Time Series Shark Attacks

Josh Balson

7/28/2021

Data Sets

https://www.sharks.org/global-shark-attack-file
Filtered for USA only
Filtered to include the time frame Jan 1972 - Dec 2020

```
sharks <- read.csv("sharks.csv", header = TRUE)
head(sharks)
```

```
##   year month attacks
## 1 1972   Jan       0
## 2 1972   Feb       0
## 3 1972   Mar       1
## 4 1972   Apr       0
## 5 1972   May       1
## 6 1972   Jun       1
```

```
ice.cream <- udf_make_icecream('IceCreamIndex.csv')
head(ice.cream)
```

```
##   year month     icpi
## 1 1972   Jan  59.9622
## 2 1972   Feb  67.0605
## 3 1972   Mar  74.2350
## 4 1972   Apr  78.1120
## 5 1972   May  84.7636
## 6 1972   Jun 100.5960
```

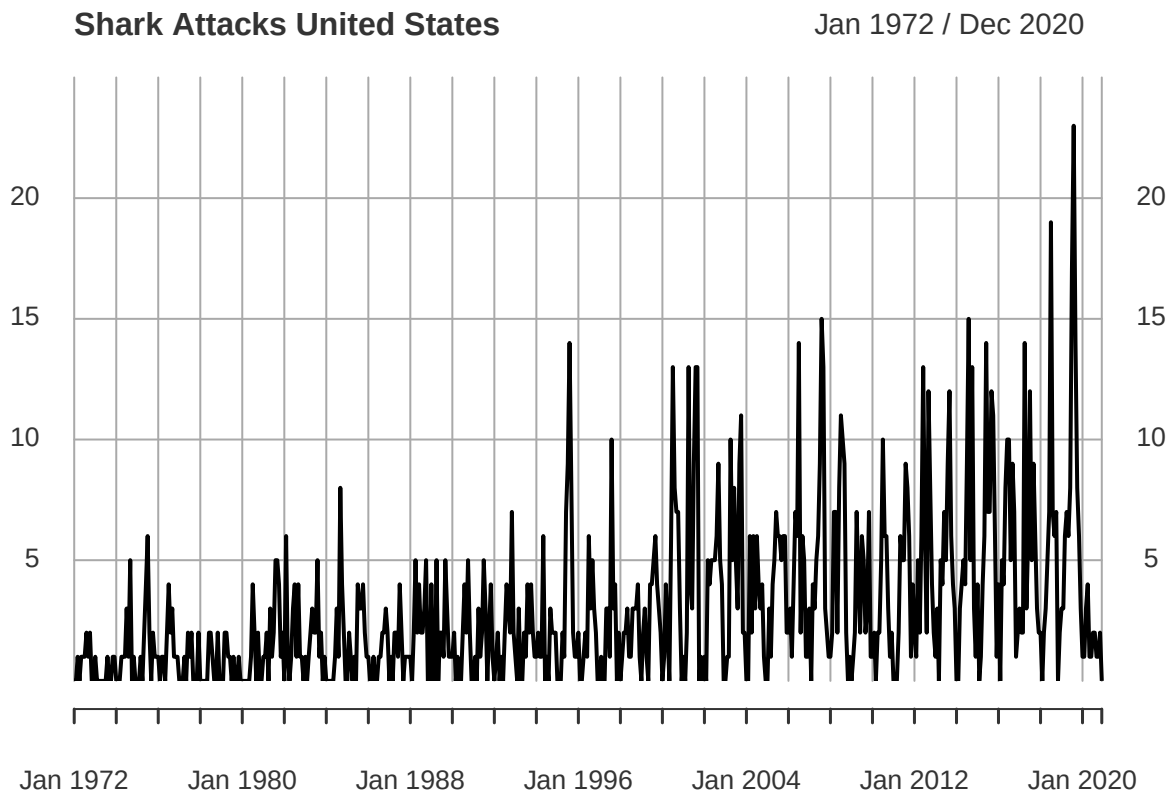
Data Preprocessing

```
data <- udf_make_data(sharks, ice.cream)

#ts1
ts.sharks <- data[[1]]
ts.sharks.train <- data[[2]]
ts.sharks.test <- data[[3]]

#ts2
```

```
ts.ice.cream <- data[[4]]
ts.ice.cream.train <- data[[5]]
ts.ice.cream.test <- data[[6]]
```

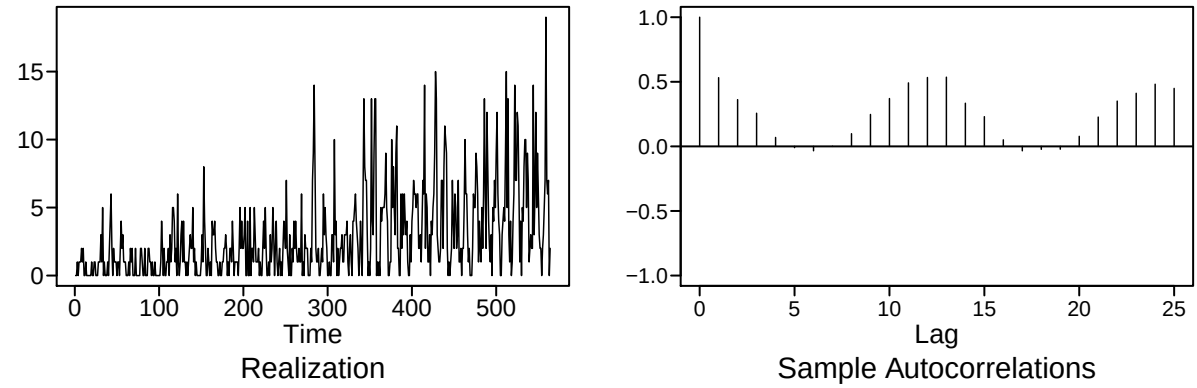
Plot data

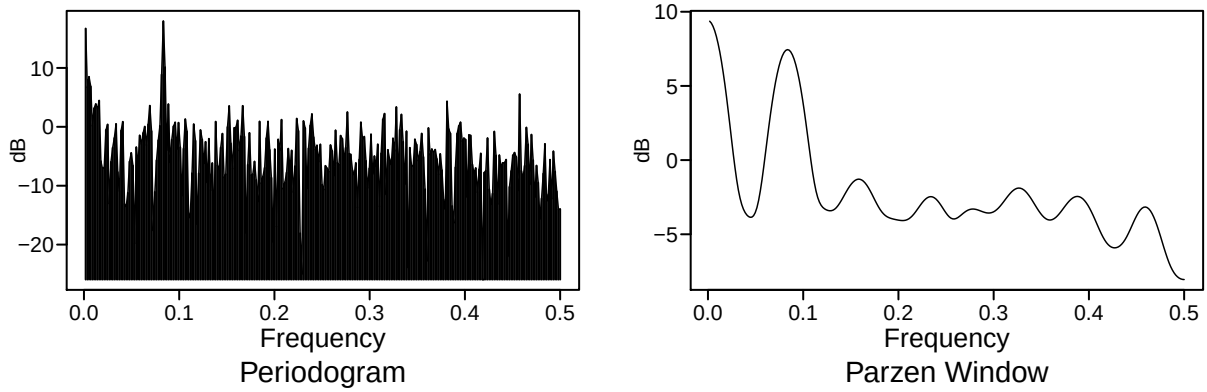
```
plot(as.xts(ts.sharks), major.format='%Y-%m',main='Shark Attacks United States',ylim=c(0,25))
```



ACF and Spectral Density

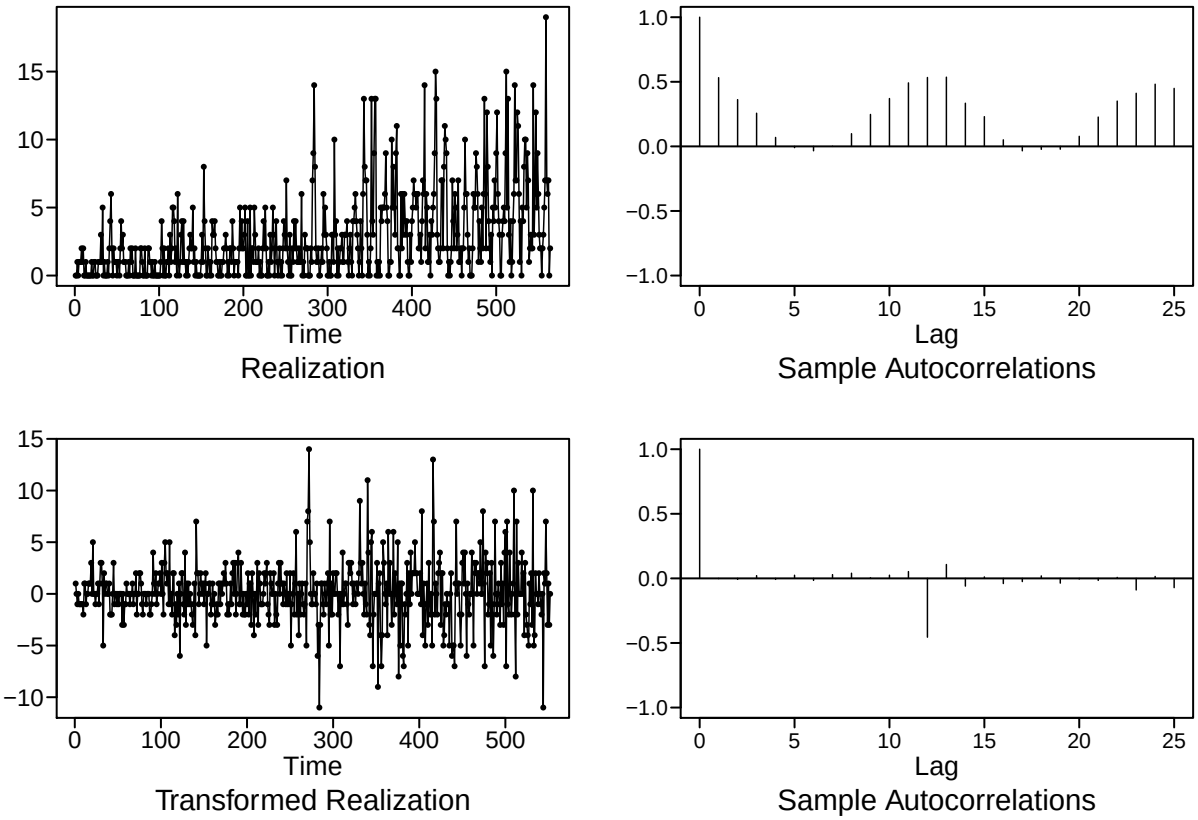
```
plotts.sample.wge(ts.sharks.train)
```





Build Forecast Models

```
models <- udf_make_models(ts.sharks.train, ts.sharks.test, ts.ice.cream, ts.ice.cream.train)
```



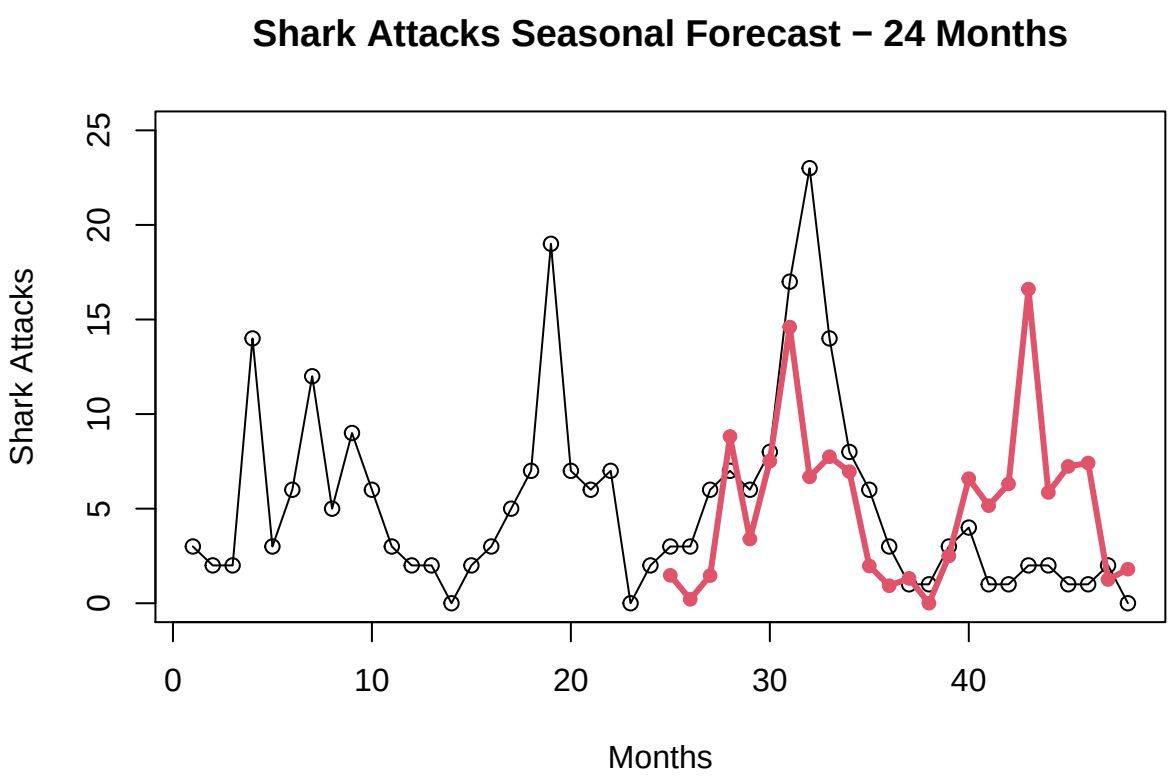
Seasonal Model
ARIMA(15,3) with s = 12

$$(1 - 0.3645B + 0.3245B^2 + 0.5476B^3 - 0.0081B^4 - 0.0304B^5 + 0.0216B^6...$$
$$...-0.06B^7-0.0301B^8-0.0137B^9-0.0364B^{10}-0.0781B^{11}+0.4743B^{12}-0.318B^{13}+0.1924B^{14}+0.1346B^{15})X_t = ...$$
$$...(1 - 0.2669B - 0.2659238B^2 + 0.7336743B^3)a_t$$

$$\sigma^2 = 6.0746$$

Plot Seasonal 24 Month Forecast

```
plot(ts.sharks[541:588],main='Shark Attacks Seasonal Forecast - 24 Months'
     ,ylim=c(0,25), ylab='Shark Attacks',type='o', xlab='Months')
points(models[[3]][541:588],type='o',lty=1,lwd=3,col=2,cex=.6)
```



Get 24 Month ASE for Seasonal

```
ASE_seas <- models[[2]]
ASE_seas
```

```
## [1] 30.8087
```

Get 12 Month ASE for Seasonal

```
ASE_seas12 <- models[[4]]
ASE_seas12
```

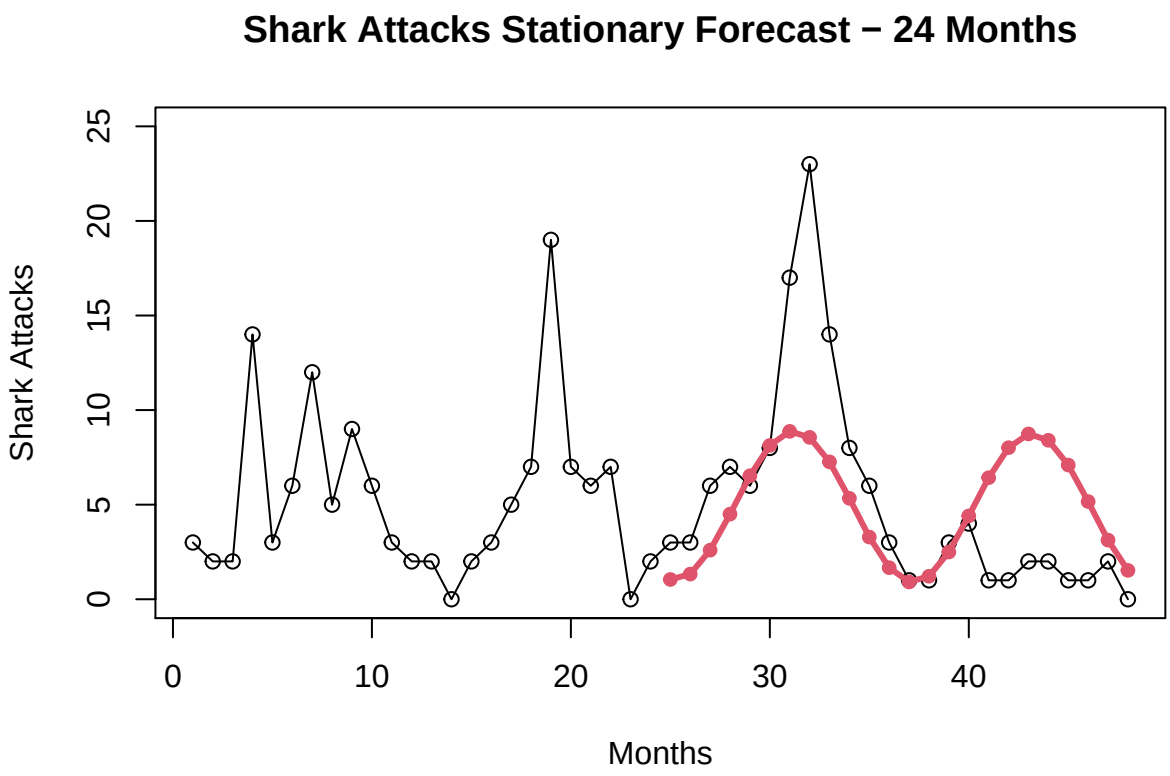
```
## [1] 31.14982
```

Stationary Model ARMA(3,3)

$$(1 - 2.7255B + 2.7211B^2 - 0.9942B^3)(X_t - 2.828) = (1 - 2.6349307B + 2.5456529B^2 - 0.8923811B^3)a_t$$

Plot Stationary 24 Month Forecast

```
plot(ts.sharks[541:588],main='Shark Attacks Stationary Forecast - 24 Months'
     ,ylim=c(0,25), ylab='Shark Attacks',type='o', xlab='Months')
points(models[[7]][541:588],type='o',lty=1,lwd=3,col=2,cex=.6)
```



Get 24 Month ASE for Stationary

```
ASE_stat <- models[[6]]
ASE_stat
```

```
## [1] 24.35539
```

Get 12 Month ASE for Stationary

```
ASE_stat12 <- models[[8]]
ASE_stat12
```

```
## [1] 30.07936
```

Signal Plus Noise

```
# test signal significance
summary(models[[9]])
```

```
##
## Call:
## lm(formula = x ~ t, data = df_tmp)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -5.4443 -1.7677 -0.3936  1.1359 13.5930
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.1930892  0.2337360   0.826    0.409
## t           0.0093272  0.0007169  13.011   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.772 on 562 degrees of freedom
## Multiple R-squared:  0.2315, Adjusted R-squared:  0.2301
## F-statistic: 169.3 on 1 and 562 DF,  p-value: < 2.2e-16
```

```
# cochrane-orcutt test
summary(models[[10]])
```

```
## Call:
## lm(formula = x ~ t, data = df_tmp)
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.2034656  0.3556290   0.572   0.5675
## t           0.0092788  0.0010879   8.529   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.5537 on 561 degrees of freedom
## Multiple R-squared:  0.1148 ,  Adjusted R-squared:  0.1132
## F-statistic: 72.7 on 1 and 561 DF,  p-value: < 1.373e-16
##
## Durbin-Watson statistic
## (original):    1.21686 , p-value: 4.054e-21
## (transformed): 2.01496 , p-value: 5.538e-01
```

Signal Plus Noise Model Linear with ARMA(7,4)

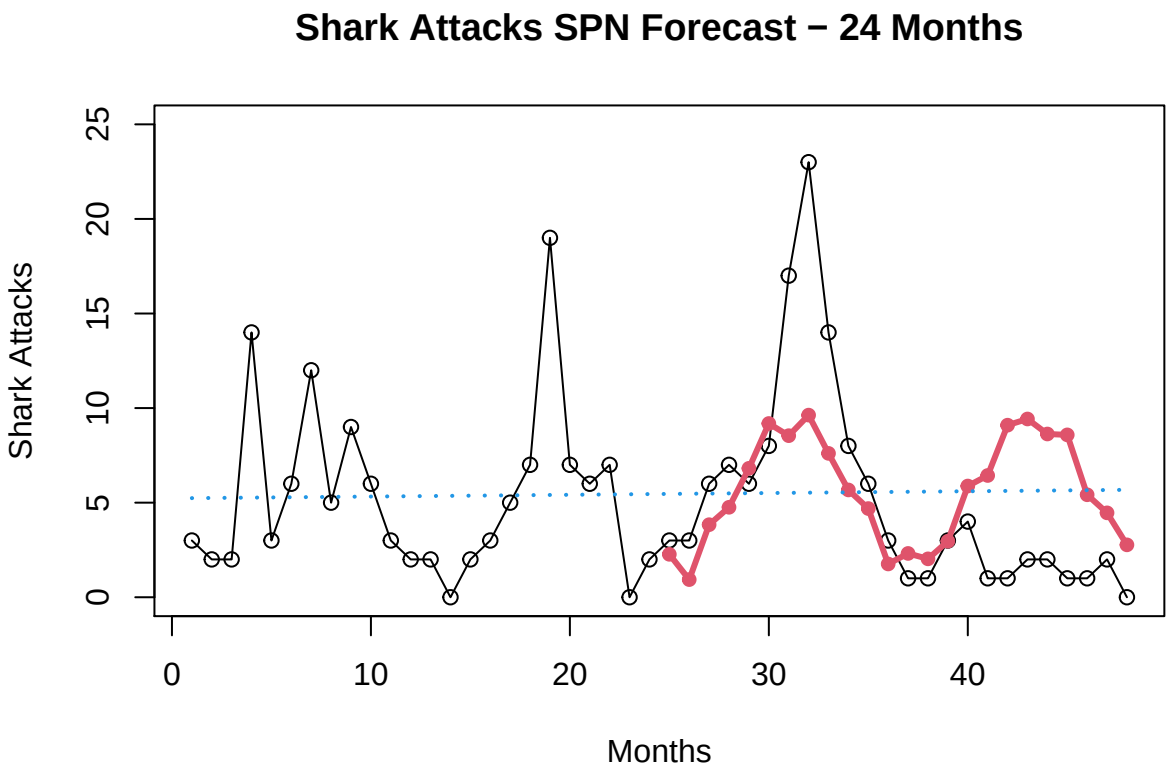
$$X_t = s_t + Z_t, where$$

$$s_t = 0.0093272 * t + 0.1930892, and$$

$$Z_t = (1 - 0.2826B^1 - 0.6936B^2 - 0.1279B^3 + 1.0027B^4 - 0.073B^5 + 0.0328B^6...$$

$$... - 0.0748B^7)(X_t - 2.828014) - (1 - 0.139B^1 - 0.7524B^2 - 0.1588B^3 + 0.9437B^4)a_t$$

```
plot(ts.sharks[541:588],main='Shark Attacks SPN Forecast - 24 Months'
     ,ylim=c(0,25), ylab='Shark Attacks',type='o', xlab='Months')
points(models[[14]][541:588],type='o',lty=1,lwd=3,col=2,cex=.6)
points(models[[12]][541:588],type='l',lty=3,lwd=2,col=4,cex=.6)
```



Get 24 Month ASE for SPN

```
ASE_spn <- models[[13]]
ASE_spn
```

```
## [1] 25.32576
```

Get 12 Month ASE for SPN

```
ASE_spn12 <- models[[15]]
ASE_spn12
```

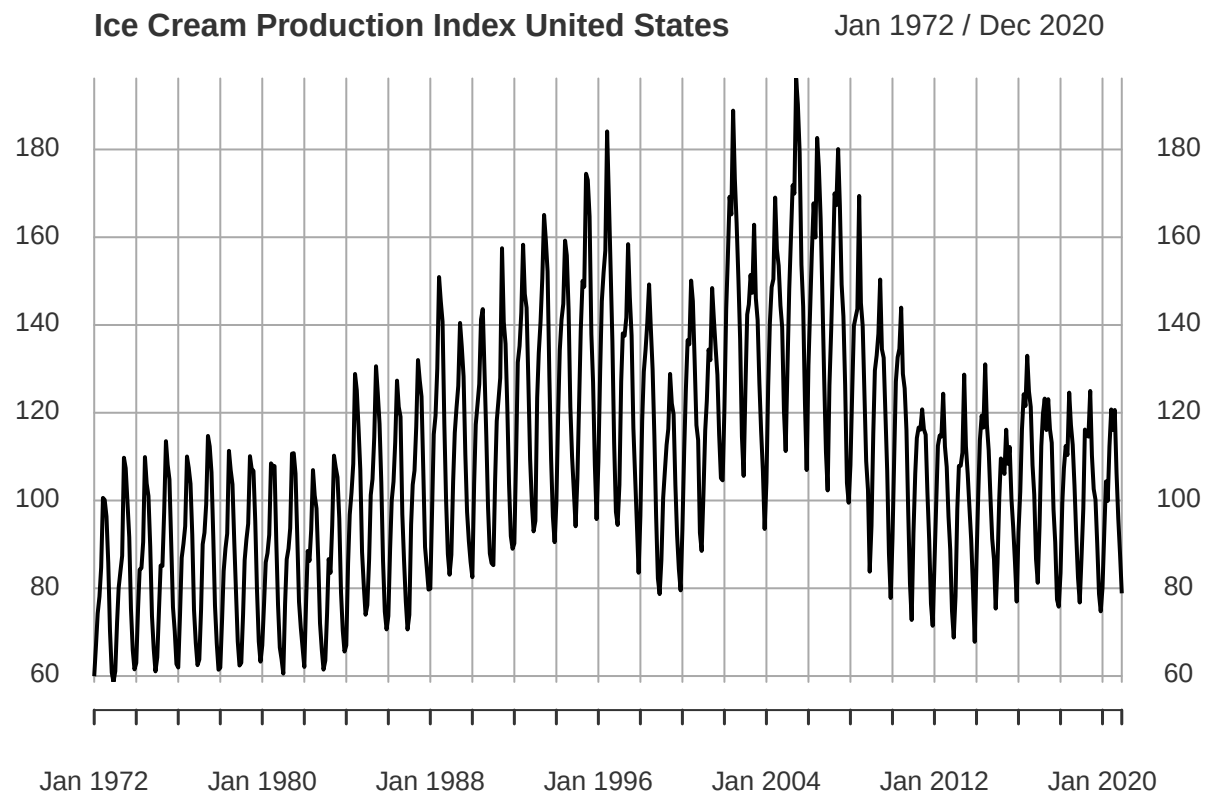
```
## [1] 26.37848
```

Bivariate Models

VAR
MLP
Neural Network

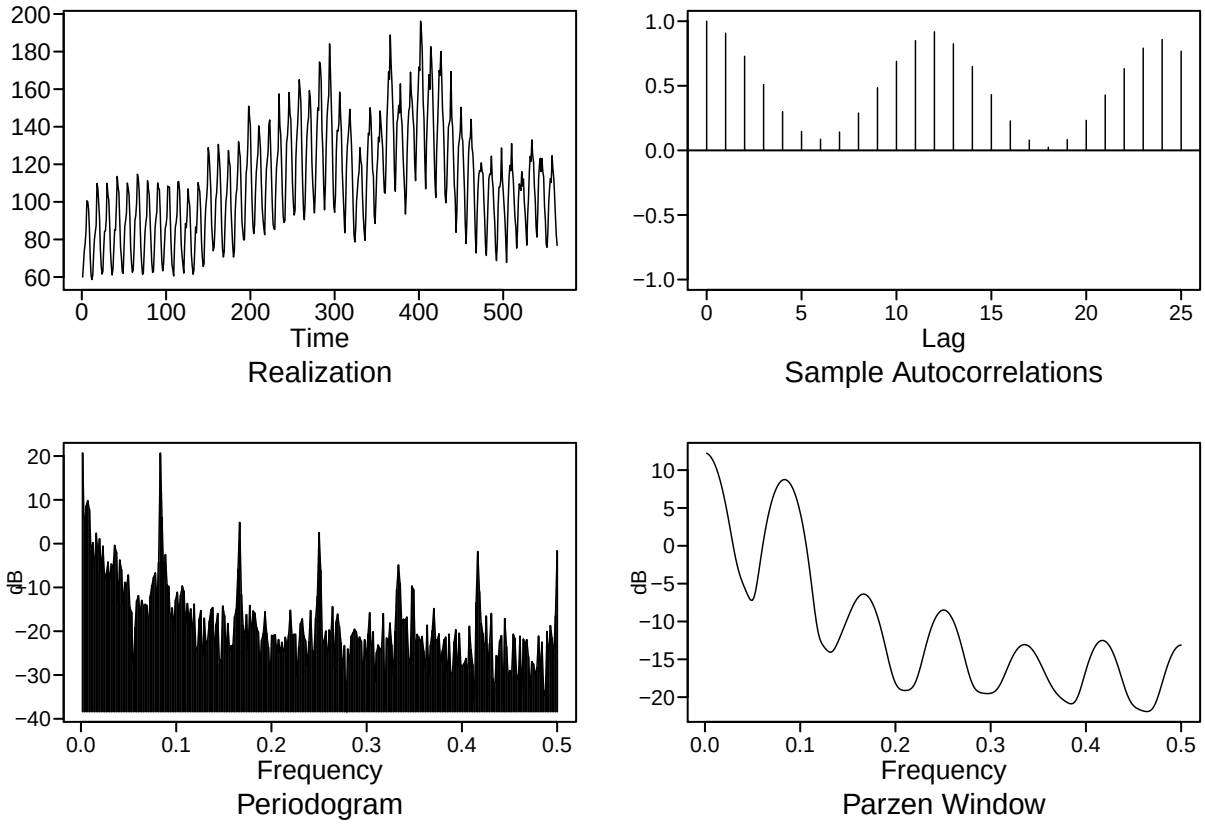
View New Data - Ice Cream Production Index
https://fred.stlouisfed.org/series/IPN31152N

```
plot(as.xts(ts.ice.cream), major.format='%Y-%m',main='Ice Cream Production Index United States')
```



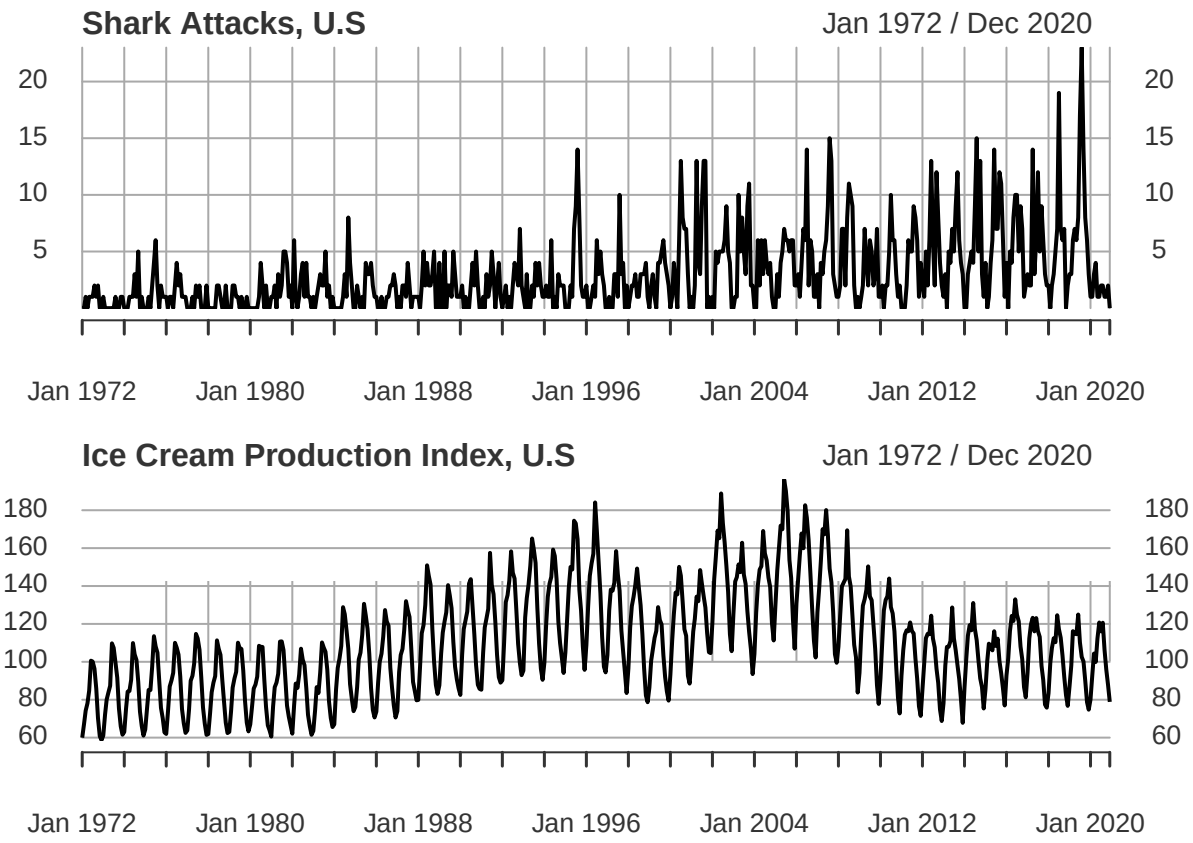
Look at ACF and Spectral Density for Ice Cream Production

```
plotts.sample.wge(ts.ice.cream.train)
```



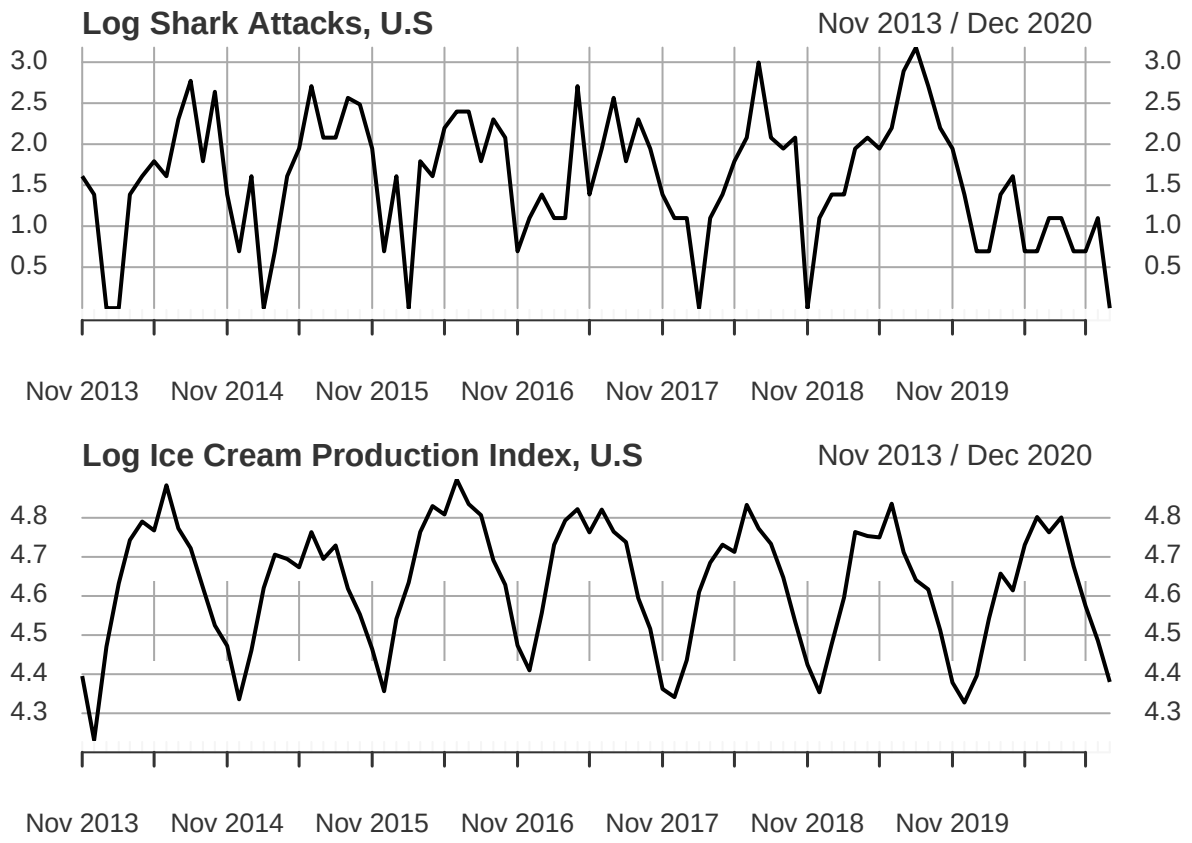
View Both Data Sets

```
par(mfrow=c(2,1))
plot(as.xts(ts.sharks), main='Shark Attacks, U.S')
plot(as.xts(ts.ice.cream), main='Ice Cream Production Index, U.S')
```



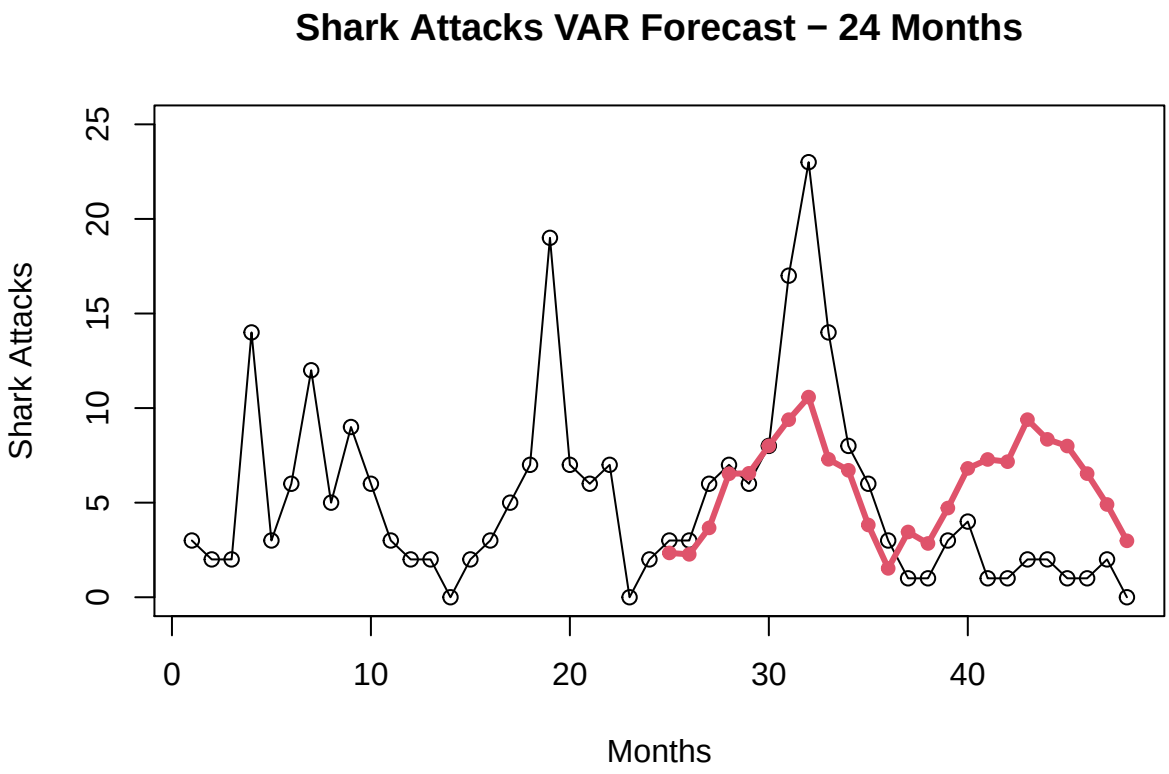
Zoom-in Past Seven Years - Log Transforms

```
par(mfrow=c(2,1))
plot(as.xts(log1p(ts.sharks))[503:588], main='Log Shark Attacks, U.S')
plot(as.xts(log1p(ts.ice.cream))[503:588], main='Log Ice Cream Production Index, U.S')
```

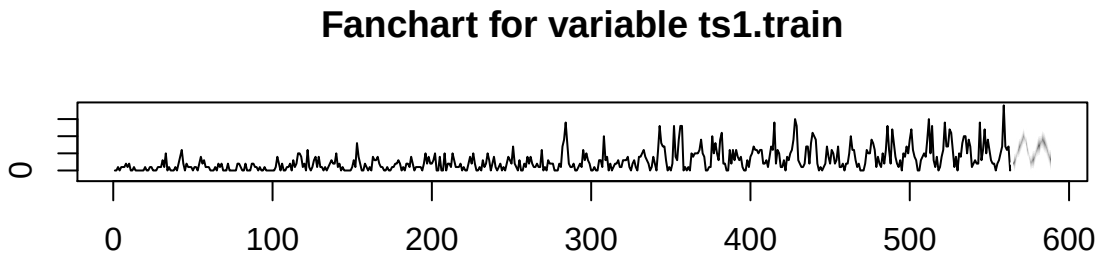


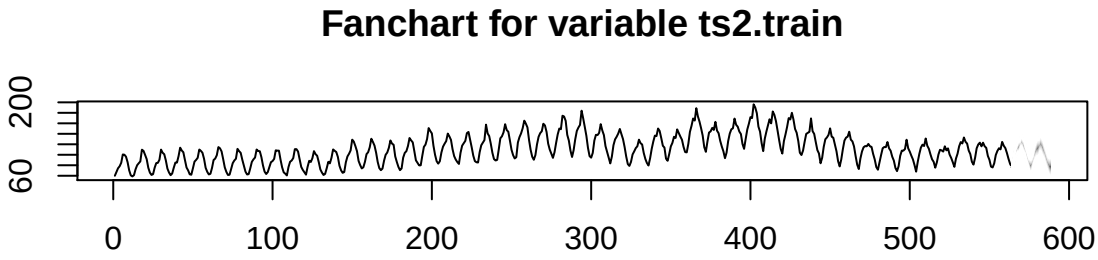
VAR model with Ice Cream Production

```
plot(ts.sharks[541:588],main='Shark Attacks VAR Forecast - 24 Months'
     ,ylim=c(0,25), ylab='Shark Attacks',type='o', xlab='Months')
points(models[[18]][541:588],type='o',lty=1,lwd=3,col=2,cex=.6)
```




```
fanchart(models[[16]])
```





Get 24 Month ASE for VAR

```
ASE_VAR <- models[[17]]
ASE_VAR
```

```
## [1] 23.42438
```

Get 12 Month ASE for SPN

```
ASE_VAR12 <- models[[19]]
ASE_VAR12
```

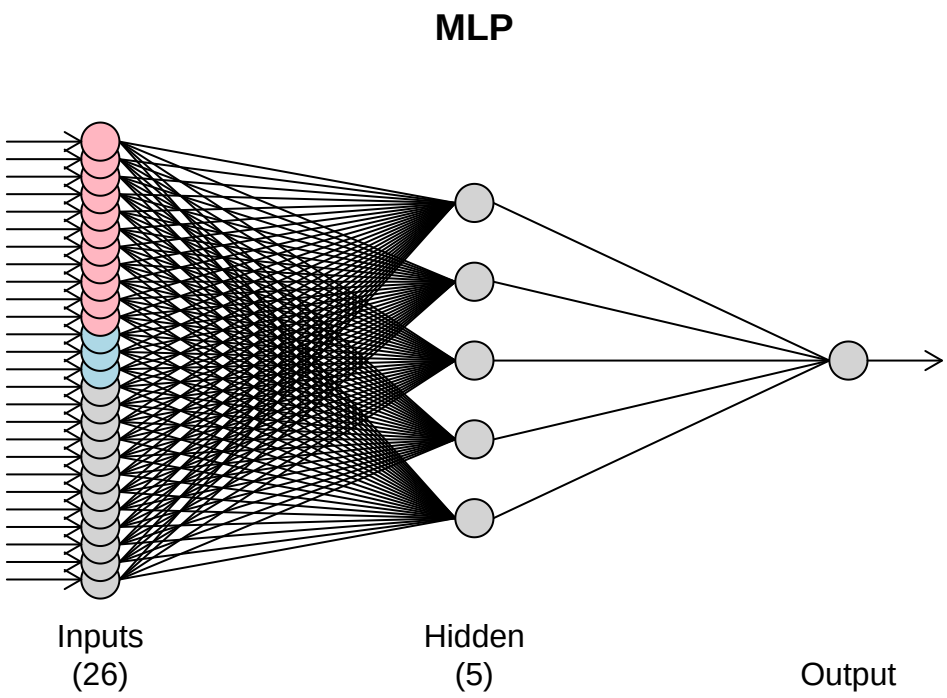
```
## [1] 22.72306
```

MLP model with Ice Cream Production

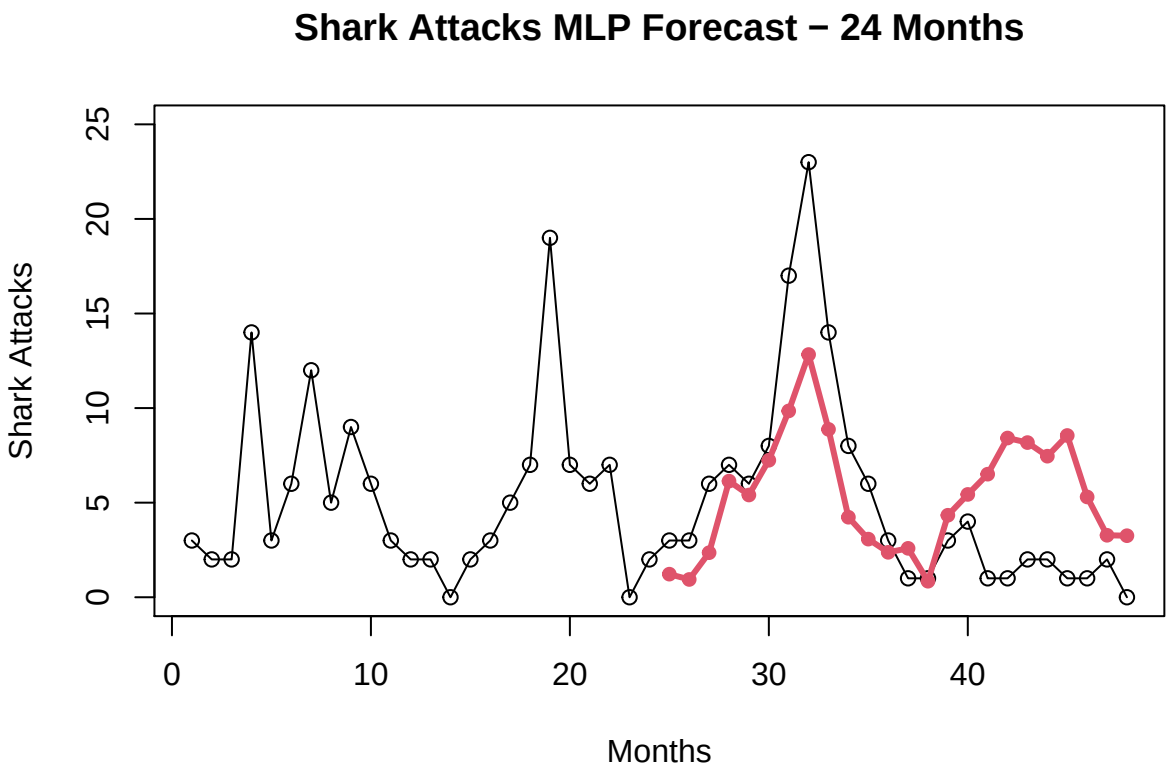
```
models[[20]]
```

```
## MLP fit with 5 hidden nodes and 20 repetitions.
## Series modelled in differences: D1.
## Univariate lags: (1,2,3,4,5,6,7,8,9,10,11,12)
## 1 regressor included.
## - Regressor 1 lags: (6,10,11)
## Deterministic seasonal dummies included.
## Forecast combined using the mean operator.
## MSE: 1.2709.
```

```
plot(models[[20]])
```




```
plot(ts.sharks[541:588],main='Shark Attacks MLP Forecast - 24 Months'
     ,ylim=c(0,25), ylab='Shark Attacks',type='o', xlab='Months')
points(models[[22]][541:588],type='o',lty=1,lwd=3,col=2,cex=.6)
```



Get 24 Month ASE for MLP

```
ASE_MLP <- models[[21]]
ASE_MLP
```

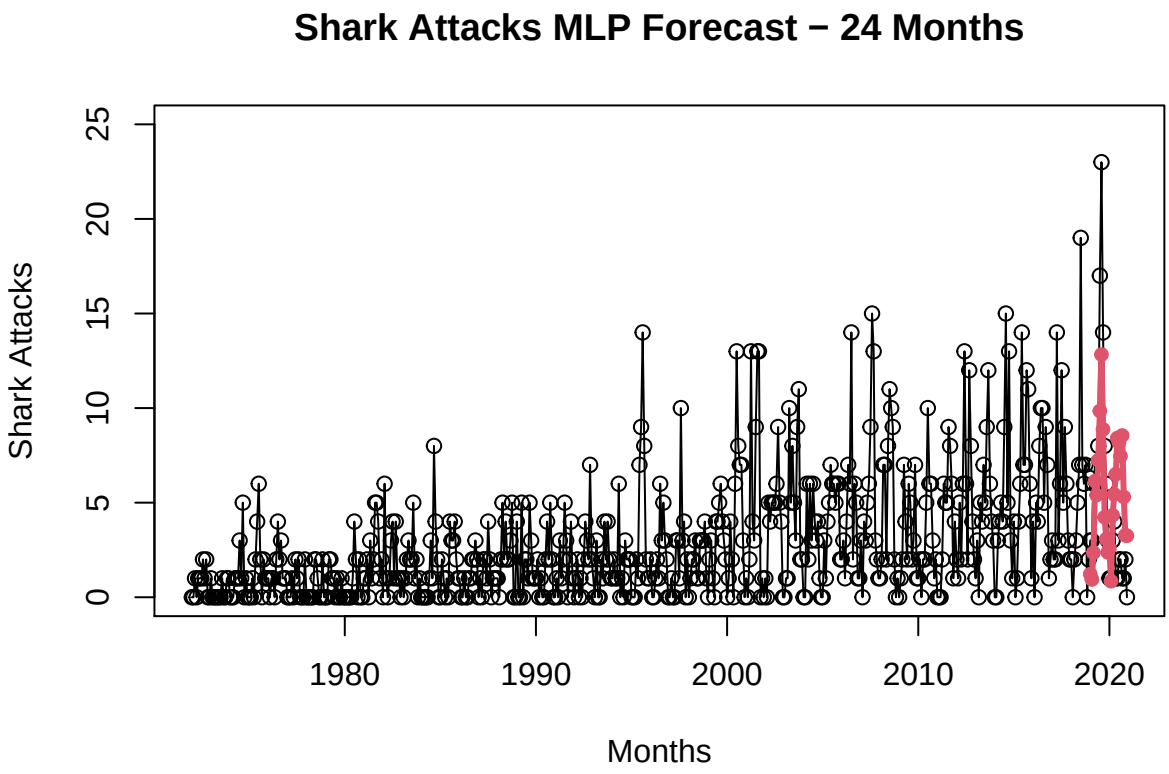
```
## [1] 19.73803
```

Get 12 Month ASE for MLP

```
ASE_MLP12 <- models[[23]]
ASE_MLP12
```

```
## [1] 18.86061
```

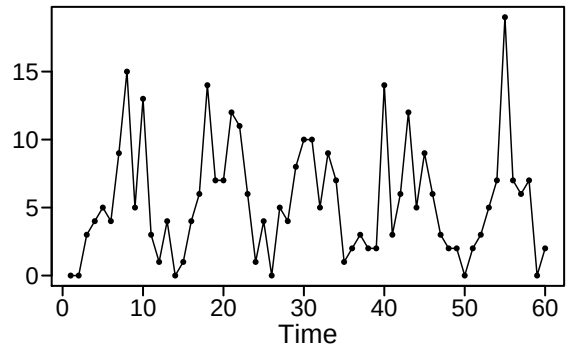
```
plot(ts.sharks,main='Shark Attacks MLP Forecast - 24 Months'
     ,ylim=c(0,25), ylab='Shark Attacks',type='o', xlab='Months')
points(models[[22]],type='o',lty=1,lwd=3,col=2,cex=.6)
```



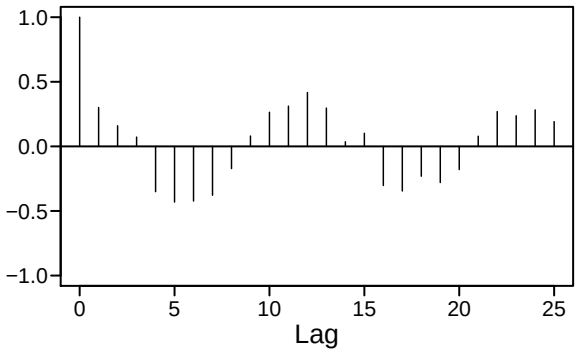
Generate Forecasts Using Different Time Ranges for Training

(note: plots from artrans.wge could not be suppressed using 'plottr = F' when called by the user defined functions)

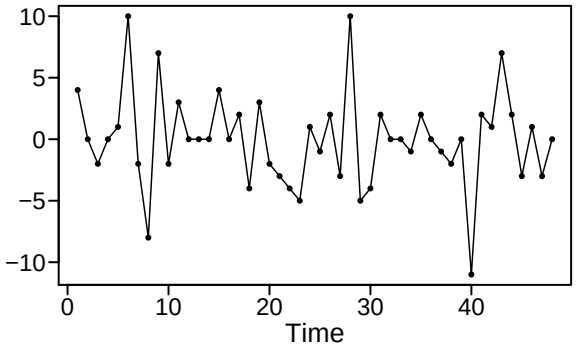
```
forecast_years <- list(5,10,20,30,40,47)
# forecast_years <- list(5,10)
ln <- F
modelsAll <- list()

for (f in forecast_years){

  data <- udf_make_data(sharks, ice.cream, f, 2, ln = ln)

  m <- udf_make_models(data[[2]], data[[3]], data[[4]], data[[5]], ln = ln)

  tmp <- m

  modelsAll <- append(modelsAll,list(tmp))
}
```



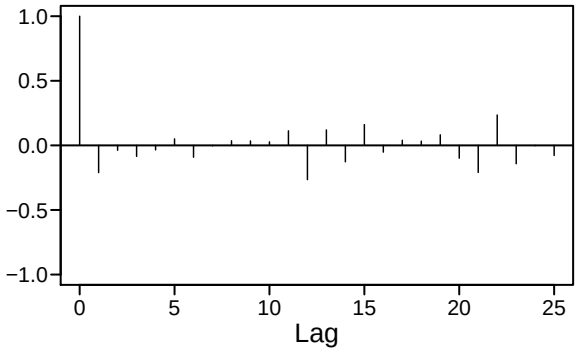
Realization



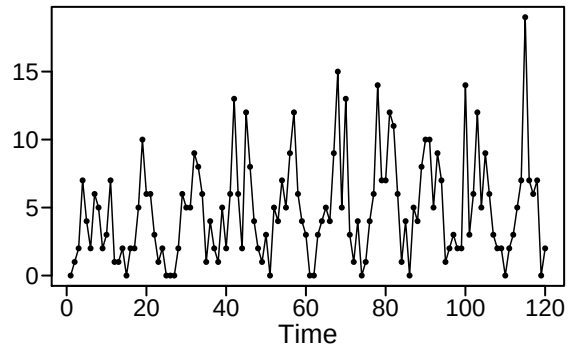
Sample Autocorrelations



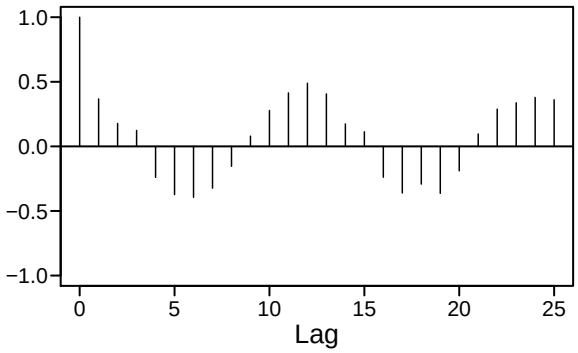
Transformed Realization



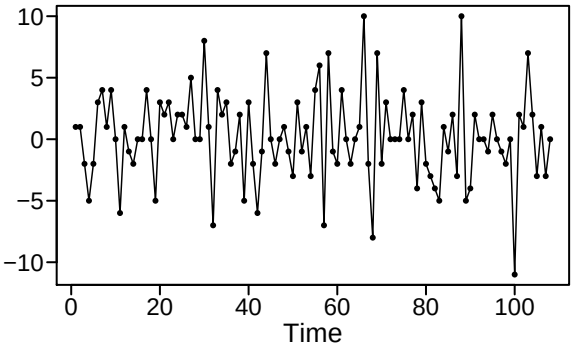
Sample Autocorrelations



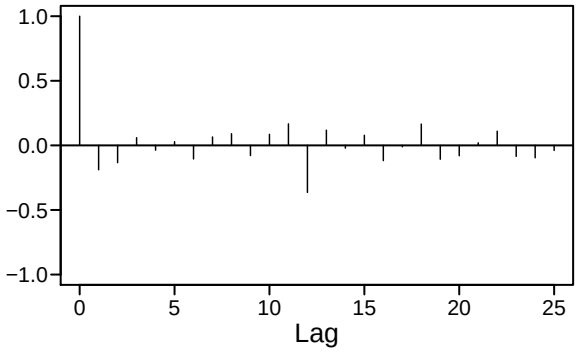
Realization



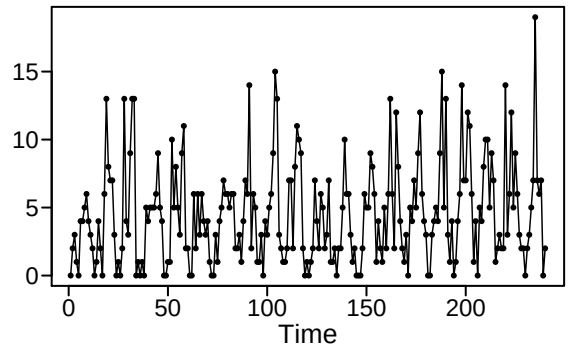
Sample Autocorrelations



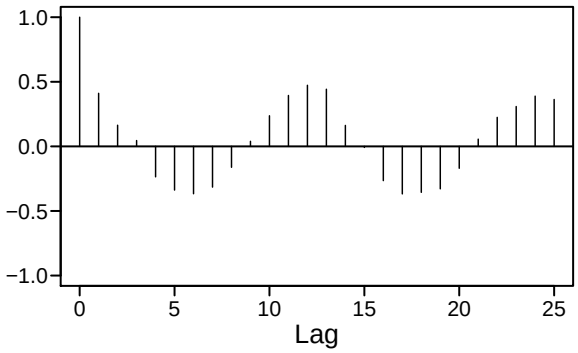
Transformed Realization



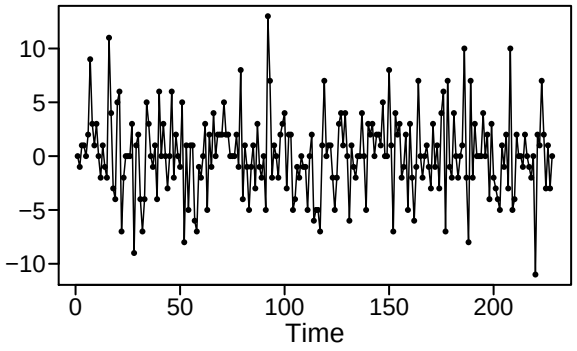
Sample Autocorrelations



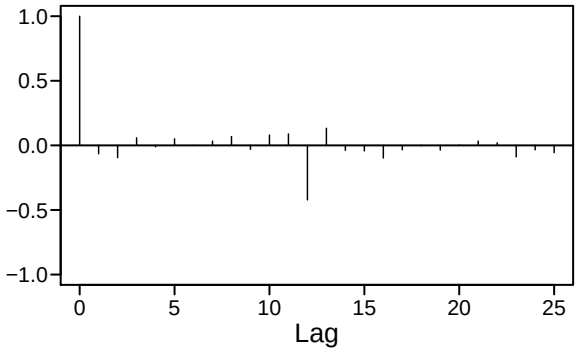
Realization



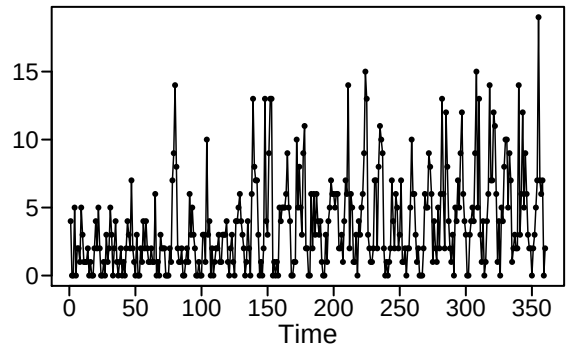
Sample Autocorrelations



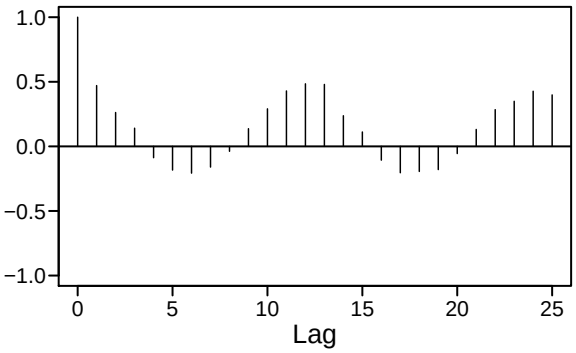
Transformed Realization



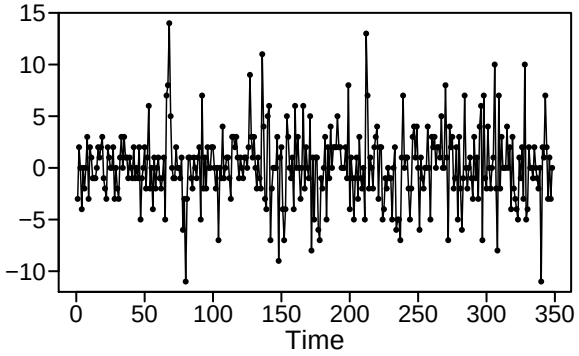
Sample Autocorrelations



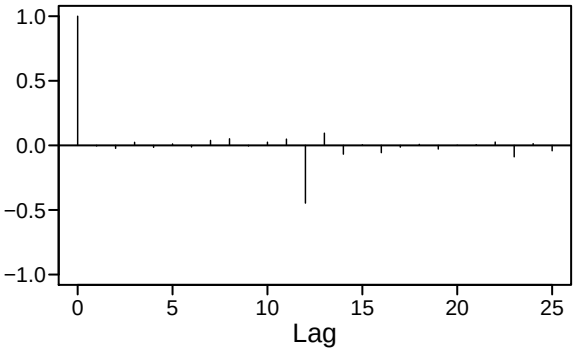
Realization



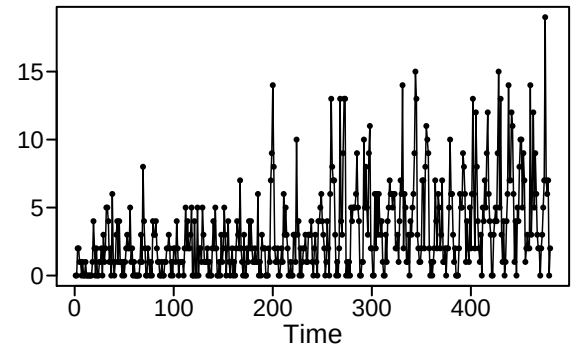
Sample Autocorrelations



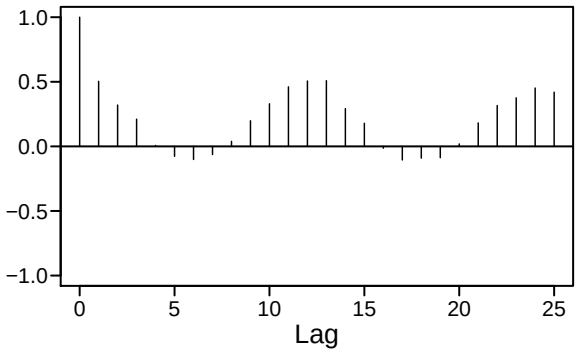
Transformed Realization



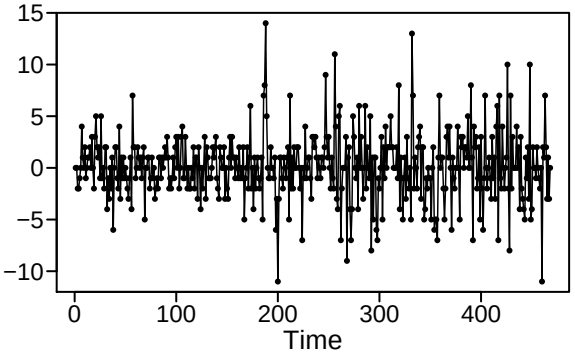
Sample Autocorrelations



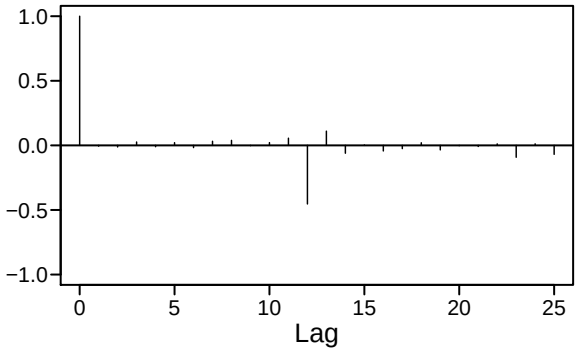
Realization



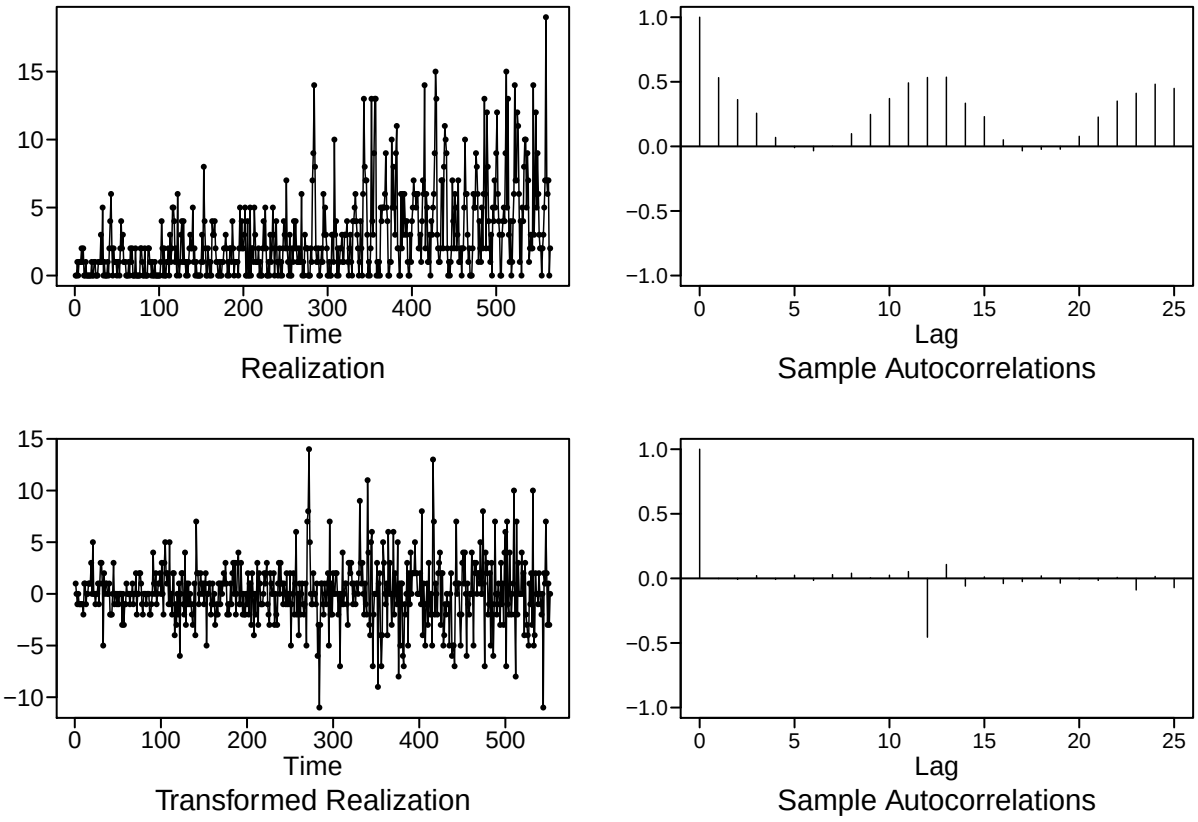
Sample Autocorrelations



Transformed Realization



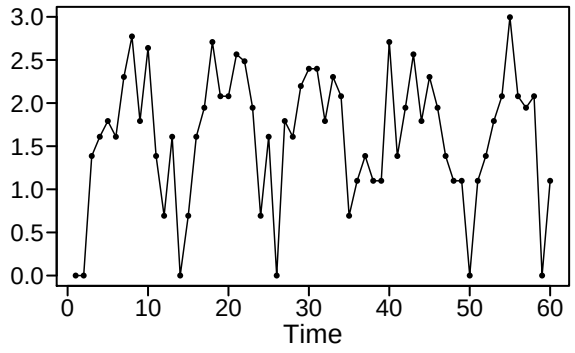
Sample Autocorrelations



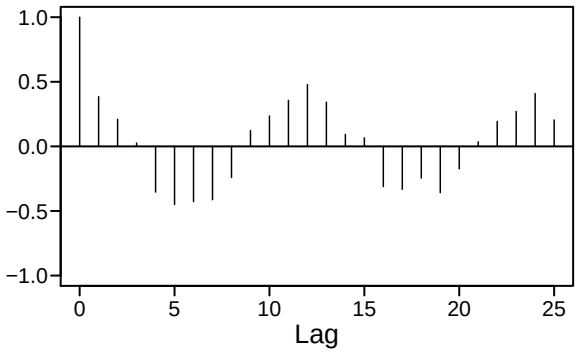
Generate Log Forecasts Using Different Time Ranges for Training

(note: plots from artrans.wge could not be suppressed using 'plottr = F' when called by the user defined functions)

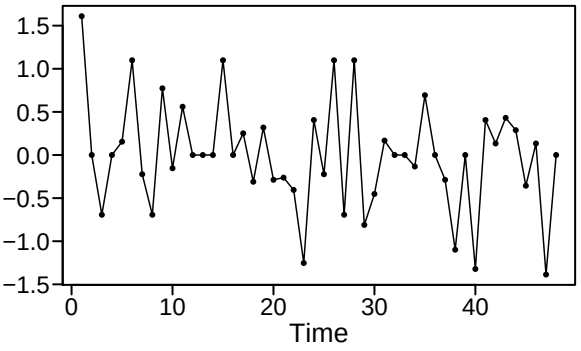
```
ln <- T
modelsAllLog <- list()

for (f in forecast_years){

  data <- udf_make_data(sharks, ice.cream, f, 2, ln = ln)

  m <- udf_make_models(data[[2]], data[[3]], data[[4]], data[[5]], ln = ln)

  tmp <- m

  modelsAllLog <- append(modelsAllLog,list(tmp))
}
```



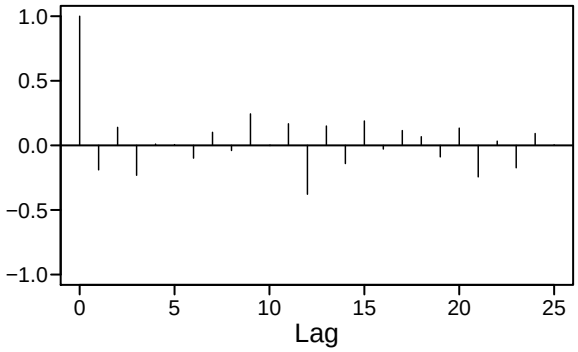
Realization



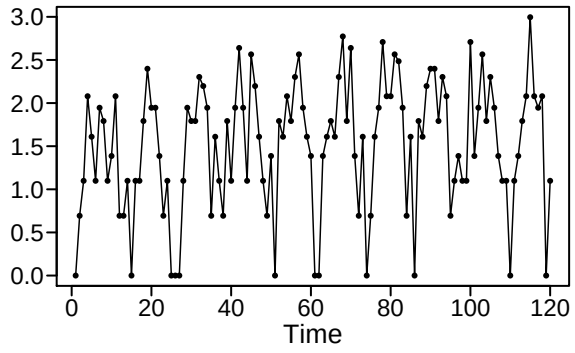
Sample Autocorrelations



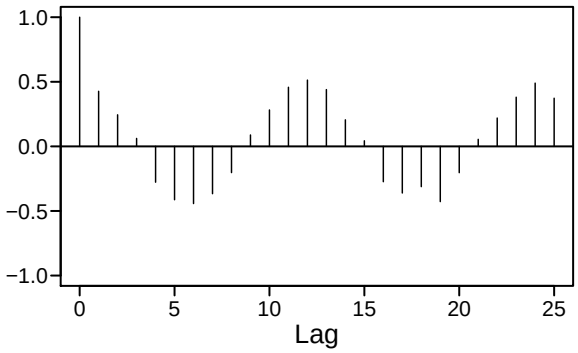
Transformed Realization



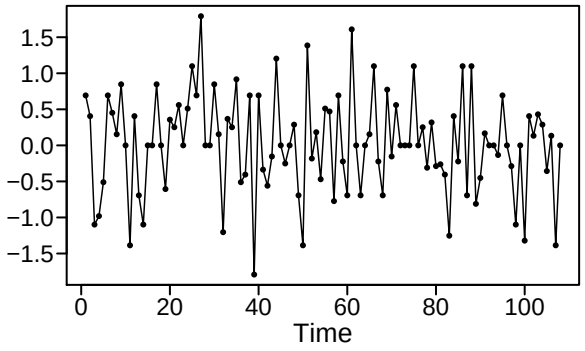
Sample Autocorrelations



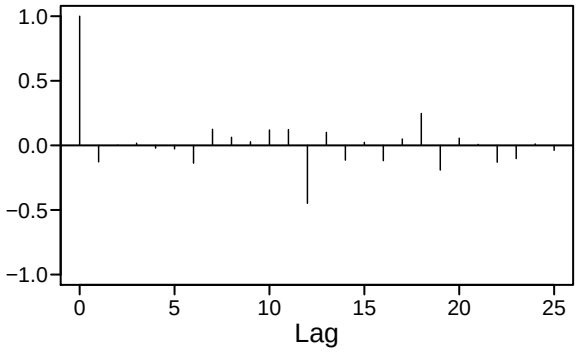
Realization



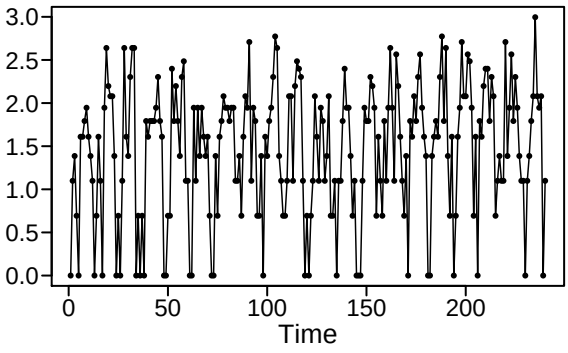
Sample Autocorrelations



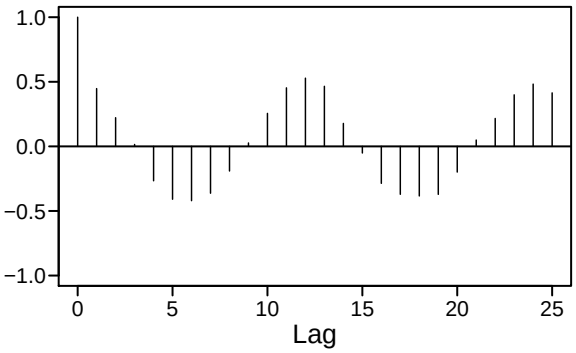
Transformed Realization



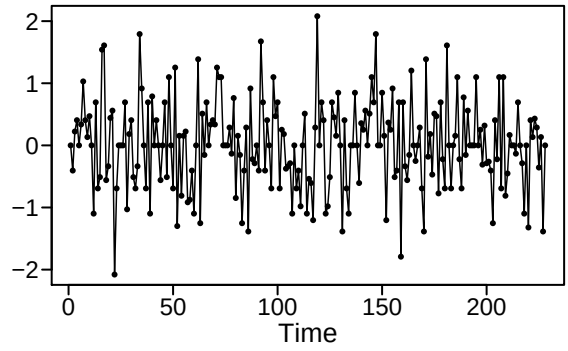
Sample Autocorrelations



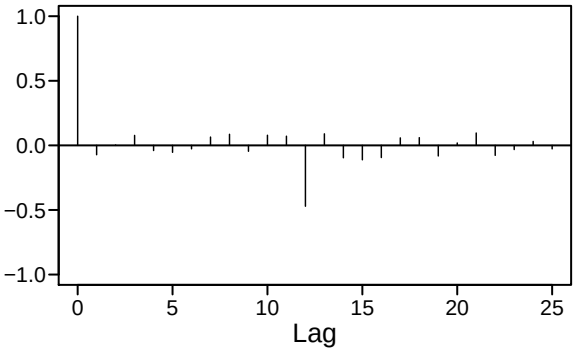
Realization



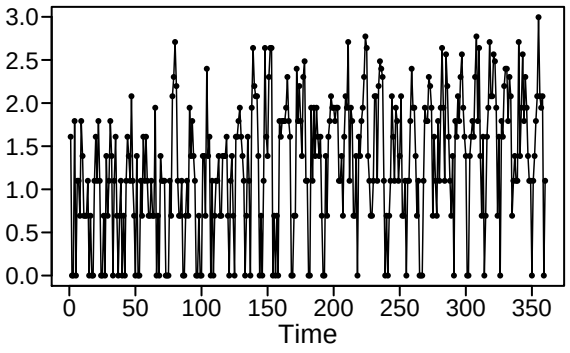
Sample Autocorrelations



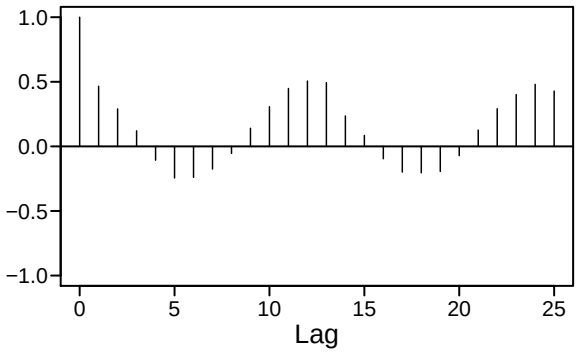
Transformed Realization



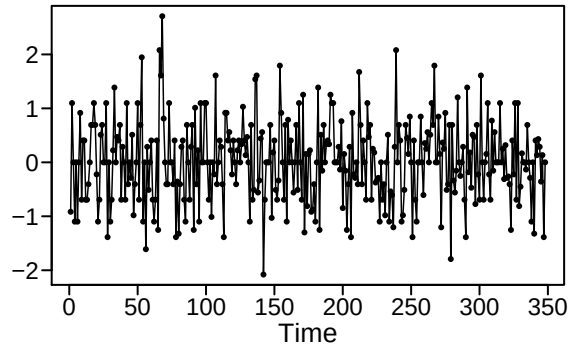
Sample Autocorrelations



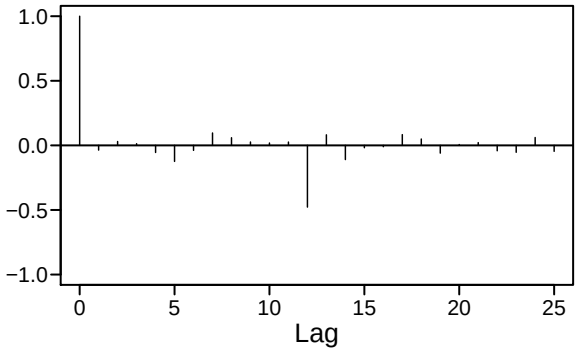
Realization



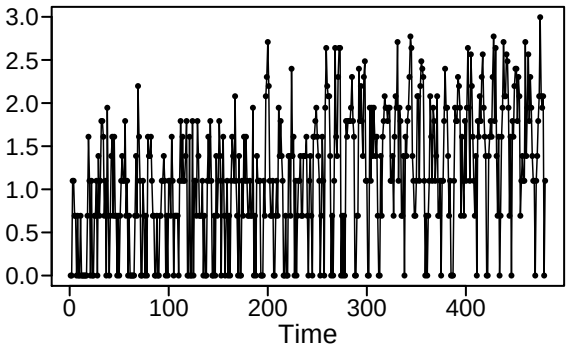
Sample Autocorrelations



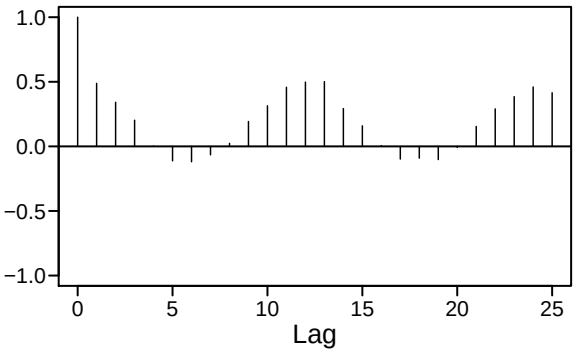
Transformed Realization



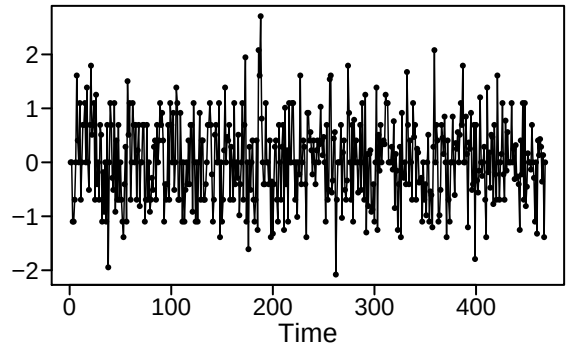
Sample Autocorrelations



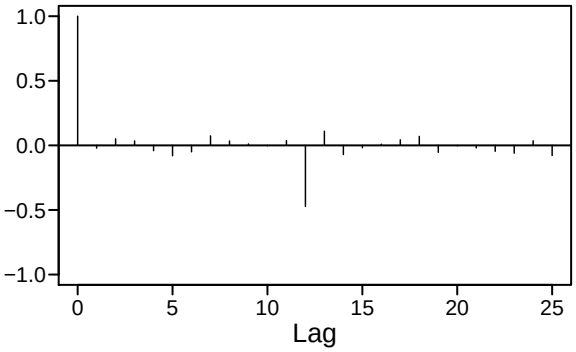
Realization



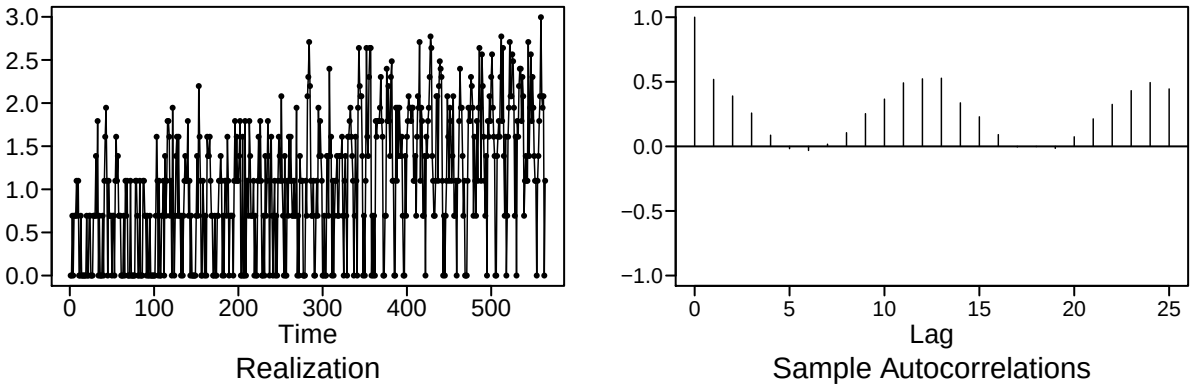
Sample Autocorrelations

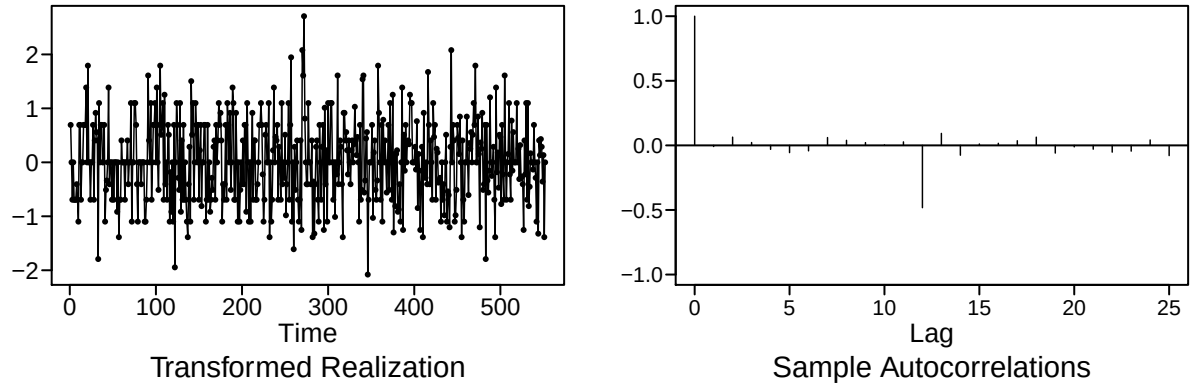


Transformed Realization



Sample Autocorrelations





Compile ASEs for all Models

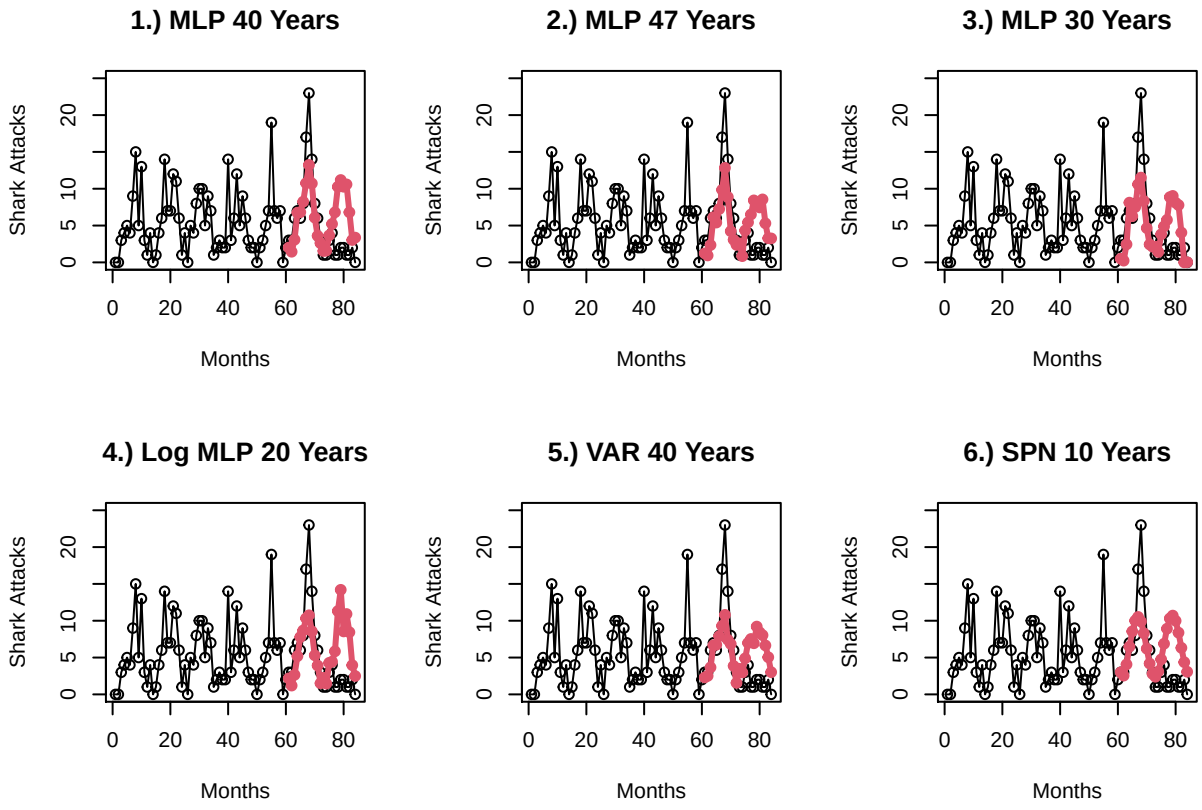
ASEs in ascending order (12 month)

```
##    forecast_type train_yrs     ASE24    ASE12 model_type
## 25           MLP        40  24.17627 13.94646    non-log
## 30           MLP        47  19.73803 18.86061    non-log
## 20           MLP        30  20.45168 21.47237    non-log
## 45           MLP        20  31.22487 21.70809        log
## 24           VAR        40  23.32084 22.19930    non-log
## 8            SPN        10  28.00209 22.31104    non-log
## 29           VAR        47  23.42438 22.72306    non-log
## 19           VAR        30  23.51871 23.90235    non-log
## 15           MLP        20  20.17811 24.04659    non-log
## 23           SPN        40  23.60285 24.19240    non-log
## 50           MLP        30  30.68542 24.52541        log
## 18           SPN        30  25.91957 24.68783    non-log
## 14           VAR        20  22.86156 25.98764    non-log
## 28           SPN        47  25.32576 26.37848    non-log
## 13           SPN        20  33.51356 26.76770    non-log
## 49           VAR        30  23.61648 27.25673        log
## 54           VAR        40  23.27310 27.36579        log
## 59           VAR        47  24.53158 27.77862        log
## 22    Stationary        40  21.94636 28.37581    non-log
## 17    Stationary        30  21.96539 28.43889    non-log
```

```
## 16      Seasonal        30  31.08469 29.93559    non-log
## 27    Stationary        47  24.35539 30.07936    non-log
## 21      Seasonal        40  31.46822 30.34719    non-log
## 44           VAR        20  23.87066 30.64914        log
## 26      Seasonal        47  30.80870 31.14982    non-log
## 7     Stationary        10  25.01512 31.44783    non-log
## 9            VAR        10  33.11336 31.86288    non-log
## 55           MLP        40  25.99166 32.54264        log
## 10           MLP        10  46.46630 32.83943    non-log
## 6       Seasonal        10  35.21148 33.80408    non-log
## 1       Seasonal         5  35.20833 33.83333    non-log
## 11      Seasonal        20  35.20833 33.83333    non-log
## 31      Seasonal         5  35.20833 33.83333        log
## 5            MLP         5  37.17581 34.12156    non-log
## 56      Seasonal        47  32.19038 34.59104        log
## 51      Seasonal        40  32.15402 34.71891        log
## 42    Stationary        20  23.58913 35.35585        log
## 3            SPN         5  26.65559 35.39763    non-log
## 60           MLP        47  25.48046 35.39768        log
## 46      Seasonal        30  32.69728 35.56915        log
## 2     Stationary         5  26.11337 35.65985    non-log
## 41      Seasonal        20  44.32736 37.47984        log
## 47    Stationary        30  23.56662 38.80460        log
## 35           MLP         5  33.90482 39.75578        log
## 52    Stationary        40  23.69754 40.17877        log
## 12    Stationary        20  38.75814 41.03829    non-log
## 36      Seasonal        10  32.64608 41.14545        log
## 57    Stationary        47  24.00591 41.44270        log
## 48           SPN        30  28.86265 43.98176        log
## 32    Stationary         5  27.91553 44.54410        log
## 53           SPN        40  28.77679 44.82863        log
## 39           VAR        10  46.13437 44.84012        log
## 58           SPN        47  28.78868 45.28191        log
## 38           SPN        10  30.47405 46.20524        log
## 37    Stationary        10  26.80552 48.31432        log
## 33           SPN         5  29.93663 49.94045        log
## 43           SPN        20  32.80784 54.89691        log
## 40           MLP        10  34.53767 55.87476        log
## 34           VAR         5  35.30869 56.62109        log
## 4            VAR         5 115.18472 83.89949    non-log
```

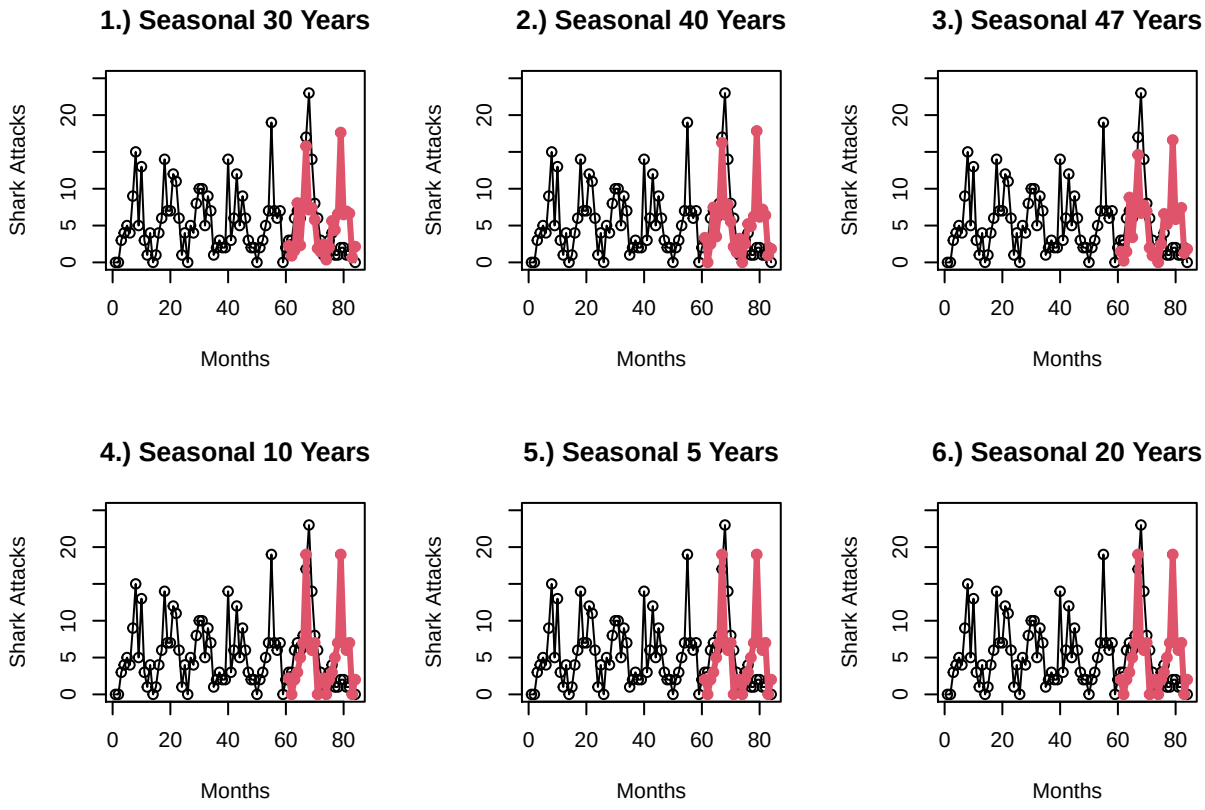
Look at Top Six Models Using 12 Month ASE

```
par(mfrow=c(2,3))
udf_make_plots(sharks, modelsAll[[5]][[22]], yrs_train = 40, title = '1.) MLP 40 Years')
udf_make_plots(sharks, modelsAll[[6]][[22]], yrs_train = 47, title = '2.) MLP 47 Years')
udf_make_plots(sharks, modelsAll[[4]][[22]], yrs_train = 30, title = '3.) MLP 30 Years')
udf_make_plots(sharks, modelsAllLog[[3]][[22]], yrs_train = 20, title = '4.) Log MLP 20 Years')
udf_make_plots(sharks, modelsAll[[5]][[18]], yrs_train = 40, title = '5.) VAR 40 Years')
udf_make_plots(sharks, modelsAll[[2]][[14]], yrs_train = 10, title = '6.) SPN 10 Years')
```



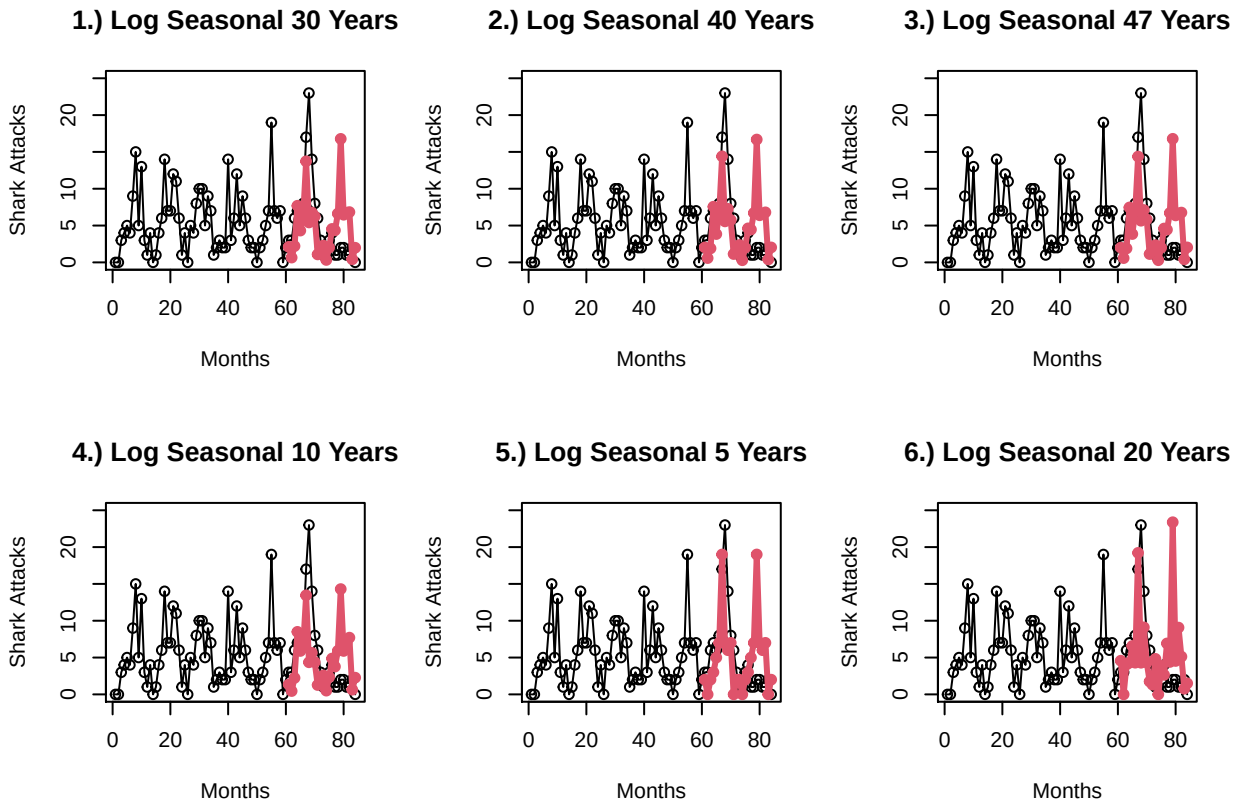
Look at All Seasonal Models by 12 Month ASE

```
par(mfrow=c(2,3))
udf_make_plots(sharks, modelsAll[[4]][[3]], yrs_train = 30, title = '1.) Seasonal 30 Years')
udf_make_plots(sharks, modelsAll[[5]][[3]], yrs_train = 40, title = '2.) Seasonal 40 Years')
udf_make_plots(sharks, modelsAll[[6]][[3]], yrs_train = 47, title = '3.) Seasonal 47 Years')
udf_make_plots(sharks, modelsAll[[2]][[3]], yrs_train = 10, title = '4.) Seasonal 10 Years')
udf_make_plots(sharks, modelsAll[[1]][[3]], yrs_train = 5, title = '5.) Seasonal 5 Years')
udf_make_plots(sharks, modelsAll[[3]][[3]], yrs_train = 20, title = '6.) Seasonal 20 Years')
```



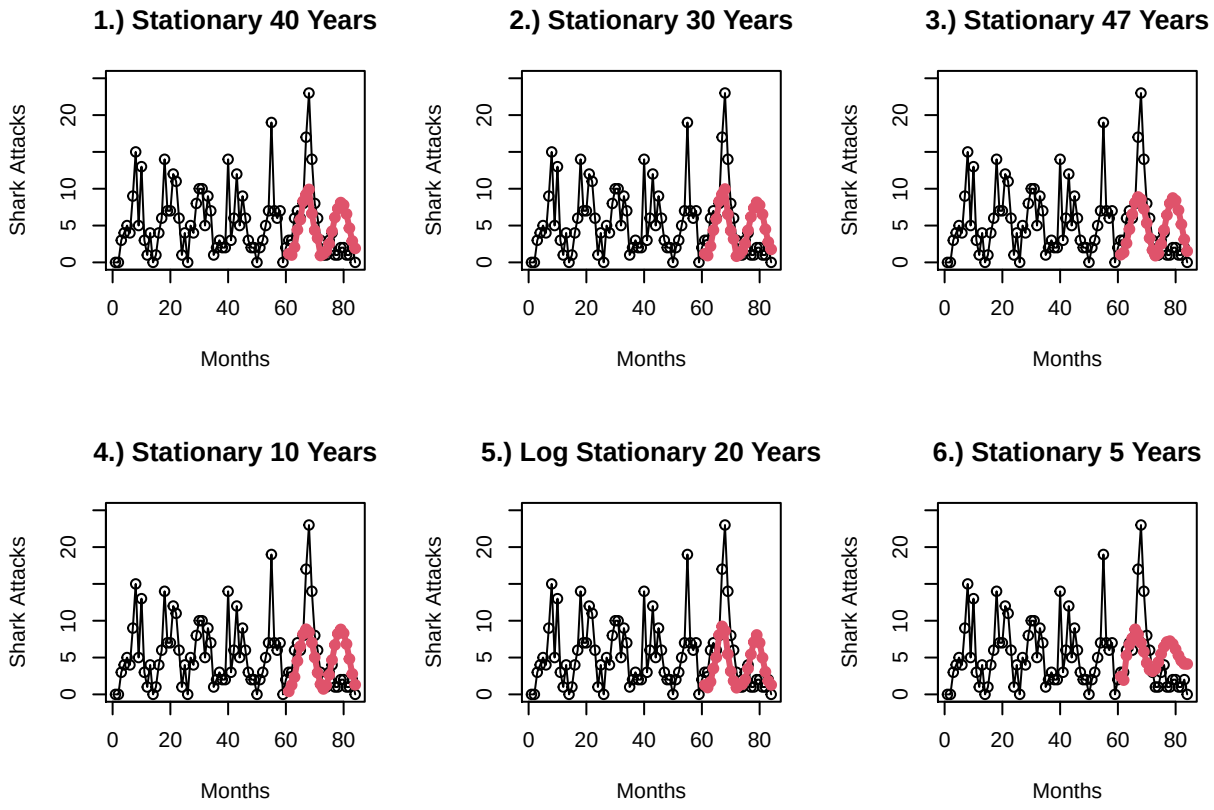
Look at All Seasonal Log Models by 12 Month ASE

```
par(mfrow=c(2,3))
udf_make_plots(sharks, modelsAllLog[[4]][[3]], yrs_train = 30, title = '1.) Log Seasonal 30 Years')
udf_make_plots(sharks, modelsAllLog[[5]][[3]], yrs_train = 40, title = '2.) Log Seasonal 40 Years')
udf_make_plots(sharks, modelsAllLog[[6]][[3]], yrs_train = 47, title = '3.) Log Seasonal 47 Years')
udf_make_plots(sharks, modelsAllLog[[2]][[3]], yrs_train = 10, title = '4.) Log Seasonal 10 Years')
udf_make_plots(sharks, modelsAllLog[[1]][[3]], yrs_train = 5, title = '5.) Log Seasonal 5 Years')
udf_make_plots(sharks, modelsAllLog[[3]][[3]], yrs_train = 20, title = '6.) Log Seasonal 20 Years')
```



Look at Top Six Stationary Models by 12 Month ASE

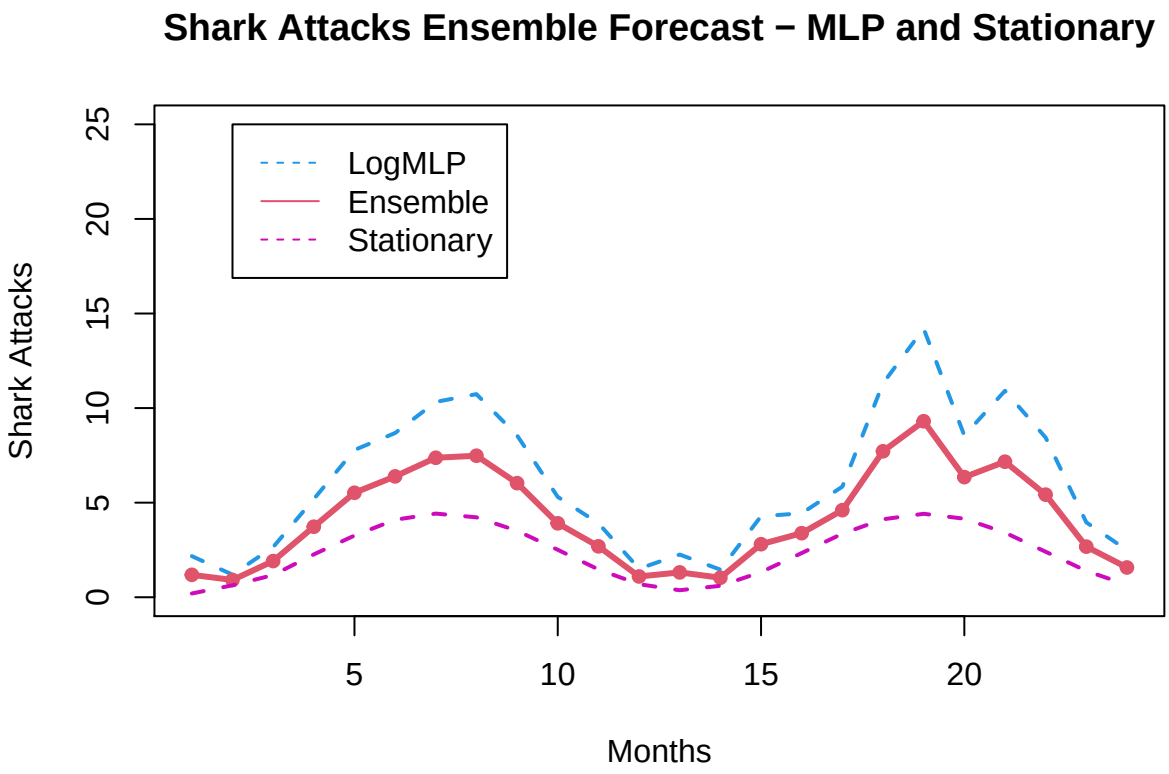
```
par(mfrow=c(2,3))
udf_make_plots(sharks, modelsAll[[5]][[7]], yrs_train = 40, title = '1.) Stationary 40 Years')
udf_make_plots(sharks, modelsAll[[4]][[7]], yrs_train = 30, title = '2.) Stationary 30 Years')
udf_make_plots(sharks, modelsAll[[6]][[7]], yrs_train = 47, title = '3.) Stationary 47 Years')
udf_make_plots(sharks, modelsAll[[2]][[7]], yrs_train = 10, title = '4.) Stationary 10 Years')
udf_make_plots(sharks, modelsAllLog[[3]][[7]], yrs_train = 20, title = '5.) Log Stationary 20 Years')
udf_make_plots(sharks, modelsAll[[1]][[7]], yrs_train = 5, title = '6.) Stationary 5 Years')
```



Ensemble Model

Look at one ensemble model Use lower amplitude stationary forecasts to ensemble with Log MLP forecast

```
fc_mlpLog20yrs <- modelsAllLog[[3]][[22]][241:264]
fc_stat10yrs <- modelsAll[[2]][[7]][121:144]

ensemble <- (fc_mlpLog20yrs + (.5 * fc_stat10yrs)) / 2 # wtd avg. stationary 1/2 the weight

t <- 1:length(ensemble)

plot(t,fc_mlpLog20yrs,main='Shark Attacks Ensemble Forecast - MLP and Stationary'
     ,ylim=c(0,25), ylab='Shark Attacks',type='l',lty=2, xlab='Months', col=4,lwd=2)
points(t,ensemble,type='o',lty=1,lwd=3,col=2,cex=.6)
points(t,0.5*fc_stat10yrs,type='l',lty=2, col=6,lwd=2)
legend(2,25, legend=c('LogMLP','Ensemble','Stationary'), col=c(4,2,6), lty=2:1)
```



Final Model Forecast - Piecewise

Ensemble for months 1 - 12
Log MLP 20 Years for months 13 - 24

where Ensemble is the weighted average of Log MLP 20 year and Stationary 10 year

$$forecast_p = modelA + modelB$$

$$where$$

$$modelA = MLPLog20yrs(1:12) + (1/2)Stationary10yrs(1:12)$$

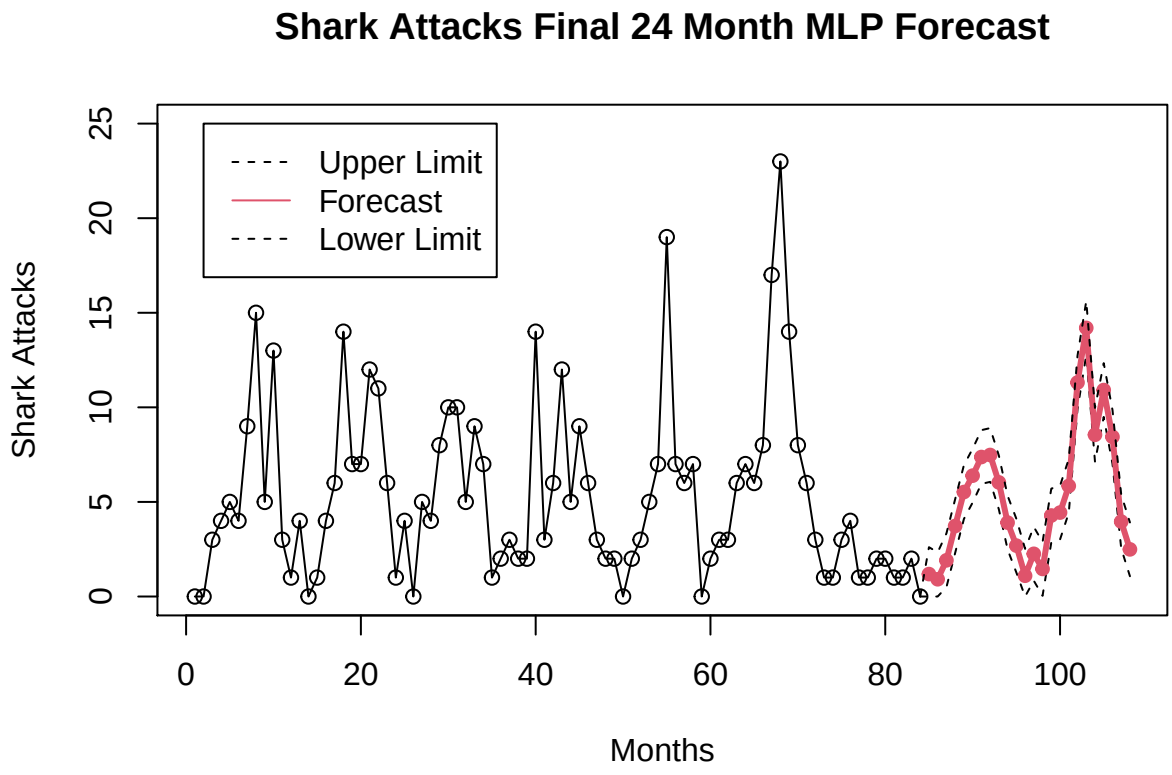
$$and$$

$$modelB = MLPLog20yrs(13:24)$$

Plot Final Forecast

```
piecewise <- c(ensemble[1:12],fc_mlpLog20yrs[13:24])
# 24 month

# expand time for plotting
piecewise <- c(rep(NA,288),piecewise)
```

```
piecewise <- piecewise[(length(piecewise) - 108 + 1): length(piecewise)]

# get actual data, line up for plotting
ts.sharks.20yr <- ts.sharks[(length(ts.sharks) - 84 + 1): length(ts.sharks)]
ts.sharks.20yr <- c(ts.sharks.20yr,rep(NA,24))

# plot final model with CIs
t <- 1:length(ts.sharks.20yr)

# get CIs
CU <- piecewise + 1.96 * std.error(piecewise)
CU <- c(CU[1:83],0,CU[85:108])
CL <- piecewise - 1.96 * std.error(piecewise)
CL[CL<0] <- 0
CL <- c(CL[1:83],0,CL[85:108])

plot(t,ts.sharks.20yr,main='Shark Attacks Final 24 Month MLP Forecast'
     ,ylim=c(0,25), ylab='Shark Attacks',type='o', xlab='Months')
points(t,CL,type='l',lty=2)
points(t,piecewise,type='o',lty=1,lwd=3,col=2,cex=.6)
points(t,CU,type='l',lty=2)
legend(2,25, legend=c('Upper Limit','Forecast','Lower Limit'), col=c(1,2,1), lty=2:1)
```



Total Shark Attacks Projected in the U.S - 2021

```
f_2021 <- round(ts(piecewise[!is.na(piecewise)][1:12], frequency = 12, start = 2021),0)
print(f_2021)
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2021   1   1   2   4   6   6   7   7   6   4   3   1
```

```
print(paste('2021 Total Shark Attacks Projected -',sum(f_2021), sep=' '))
```

```
## [1] "2021 Total Shark Attacks Projected - 48"
```

Total Shark Attacks Projected in the U.S - 2022

```
f_2022 <- round(ts(piecewise[!is.na(piecewise)][13:24], frequency = 12, start = 2022),0)
print(f_2022)
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2022   2   1   4   4   6  11  14   9  11   8   4   2
```

```
print(paste('2022 Total Shark Attacks Projected -',sum(f_2022), sep=' '))
```

```
## [1] "2022 Total Shark Attacks Projected - 76"
```

Bring Prior Three Years of Actual Shark Attacks for Forecast Perspective

```
act_2020 <- sum(ts.sharks.20yr[!is.na(ts.sharks.20yr)][73:84])
act_2019 <- sum(ts.sharks.20yr[!is.na(ts.sharks.20yr)][61:72])
act_2018 <- sum(ts.sharks.20yr[!is.na(ts.sharks.20yr)][49:60])

actualPast3 <- ts(c(act_2018,act_2019,act_2020),frequency=1,start=2018)
proj2 <- ts(c(sum(f_2021),sum(f_2022)),frequency=1,start=2021)

print(actualPast3)
```

```
## Time Series:
## Start = 2018
## End = 2020
## Frequency = 1
## [1]  60 104  19
```

```
print(proj2)
```

```
## Time Series:
## Start = 2021
## End = 2022
## Frequency = 1
## [1] 48 76
```

Upper Limits - Total Shark Attacks Projected in the U.S - 2021

```
f_2021 <- round(ts(CU[!is.na(CU)][1:12], frequency = 12, start = 2021),0)
print(f_2021)
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2021   0   3   2   3   5   7   8   9   9   7   5   4
```

```
print(paste('2021 Total Shark Attacks Projected -',sum(f_2021), sep=' '))
```

```
## [1] "2021 Total Shark Attacks Projected - 62"
```

Lower Limits - Total Shark Attacks Projected in the U.S - 2021

```
f_2021 <- round(ts(CL[!is.na(CL)][1:12], frequency = 12, start = 2021),0)
print(f_2021)
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2021   0   0   0   0   2   4   5   6   6   5   2   1
```

```
print(paste('2021 Total Shark Attacks Projected -',sum(f_2021), sep=' '))
```

```
## [1] "2021 Total Shark Attacks Projected - 31"
```

Upper Limits - Total Shark Attacks Projected in the U.S - 2022

```
f_2022 <- round(ts(CU[!is.na(CU)][13:24], frequency = 12, start = 2022),0)
print(f_2022)
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2022   3   4   3   6   6   7  13  16  10  12  10   5
```

```
print(paste('2022 Total Shark Attacks Projected -',sum(f_2022), sep=' '))
```

```
## [1] "2022 Total Shark Attacks Projected - 95"
```

Lower Limits - Total Shark Attacks Projected in the U.S - 2022

```
f_2022 <- round(ts(CL[!is.na(CL)][13:24], frequency = 12, start = 2022),0)
print(f_2022)
```

```
##      Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
## 2022   0   1   0   3   3   4  10  13   7   9   7   3
```

```
print(paste('2022 Total Shark Attacks Projected -',sum(f_2022), sep=' '))
```

```
## [1] "2022 Total Shark Attacks Projected - 60"
```