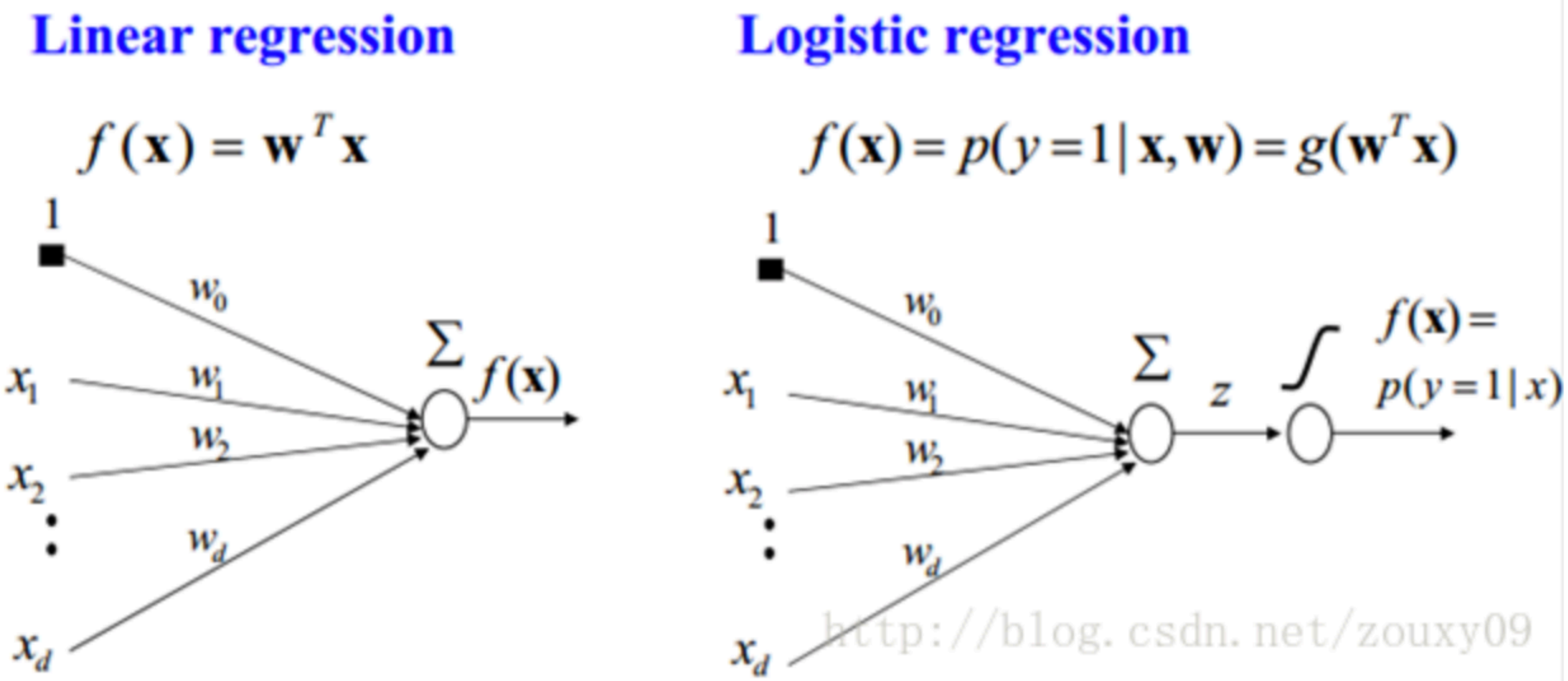
# HOMEWORK1

# LOGISTIC REGRESSION

Reporter: Zhen Qin

# DIFFERENCE




Logistic regression-binary dependent variables—that is, where the output can take only two values, "0" and "1", which represent outcomes such as pass/fail, win/lose, alive/dead or healthy/sick

# EXAMPLE

A group of 20 students spend between 0 and 6 hours studying for an exam. How does the number of hours spent studying affect the probability that the student will pass the exam?

| Hours | Pass |
|---|---|
| 0.5 | 0 |
| 0.75 | 0 |
| 1 | 0 |
| 1.25 | 0 |
| 1.5 | 0 |
| 1.75 | 0 |
| 1.75 | 1 |
| 2 | 0 |
| 2.25 | 1 |
| 2.5 | 0 |
| 2.75 | 1 |
| 3 | 0 |
| 3.25 | 1 |
| 3.5 | 0 |
| 4 | 1 |
| 4.25 | 1 |
| 4.5 | 1 |
| 4.75 | 1 |
| 5 | 1 |
| 5.5 | 1 |

1:pass
0:no pass

The logistic regression analysis gives the following output

| Coefficient | Std.Error | z-value | P-value (Wald) | |
|---|---|---|---|---|
| Intercept | −4.0777 | 1.761 | −2.316 | 0.0206 |
| Hours | 1.5046 | 0.6287 | 2.393 | 0.016 |

For a student who studies 4 hours, the estimated probability of passing the exam is 0.87:

Probability of passing exam=1/1+exp(−(1.5046 · 4−4.0777))=0.87

# HOMEWORK3

# (DIGITAL SIGNATURE ALGORITHM)

Reporter : Qin Zhen

Instructor : Wolfgang Härdle

# DEFINITION

- Digital signatures are essential to **verify the sender of a document's identity.**.
- The signature is generated by the use of **a private key** that known only to **the user.**.

# FUNCTION

- Digital signatures are used to detect unauthorized modifications to data.
- Using digitally signed document in proving to a third party
- Digital signature algorithms can be used any application that would need to assure the integrity and originality of data.

## Public key and private key generation

- Choose a prime number q,
- Choose another primer number p, s.t.p-1 mod q = 0
- Choose an integer g, s.t. 1 < g < p, g**q mod p = 1 and g = h**((p–1)/q) mod p. q is also called g's multiplicative order modulo p.
- Choose an integer, such that 0 < x < q.
- Compute y as g**x mod p.
- Package the public key as {p,q,g,y}.
- Package the private key as {p,q,g,x}.

# To verify a message signature,

- Generate the message digest h, using the same hash algorithm.
- Compute w, such that s*w mod q = 1. w is called the modular multiplicative inverse of s modulo q.
- Compute u1 = h*w mod q.
- Compute u2 = r*w mod q.
- Compute v = (((g**u1)*(y**u2)) mod p) mod q.
- If v == r, the digital signature is valid.

# Signature generation and verification

- To generate a message signature, the sender can follow these 1.Generate the message digest h, using a hash algorithm like SHA1.
- Generate a random number k, such that 0 < k < q.
- Compute r as (g**k mod p) mod q. If r = 0, select a different k.
- Compute i, such that k*i mod q = 1. i is called the modular multiplicative inverse of k modulo q.
- Compute s = i*(h+r*x) mod q. If s = 0, select a different k.
- Package the digital signature as {r,s}.

R-code:

```
>library(RJSONIO)
> letter<-LETTERS[1:10]
>country<-c("China","the US","the UK","Russia",

"Korea","Japan","Italy","Brazil","India","Germany")
> data<-data.frame(letter,country)
> da<-as.matrix(data)
>cat(toJSON(da))
```

```
[ {
 "letter": "A",
"country": "China"
},

{
 "letter": "B",
"country": "the US"
},

{
 "letter": "C",
"country": "the UK"
},

{
 "letter": "D",
"country": "Russia"
},

{
 "letter": "E",
"country": "Korea"
},

{
 "letter": "F",
"country": "Japan"
},

{
 "letter": "G",
"country": "Italy"
},

{
 "letter": "H",
"country": "Brazil"
},

{
 "letter": "I",
"country": "India"
},

{
 "letter": "J",
"country": "Germany"
} ]
```

# HOMEWORK4

Qin zhen

# HISTOGRAM

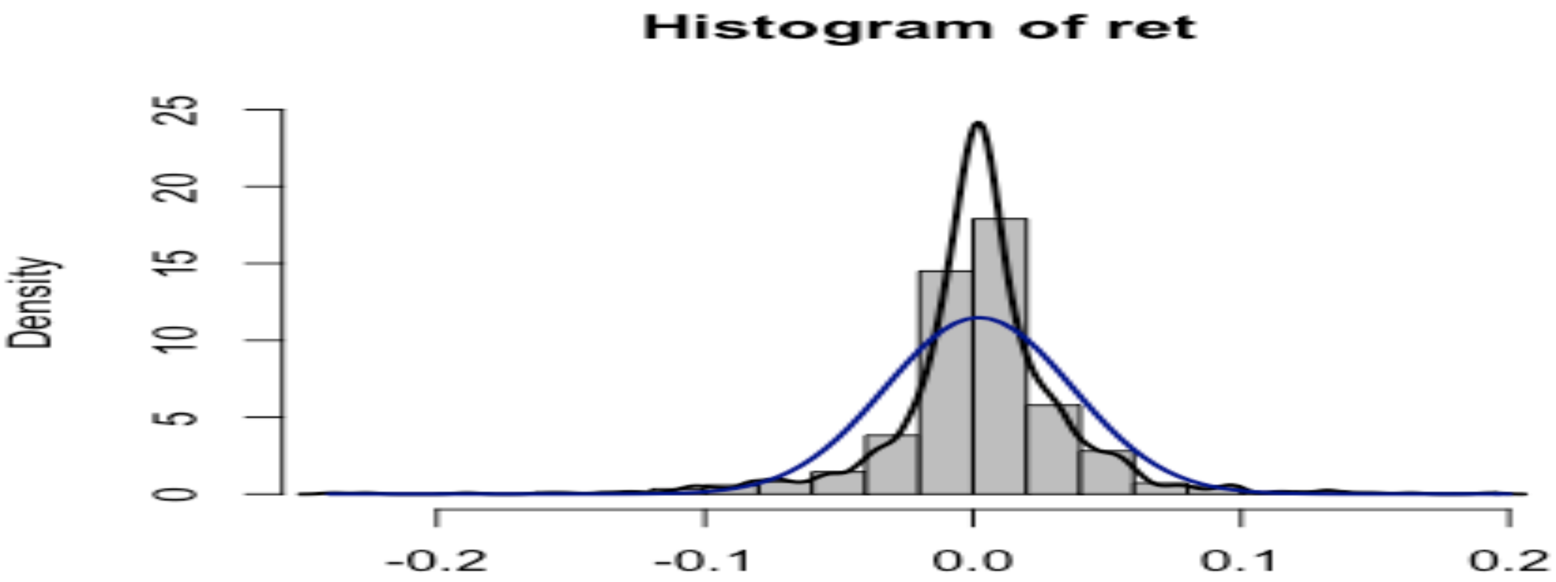
- hist(ret, col = "grey", breaks = 20, freq = FALSE, ylim = c(0, 25), xlab = NA)
- lines(density(ret), lwd = 2)
- mu = mean(ret)
- sigma = sd(ret)
- x = seq(-4, 4, length = 100)
- curve(dnorm(x, mean = mean(ret), sd = sd(ret)), add = TRUE, col = "darkblue", lwd = 2)

# FIG 3,4,5,6 OF THE „ECONOMETRICS OF CRIX“ PAPER.

- #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)
- 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))
- plot(ret)
- ts.plot(ret)

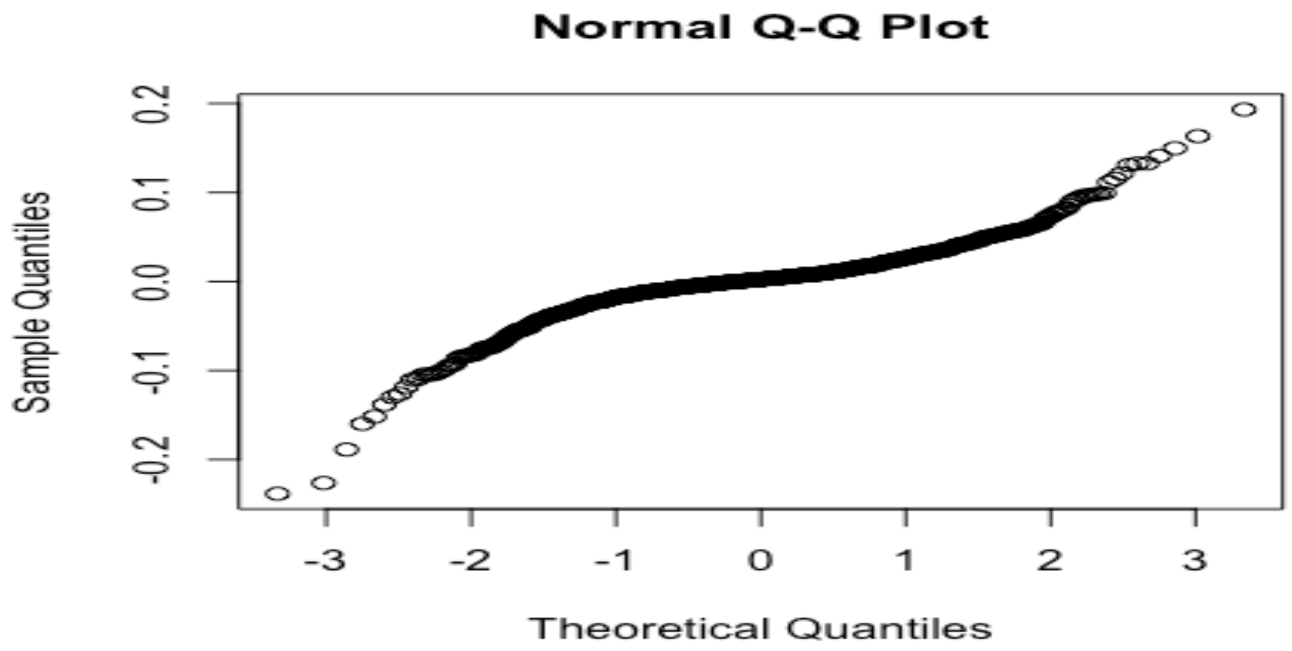
# HISTOGRAM



# QQ-PLOT

- # qq-plot
- qqnorm(ret)
- qqline(ret, col = "blue", lwd = 3)

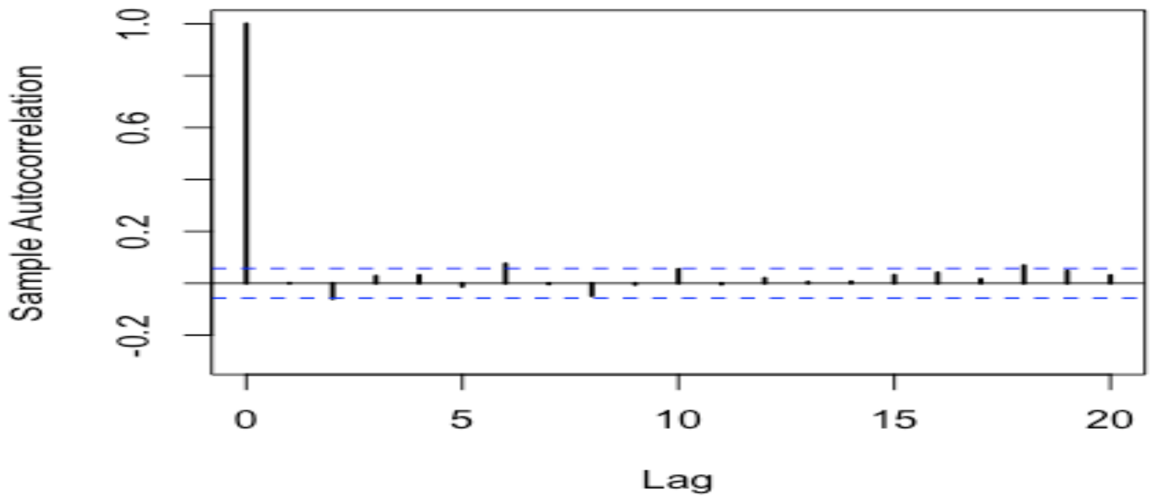
# QQ-PLOT



Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles
-0.2
-0.1
0.0
0.1
0.2
-3
-2
-1
0
1
2
3

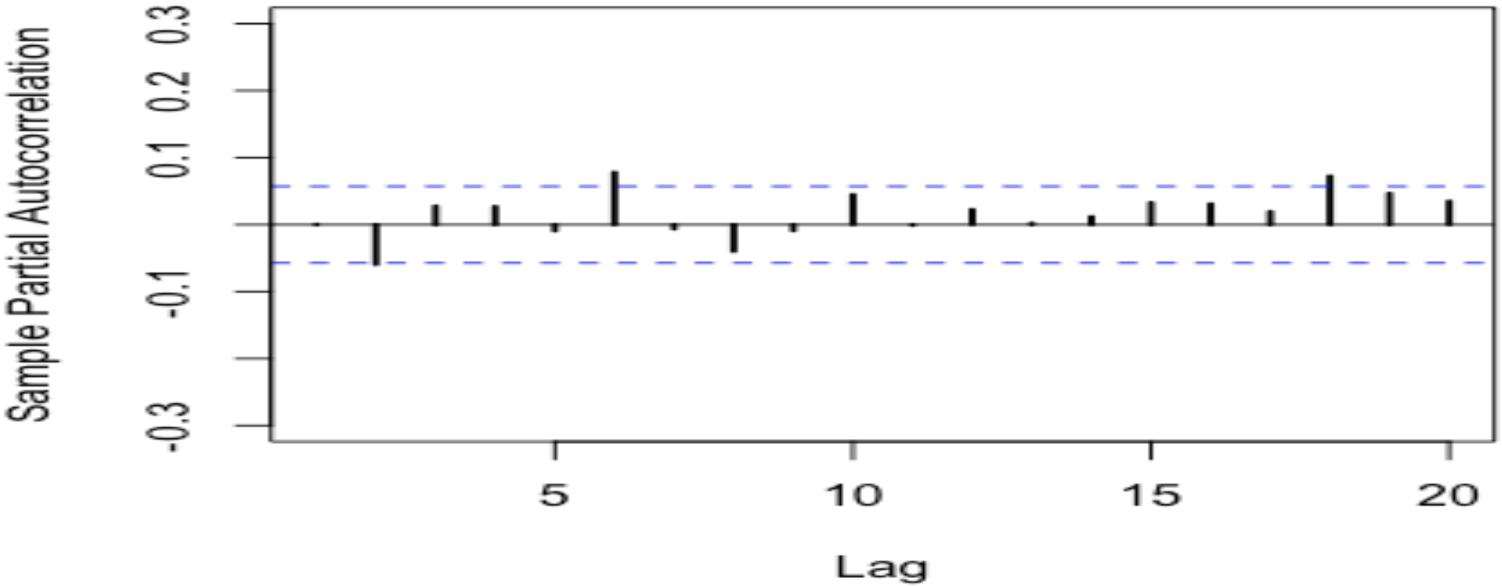
# SAMPLE AUTOCORRELATION

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



# SAMPLE PARTIAL AUTOCORRELATION

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

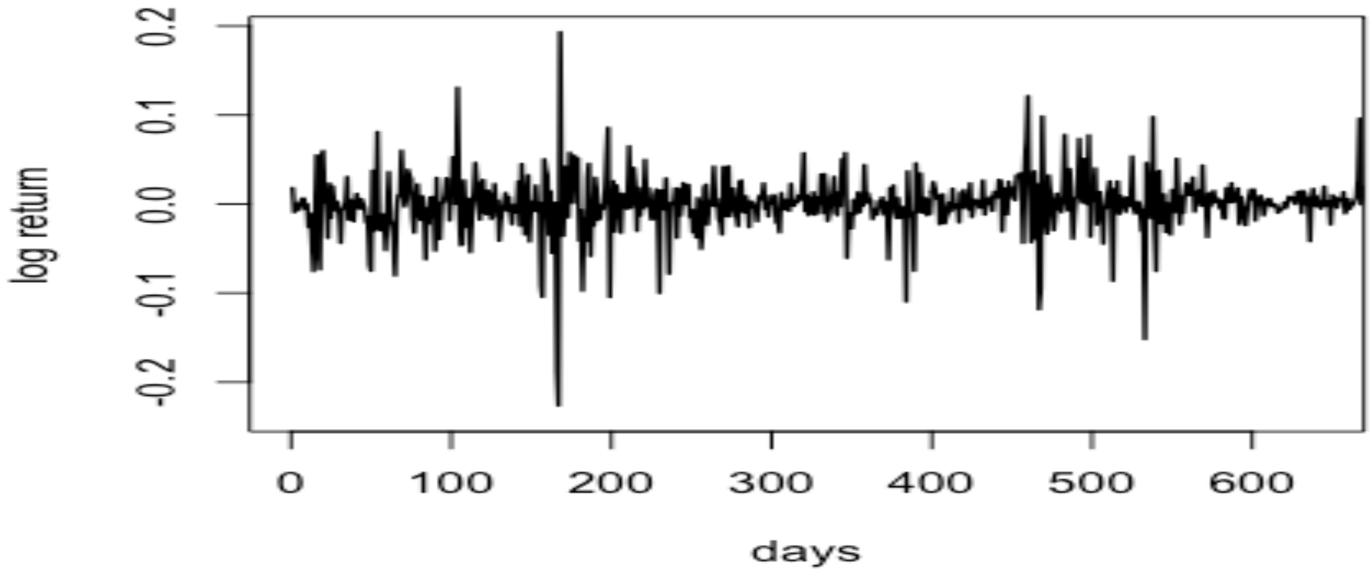


- select p and q order of ARIMA model
- fit4 = arima(ret, order = c(2, 0, 3))
- tsdiag(fit4)
- Box.test(fit4$residuals, lag = 1)

- fitr4 = arima(ret, order = c(2, 1, 3))
- tsdiag(fitr4)
- Box.test(fitr4$residuals, lag = 1)

- # to conclude, 202 is better than 213
- fit202 = arima(ret, order = c(2, 0, 2))
- tsdiag(fit202)
- tsdiag(fit4)
- tsdiag(fitr4)

- # 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


log return
0.2
0.1
0.0
-0.1
-0.2
0
100
200
300
400
500
600
days