# STAT 626 Project - Version 3

Ken Marciel

7/30/2021

### Packages

```
library(readxl) # for read_excel function
library(astsa) # for time series functions
```

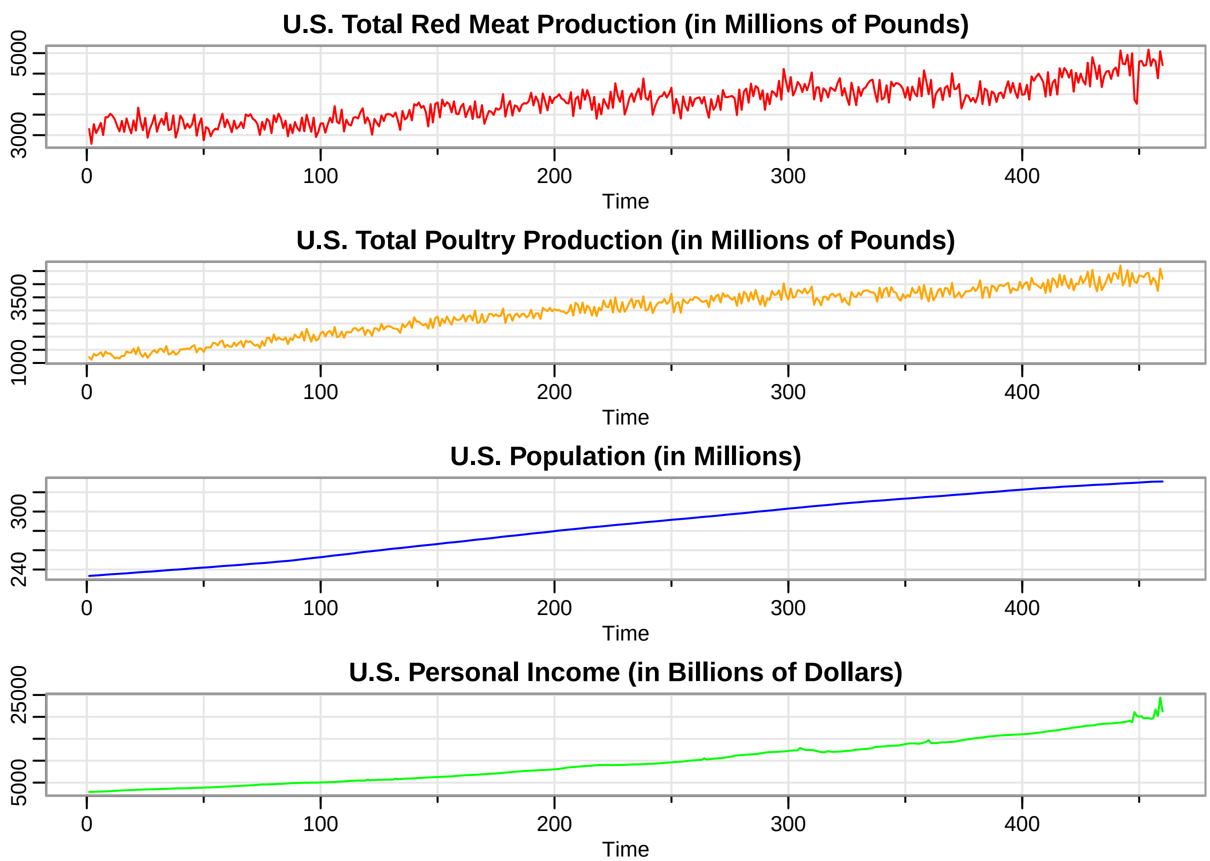
### Data

```
# Read data from Excel file (compiled by Jack Kramer)
setwd("C:/Users/keoka/OneDrive - Texas A&M University/Courses/STAT_626/Project/Data Analysis/Jack/")
data = read.csv("consolidated_data.csv")
class(data) # data frame
dim(data) # 460 rows, 41 columns
head(data) # preview the first six rows of data set
names(data) # display the column names of data set

# Keep only the columns desired for analysis
match("year", colnames(data)) # column 41
match("month", colnames(data)) # column 40
match("total.red.meat", colnames(data)) # column 7
match("total.poultry", colnames(data)) # column 16
match("Population_Millions", colnames(data)) # column 21
match("Personal.Income", colnames(data)) # column 19
data = data[,c(41,40,7,16,21,19)]

# Change column names to have a consistent format
colnames(data) = c("Year", "Month","Red_Meat", "Poultry", "Population",
                   "Personal_Income")

# Review structure of the data set
class(data) # data frame
dim(data) # 460 rows, 6 columns
head(data) # preview the first six rows of reduced data set
str(data)

# Create variables to store data columns
year = data[,1]
mont = data[,2]
meat = data[,3] # total U.S. red meat production in millions of pounds
poul = data[,4] # total U.S. poultry production in millions of pounds
popu = data[,5] # U.S. population in millions
inco = data[,6] # U.S. personal income in billions of dollars
```

```r
# Range of observations for the Great Depression
data[which(data$Year==2007 & data$Month==12),] # row 300
data[which(data$Year==2009 & data$Month==6),] # row 318

# Start of observations for the COVID-19 Pandemic
data[which(data$Year==2020 & data$Month==3),] # row 447
```

### Exploratory Data Analysis: Time Series Plotted Separately

```r
par(mfrow=c(4,1))
tsplot(meat, main="U.S. Total Red Meat Production (in Millions of Pounds)",
       col="red", ylab="")
tsplot(poul, main="U.S. Total Poultry Production (in Millions of Pounds)",
       col="orange", ylab="")
tsplot(popu, main="U.S. Population (in Millions)",
       col="blue", ylab="")
tsplot(inco, main="U.S. Personal Income (in Billions of Dollars)",
       col="green", ylab="")
```

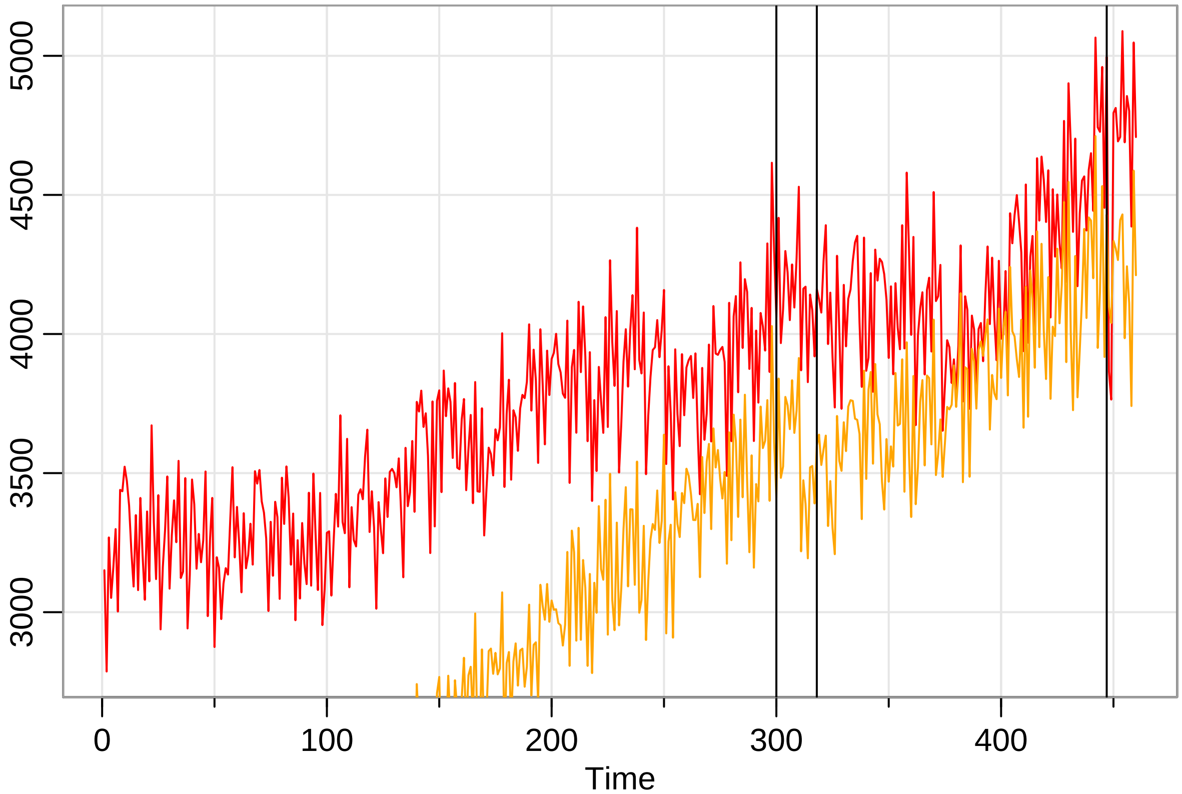



Note the upward trend in all of the series, and the apparent seasonality in the meat and poultry series.

### Exploratory Data Analysis: Time Series Plotted Together

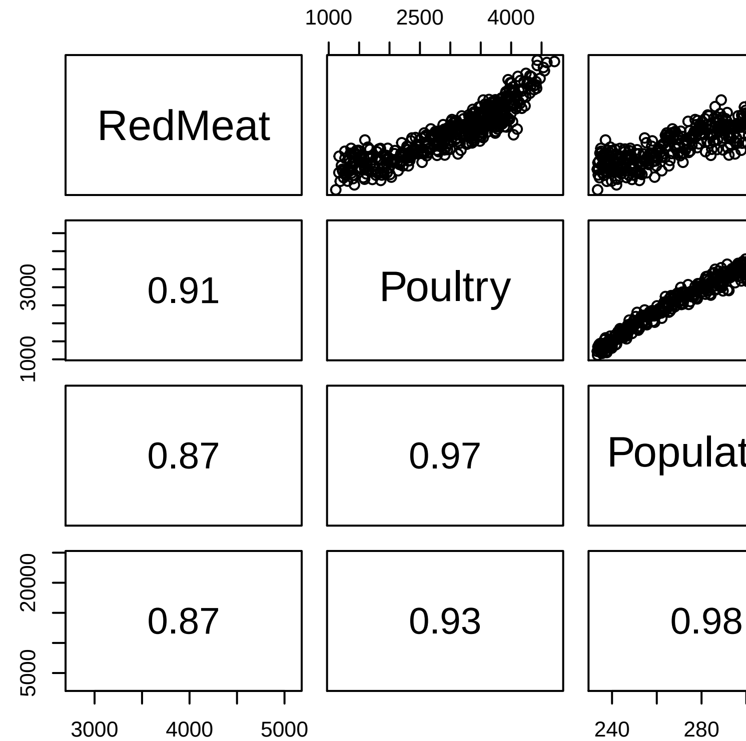
```r
tsplot(meat, main="", ylab="", col="red")
lines(poul, col="orange")
```

```r
abline(v=c(300,318,447))
```




Note that poultry has a steeper upward trend than red meat. Both series display a downard trend during the Great Recession (December 2007 to June 2009), and a leveling off during the COVID-19 pandemic (March 2020 to present). Both series have similar seasonality.

```r
# Code from page 43 of textbook: Shumway, Stoffer (2019)
panel.cor = function(x, y, ...){
  usr = par("usr"); on.exit(par(usr))
  par(usr = c(0,1,0,1))
  r = round(cor(x, y), 2)
  text(0.5, 0.5, r, cex=1.75)
}
pairs(cbind(RedMeat=meat, Poultry=poul, Population=popu, Income=inco), lower.panel=panel.cor)
```




**Exploratory Data Analysis: Scatterplot Matrix**

The scatterplot matrix indicates that red meat is nonlinearly related to population and income, and the same is true for poultry. Population and income are highly correlated, and have nearly identical correlation coefficients for red meat and poultry.

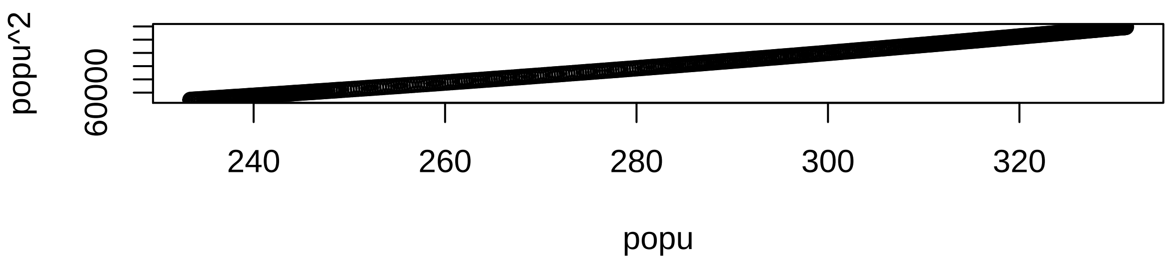
For ease, let $M_t$ denote total red meat production, $P_t$ denote total poultry production, $L_t$ denote total population, and $I_t$ denote total personal income. Also, let $L.$ and $I.$ denote the mean of population, and the mean of personal income, respectively.

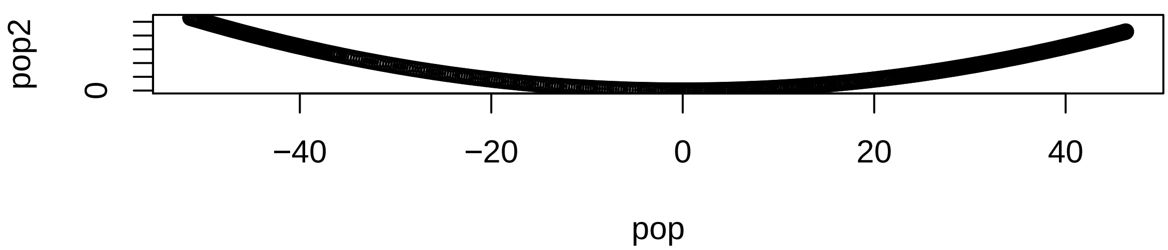
```r
par(mfrow=2:1)
plot(popu, popu^2) # collinear
cor(popu, popu^2)
```

**Exploratory Data Analysis: Collinearity**

```
## [1] 0.9990233
```

```r
pop = popu - mean(popu) # center population
pop2 = pop^2
plot(pop, pop2) # not collinear
```



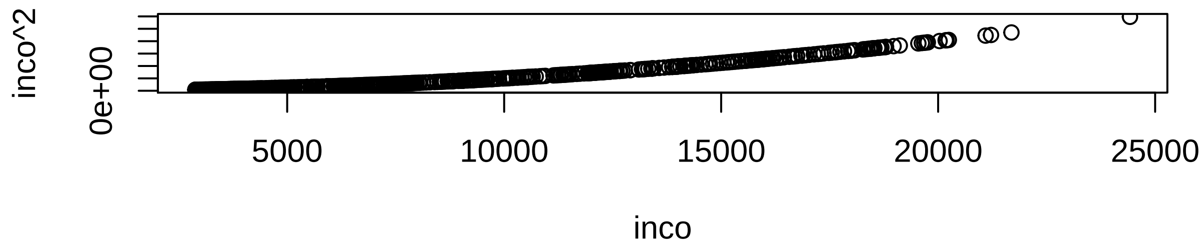

```
cor(pop, pop2)
```

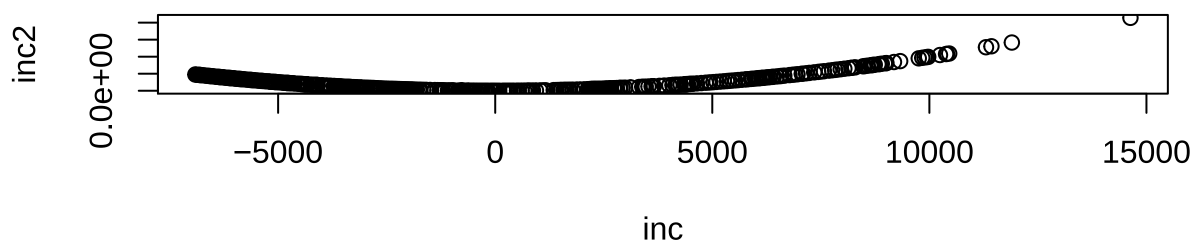
```
## [1] -0.1503336
```

```
par(mfrow=2:1)
plot(inco, inco^2) # collinear
cor(inco, inco^2)
```

```
## [1] 0.9763314
```

```
inc = inco - mean(inco) # center income
inc2 = inc^2
plot(inc, inc2)
```




```r
cor(inc, inc2)
```

```
## [1] 0.4068634
```

For this data set, $L_t$ and $L_t^2$ are highly collinear, but $L_t - L.$ and $(L_t - L.)^2$ are not. Similarly, $I_t$ and $I_t^2$ are highly collinear, while $I_t - I.$ and $(I_t - I.)^2$ are moderately collinear. Therefore, it is better to include $L_t$ instead of $I_t$ in the model.

### Model Formulation: Multiple Linear Regression

Based on the scatterplot, three models will be entertained for each of red meat and poultry. They are

M1: $M_t = \beta_0 + \beta_1 + w_t$
M2: $M_t = \beta_0 + \beta_1 + \beta_2(L_t - L.) + w_t$
M3: $M_t = \beta_0 + \beta_1 + \beta_2(L_t - L.) + \beta_3(L_t - L.)^2 + w_t$

P1: $P_t = \beta_0 + \beta_1 + w_t$
P2: $P_t = \beta_0 + \beta_1 + \beta_2(L_t - L.) + w_t$
P3: $P_t = \beta_0 + \beta_1 + \beta_2(L_t - L.) + \beta_3(L_t - L.)^2 + w_t$

where we adjust population for its mean, $L.$, to avoid collinearity problems.

Note that M1 and P1 are trend only models, M2 and P2 add a linear population term, M3 and P3 add a curvilinear population term.

### Multiple Linear Regression: Total Meat Production

```r
trend_m = time(meat)
num_m = length(meat) # 460 observations
```

```r
# Model M1
fit_m1 = lm(meat ~ trend_m, na.action=NULL)
(m1_out = summary(fit_m1))
```

```
##
## Call:
## lm(formula = meat ~ trend_m, na.action = NULL)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -687.72 -146.80   13.56  166.01  634.52
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.109e+03  2.105e+01  147.68   <2e-16 ***
## trend_m     2.991e+00  7.914e-02   37.79   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 225.4 on 458 degrees of freedom
## Multiple R-squared:  0.7572, Adjusted R-squared:  0.7567
## F-statistic:  1428 on 1 and 458 DF,  p-value: < 2.2e-16
```

```r
summary(aov(fit_m1))
```

```
##              Df   Sum Sq  Mean Sq F value Pr(>F)
## trend_m       1 72568456 72568456    1428 <2e-16 ***
## Residuals   458 23268197    50804
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
AIC(fit_m1)/num_m - log(2*pi)
```

```
## [1] 11.84442
```

```r
BIC(fit_m1)/num_m
```

```
## [1] 13.70923
```

```r
rss_m1 = m1_out$sigma

# Model M2
fit_m2 = lm(meat ~ trend_m + pop, na.action=NULL)
(m2_out = summary(fit_m2))
```

```
##
## Call:
## lm(formula = meat ~ trend_m + pop, na.action = NULL)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -682.01 -146.13   15.06  164.16  639.33
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3171.069    274.895  11.536   <2e-16 ***
```

```
## trend_m        2.722      1.192   2.284   0.0228 *
## pop            1.179      5.206   0.226   0.8210
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 225.6 on 457 degrees of freedom
## Multiple R-squared:  0.7572, Adjusted R-squared:  0.7562
## F-statistic: 712.7 on 2 and 457 DF,  p-value: < 2.2e-16
```

```
summary(aov(fit_m2))
```

```
##              Df   Sum Sq  Mean Sq  F value Pr(>F)
## trend_m       1 72568456 72568456 1425.444 <2e-16 ***
## pop           1     2609     2609    0.051  0.821
## Residuals   457 23265587    50909
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(fit_m2)/num_m - log(2*pi)
```

```
## [1] 11.84865
```

```
BIC(fit_m2)/num_m
```

```
## [1] 13.72245
```

```
rss_m2 = m2_out$sigma

# Model M3
fit_m3 = lm(meat ~ trend_m + pop + pop2, na.action=NULL)
(m3_out = summary(fit_m3))
```

```
##
## Call:
## lm(formula = meat ~ trend_m + pop + pop2, na.action = NULL)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -680.54 -147.82   14.43  161.39  638.49
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.411e+03  3.805e+02   8.964   <2e-16 ***
## trend_m     1.607e+00  1.706e+00   0.942    0.347
## pop         6.106e+00  7.499e+00   0.814    0.416
## pop2        1.808e-02  1.980e-02   0.913    0.362
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 225.7 on 456 degrees of freedom
## Multiple R-squared:  0.7577, Adjusted R-squared:  0.7561
## F-statistic: 475.3 on 3 and 456 DF,  p-value: < 2.2e-16
```

```
summary(aov(fit_m3))
```

```
##              Df   Sum Sq  Mean Sq  F value Pr(>F)
## trend_m       1 72568456 72568456 1424.925 <2e-16 ***
## pop           1     2609     2609    0.051  0.821
```

```
## pop2          1    42465    42465    0.834  0.362
## Residuals   456 23223123    50928
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(fit_m3)/num_m - log(2*pi)
```

```
## [1] 11.85117
```

```
BIC(fit_m3)/num_m
```

```
## [1] 13.73395
```

```
rss_m3 = m3_out$sigma

# Comparison of M1 (reduced) to M2 (full) model using residual sums of squares
((rss_m1 - rss_m2)/(3-2))/(rss_m2/(460-3-1)) # 4.1159 significant
```

```
## [1] -0.4725471
```

```
qf(.05, 3-2, 460-3-1, lower.tail=FALSE) # 3.8619
```

```
## [1] 3.861932
```

```
# Comparison of M1 (reduced) to M3 (full) model using residual sums of squares
((rss_m1 - rss_m3)/(4-2))/(rss_m3/(460-4-1)) # 2.1425 nonsignificant
```

```
## [1] -0.2770796
```

```
qf(.05, 3-2, 460-3-1, lower.tail=FALSE) # 3.8619
```

```
## [1] 3.861932
```

R-squared is nearly the same for all three models for meat production. M1 has the lowest AIC and BIC scores of the three models. Both of the estimated coefficients for M1 are statistically significant. Comparison of M1 (reduced) to M2 (full) models using residual sums of squares produced a statistically significant F-test, whereas comparison of M1 to M3 did not. However, the estimated coefficients of the linear and quadradtic terms, in M2 and M3, were not statistically significant. Therefore, M1 does the best of the three models.

### Multiple Linear Regression: Total Poultry Production

```
trend_p = time(poul)
num_p = length(poul)

# Model P1
fit_p1 = lm(poul ~ trend_p, na.action=NULL)
(p1_out = summary(fit_p1))
```

```
##
## Call:
## lm(formula = poul ~ trend_p, na.action = NULL)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -704.93 -185.50   -3.61  177.14  629.47
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.448e+03  2.163e+01   66.96   <2e-16 ***
```

```
## trend_p     6.546e+00  8.131e-02   80.50   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 231.6 on 458 degrees of freedom
## Multiple R-squared:  0.934,  Adjusted R-squared:  0.9339
## F-statistic:  6481 on 1 and 458 DF,  p-value: < 2.2e-16
```

```
summary(aov(fit_p1))
```

```
##               Df    Sum Sq   Mean Sq F value Pr(>F)
## trend_p        1 347532050 347532050    6481 <2e-16 ***
## Residuals    458  24559260     53623
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(fit_p1)/num_p - log(2*pi)
```

```
## [1] 11.89842
```

```
BIC(fit_p1)/num_p
```

```
## [1] 13.76324
```

```
rss_p1 = p1_out$sigma

# Model P2
fit_p2 = lm(poul ~ trend_p + pop, na.action=NULL)
(p2_out = summary(fit_p2))
```

```
##
## Call:
## lm(formula = poul ~ trend_p + pop, na.action = NULL)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -508.45 -135.67    4.86  134.34  589.22
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4278.504    249.275  17.164  < 2e-16 ***
## trend_p       -5.733      1.081  -5.305 1.76e-07 ***
## pop           53.759      4.721  11.387  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 204.6 on 457 degrees of freedom
## Multiple R-squared:  0.9486, Adjusted R-squared:  0.9484
## F-statistic:  4216 on 2 and 457 DF,  p-value: < 2.2e-16
```

```
summary(aov(fit_p2))
```

```
##               Df    Sum Sq   Mean Sq F value Pr(>F)
## trend_p        1 347532050 347532050  8301.8 <2e-16 ***
## pop            1   5428313   5428313   129.7 <2e-16 ***
## Residuals    457  19130947     41862
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
AIC(fit_p2)/num_p - log(2*pi)
```

```
## [1] 11.65298
```

```r
BIC(fit_p2)/num_p
```

```
## [1] 13.52678
```

```r
rss_p2 = p2_out$sigma

# Model P3
fit_p3 = lm(poul ~ trend_p + pop + pop2, na.action=NULL)
(p3_out = summary(fit_p3))
```

```
##
## Call:
## lm(formula = poul ~ trend_p + pop + pop2, na.action = NULL)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -445.65 -109.53    0.46  105.22  598.55
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1603.4964   294.0508   5.453 8.13e-08 ***
## trend_p        6.6791     1.3184   5.066 5.91e-07 ***
## pop           -1.1104     5.7947  -0.192    0.848
## pop2          -0.2013     0.0153 -13.158  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 174.4 on 456 degrees of freedom
## Multiple R-squared:  0.9627, Adjusted R-squared:  0.9625
## F-statistic:  3927 on 3 and 456 DF,  p-value: < 2.2e-16
```

```r
summary(aov(fit_p3))
```

```
##              Df    Sum Sq   Mean Sq F value Pr(>F)
## trend_p       1 347532050 347532050 11428.9 <2e-16 ***
## pop           1   5428313   5428313   178.5 <2e-16 ***
## pop2          1   5264801   5264801   173.1 <2e-16 ***
## Residuals   456  13866146     30408
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
AIC(fit_p3)/num_p - log(2*pi)
```

```
## [1] 11.33547
```

```r
BIC(fit_p3)/num_p
```

```
## [1] 13.21826
```

```r
rss_p3 = p3_out$sigma

# Model P4
fit_p4 = lm(poul ~ trend_p + pop2, na.action=NULL)
(p4_out = summary(fit_p4))
```

```
##
## Call:
## lm(formula = poul ~ trend_p + pop2, na.action = NULL)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -446.8 -109.6    0.4  106.0  600.6
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1659.71755   19.78891   83.87   <2e-16 ***
## trend_p        6.42678    0.06149  104.52   <2e-16 ***
## pop2          -0.19919    0.01061  -18.77   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 174.2 on 457 degrees of freedom
## Multiple R-squared:  0.9627, Adjusted R-squared:  0.9626
## F-statistic:  5903 on 2 and 457 DF,  p-value: < 2.2e-16
```

```
summary(aov(fit_p4))
```

```
##              Df    Sum Sq   Mean Sq F value Pr(>F)
## trend_p       1 347532050 347532050 11453.0 <2e-16 ***
## pop2          1  10691997  10691997   352.4 <2e-16 ***
## Residuals   457  13867263     30344
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
AIC(fit_p4)/num_p - log(2*pi)
```

```
## [1] 11.33121
```

```
BIC(fit_p4)/num_p
```

```
## [1] 13.20501
```

```
rss_p4 = p4_out$sigma

# Comparison of P1 (reduced) to P2 (full) model using residual sums of squares
((rss_p1 - rss_p2)/(3-2))/(rss_p2/(460-3-1)) # 4.1159 significant
```

```
## [1] 60.09494
```

```
qf(.05, 3-2, 460-3-1, lower.tail=FALSE) # 3.8619
```

```
## [1] 3.861932
```

```
# Comparison of P1 (reduced) to P3 (full) model using residual sums of squares
((rss_p1 - rss_p3)/(4-2))/(rss_p3/(460-4-1)) # 2.1425 nonsignificant
```

```
## [1] 74.60703
```

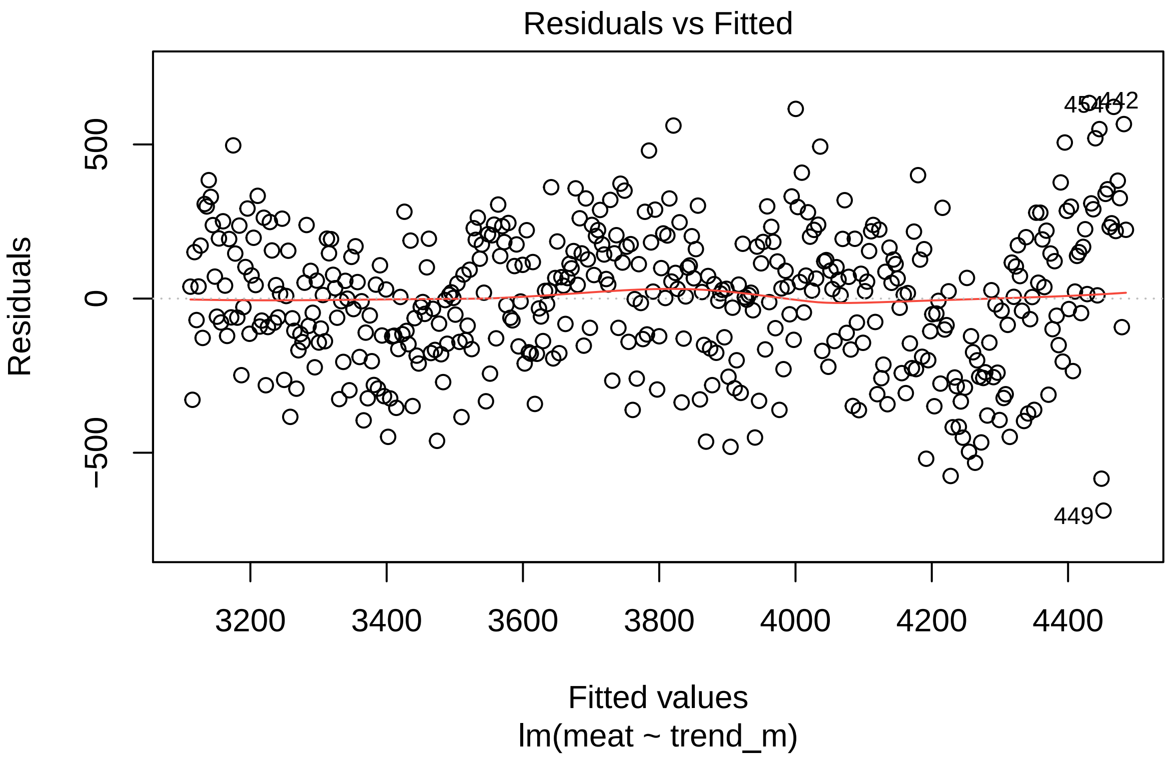
```
qf(.05, 3-2, 460-3-1, lower.tail=FALSE) # 3.8619
```

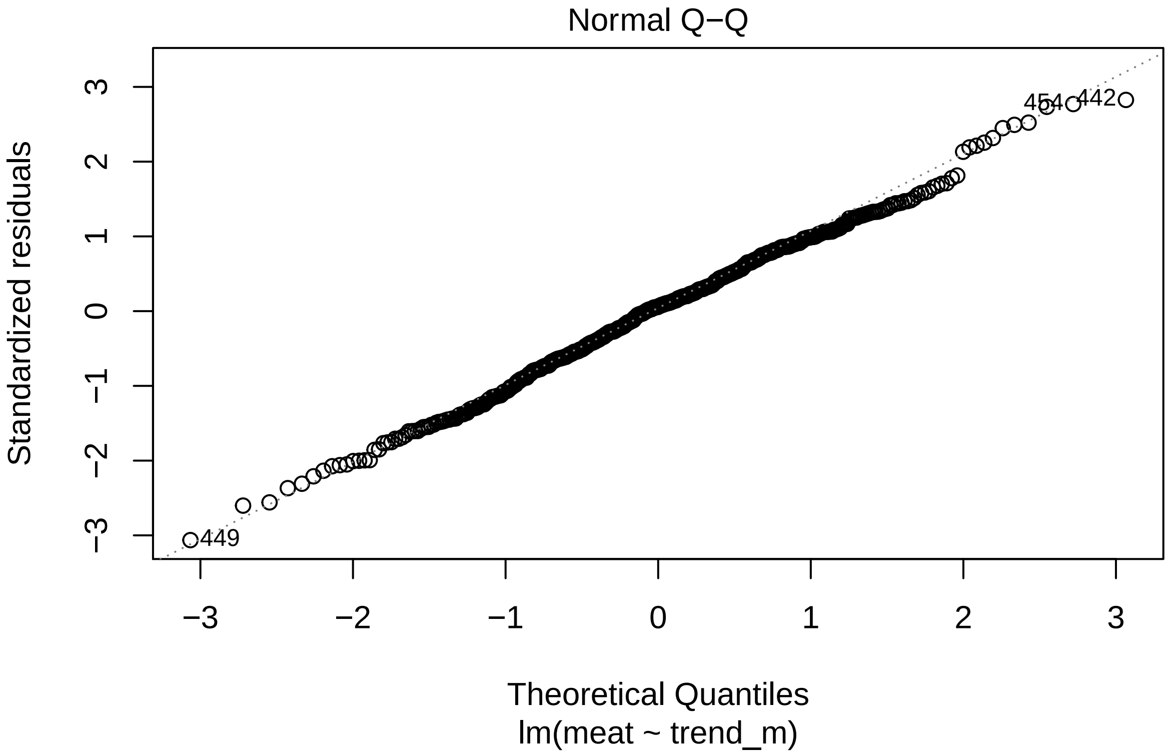
```
## [1] 3.861932
```

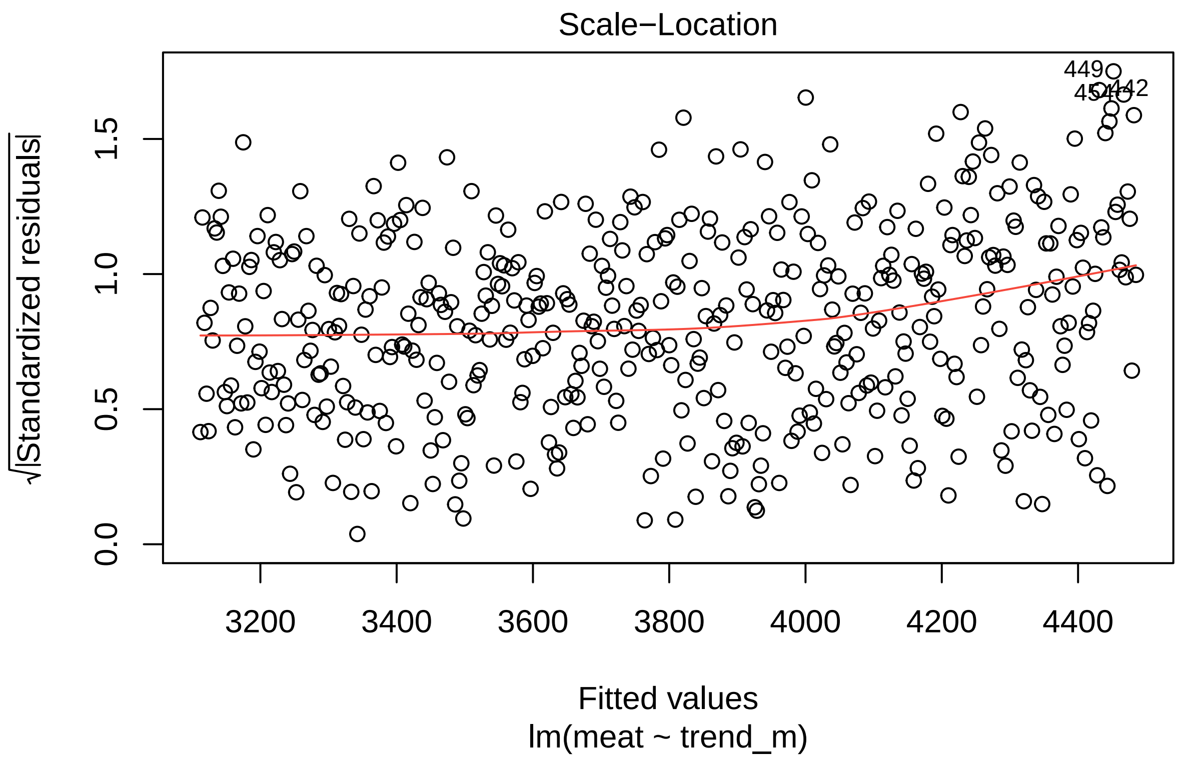
P3 has the highest R-squared, as well as the lowest AIC and BIC of the three poultry models. The income term, $I$, in the P3 model has an estimated coefficient that is not statistically significant. However, when a fourth model was fitted with that term removed, there is essentially no change in R-squared, AIC, or BIC. Therefore, I will leave the linear term in the model and select P3 from the three poultry models.

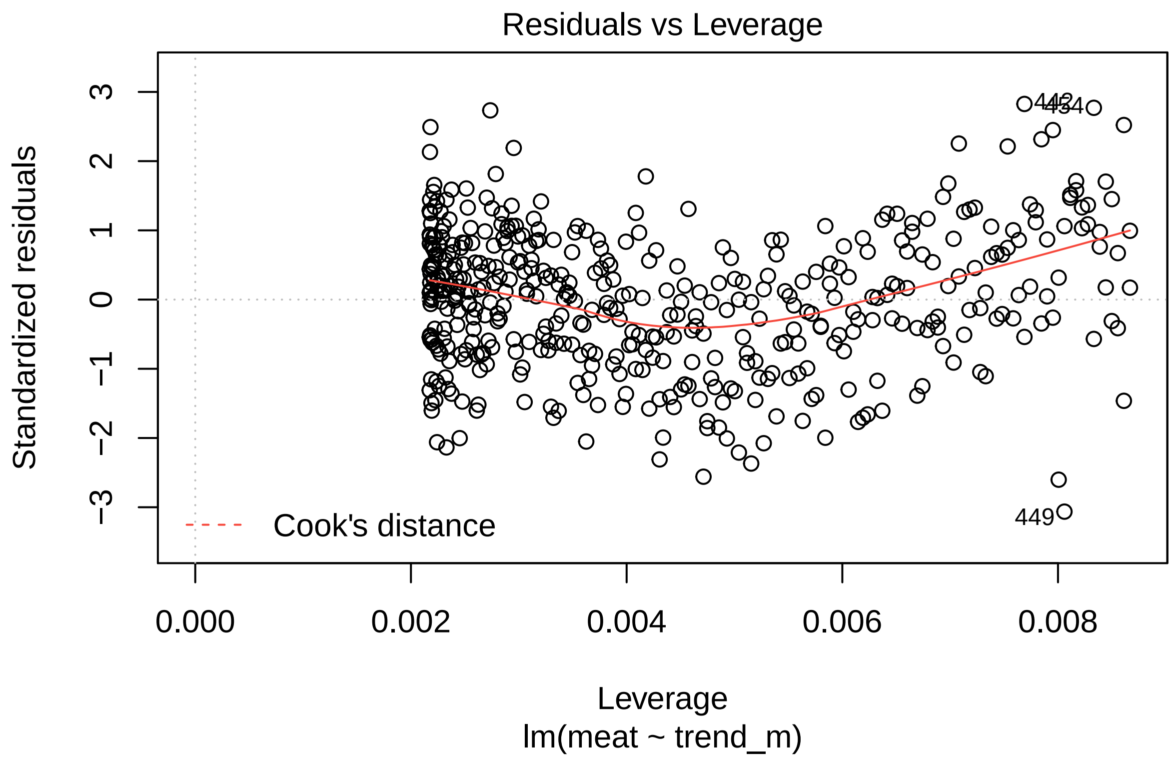
**Model Formulation: Total Meat Production**

```
plot(fit_m1)
```

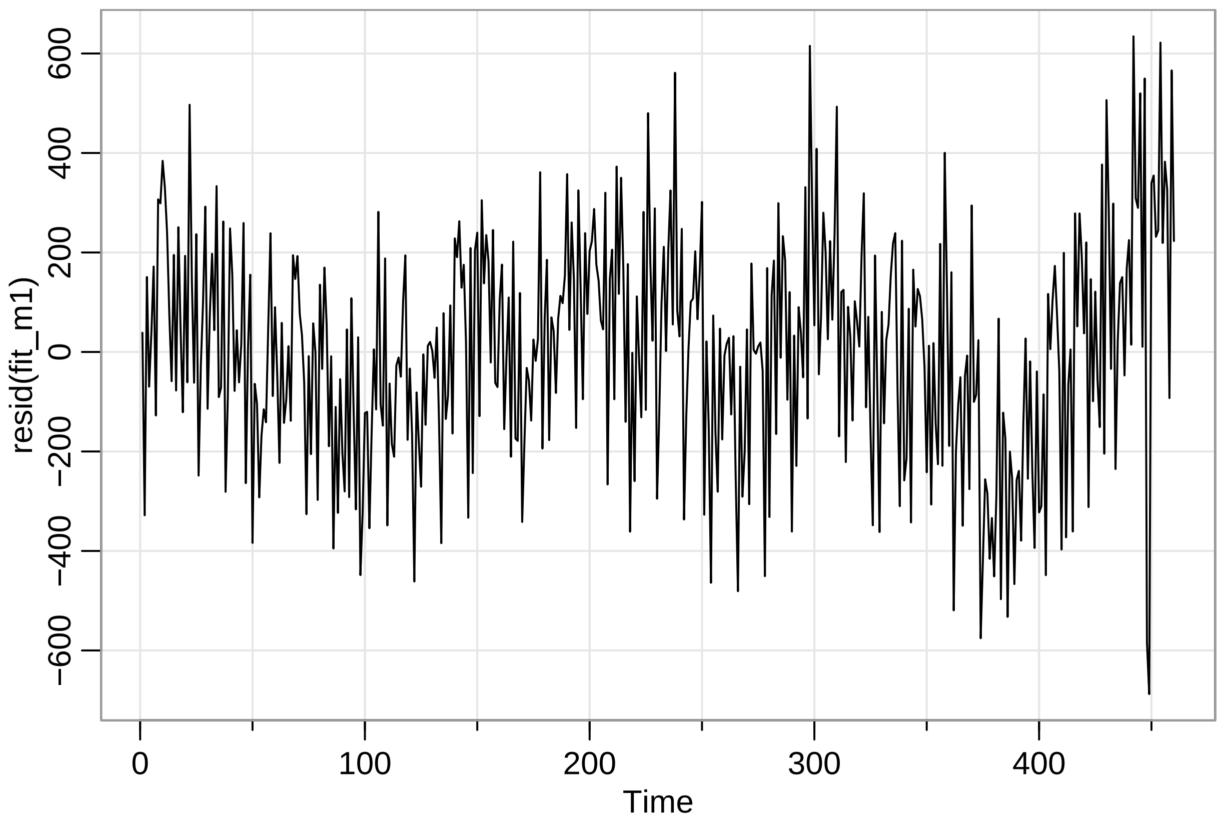



Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(meat ~ trend_m)
449
454
442
−3
−2
−1
0
1
2
3

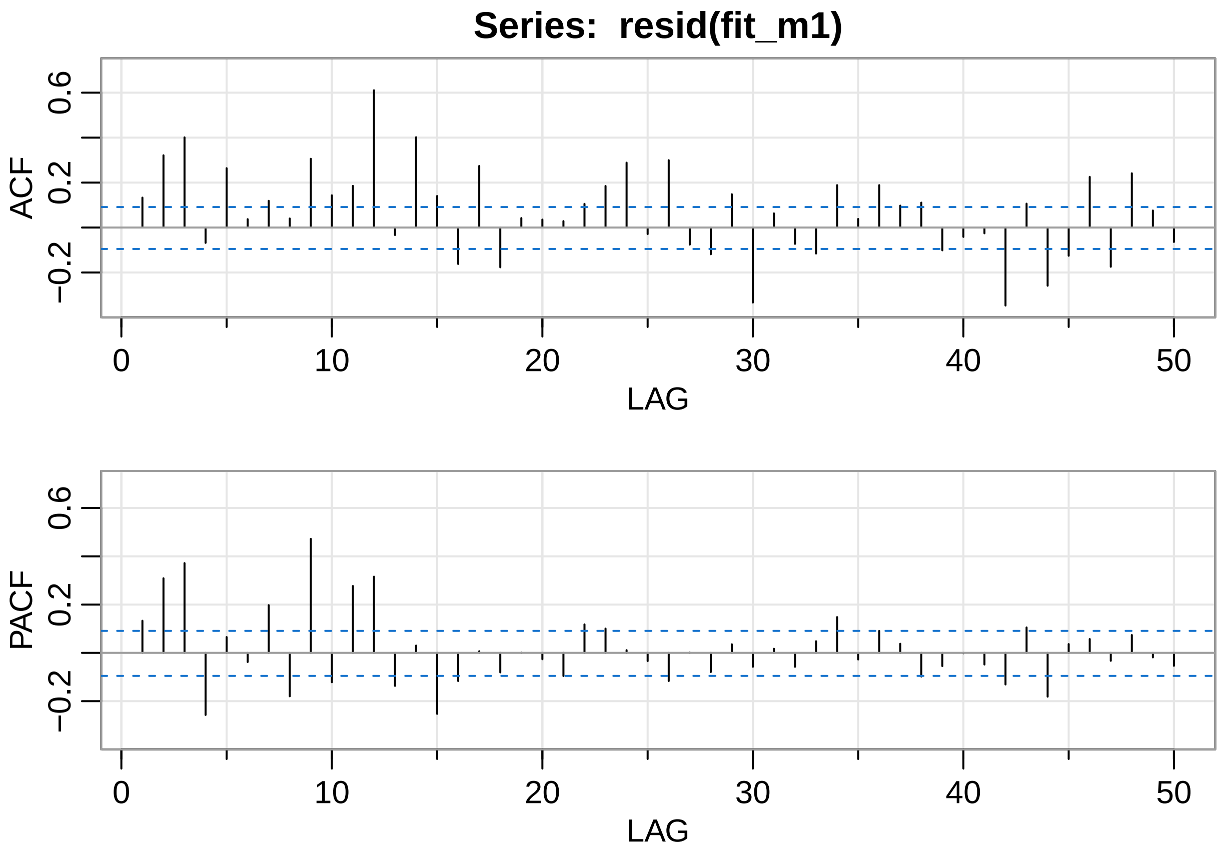

Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
3200
3400
3600
3800
4000
4200
4400
449
454
442
Fitted values
lm(meat ~ trend_m)




```
tsplot(resid(fit_m1))
```




```
acf2(resid(fit_m1), max.lag=50)
```




The residual plots for the M2 model show that the assumption of constant variance is valid. Tne Q-Q plot shows that the assumption of normality is valid. There are three outliers, but none have high leverage based on Cook's distance. Therefore, log transformation of the series may not be necessary. The time series plot suggests that the data alternate between upward and downward trends. Therefore, differencing is indicated to detrend the data.

The time series plot suggests the possibility of a seasonal pattern in the data, and therefore autocorrelated errors. The correlograms confirm the presence of significant autocorrelation (within a confidence band of two standard errors).

**Seasonal:** It appears that at the seasons (s = 12) the ACF is tailing off at lags 1s, 2s, 3s, 4s. This slow decay indicates seasonal differencing. Typically, differencing of order one is sufficient to obtain seasonal stationarity. The PACF appears to cut off after lag 1s. These results imply an SAR(1), $P = 1$, $D = 1$, $Q = 0$, in the seasonal component.

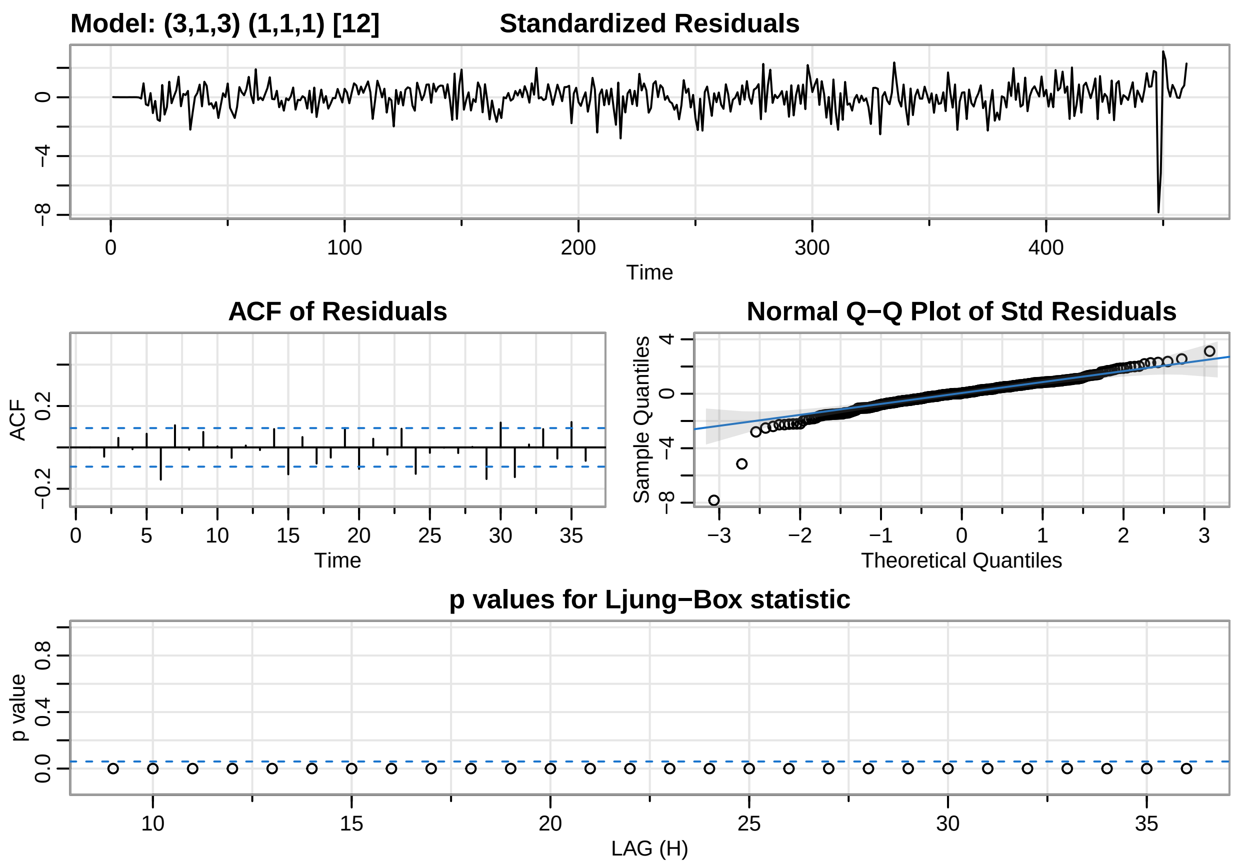
**Non-Seasonal:** Inspecting the sample ACF and PACF at the first few lags, it appears as though the ACF tails off, whereas the PACF cuts off at lag 3. This suggests an AR(3) within the seasons, $p = 3$ and $q = 0$.

Therefore, I would choose the following model:

ARIMA(3,1,0)x$(1,1,0)_{12}$

### Model Estimation: Total Meat Production

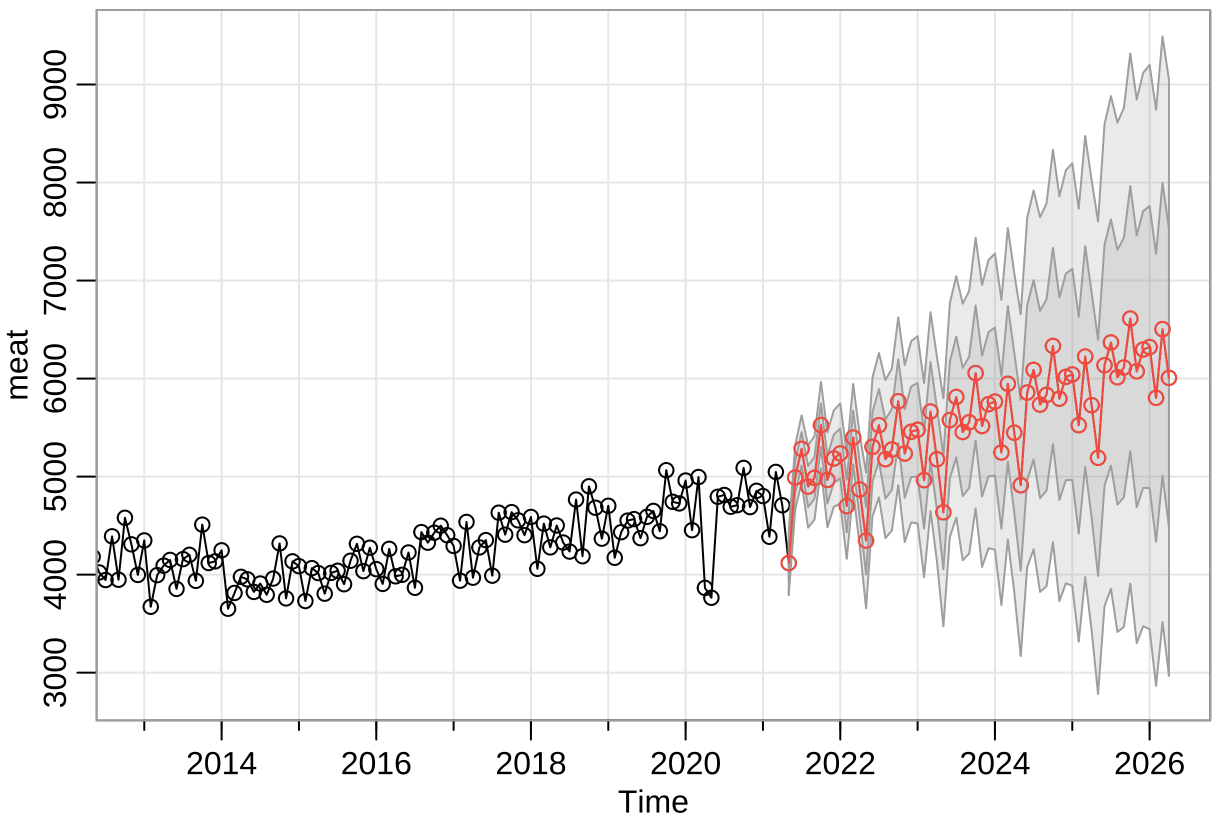
```
sarima(meat, 3,1,3, 1,1,1,12, xreg=cbind(trend_m+pop+pop2))
```




The plot of standardized residuals displays no obvious pattern. The ACF plot of residuals suggests that most of the autocorrelation is nonsignificant. the Q-Q plot suggests that the normality assumption is valid, despite the presence of four outliers which were previously not found to have problematic leverage. The q-statistic has *p*-values that are all significant, leading to rejection of the null hypothesis that the residuals are white.

**Forecasting: Total Meat Production**

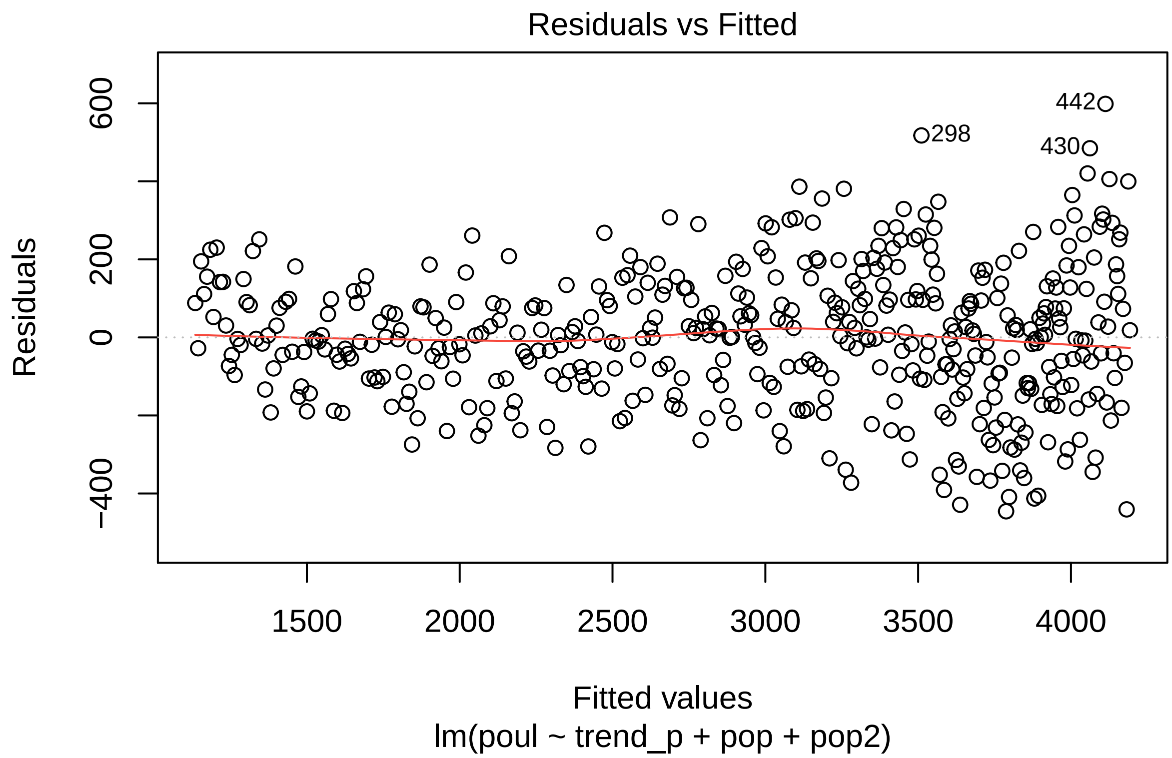
```
# End of data set
data[which(data$Year==2021 & data$Month==4),] # row 460
meat = ts(meat, start = c(1983,1), frequency = 12)
sarima.for(meat, 60, 3,1,0, 1,1,0,12)
abline(v=460)
```



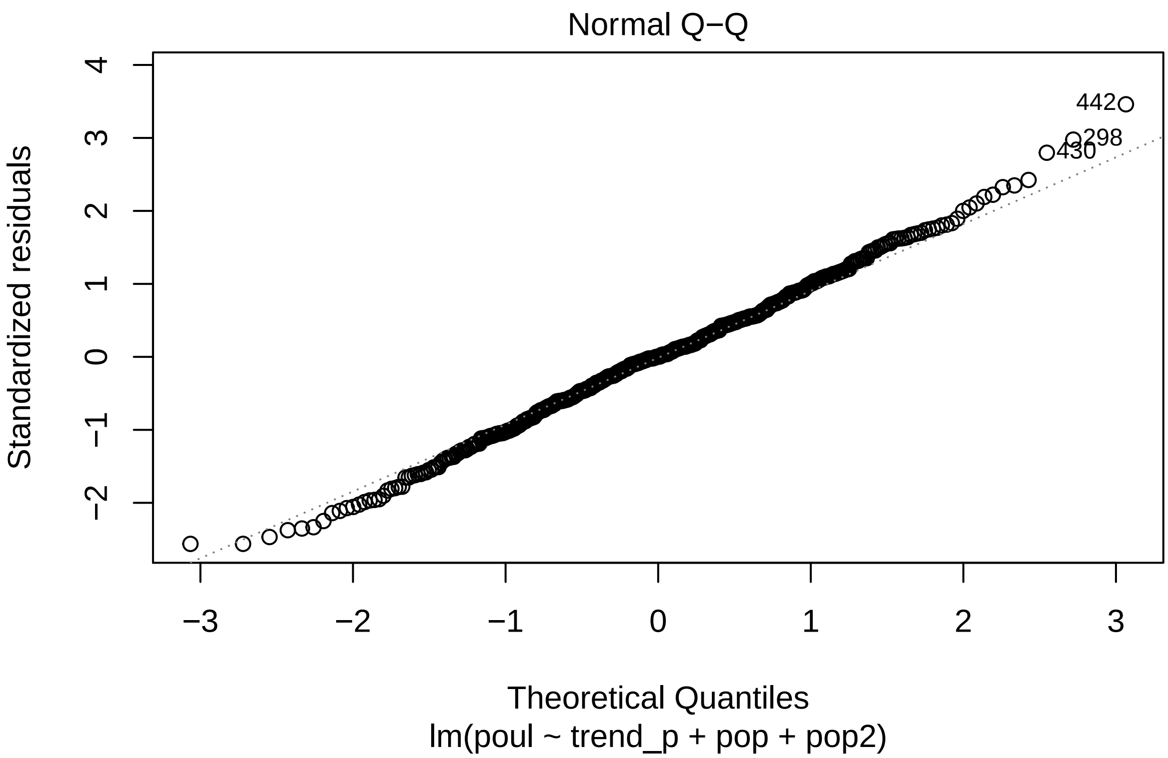

The forecasts out five years for total meat production is shown above.

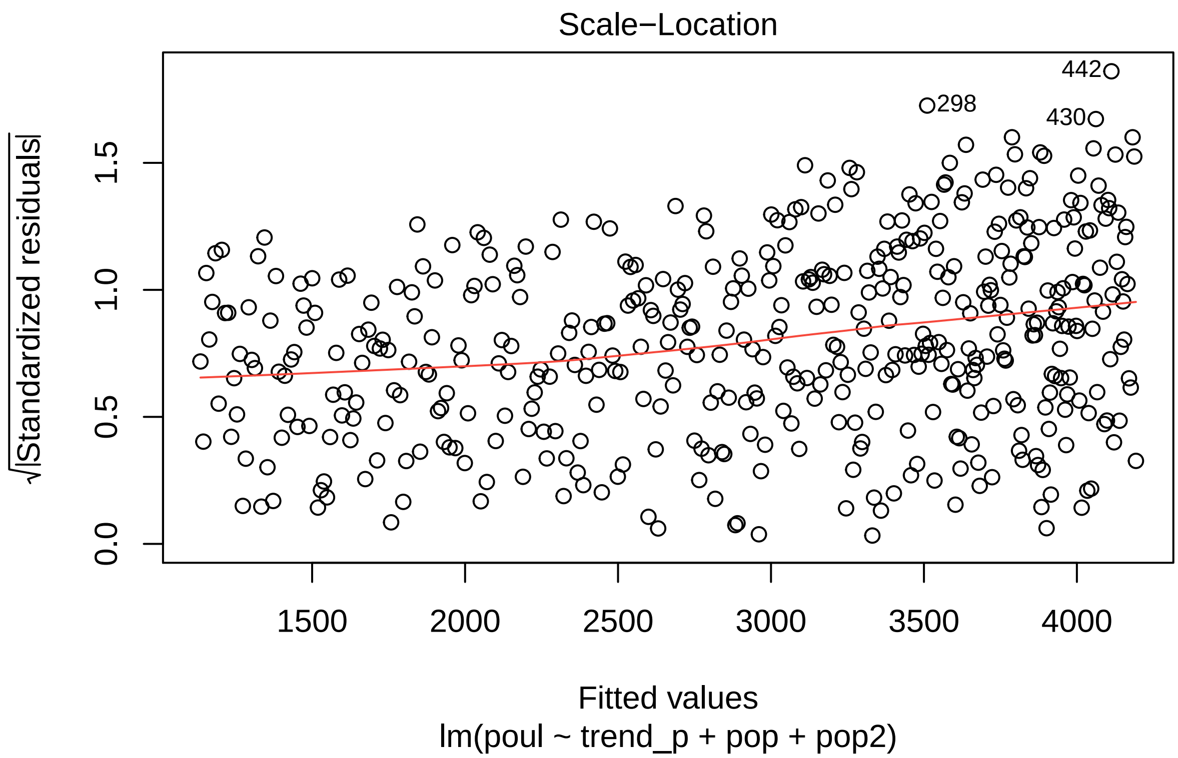
**Model Formulation: Total Poultry Production**

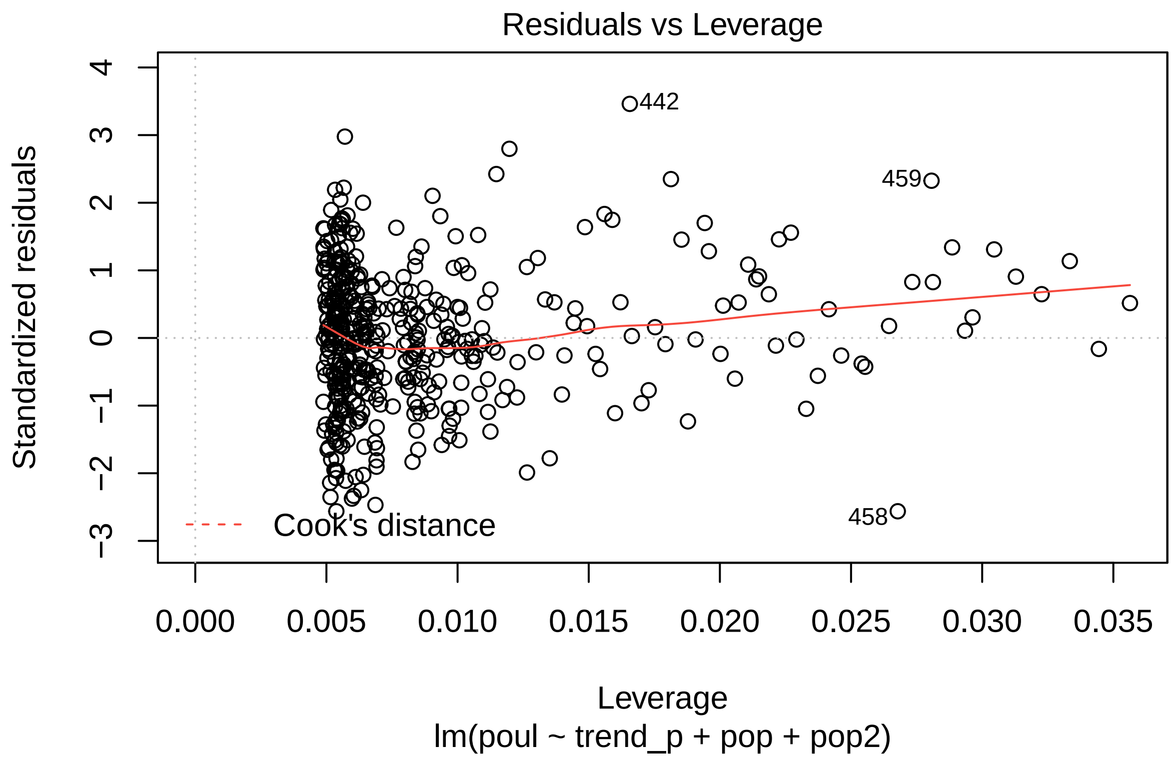
```
plot(fit_p3)
```



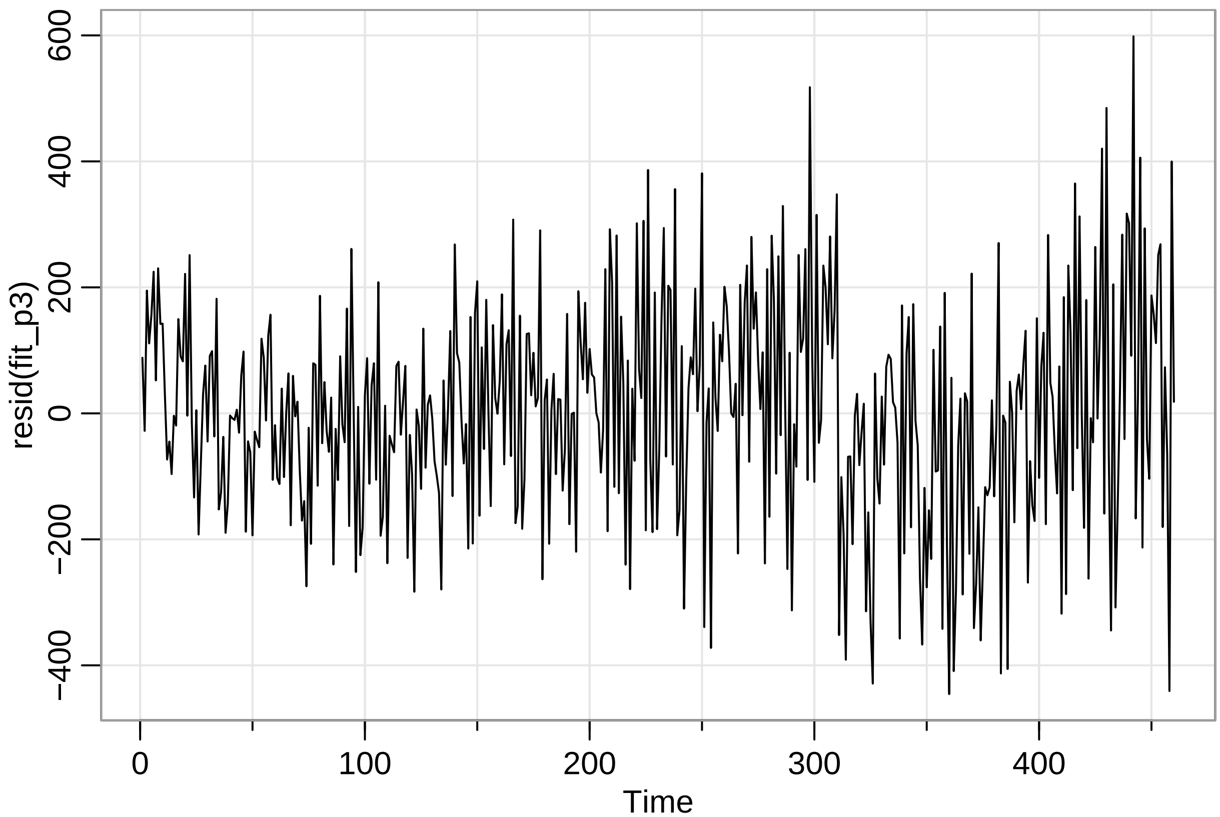
Residuals vs Fitted
Residuals
Fitted values
lm(poul ~ trend_p + pop + pop2)
600
200
0
−400
1500
2000
2500
3000
3500
4000
298
442
430


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(poul ~ trend_p + pop + pop2)
442
298
430
4
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
3

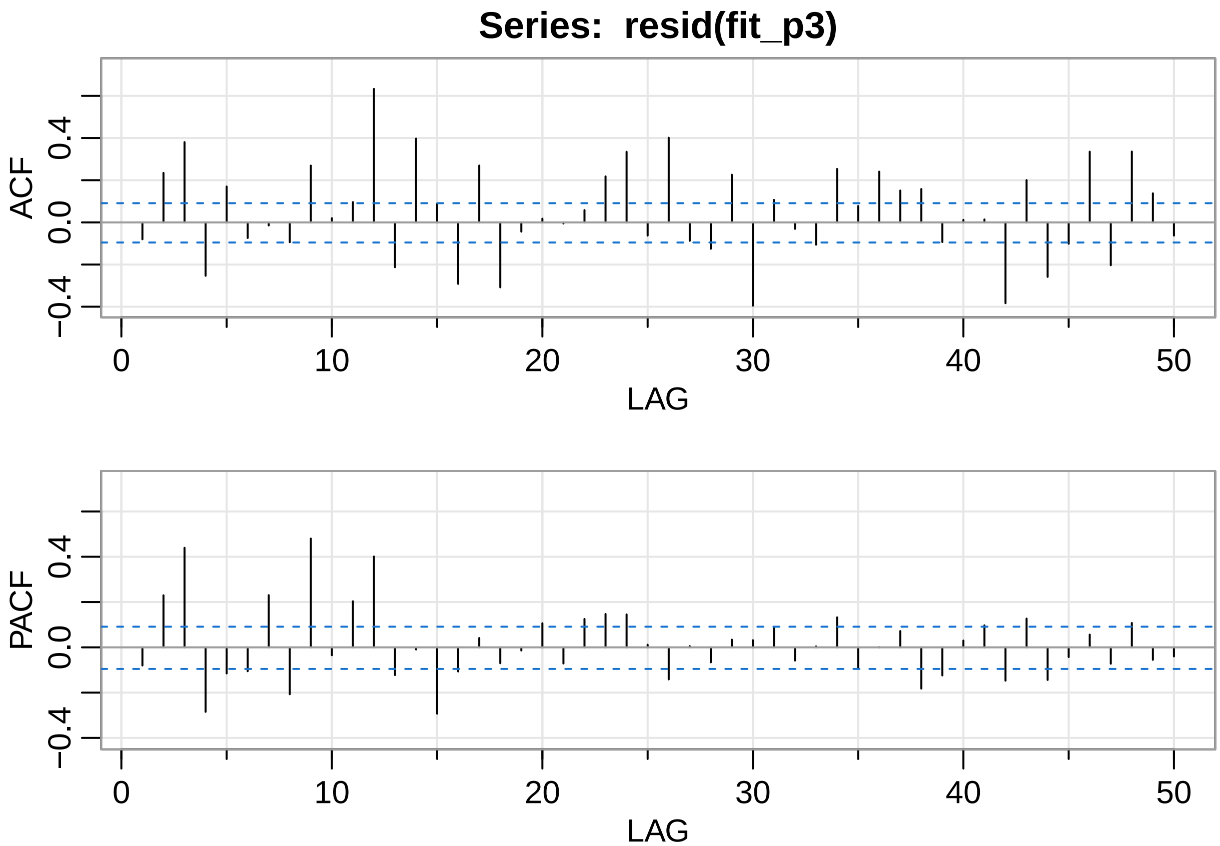

Scale−Location
442
298
430
√|Standardized residuals|
0.0
0.5
1.0
1.5
1500
2000
2500
3000
3500
4000
Fitted values
lm(poul ~ trend_p + pop + pop2)




```
tsplot(resid(fit_p3))
```



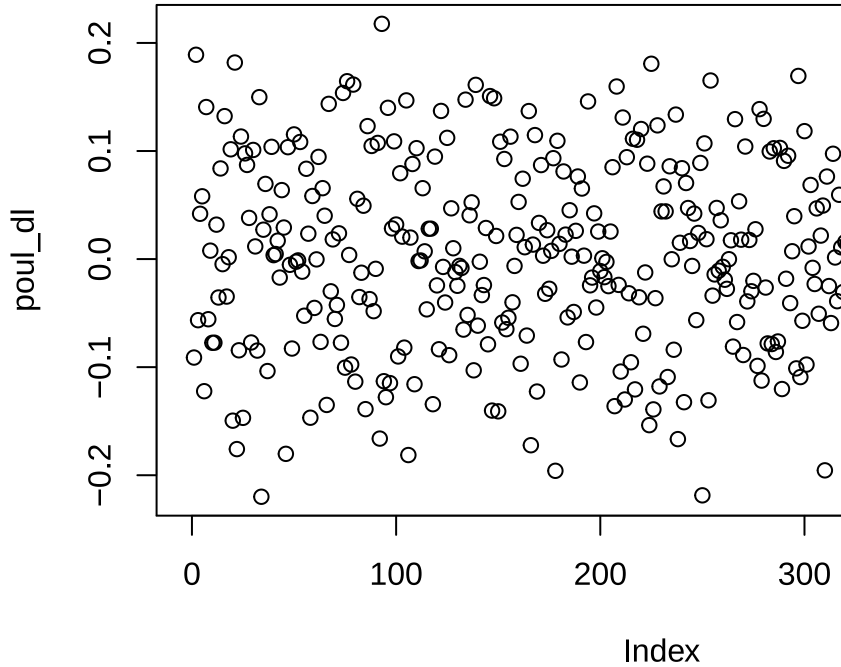

```
acf2(resid(fit_p3), max.lag=50)
```




The residual plots for the P3 model suggest that the variance is nonconstant. The Q-Q plot shows that the assumption of normality is valid. There is a high-leverage outlier at month 459, corresponding to an unusual rise and fall in production for February and March 2021. The time series plot suggests a shifting trend, and therefore a lack of stationarity. Therefore, the data can be logged to stabilize the variance, then differenced to remove the trend.

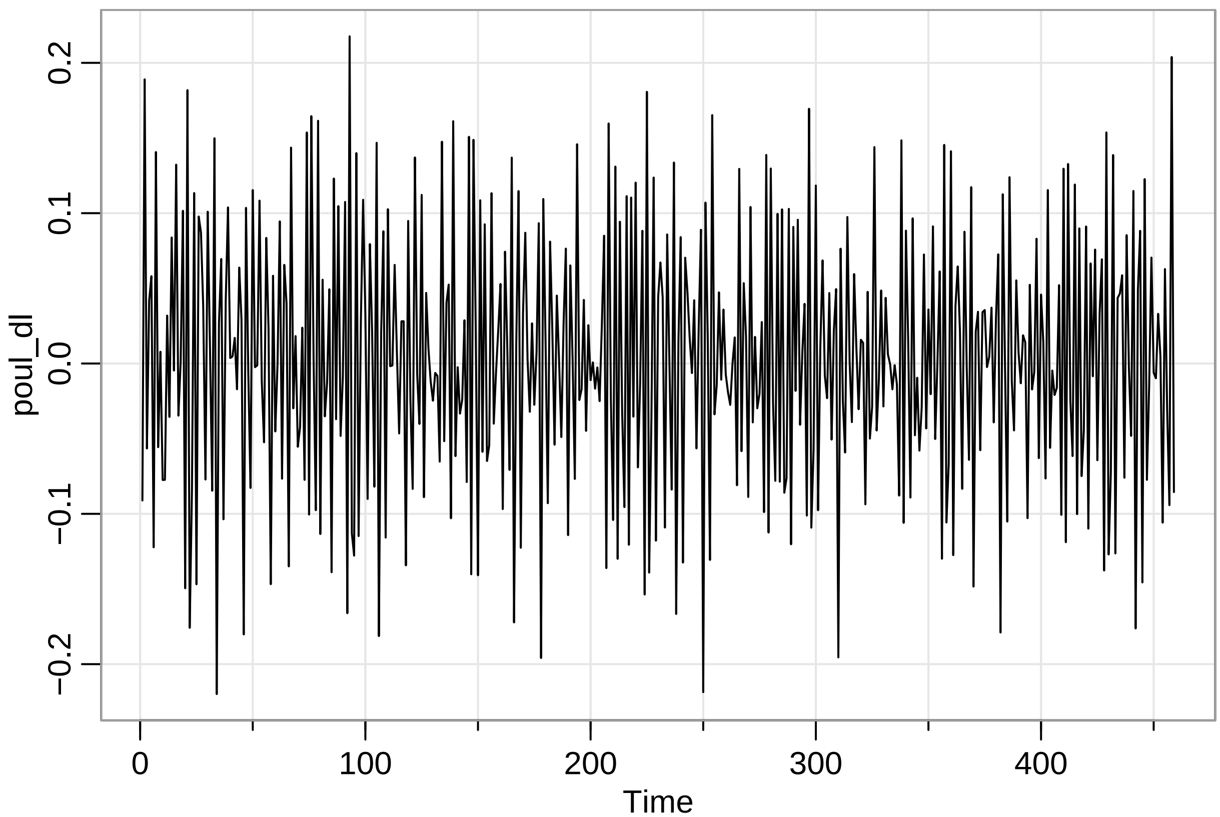
The time series plot suggests the possibility of a seasonal pattern in the data, and therefore autocorrelated errors. The correlograms confirm the presence of significant autocorrelation (within a confidence band of two standard errors).

```
poul_dl = diff(log(poul)) # apply logarithm, then difference
plot(poul_dl)
```

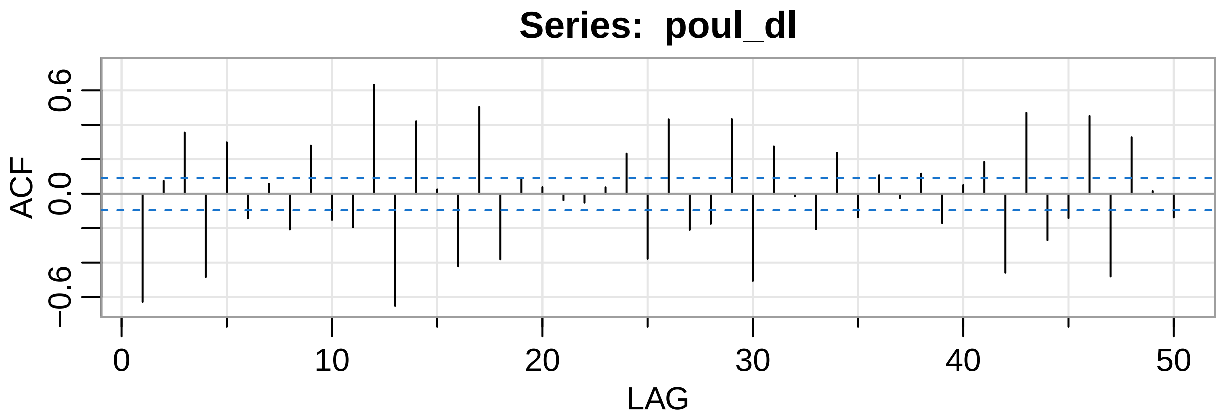


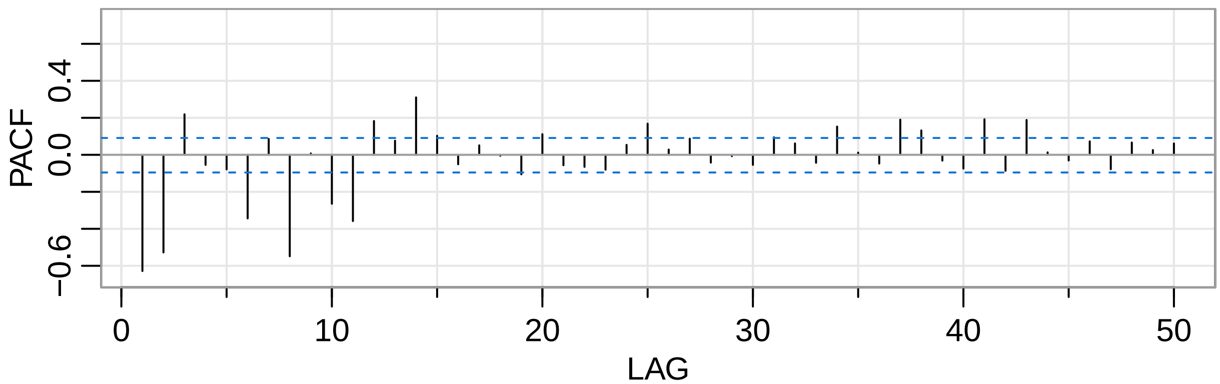

**Transformations: Total Poultry Production**

```
tsplot(poul_dl)
```




```
acf2(poul_dl, max.lag=50)
```




**Seasonal:** It appears that at the seasons (s = 12) the ACF is tailing off at lags approximately near 1s, 2s, 3s, 4s. This slow decay indicates seasonal differencing. Typically, differencing of order one is sufficient to obtain seasonal stationarity. The PACF appears to cut off after about lag 1s. These results imply an SAR(1), $P = 1$, $D = 1$, $Q = 0$, in the seasonal component.

**Non-Seasonal:** Inspecting the sample ACF and PACF at the first few lags, it appears as though the ACF tails off, whereas the PACF cuts off at lag 2. This suggests and ARMA(2,0) within the seasons, $p = 2$ and $q = 0$.

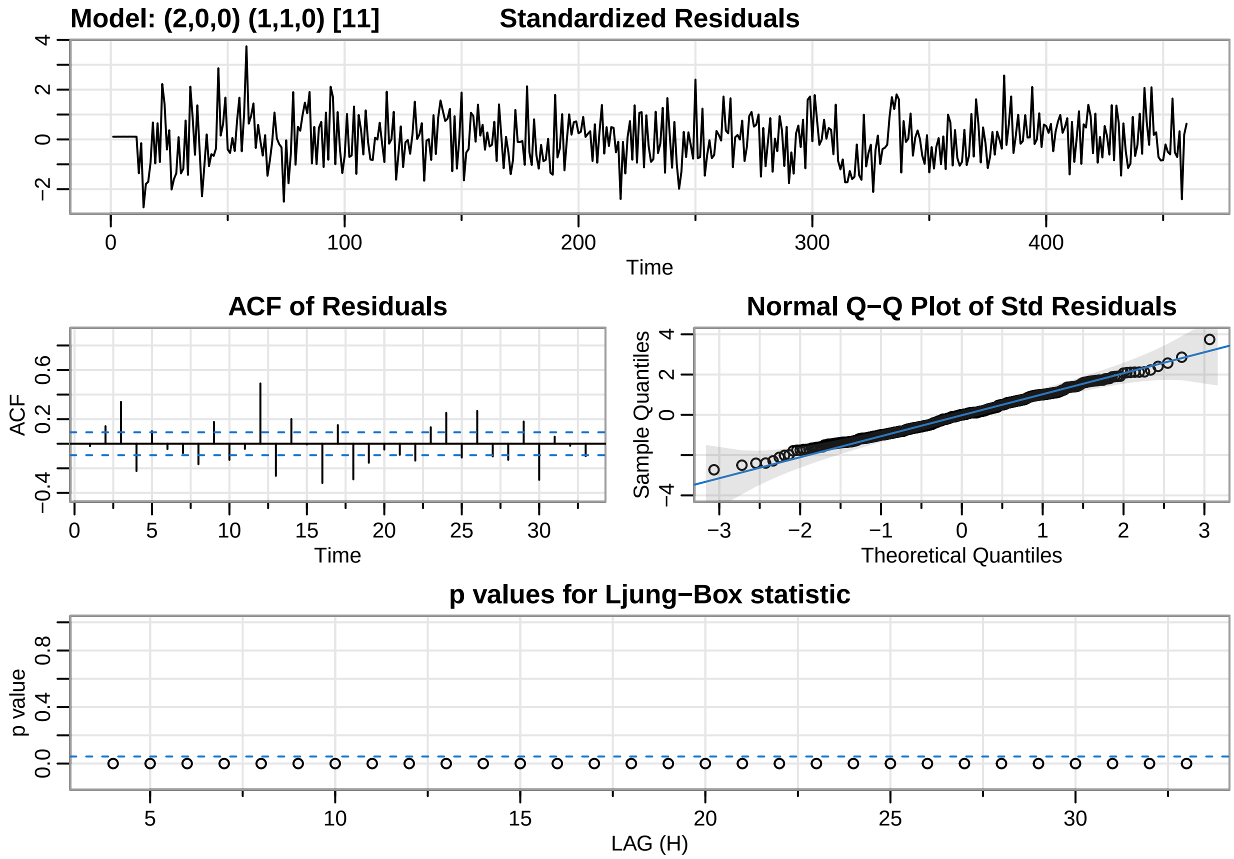
Therefore, I would choose the following model:

ARIMA(2,1,0)x$(1,1,0)_{12}$

**Model Estimation: Total Poultry Production**

```
sarima(log(poul), p=2,d=0,q=0, P=1,D=1,Q=0,S=11, xreg=cbind(trend_p, pop, pop2))
```

```
## Warning in sqrt(diag(fitit$var.coef)): NaNs produced

## Warning in sqrt(diag(fitit$var.coef)): NaNs produced
```

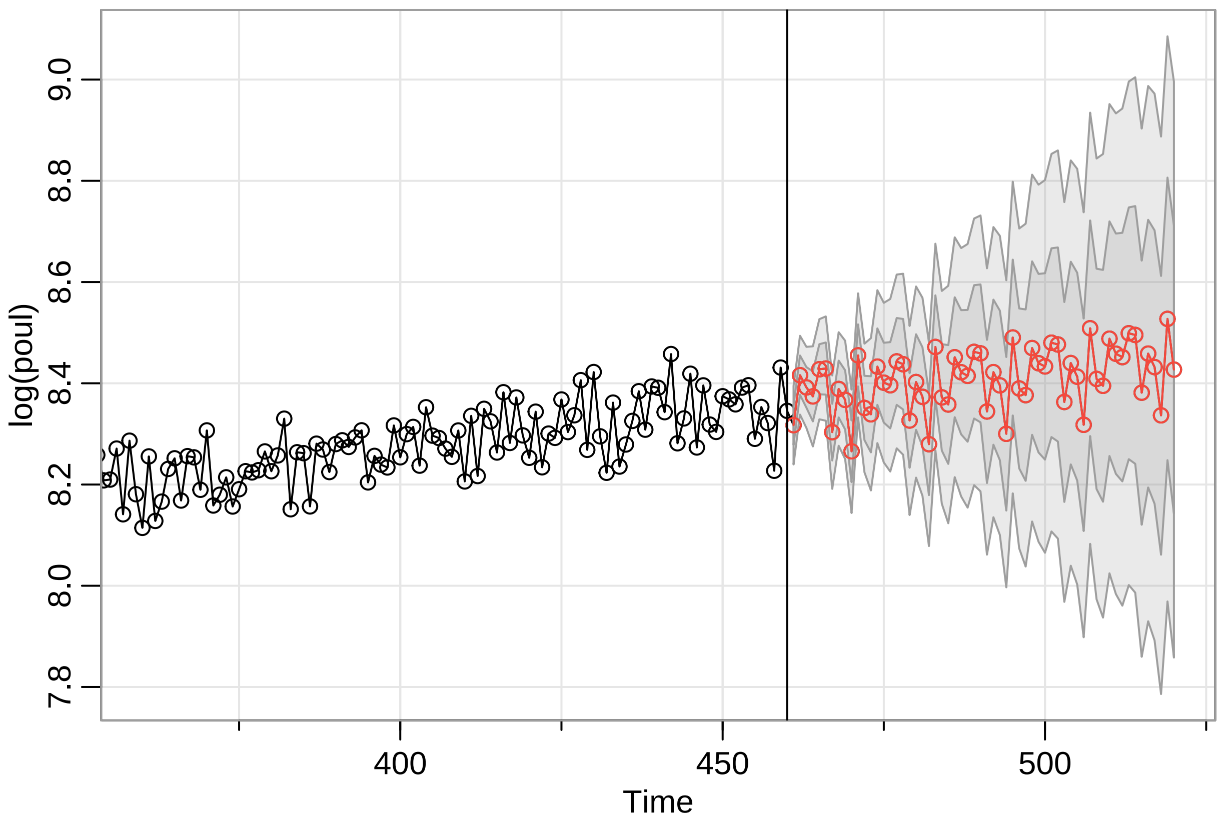



```
## Warning in sqrt(diag(x$var.coef)): NaNs produced
```

The plot of standardized residuals displays no obvious pattern. The ACF plot of residuals shows the presence of significant autocorrelation. the Q-Q plot suggests that the normality assumption is valid. The q-statistic has *p*-values that are all significant, leading to rejection of the null hypothesis that the residuals are white.

#### Forecasting: Total Poultry Production

```
sarima.for(log(poul), 60, 3,1,0, 1,1,0,12)
abline(v=460)
```




The forecasts out five years for the log of total poultry production is shown above.