**Chenyuan Wang 王陈圆 15620161152281**

**HW Unit 1:**

* 1. **Calculate the increase of memory of PCs over the last 30 years and check whether theFMRI analysis could have been done 20 years ago.**

Answer:

|  |  |
| --- | --- |
| Year | RAM /G |
| 1988 | 0.002 |
| 1991 | 0.004 |
| 1996 | 0.5 |
| 2000 | 1 |
| 2003 | 2 |
| 2007 | 8 |
| 2014 | 16 |

The FMRI analysis could have been done 20 years ago

* 1. **prepare 2-5 slides explaining logistic regression**

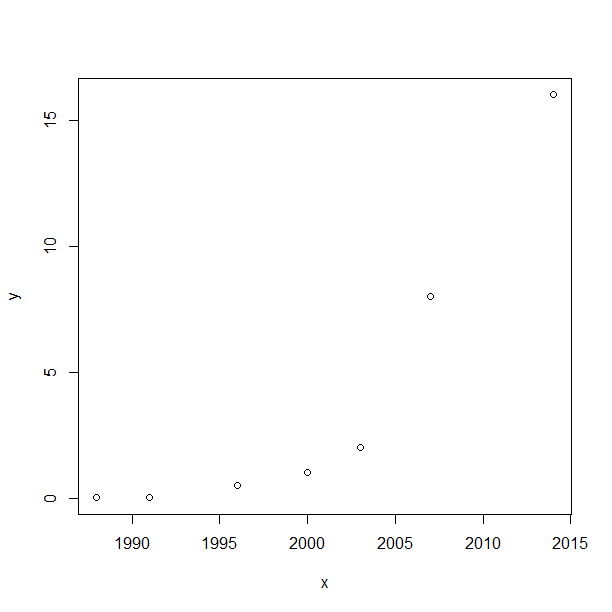
Answer:

* In statistics, logistic regression, or logit regression, or logit model is a regression model where the dependent variable (DV) is categorical.Cases where the dependent variable has more than two outcome categories may be analysed in multinomial
* In the terminology of economics, logistic regression is an example of a qualitative response/discrete choice model.
* Logistic regression was developed by statistician David Cox in 1958. The binary logistic model is used to estimate the probability of a binary response based on one or more predictor (or independent) variables (features). It allows one to say that the presence of a risk factor increases the odds of a given outcome by a specific factor
* For example, to explore the risk factors for the occurrence of gastric cancer, you can choose two groups of people, one group is gastric cancer group, one group is non-gastric cancer group.
* Here the dependent variable is whether the gastric cancer, that is, "yes" or "no", for the two categories of variables, independent variables can include a lot, such as age, gender, eating habits, Helicobacter pylori infection. The independent variables can be either continuous or categorical. By logistic regression analysis, you can get a general idea of which factors are risk factors for gastric cancer.

**1.3 install R and run simple programs from Quantlet.de, make sure you have a Github(GH) account.**

**HW Unit 2**

**2.1 make an R quantlet to solve HW #1 from unit 1 with R and show it on Github (GH). hint: use the CMB Qs for this work**

Answer:

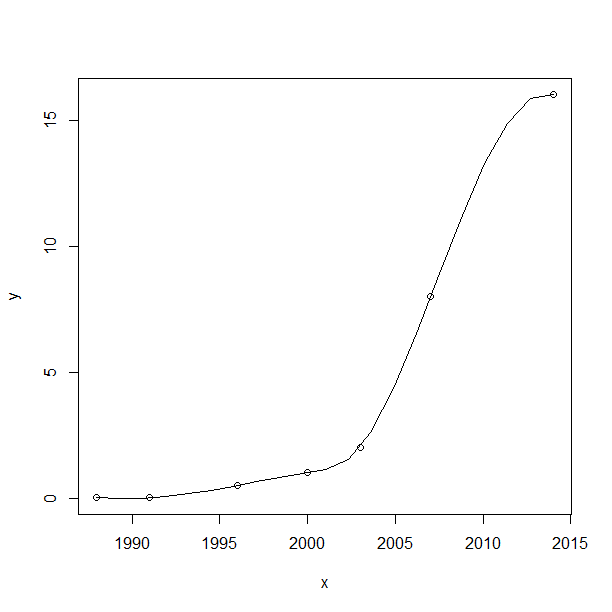
data<-read.table(“file",header=TRUE,sep=",")

x<-data$Year

y<-data$RAM

plot(x,y)

**2. 2 use R with B-spline code to solve HW#1, any comments?**

Answer:

data<-read.table(“file",header=TRUE,sep=",")

x<-data$Year

y<-data$RAM

plot(x,y)

lines(spline(x,y))

comment:memory of PCs over the last 30

years are increasing, especially in the last

10 years.

**2.3 Suppose you observe that in n=1000 mails (in 1 week) you have about 2 scams. Use the LvB /Poisson cdf to calculate that you have 6 scam emails in 2 weeks. In Scammyland you have 5 scams on average, what is the probability to have no scam mail.**

Answer:

x = 6

n = 1000

lambda = 2

p = lambda / n

dbinom (x,2\*n,p) # binomial probability mass function

[1] 0.1042477

dpois (x, 2\*lambda ) # Poisson probability mass function

[1] 0.1041956

dpois (0, 5 )

[1] 0.006737947

**HW Unit 3**

* 1. **make an R quantlet on GH to produce hash code for the 2 sentences: „I learn a lot from this class when I am proper listening to the professor“, „I do not learn a lot from this class when I am absent and playing on my Iphone“. Compare the 2 hash sequences**

Answer:

# call the library doing the hashes

library("digest")

# now do the hash code calculation

digest ("I learn a lot from this class when I am proper listening to the professor")

[1] "a8d3e4701672195e5dcd16ea9b062279"

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

[1] "497edecd95aca5cc9a581e4835c3cccd"

**3.2 Make 3-5 slides (in PPTX) on the DSA (Digital Signature Algorithms)**

* The Digital Signature Algorithm (DSA) is a Federal Information Processing Standard for digital signatures.
* A digital signature algorithm is a subset of the digital signature standard that represents a specific public key algorithm that is used only as a digital signature.
* The key runs on the message hash generated by SHA-1: To verify a signature, recalculate the message's hash, use the public key to decrypt the signature and then compare the results.
* The implementation of a digital signature is usually done by the sender of the message through a one-way function to process the message to be transmitted to produce a string of digits that can not be forged by another person to authenticate the source of the message and to detect whether the message has been modified.
* The message receiver decrypts the received message encrypted with the sender's private key with the sender's public key, and determines the source and integrity of the message, and the sender can not deny the signature.

**3. 3 Make slides with R code where you create a JSON data set that you save and read again.**

Answer:

* **Creat json data:**

{

"ID":["1","2","3","4","5"],

"Name":["Alice","Bob","Dan","Pat","Hua"],

"Grades":["88","89","98","94","90"]

}

* **Save data** :  
   C:/Users/王陈圆/Desktop/hw3jsondata.json
* **Read Data:**

# Load the package required to read JSON files.

ibrary("rjson")

# Give the input file name to the function.

result <- fromJSON(file = "C:/Users/王陈圆/Desktop/hw3jsondata.json")

# Print the result.

print(result)

* **Result**

$ID

[1] "1" "2" "3" "4" "5"

$Name

[1] "Alice" "Bob" "Dan" "Pat" "Hua"

$Grades

[1] "88" "89" "98" "94" "90"

**3.4 Download the CRIX data and make a plot of the time series, analyse its properties, i.e. fit ARMA, ARIMA etc. Is there a GARCH effect?**

**Answer:**

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)

a<-seq(1,2348,2)

b<-seq(2,2348,2)

date<-t(crix\_data\_frame[1,a])

price<-t(crix\_data\_frame[1,b])

return<-1:1174

for(i in 1:1174)

{return[i+1]<-log(price[i+1]/price[i])}

new<-data.frame(date,price,return[1:1174])

names(new)<-c("date","price","return")

plot(new$date,new$return)

arima(new$return, order = c(2,0,1))

#These results suggest that the CRIX return series can be modeled by some ARIMA process, for example ARIMA(2, 0, 2).

Call:

arima(x = new$return, order = c(2, 0, 2))

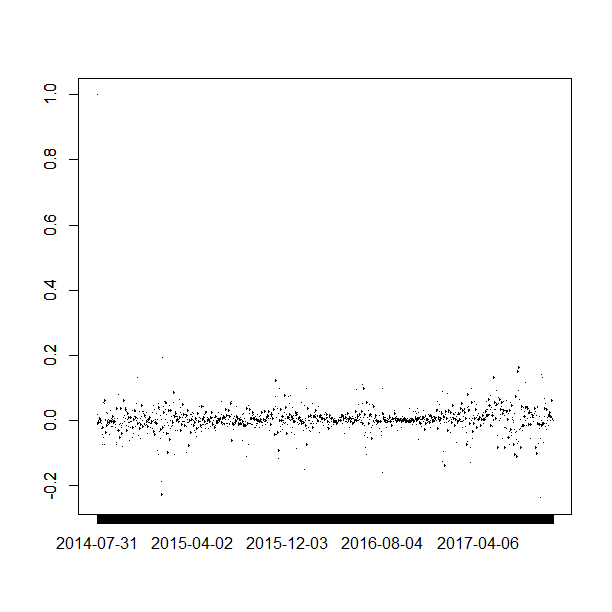
Coefficients:

ar1 ar2 ma1 ma2 intercept

-0.7564 -0.8752 0.7584 0.8281 0.0031

s.e. 0.0757 0.2050 0.0881 0.2261 0.0013

sigma^2 estimated as 0.002041: log likelihood = 1970.32, aic = -3928.65



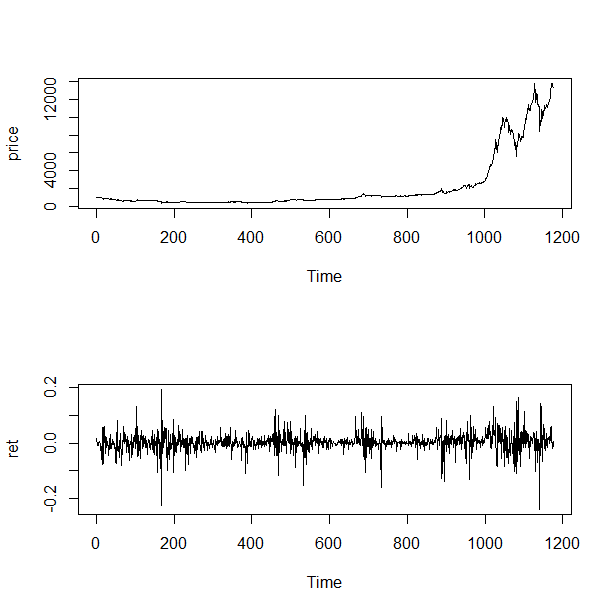
**HW Unit 4**

**(all this to be done on perfect PPTX slides)**

**4.1. improve the R quantlets on GH (from CRIX directory on quantlet.de) and make excellent graphics that follow Fig 3,4,5,6 of the „Econometrics of CRIX“ paper.**

**Answer：**

* **The daily value of indices in the CRIX and The log returns of CRIX index**

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)

n<-dim(crix\_data\_frame)

a<-seq(1,n[2],2)

b<-seq(2,n[2],2)

date<-t(crix\_data\_frame[1,a])

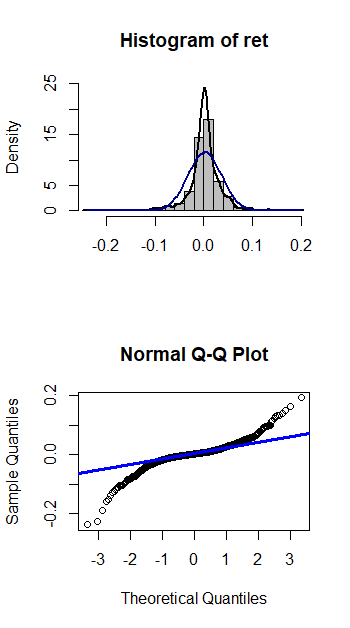
price<-t(crix\_data\_frame[1,b])

ts.plot(price)

ret<-diff(log(price))

par(mfrow=c(2,1))

ts.plot(ret)

* **Histogram and QQ plot of CRIX returns**

# histogram of returns

par(mfrow=c(2,1))

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)

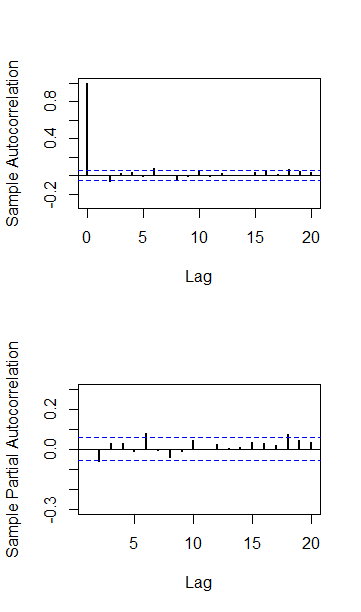
curve(dnorm(x, mean = mean(ret), sd = sd(ret)),

add = TRUE, col = "darkblue", lwd = 2)

# qq-plot

qqnorm(ret)

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

* **The sample ACF and PACF of CRIX returns**

# acf plot

autocorr = acf(ret, lag.max = 20,

ylab = "Sample Autocorrelation", main = NA,

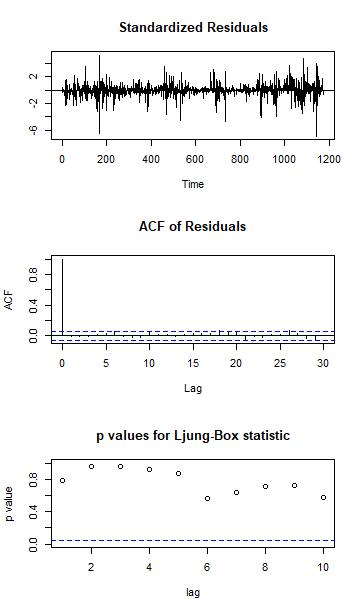
lwd = 2, ylim = c(-0.3, 1))

# plot of pacf

autopcorr = pacf(ret, lag.max = 20,

ylab = "Sample Partial Autocorrelation",

main = NA, ylim = c(-0.3, 0.3), lwd = 2)

* **Diagnostic Checking**

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

tsdiag(fit202)

**4.2. make your R code perfect as in the R examples on quantlet.de i.e. make sure that the code is „time independent“ by using actual dimensions of the data that you are collecting from crix.hu-berlin.de RecreateFig 7 from „Econometrics of CRIX“.**

Answer:

CRIX returns and predicted values

# arima202 predict

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

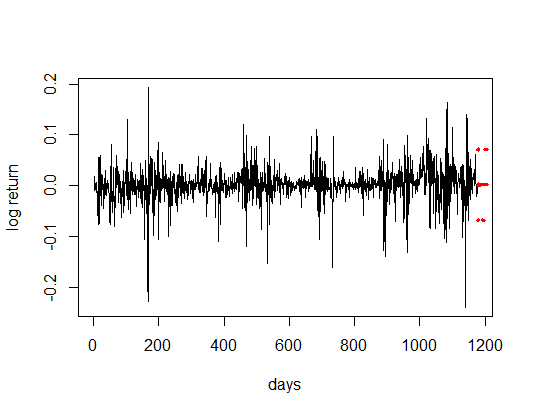
crpre = predict(fit202, n.ahead = 30)

plot(ret, type = "l", 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)



**HW Unit 5**

**5.1 Do a word cloud for Shakesspeare’s dramas. Romeo and Julia, Julius Caesar, Hamlet.**

Answer:

#pickup some sentenes form ROMEO and JULIET

vec\_abs<-c(

"ROMEO: She speaks: O, speak again, bright angel!for thou art As glorious to this night, being o'er my head As is a winged messenger of heaven Unto the white-upturned wondering eyes Of mortals that fall back to gazeon himWhen he bestrides the lazy-pacing clouds And sails upon the bosomof the air. ",

"JULIET: O Romeo, Romeo! whereforeart thou Romeo? Deny thy father and refuse thy name; Or, if thou wilt not，be but sworn my love, And I'll no longer be a Capulet.",

"ROMEO: Shall I hear more, or shall I speak at this?",

"JULIET: 'Tisbut thy name that is my enemy; Thou art thyself, though not a Montague. What's Montague? it is nor hand, nor foot, Nor arm, nor face, nor any other part Belonging to a man. O, be some other name! What's in a name? that which we call a roseBy any other name would smell as sweet; So Romeo would, were he not Romeo call'd, Retain that dear perfection which he owes Without that title. Romeo, doffthy name, And for that name which is no part of thee Take all myself.",

"ROMEO: I take thee at thy word: Call me but love, and I'll be new baptized, HenceforthI never will be Romeo. ",

"JULIET: What man art thou that thus bescreen'din night So stumblest on my counsel? ",

"ROMEO: By a nameI know not haw to tell thee who I am: My name, dear saint, is hateful to myself, Because it is an enemy to thee, Had I it written,I would tear the word. ",

"JULIET: My ears have not yet drunk a hundred words Of that tongue's utterance, yet I know the sound:Art thou not Romeo and a Montague? ",

"ROMEO: Neither, fair saint, if either thee dislike. ",

"JULIET: How camest thou hither, tell me, and wherefore? The orchard walls are high and hard to climb, And the place death, considering who thou art, If any of my kinsmen find thee here.",

"ROMEO: With love's light wings did I o'er-perch these walls, For stonylimits cannot hold love out,And what love can do that dares love attempt; Therefore thy kinsmen are no let to me. ",

"JULIET: If they do see thee, they will murder thee."

)

#Create a corpus & Term Document Matrix

library(NLP)

library(tm)

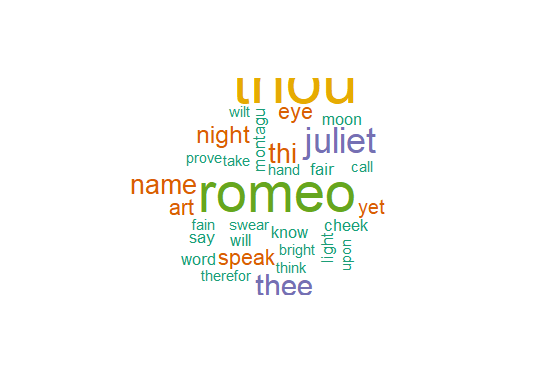
library(SnowballC)

abs = Corpus(VectorSource(vec\_abs))

abs\_dtm = DocumentTermMatrix(abs, control = list(

stemming = TRUE, stopwords = TRUE, minWordLength = 3,

removeNumbers = TRUE, removePunctuation = TRUE))

dim(abs\_dtm)

inspect(abs\_dtm)

#Find the words that occur more than 3 times

findFreqTerms(abs\_dtm, 3)

#Remove sparse terms

removeSparseTerms(abs\_dtm, 0.5)

inspect(removeSparseTerms(abs\_dtm, 0.5))

# do a word cloud

library(ggplot2)

library(RColorBrewer)

library(wordcloud)

freq = colSums(as.matrix(abs\_dtm))

wf = data.frame(word=names(freq), freq=freq)

dark2 = brewer.pal(6, "Dark2")

wordcloud(names(freq), freq, max.words=100, rot.per=0.2, colors=dark2)

**2. Calculate the histogram of words**

**Answer:**

plot = ggplot(subset(wf, freq>1), aes(word, freq))

plot = plot + geom\_bar(stat="identity")

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

plot