

Project Code

Jared Yu

March 19, 2019

```
library(xlsx); library(astsa); library(forecast); library(pracma)
nh <- read.xlsx('Temp-NH.xlsx', sheetIndex = 1)

# Box-Cox not utilized
# lambda <- BoxCox.lambda(nh$Temp.NH); lambda
# nh_trans <- BoxCox(nh$Temp.NH, lambda)
# plot(time_index, nh_trans, type = 'l')
# nh$Temp.NH <- nh_trans

# Span Calculation
n <- nrow(nh)
# (2*q + 1) / n = .5
(.5 * n - 1) / 2 # Test span .5

# Plot the raw data
par(mfrow=c(2,2))
time_index <- rownames(nh)
loess_fit <- loess(Temp.NH~Year, span = .25, data = nh)
plot(time_index, nh$Temp.NH, type = 'l', xlab = 'Time',
      ylab = 'Annual Temperature Anomalies',
      main = 'Annual Temperature Anomalies (1850-2018)')
lines(time_index, loess_fit$fitted, type='l', lty=2, col='red')
legend('topleft', c('Temperature Annomaly', 'Loess'), lty = c(1,2),
      bty = 'n', col = c('black', 'red'))

# plot the rough against time
initial_rough <- loess_fit$residuals
plot(time_index, initial_rough, type = 'l', main = 'Loess: Rough',
      ylab = 'Rough', xlab = 'Time')

# hist of rough
hist(initial_rough, freq = FALSE,
      xlab = 'Rough', main = 'Histogram: Loess rough', breaks = 20)
curve(dnorm(x, mean = mean(initial_rough),
      sd = sd(initial_rough)), add=TRUE)

# QQ-plot
qqnorm(initial_rough); qqline(initial_rough)
qq <- qqnorm(initial_rough)
cor(qq$x, qq$y)
dev.off()

# Shapiro-Wilk test
shapiro.test(initial_rough)
```

```

# Ljung-Box
Box.test(initial_rough, lag = 10, type = "Ljung-Box")

# ACF
par(mfrow=c(2,1))
acf(initial_rough, lag.max = 30,
     main = 'Autocorrelation Function of Loess Residuals')
pacf(initial_rough, lag.max = 50,
     main = 'Partial Autocorrelation Function of Loess Residuals')
dev.off()

### periodogram
par(mfrow=c(2,2))
p_0a <- spec.pgram(initial_rough, log = 'no',
                   main = 'Periodogram of Loess Residuals',
                   ylim = c(0,0.2))
# 3 smoothed periodogram (modified Daniell) spans 5, 15, 25
p_5a <- spec.pgram(x = initial_rough, spans = 5, log = 'no',
                   main = 'Smoothed Periodogram, span = 5', ylim = c(0,0.2))
p_15a <- spec.pgram(x = initial_rough, spans = 15, log = 'no',
                    main = 'Smoothed Periodogram, span = 15', ylim = c(0,0.2))
p_25a <- spec.pgram(x = initial_rough, spans = 25, log = 'no',
                    main = 'Smoothed Periodogram, span = 25', ylim = c(0,0.2))
dev.off()

loess_fit
0.1897^2

# Preliminary model
# MA(3), AR(6)
# test_fit <- arima(initial_rough, order = c(3,0,4))
test_fit <- arima(initial_rough, order = c(3,0,6),
                  optim.control = list(maxit=1000))
# optim gave code 1 warning error
test_rough <- test_fit$residuals

par(mfrow=c(2,2))
# plot the rough against time
plot(time_index, test_rough, type = 'l', main = 'Preliminary Model: Rough',
     ylab = 'Rough', xlab = 'Time')

# hist of rough
hist(test_rough, freq = FALSE,
     xlab = 'Rough', main = 'Histogram: Preliminary Model: Rough', breaks = 20)
curve(dnorm(x, mean = mean(test_rough),
            sd = sd(test_rough)), add=TRUE)

# QQ-plot
qqnorm(test_rough); qqline(test_rough)
qq <- qqnorm(test_rough)
cor(qq$x, qq$y)
dev.off()

```

```

# Shapiro-Wilk test
shapiro.test(test_rough)

# Ljung-Box
Box.test(test_rough, lag = 10, type = "Ljung-Box")

# ACF
par(mfrow=c(2,1))
acf(test_rough, lag.max = 30,
    main = 'Autocorrelation Function of Preliminary Model Rough')
pacf(test_rough, lag.max = 50,
    main = 'Partial Autocorrelation Function of Preliminary Model Rough')
dev.off()

par(mfrow=c(2,2))
p_0b <- spec.pgram(test_rough, log = 'no',
    main = 'Periodogram of Preliminary Model Rough',
    ylim = c(0,0.12))
# 3 smoothed periodogram (modified Daniell) spans 5, 15, 25
p_5b <- spec.pgram(x = test_rough, spans = 5, log = 'no',
    main = 'Smoothed Periodogram, span = 5', ylim = c(0,0.12))
p_15b <- spec.pgram(x = test_rough, spans = 15, log = 'no',
    main = 'Smoothed Periodogram, span = 15', ylim = c(0,0.12))
p_25b <- spec.pgram(x = test_rough, spans = 25, log = 'no',
    main = 'Smoothed Periodogram, span = 25', ylim = c(0,0.12))
dev.off()

### auto
# auto_model <- auto.arima(loess_fit$residuals, ic = 'aicc',
#                             stepwise = FALSE, approximation = FALSE)

### manual
fitting_sarima = function(i, k, rough_data = loess_fit$residuals){
  sarima(rough_data, p = i, d = 0, q = k, Model = FALSE, details = FALSE)
}

pq_choices <- data.frame(p = rep(0:4, each = 5), q = rep(0:4, 5))
all_sarima <- mapply(fitting_sarima, pq_choices$p, pq_choices$q)
npair = dim(all_sarima)[2]
AICC_result = sapply(1:npair, function(x) all_sarima[,x]$AICc)
pq <- pq_choices[which.min(AICC_result),]
arima_p0q <- sarima(loess_fit$residuals, # Fit ARIMA(p,0,q) model
    p = pq[,1], d = 0, q = pq[,2],
    details = FALSE, Model = FALSE)

sink('est.txt')
arima_p0q
sink()

sink('ttable.txt')
arima_p0q$ttable
sink()

# diagnostic on final model

```

```

final_model <- arima(loess_fit$residuals, order = c(4,0,4))
final_model_rough <- final_model$residuals

par(mfrow=c(2,2))
final_model_fitted <- nh$Temp.NH - final_model$residuals
plot(time_index, nh$Temp.NH, type = 'l', xlab = 'Time',
      ylab = 'Annual Temperature Anomalies',
      main = 'Annual Temperature Anomalies (1850-2018)')
lines(time_index, final_model_fitted, type = 'l', lty = 2, col = 'red')
legend('topleft', c('Temperature Annomaly', 'Fitted Final Model Values'),
      lty = c(1,2), bty = 'n', col = c('black', 'red'))

plot(time_index, final_model_rough, type = 'l',
      xlab = 'Time', ylab = 'Rough', main = 'Final Model: Rough')

# hist of rough
hist(final_model_rough, freq = FALSE,
      xlab = 'Rough', main = 'Histogram: Final Model Rough', breaks = 20)
curve(dnorm(x, mean = mean(final_model_rough),
      sd = sd(final_model_rough)), add=TRUE)

# QQ-plot
qqnorm(final_model_rough); qqline(final_model_rough)
qq <- qqnorm(final_model_rough)
cor(qq$x, qq$y)
dev.off()

# Shaprio-Wilk test
shapiro.test(final_model_rough)

# Ljung-Box
Box.test(final_model_rough, lag = 10, type = "Ljung-Box")

# ACF/PACF
par(mfrow=c(3,1))
plot(final_model_rough, ylab = 'Rough',
      main = 'Loess Residuals Modeled with ARIMA(4,0,4)')
acf(final_model_rough, lag.max = 30,
      main = 'ACF of Loess Residuals Modeled with ARIMA(4,0,4)')
pacf(final_model_rough, lag.max = 50,
      main = 'PACF of Loess Residuals Modeled with ARIMA(4,0,4)')
dev.off()

# Periodogram
par(mfrow=c(2,2))
p_0c <- spec.pgram(final_model_rough, log = 'no',
      main = 'Periodogram of Loess Residuals Modeled with ARIMA(4,0,4)',
      ylim = c(0,0.1))
# 3 smoothed periodogram (modified Daniell) spans 5, 15, 25
p_5c <- spec.pgram(x = final_model_rough, spans = 5, log = 'no',
      main = 'Smoothed Periodogram, span = 5', ylim = c(0,0.1))
p_15c <- spec.pgram(x = final_model_rough, spans = 15, log = 'no',
      main = 'Smoothed Periodogram, span = 15', ylim = c(0,0.1))

```

```

p_25c <- spec.pgram(x = final_model_rough, spans = 25, log = 'no',
                    main = 'Smoothed Periodogram, span = 25', ylim = c(0,0.1))
dev.off()

specselect = function(y,kmax){
  # Obtains the values of the criterion function for
  # obtaining the the optimal number of neighbors for
  # spectral density estimate for modified Daniell's method.
  # input: y, observed series; kmax=max number of nighbors to
  # be considered
  # output: ctr - the criterion function
  # output: kopt - the value of k at which the criterion function
  # is minimized
  ii=spec.pgram(y,log="no",plot=FALSE)
  ii=ii$spec
  cc=norm(as.matrix(ii),type="F")^2
  ctr=rep(1,kmax)
  for(k in 1:kmax) {
    ss=2*k+1; kk=1/(2*k)
    ff=spec.pgram(y,spans=ss,log="no",plot=FALSE)
    fspec=ff$spec
    ctr[k]=norm(as.matrix(ii-fspec),type="F")^2+kk*cc
  }
  kopt=which.min(ctr)
  result=list(ctr=ctr,kopt=kopt)
  return(result)
}

arma_fit <- arma.spec(ar = arima_p0q$fit$coef[1:4],
                      ma = arima_p0q$fit$coef[5:8], log = 'no',
                      var.noise = arima_p0q$fit$sigma2)

par(mfrow=c(2,1))
n <- nrow(nh)
model_kmax <- floor((n - 1) / 2)
spec_select_plot <- specselect(final_model_rough, kmax = model_kmax)
plot(1:model_kmax, spec_select_plot$ctr, type = 'l',
     xlab = 'Number of Neighbors (k)', ylab = 'Criterion Value',
     main = 'Criterion Function of Loess Residuals Modeled with ARIMA(4,0,4)')
points(1:length(spec_select_plot$ctr), spec_select_plot$ctr)
abline(h = spec_select_plot$ctr[12], lty = 2)
abline(v = 12, lty = 2)
legend('topright', legend = c('Criterion Value', 'k = 12'),
      lty = c(1,2), bty = 'n')
p_25d <- spec.pgram(x = final_model_rough, spans = 21, log = 'no',
                    main = 'Smoothed Periodogram, span = 21', ylim = c(0,0.1))
dev.off()

par(mfrow=c(2,1))
spec_select_plot_b <- specselect(initial_rough, kmax = model_kmax)
plot(1:model_kmax, spec_select_plot_b$ctr, type = 'l',
     xlab = 'Number of Neighbors (k)', ylab = 'Criterion Value',
     main = 'Criterion Function of Loess Residuals')

```

```

points(1:length(spec_select_plot_b$ctr), spec_select_plot_b$ctr)
abline(h = spec_select_plot_b$ctr[8], lty = 2)
abline(v = 8, lty = 2)
legend('topright', legend = c('Criterion Value', 'k = 8'),
      lty = c(1,2), bty = 'n')
initial_rough_span <- 2 * 8 + 1
p_25e <- spec.pgram(x = initial_rough, spans = initial_rough_span, log = 'no',
                  main = 'Smoothed Periodogram, span = 17', ylim = c(0,0.09))
lines(arma_fit$freq, arma_fit$spec, lty = 2)
legend('topleft', legend = c('Smoothed Periodogram', 'Spectral Density'),
      lty = c(1,2), bty = 'n')
dev.off()

### linear extrapolation
nh_forecast <- head(nh, -6)
n2 <- nrow(nh_forecast)
nh_forecast_fit <- loess(Temp.NH~Year, data = nh_forecast)
nh_forecast_arima <- arima(nh_forecast_fit$residuals, order = c(4,0,4))
rough_pred <- predict(nh_forecast_arima, n.ahead = 6)

last_2 <- tail(nh_forecast_fit$fitted, 2)

linear_extrapolation <- function(x1, x2, y1, y2, x) {
  y1 + ((y2 - y1) / (x2 - x1)) * (x - x1)
}

m_6 <- linear_extrapolation(x1 = (n2 - 1), x2 = n2, y1 = last_2[1],
                          y2 = last_2[2], x = (n2 + 1:6))

pred_val <- as.numeric(rough_pred$pred) + m_6

plot(time_index, nh$Temp.NH,
     main = 'Forecast of Final Model Using Linear Extrapolation (n-6,...,n)',
     xlab = 'Time Index', ylab = 'Temperature Anomalies',
     type = 'l')
points(nh$Temp.NH)
pred_vec <- c(rep(NA, length(time_index) - 6), pred_val)
points(time_index, pred_vec, col = 'red', pch = 22)
lines(time_index, pred_vec, col = 'red', pch = 22)
legend("topleft", legend = c('Observed', 'Predicted'),
      col = c('black', 'red'), pch = c(1,7), lty = c(1, NA))

```