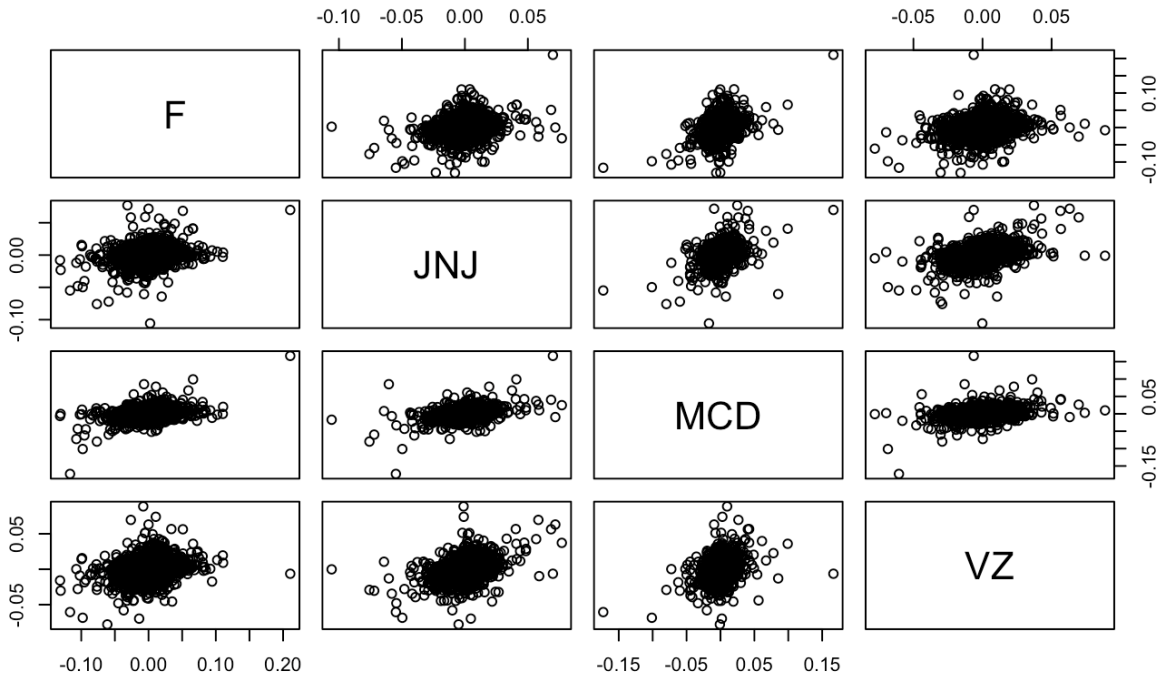
```r
#1a
rm(list=ls())
load("/Users/kartikayjain/Downloads/returns.Jan.19.2024.RData")
R = returns[, c("F","JNJ","MCD","VZ")]
pairs(as.matrix(R))
#there is a lot of tail dependence as extreme values cooccur together so t copula is best
```



```r
#1b
library(rugarch)
Ford=returns[,"F"]
est.F = as.numeric(fitdist(distribution="std",Ford)$pars)
cat("Estimated df of Ford is ", signif(est.F[3], digits=3))
cat("\n")
JNJ=returns[,"JNJ"]
est.JNJ = as.numeric(fitdist(distribution="std",JNJ)$pars)
cat("Estimated df of JNJ is ", signif(est.JNJ[3], digits=3))
cat("\n")
MCD=returns[,"MCD"]
est.MCD = as.numeric(fitdist(distribution="std",MCD)$pars)
cat("Estimated df of MCD is ", signif(est.MCD[3], digits=3))
cat("\n")
VZ=returns[,"VZ"]
est.VZ= as.numeric(fitdist(distribution="std",VZ)$pars)
cat("Estimated df of VZ is ", signif(est.VZ[3], digits=3))
cat("\n")
```

```
Loading required package: parallel

Attaching package: ‘rugarch’

The following object is masked from ‘package:stats’:

    sigma

Estimated df of Ford is  2.95
Estimated df of JNJ is  3.32
Estimated df of MCD is  3.22
Estimated df of VZ is  3.65
```

```{r}
#1c

kend_cor <- cor(R,method="kendall")
om = sin((pi/2)*kend_cor)

om
cat("\n")

eigen(om)$values

#Since all the eigenvalues are positive the matrix is positive definite
```

```
            F       JNJ       MCD        VZ
F   1.0000000 0.2494333 0.3330756 0.2760072
JNJ 0.2494333 1.0000000 0.4188353 0.4168419
MCD 0.3330756 0.4188353 1.0000000 0.3434801
VZ  0.2760072 0.4168419 0.3434801 1.0000000

[1] 2.0259874 0.7841525 0.6477807 0.5420794
```

```{r}
#1d

om_mat=c(om[1,2],om[1,3],om[1,4],om[2,3],om[2,4],om[3,4])

library(copula)

t_cop <- tCopula(om_mat,dim=4,dispstr = "un",df=4)
data = cbind(pdist(distribution="std",Ford,est.F[1],est.F[2],est.F[3]),pdist(distribution="std",JNJ,est.JNJ[1],est.JNJ[2],est.JNJ[3]),pdist(distribution="std",MCD,est.MCD[1],est.MCD[2],est.MCD[3]),pdist(distribution="std",VZ,est.VZ[1],est.VZ[2],est.VZ[3]))

cop_fitted = fitCopula(t_cop, data,method="ml",start=c(om_mat,3))
cop_fitted

#estimated df for the copula is 8.0250
#the correlation coefficient between JNJ and Verizon returns which is rho.5 is 0.4620
```

```
Call: fitCopula(t_cop, data = data, ... = pairlist(method = "ml", start = c(om_mat,
    3)))
Fit based on "maximum likelihood" and 2780 4-dimensional observations.
Copula: tCopula
 rho.1  rho.2  rho.3  rho.4  rho.5  rho.6     df
0.3024 0.4060 0.3206 0.4841 0.4620 0.4053 8.0250
The maximized loglikelihood is 977.9
Optimization converged
```

```{r}
#1e
library(fGarch)

mvdc_metat = mvdc(t_cop, c("std","std","std","std"),
list(list(mean=est.F[1],sd=est.F[2],nu=est.F[3]),
list(mean=est.JNJ[1],sd=est.JNJ[2],nu=est.JNJ[3]),list(mean=est.MCD[1],sd=est.MCD[2],nu=est.MCD[3]),list(mean=est.VZ[1],sd=est.VZ[2],nu=est.VZ[3])))

start=c(est.F, est.JNJ, est.MCD,est.VZ, cop_fitted@estimate)
fit_mvdc = fitMvdc(as.matrix(R),mvdc_metat,start)

fit_mvdc@estimate[1:3]
fit_mvdc@estimate[13:18]
#the above line gives the correlation values. rho.5 is for JNJ and VZ which is 0.4255
fit_mvdc@estimate[19]
#df estimate of the copula which is 6.63
summary(fit_mvdc)
#the estimated df values for the four marginal distributions are:
#Ford: 2.82
#JNJ: 3.29
#MCD: 3.19
#VZ: 3.58

```

```
[1] 0.0003433345 0.0241041117 2.8237622127
[1] 0.2482805 0.3429090 0.2747614 0.4370302 0.4255159 0.3598000
[1] 6.634429
Call: fitMvdc(data = as.matrix(R), mvdc = mvdc_metat, start = start)
Maximum Likelihood estimation based on 2780 4-dimensional observations.
Copula:  tCopula
Margin 1 :
         Estimate Std. Error
m1.mean 0.0003433      0.000
m1.sd   0.0241041      0.001
m1.nu   2.8237622      0.162
Margin 2 :
         Estimate Std. Error
m2.mean 0.0006412      0.000
m2.sd   0.0114853      0.000
m2.nu   3.2943977      0.191
Margin 3 :
         Estimate Std. Error
m3.mean 0.0006943      0.000
m3.sd   0.0118854      0.000
m3.nu   3.1939436      0.178
Margin 4 :
         Estimate Std. Error
m4.mean 0.0003354      0.000
m4.sd   0.0122780      0.000
m4.nu   3.5830426      0.248
t-copula, dim. d = 4
      Estimate Std. Error
rho.1   0.2483      0.019
rho.2   0.3429      0.018
rho.3   0.2748      0.019
rho.4   0.4370      0.017
rho.5   0.4255      0.017
rho.6   0.3598      0.018
df      6.6344      0.507
The maximized loglikelihood is 34454
Optimization converged
Number of loglikelihood evaluations:
function gradient
     217       22
```

```r
#2

p_copula = function(y1,y2) {
  pCopula(c(plnorm(y1,1,0.5),pbeta(y2,2,4)),tCopula(0.85,df=4))
}
signif(p_copula(3,0.65),3)


```

```
[1] 0.577
```

```{r}
#3a

library(copula)

num_def_loans = apply((rCopula(10^6,normalCopula(dim=12,param=0.9, dispstr="ex"))<pexp(1,rate=0.1)),1,sum)
p1=mean(num_def_loans>=7)
signif(p1,3)


```

```
[1] 0.0803
```

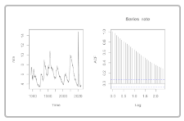
```{r}
#3b

num_def_loans_t = apply((rCopula(10^6,tCopula(dim=12,param=0.9, dispstr="ex",df=1))<pexp(1,rate=0.1)),1,sum)
p2=mean(num_def_loans_t>=7)
signif(p2,3)

```

```
[1] 0.0875
```

```{r}
#3c

signif((p1-p2) + c(-1,1)*qnorm(0.995)*sqrt((p1*(1-p1) + p2*(1-p2))/10^6),3)

#0 is not in the confidence interval so I can't say that the probability difference is 0
#i.e the probability from the 2 copulas do seem different

```

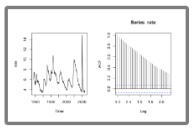
```
[1] -0.00828 -0.00626
```

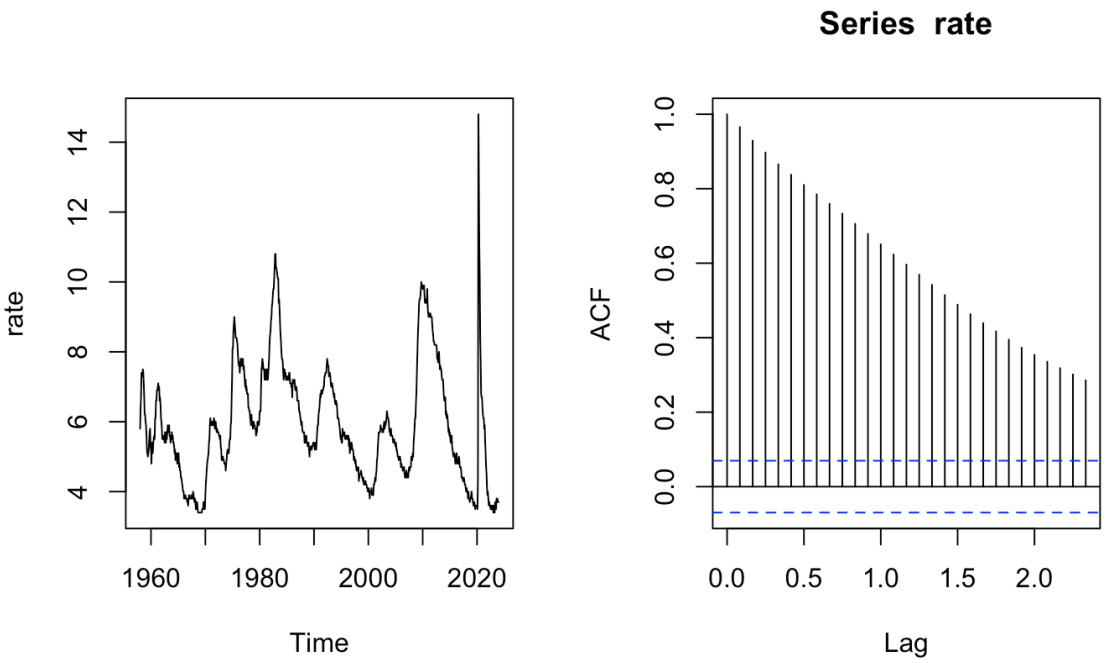
```{r}
#4a

load("/Users/kartikayjain/Downloads/UNRATE.RData")
frequency(rate)
start(rate)
end(rate)

par(mfrow=c(1,2))
plot(rate)
acf(rate)
#It seems non-stationary as it does not revert to mean and acf decays slowly and for the most part is outside of the blue bounds.

```



```
[1] 12
[1] 1958    1
[1] 2024    1
```


R Console

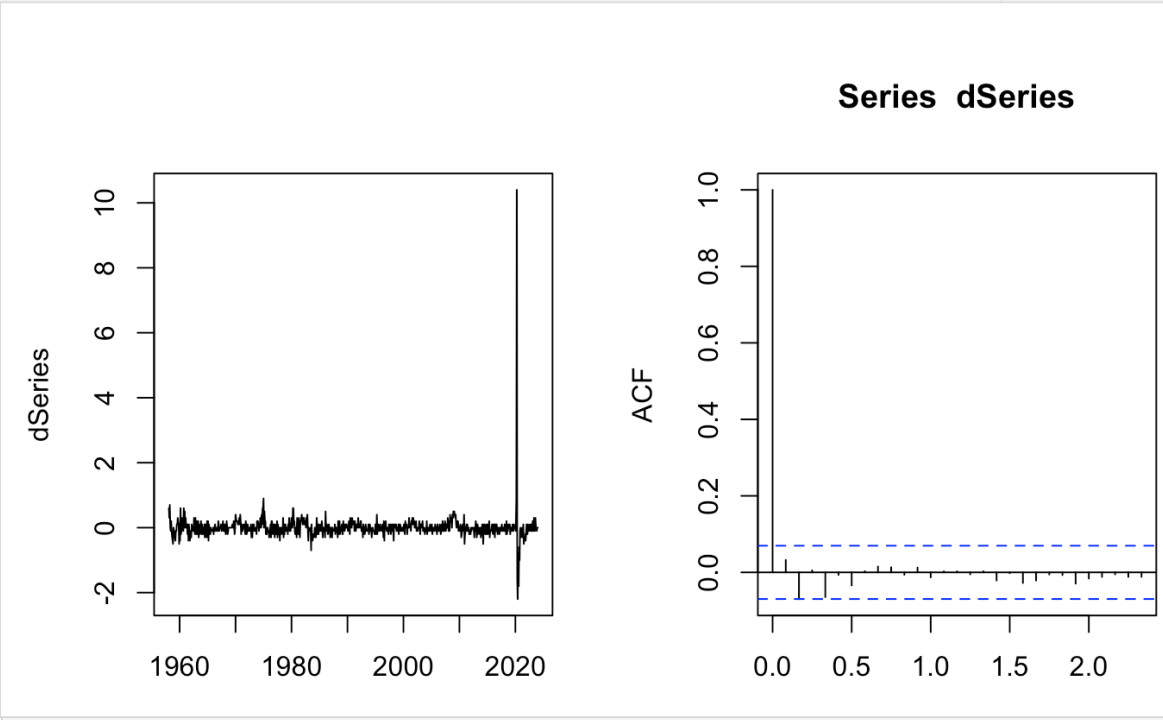



Series rate
rate
4
6
8
10
12
14
1960
1980
2000
2020
Time
ACF
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
1.5
2.0
Lag

```{r}
#4b

library(tseries)
adf.test(rate) #this adf test states that the process is not stationary as the p value is not significant (larger than 0.05)
kpss.test(rate) #kpss test states that process is stationary as the p value is not significant (larger than 0.05)
```

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

    'tseries' version: 0.10-55

    'tseries' is a package for time series analysis and computational finance.

    See 'library(help="tseries")' for details.


        Augmented Dickey-Fuller Test

data:  rate
Dickey-Fuller = -3.3151, Lag order = 9, p-value = 0.06808
alternative hypothesis: stationary


        KPSS Test for Level Stationarity

data:  rate
KPSS Level = 0.44888, Truncation lag parameter = 6, p-value = 0.05608
```

```r
#4c
dSeries = diff(rate)
par(mfrow=c(1,2))
plot(dSeries)
acf(dSeries)


#we analyze everything before 2020. ACF decays quickly and there is a mean reversion.Thus, Differentiated time series appears stationary pre-pandemic.

```



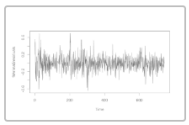
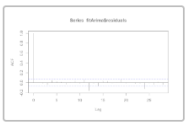
```{r}
#4d

adf.test(dSeries) #this adf test states that the process is stationary as the p value is significant
kpss.test(dSeries) #kpss test states that process is stationary as the p value is not significant
#Thus difference series is stationary
```

```
Warning: p-value smaller than printed p-value
        Augmented Dickey-Fuller Test

data:  dSeries
Dickey-Fuller = -9.1917, Lag order = 9, p-value = 0.01
alternative hypothesis: stationary

Warning: p-value greater than printed p-value
        KPSS Test for Level Stationarity

data:  dSeries
KPSS Level = 0.034175, Truncation lag parameter = 6, p-value = 0.1
```

```r

#4e
library(forecast)
len=length(dSeries)
fitArima = auto.arima(dSeries[-((len-48):len)], ic="bic",seasonal=FALSE)
summary(fitArima)


```

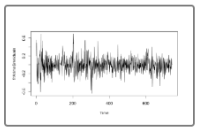
```
Series: dSeries[-((len - 48):len)]
ARIMA(1,0,2) with zero mean

Coefficients:
         ar1      ma1     ma2
      0.8398  -0.8123  0.1919
s.e.  0.0384   0.0510  0.0376

sigma^2 = 0.0292:  log likelihood = 259.76
AIC=-511.51   AICc=-511.46   BIC=-493.07

Training set error measures:
                      ME      RMSE      MAE MPE MAPE      MASE          ACF1
Training set -0.001577459 0.1705423 0.128652 NaN  Inf 0.7039805 -0.0002798374
```

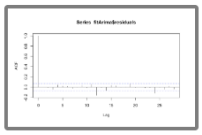
```{r}
#4f
plot(fitArima$residuals)
acf(fitArima$residuals)
Box.test(fitArima$residuals,lag=25,fitdf=3)
#there is a mean that the residuals keeps reverting to and most of the acf values are within the confidence interval and also the p value of the test is significant so the model shows good fit.
```

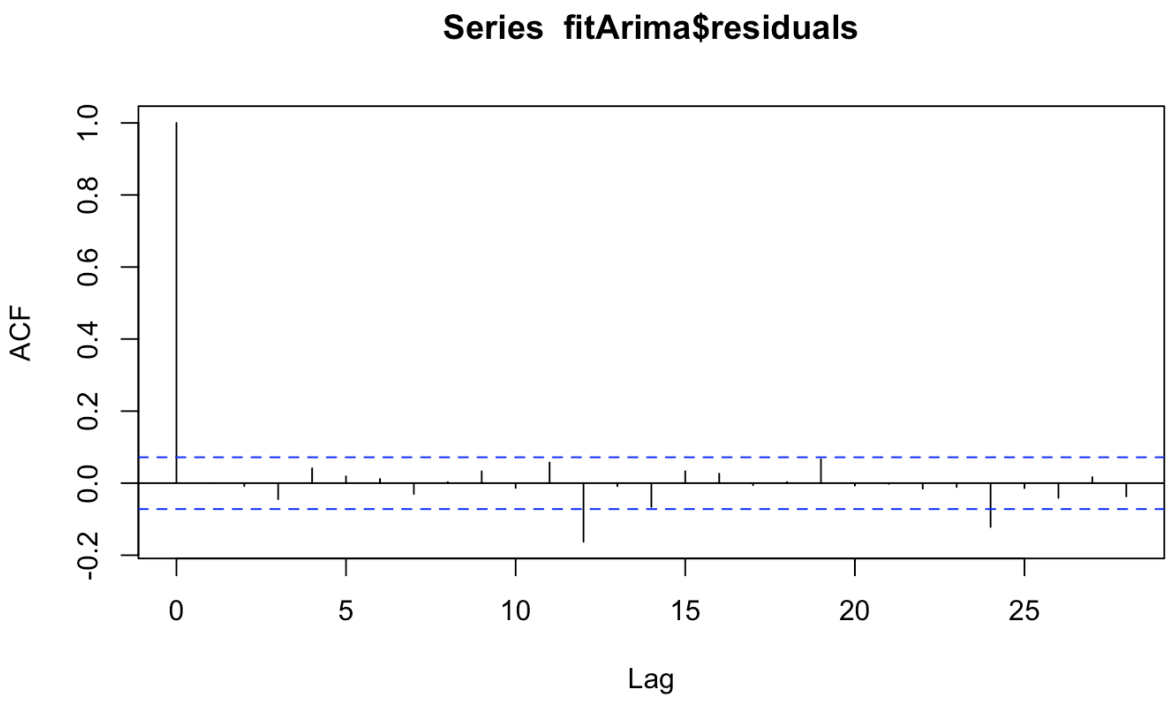


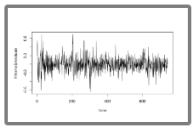
```
        Box-Pierce test

data:  fitArima$residuals
X-squared = 46.184, df = 22, p-value = 0.001872
```

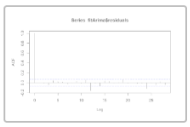




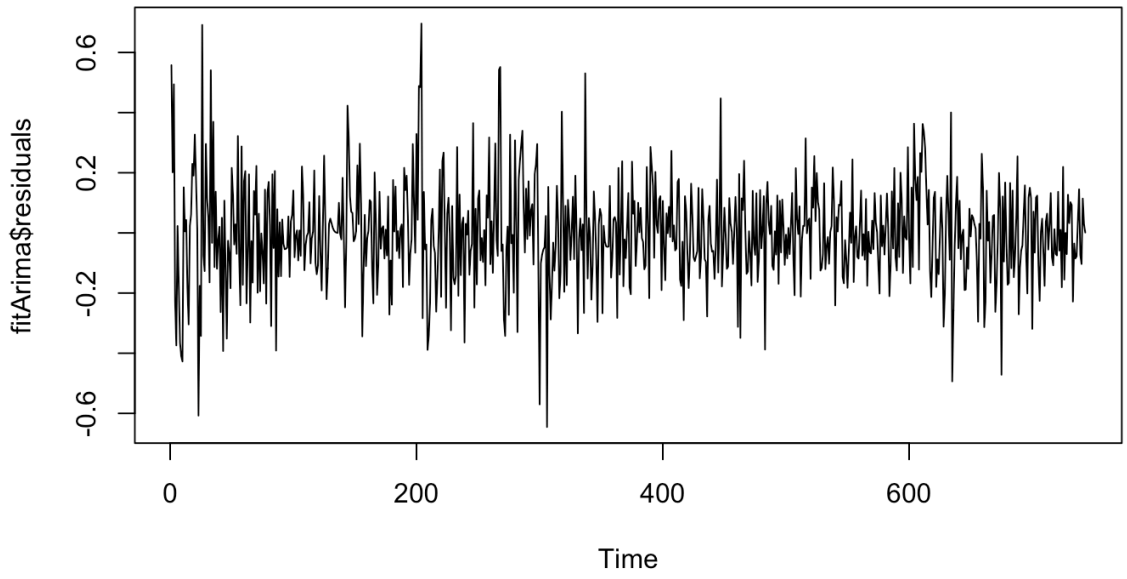

R Console


Series fitArima$residuals
ACF
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
0
5
10
15
20
25
Lag






R Console


fitArima$residuals
0.6
0.2
-0.2
-0.6
0
200
400
600
Time