Script Computergestütze Statistik

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

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

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 $k = 3$:

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

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

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

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

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

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

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

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. Strict exogeneity is a very strong assumption and we will relax it later on. For one example, it cannot be fulfilled when the regressors include lagged dependent variables.

Some Implications of Strict Exogeneity:

- The unconditional mean of the error term is zero:

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

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,

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

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: $\hat{\boldsymbol{\varepsilon}} = \mathbf{y} - \hat{\mathbf{y}} = \left(\mathbf{I}_n - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\right)\mathbf{y}$,

Proposition For the OLS residuals and the OLS fitted values it holds that

$$\begin{aligned} \mathbf{X}'\hat{\varepsilon} &= \mathbf{0}, \quad \text{and} \\ \mathbf{y}'\mathbf{y} &= \hat{\mathbf{y}}'\hat{\mathbf{y}} + \hat{\varepsilon}'\hat{\varepsilon}. \end{aligned}$$

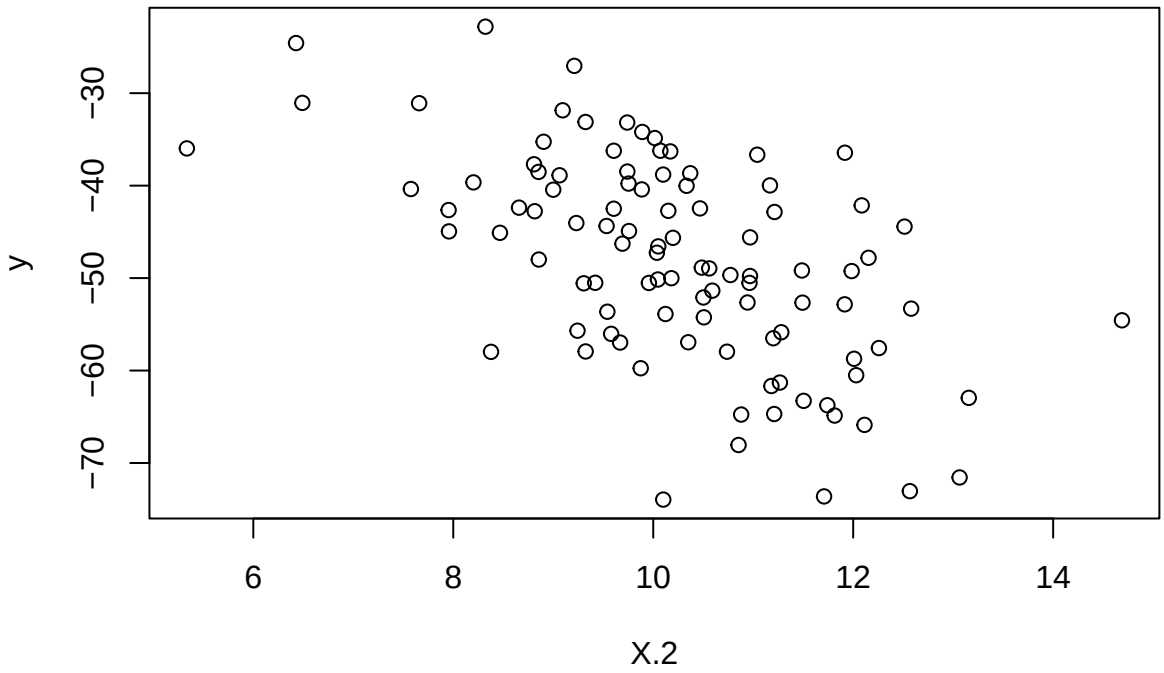
```
N         <- 100 # Number of observations
X.1       <- rep(1, N)
X.2      <- rnorm(N, mean=10, sd=1.5) # (pseudo) random numbers form a normal distr
X         <- cbind(X.1, X.2)
###Homoscedastic error term
eps       <-rnorm(N, 0,10)#
beta.vec  <- c(5,-5)
## Specify the data generating process#####
y         <- X %*% beta.vec + eps

beta.hat <- solve(t(X) %*% X) %*% t(X) %*% y
beta.hat
```

```
##          [,1]
## X.1 -4.575029
## X.2 -4.261374
```

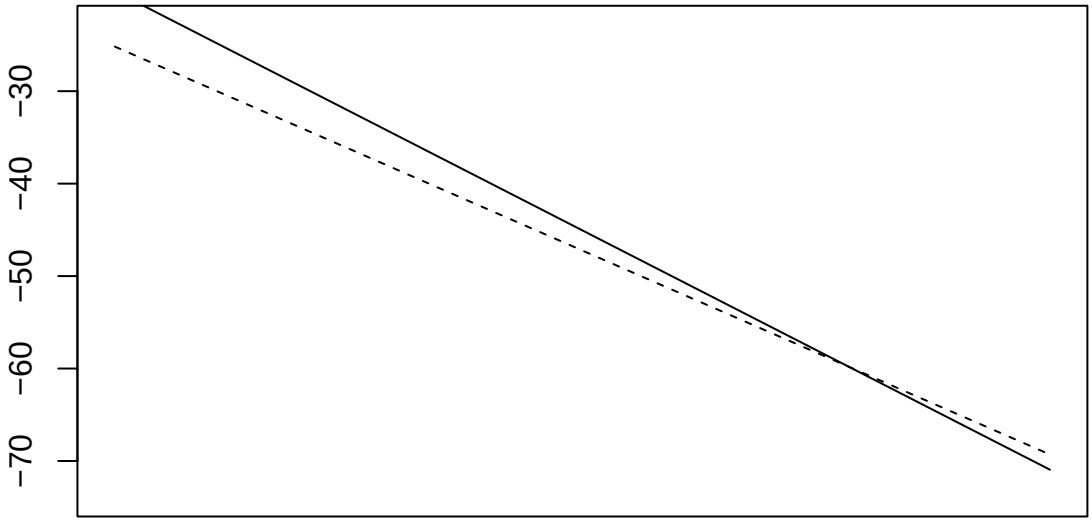
```
###########Write the results of Beta.hat into a CSV file###
as.data.frame(beta.hat)
```

```
##            V1
## X.1 -4.575029
## X.2 -4.261374
```

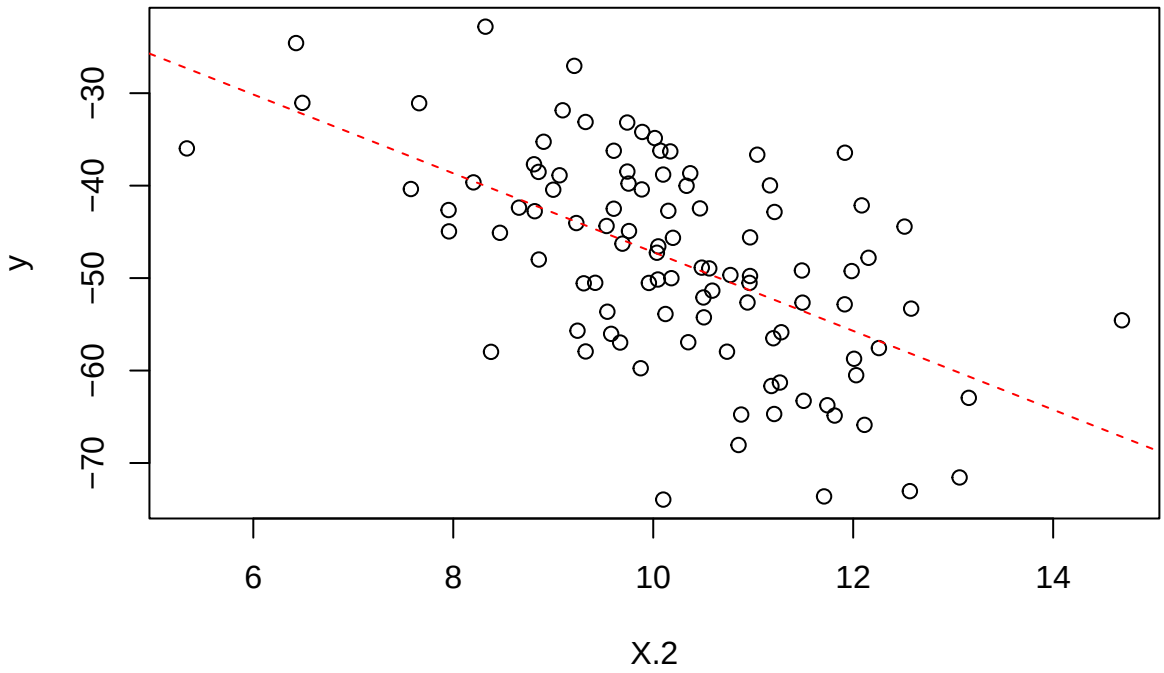
```
rownames(beta.hat)<-c("intercept","varone")
write.csv(t(beta.hat), "OLS_data.csv", row.names=FALSE, quote=FALSE)
###plot the model and the regression line###
x<-seq(min(X.2-0.5),max(X.2+0.5),length.out=100)
plot(X.2,y)
```



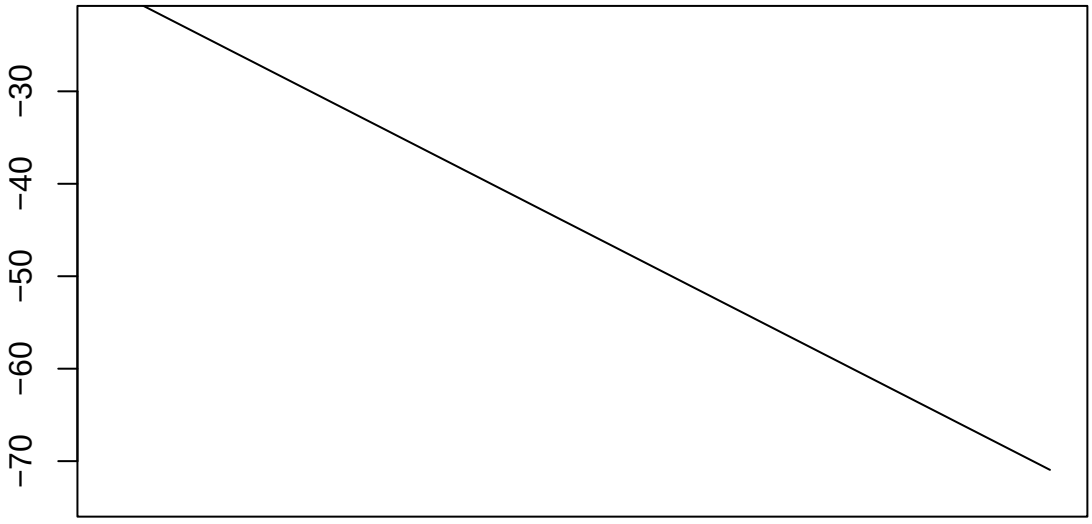

```
plot(beta.vec[1]+beta.vec[2]*x,xlab="", xaxt='n',ylab="",type="l",lty=1,ylim=c(min(y),max(y)))
lines(beta.hat[1]+beta.hat[2]*x,xlab="", xaxt='n',ylab="",lty=2,ylim=c(min(y),max(y)))
```




```
 #####We can do the same thing with a pre-installed R package#############
lm.result   <- lm(y~X)
plot(X.2,y)
abline(lm(y ~ X.2),lty=2,col="red")
```




```
plot(beta.vec[1]+beta.vec[2]*x,xlab="", xaxt='n',ylab="",type="l",lty=1,ylim=c(min(y),max(y)))
```




```
lm.summary  <- summary(lm.result)
lm.summary
```

```
##
## Call:
## lm(formula = y ~ X)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.339  -6.216   0.240   6.484  18.916
##
```

```
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -4.5750     6.3125  -0.725     0.47
## XX.1              NA         NA      NA       NA
## XX.2         -4.2614     0.6108  -6.976  3.6e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.13 on 98 degrees of freedom
## Multiple R-squared:  0.3318, Adjusted R-squared:  0.325
## F-statistic: 48.67 on 1 and 98 DF,  p-value: 3.599e-10
```

```
#####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))
```

Properties of Least Squares Estimation

Least squares estimates are random variables: for new data from the same data-generating mechanism, the data would look differently every time and hence also the least squares regression lines. Figure **??** depicts the least squares regression lines, which are based on three different realizations from the same data-generating model (i.e. three simulations from a model). We see that the estimates are varying, which means that the estimated parameters are random themselves.

Moments of least squares estimates

We assume the usual linear model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \mathbb{E}(\boldsymbol{\varepsilon}) = 0, \quad Cov(\boldsymbol{\varepsilon}) = \mathbb{E}[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'] = \sigma^2 \mathbf{I}_{nxn} \tag{8}$$

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: $\mathbf{P} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$. 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. $\varepsilon_1, ...\varepsilon_n \quad$ 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 the Proposition above 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
```

```r
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
```

```r
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
```

```r
#####calculate the fitted values#####
y.hat<- X %*% beta.hat
```

```r
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
r_squared.adj
```

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

```r
##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
```

```r
#####calculate the fitted values#####
y.hat<- X %*% beta.hat
eps.hat<-y-X %*% beta.hat
###calculate the covariance matrix#
```

```r
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
r_squared.adj
```

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

```r
### 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
```

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

```
###
#####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
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
```

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

```
r_squared.adj
```

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

Multiple regressions for a single outcome

It can be tempting to replace multiple regression with many single regressions with a single covariate. However, this is in general not a good idea.

The following example illustrates why:

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

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

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

$$y_i = \hat{y}_i = 2x_{i1} - x_{i2} \quad \text{for all}\, i \quad (\hat{\sigma}^2 = 0). \tag{9}$$

The coefficients 2 and -1 respectively, describe how y is changing when varying either $ x_{1}$ or x_2 and keeping the other covariate constant. In particular, we see that y decreases when x_2 increases.

On the other hand, if we do a simple regression of y onto x_2 (while ignoring x_1), we obtain the least squares estimate

$$y_i = \hat{y}_i = \frac{1}{9}x_{i2} - \frac{4}{3} \quad \text{for all}\, i \quad (\hat{\sigma}^2 = 1.72). \tag{10}$$

The least squares regression line describes how y changes when varying x_2 while ignoring x_1. In particular, $\hat{y}$ increases when x_2 increases, in contrast to to multiple regression!

The reason for this is that x_1 and x_2 are highly correlated: if x_2 increases then also x_1 increases. Note that in the multiple regression solution, x_1 has a larger coefficient in absolute value than x_2 and hence, an increase in x_1 has a stronger influence for changing y than x_2. The correlation among the the covariates in general makes the interpretation of the regression coefficients more subtle: in the current setting, the coefficient β_1 quantifies the influence of x_1 on y after having subtracted the effect of x_2 on y.

In the case of orthogonal covariates, this is not a problem as can easily be seen by the following calculation. Orthogonality means that $\mathbf{X'X} = diag(\sum_{i=1}^n x_{i1}^2, ..., \sum_{i=1}^n x_{ik}^2)$ and hence the least squares estimator is

$$\hat{\beta}_j = \sum_{i=1}^n x_{ij}y_i / \sum_{i=1}^n x_{ij}^2 \quad (j = 1, ..., k), \tag{11}$$

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 relevant covariates are not considered in the regression, it is partly caused by what can be called near multicollinearity.

Tests and Confidence Regions

Hypothesis testing

The idea behind testing is to somehow quantify the likelihood of observing a certain outcome (data) under a given Null hypothesis, i.e. how likely is the observed data when positing that the true parameter is the one given under the Null hypothesis.

We make assumptions 1.1-1.5 (i.e. including normality of the errors). As stated above, this implies that $\hat{\boldsymbol{\beta}}$ are normally distributed.

If we are interested in testing whether the j'th covariate is statistically significant (relevant), we can test the Null hypothesis $H_{0,j} : \beta_j = 0$ against the alternative $H_{1,j} : \beta_j \neq 0$. We can then easily derive from the normal distribution of b_j that

$$\frac{b_j}{\sqrt{\sigma^2(\mathbf{X}'\mathbf{X})^{-1}_{jj}}} \sim \mathcal{N}(0,1) \tag{12}$$

Since σ^2 is unkown, this quantity is not useful, but if we substitute it with the estimate $\hat{\sigma}^2$ we obtain the so-called t-statistic

$$T_j = \frac{b_j}{\sqrt{\hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1}_{jj}}} \sim t_{n-k} \quad \text{under the null-hypothesis } H_{0,j} \tag{13}$$

which has a slightly different distribution than standard normal $\mathcal{N}(0,1)$. The corresponding test is then called the t-test. In practice we can thus quantify the relevance of individual outcome variables by looking at the test-statistic T_j or at the corresponding P-values, which might be more informative.

```
###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
```

```
####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

$$H_0 : \beta_2 = 0$$

$$\underset{R}{\begin{pmatrix} 0 & 0 & 1 & 0 \end{pmatrix}} \underset{\beta}{\begin{pmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \end{pmatrix}} = \underset{r}{(0)} \tag{14}$$

```
###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:

$$b_j \pm \sqrt{\hat{\sigma}^2 (X'X)^{-1}_{jj}} \times t_{n-k;1-\alpha/2} \tag{15}$$

The interpretation of the confidence interval is very important: with $\alpha = 0.05$, i.e. the standard 95% confidence interval, the correct interpretation is that if we calculated the confidence interval in this way 100 times, in 95 times of the cases the true parameter would lie within the bounds of the confidence interval.

Incorrect (but often encountered) interpretation: There is a 95% chance that the parameter lies in the (specific, stochastic) confidence interval. It seems very close to true, but it isn't because the population parameter value is fixed. So, it is either in the interval or not. This is subtle but important. Why don't we always use a 99% confidence level? Seems to make sense right? Get the confidence level as high as you can! Well, as the confidence level increases, the margin of error increases. That means the interval is wider. So, it may be that the interval is so large it is useless! For example, what if I said that I am 99% confident that you will score between a 10 and a 100 on your next exam? How useful is that in predicting your performance? The interval is simply too wide. There are some instances where it doesn't matter as much, but that is on a case by case basis.

```
##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
```

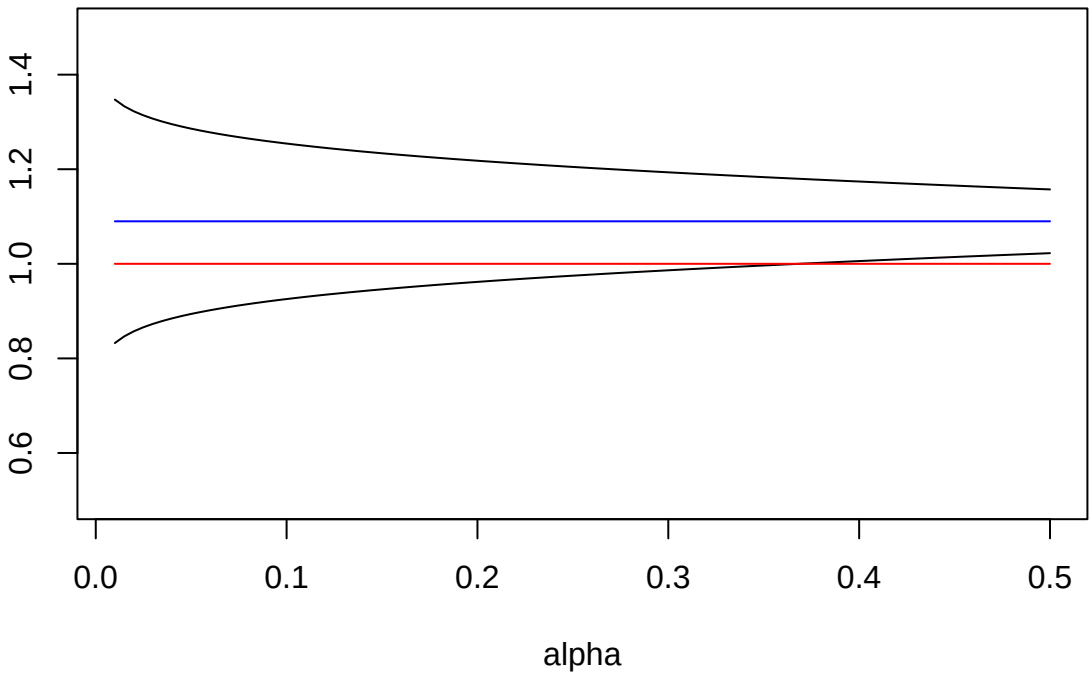
```
##          [,1]
## k1  1.0899015
##    -0.2028447
##     0.4213416
##     0.2257893
```

```
####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
```

```
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
```

Size and Power

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

The Size: The **size** of a hypothesis test is the probability of a (undesired) false rejection of the nullhypothesis (type I error); see Figure 1. By choosing a certain α-level we determine the so-called **nominal size**, which shall control (at least asymptotically) the likelihood of committing a type I error. For compound nullhypotheses, e.g.~$H_0 : \beta_k \leq \bar{\beta}_k$, the test's size is defined to be its highest size on the compound null (worst-case), which will here be attained as $\beta_k = \bar{\beta}_k$.

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.

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.

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 error term ε_i and checking whether the model is appropriate.

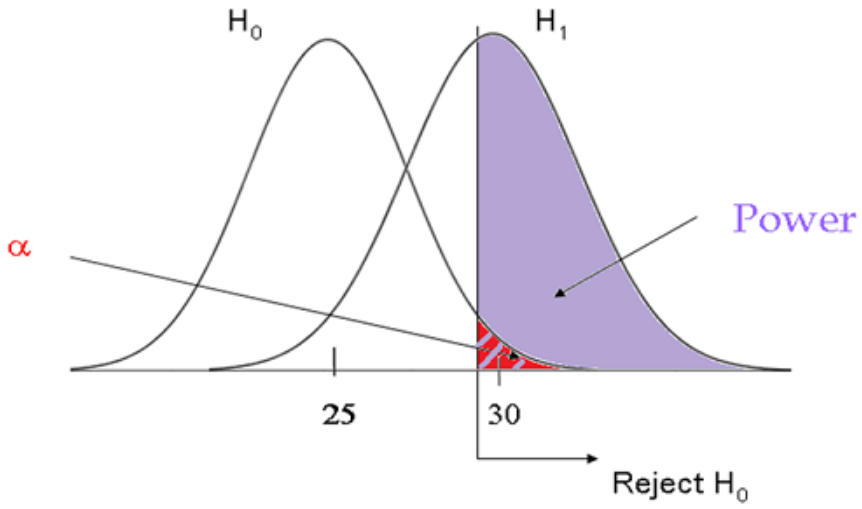


Figure 1: Visualization of size and power of a statistical hypothesis test (fix sample size n).

The Tukey-Anscombe Plot

The Tukey-Anscombe is a graphical tool: we plot the residuals $\hat{\varepsilon}_i$ (one the y-axis) versus the fitted values $\hat{y}_i$ (on the x-axis). A reason to plot against the fitted values is that the sample correlation between $\hat{\varepsilon}_i$ and $\hat{y}_i$ is always zero. In the ideal case, the points in the Tukey-Anscombe plot "fluctuate randomly" around the horizontal line through zero. An often encountered deviation is non-constant variability of the residuals, i.e. heteroscedasticity, an indication that the variance of ε_i increases/decreases with the response variable y_i (see figure 2). If the Tukey-Anscombe plot shows a trend, there is some evidence that the linear model assumption is not correct (the expectation of the error is not zero which indicates a systematic error). In the case that we suspect heteroskedasticity, we should either transform the response variable or perform a weighted regression. If the standard deviation grows linearly with the fitted values, the log transform $y \to log(y)$ stabilizes the variance.

The Normal plot

Assumptions for the distribution of random variables can be graphically checked with the QQ (quantile-quantile) plot. In the special case of checking for the normal distribution, the QQ plot is also referred as the normal plot.

In the linear model application, we plot the empirical quantiles of the residuals (on the y-axis), versus the theoretical quantiles of a $\mathcal{N}(0,1)$ distribution (on the x-axis). If the residuals were normally distributed with expectation μ and variance σ^2, the normal plot would approximate a straight line with intercept μ and slope σ.

Detecting serial correlation

For checking independence of the errors we plot the residuals $\hat{\varepsilon}_i$ versus the observation number i (or if available, the time t_i, of recording the ith observation). If the residuals vary randomly around the zero line, there are no indications for serial correlations among the errors ε_i. On the other hand, if neighbouring (with

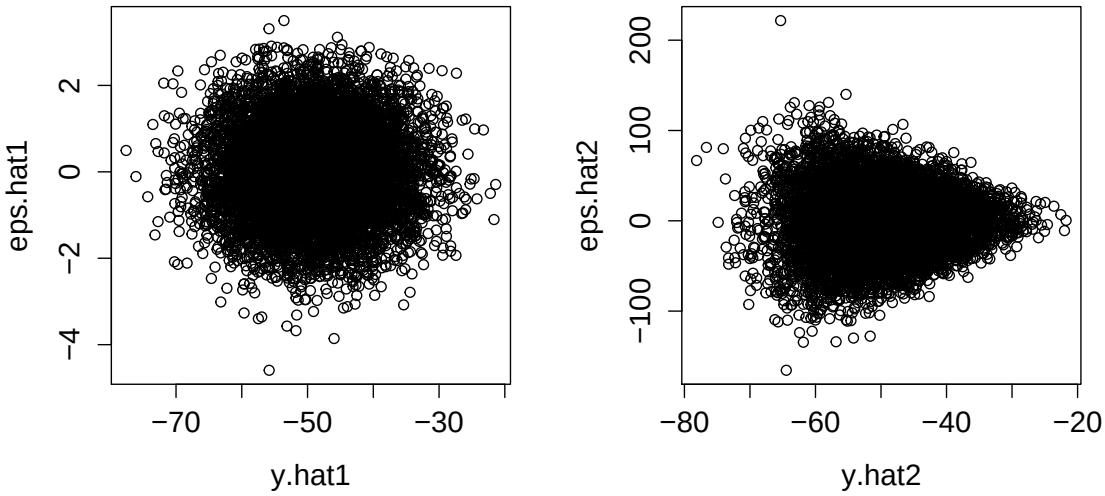


Figure 2: Tukey-Anscombe Plot, plotting error residuals against fitted values $\hat{y}_i$

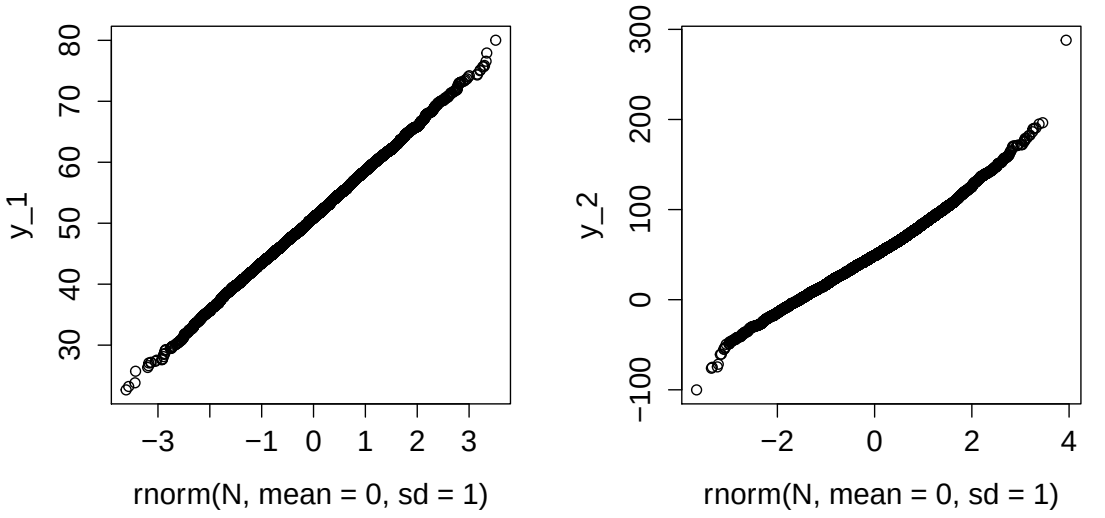


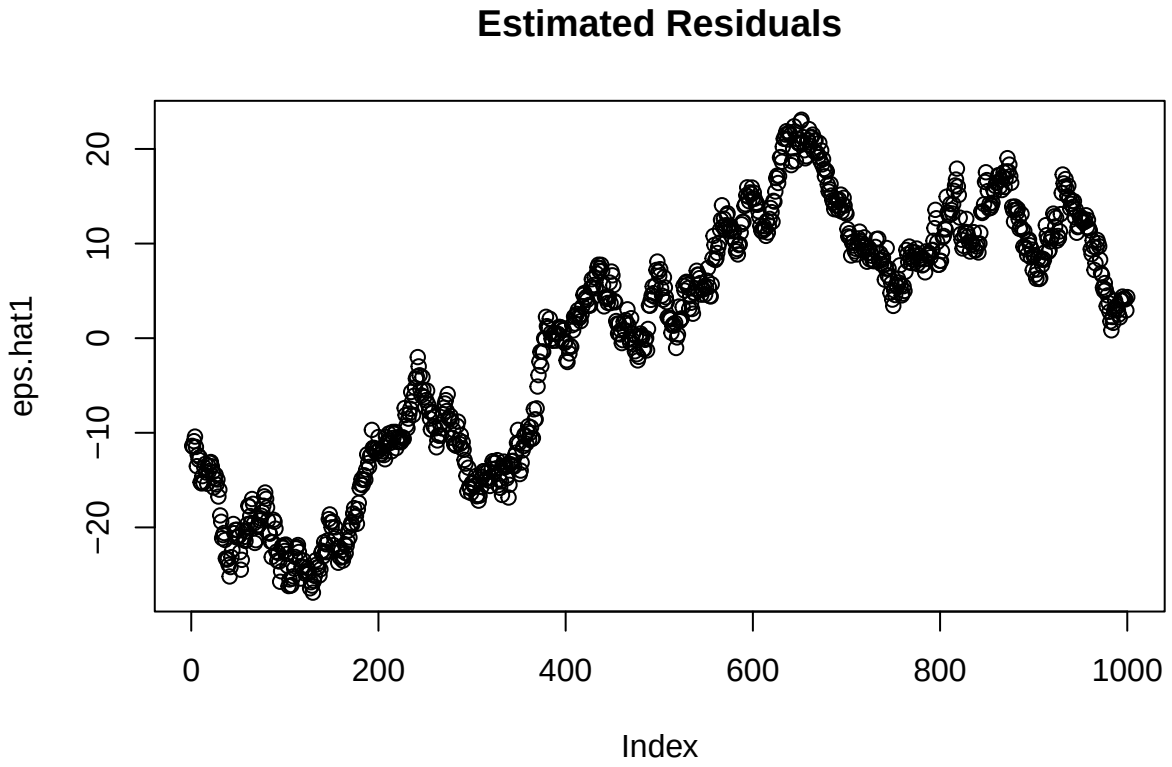
Figure 3: QQ Plot, plotting empirical residuals against the quantiles of the standard normal distribution.

respect to the x-axis) residuals look similar, the independence assumption for the errors seems violated.

```
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 funct
#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
```

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



Nonparametric Density Estimation

Introduction

For a moment we will go back to simple data structures: we have observations which are realizations of univariate random variables

```
$$x_1, x_2, ..., x_n \, i.i.d. \sim F,$$
```

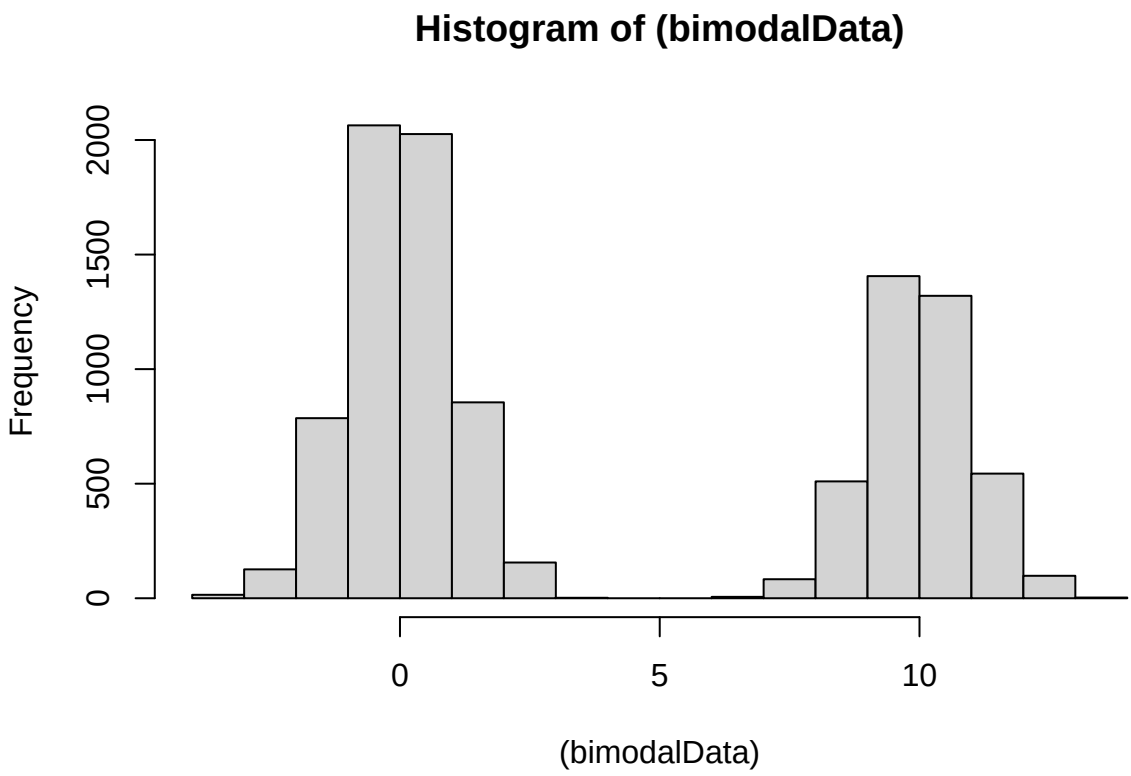
where F is an unkown cumulative distribution function. The goal is to estimate the distribution F. In particular, we are interested in estimating the density $f = F'$, assuming that it exists.

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.

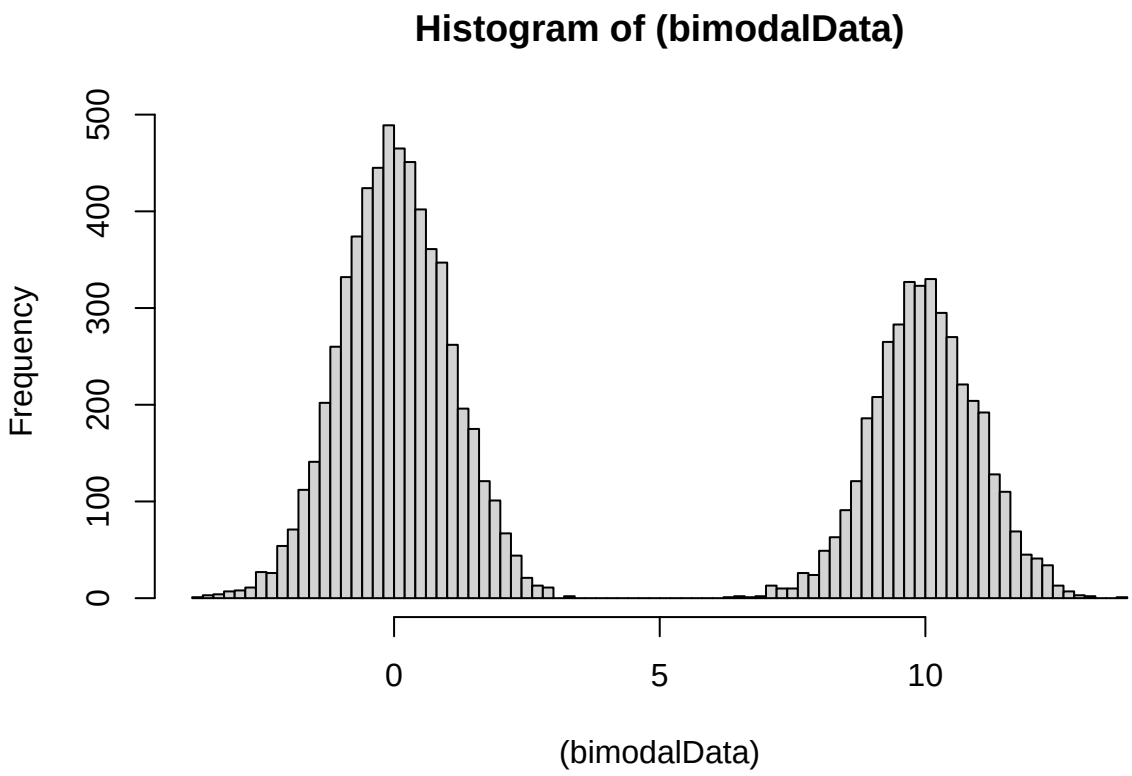
Estimation of a Density

We simulate a bi-modal data set according to:

```
#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 Bernoulli, 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.

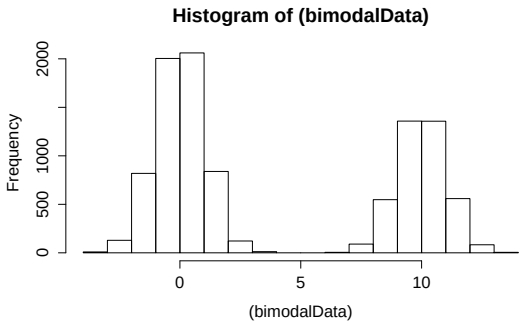
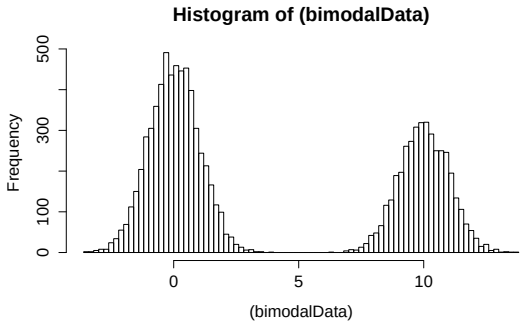


Figure 4: Histogram with two different binwidths H



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

$$f(x) = \lim_{h \to 0} \frac{1}{2h} \mathbb{P}[x - h < X \leq x + h] \tag{16}$$

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

$$\hat{f}(x) = \frac{1}{2hn} \#\{i; X_i \in (x - h < X \leq x + h]\}. \tag{17}$$

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

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) \tag{18}$$

$$K(x) \geq 0, \int_{-\infty}^{\infty} K(x)dx = 1, K(x) = K(-x) \tag{19}$$

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 may 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

$$MISE = \mathbb{E}\left[\int \left(\hat{f}(x) - f(x)\right)^2 dx\right] = \mathbb{E}\left[ISE\right] \tag{20}$$

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:

$$\mathbb{E}\left[\hat{f}(x)\right] = \int \frac{1}{h} K\left(\frac{x - X_i}{h}\right) f(X_i) dX_i \tag{21}$$

$$\tag{22}$$

$$\begin{aligned}
Var(\hat{f}(x)) &= \frac{1}{nh^2} Var\left(K\left(\frac{x - X_i}{h}\right)\right) && (23)\\
&= \frac{1}{nh^2}\mathbb{E}\left[K\left(\frac{x - X_i}{h}\right)^2\right] - \frac{1}{nh^2}\mathbb{E}\left[K\left(\frac{x - X_i}{h}\right)\right]^2 && (24)\\
&= n^{-1}\int \frac{1}{h^2} K\left(\frac{x - X_i}{h}\right)^2 f(X_i) dX_i - n^{-1}\left(\int \frac{1}{h} K\left(\frac{x - X_i}{h}\right) f(X_i) dX_i\right)^2 && (25)
\end{aligned}$$

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

$$\begin{aligned}
Bias(\hat{f}(x)) &= \int \frac{1}{h} K\left(\frac{x - X_i}{h}\right) f(X_i) dX_i - f(x) && (26)\\
&\underbrace{=}_{z=(X_i - x)/h, dz = 1/h, X_i = x + zh} \int K(z) f(x + hz) dz - f(x) && (27)\\
&= \int K(z)(f(x + hz) - f(x)) dz && (28)
\end{aligned}$$

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: $f(x+hz) = f(x) + hzf'(x) + \frac{1}{2}h^2z^2f''(x) + ...$ plugging this into 28 yields

$$\begin{aligned} Bias(\hat{f}(x)) &= \underbrace{\int zK(z)dz}_{=0} + \frac{1}{2}h^2 f''(x) \int z^2 K(z)dz + \dots & (29) \\ &= \frac{1}{2}h^2 f''(x) \int z^2 K(z)dz + \text{ higher order terms in } h. & (30) \end{aligned}$$

For the variance, we get from (25)

$$\begin{aligned} Var(\hat{f}(x)) &= n^{-1} \int \frac{1}{h^2} K\left(\frac{x - X_i}{h}\right)^2 f(X_i)dX_i - n^{-1}(f(x) + Bias(\hat{f}(x)))^2 & (31) \\ &= n^{-1}h^{-1} \int f(x - hz)K(z)^2 dz - \underbrace{n^{-1}(f(x) + Bias(\hat{f}(x)))^2}_{=O(n^{-1})} & (32) \\ &= n^{-1}h^{-1} \int f(x - hz)K(z)^2 dz + O(n^{-1}) = n^{-1}h^{-1}f(x) \int K(z)^2 dz + o(n^{-1}h^{-1}) & (33) \end{aligned}$$

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 $n \to \infty$.

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

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

$$h_{opt}(x) = n^{-1/5} \left(\frac{f(x) \int K^2(z)dz}{(f''(x))^2 (\int z^2 K(z)dz)^2} \right)^{1/5} \quad (34)$$

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*

$$asympt.MISE = \int Bias(x)^2 + Var(\hat{f}(x))dx = \frac{1}{4}h^4 R(f'')\sigma_K^4 + R(K)/(nh) \tag{35}$$

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

$$h_{opt} = n^{-1/5}(R(K)/\sigma_K^4 \times 1/R(f''))^{1/5} \tag{36}$$

By replacing h with h_{opt}, e.g. in 35 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 36, 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 36 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

$$\begin{aligned} ISE(\hat{f}, f) &= \int \left(\hat{f}(x) - f(x)\right)^2 dx & (37)\\ &= \int \hat{f}(x)^2 dx - 2\int \hat{f}(x)f(x)dx + \int f(x)^2 dx & (38) \end{aligned}$$

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

$$ISE^*(\hat{f}, f) = \int \hat{f}(x)^2 dx - 2\int \hat{f}(x)f(x)dx \tag{39}$$

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

$$\hat{f}_{-i}(x) = \frac{1}{(n-1)\,h} \sum_{j=1, j\neq i}^{n} K\left(\frac{x_j - x}{h}\right) \tag{40}$$

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.

$$\hat{f}_{-i}(x) = n^{-1} \sum_{j=1}^{n} \hat{f}_{-j}(x) \tag{41}$$

When we insert $\hat{f}_{-i}(x)$ into 39 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 $n = 100$.

Nonparametric Regression

Introduction

We consider nonparametric estimation with one predictor variable only, mainly for practical reasons as generalizations to two or three variables are not so easy due to the curse of dimensionality. The basic nonparametric regression model is specified as

$$y_i = m(x_i) + \varepsilon_i \tag{42}$$

where $\varepsilon_1, ..., \varepsilon_n$ $i.i.d$ with $\mathbb{E}[\varepsilon_i] = 0$ and $m : \mathbb{R} \to \mathbb{R}$ is an "arbitrary" or completely unspecified function. $m(\cdot)$ is called the nonparametric regression function and it satisfies $m(x) = \mathbb{E}[y|x]$, i.e. it is the conditional expectation of y given x. We will not go into the theory here, but analogously to the kernel density estimator, we will make some smoothness assumptions on $m(x)$ that at least ensure that the first and second derivatives exist. The flexibility that such a function allows us makes us less dependent on assumptions regarding the relationship of the conditional expectation of y given x, but this flexibibility comes with the cost that the estimation accuracy is inferior to that of linear regression.

In this section we will present two different approaches to kernel regression estimation: the local constant, also called the Nadaraya-Watson estimator and the local linear estimator.

The kernel regression estimator I: The local constant

We can view the function in 42 as

$$m(x) = \mathbb{E}[y|X = x]$$

assuming that X is random and $X_i = x_i$ are realized values of the random variables). We can express this conditional expectation as

$$\int_{\mathbb{R}} y f_{Y|X}(y|x)dy = \frac{\int_{\mathbb{R}} y f_{X,Y}(x,y)dy}{f_X(x)}$$

where $f_{Y|X}$, $f_{Y,X}$, f_X denote the conditional, joint and marginal densities. We can now plug in the univariate and bivariate kernel density (all with the same univariate kernel K) estimates

$$\hat{f}_X(x) = \frac{\sum_{i=1}^n K(\frac{x-x_i}{h})}{nh},\ \hat{f}_{X,Y}(x,y) = \frac{\sum_{i=1}^n K\left(\frac{x-x_i}{h}\right)\sum_{i=1}^n K(\frac{y-y_i}{h})}{nh^2}$$

into the formula above which yields the so-called Nadaraya-Watson kernel estimator.

An alternative derivation reveals the origin of the name of the local constant estimator. Instead of deriving the conditional mean explicitly from the nonparametric estimator of the conditional density, we can think of estimating the unknown function $m(x)$ as that which minimizes the weighted squared distance between the function itself and y. This is a weighted average where the weights vary by x. Analagously to how we constructed the OLS estimator, we solve

$$\underset{\alpha,\beta}{argmin} \sum_{i=1}^n (y_i - \alpha - x_i\beta)^2$$

by setting the first-order conditions equal to zero and obtaining the slope and intercept parameter estimates. If now replace $\alpha - x_i\beta$ with $m(x)$ and introduce kernel weights, we have

$$\underset{a}{argmin} \sum_{i=1}^n [y_i - m(x)]^2 K\left(\frac{x-x_i}{h}\right)$$

which has the first order condition

$$-2\sum_{i=1}^n [y_i - a]K\left(\frac{x-x_i}{h}\right)$$

Solving this yields

$$a = \hat{m}(x) = \frac{\sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) y_i}{\sum_{i=1}^n K\left(\frac{x-x_i}{h}\right)}$$

which is identical to the expression above. We essentially regress a constant, locally, on Y to determine our function at a point, hence the name local constant estimator.

The Bias

$$Bias[\hat{m}(x)] \approx \frac{R(f'')}{2f(x)} h^2 B(x) \tag{43}$$

where we say $\approx$ because we ignore some smaller order terms from the second order taylor expansion. $B(x) = 2m'(x)f(x) + m''(x)f(x)$, where $m'(x)$ and $m''(x)$ are the first and second derivatives of the conditional mean w.r.t. x. $R(f'')$ is defined as above. A few things to notice:

- The bias is independent of the sample size, except through the choice of h.
- The bias is equal to zero (i.e. the estimator is unbiased) if we estimate a constant, i.e. $m' = 0$ and $m'' = 0$.

This means that the local constant estimator is biased in finite samples, unless the underlying function is a constant. The bias of a nonparametric estimator depends on the first and second derivatives of the conditional mean in the univariate setting. Further, if the true underlying function is linear, the local constant estimator is biased unless the underlying data follow a uniform distribution.

The kernel regression estimator II: The local linear estimator

The idea behind the local linear estimator is that it is preferable to locally fit a line instead of a constant. The local linear approximation can be viewed as the equivalent of a local Taylor expansion at any point x. That is, for the relationship $y = m(x) + \varepsilon$, we have data $(y_1, x_1), ..., (y_n, x_n)$ and as such we can for each point x_i we can take a linear Taylor approximation for the point x. The idea is to fit the local model

$$y_i = \alpha + \beta'(x_i - x) + \varepsilon_i$$

the reason for using $x_i - x$ instead of x_i is so that the intercept equals $f(x) = \mathbb{E}[y_i|x_i = x]$, i.e. the constant equals the conditional mean. Once we get the estimates $\hat{\alpha}(x), \hat{\beta}(x)$, we then set $\hat{g}(x) = \hat{\alpha}(x)$. Furthermore, we can use $\hat{\beta}(x)$ to estimate $\frac{\partial}{\partial x} g(x)$. If we simply fit a linear regression through observations, we minimize

$$\underset{\alpha,\beta}{argmin} \sum_{n=1}^{n} [y_i - \alpha - \beta(x_i - x)] K\left(\frac{x - x_i}{h}\right)$$

By writing

$$Z_i = \begin{pmatrix} 1 \\ x_i - x \end{pmatrix}$$

We can derive the explicit expression

$$\begin{pmatrix} \hat{\alpha} \\ \hat{\beta} \end{pmatrix} = \left(\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) Z_i Z_i'\right)^{-1} \left(\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) Z_i y_i\right)$$

This is a locally weighted regression of y_i on x_i and algebraically, equals the weighted least squares estimator. In contrast to the NW estimator, the local linear estimator preserves linear data, i.e. the local linear estimator has zero bias when the true regression model is linear. As $h \to \infty$ the ll estimator collapses to the OLS regression of y_i on x_i. In this sense, the local linear estimator is a natural nonparametric generalization of least-squares regression.

The role of the bandwidth

The bandwidth h controls the bias-variance trade-off: a large bandwidth implies a high bias and a low variance, resulting in a slowly varying curve, and vice-versa. Visually, this can be represented by showing the calculations for the function value for one example data point, represented in figure 7.

Quantifying uncertainty: The Bootstrap

The bootstrap, proposed by Efron (1979) is considered one of the greatest innovations in statistics. Essentially, the bootstrap allows us to simulate a (conditional) distribution from an estimated model, in order to make

statistical inference (testing and confidence bands) possible for a broad range of problems where traditional methods fail.

Efron's nonparametric bootstrap

Consider the situation where the data are realizations of

$$Z_1, ..., Z_n \ i.i.d. \sim F,$$

where F denotes an unknown distribution. The variables Z_i can be real valued (in the case of (univariate) density estimation) or vector values, e.g. in a regression framework where $\mathbf{Z}_i = (X_i, Y_i)$ with $X_i \in \mathbb{R}^p$ and $Y_i \in \mathbb{R}$.

We can denote an estimator by

$$\hat{\theta}_n = g(Z_1, ..., Z_n) \tag{44}$$

The estimator $\hat{\theta}_n$ can be either a parametric estimator or a curve estimator (nonparametric estimator). Whenever we want to make statistical inference, we would like to know the probability distribution of $\hat{\theta}_n$. For example, constructing a confidence interval for a true parameter θ requires knowledge about the distribution of $\hat{\theta}_n$, because we require the quantiles of this distribution, the same is true for testing, since we need the quantiles of the distribution of $\hat{\theta}_n$ under the null-hypothesis. Deriving the exact distribution is usually impossible, unless g is simple and F is a mathematically convenient distributions, such as the normal distribution. In the case of the OLS estimator, the distribution of b is inherited from the normality assumption on the error term. If one is not willing to make this assumption, we can sometimes develop asympotic expressions of the distribution. In the case of nonparametric estimators, asymptotic expressions of the bias and variance are available, however:

- these are only asymptotic expressions, finite sample performance is (analytically) unknown.
- more importantly: the variance expression contains the true function in some way, which needs to be estimated and introduces additional uncertainty.

Instead, Efron's idea is this: suppose we knew the distribution F is: we could then simulate to obtain the distribution of any $\hat{\theta}_n$ with arbitrary accuracy (when simulating enough). Because F is unkown, we use the empirical distribution $\hat{F}_n$ which places probability mass $1/n$ on every data point Z_i, $i = 1, ..., n$. The recipe is then to simulate from $\hat{F}_n$, generate simulated data

$$Z_1^*, ..., Z_n^* \ i.i.d. \sim \hat{F}_n.$$

We call such a simulated sample a bootstrap sample. We can now compute our estimator $\hat{\theta}_n^* = g(Z_1^*, ..., Z_n^*)$ based on the bootstrap sample and then repeat this many times to get an approximate distribution.

The bootstrap algorithm

Bootstrapping an estimator can be done as follows.

1. Generate a bootstrap sample
$$Z_1^*, ..., Z_n^* \; i.i.d. \sim \hat{F}_n.$$
This can be realized by doing n uniform drawings with replacement from the data set $Z_1, ..., Z_n$, yielding the bootstrap sample.

2. Compute the bootstrapped estimator
$$\hat{\theta}_n^* = g(Z_1^*, ..., Z_n^*)$$
based on the bootstrapped sample, the function g is the sampe as before.

3. Repeat steps 1 and 2 B times to obtain
$$\hat{\theta}_n^{*1}, ..., \hat{\theta}_n^{*B}$$

4. The B bootstrapped estimators in 3 can then be used as approximations for the bootstrap expectation, the bootstrap variance and the bootstrap quantiles:
$$\mathbb{E}^*[\hat{\theta}^*] \approx \frac{1}{B}\sum_{i=1}^{B} \hat{\theta}^{*i}$$
$$Var^*(\hat{\theta}^*) \approx \frac{1}{B-1}\sum_{i=1}^{B}\left(\hat{\theta}^{*i} - \frac{1}{B}\sum_{j=1}^{B}\hat{\theta}^{*j}\right)$$
$$\alpha - \text{quantile of distribution of} \hat{\theta}^* \approx \text{empirical-quantile of } \hat{\theta}_n^{*1}, ..., \hat{\theta}_n^{*B}$$

The bootstrap confidence interval

Consistent confidence intervals can be constructed from the bootstrapped estimators. A two-sided with coverage $1-\alpha$ for a parameter θ is given by
$$[\hat{\theta}_n - q_{1-\alpha/2}, \hat{\theta}_n - q_{\alpha/2}]$$
where $q_\alpha = \alpha-$ quantile of $\hat{\theta}_n$. The bootstrap estimated confidence interval is the defined as
$$[\hat{\theta}_n - \hat{q}_{1-\alpha/2}, \hat{\theta}_n - \hat{q}_{\alpha/2}]$$
where $\hat{q}_\alpha = \alpha-$ quantile of $\hat{\theta}_n^* - \hat{\theta}_n$. Due to invariance of the quantile:
$$\hat{q}_\alpha = q_\alpha^* - \hat{\theta}_n$$
where $\hat{q}_\alpha^* = \alpha-$ quantile of $\hat{\theta}_n^* - \hat{\theta}_n$.

Therefore, the bootstrap confidence interval becomes
$$[2\hat{\theta}_n - q_{1-\alpha/2}^*, 2\hat{\theta}_n - q_{\alpha/2}^*].$$

Note that this is not the simple bootstrap quantile. The reason for this is, that the bootstrap estimators are

centered around the estimator $\hat{\theta}_n$, not around the true parameter θ.

```
##Example: bootstrap###
####generate a sample of realizations of Z with an unknown distribution###
Z = c(30,37,36,43,42,43,43,46,41,42)
sort(Z)
n = length(Z)
##Calculate the sample mean
Zbar = mean(Z)
###Number of bootstrap draws.
B = 20
###Generate one bootstrap sample
tmpdata = sample(Z,n, replace=TRUE)
sort(tmpdata)
##Sample mean of the bootstrap data
mean(tmpdata)
###Generate B bootstrap samples##
tmpdata = sample(Z,n*B, replace=TRUE)
###Transform into matrix###
bootstrapsample = matrix(tmpdata, nrow=n, ncol=B)
##Calculate consistent confidence intervals
tmp<-sort(colMeans(bootstrapsample-Zbar))
conf_int<-c(Zbar-tmp[18],Zbar-tmp[2])
conf_int

####Why not this?####
tmp2<-sort(colMeans(bootstrapsample))
conf_int2<-c(tmp2[2],tmp2[18])
```

Maximum Likelihood Estimation

Definition

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{argmax}\{f(w_1, ..., w_n; \boldsymbol{\theta})\}$$

i.e. the estimator maximizes the likelihood function.

For computational reasons, we equivalently define

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{argmax}\{log(f(w_1, ..., w_n; \boldsymbol{\theta}))\}$$

In the majority of ML applications the data are assumed to be i.i.d (which we will also do going forward.)

In this case, the above specializes to

$$\begin{aligned}\hat{\boldsymbol{\theta}} &= \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{argmax}\left\{log\left(\prod_{i=1}^{n} f(w_i;\boldsymbol{\theta})\right)\right\} = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{argmax}\left\{\sum_{i=1}^{n} log\, f(w_i;\boldsymbol{\theta})\right\} \\ &= \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{argmax}\left\{\frac{1}{n}\sum_{i=1}^{n} log\, f(w_i;\boldsymbol{\theta})\right\}\end{aligned}$$

The re-scaling by $1/n$ turns the ML-estimator into an M-estimator.

If we let $\mathbf{w}_i = \{\mathbf{x}_i, y_i\}$ (here $\mathbf{x}_i$ subsumes endogenous regressors), the ML estimator can be factorized as

$$f(\mathbf{x}_i, y_i; \boldsymbol{\theta}) = f(y_i|\mathbf{x}_i;\boldsymbol{\theta}) f(\mathbf{x}_i;\boldsymbol{\theta}),$$

In most applications $f(\mathbf{x}_i;\boldsymbol{\theta})$ is constant in $\boldsymbol{\theta}$. In these cases, the problem further simplifies to

$$\hat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}\in\boldsymbol{\Theta}}{argmax}\left\{\frac{1}{n}\sum_{i=1}^{n} log\, f(y_i|\mathbf{x}_i,\boldsymbol{\theta})\right\}$$

which is the most common notation used for the ML estimator in practice.

Without showing the derivation, the variance of the ML-Estimator is given by:

```
$$\sqrt{n}(\hat{\boldsymbol{\theta}}_{ML}-\boldsymbol{\theta}_0)\overset{d}{\rightarrow} N(0, -n\mathbf
```

Where the $\mathbf{I}$ is the Fisher Information matrix, given by the (negative) expectation of the second derivative w.r.t. the parameter vector. $\mathbf{I}(\boldsymbol{\theta}_0) = -\mathbb{E}\frac{\partial^2 log f(\underset{\approx}{\prec};\boldsymbol{\theta}_0)}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}$, where $\mathbf{w} = (\mathbf{w}_1, .., \mathbf{w}_n)$ collects all data.

ML Example: The logit model

As an example of maximum likelihood regression, we will study logistic regression for binary data a little closer. We assume that the random variable Y_i can only take two values, 0 and 1, for example person a person can be unemployed or employed.

$$y_i = \begin{cases} 1 & \text{if the i'th person is employed} \\ 0 & \text{otherwise.} \end{cases}$$

The distribution of Y_i is then the Bernoulli distribution with success probability π_i and the conditional density function of the Bernoulli distribution

$$P(Y_i = y_i) = \pi_i^{y_i}(1-\pi_i)^{1-y_i} \tag{45}$$

with

$$\mathbb{E}(Y_i) = \mu_i = \pi_i \quad \text{and} \quad var(Y_i) = \sigma_i^2 = \pi_i(1-\pi_i).$$

Note that the both mean and variance depend on the probability π_i, which means that anything that alters the probability will not only alter the mean, but also the variance of the observations, indicating that a linear model that allows predictors to affect the mean but assumes the variance to be constant will not be adequate for analyzing this type of data.

The logit transformation

In the above example, we would like to make the probabilities of being employed dependent on certain individual characteristics of a person, e.g. education or experience. The simplest way to do that, is to let π_i be a linear function of the covariates, say

$$\pi_i = \mathbf{x}_i'\boldsymbol{\beta} \tag{46}$$

This model is also sometimes called a linear probability model and can be estimated using OLS. One problem with this model is that π_i is a probability and as such is constrained to lie between 0 and 1, but the linear predictor on the right can take any real value, so the fitted values from the model can predict values that are not in the correct range. One possibility is to transform the probability, and model the transformation as a linear function of the covariates. First, we move from the probability π_i to the odds

$$odds_i = \frac{\pi_i}{1-\pi_i} \tag{47}$$

defined as the ratio of the probability to its complement. The key here is that odds can take any value larger than zero. Second, we take the log odds

$$\eta_i = logit(\pi_i) = log\left(\frac{\pi_i}{1-\pi_i}\right)$$

which removes the floor restriction. As the probability goes to zero, the odds approach zero and the logit approaches $-\infty$. At the other extreme, as π_i approaches 1, the odds approach $+\infty$ and so does the logit. Thus, the logit maps the probabilities from 0,1 to the entire real line. When the odds are exactly 1/2, the odds are even and the logit is 0, while negative logits represent probabilities below one half and positive logits represent probabilities above 1/2. Figure depicts this relationship.

The logistic regression model

Suppose we have n independent observations $y_1, ..., y_n$ and the $i'th$ observation is a realization of a random variable Y_i. We assume a Bernoulli distribution for Y_i, i.e.$Y_i \sim B(\pi_i)$ and that the logit of the underlying probability π_i is a linear function of the predictors

$$logit(\pi_i) = \mathbf{x}_i'\boldsymbol{\beta} \tag{48}$$

where $\mathbf{x}_i$ is the vector of covariates and $\boldsymbol{\beta}$ the corresponding vector of regression coefficients. The interpretation of the estimated coefficients is straightforward, in that β_j represents the change in the logit of the probability associated with a unit change in the j-th covariate holding all other regressors constant. Exponentiating

equation 48 we find the odds for the i'th unit are given by

$$\frac{\pi_i}{1-\pi_i} = exp\{\mathbf{x}_i'\boldsymbol{\beta}\} \tag{49}$$

solving for π_i in the logit model in 48 gives

$$\pi_i = \frac{exp\{\mathbf{x}_i'\boldsymbol{\beta}\}}{1+exp\{\mathbf{x}_i'\boldsymbol{\beta}\}} \tag{50}$$

while the left-hand side is in the familiar probability scale, the right-hand side is a non-linear function of the covariates, and there is no easy way to express the effect of increasing a covariate by one unit on the probability. We can obtain an approximate answer by taking derivatives with respect to x_j and using the quotient rule:

$$\frac{\partial \pi_i}{\partial x_{ij}} = \beta_j \pi_i (1-\pi_i)$$

This means that the effect of the covariate $\mathbf{x}_i j$ depends not only on the coefficient, but also on the baseline probability. This makes intuitive sense: assume that the baseline probability for a particular group of people (e.g. highly educated, 35-40 year old) for being employed is relatively high (95&). Then it would make sense that the effect of an additional year of experience would not increase the employment probability as much as it would for low educated, 20-25 year old workers. This is another reason beside the obvious floor and ceiling problem of why a linear probability model might not be the best modeling choice.

We will talk about how to interpret coefficient estimates in more detail below.

Estimation

The estimation of the logistic regression model for k independent Bernoulli observations is a product of densities given by equation 45. We take logs and find the log-likelihood function

$$log(\mathcal{L}) = \sum y_i \, log(\pi_i) + (1-y_i) \, log\,(1-\pi_i) \tag{51}$$

where π_i depends on the covariates $\mathbf{x}_i$ and a vector of parameters $\boldsymbol{\beta}$ through the logit transformation. The estimation procedure is then to calculate first and second derivatives and plug in starting values. Then we start an iterative procedure to maximize the log-likelihood function, which is typically performed by pre-installed software routines.

The variance estimator is taken from above: The standard errors are calculated as the square root of the diagonal of the negative Hessian matrix (the matrix of second derivatives).

We can also derive the logit model from a latent variable model. Suppose that

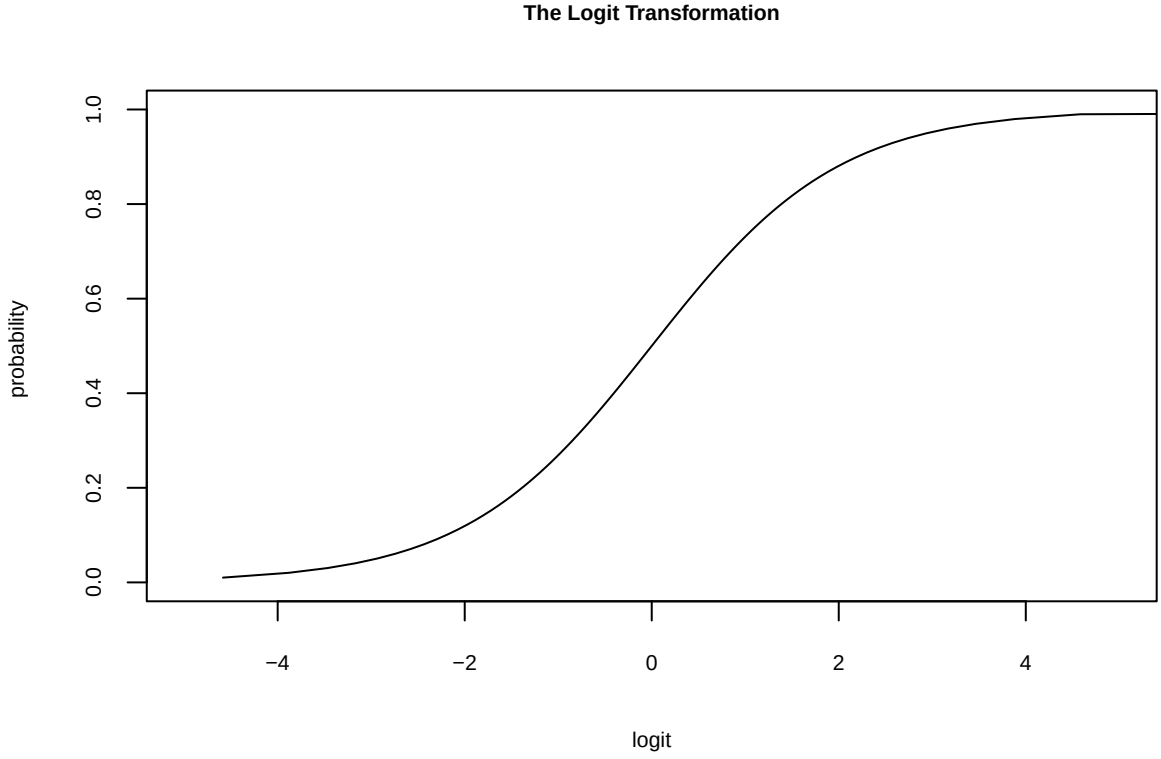
$$y_i^* = \mathbf{x}_i\boldsymbol{\beta} + \varepsilon_i$$

and y_i^* is a hypothetical continuous variable that affects the choice of agents (choosing $y_i = 1 \quad \text{or} \quad y_i = 0$).

$$y_i = \begin{cases} 1 & \text{if } y_i^* > 0 \\ 0 & \text{if } y_i^* \leq 0. \end{cases}$$

In the logit model we assume that the error term ε_i follows a logit distribution.

```
##The logit transformation#

####Generate a vector of probabilities########
pi=seq(0,0.99999999,le=100)
logit<-function(x)
{
  logit<-log(x/(1-x))
  return(logit)
}

plot(logit(pi),pi,type="l", xlim=c(-5,5), main="The Logit Transformation",xlab="logit",ylab="probabilit
```



Maximum Likelihood estimation: Logit

General Syntax

Interpreting logit coefficients

Consider the following model

$$\mathbb{E}(Y|x_1, x_2) = F(\underbrace{\beta_0 + \beta_1 x_1 + \beta_2 x_2}_{\nu}) \tag{52}$$

where here F is the logit transformation and x_1 is a continous variable.

Traditionally:

- Slope coefficients (β) are the rate of change in Y as x_1 changes.
- Now: the slope coefficient β_1 is interpreted as the rate of change in the "log odds" as x_1 changes.

It can be misleading to compare coefficients across models because the variance of the underlying latent variable ($y*$) is not identified and can differ across models. Alternatives

1. Marginal effects: $\frac{\partial \mathbb{E}(Y|x_1,x_2)}{\partial x_1} = \frac{\partial F}{\partial \nu}\frac{\partial \nu}{\partial x_1}$ The marginal effects depend on the values of the independent variables, so, it is often useful to evaluate the marginal effects at the means of the independent variables.
2. Predicted Probabilities: Gives predicted values at substantively meaningful values of $\mathbf{x}_1$.
3. Standardized coefficients. Gives the standard deviation increase in y* given a one unit increase in x_k, holding all other variables constant.
4. Fully standardized coefficients: gives the standard deviation increase in in y*, given a one standard deviation increase in xk, holding all other variables constant.

An interpretation of the logit coefficient which is usually more intuitive (especially for dummy independent variables) is the "odds ratio"– *expB* is the effect of the independent variable on the "odds ratio" [the odds ratio is the probability of the event divided by the probability of the nonevent]. For example, if expB3 =2, then a one unit change in X3 would make the event twice as likely (.67/.33) to occur. Odds ratios equal to 1 mean that there is a 50/50 chance that the event will occur with a small change in the independent variable. Negative coefficients lead to odds ratios less than one: if expB2 =.67, then a one unit change in X2 leads to the event being less likely (.40/.60) to occur. {Odds ratios less than 1 (negative coefficients) tend to be harder to interpret than odds ratios greater than one(positive coefficients).} Note that odds ratios for continuous independent variables tend to be close to one, this does NOT suggest that the coefficients are insignificant. Use the Wald statistic to test for statistical significance.

- the baseline odds-the odds of having a high job for white women without a college degree is 0.32, meaning that within this category, we expect to find 0.32 women with a high job for every woman with a low job.
- The odds ratio for collgrad is 2.47, which means that the odds of having a high job is 2.47 times higher for women with a college degree.
- Here there is also an interaction effect, how to interpret this later.

The interaction effect in nonlinear models

A common mistake: interpreting the first derivative of the multiplicative term between two explanatory variables as the interaction effect.

Recall the model above, with the following amendement:

Consider the following model

$$\mathbb{E}(Y|x_1, x_2) = F(\underbrace{\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2}_{\nu})$$

The problem with this is that we want the interaction effect between two variables (x1 and x2) to represent how much the effect of x1 changes for a unit change in x2.

The marginal effect for x1 then is given by:

$$\frac{\partial \mathbb{E}(Y|x_1, x_2)}{\partial x_1} = \frac{\partial F}{\partial \nu}\frac{\partial \nu}{\partial x_1} = \frac{\partial F}{\partial \nu}(\beta_1 + \beta_{12} x_2)$$

In contrast to a linear model, the marginal effect of an explanatory variable in a nonlinear model is not constant over its entire range, even in the absence of interaction terms. And through $\frac{\partial F}{\partial \nu}$ the marginal effect also depends on the other variables, even without interaction effects[1]

% The effect of x1, in the marginal effects metric, is the first derivative of the expected value of the dependent variable ($E[y]$) with respect to x1, which is an approximation of how much $E[y]$ changes for a unit change in x1. The interaction effect should thus be the cross partial derivative of $E[y]$ with respect to x1 and x2, that is, an approximation of how much the derivative of $E[y]$ with respect to x1 changes for a unit change in x2. In nonlinear models, this is typically different from the first derivative of $E[y]$ with respect to the multiplicative term $x1 \times x2$.

In the practical example from before:

The marginal effect can be approximated by the difference in the expected odds ratios from the baseline category:

- The baseline category for white women is 0.32
- having a degree increases the odds by 0.47 (from 0.32 to 0.79)
- This is the marginal effect, while the marginal effect for black women is 0.35 (from 0.14 to 0.49).
- The interaction effect from before showed that the effect of a college degree is 1.48 times higher for black women, so should not the marginal effect be larger for black women? No, because the baseline odds are much worse for black women.

%the interaction effect means that the effect of a college degree for black women is 1.48 times higher than for white women; however, these refer to very different baseline odds. (the baseline odds for black women to be in a high job are only 0.14). Then the expected increase (i.e. the marginal effect) in odds for white women is 0.47 (from 0.32 to 0.79) while the marginal effect for black women is 0.35 (from 0.14 to 0.49)

[1] for further reading see Karaca-Mandic, Norton Dowd (2012), *Interaction Terms in Nonlinear Models* Health Services Research 47(1).

Multinomial Choice modeling

Suppose instead of a binary response model we want to estimate a model where y_i can take more than two values.

Then the probabilities are given by:

$$\begin{aligned} P(Y=1) &= \frac{e^{\beta_1 \mathbf{X}_i}}{1+\sum_{k=1}^{K-1} e^{\beta_k \mathbf{X}_i}} \\ P(Y=2) &= \frac{e^{\beta_2 \mathbf{X}_i}}{1+\sum_{k=1}^{K-1} e^{\beta_k \mathbf{X}_i}} \\ &\vdots \\ P(Y=K) &= \frac{e^{\beta_K x \mathbf{X}_i}}{1+\sum_{k=1}^{K-1} e^{\beta_k \mathbf{X}_i}} \end{aligned}$$

As you can see above, typically we pick the last category as a baseline and calculate the odds that a member of group i falls in category j as opposed to the baseline as $p_{i1} = p_{ij}$.

In the multinomial logit model we assume that the log-odds of each response follow a linear model:

$$\eta_{ij} = log\left(\frac{p_{ij}}{p_{iJ}}\right) = \alpha_j + \mathbf{x}_i'\beta_j \tag{53}$$

Since the probabilities for the individual outcomes depend on the characteristics of all the other outcomes, we all the probabilities have to sum to one, therefore not all coefficients can be identified. Therefore we only have K-1. This makes it even more clear, that the model depends on the IIA assumption (Independence of irrelevant alternatives), i.e. the odds of a choice j over J should not depend on the choice set for all pairs j,k. A classical example where the multinomial logit model does not work well is the so-called red/blue bus problem.

Suppose you have a choice of transportation between

- a train
- a red bus
- and a blue bus

Suppose half the people take the train and half take the bus and that people who take the bus are indifferent to color, so they distribute themselves equally between the red and the blue buses. If the red bus is discontinued, this should have no effect on the relative odds between taking the blue bus and taking the train, which is clearly not true.

Whether the IIA is violated can be checked via for example a likelihood ratio test, where you exclude some choices and evaluate the likelihood ratio. However, this can be infeasible when there are many choices and there are some simulation results that these tests do not work well in applied work.

Alternatives:

- Mixed Logit Model (computationally intensive, but feasible, correlational structure has to be specified).
- Multinomial independent probit (difficult to compute, has to be simulated.)
- Nested Logit Model: construct "nests" with similar alternatives for example (red and blue bus in one "nest").

Ridge Regression and model selection with the Lasso

Consider the linear regression model

$$y_i = x_i'\beta + \epsilon_i \tag{54}$$

where $\mathbf{x}_i = (x_{i1}, ..., x_{iK})$ with K being the number of regressors. Recall the loss function of the OLS-estimator

$$\hat{\boldsymbol{\beta}}_{OLS} = \underset{\beta}{argmin} \sum_{i=1}^{n} (y_i - \mathbf{x}_i'\beta)^2 \tag{55}$$

is an unconstrained maximization problem. Adding additional covariates to the model has little/no costs even if they are (mostly) uninformative. Additionally, adding groups of highly correlated covariates (mostly a problem when K is large), leads to spuriously insignificant results.

What if we penalize the loss function above with the number and size of the parameters to be estimated in the model? This leads to the idea of Ridge Regression.

Ridge Regression

$$\hat{\boldsymbol{\beta}}_R(s) = \underset{\sum_{j=1}^{K} \beta_j^2 \leq s}{argmin} \sum_{i=1}^{n} (y_i - \mathbf{x}_i'\beta)^2 \tag{56}$$

for any arbitrary s is a constrained maximization problem that can be written as an unocnstrained problem with a penalty:

$$\hat{\boldsymbol{\beta}}(\lambda)_R = \underset{\beta}{argmin} \left\{ \sum_{i=1}^{n} (y_i - \mathbf{x}_i'\beta)^2 + \lambda (\sum_{j=1}^{K} \beta_j^2) \right\} \tag{57}$$

with a one-to-one mapping between s and λ. Setting derivatives equal to zero will result in the normal equations:

$$(\mathbf{X}'\mathbf{X} + \lambda \mathbf{I})\, \hat{\boldsymbol{\beta}}_R^* = \mathbf{X}'\mathbf{y} \tag{58}$$

- $(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})$ non-singular for $\lambda > 0$ large enough so that $(\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})$ has full rank.
- Ridge penalty: $b_j(\lambda) \to 0$ when $\lambda \to \infty$ and also $b_j(\lambda) \to b_{j'}^*$ ("shrinking together") for two different coefficients.
- In general: $\mathbb{E}[\hat{\boldsymbol{\beta}}_R^*] \neq \boldsymbol{\beta}$ if the true model is linear, i.e. there is some bias, but: variances can be smaller, which improves prediction.

This implies that: $\lambda \uparrow$: more bias and less variance. Ridge regression is useful for prediction, but not for model selection, because the probability of selection $\hat{\boldsymbol{\beta}}_R^* = 0$ (i.e. the corner solution) is equal to zero, because of the shape of the constraint region (in the two-dimensional case the constraint region is a circle).

The Lasso

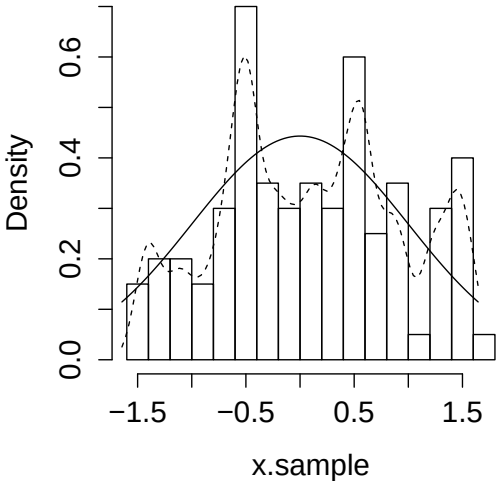
The lasso stands for **L**east **a**bsolute **s**hrinkage and **s**election **o**perator. The main difference between the lasso and ridge regression lies in the shape of the constraint region. The basic idea is to restrict absolute values of coefficients instead of squared values.

$$\hat{\boldsymbol{\beta}}_L(s) = \underset{\beta \in \mathbb{R}^K, \sum_{j=1}^K |\beta_j| \leq s}{argmin} \sum_{i=1}^n (y_i - \mathbf{x}_i'\beta)^2 \tag{59}$$

for any arbitrary s is a constrained maximization problem that can be written as an unoconstrained problem with a penalty:

$$\hat{\boldsymbol{\beta}}(\lambda)_L = \underset{\beta \in \mathbb{R}^K}{argmin} \left\{ \sum_{i=1}^n (y_i - \mathbf{x}_i'\beta)^2 + \lambda (\sum_{j=1}^K |\beta_j|) \right\} \tag{60}$$

There is no analytical solution because the regression contains the absolute value, which is not differentiable. This means that an optimization algorithm is required, for example gradient descent. The solution will choose some b_j to be exactly zero $\rightarrow$ model selection.



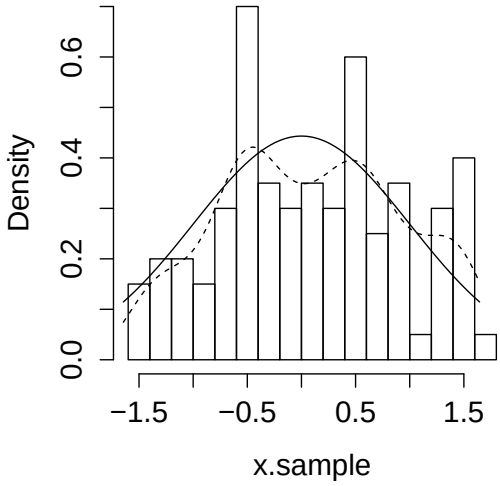
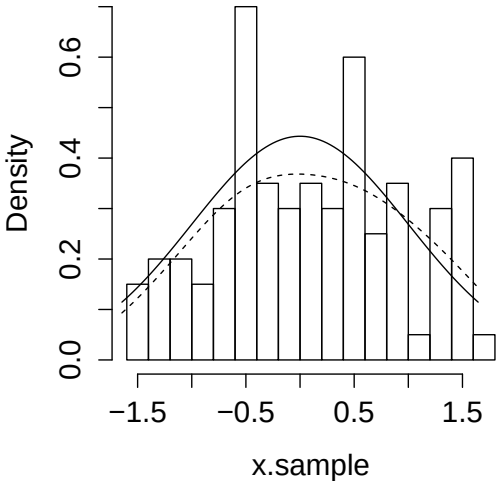
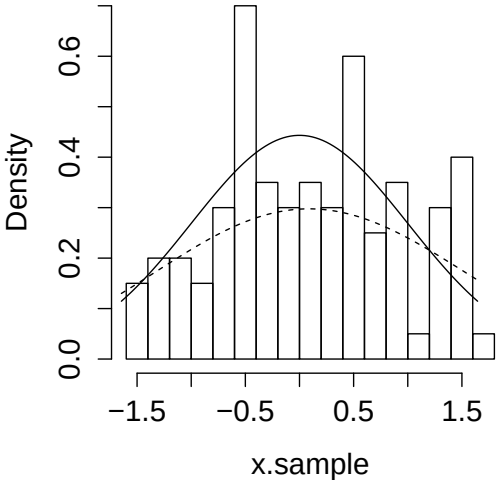
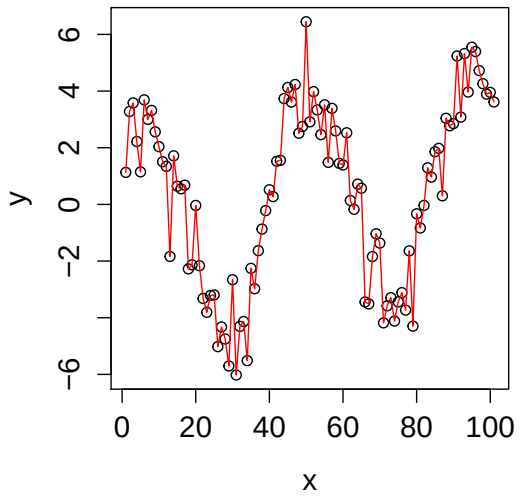


Figure 5: Histogram, true model (solid line) and estimated density with four different bandwidths.







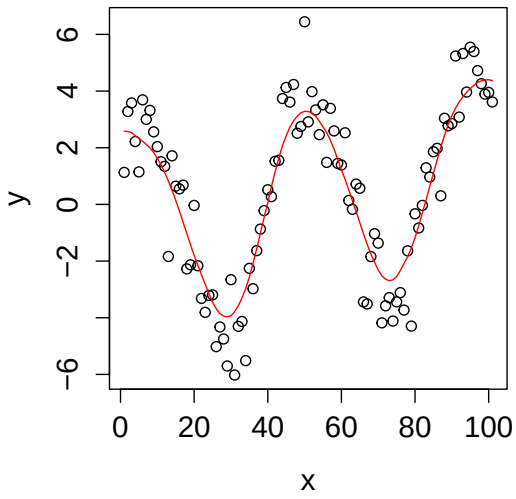
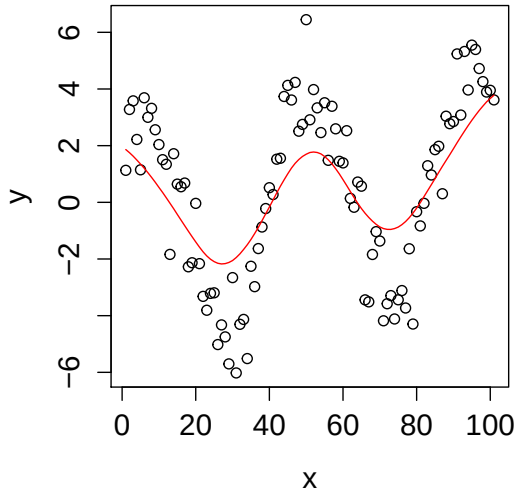
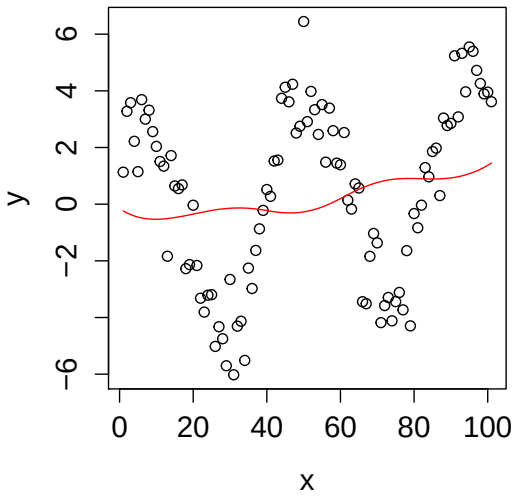
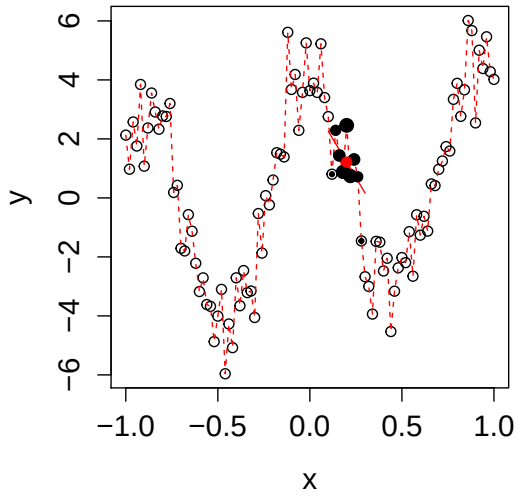


Figure 6: The local constant estimator for the simulated data as given by $y_i = x_i + 4cos(7x_i) + \varepsilon_i$, $\varepsilon_i \sim \mathcal{N}(0, 1)$. The points are the data pairs (x_i, y_i), the red line is the estimated nonparametric function $\hat{m}(x)$ with $h = 0.01, 0.2, 0.4, 0.8$.







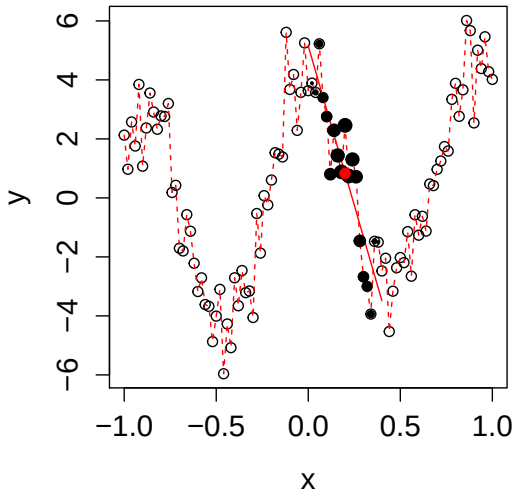
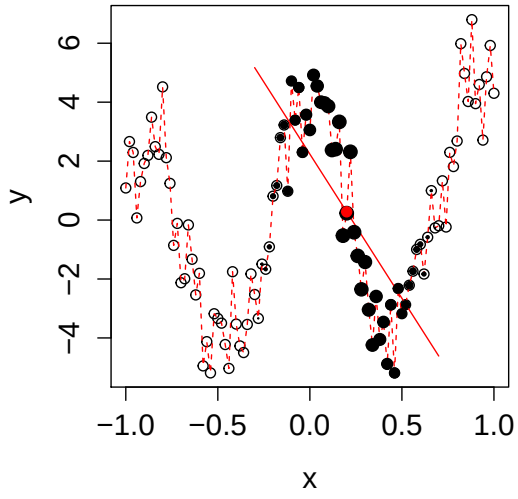
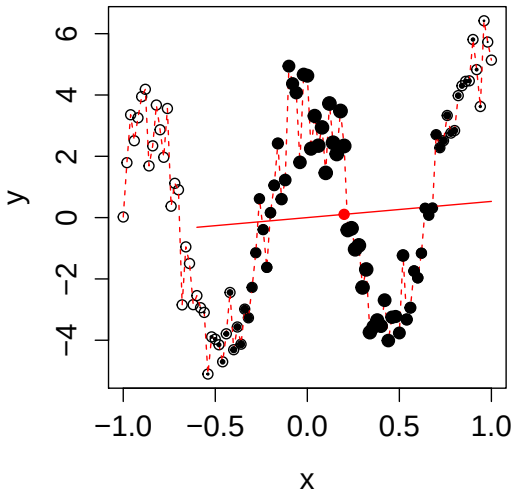
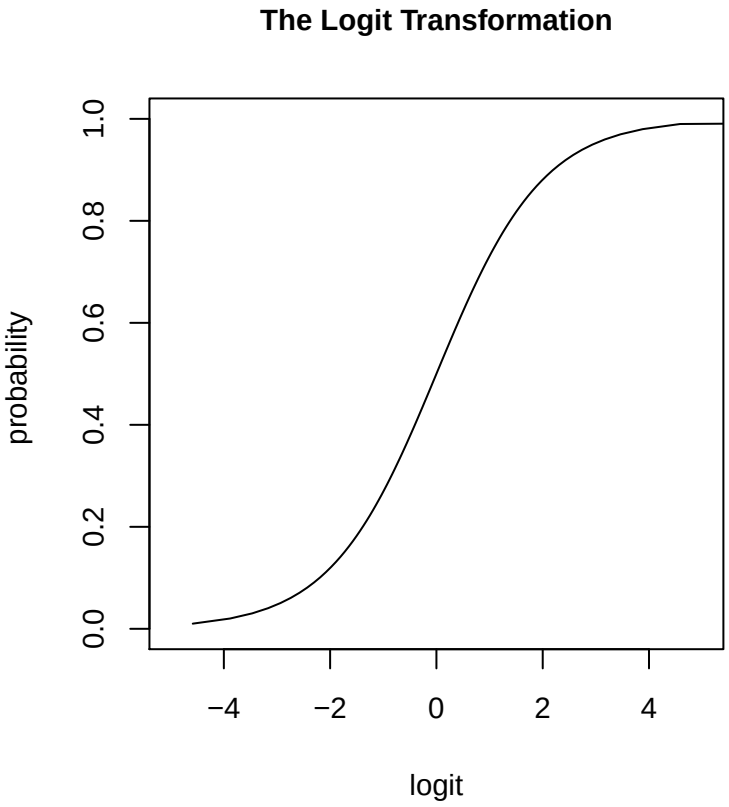


Figure 7: The local linear estimator for one point $x_0 = 0.2$. The points are the data pairs (x_i, y_i), the solid points symbolize the points that are covered by the chosen bandwidth, with $h = 0.1, 0.2, 0.5, 0.8$







Logistic regression — Number of obs = 2211
Wald chi2(4) = 504.62
Log likelihood = -1199.4399 — Prob > chi2 = 0.0000

high_occ	Odds Ratio	Std. Err.	z	P>|z|	[95% Conf.	Interval]
1.black	.4194072	.0655069	-5.56	0.000	.3088072	.5696188
1.collgrad	2.465411	.293568	7.58	0.000	1.952238	3.113478
black# collgrad 1 1	1.479715	.4132536	1.40	0.161	.8559637	2.558003
baseline	.3220524	.0215596	-16.93	0.000	.2824512	.3672059

Figure 8: Stata Output from Buis (2010)

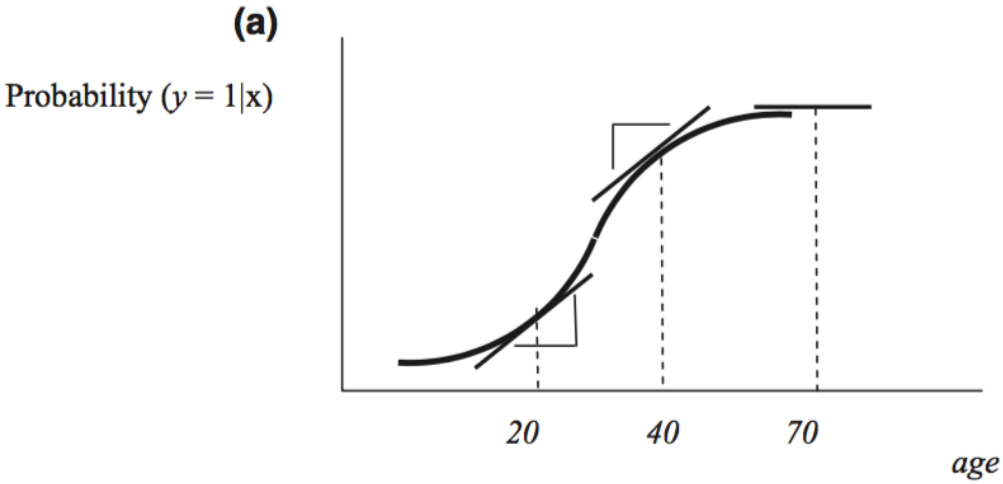


Figure 9: Model with one explanatory variable age, Karaca-Mandic, Norton Dowd (2012)

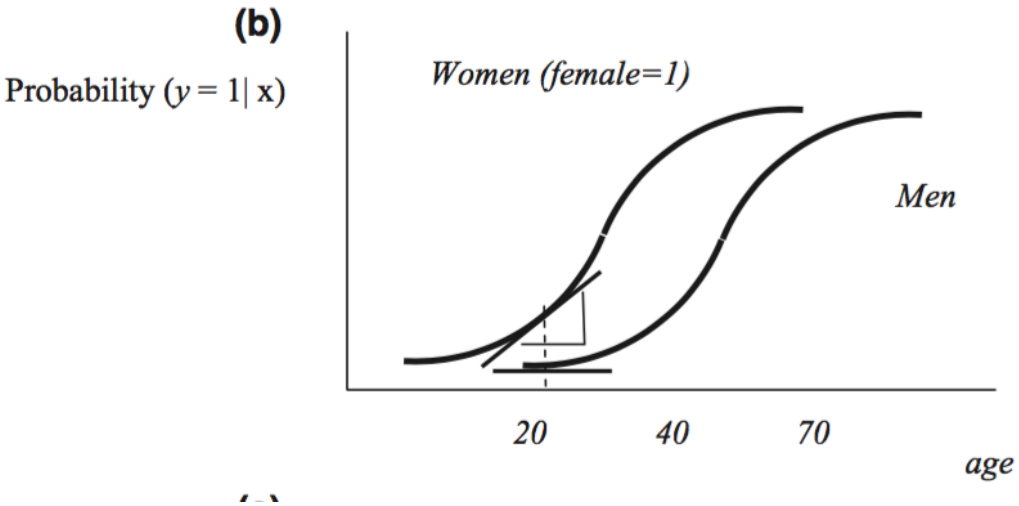


Figure 10: Model with two explanatory variables: age and female, Karaca-Mandic, Norton Dowd (2012)

	Margin	Delta-method Std. Err.
black# collgrad		
0 0	.3220524	.0215596
0 1	.7939914	.078188
1 0	.1350711	.0190606
1 1	.4927536	.1032487

incom 0.black#1.collgrad - 0.black#(

Figure 11: Stata Output from Buis (2010)