harinris_Homework2

Harin Rishabh

2024-02-22

Chapter 1 - R Commands

Loading library

```
library(TSA)
```

```
##
## Attaching package: 'TSA'
```

```
## The following objects are masked from 'package:stats':
##
##     acf, arima
```

```
## The following object is masked from 'package:utils':
##
##     tar
```

Exhibit 1.1

```
data(larain)
plot(larain,ylab='Inches',xlab='Year',type='o')
```

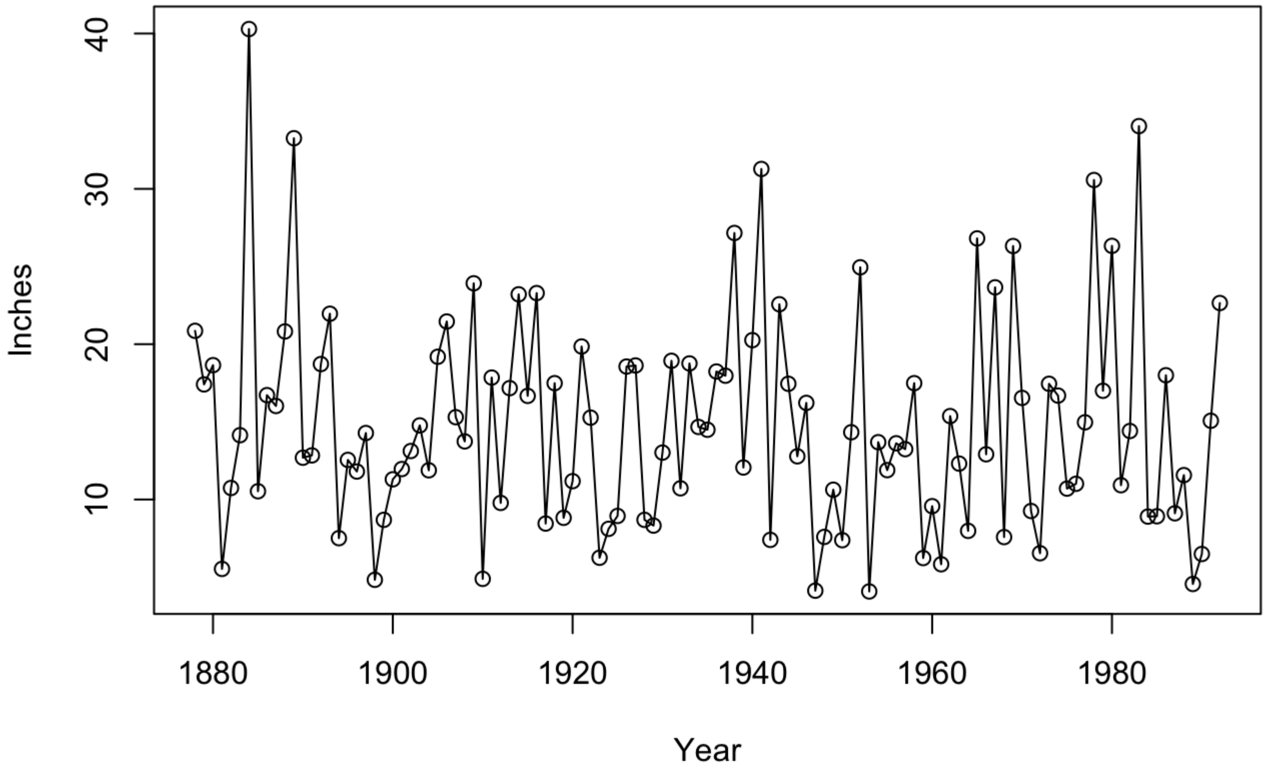


Exhibit 1.2

```
plot(y=larain,x=zlag(larain),ylab='Inches',xlab='Previous Year Inches')
```

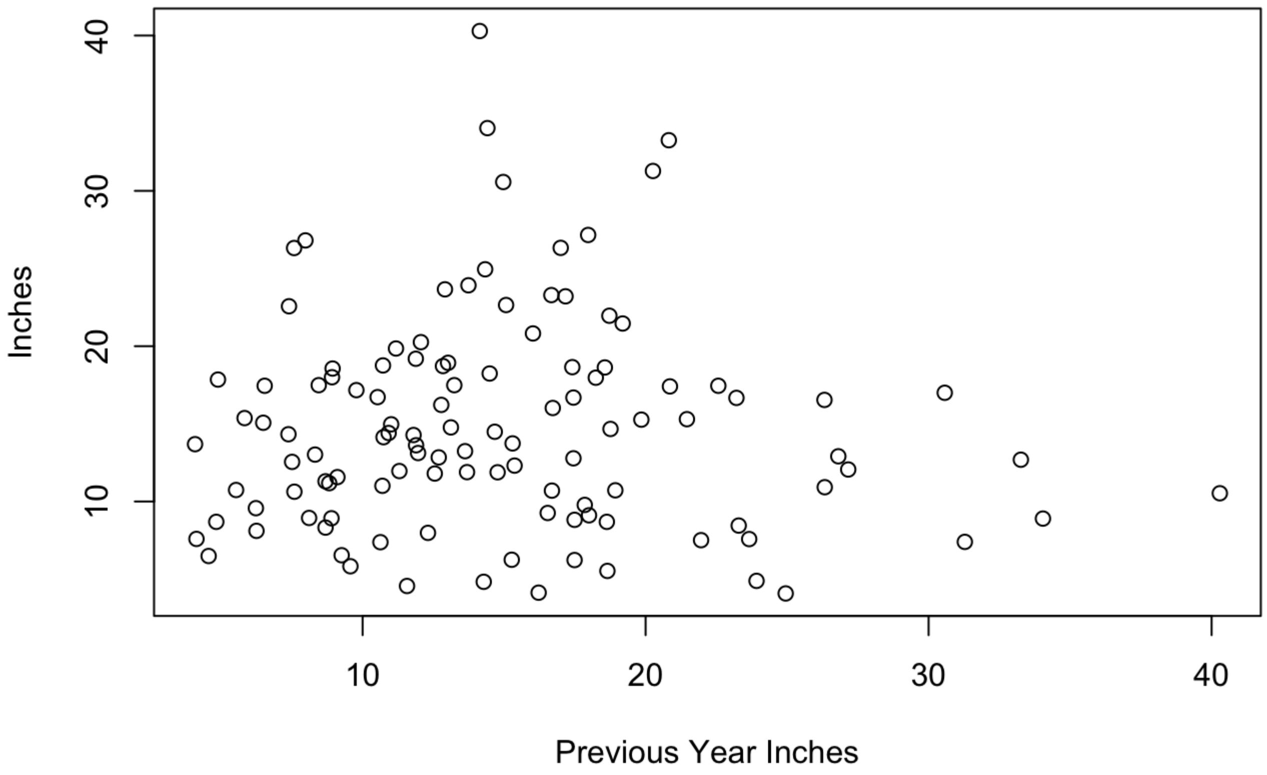


Exhibit 1.3

```
data(color)
plot(color,ylab='Color Property',xlab='Batch',type='o')
```

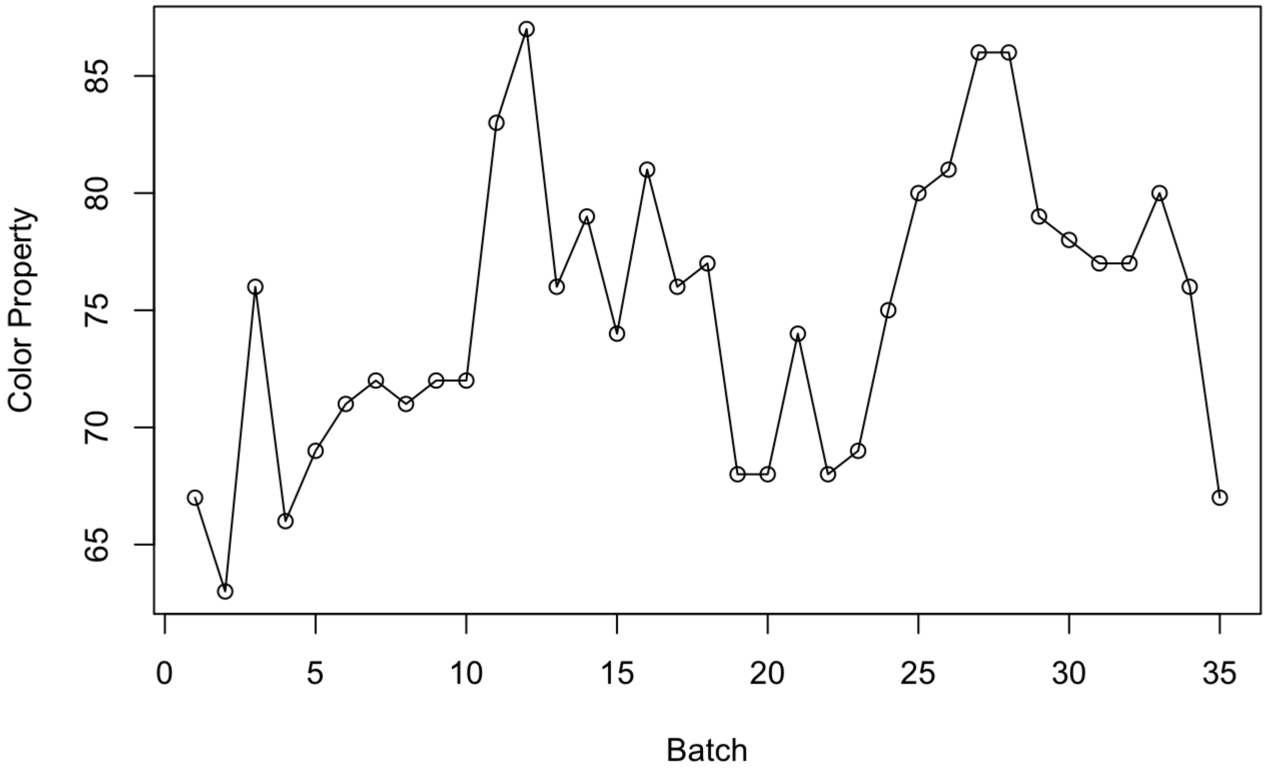


Exhibit 1.4

```
plot(y=color,x=zlag(color),ylab='Color Property',
xlab='Prevous Batch Color Property')
```

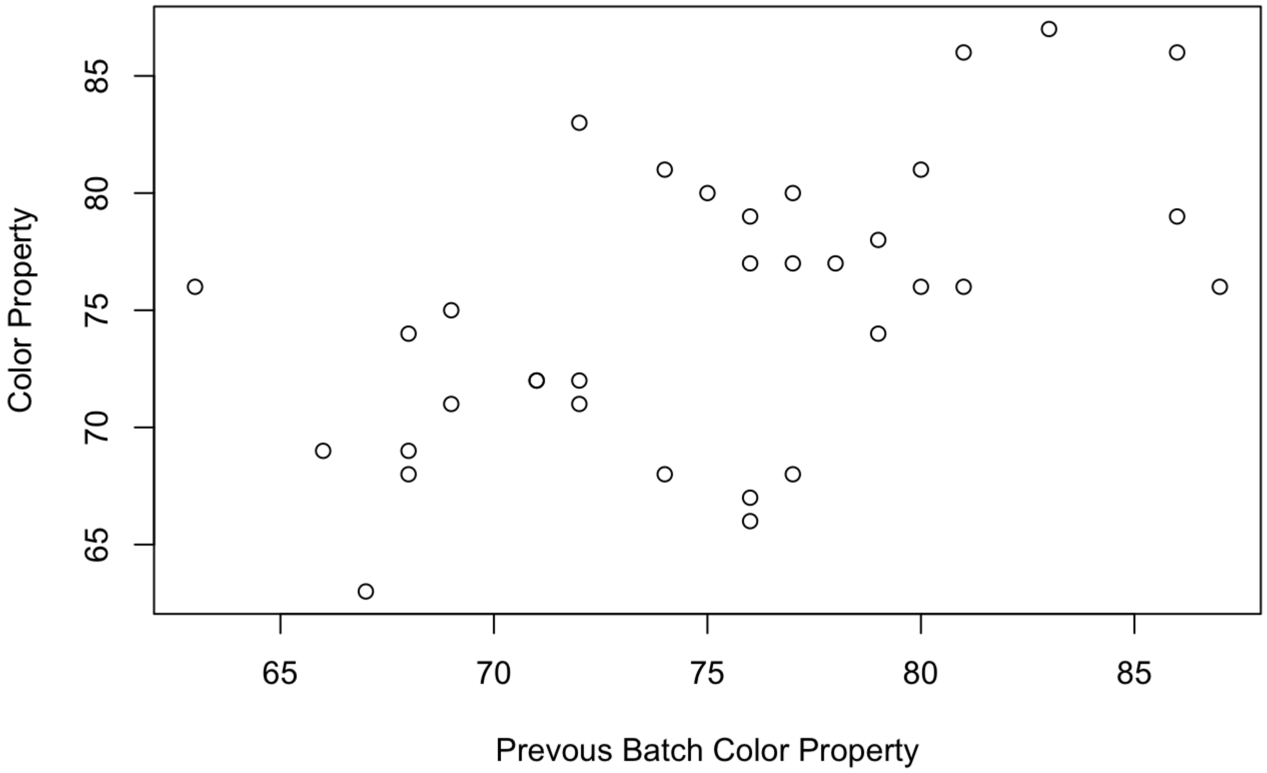


Exhibit 1.5

```
data(hare)
plot(hare,ylab='Abundance',xlab='Year',type='o')
```

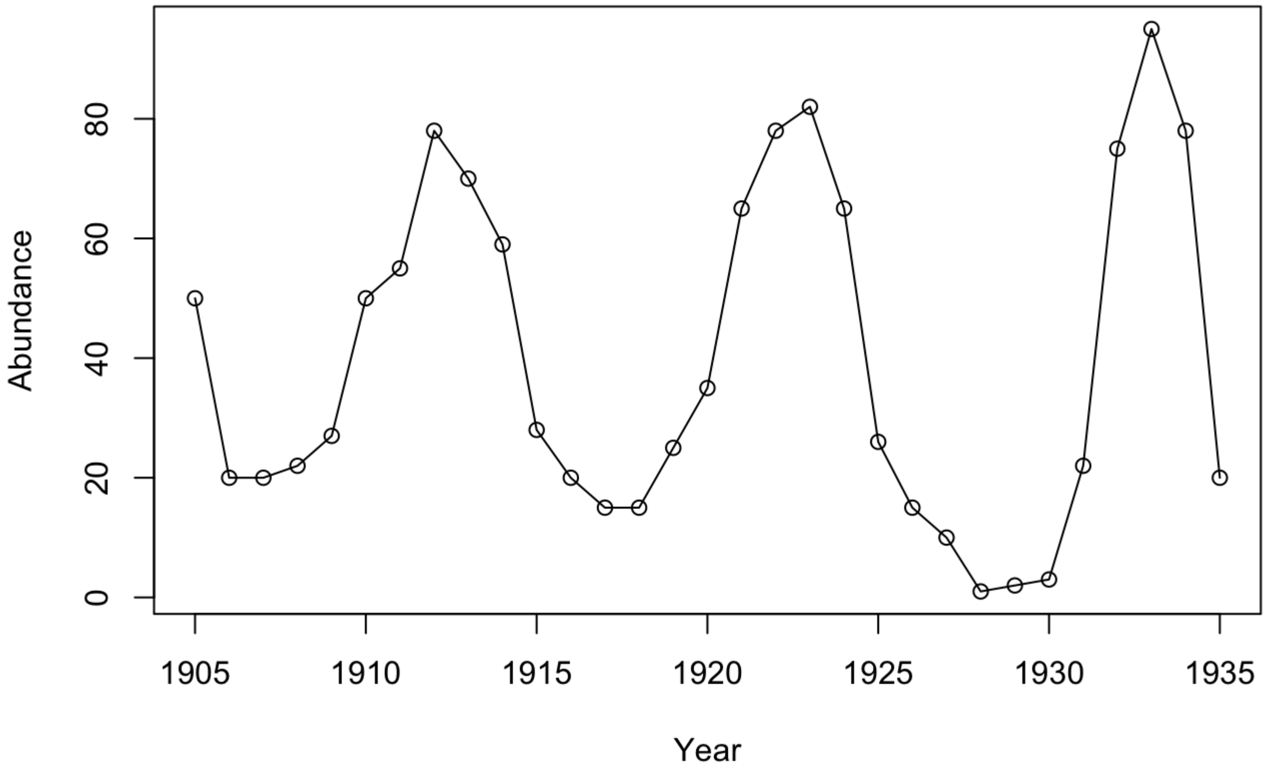


Exhibit 1.6

```
plot(y=hare,x=zlag(hare),ylab='Abundance',xlab='Previous Year Abundance')
```

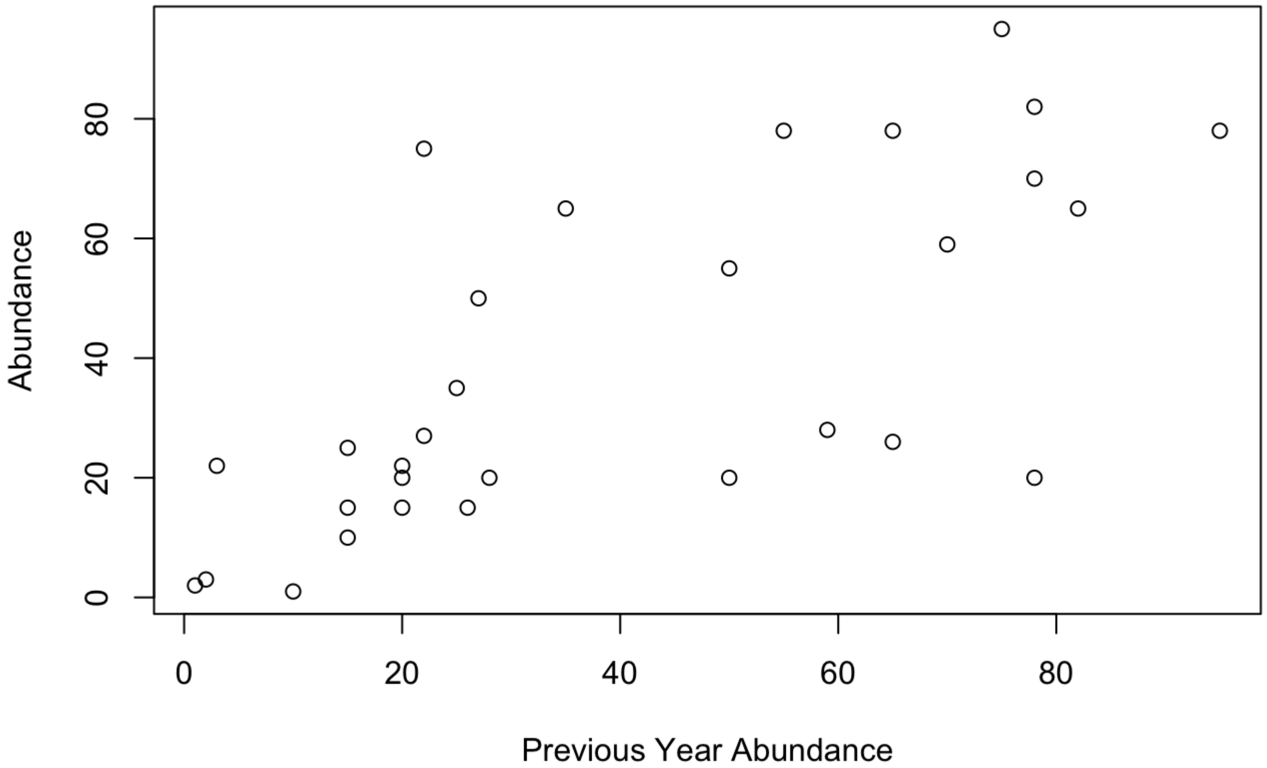


Exhibit 1.7

```
data(tempdub)
plot(tempdub,ylab='Temperature',type='o')
```

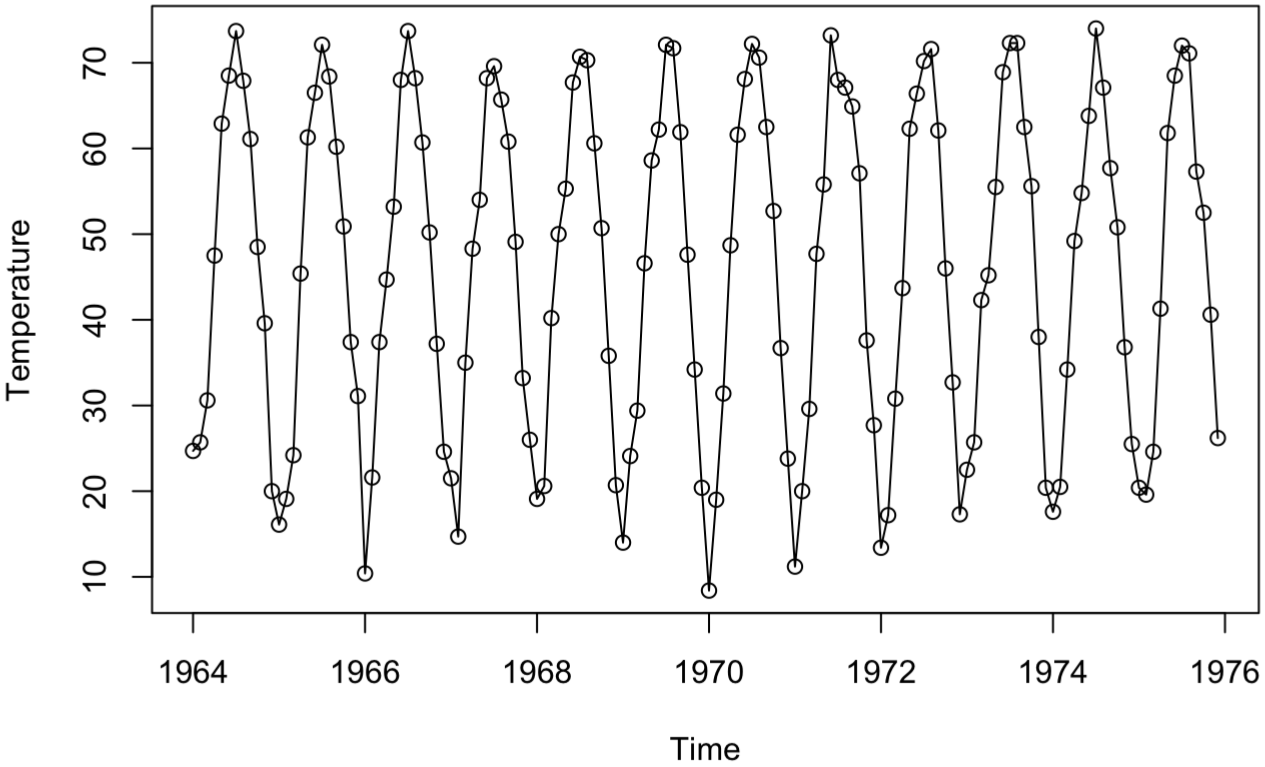


Exhibit 1.8

```
data(oilfilters)
plot(oilfilters,type='o',ylab='Sales')
```

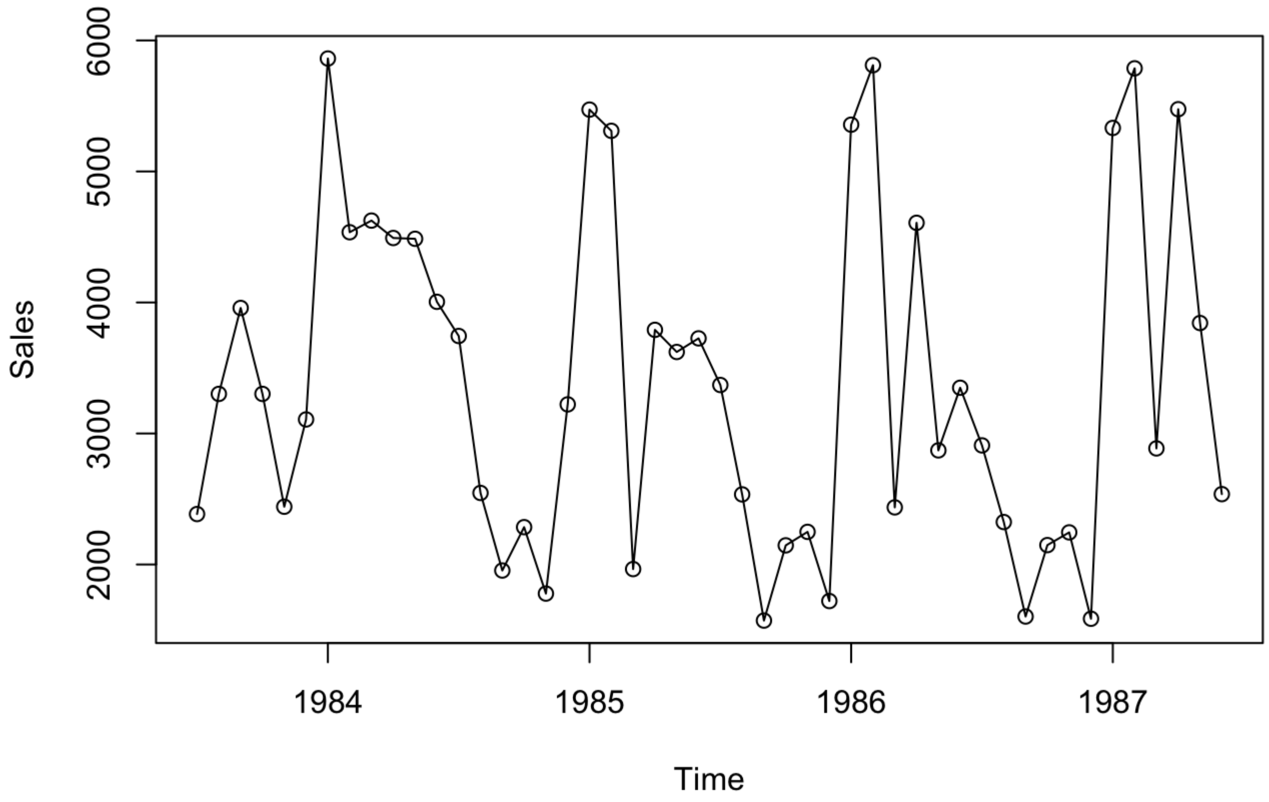
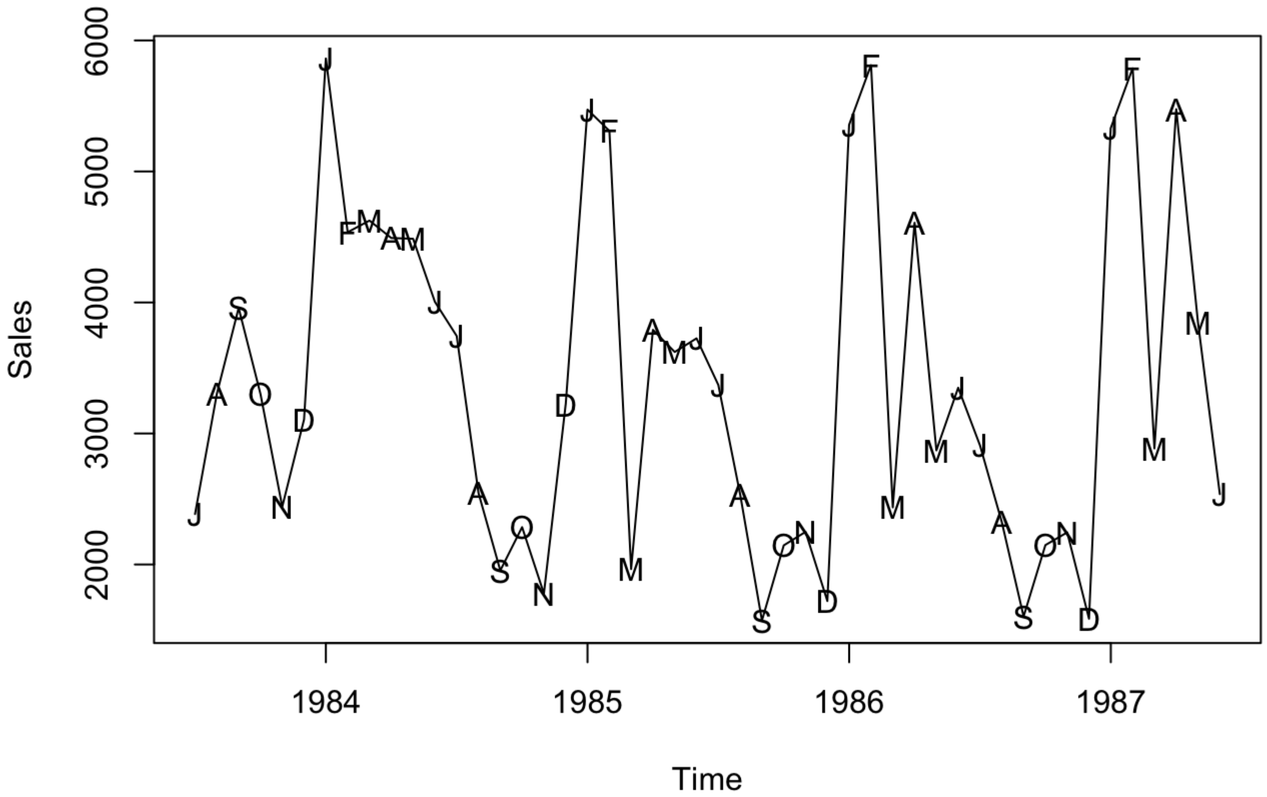


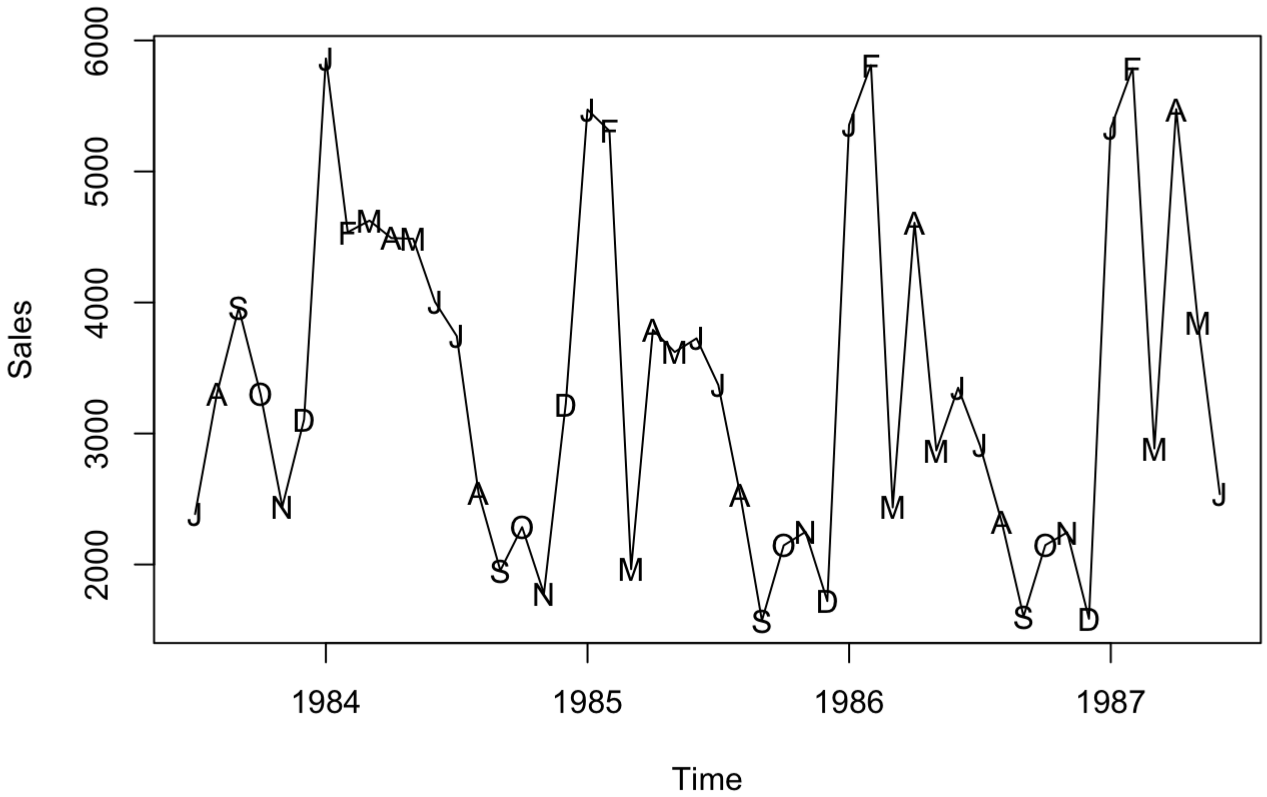
Exhibit 1.9

```
plot(oilfilters,type='l',ylab='Sales')
Month=c("J","A","S","O","N","D","J","F","M","A","M","J")
points(oilfilters,pch=Month)
```



Alternatively, the exhibit can be reproduced by the following commands

```
plot(oilfilters,type='l',ylab='Sales')
points(y=oilfilters,x=time(oilfilters),pch=as.vector(season(oilfilters)))
```

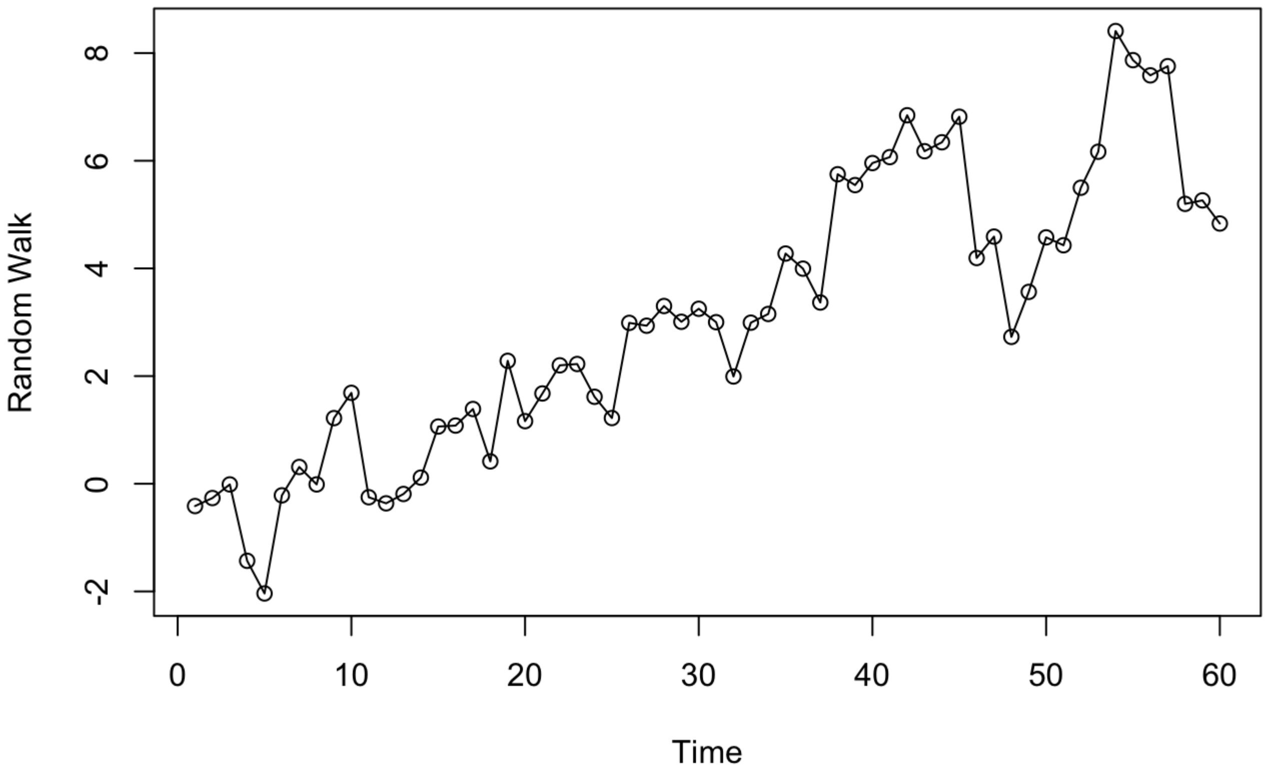


Chapter 2 - R Commands

Exhibit 2.1

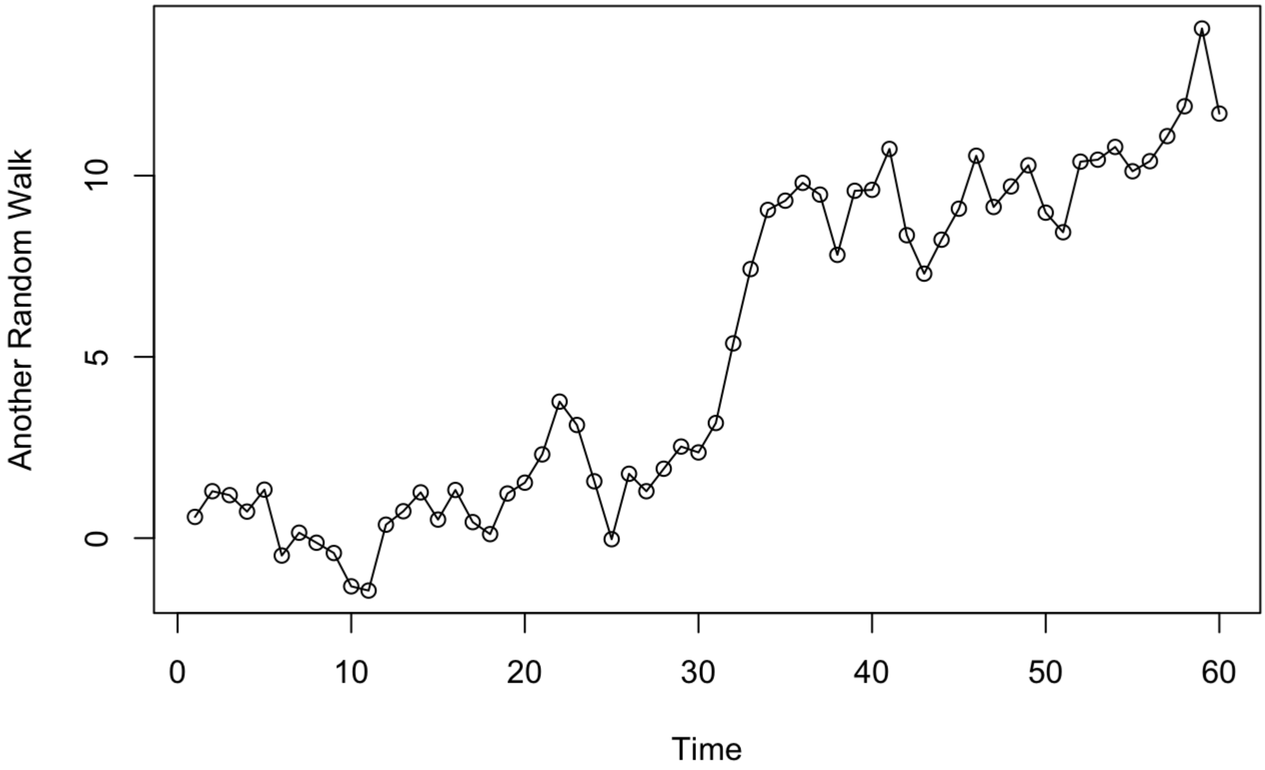
rwalk contains a simulated random walk

```
data(rwalk)
plot(rwalk,type='o',ylab='Random Walk')
```



R code for simulating a random walk with, say 60, independant standard normal errors

```
n=60
set.seed(12345)
sim.random.walk=ts(cumsum(rnorm(n)),freq=1,start=1)
plot(sim.random.walk,type='o',ylab='Another Random Walk')
```



Chapter 3 - R Commands

Exhibit 3.1

time(rwalk) yields a time series of the time epoches when the random walk was sampled.

```
data(rwalk)
model1=lm(rwalk~time(rwalk))
summary(model1)
```

```
##
## Call:
## lm(formula = rwalk ~ time(rwalk))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.70045 -0.79782  0.06391  0.63064  2.22128
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.007888   0.297245  -3.391  0.00126 **
## time(rwalk)  0.134087   0.008475  15.822  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.137 on 58 degrees of freedom
## Multiple R-squared:  0.8119, Adjusted R-squared:  0.8086
## F-statistic: 250.3 on 1 and 58 DF,  p-value: < 2.2e-16
```

Exhibit 3.2

rwalk contains a simulated random walk

```
plot(rwalk,type='o',ylab='y')
abline(model1) # add the fitted least squares line
```

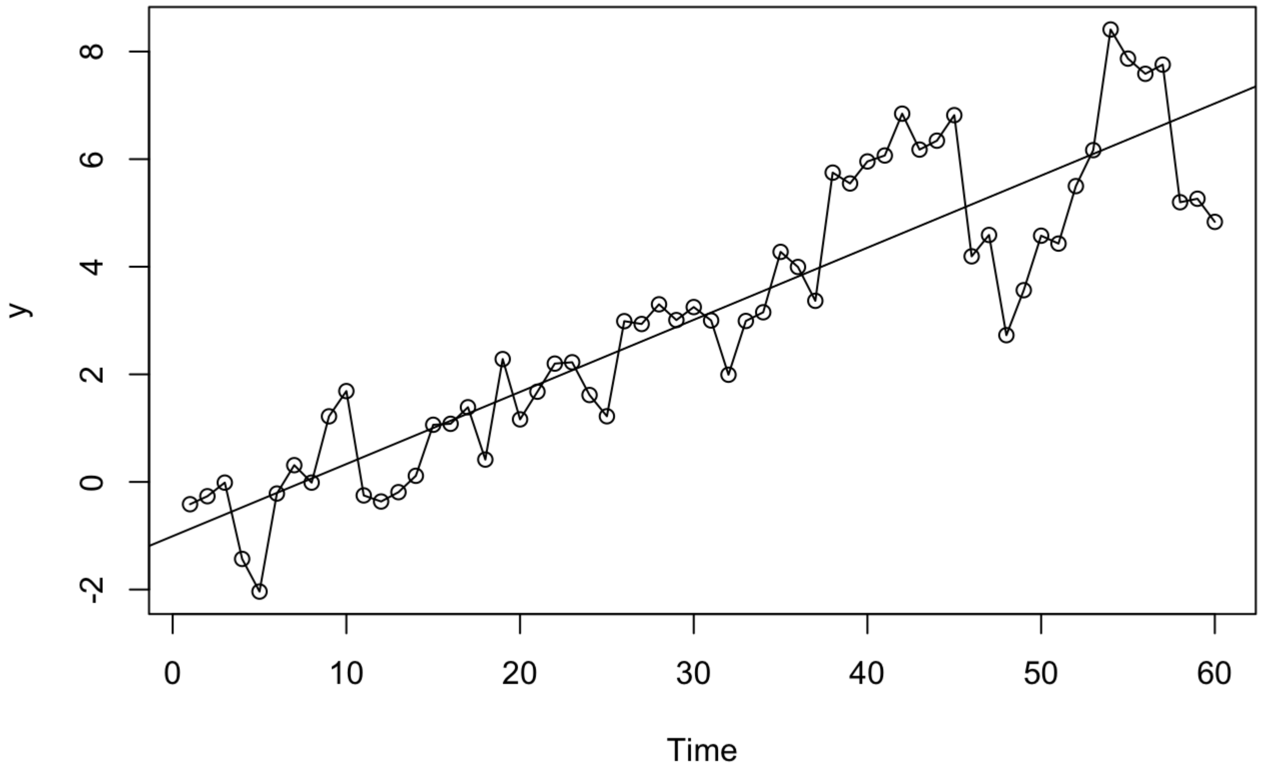


Exhibit 3.3

season(tempdub) creates a vector of the month index of the data as a factor

```
data(tempdub)
month.=season(tempdub) # the period sign is included to make the printout from
# the commands two line below clearer; ditto below.
model2=lm(tempdub~month.-1) # -1 removes the intercept term
summary(model2)
```

```
##
## Call:
## lm(formula = tempdub ~ month. - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.2750 -2.2479  0.1125  1.8896  9.8250
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## month.January     16.608      0.987   16.83   <2e-16 ***
## month.February    20.650      0.987   20.92   <2e-16 ***
## month.March       32.475      0.987   32.90   <2e-16 ***
## month.April       46.525      0.987   47.14   <2e-16 ***
## month.May         58.092      0.987   58.86   <2e-16 ***
## month.June        67.500      0.987   68.39   <2e-16 ***
## month.July        71.717      0.987   72.66   <2e-16 ***
## month.August      69.333      0.987   70.25   <2e-16 ***
## month.September   61.025      0.987   61.83   <2e-16 ***
## month.October     50.975      0.987   51.65   <2e-16 ***
## month.November    36.650      0.987   37.13   <2e-16 ***
## month.December    23.642      0.987   23.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.419 on 132 degrees of freedom
## Multiple R-squared:  0.9957, Adjusted R-squared:  0.9953
## F-statistic:  2569 on 12 and 132 DF,  p-value: < 2.2e-16
```

Exhibit 3.4

```
model3=lm(tempdub~month.) # intercept is automatically included so one month (Jan) is
dropped
summary(model3)
```

```
## 
## Call:
## lm(formula = tempdub ~ month.)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -8.2750 -2.2479  0.1125  1.8896  9.8250 
## 
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)    
## (Intercept)       16.608      0.987  16.828  < 2e-16 ***
## month.February     4.042      1.396   2.896  0.00443 ** 
## month.March       15.867      1.396  11.368  < 2e-16 ***
## month.April       29.917      1.396  21.434  < 2e-16 ***
## month.May         41.483      1.396  29.721  < 2e-16 ***
## month.June        50.892      1.396  36.461  < 2e-16 ***
## month.July        55.108      1.396  39.482  < 2e-16 ***
## month.August      52.725      1.396  37.775  < 2e-16 ***
## month.September   44.417      1.396  31.822  < 2e-16 ***
## month.October     34.367      1.396  24.622  < 2e-16 ***
## month.November    20.042      1.396  14.359  < 2e-16 ***
## month.December     7.033      1.396   5.039 1.51e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 3.419 on 132 degrees of freedom
## Multiple R-squared:  0.9712, Adjusted R-squared:  0.9688 
## F-statistic: 405.1 on 11 and 132 DF,  p-value: < 2.2e-16
```

Exhibit 3.5

first creates the first pair of harmonic functions and then fit the model

```
har.=harmonic(tempdub,1)
model4=lm(tempdub~har.)
summary(model4)
```

```
##
## Call:
## lm(formula = tempdub ~ har.)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.1580  -2.2756  -0.1457   2.3754  11.2671
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      46.2660     0.3088 149.816  < 2e-16 ***
## har.cos(2*pi*t) -26.7079     0.4367 -61.154  < 2e-16 ***
## har.sin(2*pi*t)  -2.1697     0.4367  -4.968 1.93e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.706 on 141 degrees of freedom
## Multiple R-squared:  0.9639, Adjusted R-squared:  0.9634
## F-statistic:  1882 on 2 and 141 DF,  p-value: < 2.2e-16
```

Exhibit 3.6

```
plot(ts(fitted(model4),freq=12,start=c(1964,1)),ylab='Temperature',type='l',
ylim=range(c(fitted(model4),tempdub))) # the ylim option ensures that the
# y axis has a range that fits the raw data and the fitted values
points(tempdub)
```

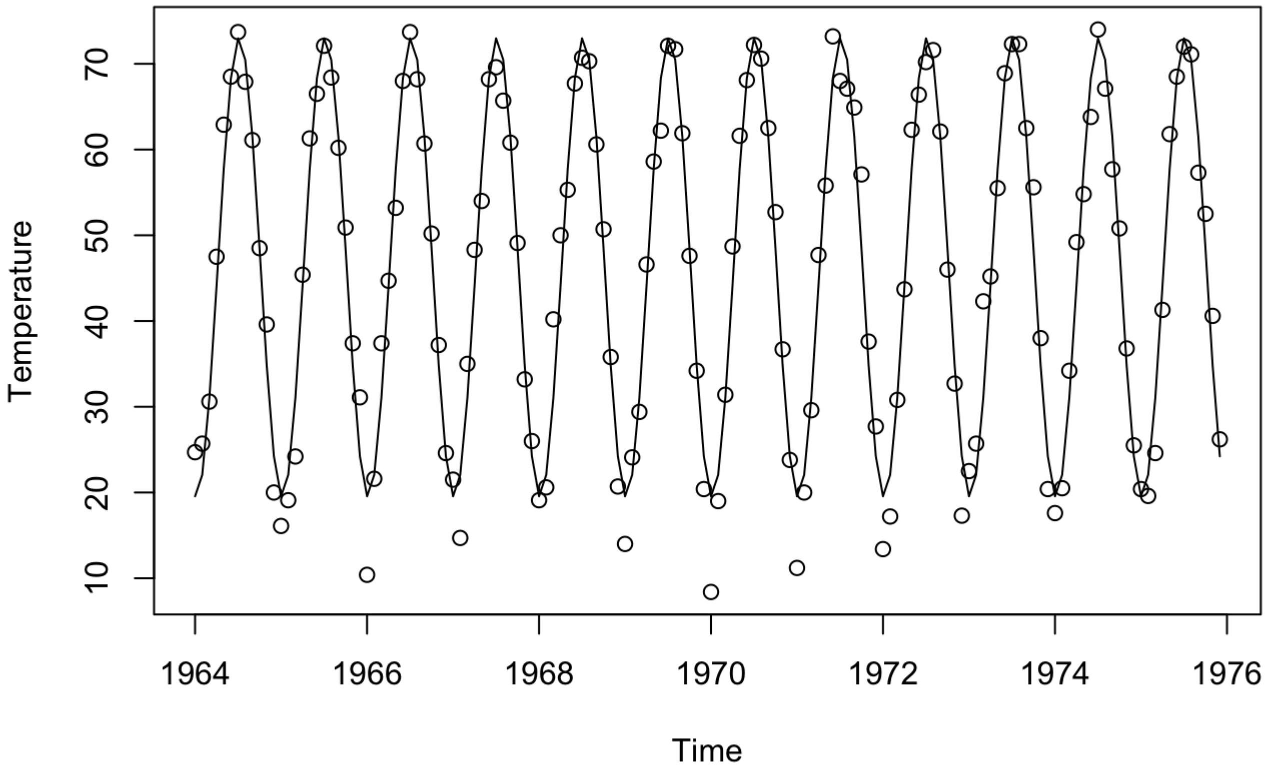


Exhibit 3.7

```
data(rwalk)
model1=lm(rwalk~time(rwalk))
summary(model1)
```

```
##
## Call:
## lm(formula = rwalk ~ time(rwalk))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.70045 -0.79782  0.06391  0.63064  2.22128
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.007888   0.297245  -3.391  0.00126 **
## time(rwalk)  0.134087   0.008475  15.822  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.137 on 58 degrees of freedom
## Multiple R-squared:  0.8119, Adjusted R-squared:  0.8086
## F-statistic: 250.3 on 1 and 58 DF,  p-value: < 2.2e-16
```

Exhibit 3.8

```
plot(y=rstudent(model3),x=as.vector(time(tempdub)),xlab='Time',
ylab='Standardized Residuals',type='o')
```

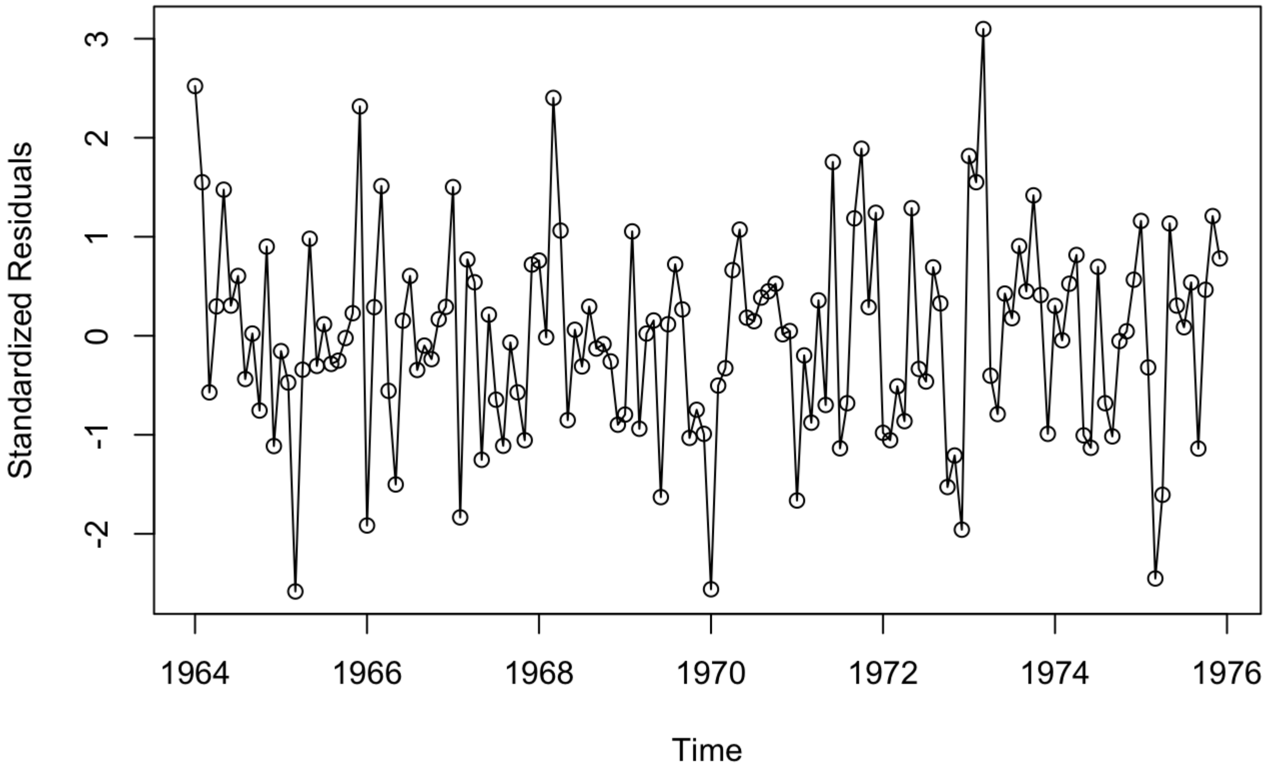


Exhibit 3.9

```
plot(y=rstudent(model3),x=as.vector(time(tempdub)),xlab='Time',
ylab='Standardized Residuals',type='l')
points(y=rstudent(model3),x=as.vector(time(tempdub)),
pch=as.vector(season(tempdub)))
```

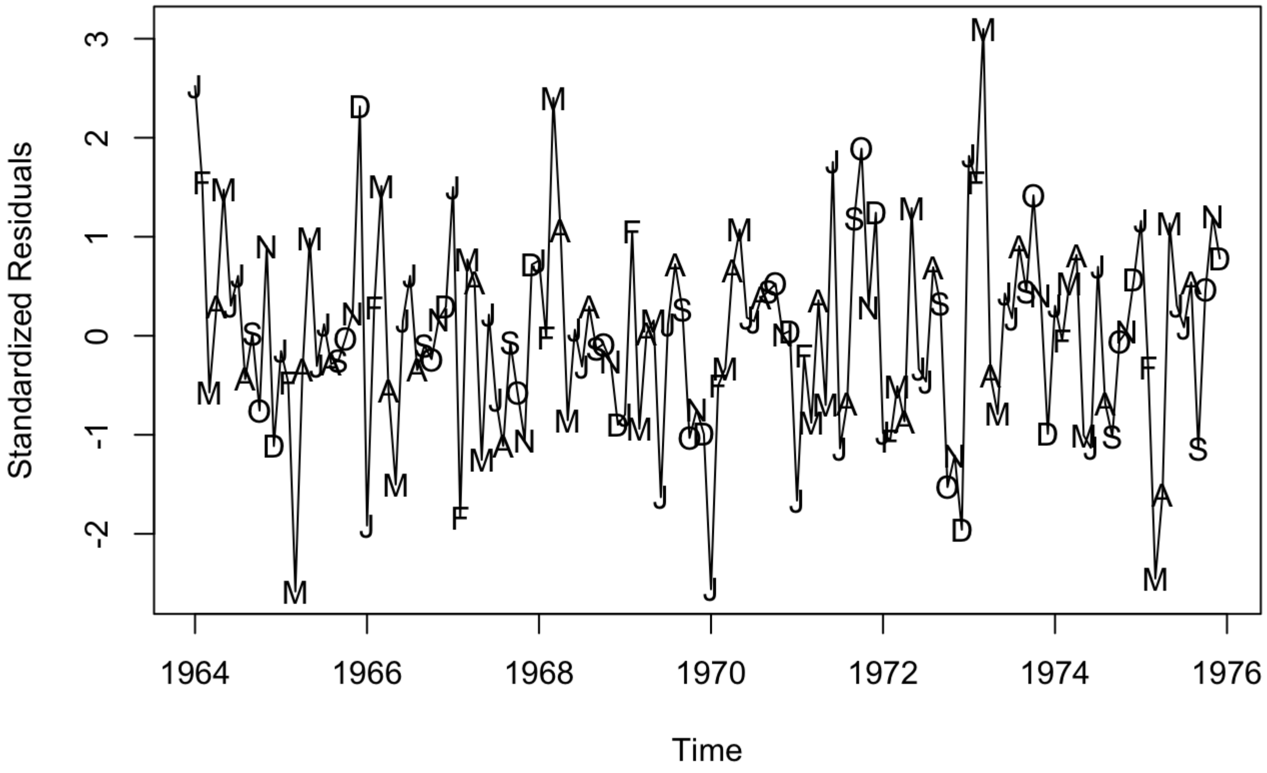


Exhibit 3.10

```
plot(y=rstudent(model3),x=as.vector(fitted(model3)),xlab='Fitted Trend Values',
ylab='Standardized Residuals',type="n")
points(y=rstudent(model3),x=as.vector(fitted(model3)),
pch=as.vector(season(tempdub)))
```

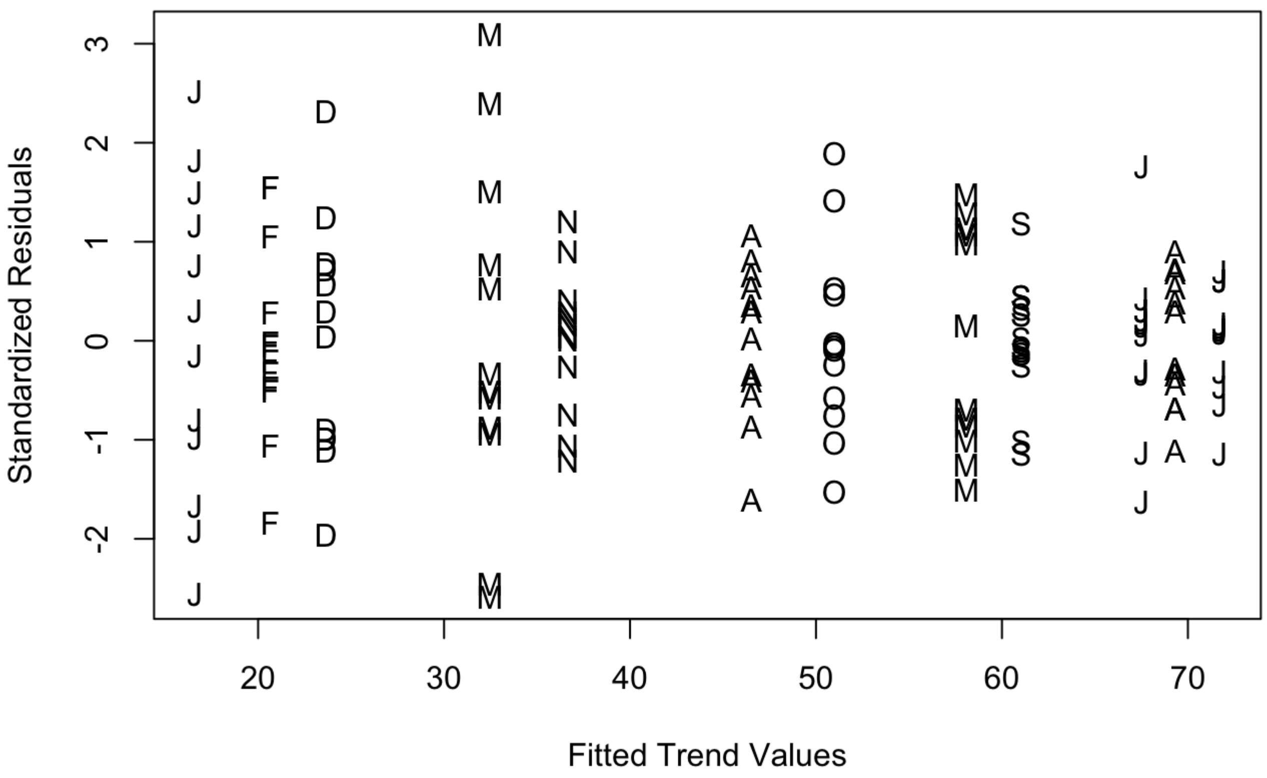


Exhibit 3.11

```
hist(rstudent(model3),xlab='Standardized Residuals',main='')
```

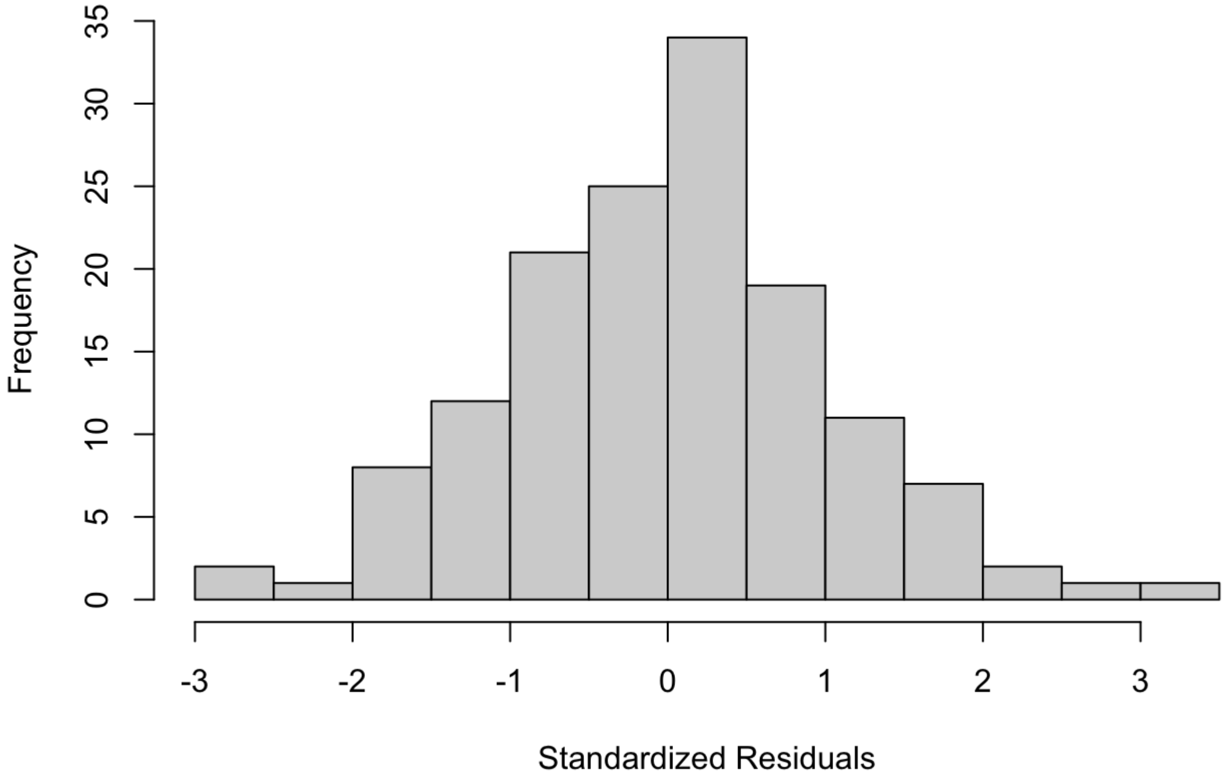


Exhibit 3.12

```
qqnorm(rstudent(model3),main='')
```

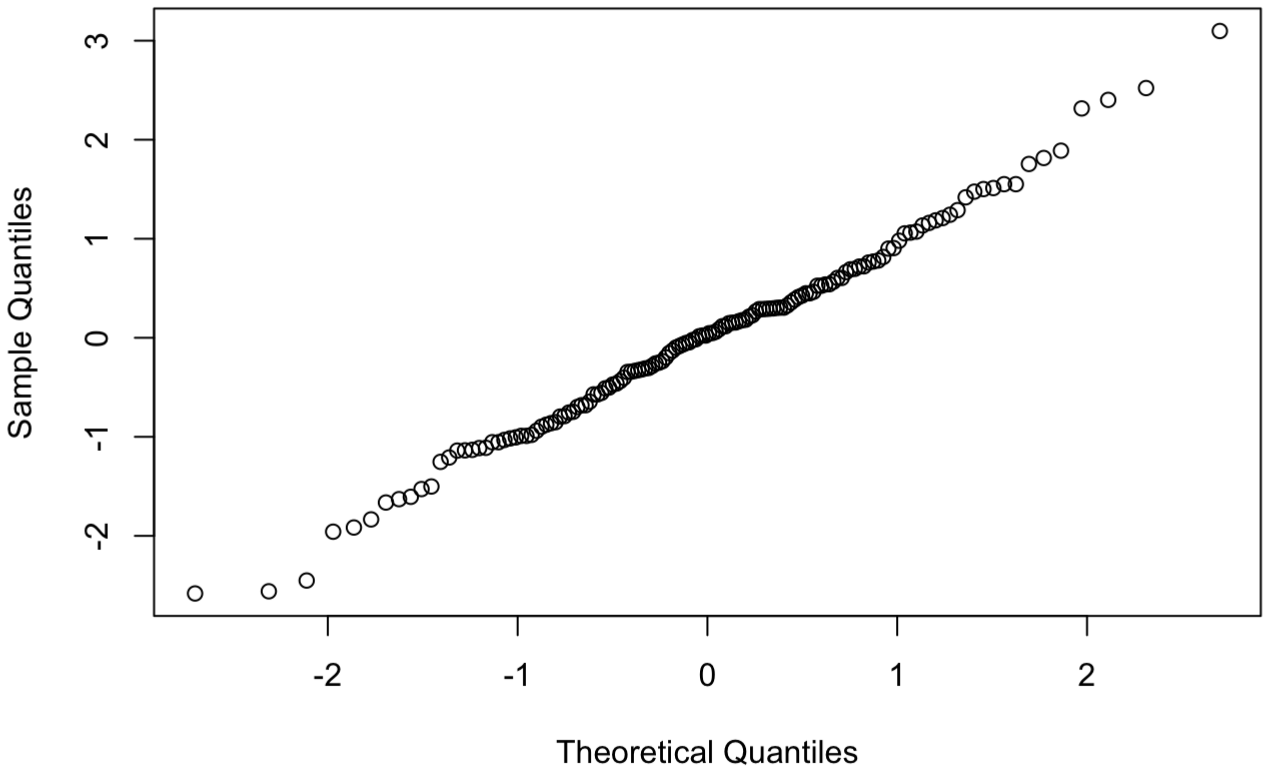


Exhibit 3.13

```
acf(rstudent(model3),main='')
```

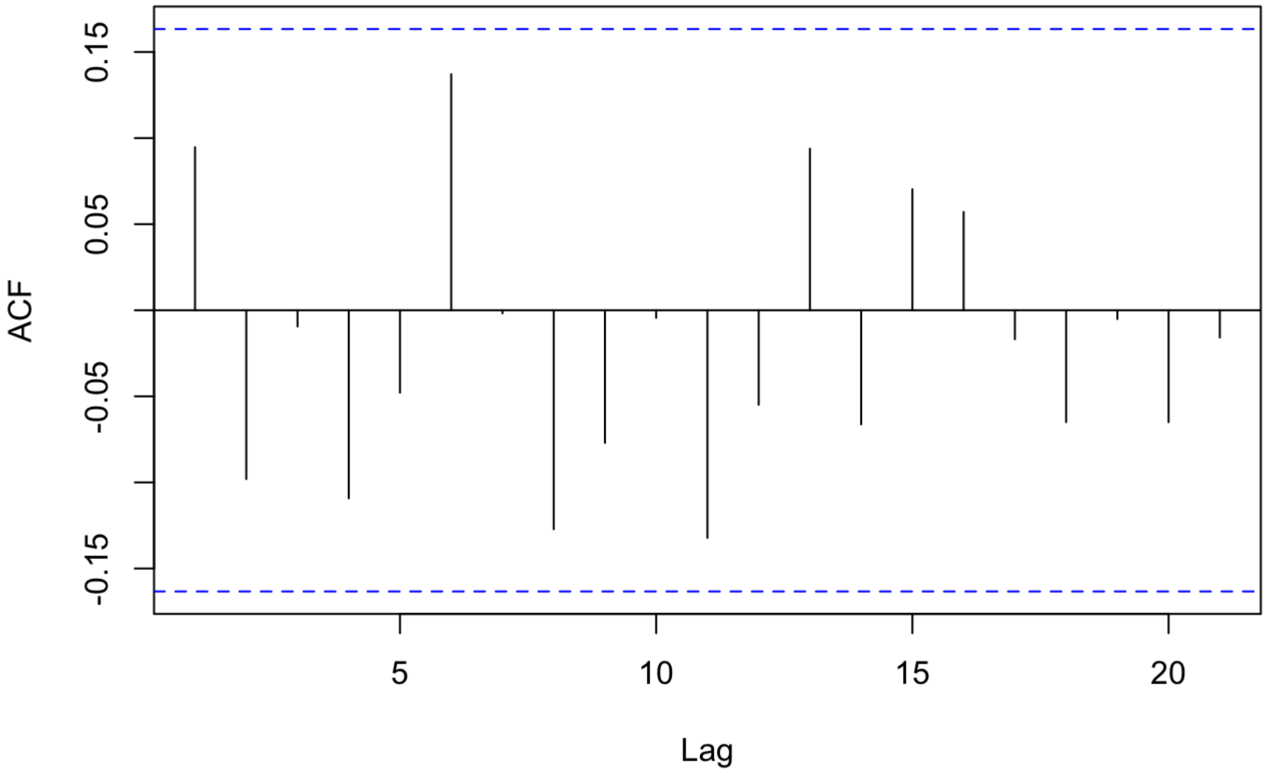


Exhibit 3.14

```
plot(y=rstudent(model1),x=as.vector(time(rwalk)),ylab='Standardized Residuals',
xlab='Time',type='o')
```

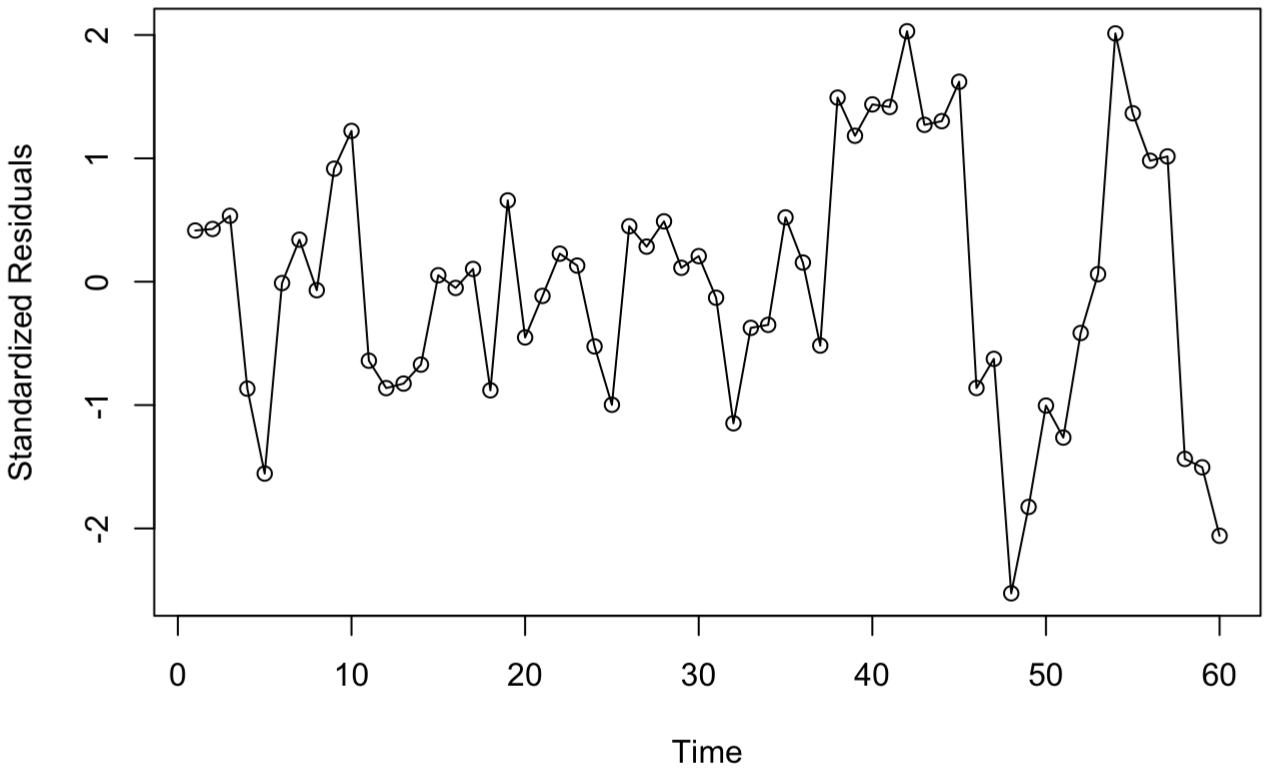


Exhibit 3.15

```
plot(y=rstudent(model1),x=fitted(model1),ylab='Standardized Residuals',
xlab='Fitted Trend Values',type='p')
```

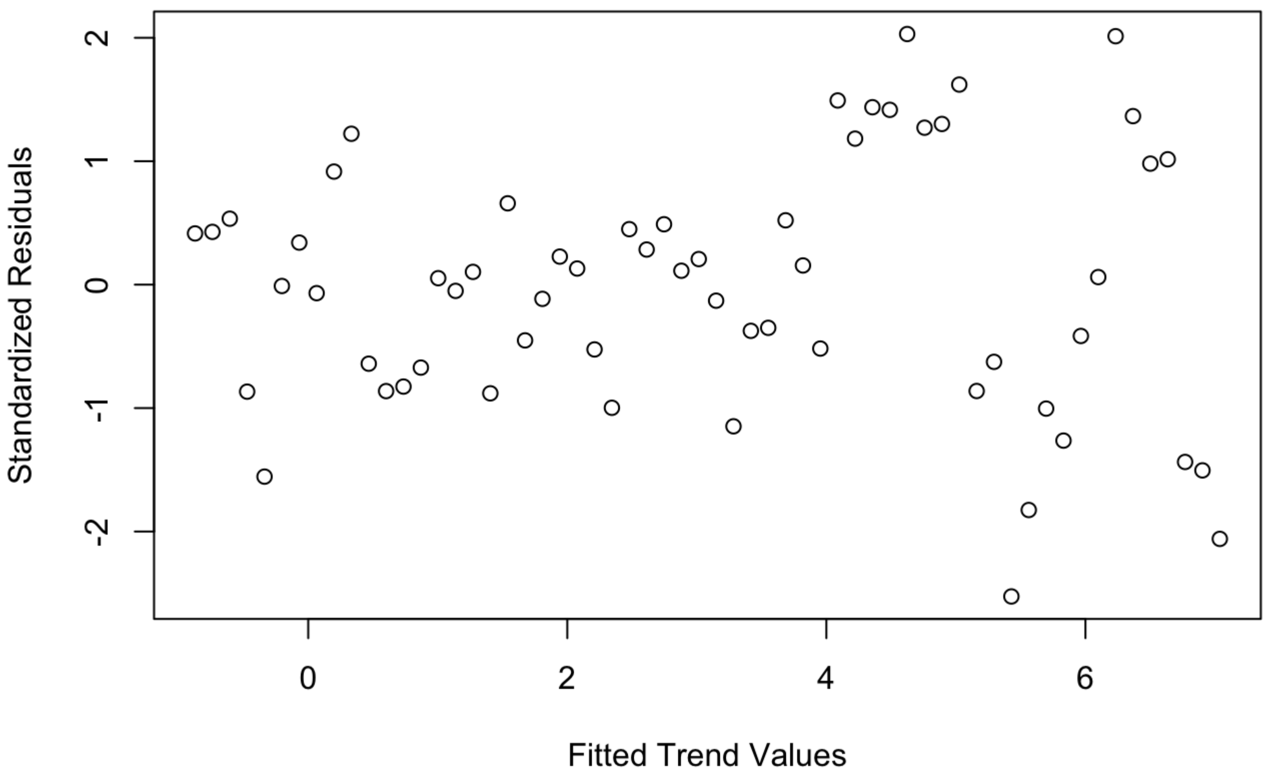
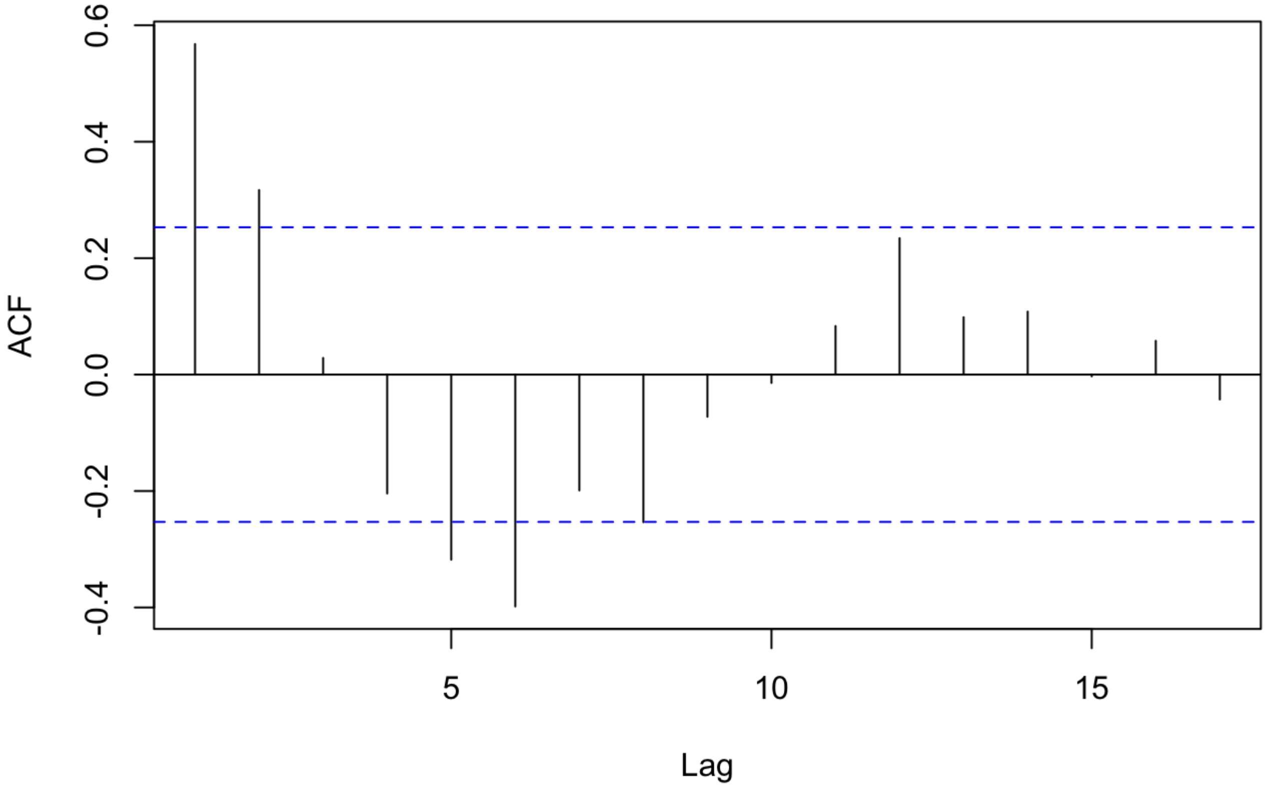


Exhibit 3.16

```
acf(rstudent(model1),main='')
```

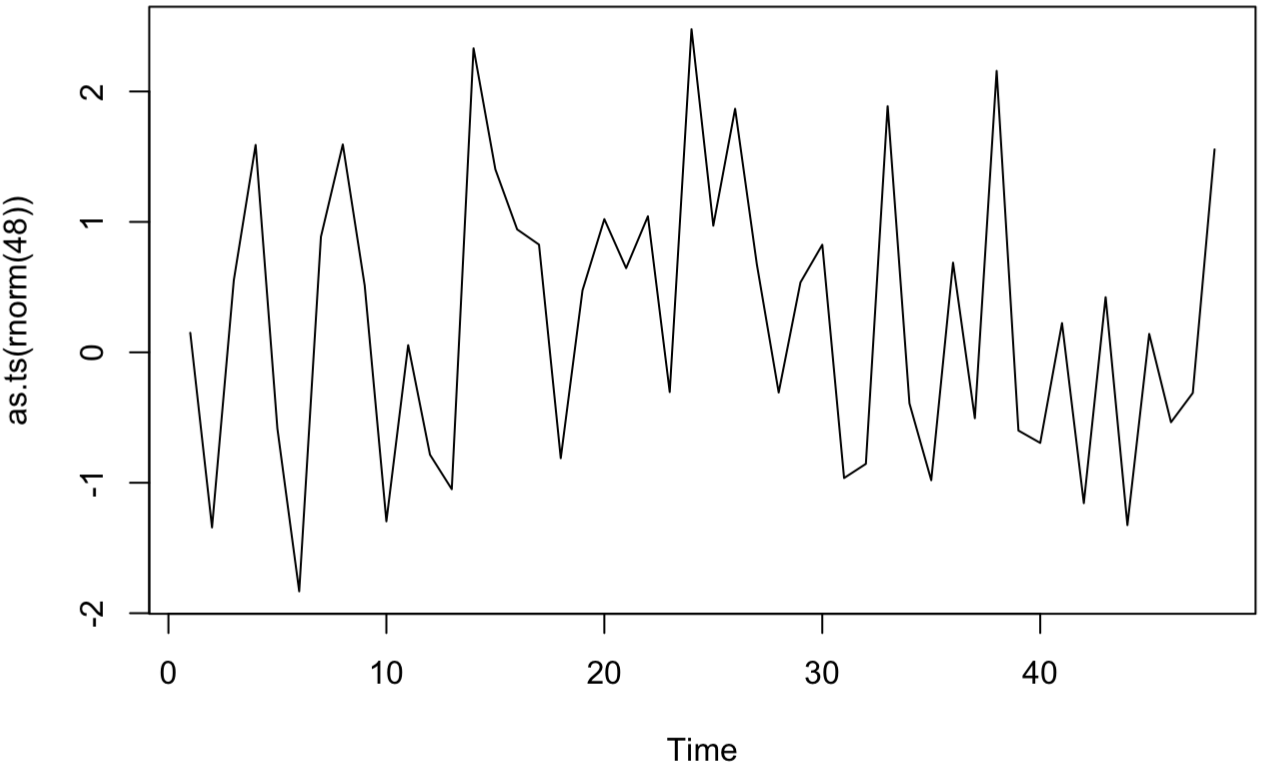


Exercise 1

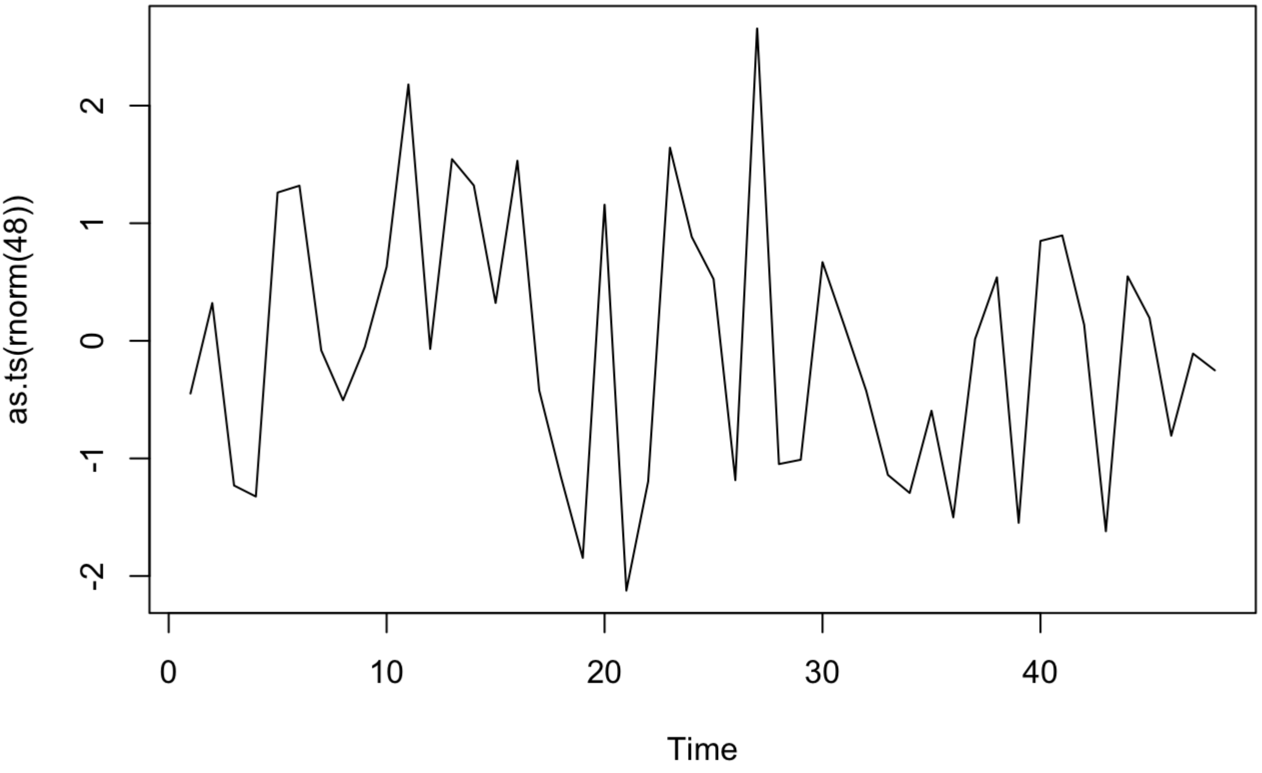
1.3

Answer - The plots are random.

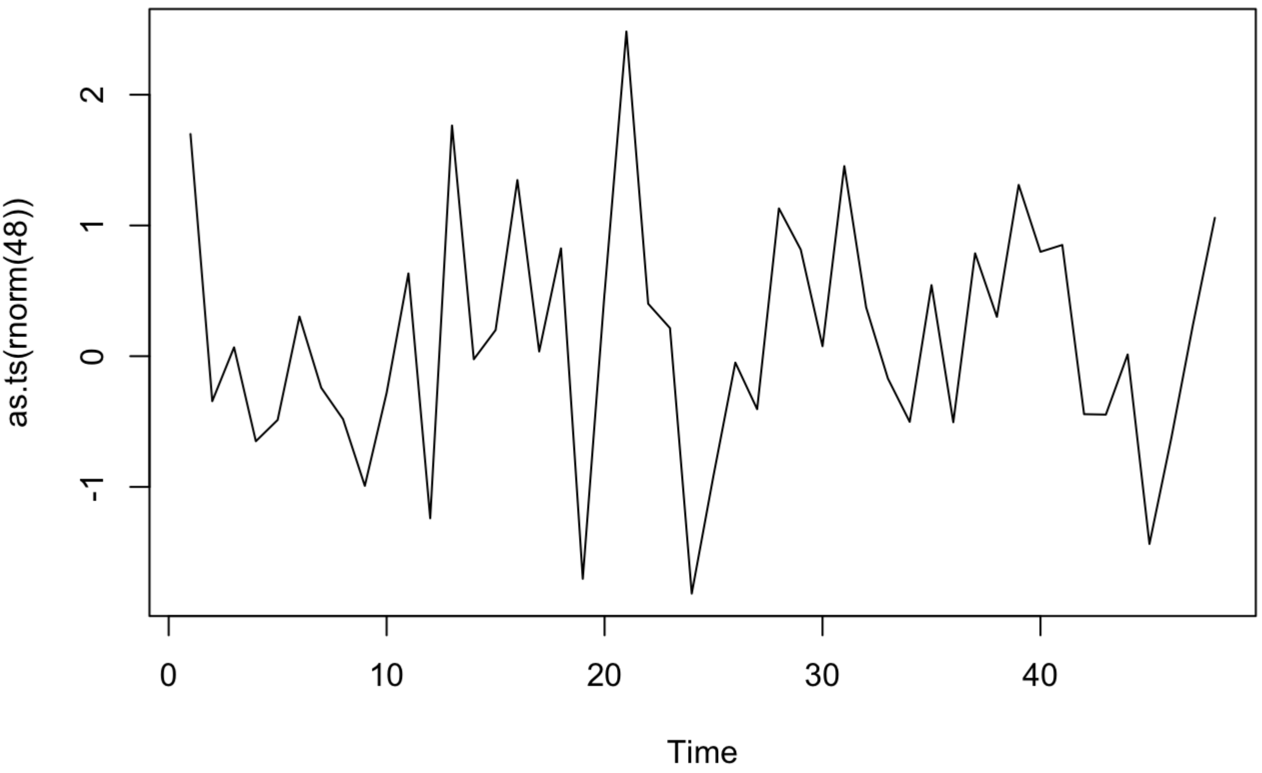
```
plot(as.ts(rnorm(48)))
```



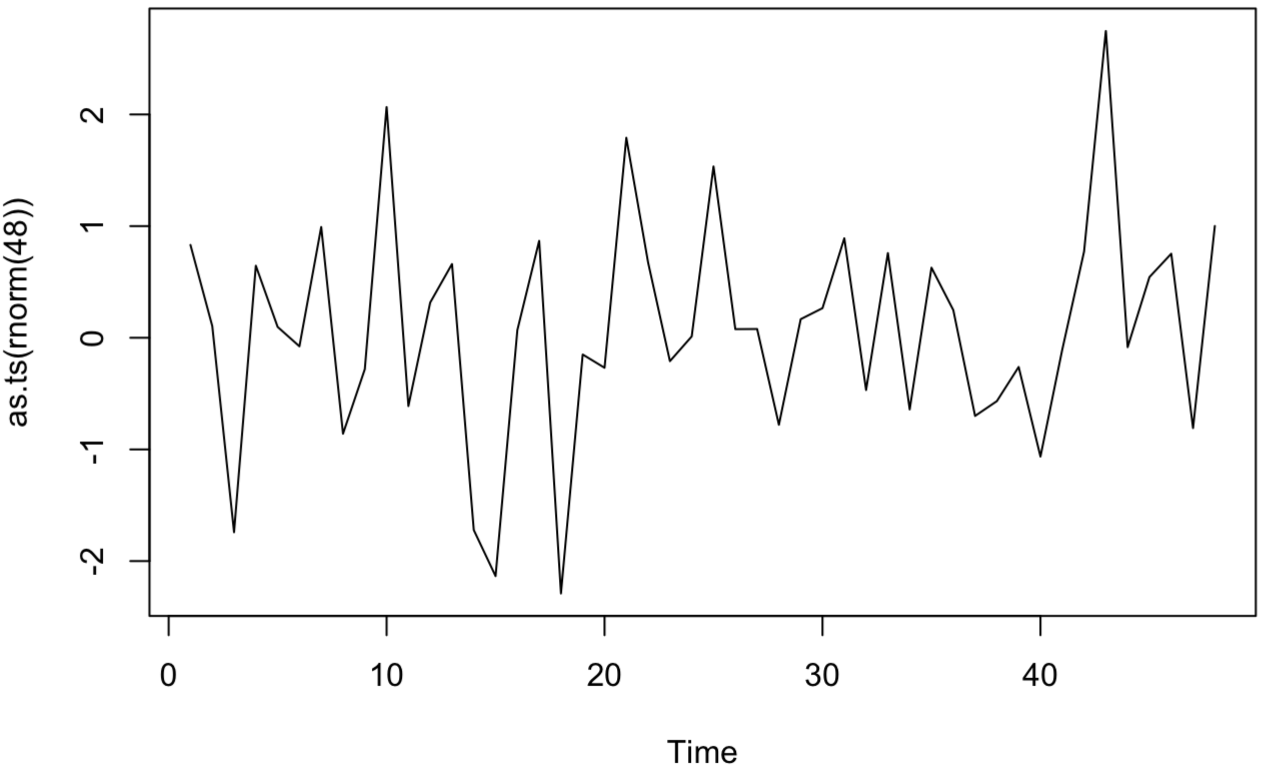
```
plot(as.ts(rnorm(48)))
```



```
plot(as.ts(rnorm(48)))
```



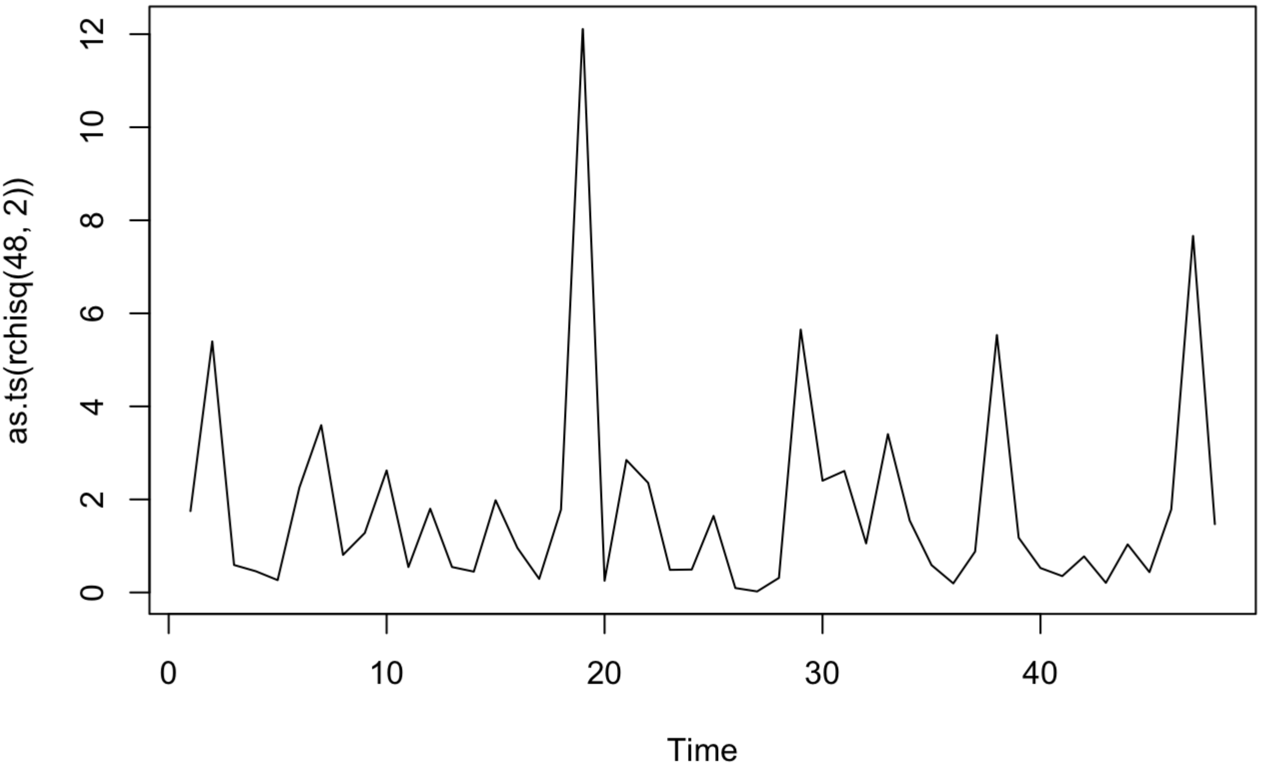
```
plot(as.ts(rnorm(48)))
```



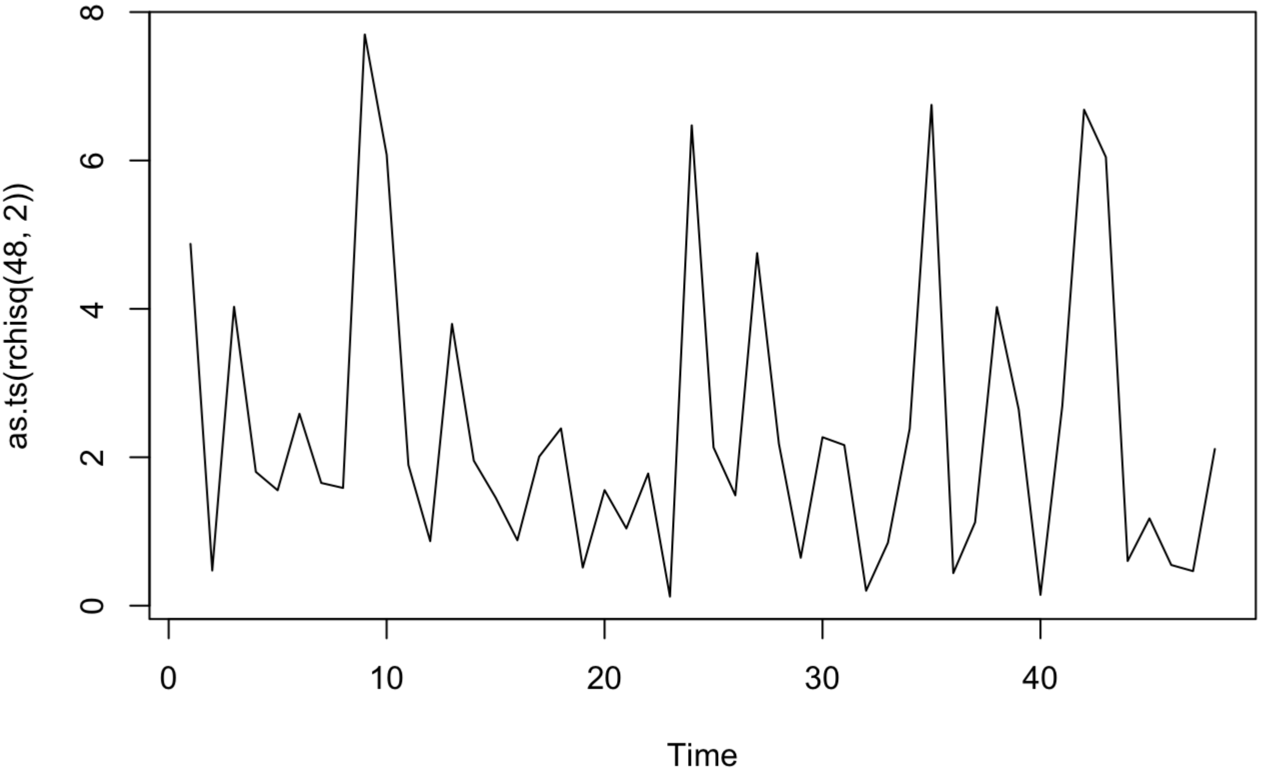
1.4

Answer - Plots are random and non-normal.

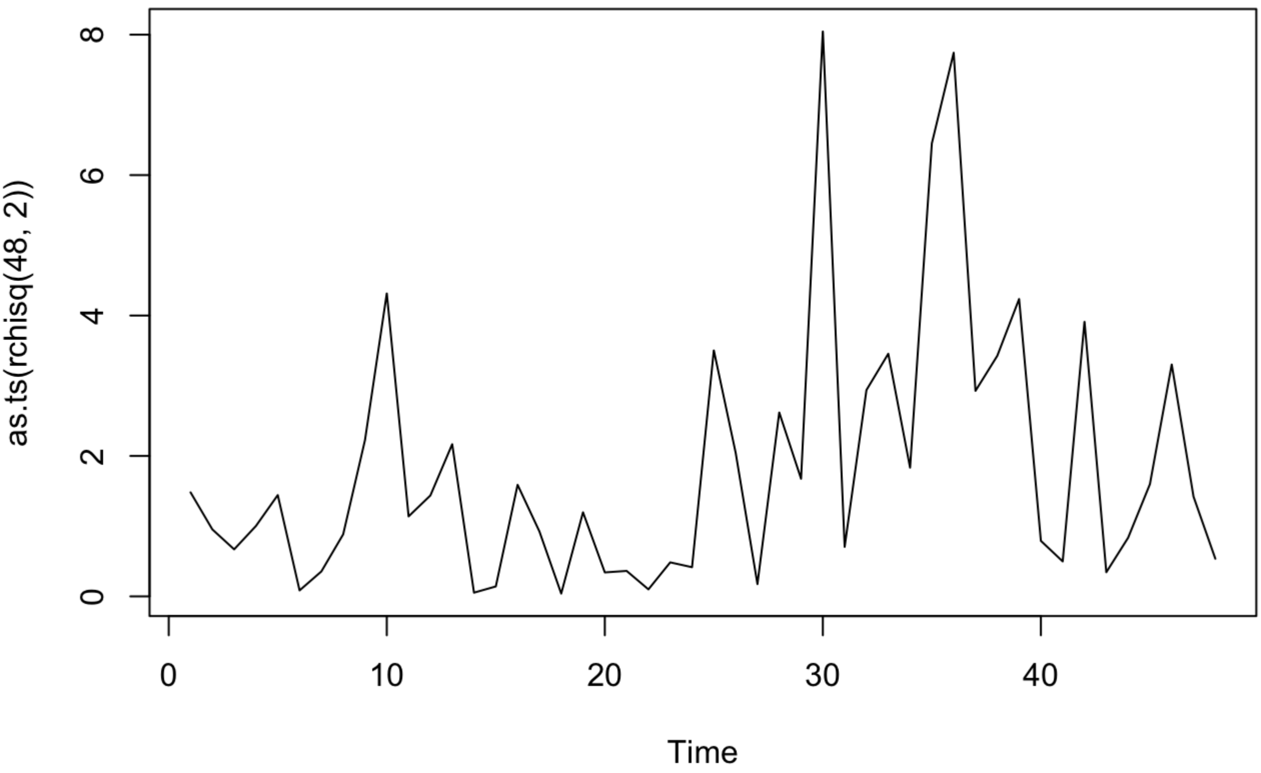
```
plot(as.ts(rchisq(48, 2)))
```



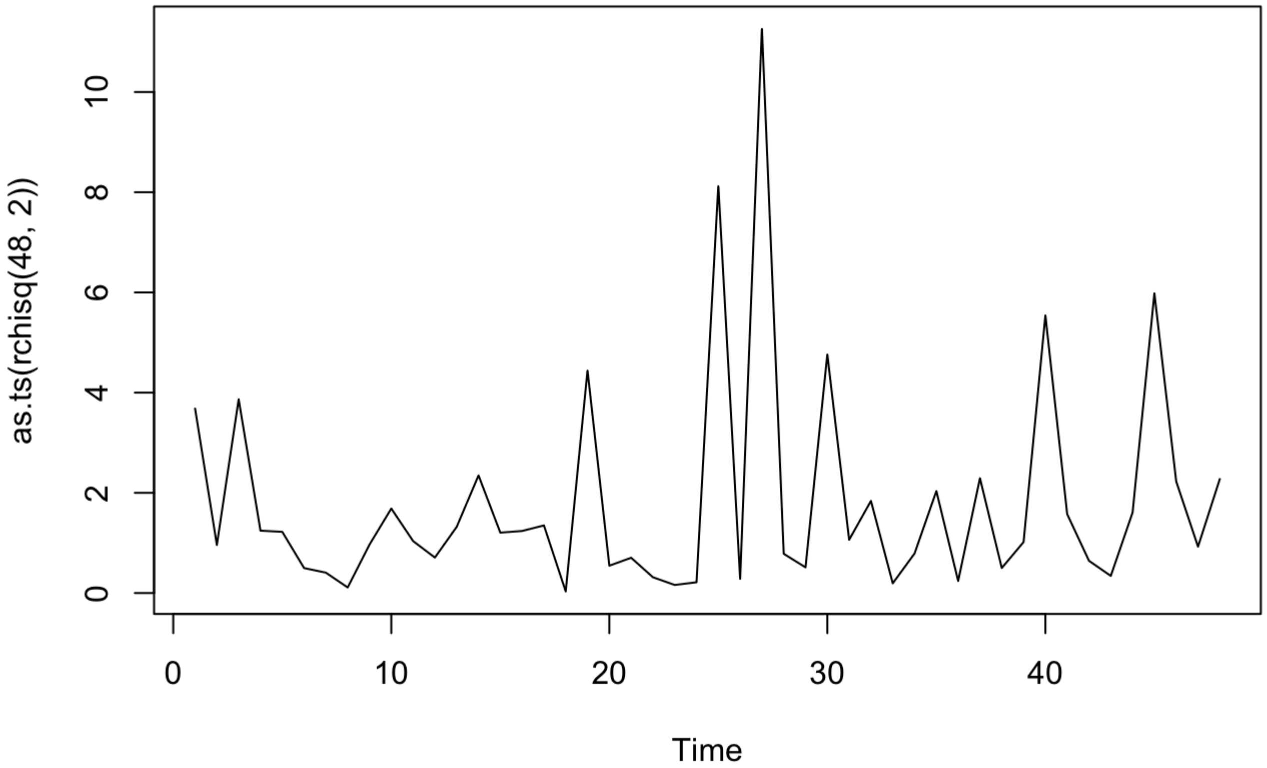
```
plot(as.ts(rchisq(48, 2)))
```



```
plot(as.ts(rchisq(48, 2)))
```



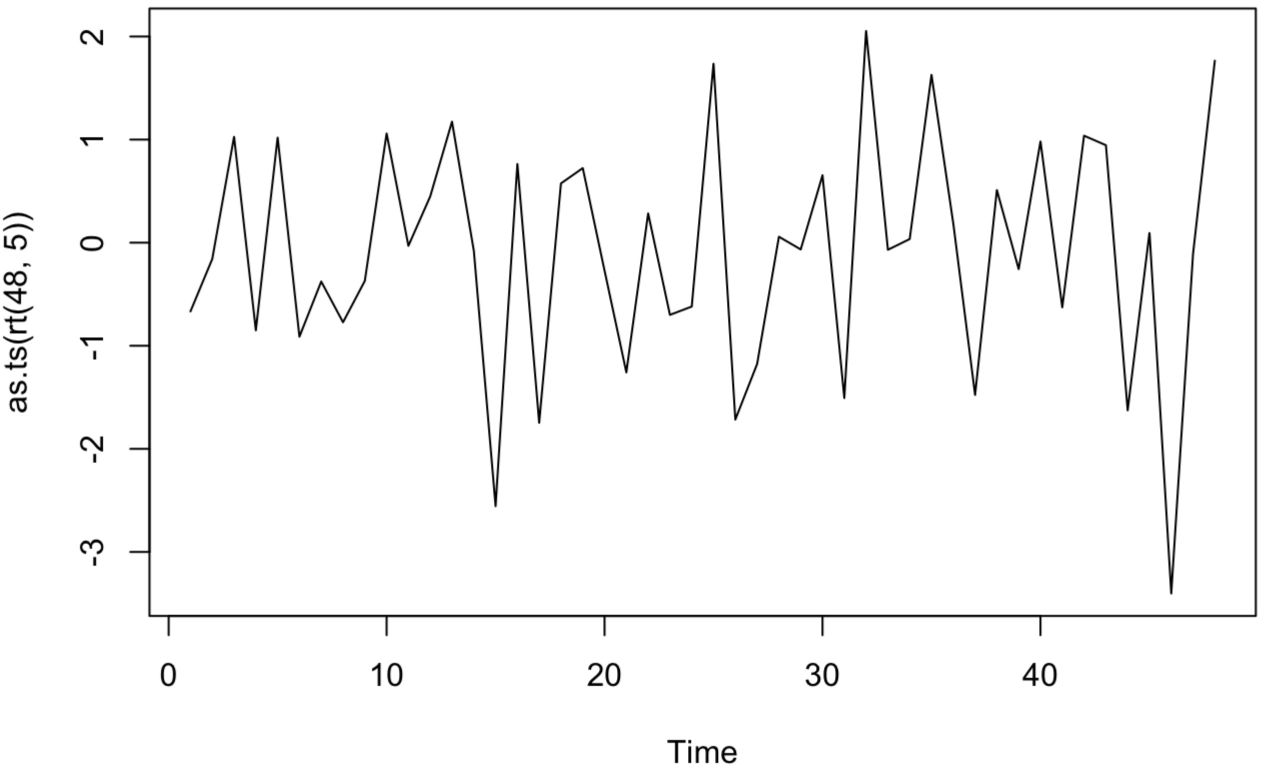
```
plot(as.ts(rchisq(48, 2)))
```



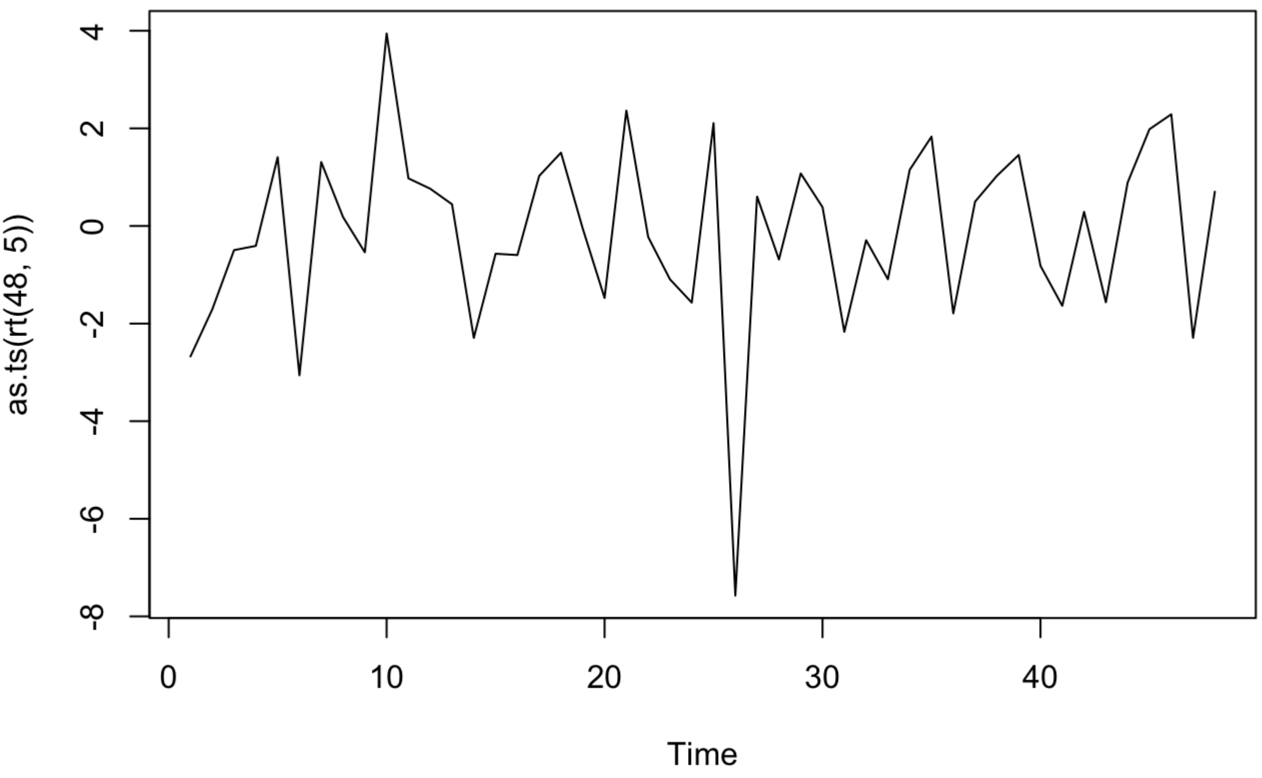
1.5

Answer - Plots look to be random and non-normal.

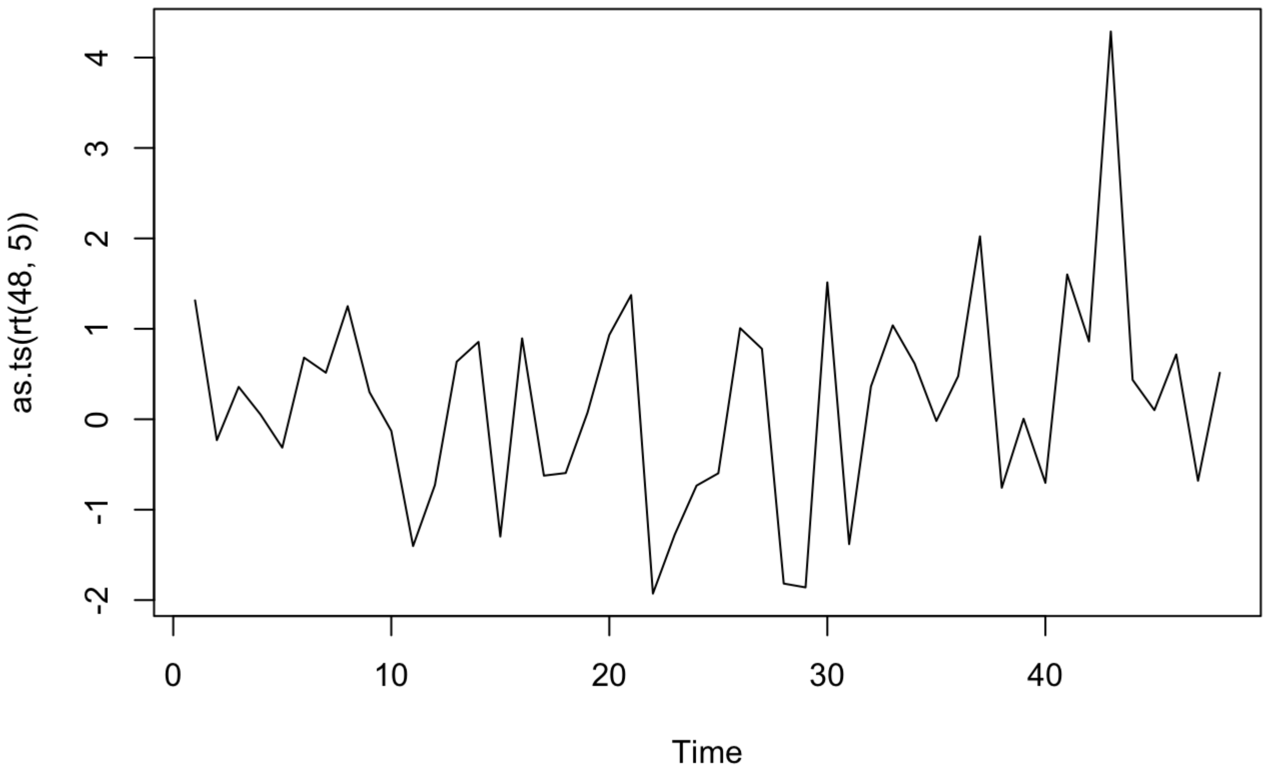
```
plot(as.ts(rt(48, 5)))
```



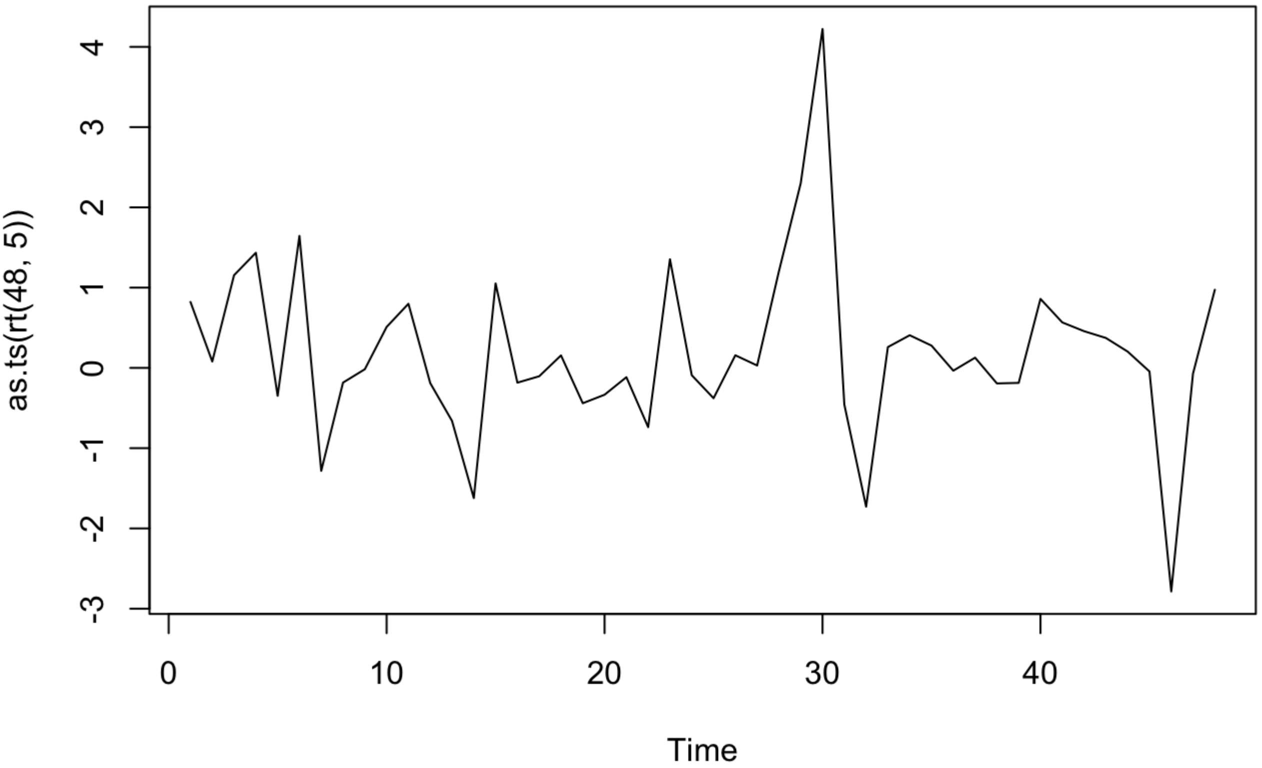
```
plot(as.ts(rt(48, 5)))
```



```
plot(as.ts(rt(48, 5)))
```



```
plot(as.ts(rt(48, 5)))
```


as.ts(rt(48, 5))
Time
-3
-2
-1
0
1
2
3
4
0
10
20
30
40