Week 9 - Evaluation of Regression Models

Introduction:

Welcome to this unit on methods of evaluating regression models. Regression models are essential in mathematical modeling and data analysis, and their performance needs to be assessed rigorously. Without proper evaluation, researchers may misinterpret the relationships between variables. They might erroneously conclude that certain predictors have a significant impact on the outcome when they do not or vice versa. Inadequately evaluated regression models may produce inaccurate predictions. If the model does not capture the underlying relationships between variables effectively, its predictive performance may be poor, leading to unreliable forecasts or estimates. In this unit, we will explore various evaluation methods and metrics used to assess the quality and accuracy of regression models.

Importance of Model Evaluation:

Model evaluation is critical to determine how well a regression model fits the data and makes predictions. It helps in selecting the best model among competing models and guides model refinement. Accurate model evaluation is essential for making informed decisions in various fields, including economics, engineering, and the sciences.

Common Evaluation Metrics:

Several metrics are commonly used to evaluate regression models:

1. Mean Squared Error (MSE):

MSE measures the average squared difference between predicted and actual values. It penalizes larger errors more than smaller ones, making it sensitive to outliers. The formula for MSE is:

$$MSE = \frac{\sum(y_i - \bar{y})^2}{n}$$

where y_i is the actual value, $\bar{y}$ is the mean of actual values, and n is the number of data points.

2. Root Mean Squared Error (RMSE):

RMSE is the square root of MSE and is in the same units as the dependent variable. It provides a more interpretable measure of error.

$$RMSE = \sqrt{MSE}$$

3. Mean Absolute Error (MAE):

MAE measures the average absolute difference between predicted and actual values. It is less sensitive to outliers compared to MSE.

$$MAE = \frac{\sum |y_i - \bar{y}|}{n}$$

where ŷ is the predicted value.

4. R-squared (R^2):

R^2 represents the proportion of variance in the dependent variable explained by the model. It ranges from 0 to 1, where a higher value indicates a better fit.

$$R^2 = 1 - (SSE/SST)$$

where SSE is the sum of squared errors and SST is the total sum of squares.

5. Adjusted R-squared (Adjusted R^2):

Adjusted R^2 adjusts R^2 for the number of predictors in the model. It penalizes the addition of unnecessary variables.

$$AdjustedR^2 = 1 - [(1 - R^2) * ((n - 1)/(n - p - 1))]$$

where n is the number of data points and p is the number of predictors.

Model Validation Techniques in R:

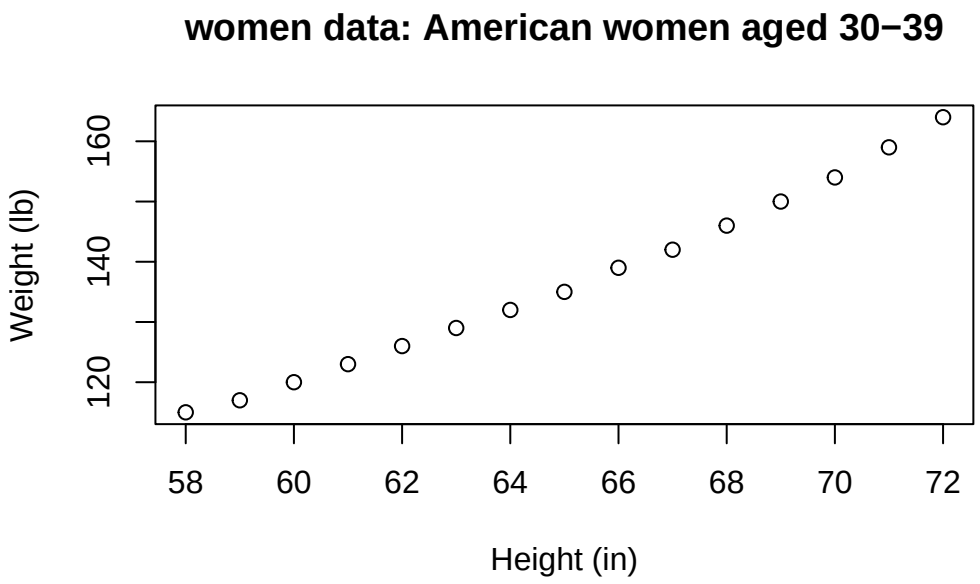
Summary Statistics:

In this example, we will do an exploration of a dataset which includes American women's height and weight. The question we are asking, is can we predict a woman's height given her weight?

```
 # You may need to install the datasets library (by using install.packages('datasets'))
library(datasets)
```

In this example, it's worth plotting the data to get a sense of whether there may be a trend or not.

```
plot(women, xlab = "Height (in)", ylab = "Weight (lb)",
     main = "women data: American women aged 30-39")
```



The summary() function provides a summary of the regression model, including coefficients, standard errors, t-values, and p-values for each predictor.

Example:

```
summary(lm_model)
```

Below we have created a linear regression model, where we use height as the dependent variable and weight as the independent variable.

```
lmHeight = lm(height~weight, data = women) #Create the linear regression
summary(lmHeight) #Review the results
```

```
Call:
lm(formula = height ~ weight, data = women)

Residuals:
     Min       1Q   Median       3Q      Max
-0.83233 -0.26249  0.08314  0.34353  0.49790

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 25.723456   1.043746   24.64 2.68e-12 ***
weight       0.287249   0.007588   37.85 1.09e-14 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.44 on 13 degrees of freedom
Multiple R-squared:  0.991, Adjusted R-squared:  0.9903
F-statistic:  1433 on 1 and 13 DF,  p-value: 1.091e-14
```

The most important results are, our Multiple R-Squared and Adjusted R

Coefficient of Determination (R-squared):

R-squared measures the proportion of the variance in the dependent variable that is explained by the independent variables in the model. Higher values indicate better fit.

If we just want to isolate this statistic, we can call the following function.

Example:

```
summary(lm_model)$r.squared
```

Adjusted R-squared:

Adjusted R-squared penalizes the inclusion of unnecessary predictors in the model and provides a more conservative measure of model fit, especially for models with multiple predictors.

If we just want to isolate this statistic, we can call the following function.

Example:

```
summary(lm_model)$adj.r.squared
```

However, these statistics are included in the summary report given above.

Root Mean Squared Error (RMSE) or Mean Absolute Error (MAE):

These metrics quantify the average difference between observed and predicted values. Lower values indicate better predictive accuracy.

Example

```
RMSE: sqrt(mean(residuals(lm_model)^2))
```

```
rmse = sqrt(mean(residuals(lmHeight)^2))
rmse
```

```
[1] 0.4096541
```

Multiple Linear Regression

Generally speaking, modelling is going to be concerned with looking at more than one factor. In this case, what we can do is use multiple linear regression to assess whether multiple factors are more useful to predict a value.

In this case, we are looking at predicting a cars price based on some characteristics of the car.

```
automobile <- read.csv("~/FP041/automobile.csv")
```

For example, we could look at whether city.mpg is a good predictor of price but doing a simple linear regression model.

```r
model = lm(price ~ `city.mpg`, data = automobile) #Create the linear regression
summary(model) #Review the results
```

```
Call:
lm(formula = price ~ city.mpg, data = automobile)

Residuals:
   Min     1Q Median     3Q    Max
 -7687  -3292  -1747   1595  21980

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 35947.36    1693.22   21.23   <2e-16 ***
city.mpg     -894.81      64.84  -13.80   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5739 on 191 degrees of freedom
Multiple R-squared:  0.4993,    Adjusted R-squared:  0.4967
F-statistic: 190.5 on 1 and 191 DF,  p-value: < 2.2e-16
```

Not so great with a multiple R squared of 0.4993 and an adjusted R squared of 0.4967. Maybe we can use another variable and see if this helps the model. Lets add highway.mpg. Note, all we do to add another variable is use the + sign and then the next variable.

```r
model = lm(price ~ `city.mpg` + `highway.mpg`, data = automobile) #Create the linear regre
summary(model) #Review the results
```

```
Call:
lm(formula = price ~ city.mpg + highway.mpg, data = automobile)

Residuals:
   Min     1Q Median     3Q    Max
 -8304  -3480  -1191   1218  20301

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  39086.4     2022.7  19.324  < 2e-16 ***
```

```
city.mpg      -174.0      271.2  -0.642  0.52190
highway.mpg   -694.9      254.2  -2.734  0.00685 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5644 on 190 degrees of freedom
Multiple R-squared:  0.5183,    Adjusted R-squared:  0.5132
F-statistic: 102.2 on 2 and 190 DF,  p-value: < 2.2e-16
```

This has improved our multiple R squared to 0.5183 and adjusted R squared to 0.5132. Maybe we can add horsepower to our multiple linear regression model.

```r
model = lm(price ~ `city.mpg` + `highway.mpg` + horsepower, data = automobile) #Create the
summary(model) #Review the results
```

```
Call:
lm(formula = price ~ city.mpg + highway.mpg + horsepower, data = automobile)

Residuals:
    Min      1Q  Median      3Q     Max
-9055.8 -2391.2  -392.6  1797.2 16163.5

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  3396.04    4002.92   0.848  0.39729
city.mpg      565.82     234.08   2.417  0.01659 *
highway.mpg  -666.33     207.61  -3.209  0.00156 **
horsepower    155.33      15.87   9.789  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4610 on 189 degrees of freedom
Multiple R-squared:  0.6803,    Adjusted R-squared:  0.6753
F-statistic: 134.1 on 3 and 189 DF,  p-value: < 2.2e-16
```

We have improved this more dramatically this time, and our multiple R squared value rises to 0.6803 and adjusted R squared value rises to 0.6753.

We have dealt so far with just numerical data, lets see what happens if we use some categorical data by adding make to our linear regression.

```r
model = lm(price ~ `city.mpg` + `highway.mpg` + horsepower + make, data = automobile) #Cre
summary(model) #Review the results
```

```
Call:
lm(formula = price ~ city.mpg + highway.mpg + horsepower + make,
    data = automobile)

Residuals:
    Min      1Q  Median      3Q     Max
-5674.0 -1276.7   -29.0   929.2 10602.1

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        1776.89    3132.51   0.567   0.5713
city.mpg            117.34     154.00   0.762   0.4472
highway.mpg         -86.83     142.55  -0.609   0.5432
horsepower          108.92      11.26   9.674  < 2e-16 ***
makeaudi           3469.98    1764.92   1.966   0.0509 .
makebmw            9145.77    1676.07   5.457 1.70e-07 ***
makechevrolet     -3383.02    2201.06  -1.537   0.1262
makedodge         -3514.29    1702.78  -2.064   0.0406 *
makehonda         -2816.83    1642.05  -1.715   0.0881 .
makeisuzu         -2307.49    2288.03  -1.009   0.3147
makejaguar        10441.20    2125.48   4.912 2.11e-06 ***
makemazda          -872.09    1626.24  -0.536   0.5925
makemercedes-benz 15593.52    1759.28   8.864 1.05e-15 ***
makemercury       -4480.06    2896.83  -1.547   0.1238
makemitsubishi    -4092.26    1592.08  -2.570   0.0110 *
makenissan        -2838.87    1565.07  -1.814   0.0715 .
makepeugot         2518.31    1654.84   1.522   0.1299
makeplymouth      -3595.74    1732.54  -2.075   0.0395 *
makeporsche        8980.99    2004.73   4.480 1.37e-05 ***
makesaab           -362.34    1749.31  -0.207   0.8362
makesubaru        -3049.63    1645.62  -1.853   0.0656 .
maketoyota        -2366.11    1524.52  -1.552   0.1225
makevolkswagen     -852.84    1627.40  -0.524   0.6009
makevolvo          2101.16    1628.07   1.291   0.1986
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2470 on 169 degrees of freedom
```

```
Multiple R-squared:  0.9179,    Adjusted R-squared:  0.9068
F-statistic: 82.18 on 23 and 169 DF,  p-value: < 2.2e-16
```

So some things to note with this output. Firstly, our multiple R squared value rises again to 0.9179, and our adjusted R squared value rises to 0.908. But our output looks different. We can see that each make of car is represented, and that some have some * next to their entries, which denotes significance (we can see that these models are a lot better (significant) than others).

Lets check to see if this is right. Remember, we can create a subset of values by using the which command. Lets put all mercedez cars into a smaller dataframe. To compare, lets also put all hondas into a smaller dataframe.

```
# This code takes all rows where make is mercedes-benz and puts it into a new dataframe ca
mercedes <- automobile[which(automobile$make == 'mercedes-benz'),]
# This code takes all rows where make is honda and puts it into a new dataframe called hon
honda <- automobile[which(automobile$make == 'honda'),]
```

Now lets run our linear regression model, just for all mercedes cars and all honda cars.

```
model = lm(price ~ `city.mpg` + `highway.mpg` + horsepower, data = mercedes) #Create the l
summary(model) #Review the results
```

```
Call:
lm(formula = price ~ city.mpg + highway.mpg + horsepower, data = mercedes)

Residuals:
   59    60    61    62    63    64    65    66
-2842  -146  -218  3206  -436   436 -2220  2220

Coefficients: (1 not defined because of singularities)
            Estimate Std. Error t value Pr(>|t|)
(Intercept)    64192       4479  14.331 2.98e-05 ***
city.mpg      -23734       9725  -2.440   0.0586 .
highway.mpg    19454       8549   2.276   0.0719 .
horsepower        NA         NA      NA       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2394 on 5 degrees of freedom
Multiple R-squared:  0.9112,    Adjusted R-squared:  0.8757
F-statistic: 25.65 on 2 and 5 DF,  p-value: 0.002351
```

```r
model = lm(price ~ `city.mpg` + `highway.mpg` + horsepower, data = honda) #Create the line
summary(model) #Review the results
```

```
Call:
lm(formula = price ~ city.mpg + highway.mpg + horsepower, data = honda)

Residuals:
     Min       1Q   Median       3Q      Max
-1240.74  -236.10   -51.57   283.80  1586.98

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -14831.31    5357.73  -2.768 0.021817 *
city.mpg       395.26     255.35   1.548 0.156049
highway.mpg   -213.45     232.91  -0.916 0.383335
horsepower     231.52      41.21   5.619 0.000326 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 850.9 on 9 degrees of freedom
Multiple R-squared:  0.8722,    Adjusted R-squared:  0.8297
F-statistic: 20.48 on 3 and 9 DF,  p-value: 0.000233
```

We can see that the model is better for mercedes cars 0.9112 and 0.8757, than for honda cars 0.8722 and 0.8297.

Practical Questions

The main purpose of this week, is to see how we evaluate linear regression models. In our dataset, we have a number of different variables described below.

make: alfa-romero, audi, bmw, chevrolet, dodge, honda, isuzu, jaguar, mazda, mercedes-benz, mercury, mitsubishi, nissan, peugot, plymouth, porsche, renault, saab, subaru, toyota, volkswagen, volvo fuel-type: diesel, gas. aspiration: std, turbo. num-of-doors: four, two. body-style: hardtop, wagon, sedan, hatchback, convertible. drive-wheels: 4wd, fwd, rwd. engine-location: front, rear. wheel-base: continuous from 86.6 120.9. length: continuous from 141.1 to 208.1. width: continuous from 60.3 to 72.3. height: continuous from 47.8 to 59.8. curb-weight: continuous from 1488 to 4066. engine-type: dohc, dohcv, l, ohc, ohcf, ohcv, rotor. num-of-cylinders: eight, five, four, six, three, twelve, two. engine-size: continuous from 61 to 326. fuel-system: 1bbl, 2bbl, 4bbl, idi, mfi, mpfi, spdi, spfi. bore: continuous from 2.54

to 3.94. stroke: continuous from 2.07 to 4.17. compression-ratio: continuous from 7 to 23. horsepower: continuous from 48 to 288. peak-rpm: continuous from 4150 to 6600. city-mpg: continuous from 13 to 49. highway-mpg: continuous from 16 to 54. price: continuous from 5118 to 45400.

Create models to answer the following questions in a R Markdown document.

1. What is the best single variable for predicting price?
2. Rank, in order, each variable according to Adjusted R squared and also Root Mean Squared. Are the orders different?
3. Is there a combination of variables that improves our model? HINT: Combine the single variables that scored best in various ways to see if you can improve the model.
4. Now that you have found your best combination, answer the following questions:
5. Does the model perform better for certain makes of car?
6. Does the model work better for standard or turbo cars?
7. Does the model work for cars with two or four doors, or is there no difference?

Conclusion:

Evaluating regression models is essential to determine their accuracy, reliability, and suitability for a given problem. Metrics like MSE, RMSE, MAE, R^2, and adjusted R^2 provide valuable insights into a model's performance. Model validation techniques and residual analysis help ensure robust and interpretable regression models.