Economic Forecasting

Advanced forecasting topics

Sebastian Fossati

University of Alberta | E493 | 2023

Outline

Vector autoregressive models

The vector autoregressive (VAR) model is one of the most successful, flexible, and easy to use models for the analysis of multivariate time series.

- it is a natural extension of the univariate autoregressive model to dynamic multivariate time series
- has proven to be useful for describing the dynamic behavior of economic and financial time series
- it often provides superior forecasts to those from univariate time series models and elaborate theory-based models

Vector autoregression models

Stock and Watson (2001) state that macroeconomists do four things with multivariate time series:

- describe and summarize macroeconomic data
- make macroeconomic forecasts
- quantify what we do or do not know about the true structure of the macroeconomy
- advise macroeconomic policymakers

Bivariate VAR(2)

$$\begin{aligned} y_{1,t} &= c_1 + \phi_{11,1} y_{1,t-1} + \phi_{12,1} y_{2,t-1} + \phi_{11,2} y_{1,t-2} + \phi_{12,2} y_{2,t-2} + \varepsilon_{1,t} \\ y_{2,t} &= c_2 + \phi_{21,1} y_{1,t-1} + \phi_{22,1} y_{2,t-1} + \phi_{21,2} y_{1,t-2} + \phi_{22,2} y_{2,t-2} + \varepsilon_{2,t} \end{aligned}$$

- $\phi_{ij,l}$ measures the influence of the l-th lag of y_j on y_i
- $\varepsilon_{1,t}$ and $\varepsilon_{2,t}$ are white noise processes that may be contemporaneously correlated (that is $\mathrm{cov}(\varepsilon_{1,t},\varepsilon_{2,t}) \neq 0$)

Vector autoregression models

Remarks:

- each equation has the same regressors: lagged values of $y_{1,t}$ and $y_{2,t}$
- endogeneity is avoided by using lagged values of $y_{1,t}$ and $y_{2,t}$
- if the series are stationary, we forecast fitting a VAR to the data directly (a “VAR in levels”)
- if the series are non-stationary, we forecast fitting a VAR to the differenced data (a “VAR in differences”)

How many variables in the VAR?

There are two decisions one has to make when using a VAR to forecast:

1. How many *variables* (denoted by K).
2. How many *lags* (denoted by p) should be included in the system.

How many variables in the VAR?

Remarks:

- the number of coefficients to be estimated in a VAR is $K + pK^2$
- for example, a VAR with $K = 5$ and $p = 3$ has a total of 80 coefficients to be estimated!
- the more coefficients that need to be estimated, the larger the estimation error (less accurate forecasts)
- in practice, keep K small and use information criteria to select the number of lags to be included (BIC preferred)

How many lags in the VAR?

The lag length for the VAR(p) model may be determined using **model selection criteria**. For forecasting with VAR models, we prefer to use the BIC.

$$\text{BIC}(p) = \log\left[\det(\hat{\Sigma}_e)\right] + \frac{\log T}{T}(K + pK^2)$$

- the i, j element of $\hat{\Sigma}_e$ is $1/T \sum_{t=1}^{T} e_{i,t}e_{j,t}$
- $e_{i,t}$ is the residual from the i-th equation and $e_{j,t}$ is the residual from the j-th equation
- $\det(\hat{\Sigma}_e)$ is the determinant of matrix $\hat{\Sigma}_e$
- the AIC can be obtained by replacing "$\log T$" with "2"

`vars` package in R

VAR models are implemented in the `vars` package in R.

- `VARselect()` for selecting the number of lags p using different information criteria ("SC" refers to "BIC")
- `VAR()` for fitting VAR models
- `serial.test()` for residual tests

Example: US growth rate of GDP and the term spread

Consider a bivariate VAR model for the growth rate of GDP and the term spread and the sample 1984Q1 to 2012Q4.

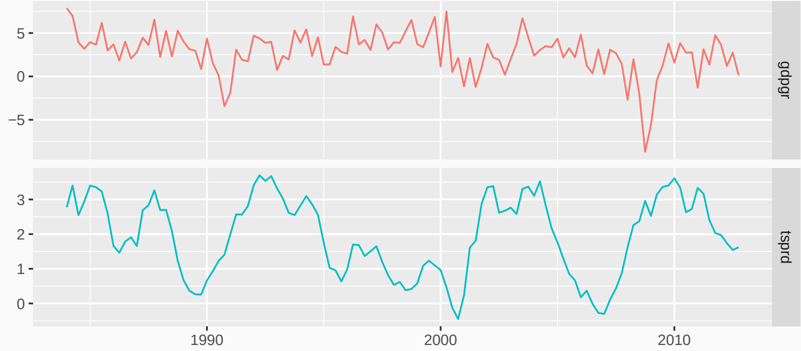
```
# read quarterly data
data <- read.csv("data/us_macro_quarterly.csv", header = TRUE)

# get time series
gdpgr <- 400*diff(log(data$GDPC96))
tsprd <- data$GS10-data$TB3MS
tsprd <- tsprd[-1]

# full sample
vardata0 <- ts(cbind(gdpgr,tsprd), start=c(1957,2), frequency=4)
vardata0 <- window(vardata0, start=c(1984,1), end=c(2012,4))
```

Example: US growth rate of GDP and the term spread

```
# plot data
autoplot(vardata0, facets = TRUE, colour=TRUE) +
  ylab("") + xlab("") + guides(colour="none")
```



Example: US growth rate of GDP and the term spread

We can use `VARselect()` to determine the number of lags p using information criteria.

```
# find optimal number of lags
VARselect(vardata0, lag.max = 8, type="const")[["selection"]]
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      2      2      2      2
```

Example: US growth rate of GDP and the term spread

Now estimate a VAR(2) and test for white noise residuals.

```
# estimate VAR(2)
model0 <- VAR(vardata0, p = 2, type = "const")
serial.test(model0, lags.pt = 10, type = "PT.asymptotic")
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object model0
## Chi-squared = 39, df = 32, p-value = 0.2
```

- the residuals appear to be white noise

Example: US growth rate of GDP and the term spread

The output from a VAR model can be very long…

```
## VAR Estimation Results:
## =========================
## Endogenous variables: gdpgr, tsprd
## Deterministic variables: const
## Sample size: 126
## Log Likelihood: -371.01
## Roots of the characteristic polynomial:
##  0.82  0.82 0.1678 0.1678
## Call:
## VAR(y = vardata0, p = 2, type = "const")
##
```

Example: US growth rate of GDP and the term spread

```
## Estimation results for equation gdpgr:
## ======================================
## gdpgr = gdpgr.l1 + tsprd.l1 + gdpgr.l2 + tsprd.l2 + const
##
##          Estimate Std. Error t value Pr(>|t|)
## gdpgr.l1  0.31422    0.08240   3.813 0.000217 ***
## tsprd.l1 -0.81664    0.39908  -2.046 0.042894 *
## gdpgr.l2  0.24786    0.08038   3.084 0.002534 **
## tsprd.l2  1.12249    0.38871   2.888 0.004598 **
## const     0.63222    0.46786   1.351 0.179119
## ---
## Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
##
##
## Residual standard error: 2.335 on 121 degrees of freedom
## Multiple R-Squared: 0.3296,  Adjusted R-squared: 0.3074
## F-statistic: 14.87 on 4 and 121 DF,  p-value: 6.527e-10
##
```

Example: US growth rate of GDP and the term spread

```
## Estimation results for equation tsprd:
## ======================================
## tsprd = gdpgr.l1 + tsprd.l1 + gdpgr.l2 + tsprd.l2 + const
##
##          Estimate Std. Error t value Pr(>|t|)
## gdpgr.l1  0.01217    0.01753   0.695 0.488682
## tsprd.l1  1.02810    0.08490  12.110  < 2e-16 ***
## gdpgr.l2 -0.05979    0.01710  -3.497 0.000659 ***
## tsprd.l2 -0.19443    0.08269  -2.351 0.020322 *
## const     0.47068    0.09953   4.729 6.16e-06 ***
## ---
## Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
##
##
## Residual standard error: 0.4968 on 121 degrees of freedom
## Multiple R-Squared: 0.8094,  Adjusted R-squared: 0.8031
## F-statistic: 128.5 on 4 and 121 DF,  p-value: < 2.2e-16
##
```

Forecasting

Forecasts are generated from a VAR in a recursive manner.

Bivariate VAR(1)

The one-step-ahead forecasts are:

$$\begin{aligned}
\hat{y}_{1,T+1|T} &= \hat{c}_1 + \hat{\phi}_{11,1} y_{1,T} + \hat{\phi}_{12,1} y_{2,T} \\
\hat{y}_{2,T+1|T} &= \hat{c}_2 + \hat{\phi}_{21,1} y_{1,T} + \hat{\phi}_{22,1} y_{2,T}
\end{aligned}$$

The two-step-ahead forecasts are:

$$\begin{aligned}
\hat{y}_{1,T+2|T} &= \hat{c}_1 + \hat{\phi}_{11,1} \hat{y}_{1,T+1|T} + \hat{\phi}_{12,1} \hat{y}_{2,T+1|T} \\
\hat{y}_{2,T+2|T} &= \hat{c}_2 + \hat{\phi}_{21,1} \hat{y}_{1,T+1|T} + \hat{\phi}_{22,1} \hat{y}_{2,T+1|T}
\end{aligned}$$

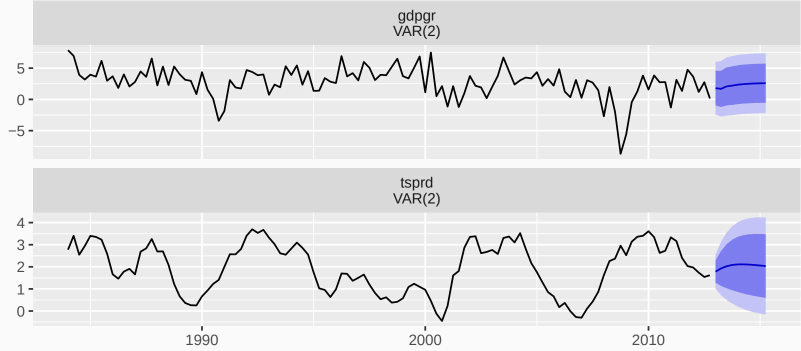
Example: US growth rate of GDP and the term spread

```r
# forecasts
fcast0 <- forecast(model0, h=2)
fcast0
```

```
## gdpgr
##         Point Forecast   Lo 80 Hi 80  Lo 95 Hi 95
## 2013 Q1          1.798 -0.9704 4.566 -2.436 6.031
## 2013 Q2          1.689 -1.2053 4.584 -2.738 6.116
##
## tsprd
##         Point Forecast Lo 80 Hi 80  Lo 95 Hi 95
## 2013 Q1          1.781 1.280 2.282 1.0146 2.548
## 2013 Q2          1.927 1.135 2.719 0.7155 3.138
```

Example: US growth rate of GDP and the term spread

```r
# forecasts
autoplot(forecast(model0, h=10)) + xlab("")
```



Granger causality

One of the main uses of VAR models is forecasting. The following notion of a variable's forecasting ability is due to Granger (1969).

- if a variable y_1 is found to be helpful for predicting another variable y_2 then y_1 is said to *Granger-cause* y_2
- formally, y_1 fails to Granger-cause y_2 if for all $s > 0$ the MSE of a forecast of $y_{2,T+s}$ based on $(y_{2,T}, y_{2,T-1}, \ldots)$ is the same as the MSE of a forecast of $y_{2,T+s}$ based on $(y_{2,T}, y_{2,T-1}, \ldots)$ and $(y_{1,T}, y_{1,T-1}, \ldots)$
- the notion of Granger causality does not imply true causality, only implies forecasting ability

Bivariate VAR(2)

In a bivariate VAR(1), y_2 fails to Granger-cause y_1 if:

$$\begin{aligned} y_{1,t} &= c_1 + \phi_{11,1} y_{1,t-1} + \varepsilon_{1,t} \\ y_{2,t} &= c_2 + \phi_{21,1} y_{1,t-1} + \phi_{22,1} y_{2,t-1} + \varepsilon_{2,t} \end{aligned}$$

Example: US growth rate of GDP and the term spread

```r
# test for Granger causality
causality(model0,cause="tsprd")$Granger
```

```
##
##  Granger causality H0: tsprd do not Granger-cause gdpgr
##
## data:  VAR object model0
## F-Test = 0.49, df1 = 2, df2 = 218, p-value = 0.6
```

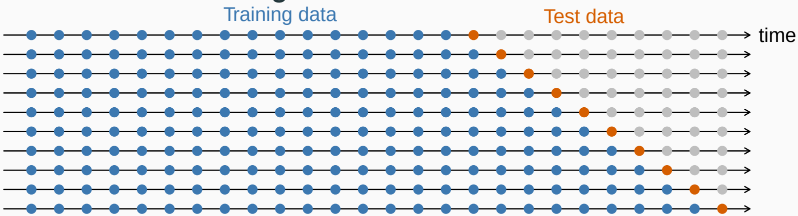
```r
causality(model0,cause="gdpgr")$Granger
```

```
##
##  Granger causality H0: gdpgr do not Granger-cause tsprd
##
## data:  VAR object model0
## F-Test = 6.1, df1 = 2, df2 = 218, p-value = 0.003
```

Outline

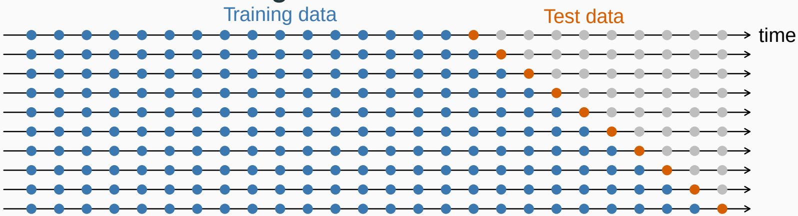
Recursive estimation

Models estimated using recursive estimation



Recursive estimation

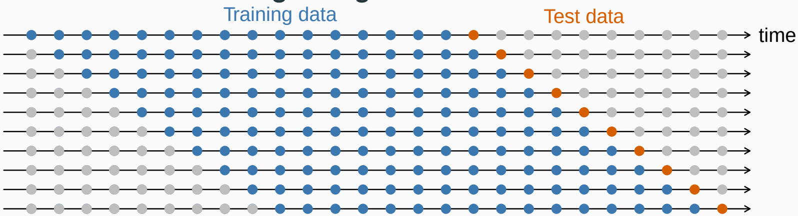
Models estimated using recursive estimation



- models estimated using an expanding window (that is, using all available data to estimate parameters)
- if DGP is stable (no structural breaks), increasing the sample size reduces the variance of the parameter estimates

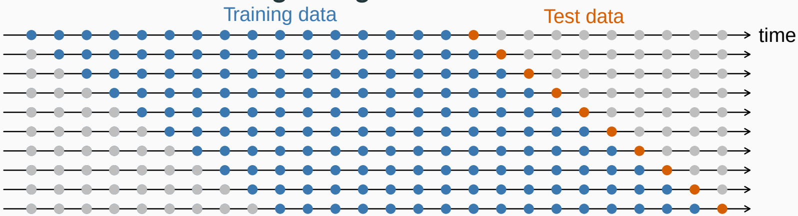
Rolling estimation

Models estimated using rolling estimation



Rolling estimation

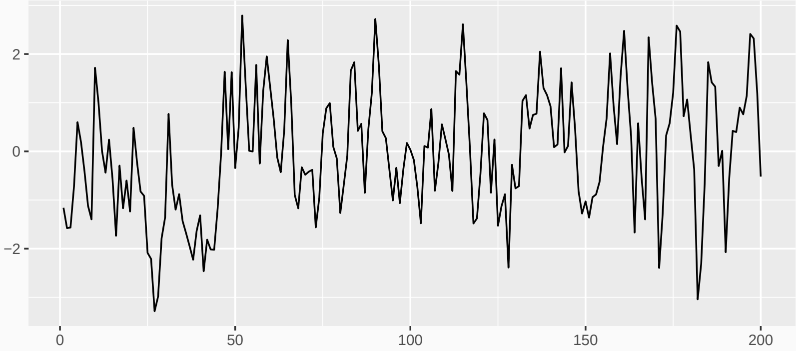
Models estimated using rolling estimation



- models estimated using a fixed window (that is, using only last R observations to estimate parameters)
- if DGP is unstable (structural breaks), using earliest data may lead to biased parameter estimates and forecasts

Example 1: no structural breaks

```r
# simulate data
y1 <- arima.sim(list(ar = 0.5), n = 200)
autoplot(y1) + xlab("") + ylab("")
```



Example 1: no structural breaks

```
n.end <- 50
for(i in 1:150){
  # recursive forecast
  tmp0 <- 1
  tmp1 <- n.end+(i-1)
  tmp <- window(y1,tmp0,tmp1)
  pred.rec[i,1] <- tmp1-tmp0+1 # obs for estimation
  pred.rec[i,2] <- window(y1,tmp1+1,tmp1+1) # actual
  pred.rec[i,3] <- forecast(Arima(tmp, order=c(1,0,0)), h=1)$mean
  # rolling forecast
  tmp0 <- i
  tmp1 <- n.end+(i-1)
  tmp <- window(y1,tmp0,tmp1)
  pred.rol[i,1] <- tmp1-tmp0+1 # obs for estimation
  pred.rol[i,2] <- window(y1,tmp1+1,tmp1+1) # actual
  pred.rol[i,3] <- forecast(Arima(tmp, order=c(1,0,0)), h=1)$mean
}
```

Example 1: no structural breaks

Rolling estimation implies using only the last R observations to estimate the models (a fixed window) while recursive uses all available data.

```
# recursive vs rolling
cbind(pred.rec[,1],pred.rol[,1])[1:5,]
```

```
##      [,1] [,2]
## [1,]   50   50
## [2,]   51   50
## [3,]   52   50
## [4,]   53   50
## [5,]   54   50
```

Example 1: no structural breaks

Example 1: no structural breaks

Compute RMSE.

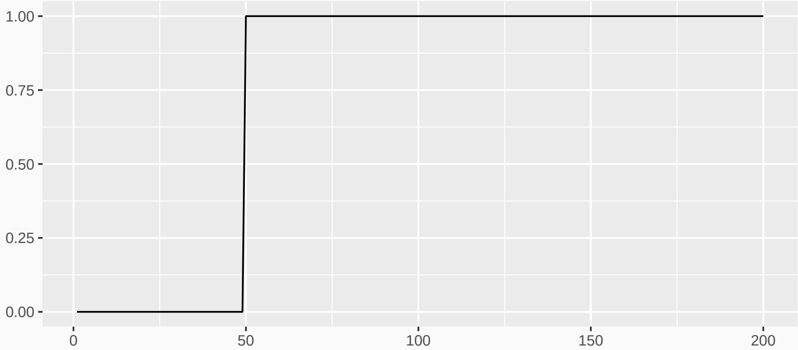
```r
# display rmse
fit.rec <- pred.rec[,2] - pred.rec[,3]
fit.rol <- pred.rol[,2] - pred.rol[,3]
rmse.rec <- sqrt(mean(fit.rec^2, na.rm=TRUE))
rmse.rol <- sqrt(mean(fit.rol^2, na.rm=TRUE))
cbind(rmse.rec,rmse.rol)
```

```
##      rmse.rec rmse.rol
## [1,]    1.043    1.047
```

- recursive slightly better but no substantial difference (why?)

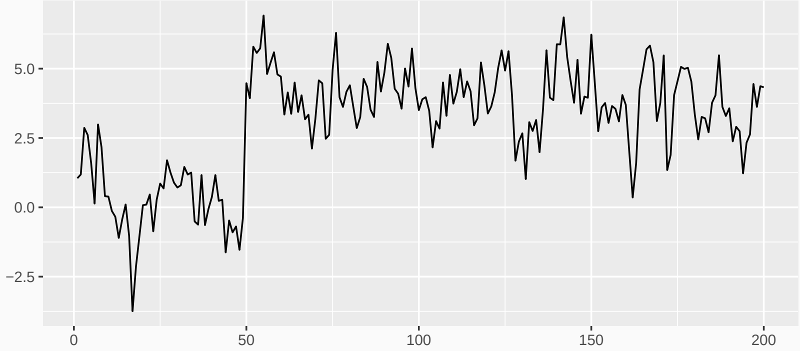
Example 2: one structural break

We can simulate data with a structural break.



Example 2: one structural break

```
# data with one break at t = 100
y1 <- 4*ls + arima.sim(list(ar = 0.5), n = 200)
autoplot(y1) + xlab("") + ylab("")
```



Example 2: one structural break

```r
# pre-break model
(Arima(window(y1,1,100), order=c(1,0,0)))
```

```
## Series: window(y1, 1, 100)
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1    mean
##        0.865   2.312
## s.e.   0.048   0.797
##
## sigma^2 = 1.34:  log likelihood = -156.2
## AIC=318.4   AICc=318.6   BIC=326.2
```

Example 2: one structural break

```r
# post-break model
(Arima(window(y1,101,200), order=c(1,0,0)))
```

```
## Series: window(y1, 101, 200)
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1   mean
##        0.513  3.806
## s.e.   0.085  0.214
##
## sigma^2 = 1.13:  log likelihood = -146.9
## AIC=299.9   AICc=300.1   BIC=307.7
```

Example 2: one structural break

```r
# full sample model
(Arima(y1, order=c(1,0,0)))
```

```
## Series: y1
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1    mean
##         0.82   3.042
## s.e.    0.04   0.440
##
## sigma^2 = 1.32:  log likelihood = -310.9
## AIC=627.7   AICc=627.8   BIC=637.6
```

Example 2: one structural break

```
n.end <- 50
for(i in 1:150){
  # recursive forecast
  tmp0 <- 1
  tmp1 <- n.end+(i-1)
  tmp <- window(y1,tmp0,tmp1)
  pred.rec[i,1] <- tmp1-tmp0+1 # obs for estimation
  pred.rec[i,2] <- window(y1,tmp1+1,tmp1+1) # actual
  pred.rec[i,3] <- forecast(Arima(tmp, order=c(1,0,0)), h=1)$mean
  # rolling forecast
  tmp0 <- i
  tmp1 <- n.end+(i-1)
  tmp <- window(y1,tmp0,tmp1)
  pred.rol[i,1] <- tmp1-tmp0+1 # obs for estimation
  pred.rol[i,2] <- window(y1,tmp1+1,tmp1+1) # actual
  pred.rol[i,3] <- forecast(Arima(tmp, order=c(1,0,0)), h=1)$mean
}
```

Example 2: one structural break

Rolling estimation implies using only the last R observations to estimate the models (a fixed window) while recursive uses all available data.

```r
# recursive vs rolling
cbind(pred.rec[,1],pred.rol[,1])[1:5,]
```

```
##      [,1] [,2]
## [1,]   50   50
## [2,]   51   50
## [3,]   52   50
## [4,]   53   50
## [5,]   54   50
```

Example 2: one structural break

Example 2: one structural break

Compute RMSE.

```r
# display rmse
fit.rec <- pred.rec[,2] - pred.rec[,3]
fit.rol <- pred.rol[,2] - pred.rol[,3]
rmse.rec <- sqrt(mean(fit.rec^2, na.rm=TRUE))
rmse.rol <- sqrt(mean(fit.rol^2, na.rm=TRUE))
cbind(rmse.rec,rmse.rol)
```

```
##      rmse.rec rmse.rol
## [1,]    1.152      1.1
```

- rolling now better than recursive (why?)

Outline

Forecast evaluation

Given different forecasts, we may want to ask the following questions:

1. How good is a particular set of forecasts?
2. Is one set of forecasts better than another one?
3. Is it possible to get a better forecast as a combination of various forecasts for the same variable?

Unbiasedness

Forecasts should be *unbiased*, meaning that the expected value of the forecast error should be equal to zero.

Testing unbiasedness

Test $\tau = 0$ in

$$e_{t+h|t} = y_{t+h} - \hat{y}_{t+h|t} = \tau + v_{t+h}, \quad t = R, \ldots, T - h$$

A robust (HAC) test is needed as errors can be autocorrelated.

- that is, on average the forecast should be correct (no systematic errors)

Unbiasedness

In addition, since $y_{t+h} = \hat{y}_{t+h|t} + e_{t+h|t}$ we have that

$$var(y_{t+h}) = var(\hat{y}_{t+h|t}) + var(e_{t+h|t}).$$

- the variance of the variable should be larger than the variance of a (good) forecast

Efficiency

Forecasts should be *efficient*, meaning that the optimal forecast error should be uncorrelated with available information at the time the forecast was made.

Testing efficiency

Test $\gamma = 0$ in

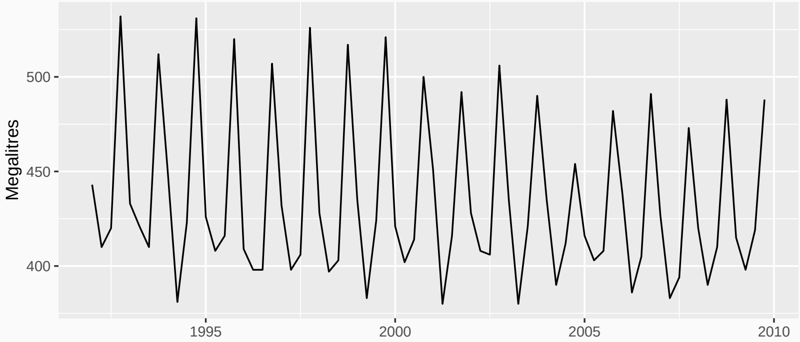
$$e_{t+h|t} = y_{t+h} - \hat{y}_{t+h|t} = \gamma' z_t + v_{t+h}, \qquad t = R, \ldots, T - h$$

A robust (HAC) test is needed as errors can be autocorrelated.

- weak efficiency tests against past prediction errors
- strong efficiency tests against other variables

Example: quarterly beer production

```r
# beer data
beer1 <- window(ausbeer, start=1992,end=c(2009,4))
autoplot(beer1) + xlab("") + ylab("Megalitres")
```

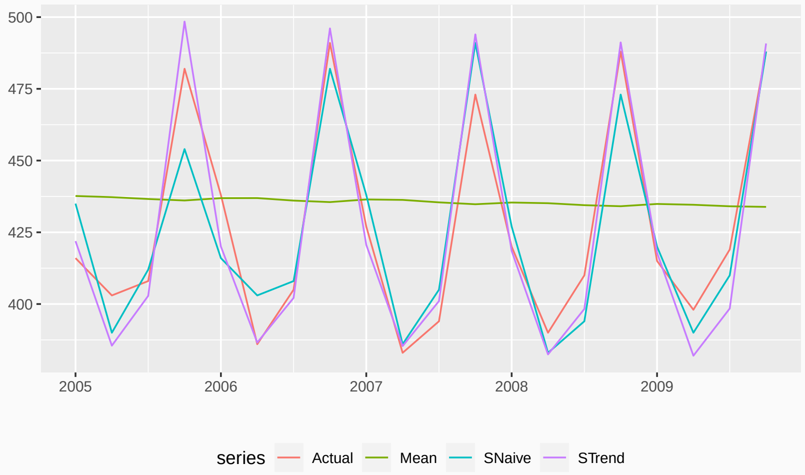


Example: quarterly beer production

```
# end of training set, 2004Q4
n.end <- 2004.75

# set matrix for storage, 20 obs in test set
pred <- matrix(rep(NA,80),20,4)
# loop
for(i in 1:20){
  tmp0 <- 1992
  tmp1 <- n.end+(i-1)*.25
  tmp <- window(beer1,tmp0,tmp1)
  pred[i,1] <- window(beer1,tmp1+.25,tmp1+.25) # actual
  # compute forecasts
  pred[i,2] <- meanf(tmp, h=1)$mean # mean forecast
  pred[i,3] <- snaive(tmp, h=1)$mean # seasonal last value
  pred[i,4] <- forecast(tslm(tmp~trend+season), h=1)$mean # trend + seasonal dummies
}
```

Example: quarterly beer production



Example: quarterly beer production

Test for unbiasedness of mean forecast.

```
beerfit1 <- pred[,1] - pred[,2]
e1 <- ts(beerfit1, start=2005, frequency=4)
# robust tests only need for h>1
test.eq <- tslm(e1~1)
coeftest(test.eq)
```

```
##
## t test of coefficients:
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -8.92       8.33   -1.07      0.3
```

- we fail to reject the null hypothesis and conclude that the forecasts are unbiased

Example: quarterly beer production

Evaluate (weak) efficiency of mean forecast.

```
# robust tests only need for h>1
test.eq <- Arima(e1,order=c(1,0,0))
coeftest(test.eq)
```

```
##
## z test of coefficients:
##
##            Estimate Std. Error z value Pr(>|z|)
## ar1         -0.0773     0.2365   -0.33     0.74
## intercept   -9.1043     7.5635   -1.20     0.23
```

- we fail to reject the null hypothesis and conclude that the forecasts are weakly efficient

Example: quarterly beer production

Evaluate (strong) efficiency of mean forecast.

```
# robust tests only need for h>1
test.eq <- tslm(e1~season)
coeftest(test.eq)
```

```
##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -13.05       3.89   -3.35    0.004 **
## season2       -30.99       5.50   -5.63  3.8e-05 ***
## season3       -15.07       5.50   -2.74    0.015 *
## season4        62.59       5.50   11.37  4.5e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- we reject the null hypothesis and conclude that the forecasts are not efficient (with respect to seasonality)

Example: quarterly beer production

Test for unbiasedness of seasonal naive forecast.

```
beerfit2 <- pred[,1] - pred[,3]
e2 <- ts(beerfit2, start=2005, frequency=4)
# robust tests only need for h>1
test.eq <- tslm(e2~1)
coeftest(test.eq)
```

```
##
## t test of coefficients:
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.45       3.02    0.48     0.64
```

- we fail to reject the null hypothesis and conclude that the forecasts are unbiased

Example: quarterly beer production

Evaluate (weak) efficiency of seasonal naive forecast.

```r
# robust tests only need for h>1
test.eq <- Arima(e2,order=c(1,0,0))
coeftest(test.eq)
```

```
##
## z test of coefficients:
##
##           Estimate Std. Error z value Pr(>|z|)
## ar1          0.104      0.231    0.45     0.65
## intercept    1.324      3.269    0.41     0.69
```

- we fail to reject the null hypothesis and conclude that the forecasts are weakly efficient

Example: quarterly beer production

Evaluate (strong) efficiency of seasonal naive forecast.

```r
# robust tests only need for h>1
test.eq <- tslm(e2~season)
coeftest(test.eq)
```

```
##
## t test of coefficients:
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -4.00       6.31   -0.63     0.53
## season2         5.60       8.92    0.63     0.54
## season3         5.40       8.92    0.61     0.55
## season4        10.80       8.92    1.21     0.24
```

- we fail to reject the null hypothesis and conclude that the forecasts are efficient (with respect to seasonality)

Equal predictive ability

Let $\{y_t\}$ denote the series to be forecast and let $y^1_{t+h|t}$ and $y^2_{t+h|t}$ denote two competing forecasts of y_{t+h} based on I_t.

For example, $y^1_{t+h|t}$ could be from a model that uses the variable x^1_t as predictor and $y^2_{t+h|t}$ from a model that uses x^2_t as predictor.

The forecast errors from the two models are

$$\begin{aligned} e^1_{t+h|t} &= y_{t+h} - y^1_{t+h|t} \\ e^2_{t+h|t} &= y_{t+h} - y^2_{t+h|t} \end{aligned}$$

The $h-$step forecasts are computed for $t = R, \ldots, T-h$ for a total of P forecasts giving $\{e^1_{t+h|t}\}_R^{T-h}$ and $\{e^2_{t+h|t}\}_R^{T-h}$.

Equal predictive ability

The accuracy of each forecast is measured by a particular loss function

$$\mathrm{L}(y_{t+h}, y^i_{t+h|t}) = \mathrm{L}(e^i_{t+h|t}), \quad i = 1, 2$$

Some popular loss functions are:

- quadratic loss: $\mathrm{L}(e^i_{t+h|t}) = (e^i_{t+h|t})^2$
- absolute loss: $\mathrm{L}(e^i_{t+h|t}) = |e^i_{t+h|t}|$

The **null hypothesis of equal predictive ability** is

$$H_0 : \mathrm{E}[\mathrm{L}(e^1_{t+h|t})] = \mathrm{E}[\mathrm{L}(e^2_{t+h|t})].$$

Equal predictive ability

Let $d_t = L(e^1_{t+h|t}) - L(e^2_{t+h|t})$ be the sample loss difference. The **Diebold-Mariano test** statistic is

$$DM = P^{1/2}\bar{d}/s_d$$

where

- $\bar{d}$ is the sample average of d_t, $\bar{d} = P^{-1}\sum_{t=R}^{T-h} d_t$
- s_d^2 is the sample variance of d_t

Diebold and Mariano (1995) show that under the null of equal predictive accuracy $DM \overset{a}{\sim} N(0,1)$ and the null of equal predictive accuracy is rejected at the 5% level if $|DM| > 1.96$.

Equal predictive ability

A robust (HAC) test is needed as the sample of loss differentials $\{d_t\}_R^{T-h}$ are serially correlated for $h > 1$.

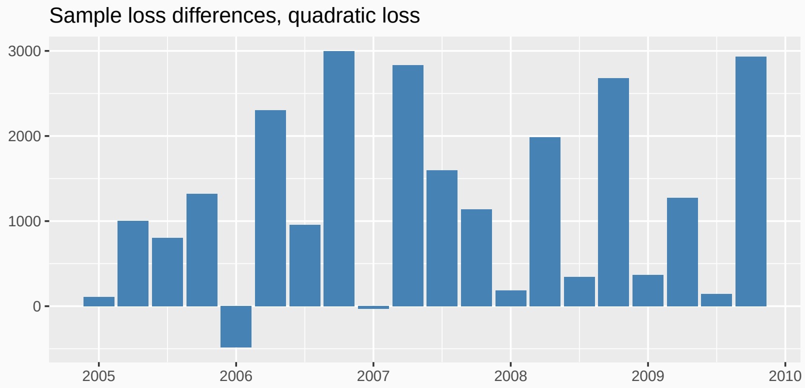
The sample variance of d_t, s_d^2, can be estimated using the Newey-West estimator

$$\hat{s}_d^2 = \hat{\gamma}_0 + 2\sum_{j=1}^{q-1}\left(1 - \frac{j}{q}\right)\hat{\gamma}_j,$$

where

$$\hat{\gamma}_j = P^{-1}\sum_{t=j+1}^{P}(d_t - \bar{d})(d_{t-j} - \bar{d}).$$

Example: quarterly beer production



- sample loss differences (mean forecast – seasonal naive forecast), $d_t = (e^1_{t+h|t})^2 - (e^2_{t+h|t})^2$, are typically > 0

Example: quarterly beer production

Test the hypothesis of equal predictive ability for mean and seasonal naive forecasts using a quadratic loss function.

```r
# compare model 1 (mean) with model 2 (seasonal naive)
# compute Diebold-Mariano statistic
dm.test(beerfit1, beerfit2, h=1, power=2)
```

```
##
##  Diebold-Mariano Test
##
## data:  beerfit1beerfit2
## DM = 5, Forecast horizon = 1, Loss function power = 2, p-
value = 7e-05
## alternative hypothesis: two.sided
```

- we reject the null hypothesis of equal predictive ability, the seasonal naive forecasts are more accurate ($DM > 0$)

Outline

Forecast combination

When alternative forecasts are available, rather than selecting one of them we can combine them.

Let $\hat{y}^1_{t+h|t}$ and $\hat{y}^2_{t+h|t}$ denote two forecasts for the same variable y_{t+h}, with $e^1_{t+h|t}$ and $e^2_{t+h|t}$ the associated forecast errors.

We want to construct the combined forecast

$$\hat{y}^c_{t+h|t} = \alpha \hat{y}^1_{t+h|t} + (1 - \alpha)\hat{y}^2_{t+h|t}$$

or, alternatively,

$$e^c_{t+h|t} = \alpha e^1_{t+h|t} + (1 - \alpha) e^2_{t+h|t}$$

with α is selected to minimize the MSE.

Forecast combination

Since α is not known, it needs to be estimated.

We can obtain an estimate by running the regression

$$y_{t+h} = \alpha \hat{y}^1_{t+h|t} + (1 - \alpha)\hat{y}^2_{t+h|t} + v_t, \quad t = R, \ldots, T - h$$

or, alternatively,

$$e^2_{t+h|t} = \alpha(\hat{y}^1_{t+h|t} - \hat{y}^2_{t+h|t}) + v_t$$

Remarks:

- optimal weights can vary over time
- in the presence of a large number of alternative forecasts $\hat{y}^1_{t+h|t}, \ldots, \hat{y}^M_{t+h|t}$ a simple average tends to work well

Missing values

Missing data can arise for many reasons and the effects on forecasts depend on the specific context:

- in some situations, the missingness may be essentially random (for example, someone may have forgotten to record the sales figures)
- if the timing of the missing data is not informative for the forecasting problem, then the missing values can be handled more easily

Missing values

Remarks:

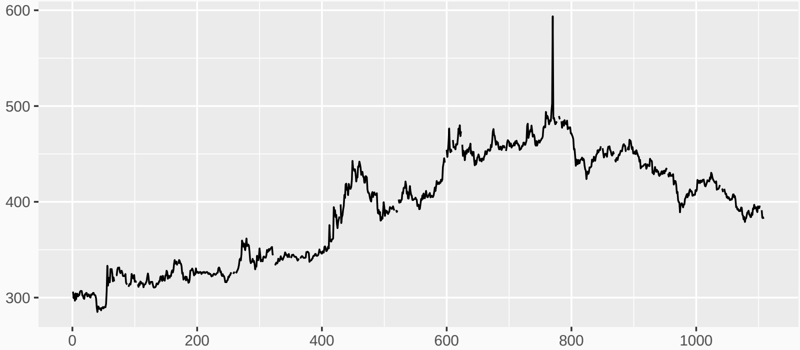
- some functions which can handle missing values (for example: `auto.arima()`, `Arima()`, `tslm()`)
- some cannot handle missing values… (for example, `ets()`, `stl()`, `stlf()`, etc.)

What to do?

1. Model section of data after last missing value.
2. Estimate missing values with `na.interp()`.

Example: missing values

```r
# gold price
autoplot(gold) + ylab("") + xlab("")
```



Example: missing values

```r
# fill in missing values
gold2 <- na.interp(gold)
autoplot(gold2, series="Interpolated") + autolayer(gold, series="Original") +
  theme(legend.position = "bottom")
```

Outliers

Outliers are observations that are very different from the majority of the observations in the time series.

- they may be errors, or they may simply be unusual
- all of the methods we have considered will not work well if there are extreme outliers in the data
- we may wish to replace them with missing values, or with an estimate that is more consistent with the majority of the data

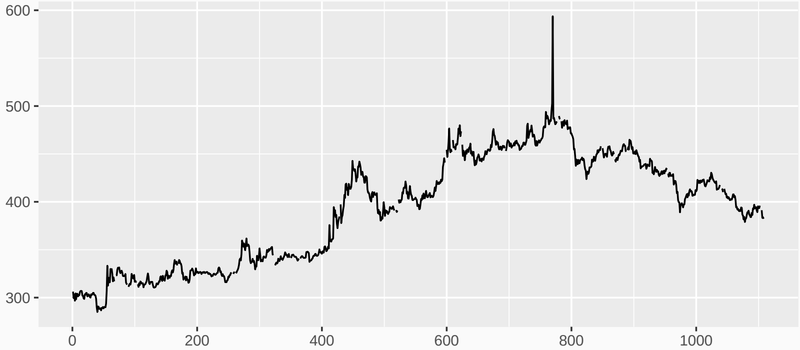
Outliers

However...

- simply replacing outliers without thinking about why they have occurred is a dangerous practice
- they may provide useful information about the process that produced the data, and which should be taken into account when forecasting
- if we are willing to assume that the outliers are genuinely errors, or that they won't occur in the forecasting period, then replacing them can make the forecasting task easier

Example: outliers

```r
# gold price
autoplot(gold) + ylab("") + xlab("")
```



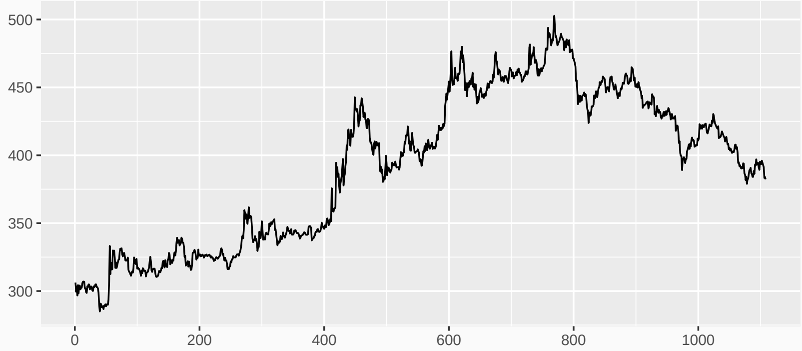
Example: outliers

```r
# find outliers
tsoutliers(gold)
```

```
## $index
## [1] 770
##
## $replacements
## [1] 494.9
```

Example: outliers

```r
# remove outliers
gold2 <- tsclean(gold)
autoplot(gold2) + ylab("") + xlab("")
```



Outline

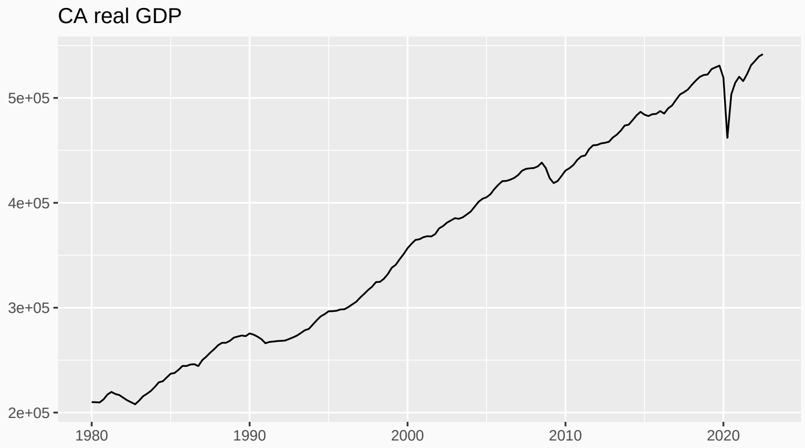
Some very current data

Let's take a look at some recent macro data (for the sample 1980Q1 to 2022Q3 and Q4).

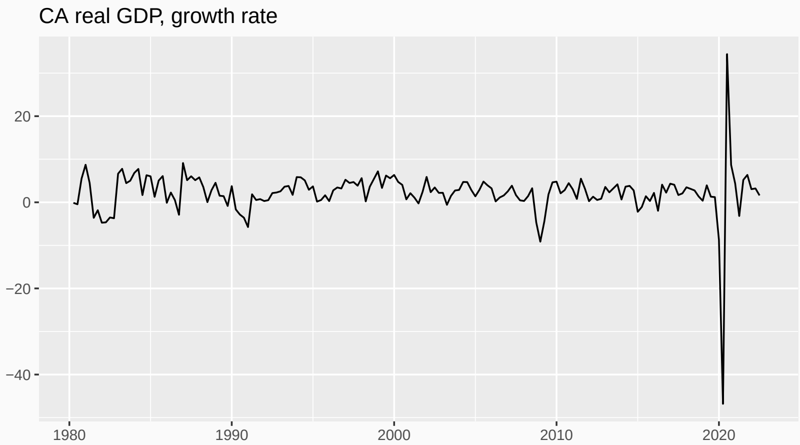
```r
# read quarterly data
data <- read.csv("data/recent_macro_data.csv", header = TRUE)
tail(data, 4)
```

```
##          dates rgdp_ca unem_ca cpi_ca rgdp_us unem_us cpi_us
## 169 2022-01-01  535240   5.767  116.1   19924   3.800  287.5
## 170 2022-04-01  539560   5.067  119.5   19895   3.600  294.7
## 171 2022-07-01  541719   5.167  119.9   20055   3.567  296.5
## 172 2022-10-01      NA   5.100  120.7   20198   3.600  299.0
```

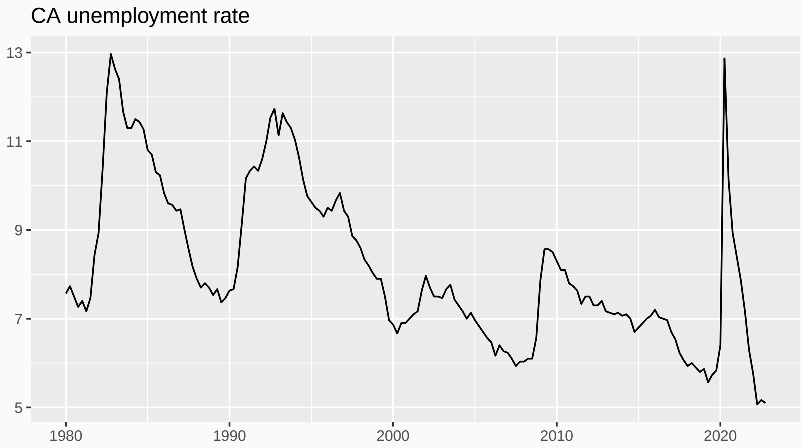
Some very current data



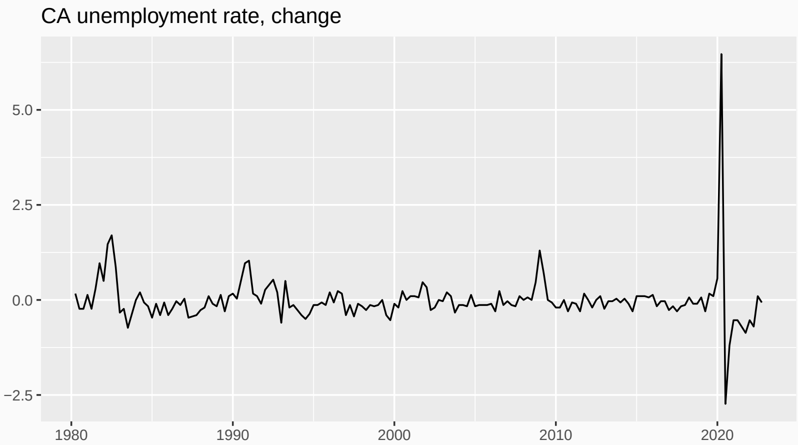
Some very current data



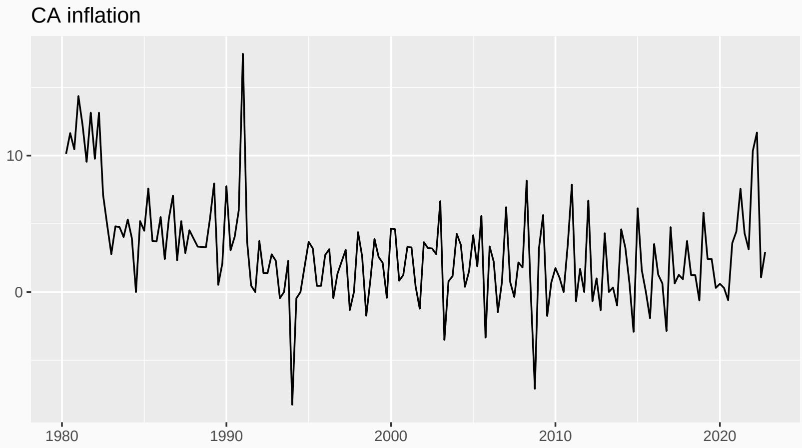
Some very current data



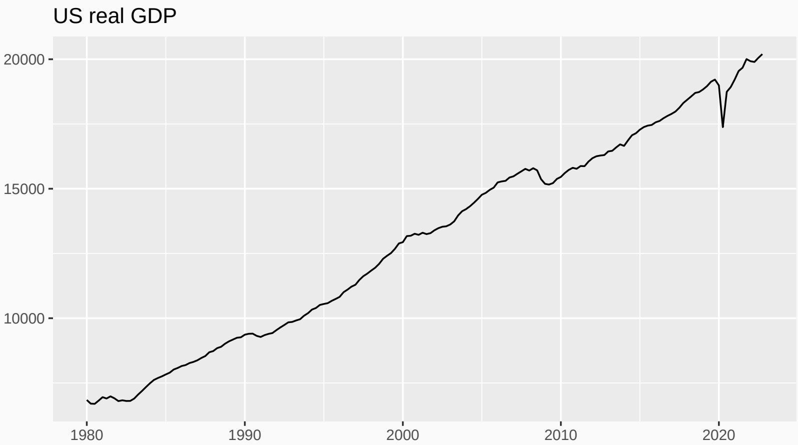
Some very current data



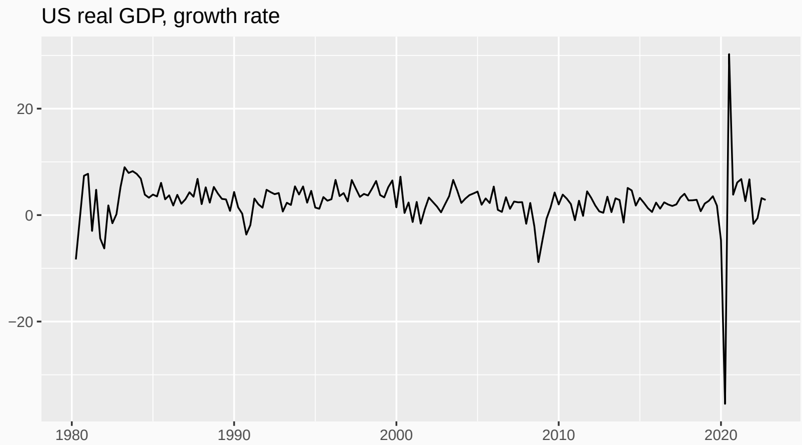
Some very current data



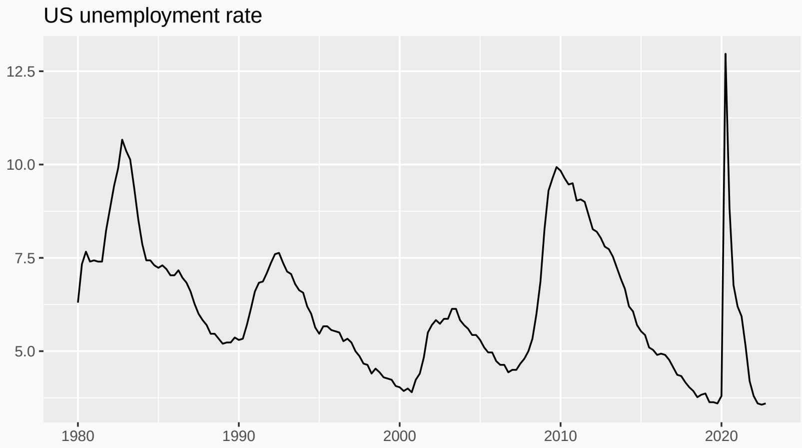
Some very current data



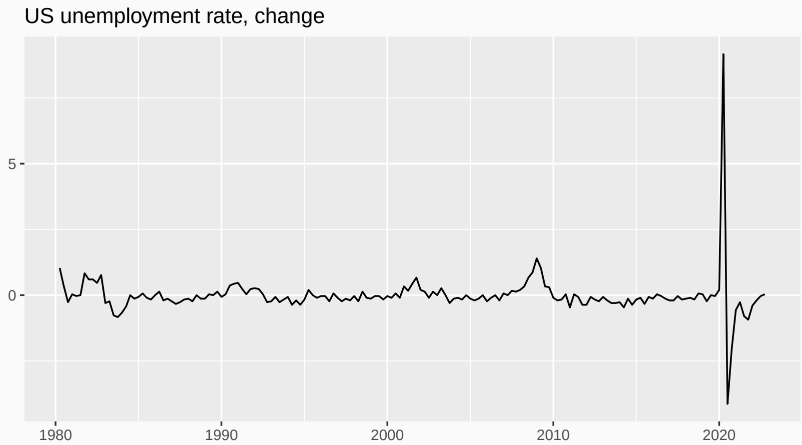
Some very current data



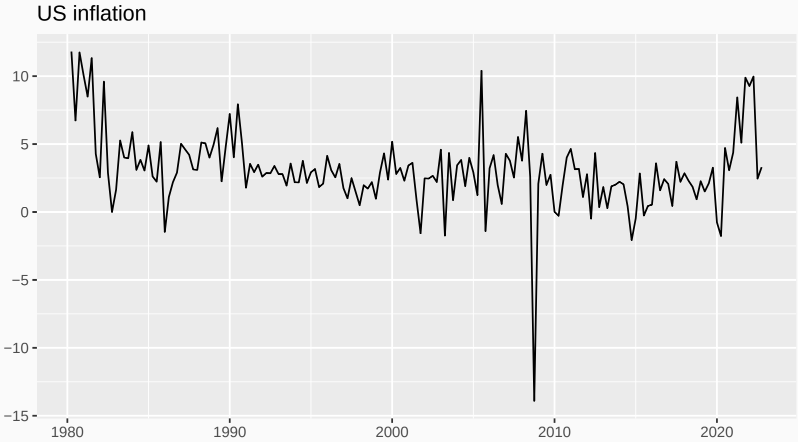
Some very current data



Some very current data



Some very current data



Outline

Example: US growth rate of GDP

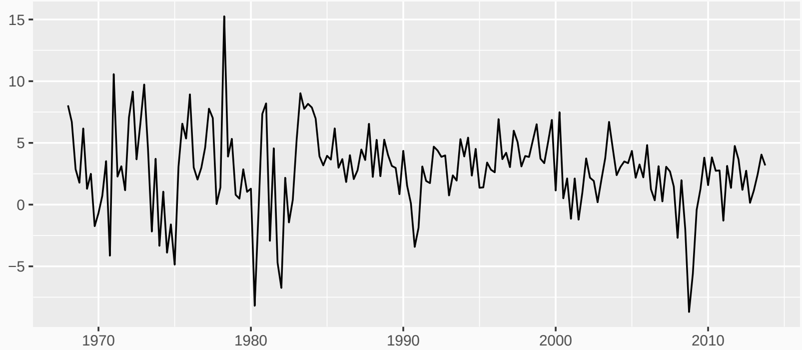
Consider forecasting the growth rate of GDP.

```r
# read quarterly data
data <- read.csv("data/us_macro_quarterly.csv", header = TRUE)

# get time series
gdpgr <-
  ts(
    400*diff(log(data$GDPC96)),
    start = c(1957,2),
    freq = 4
  )
gdpgr0 <- window(gdpgr, start = 1968)
```

Example: US growth rate of GDP

```r
# plot data
autoplot(gdpgr0) + xlab("") + ylab("")
```

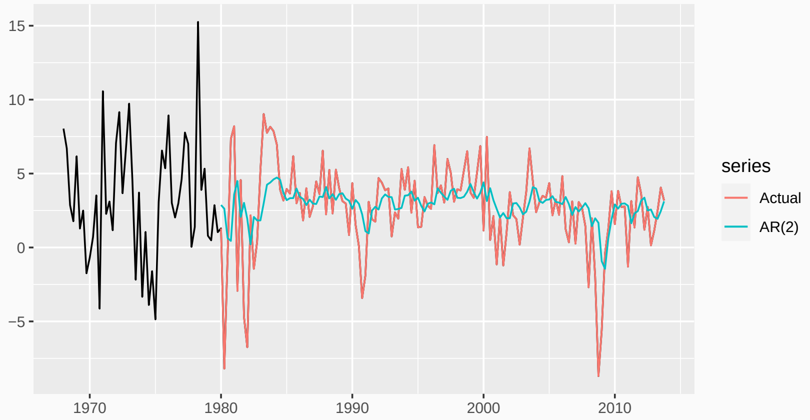


Example: US growth rate of GDP

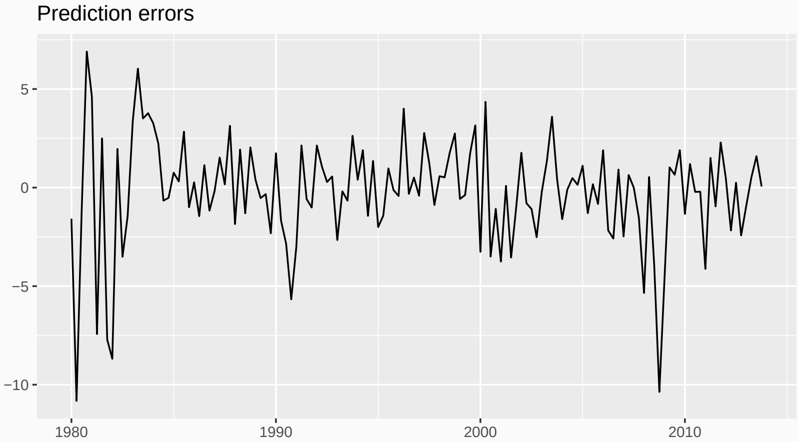
Consider an AR(2) model for the growth rate of GDP and a forecasting horizon of one quarter ($h = 1$).

```
# loop
n.end <- 1979.75 # 1979Q4
pred0 <- matrix(rep(NA,136*2),136,2)
for(i in 1:136){
  # recursive forecast
  tmp0 <- 1968
  tmp1 <- n.end+(i-1)*.25
  tmp <- window(gdpgr0, tmp0, tmp1)
  pred0[i,1] <- window(gdpgr0, tmp1+.25, tmp1+.25) # actual
  pred0[i,2] <- forecast(Arima(tmp, order = c(2,0,0)), h = 1)$mean
}
```

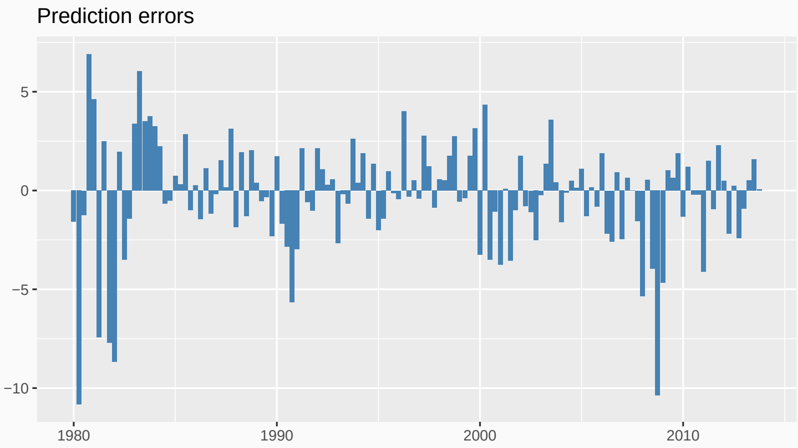
Example: US growth rate of GDP



Example: US growth rate of GDP



Example: US growth rate of GDP

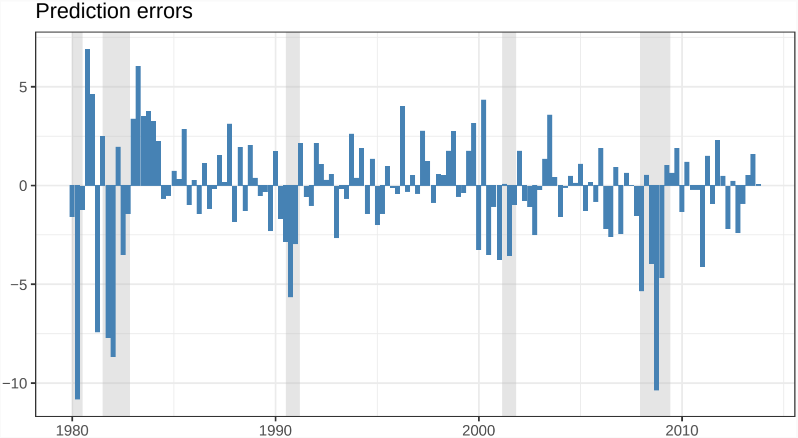


Example: US growth rate of GDP

We can get "official" NBER recession dates (monthly) from the NBER's website.

```r
# NBER peaks and troughs
peak <- c(1980+0/12, 1981+6/12, 1990+6/12, 2001+2/12, 2007+11/12)
trough <- c(1980+6/12, 1982+10/12, 1991+2/12, 2001+10/12, 2009+5/12)
recessions.df <- data.frame(Peak = peak, Trough = trough)
```

Example: US growth rate of GDP



Example: US growth rate of GDP

```r
# test unbiasedness
nber0 <- window(nber.q, start = c(1980,1))
coeftest(tslm(pred0.err ~ nber0))
```

```
##
## t test of coefficients:
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.304      0.228    1.34     0.18
## nber0         -3.603      0.580   -6.22    6e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- forecasts are unbiased in expansions (intercept not significant) but biased in recessions (indicator significant)

Example: US growth rate of GDP

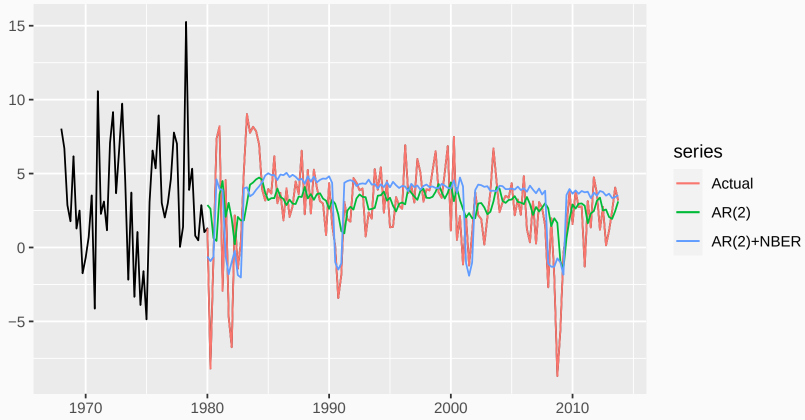
What happens if we use the NBER recession dates as a predictor?

Does it make sense?

Example: US growth rate of GDP

```r
# loop
n.end <- 1979.75 # 1979Q4
pred1 <- matrix(rep(NA,136*3),136,3)
for(i in 1:136){
  # recursive forecast
  tmp0 <- 1968
  tmp1 <- n.end+(i-1)*.25
  tmp <- window(gdpgr0, tmp0, tmp1)
  reg.est <- window(nber.q, tmp0, tmp1)
  reg.new <- window(nber.q, tmp1+.25, tmp1+.25)
  pred1[i,1] <- window(gdpgr0, tmp1+.25, tmp1+.25) # actual
  pred1[i,2] <- forecast(Arima(tmp, order = c(2,0,0)),h = 1)$mean
  pred1[i,3] <- forecast(Arima(tmp, order = c(2,0,0), xreg = reg.est),
                         h = 1, xreg = reg.new)$mean
}
```

Example: US growth rate of GDP



Example: US growth rate of GDP

```r
# get prediction errors
pred1.err <- ts(pred1[,1] - pred1[,3], start = c(1980,1), freq = 4)
pred1.err.df <- data.frame(date = time(pred1.err), error = pred1.err)

# compute rmse (h=1)
fit1 <- pred1[,1] - pred1[,2]
fit2 <- pred1[,1] - pred1[,3]
rmse1 <- sqrt(mean(fit1^2, na.rm=TRUE))
rmse2 <- sqrt(mean(fit2^2, na.rm=TRUE))

# display rmse
cbind(rmse1,rmse2)
```

```
##      rmse1 rmse2
## [1,] 2.764 2.445
```

- we get an improvement in RMSE!

Example: US growth rate of GDP

```
# test unbiasedness
coeftest(tslm(pred1.err ~ nber0))
```

```
##
## t test of coefficients:
##
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.732      0.220   -3.32   0.0012 **
## nber0          0.280      0.560    0.50   0.6185
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- forecasts are now unbiased in recessions… but biased in expansions!

Example: US growth rate of GDP

Since recession dates are not known at the time the forecasts are made, the previous analysis is only informative about a problem in our forecasts.

Two more interesting questions are:

1. Can we forecast recessions?
2. Can we use those forecasts in our model to get better forecasts of real GDP growth?

Example: US growth rate of GDP

Conclusion: forecasting is hard! However…

- we learned that an AR(2) model for US real GDP growth does not work well in recessions (biased forecasts)
- a good predictive model for US recessions is needed to improve forecasts of real GDP growth (for example, a Probit model with more predictors)
- Chauvet and Potter (2013) show that this can be done (the models are complicated and outside the scope of this course)

Example: US growth rate of GDP

This graph from Fossati (2018) shows actual US real GDP growth (black line), forecasts from an AR(2) (green line), and forecasts from an AR(2) plus a probability of recession and an real activity factor for the US economy (blue line).