Data Analysis

Exercise: Business Intelligence (Part 4)

Summer Term 2014

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

Outline

Correlation

BI Case Study

Question: Is there a correlation between value of players and goals of teams playing in the German Soccer League?

Data: bundesliga2009.csv

```
Club;PlayerValue;Goals;Points
Bayern;10.4;34;33
Wolfsburg;5.34;32;24
HSV;4.38;34;31
...
```

Accessing Data

- Reading data from file

```r
d <- read.csv("bundesliga2009.csv", sep = ";", header = TRUE)
```

- Showing first rows

```r
head(d)

##          Club PlayerValue Goals Points
## 1      Bayern       10.40    34     33
## 2   Wolfsburg        5.34    32     24
## 3         HSV        4.38    34     31
## 4 Leverkusen         4.11    35     35
## 5      Bremen        4.05    32     28
## 6   Stuttgart        4.01    16     16
```

- Calculating total value of all players

```r
sum(d$PlayerValue)

## [1] 57.09
```

Data as Point Plot

```
plot(d$PlayerValue, d$Goals, main="Bundesliga Season 2009/10",
xlab="Current Value of Players", ylab="Goals after 17 weeks")
text(d$PlayerValue, d$Goals, d$Club)
```

Pearson Correlation Coefficient

- Measures the linear correlation (dependence) between two variables
- For a stochastic variable

$$\rho_{X,Y} = \frac{\mathrm{Cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{\mathrm{E}[(X - \mathrm{E}[X])(Y - \mathrm{E}[Y])]}{\sigma_X \sigma_Y} \in [-1, +1]$$

- For a finite sample

$$r = \frac{\sum_i (x_i - \mu_x)(y_i - \mu_y)}{\sqrt{\sum_i (x_i - \mu_x)^2} \sqrt{\sum_i (y_i - \mu_y)^2}} \in [-1, +1]$$

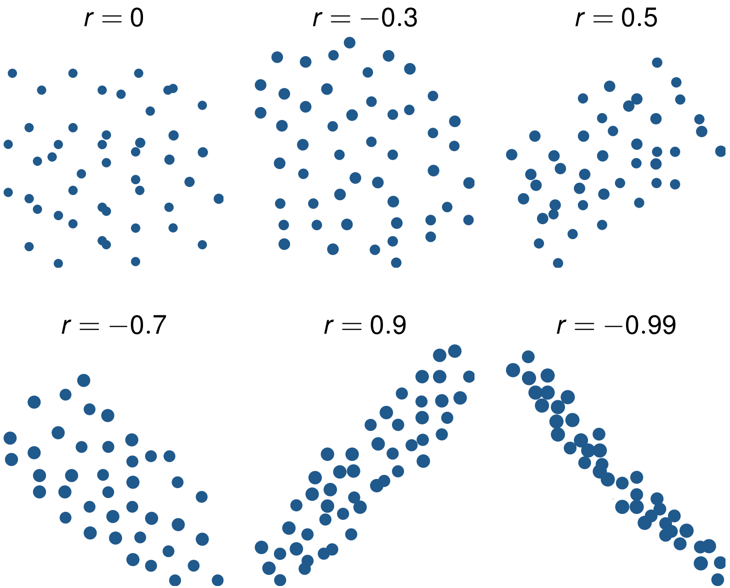
  with mean μ_x and μ_y respectively

```r
cor(d$PlayerValue, d$Goals)

## [1] 0.6525
```

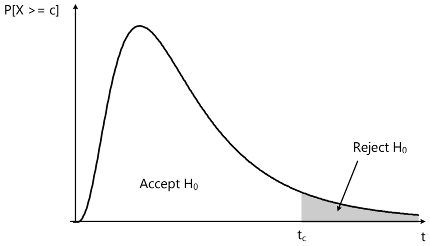
- Other correlation coefficients (such as Spearman) exist in cases when data is not normally distributed

Pearson Correlation Coefficient



Hypothesis Testing

- Results are called statistically significant if it has been predicted as unlikely to have occurred by chance alone, according to a pre-determined threshold probability, the significance level
- H_0: null hypothesis associated with a contradiction to a theory
- H_A: alternative hypothesis associated with a theory to prove
- P-value gives probability, assuming the null hypothesis is true, of observing a result t at least as extreme as the test statistic t_c



Example: Clairvoyant Card Game

- A person is tested for clairvoyance $\rightarrow$ asked which of the four suits 25 randomly chosen cards belongs to
- The number of hits (correct answers) is called X
- To find evidence of clairvoyance

$$H_0 : p = \frac{1}{4} \text{ (just guessing)} \qquad H_A : p > \frac{1}{4} \text{ (true clairvoyant)}$$

- What is the critical number t_c of hits, at which we assume clairvoyance?

$$\mathrm{P}\left[\text{reject } H_0 \,|\, H_0 \text{ is valid}\right] = \mathrm{P}\left[X \geq t_c \,|\, p = 1/4\right] \leq \alpha$$

  with maximum acceptable probability α of false positives
- We choose the smallest t_c that gives a probability below α $\rightarrow$ e. g. with $\alpha = 1\,\%$, we get $t_c = 13$

t-Test for Pearson Correlation Coefficient

- Test measures if Pearson correlation coefficients are significant given a threshold
- Null hypothesis H_0: $\rho = 0$ (i.e. no linear relationship)
- Alternative hypothesis H_A: $\rho \neq 0$ (or $\rho > 0 \vee \rho < 0$)
- Variable $t = \frac{r\sqrt{n-2}}{1-r^2}$ has Student's t-distribution in the null case, with

ρ	correlation of the population
r	correlation of the sample
n	size of sample

Pearson Correlation Coefficient

```
cor.test(d$PlayerValue, d$Goals)

##
##  Pearson's product-moment correlation
##
## data:  d$PlayerValue and d$Goals
## t = 3.445, df = 16, p-value = 0.003332
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.267 0.858
## sample estimates:
##    cor
## 0.6525
```

$\rightarrow$ Although the correlation is relatively small, the *P*-value of $0.003332 < 0.01$ indicates a significant linear dependence at the 1 %-significance level

Outline

Linear Models

- Linear Model: $\boldsymbol{y} = \alpha + \beta_1 \boldsymbol{x}_1 + \ldots + \beta_k \boldsymbol{x}_k + \boldsymbol{\varepsilon}$
  - Given $\boldsymbol{y}$ named observations, response or dependent variable
  - Given $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$ named regressors, exogenous or independent variables
  - Given residuals $\boldsymbol{\varepsilon}$ with entries $\varepsilon_1, \ldots, \varepsilon_N$
- Estimate intercept α and the coefficients $\beta_1, \ldots, \beta_k$ by minimizing error terms $\boldsymbol{\varepsilon}$, e.g. via ordinary least squares (OLS) estimator

$$\min_{\alpha,\beta_1,\ldots,\beta_k} \|\boldsymbol{\varepsilon}\| = \min_{\alpha,\beta_1,\ldots,\beta_k} \|\boldsymbol{y} - (\alpha + \beta_1 \boldsymbol{x}_1 + \ldots + \beta_k \boldsymbol{x}_k)\|$$

$\rightarrow$ important to test assumptions to avoid confounded results

Linear Regression

```
m <- lm(d$Goals ~ d$PlayerValue)
summary(m)

##
## Call:
## lm(formula = d$Goals ~ d$PlayerValue)
##
## Residuals:
##    Min      1Q Median      3Q     Max
## -10.51  -4.95   1.26   3.63   9.54
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)     15.912      2.586    6.15  1.4e-05 ***
## d$PlayerValue    2.323      0.674    3.44   0.0033 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.17 on 16 degrees of freedom
## Multiple R-squared:  0.426, Adjusted R-squared:  0.39
## F-statistic: 11.9 on 1 and 16 DF,  p-value: 0.00333
```

Notation

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

```
# Both variants yield the same result
lm(d$Goals ~ d$PlayerValue)
lm(Goals ~ PlayerValue, data = d)
```

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

```
colnames(d)

## [1] "Club"        "PlayerValue" "Goals"       "Points"
```

```
# Both variants yield the same result
lm(Goals ~ Club + PlayerValue + Points, data = d)
lm(Goals ~ ., data = d)
```

- Multivariate regressions feature more than one independent variable

R^2: Coefficient of Determination

```
## Multiple R-squared:  0.426, Adjusted R-squared:  0.39
```

Coefficient of determination, R^2, measures ratio of explained variance

Calculation

$$R^2 = \frac{SS_{\text{reg}}}{S_{\text{tot}}} \in [0,1] \text{ in OLS}$$

- Total sum of squares (proportional to sample variance)

$$SS_{\text{tot}} = \sum_i (y_i - \mu_y)^2$$

- Regression sum of squares

$$SS_{\text{reg}} = \sum_i (\hat{y}_i - \mu_y)^2$$

where $\hat{y}_i$ is the predicted value

Multivariate Regression

- Adjusted $\hat{R}^2$ is an attempt to take into account the phenomenon that R^2 automatically increases with extra explanatory variables
- Adjusted

$$\hat{R}^2 = 1 - \left(1 - R^2\right) \frac{n-1}{n-p-1} \in [0,1]$$

where p is the total number of regressors

Linear Regression Models

```
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       15.912      2.586    6.15  1.4e-05 ***
## d$PlayerValue      2.323      0.674    3.44   0.0033 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- Estimate gives the least squares estimates of α and coefficients
- Std. Error shows standard errors $\hat{\sigma}_i$ of each coefficient estimate
- *t*-value and *P*-value columns test whether any of the coefficients might be equal to zero
  - *t*-statistic is calculated as $t = \beta_i / \hat{\sigma}_i$, if errors $\boldsymbol{\varepsilon}$ follow a normal distribution
    $\rightarrow$ large values of t indicate that the null hypothesis can be rejected and that the corresponding coefficient is not zero
  - *P*-value expresses the results of the hypothesis test as a significance level; conventionally, *P*-values smaller than 0.05 are taken as evidence that the coefficient is non-zero

F-Test

- F-statistic tries to test the hypothesis that all coefficients (except the intercept) are equal to zero
- H_0: $\beta_1 = \beta_2 = \ldots = \beta_k = 0$

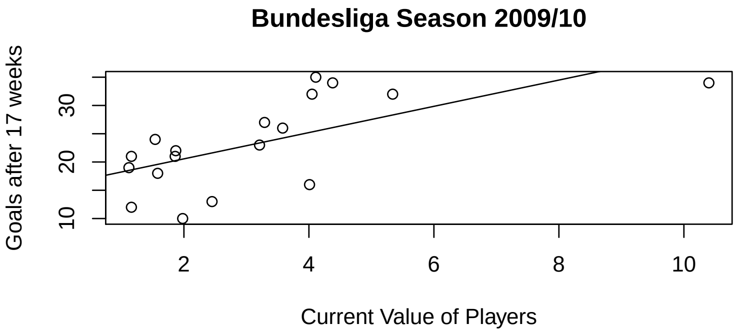
```
## F-statistic: 11.9 on 1 and 16 DF,  p-value: 0.00333
```

$\rightarrow$ With a P-value of 0.00333, we can reject the null hypothesis at the 1 %-significance level

Plot: Fitted Model

Draw line of best fit in 2 dimensions via `abline(model)`

```
plot(d$PlayerValue, d$Goals, main="Bundesliga Season 2009/10",
xlab="Current Value of Players", ylab="Goals after 17 weeks")
m <- lm(d$Goals ~ d$PlayerValue)
abline(m)
```



Outline

OLS Estimator: Assumptions

The OLS technique imposes several assumptions in order for the method to give meaningful results

1. Homoscedasticity means that the error term has the same variance σ^2 in each observation
2. Non-Autocorrelation requires that the errors are uncorrelated between observations
3. No Linear Dependence prerequisites regressors to all be linearly independent

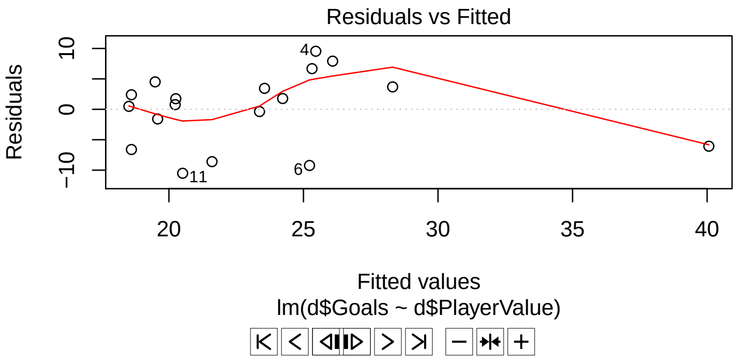
Required package `lmtest` for the following R scripts:

```
library(lmtest)   # load necessary library
```

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
```



Outline

3. Assumptions of OLS Estimator
   - Homoscedacitity
   - Non-Autocorrelation
   - No Linear Dependence

Assumption: Homoscedacitity

- Error term has the same variance σ^2 in each observation, i. e.

$$\mathrm{E}\left[\varepsilon_i^2 \,|\, X\right] = \sigma^2$$

- Violation is named heteroscedasticity
- Verify, for example, by:

Statistical Tests

- Breusch-Pagan test
- White test
- Goldfeld-Quandt test
- Harrison-McCabe test

Visual Regression Diagnostics

- Residuals vs fitted
- Residuals across observations
- Histogram or Q-Q plot to check normal distribution of residuals

Breusch-Pagan Test: Concept

Is the estimated variance of the residuals dependent on the regressors?
$\rightarrow$ suppose regression model $\boldsymbol{y} = \alpha + \beta_1 \boldsymbol{x}_1 + \ldots + \boldsymbol{x}_k + \boldsymbol{\varepsilon}$

1. Get estimated errors $\hat{\boldsymbol{\varepsilon}}$
2. Estimate of error variance can be obtained from the average of the squared values, i. e. $\hat{\boldsymbol{\varepsilon}}^2$
3. Assumption: variance of residuals $\hat{\boldsymbol{\varepsilon}}$ does not depend on the regressors $\boldsymbol{x}_1, \ldots \boldsymbol{x}_k$
4. Estimate model $\hat{\boldsymbol{\varepsilon}}^2 = \gamma_0 + \gamma_1 \boldsymbol{x}_1 + \ldots \gamma_k \boldsymbol{x}_k + \nu$
5. If an *F*-test confirms that the independent variables are jointly significant $\rightarrow$ the null hypothesis of homoscedasticity can be rejected

Breusch-Pagan Test

- Generate simple linear model $\boldsymbol{y} = \alpha + \beta \boldsymbol{x} + \boldsymbol{\varepsilon}$ as demonstration
- Generate a regressor $\boldsymbol{x}$

```
x <- rep(c(-1, 1), 50)
```

- Generate heteroscedastic and homoscedastic disturbances

```
err.heteroscedastic <- rnorm(100, sd = rep(c(1, 2), 50))
err.heteroscedastic[1:5]

## [1]  1.2630 -0.6525  1.3298  2.5449  0.4146

err.homoscedastic <- rnorm(100)
err.homoscedastic[1:5]

## [1]  0.78186 -0.77678 -0.61599  0.04658 -1.13039
```

- Create dependent variable $\boldsymbol{y}$ as a linear relationship

```
y.heteroscedastic <- 1 + x + err.heteroscedastic
y.homoscedastic <- 1 + x + err.homoscedastic
```

Breusch-Pagan Test

- Perform Breusch-Pagan test via `bptest(y ~ x1 + x2 + ...)`
- Example with heteroscedasticity $\rightarrow$ P-value ≤ 0.05

```
bptest(y.heteroscedastic ~ x)

##
##  studentized Breusch-Pagan test
##
## data:  y.heteroscedastic ~ x
## BP = 8.592, df = 1, p-value = 0.003376
```

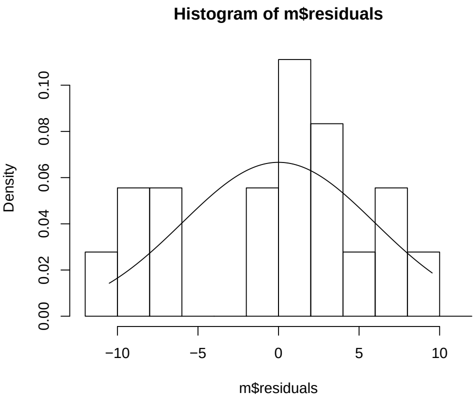
- Example with homoscedasticity $\rightarrow$ P-value > 0.05

```
bptest(y.homoscedastic ~ x)

##
##  studentized Breusch-Pagan test
##
## data:  y.homoscedastic ~ x
## BP = 0.3042, df = 1, p-value = 0.5812
```

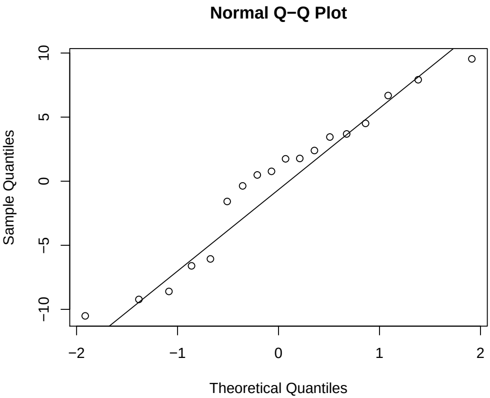
Normally Distributed Residuals

```
hist(m$residuals, freq = FALSE, breaks = seq(-12, 12, 2))
xx <- seq(min(m$residuals), max(m$residuals), 0.01)
lines(xx, dnorm(xx, mean = mean(m$residuals), sd = sd(m$residuals)))
```



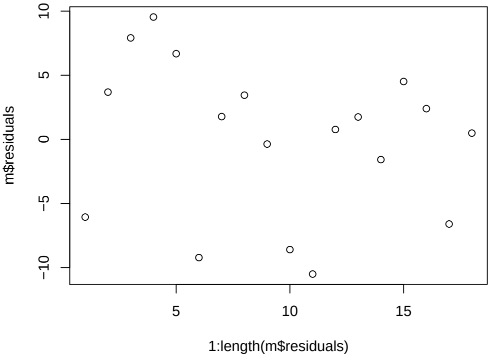
Normally Distributed Residuals

```r
qqnorm(m$residuals)  # plot sample against theoretical normal distribution
qqline(m$residuals)  # line that represents true normal distribution
```



Residuals across Observations

```
plot(1:length(m$residuals), m$residuals)
```



→ works better with residuals across time

Outline

Assumption: Non-Autocorrelation

- Errors are uncorrelated between observations, i. e.

$$\mathrm{E}[\varepsilon_i \varepsilon_j \,|\, X] = 0 \text{ for } i \neq j$$

- May be violated, e. g., in the context of time series data, panel data, cluster samples, hierarchical data
- Example: if you witnessed a stock making gains over the past, you might reasonably expect further upward movement
- Verify, for example, by
  - plotting residuals across observations
  - plotting or calculating the autocorrelation function (ACF) of the residuals
  - performing Durbin-Watson test

Autocorrelation Function

- Measures relationship between values separated from each other by a given time lag
- Given time series data $Y_1, \ldots, Y_N$ as observations, with mean $\bar{Y}$
- Autocorrelation coefficient r_h at lag h is given by

  $$r_h = \mathrm{Cor}(Y_{t+h}, Y_t) = \frac{c_h}{c_0}$$

  normalized by $c_0 = \sigma^2$ (variance of Y_t)
- Autocovariance function given by

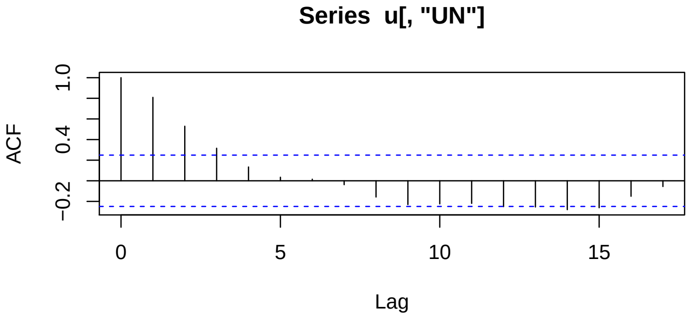
  $$c_h = \mathrm{Cov}(Y_{t+h}, Y_t) = \frac{1}{N} \sum_{t=1}^{N-h} (Y_t - \bar{Y})(Y_{t+h} - \bar{Y})$$

- Check if r_h exceeds a given significance level

Correlogram

- Plot autocorrelation function via `acf(d)`

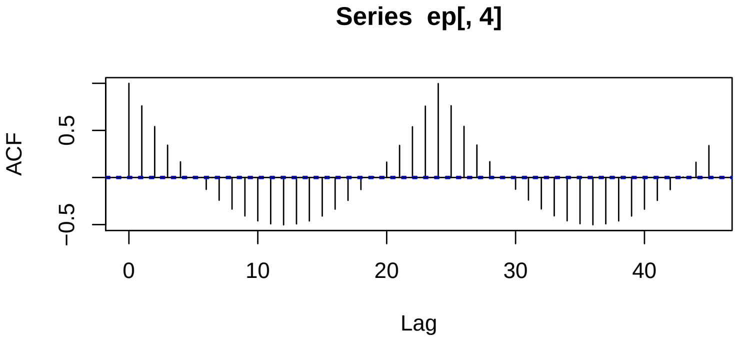
```
data(unemployment)
u <- window(unemployment, start = 1895, end = 1956)
acf(u[, "UN"])
```



→ If exceeds blue dashed line, then autocorrelation at a significant level

Correlogram

```
# Hourly electricity prices from day-ahead auctions
ep <- read.csv("epexspot_auction_de_2009-2012.csv",
               sep=",", header=FALSE)
acf(ep[,4])
```



Durbin-Watson Test

- Detects the presence of autocorrelation in the residuals
- Test statistic

$$d = \frac{\sum_{t=2}^{N} (\varepsilon_t - \varepsilon_{t-1})^2}{\sum_{t=1}^{N} \varepsilon_t^2} \approx 2(1 - r_1)$$

where r_1 is the sample autocorrelation of the residuals

Test statistic	Autocorrelation	Interpretation
$d = 2$	$r_1 = 0$	no autocorrelation
$d = 0$	$r_1 = +1$	perfect positive autocorrelation
$d = 4$	$r_1 = -1$	perfect negative autocorrelation

- H_0: no autocorrelation ($r_1 = 0$) present if $d = 2$
- H_A: autocorrelation ($r_1 \neq 0$) present if $d = 0$ or $d = 4$

Durbin-Watson Test

- Generate simple linear model $\boldsymbol{y} = \alpha + \beta \boldsymbol{x} + \boldsymbol{\varepsilon}$ as demonstration
- Generate a regressor $\boldsymbol{x}$

```
x <- rep(c(-1, 1), 50)
```

- Generate disturbances without/with autocorrelation

```
err.noac <- rnorm(100)
## generate two AR(1) error terms with parameter
## rho = 0 (white noise) and rho = 0.9 respectively
err.ac <- filter(err.noac, 0.9, method="recursive")
```

- Create dependent variable $\boldsymbol{y}$ as a linear relationship

```
y.noac <- 1 + x + err.noac
y.ac <- 1 + x + err.ac
```

Durbin-Watson Test

- Perform Durbin-Watson test via `dwtest(y ~ x1 + x2 + ...)`
- Example with no autocorrelation → P-value > 0.05

```
dwtest(y.noac ~ x)

##
##  Durbin-Watson test
##
## data:  y.noac ~ x
## DW = 1.678, p-value = 0.06347
## alternative hypothesis: true autocorrelation is greater than 0
```

- Example with autocorrelation → P-value ≤ 0.05

```
dwtest(y.ac ~ x)

##
##  Durbin-Watson test
##
## data:  y.ac ~ x
## DW = 0.3253, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

Outline

Assumption: No Linear Dependence

- Regressors $X = [\boldsymbol{x}_1 \mid \ldots \mid \boldsymbol{x}_k]$ are all linearly independent, i. e.

  $$\Pr[\operatorname{rank}(X) = k] = 1,$$

  that means X must almost surely have full column rank
- Violation called linear dependence or (perfect) multicollinearity
- Testing by Pearson correlation coefficient possible, but quite strict
- Instead: use Variance Inflation Factors or condition number of X

Correlation Matrix

- Construction of a correlation matrix for the explanatory variables will yield indications as to the likelihood that any given couplet of right-hand-side variables are creating multicollinearity problems
- Correlation values (off-diagonal elements) of at least 0.4 are interpreted as indicating a multicollinearity problem
- Example:

```
cor(as.data.frame(cbind(d$PlayerValue, d$Points)),
    use="pair")
```

```
##        V1    V2
## V1 1.000 0.544
## V2 0.544 1.000
```

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$Goals ~ d$PlayerValue + d$Points)
vif(m)

## d$PlayerValue      d$Points
##          1.42          1.42

vif(m) > 4  # problem?

## d$PlayerValue      d$Points
##         FALSE         FALSE
```

- Indication of multicollinearity if above 4

Condition Number

- Condition number κ measures the ill-conditioning of a matrix
- Equivalent to the numerical stability of its inversion (in finite precision) or how full its rank is
- Condition number κ is computed via `kappa(d)`

```
kappa(as.data.frame(cbind(d$PlayerValue, d$Points)))

## [1] 15.77
```

- If the condition number is above 30 , the regression is said to have multicollinearity

Outline

Model Selection

Motivation

- Example: Which model should we select?
  1. Model A consisting of 10 explanatory variables with an $R^2 = 0.6$
  2. Model B consisting of 6 explanatory variables with an $R^2 = 0.4$

Information Criterion

- Deals with trade-off between complexity and the goodness of fit
- Cannot tell anything about how well a model fits the data in an absolute sense
- Prefer model with the minimum information criterion value
- Examples: Akaike Information Criterion, Bayes Information Criterion

Information Criterion: AIC and BIC

- Not only rewards goodness of fit, but also includes a penalty that is an increasing function of the number of estimated parameters
- The penalty discourages overfitting

Akaike Information Criterion

- $AIC = 2\,\mathrm{df} - 2\ln L$
- df is the degrees of freedom (number of parameters including error $\boldsymbol{\varepsilon}$)
- L is the maximized value of the likelihood function

Bayesian Information Criterion

- $BIC = \mathrm{df} \cdot \ln n - 2\ln L$
- Penalty is logarithmic with observations n
- BIC puts stronger penalty on additional parameters than AIC

AIC and BIC

- `logLik(m)` extracts likelihood

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

## [1] -57.24
```

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

```
AIC(m)

## [1] 120.5

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

## [1] 120.5
```

```
BIC(m)

## [1] 123.2

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

## [1] 123.2
```

Outline

Prediction with Linear Models

BI Case Study

Until September 2009, the Freiburg soccer team has scored 19 goals with a market value of €1.12 m

Question: How many goals could be expected with a market value of €5 m?

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(Goals ~ PlayerValue, data = d)
nd <- data.frame(PlayerValue = 5)
predict(m, newdata = nd)

##     1
## 27.52
```

$\rightarrow$ the expected number of goals is 27.52

Outline

Wrap-Up: OLS Estimator

- The OLS technique imposes several assumptions in order for the method to give meaningful results
  1. Homoscedasticity means that the error term has the same variance σ^2 in each observation
  2. Non-Autocorrelation requires that the errors are uncorrelated between observations
  3. No Linear Dependence prerequisites regressors to all be linearly independent
- After verifying assumption, identify parameters with significant influence on outcome $\rightarrow$ t-value and P-value
- Look at overall model fit in terms of R^2, adjusted R^2 and F-test
- Select model that competes best in terms of information criterion
- Interpret magnitude and sign of coefficients, as well as significance level

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

Summary: Commands

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

Further Exercises

→ Available online as homework