# Time series

Aniket Majumdar

galvanize

Objectives Morning

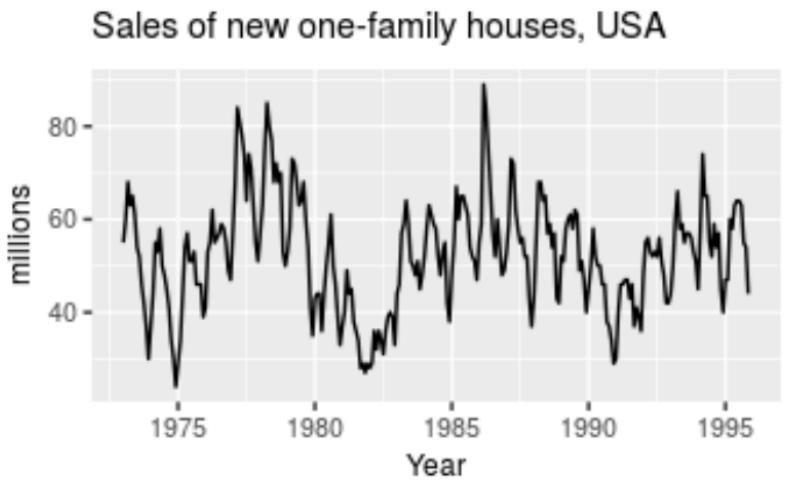
- Define key time series concepts
- Trend, seasonality & cycles
- Additive decomposition
- Exponential smoothing

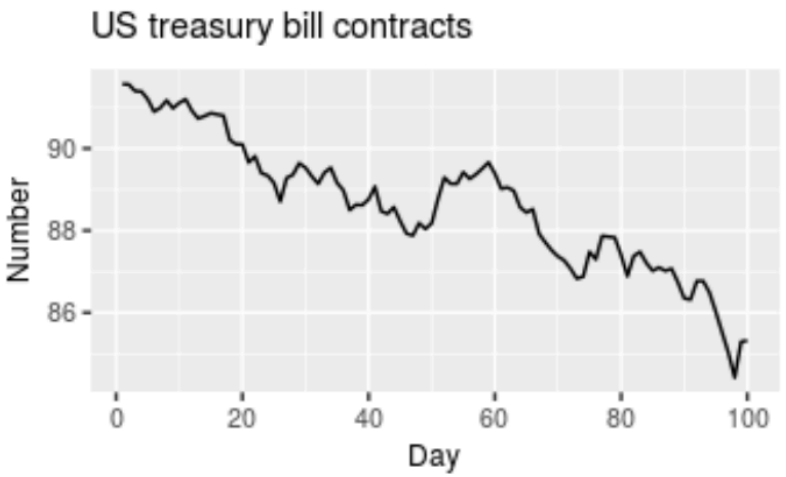
Examples:

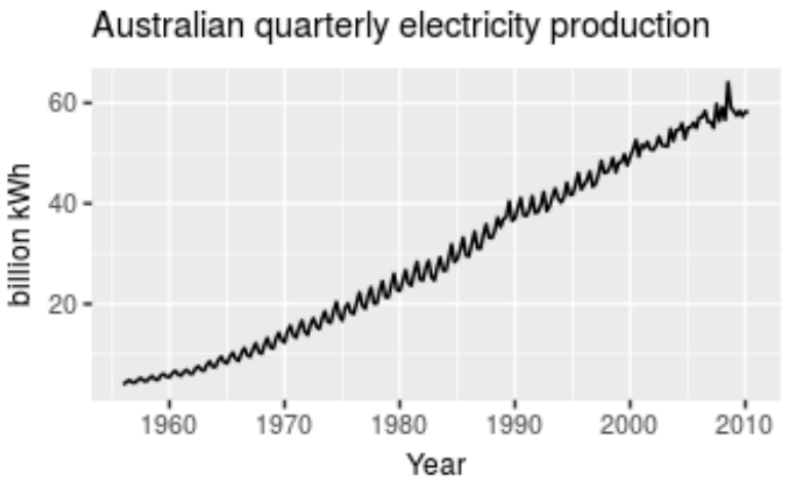
- GDP of a nation
- Price of a product or a stock
- Demand for a good
- Unemployment
- Web traffic (clicks, logins, posts…)

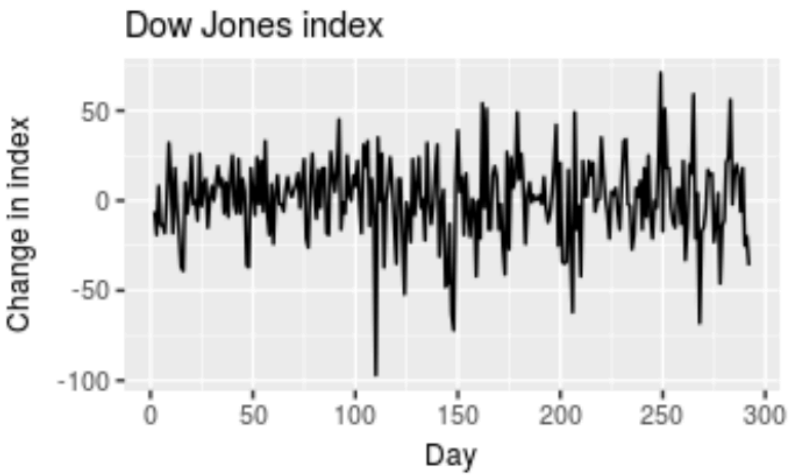
# Time series Visualization

galvanize

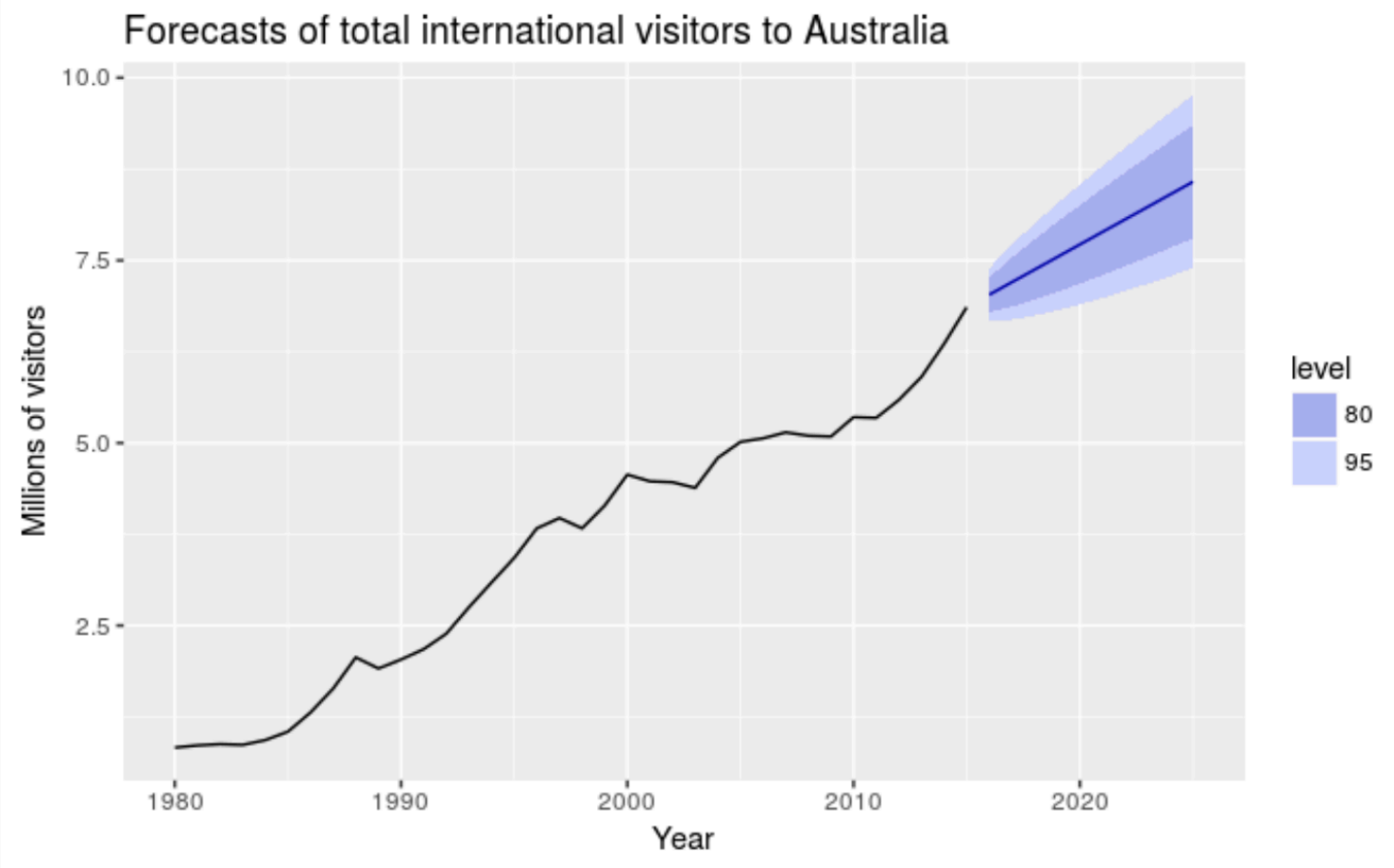


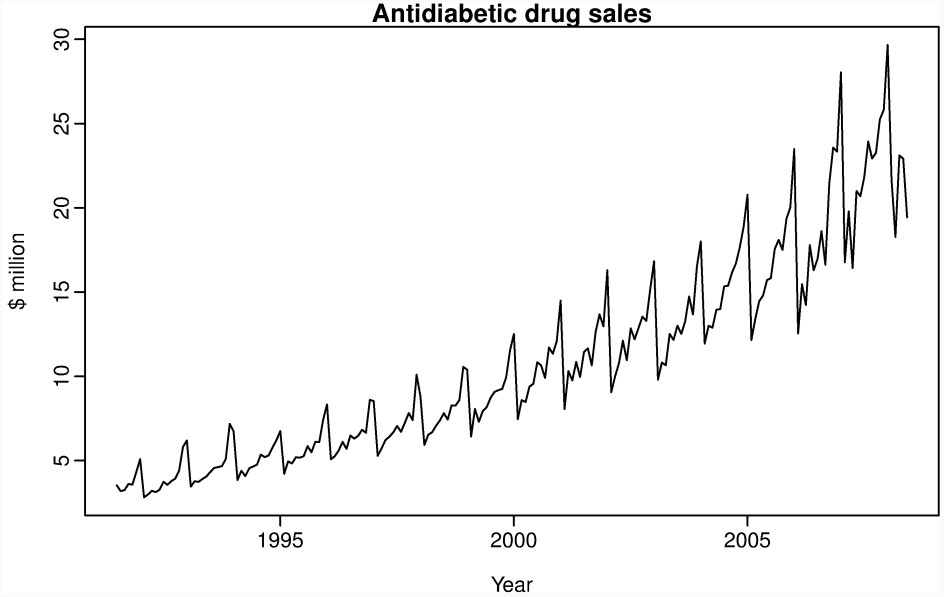



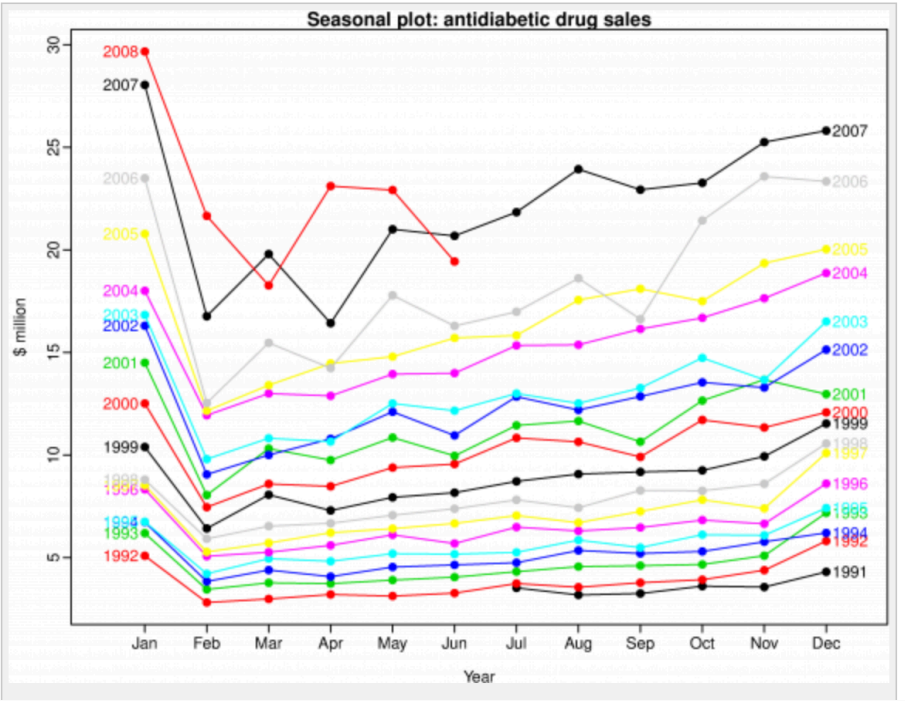
# Visualization — Forecast



# Visualization — Seasonality

galvanize



## Definition

galvanize

We assume a time series $\{y_t\}$ has the following properties:

- $y_t$ is an observation of the level of y at time t
- $\{y_t\}$ denotes the collection of observations/the series
  - $\{y_t\}$ can extend back to 0 or $-\infty$ and forward to some end time T
  - Ex: $t \in \{0, \ldots, T\}$
- Sampling at regular intervals (even if process is continuous)
- Evenly spaced data, no missing observations

## Time series difficulties

galvanize

- The main problem results from the fact that there can not be uncorrelated samples
- There is usually just one observation of the path of the process
- Limited data
- In order to model we must make strong (and often false) assumptions (i.e. about correlation)
- Projections are for areas where no data exists (future)
- For time series forecasting problems, the target and the features are the same metric (e.g. price of a stock in the past and in the future) and we are trying to model implicit (unmeasured) features

# Auto Correlation

galvanize

Just as correlation measures the extent of a linear relationship between two variables, autocorrelation measures the linear relationship between *lagged values* of a time series.

There are several autocorrelation coefficients, corresponding to each panel in the lag plot. For example, $r_1$ measures the relationship between $y_t$ and $y_{t-1}$, $r_2$ measures the relationship between $y_t$ and $y_{t-2}$, and so on.

The value of $r_k$ can be written as

$$r_k = \frac{\sum_{t=k+1}^{T} (y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^{T} (y_t - \bar{y})^2},$$

where $T$ is the length of the time series.

# Time series decomposition

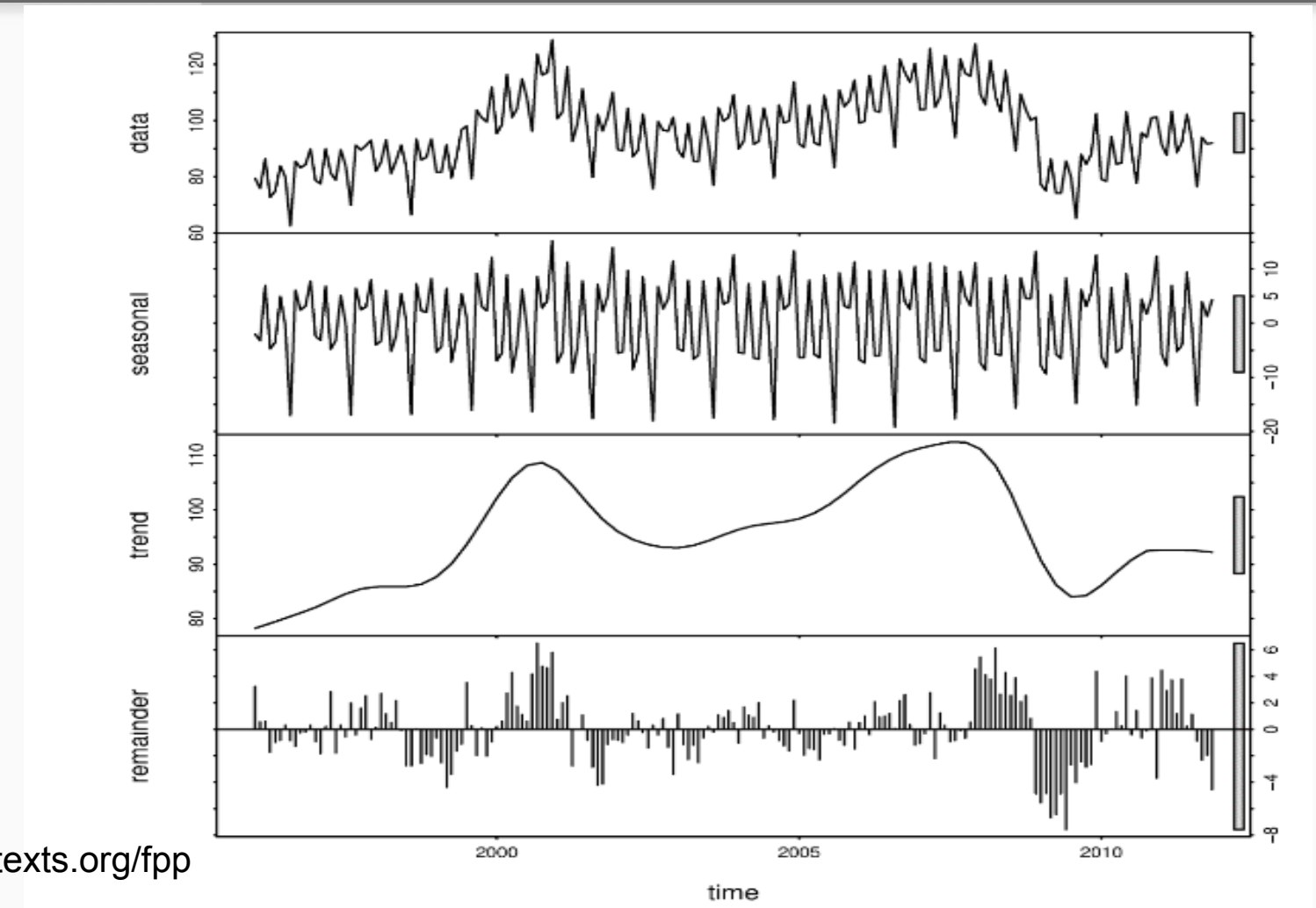
galvanize

Time series data can consist of several components:

1. Trends
    a. not necessarily linear
2. Seasonal
    a. related to calendar, fixed period
3. Periodic/cyclic
    a. not strictly coupled to calendar
    b. variable/unknown period
    c. usually cycles last >= 2 years
    d. Trends are often part of a larger cycle we cannot see if we don't have enough data (i.e. ice age cycles)
4. Irregular

These components can be additive or multiplicative

# Time series decomposition example (Hyndman et al.)




https://www.otexts.org/fpp

## Popular time series models

galvanize

1. Exponential smoothing (also “ETS” — Error, Trend, Seasonality)
    a. Smoothes out irregular shocks to trend and seasonality
    b. Updates forecast with linear combination of current value and past forecast, which has effect of smoothing
    c. Effectively a moving average method with unfixed window size and higher weights for more recent observations

1. ARIMA (afternoon)
    a. Decomposes time series into contributing parts

## Time series regression

galvanize

Model: $y_t = TR_t + \varepsilon_t$ where $\varepsilon_t \sim N(0,\sigma^2)$ (i.i.d.)

Linear trend: $y_t = \beta_0 + \beta_1 t$

Quadratic trend: $y_t = \beta_0 + \beta_1 t + \beta_2 t^2$

Polynomial: $y_t = \beta_0 + \beta_1 t + \ldots + \beta_2 t^p$

Strong assumption being that the parameters don't change over time

Most trends aren't linear, and higher order regression often get out of control outside of the range where we have data. For time series we will always predict for a range where we have no data!

## Adding seasonality

galvanize

Model: $y_t = TR_t + SN_t + \varepsilon_t$ where $\varepsilon_t \sim N(0,\sigma^2)$ (i.i.d.)

Assumption: constant seasonal variation, i.e. aside from the trend, every season of the same type behaves the same way

Create L-1 dummy variables for L seasons (one season is the baseline):

$$SN_t = \beta_{S_1} 1\{S_1\} + \beta_{S_2} 1\{S_2\} + \cdots + \beta_{S_{L-1}} 1\{S_{L-1}\}$$

where

$$1\{S_i\} = \begin{cases} 1 & \text{If } t \text{ is season } i \\ 0 & \text{Otherwise} \end{cases}$$

# General additive decomposition

galvanize

Model: $y_t = TR_t + SN_t + CL_t + IR_t$ (trend + season + cycle + irregularities)

Forecast would be $y_t = TR_t + SN_t + CL_t$ ~~$+ IR_t$~~

With a prediction interval $\hat{y}_t \pm B_{t,\alpha}$

$B_{t,\alpha}$ is the error bound in a (100 - α) % prediction interval

(Interval gets larger further into the future)

Trend and cyclic components are often combined into the trend component

Multiplicative effects become additive with a log transformation:

$$y_t = S_t \times T_t \times E_t \quad \text{is equivalent to} \quad \log y_t = \log S_t + \log T_t + \log E_t$$

## Exponential smoothing

galvanize

1. Simple exponential smoothing: For when the data has no linear trend but mean is changing over time
2. SES is trying to strike a balance between the naive method (predicting the last observed value) and the average method (predicting the average)
3. Holt's trend corrected exponential smoothing: when the series has a linear trend and a slope that changes over time
4. Holt-Winter's method: extension that adds seasonality

# Simple Exponential Smoothing

galvanize

If no trend exists and the mean remains constant:

$y_t = \beta_0 + \varepsilon_t$ where $\varepsilon_t \sim N(0,\sigma^2)$ (i.i.d.)

A simple estimate would be:

$$\hat{y} = b_0 = \frac{1}{n}\sum y_t$$

However, time series have “stickiness”, i.e. even if errors are normal (“white noise”) and every movement is random, each value $y_t$ depends on the previous value $y_{t-1}$ simply because we have a new starting point every time, and successive times are correlated. This process is called a “random walk” and can result in significant drifts over time.

# Simple Exponential Smoothing

galvanize

A simple estimate would be:

$$\hat{y} = b_0 = \frac{1}{n}\sum y_t$$

In light of the “stickiness” of most time series, maybe we should give more weight to recent observations and less to very old ones?

This is what exponential smoothing does, it adjusts the forecast with the more recent observation, over time “forgetting” very old observations.

# Simple Exponential Smoothing

galvanize

1. Initialize an estimate for t=0, simply setting the prediction to the actual observation or averaging over a subset of observations close to that point

$$l_0 = \frac{1}{n_S}\sum_{t=1}^{n_S} y_t$$

2. Compute updated estimates iteratively with a linear combination of actual value and predicted value of the last time point

$$l_t = \alpha y_t + (1-\alpha) l_{t-1}$$

3. This is equivalent to:

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha) y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots$$ hence "exponential"

4. With $l_{t-1}$ being the forecast for $y_t$, $0<\alpha<1$ being a hyperparameter called the "smoothing factor" found by optimization of the sum of squared errors (SSE)

# Simple Exponential Smoothing

galvanize

$$\begin{aligned} l_t &= \alpha y_t + (1-\alpha) l_{t-1} \\ &= \alpha y_t + l_{t-1} - \alpha l_{t-1} \\ &= l_{t-1} + \alpha(y_t - l_{t-1}) \end{aligned}$$

$$\text{SSE} = \sum_{t=1}^{T} (y_t - \hat{y}_{t|t-1})^2 = \sum_{t=1}^{T} e_t^2.$$

So the forecast at time t is simply the forecast at time t-1 plus a fraction of the forecast error for y at time t based on time t-1

We can now forecast a future value $y_{t+\tau}$ with our estimates becoming less reliable the larger τ is (wider prediction interval)

In the absence of trend and seasonality we simply predict the last smoothed value for all future times

The prediction interval is given by:

$$l_T \pm z_{\alpha/2} s \sqrt{1 + (\tau - 1)\alpha^2} \qquad \textit{where} \qquad s = \sqrt{\frac{SSE}{T-1}}$$

# Simple Exponential Smoothing

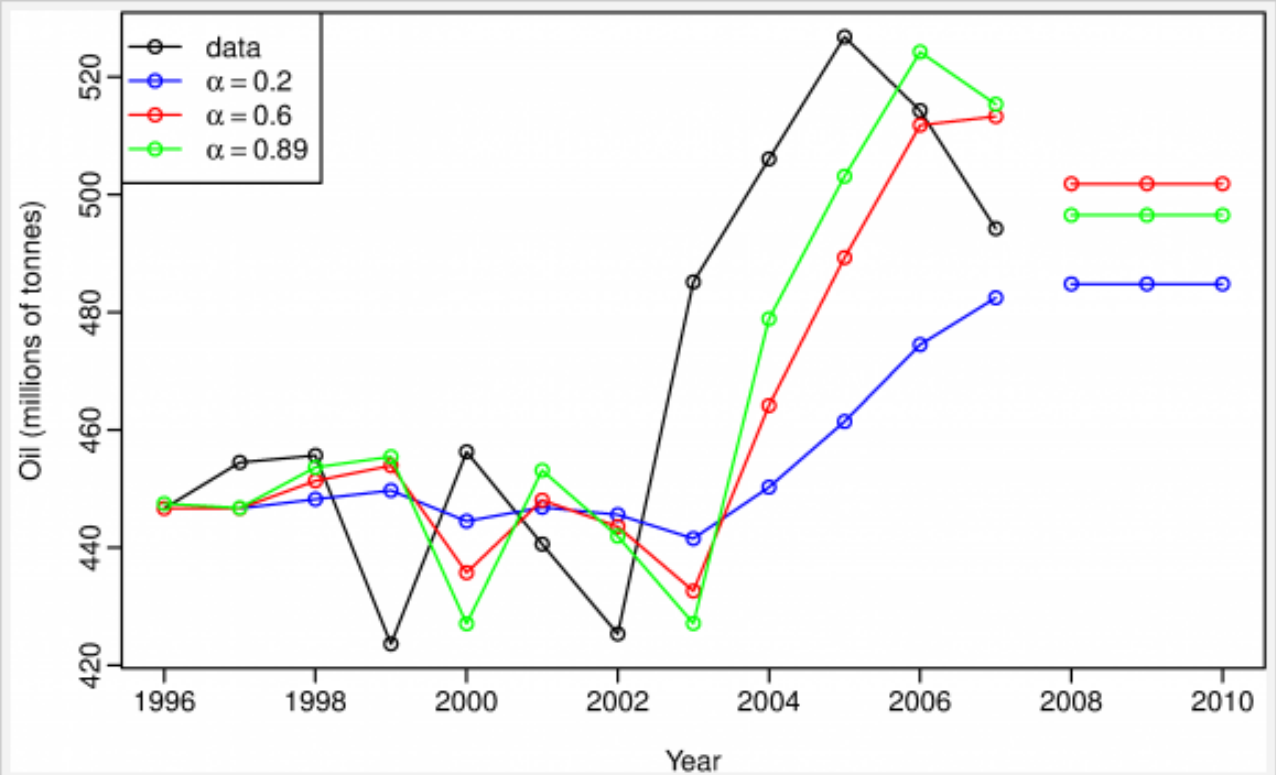



*Figure 7.2: Simple exponential smoothing applied to oil production in Saudi Arabia (1996–2007).*

https://www.otexts.org/fpp/7/1

## Holt's Trend Corrected Exponential Smoothing

galvanize

If a time series is increasing or decreasing at a fixed rate over time, we can use a linear regression model, where the parameter $\beta_1$ is simply the growth rate between t-1 and t.

$$y_t = \beta_0 + \beta_1 t + \varepsilon \qquad \textit{where} \qquad \varepsilon_t \overset{iid}{\sim} N(0, \sigma^2)$$

What if both the mean level and the rate are changing over time?

# Holt's Trend Corrected Exponential Smoothing

galvanize

1. Let $l_{t-1}$ estimate the level at time t-1 and $b_{t-1}$ estimate the growth rate
2. Our estimate of $y_t$ becomes $y_t = l_{t-1} + b_{t-1}$
3. Start with an estimate of $l_0$ and $b_0$ based on a standard linear regression for a small range of data around t=0
4. Update estimates for the rest of the data based on the following smoothing

$$l_t = \alpha y_t + (1-\alpha)[l_{t-1} + b_{t-1}]$$

This is our forecast $y_t$-hat for $y_t$

$$b_t = \gamma(l_t - l_{t-1}) + (1-\gamma)b_{t-1}$$

Actual slope

Predicted slope

# Holt-Winter's Exponential Smoothing

galvanize

- We add an additional seasonal factor
- If trend and seasonal factors were fixed:

$$y_t = TR_t + SN_t + \varepsilon \qquad \text{where} \qquad \varepsilon_t \overset{iid}{\sim} N(0, \sigma^2)$$

- If we have a changing trend but a fixed seasonal factor, we can extend Holt's method

# Holt-Winter's Exponential Smoothing

galvanize

1. Let $sn_{t-L}$ denote the most recent estimate of the seasonal factor for the season of time t, with L the number of seasons (seasonal adjustment $I_t$)
2. Begin with Holt's trend corrected method on a subset of the data
3. Compute seasonal factors as follows:
   a. De-trend data by computing $y_t - \hat{y}_t$
   b. Compute seasonal factors by averaging the de-trended values per season over all your data
4. Compute updated estimates using the following equations

$$
\begin{aligned}
l_t &= \alpha(y_t - sn_{t-L}) + (1-\alpha)[l_{t-1} + b_{t-1}] \\
b_t &= \gamma(l_t - l_{t-1}) + (1-\gamma)b_{t-1} \\
sn_t &= \delta(y_t - l_t) + (1-\delta)sn_{t-L}
\end{aligned}
$$

# ARIMA models

Afternoon lecture

galvanize

Objectives Afternoon

- Box-Jenkins methodology
- Autocorrelation functions
- Partial autocorrelation functions
- Stationarity
- AR, MA and ARIMA models

# Box-Jenkins methodology overview

galvanize

**The Box-Jenkins methodology applies autoregressive (AR) moving average (MA) models to find the best fit to a time series based on past values**

**3 stages:**

1. **Model identification**
    a. **Making sure data is stationary**
    b. **Identifying seasonality**
    c. **Plot autocorrelation functions (ACF) or partial autocorrelation functions (PACF) to decide which AR or MA components to include**
2. **Parameter estimation (MLE)**
3. **Model checking by testing the model residuals**
4. **If model inaccurate based on residuals, go back to step 1, iterate**

## Autocorrelation functions

galvanize

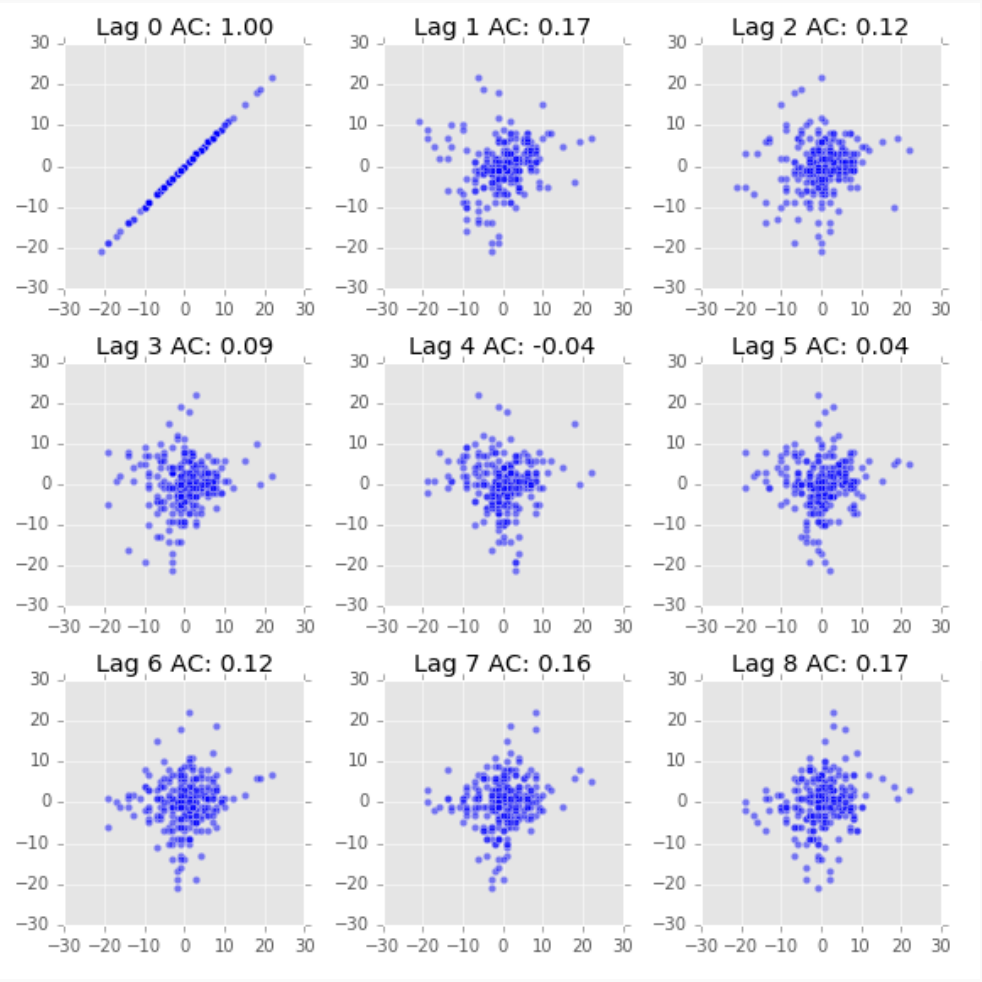
The *autocorrelation* function measures the correlation of time points based on their distance in time (also called “lags”)

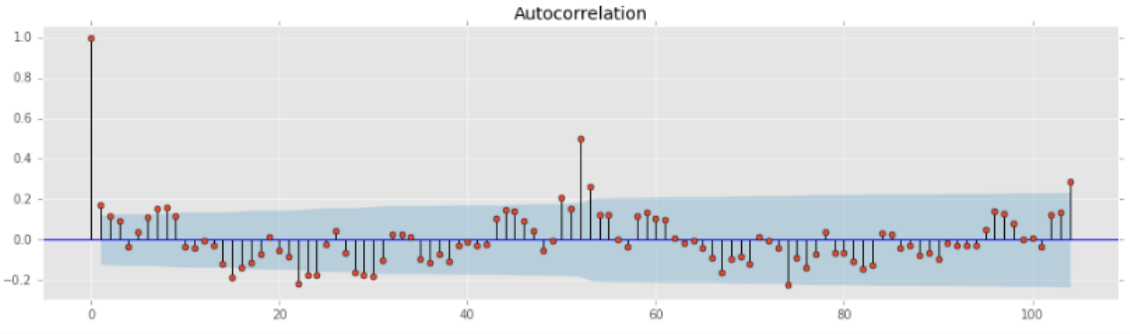
The Pearson correlation between all data points with all data points **lag h** away from them. We can calculate this number for *all possible h* and plot the ACF as a function of h

The ACF plot is also called a “correlogram”

$$r_k = \frac{\sum_{t=k+1}^{T} (y_t - \bar{y})(y_{t-k} - \bar{y})}{\sum_{t=1}^{T} (y_t - \bar{y})^2}$$

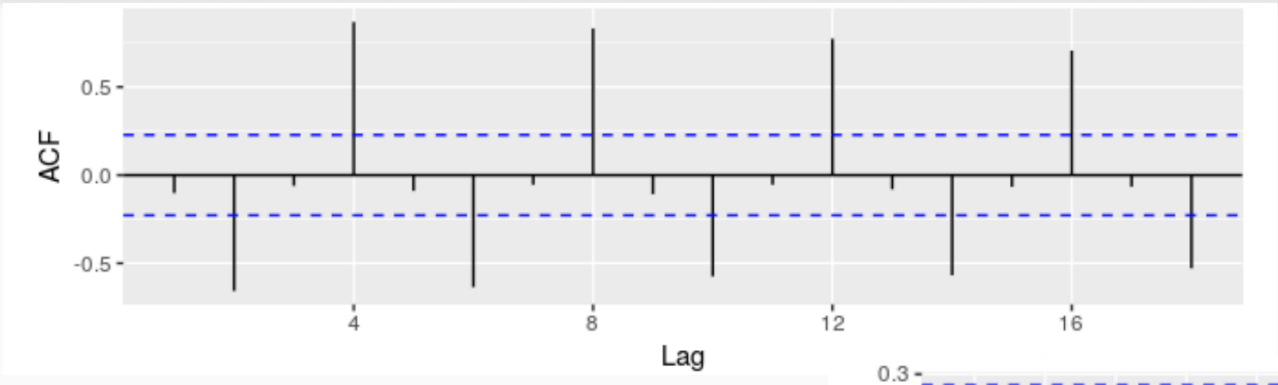
# Autocorrelation Plots

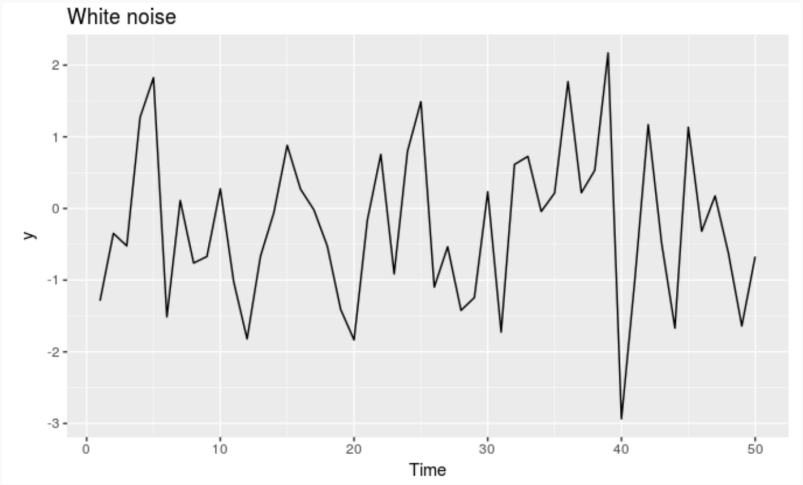


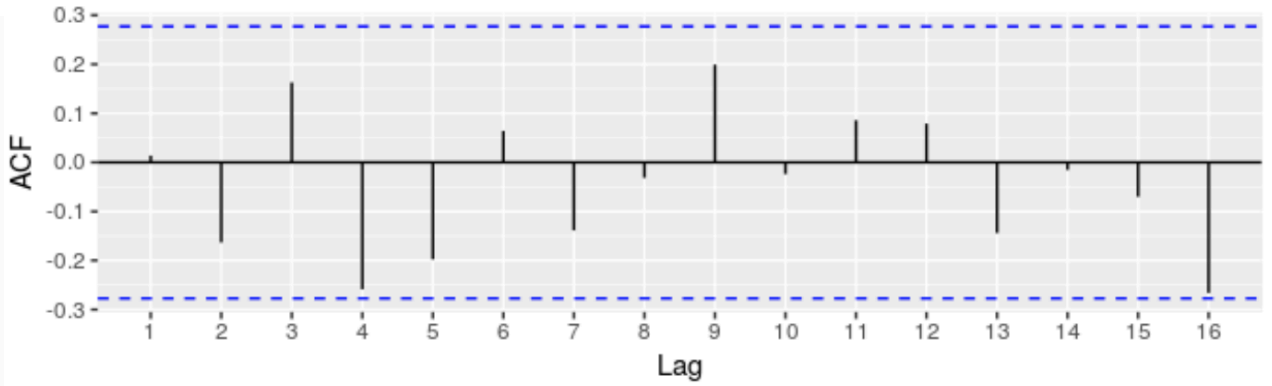



# Autocorrelation Plots

galvanize



# Partial autocorrelation functions

galvanize

If a process has a direct dependence between t and t+1, there is also an indirect dependence between t and t+2, even if t does not directly influence t+2

The ACF cannot differentiate between direct and indirect effects

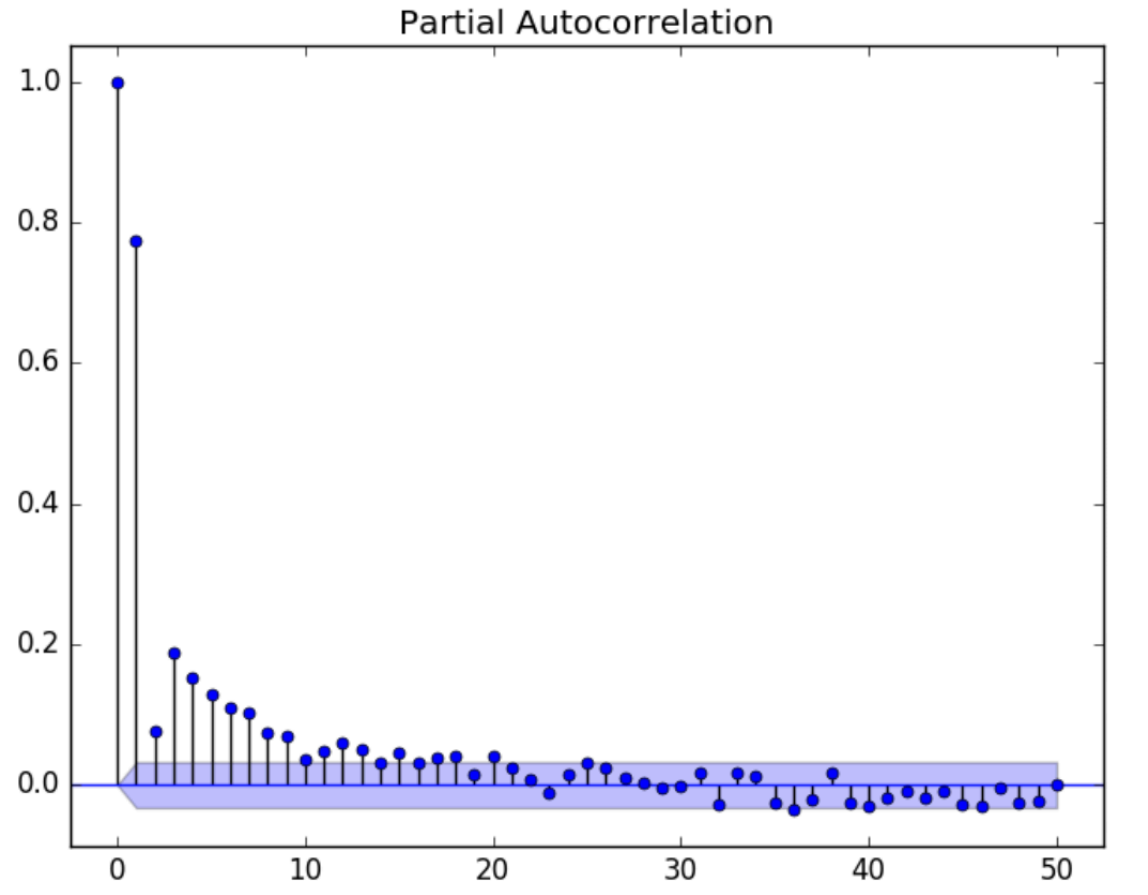
The partial autocorrelation removes all effects of intervening lags

Suppose $y_t$ is linearly dependent on the the last h time points:

$$y_t = a_1 y_{t-1} + a_2 y_{t-2} + \ldots + a_h y_{t-h}$$

The PACF at lag h will simply return the coefficient $a_h$ , removing the effects of the intervening coefficients (the math is complicated)

# Partial Autocorrelation Plots



## Stationary time series

galvanize

A stationary time series is any time series that looks identical no matter when we start observing it.

More concisely, the statistical behavior (not necessarily the exact values) of the series $y_1$, $y_2$, …, $y_t$ is identical to $y_{1+h}$, $y_{2+h}$, … ,$y_{t+h}$ for any lag h.

This is the definition for *strong stationarity* (too strong for most time series)

Real white noise is strongly stationary (normally distributed around mean 0, no matter when we start observing/sampling)

## Achieving stationarity

galvanize

Most time series are not stationary.

Here are steps we can take to make them stationary:

1. Detrending → constraint 1 (constant mean)
2. Differencing (subtracting previous values) → constraint 1
3. Transformations → constraint 2 (covariance depending only on lag)

## Achieving stationarity: detrending

galvanize

Assuming a linear trend, we can look at the residuals after linear regression

$$y_t = \beta_0 + \beta_1 t + \varepsilon_t$$

$$\hat{e}_t = y_t - \hat{\beta}_0 - \hat{\beta}_1 t$$

# Achieving stationarity: differencing

Because linearity is a strong assumption, a more popular method is differencing (often done in addition to linear detrending).

The first difference time series is defined as:

$$\nabla y_t = y_t - y_{t-1}$$

Less commonly people apply higher order differencing:

$$\begin{aligned}
\nabla^2 y_t &= \nabla[\nabla y_t] \\
&= \nabla[y_t - y_{t-1}] \\
&= [y_t - y_{t-1}] - [y_{t-1} - y_{t-2}] \\
&= y_t - 2y_{t-1} + y_{t-2}
\end{aligned}$$

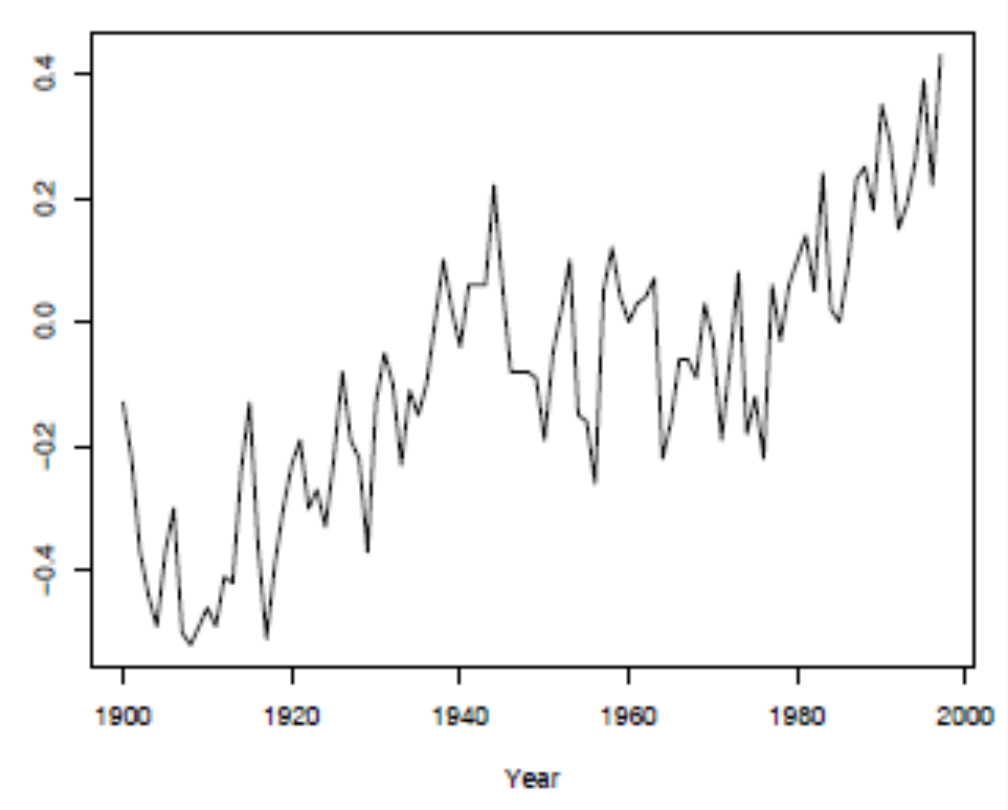
More generally: $\nabla^d y_t = (1 - B)^d y_t$

With backshift operator B: $By_t = y_{t-1}$

# Example: Global warming

galvanize

Global temperatures time series: Data is a combination of land-air temperature anomalies of the years 1900-1997

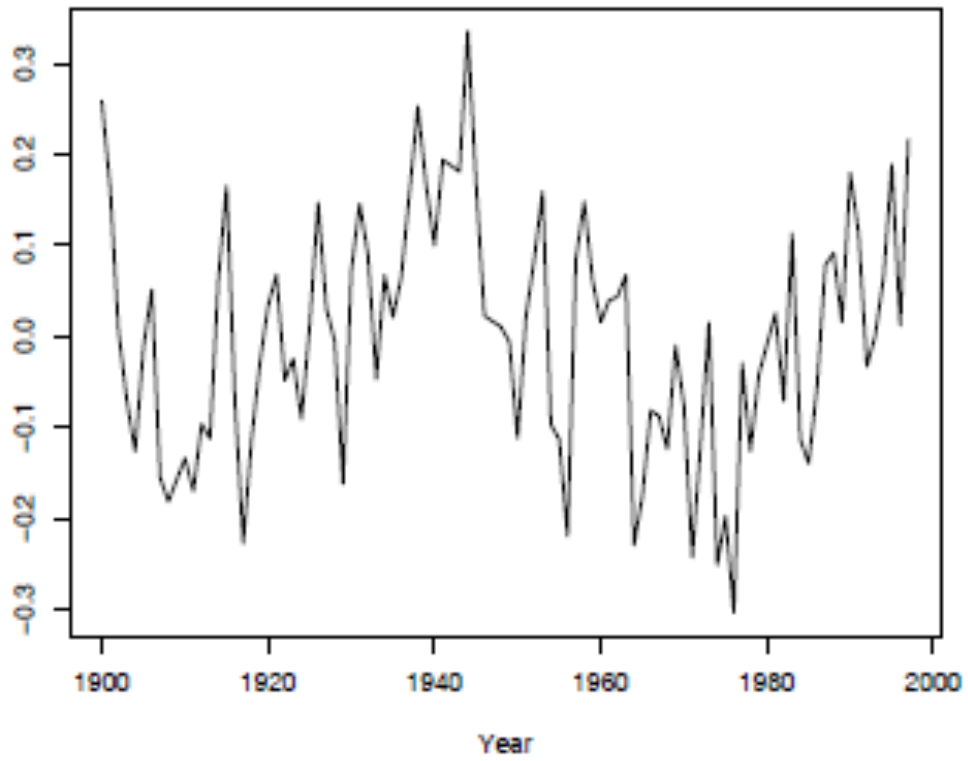


# Example: Global warming

galvanize

Let's first detrend the data by linear regression and look at the residuals

$$\widehat{Temp} = -12.2 + 0.006 * Years$$

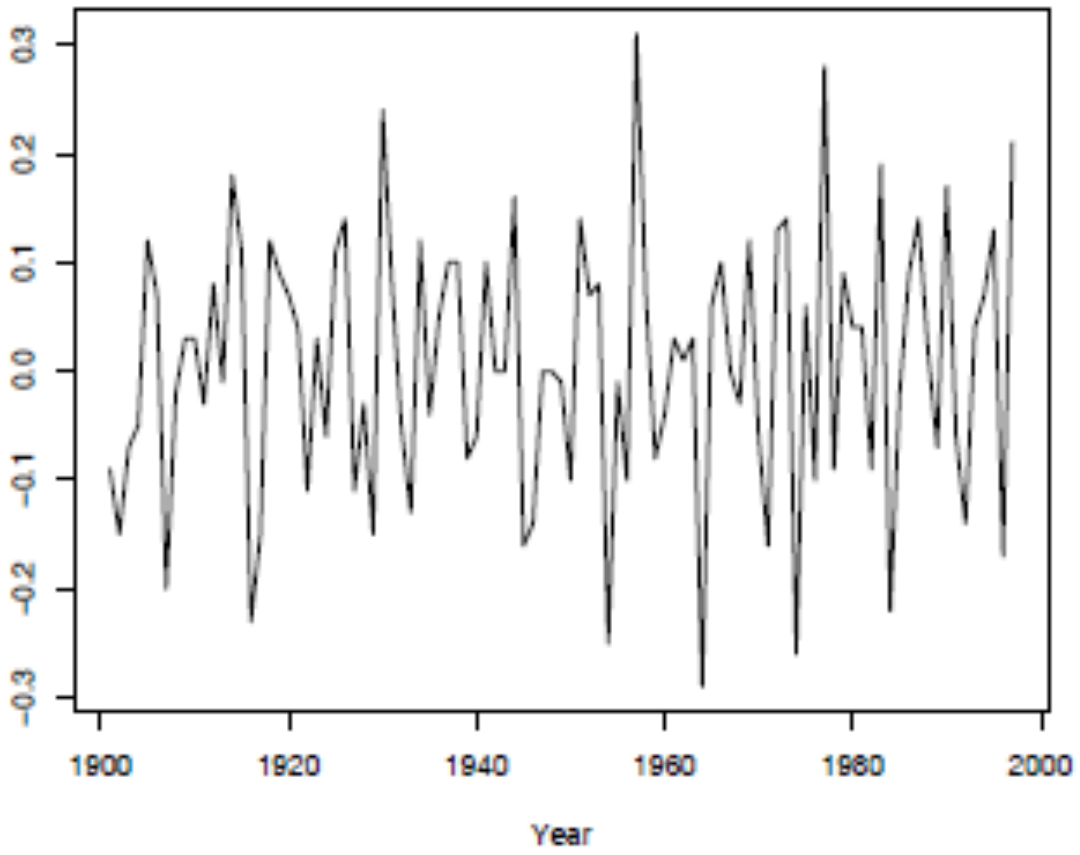



Does not look stationary yet!

# Example: Global warming

galvanize

Let's take the first difference




Looks much better!

## Achieving stationarity: transformations

galvanize

A common feature that remains after detrending and differencing is non constant variance

Treatments include the following transformations:

- The logarithmic transformation, $y_t = log(y_t)$ usually exponential-base (natural log)
- The square root transformation, $y_t = \sqrt{y_t}$, useful for count data
- More general transformations fall within the Box-Cox family

$$y_t = \begin{cases} \frac{1}{\lambda}(y_t^{\lambda} - 1) & \text{if } \lambda \neq 0 \\ \ln(y_t) & \text{if } \lambda = 0 \end{cases}$$

Box-Cox is a generalization that includes log and power transformations, we pick the λ that works best.

A good value of λ is one which makes the size of the seasonal variation about the same across the whole series, as that makes the forecasting model simpler.

## Achieving stationarity: transformations

galvanize

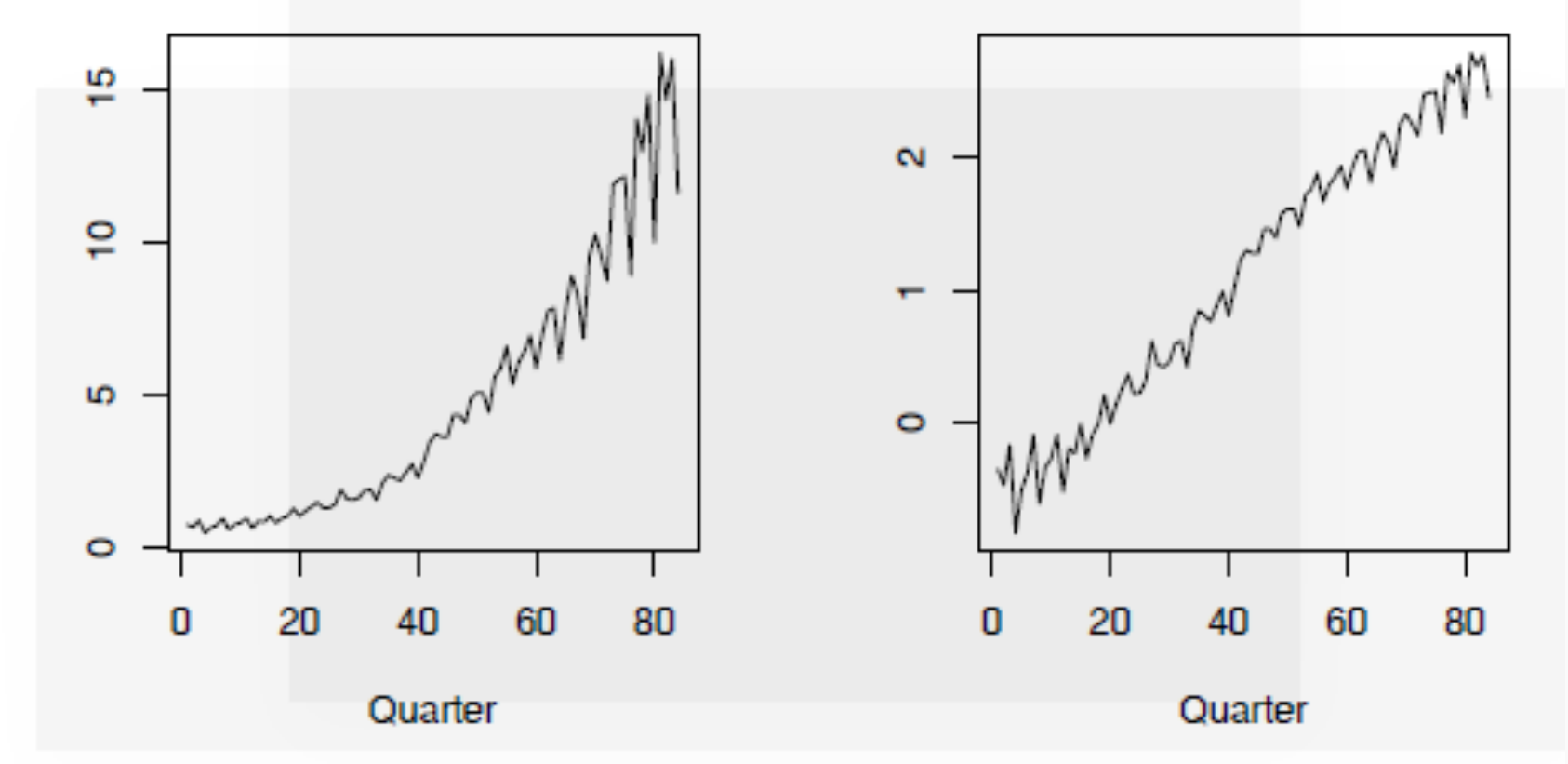
- If some $y_t \leq 0$, no power transformation is possible unless all observations are adjusted by adding a constant to all values.
- Choose a simple value of $\lambda$. It makes explanations easier.
- Forecasting results are relatively insensitive to the value of $\lambda$.
- Often no transformation is needed.
- Transformations sometimes make little difference to the forecasts but have a large effect on prediction intervals.

Other adjustments: inflation adjustment, population adjustment (per capita values instead of total), change of sampling frequency (daily instead of monthly because months have different lengths) etc

# Achieving stationarity: transformations

galvanize

Transformation example: earnings per share for Johnson&Johnson from 1960 to 1980 before and after log transformation



# Assessing stationarity

galvanize

A simple test for stationarity is the Augmented Dickey-Fuller (ADF) test, which tests the null hypothesis that ρ=1 for the model

Where $u_t$ are the errors.

If ρ=1 the series is NOT stationary, because $y_t$ is highly dependent on $y_{t-1}$

In different terms, we are testing the null hypothesis that the differences time series is expected to be zero:

(python: `statsmodels.tsa.stattools.adfuller(series)`)

The ADF test returns a t-statistic and p-value. If p ≈ 0, series is stationary

$$y_t = \rho y_{t-1} + u_t$$

$$\nabla y_t = (\rho - 1) y_{t-1} + u_t = \delta y_{t-1} + u_t$$

## Assessing stationarity: ACF

galvanize

Another assessment of stationarity is by inspection of the autocorrelation function

If ACF dies down quickly, series is stationary (exponential decay)

White noise will have an ACF that is flat at zero, a stationary time series can still show a non-zero ACF because of stickiness

If ACF does not show exponential decay we need to apply transformations to remove trend or stabilize the variance

We will then use the transformed time series as our “working” time series for the rest of the analysis

## Box-Jenkins methodology

galvanize

After achieving stationarity, we will now try to identify contributing models that explain the series' stickiness (dependence on past values or past errors), which make it different from white noise.

We can use the ACF and PACF to identify two types of models: MA (moving average) and AR (autoregressive) models

The ACF and PACF will tell us how many and which lags to consider for our MA and AR models.

If the ACF or PACF "cut off" after a certain lag k (no statistically significant autocorrelation), we will ignore contributions of lags larger than k

# Autoregressive (AR) models

galvanize

AR models describe time series whose values depend on previous values in a linear additive fashion ("autoregressive": linear regression on itself)

In an AR(p) model the value at time t can be explained by the past p values

The general pattern for an AR(p) process is an ACF that dies down exponentially and a PACF that has significant peaks up to level p and then only non significant values.

The PACF intends to estimate the coefficients $\Phi_i$

AR(1) with $\Phi_{1} = 0$ is white noise, $\Phi_{1} = 1$ is a random walk, $\Phi_{1} < 0$ will oscillate

$$AR(p) : y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \varepsilon_t$$

# Moving average (MA) models

galvanize

A moving average process describes values that are not dependent on past values but past errors

$$MA(q): y_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q}$$

An MA(q) process can be identified by an ACF that dies down after q lags.

(errors from a fit such as an AR model fit)

Resources on identifying ARIMA models
https://en.wikipedia.org/wiki/Box%E2%80%93Jenkins_method#Identify_p_and_q
http://people.duke.edu/~rnau/arimrule.htm

# AR vs MA processes

The difference between these two might not be intuitive

In an AR process, a one time shock can affect $y_t$ infinitely far into the future even if p=1 because $y_t$ influences $y_{t+1}$, which influences $y_{t+2}$ and so on.

As the influence of $y_t$ on future values $y_{t+k}$ becomes more and more diluted by random processes in between, the ACF dies down towards 0 (exponentially)

An example AR process could be rising sea levels. A shock in either direction has an obvious influence on future levels

MA processes are a bit more complex. Shocks have a finite duration influence. Values $y_t$ do not directly depend on the previous values but on the errors, i.e. on some unmeasurable effects. Examples are unmeasured effects that last for a certain duration like the circulation of a coupon which affects sales for a limited amount of time and then goes back to previous levels.

MA processes are always stationary, AR processes not necessarily

## ARIMA models

galvanize

ARMA(p,q) processes are simply a combination of an AR(p) and a MA(q) process

An ARIMA(p,d,q) adds a trend component. I(d) stands for integrated to order d (opposite of differenced). For d=1, the data is the cumulative sum of a stationary series

d is the number of times you need to difference your series to make it stationary. Usually $d \in \{0,1,2\}$

Differencing should turn an ARIMA process into an ARMA process:

```
np.diff(array, n=d)
pd.Series.diff(periods=d)
```

# ARIMA models examples

galvanize

# Box-Jenkins methodology steps

galvanize

1. Identify and apply steps to remove trends, seasonality and achieve weak stationarity (detrending, differencing, transformations)
2. Inspect time series for stationarity (visually, ACF, ADF test)
3. Plot ACF and PACF of resulting working time series to identify likely parameters for p and q of AR (use PACF) and MA (use ACF) processes
4. Fit ARIMA model with q, p and some values around those. Use d equivalent to the number of times you had to difference before achieving stationarity (but fit ARIMA model with undifferenced data)
5. Inspect residuals:
    a. If white noise (ACF zero beyond lag h=0)→ done
    b. Else: identify additional components, iterate
6. We should try and compare a few combinations of p,d,q around the ones identified visually and compare their AIC values (outputted by model fit) and pick the one with the lowest AIC as our final model

# SARIMA

galvanize

Seasonal ARIMA models include seasonality factors by including seasonal differencing

Seasonal factors can be identified by inspecting the PACF for strong spikes.

For example daily sales data might have a PACF with a strong spike at h=7, suggesting a weekly seasonal effect

SARIMA models include seasonality parameters:

$$SARIMA(p, d, q)(sp, sd, sq)_k$$

```
baseball_model = SARIMAX(baseball_series, order=(1, 1, 0), seasonal_order=(1, 0, 0, 52)).fit()
```

```
# This one is only available in the development verison of statsmodels
# Run:
#    pip install git+https://github.com/statsmodels/statsmodels.git
# to install the development version.
from statsmodels.tsa.statespace.sarimax import SARIMAX
```