Week 9: Time Series

MATH-516 Applied Statistics

Tomas Masak

Apr 24th 2023

Section 1

Stationary Time Series

What is a Time Series

Def.: A time series is a stochastic process $(X_t, t \in T)$. Alternatively, it is a set of observations indexed by time $(x_t, t \in T)$, i.e. s realization of the process.

- examples: stock prices, insurance claims, temperature, precipitation, population size, income, etc.
- we are interested in the case
  - $T \subseteq \mathbb{Z}$, i.e. observations made discretely at equidistant times
  - (X_t) is serially correlated, i.e. not i.i.d.

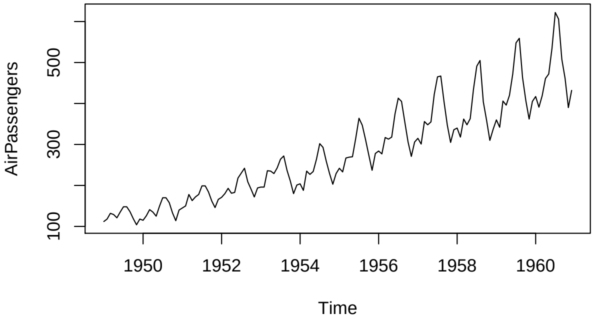
Goal: Understand the dependency patters in order to forecast, test hypotheses about the underlying process, etc.

Monthly Airline Passengers Numbers

```
str(AirPassengers)
```

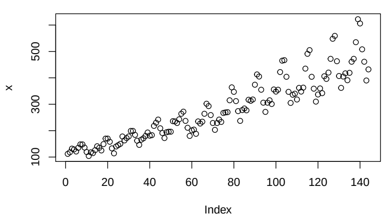
```
##  Time-Series [1:144] from 1949 to 1961: 112 118 132 129 121 135 148 148
```

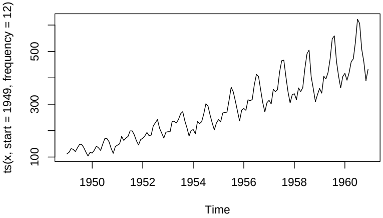
```
plot(AirPassengers)
```



ts object in R

```r
x <- as.numeric(AirPassengers)
plot(x)
plot( ts(x, start=1949, frequency=12) )
```





- data points are joined by lines for readability, despite data being discrete

Stationarity

Def.: Let $(X_t, t \in T)$ be a time series with $\mathrm{var}(X_t) < \infty$ for all t. Then $\gamma(t,s) = \mathrm{cov}(X_t, X_s)$ for $t, s \in T$ is called the autocovariance function.

Def.: A time series $(X_t, t \in T)$ is strictly stationary if vectors $(X_t, \ldots, X_{t+h})^\top$ and $(X_s, \ldots, X_{s+h})^\top$ have the same distribution for all t, s, k such that $t, \ldots, t+h, s, \ldots, s+h \in T$.

Strict stationarity implies:

1. all X_t are identically distributed,
2. $\mathbb{E} X_t = \mu$,
3. $\mathrm{cov}(X_t, X_{t+h}) = \gamma(h)$,

and much more. If instead we only assume Points 2. and 3., we speak of (weak) stationarity.

Stationarity

Def.: A time series $(X_t, t \in T)$ is (weakly) stationary if

1. $\mathbb{E}|X_t|^2 < \infty$
2. $\mathbb{E}X_t = \mu$ for all $t \in T$
3. $\gamma(t,s) = \gamma(t+h, s+h)$ for all t, s, h such that $t, s, t+h, s+h \in T$

Point 3 above implies that $\gamma(h) := \gamma(h, 0)$ contains all the information about the covariance ($h = t - s$).

- we cannot do multivariate statistics on $(x_t, t \in T)$ since from this perspective we have just a single observation
- stationarity is a structural assumption that (together with a silent but reasonable short-term dependence assumption) allows us to do statistics on this single observation
- it is impossible to prove stationarity, we can only look for evidence against it
    - stationarity is a property of the process, not of the data
    - treated as a hypothesis (plausible at best!) ((formal tests exist, but. . .))
    - in practice, it is a matter of scale

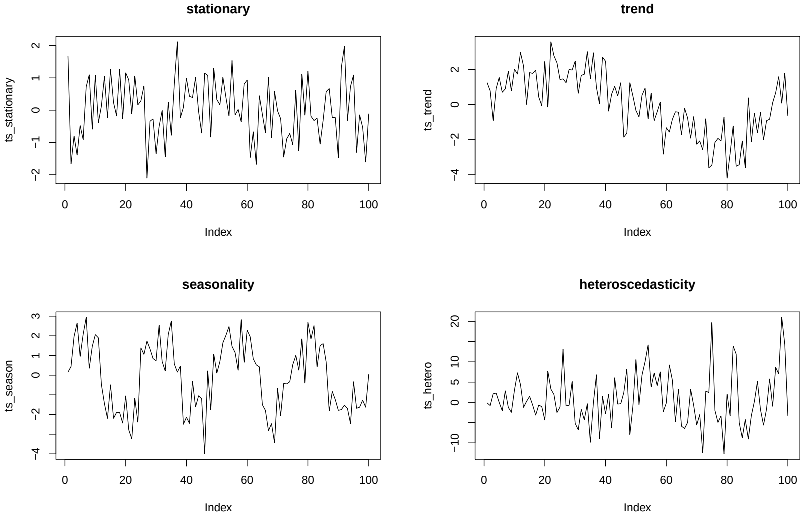
Evidence Against Stationarity

- trend
- seasonality (e.g. with period $p = 12$ for montly observations)
- non-constant variance
- structural changes
  - in the mean
  - in the dependency (i.e. 2nd order) structure

Few overly simple examples:

```r
ts_stationary <- rnorm(100) # white noise
ts_trend <- 2*sin( (1:100)/100*pi*2 ) + rnorm(100)
ts_season <- 2*sin( (1:100)/100*pi*8 ) + rnorm(100)
ts_hetero <- rnorm(100)*sqrt(1:100)
plot(ts_stationary, type="l", main="stationary")
plot(ts_trend, type="l", main="trend")
plot(ts_season, type="l", main="seasonality")
plot(ts_hetero, type="l", main="heteroscedasticity")
```

Evidence Against Stationarity



Reductions to Stationary Time Series

1. decomposition
    - when trend and seasonality are problems
    - e.g. $X_t = T_t + S_t + Y_t$ decomposes X_t into
        - the trend component T_t
        - the seasonal component S_t
        - the stationary remainder Y_t
    - regression techniques:
        - polynomials or non-parametric regression for the trend
        - Fourier basis or dummy variables for the seasonal component
2. transformations
    - when non-constant variance is the problem
    - $\log(\cdot)$ is by far the most popular
        - typically works when the time series is on a relative scale, i.e. when absolute increments changes their meaning with the level
    - other power transformations like Box-Cox can also be useful

Reductions to Stationary Time Series

3. differencing
    - when slowly varying trend and/or seasonality are present
        - $X_t = T_t + Y_t$ with $T_t \approx T_{t-1} \Rightarrow X_t - X_{t-1}$ is approximately stationary
        - $X_t = S_t + Y_t$ with $S_t = S_{t+s} \Rightarrow X_t - X_{t-s}$ is stationary
    - polynomial trends are handled explicitly
        - $X_t = T_t + Y_t$ with $T_t = at + b \Rightarrow X_t - X_{t-p} = a + Y_t - Y_{t-1}$
        - in the differenced series, a is the *drift* (a parameter that can be estimated directly)
        - similarly any polynomial trend of degree k can be reduced to a *drift* by differencing k-times

It will be useful to define the *backshift operator* B such that

- $(1-B)X_t = X_t - X_{t-1}$
- $(1-B^s)X_t = X_t - X_{t-s}$

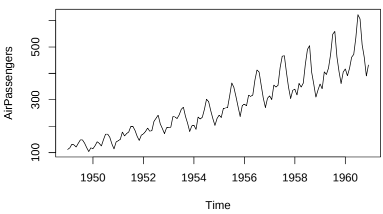
They can be applied successively or treated as polynomials, e.g.

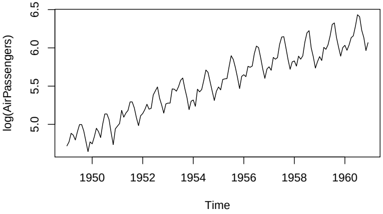
$$\begin{aligned}(1-B)^2X_t &= (1-B)(X_t - X_{t-1}) = X_t - 2X_{t-1} + X_{t-2} \\ &= (1 - 2B + B^2 2)X_t = X_t - 2X_{t-1} + X_{t-2}\end{aligned}$$

Air Passengers: Transformation

- seems like a good idea to take the log transformation first

```
plot(AirPassengers)
plot(log(AirPassengers))
```





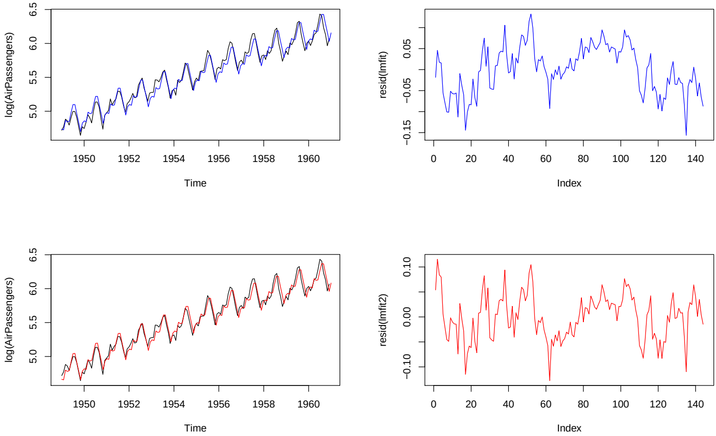
- clearly there is both trend and seasonality

Air Passengers: Regression Decompostion

- let's try reducing the logged series to a stationary one by simple regression techniques

```
lmData <- data.frame(y = log(AirPassengers),
                     month = as.factor(rep(1:12,length(AirPassengers)/12)),
                     t = 1:length(AirPassengers))
lmfit <- lm(y~., data=lmData)
plot(log(AirPassengers))
points(seq(1949, 1961, length=144),fitted(lmfit), type="l", col="blue")
plot(resid(lmfit), type="l", col="blue")
lmfit2 <- lm(y~.+I(t^2), data=lmData)
plot(log(AirPassengers))
points(seq(1949, 1961, length=144),fitted(lmfit2), type="l", col="red")
plot(resid(lmfit2), type="l", col="red")
```

Air Passengers: a Smoother

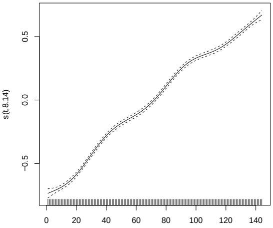


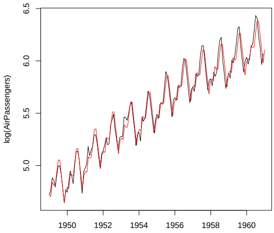
- seems like removing linear trend is not enough, but quadratic is not much better, maybe non-parametric regression is needed

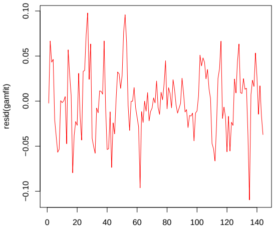
Air Passengers: Differencing

- just for illustration: non-parametric fit to the trend
  - still not perfect (variance seems to be changing), maybe different power transformation to begin with? Or a change in customer behavior (vacation sooner in the summer, etc.)?

```r
library(mgcv) # package for generalized additive models
gamfit <- gam(y~s(t)+month, data=lmData)
plot(gamfit)
plot(log(AirPassengers))
points(seq(1949, 1961, length=144), fitted(gamfit), type="l", col="red")
plot(resid(gamfit), type="l", col="red")
```



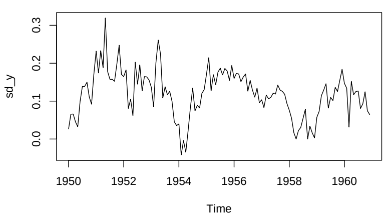


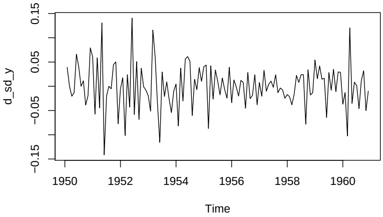


Air Passengers Revisited

- notice that seasonal differencing can take care of a trend as well
  - e.g. if $T_t = at + b$, then $T_t - T_{t-12} = 12a$
- but we do not want to rely on this, which is why below we do not end up with the left plot (we know that there is a trend in the data)

```
y <- log(AirPassengers)
sd_y <- diff(y, lag=12)
plot(sd_y)
d_sd_y <- diff(sd_y)
plot(d_sd_y)
```





Section 2

Linear Processes

Autocorrelation and Partial Autocorrelation Funtions

For a stationary time series $(X_t, t \in \mathbb{Z})$:

- $\gamma(h) = \mathrm{cov}(X_{t+h}, X_t)$ is the autocovariance function
- $\rho(h) := \gamma(h)/\gamma(0)$ is the autocorrelation function (ACF)
- $\alpha(h)$ defined below is the partial autocorrelation function (PACF)

$$\alpha(h) = \begin{cases} 1 & h = 0 \\ \rho(1) = \mathrm{corr}(X_2, X_1) & h = 1 \\ \mathrm{corr}\Big(X_{h+1} - \mathbb{E}_2[X_{h+1} \mid X_h, \ldots, X_2], X_1 - \mathbb{E}_2[X_1 \mid X_h, \ldots, X_2]\Big) & h > 1 \end{cases}$$

where $\mathbb{E}_2[X|Y]$ denotes the best linear unbiased predictor of X given Y

- think of $\alpha(h)$ as the correlation between X_1 and X_{h+1} that remains after the correlation due to all the intermediate observations $X_2, \ldots, X_h$ is taken away

Simple Processes

Def.: A time series $(Z_t, t \in \mathbb{Z})$ is white noise if it is stationary with $\mathbb{E}Z_t = 0$ for all t and $\gamma(h) = 0$ for $h \neq 0$.

- if a white noise is Gaussian, then it is formed by i.i.d. variables
- from now on $(Z_t, t \in \mathbb{Z})$ will denote white noise

Def.: $(X_t, t \in \mathbb{Z})$ is a moving average of order q, denoted MA(q), if

$$X_t = Z_t + \theta_1 Z_{t-1} + \ldots \theta_q Z_{t-q}$$

- MA(q) has autocorrelation function (ACF) $\gamma(h) = 0$ for $h > q$

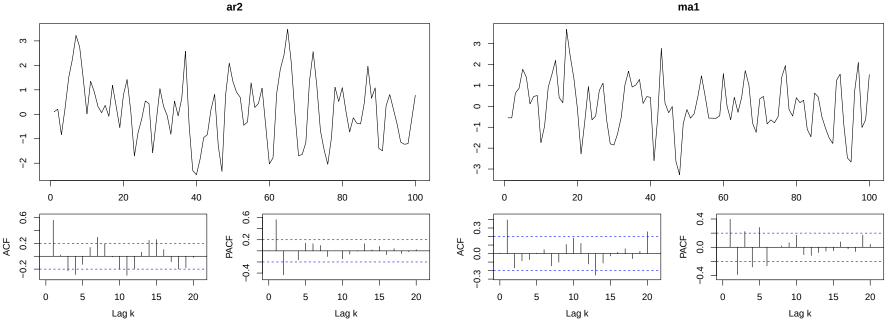
Def.: $(X_t, t \in \mathbb{Z})$ is an autoregressive process of order p, denoted AR(p), if

$$X_t = \phi_1 X_{t-1} + \ldots + \phi_p X_{t-p} + Z_p$$

- AR(p) has partial autocorrelation function (PACF) $\alpha(h) = 0$ for $h > p$

Examples

```r
library(DescTools)
set.seed(516)
ar2 <- arima.sim(list(ar=c(0.8,-0.4)), 100)
ma1 <- arima.sim(list(ma=c(0.8)), 100)
PlotACF(ar2)
PlotACF(ma1)
```



- play around with simulations to get some feeling for this!
- be careful whether lag 0 is displayed on a ACF/PACF plot or not

ARMA(p,q)

Def.: $(X_t, t \in \mathbb{Z})$ is an autoregressive moving average process of orders p and q, denoted ARMA(p,q), if

$$X_t - \phi_1 X_{t-1} - \ldots - \phi_p X_{t-p} = Z_t + \theta_1 Z_{t-1} + \ldots \theta_q Z_{t-q}$$

Fitting an ARMA process to data requires:

1. choosing the orders p and q
2. estimating the model parameters (typically Gaussian MLE)
3. model diagnostics
   - residuals should be white noise (check their ACF and PACF)
   - residuals should be roughly Gaussian (QQ-plot)
   - simulation from the fitted model (visual comparison with the original)

Comparison of competing models:

1. comparing diagnostics (ACF/PCF inspection)
2. likelihood criteria (e.g. AIC)
3. predictive checking (rolling)

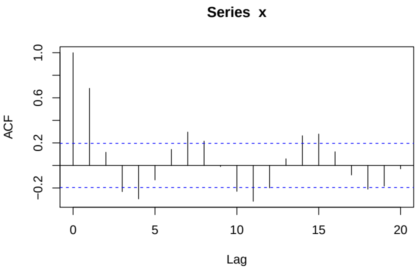
Estimating the Model Parameters

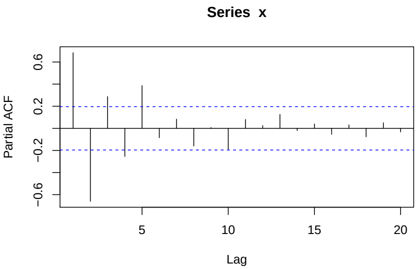
- there is a one-to-one correspondence between the autocovariance function $\gamma(h)$ and the model parameters $\phi_1, \ldots, \phi_p, \theta_1, \ldots, \theta_q, \sigma^2$ (where σ^2 is variance of the white noise)
- for ACF and PACF, the autocovariance function $\gamma(h)$ is usually estimated by method of moments and ACF and PACF are then derived from the autocovariance function
  - $\widehat{\gamma}(h) = n^{-1} \sum_{t=1}^{n-h} (x_{t+h} - \bar{x}_n)(x_t - \bar{x}_n)$ for observations $x_1, \ldots, x_n$
  - notice that $\widehat{\gamma}(h)$ is not unbiased - for that n^{-1} would have to be $(n-h)^{-1}$, but that would lead to all kinds of troubles
  - with increasing h, fewer products are averaged to obtain $\widehat{\gamma}(h)$ - ROT is to estimate only lags smaller than $10 \log_{10}(n)$
- estimation of the parameters is typically done numerically by Gaussian MLE

Choosing the ARMA orders p and q

	ACF	PACF
AR(p)	exponential decay	cut-off at lag p
MA(q)	cut-off at lag q	exponential decay
ARMA(p,q)	mix decay/cut-off	mix decay/cut-off

```r
set.seed(516)
x <- arima.sim(list(ar=c(0.8,-0.4),ma=c(0.8)), 100)
acf(x); pacf(x)
```

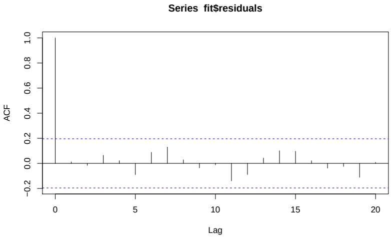


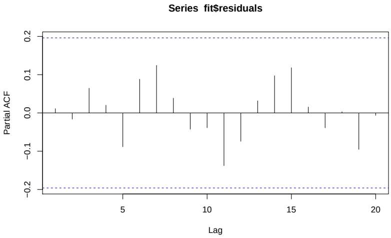


Diagnostics

- based on the ACF/PACF plot, let's start with ARMA(2,2) and check whether we can reduce one (or both) of the orders
  - actually, ACF/PACF suggests ARMA(2,1), beware that ACF shows lag 0 while PACF starts at lag 1

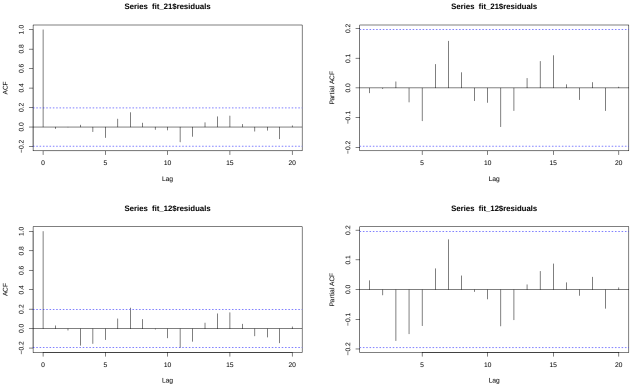
```r
fit <- arima(x, order=c(2,0,2)) # meaning of the 0 will become clear later
acf(fit$residuals); pacf(fit$residuals)
```





Diagnostics

```
fit_21 <- arima(x, order=c(2,0,1))
acf(fit_21$residuals); pacf(fit_21$residuals)
fit_12 <- arima(x, order=c(1,0,2))
acf(fit_12$residuals); pacf(fit_12$residuals)
```



- the last one seems just very slightly worse, let's compare AICs

AIC and Predictive Checking

```
c(fit$aic, fit_21$aic, fit_12$aic)
```

```
## [1] 284.3185 283.2390 287.6737
```

- so AIC correctly identifies the smaller model
- we can also do predictive checking, which also suggests the smaller model:

```
n <- length(x)
train <- 1:(n-floor(n/3))
Err <- array(0,c(3, floor(n/3)-2+1 )) # rolling 2-step ahead forecast
for(j in 0:( floor(n/3)-2 )){
  fit <- arima(x[train+j], order=c(2,0,2))
  Err[1,j+1] <- sum(x[c(1,2) + end(train)[1]+j] - predict(fit, n.ahead=2)$pred)^2
  fit <- arima(x[train+j], order=c(2,0,1))
  Err[2,j+1] <- sum(x[c(1,2) + end(train)[1]+j] - predict(fit, n.ahead=2)$pred)^2
  fit <- arima(x[train+j], order=c(1,0,2))
  Err[3,j+1] <- sum(x[c(1,2) + end(train)[1]+j] - predict(fit, n.ahead=2)$pred)^2
}
rowMeans(Err)
```

```
## [1] 5.697470 5.103734 6.777880
```

From ARMA to SARIMA

Recall the form of an ARMA process

$$X_t - \phi_1 X_{t-1} - \ldots - \phi_p X_{t-p} = Z_t + \theta_1 Z_{t-1} + \ldots \theta_q Z_{t-q},$$

and notice that it can be equivalently written using the backshift operator B as

$$\phi_{[p]}(B) X_t = \theta_{[q]}(B) Z_t,$$

where $\phi_{[p]}(x) = 1 - \phi_1 x - \ldots - \phi_p x^p$ and $\theta_{[q]}(x) = 1 + \theta_1 x + \ldots + \theta_q x^q$ are polynomials.

Now, it is easy to include (seasonal) differencing directly into the formula as

$$\phi_{[p]}(B)(1 - B^s)^D (1 - B)^d X_t = \theta_{[q]}(B) Z_t,$$

and taking into account the fact that e.g. this January can be dependent on previous Januaries in a special manner, we might want to throw in seasonal polynomials:

SARIMA Model

Def.: $(X_t, t \in \mathbb{Z})$ is a seasonal autoregressive integrated moving average process, denoted SARIMA$(p, d, q) \times (P, D, Q)_s$, if

$$\Phi_{[P]}(B^s)\phi_{[p]}(B)(1-B^s)^D(1-B)^d X_t = \Theta_{[Q]}(B^s)\theta_{[q]}(B)Z_t$$

where $\Phi_{[P]}, \phi_{[p]}, \Theta_{[Q]}$ and $\theta_{[q]}$ are polynomials of the designated order.

- additionally we can add a drift a and mean μ to obtain the following model:

$$\Phi_{[P]}(B^s)\phi_{[p]}(B)(1-B^s)^D(1-B)^d(X_t - \mu) = a + \Theta_{[Q]}(B^s)\theta_{[q]}(B)Z_t$$

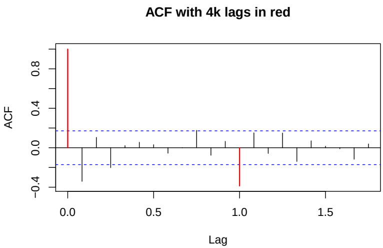
- all parameters are estimated by maximum likelihood using function `arima()` in `R`

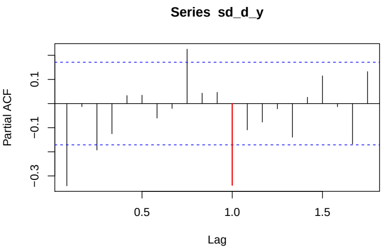
Choosing SARIMA Parameters

1. seasonality s (usually clear)
2. orders d and D of normal resp. seasonal differencing
   - like above, aiming for a stationary series
3. polynomial orders p and q
   - like above using ACF and PACF, but only looking at lags $< s$
4. seasonal polynomial orders P and Q
   - similarly, but only looking at lags ks for $k = 1, 2, \dots$ (such that $ks < 10 \log_{10}(n)$)
   - so we aim for parsimony here

- we also have to decide whether to include mean and/or drift
  - mean is like an intercept, included by default, and there is little harm in keeping it
  - drift is tricky as it somewhat clashes with (seasonal) differencing

Air Passangers Revisited





- ACF cuts off after lag 1 (or tails off really fast)
- same for the PACF
- PACF cuts off after red lag 1
  - at least we have to hope for, we don't have data for more complicated behavior
- same for PACF

Overall, SARIMA$(1,1,1) \times (1,1,1)_{12}$ and sub-models reducing one of (p,q) and one of (P,Q) to zero are the candidates (remember that ACF and PACF are the same at lag 1, and typically similar at the red lag 1)

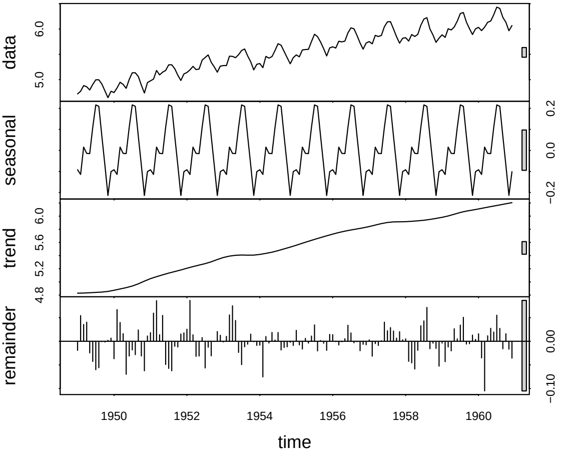
Two Popular Automated Functions

1. `stl()`
   - decomposition into trend, seasonal component, and remainder by Loess (non-parametric smoother) in an iterative manner
   - arbitrary but powerful for a quick look at the data
2. `auto.arima()` from `forecast` package
   - Box-Cox transformation performed first (so no need to take log of the data yourself)
   - it allows by default for a drift, mean, and the remaining SARIMA parameters (for fixed orders d, D, p, P, q, Q)
   - search over small orders is performed and the best model picked by AIC
   - essentially fully automatizes everything we discussed today, but...

Never use those as the first thing you do!

Air Passengers Revisited

```
plot(stl(log(AirPassengers), s.window = "periodic"))
# very similar results to GAM above
```



Air Passengers Revisited

- `auto.arima()` would choose SARIMA$(0,1,1) \times (0,1,1)_{12}$ (without drift) on the logged data, but Box-Cox chooses another transformation

```
library(forecast)
autofit <- auto.arima(AirPassengers)
autofit
```

```
## Series: AirPassengers
## ARIMA(2,1,1)(0,1,0)[12]
##
## Coefficients:
##          ar1     ar2      ma1
##       0.5960  0.2143  -0.9819
## s.e.  0.0888  0.0880   0.0292
##
## sigma^2 estimated as 132.3:  log likelihood=-504.92
## AIC=1017.85   AICc=1018.17   BIC=1029.35
```

leading to the SARIMA$(2,1,1) \times (0,1,0)_{12}$ model

Section 3

Project 5

Data

There are 3 time series:

1. temperature anomalies
   - anomalies are carefully defined in ecology as departures from a reference value, usually taken as some sort of long-term average (here it is probably monthly averages from some time point until 1960)
   - the temperature from which the anomalies are calculated is a global average
2. CO2 emissions
   - worldwide (when `Entity="World"` is chosen)
3. atmospheric CO2 concentrations
   - measured at a specific locations (that's why they show very strong seasonality)

Goal: Fit s SARIMA model, produce short-term forecasts, decide on the form of global warming.

Notes

1. Note that given how our temperature series is constructed, it is either possible that there is no yearly seasonality, which wouldn't be too surprising given the fact that we are analyzing anomalies from a global average. On the other hand, given that most of the earth's landmass is in the northern globe, there may be some sort of a seasonal behavior. This is an interesting **question** to be investigated.
2. The second interesting question is an obvious one: what is the form of global warming? I remember seeing a paper once where the claim was that the trend is exponential, predicting inferno on earth until 2050 (if I remember correctly, this was a paper from about 2005, and I was reading it around 2015, no I cannot find it anymore). Can we **prove or refute the exponential trend** and see some predictions of our own?
3. The third question, which is however too involved and I do not expect anyone to try to tackle this, is how CO2 influences the temperature. Initially, I thought we could produce results similar in flavor to these plots, and I have done something, but even the simple things we could actually do (as opposed to proper climate modeling, check out e.g. here) seem too complicated, too arbitrary (see the confidence intervals over the observations on the right plot), and quite questionable from the mathematical perspective. Hence we are only going to analyze the temperature series, not the two CO2-related series. Still, you should just take a look at the CO2-related series, and you can at least do some mental exercise about this.

Specific Tasks

1. Read the three data frames, form them into `ts()` objects and plot.
2. Build a SARIMA model for (a part of) the temperature series.
   - choose at least one model "manually", i.e. by differencing, visualizing data, ACF, PACF, etc.
3. Perform residual diagnostics and predictive checking for the candidate models to select your final model.
4. Visualize your predictions until 2050 using the final model.
5. After Lecture 10 try to answer the two questions above (is there seasonality in the temperature time series, and what is the form of global warming). Now, what form does global warming take?
   - if you feel a bit lost, focus first on the simpler 1970-2005 period, where the trend can be modeled simply as linear in time