Overview: Statistical Modelling (Linear and Logistic Regression)

In the first half of this subject, we have covered the basic probability axioms, probability distributions and statistical inference concepts such as hypothesis testing and parameter estimation. In this week, we will be starting on statistical modelling which involves the fitting of a statistical model to the data set. In general, the modelling process consists of the following steps (Wu and Coggeshall, 2012):

1. Define the goals
2. Gather data
3. Decide the model structure
4. Prepare the data
5. Select and/or eliminate variables
6. Build candidate models
7. Select an appropriate final model
8. Implement and monitor

The choice of the model is of concern in steps 3, 6 and 7 in the list above. In this subject we will be looking at three commonly used statistical models for predictive modelling, namely linear regression, logistic regression and generalized linear models. Linear and logistic regression models will be covered in this week's content whereas generalized linear models will be covered in Week 5.

At this point, it is good to keep in mind that linear regression models are used for continuous response variables whereas logistic regression models are for categorical response variables or classification. Both are used to model the relationship between a set of predictor variables and the response variable.

4.1 Linear Regression Analysis

Introduction

Linear regression analysis is a statistical modelling tool which is used to model the relationship between a set of predictor variables and a continuous response variable Y. It is also known as supervised learning since it involves training on a data set in which the outcome is known, and the trained model is to be applied on future data in which the outcome is not known yet. It is important to note here that linear regression implies an underlying linear relationship between the response variable and predictor variable(s). Therefore, it is good practice to explore the relationship between the identified predictor and response variables first prior to performing a linear regression analysis. We can use a scatter plot (from Week 1) to visualize this relationship.

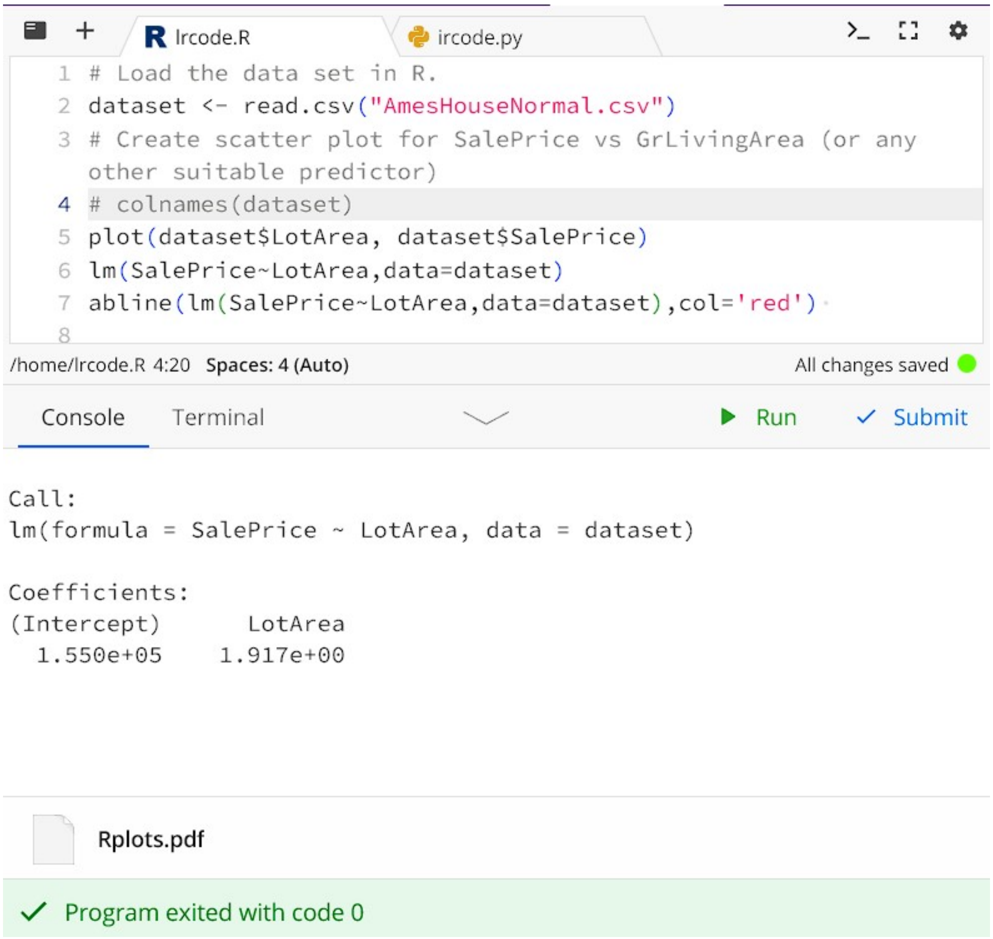
A linear regression analysis may start with plotting a scatter plot to visualize the relationship between the identified response and predictor variables.

Engage Activity: Scatter Plot for Linear Regression Analysis

For this topic, we will be using the data set AmesHouseNormal.csv which contains information from the Ames Assessor's Office for individual residential properties sold in Ames, IA from 2006 to 2010.

1. Read through the variables' descriptions (https://jse.amstat.org/v19n3/decock/DataDocumentation.txt) and load the data set in R.
2. Identify the response and predictor variable(s) associated with the problem.
3. Create a scatter plot to visualize and comment on the relationship between the response variable and one of the predictor variables.

Save your R code in a file named lrcode.R.



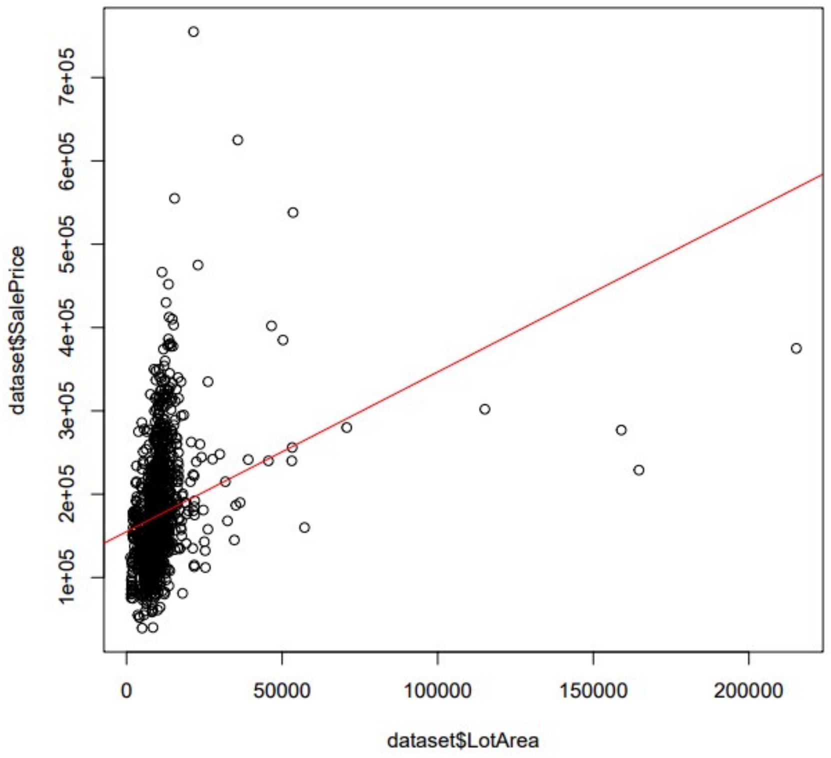
```
# Load the data set in R.
dataset <- read.csv("AmesHouseNormal.csv")
# Create scatter plot for SalePrice vs GrLivingArea (or any
other suitable predictor)
# colnames(dataset)
plot(dataset$LotArea, dataset$SalePrice)
lm(SalePrice~LotArea,data=dataset)
abline(lm(SalePrice~LotArea,data=dataset),col='red')
```

/home/lrcode.R 4:20 Spaces: 4 (Auto)

All changes saved

```
Call:
lm(formula = SalePrice ~ LotArea, data = dataset)

Coefficients:
(Intercept)      LotArea
  1.550e+05    1.917e+00
```

Program exited with code 0



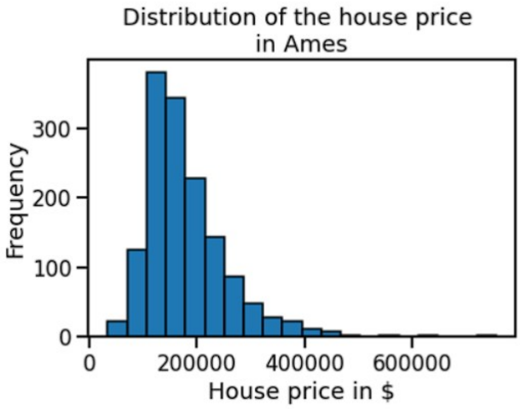
4.1.1 Linear Regression Models

Simple Linear Regression Model

Simple linear regression models the relationship between a single predictor variable and a continuous response variable. Recall from Week 1 that the concept of correlation is used to measure and describe the strength of the relationship between two variables. Regression goes a step further to quantify the nature of the relationship between the variables.

Simple linear regression attempts to estimate how much the response variable *Y* will change when *X* changes by a certain quantity, and is described by the simple linear regression model:

$$\mathrm{Y} = \beta_0 + \beta_1 \mathrm{X} + \epsilon$$



Source: The Ames housing dataset (Scikit-learn n.d.)

For example, we have identified from the AmesHouseNormal.csv data set that response variable Y is the sale price of houses. One of the potential predictors is ground living area (in square feet), which is denoted as X. In the regression equation as shown above, β_0 and β_1 are unknown constants that represent the intercept and slope in the model, respectively. They are called the regression coefficients or parameters. We use a data set to train the model to produce the estimates of the parameters and obtain the fitted linear equation:

$$\widehat{Y} = \widehat{\beta}_0 + \widehat{\beta}_1 X$$

The hat symbol (^) denotes the estimated value for an unknown parameter, or to denote the predicted value of the response variable. This fitted model can then be used to:

- explain the relationship between Y and X (explanatory)
- predict future values of Y for a particular value of X (predictive)

A regression model can be used for either explanatory or predictive purposes.

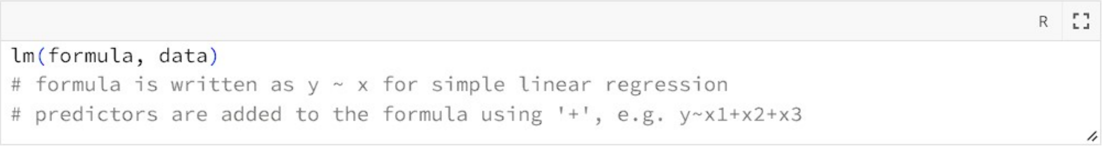
Given a data set, how are the regression coefficients or parameters in the simple linear regression model estimated? Since a simple linear regression is basically a linear equation representing a straight line, the goal here is to find the "best" line that fits the data points from the data set. The most commonly used approach to measure this "closeness" is the least squares criterion, which minimizes the residual sum of squares. For the interested reader, the following video explains the logic behind least squares regression line.

Introduction to residuals and least squares regression

https://www.youtube.com/watch?v=yMgFHbjbAW8

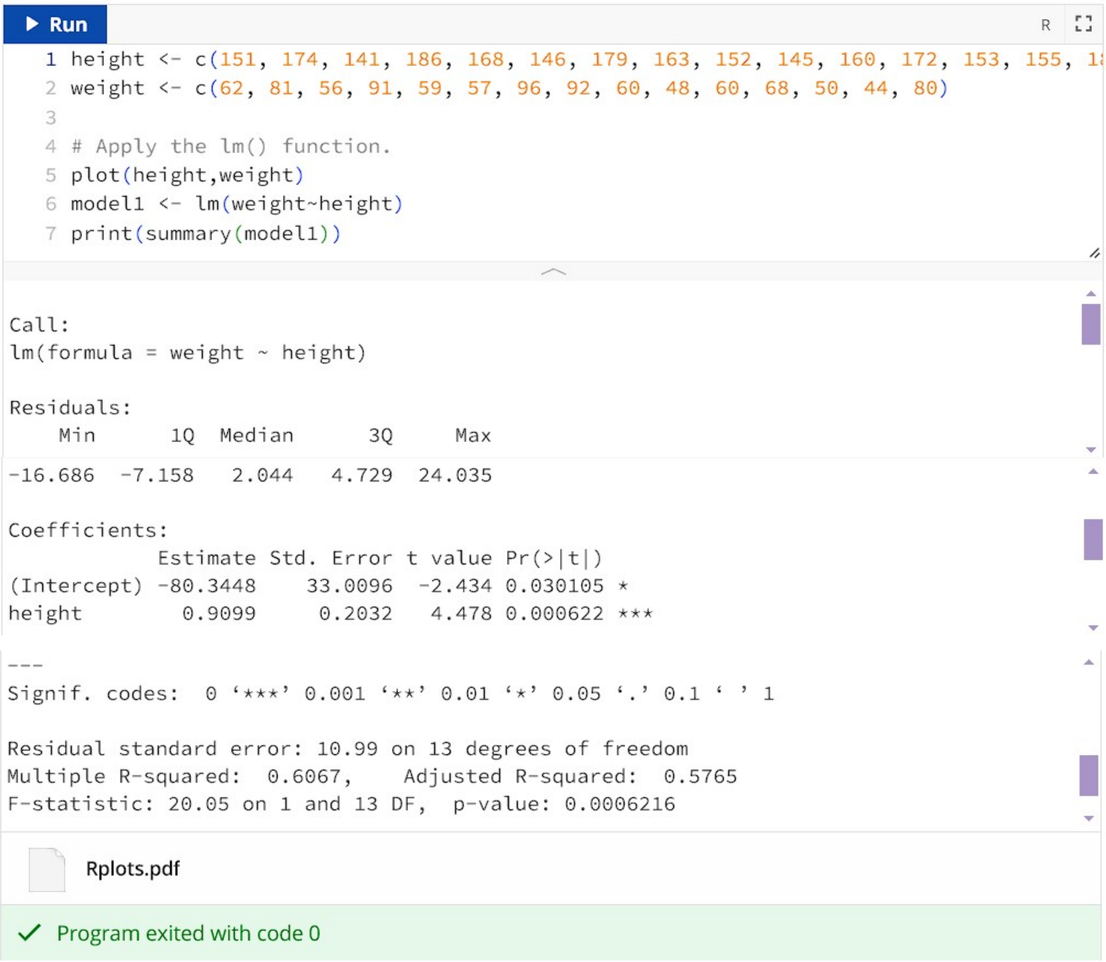
Source: (Khan Academy, 2017)

To fit a linear regression model in R, we can use the lm function. The basic syntax for fitting a linear regression model in R is given as follows:



```
lm(formula, data)
# formula is written as y ~ x for simple linear regression
# predictors are added to the formula using '+', e.g. y~x1+x2+x3
```

As an illustration, let's say we would like to fit a linear regression model to a simple data set consisting of 15 observations. The variables are height (in cm), weight (in kg) and gender (0 = male, 1 = female). Let us start with fitting a simple linear regression model with weight as the response variable and height as the predictor.

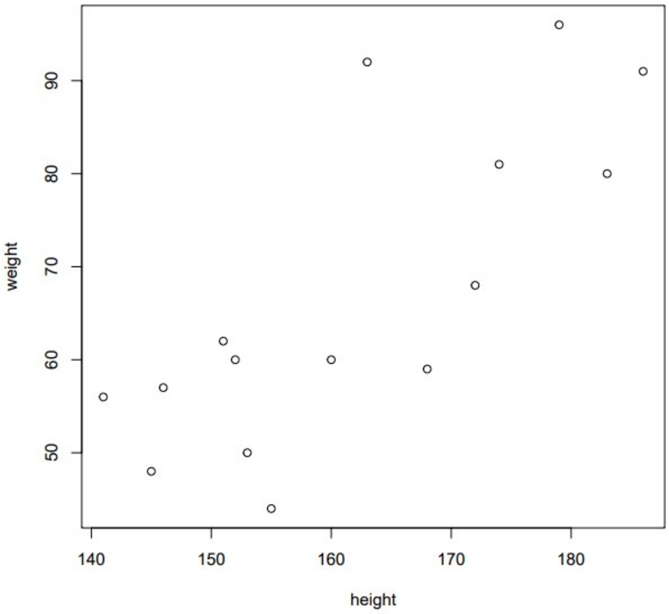


```r
height <- c(151, 174, 141, 186, 168, 146, 179, 163, 152, 145, 160, 172, 153, 155, 1
weight <- c(62, 81, 56, 91, 59, 57, 96, 92, 60, 48, 60, 68, 50, 44, 80)

# Apply the lm() function.
plot(height,weight)
model1 <- lm(weight~height)
print(summary(model1))
```

```
Call:
lm(formula = weight ~ height)

Residuals:
    Min      1Q  Median      3Q     Max
-16.686  -7.158   2.044   4.729  24.035

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -80.3448    33.0096  -2.434 0.030105 *
height        0.9099     0.2032   4.478 0.000622 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.99 on 13 degrees of freedom
Multiple R-squared:  0.6067,	Adjusted R-squared:  0.5765
F-statistic: 20.05 on 1 and 13 DF,  p-value: 0.0006216
```

Rplots.pdf

✓ Program exited with code 0



From the results output, the fitted simple linear regression equation is $\hat{y} = -80.3448 + 0.9099x$. This can be interpreted as for every one-cm increase in height, the weight will increase by 0.9099 kg. We will discuss the remaining of the output later.

Multiple Linear Regression Model

In real life situations, it is more realistic to have multiple predictors instead of a single predictor in the model. A linear regression model with multiple predictors is known as *multiple linear regression*:

$$\mathrm{Y} = \beta_0 + \beta_1\mathrm{X}_1 + \beta_2\mathrm{X}_2 + \ldots + \beta_\mathrm{k}\mathrm{X}_\mathrm{k} + \epsilon$$

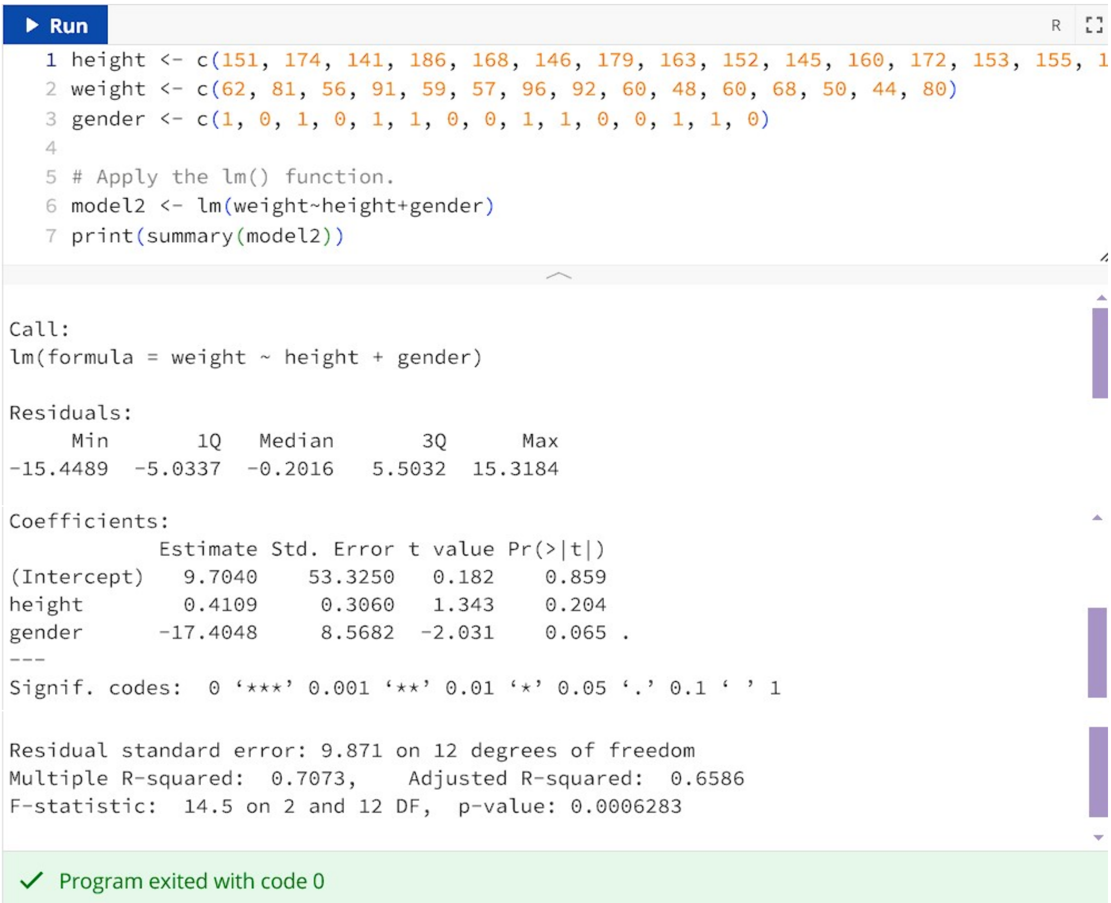
Similar to simple linear regression, the fitted multiple linear regression equation takes the form:

$$\mathrm{Y} = \hat{\beta}_0 + \beta_1\mathrm{X}_1 + \hat{\beta}_2\mathrm{X}_2 + \ldots + \hat{\beta}_\mathrm{k}\mathrm{X}_\mathrm{k}$$

where $X_1, X_2, \ldots, X_k$ denote the *k* predictors included in our model.

Unlike in simple linear regression whereby the parameter estimates using least-squares method can be easily obtained by solving a system of linear equations, the least-squares estimate for the parameters in a multiple linear regression model requires solving the normal equations using matrix operations. The solution of this linear system is done by the statistical software, so do not worry about it.

To illustrate fitting a multiple linear regression model, let us now try to add another predictor, i.e. gender into the model fitted in our earlier illustration in simple linear regression. Since gender is a categorical variable, it needs to be recoded as the model requires numerical inputs. The most common approach is to convert a variable into a set of binary dummy variables. In our example below, gender has been recoded such that '0' represents a male and '1' represents a female.



```
height <- c(151, 174, 141, 186, 168, 146, 179, 163, 152, 145, 160, 172, 153, 155, 1
weight <- c(62, 81, 56, 91, 59, 57, 96, 92, 60, 48, 60, 68, 50, 44, 80)
gender <- c(1, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0)

# Apply the lm() function.
model2 <- lm(weight~height+gender)
print(summary(model2))
```

```
Call:
lm(formula = weight ~ height + gender)

Residuals:
     Min       1Q   Median       3Q      Max 
-15.4489  -5.0337  -0.2016   5.5032  15.3184 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)   9.7040    53.3250   0.182    0.859  
height        0.4109     0.3060   1.343    0.204  
gender      -17.4048     8.5682  -2.031    0.065 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.871 on 12 degrees of freedom
Multiple R-squared:  0.7073,	Adjusted R-squared:  0.6586 
F-statistic:  14.5 on 2 and 12 DF,  p-value: 0.0006283
```

✓ Program exited with code 0

The fitted multiple linear regression equation is $\hat{y} = 9.7040 + 0.4109x_1 - 17.4048x_2$ where x_1 and x_2 denotes height and gender, respectively.

Inference on the Parameter Estimates

This section revisits the concept on Hypothesis Tests from Week 2.

Often times we would like to determine if there is a "statistically significant" relationship between the predictor and response variable. To do this for a predictor variable , we perform the hypothesis test:

$$H_0: \beta_j = 0$$

$$H_1: \beta_j \neq 0$$

The test statistic (or usually indicated as t Value in R or other statistical software output) is calculated using the formula $t_b = \frac{\hat{\beta}}{s.e.\,(\hat{\beta})}$, where s.e. is the standard error.

Under linear regression assumptions, this statistic follows the t-distribution with *n*−*2* degrees of freedom. Its corresponding p-value measures the extent to which a coefficient is “statistically significant”. A higher t-statistic (and lower p-value)

indicates a more significant predictor. These values are used to aid in choosing which predictors to be included in the model in line with the principle of parsimony.
In the example above, from the R output we see that the t-value and p-value for the predictor variable height is 1.343 and 0.204, respectively. At a significance level of *α=0.05*, height is said to not have a statistically significant linear relationship with weight since the p-value is more than α. This conclusion is based on the sample data set hence a different result may arise given a different data set.

4.1.2 Linear Regression Model Assessment

After fitting the data set with a model, we would like to assess the performance of the model.

Root Mean Squared Error

The root mean squared error (RMSE) is a commonly used metric for model performance evaluation and is calculated as the square root of the average squared error in the predicted values:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$$

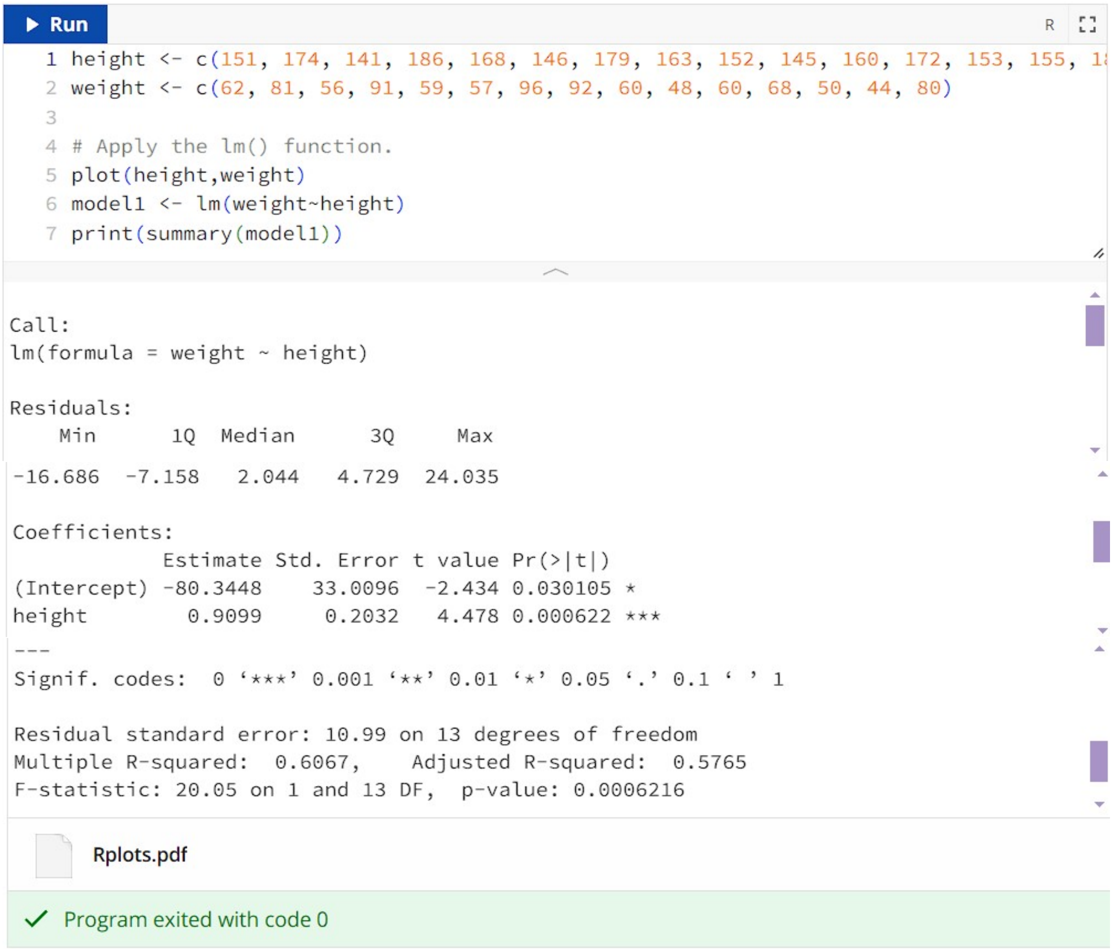
RMSE measures the overall accuracy of the model. Since its calculation is independent of the model specification, it can be used for comparison with other models (including models which are fit using machine learning techniques).

Residual Standard Error

The residual standard error (RSE) is similar in nature to RMSE, but the denominator is taken as the degrees of freedom instead of number of observations, with p denoting the number of predictors in the model:

$$RSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n - p - 1}}$$

Another variant is the adjusted R-squared, which penalizes the addition of more predictors into the model.

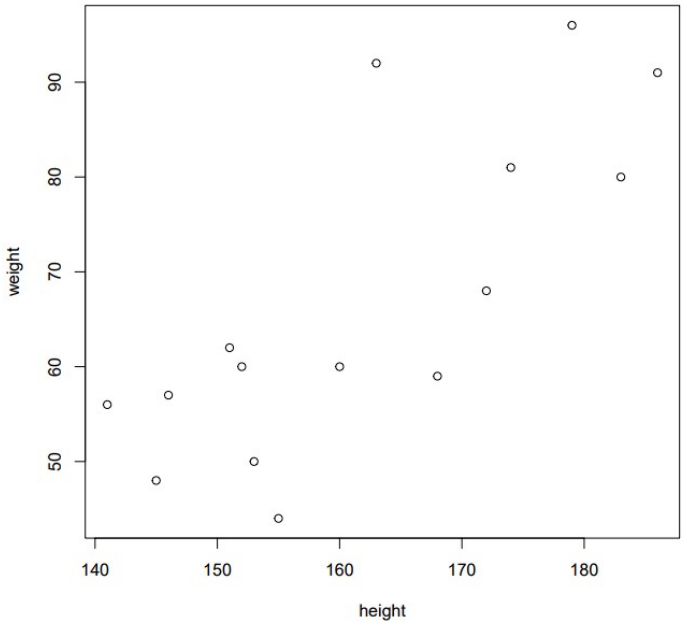


```r
height <- c(151, 174, 141, 186, 168, 146, 179, 163, 152, 145, 160, 172, 153, 155, 1
weight <- c(62, 81, 56, 91, 59, 57, 96, 92, 60, 48, 60, 68, 50, 44, 80)

# Apply the lm() function.
plot(height,weight)
model1 <- lm(weight~height)
print(summary(model1))
```

```
Call:
lm(formula = weight ~ height)

Residuals:
    Min      1Q  Median      3Q     Max 
-16.686  -7.158   2.044   4.729  24.035 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -80.3448    33.0096  -2.434 0.030105 *  
height        0.9099     0.2032   4.478 0.000622 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 10.99 on 13 degrees of freedom
Multiple R-squared:  0.6067,    Adjusted R-squared:  0.5765 
F-statistic: 20.05 on 1 and 13 DF,  p-value: 0.0006216
```

Rplots.pdf

✓ Program exited with code 0



Using the same data set and code that we have seen in 4.1.2, the R-squared value for the fitted model is 0.6067 and adjusted R-squared is 0.5765.

Explore Activity: Linear Regression Model Assessment

Time: 30 minutes

Purpose: The purpose of this activity is to explore model assessment metrics using R.

Task: Explore the data set given in the example in 4.1.1 by fitting a linear regression model with:

- only height as the predictor variable
- height and gender as the predictor variable

What do you notice about the R-squared and adjusted R-squared values? Post your answer in the Discussion Board.

4.1.3 Model Selection and Stepwise Regression

Engage Activity

Time: 20 minutes

Purpose: In some problems, many variables could potentially be included as predictors in a regression model. The purpose of this activity is to get started with discussing model selection in linear regression modelling.

Task: By examining the AmesHouse.csv data set, what are the potential predictor variables for house sale price?

Model Selection and Stepwise Regression

It is important to bear in mind that having more predictors does not necessarily translate into having more information or a better model. We use the principle of Occam's razor.

Occam's razor: all things being equal, a simpler model should be used in preference to a more complicated model.

The following are some metrics which are helpful in guiding model selection:

- Adjusted R^2: $R_{adj}^2 = 1 - (1 - R^2)\frac{n-1}{n-p-1}$
- Akaike's Information Criterion (AIC): $\mathrm{AIC} = 2p + n\log\frac{\mathrm{RSS}}{n}$

- Schwarz's Information Criterion (SIC)
- Bayesian Information Criterion (BIC)

In the model selection process, there are three possible approaches:

- *all subset regression*: searches through all possible models to look for the one which maximizes the adjusted R^2 and minimizes AIC. However, this is computationally expensive and is obviously not feasible for problems with many variables.
- stepwise regression, which has two variations: backward elimination or forward selection. Backward elimination starts with a full model and successively drops variables which are not significant. Forward selection starts with an empty model and successively adds variables which are significant.
- Successively adds and drops predictors to find a model that lowers AIC or increases adjusted.
- Penalized regression methods such as ridge regression or lasso regression, which incorporates a constraint that penalizes the model for too many variables. Penalized regression applies the penalty by reducing coefficients.

For further details, you can watch the following videos:

1. Brandon Foltz. (2020, Aug 10). Statistics 101: Model building methods - Forward, backward, stepwise, and subsets [Video]. Youtube. https://www.youtube.com/watch?v=-inJu1jHqb8
2. Brandon Foltz. (2021, Aug 31). Statistics 101: Multiple regression, AIC, AICc, and BIC basics [Video]. Youtube. https://youtu.be/-BR4WElPIXg

4.1.4 Prediction Using Regression

Extrapolation

It is important to note that a fitted regression model is valid only for predictor values for which the data has sufficient values. Regression models should not be used to extrapolate beyond the range of the data.

Confidence and Prediction Intervals

Confidence intervals are reported along with the estimated regression coefficients to provide a picture on the uncertainty surrounding the estimates. A confidence interval pertains to a mean or other statistic calculated from multiple values (refer to Week 3's discussion on confidence interval estimates). On the other hand, a prediction interval pertains to uncertainty around a single value. Therefore, a prediction interval is typically wider than a confidence interval for the same value. For the purpose of specific individual predictions, a prediction interval would be more appropriate. Using a confidence interval instead would underestimate (since it is narrower than a prediction interval) the uncertainty in a given predicted value.

You can watch the following video for further explanation on confidence and prediction intervals.

Confidence and prediction intervals (1 of 1)

https://www.youtube.com/watch?v=TO9QO-6v8qw

Source: (Amelia McNamara, 2020)

Interpreting the Regression Equation

It is quite common to find that predictor variables within a regression model are correlated with each other. Having correlated predictors can make it difficult to interpret the sign and value of regression coefficients and can inflate the standard error of the estimates.

An extreme case of correlated predictors results in multicollinearity causing redundance among the predictor variables. In linear regression models, variables must be removed until multicollinearity is eliminated.

When an important variable is not included in the regression equation, this variable is known as a confounding variable. For example, omitting location in predicting house prices may result in unexpected results.

In a regression model, it is also of interest to distinguish between main effects and interactions between main effects. When only main effects are used in the model, the relationship between a predictor variable and the response is assumed to be independent of the other predictor variables, which is often not the case. Prior knowledge and intuition can guide the choice of which interaction terms to include in the model.

Cross-Validation

Classic statistical regression metrics such as (R^2) and p-values are calculated and applied to the same data that was used to fit the model. Ideally, we would want to assess the model performance using a different set of observations but often times we will not have these data. One way to circumvent this is by splitting our sample into training set for model building and validation set for model evaluation. The validation set is also known as holdout data. A holdout data is part of the original data which has been set aside and is not involved in the initial model fitting process. This approach is known as cross-validation.

However, a holdout sample is subject to some uncertainty due to possibly small sample size, and also the variability that arises from different holdout samples. To overcome this, we can perform k-fold cross-validation by having multiple sequential hold-out samples.

k-Fold Cross-Validation

https://www.youtube.com/watch?v=kituDjzXwfE

Source: (David Caughlin, 2020)

4.2 Logistic Regression Analysis

Introduction

In the previous topic, we use multiple linear regression to model the linear relationship between a continuous response variable with a set of predictor variables. If the response variable is categorical, we can use logistic regression instead.

Statistical Learning: 4.2 Logistic Regression

https://www.youtube.com/watch?v=kr_Be9NVXOM

Source: (Stanford Online, 2022)

In a logistic regression model, we model the probability of a level of the categorical variable instead of the outcome itself. In particular, let us consider the case whereby the response variable falls into one of two categories, Yes or No. For example, in loan default problem, we model the probability of a borrower defaulting instead of the actual default status.

In our case study data set on campaign success, there are two possible outcomes for the response variable success - Yes (denoted as 1) and No (denoted as 0) hence we model the probability of the campaign being a success. Why don't we directly model on the status itself? The relationship between a categorical response variable and its predictors is not linear. However, the relationship between the log of odds (also known as logit) and the predictor(s) can be linear and this is the basis of a logistic regression model.

For a binary categorical variable (categorical variable with two possible outcomes or levels), its outcomes can be categorized as "successes" versus "nonsuccesses". Odds are the ratio of the probability of a "success" to the probability of a "nonsuccess". If the probability of a "success" is denoted as *p*, then

$$Odds = \frac{p}{1-p}$$

As such, the logistic regression model is defined as:

$$\mathrm{logit}(p) = \log\frac{\mathrm{p}}{1-\mathrm{p}} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k$$

The *logit* function maps the probability pp from (0, 1) to any value *(∞,−∞)*. We would like to stress here that the response in the logistic regression formula above is the log odds of a binary outcome of "success". We observe only the binary outcome, not the log odds, so special statistical methods are needed to fit the equation. Logistic regression is actually a special case of generalized linear models. Generalized linear models will be discussed in further details in the next week.

As can be seen from the logistic regression equation above, the predicted value from logistic regression is in terms of the log odds: $\widehat{Y} = \log(\mathrm{Odd}(\mathrm{Y}) - 1)$. The predicted probability can then be derived from the formula:

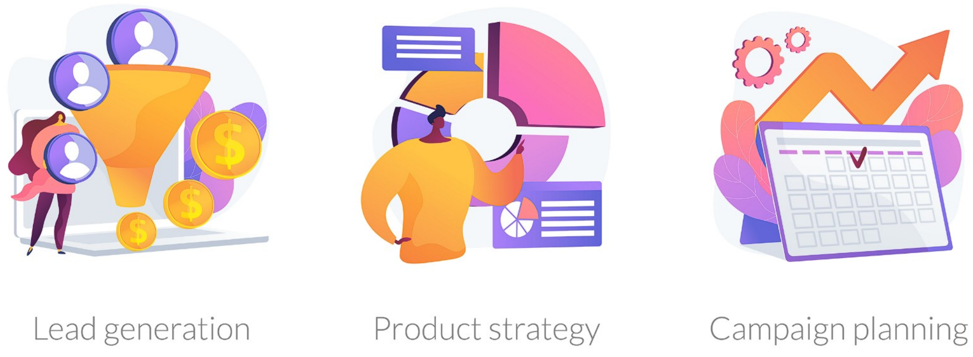
$$\hat{p} = \frac{1}{1 + e^{-\widehat{Y}}}$$

The probabilities are on a scale from 0 to 1 and do not declare the predicted outcome. The predicted outcome is declared based on a pre-determined prediction threshold. For example, with a prediction threshold set at 0.5, one might predict success = Yes for any observation in which *p>0.5* based on the logistic regression model.

One advantage of logistic regression is that it produces a model that can be scored to new data rapidly, without the need for re-computation. Another is the relative ease of interpretation of the model, as compared with other classification methods. The key conceptual idea is understanding an odds ratio.

4.2.1 Model Fitting and Assessment

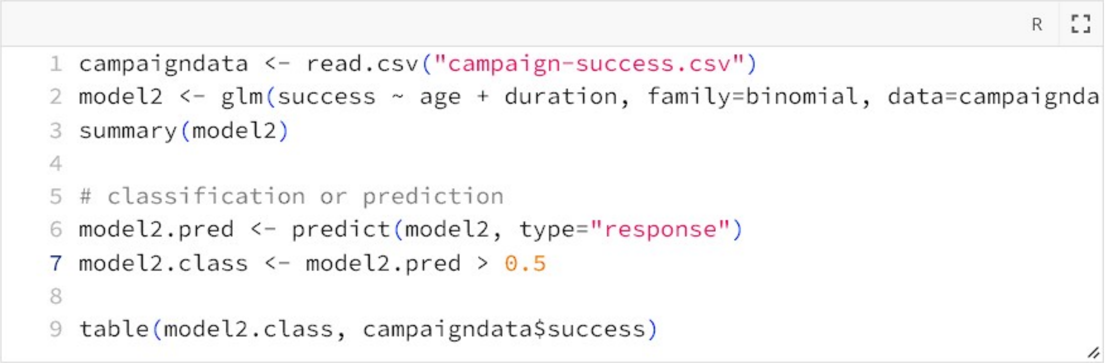
Model Fitting



Source: Marketing Industry (Adobe Stock n.d.)

As logistic regression does not have a closed-form solution, model fitting is performed through maximum likelihood (ML) estimation. Recall that we have discussed ML estimation in Week 3. In logistic regression, the ML estimation process finds the solution such that the estimated log odds best describes the observed outcome. The solution is found through quasi-Newton optimization (https://optimization.cbe.cornell.edu/index.php?title=Quasi-Newton_methods) that iterates between a scoring step (Fisher’s scoring), based on the current parameters, and an update to the parameters to improve the fit. In statistical computing software such as R, the optimization is handled by the algorithm's software.

In R, we use the glm() function to perform logistic regression analysis. The table() function is used to build a confusion matrix for the fitted model. In the example code below, the prediction threshold used is 0.5.



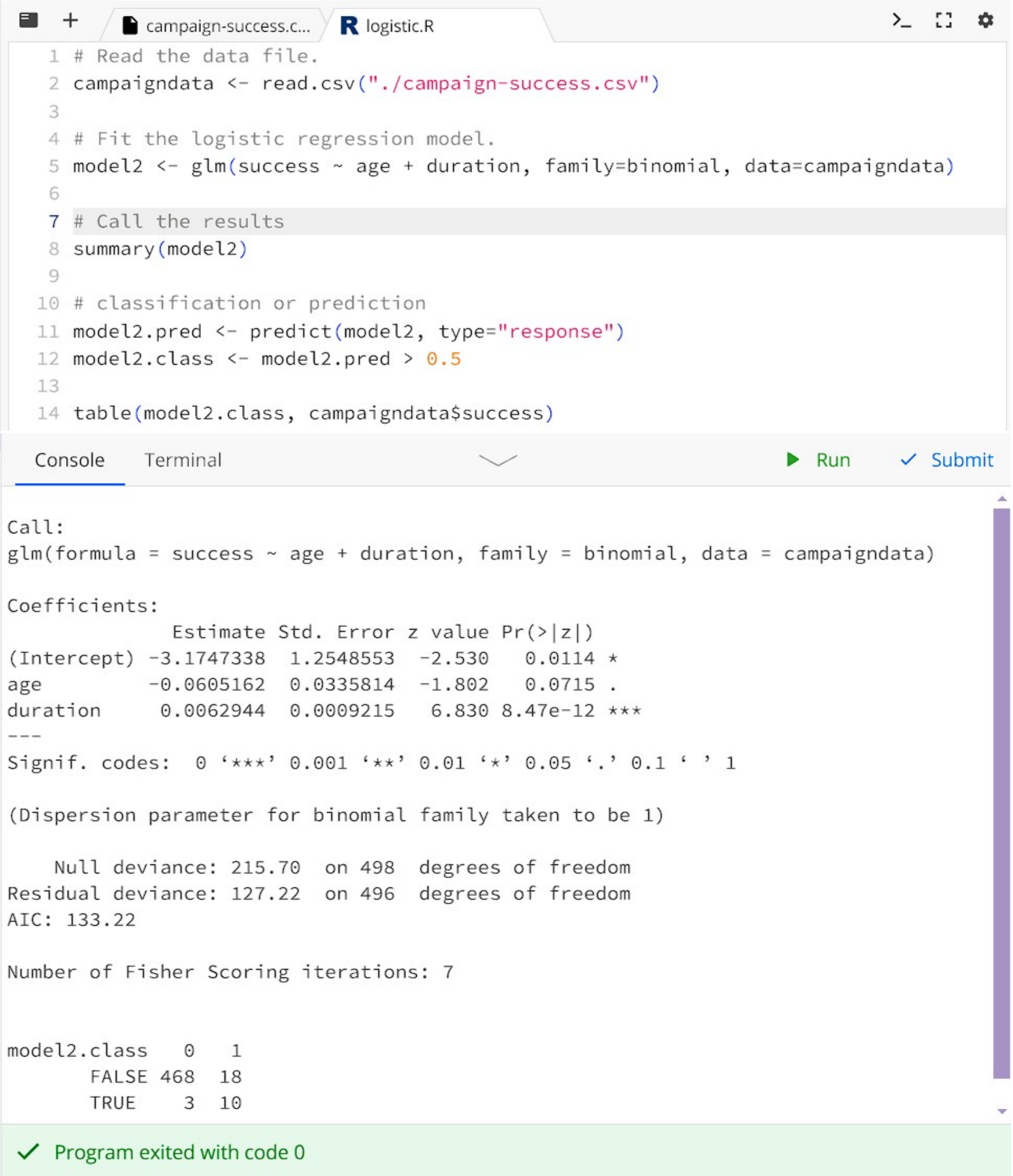
```r
campaigndata <- read.csv("campaign-success.csv")
model2 <- glm(success ~ age + duration, family=binomial, data=campaignda
summary(model2)

# classification or prediction
model2.pred <- predict(model2, type="response")
model2.class <- model2.pred > 0.5

table(model2.class, campaigndata$success)
```

Model Assessment

Similar to linear regression, the estimated coefficients, standard error of the coefficients, z-value and p-value are used for examining and improving the model. Since the response variable in logistic regression is binary, we do not use RMSE or R-squared. As one of its purposes is for classification, logistic regression model can be assessed using accuracy, confusion matrix and ROC curve. A commonly used classifier threshold is 0.5.

Based on the confusion matrix constructed for campaign-success.csv data used in our example, using a threshold of 0.5 the model predicted 10 out of 28 successes and 468 out of 471 non-successes correctly. Area under the curve evaluated for the fitted model is 0.9163.

Linear and logistic regression models share similar approaches when it comes to variable or model selection, confounding and correlated variables.



```r
# Read the data file.
campaigndata <- read.csv("./campaign-success.csv")

# Fit the logistic regression model.
model2 <- glm(success ~ age + duration, family=binomial, data=campaigndata)

# Call the results
summary(model2)

# classification or prediction
model2.pred <- predict(model2, type="response")
model2.class <- model2.pred > 0.5

table(model2.class, campaigndata$success)
```

```
Call:
glm(formula = success ~ age + duration, family = binomial, data = campaigndata)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.1747338  1.2548553  -2.530   0.0114 *
age         -0.0605162  0.0335814  -1.802   0.0715 .
duration     0.0062944  0.0009215   6.830 8.47e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 215.70  on 498  degrees of freedom
Residual deviance: 127.22  on 496  degrees of freedom
AIC: 133.22

Number of Fisher Scoring iterations: 7


model2.class   0   1
       FALSE 468  18
       TRUE    3  10
```

✓ Program exited with code 0

4.2.2 Imbalanced Data

Imbalanced Data

In certain fields such as fraud detection or loan status modelling, the response classes are imbalanced. We observe far more “non-successes” than “successes”, as can be seen from our example data set campaign-success.csv and this could affect the model’s performance. There are strategies which can be used to overcome class imbalance.

If you have enough data, one solution is to undersample the prevalent class. One criticism of the undersampling method is that it throws away data and is not using all the information at hand. If the data at hand is relatively small and the rarer class contains a few hundred or a few thousand observations, then undersampling risks throwing out useful information. In this case, we can opt to oversample the rarer class instead by drawing additional rows with replacement (bootstrapping). A similar effect can be achieved by weighting the data.

Other than bootstrapping, we can also create new observations that are similar but not identical to existing observations. One such implementation is the SMOTE algorithm, which stands for Synthetic Minority Oversampling Technique. The SMOTE algorithm finds a record of the observation that is similar to the record being upsampled and creates a synthetic record that is a randomly weighted average of the original record and the neighboring record, where the weight is generated separately for each predictor. The number of synthetic oversampled records created depends on the oversampling ratio required to bring the data set into approximate balance with respect to outcome classes.

4.2.3 Multinomial Logistic Regression

In our earlier discussions on logistic regression, the response variable has only two classes (e.g. Yes and No). In some cases, we may wish to classify a response variable that has more than two classes (e.g. Yes, No, Neutral). It is possible to extend the two-class logistic regression to *K>2* classes, in which we have the multinomial logistic regression. To do this, a class is selected to serve as the baseline. Without loss of generality, we can select the *K-th* class for this role. It can be shown that for *k=1,…,K−1*,

$$\log\frac{\Pr(\mathrm{Y}=\mathrm{k}|\mathrm{X}=\mathrm{x})}{\Pr(\mathrm{Y}=\mathrm{K}|\mathrm{X}=\mathrm{x})}=\beta_{k_0}+\beta_{k_1x_1}+\beta_{k_2x_2}+\ldots+\beta_{k_qx_q}$$

The coefficient estimates will differ since they depend on the choice of baselines. Therefore, interpretation of the coefficients in a multinomial logistic regression model must be done with care. However, the fitted values (predictions), log odds between any pair of classes and other key model outputs will remain the same hence the choice of K-th class is unimportant.

Watch the following video for an explanation and example on multinomial logistic regression.

The regression analysis results in the video are generated using STATA, but the principles of interpretation is the same for output generated using R.

Multinominal logistic regression, Part 1: Introduction

https://www.youtube.com/watch?v=JcCBIPqcwFo

Source: (NCRMUK, 2021)

4.2.4 Application Activity: Logistic Regression Analysis

Use the campaign-success.csv data set for this activity.

Time: 30 minutes

Purpose: The purpose of this activity is to perform logistic regression analysis on the campaign-success.csv data set.

Task: Identify the response variable and potential predictor variable(s) to perform a logistic regression analysis on the campaign-success.csv data set. Run the logistic regression analysis using R.

4.3 Regression Diagnostics

Introduction

In explanatory modelling, besides the model assessment metrics discussed in the earlier topics, it is also of interest to perform regression diagnostics. Regression diagnostics are mostly based on analysis of the residuals and is used to assess how well the model fits the data. These steps are not for addressing predictive accuracy but can provide useful insight in a predictive setting.

Influential Values

An influential observation is one which, either individually or together with several other observations, has a demonstrably larger impact on the calculated values of various estimates … than is the case for most of the other observations. – Besley, Kuh and Welsch (1980)

An influential observation is a data point whose absence would significantly change the regression equation. An influential point may not be a large outlier but it has a high leverage on the regression. However, an observation may not have the same impact on all regression outputs. It may have influence on $\widehat{\beta}$, a linear combination of $\widehat{\beta}$, the estimated variance of $\widehat{\beta}$, the fitted values, and/or the goodness-of-fit statistics. The primary goal of the analysis should determine which influence to consider. For example, if the primary purpose of modelling is for prediction, then measuring influence on the fitted values may be more appropriate than measuring influence on $\widehat{\beta}$.

Influence measures can be classified into five groups:

- Measures based on residuals
- Measures based on the prediction matrix
- Measures based on the volume of confidence ellipsoids
- Measures based on influence functions
- Measures based on partial influence

Each influence measure is designed to detect a specific phenomenon in the data and are closely related. In any particular application, the analyst does not have to look at all of these measures since there is a great deal of redundancy in them.

In linear regression modelling, various diagnostics plots such as Cook's D plot, DFFits or DFBetas plots are used to identify influential points.

Outliers

One of the aims in regression diagnostics is outlier detection. An outlier is an observation which is distant from most of the other observations.

In regression, an outlier can also be understood as an observation whose actual y value is distant from the predicted y value. In Week 1's topic on descriptive statistics, outliers in a data set can be visualized using a box plot and we have also briefly discussed the effect that outliers could have on measures of location and dispersion. Similarly, an outlier may have an effect in regression modelling. In the event that the outlier has little influence on the regression results, its existence is not a point of concern.

Outliers can be detected by examining the standardized residual, which is the residual divided by the standard error of the residuals and can be interpreted as "the number of standard errors away from the regression line". However, influential observations need not be outliers in the sense of having large residuals. Therefore, graphical methods based on residuals alone will fail to detect these unusual points. To study this problem we need the additional concept of "leverage".

```
influencePlot(fit, id.method="identify", main="Influence
Plot", sub="Circle size is proportial to Cook's Distance"
```

Outliers may occur due to errors in data entry or mismatch of units (e.g. reporting a sale in thousands of dollars rather than simply in dollars). In big data problems, outliers may not be a serious problem in fitting the regression for predictive purposes. However, outliers are central to anomaly detection or fraud detection.

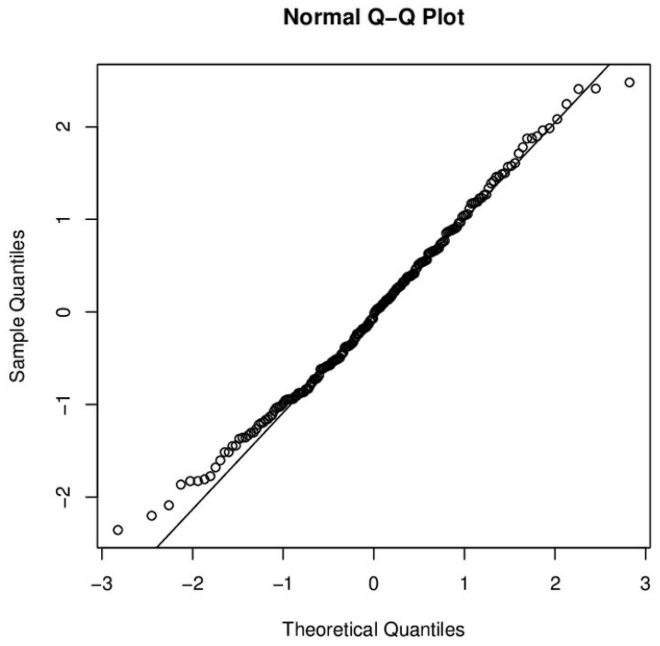
Multicollinearity

Collinearity or multicollinearity describes a situation whereby the explanatory or predictor variables in the fitted model is linearly related with each other. Multicollinearity may cause a problem in the fitted linear regression model because individual predictor effects could be masked. Presence of multicollinearity can be detected from *variance inflation factors (VIF)* or condition indices. Predictors with VIF values that are more than 10, or condition indices higher than 30 indicate presence of multicollinearity and requires further examination.

4.3.1 Regression Assumptions

Underlying linear regression modelling is a set of assumptions which should be verified for the validity of statistical inference such as hypothesis tests and p-values. The assumptions are:

- Linearity: the relationship between response and predictor variables should be linear. Therefore, it is good practice to check this assumption through a scatterplot before proceeding to model fitting.
- Normally and independently distributed residuals: A QQ-plot, Kolmogorov-Smirnov test or Shapiro-Wilk test can be used to check for assumption of normality, especially when the sample size is small.



Source: Normal QQ plot for standardised residuals (Kristin A Linn, 2015)

- Homoskedasticity: constant residual variance across the range of predicted values

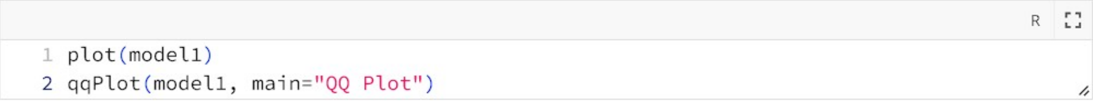
The following video provides further explanation on assumptions of linear regression. Closed captions are not available for this video. However, you can expand the tab to access the video transcript.

Assumptions of Linear Regression

https://www.youtube.com/watch?v=sDrAoR17pNM

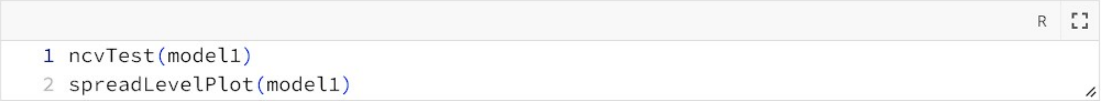
Source: (Datatab, 2021)

In R, the QQ-plot and other plots for checking regression assumptions can be requested using the following code:



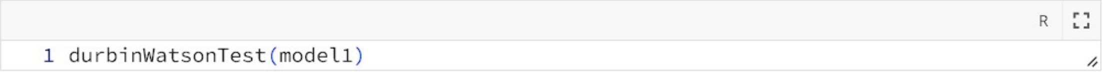
```r
plot(model1)
qqPlot(model1, main="QQ Plot")
```

The opposite of homoskedasticity is known as heteroskedasticity. Heteroskedasticity indicates that the prediction errors differ for different ranges of the predicted value, and may suggest an incomplete model. Assumption of homoskedasticity or homogeneity of variance can be checked from a plot of the studentized residuals versus predicted values. The R code to conduct a formal test (nonconstant variance error test ncvTest) and request for the plot on homoskesdasticity assumption (spreadLevelPlot) is given as follows:



```r
ncvTest(model1)
spreadLevelPlot(model1)
```

Assumption on independently distributed residuals can be checked using the Durbin-Watson statistic. The Durbin-Watson statistic detects if there is significant autocorrelation in regression involving time series data. If the errors from a regression model are correlated, then this information can be useful in making short-term forecasts and should be built into the model. For longer-term forecasts or explanatory models, excess autocorrelated data at the microlevel may distract. In that case, smoothing, or less granular collection of data is preferred.



```r
durbinWatsonTest(model1)
```

Question 1: Which one of the following statistical model is appropriate for classifying whether an e-mail is spam?

Ans: Logistic regression

Explanation: Logistic regression is used for modelling categorical response variables.

Question 2: Which of the following are assumptions in linear regression modelling? Select all that applies.

Ans:

- Residuals are normally distributed
- Homogeneity of variance
- Response and independent variables are linearly related

Explanation: The assumption of normality is for the response or residuals, and is not required for the independent variables in the model.

Question 3: Which one of the following can be used for model selection or model building in linear regression modelling? Select all that applies.

Ans:

- Adjusted R-squared
- Akaike's Information Criterion (AIC)
- Backward elimination algorithm