Time Series

Emerson Amirhosein Azarbakht

Time Series (TS)

Used in

- control of inventory, based on demand trends
- airline's decision to buy airplanes bc of passenger trends and decision to increase/maintain market share
- climate change decisions based on temperature change trends
- business/sales forecasting
- everyday operational decisions
- long-term effects of proposed water management policies by simulating daily rainfall and sea state time series
- understanding fluctuations in monthly sales
- basis for signal processing in telecommunications <?>
- disease incidence tracking, yearly rates
- census analysis
- tracking monthly unemployment rate; as an economic indicator used by decision makers

Used to

- to understand the past, and predict the future
- forcasting (predicting inference, a subset of statistical inference). assumes that present trends continue. This assumption cannot be checked empirically, but, when we identify the likely causes for a trend, we can justify the forecasting(extrapolating it) for a few time-steps at least
- anomaly detection
- clustering
- classification (assigning a time series pattern to a specific category: e.g. gesture recognition of hand movements in sign language videos)
- query by content ~ Content-based image retrieval

Data: a variable measured sequentially in time, or at a fixed [sampling] interval

serial dependence problem: observations close together in time tend to be correlated (serially dependent)

TS tries to explain this correlation (serial dependence) autocorrelation analysis examines this serial dependence <?>

conditions (assumptions of TS)

- stationary process ?
- Ergodic process ?

```r
plot(AirPassengers)
start(AirPassengers)
```

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

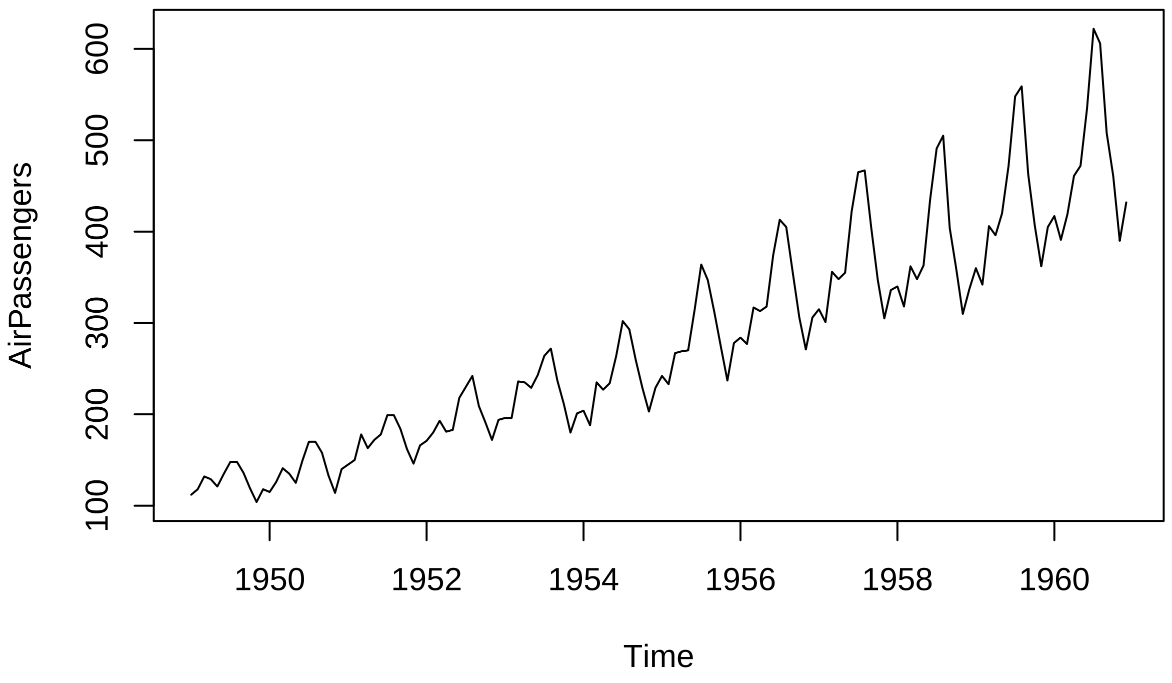
```r
end(AirPassengers)
```

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

```r
frequency(AirPassengers)
```

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

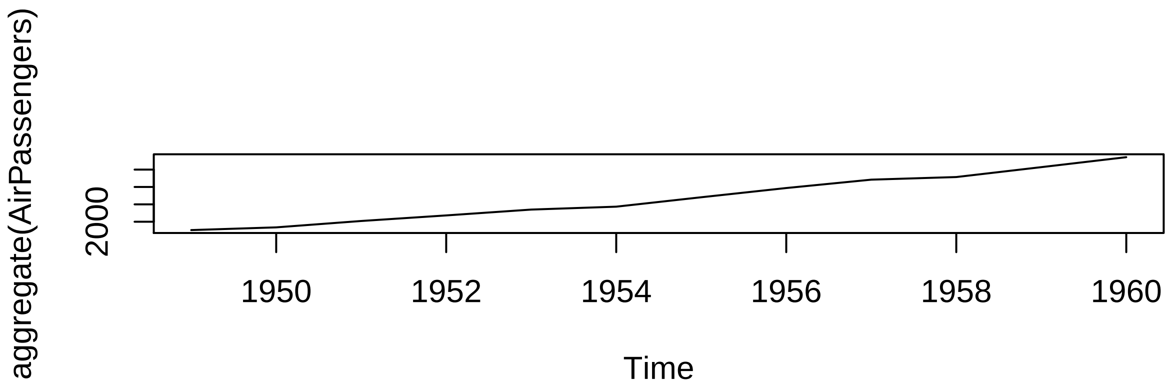
```r
plot(AirPassengers)
```

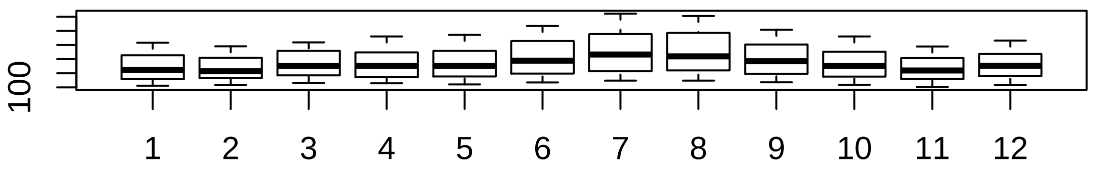


```r
summary(AirPassengers)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   104.0   180.0   265.5   280.3   360.5   622.0
```

```r
layout(1:2)
# takes an input matrix for the location of each plot in the graphics window
plot(aggregate(AirPassengers))
boxplot(AirPassengers ~ cycle(AirPassengers))
```

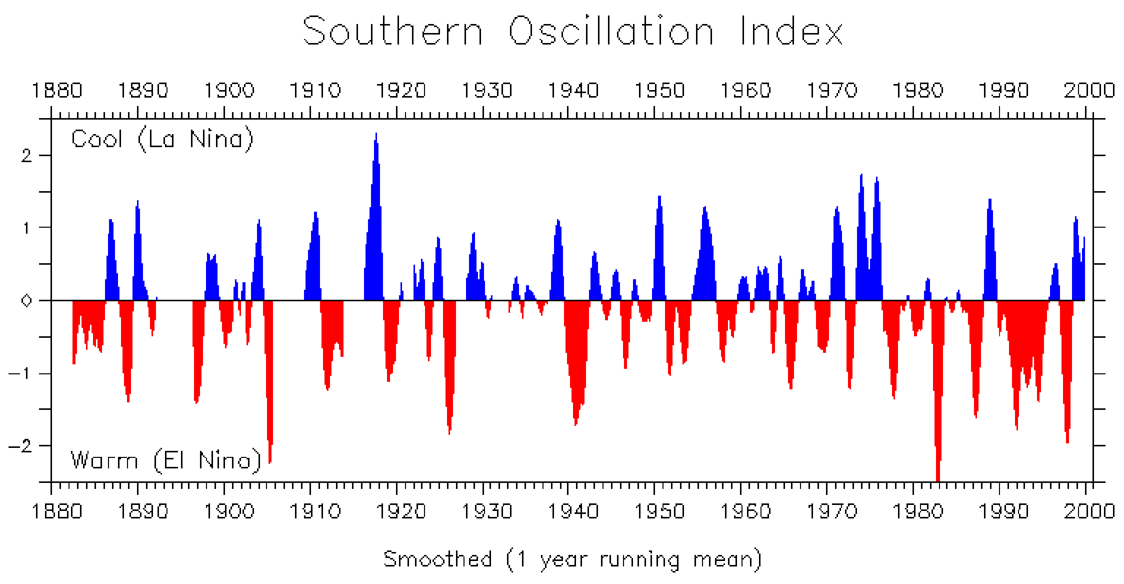




plotting shows *patterns*, and *features* of the data + *outliers* and *erroneous* values

patterns

1. trend = a non-periodic systematic change in a TS
   - can be modeled simply by a linear increase or decrease. (only if it's deterministic ~ it's non-stochastic)
   - stochastic trend: seems to change direction at unpredictable times rather than displaying a consistent pattern (e.g. like the air passenger series)
2. seasonal variation = a repeating pattern within a fixed period (e.g. each year)
3. cycles = a non-fixed-period cycle (without a fixed period). example: El-Nino



```r
# monthly unemployment rate for the US state of Maine from January 1996 until August 2006
Maine.month <- read.table("http://staff.elena.aut.ac.nz/Paul-Cowpertwait/ts/Maine.dat", header = TRUE)
# header TRUE means treat first row as column names
attach(Maine.month)
str(Maine.month)
```

Monthly unemployment rate for a state

```
## 'data.frame':    128 obs. of  1 variable:
##  $ unemploy: num  6.7 6.7 6.4 5.9 5.2 4.8 4.8 4 4.2 4.4 ...
```

```r
head(Maine.month)
```

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

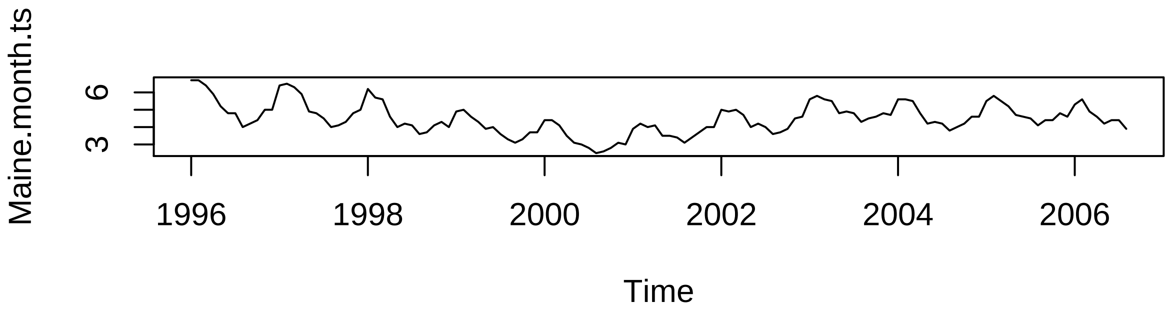
```r
class(Maine.month)
```

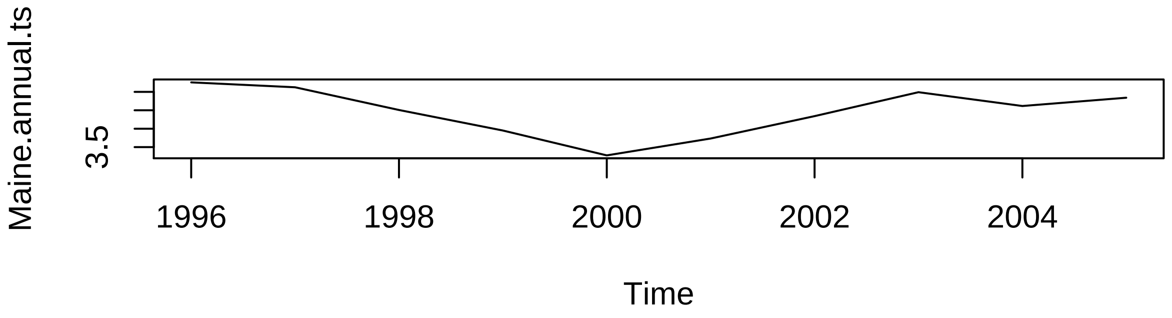
```
## [1] "data.frame"
```

```r
# it's a data.frame, not a ts object. So, we need to convert it to ts
Maine.month.ts <- ts(unemploy, start = c(1996, 1), freq = 12)
Maine.month.ts
```

```
##      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
```

```
Maine.annual.ts <- aggregate(Maine.month.ts)/12
layout(1:2)
plot(Maine.month.ts)
plot(Maine.annual.ts)
```





```
Maine.Feb <- window(Maine.month.ts, start = c(1996,2), freq = TRUE)
Maine.Aug <- window(Maine.month.ts, start = c(1996,8), freq = TRUE)
Feb.ratio <- mean(Maine.Feb) / mean(Maine.month.ts)
Aug.ratio <- mean(Maine.Aug) / mean(Maine.month.ts)
```

Multiple TS

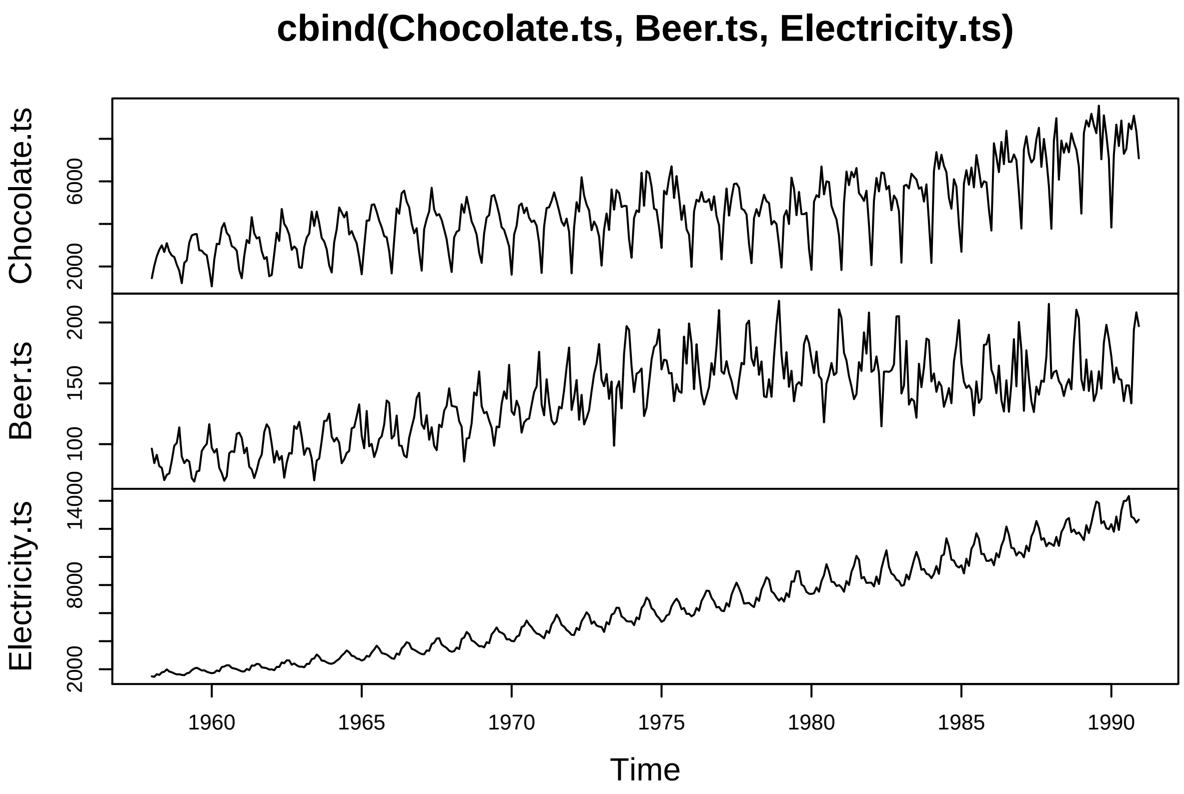
```
ChocolateBeerElectricity <- read.table("http://staff.elena.aut.ac.nz/Paul-Cowpertwait/ts/cbe.dat", head
class(ChocolateBeerElectricity)
```

```
## [1] "data.frame"
```

```
str(ChocolateBeerElectricity)
```

```
## 'data.frame':    396 obs. of  3 variables:
##  $ choc: int  1451 2037 2477 2785 2994 2681 3098 2708 2517 2445 ...
##  $ beer: num  96.3 84.4 91.2 81.9 80.5 70.4 74.8 75.9 86.3 98.7 ...
##  $ elec: int  1497 1463 1648 1595 1777 1824 1994 1835 1787 1699 ...
```

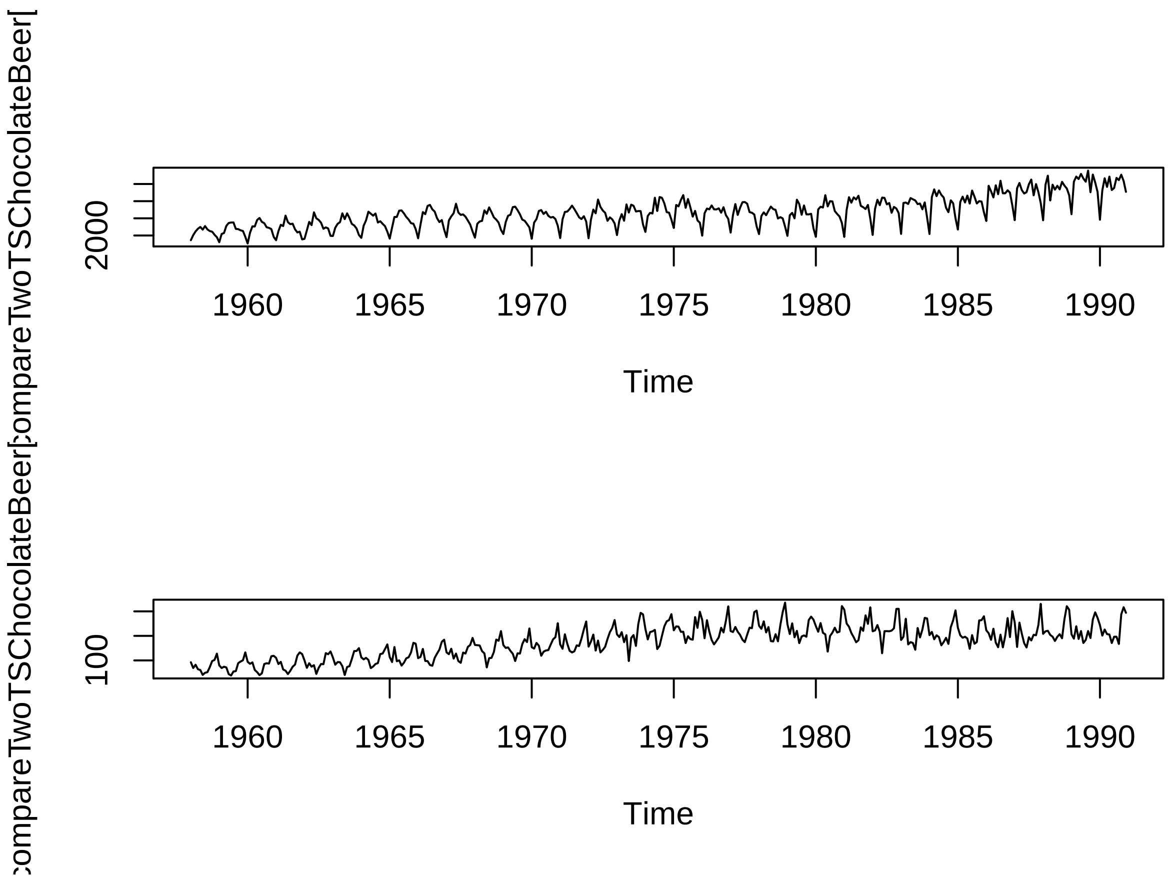
```
Chocolate.ts <- ts(ChocolateBeerElectricity[,1], start = 1958, frequency = 12)
Beer.ts <- ts(ChocolateBeerElectricity[,2], start = 1958, frequency = 12)
Electricity.ts <- ts(ChocolateBeerElectricity[,3], start = 1958, frequency = 12)

plot(cbind(Chocolate.ts, Beer.ts, Electricity.ts))
```



```
# intersection of multiple TS
compareTwoTSChocolateBeer <- ts.intersect(Chocolate.ts, Beer.ts)
# bind time series which have a common frequency
?ts.intersect()
head(compareTwoTSChocolateBeer)
```

```
##      Chocolate.ts Beer.ts
## [1,]         1451    96.3
## [2,]         2037    84.4
## [3,]         2477    91.2
## [4,]         2785    81.9
## [5,]         2994    80.5
## [6,]         2681    70.4
```

```
layout(1:2)
plot(compareTwoTSChocolateBeer[,1])
plot(compareTwoTSChocolateBeer[,2])
```



```
cor(Chocolate.ts, Beer.ts)
```

```
## [1] 0.3774314
```

```
# correlation
```

```
# Global temperature anomalies from the monthly means over the period
GlobalTemperatures <- scan("http://staff.elena.aut.ac.nz/Paul-Cowpertwait/ts/global.dat")
```

```r
# scan to read data into a vector or list from the console or file.
?scan
str(GlobalTemperatures)
```

Climate change

```
##  num [1:1800] -0.384 -0.457 -0.673 -0.344 -0.311 -0.071 -0.246 -0.235 -0.38 -0.418 ...
```

```r
GlobalTemperatures.ts <- ts(GlobalTemperatures, st = c(1856,1), end = c(2005,1), frequency = 12)
head(GlobalTemperatures.ts)
```

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

```r
tail(GlobalTemperatures.ts)
```

```
## [1] 0.436 0.452 0.494 0.586 0.385 0.502
```

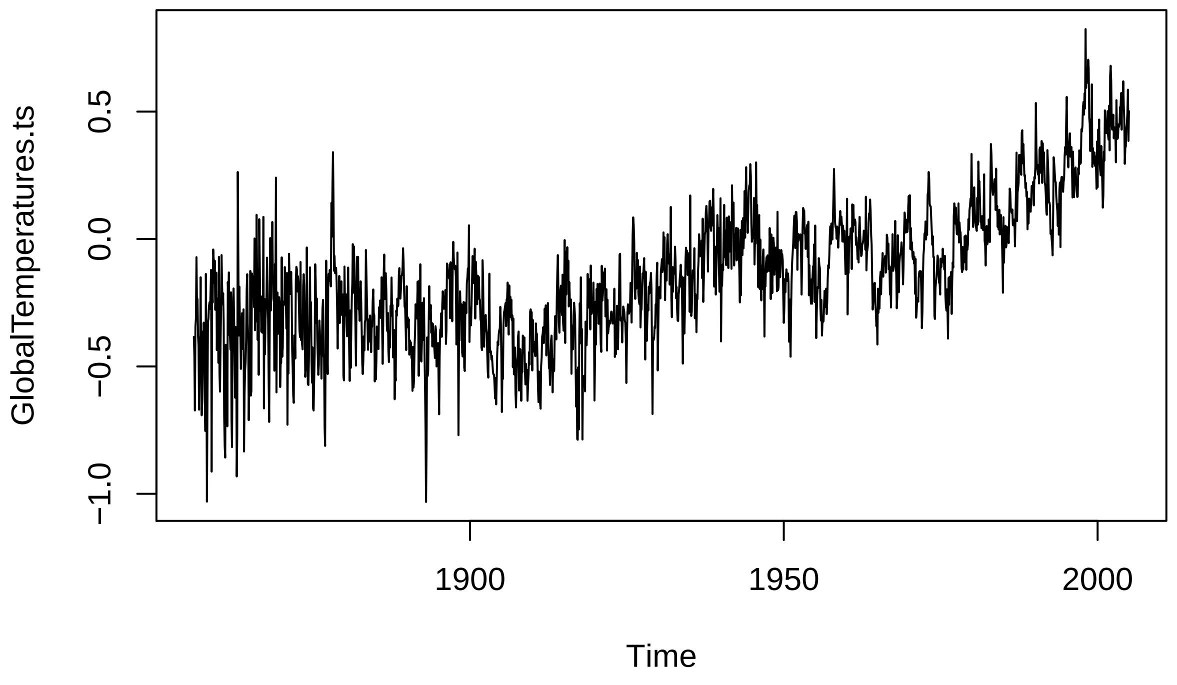
```r
Global.annual <- aggregate(GlobalTemperatures.ts, FUN = mean)
head(Global.annual)
```

```
## [1] -0.3812500 -0.4611667 -0.4153333 -0.2252500 -0.3697500 -0.4003333
```

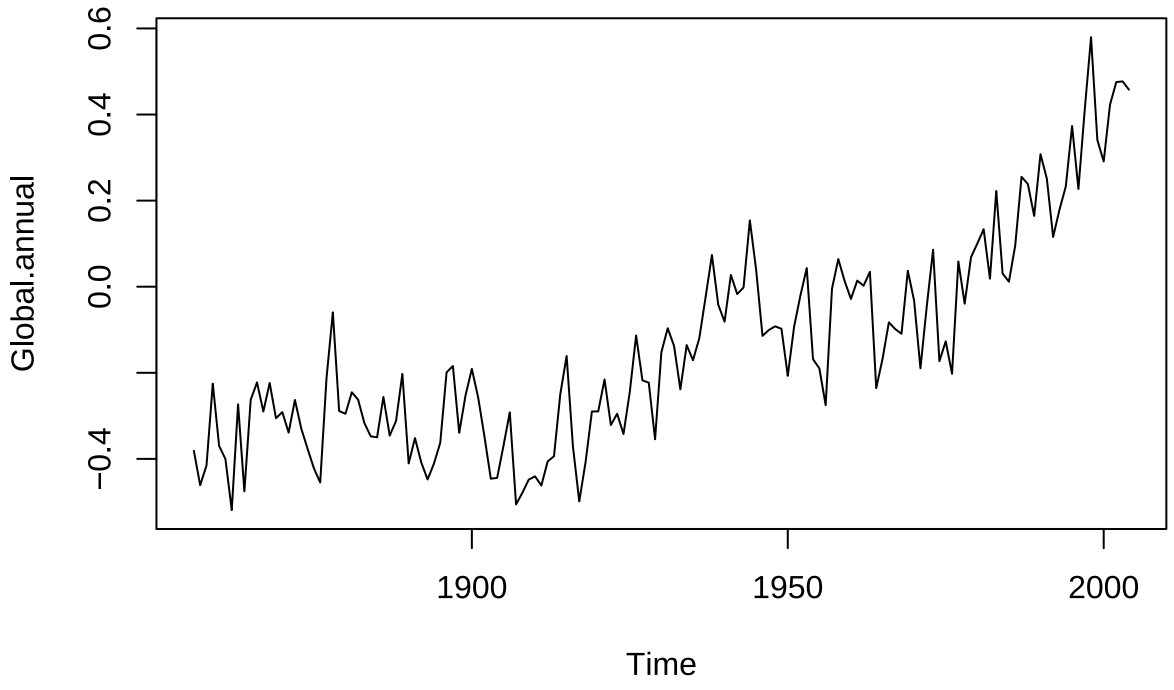
```r
Global.annual
```

```
## Time Series:
## Start = 1856
## End = 2004
## Frequency = 1
##   [1] -0.381250000 -0.461166667 -0.415333333 -0.225250000 -0.369750000
##   [6] -0.400333333 -0.518916667 -0.273333333 -0.475333333 -0.262583333
##  [11] -0.222416667 -0.289916667 -0.223583333 -0.305500000 -0.291333333
##  [16] -0.339000000 -0.263083333 -0.329583333 -0.376500000 -0.421583333
##  [21] -0.454750000 -0.212166667 -0.059500000 -0.288916667 -0.295166667
##  [26] -0.245333333 -0.262416667 -0.317166667 -0.347583333 -0.349916667
##  [31] -0.256083333 -0.345833333 -0.311916667 -0.202750000 -0.410416667
##  [36] -0.351750000 -0.408583333 -0.447833333 -0.411083333 -0.362666667
##  [41] -0.199083333 -0.184250000 -0.339416667 -0.252416667 -0.190916667
##  [46] -0.257583333 -0.348583333 -0.446000000 -0.444166667 -0.370000000
##  [51] -0.291916667 -0.505750000 -0.478750000 -0.448333333 -0.440500000
##  [56] -0.461833333 -0.405916667 -0.393916667 -0.249333333 -0.161000000
##  [61] -0.371416667 -0.498666667 -0.407333333 -0.289833333 -0.289750000
##  [66] -0.215500000 -0.321333333 -0.295083333 -0.342250000 -0.244333333
##  [71] -0.113416667 -0.217416667 -0.222916667 -0.354416667 -0.151833333
##  [76] -0.096500000 -0.137500000 -0.238333333 -0.135666667 -0.170750000
##  [81] -0.119416667 -0.023666667  0.073583333 -0.041916667 -0.081166667
##  [86]  0.027250000 -0.017166667 -0.001833333  0.154083333  0.038583333
##  [91] -0.114250000 -0.100666667 -0.092083333 -0.097666667 -0.207083333
##  [96] -0.093000000 -0.020416667  0.043250000 -0.168416667 -0.189583333
## [101] -0.275166667 -0.004750000  0.064166667  0.013250000 -0.028416667
## [106]  0.014166667  0.002166667  0.034500000 -0.235583333 -0.167583333
## [111] -0.082750000 -0.098416667 -0.109333333  0.037000000 -0.032500000
## [116] -0.189500000 -0.045833333  0.085833333 -0.173083333 -0.127000000
## [121] -0.202250000  0.058416667 -0.039416667  0.068416667  0.100333333
## [126]  0.133583333  0.018666667  0.222416667  0.030666667  0.011666667
## [131]  0.095500000  0.255166667  0.238666667  0.164416667  0.307916667
## [136]  0.251250000  0.115666667  0.178666667  0.232583333  0.373166667
## [141]  0.227000000  0.411000000  0.579666667  0.340500000  0.291083333
## [146]  0.422916667  0.475416667  0.476916667  0.457750000
```

```r
plot(GlobalTemperatures.ts)
```



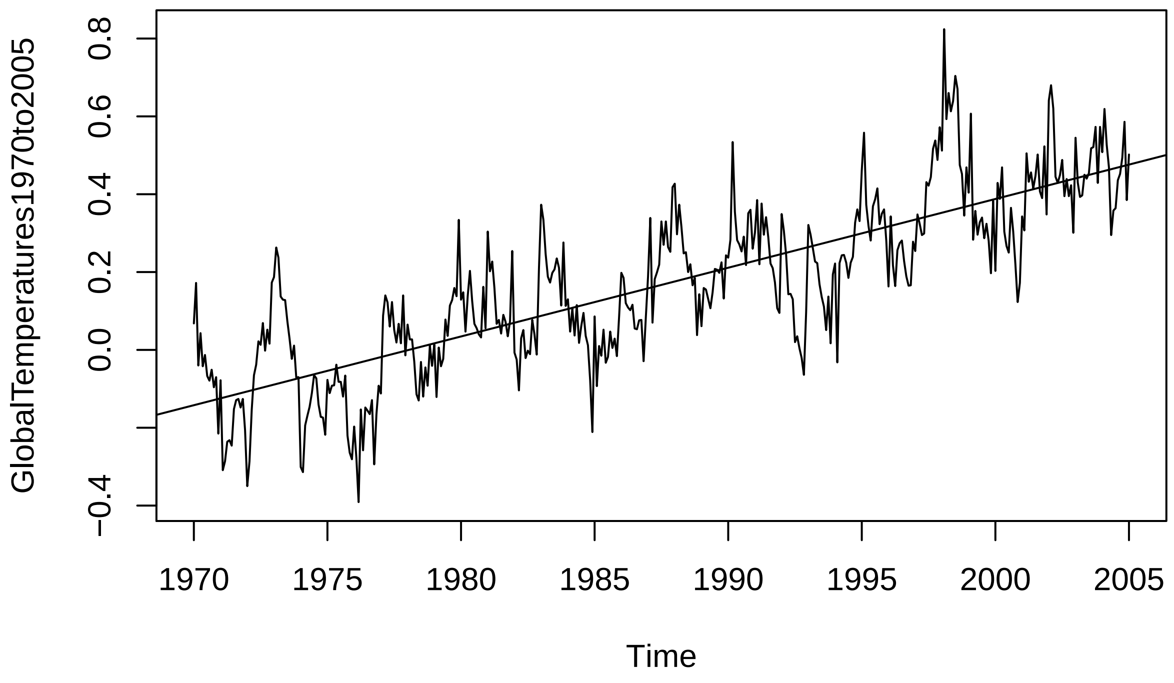
```r
plot(Global.annual)
```



```
#
GlobalTemperatures1970to2005 <- window(GlobalTemperatures.ts, start=c(1970, 1), end=c(2005, 12))
```

```
## Warning in window.default(x, ...): 'end' value not changed
```

```
# subset a time window
?window
GlobalTemperatures1970to2005.time <- time(GlobalTemperatures1970to2005)
# create the vector of times at which a time series was sampled
?time
plot(GlobalTemperatures1970to2005)
abline(reg = lm(GlobalTemperatures1970to2005 ~ GlobalTemperatures1970to2005.time))
```


GlobalTemperatures1970to2005
−0.4
0.0
0.2
0.4
0.6
0.8
1970
1975
1980
1985
1990
1995
2000
2005
Time