Data Analysis

Business Analytics
Stefan Feuerriegel

Today's Lecture

Objectives

1. Understanding the concept of linear regressions
2. Testing necessary requirements to perform ordinary least squares
3. Selecting and comparing models in terms of fit

Outline

1. Linear Models
2. Model Selection
3. Linear Prediction Models

Outline

Linear Regression

```
d <- read.csv("countries.csv", sep=",")
m <- lm(d$PerCapitaIncome ~ d$Literacy)
summary(m)
```

```
##
## Call:
## lm(formula = d$PerCapitaIncome ~ d$Literacy)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -20918  -8268  -1370   9517  18578
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -28153.4    13062.8  -2.155  0.04577 *
## d$Literacy     532.7      148.1   3.598  0.00222 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11620 on 17 degrees of freedom
## Multiple R-squared:  0.4323,Adjusted R-squared:  0.3989
## F-statistic: 12.94 on 1 and 17 DF,  p-value: 0.00222
```

Notation

- Alternatively, data can be specified via parameter `data=`

```r
# Both variants yield the same result
lm(d$PerCapitaIncome ~ d$Literacy)
lm(PerCapitaIncome ~ Literacy, data=d)
```

- Operator `dependent ~ .` uses all other columns as regressors

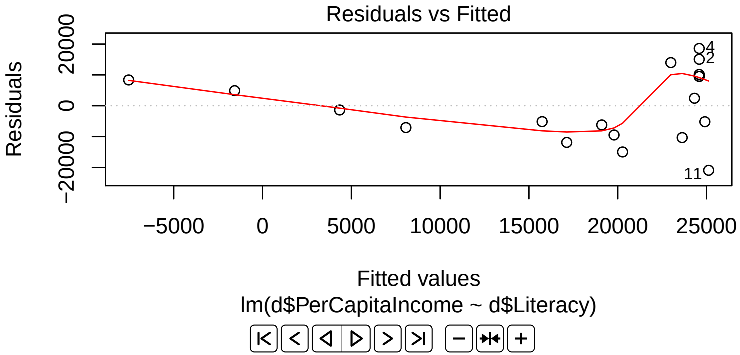
```r
colnames(d)

## [1] "Country"          "PerCapitaIncome" "Literacy"
## [5] "LifeExpectancy"
```

```r
# Both variants yield the same result
lm(PerCapitaIncome ~ Country + Literacy +
   InfantMortality + LifeExpactancy, data=d)
lm(PerCapitaIncome ~ ., data=d)
```

Regression Diagnostics

Perform default regression diagnostics, such as plots with residuals vs fitted values, and Q-Q plot of residuals

```
plot(m) # show 4 plots with regression diagnostics
```



Variance Inflation Factors

- Quantifies the severity of multicollinearity
- Measures how much the variance (the square of the estimate's standard deviation) of an estimated regression coefficient has increased because of collinearity
- Load necessary library `car`

```
library(car) # load necessary library
```

- Calculate via `vif(m)` for an already estimated model `m`

```
m <- lm(d$PerCapitaIncome ~ d$Literacy + d$LifeExpectancy)
vif(m)

##       d$Literacy d$LifeExpectancy
##         2.239719         2.239719

vif(m) > 4 # problem?

##       d$Literacy d$LifeExpectancy
##            FALSE            FALSE
```

- Indication of multicollinearity if above 4

Outline

AIC and BIC

- `logLik(m)` extracts likelihood

```r
# 3 degrees of freedom: alpha, beta, epsilon
m <- lm(d$PerCapitaIncome ~ d$Literacy)
logLik(m)[1] # extract likelihood from package stats
```

```
## [1] -203.7486
```

- Use commands `AIC(m)` and `BIC(m)` to calculate each criterion

```r
AIC(m)
```

```
## [1] 413.4971
```

```r
2*3-2*logLik(m)[1]
```

```
## [1] 413.4971
```

```r
BIC(m)
```

```
## [1] 416.3304
```

```r
3*log(18)-2*logLik(m)[1]
```

```
## [1] 416.1682
```

Outline

Prediction with Linear Models

- An already estimated linear model $\boldsymbol{y} = \alpha + \beta_1 \boldsymbol{x}_1 + \ldots + \beta_k \boldsymbol{x}_k + \boldsymbol{\varepsilon}$ can be used to evaluate with new values $x_1', \ldots, x_k'$ giving

$$y' = \alpha + \beta_1 x_1' + \ldots + \beta_k x_k'$$

- Use the command `predict(m, newdata=d)` for a model `m` and new data `d`
- Example

```
m <- lm(PerCapitaIncome ~ Literacy, data=d)
nd <- data.frame(Literacy = 5)
predict(m, newdata = nd)
```

```
##         1
## -25489.84
```

Summary: Commands

Estimating Linear Models

`cor(x, y)`	Correlation coefficient
`cor.test(x, y)`	*t*-Test for Pearson correlation coefficient
`lm(y ~ x1 + ...)`	Estimate linear model
`summary(model)`	Detailed regression statistics
`abline(model)`	Draw line of best fit

Verifying Assumptions of OLS Estimator

`plot(model)`	Plots with regression diagnostics
`bptest(model)`	Breusch-Pagan test $\rightarrow$ heteroscedasticity
`acf(d)`	Plot autocorrelation function
`dwtest(model)`	Durbin-Watson test $\rightarrow$ non-autocorrelation
`vif(model)`	Variance Inflation Factor $\rightarrow$ no linear dependence
`kappa(X)`	Condition number of matrix

Model Selection and Prediction

`logLik(model)[1]`	Model likelihood
`AIC(model)`	Akaike Information Criterion
`BIC(model)`	Bayesian Information Criterion
`predict(model, newdata=d)`	Prediction model outcome for new data