Time Series: A First Course with Bootstrap Starter

Contents

Lesson 1: Takes and tools

Takes

- smoothness: greater correlation easier foreacstin
- log as reducer of varibility : formal proof ?
- formalization of cycle: "High association can occur for non-adjacent random variables"
- rarely a good idea to regress on time only

Tools

- windowing, spanning and creating a movie

Links to JD+

algos, doc and related training

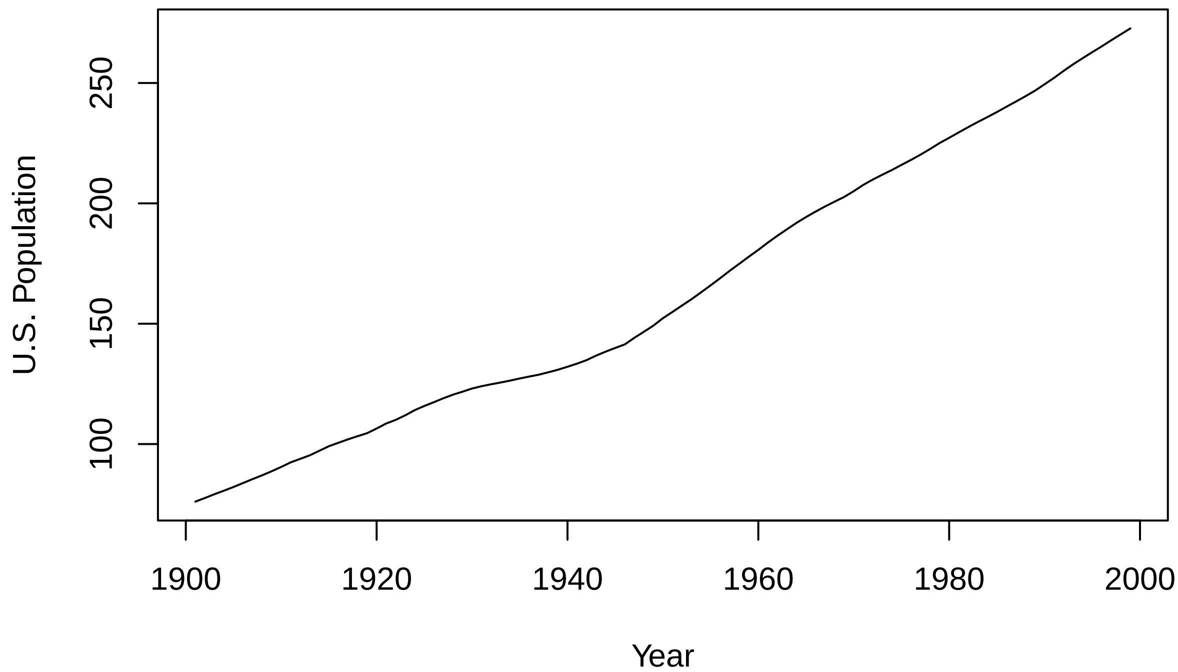
Lesson 1-1: Time Series Data

- A *time series* is a dataset where the observations are recorded at discrete time intervals.
- We denote the observation a time t by x_t, and the random variable by X_t.
- We have times $t = 1, 2, \ldots, n$ for the *sample* $X_1, X_2, \ldots, X_n$.
- Time series data might not be i.i.d. (Independent and Identically Distributed)!

Example 1.1.3. U.S. Population

- U.S. Population growth over the twentieth century.

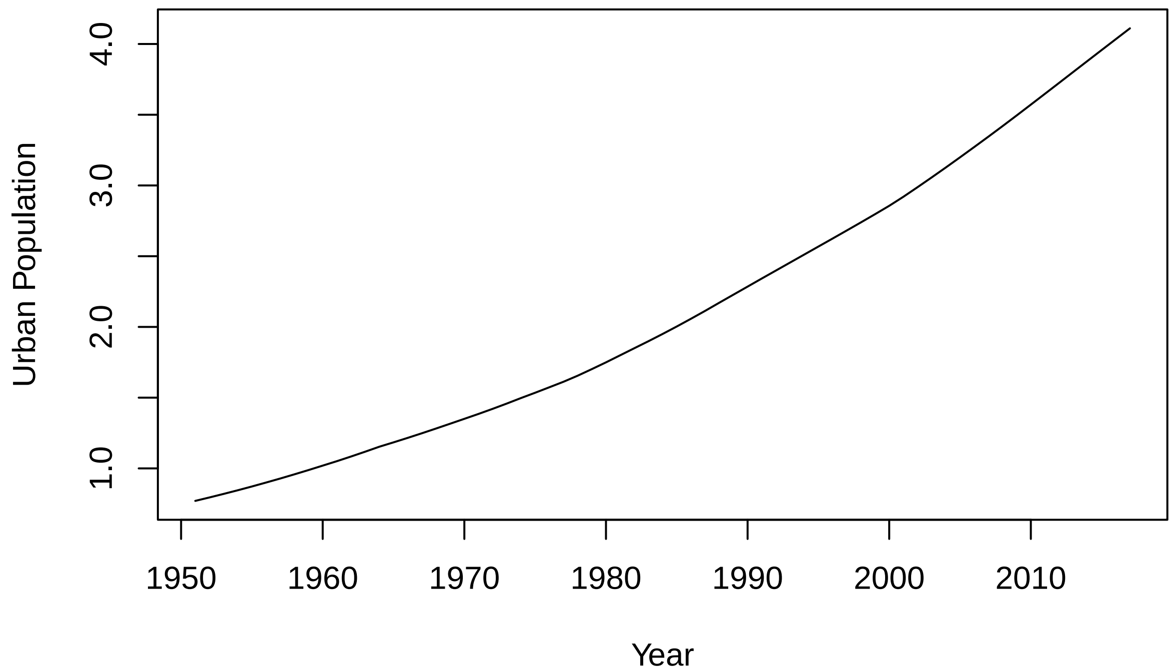
```
pop <- read.table("USpop.dat")
pop <- ts(pop, start = 1901)
plot(pop*10e-7,xlab="Year",ylab="U.S. Population",lwd=1)
```



Example 1.1.4. Urban World Population

- Urban World Population trends upwards more strongly after WWII.

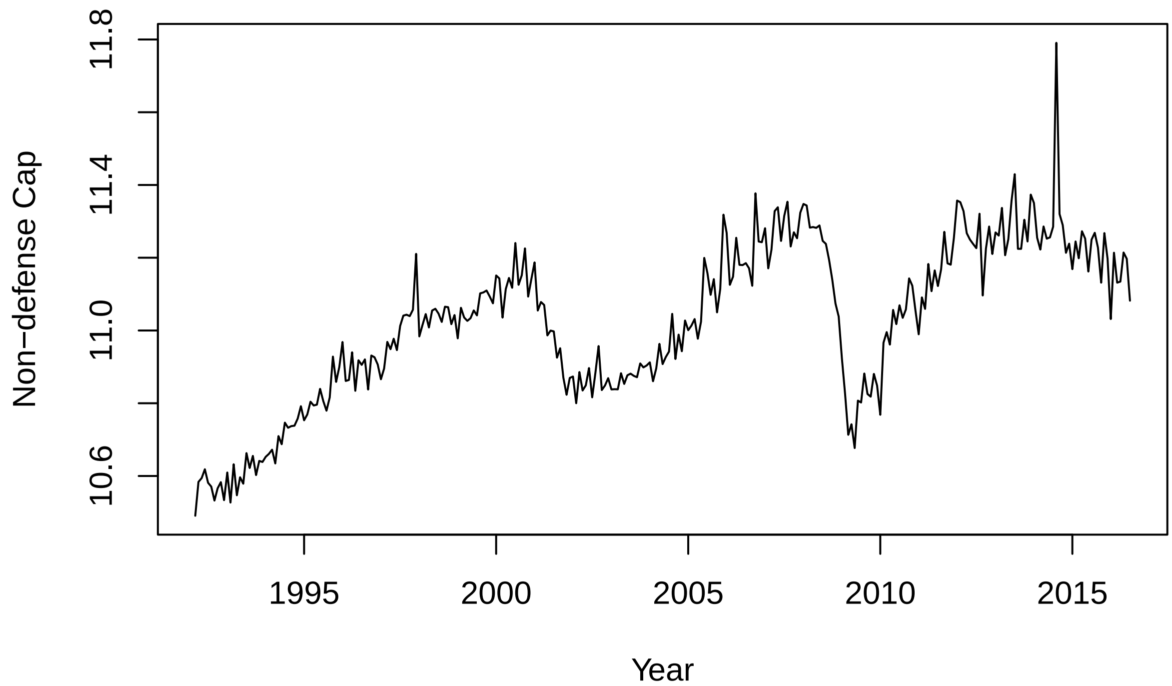
```
urban <- read.table("urbanpop.dat")
urban <- ts(urban[67:1,], start = 1951)
plot(urban*1e-9,xlab="Year",ylab="Urban Population",lwd=1)
```



Example 1.1.5. Non-Defense Capitalization

- Non-Defense Capitalization (New Orders) shows non-monotonic trend, and is more *noisy*.

```
ndc <- read.table("Nondefcap.dat")
ndc <- ts(ndc[,2],start=c(1992,3),frequency=12,names= "NewOrders")
plot(ndc,xlab="Year",ylab="Non-defense Cap")
```



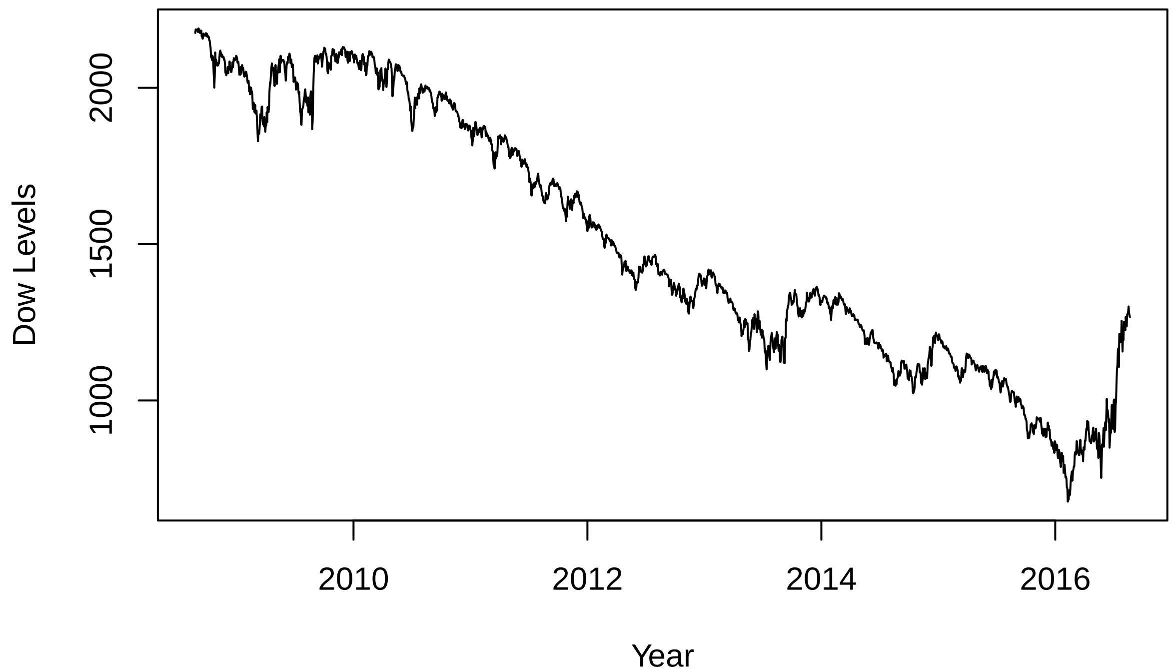
Serial Dependence and Forecasting

- Smoothness corresponds to high positive association (correlation) between adjacent variables.
- High association means forecasting (prediction of future values) is easier.

Example 1.1.8. Dow Jones Industrial Average

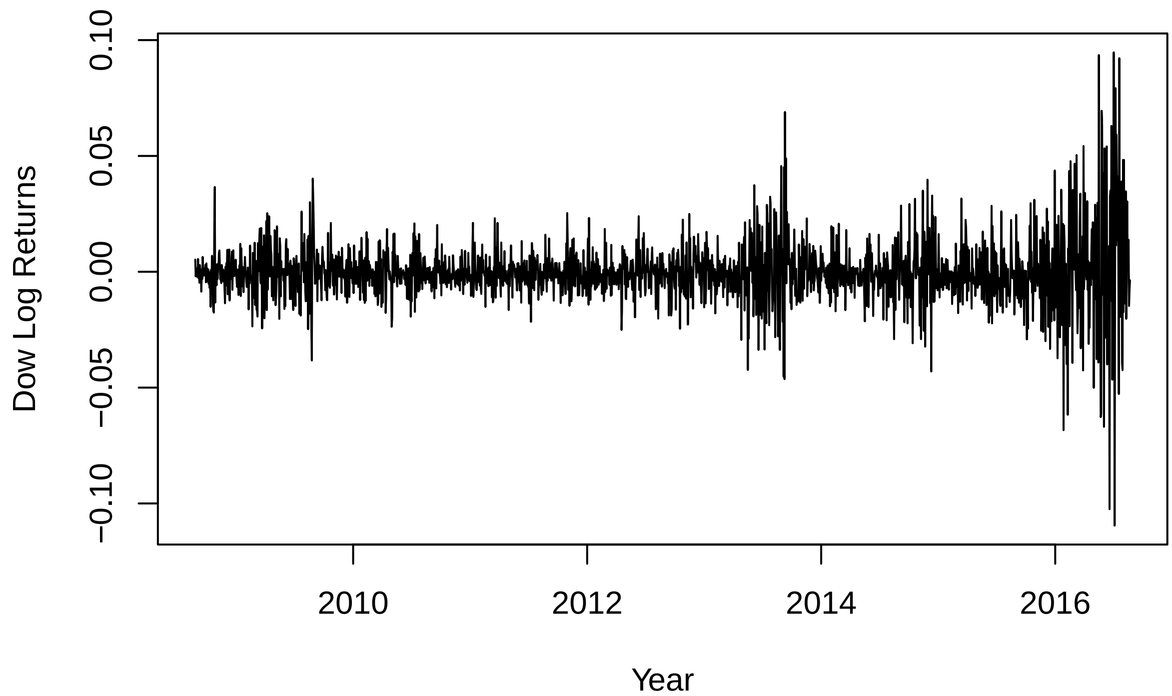
- Dow Jones Industrial Average has trend, but is hard to forecast one day ahead.

```
dow <- read.table("dow.dat")
dow <- ts(dow,start=c(2008,164),frequency=252)
plot(dow,xlab="Year",ylab="Dow Levels")
```



- We can plot the log returns (consecutive difference of logged data), which shows how volatility is not constant.

```
dow.diff <- diff(log(dow[,1]))
plot(dow.diff,xlab="Year",ylab="Dow Log Returns")
```



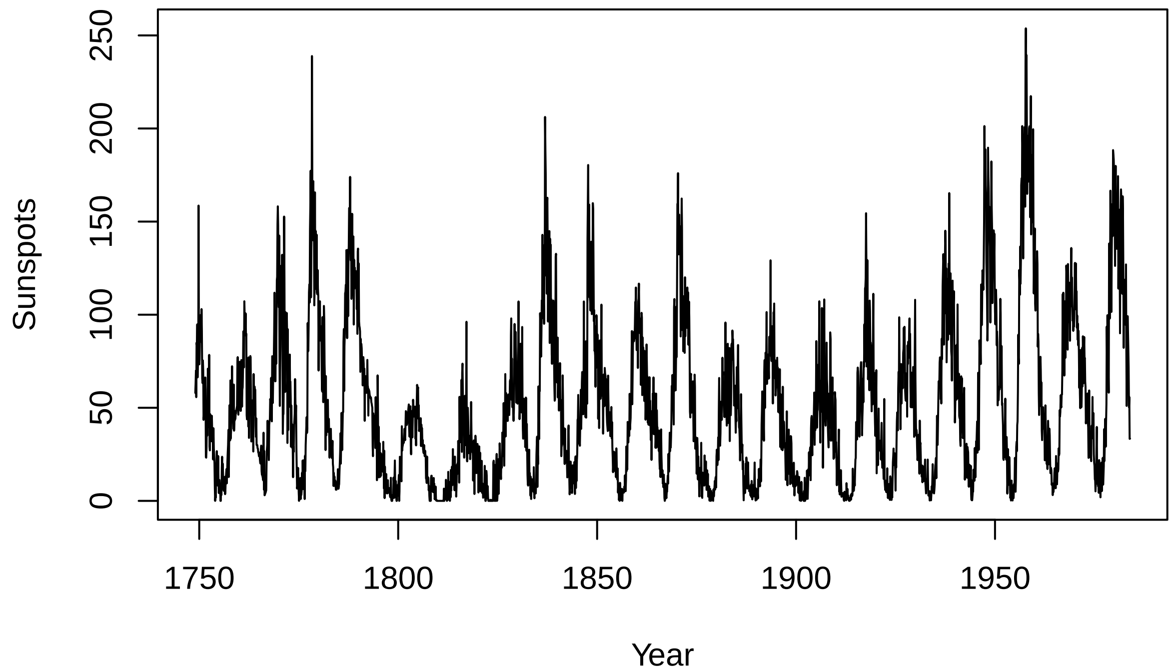
Lesson 1-2: Cycles

- High association can occur for non-adjacent random variables.
- For some fixed $h > 1$, we may have X_t and X_{t-h} associated for all t.
- This is a *periodic* effect, called a *cycle*, of period h.

Example 1.2.1. Sunspots

- Wolfer sunspots series measures number of sunspots recorded each month.
- Cycles are roughly 11 years (so $h \approx 132$).

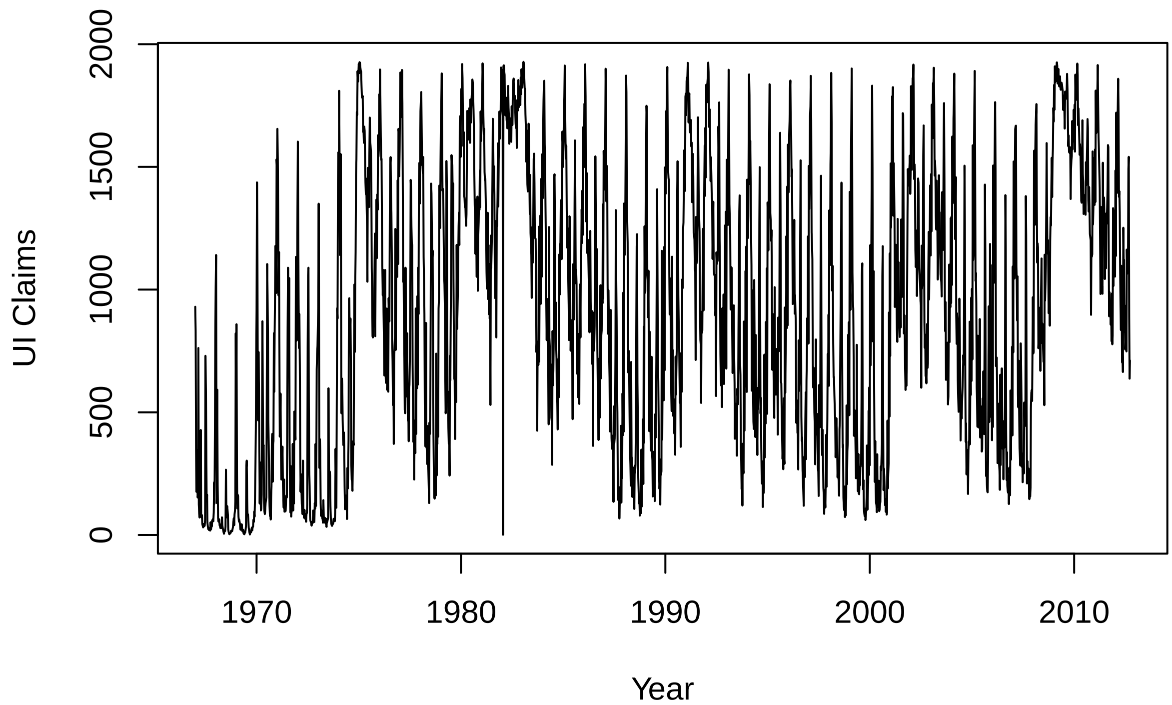
```
wolfer <- read.table("wolfer.dat")
wolfer <- ts(wolfer,start=1749,frequency=12)
plot(wolfer,xlab="Year",ylab="Sunspots")
```



Example 1.2.2. Unemployment Insurance Claims

- Weekly measurements of claims for unemployment insurance (pre-Covid).
- There is a weekly cylical pattern, corresponding to a cycle of annual period ($h = 52$).

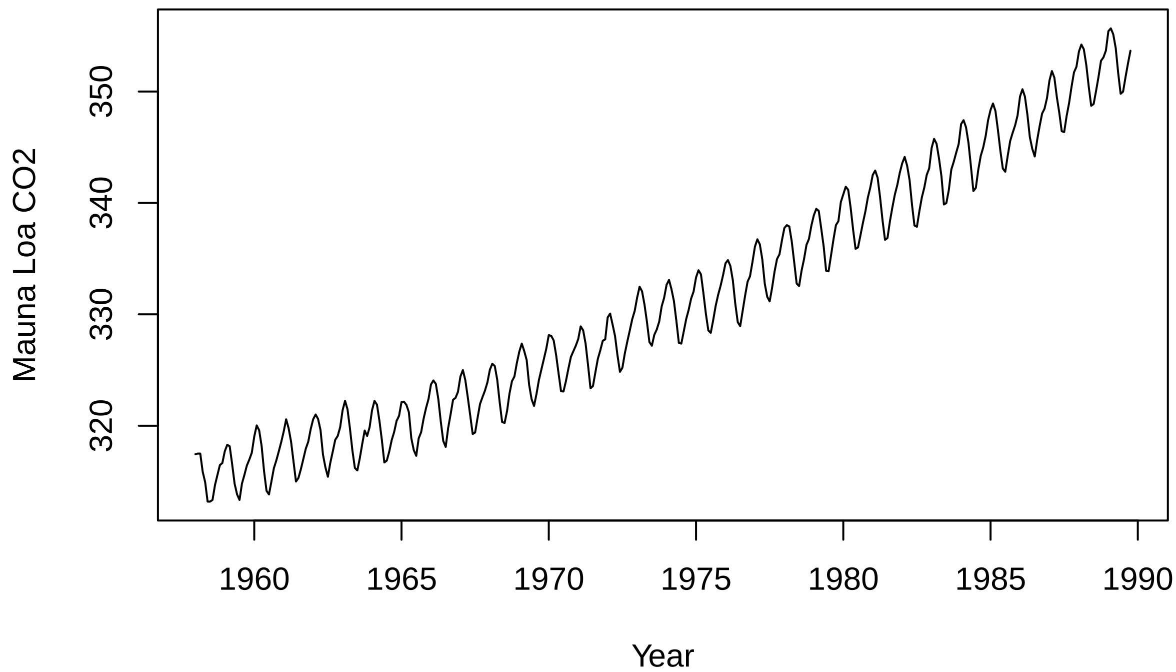
```
ui <- read.table("ui.dat")
ui <- ts(ui,start=1967,frequency=52)
plot(ui,ylab="UI Claims",xlab="Year")
```



Example 1.2.3. Mauna Loa carbon Dioxide

- Monthly measurements of CO2 levels on mount Mauna Loa.
- Apparent upward trend and monthly ($h = 12$) cycle.

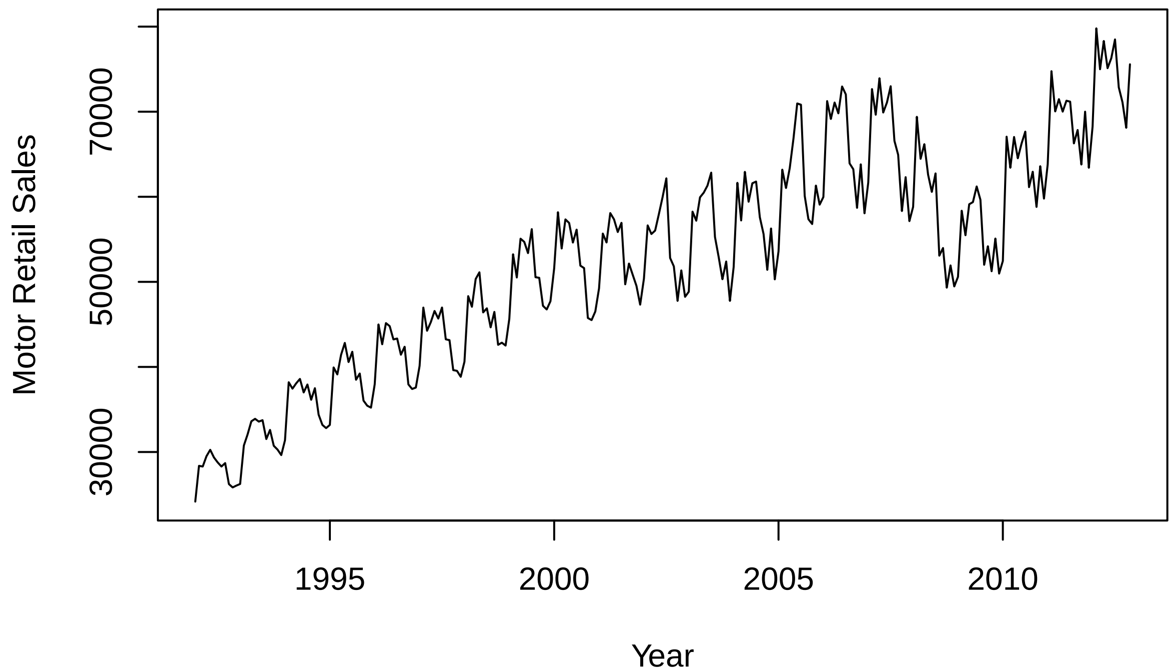
```
mau <- read.table("mauna.dat",header=TRUE,sep="")
mau <- ts(mau,start=1958,frequency=12)
plot(mau,ylab="Mauna Loa CO2",xlab="Year")
```



Example 1.2.4. Retail Sales of Motor Vehicles and Parts Dealers

- Monthly measurements of retail sales.
- Shows trend, monthly ($h = 12$) cycle, and Great Recession.

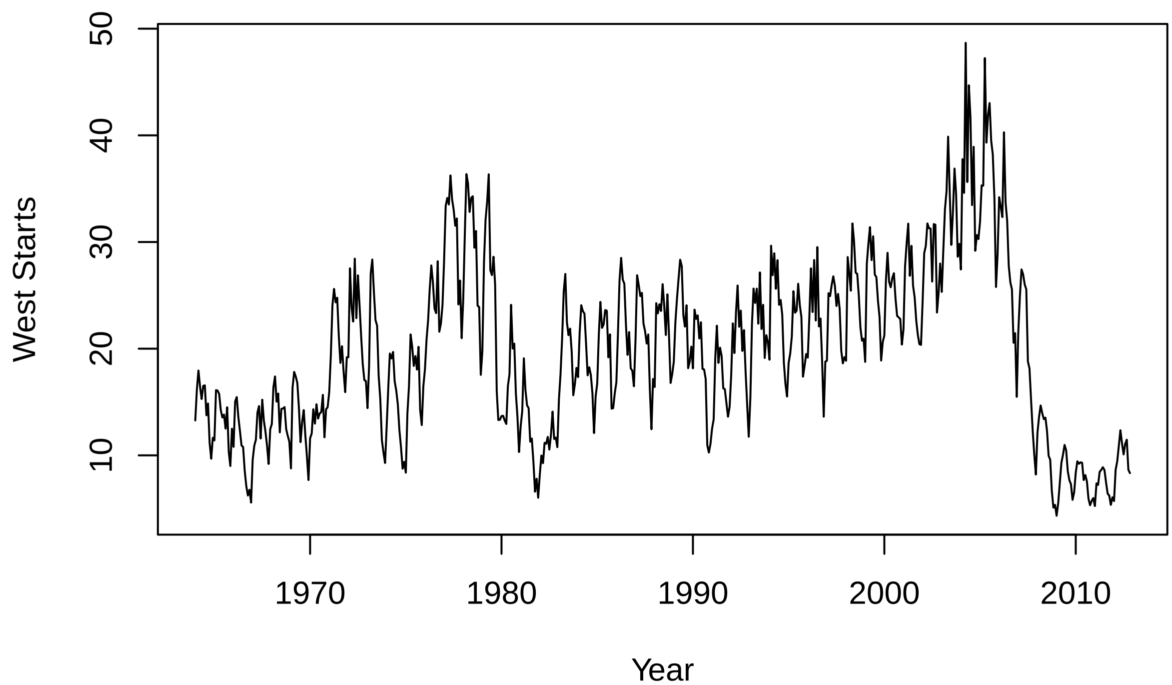
```
Ret441 <- read.table("retail441.b1",header=TRUE,skip=2)[,2]
Ret441 <- ts(Ret441,start = 1992,frequency=12)
plot(Ret441, ylab="Motor Retail Sales",xlab="Year")
```



Example 1.2.5. Housing Starts

- Monthly measurements of housing construction started (West Region).
- Shows trend, monthly ($h = 12$) cycle, and some recessions/expansions.

```
Wstarts <- read.table("Wstarts.b1",header=TRUE,skip=2)[,2]
Wstarts <- ts(Wstarts,start = 1964,frequency=12)
plot(Wstarts, ylab="West Starts",xlab="Year")
```



Lesson 1-3: Windows and Transforms

- To better visualize a time series, we may examine sub-spans, or use a transformation.

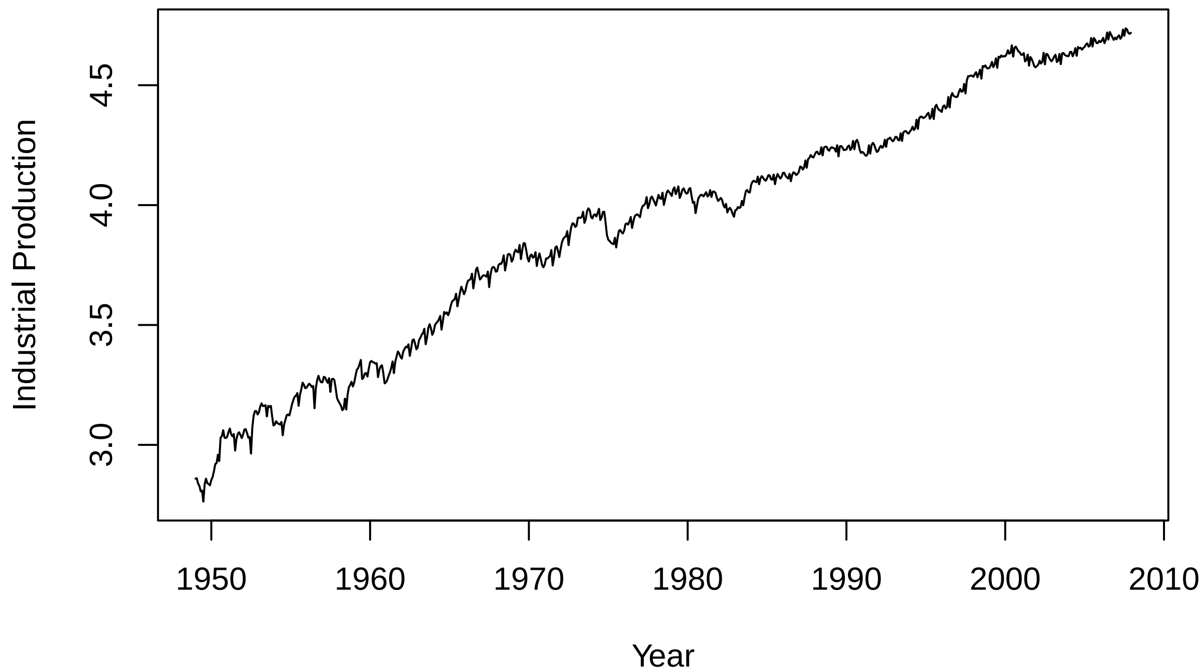
Windowing

- Focusing on a sub-section of the time series is called *windowing*.
- A *window* has a fixed width, and the starting and ending times change as it slides through the data.
- Windowing is useful for exploratory analysis, to visualize changes.

Example 1.3.2. Industrial Production

- Industrial Production is a monthly time series, starting in 1949.
- It has strong trend and moderate seasonality.

```
indprod <- read.table("ind.dat")
indprod <- ts(indprod,start=1949,frequency=12)
plot(indprod,xlab="Year",ylab="Industrial Production")
```



- We can create a moving window through the data.

```
### Movie
movie <- FALSE
delay <- 0
window <- 20
n <- length(indprod)/12
if(movie) {
for(t in 1:(n-window +1))
{
  Sys.sleep(delay)
  subsamp <- indprod[((t-1)*12+1):((t-1+window)*12)]
  newyear <- 1948 + t
  plot(ts(subsamp,start=newyear,frequency=12),ylab="")
} }
```

Log Transformation

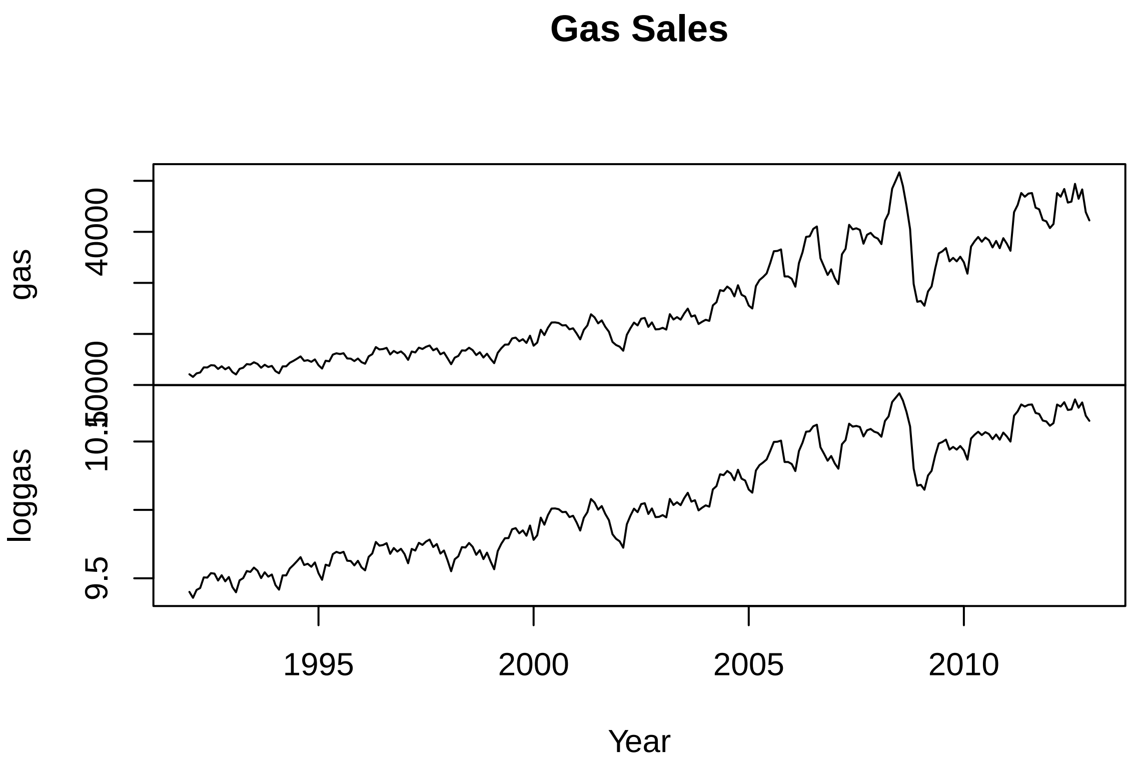
- To visualize and model time series better, sometimes we apply a log transform (if the data is positive).
- If cycle amplitude depends on trend level, applying a log may separate this effect so that cycle amplitude is no longer growing.
- Some extreme effects can be attenuated through the log transform.

Example 1.3.4. Gasoline Sales

- Monthly measurements of sales at gasoline stations.

- Variation depends on level, so we apply a log transformation.

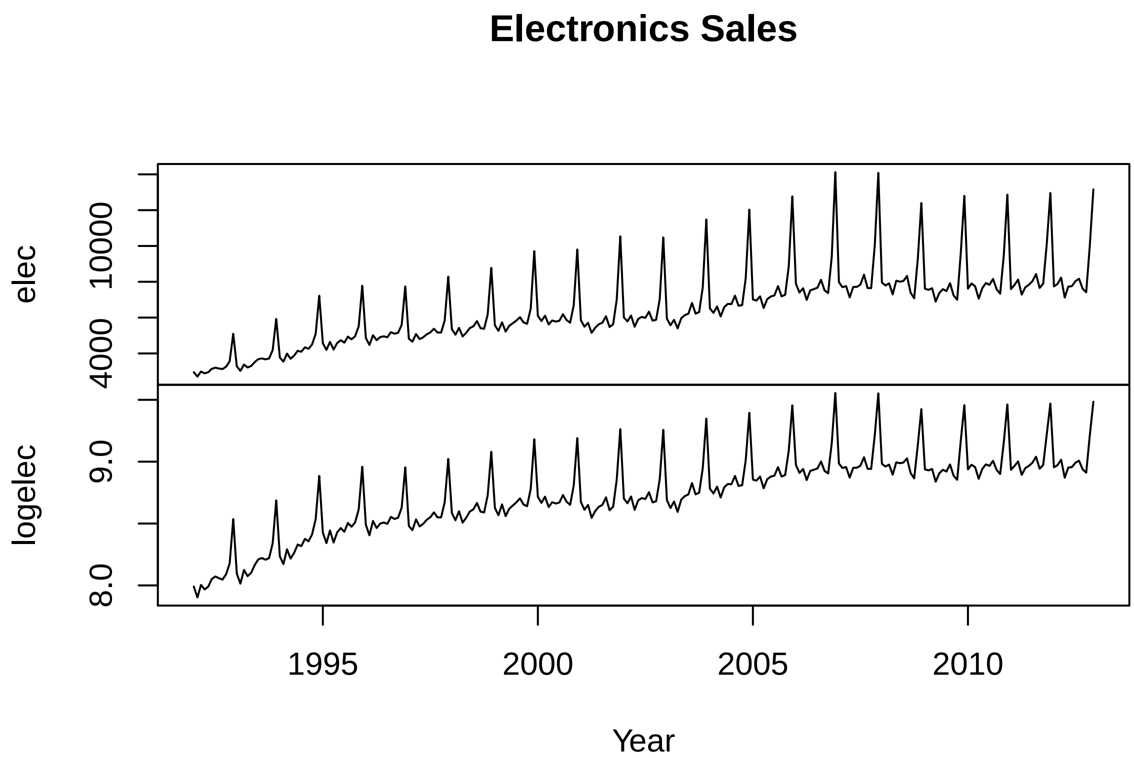
```
gas <- read.table("GasRaw_2-11-13.dat")[,1]
loggas <- log(gas)
gas_trans <- ts(cbind(gas,loggas),start = 1992,frequency=12)
plot(gas_trans,xlab="Year",main="Gas Sales")
```



Example 1.3.5. Electronics and Appliance Stores

- Monthly measurements of sales at electronics stores
- Large seasonal movements due to December sales
- We apply a log transformation

```
elec <- read.table("retail443.b1",header=FALSE,skip=2)[,2]
logelec <- log(elec)
elec_trans <- ts(cbind(elec,logelec),start = 1992,frequency=12)
plot(elec_trans, xlab="Year",main="Electronics Sales")
```



Lesson 1-4: Time Series Regression and Autoregression

- It is tempting to regress time series data on time $t = 1, \ldots, n$, as a covariate.

Rarely does this provide satisfactory results (when used alone).

Regression on Time Trend

- Regression model with time trend:

$$X_t = \beta_0 + \beta_1 t + Z_t.$$

  Is $\{Z_t\}$ i.i.d.? Usually: **No**.

Example 1.1.3. U.S. Population

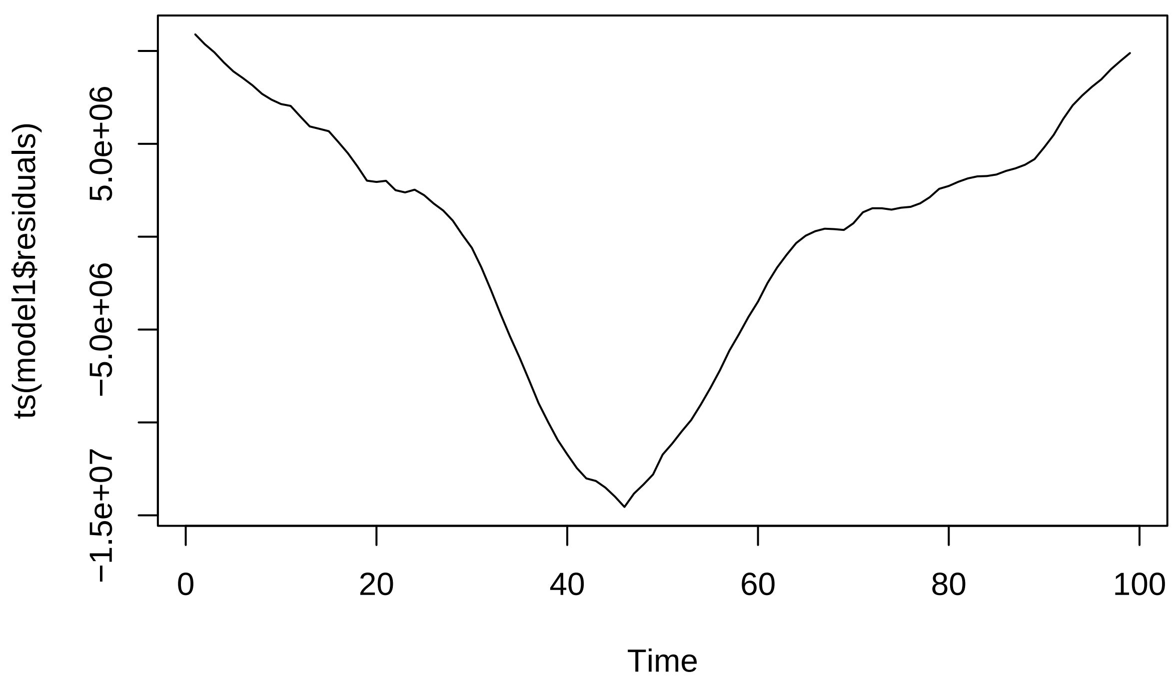
- Try out time trend regression for U.S. population.

```
pop <- read.table("USpop.dat")
pop <- ts(pop, start = 1901)
n <- length(pop)
time <- seq(1,n)
model1 <- lm(pop ~ time)
summary(model1)
```

```
##
## Call:
## lm(formula = pop ~ time)
```

```
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -14549597  -4786782   1606479   4952580  10891620
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 63186029    1432136   44.12   <2e-16 ***
## time         2016351      24868   81.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7071000 on 97 degrees of freedom
## Multiple R-squared:  0.9855, Adjusted R-squared:  0.9853
## F-statistic:  6575 on 1 and 97 DF,  p-value: < 2.2e-16
```

```r
plot(ts(model1$residuals))
```



- **Highly structured residuals!** We could add higher order polynomial effects to time trend, but it won't really help.

Regression on Past of Self

- Let past values of the time series be the covariates. Called *Autoregression*.
- Autoregressive model:

$$X_t = \rho X_{t-1} + Z_t.$$

Assume $\{Z_t\}$ i.i.d.

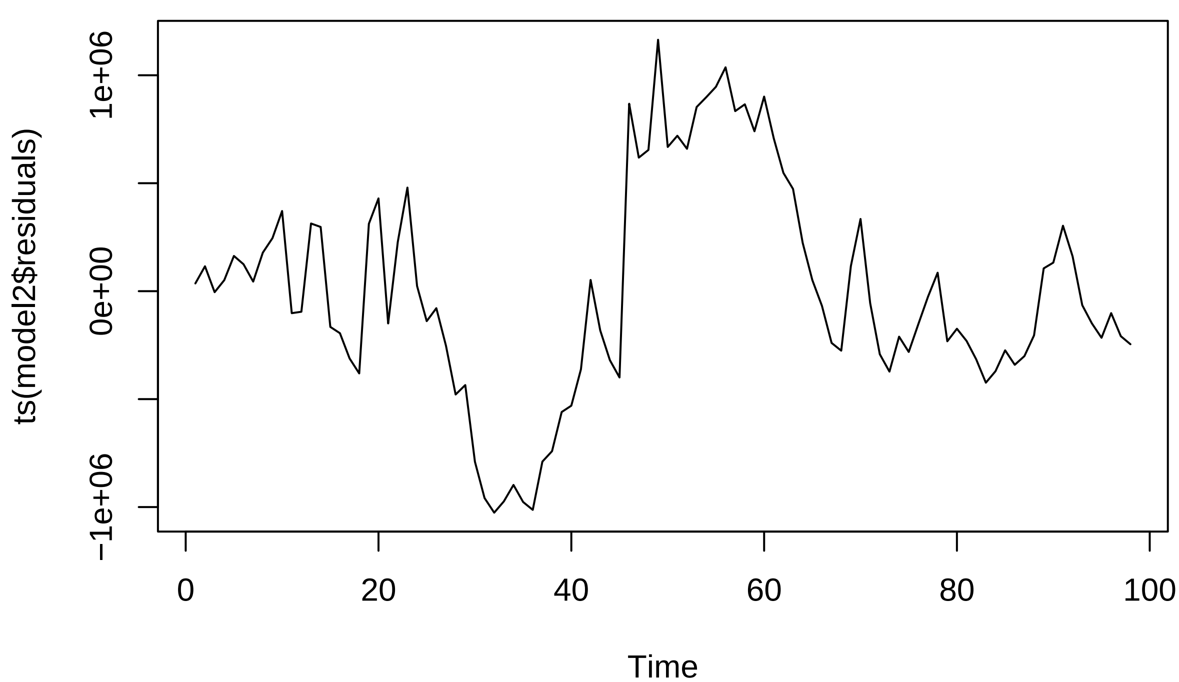
Example 1.1.3. U.S. Population

- Try out autoregression for U.S. population.
- Here we include a constant regressor as well.

```
model2 <- lm(pop[-1] ~ pop[-n])
summary(model2)
```

```
##
## Call:
## lm(formula = pop[-1] ~ pop[-n])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1025866  -289498   -73620   301639  1164336
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept) 9.705e+05  1.522e+05    6.376 6.33e-09 ***
## pop[-n]     1.006e+00  8.814e-04 1141.752  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 499900 on 96 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9999
## F-statistic: 1.304e+06 on 1 and 96 DF,  p-value: < 2.2e-16
```

```
plot(ts(model2$residuals))
```

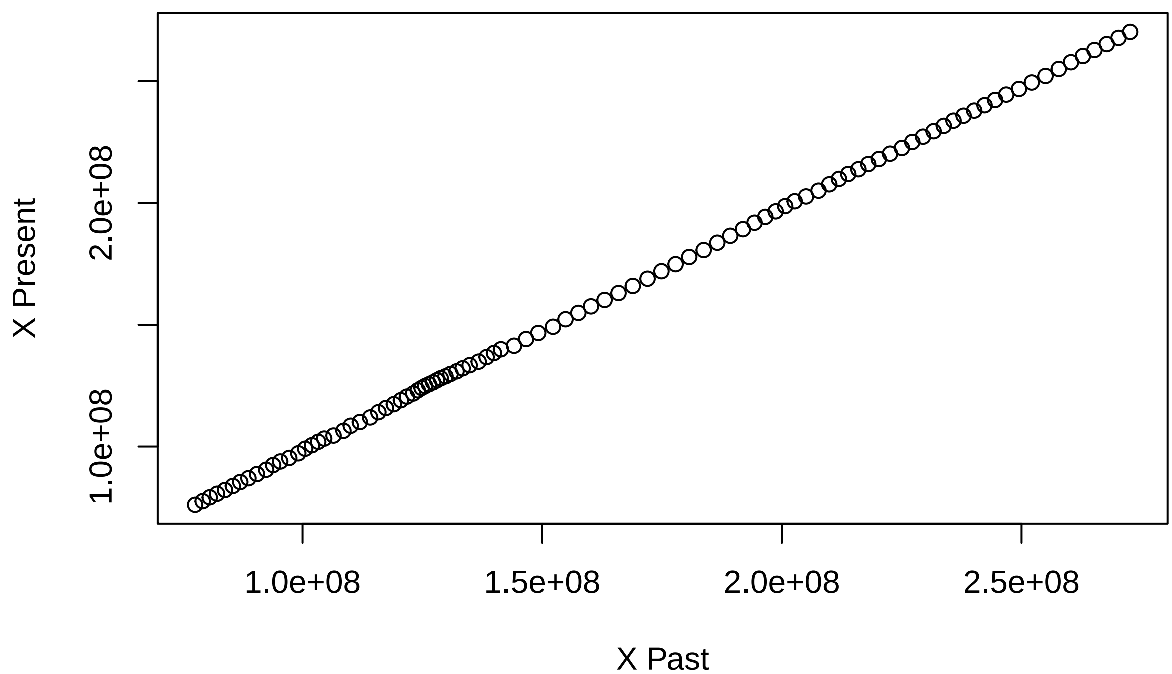


- Residuals are less structured, though maybe still not i.i.d.
- Why does this seem to work?

```r
cor(pop[-1],pop[-n])
```

```
## [1] 0.9999632
```

```r
plot(pop[-1],pop[-n],xlab="X Past",ylab="X Present")
```



Incorporate Time Trend

- We can incorporate time trend into an autoregression.
- One way to do it:

$$X_t = \rho X_{t-1} + \beta_0 + \beta_1 t + Z_t,$$

- This makes the mean $\mathbf{E}[X_t]$ depend on ρ.
- Another way to do it:

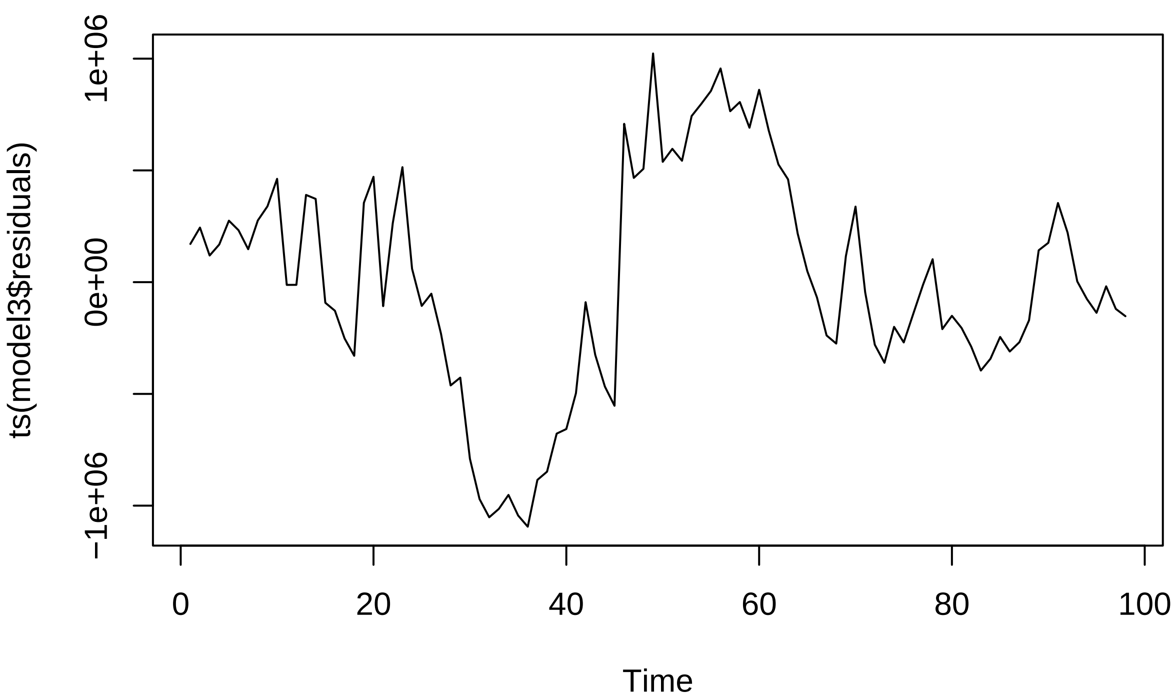
$$Y_t = \beta_0 + \beta_1 t + X_t$$
$$X_t = \rho\, X_{t-1} + Z_t.$$

```r
model3 <- lm(pop[-1] ~ pop[-n] + time[-1])
summary(model3)
```

```
##
## Call:
## lm(formula = pop[-1] ~ pop[-n] + time[-1])
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1094117  -273594   -31978   350442  1023219
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.636e+06  4.538e+05   3.606 0.000498 ***
## pop[-n]     9.952e-01  7.201e-03 138.202  < 2e-16 ***
## time[-1]    2.269e+04  1.458e+04   1.556 0.122985
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 496200 on 95 degrees of freedom
## Multiple R-squared:  0.9999, Adjusted R-squared:  0.9999
## F-statistic: 6.615e+05 on 2 and 95 DF,  p-value: < 2.2e-16
```

```r
plot(ts(model3$residuals))
```



- This implements the first approach
- Notice slope coefficient β_1 is not significant, and residuals resemble those of the pure autoregressive model. So not much benefit to using time covariate.