



报告题目： **Final Exam Project\_\_**

姓 名： **陈 睿 15620161152244**

任课老师： **Wolfgang 教授\_\_\_\_**

日 期：  **2017 年 10月 28 日\_**

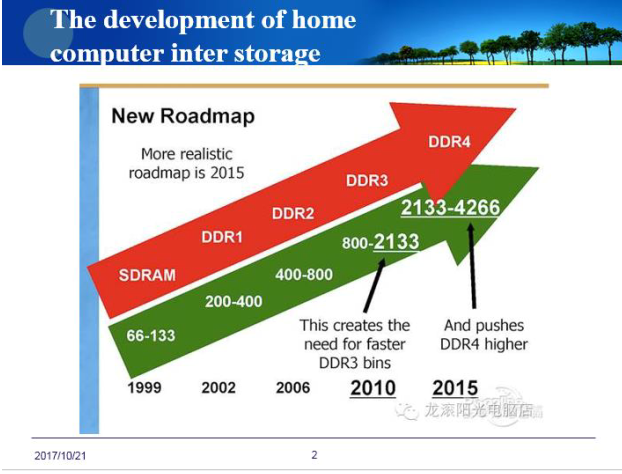
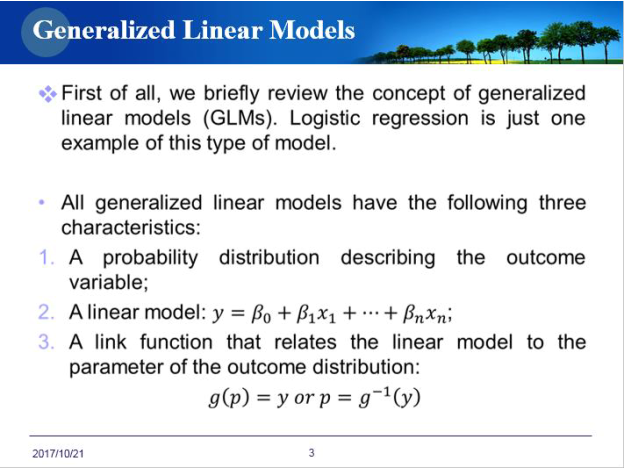
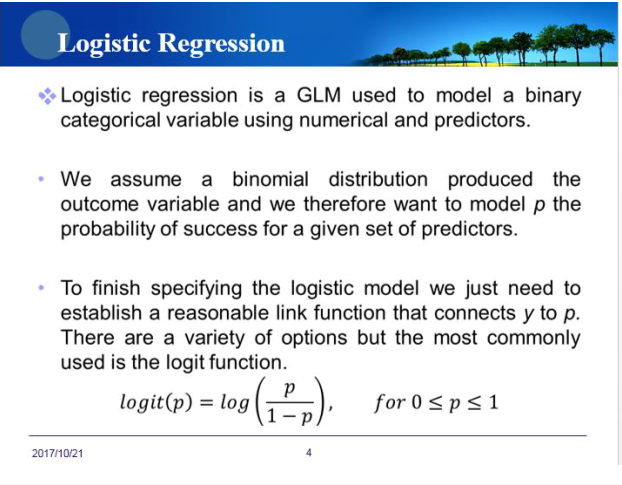
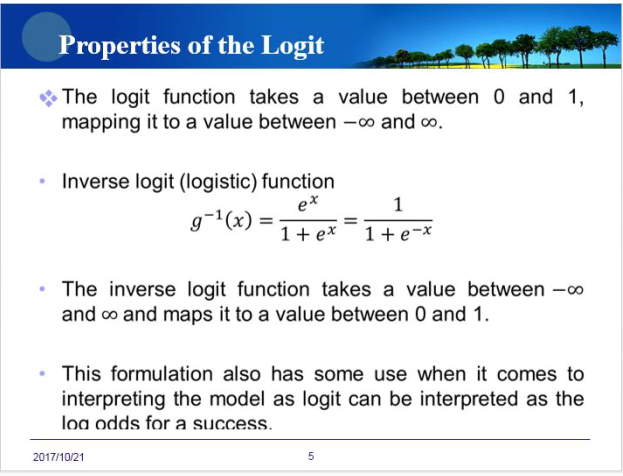
**HW Unit1**

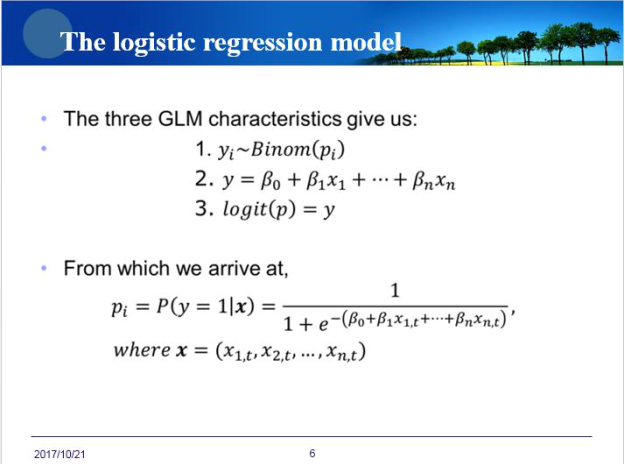
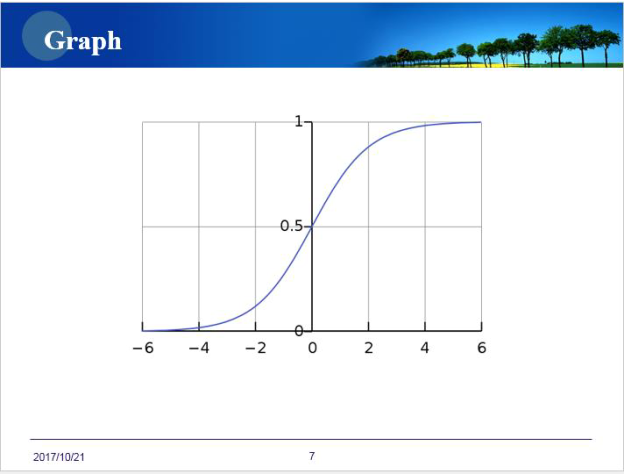
* 1. **Calculate the increase of memory of PCs over the last 30 years and check whether the FMRI analysis could have been done 20 years ago.**

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

**Comments: There was the rather slow growth before 2000, while there is dramatically great increase after 2000.**

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

* 1. **Apply for an account for Github.**

**My account is** [**22457114@qq.com**](mailto:22457114@qq.com)**, RuiChan244.**

**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.**

memory.df = read.csv("byte.csv",header = TRUE)

plot(memory.df$Byte~memory.df$year,type="o",main="The development of internal memory")

**2.2 Use R with B-spline code to solve HW#1, any comments?**

splines.reg.l1 = smooth.spline(x = memory.df$year, y = memory.df$Byte, spar = 0.2)

splines.reg.l2 = smooth.spline(x = memory.df$year, y = memory.df$Byte, spar = 1)

splines.reg.l3= smooth.spline(x = memory.df$year, y = memory.df$Byte, spar = 2)

lines(splines.reg.l1, col = "green", lwd = 2)

lines(splines.reg.l2, col = "pink", lwd = 2)

lines(splines.reg.l3, col = "blue", lwd = 2)

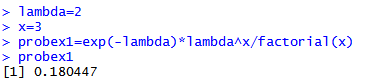
**2.3 Suppose you observe that in n=1000 mails (in 1 week) you have about 2 scams. Use the LvB /Poisson cdf to calculate that you have 6 scam emails in 2 weeks. In Scammyland you have 5 scams on average, what is the probability to have no scam mail.**

lambda=2

x=3

probex1=exp(-lambda)\*lambda^x/factorial(x)

probex1

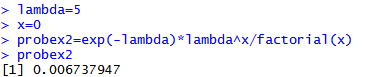


lambda=5

x=0

probex2=exp(-lambda)\*lambda^x/factorial(x)

probex2



**HW Unit 3**

**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.**

# install stuff for hash calculation

install.packages("digest")

# call the library doing the hashes

library("digest")

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

#"a8d3e4701672195e5dcd16ea9b062279"

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

#"059ab10d478614d2eab3d70cfccd3fcc"

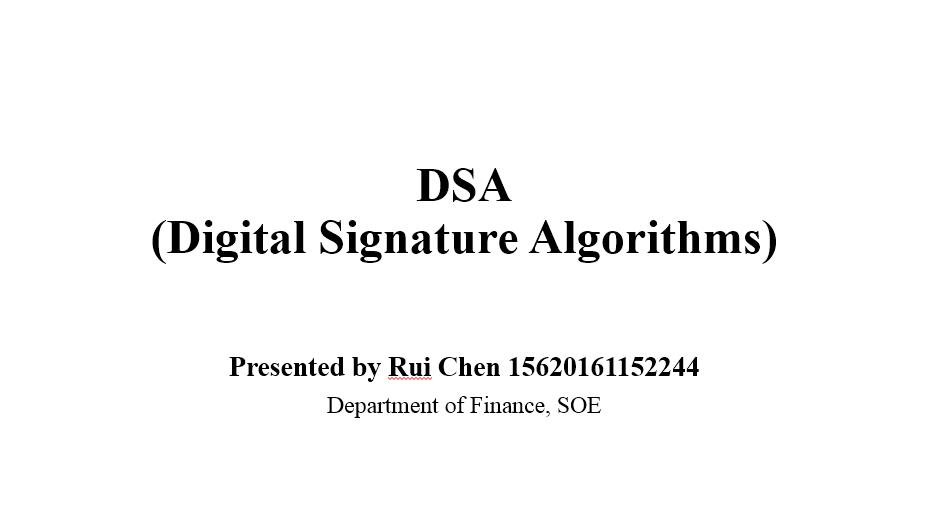
digest("I learn a lot from this class when I am proper listening to the professor","sha256")

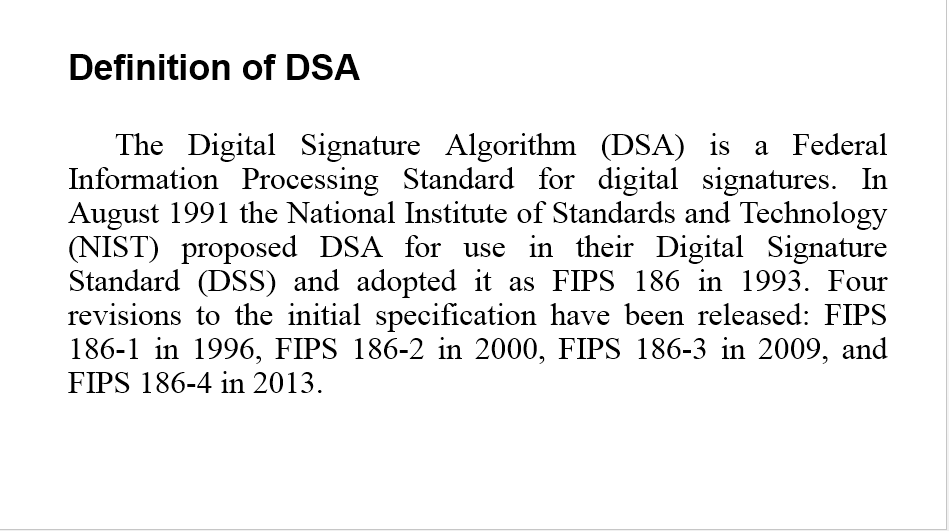
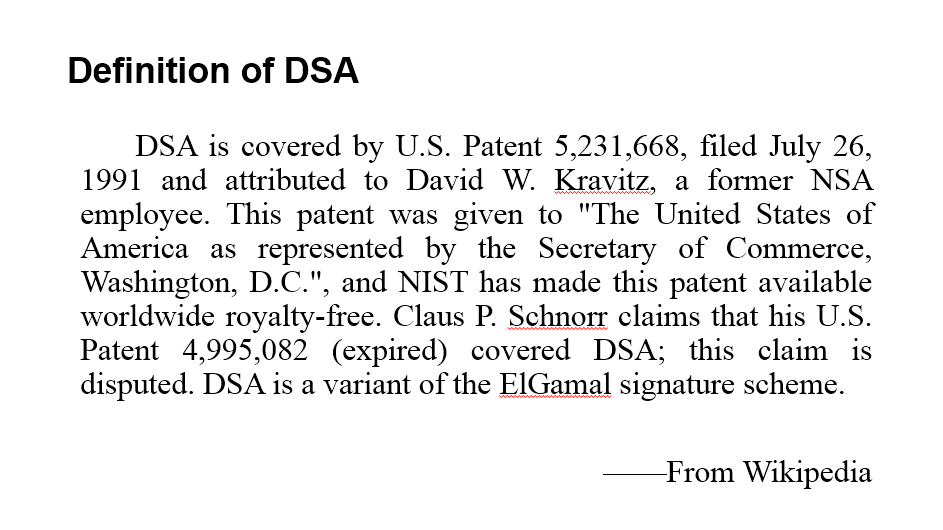
#"c16700de5a5c1961e279135f2be7dcf9c187cb6b21ac8032308c715e1ce9964c"

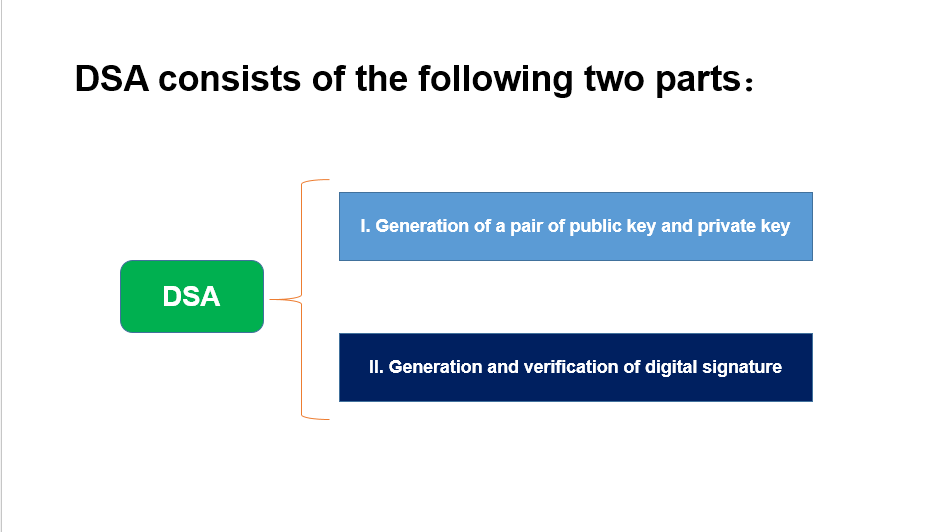
digest("I do not learn a lot from this class when I am absent and playing on my phone","sha256")

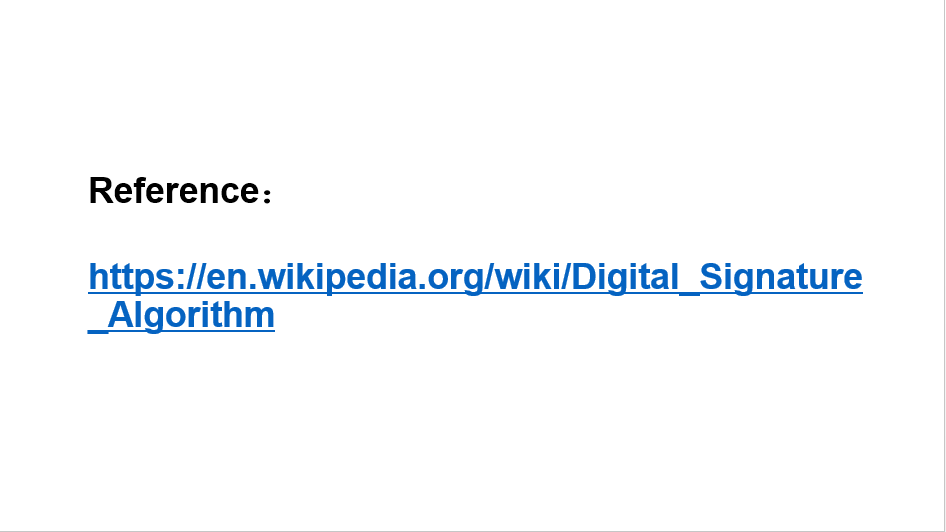
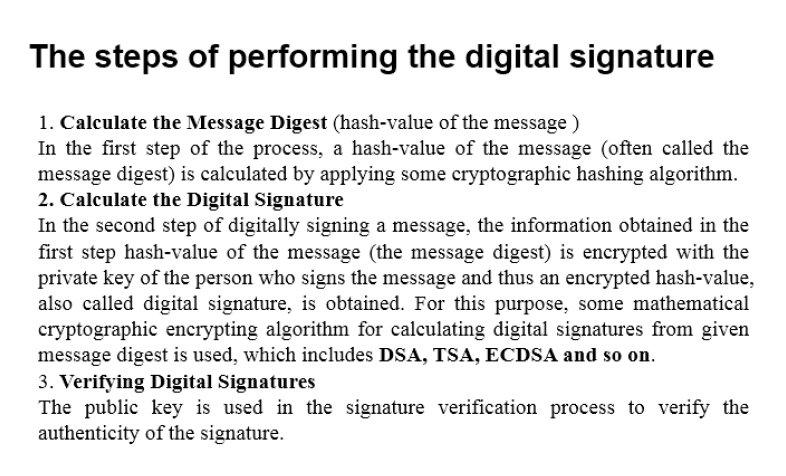
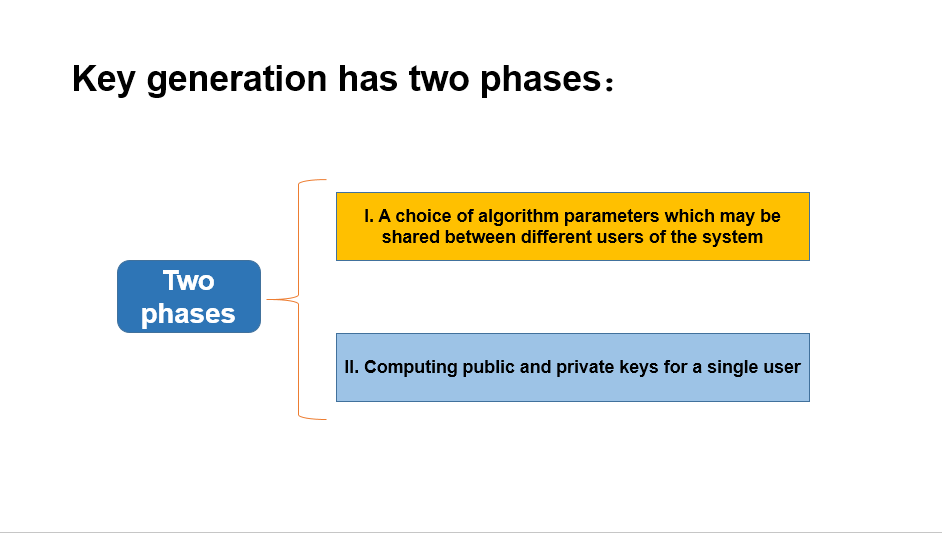
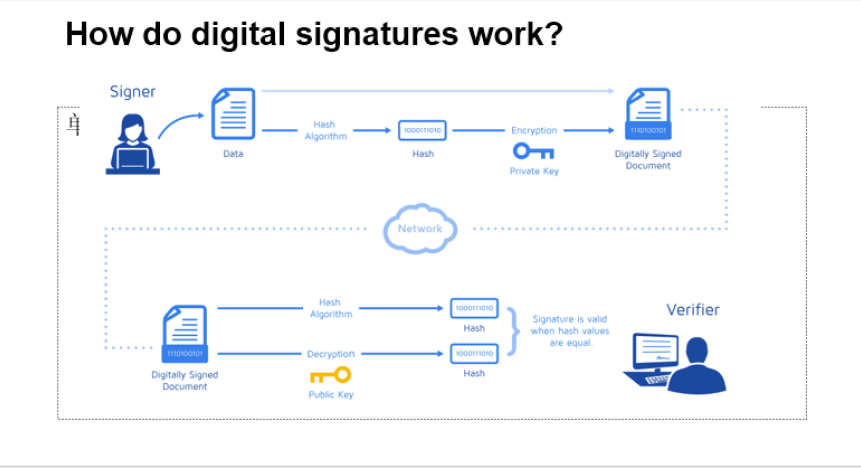
#"f5e2cba48dac097355d0bb310fdbd5bd38a22a5c8e8215cd1ae67014cfc35b91"

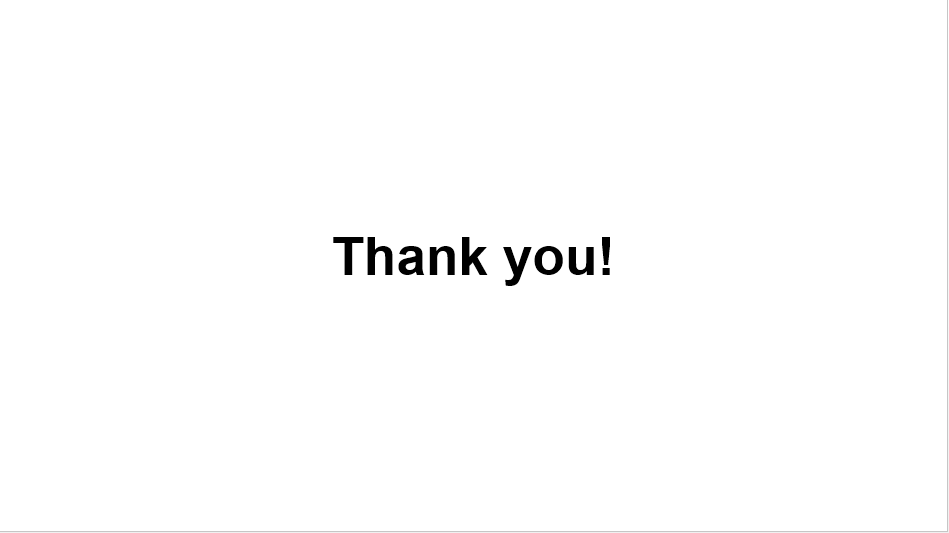
**3.2 Make 3-5 slides (in PPTX) on the DSA (Digital Signature Algorithms)**









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

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)

x = as.data.frame(json\_data) date1=c(json\_data[[1]]$date)

for (i in 1:50){date1[i]=c(json\_data[[i]]$date)}

price1=c(json\_data[[1]]$price)

for (i in 1:50){ price1[i]=c(json\_data[[i]]$price)}

date=date1

price=price1

crix=data.frame(date,price)

plot(crix$price~as.Date(crix$date))

plot(crix$price~crix$date,type="b")

plot(ts(crix$price,freq=1),type='l',xlab='Day',ylab='Price')

**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?**

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

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

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

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

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

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

library(caschrono)

library(TTR)

library(fGarch)

library(rugarch)

library(forecast)

library(TSA)

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ARIMA model\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

xy.acfb(crix$price,numer=FALSE)

adf.test(crix$price)

#Augmented Dickey-Fuller Test:not stationary

##\*\*\*\*\*1)return

r=diff(log(crix$price))\*100

plot(r,type="b")

abline(h = 0)

plot(r,type="l")

xy.acfb(r,numer=FALSE)

#\*\*\*\*\*2)Parameter Estimation

#estimation of p and q

a.fin2=arima(r,order=c(2,0,2))

summary(a.fin2) f=forecast(a.fin2,h=3,level=c(99.5)) acf(f$residuals,lag.max = 20) Box.test(f$residuals,lag=20,type='Ljung-Box') #the residuals follow Gaussian distribution plot.ts(f$residuals)

#\*\*\*\*3)some evidence to GARCH model

#get ACF and PACF of the residuals xy.acfb(residuals(a.fin2),numer=FALSE) xy.acfb((residuals(a.fin2))^2,numer=FALSE)+ xy.acfb(abs(residuals(a.fin2)),numer=FALSE)

#get the Conditional heteroskedasticity test McLeod.Li.test(y=residuals(a.fin2))

#p-values are all included in the test, it formally shows strong evidence for ARCH in this data.

#\*\*Normality of the Residuals

qqnorm(residuals(a.fin2))

qqline(residuals(a.fin2))

shapiro.test(residuals(a.fin2))

#The QQ plot suggest that the distribution of returns may have a tail thicker that of a

#normal distribution and maybe somewhat skewed to the right

#p-value<0.05 reject the normality hypothesis

g1=garchFit(~garch(1,1),data=residuals(a.fin2),trace=FALSE,include.me an=TRUE, na.action=na.pass)

summary(g1) g2=garchFit(~garch(1,2),data=residuals(a.fin2),trace=FALSE,include.me an=TRUE, na.action=na.pass)

summary(g2) g3=garchFit(~garch(2,1),data=residuals(a.fin2),trace=FALSE,include.me an=TRUE, na.action=na.pass)

summary(g3) g4=garchFit(~garch(2,2),data=residuals(a.fin2),trace=FALSE,include.me an=TRUE, na.action=na.pass)

summary(g4)

#The best one is Garch(1,1) model which has the smallest AIC.

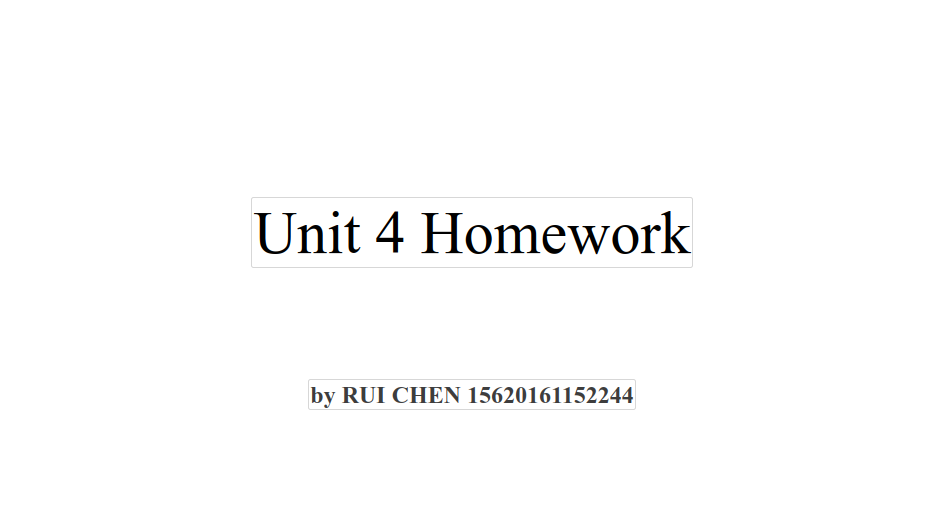
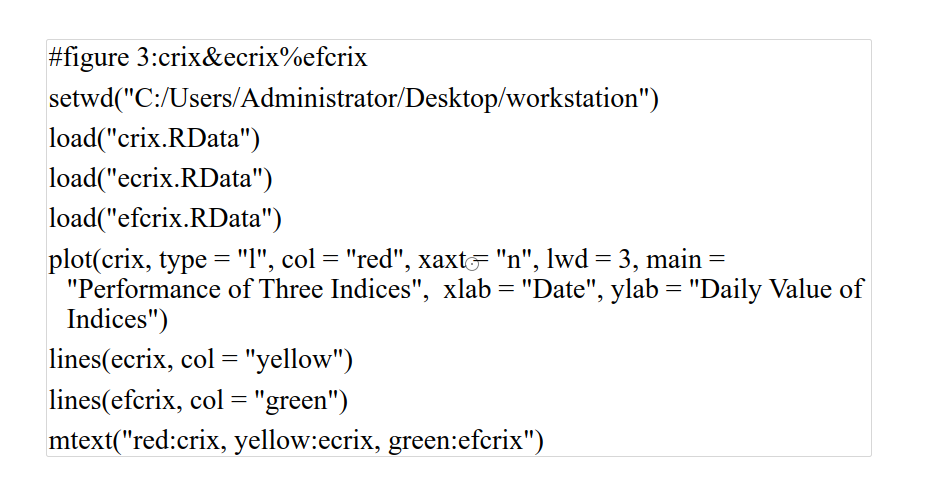
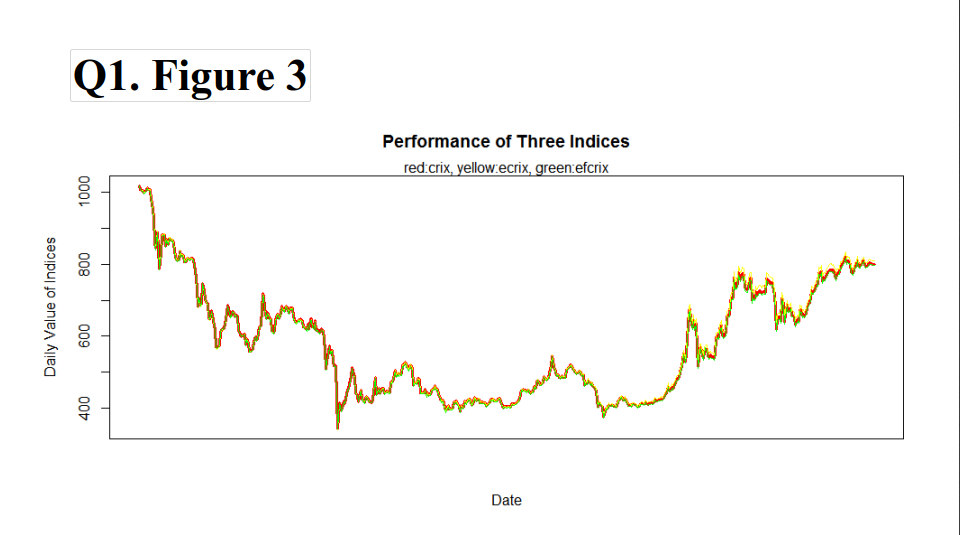
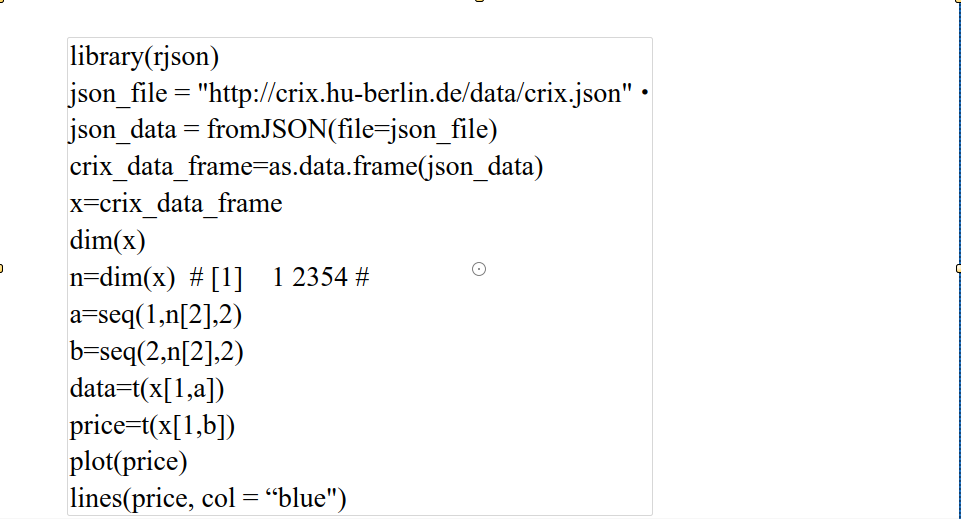
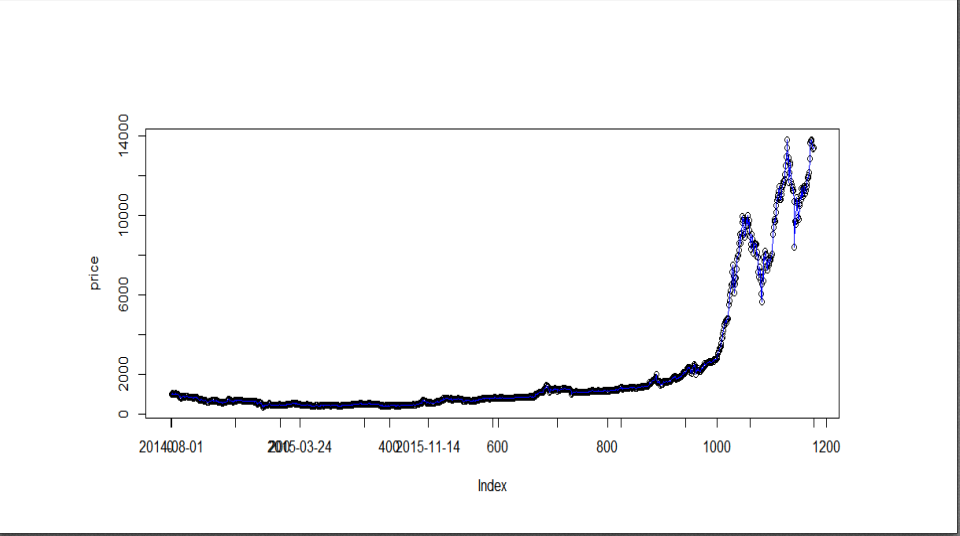
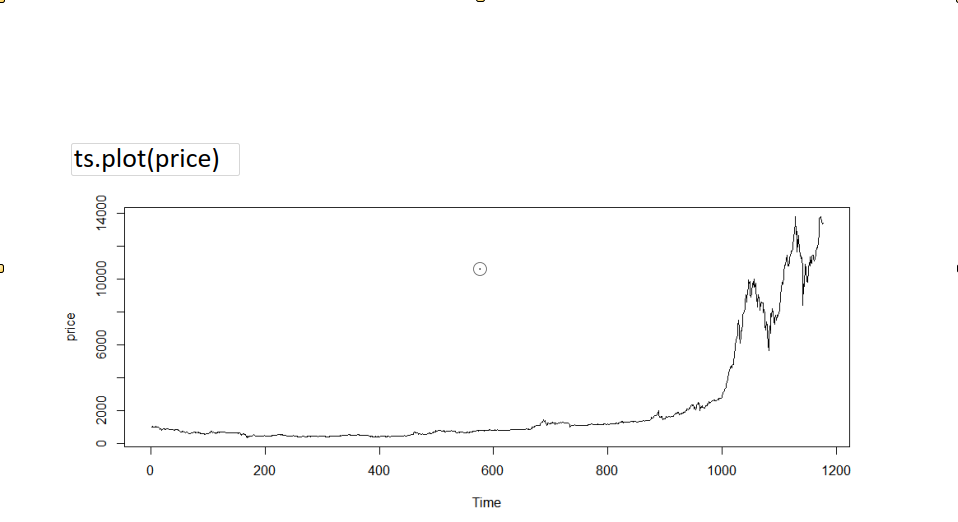
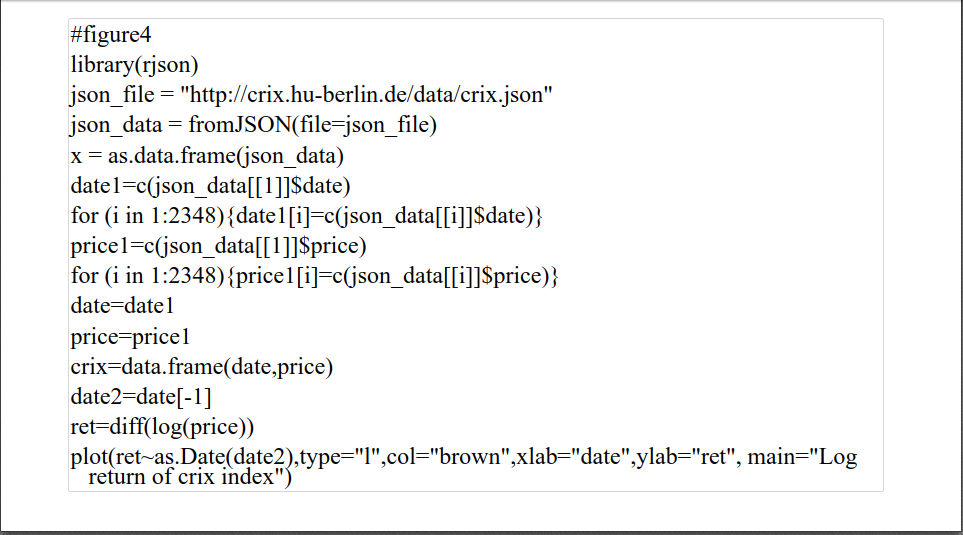
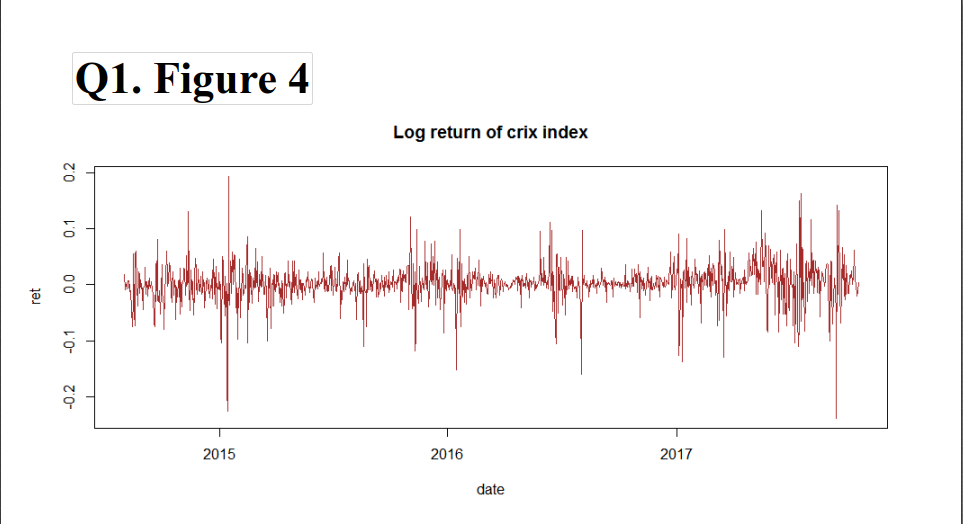
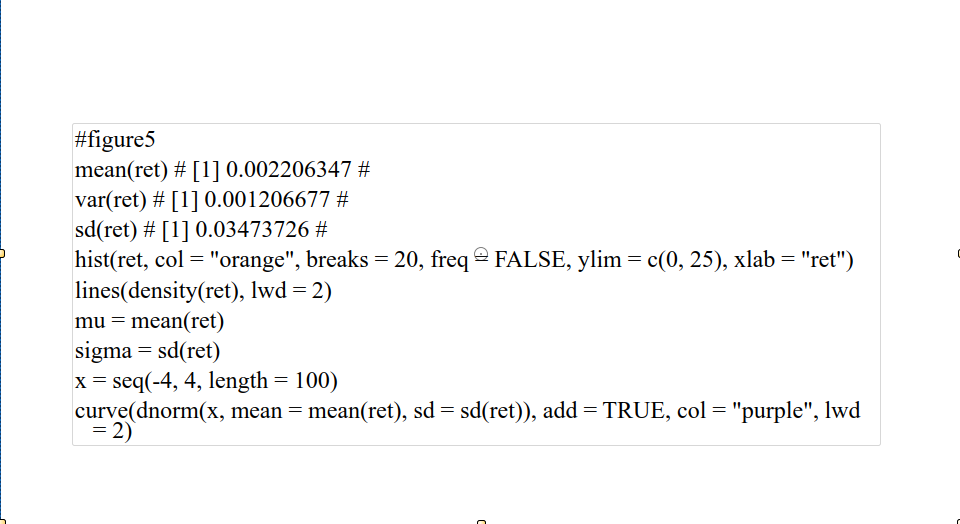
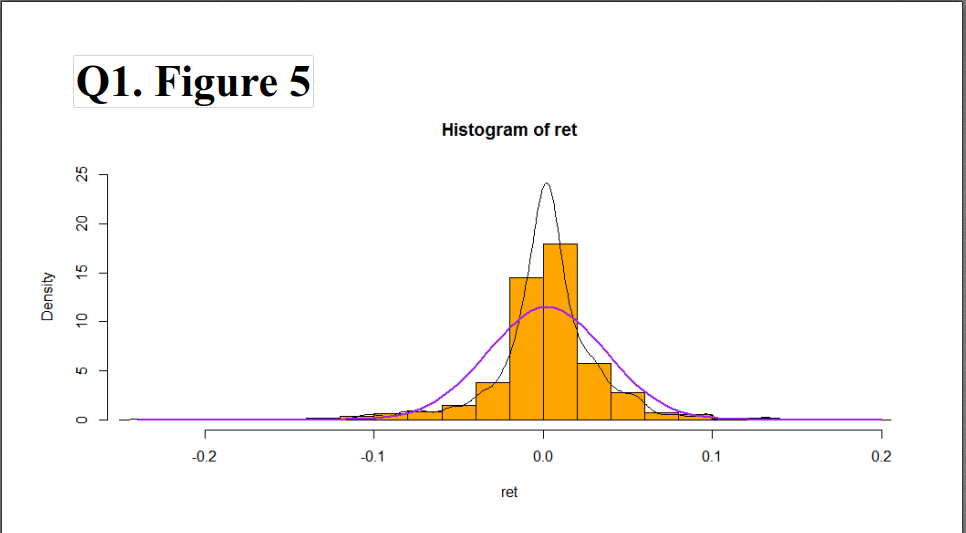
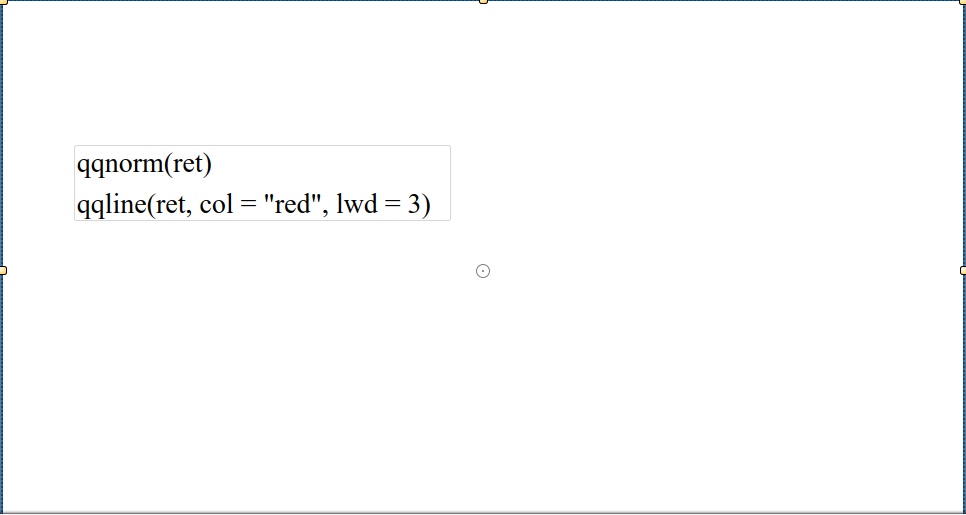
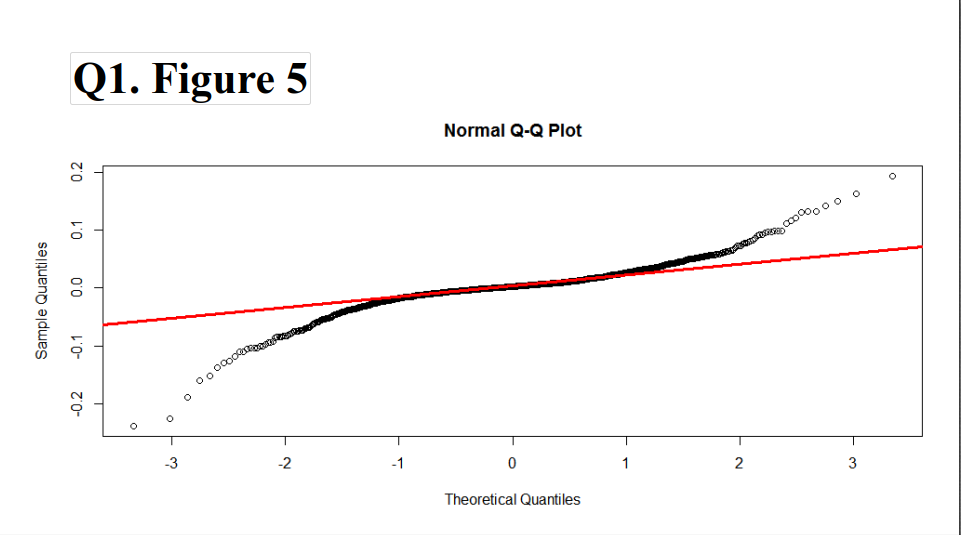
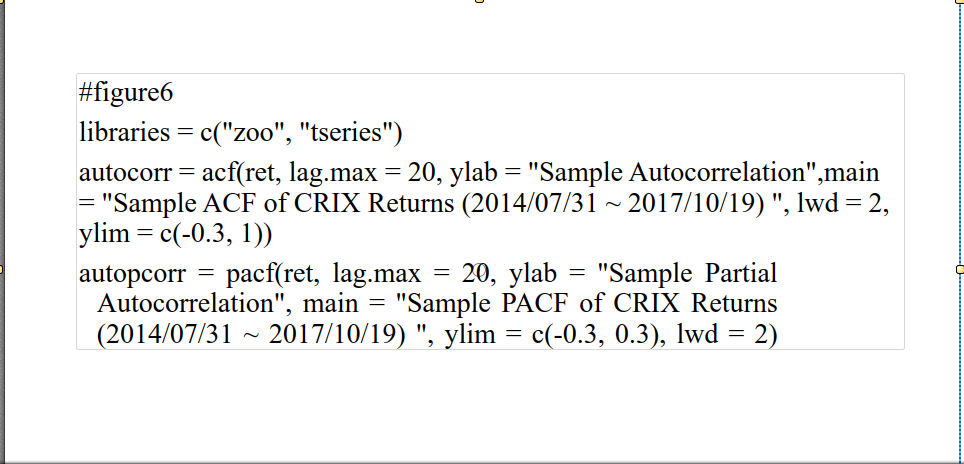
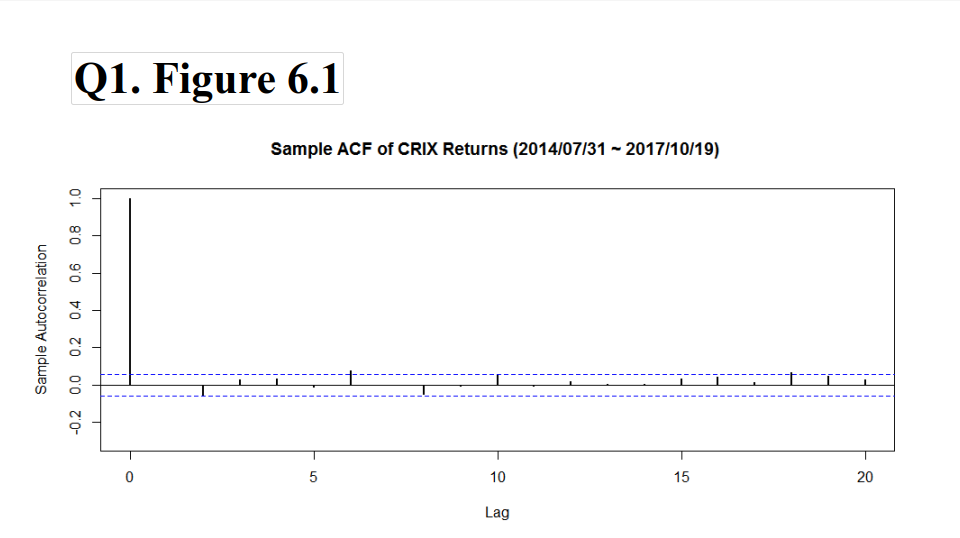
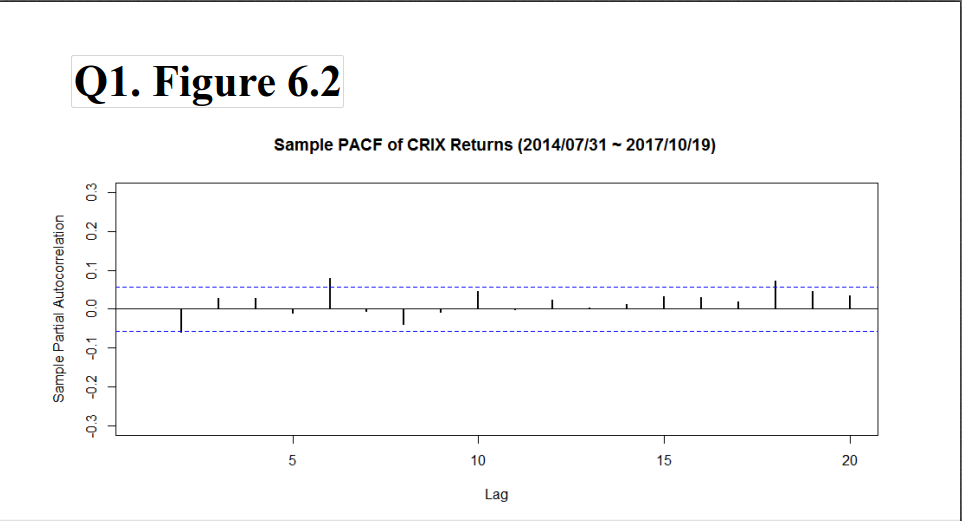
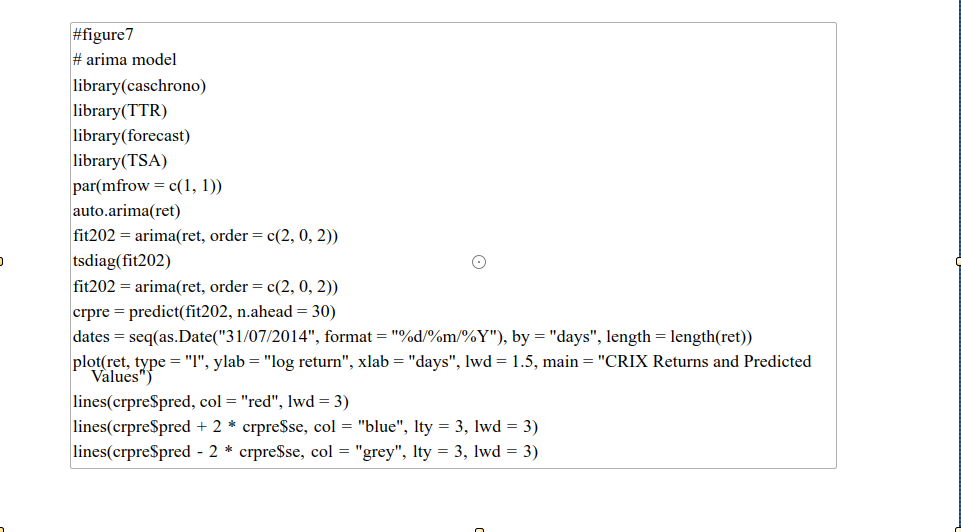
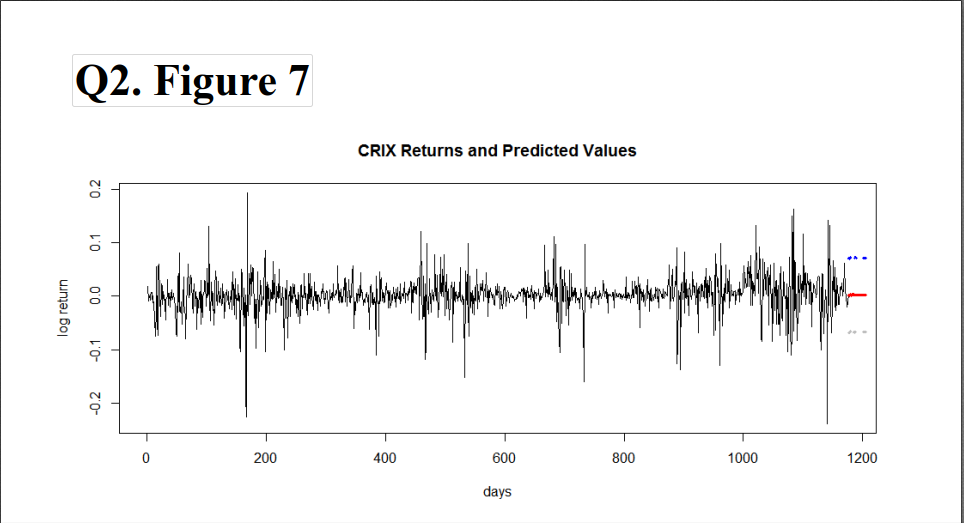
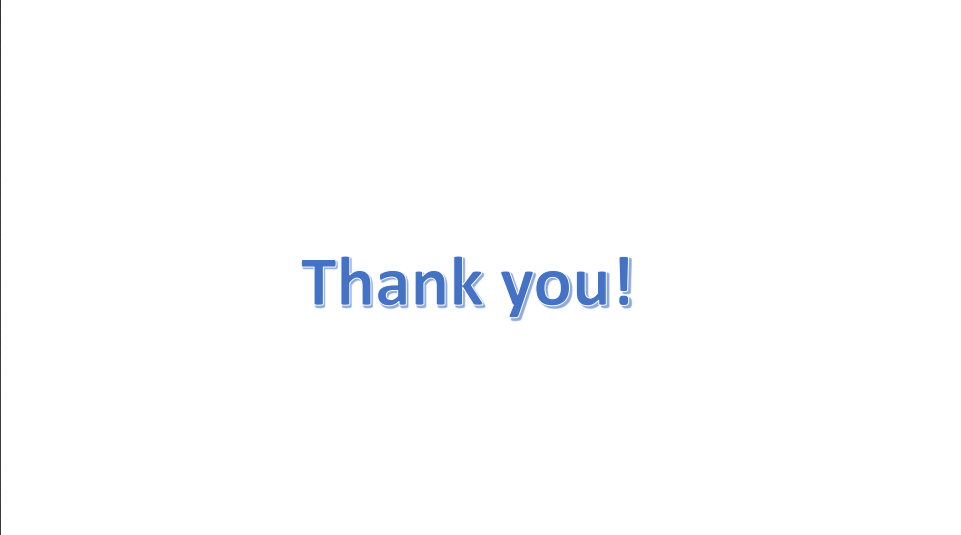
**Unit 4 HW**

**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.**

**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“.**

**4.3 Redo as many ﬁgures as you can.**

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

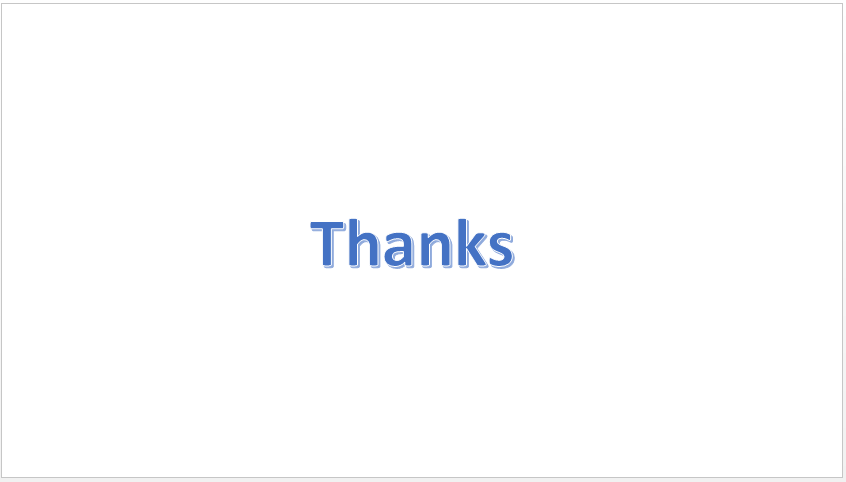
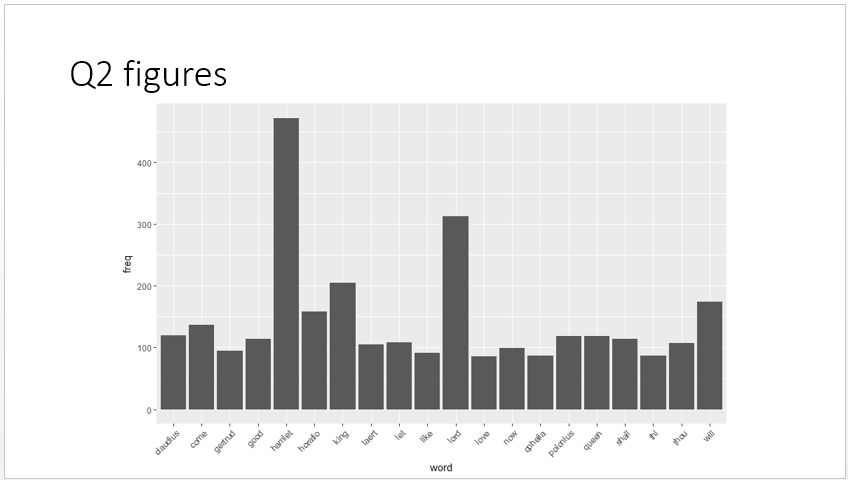
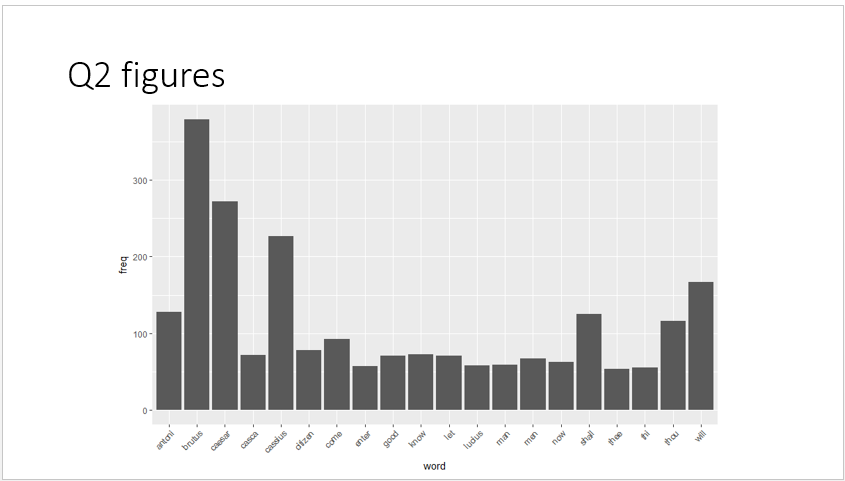
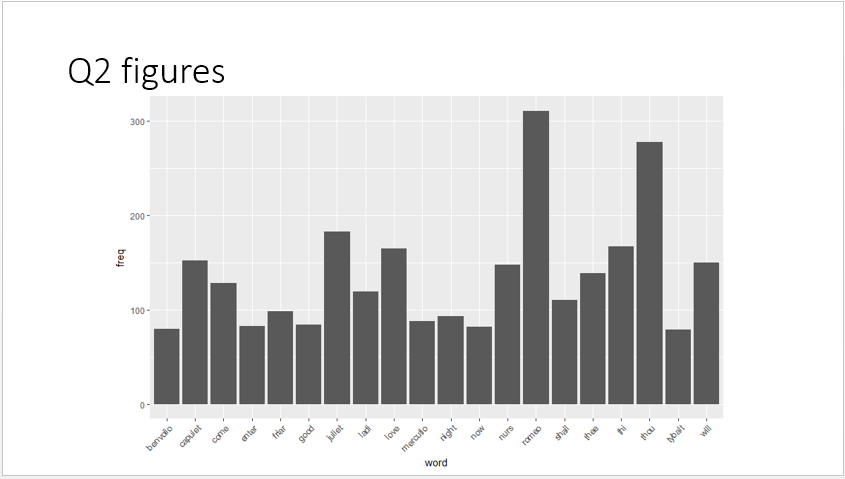
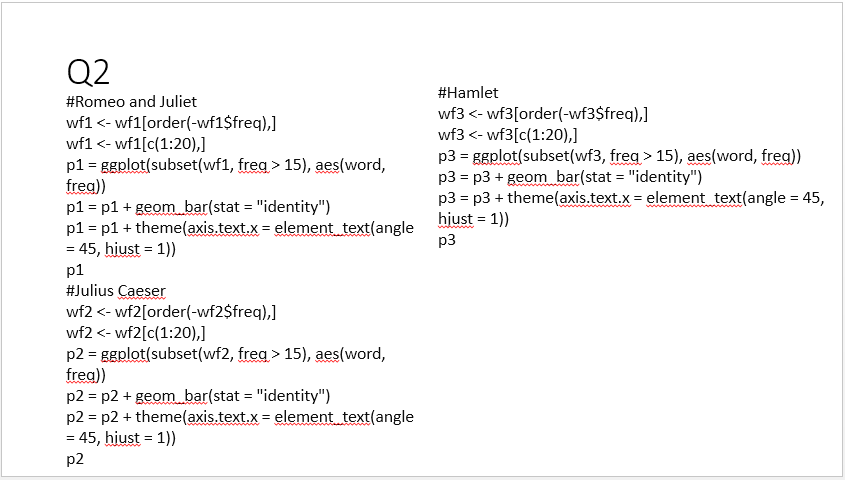
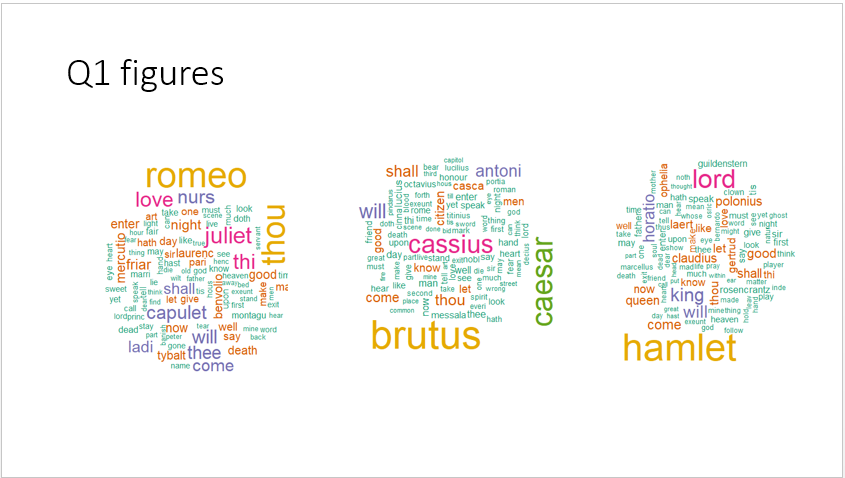
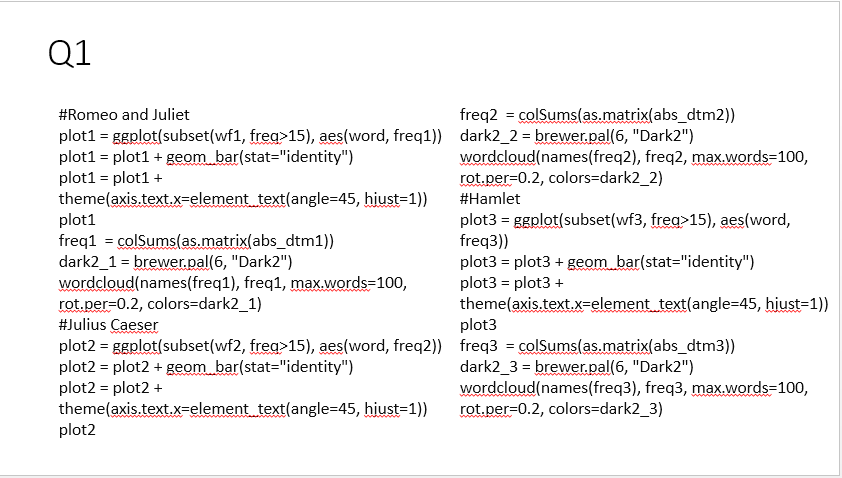
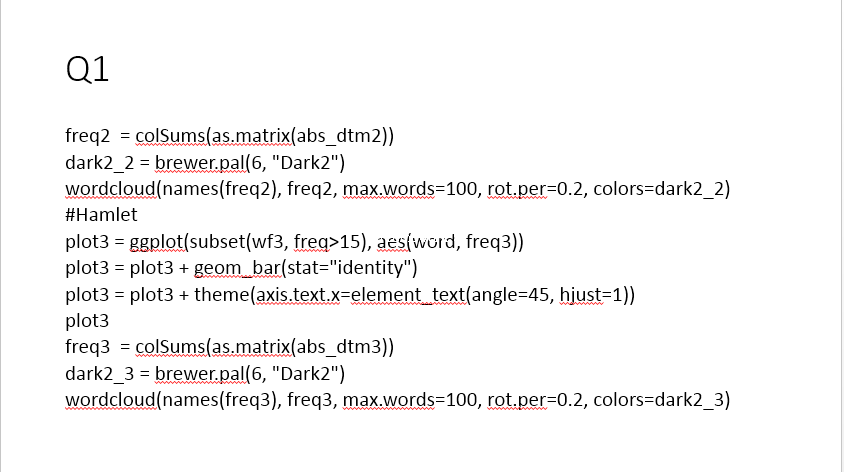
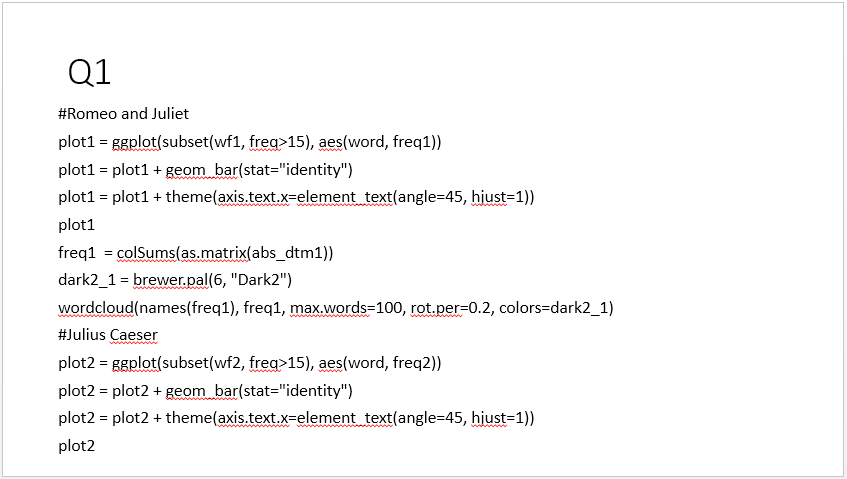
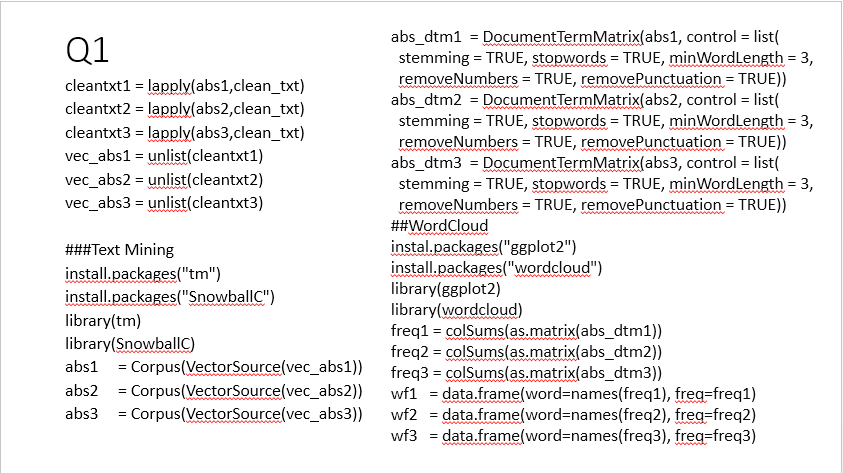
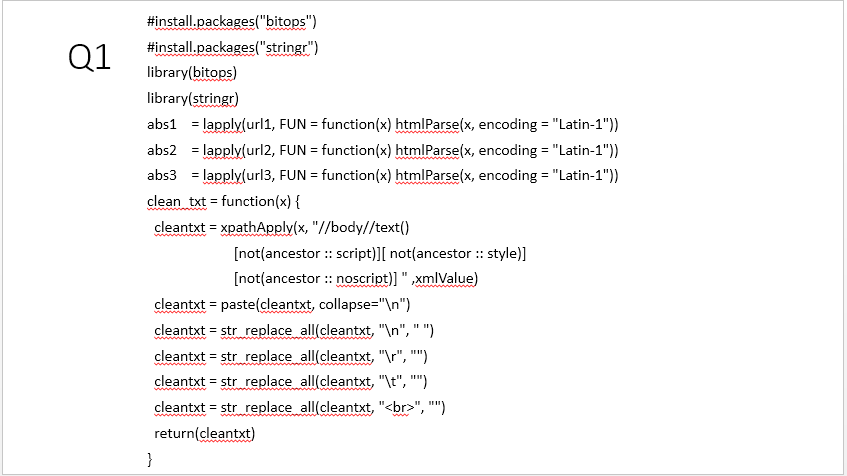
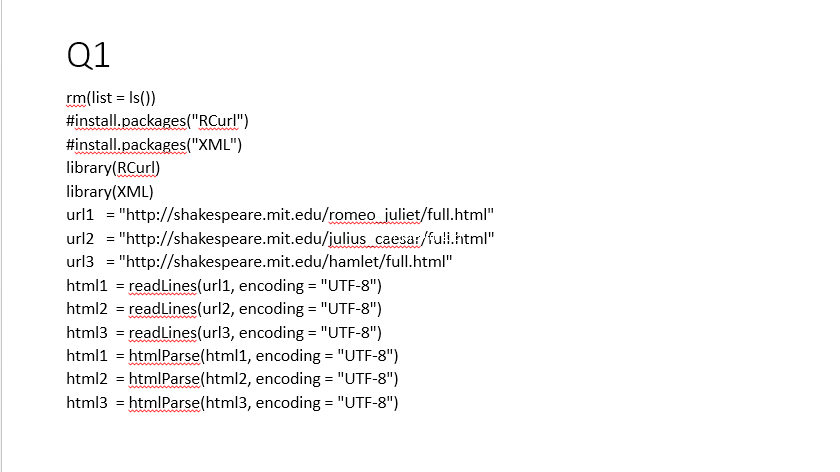
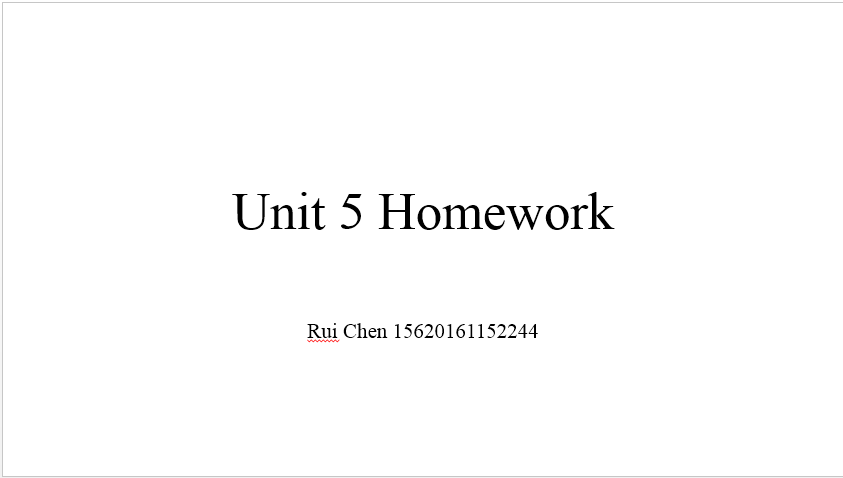
**Unit 5 HW**

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

**5.2 Calculate the histogram of words.**

**5.3 Map the Shakesspeare words into a dictionary to check its sentiment.**

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



**Final EXAM**

**Collect all HW s in one word file and leave it on GH for evaluation.**