Computing with R

The `scale()` function

www.Stats-Lab.com

Twitter: @StatsLabDublin

Computing with R : The scale() function

Data:

The X and Y variables are the *wt* and *mpg* variables from the **mtcars** data set.

```
X <- mtcars$wt
Y <- mtcars$mpg
```

Computing with R : The scale() function

- The scale function is used to determine standardized values for each element in a data set.
- This is a data transformation technique that can be used in regression and clustering analysis.

Computing with R : The scale() function

- A standardized value for an element is simply the number of standard deviations away from the mean.
- Suppose z_i is the standardized value for x_i, an element of a sample data set with mean $\bar{x}$ and standard deviaton s.

$$z_i = \frac{x_i - \bar{x}}{s}$$

```
.....
[32,] -0.446876870
attr(,"scaled:center")
[1] 3.21725
attr(,"scaled:scale")
[1] 0.9784574
> mean(X)
[1] 3.21725
> sd(X)
[1] 0.9784574
```

Computing with R
The `dist()` function

www.Stats-Lab.com

Twitter: @StatsLabDublin

Computing with R : The dist() function

Data:

The first 8 rows and first 5 columns from the **mtcars** dataset.

```
X <- mtcars[1:8,1:5]
```

Computing with R : The dist() function

- The dist() function is used to compute the **distance matrix**.
- The distance matrix is comprised of distance measures for each pair of cases in the data set.
- A distance measure is a measure of **similarity** between two cases, based on a set of numeric values.

Computing with R : The dist() function

- The default distance measure is the **Euclidean Distance**.
- Given three numeric variables X, Y and Z, the Euclidean distance between case 1 and case 2 is computed as

$$ED_{12} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

Computing with R : The dist() function

Other types of distance measure that can be specified are

- the "maximum" measure,
- the "Manhattan" measure,
- the "Canberra" measure,
- the "binary" measure
- the "Minkowski" measure.

Computing with R : The dist() function

Transforming the Data

- Sometimes it would beneficial to transform one or more of the variables to prevent them being unduly influential, at the expense of other variables.
- One approach is to use standardized values. Standardization can be performed using the `scale()` function.
- Another approach is logarithmic transformation, which can be performed using the `log()` function.

Computing with R

Useful Regression Functions

www.Stats-Lab.com

Twitter: @StatsLabDublin

Useful Functions for Linear Regression

Data:

The `X` and `Y` variables are the *wt* and *mpg* variables from the **mtcars** data set.

```
X <- mtcars$wt
Y <- mtcars$mpg

Fit <- lm(Y~X)
```

Useful Functions for Linear Regression

- `summary()` - very detailed statistical summary of the fitted model,
- `coef()` - prints out the regression coefficients for the fitted model,
- `fitted()` - prints out the fitted values for the fitted model,
- `resid()` - prints out the residual for the fitted model,
- `anova()` - prints out the ANOVA table for the fitted model.

Computing with R

Confidence Intervals for Regression Coefficients

www.Stats-Lab.com

Twitter: @StatsLabDublin

Confidence Intervals for Regression Coefficients

Data:

The `X1`, `X2` and `Y` variables are the *wt* , *hp* and *mpg* variables from the **mtcars** data set.

```
X1 <- mtcars$wt
X2 <- mtcars$hp
Y <- mtcars$mpg

Fit <- lm(Y~X1+X2)
```

Confidence Intervals for Regression Coefficients

- To compute the confidence intervals, we use the `confint()` function, specifying the name of the fitted model.
- The default confidence level is 95%. We can adjust it by changing the `level=` argument. (e.g. `level = 0.90`).
- We can specify the confidence interval for particular regression coefficients using the `parm=` argument.

Computing with R

Standardized Regression Coefficients

www.Stats-Lab.com

Twitter: @StatsLabDublin

Standardized Regression Coefficients

Data:

The `X` and `Y` variables are the *wt* and *mpg* variables from the **mtcars** data set.

```
X1 <- mtcars$wt
X2 <- mtcars$hp
Y <- mtcars$mpg

Fit.u <- lm(Y ~ X1 + X2)
```

Standardized Regression Coefficients

- In some statistical analyses, it is useful to work with standardized values, rather than observed values.
- A standardized value for an element is simply the number of standard deviations away from the mean.

Standardized Regression Coefficients

- To compute a regression model on standardized values, use the `scale()` function to standardize all of the relevant variables.

```
Fit.s <- lm( scale(Y) ~ scale(X1) +
  scale(X2) )
```