MATH5743M: Statistical Learning - Lecture 5

Dr Seppo Virtanen, s.virtanen@leeds.ac.uk

Semester 2: 21 Feb 2022

Outline

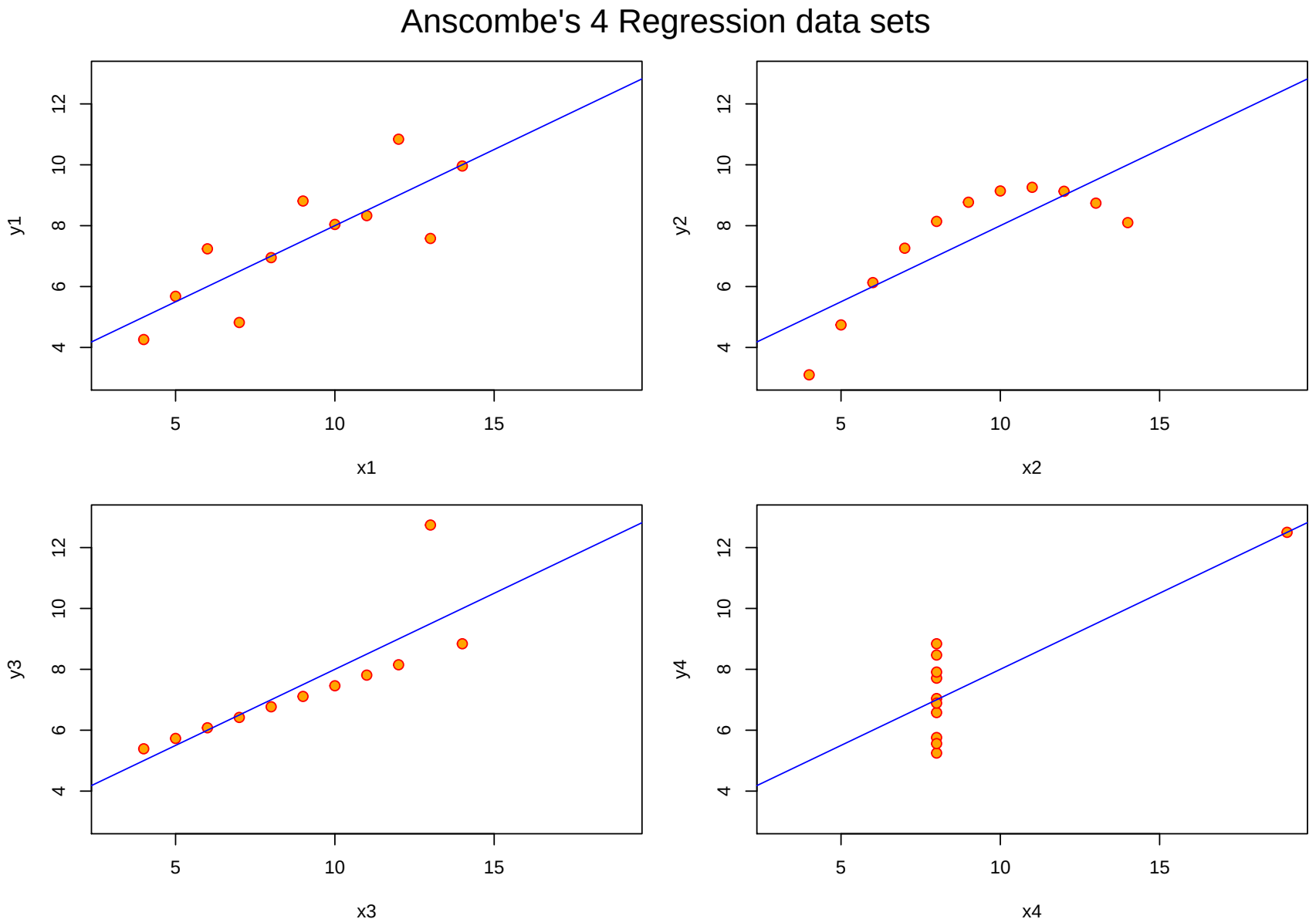
By far, we know how to estimate the parameters of a model that we have chosen. But how do we choose the right model for our data?

It is impossible to do statistics without making assumptions. Statistics is always about quantifying rigorously what is implied by both the data and our own prior knowledge.

For example, how many and which inputs should we use for multiple linear regression preventing overfitting and ensuring interpretability?

‘Anscombe’s Quartet’

Four data sets that give almost identical results when using linear regression.



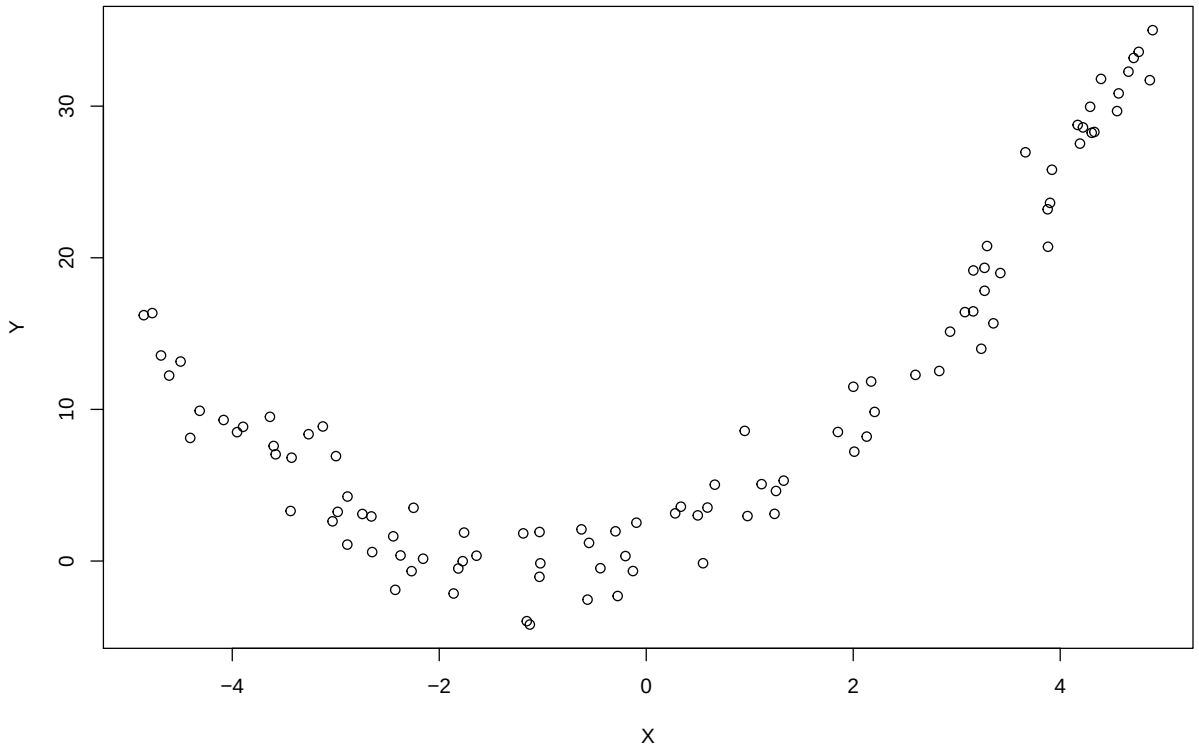
Feature selection

Deciding which inputs to use in a model is described as *feature selection*: we are selecting 'features' of the data that are important for predicting the output we are interested in.

We want to identify which of the inputs already in the data are useful for predicting the output.

The problem of overfitting

Imagine you are presented with the data shown in the plot below:



We could use polynomial transformations of X as extra inputs in a linear regression. The curve looks quite like a quadratic, but it could also be cubic or even quartic; how should you decide which to use?

The problem of overfitting

The true relationship is quadratic:

```
X = runif(100)*10 -5
Y = 1 + 2*X + X^2 + rnorm(100, 0, 2)
```

Lets see what happens when we try to fit a simple linear regression, a quadratic model and a cubic model.

```
mydata = data.frame(Y = Y, X1 = X, X2=X^2, X3 = X^3)
model1 = glm(Y ~ X, data = mydata)
model2 = glm(Y ~ X + X2, data = mydata)
model3 = glm(Y ~ X + X2 + X3, data = mydata)
```

The problem of overfitting

We can compare the likelihood of each model evaluated at the maximum-likelihood estimate parameters:

```
## [1] "log-likelihood for linear model:  -346.615003359587
```

```
## [1] "log-likelihood for quadratic model:  -206.925949487
```

```
## [1] "log-likelihood for cubic model:  -206.615069914414"
```

The problem of overfitting

'Over-fitting': more complicated models typically produce closer fits to the data you use to learn the model parameters than less complex ones.

We usually want our models to be as simple as possible, while still capturing the important features of the data.

The problem of overfitting

Consider first the quadratic model:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \epsilon \tag{1}$$

Compare this to the cubic model:

$$y = \alpha_0 + \alpha_1 x + \alpha_2 x^2 + \alpha_3 x^3 + \epsilon \tag{2}$$

The maximum likelihood of the cubic model must be at least as large of the maximum likelihood for the quadratic model, and will usually be greater, i.e.

$$\max \mathcal{L}_{\text{cubic}} \geq \max \mathcal{L}_{\text{quadratic}} \tag{3}$$

Therefore we cannot use the model likelihood alone to select which inputs to use in our model, since the likelihood will always increase if we add more inputs.

Akaike Information Criterion (AIC)

AIC is defined as:

$$\text{AIC} = 2k - 2\log L^*, \tag{4}$$

where k is the number of parameters and $\log L^*$ is the usual maximum log-likelihood.

Penalised likelihood that explicitly 'punishes' the extra parameters.

The AIC for a model fitted using **glm** is conveniently provided as part of the **summary** command.

Cross-validation

An alternative method for avoiding problems of overfitting is to test how well a model can predict *unseen* data, i.e. data that has not been used to determine the model parameters. We use the unseen data to validate the model that has been fitted.

Intuitive since the goal of statistical learning is often to develop a model that is useful for prediction.

Cross-validation

- Identify or create two subsets of the data. Label one the 'training' data and the other the 'test' data.
- Use the training data to fit the statistical model you want to test, i.e. to learn the parameters of that model
- Use the fitted model to predict the outputs of the test data, based on the inputs of the test data
- Evaluate the accuracy of those predictions
- Potentially repeat the process with two new data subsets

Example

```
set.seed(11)
X1 = runif(200)-0.5;
X2 = runif(200)-0.5
X3 = runif(200)-0.5
Y = 2*X1 + 1*X3 + rnorm(200, 0, 1)
mydata = data.frame(X1, X2, X3, Y)
idx = sample(1:200, 100) #sample 100 points in 1...200 wit
train_data = mydata[idx, ]; test_data = mydata[-idx, ]
```

Example

7 possible combinations of inputs (if we exclude the possibility of having no inputs):

1. X1
2. X2
3. X3
4. X1 & X2
5. X1 & X3
6. X2 & X3
7. X1, X2 & X3

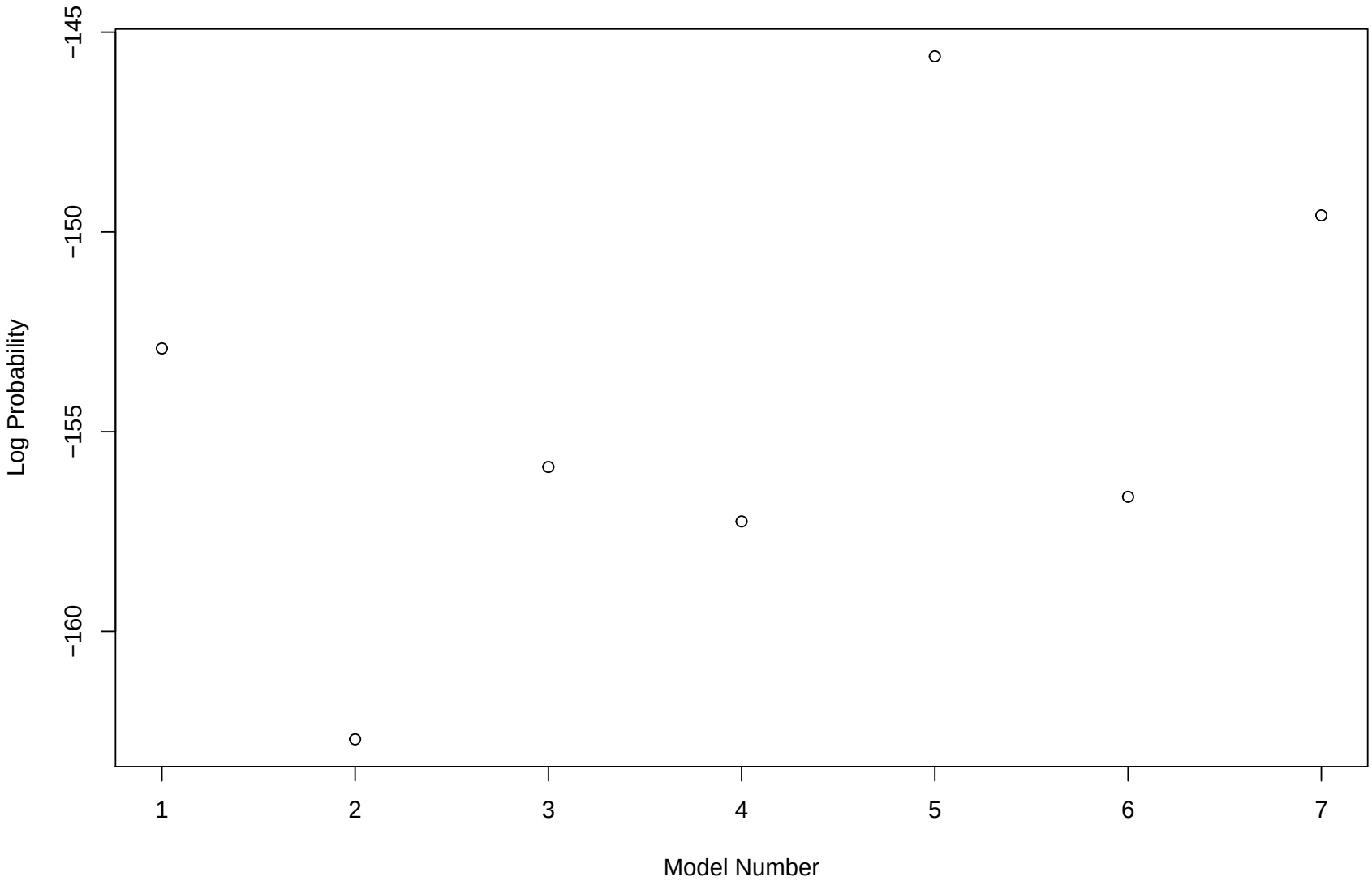
```
#List all possible models
formulas = c("Y ~ X1", "Y ~ X2", "Y ~ X3",
             "Y ~ X1 + X2", "Y ~ X1 + X3", "Y ~ X2 + X3",
             "Y ~ X1 + X2 + X3")
```

Example

```
predictive_log_likelihood = rep(NA, length(formulas))
for (i in 1:length(formulas)){
#First fit a linear regression model
current_model = glm(formula = formulas[i], data = train_dat

#Extract the 'dispersion parameter' from the model
sigma = sqrt(summary(current_model)$dispersion)

#Evaluate the probability of the test outputs
#Get the predicted mean for each new data point
ypredict_mean = predict(current_model, test_data)

  #Predictive log probability by summing the
  #log probability of each output in the test data
  predictive_log_likelihood[i] = sum(dnorm(test_data$Y,
                        ypredict_mean, sigma, log=TRUE))
}
```

Example

```
plot(1:length(formulas), predictive_log_likelihood,
     xlab="Model Number", ylab="Log Probability")
```

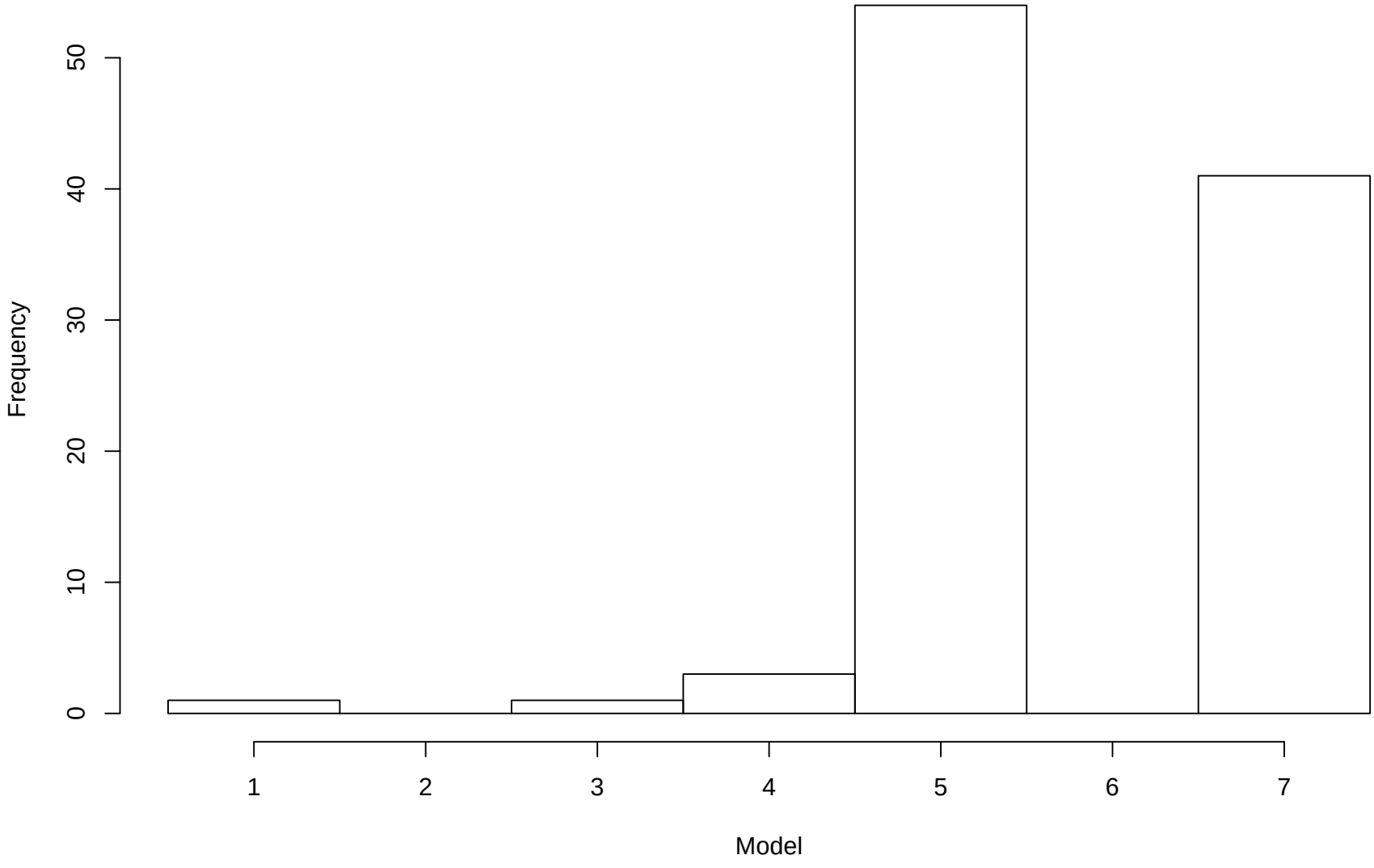


Example

Reduce the sensitivity to the particular split we made between training and test data. A simple way to do this is to repeat the whole procedure many times, choosing a new random split in the data each time.

Repeat the whole process above 100 times (I'm choosing 100 at random here!). Each time I'll record which model 'wins' - i.e. which model has the best predictive log-likelihood, as model 5 did above. I'll record for each model the number of times it 'wins' and plot that as a bar chart. This will give us a good idea of how robust the result above is.

Example



Example

Notice two things: (i) overall model 5 is favoured but (ii) there are many occasions on which model 7 wins.

In general, the more splits you can do the better, so do as many as you can that time and the power of your computer will allow.