Stat 436/536: Time Series Analysis: Chapters 1 and 2 Notes

Time Series:

- Single variable measured sequentially over time at a fixed sampling interval.
    - Treated as a realization of a sequence of random variables
    - Inferences to the true process that generated the observed realization of random variables

- Based on the specific sampling interval, time series can contain seasonal cycles
    - Cyclic patterns on a fixed and known period that correspond to sampling interval
        - Monthly average temperatures have a seasonal pattern

- Or periodic cycles
    - Observed cyclic patterns that have fixed periodicity but may not correspond to the sampling interval
        - El-Nino/La-Nina impacts on ocean temperatures: period of approximately 10 years

    - Quasi-periodic: the periodicity varies over time

Time Series research questions:

1. Description/ Understanding

2. Accommodation

3. Monitoring/ Control

4. Forecasting

First, some R basics…

- Downloading R: http://cran.rstudio.com/
- Use R-studio to manage your R work: http://www.rstudio.com/
    - Use R-markdown if at all possible for HW and Exams
- Update your installation of R and R-studio if you have not done so in the last 6 months!

  Note for dealing with nominal (categorical) data:
    - If the variable is coded non-numerically, it is automatically read as a factor variable
        - Baseline category is first alphabetically

    - If the variable is numeric, then use the `factor()` function to convert it to a categorical variable, smallest number is baseline category
    - The `relevel()` function allows you to alter the baseline category for the variable or use `factor(,levels=c("firstlevel",…,"lastlevel"))`

Installing and loading packages

Reading data into R:

- Directly from dropbox
- Questions?

The ts data class:

```
>?ts
```

- `data=`
- `start=, end=`
- `frequency` or `deltat`
    - For cyclical time series, the frequency of measurement (sampling interval) is important

    - Period: length of time (or number of observations) to complete a cycle

    - Frequency: 1/period = number of cycles completed per unit time (or per observation)

    - Monthly time series, yearly cycle

- Time series measured every 30 minutes – diurnal (daily) cycle:

    - and possibly a yearly cycle:

- Daily data with yearly cycle:

- `time` function can be applied to ts object to extract fractional date
- `cycle` function can be applied to ts object to extract numerical seasonal coding
- `season` function can be applied to ts object to extract categorical date information

Data repositories:

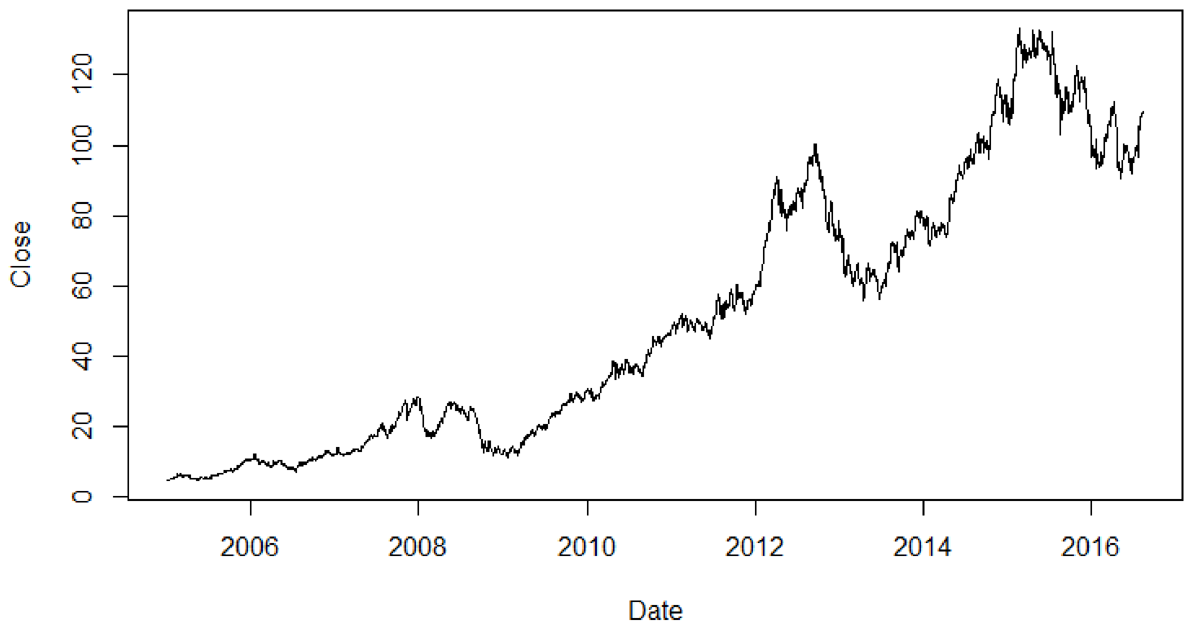
- The last few years have seen massive growth in open-access data repositories and tools for scraping data from various web locations

- Rob Hyndman's Time Series Data Library: https://datamarket.com/data/list/?q=provider:tsdl
    - R package `rdatamarket`

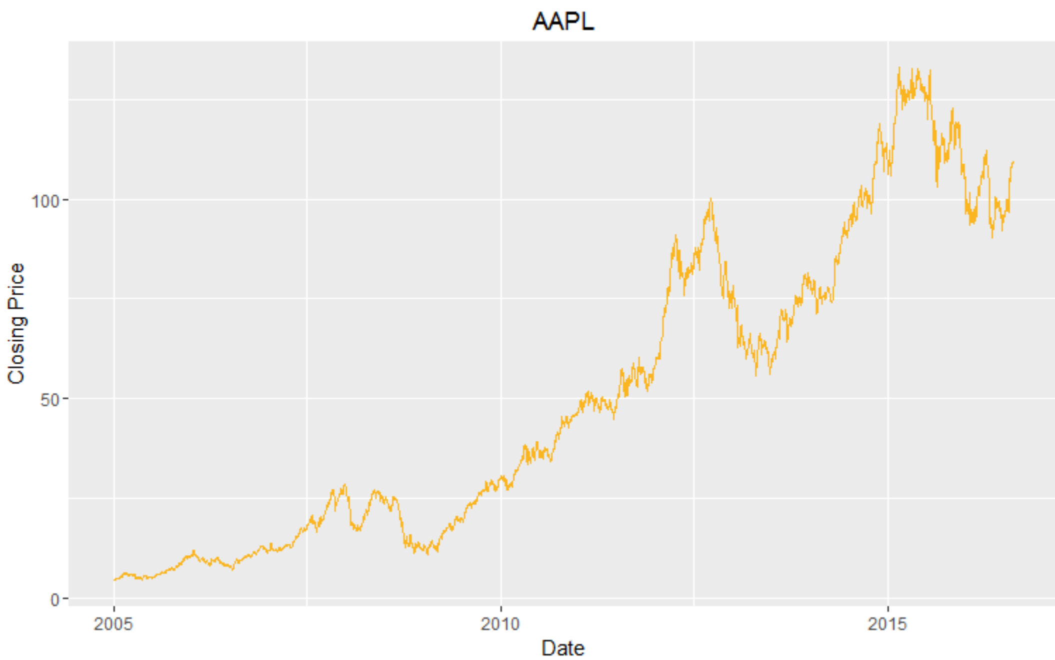
- Quandl with the R package `Quandl`

```
> require(Quandl)
> data_series <- Quandl("GOOG/NASDAQ_AAPL", start_date="2005-01-01")[,c(1,5)]
> head(data_series)
        Date  Close
1 2016-08-19 109.36
2 2016-08-18 109.08
3 2016-08-17 109.22
4 2016-08-16 109.38
5 2016-08-15 109.48
6 2016-08-12 108.18
> tail(data_series)
           Date Close
2939 2005-01-10  4.93
2940 2005-01-07  4.95
2941 2005-01-06  4.61
2942 2005-01-05  4.61
2943 2005-01-04  4.57
2944 2005-01-03  4.52
> require(car)
> some(data_series)
           Date Close
511  2014-08-12 95.97
972  2012-10-10 91.56
1431 2010-12-15 45.77
1525 2010-08-04 37.57
1632 2010-03-08 31.30
2407 2007-02-22 12.79
2420 2007-02-02 12.11
2707 2005-12-09 10.62
2777 2005-08-31  6.70
```

```
2811 2005-07-14  5.82
> plot(data_series,type="l")
```



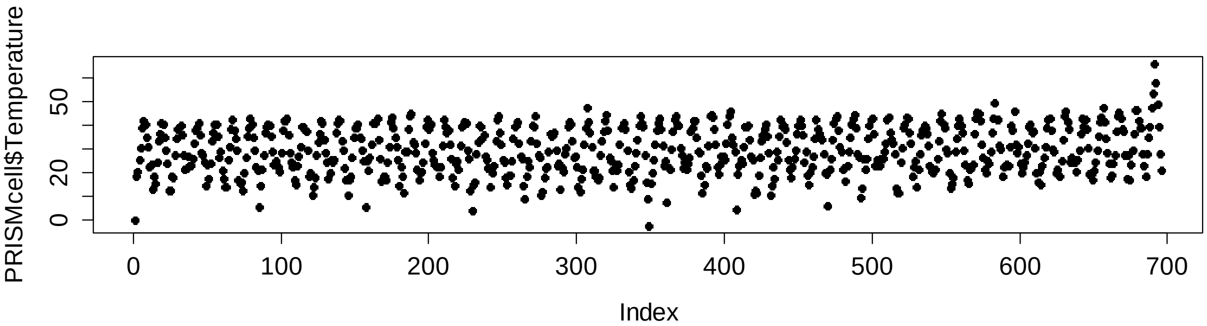

```
> require(ggplot2)
> my.plot <- ggplot(data=data_series, aes(x=Date, y=Close)) +
+  geom_line(color="#FAB521") + xlab("Date") + ylab("Closing Price") + ggtitl
e("AAPL") # Adding titles
> my.plot #
```

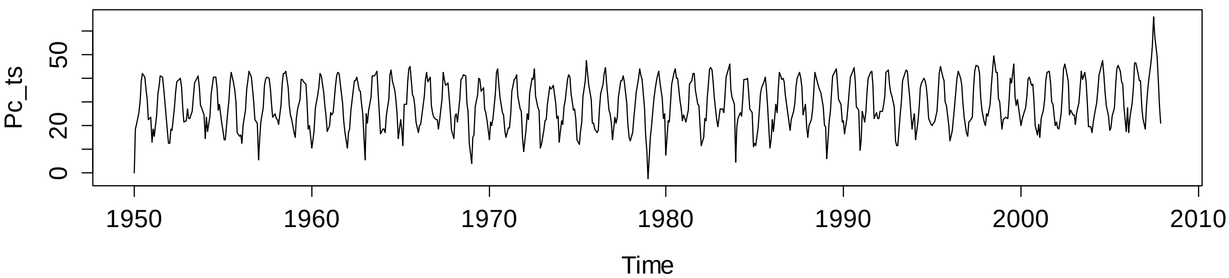


Monthly average temperature in a PRISM (Parameter-elevation Regressions on Independent Slopes Model, http://www.prism.oregonstate.edu/) cell from 1950 to 2007:

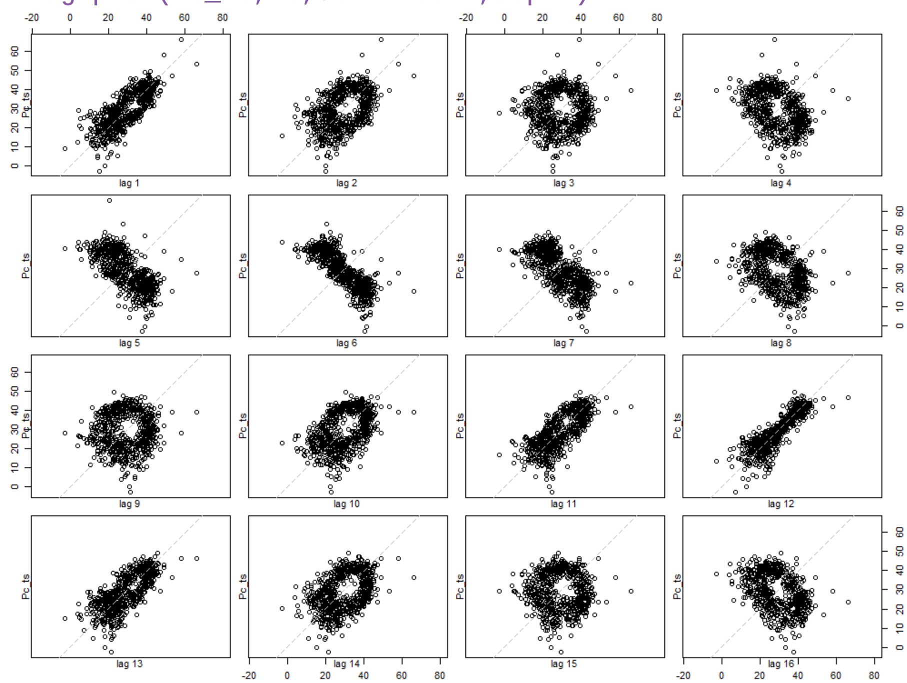
```
> PRISMcell<-read.csv("https://dl.dropboxusercontent.com/u/77307195/PRISMcell
.csv",header=T)
> par(mfrow=c(2,1))
> plot(PRISMcell$Temperature,pch=16)
> Pc_ts<-ts(PRISMcell$Temperature,start=c(1950,1),frequency=12)
```

```
> Pc_ts
        Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct    Nov    Dec
1950 -0.166 18.104 20.228 25.196 30.362 38.984 41.666 40.496 34.628 30.992 22.496 23.342
1951 12.704 18.104 15.206 24.278 33.440 36.284 40.712 40.244 34.610 29.174 23.864 12.380
1952 12.038 18.338 18.032 27.140 34.376 38.390 38.804 39.956 35.798 27.392 21.200 21.722
1953 26.690 22.748 22.712 25.646 32.288 38.048 38.966 40.676 35.888 28.526 27.266 24.188
1954 14.072 23.108 17.186 24.044 33.188 36.806 40.424 40.100 35.510 26.186 28.598 20.732

…

2003 24.206 20.192 25.268 29.552 36.176 42.908 39.434 42.152 38.462 30.128 19.328 19.364
2004 16.718 20.732 24.098 29.120 35.114 41.360 43.538 47.138 41.126 34.430 28.220 23.936
2005 17.600 18.734 23.918 27.230 37.274 43.718 45.320 42.656 38.282 37.328 27.068 17.348
2006 27.266 16.826 23.504 29.300 36.464 46.166 46.382 41.612 38.930 38.957 27.995 22.730
2007 18.347 27.923 35.051 39.281 47.219 53.258 66.047 58.100 49.073 39.218 27.617 20.777
> plot(Pc_ts)
```






```
> lag.plot(Pc_ts,16,do.lines=F,asp=1)
```

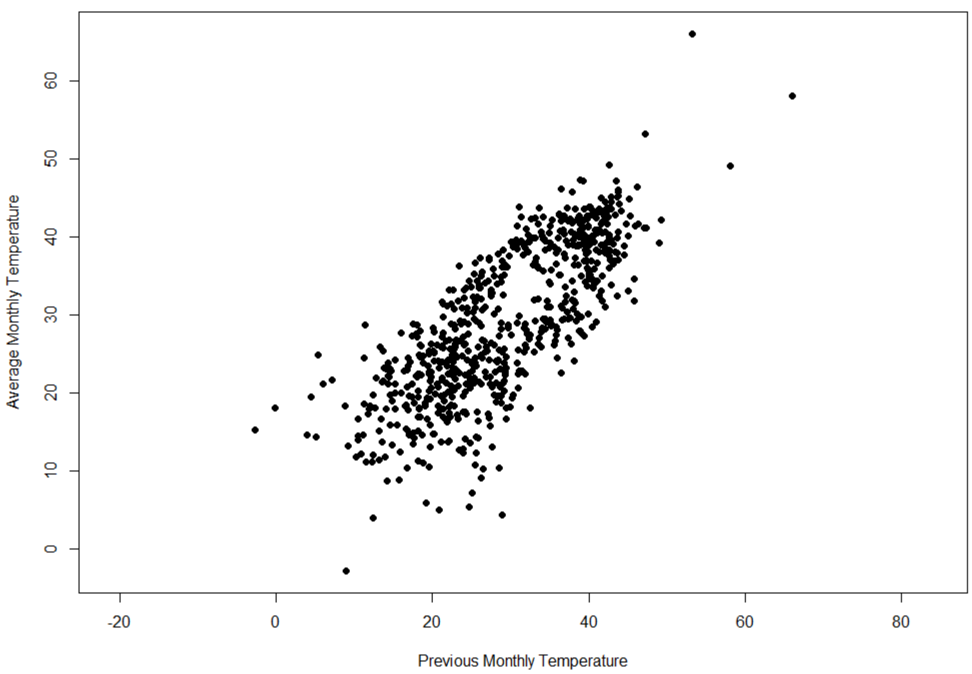


- Lagging involves backshifting a version of time series relative to other series using the backshift operator, B,
    - which is available using the `zlag` function from the `TSA` package

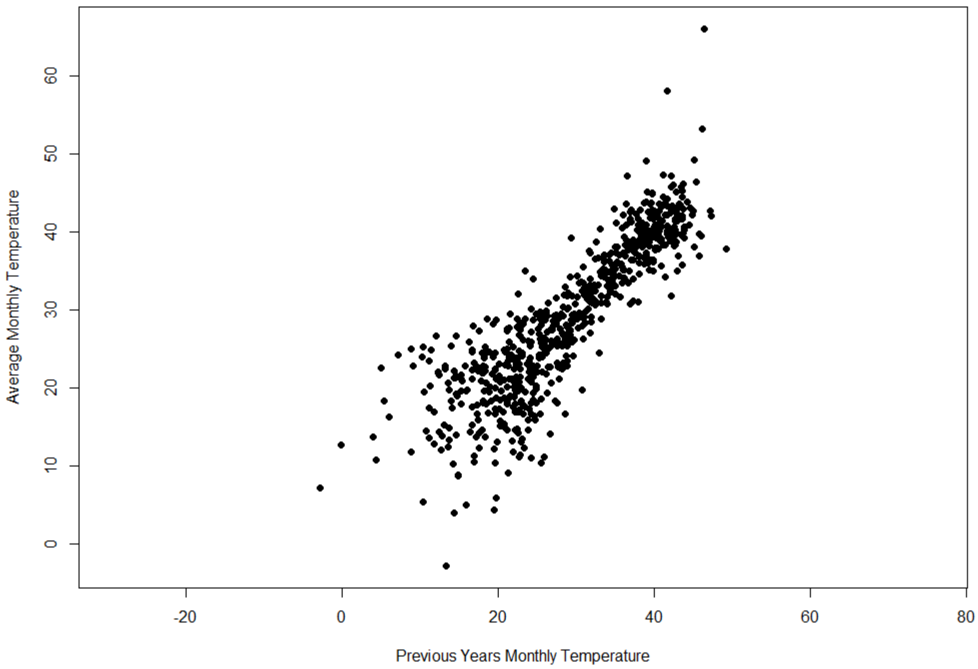
```
> require(TSA)
> data.frame(Pc_ts,zlag(Pc_ts))
     Pc_ts zlag.Pc_ts.
1   -0.166          NA
2   18.104      -0.166
3   20.228      18.104
4   25.196      20.228
5   30.362      25.196
6   38.984      30.362
```

- This leads to how we can write out (lag 1) differencing a time series:

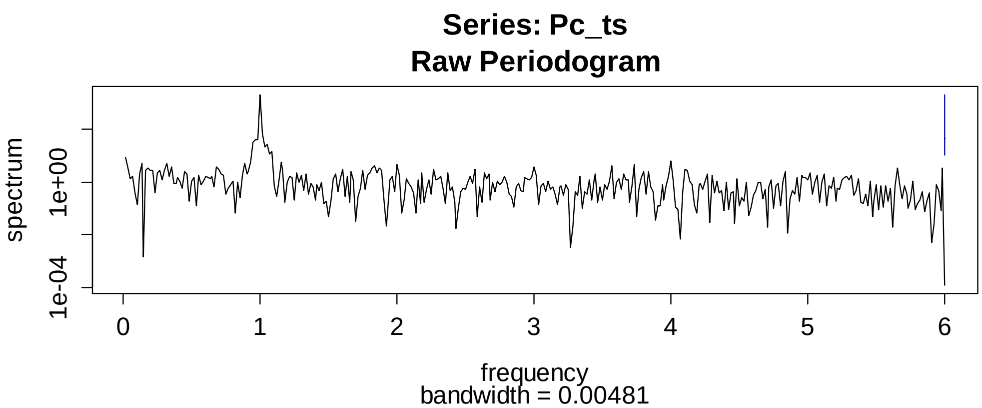
```
> plot(Temperature~zlag(Temperature),data=PRISMcell,pch=16,ylab='Average Mont
hly Temperature',xlab='Previous Monthly Temperature',asp=1)
```

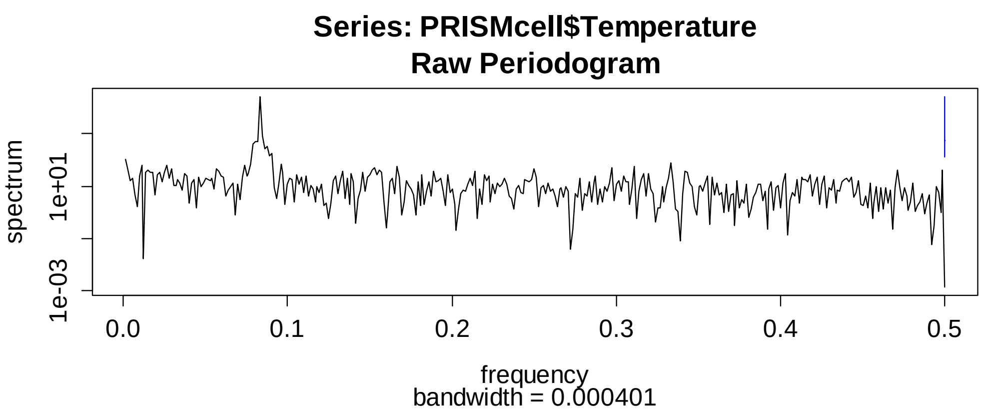



```
> plot(Temperature~zlag(Temperature,12),data=PRISMcell,pch=16,ylab='Average M
onthly Temperature',xlab='Previous Years Monthly Temperature',asp=1)
```



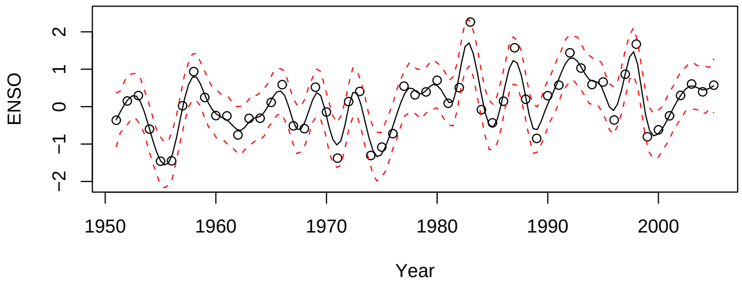

```
> par(mfrow=c(2,1))
> spec.pgram(Pc_ts)
> spec.pgram(PRISMcell$Temperature)
> 1/12
[1] 0.08333333
```

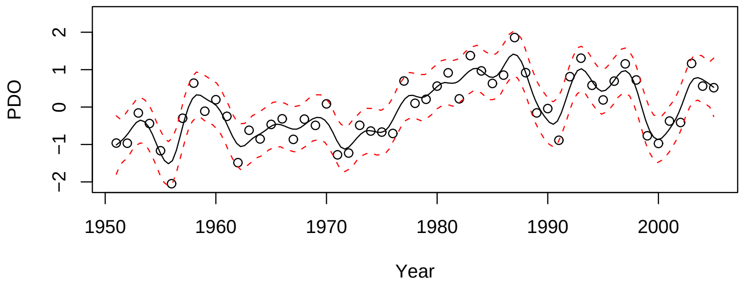





```
> 1/120
[1] 0.008333333
```

Two different climatic yearly averaged indices: El Nino/ Southern Oscillation (ENSO) and Pacific Decadal Oscillation (PDO) with penalized regression spline time trend estimates and 95% CIs



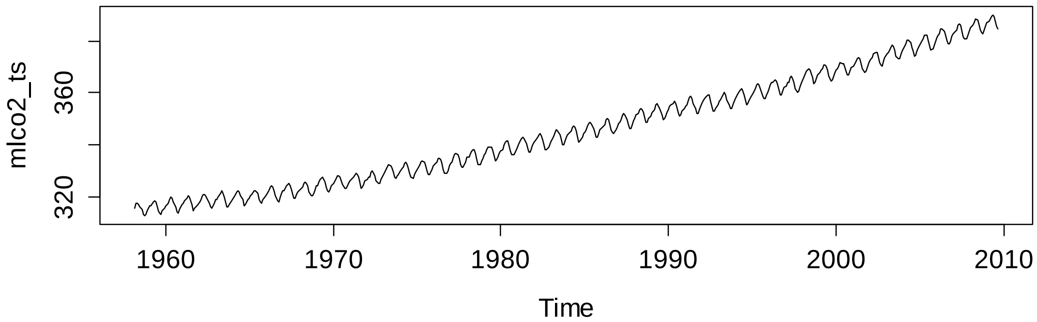


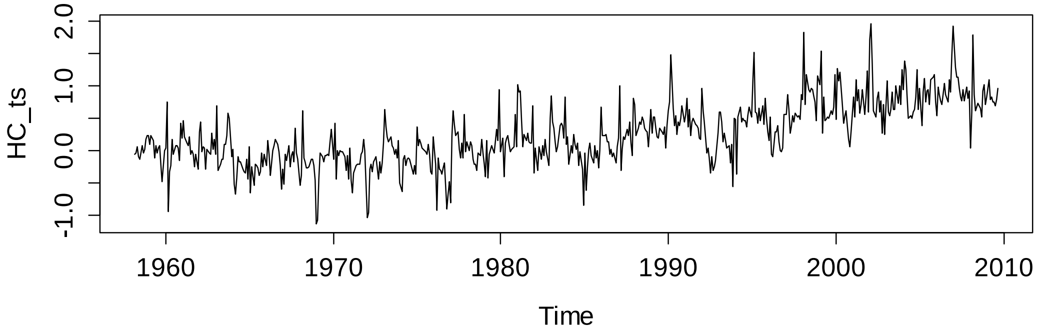
- Observations of monthly CO_2 atmospheric concentration averages from the Mauna Loa Observatory, Mauna Loa, Hawaii, USA. Obtained from the ESRL Global Monitoring Division of the National Oceanic and Atmospheric Administration at http://www.esrl.noaa.gov/gmd/dv/data/index.php?parameter_name=Carbon%2BDioxide
    - Dataset downloaded Oct 1, 2011.
    - CO_2 measured in parts per million

```
> require(multitaper)
> data(mlco2)
> head(mlco2)
  Year M    CO2
1 1958 3 315.71
2 1958 4 317.45
3 1958 5 317.50
4 1958 6 317.11
5 1958 7 315.86
6 1958 8 314.93
> mlco2_ts<-ts(mlco2$CO2,start=c(1958,3),freq=12)
> mlco2_ts
        Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct    Nov    Dec
1958               315.71 317.45 317.50 317.11 315.86 314.93 313.20 312.61 313.33 314.67
1959 315.62 316.38 316.71 317.72 318.29 318.16 316.55 314.80 313.84 313.26 314.80 315.59
1960 316.43 316.97 317.58 319.02 320.02 319.59 318.18 315.91 314.16 313.83 315.00 316.19
```

- Hadley Climate Research Unit Temperature anomaly (Northern Hemisphere) time series. Consists of monthly observations, truncated to start at March 1958 and extend to September 2009, to match mlco2 dataset. This dataset was retrieved from the Hadley CRU on Oct 1, 2011.

```
> data(HadCRUTnh)
> HC_ts<-ts(HadCRUTnh$Temp,start=c(1958,3),freq=12)
> par(mfrow=c(2,1))
> plot(mlco2_ts)
> plot(HC_ts)
```

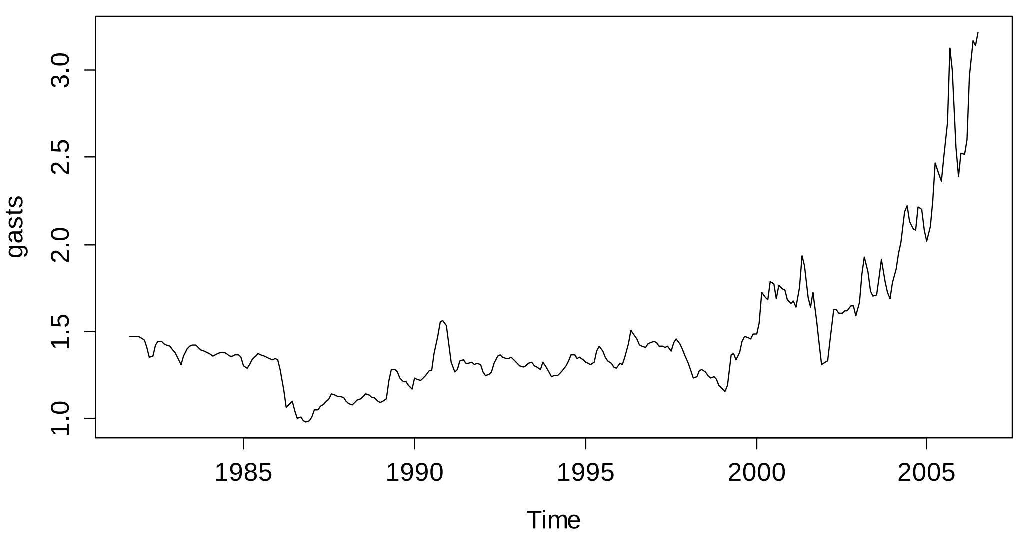




Gas Prices from 1981 to 2006

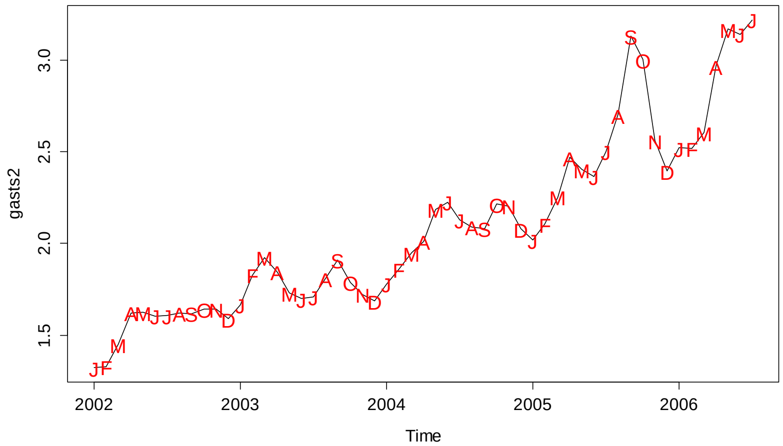
- Average monthly price for the entire US from the US Bureau of Labor Statistics (by month):

```
> gas<-read.table("D:/usa.txt")
> colnames(gas)<-c("Date","Price")
> gasts<-ts(gas$Price,frequency=12,start=c(1981,9),end=c(2006,7))
> ts.plot(gasts)
```

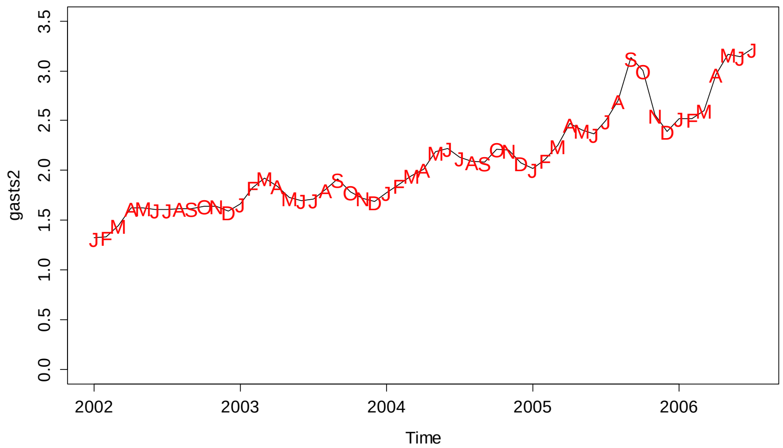


Subset of gas price time series with seasonal labels:

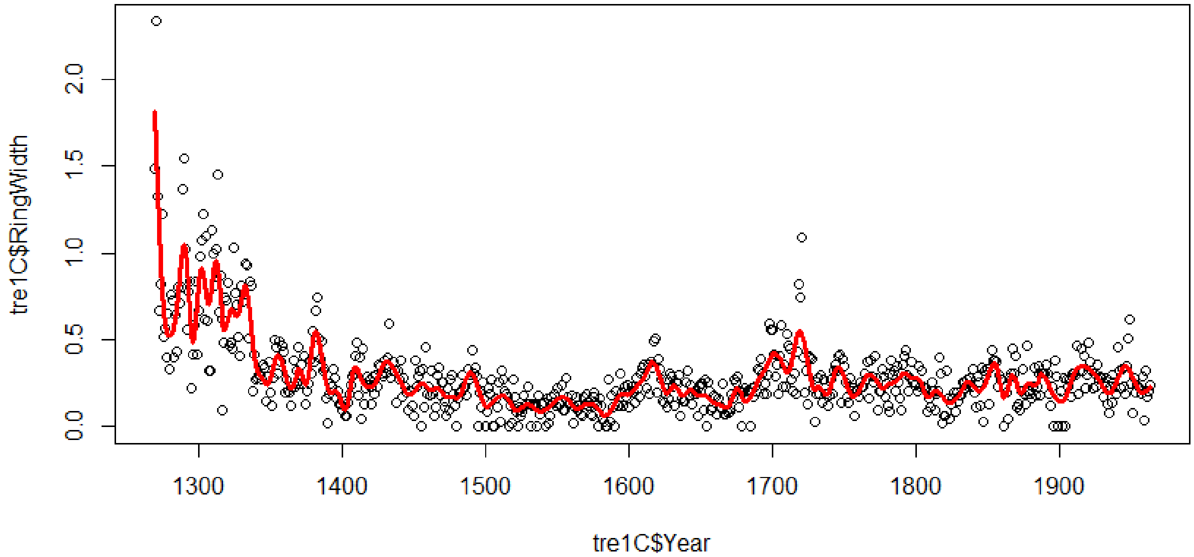
```
> require(TSA)
> gasts2<-ts(gas$Price[245:299],frequency=12,start=c(2002,1),end=c(2006,7))
> plot(gasts2)
> points(y=gasts2,x=time(gasts2),pch=as.vector(season(gasts2)),cex=1.2,col="red")
```



```
> plot(gasts2,ylim=c(0,3.5))
> points(y=gasts2,x=time(gasts2),pch=as.vector(season(gasts2)),cex=1.2,col="red")
```



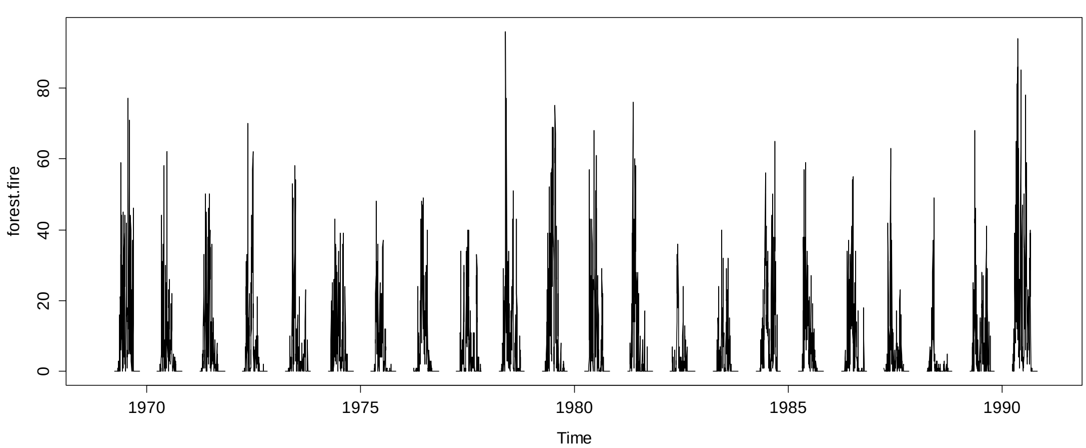
```
> require(detrendeR)
> data(co021, package="dplR")
>
> tre1<-data.frame(Year=as.numeric(row.names(co021)),co021[,1])
> head(tre1)
  Year co021...1.
1 1176         NA
2 1177         NA
3 1178         NA
4 1179         NA
5 1180         NA
6 1181         NA
> tre1C<-na.omit(tre1)
> names(tre1C)[2]<-"RingWidth"
> head(tre1C)
    Year RingWidth
95  1270      1.48
96  1271      2.33
97  1272      1.32
98  1273      0.67
99  1274      0.82
100 1275      1.22
> m1<-smooth.spline(tre1C$RingWidth~tre1C$Year)
> plot(tre1C$RingWidth~tre1C$Year)
> lines(fitted(m1)~tre1C$Year,col="red",lwd=3)
```



Daily fire seats(?) in Irkutsk Region, USSR April 1 1969 to October 31 1991 (4708 observations)

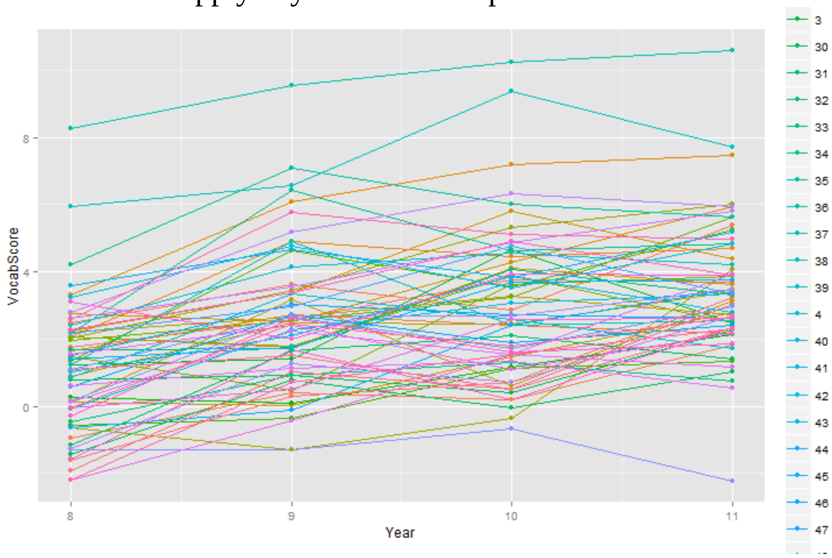
- Available in `mar1s` package forest.fire data set:

```
> require(mar1s)
> data(forest.fire)
> is.ts(forest.fire)
[1] TRUE
> table(forest.fire)
forest.fire
   0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18 
2271  322  234  177  131  120  104   81   73   73   63   57   65   52   55   50   38   30   41 
  19   20   21   22   23   24   25   26   27   28   29   30   31   32   33   34   35   36   37 
  39   31   34   37   33   32   24   29   23   21   23   24   19   21   19   23   14   17   11 
  38   39   40   41   42   43   44   45   46   47   48   49   50   51   52   53   54   55   56 
  12   17   15    4   11   16   10    4    6    7    3    6    9    4    5    6    3    3    5 
  57   58   59   60   61   62   63   64   65   66   67   68   69   70   71   75   76   77   78 
   6    6    4    3    3    4    3    1    2    1    1    2    2    1    1    1    1    2    2 
  81   85   86   94   96 
   1    1    1    1    1 
> summary(forest.fire)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
  0.000   0.000   1.000   7.098   9.000  96.000    3322 
> plot(forest.fire~time(forest.fire),type="l")
```



What Time Series Analysis is not:

- Analysis of repeated measures/ longitudinal data (a few measurements on multiple sites or subjects)
    - Our discussion of generalized least squares and correlation structures does pertain to mixed models that are useful for longitudinal data but we won't explicitly apply any methods to repeated measures data



What we won't get to spend too much time in this course on:

- Multivariate time series (long series of observations taken on a small number of different variables or subjects or sites)
    - State space models provide a way of generalizing some of these models

- Frequency domain models:
    - Switching from focusing on the mean of the process over time to focusing on power of oscillations at different frequencies

- Wavelet techniques:
    - Alternative to conventional frequency domain methods that are better suited to non-smooth series

- Nonlinear, nonnormal time series
    - There are models that incorporate other types of dependency than what we start with learning

- Forecasting
    - While many of the models we encounter can be used to forecast into the future, we will focus on fitting models within the scope of the time series first.

But the methods and tools we are going to learn will provide the fundamentals to better understand all of these techniques if you start reading more about them.

CC 1.2: Box-Jenkins (1976) Model Building Strategy:

1. Model Specification

2. Model fitting

3. Model diagnostics

4. Iterate from 1 to 3 until "convergence" on a model

Principle of Parsimony: "model used should require the smallest number of parameters that will adequately represent the time series"

CC Ch 2: (Start with Appendix A especially properties of Variance and Covariance if you haven't seen this before)

$x_1, x_2, x_3, \ldots$ or $\{x_t: t=0, \pm1, \pm2\}$ or $\{Y_t: t=0, \pm1, \pm2,\ldots\}$ are a time series sequence of random variables or a sequence of observations

$\{Y_t: t=0, \pm1, \pm2,\ldots\}$ is called a stochastic process and will be our route to specifying a time series model

Suppose Y_t ~iid Normal, what do we need to "know" to model Y_t?

What are some ways that we can relax that assumption?

CC 2.2: Means, Variances, Covariances for Stochastic Processes (in general)
$\{Y_t: t=0, \pm1, \pm2,\ldots\}$:

Mean Function: $\mu_t = E(Y_t) = \int_{-\infty}^{\infty} x f_t(x)dx$

Autocovariance function: $Cov(Y_t,Y_s)=\gamma_{t,s}=\gamma(t,s)=E[(Y_t-\mu_t)(Y_s-\mu_s)]$
for all possible s,t

=

- Smooth series have large autocovariances when t and s far apart
- Rough series have near zero autocovariances for t and s far apart
- $\gamma(s,t)=0$ implies no ***linear*** relationship
  - Does not imply no relationship
- Bivariate normality and $\gamma(s,t)=0$ does imply independence

$$\gamma(t,t)=E[(Y_t-\mu_t)^2]$$

- depends on size of variance so comparison is difficult

Autocorrelation function (ACF): Theoretical

- Re-scaled autocovariance to be between -1 and 1

$$\rho(t,s)=\frac{\gamma(t,s)}{\sqrt{\gamma(t,t)\gamma(s,s)}}=Corr(Y_t,Y_s)$$

- Unitless measure of the ability to linearly predict Y_t using Y_s

Properties of autocovariance and autocorrelation:

Covariances and Variances of linear combinations of random variables (some details omitted):