STAT302: Time Series Analysis

Chapter 3. Time Series Regression

Sangbum Choi, Ph.D

Department of Statistics, Korea University

Outline

Linear Models with Time Series

Harmonic Regression

Splines for Nonlinear Trend

Model Selection and Forecasting

Decompositon of time series

- Our general strategy is to decompose Y_t by non-stationary parts and stationary part (Wold decomposition, Doob-Meier decomposition).
- For example,

$$Y_t = T_t + S_t + R_t$$

  - $T_t =$ trend;
  - $S_t =$ seasonality with period d in the sense that $S_t = S_{t+d}$;
  - $R_t =$ weakly stationary errors

Decompositon of time series

- Thus, before estimating mean and covariance of R_t, we will first model/remove trend and seasonality.
- Four major methods are
  1. Regression
  2. Decomposition
  3. Smoothing (local regression)
  4. Differencing

Multiple linear regression

We may consider a time series regression:

$$Y_t = \beta_0 + \beta_1 X_{1,t} + \beta_2 X_{2,t} + \cdots + \beta_k X_{k,t} + \varepsilon_t.$$

- Y_t is the "response" variable
- Each $X_{j,t}$ is numerical and is called a "predictor". They are usually assumed to be known for all past and future times.
- The coefficients $\beta_1, \ldots, \beta_k$ measure the effect of each predictor after taking account of the effect of all other predictors in the model.
- That is, the coefficients measure the **marginal effects**.
- ε_t is a white noise error term.

The graph of the population data, which contains no apparent periodic component, suggests trying a model of the form

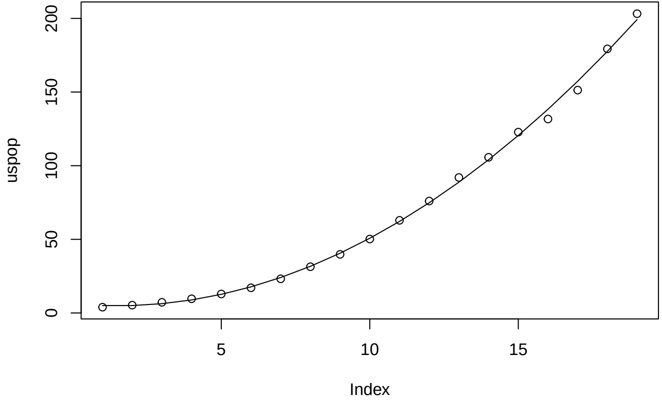
$$Y_t = T_t + R_t$$

with a 2nd-order polynormial regression

$$T_t = a_0 + a_1 t + a_2 t^2.$$

uspop data

```
uspop=as.numeric(uspop)
time=1:length(uspop)
fit=lm(uspop~time+I(time^2))
plot(uspop)
lines(predict(fit)~time)
```



Linear regression in matrix formulation

- Let $\boldsymbol{Y} = (Y_1, \ldots, Y_T)'$, $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_T)'$, and

$$\boldsymbol{X} = \begin{bmatrix} 1 & X_{1,1} & X_{2,1} & \ldots & X_{k,1} \\ 1 & X_{1,2} & X_{2,2} & \ldots & X_{k,2} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & X_{1,T} & X_{2,T} & \ldots & X_{k,T} \end{bmatrix}.$$

- Then, the linear regression takes the form:

$$\boldsymbol{Y} = \boldsymbol{X}\beta + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_k)'$ is the regression coefficient parameter.

Least squares estimation (LSE)

- Ordinary least squares (OLS) estimation finds the coefficient β by minimizing the error sum of squares (SSE):

$$Q(\beta) = (\boldsymbol{Y} - \boldsymbol{X}\beta)'(\boldsymbol{Y} - \boldsymbol{X}\beta)$$

- Differentiating it wrt $\boldsymbol{\beta}$ gives the normal equation:

$$\boldsymbol{X}'(\boldsymbol{Y} - \boldsymbol{X}\beta) = 0,$$

which results in the least-squares estimator (LSE):

$$\hat{\beta} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}$$

- The variance can be estimated by

$$\hat{\sigma}^2 = \frac{1}{n-k-1}(\boldsymbol{Y} - \boldsymbol{X}\hat{\beta})'(\boldsymbol{Y} - \boldsymbol{X}\hat{\beta})$$

Maximum likelihood estimator (MLE)

- If the errors are iid and normally distributed, then

$$\boldsymbol{Y} \sim N(\boldsymbol{X}\beta, \sigma_\varepsilon^2 \boldsymbol{I}).$$

- The likelihood function is

$$L(\beta) = \frac{1}{\sigma^n (2\pi)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}(\boldsymbol{Y} - \boldsymbol{X}\beta)'(\boldsymbol{Y} - \boldsymbol{X}\beta)\right)$$

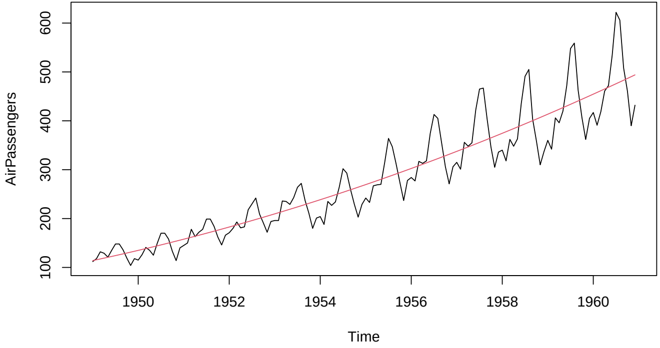
which is maximized when $Q(\beta)$ is minimized.
- So, **MLE = OLS** under the normality assumption.
- Moreover, $\hat{\beta}$ is asymptotically normally distributed in the sense:

$$\sqrt{n}(\hat{\beta} - \beta) \approx N(0, \sigma_\varepsilon^2 C), \quad C = (\boldsymbol{X}'\boldsymbol{X})^{-1}$$

AirPassengers data

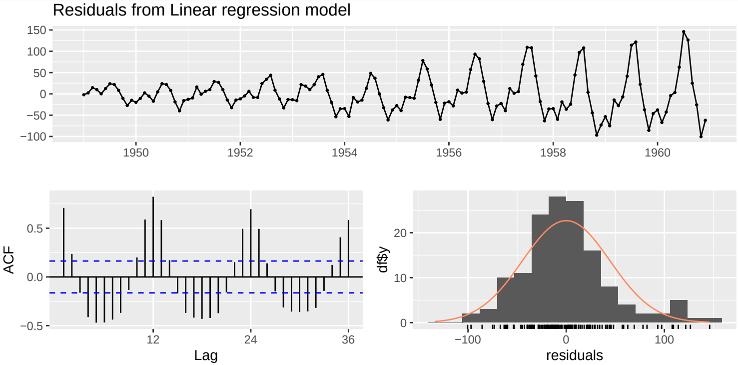
A naive regression approach, however, cannot afford to explain oscillations by seasonal effects and heterogeneity of variance.

```
library(forecast)
time=time(AirPassengers)
fit=tslm(AirPassengers~time+I(time^2))
ts.plot(AirPassengers)
lines(fitted(fit),col=2)
```



AirPassengers data

```r
library(fpp2)
checkresiduals(fit)
```

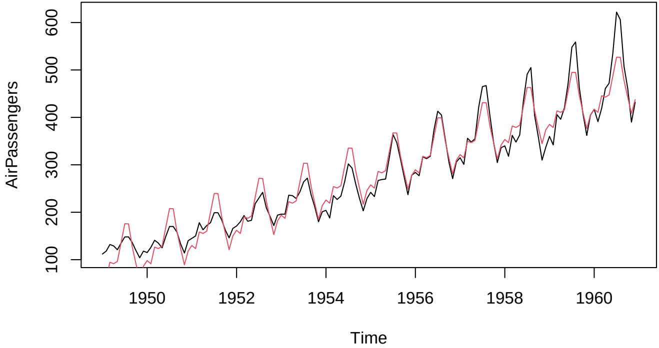



```
##
##  Breusch-Godfrey test for serial correlation of order up to 24
##
## data:  Residuals from Linear regression model
## LM test = 137.86, df = 24, p-value < 2.2e-16
```

AirPassengers data

When `month` is added in the model, the fit becomes slightly better.

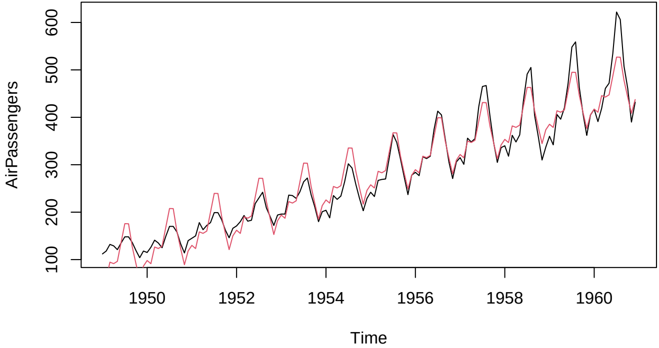
```
library(tidyverse)
month = AirPassengers %>% cycle %>% as.factor
fit=tslm(AirPassengers~time+month)
ts.plot(AirPassengers)
lines(fitted(fit),col=2)
```



AirPassengers data

- TS linear regression can be implemented by calling the `tslm` function in the `forecast` library.
- Here, `trend` is a time variable and `season` is a dummy variable for seasonal effect.

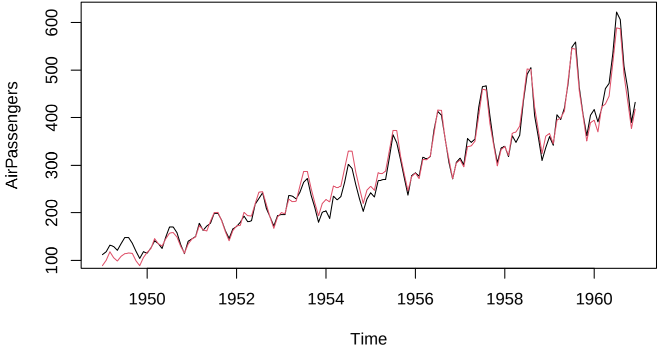
```
library(forecast)
fit=tslm(AirPassengers ~ trend+season)
ts.plot(AirPassengers)
lines(fitted(fit),col=2)
```



AirPassengers data

- Inclusion of the interaction term seems to improve the fit very much. Notice that additive model vs. multiplicative model.

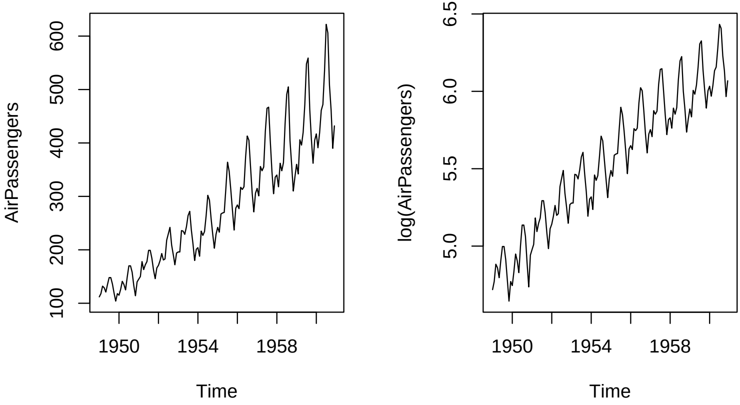
```
fit=tslm(AirPassengers ~ trend*season)
ts.plot(AirPassengers)
lines(fitted(fit),col=2)
```



Variance stabilization

- Sometimes, it is very helpful to take some transformation of the time series variable for variace stabilization.

```
par(mfrow=c(1,2))
ts.plot(AirPassengers)
ts.plot(log(AirPassengers))
```



Power transformation

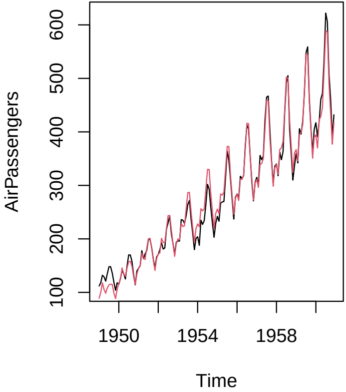
- **Box-Cox transformation** is a family of functions applied to create a monotonic transformation of data using power functions.

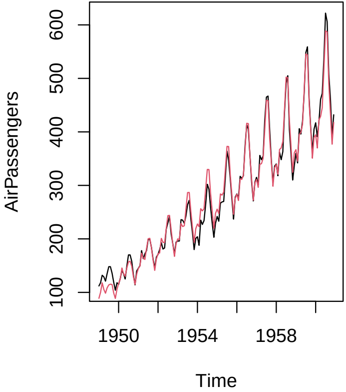
$$y^{(\lambda)} = \begin{cases} \frac{y^\lambda - 1}{\lambda} & \text{if } \lambda \neq 0 \\ \log(y) & \text{if } \lambda = 0 \end{cases}$$

- It is a data transformation technique used to stabilize variance, make the data more normal distribution-like, improve the validity of measures of association.
- You might take a log-transformation by setting `lambda=0`:

Power transformation

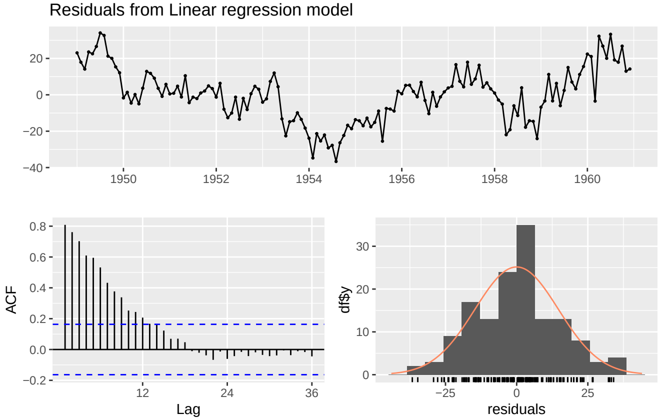
```
fit1=tslm(AirPassengers ~ trend*season)
lambda = BoxCox.lambda(AirPassengers)
fit2=tslm(AirPassengers ~ trend*season, lamabda = lambda)
par(mfrow=c(1,2))
ts.plot(AirPassengers)
lines(fitted(fit1),col=2)
ts.plot(AirPassengers)
lines(fitted(fit2),col=2)
```





Power transformation

```r
checkresiduals(fit2)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 27
##
## data:  Residuals from Linear regression model
## LM test = 113.25, df = 27, p-value = 1.558e-12
```

US consumption data

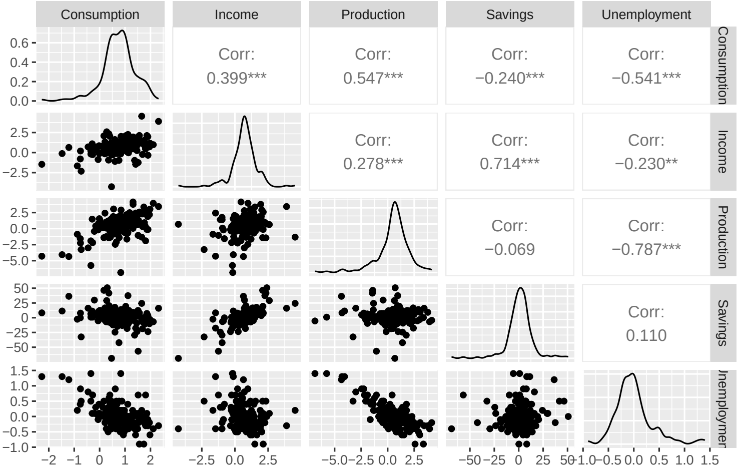
```r
library(fpp2)
autoplot(uschange[,c("Consumption","Income")]) +
  ylab("% change") + xlab("Year")
```

US consumption data

```
tslm(Consumption ~ Income, data=uschange) %>% summary
##
## Call:
## tslm(formula = Consumption ~ Income, data = uschange)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.40845 -0.31816  0.02558  0.29978  1.45157
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.54510    0.05569   9.789  < 2e-16 ***
## Income       0.28060    0.04744   5.915 1.58e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6026 on 185 degrees of freedom
## Multiple R-squared:  0.159,  Adjusted R-squared:  0.1545
## F-statistic: 34.98 on 1 and 185 DF,  p-value: 1.577e-08
```

US consumption data

```r
uschange %>% as.data.frame %>% GGally::ggpairs()
```

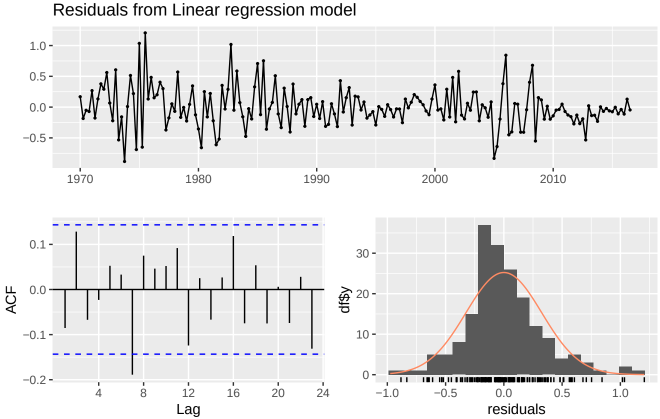


US consumption data

```r
fit.consMR <- tslm(
  Consumption ~ Income + Production + Unemployment + Savings, data=uschange)
summary(fit.consMR)
##
## Call:
## tslm(formula = Consumption ~ Income + Production + Unemployment +
##     Savings, data = uschange)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.88296 -0.17638 -0.03679  0.15251  1.20553
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.26729    0.03721   7.184 1.68e-11 ***
## Income        0.71449    0.04219  16.934  < 2e-16 ***
## Production    0.04589    0.02588   1.773   0.0778 .
## Unemployment -0.20477    0.10550  -1.941   0.0538 .
## Savings      -0.04527    0.00278 -16.287  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3286 on 182 degrees of freedom
## Multiple R-squared:  0.754,  Adjusted R-squared:  0.7486
## F-statistic: 139.5 on 4 and 182 DF,  p-value: < 2.2e-16
```

US consumption data

```r
checkresiduals(fit.consMR)
```




```
##
##  Breusch-Godfrey test for serial correlation of order up to 8
##
## data:  Residuals from Linear regression model
## LM test = 14.874, df = 8, p-value = 0.06163
```

Residual diagnostics

- For forecasting purposes, we require the following assumptions:
  - ε_t are uncorrelated and zero mean
  - ε_t are uncorrelated with each $x_{j,t}$.
- It is **useful** to also have $\varepsilon_t \sim N(0, \sigma^2)$ when producing prediction intervals or doing statistical tests.
- Useful for spotting outliers and whether the linear model was appropriate.
  - Scatterplot of residuals ε_t against each predictor $x_{j,t}$.
  - Scatterplot residuals against the fitted values $\hat{y}_t$
  - Expect to see scatterplots resembling a horizontal band with no values too far from the band and no patterns such as curvature or increasing spread.

Residual diagnostics

- If a plot of the residuals vs any predictor in the model shows a pattern, then the relationship is nonlinear.
- If a plot of the residuals vs any predictor **not** in the model shows a pattern, then the predictor should be added to the model.
- If a plot of the residuals vs fitted values shows a pattern, then there is heteroscedasticity in the errors. (Could try a transformation.)

Durbin-Watson statistic

- The **Durbin–Watson statistic** is a test statistic used to detect the presence of autocorrelation at lag 1 in the residuals (prediction errors) from a regression analysis.
- It tests $H_0 : \phi = 0$ in the AR(1) model: $e_t = \phi e_{t-1} + \nu_t$.
- If e_t is the residual, the Durbin-Watson test statistic is

$$d = \frac{\sum_{t=2}^{n}(e_t - e_{t-1})^2}{\sum_{t=1}^{n} e_t^2},$$

  where n is the number of observations.
- For large n, d is approximately equal to $2(1-\hat{\rho})$, where $\hat{\rho}$ is the sample autocorrelation of the residuals, $d = 2$ therefore indicates no autocorrelation.
- The value of d always lies between 0 and 4.

Durbin-Watson statistic

To test for positive autocorrelation at significance α, the test statistic d is compared to lower and upper critical values.

- If $d < d_{L,\alpha}$, there is statistical evidence that the error terms are positively autocorrelated.
- If $d > d_{U,\alpha}$, there is no statistical evidence that the error terms are positively autocorrelated.
- If $d_{L,\alpha} < d < d_{U,\alpha}$, the test is inconclusive.

Positive serial correlation is serial correlation in which a positive error for one observation increases the chances of a positive error for another observation.

Durbin-Watson statistic

To test for negative autocorrelation at significance α, the test statistic $(4 - d)$ is compared to lower and upper critical values:

- If $(4 - d) < d_{L,\alpha}$, there is statistical evidence that the error terms are negatively autocorrelated.
- If $(4 - d) > d_{U,\alpha}$, there is no statistical evidence that the error terms are negatively autocorrelated.
- If $d_{L,\alpha} < (4 - d) < d_{U,\alpha}$, the test is inconclusive.

Negative serial correlation implies that a positive error for one observation increases the chance of a negative error for another observation.

Ljung–Box and Breusch-Godfrey tests

- The Ljung–Box or Breusch-Godfrey test is a type of statistical test of whether any of a group of autocorrelations of a time series are different from zero.
- If R^2 statistic is calculated, then

$$(n-p)R^2 \sim \chi^2_p,$$

when there is no serial correlation up to lag p, and T, length of series.

- Breusch-Godfrey test better than Ljung-Box for regression models.

```
checkresiduals(fit.consMR, plot=FALSE)
##
##  Breusch-Godfrey test for serial correlation of order up to 8
##
## data:  Residuals from Linear regression model
## LM test = 14.874, df = 8, p-value = 0.06163
```

Outline

Harmonic regression

- Joseph Fourier (1768-1830) showed that

$$\{1, \cos x, \cos 2x, \cos 3x, \ldots, \sin x, \sin 2x, \ldots\}$$

forms a basis for $L^2(-\pi, \pi]$, hence f in $L^2(-\pi, \pi]$ can be represented as

$$f(x) = a_0 + \sum_{n=1}^{\infty} (a_n \cos(nx) + b_n \sin(nx))$$

Harmonic regression

- Based on this theory, we will consider a finite order approximation of s_t:

$$s_t = a_0 + \sum_{j=1}^{k} \left(a_j \cos\left(\lambda_j t\right) + b_j \sin\left(\lambda_j t\right)\right)$$

where $a_0, a_1, ..., a_k$ and $b_1, ..., b_k$ are unknown parameters and $\lambda_1, ..., \lambda_k$ are fixed frequencies, each being some integer multiple of $2\pi/d$.

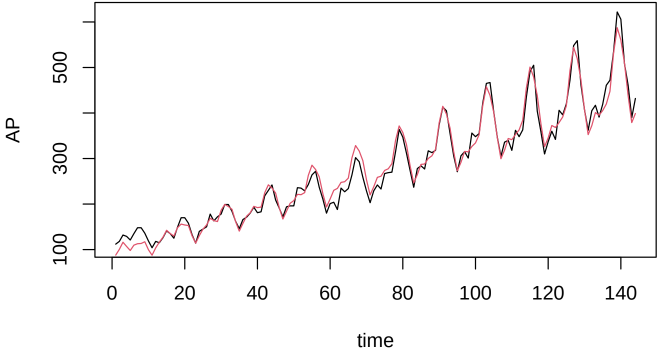
Harmonic regression

- Once k, the number of basis, and corresponding λ_j is selected, we can simply apply OLS to get estimates of coefficients.
- We will assume that k is known. Otherwise, as in the regression, you can apply variable selection to choose k. In practice $k = 1, ..., 4$.
- How to choose λ_j ?
  1. Set $f_1 = [n/d]$. This is a number of cycles that s_t repeated in the data. Take $f_j = jf_1$.
  2. $\lambda_j = f_j(2\pi/n)$
- For example if $n = 72$ and $d = 12$,

$$f_1 = [72/12] = 6, \quad \lambda_j = j \times 6 \times 2\pi/72$$

AirPassengers data

```r
L=12
AP = as.numeric(AirPassengers); time = 1:length(AP)
sin1=sin(2*pi*time/L); sin2=sin(2*pi*time/L*2); sin4=sin(2*pi*time/L*4)
cos1=cos(2*pi*time/L); cos2=cos(2*pi*time/L*2); cos4=cos(2*pi*time/L*4)
fit=lm(AP~time*(sin1+cos1+sin2+cos2+sin4+cos4))
plot(AP~time, type="l")
lines(time,predict(fit),col=2)
```




```r
fit=tslm(AirPassengers ~ trend * fourier(AirPassengers, K=2))
```

Outline

Linear Models with Time Series

Harmonic Regression

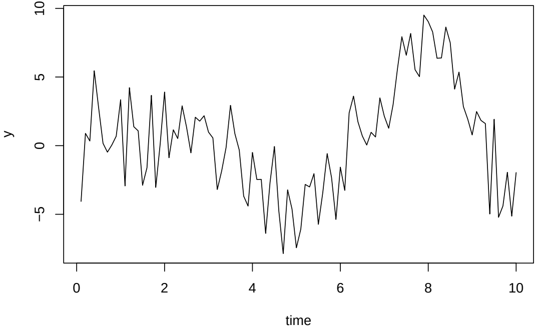
Splines for Nonlinear Trend

Model Selection and Forecasting

Simple spline examples

- Consider the following times series data with non-linear trend.

```
time = c(1:100)/10
y = time * sin(time) + rnorm(100,sd=2)
plot(y ~ time, type="l")
```

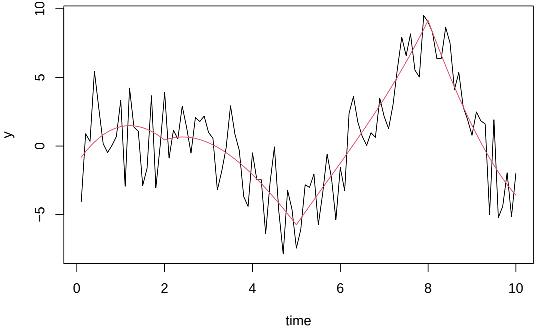


- Clearly, a linear regression is not a good choice to take off the non-linear trend.
- We may use a spline regression with (2, 5, 8) as knot points.

Simple spline examples

- We may use a linear spline regression with (2, 5, 8) as knot points.

```
x1 = time; x2 = time^2; x3 = time^3
z1 = pmax(time, 2); z2 = pmax(time, 5); z3 = pmax(time, 8)
fit = lm(y ~ x1+x2+x3+z1+z2+z3)
plot(y ~ time, type="l")
lines(time,predict(fit),col=2)
```



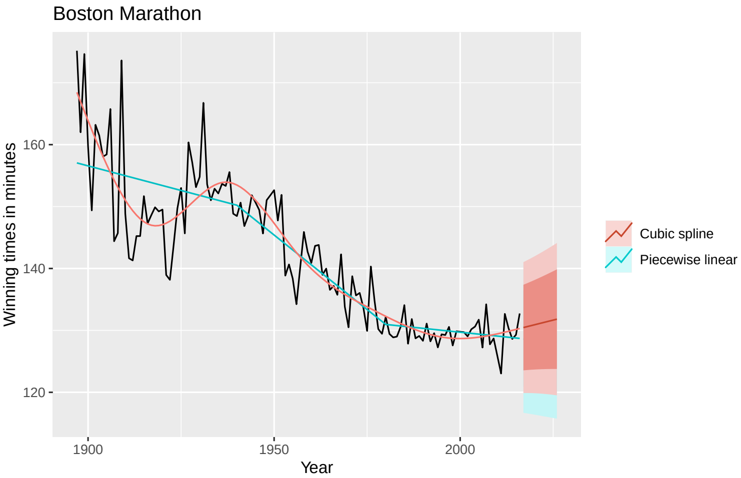
Interpolating splines for non-linear trend

- A spline is a continuous function $f(x)$ interpolating all points (κ_j, y_j) for $j = 1, \ldots, K$ and consisting of polynomials between each consecutive pair of 'knots' κ_j and κ_{j+1}.
- Parameters constrained so that $f(x)$ is continuous.
- Further constraints imposed to give continuous derivatives.
- For example, we can use a natural spline as follows:
  - Let $\kappa_1 < \kappa_2 < \cdots < \kappa_K$ be **knots** in interval (a, b).
  - Let $x_1 = x$, $x_j = (x - \kappa_{j-1})_+$ for $j = 2, \ldots, K+1$.
  - Then the regression is piecewise linear with bends at the knots.
  - Let $x_1 = x$, $x_2 = x^2$, $x_3 = x^3$, $x_j = (x - \kappa_{j-3})_+^3$ for $j = 4, \ldots, K+3$.
  - Then the regression is piecewise cubic, but smooth at the knots.

Boston marathon winning times

```
library(splines)
t <- time(marathon)
fit.splines <- lm(marathon ~ ns(t, df=6))
summary(fit.splines)
##
## Call:
## lm(formula = marathon ~ ns(t, df = 6))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.0028  -2.5722   0.0122   2.1242  21.5681
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      168.447      2.086  80.743  < 2e-16 ***
## ns(t, df = 6)1    -6.948      2.688  -2.584    0.011 *
## ns(t, df = 6)2   -28.856      3.416  -8.448 1.16e-13 ***
## ns(t, df = 6)3   -35.081      3.045 -11.522  < 2e-16 ***
## ns(t, df = 6)4   -32.563      2.652 -12.279  < 2e-16 ***
## ns(t, df = 6)5   -64.847      5.322 -12.184  < 2e-16 ***
## ns(t, df = 6)6   -21.002      2.403  -8.741 2.46e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

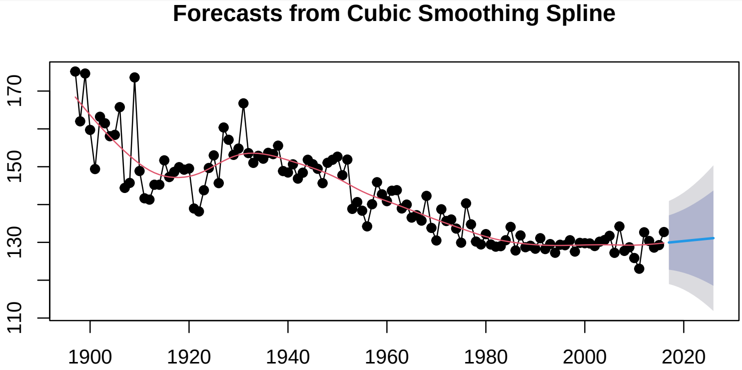
Boston marathon winning times



Spline forecasting with `splinef`

A slightly different type of spline is provided by `splinef`

```
fc = splinef(marathon)
plot(fc)
```



Spline forecasting with `splinef`

- Cubic **smoothing** splines (rather than cubic regression splines).
- Still piecewise cubic, but with many more knots (one at each observation).
- Coefficients constrained to prevent the curve becoming too "wiggly".
- Degrees of freedom selected automatically.
- Equivalent to ARIMA(0,2,2) and Holt's method.

Outline

Linear Models with Time Series

Harmonic Regression

Splines for Nonlinear Trend

Model Selection and Forecasting

Comparing regression models

Computer output for regression will always give the R^2 value. This is a useful summary of the model.

- It is equal to the square of the correlation between y and $\hat{y}$.
- It is often called the "coefficient of determination".
- It can also be calculated as follows:

$$R^2 = \frac{\sum(\hat{y}_t - \bar{y})^2}{\sum(y_t - \bar{y})^2}$$

- It is the proportion of variance accounted for (explained) by the predictors.

Comparing regression models

However,

- R^2 does not allow for "degrees of freedom".
- Adding *any* variable tends to increase the value of R^2, even if that variable is irrelevant.

To overcome this problem, we can use *adjusted* R^2:

$$\bar{R}^2 = 1 - (1 - R^2)\frac{n-1}{n-k-1}$$

where $k =$ no. predictors and $n =$ no. observations.

Cross-validation (CV)

Cross-validation for regression

(Assuming future predictors are known)

- Select one observation for test set, and use *remaining* observations in training set. Compute error on test observation.
- Repeat using each possible observation as the test set.
- Compute accuracy measure over all errors.

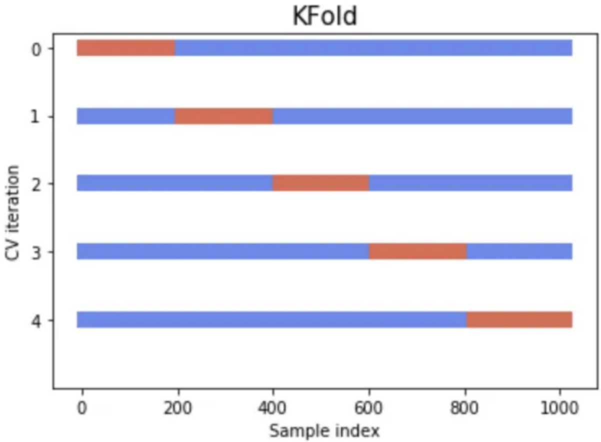
Cross-validation (CV)

Cross-validation:

1. Split randomly data in train and test set.
2. Focus on train set and split it again randomly in chunks (called folds).
3. Let's say you got 5 folds; train on 4 of them and test on the 5th.
4. Repeat step three 5 times to get 5 accuracy measures on 5 different and separate folds.
5. Compute the average of the 5 accuracies which is the final reliable number telling us how the model is performing.

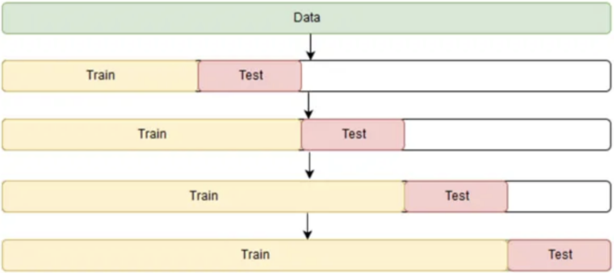
The best model is the one with minimum CV.

Conventional CV



Time series CV

- In the case of time series, the cross-validation is not trivial.
- We may use cross-validation on a time-rolling basis.



Akaike's Information Criterion (AIC)

$$\text{AIC} = -2\log(L) + 2(k+2)$$

where L is the likelihood and k is the number of predictors in the model.

- This is a *penalized likelihood* approach.
- *Minimizing* the AIC gives the best model for prediction.
- AIC penalizes terms more heavily than $\bar{R}^2$.
- Minimizing the AIC is asymptotically equivalent to minimizing MSE via leave-one-out cross-validation.

Corrected AIC (AICc)

For small values of n, the AIC tends to select too many predictors, and so a bias-corrected version of the AIC has been developed.

$$\text{AIC}_{\text{C}} = \text{AIC} + \frac{2(k+2)(k+3)}{n-k-3}$$

As with the AIC, the AIC_{C} should be minimized.

Bayesian Information Criterion (BIC)

$$\text{BIC} = -2\log(L) + (k+2)\log(n)$$

where L is the likelihood and k is the number of predictors in the model.

- BIC penalizes terms more heavily than AIC
- Also called SBIC and SC.
- Minimizing BIC is asymptotically equivalent to leave-v-out cross-validation when $v = n[1 - 1/(\log(n) - 1)]$.

Choosing informative regression variables

Best subsets regression

- Fit all possible regression models using one or more of the predictors.
- Choose the best model based on one of the measures of predictive ability (CV, AIC, AICc).

Warning!

- If there are a large number of predictors, this is not possible.
- For example, 44 predictors leads to 18 trillion possible models!

Performance metrics

We may also consider the mean square error (MSE), root-mean-square error (RMSE), and mean absolute percentage error (MAPE), to evaluate the model's performance:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

$$\text{MAPE}(\%) = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100$$

Model selection

```
tslm(Consumption ~ Income + Production + Unemployment + Savings,
  data=uschange) %>% CV()
##          CV          AIC         AICc          BIC        AdjR2
##   0.1163477 -409.2980298 -408.8313631 -389.9113781    0.7485856
tslm(Consumption ~ Income + Production + Unemployment,
  data=uschange) %>% CV()
##          CV          AIC         AICc          BIC        AdjR2
##   0.2776928 -243.1635677 -242.8320760 -227.0080246    0.3855438
tslm(Consumption ~ Income + Production + Savings,
  data=uschange) %>% CV()
##          CV          AIC         AICc          BIC        AdjR2
##   0.1178681 -407.4669279 -407.1354362 -391.3113848    0.7447840
tslm(Consumption ~ Income + Unemployment + Savings,
  data=uschange) %>% CV()
##          CV          AIC         AICc          BIC        AdjR2
##   0.1160223 -408.0941325 -407.7626408 -391.9385894    0.7456386
tslm(Consumption ~ Production + Unemployment + Savings,
  data=uschange) %>% CV()
##          CV          AIC         AICc          BIC        AdjR2
##   0.2927095 -234.3734580 -234.0419663 -218.2179149    0.3559711
```

Building a predictive regression model

- Assumes possible scenarios for the predictor variables
- Prediction intervals for scenario based forecasts do not include the uncertainty associated with the future values of the predictor variables.
- If getting forecasts of predictors is difficult, you can use lagged predictors instead.

$$Y_t = \beta_0 + \beta_1 X_{1,t-h} + \cdots + \beta_k X_{k,t-h} + \varepsilon_t$$

- A different model for each forecast horizon h.

Regression forecasting

- Optimal forecasts:

$$\hat{y}^* = \mathrm{E}(y^*|\boldsymbol{Y}, \boldsymbol{X}, \boldsymbol{x}^*) = \boldsymbol{x}^*\hat{\beta} = \boldsymbol{x}^*(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y}$$

  where $\boldsymbol{x}^*$ is a row vector containing the values of the predictors for the forecasts (in the same format as $\boldsymbol{X}$).
- Forecast variance:

$$\mathrm{Var}(y^*|\boldsymbol{X}, \boldsymbol{x}^*) = \sigma^2\left[1 + \boldsymbol{x}^*(\boldsymbol{X}'\boldsymbol{X})^{-1}(\boldsymbol{x}^*)'\right]$$

  - This ignores any errors in $\boldsymbol{x}^*$.
  - 95% prediction intervals assuming normal errors:

$$\hat{y}^* \pm 1.96\sqrt{\mathrm{Var}(y^*|\boldsymbol{X}, \boldsymbol{x}^*)}.$$

Regression forecasting

- Fitted values:

$$\hat{\boldsymbol{Y}} = \boldsymbol{X}\hat{\beta} = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{Y}$$

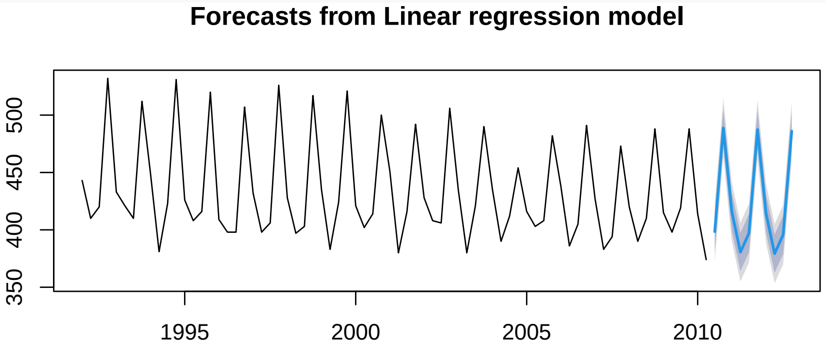
  where $\boldsymbol{H} = \boldsymbol{X}(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'$ is the "hat matrix"
- **Leave-one-out residuals**
- Let $h_1, \ldots, h_n$ be the diagonal values of $\boldsymbol{H}$, then the cross-validation statistic is

$$\text{CV} = \frac{1}{n}\sum_{t=1}^{n}[e_t/(1-h_t)]^2,$$

  where e_t is the residual obtained from fitting the model to all n observations.

Beer production data

```r
beer2 = window(ausbeer, start=1992)
fit.beer = tslm(beer2 ~ trend + season)
fcast = forecast(fit.beer)
plot(fcast)
```

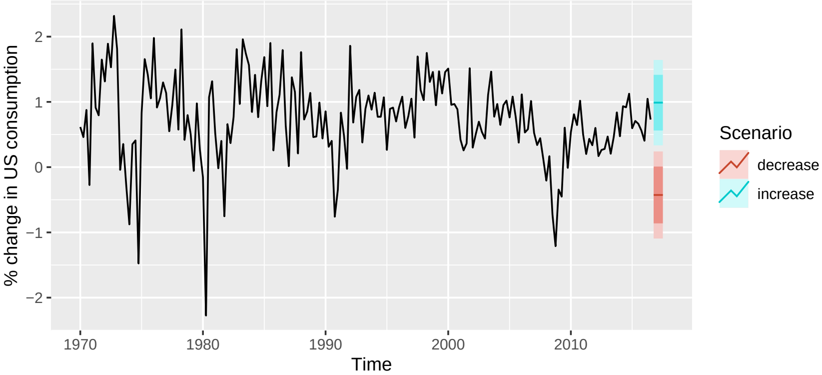


US consumption data

```r
fit.consBest <- tslm(
  Consumption ~ Income + Savings + Unemployment,
  data = uschange)
h <- 4
newdata <- data.frame(
    Income = c(1, 1, 1, 1),
    Savings = c(0.5, 0.5, 0.5, 0.5),
    Unemployment = c(0, 0, 0, 0))
fcast.up <- forecast(fit.consBest, newdata = newdata)
newdata <- data.frame(
    Income = rep(-1, h),
    Savings = rep(-0.5, h),
    Unemployment = rep(0, h))
fcast.down <- forecast(fit.consBest, newdata = newdata)
```

US consumption data

```r
autoplot(uschange[, 1]) +
  ylab("% change in US consumption") +
  autolayer(fcast.up, PI = TRUE, series = "increase") +
  autolayer(fcast.down, PI = TRUE, series = "decrease") +
  guides(colour = guide_legend(title = "Scenario"))
```



Correlation is not causation

- When X is useful for predicting Y, it is not necessarily causing Y.
- e.g., predict number of drownings Y using number of ice-creams sold X.
- Correlations are useful for forecasting, even when there is no causality.
- Better models usually involve causal relationships (e.g., temperature X and people Z to predict drownings Y).

Multicollinearity

In regression analysis, multicollinearity occurs when:

- Two predictors are highly correlated (i.e., the correlation between them is close to ± 1).
- A linear combination of some of the predictors is highly correlated with another predictor.
- A linear combination of one subset of predictors is highly correlated with a linear combination of another subset of predictors.

Multicollinearity

If multicollinearity exists,

- the numerical estimates of coefficients may be wrong (worse in Excel than in a statistics package)
- don't rely on the p-values to determine significance.
- there is no problem with model *predictions* provided the predictors used for forecasting are within the range used for fitting.
- omitting variables can help.
- combining variables can help.

Outliers and influential observations

Things to watch for

- *Outliers*: observations that produce large residuals.
- *Influential observations*: removing them would markedly change the coefficients. (Often outliers in the X variable).
- *Lurking variable*: a predictor not included in the regression but which has an important effect on the response.
- Points should not normally be removed without a good explanation of why they are different.