STA 137 Final Project

12/01/2022

Introduction

Temperature anomalies is one of the most important measures of climate change, which gives a big picture of the average temperatures for the northern hemisphere compared to a reference value. Nowadays, the Earth is warming and the increase of global temperature may leads to more extreme weather events, such as longer fire seasons and more frequent floods. Thus, analyzing the data and forecasting the future temperature anomalies may help people understand how the temperatures for the northern hemisphere change.

The data we use in this project includes the annual temperature anomalies from 1850 to 2021 for the northern hemisphere. It is a time series since it is a series of data points listed in time order.

By analyzing the data, we want to figure out the following questions:

- How did the temperature anomalies from 1850 to 2021 for the northern hemisphere change? What's the trend of temperature anomalies?
- What will the temperature anomalies for the future 6 years be?

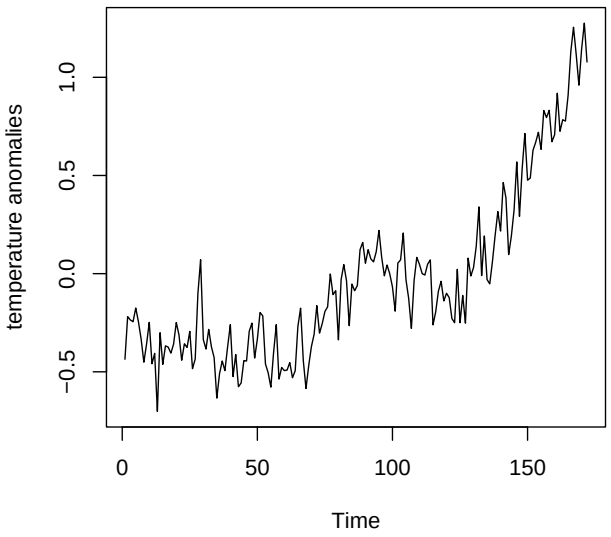
Materials and Methods

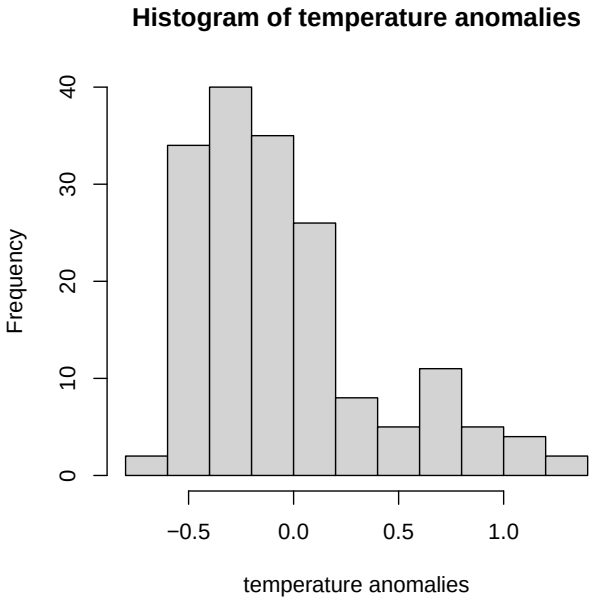
Description of the data

The data contains 172 observations on 2 variables, one is the the annual temperature anomalies for the northern hemisphere, one is the corresponding years from 1850 to 2021.

To get a preview of the data, we use graphical techniques to understand the nature of variation in the data and determine if the series is stationary or if we need to difference the series to achieve stationarity.

First, we get a glance of the raw data to check its stationarity.



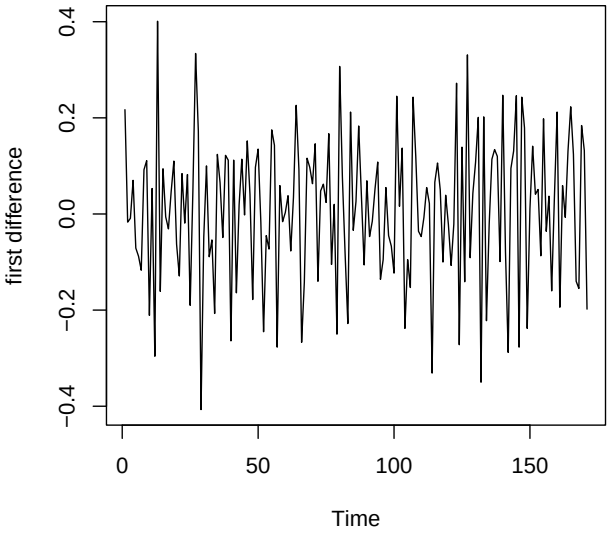


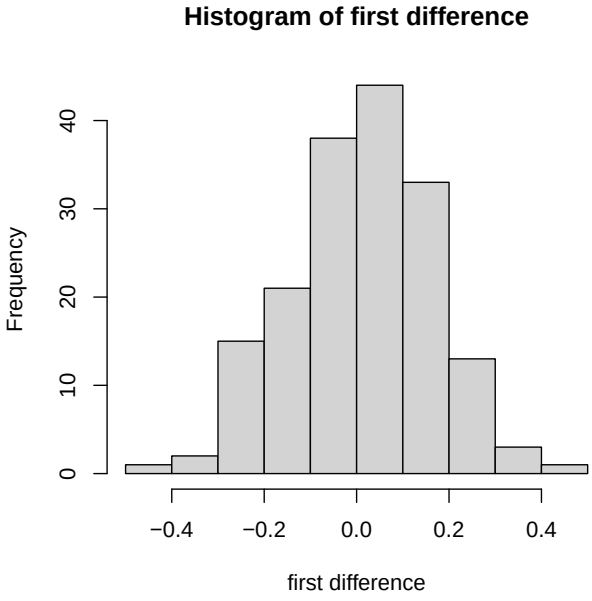
From its time series plot, we can see that its variation changes through the time and there is a clear trend, which indicates the raw data is not stationary and may not be a good choice for the modeling. The histogram indicates the distribution of the temperature anomalies is skewed, which tells us that the series needed to be processed before forecasting.

The "Ljung-Box" test presents that there is autocorrelations among the temperature anomalies data.

```
##
##  Box-Ljung test
##
## data:  y
## X-squared = 1096.9, df = 10, p-value < 2.2e-16
```

The results we have found led us to think about some changes to the raw data. Intuitively, we tried the first difference method and got an exciting result.



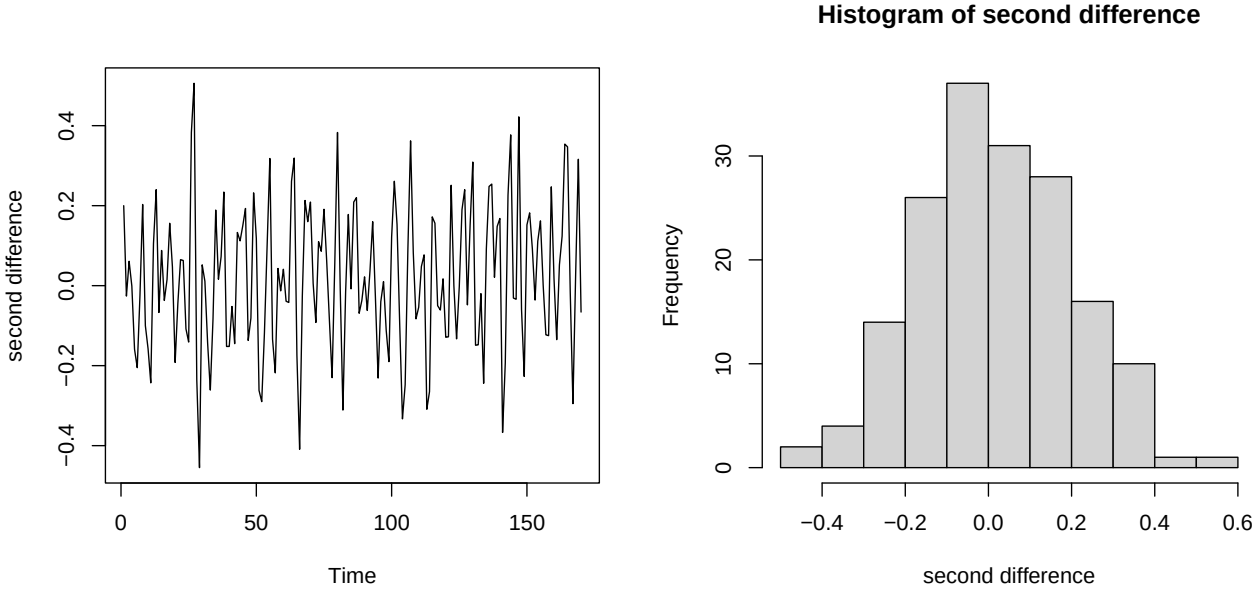


The plots above shows that the first difference of temperature anomalies should be stationary. By this stationary series, we can directly model the series without caring about the trend or seasonality problems, which makes the problem easier.

The "Ljung-Box" test tells us there are autocorrelations in this series, so that we need to build a models to capture these features.

```
##
##  Box-Ljung test
##
## data:  yd1
## X-squared = 28.619, df = 10, p-value = 0.001436
```

For a further check, we plotted the second difference of the series and found no bigger improvement than the first difference series. Therefore, we decided to use the first difference series as our target of modeling.



Methods

We mainly use two methods to model the time series and make forecasts. We start with modeling the first difference series, and we will compare it with the method of combining the trend and rough part to forecast the series later.

Method 1: ARIMA model Both ARMA model and ARIMA model are fitted to time series data either to better understand the data or to predict future points in the series (forecasting).[1]

ARMA model is the autoregressive moving average (ARMA) model for analyzing stationary time series data, which contains autoregressive terms (AR) and moving-average (MA) terms. [2]

An autoregressive integrated moving average (ARIMA) model is a generalization of an autoregressive moving average (ARMA) model. ARIMA models are for cases where an initial differencing step can be applied one or more times to eliminate the non-stationarity of the data.[1]

Since after applying the first difference, our time series data seems to be stationary, we use ARIMA model to fit the annual temperature anomalies from 1850 to 2015 and to forecast the temperature anomalies for the years 2016-2021.

Method 2: Spline model + ARMA model Instead of modeling the data directly, we use a cubic spline to model the trend of the annual temperature anomalies from 1850 to 2015, and use an ARMA model to model the corresponding rough part.

$$Yt = mt + Xt, \qquad t = 1, ..., n = 166$$

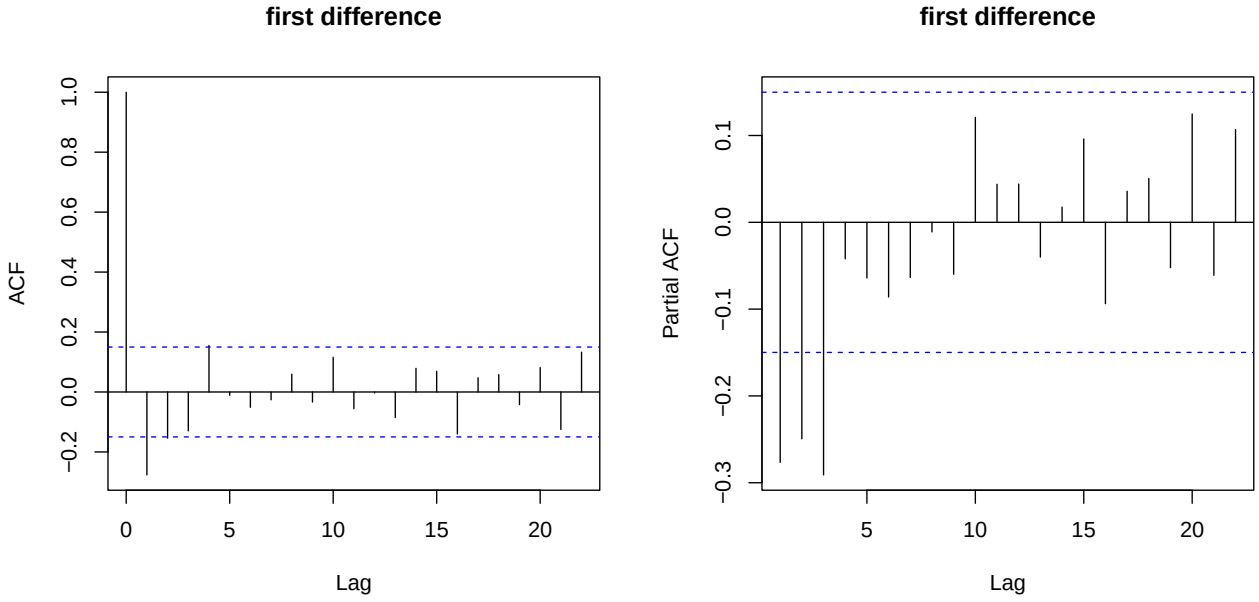
where Y_t is the observed annual temperature anomalies, m_t is the trend of data, and X_t is the rough of the data.

A a cubic spline is a locally cubic polynomial. Here, we use the R function "trend_spline" from Professor which starts with 9 equispaced knots resulting in 12 independent variables, and uses a backward stepwise method to obtain a spline estimate of the trend. [Handout 4] Then, to forecast the temperature anomalies for the years 2016-2021, we use the function ApproxExtrap in the Hmisc package to forecast the trend and use the fitted ARMA model to forecast the rough.

Results

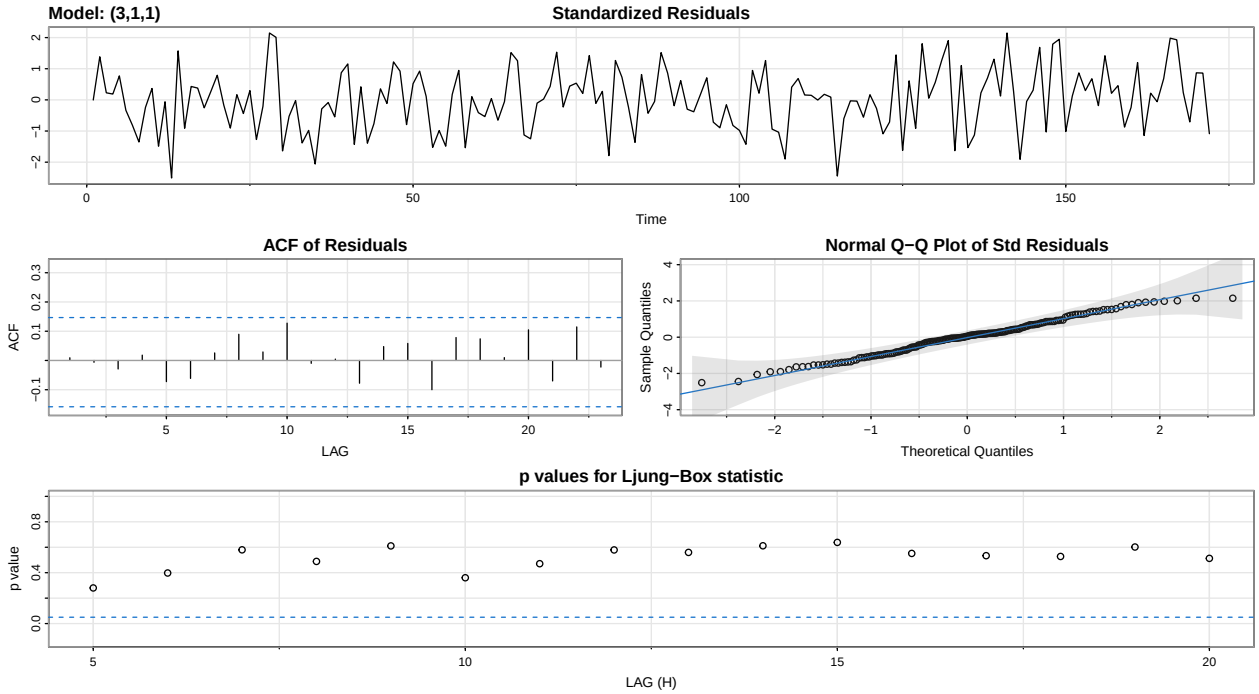
Method 1: ARIMA model

We started from using the ACF and PACF plots to determine a preliminary model. The ACF and PACF plots of the first difference series are as follows.

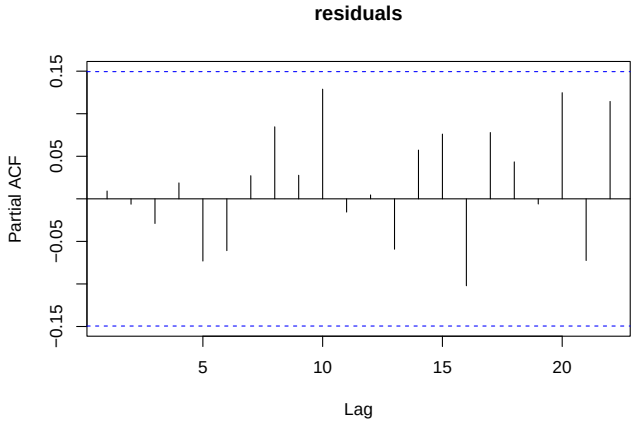


For data after applying first difference, the autocorrelation of lag 1 seems to be non-zero and the rest seems to be insignificant, and the partial autocorrelation of lag 1, 2, 3 seems to be non-zero and the rest seems to be insignificant. These findings suggest that we may model the data by ARIMA(3,1,1).

By fitting ARIMA(3,1,1), we got the following plots of its residuals. If the model is a good fit, the residuals should resemble white noise.



Except for the plots we got above, we also want to check the partial autocorrelations of the residuals.



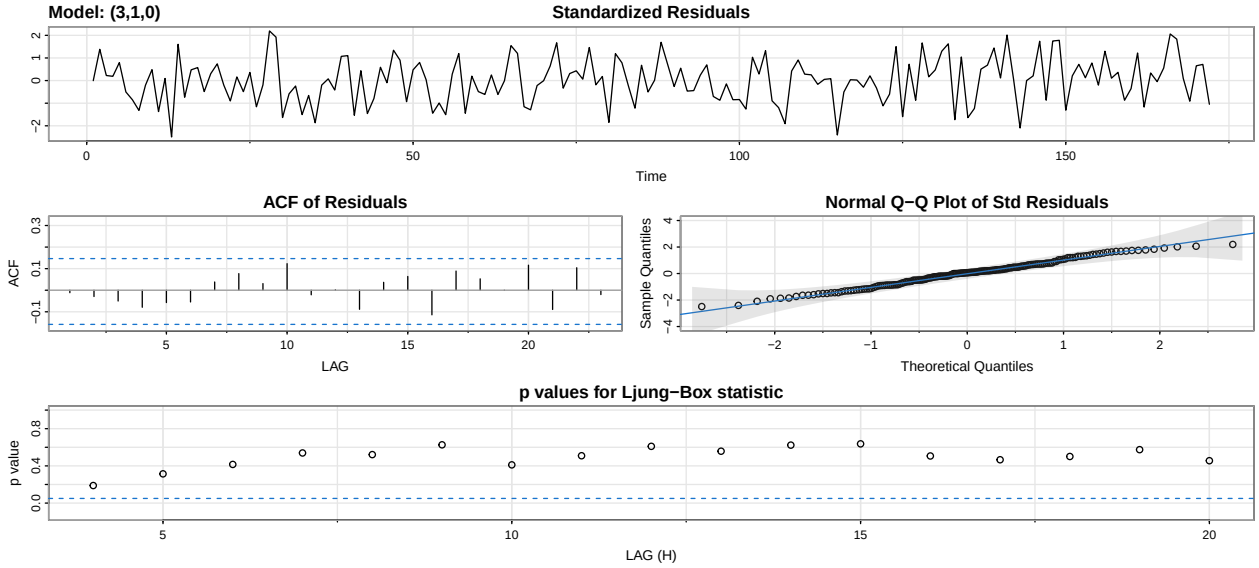
Based on the "Standardized Residuals" plot, the residuals looks like a white noise. The ACF shows the autocorrelations of all lags are not significant, which means that no serious correlations between residuals. The PACF shows the partial autocorrelations of all lags are not significant, which means that no serious partial correlations between residuals. The Q-Q plot presents an almost straight line, which suggests that the residuals is normal distributed. According to the Box-Ljung test that all p-values are bigger than 0.05, we fail to reject the null hypothesis, which means the residuals are identically independently distributed.All these results suggest that ARIMA(3,1,1) is a good fit for the series.

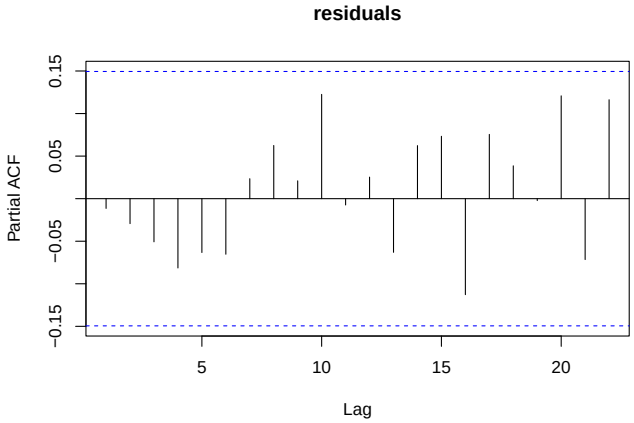
However, we want to find the best model by model selection criterion AIC. We use "sarima" function in R to fit the model and get the model selection criterion AIC.

Table 1: AIC table

	0	1	2	3
0	-0.9229414	-1.094005	-1.119878	-1.108348
1	-0.9922384	-1.114102	-1.108237	-1.109690
2	-1.0455561	-1.114930	-1.112556	-1.114580
3	-1.1233958	-1.116518	-1.110751	-1.103335

Based on smallest AIC from Table 1, we select ARIMA(3,1,0) as the final model. By fitting it to the first difference series, we got the plots as follows.





As what we have discussed above, if the model is a good fit, the residuals should resemble white noise. For this final model, we got the same conclusion with the former one.

- Based on the "Standardized Residuals" plot, the residuals seems a white noise.
- ACF and PACF plots suggest that there are no autocorrelations and partial autocorrelations among the residuals.
- The Q-Q plot shows that the residuals are normal distributed.

- The Box-Ljung test fails to reject the null hypothesis, which means that the residuals are identically independently distributed.

Thus, the fitted model ARIMA(3,1,0) can be a good fit for the first difference series, and is shown as follows.

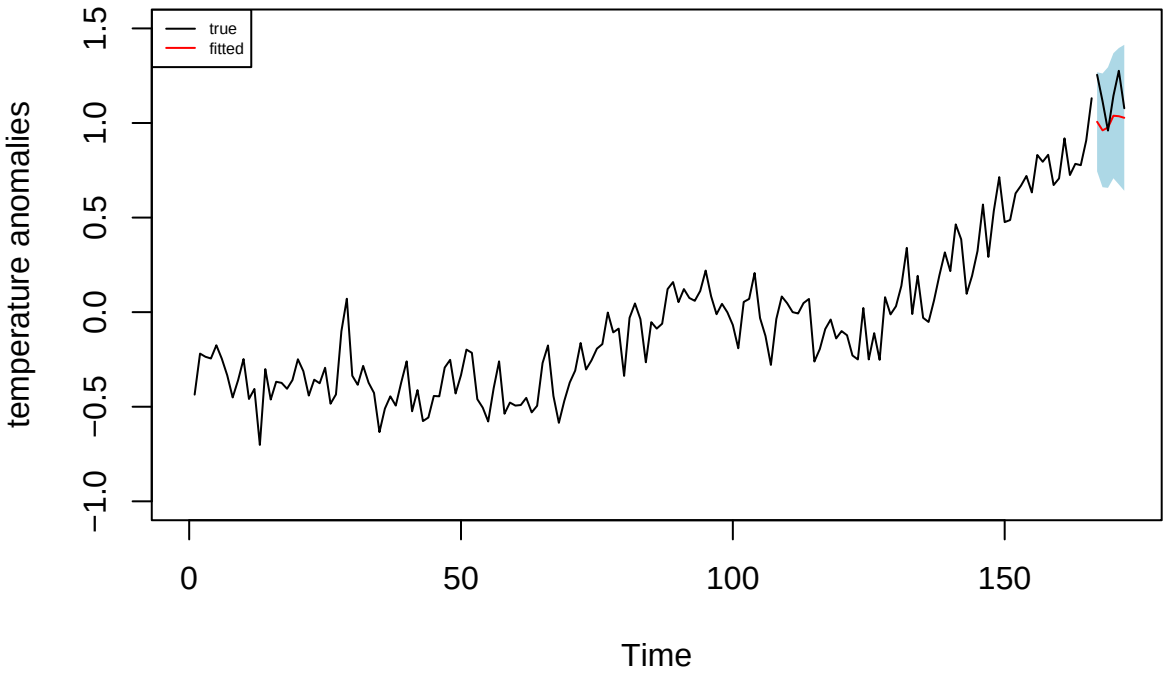
$$\nabla X_t = 0.0085 - 0.4234\nabla X_{t-1} - 0.3557\nabla X_{t-2} - 0.2947\nabla X_{t-3} + \epsilon_t,\ \nabla X_t = Y_t - Y_{t-1}$$

The parameters and their standard errors are shown in Table 2.

Table 2: Parameters and Standard Errors

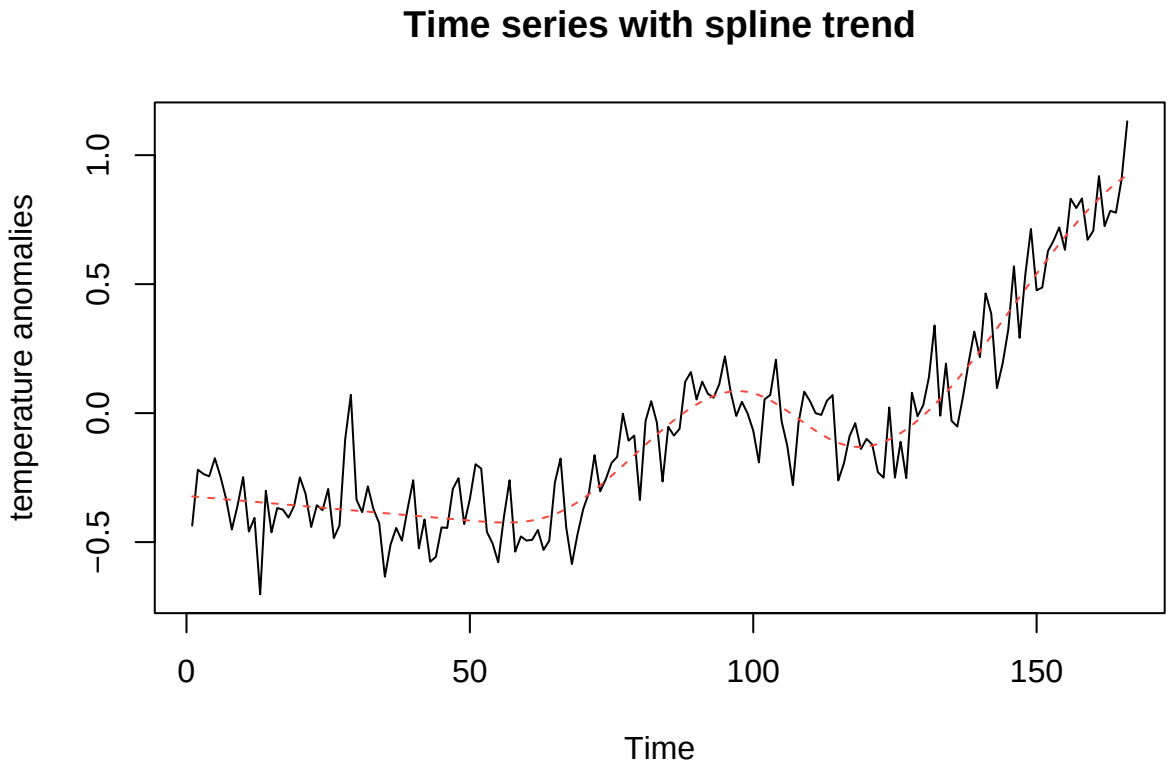
	Estimate	SE
ar1	-0.4234	0.0735
ar2	-0.3557	0.0756
ar3	-0.2947	0.0735
constant	0.0085	0.0050

In order to do forecasting, we refit the final model with all the data except for last 6 years, and use it to forecast the last 6 years' data. Finally, we got the plot below, where the red line represents our forecasting results, the blue shaded area represents the 95% confidence interval, and the black line is the real data. Although our forecasts are not exact the same with the real values, the real values are within the confidence interval, which is a good sign for forecasting.

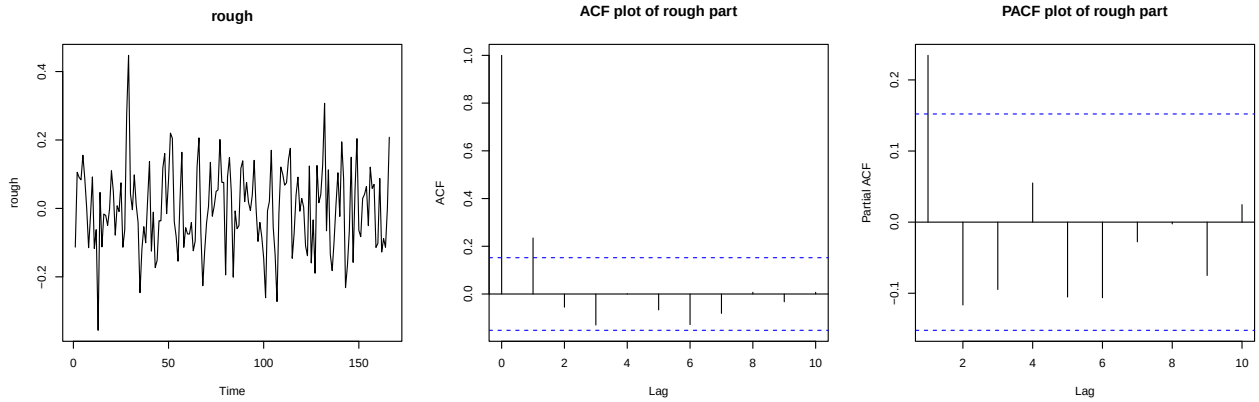


Method 2: Spline model + ARMA model

We first check whether the data need transformation. Since the data contains negative numbers, we shift the data so that the minimum of the data is 1. By the function trend_spline, we find that the optimal choice is not to do transformation, so we fit the trend by the original annual temperature anomalies from 1850 to 2015.



From trend plot, the spline model captures the trend of the time series data well. Now, let's take a look the the plots of the rough part.



From time series plot of the rough part, we can see that its variation seems to be stable, which indicates the rough data is stationary. Moreover, the autocorrelation of lag 1 seems to be non-zero and the rest seems to be insignificant. The partial autocorrelation of lag 1 seems to be non-zero and the rest seems to be insignificant. These findings suggest that we may use the ARMA model to fit the rough part.

Therefore, We again use sarima function in R to fit the ARMA model and get the model selection criterion AIC. Based on smallest AIC from Table 3, we choose the ARMA(2,1) model.

Table 3: AIC table

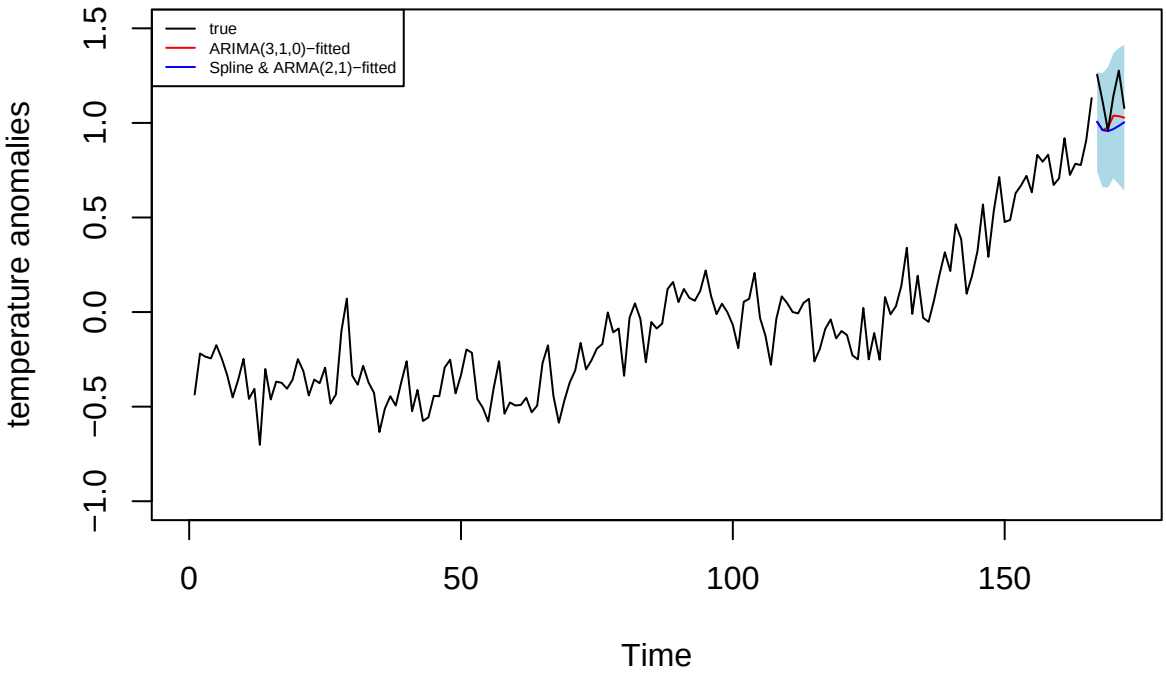
	0	1	2	3
0	-1.350245	-1.402853	-1.391080	-1.406964
1	-1.395766	-1.390922	-1.392361	-1.478807
2	-1.398327	-1.486497	-1.396170	-1.471198
3	-1.395839	-1.391304	-1.471875	-1.459815

After fitting the trend and rough, we want to forecast the temperature anomalies for the years 2016-2021. We use the function approxExtrap in the Hmisc package to forecast the trend and use the ARMA(2,1) model to forecast the rough, and we get the predicated temperature anomalies by adding the predicted trend and the predicted rough.

Comparison

We compare the real values with the forecasting results from two methods, ARIMA model and Spline model & ARMA model.

In the above plot, the red line represents the forecasts of ARIMA(3,1,0) model, the blue shaded area represents the 95% confidence interval of the ARIMA model prediction, the blue line represents the forecasts of spline model and ARMA(2,1) model, and the black line is the real data. The forecasts by two methods are quite close, but the forecasts by ARIMA(3,1,0) follows the trend of real data and performs better.



Conclusion and Discussion

We use two methods to model the annual temperature anomalies from year 1850 to 2015 and make forecasts of the temperature anomalies in 2016 to 2021. The first method is using ARIMA(3,1,0) model and the prediction of the temperature anomalies in 2016 to 2021 turns out to be within the 95% confidence interval, which means the forecasts seems to be reasonable. On the other hand, the second method is using a cubic

spline model to model the trend of the annual temperature anomalies with ARMA(2,1) model to model the rough part. The predictions of it also seem to be very close to the true values.

With a closer look of the last plot, we notice that the forecasts by ARIMA(3,1,0) capture more of the trend of the real data than the other method. However, overall, the predictions from both methods are all being reasonably close to the true value, which means two methods are both feasible for modeling and forecasting the time series of the annual temperature anomalies.

Reference

[1] Wikipedia contributors. (2022, October 15). Autoregressive integrated moving average. In Wikipedia, The Free Encyclopedia. Retrieved 02:21, December 5, 2022, from https://en.wikipedia.org/w/index.php?title=Autoregressive_integrated_moving_average&oldid=1116193509

[2] Shumway, Robert H. (2000). Time series analysis and its applications. David S. Stoffer. New York: Springer. p. 98. ISBN 0-387-98950-1. OCLC 42392178.

Session info

```
sessionInfo()
```

```
## R version 4.1.1 (2021-08-10)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Monterey 12.4
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1-arm64/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] Hmisc_4.8-0      ggplot2_3.4.1    Formula_1.2-4    survival_3.5-3
## [5] lattice_0.20-45  astsa_2.0        kableExtra_1.3.4 knitr_1.42
## [9] readxl_1.4.2
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.10         svglite_2.1.1       deldir_1.0-6
##  [4] png_0.1-8           digest_0.6.31       utf8_1.2.3
##  [7] R6_2.5.1            cellranger_1.1.0    backports_1.4.1
## [10] evaluate_0.20       highr_0.10          httr_1.4.4
## [13] pillar_1.8.1        rlang_1.0.6         data.table_1.14.8
## [16] rstudioapi_0.14     rpart_4.1.19        Matrix_1.4-0
## [19] checkmate_2.1.0     rmarkdown_2.20      splines_4.1.1
## [22] webshot_0.5.4       stringr_1.5.0       foreign_0.8-84
## [25] htmlwidgets_1.6.1   munsell_0.5.0       compiler_4.1.1
## [28] xfun_0.37           pkgconfig_2.0.3     systemfonts_1.0.4
```

```
## [31] base64enc_0.1-3     htmltools_0.5.4     nnet_7.3-18
## [34] tidyselect_1.2.0    tibble_3.1.8        gridExtra_2.3
## [37] htmlTable_2.4.1     fansi_1.0.4         viridisLite_0.4.1
## [40] dplyr_1.1.0         withr_2.5.0         grid_4.1.1
## [43] gtable_0.3.1        lifecycle_1.0.3     magrittr_2.0.3
## [46] scales_1.2.1        cli_3.6.0           stringi_1.7.12
## [49] latticeExtra_0.6-30 xml2_1.3.3          generics_0.1.3
## [52] vctrs_0.5.2         RColorBrewer_1.1-3  tools_4.1.1
## [55] interp_1.1-3        glue_1.6.2          jpeg_0.1-10
## [58] fastmap_1.1.0       yaml_2.3.7          colorspace_2.1-0
## [61] cluster_2.1.4       rvest_1.0.3
```

Code Appendix

```
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE)
knitr::opts_chunk$set(fig.pos = 'H')
library(readxl)
library(knitr)
library(kableExtra)
library(astsa)
library(Hmisc)
library(ggplot2)
library(astsa)
temp <- data.frame(read_excel('TempNH_1850_2021.xlsx'))
x <- temp$Year
y <- temp$Anomaly
par(mfrow=c(1,2))
plot.ts(y, ylab = 'temperature anomalies')
hist(y, main = 'Histogram of temperature anomalies', xlab = 'temperature anomalies')
par(mfrow=c(1,1))
Box.test(y, lag = 10, type = 'Ljung-Box')
yd1 <- diff(y,1)
par(mfrow=c(1,2))
plot.ts(yd1, ylab = 'first difference')
hist(yd1, xlab = 'first difference', main = 'Histogram of first difference')
par(mfrow=c(1,1))
Box.test(yd1, lag = 10, type = 'Ljung-Box')
yd2 <- diff(y,2)
par(mfrow=c(1,2))
plot.ts(yd2, ylab = 'second difference')
hist(yd2, xlab = 'second difference', main = 'Histogram of second difference')
par(mfrow=c(1,1))
par(mfrow=c(1,2))
acf(yd1, main = 'first difference')
pacf(yd1, main = 'first difference')
par(mfrow=c(1,1))
fit_pre <- sarima(y, p = 3, d = 1, q = 1) #arima(y, order = c(3,1,1))
pacf(fit_pre$fit$residuals, main = 'residuals')
# library(astsa)
AIC <- matrix(0, 4, 4)
for (i in 1:4){
```

```
  for (j in 1:4){
    AIC[i,j]<-sarima(y,p=i-1,d=1,q=j-1,details=FALSE)$AIC
  }
}
AIC <- data.frame(AIC)
rownames(AIC) <- c(0:3)
colnames(AIC) <- c(0:3)
kable(AIC, booktabs = TRUE, caption = "AIC table") %>%
  kable_classic_2(full_width = F,latex_options = "striped") %>%
  kable_styling(latex_options = "HOLD_position")

#which(AIC == min(AIC), arr.ind = TRUE)
#rownames(AIC)[which(AIC == min(AIC), arr.ind = TRUE)[1]]
#colnames(AIC)[which(AIC == min(AIC), arr.ind = TRUE)[2]]
fit_final2 <- sarima(y, p = 3, d = 1, q = 0)
#hist(fit_final2$fit$residuals, main = 'Histogram of residuals [Final]')
pacf(fit_final2$fit$residuals, main = 'residuals')
para_se <- data.frame(fit_final2$ttable[,1:2])
kable(para_se, booktabs = TRUE, caption = "Parameters and Standard Errors") %>%
  kable_classic_2(full_width = F,latex_options = "striped") %>%
  kable_styling(latex_options = "HOLD_position")
n <- length(y)
h <- 6
m <- n-h
ynew <- y[1:(n-h)]
ylast <- y[(n-h+1):n]
# fit
# Use "sarima.for" to forecast
fit_ref <- sarima.for(ynew, h, p = 3, d = 1, q = 0, plot = F) # prediction
upper <- fit_ref$pred+1.96*fit_ref$se
lower <- fit_ref$pred-1.96*fit_ref$se
# Use "arima"
# fit_ref <- arima(ynew, order = c(3,1,0)) # prediction
# fcast <- predict(fit_ref, n.ahead=h)
# upper <- fcast$pred+1.96*fcast$se
# lower <- fcast$pred-1.96*fcast$se
# plot
plot.ts(ynew, xlim=c(0,n), ylim=c(-1,1.5), xlab='Time', ylab = 'temperature anomalies')
polygon(x=c(m+1:h, m+h:1), y=c(upper, rev(lower)), col='lightblue', border=NA)
#lines(x=m+(1:h), y=fcast$pred, col='red')
lines(x=m+(1:h), y=fit_ref$pred, col='red')
lines(x=m+(1:h), y=ylast, col='black')
legend('topleft', legend = c('true', 'fitted'), cex=0.5, lty=c(1,1), col=c('black', 'red'))

trend_spline=function(y, lam) {
# Fits cubic spline estimate of trend
# If lam contains a single number, then the corresponding
# Box-Cox transformation is made, and a spline model is fitted
# If lam is a vector, then the best transformation is obtained from the
# candidates in 'lam', and then spline is
# fitted for the best transformation after
# deleting knots using backward stepwise regression
#Output,
```

```
# 1. transformed y: ytran (if lam is a vector, this corresponds to the
# best transformation)
#  2. trend: the fitted spline estimate
# 3. residual: the remainder, ie, ytran-trend
# 4. rsq, R^2 values for different transformations
# 5. lamopt: the best chosen transformation from lam
n=length(y);
p=length(lam)
rsq=rep(0, p)
y=sapply(y,as.numeric)
tm=seq(1/n, 1, by=1/n)
xx=cbind(tm, tm^2, tm^3)
knot=seq(.1, .9, by=.1)
m=length(knot)
for (j in 1:m) {
u=pmax(tm-knot[j], 0); u=u^3
xx=cbind(xx,u)
}
for (i in 1:p) {
if (lam[i]==0) {
ytran=log(y)
} else {
ytran=(y^lam[i]-1)/lam[i]
}
ft=lm(ytran~xx)
res=ft$resid; sse=sum(res^2)
ssto=(n-1)*var(ytran);
rsq[i]=1-sse/ssto
}
ii=which.max(rsq); lamopt=lam[ii]
if (lamopt==0) {
ytran=log(y)
} else {
ytran=y^lamopt
}
newdat=data.frame(cbind(ytran,xx))
ft=lm(ytran~.,data=newdat);
best_ft=step(ft, trace=0)
fit=best_ft$fitted; res=best_ft$resid
result=list(ytrans=ytran, fitted=fit, residual=res, rsq=rsq, lamopt=lamopt)
return(result)
}

temp$Time <- 1:nrow(temp)
temp$Anomaly_shift <- temp$Anomaly+abs(min(temp$Anomaly))+1
# get the data for fit
# ynew: temperature anomalies for the years 1850-2015
# s_time: year 1850-2015 represented with number 1-166
# ynew_shift: shifted temperature anomalies for the years 1850-2015
s_time <- temp$Time[1:(nrow(temp)-6)]
ynew_shift <- temp$Anomaly_shift[1:(nrow(temp)-6)]
# Fit spline
splinefit <- trend_spline(ynew_shift, lam = c(-1,-0.5,0,0.5,1))
```

```r
#splinefit$lamopt
# Get the trend of spline, i.e. fitted values of the function
splinefit <- trend_spline(ynew, lam = 1)
splinetrend <- splinefit$fitted
# Plot yt and the three trend estimates against time on the same graph
plot(s_time, ynew, type='l', lty=1, col=1, xlab='Time', ylab = 'temperature anomalies',
     main="Time series with spline trend")
points(s_time, splinetrend, type='l', lty=2, col=2)
legend(1,3e6, c("temp", "spline"), lty=c(1,2),
       col = c(1,2))
par(mfrow = c(1,3))
plot(s_time, splinefit$residual, type = 'l', xlab="Time", ylab="rough",
     main="rough")
# ACF plot
acf(splinefit$residual, lag.max = 10, main = 'ACF plot of rough part')
# PACF plot
pacf(splinefit$residual, lag.max = 10, main = 'PACF plot of rough part')
AIC <- matrix(0,4,4)
for (i in 1:4) {
for (j in 1:4) {
    AIC[i,j] <- sarima(splinefit$residual, p = i-1, d = 0, q = j-1, details = F)$AIC
  }
}
AIC <- data.frame(AIC)
rownames(AIC) <- c(0:3)
colnames(AIC) <- c(0:3)
kable(AIC, booktabs = TRUE, caption = "AIC table") %>%
  kable_classic_2(full_width = F,latex_options = "striped") %>%
  kable_styling(latex_options = "HOLD_position")

#which(AIC == min(AIC), arr.ind = TRUE)
#rownames(AIC)[which(AIC == min(AIC), arr.ind = TRUE)[1]]
#colnames(AIC)[which(AIC == min(AIC), arr.ind = TRUE)[2]]
# forcast the rough
fitrough <- sarima.for(splinefit$residual, p = 2, d = 0, q = 1, n.ahead = 6, plot = F)
#fitrough$pred

#data.frame(observed = xlast, forecast = fcast$pred)

#upper <- fcast$pred+1.96*fcast$se
#lower <- fcast$pred-1.96*fcast$se


# forcast the trend using function approxExtrap in the Hmisc package
# library(Hmisc)
fittrend <- approxExtrap(s_time, splinefit$fitted, xout=c(167:172))


# the forecasts of the anomalies for the years 2016-2021
fitpred <- fitrough$pred + fittrend$y
#fitpred
# plot
plot.ts(ynew, xlim=c(0,n), ylim=c(-1,1.5), xlab='Time', ylab = 'temperature anomalies')
```

```
# CI from ARIMA(3,1,0)
polygon(x=c(m+1:h, m+h:1), y=c(upper, rev(lower)), col='lightblue', border=NA)
lines(x=m+(1:h), y=fit_ref$pred, col='red')
lines(x=m+(1:h), y=fitpred, col='blue')
lines(x=m+(1:h), y=ylast, col='black')
legend('topleft', legend = c('true', 'ARIMA(3,1,0)-fitted', 'Spline & ARMA(2,1)-fitted'), cex=0.5, lty=
sessionInfo()
```