

PSTAT 126

Regression Analysis

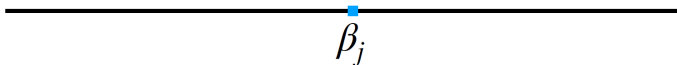
Laura Baracaldo & Rodrigo Targino

Lecture 8 & 9
Confidence intervals and Diagnostics

Confidence intervals (CIs) for β

We consider the MLR model: $\mathbf{y} = \mathbf{X}\beta + \epsilon$, $\epsilon \sim N_n(\mathbf{0}, \sigma^2 \mathbf{I}_n)$

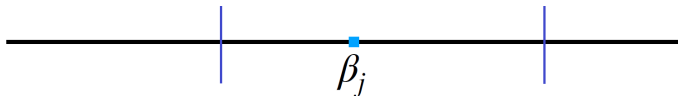
Confidence intervals (CIs) provide an alternative way of expressing the uncertainty in the estimates of β . We construct confidence intervals based on a confidence level $(1 - \alpha) * 100\%$, which represents the proportion of intervals among all possible intervals that contain the true value of the unknown parameter.



Confidence intervals (CIs) for β

We consider the MLR model: $\mathbf{y} = \mathbf{X}\beta + \epsilon$, $\epsilon \sim N_n(\mathbf{0}, \sigma^2 \mathbf{I}_n)$

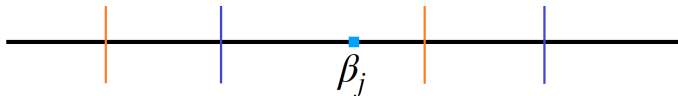
Confidence intervals (CIs) provide an alternative way of expressing the uncertainty in the estimates of β . We construct confidence intervals based on a confidence level $(1 - \alpha) * 100\%$, which represents the proportion of intervals among all possible intervals that contain the true value of the unknown parameter.



Confidence intervals (CIs) for β

We consider the MLR model: $\mathbf{y} = \mathbf{X}\beta + \epsilon$, $\epsilon \sim N_n(\mathbf{0}, \sigma^2 \mathbf{I}_n)$

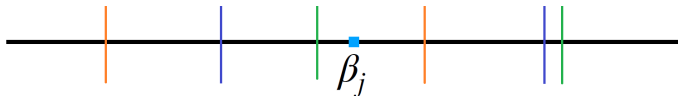
Confidence intervals (CIs) provide an alternative way of expressing the uncertainty in the estimates of β . We construct confidence intervals based on a confidence level $(1 - \alpha) * 100\%$, which represents the proportion of intervals among all possible intervals that contain the true value of the unknown parameter.



Confidence intervals (CIs) for β

We consider the MLR model: $\mathbf{y} = \mathbf{X}\beta + \epsilon$, $\epsilon \sim N_n(\mathbf{0}, \sigma^2 \mathbf{I}_n)$

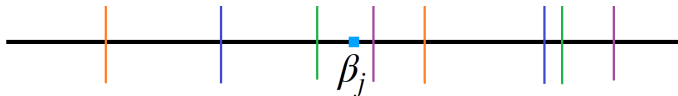
Confidence intervals (CIs) provide an alternative way of expressing the uncertainty in the estimates of β . We construct confidence intervals based on a confidence level $(1 - \alpha) * 100\%$, which represents the proportion of intervals among all possible intervals that contain the true value of the unknown parameter.



Confidence intervals (CIs) for β

We consider the MLR model: $\mathbf{y} = \mathbf{X}\beta + \epsilon$, $\epsilon \sim N_n(\mathbf{0}, \sigma^2 \mathbf{I}_n)$

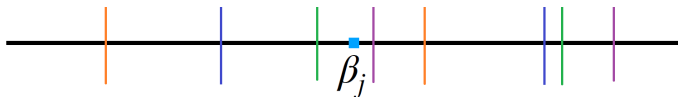
Confidence intervals (CIs) provide an alternative way of expressing the uncertainty in the estimates of β . We construct confidence intervals based on a confidence level $(1 - \alpha) * 100\%$, which represents the proportion of intervals among all possible intervals that contain the true value of the unknown parameter.



Confidence intervals (CIs) for β

We consider the MLR model: $\mathbf{y} = \mathbf{X}\beta + \epsilon$, $\epsilon \sim N_n(\mathbf{0}, \sigma^2 \mathbf{I}_n)$

Confidence intervals (CIs) provide an alternative way of expressing the uncertainty in the estimates of β . We construct confidence intervals based on a confidence level $(1 - \alpha) * 100\%$, which represents the proportion of intervals among all possible intervals that contain the true value of the unknown parameter.



Each data set will produce a different confidence interval. If we construct an infinite number of intervals based on an infinite number of data sets, approximately 95% of them will contain the true value of β_j .

CIs for individual parameters

For individual parameter β_j , the CI takes the form:

$$\hat{\beta}_j \pm t_{(1-\alpha/2; n-p^*)} SE(\hat{\beta}_j)$$

Where $SE = \hat{\sigma} \sqrt{(\mathbf{X}^T \mathbf{X})_{jj}^{-1}}$, is the standard error of $\hat{\beta}_j$.

Duality between CIs and hypothesis tests: For a significance level α , we can test the hypothesis: $H_0 : \beta_j = 0$ vs $H_1 : \beta_j \neq 0$, by using the CI with confidence level $(1 - \alpha)$:

If the CI contains zero, we fail to reject H_0

CIs for Predictions

There are two kind of predictions made from regression models:

- 1 Prediction of response mean: We want to predict $\mu_0 = E(y_0|x_0) = x_0^T \beta$, which represents the “average” response for a subject with characteristics x_0 .
- 2 Prediction of a future observation: We seek to predict $y_0 = \mu_0 + \epsilon_0 = x_0^T \beta + \epsilon_0$, which represents the response of a new observation with characteristics x_0 .

Point-wise Prediction: For a new set of predictors x_0 , the predicted response is $\hat{y}_0 = x_0^T \hat{\beta}$ for both cases 1 and 2.

CI for Predictions

Interval Prediction: We need to assess the uncertainty of this prediction as decision makers need more than just a point estimate to make rational choices.

- ❶ $100 * (1 - \alpha)\%$ CI for the response mean:

$$\hat{y}_0 \pm t_{(1-\alpha/2; n-p^*)} \hat{\sigma} \sqrt{x_0^T (\mathbf{X}^T \mathbf{X})^{-1} x_0}$$

- ❷ $100 * (1 - \alpha)\%$ CI for a single future observation:

$$\hat{y}_0 \pm t_{(1-\alpha/2; n-p^*)} \hat{\sigma} \sqrt{1 + x_0^T (\mathbf{X}^T \mathbf{X})^{-1} x_0}$$

Regression Diagnostics

Consider the MLR model:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N_n(\mathbf{0}, \sigma^2 \mathbf{I}_n)$$

Inference from this model depends on several assumptions, that should be checked by using *regression diagnostics* before using the model. Potential problems can be divided into three categories:

- 1 **Error:** We have assumed $\boldsymbol{\epsilon} \sim N_n(\mathbf{0}, \sigma^2 \mathbf{I}_n)$, which involves three assumptions on the errors: 1) Independence, 2) Constant variance and 3) Normality.
- 2 **Unusual observations** There might be a few observations that do not fit the model.
- 3 **Model structure:** We have assumed linearity of the response:
 $E(\mathbf{y}) = \mathbf{X}\boldsymbol{\beta}$.

Checking Error Assumptions

Note that errors ϵ are not observable, however we can examine the residuals $\hat{\epsilon}$ as proxies. Recall that:

$$\hat{\epsilon} = \mathbf{y} - \hat{\mathbf{y}} = (\mathbf{I} - \mathbf{H})\epsilon$$

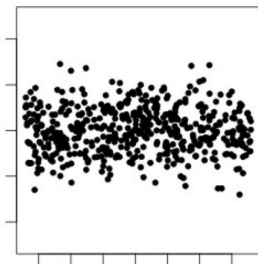
with $\mathbf{H} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$. Although the errors and the residuals are not interchangeable in general (residuals have no constant variance, neither are they independent, since $Var(\hat{\epsilon}) = \sigma^2(\mathbf{I} - \mathbf{H})$), diagnostics can reasonably be applied to the residuals in order to check assumptions on the error.

Checking Error Assumptions

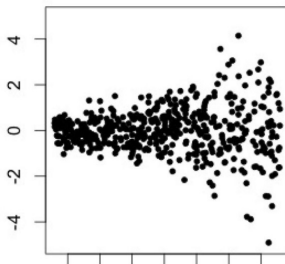
- ① **Constant Variance:** The most useful diagnostic is the plot of the residuals $\hat{\epsilon}$ (vertical axis) against the fitted values \hat{y} (horizontal axis).

If $Var(\epsilon) = \sigma^2$, $i = 1, \dots, n$, we should observe constant symmetrical variation (*Homoscedasticity*) in the vertical direction.

Homoscedasticity



Heteroscedasticity



Checking Error Assumptions

- 2 **Normality:** Recall all inference (hypotheses tests/CIs) is based on the assumption of normal errors.
- The most common used diagnostic is the **Q-Q plot**, which compares the residuals to ideal normal distributions based on the quantiles. We plot the sorted residuals against $\phi^{-1}\left(\frac{i}{n+1}\right)$, $i = 1, \dots, n$.
- A more formal way to test normality is by using the **Shapiro-Wilk** test, whose null hypothesis is H_0 : Residuals are normal.

Checking Error Assumptions

- ③ **Uncorrelated errors:** Difficult to check since there are too many possible patterns of correlation that may occur. Examples of correlated data: Temporal & Spatial: Variations are explained by structural dependencies in time and space.
- In temporal data it would be useful to plot *time* against *residuals*. If the errors were uncorrelated, we would expect a random scatter of points above and below $\hat{e} = 0$.
- Alternative plot: Plot successive pair of residuals $(\hat{e}_i, \hat{e}_{i+1})$. Any trend will mean correlation.
- *Durbin-Watson Test*: The null hypothesis: H_0 : Uncorrelated errors. This test uses the statistic:

$$DW = \frac{\sum_{i=2}^n (\hat{e}_i - \hat{e}_{i-1})^2}{\sum_{i=1}^n \hat{e}_i^2}$$

Species Example: 95% Confidence Intervals

```
summary(fit1)$coefficients
```

```
##              Estimate Std. Error    t value    Pr(>|t|)
## (Intercept)  7.068220709 19.15419782  0.369016796 7.153508e-01
## Area        -0.023938338  0.02242235 -1.067610554 2.963180e-01
## Elevation    0.319464761  0.05366280  5.953187968 3.823409e-06
## Nearest      0.009143961  1.05413595  0.008674366 9.931506e-01
## Scruz        -0.240524230  0.21540225 -1.116628222 2.752082e-01
## Adjacent     -0.074804832  0.01770019 -4.226216850 2.970655e-04
```

```
CIs<- cbind(summary(fit1)$coefficients[, 1]-qt(0.975, fit1$df.residual)*summary(fit1)$coefficients[, 2] ,
            summary(fit1)$coefficients[, 1]+qt(0.975, fit1$df.residual)*summary(fit1)$coefficients[, 2])
```

CIs ## Using Formula

```
##              [,1]      [,2]
## (Intercept) -32.4641006 46.60054205
## Area        -0.0702158  0.02233912
## Elevation    0.2087102  0.43021935
## Nearest      -2.1664857  2.18477363
## Scruz        -0.6850926  0.20404416
## Adjacent     -0.1113362 -0.03827344
```

confint(fit1) #Using Built in Function

```
##              2.5 %      97.5 %
## (Intercept) -32.4641006 46.60054205
## Area        -0.0702158  0.02233912
## Elevation    0.2087102  0.43021935
## Nearest      -2.1664857  2.18477363
## Scruz        -0.6850926  0.20404416
## Adjacent     -0.1113362 -0.03827344
```


Species Example: Prediction Interval

```
x0<- data.frame(Area=20.6, Elevation= 46, Nearest=1.9, Scrutz=8.0, Adjacent=0.78 )
pred.y0<-t(x0.vector)%*%fit1$coefficients ; pred.y0
```

```
##           [,1]
## [1,] 19.3053
se2.betas<- t(x0.vector)%*(solve(t(X)%*%X))%*%x0.vector
sigma<- sigma(fit1)
predmean.CI<- c(pred.y0 - sigma*qt(0.975,fit1$df.residual)*sqrt(se2.betas),
                pred.y0 + sigma* qt(0.975,fit1$df.residual)*sqrt(se2.betas)); predmean.CI #Using Formula
```

```
## [1] -16.42255  55.03315
predict(fit1, newdata = x0, interval = "confidence") #Using Built-in function
```

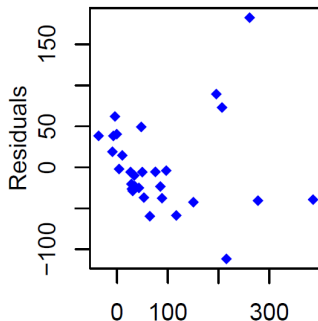
```
##           fit           lwr           upr
## 1 19.3053 -16.42255 55.03315
pred.CI<- c(pred.y0 - sigma*qt(0.975,fit1$df.residual)*sqrt(1+se2.betas),
            pred.y0 + sigma* qt(0.975,fit1$df.residual)*sqrt(1+se2.betas)); pred.CI #Using Formula
```

```
## [1] -111.5146  150.1252
predict(fit1, newdata = x0, interval = "prediction") #Using Built-in function
```

```
##           fit           lwr           upr
## 1 19.3053 -111.5146 150.1252
```

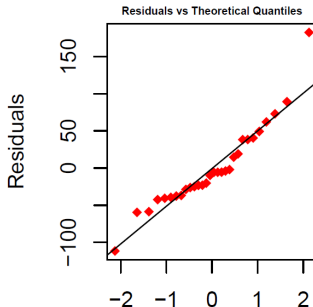
Species Example: Checking Homoscedasticity

```
par(mar = c(5, 5, 1, 2))  
plot(fitted(fit1), residuals(fit1), xlab="",  
     ylab="", col="blue", pch=18);mtext(side=2, text="Residuals", line=2)
```



Species Example: Checking Normality

```
par(mar = c(5, 5, 1, 2))
qqnorm(residuals(fit1), ylab="Residuals",
       main="", pch=18, col="red")
qqline(residuals(fit1));title("Residuals vs Theoretical Quantiles", cex.main=0.5)
```



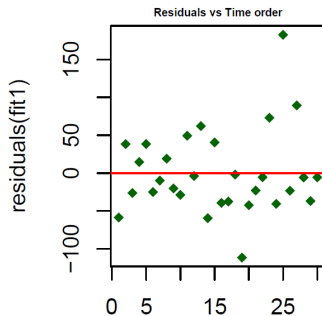
Species Example: Checking Normality

```
shapiro.test(residuals(fit1))
```

```
##  
##  Shapiro-Wilk normality test  
##  
## data:  residuals(fit1)  
## W = 0.91351, p-value = 0.01826
```

Species Example: Checking Correlation

```
par(mar = c(5, 5, 1, 2))  
plot(1:length(gala$Species), residuals(fit1), col="darkgreen", pch=18)  
abline(h=0. ,col="red")  
title("Residuals vs Time order", cex.main=0.5)
```



Species Example: Checking Correlation

```
dwtest( Species ~ Area+Elevation+Nearest+ Scruz+ Adjacent, data=gala)
```

```
##  
## Durbin-Watson test  
##  
## data: Species ~ Area + Elevation + Nearest + Scruz + Adjacent  
## DW = 2.4759, p-value = 0.9017  
## alternative hypothesis: true autocorrelation is greater than 0
```

Finding Unusual Observations

Consider the MLR model:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim N_n(\mathbf{0}, \sigma^2 \mathbf{I}_n)$$

Which fits observations $(y_1, \mathbf{X}_1), (y_2, \mathbf{X}_2) \dots, (y_n, \mathbf{X}_n)$. However there may be three kind of observations that don't "agree" with this model.

- **High Leverage Points:** Data points that are extreme in the predictor space. (Extreme values \mathbf{X}_i).
- **Outliers:** Extreme observations in the response given the predictors. (Unusual values y_i given \mathbf{X}_i).
- **Influential Observations:** Data points that change the fit of the model substantially.

Leverage

Recall the hat matrix: $\mathbf{H} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$. The **leverage score** for the i^{th} independent observation \mathbf{X}_i are given by the diagonal elements of \mathbf{H} : $h_i = [\mathbf{H}]_{ii}$.

Interpretation: The leverage score can be seen as the “weighted” distance between \mathbf{X}_i and the average point of the \mathbf{X}_i ’s. It can also be interpreted as the degree by which y_i influences \hat{y}_i .

Properties:

- ① $0 \leq h_{ii} \leq 1 \quad \forall i = 1, \dots, n.$
- ② $\sum_{i=1}^n h_{ii} = p^* = p + 1$

Leverage properties:

- ❶ **Proof:** We use that the hat matrix is symmetric and idempotent:

$$\begin{aligned} \mathbf{H}^2 = \mathbf{H} &\Rightarrow [\mathbf{H}]_{ii}^2 = [\mathbf{H}]_{ii} \\ &\Rightarrow h_{ii} = h_{ii}^2 + \sum_{j \neq i} h_{ij} h_{ji} = h_{ii}^2 + \sum_{j \neq i} h_{ij}^2 \geq 0 \\ \sum_{j \neq i} h_{ij}^2 \geq 0 &\Rightarrow h_{ii} \geq h_{ii}^2 \Rightarrow h_{ii} \leq 1 \\ &\Rightarrow 0 \leq h_{ii} \leq 1 \end{aligned}$$

- ❷ **Proof:** We use $\text{Tr}(\mathbf{A}^T \mathbf{B}) = \text{Tr}(\mathbf{A} \mathbf{B}^T)$

$$\begin{aligned} \sum_{i=1}^n h_{ii} &= \text{Tr}(\mathbf{H}) = \text{Tr}(\mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T) \\ &= \text{Tr}(\mathbf{X}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1}) = \text{Tr}(\mathbf{I}_{p^*}) = p^* = p + 1 \end{aligned}$$

Effect on Residual Variance

Recall that $\hat{\epsilon} = (I - H)y \Rightarrow \text{Var}(\hat{\epsilon}) = \sigma^2(I - H)$. This implies:

$$\text{Var}(\hat{\epsilon}_i) = \sigma^2(1 - h_i)$$

This means that large leverage score h_i will make $\text{Var}(\hat{\epsilon}_i) = \text{Var}(y_i - \hat{y}_i)$ small, so that the regression line will be attracted toward y_i .

Extreme X_i 's \Rightarrow Large values of $h_i \Rightarrow$ Large influence of \hat{y}_i on y_i

Since $\sum_{i=1}^n h_i = p^* \Rightarrow \bar{h} = \frac{p^*}{n}$. We should look at observations that greatly exceed \bar{h} . **The Rule of thumb:** We consider large leverage score $h^* > 2\bar{h}$.

High Leverage points - Savings data

Savings data: - sr: savings rate - pop15: % population under age 15 - pop75: % population under age 75 - dpi: per-capita disposable income - ddpi: percent growth rate of dpi

```
library(faraway);data(savings)
```

```
## Warning: package 'faraway' was built under R version 4.1.3
```

```
lmod<- lm(sr ~ pop15 + pop75+ dpi+ ddpi, savings)
X<- model.matrix(lmod)
hatv <- diag(X%*(solve(t(X)%*X))%*t(X)) #Using formula
hatv2<- hatvalues(lmod) #Using built in formula
all.equal(hatv, hatv2)
```

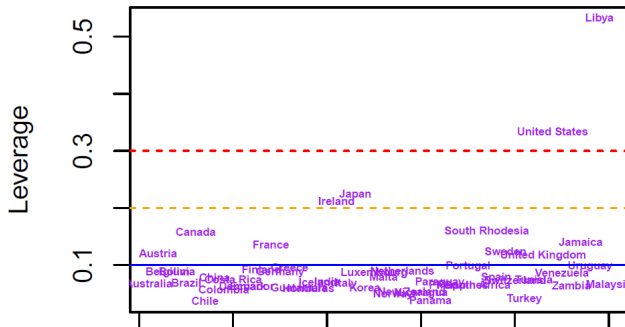
```
## [1] TRUE
```

```
sum(hatv2) # sum h_i = p+1
```

```
## [1] 5
```

High Leverage points - Savings data

```
data.lev<- data.frame(index=seq(length(hatv)),  
  Leverage=hatv, namesC=rownames(savings))  
par(mar = c(4, 4, 0.5, 0.5))  
plot(Leverage ~ index, data=data.lev, col="white", pch=NULL)  
text(Leverage ~ index, labels=namesC, data=data.lev, cex=0.4, font=2, col="purple")  
abline(h = sum(hatv2)/dim(data.lev)[1], col="blue")  
abline(h = 2*sum(hatv2)/dim(data.lev)[1], col="orange", lty=2)  
abline(h = 3*sum(hatv2)/dim(data.lev)[1], col="red", lty=2)
```



Outliers

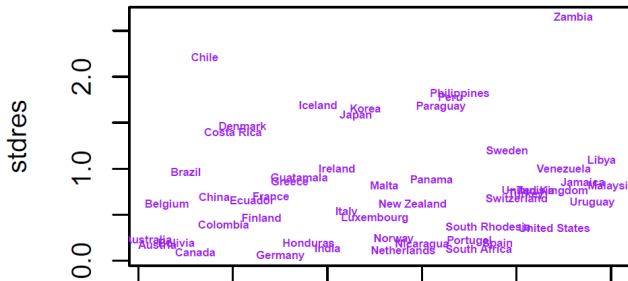
We define the **outliers** as points that does not fit the model well, data point for which $y_i - \hat{y}_i$ is large. Outliers may be the result of wrong measurements, or true unusual observations. How do we identify outliers?

- *Residuals vs fitted values* plots
- We consider the **Standardized residuals**: $r_i = \frac{y_i - \bar{y}_i}{\hat{\sigma}\sqrt{1-h_i}}$. We use *the rule of thumb*: Outliers $|r_i| \geq 3$

Outliers - Savings data

```
r1<- residuals(lmod)/(sigma(lmod)*sqrt(1-hatv)) ## Using formula
r2<- rstandard(lmod) ##Built-in function
all.equal(r1,r2)
```

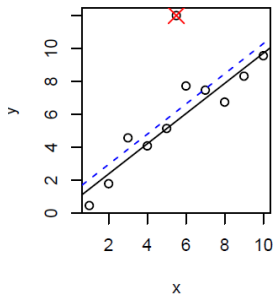
```
## [1] TRUE
data.sres<- data.frame(index=seq(length(r2)),
                        stdres=abs(r2), namesC=rownames(savings))
par(mar = c(4, 4, 0.5, 0.5))
plot(stdres ~ index, data=data.sres, col="white", pch=NULL)
text(stdres ~ index, labels=namesC, data=data.sres, cex=0.4, font=2, col="purple")
abline(h=3, col="red", lty=2)
```



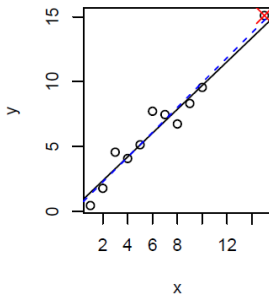
Influential Observations

An **influential point** is one whose removal from the data set would cause a significant change in the model. It may or may not be an outlier and may or may not be a high leverage point. (They tend to have at least one of these two properties)

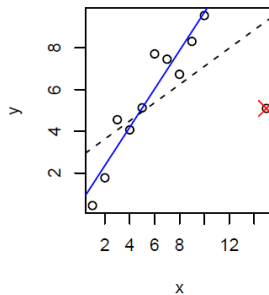
Outlier



High Leverage



Influential



Cook's Distance

We use Cook Statistics for influence diagnostics, which is defined as:

$$D_i = \frac{(\hat{y} - \hat{y}_{(i)})^T (\hat{y} - \hat{y}_{(i)})}{p^* \hat{\sigma}^2} = \frac{1}{p^*} r_i^2 \frac{h_i}{1 - h_i}$$

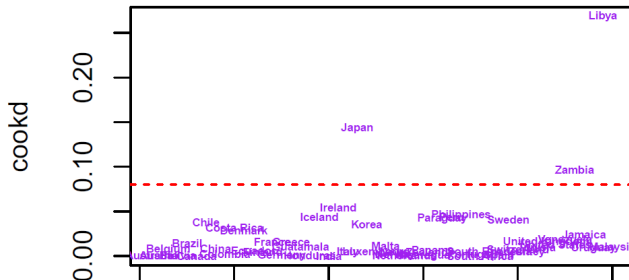
Where $\hat{y}_{(i)}$ indicates the fit without case i , r_i is the standardized residual, h_i is the leverage score. **The rule of thumb:** $D_i > 4/n$

Influential Observations - Savings data

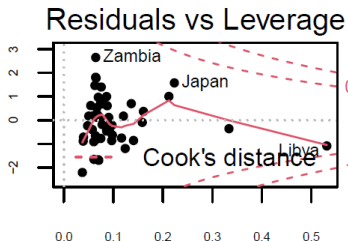
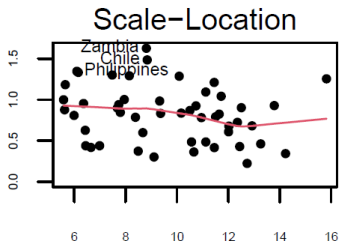
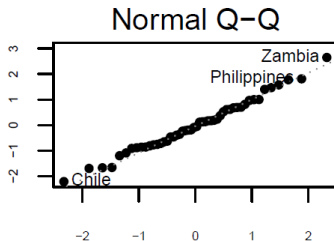
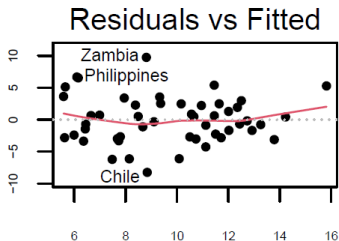
```
cook1<- (r1^2/dim(X)[2])* hatv/(1-hatv) ## Using formula
cook2<- cooks.distance(lmod) ##Built-in function
all.equal(cook1,cook2)

## [1] TRUE

data.cook<- data.frame(index=seq(length(r2)),
                        cookd=abs(cook2), namesC=rownames(savings))
par(mar = c(4, 4, 0.5, 0.5))
plot(cookd ~ index, data=data.cook, col="white", pch=NULL)
text(cookd ~ index, labels=namesC, data=data.cook, cex=0.4, font=2, col="purple")
abline(h=4/dim(X)[1], col="red", lty=2)
```



Unusual Observations in a single plot



What to do with Unusual Observations?

What should we do once we find such observations?

- 1 Check if there is data-entry error
- 2 Exclude the points
- 3 Try re-including them later if the model is changed

Checking Model Structure

Diagnostics can be used to detect deviations to the linearity assumption $E(y) = X\beta$. **Residuals** can show any sign of systematic structure that deviates from the linear assumption.

- Residuals against fitted values
- Residuals against predictors

Model Structure - Savings data

```
par(mar = c(3, 2, 1.5, 0.5))  
plot(lmod, cex.main=1, cex.lab=0.5, cex.axis=0.5, pch=20)
```

