Statistical Programming Languages – Day 2

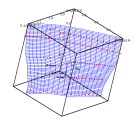
SVN-revision: 0

Uwe Ziegenhagen

Institut für Statistik and Ökonometrie
Humboldt-Universität zu Berlin
http://www.uweziegenhagen.de

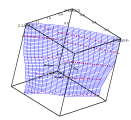
Agenda for Today

- Data frames
- Reading and Writing Data
- Exploratory Data Analysis



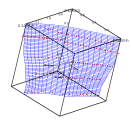
Data Frames

- matrices can only have one datatype
- data frames: several type allowed
- equal length of all elements required



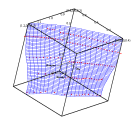
Data Frame Example

```
a <- c(10,20,15,43,76,41,25,46) # numeric
# Factor 'sex'
b <- factor(c("m","f","m","f","m","f","m","f"))
# siblings, numeric
c <- c(2,5,8,3,6,1,5,6)
myframe <- data.frame(a,b,c)
myframe
colnames(myframe) <- c("Age", "Sex", "Siblings")
```

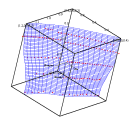
Factor variables

- Factor variables: categorical variables (numeric or string)
- advantages:
  - implemented correctly in statistical modeling
  - very useful in many different types of graphics
  - correct number of degrees of freedom



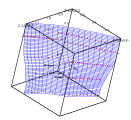
Adressing Components

```
myframe[,1]
myframe["Age"]
myframe$Age
myframe[3,3]<-2 # change value
myframe[,-2] # all vars except 2nd
```

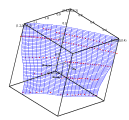
Overview of Objects II

```
1 # add object set to search path
2 attach(name)
3 # remove from search path
4 detach(name)
```

Subgrouping Data Frames

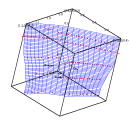
```
> subset(myframe,myframe$Age>30) # 4 entries
> mean(subset(myframe$Age,myframe$Sex=="m"))
[1] 31.5
> mean(subset(myframe$Age,myframe$Sex=="f"))
[1] 37.5
myframe[(myframe$Sex=="m") & (myframe$Age>30),]
# males over 30
myframe[(myframe$Sex=="m") | (myframe$Age>30),]
# male or over 30
```

Data Frames - Variables

```
> myframe <- cbind (myframe , "Income (USD)"=
  c(1700,2100,2300,2050,2800,1450,3400,2000))
> names(myframe)[names(myframe)=="Income (USD)"] <-
  "IncomeUSD"
```

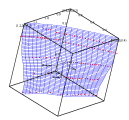
Task: Add variable `IncomeEUR`.



Search and Replace

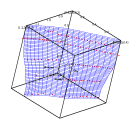
Use gsub to perform replacement of matches determined by regular expressions.

```
> names(myframe) <- gsub("In","Out",names(myframe))
> myframe
  Age Sex Siblings OutcomeUSD
1  10   m        2        1700
2  20   f        5        2100
```

Deleting and Sorting

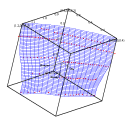
```
> myframe$Age <- NULL
> myframe

> myframe[order(myframe$Age),]
  Age Sex Siblings OutcomeUSD
1  10   m        2       1700
3  15   m        2       2300
```

Deleting and Sorting

1. Sortieren nach Sex
2. Sortieren nach Age

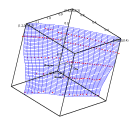
```
> myframe[order(myframe$Sex,partial=myframe$Age),]
   Age  Sex  Siblings  OutcomeUSD
2   20    f         5        2100
6   41    f         1        1450
8   46    f         6        2000
4   43    f         3        2050
1   10    m         2        1700
3   15    m         2        2300
5   76    m         6        2800
7   25    m         5        3400
```

Short excursion sed & awk

sed:

- stream **ed**itor, rowwise
- examples
  - `sed 's/abc/def/' input.txt >output.txt`
  - `sed 's|/|\|g' input.txt >output.txt`
  - **example using regular expression**
- extremely useful to process large amounts of data
- Tutorial: `http://www.grymoire.com/Unix/Sed.html`

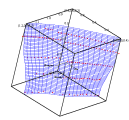


Short excursion sed & awk

awk:

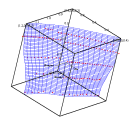
- **A**ho, **W**einberger, **K**ernighan
- in general used to work on columns
  - awk 'print 12' concatenates columns 1 and 2
  - awk 'print 1,3' prints columns 1 and 3
  - **another example using a sum**
- Tutorial: `http://www.vectorsite.net/tsawk.html`

In general: Avoid data processing inside *R*, try to do it outside.



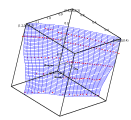
Data Management

- Sources of data:
  - Data in human readable format (CSV, TXT)
  - Data in binary format (Excel, SPSS, STATA)
  - Data from relational databases
- *R* has 100 built-in datasets: `objects(package:datasets)`
- many packages bring their own datasets



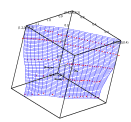
Loading data from library

```
1 library("datasets")  # loads dataset library
2                      # (automatically loaded)
3 data("pressure")     # loads dataset
4 data(pressure)       # alternative
5 pressure             # output pressure data
```

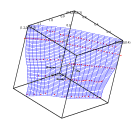
Data Management

```
1 objects(package:datasets)
2 help(Titanic)
3 data(Titanic)
4 objects()
```

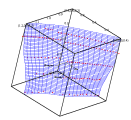
Reading & Writing Data

```
data <- read.table("filename", header=TRUE)
# guess type of variable: int, double, text
# header with columnnames is available
names(data) # variable names
str(data)   # show structure of dataframe
head(data) # show first rows
```

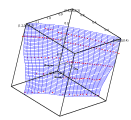
Reading & Writing Data

```
1 # check if datafile has header
2 # may generate string matrix only
3 # if 1st row less than 2nd assume head
4 data <- read.table("filename")
```

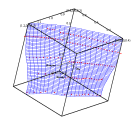
Reading & Writing Data

```
# using wrong separator
data <- read.table("file", sep="\t")
# assumes tabulator, may read whole
# line as one variable
```

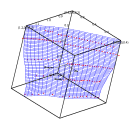
Reading & Writing Data

```
1 # reading NaNs
2 data <- read.table("file", na.strings=".")
3 # assumes NaN to be represented as '.'
```

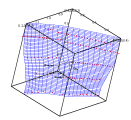
Reading & Writing Data

```
# reading CSV
# decimal sep '.', var. sep ','
data <- read.csv("file") #
# decimal sep ',', var. sep ';'
data2 <- read.csv2("file")
# direct import from Excel
data <- read.table(file="clipboard")
```

Reading & Writing Data

```
x <- read.csv("beispiel.csv", sep=";")
dim(x)
names(x)
x
# write to file
write.table(x,file="test.csv",sep=";",
 row.names= FALSE, quote=FALSE)
```

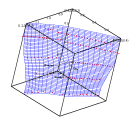
Univariate Statistics

d*distrib* density function

p*distrib* distribution function

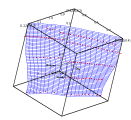
q*distrib* quantile function

r*distrib* random numbers



Univariate Statistics

```
dnorm(0) # density value of N(0,1)
pnorm(0) # cum. density up to 0
qnorm(0.5) # quantile for 0.5
rnorm(100) # vector with 100 random numbers
```

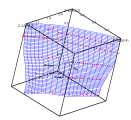
Univariate Statistics

d*distrib* density function

p*distrib* distribution function

q*distrib* quantile function

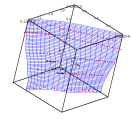
r*distrib* random numbers



Distributions in standard *R*

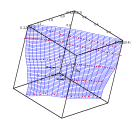
<key>binom	Binomial
<key>chisq	Chi-Squared
<key>exp	Exponential
<key>f	F
<key>hyper	Hypergeometric
<key>multinom	Multinomial
<key>logis	Logistic
<key>norm	Normal
<key>pois	Poisson
<key>t	Student t
<key>unif	Uniform

with <key>= `d`, `p`, `q` or `r`



Empirical Distributions in *R*

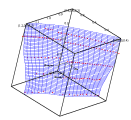
```
1 density() # KDE using Gaussian kernel
2 ecdf() # empirical cdf
```

Sampling in *R*

```
sample(n) # sample 1:n vector
sample(x) # shuffle the x vector
sample(x, replace=TRUE) bootstrap x vector
sample(x, n) # draw sample of size n from x
sample(x, n, replace = TRUE) # bootstrap sample from
   x
```

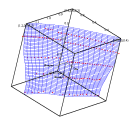
Seed is stored in `.Random.seed`, for simulations use `set.seed()`



Summary statistics

```
mean(x) # mean *
median(x) # median
var(x) # sample variance
sd(x) # sample std. deviation
cov(y) # cov of matrix y
quantile(x,p) # sample quantile *
min(x) # minimum of x *
max(x) # maximum of x *
range() # range of x *
skewness(x) # skewness
kurtosis(x) # kurtosis
```

* can remove **NaN**s using parameter `na.rm=T`



Linear Regression

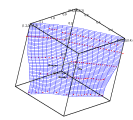
linear regression model

- tries to model relation between dependent variable Y and $1 \ldots n$ indep. variables $X_1, \ldots, X_n$
- influence of variables is linear, first regressor X_1 usually set to constant
- sample of size n is fitted to model:

$$y_i = \beta_1 + \beta_2 \cdot x_2 + \cdots + \beta_n \cdot x_n + \varepsilon_i$$

$$y_i = x_i^\top \beta + \varepsilon_i$$

$$y = X\beta + \varepsilon$$

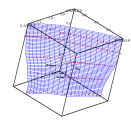


Linear Regression

Goals:

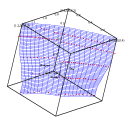
- estimate unknown βs using *least squares*
- decide if all variables are needed
- check if resulting model explains data well enough
- use model to forecast

$$\widehat{\beta} = (X^\top X)^{-1} X^\top y$$



Linear Regression

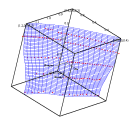
```
1 # standard model
2 lm(y~x + z)
3 # no intercept
4 lm(y~x - 1)
5 # using data frame
6 lm(amount ~ price, data = consumption)
7 # using data frame and attach()
8 lm(amount ~ price)
```

Exercise

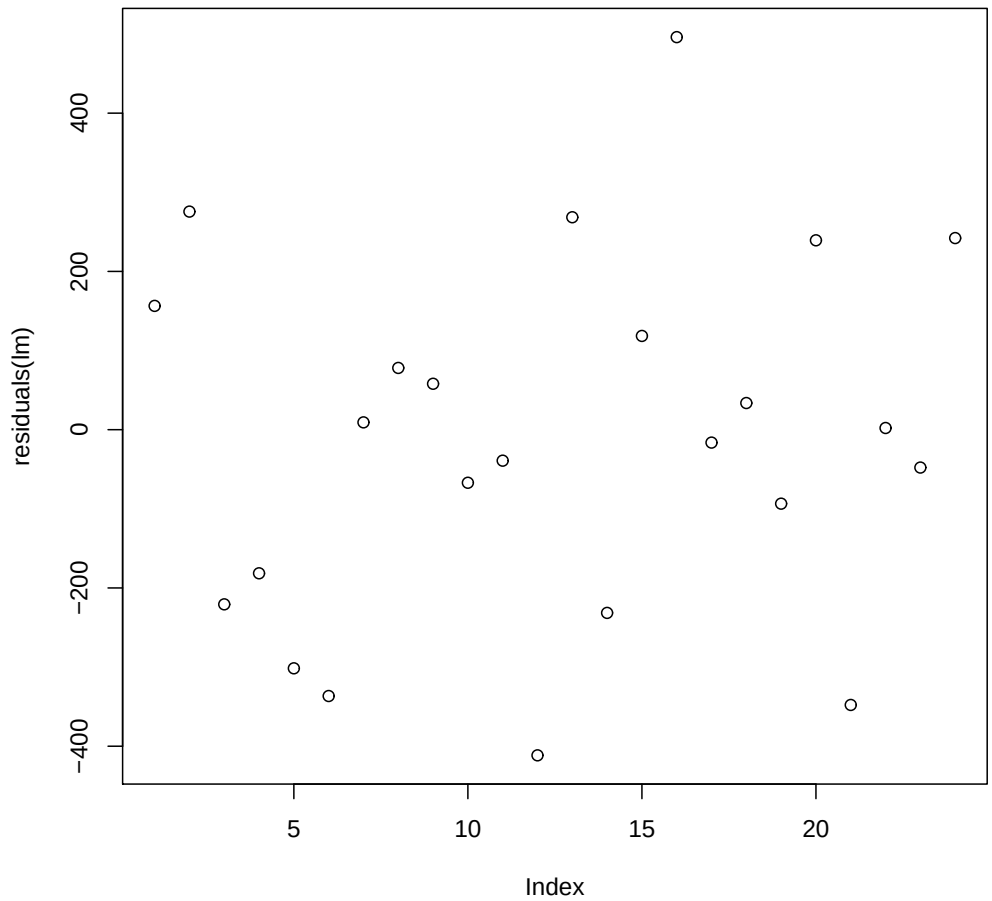
Download Hubble data from `http://lib.stat.cmu.edu/DASL/Datafiles/Hubble.html` and estimate the hubble constant H by the model

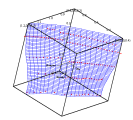
$$\text{recession-velocity} = H \cdot \text{distance}$$



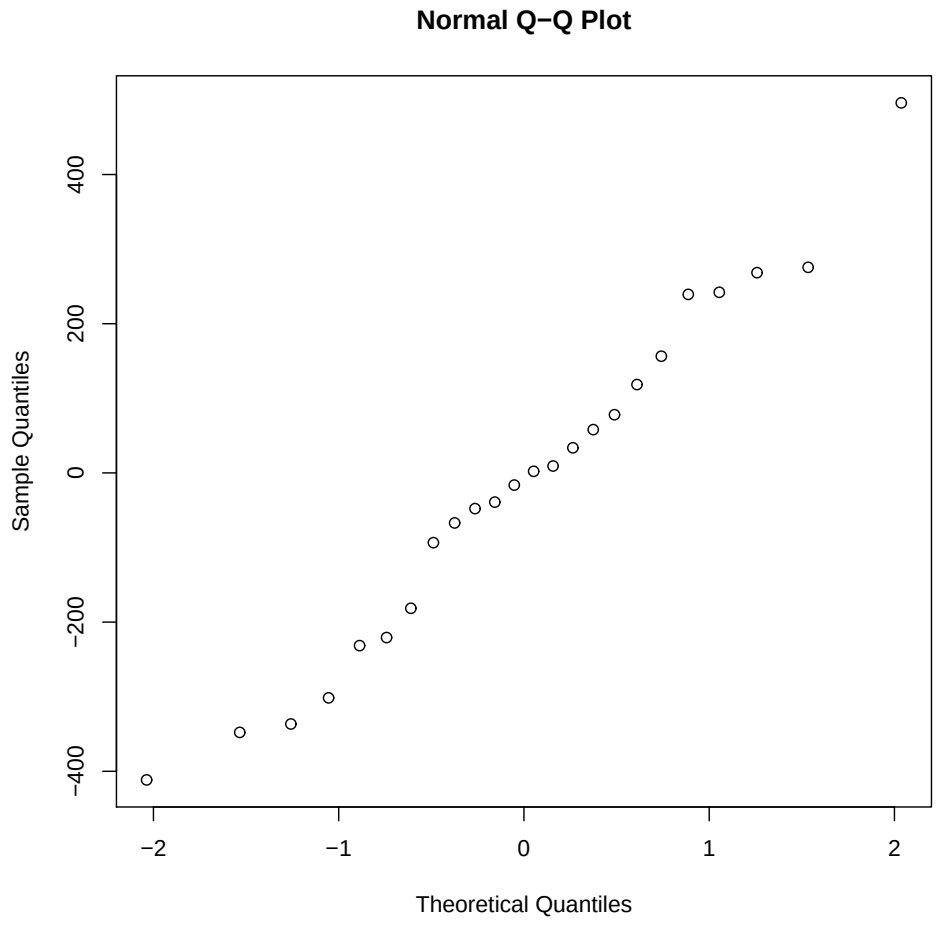
```
> summary(lm)

Call:
lm(formula = rec.vel ~ distance - 1)

Residuals:
     Min       1Q   Median       3Q      Max
-411.544 -191.302   -7.103  127.951  496.063

Coefficients:
         Estimate Std. Error t value Pr(>|t|)
distance   423.94      42.15   10.06 6.87e-10 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 229 on 23 degrees of freedom
Multiple R-squared: 0.8147,     Adjusted R-squared: 0.8067
F-statistic: 101.1 on 1 and 23 DF,  p-value: 6.869e-10
```

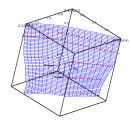
Residuals





Residuals


Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
400
200
0
−200
−400
−2
−1
0
1
2



Multiple Linear Regression

1. Download Cereal data from DASL
2. Read data as dataframe
3. Run linear regression rating $=$ sugars $+$ fat

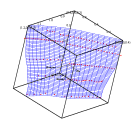
```
Call:
lm(formula = rating ~ sugars + fat, data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-14.6640  -5.6937   0.2078   4.7660  32.6163

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  61.0886     1.9527  31.284  < 2e-16 ***
sugars       -2.2128     0.2347  -9.428 2.59e-14 ***
fat          -3.0658     1.0365  -2.958  0.00416 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 8.755 on 74 degrees of freedom
Multiple R-squared: 0.6218,     Adjusted R-squared: 0.6116
F-statistic: 60.84 on 2 and 74 DF,  p-value: 2.371e-16
```

t-Test

- checks if certain coefficient β_j is different from 0.
- teststatistics

$$t = \frac{\hat{\beta}_j}{\mathrm{SD}(\hat{\beta}_j)}$$

- under $H_0 : t \sim t_{n-p}$ with p as number of independ. variables

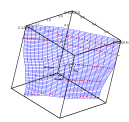


F-Test

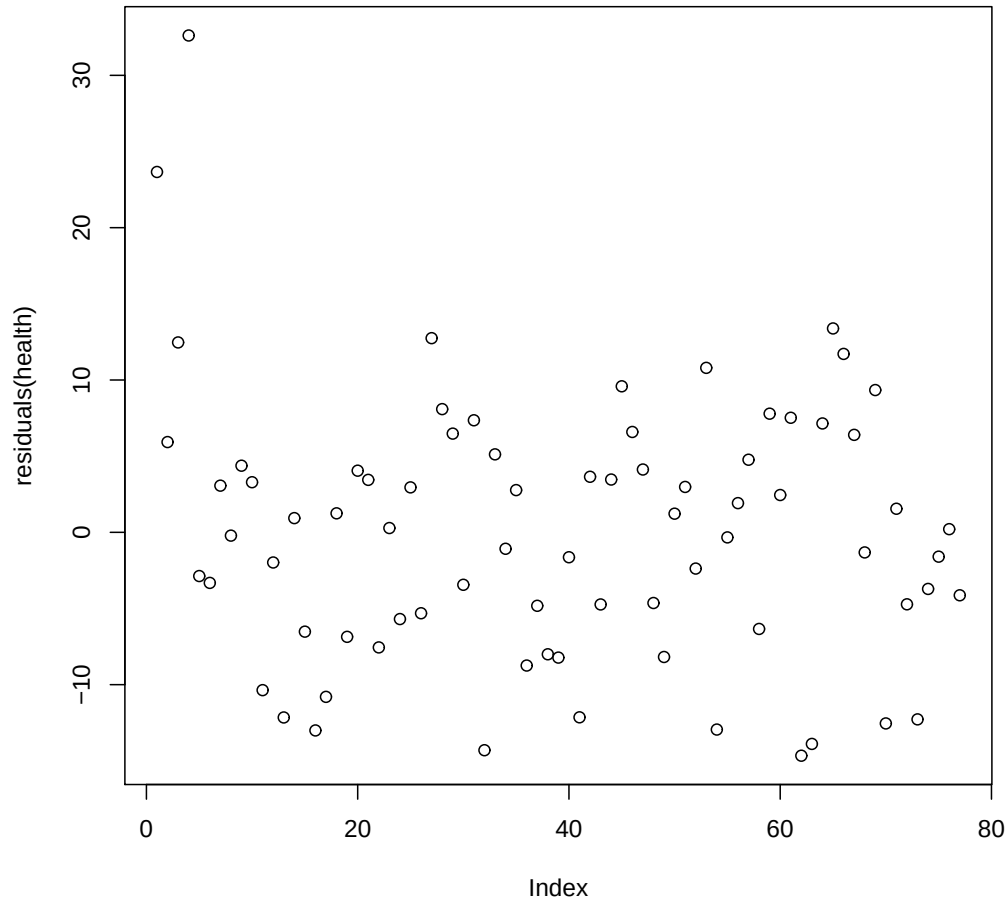
- idea: check if sum of squared residuals is reduced **significantly** if one regressor is added
- add one regressor $\Rightarrow$ model gets better, but significantly?
- Compute RSS1 for full model with k parameters, compute RSS2 for simplified model with $k - q$ parameters
- Compute teststatistics

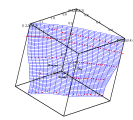
$$F = \frac{(RSS2 - RSS1)/q}{RSS1/(n-k)}$$

- under $H_0 : F \sim F_{(n-1,n-q-1)}$
- for $q = 1$ F-test equivalent to t-Test

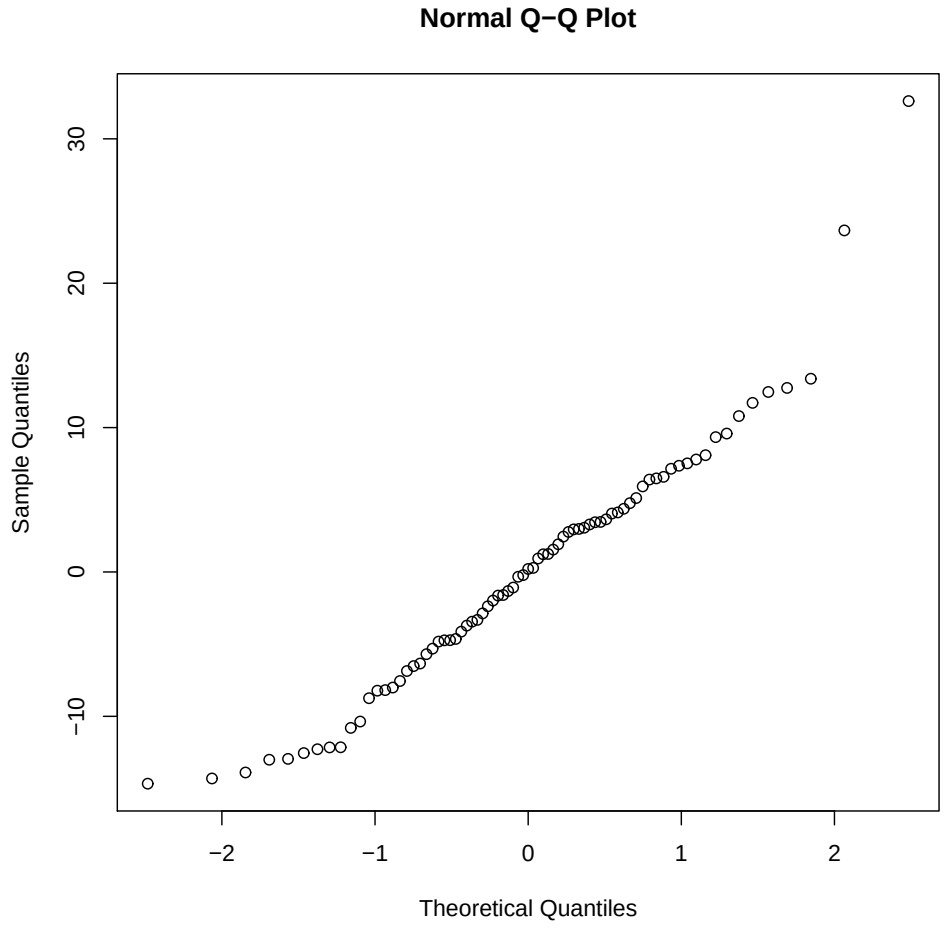


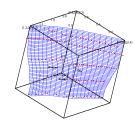
Residuals



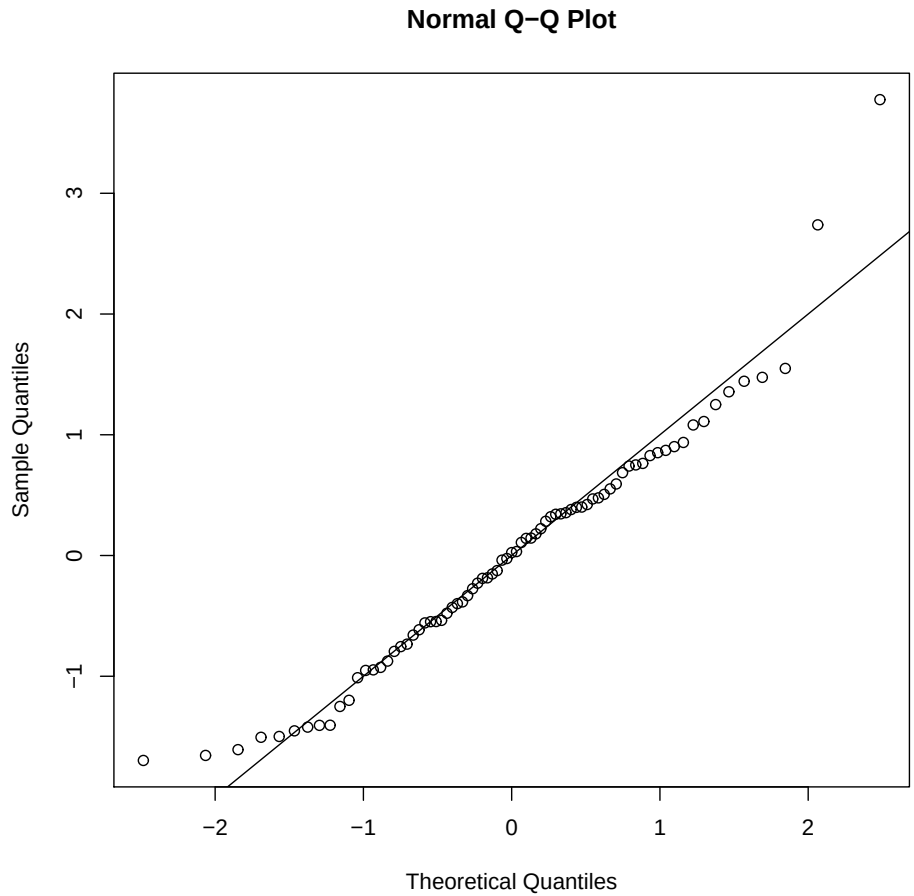


Residuals


Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
30
20
10
0
−10
−2
−1
0
1
2



Residuals


Normal Q−Q Plot
Sample Quantiles
Theoretical Quantiles
−2
−1
0
1
2
−1
0
1
2
3