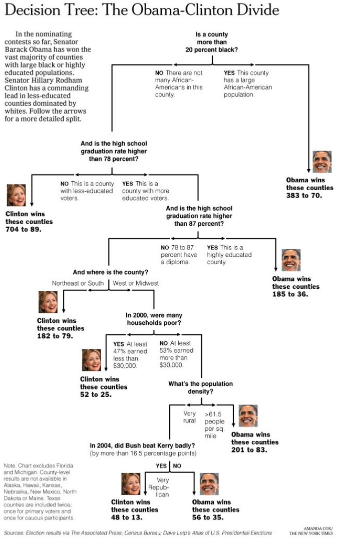
Chapter 6

Generalizations of regression

Regression trees

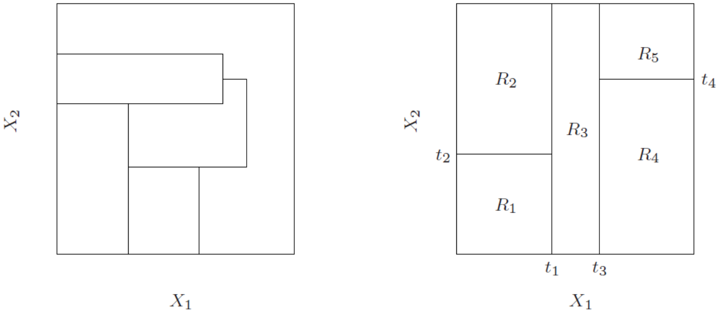


Note: a simple linear regression is too restrictive for large data sets.

Regression trees offer a flexible technique with results, which are easy to interpret

General strategy:

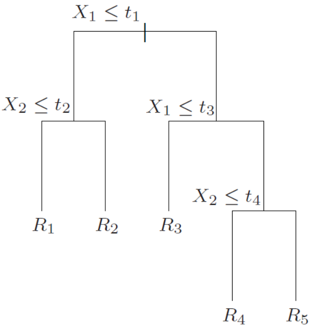
- The values of the explanatory variables are split into P disjunct regions (rectangles) $R_1, \ldots, R_P$: binary splitting

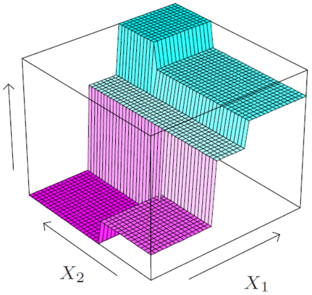


Source: Hastie et al. (2001)

Universität Augsburg

- In each rectangle we fit a simple model
 e.g. a constant, i.e. the forecast in rectangle R_p is the mean of all Y-values falling into this rectangle.





Source: Hastie et al. (2001)

Question: how to determine the regions?

OLS method:

$$\sum_{i=1}^{n}(y_i - \boldsymbol{x}_i'\boldsymbol{b})^2 \quad \longrightarrow min, \quad bzgl. \quad \boldsymbol{b}.$$

For regression trees:

$$\sum_{p=1}^{P}\sum_{i\in R_p}(y_i - \hat{y}_{R_p})^2 \quad \longrightarrow min, \quad w.r.t. \quad R_1, \dots, R_p,$$

where $\hat{y}_{R_p}$ is the mean of observations in the p-th rectangle.

Note: direct optimization is hardly possible $\rightsquigarrow$ recursive binary splitting

Step 1

- Find the variable X_j and the splitting point s, which separates the space into two regions:

$$R_1(j,s) = \{\boldsymbol{X} | X_j \leq s\} \quad \text{and} \quad R_2(j,s) = \{\boldsymbol{X} | X_j > s\}.$$

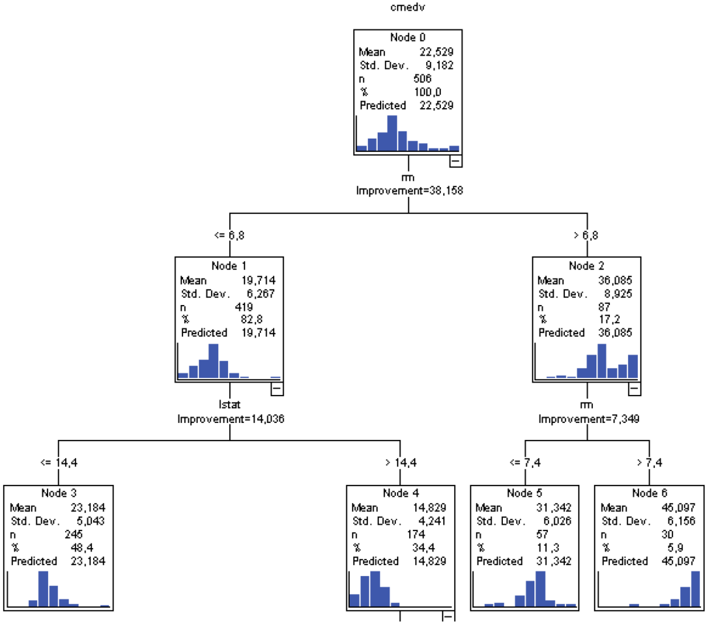
- j and s are determined using the following objective function

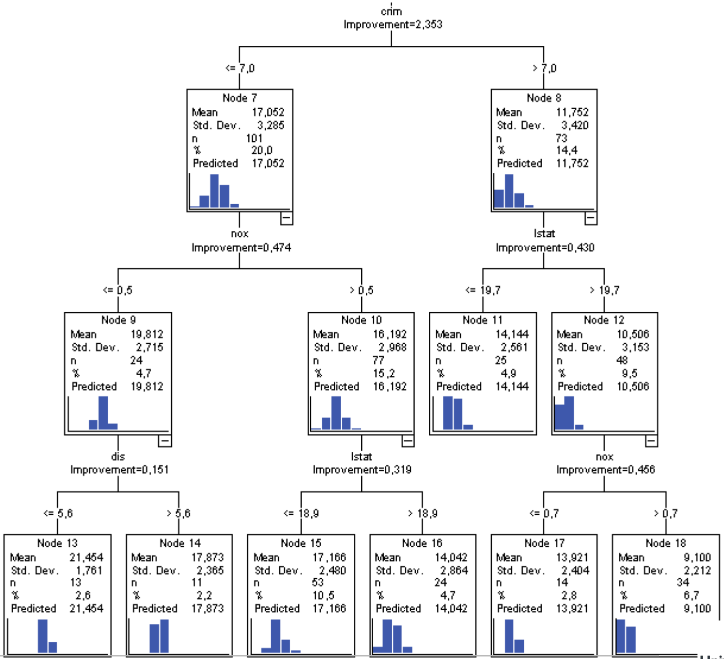
$$\sum_{i:\boldsymbol{x}_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i:\boldsymbol{x}_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2,$$

  where $\hat{y}_{R_1}$ and $\hat{y}_{R_2}$ are averages in R_1 and R_2.

Step 2

Repeat Step 1 to split regions R_1 and R_2 recursively.

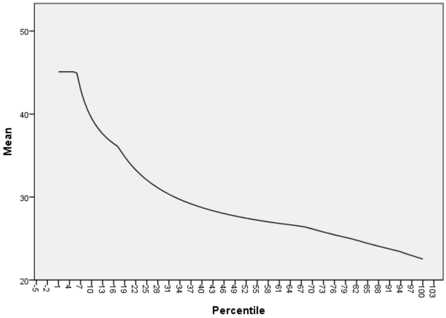

cmedv
Node 0
Mean 22,529
Std. Dev. 9,182
n 506
% 100,0
Predicted 22,529
rm
Improvement=38,158
<= 6,8
> 6,8
Node 1
Mean 19,714
Std. Dev. 6,267
n 419
% 82,8
Predicted 19,714
Node 2
Mean 36,085
Std. Dev. 8,925
n 87
% 17,2
Predicted 36,085
lstat
Improvement=14,036
rm
Improvement=7,349
<= 14,4
> 14,4
<= 7,4
> 7,4
Node 3
Mean 23,184
Std. Dev. 5,043
n 245
% 48,4
Predicted 23,184
Node 4
Mean 14,829
Std. Dev. 4,241
n 174
% 34,4
Predicted 14,829
Node 5
Mean 31,342
Std. Dev. 6,026
n 57
% 11,3
Predicted 31,342
Node 6
Mean 45,097
Std. Dev. 6,156
n 30
% 5,9
Predicted 45,097



Universität Augsburg

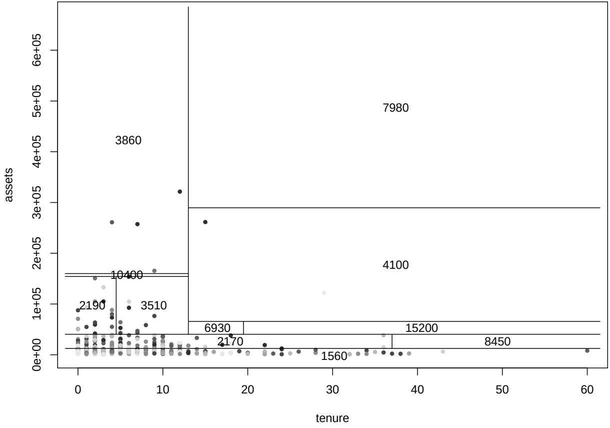
Gain Summary for Nodes

Node	Node-by-Node			Cumulative		
	N	Percent	Mean	N	Percent	Mean
6	30	5,9%	45,10	30	5,9%	45,10
5	57	11,3%	31,34	87	17,2%	36,09
3	245	48,4%	23,18	332	65,6%	26,56
13	13	2,6%	21,45	345	68,2%	26,37
14	11	2,2%	17,87	356	70,4%	26,11
15	53	10,5%	17,17	409	80,8%	24,95
11	25	4,9%	14,14	434	85,8%	24,33
16	24	4,7%	14,04	458	90,5%	23,79
17	14	2,8%	13,92	472	93,3%	23,50
18	34	6,7%	9,10	506	100,0%	22,53

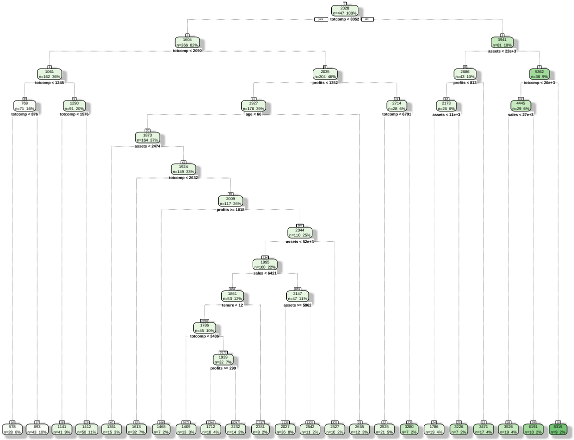


Universität Augsburg

```
> library("tree")
> tree.ceo = tree(salary ~ tenure + assets, data=ceo)
> plot(ceo$tenure,ceo$assets, type="p", pch=20, xlab="tenure", ylab="assets")
> partition.tree(tree.ceo, ordvars=c("tenure","assets"), add=TRUE)
```



```
> library("rpart")
> rpart.ceo = rpart(salary ~ .,data=ceo,control=rpart.control(cp = 0.001))
> fancyRpartPlot(rpart.ceo)
```



Rattle 2017−Nov−16 09:30:27 okhrinya

Note: Using CART we can grow the tree to saturation.

- Fix the maximal number of splittings and a lower bound for the number of observations per region.
- Fix the minimal change in the objective function.
- tree prunning: after the optimal tree is found, it is shortened

$$R_\alpha(T) = \frac{1}{\sum_i (y_i - \bar{y})^2} \sum_{m=1}^{|T|} \sum_{i:\boldsymbol{x}_i \in R_m} (y_i - \hat{y}_{R_m})^2 + \alpha |T|$$

where $|T|$ is the number of terminal nodes in a tree and α is the complexity parameter.

Key properties of CARTs

- For given α it is possible to determine the tree $T(\alpha)$ with the smallest $R_\alpha(T)$ uniquely
- If $\alpha > \beta$ then $T(\alpha) = T(\beta)$ or $T(\alpha)$ is a strict subtree of $T(\beta)$.

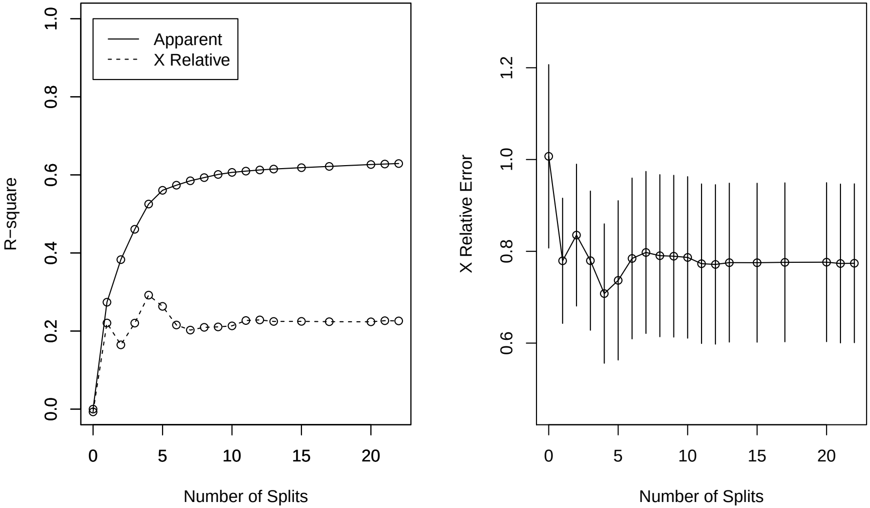
```
> printcp(rpart.ceo)
> printcp(rpart.ceo)

Regression tree:
rpart(formula = salary ~ ., data = ceo, control = rpart.control(cp = 0.001, xval = 10))
Variables actually used in tree construction:
[1] age     assets  profits sales   tenure  totcomp
Root node error: 1323386794/447 = 2960597

n= 447

          CP nsplit rel error  xerror    xstd
1  0.2738266      0   1.00000 1.00703 0.19979
2  0.1091070      1   0.72617 0.77938 0.13638
3  0.0777412      2   0.61707 0.83535 0.15451
4  0.0646524      3   0.53933 0.77968 0.15159
5  0.0351651      4   0.47467 0.70797 0.15182
6  0.0130789      5   0.43951 0.73676 0.17353
7  0.0113130      6   0.42643 0.78445 0.17517
8  0.0081763      7   0.41512 0.79755 0.17644
9  0.0080167      8   0.40694 0.79045 0.17654
10 0.0052976      9   0.39892 0.78938 0.17620
11 0.0032733     10   0.39363 0.78670 0.17590
12 0.0029769     11   0.39035 0.77301 0.17377
13 0.0022593     12   0.38737 0.77146 0.17367
14 0.0017931     13   0.38512 0.77539 0.17310
15 0.0016171     15   0.38153 0.77520 0.17313
16 0.0016099     17   0.37829 0.77603 0.17313
17 0.0012678     20   0.37346 0.77635 0.17313
18 0.0012449     21   0.37220 0.77348 0.17303
19 0.0010000     22   0.37095 0.77403 0.17301
```

Q: How to choose the overall optimal α or subtree? $\rightsquigarrow$ cross-validation

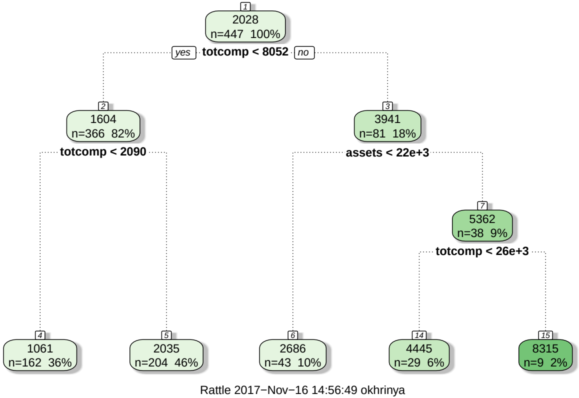
- The sequence of trees T_0 (no splits) to T_m (m splits) uniquely determines the sequence of possible α's

$$\infty, \alpha_1, \ldots, \alpha_{m-1}, \alpha_{min}$$

- Any α between $(\alpha_i, \alpha_{i+1}]$ leads to the same optimal subtree
- Define $\beta_i = \sqrt{\alpha_i \alpha_{i+1}}$ as an “average” CP for every interval
- Split the data into B subsets $G_1, \ldots, G_B$ (10 by default)
  - For every subset excluding the G_i's determine $T_{\beta_1}, \ldots T_{\beta_m}$
  - Compute the relative MSE as the forecast loss for elements in G_i
- Compute the average loss over all G_i's and choose β (and thus the optimal subtree) which corresponds to the smallest one.


Apparent
X Relative
R−square
Number of Splits
X Relative Error
Number of Splits
1.0
0.8
0.6
0.4
0.2
0.0
1.2
1.0
0.8
0.6
0
5
10
15
20

```
> cp.min = which.min(rpart.ceo$cptable[,4]);
> rpart.ceo.prune=prune(rpart.ceo, cp=rpart.ceo$cptable[cp.min,1])
> rpart.ceo.prune$variable.importance/sum(rpart.ceo.prune$variable.importance)
   totcomp     assets      sales    profits     tenure        age
0.52984007 0.18398873 0.10229360 0.09962347 0.05557637 0.02867777
```

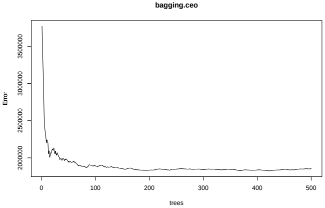


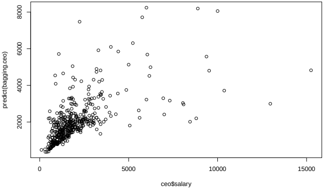
Generalizations

- Bagging: if you use for CART just a subsample, then you obtain a completely different tree.
  - Fit a CART to B random subsamples (*bootstrap*).
  - The error is measured on the remaining observations out-of-bag.
  - The final forecast is:

$$\hat{f}_{avr}(\boldsymbol{x}_0) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^b(\boldsymbol{x}_0).$$

```
> bagging.ceo= randomForest(salary ~ ., data=ceo, mtry=6)
> predict(bagging.ceo);
> cor(ceo$salary,predict(bagging.ceo))
[1] 0.6180269
```





Random Forests: is a generalization of *Bagging*

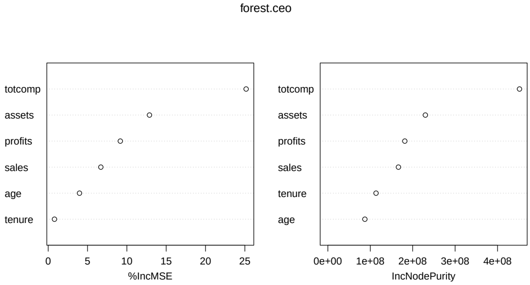
- For each splitting you consider not all explanatory variables bus just a subset of size $M \approx \sqrt{J}$
- ... this makes the trees more heterogenous and “uncorrelated” in terms of forecasts
- Each tree is grown on a bootstrap sample (as for bagging)
- The importance of a variable is measured by increase in (a) MSE ; (b) in node impurity over the out-of-bag sample if the variable is permuted

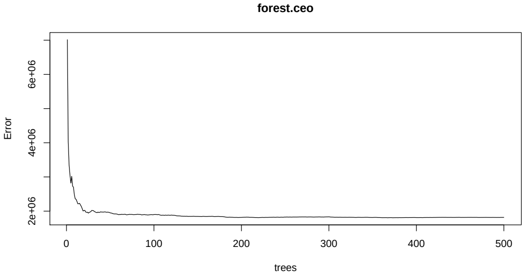
$$\Delta MSE_{j,b} =$$

$$\frac{1}{|\overline{\mathcal{B}}_b|}\sum_{k\in\overline{\mathcal{B}}_b}\hat{u}^2(x_{1k},\ldots,x_{Jk}) - \frac{1}{|\overline{\mathcal{B}}_b|}\sum_{k\in\overline{\mathcal{B}}_b}\tilde{\hat{u}}_k^2(x_{1k},...,x_{j-1,k},\tilde{x}_{jk},x_{j+1,k},\ldots,x_{Jk}),$$

where $\tilde{\boldsymbol{x}}_j$ are the randomly permuted (reordered) observations on the jth variable and $\overline{\mathcal{B}}_b$ is the bth out-of-bag subsample.

```
> forest.ceo= randomForest(salary ~ ., data=ceo, importance=T)
> varImpPlot(forest.ceo)
```

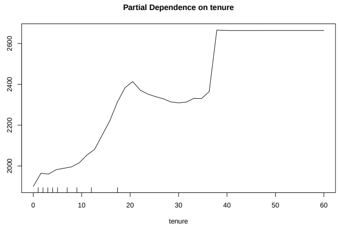

forest.ceo
totcomp
assets
profits
sales
age
tenure
0
5
10
15
20
25
%IncMSE
totcomp
assets
profits
sales
tenure
age
0e+00
1e+08
2e+08
3e+08
4e+08
IncNodePurity


forest.ceo
Error
2e+06
4e+06
6e+06
0
100
200
300
400
500
trees

Partial dependence plots: visualize the marginal impact of a variable/feature

$$\tilde{f}_j(x) = \frac{1}{K}\sum_{k=1}^{K}\hat{f}(x_{1k},...,x_{j-1,k},x,x_{j+1,k},\ldots,x_{Jk})$$

```
> partialPlot(forest.ceo, pred.data=ceo, x.var=tenure)
```



CHAID

- An alternative approach is CHAID (Chi-square Automatic Interaction Detectors): allows not only for binary splitting and is similar to ANOVA.
- Analysis is a generalization of two-sample test for the mean.
- Idea: let G be the number of splittings for variable X. We test if there is a significant difference between the means of Y in different regions.

Total sum of squares

$$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2 = \sum_{g=1}^{G} \sum_{i:\boldsymbol{x}_i \in R_g} (y_i - \bar{y})^2$$

Within sum of squares

$$WSS = \sum_{g=1}^{G} \sum_{i:\boldsymbol{x}_i \in R_g} (y_i - \bar{y}_{R_g})^2$$

Between sum of squares

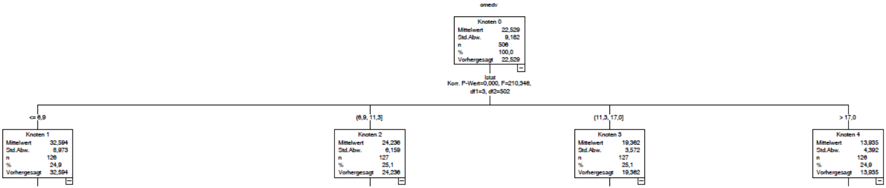
$$BSS = TSS - WSS = \sum_{g=1}^{G} |R_g| (\bar{y}_{R_g} - \bar{y})^2.$$

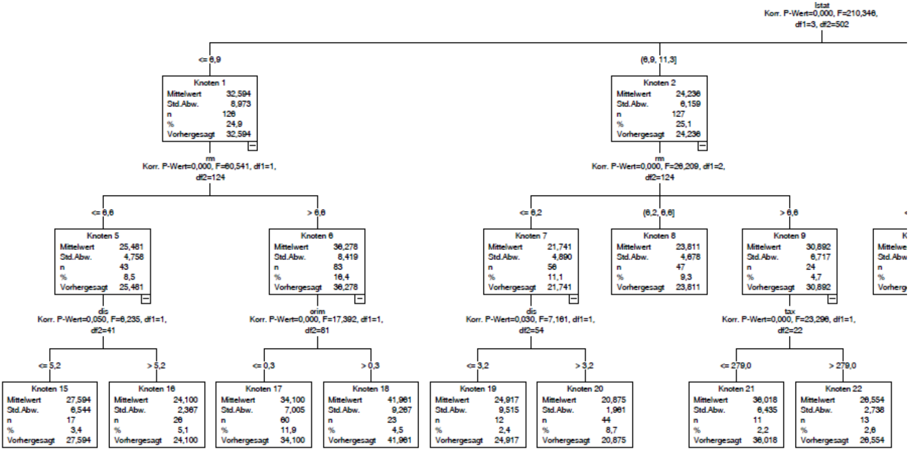
$$H_0 : \mu_1 = \cdots = \mu_G \quad \text{vs} \quad H_1 : \mu_i \neq \mu_j \text{ for at least one pair } i, j$$

$$\text{Test statistic:} \quad F = \frac{BSS/(G-1)}{WSS/(n-G)} \sim F_{G-1,n-G}$$

Idea:

- For each predictor we determine the optimal splitting, i.e. the regions with the smallest p-value of the test.
- The p-values should be corrected due to multiple testing (Bonferroni correction).
- The predictor with the smallest corrected p-value is used for splitting.




lstat
Korr. P-Wert=0,000, F=210,346, df1=3, df2=502
<= 6,9
(6,9, 11,3]
Knoten 1
Mittelwert 32,594
Std.Abw. 8,973
n 126
% 24,9
Vorhergesagt 32,594
Knoten 2
Mittelwert 24,236
Std.Abw. 6,159
n 127
% 25,1
Vorhergesagt 24,236
rm
Korr. P-Wert=0,000, F=60,541, df1=1, df2=124
rm
Korr. P-Wert=0,000, F=26,209, df1=2, df2=124
<= 6,6
> 6,6
<= 6,2
(6,2, 6,6]
> 6,6
Knoten 5
Mittelwert 25,481
Std.Abw. 4,758
n 43
% 8,5
Vorhergesagt 25,481
Knoten 6
Mittelwert 36,278
Std.Abw. 8,419
n 83
% 16,4
Vorhergesagt 36,278
Knoten 7
Mittelwert 21,741
Std.Abw. 4,890
n 56
% 11,1
Vorhergesagt 21,741
Knoten 8
Mittelwert 23,811
Std.Abw. 4,678
n 47
% 9,3
Vorhergesagt 23,811
Knoten 9
Mittelwert 30,892
Std.Abw. 6,717
n 24
% 4,7
Vorhergesagt 30,892
dis
Korr. P-Wert=0,050, F=6,235, df1=1, df2=41
crim
Korr. P-Wert=0,000, F=17,392, df1=1, df2=81
dis
Korr. P-Wert=0,030, F=7,161, df1=1, df2=54
tax
Korr. P-Wert=0,000, F=23,296, df1=1, df2=22
<= 5,2
> 5,2
<= 0,3
> 0,3
<= 3,2
> 3,2
<= 279,0
> 279,0
Knoten 15
Mittelwert 27,594
Std.Abw. 6,544
n 17
% 3,4
Vorhergesagt 27,594
Knoten 16
Mittelwert 24,100
Std.Abw. 2,367
n 26
% 5,1
Vorhergesagt 24,100
Knoten 17
Mittelwert 34,100
Std.Abw. 7,005
n 60
% 11,9
Vorhergesagt 34,100
Knoten 18
Mittelwert 41,961
Std.Abw. 9,267
n 23
% 4,5
Vorhergesagt 41,961
Knoten 19
Mittelwert 24,917
Std.Abw. 9,515
n 12
% 2,4
Vorhergesagt 24,917
Knoten 20
Mittelwert 20,875
Std.Abw. 1,961
n 44
% 8,7
Vorhergesagt 20,875
Knoten 21
Mittelwert 36,018
Std.Abw. 6,435
n 11
% 2,2
Vorhergesagt 36,018
Knoten 22
Mittelwert 26,554
Std.Abw. 2,738
n 13
% 2,6
Vorhergesagt 26,554

Baumtabelle

Knoten	Mittelwert	Standardabweichung	N	Prozent	Vorhergesager Mittelwert	Übergeordneter Knoten	Primäre unabhängige Variable					
							Variable	Sig.[a]	F	df1	df2	Werte aufteilen
0	22,53	9,182	506	100,0%	22,53							
1	32,59	8,973	126	24,9%	32,59	0	lstat	,000	210,346	3	502	<= 6,9
2	24,24	6,159	127	25,1%	24,24	0	lstat	,000	210,346	3	502	(6,9, 11,3]
3	19,36	3,572	127	25,1%	19,36	0	lstat	,000	210,346	3	502	(11,3, 17,0]
4	13,93	4,392	126	24,9%	13,93	0	lstat	,000	210,346	3	502	> 17,0
5	25,48	4,758	43	8,5%	25,48	1	rm	,000	60,541	1	124	<= 6,6
6	36,28	8,419	83	16,4%	36,28	1	rm	,000	60,541	1	124	> 6,6
7	21,74	4,890	56	11,1%	21,74	2	rm	,000	26,209	2	124	<= 6,2
8	23,81	4,678	47	9,3%	23,81	2	rm	,000	26,209	2	124	(6,2, 6,6]
9	30,89	6,717	24	4,7%	30,89	2	rm	,000	26,209	2	124	> 6,6
10	22,59	3,393	17	3,4%	22,59	3	tax	,000	18,192	1	125	<= 279,0
11	18,86	3,345	110	21,7%	18,86	3	tax	,000	18,192	1	125	> 279,0
12	19,31	2,881	14	2,8%	19,31	4	nox	,000	36,153	2	123	<= ,5
13	15,72	3,623	44	8,7%	15,72	4	nox	,000	36,153	2	123	(,5, ,6]
14	11,67	3,554	68	13,4%	11,67	4	nox	,000	36,153	2	123	> ,6
15	27,59	6,544	17	3,4%	27,59	5	dis	,050	6,235	1	41	<= 5,2
16	24,10	2,367	26	5,1%	24,10	5	dis	,050	6,235	1	41	> 5,2
17	34,10	7,005	60	11,9%	34,10	6	crim	,000	17,392	1	81	<= ,3
18	41,96	9,267	23	4,5%	41,96	6	crim	,000	17,392	1	81	> ,3
19	24,92	9,515	12	2,4%	24,92	7	dis	,030	7,161	1	54	<= 3,2
20	20,88	1,961	44	8,7%	20,88	7	dis	,030	7,161	1	54	> 3,2
21	36,02	6,435	11	2,2%	36,02	9	tax	,000	23,296	1	22	<= 279,0
22	26,55	2,738	13	2,6%	26,55	9	tax	,000	23,296	1	22	> 279,0
23	19,51	2,847	75	14,8%	19,51	11	nox	,007	9,596	1	108	<= ,6
24	17,47	3,911	35	6,9%	17,47	11	nox	,007	9,596	1	108	> ,6
25	17,24	3,698	24	4,7%	17,24	13	rad	,022	11,566	1	42	2,0; 5,0; 6,0; 24
26	13,90	2,596	20	4,0%	13,90	13	rad	,022	11,566	1	42	4,0
27	10,85	3,087	56	11,1%	10,85	14	dis	,000	22,769	1	66	<= 2,1
28	15,53	3,094	12	2,4%	15,53	14	dis	,000	22,769	1	66	> 2,1

Nonlinear regression

The general form of a nonlinear regression is:

$$y_k = h(\boldsymbol{x}_k, \boldsymbol{\beta}) + u_k,$$

where $h(\cdot,\cdot)$ is some unknown function of the regressors and parameters.

- $y = e^{\beta_0} e^{\beta_1 x_1} e^{\beta_2 x_2} e^{u}$ - can be linearized
- $y = \beta_0 + \beta_1 e^{\beta_2 x_1} + u$ - cannot be linearized
- $y = \beta_0 + \beta_1 x_1^{\gamma} + u$ - cannot be linearized

A popular special case of the non-linear regression is the single-index model

$$y_k = h(\boldsymbol{x}_k' \boldsymbol{\beta}) + u_k,$$

thus h is a function of a linear combination of the regressors.

Assumptions

- as before +
- $E(\boldsymbol{u}|\boldsymbol{X}) = \boldsymbol{0}$ is replaced with $E(u_i|h(\boldsymbol{x}_i, \boldsymbol{\beta})) = 0$: if u is uncorrelated with $\boldsymbol{x}$ it still may be correlated with some function of $\boldsymbol{x}$. In general $E(\boldsymbol{u}|\boldsymbol{X}) = \boldsymbol{0}$ is not needed.
- Identifiability of the model parameters: the model is identifiable if there is no a non-zero parameter $\boldsymbol{\beta}_0$, such that $h(\boldsymbol{x}_i, \boldsymbol{\beta}_0) = h(\boldsymbol{x}_i, \boldsymbol{\beta})$ for all $\boldsymbol{x}_i$.

  Note: in the linear regression it is sufficient to assume $rank(\boldsymbol{X}'\boldsymbol{X}) = J + 1$. Here it is not enough.

$$y = \frac{\beta_0 + \beta_1 x_1}{\beta_2 + \beta_3 x_2} + u.$$

Estimation: the LS estimation can be used, but the asymptotic theory follows in a straightforward way from the quasi (!) maximum-likelihood estimation.
Assuming Gaussian residuals it holds:

$$\begin{aligned}\mathcal{L}(y_k|\boldsymbol{x}_k,\boldsymbol{\beta},u_k) &= \frac{1}{K}\sum_{k=1}^{K}\ln f(y_k|\boldsymbol{x}_k,\boldsymbol{\beta},u_k)\\ &= \frac{1}{K}\sum_{k=1}^{K}\ln\left\{\frac{1}{\sqrt{2\sigma^2}}exp\left(-\frac{1}{2\sigma^2}(y_k-h(\boldsymbol{x}_k,\boldsymbol{\beta}))^2\right)\right\}\\ &= \frac{1}{K}\sum_{k=1}^{K}\left[-\ln\sqrt{2\sigma^2}-\frac{1}{2\sigma^2}(y_k-h(\boldsymbol{x}_k,\boldsymbol{\beta}))^2\right]\longrightarrow max\end{aligned}$$

Thus the first order conditions for $\boldsymbol{\beta}$ are

$$\frac{\partial\mathcal{L}(y_k|\boldsymbol{x}_k,\boldsymbol{\beta},u_k)}{\partial\boldsymbol{\beta}}=\sum_{k=1}^{K}(y_k-h(\boldsymbol{x}_k,\boldsymbol{\beta}))\frac{\partial h(\boldsymbol{x}_k,\boldsymbol{\beta})}{\partial\boldsymbol{\beta}}=\mathbf{0}.$$

$\rightsquigarrow$ mostly a highly nonlinear system of equations solved numerically.

Consequences: since the resulting $\hat{\boldsymbol{\beta}}$ is a non-linear function of the residuals u_k

- ... the unbiasedness can not be proven in simple fashion;
- ... the variance of $\hat{\boldsymbol{\beta}}$ is not easy to derive;
- ... the exact distribution of $\hat{\boldsymbol{\beta}}$ is not Gaussian;
- ... all the inferences, like tests, are valid only asymptotically.

but the ML estimators are consistent and efficient (they possess the smallest variance among all consistent and asymptotically normal estimators)

Where all this comes from?

- Taylor expansion of $f(x)$ in neighborhood of x_0

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2} f''(x_0)(x - x_0)^2 + ...$$

- Exact Taylor expansion of $f(x)$ in neighborhood of x_0 (mean-value theorem)

$$f(x) = f(x_0) + f'(x_+)(x - x_0),$$

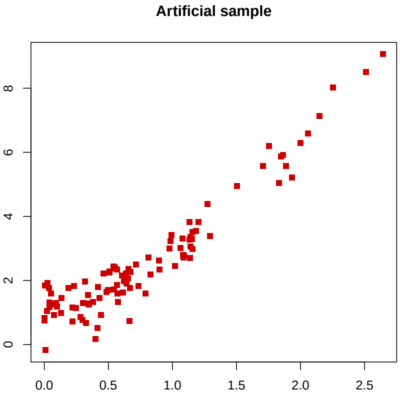
  where x_+ lies between x and x_0.

- $$\left.\frac{\partial \mathcal{L}(y_k|\boldsymbol{x}_k, \boldsymbol{\beta}_k, u_k)}{\partial \boldsymbol{\beta}}\right|_{\hat{\boldsymbol{\beta}}} = \left.\frac{\partial \mathcal{L}(y_k|\boldsymbol{x}_k, \boldsymbol{\beta}_k, u_k)}{\partial \boldsymbol{\beta}}\right|_{\boldsymbol{\beta}} + \left.\frac{\partial^2 \mathcal{L}(y_k|\boldsymbol{x}_k, \boldsymbol{\beta}_k, u_k)}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}'}\right|_{\boldsymbol{\beta}_+} (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})$$

- $$\sqrt{K}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) = -\left(\left.\frac{\partial^2 \mathcal{L}(y_k|\boldsymbol{x}_k, \boldsymbol{\beta}_k, u_k)}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}'}\right|_{\boldsymbol{\beta}_+} \right)^{-1} \sqrt{K} \left.\frac{\partial \mathcal{L}(y_k|\boldsymbol{x}_k, \boldsymbol{\beta}_k, u_k)}{\partial \boldsymbol{\beta}}\right|_{\boldsymbol{\beta}}$$

- $$\sqrt{K}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \overset{approx}{\sim} N(\boldsymbol{0}, \boldsymbol{A}^{-1} \boldsymbol{B} \boldsymbol{A}^{-1})$$

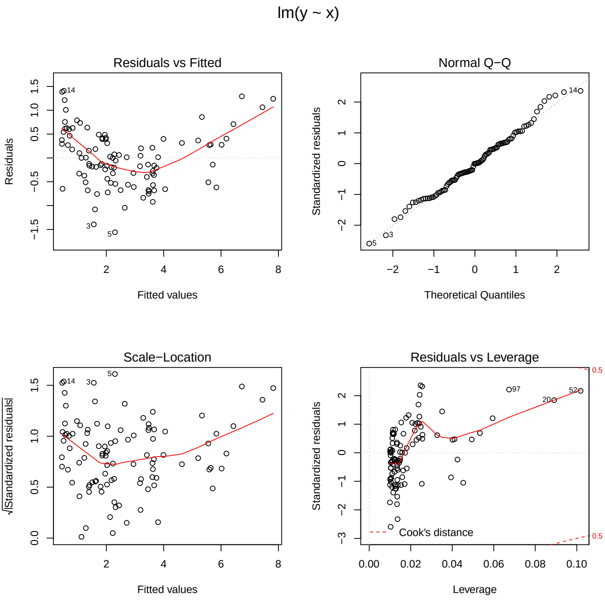
Example:



- Model 1 : $y = \beta_0 + \beta_1 x + u$
- Model 2 : $y = \beta_0 + \beta_1 x^{\beta_2} + u$

```
  > z1 = lm(y~x);
  > z2 = nls(y~ b0+b1*x^b2, start=list(b0=0, b1=1, b2=2))
  > summary(z1)

  Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.44547    0.09694   4.595 1.29e-05 ***
x            2.78805    0.09787  28.488  < 2e-16 ***
---
Residual standard error: 0.604 on 98 degrees of freedom
Multiple R-squared:  0.8923,    Adjusted R-squared:  0.8912
F-statistic: 811.5 on 1 and 98 DF,  p-value: < 2.2e-16

  > summary(z2)
  Formula: y ~ b0 + b1 * x^b2

Parameters:
   Estimate Std. Error t value Pr(>|t|)
b0  1.08702    0.09345   11.63   <2e-16 ***
b1  1.77926    0.13041   13.64   <2e-16 ***
b2  1.56810    0.08382   18.71   <2e-16 ***
---

Residual standard error: 0.4678 on 97 degrees of freedom

Number of iterations to convergence: 5
Achieved convergence tolerance: 2.905e-06
```

True model: $y = 1 + 2x^{1.5} + u$, $u \sim N(0, 0.5^2)$.


lm(y ~ x)
Residuals vs Fitted
Residuals
Fitted values
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Leverage
Cook's distance
0.5
0.5

Lasso regression

Problems:

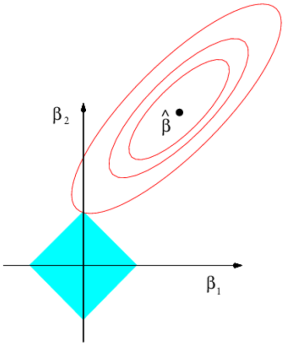
- accuracy: In K is much larger than J, then the variances are small and the inferences are precise. Low number of observations per parameter implies general high variability.
- interpretability: In large data sets there always irrelevant variables which make the economic interpretability difficult.
- sparsity: only a subset of the explanatory variables is relevant economically and statistically.

Solution: stepwise variable selection procedures based on statistical properties of the estimators or lasso regression

Idea: minimize the sum of squared residuals with constrains on the parameters.

The objective function of the OLS procedure is replaced with

$$\sum_{k=1}^{K}(y_k - \beta_0 - \sum_{j=1}^{J}\beta_j x_{kj})^2 \; + \; \lambda \sum_{j=1}^{J}|\beta_j| \longrightarrow min, \text{ w.r.t } \beta_j\text{'s}$$



Note:

- The problem is equivalent to the following problem, i.e. for each λ there exists s such that both problems lead to the same lasso-coefficients.

$$\sum_{k=1}^{K}(y_k - \beta_0 - \sum_{j=1}^{J}\beta_j x_{kj})^2 \longrightarrow min, \text{ w.r.t. } \beta_j\text{'s}$$

$$\text{s.t.} \quad \sum_{j=1}^{J}|\beta_j| \leq s.$$

- Minimizing the objective is not trivial and there many specific numerical methods developed for this purpose.
- Selecting a good value for λ is crucial. The optimal value is chosen by cross-validation.

Special case

Assume an individual constant for each observation:

$$\sum_{k=1}^{K}(y_k - \beta_k)^2$$

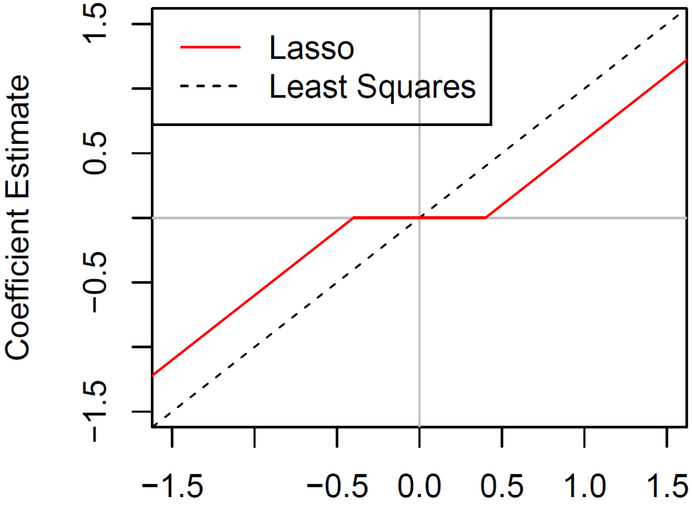
with the OLS solution $\hat{\beta}_k = y_k$.

With lasso we obtain:

$$\sum_{k=1}^{K}(y_k - \beta_k)^2 + \lambda\sum_{k=1}^{K}|\beta_k| \longrightarrow min$$

with the solution

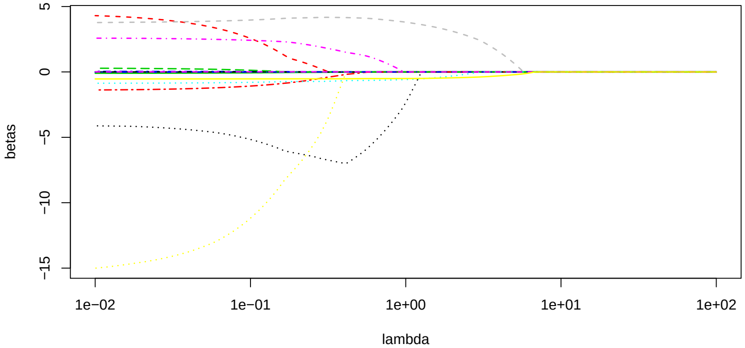
$$\hat{\beta}_k^{(lasso)} = \begin{cases} y_k - \lambda/2, & \text{if } y_k \geq \lambda/2 \\ y_k + \lambda/2, & \text{if } y_k \leq -\lambda/2 \\ 0, & \text{if } |y_k| \leq \lambda/2 \end{cases}$$



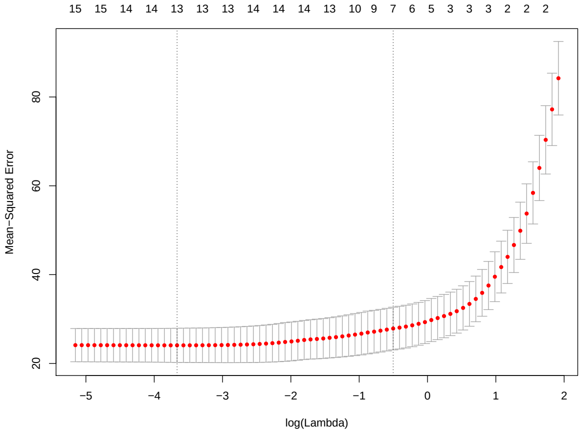
Universität Augsburg

Example:

```
> grid = 10^seq(2,-2, length=100)
> lasso = glmnet(X, y.boston, alpha=1, lambda=grid);
> plot(lasso$beta)
```



```
> cv.lasso = cv.glmnet(X, y.boston, alpha=1);
> plot(cv.lasso)
> cv.lasso$lambda.min
[1] 0.0255856
```



```
> lasso.coef = predict(lasso, type="coefficients", s=cv.lasso$lambda.min);
> lasso.coef
16 x 1 sparse Matrix of class "dgCMatrix"
                        1
(Intercept) -4.392079e+02
lon         -4.250433e+00
lat          4.017435e+00
crim        -9.553123e-02
zn           4.149031e-02
indus        .
chas1        2.557345e+00
nox         -1.432740e+01
rm           3.816713e+00
age          .
dis         -1.329954e+00
rad          2.572257e-01
tax         -1.071061e-02
ptratio     -8.520927e-01
b            8.974756e-03
lstat       -5.326668e-01
```

Chapter 7

Modeling binary, nominal and count data

Modeling binary variables

Practical question: a bank should decide about granting loans to new clients, i.e. forecast of the solvency

$$Y_i = \begin{cases} 0, & \text{the client } i \text{ is solvent} \\ 1, & \text{the client } i \text{ is insolvent} \end{cases}$$

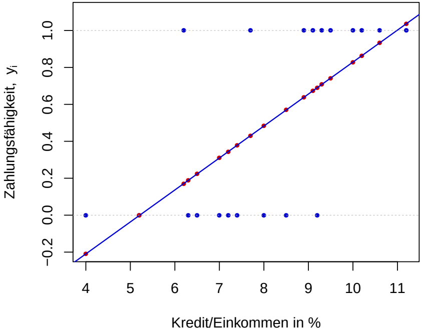
X_{1i}	–	debt-to-income ratio ($\times 100$);
X_{2i}	–	years with the current employer;
X_{3i}	–	other debts (in 1000 Euro);
X_{4i}	–	age (in years).

Question: can we use a linear regression model for binary variales? $\rightsquigarrow$ linear probability model

Universität Augsburg

Linear Prob.-model

$$Y_i = \beta_0 + \beta_1 \cdot X_i + u_i$$



Note:

- (+) the forecast $\hat{Y}_i$ can be seen as probability

$$E(Y_i|X_i) = 1 \cdot P(Y_i = 1|X_i) + 0 \cdot P(Y_i = 0|X_i) = p_i$$

- (−) $\hat{Y}_i$ may lie outside of [0,1]
- (−) R^2 is useless as a goodness-of-fit measure
- (−) the residuals are not normally distributed
- (−) $Var(Y_i|X_i) = p_i(1-p_i) \neq const \rightsquigarrow$ heteroscedastic

Transition to Logit/Probit

Let Y_i be the observed binary variable and Y_i^* the corresponding unobserved metric variable. For Y_i^* it holds:

$$Y_i^* = \beta_0 + \beta_1 X_{1i} + \cdots + \beta_k X_{ki} + u_i = \boldsymbol{X}_i' \boldsymbol{\beta} + u_i.$$

Example: Y_i^* is an unobserved solvency of the client i with

$$Y_i = 1 \text{ if } Y_i^* > 0 \text{ and } Y_i = 0 \text{ if } Y_i^* \leq 0.$$

$$\begin{aligned} P(Y_i = 1|\boldsymbol{X}_i) &= P(Y_i^* > 0|\boldsymbol{X}_i) = P(\boldsymbol{x}_i'\boldsymbol{\beta} + u_i > 0|\boldsymbol{X}_i) \\ &= P(-u_i < \boldsymbol{X}_i'\boldsymbol{\beta}|\boldsymbol{X}_i) = F(\boldsymbol{X}_i'\boldsymbol{\beta}), \end{aligned}$$

where $F(\cdot)$ is the cdf of the residuals.

- $F(z) = \frac{1}{1+e^{-z}}$ - the cdf of the logistic distribution $\rightsquigarrow$ logit
- $F(z)$ - the cdf of the normal distribution $\rightsquigarrow$ probit

Universität Augsburg

Logistic regression

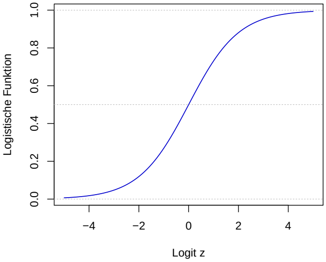
Idea: transformation with the logistic function

Logistic model

$$P(Y_i = 1|\boldsymbol{X}_i) = \frac{1}{1 + e^{-z_i}};$$

for logits z_i it holds

$$z_i = \beta_0 + \beta_1 X_{1i} + \cdots + \beta_k X_{ki}$$



Note: Alternatively we may use the CDF $\Phi(z_i)$ of $N(0, 1)$ $\rightsquigarrow$ probit-model

Estimation of the parameters

The parameters are estimated using ML:

$$L = \prod_{i=1}^{n} \underbrace{\left(\frac{1}{1+e^{-z_i}}\right)^{y_i}}_{P(Y_i=1)} \cdot \underbrace{\left(1-\frac{1}{1+e^{-z_i}}\right)^{1-y_i}}_{P(Y_i=0)} \longrightarrow max, \text{ w.r.t. } \beta_0, \ldots, \beta_k.$$

Note:

- In contrary to the LR the estimation is always numeric.
- Likelihood-Ratio tests can be used to check the significance of the parameters.
- R: `glm(y ~ X,data=data, family=binomial(logit))`

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.434785   0.482326  -2.975  0.00293 **
debtinc      0.121391   0.019023   6.381 1.76e-10 ***
employer -0.161795   0.023742  -6.815 9.44e-12 ***
debts       0.093460   0.045045   2.075  0.03801 *
age        -0.004397   0.014212  -0.309  0.75701
---
```

Example: a data set with 700 observations

	debtinc	employer	debts	age
$\hat{\beta}_i$	0.121*	-0.162*	0.093*	-0.004
$e^{\hat{\beta}_i}$	1.129	0.851	1.098	0.996

(*) - significant with $\alpha = 0.05$

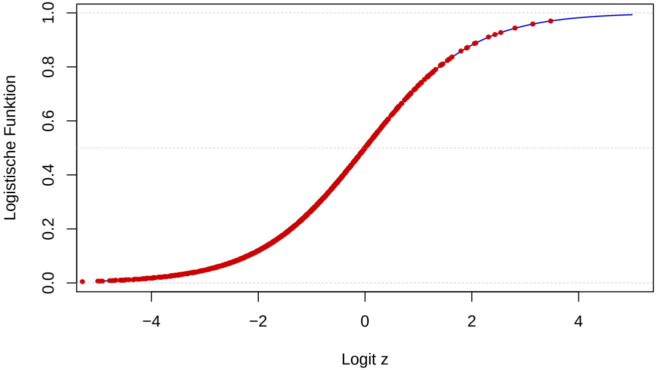
Odds of the logistic regression

$$\text{Odds} = \frac{P(Y=1|\boldsymbol{X})}{P(Y=0|\boldsymbol{X})} = e^z$$

	Logit (z)	Odds	$P(Y=1\|\boldsymbol{X})$
$\beta > 0$	rises by β	rises by e^β	rises
$\beta < 0$	falls by β	falls by e^β	falls

Forecasts:

$$P(\widehat{Y_0 = 1}|\boldsymbol{X}_0) = \hat{\pi}_0 = \frac{1}{1 + e^{-\hat{b}_0 - \hat{b}_1 X_{10} - \ldots - \hat{b}_k X_{k0}}}.$$



Goodness of the model

Problem: classical measures, such as R^2, cannot be used $\rightsquigarrow$ pseudo-R^2; classification tables; graphical measures (ROC-curve)

- pseudo-R^2:
  - Let LL_0 be the Log-Likelihood of the null model ($b_1 = \cdots = b_k = 0$)
  - Let LL_v be the Log-Likelihood of the full model (with all variables)
  - Let LL_s be the Log-Likelihood of the saturated model (model with perfect fit, here $LL_s = 0$)
  - Deviance: $D = -2 \cdot LL_v$ (close 0)
  - Mc-Faddens-R^2: $1 - LL_v / LL_0$ (starting from 0.4)

```
    Null deviance: 804.36  on 699  degrees of freedom
Residual deviance: 626.49  on 695  degrees of freedom
```


Universität Augsburg

- Classification table

		predicted		
		$\hat{Y}=1$	$\hat{Y}=0$	
truth	1	$n_{11}=TP$	$n_{01}=FN$	$n_{.1}=P$
	0	$n_{10}=FP$	$n_{00}=TN$	$n_{.0}=N$
		$n_{1.}$	$n_{0.}$	

Let $\hat{y}_i = 1$ if $\widehat{P(Y_i = 1|\boldsymbol{X}_i)} > 0.5$ and 0 else.

	$\hat{Y}=1$	$\hat{Y}=0$
$Y=1$	72	111
$Y=0$	38	479

$\rightsquigarrow$ (479+72)/700 = 78,71% are correctly predicted.

But: there are 73,86% solvent clients in the sample.

Quaestion: is the threshold 0.5 a good choice?

Goodness of the model and the choice of the threshold

- ROC (*receiver operating characteristics*), Lift and Gain curves are used to visualize and to quantify the goodness of the classification algorithms.

$$\begin{aligned}
\text{sensitivity} &= \frac{n_{11}}{n_{\cdot 1}} = \frac{n_{11}}{n_{11} + n_{01}} \\
\text{specifity} &= \frac{n_{00}}{n_{\cdot 0}} = \frac{n_{00}}{n_{10} + n_{00}}
\end{aligned}$$

Sensitivity: the fraction of correctly classified 1-values among all true 1-objects.

Specifity: the fraction of correctly classified 0-values among all true 0-object.

- Sensitivity$= 72/(72 + 111) = 0.39$ - only 39% of insolvent clients are classified as insolvent
- Specifity$= 479/(479 + 38) = 0.92$ - 92% of solvent clients are classified as solvent

$$
\begin{aligned}
\text{PPV or PV+} &= \frac{n_{11}}{n_{1\cdot}} = \frac{n_{11}}{n_{11}+n_{10}} \\
\text{NPV or PV-} &= \frac{n_{00}}{n_{0\cdot}} = \frac{n_{00}}{n_{01}+n_{00}}
\end{aligned}
$$

PPV: the fraction of correctly classified 1-values among all objects classified as 1.

NPV: the fraction of correctly classified 0-values among all objects classified as 0.

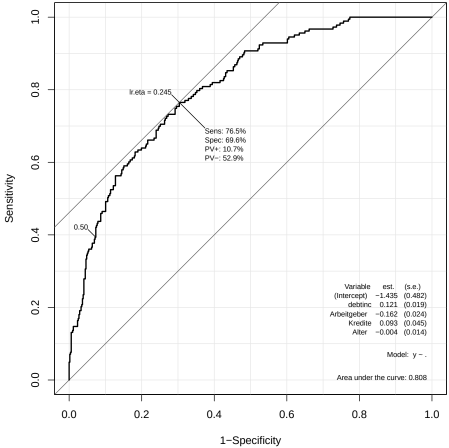
(PPV-positive predicted value, NPV-negative predicted value)

- PPV$= 72/(72+38) = 0.65$ - only 65% of all as insolvent classified clients are really insolvent
- NPV$= 479/(479+111) = 0.81$ - 81% of all as solvent classified clients are really solvent

ROC-curve: senstitivity values as a function of specifity

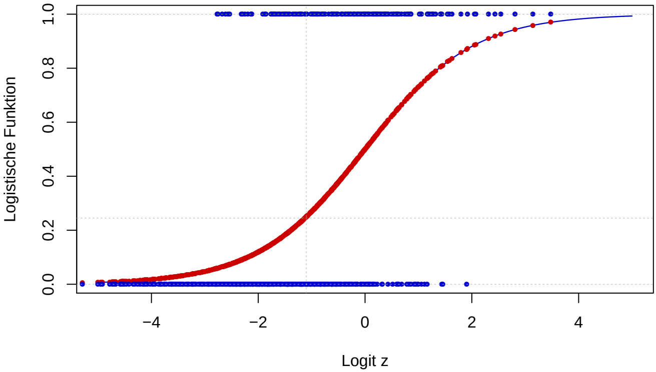
- The steeper the function, the better the algorithm. ROC-value is the square under the curve.
- If the curve is close to the diagonal, then the algorithm is as good as random assignments.

R: `roc`-function from the `pROC`-package

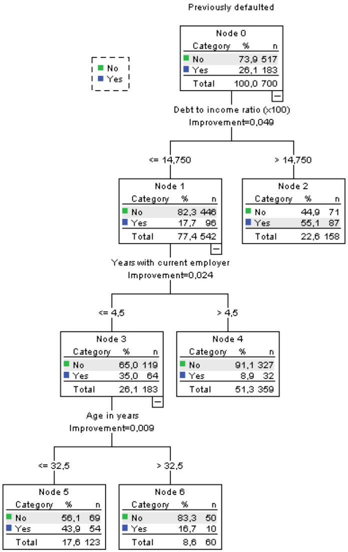


Now let $\hat{y}_i = 1$ if $\widehat{P(Y_i = 1|\boldsymbol{X}_i)} > 0.245$ and 0 else.

	$\hat{Y} = 1$	$\hat{Y} = 0$
$Y = 1$	140	43
$Y = 0$	158	359



The CART method can be applied to binary data: classification trees



Classification

Observed	Predicted No	Predicted Yes	Predicted Percent Correct
No	446	71	86,3%
Yes	96	87	47,5%
Overall Percentage	77,4%	22,6%	76,1%

Growing Method: CRT

Dependent Variable: Previously defaulted

Modelling nominal data

Practical question:

- Choice of the political party depending of the characteristics of the voters;
- Choice of a product brand depending on the characteristics of the client;

Example:

mode	–	“car”, “air”, “train”, oder “bus”
choice	–	decision
wait	–	waitnning time, 0 for “car”
vcost	–	variable costs
travel	–	time
gcost	–	total costs
income	–	income
size	–	number of persons

	individual	mode	choice	wait	vcost	travel	gcost	income	size
1	1	air	no	69	59	100	70	35	1
2	1	train	no	34	31	372	71	35	1
3	1	bus	no	35	25	417	70	35	1
4	1	car	yes	0	10	180	30	35	1
5	2	air	no	64	58	68	68	30	2
6	2	train	no	44	31	354	84	30	2

Multinomial logit model

For the simple logit model it holds:

$$P(Y=1|\boldsymbol{x}) = \frac{exp(\boldsymbol{x}'\boldsymbol{\beta})}{1+exp(\boldsymbol{x}'\boldsymbol{\beta})}$$

$$\ln\left(\frac{P(Y=0|\boldsymbol{x})}{P(Y=1|\boldsymbol{x})}\right) = \boldsymbol{x}'\boldsymbol{\beta}$$

For the k categories of Y we define:

$$\ln\left(\frac{P(Y=r|\boldsymbol{x})}{P(Y=k|\boldsymbol{x})}\right) = \boldsymbol{x}'\boldsymbol{\beta}_r, \quad r=1,\dots,k-1$$

with

$$\begin{aligned} P(Y=r|\boldsymbol{x}) &= \frac{exp(\boldsymbol{x}'\boldsymbol{\beta}_r)}{1+\sum_{s=1}^{k-1} exp(\boldsymbol{x}'\boldsymbol{\beta}_s)}, \quad r=1,\dots,k-1 \\ P(Y=k|\boldsymbol{x}) &= \frac{1}{1+\sum_{s=1}^{k-1} exp(\boldsymbol{x}'\boldsymbol{\beta}_s)}. \end{aligned}$$

One category, i.e. the k-th, is the reference category.

Note:

- Estimation via ML assuming independence of the observations. This is a questionable assumption:
  - similar categories;
  - odds do not depend on other categories, etc.
  - Solution: Hausmann/McFadden test
- Goodness-of-fit, tests as for logit.

Global and category specific variables

$$\boldsymbol{x}'\boldsymbol{\beta}_r \quad \mapsto \quad \boldsymbol{x}'_{glob}\boldsymbol{\beta}^*_r + \boldsymbol{x}'_{spec,r}\boldsymbol{\alpha}$$

- Global variables (income, number of persons) do not depend on the categories and have individual parameters for each category: $\boldsymbol{x}'_{glob}\boldsymbol{\beta}^*_r$.
  The sign of the parameters cannot be interpreted.
- The category specific variables (waiting time, costs) depend on the categories and are evaluated relatively to the reference category.

$$(\boldsymbol{x}_{spec,r} - \boldsymbol{x}_{spec,k})'\boldsymbol{\alpha} \quad \text{or} \quad \boldsymbol{x}'_{spec,r}\boldsymbol{\alpha}$$

The sign of the parameters can be interpreted.

Let *gcost* and *wait* be category specific and *income* and *size* are global variables. The reference category is *air*.

```
> library("mlogit")
> mlogit(choice~wait+gcost|income+size, ...)

Coefficients :
                    Estimate Std. Error t-value  Pr(>|t|)
train:(intercept) -2.3115942  0.7525161 -3.0718 0.0021276 **
bus:(intercept)   -3.4504941  0.9064886 -3.8064 0.0001410 ***
car:(intercept)   -7.8913907  0.9880615 -7.9867 1.332e-15 ***
wait              -0.1013180  0.0112207 -9.0296 < 2.2e-16 ***
gcost             -0.0197064  0.0053844 -3.6599 0.0002523 ***
train:income      -0.0589804  0.0154532 -3.8167 0.0001352 ***
bus:income        -0.0277037  0.0169812 -1.6314 0.1027991
car:income        -0.0041153  0.0127301 -0.3233 0.7464866
train:size         1.3289497  0.3141683  4.2301 2.336e-05 ***
bus:size           1.0090796  0.3952899  2.5528 0.0106874 *
car:size           1.0392585  0.2665513  3.8989 9.663e-05 ***
---

Log-Likelihood: -176.77
McFadden R^2:  0.37705
Likelihood ratio test : chisq = 213.98 (p.value = < 2.22e-16)
```

With the estimated paremeters we can estimate the probabilities $P(Y_i = r|\boldsymbol{x}_i)$ for all r.

```
         air      train        bus       car
[1,] 0.2368302 0.00000000 0.24496423 0.5182056
[2,] 0.2083323 0.27785076 0.00000000 0.5138170
[3,] 0.0000000 0.12686485 0.23058033 0.6425548
[4,] 0.1151004 0.05063597 0.02141839 0.8128452
[5,] 0.3405917 0.20694648 0.05624436 0.3962174
[6,] 0.1316850 0.36965292 0.26144217 0.2372200
```

Modelling for count data

Practical questions:

- the number of claims by an insurance company per time period;
- the number of cosultations by a doctor per year ;
- the number of insolvent companies per time period;
- occurrences of a seldom disease per season;
- ....

Note: the modelling is particularly important for small values of the target variable (rare events) and the distribution is heavily skewed.

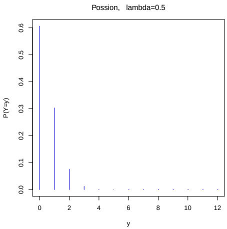
Poisson distribution

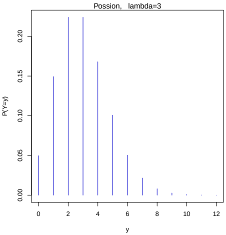
The Poisson distribution is frequently used to model rare events

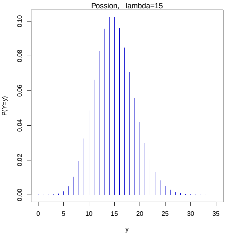
$$P(Y=y)=\begin{cases}\frac{\lambda^y}{y!}e^{-y}, & \text{for } y=0,1,2,...\\ 0, & \text{else,}\end{cases}$$

with the intensity parameter λ. It fulfils the *equidispersion*-condition:

$$E(Y)=Var(Y)=\lambda$$







Poisson regression model

Let $Y_i, \boldsymbol{x}_i$ be independent realisations, while Y_i follows Poisson distribution with

$$E(Y_i|\boldsymbol{x}_i) = h(\boldsymbol{x}_i'\boldsymbol{\beta}) = exp(\boldsymbol{x}_i'\boldsymbol{\beta}) = \lambda_i.$$

- The interpretation of the parameters follows as for the logit model.
- The parameters are estimated via ML:

$$LL(\boldsymbol{\beta}) = \sum_{i=1}^{n} y_i \ln(h(\boldsymbol{x}_i'\boldsymbol{\beta})) - h(\boldsymbol{x}_i'\boldsymbol{\beta}) - \ln(y_i!) \longrightarrow max, \text{ w.r.t. } \boldsymbol{\beta}$$

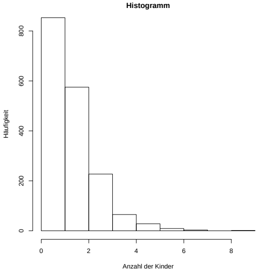
Goodness of the model

To measure the goodness of the model we use deviance, i.e. the difference between the log-likelihood for the actual observations (perfect/saturiertes model) and the log-likelihood for the predicted values:

$$D = -2\sum_{i=1}^{n}[LL_i(\hat{Y}_i) - LL_i(Y_i)] = 2\sum_{i=1}^{n}\left[Y_i \ln(Y_i/\hat{\lambda}_i)\right] \sim \chi^2_{n-p}$$

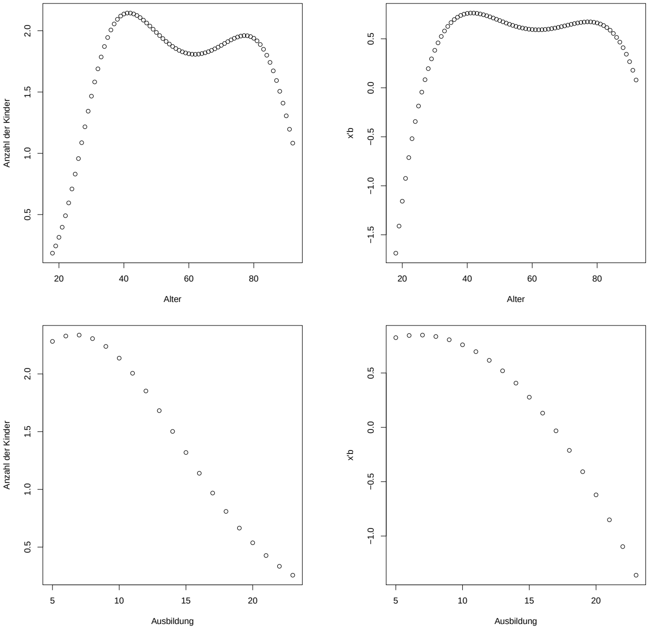
Example: number of children

child	-	number of children
age	-	age of the woman
dur	-	years at school/college
nation	-	nationality, 0 = german , 1 = else
god	-	trust in God: 1 = strong, ..., 6 = never thought about it
univ	-	university degree: 0 = no, 1 = yes




```
mean(children$child)
[1] 1.57297
> var(children$child)
[1] 1.552769
```

```
glm(formula = child ~ age + I(age^2) + I(age^3) + I(age^4) +
    dur + I(dur^2) + nation + god + univ, family = poisson(link = log),
    data = children)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1514  -0.7559   0.0102   0.4832   3.6715

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.228e+01  1.484e+00  -8.277  < 2e-16 ***
age          9.359e-01  1.239e-01   7.553 4.26e-14 ***
I(age^2)    -2.490e-02  3.786e-03  -6.577 4.80e-11 ***
I(age^3)     2.842e-04  4.915e-05   5.781 7.42e-09 ***
I(age^4)    -1.180e-06  2.297e-07  -5.137 2.80e-07 ***
dur          1.118e-01  6.652e-02   1.680 0.092904 .
I(dur^2)    -8.328e-03  2.997e-03  -2.779 0.005454 **
nation1      5.686e-02  1.386e-01   0.410 0.681599
god2        -1.025e-01  5.903e-02  -1.736 0.082599 .
god3        -1.448e-01  6.780e-02  -2.136 0.032683 *
god4        -1.279e-01  7.088e-02  -1.805 0.071128 .
god5        -3.621e-02  6.695e-02  -0.541 0.588569
god6        -9.241e-02  7.505e-02  -1.231 0.218239
univ1        6.372e-01  1.729e-01   3.686 0.000228 ***
---

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 2067.4  on 1760  degrees of freedom
Residual deviance: 1718.6  on 1747  degrees of freedom
AIC: 5196.8
```


Anzahl der Kinder
Alter
x'b
Alter
Anzahl der Kinder
Ausbildung
x'b
Ausbildung

Note: for the Poisson distribution it should hold $E(Y_i) = Var(Y_i) = \lambda_i$.

If this assumption is not fulfilled then we have *overdispersion/underdispersion*.

Solution: as an alternative we can use Quasi-Poisson- or the negative binomial distribution (negbin). Both distributions allow for different expectations and variances.

For negbin it holds:

$$P(Y_i|\boldsymbol{x}_i) = \frac{\Gamma(Y_i + \nu)}{\Gamma(\nu)\Gamma(Y_i + 1)} \cdot \left(\frac{\lambda_i}{\lambda_i + \nu}\right)^{Y_i} \cdot \left(\frac{\nu}{\lambda_i + \nu}\right)^{\nu}$$

with $E(Y_i) = \lambda_i = exp(\boldsymbol{x}_i'\boldsymbol{\beta})$ and $Var(Y_i) = \lambda_i + \lambda_i^2/\nu$.

```
glm(formula = child ~ age + I(age^2) + I(age^3) + I(age^4) +
    dur + I(dur^2) + nation + god + univ, family = negative.binomial(theta = 1,
    link = log), data = children)

Deviance Residuals:
     Min        1Q    Median        3Q       Max
-1.56820  -0.50984  -0.01054   0.29990   1.90633

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.338e+01  1.267e+00 -10.555  < 2e-16 ***
age          1.022e+00  1.075e-01   9.502  < 2e-16 ***
I(age^2)    -2.730e-02  3.342e-03  -8.169 5.90e-16 ***
I(age^3)     3.126e-04  4.395e-05   7.113 1.65e-12 ***
I(age^4)    -1.302e-06  2.074e-07  -6.277 4.34e-10 ***
dur          1.269e-01  5.990e-02   2.118 0.034294 *
I(dur^2)    -9.577e-03  2.637e-03  -3.632 0.000289 ***
nation1      8.309e-02  1.349e-01   0.616 0.538128
god2        -1.186e-01  5.849e-02  -2.028 0.042743 *
god3        -1.681e-01  6.642e-02  -2.530 0.011483 *
god4        -1.563e-01  6.923e-02  -2.258 0.024075 *
god5        -3.273e-02  6.602e-02  -0.496 0.620135
god6        -1.205e-01  7.384e-02  -1.632 0.102848
univ1        7.749e-01  1.581e-01   4.900 1.04e-06 ***
---

(Dispersion parameter for Negative Binomial(1) family taken to be 0.3516262)

    Null deviance: 1023.1  on 1760  degrees of freedom
Residual deviance:  852.3  on 1747  degrees of freedom
AIC: 5911.9
```