Time Series Analysis of US New Single Family Home Sales and the Nasdaq-100 Index

STAT619 Spring 2021 - Erik Duus

House Sales - Introduction

FRED New One Family Houses Sold: https://fred.stlouisfed.org/series/HSN1FNSA

Monthly data series NOT seasonally adjusted

Choose a 20 year window for analysis. Use ensuing 7 months for prediction error

```
# New single family House sales - not seasonally-adjusted

house.df <- read.csv('./HSN1FNSA.csv', header=TRUE)
house.full_ts <- ts(data=house.df$HSN1FNSA, frequency=12, start=1963 )

# arbitrary 20-year window - avoids the GFC and covid
house.ts <- window(house.full_ts, start=1982, end=2002)
house.future <- window(house.full_ts, start=c(2002,2), end=c(2002,8))
```

Exploratory Data Analysis

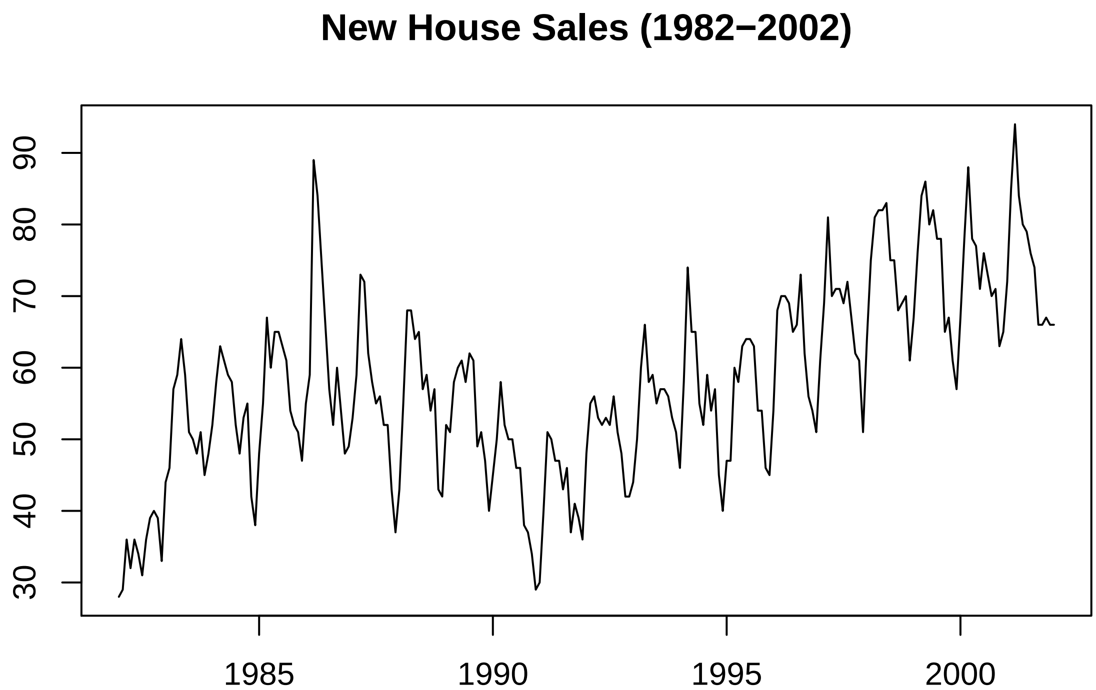
Time series shows seasonality and trend - clearly not stationary

1 month differencing to remove trend

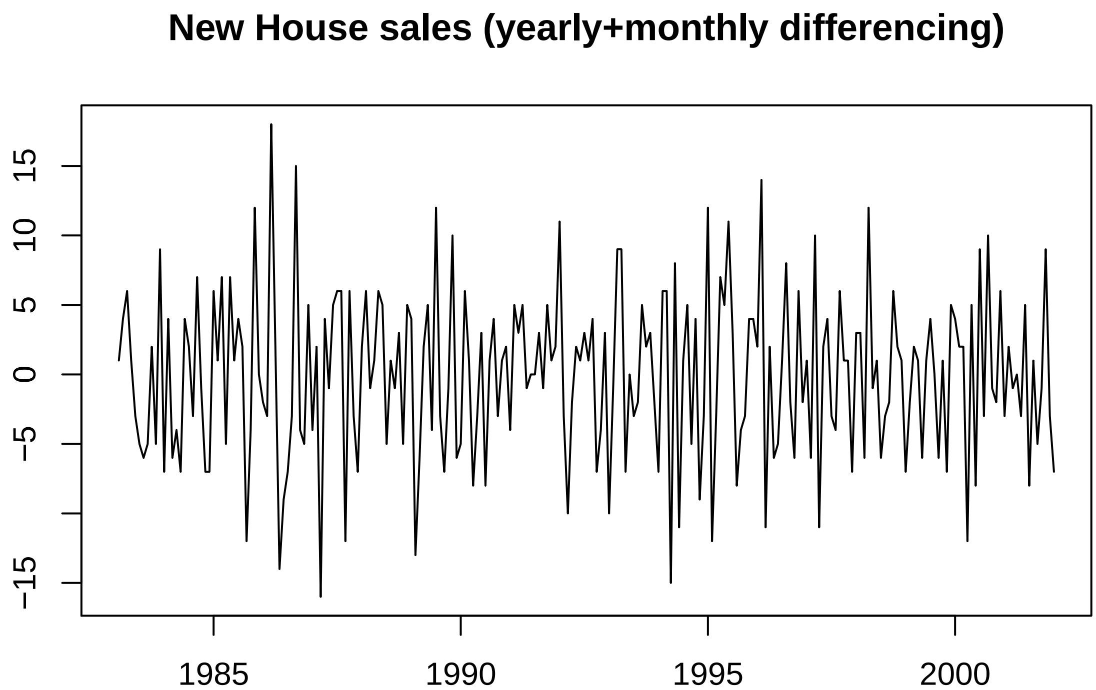
12 month differencing to remove seasonality

Resulting time series plot has constant mean and reasonably constant variance

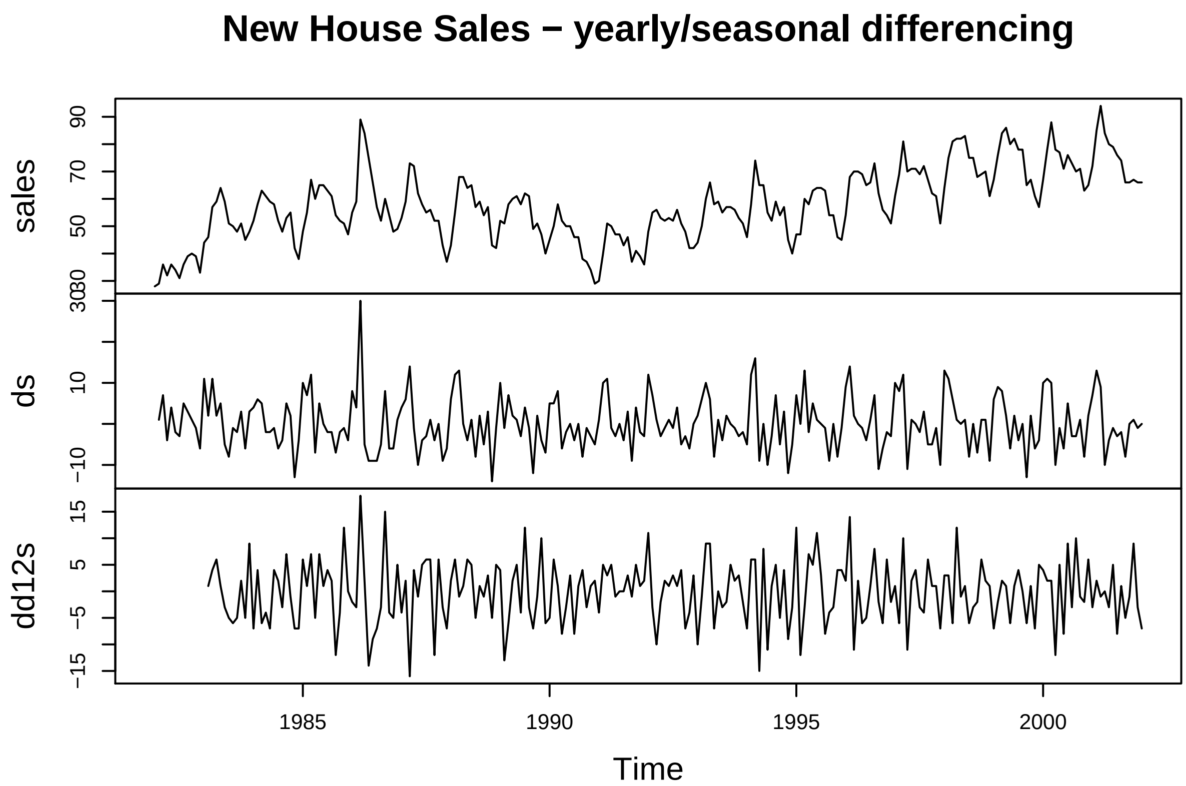
```
sales <- house.ts
ds <- diff(house.ts,1)
dd12s <- diff(ds, 12)
plot(house.ts, main="New House Sales (1982-2002)", ylab='', xlab='')
```



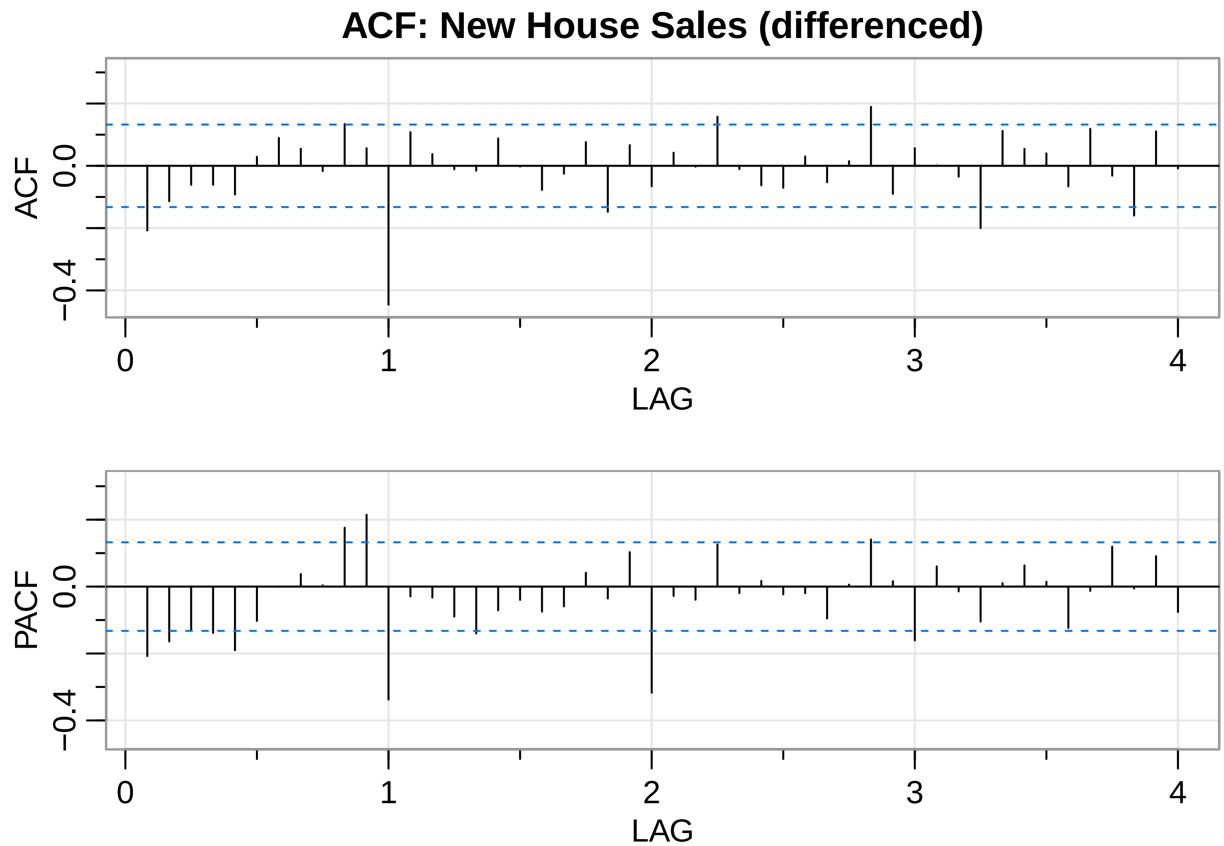

```
plot(dd12s, main='New House sales (yearly+monthly differencing)', ylab='', xlab='')
```




```r
plot(cbind(sales, ds, dd12s), main='New House Sales - yearly/seasonal differencing')
```




```
acf2(dd12s, main='ACF: New House Sales (differenced)')
```




```
##       [,1]  [,2]  [,3]  [,4]  [,5]  [,6] [,7] [,8]  [,9] [,10] [,11] [,12]
## ACF  -0.21 -0.11 -0.06 -0.06 -0.09  0.03 0.09 0.05 -0.02  0.13  0.06 -0.45
## PACF -0.21 -0.16 -0.13 -0.14 -0.19 -0.10 0.00 0.04  0.00  0.18  0.21 -0.34
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## ACF   0.11  0.04 -0.01 -0.02  0.09  0.00 -0.08 -0.03  0.08 -0.15  0.07 -0.07
## PACF -0.03 -0.03 -0.09 -0.14 -0.07 -0.04 -0.07 -0.06  0.04 -0.04  0.10 -0.32
##      [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36]
## ACF   0.04  0.00  0.16 -0.01 -0.06 -0.07  0.03 -0.05  0.02  0.19 -0.09  0.06
## PACF -0.03 -0.04  0.13 -0.02  0.02 -0.02 -0.02 -0.10  0.01  0.14  0.02 -0.16
##      [,37] [,38] [,39] [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48]
## ACF   0.00 -0.03  -0.2  0.11  0.05  0.04 -0.07  0.12 -0.03 -0.16  0.11 -0.01
## PACF  0.06 -0.01  -0.1  0.01  0.06  0.01 -0.12 -0.01  0.12 -0.01  0.09 -0.08
```

Statistical Analysis

stationarity test show no unit root - mean is stationary

model identification:

- seasonal ACF/PACF at lags: PACF tails off at 2 lags. ACF cuts off at 1 lag. likely MA(2)

- between season lags: both tail off: ARMA(1,1)

initial model: (1,1,1) / (0,1,2)(12)

- sma2 not significant

next: (1,1,1) / (0,1,1)(12)

- all coefficients significant
- ljung-box statistics are not significant

parsimonious:

- (1,1,0) / (0,1,1)(12): bad ljung box
- (0,1,1) / (0,1,1)(12): bad ljung box

```r
# null hypothesis not stationary
adf.test(dd12s, k=0)
```

```
## Warning in adf.test(dd12s, k = 0): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  dd12s
## Dickey-Fuller = -18.448, Lag order = 0, p-value = 0.01
## alternative hypothesis: stationary
```

```r
adf.test(dd12s)
```

```
## Warning in adf.test(dd12s): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  dd12s
## Dickey-Fuller = -7.8388, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```r
pp.test(dd12s)
```

```
## Warning in pp.test(dd12s): p-value smaller than printed p-value
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  dd12s
## Dickey-Fuller Z(alpha) = -230.63, Truncation lag parameter = 4, p-value
## = 0.01
## alternative hypothesis: stationary
```

```r
# null hypothesis stationary
kpss.test(dd12s)
```

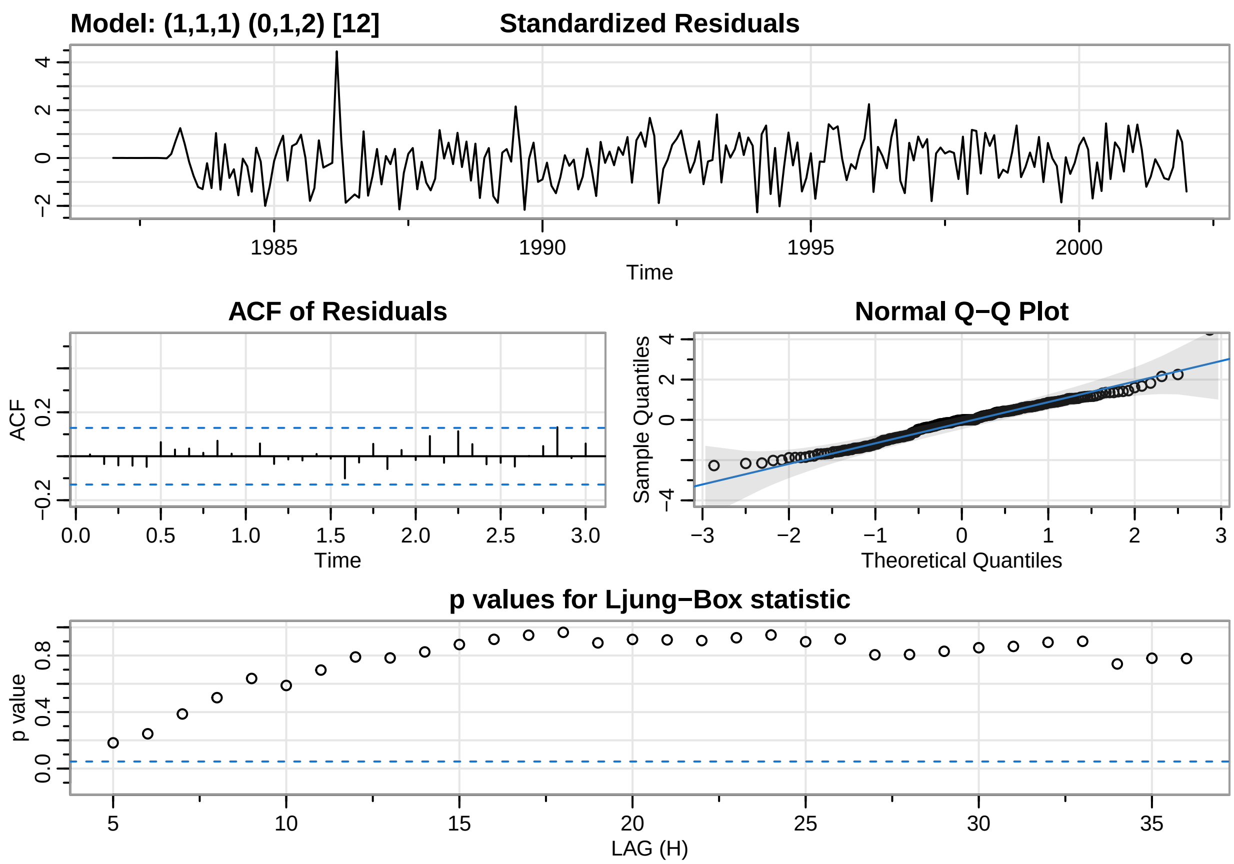
```
## Warning in kpss.test(dd12s): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  dd12s
## KPSS Level = 0.028947, Truncation lag parameter = 4, p-value = 0.1
```

```r
sarima(house.ts, 1,1,1, 0,1,2,12)
```

```
## initial  value 1.783928
## iter   2 value 1.602423
## iter   3 value 1.571099
## iter   4 value 1.557496
## iter   5 value 1.542675
## iter   6 value 1.541910
## iter   7 value 1.540723
## iter   8 value 1.539893
## iter   9 value 1.536277
## iter  10 value 1.532307
## iter  11 value 1.528518
## iter  12 value 1.526342
## iter  13 value 1.523714
## iter  14 value 1.522852
## iter  15 value 1.521819
## iter  16 value 1.521766
## iter  17 value 1.521764
```

```
## iter  18 value 1.521762
## iter  18 value 1.521762
## iter  18 value 1.521762
## final  value 1.521762
## converged
## initial  value 1.507432
## iter   2 value 1.503267
## iter   3 value 1.501312
## iter   4 value 1.497984
## iter   5 value 1.497634
## iter   6 value 1.497550
## iter   7 value 1.497261
## iter   8 value 1.496364
## iter   9 value 1.495918
## iter  10 value 1.495736
## iter  11 value 1.495629
## iter  12 value 1.495551
## iter  13 value 1.495547
## iter  14 value 1.495546
## iter  15 value 1.495546
## iter  16 value 1.495546
## iter  17 value 1.495545
## iter  17 value 1.495545
## final  value 1.495545
## converged
```



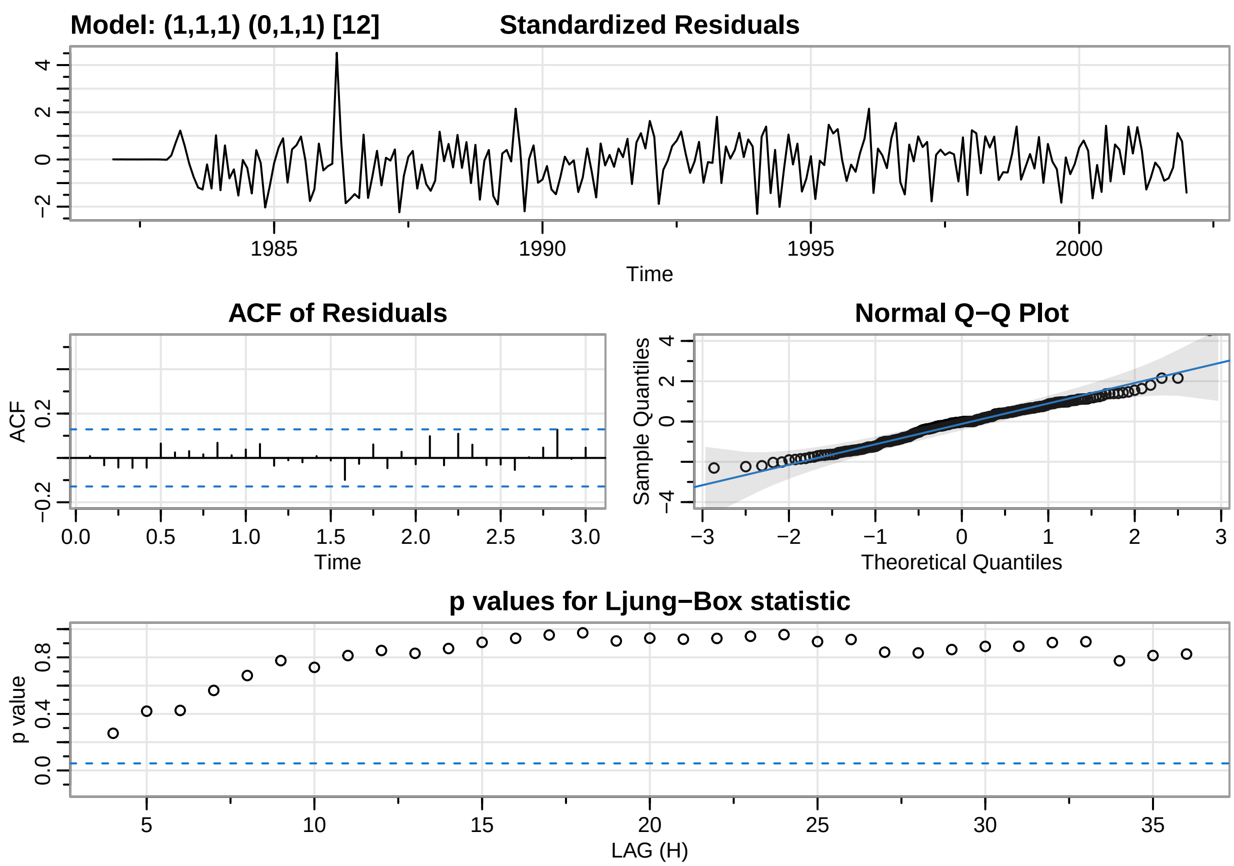

```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ma1     sma1     sma2
##       0.4993  -0.7787  -0.8368  -0.0518
## s.e.  0.1125   0.0791   0.0775   0.0746
##
## sigma^2 estimated as 18.37:  log likelihood = -664.5,  aic = 1339
##
## $degrees_of_freedom
## [1] 224
##
## $ttable
##      Estimate     SE  t.value p.value
## ar1    0.4993 0.1125   4.4367  0.0000
## ma1   -0.7787 0.0791  -9.8430  0.0000
```

```
## sma1  -0.8368 0.0775 -10.7980  0.0000
## sma2  -0.0518 0.0746  -0.6945  0.4881
##
## $AIC
## [1] 5.60253
##
## $AICc
## [1] 5.603245
##
## $BIC
## [1] 5.674274
```

```r
sarima(house.ts, 1,1,1, 0,1,1,12) # best
```

```
## initial  value 1.783928
## iter   2 value 1.609478
## iter   3 value 1.580394
## iter   4 value 1.561150
## iter   5 value 1.559396
## iter   6 value 1.546729
## iter   7 value 1.542597
## iter   8 value 1.540415
## iter   9 value 1.540111
## iter  10 value 1.539229
## iter  11 value 1.533189
## iter  12 value 1.530920
## iter  13 value 1.528923
## iter  14 value 1.523019
## iter  15 value 1.522282
## iter  16 value 1.522218
## iter  17 value 1.522140
## iter  18 value 1.522116
## iter  19 value 1.522109
## iter  20 value 1.522109
## iter  20 value 1.522109
## iter  20 value 1.522109
## final  value 1.522109
## converged
## initial  value 1.507962
## iter   2 value 1.502481
## iter   3 value 1.500149
## iter   4 value 1.497009
## iter   5 value 1.496777
## iter   6 value 1.496723
```

```
## iter   7 value 1.496625
## iter   8 value 1.496622
## iter   8 value 1.496622
## final  value 1.496622
## converged
```

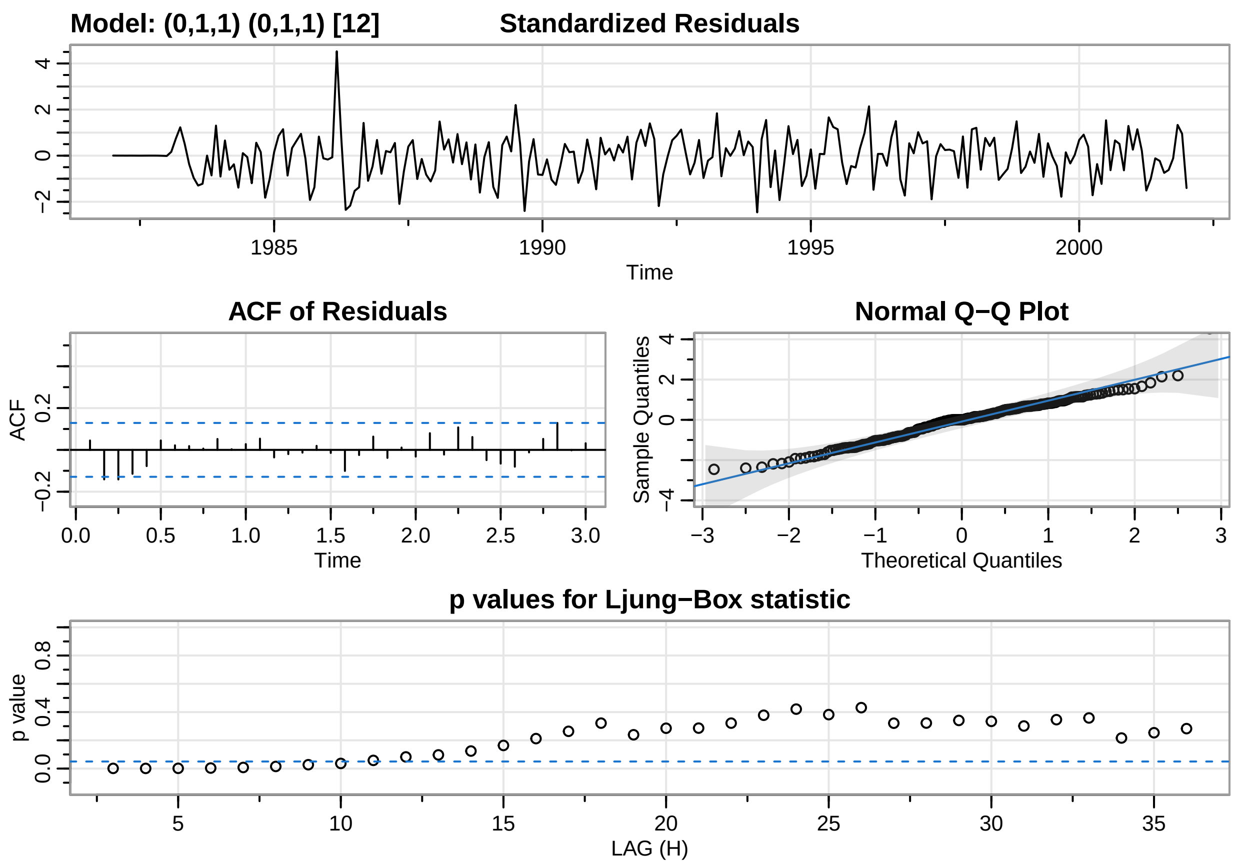



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ma1     sma1
##       0.4991  -0.7764  -0.8710
## s.e.  0.1116   0.0783   0.0584
##
## sigma^2 estimated as 18.49:  log likelihood = -664.75,  aic = 1337.5
##
## $degrees_of_freedom
```

```
## [1] 225
##
## $ttable
##      Estimate     SE  t.value p.value
## ar1    0.4991 0.1116   4.4738       0
## ma1   -0.7764 0.0783  -9.9193       0
## sma1  -0.8710 0.0584 -14.9162       0
##
## $AIC
## [1] 5.596216
##
## $AICc
## [1] 5.596644
##
## $BIC
## [1] 5.653611
```

```
sarima(house.ts, 0, 1, 1, 0,1,1,12)
```

```
## initial  value 1.781792
## iter   2 value 1.593674
## iter   3 value 1.562531
## iter   4 value 1.548808
## iter   5 value 1.543731
## iter   6 value 1.541488
## iter   7 value 1.540108
## iter   8 value 1.539769
## iter   9 value 1.539755
## iter  10 value 1.539755
## iter  10 value 1.539755
## iter  10 value 1.539755
## final  value 1.539755
## converged
## initial  value 1.529144
## iter   2 value 1.521685
## iter   3 value 1.521407
## iter   4 value 1.521392
## iter   5 value 1.521391
## iter   5 value 1.521391
## iter   5 value 1.521391
## final  value 1.521391
## converged
```

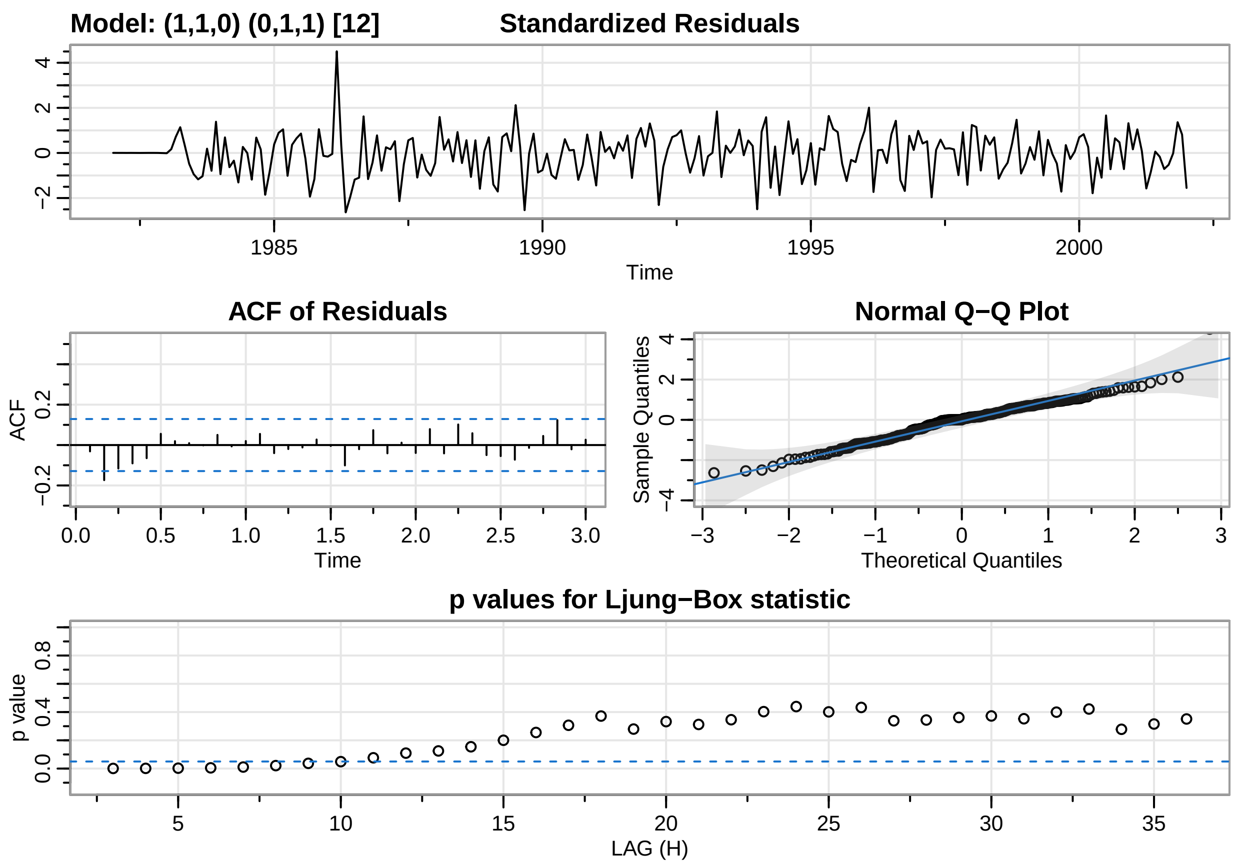



```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##           ma1     sma1
##       -0.2785  -0.8617
## s.e.   0.0862   0.0551
##
## sigma^2 estimated as 19.51:  log likelihood = -670.4,  aic = 1346.79
##
## $degrees_of_freedom
## [1] 226
##
## $ttable
##      Estimate     SE  t.value p.value
## ma1   -0.2785 0.0862  -3.2318  0.0014
## sma1  -0.8617 0.0551 -15.6295  0.0000
```

```
##
## $AIC
## [1] 5.635105
##
## $AICc
## [1] 5.635318
##
## $BIC
## [1] 5.678152
```

```r
sarima(house.ts, 1, 1, 0, 0,1,1,12)
```

```
## initial  value 1.783928
## iter   2 value 1.601635
## iter   3 value 1.570782
## iter   4 value 1.554942
## iter   5 value 1.549406
## iter   6 value 1.547181
## iter   7 value 1.545637
## iter   8 value 1.545330
## iter   9 value 1.545318
## iter  10 value 1.545317
## iter  10 value 1.545317
## iter  10 value 1.545317
## final  value 1.545317
## converged
## initial  value 1.535134
## iter   2 value 1.529860
## iter   3 value 1.529371
## iter   4 value 1.529362
## iter   4 value 1.529362
## iter   4 value 1.529362
## final  value 1.529362
## converged
```




```
## $fit
##
## Call:
## arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D, Q), period = S),
##     include.mean = !no.constant, transform.pars = trans, fixed = fixed, optim.control
##         REPORT = 1, reltol = tol))
##
## Coefficients:
##           ar1     sma1
##       -0.1763  -0.8606
## s.e.   0.0658   0.0538
##
## sigma^2 estimated as 19.84:  log likelihood = -672.21,  aic = 1350.43
##
## $degrees_of_freedom
## [1] 226
##
## $ttable
##      Estimate     SE  t.value p.value
## ar1   -0.1763 0.0658  -2.6809  0.0079
## sma1  -0.8606 0.0538 -16.0062  0.0000
```

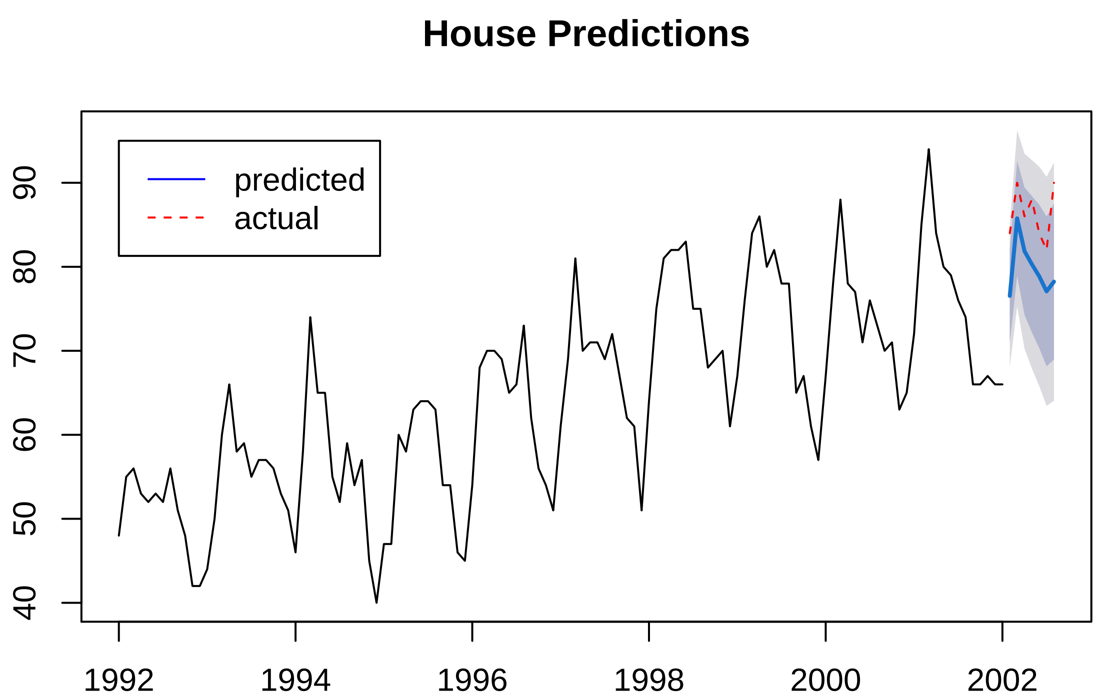
```
##
## $AIC
## [1] 5.650315
##
## $AICc
## [1] 5.650527
##
## $BIC
## [1] 5.693361
```

```
# refit best model using arima() so we can use forecast()
mod1.fit = arima(house.ts, order=c(1,1,1), seasonal=list(order=c(0,1,1),period=12))
```

predict 7 months from end of series:

- use true data to compute RMSE

```
pred1 = forecast(window(house.ts, start=1992), model=mod1.fit, h=7)
plot(pred1, main="House Predictions")
lines(house.future, lty='dashed', col='red')
legend(1992,95, legend=c('predicted','actual'), col=c('blue', 'red'), lty=1:2)
```



```r
library(Metrics)
```

```
## Warning: package 'Metrics' was built under R version 4.0.5
```

```
##
## Attaching package: 'Metrics'
```

```
## The following object is masked from 'package:forecast':
##
##     accuracy
```

```r
rmse(house.future, pred1$mean)
```

```
## [1] 6.960795
```

Model2: Regression + ARMA errors

FRED Leading Index for the United States https://fred.stlouisfed.org/series/USSLIND

Build regression model house ~ lead + ARMA errors

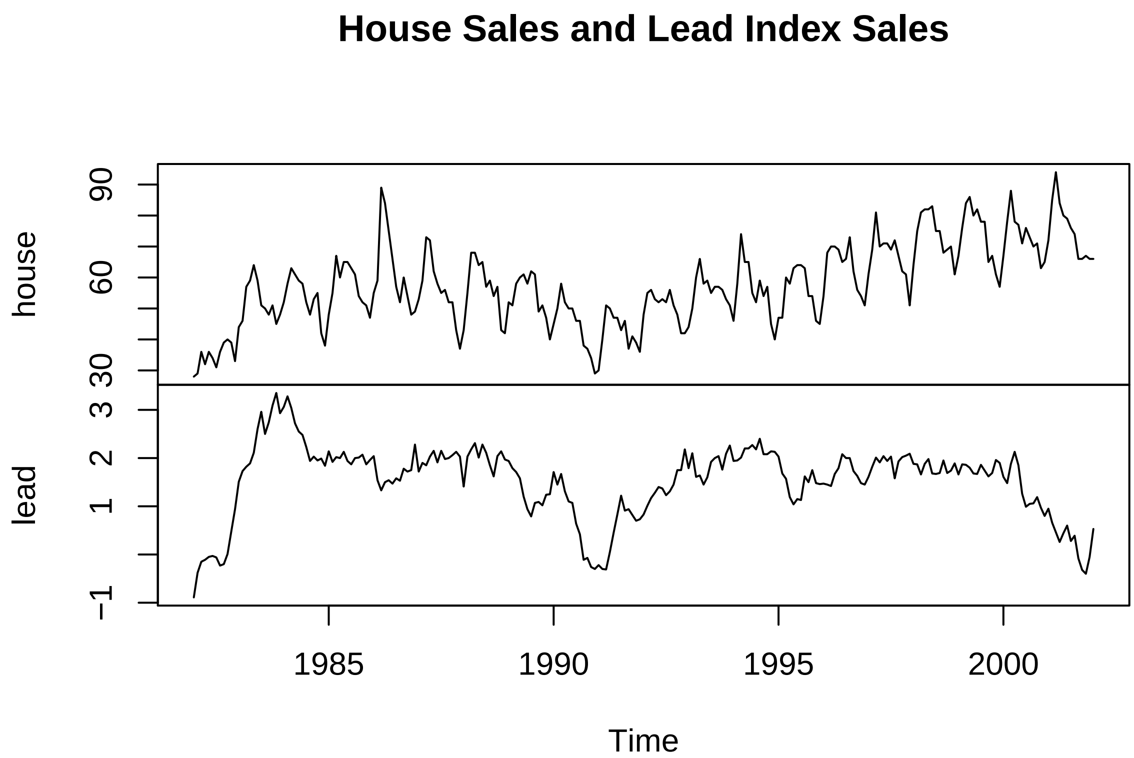
```r
lead.df <- read.csv('./USSLIND.csv', header=TRUE)
lead.ts <- ts(data=lead.df$USSLIND, frequency=12, start=1982)
lead.future = window(lead.ts, start=c(2002,2), end=c(2002,8))
```

Exploratory Data Analysis

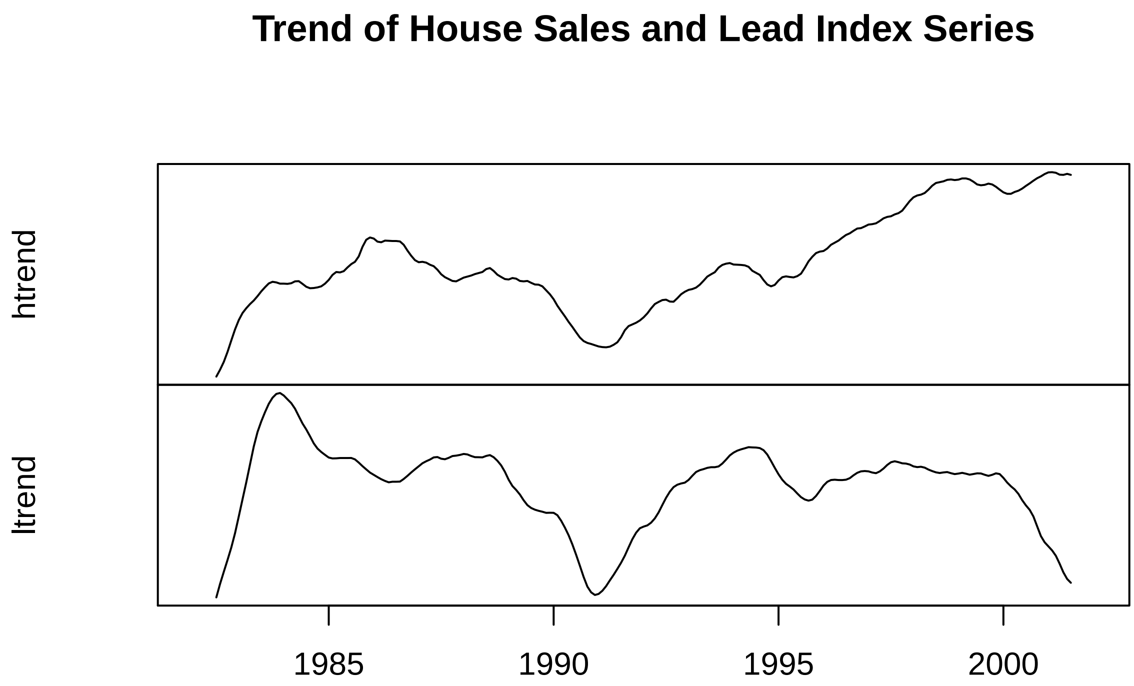
LEAD trend appears to have some correlation with HOUSE

```r
hsle <- ts.intersect(house = house.ts, lead = lead.ts)

plot(hsle, main='House Sales and Lead Index Sales')
```




```
h <- decompose(hsle[,'house'])
l <- decompose(hsle[,'lead'])
htrend <- h$trend
ltrend <- l$trend
plot(cbind(htrend,ltrend), main='Trend of House Sales and Lead Index Series',
     xlab='',ylab=cbind('',''), yaxt='n')
```



Statistical Analysis

fit regression house ~ lead

regression of house sales vs leading economic indicators

- coefficients are significant

examine residuals:

- trend and seasonality
- yearly and monthly difference the series
- constant mean, constant variance

check for stationarity: all tests pass. mean is stationary

variance is pretty constant

ACF analysis:

- seasonal lags - both seem to cut off at 1. would suggest ARMA(0,0) try (1,1)

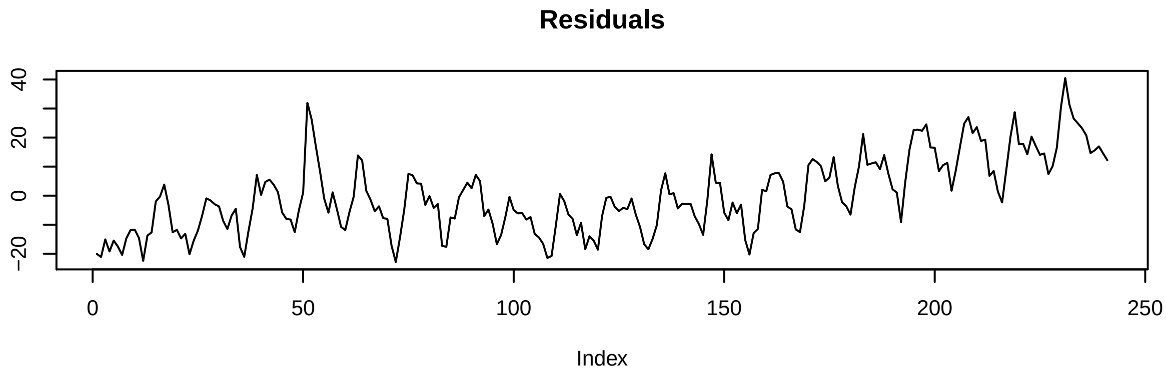
- between-season lags: tail off ARMA(1,1)

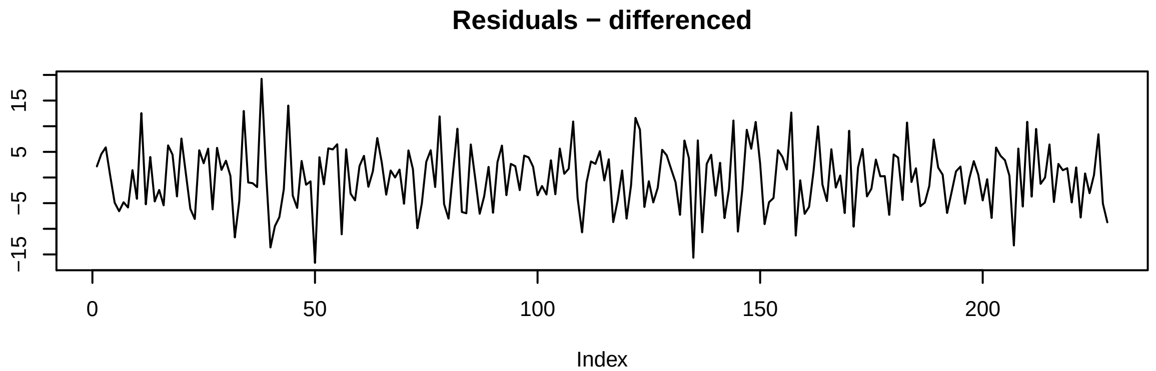
```
summary(mod2.lm <-lm(house ~ lead, data=hsle))
```

```
##
## Call:
## lm(formula = house ~ lead, data = hsle)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.872  -9.350  -2.024   8.451  40.467
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   51.676      1.856   27.84  < 2e-16 ***
## lead           4.037      1.088    3.71 0.000257 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.86 on 239 degrees of freedom
## Multiple R-squared:  0.05447,    Adjusted R-squared:  0.05051
## F-statistic: 13.77 on 1 and 239 DF,  p-value: 0.0002574
```

```
res = resid(mod2.lm)
dd12res = diff(diff(res,12),1)

layout(matrix(c(1,1,1,1,
                2,2,2,2),
              nrow=2, byrow=TRUE))
plot(res, type="l", main='Residuals', ylab='')
plot(dd12res, type="l", main='Residuals - differenced', ylab='')
```






```r
# null hypothesis not stationary
adf.test(dd12res, k=0)
```

```
## Warning in adf.test(dd12res, k = 0): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  dd12res
## Dickey-Fuller = -17.118, Lag order = 0, p-value = 0.01
## alternative hypothesis: stationary
```

```r
adf.test(dd12res)
```

```
## Warning in adf.test(dd12res): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  dd12res
## Dickey-Fuller = -8.483, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```r
pp.test(dd12res)
```

```
## Warning in pp.test(dd12res): p-value smaller than printed p-value
```

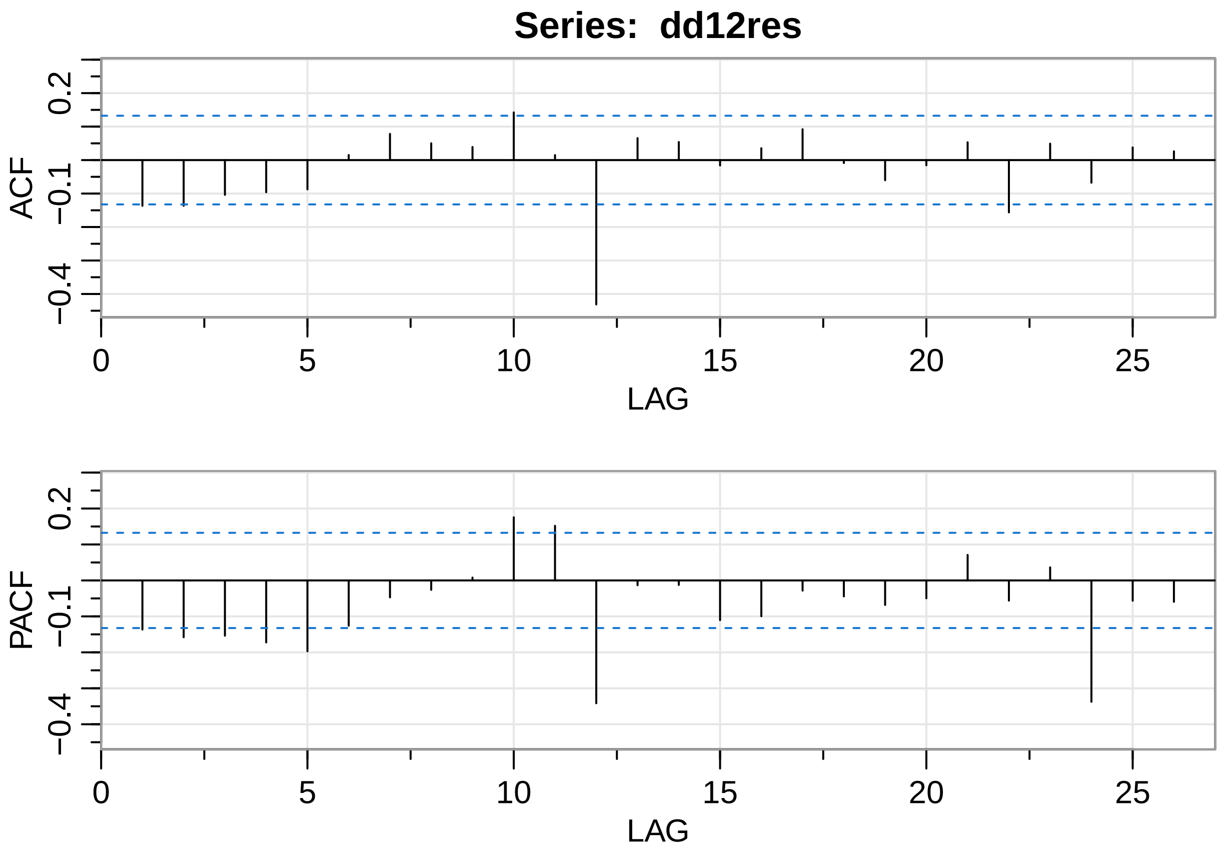
```
##
##  Phillips-Perron Unit Root Test
##
## data:  dd12res
## Dickey-Fuller Z(alpha) = -211.89, Truncation lag parameter = 4, p-value
## = 0.01
## alternative hypothesis: stationary
```

```r
# null hypothesis stationar
kpss.test(dd12res)
```

```
## Warning in kpss.test(dd12res): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  dd12res
## KPSS Level = 0.016421, Truncation lag parameter = 4, p-value = 0.1
```

```r
acf2(dd12res)
```

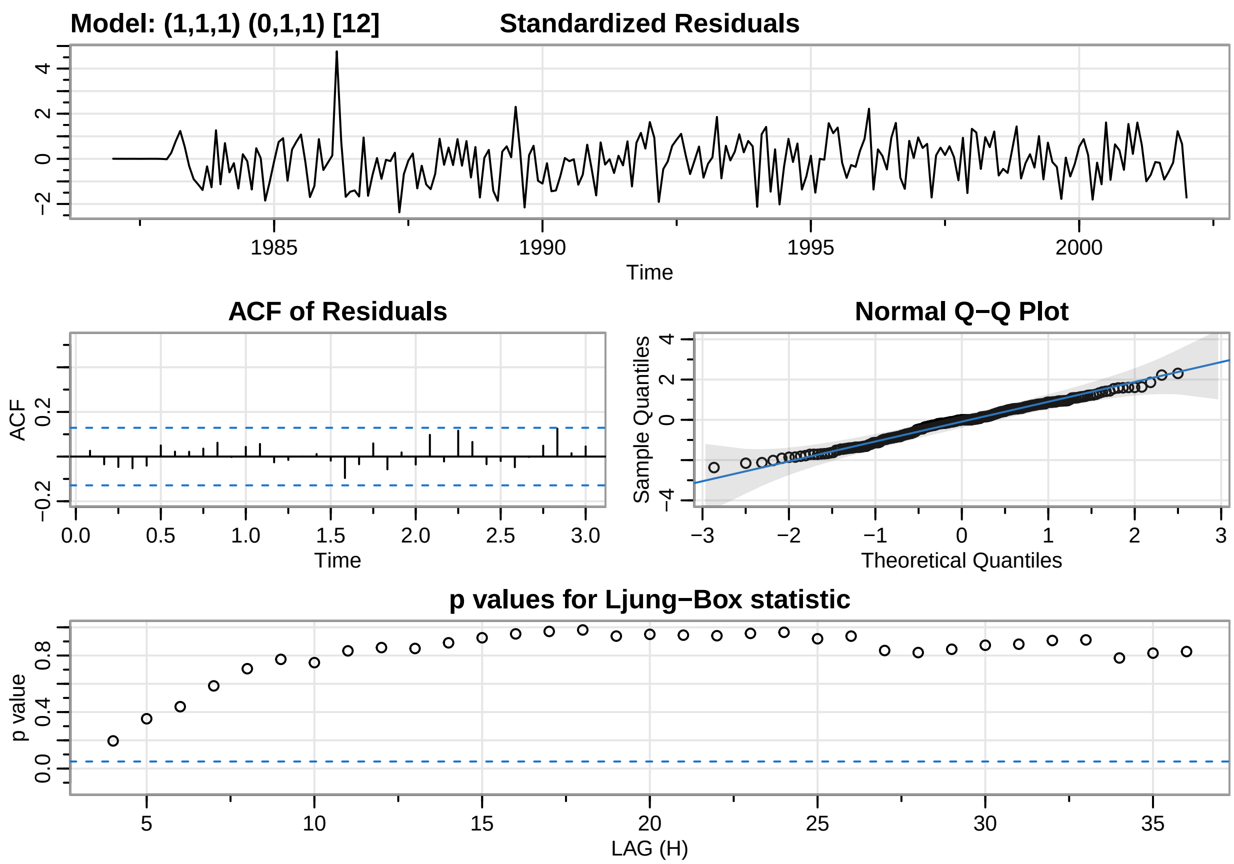



```
##       [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8] [,9] [,10] [,11] [,12]
## ACF  -0.14 -0.14 -0.10 -0.10 -0.09  0.02  0.08  0.05 0.04  0.14  0.01 -0.43
## PACF -0.14 -0.16 -0.15 -0.17 -0.20 -0.13 -0.05 -0.03 0.01  0.18  0.15 -0.34
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## ACF   0.07  0.05 -0.02  0.04  0.09 -0.01 -0.06 -0.02  0.05 -0.16  0.05 -0.07
## PACF -0.01 -0.01 -0.11 -0.10 -0.03 -0.04 -0.07 -0.05  0.07 -0.06  0.04 -0.34
##      [,25] [,26]
## ACF   0.04  0.03
## PACF -0.06 -0.06
```

```
sarima(hsle[,'house'], 1,1,1,0,1,1,12,xreg=hsle[,'lead'])
```

```
## initial  value 1.780655
## iter   2 value 1.610721
## iter   3 value 1.577212
## iter   4 value 1.558626
## iter   5 value 1.550345
## iter   6 value 1.543254
## iter   7 value 1.541469
## iter   8 value 1.540470
```

```
## iter   9 value 1.539425
## iter  10 value 1.536463
## iter  11 value 1.534845
## iter  12 value 1.533325
## iter  13 value 1.529449
## iter  14 value 1.524382
## iter  15 value 1.521811
## iter  16 value 1.520855
## iter  17 value 1.520331
## iter  18 value 1.520014
## iter  19 value 1.518914
## iter  20 value 1.517372
## iter  21 value 1.517241
## iter  22 value 1.515813
## iter  23 value 1.515791
## iter  24 value 1.515790
## iter  25 value 1.515789
## iter  25 value 1.515789
## iter  25 value 1.515789
## final  value 1.515789
## converged
## initial  value 1.501219
## iter   2 value 1.496087
## iter   3 value 1.492125
## iter   4 value 1.489485
## iter   5 value 1.489233
## iter   6 value 1.489059
## iter   7 value 1.489014
## iter   8 value 1.488954
## iter   9 value 1.488954
## iter  10 value 1.488954
## iter  10 value 1.488954
## final  value 1.488954
## converged
```




```
## $fit
##
## Call:
## stats::arima(x = xdata, order = c(p, d, q), seasonal = list(order = c(P, D,
##     Q), period = S), xreg = xreg, transform.pars = trans, fixed = fixed, optim.contro
##     REPORT = 1, reltol = tol))
##
## Coefficients:
##          ar1      ma1     sma1    xreg
##       0.5522  -0.8426  -0.8750  2.1029
## s.e.  0.0959   0.0651   0.0578  1.0848
##
## sigma^2 estimated as 18.16:  log likelihood = -663,  aic = 1336
##
## $degrees_of_freedom
## [1] 224
##
## $ttable
##      Estimate     SE  t.value p.value
## ar1    0.5522 0.0959   5.7569  0.0000
## ma1   -0.8426 0.0651 -12.9509  0.0000
```

```
## sma1  -0.8750 0.0578 -15.1281  0.0000
## xreg   2.1029 1.0848   1.9386  0.0538
##
## $AIC
## [1] 5.589953
##
## $AICc
## [1] 5.590669
##
## $BIC
## [1] 5.661697
```

```r
mod2.fit = arima(hsle[,'house'],xreg=hsle[,'lead'],order=c(1,1,1), seasonal=list(order=c
summary(mod2.fit)
```

```
##
## Call:
## arima(x = hsle[, "house"], order = c(1, 1, 1), seasonal = list(order = c(0,
##     1, 1), period = 12), xreg = hsle[, "lead"])
##
## Coefficients:
##          ar1      ma1      sma1  hsle[, "lead"]
##       0.5522  -0.8426  -0.8750          2.1029
## s.e.  0.0959   0.0651   0.0578          1.0848
##
## sigma^2 estimated as 18.16:  log likelihood = -663,  aic = 1336
##
## Training set error measures:
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -0.3085381 4.144945 3.191554 -0.9721627 5.629338 0.6425947
##                    ACF1
## Training set 0.02620114
```

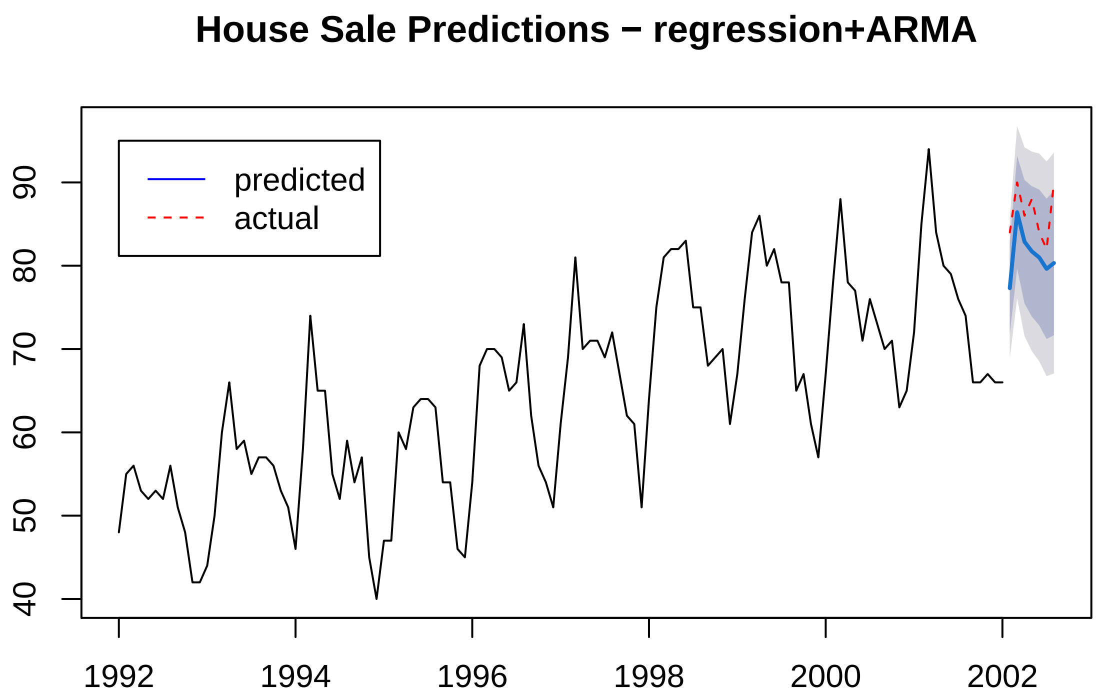
```r
coeftest(mod2.fit)
```

```
##
## z test of coefficients:
##
##                  Estimate Std. Error  z value  Pr(>|z|)
## ar1              0.552154   0.095911   5.7569 8.566e-09 ***
## ma1             -0.842614   0.065062 -12.9509 < 2.2e-16 ***
## sma1            -0.875011   0.057840 -15.1281 < 2.2e-16 ***
## hsle[, "lead"]  2.102899   1.084771   1.9386   0.05255 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
mod2.fit2 <- Arima(window(hsle[,1], start=1992),
                   model=mod2.fit,
                   xreg=window(hsle[,2], start=1992))


pred2 = forecast(mod2.fit2, xreg=lead.future)

plot(pred2, main="House Sale Predictions - regression+ARMA")
lines(house.future, lty='dashed', col='red')
legend(1992,95, legend=c('predicted','actual'), col=c('blue', 'red'), lty=1:2)
```




```r
rmse(house.future, pred2$mean)
```

```
## [1] 5.543404
```