CAPM Model for RIL vs NIFTY 50

Harsh Mittal

2024-07-06

```
library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.2.3
```

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

```
start_date = Sys.Date() - 90
end_date = Sys.Date()
rel = get.hist.quote(instrument = "RELIANCE.NS",
                     start=start_date, end = end_date,
                     quote = "AdjClose", provider = "yahoo")
```

```
## time series ends   2024-07-12
```

```
nifty = get.hist.quote(instrument = "^NSEI",
                     start=start_date, end = end_date,
                     quote = "AdjClose", provider = "yahoo")
```

```
## time series ends   2024-07-12
```

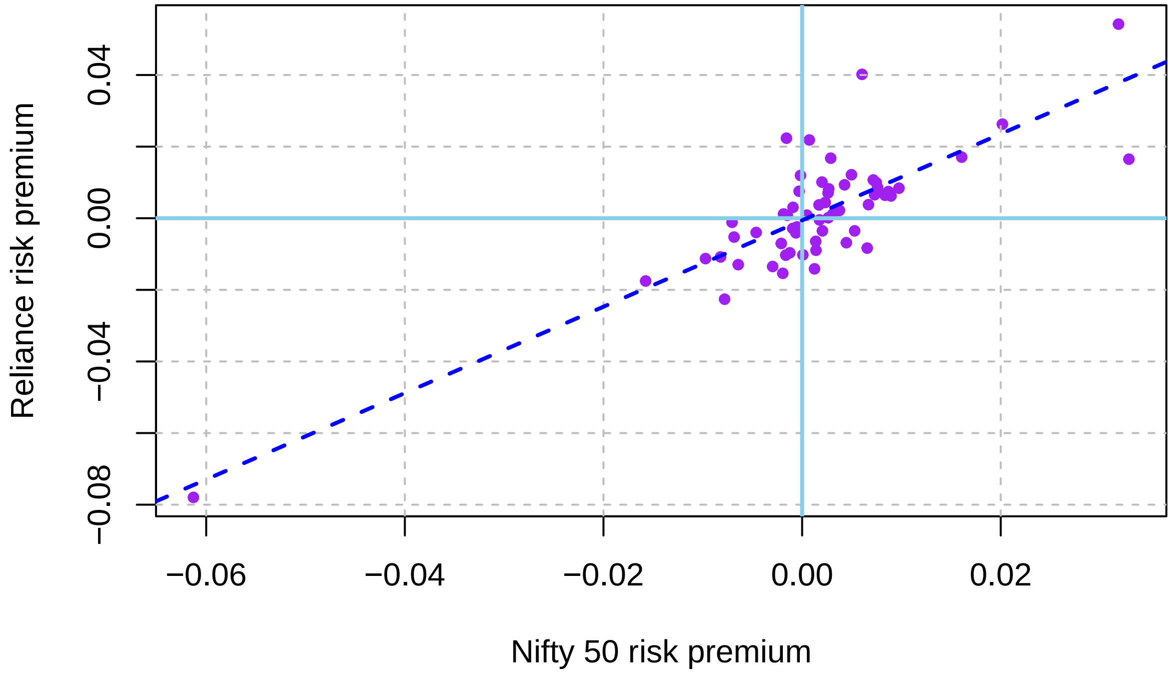
```
data = merge(nifty,rel)
rt = diff(log(data))
head(rt*100)
```

```
##            Adjusted.nifty Adjusted.rel
## 2024-04-18     -0.6888920  -0.09727048
## 2024-04-19      0.6848266   0.39530791
## 2024-04-22      0.8515605   0.65932836
## 2024-04-23      0.1413714  -1.39667482
## 2024-04-24      0.1536747  -0.62896941
## 2024-04-25      0.7468967   0.67350231
```

```
risk_free_rate = 0.06/365
```

risk premium

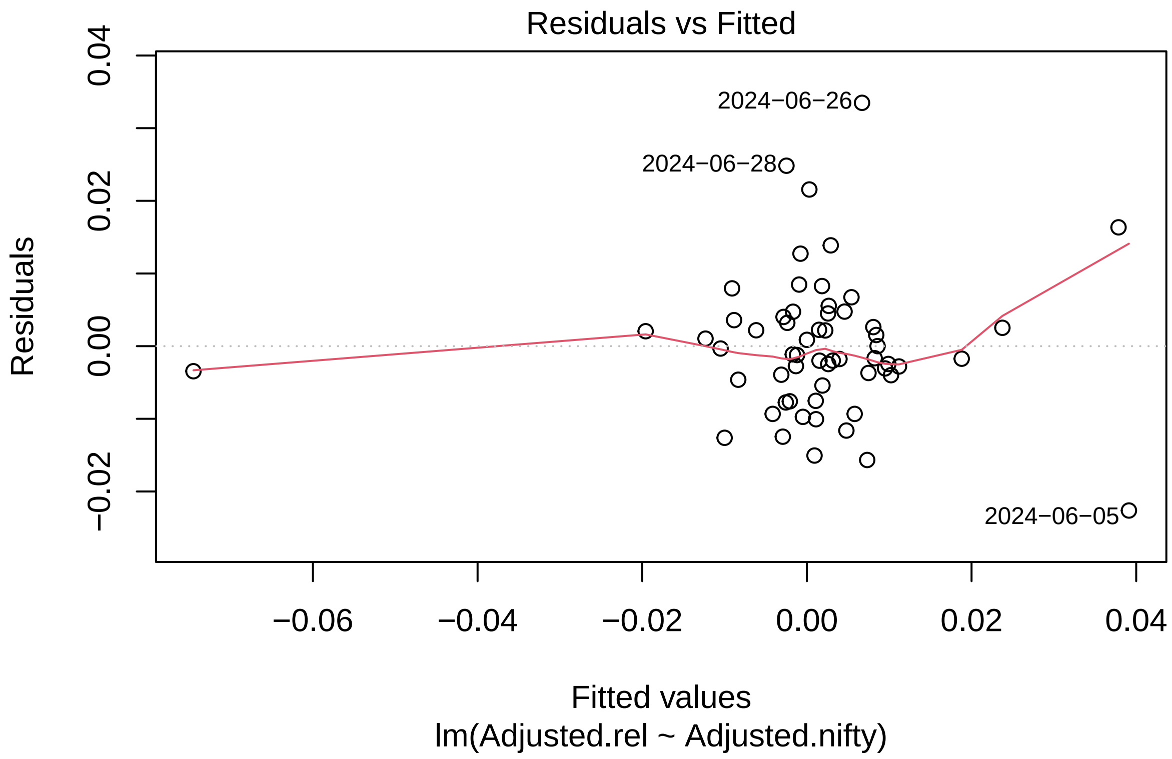
```
rt = rt-risk_free_rate


plot(rt$Adjusted.nifty, rt$Adjusted.rel, pch = 20, col = 'purple',
     xlab = 'Nifty 50 risk premium', ylab = 'Reliance risk premium')
grid(col='grey', lty=2)
abline(h=0, col = 'skyblue', lwd = 2)
abline(v=0, col = 'skyblue', lwd = 2)
abline(lm(Adjusted.rel~Adjusted.nifty, data = rt),
       col='blue', lwd = 2, lty = 2)
```

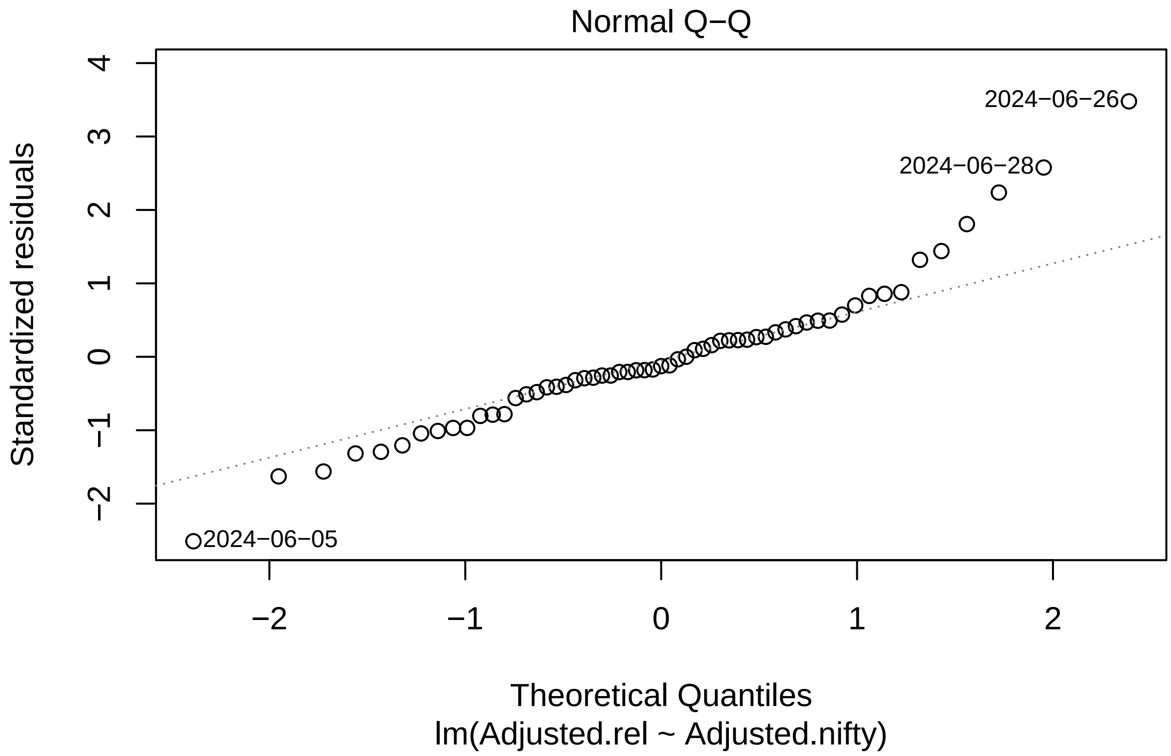


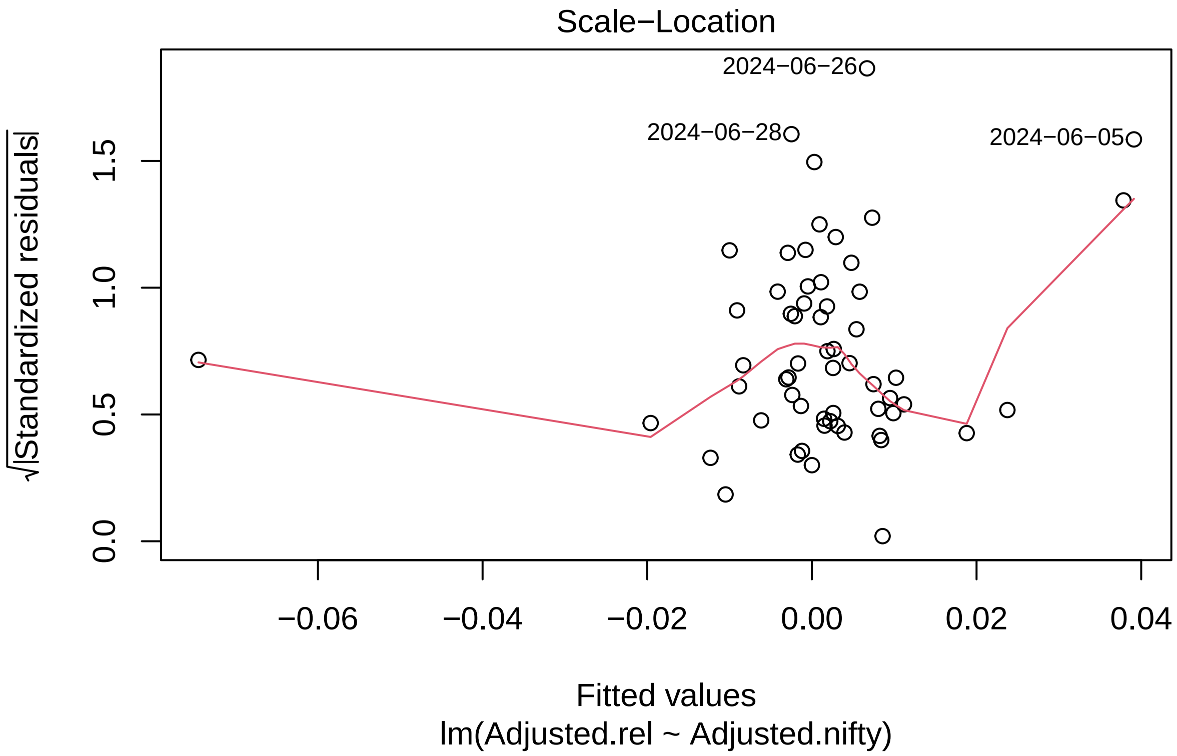
```
CAPM = lm(Adjusted.rel~Adjusted.nifty, data = rt)
summary(CAPM)
```

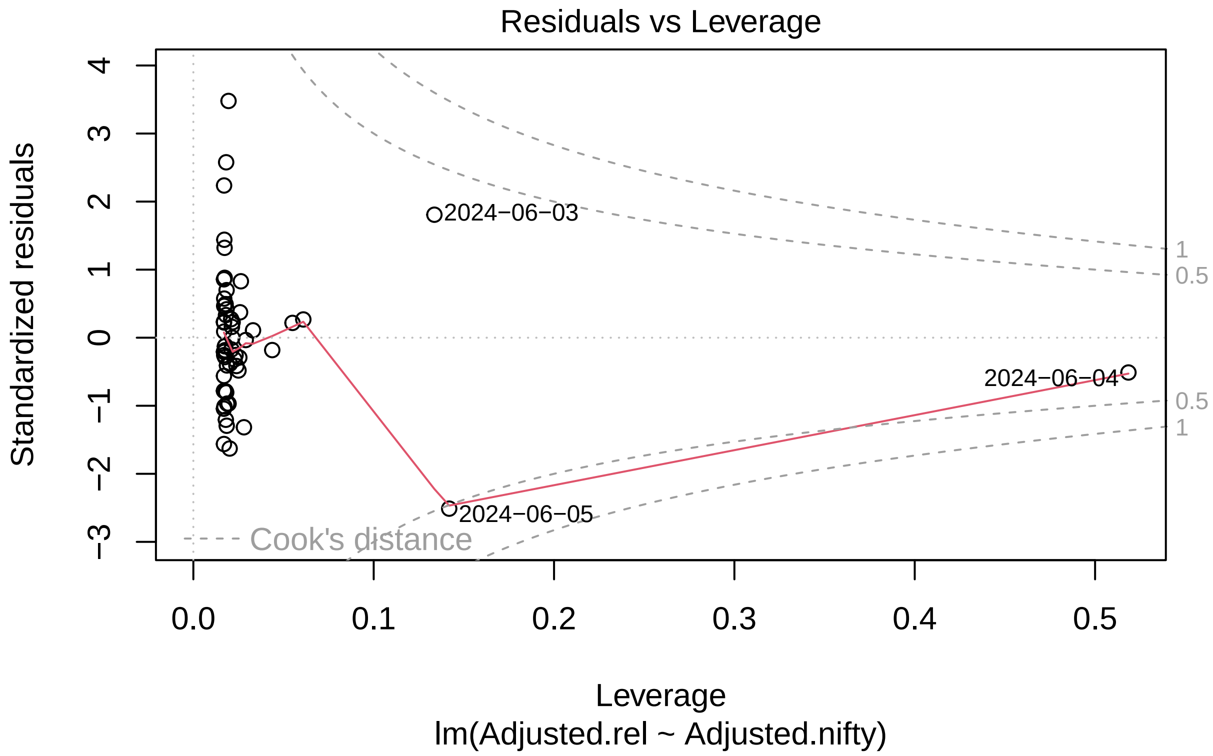
```
##
## Call:
## lm(formula = Adjusted.rel ~ Adjusted.nifty, data = rt)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.022619 -0.004309 -0.001223  0.003800  0.033487
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.0005809  0.0012769  -0.455    0.651
## Adjusted.nifty  1.2063346  0.1095622  11.011 9.81e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.009721 on 57 degrees of freedom
## Multiple R-squared:  0.6802, Adjusted R-squared:  0.6746
## F-statistic: 121.2 on 1 and 57 DF,  p-value: 9.813e-16
```

```r
plot(CAPM)
```


Residuals vs Fitted
Residuals
0.04
0.02
0.00
−0.02
2024−06−26
2024−06−28
2024−06−05
−0.06
−0.04
−0.02
0.00
0.02
0.04
Fitted values
lm(Adjusted.rel ~ Adjusted.nifty)


Normal Q−Q
Standardized residuals
4
3
2
1
0
−1
−2
2024−06−26
2024−06−28
2024−06−05
−2
−1
0
1
2
Theoretical Quantiles
lm(Adjusted.rel ~ Adjusted.nifty)

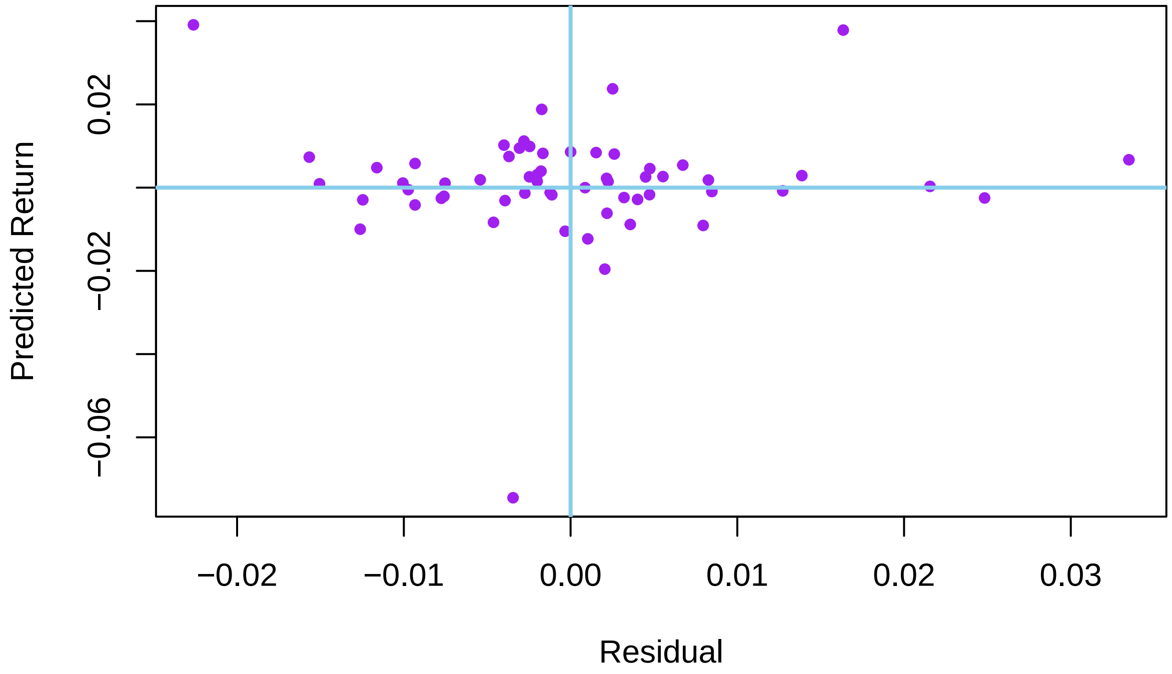

Scale−Location
2024−06−26
2024−06−28
2024−06−05
√|Standardized residuals|
0.0
0.5
1.0
1.5
−0.06
−0.04
−0.02
0.00
0.02
0.04
Fitted values
lm(Adjusted.rel ~ Adjusted.nifty)



reliance prices are fairly priced

Check linearity

```r
resid = CAPM$residuals
y_hat = CAPM$fitted.values
plot(resid,y_hat, xlab = 'Residual', ylab = 'Predicted Return', pch = 20,
     col = 'purple')
abline(h=0, col = 'skyblue', lwd = 2)
abline(v=0, col = 'skyblue', lwd = 2)
```



Rank test for Randomness

```r
options(repos = c(CRAN = "https://cran.rstudio.com/"))
install.packages('randtests')
```

```
## Installing package into 'C:/Users/harsh.hm.mittal/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
## package 'randtests' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\harsh.hm.mittal\AppData\Local\Temp\RtmpqYSEDs\downloaded_packages
```

```r
library(randtests)
```

```
## Warning: package 'randtests' was built under R version 4.2.3
```

```
##
## Attaching package: 'randtests'
```

```
## The following object is masked from 'package:tseries':
##
##     runs.test
```

```r
randtests::bartels.rank.test(resid)
```

```
##
##  Bartels Ratio Test
##
```

```
## data:  resid
## statistic = 0.34151, n = 59, p-value = 0.7327
## alternative hypothesis: nonrandomness
```

Looks like, assumptions of randomness is okay!

##Breusch-Pagan Test for homoskedasticity

```r
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 4.2.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.2.3

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

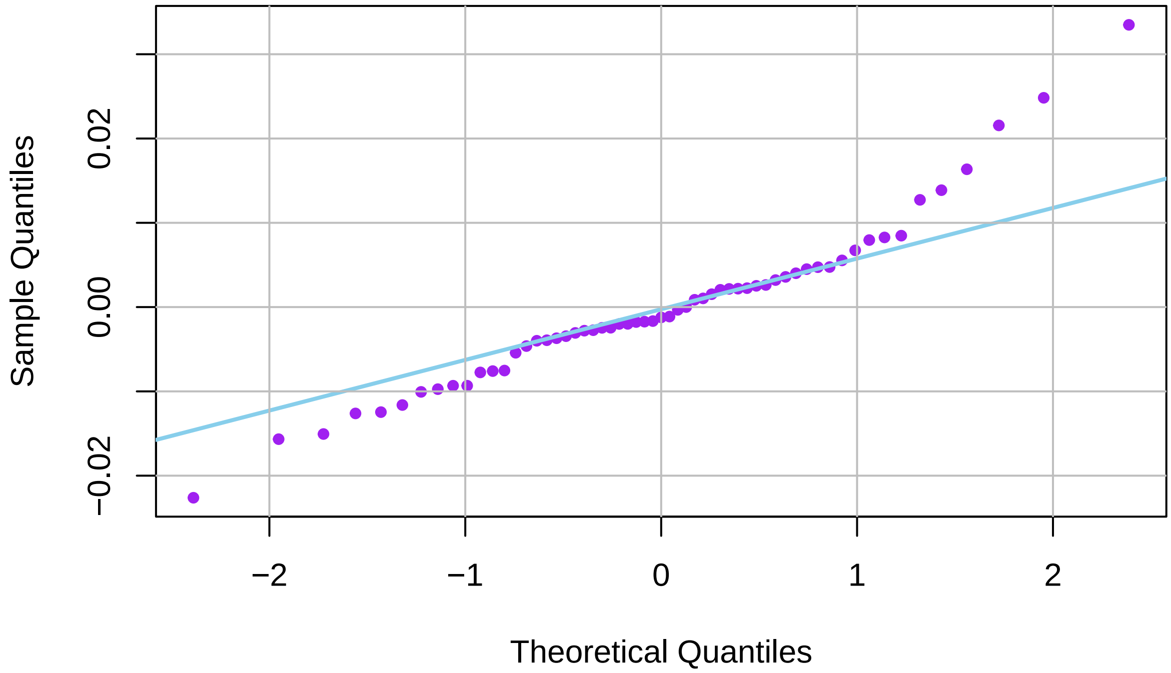
```r
lmtest::bptest(CAPM)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  CAPM
## BP = 2.28, df = 1, p-value = 0.1311
```

Looks like, assumptions of homoskedasticity is okay!

Check Normality

```r
qqnorm(resid,pch=20,col='purple', main = '')
qqline(resid, lwd = 2, col = 'skyblue')
grid(col='grey', lty = 1)
```



## Is the assumption of normality okay?

```r
stats::ks.test(resid, pnorm)
```

```
##
##  Exact one-sample Kolmogorov-Smirnov test
##
## data:  resid
## D = 0.49098, p-value = 1.223e-13
## alternative hypothesis: two-sided
```

OOps!! assumptions of Normality is not correct! What to do?

Bootstrap Regression

paired resampling method

rt = a + b rm + e; e ~ F(), Var(e) = s^2

```r
set.seed(6587)
rt1 = data.frame(rt)
n = nrow(rt1)
B = 1000 ## Bootstrap simulation size
beta_star = matrix(NA,nrow = B, ncol = 2)
colnames(beta_star) = c('alpha', 'beta')
R.squared_star.pair = rep(NA, B)
```

Bootstrap simulation starts

```
for(b in 1:B){
  id_star = sample(1:n, n, replace=TRUE)
  rt_star = rt1[id_star, ]

  CAPM_star = lm(Adjusted.rel~Adjusted.nifty, data = rt_star)
  sum_star = summary(CAPM_star)
  beta_star[b, ] = coef(CAPM_star)
  R.squared_star.pair[b]=sum_star$adj.r.squared
}


sum_boot = cbind(apply(beta_star, 2, mean)
                 ,apply(beta_star, 2, sd)
                 ,apply(beta_star, 2, quantile, probs = 0.025)
                 ,apply(beta_star, 2, quantile, probs = 0.975))
colnames(sum_boot) = c('Estimate', 'Std. Error', '2.5%', '97.5%')
summary(CAPM)
```

```
##
## Call:
## lm(formula = Adjusted.rel ~ Adjusted.nifty, data = rt)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.022619 -0.004309 -0.001223  0.003800  0.033487
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.0005809  0.0012769  -0.455    0.651
## Adjusted.nifty  1.2063346  0.1095622  11.011 9.81e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.009721 on 57 degrees of freedom
## Multiple R-squared:  0.6802, Adjusted R-squared:  0.6746
## F-statistic: 121.2 on 1 and 57 DF,  p-value: 9.813e-16
```

```
sum_boot
```

```
##            Estimate  Std. Error         2.5%       97.5%
## alpha -0.0005358204 0.001233993 -0.002901431 0.002014621
## beta   1.1971163919 0.150445964  0.825617730 1.480900515
```

```
summary(R.squared_star.pair)
```

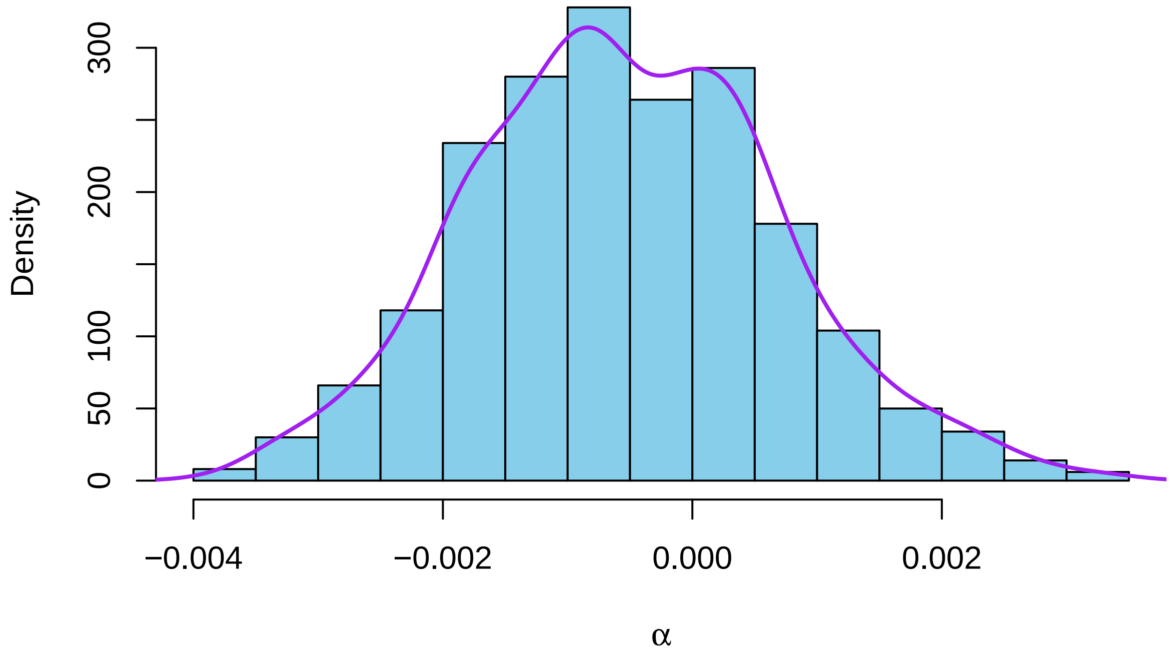
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1594  0.5437  0.6726  0.6386  0.7580  0.9013
```

```
quantile(R.squared_star.pair, probs = c(0.025, 0.975))
```

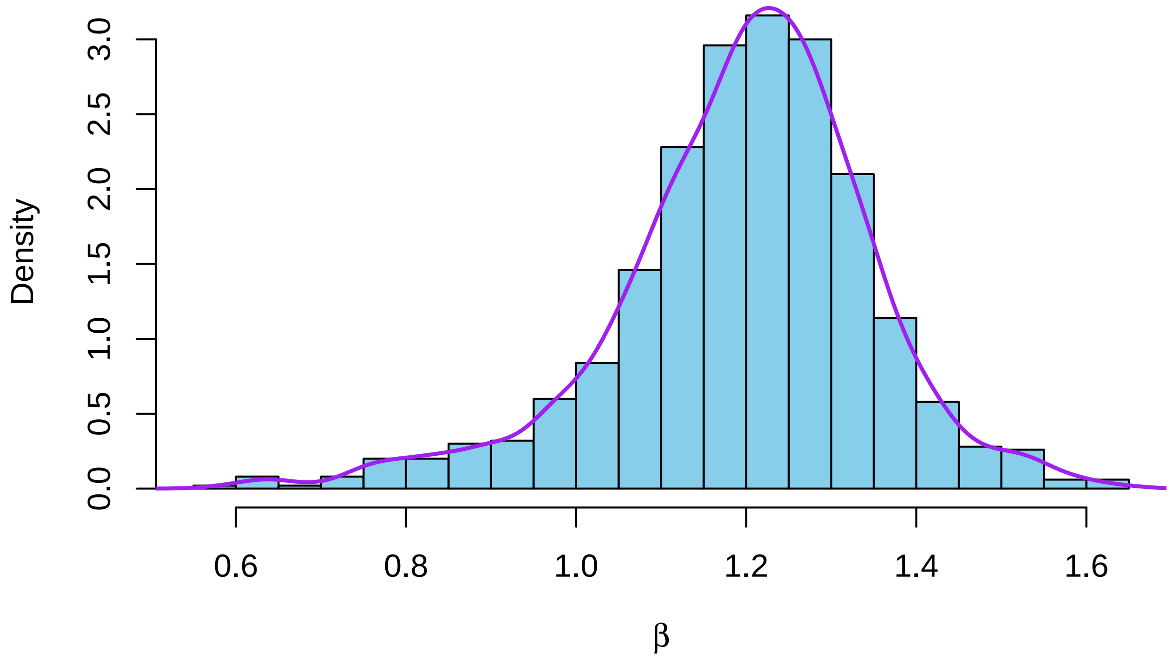
```
##      2.5%     97.5%
## 0.2655320 0.8631489
```

```
hist(beta_star[, 'alpha'], main = '', col='skyblue', freq=FALSE,
     xlab = expression(alpha), nclass = 20)
```

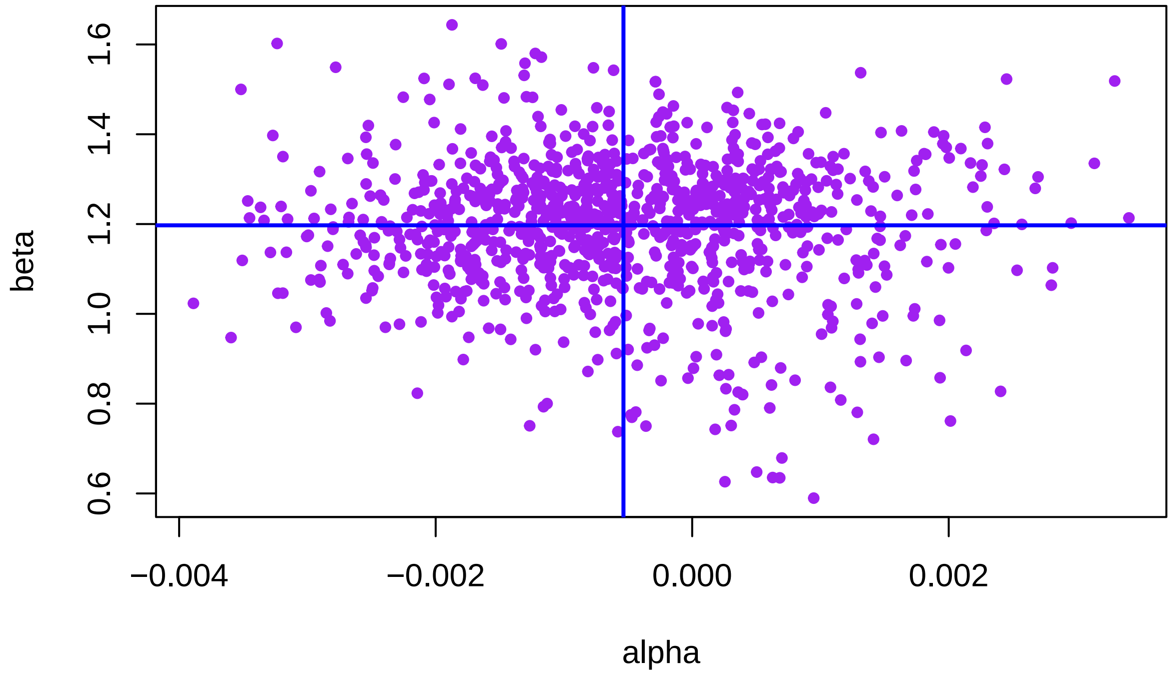
```
lines(density(beta_star[, 'alpha']), col='purple', lwd=2)
```



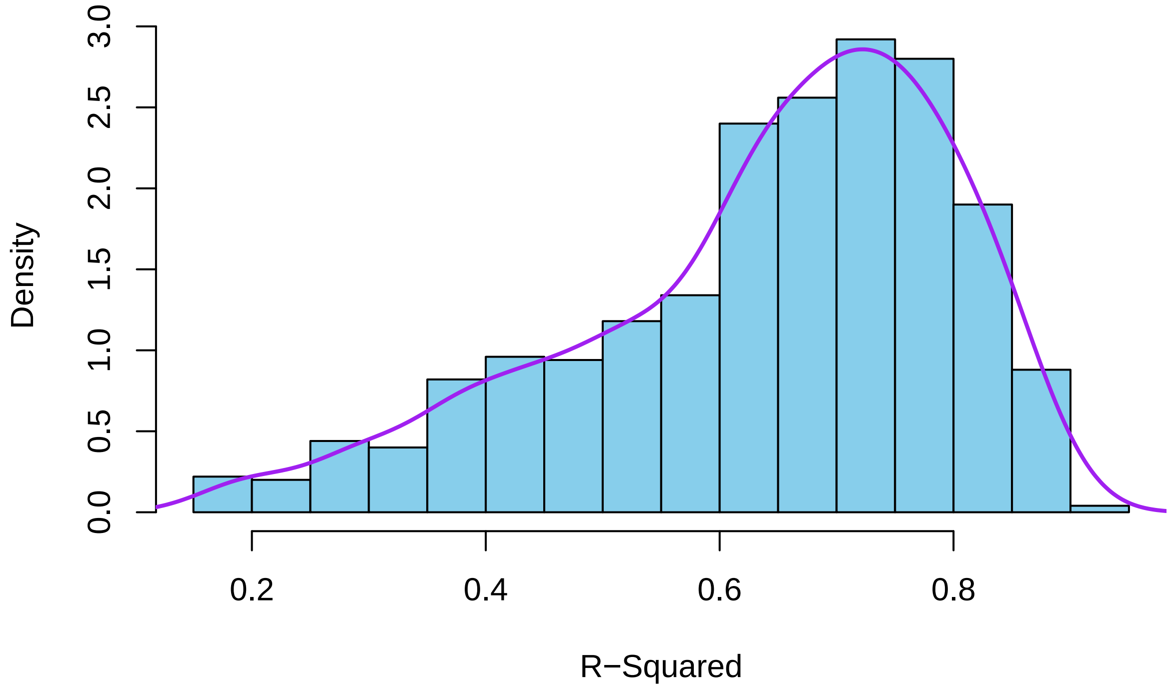
```
hist(beta_star[, 'beta'], main = '', col='skyblue', freq=FALSE,
     xlab = expression(beta), nclass = 20)
lines(density(beta_star[, 'beta']), col='purple', lwd=2)
```



```r
plot(beta_star, pch = 20, col = 'purple')
abline(h=mean(beta_star[,'beta']), col='blue', lwd=2)
abline(v=mean(beta_star[,'alpha']), col='blue', lwd=2)
```



```
hist(R.squared_star.pair, main = '', col='skyblue', freq=FALSE,
     xlab = 'R-Squared', nclass = 20)
lines(density(R.squared_star.pair), col='purple', lwd=2)
```



```
quantile(R.squared_star.pair, probs = c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.2655320 0.8631489
```

Residual Bootstrap Regression Remapling

```
set.seed(6587)

ols_resid = CAPM$residuals
ols_pred = CAPM$fitted.values



Adjusted.nifty = rt$Adjusted.nifty
B = 1000
beta_star2 = matrix(NA,nrow = B, ncol = 2)
colnames(beta_star2) = c('alpha', 'beta')
R.squared_star.resid = rep(NA, B)
```

Bootstrap simulation

```
for(b in 1:B){
  id_star = sort(sample(1:n, n, replace = TRUE))
  resid_star = ols_resid[id_star]
```

```r
  pred_star = ols_pred + resid_star
  CAPM_star = lm(pred_star~Adjusted.nifty)
  sum_star = summary(CAPM_star)
  beta_star2[b,] = coef(CAPM_star)
  R.squared_star.resid[b] = sum_star$adj.r.squared
}

sum_boot2 = cbind(apply(beta_star2, 2, mean)
                 ,apply(beta_star2, 2, sd)
                 ,apply(beta_star2, 2, quantile, probs = 0.025)
                 ,apply(beta_star2, 2, quantile, probs = 0.975))
colnames(sum_boot2) = c('Estimate', 'Std. Error', '2.5%', '97.5%')
summary(CAPM)
```

```
##
## Call:
## lm(formula = Adjusted.rel ~ Adjusted.nifty, data = rt)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.022619 -0.004309 -0.001223  0.003800  0.033487
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.0005809  0.0012769  -0.455    0.651
## Adjusted.nifty  1.2063346  0.1095622  11.011 9.81e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.009721 on 57 degrees of freedom
## Multiple R-squared:  0.6802, Adjusted R-squared:  0.6746
## F-statistic: 121.2 on 1 and 57 DF,  p-value: 9.813e-16
```

```r
sum_boot2
```

```
##           Estimate  Std. Error       2.5%       97.5%
## alpha -0.000581262 0.001251092 -0.0029502 0.002087627
## beta   1.203706764 0.081215634  1.0414498 1.372122267
```

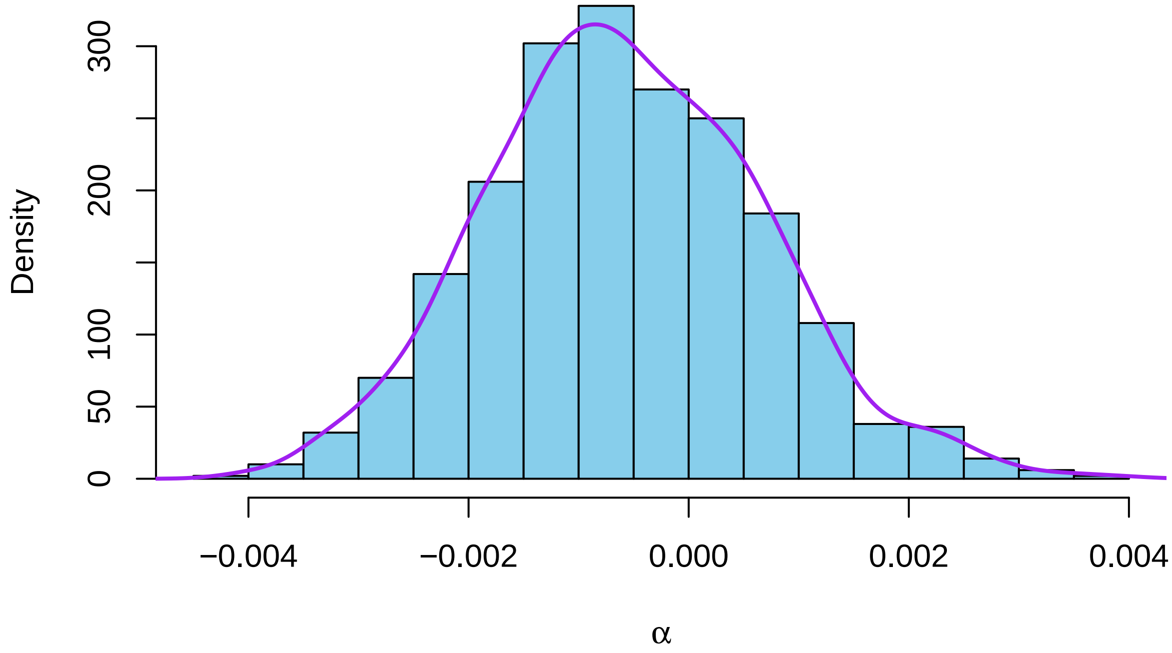
```r
summary(R.squared_star.resid)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.5057  0.6406  0.6848  0.6836  0.7285  0.8651
```

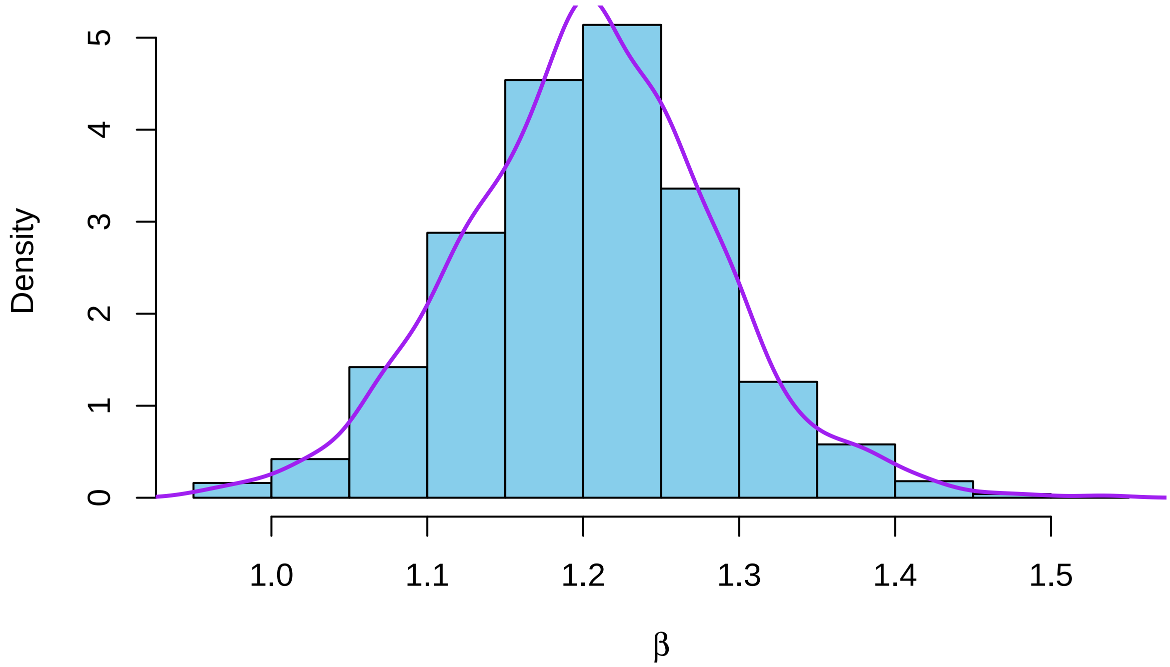
```r
quantile(R.squared_star.resid, probs = c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.5555642 0.7988795
```

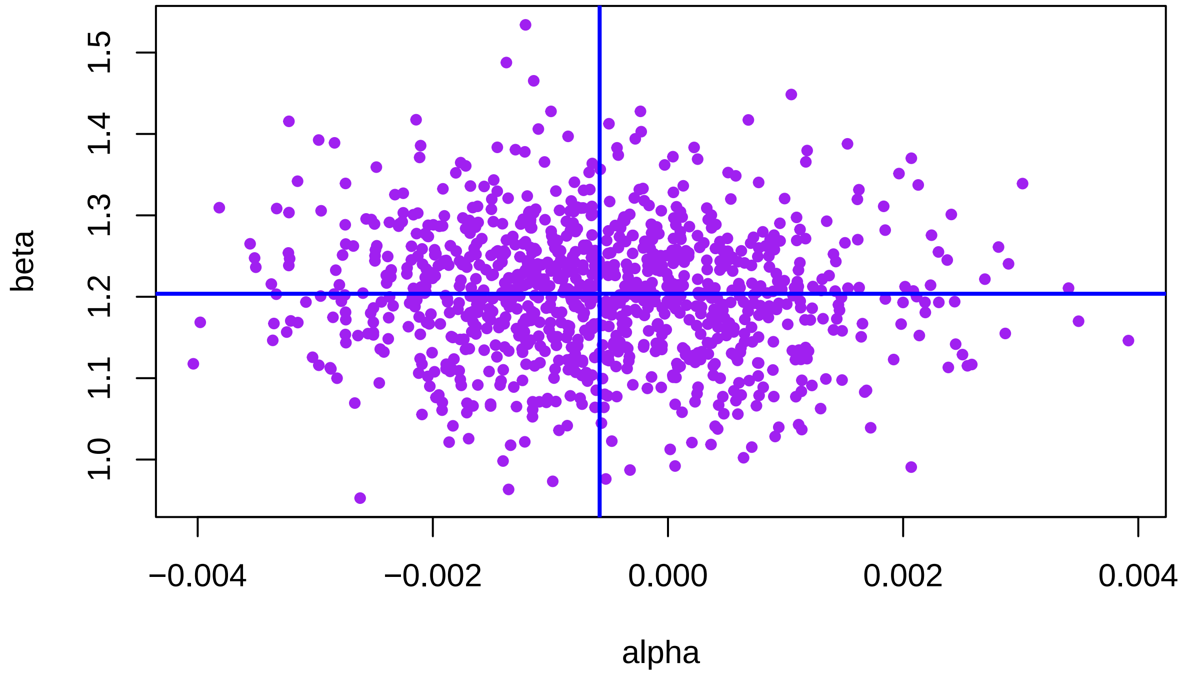
```r
hist(beta_star2[, 'alpha'], main = '', col='skyblue', freq=FALSE,
     xlab = expression(alpha), nclass = 20)
lines(density(beta_star2[, 'alpha']), col='purple', lwd=2)
```



```
hist(beta_star2[, 'beta'], main = '', col='skyblue', freq=FALSE,
     xlab = expression(beta), nclass = 20)
lines(density(beta_star2[, 'beta']), col='purple', lwd=2)
```



```
plot(beta_star2, pch = 20, col = 'purple')
abline(h=mean(beta_star2[,'beta']), col='blue', lwd=2)
abline(v=mean(beta_star2[,'alpha']), col='blue', lwd=2)
```


beta
1.0
1.1
1.2
1.3
1.4
1.5
−0.004
−0.002
0.000
0.002
0.004
alpha