Problem Solving Set 4

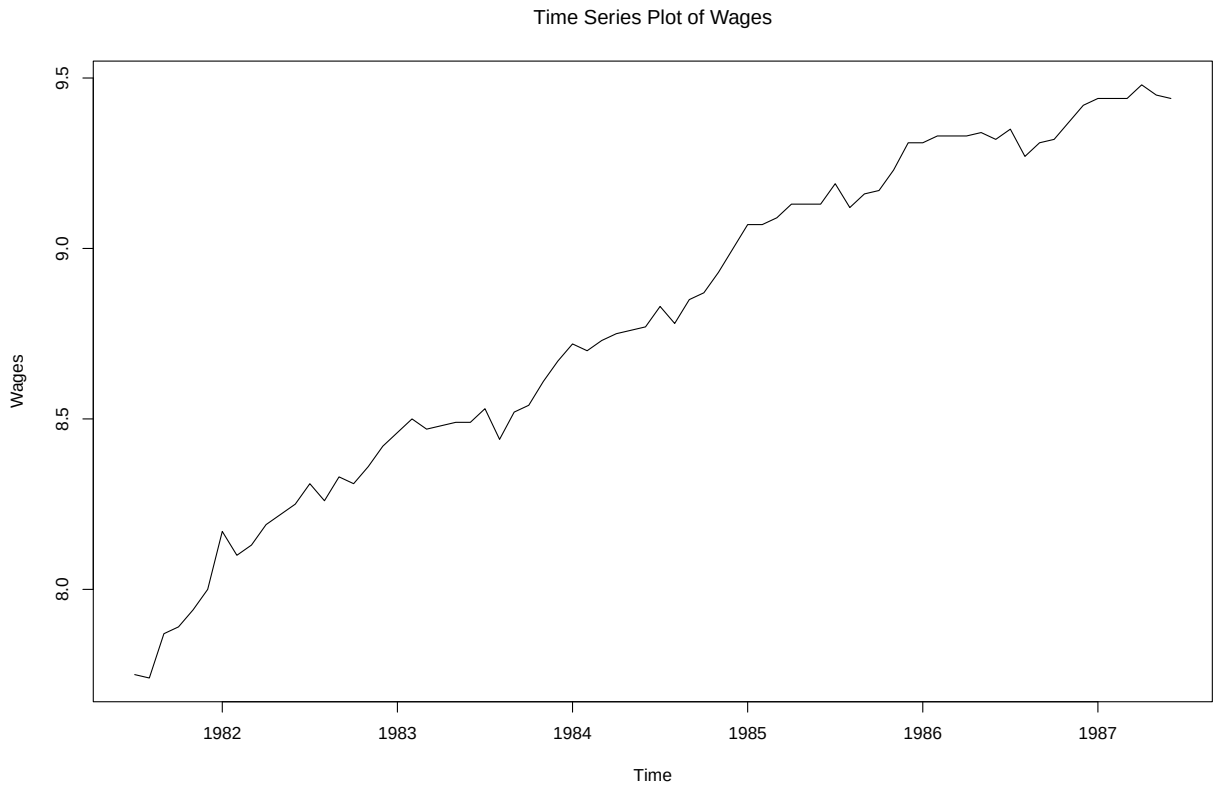
Sven Bergmann

February 12, 2024

```
library(TSA)
library(tseries)
library(forecast)
```

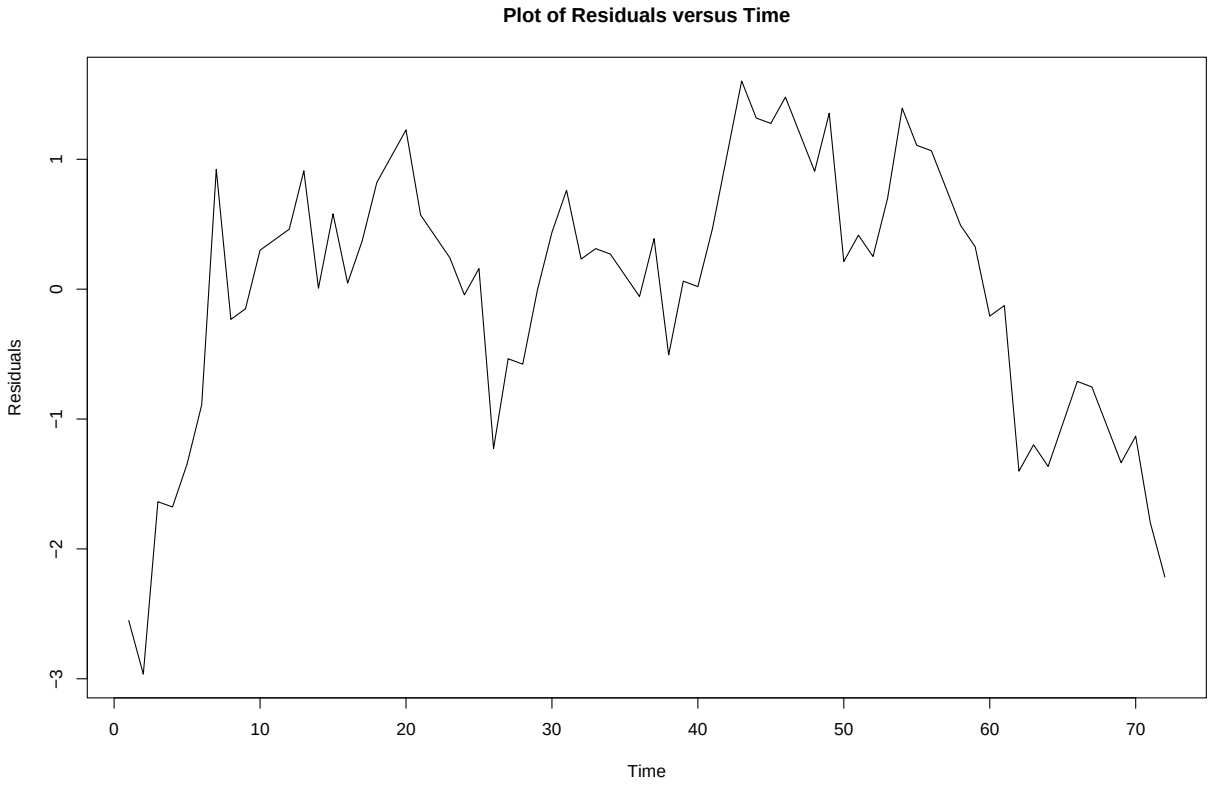
Problem 3

```
data(wages)
plot(wages, xlab = expression("Time"), ylab = expression("Wages"),
    main = expression("Time Series Plot of Wages"))
```

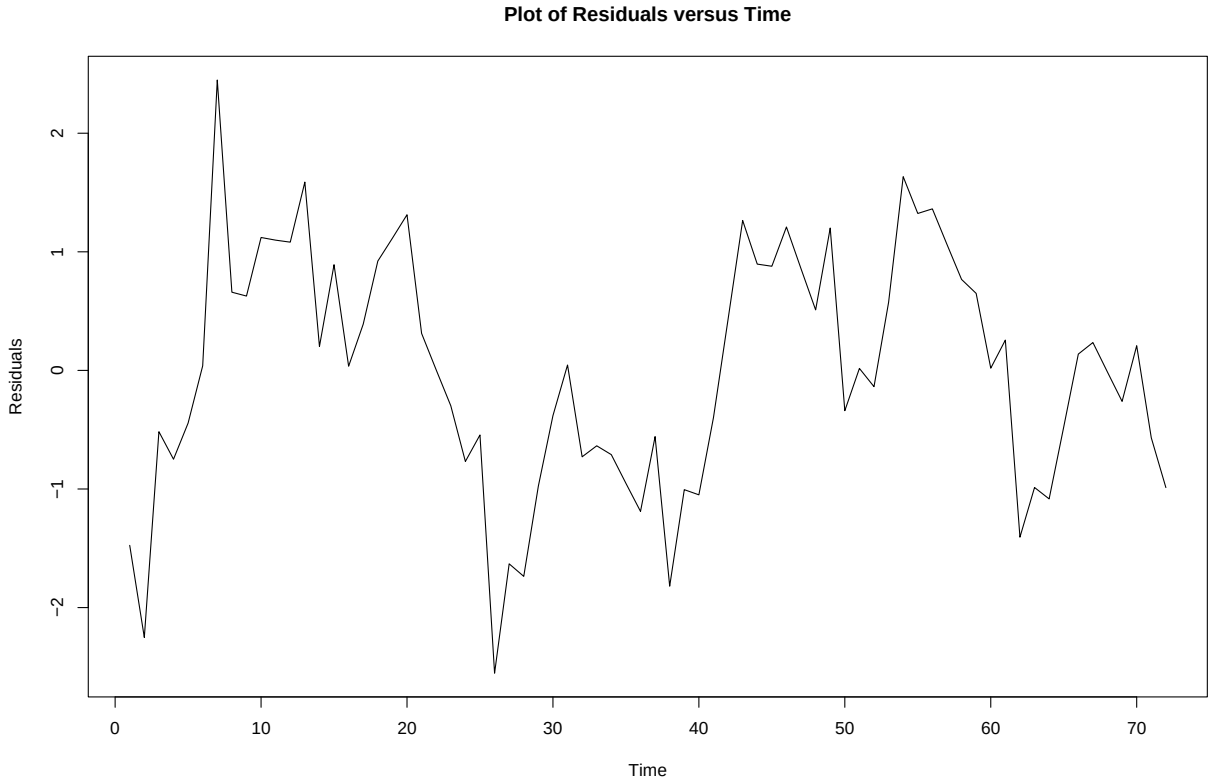


Detrend

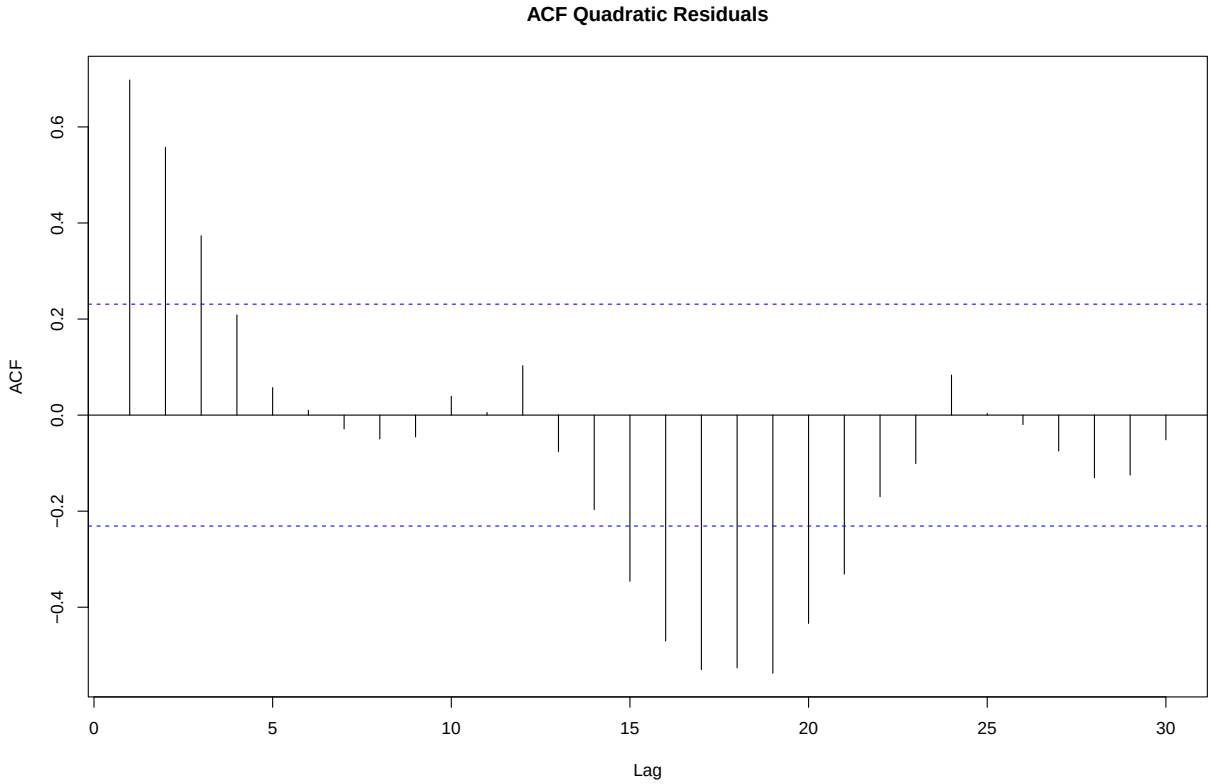
```
model1 <- lm(wages ~ time(wages))
res <- as.ts(rstandard(model1))
plot(res, xlab = expression("Time"), ylab = expression("Residuals"),
    main = "Plot of Residuals versus Time")
```




```
sq <- time(wages)^2
model2 <- lm(wages ~ sq + time(wages))
res2 <- as.ts(rstandard(model2))
plot(res2, xlab = expression("Time"), ylab = expression("Residuals"),
    main = "Plot of Residuals versus Time")
```



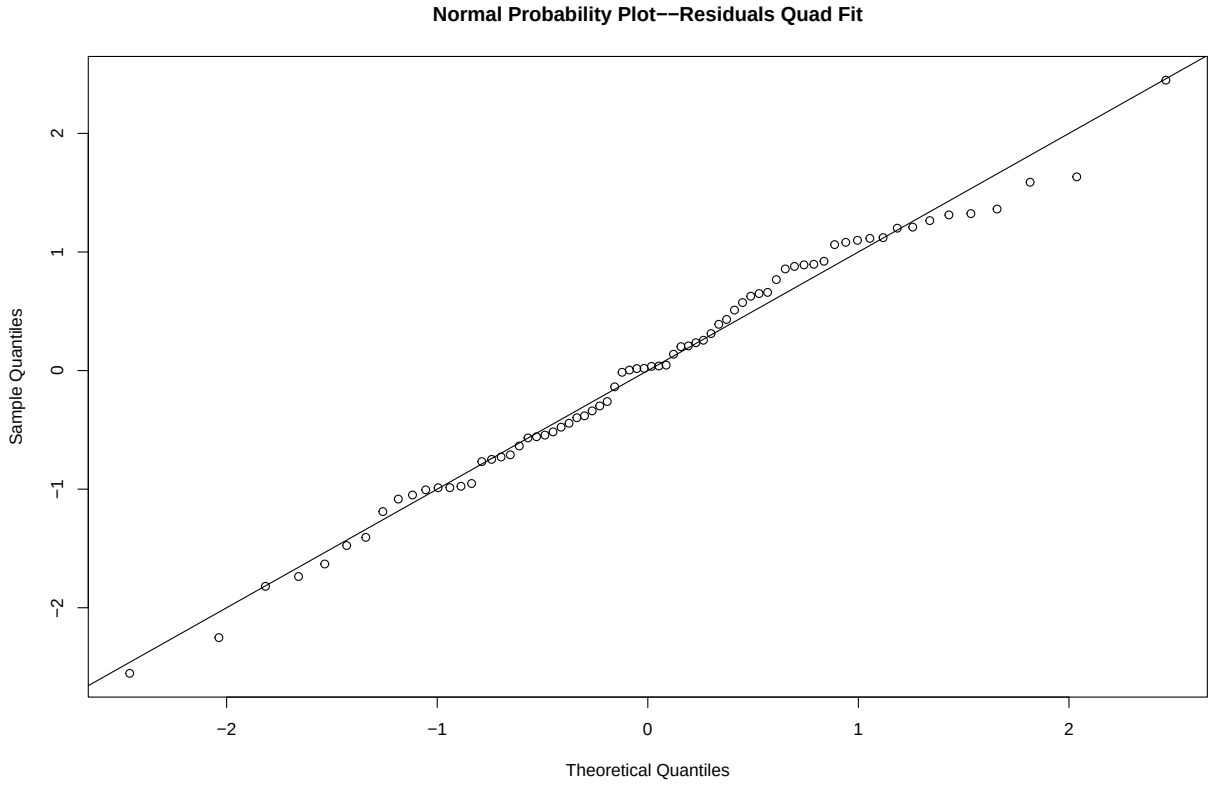

```
acf(res2, lag.max = 30, main = "ACF Quadratic Residuals")$acf
```



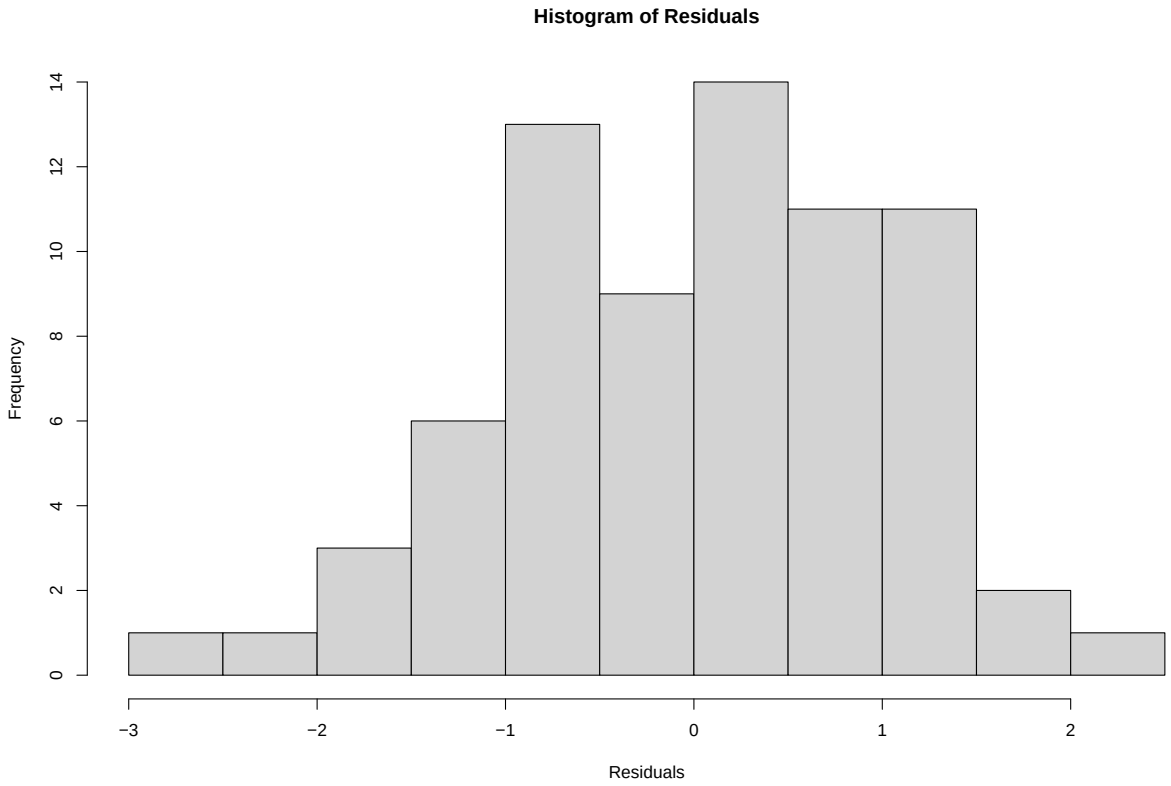

```
## , , 1
##
##               [,1]
##  [1,]  0.697678669
##  [2,]  0.557745779
##  [3,]  0.373452096
##  [4,]  0.208637107
##  [5,]  0.057231746
##  [6,]  0.009739095
##  [7,] -0.028833668
##  [8,] -0.049714986
##  [9,] -0.045381368
## [10,]  0.039285113
## [11,]  0.005610692
## [12,]  0.102837386
## [13,] -0.076188597
## [14,] -0.196688177
## [15,] -0.346062745
## [16,] -0.470227582
## [17,] -0.529751323
## [18,] -0.526201183
## [19,] -0.537344532
## [20,] -0.433720543
## [21,] -0.331081130
## [22,] -0.170037300
## [23,] -0.100865087
## [24,]  0.083418566
```

```
## [25,]  0.003186551
## [26,] -0.019794694
## [27,] -0.074922076
## [28,] -0.130297152
## [29,] -0.124916160
## [30,] -0.051461820
```

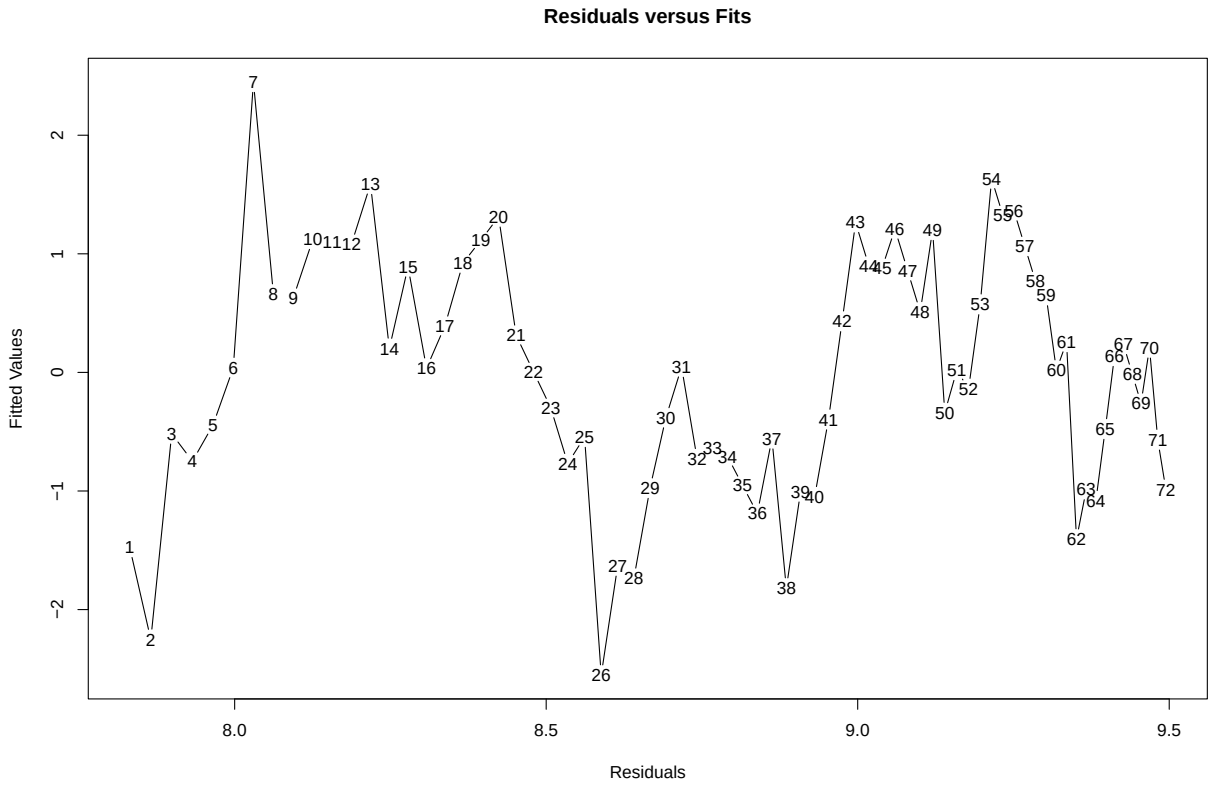
```r
# par(mfrow=c(2,2))
qqnorm(res2, main = "Normal Probability Plot--Residuals Quad Fit")
abline(a = 0, b = 1)
```




```r
hist(res2, xlab = "Residuals", main = "Histogram of Residuals")
```



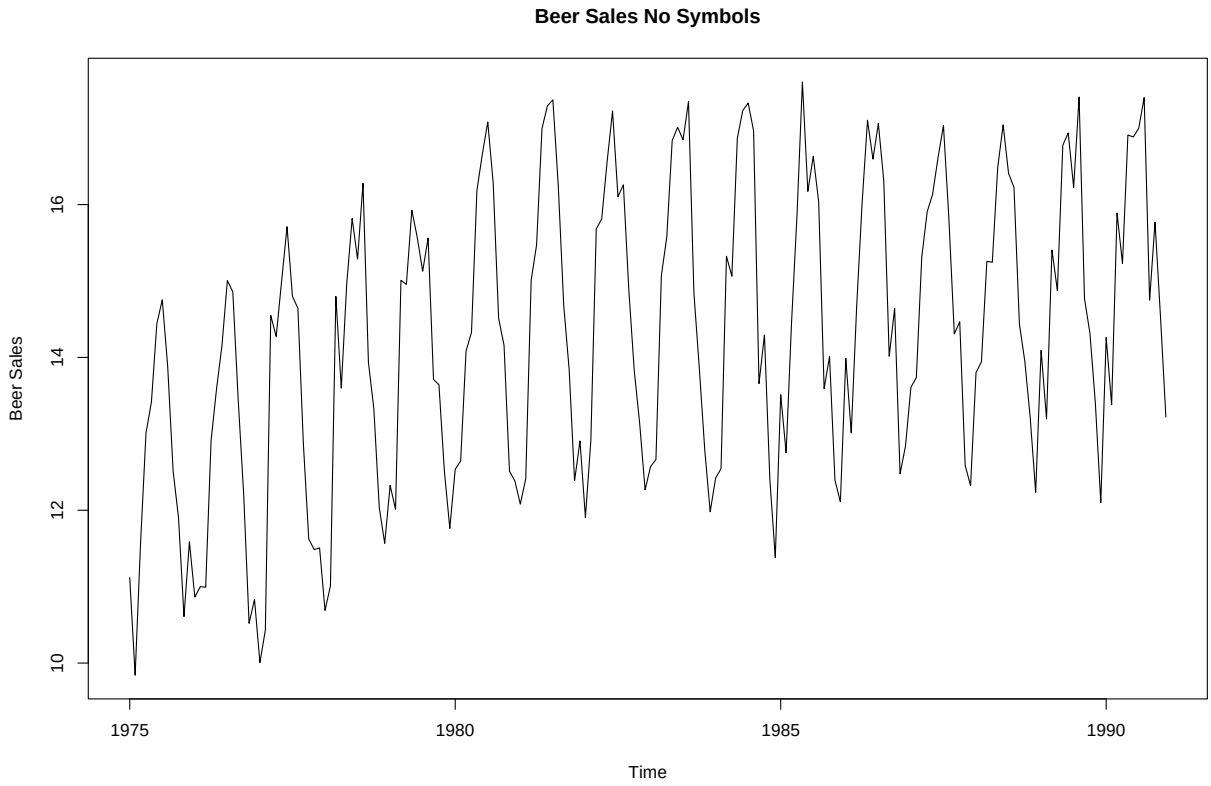

```
fits <- as.ts(model2$fitted.values)
plot(fits, res2, xlab = "Residuals", ylab = "Fitted Values",
    main = "Residuals versus Fits")
```



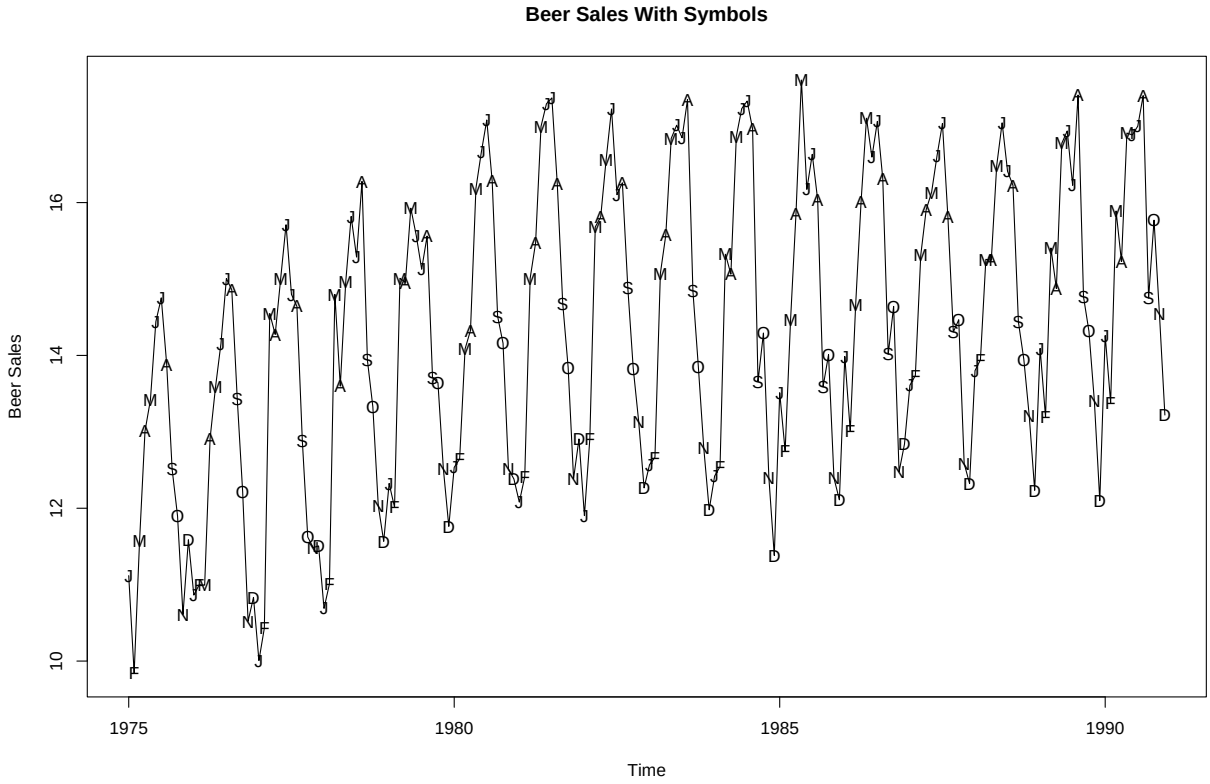
Problem 4

```
data(beersales)
```

```
plot(beersales, xlab = "Time", ylab = "Beer Sales", main = "Beer Sales No Symbols")
```




```
plot(beersales, xlab = "Time", ylab = "Beer Sales", main = "Beer Sales With Symbols")
points(y = beersales, x = time(beersales), pch = as.vector(season(beersales)))
```

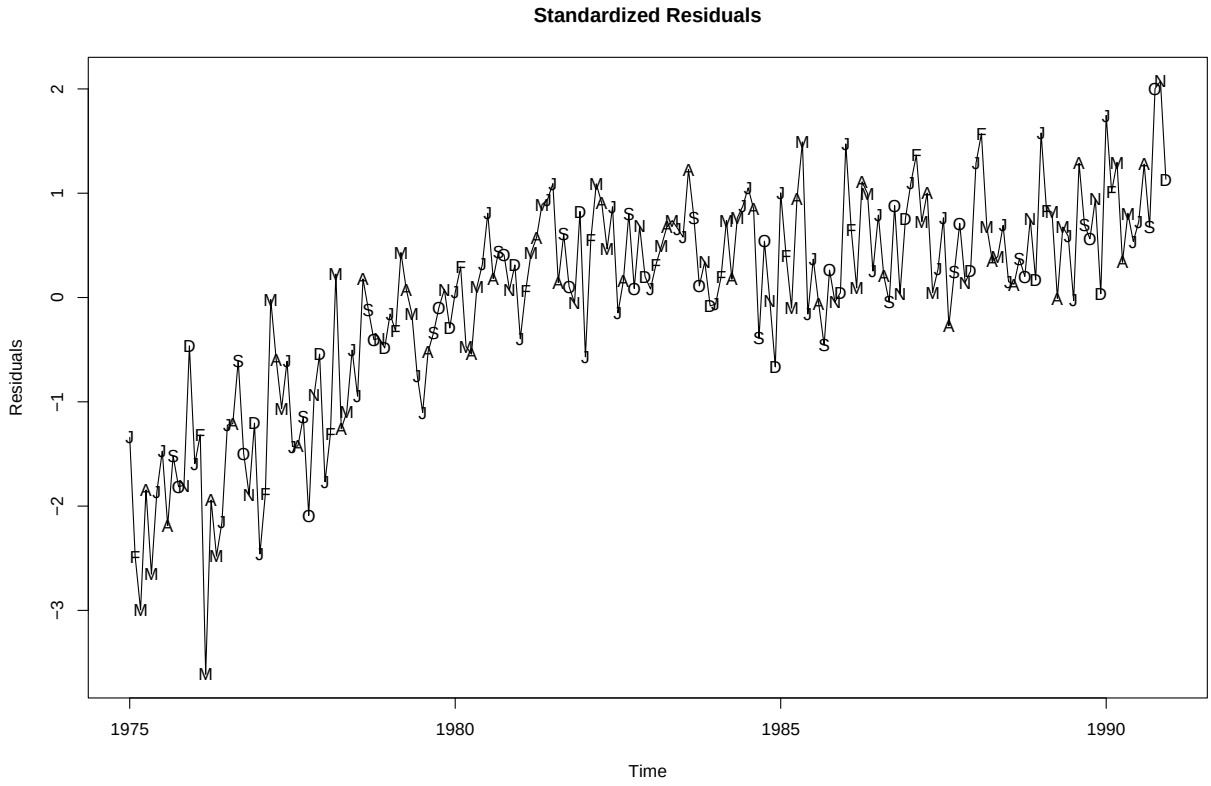



```
months <- season(beersales)
model <- lm(beersales ~ months - 1)
summary(model)
```

```
##
## Call:
## lm(formula = beersales ~ months - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5745 -0.4772  0.1759  0.7312  2.1023
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## monthsJanuary    12.4857     0.2639   47.31   <2e-16 ***
## monthsFebruary   12.3431     0.2639   46.77   <2e-16 ***
## monthsMarch      14.5679     0.2639   55.20   <2e-16 ***
## monthsApril      14.8833     0.2639   56.39   <2e-16 ***
## monthsMay        16.0846     0.2639   60.95   <2e-16 ***
## monthsJune       16.3354     0.2639   61.90   <2e-16 ***
## monthsJuly       16.2543     0.2639   61.59   <2e-16 ***
## monthsAugust     16.0945     0.2639   60.98   <2e-16 ***
## monthsSeptember  14.0585     0.2639   53.27   <2e-16 ***
## monthsOctober    13.7401     0.2639   52.06   <2e-16 ***
## monthsNovember   12.4377     0.2639   47.13   <2e-16 ***
## monthsDecember   12.0626     0.2639   45.71   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.056 on 180 degrees of freedom
## Multiple R-squared:  0.995,  Adjusted R-squared:  0.9946
## F-statistic:  2964 on 12 and 180 DF,  p-value: < 2.2e-16
```

```
plot(rstudent(model), x = as.vector(time(beersales)), xlab = "Time",
    ylab = "Residuals", main = "Standardized Residuals", type = "l")
points(y = rstudent(model), x = as.vector(time(beersales)), pch = as.vector(season(beersales)))
```

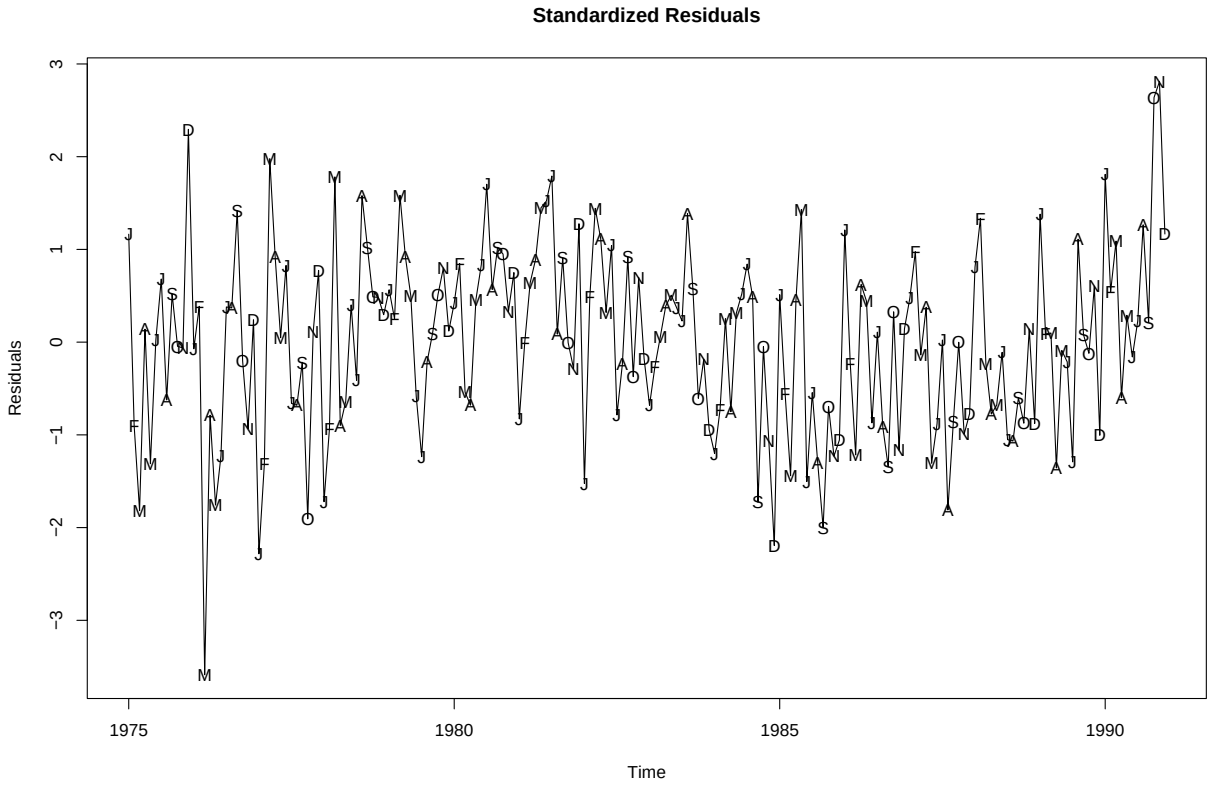



```
time <- time(beersales)
time2 <- time^2
model2 <- lm(beersales ~ (months - 1) + time + time2)
summary(model2)
```

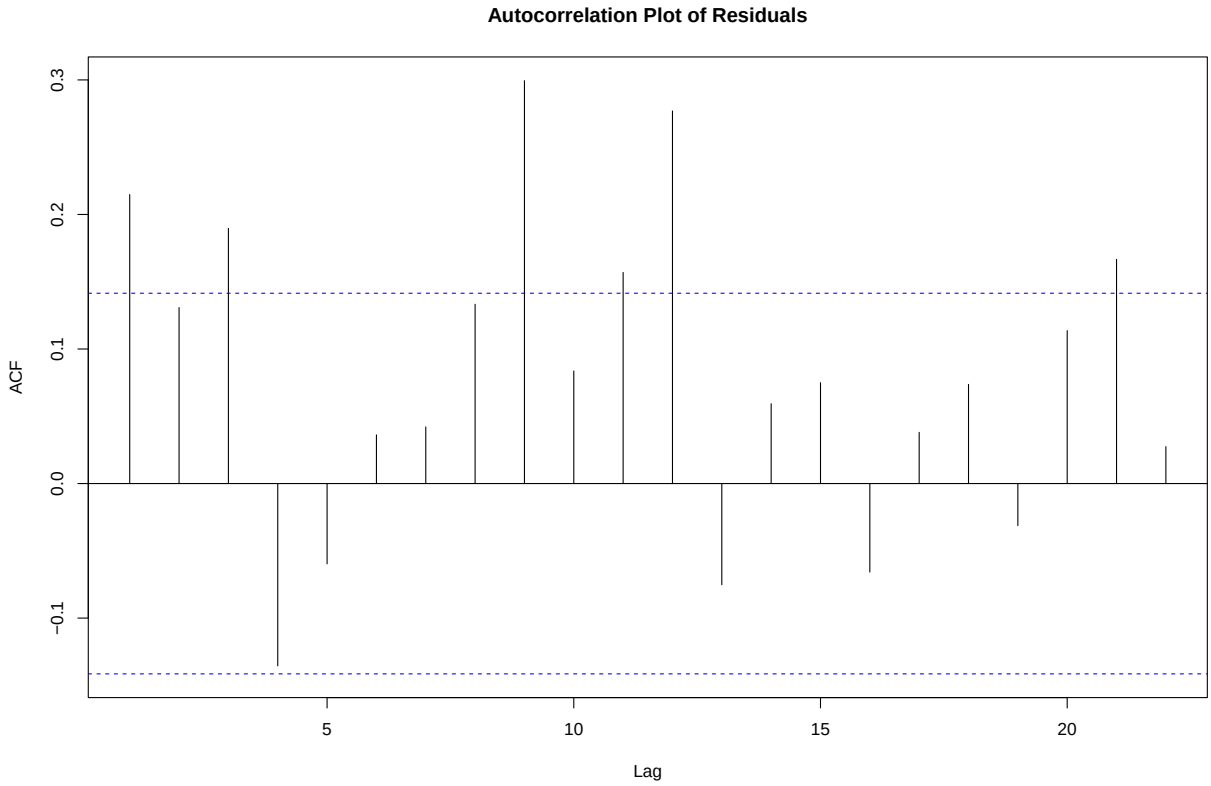
```
##
## Call:
## lm(formula = beersales ~ (months - 1) + time + time2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.03203 -0.43118  0.04977  0.34509  1.57572
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
```

```
## monthsJanuary   -7.150e+04  8.791e+03  -8.133 6.93e-14 ***
## monthsFebruary  -7.150e+04  8.791e+03  -8.133 6.93e-14 ***
## monthsMarch     -7.150e+04  8.791e+03  -8.132 6.94e-14 ***
## monthsApril     -7.150e+04  8.791e+03  -8.132 6.94e-14 ***
## monthsMay       -7.149e+04  8.791e+03  -8.132 6.95e-14 ***
## monthsJune      -7.149e+04  8.791e+03  -8.132 6.95e-14 ***
## monthsJuly      -7.149e+04  8.791e+03  -8.132 6.95e-14 ***
## monthsAugust    -7.149e+04  8.791e+03  -8.132 6.95e-14 ***
## monthsSeptember -7.150e+04  8.791e+03  -8.133 6.94e-14 ***
## monthsOctober   -7.150e+04  8.791e+03  -8.133 6.94e-14 ***
## monthsNovember  -7.150e+04  8.791e+03  -8.133 6.93e-14 ***
## monthsDecember  -7.150e+04  8.791e+03  -8.133 6.93e-14 ***
## time             7.196e+01  8.867e+00   8.115 7.70e-14 ***
## time2           -1.810e-02  2.236e-03  -8.096 8.63e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5911 on 178 degrees of freedom
## Multiple R-squared:  0.9984, Adjusted R-squared:  0.9983
## F-statistic:  8132 on 14 and 178 DF,  p-value: < 2.2e-16
```

```r
res2 <- as.ts(rstandard(model2))
plot(res2, x = as.vector(time(beersales)), xlab = "Time", ylab = "Residuals",
    main = "Standardized Residuals", type = "l")
points(y = res2, x = as.vector(time(beersales)), pch = as.vector(season(beersales)))
```



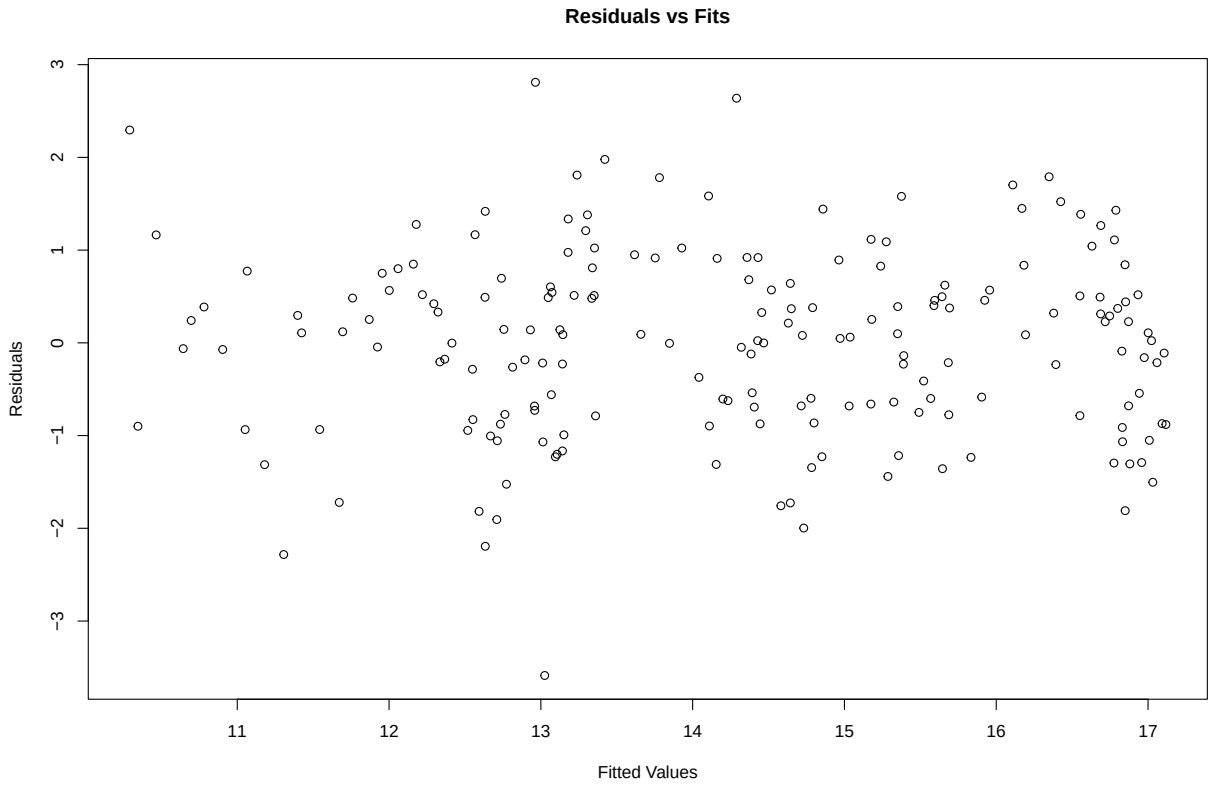
```r
acf(res2, main = "Autocorrelation Plot of Residuals")$acf  #gives ACF values
```



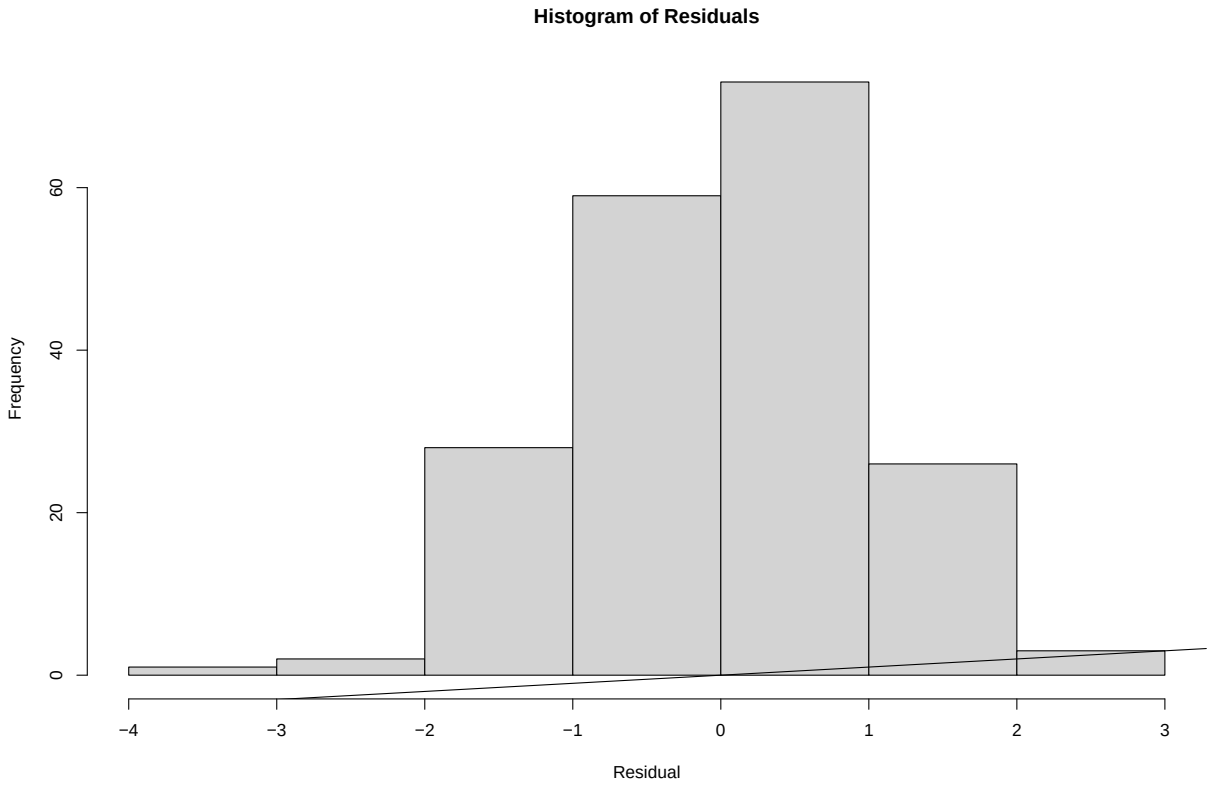

```
## , , 1
##
##              [,1]
##  [1,]  0.21482684
##  [2,]  0.13083367
##  [3,]  0.18968496
##  [4,] -0.13551269
##  [5,] -0.05967941
##  [6,]  0.03619658
##  [7,]  0.04213090
##  [8,]  0.13328029
##  [9,]  0.29943835
## [10,]  0.08367388
## [11,]  0.15692698
## [12,]  0.27693645
## [13,] -0.07525813
## [14,]  0.05937381
## [15,]  0.07497894
## [16,] -0.06580710
## [17,]  0.03808102
## [18,]  0.07372621
## [19,] -0.03125110
## [20,]  0.11367803
## [21,]  0.16666030
```

```
## [22,]  0.02743557
```

```r
fits <- as.ts(model2$fitted.values)
plot(fits, res2, xlab = "Fitted Values", ylab = "Residuals",
    main = "Residuals vs Fits")
```




```r
hist(res2, xlab = "Residual", main = "Histogram of Residuals")
abline(a = 0, b = 1)
```



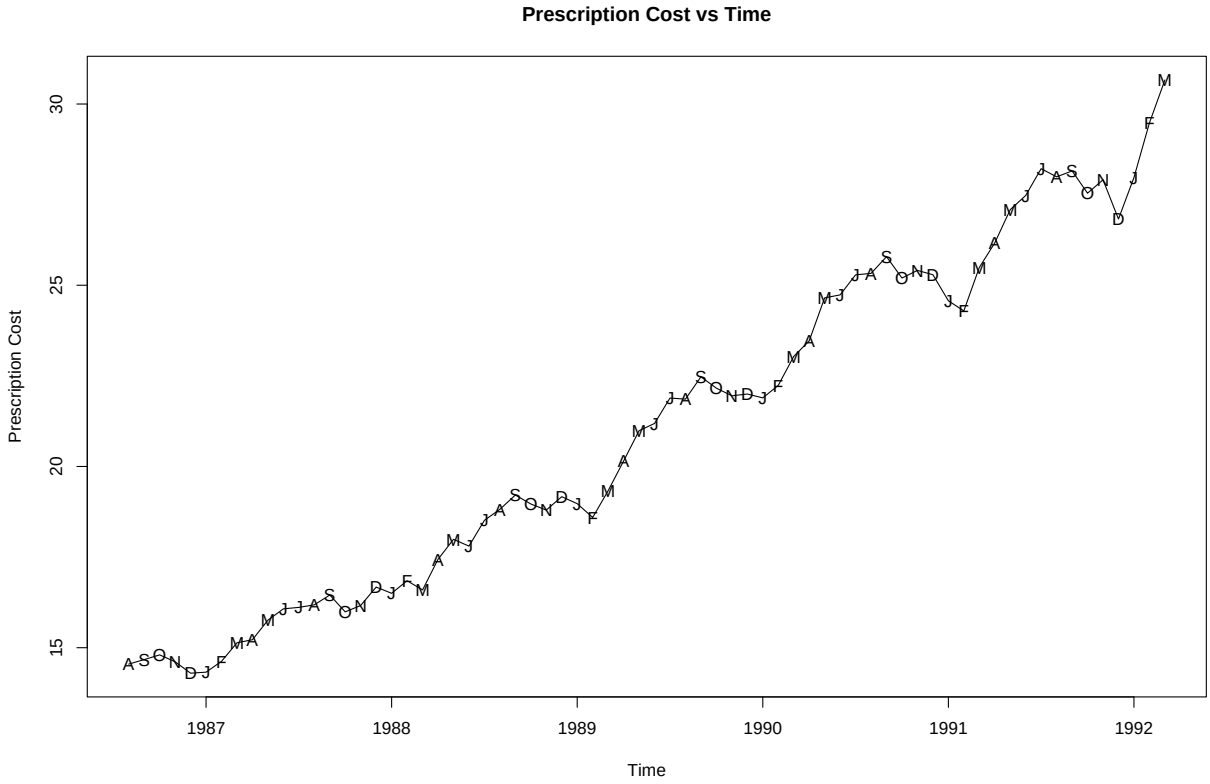

```
shapiro.test(res2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res2
## W = 0.99307, p-value = 0.5019
```

Problem 5

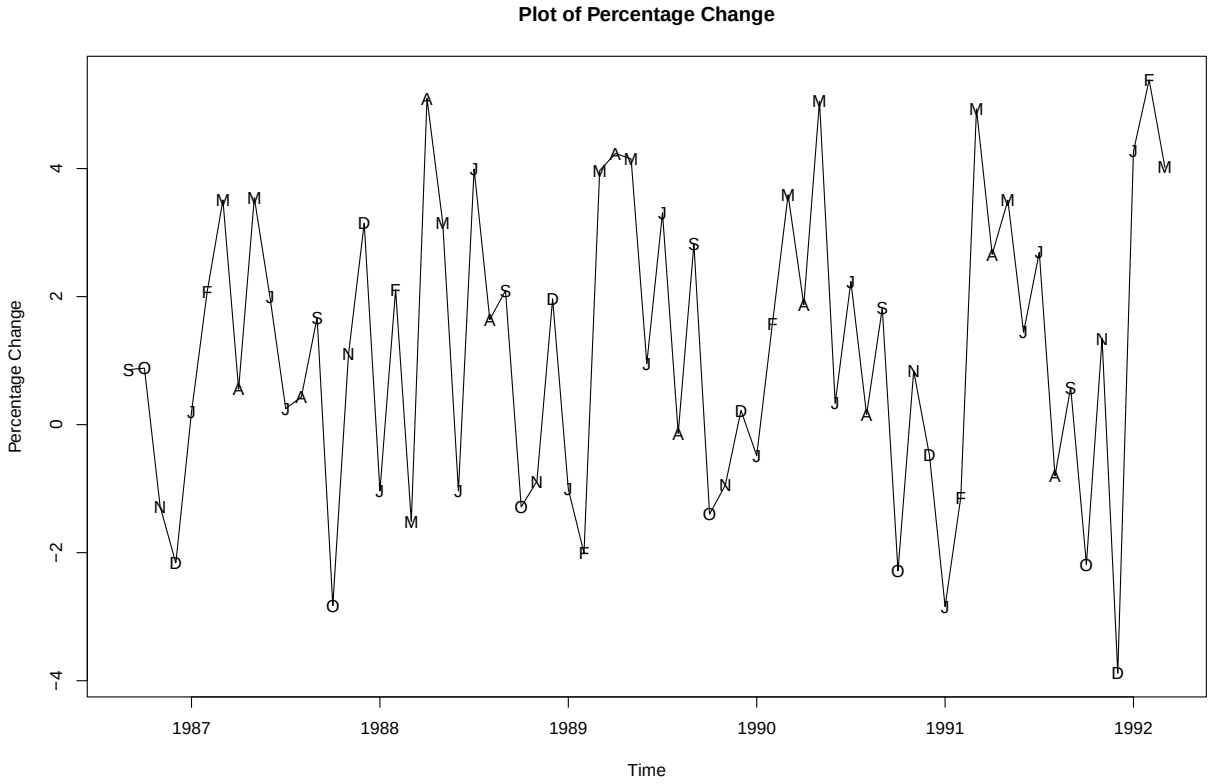
```
data(prescrip)
```

```
plot(prescrip, xlab = "Time", ylab = "Prescription Cost", main = "Prescription Cost vs Time")
points(y = prescrip, x = time(prescrip), pch = as.vector(season(prescrip)))
```




```
y <- NULL
n <- length(prescrip)

for (i in 2:n) {
    y <- c(y, 100 * (prescrip[i]/prescrip[i - 1] - 1))
}
pct.change <- ts(y, frequency = 12, start = c(1986, 9), end = c(1992,
    3))
plot(pct.change, ylab = "Percentage Change", main = "Plot of Percentage Change")
points(y = pct.change, x = time(pct.change), pch = as.vector(season(pct.change)))
```

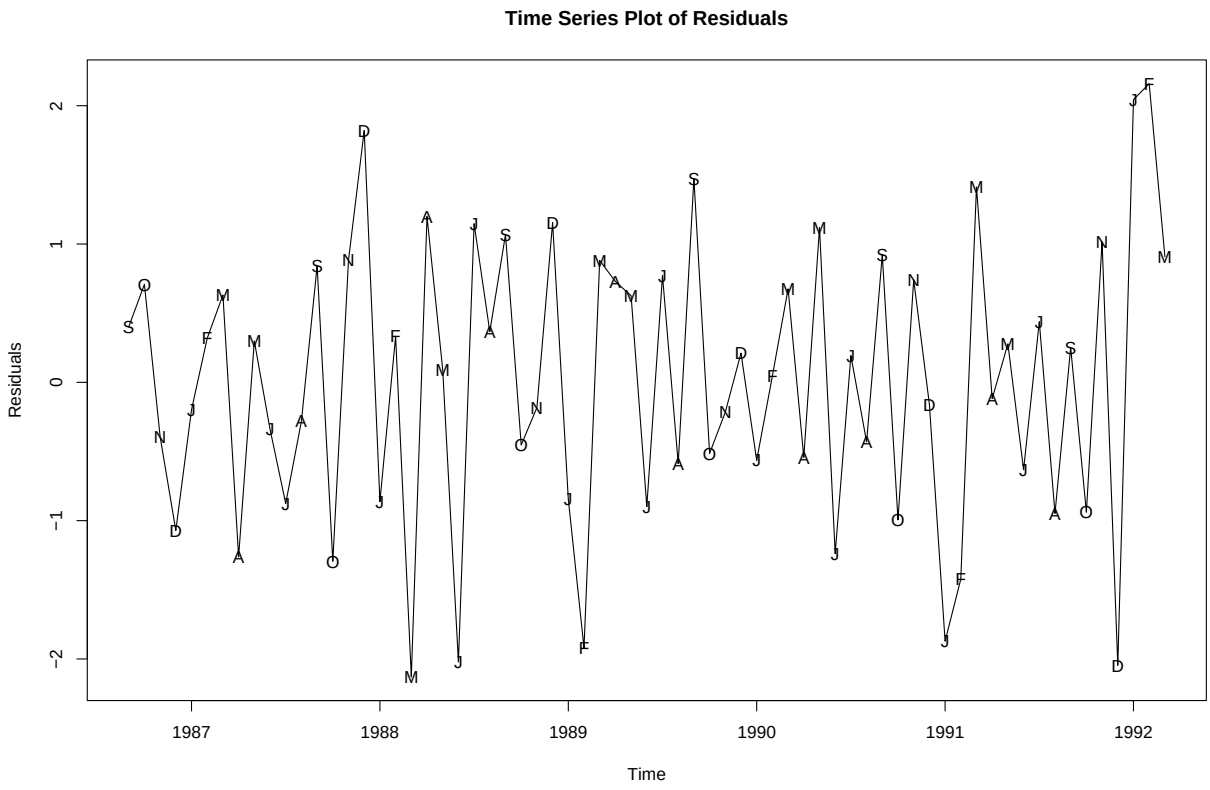



```
har. <- harmonic(pct.change, 1)
model.presc <- lm(pct.change ~ har.)
summary(model.presc)
```

```
##
## Call:
## lm(formula = pct.change ~ har.)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8444 -1.3742  0.1697  1.4069  3.8980
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       1.2217     0.2325   5.254 1.82e-06 ***
## har.cos(2*pi*t)  -0.6538     0.3298  -1.982   0.0518 .
## har.sin(2*pi*t)   1.6596     0.3269   5.077 3.54e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.897 on 64 degrees of freedom
## Multiple R-squared:  0.3148, Adjusted R-squared:  0.2933
## F-statistic:  14.7 on 2 and 64 DF,  p-value: 5.584e-06
```

```
res <- ts(rstudent(model.presc), frequency = 12, start = c(1986,
    9), end = c(1992, 3))
```

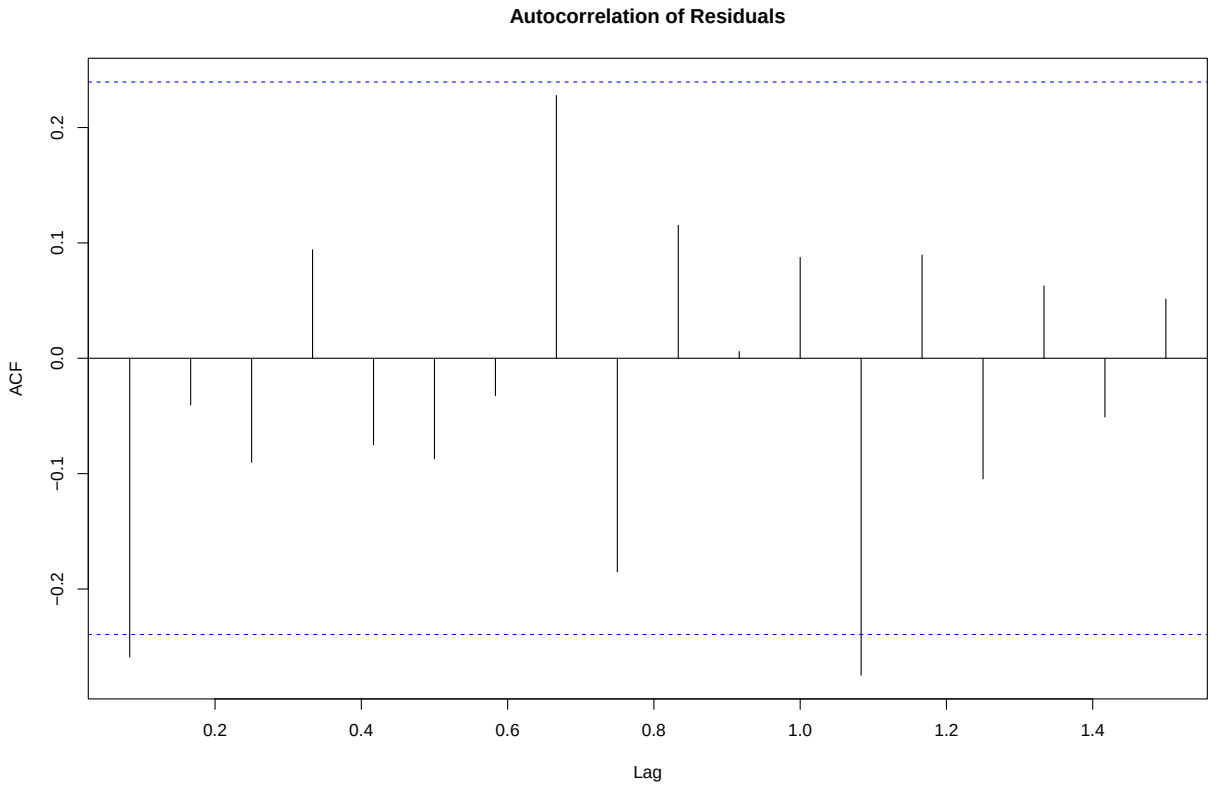
```
plot(res, ylab = "Residuals", main = "Time Series Plot of Residuals")
points(y = res, x = time(res), pch = as.vector(season(res)))
```



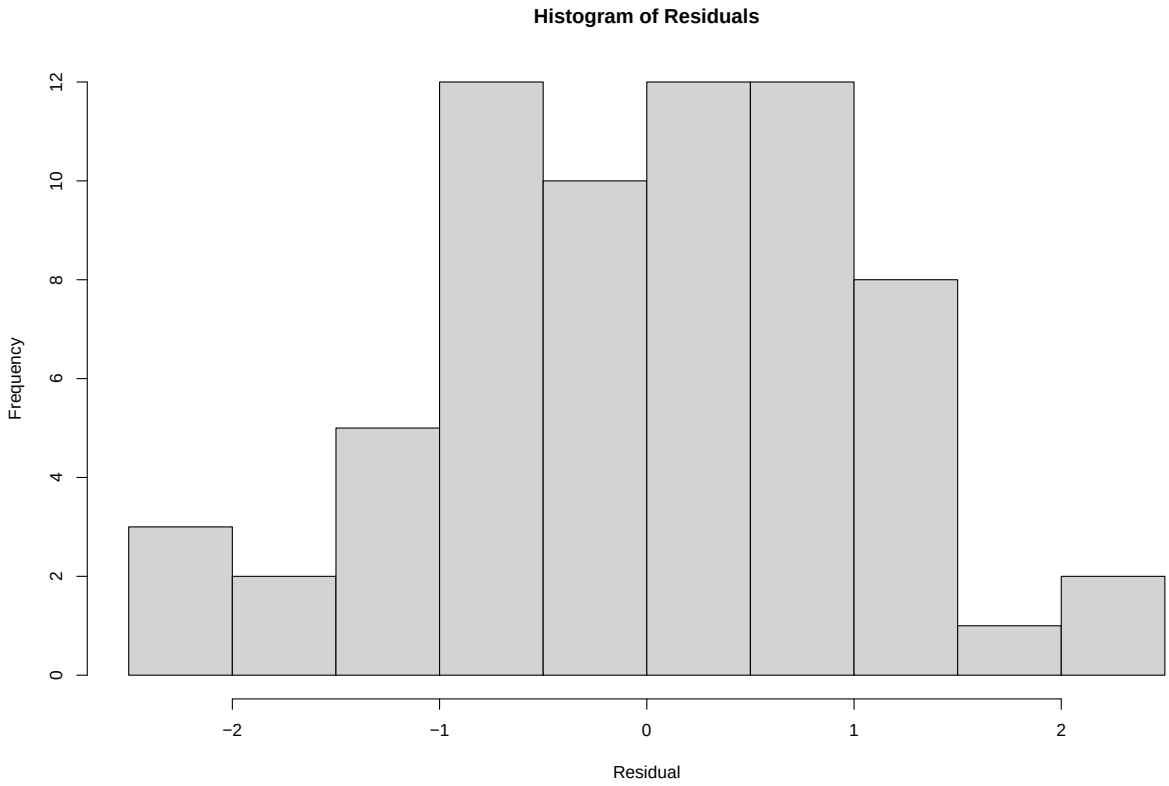

```
adf.test(res)
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  res
## Dickey-Fuller = -3.7193, Lag order = 4, p-value = 0.02997
## alternative hypothesis: stationary
```

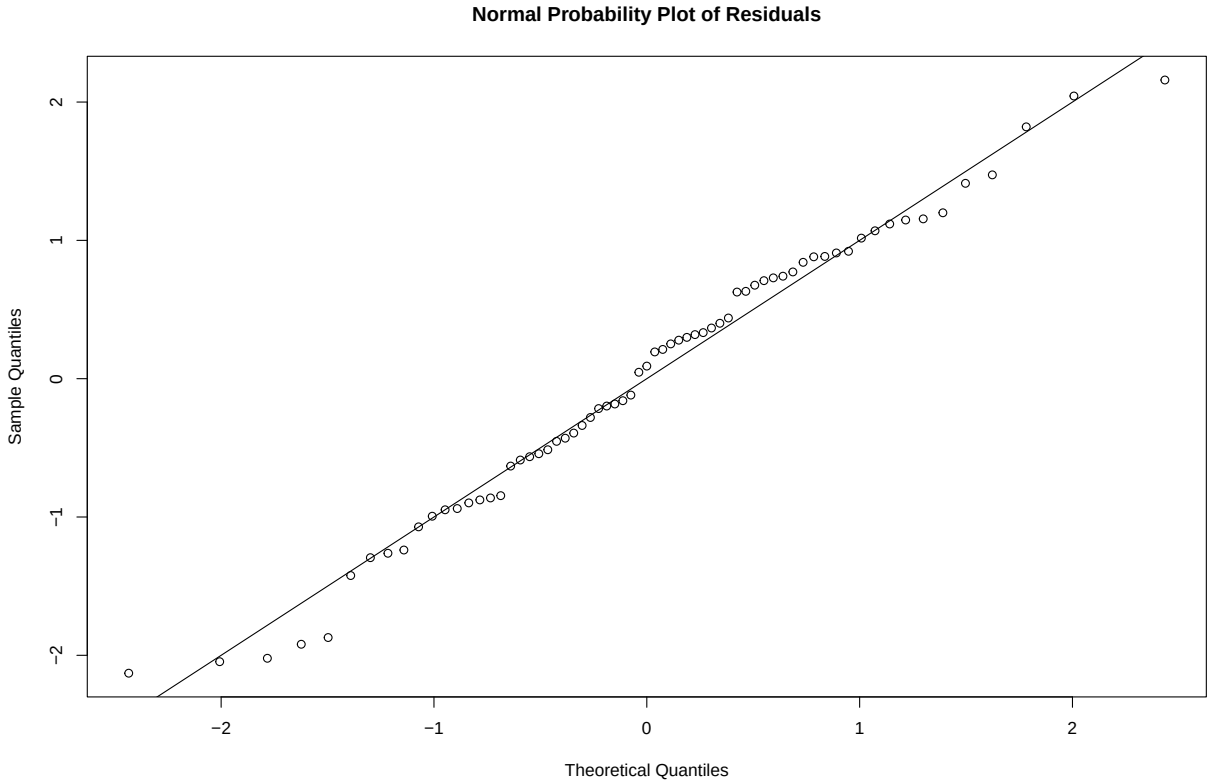
```
acf(res, main = "Autocorrelation of Residuals")
```




```
hist(res, xlab = "Residual", main = "Histogram of Residuals")
```




```r
qqnorm(res, main = "Normal Probability Plot of Residuals")
abline(a = 0, b = 1)
```




```
shapiro.test(res)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res
## W = 0.98273, p-value = 0.4781
```