# MAST7866 Linear Regression

Computing Session 1

For this session, you will need to download the following files from the MAST7866 Moodle page: `production.txt`, `SaltBP.txt`, `diamonds.txt`.

It would be useful to create a MAST7866 directory which contains all the data that you use in this part of the module. This directory can be set to be your working directory in RStudio by choosing `Choose Directory...` in the `Set Working Directory` tab under `Session` and selecting your MAST7866 directory.

## 1 Production runs data

The production runs data discussed in the lectures is contained in the text file `production.txt`. To load the data, assuming that you have copied it to your MAST7866 directory and made this your working directory, by typing:

```
production <- read.table("production.txt", header = T)
names(production)
```

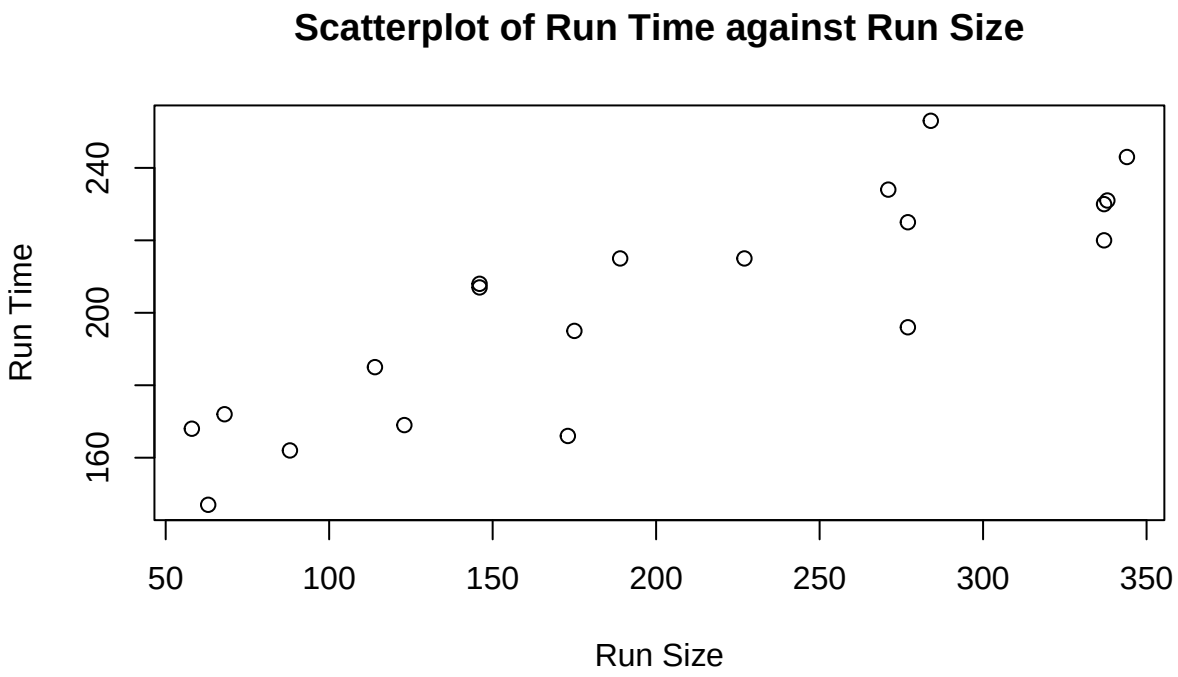
```
[1] "Case"    "RunTime" "RunSize"
```

Note: the `header = T` command tells RStudio to use the first row of the data file as the names of each variable/column, so the names of the variables are Case, RunTime, and RunSize.

This command stores the data in a data frame object called production. You can view the data by typing the object name. You can access all variables inside the data by using the `$` together with the name of the object, as we will see below.

Initially, it is useful to draw a scatterplot of the data which can be drawn using the `plot` function. The graph can be made tidier by including a title and giving the axes proper names.

```
plot(production$RunSize, production$RunTime,
     xlab = "Run Size", ylab = "Run Time",
     main = "Scatterplot of Run Time against Run Size")
```




A simple linear regression model can be fitted to the data by typing

```
fit_production <- lm(RunTime ~ RunSize, data = production)
```

The function is the `lm` function. This has two arguments:

- The model equation, which is `RunTime ~ RunSize`. This is translated as the regression model RunTime $= \alpha + \beta$ RunSize + e. There are a couple of things to note: `R` will automatically include an intercept term ($\alpha$), and the names of the variables in the data frame are used in the regression equation.
- The second argument `data` tells R which data frame to use. The output of the `lm` function is stored in the object `fit_production`. This is just a name that I have given it and you could use any name (although, an informative one is often useful!).

Typing

```
fit_production ##yields the following output
```

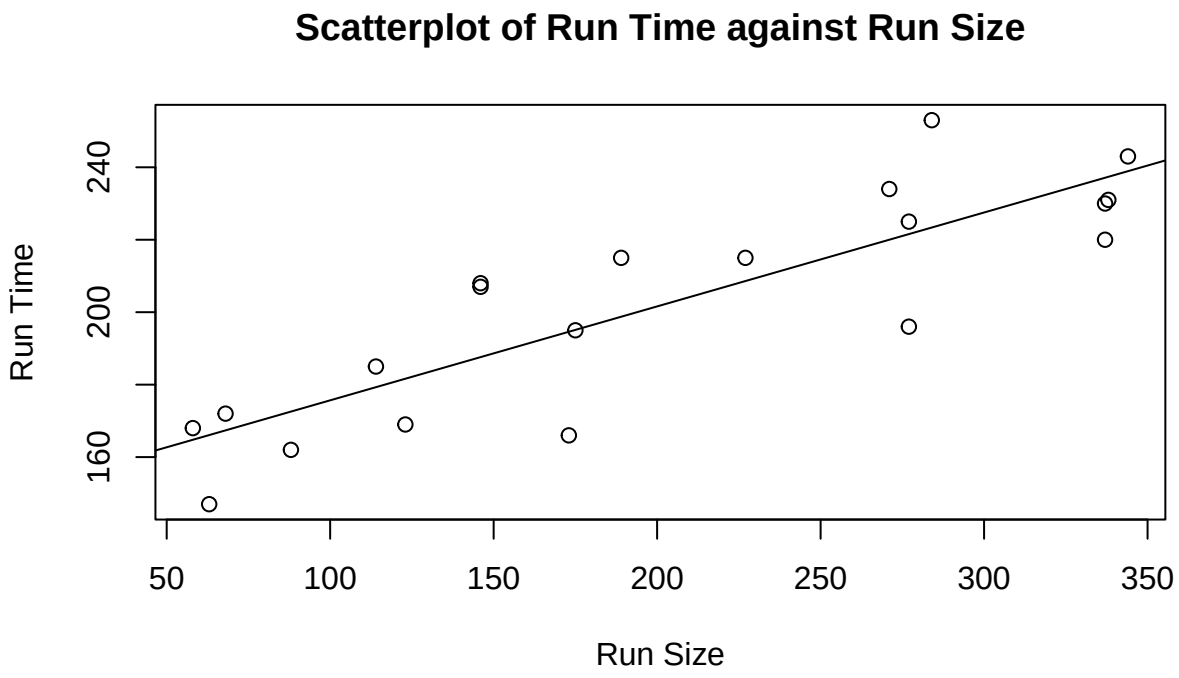
```
Call:
lm(formula = RunTime ~ RunSize, data = production)

Coefficients:
(Intercept)      RunSize
   149.7477       0.2592
```

This calls the `lm` object that you have created before with the parameter estimates which are $\hat{\alpha}$ =149.7477 and $\hat{\beta} = 0.2592$.

A fitted regression line can be added to the scatterplot of Run Time against Run Size by typing

```
plot(production$RunSize, production$RunTime,
     xlab = "Run Size", ylab = "Run Time",
```

```
     main = "Scatterplot of Run Time against Run Size")
abline(fit_production)
```




RStudio will provide more information from fitting the model if you type

```
summary(fit_production) ##This produces the output below
```

```
Call:
lm(formula = RunTime ~ RunSize, data = production)

Residuals:
    Min      1Q  Median      3Q     Max
-28.597 -11.079   3.329   8.302  29.627

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 149.74770    8.32815   17.98 6.00e-13 ***
RunSize       0.25924    0.03714    6.98 1.61e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.25 on 18 degrees of freedom
Multiple R-squared:  0.7302,    Adjusted R-squared:  0.7152
F-statistic: 48.72 on 1 and 18 DF,  p-value: 1.615e-06
```

The meaning of a lot of this output will become clearer as the module progresses. One piece of information that we want at the moment is the estimated error standard deviation $\hat{\sigma}$. RStudio gives the *estimated error standard deviation* as **Residual standard error** which is 16.25 for these data.

## 2 The effect of salt on blood pressure

The file `saltBP.txt` contains measurements on 25 elderly people. The variables measured are

- BP – denotes the systolic blood pressure
- salt – the average daily intake of salt in grams.

```
saltBP <- read.table("saltBP.txt", header = T)
names(saltBP)
```

```
[1] "BP"   "salt"
```

**1. Draw a scatterplot of BP against salt**

Is a linear regression model suitable?

**2. Fit a linear regression model to predict the systolic blood pressure from the daily intake of salt.**

What is the fitted regression equation?

**3. Use the fitted regression model to answer the following questions:**

**3(a) What is the effect on blood pressure of increasing salt intake by one gram?**

**3(b) What is the average blood pressure of someone who has a salt intake of six grams per day?**

**3(c) I currently consume nine grams of salt per day and I want to reduce my blood pressure by 5.**

By how much should I reduce my average daily salt intake?

## 3 Pricing diamond rings

This example looks at developing a regression model to predict the price of diamond rings from the size of their diamond stones (in carats). The data was taken from an advert placed in the Straits Times newspaper.

```
diamonds <- read.table("diamonds.txt", header=T)
names(diamonds)
```

```
## [1] "Size"  "Price"
```

**1. Plot a suitable graph of the data.**

**2. Fit a linear regression model to predict the price of a diamond ring from the size of the diamond stone. What is the fitted regression equation?**

**3. What mean price does your model estimate for a diamond ring with 0.25 carats? What mean price does your model estimate for a diamond ring with 0.15 carats?**

**4. What is the estimated error standard deviation, $\hat{\sigma}$?**

**5. Are there any weaknesses to the model that you have fitted?**