Time Series Analysis

Stationarity, differencing and linear modeling

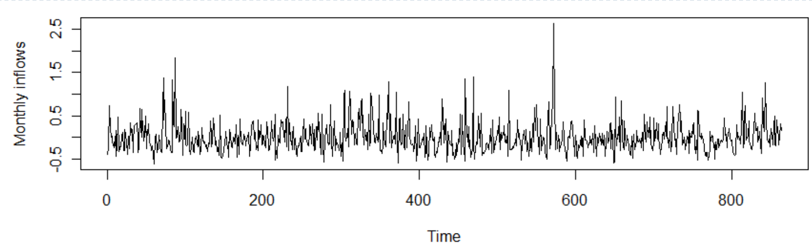
Concepts

- Elements of exploratory time series analysis
  - Stationarity and differencing
- Models of time series
  - Linear models and stochastic processes

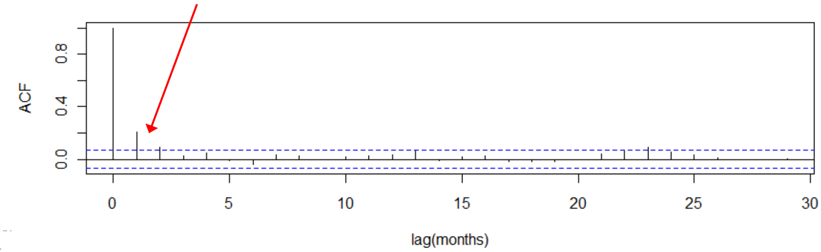
What is a stationary time series?

- "…a time series is said to be stationary if there is no systematic change in mean (no trend), if there is no systematic change in variance and if strictly periodic variations have been removed." (Chatfield: 13)
- Correlogram has very few significant spikes at very small lags and cuts off drastically/dies down quickly (at 2 or 3 lags)

Example: Stationary series



The correlogram has only a couple of significant spikes at lags 1 and 2
So the correlogram confirms that the series is stationary



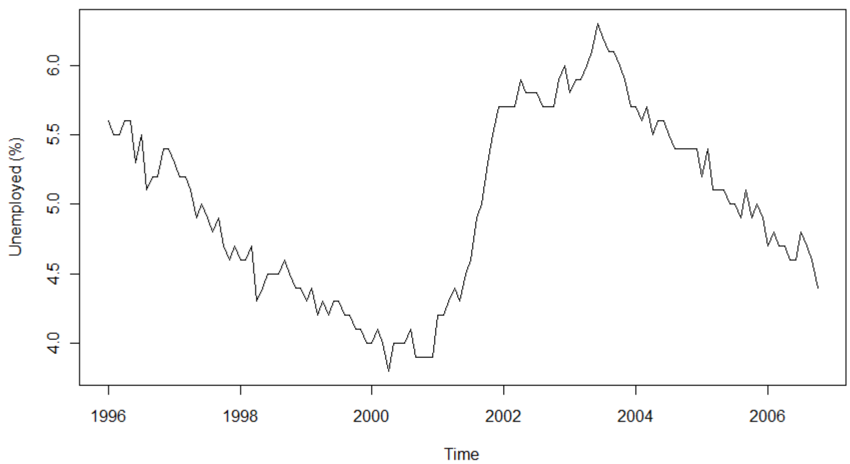
Why do we care about stationarity?

- Most time series models only work if data are stationary
  - Called "stationary time series models"
- So, first step in an analysis is to check for evidence of a trend or seasonality
- Depending on the type of trend or seasonality, we can remove nonstationarity in different ways
  - Simple differencing subtracts trend/seasonal component from original series
  - Regression can model trend/seasonal components
    - Use the time series of residuals in model

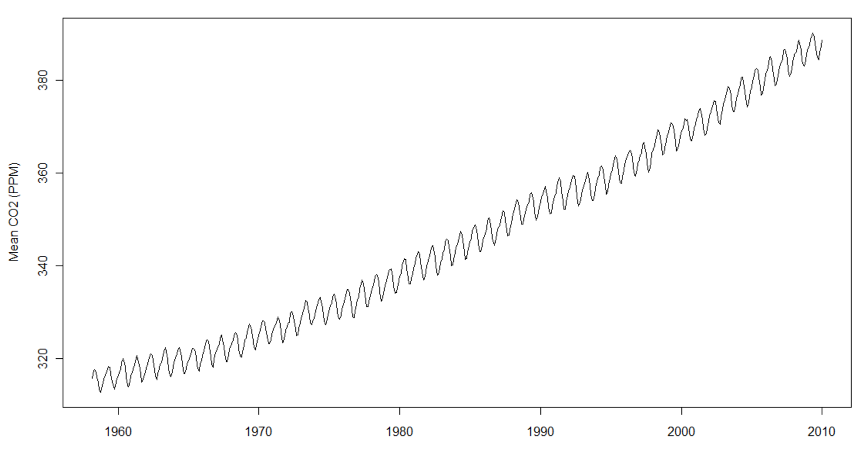
Stochastic vs. deterministic

- Used to describe the "quality" of nonstationarity of a series
  - Help determine what type of TS approach to take
- Stochastic = inexplicable changes in direction
  - Often found in economic processes, sometimes climate
  - "Random walk" process
  - Use differencing and autoregressive models
- Deterministic = plausible physical explanation for a trend or seasonal cycle
  - Increase in population, orbit of the earth
  - Use regression models

Example: stochastic TS



Example: deterministic TS



To make a series stationary

1. Check if there is variance that changes with time
   - YES → make variance constant with log or square root transformation
2. Is the trend stochastic or deterministic?
   - If stochastic → use differencing
   - If deterministic → use regression
3. Remove the trend in mean with:
   - 1st/2nd order differencing
   - Smoothing and differencing (seasonality)
4. If there is seasonality in the data:
   - Moving average and differencing
   - Smoothing

Differencing

Making non-stationary time series stationary

Stochastic seasonality and trends

- If the series has a stable long-run trend and tends to revert to the trend line following a disturbance
  - **trend-stationary**
- Someties even de-trending is not sufficient to make the series stationary, in which case it may be necessary to transform it into a series of period-to-period and/or season-to-season *differences*.
- If the mean, variance, and autocorrelations of the original series are not constant in time, even after detrending, perhaps the statistics of the *changes* in the series between periods or between seasons *will* be constant. Such a series is said to be **difference-stationary**.

Differencing

- Transformation of the series to a new time series where the values are the differences between consecutive values
  - Procedure may be applied consecutively more than once, giving rise to the "first differences", "second differences", etc.
- The first order differences are computed as:

  $d^1{}_t = x_t - x_{t-1}$

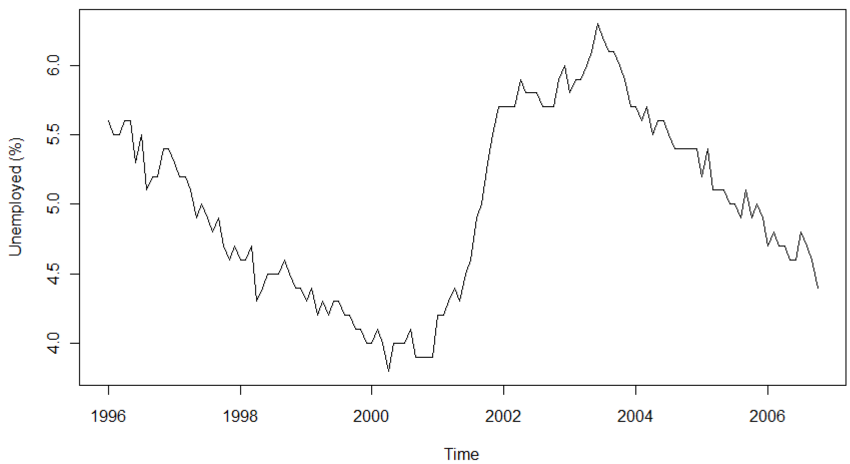
  The second order differences can the be computed as:

  $d^2{}_t = d^1{}_t - d^1{}_{t-1}$

  Seasonal differences can be computed as:

  $d^{12}{}_t = d_t - d_{t-12}$

US Unemployment



Checking for stationarity

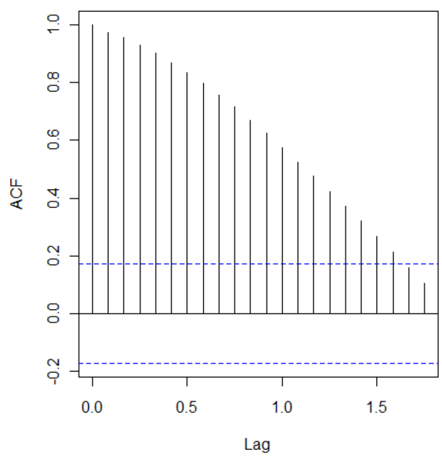
```
> install.packages("tseries")
> library(tseries)

> kpss.test(US.ts, null = "Trend")

        KPSS Test for Trend Stationarity
data:  US.ts
KPSS Trend = 0.5118, Truncation lag parameter = 2, p-value = 0.01

> adf.test(US.ts, alternative = "stationary") # can also add k=n

        Augmented Dickey-Fuller Test
data:  US.ts
Dickey-Fuller = -1.2823, Lag order = 5, p-value = 0.8749
```

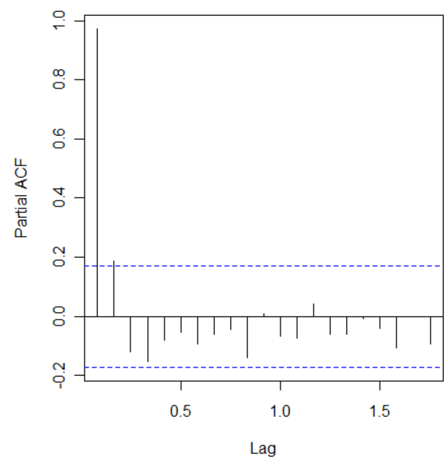
Can reject H_0 - means the series is non-stationary

Cannot reject H_0 - means the series is non-stationary

Example

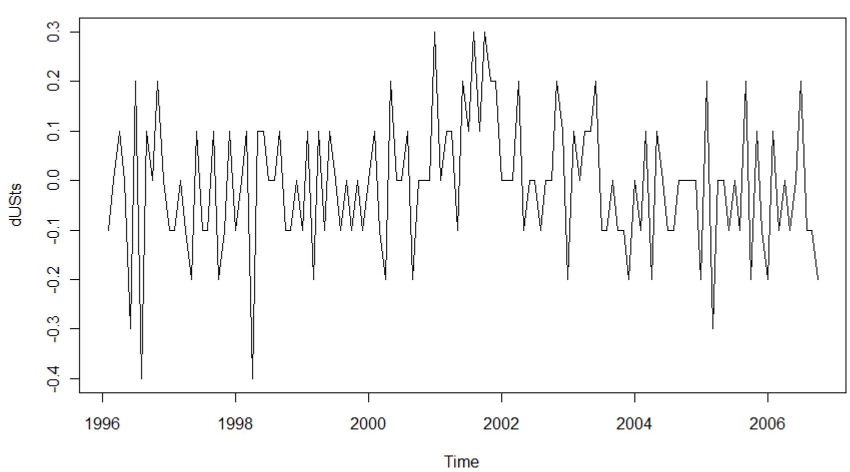
```
> acf(US.ts)
> pacf(US.ts)
```



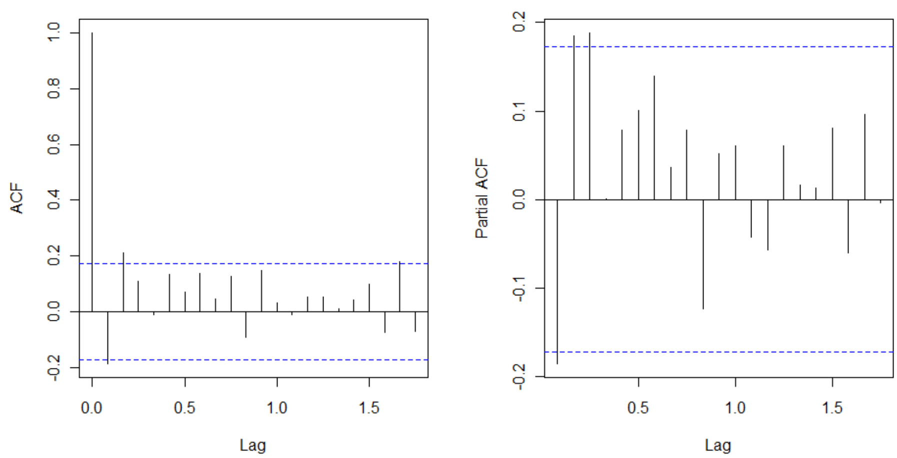


Example

```
> dUSts<-diff(US.ts)
> plot(dUSts)
```



Example

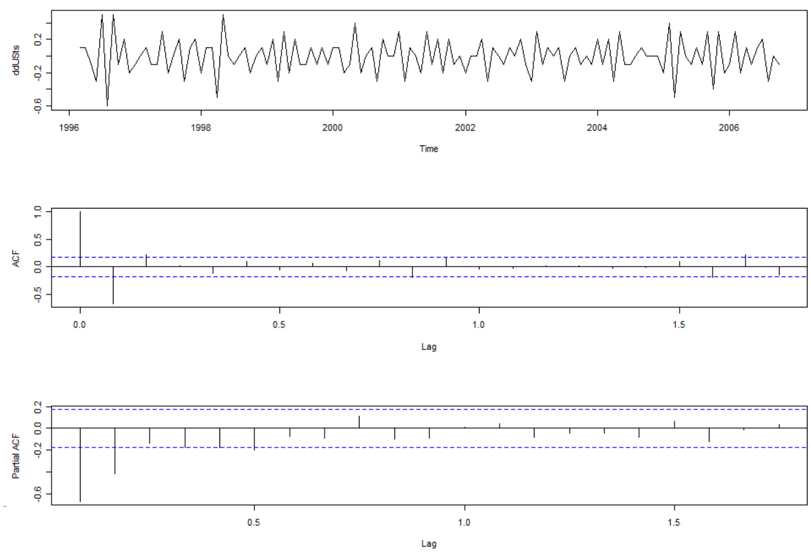


Checking for stationarity

```
> kpss.test(dUSts, null = "Trend")

        KPSS Test for Trend Stationarity
data:  dUSts
KPSS Trend = 0.3197, Truncation lag parameter = 2, p-value = 0.01

> adf.test(dUSts, alternative = "stationary")

        Augmented Dickey-Fuller Test
data:  dUSts
Dickey-Fuller = -3.059, Lag order = 5, p-value = 0.1363
```

Example

```
> ddUSts<-diff(dUSts)
```



Example

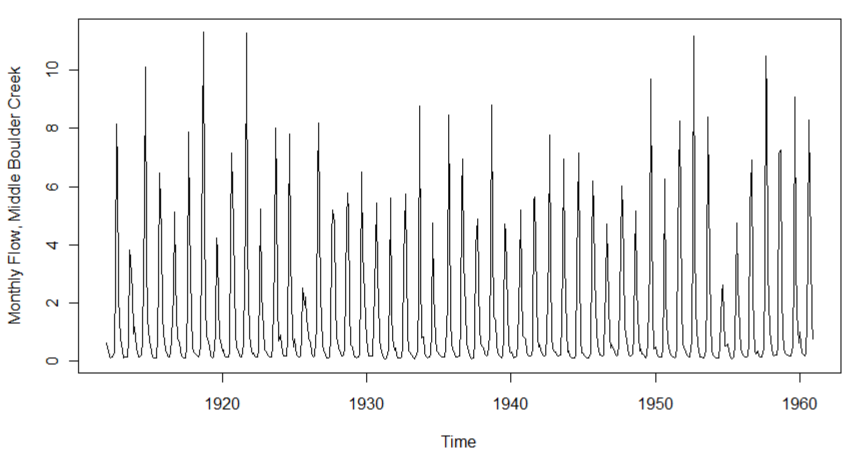
```
> kpss.test(ddUSts, null = "Trend")

        KPSS Test for Trend Stationarity
data:  ddUSts
KPSS Trend = 0.0139, Truncation lag parameter = 2, p-value = 0.1

> adf.test(ddUSts, alternative = "stationary")

        Augmented Dickey-Fuller Test
data:  ddUSts
Dickey-Fuller = -8.2463, Lag order = 5, p-value = 0.01
```

Seasonality and smoothing

- Smoothing is often used to remove an underlying signal or trend (such as a seasonal cycle)

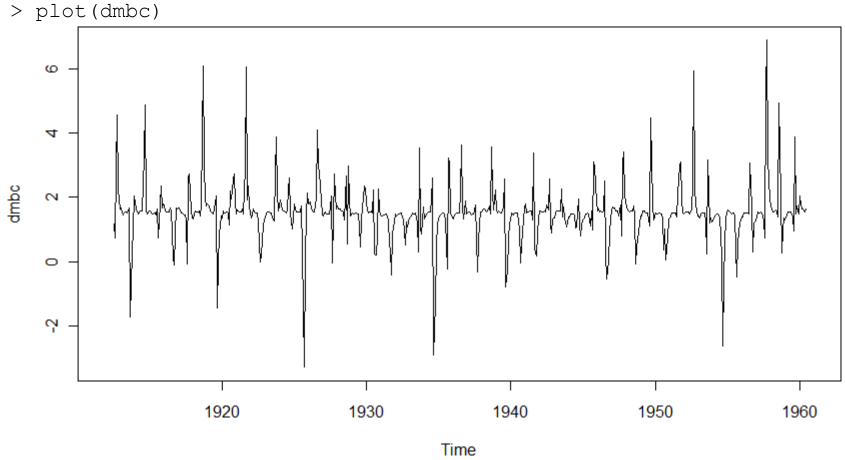
- Common method is the centered moving average
  - Average a specified number of time series values around each value in the time series
  - Length of moving average is chosen to average out seasonal effects
    - Seasonal = 12 point moving average
    - Quarterly = 4 point moving average

Example: river flow rates



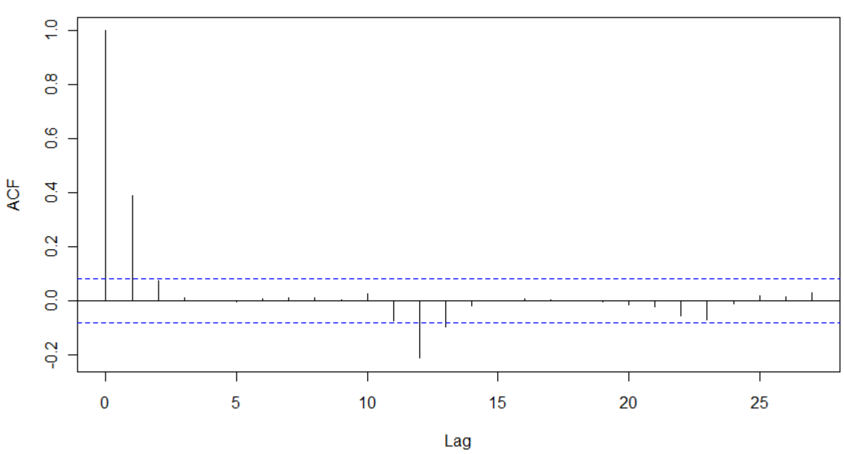
Example

```
> mbc.d<-decompose(mbc)
> dmbc<-(mbc.d$trend+mbc.d$rand)
> plot(dmbc)
```



Example

```
> acf(dmbc[7:582]) # moving averages leave out first/last 6 obs
```



What now? Modeling a stationary series

- Once you have a stationary series, you can model it using an autoregressive (AR) model

$$x_t = \beta_1 x_{t-1} + \ldots + \beta_p x_{t-p} + r_t$$

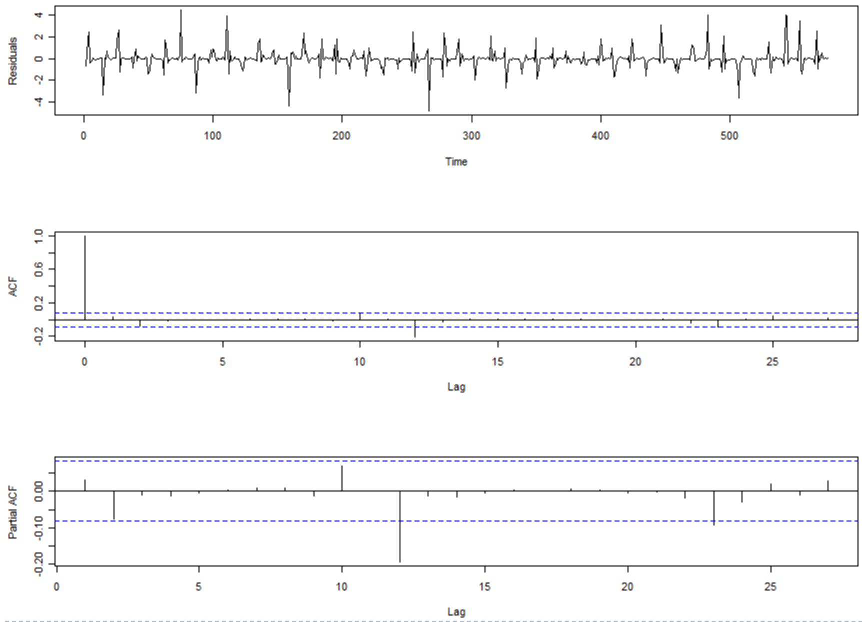
where:
x time series observation
p process or the order (lag)
r random element (error)

  - We simply regress x on lagged x
- If the model successfully captures the dependence structure in the data the residuals should look random
  - Check the residuals from the AR model for any "left-over" dependence

Autoregressive model

```
> mbc.ar<-ar(dmbc[7:582], method="mle")
> mean(dmbc[7:582])
[1] 1.530268

> mbc.ar

Call:
ar(x = dmbc[7:582], method = "mle")
Coefficients:
     1
0.3896

Order selected 1  sigma^2 estimated as  0.6637

> plot(mbc.ar$res); acf(mbc.ar$res[7:582]); pcf(mbc.ar$res[7:582])
```

$$z_t = 1.55 + 0.38(z_{t-1} - 1.55)$$



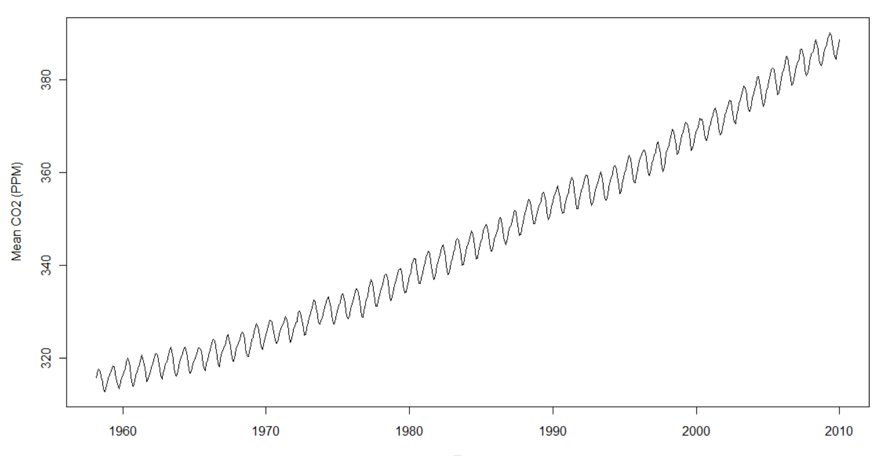
Regression

Making non-stationary time series stationary
Working with deterministic time series

Regression

- Deterministic trends can be modeled using regression techniques
  - Both linear and nonlinear
- Seasonal trends can be modeled using regression techniques
  - Indicator variables
  - Harmonic variables
- Residuals from models are the "random error" component
  - Use these for time series models

Example: Mona Loa CO_2 Concentrations



Fitting linear models to TS data

- Regress observation against time

  $X_t = t$

  - Typically use generalized least squares (GLS) because errors are correlated
    - Type of maximum likelihood method

- Examine coefficients and SE
- Examine correlogram of residuals

- Fit appropriate TS model to residuals

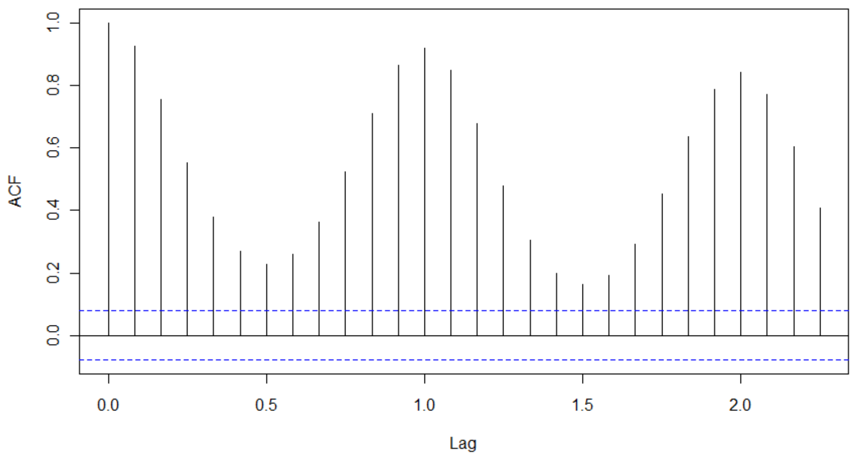
Example

```
> mllm<-lm(mlco2~mlco2.t)
> acf(resid(mllm))$acf[2]
[1] 0.9261262

> mlnlm<-gls(mlco2~mlco2.t, cor=corAR1(0.93))
> mlnlm
Generalized least squares fit by REML
  Model: mlco2 ~ mlco2.t
  Log-restricted-likelihood: -993.7159

Coefficients:
 (Intercept)      mlco2.t
-2488.298538     1.428678

Degrees of freedom: 623 total; 621 residual
Residual standard error: 3.551621
> confint(mlnlm)
                   2.5 %       97.5 %
(Intercept) -2686.601893 -2289.995183
mlco2.t         1.328734     1.528622
```

CIs for slope do not encompass 0
1) Estimates are significant
2) Trend is significant

Example

```
> acf(mlnlm$residuals)
```



Seasonal indicator (linear) model

- To take account of seasonal effects, add predictor variables for season
- Basically, a set of dummy variables that indicate the season, quarter, month, etc.

$$X_t = \alpha_1 T + S_t + z_t$$

Creating seasonal indicators

```
> Seas<-cycle(mlco2)
> Seas
     Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
1958           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
1961   1   2   3   4   5   6   7   8   9  10  11  12

> Time<-time(mlco2)
> Time
          Jan      Feb      Mar      Apr      May      Jun      Jul      Aug    …
1958                   1958.167 1958.250 1958.333 1958.417 1958.500 1958.583    …
1959 1959.000 1959.083 1959.167 1959.250 1959.333 1959.417 1959.500 1959.583    …
1960 1960.000 1960.083 1960.167 1960.250 1960.333 1960.417 1960.500 1960.583    …
1961 1961.000 1961.083 1961.167 1961.250 1961.333 1961.417 1961.500 1961.583    …
1962 1962.000 1962.083 1962.167 1962.250 1962.333 1962.417 1962.500 1962.583    …
```

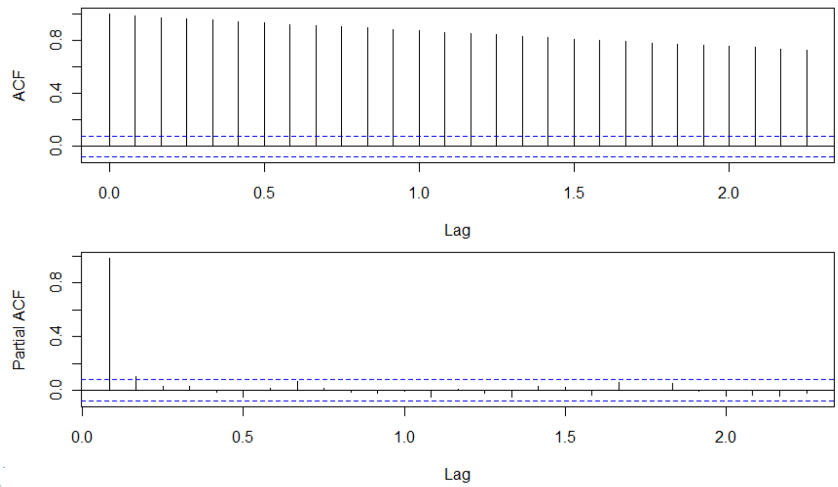
Results

```
> ml.gls<-gls(mlco2~Time+factor(Seas), cor=corAR1(0.985))
```

Variable	Coefficient
(Intercept)	-2492.85
Time	1.43
factor(Seas)2	0.63
factor(Seas)3	1.39
factor(Seas)4	2.51
factor(Seas)5	2.92
factor(Seas)6	2.27
factor(Seas)7	0.67
factor(Seas)8	-1.45
factor(Seas)9	-3.13
factor(Seas)10	-3.26
factor(Seas)11	-2.11
factor(Seas)12	-0.94

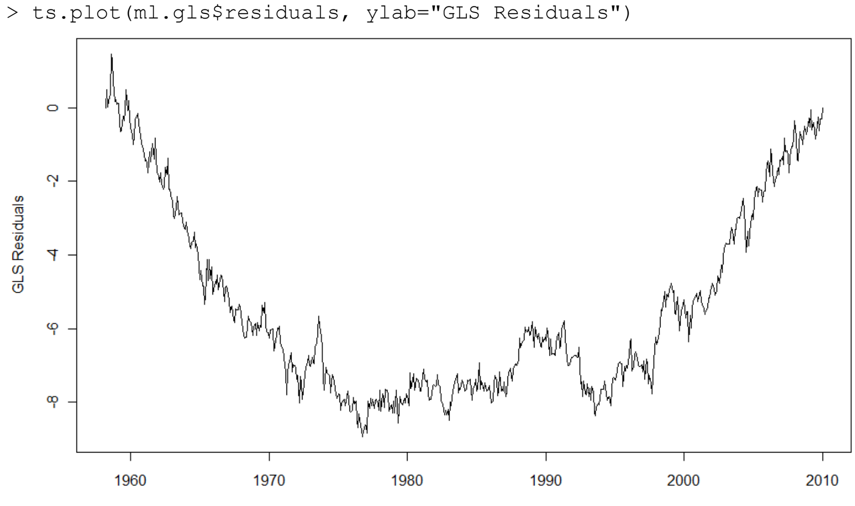
Results

```
> acf(ml.gls$residuals)
> pacf(ml.gls$residuals)
```



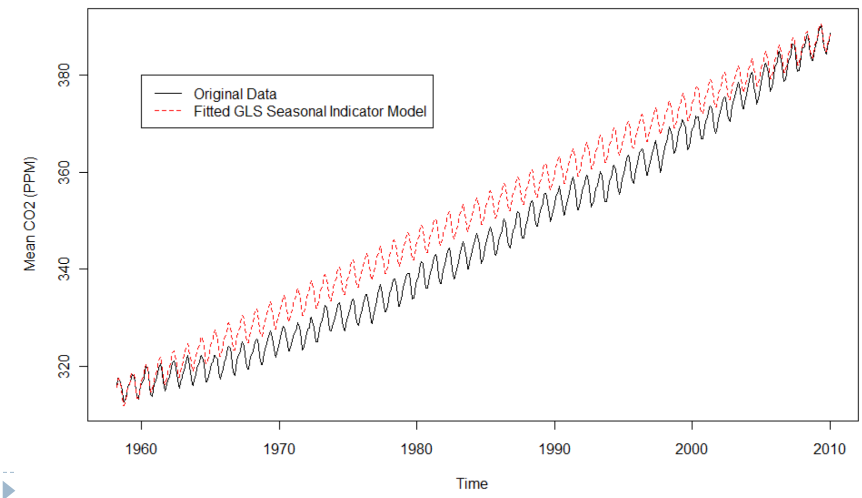
Results

```
> ts.plot(ml.gls$residuals, ylab="GLS Residuals")
```



Results

```
> ts.plot(cbind(mlco2, ml.gls$fitted),lty=1:2, col=c(1,2))
> legend(1960,380,c("Original", "Fitted"),col=c(1,2),lty=c(1, 2))
```



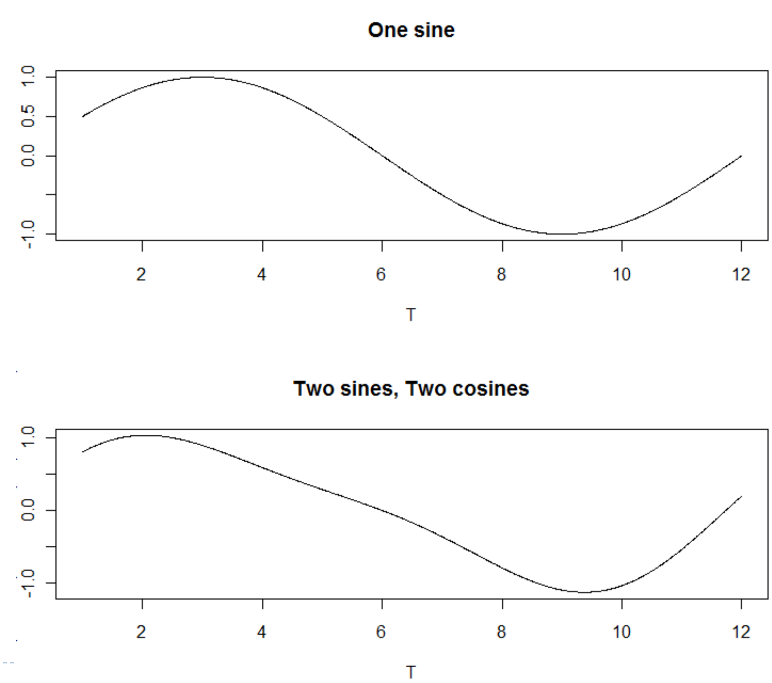
Results

```
> ml<-lm(mlco2~mlco2.t)
> AIC(ml)
[1] 3248.806

> ml.lm<-lm(mlco2~Time+factor(Seas))
> AIC(ml.lm)
[1] 2958.362

> ml.gls<-gls(mlco2~Time+factor(Seas), cor=corAR1(0.985))
> AIC(ml.gls)
[1] 391.0092
```

Seasonal harmonic model

- Rational is that seasonal effects vary smoothly over the seasons
- Instead of using an indicator of season (1=January, 0=Not), can use a smooth function
  - Sine and/or cosine wave that oscillates at a certain number of cycles per season

$$X_t = T_t + \sum_{i=1}^{s/2} \{ s_i \sin(2\pi i t / s) + c_i \cos(2\pi i t / s) \} + z_t$$



Example

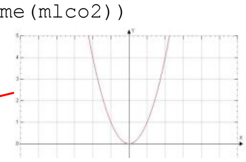
- First, we need to simulate a set of empty sine and cosine waves at different frequencies (1-6)

```
> SIN<-COS<-matrix(nr=length(mlco2), nc=6)
> for (i in 1:6) {
  COS[,i] <-cos(2*pi*i*time(mlco2))
  SIN[,i] <-sin(2*pi*i*time(mlco2))   }
```

- We don't know how many harmonics we need to include
- Fortunately, harmonic coefficients are known to be independent
  - Can add them all in, check for statistical significance and only keep the ones that are

Example

```
> TIME <-(time(mlco2) - mean(time(mlco2)))/sd(time(mlco2))
> mean(time(mlco2))
> sd(time(mlco2))
```



```
> ml.shgls<-gls(mlco2~ TIME + I(TIME^2) +
               COS[,1]+SIN[,1]+COS[,2]+SIN[,2]+
               COS[,3]+SIN[,3]+COS[,4]+SIN[,4]+
               COS[,5]+SIN[,5]+COS[,6]+SIN[,6], corr=corAR1(0.908))

 (Intercept)         TIME     I(TIME^2)      COS[, 1]
1600.58752636  155.24459467   18.22074618   -9.27557755
     SIN[, 1]      COS[, 2]      SIN[, 2]      COS[, 3]
  80.77815539   19.75028133  -37.63167836   -2.19576028
     SIN[, 3]      COS[, 4]      SIN[, 4]      COS[, 5]
  -6.27532860    2.24416895    5.78236098    0.97726215
     SIN[, 5]      COS[, 6]      SIN[, 6]
   2.60128993    0.02081744    0.64617844
```

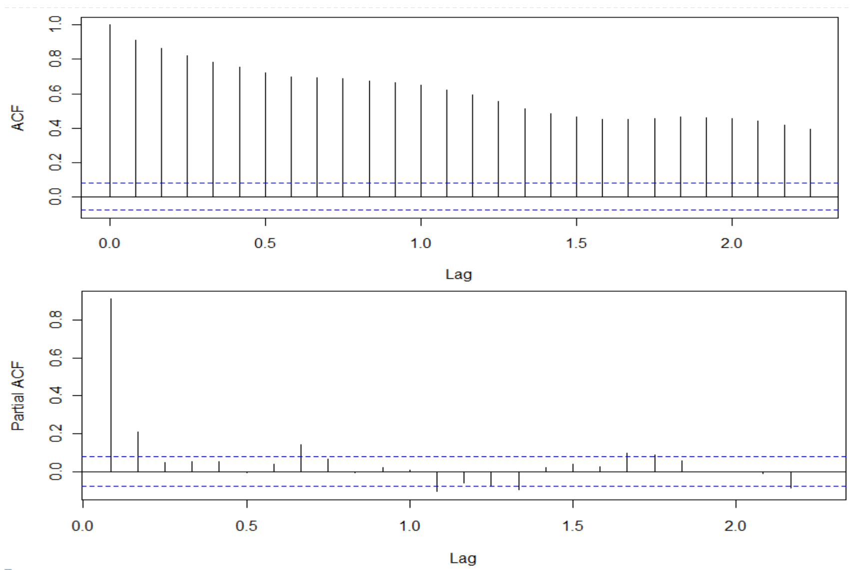
Results

```
> ml.shgls2<-gls(mlco2~ TIME + I(TIME^2) +
                COS[,1]+SIN[,1]+COS[,2]+SIN[,2]+
                COS[,3]+SIN[,3]+COS[,4]+SIN[,4]+
                SIN[,5], corr=corAR1(0.908))
> coef(ml.shgls2)/sqrt(diag(vcov(ml.shgls2)))
(Intercept)        TIME   I(TIME^2)    COS[, 1]    SIN[, 1]
1599.340883  155.135181   18.209209   -9.296485   80.892083
   COS[, 2]    SIN[, 2]    COS[, 3]    SIN[, 3]    COS[, 4]
  19.878676  -37.752959   -2.203375   -6.278049    2.251647
   SIN[, 4]    SIN[, 5]
   5.792330    2.595673

> AIC(ml.shgls2)
[1] 367.8002
```

$$x_t = 1599 + 155.1t + 18.2t^2 - 9.3\cos(2\pi t/12) + 80.9\sin(2\pi t/12) + 19.9\cos(4\pi t/12) - 37.8\sin(4\pi t/12) - 2.2\cos(6\pi t/12) + \ldots + 2.6\sin(10\pi t/12) + z_t$$

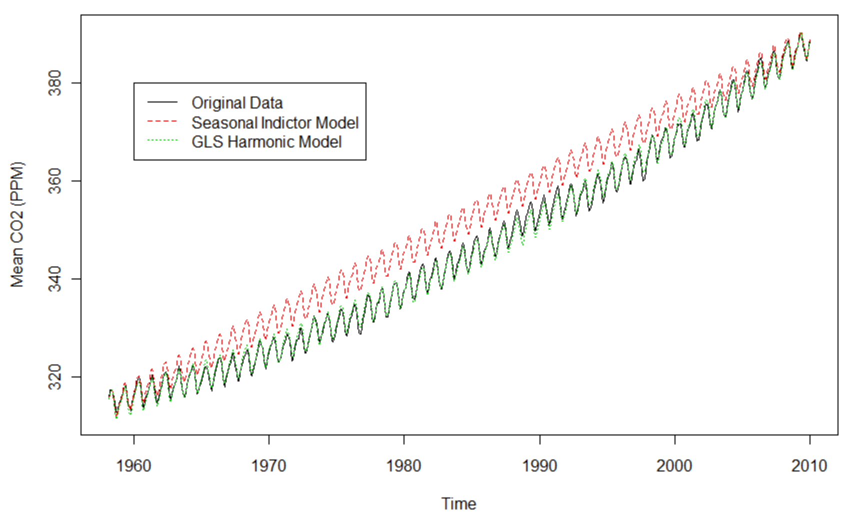
Plot residuals and acf/pacf for ml.shgls2 residuals

Results



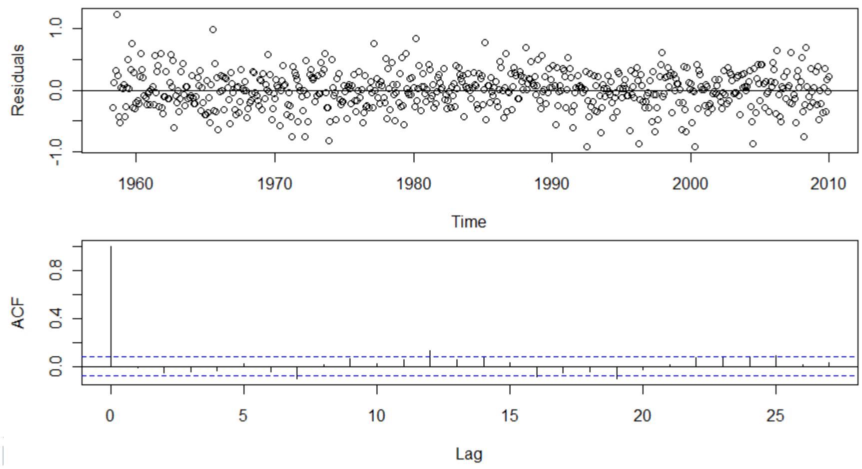
Results

Results



Now the autoregressive model

```
> ml.ar<-ar(resid(ml.shgls2), order=2)
> plot(ml.ar$resid), ylab="Residuals", type="p"); abline(h=0)
> acf(ml.ar$resid[-(1:2)], main="")
```



What's the equation?

mean of t

SD of t

$$x_t = 1599 + \frac{155.1(t-1984)}{15} + \frac{18.2(t-1984)^2}{15} - 9.3\cos(2\pi t/12) + 80.9\sin(2\pi t/12) + 19.9\cos(4\pi t/12) - 37.8\sin(4\pi t/12) - 2.2\cos(6\pi t/12) - 6.3\sin(6\pi t/12) + 2.3\cos(8\pi t/12) + 5.8\sin(8\pi t/12) + 2.6\sin(10\pi t/12) + \boxed{z_t}$$

where the residual series follows an AR(2) process:

$$z_t = .72e_{t-1} + .21e_{t-2} + e_t$$

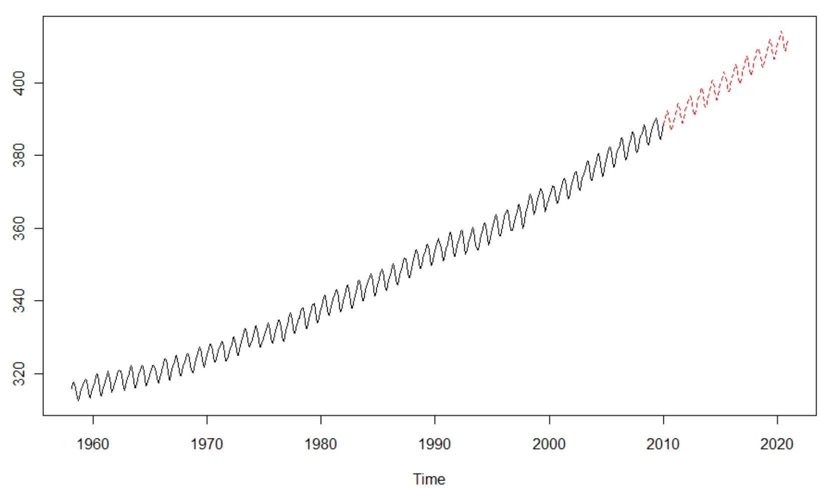
Making predictions

- The generic way of making predictions is `predict()`
  - But, must create a new data frame with values used to predict properly labeled

```
> new.T<-time(ts(start=2010, end=c(2020, 12), fr=12))
> TIME<-(new.T-mean(time(mlco2)))/sd(time(mlco2))
> SIN<-COS<-matrix(nr=length(new.T), nc=6)
> for (i in 1:6) {
  COS[,i] <-cos(2*pi*i*time(new.T))
  SIN[,i] <-sin(2*pi*i*time(new.T))   }
> SIN<-SIN[,-6]
> COS<-COS[,-(5:6)]
> new.dat<-data.frame(TIME=as.vector(TIME), SIN=SIN, COS=COS)
> ml.pred.ts<-ts(predict(ml.shgls2, new.dat), st=2010, fr=12)
```

Results

```
> plot(mlco2, ml.pred.ts, lty=1:2, col=c(1,2))
```



What have we learned?

- The importance of stationarity
- The difference between stochastic and deterministic trends in the data
- Methods for making stochastic TS stationary
  - Differencing
  - Smoothing
- Regression techniques for modeling deterministic TS
  - Removes both trend and seasonality

What's next?

- Examining and identifying stochastic processes that give rise to an observed time series
  - Autoregressive processes (AR)
    - How strongly the past influences the present
  - Moving average processes (MA)
    - Model present using past errors of prediction
  - Autoregressive moving average processes (ARMA)