An introduction to predictive models

Conceptual Model

Target/dependent variable + explanatory/independent variables

[Is this a causal model?]

- Dependent variable choice depends on objective/research question
- Explanatory variables originates from theoretical/conceptual considerations.

  - Model building example : Effectiveness of search analytics on sales (work backwards)

Conceptual Model Example: Website visitors (Multiple steps)

First increase webpage traffic – Not enough if visitor leave without purchasing today or in the future (think in terms of customer lifetime value).

Intermediate step – make customer search and direct them to useful products. Also, try to register them so we can send them latter info and promotions that could make a sale.

When shopper adds products to basket– Make a sale!

Conceptualization/theory: Targeting

Each step may require a different tool. Which?

Does one method fit all customers? How should we target different customers?

Which webpage layout should I present to different customers?

Which products are more likely to make a sale? On which type of customers?

Examples

- Smart spam classifiers protect our email by learning from massive amounts of spam data and user feedback;
- Advertising systems learn to match the right ads with the right context;
- Fraud detection systems protect banks from malicious attackers;
- Anomaly event detection systems help experimental physicists to find events that lead to new physics, store sales prediction;
- Web text classification;
- Customer behavior prediction;
- Ad click through rate prediction;
- Malware classification;
- Product categorization;
- Hazard risk prediction;

Planning step by step

1. Translate the business problem
2. Select the appropriate data

Business view

3. Fix problems with the data (preprocess)
4. Get to know the data (explore)

Chapter 2

5. Build models (estimation)
6. Assess models (validation)

Chapters 3 and 4

7. Deploy models
8. Assess results (compare to objectives established in 1.)

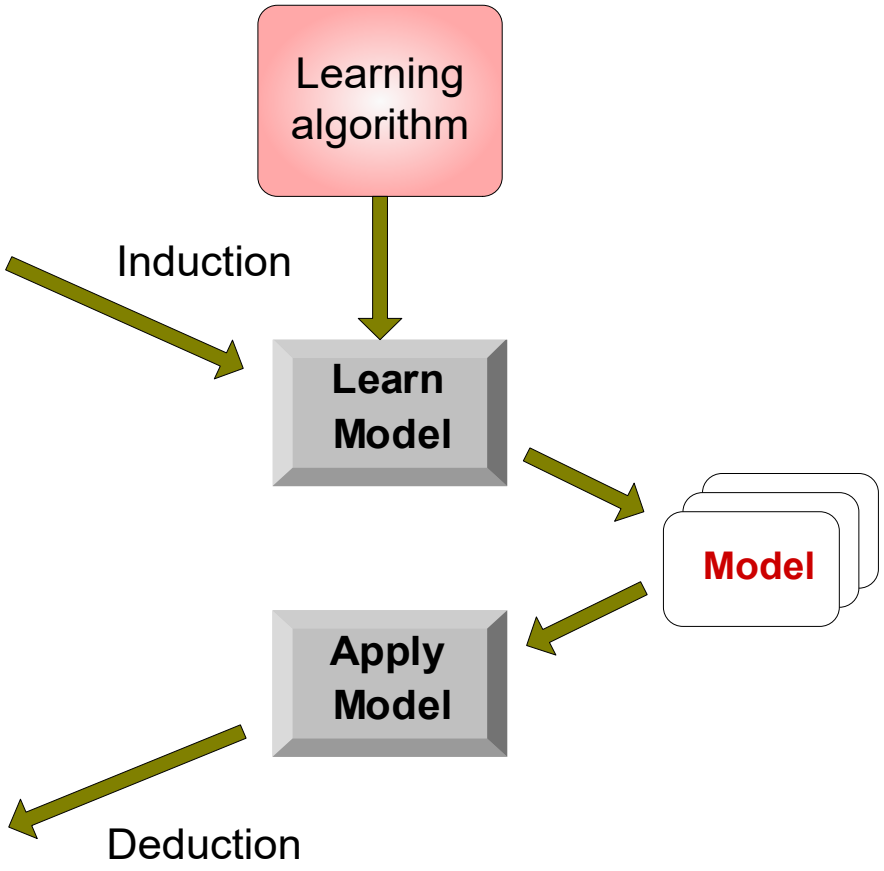
Model building and assessment

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Model types

Directed data mining or supervised learning

Undirected data mining or unsupervised learning

Classical techniques for model building (supervised learning/ directed data mining)

- Table Lookup Models
- Naïve Bayes + LDA
- (Multiple) Linear Regression
- (Multiple) Logistic Regression

John went to an auto-dealer to buy a second-hand car and is not sure if the price is fair. How can he make a decision?

A. Ask a friend

B. Search for the price of similar cars

C. Check the price of the car when new

Table lookup

Based on the idea of similarity.

Building a lookup table:

- Choose input variables and the output score.
- Non-categorical variables must be “discretized”.
- Train the model by looking at the output for given set of input variables. E.g. Average second hand car prices per set of attributes. Average price is the output.

Using the lookup table

- New observations are compared with the elements in the lookup table. A label is assigned.
- The value score (output) is assigned.

Table Lookup example

Risk rating	Age	Income	Effort	Home-owner
5	1	1	1	1
4	2	1	1	1
3	2	2	1	1
3	2	1	2	1
2	2	2	2	1
4	1	2	1	1
2	1	2	2	1
3	1	1	2	1
4	1	1	1	2
3	2	1	1	2
2	2	2	1	2
2	2	1	2	2
1	2	2	2	2
3	1	2	1	2
2	1	2	2	2
2	1	1	2	2

Lookup table

- How to score the risk rating for an individual with 30 years of age, not owing a house (and no other loans), earning 30,000 euros per year and asking for a loan of 10,000 euros?

Income		Age		effort (loan to income)		Home owner	
1	<20,000	1	<25	1	<2,5	1	Yes
2	>20,000	2	>25	2	>2,5	2	No

Variable description

Table lookup

Choose	Choosing dimensions • Dimensions should affect the target variable. • Dimensions should not be correlated with each other (income and age?). • Increase cell estimate accuracy. Avoid cells with few training examples (e.g. [2,1,2,1]?). • Trade-off number of dimensions vs. partitions (levels) of each dimension.
Partition	Partitioning dimensions • Large (finer) partitions (levels) increase accuracy of the target while reducing the accuracy of the estimate. • Nominal dimensions are naturally discrete. Some could be aggregated together (e.g. number of children: 0,1,+2). • Metric dimensions can be discretized (e.g. income, age, etc). Choose equal sizes (e.g. quantiles).

Table lookup

- Estimation [or Training]
  - For numeric target variables: Choose average (median) per class.
  - For categorical target variables: Use a score (proportion of each cell that possesses the given class label).
- Spare and missing data
  - If some cells do not have enough training cases either (i) reduce the number of partitions or (ii) reduce the number of dimensions.

Estimation/Training

Example: three factors with two levels each (1,2) and one categorical outcome with three levels (A,B,C)

Training set

X1	X2	X3	Y
1	1	1	C
1	1	2	A
1	1	2	A
1	1	2	C
1	2	1	B
1	2	2	C
1	2	2	A
1	2	2	C
2	1	1	B
2	1	1	A
2	1	1	A
2	1	1	B
2	1	1	A
2	1	2	C
2	1	2	A
2	2	2	C
2	2	2	C

Model

X1	X2	X3	Predict?
1	1	1	C
1	1	2	A
1	2	1	B
1	2	2	C
2	1	1	A
2	1	2	A/C?
2	2	1	Overall?
2	2	2	C

How can I compare the profitability of multiple customers in my store?

A. By how much they spend
B. By how frequently they buy
C. By the last time they bought something
D. By the type of products they buy

RFM – A Table lookup model

Recency – how recently has the customer made a purchase. The lower the recency the less likely a customer is to make a purchase.

Frequency – How frequently the customer makes a purchase. Customer with frequent purchases are more likely to buy.

Monetary – How much have the customer spent. Customers who spend large amounts are more likely to spend more again.

Table lookup with tree dimensions

DIY – Analysis from transaction data

File: rfm_transactions.sav

- How are the R/F/M scores constructed/distributed?
- How is the combined RFM score calculated? Importance?

SPSS – creating an RFM model

Get Ch03_*rfm_transactions.sav and Ch03_customer_information.sav* from Moodle.

In a transaction data file, each row represents a separate transaction, rather than a separate customer, and there can be multiple transaction rows for each customer.

RFM model

The transaction dataset contains variables with the following information:

- An id (e.g. customer).
- The date of each transaction.
- The monetary value of each transaction.

IBM SPSS

RFM model

The transaction dataset contains variables with the following information:

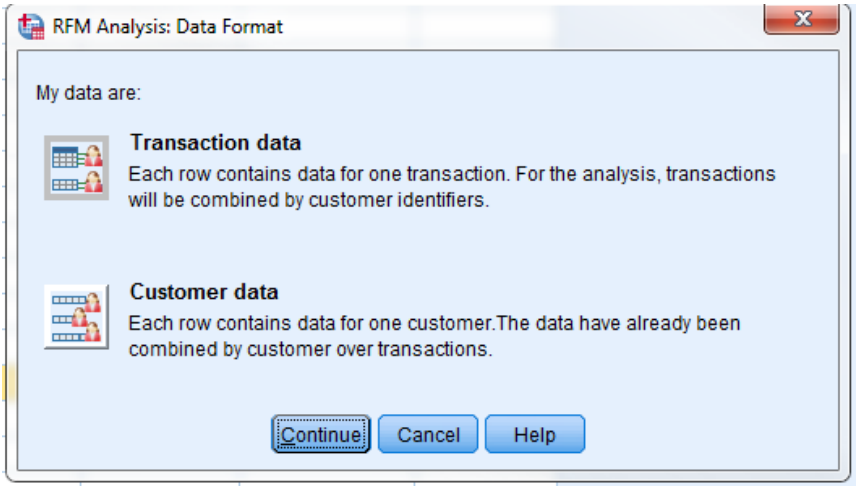
- An id (e.g. customer).
- The date of each transaction.
- The monetary value of each transaction.

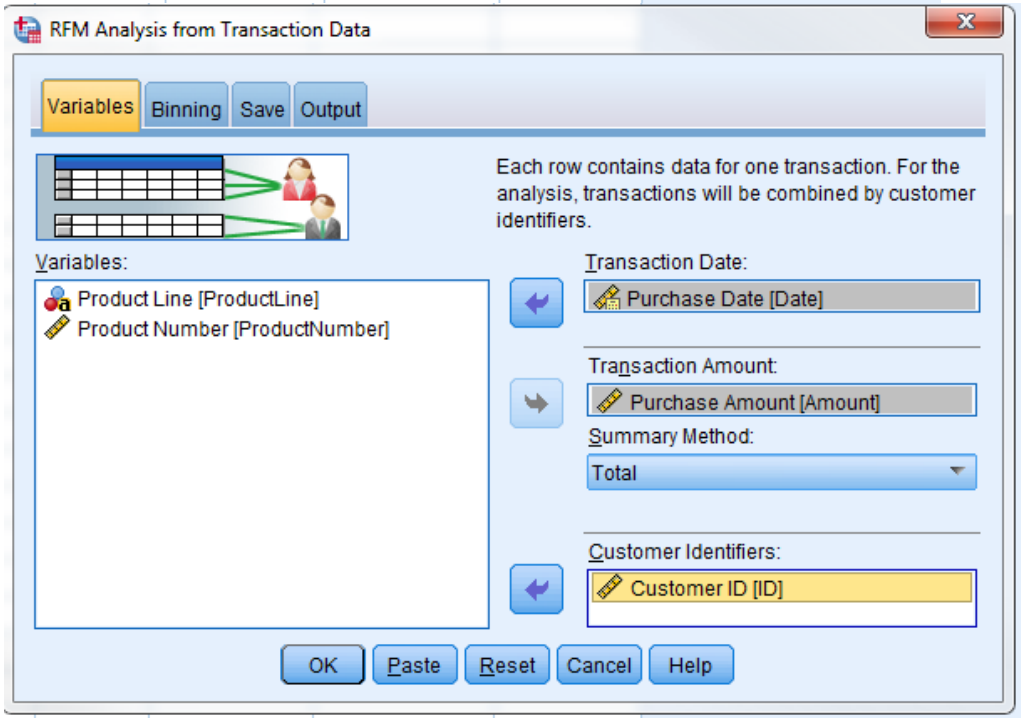
	Name	Type	Width	Decimals	Label
1	ID	Numeric	11	0	Customer ID
2	ProductLine	String	5	0	Product Line
3	ProductNu...	Numeric	11	0	Product Number
4	Date	Date	11	0	Purchase Date
5	Amount	Numeric	11	0	Purchase Amo...

IBM SPSS

RFM model

Direct Marketing > Choose Technique



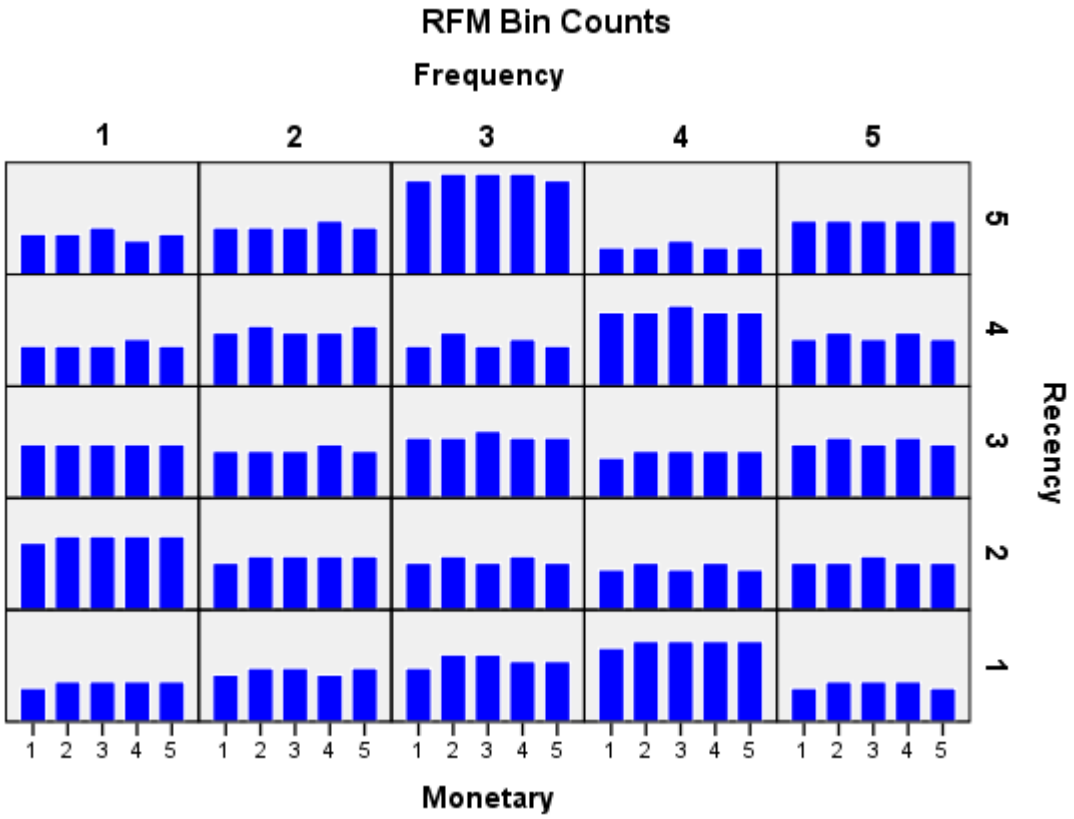


The new dataset contains only one row (record) for each customer. The original transaction data has been aggregated by values of the customer identifier variables. The identifier variables are always included in the new dataset; otherwise you would have no way of matching the RFM scores to the customers.

The combined RFM score for each customer is simply the concatenation of the three individual scores, computed as: (recency x 100) + (frequency x 10) + monetary.

Note that 353 is not “larger” than 335.

The chart of bin counts displayed in the Viewer window shows the number of customers in each RFM category.



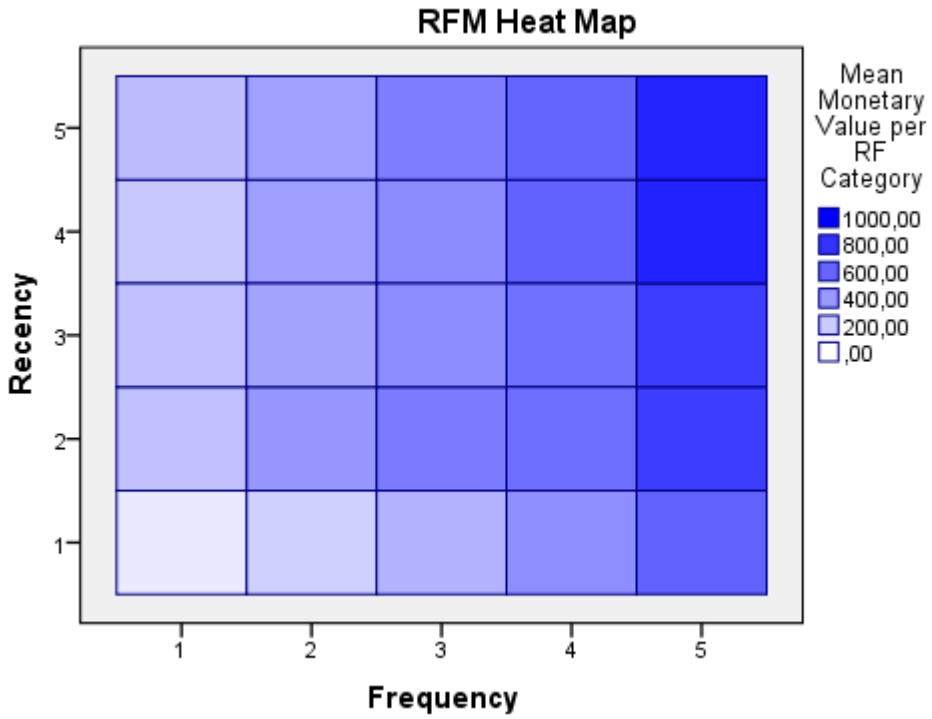
The chart of bin counts displays the bin distribution for the selected binning method. Each bar represents the number of customers that will be assigned each combined RFM score. Although you typically want a fairly even distribution, with all (or most) bars of roughly the same height, a certain amount of variance should be expected when using the default binning method that assigns tied values to the same bin. Extreme fluctuations in bin distribution and/or many empty bins may indicate that you should try another binning method (fewer bins and/or random assignment of ties) or reconsider the suitability of RFM analysis.

Number of customers in each of the 5x5x5 (125) RFM categories.

Ideally, relatively even distribution of customers across RFM categories.

If there are many empty/uneven categories change the binning method:
- Use nested instead of independent binning.
- Reduce the number of possible score categories (bins).
- When there are large numbers of tied values, randomly assign cases with the same scores to different categories.

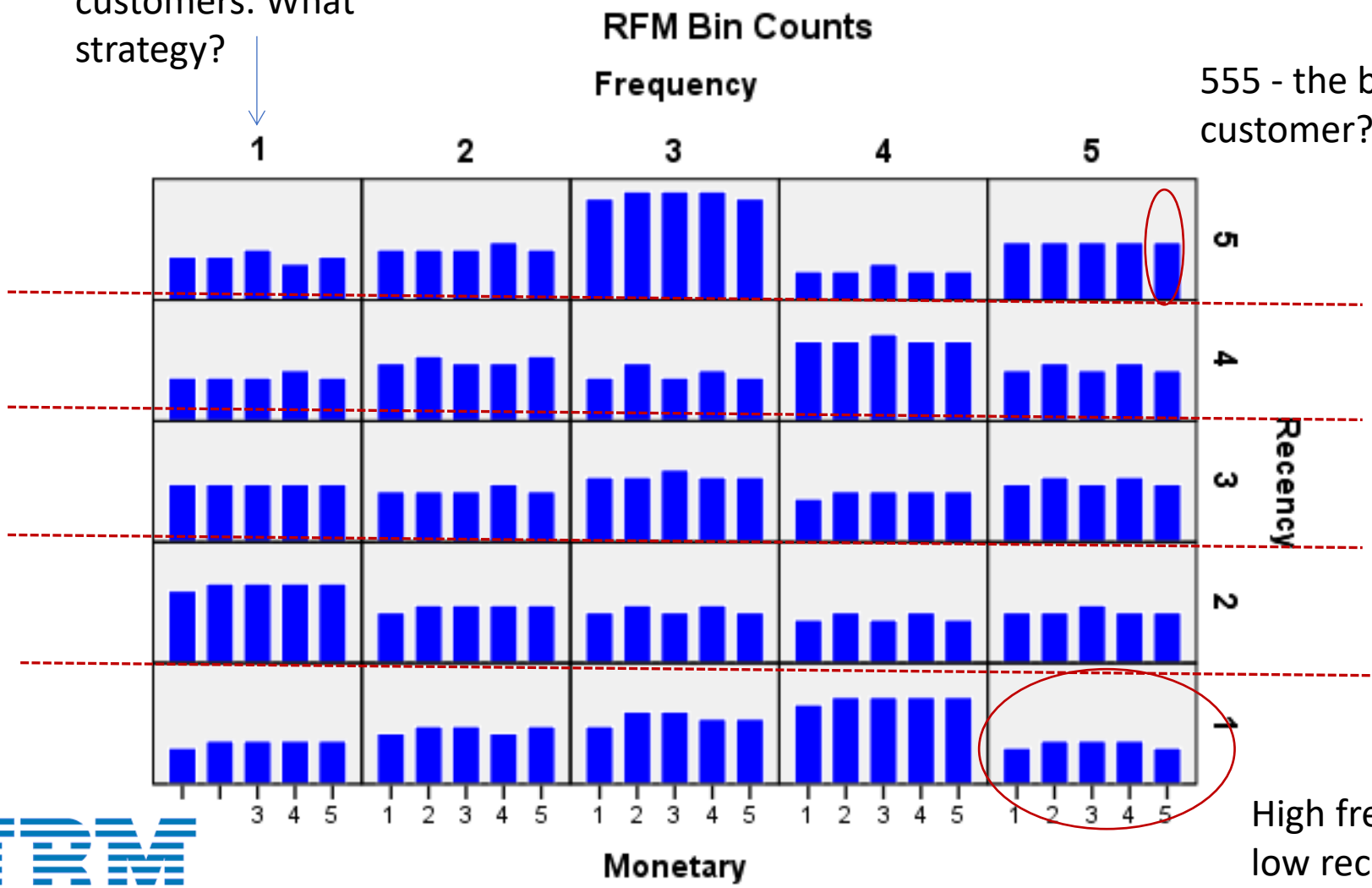
IBM SPSS



The heat map of mean monetary distribution shows the average monetary value for categories defined by recency and frequency scores. Darker areas indicate a higher average monetary value. In other words, customers with recency and frequency scores in the darker areas tend to spend more on average than those with recency and frequency scores in the lighter areas.

Classification

Low frequency
Perhaps new customers. What strategy?

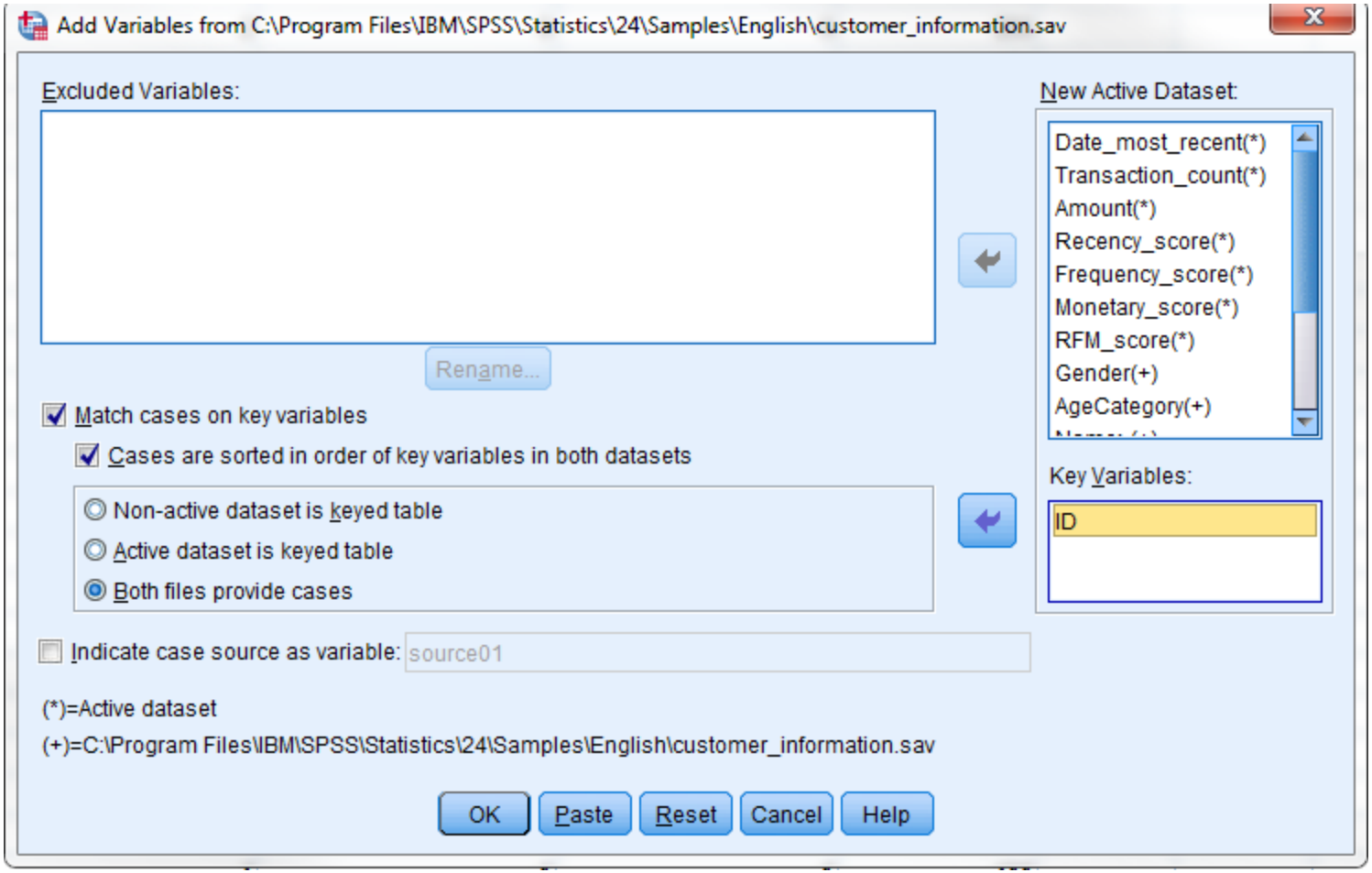


555 - the best customer?

High frequency, low recency.
Lost customers?

IBM SPSS

Data > Merge Files > Add Variables



The final dataset ready for use

	Name	Type	Width	Decimals	Label	Values
1	ID	Numeric	11	0	Customer ID	None
2	Date_most_...	Date	11	0	Date of most re...	None
3	Transaction...	Numeric	7	0	Number of tran...	None
4	Amount	Numeric	8	2	Amount	None
5	Recency_s...	Numeric	3	0	Recency score	None
6	Frequency_...	Numeric	3	0	Frequency score	None
7	Monetary_s...	Numeric	3	0	Monetary score	None
8	RFM_score	Numeric	3	0	RFM score	None
9	Gender	Numeric	2	0		{0, Female}...
10	AgeCategory	Numeric	2	0	Age Category	{1, <25}...
11	Name	String	10	0		None
12	Address	String	10	0		None
13	City	String	10	0		None
14	State_Provi...	String	10	0	State/Province	None
15	PostalCode	String	10	0	Postal Code	None
16	Country	String	10	0		None

John was so successful buying a car that he now wants to use the expertise he gained into the car market to predict the price of all cars in the market.

A. That is easy he just needs to do Table lookup with all the car prices and characteristics.
B. Difficult since there are too many cars.
C. That is impossible since cars vary in so many different ways, it is impossible to get enough cars to compare.

Table Lookup Models

Naïve Bayes + LDA

(Multiple) Linear Regression

(Multiple) Logistic Regression

Naïve bayes

Table lookup quickly become intractable (curse of dimensionality):

Example: 2 inputs with 10 levels is 100 cells, 3 inputs is 1,000 and 4 inputs is 10,000. A sample of 100,000 individuals gives an average of 10 per cell!

Naïve Bayes works around this.

How: Independence assumption. Use prediction of each input on target, independently from the other inputs.

Example: 4 inputs with 10 levels means 10+10+10+10 instead of 10*10*10*10.

Naïve bayes

Bayes Law

$$P(A|B)=P(B|A)*P(A)/P(B)$$

- This is useful because in many cases directly estimating one of the (conditional) probabilities is easier than estimating the other.

- The method is naïve because it (naively) assumes that the variables are **independent** from each other.

Naïve Bayes

Derivation

$$\Pr(Y|X1,\dots,Xk) \overset{by\ Bayes'\ Theorem}{=}$$

$$\frac{P(Y)*P(X1,\dots,Xk|Y)}{P(X1,\dots,Xk)} \overset{by\ independence}{=}$$

$$\frac{P(Y)*P(X1|Y)*P(X2|Y)*\cdots*P(Xk|Y)}{P(X1,\dots,Xk)}$$

- If one of the inputs is not know, say the kth, just drop it from the equation.
- All elements in the numerator are very easy to obtain, the denominator is more difficult. Fortunately, it is fixed so that it can be treated as a constant.

Naïve Bayes

In the binary case the odds ratio is:

$$\frac{\Pr(Y=1|X1,\ldots,Xk)}{\Pr(Y=0|X1,\ldots,Xk)} = \frac{\Pr(Y=1)}{\Pr(Y=0)} * \frac{\Pr(X1|Y=1)}{\Pr(X1|Y=0)} * \cdots * \frac{\Pr(Xk|Y=1)}{\Pr(Xk|Y=0)}$$

So Pr(X1,…,Xk) drops from the equation.

From the odds ratio we can recover the probability since:

$$\text{Odds}=\frac{\Pr(Y=1|X)}{\Pr(Y=0|X)} = \frac{\Pr(Y=1|X)}{1-\Pr(Y=1|X)} \text{ or } \Pr(Y=1|X) = 1 - \frac{1}{1+odds}$$

Main disadvantage of Naïve Bayes is the independence assumption. It can be checked!

Example

Classify a Red, SUV, Domestic for getting stolen

Example No	Color	Type	Origin	Stolen?
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

Need to calculate the following probabilities:

P(Red|Yes), P(SUV|Yes), P(Domestic|Yes) ,

P(Red|No) , P(SUV|No), P(Domestic|No)

Results:

P(Red|Yes) = 3/5 = .0.6

P(Red|No) = 2/5 = 0.4

P(SUV |Yes) = 1/5 = 0.2

P(SUV |No) = 3/5 = 0.6

P(Domestic|Yes) = 2/5 = 0.4

P(Domestic|No) = 3/5 = 0.6

P(Yes) = 5/10 = 0.5

$$\textbf{Odds}=\frac{P(Yes|Red,SUV,Domestic)}{P(No|Red,SUV,Domestic)}=\frac{P(Yes)*P(Red|Y)*P(SUV|Y)*P(Domestic|Y)}{P(No)*P(Red|N)*P(SUV|N)*P(Domestic|N)}$$

$$=\frac{0.5*0.6*0.2*0.4}{0.5*0.4*0.6*0.6}=0.333$$

Answer: A Red domestic SUV is 3 times more likely not to get stolen than to get stolen. Or it has a 25% probability of getting stolen.

DIY – Predictor Selection

Objective: To identify characteristics that are indicative of people who are likely to default on loans. Use those characteristics to identify good and bad credit risks.

Data: File ch03_bank_loan.sav.

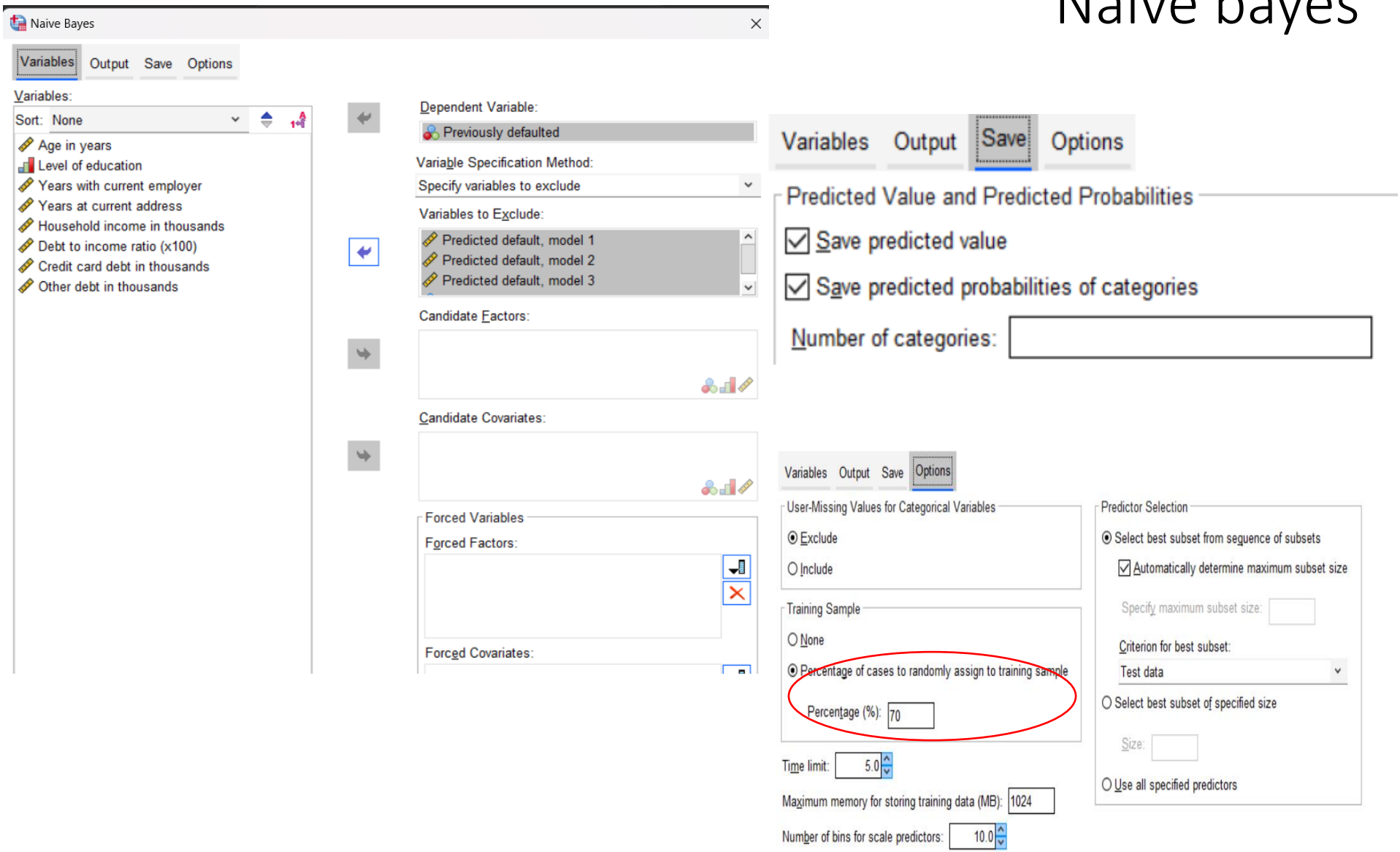
> 850 past and prospective customers. First 700 cases are customers who were previously given loans. Next 150 cases are “unclassified”.

> Split the 700 customers into training and test samples in order to create and validate a model.

> For the remaining 150 cases, since they have valid values for the predictors, the procedure will generate model-predicted probabilities for these cases when you save these values to the dataset.

- Note: Naïve Bayes in SPSS can also be called as a FUNCTION available with in the SPSS statistics server. Requires Syntax (i.e. non menu based) to run. Command syntax for reproducing these analyses can be found in *ch03_bank_loan.sps*.

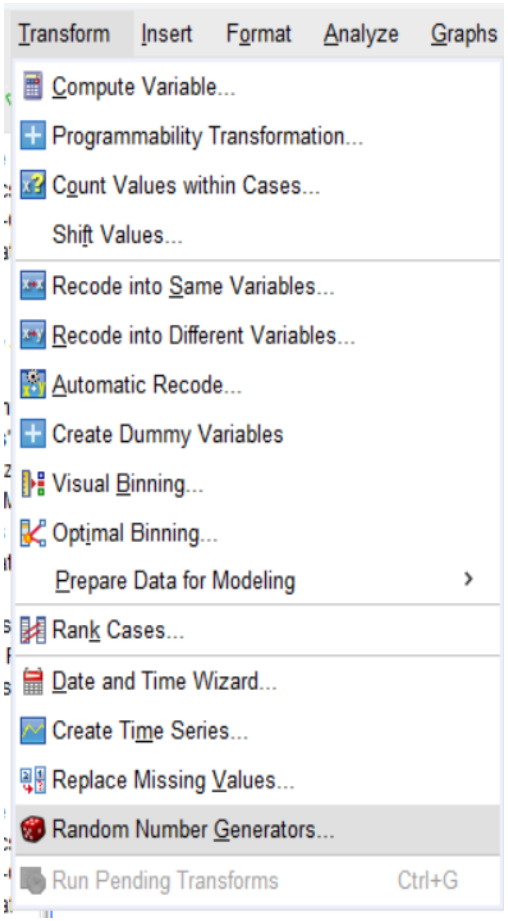
Naïve bayes

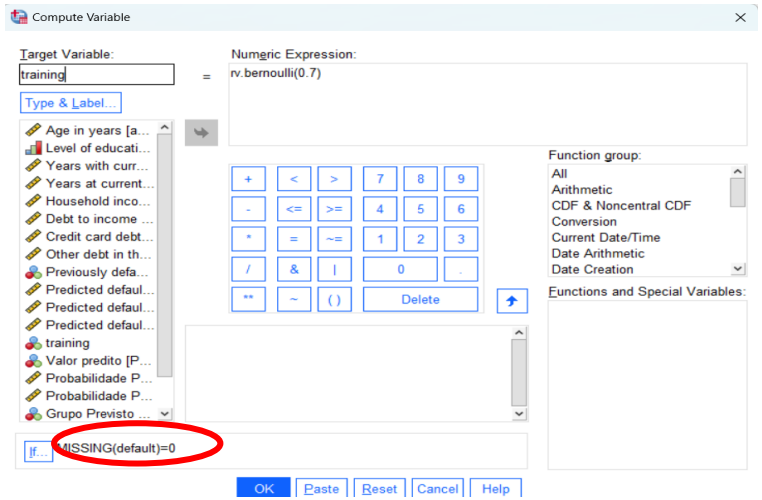

Naive Bayes
Variables
Output
Save
Options
Variables:
Sort: None
Age in years
Level of education
Years with current employer
Years at current address
Household income in thousands
Debt to income ratio (x100)
Credit card debt in thousands
Other debt in thousands
Dependent Variable:
Previously defaulted
Variable Specification Method:
Specify variables to exclude
Variables to Exclude:
Predicted default, model 1
Predicted default, model 2
Predicted default, model 3
Candidate Factors:
Candidate Covariates:
Forced Variables
Forced Factors:
Forced Covariates:
Predicted Value and Predicted Probabilities
Save predicted value
Save predicted probabilities of categories
Number of categories:
User-Missing Values for Categorical Variables
Exclude
Include
Training Sample
None
Percentage of cases to randomly assign to training sample
Percentage (%): 70
Time limit: 5.0
Maximum memory for storing training data (MB): 1024
Number of bins for scale predictors: 10.0
Predictor Selection
Select best subset from sequence of subsets
Automatically determine maximum subset size
Specify maximum subset size:
Criterion for best subset:
Test data
Select best subset of specified size
Size:
Use all specified predictors

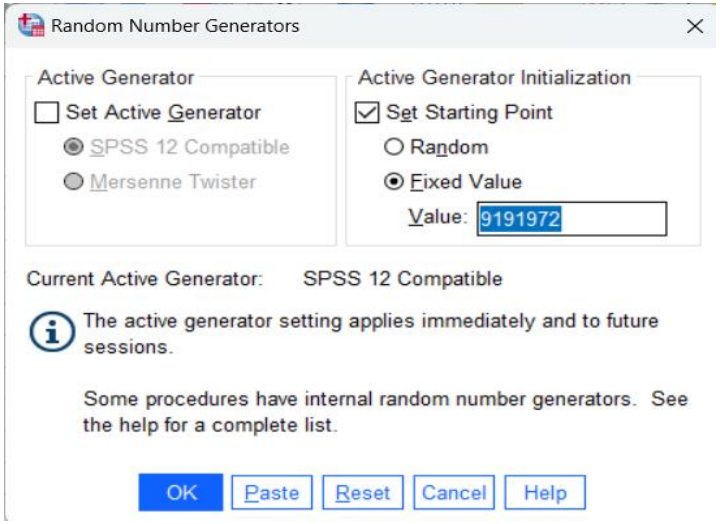
Select a training and test sample

We can also start by randomly selecting 70% of the valid sample. We will use the 30% for testing. But to guarantee replicability we must set the seed of the random number generator.

- This gives you more control over the samples.







Syntax

```
// First set the random seed and select about 70% of the cases for model building
SET SEED 9191972.
IF (MISSING(default)=0) training = rv.bernoulli(.7) .
EXECUTE .

// Naivebayes command with selected variables (exclude preddef1 preddef2 preddef3 training from predictors)
NAIVEBAYES default
  /EXCEPT VARIABLES=preddef1 preddef2 preddef3 training
  /TRAININGSAMPLE VARIABLE=training
  /SAVE PREDVAL PREDPROB.


// Produce a means table and crosstabulation to look at the distributions of predictors by PredictedValue

MEANS
  TABLES=employ address debtinc creddebt  BY PredictedValue
  /CELLS MEAN COUNT STDDEV  .

CROSSTABS
  /TABLES=ed  BY PredictedValue
  /FORMAT= AVALUE TABLES
  /CELLS= COUNT ROW
  /COUNT ROUND CELL .
```

NaiveBayes output

Case Processing Summary

		N	Percent
Previously defaulted	No	375	75,2%
	Yes	124	24,8%
Valid		499	100,0%
Excluded		0	
Total		499	

Selected Predictors

	Predictors
Categorical	ed
Scale	address creddebt debtinc employ

Subset Summary

Subset	Predictor Added	Rank	Test Data Criterion	Average Log-Likelihood
0	(Initial Subset)[a]			
1	Years with current employer	8	,620	-,486
2	Debt to income ratio (x100)	5	,493	-,425
3	Years at current address	3	,488	-,407
4	Credit card debt in thousands	2	,480	-,389
5	Level of education	1	,476	-,388
6	Other debt in thousands	4	,493	-,390
7	Household income in thousands	6	,540	-,390
8	Age in years	7	,578	-,407

a. The initial subset is empty.

Classification

		Predicted		
Sample	Observed	No	Yes	Percent Correct
Training	No	352	23	93,9%
	Yes	64	60	48,4%
	Overall Percent	83,4%	16,6%	82,6%
Test	No	133	9	93,7%
	Yes	35	24	40,7%
	Overall Percent	83,6%	16,4%	78,1%

Dependent Variable: Previously defaulted

How the predictors affect the model-predicted probability of response?

- **Analyze** > **Compare Means** > **Means...**

How the predictors affect the model-predicted probability of response?

Case Processing Summary

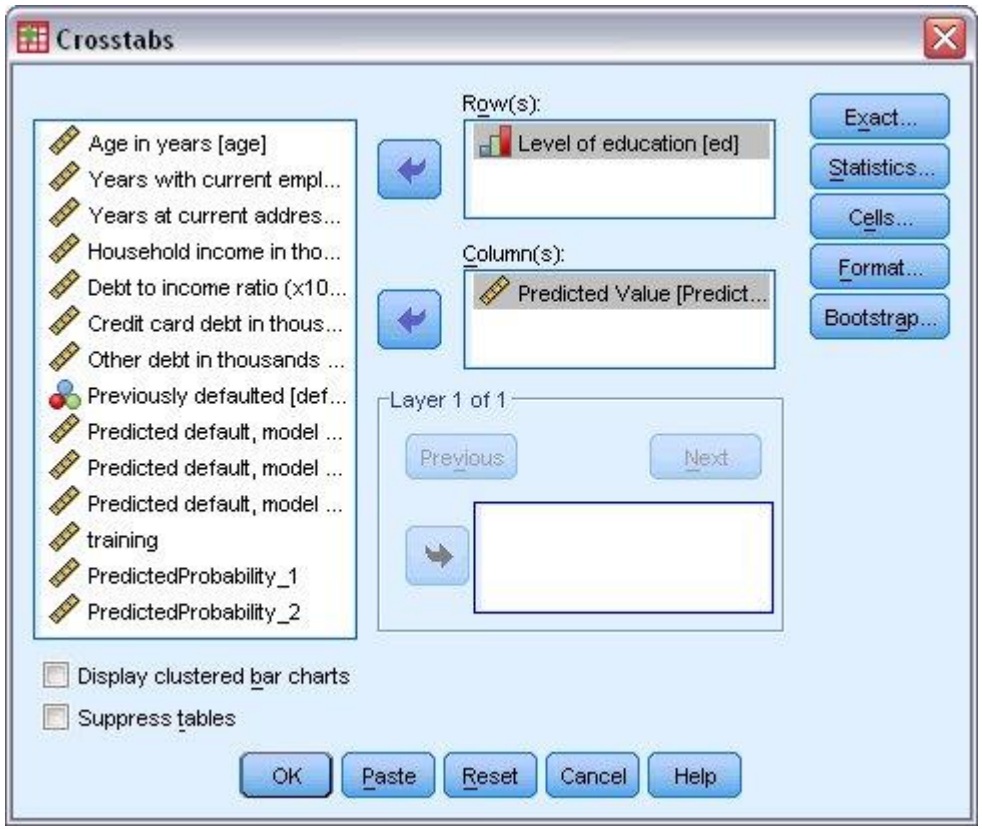
	Cases					
	Included		Excluded		Total	
	N	Percent	N	Percent	N	Percent
Years with current employer * Predicted Value	850	100,0%	0	0,0%	850	100,0%
Years at current address * Predicted Value	850	100,0%	0	0,0%	850	100,0%
Debt to income ratio (x100) * Predicted Value	850	100,0%	0	0,0%	850	100,0%
Credit card debt in thousands * Predicted Value	850	100,0%	0	0,0%	850	100,0%

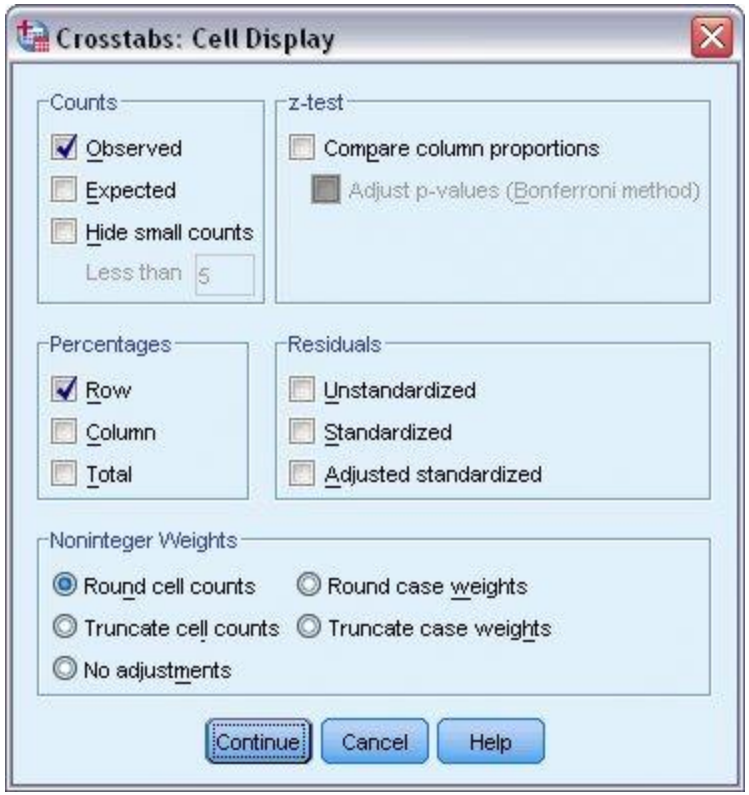
Report

Predicted Value		Years with current employer	Years at current address	Debt to income ratio (x100)	Credit card debt in thousands
0	Mean	9,38	8,97	8,7620	1,3264
	N	716	716	716	716
	Std. Deviation	6,566	6,977	5,61473	1,55796
1	Mean	4,22	5,16	17,7037	2,9149
	N	134	134	134	134
	Std. Deviation	6,236	5,432	7,13338	3,69567
Total	Mean	8,57	8,37	10,1716	1,5768
	N	850	850	850	850
	Std. Deviation	6,778	6,895	6,71944	2,12584

How the predictors affect the model-predicted probability of response?

- **Analyze > Descriptive Statistics > Crosstabs...**





IBM SPSS

Lisbon School of Economics & Management

Crosstabs

Case Processing Summary

	Cases					
	Valid		Missing		Total	
	N	Percent	N	Percent	N	Percent
Level of education * Predicted Value	850	100,0%	0	0,0%	850	100,0%

Level of education * Predicted Value Crosstabulation

			Predicted Value		
			0	1	Total
Level of education	Did not complete high school	Count	422	38	460
		% within Level of education	91,7%	8,3%	100,0%
	High school degree	Count	193	42	235
		% within Level of education	82,1%	17,9%	100,0%
	Some college	Count	69	32	101
		% within Level of education	68,3%	31,7%	100,0%
	College degree	Count	27	22	49
		% within Level of education	55,1%	44,9%	100,0%
	Post-undergraduate degree	Count	5	0	5
		% within Level of education	100,0%	0,0%	100,0%
Total		Count	716	134	850
		% within Level of education	84,2%	15,8%	100,0%

Discriminant Analysis

Let us return to the derivation with one explanatory variable (k=1) and impose a normal conditional distribution (and assume equal co-variances for all K groups):

$$\Pr(Y=k|x)=\frac{P(Y=k)*P(x|Y=k)}{P(x)}=\frac{P(Y=k)*\left(\frac{1}{\sqrt{2\pi\sigma}}\right)e^{-\frac{1}{2}(x-\mu(\mathrm{k}))/\sigma^2}}{\sum_{l=1}^{K}P(Y=l)*\left(\frac{1}{\sqrt{2\pi\sigma}}\right)e^{-\frac{1}{2}(x-\mu(\mathrm{l}))/\sigma^2}}$$

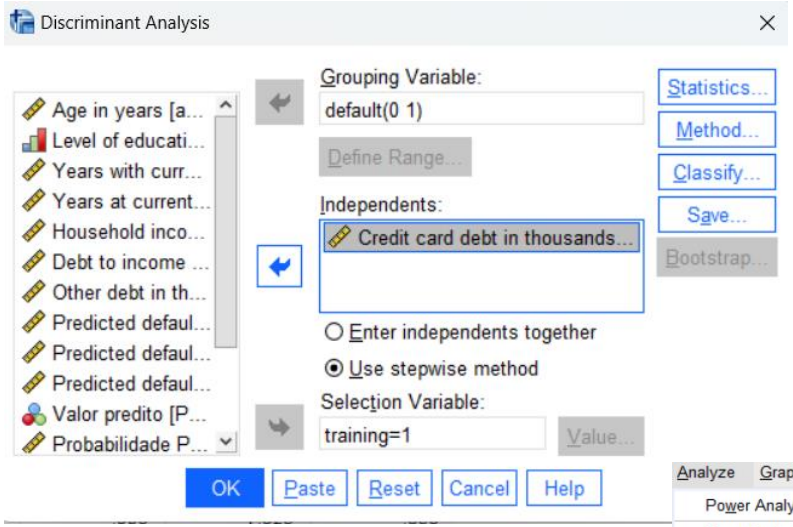
Or the ***discriminant score*** becomes:

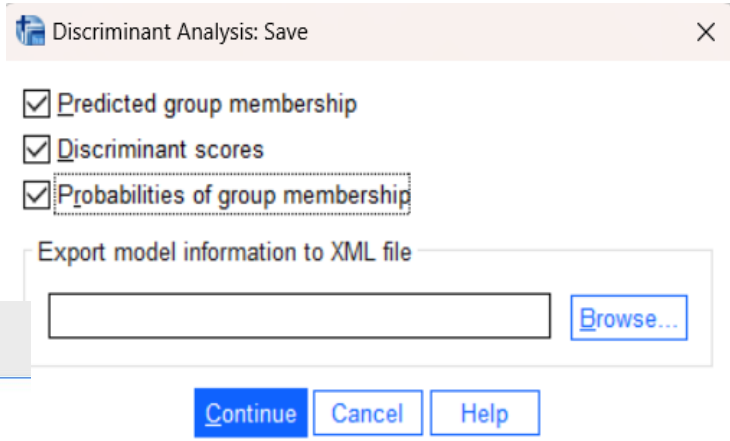
$$\delta_k(x)=\frac{[x-\mu(\mathrm{k})]^2}{\sigma^2}+\ln(P(Y=k))$$

- This means that we can "estimate" the probabilities (the scores) by simply plugging the mean and variance estimates.

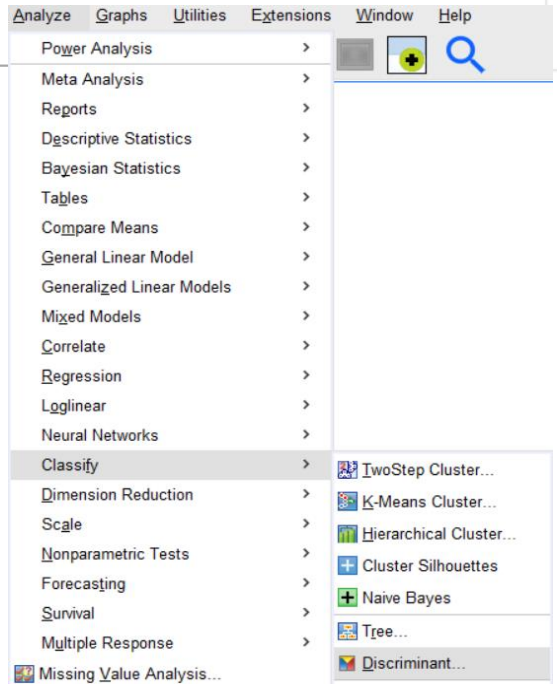
- Each observation is then classified as one of k possibilities according the the highest score.

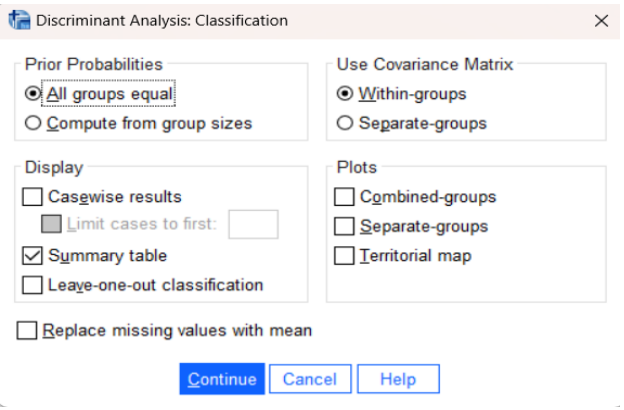
Running analysis

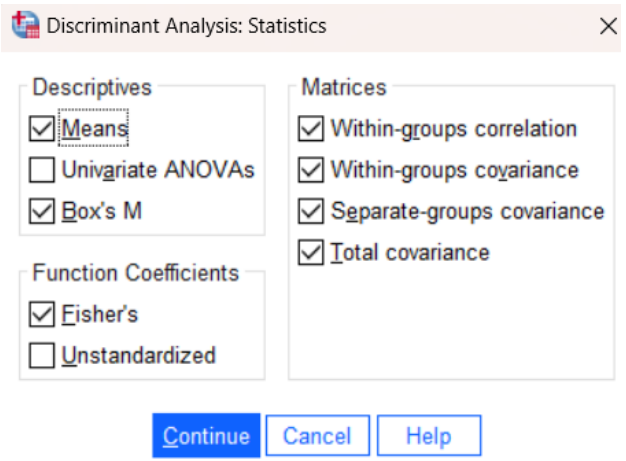




We use just one variable as an illustration (p=1).







Descriptives and assumptions

Analysis Case Processing Summary

Unweighted Cases		N	Percent
Valid		499	58.7
Excluded	Missing or out-of-range group codes	150	17.6
	At least one missing discriminating variable	0	.0
	Both missing or out-of-range group codes and at least one missing discriminating variable	0	.0
	Unselected	201	23.6
	Total	351	41.3
Total		850	100.0

Group Statistics

Previously defaulted		Mean	Std. Deviation	Valid N (listwise) Unweighted	Weighted
No	Credit card debt in thousands	1.2554	1.41769	375	375.000
Yes	Credit card debt in thousands	2.3656	3.36732	124	124.000
Total	Credit card debt in thousands	1.5313	2.13087	499	499.000

Different variances. We can actually test this selecting Box's M in statistics.

Test Results

Box's M		167.333
F	Approx.	166.924
	df1	1
	df2	349307.254
	Sig.	<.001

Tests null hypothesis of equal population covariance matrices.

Covariance Matrices[a]

.Previously defaulted		Credit card debt in thousands
No	Credit card debt in thousands	2.010
Yes	Credit card debt in thousands	11.339
Total	Credit card debt in thousands	4.541

a. The total covariance matrix has 498 degrees of freedom.

Model fit

Eigenvalues

Function	Eigenvalue	% of Variance	Cumulative %	Canonical Correlation
1	.054[a]	100.0	100.0	.225

a. First 1 canonical discriminant functions were used in the analysis.

Wilks' Lambda

Test of Function(s)	Wilks' Lambda	Chi-square	df	Sig.
1	.949	25.884	1	<.001

Results

Classification Function Coefficients

	Previously defaulted	
	No	Yes
Credit card debt in thousands	.291	.548
(Constant)	-.876	-1.341

Fisher's linear discriminant functions

Classification Results[a,b]

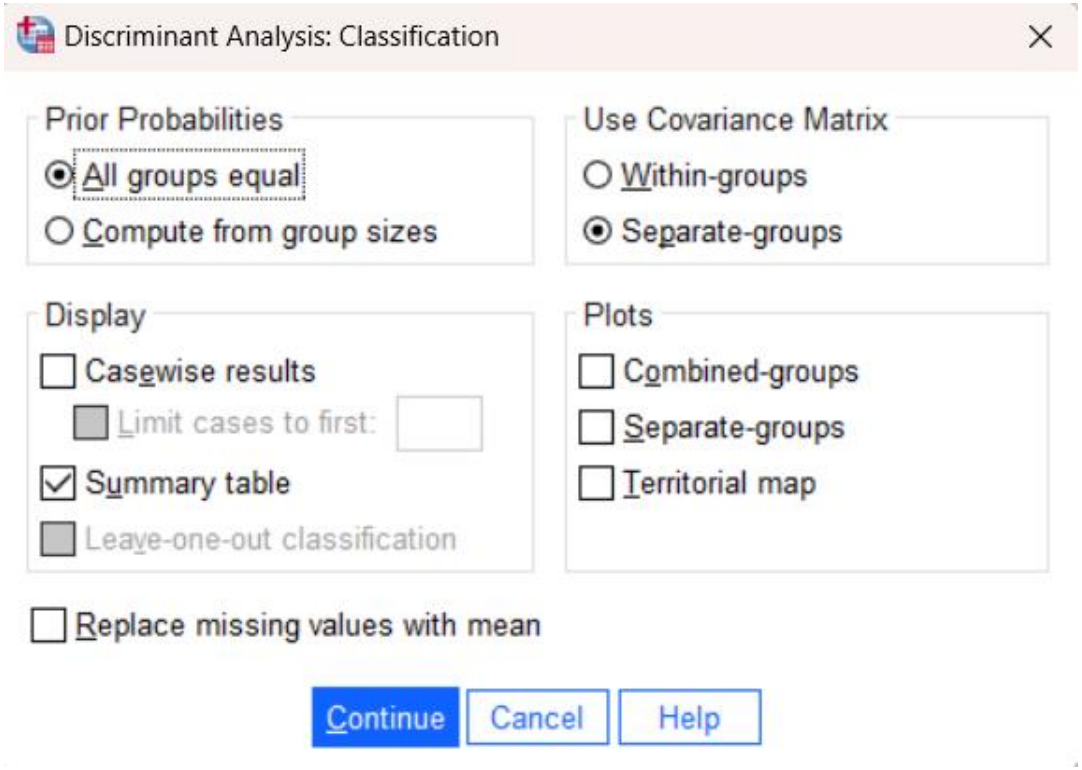
				Predicted Group Membership		
			Previously defaulted	No	Yes	Total
Cases Selected	Original	Count	No	297	78	375
			Yes	76	48	124
		%	No	79.2	20.8	100.0
			Yes	61.3	38.7	100.0
Cases Not Selected	Original	Count	No	107	35	142
			Yes	35	24	59
			Ungrouped cases	109	41	150
		%	No	75.4	24.6	100.0
			Yes	59.3	40.7	100.0
			Ungrouped cases	72.7	27.3	100.0

a. 69.1% of selected original grouped cases correctly classified.

b. 65.2% of unselected original grouped cases correctly classified.

Rerun

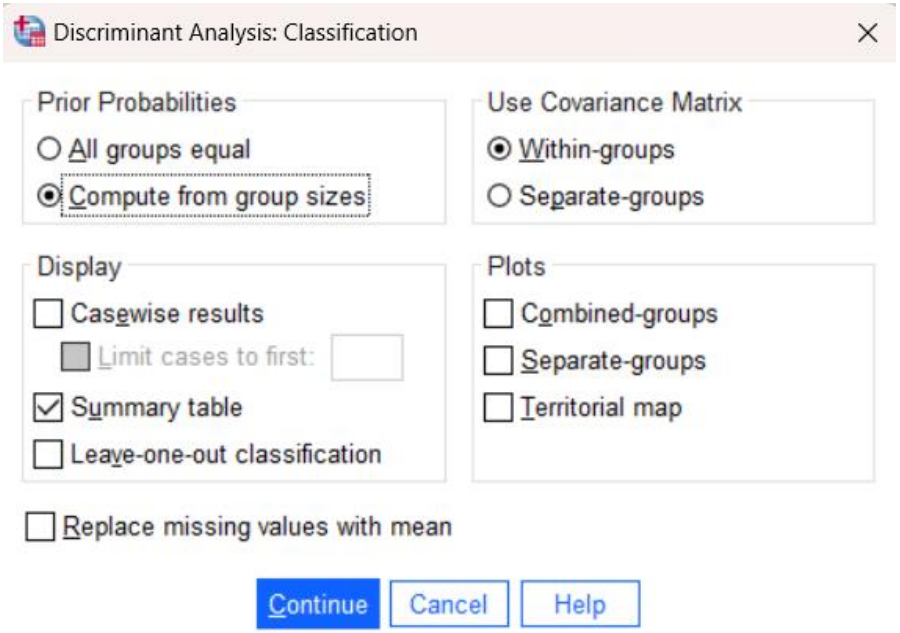
Given the rejection of equal covariances, rerun the analysis without this assumption. Note that when there are many variables this implies estimating a lot more parameters (p*(p-1)/2)
What do you conclude about your classification?



Reminder: Three main Assumptions.

1. Cases should be independent.
2. Predictor variables should have a multivariate normal distribution, and
3. Within-group variance-covariance matrices should be equal across groups.

Note



We assumed "uninformative" prior probabilities (i.e. uniform distribution). However, if we believe that the group sizes are according to the actual probabilities we should observe in the data, we can use these probabilities.

Questions

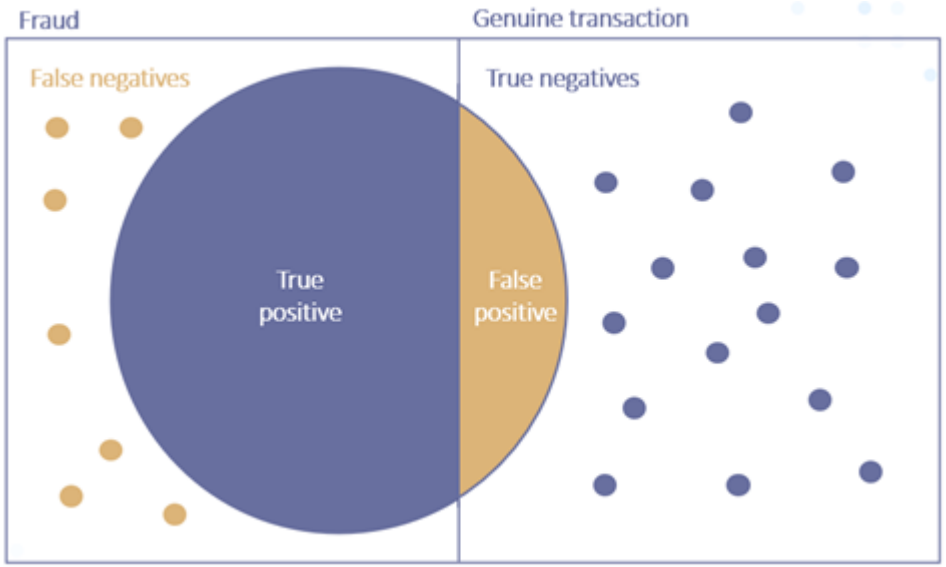
- Redo the analysis with all the explanatory variables. Run separate the baseline case with equal variances and priors and different variances and priors.

- Now cross-tab the predictions of LDA and NB. Which model predicts best? Can you explain the differences?

- What if you look only for the cross-tabs in the subsample not used for training?

Measuring fit

- Imagine we obtain the results below
- There is 92% accuracy.
- Is this a good fit?



Predicted stolen?

	Yes	No
Truth Yes	2	5
No	3	90

Measuring fit

- Accuracy
- Precision
- Recall
- F1 score

There is more than one measure of fit.

- **Accuracy** is a measure of overall fit:

#correct cases/#total cases=92/100=92%

- **Precision** measures the fraction of predicted yes that were correct:

correct yes/(#correct yes+# incorrect yes)=2/(2+3)=40%

- **Recall** measures the fraction of yes that were correctly predicted:

correct yes/(#correct yes+# incorrect no)=2/(2+5)=28.5%

- **F1 score** is the harmonic mean of precision and recall

2*precision*recall/(precision+recall)=33

Think

- What is better? A model with 90% accuracy and an F1 score of 40 or a model with 75% accuracy and an F1 score of 60?

Balanced accuracy for imbalanced data

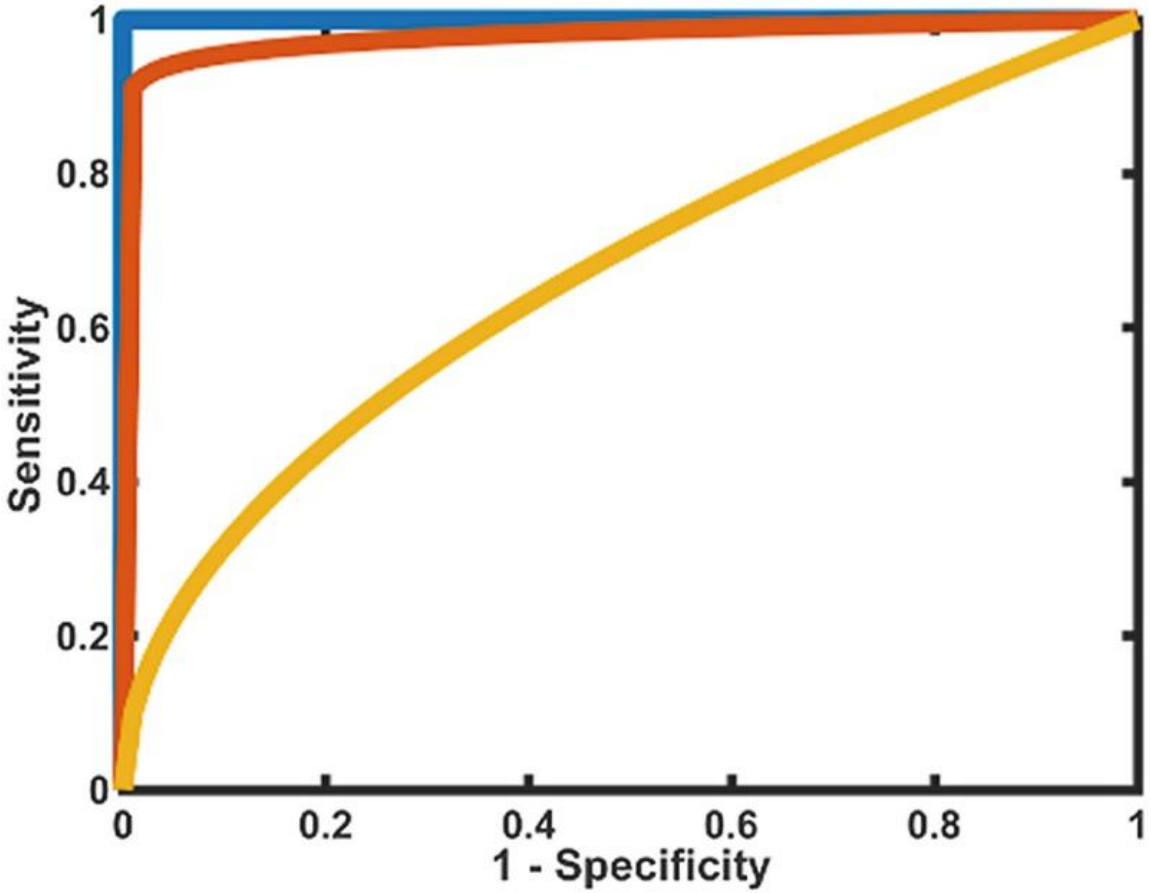
- If the data is very imbalanced, imagine you are trying to predict credit card fraud that typically happens less than 1 out of 1000 times, a model that predicts no fraud will have an accuracy of 99.9%.
- In these cases prediction and recall may be useful to look at.
- Alternatively, we can use balanced accuracy
- Balanced accuracy=(sensitivity+specificity)/2=62.7%

Where

- Sensitivity=true positive rate (recall)=2/(2+5)=28.5%
- Specificity=true negative rate=90/(90+3)=96.7%

Measuring fit

- ROC (receiver operating characteristic) curve



- Sensitivity (true positive rate)
- Specificity: true negative rate

The higher the (AUC) area under the (ROC) curve, the better.

Max AUC is 1, while 0.5 is a random classifier.

Let the price of car be Y and the characteristics of cars be X. To predict Y as a function of X:

A. We can use an LDA model.

B. We need to use na alternative model.

Table Lookup Models

Naïve Bayes + LDA

(Multiple) Linear Regression

(Multiple) Logistic Regression

Multiple Linear Regression

REVISION

ANY QUESTIONS?

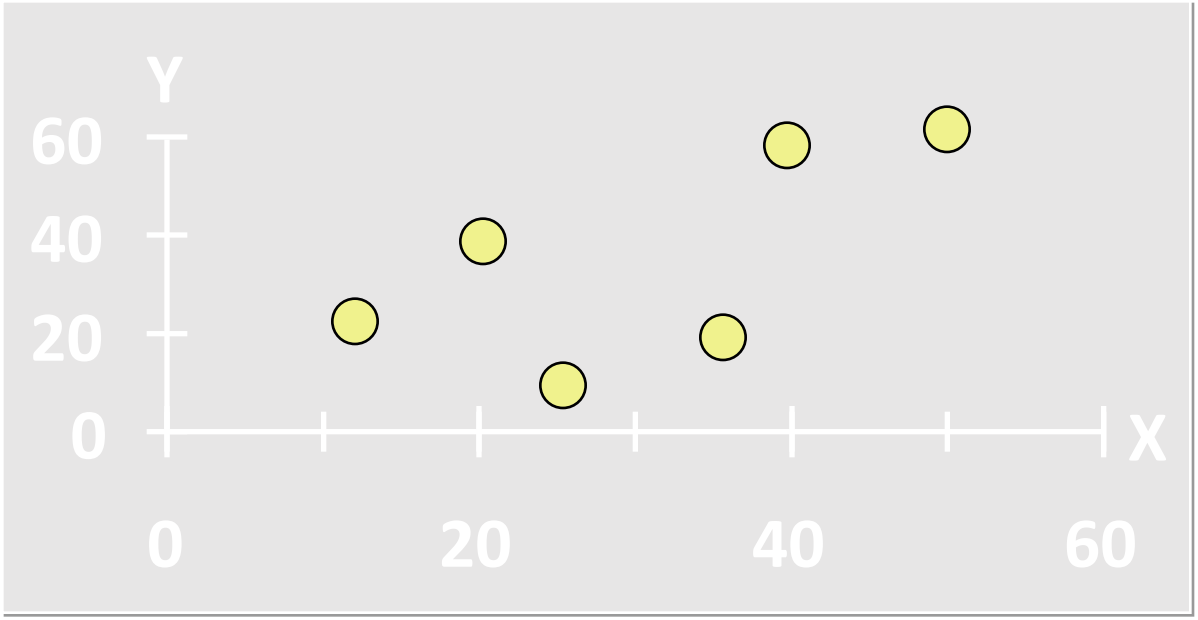
REVISION: MULTIPLE LINEAR REGRESSION

Target variable is continuous (different from previous cases)

Objective is to get a best prediction for the outcome variable

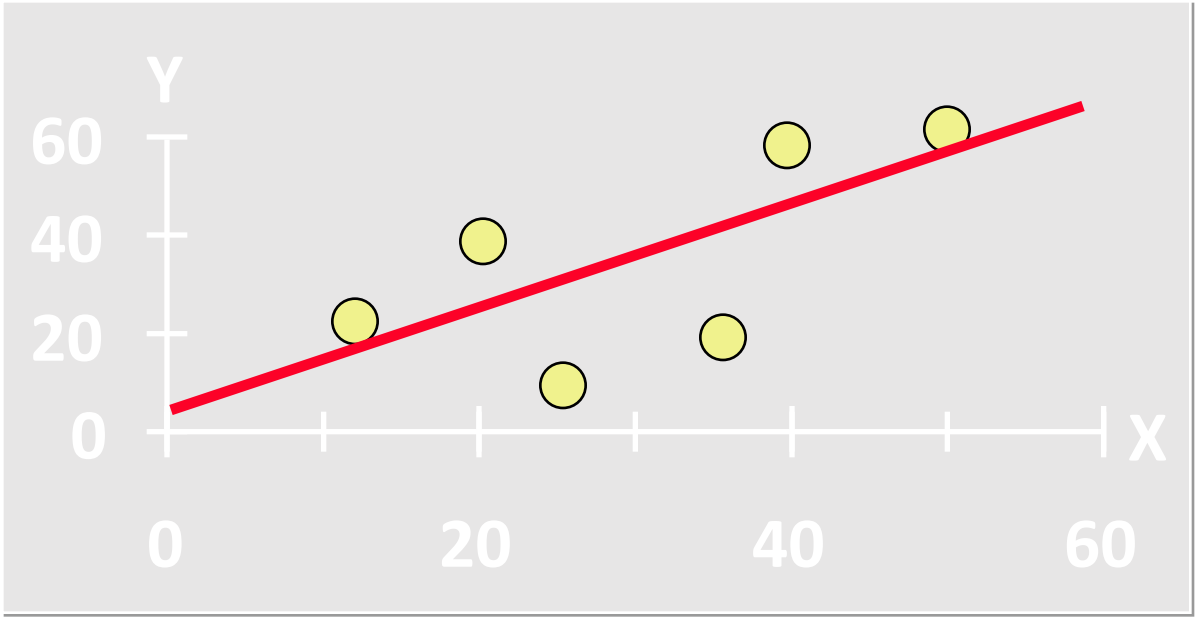
Example:

1. Plot of All (X_i, Y_i) Pairs
2. Suggests How Well Model Will Fit



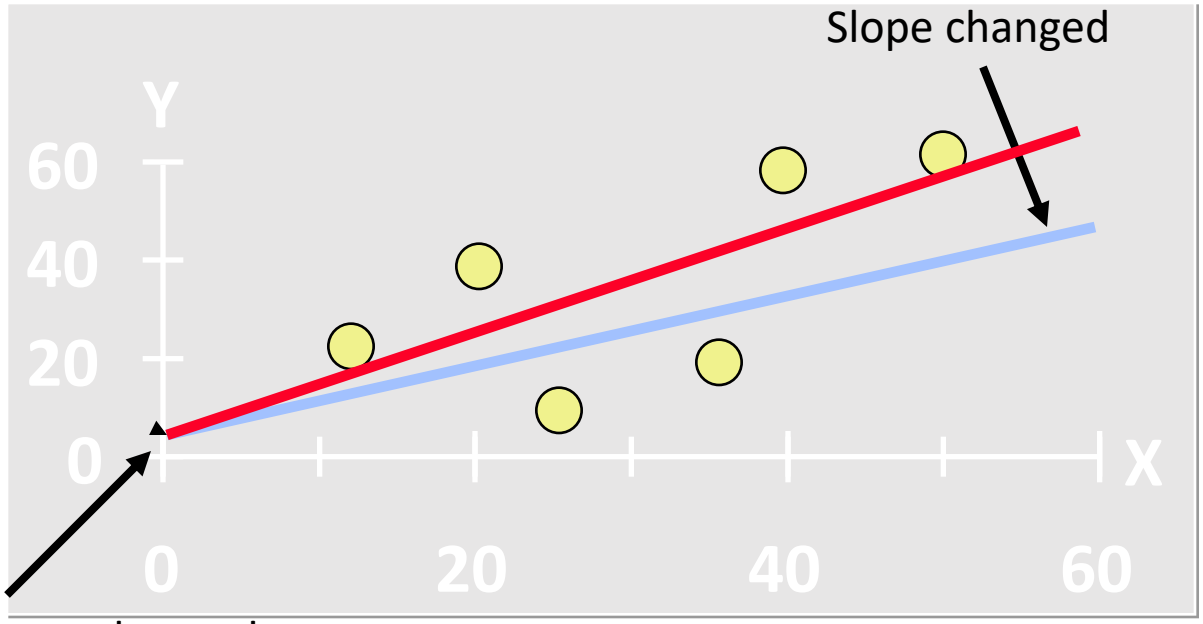
Revision: Linear regression

How would you draw a line through the points? How do you determine which line 'fits best'?



Revision: Linear regression

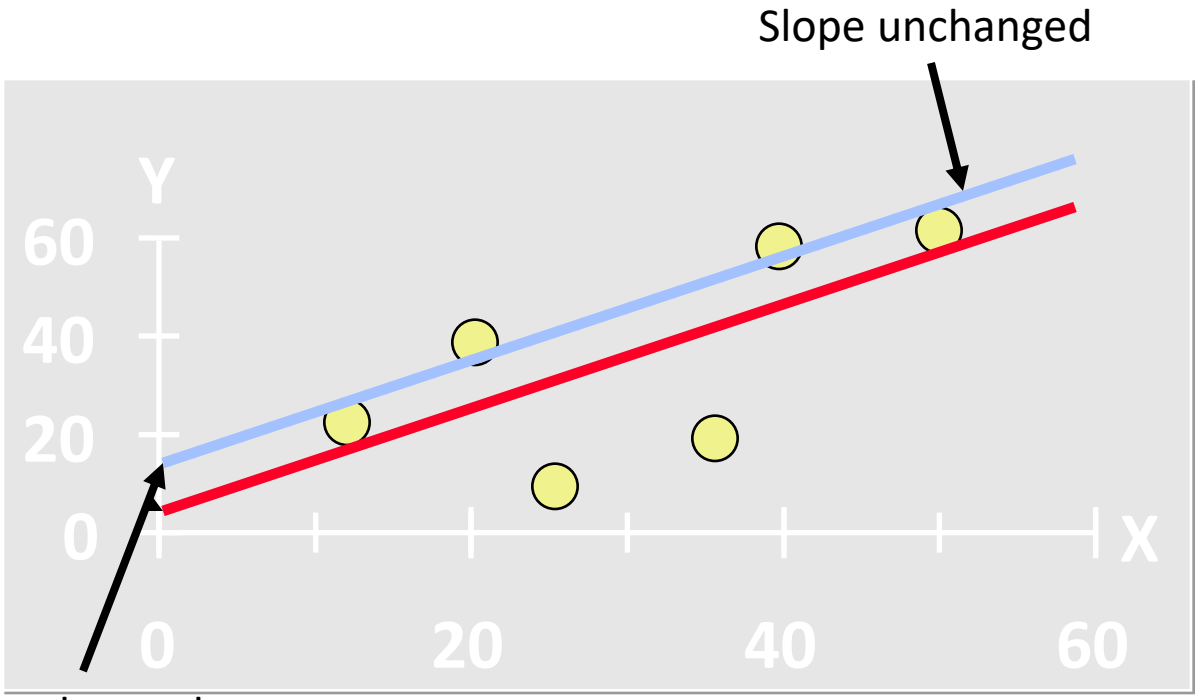
How would you draw a line through the points? How do you determine which line 'fits best'?



Intercept unchanged

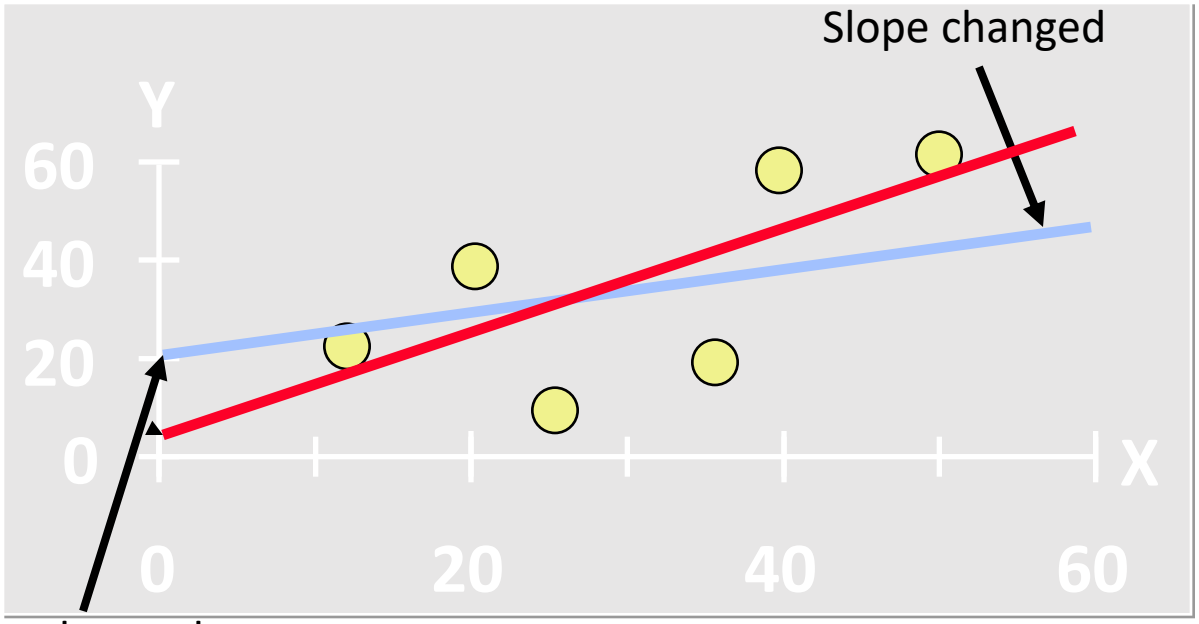
Revision: Linear regression

How would you draw a line through the points? How do you determine which line 'fits best'?



Revision: Linear regression

How would you draw a line through the points? How do you determine which line 'fits best'?



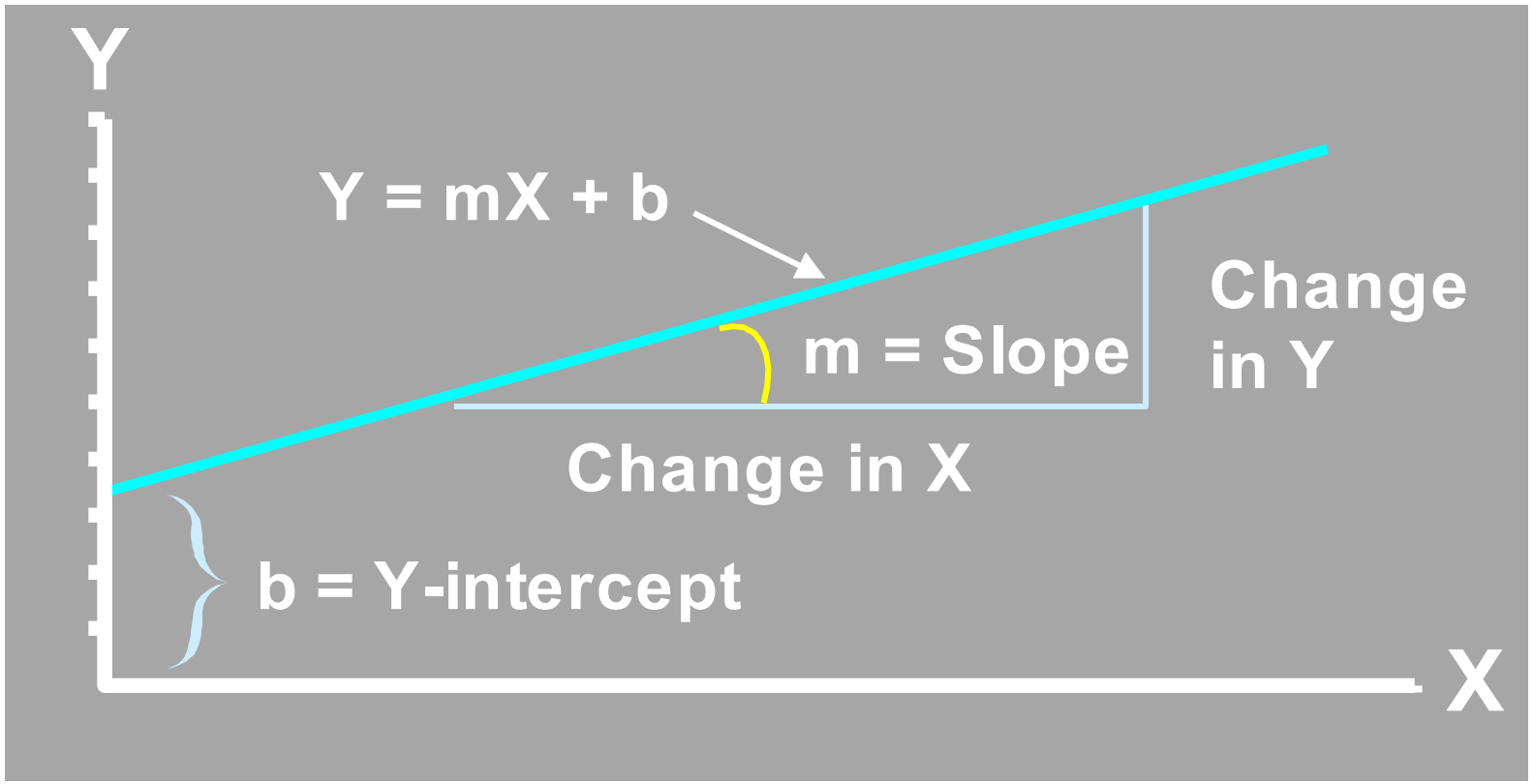
Revision: Least Squares

1. ‘Best Fit’ Means Difference Between Actual Y Values & Predicted Y Values Are a Minimum. *But* Positive Differences Off-Set Negative ones.

2. How good is the fit? R^2 measures the % of the variation in Y explained by the variation in X. It varies from 0 to 1.

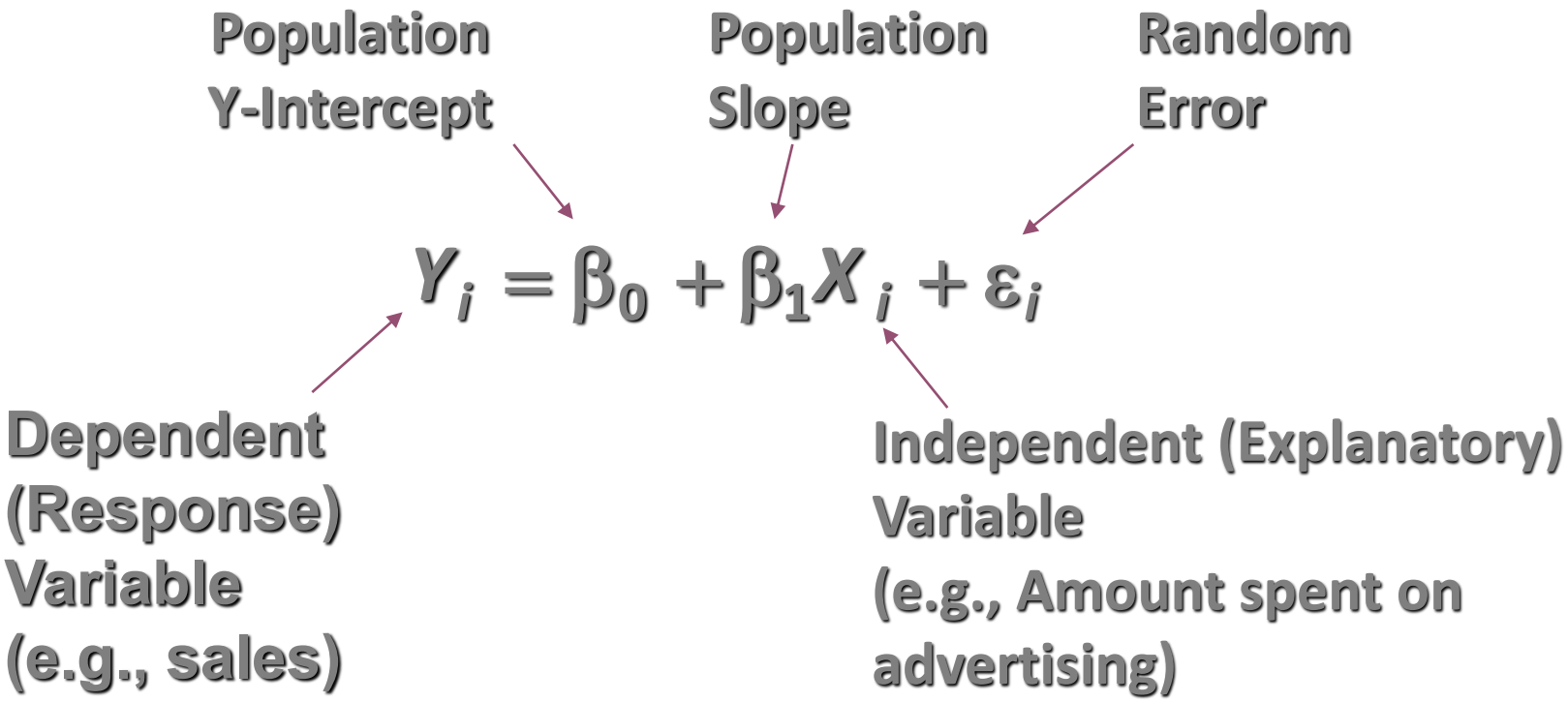
Revision: Linear regression

Linear Equation



Revision: Linear Regression Model

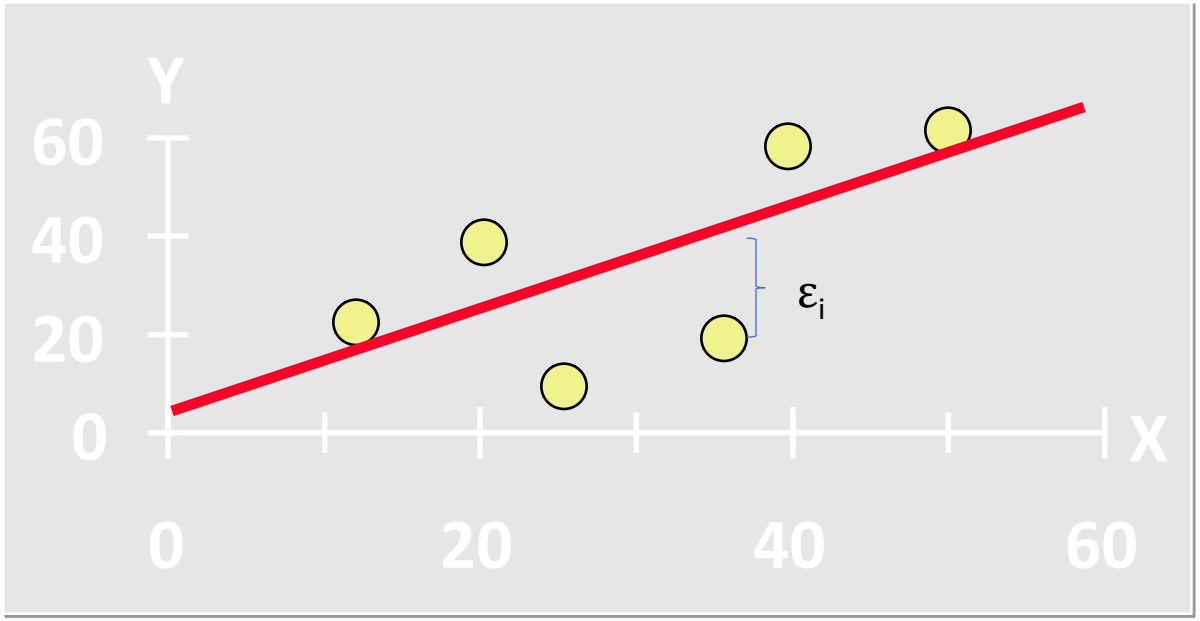
Relationship Between Variables Is a Linear Function.



Revision: Least squares

Least squares minimizes **vertical** (squared) distances.

- i.e. Min $\sum_{i=1}^{N} \varepsilon_i^2$



Revision: Multiple case

- More than two variables cannot be presented graphically (with two we can still have a 3D plot..).

- Equation becomes:

$$Y_i = \beta_0 + \beta_1 X_{1i} + \ldots + \beta_k X_{ki} + \varepsilon_i$$

- Ys and Xs are known (so take them as constants) and we want to learn (infer) the βs.

Table Lookup Models

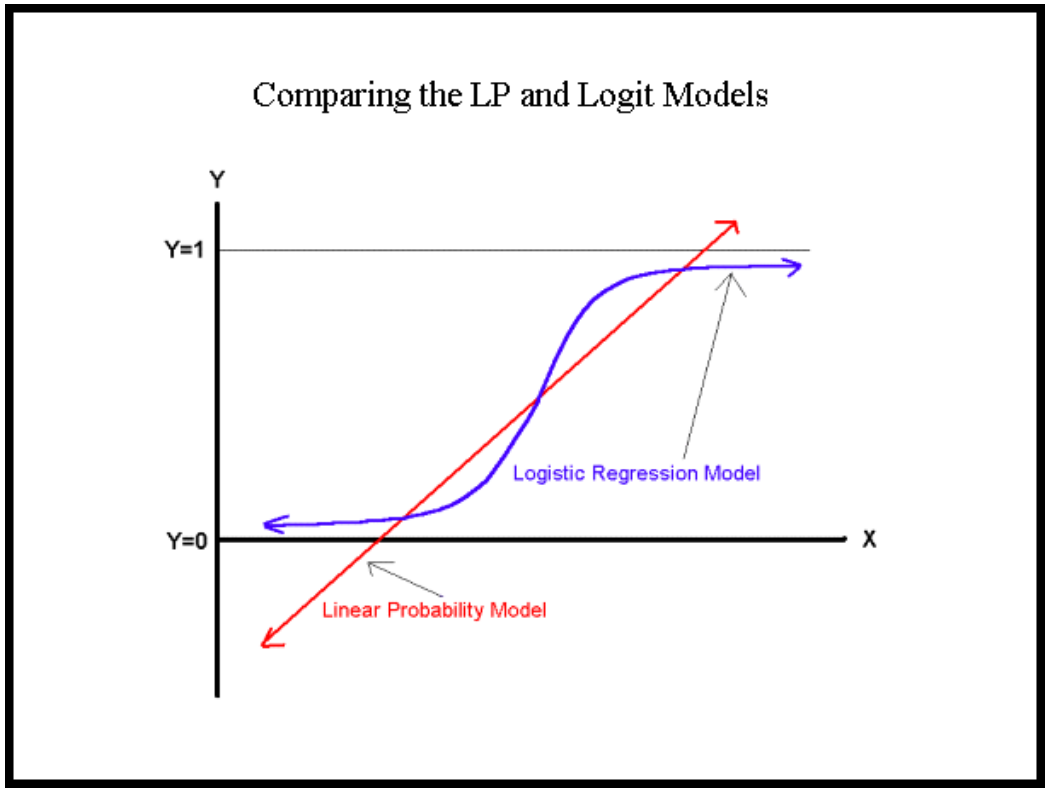
Naïve Bayes + LDA

(Multiple) Linear Regression

(Multiple) Logistic Regression

Logistic regression

- Similar to multiple regression, except that the dependent variable is categorical.
- Note: Linear regression applied to binary variables is called a linear probability model



- Predicted values above 1 and below 0?
- Logistic model transforms linear into an S-shaped function always between 0 and 1 (logistic function).

Logistic regression

- Transform probabilities to log-odds.
- Fit a linear regression model to log-odds.

$$\ln\left(\frac{p_i}{1-pi}\right) = \beta_0 + \beta_1 X_{1i} + \ldots + \beta_k X_{ki}$$

Or

$$pi = \frac{1}{1 + \exp -(\beta_0 + \beta_1 X_{1i} + \ldots + \beta_k X_{ki})}$$

- **Maximum likelihood estimation *vs.* least squares estimation:** The principle for fitting the curve is no longer minimizing the residuals (as was for linear regression) but instead it is maximizing the likelihood of having observed the given values.

DIY – Model vehicle sales

Two exercises

1. Redo the analysis in Naïve Bayes and LDA with Logistic regression

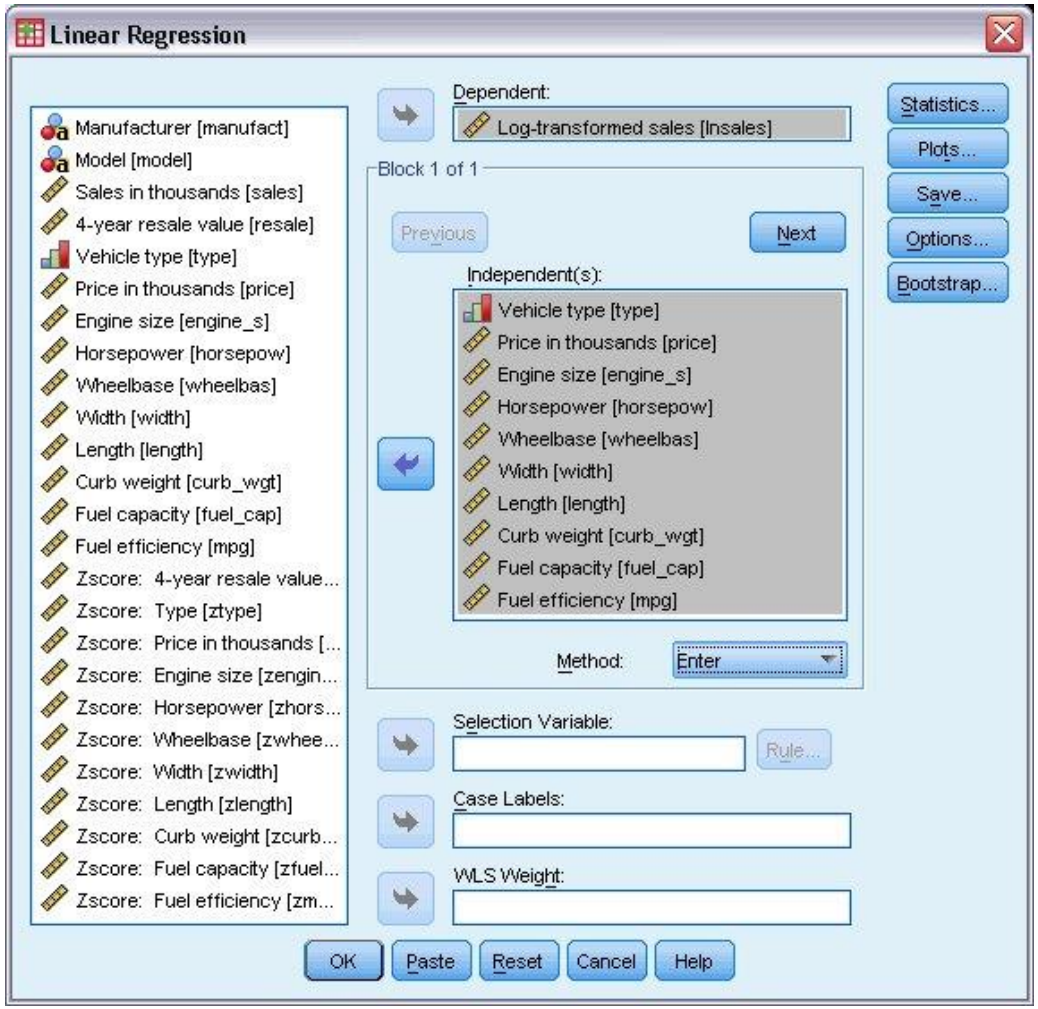
2. Objective: Modeling vehicle sales.

   Data file: ch03_car_sales.sav

   1. Why do we use log sales as dependent variable?
   2. Why is multicollinearity a concern?
   3. How do you select which variables to include?

DIY – Model vehicle sales

Analyze > Regression > Linear...



IBM SPSS

Results

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	130.300	10	13.030	13.305	.000[a]
	Residual	138.082	141	.979		
	Total	268.383	151			

a. Predictors: (Constant), Fuel efficiency, Length, Price in thousands, Vehicle type, Width, Engine size, Fuel capacity, Wheelbase, Curb weight, Horsepower

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	.697[a]	.486	.449	.98960

a. Predictors: (Constant), Fuel efficiency, Length, Price in thousands, Vehicle type, Width, Engine size, Fuel capacity, Wheelbase, Curb weight, Horsepower

		Statistics				
		Unstandardized Coefficients		Standardized Coefficients		
Model	Variables	B	Std. Error	Beta	t	Sig.
1	(Constant)	-3.017	2.741		-1.101	.273
	Vehicle type	.883	.331	.293	2.670	.008
	Price in thousands	-.046	.013	-.502	-3.596	.000
	Engine size	.356	.190	.281	1.871	.063
	Horsepower	-.002	.004	-.092	-.509	.611
	Wheelbase	.042	.023	.241	1.785	.076
	Width	-.028	.042	-.073	-.676	.500
	Length	.015	.014	.148	1.032	.304
	Curb weight	.156	.350	.075	.447	.655
	Fuel capacity	-.057	.047	-.167	-1.203	.231
	Fuel efficiency	.081	.040	.262	2.023	.045

Results

Simple correlation

Conditional correlations

Model		Correlations			Collinearity Statistics	
		Zero-order	Partial	Part	Tolerance	VIF
1	Vehicle type	.274	.219	.161	.304	3.293
	Price in thousands	-.552	-.290	-.217	.187	5.337
	Engine size	-.135	.156	.113	.162	6.159
	Horsepower	-.389	-.043	-.031	.112	8.896
	Wheelbase	.292	.149	.108	.200	4.997
	Width	.037	-.057	-.041	.313	3.193
	Length	.215	.087	.062	.178	5.605
	Curb weight	-.041	.038	.027	.131	7.644
	Fuel capacity	-.016	-.101	-.073	.189	5.303
	Fuel efficiency	.121	.168	.122	.217	4.604

Tolerance is the percentage of the variance in a given predictor that cannot be explained by the other predictors. Close to zero means multicollinearity.

VIF – Variance inflated factors: Above 2 shows signs of multicollinearity.

Correlations – See next slide

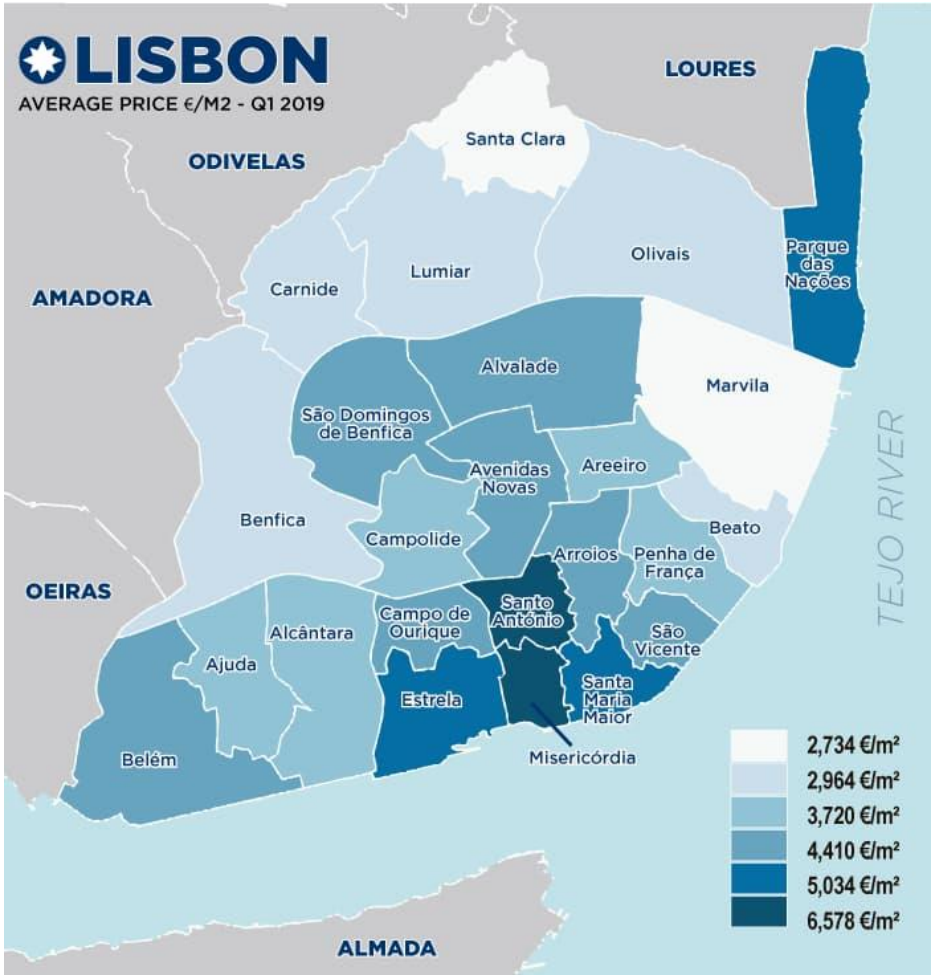
Model	Dimension	Eigenvalue	Condition Index
1	1	9.920	1.000
	2	.733	3.678
	3	.259	6.193
	4	.050	14.051
	5	.019	22.589
	6	.008	35.942
	7	.005	44.275
	8	.003	58.480
	9	.002	76.175
	10	.001	130.747
	11	.000	148.267

Collinearity diagnostics
Several eigenvalues are close to 0, indicating that the predictors are highly intercorrelated and that small changes in the data values may lead to large changes in the estimates of the coefficients.

Different correlations

- **Zero-order correlation** is the simple correlation between dependente and independente variable
- **Partial correlation** is the correlation between the dependent and independente variable conditional on all the remaining independente variables
- **Part correlation** is the correlation between the dependente variable and the independente variable conditional on all the remaining variables (only the independente variable is conditioned upon). The primary reason for conducting the part correlation would be to see how much unique variance the independent variable explains in relation to the total variance in the dependent variable, rather than just the variance unaccounted for by the control variables.

House prices



House prices depend on location latitude (lat) and longitude (lon). John capture this in a linear model of the form:

$$Y_i = \beta_0 + \beta_1 \text{lat}_i + \beta_2 \text{lon}_i + \varepsilon_i$$

A. True
B. False

Modern techniques for model building

- Modern regression: ridge
- Modern regression: lasso
- Prediction trees
- Artificial Neural Networks
- Others (e.g. SVM)

Ridge regression

Modern regression

House characteristics

- Location
- Area
- Typology
- Construction year
- Bathrooms
- Energy certificate
- Central heating
- AC
- Garage

- Garden
- Fireplace
- Gatted community
- Equipped kitchen
- Storage
- Swimming pool
- Suite
- Terrace
- Balcony
- Security
- Sea View

As we add more characteristics, the prediction error of our model should:

A. Increase
B. Decrease
C. Stay the same

Ridge regression

Reminder: shortcomings of linear regression

1. **Predictive ability**: We can decompose prediction error into squared bias and variance. Linear regression has low bias (zero bias) but suffers from high variance. So it may be worth sacrificing some bias to achieve a lower variance.

2. **Interpretative ability**: with a large number of predictors, it can be helpful to identify a smaller subset of important variables. Linear regression doesn't do this.

Also: linear regression is not defined when p > n

General setup

Given fixed covariates $x_i \in \Re^p, i = 1,..,n$ we formalize the model

$$y_i = f(x_i) + \varepsilon_i, i = 1,..,n$$

where the function is unknown (could be linear).

Our data contains information on (y,x).

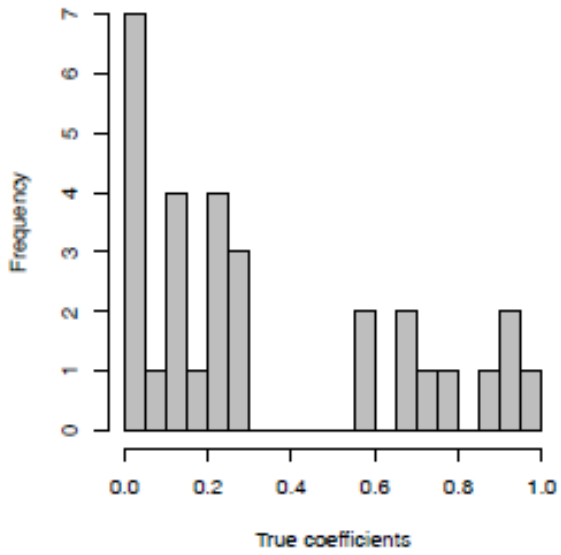
This setup is valid for any dependence technique: regression, decision trees, artificial neural networks, etc..

Example: subset of small coefficients

Let n = 50, p = 30, and σ^2=1. The true model is linear with 10 large coefficients (between 0.5 and 1) and 20 small ones (between 0 and 0.3).

Note:

Prediction error=True noise+Bias^2+Variance of estimates



The linear regression fit:

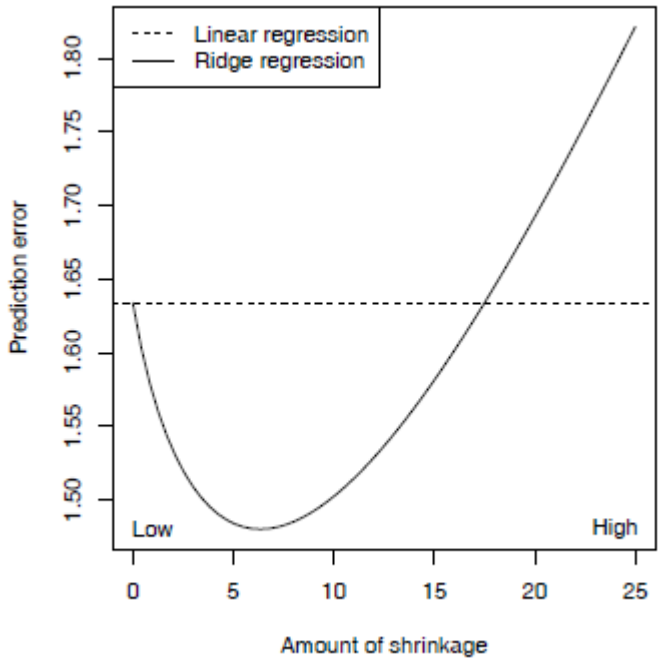
Squared bias ≈ 0.006

Variance ≈ 0.627

Pred. error ≈ 1+0.006+0.627=1.633

Question: Can we do better by shrinking the coefficients to reduce variance?

Example: subset of small coefficients



Basic idea of Ridge: More complex models can have small bias but have larger variance. Simplify the model by penalizing complexity and reduce variance of the estimated coefficients. [Limit when all coefficients equal 0].

The linear regression:
Squared bias ≈ 0.006
Variance ≈ 0.627
Pred. error ≈ 1+0.006+0.627=1.633

Ridge regression:
Squared bias ≈ 0.077
Variance ≈ 0.403
Pred. error ≈ 1+0.077+0.403=**1.48**

Ridge regression

Ridge regression is like least squares but shrinks the estimated coefficients towards zero. Given a response vector y and a predictor matrix X, the ridge regression coefficients are obtained by minimizing:

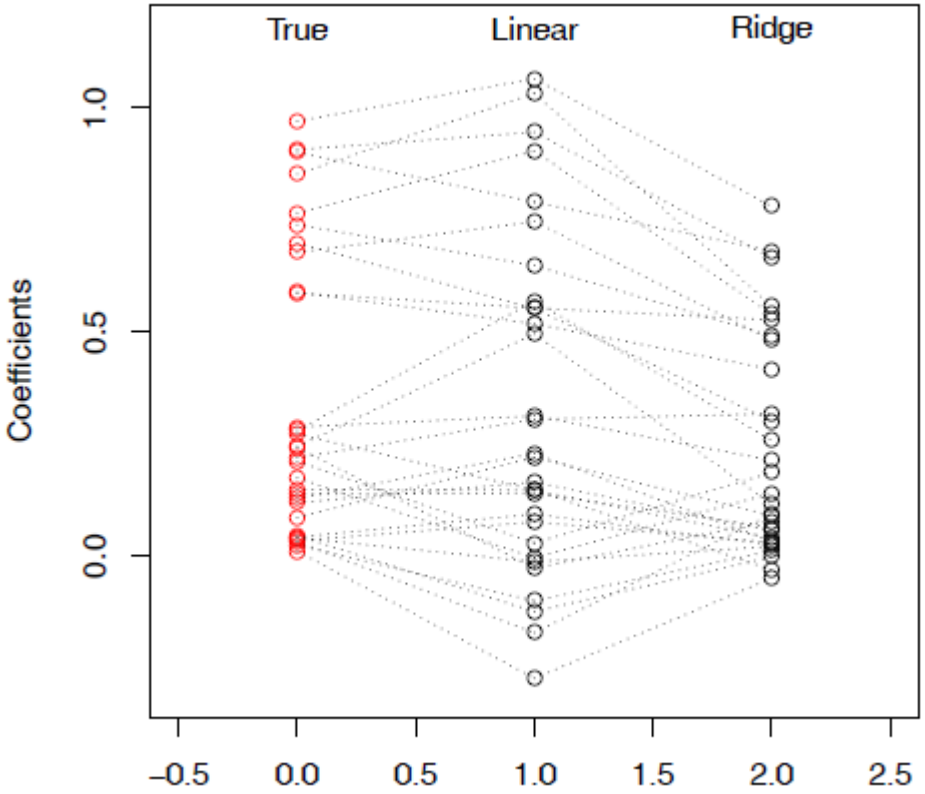
$$\underbrace{\sum_{i=1}^{n}(y_i - x'_i\beta)^2}_{Loss} + \lambda \underbrace{\sum_{j=1}^{p}\beta_j^2}_{Penalty}$$

Here $\lambda \geq 0$ is a tuning parameter, which controls the strength of the penalty term. Note that:

- When $\lambda = 0$, we get standard linear regression.
- When $\lambda = \infty$, we get $\beta = 0$.
- For λ in between, we are balancing two ideas: fitting a linear model of y on X, and shrinking the coefficients.

Example: visual representation of ridge coefficients

Recall our last example. Here is a visual representation of the ridge regression coefficients for $\lambda = 25$:



Important details

- **Intercept**: When including an intercept term in the regression, we usually leave this coefficient unpenalized. Otherwise, we could add some constant amount c to the vector y, and this would not result in the same solution. Hence ridge regression with intercept minimizes

$$\sum_{i=1}^{n}(y_i-\beta_0-x'_i\beta)^2+\lambda\sum_{j=1}^{p}\beta_j^2$$

  If we center the columns of X, then the intercept estimate ends up just being $\beta_0=y$, so we usually just re-center y and X, and don't include an intercept. [Think about R2 in simple linear regression].

- **Normalization**: The penalty term is unfair if the predictor variables are not on the same scale. (Can you see why?) Therefore, if we know that the variables are not measured in the same units, we typically scale the columns of X (to have sample variance 1), and then we perform ridge regression.

Bias and variance of ridge regression

The bias and variance are not quite as simple to write down for ridge regression as they are for linear regression, but closed-form expressions are still possible.

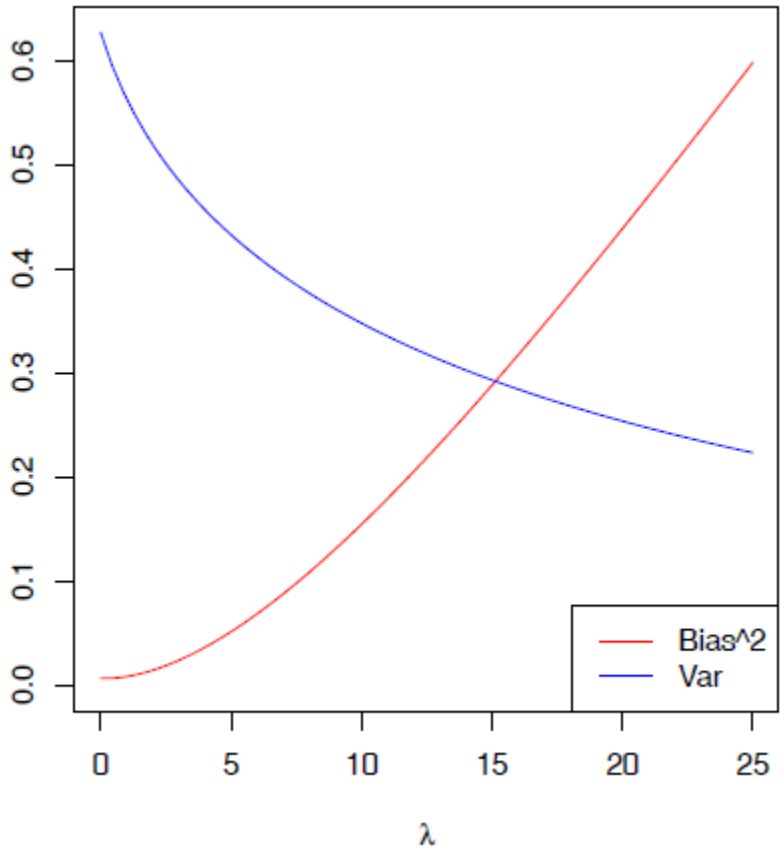
The **general trend** is:

- **The bias increases as λ (amount of shrinkage) increases.**
- **The variance decreases as λ (amount of shrinkage) increases.**

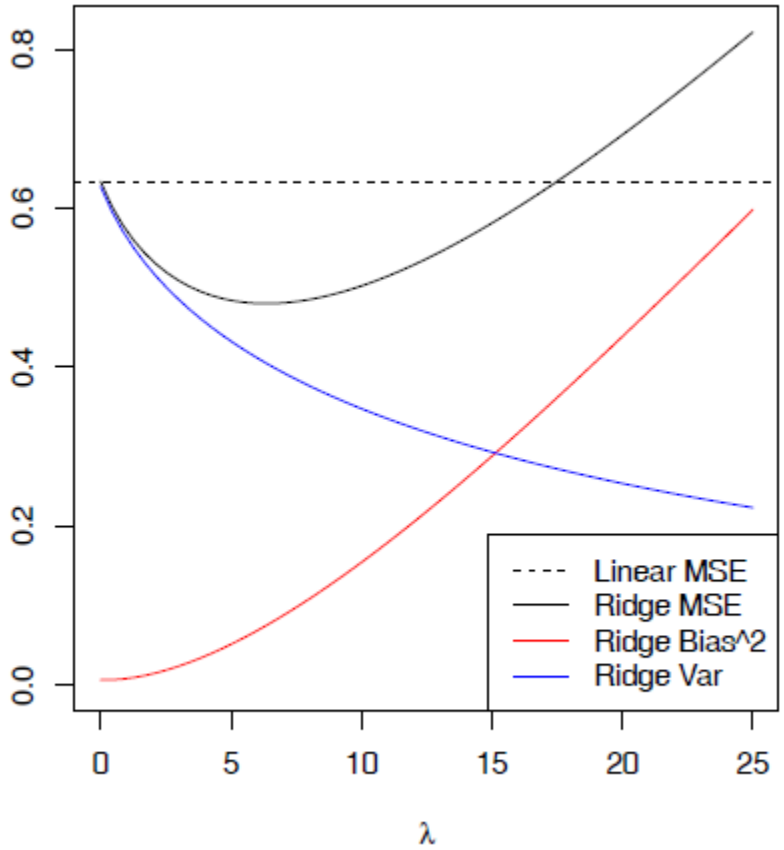
Question: What is the bias at $\lambda = 0$? The variance at $\lambda = \infty$?

Example: bias and variance of ridge regression

Bias and variance for the last example:



Mean squared error for the last example:



Questions

Question 1: OK, but this only works for some values of λ. So how would we choose λ in practice?“

A. Calibrate with numbers used in the literature

B. Trial and error.

C. You can’t

Questions

Question 1: OK, but this only works for some values of λ. So how would we choose λ in practice?“

A. Calibrate with numbers used in the literature

B. Trial and error.

C. You can’t

This is actually quite a hard question. We will use cross validation methods in the following sections.

Questions

Question 2: “What happens when none of the coefficients are small?“

A. λ is smaller because there is no shrinkage to be done.

B. λ is larger since we have to shrink the coefficients even further.

C. λ is set independently of the size of the coefficients.

Questions

Question 2: “What happens when none of the coefficients are small?“

A. λ is smaller because there is no shrinkage to be done.

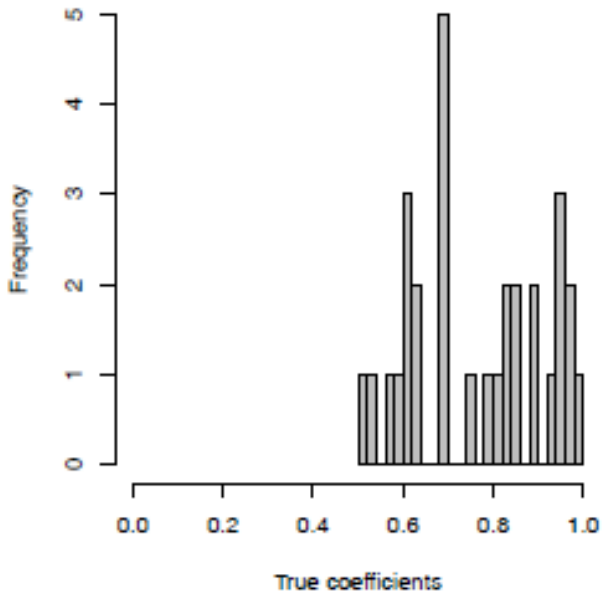
B. λ is larger since we have to shrink the coefficients even further.

C. λ is set independently of the size of the coefficients.

In other words, if all the true coefficients are moderately large, is it still helpful to shrink the coefficient estimates? The answer is (perhaps surprisingly) still “yes". But the advantage of ridge regression here is less dramatic, and the corresponding range for good values of λ is smaller.

Example: moderate regression coefficients

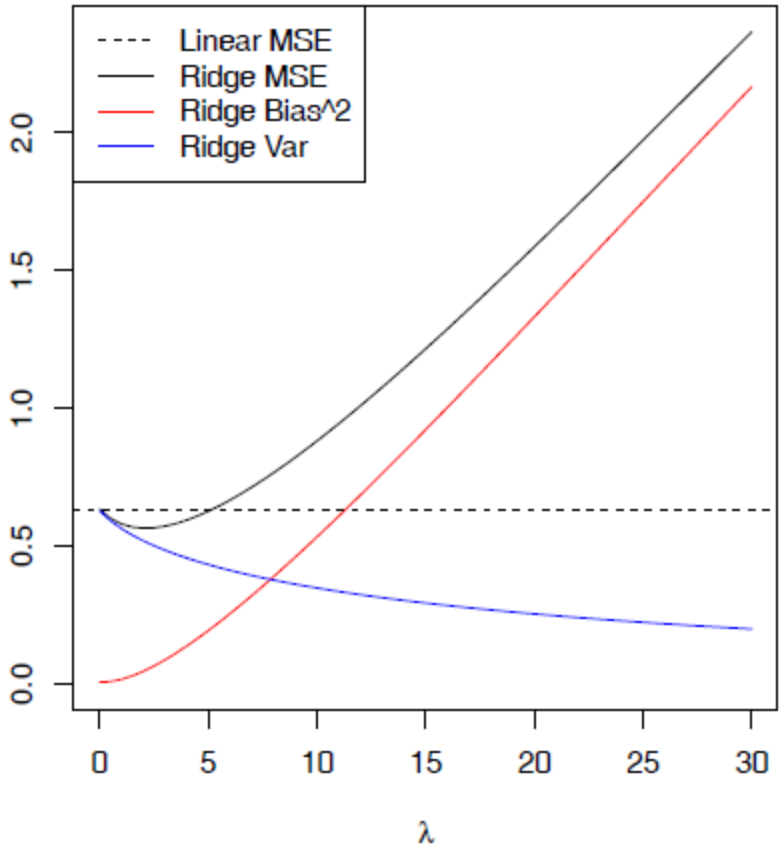
Same setup as before, except now the true coefficients are all moderately large (between 0.5 and 1).



The linear regression fit:
Squared bias ≈ 0.006
Variance ≈ 0.628
Pred. error ≈ 1+0.006+0.628=1.634

Question: Why are these numbers essentially the same as those from the last example, even though the true coefficients changed?

Ridge regression can still outperform linear regression in terms of mean squared error:



Only works for λ less than 5, otherwise it is very biased. Why?

Variable selection

To the other extreme (of a subset of small coefficients), suppose a group of true coefficients are identically zero and response does not depend on these predictors.

The problem of picking out the relevant variables from a larger set is called **variable selection**. This means estimating some coefficients to be exactly zero.

Again: Predictive ability *vs.* interpretative ability.

Question 3: “How does ridge regression perform if a group of the true coefficients was exactly zero?“

A. If some coefficients are zero, ridge will shrink them substantially
B. If some coefficients are zero, ridge will set them to zero
C. If some coefficients are zero, ridge will not perform any shrinkage

Variable selection

To the other extreme (of a subset of small coefficients), suppose a group of true coefficients are identically zero and response does not depend on these predictors.

The problem of picking out the relevant variables from a larger set is called **variable selection**. This means estimating some coefficients to be exactly zero.

Again: Predictive ability *vs.* interpretative ability.

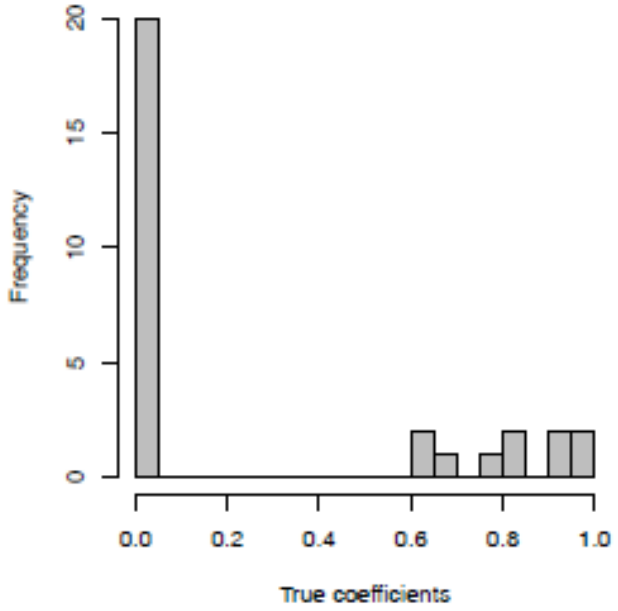
Question 3: “How does ridge regression perform if a group of the true coefficients was exactly zero?“

A. If some coefficients are zero, ridge will shrink them substantially
B. If some coefficients are zero, ridge will set them to zero
C. If some coefficients are zero, ridge will not perform any shrinkage

Answer: It depends on whether we are interested in prediction or interpretation. We'll consider the former first.

Example: subset of zero coefficients

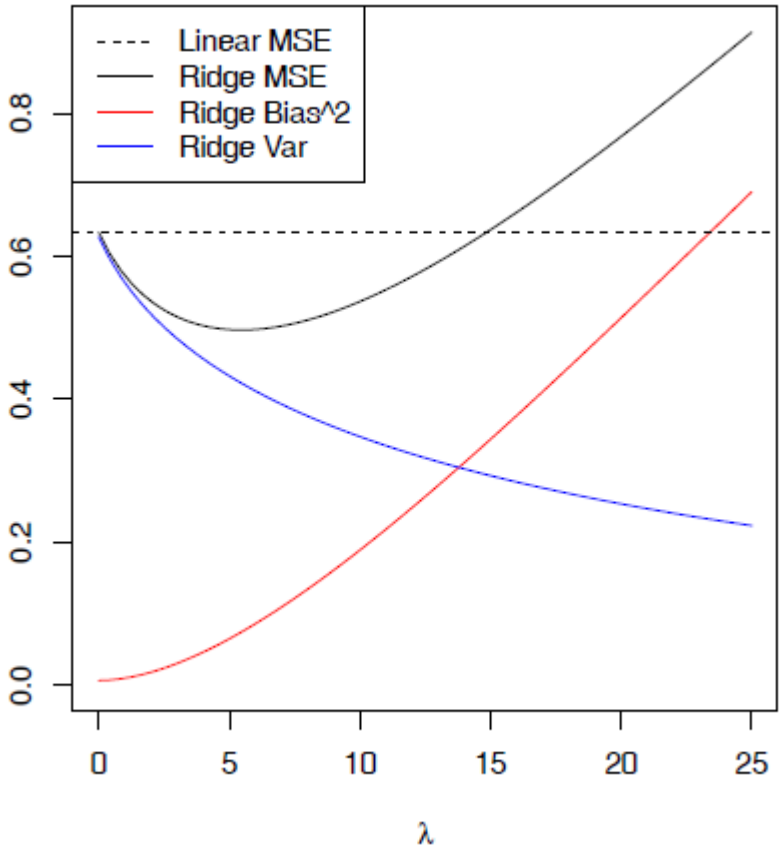
Same setup as before, except now 10 true coefficients are large (between 0.5 and 1) and 20 are exactly 0.



The linear regression fit:
Squared bias ≈ 0.006
Variance ≈ 0.627
Pred. error ≈ 1+0.006+0.627=1.633

Again these numbers are essentially the same .

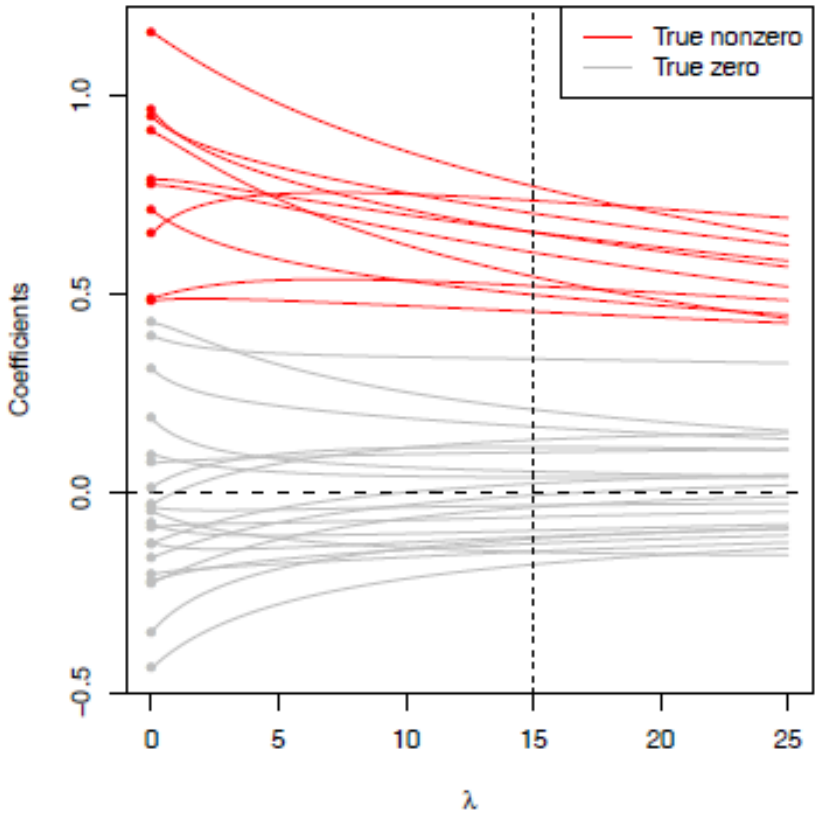
Ridge regression performs well in terms of mean-squared error.



Prediction

Why is the bias not as large here for large λ?

As we vary λ we get different ridge regression coefficients, the larger the λ the more shrinkage. Here we plot them against λ.



The red paths correspond to the true nonzero coefficients; the gray paths correspond to true zeros. The vertical dashed line at λ = 15 marks the point above which ridge regression's MSE starts losing to linear regression.

☹ **Interpretation**

Note: The gray coefficient paths are not exactly zero; they are shrunken, but still nonzero.

Ridge regression doesn't perform variable selection

We can show that ridge regression doesn't set coefficients exactly to zero unless $\lambda = \infty$, in which case they are all set to zero.

In summary, ridge regression cannot perform variable selection, and even though it performs well in terms of prediction accuracy, it does poorly in terms of offering a clear interpretation.

Recap: ridge regression

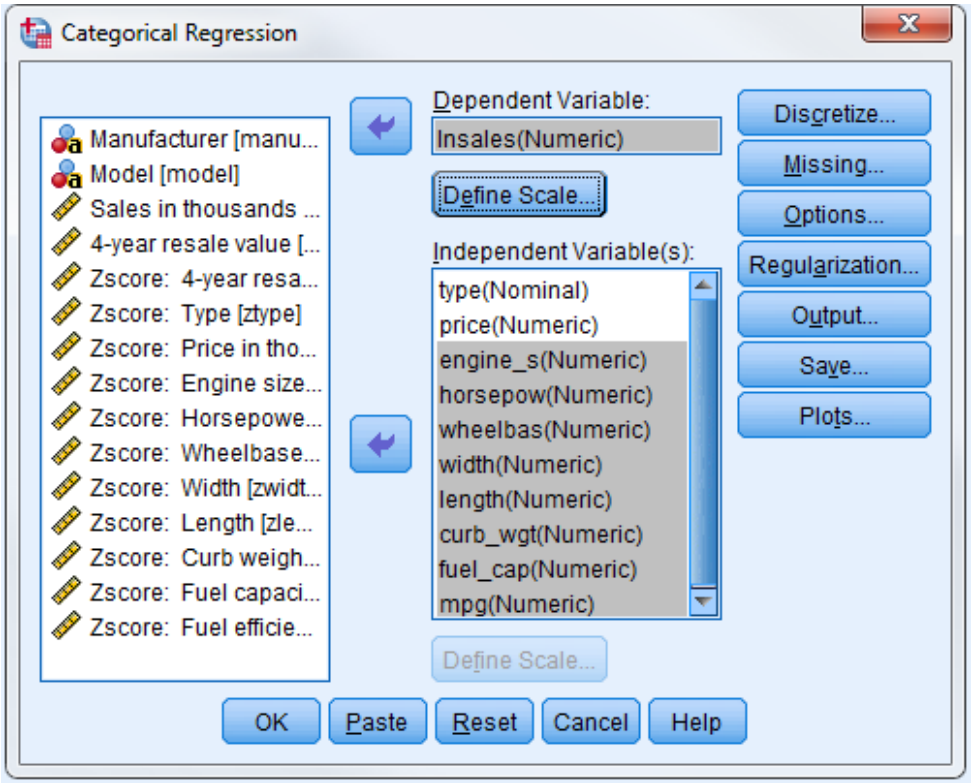
- We learned ridge regression, which minimizes the usual regression criterion plus a penalty term on the squared L2 norm of the coefficient vector. As such, it shrinks the coefficients towards zero. This introduces some bias, but can greatly reduce the variance, resulting in a better mean-squared error.

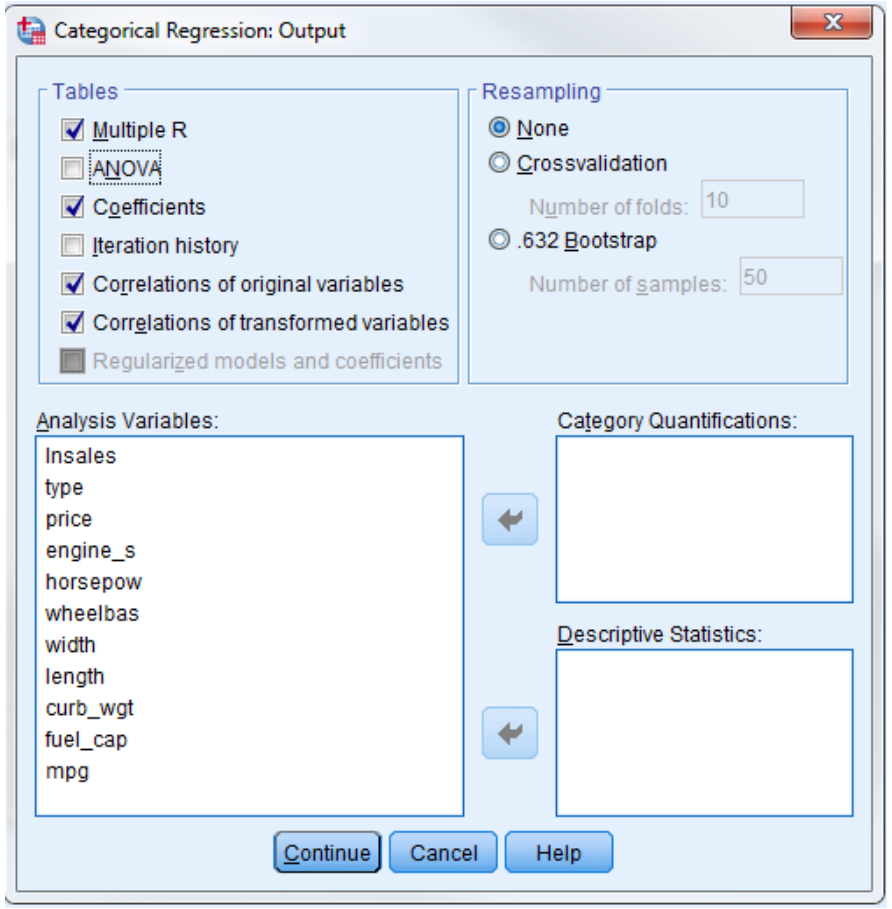
- The amount of shrinkage is controlled by λ, the tuning parameter that multiplies the ridge penalty. Large λ means more shrinkage, and so we get different coefficient estimates for different values of λ. Choosing an appropriate value of λ is important, and also difficult. This can be done using cross validation. (we will discuss this later).

- Ridge regression performs particularly well when there is a subset of true coefficients that are small or even zero. It doesn't do as well when all of the true coefficients are moderately large; however, in this case it can still outperform linear regression over a pretty narrow range of (small) λ values.

DIY

- Data file: car_sales.sav
- Reproduce the previous analysis of car sales using the SPSS command: Analyze > regression > optimal scalling (CATREG).
- Then use a Ridge model.
- First recode the variable type so that a “0” becomes a “2”. (command CATREG does not deal with zero valued variables).


Categorical Regression
Manufacturer [manu...
Model [model]
Sales in thousands ...
4-year resale value [...
Zscore: 4-year resa...
Zscore: Type [ztype]
Zscore: Price in tho...
Zscore: Engine size...
Zscore: Horsepowe...
Zscore: Wheelbase...
Zscore: Width [zwidt...
Zscore: Length [zle...
Zscore: Curb weigh...
Zscore: Fuel capaci...
Zscore: Fuel efficie...
Dependent Variable:
Insales(Numeric)
Define Scale...
Independent Variable(s):
type(Nominal)
price(Numeric)
engine_s(Numeric)
horsepow(Numeric)
wheelbas(Numeric)
width(Numeric)
length(Numeric)
curb_wgt(Numeric)
fuel_cap(Numeric)
mpg(Numeric)
Define Scale...
Discretize...
Missing...
Options...
Regularization...
Output...
Save...
Plots...
OK
Paste
Reset
Cancel
Help


Categorical Regression: Output
Tables
Multiple R
ANOVA
Coefficients
Iteration history
Correlations of original variables
Correlations of transformed variables
Regularized models and coefficients
Resampling
None
Crossvalidation
Number of folds: 10
.632 Bootstrap
Number of samples: 50
Analysis Variables:
Insales
type
price
engine_s
horsepow
wheelbas
width
length
curb_wgt
fuel_cap
mpg
Category Quantifications:
Descriptive Statistics:
Continue
Cancel
Help

IBM SPSS

Correlations Original Variables

	Vehicle type	Price in thousands	Engine size	Horsepower	Wheelbase	Width	Length	Curb weight	Fuel capacity	Fuel efficiency
Vehicle type	1,000	,019	-,274	-,017	-,361	-,243	-,151	-,489	-,620	,574
Price in thousands	,019	1,000	,631	,830	,210	,316	,197	,568	,471	-,545
Engine size	-,274	,631	1,000	,816	,553	,658	,574	,789	,708	-,761
Horsepower	-,017	,830	,816	1,000	,309	,509	,369	,616	,533	-,627
Wheelbase	-,361	,210	,553	,309	1,000	,652	,824	,707	,664	-,465
Width	-,243	,316	,658	,509	,652	1,000	,688	,688	,591	-,541
Length	-,151	,197	,574	,369	,824	,688	1,000	,683	,597	-,433
Curb weight	-,489	,568	,789	,616	,707	,688	,683	1,000	,837	-,795
Fuel capacity	-,620	,471	,708	,533	,664	,591	,597	,837	1,000	-,801
Fuel efficiency	,574	-,545	-,761	-,627	-,465	-,541	-,433	-,795	-,801	1,000
Dimension	1	2	3	4	5	6	7	8	9	10
Eigenvalue	6,030	1,537	1,143	,393	,250	,192	,151	,130	,107	,068

Correlations Transformed Variables

	Vehicle type	Price in thousands	Engine size	Horsepower	Wheelbase	Width	Length	Curb weight	Fuel capacity	Fuel efficiency
Vehicle type	1,000	-,019	,274	,017	,361	,243	,151	,489	,620	-,574
Price in thousands	-,019	1,000	,631	,830	,210	,316	,197	,568	,471	-,545
Engine size	,274	,631	1,000	,816	,553	,658	,574	,789	,708	-,761
Horsepower	,017	,830	,816	1,000	,309	,509	,369	,616	,533	-,627
Wheelbase	,361	,210	,553	,309	1,000	,652	,824	,707	,664	-,465
Width	,243	,316	,658	,509	,652	1,000	,688	,688	,591	-,541
Length	,151	,197	,574	,369	,824	,688	1,000	,683	,597	-,433
Curb weight	,489	,568	,789	,616	,707	,688	,683	1,000	,837	-,795
Fuel capacity	,620	,471	,708	,533	,664	,591	,597	,837	1,000	-,801
Fuel efficiency	-,574	-,545	-,761	-,627	-,465	-,541	-,433	-,795	-,801	1,000
Dimension	1	2	3	4	5	6	7	8	9	10
Eigenvalue	6,030	1,537	1,143	,393	,250	,192	,151	,130	,107	,068

Model Summary

	Multiple R	R Square	Adjusted R Square	Apparent Prediction Error
Standardized Data	,685	,469	,432	,531

Dependent Variable: Log-transformed sales
Predictors: Vehicle type Price in thousands Engine size Horsepower Wheelbase Width Length Curb weight Fuel capacity Fuel efficiency

Coefficients

	Standardized Coefficients				
	Beta	Bootstrap (1000) Estimate of Std. Error	df	F	Sig.
Vehicle type	,163	,103	1	2,508	,116
Price in thousands	-,623	,127	1	24,186	,000
Engine size	,417	,139	1	8,951	,003
Horsepower	-,189	,161	1	1,377	,243
Wheelbase	,161	,128	1	1,585	,210
Width	-,046	,104	1	,193	,661
Length	,004	,150	1	,001	,976
Curb weight	,153	,164	1	,869	,353
Fuel capacity	-,027	,177	1	,023	,880
Fuel efficiency	,201	,140	1	2,059	,153

Dependent Variable: Log-transformed sales

Correlations and Tolerance

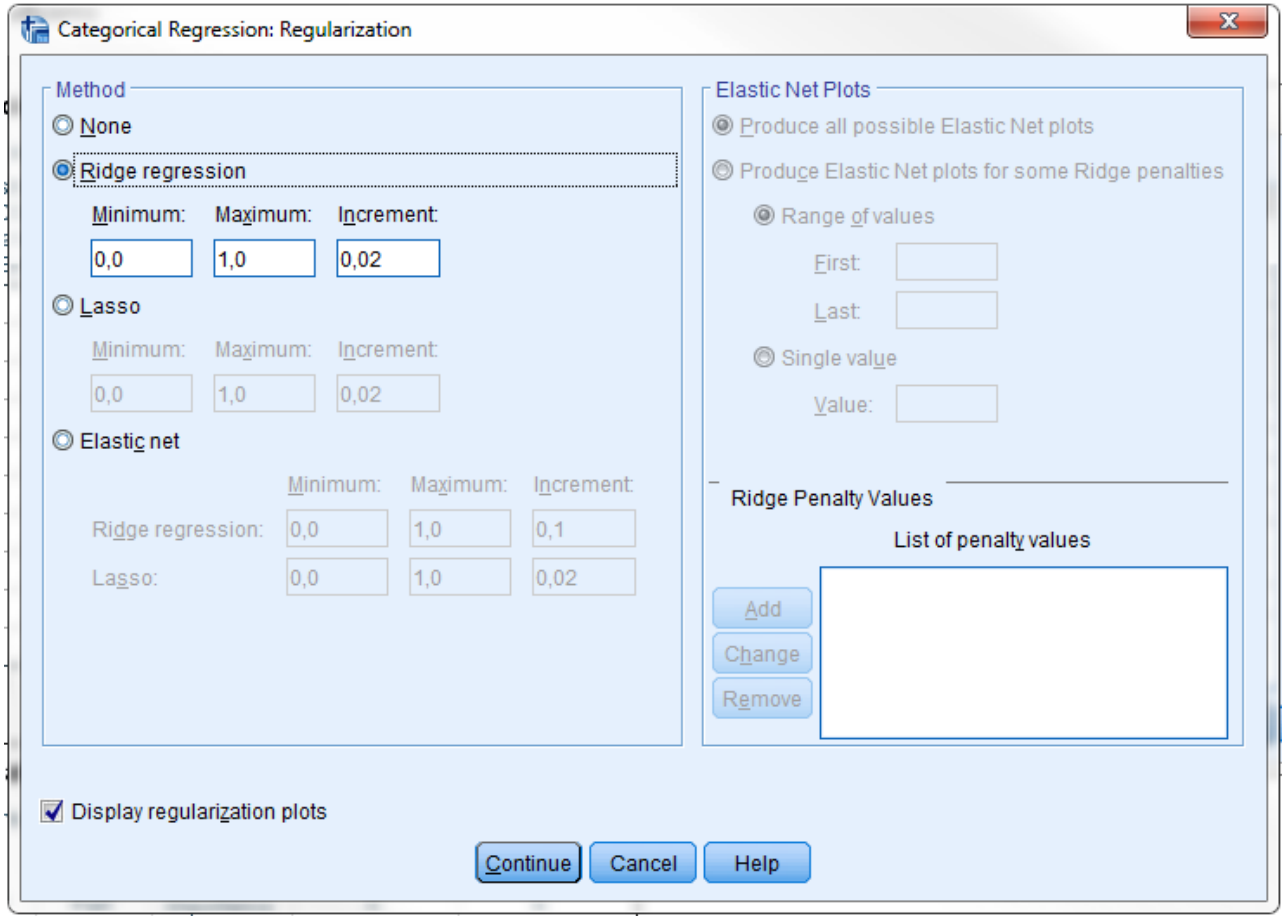
	Correlations				Tolerance	
	Zero-Order	Partial	Part	Importance	After Transformation	Before Transformation
Vehicle type	,277	,115	,084	,096	,269	,269
Price in thousands	-,535	-,356	-,278	,710	,198	,198
Engine size	-,077	,227	,170	-,068	,166	,166
Horsepower	-,381	-,097	-,071	,153	,142	,142
Wheelbase	,231	,106	,078	,079	,233	,233
Width	,063	-,039	-,028	-,006	,386	,386
Length	,178	,003	,002	,002	,189	,189
Curb weight	-,008	,077	,056	-,002	,133	,133
Fuel capacity	,025	-,016	-,011	-,001	,181	,181
Fuel efficiency	,095	,124	,091	,041	,205	,205

Dependent Variable: Log-transformed sales

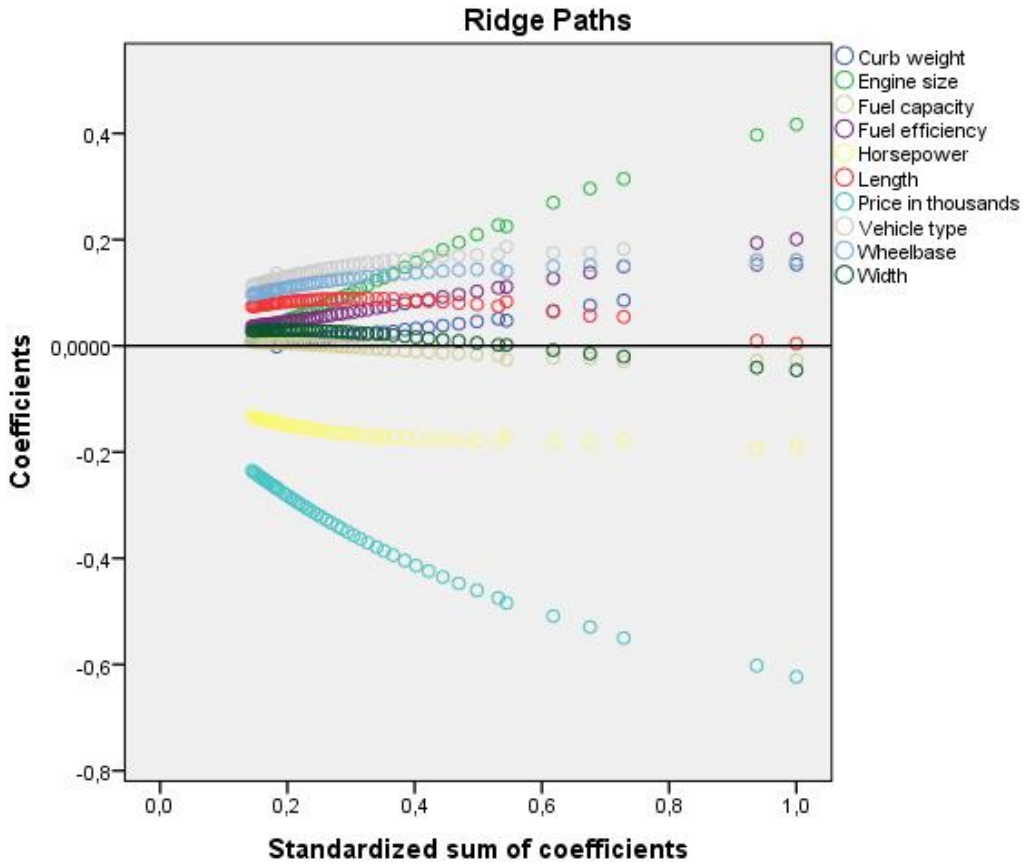
Compare to linear regression results

IBM SPSS

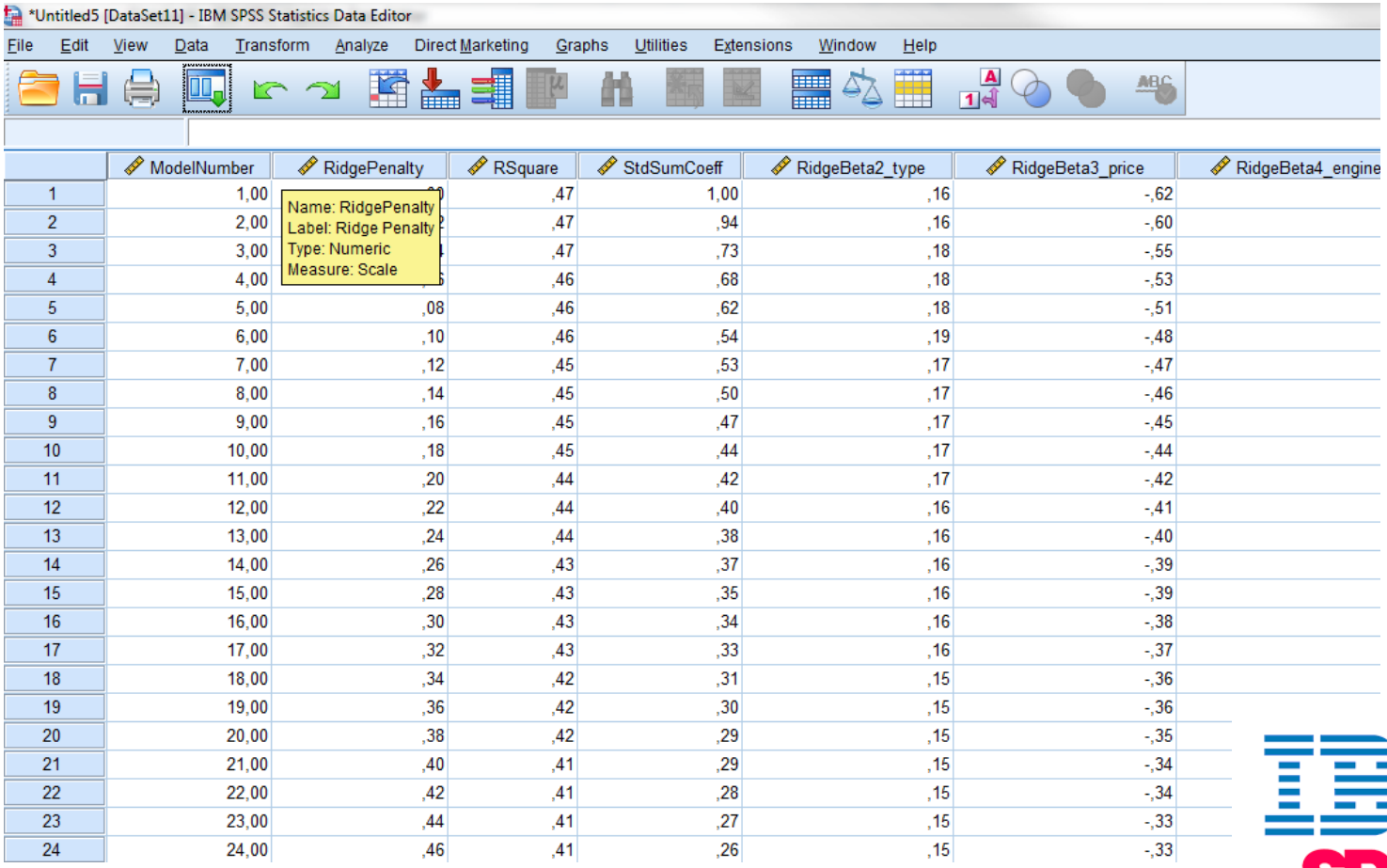
Ridge



Coefficients for different levels of penalty



New dataset created with coefficients for each penalty level



	ModelNumber	RidgePenalty	RSquare	StdSumCoeff	RidgeBeta2_type	RidgeBeta3_price	RidgeBeta4_engine
1	1,00		,47	1,00	,16	-,62	
2	2,00		,47	,94	,16	-,60	
3	3,00		,47	,73	,18	-,55	
4	4,00		,46	,68	,18	-,53	
5	5,00	,08	,46	,62	,18	-,51	
6	6,00	,10	,46	,54	,19	-,48	
7	7,00	,12	,45	,53	,17	-,47	
8	8,00	,14	,45	,50	,17	-,46	
9	9,00	,16	,45	,47	,17	-,45	
10	10,00	,18	,45	,44	,17	-,44	
11	11,00	,20	,44	,42	,17	-,42	
12	12,00	,22	,44	,40	,16	-,41	
13	13,00	,24	,44	,38	,16	-,40	
14	14,00	,26	,43	,37	,16	-,39	
15	15,00	,28	,43	,35	,16	-,39	
16	16,00	,30	,43	,34	,16	-,38	
17	17,00	,32	,43	,33	,16	-,37	
18	18,00	,34	,42	,31	,15	-,36	
19	19,00	,36	,42	,30	,15	-,36	
20	20,00	,38	,42	,29	,15	-,35	
21	21,00	,40	,41	,29	,15	-,34	
22	22,00	,42	,41	,28	,15	-,34	
23	23,00	,44	,41	,27	,15	-,33	
24	24,00	,46	,41	,26	,15	-,33	

IBM SPSS

The lasso

Modern regression

Variable selection

Ridge regression:

- Can have better prediction error than linear regression in a variety of scenarios. It works best when there is a subset of the true coefficients that are small or zero.

- But it will never sets coefficients to zero exactly, and therefore cannot perform variable selection in the linear model. While this does not seem to hurt its prediction ability, it is not desirable for the purposes of interpretation (especially if the number of variables p is large).

The lasso can!

The lasso

The lasso estimate is defined as minimizing

$$\sum_{i=1}^{n}(y_i - x'_i\beta)^2 + \lambda\sum_{j=1}^{p}|\beta_j|$$

Loss *Penalty*

The difference between the lasso problem and ridge regression is the use of a L1 (absolute value) *vs.* an L2 (squared) penalty.

Even though both problems look similar, their solutions behave very differently.

Note: “Lasso" is an acronym for: **Least Absolute Selection and Shrinkage Operator.**

Tuning

The tuning parameter (λ) controls the strength of the penalty, and (like for ridge regression) we get:

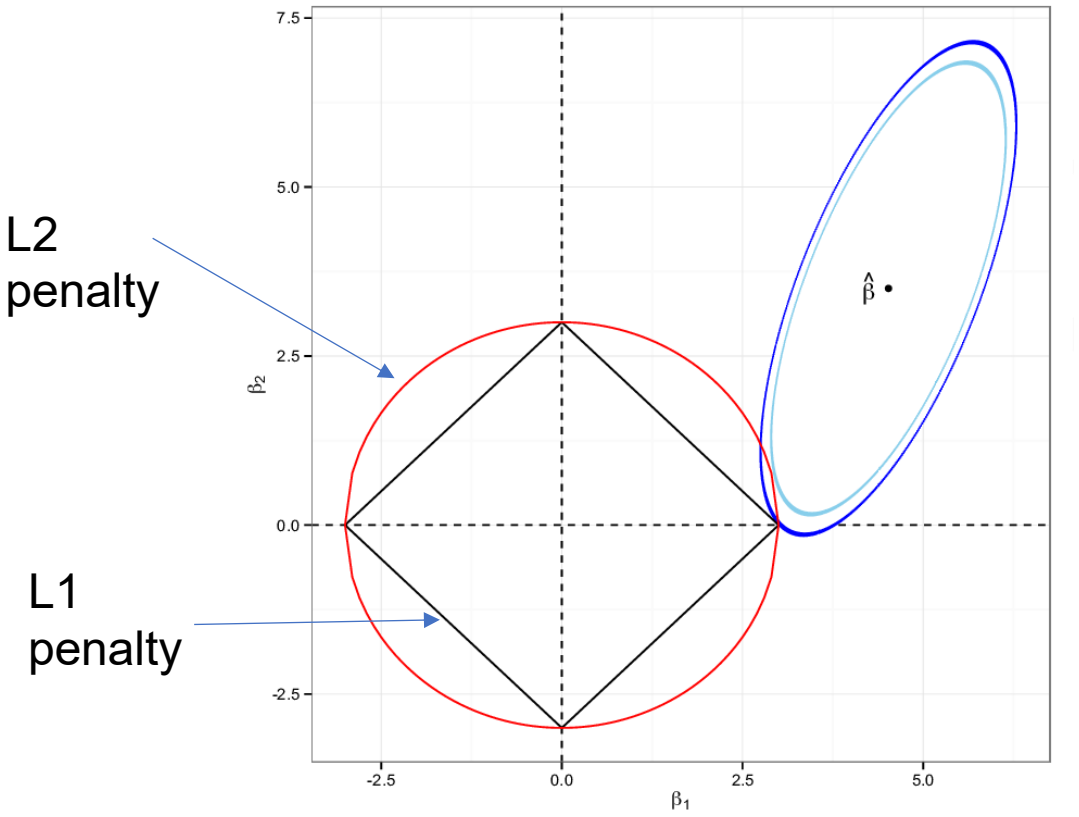
- $\beta^{lasso} = \beta^{OLS}$ when $\lambda = 0$, and
- $\beta^{lasso} = 0$ when $\lambda = \infty$.

For λ in between these two extremes, we are balancing two ideas: fitting a linear model of y on X, and shrinking the coefficients.

However, the nature of the L1 penalty causes some coefficients to be shrunken exactly to zero.

Lasso vs. Ridge

The nature of the L1 penalty causes some coefficients to be shrunken exactly to zero.



Lasso vs. Ridge

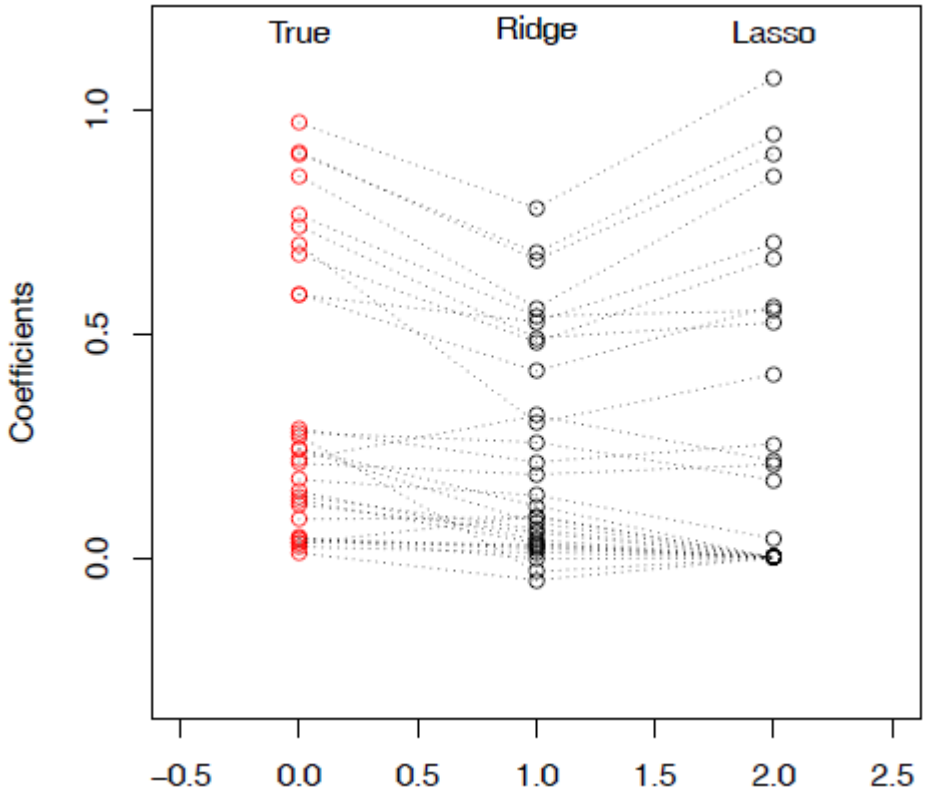
The lasso is substantially different from ridge regression on one dimension: it is able to perform **variable selection** in the linear model.

As λ increases:

1. More coefficients are set to zero (less variables are selected), and
2. Among the nonzero coefficients, more shrinkage is employed.

Example: visual representation of lasso coefficients

Our running example with n = 50, p = 30, 10 large true coefficients, 20 small. Here is a visual representation of lasso *vs*. ridge coefficients (with the same degrees of freedom):



Important details

› **Intercept**: When including an intercept term, we usually leave it unpenalized, just as in ridge. Hence the lasso problem with intercept minimizes

$$\sum_{i=1}^{n}(y_i-\beta_0-x'_i\beta)^2+\lambda\sum_{j=1}^{p}|\beta_j|$$

As we've seen before, if we center the columns of X, then the intercept estimate turns out to be $\beta_0 = \bar{y}$. Typically, re-center y and X, and don't include an intercept.

› **Normalization**: As with ridge regression, the penalty term is unfair if the predictor variables are on different scales. First, scale the columns of X (to have sample variance 1), and then solve the lasso problem.

Bias and variance of the lasso

The bias and variance are not quite as simple to write down for lasso regression as they are for linear regression, but closed-form expressions are still possible.

The **general trend** is:

- **The bias increases as λ (amount of shrinkage) increases.**
- **The variance decreases as λ (amount of shrinkage) increases.**

Question 1: What is the bias at $\lambda = 0$?

A. Highest

B. Lowest

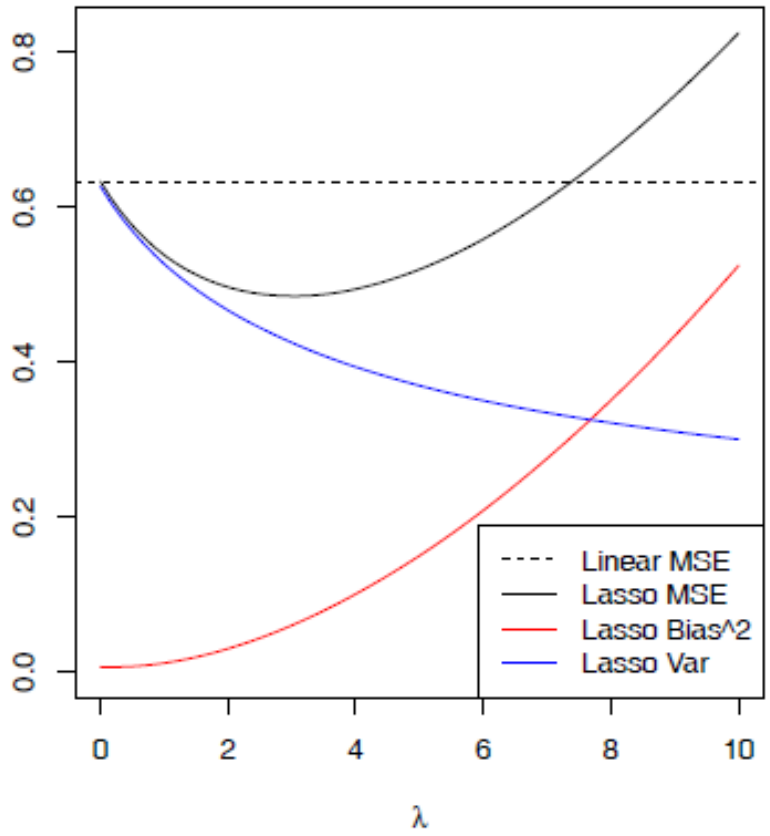
Question 2: The variance at $\lambda = \infty$?

A. Highest

B. Lowest

Example: subset of small coefficients

Example: n = 50, p = 30; true coefficients: 10 large, 20 small

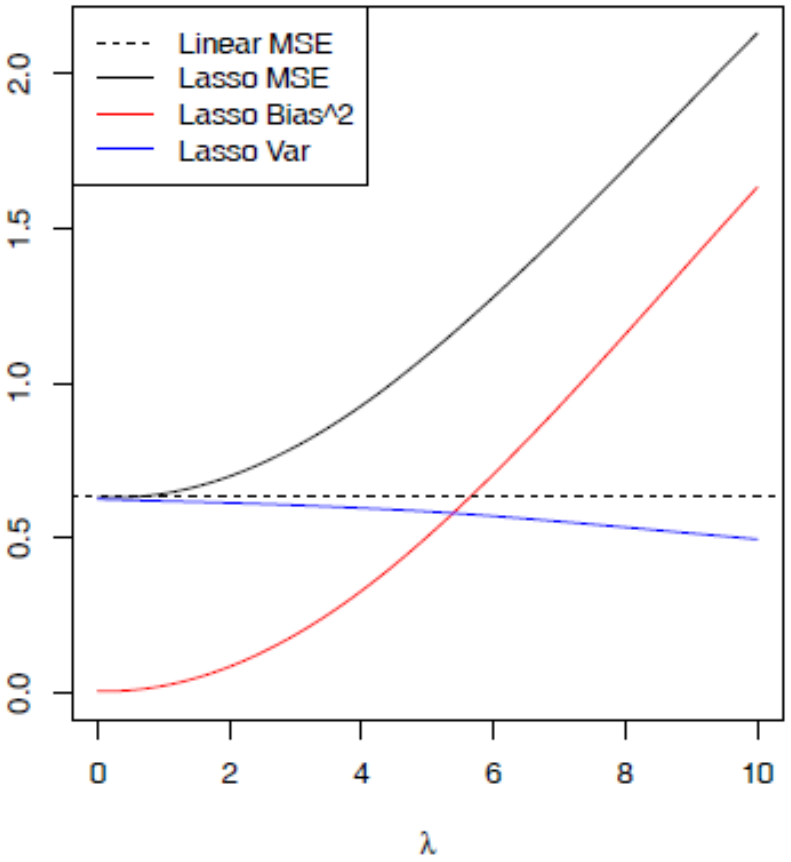


In terms of prediction error (or mean squared error), the lasso performs comparably to ridge regression.

Example: all moderate coefficients

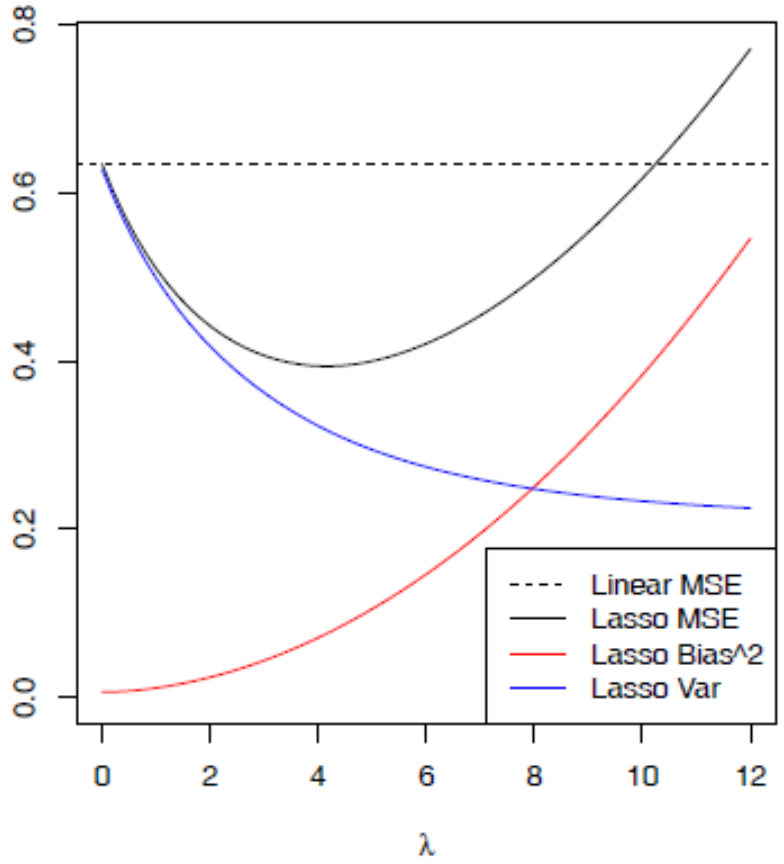
Example: n = 50, p = 30; 30 moderately large

Note that here, as opposed to ridge regression, the variance doesn't decrease fast enough to make the lasso favorable for small λ



Example: subset of zero coefficients

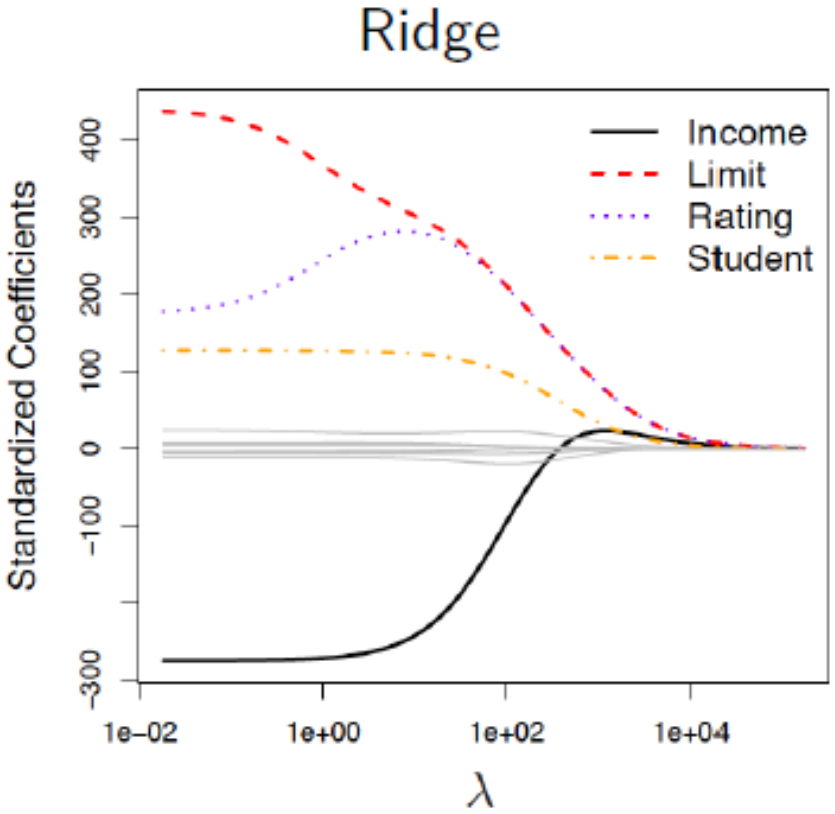
Example: n = 50, p = 30; 10 large, 20 zero

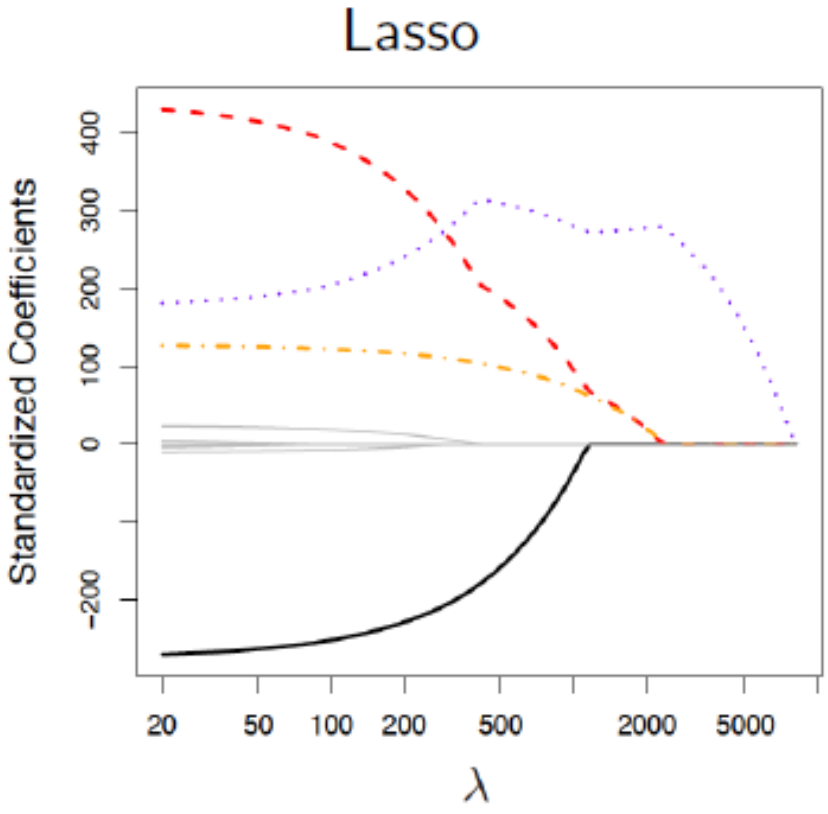


Example: credit data

Response variable is average credit debt.

Predictors are income, limit (credit limit), rating (credit rating), student (indicator), and others.

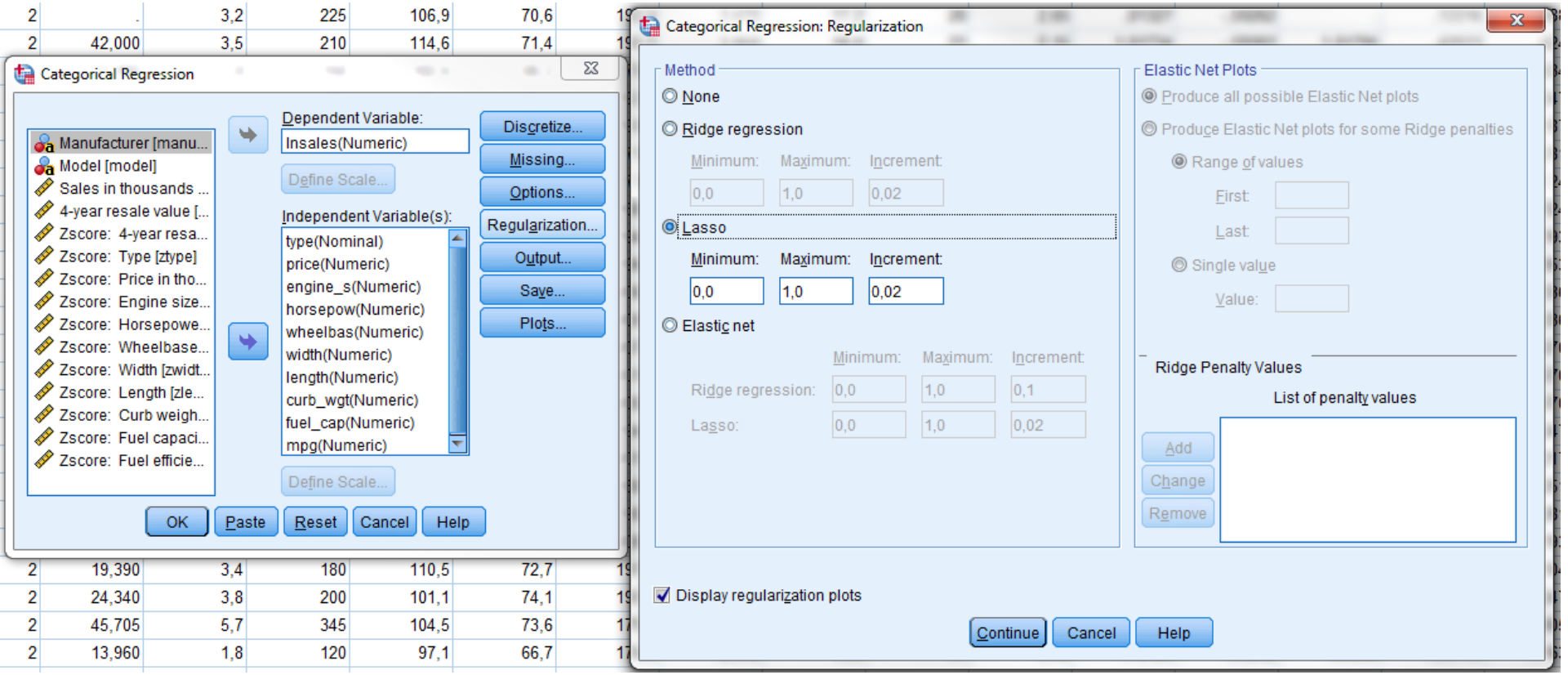




Recap: the lasso

- The lasso is a variable selection method in the linear model setting. The lasso uses a penalty like ridge regression, except the penalty is the L1 norm of the coefficient vector, which causes the estimates of some coefficients to be exactly zero. This is in contrast to ridge regression which never sets coefficients to zero.

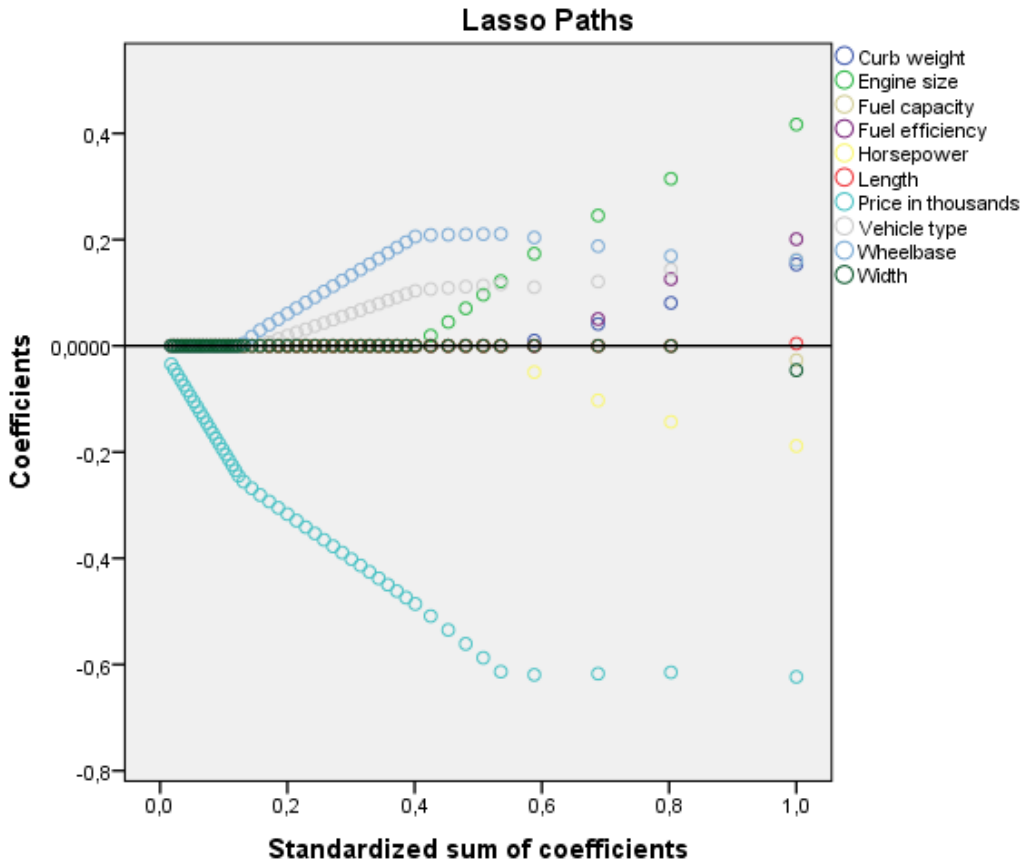
- The tuning parameter controls the strength of the L1 penalty. The lasso estimates are generally biased, but have good mean squared error (comparable to ridge regression). On top of this, the fact that it sets coefficients to zero is good for interpretation.



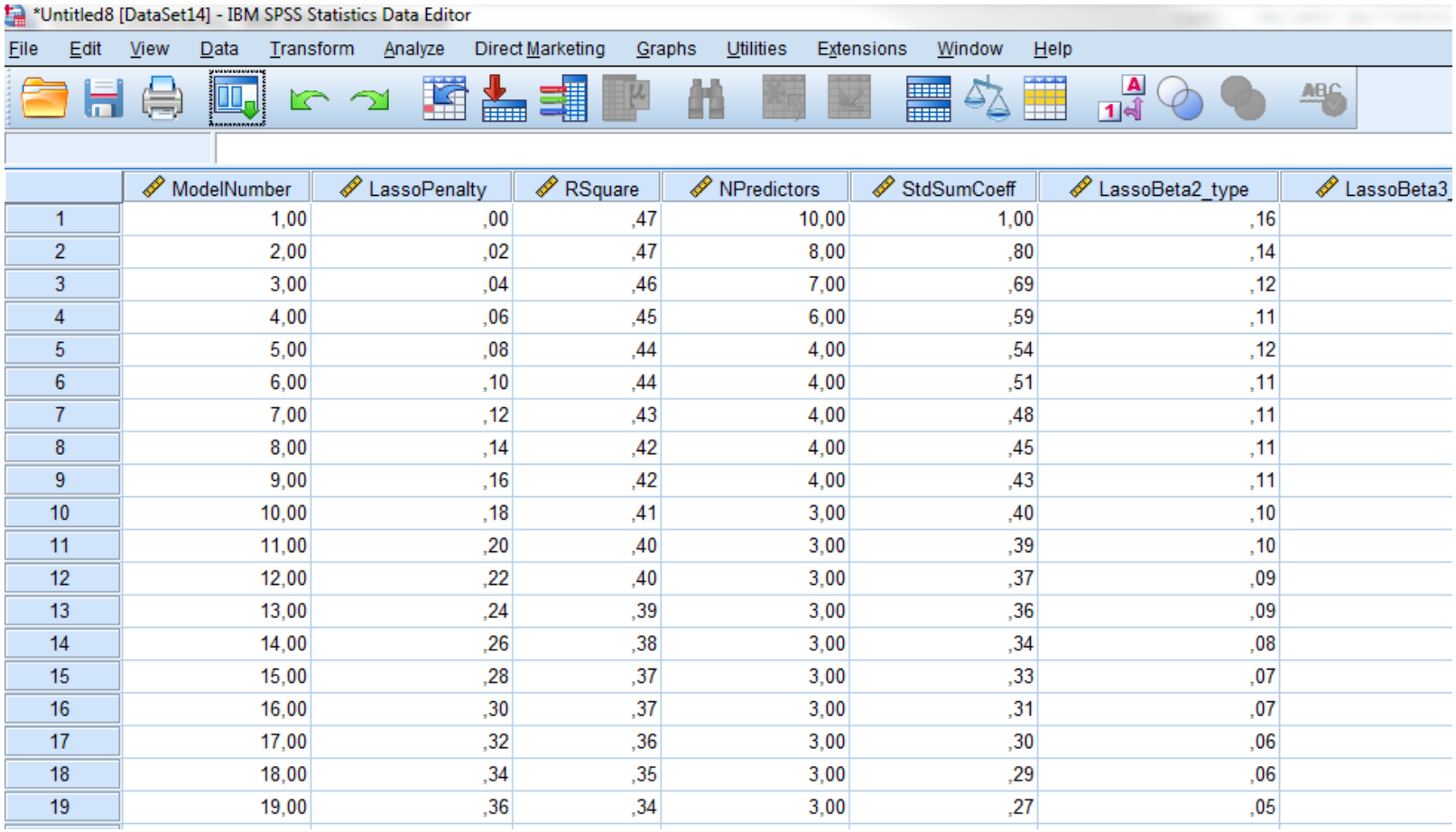
IBM SPSS

Lisbon School of Economics & Management

Coefficients for different levels of penalty



New dataset created with coefficients for each penalty level



*Untitled8 [DataSet14] - IBM SPSS Statistics Data Editor

File Edit View Data Transform Analyze Direct Marketing Graphs Utilities Extensions Window Help

	ModelNumber	LassoPenalty	RSquare	NPredictors	StdSumCoeff	LassoBeta2_type	LassoBeta3
1	1,00	,00	,47	10,00	1,00	,16	
2	2,00	,02	,47	8,00	,80	,14	
3	3,00	,04	,46	7,00	,69	,12	
4	4,00	,06	,45	6,00	,59	,11	
5	5,00	,08	,44	4,00	,54	,12	
6	6,00	,10	,44	4,00	,51	,11	
7	7,00	,12	,43	4,00	,48	,11	
8	8,00	,14	,42	4,00	,45	,11	
9	9,00	,16	,42	4,00	,43	,11	
10	10,00	,18	,41	3,00	,40	,10	
11	11,00	,20	,40	3,00	,39	,10	
12	12,00	,22	,40	3,00	,37	,09	
13	13,00	,24	,39	3,00	,36	,09	
14	14,00	,26	,38	3,00	,34	,08	
15	15,00	,28	,37	3,00	,33	,07	
16	16,00	,30	,37	3,00	,31	,07	
17	17,00	,32	,36	3,00	,30	,06	
18	18,00	,34	,35	3,00	,29	,06	
19	19,00	,36	,34	3,00	,27	,05	

Model selection and validation

Regularization

Linear regression has generally small bias (zero bias, when the true model is linear) but high variance, leading to poor predictions.

Modern methods introduce some bias but significantly reduce the variance, leading to better predictive accuracy. More generally, modern methods minimize

$$||y - X\beta||_2^2 + \lambda R(B)$$

The term R is called a **penalty** or **regularizer**, and modifying the regression problem in this way is called applying **regularization**:

- Note: Regularization can be applied beyond regression: e.g., it can be applied to classification, clustering, principal component analysis.

Regularization

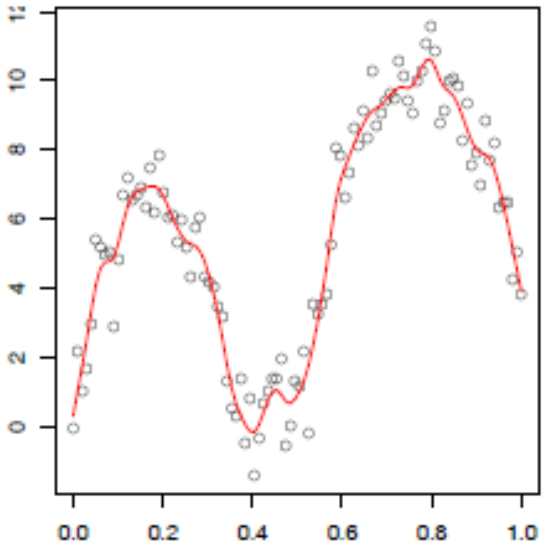
$$||y - X\beta||_2^2 + \lambda R(B)$$

In Ridge R(B)= $\sum_{j=1}^{p} \beta_j^{\,2}$

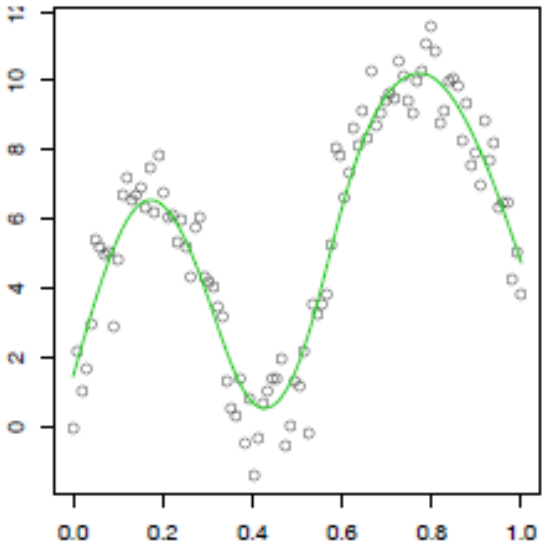
In Lasso R(B)= $\sum_{j=1}^{p} |\beta_j|$

Example: smoothing splines

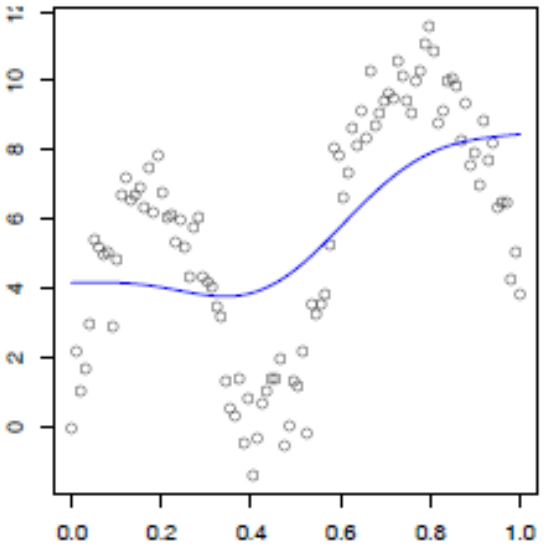
Example with $n = 100$ points:



λ too small



λ just right



λ too big

Setting the tuning parameter

- Each **regularization** method has an associated **tuning parameter**: e.g., this was λ for lasso and ridge regression in the penalized forms.

- The **tuning parameter** controls the amount of **regularization**, so choosing a good value of the tuning parameter is crucial. Each tuning parameter value corresponds to a fitted model. We also refer to this task as **model selection**.

- A **good choice of tuning parameter**, depends on whether our goal is **prediction accuracy** or **interpretation**. We'll cover choosing the tuning parameter for the purposes of prediction; choosing the tuning parameter for the latter purpose is a harder problem.

House characteristics

- Location
- Area
- Typology
- Construction year
- ~~Bathrooms~~
- ~~Energy certificate~~
- Central heating
- AC
- Garage
- Garden
- ~~Fireplace~~
- Gatted community
- ~~Equipped kitchen~~
- Storage
- Swimming pool
- ~~Suite~~
- Terrace
- Balcony
- Security
- Sea View

John built his model to predict house prices with 5 variables considered irrelevant and thus removed. The obtained R^2 is 95%. He now knows that he can predict 95% of the variation in future house prices.

A. True

B. False

Prediction error and test error

The setup is:

$$y_i = f(x_i) + \varepsilon_i,\ i = 1,\ldots,n$$

x_i are fixed (nonrandom) predictor measurements, f(.) is the true function we are trying to predict and ε_i are random errors.

Call (x_i , y_i), i=1,..,n the **training data**. Given an **estimator** $\hat{f}$ built on the training data, consider the average prediction error over all inputs

$$PE(\hat{f}) = E\left[\frac{1}{L}\sum_{l=1}^{L}(y'_l - \hat{f}(x_l))^2\right]$$

Where y_l, l=,..,L (the **test data**) are another set of observations, independent of $y_1,\ldots, y_n$.

Suppose that $\hat{f} = \hat{f}_\theta$ depends on a **tuning parameter θ**, and we want to choose θ to minimize the **average prediction error** PE($\hat{f}_\theta$).

If we actually had **training data** $y_1,\ldots, y_n$ and **test data** $y'_1,\ldots, y'_L$ (meaning that we don't use this to build $\hat{f}_\theta$), we could simply calculate the average test error:

$$TestErr(\hat{f}_\theta) = \frac{1}{L}\sum_{l=1}^{L}(y'_l - \hat{f}_\theta(x_l))^2$$

as an estimate for PE($\hat{f}_\theta$). The larger L is, the better this estimate.

We usually don't have test data. So what to do instead?

What's wrong with training error?

It may seem like

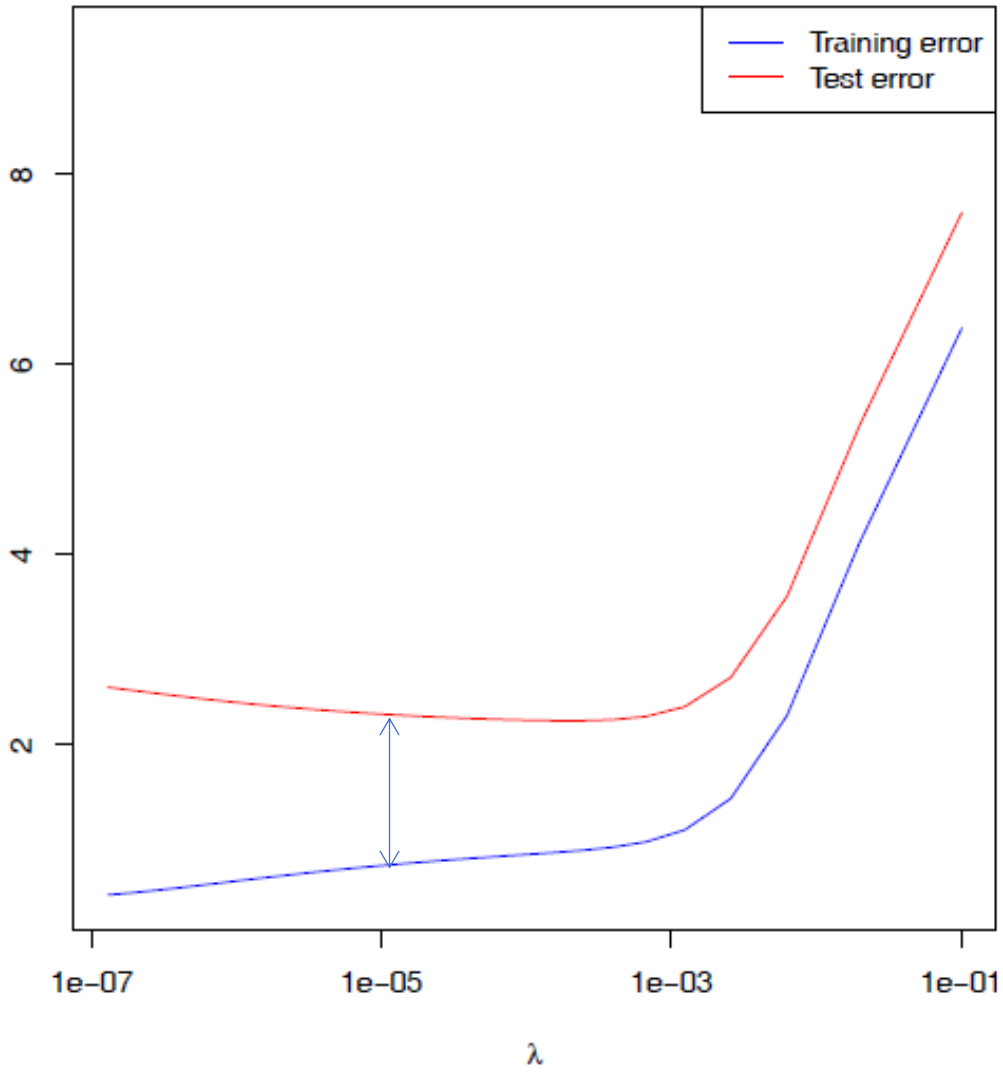
$$TestErr(\hat{f}_\theta) = \frac{1}{L}\sum_{l=1}^{L}(y'_l - \hat{f}_\theta(x_l))^2$$

, and

$$TestErr(\hat{f}_\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}_\theta(x_i))^2$$

shouldn't be too different. After all, y_i and y'_l are independent copies of each other. The second quantity is called the training error: this is the error of $\hat{f}$ as measured by the data we used to build it (in sample).

But actually, the training (out of sample) and test (in sample) error curves are fundamentally different. Why?



Training and test error curves, averaged over 100 simulations.

Test sample

If the problem is getting a test sample, just randomly split the data in two samples one for estimation (training set) and one for validation (test set).

What is the problem of this approach?

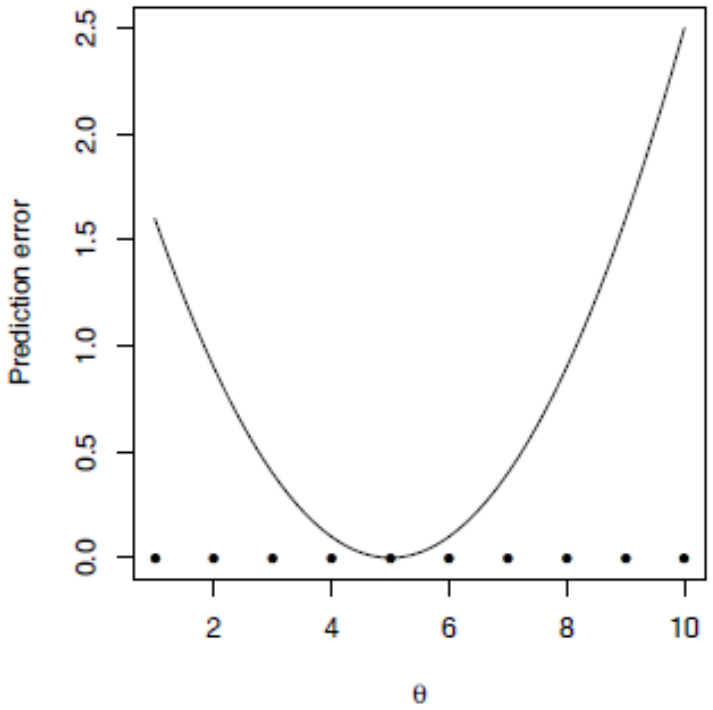
A. Losing observations for training (reduced sample size)

B. Losing observations for testing (reduced sample size)

Cross-validation

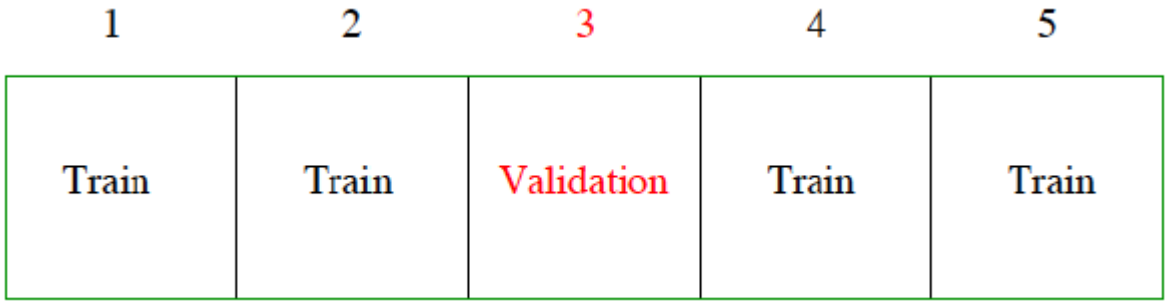
Cross-validation is a simple, intuitive way to estimate prediction error, given training data (x_i , y_i), i=1,..,n, and an estimator $\hat{f}_\theta$, that depends on a tuning parameter θ.

Even if θ is a continuous parameter, it's usually not practically feasible to consider all possible values of θ, so we discretize the range and consider choosing over some discrete set $\{\theta_1,.., \theta_m\}$.



K-fold cross validation

For a number K, we split the training pairs into K parts or “folds“ (commonly K = 5 or K = 10)



K-fold cross validation considers training on all but the kth part, and then validating on the kth part, iterating over k=1,..,K.

Note: When K = n, we call this **leave-one-out** cross-validation, because we leave out one data point at a time.

K-fold cross validation: Procedure

- Randomly divide the set {1,..,n} into K subsets (i.e., folds) of roughly equal size, $F_1,...,F_K$.
- For k=1,..,K:
  - Consider training on F_{-k} and validating on F_k.
  - For each value of the tuning parameter $\{\theta_1,..,\theta_m\}$ compute the estimate $\hat{f}_\theta^{-k}$ on the training set, and record the total error on the validation set:

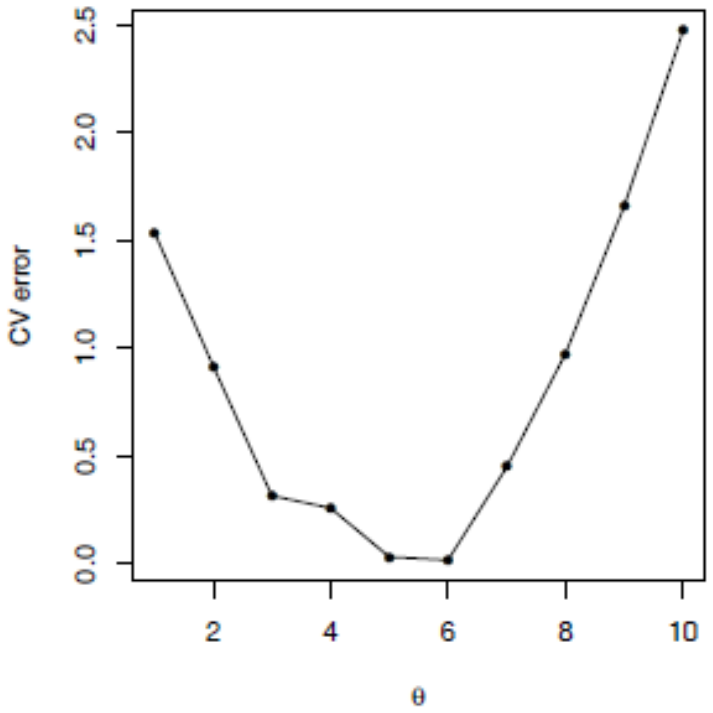
$$e_k(\theta) = \sum_{i \in F_k} (y_i - \hat{f}_\theta^{-k}(x_i))^2$$

- For each tuning parameter value , compute the average error over all folds,

$$CV(\theta) = \frac{1}{n} \sum_{k=1}^{K} e_k(\theta)$$

K-Fold cross validation

Having done this, we get a cross-validation error curve CV(θ) (this curve is a function of θ):

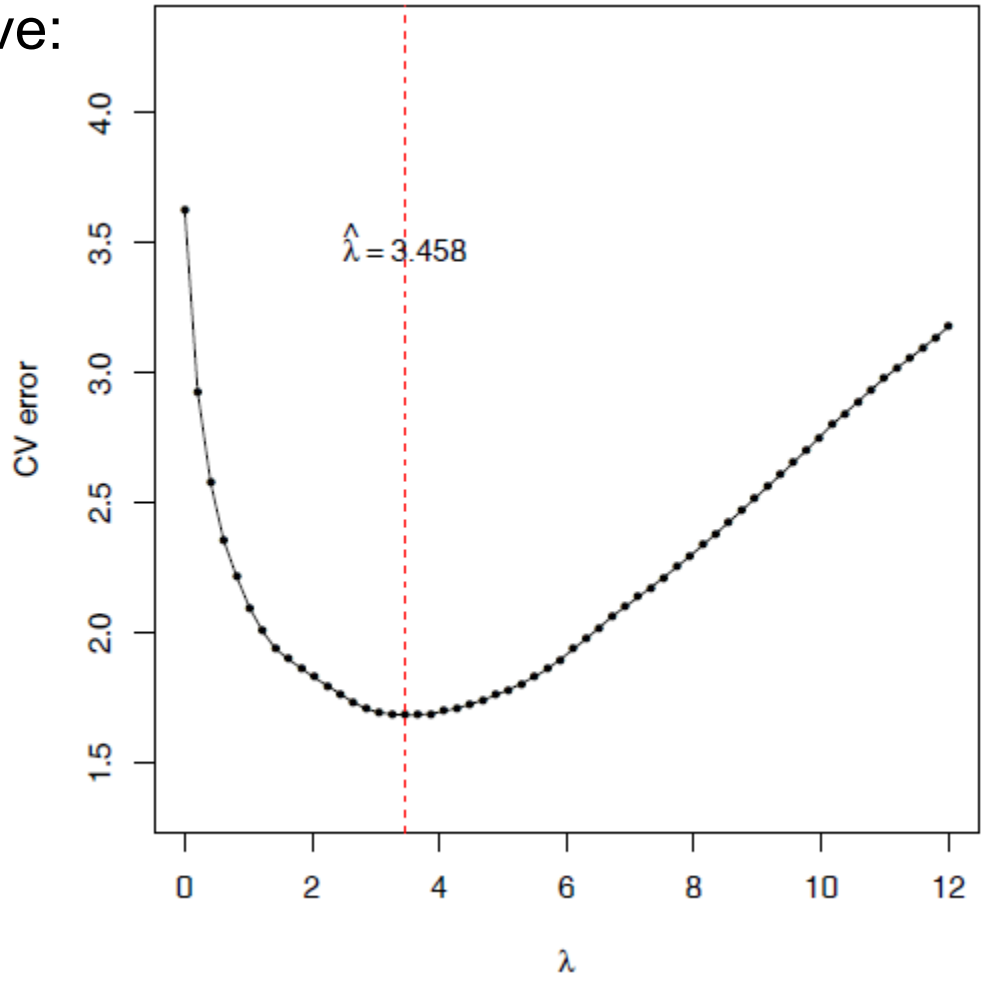


Choose the value of tuning parameter that minimizes this curve.

Example: choosing λ for the lasso

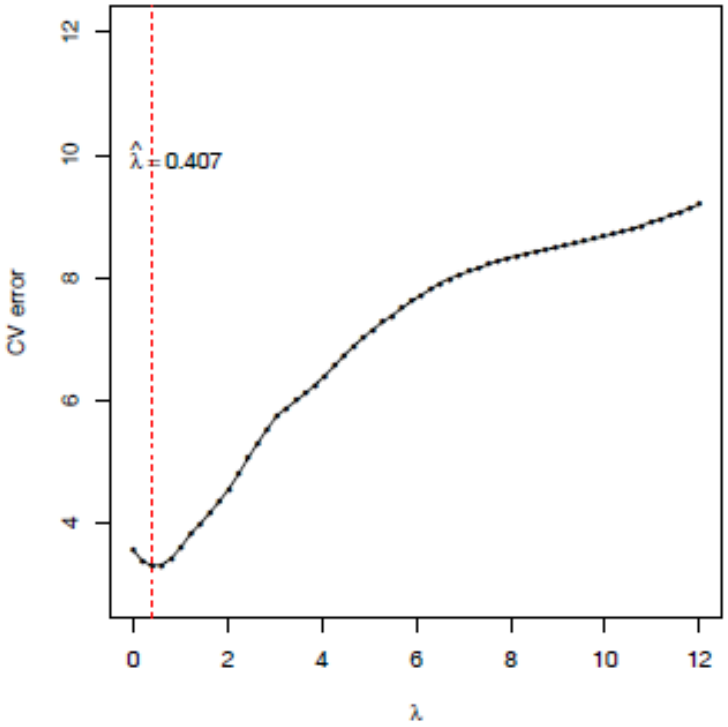
Recall our running example from last time: n = 50, p = 30, and the true model is linear with 10 nonzero coefficients. Consider the lasso estimator and use 5-fold cross-validation.

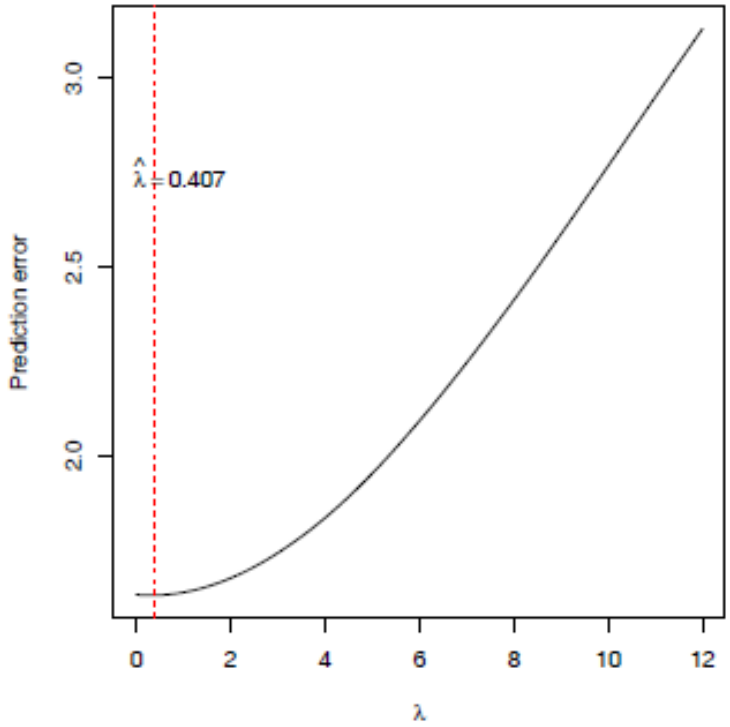
The resulting cross-validation error curve:



What happens if we really shouldn't be shrinking in the first place? We'd like cross-validation, our automated tuning parameter selection procedure, to choose a small value of λ.

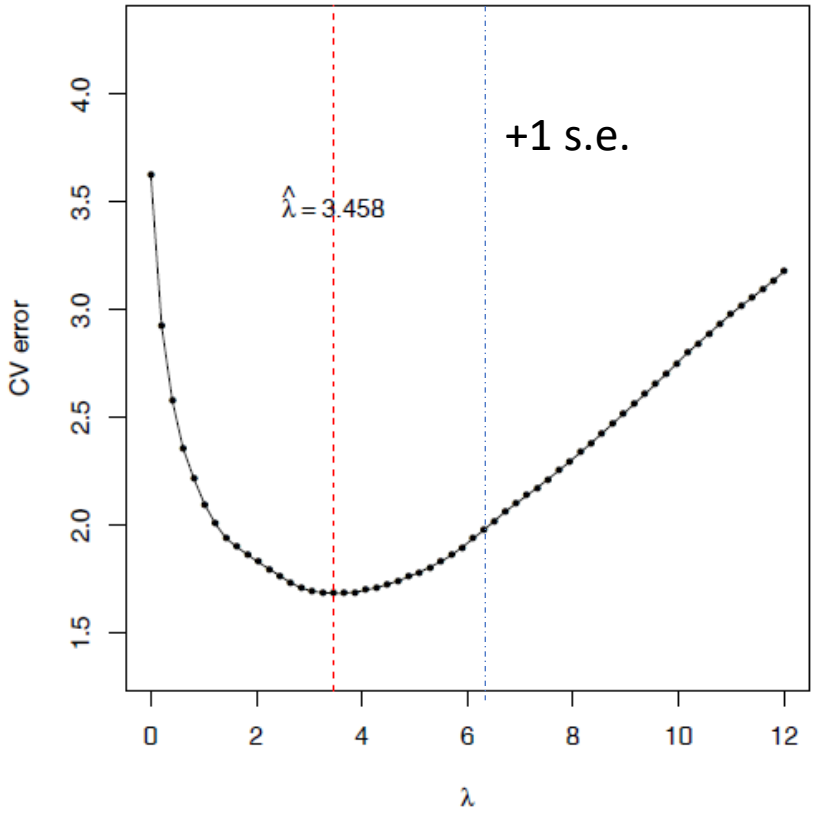
Recall the example where n = 50, p = 30, and the true model is linear with all moderately large coefficients:





Note

- The test error is a random variable subject to uncertainty.
- As such we can also calculate its standard error (the standard error of the test error).
- In some cases we might opt for a model that is one s.e. away from the the minimization error. This is a conservative strategy.



What to do next?

- After having used cross-validation to choose a value of the tuning parameter we now fit our estimator to the entire training set (x_i , y_i), i=1,..,n, using the tuning parameter value.

- Example: In the lasso case, we solve the problem on all of the training data, with λ=0.407.

- We can then use this estimator to make future predictions.

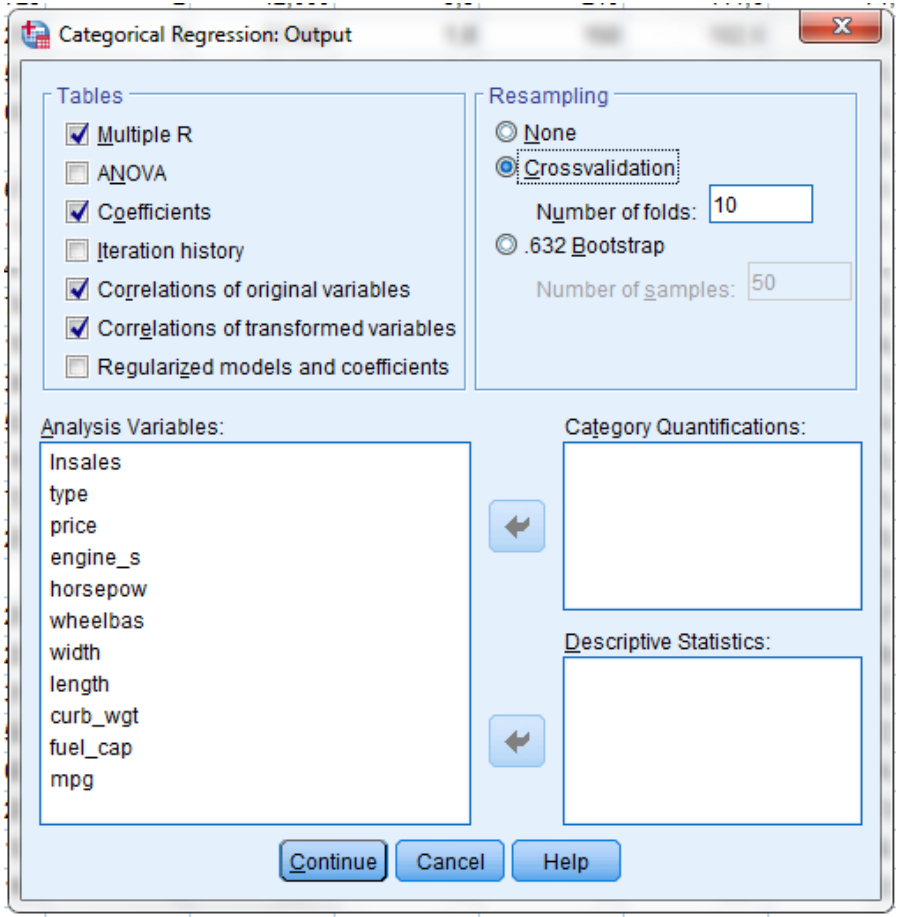
Recap: cross-validation

- **Training error**, the error of an estimator as measured by the data used to fit it, is not a good surrogate for prediction error. It just keeps decreasing with increasing model complexity.

- **Cross-validation**, on the other hand, much more accurately reflects prediction error. If we want to choose a value for the tuning parameter of a generic estimator (and minimizing prediction error is our goal), cross-validation is a standard tool.

- We usually pick the tuning parameter that minimizes the cross-validation error curve.

Ridge with cross validation

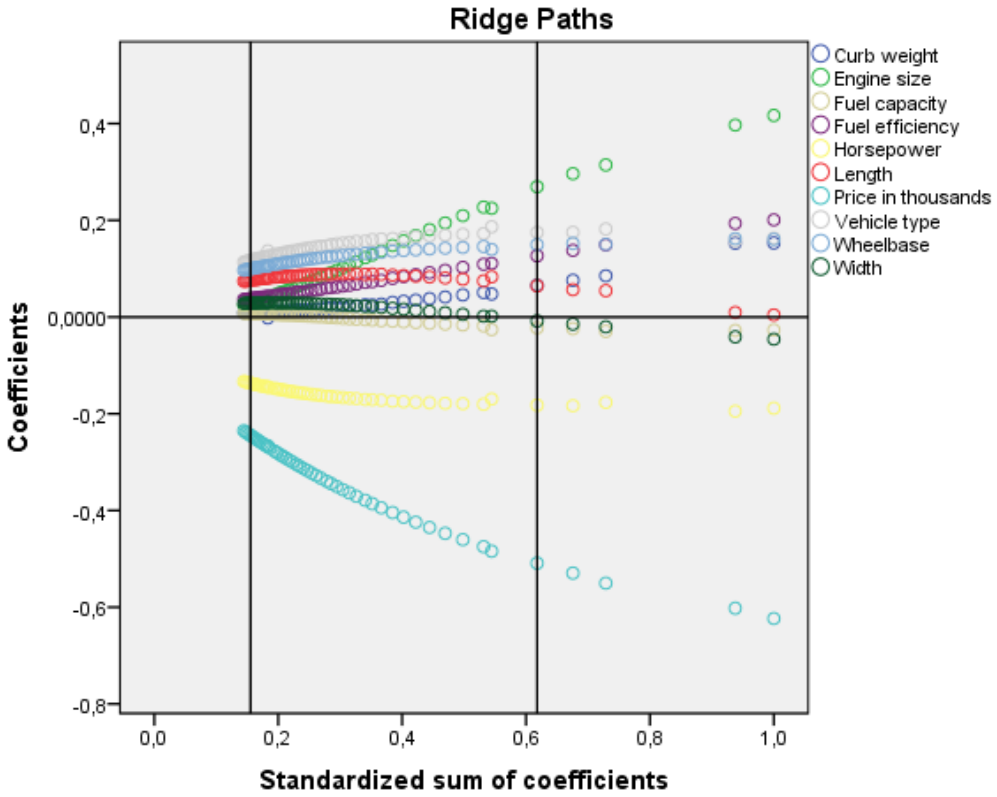


Model Summary

	Multiple R	R Square	Adjusted R Square	Regularization "R Square" (1-Error)	Apparent Prediction Error	Expected Prediction Error		
						Estimate[a]	Std. Error	N
Standardized Data	,653	,427	,386	,358	,642	,671	,075	152
Raw Data					1,400	1,457	,161	

Penalty ,920
Dependent Variable: Log-transformed sales
Predictors: Vehicle type Price in thousands Engine size Horsepower Wheelbase Width Length Curb weight Fuel capacity Fuel efficiency

a. Mean Squared Error (10 fold Cross Validation).



Coefficients

	Standardized Coefficients				
	Beta	Bootstrap (1000) Estimate of Std. Error	df	F	Sig.
Vehicle type	,117	,024	1	24,429	,000
Price in thousands	-,245	,020	1	144,060	,000
Engine size	,029	,025	1	1,337	,250
Horsepower	-,137	,021	1	41,433	,000
Wheelbase	,100	,023	1	18,757	,000
Width	,029	,026	1	1,219	,271
Length	,076	,026	1	8,293	,005
Curb weight	,009	,020	1	,204	,652
Fuel capacity	,007	,028	1	,068	,794
Fuel efficiency	,038	,023	1	2,744	,100

Dependent Variable: Log-transformed sales

X-axis reference lines at optimal model and at most parsimonious model within 1 Std. Error.

IBM SPSS

New dataset created with coefficients for each penalty level with EPE and APE

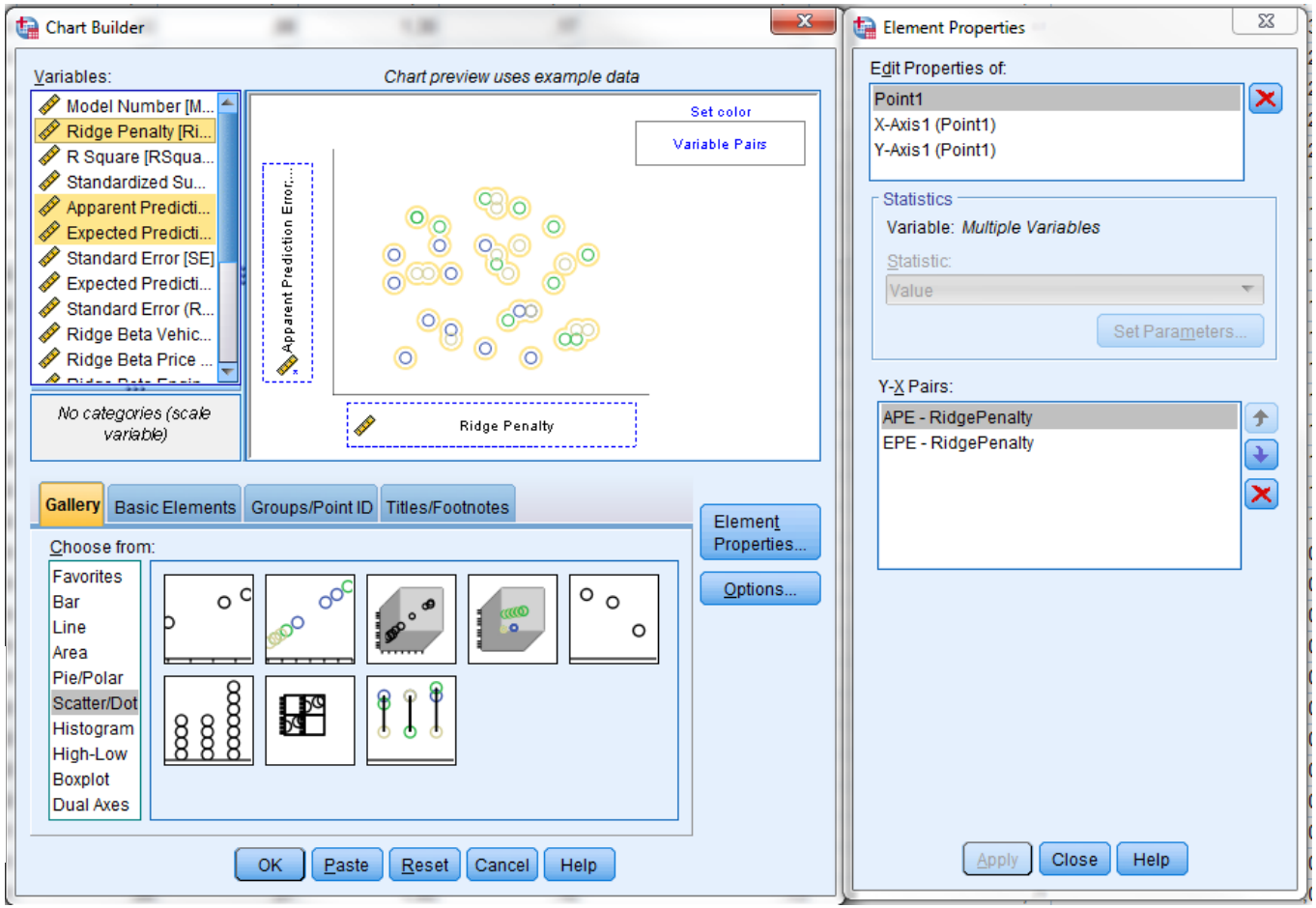
*Untitled6 [DataSet12] - IBM SPSS Statistics Data Editor

File Edit View Data Transform Analyze Direct Marketing Graphs Utilities Extensions Window Help

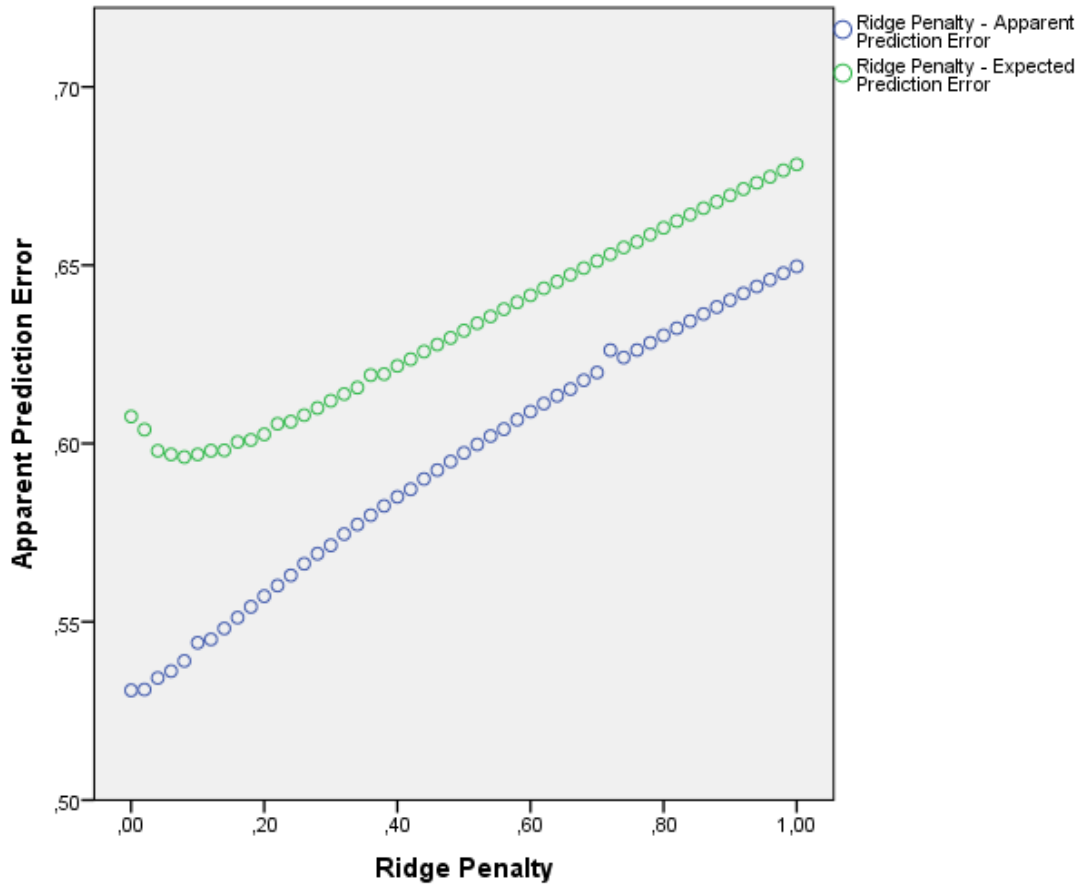
	ModelNumber	RidgePenalty	RSquare	StdSumCoeff	APE	EPE	SE	EPE_Raw	SE_Raw
1	1,00	,00	,47	1,00	,53	,61	,08	1,32	,1
2	2,00	,02	,47	,94	,53	,60	,08	1,31	,1
3	3,00	,04	,47	,73	,53	,60	,08	1,30	,1
4	4,00	,06	,46	,68	,54	,60	,08	1,30	,1
5	5,00	,08	,46	,62	,54	,60	,08	1,30	,1
6	6,00	,10	,46	,54	,54	,60	,08	1,30	,1
7	7,00	,12	,45	,53	,55	,60	,08	1,30	,1
8	8,00	,14	,45	,50	,55	,60	,07	1,30	,1
9	9,00	,16	,45	,47	,55	,60	,07	1,31	,1
10	10,00	,18	,45	,44	,55	,60	,07	1,31	,1
11	11,00	,20	,44	,42	,56	,60	,07	1,31	,1
12	12,00	,22	,44	,40	,56	,61	,07	1,32	,1
13	13,00	,24	,44	,38	,56	,61	,07	1,32	,1
14	14,00	,26	,43	,37	,57	,61	,07	1,32	,1
15	15,00	,28	,43	,35	,57	,61	,07	1,33	,1
16	16,00	,30	,43	,34	,57	,61	,07	1,33	,1

Plot APE and EPE

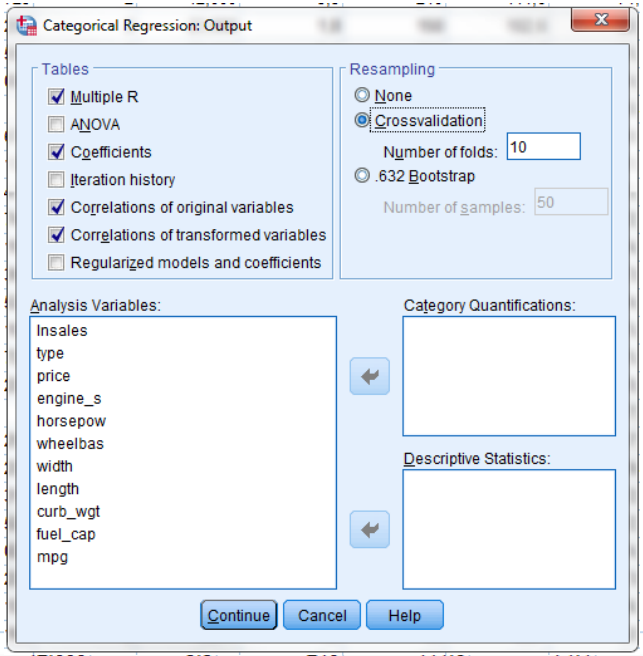
- Graphs > Chart builder

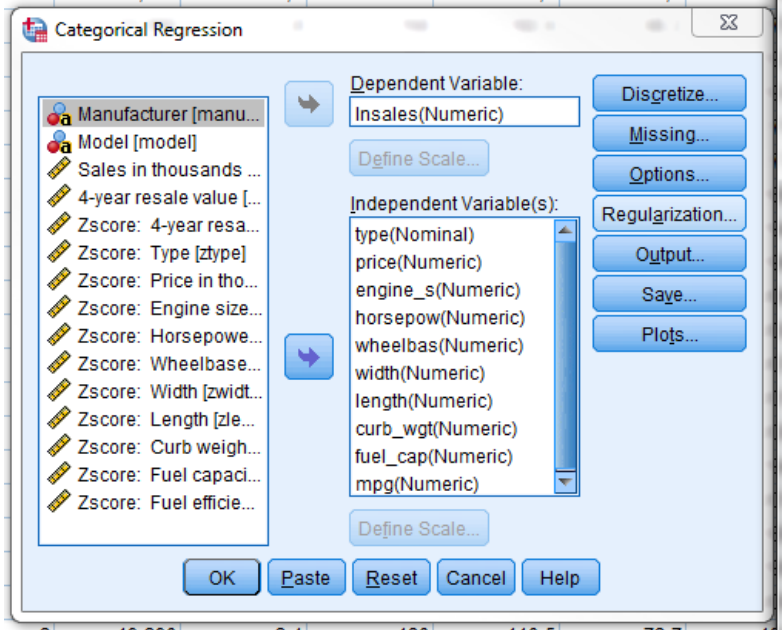


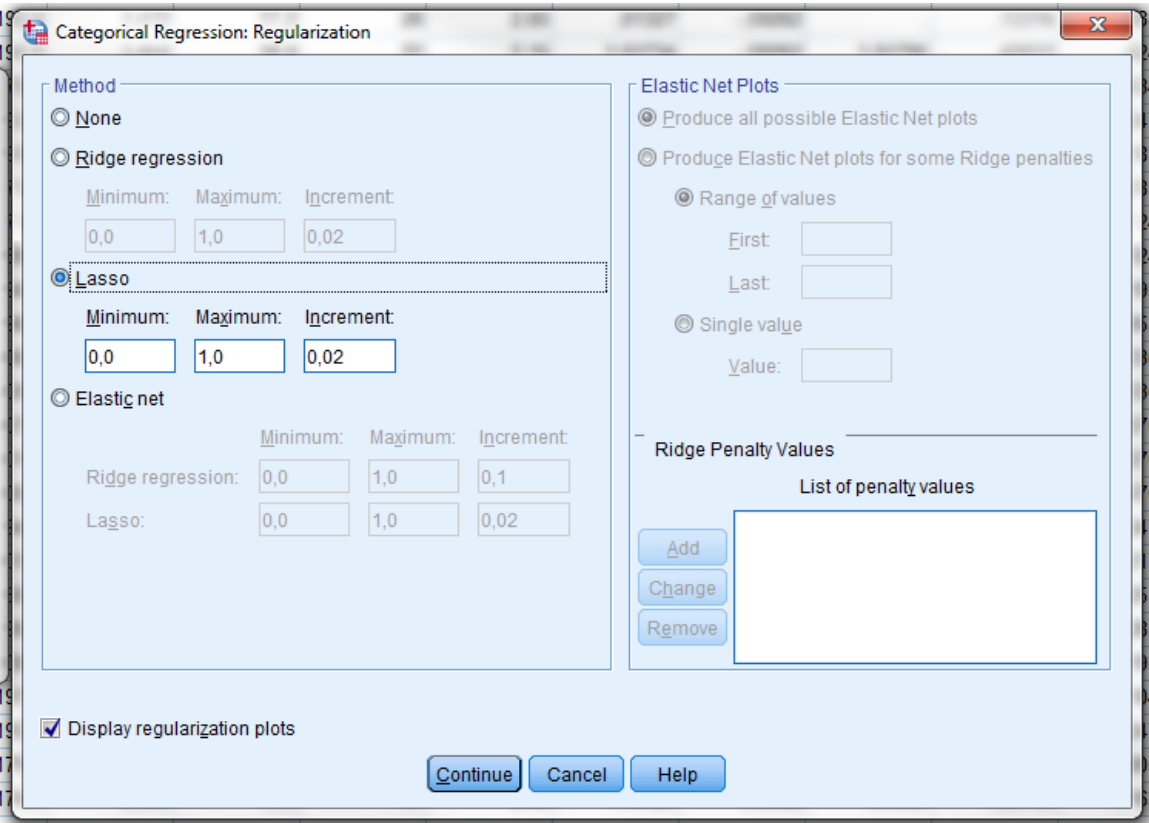
APE vs. EPE



Lasso with crossvalidation







2	19,390	3,4	180	110,5	72,7	19
2	24,340	3,8	200	101,1	74,1	19
2	45,705	5,7	345	104,5	73,6	17
2	13,960	1,8	120	97,1	66,7	17

Model Summary

	Multiple R	R Square	Adjusted R Square	Regularization "R Square" (1-Error)	Apparent Prediction Error	Expected Prediction Error		
						Estimate[a]	Std. Error	N
Standardized Data	,642	,412	,400	,358	,642	,667	,077	152
Raw Data					1,400	1,448	,167	

Penalty ,320
Dependent Variable: Log-transformed sales
Predictors: Vehicle type Price in thousands Engine size Horsepower Wheelbase Width Length Curb weight Fuel capacity Fuel efficiency

a. Mean Squared Error (10 fold Cross Validation).

Coefficients

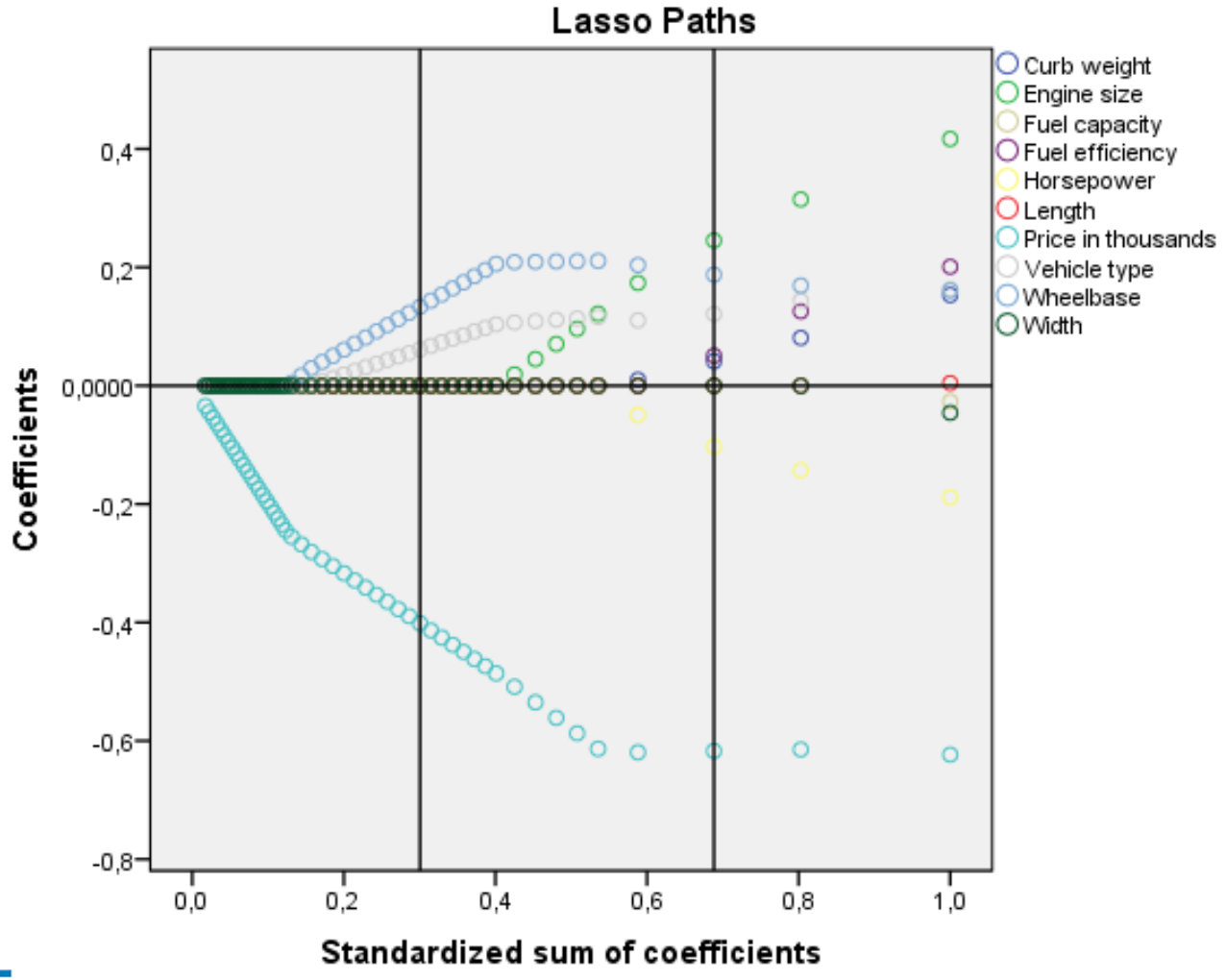
	Standardized Coefficients				
	Beta	Bootstrap (1000) Estimate of Std. Error	df	F	Sig.
Vehicle type	,061	,052	1	1,392	,240
Price in thousands	-,402	,059	1	45,989	,000
Engine size	,000	,005	0	,000	.
Horsepower	,000	,003	0	,000	.
Wheelbase	,133	,065	1	4,227	,042
Width	,000	,008	0	,000	.
Length	,000	,041	0	,000	.
Curb weight	,000	,003	0	,000	.
Fuel capacity	,000	,008	0	,000	.
Fuel efficiency	,000	,000	0	.	.

Dependent Variable: Log-transformed sales

Correlations and Tolerance

	Correlations			Tolerance	
	Zero-Order	Partial	Part	After Transformation	Before Transformation
Vehicle type	,277	,118	,086	,269	,269
Price in thousands	-,535	-,354	-,276	,198	,198
Engine size	-,077	,223	,167	,166	,166
Horsepower	-,381	-,096	-,070	,142	,142
Wheelbase	,231	,107	,078	,233	,233
Width	,063	-,038	-,028	,386	,386
Length	,178	,006	,004	,189	,189
Curb weight	-,008	,070	,051	,133	,133
Fuel capacity	,025	-,015	-,011	,181	,181
Fuel efficiency	,095	,120	,088	,205	,205

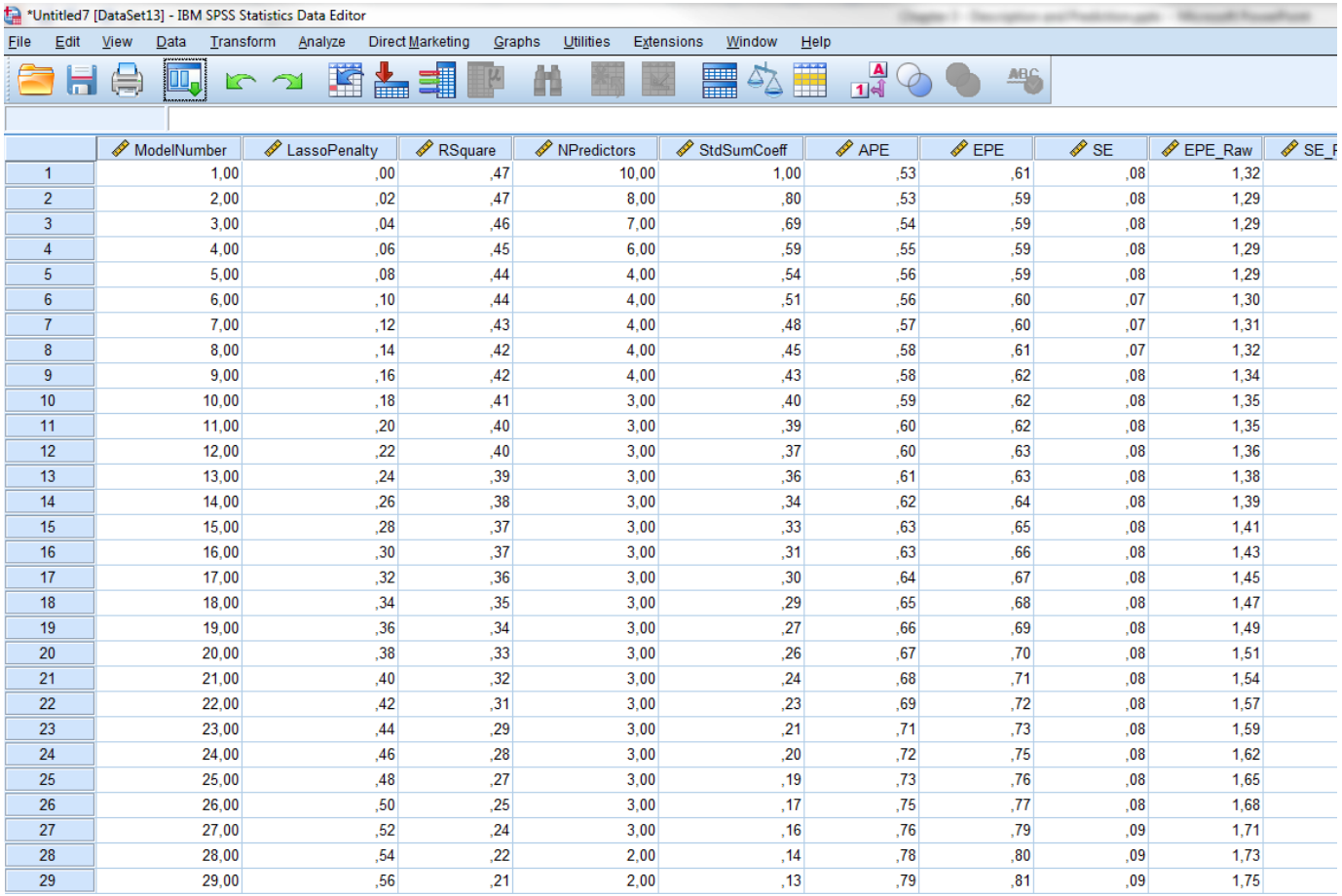
Dependent Variable: Log-transformed sales



X-axis reference lines at optimal model and at most parsimonious model within 1 Std. Error.

IBM SPSS

New dataset created with coefficients for each penalty level with EPE and APE



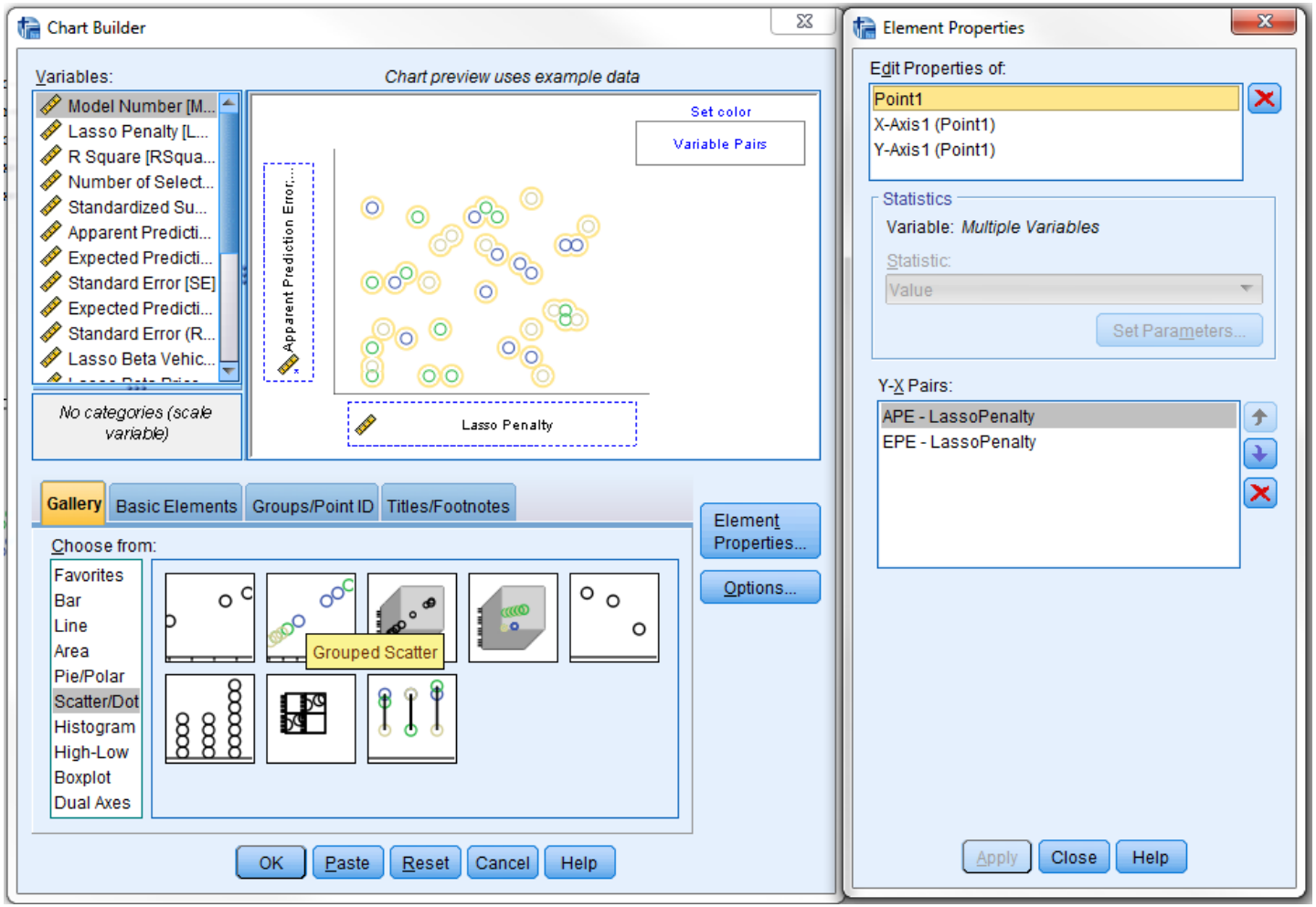
	ModelNumber	LassoPenalty	RSquare	NPredictors	StdSumCoeff	APE	EPE	SE	EPE_Raw	SE_R
1	1,00	,00	,47	10,00	1,00	,53	,61	,08	1,32	
2	2,00	,02	,47	8,00	,80	,53	,59	,08	1,29	
3	3,00	,04	,46	7,00	,69	,54	,59	,08	1,29	
4	4,00	,06	,45	6,00	,59	,55	,59	,08	1,29	
5	5,00	,08	,44	4,00	,54	,56	,59	,08	1,29	
6	6,00	,10	,44	4,00	,51	,56	,60	,07	1,30	
7	7,00	,12	,43	4,00	,48	,57	,60	,07	1,31	
8	8,00	,14	,42	4,00	,45	,58	,61	,07	1,32	
9	9,00	,16	,42	4,00	,43	,58	,62	,08	1,34	
10	10,00	,18	,41	3,00	,40	,59	,62	,08	1,35	
11	11,00	,20	,40	3,00	,39	,60	,62	,08	1,35	
12	12,00	,22	,40	3,00	,37	,60	,63	,08	1,36	
13	13,00	,24	,39	3,00	,36	,61	,63	,08	1,38	
14	14,00	,26	,38	3,00	,34	,62	,64	,08	1,39	
15	15,00	,28	,37	3,00	,33	,63	,65	,08	1,41	
16	16,00	,30	,37	3,00	,31	,63	,66	,08	1,43	
17	17,00	,32	,36	3,00	,30	,64	,67	,08	1,45	
18	18,00	,34	,35	3,00	,29	,65	,68	,08	1,47	
19	19,00	,36	,34	3,00	,27	,66	,69	,08	1,49	
20	20,00	,38	,33	3,00	,26	,67	,70	,08	1,51	
21	21,00	,40	,32	3,00	,24	,68	,71	,08	1,54	
22	22,00	,42	,31	3,00	,23	,69	,72	,08	1,57	
23	23,00	,44	,29	3,00	,21	,71	,73	,08	1,59	
24	24,00	,46	,28	3,00	,20	,72	,75	,08	1,62	
25	25,00	,48	,27	3,00	,19	,73	,76	,08	1,65	
26	26,00	,50	,25	3,00	,17	,75	,77	,08	1,68	
27	27,00	,52	,24	3,00	,16	,76	,79	,09	1,71	
28	28,00	,54	,22	2,00	,14	,78	,80	,09	1,73	
29	29,00	,56	,21	2,00	,13	,79	,81	,09	1,75	

IBM SPSS

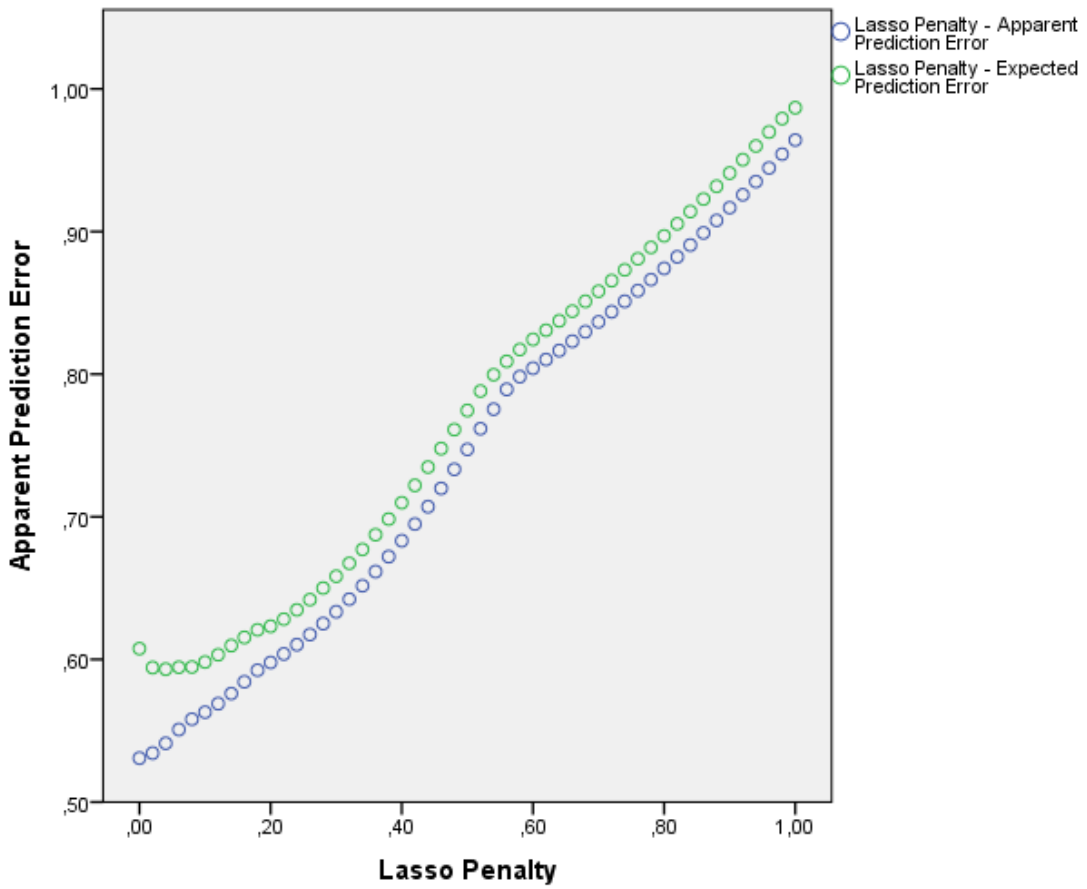
Lisbon School of Economics & Management

Plot APE and EPE

- Graphs > Chart builder



APE vs. EPE


Lasso Penalty - Apparent Prediction Error
Lasso Penalty - Expected Prediction Error
Apparent Prediction Error
1,00
,90
,80
,70
,60
,50
,00
,20
,40
,60
,80
1,00
Lasso Penalty