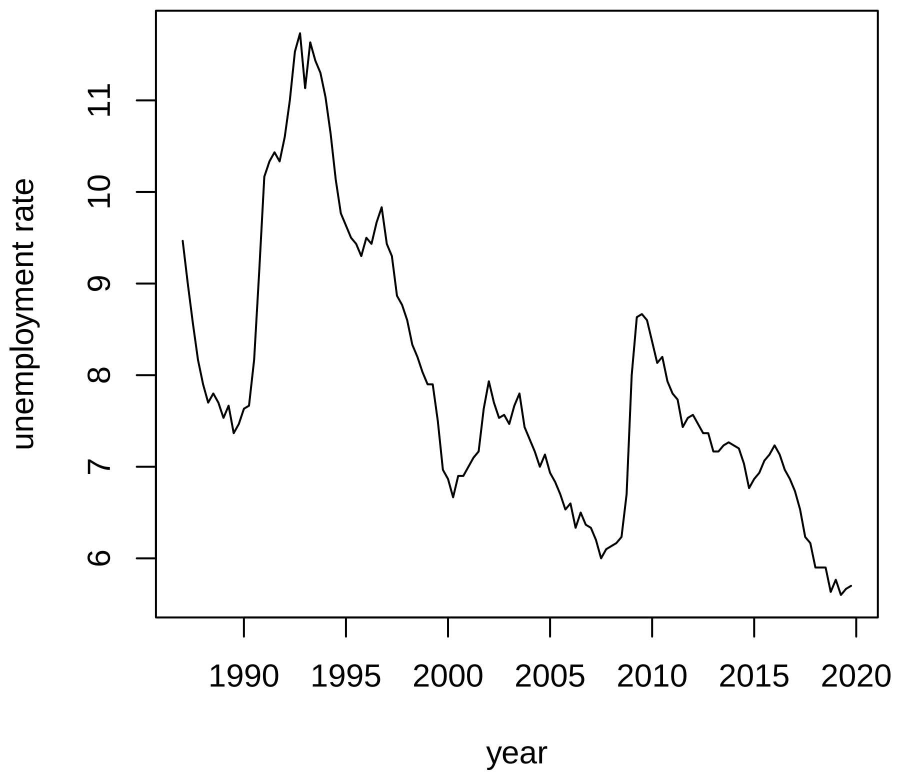
Forecasting unemployment rate, with/without GDP

2026 January 19

```
source("esm-fcrules.R")
source("simple-fcrules.R")
source("linear-fcrules.R")
```

```
# years 1961 to 2019 or 59 years 236 quarters
df = read.csv("gdp-unemploy-FRED.csv", header=T, nrow=236)
# restrict to start in 1987, omit 104 rows

names(df)
#> [1] "date"        "gdp"         "unempl"      "diflngdp"    "diflnunempl"


df = df[-(1:104),]

head(df)
#>           date      gdp   unempl  diflngdp diflnunempl
#> 105 1987-01-01 265295.0 9.466666 2.2850937   0.3527305
#> 106 1987-04-01 268693.5 9.000000 1.2728910  -5.0552209
#> 107 1987-07-01 272798.8 8.566667 1.5163202  -4.9345835
#> 108 1987-10-01 276323.0 8.166667 1.2835944  -4.7817872
#> 109 1988-01-01 280349.0 7.900000 1.4464784  -3.3198110
#> 110 1988-04-01 282836.8 7.700000 0.8834797  -2.5642431

# differences of log of consecutive values have been multiplied by 100
print(log(df$gdp[2])-log(df$gdp[1]))
#> [1] 0.01272891

print(log(df$unempl[2])-log(df$unempl[1]))
#> [1] -0.05055221

tail(df)
#>           date      gdp   unempl  diflngdp diflnunempl
#> 231 2018-07-01 551962.3 5.900000 0.5031640   0.0000000
#> 232 2018-10-01 553262.5 5.633333 0.2352826  -4.6251077
#> 233 2019-01-01 554594.0 5.766667 0.2403742   2.3392997
#> 234 2019-04-01 560665.5 5.600000 1.0888156  -2.9327673
#> 235 2019-07-01 562233.3 5.666667 0.2792417   1.1834516
#> 236 2019-10-01 563838.5 5.700000 0.2850975   0.5865061


# first column is date
gdp =  df[,2];
unempl =  df[,3]
n = length(unempl)
print(n)
```

```
#> [1] 132

ntrain = 80  # holdout set 13 years with 4 quarters per year
train = unempl[1:ntrain]
holdout = unempl[(ntrain+1):n]

plot(seq(1987,2019.75,0.25),unempl,xlab="year",ylab="unemployment rate",type="l")
```



Holt and some simple forecasting rules

```
fit_hw = HoltWinters(train, gamma=F)
print(fit_hw)
#> Holt-Winters exponential smoothing with trend and without seasonal component.
#>
#> Call:
#> HoltWinters(x = train, gamma = F)
#>
#> Smoothing parameters:
#>  alpha: 1
#>  beta : 0.4090381
#>  gamma: FALSE
```

```
#>
#> Coefficients:
#>          [,1]
#> a  6.3666670
#> b -0.0631809

holt = lholt_fc(train,holdout, alpha=fit_hw$alpha, beta=fit_hw$beta,
  level=fit_hw$coefficients[1],
  slope=fit_hw$coefficients[2], iprint=F)

cat("holt_rmse=", holt$rmse,"\n")
#> holt_rmse= 0.2443131

persist = persist_fc(train,holdout,iprint=F)
cat("persist_rmse=", persist$rmse,"\n")
#> persist_rmse= 0.2520853

iid = iid_fc(train,holdout,iprint=F)
cat("iid_rmse=", iid$rmse,"\n")
#> iid_rmse= 1.723211

ar1 = lm(train[-1] ~ train[-ntrain])
print(summary(ar1))
#>
#> Call:
#> lm(formula = train[-1] ~ train[-ntrain])
#>
#> Residuals:
#>      Min       1Q   Median       3Q      Max
#> -0.52639 -0.17106 -0.07688  0.16062  1.07941
#>
#> Coefficients:
#>                Estimate Std. Error t value Pr(>|t|)
#> (Intercept)     0.05068    0.19869   0.255    0.799
#> train[-ntrain]  0.98941    0.02306  42.905   <2e-16 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 0.3027 on 77 degrees of freedom
#> Multiple R-squared:  0.9599, Adjusted R-squared:  0.9593
#> F-statistic:  1841 on 1 and 77 DF,  p-value: < 2.2e-16
bcoef = ar1$coefficient
ar1 = linear_fc(train,holdout,bcoef,iprint=F)
cat("ar1_rmse=", ar1$rmse,"\n")
#> ar1_rmse= 0.2512875
```

Prediction with previous difference log(gdp) and difference log(unemploy)

```
# dataframe with previous
df2 = data.frame(
   gprev=df$diflngdp[2:(n-1)],
   uprev=df$diflnunempl[2:(n-1)], ut=df$diflnunempl[3:n])
# number of rows is ntrain-2
print(dim(df2))
#> [1] 130   3

# 1. Fit regression on previous diflngdp with training set
reg1 = lm(ut ~ gprev, data=df2[1:(ntrain-2),])
print(summary(reg1))
#>
#> Call:
#> lm(formula = ut ~ gprev, data = df2[1:(ntrain - 2), ])
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -5.3767 -2.0299 -0.2379  1.6508  6.9609
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)    1.813      0.466   3.892 0.000212 ***
#> gprev         -3.354      0.517  -6.488 7.94e-09 ***
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 2.738 on 76 degrees of freedom
#> Multiple R-squared:  0.3565, Adjusted R-squared:  0.348
#> F-statistic:  42.1 on 1 and 76 DF,  p-value: 7.943e-09

# predicted value for 100* ( log(unempl[t]) -  log(unempl[t-1]) )
pred1 = predict(reg1, newdata=df2)
u = df$unempl[1:n]
# predicted value for unempl[t] is exp(pred1/100) * unempl[t-1]
# first pred1 is for time index 3
upred1 = exp(pred1/100) * u[2:(n-1)]

# 1-step ahead point forecasts for holdout set
u1fc = tail(upred1,n-ntrain)
uholdout = tail(u,n-ntrain)
rmse1 = sqrt(mean((u1fc-uholdout)^2))
cat("reg1_rmse=", rmse1,"\n")
#> reg1_rmse= 0.2198002


# 2. Fit regression on previous diflngdp and diflnunempl with training set
reg2 = lm(ut ~ gprev+uprev, data=df2[1:(ntrain-2),])
summary(reg2)
#>
#> Call:
#> lm(formula = ut ~ gprev + uprev, data = df2[1:(ntrain - 2), ])
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -5.5010 -1.7215 -0.2973  1.6175  6.7565
```

```
#>
#> Coefficients:
#>             Estimate Std. Error t value Pr(>|t|)
#> (Intercept)   1.6089     0.5157   3.120  0.00257 **
#> gprev        -2.9728     0.6604  -4.502 2.43e-05 ***
#> uprev         0.1082     0.1163   0.930  0.35548
#> ---
#> Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#>
#> Residual standard error: 2.741 on 75 degrees of freedom
#> Multiple R-squared:  0.3638, Adjusted R-squared:  0.3468
#> F-statistic: 21.44 on 2 and 75 DF,  p-value: 4.31e-08

# predicted value for unempl[t] is exp(pred1/100) * unempl[t-1]
pred2 = predict(reg2, newdata=df2)
u = df$unempl[1:n]
# first pred1 is for time index 3
upred2 = exp(pred2/100) * u[2:(n-1)]

# 1-step ahead point forecasts for holdout set
u2fc = tail(upred2,n-ntrain)
uholdout = tail(u,n-ntrain)
rmse2 = sqrt(mean((u2fc-uholdout)^2))
cat("reg2_rmse=", rmse2,"\n")
#> reg2_rmse= 0.2175731
```

Summary of forecast comparisons

```
library(lubridate)
yr = year(df$date); quarter = month(df$date)
yrq = yr+quarter/100;

out = cbind(yrq[(ntrain+1):n],uholdout,holt$fc,persist$fc,iid$fc, ar1$fc, u1fc,u2fc)
colnames(out) = c("yearq","holdout","holt","persist","iid","ar1","prevgdp","prev2var")
print(round(out,2))
#>       yearq holdout holt persist  iid  ar1 prevgdp prev2var
#> 79  2007.01    6.33 6.30    6.37 8.46 6.35    6.40     6.38
#> 80  2007.04    6.20 6.28    6.33 8.46 6.32    6.31     6.31
#> 81  2007.07    6.00 6.12    6.20 8.46 6.19    6.11     6.11
#> 82  2007.10    6.10 5.87    6.00 8.46 5.99    6.03     6.01
#> 83  2008.01    6.13 6.06    6.10 8.46 6.09    6.18     6.19
#> 84  2008.04    6.17 6.13    6.13 8.46 6.12    6.23     6.22
#> 85  2008.07    6.23 6.18    6.17 8.46 6.15    6.20     6.20
#> 86  2008.10    6.70 6.27    6.23 8.46 6.22    6.18     6.19
#> 87  2009.01    8.00 6.91    6.70 8.46 6.68    7.09     7.10
#> 88  2009.04    8.63 8.66    8.00 8.46 7.97    8.79     8.87
#> 89  2009.07    8.67 9.28    8.63 8.46 8.59    9.12     9.14
#> 90  2009.10    8.60 9.06    8.67 8.46 8.63    8.69     8.69
#> 91  2010.01    8.37 8.81    8.60 8.46 8.56    8.42     8.44
#> 92  2010.04    8.13 8.39    8.37 8.46 8.33    8.18     8.18
#> 93  2010.07    8.20 8.05    8.13 8.46 8.10    8.14     8.11
#> 94  2010.10    7.93 8.18    8.20 8.46 8.16    8.16     8.17
#> 95  2011.01    7.80 7.81    7.93 8.46 7.90    7.78     7.77
#> 96  2011.04    7.73 7.67    7.80 8.46 7.77    7.74     7.74
#> 97  2011.07    7.43 7.63    7.73 8.46 7.70    7.83     7.81
#> 98  2011.10    7.53 7.25    7.43 8.46 7.41    7.23     7.22
#> 99  2012.01    7.57 7.47    7.53 8.46 7.50    7.47     7.49
#> 100 2012.04    7.47 7.54    7.57 8.46 7.54    7.69     7.68
#> 101 2012.07    7.37 7.41    7.47 8.46 7.44    7.52     7.50
#> 102 2012.10    7.37 7.29    7.37 8.46 7.34    7.47     7.44
#> 103 2013.01    7.17 7.32    7.37 8.46 7.34    7.45     7.44
#> 104 2013.04    7.17 7.06    7.17 8.46 7.14    7.08     7.07
#> 105 2013.07    7.23 7.10    7.17 8.46 7.14    7.16     7.16
#> 106 2013.10    7.27 7.22    7.23 8.46 7.21    7.17     7.18
#> 107 2014.01    7.23 7.27    7.27 8.46 7.24    7.15     7.16
#> 108 2014.04    7.20 7.22    7.23 8.46 7.21    7.32     7.31
#> 109 2014.07    7.03 7.18    7.20 8.46 7.17    7.11     7.12
#> 110 2014.10    6.77 6.95    7.03 8.46 7.01    6.94     6.93
#> 111 2015.01    6.87 6.61    6.77 8.46 6.75    6.73     6.71
#> 112 2015.04    6.93 6.82    6.87 8.46 6.84    7.12     7.11
#> 113 2015.07    7.07 6.93    6.93 8.46 6.91    7.13     7.11
#> 114 2015.10    7.13 7.12    7.07 8.46 7.04    7.11     7.12
#> 115 2016.01    7.23 7.19    7.13 8.46 7.11    7.25     7.24
#> 116 2016.04    7.13 7.31    7.23 8.46 7.21    7.22     7.23
#> 117 2016.07    6.97 7.14    7.13 8.46 7.11    7.39     7.35
#> 118 2016.10    6.87 6.90    6.97 8.46 6.94    6.85     6.85
#> 119 2017.01    6.73 6.79    6.87 8.46 6.84    6.86     6.85
#> 120 2017.04    6.53 6.63    6.73 8.46 6.71    6.57     6.58
#> 121 2017.07    6.23 6.39    6.53 8.46 6.51    6.42     6.41
#> 122 2017.10    6.17 6.03    6.23 8.46 6.22    6.34     6.30
#> 123 2018.01    5.90 6.02    6.17 8.46 6.15    6.17     6.17
#> 124 2018.04    5.90 5.70    5.90 8.46 5.89    5.78     5.77
#> 125 2018.07    5.90 5.78    5.90 8.46 5.89    5.85     5.86
#> 126 2018.10    5.63 5.83    5.90 8.46 5.89    5.91     5.91
#> 127 2019.01    5.77 5.48    5.63 8.46 5.62    5.69     5.66
#> 128 2019.04    5.60 5.73    5.77 8.46 5.76    5.83     5.83
#> 129 2019.07    5.67 5.51    5.60 8.46 5.59    5.50     5.49
#> 130 2019.10    5.70 5.64    5.67 8.46 5.66    5.72     5.72

rmse_vec = c(holt$rmse, persist$rmse, iid$rmse, ar1$rmse, rmse1, rmse2)
cat(round(rmse_vec,3), "\n")
#> 0.244 0.252 1.723 0.251 0.22 0.218

# Later, check if ARMAX leads to better forecasts.
```