Eco 5316 Time Series Econometrics

Lecture 9 Seasonal Models

Seasonal Models

- retail sales, labor force, unemployment rate, revenues of companies, construction spending, housing starts and total miles traveled by air are just some examples of a large number of series that exhibit regular monthly or quarterly periodic pattern
- a lot of time series are officially published after performing seasonal adjustment that theoretically should remove seasonal component
- it is however not uncommon to encounter officially published seasonally adjusted data in which the seasonal pattern is still present

Pure Seasonal Models

- simple pure seasonal AR model

$$y_t = \phi_s y_{t-s} + \varepsilon_t$$

ACF: spike at each multiple of s (lags 4, 8, 12,... in case of quarterly data, lags 12, 24, 36, ... in case of monthly data)
PACF: single spike at lag s

- simple pure seasonal MA model

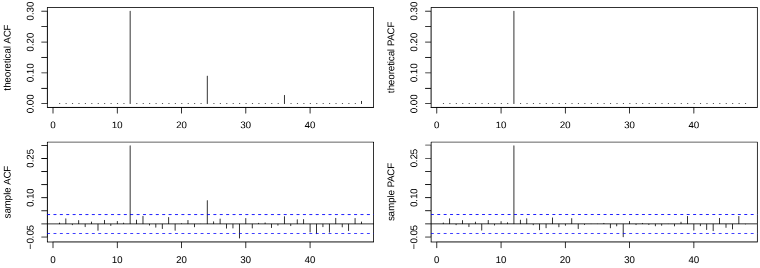
$$y_t = \varepsilon_t + \theta_s \varepsilon_{t-s}$$

PACF: spike at each multiple of s (lags 4, 8, 12,... in case of quarterly data, lags 12, 24, 36, ... in case of monthly data)
ACF: single spike at lag s

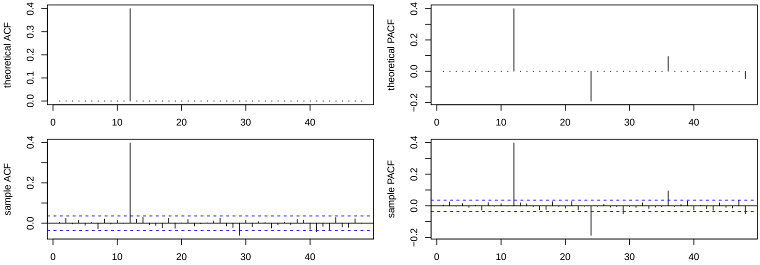
- in practice however most time series contain a seasonal AR or MA component as well as a regular AR or MA component

Pure Seasonal Models

- example: $s = 12$ and $\phi_s = 0.3$



- example: $s = 12$ and $\theta_s = 0.5$

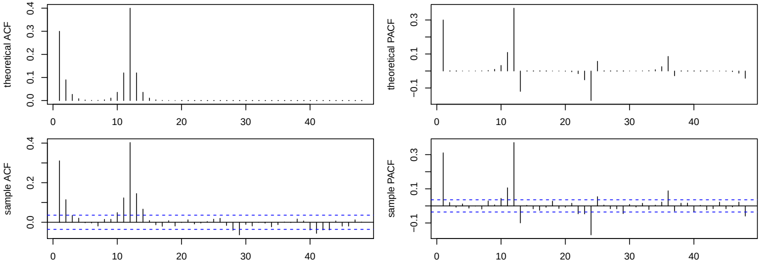


Additive Seasonal AR model

- AR model with an additive seasonal MA component

$$(1-\phi)y_t = (1+\Theta L^s)\varepsilon_t$$

  so that $y_t = \phi y_{t-1} + \varepsilon_t + \Theta\varepsilon_{t-s}$
- if $\phi > 0$, $\Theta > 0$
  - ACF will exhibit exponential decay interrupted by a rise in correlation coefficents around lag s but no similar increase around $2s$, $3s$, ...
  - PACF will be non-zero at lag 1 and unlike nonseasonal AR(1) model also show a pattern of increasing non-zero elements before *each* multiple of s
- example: $s = 12$ and $\phi = 0.3$, $\Theta = 0.5$

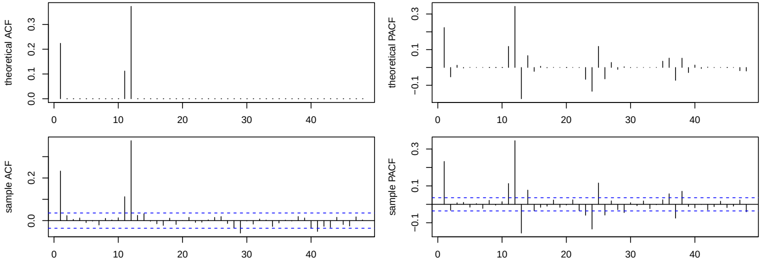


Additive Seasonal MA model

- MA model with an additive seasonal MA component

$$y_t = (1 + \theta L + \Theta L^s)\varepsilon_t$$

  so that $y_t = \varepsilon_t + \theta\varepsilon_{t-1} + \Theta\varepsilon_{t-s}$
- ACF: $\rho_1 \neq 0$, $\rho_{s-1} \neq 0$, $\rho_s \neq 0$, all other elements zero
- PACF: larger non-zero elements around each multiple of s
- example: $s = 12$ and $\theta = 0.3$, $\Theta = 0.5$

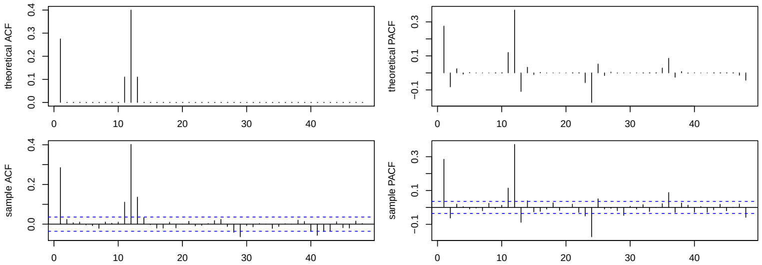


Multiplicative Seasonal MA model

- Multiplicative seasonal MA model

$$y_t = (1+\theta L)(1+\Theta L^s)\varepsilon_t$$

  so that $y_t = \varepsilon_t + \theta\varepsilon_{t-1} + \Theta\varepsilon_{t-s} + \theta\Theta\varepsilon_{t-s-1}$
- ACF: $\rho_1 \neq 0$, $\rho_{s-1} \neq 0$, $\rho_s \neq 0$, $\rho_{s+1} \neq 0$, all other elements zero
- PACF: larger non-zero elements around each multiple of s
- compared to the additive model, multiplicative model allows for *interaction of regular and seasonal components*
- example: $s = 12$ and $\theta = 0.3$, $\Theta = 0.5$

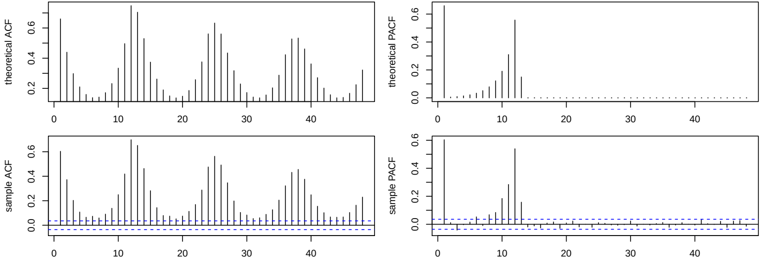


Multiplicative Seasonal AR model

- multiplicative AR model with seasonal component

$$(1-\phi L)(1-\Phi L^s)y_t = \varepsilon_t$$

  so that $y_t = \phi y_{t-1} + \Phi y_{t-s} + \phi\Phi y_{t-s-1} + \varepsilon_t$
- if $\phi > 0$, $\Phi > 0$
  - ACF: exponential decay interrupted by increasing autocorrelations around *each* multiple of s
  - PACF: large spikes at lag 1 and lag s and multiple smaller spikes between lag 2 and lag $s+1$
- example: $\phi_1 = 0.3$, $\Phi_1 = 0.5$.



Multiplicative Seasonal AR model

- Q: why are lags 2 to 11 significant in PACF for $(1-\phi L)(1-\Phi L^{12})y_t = \varepsilon_t$, that is if y_t is generated from
$$y_t = \phi y_{t-1} + \Phi y_{t-12} + \phi\Phi y_{t-13} + \varepsilon_t \tag{1}$$
- A: note that the above model implies that
$$y_{t-10} = \phi y_{t-11} + \Phi y_{t-22} + \phi\Phi y_{t-23} + \varepsilon_{t-10} \tag{2}$$
$$y_{t-11} = \phi y_{t-12} + \Phi y_{t-23} + \phi\Phi y_{t-24} + \varepsilon_{t-11} \tag{3}$$
and consider the OLS regressions to obtain PACFs for lags 10 to 14
$$\begin{aligned}
(OLS_{10}) \quad & y_t = \beta_0 + \beta_1 y_{t-1} + \ldots + \beta_{10} y_{t-10} + e_t \\
(OLS_{11}) \quad & y_t = \beta_0 + \beta_1 y_{t-1} + \ldots + \beta_{10} y_{t-10} + \beta_{11} y_{t-11} + e_t \\
(OLS_{12}) \quad & y_t = \beta_0 + \beta_1 y_{t-1} + \ldots + \beta_{10} y_{t-10} + \beta_{11} y_{t-11} + \beta_{12} y_{t-12} + e_t \\
(OLS_{13}) \quad & y_t = \beta_0 + \beta_1 y_{t-1} + \ldots + \beta_{10} y_{t-10} + \beta_{11} y_{t-11} + \beta_{12} y_{t-12} + \beta_{13} y_{t-13} + e_t
\end{aligned}$$
- $\hat{\beta}_{13} \neq 0$ in OLS_{13}: y_t depends on y_{t-13} in (1) as long as $\phi \neq 0$, $\Phi \neq 0$
- $\hat{\beta}_{12} \neq 0$ in OLS_{12}: y_t depends on y_{t-12} in (1) as long as $\Phi \neq 0$
- $\hat{\beta}_{11} \neq 0$ in OLS_{11}: y_t depends on y_{t-12} in (1), y_{t-11} depends on y_{t-12} in (2), there in omitted variable, since y_{t-12} does not appear in OLS_{11}
- $\hat{\beta}_{10} \neq 0$ in OLS_{10}: y_t depends on y_{t-12} in (1), y_{t-10} depends on y_{t-11} in (2), y_{t-11} depends on y_{t-12} in (3), there is omitted variable bias, y_{t-12} does not appear in OLS_{10}

Seasonal Differencing

- we discussed that for economic data that is nonstationarity due to economic growth a common approach is to transform data using a logarithm and apply regular differencing
$$w_t = \Delta \log y_t$$
where $\Delta = 1-L$, so that $w_t = (1-L)\log y_t$
- for economic data that is both nonstationarity due to economic growth and shows seasonal pattern the approach is to transform data using a logarithm and apply both regular and seasonal differencing
$$w_t = \Delta_s \Delta \log y_t$$
where $\Delta = 1-L$ and $\Delta_s = 1-L^s$, so we have $w_t = (1-L^s)(1-L)\log y_t$
- occasionally, when multiple unit roots are present, data has to be diffferenced more than once by applying $\Delta^d = (1-L)^d$ or $\Delta_s^D = (1-L^s)^D$

General Multiplicative ARIMA model

- multiplicative models are written in the form ARIMA$(p, d, q)(P, D, Q)_s$

$$
\begin{aligned}
(1 - \phi_1 L - \ldots \phi_p L^p)(1 - \Phi_1 L^s - \ldots - \Phi_P L^{sP})\Delta_s^D \Delta^d y_t \\
= (1 - \theta_1 L - \ldots \theta_p L^p)(1 - \Theta_1 L^s - \ldots - \Theta_Q L^{sQ})\varepsilon_t
\end{aligned}
$$

- in practice ARIMA$(1, 1, 0)(0, 1, 1)_s$ and ARIMA$(0, 1, 1)(0, 1, 1)_s$ occur routinely

$$
(1 - \phi_1 L)\Delta_s \Delta y_t = (1 - \Theta_1)\varepsilon_t
$$

$$
\Delta_s \Delta y_t = (1 - \theta_1 L)(1 - \Theta_1)\varepsilon_t
$$

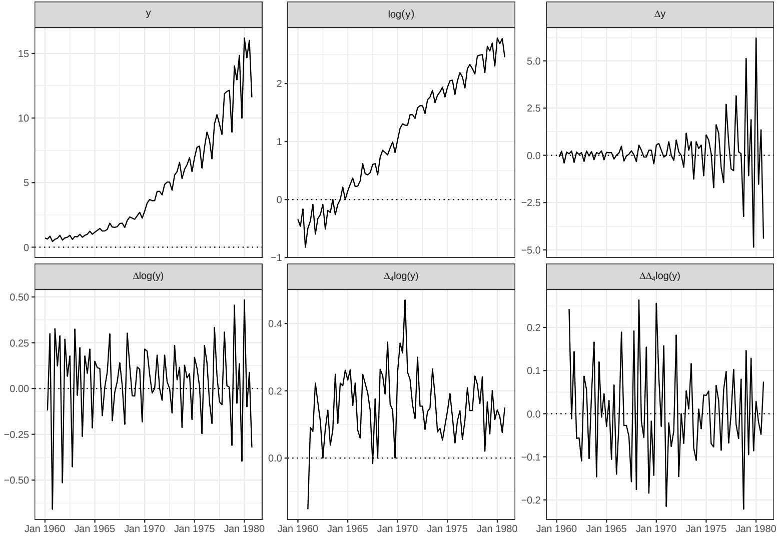
Example: Johnson & Johnson quarterly earnings per share

```r
# load necessary packages
library(readr)
library(dplyr)
library(tidyr)
library(purrr)
library(ggplot2)
library(ggfortify)
library(zoo)
library(timetk)
library(tibbletime)
library(lubridate)
library(forecast)
library(broom)
library(sweep)

# import the data on earnings per share for Johnson and Johnson
# then construct log, change, log-change, seasonal log change
tbl.wide.all <-
    read_table("http://faculty.chicagobooth.edu/ruey.tsay/teaching/fts3/q-jnj.txt", col_names = "y") %>%
    ts(start = c(1960,1), frequency = 4) %>%
    tk_tbl() %>%
    mutate(ly = log(y),
           dy = y - lag(y),
           dly1 = ly - lag(ly),
           dly4 = ly - lag(ly, 4),
           dly4_1 = dly4 - lag(dly4))

# split sample into two parts - estimation sample and prediction sample
fstQ <- 1960.00  # 1960Q1
lstQ <- 1978.75  # 1978Q4
tbl.wide.1 <- tbl.wide.all %>%
    filter(index <= as.yearqtr(lstQ))
```

Original and transformed data

```r
# set default theme for ggplot2
theme_set(theme_bw())

# plot time series: levels, logs, differences
tbl.wide.all %>%
    gather(variable, value, -index) %>%
    mutate(variable.f = factor(variable, ordered = TRUE,
                               levels = c("y", "ly", "dy", "dly1", "dly4", "dly4_1"),
                               labels = c("y", "log(y)",
                                          expression(paste(Delta,"y")),
                                          expression(paste(Delta,"log(y)")),
                                          expression(paste(Delta[4],"log(y)")),
                                          expression(paste(Delta,Delta[4],"log(y)"))))) %>%
    ggplot(aes(x = index, y = value)) +
        geom_hline(aes(yintercept = 0), linetype = "dotted") +
        geom_line() +
        scale_x_yearmon() +
        labs(x = "", y = "") +
        facet_wrap(~variable.f, ncol = 3, scales = "free_y", labeller = label_parsed)
```

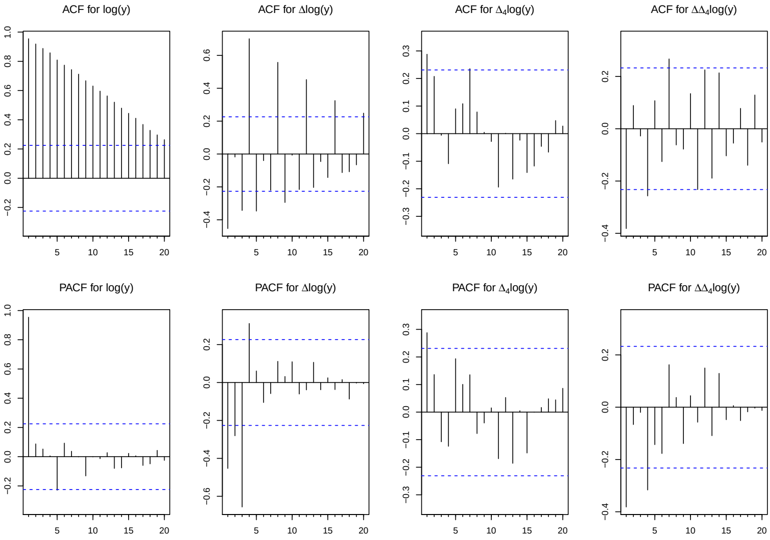
Original and transformed data



ACFs and PACF

```r
# plot ACF and PACF for 20 lags
maxlag <- 20
par(mfrow=c(2,4), mar=c(3,3,4,2))

Acf(tbl.wide.1$ly, type="correlation", lag.max = maxlag, ylab="",
    main = expression(paste("ACF for log(y)")))
Acf(tbl.wide.1$dly1, type="correlation", lag.max = maxlag, ylab="",
    main = expression(paste("ACF for ", Delta,"log(y)")))
Acf(tbl.wide.1$dly4, type="correlation", lag.max = maxlag, ylab="",
    main = expression(paste("ACF for ", Delta[4], "log(y)")))
Acf(tbl.wide.1$dly4_1, type="correlation", lag.max = maxlag, ylab="",
    main = expression(paste("ACF for ", Delta, Delta[4], "log(y)")))
Acf(tbl.wide.1$ly, type="partial", lag.max = maxlag, ylab="",
    main = expression(paste("PACF for log(y)")))
Acf(tbl.wide.1$dly1, type="partial", lag.max = maxlag, ylab="",
    main = expression(paste("PACF for ", Delta, "log(y)")))
Acf(tbl.wide.1$dly4, type="partial", lag.max = maxlag, ylab="",
    main = expression(paste("PACF for ", Delta[4], "log(y)")))
Acf(tbl.wide.1$dly4_1, type="partial", lag.max = maxlag, ylab="",
    main = expression(paste("PACF for ", Delta,Delta[4], "log(y)")))
```

ACFs and PACFs



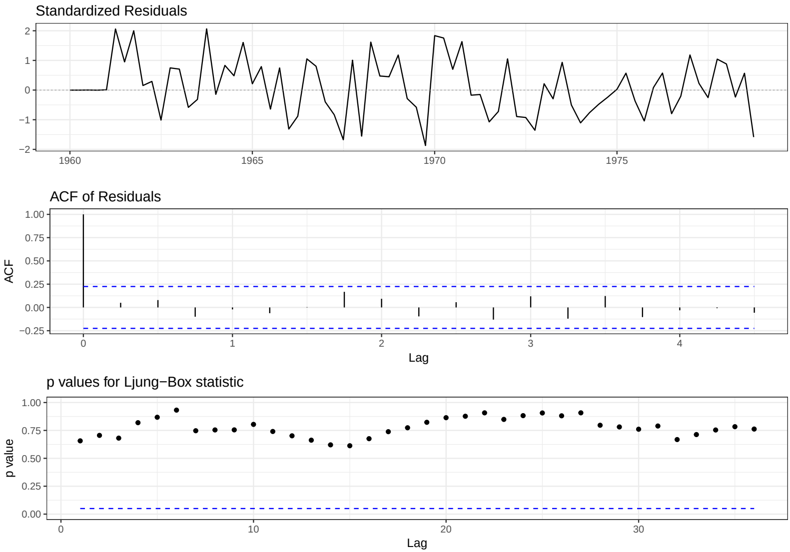
Estimate ARIMA$(0,1,1)(0,1,1)_4$ model

```r
# estimate model
m1 <- tbl.wide.1 %>%
    tk_ts(select = ly, start = fstQ, frequency = 4) %>%
    Arima(order = c(0,1,1), seasonal = list(order = c(0,1,1), period = 4))
m1
```

```
## Series: .
## ARIMA(0,1,1)(0,1,1)[4]
##
## Coefficients:
##           ma1     sma1
##       -0.6559  -0.3492
## s.e.   0.1094   0.1104
##
## sigma^2 estimated as 0.008652:  log likelihood=68.28
## AIC=-130.57   AICc=-130.21   BIC=-123.78
```

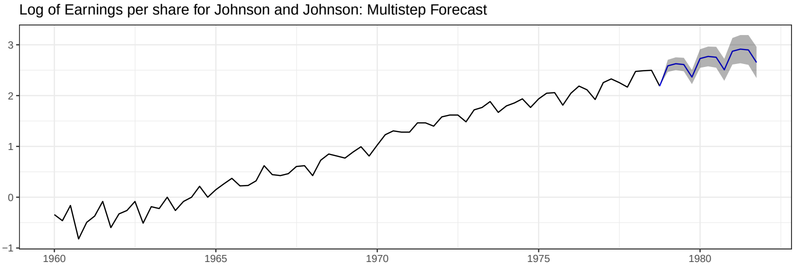
Check Model Adequacy

```
ggtsdiag(m1, gof.lag=36)
```



Forecasts - Multistep

```r
# construct 1-quarter to 12-quarters ahead forecasts
hmax <- 12
m1.f.1.to.hmax <- forecast(m1, hmax)

# plot 1-quarter to 12-quarters ahead forecasts - logs
autoplot(m1.f.1.to.hmax) +
    labs(x = "", y = "",
         title = "Log of Earnings per share for Johnson and Johnson: Multistep Forecast")
```

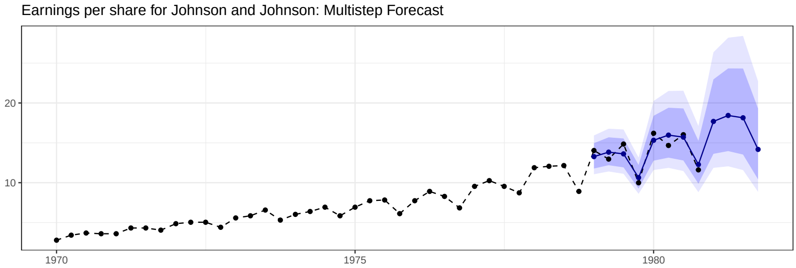


Forecasts - Multistep

```r
# actual data
tbl.2 <-
    tbl.wide.all %>%
    mutate(key = "actual",
           date = as.Date(index)) %>%
    select(date, key, y)

# extract the multistep forecasts, convert to levels
tbl.f.1.to.hmax <-
    m1.f.1.to.hmax %>%
    sw_sweep() %>%
    filter(key == "forecast") %>%
    mutate_at(vars(ly, lo.80, lo.95, hi.80, hi.95), funs(exp)) %>%
    mutate(date = as.Date(index)) %>%
    rename(y = ly) %>%
    select(date, key, y, lo.80, lo.95, hi.80, hi.95)

# forecast & actual data in a single tibble
tbl.f.1.to.hmax <- bind_rows(tbl.2, tbl.f.1.to.hmax)
```

Forecasts - Multistep

```
# plot 1-quarter to 12-quarters ahead forecasts - levels
tbl.f.1.to.hmax %>%
    filter(date >= "1970-01-01") %>%
    ggplot(aes(x = date, y = y, col = key, linetype = key)) +
        geom_ribbon(aes(ymin = lo.95, ymax = hi.95), linetype = "blank", fill = "blue", alpha = 0.1) +
        geom_ribbon(aes(ymin = lo.80, ymax = hi.80), linetype = "blank", fill = "blue", alpha = 0.2) +
        geom_line() +
        geom_point() +
        scale_color_manual(values = c("black","darkblue")) +
        scale_linetype_manual(values = c("dashed","solid")) +
        labs(x = "", y = "",
             title = "Earnings per share for Johnson and Johnson: Multistep Forecast") +
        theme(legend.position = "none")
```

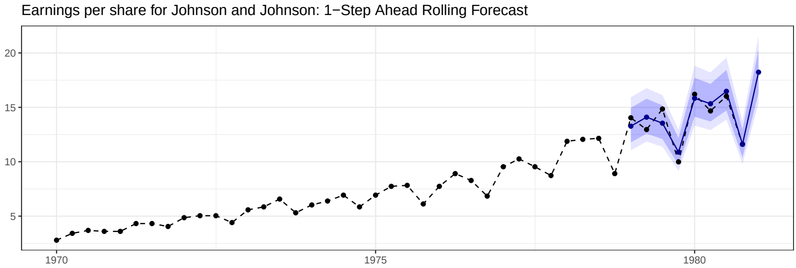


Forecasts - Rolling Scheme

```r
# window length for rolling SARIMA
window.length <- nrow(tbl.wide.1)

# create rolling SARIMA function with rollify from tibbletime package
roll_sarima <- rollify(~Arima(.x, order = c(0,1,1), seasonal = list(order = c(0,1,1), period = 4)),
                      window = window.length, unlist = FALSE)

# estimate rolling SARIMA model, create 1 step ahead forecasts
results <-
    tbl.wide.all %>%
    mutate(date = as.Date(index)) %>%
    as_tbl_time(index = date) %>%
    mutate(sarima.model = roll_sarima(ly)) %>%
    filter(!is.na(sarima.model)) %>%
    mutate(sarima.coefs = map(sarima.model, tidy, conf.int = TRUE),
           sarima.fcst  = map(sarima.model, (. %>% forecast(1) %>% sw_sweep())))

# extract the 1 period ahead rolling forecasts, convert to levels
m1.f.1.rol <-
    results %>%
    select(date, sarima.fcst) %>%
    unnest(sarima.fcst) %>%
    filter(key == "forecast") %>%
    mutate(date = date %m+% months(3)) %>%
    mutate_at(vars(value, lo.80, lo.95, hi.80, hi.95), funs(exp)) %>%
    rename(y = value) %>%
    select(date, key, y, lo.80, lo.95, hi.80, hi.95)

# forecast & actual data in a single tibble
tbl.f.1.rol <- bind_rows(tbl.2, m1.f.1.rol)
```

Forecasts - Rolling Scheme

```r
# plot 1-quarter ahead rolling forecasts - levels
tbl.f.1.rol %>%
    filter(date >= "1970-01-01") %>%
    ggplot(aes(x = date, y = y, col = key, linetype = key)) +
        geom_ribbon(aes(ymin = lo.95, ymax = hi.95), linetype = "blank", fill = "blue", alpha = 0.1) +
        geom_ribbon(aes(ymin = lo.80, ymax = hi.80), linetype = "blank", fill = "blue", alpha = 0.2) +
        geom_line() +
        geom_point() +
        scale_color_manual(values = c("black","darkblue")) +
        scale_linetype_manual(values = c("dashed","solid")) +
        labs(x = "", y = "",
             title = "Earnings per share for Johnson and Johnson: 1-Step Ahead Rolling Forecast") +
        theme(legend.position = "none")
```



Forecasts - Accuracy

```
# convert actual data in prediction sample into ts format
y.ts.2 <- tbl.wide.all %>%
    filter(index > as.yearqtr(lstQ)) %>%
    tk_ts(select = y, start = lstQ+0.25, frequency = 4)
```

```
# evaluate accuracy of forecasts - multistep forecast - logs
accuracy(m1.f.1.to.hmax$mean, log(y.ts.2))
```

```
##                       ME      RMSE        MAE        MPE     MAPE       ACF1 Theil's U
## Test set -0.006368645 0.0640049 0.06092302 -0.3352225 2.347299 -0.8052326 0.2240991
```

```
# evaluate accuracy of forecasts - 1 step ahead rolling scheme forecast - logs
accuracy(log(m1.f.1.rol$y), log(y.ts.2))
```

```
##                       ME       RMSE        MAE        MPE     MAPE       ACF1 Theil's U
## Test set -0.008761921 0.06052803 0.05165746 -0.4089234 2.003344 -0.8127797 0.2103379
```

```
# evaluate accuracy of forecasts - multistep forecast - levels
accuracy(exp(m1.f.1.to.hmax$mean), y.ts.2)
```

```
##                     ME      RMSE       MAE        MPE     MAPE       ACF1 Theil's U
## Test set -0.0379115 0.8878123 0.8350954 -0.8424076 6.124417 -0.7756951 0.2255927
```

```
# evaluate accuracy of forecasts - 1 step ahead rolling scheme forecast - levels
accuracy(m1.f.1.rol$y, y.ts.2)
```

```
##                      ME      RMSE       MAE      MPE    MAPE       ACF1 Theil's U
## Test set -0.08727194 0.8036864 0.6987512 -1.06022 5.20146 -0.7713201 0.1990854
```