Time Series Analysis of Aviation Data

Dr. Richard Xie

February, 2012

METRON AVIATION
THE SCIENCE OF HARMONIZING AIR TRAFFIC

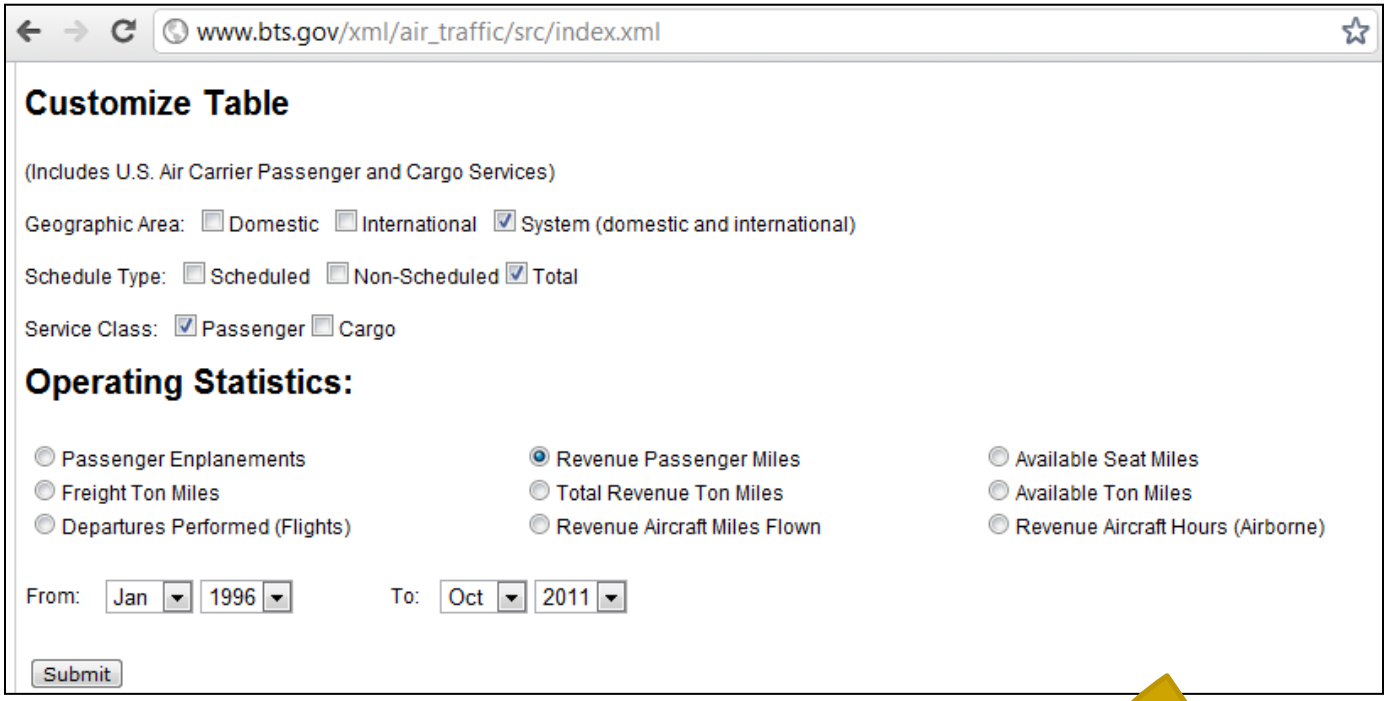
GEORGE MASON UNIVERSITY

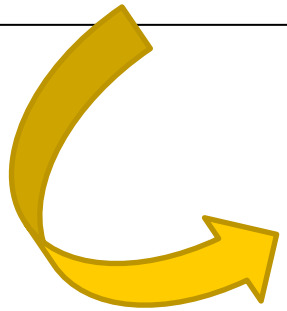
What is a Time Series

- A *time series* is a sequence of observations in chorological order, such as
  - Daily closing price of stock MSFT in the past ten years
  - Weekly unemployment claims in the past 2 years
  - **Monthly airline revenue passenger miles in the past ten years**
- Time series analysis is useful when
  - No other data available
  - System too complicated to model in detail

GEORGE MASON UNIVERSITY

Where to Get the Data?





	A	B
1	YYYYMM	Total
2	Jan-96	41,972,194
3	Feb-96	42,054,796
4	Mar-96	50,443,045
5	Apr-96	47,112,397
6	May-96	49,118,248
7	Jun-96	52,880,510
8	Jul-96	55,664,750
9	Aug-96	57,723,208
10	Sep-96	47,035,464
11	Oct-96	49,263,120
12	Nov-96	43,937,074
13	Dec-96	48,539,606
14	Jan-97	45,850,623
15	Feb-97	42,838,949
16	Mar-97	53,620,994
17	Apr-97	49,282,817
18	May-97	51,191,842
19	Jun-97	54,707,221
20	Jul-97	57,995,025
21	Aug-97	59,715,433

GEORGE MASON UNIVERSITY

What Information Are You Interested In?

- How the data changes from month to month, year to year?
  - Any trend?
  - How fluctuated the curve is?
  - Any seasonal effects?
  - Any unusual years/months which have significantly small or large number?
- Can we forecast future value based on the time series?

GEORGE MASON UNIVERSITY

Let's Work on the Data

- But first, what tool will you use?
  - Pencil and quadrille pad (or back of an envelope)
  - Excel
  - Matlab, Mathematica, Maple
  - SAS, SPSS, STATA, R
  - ROOT, PAW, KNIME, Data Applied, etc.
  - Others

GEORGE MASON UNIVERSITY

Use R!

- R is free
- R is a language, not just a statistical tool
- R makes graphics and visualization of the best quality
- A flexible statistical analysis toolkit
- Access to powerful, cutting-edge analytics
- A robust, vibrant community
- Unlimited possibilities

GEORGE MASON UNIVERSITY

Where to Download R

- To download R
  - Go to http://www.r-project.org/
  - Choose a CRAN Mirror, such as http://cran.cnr.berkeley.edu/
  - Click the link to download R according to your operating system (Linux, MacOS X, or Windows)
- Or Enhanced Version of R Distributed by 3rd Party
  - Revolution R Community (http://www.revolutionanalytics.com/products/revolution-r.php)
  - Free academic version of Revolution R Enterprise (http://www.revolutionanalytics.com/products/revolution-enterprise.php)

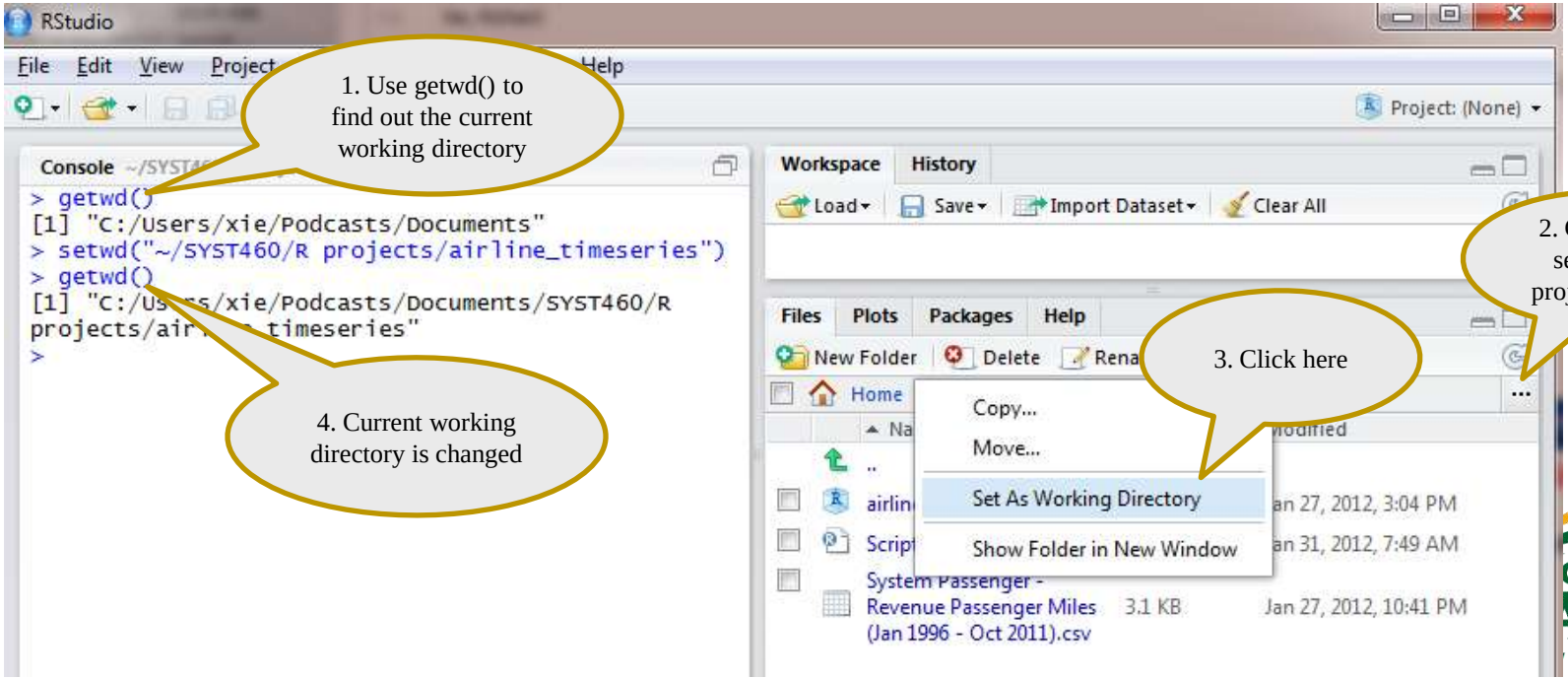
R References

- *An Introduction to R*, W.N.Venables, D.M.Smith and R Development Core Team
- More Documents/Tutorials, go to
  - http://cran.cnr.berkeley.edu/other-docs.html

GEORGE MASON UNIVERSITY

Start an R Project

- Recommend using RStudio as the console (http://rstudio.org/download/)
- Create a project folder for storing R scripts, data, etc.
  - e.g. C:/Users/xie/Documents/SYST460/R projects/airline_timeseries
- Open RStudio, navigate to the project folder



Work on Data, Finally!

- Use Ctr+Shft+N to create a new script
- > *rm(list=ls(all=TRUE))* to clear the existing variables in workspace, if any
- > *rpm = read.csv("System Passenger - Revenue Passenger Miles (Jan 1996 - Oct 2011).csv")*
- Use Ctr+Enter to run the current line or selection

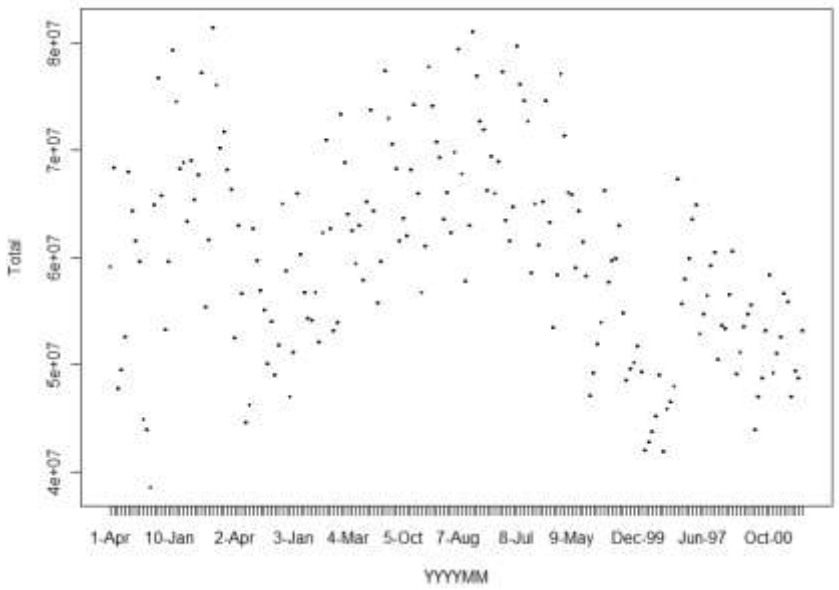
GEORGE MASON UNIVERSITY

What the Data Looks Like?

- > *ls(rpm)*

```
> ls(rpm)
[1] "Total"  "YYYYMM"
```

- > *plot(rpm)*, equivalent to
- > *plot(rpm$Total~rpm$YYYYMM)*



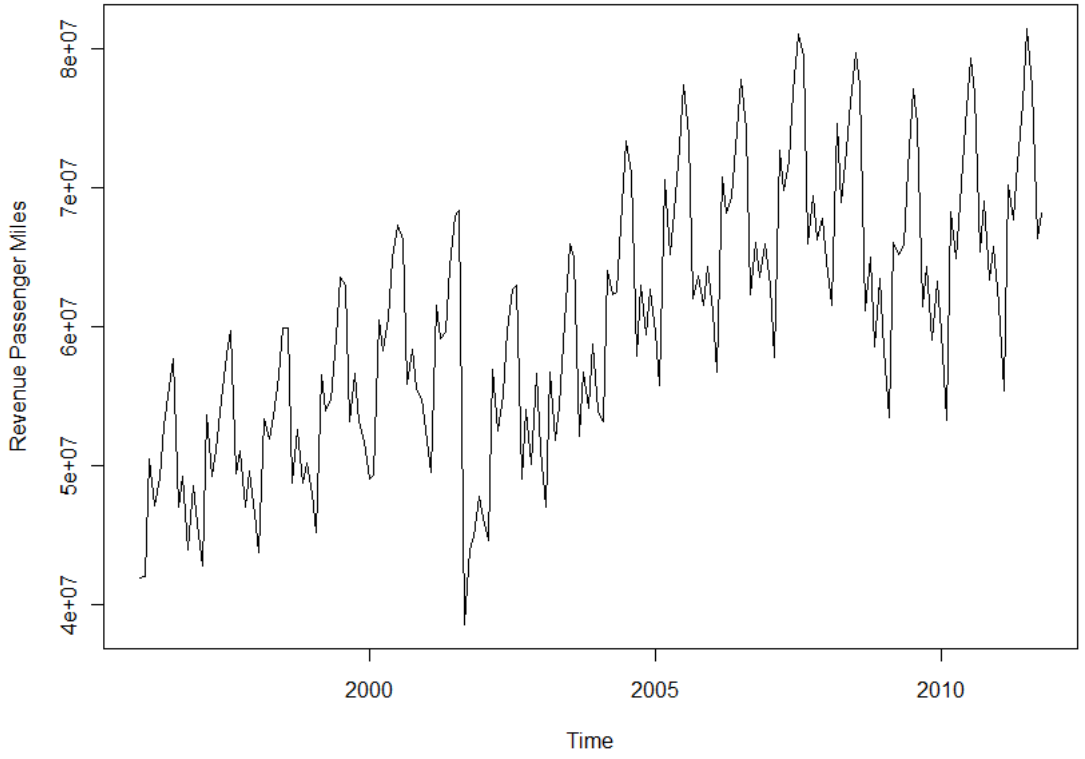
GEORGE MASON UNIVERSITY

Plot It As A Time Series

- *> rpm.ts = ts(as.numeric(rpm$Total), start = c(1996,01),freq=12)*
- *> plot(rpm.ts,ylab='Revenue Passenger Miles')*

Commands to check properties of *rmp.ts*

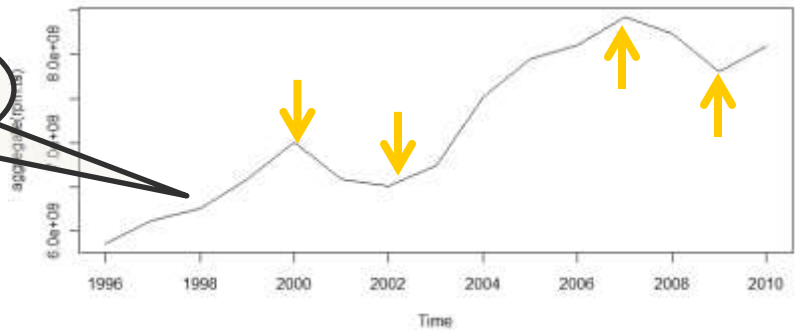
```
> class(rpm.ts)
[1] "ts"
> start(rpm.ts)
[1] 1996    1
> end(rpm.ts)
[1] 2011   10
> frequency(rpm.ts)
[1] 12
> summary(rpm.ts)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
38600000 53220000 60110000 60390000 66320000 81420000
```

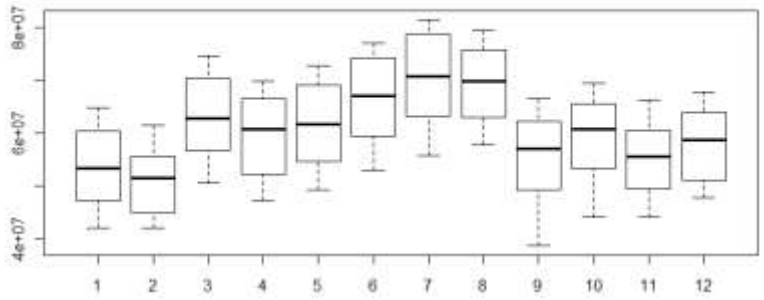


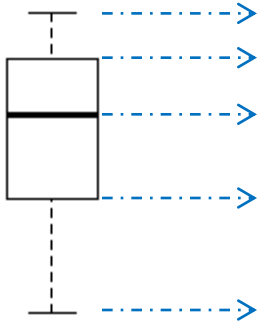
Trends and Seasonal Variation

- *> layout(1:2)*
- *> plot(aggregate(rpm.ts))*
- *> boxplot(rpm.ts~cycle(rpm.ts))*

Overall trend of increasing over years







Max value
Upper Quartile
Median
Lower Quartile
Min value

Window Function

- Extract a part of the time series between specified start and end points
- *> rpm.Feb <- window(rpm.ts, start = c(1996,02), freq = TRUE)*
- *> rpm.Aug <- window(rpm.ts, start = c(1996,08), freq = TRUE)*
- *> mean(rpm.Feb)/mean(rpm.Aug)*

```
> mean(rpm.Feb)/mean(rpm.Aug)
[1] 0.7329244
```

GEORGE MASON UNIVERSITY

Modeling Time Series - Notations

- Represent a time series of length n by $\{x_t : t = 1, \ldots, n\} = \{x_1, x_2, \ldots, x_n\}$
- A time series model is a sequence of random variables, and the observed time series is considered a realization from the model.
- $\bar{x}$ means sample mean: $\bar{x} = \sum x_i / n$
- $\hat{x}$ means a forecast: $\hat{x}_{t+k|t}$ is the forecast made at time t for the value at $(t+k)$

GEORGE MASON UNIVERSITY

Modeling Time Series – Decomposition Models

- Additive Decomposition Model

$$x_t = m_t + s_t + z_t$$

- Multiplicative Decomposition Model

$$x_t = m_t \times s_t + z_t$$

which can be converted to an additive model as

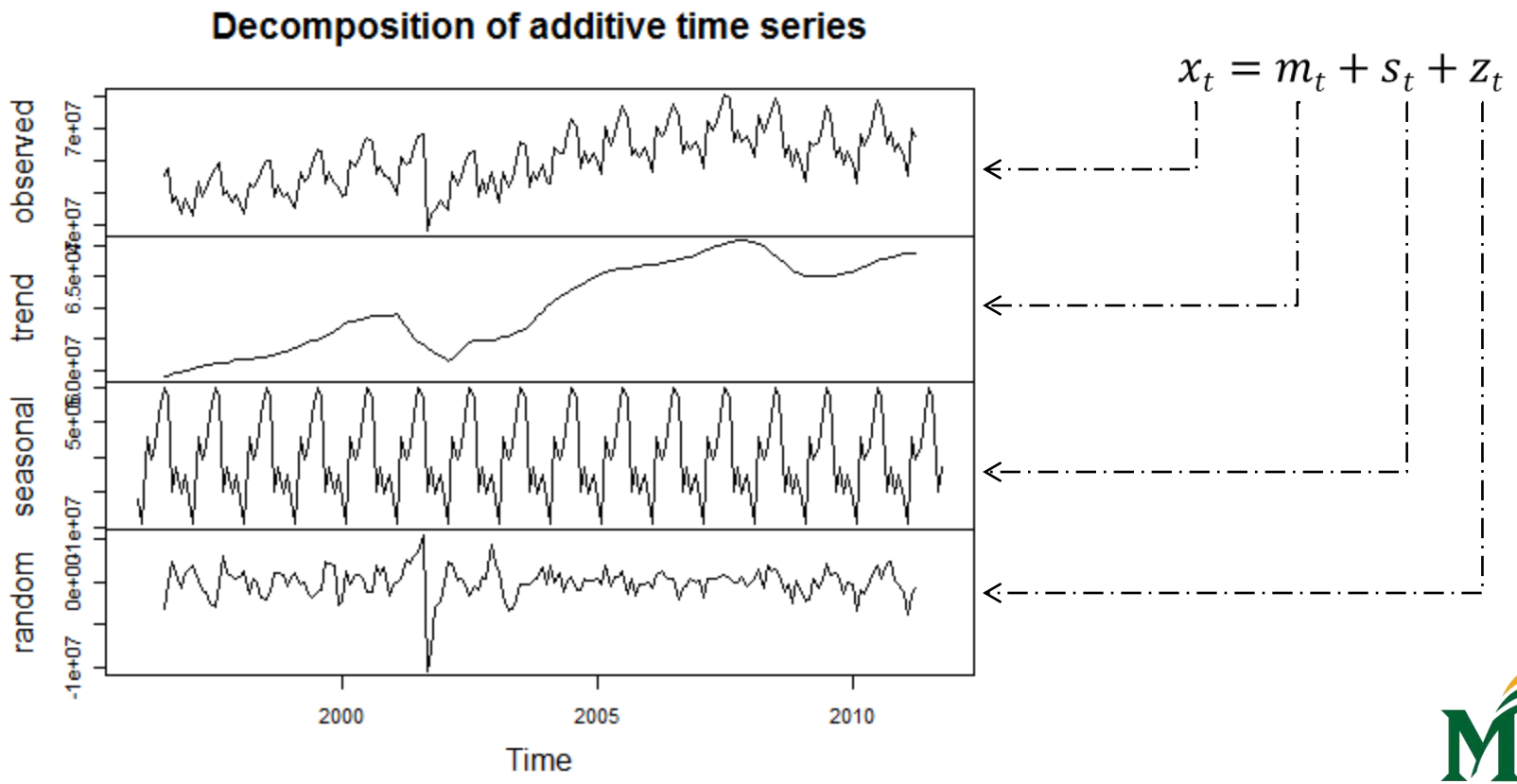
$$\log(x_t) = m'_t + s'_t + z'_t$$

If $\{x_t\}$ are positive.

GEORGE MASON UNIVERSITY

Time Series Decomposition in R

- *> rpm.decom=decompose(rpm.ts)*
- *> plot(rpm.decom)*



GEORGE MASON UNIVERSITY

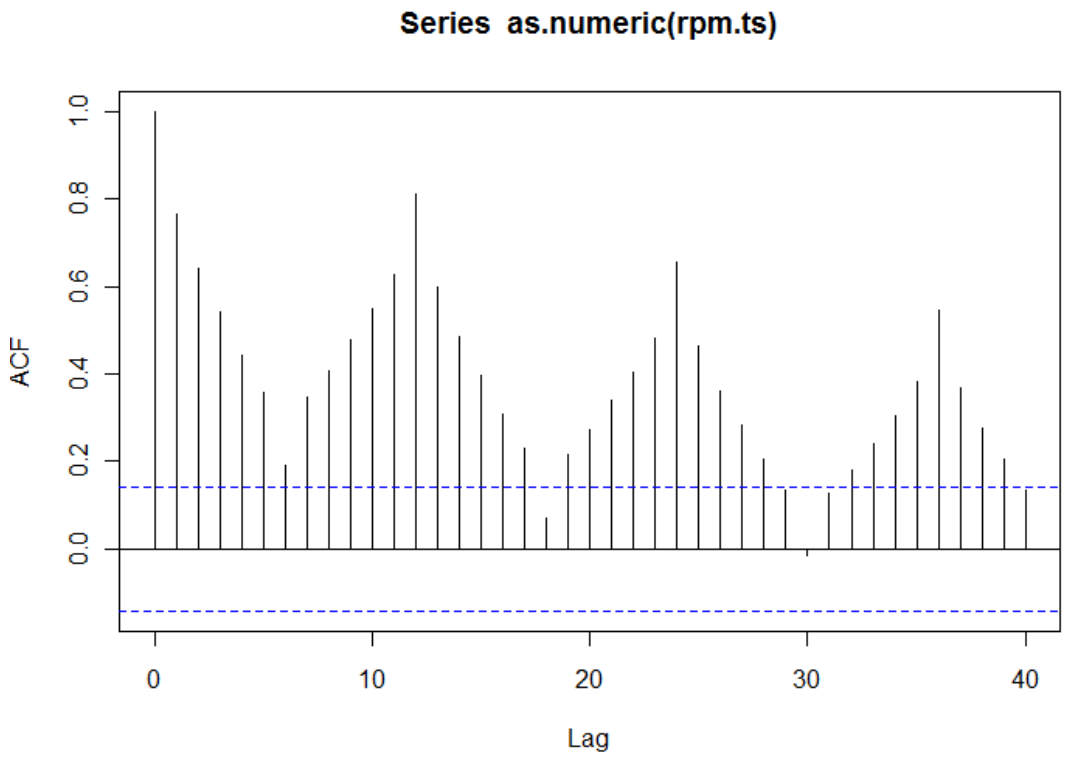
Auto-Correlation

- Population auto-covariance:
  $\Upsilon_k = E[(x_t - u)(x_{t+k} - u)]$
- Population auto-correlation: $\rho_k = \frac{\Upsilon_k}{\sigma^2}$, σ^2 is population variance
- Sample auto-covriance:
  $c_k = \frac{1}{n}\sum_{t=1}^{n-k}(x_t - \bar{x})(x_{t+k} - \bar{x})$

GEORGE MASON UNIVERSITY

AutoCorrelation and Correlogram

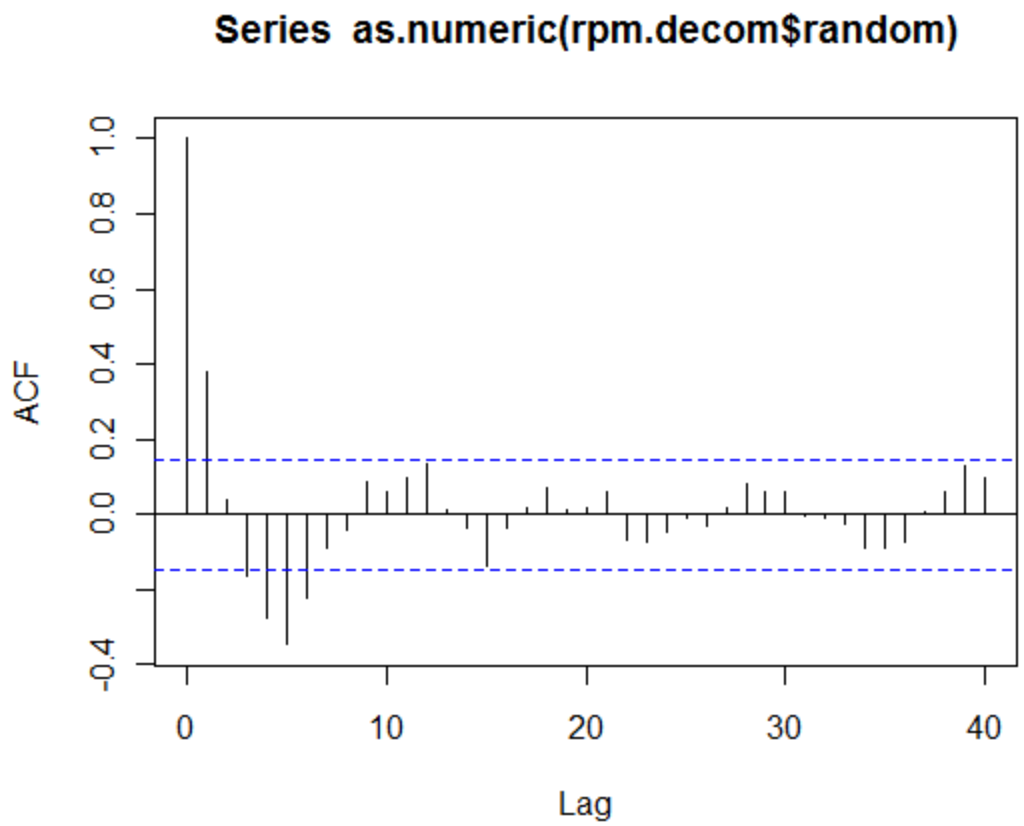
- *> rpm.acf=acf(as.numeric(rpm.ts),lag.max=40)*



GEORGE MASON UNIVERSITY

Correlogram after Decomposition

- *>acf(as.numeric(rpm.decom$random),na.action=na.omit,lag.max=40)*



Regression

- Trends: stochastic trends, deterministic trends
- Deterministic trends and seasonal variation can be modeled using regression
- Deterministic trends are often used for prediction
- Time series regression differs from standard regression as time series tends to be serially correlated

GEORGE MASON UNIVERSITY

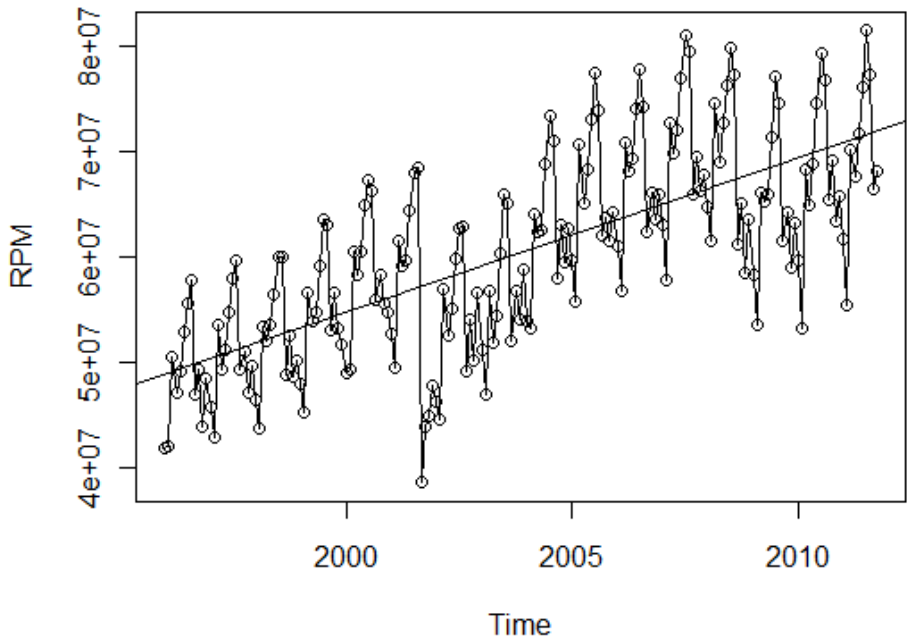
Linear Models

- $x_t = a_0 + a_1 u_{1,t} + a_2 u_{2,t} + \cdots + amu_{m,t} + zt$
- "Linear" is referenced to the summation of model parameters
- Examples:
  - $x_t = a_0 + a_1 t + a_2 t^2 + \cdots + a_p t_p + z_t$
  - $x_t = a_0 + a_1 t + z_t$

GEORGE MASON UNIVERSITY

Fit A Linear Regression Model

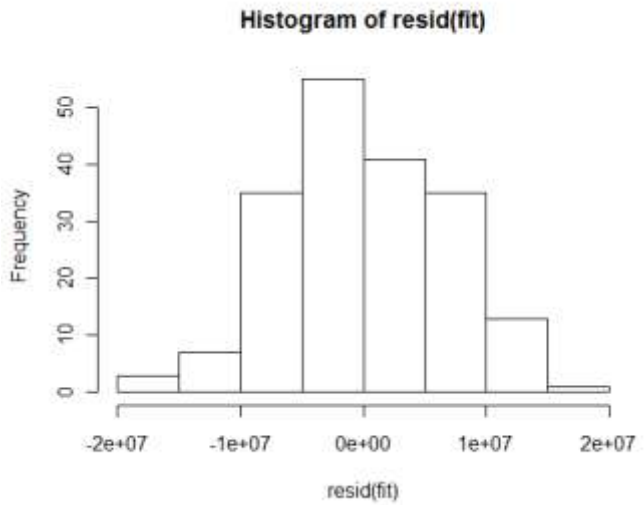
- > *fit = lm(rpm.ts ~ time(rpm.ts))*
- > *plot(rpm.ts, type="o", ylab="RPM")*
- > *abline(fit)*



GEORGE MASON UNIVERSITY

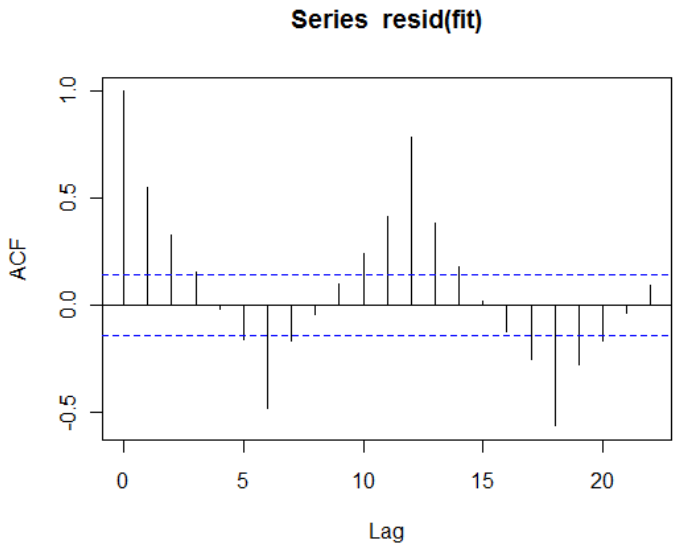
Diagnostic Plots

- > *hist(resid(fit))*



- > *acf(resid(fit))*
- > *AIC(fit)*

```
> AIC(fit)
[1] 6518.944
```



GEORGE MASON UNIVERSITY

Linear Model with Seasonal Variables

- Additive Seasonal Indicator

$$x_t = m_t + s_t + z_t$$

- For example, a time series with monthly observation starting from January:

$$x_t = \alpha_1 t + s_t + z_t = \begin{cases} \alpha_1 t + \beta_1 + z_t & t = 1, 13, \ldots \\ \alpha_1 t + \beta_2 + z_t & t = 2, 14, \ldots \\ \vdots & \\ \alpha_1 t + \beta_{12} + z_t & t = 12, 24, \ldots \end{cases}$$

GEORGE MASON UNIVERSITY

Linear Model with Seasonal Variables - *R*

- > *Seas = cycle(rpm.ts)*
  - Gives the positions in the cycle of each obsv.
- > *Time = time(rpm.ts)*
  - Creates the vector of times at which *rpm.ts* was sampled
- > *rpm.lm = lm(rpm.ts~0+Time+factor(Seas))*
  - Fit rpm.ts to the linear model with seasonal variables

GEORGE MASON UNIVERSITY

Take a Look at *rpm.lm*

- *> summary(rpm.lm)*

```
> summary(rpm.lm)

Call:
lm(formula = rpm.ts ~ 0 + Time + factor(Seas))

Residuals:
      Min        1Q    Median        3Q       Max
-13851942  -2461380    273581   2822312   5518156

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
Time            1.433e+06  5.730e+04   25.00   <2e-16 ***
factor(Seas)1  -2.816e+09  1.148e+08  -24.53   <2e-16 ***
factor(Seas)2  -2.820e+09  1.148e+08  -24.56   <2e-16 ***
factor(Seas)3  -2.808e+09  1.148e+08  -24.45   <2e-16 ***
factor(Seas)4  -2.811e+09  1.148e+08  -24.48   <2e-16 ***
factor(Seas)5  -2.809e+09  1.148e+08  -24.46   <2e-16 ***
factor(Seas)6  -2.804e+09  1.148e+08  -24.42   <2e-16 ***
factor(Seas)7  -2.800e+09  1.148e+08  -24.39   <2e-16 ***
factor(Seas)8  -2.802e+09  1.148e+08  -24.40   <2e-16 ***
factor(Seas)9  -2.815e+09  1.148e+08  -24.51   <2e-16 ***
factor(Seas)10 -2.812e+09  1.149e+08  -24.48   <2e-16 ***
factor(Seas)11 -2.815e+09  1.148e+08  -24.52   <2e-16 ***
factor(Seas)12 -2.813e+09  1.148e+08  -24.50   <2e-16 ***
---
Signif. codes:  0 ?**?0.001 ?*?0.01 ??0.05 ??0.1 ??1

Residual standard error: 3606000 on 177 degrees of freedom
Multiple R-squared: 0.9968, Adjusted R-squared: 0.9965
F-statistic:  4187 on 13 and 177 DF,  p-value: < 2.2e-16
```

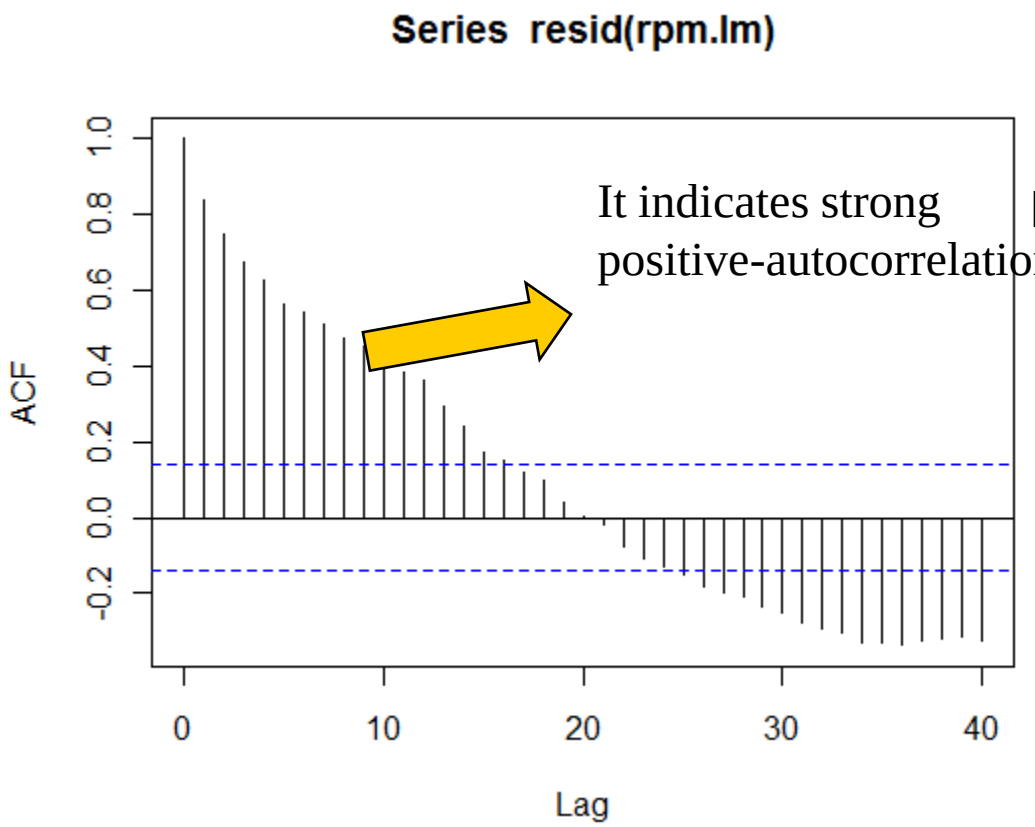
Model

Residuals from fitting

Coefficients

Correlogram of rpm.lm Residual

- *> acf(resid(rpm.lm),lag.max=40)*



It indicates strong positive-autocorrelation

Residuals are not pure random numbers, so it should be further modeled

GEORGE MASON UNIVERSITY

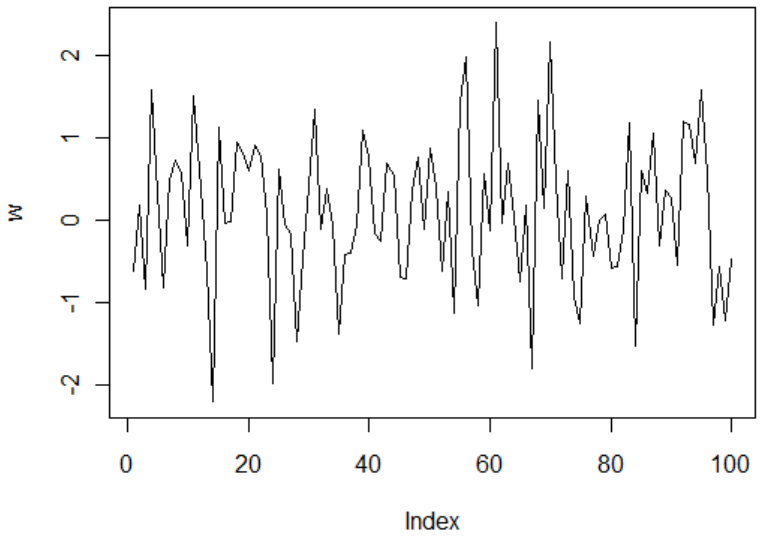
How Random is Random? - White Noise

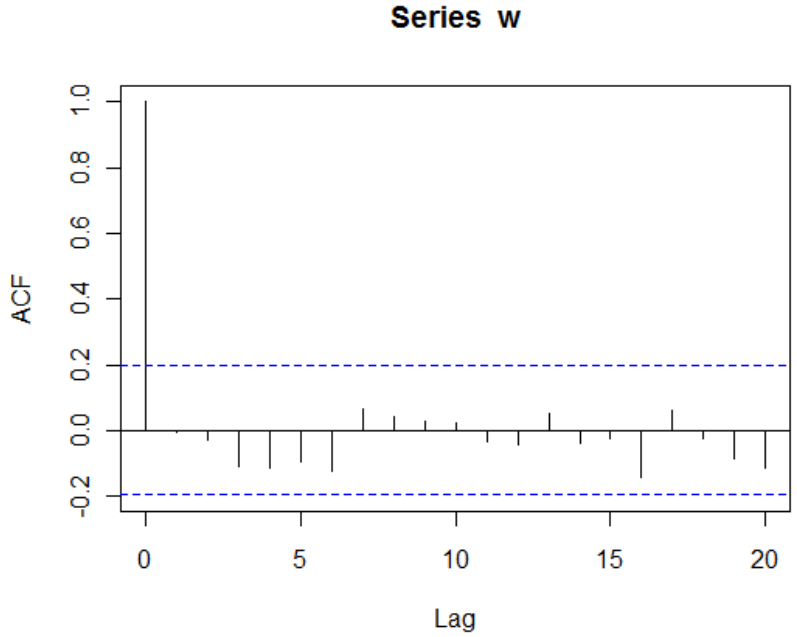
- A time series $\{w_t: t = 1, 2, \ldots, n\}$ is discrete white noise (DWN) if the variables $w_1, w_2, \ldots, w_n$ are **independent** and **identically distributed** with a **mean of zero**.
- This implies that the variables all have the same variance σ^2 and $\text{Cor}(w_i, w_j) = 0$ for all $i \neq j$.
- If, in addition, the variables also follow a normal distribution (i.e., $w_t \sim \text{N}(0, \sigma^2)$) the series is called **Gaussian white noise**.

GEORGE MASON UNIVERSITY

Simulate White Noise in *R*

- > *set.seed(1)*
- > *w = rnorm(100)*
- > *plot(w, type = "l")*
- > *acf(w)*





GEORGE MASON UNIVERSITY

Random Walk

Let $\{x_t\}$ be a time series. Then $\{x_t\}$ is a random walk if

- $x_t = x_{t-1} + w_t$

where $\{w_t\}$ is a white noise series. Substituting $x_{t-1} = x_{t-2}+w_{t-1}$ and then substituting for x_{t-2}, followed by x_{t-3} and so on gives:

- $x_t = w_t + w_{t-1} + w_{t-2} + \ldots$

In practice, the series will start at some time

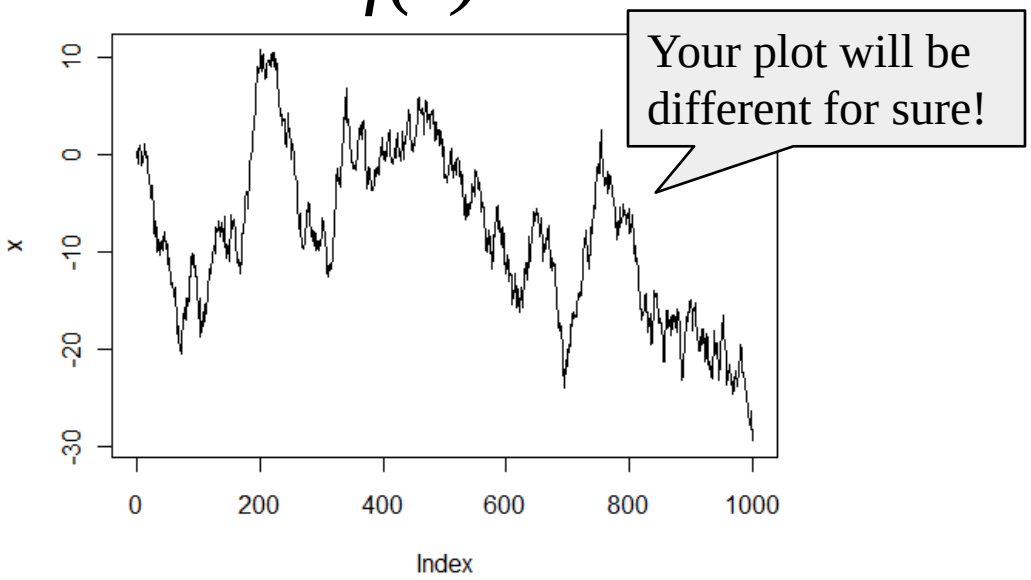
$t = 1$. Hence,

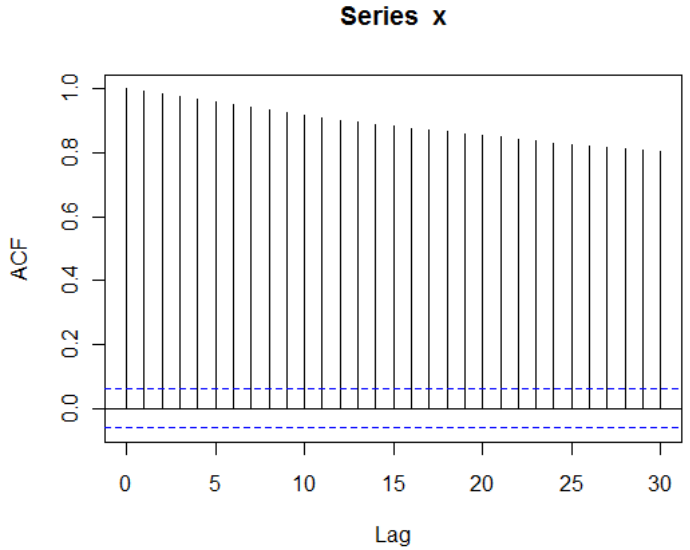
- $x_t = w_1 + w_2 + \ldots + w_t$

GEORGE MASON UNIVERSITY

Simulate A Random Walk in *R*

- *> x <- w <- rnorm(1000)*
- *> for (t in 2:1000) x[t] <- x[t - 1] + w[t]*
- *> plot(x, type = "l")*
- *> acf(x)*





GEORGE MASON UNIVERSITY

Stationarity

- Strict stationary
  - the joint statistical distribution of $x_{t_1}, \ldots, x_{t_n}$ is the same as the joint distribution of $x_{t_{m+1}}, \ldots, x_{t_{m+n}}$ for all $t_1, \ldots, t_n$ and m, so that the distribution is unchanged after an arbitrary time shift.
- Second-order stationary
  - Mean and standard deviation are constant in time
  - Autocorrelation depends only on the lag
- Non-stationary time series

Auto-Regressive (AR) Models

- The series $\{x_t\}$ is an autoregressive process of order p, abbreviated to AR(p), if

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \ldots + \alpha_p x_{t-p} + w_t$$

where $\{w_t\}$ is white noise and the α_i are the model parameters with $\alpha_p \neq 0$ for an order p process.

- The model is a regression of x_t on past terms from the same series; hence the use of the term ‘autoregressive’.

GEORGE MASON UNIVERSITY

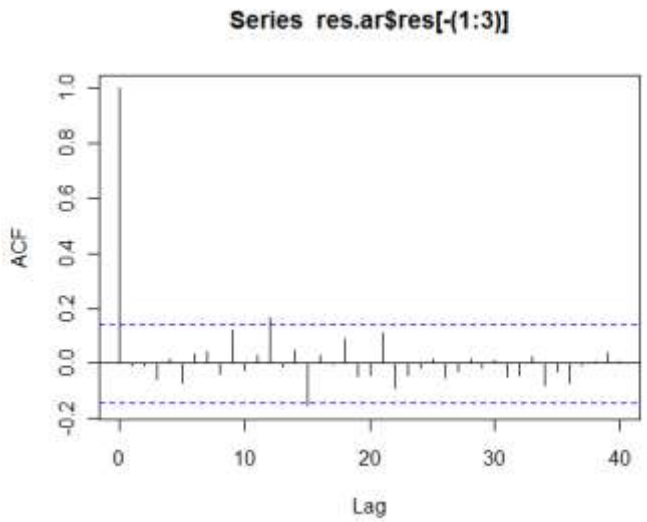
Fit an AR Model

- > *res.ar=ar(resid(rpm.lm),method="ols")*
- > *res.ar*

$$x_t = 0.7153x_{t-1} + 0.1010x_{t-2} + 0.0579x_{t-3} + w_t$$

```
> res.ar

Call:
ar(x = resid(rpm.lm), method = "ols")

Coefficients:
     1       2       3
0.7153  0.1010  0.0579

Intercept: -3341 (135845)

Order selected 3  sigma^2 estimated as  3.45e+12
```

- > *acf(res.ar$res[-(1:3)])*



OLS: Ordinary Least Square

GEORGE MASON UNIVERSITY

Moving Average Model

- A moving average (MA) process of order q is a linear combination of the current white noise term and the q most recent past white noise terms and is defined by
- $x_t = w_t + \beta_1 w_{t-1} + \cdots + \beta_q w_{t-q}$
- $\{x_t\}$ is a stationary process because $\{w_t\}$'s are stationary

GEORGE MASON UNIVERSITY

Simulate an MA Process

- *> set.seed(1)*
- *> b <- c(0.8, 0.6, 0.4)*
- *> x <- w <- rnorm(1000)*
- *> for (t in 4:1000) {*

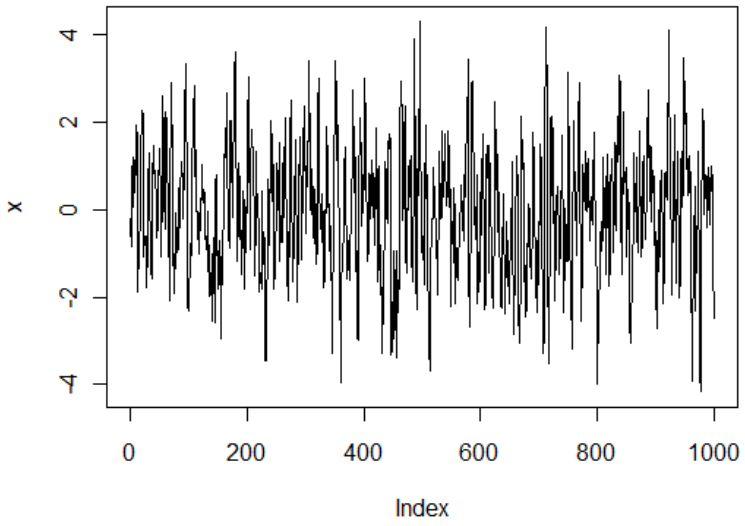
*for (j in 1:3) x[t] <- x[t] + b[j] * w[t - j]*

}

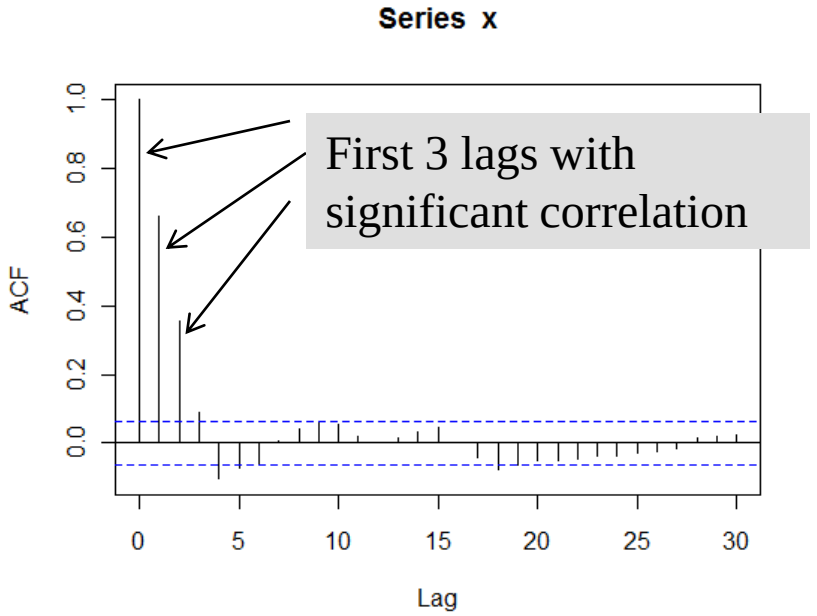
- *> plot(x, type = "l")*
- *> acf(x)*

GEORGE MASON UNIVERSITY

Plot Results



Simulated MA Process



Correlogram of the Simulated MA Process

GEORGE MASON UNIVERSITY

ARMA Model

- A time series $\{x_t\}$ follows an autoregressive moving average (ARMA) process of order (p, q), denoted ARMA(p, q), when

$$x_t = \alpha_1 x_{t-1} + \alpha_2 x_{t-2} + \cdots + \alpha_p x_{t-p} + \beta_1 w_{t-1} + \beta_2 w_{t-2} + \cdots + \beta_q w_{t-q}$$

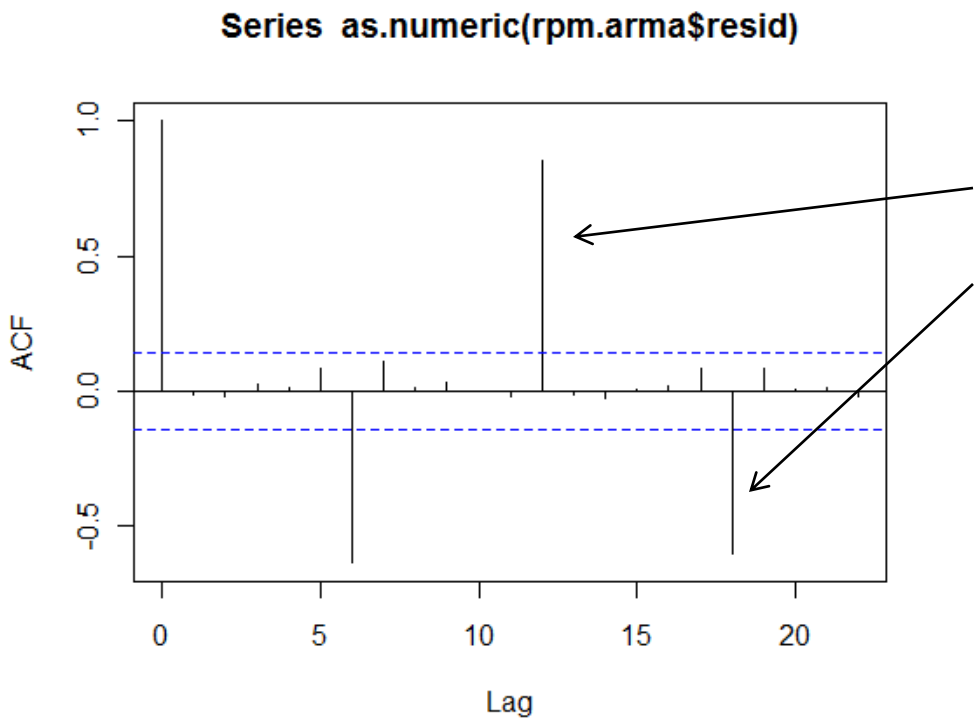
where $\{w_t\}$ is white noise

- AR(p) is ARMA$(p,0)$

GEORGE MASON UNIVERSITY

Fit An ARMA Model

- *> rpm.arma=arima(rpm.ts,order=c(1,0,1))*
- *> acf(as.numeric(rpm.arma$resid))*



Strong seasonal information left in residuals

GEORGE MASON UNIVERSITY

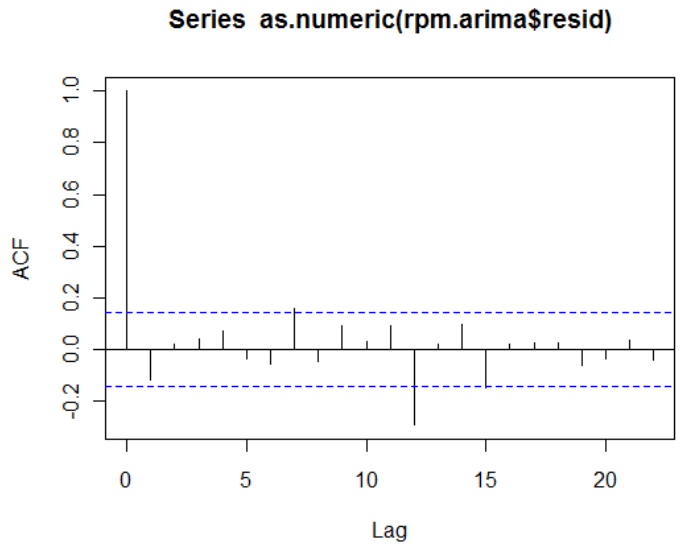
ARIMA and SARIMA Model

- A time series $\{x_t\}$ follows an ARIMA(p, d, q) process if the d^{th} differences of the $\{x_t\}$ series are an ARMA(p, q) process
- SARIMA is Seasonal ARIMA model which extends ARIMA model with seasonal terms

GEORGE MASON UNIVERSITY

Fit A SARIMA Model

- *> rpm.arima=arima(rpm.ts,order=c(1,1,1), seas=list(order=c(1,0,0),12))*
  - *First c(1,1,1):* AR(1), first-order difference, MA(1)
  - *Second c(1,0,0):* seasonal terms on AR process, frequency 12
- *> acf(as.numeric(rpm.arima$resid))*



GEORGE MASON UNIVERSITY

Forecast

- Predicting future values of a time series, x_{n+m}, using the set of present and past values of the time series, $x=\{x_n, x_{n-1}, \ldots, x_1\}$
- The minimum mean square error predictor of x_{n+m} is $x_{n+m}^{n} = E(x_{n+m}|x)$
- *predict(model, newdata)* method
  - *model:* a model object used for prediction
  - *newdata*: value of explanatory variables

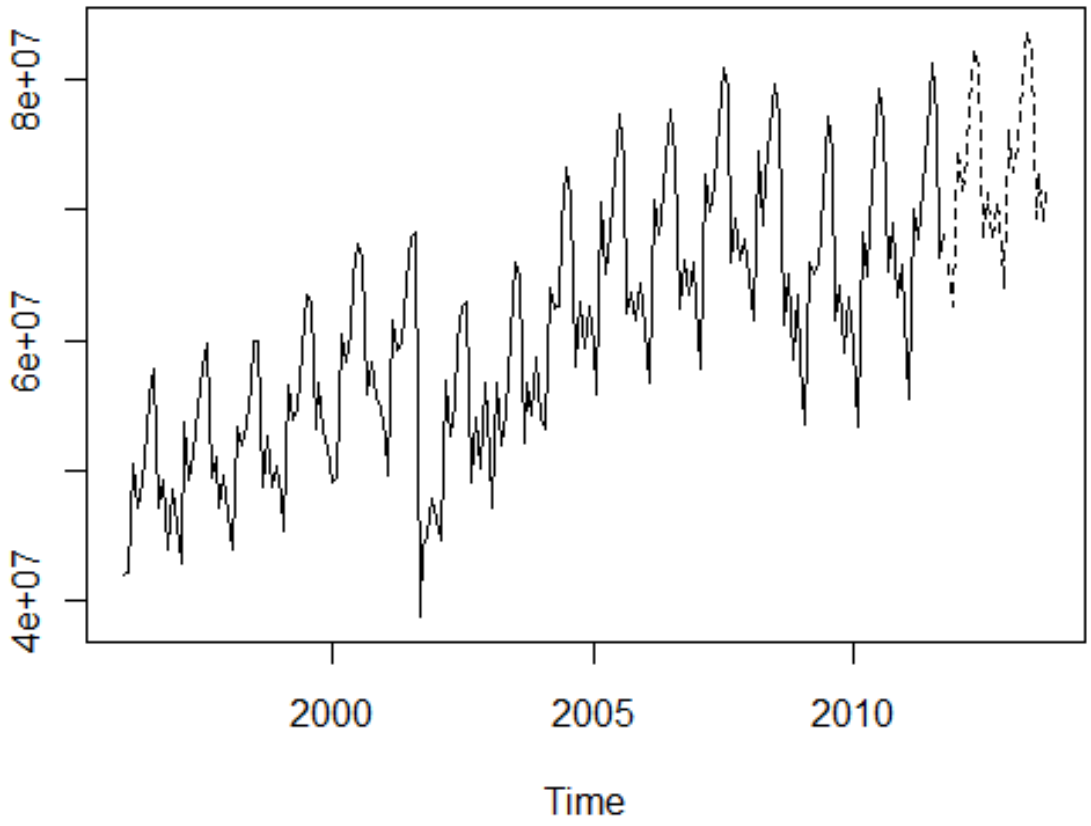
GEORGE MASON UNIVERSITY

Forecast in *R*

- *> new.t <- seq(2011.750, len = 2 * 12, by = 1/12)*
- *> new.dat <- data.frame(Time = new.t, Seas = rep(1:12, 2))*
- *> rpm.pred=ts(predict(rpm.lm, new.dat)[1:24],start=c(2011,11),freq=12)*
- *> ts.plot(rpm.ts,rpm.pred,lty=1:2)*

GEORGE MASON UNIVERSITY

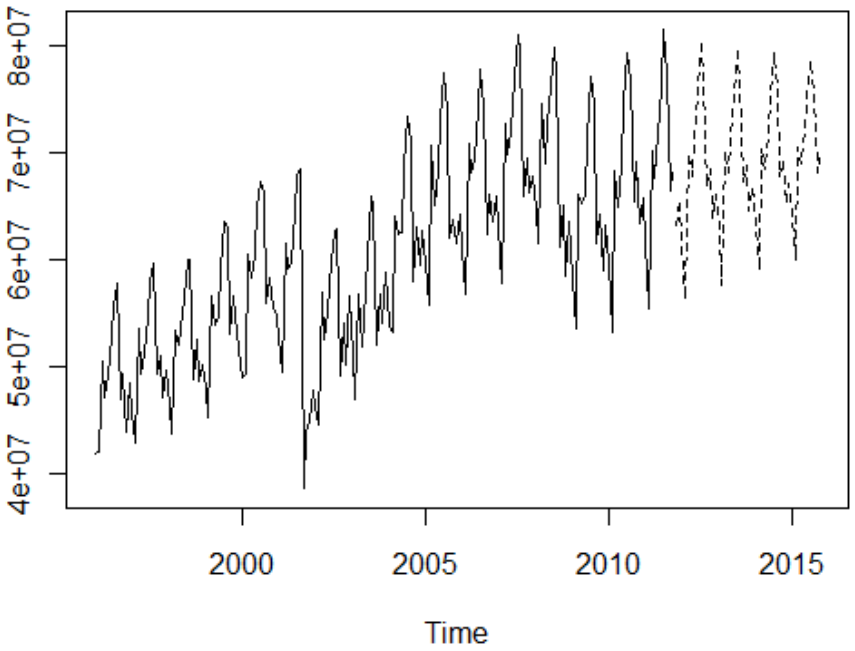
Plot of Forecast



GEORGE MASON UNIVERSITY

Forecast with SARIMA Model

- *> ts.plot(cbind(window(rpm.ts,start = c(1996,1)), predict(rpm.arima,48)$pred), lty = 1:2)*



Homework

- Download and Install *R* with Rstudio
- Read *An Introduction to R*
- Download System Passenger - Revenue Aircraft Miles Flown (000) (Jan 1996 - Oct 2011) data from BTS
- Read the data into *R* using Rstudio
- Create a time series plot of the data, and plot its auto-correlation correlogram
- Decompose the time series and save the plot

GEORGE MASON UNIVERSITY

Homework (Ctd.)

- Construct a linear regression model without seasonal factors, and plot the correlogram of the model's residual data
- Construct a linear regression model with seasonal factors, and identifies the characteristics of the model residual.
- Fit an AR model to the model residual of the above model
- Forecast the time series data into next 24 months using the seasonal model

GEORGE MASON UNIVERSITY

Exam Questions

- What are the data elements in a time series?
- What does auto-correlation mean?
- What are white noise and random walk?
- What are stationary models and non-stationary models?

GEORGE MASON UNIVERSITY