Data Science and Statistics in Research: unlocking the power of your data

Session 3.3: Fitting a model to your data

Introduction

In this session we will do some linear and multiple regression in `R`.

Preliminaries

We need the following package

- `ggplot2` - Package to implement the ggplot language for graphics in `R`.

Make sure that this package is downloaded and installed in `R`. We use the `require()` function to load it into the `R` library.

```
# Loading packages
require(ggplot2)
```

Data

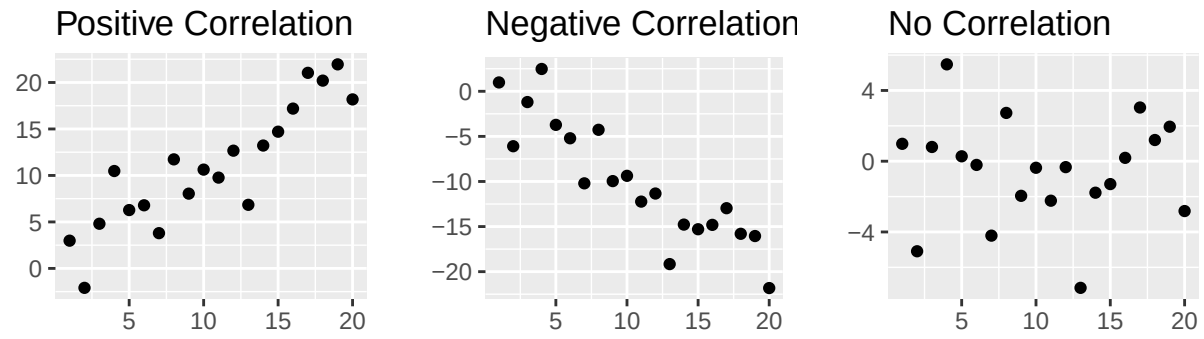
To demonstrate how to perform linear and multiple regression in `R` we will use the `mtcars` dataset. This dataset consists of fuel consumption and other aspects of automobile design and performance for 32 cars from the 1973-74 Motor Trend US magazine. We load this dataset, using the `data()` function.

```
# Loading mtcars dataset
data(mtcars)
```

Make sure you are familiar with the contents of this dataset before continuing on with the rest of this practical, by typing `?mtcars` into `R`.

Correlation

Correlation is a measure of the association between two variables. The correlation coefficient quantifies the strength of any association. It takes values between -1 and +1. Values between 0 and +1 indicate a positive association, (as one variable increases the other increases). Values between -1 and 0 indicate a negative association, (as one variable increases the other decreases). Values around 0 indicates no relationship.



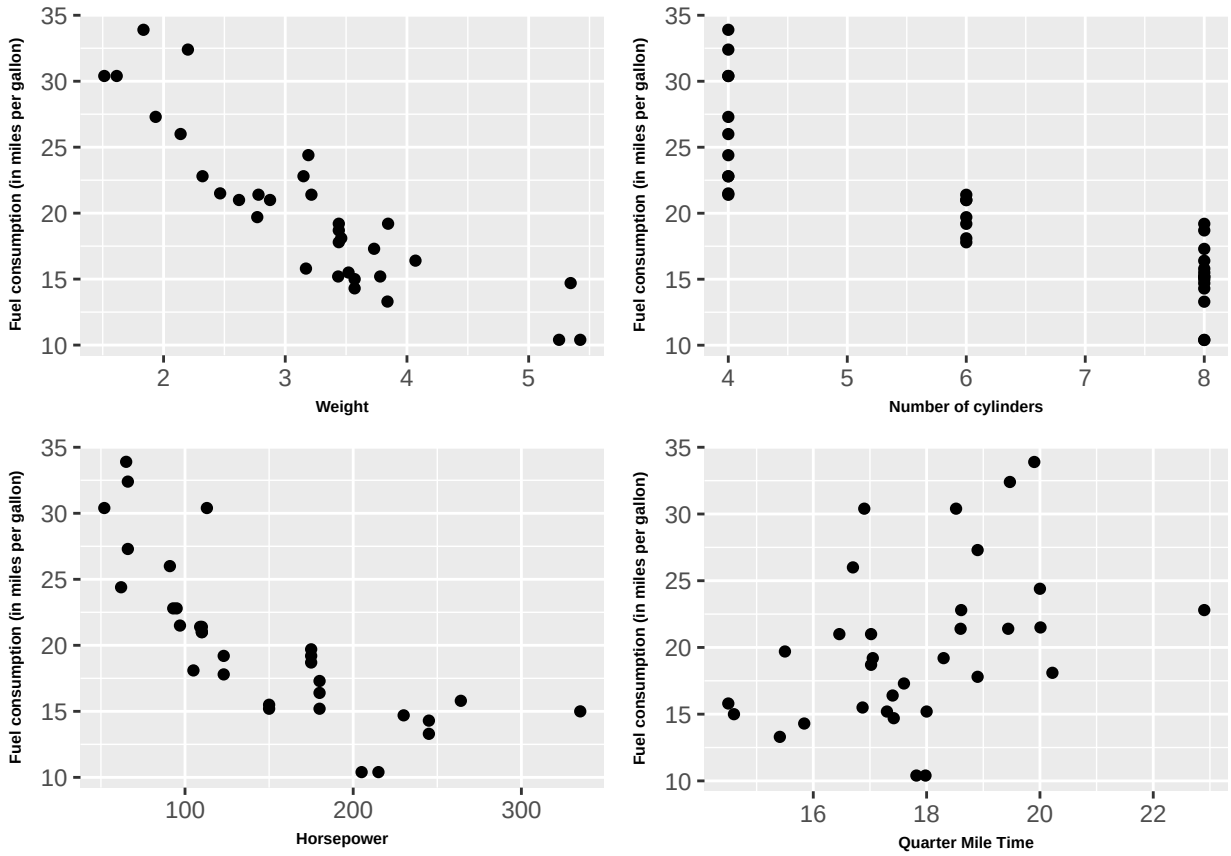
Suppose a car company are interested in checking for correlation between fuel consumption and weight, horsepower, number of cylinders and quarter mile time. Firstly, we can visually check for correlation by creating scatter plots. To create scatter plots we use the `ggplot2` package.

```
# Plotting fuel consumption against weight
ggplot(mtcars) + # ggplot with the desired data
  geom_point(aes(x=wt,y=mpg)) +
  labs(x='Weight',y='Fuel consumption (in miles per gallon)') # Labels

# Plotting fuel consumption against horsepower
ggplot(mtcars) + # ggplot with the desired data
  geom_point(aes(x=hp,y=mpg)) + # Specifying scatterplot
  labs(x='Horsepower',y='Fuel consumption (in miles per gallon)') # Labels

# Plotting fuel consumption against number of cylinders
ggplot(mtcars) + # ggplot with the desired data
  geom_point(aes(x=cyl,y=mpg)) + # Specifying scatterplot
  labs(x='Number of cylinders',y='Fuel consumption (in miles per gallon)') # Labels

# Plotting fuel consumption against quarter mile time
ggplot(mtcars) + # ggplot with the desired data
  geom_point(aes(x=qsec,y=mpg)) + # Specifying scatterplot
  labs(x='Quarter Mile Time',y='Fuel consumption (in miles per gallon)') # Labels
```



Information about the `ggplot2` package can be found in the 'Packages' pane.

We can also calculate correlation coefficients between these variables. To do this we use the `cor()` function.

```
# Calculating the correlation coefficient between fuel consumption
# weight, horsepower, number of cylinders and quarter mile time
cor(mtcars[,c('mpg','wt','hp','cyl','qsec')])
```

```
            mpg         wt         hp        cyl       qsec
mpg   1.0000000 -0.8676594 -0.7761684 -0.8521620  0.4186840
wt   -0.8676594  1.0000000  0.6587479  0.7824958 -0.1747159
hp   -0.7761684  0.6587479  1.0000000  0.8324475 -0.7082234
cyl  -0.8521620  0.7824958  0.8324475  1.0000000 -0.5912421
qsec  0.4186840 -0.1747159 -0.7082234 -0.5912421  1.0000000
```

More information about the `cor()` function can be found by typing `?cor` into `R`.

We can see that there is strong negative correlation between fuel consumption and weight, horsepower and number of cylinders. There is low positive correlation between fuel consumption and quarter mile time.

Linear Regression

Variables can be classified as explanatory or response.

- Response – we are interested in changes in the response, i.e. our variable of primary interest.
- Explanatory – variables that may explain changes the response variable.

Suppose we are interested in how fuel consumption (in miles per gallon) is affected by weight, number of cylinders and transmission. Fuel consumption is our response variable and with weight, number of cylinders

or transmission being the explanatory variables.

Linear regression allows us to analyse the relationship between two variables. The most straightforward relationship is a linear, or straight line, relationship. This involves fitting a straight line through the data points.

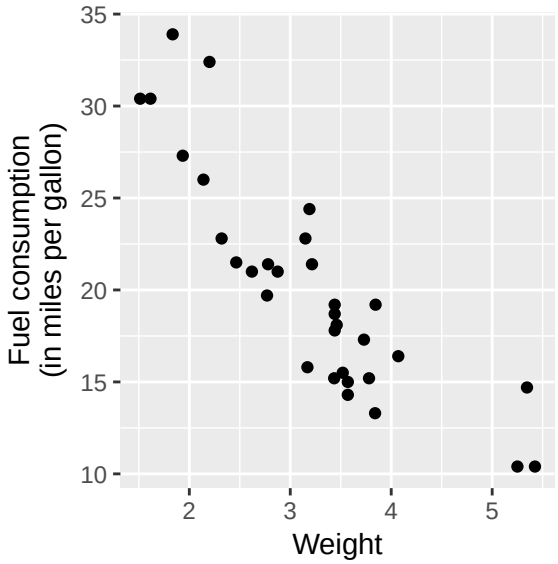
Straight lines are described by the formula

$$y = a + bx.$$

- y - the response variable.
- x - the explanatory variable.
- a is the intercept; the point at which the line cuts the y axis. It tells us what response we would expect if the explanatory variable was equal to zero.
- b is the slope of the line. It is the increase (or decrease) of the response variable per unit increase in the explanatory variable.

Suppose we are interested in how fuel consumption (in miles per gallon) is affected/associated by weight.Before attempting to fit a linear model, you should check if the relationship between the response and explanatory variable are linear. A scatter plot can be useful to assess this. The correlation coefficient indicates how well a straight line fits the data. As we are interested in the relationship between fuel consumption and weight we create a scatter plot containing both of these variables. To create scatter plots we use the `ggplot2` package.

```
# Plotting fuel consumption against weight
ggplot(mtcars) + # ggplot with the desired data
  geom_point(aes(x=wt,y=mpg)) +
  labs(x='Weight',y='Fuel consumption\n(in miles per gallon)')
```



Information about the `ggplot2` package can be found in the 'Packages' pane.

The correlation coefficient between weight and fuel consumption is -0.87. Using both of these, we can see that there is an association there and it looks like a straight line will represent this relationship well.

We describe the fuel consumption (mpg) of a car using the formula

$$mpg = a + b * weight.$$

To fit this model in `R` we use the `lm()` function.

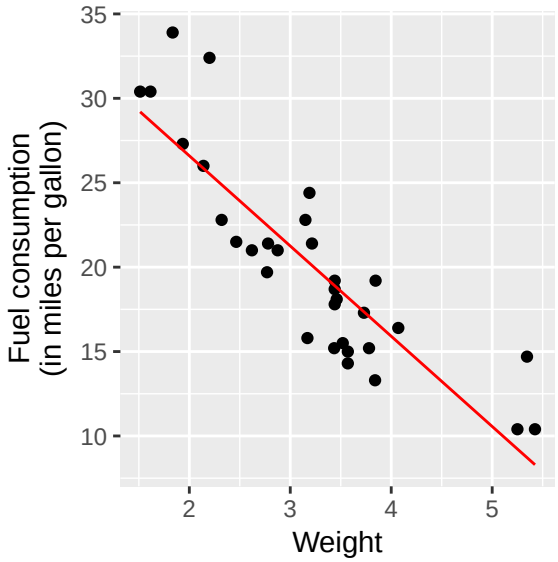
```
# Model formula mpg ~ 1 + wt (mpg = a + b * weight)
formula <- mpg ~ 1 + wt
```

```r
# Fitting the linear model
mod <- lm(formula, data=mtcars)
```

More information about the `lm()` function can be found by typing `lm()` into `R`.

Using `mod` we can plot the resulting straight line. To do this, we extract the values which are the miles per gallon we would expect given the particular weights in our dataset. We can then create the plot using the `ggplot2` package.

```r
# Plotting fuel consumption against weight
ggplot(mtcars, aes(x=wt,y=mpg)) + # ggplot with the desired data
  geom_point() +
  geom_line(aes(x=wt,y=fitted(mod)),colour='red',) +
  labs(x='Weight',y='Fuel consumption\n(in miles per gallon)')
```



Information about the `ggplot2` package can be found in the 'Packages' pane.

Outputs are summarised using estimates of the model coefficients together with standard errors. To summarise the model, we use the `summary()` function.

```r
# Summarising the fitted model
summary(mod)

Call:
lm(formula = formula, data = mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-4.5432 -2.3647 -0.1252  1.4096  6.8727

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  37.2851     1.8776  19.858  < 2e-16 ***
wt           -5.3445     0.5591  -9.559 1.29e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.046 on 30 degrees of freedom
```

```
Multiple R-squared:  0.7528,    Adjusted R-squared:  0.7446
F-statistic: 91.38 on 1 and 30 DF,  p-value: 1.294e-10
```

More information about the `summary()` function can be found by typing `summary()` into `R`.

Hypothesis tests can be used to check whether there is a significant relationship between the response and explanatory variables

- null: $b = 0$
- alternative: $b \neq 0$.

As default, `R` will test the hypothesis that if the true intercept and slope terms are greater than zero. The interest is usually in the effects of the explanatory variables rather than the intercept. So we construct the hypotheses

- **null**: there is no effect or association of weight on fuel consumption ($b = 0$)
- **alternative**: there is an effect or association of weight on fuel consumption ($b \neq 0$)

We choose a significance level of 0.05 and construct the statistical decision rule

- **IF** the p-value is less than 0.05
- **THEN** we have enough evidence to reject the null hypothesis
- **OTHERWISE** there is not enough evidence to reject the null hypothesis.

Using the summary above we see that we have a p-value of <0.0001. Therefore, there is a significant association between weight and fuel consumption.

We can also construct 95% confidence interval for the coefficient of weight using the `confint()` function.

```
# Calculating confidence intervals of model coefficients
confint(mod, # Linear Model fit
        level = 0.95) # Significance level
                2.5 %    97.5 %
(Intercept) 33.450500 41.119753
wt          -6.486308 -4.202635
```

More information about the `confint()` function can be found by typing `?confint` into `R`.

R^2 is a measure of how well the model fits the data. It indicates the proportion of variance of the response explained by the model. It takes values between 0 and 1, with 0 indicating the model does not explain changes in the response and 1 indicating a 'perfect' model. High values of R^2 indicate good model fit.

R^2 will always increase as more explanatory variables are added to the model and the adjusted R^2 penalises models with lots of parameters. Using the summary above we have a R^2 of 0.75 and an adjusted R^2 of 0.74. Therefore we have a pretty good fit using weight as an explanatory variable.

Once we have fitted a model, we can use it predict values of the responses for a set of values for the explanatory variables. This is done by plugging the values into the regression equation. Suppose we wanted to predict the fuel consumption of a car that weighs 3.5. We could use the `predict()` function to do this.

```
# Predicting for a weight of 3.5
predict(mod, # Fitted model
        newdata = data.frame(mpg=NA, wt=3.5)) # Data that we want to predict for
       1
18.57948
```

Therefore we see that the fuel consumption of a car that weighs 3.5 will be `predict(mod,newdata=data.frame(mpg=NA, wt=3.5))` miles per gallon.

Multiple Regression

Multiple regression is a natural extension of the linear regression model. It is used to predict values of a response from several explanatory variables. Each explanatory variable has its own coefficient. The response variable is predicted from a combination of all the variables multiplied by their respective coefficients.

Multiple regressions are described by the formula

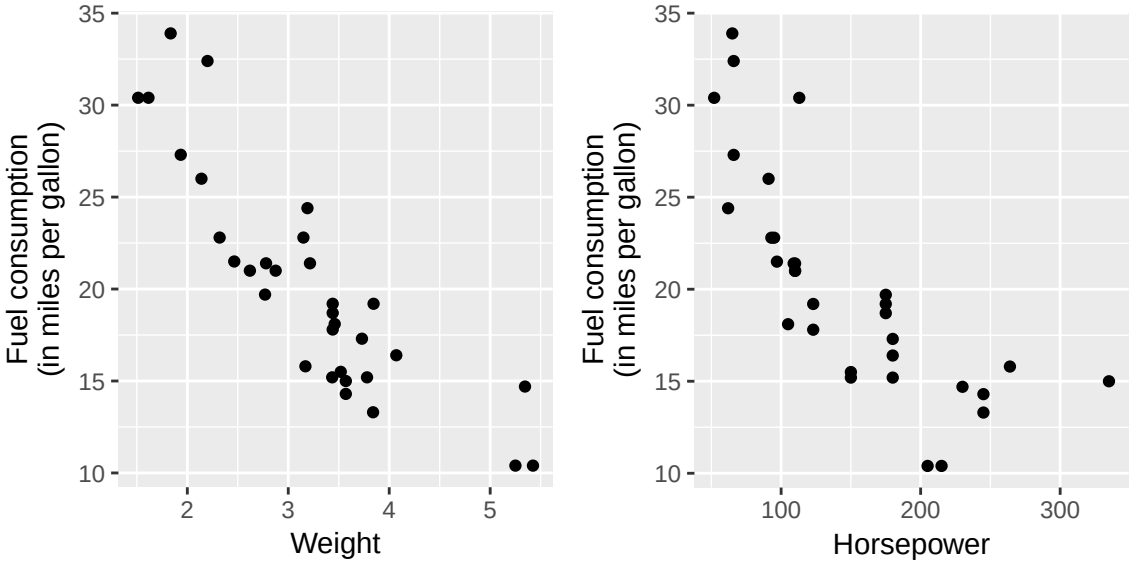
$$y = a + bx_1 + cx_2 + dx_3 + ...$$

* y - response variable. * $x_1, ..., x_n$ - explanatory variables. * a is the intercept; the point at which the line cuts the y axis. It tells us what response we would expect if the explanatory variable was equal to zero. * $b, c, d...$ are the coefficients of the i^{th} explanatory variables. It is the increase (or decrease) of the response variable per unit increase in the i^{th} explanatory variable.

Suppose we are interested in how fuel consumption (in miles per gallon) is affected/associated by weight and horsepower. Before attempting to fit the linear model, we check to see if the relationship between the response and explanatory variables are linear. As we are interested in the relationships between fuel consumption and weight and fuel consumption and horsepower we create a scatter plot of these. To create scatter plots we use the `ggplot2` package.

```
# Plotting fuel consumption against weight
ggplot(mtcars) + # ggplot with the desired data
  geom_point(aes(x=wt,y=mpg)) +
  labs(x='Weight',y='Fuel consumption\n(in miles per gallon)')

# Plotting fuel consumption against weight
ggplot(mtcars) + # ggplot with the desired data
  geom_point(aes(x=hp,y=mpg)) +
  labs(x='Horsepower',y='Fuel consumption\n(in miles per gallon)')
```



Information about the `ggplot2` package can be found in the 'Packages' pane.

We describe the fuel consumption (mpg) of a car using the formula

$$mpg = a + b * weight + c * horsepower.$$

To fit this model in `R` we use the `lm()` function.

```r
# Model formula mpg ~ 1 + wt (mpg = a + b * weight)
formula <- mpg ~ 1 + wt + hp

# Fitting the linear model
mod <- lm(formula, data=mtcars)
```

More information about the `lm()` function can be found by typing `lm()` into `R`.

In multiple regression, it is much more difficult to view the fitted line as we are in higher dimension. So we can just summarise the outputs, using estimates of the model coefficients together with standard errors. To summarise the model, we use the `summary()` function.

```r
# Summarising the fitted model
summary(mod)

Call:
lm(formula = formula, data = mtcars)

Residuals:
   Min     1Q Median     3Q    Max
-3.941 -1.600 -0.182  1.050  5.854

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 37.22727    1.59879  23.285  < 2e-16 ***
wt          -3.87783    0.63273  -6.129 1.12e-06 ***
hp          -0.03177    0.00903  -3.519  0.00145 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.593 on 29 degrees of freedom
Multiple R-squared:  0.8268,    Adjusted R-squared:  0.8148
F-statistic: 69.21 on 2 and 29 DF,  p-value: 9.109e-12
```

More information about the `summary()` function can be found by typing `summary()` into `R`.

We use hypothesis tests to check whether there is a significant relationship between the response and explanatory variables. We construct the hypotheses

- **null**: there is no effect or association of weight (or horsepower) on fuel consumption ($b = 0$ or $c = 0$)
- **alternative**: there is an effect or association of weight (or horsepower) on fuel consumption ($b \neq 0$ or $c \neq 0$)

We choose a significance level of 0.05 and construct the statistical decision rule

- **IF** the p-value is less than 0.05
- **THEN** we have enough evidence to reject the null hypothesis
- **OTHERWISE** there is not enough evidence to reject the null hypothesis.

Using the summary, we see that both weight and horsepower have p-values <0.0001 and 0.0015. Therefore, there is a significant association of weight and horsepower on fuel consumption.

We can also construct 95% confidence interval for the coefficient of weight using the `confint()` function.

```r
# Calculating confidence intervals of model coefficients
confint(mod, # Linear Model fit
        level = 0.95) # Significance level
                2.5 %      97.5 %
```

```
(Intercept) 33.95738245 40.49715778
wt          -5.17191604 -2.58374544
hp          -0.05024078 -0.01330512
```

More information about the `confint()` function can be found by typing `?confint` into `R`.

Using the summary, we see that we have an R^2 of 0.83 and an adjusted R^2 of 0.81. Therefore we see that we have a better fitting model with weight and horsepower

Suppose we wanted to predict the fuel consumption of a car that weighs 3.5 and has a horsepower of 200. We could use the `predict()` function to do this.

```
# Predicting for a weight of 3.5 and horsepower of 200
predict(mod, # Fitted model
        newdata = data.frame(mpg=NA, wt=3.5, hp=200)) # Data that we want to predict for
       1
17.30027
```

Therefore we see that the fuel consumption of a car that weighs 3.5 and has a horsepower of 200 will be `predict(mod,newdata=data.frame(mpg=NA, wt=3.5, hp=200))` miles per gallon.

Choosing your model

Often there will be choices of which explanatory variables to include. This is known as model selection. One of the most common methods is to compare values of the R^2 statistic choose the model which has the largest R^2. Others include Akaike Information Criteria (AIC) and Analysis of Variance (ANOVA).

We saw that both the weight and horsepower are linearly related to fuel consumption. There are three possible models

- Weight
- Horsepower
- Weight and Horsepower.

Lets fit these models in `R` using the `lm()` function.

```
# Fitting model with weight
formula1 <- mpg ~ 1 + wt
mod1 <- lm(formula1, data = mtcars)

# Fitting model with horsepower
formula2 <- mpg ~ 1 + hp
mod2 <- lm(formula2, data = mtcars)

# Fitting model with both weight and horsepower
formula3 <- mpg ~ 1 + wt + hp
mod3 <- lm(formula2, data = mtcars)
```

More information about the `summary()` function can be found by typing `summary()` into `R`.

Let's extract the R^2 from all models to compare. To do this we use the `summary()` function.

```
# Extracting adjusted R2 from model 1
summary(mod1)$adj.r.squared
[1] 0.7445939

# Extracting adjusted R2 from model 2
```

```
summary(mod2)$adj.r.squared
[1] 0.5891853

# Extracting adjusted R2 from model 3
summary(mod3)$adj.r.squared
[1] 0.5891853
```

If we use R^2 to select our model, we would choose a model with both weight and horsepower.