**Final: Homework 1-5**

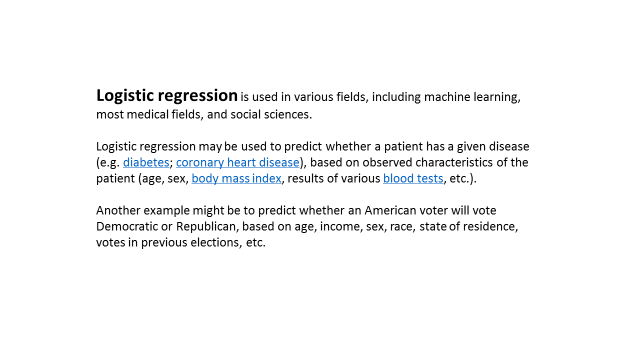
(By: Qian Liu 15620161152269)

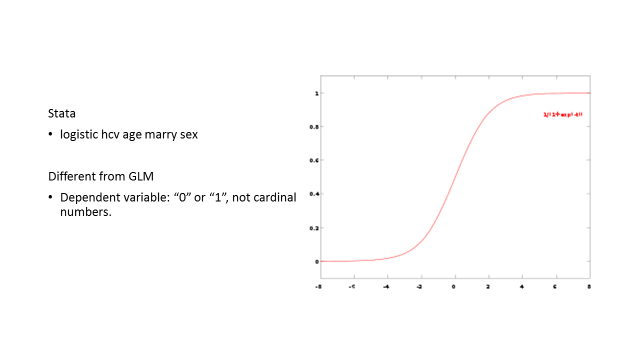
**Homework 1**

**Homework 1.1**

|  |  |  |  |
| --- | --- | --- | --- |
| **Year** | **Byte** | **Year** | **Byte** |
| 1970 | 262144 | 1990 | 2097152 |
| 1971 | 262144 | 1991 | 16777216 |
| 1972 | 262144 | 1992 | 16777216 |
| 1973 | 262144 | 1993 | 16777216 |
| 1974 | 262144 | 1994 | 16777216 |
| 1975 | 262144 | 1995 | 16777216 |
| 1976 | 262144 | 1996 | 268435456 |
| 1977 | 262144 | 1997 | 268435456 |
| 1978 | 262144 | 1998 | 1073741824 |
| 1979 | 262144 | 1999 | 1073741824 |
| 1980 | 262144 | 2000 | 1073741824 |
| 1981 | 262144 | 2004 | 4294967296 |
| 1982 | 262144 | 2009 | 8589934592 |
| 1988 | 2097152 | 2014 | 17179869184 |
| 1989 | 2097152 |  |  |

**Homework 1.2** （PPT Presentation）





**Homework 2**

**Homework 2.1**

>year=c(1970,1971,1972,1973,1974,1975,1976,1977,1978,1979,1980,1981,1982,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,1999,2000,2004,2009,2014)

>byte=c(262144,262144,262144,262144,262144,262144,262144,262144,262144,262144,262144,262144,262144,2097152,2097152,2097152,16777216,16777216,16777216,16777216,16777216,268435456,268435456,1073741824,1073741824,1073741824,4294967296,8589934592,17179869184)

> plot(year,byte)



**Homework 2.2**

>year=c(1970,1971,1972,1973,1974,1975,1976,1977,1978,1979,1980,1981,1982,1988,1989,1990,1991,1992,1993,1994,1995,1996,1997,1998,1999,2000,2004,2009,2014)

>byte=c(262144,262144,262144,262144,262144,262144,262144,262144,262144,262144,262144,262144,262144,2097152,2097152,2097152,16777216,16777216,16777216,16777216,16777216,268435456,268435456,1073741824,1073741824,1073741824,4294967296,8589934592,17179869184)

> plot(year,byte)

> splines.reg.1= smooth.spline(x = year, y = byte, spar = 0.2)

> splines.reg.2 = smooth.spline(x = year, y = byte, spar = 1)

> splines.reg.3= smooth.spline(x = year, y = byte, spar = 2)

> plot(year,byte)

> lines(splines.reg.1, col = "red", lwd = 2)

> lines(splines.reg.2, col = "green", lwd = 2)

> lines(splines.reg.3, col = "blue", lwd = 2)



**Homework 2.3**

2.3.1 Suppose you observe that in n=1000 mails (in 1 week) you have about 2 scams. Use the LvB /Possion pdf to calculate that you have 6 scam emails in 2 weeks.

R Code:

> lambda=2

> x=3

> dpois(x,lambda)

2.3.2 In Scammyland you have 5 scams on average, what is the probability to have no scam mail.

R Code:

> lambda=5

> x=0

> dpois(x,lambda)

**Homework 3**

**Homework 3.1**

> install.packages("digest", repos='http://cran.us.r-project.org')

> library("digest")

> sentence1=digest("I learn a lot from this class when I am proper listening to the professor","sha256")

> sentence2=digest("I do not learn a lot from this class when I am absent and playing on my Iphone","sha256")

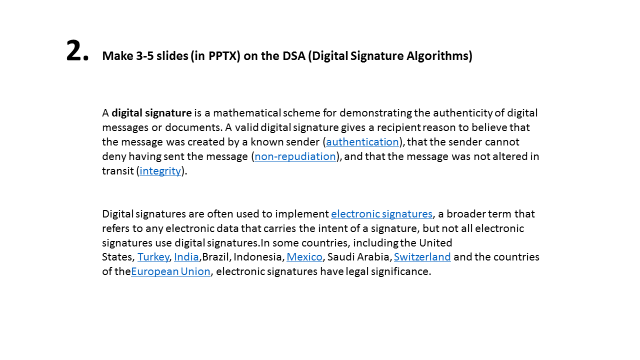
> sentence1

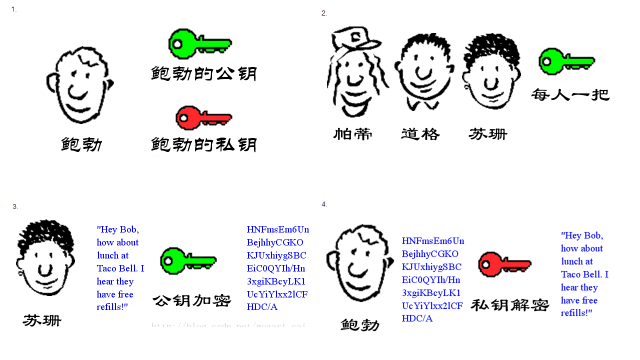
[1] "c16700de5a5c1961e279135f2be7dcf9c187cb6b21ac8032308c715e1ce9964c"

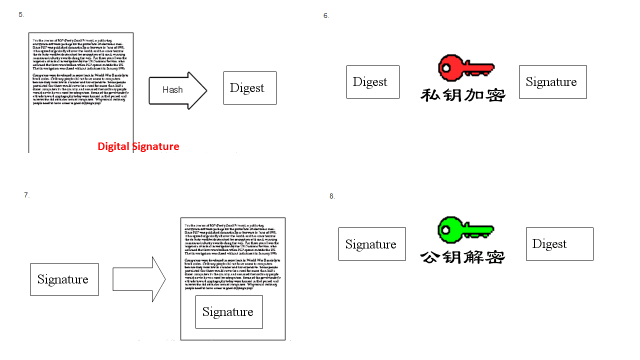
> sentence2

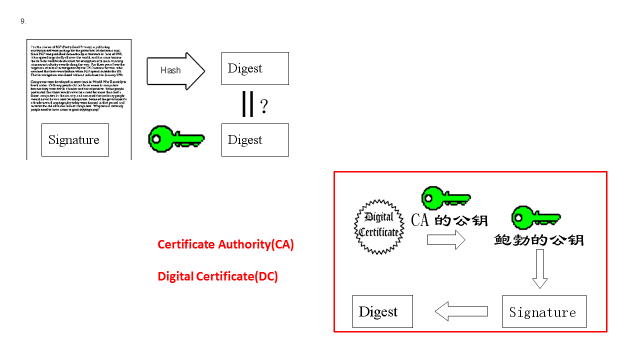
[1] "2533d529768409d1c09d50451d9125fdbaa6e5fd4efdeb45c04e3c68bcb3a63e"

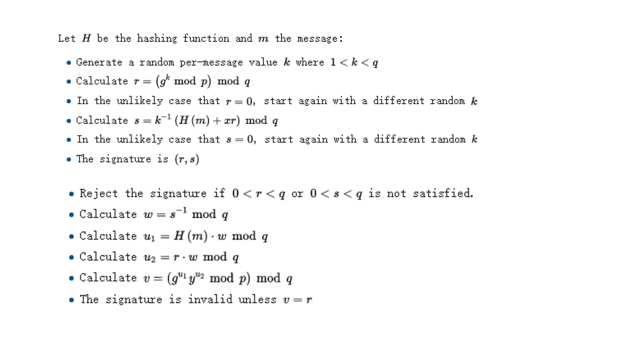
**Homework 3.2** （PPT Presentation）











**Homework 3.3**

> install.packages("rjson",repos=‘http://cran.us.r-project.org’)

> library("rjson")

> name<-c("Bob","Jon","Linda","Kim","Susan","Amy")

> gender<-c("M","M","F","M","F","F" )

> individualfeature<-data.frame(name,gender)

> data<-as.matrix( individualfeature)

> cat(toJSON(data))

["Bob","Jon","Linda","Kim","Susan","Amy","M","M","F","M","F","F"]

**Homework 3.4**

> install.packages("rjson",repos=‘http://cran.us.r-project.org’)

> library("rjson")

> json\_file="http://crix.hu-berlin.de/data/crix.json"

> json\_data=fromJSON(file=json\_file)

> crix\_data\_frame=as.data.frame(json\_data)

> x=crix\_data\_frame

> n=dim(x)

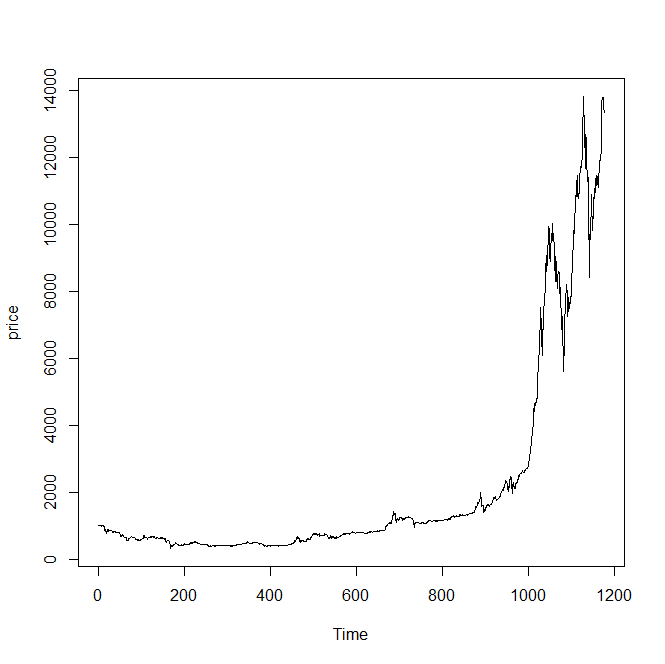
> a=seq(1,n[2],2)

> b=seq(2,n[2],2)

> date=t(x[1,a])

> price=t(x[1,b])

> ts.plot(price)



**Homework 4**

**Homework 4.1**

**#Figure 3**

>libraries = c("ccgarch", "rmgarch", "xts", "zoo")

>lapply(libraries, function(x) if (!(x %in% installed.packages())) { install.packages(x) })

>lapply(libraries, library, quietly = TRUE, character.only = TRUE)

>library(“rjson”)

>crix\_file="http://crix.hu-berlin.de/data/crix.json"

>ecrix\_file="http://crix.hu-berlin.de/data/ecrix.json"

>efcrix\_file="http://crix.hu-berlin.de/data/efcrix.json"

>ecrix1 = zoo(ecrix, order.by = index(crix1))

>efcrix1 = zoo(efcrix, order.by = index(crix1))

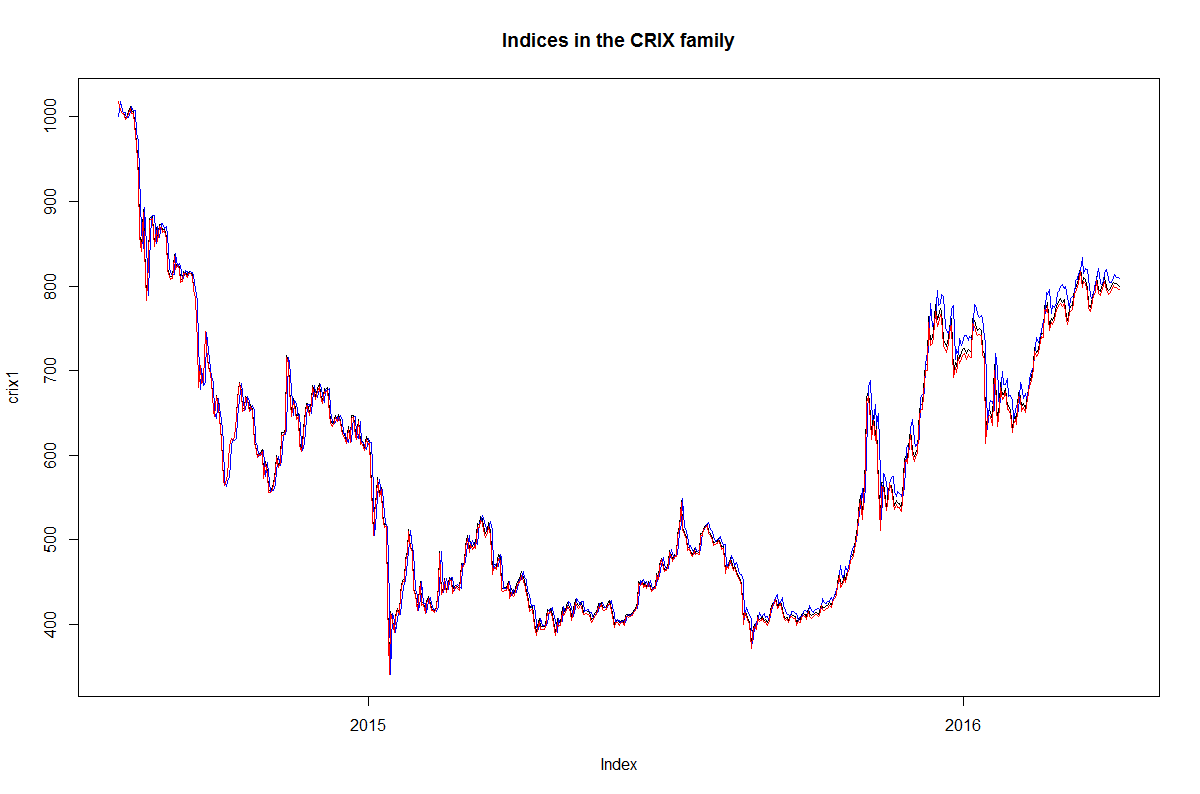
>my.panel <- function(x, ...) {

+lines(x, ...)

+lines(ecrix1, col = "blue")

+lines(efcrix1, col = "red") }

>plot.zoo(crix1, plot.type = "multiple", type = "l", lwd = 1.5, panel = my.panel, main = "Indices in the CRIX family")



**#Figure 4**

>libraries = c("zoo", "tseries", "xts")

>lapply(libraries, function(x) if (!(x %in% installed.packages())) {install.packages(x)})

>lapply(libraries, library, quietly = TRUE, character.only = TRUE)

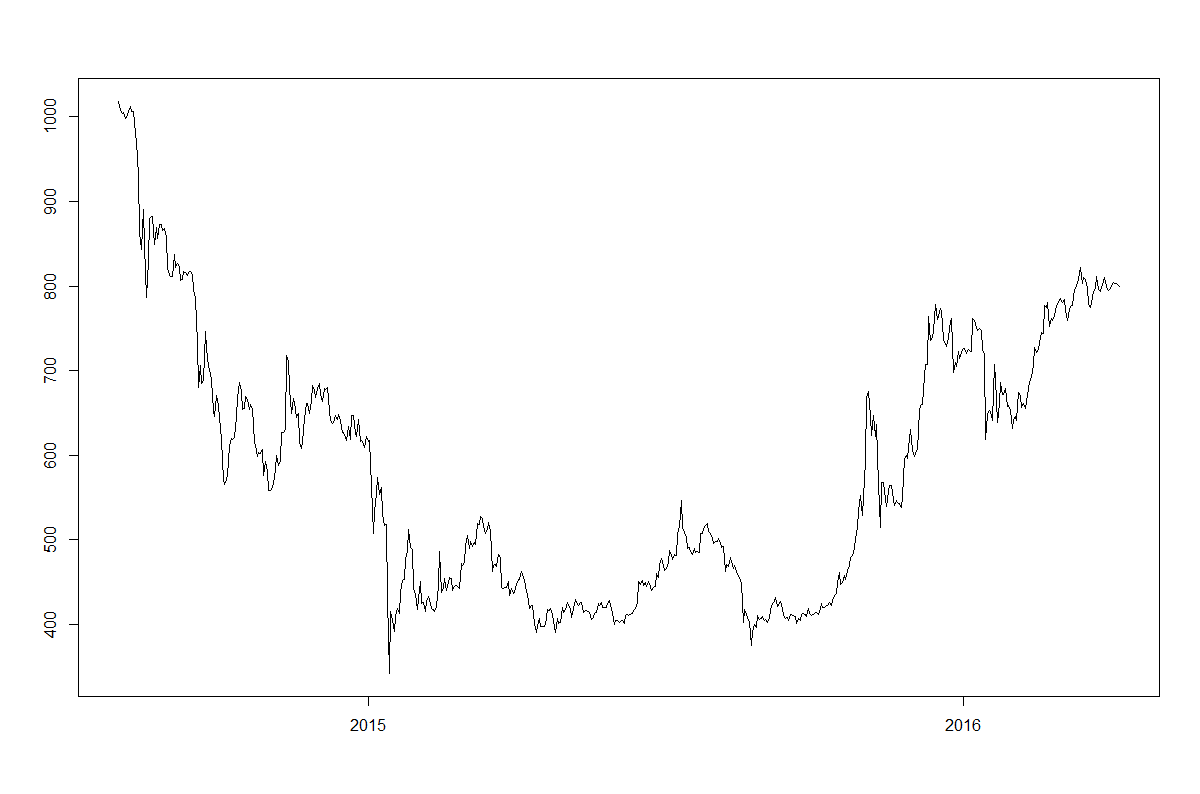
> library("rjson")

> crix\_file=“http://crix.hu-berlin.de/data/crix.json”

>crix\_data=fromJSON(file=crix\_file)

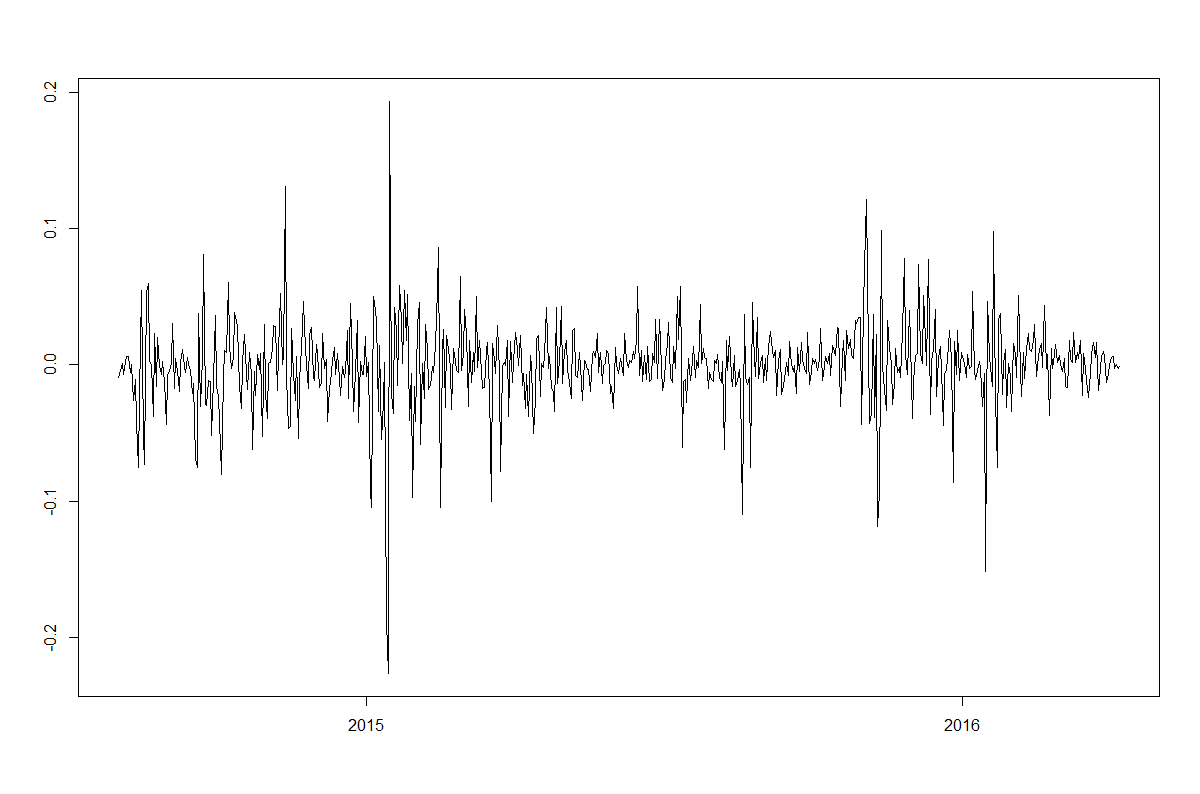
>crix\_data\_frame=as.data.frame(json\_data)

>plot(crix1, ylab = NA, xlab = NA)



>ret=diff(log(crix1))

>plot(ret, ylab = NA, xlab = NA)



**#Figure 5**

>libraries = c("zoo", "tseries", "xts")

>lapply(libraries, function(x) if (!(x %in% installed.packages())) { install.packages(x) })

>lapply(libraries, library, quietly = TRUE, character.only = TRUE)

>crix\_file="http://crix.hu-berlin.de/data/crix.json"

>ret=diff(log(crix1))

>mean(ret)

>var(ret)

>sd(ret)

>hist(ret, col = "grey", breaks = 20, freq = FALSE, ylim = c(0, 25), xlab = NA)

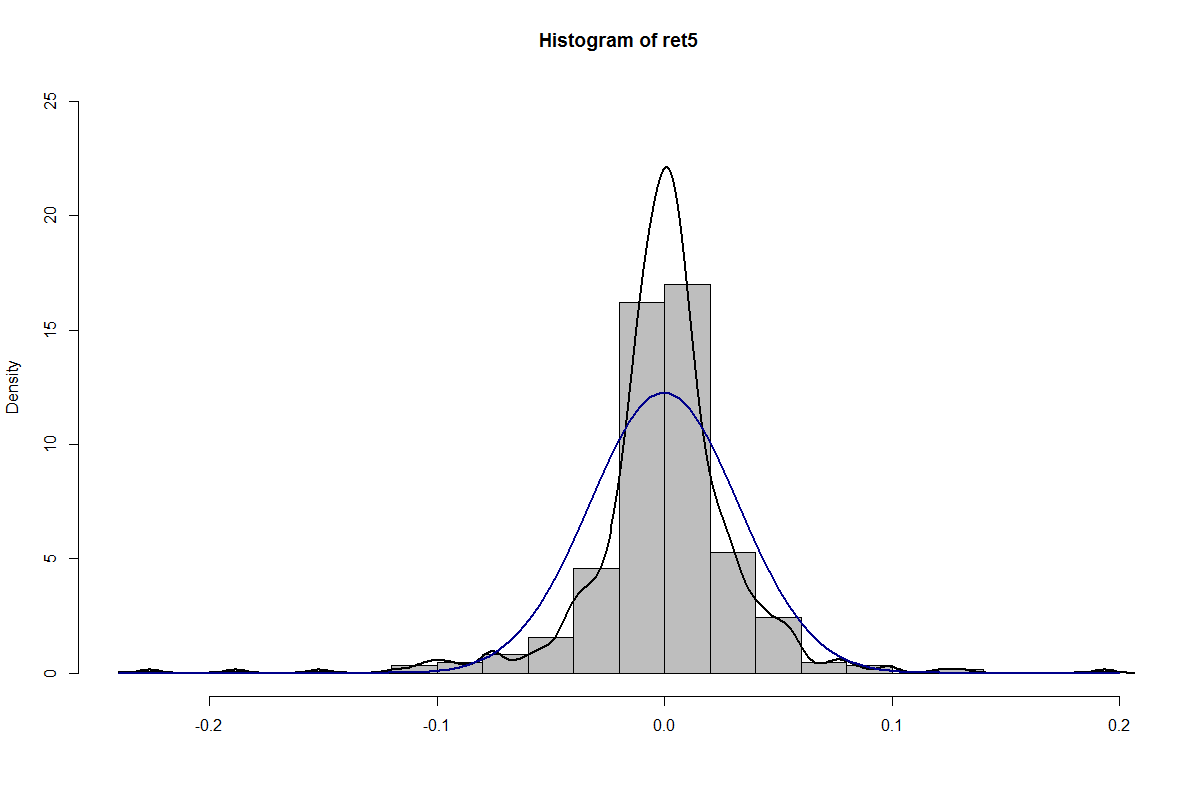
>lines(density(ret), lwd = 2)

>mu = mean(ret)

>sigma = sd(ret)

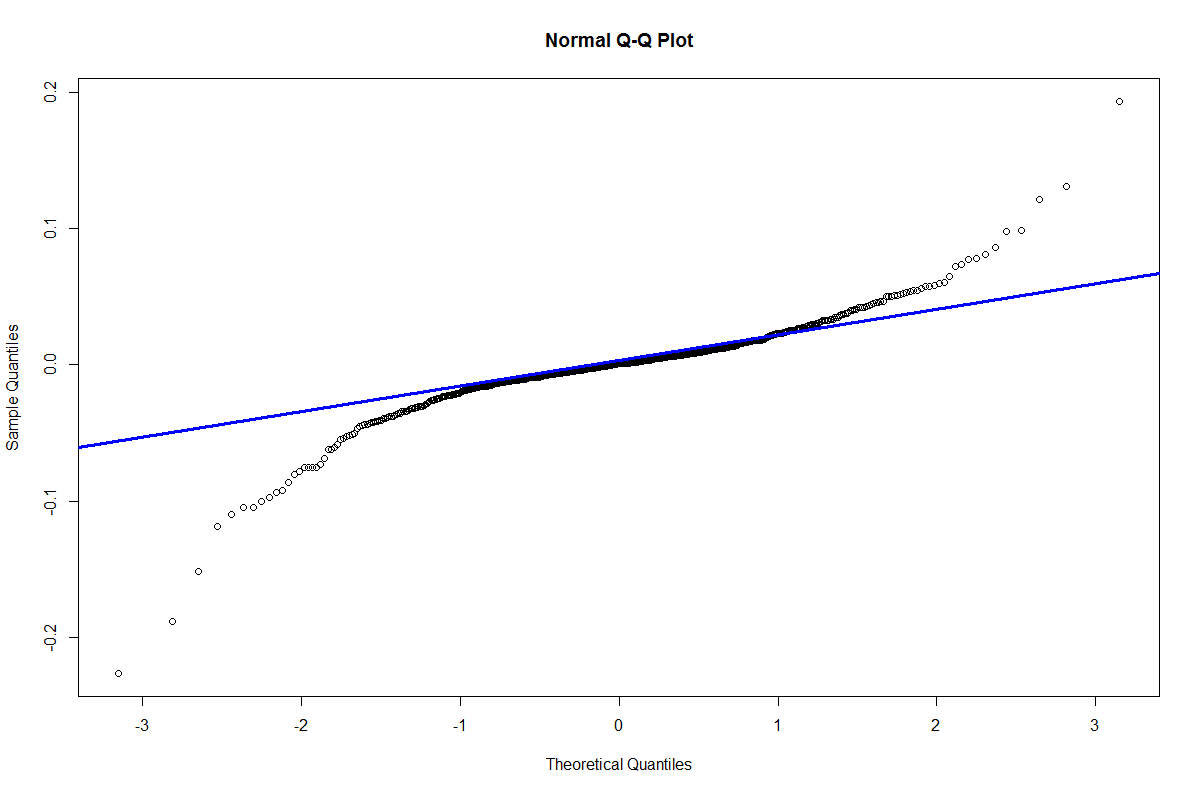
>x = seq(-4, 4, length = 100)

>curve(dnorm(x, mean = mean(ret), sd = sd(ret)), add = TRUE, col = "darkblue", lwd = 2)



>qqnorm(ret)

>qqline(ret, col = "blue", lwd = 3)



**#Figure 6**

>libraries = c("zoo", "tseries")

>lapply(libraries, function(x) if (!(x %in% installed.packages())) {install.packages(x)})

>lapply(libraries, library, quietly = TRUE, character.only = TRUE)

> library("rjson")

> crix\_file=“http://crix.hu-berlin.de/data/crix.json”

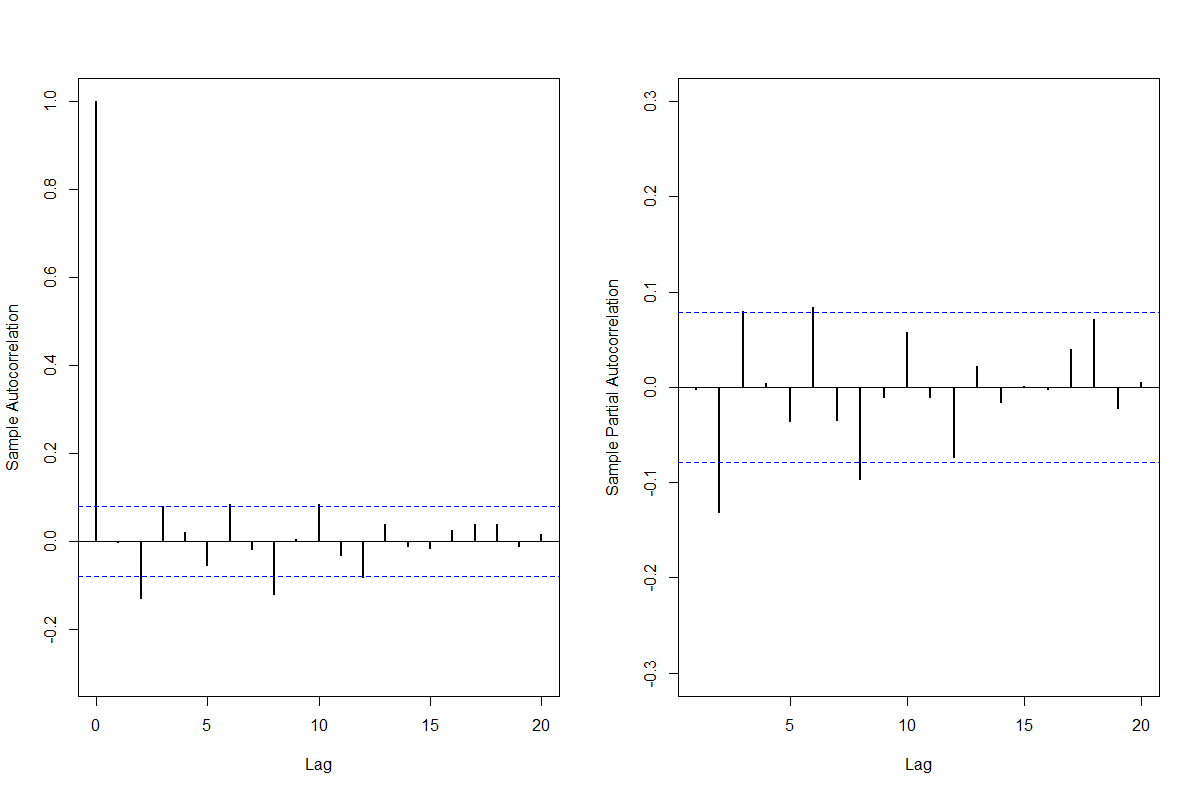
>crix\_data=fromJSON(file=crix\_file)

>crix\_data\_frame=as.data.frame(crix\_data)

>ret = diff(log(crix))

>autocorr = acf(ret, lag.max = 20, ylab = "Sample Autocorrelation", main = NA, lwd = 2, ylim = c(-0.3, 1))

>autopcorr = pacf(ret, lag.max = 20, ylab = "Sample Partial Autocorrelation", main = NA, ylim = c(-0.3, 0.3), lwd = 2)



**Homework 4.2**

**#Figure 7**

>libraries = c("zoo", "tseries", "xts")

>lapply(libraries, function(x) if (!(x %in% installed.packages())) {install.packages(x)})

>lapply(libraries, library, quietly = TRUE, character.only = TRUE)

>json\_file="http://crix.hu-berlin.de/data/crix.json"

>json\_data=fromJSON(file=json\_file)

>crix\_data\_frame=as.data.frame(json\_data)

>x=crix\_data\_frame

>n=dim(x)

>a=seq(1,n[2],2)

>b=seq(2,n[2],2)

>date=t(x[1,a])

>price=t(x[1,b])

>ret=diff(log(price))

# arima202 predict

>fit202 = arima(ret, order = c(2, 0, 2))

> crpre = predict(fit202, n.ahead = 30)

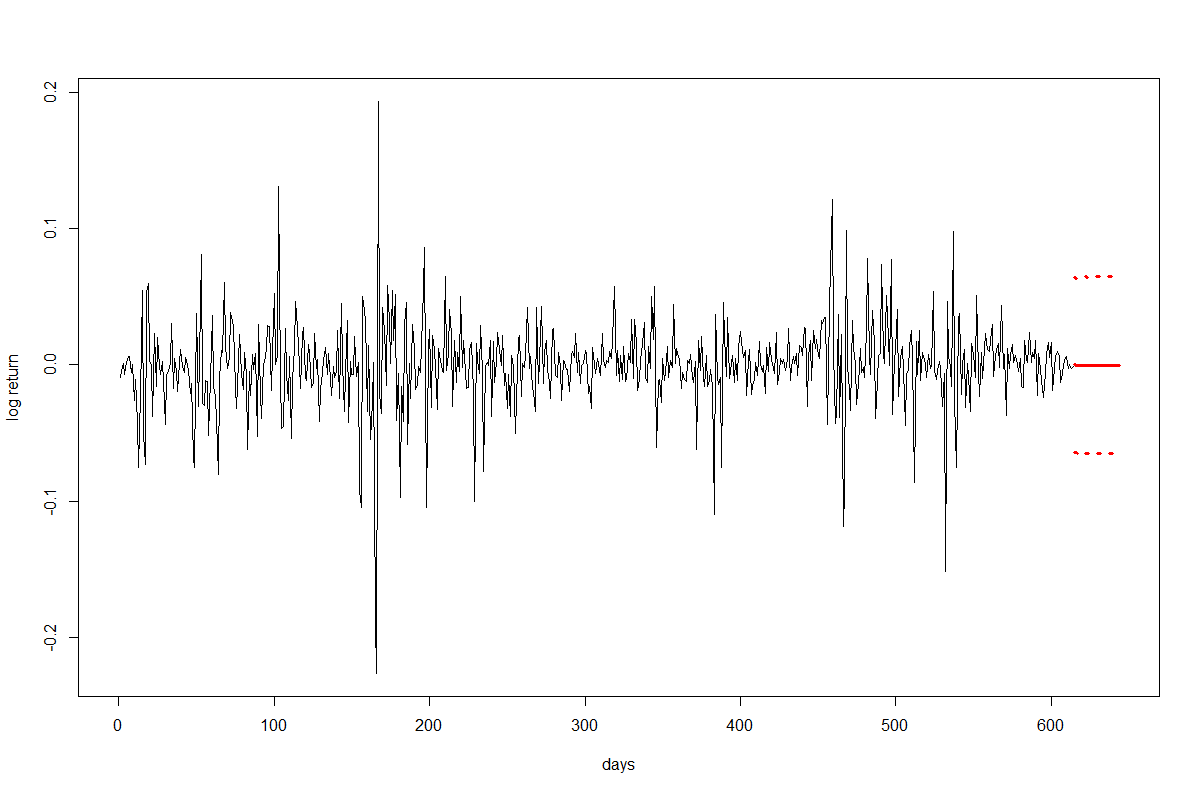
> dates = seq(as.Date("02/08/2014", format = "%d/%m/%Y"), by = "days", length = length(ret))

> plot(ret, type = "l", xlim = c(0, 644), ylab = "log return", xlab = "days", lwd = 1.5)

> lines(crpre$pred, col = "red", lwd = 3)

> lines(crpre$pred + 2 \* crpre$se, col = "red", lty = 3, lwd = 3)

> lines(crpre$pred - 2 \* crpre$se, col = "red", lty = 3, lwd = 3)



**Homework 4.3**

>libraries = c("FinTS", "tseries", "forecast", "fGarch")

>lapply(libraries, function(x) if (!(x %in% installed.packages())) {install.packages(x)})

>lapply(libraries, library, quietly = TRUE, character.only = TRUE)

>crix\_file="http://crix.hu-berlin.de/data/crix.json"

>ret = diff(log(crix1))

>fit202 = arima(ret, order = c(2, 0, 2))

>par(mfrow = c(1, 1))

>res = fit202$residuals

>res2 = fit202$residuals^2

>fg11 = garchFit(data = res, data ~ garch(1, 1))

>summary(fg11)

>fg12 = garchFit(data = res, data ~ garch(1, 2))

>summary(fg12)

>fg21 = garchFit(data = res, data ~ garch(2, 1))

>summary(fg21)

>fg22 = garchFit(data = res, data ~ garch(2, 2))

>summary(fg22)

>reszo = zoo(fg11@residuals, order.by = index(crix1))

>plot(reszo, ylab = NA, lwd = 2)

>par(mfrow = c(1, 2))

>fg11res2 = fg11@residuals

>acfres2 = acf(fg11res2, lag.max = 20, ylab = "Sample Autocorrelation", main = NA, lwd = 2)

>pacfres2 = pacf(fg11res2, lag.max = 20, ylab = "Sample Partial Autocorrelation", main = NA, lwd = 2, ylim = c(-0.5, 0.5))

>fg12res2 = fg12@residuals

>acfres2 = acf(fg12res2, lag.max = 20, ylab = "Sample Autocorrelation", main = NA, lwd = 2)

>pacfres2 = pacf(fg12res2, lag.max = 20, ylab = "Sample Partial Autocorrelation", main = NA, lwd = 2, ylim = c(-0.5, 0.5))

# qq plot

>par(mfrow = c(1, 1))

>plot(fg11, which = 13) #9,10,11,13

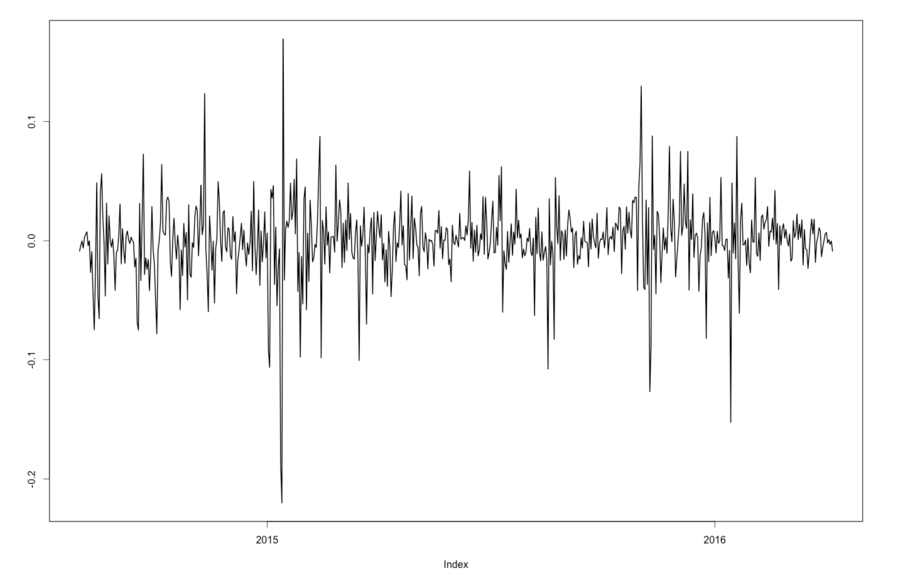
# kp test

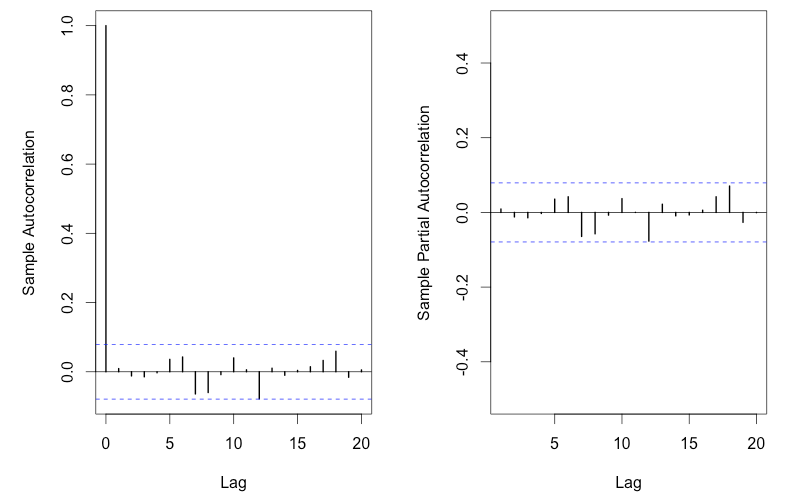
>set.seed(100)

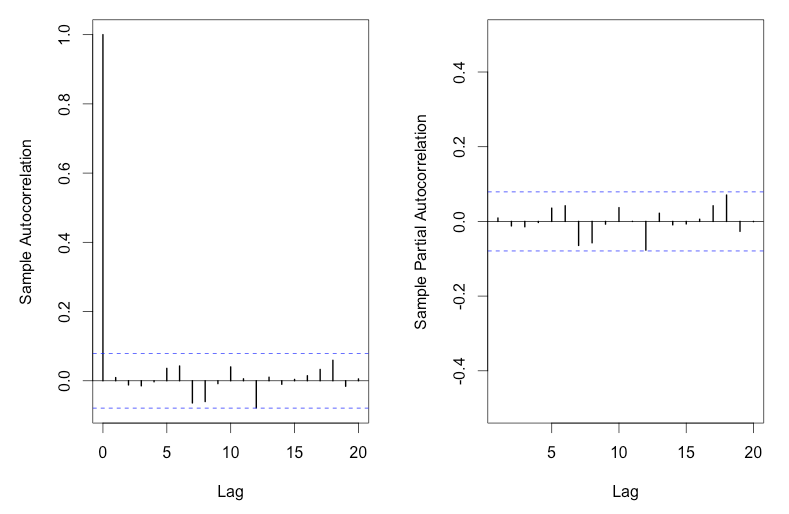
>x = rnorm(200)

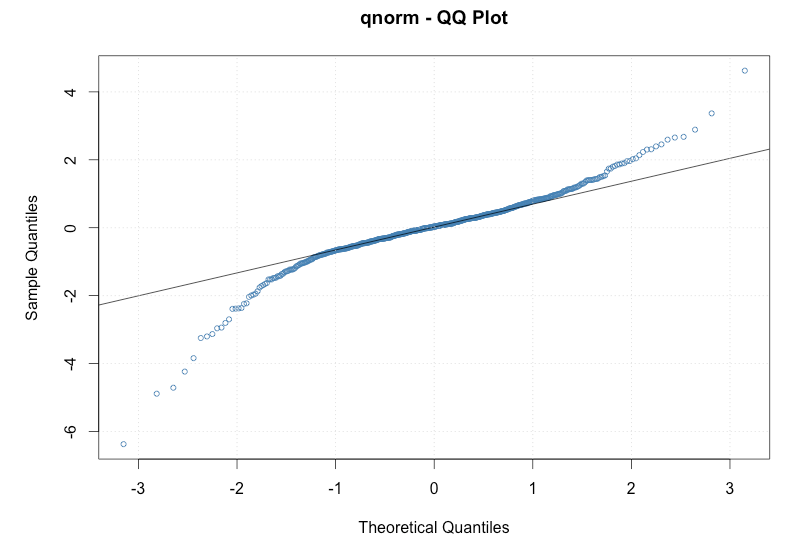
# Do x and y come from the same distribution?

>ks.test(x, fg11@residuals)









**Homework 5**

**Homework 5.1**

>install.packages("RCurl")

>install.packages("XML")

>library(RCurl)

>library(XML)

>url1 = "http://shakespeare.mit.edu/romeo\_juliet/full.html"

>html1 = readLines(url1, encoding = "UTF-8")

>html1 = htmlParse(html1, encoding = "UTF-8")

>url2 = "http://shakespeare.mit.edu/julius\_caesar/full.html"

>html2 = readLines(url2, encoding = "UTF-8")

>html2 = htmlParse(html2, encoding = "UTF-8")

>url3 = "http://shakespeare.mit.edu/hamlet/full.html"

>html3 = readLines(url3, encoding = "UTF-8")

>html3 = htmlParse(html3, encoding = "UTF-8")

>install.packages("bitops")

>install.packages("stringr")

>library(bitops)

>library(stringr)

>abs1 = lapply(url1, FUN = function(x) htmlParse(x, encoding = "Latin-1"))

>abs2 = lapply(url2, FUN = function(x) htmlParse(x, encoding = "Latin-1"))

>abs3 = lapply(url3, FUN = function(x) htmlParse(x, encoding = "Latin-1"))

>clean\_txt = function(x) {

+ cleantxt = xpathApply(x, "//body//text()

+ [not(ancestor :: script)][ not(ancestor :: style)]

+ [not(ancestor :: noscript)] " ,xmlValue)

+ cleantxt = paste(cleantxt, collapse="\n")

+ cleantxt = str\_replace\_all(cleantxt, "\n", " ")

+ cleantxt = str\_replace\_all(cleantxt, "\r", "")

+ cleantxt = str\_replace\_all(cleantxt, "\t", "")

+ cleantxt = str\_replace\_all(cleantxt, "<br>", "")

+ return(cleantxt)

+}

>cleantxt1 = lapply(abs1,clean\_txt)

>cleantxt2 = lapply(abs2,clean\_txt)

>cleantxt3 = lapply(abs3,clean\_txt)

>vec\_abs1 = unlist(cleantxt1)

>vec\_abs2 = unlist(cleantxt2)

>vec\_abs3 = unlist(cleantxt3)

>install.packages("tm")

>install.packages("SnowballC")

>library(tm)

>library(SnowballC)

>abs1 = Corpus(VectorSource(vec\_abs1))

>abs\_dtm1 = DocumentTermMatrix(abs1, control = list(stemming = TRUE, stopwords = TRUE, minWordLength = 3,removeNumbers = TRUE, removePunctuation = TRUE))

>abs2 = Corpus(VectorSource(vec\_abs2))

>abs\_dtm2 = DocumentTermMatrix(abs2, control = list(stemming = TRUE, stopwords = TRUE, minWordLength = 3,removeNumbers = TRUE, removePunctuation = TRUE))

>abs3 = Corpus(VectorSource(vec\_abs3))

>abs\_dtm3 = DocumentTermMatrix(abs3, control = list(stemming = TRUE, stopwords = TRUE, minWordLength = 3,removeNumbers = TRUE, removePunctuation = TRUE))

>install.packages("ggplot2")

>install.packages("wordcloud")

>library(ggplot2)

>library(wordcloud)

>freq1 = colSums(as.matrix(abs\_dtm1))

>wf1 = data.frame(word=names(freq1), freq=freq1)

>freq2 = colSums(as.matrix(abs\_dtm2))

>wf2 = data.frame(word=names(freq2), freq=freq2)

>freq3 = colSums(as.matrix(abs\_dtm3))

>wf3 = data.frame(word=names(freq3), freq=freq3)

>plot1 = ggplot(subset(wf1, freq>15), aes(word, freq1))

>plot1 = plot1 + geom\_bar(stat="identity")

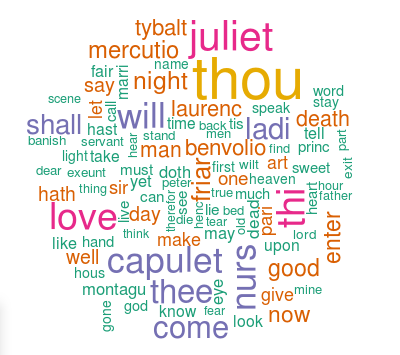
>plot1 = plot1 + theme(axis.text.x=element\_text(angle=45, hjust=1))

>plot1

>freq1 = colSums(as.matrix(abs\_dtm1))

>dark2\_1 = brewer.pal(6, "Dark2")

>wordcloud(names(freq1), freq1, max.words=100, rot.per=0.2, colors=dark2\_1)



>plot2 = ggplot(subset(wf2, freq>15), aes(word, freq2))

>plot2 = plot2 + geom\_bar(stat="identity")

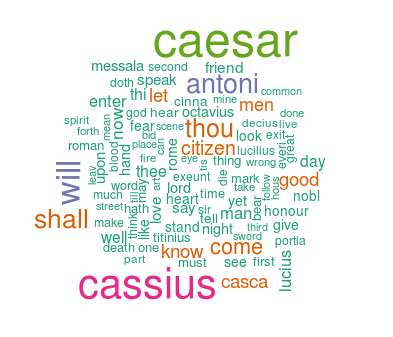
>plot2 = plot2 + theme(axis.text.x=element\_text(angle=45, hjust=1))

>plot2

>freq2 = colSums(as.matrix(abs\_dtm2))

>dark2\_2 = brewer.pal(6, "Dark2")

>wordcloud(names(freq2), freq2, max.words=100, rot.per=0.2, colors=dark2\_2)



>plot3 = ggplot(subset(wf3, freq>15), aes(word, freq3))

>plot3 = plot3 + geom\_bar(stat="identity")

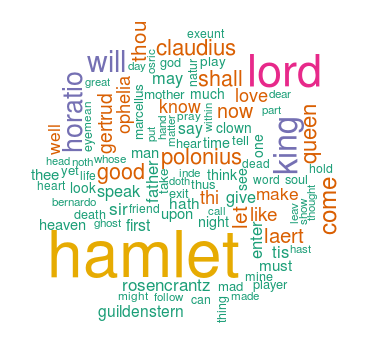
>plot3 = plot3 + theme(axis.text.x=element\_text(angle=45, hjust=1))

>plot3

>freq3 = colSums(as.matrix(abs\_dtm3))

>dark2\_3 = brewer.pal(6, "Dark2")

>wordcloud(names(freq3), freq3, max.words=100, rot.per=0.2, colors=dark2\_3)



**Homework 5.2**

#Romeo and Juliet

>wf1 <- wf1[order(-wf1$freq),]

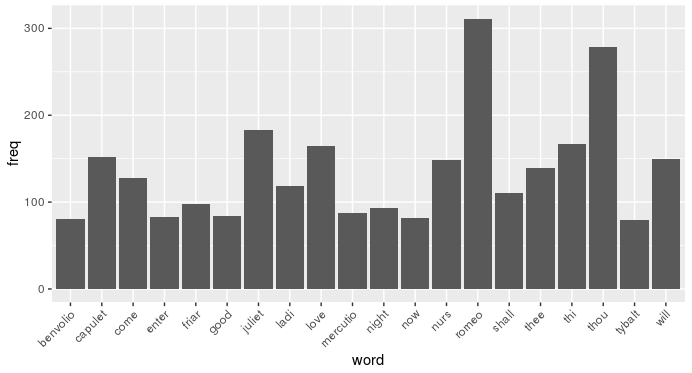
>wf1 <- wf1[c(1:20),]

>p1 = ggplot(subset(wf1, freq > 15), aes(word, freq))

>p1 = p1 + geom\_bar(stat = "identity")

>p1 = p1 + theme(axis.text.x = element\_text(angle = 45, hjust = 1))

>p1



#Julius Caeser

>wf2 <- wf2[order(-wf2$freq),]

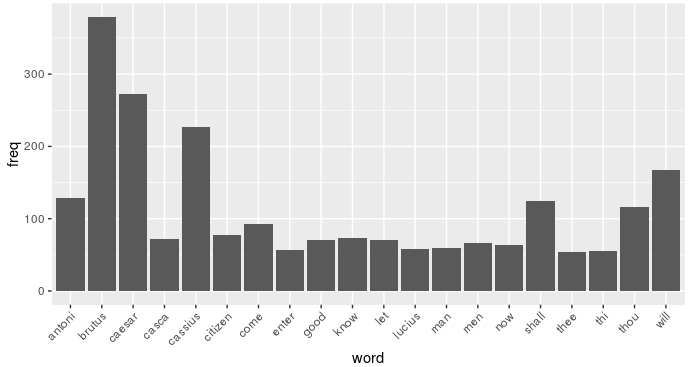
>wf2 <- wf2[c(1:20),]

>p2 = ggplot(subset(wf2, freq > 15), aes(word, freq))

>p2 = p2 + geom\_bar(stat = "identity")

>p2 = p2 + theme(axis.text.x = element\_text(angle = 45, hjust = 1))

>p2



#Hamlet

>wf3 <- wf3[order(-wf3$freq),]

>wf3 <- wf3[c(1:20),]

>p3 = ggplot(subset(wf3, freq > 15), aes(word, freq))

>p3 = p3 + geom\_bar(stat = "identity")

>p3 = p3 + theme(axis.text.x = element\_text(angle = 45, hjust = 1))

>p3

