## Week 10 Hour 1

## Shapiro-Wilks Test (from last time)

## Cross-Validation

## Week 10 Hour 2

## Missing Data

## Shapiro-Wilks Test

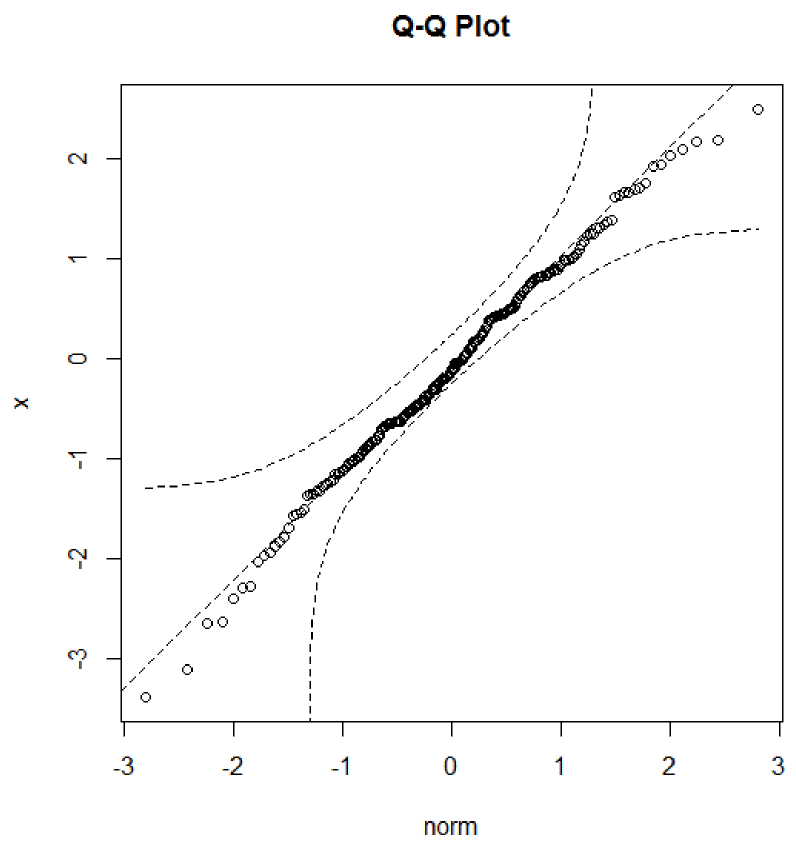
The Shapiro test is a hypothesis test for normality.
It works like other tests the Kruskal-Wallis and the Bartlett tests for equal variance.

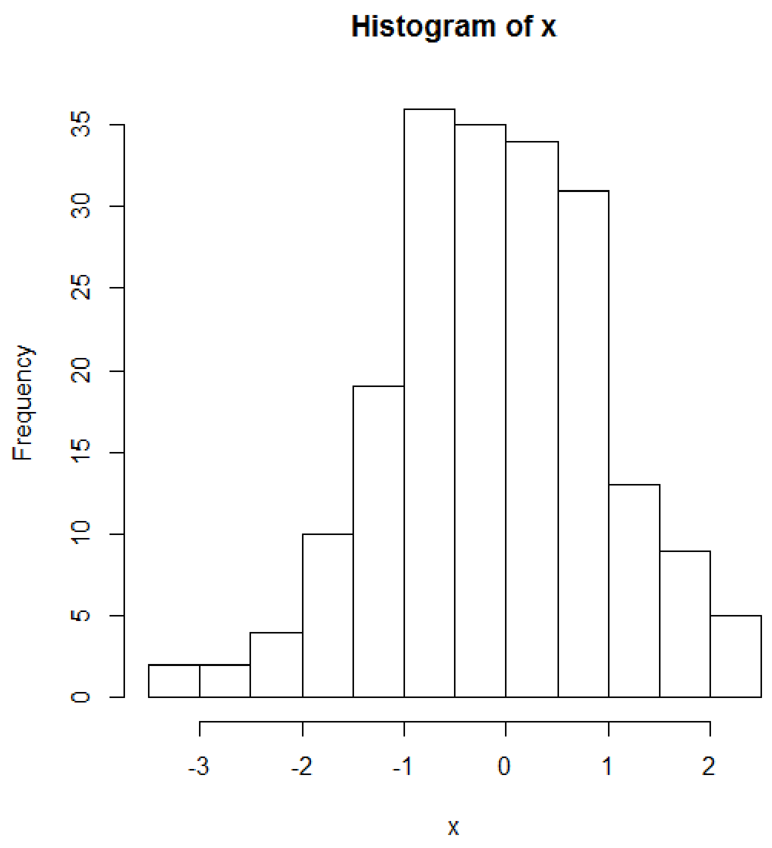
Your null hypothesis is the no-problem scenario. In the Shapiro test's case, this is 'your data is normally distributed'.

- If the p-value is large, there is no evidence against normality.
- If the p-value is small, you have evidence of non-normality.

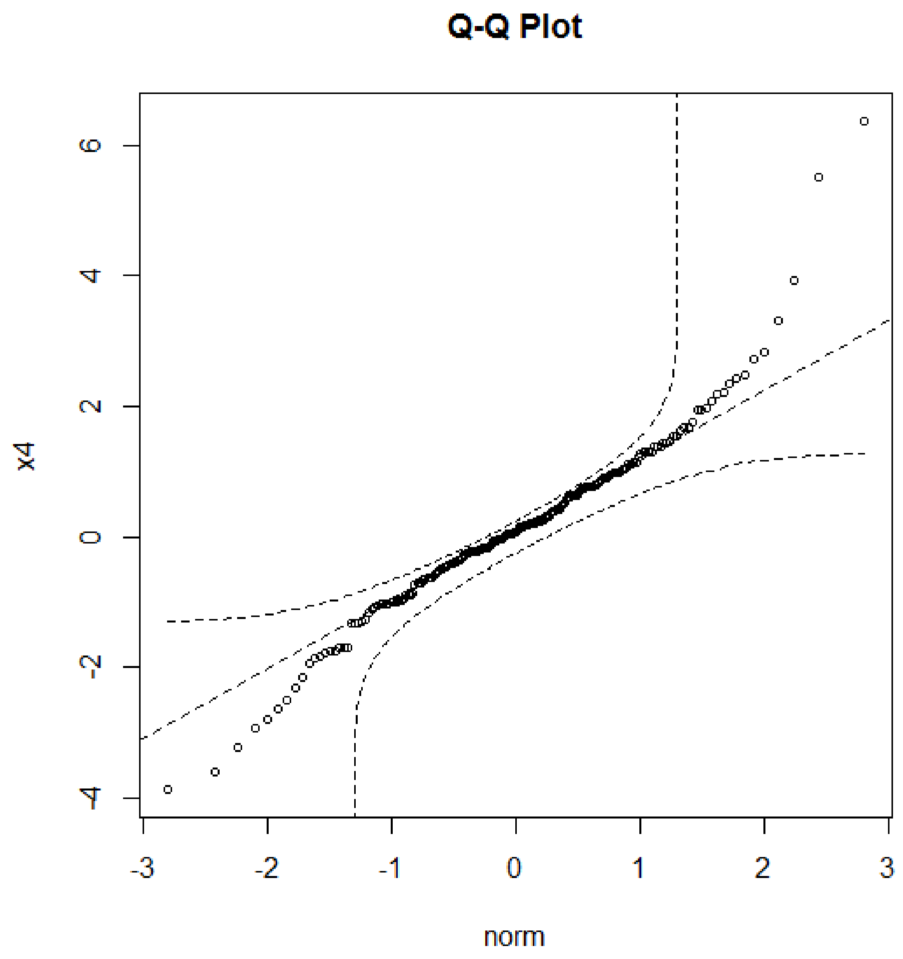
Recall the quantile-quantile plots from earlier:

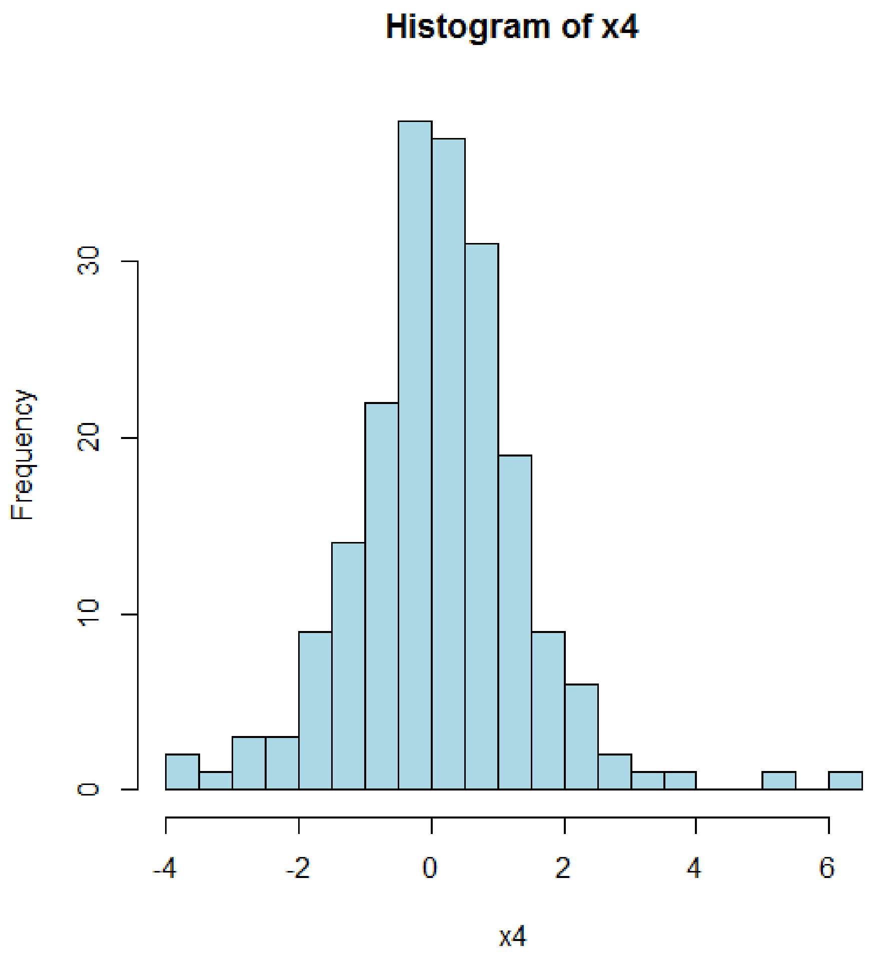
This is the Q-Q plot and histogram for a ***normal*** distribution.



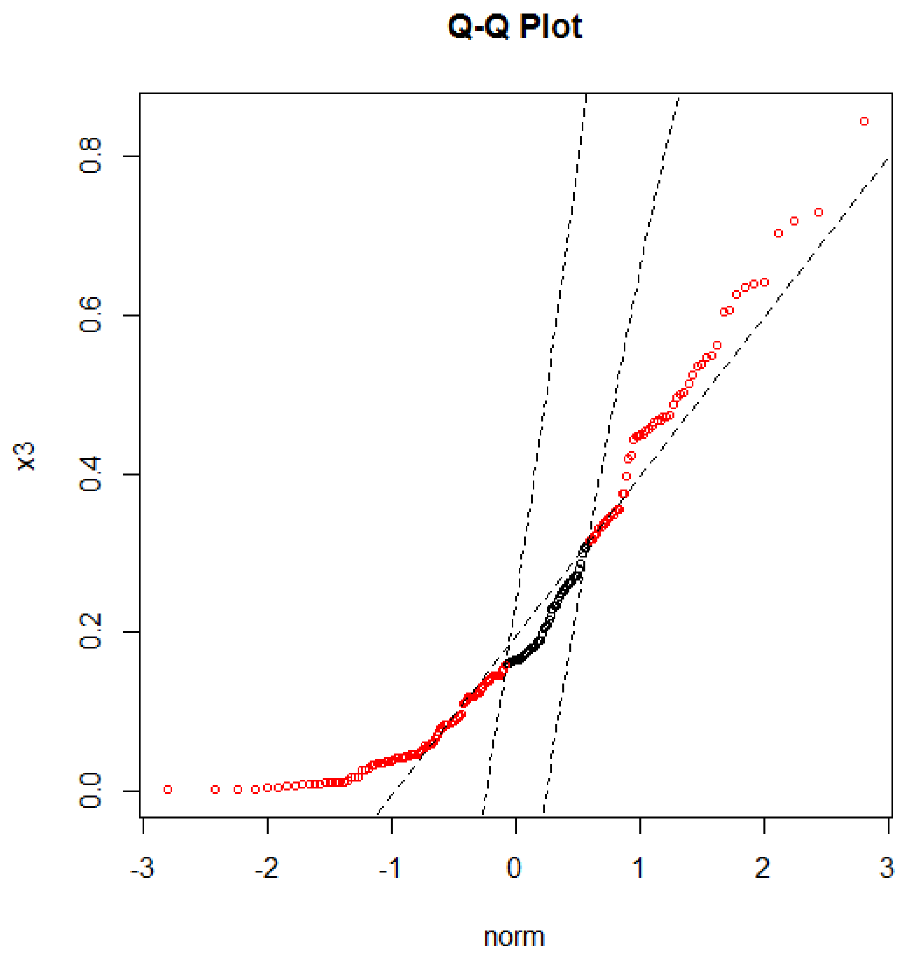


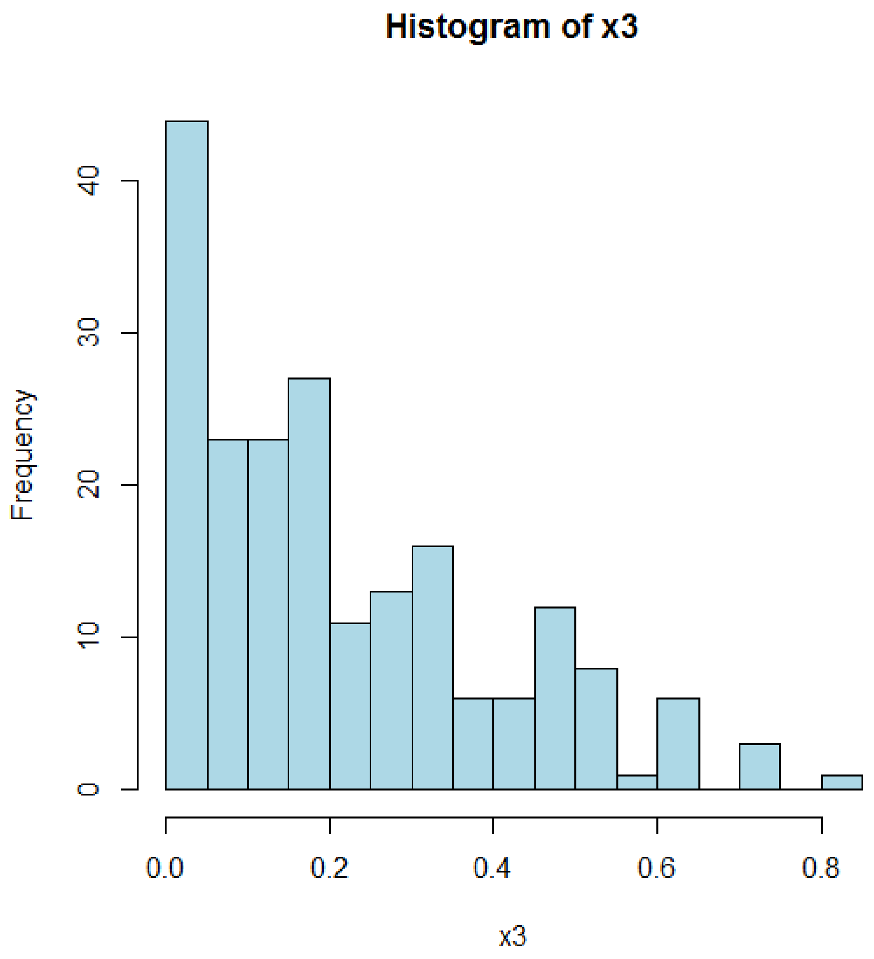
This is the Q-Q plot and histogram for a distribution with ***extreme values***. Specifically, the t-distribution with df=4.



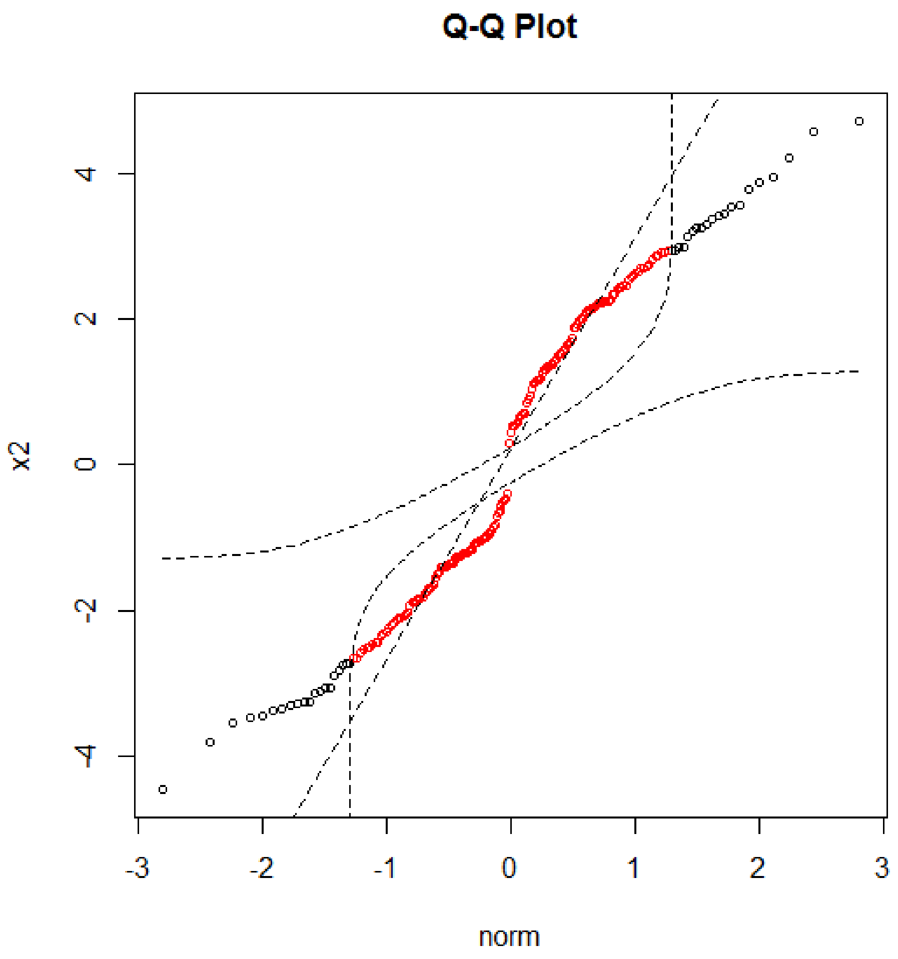


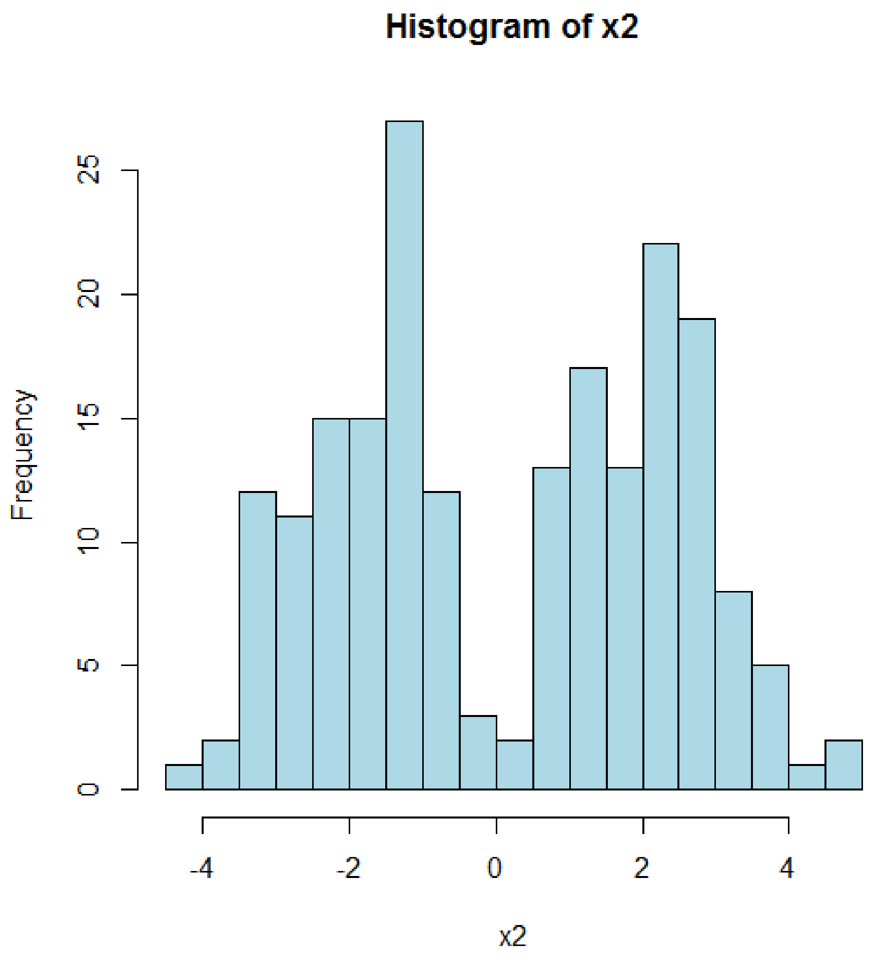
This is the Q-Q plot and histogram for a ***skewed*** distribution, meaning it has more extreme values on one side.



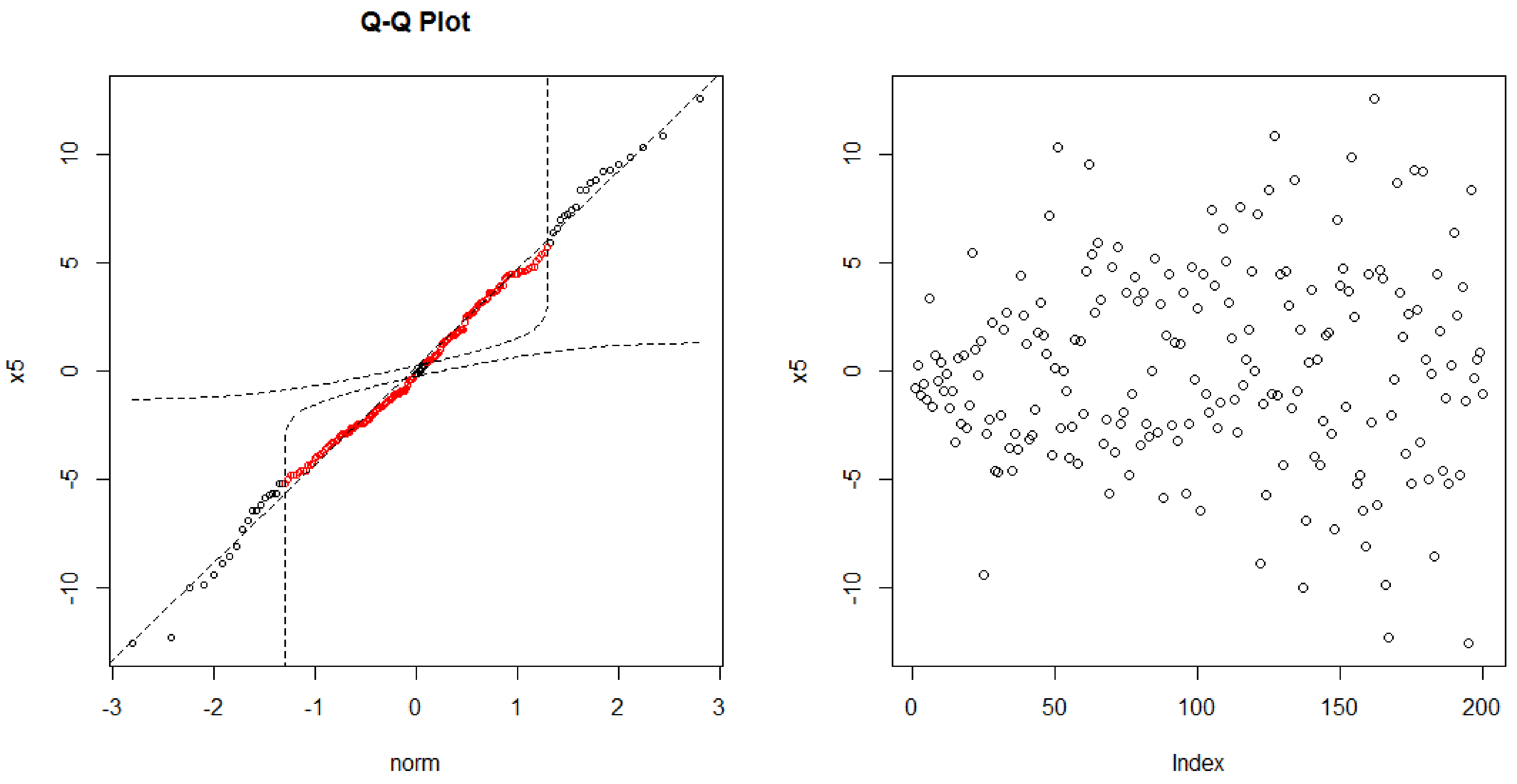


This is the Q-Q plot and histogram for a ***bimodal*** distribution, meaning it has two very distinct peaks or ***modes***.





Finally, this is the Q-Q plot and scatterplot for points that are from normal distributions with different amounts of variance.



In these five examples, the Shapiro-Wilk test produces the following p-values. For reference, the sample size n = 200.

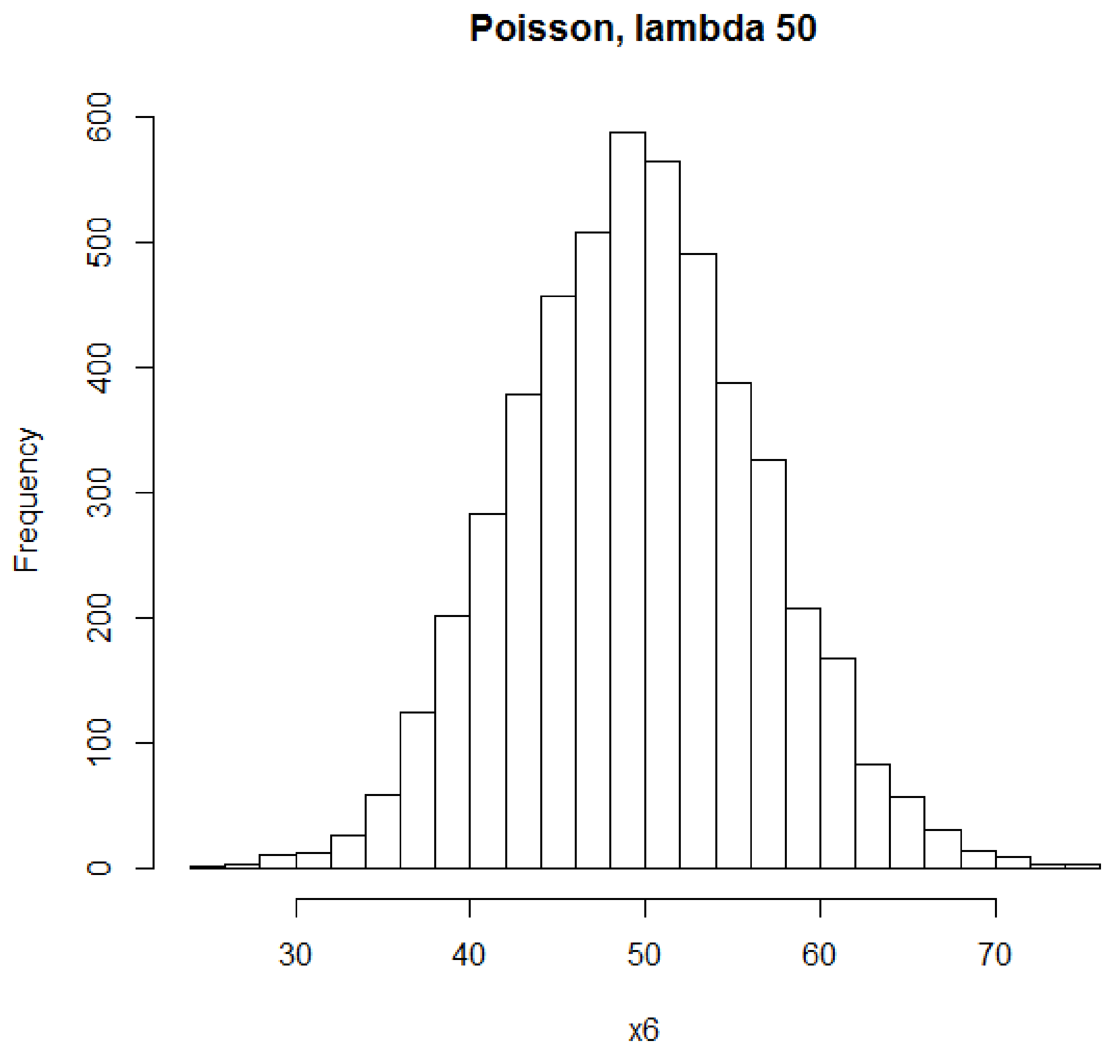
| Situation | Shapiro Test |
|---|---|
| Normal | $p = 0.6287$ |
| Extreme Values (t dist, df=4) | $p < 0.0001$ |
| Bimodal | $p < 0.0001$ |
| Skewed | $p < 0.0001$ |
| Normal, but with uneven variance | $p = 0.6335$ |

Like other hypothesis tests, sample size matters.

- The Shapiro test will be unable to find most non-normality in a small sample.

| Skewed Distribution | Shapiro Test |
|---|---|
| N = 10 | p = 0.2331 |
| N = 20 | p = 0.0128 |
| N = 30 | p = 0.0008 |
| N = 50 | p < 0.0001 |
| N = 200 | p < 0.0001 |

## It will detect minor non-normality when the sample is ***large***.



| Poisson | Shapiro Test |
|---|---|
| N = 10 | p = 0.8919 |
| N = 100 | p = 0.2454 |
| N = 1000 | p = 0.0182 |
| N = 2000 | p = 0.0016 |
| N = 5000 | p < 0.0001 |

Look hard enough and you'll see the impact of little things.

## Overfitting – The ‘Why’ of Cross Validation?

Recall from last week’s example problems that we interpreted the ‘potash’ variable in two very different ways.

1. As a numeric variable, from which we estimated coefficients for a linear term and a squared term.

2. As a categorical variable, from which we estimated coefficients as differences in mean values from some baseline.

Treating ‘potash’ as a categorical variable had some unique advantages:

1. We could identify the point where adding potash reduced yield instead of improving it.

2. We were able to fit the model much better, achieving a much higher R-squared.

…and one big disadvantage:
There was no way to apply the model when ‘potash’ was not exactly 0,1,2, 4, or 6 units.

A more general and shorter way to describe this drawback is ***overfitting***.

A model is overfitted if it is designed to fit the observed data very well, but would do a poor job of fitting (i.e. predicting) additional related data.

It doesn’t have to be impossible to apply a model to additional observations. If predictions of new observations are unreliable or invalid, you could still have overfitted.

Overfitting is one of the reasons for criteria such as AIC and BIC. Garbage terms in models may improve the fit to the observed data, but they offer no predictive power.

Selection based on AIC and BIC will produce models with fewer garbage terms than selection based on R-Squared.

Consider a dataset with many variables of random noise, where none of the explanatory variables have anything to do with each other, or with response variable.

An extreme example of overfitting is the saturated model.

A ***saturated*** model is one that fits the data perfectly, but has no degrees of freedom left for residuals, and thus no means of measuring its uncertainty.

In a saturated model, the r-squared is a perfect 1.000. There are no outliers, and all residuals are exactly zero.

## “Laboratory” vs “Real World”: The Pepsi Challenge

When offered a quick sip, tasters generally prefer the sweeter of two beverages – even if they prefer a less sweet beverage over the course of an entire can. Just because a taster prefers a single sip of the sweeter beverage, Gladwell argues, doesn’t mean he’d prefer to have an entire case of it at home.

Coca-Cola found this out the hard way when they introduced “New Coke”, a soft drink ***completely redesigned to match Pepsi’s success in the sip test***.

The results were catastrophic.

Are you intrigued enough to see how this works?

## The ‘What’ and ‘How’ of Cross-Validation

***Cross Validation*** (CV) is a method to make a statistical model more valid and ready to apply to situations beyond the sample we’re given.

It directly checks a model’s ability to predict NEW observations, instead of just fitting the observations we already have.

If you are planning to use cross-validation, you need to split your data up into a training set and a test set.

The ***training*** data set is the one used to build the model. It is typically the larger part of the data.

The ***test*** data set is set aside for later. After the model has been built, the predictions from the model are compared to the actual values.

Sometimes the test set is called the ***holdout*** set, it’s the set of observations that are ‘hidden’ from the model.

The model you make has to do two jobs now.

1. It has to fit the data in the training set well.
2. It has to predict the responses in the test set *without knowing what the responses are.*

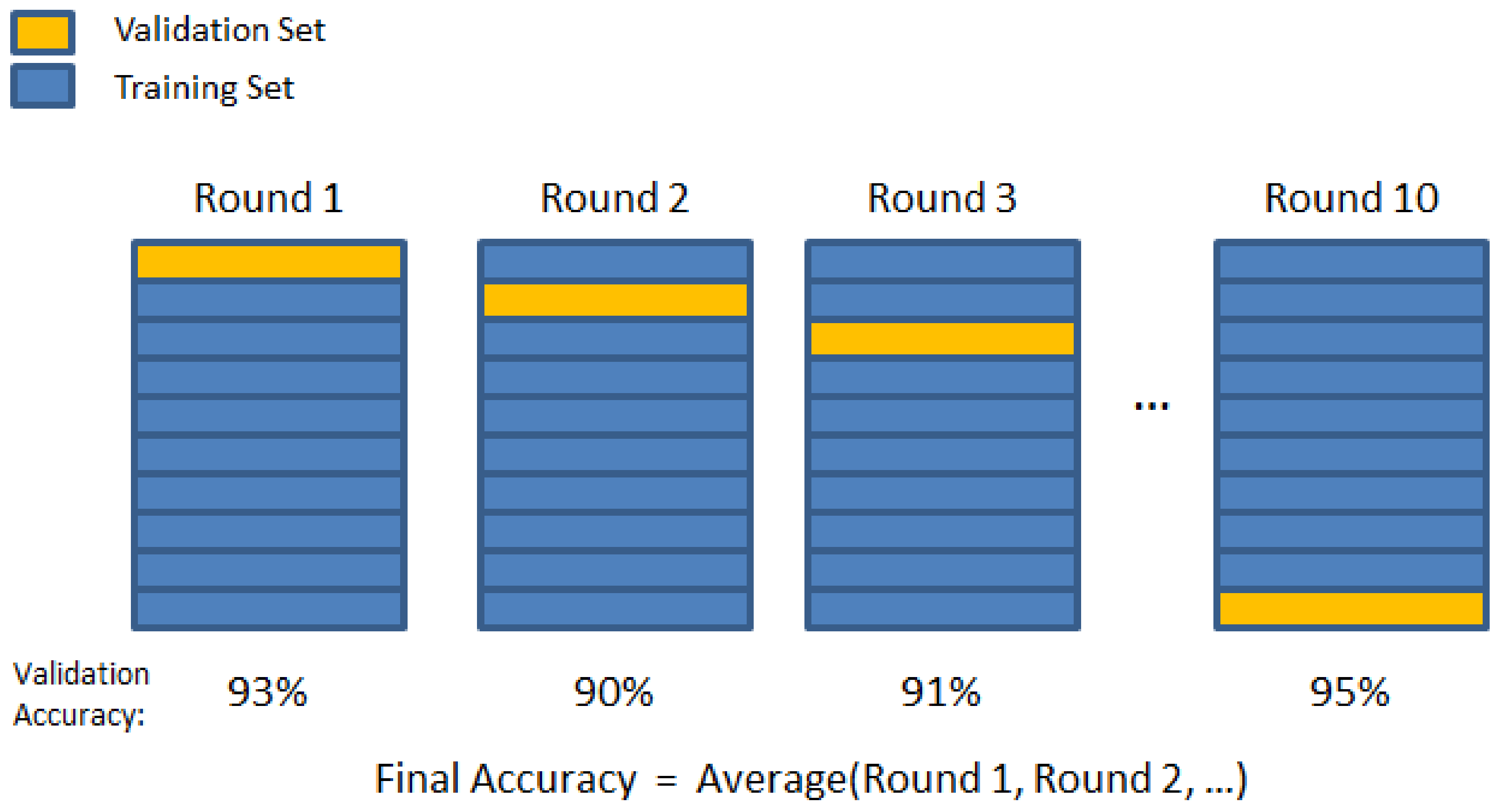
If the model overfits the training set, but the model's prediction of values in the test/holdout set are going to be poor.

The cv.glm() function in the boot package in R can run a common type of cross-validation analysis called "K-fold cross validation".

In a K-fold cross validation, 1/K of the data* becomes the test-set and the rest becomes the training set.

This repeats K times, so that every observation is part of the test set once and the training set every other time.

* Rounded to the nearest observation.



Then the ***PRESS statistic***, (based on the Predicted REsidual Sum of Squares) is taken from each round and averaged.

The average PRESS statistic is our measure of how big the errors are in prediction beyond the data.

We can compare it to the ***mean squared error (MSE)*** or mean square residual in an ANOVA table to see how much bigger the errors when we are predicting results rather than fitting them.

In the case of regression, the PRESS will always be bigger than the residual SS. This is because fitting observed data is easier than predicting new data.

The bigger K is…

…the bigger the training set is, and…

…the smaller the test set is, therefore…

**…the better the predictions will be.**

Better? Better how?

The PRESS is smaller as K gets larger.

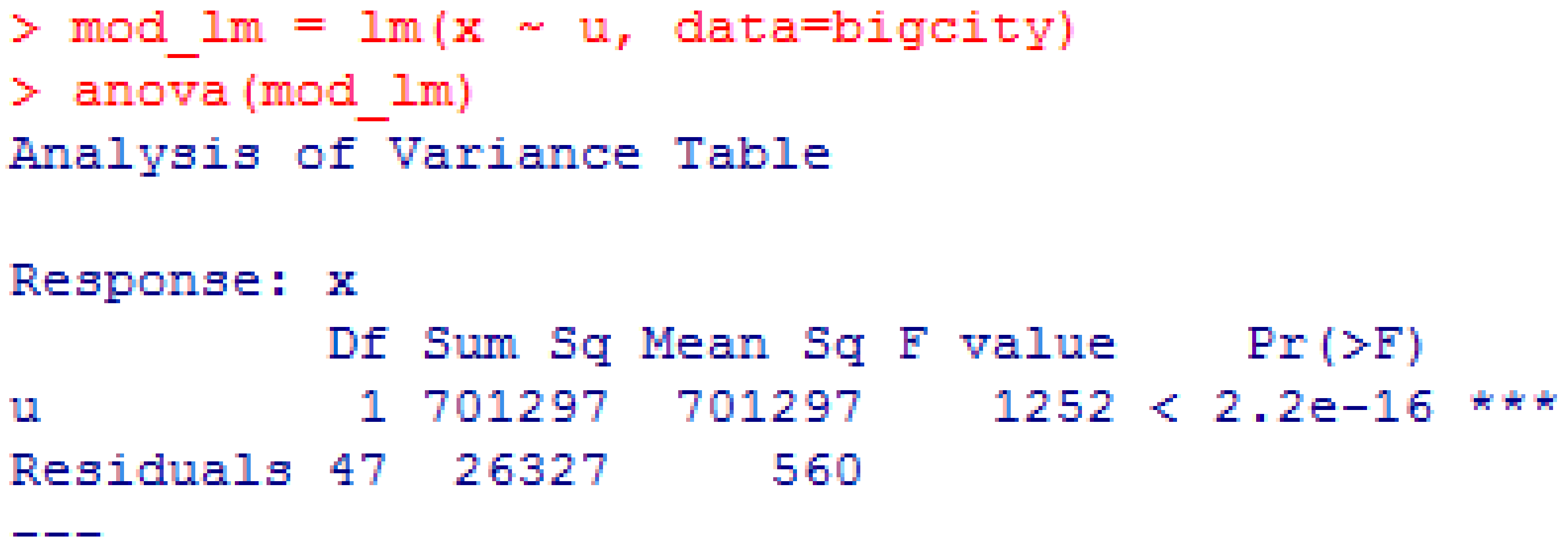
In other words:

Predictions improve when there is more data to work with (and less to predict).

The largest possible K is the number of observations. In that situation, 1/K is only a single observation each time.

This method has a special name:
***Leave-one-out*** cross validation.

Usually when people refer to the PRESS statistic, they are referring to the prediction sum of squares from a leave-one-out cross-validation.

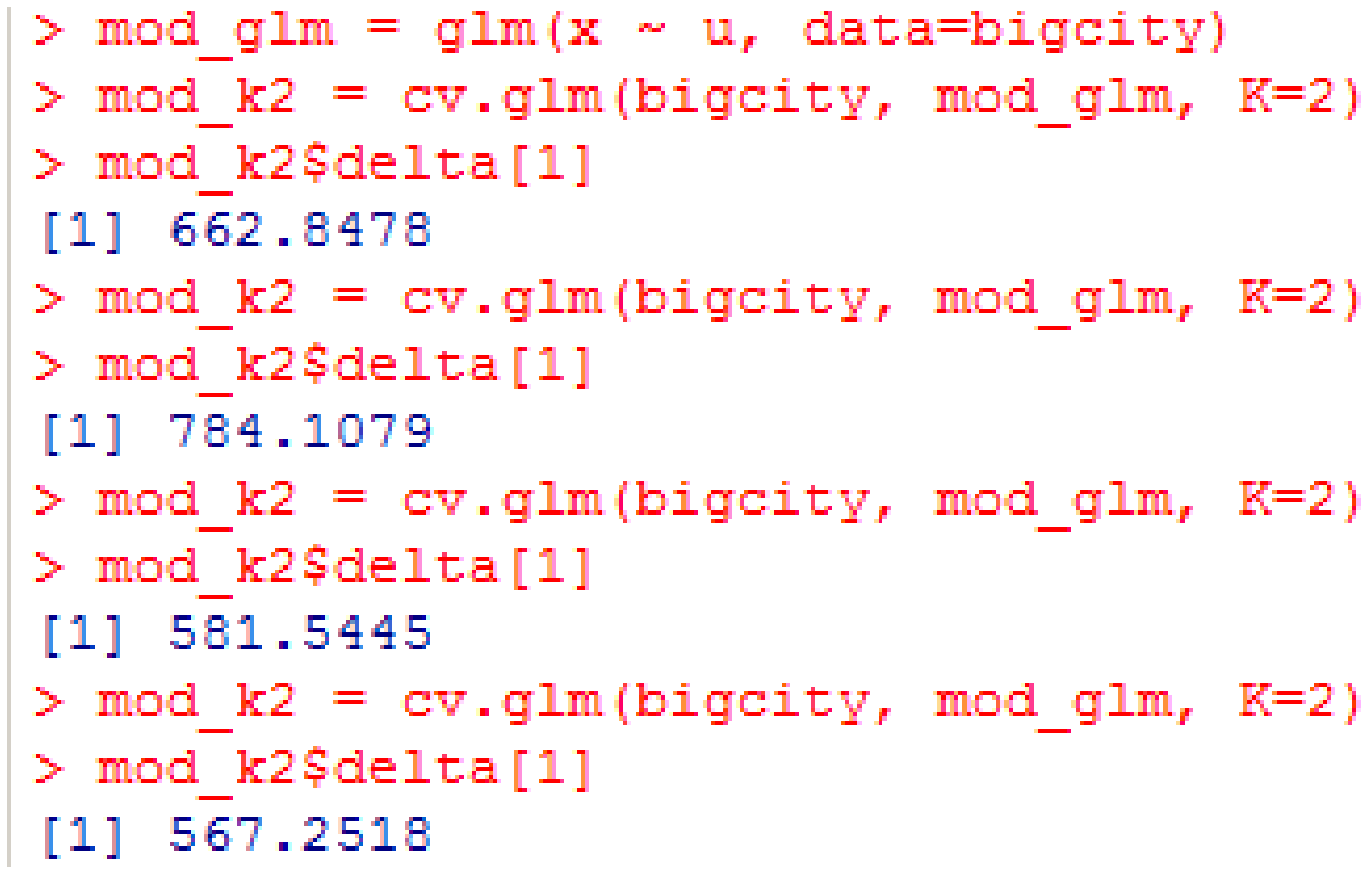


```
> mod_lm = lm(x ~ u, data=bigcity)
> anova(mod_lm)
Analysis of Variance Table

Response: x
          Df Sum Sq Mean Sq F value    Pr(>F)
u          1 701297  701297    1252 < 2.2e-16 ***
Residuals 47  26327     560
---
```

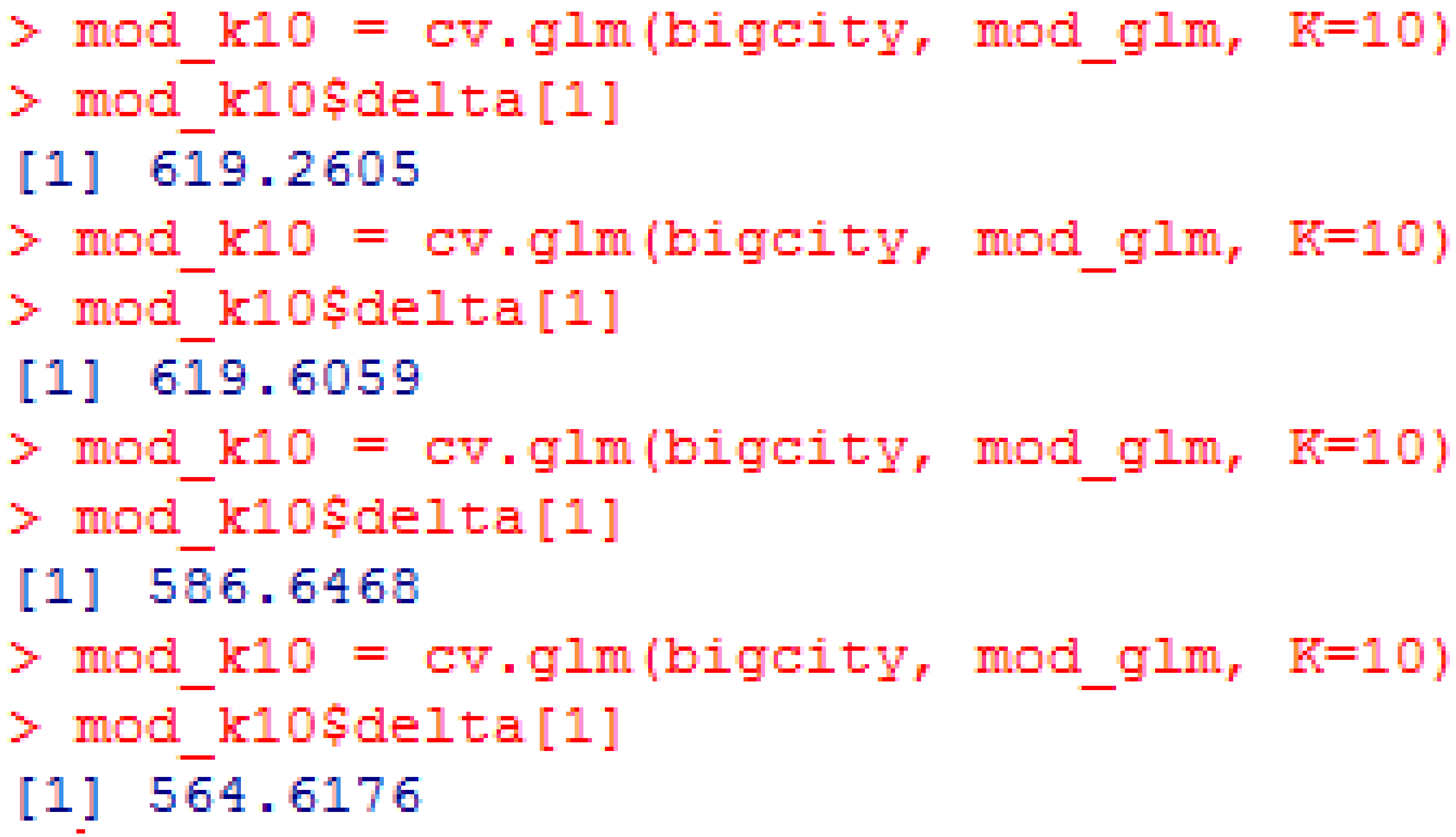
Consider the ‘bigcity’ dataset, a data set of 49 rows and 2 columns in the ‘boot’ package.

When we use linear regression to fit the responses ‘x’, the mean-squared error is 560.



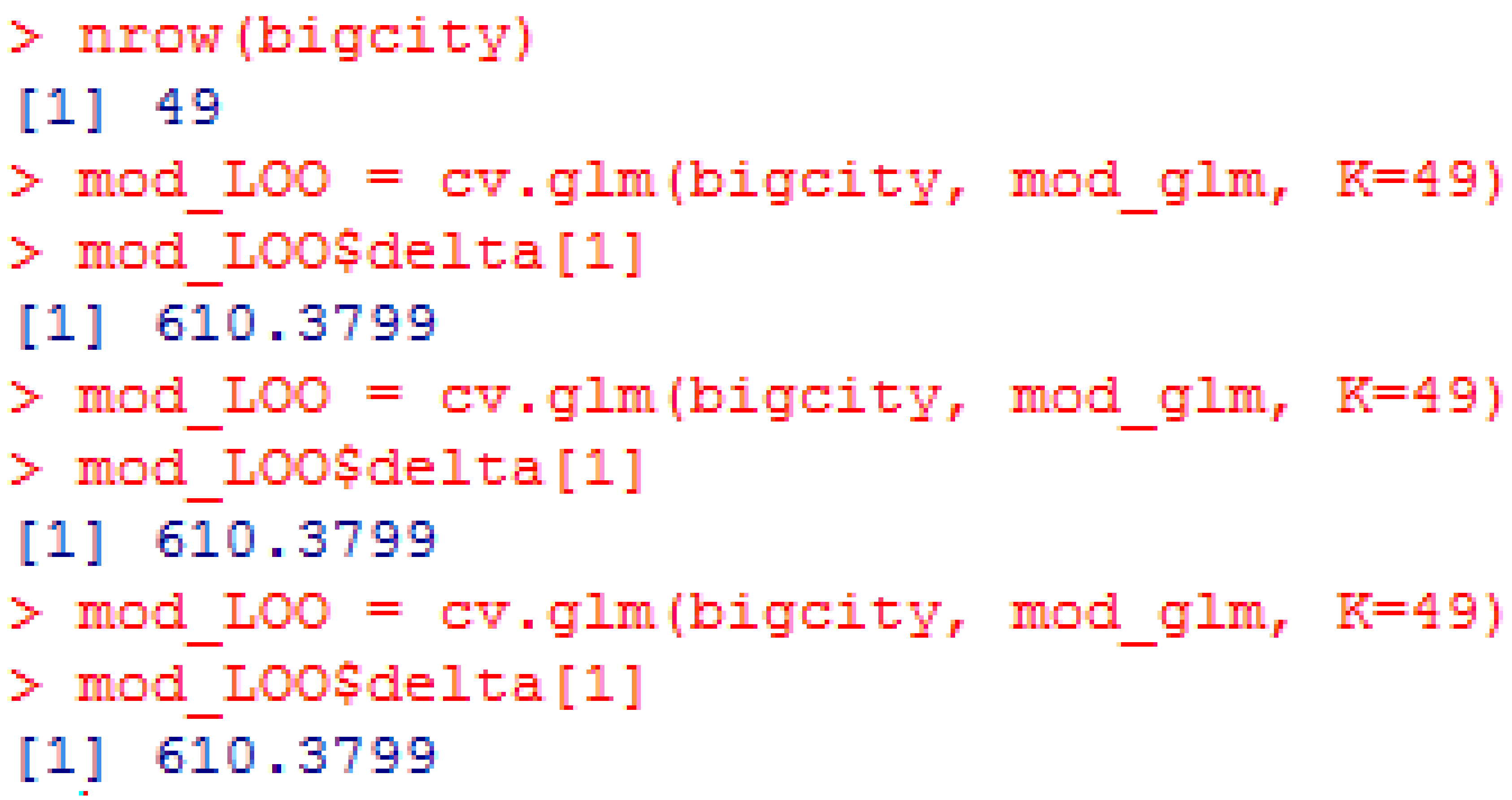
```
> mod_glm = glm(x ~ u, data=bigcity)
> mod_k2 = cv.glm(bigcity, mod_glm, K=2)
> mod_k2$delta[1]
[1] 662.8478
> mod_k2 = cv.glm(bigcity, mod_glm, K=2)
> mod_k2$delta[1]
[1] 784.1079
> mod_k2 = cv.glm(bigcity, mod_glm, K=2)
> mod_k2$delta[1]
[1] 581.5445
> mod_k2 = cv.glm(bigcity, mod_glm, K=2)
> mod_k2$delta[1]
[1] 567.2518
```

A 2-fold cross validation produces prediction errors of larger than 560. ( From many runs, about 30% more than MSE)



```
> mod_k10 = cv.glm(bigcity, mod_glm, K=10)
> mod_k10$delta[1]
[1] 619.2605
> mod_k10 = cv.glm(bigcity, mod_glm, K=10)
> mod_k10$delta[1]
[1] 619.6059
> mod_k10 = cv.glm(bigcity, mod_glm, K=10)
> mod_k10$delta[1]
[1] 586.6468
> mod_k10 = cv.glm(bigcity, mod_glm, K=10)
> mod_k10$delta[1]
[1] 564.6176
```

A 10-fold cross validation produces prediction errors that were closer to the MSE and more consistent. (About 10% more than MSE over many runs)



```
> nrow(bigcity)
[1] 49
> mod_LOO = cv.glm(bigcity, mod_glm, K=49)
> mod_LOO$delta[1]
[1] 610.3799
> mod_LOO = cv.glm(bigcity, mod_glm, K=49)
> mod_LOO$delta[1]
[1] 610.3799
> mod_LOO = cv.glm(bigcity, mod_glm, K=49)
> mod_LOO$delta[1]
[1] 610.3799
```

A 49-fold cross validation, also called a leave-one-out cross validation produces prediction errors of 610.38 every time. (About 9% more than MSE).

A few other notes:

Influential observations are those which make a big change to the model. When these are removed from the model (i.e. put in the test set), that influence is gone. As such, the training set model does a poor job of predicting influential observations.

In short, *influential points make predictions worse.*

The observations that get selected to be the first, second, … , kth test set are randomly selected, so the results will vary a little from run to run. (Except for leave-one-out)

“Does it Cross-Validate” is the $million question

(End of Hour 1)