Time Series and Forecasting
Building SARIMA Models

Maria Eduarda Silva

FEP.UP

General Procedure (Box-Jenkins approach)

Basic steps to fitting SARIMA models to time series data

- Plot and identify important characteristics of the data
- Consider transforming the data if necessary: Box-Cox to stabilize variance
- If the data is non stationary: take first and/or seasonal differences until data appears stationary
- Examine the ACF/PACF to check stationarity and model order
- Parameter estimation
- Diagnostics
  - Adequacy of the model- analysis of the residuals
  - Statistical significance of the model
- Use AIC_C to choose among models
- Compute forecasts

Box-Jenkins approach

Step 1 Identify the model

Step 2 Estimate the model

Step 3 Diagnostics check

- Adequacy Are the residuals uncorrelated?
- Statistical significance Are all the parameters statistically significant?

Box-Jenkins approach

Step 1 Identify the model

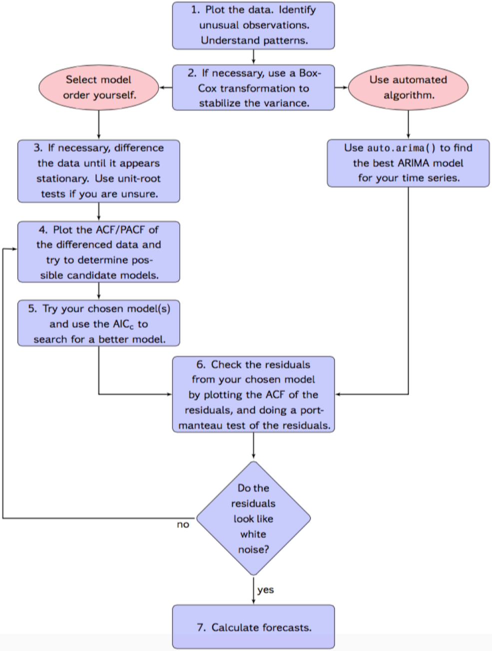
Step 2 Estimate the model

Step 3 Diagnostics check

- Adequacy Are the residuals uncorrelated?
- Statistical significance Are all the parameters statistically significant?

If both Adequacy and Statistical significance are true you found an adequate model

If at least one of Adequacy or Statistical significance is false return to Step1


1. Plot the data. Identify unusual observations. Understand patterns.
2. If necessary, use a Box-Cox transformation to stabilize the variance.
Select model order yourself.
Use automated algorithm.
3. If necessary, difference the data until it appears stationary. Use unit-root tests if you are unsure.
Use auto.arima() to find the best ARIMA model for your time series.
4. Plot the ACF/PACF of the differenced data and try to determine possible candidate models.
5. Try your chosen model(s) and use the AIC$_c$ to search for a better model.
6. Check the residuals from your chosen model by plotting the ACF of the residuals, and doing a portmanteau test of the residuals.
Do the residuals look like white noise?
no
yes
7. Calculate forecasts.

Characterization of the time series

- Plot the data in a **chronogram**.
- Check for:
  - discontinuities such as level changes
  - unusual observations- outliers
  - changes in variance
  - seasonality
  - trend
  - cycles

  Some deterministic componentes due to physical phenomena may be present and may be removed by deterministic functions: yearly periodicities

Exemple: a series with a discontinuity

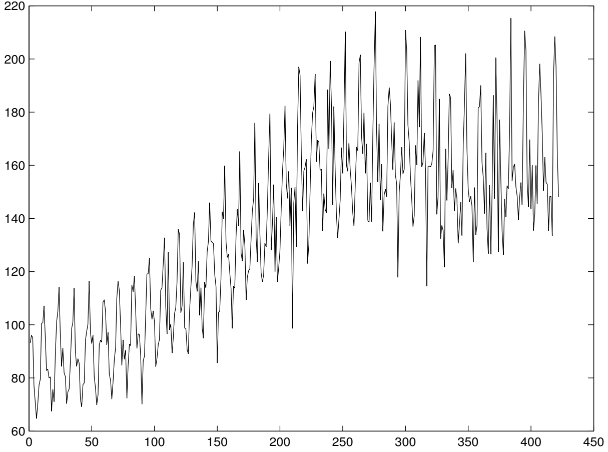
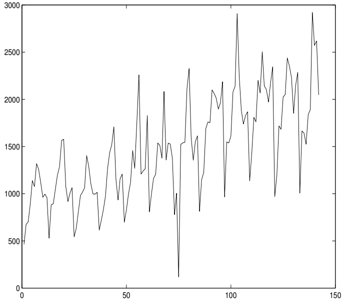


Figure: Australian beer production Jan 1956 - Abril de 1990.

Exemple: outlier

(a)



(b)

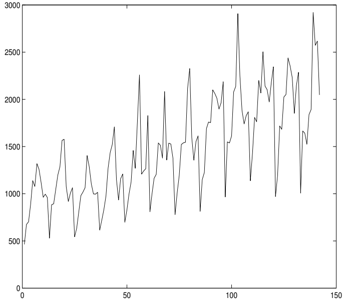


Figure: Monthly sales (kl) of red wine in Australia Jan 1980 - Out 1991: the outliers was an input error at $t = 75$ (a) original series (b).

Exemple: trend, seasonality and heterocedasticity

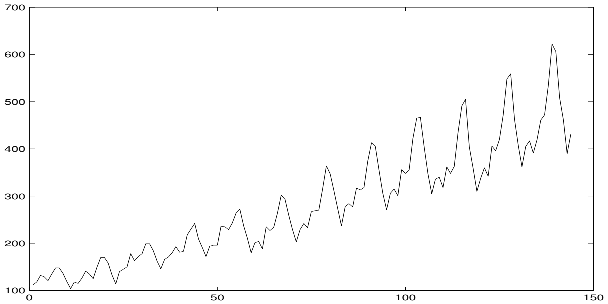


Figure: Number of airline passengers ($\times 10^3$) Jan 49 -Dec 60.

Box-Cox transforms

Stabilize the variance

$$U_t = \begin{cases} \frac{X_t^\lambda - 1}{\lambda} & se \quad \lambda \neq 0 \\ \log X_t & se \quad \lambda = 0 \end{cases}$$

Choose λ that minimizes variance of data

Example

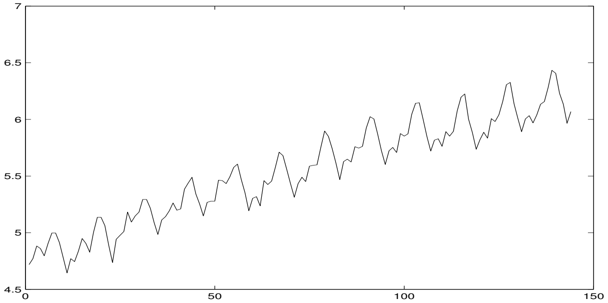


Figure: Log airline passengers

Transformations to stabilize the variance (Hyndman, MelbourneRUG.pdf, page 101)

If the data show different variation at different levels of the series, then a transformation can be useful.
Denote original observations as $y_1, \ldots, y_n$ and transformed observations as $w_1, \ldots, w_n$.

$$w_t = \begin{cases} \log(y_t), & \lambda = 0; \\ (y_t^\lambda - 1)/\lambda, & \lambda \neq 0. \end{cases}$$

- $\lambda = 1$: (No substantive transformation)
- $\lambda = \frac{1}{2}$: (Square root plus linear transformation)
- $\lambda = 0$: (Natural logarithm)
- $\lambda = -1$: (Inverse plus 1)

Back-transformations (Hyndman, MelbourneRUG.pdf, page 104)

We must reverse the transformation (or *back-transform*) to obtain forecasts on the original scale. The reverse Box-Cox transformations are given by

$$y_t = \begin{cases} \exp(w_t), & \lambda = 0; \\ (\lambda W_t + 1)^{1/\lambda}, & \lambda \neq 0. \end{cases}$$

```
lam <- BoxCox.lambda(a10) # = 0.131
fit <- ets(a10, additive=TRUE, lambda=lam)
plot(forecast(fit))
plot(forecast(fit),include=60)
```

Other transforms

- Length of the month: since the different months of the year have different number of days and also because of leap year, one may adjust to the length of the month as follows:

$$W_t = X_t \times \frac{365.25/12}{\text{no days in month } \; t}$$

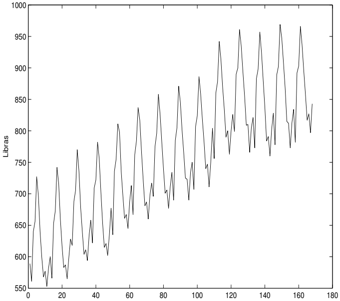
- Number of working days: after adjusting for the length of the month

$$W_t = X_t \times \frac{\text{mean number of working in a month}}{\text{number of working days in month t}}$$

- Adjust for moving holidays and interventions in general.

Exemple: monthly milk production per cow

(a)



(b)

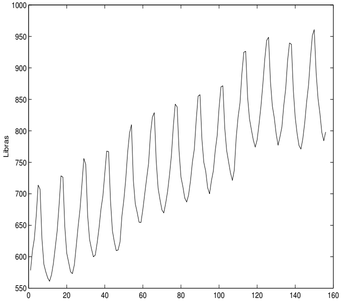
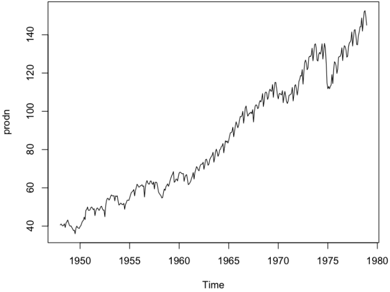


Figure: monthly milk production per cow **(a)** adjusted for length of month **(b)**.

Statistical tests to determine the required order of differencing

- Augmented Dickey Fuller test: null hypothesis is that the data are non-stationary and non-seasonal.
- Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test: null hypothesis is that the data are stationary and non-seasonal.
- Other tests available for seasonal data.

Example:Federal Reserve Board Production Index data




```
library(tseries)
adf.test(prodn,alternative="s")

        Augmented Dickey-Fuller Test

data:  prodn
Dickey-Fuller = -2.9333, Lag order = 7, p-value = 0.183
alternative hypothesis: stationary
```

p-value$>$ 0.05 indicates the need for first difference

KPSS: Test for *unit root*

- Kwiatkowski-Phillips-Schmidt-Shin (KPSS) test: reverses the hypotheses, so the null-hypothesis is that the data are stationary in level or trend
- In this case, small p-values (e.g., less than 0.05) suggest that differencing is required.

```
kpss.test(prodn)

        KPSS Test for Level Stationarity

data:  prodn
KPSS Level = 7.3711, Truncation lag parameter = 4, p-value = 0.01
```

`ndiffs`, `nsdiffs` from package `forecast`

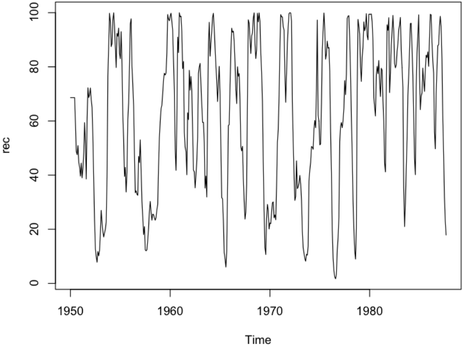
Determine the lowest number of non-seasonal and seasonal necessary for the series to become stationary

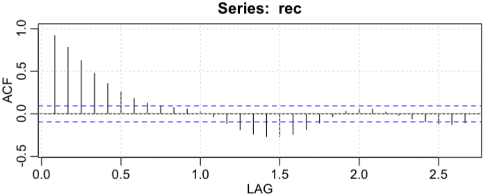
```
library(forecast)
ndiffs(WWWusage)
[1] 1
nsdiffs(log(AirPassengers))
[1] 1
ndiffs(diff(log(AirPassengers),12))
[1] 1
```

Identify the dependence orders of the model

- Compute and plot Sample ACF, SACF, e Sample PACF, SPACF
- Try to identify tentative orders for AR and/or MA components

Example: identify orders of the model: recruitment data





```r
library(astsa)
str(rec)
plot(rec)
acf2(rec)
```

The parameters must be significantly different from zero: at a 5% level parameter θ estimated by $\hat{\theta}$ with standard error *se* is significantly different from zero if $0 \notin \hat{\theta} \pm 2$ *se*.

Coefficients: ar1 ar2 intercept 1.3512 -0.4612 61.8585 s.e. 0.0416 0.0417 4.0039

both parameters and mean are statistically different from 0

Testing the residuals

The residuals must be UNCORRELATED

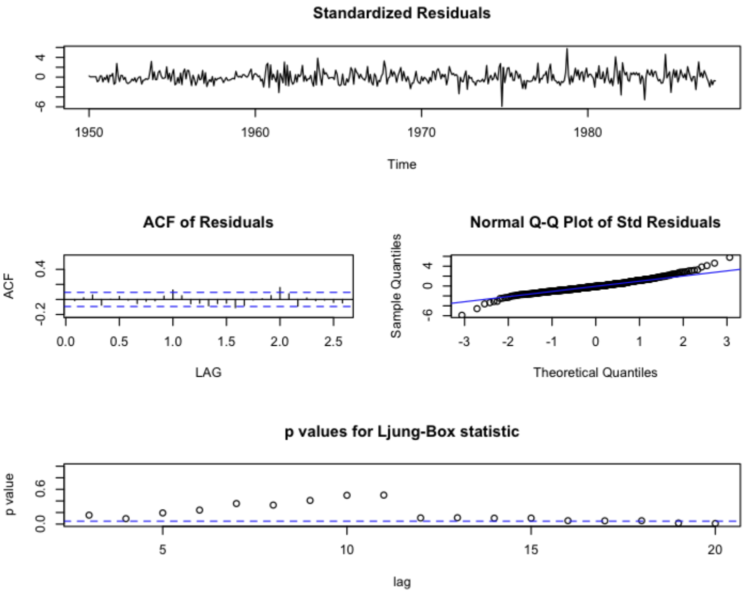
- Bartlett test: if the residuals are approximately iid then the sample acf of the residuals is $N(0, 1/n)$.
- Ljung-Box test: under the hypothesis of iid residuals $Q_{LB} = n(n+2)\sum_{j=1}^{h} \hat{\rho}^2(j)/(n-j) \sim \chi^2_h$ and graph the p-values
- Normal probability plot to check for departures from Gaussianity

The sarima function produces the necessary plots

rec data

```
sarima(rec,2,0,0)

Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
    Q), period = S), xreg = xmean, include.mean = FALSE, optim.control = li
    REPORT = 1, reltol = tol))

Coefficients:
         ar1      ar2    xmean
      1.3512  -0.4612  61.8585
s.e.  0.0416   0.0417   4.0039

sigma^2 estimated as 89.33:  log likelihood = -1661.51,  aic = 3331.02

$AIC#$
[1] 5.505631

$AICc#$
[1] 5.510243

$BIC#$
[1] 4.532889
```

Checking the residuals for rec data

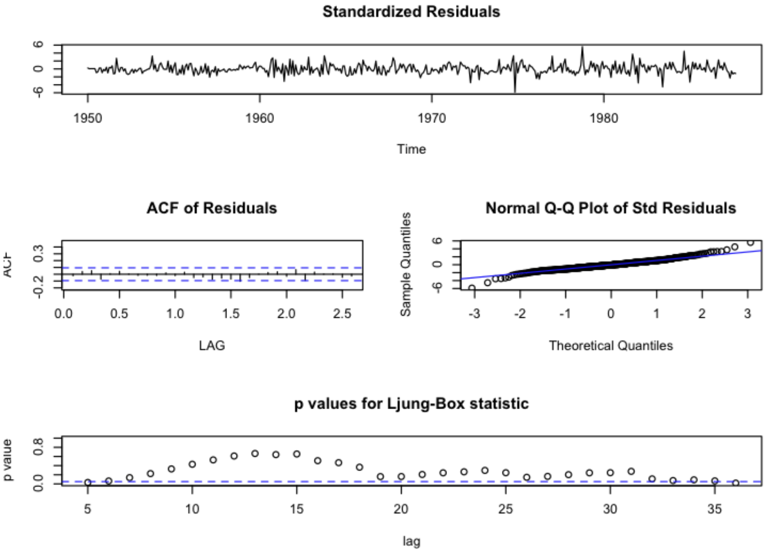


rec data

```
sarima(rec,3,0,0)

Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
    Q), period = S), xreg = xmean, include.mean = FALSE, optim.control = li
    REPORT = 1, reltol = tol))

Coefficients:
         ar1      ar2      ar3    xmean
      1.3318  -0.4043  -0.0421  61.9256
s.e.  0.0469   0.0759   0.0469   3.8411

sigma^2 estimated as 89.17:  log likelihood = -1661.11,  aic = 3332.22
```

The coefficiente for the AR(3) is not significant and the residuals checks worsened.

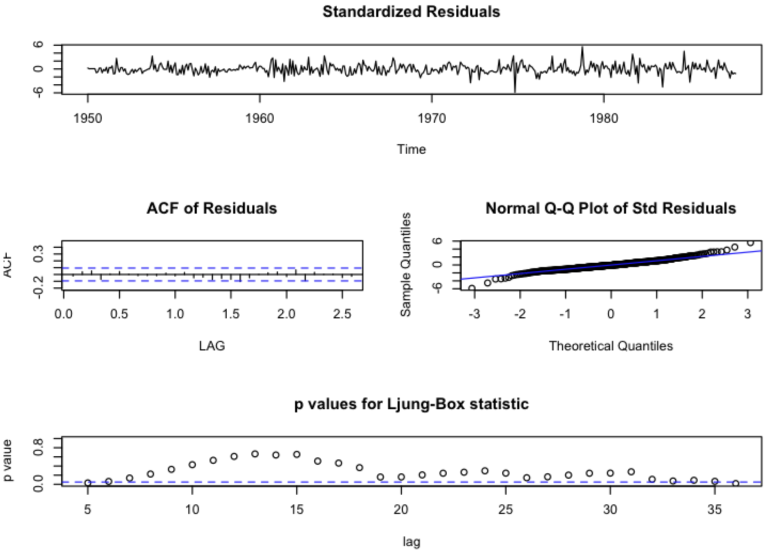
Checking the residuals for rec data



rec data

```
sarima(rec,p=2,d=0,q=0,P=2,S=12)
Call:
stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
    Q), period = S), xreg = xmean, include.mean = FALSE, optim.control = li
    REPORT = 1, reltol = tol))

Coefficients:
         ar1      ar2    sar1    sar2    xmean
      1.3256  -0.4217  0.1116  0.1641  61.6089
s.e.  0.0431   0.0437  0.0460  0.0481   6.0800

sigma^2 estimated as 85.54:  log likelihood = -1652.14,  aic = 3316.28

$AIC#$
[1] 5.471106

$AICc#$
[1] 5.475937

$BIC#$
[1] 4.516536
```

Checking the residuals for rec data



Information Criteria

Akaike (1969, 1973, 1974) suggested measuring the goodness of a model by balancing the error of the fit against the number of parameters in the model. Thus Akaike Information Criteria was born. Later developed into AICc and BIC (Bayesian Information Criteria)

Definition 2.1 Akaike's Information Criterion (AIC)

$$\mathrm{AIC} = \log\ \widehat{\sigma}_k^2 + \frac{n+2k}{n}, \tag{2.16}$$

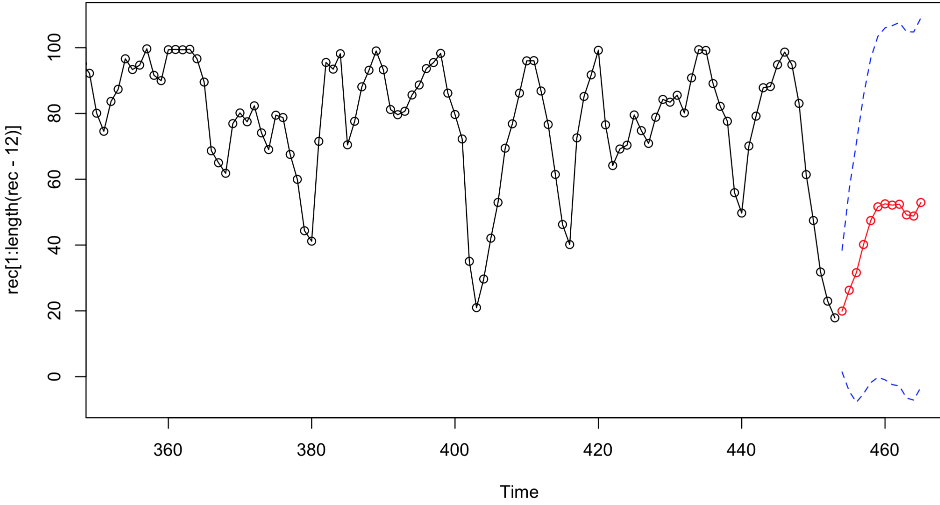
where $\widehat{\sigma}_k^2$ is given by (2.15) and k is the number of parameters in the model.

Model selection for rec data

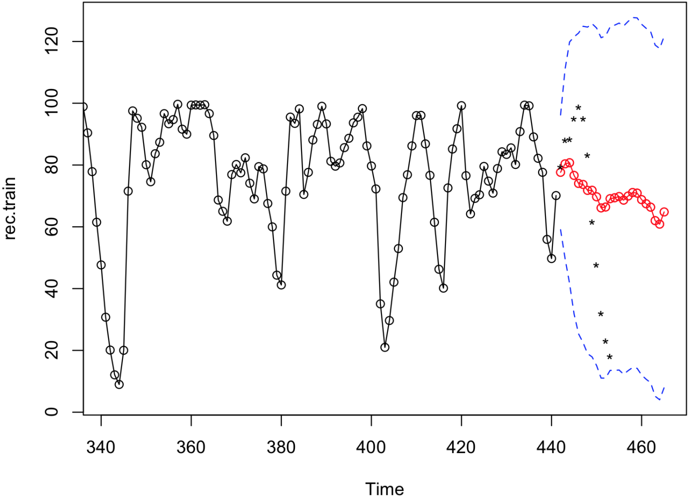
2 adequate models for rec data. Choose the model with minimum AIC (BIC): SARIMA$(2,0,0,2,0,0)_{12}$

	Model	
Coefficients	AR(2)	SARIMA$(2,0,0,2,0,0)_{12}$
Mean	61.86	61.61
	(4.0)	(6.08)
AR1	1.354	1.33
	(0.040)	(0.040)
AR2	-0.46	-0.42
	(0.040)	(0.040)
SAR1		0.11
		(0.05)
SAR2		0.16
		(0.05)
AIC	5.50	5.47
BIC	4.53	4.51

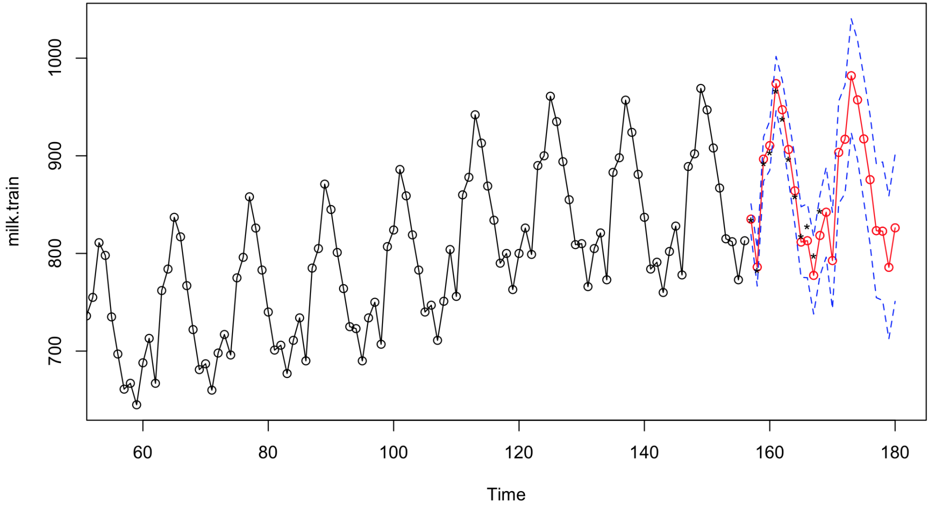
Forecasting for rec data



Forecasting for rec data



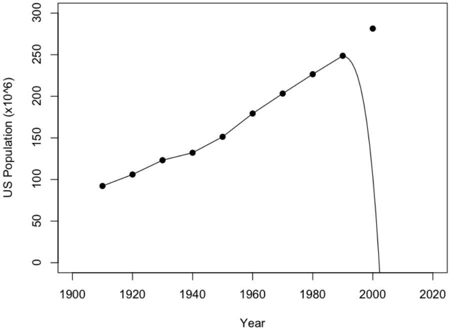
Forecasting for milk data



Overfitting

- Be aware of overfitting
- More is not always synonym of better
- Overfitting leads to less precise estimators
- Adding more parameters may fit the data better but may also lead to bad forecasts
- Example: The fit dor the U.S. population by official census from 1910 to 1990 is perfect but the forecasts are terrible: negative population sometime in 2002! The fit is obtained from polinomial of degree 8!

Example:US population




```
xpop=USPop$population[13:21]
tt=USPop$year[13:21]
tt1=tt-mean(tt)
xpop.pred=predict.lm(xpop.fit,new)
plot(USPop$year[13:22],USPop$population[13:22],pch=19,xlim=c(1900,2020),yli
lines(tt,xpop.fit$fitted.values)
lines(c(1990,seq(1991,2010,1)),c(xpop.fit$fitted.values[21],xpop.pred))
```

Hands on

Now try with the following data sets:

Quarterly U.S. GNP, gnp from the astsa package

AirPassengers