In-Class Analysis 3

Jillian Warburton

2023-02-08

Exploratory Data Analysis

1. To make plots with appropriate axes that correspond to the year and month, in the dataset, create a new `YrMon` variable. I suggest setting `myData$YrMon <- myData$Year + (myData$Month - 0.5)/12`.

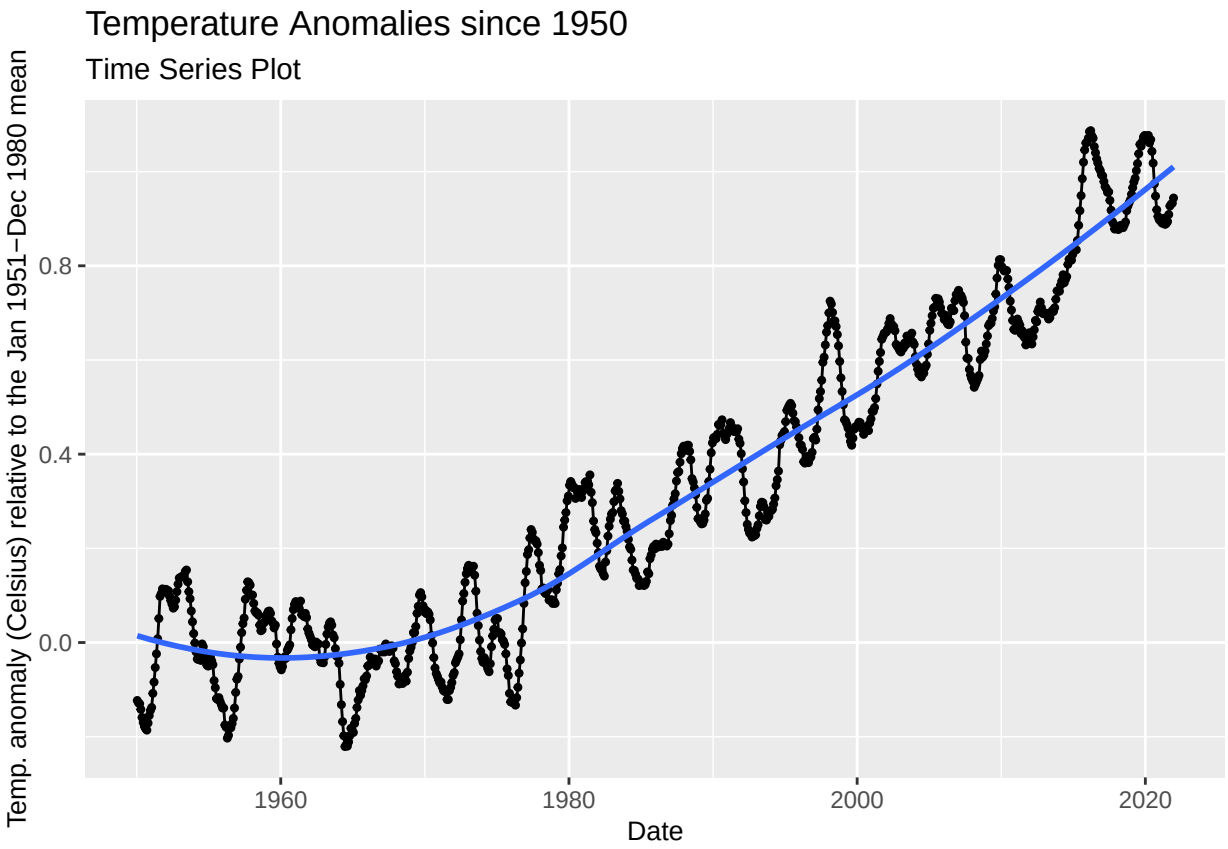
```
data$YrMon <- data$Year + (data$Month - 0.5)/12 #the -.5 is to set the date in the middle of the month
head(data)
```

```
## # A tibble: 6 x 4
##    Year Month AnnAnom YrMon
##   <dbl> <dbl>   <dbl> <dbl>
## 1  1950     1  -0.123 1950.
## 2  1950     2  -0.128 1950.
## 3  1950     3  -0.13  1950.
## 4  1950     4  -0.142 1950.
## 5  1950     5  -0.159 1950.
## 6  1950     6  -0.17  1950.
```

2. Draw a time series plot of the temperature anomalies with `YrMon` along the `x`-axis with a smooth curve overlaid to emphasize the non-linear aspect of the data.

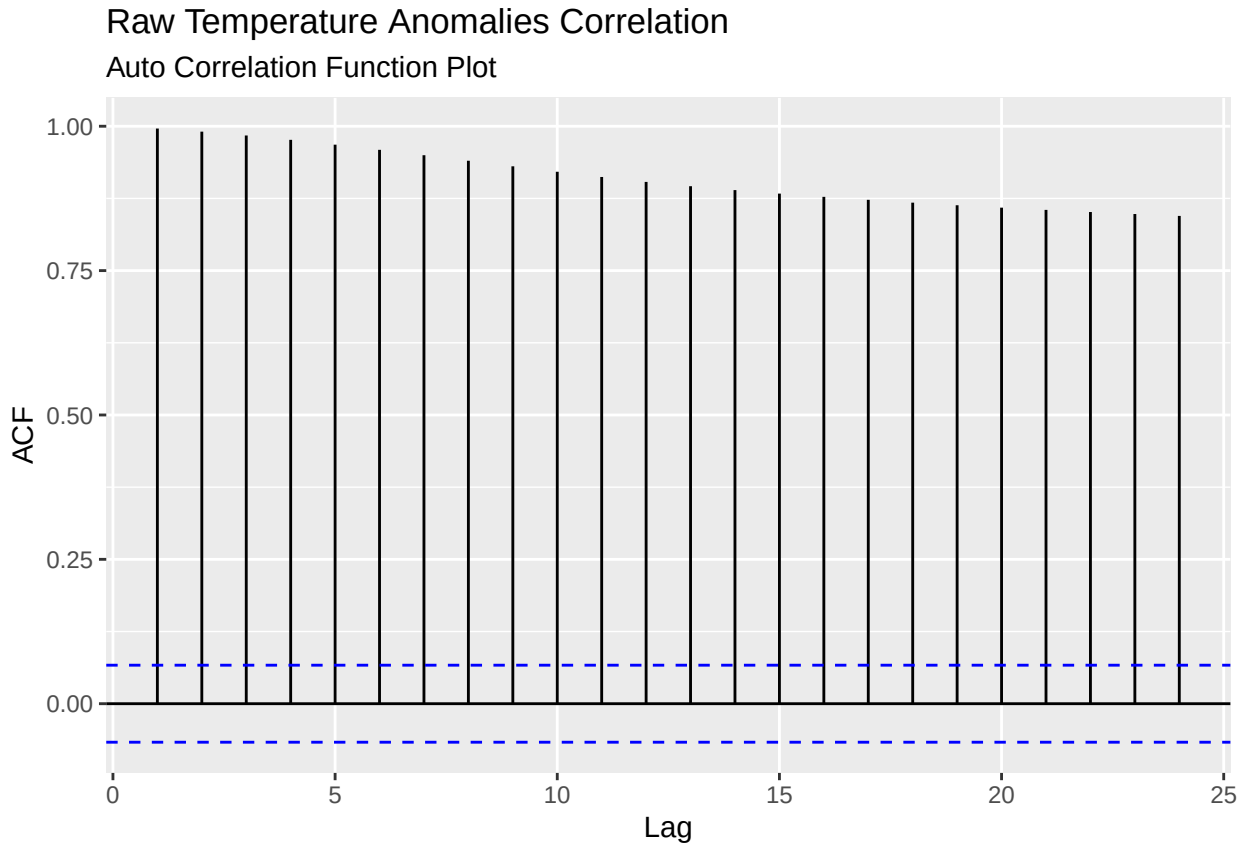
```
ggplot(data = data, mapping = aes(x=YrMon, y=AnnAnom)) +
  geom_point(size = 0.9) +
  geom_line() +
  geom_smooth(se = FALSE) +
  labs(title = "Temperature Anomalies since 1950",
       subtitle = "Time Series Plot") +
  ylab("Temp. anomaly (Celsius) relative to the Jan 1951-Dec 1980 mean") +
  xlab("Date") +
  theme(axis.title=element_text(size=10))
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



3. Even though we are supposed to draw ACFs of residuals (which we will in the next section after we learn how to fit splines), for now, calculate and draw the ACF of the raw temperature anomalies to show the strong seasonal correlation in the data (note: you may have to increase `lag.max` to show 2 or 3 seasonal cycles).

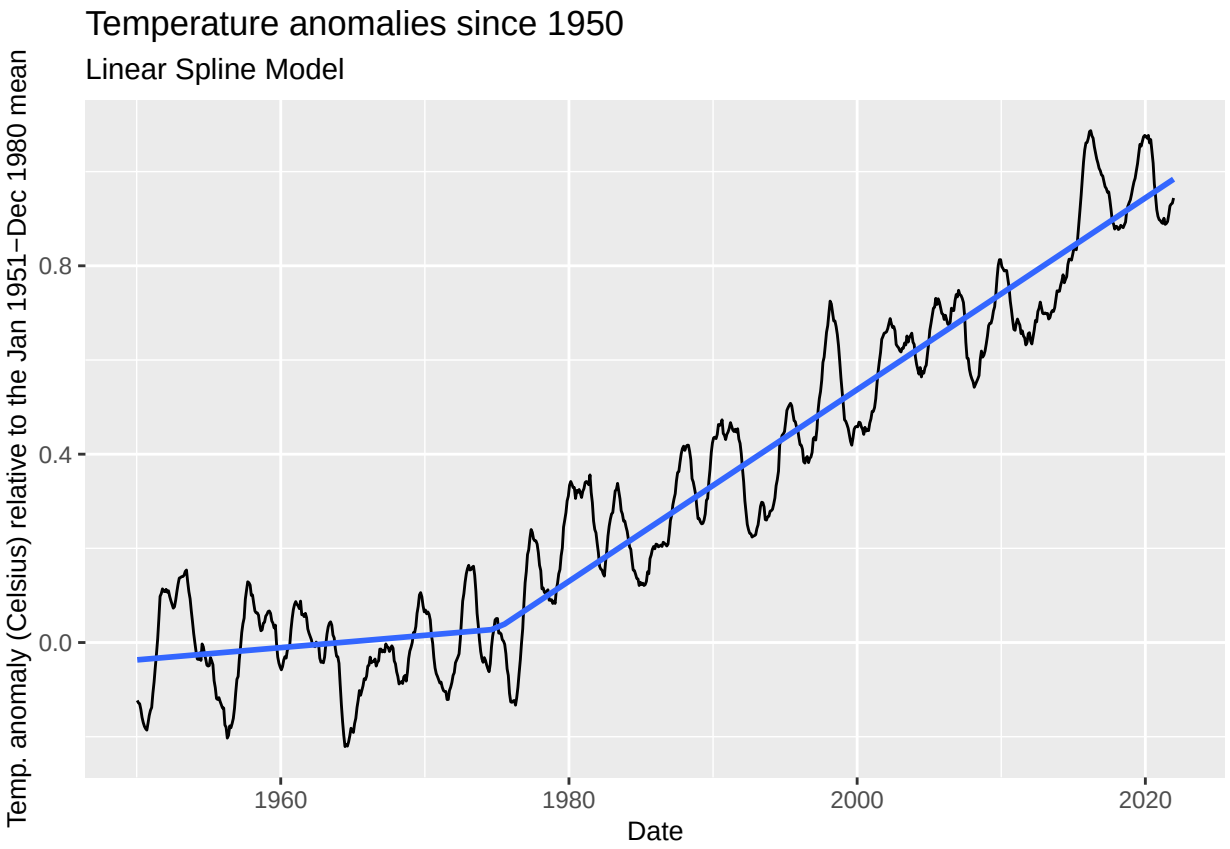
```
#my.ACF <- acf(x=stdres(clim.lm), lag.max=3)
#plot(my.ACF) #base R ACF plot

## ggplot() of ACF function
ggAcf(x=data$AnnAnom, lag.max=24) +
  labs(title = "Raw Temperature Anomalies Correlation",
       subtitle = "Auto Correlation Function Plot")
```



Fitting Time Series Models

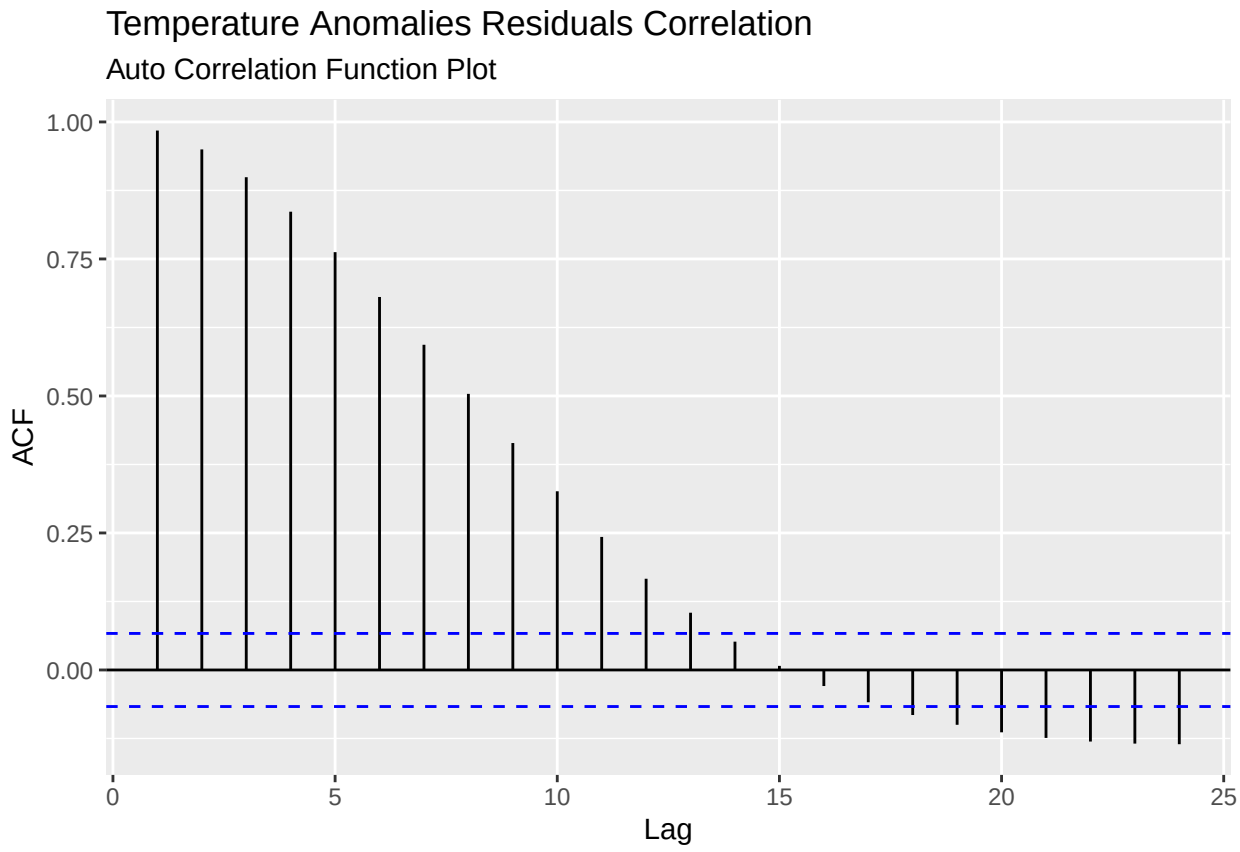
1. Fit a linear spline model with a knot point at 1975 to the annual temperature anomalies (i.e. use `lm()`). Create a plot of the fitted regression line on top of a time series plot to verify that the linear spline fits the data well (hint: add a `geom_smooth(method=lm, formula=y~x+pmax(x-1975, 0), se=FALSE)` to the time series plot).

```
ggplot(data = data, mapping = aes(x=YrMon, y=AnnAnom)) +
  geom_line() +
  geom_smooth(method=lm, formula=y~x+pmax(x-1975, 0), se=FALSE) +
  labs(title = "Temperature anomalies since 1950",
       subtitle = "Linear Spline Model") +
  ylab("Temp. anomaly (Celsius) relative to the Jan 1951-Dec 1980 mean") +
  xlab("Date") +
  theme(axis.title=element_text(size=10))
```



2. Draw an ACF of the residuals from your model in #2 to verify that there is indeed still temporal correlation in the residuals that we will need to model.

```
clim.lm <- lm(formula = AnnAnom~YrMon+pmax(YrMon-1975, 0), data = data)
ggAcf(x=stdres(clim.lm), lag.max=24) +
  labs(title = "Temperature Anomalies Residuals Correlation",
       subtitle = "Auto Correlation Function Plot")
```



3. Define a time series object of `AnnAnom` as use this object in all your analyses below.

```
my.ts <- ts(data=data$AnnAnom, start=c(1950, 1), frequency=12)
head(my.ts, 24)
```

```
##         Jan    Feb    Mar    Apr    May    Jun    Jul    Aug    Sep    Oct
## 1950 -0.123 -0.128 -0.130 -0.142 -0.159 -0.170 -0.178 -0.183 -0.186 -0.171
## 1951 -0.138 -0.108 -0.084 -0.053 -0.024  0.008  0.051  0.098  0.104  0.114
##         Nov    Dec
## 1950 -0.155 -0.144
## 1951  0.109  0.108
```

4. Create an $\mathbf{X}$ matrix for a linear spline for `YrMon` with a knot location at 1975 and without an intercept (the `Arima()` function automatically adds in an intercept for you so you don't need to include an intercept column in $\mathbf{X}$).

```
X <- model.matrix(AnnAnom~-1+YrMon+pmax(YrMon-1975, 0), data=data)
head(X)
```

```
##      YrMon pmax(YrMon - 1975, 0)
## 1 1950.042                     0
## 2 1950.125                     0
## 3 1950.208                     0
## 4 1950.292                     0
```

```
## 5 1950.375                     0
## 6 1950.458                     0
```

```
tail(X)
```

```
##        YrMon pmax(YrMon - 1975, 0)
## 859 2021.542              46.54167
## 860 2021.625              46.62500
## 861 2021.708              46.70833
## 862 2021.792              46.79167
## 863 2021.875              46.87500
## 864 2021.958              46.95833
```

5. Using `auto.arima()` choose which `p, d, q, P, D, Q` to use by comparing AIC or BIC values for models with an order of 2 or less (i.e. `max.p=max.q=max.Q=max.P=2`) and a max differencing of 1 (i.e. `max.d=max.D=1`). Make sure to use your linear spline as the `xreg=` value.

```
# auto.arima(my.ts, max.p=2, max.q=2,max.P=2, max.Q=2, max.d=1, max.D=1,
#            ic='aic', stepwise=FALSE, xreg=X)

#This is the response I got from the above code
# Series: my.ts
# Regression with ARIMA(2,0,1)(2,0,0)[12] errors
#
# Coefficients:
#          ar1      ar2      ma1     sar1     sar2  intercept   YrMon  pmax(YrMon - 1975, 0)
#       1.8859  -0.8938  -0.4566  -0.7299  -0.3361    -5.9166  0.0030                 0.0173
# s.e.  0.0209   0.0208   0.0422   0.0328   0.0328     3.5840  0.0018                 0.0024
#
# sigma^2 = 9.451e-05:  log likelihood = 2775.23
# AIC=-5532.47   AICc=-5532.26   BIC=-5489.61
```

6. Fit your chosen model and examine the model estimates.

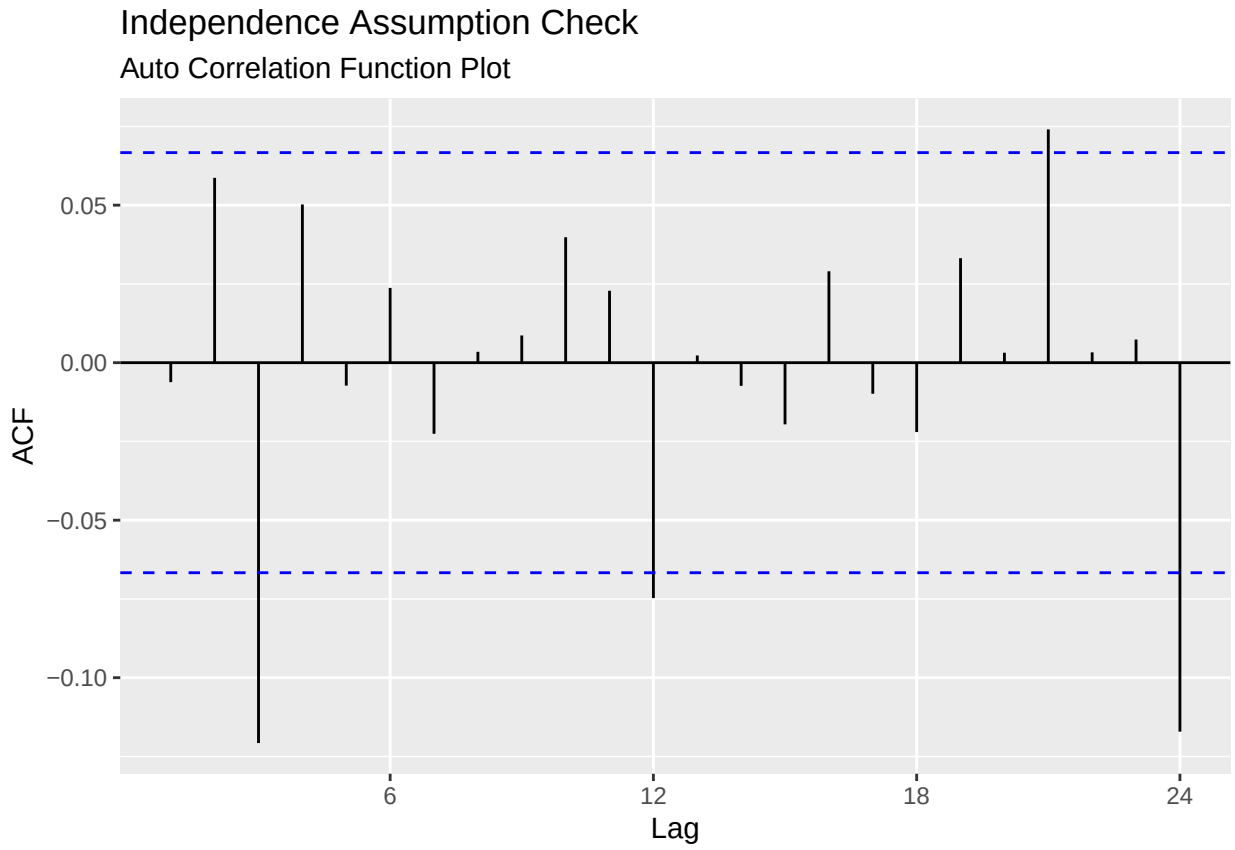
```
my.sarima.model <- Arima(my.ts, order=c(2, 0, 1), seasonal=c(2, 0, 0), xreg=X)
coef(my.sarima.model)
```

```
##                  ar1                  ar2                  ma1
##           1.88585946          -0.89376091          -0.45656850
##                 sar1                 sar2            intercept
##          -0.72989790          -0.33607136          -5.91658187
##                YrMon pmax(YrMon - 1975, 0)
##           0.00301119           0.01725758
```

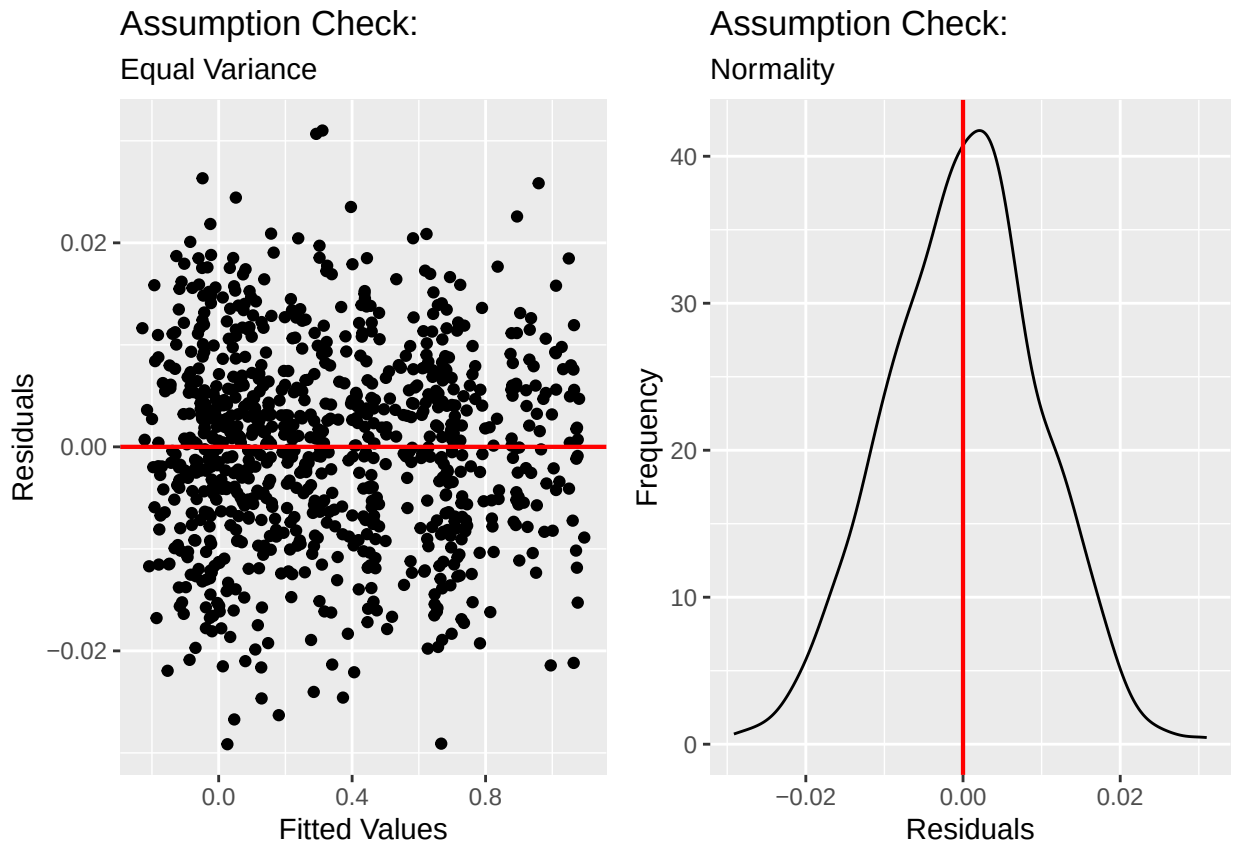
Model Validation

1. Verify your assumptions of independence, normality and equal variance by drawing an ACF of the decorrelated residuals, a fitted values vs. residual plot and a histogram (or density plot) of the decorrelated residuals.

```
ggAcf(x=resid(my.sarima.model), lag.max=24) +
  labs(title = "Independence Assumption Check",
       subtitle = "Auto Correlation Function Plot")
```

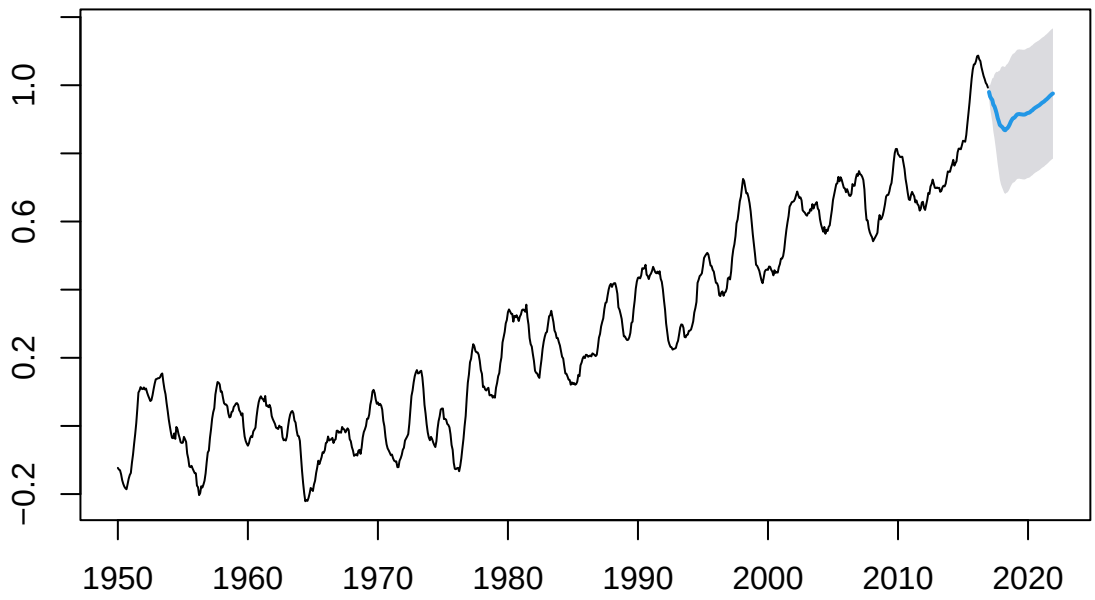



```
ad.resids_fit2 <- ggplot(data, aes(x=fitted(my.sarima.model), y=resid(my.sarima.model))) +
  geom_point() +
  xlab('Fitted Values') +
  ylab('Residuals') +
  ggtitle('Assumption Check:', subtitle ='Equal Variance') +
  geom_hline(yintercept = 0, col = "red", lwd = 0.75)
#Normality Assumption
stdres.freq2 <- ggplot() +
  geom_density(mapping=aes(x=resid(my.sarima.model))) +
  xlab('Residuals') +
  ylab('Frequency') +
  ggtitle('Assumption Check:', subtitle = 'Normality') +
  geom_vline(xintercept = 0, col = "red", lwd = 0.75)

suppressMessages(grid.arrange(ad.resids_fit2, stdres.freq2, nrow=1))
```



2. Validate your predictions by performing a cross validation. Split the last 60 time periods in your data into a test set and use the remaining as a training set (note you'll have to split your $\mathbf{X}$ matrix too). Fit your best model to the training set and predict the values in the test set. Calculate RPMSE and coverage.

```
train_X <- X[1:804,]
test_X <- X[805:nrow(X),]

train_ts_set <- data$AnnAnom[1:804]
test_ts_set <- data$AnnAnom[805:nrow(data)]

train_ts <- ts(train_ts_set, start = c(1950,1), frequency = 12)
arima.train <- Arima(train_ts, order=c(2,0,1), seasonal=c(2,0,0), xreg=train_X)

forecast.obj <- forecast(arima.train, h=60, xreg=test_X, level=95)

#rpmse[cv] <- (test.set[['Visits']]-my.preds[,'Prediction'])^2 %>% mean() %>% sqrt()
RPMSE <- sqrt(mean((test_ts_set - forecast.obj$mean)^2))
stddev <- sd(data$AnnAnom)
#cvg[cv] <- ((test.set[['Visits']] > my.preds[,'lwr']) & (test.set[['Visits']] < my.preds[,'upr'])) %>%
cvg <- mean((test_ts_set > forecast.obj$lower) & (test_ts_set < forecast.obj$upper))

plot(forecast.obj)
```



The graph above shows predictions of the last 5 years of temperature anomalies. The root predictive mean square error, or RPMSE, is 0.0749319, which is less than the standard deviation of the model at 0.3436089. The coverage is 1.

Statistical Inference

1. Calculate the p-value for a test that $H_0 : \beta_2 = 0$ vs $H_A : \beta_2 > 0$. Also, calculate a 95% confidence interval for β_2.

```
a.matrix <- c(0, 0, 0, 0, 0, 0, 0, 1)
mytest <- glht(my.sarima.model, linfct = t(a.matrix), rhs=0, alternative='greater')
summary(mytest)
```

```
##
##   Simultaneous Tests for General Linear Hypotheses
##
## Fit: Arima(y = my.ts, order = c(2, 0, 1), seasonal = c(2, 0, 0), xreg = X)
##
## Linear Hypotheses:
##        Estimate Std. Error z value   Pr(>z)
## 1 <= 0 0.017258   0.002423   7.121 5.35e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Adjusted p values reported -- single-step method)
```

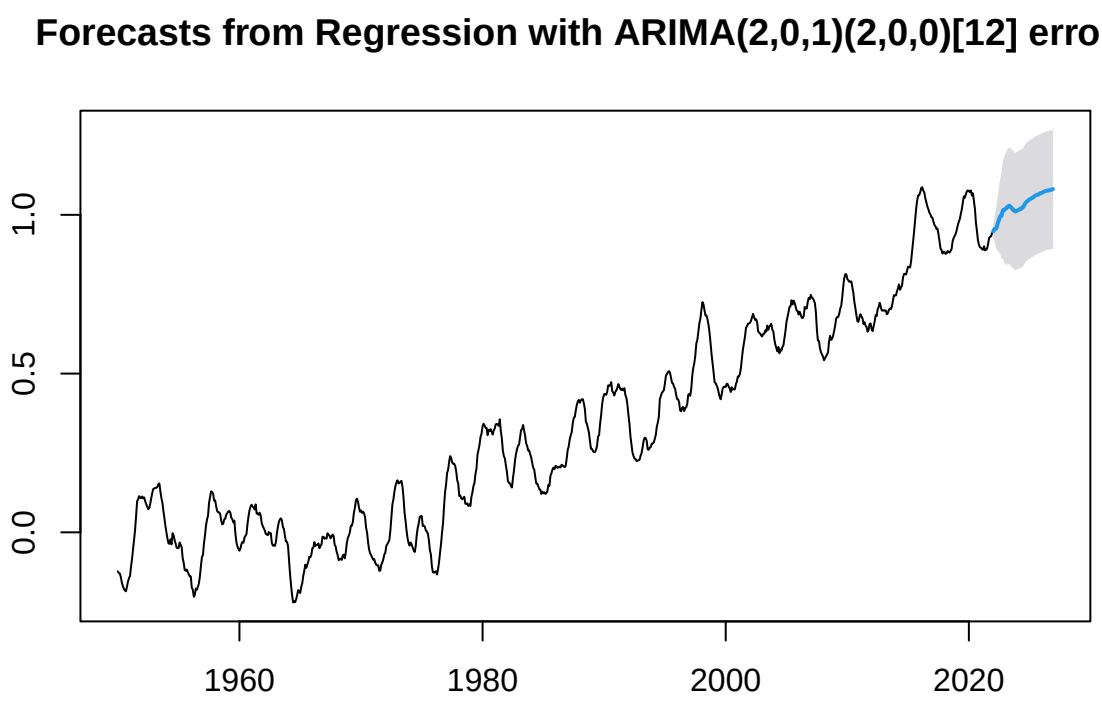
```
confint(my.sarima.model, level = 0.95) #beta 2 is pmax
```

```
##                                2.5 %       97.5 %
## ar1                     1.844908e+00  1.926811183
## ar2                    -9.344651e-01 -0.853056697
## ma1                    -5.393737e-01 -0.373763326
## sar1                   -7.941244e-01 -0.665671429
## sar2                   -4.004078e-01 -0.271734948
## intercept              -1.294110e+01  1.107941236
## YrMon                  -5.642463e-04  0.006586627
## pmax(YrMon - 1975, 0)  1.250785e-02  0.022007310
```

The p-value for a test that $H_0 : \beta_2 = 0$ is 0.000000000000535, so we reject the null hypothesis and conclude that there is evidence for $H_A : \beta_2 > 0$. We are 95% confident that the true value of β_2 is between 0.01250785 and 0.022007310.

2. Predict the temperature anomalies forward 60 months (5 years). To do this, you will have to set up an $\mathbf{X}$ matrix for the times you want to predict.

```
Xpreds <- tail(data$YrMon, 60) + 5
Xpreds <- cbind(Xpreds, pmax(Xpreds-1975,0))
forecast.obj2 <- forecast(my.sarima.model, h=60, xreg=Xpreds, level=0.95)
plot(forecast.obj2)
```



The graph above shows the predicted temperature anomalies forward 60 months (5 years). It shows a general upward trend of temperature anomalies.