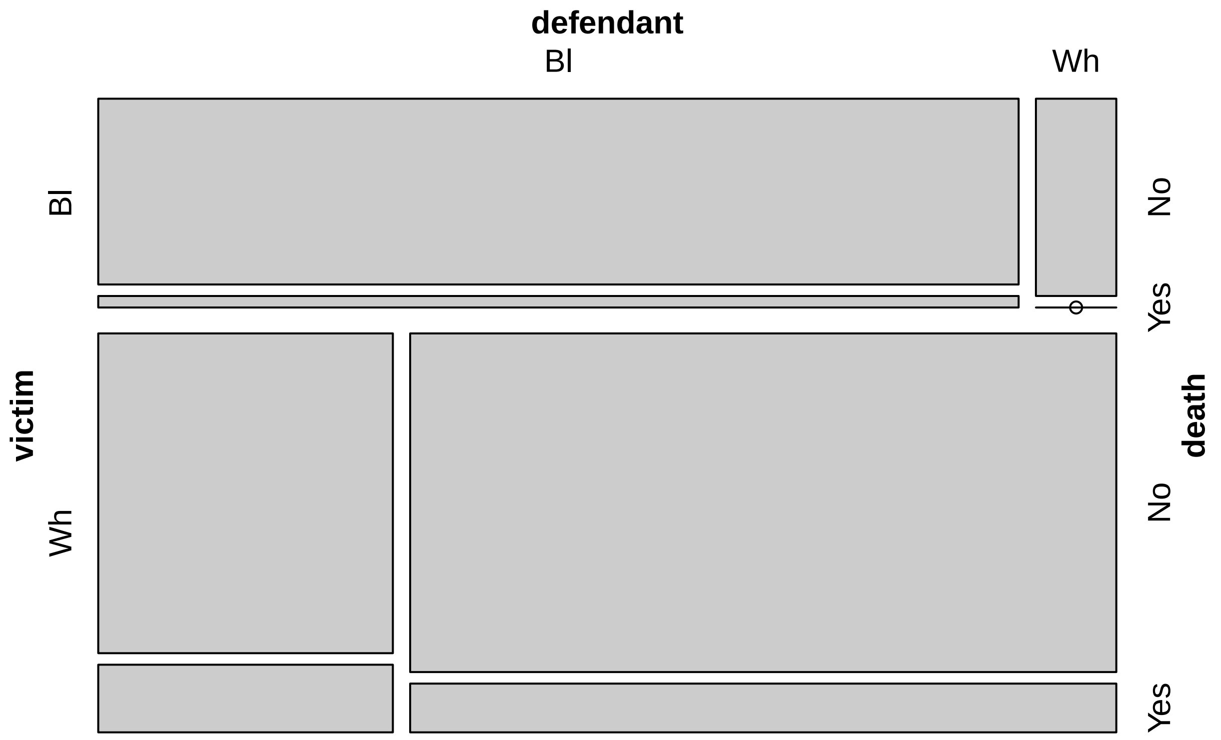
This is known as **Simpson's paradox** where an incorrect conclusion gets drawn due to a hidden or lurking variable, in this case, the race of the victim.

This can be visualized using a mosaic plot which requires the vcd package.

```
install.packages("vcd") #You may have to install fortran and lmtest package before this.
```

```
require(vcd)
```

```
vcd::mosaic( ~ victim + defendant + death, data = DeathPenalty)
```



Exercises

Exercise 5: Try the commands in this section with a different dataset such as "Titanic". In particular, draw mosaicplots for Sex, Class and Survived. Use the ? command to find out what each column stands for.

Exercise 6: The pulse variable in the LittleSurvey data set contains self-reported pulse rates.

a) Make a histogram of these values. What problem does this histogram reveal?

b) Make a decision about what values should be removed from the data and make a histogram of the remaining values. (You can use filter() to create a subset of the data frame and make a histogram from that.)

c) Compute the mean and median of your restricted set of pulse rates.

Exercise 7: Some students in introductory statistics courses were asked to select a number between 1 and 30 (inclusive). The results are in the number variable in the LittleSurvey data set.

a) Make a table showing the frequency with which each number was selected using tally().

b) Make a histogram of these values with bins centered at the integers from 1 to 30.

c) What numbers were most frequently chosen? Can you get R to find them for you?

d) What numbers were least frequently chosen? Can you get R to find them for you?

e) Make a table showing how many students selected odd versus even numbers.

References

1. Randall Pruim, Foundations and Applications of Statistics, An Introduction Using R, Second Edition, American Mathematical Society, 2018