[⋆] Big Data: Spatiotemporal Data

Matt Taddy, University of Chicago Booth School of Business

faculty.chicagobooth.edu/matt.taddy/teaching

Space-Time data is everpresent.

In this class we've focused on
independent and identically distributed (iid) data.
We haven't talked much about space-time dependence.

The first-order trick with such data is de-trending:
Use 'standard' methods to get data that should be iid.

- Using returns rather than prices.
- 'seasonally adjusted': financial data is often pre-cleaned.
- Include time + space variables in your regression.

This class has made the 'include time variables' step trivial,
since we just throw everything in and let CV-lasso sort it out.

Spatiotemporal Trends

The data often include time t (year, day, second)
or space $\mathbf{s} = [s_1, s_2]$ (latitude, longitude).

If you have t or $\mathbf{s}$, just put them in the regression.
e.g., california housing data in trees lecture.

You can also add quadratic versions:
When regressing crime on abortion and controls (09),
we included effects for time and time^2: $t\beta_t + t^2\beta_{t2}$
and interaction of these trends with all controls so that the effect of, say, beer consumption was $\beta_{\text{beer}} + \beta_{\text{beer},t}t + \beta_{\text{beer},t2}t^2$.

Then use CV-lasso to only keep what matters.
Alternatively, if you have lots of data, just do `glm`.

Seasonality

In temporal data, we often need to account for seasonality:

cyclical month/quarter/day/etc effects.

For example, maybe sales are always higher in spring/summer, or beginning of quarter months are leaner than end of quarter.

Again, the solution is: put them in the model!

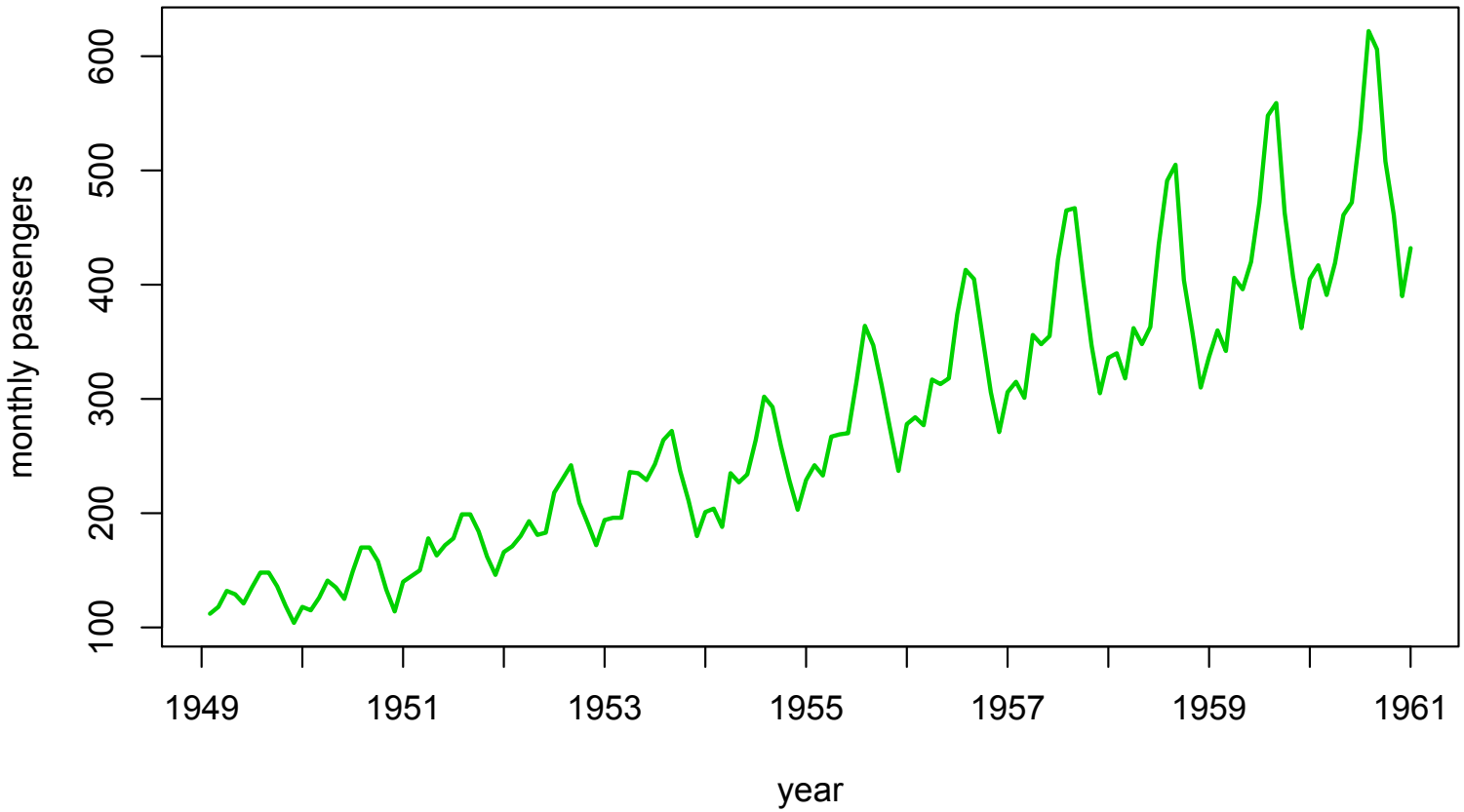
This time, you include seasonal indicators as dummy variables. For example,

```
month = factor(month) # levels "jan","feb",...
x = model.matrix(~.+month, data=D) # or .*month
```

Something like this would allow for 'month effects'.

Example: airline data

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

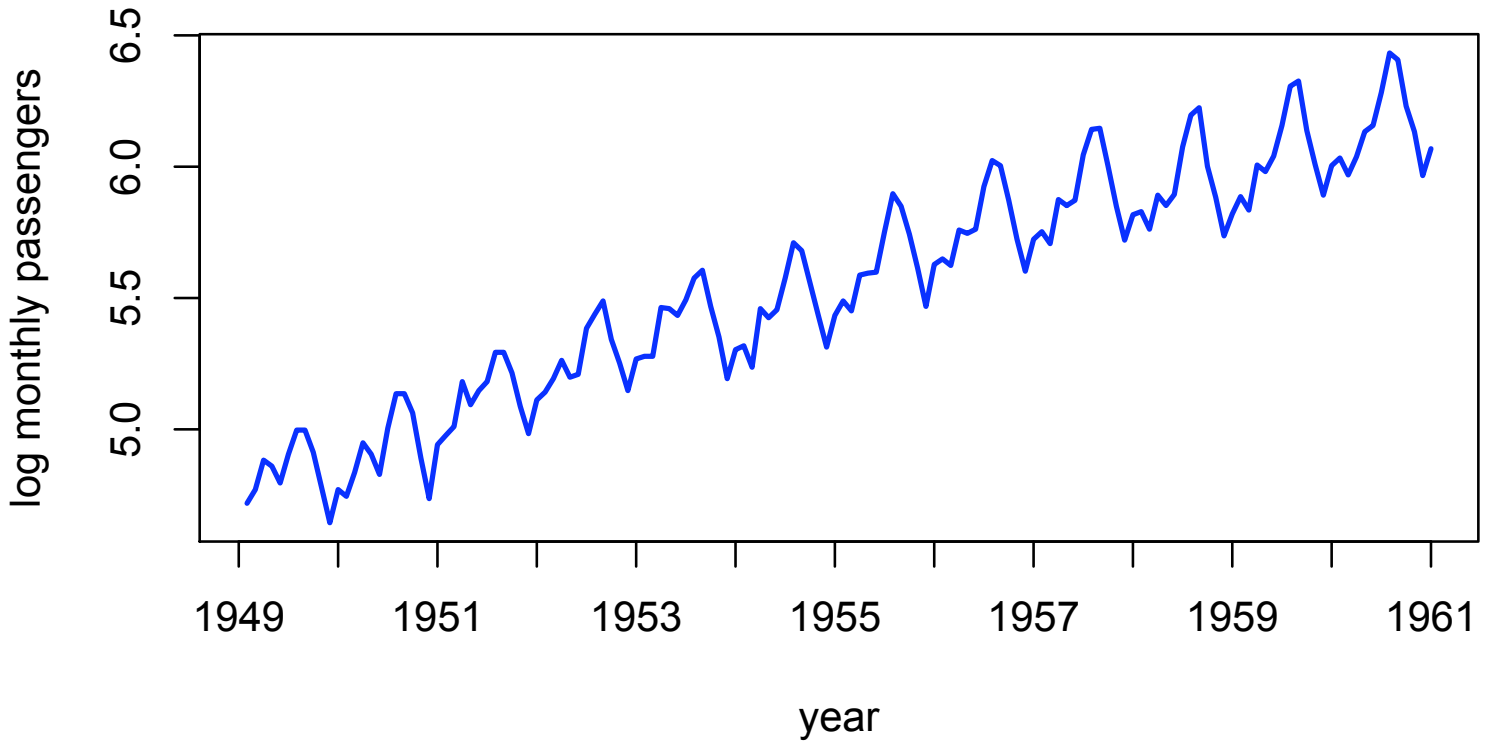


We see an increasing annual oscillation and positive linear trend.

Air Travel

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

- The series variance is increasing in time.
- Passenger numbers are like sales volume.
- We should be working on log scale!



Air Travel

The series shows a linear trend and an oscillation of period 12 (i.e., it looks like we need month effects β_{m_t})

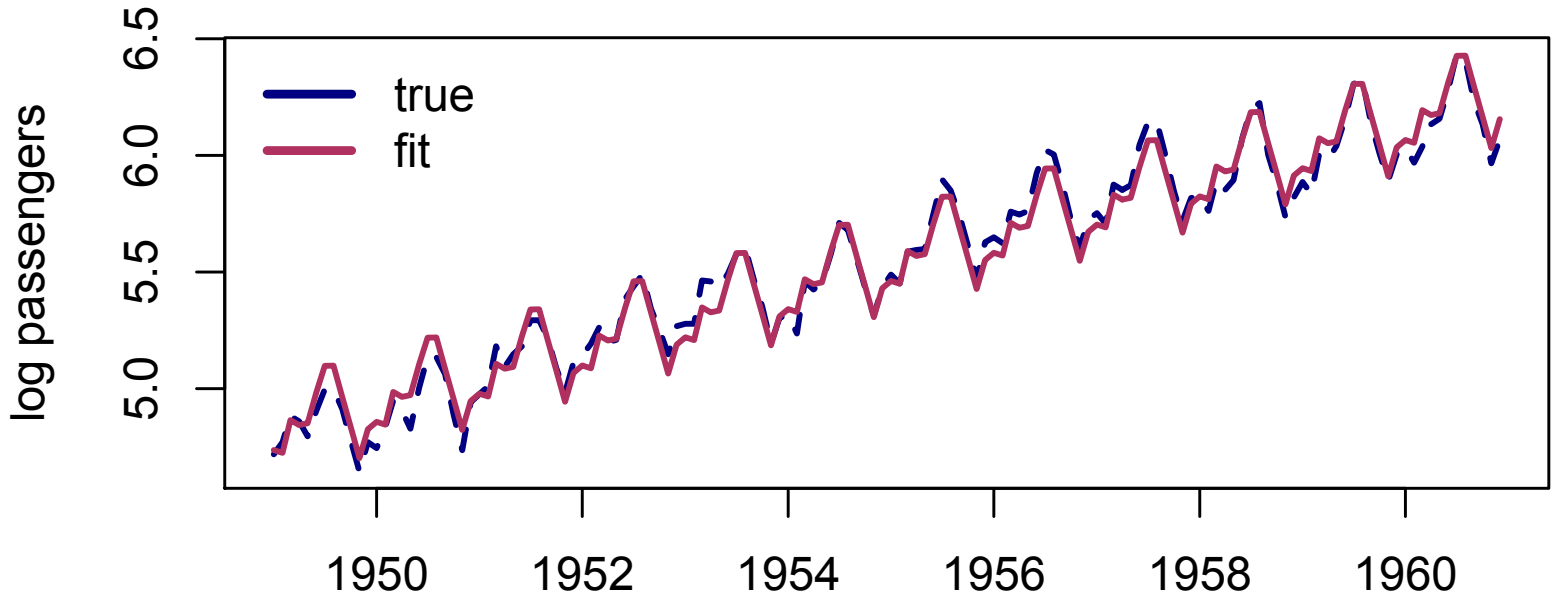
$$\log(y_t) = \alpha + \beta_t t + \beta_{m_t} + \varepsilon_t$$

```
month <- factor(airline$Month)
time <- (year-min(year))*12 + airline$Month
summary(air <- glm(log(passengers) ~ time + month))
```

```
>              Estimate Std. Error t value Pr(>|t|)
> time         0.0100688  0.0001193  84.399  < 2e-16 ***
> month2      -0.0220548  0.0242109  -0.911  0.36400
> month3       0.1081723  0.0242118   4.468 1.69e-05 ***
...
```

Airline Travel: model fit

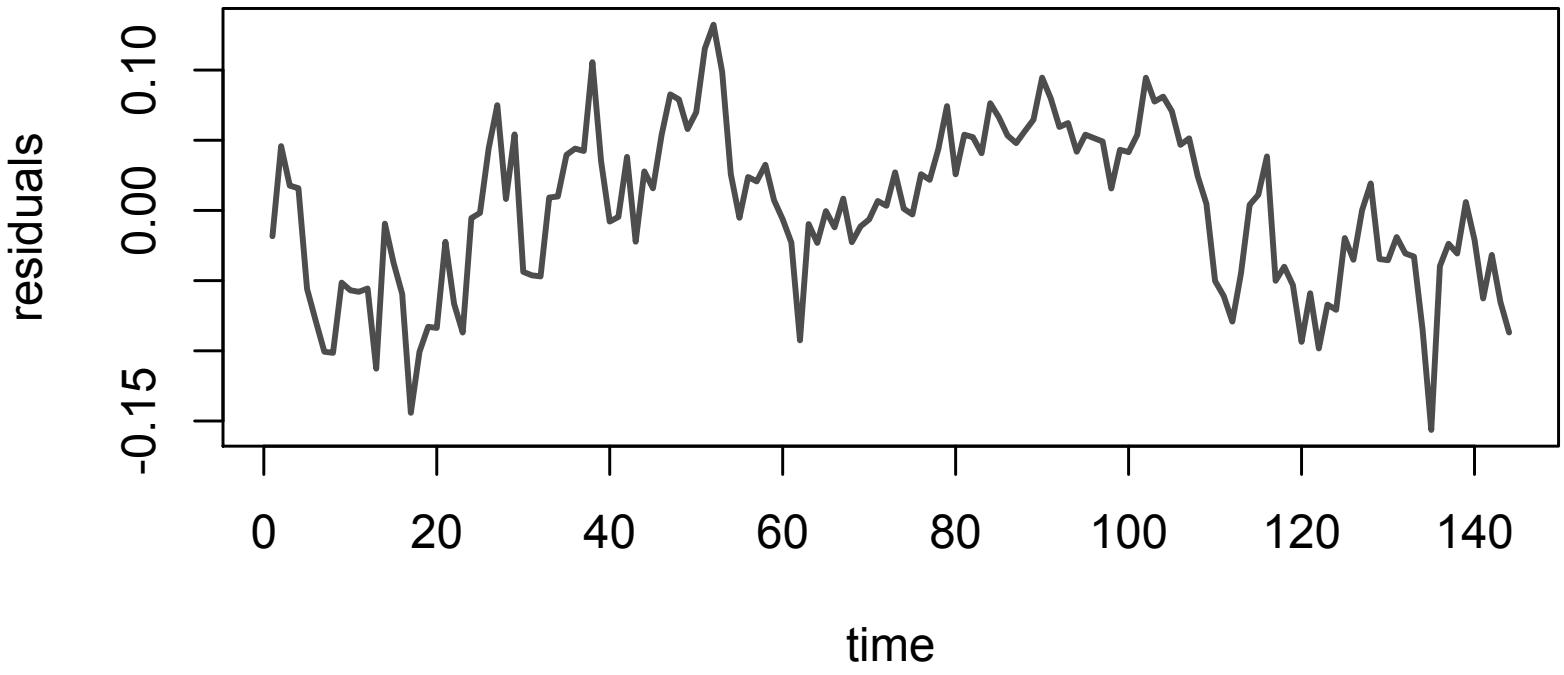
The model predictions look pretty good!



The month and time effects seem to capture the trends.

However, if we look close...

Airline Travel: autocorrelation



Residuals appear correlated in time!

This violates our basic iid assumption: $\varepsilon_i \perp\!\!\!\perp \varepsilon_j$.

This is called autocorrelation: correlation with yourself.

Time Series Data and Dependence

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

- Annual GDP.
- Quarterly production levels
- Weekly sales.
- Daily temperature.
- 5 minute Stock returns.

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

Time Series Data and Dependence

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?

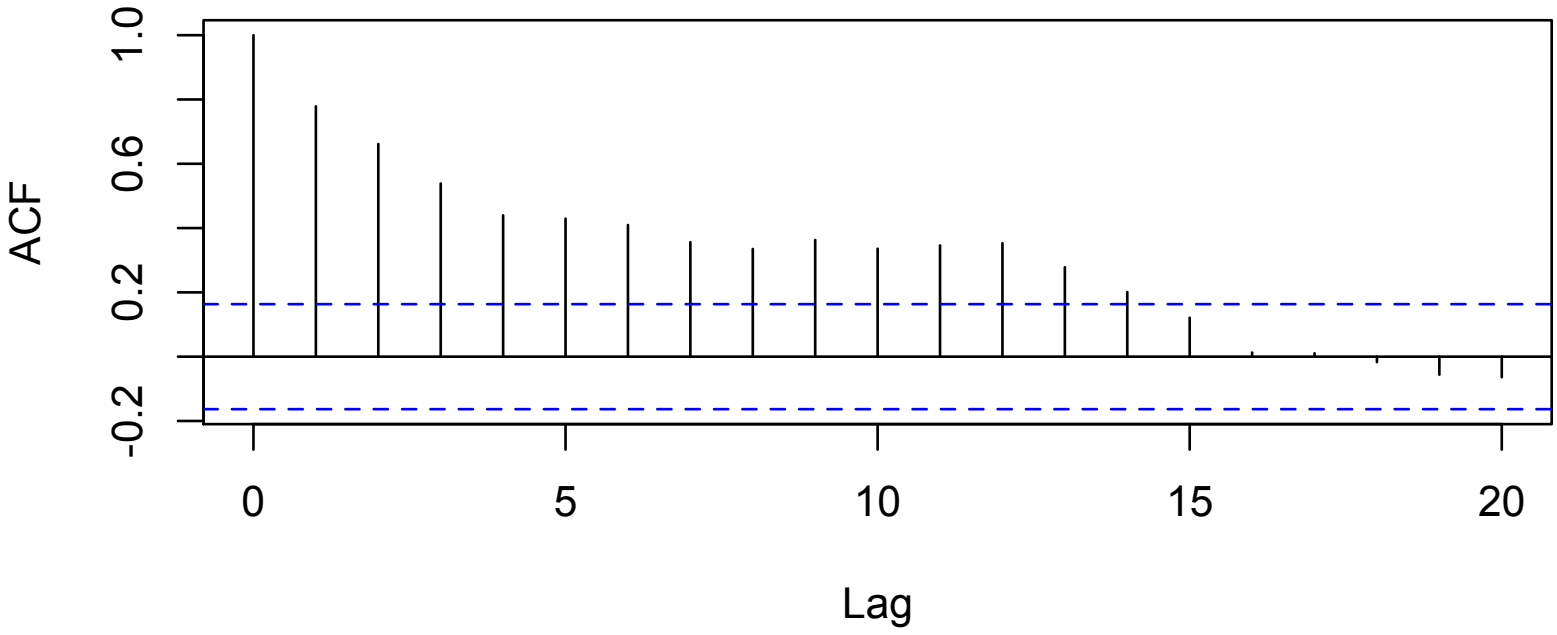
How would this change if the readings were iid?

Correlated errors require fundamentally different techniques.

The autocorrelation function

Summarize dependence with lag-l correlations:

The autocorrelation function (ACF) is $r(l) = \mathrm{cor}(\varepsilon_t, \varepsilon_{t-l})$




```
> print(acf(air$resid))
Autocorrelations of series air$resid, by lag
     0      1      2      3      4      5      6
 1.000  0.779  0.662  0.539  0.440  0.429  0.410
```

Autoregression

How do we model this type of data?

Suppose $y_1 = \varepsilon_1$, $y_2 = \varepsilon_1 + \varepsilon_2$, $y_3 = \varepsilon_1 + \varepsilon_2 + \varepsilon_3 + ...$,

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}.

Autoregression

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

The autoregressive model of order one holds

$$AR(1): y_t = \beta_0 + \beta_1 y_{t-1} + \varepsilon$$

This is just an SLR model of y_t regressed onto lagged y_{t-1}, and 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.

Autoregression

$$AR(1) : y_t = \beta_0 + \beta_1 y_{t-1} + \varepsilon$$

Again, y_t depends on the past only through y_{t-1}.

AR(1) means that previous lag values $(y_{t-2}, y_{t-3}, \ldots)$ do not help predict y_t if you already know y_{t-1}.

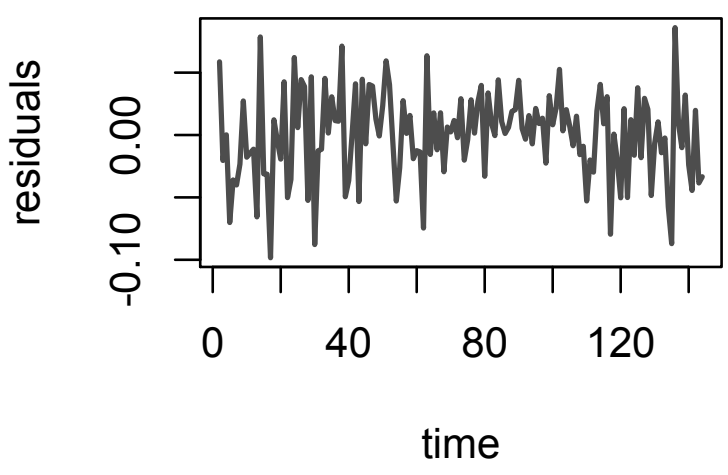
Think about daily temperatures:

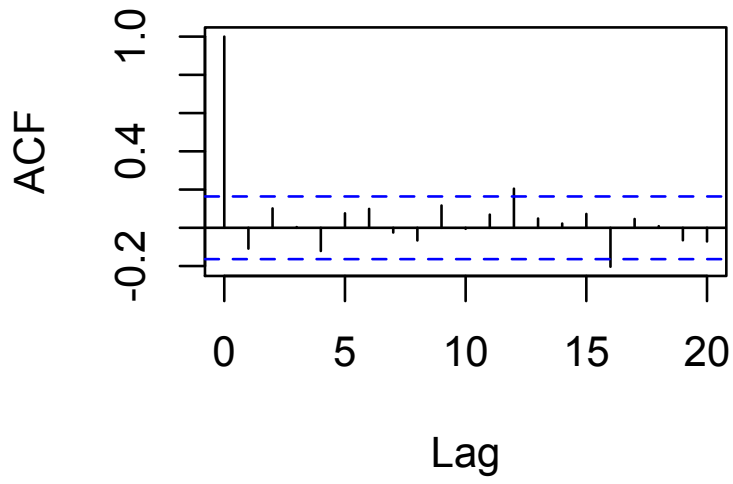
- If I want to guess tomorrow's temperature (without knowing the forecast!), I base my prediction on today's temperature and will probably ignore yesterday's temperature.

In this model β_1 is called the AR term

Fitting the AR model

```
> lag <- head(log(passengers),-1)## see help(head)
> passengers <- passengers[-1]
> month <- month[-1]
> time <- time[-1]
> summary(airAR <- glm(log(passengers) ~ time + month + lag))
...
lag           0.7930716  0.0548993  14.446  < 2e-16 ***
```





Now the residuals look nice and independent.

AR(1) Autoregression

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

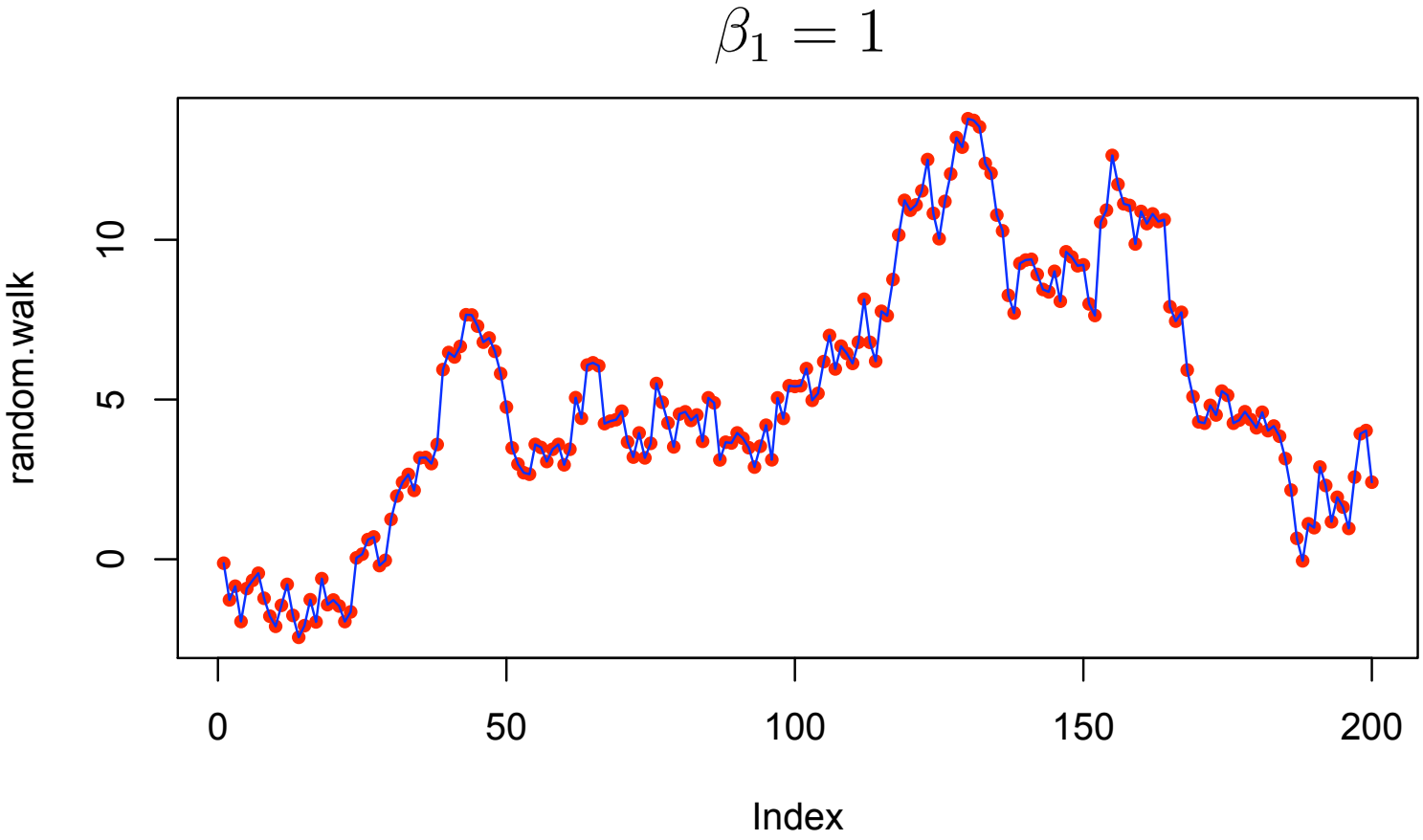
$$AR(1) : Y_t = \beta_0 + \beta_1 Y_{t-1} + \varepsilon$$

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.

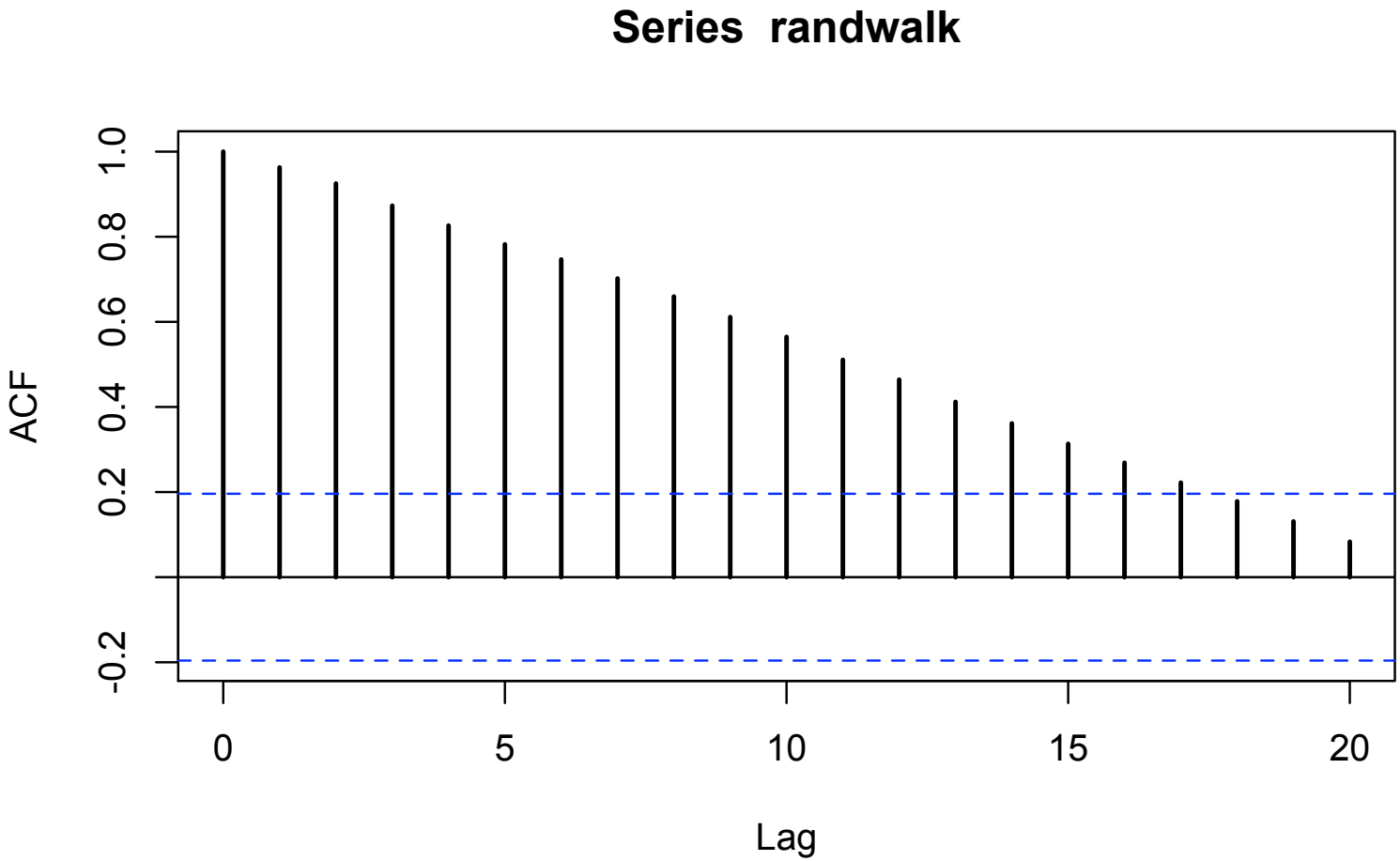
Random Walk

In a random walk, the series just wanders around.



Random Walk

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



Random Walk

The random walk has some special properties...

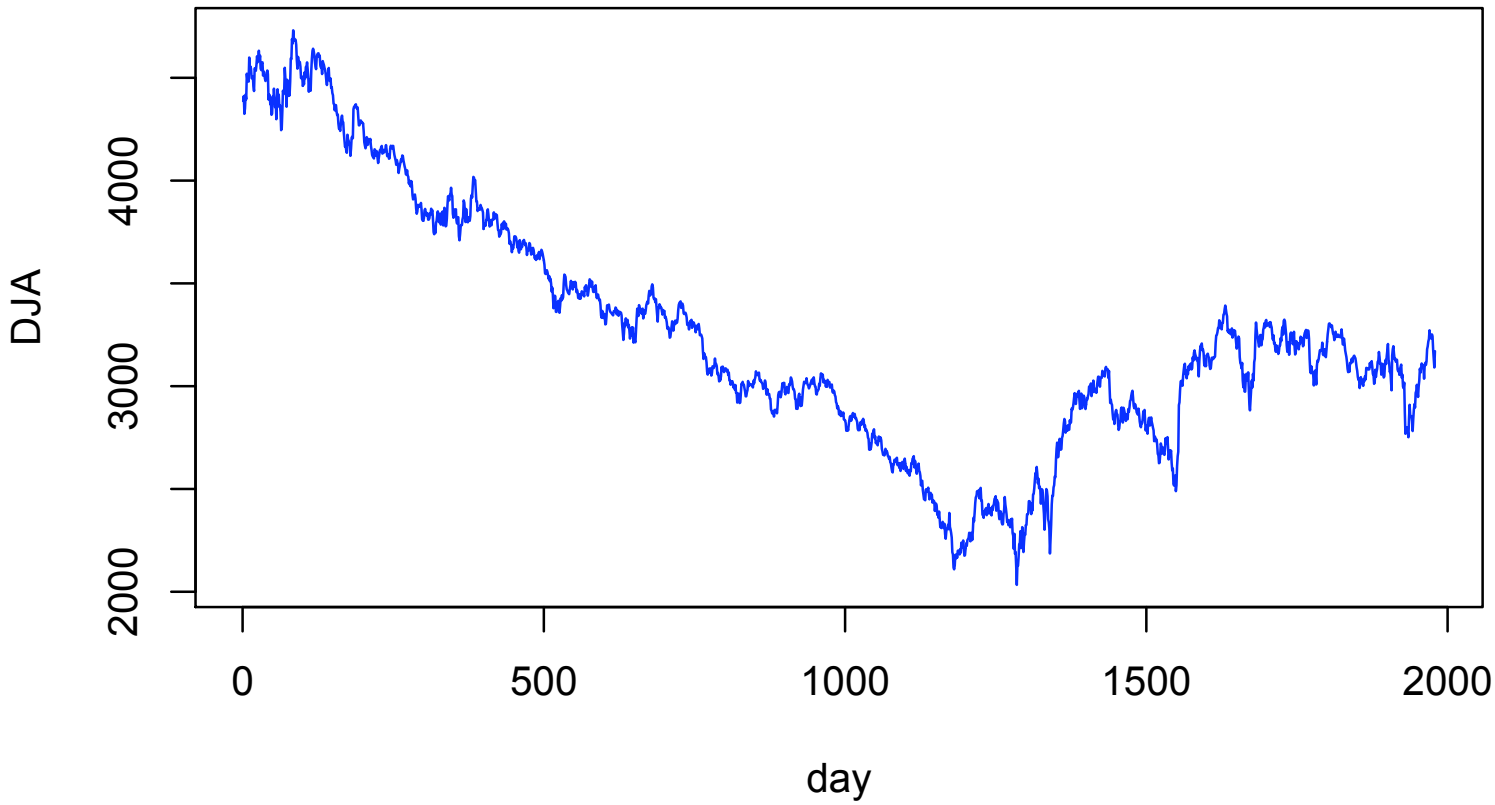
$Y_t - Y_{t-1} = \beta_0 + \varepsilon$, and β_0 is called the "drift parameter".

The series is nonstationary: it has no average level that it wants to be near, but rather just wanders off into space.

Since $\mathbb{E}[Y_t] = \mathbb{E}[Y_{t-1}]$ (e.g., tomorrow $\approx$ today), the random walk without drift is a common model for simple processes.

Random Walk Example: Dow Jones

For example, consider the monthly Dow Jones composite index from 2000 - 2007



DJA appears as though it is just wandering around.

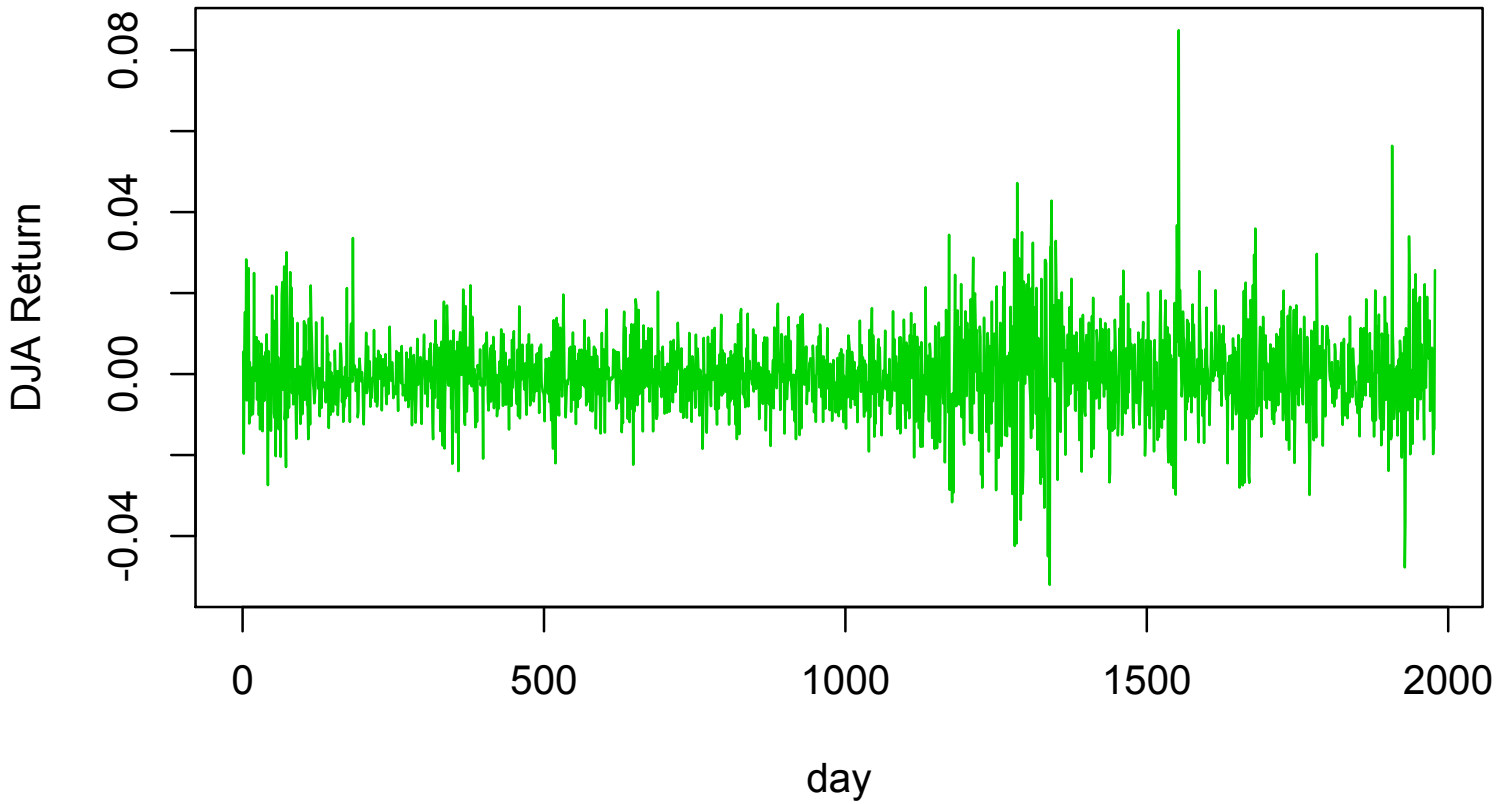
Random Walk Example: Dow Jones

Sure enough, the regression fit looks like a random walk ($b_1 \approx 1$).

```
> summary(ARdj <- glm(dja[2:n] ~ dja[1:(n-1)]))

...

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     7.05419    4.00385   1.762   0.0782 .
dja[1:(n - 1)]  0.99764    0.00121 824.298   <2e-16 ***
```

Random Walk Example: Dow Jones

When you switch to returns, however, it's just white noise.



$(Y_t - Y_{t-1})/Y_{t-1}$ appears to remove the dependence.

Random Walk Example: Dow Jones

And now the AR term is not significant.

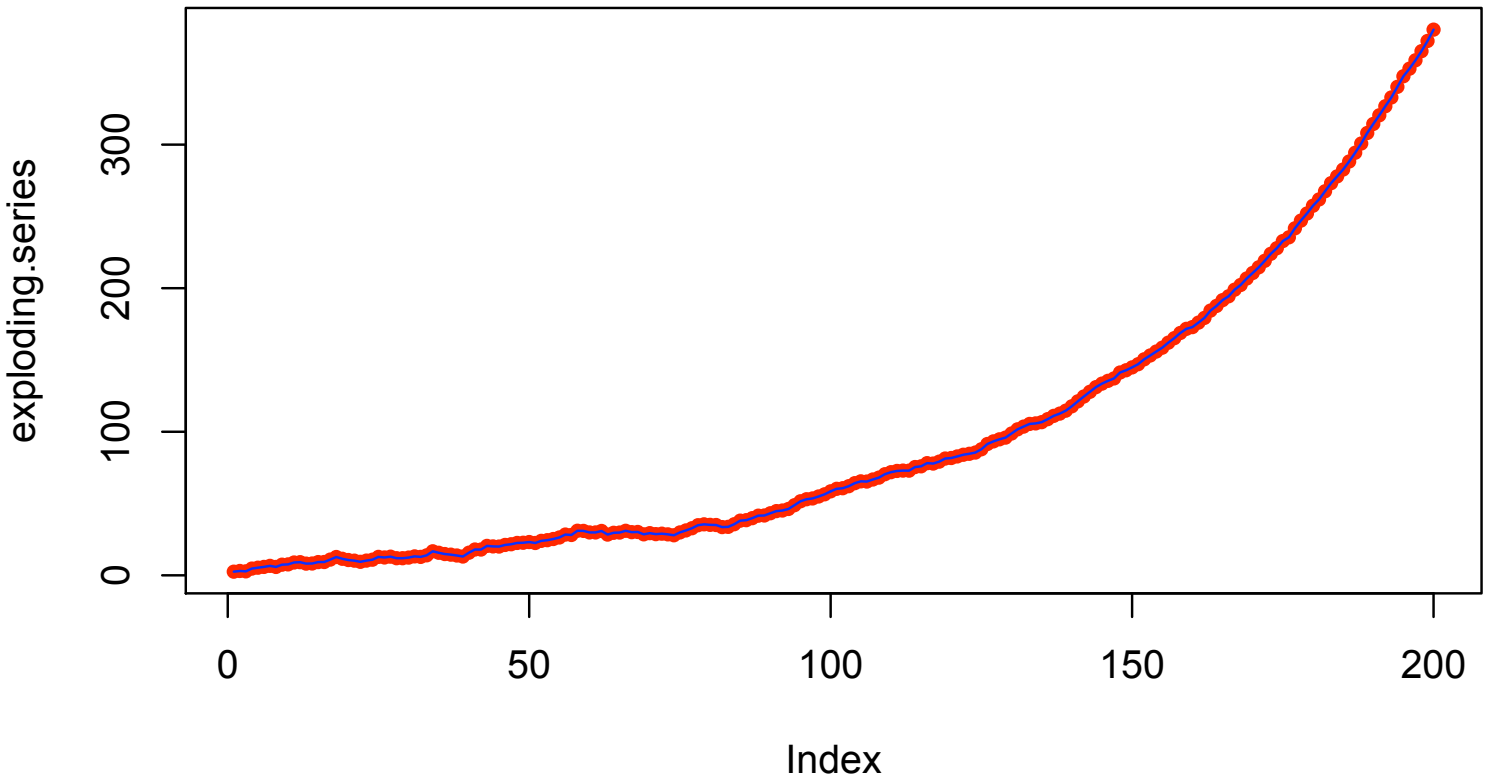
```
> returns <- (dja[2:n]-dja[1:(n-1)])/dja[1:(n-1)]
> summary( glm(returns[2:n] ~ returns[1:(n-1)]) )

...

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)        -0.0001138  0.0002363  -0.482    0.630
returns[1:(n - 1)] -0.0144411  0.0225321  -0.641    0.522
```

This is common with random walks: difference $Y_t - Y_{t-1}$ is iid.

Exploding Series

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

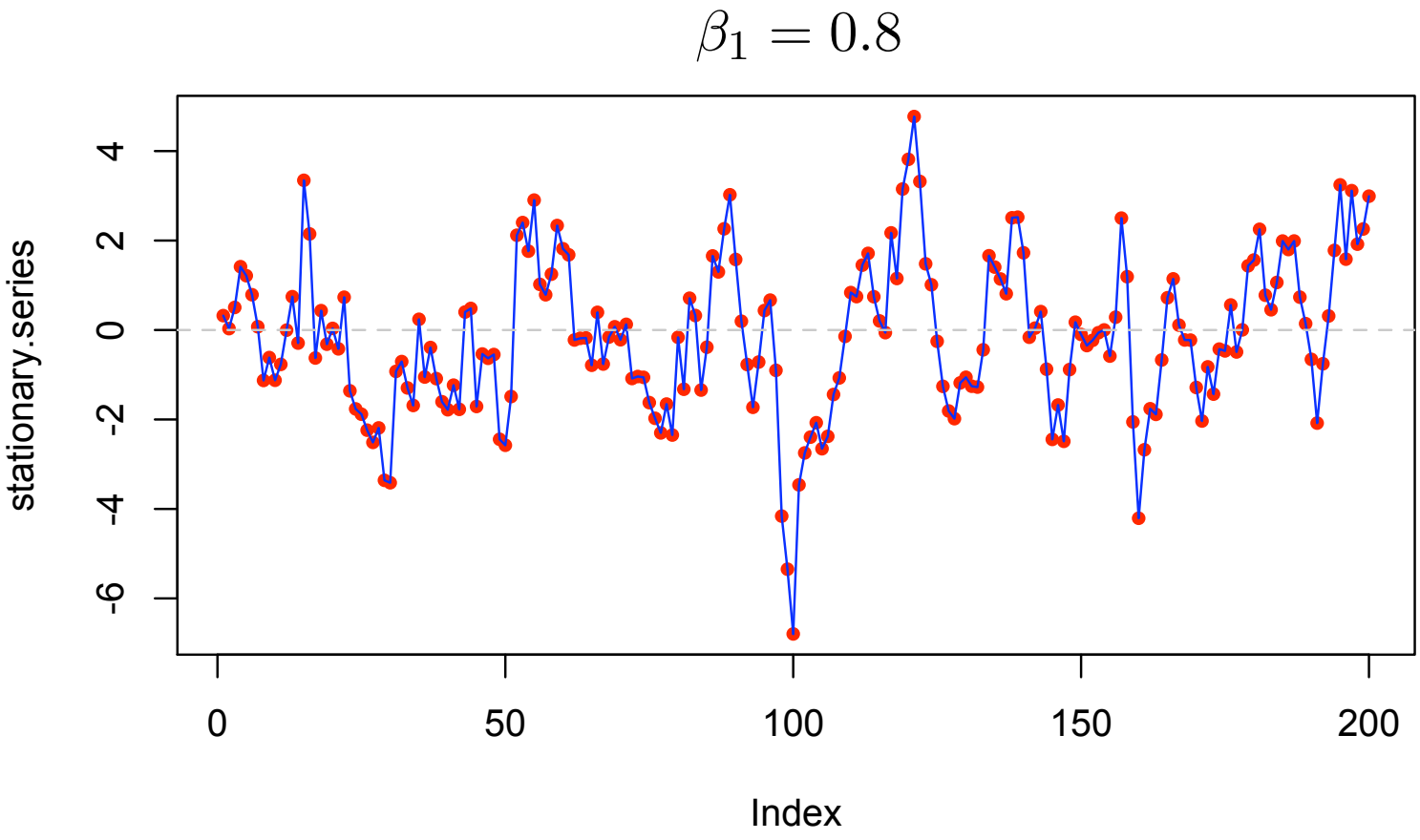
$$\beta_1 = 1.02$$



Since the series explodes, it is useless for modeling and prediction.

Stationary Series

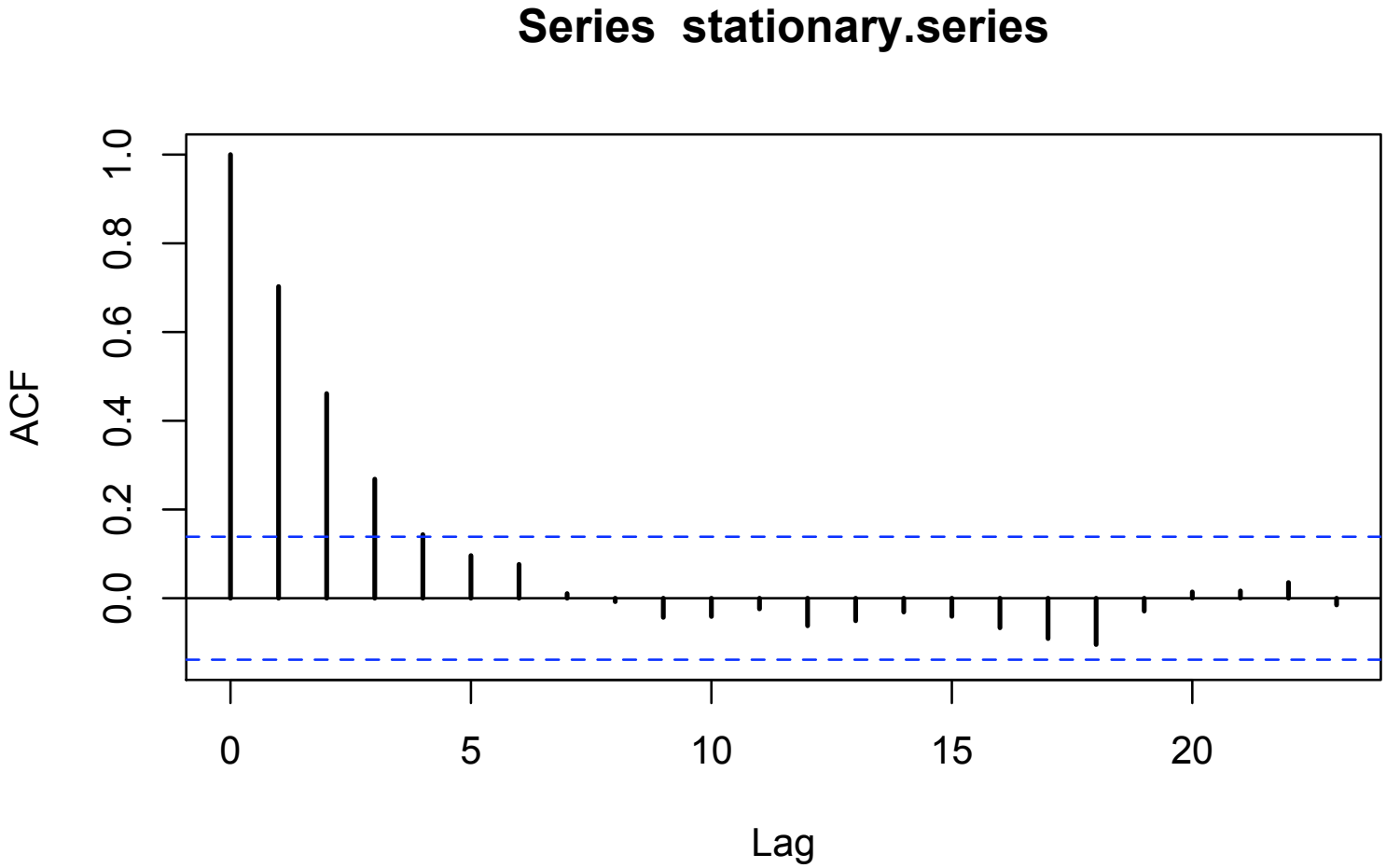
For AR term < 1, Y_t is always pulled back towards the mean.



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

Stationary Series

Autocorrelation for the stationary series drops off right away.



The past matters, but with limited horizon.

Mean Reversion

An important properties 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 the μ than Y_{t-1}.

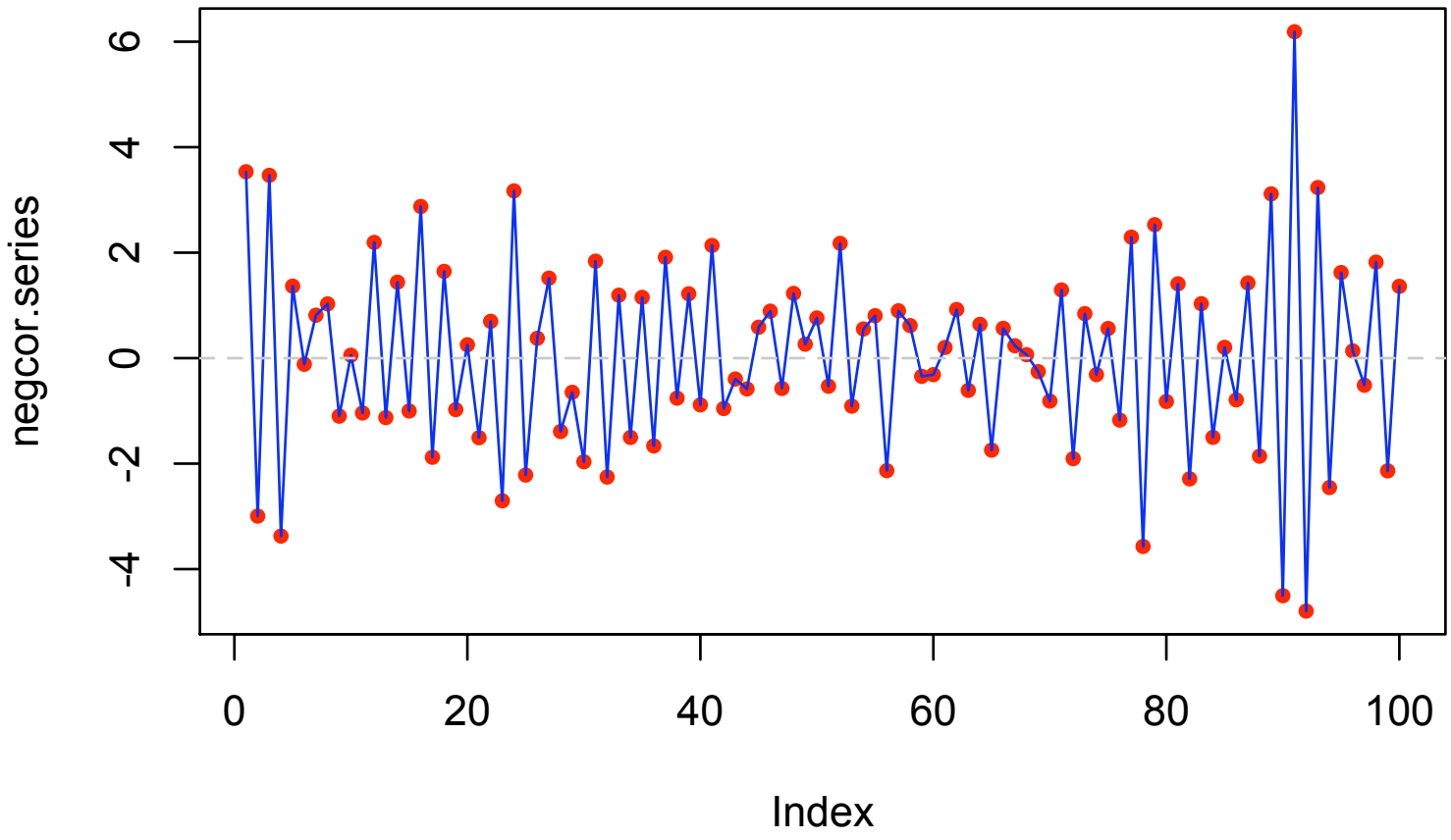
Mean reversion is all over, and helps predict future behaviour.

- "alpha" in repeated CAPM models.
- Weekly sales numbers.
- Daily temperature.

Negative Correlation

It is also possible to have negatively correlated AR(1) series, but you see these far less often in practice.

$$\beta_1 = -0.8$$



Summary of AR(1) Behavior

$\lvert\beta_1\rvert < 1$	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.
$\lvert\beta_1\rvert = 1$	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.
$\lvert\beta_1\rvert > 1$	The series explodes, is nonstationary, and pretty much useless.

AR(p) Models

It is possible to expand the AR idea to higher lags

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

Drawback: you lose the stationary/nonstationary intuition.

And often the need for higher lags is symptomatic of your missing a more persistent trend or periodicity in the data.

Previous classes probably warned against using $p > 1$.

However: with lasso at your disposal, I'm OK with you just throwing in higher lags and letting it choose what you need. Datamine the lags!

Spatial extensions

Space is just like time, but with another dimensions!

To include dependence, you again include $y_{\mathbf{s}}$ for other $\mathbf{s}$.

For example, in your regression you can include

- the average for neighboring states.
- the average of the neighboring pixels.

There are a ton of more compicated routines for when modeling temporal or spatial effects is your primary interest.
gaussian processes, kriging, state-space models, GARCH, etc...
Another time in another class!