Script_markdown2

Multiple Linear Regression

Linear regression is one of the most widely used techniques in statistical and econometric applications.

The Linear Model

The Multiple Regression Model:

- Given is one response variable up to some random errors.
- Is a linear function of several predictors (or covariates)
- The linear function involves unknown parameters. The goal is to estimate these parameters, to study their relevance and to estimate the error variance.

Notation:

- y_i dependent variable.
- x_{ik} kth independent variable (or regressor) with $k = 1, \ldots, K$.
  Can be stochastic or deterministic.
- β_k vector of unkown parameters
- ε_i stochastic, unknown error term
- i indexes the ith individual with $i = 1, \ldots, n$, where n is the sample size

Assumption 1.1: Linearity

$$y_i = \sum_{k=1}^{K} \beta_k x_{ik} + \varepsilon_i, \quad i = 1, \ldots, n. \tag{1}$$

Usually, a constant (or intercept) is included, in this case $x_{i1} = 1$ for all i. In the following we will always assume that a constant is included in the linear model, unless otherwise stated. A special case of the above defined linear model is the so-called *simple linear model*, defined as

$$y_i = \beta_1 + \beta_2 x_i + \varepsilon_i, \quad i = 1, \ldots, n. \tag{2}$$

Often it is convenient to write Eq.~(1) using matrix notation

$$y_i = \mathbf{x}_i' \boldsymbol{\beta} + \varepsilon_i, \quad i = 1, \ldots, n,$$

where $\mathbf{x}_i = (x_{i1}, \ldots, x_{iK})'$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_K)'$. Stacking all individual rows i leads to

$$\underset{(n\times 1)}{\mathbf{y}} = \underset{(n\times K)}{\mathbf{X}} \underset{(K\times 1)}{\boldsymbol{\beta}} + \underset{(n\times 1)}{\boldsymbol{\varepsilon}}, \tag{3}$$

where

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ mathbfy_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} x_{11} & \ldots & x_{1K} \\ \vdots & \ddots & \vdots \\ mathbfx_{n1} & \ldots & x_{nK} \end{pmatrix}, \quad \text{and} \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

Stochastic Models

The linear model in equation 1 involves some stochastic (random) components: the error terms ε_i are random variables and hence the response variables y_i are as well. The predictor variables/covariates x_{ik} are assumed to be deterministic here, but they could also be stochastic. Since we will not concern ourselves with asymptotic analysis, this is not going to matter a lot.

The stochastic nature of the error terms ε_i can be assigned to various sources: for example, measurement errors or inability to capture all underlying non-systematic effects which are then summarized by a random variable with expectation zero. The stochastic modelling approach will allow to quantify uncertainty, to assign significance to various components, e.g. significance of predictor variables in model 1 , and to find a good compromise between the size of a model and the ability to describe the data. The observed response in the data is always assumed to be realizations of the random variables $y_i, ..., y_n$; the x_{ik}'s are non-random and equal to the observed predictors in the data.

The quadratic regression model with $k = 3$:

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ mathbfy_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & x_1 & x_1^2 \\ \vdots & \vdots & \\ 1 & x_n & x_n^2 \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix}, \quad \text{and} \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

Notice how the function is *quadratic* in the covariates x_{ik}, but *linear* in the coefficients β_k and therefore a special case of the linear model in 1.

Regression with transformed predictor variables

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ mathbfy_n \end{pmatrix}, \quad \mathbf{X} = \begin{pmatrix} 1 & log(x_{12}) & sin(\pi x_{13}) \\ \vdots & \vdots & \\ 1 & log(x_{n2}) & sin(\pi x_{n3}) \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix}, \quad \text{and} \quad \boldsymbol{\varepsilon} = \begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}.$$

Again, the model is *linear* in the coefficients in the coefficients β_k but nonlinear in the x_{ik}'s.

In Summary:

{The model in 1} is called linear in the coefficients β_k. The covariates and the outcome variable can be transformed versions of the original covariates.

Goals of the linear regression analysis

- **A good "fit"**: Fitting or estimating a (hyper) plane over the covariates to explain outcome variables such that the errors are "small". The standard too for this is the method of *leastsquares*.
- **Good parameter estimates:** This is useful to describe the change of the outcome variable when varying some covariates.
- **Good prediction:** This is useful to predict a new response as a function of new covariate (-values).
- **Uncertainties and significance for the three goals above:** Confidence intervals and statistical test are useful tools for this goal.
- **Development of a good model:** In an interactive process, using methods for the goals mentioned above, we may change parts of the initial model to come up with a better model. Whether we get to model selection will depend on time.

The first and third goal can be opposing to each other, which we will discuss in more detail in the section on nonparametric density estimation.

Least Squares Regression

We assume the linear model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ and are looking for a "good" estimate of $\boldsymbol{\beta}$.

The OLS estimator $\hat{\boldsymbol{\beta}}$ is defined as the minimizer of a specific loss function termed *the sum of squared residuals*

$$SSR(\hat{\boldsymbol{\beta}}^*) \;=\; \sum_{i=1}^{n}(y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}^*)^2 \;=\; (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}^*)'(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}^*).$$

I.e., we have

$$\hat{\boldsymbol{\beta}} \;:=\; \arg\min_{\hat{\boldsymbol{\beta}}^* \in \mathbb{R}^K} S(\hat{\boldsymbol{\beta}}^*),$$

We can easily minimize $SSR(\hat{\boldsymbol{\beta}}^*)$ in closed form:

$$\begin{aligned} SSR(\hat{\boldsymbol{\beta}}^*) &= (\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}^*)'(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}^*) \\ &= \mathbf{y}'\mathbf{y} - (\mathbf{X}\hat{\boldsymbol{\beta}}^*)'\mathbf{y} - \mathbf{y}'\mathbf{X}\hat{\boldsymbol{\beta}}^* + \hat{\boldsymbol{\beta}}^{*'}\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}}^* \\ &= \mathbf{y}'\mathbf{y} - 2\mathbf{y}'\mathbf{X}\hat{\boldsymbol{\beta}}^* + \hat{\boldsymbol{\beta}}^{*'}\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}}^* \\ \Rightarrow \quad \frac{d}{d\hat{\boldsymbol{\beta}}^*}SSR(\hat{\boldsymbol{\beta}}^*) &= -2\mathbf{X}'\mathbf{y} + 2\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}}^* \end{aligned}$$

Setting the first derivative so zero yields the so-called *normal equations*

$$\mathbf{X}'\mathbf{X}\hat{\boldsymbol{\beta}} \;=\; \mathbf{X}'\mathbf{y},$$

which lead to the OLS estimator

$$\hat{\boldsymbol{\beta}} \;=\; (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}, \tag{4}$$

where $(\mathbf{X}'\mathbf{X})^{-1}$ exists because of our full rank assumption (Assumption 3).

The vector of residuals $\hat{\boldsymbol{\varepsilon}}$ has only $n-K$ so-called *degrees of freedom*. The vector looses K degrees of freedom, since it has to satisfy the K linear restrictions ($\mathbf{X}'\hat{\boldsymbol{\varepsilon}} = \mathbf{0}$). Particularly, in the case with intercept we have that $\sum_{i=1}^{n}\hat{\boldsymbol{\varepsilon}}_i = \mathbf{0}$.\

This loss of K degrees of freedom also appears in the definition of the *unbiased* variance estimator

$$s^2 \;=\; \frac{1}{n-K}\sum_{i=1}^{n}\hat{\varepsilon}_i^2 = \frac{\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}}}{n-k}. \tag{5}$$

Assumption 1.2: Strict Exogeneity

$$\mathbb{E}(\varepsilon_i|\mathbf{X}) = 0$$

or equivalently stated for the vector $\boldsymbol{\varepsilon}$

$$\mathbb{E}(\boldsymbol{\beta}|\mathbf{X}) = \mathbf{0}.$$

`Notice that in the presence of a constant regressor, setting the expectation to zero is a normalization`

Some Implications of Strict Exogeneity:

- The unconditional mean of the error term is zero:

$$\mathbb{E}(\varepsilon_i) \;=\; 0 \quad (i = 1, \ldots, n) \tag{6}$$

Generally, two random variables x and y are said to be **orthogonal** if their cross moment is zero: $\mathbb{E}(xy) = 0$. Under strict exogeneity, the regressors are orthogonal to the error term for *all* observations, i.e.,

$$\mathbb{E}(x_{jk}\varepsilon_i) \quad = \quad 0 \quad (i, j = 1, \ldots, n; k = 1, \ldots, K) \tag{7}$$

$$\begin{aligned} \mathbf{Cov}(\varepsilon_i, x_{jk}) &= \mathbb{E}(x_{jk}\varepsilon_i) - \mathbb{E}(x_{jk})\,\mathbb{E}(\varepsilon_i) \quad \text{(By Def. of Cov)} \\ &= \mathbb{E}(x_{jk}\varepsilon_i) \quad \text{(Since } \mathbb{E}(\varepsilon_i) = 0\text{; see Eq. (6))} \\ &= 0 \quad \text{(By the orthogonality result; see Eq. (7))} \quad \square \end{aligned}$$

Assumption 1.3: Rank Condition (no multicollinearity)

$$rank(\mathbf{X}) = K \quad \text{a.s.}$$

This assumption demands that the event of one regressor being linearly dependent on the others occurs with a probability equal to zero. (This is the literal translation of the "almost surely (a.s.)" concept.) It implies that $n \geq K$. \

This assumption is a bit dicey and its violation belongs to one of the classic problems in applied econometrics (keywords: multicollinearity, dummy variable trap, variance inflation). The violation of this assumption harms any economic interpretation as we cannot disentangle the regressors' individual effects on $\mathbf{y}$. Therefore, we will later think of this assumption as an *identification* assumption.\

Assumption 1.4: Homoskedasticity

$$\begin{aligned} \mathbb{E}(\varepsilon_i^2|\mathbf{X}) &= \sigma^2 > 0 \\ \mathbb{E}(\varepsilon_i\varepsilon_j|\mathbf{X}) &= 0, \qquad i \neq j. \end{aligned}$$

Or more compactly written as,

$$\mathbb{E}(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'|\mathbf{X}) \quad = \quad \sigma^2 I_n, \qquad \sigma^2 > 0.$$

Thus, we assume that, for a given realization of $\mathbf{X}$, the error process is uncorrelated ($\mathbb{E}(\varepsilon_i\varepsilon_j|\mathbf{X}) = 0$, for all $i \neq j$) and homoscedastic (same σ^2, for all i).

Obviously, the strict exogeneity assumption implies that this assumption can be stated more conveniently as

- $Var(\varepsilon_i|\mathbf{X}) = \mathbb{E}(\varepsilon_i^2|\mathbf{X}) - \mathbb{E}(\varepsilon_i|\mathbf{X})^2 = \mathbb{E}(\varepsilon_i^2|\mathbf{X}) = \sigma^2$
- $\mathbf{Cov}(\varepsilon_i, \varepsilon_j|\mathbf{X}) = \mathbb{E}(\varepsilon_i\,\varepsilon_j|\mathbf{X}) - \mathbb{E}(\varepsilon_i|\mathbf{X})\,\mathbb{E}(\varepsilon_j|\mathbf{X}) = \mathbb{E}(\varepsilon_i\,\varepsilon_j|\mathbf{X}) = 0.$

Assumption 1.5: Normality

$$\boldsymbol{\varepsilon}|\mathbf{X} \sim N(\mathbf{0}, \sigma^2\mathbf{I}_n)$$

The assumption immediately implies that

$$(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})|\mathbf{X} \sim N(\mathbf{0}, \sigma^2(\mathbf{X}'\mathbf{X})^{-1}),$$

i.e. the variance of $\hat{\boldsymbol{\beta}}$, $Var(\hat{\boldsymbol{\beta}}) = \hat{\sigma}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}$.

Some quantities of interest:

- The *(OLS) fitted value*: $\hat{y}_i = \mathbf{x}_i\hat{\boldsymbol{\beta}}$
  In matrix notation: $\hat{\mathbf{y}} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}$
- The *(OLS) residual*: $\hat{\varepsilon}_i = y_i - \hat{y}_i$
  In matrix notation: $\hat{\boldsymbol{\varepsilon}} = \mathbf{y} - \hat{\mathbf{y}} = \left(\mathbf{I}_n - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\right)\mathbf{y}$,

```
\mathbf{y}'\mathbf{y}&=&\hat{\mathbf{y}}'\hat{\mathbf{y}}+\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymb
```

\end{equation}

This just means that the assumptions 1-4 from section and are satisfied. It can be shown that:

(i) $\mathbb{E}[\hat{\boldsymbol{\beta}}] = \boldsymbol{\beta}$

(ii) $\mathbb{E}[\hat{\mathbf{y}}] = \mathbf{X}\boldsymbol{\beta}$ which follows from (i)

(iii) $\widehat{Cov(\hat{\boldsymbol{\beta}})} = \sigma^2(\mathbf{X}'\mathbf{X})^{-1}$

(iv) $Cov(\hat{\mathbf{y}})$: Define a a so-called projection matrix: $\mathbf{P} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-\mathbf{1}}\mathbf{X}'$. Then $Cov(\hat{\mathbf{y}}) = \sigma^2\mathbf{P}, Cov(\mathbf{y} - \hat{\mathbf{y}}) = \sigma^2(\mathbf{I} - \mathbf{P})$.

(v) $s^2 = \hat{\sigma}^2 = \frac{1}{n-K}\sum_{i=1}^{n}\hat{\varepsilon}_i^2 = \frac{\hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}}}{n-k}$

Distribution of least squares estimates assuming Gaussian errors

We assume the linear model from above and we assume normality, i.e. $\varepsilon_1, ...\varepsilon_n \quad \text{i.i.d} \sim \mathcal{N}(0, \sigma^2)$.

(i) $\hat{\boldsymbol{\beta}} \sim \mathcal{N}_k(\boldsymbol{\beta}, \sigma^2(\mathbf{X}'\mathbf{X})^{-1})$

(ii) $\hat{\mathbf{y}} \sim \mathcal{N}_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{P})$, $\mathbf{y} - \hat{\mathbf{y}} \sim \mathcal{N}_n(\mathbf{0}, \sigma^2(\mathbf{I} - \mathbf{P}))$

(iii) $\hat{\sigma}^2 \sim \frac{\sigma^2}{n-k}\chi^2_{n-k}$

The normality assumptions of the errors ε_i is often not (approximately) fulfilled in practice. We can then rely on the central limit theorem which implies that for large sample size n, the properties (i)-(iii) above are still approximately true. This is the usual justification in practice to use these properties for constructing confidence intervals and tests for the linear model parameters. However, it is often much better to use robust methods in case of non-gaussian errors which we are not discussing here.

Coefficient of determination

The total sample variance of the dependent variable $\sum_{i=1}^{n}(y_i - \bar{y})^2$, where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$, can be decomposed as following:

Variance decomposition For the OLS regression of the linear model (1) with intercept it holds that

$$\underbrace{\sum_{i=1}^{n}(y_i-\bar{y})^2}_{\text{total variance}} = \underbrace{\sum_{i=1}^{n}(\hat{y}_i-\bar{y})^2}_{\text{explained variance}} + \underbrace{\sum_{i=1}^{n}(y_i-\hat{y}_i)^2}_{\text{unexplained variance}=\sum_{i=1}^{n}\hat{\varepsilon}_i^2} .$$

- As a consequence of Prop. **??** we have for regressions with intercept: $\sum_{i=1}^{n}\hat{\varepsilon}_i = 0$. Hence, from $y_i = \hat{y}_i + \hat{\varepsilon}_i$ it follows that

$$\begin{aligned} \frac{1}{n}\sum_{i=1}^{n} y_i &= \frac{1}{n}\sum_{i=1}^{n}\hat{y}_i + \frac{1}{n}\sum_{i=1}^{n}\hat{\varepsilon}_i \\ \bar{y} &= \bar{\hat{y}}_i + 0 \end{aligned}$$

- From Prop. **??** we know that:

$$\begin{aligned} \mathbf{y}'\mathbf{y} &= \hat{\mathbf{y}}'\hat{\mathbf{y}} + \hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}} \\ \mathbf{y}'\mathbf{y} - n\bar{y}^2 &= \hat{\mathbf{y}}'\hat{\mathbf{y}} - n\bar{y}^2 + \hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}} \\ \mathbf{y}'\mathbf{y} - n\bar{y}^2 &= \hat{\mathbf{y}}'\hat{\mathbf{y}} - n\bar{\hat{y}}^2 + \hat{\boldsymbol{\varepsilon}}'\hat{\boldsymbol{\varepsilon}} \quad \text{(By our result above.)} \\ \sum_{i=1}^{n} y_i^2 - n\bar{y}^2 &= \sum_{i=1}^{n}\hat{y}_i^2 - n\bar{\hat{y}}^2 + \sum_{i=1}^{n}\hat{\varepsilon}_i^2 \\ \sum_{i=1}^{n}(y_i-\bar{y})^2 &= \sum_{i=1}^{n}(\hat{y}_i-\bar{\hat{y}})^2 + \sum_{i=1}^{n}\hat{\varepsilon}_i^2 \quad \square \end{aligned}$$

  The larger the proportion of the explained variance, the better is the fit of the model. This motivates the definition of the so-called R^2 coefficient of determination:

$$R^2 = \frac{\sum_{i=1}^{n}\left(\hat{y}_i-\bar{\hat{y}}\right)^2}{\sum_{i=1}^{n}\left(y_i-\bar{y}\right)^2} = 1 - \frac{\sum_{i=1}^{n}\hat{u}_i^2}{\sum_{i=1}^{n}\left(y_i-\bar{y}\right)^2}$$

  Obviously, we have that $0 \leq R^2 \leq 1$. The closer R^2 lies to 1, the better is the fit of the model to the observed data. However, a high/low R^2 does not mean a validation/falsification of the estimated model. Any relation (i.e., model assumption) needs a plausible explanation from relevant economic theory.

The most often criticized disadvantage of the R^2 is that additional regressors (relevant or not) will always increase the R^2.

R^2 increase

Let R_1^2 and R_2^2 result from

$$\begin{aligned} \mathbf{y} &= \mathbf{X}_1\hat{\boldsymbol{\beta}}_{11} + \hat{\boldsymbol{\varepsilon}}_1 \quad \text{and} \\ \mathbf{y} &= \mathbf{X}_1\hat{\boldsymbol{\beta}}_{21} + \mathbf{X}_2\hat{\boldsymbol{\beta}}_{22} + \hat{\boldsymbol{\varepsilon}}_2. \end{aligned}$$

It then holds that $R_2^2 \geq R_1^2$.

Because of this, the R^2 cannot be used as a criterion for model selection. Possible solutions are given by

penalized criterions such as the so-called *adjusted* R^2 defined as

$$\begin{aligned}
\overline{R}^2 &= 1 - \frac{\frac{1}{n-K}\sum_{i=1}^{n}\hat{u}_i^2}{\frac{1}{n-1}\sum_{i=1}^{n}(y-\bar{y})_i^2} \\
&= 1 - \frac{n-1}{n-K}\left(1-R^2\right) \\
&= 1 - \frac{n-1}{n-K} + \frac{n-1}{n-K}R^2 \quad + \frac{K-1}{n-K}R^2 - \frac{K-1}{n-K}R^2 \\
&= 1 - \frac{n-1}{n-K} + R^2 \quad + \frac{K-1}{n-K}R^2 \\
&= -\frac{K-1}{n-K} + R^2 \quad + \frac{K-1}{n-K}R^2 \\
&= R^2 - \frac{K-1}{n-K}\left(1-R^2\right) \leq R^2
\end{aligned}$$

The adjustment is in terms of degrees of freedom.

```
######With all three regressors###
###Simulating OLS samples and plot the regression lines####
#set.seed(32323)
## Two explanatory variables plus an intercept:
set.seed(20)
N         <- 1000 # Number of observations
X.1       <- rep(1, N)
X.2       <- rnorm(N, mean=10, sd=1.5) # (pseudo) random numbers form a normal distribution
X.3       <- rnorm(N, mean=5, sd=1.5) # (pseudo) random numbers form a normal distribution
X         <- cbind(X.1, X.2,X.3)
###Homoscedastic error term
eps       <-rnorm(N, 0,2)#
beta.vec  <- c(5,-5,1)
## Model
y         <- X %*% beta.vec + eps
##Solving for beta hat###
#X         <- cbind(X.1, X.2,X.3)
beta.hat <- solve(t(X) %*% X) %*% t(X) %*% y
beta.hat
```

```
##          [,1]
## X.1  5.066746
## X.2 -5.017095
## X.3  1.029851
```

```
K<-dim(X)[2]
#####We can do the same thing with a pre-installed R package##
lm.result   <- lm(y~X)
lm.summary  <- summary(lm.result)
lm.summary
```

```
##
## Call:
## lm(formula = y ~ X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.7078 -1.3284 -0.0267  1.3057  7.1035
##
```

```
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.06675    0.45597   11.11   <2e-16 ***
## XX.1              NA         NA      NA       NA
## XX.2        -5.01709    0.04095 -122.51   <2e-16 ***
## XX.3         1.02985    0.04288   24.02   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.975 on 997 degrees of freedom
## Multiple R-squared:  0.939,  Adjusted R-squared:  0.9389
## F-statistic:  7673 on 2 and 997 DF,  p-value: < 2.2e-16
```

```
#####calculate the fitted values#####
y.hat<- X %*% beta.hat
eps.hat<-y-X %*% beta.hat
###calculate the covariance matrix#
se<-(t(eps.hat)%*%(eps.hat))/(N-K)
cov<-se[1]*solve(t(X) %*% X)
d1<-sqrt(diag(cov))
#Calculate the coefficient of determination#
r_squared<-sum((y.hat-mean(y.hat))^2)/sum((y-mean(y))^2)
##adjusted r squared
k<-dim(X)[2]
r_squared.adj<-r_squared-(k-1)/(N-k)*(1-r_squared)

r_squared
```

```
## [1] 0.9389932
```

```
r_squared.adj
```

```
## [1] 0.9388708
```

```
##Two explanatory variables (including the intercept) to simulate the model,
#Three covariates included in the estimation
set.seed(20)
N         <- 1000 # Number of observations
X.1       <- rep(1, N)
X.2       <- rnorm(N, mean=10, sd=1.5) # (pseudo) random numbers form a normal distribution
X.3       <- rnorm(N, mean=5, sd=1.5) # (pseudo) random numbers form a normal distribution
X         <- cbind(X.1, X.2)###Covariates that generate the y data
###Homoscedastic error term
eps       <-rnorm(N, 0,4)#
beta.vec  <- c(5,-5)
## Model
y         <- X %*% beta.vec + eps
##Solving for beta hat with three covariates###
X         <- cbind(X.1, X.2,X.3)###The matrix of covariates we use for estimation
beta.hat <- solve(t(X) %*% X) %*% t(X) %*% y
beta.hat
```

```
##            [,1]
## X.1  5.13349142
## X.2 -5.03418939
## X.3  0.05970101
```

```
#####calculate the fitted values#####
y.hat<- X %*% beta.hat
eps.hat<-y-X %*% beta.hat
###calculate the covariance matrix#
se<-(t(eps.hat)%*%(eps.hat))/(N-2)
cov<-se[1]*solve(t(X) %*% X)
d1<-sqrt(diag(cov))
#Calculate the coefficient of determination#
r_squared<-sum((y.hat-mean(y.hat))^2)/sum((y-mean(y))^2)
##adjusted r squared
k<-dim(X)[2]###k is the number of collumns of the covariate matrix used for estimation.
r_squared.adj<-r_squared-(k-1)/(N-k)*(1-r_squared)

r_squared
```

```
## [1] 0.7913872
```

```
r_squared.adj
```

```
## [1] 0.7909687
```

######With all three###

###Simulating OLS samples and plot the regression lines#### #set.seed(32323) ## Two explanatory variables plus an intercept: set.seed(20) N <- 1000 # Number of observations X.1 <- rep(1, N) X.2 <- rnorm(N, mean=10, sd=1.5) # (pseudo) random numbers form a normal distribution X.3 <- rnorm(N, mean=5, sd=1.5) # (pseudo) random numbers form a normal distribution X <- cbind(X.1, X.2,X.3) ###Homoscedastic error term eps <-rnorm(N, 0,2)# beta.vec <- c(5,-5,1)

Model

y <- X %% *beta.vec + eps ##Solving for beta hat### #X <- cbind(X.1, X.2,X.3) beta.hat <- solve(t(X)* %% X) %% *t(X)* %% y beta.hat ### #####We can do the same thing with a pre-installed R package## lm.result <- lm(y~X) lm.summary <- summary(lm.result) lm.summary #####calculate the fitted values##### y.hat<- X %% *beta.hat eps.hat<-y-X* %% beta.hat k<-dim(X)[2] ###calculate the covariance matrix# se<-(t(eps.hat)%%*(eps.hat))/(N-k) cov<-se[1]*solve(t(X) %% *X) d1<-sqrt(diag(cov)) #Calculate the coefficient of determination# r_squared<-sum((y.hat-mean(y.hat))^2)/sum((y-mean(y))^2) ##adjusted r squared r_squared.adj<-r_squared-(k-1)/(N-k)*(1-r_squared)

r_squared r_squared.adj

```
\subsubsection{Multiple regressions for a single outcome}
It can be tempting to replace multiple regression with many single regressions with a single covariate.

\bigskip

The following example illustrates why:
\bigskip

Consider two covariates $\mathbf{x_1}$ and $\mathbf{x_2}$ with the following values:


\bigskip
\begin{tabular}{ l| c ccccccc }
 $ x_{i1}$&0&1 & 2 & 3&0&1 & 2 & 3 \\
 $x_{i2}$&-1 &0&1 & 2 &1 & 2 & 3&4\\
 $y_i$&1 & 2 & 3&4 &-1 &0&1 & 2\\
```

```
\end{tabular}

\bigskip
Multiple regression yields the least squares solution which describes the data points exactly

\begin{equation}
y_i=\hat{y}_i=2x_{i1}-x_{i2} \quad \mbox{for all} \, i \quad (\hat{\sigma}^2=0).
\end{equation}

The coefficients 2 and -1 respectively, describe how ${y}$ is changing when varying either $ x_{1}$ or $
\smallskip

On the other hand, if we do a simple regression of $y$ onto $x_{2}$  (while ignoring $x_{1}$), we obtai
\begin{equation}
y_i=\hat{y}_i=\frac{1}{9}x_{i2}-\frac{4}{3} \quad \mbox{for all} \, i \quad (\hat{\sigma}^2=1.72).
\end{equation}

The least squares regression line describes how $y$ changes when varying $x_{2}$ while ignoring $x_1$. I
\smallskip

The reason for this is that $x_{1}$ and $x_{2}$ are highly correlated: if $x_{2}$ increases then also $x

\smallskip

In the case of orthogonal covariates, this is not a problem as can easily be seen by the following calcu
\begin{equation}
\hat{\beta}_j=\sum_{i=1}^nx_{ij}y_i/\sum_{i=1}^nx^2_{ij} \quad(j=1,...,k),
\end{equation}
i.e. $\hat{\beta}_j$ depends only on the response variable $y_i$ and the $j$'th covariate $x_{ij}$.

This is not in opposition to the result that in general the OLS estimator is consistent even when releva



\subsection{Tests and Confidence Regions}


\subsubsection{Hypothesis testing}

The idea behind testing is to somehow quantify the likelihood of observing a certain outcome (data) unde

We make assumptions 1.1-1.5 (i.e. including normality of the errors). As stated above, this implies that
\smallskip

If we are interested in testing whether the $j$'th covariate is statistically significant (relevant),  w
\begin{equation}
\frac{b_j}{\sqrt{\sigma^2(\mathbf{X'X})^{-1}_{jj}}}\sim \mathcal{N} (0,1)
\end{equation}

Since $\sigma^2$ is unkown, this quantity is not useful, but if we substitute it with the estimate $\hat

\begin{equation}
T_j=\frac{b_j}{\sqrt{\hat{\sigma}^2(\mathbf{X'X})^{-1}_{jj}}}\sim t_{n-k} \quad \mbox{under the null-hyp
\end{equation}
```

which has a slightly different distribution than standard normal $\mathcal{N}(0,1)$. The corresponding

```r
###Simulating OLS samples and plot the regression lines####
#set.seed(32323)
## Two explanatory variables plus an intercept:
#set.seed(20)
N         <- 1000 # Number of observations
X.1       <- rep(1, N)
X.2       <- rnorm(N, mean=10, sd=1.5) # (pseudo) random numbers form a normal distribution # (pseudo)
X         <- cbind(X.1, X.2)
###Homoscedastic error term
eps       <-rnorm(N, 0,10)#
beta.vec  <- c(5,-5)
## Model
y         <- X %*% beta.vec + eps
##Solving for beta hat###
#X         <- cbind(X.1, X.2,X.3)
beta.hat <- solve(t(X) %*% X) %*% t(X) %*% y
beta.hat
```

```
##          [,1]
## X.1  8.081389
## X.2 -5.315418
```

```r
####Now, let's calculate the  value for the t-test####
#give out the dimensions of the X vector#
xx<-dim(X)
length.x<-xx[2]
###calculate the predicted values from the model
y.hat<- X %*% beta.hat
#calculate the unexplained variance
eps.hat<-y-X %*% beta.hat
#calculate the estimated standard errors
se<-(t(eps.hat)%*%(eps.hat))/(N-length.x)
cov<-se[1]*solve(t(X) %*% X)
d1<-sqrt(diag(cov))
#calculate the value of the t statistic
t<-beta.hat/d1
###Find the critical values from the t-distribution###
t.crit_0.05<-abs(qt(0.05/2, N-length.x))
t.crit_0.1<-abs(qt(0.1/2, N-length.x))
##Significant or not###
sig_0.05<-abs(t)>t.crit_0.05
sig_0.1<-abs(t)>t.crit_0.1
####
sig_0.05
```

```
##     [,1]
## X.1 TRUE
## X.2 TRUE
```

```
sig_0.1
```

```
##     [,1]
## X.1 TRUE
## X.2 TRUE
```

When there is a correlation between covariates, it is possible that there is no covariate for which the t-test rejects the Null, even though there really is a significant effect. The reason is that multicollinearity will result in the variables mutually increasing each other's standard error, thus giving rise to the insignificance with the t-test. This can be checked by testing whether all covariates are jointly significant with $H_0 : \beta_2 = ... = \beta_k = 0$ with $H_1 : \beta_j \neq 0$ for at least one $j \in 2, ..., k$; we assume here that the first covariate is the constant. Such a test can be developed with an analysis of variance (anova), whereby we decompose the the total squared error around the mean $\bar{\mathbf{Y}} = n^{-1} \sum_{i=1}^{n} y_i$: recall the variance decomposition

$$\underbrace{\sum_{i=1}^{n} (y_i - \bar{y})^2}_{\text{total variance}} = \underbrace{\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2}_{\text{explained variance}} + \underbrace{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}_{\text{unexplained variance}}$$

i.e. it is decomposed as a sum of the squared error due to the regression (the amount that the fitted values vary around the global arithmetic mean) and the squared residual error.

We can summarize such a decomposition by an ANOVA table:

	sum of squares	degrees of freedom	mean square
regression	$\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$	K-1	$\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 / (K-1)$
error	$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2$	n-K	$\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 / (n-K)$
total around global mean	$\sum_{i=1}^{n} (y_i - \bar{y})^2$	n-1	-

In the case of a global null-hypothesis, there is no effect of any covariate, therefore the ratio of the explained variance and the total variance should be close to one. The denominator is an average of the sample variances for each group, which is an estimate of the overall population variance (assuming all groups have equal variances). So when the null of all means equal is true then the 2 measures (with some extra terms for degrees of freedom) will be similar and the ratio will be close to 1. If the null is false, then the numerator will be large relative to the denominator and the ratio will be greater than 1.

From this we can derive that the ratio of the explained variance and the unexplained variance follows a F-distribution with

$$F = \frac{\sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 / (K-1)}{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 / (n-K)} \sim F_{K-1, n-K} \quad \text{under the global null-hypothesis} \quad H_0$$

Looking up this ratio on the F-table (or computing it with a function like pf in R) will give the p-value. When evaluating the test statistic, we consider the the $1 - \alpha$ quantile of the F-distribution. Why is this not a two-sided test?

What do we do when we want to test compound hypothesis? Let us suppose for example we want to test whether a model with four covariates fit the data better than the model with just two covariates. There are three different tests that can test for these types of linear restrictions on the model. Here, we will restrict ourselves to the Wald test.

Testing linear combinations of hypotheses (so-called **linear restrictions**) on $\beta_1, \ldots, \beta_K$:

$$\mathrm{H}_0 : \mathbf{R}\boldsymbol{\beta} = \mathbf{r},$$

where the $(\#\mathbf{r} \times K)$ dimensional matrix $\mathbf{R}$ and the vector $\mathbf{r}$ are known an specified by the hypothesis, and $\#\mathbf{r}$ is the number of elements in $\mathbf{r}$ (i.e., the number of linear equations in the nullhypothesis). To make sure that there are no redundant equations it is required that $rank(R) = \#\mathbf{r}.\backslash$

Based on the normality assumption we can test the nullhypothesis using the χ^2-distributed test statistic

$$\mathrm{W} = \frac{(\mathbf{R}\hat{\boldsymbol{\beta}} - \mathbf{r})'(\mathbf{R}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}')^{-1}(\mathbf{R}\hat{\boldsymbol{\beta}} - \mathbf{r})}{\sigma^2} \sim \chi^2_{\#\mathbf{r}},$$

where $\chi^2_{\#\mathbf{r}}$ denotes the χ^2-distribution with $\#\mathbf{r}$ degrees of freedom. If σ^2 is unknown we have to plug-in its estimator $\hat{\sigma}^2$, which then changes the distribution of the test statistic:

$$\begin{aligned} \mathrm{F}_{Wald} &= \frac{(\mathbf{R}\hat{\boldsymbol{\beta}} - \mathbf{r})'(\mathbf{R}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{R}')^{-1}(\mathbf{R}\hat{\boldsymbol{\beta}} - \mathbf{r})}{\hat{\sigma}^2 \#\mathbf{r}} \sim F_{\#\mathbf{r},n-K}, \\ \text{alternatively} & \\ \mathrm{F}_{Wald} &= \frac{(\mathbf{R}\hat{\boldsymbol{\beta}} - \mathbf{r})'(\mathbf{R}\,\widehat{Cov}(\hat{\boldsymbol{\beta}})\,\mathbf{R}')^{-1}(\mathbf{R}\hat{\boldsymbol{\beta}} - \mathbf{r})}{\#\mathbf{r}} \sim F_{\#\mathbf{r},n-K}, \end{aligned}$$

where $F_{\#\mathbf{r},n-K}$ is the F-distribution with $\#\mathbf{r}, n-K$ degrees of freedom.

Example 2 Consider the linear regression model with four parameters $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \beta_3)$. We want to test the hypothesis that

$$H_0 : \beta_2 = 0$$

$$\underset{R}{\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}} \underset{\beta}{\begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix}} = \underset{r}{(0)} \tag{8}$$

```
###R Code for the Wald test, Example 1###
set.seed(50)
N=10000
M=3##number of variables except for the intercept.
##incidence###
k1<-rep(1, N)##generating the constant.
beta.vec  <- c(1,-0.2,0.5,0.10)
X.1<-matrix( rnorm(N*M,mean=0,sd=1), N, M)
X         <- cbind(k1,X.1)
eps       <-rnorm(N, 0,10)#
###generate the model####
y         <- X %*% beta.vec + eps
##Solving for beta hat###
beta.hat <- solve(t(X) %*% X) %*% t(X) %*% y
beta.hat
```

```
##          [,1]
## k1  1.0899015
##    -0.2028447
##     0.4213416
##     0.2257893
```

```
eps.hat<-y-X %*% beta.hat
se<-(t(eps.hat)%*%(eps.hat))/(N-(M+1))
cov<-se[1]*solve(t(X) %*% X)
###Specify the restrictions vector#
#beta_2=0#
R<-c(0,1,-1,0)
###We need to force the vector into a row vector to make the
R<-rbind(R)
b<-beta.hat
r<-0
W<-(t(R%*%b-r)*solve((R%*%cov%*%t(R)))*(R%*%b-r))/(length(r))
W
```

```
##             R
## [1,] 19.33328
```

```
####Calculate the critical value of the F distribution with the correct degrees of freedom##
f.crit<-qf(.95, df1=length(r), df2=N-M+1)
f.crit
```

```
## [1] 3.842389
```

Confidence intervals

Similarly to the t-tests, one can derive confidence intervals for the unknown parameters β_j:

$$b_j \pm \sqrt{\hat{\sigma}^2 (X'X)^{-1}_{jj}} \times t_{n-k;1-\alpha/2} \tag{9}$$

```
The interpretation of the confidence interval is very important: with $\alpha=0.05$, i.e. the standar

\bigskip

\textcolor{red}{Incorrect (but often encountered) interpretation}: There is a 95\% chance that the pa
  Seems to make sense right? Get the confidence level as high as you can! Well, as the confidence lev
```

```
##Confidence intervals##
set.seed(50)
N=10000
M=3##number of variables except for the intercept.
##incidence###
k1<-rep(1, N)##generating the constant.
beta.vec  <- c(1,-0.2,0.5,0.10)
X.1<-matrix( rnorm(N*M,mean=0,sd=1), N, M)
X         <- cbind(k1,X.1)
eps       <-rnorm(N, 0,10)#
y         <- X %*% beta.vec + eps
##Solving for beta hat###
beta.hat <- solve(t(X) %*% X) %*% t(X) %*% y
beta.hat
```

```
##          [,1]
## k1  1.0899015
##    -0.2028447
##     0.4213416
##     0.2257893
```

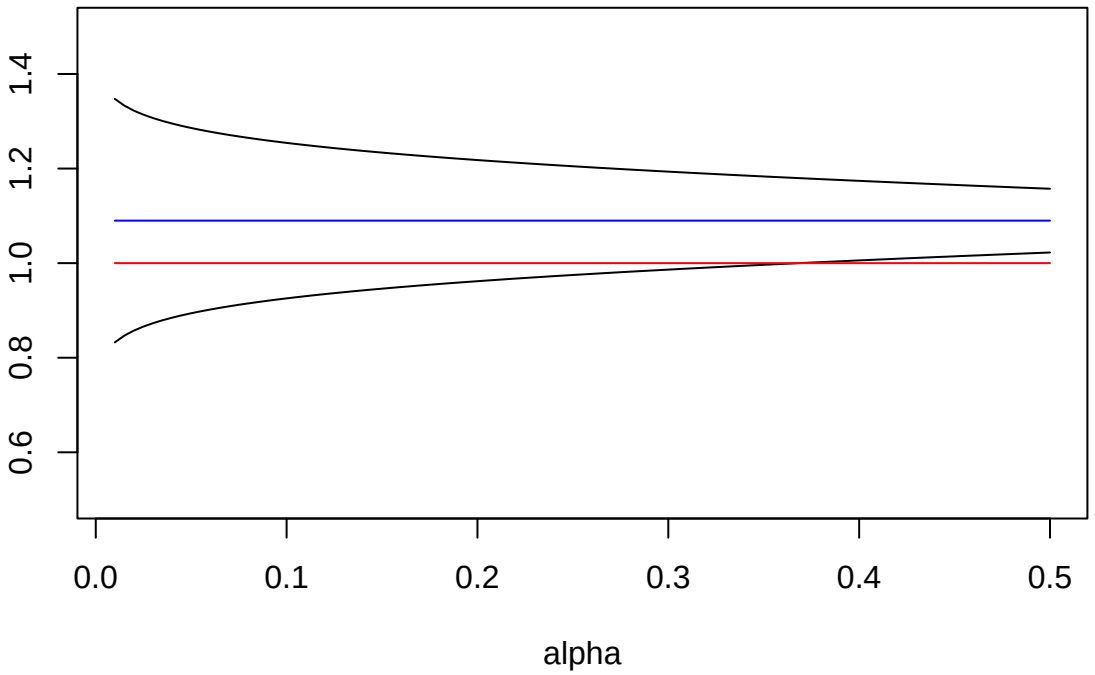
```r
####Now, let's calculate the  value of the standard deviation####
      xx<-dim(X)
      length.x<-xx[2]
      ###calculate the fitted model
      y.hat<- X %*% beta.hat
      #calculate the residuals
      eps.hat<-y-X %*% beta.hat
      ##calculate sigma.hat and the covariance matrix
      se<-(t(eps.hat)%*%(eps.hat))/(N-length.x)
      cov<-se[1]*solve(t(X) %*% X)
      d1<-sqrt(diag(cov))###vector of standard deviations
      #Confidence intervals#
      conf<-function(beta.hat,d1,alpha,dist)
      {
        if(dist==1)
        {
          crit<-abs(qnorm(alpha/2))
        }
        if(dist==2)
        {
          crit<-abs(qt(alpha/2, N-length.x))
        }
        conf<-cbind((beta.hat-(d1*crit)),((beta.hat+(d1*crit))))
        return(conf)
      }

      ##calculating the confidence interval for alpha=0.05, normal critical
      ##values
      conf(beta.hat,d1,0.05,2)
```

```
##           [,1]         [,2]
## k1  0.89400492  1.285798116
##    -0.39914471 -0.006544699
##     0.22409126  0.618591934
##     0.02972906  0.421849562
```

```r
      alpha<-seq(0.01,0.5,le=100)
      conf_high=c()
      conf_low=c()
      a=length(alpha)
      for(i in 1:a)
      {
        A=conf(beta.hat,d1,alpha[i],1)
        conf_low[i]=A[1,1]
        conf_high[i]=A[1,2]
      }

      ###Plot our results####
      plot(alpha,conf_low,ylim=c(0.5,1.5), type="l",ylab="")
      lines(alpha,conf_high)
      lines(alpha,beta.hat[1]*rep(1,length(alpha)), col="blue")
      lines(alpha,beta.vec[1]*rep(1,length(alpha)),col="red")
```




```
      ##Check our work##

      lm.result   <- lm(y~X)
      confint(lm(y~X.1[,1]+X.1[,2]+X.1[,3]), level = 0.95)
```

```
##                   2.5 %       97.5 %
## (Intercept)  0.89400492  1.285798116
## X.1[, 1]    -0.39914471 -0.006544699
## X.1[, 2]     0.22409126  0.618591934
## X.1[, 3]     0.02972906  0.421849562
```

```
  \bigskip

  \subsubsection{Size and Power }

  We conclude this section by thinking some more about hypothesis tests and review some terminology.

  \bigskip

  \textbf{The Size:} The \textbf{size} of a hypothesis test is the probability of a (undesired) false r
  For compound nullhypotheses, e.g.~H$_0:\beta_k\leq\bar\beta_k$, the test's size is defined to be its
```

The Power: The **power** of a hypothesis test *against a given alternative* is the probability of a (desired) rightful rejection of a false nullhypothesis (avoiding a type II or β error); see Figure 1. Factors that affect statistical power include the sample size, the specification of the parameter(s) in the null and alternative hypothesis, i.e. how far they are from each other, the precision or uncertainty the researcher allows for the study (generally the confidence or significance level) and the distribution of the parameter to be estimated. Ceteris Paribus, the power is a continous function of the sample size and the specification of the parameters. In general, power calculations can be used in two ways:

1. Before the data collection: calculate the necessary sample size for a hypothesized parameter size for given data assumptions.
2. After data collection, to check whether insignificant results can be attributed to an insufficient sample

size for an estimated effect size.

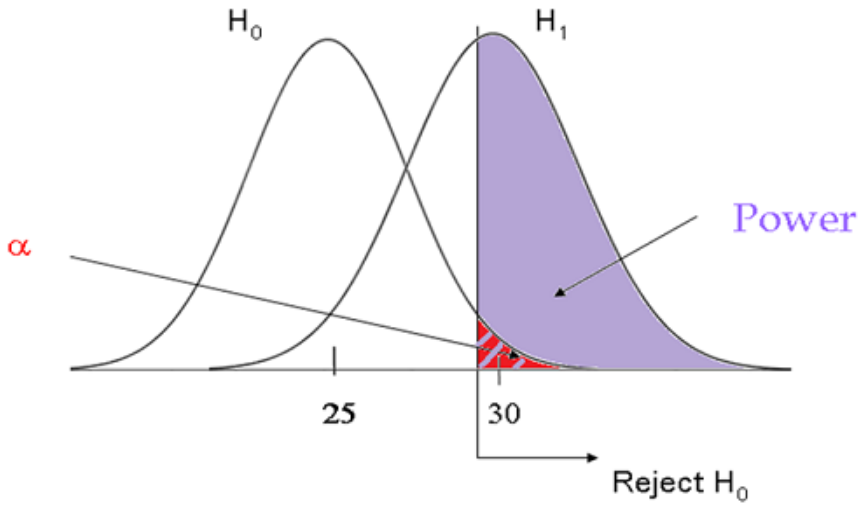


Figure 1: Visualization of size and power of a statistical hypothesis test (fix sample size n).

Consistent Tests: As a minimal criterion for hypothesis tests, we demand that the power to reject any false nullhypothesis goes asymptotically (as $n \to \infty$) against one. A test that fulfills this minimal criterion is termed a **consistent test**.

For instance, the t-test is a consistent test, since (under assumptions 1.1-1.5, but a false nullhypothesis) its numerator $\sqrt{n}(b_k - \bar{\beta}_k)$ will diverge to plus or minus infinity (with rate $\sqrt{n}$) if $b_k \xrightarrow{p} \beta_k \neq \bar{\beta}_k$, whereas its denominator still converges. The argument for the Wald test is similar. So, the power of a continuous test is also a monotonically increasing function of the sample size n.

Ideally, we want to use test statistics that have a very small size, but a very high power, though, these are conflicting aims; see Figure 1. This problem is usually resolved by committing to a pre-specified size - say $\alpha = 0.05$ - and then attempting to maximize a test's power for a given sample size n.

```
\subsection{Analysis of residuals and checking of model assumptions}

The residuals $\hat{\varepsilon}_i=y_i-\hat{y}_i$ can serve of an approximation of the unobservable e

\subsubsection{The Tukey-Anscombe Plot}
The Tukey-Anscombe is a graphical tool: we plot the residuals $\hat{\varepsilon}_i$ (one the y-axis)
\smallskip
In the ideal case, the points in the Tukey-Anscombe plot "fluctuate randomly" around the horizontal l
\begin{figure}
\includegraphics[width=0.45\textwidth]{plot_homoskedastic.pdf}
\includegraphics[width=0.45\textwidth]{plot_heteroskedastic.pdf}
\caption{Tukey-Anscombe Plot, plotting error residuals against fitted values $\hat{y}_i$}\label{fig:h
\end{figure}


\subsubsection{The Normal plot}
Assumptions for the distribution of random variables can be graphically checked with the QQ (quantile
\smallskip

In the linear model application, we plot the empirical quantiles of the residuals (on the y-axis), ve
```

```
\begin{figure}
\includegraphics[width=0.45\textwidth]{plot_qq_normal.pdf}
\includegraphics[width=0.45\textwidth]{plot_qq_notnormal.pdf}
\caption{QQ Plot, plotting empirical residuals against the quantiles of the standard normal distributi
\end{figure}




\subsubsection{Detecting serial correlation}
For checking independence of the errors we plot the residuals $\hat{\varepsilon}_i$ versus the observa
```

````
```r
N<-1000
X.1       <- rep(1, N)
X.2       <- rnorm(N, mean=10, sd=1.5)
X         <- cbind(X.1, X.2)
#serial correlation
eps_1 <-  diffinv(rnorm(999),lag=1)#Computes the inverse function of the lagged differences function o
#nonstationary, serially correlated erros through a brownian motion.
# eps_2<- filter(rnorm(N), filter=rep(1,1), circular=TRUE)# stationary serial correlation
# plot(eps_2)
beta.vec  <- c(1,-5)
#beta.vec  <- c(1)
y_1         <- X %*% beta.vec + eps_1
beta.hat.vec1 <- solve(t(X) %*% X) %*% t(X) %*% y_1
beta.hat.vec1
```

```
##          [,1]
## X.1  9.470060
## X.2 -4.740692
```

```r
y.hat1<- X %*% beta.hat.vec1
eps.hat1<-y_1-X %*% beta.hat.vec1
plot(eps.hat1, main="Estimated Residuals")
```

![](script_markdownII_files/figure-latex/serial-1.pdf)<!-- -->
````

```
\section{Nonparametric Density Estimation}
\subsection{Introduction}
For a moment we will go back to simple data structures: we have observations which are realizations o
```

```
  $$x_1, x_2, ..., x_n \, i.i.d. \sim F,$$
    where $F$ is an unkown cumulative distribution function. The goal is to estimate the distribution $
```

Instead of assuming a parametric model for the distribution, e.g. the normal distribution with unkown mean and variance, we aim to be as "general as possible" or also, as "data driven as possible". This means that we only assume that the density exists is suitably smooth (e.g. usually this means at least twice differentiable). It is then possible to estimate the unknown density function $f(\cdot)$. Mathematically, a function is an **infinite-dimensional object**. Density estimation will become a "basic principle" how to do estimation for infinite-dimensional objects. We will make use of such a principle also in the section on nonparametric regression.

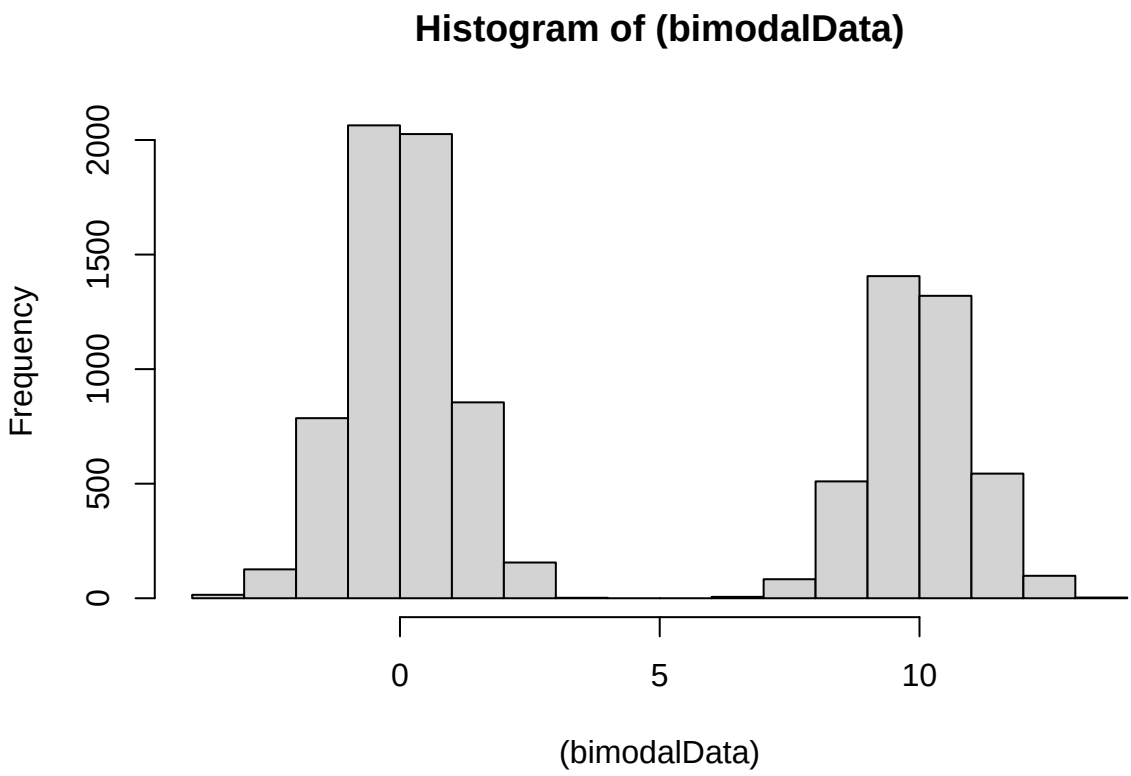
Estimation of a Density

We simulate a bi-modal data set according to:

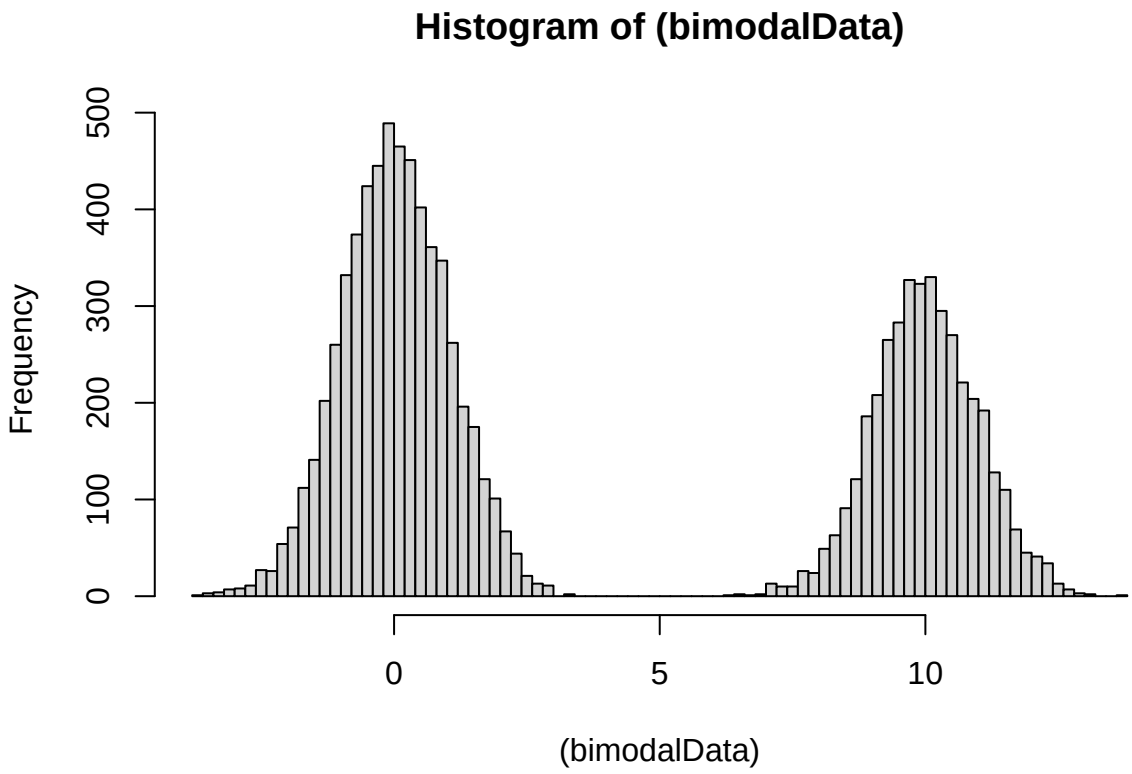
```r
#Nonparametric density estimation
mu1 <- (0)   #The first mean
mu2 <- (10) #The second mean
sig1 <- (1) # Standard deviation
sig2 <- (1)
p <- 0.4 #  Success probability of the binomial, determining
n=10000

bimodalDistFunc <- function (n,p, mu1, mu2, sig1, sig2) {
  y0 <- rnorm(n,mean=mu1, sd = sig1)
  y1 <- rnorm(n,mean=mu2, sd = sig2)
  flag <- rbinom(n,size=1,prob=p)
  y <- y0*(1 - flag) + y1*flag
  return(y)
}

bimodalData <- bimodalDistFunc(n=10000,p,mu1,mu2, sig1,sig2)
hist((bimodalData),breaks=20)
```




```
hist((bimodalData),breaks=100)
```



Histogram

The histogram is the oldest and most popular density estimator. We need to specify an origin x_0 and the class width h for the specifications of the intervals

$$I_j = (x_0 + j \cdot h, x_0 + (j+1) \cdot h](j = ..., -1, 0, 1, ...)$$

for which the histograms counts the number of observations falling into each I_j: we then plot the histogram such that the area of each bar is proportional to the number of observations falling into the corresponding class (interval I_j). The choice of the origin x_0 is highly arbitrary, whereas the role of the class width is immediately clear for the user. The form of the histogram depends very much on these two tuning parameters.

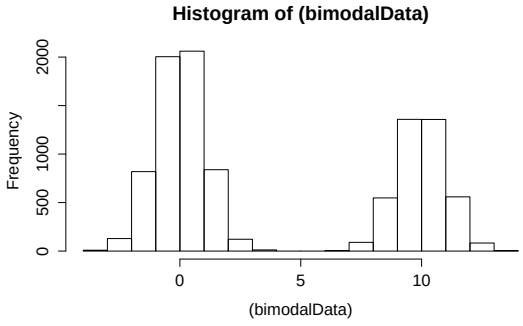
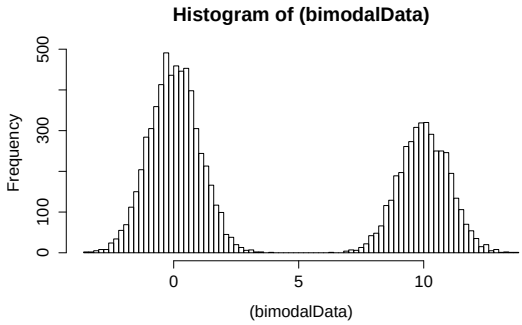


Figure 2: Histogram with two different binwidths H



Kernel estimator

The naive estimator Similar to the histogram, we can compute the relative frequency of observations falling into a small region. The density function $f(\cdot)$ at a point x can be represented as

$$f(x) = \lim_{h \to 0} \frac{1}{2h} \mathbb{P}[x - h < X \leq x + h] \tag{10}$$

The naive estimator is then constructed without taking the limit and by replacing probabilities with relative frequencies:

$$\hat{f}(x) = \frac{1}{2hn} \#\{i; X_i \in (x - h < X \leq x + h]\}. \tag{11}$$

The naive estimator is only piecewise constant since every X_i is either in or out of the interval

$$(x - h < X \leq x + h]$$

. As for histograms, we need to specify the so-called bandwidth h, but we do not need to specify an origin x_0. An alternative representation of the naive estimator is as follows. Define the weight function

$$w(x) = \begin{cases} 1/2 if |x| \leq 1, \\ 0 \quad \text{otherwise} \end{cases}$$

then

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} w\left(\frac{x - X_i}{h}\right)$$

If we choose instead of the rectangle weight function $w()$ a general, typically more smooth kernel function $K(\cdot)$, we have the definition of the kernel density estimator

$$\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) \tag{12}$$

$$K(x) \geq 0, \int_{-\infty}^{\infty} K(x)dx = 1, K(x) = K(-x) \tag{13}$$

The estimator depends on the bandwidth $h > 0$ which acts as a tuning parameter. For a large bandwidth h, the estimate $\hat{f}(x)$ tends to be very slowly varying as a function of x, while small bandwidths will produce a more wiggly function estimate. The positivity of the kernel function $K(\cdot)$ guarantees a positive density estimate $\hat{f}(x)(\cdot)$ and the normalization $\int_{-\infty}^{\infty} K(x)dx = 1$ implies that $\int_{-\infty}^{\infty} \hat{f}(x)dx = 1$ which is necessary for $\hat{f}(x)(\cdot)$ to be a density. Typically, the kernel function $K(\cdot)$ is chosen as a probability density which is symmetric around 0. The smoothness of $\hat{f}(x)(\cdot)$ is inherited from the kernel: if the rth derivative $K^r(x)$ exists for all x, then $\hat{f}^r(x)$ exists as well for all x (easy to verify using the chain rule for differentiation). Popular kernels are the Gaussian Kernel

$$K(x) = (2\pi)^{-1/2} e^{-x^2/2} \quad \text{the density of the } \mathcal{N}(0,1)$$

or a kernel with finite support such as $K(x) = \frac{\pi}{4} cos\,(\pi 2x)\,\mathbf{1}(|x| \leq 1)$. The Epanechnikov kernel, which is optimal with respect to mean squared error, is

$$K(x) = \frac{3}{4}\left(1 - |x|^2\right)\mathbf{1}(|x| \leq 1)$$

Far more important than the choice of kernel is the choice of the bandwidth, as we will see in the next section.

The role of the bandwidth

The bandwidth h is often also called the "smoothing parameter": for $h \to 0$, we will have "δ-spikes" at every observation X_i, whereas $\hat{f}(\cdot) = \hat{f}_h(\cdot)$ becomes smoother as h increases.

The bias-variance trade-off

We can formalize the behavior of $\hat{f}(\cdot)$ when varying the bandwidth h in terms of bias and variance of the estimator. It is important to understand heuristically that the **the absolute value of the bias of $\hat{f}$ increases and the variance of $\hat{f}$ decreases.**

Therefore, if we want to minimize the mean squared error ($MSE(\hat{f}(x))$) at a point x,

$$MSE(\hat{f}(x)) = \mathbb{E}\left[\left(\hat{f}(x) - f(x)\right)^2\right] = \left([\hat{f}(x) - f(x)]\right)^2 + Var(\hat{f}(x))$$

we are confronted with the **bias-variance trade-off**. As a consequence, this allows, at least conceptually, to optimize the bandwidth parameter (namely to minimize the mean squared error) in a well-defined, coherent way. Instead of optimizing the mean squared error at a point x, one my want to optimize the integrated mean squared error (IMSE)

$$IMSE = \int MSE(x)dx$$

which yields an integrated decomposition of squared bias and variance (integration is over the support of X). Since the integrand is non-negative, the order of integration (over the support of X and over the probability space pf X) can be reversed, denoted as MISE (mean integrated squared error) and written as

$$MISE = \mathbb{E}\left[\int \left(\hat{f}(x) - f(x)\right)^2 dx\right] = \mathbb{E}\left[ISE\right] \tag{14}$$

where $ISE = \int \left(\hat{f}(x) - f(x)\right)^2 dx$.

Asymptotic bias and variance

It is straightforward (using definitions) to give an expression for the exact bias and variance:

$$\mathbb{E}\left[\hat{f}(x)\right] = \int \frac{1}{h} K\left(\frac{x - X_i}{h}\right) f(X_i) dX_i \tag{15}$$

$$\tag{16}$$

$$\begin{aligned}
Var(\hat{f}(x)) &= \frac{1}{nh^2} Var\left(K\left(\frac{x - X_i}{h}\right)\right) && (17)\\
&= \frac{1}{nh^2}\mathbb{E}\left[K\left(\frac{x - X_i}{h}\right)^2\right] - \frac{1}{nh^2}\mathbb{E}\left[K\left(\frac{x - X_i}{h}\right)\right]^2 && (18)\\
&= n^{-1}\int \frac{1}{h^2} K\left(\frac{x - X_i}{h}\right)^2 f(X_i) dX_i - n^{-1}\left(\int \frac{1}{h} K\left(\frac{x - X_i}{h}\right) f(X_i) dX_i\right)^2 && (19)
\end{aligned}$$

For the bias we therefore get (by a change of variable and $K(-z) = K(z)$)

$$\begin{aligned}
Bias(\hat{f}(x)) &= \int \frac{1}{h} K\left(\frac{x - X_i}{h}\right) f(X_i) dX_i - f(x) && (20)\\
&\underbrace{=}_{z=(X_i - x)/h, dz = 1/h, X_i = x + zh} \int K(z) f(x + hz) dz - f(x) && (21)\\
&= \int K(z)(f(x + hz) - f(x)) dz && (22)
\end{aligned}$$

To approximate this expression in general, we invoke an asymptotic argument. We assume that $h \to 0$ as sample size $n \to \infty$, that is:

$$h = h_n \to 0 \text{ with } nh_n \to \infty.$$

This will imply that the bias goes to zero since $h_n \to 0$; the second condition requires that h_n is going to zero more slowly than $\frac{1}{n}$ which turns out to imply that also the variance of the estimator will go to zero as $n \to \infty$.To see this, we use a Taylor expansion of f, assuming that f is sufficiently smooth: $f(x + hz) = f(x) + hzf'(x) + \frac{1}{2}h^2 z^2 f''(x) + ...$ plugging this into 22 yields

$$\begin{aligned}
Bias(\hat{f}(x)) &= \underbrace{\int zK(z)dz}_{=0} + \frac{1}{2}h^2 f''(x)\int z^2 K(z)dz + \dots && (23)\\
&= \frac{1}{2}h^2 f''(x)\int z^2 K(z)dz + \text{ higher order terms in } h. && (24)
\end{aligned}$$

For the variance, we get from (19)

$$\begin{aligned} Var(\hat{f}(x)) &= n^{-1}\int \frac{1}{h^2}K\left(\frac{x-X_i}{h}\right)^2 f(X_i)dX_i - n^{-1}(f(x)+Bias(\hat{f}(x)))^2 && (25)\\ &= n^{-1}h^{-1}\int f(x-hz)K(z)^2dz - \underbrace{n^{-1}(f(x)+Bias(\hat{f}(x)))^2}_{=O(n^{-1})} && (26)\\ &= n^{-1}h^{-1}\int f(x-hz)K(z)^2dz + O(n^{-1}) = n^{-1}h^{-1}f(x)\int K(z)^2dz + o(n^{-1}h^{-1}) && (27)\end{aligned}$$

assuming that f is smooth and hence $f(x-hz)\to f(x)$ as $h_n\to 0$. In summary: for $h=h_n\to 0$, $h_n n\to\infty$ as $n\to\infty$.

$$\begin{aligned} Bias(\hat{f}(x)) &= h^2 f''(x)\int z^2K(z)dz/2 + o(h^2) \quad (n\to\infty)\\ Var(\hat{f}(x)) &= n^{-1}h^{-1}f(x)\int K(z)^2dz + o(n^{-1}h^{-1}) \quad (n\to\infty)\end{aligned}$$

The optimal bandwidth $h=h_n$ which minimizes the leading term in the asymptotic $MSE(x)$ can be calculated straightforwardly by solving $\frac{\partial}{\partial h}MSE(x)=0$,

$$h_{opt}(x) = n^{-1/5}\left(\frac{f(x)\int K^2(z)dz}{(f''(x))^2(\int z^2K(z)dz)^2}\right)^{1/5} \tag{28}$$

Since it is not straightforward to estimate and use a local bandwidth $h(x)$, one rather considers minimizing the MISE, i.e. $\int MSE(x)dx$ which is *asymptotically*

$$asympt.MISE = \int Bias(x)^2 + Var(\hat{f}(x))dx = \frac{1}{4}h^4R(f'')\sigma_K^4 + R(K)/(nh) \tag{29}$$

where $R(g)=\int g^2(x)dx$, $\sigma_K^2=\int x^2K(x)dx$ and the "global" asymptotically optimal bandwidth becomes

$$h_{opt} = n^{-1/5}(R(K)/\sigma_K^4 \times 1/R(f''))^{1/5} \tag{30}$$

By replacing h with h_{opt}, e.g. in 29 we see that both variance and bias terms are of order $O(n^{-4/5})$, the optimal rate for the MISE and $MSE(x)$. This rate is also optimal for a much larger class of density estimators.

Estimating the bandwidth

The plug-in method As seen from 30, the asymptotically best bandwidth depends on $R(f'')=\int f''^2(x)dx$ which is unknown (whereas $R(K)$ and σ_K^2 are known). It is possible to estimate the f'' again by a kernel estimator with an "initial" bandwidth h_{init} (sometimes called a pilot bandwidth) yielding $\hat{f}''$. Plugging this estimate into 30 yields an estimated bandwidth $\hat{h}$ for the estimator $\hat{f}(\cdot)$ (the original problem). $\hat{h}$ obviously depends on the initial bandwidth h_{init}, but choosing h_{init} in an ad-hoc way has much smaller consequences than choosing the bandwidth h itself. Furthermore, methods have been developed to choose h_{init} and h simulatenously.

Least-squares Cross-Validation The most commonly used data-driven bandwidth selection algorithm is least-squares cross validation. The goal of LSCV is to minimize the difference between the estimator of the

density and the density itself. We define the **integrated squared error (ISE)** as

$$ISE(\hat{f}, f) = \int \left(\hat{f}(x) - f(x)\right)^2 dx \tag{31}$$

$$= \int \hat{f}(x)^2 dx - 2\int \hat{f}(x)f(x)dx + \int f(x)^2 dx \tag{32}$$

The last term only involves the true density function and therefore does not depend on the chosen smoothing parameter. Therefore, when we would like to minimize the $ISE(\hat{f}, f)$ with respect to the bandwidth h, we minimize

$$ISE^*(\hat{f}, f) = \int \hat{f}(x)^2 dx - 2\int \hat{f}(x)f(x)dx \tag{33}$$

Since the true density $f(x)$ is unknown, we must select an estimator for $f(x)$. We could, of course, choose a kernel density estimator; however, this would lead to minimizing $ISE^*(\hat{f}, \hat{f}) = -/int\hat{f}(x)^2 dx$, which would result in a bandwidth of 0, regardless of the underlying density. This is because setting a bandwidth of 0 places weight only on the sample observations and integrating over a fixed number of points will return $-\infty$. As an alternative, we consider the leave-one-out estimator

$$\hat{f}_{-i}(x) = \frac{1}{(n-1)\,h} \sum_{j=1, j\neq i}^{n} K\left(\frac{x_j - x}{h}\right) \tag{34}$$

which is the density estimator constructed using all of the observations except x_i. At first glance you may feel that there is a typo in the equation above, since i does not appear on the right-hand side save for the summation. Our data vector $(x_1, x_2, ..., x_i, ..., x_n)$ contains the $i'th$ observation, but we only sum over $(x_1, x_2, ..., x_{i-1}, x_{i+1}..., x_n)$ when calculating $\hat{f}_{-i}(x)$. There is no reason why we should leave out the $i'th$ observation, we could have omitted the $j'th$ or the $l'th$ observation instead. Accordingly, we construct an estimator based on averaging our leave-one-out estimator over all observations.

$$\hat{f}_{-i}(x) = n^{-1} \sum_{j=1}^{n} \hat{f}_{-j}(x) \tag{35}$$

When we insert $\hat{f}_{-i}(x)$ into 33 we can find the bandwidth h that minimizes the cross-validation criterion through a simple grid search. It can be shown that minimizing $ISE^*(\hat{f}(x), \hat{f}_{-i}(x))$ w.r.t. h is equivalent to minimizing ISE. This method works quite well (usually), but is very intensive in terms of computing time. There are some methods that use convolution kernels to reduce computing time, but we will not discuss these here.

Which of these data-driven methods one should use is up for debate and we will illustrate this in an example in the take home exercise.

Higher dimensions

Many applications involve multivariate data. For simplicity, consider data which are i.i.d. realizations of d-dimensional random variables

$$\mathbf{X}_1, ..., \mathbf{X}_n \sim f(x_1, ..., x_d)d_{x_1}...d_{xd}$$

where $f(\cdot)$ denotes the multivariate density.

The multivariate kernel density is, in its simplest form, defined as

$$\hat{f}(\mathbf{x}) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{X_i}}{h}\right),$$

where the kernel $K(\cdot)$ is now a function, defined for d-dimensional $\mathbf{x}$, satisfying

$$K(\mathbf{u}) \geq 0, \quad \int_{\mathbb{R}^d} K(\mathbf{u})du = 1, \int_{\mathbb{R}^d} \mathbf{u}K(\mathbf{u})du = 0, \int_{\mathbb{R}^d} \mathbf{u}\mathbf{u}^T K(\mathbf{u})du = I_d$$

. Usually, the kernel is chosen as a product of a kernel K_{univ} for univariate density estimation

$$K(\mathbf{u}) = \prod_{j=1}^{d} K_{univ}(u_j)$$

The curse of dimensionality

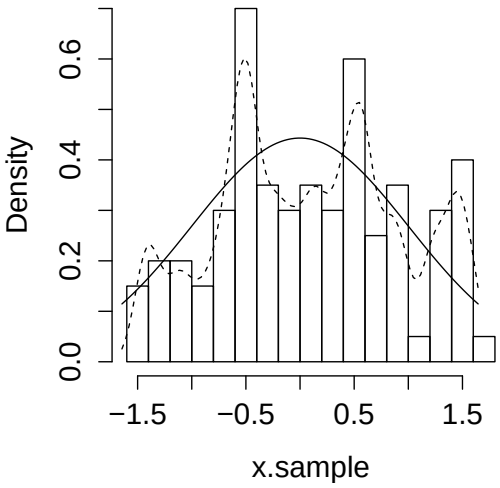
In practice is multivariate kernel estimation often restricted to dimension $d = 2$. The reason is, that a higher dimensional space (with d of medium size or large) will be only very sparsely populated by data points. Or in other words, there will be only very few neihgbouring data points to any value $\mathbf{x}$ in a higher dimensional space, unless the sample size is extremely large. This phenomenon is also known as the curse of dimensionality. An implication of the curse of dimensionality is the following lower bound for the best mean squared error of nonparametric density estimators (assuming that the underlying density is twice differentiable): it has been shown that the best possible MSE rate is

$$O(n^{-4/(4+d)})$$

The following table evaluates $n^{-4/(4+d)}$ for various n and d:

$n^{-4/(4+d)}$	$d = 1$	$d = 2$	$d = 3$	$d = 5$	$d = 10$
n=100	0.025	0.046	0.072	0.129	0.268
n=1000	0.004	0.010	0.019	0.046	0.139
n=100.000	$1.0\times^{-4}$	$4.6.0\times^{-4}$	$13.9\times^{-4}$	0.006	0.037

Thus, for $d = 10$, the rate with $ n=100.000$ is still 1.5 times worse than for $d = 1$ and $n = 100$.



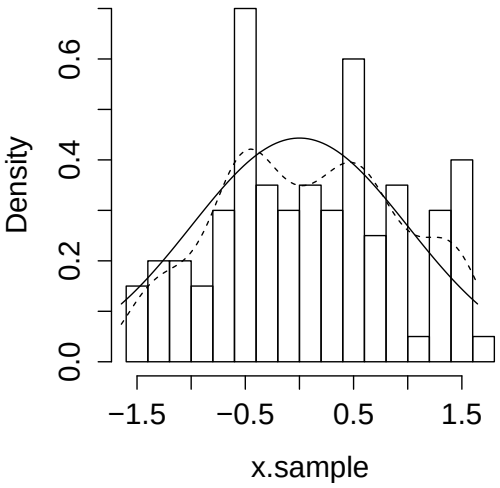


Figure 3: Histogram, true model (solid line) and estimated density with four different bandwidths.