Intro to Time Series

Michael Winton

June 16, 2018

Introduction

Observations that have been collected over fixed sampling intervals are *historical time series*. They are treated as realizations of sequences of random variables.

```r
# load sample dataset
data(AirPassengers)
(ap <- AirPassengers)
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1949 112 118 132 129 121 135 148 148 136 119 104 118
## 1950 115 126 141 135 125 149 170 170 158 133 114 140
## 1951 145 150 178 163 172 178 199 199 184 162 146 166
## 1952 171 180 193 181 183 218 230 242 209 191 172 194
## 1953 196 196 236 235 229 243 264 272 237 211 180 201
## 1954 204 188 235 227 234 264 302 293 259 229 203 229
## 1955 242 233 267 269 270 315 364 347 312 274 237 278
## 1956 284 277 317 313 318 374 413 405 355 306 271 306
## 1957 315 301 356 348 355 422 465 467 404 347 305 336
## 1958 340 318 362 348 363 435 491 505 404 359 310 337
## 1959 360 342 406 396 420 472 548 559 463 407 362 405
## 1960 417 391 419 461 472 535 622 606 508 461 390 432
```

```r
class(ap)  # Note there's a specific class called ts
```

```
## [1] "ts"
```

```r
start(ap)
```

```
## [1] 1949    1
```

```r
end(ap)
```

```
## [1] 1960   12
```

```r
frequency(ap)
```

```
## [1] 12
```

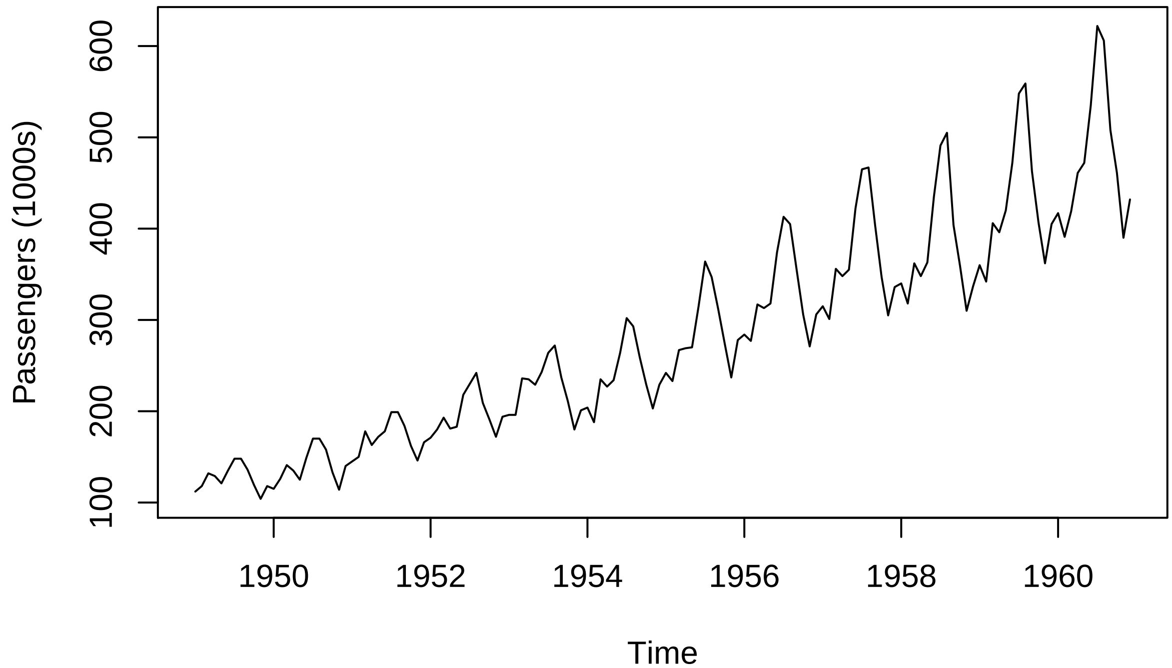
```r
# rollups by year
aggregate(ap)
```

```
## Time Series:
## Start = 1949
## End = 1960
## Frequency = 1
```

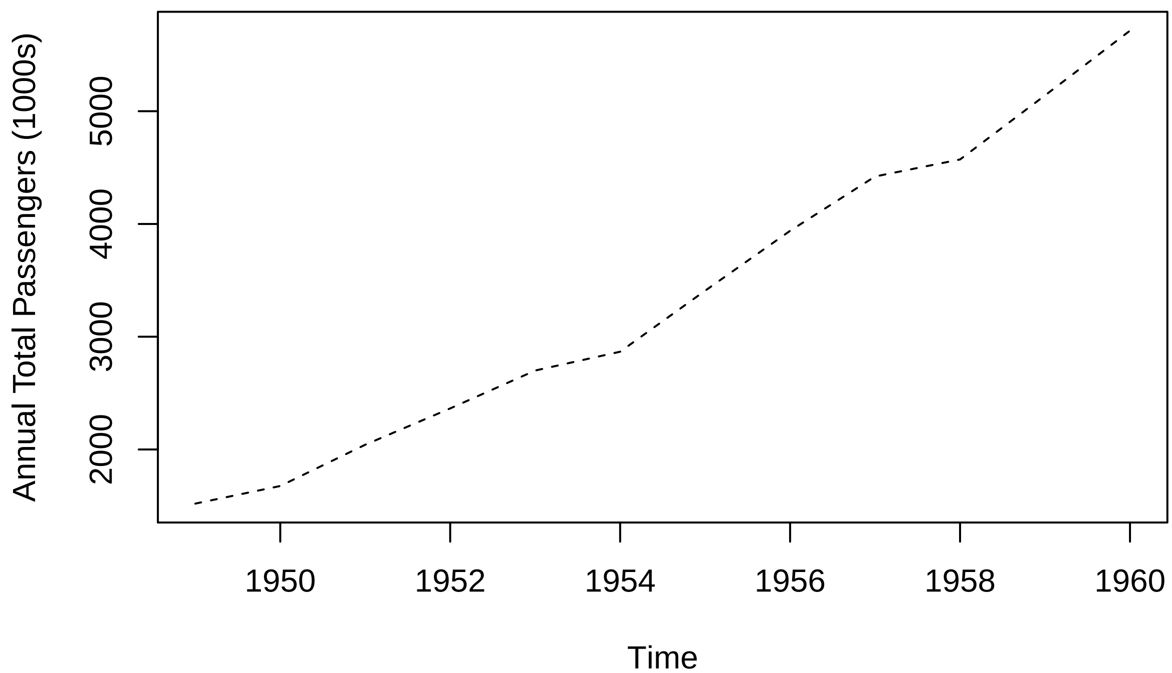
```
##  [1] 1520 1676 2042 2364 2700 2867 3408 3939 4421 4572 5140 5714
```

Several common plots are built in:

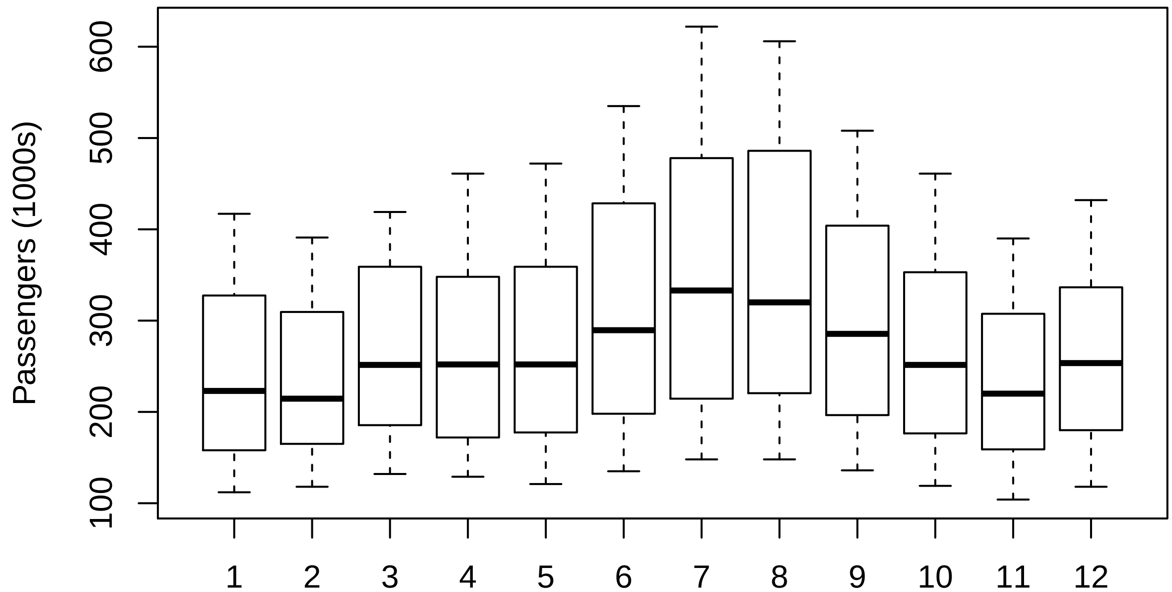
```r
# raw time series
plot(ap, ylab='Passengers (1000s)')
```




```r
# annual total counts
plot(aggregate(ap), lty=2, ylab='Annual Total Passengers (1000s)')
```



```r
# box plots by month
boxplot(ap ~ cycle(ap), ylab='Passengers (1000s)')
```



The simplest model for a (long-term) *trend* is often a linear approximation.

Converting dataframe to time series:

```r
# load data from internet and transform to a ts object
url <- 'https://raw.githubusercontent.com/mwinton/Introductory_Time_Series_with_R_datasets/mas
ME_month_df <- read.table(url, header=TRUE)
head(ME_month_df)
```

```
##   unemploy
## 1      6.7
## 2      6.7
## 3      6.4
## 4      5.9
## 5      5.2
## 6      4.8
```

```r
(ME_month_ts <- ts(ME_month_df$unemploy, start=c(1996, 1), freq=12))
```

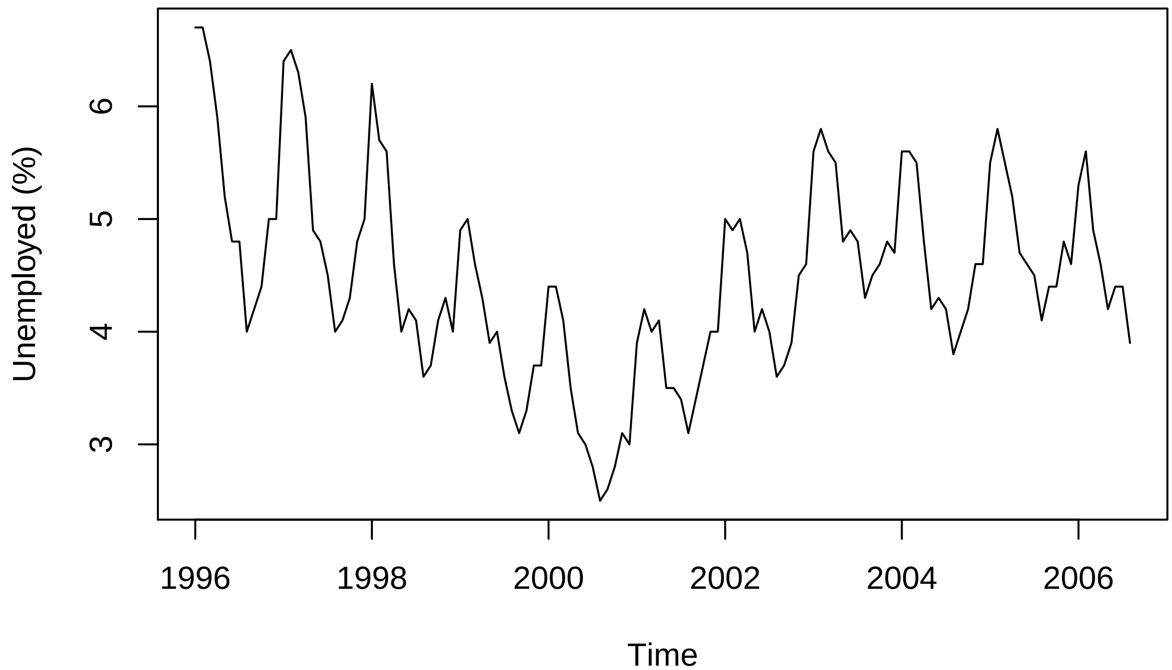
```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 1996 6.7 6.7 6.4 5.9 5.2 4.8 4.8 4.0 4.2 4.4 5.0 5.0
## 1997 6.4 6.5 6.3 5.9 4.9 4.8 4.5 4.0 4.1 4.3 4.8 5.0
## 1998 6.2 5.7 5.6 4.6 4.0 4.2 4.1 3.6 3.7 4.1 4.3 4.0
## 1999 4.9 5.0 4.6 4.3 3.9 4.0 3.6 3.3 3.1 3.3 3.7 3.7
## 2000 4.4 4.4 4.1 3.5 3.1 3.0 2.8 2.5 2.6 2.8 3.1 3.0
## 2001 3.9 4.2 4.0 4.1 3.5 3.5 3.4 3.1 3.4 3.7 4.0 4.0
## 2002 5.0 4.9 5.0 4.7 4.0 4.2 4.0 3.6 3.7 3.9 4.5 4.6
## 2003 5.6 5.8 5.6 5.5 4.8 4.9 4.8 4.3 4.5 4.6 4.8 4.7
## 2004 5.6 5.6 5.5 4.8 4.2 4.3 4.2 3.8 4.0 4.2 4.6 4.6
## 2005 5.5 5.8 5.5 5.2 4.7 4.6 4.5 4.1 4.4 4.4 4.8 4.6
```

```
## 2006 5.3 5.6 4.9 4.6 4.2 4.4 4.4 3.9
```

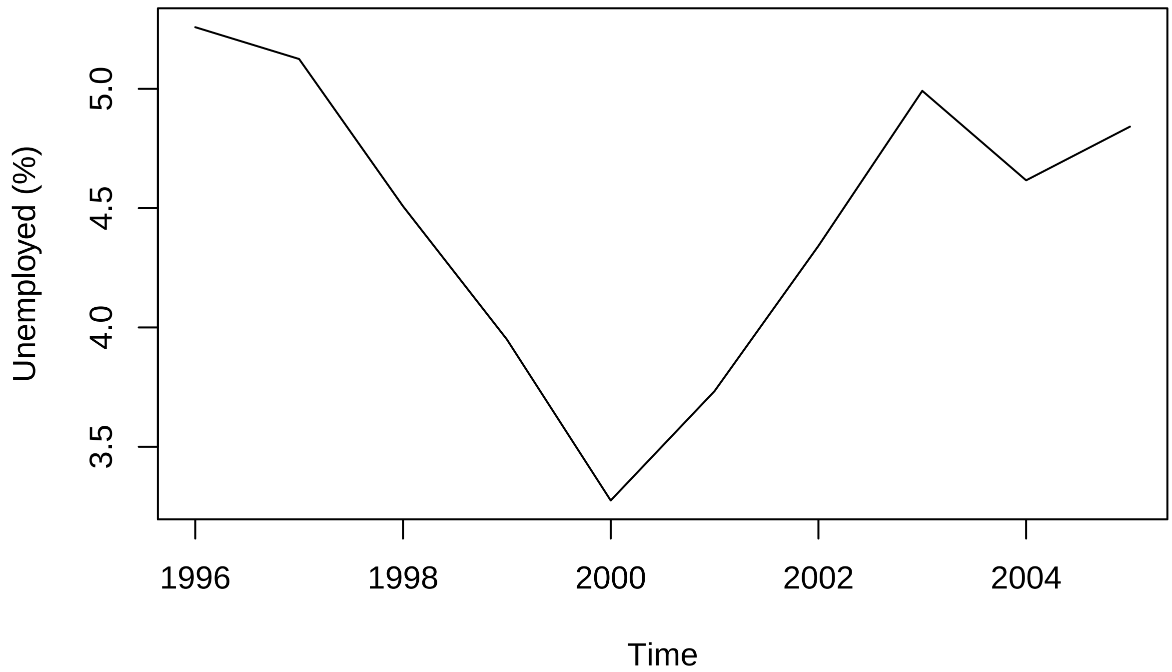
```r
(ME_annual_ts <- aggregate(ME_month_ts, FUN=mean))  # without FUN=mean, we get a sum by defaul
```

```
## Time Series:
## Start = 1996
## End = 2005
## Frequency = 1
##  [1] 5.258 5.125 4.508 3.950 3.275 3.733 4.342 4.992 4.617 4.842
```

```r
plot(ME_month_ts, ylab='Unemployed (%)')
```




```r
plot(ME_annual_ts, ylab='Unemployed (%)')
```



We can extract just a portion of the ts:

```
# Extract all February and August numbers
(ME_feb <- window(ME_month_ts, start=c(1996,2), freq=TRUE))
```

```
## Time Series:
## Start = 1996
## End = 2006
## Frequency = 1
##  [1] 6.7 6.5 5.7 5.0 4.4 4.2 4.9 5.8 5.6 5.8 5.6
```

```
(ME_aug <- window(ME_month_ts, start=c(1996,8), freq=TRUE))
```

```
## Time Series:
## Start = 1997
## End = 2007
## Frequency = 1
##  [1] 4.0 4.0 3.6 3.3 2.5 3.1 3.6 4.3 3.8 4.1 3.9
```

```
# Compare Feb, Aug to average monthly unemployment (from entire ts)
(feb_ratio <- mean(ME_feb) / mean(ME_month_ts))
```

```
## [1] 1.223
```

```
(aug_ratio <- mean(ME_aug) / mean(ME_month_ts))
```

```
## [1] 0.8164
```

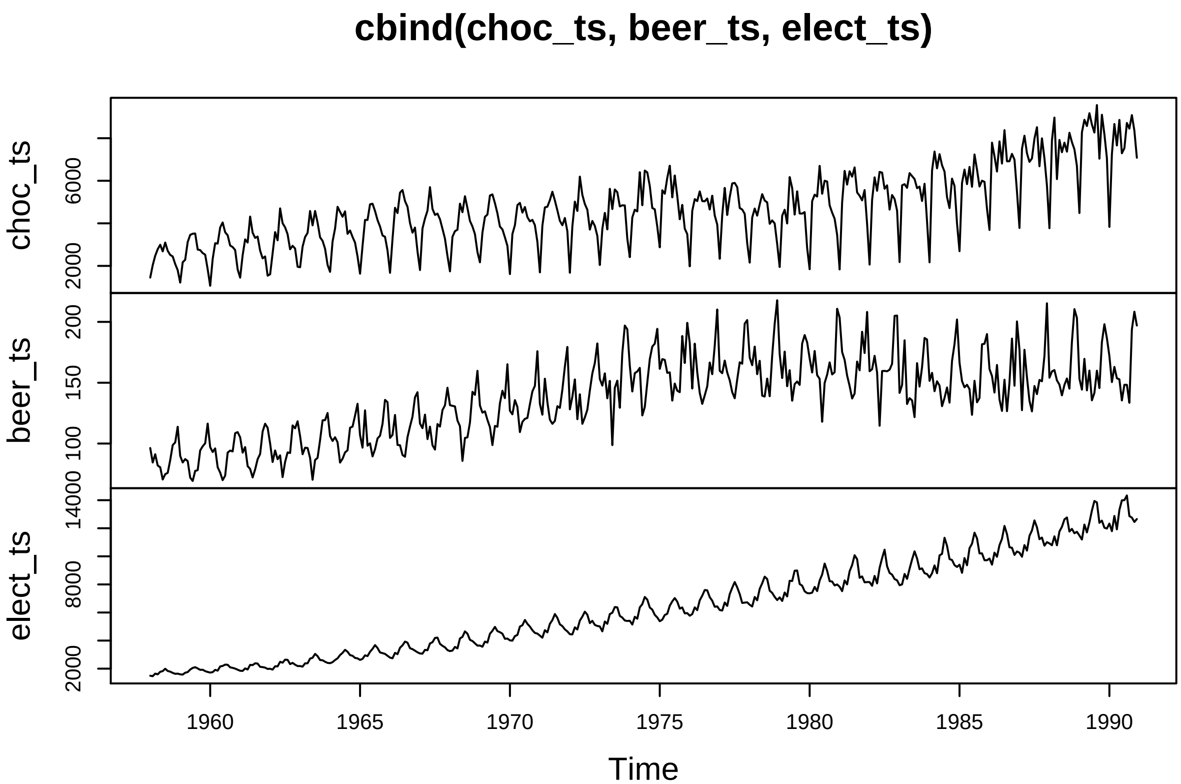
Multiple time series:

```
url <- 'https://raw.githubusercontent.com/mwinton/Introductory_Time_Series_with_R_datasets/mas
cbe_df <- read.table(url, header=TRUE)
head(cbe_df)
```

```
##   choc beer elec
## 1 1451 96.3 1497
## 2 2037 84.4 1463
## 3 2477 91.2 1648
## 4 2785 81.9 1595
## 5 2994 80.5 1777
## 6 2681 70.4 1824
```

```
choc_ts <- ts(cbe_df$choc, start=1958, freq=12)
beer_ts <- ts(cbe_df$beer, start=1958, freq=12)
elect_ts <- ts(cbe_df$elec, start=1958, freq=12)

# cool trick to use cbind to join plots together on shared t axis
plot(cbind(choc_ts, beer_ts, elect_ts))
```




```
# intersection of multiple time series
isect_ap_elec <- ts.intersect(ap, elect_ts)
start(isect_ap_elec)
```

```
## [1] 1958    1
```

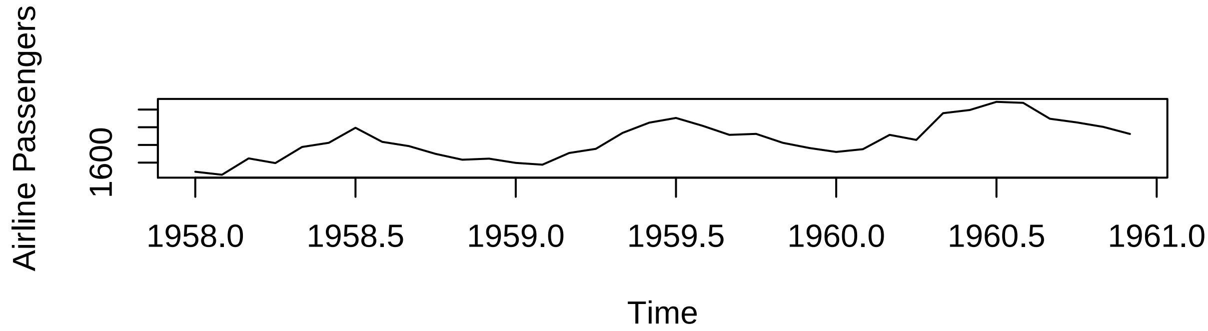
```r
end(isect_ap_elec)
```

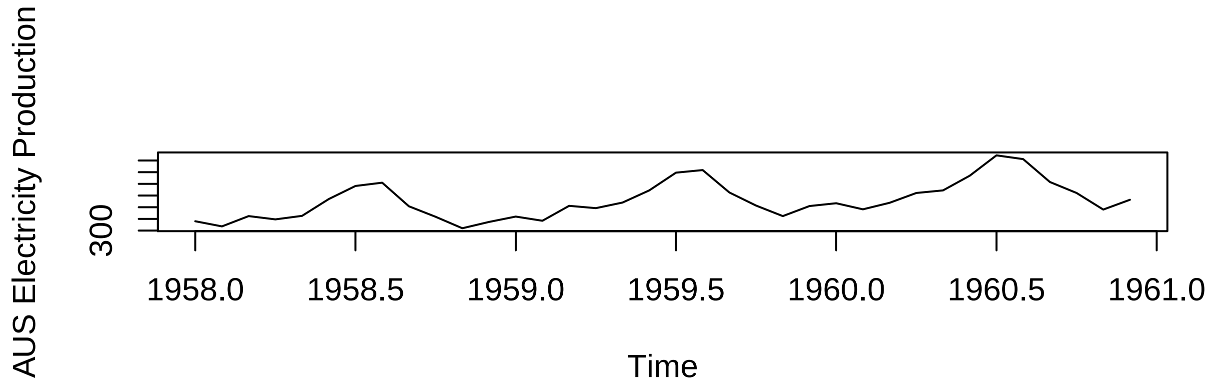
```
## [1] 1960   12
```

```r
head(isect_ap_elec)
```

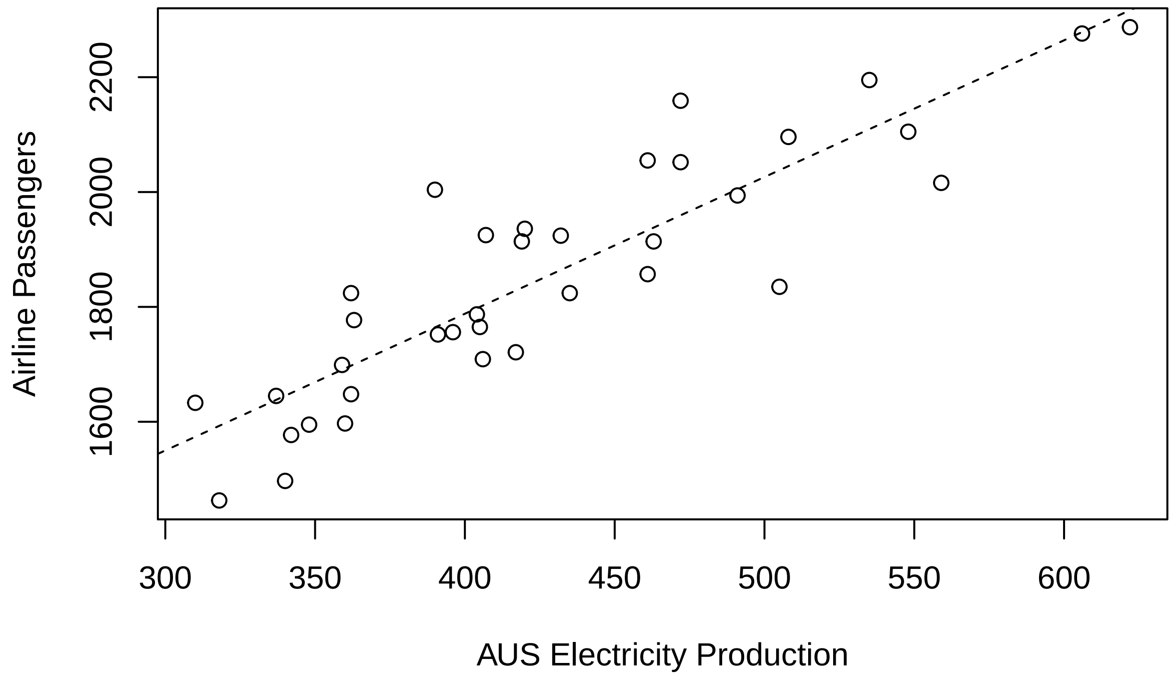
```
##       ap elect_ts
## [1,] 340     1497
## [2,] 318     1463
## [3,] 362     1648
## [4,] 348     1595
## [5,] 363     1777
## [6,] 435     1824
```

```r
# plot both
layout(1:2)
plot(isect_ap_elec[,2], ylab="Airline Passengers")
plot(isect_ap_elec[,1], ylab="AUS Electricity Production")
```






```r
# need to convert to vectors to do scatterplot
layout(1:1)
plot(as.vector(isect_ap_elec[,1]), as.vector(isect_ap_elec[,2]),
     xlab="AUS Electricity Production", ylab="Airline Passengers")
abline(lm(as.vector(isect_ap_elec[,2]) ~ isect_ap_elec[,1]), lty=2)
```



From this example, we see how easy it is to find correlation in two completely unrelated things. Because of this, its common to remove *trends* and *seasonal variation* before doing multiple time series analysis. This could mean working with residuals from a regression model that has terms to represent them.

Stochastic Trends in Financial Data

Day-to-day changes can be hard to explain, and it may be unrealistic to assume any *deterministic* component in a time series model. *Stochastic trends* can sometimes be fit with *random walk* models. They're common in financial data. There are also statistical tests to test for stochastic trends.

```
url <- 'https://raw.githubusercontent.com/mwinton/Introductory_Time_Series_with_R_datasets/mas
exchange_df <- read.table(url, header=TRUE)
head(exchange_df)
```

```
##   xrate
## 1 2.924
## 2 2.942
## 3 3.172
## 4 3.254
## 5 3.348
## 6 3.507
```

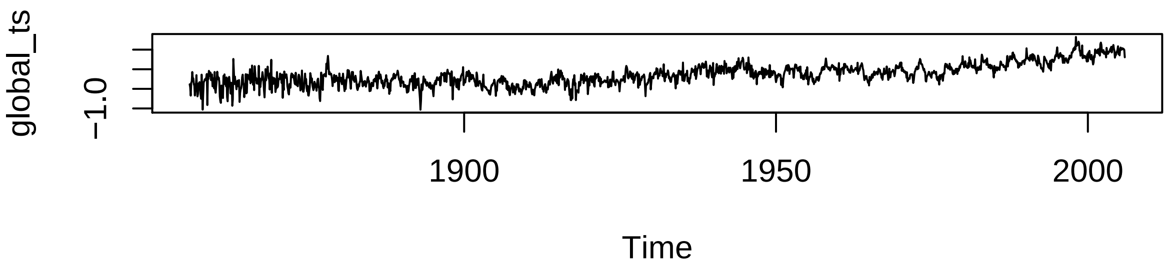
```
exchange_ts <- ts(exchange_df$xrate, start=1991, freq=4)
plot(exchange_ts, xlab='Quarter', ylab='Exchange Rate (NZ$ to GBP)')
```

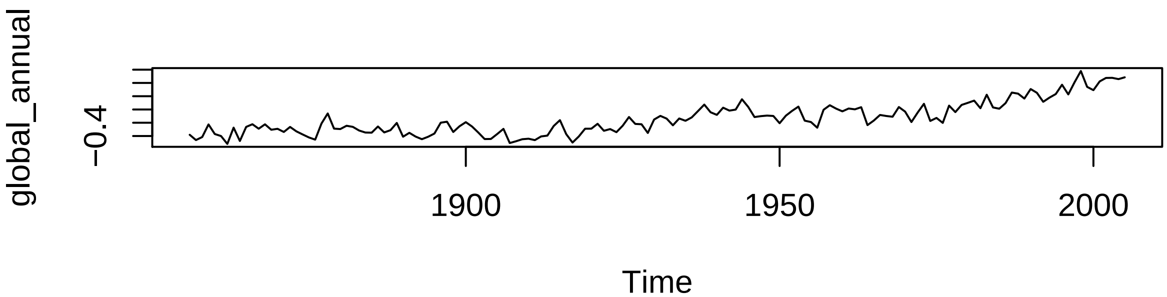

```
url <- 'https://raw.githubusercontent.com/mwinton/Introductory_Time_Series_with_R_datasets/mas
global_df <- scan(url)
head(global_df)
```

```
## [1] -0.384 -0.457 -0.673 -0.344 -0.311 -0.071
```

```
global_ts <- ts(global_df, start=c(1856, 1), end=c(2005, 12), freq=12)
# create ts for average annual value
global_annual <- aggregate(global_ts, FUN=mean)

# plot time series
layout(1:2)
plot(global_ts)
plot(global_annual)
```






```r
# plot subset of our ts
global_recent_ts <- window(global_ts, start=c(1970, 1), end=c(2005, 12))
recent_time_axis <- time(global_recent_ts)  # for regression
layout(1:1)
plot(global_recent_ts, ylab='Temp Increase (C)')
abline(reg=lm(global_recent_ts ~ recent_time_axis), lty=2)
```