# 10.18 Notes

*Hongdou Li*

*10/18/2018*

## What is a Time Series?

Atime series is a collection of data points corresponding to temporal measurements of some quantitative variable.

A time series is best visualized as a scatter plot of y versus t with adjacent points connected by a straight line. We call such plot a time series plot.

A time series plot clearly displays the relationship between the variable y and time. Time series analysis typically refers to modeling the relationship between the variable y and time.

A time series model characterizes the nature of the relationship between $y_{t+1}$ and $\{y_1, y_2, ..., y_t\}$. in general a model will take the form

$$y_{t+1} = f(y_1, y_2, ...y_t)$$

and we use such a model to predict the value of $y_{t+1}$ given the history $\{y_1, y_2, ..., y_t\}$ already observed. we call this form of prediction forecasting.

## What is TS Analysis & Forecasting?

At a very general level we can think of time series analysis and forecasting as: Trying to understand the past to predict the future.

Time series models can be broadly classified into two different types:

- Univariate

** Future values of y are forecasted using only knowledge of past values of y.

- Multivariate

** Future values of y are forecasted using past values of y and past values of one or more other variables $x_1, x_2, x_3, ....$

## Modeling a time Series

Regardless of the classification, effective time series models (of either kind) account for the following three important features of a time series.

** Serial Correlation

Serial correlation refers to the phenomenon whereby observations closer together in time tend to be more similar than observations further apart in time.

Serial correlation is quantified by the autocorrelation function.

we can visualize the extent of sutocorrelation in a given time series using ACF plots - plots of the autocorrelation function Corr$(y_t, y_{t+h})$ versus the lag h.

** Trend

Trend is consistent directional movement in a time series.

Trend may correspond to consistent increases and/or decreases in a given time series.

Trend typifies the general, smoothed, behavior of a time series.

** Seasonality

Seasonality is a characteristic of a time series in which the data experiences regular and predictable fluctuations according to some period. we would say that the autocorrelation of lag s ($Corr(y_t, y_{t+s})$) is large.

## Modeling a Time Series

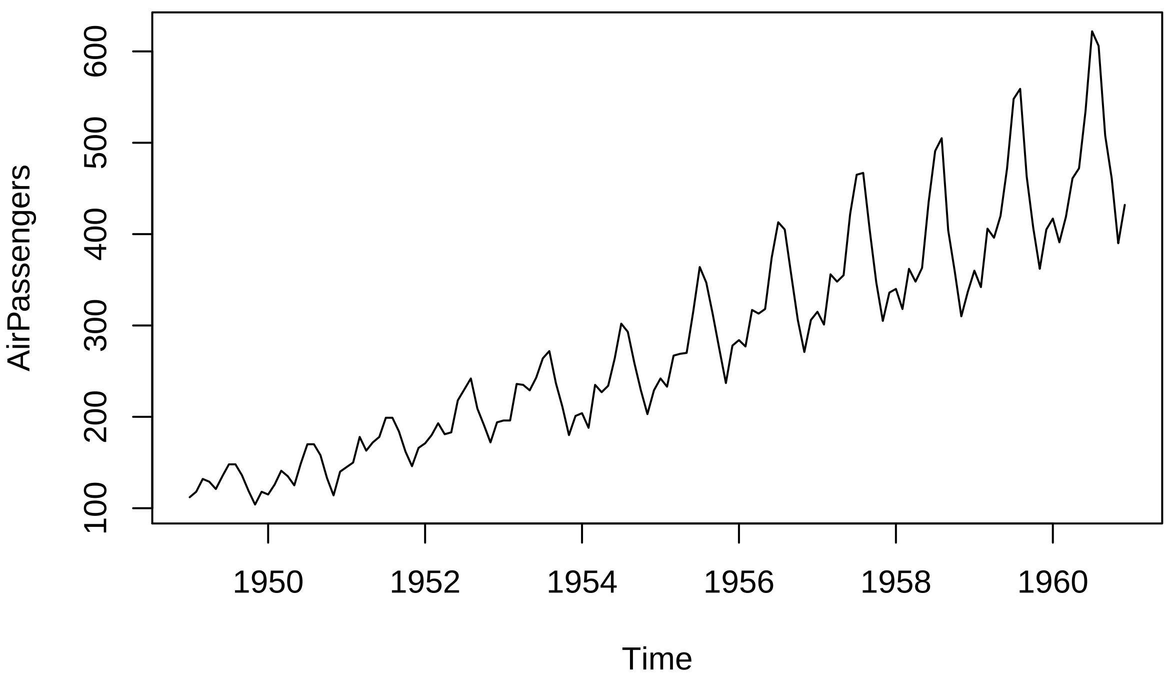
Effective time series models handle both trend and seasonality by accounting for various autocorrelation structures in the observed data. Random variation is unavoidable and the chief contributor to model uncertainty.

Time Series Decomposition: A time series model that can capture the trend and/or seasonality in a dataset will effectively explain the behavior of the time series on average.

And a model that captures the expected behavior of the time series should be able to accurately forecast future values.

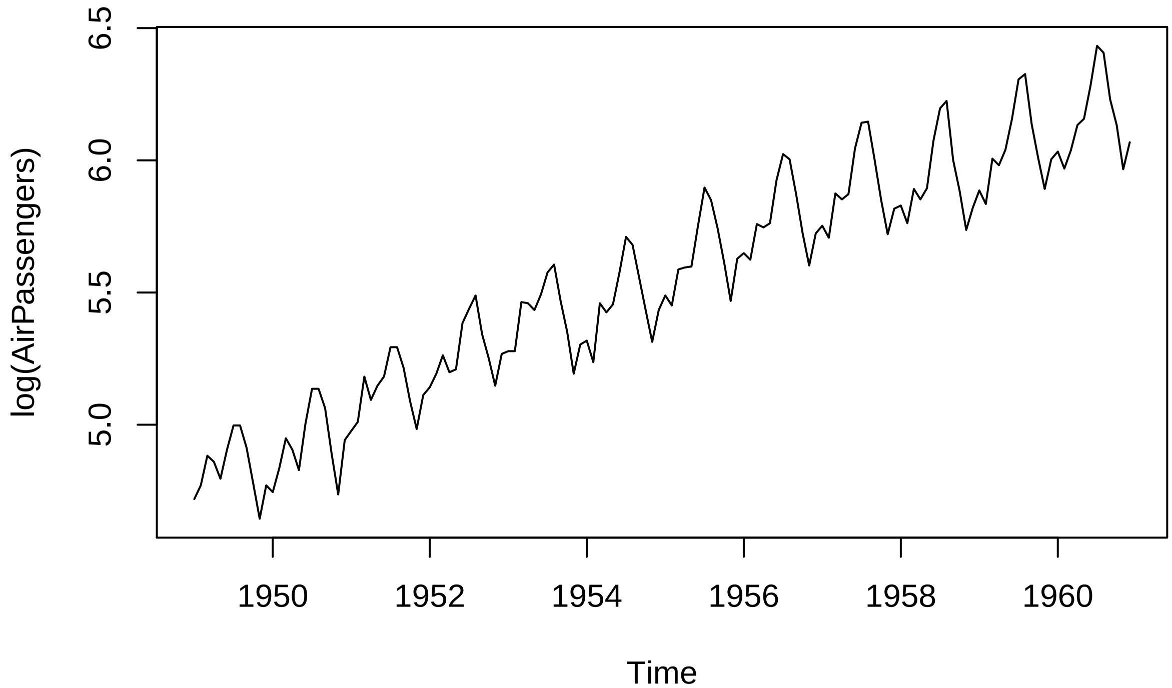
With this goal in mind- trying to capture trend and/or seasonality- we now discuss some specific modeling approaches.

```r
data(AirPassengers)
plot(AirPassengers)
```




Plooting the time series data. Notice that the data is already in the form of a time series. but the variability becomes bigger and bigger. so we need to do a transformation

```r
plot(log(AirPassengers))
```




```
#prepare for modeling
t <- time(AirPassengers) # Extracting time as the explanatory variate from the time series framework of
cycle(AirPassengers) # introducing month as the season
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1949   1   2   3   4   5   6   7   8   9  10  11  12
## 1950   1   2   3   4   5   6   7   8   9  10  11  12
## 1951   1   2   3   4   5   6   7   8   9  10  11  12
## 1952   1   2   3   4   5   6   7   8   9  10  11  12
## 1953   1   2   3   4   5   6   7   8   9  10  11  12
## 1954   1   2   3   4   5   6   7   8   9  10  11  12
## 1955   1   2   3   4   5   6   7   8   9  10  11  12
## 1956   1   2   3   4   5   6   7   8   9  10  11  12
## 1957   1   2   3   4   5   6   7   8   9  10  11  12
## 1958   1   2   3   4   5   6   7   8   9  10  11  12
## 1959   1   2   3   4   5   6   7   8   9  10  11  12
## 1960   1   2   3   4   5   6   7   8   9  10  11  12
```

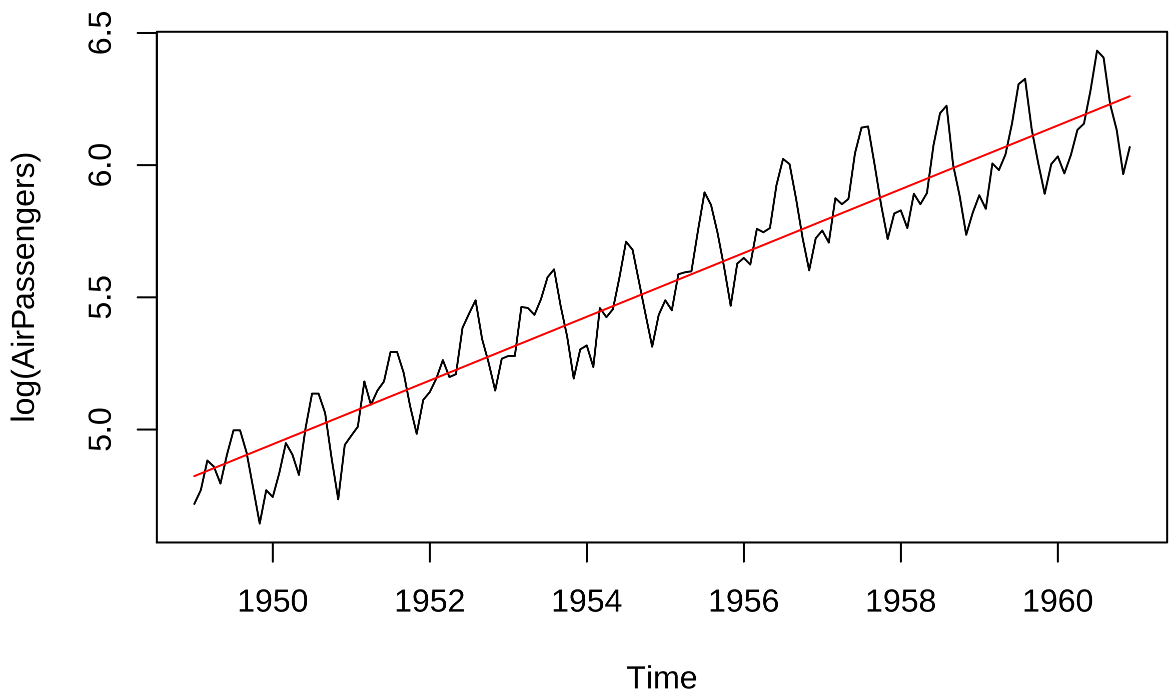
```
month <- as.factor(cycle(AirPassengers))
```

```
# Model the data
reg0 <- lm(log(AirPassengers)~t)
summary(reg0)
```

```
##
## Call:
## lm(formula = log(AirPassengers) ~ t)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.30858 -0.10388 -0.01796  0.09738  0.29538
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.302e+02  6.539e+00  -35.20   <2e-16 ***
## t            1.206e-01  3.345e-03   36.05   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.139 on 142 degrees of freedom
## Multiple R-squared:  0.9015, Adjusted R-squared:  0.9008
## F-statistic:  1300 on 1 and 142 DF,  p-value: < 2.2e-16
```

```r
plot(log(AirPassengers))
points(t,predict.lm(reg0),type='l',col='red')
```

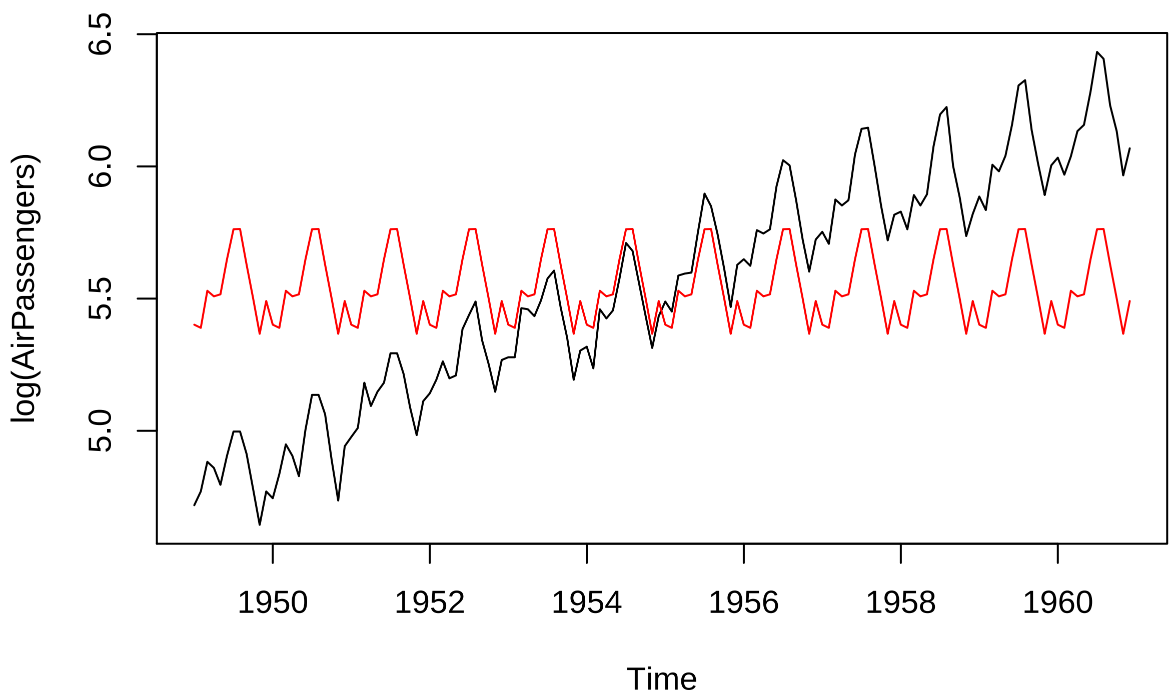



```r
reg1 <- lm(log(AirPassengers)~month)
summary(reg1)
```

```
##
## Call:
## lm(formula = log(AirPassengers) ~ month)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.76598 -0.33024  0.00432  0.37042  0.67052
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.40139    0.12691  42.561   <2e-16 ***
## month2      -0.01199    0.17948  -0.067   0.9469
## month3       0.12831    0.17948   0.715   0.4759
## month4       0.10711    0.17948   0.597   0.5517
## month5       0.11481    0.17948   0.640   0.5235
## month6       0.24702    0.17948   1.376   0.1711
## month7       0.36103    0.17948   2.012   0.0463 *
```

```
## month8       0.36181    0.17948   2.016   0.0458 *
## month9       0.22724    0.17948   1.266   0.2077
## month10      0.09915    0.17948   0.552   0.5816
## month11     -0.03450    0.17948  -0.192   0.8479
## month12      0.08944    0.17948   0.498   0.6191
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4396 on 132 degrees of freedom
## Multiple R-squared:  0.08454,    Adjusted R-squared:  0.008254
## F-statistic: 1.108 on 11 and 132 DF,  p-value: 0.3598
```

```r
plot(log(AirPassengers))
points(t,predict.lm(reg1), type='l',col='red')
```

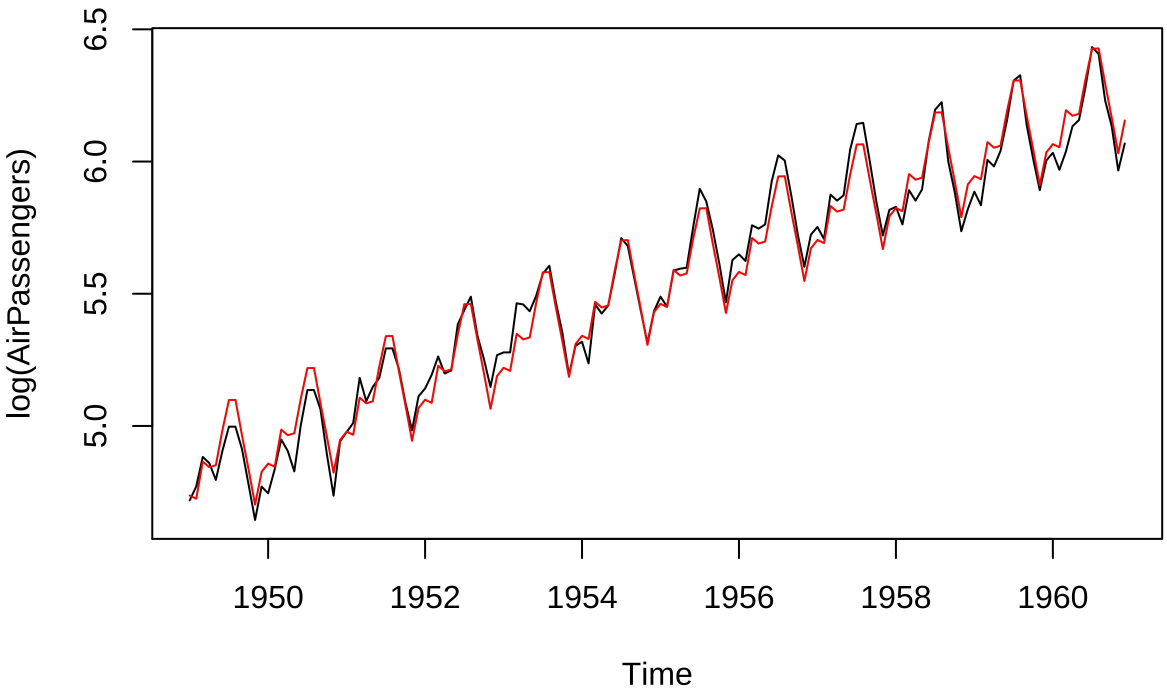



```r
reg2 <- lm(log(AirPassengers)~t+month)
summary(reg2)
```

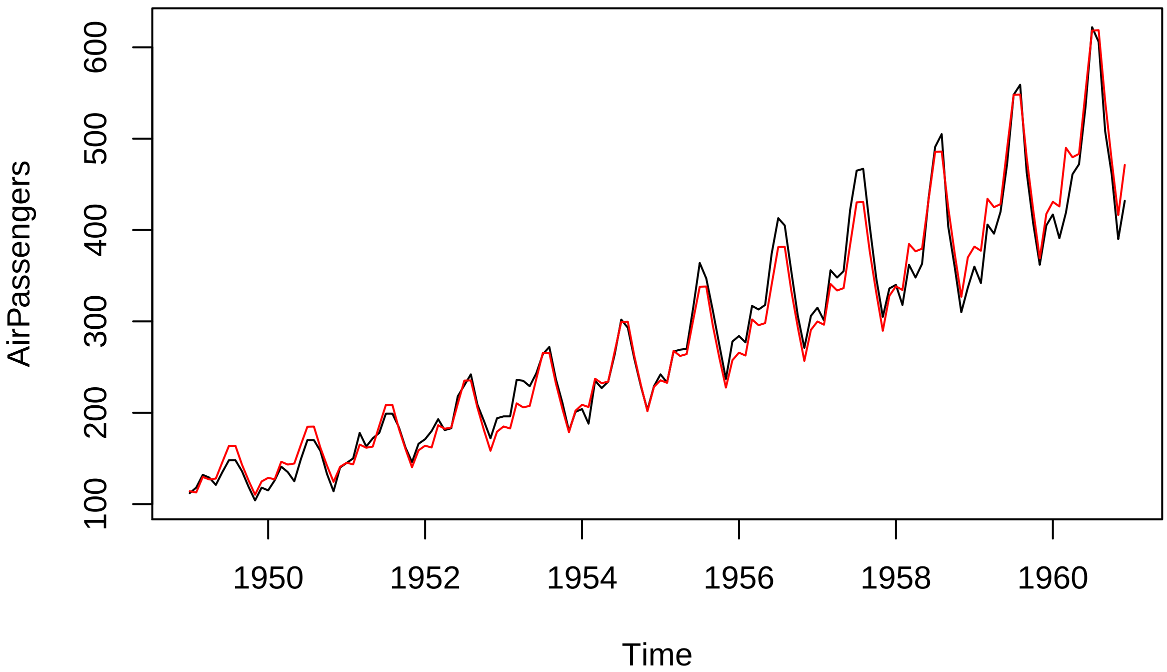
```
##
## Call:
## lm(formula = log(AirPassengers) ~ t + month)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.156370 -0.041016  0.003677  0.044069  0.132324
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.308e+02  2.798e+00 -82.467  < 2e-16 ***
## t            1.208e-01  1.432e-03  84.399  < 2e-16 ***
## month2      -2.206e-02  2.421e-02  -0.911  0.36400
## month3       1.082e-01  2.421e-02   4.468 1.69e-05 ***
## month4       7.690e-02  2.421e-02   3.176  0.00186 **
```

```
## month5       7.453e-02  2.422e-02   3.078  0.00254 **
## month6       1.967e-01  2.422e-02   8.121 2.98e-13 ***
## month7       3.006e-01  2.422e-02  12.411  < 2e-16 ***
## month8       2.913e-01  2.423e-02  12.026  < 2e-16 ***
## month9       1.467e-01  2.423e-02   6.054 1.39e-08 ***
## month10      8.532e-03  2.423e-02   0.352  0.72537
## month11     -1.352e-01  2.424e-02  -5.577 1.34e-07 ***
## month12     -2.132e-02  2.425e-02  -0.879  0.38082
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0593 on 131 degrees of freedom
## Multiple R-squared:  0.9835, Adjusted R-squared:  0.982
## F-statistic: 649.4 on 12 and 131 DF,  p-value: < 2.2e-16
```

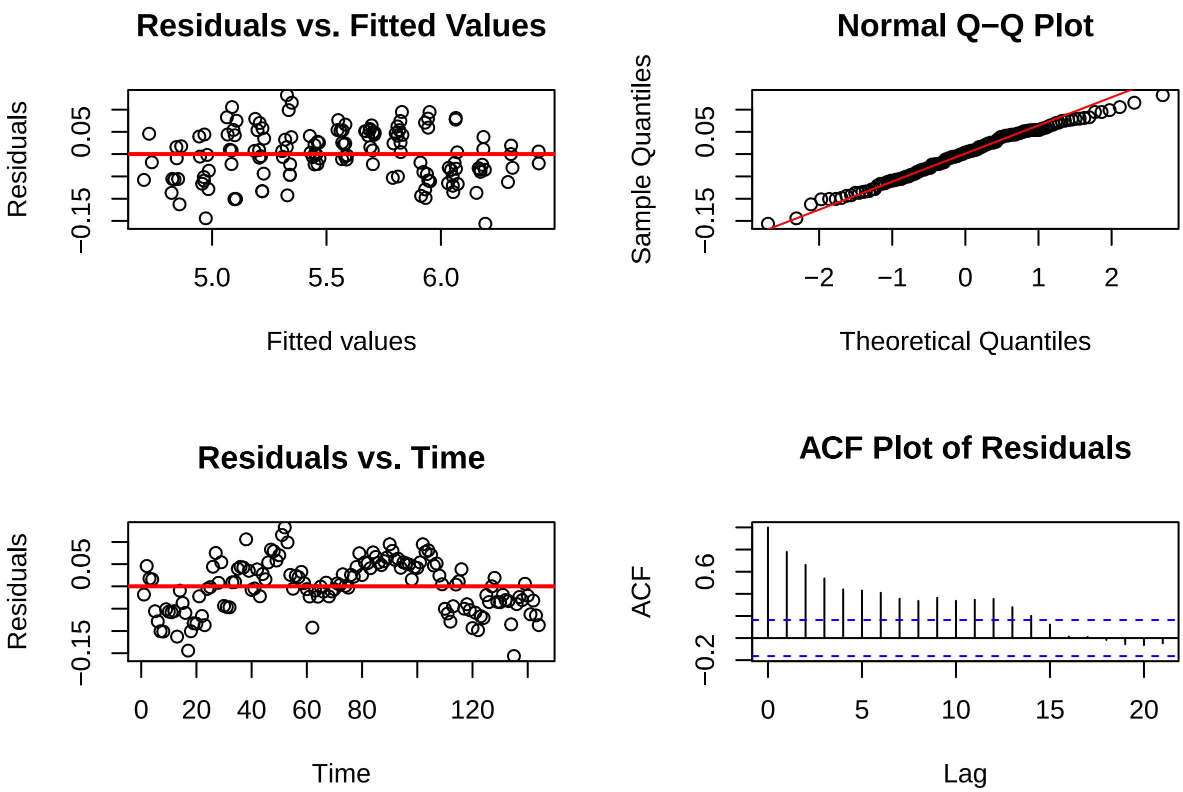
```
plot(log(AirPassengers))
points(t,predict.lm(reg2), type='l',col='red')
```




```
# Exponentiating the fitted values to reverse the log transformation
par(mfrow=c(1,1))
plot(AirPassengers)
points(t,exp(predict.lm(reg2)),type='l',col='red')
```




```
#Diagnostic plots for reg2 model
par(mfrow=c(2,2))
plot(reg2$fitted, reg2$residuals, main='Residuals vs. Fitted Values', ylab='Residuals', xlab='Fitted va
abline(h=0,col='red',lwd=2)
qqnorm(reg2$residuals)
qqline(reg2$residuals,col='red')
plot(reg2$residuals, main='Residuals vs. Time', ylab='Residuals', xlab='Time')
abline(h=0,col='red',lwd=2)
acf(reg2$residuals,main='ACF Plot of Residuals')
```



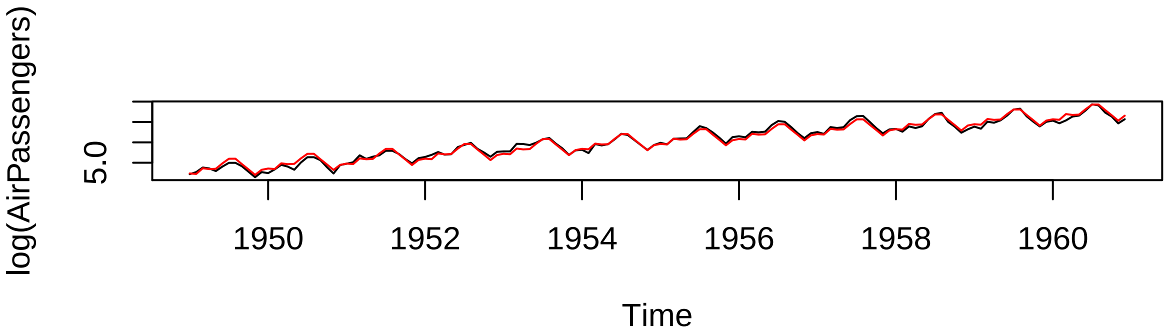

Notice that the residual seems non-linear, we add a quadratic term.

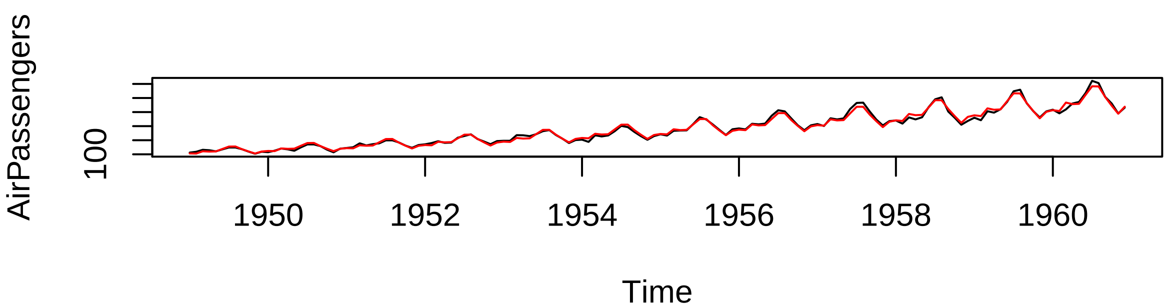
```r
par(mfrow=c(2,1))
t2 <- t^2
reg4 <- lm(log(AirPassengers)~t+t2+month)
summary(reg4)
```

```
##
## Call:
## lm(formula = log(AirPassengers) ~ t + t2 + month)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.12748 -0.03709  0.00418  0.03197  0.11529
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.205e+04  1.430e+03  -8.426 5.79e-14 ***
## t            1.222e+01  1.463e+00   8.347 8.95e-14 ***
## t2          -3.093e-03  3.743e-04  -8.265 1.41e-13 ***
## month2      -2.227e-02  1.968e-02  -1.132 0.259839
## month3       1.078e-01  1.968e-02   5.477 2.15e-07 ***
## month4       7.639e-02  1.968e-02   3.882 0.000164 ***
## month5       7.393e-02  1.968e-02   3.756 0.000259 ***
## month6       1.960e-01  1.968e-02   9.959  < 2e-16 ***
## month7       3.000e-01  1.969e-02  15.238  < 2e-16 ***
## month8       2.907e-01  1.969e-02  14.765  < 2e-16 ***
## month9       1.462e-01  1.969e-02   7.423 1.33e-11 ***
## month10      8.145e-03  1.970e-02   0.414 0.679912
## month11     -1.354e-01  1.970e-02  -6.873 2.36e-10 ***
```

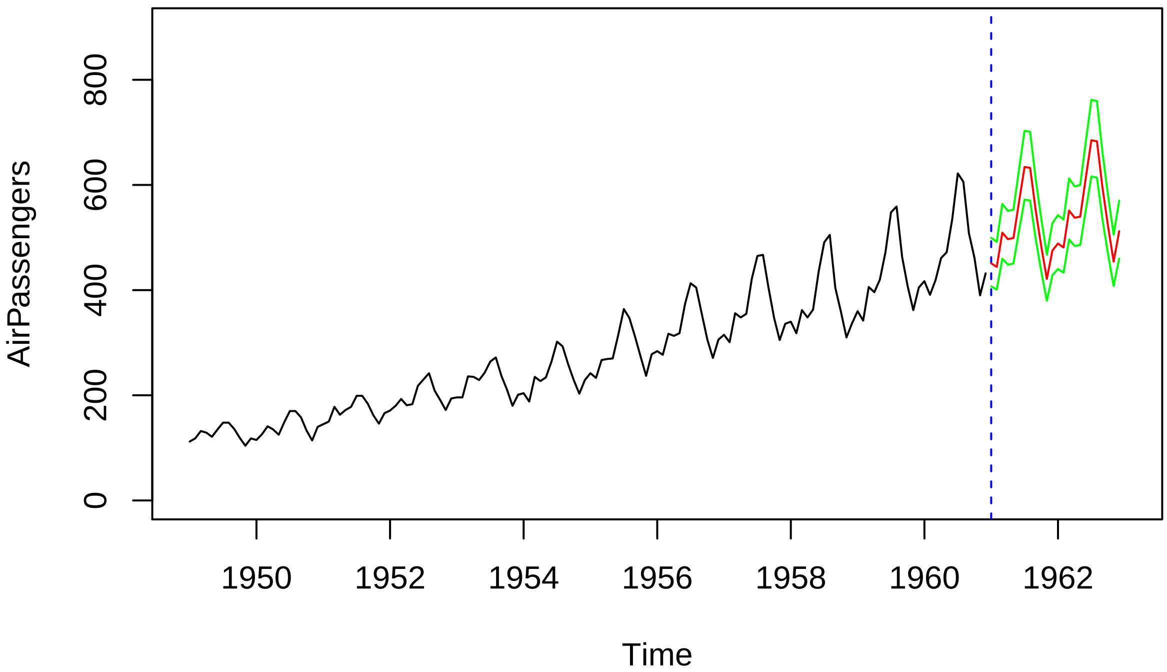
```
## month12     -2.132e-02  1.971e-02  -1.082 0.281286
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0482 on 130 degrees of freedom
## Multiple R-squared:  0.9892, Adjusted R-squared:  0.9881
## F-statistic: 912.7 on 13 and 130 DF,  p-value: < 2.2e-16
```

```
plot(log(AirPassengers))
points(t,predict.lm(reg2), type='l',col='red')
plot(AirPassengers)
points(t,exp(predict.lm(reg4)),type='l',col='red')
```

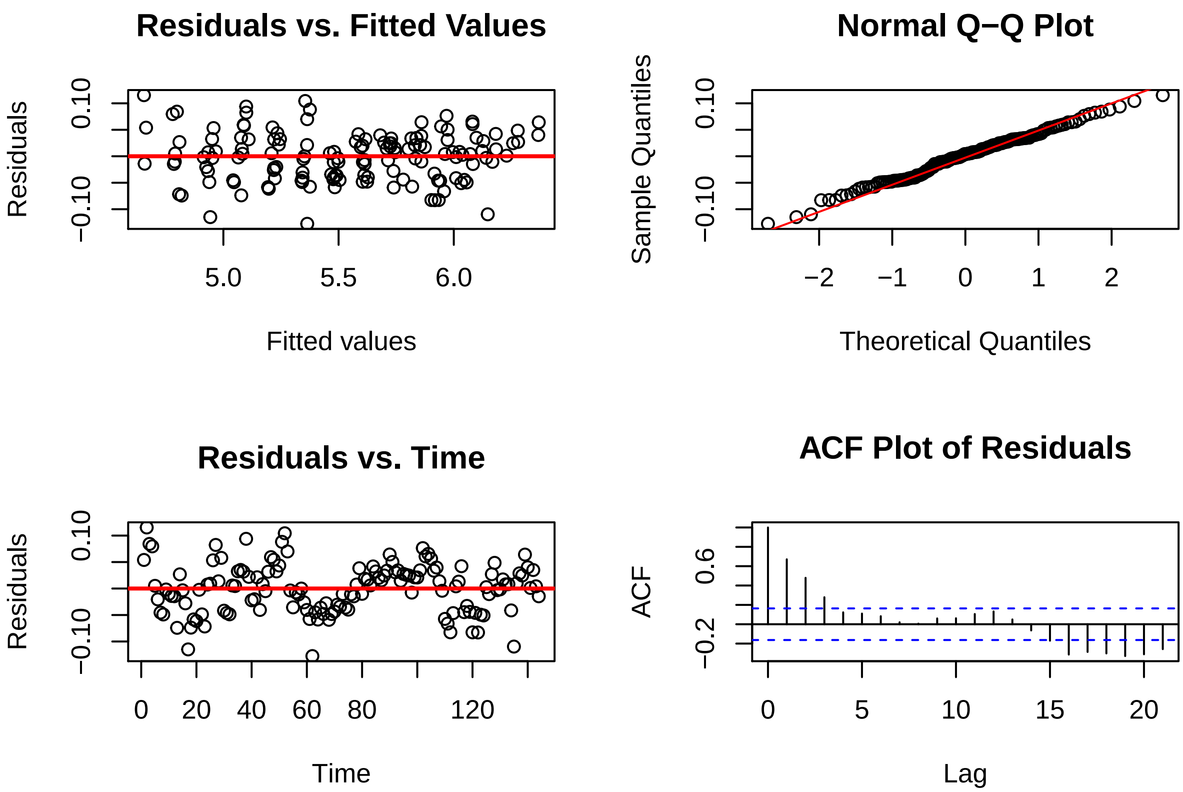



```
# Prediction in data
t.new <- seq(1961, 1963, length=25)[1:24]
t2.new <- t.new^2
month.new <- factor(rep(1:12,2))
new <- data.frame(t=t.new,t2=t2.new,month=month.new)
pred <- predict.lm(reg4,new,interval='prediction')
#par(mforw=c(1,1))
plot(AirPassengers,xlim=c(1949,1963),ylim=c(0,900))
abline(v=1961,col='blue',lty=2)
lines(exp(pred[,1])~t.new,type='l',col='red')
lines(exp(pred[,2])~t.new,col='green')
lines(exp(pred[,3])~t.new,col='green')
```




```
par(mfrow=c(2,2))
plot(reg4$fitted, reg4$residuals, main='Residuals vs. Fitted Values', ylab='Residuals', xlab='Fitted val
abline(h=0,col='red',lwd=2)
qqnorm(reg4$residuals)
qqline(reg4$residuals,col='red')
plot(reg4$residuals, main='Residuals vs. Time', ylab='Residuals', xlab='Time')
abline(h=0,col='red',lwd=2)
acf(reg4$residuals,main='ACF Plot of Residuals')
```


Residuals vs. Fitted Values
Residuals
0.10
−0.10
5.0
5.5
6.0
Fitted values
Normal Q−Q Plot
Sample Quantiles
0.10
−0.10
−2
−1
0
1
2
Theoretical Quantiles
Residuals vs. Time
Residuals
0.10
−0.10
0
20
40
60
80
120
Time
ACF Plot of Residuals
ACF
0.6
−0.2
0
5
10
15
20
Lag