STA hwk 2 (2)

2024-10-22

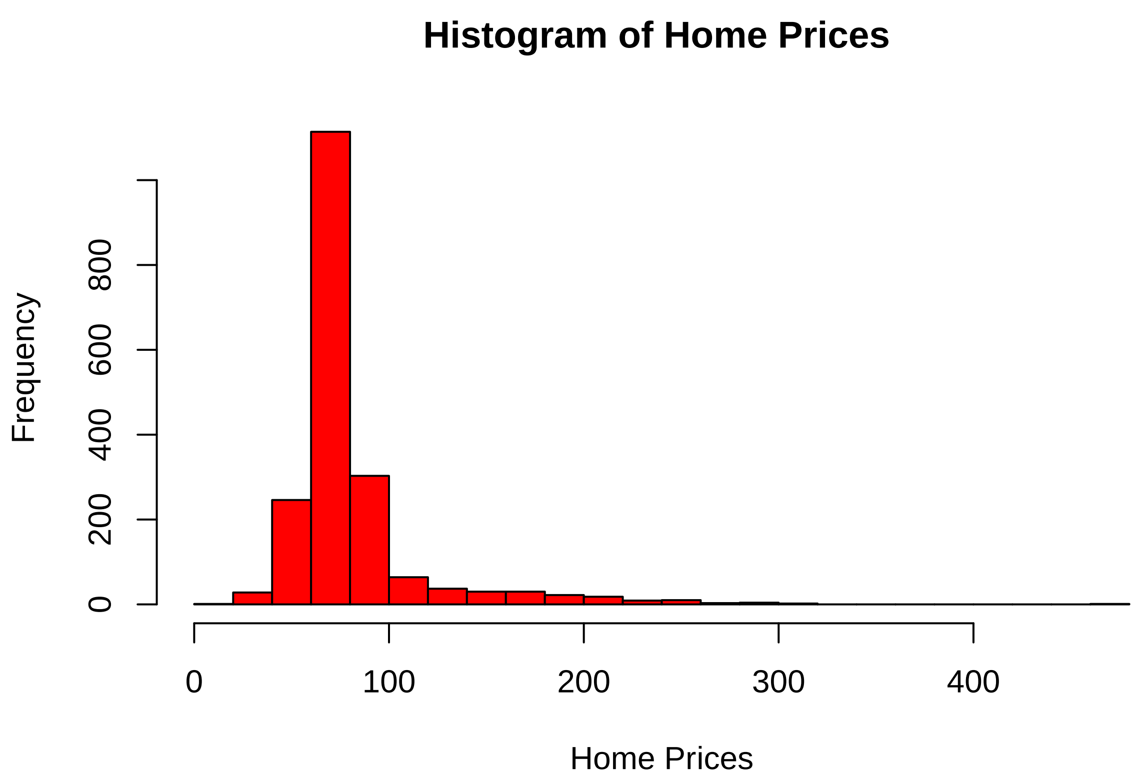
```r
library(car)
```

```
## Loading required package: carData
```

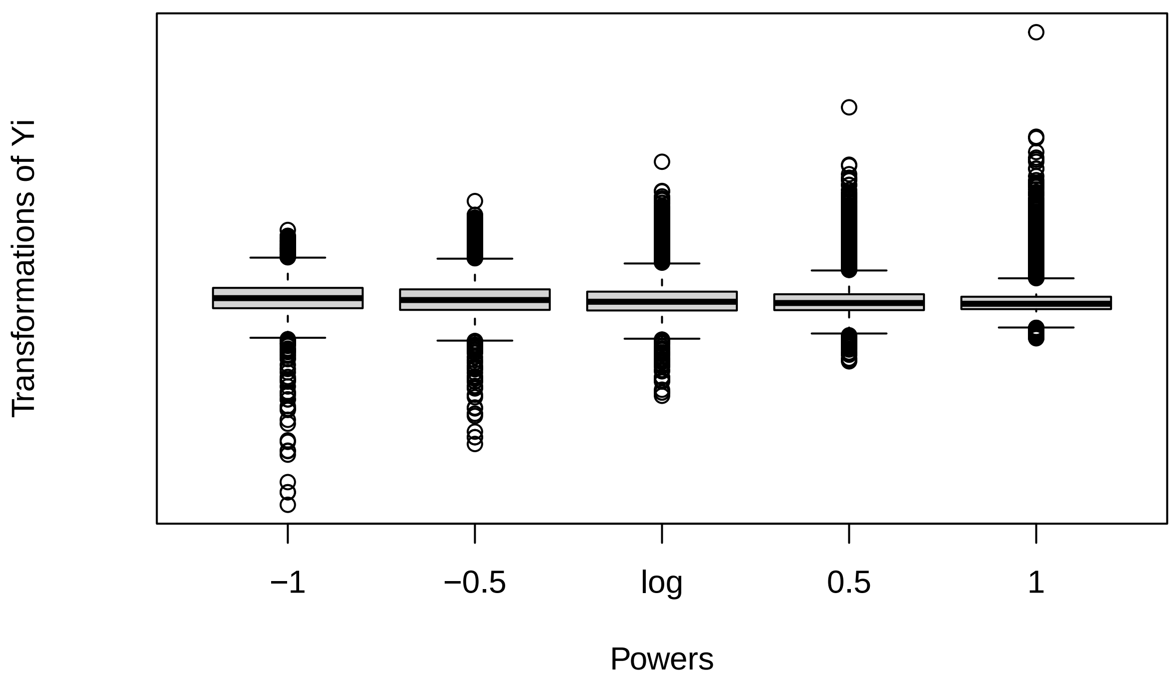
```r
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

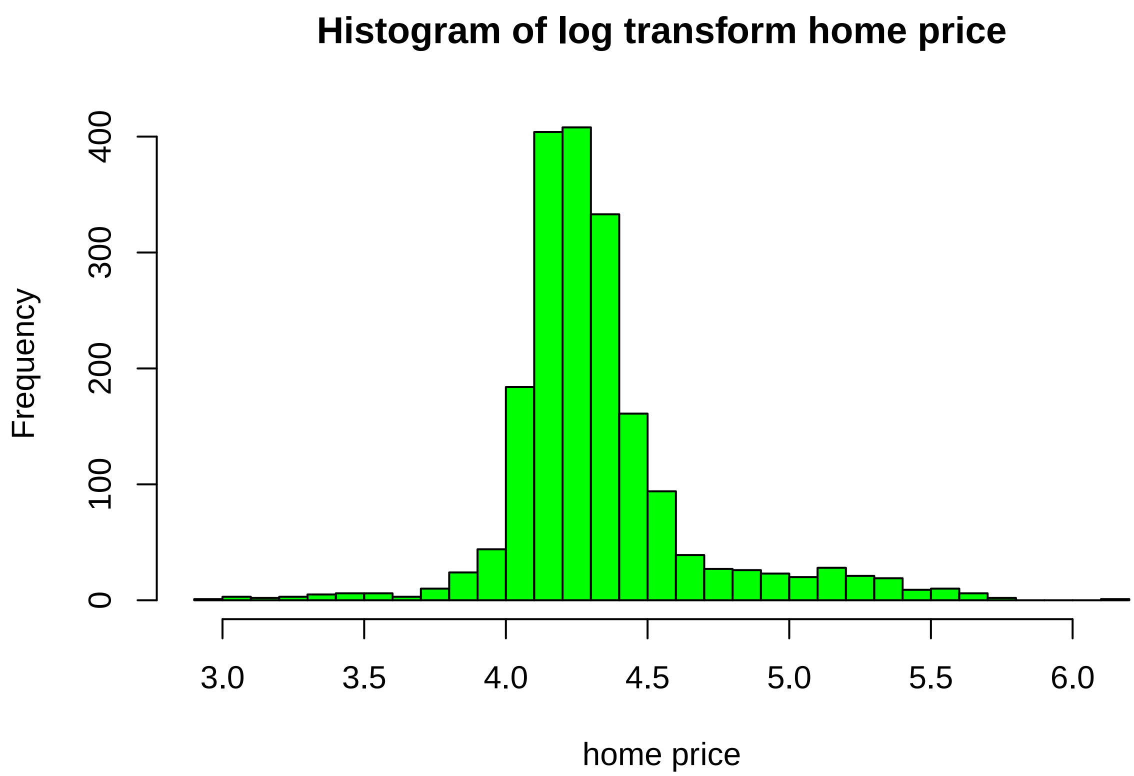
```r
library(moments)


#1) I would not hiring this TA
#The F-value formula that TA provided seems wrong, the numerator are the difference
#between two means, the denominator are the addition between two groups variance.
#thats actually more like a t-test, not a f-test. The correct F-test formula should be
#F = MSB(The mean square of between groups)/MSE(The mean square of error).
#TA's word seems confusing to me because the formula is inappropriate for the context.
#If explaining ANOVA, the TA should clarify that the F-statistic tests whether the
#variance between group means is significantly larger than the variance within groups.

#2a)
houstonrealesate = read.csv("/Users/shuhong/Desktop/2/houstonrealesate.csv")

hist(houstonrealesate$Yi, breaks = 30,
     main = "Histogram of Home Prices ",
     xlab = "Home Prices", col = "red")
```




```r
#b)
symbox(~Yi,data=houstonrealesate)
```



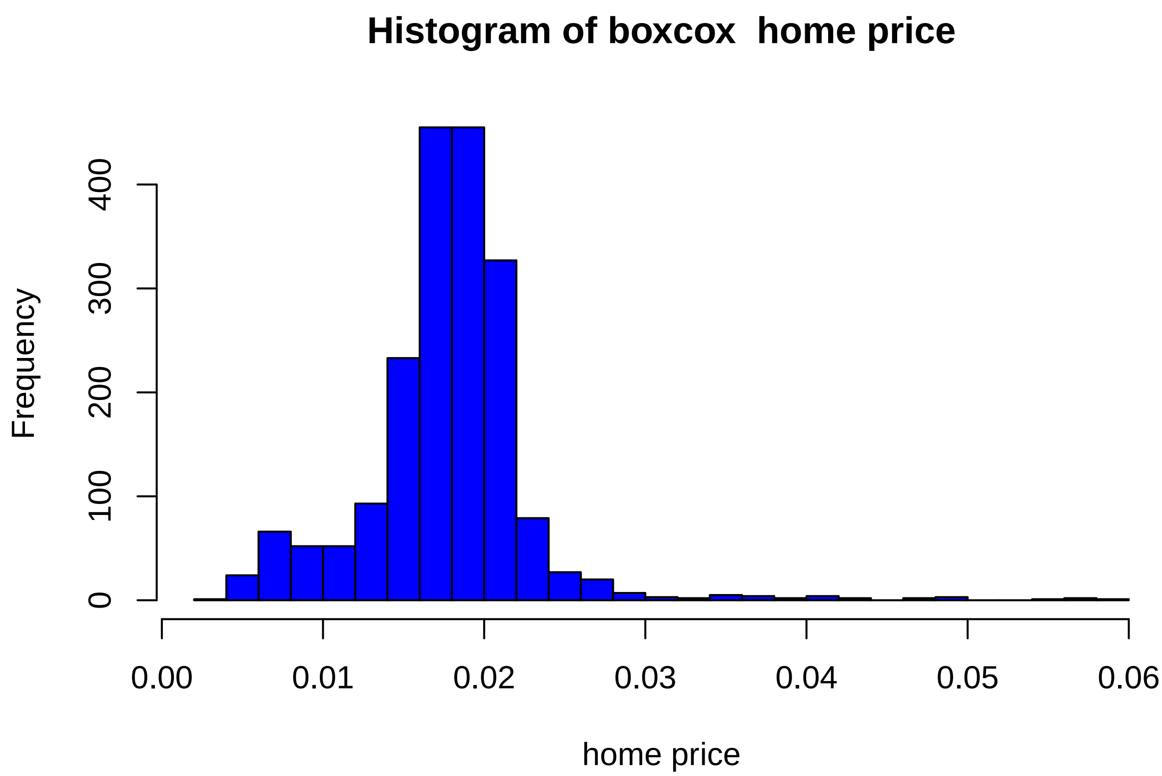

```r
#c) i will choose log transformed
log.Yi = log(houstonrealesate$Yi)

hist(log.Yi, breaks = 30, main = "Histogram of log transform home price",
     xlab = "home price", col = "green")
```




```r
#d)
library(forecast)
lambda=BoxCox.lambda(houstonrealesate$Yi)
lambda
```

```
## [1] -0.9423097
```

```r
boxcox.Yi=houstonrealesate$Yi^-0.9423097
boxcox.Yi=houstonrealesate$Yi^-0.9423097
hist(boxcox.Yi, breaks = 30, main = "Histogram of boxcox  home price",
     xlab = "home price", col = "blue")
```




```r
#e) I would say log-transform a bit better the best since it looks more symmetric than the boxcox one.
```

```r
#f)


skewness.log = skewness(log.Yi)
skewness.boxcox = skewness(boxcox.Yi)

skewness.log
```

```
## [1] 1.274064
```

```r
skewness.boxcox
```

```
## [1] 1.608676
```

```r
#since the skewness index of log is smaller than boxcox, log transformed histogram is better.
```

```r
#g)
shapiro.test(houstonrealesate$Yi)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  houstonrealesate$Yi
## W = 0.64268, p-value < 2.2e-16
```

```r
shapiro.test(log.Yi)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  log.Yi
```

```
## W = 0.85284, p-value < 2.2e-16
```

```r
shapiro.test(boxcox.Yi)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  boxcox.Yi
## W = 0.84442, p-value < 2.2e-16
```

```r
#h)
model.Yi = lm(Yi ~ x1i + x2i, data = houstonrealesate)
summary(model.Yi)
```

```
##
## Call:
## lm(formula = Yi ~ x1i + x2i, data = houstonrealesate)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -59.54 -16.64  -9.02   1.94 385.75
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   89.469      1.138  78.603   <2e-16 ***
## x1i            5.867      3.023   1.941   0.0524 .
## x2i          -90.896      5.915 -15.368   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 34.27 on 1919 degrees of freedom
## Multiple R-squared:  0.1209, Adjusted R-squared:   0.12
## F-statistic:   132 on 2 and 1919 DF,  p-value: < 2.2e-16
```

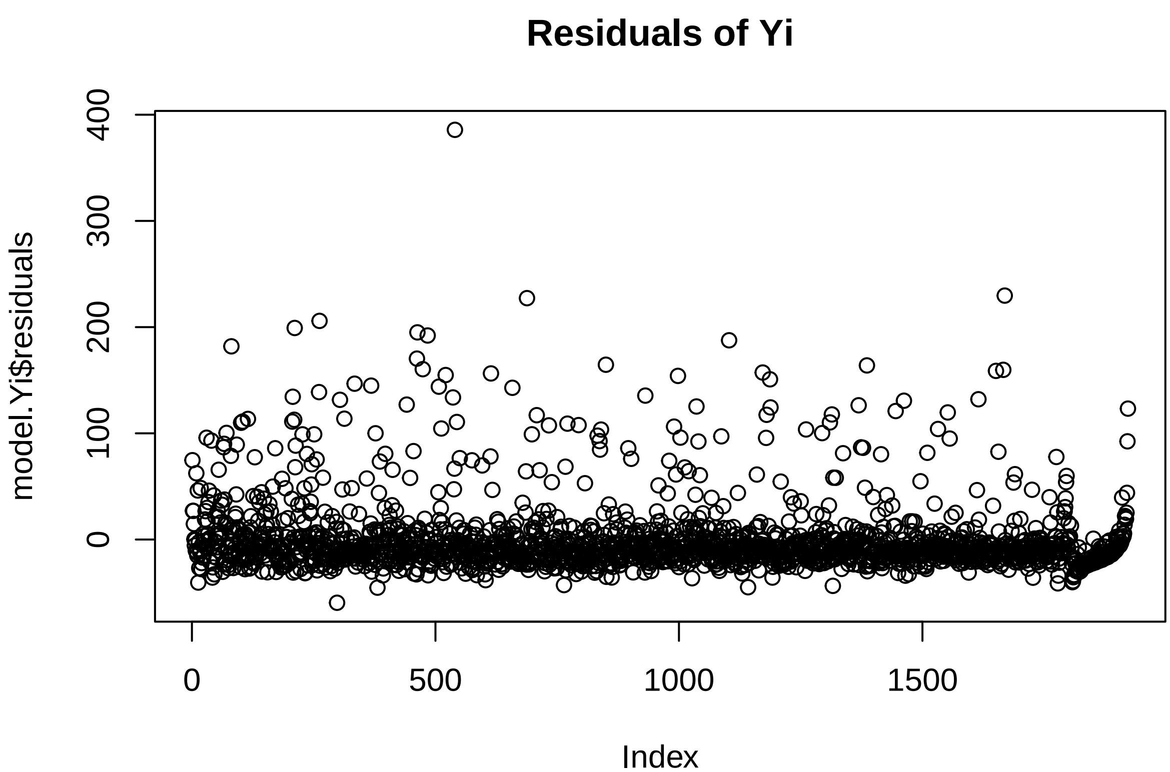
```r
model.logYi = lm(log.Yi ~ x1i + x2i, data = houstonrealesate)
summary(model.logYi)
```

```
##
## Call:
## lm(formula = log.Yi ~ x1i + x2i, data = houstonrealesate)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.03501 -0.16497 -0.06736  0.07206  1.72991
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.433868   0.009705 456.851  < 2e-16 ***
## x1i          0.081786   0.025773   3.173  0.00153 **
## x2i         -1.136640   0.050433 -22.538  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2922 on 1919 degrees of freedom
## Multiple R-squared:   0.23,  Adjusted R-squared:  0.2292
## F-statistic: 286.5 on 2 and 1919 DF,  p-value: < 2.2e-16
```

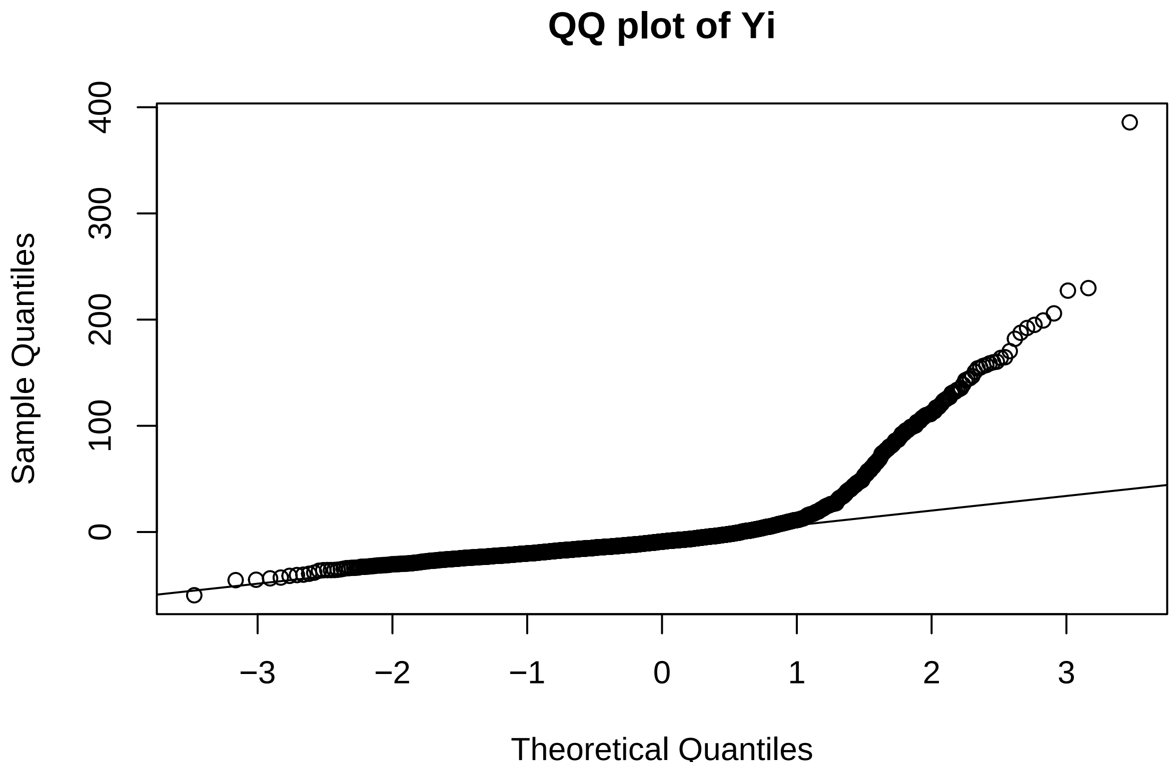
```r
model.boxcoxYi = lm(boxcox.Yi ~ x1i + x2i, data = houstonrealesate)
summary(model.boxcoxYi)
```

```
##
## Call:
## lm(formula = boxcox.Yi ~ x1i + x2i, data = houstonrealesate)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0133721 -0.0018268  0.0004361  0.0022037  0.0287460
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.0156602  0.0001397 112.089  < 2e-16 ***
## x1i         -0.0012192  0.0003710  -3.286  0.00103 **
## x2i          0.0213607  0.0007260  29.422  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.004207 on 1919 degrees of freedom
## Multiple R-squared:  0.3332, Adjusted R-squared:  0.3325
## F-statistic: 479.5 on 2 and 1919 DF,  p-value: < 2.2e-16
```

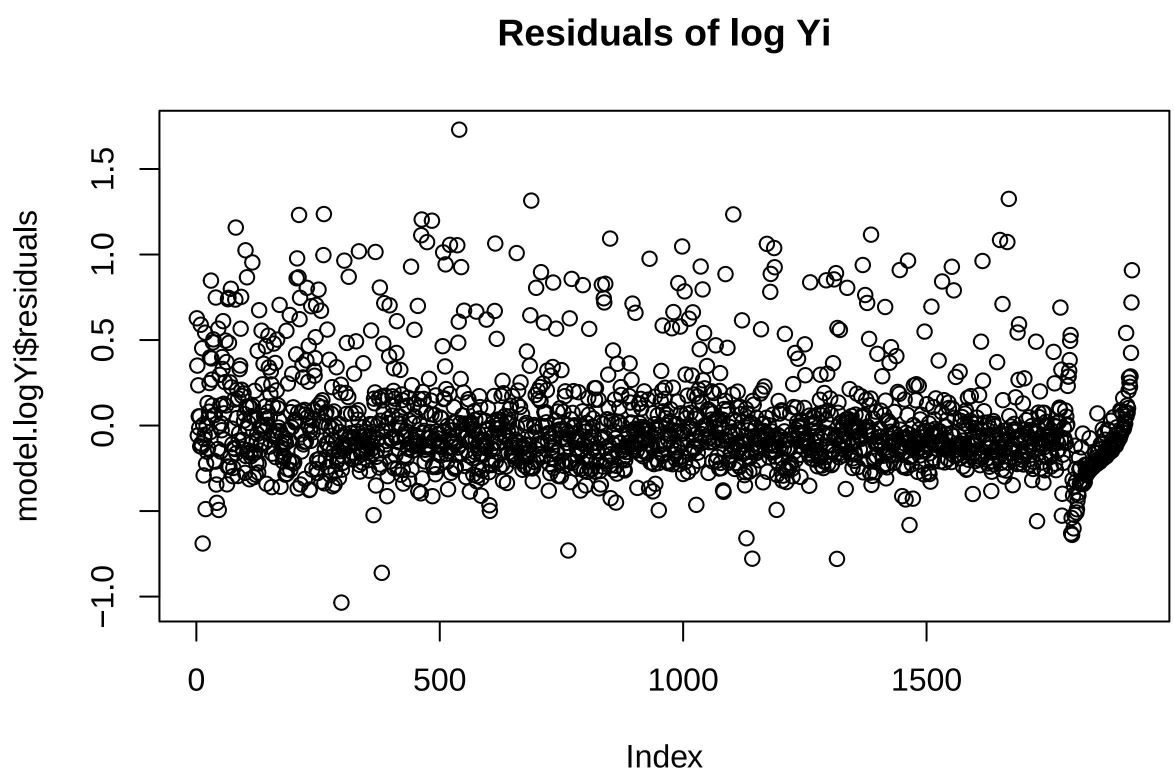
```r
plot(model.Yi$residuals, main = "Residuals of Yi")
```



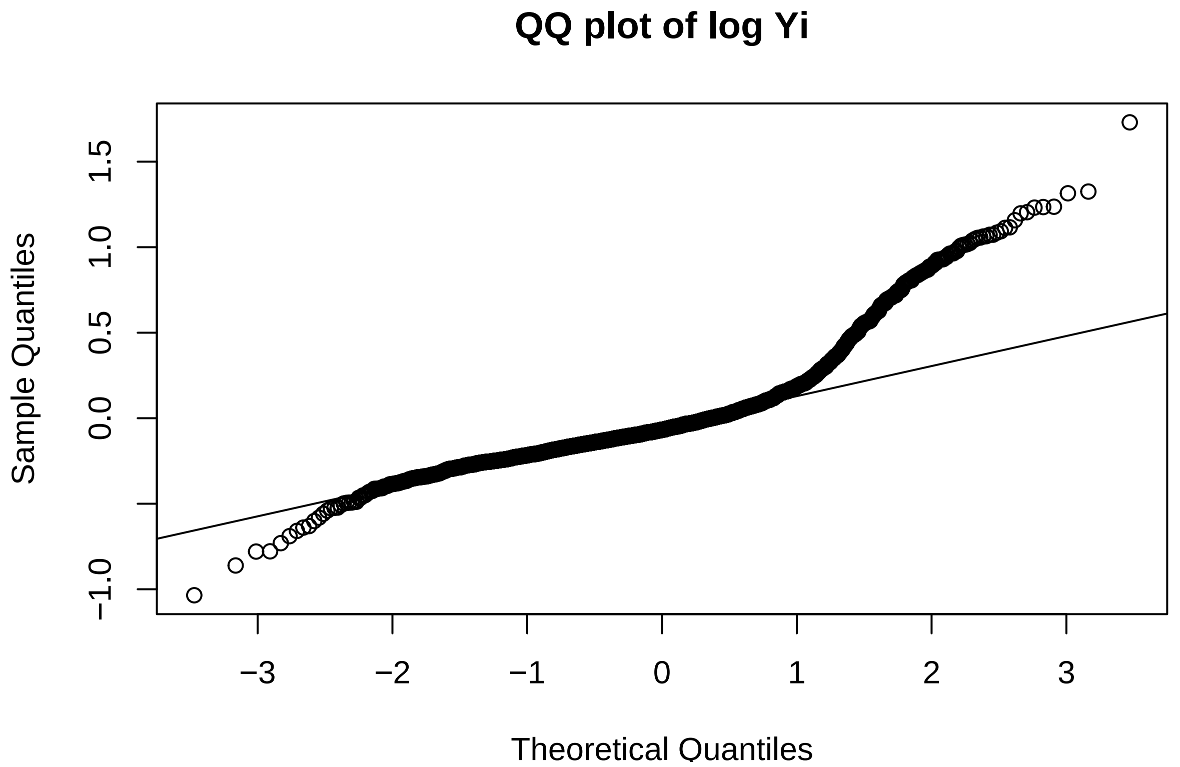

```r
qqnorm(model.Yi$residuals, main = "QQ plot of Yi")
qqline(model.Yi$residuals)
```



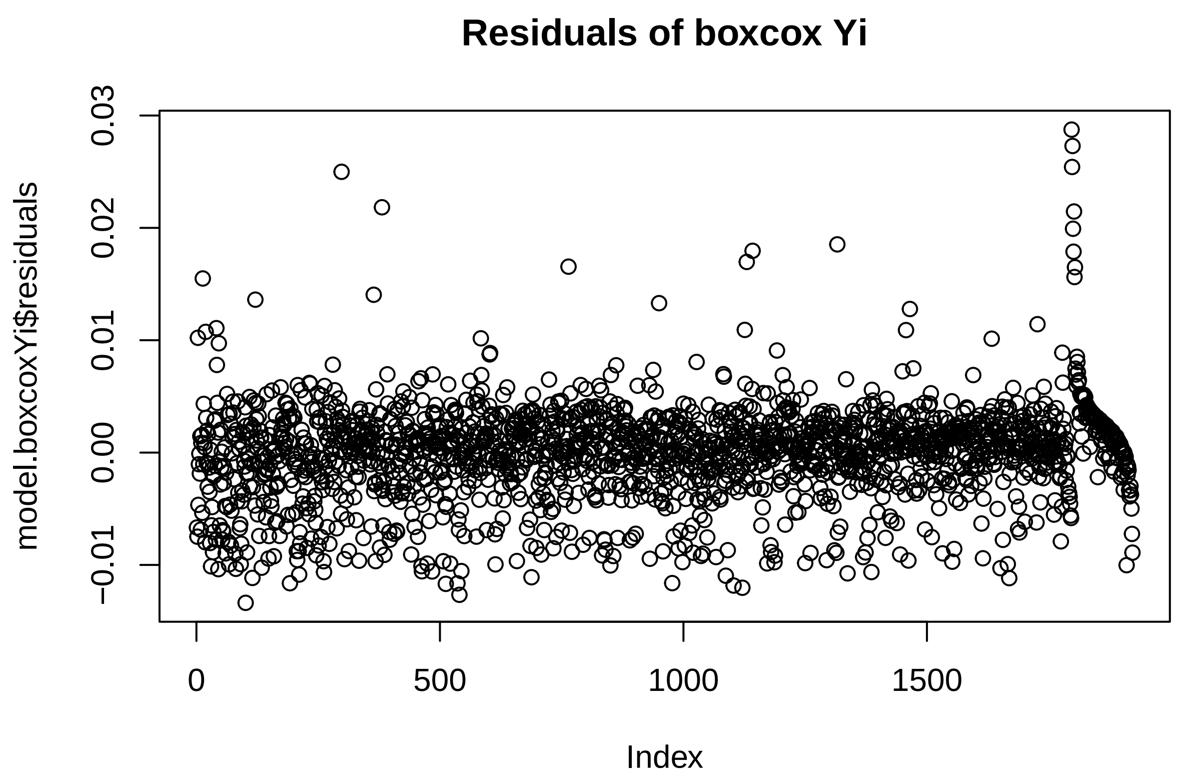

```r
plot(model.logYi$residuals, main = "Residuals of log Yi")
```



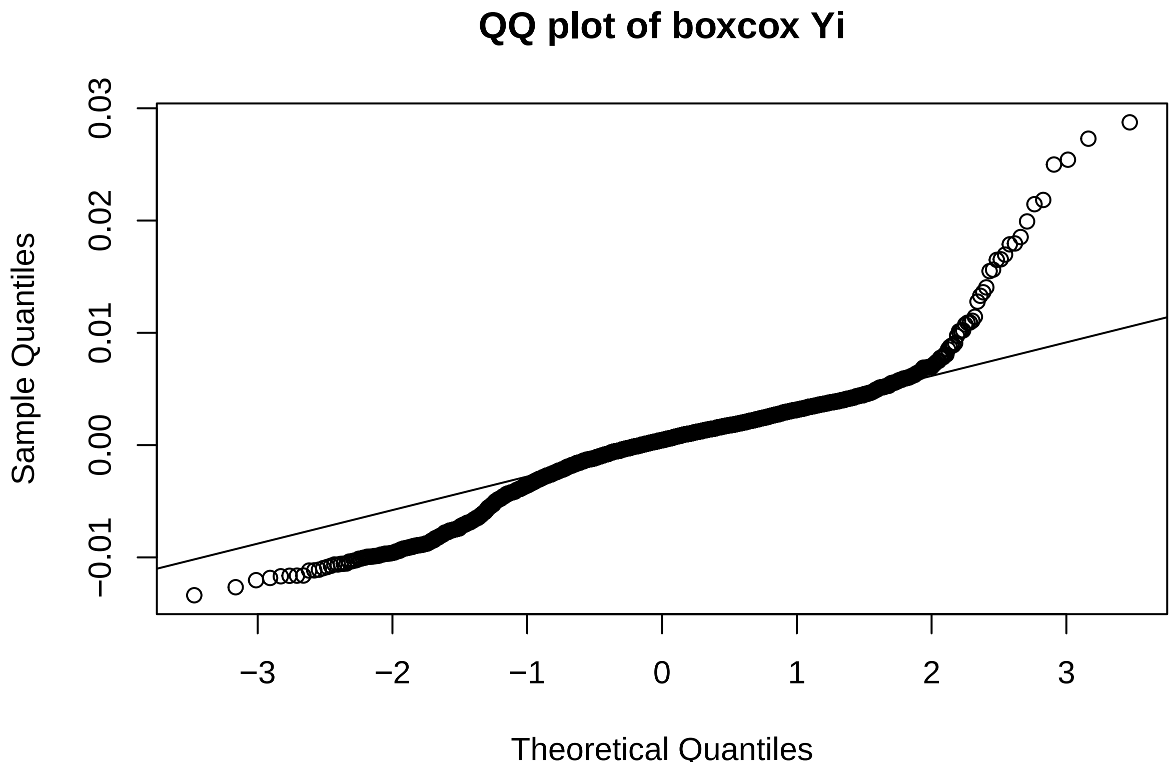

```r
qqnorm(model.logYi$residuals, main = "QQ plot of log Yi")
qqline(model.logYi$residuals)
```




```r
plot(model.boxcoxYi$residuals, main = "Residuals of boxcox Yi")
```




```r
qqnorm(model.boxcoxYi$residuals, main = "QQ plot of boxcox Yi")
qqline(model.boxcoxYi$residuals)
```




```
#I think Box-Cox transformed model's plots (both residual and QQ plots) are likely to be the best.
#Because the transformation seems right for skewness, resulting in more normally distributed residuals
#and a more appropriate model fit. Thus, Box-Cox transformation should be the best model for h)
#as it addresses non-linearity better than the original or log-transformed one.
```