VectorBiTE Training 2018
Methods Workshop

Introduction to Autocorrelated Data and Time Series

www.vectorbite.org

Time series data and dependence

Time-series data are simply a collection of observations gathered over time. For example, suppose $y_1, \ldots, y_T$ are

- daily temperature,
- solar activity,
- CO_2 levels,
- GDP,
- yearly population size.

In each case, we might expect what happens at time t to be correlated with time $t - 1$.

Suppose we measure temperatures, daily, for several years.

Which would work better as an estimate for today's temp:

- The average of the temperatures from the previous year?
- The temperature on the previous day?

Would this change if the readings were iid $\mathcal{N}(\mu, \sigma^2)$?

Suppose we measure temperatures, daily, for several years.

Which would work better as an estimate for today's temp:

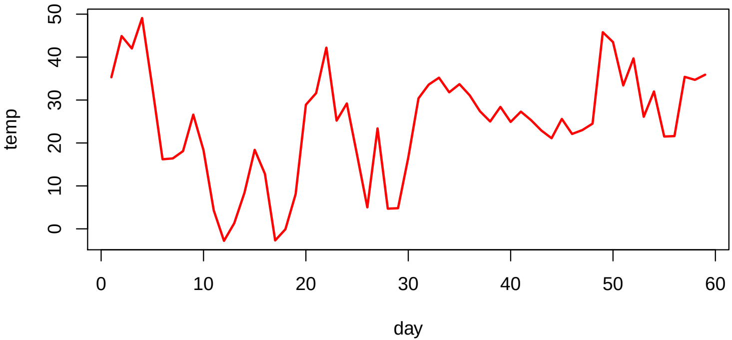
- The average of the temperatures from the previous year?
- The temperature on the previous day?

Would this change if the readings were iid $\mathcal{N}(\mu, \sigma^2)$?

Correlated errors require fundamentally different techniques.

Example: $Y_t =$ average daily temp. at O'Hare, Jan-Feb 1997.

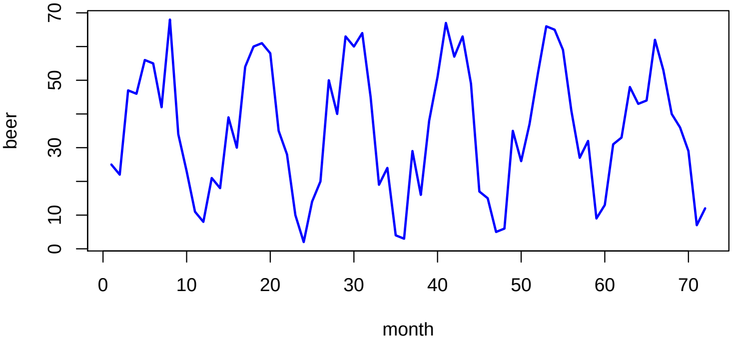
```
> weather <- read.csv("weather.csv")
> plot(weather$temp, xlab="day", ylab="temp", type="l",
+   col=2, lwd=2)
```



- "sticky" sequence: today tends to be close to yesterday.

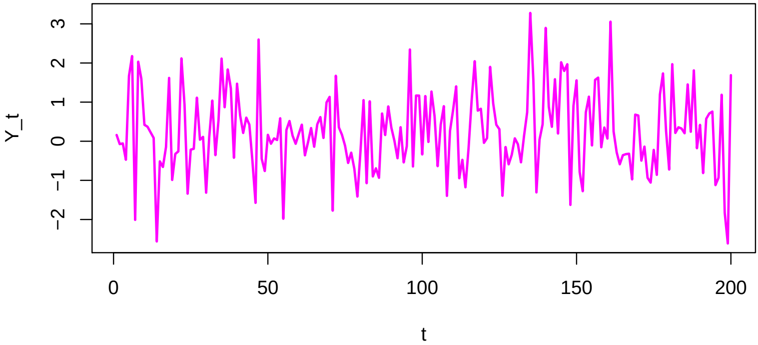
Example: Y_t = monthly U.S. beer production (Mi/barrels).

```
> beer <- read.csv("beer.csv")
> plot(beer$prod, xlab="month", ylab="beer", type="l",
+   col=4, lwd=2)
```



- The same pattern repeats itself year after year.

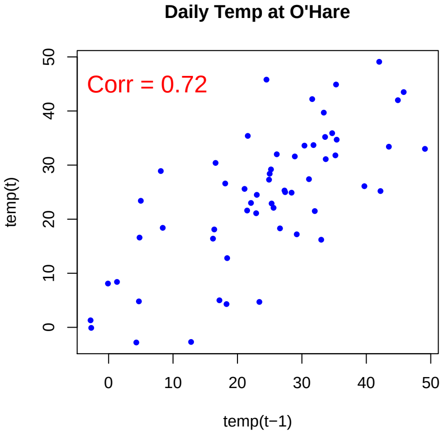
```
> plot(rnorm(200), xlab="t", ylab="Y_t", type="l",
+   col=6, lwd=2)
```



- It is tempting to see patterns even where they don't exist.
  How do we check?

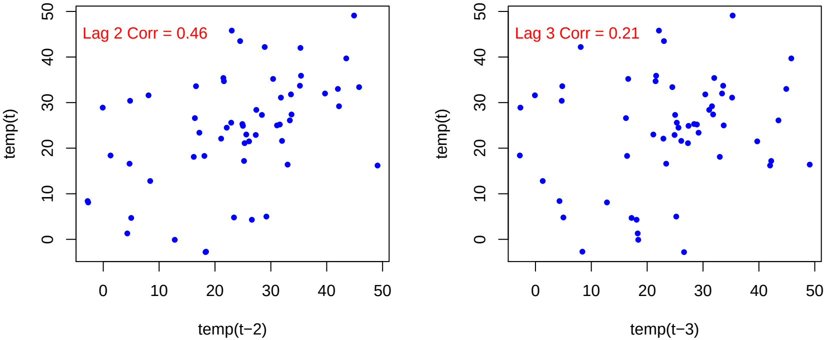
Checking for dependence

To see if Y_{t-1} would be useful for predicting Y_t, just plot them together and see if there is a relationship.



- Correlation between Y_t and Y_{t-1} is called autocorrelation.

We can plot Y_t against $Y_{t-\ell}$ to see ℓ-period lagged relationships.



- It appears that the correlation is getting weaker with increasing ℓ.

Autocorrelation

To summarize the time-varying dependence, compute lag-ℓ correlations for $\ell = 1, 2, 3, \ldots$

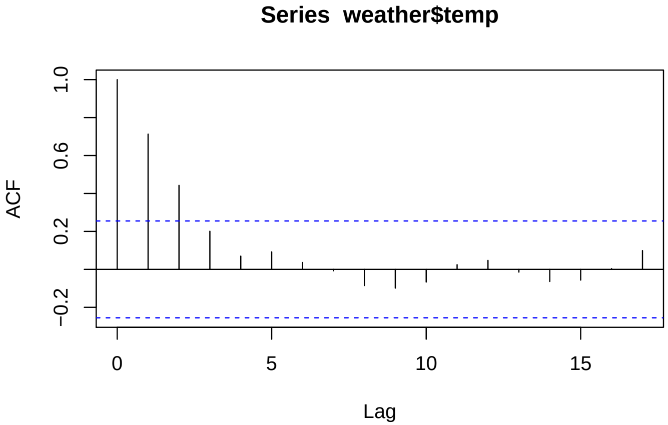
In general, the autocorrelation function (ACF) for Y is

$$r(\ell) = \mathrm{cor}(Y_t, Y_{t-\ell})$$

For our O'Hare temperature data:

```
> print(acf(weather$temp))
    0     1     2     3     4     5     6     7     8
 1.00  0.71  0.44  0.20  0.07  0.09  0.04 -0.01 -0.09
    9    10    11    12    13    14    15    16    17
-0.10 -0.07  0.03  0.05 -0.01 -0.06 -0.06  0.00  0.10
```

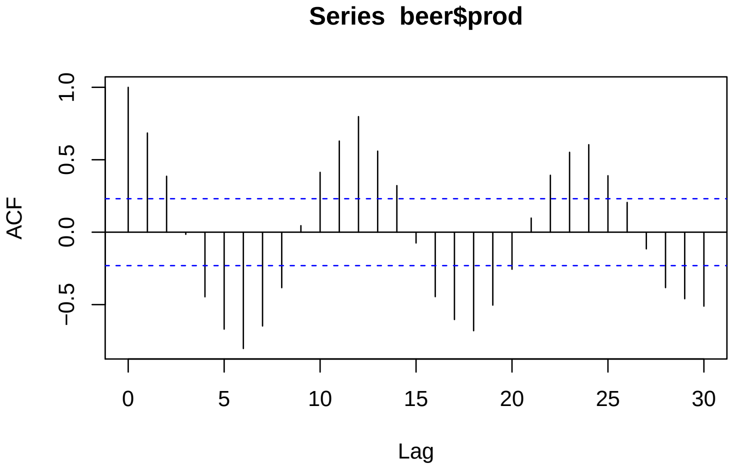
R's `acf` function shows the ACF visually.



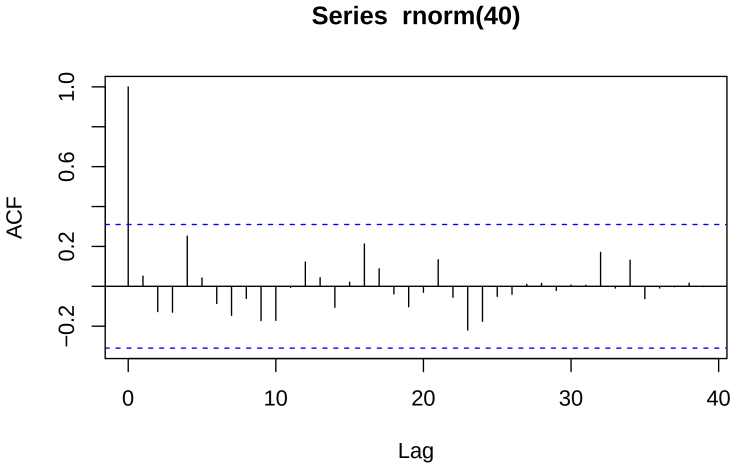
It provides a visual summary of our data dependence.

(Blue lines mark "statistical significance" for the `acf` values.)

The acf plot for the beer data shows an alternating dependence structure which causes time series oscillations.



An acf plot for *iid* normal data shows no significant correlation.



Autoregression

How do we model data that exhibits autocorrelation?

Suppose $Y_1 = \varepsilon_1$, $Y_2 = \varepsilon_1 + \varepsilon_2$, $Y_3 = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$, etc.

Then $Y_t = \sum_{i=1}^{t} \varepsilon_i = Y_{t-1} + \varepsilon_t$ and $\mathbb{E}[Y_t] = Y_{t-1}$.

This is called a random walk model for Y_t:

- the expectation of what will happen is always what happened most recently.

Even though Y_t is a function of errors going all the way back to the beginning, you can write it as depending only on Y_{t-1}.

Random walks are just a version of a more general model ...

The autoregressive model of order one holds that

$$AR(1): Y_t = \beta_0 + \beta_1 Y_{t-1} + \varepsilon_t, \quad \varepsilon_t \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2).$$

This is just a SLR model of Y_t regressed onto lagged Y_{t-1}.

It assumes all of our standard regression model conditions.

- The residuals should look *iid* and be uncorrelated with $\hat{Y}_t$.
- All of our previous diagnostics and transforms still apply.

$$AR(1) : Y_t = \beta_0 + \beta_1 Y_{t-1} + \varepsilon_t$$

Again, Y_t depends on the past only through Y_{t-1}.

- Previous lag values ($Y_{t-2}, Y_{t-3}, \ldots$) do not help predict Y_t if you already know Y_{t-1}.

Think about daily temperatures:

- If I want to guess tomorrow's temperature (without the help of a meterologist!), it is sensible to base my prediction on today's temperature, ignoring yesterday's.

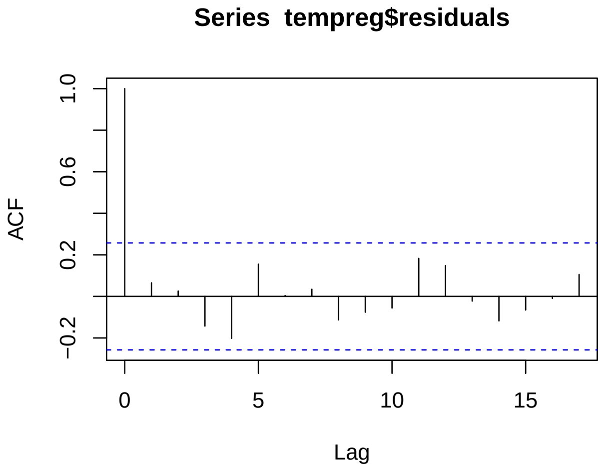
For the O'Hare temperatures, there is a clear autocorrelation.

```
> tempreg <- lm(weather$temp[2:59] ~ weather$temp[1:58])
> summary(tempreg)  ## abbreviated output

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)         6.70580    2.51661   2.665   0.0101 *
weather$temp[1:58]  0.72329    0.09242   7.826  1.5e-10 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 8.79 on 56 degrees of freedom
Multiple R-squared:  0.5224,  Adjusted R-squared:  0.5138
F-statistic: 61.24 on 1 and 56 DF,  p-value: 1.497e-10
```

- The autoregressive term ($b_1 \approx 0.7$) is highly significant!

We can check residuals for any "left-over" correlation.

```
> acf(tempreg$residuals)
```



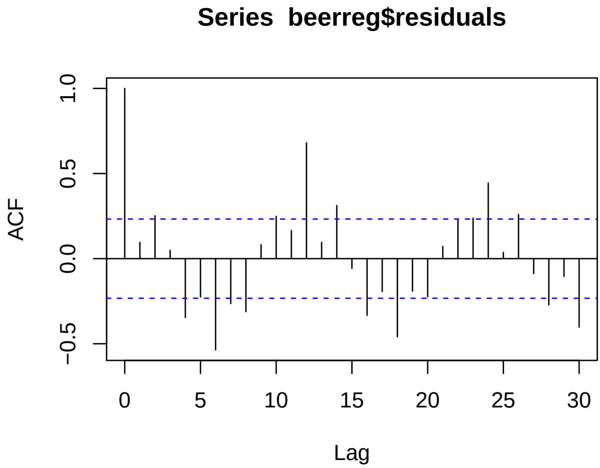
- Looks like we've got a good fit.

For the beer data, the autoregressive term is also highly significant.

```
> beerreg <- lm(beer$prod[2:72] ~  beer$prod[1:71])
> summary(beerreg) ## abbreviated output

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     10.64818    3.56983   2.983  0.00395 **
beer$prod[1:71]  0.69960    0.08748   7.997 2.02e-11 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 14.08 on 69 degrees of freedom
Multiple R-squared:  0.481, Adjusted R-squared:  0.4735
F-statistic: 63.95 on 1 and 69 DF,  p-value: 2.025e-11
```

But residuals show a clear pattern of left-over autocorrelation.

```
> acf(beerreg$residuals)
```



- We'll talk later about how to model this type of pattern ...

Many different types of series may be written as an AR(1).

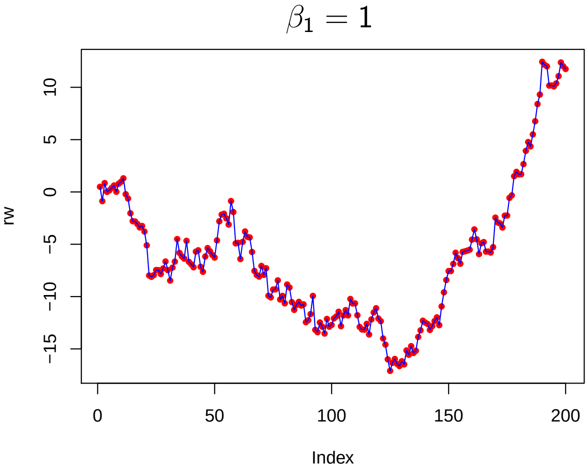
$$AR(1): Y_t = \beta_0 + \beta_1 Y_{t-1} + \varepsilon_t$$

The value of β_1 is key!

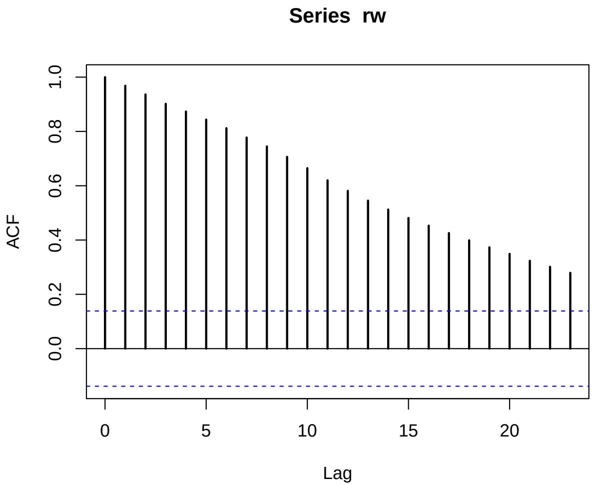
- If $|\beta_1| = 1$, we have a *random walk*.
- If $|\beta_1| > 1$, the series *explodes*.
- If $|\beta_1| < 1$, the values are *mean reverting*.

Random walk

In a random walk, the series just wanders around.



Autocorrelation of a random walk stays high for a long time.



The random walk has some special properties ...

$Y_t - Y_{t-1} = \beta_0 + \varepsilon_t$, and β_0 is called the "drift parameter".

The series is nonstationary:

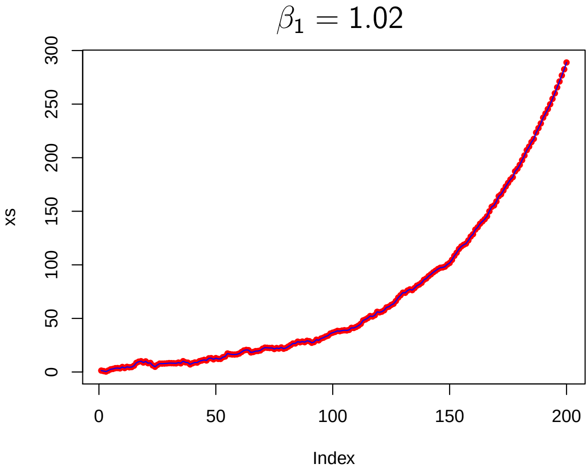
- it has no average level that it wants to be near, but rather just wanders off into space.

The random walk without drift ($\beta_0 = 0$) is a common model for simple processes

- since $\mathbb{E}[Y_t] = \mathbb{E}[Y_{t-1}]$, e.g., tomorrow $\approx$ today

Exploding series

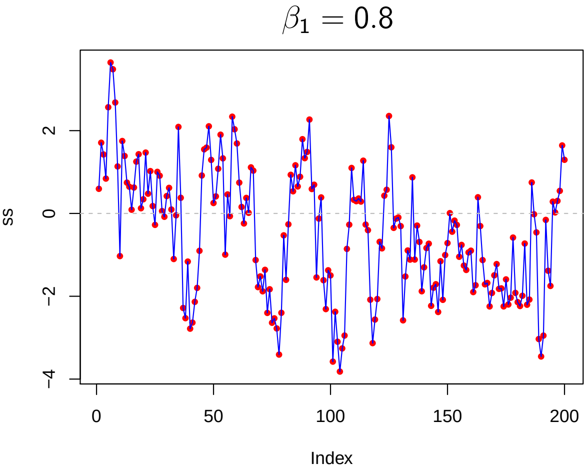
For AR term > 1, the Y_t's move exponentially far from Y_1.



- Useless for modeling and prediction.

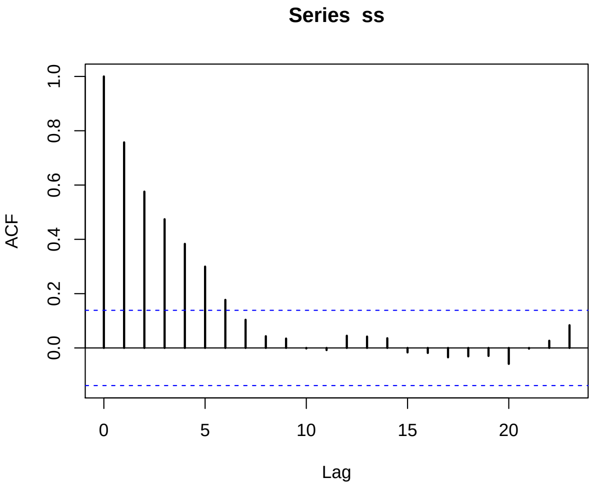
Stationary series

For AR term < 1, Y_t is always pulled back towards the mean.



- These are the most common, and most useful, type of AR series.

Autocorrelation for the stationary series drops off right away.



- The past matters, but with limited horizon.

Mean reversion

An important property of stationary series is mean reversion.

Think about shifting both Y_t and Y_{t-1} by their mean μ.

$$Y_t - \mu = \beta_1(Y_{t-1} - \mu) + \varepsilon_t$$

Since $|\beta_1| < 1$, Y_t is expected to be closer to μ than Y_{t-1}.

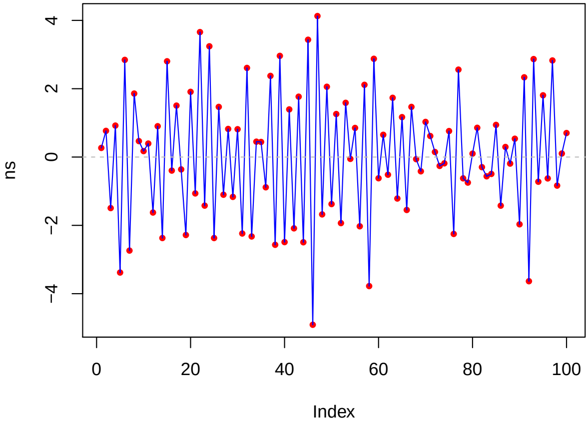
Mean reversion is all over, and helps predict future behaviour:

- weekly sales numbers,
- daily temperature.

Negative correlation

It is also possible to have negatively correlated AR(1) series.

$\beta_1 = -0.8$



- But you see these far less often in practice.

Summary of AR(1) behavior

$|\beta_1| < 1$: The series has a mean level to which it reverts. For positive β_1, the series tends to wander above or below the mean level for a while. For negative β_1, the series tends to flip back and forth around the mean. The series is stationary, meaning that the mean level does not change over time.

$|\beta_1| = 1$: A random walk series. The series has no mean level and, thus, is called nonstationary. The drift parameter β_0 is the direction in which the series wanders.

$|\beta_1| > 1$: The series explodes, is nonstationary, and pretty much useless.

AR(p) models

It is possible to expand the AR idea to higher lags

$$AR(p) : Y_t = \beta_0 + \beta_1 Y_{t-1} + \cdots + \beta_p Y_{t-p} + \varepsilon.$$

For example, a 12 month lag for the beer data:

```
> beerreg12 <- lm( beer$prod[12:72] ~  beer$prod[1:61])
> summary(beerreg12) ## abbreviated output

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)     10.25404    3.69629   2.774   0.0074 **
beer$prod[1:61]  0.70093    0.09102   7.701 1.75e-10 ***
```

AR(p) models

It is possible to expand the AR idea to higher lags

$$AR(p) : Y_t = \beta_0 + \beta_1 Y_{t-1} + \cdots + \beta_p Y_{t-p} + \varepsilon.$$

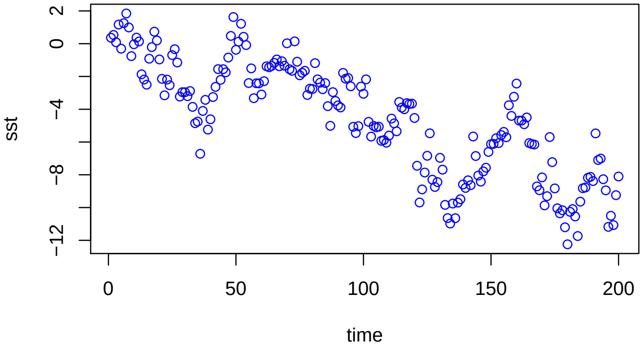
However, it is seldom necessary to fit AR lags for $p > 1$.

- Like having polynomial terms higher than 2, this just isn't usually required in practice.
- You lose all of the stationary/nonstationary intuition.
- Often, the need for higher lags is symptomatic of (missing) a more persistent trend or periodicity in the data ...

Trending series

Often, you'll have a linear trend in your time series.

$\Rightarrow$ AR structure, sloping up or down in time.



This is easy to deal with: just put "time" in the model.

AR with linear trend:

$$Y_t = \beta_0 + \beta_1 Y_{t-1} + \beta_2 t + \varepsilon_t$$

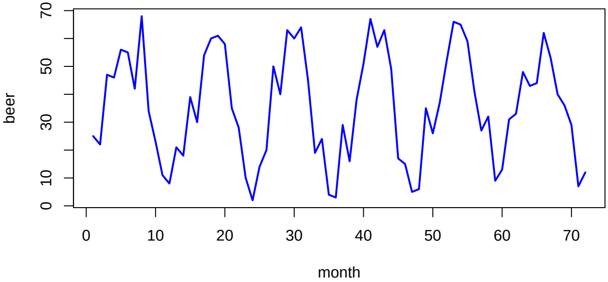
```
> t <- 1:199
> sst.fit <- lm(sst[2:200] ~ sst[1:199] + t)
> summary(sst.fit)  ## abbreviated output

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.571525   0.178110  -3.209  0.00156 **
sst[1:199]   0.735840   0.048062  15.310  < 2e-16 ***
t           -0.009179   0.002160  -4.249 3.32e-05 ***
```

Periodic models

It is very common to see seasonality or periodicity in series.

- Temperature goes up in Summer and down in Winter.
- Natural gas consumption in London or Chicago would do the opposite.

Recall the monthly beer production data:



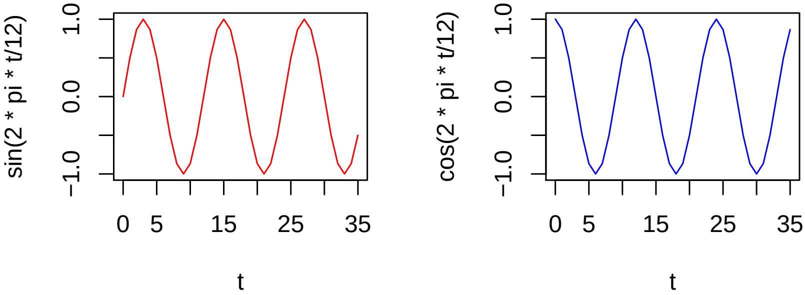
- Appears to oscillate on a 12-month cycle.

The straightforward solution: Add periodic predictors.

$\mathrm{Period}-k$ model:

$$Y_t = \beta_0 + \beta_1 \sin(2\pi t/k) + \beta_2 \cos(2\pi t/k) + \varepsilon_t$$

Remember your sine and cosine!



- Repeating themselves every 2π.

Period$-k$ model:

$$Y_t = \beta_0 + \beta_1 \sin(2\pi t/k) + \beta_2 \cos(2\pi t/k) + \varepsilon_t$$

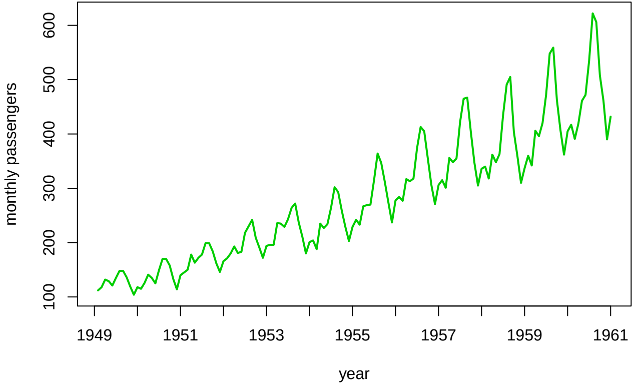
It turns out that you can represent any smooth periodic function as a sum of sines and cosines.

You choose k to be the number of "times" in a single period.

- For monthly data, $k = 12$ implies an annual cycle.
- For quarterly data, usually $k = 4$.
- For hourly data, $k = 24$ gives you a daily cycle.

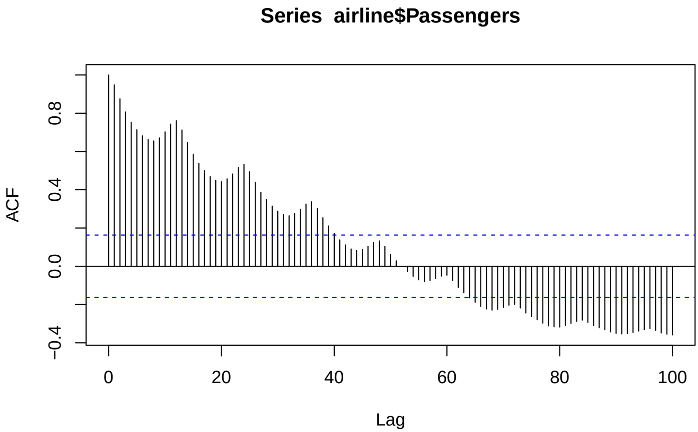
Putting it all together: Airline data

- $Y_t =$ monthly total international passengers, 1949-1960.



- What do you notice in the data?

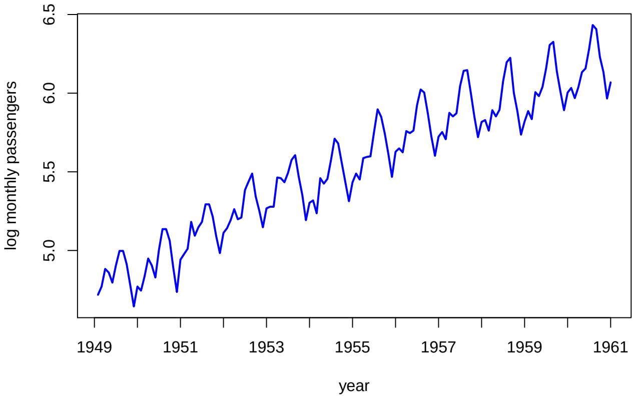
The data shows a strong persistence in correlation.



Annual (12 month) periodicity shows up here as well.

Fitting the model: first, don't forget your fundamentals!

- The series variance is increasing in time.
- We need to work on log/sqrt scale!



The series shows a linear trend, an oscillation of period 12, and we expect to find autoregressive errors.

$$\log(Y_t) = \beta_0 + \beta_1 \log(Y_{t-1}) + \beta_2 t + \beta_3 \sin\left(\frac{2\pi t}{12}\right) + \beta_4 \cos\left(\frac{2\pi t}{12}\right) + \varepsilon_t$$

Open the `VB_TS_exercise.Rmd` file, and start fitting some TS models.