Final Project - Time Series Analysis of CA Temperature Data

Nikash Narula

12/3/2021

ABSTRACT:

The term "Global Warming" has always been a concept that I was familiar with, but I never got the chance to explore the reality of it myself. In this time series project, the main questions I addressed were: How drastically has the temperature in CA increased in the last century? Is there a visible pattern in the rise or fall of CA temperature? How much warmer is CA expected to get within the next 10 years? What are the environmental and social problems that are induced by higher temperatures? Employing the Box-Jenkins approach consisting of stationarity analysis, model identification, estimation and forecasting, I highlight the imminent global warming our state and world faces. Some of these key results include an average rise in temperature of over 3 degrees and an predicted mean temperature close to 60 degrees Fahrenheit in the coming years.

INTRODUCTION:

Living in California, it's often hard to pay close attention to the world's climate problems as we typically enjoy great weather year-round. However, global warming and climate change as a whole continues to be an immense problem our planet faces. The data being analyzed comes from the National Oceanic and Atmospheric Administration and details the annual average temperature (in Fahrenheit) of California for the past 90 years. It's crucial that scientists and data analysts be able to accurately predict and forecast weather data to better prepare humans for extreme climate conditions that are linked to health complications, damage to agriculture and water supply. Results of the forecasted weather data show an increase of close to 1 whole degree hotter on average in just the next 10 years. In this report, the Box-Jenkins methodology is applied and includes techniques such as ADF trend analysis, transforms, differencing, ACF/PACF analysis, ARIMA modeling and forecasting. All code is generated using R.

SECTIONS:

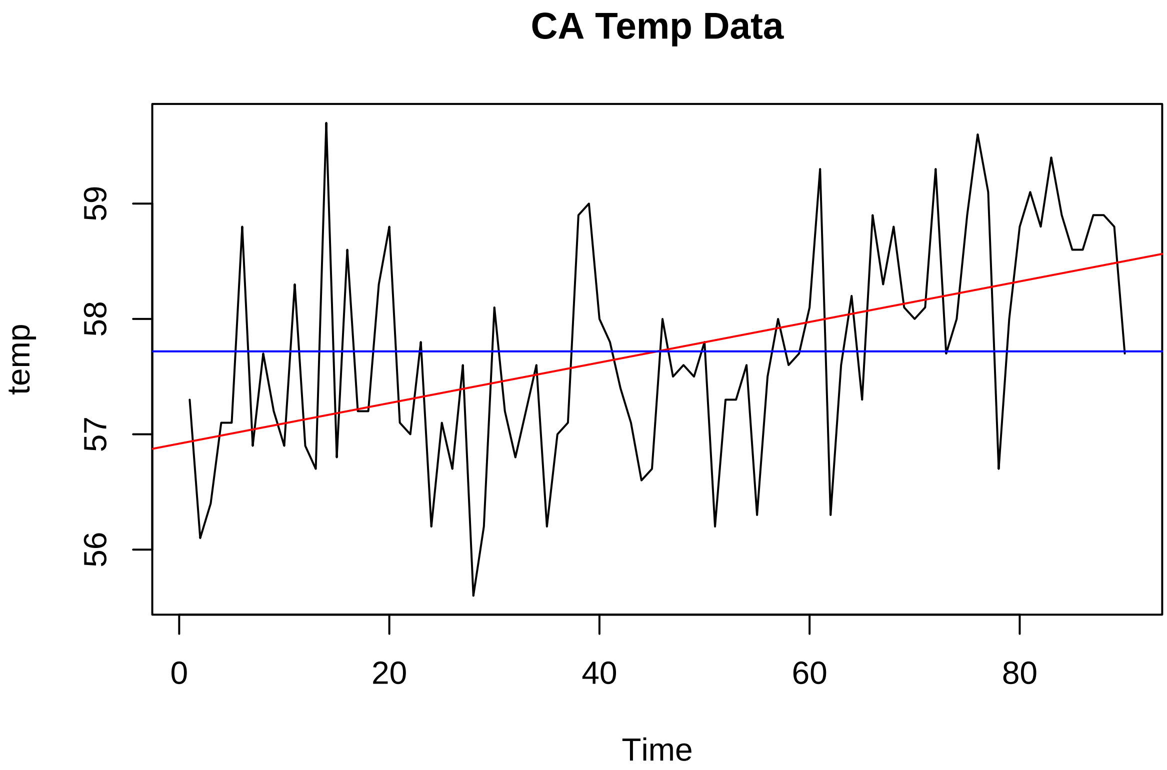
I: **Plot and Analyze**

The initial plot of the data suggests a pretty stable variance, no apparent seasonality, and a strong positive linear trend. There are also no sudden sharp changes in behavior. The mean of the data is 57.7 degrees Fahrenheit. The histogram is slightly right-skewed, but overall symmetric. The ACF decays slowly which could signify a nonstationary series.

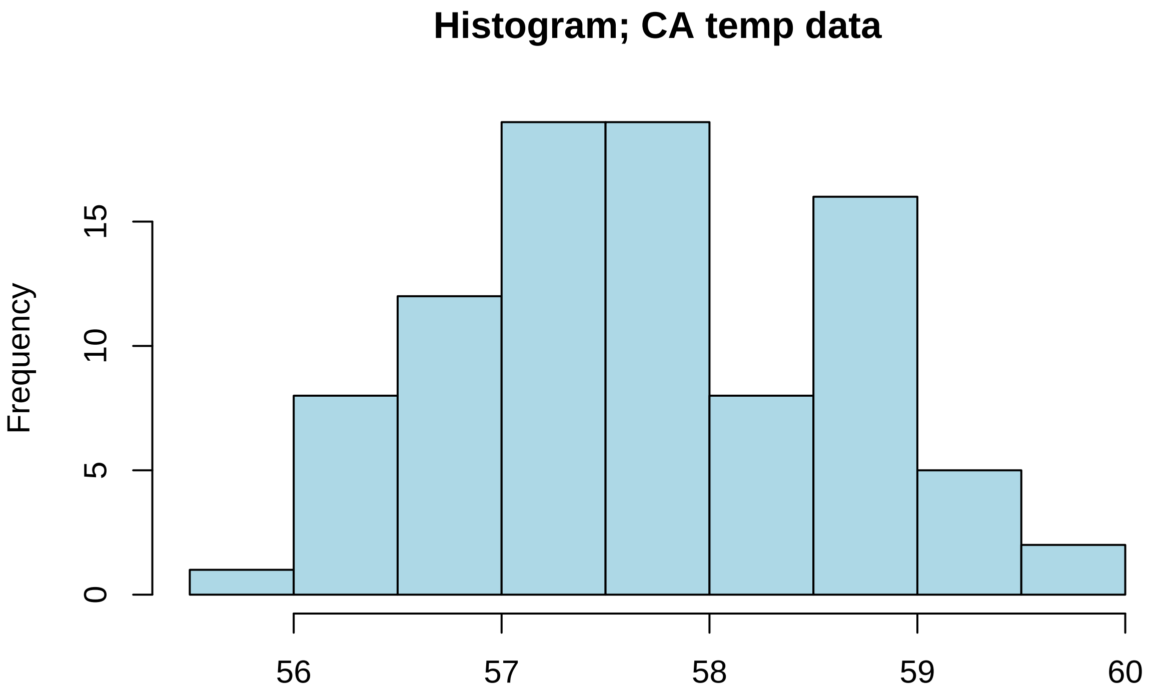
```r
# load the Data
temp_data = read.csv("CATemp.csv")
temp = as.numeric(temp_data$Average.Temperature[4:93])
temp.test = as.numeric(temp_data$Average.Temperature[94:103]) # leave 10 points for model validation
plot.ts(temp, main="CA Temp Data") # stable variance, no apparent seasonality, linear trend
nt = length(temp)
fit = lm(temp ~ as.numeric(1:nt)); abline(fit, col="red")
mean(temp) # 57.71889
```

```
## [1] 57.71889
```

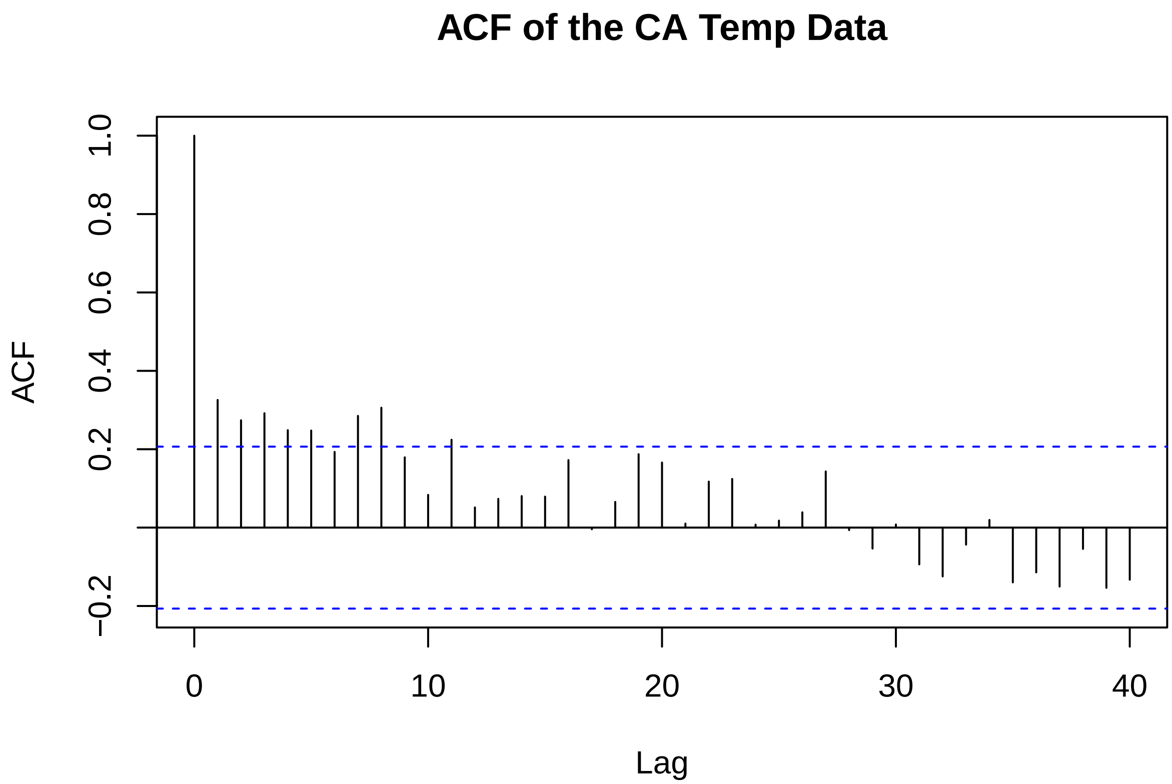
```r
abline(h=mean(temp), col="blue")
```




```r
hist(temp, col="light blue", xlab="", main="Histogram; CA temp data") # slightly skewed right, but some
```



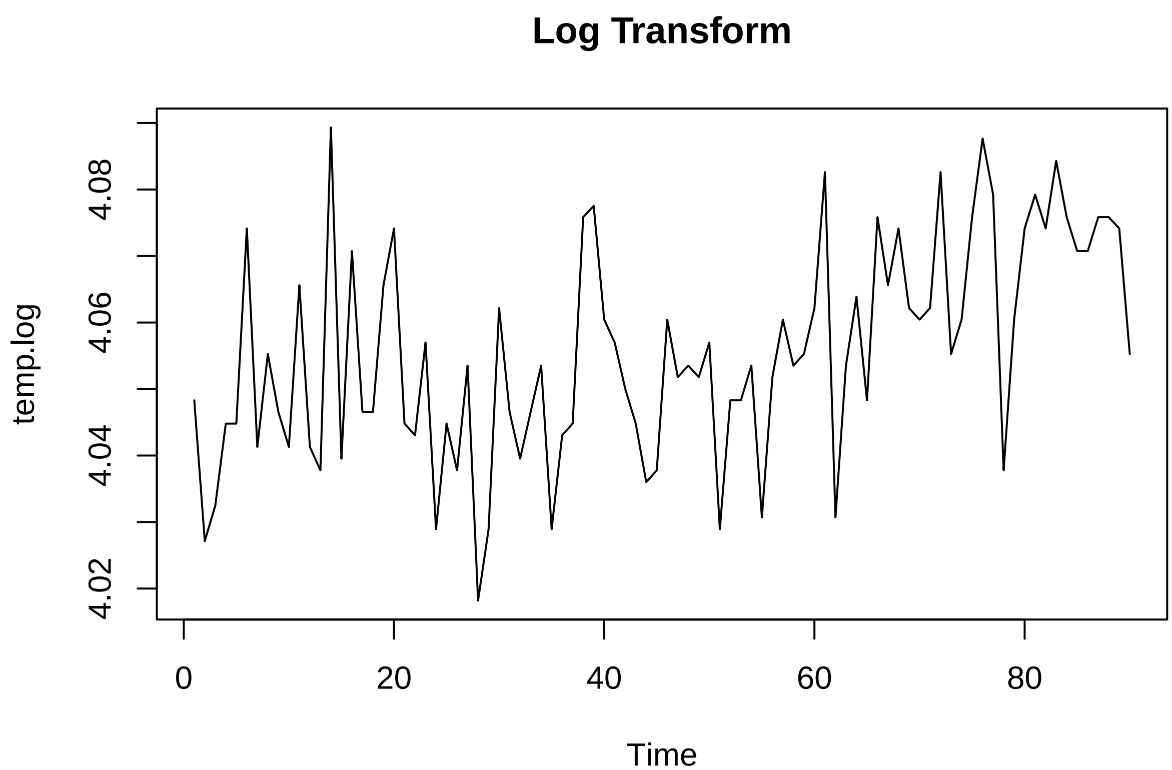

```r
acf(temp,lag.max=40, main="ACF of the CA Temp Data") # outside at lags 1,2,3,4,5,7,8, maybe 11
```



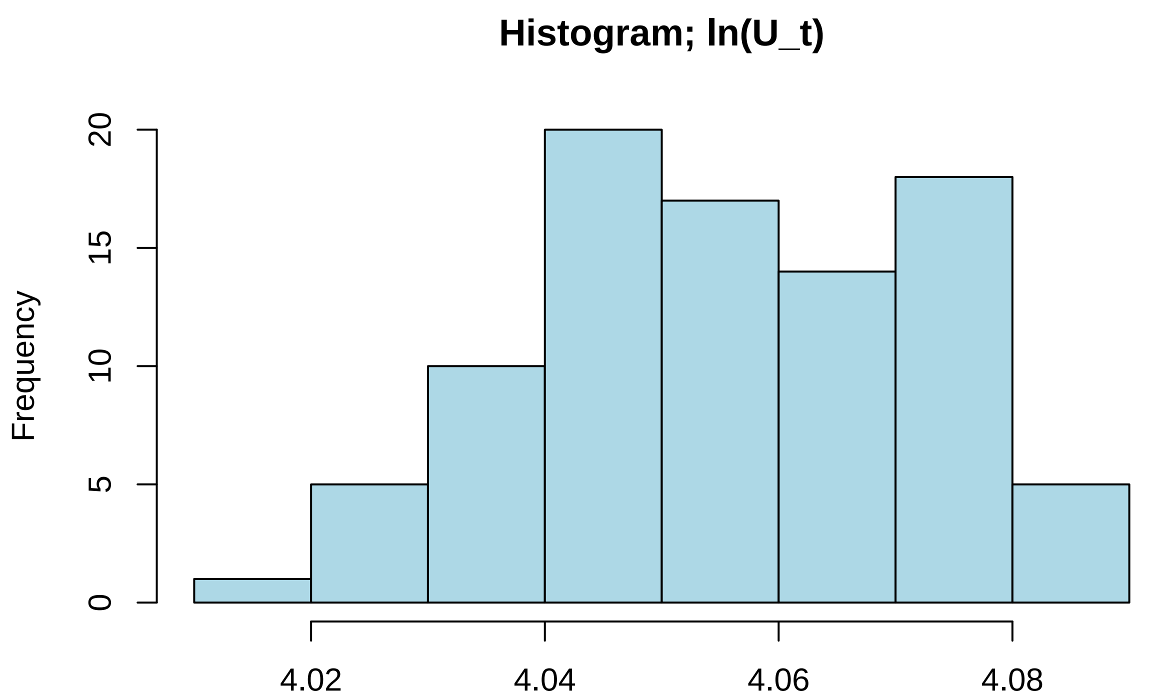
II: **Transformations/Differencing**

As the initial plot of the data displayed a stable variance, it does not seem likely that a variance stabilization transform is needed, however, it is still good to prove so. Plots of the both the log and Box-cox transformations show little to no improvement in overall variance. Their corresponding histograms also become more skewed - thus, no variance transformation is needed. To eliminate the linear trend, a differencing technique is used. Differencing at lag 1, the trend is eliminated and the plot starts to look much more stationary. The histogram of the differenced data looks symmetric and almost Gaussian. Further, the stationarity is confirmed using the Augmented Dickey-Fuller (ADF) Test which tests the null hypothesis that a unit root is present in a time series sample. For p-values < 0.05, one can reject the null hypothesis and conclude that series is stationary. Thus, with a p-value of 0.09 for non-differenced data and 0.01 for the differenced data, stationarity is suggested for the data differenced at lag 1. Lastly, differencing again at lag 1 leads to a higher variance, which implies overfitting - just 1 difference at lag 1 is the better option.

```
# Try log or BC transform to improve variance, although variance already looks stable
# log transform
temp.log = log(temp)
plot.ts(temp.log, main="Log Transform")
```



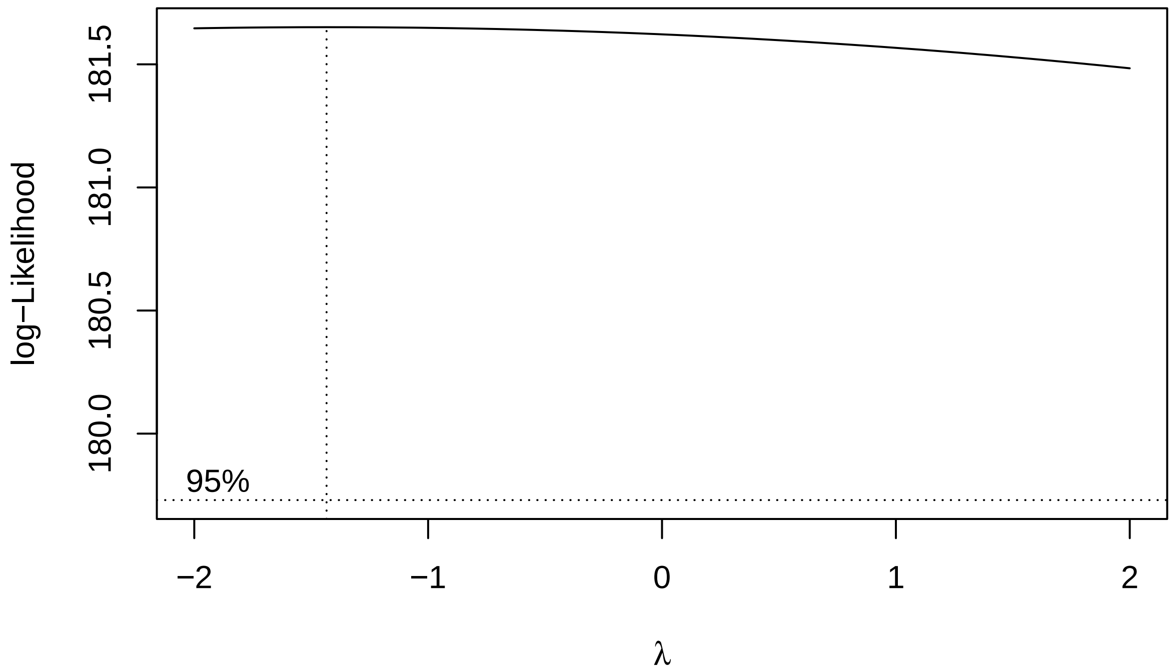

```
hist(temp.log, col="light blue", xlab="", main="Histogram; ln(U_t)")
```




```
# BC transform
library("MASS")
```

```
## Warning: package 'MASS' was built under R version 4.1.2
```

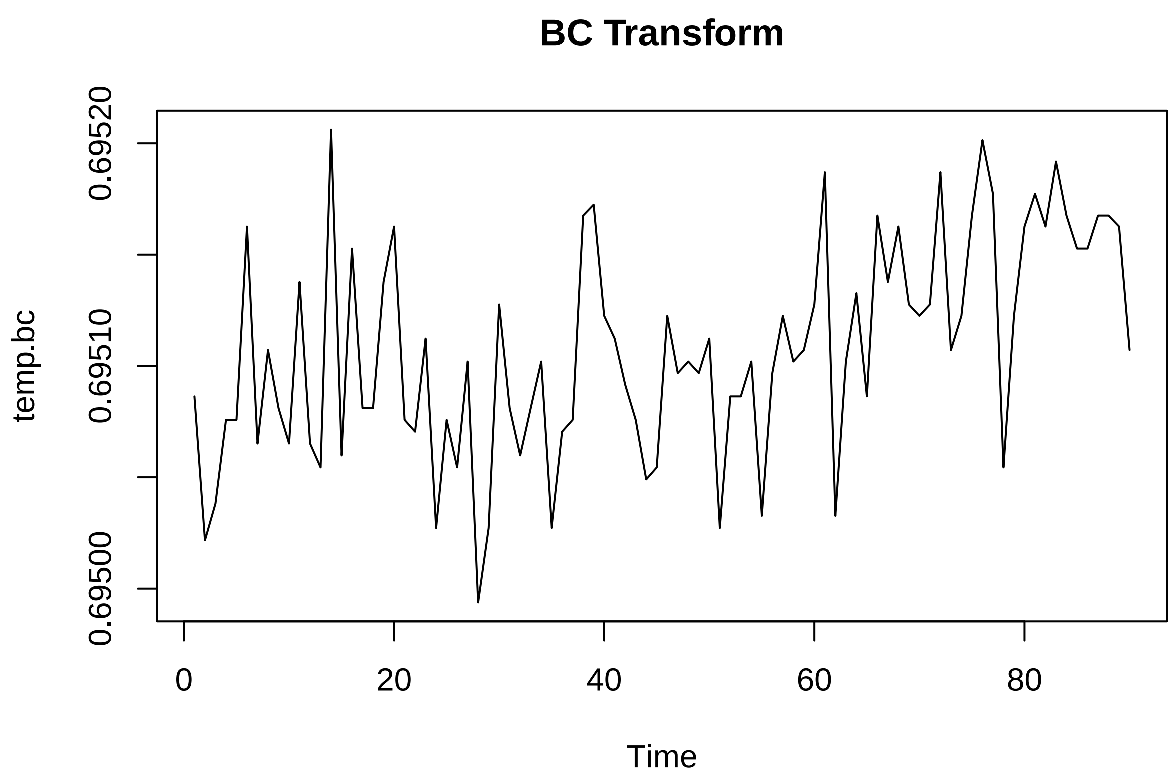
```
bcTransform = boxcox(temp ~ as.numeric(1:length(temp)))
```



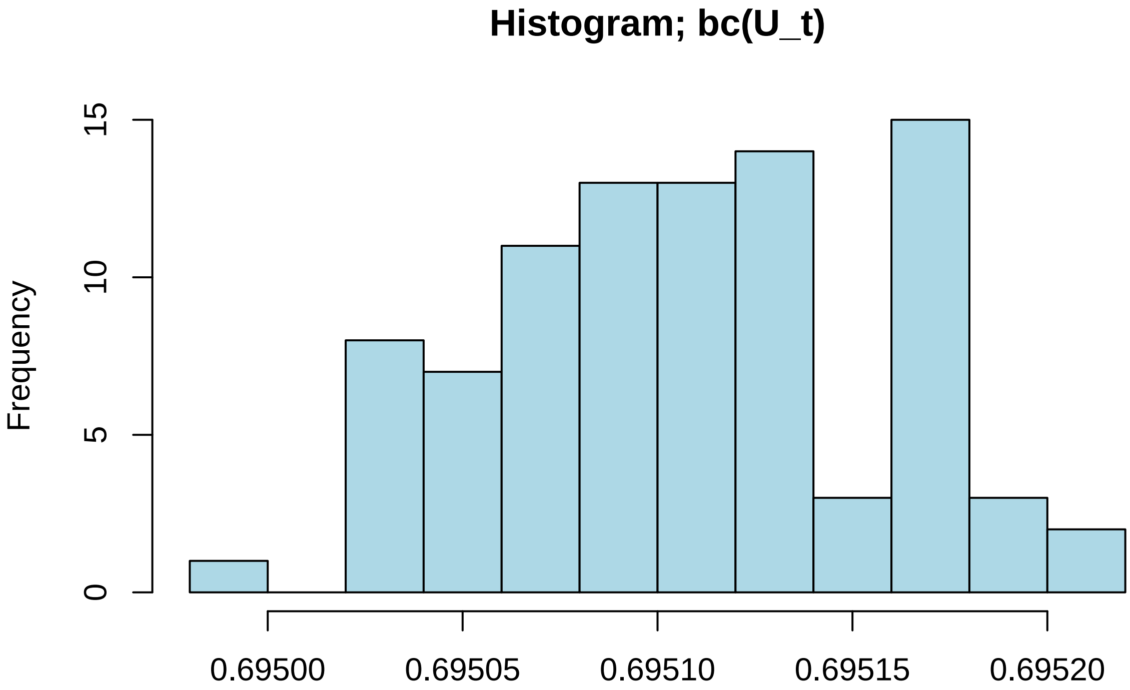

```
lambda = bcTransform$x[which(bcTransform$y == max(bcTransform$y))]
lambda # -1.43
```

```
## [1] -1.434343
```

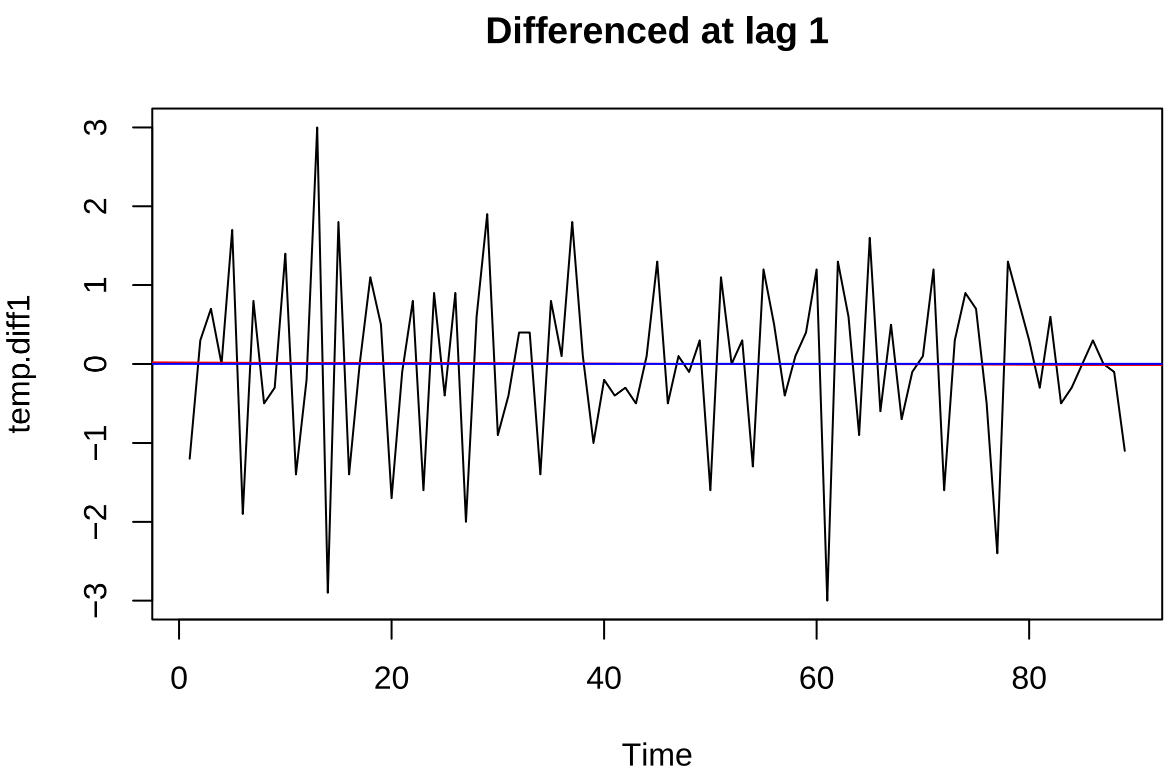
```
temp.bc = (1/lambda)*(temp^lambda-1)
plot.ts(temp.bc, main="BC Transform") # slightly less variant
```



```r
hist(temp.bc, col="light blue", xlab="", main="Histogram; bc(U_t)")
```




```r
# Not much change for either, histograms become more skewed and imply no transformation
# is necessary.

# Try differencing to remove linear trend.
temp.diff1 = diff(temp, lag=1)
plot.ts(temp.diff1, main="Differenced at lag 1")
fit_diff = lm(temp.diff1 ~ as.numeric(1:length(temp.diff1))); abline(fit_diff, col="red") # differencin
abline(h=mean(temp.diff1), col="blue")
```




```
hist(temp.diff1, density=20, breaks=20, col="blue", xlab="", prob=TRUE, main="Differenced at lag 1")
m = mean(temp.diff1)
std = sqrt(var(temp.diff1))
curve(dnorm(x,m,std), add=TRUE)
var(temp) # 0.87301
```

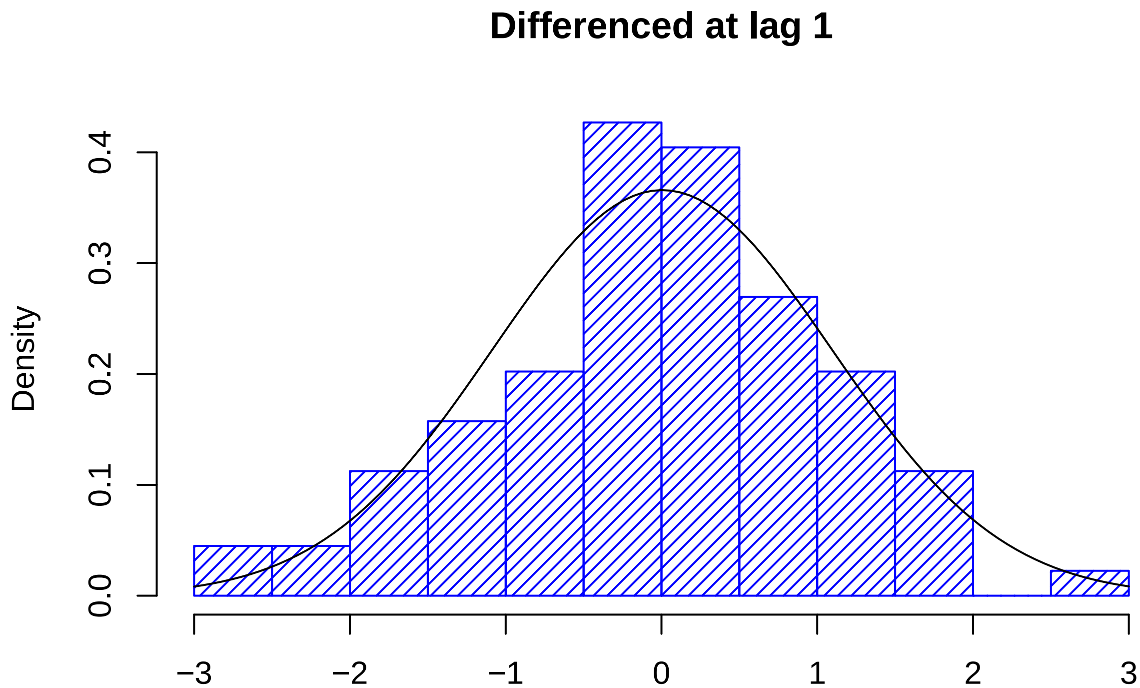
```
## [1] 0.87301
```

```
temp.diff11 = diff(temp.diff1, lag=1) # difference again
var(temp.diff11) # 3.49 = overfitting
```

```
## [1] 3.496895
```

```
library(tseries) # perform ADF test for unit root/stationarity
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```




```
adf.test(temp) # p-value of 0.09 = not stationary
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  temp
## Dickey-Fuller = -3.1878, Lag order = 4, p-value = 0.09488
## alternative hypothesis: stationary
```

```
adf.test(temp.diff1) # p-value of 0.01 = stationary!
```

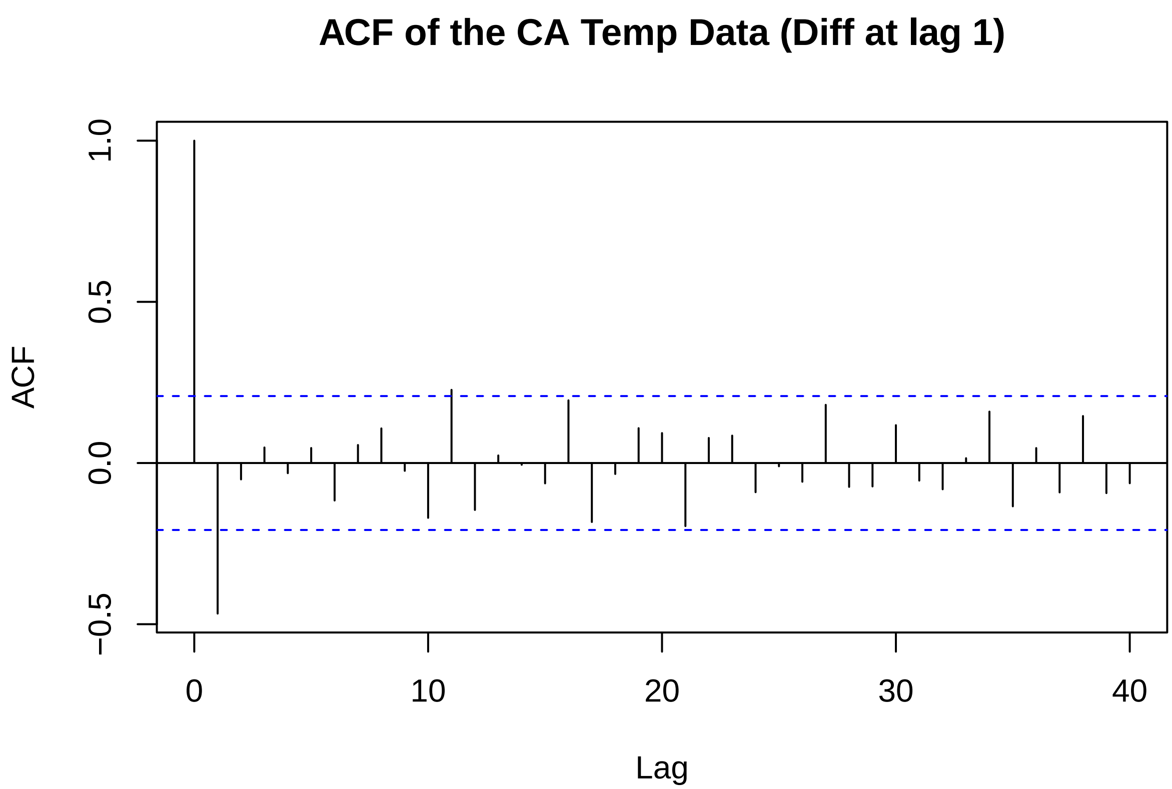
```
## Warning in adf.test(temp.diff1): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  temp.diff1
## Dickey-Fuller = -6.376, Lag order = 4, p-value = 0.01
## alternative hypothesis: stationary
```

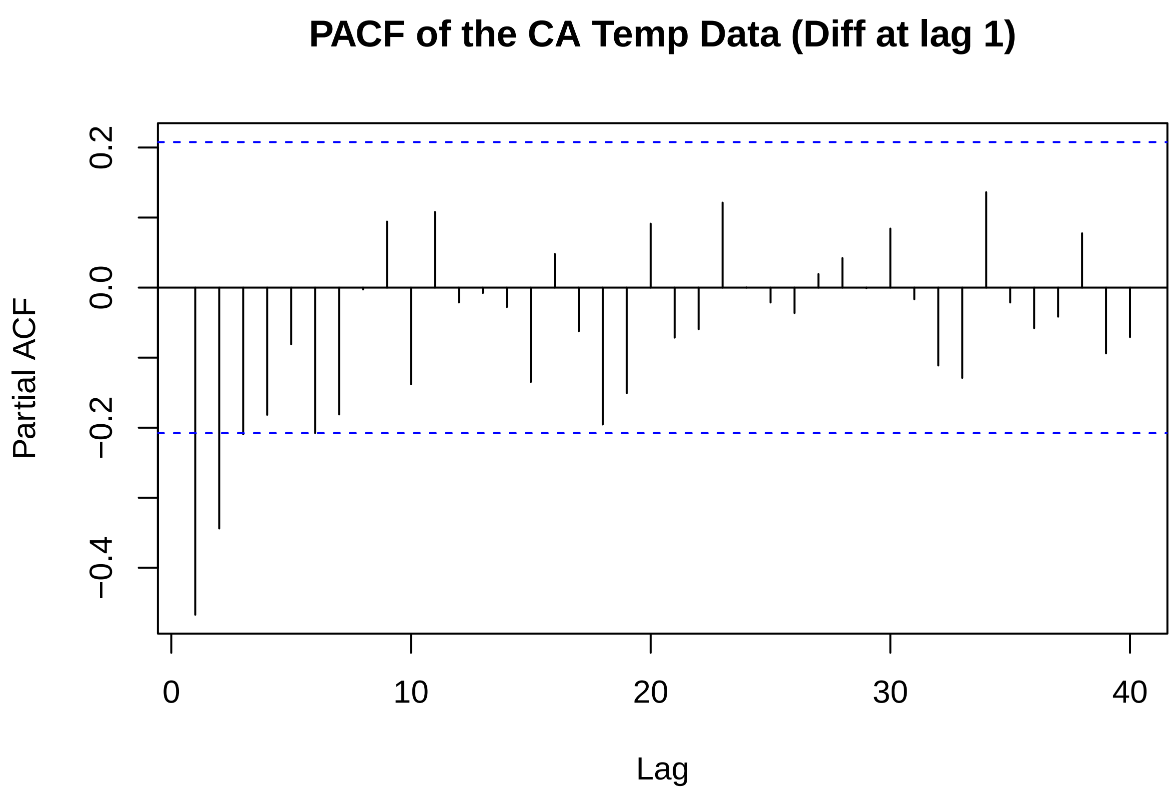
III: **Model Identification with ACF/PACF**

Plotting the ACF and PACF of the new differenced data, it's clear that the ACF is now truncated after lag 1 and fast-decaying, implying stationarity. The PACF is truncated after lag 2. From these graphs, 3 proposed models are: ARIMA(2,1,1) or ARIMA(2,1,0) or ARIMA(0,1,1).

```
acf(temp.diff1,lag.max=40, main="ACF of the CA Temp Data (Diff at lag 1)") # ACF outside at lags 1
```




```
pacf(temp.diff1,lag.max=40, main="PACF of the CA Temp Data (Diff at lag 1)") # PACF outside at lag 1 an
```




```
# Proposed models to try: ARIMA(2,1,1) or ARIMA(2,1,0) or ARIMA(0,1,1) --> look at lowest AIC
```

IV: **Fitting the Model**

The 3 proposed models from best to worst (based on AIC values) are ARIMA(0,1,1), ARIMA(2,1,1), ARIMA(2,1,0). Their corresponding AIC values and variances are 224.08 ($\sigma^2 = 0.6843$) , 227.9 ($\sigma^2 = 0.6829$) and 238.78 ($\sigma^2 = 0.7959$). The 2 best models based on AIC are ARIMA(0,1,1) and ARIMA(2,1,1). Coefficient estimates for ARIMA(0,1,1) are $\theta_1 = -0.8479$, while estimates for ARIMA(2,1,1) are $\phi_1 = -0.0154$, $\phi_2 = -0.0535$, and $\theta_1 = -0.8298$. Going forward, I will refer to **model 1 = ARIMA(0,1,1)** and **model 2 = ARIMA(2,1,1)**. For model 1, this is clearly stationary as it is a moving average model with no AR coefficients. It is also invertible as $|\theta_1| < 1$. For model 2, stationarity can be checked by evaluating if the roots of $\phi(z)$ lie outside the unit circle. Invertibility can be checked by evaluating if the roots of $\theta(z)$ lie outside the unit circle. The corresponding roots for $\phi(z)$ are 4.181847 and -4.469697, and 1.20511 for $\theta(z)$. Thus, both chosen models are stationary and invertible. Now, diagnostic checking is performed.

Model 1's histogram of residuals appears symmetric and normal. There is also no visible trend or change of variance, and the QQ-plot of residuals is indicated as normal. All ACF and PACF of the residuals are within the confidence intervals and can be counted as 0. The Shapiro-Wilk test for normality yields a p-value of $0.6976 > 0.05$. Other Portmanteau tests such as the Box-Pierce Test (p-value = 0.8157), Ljung-Box Test (p-value = 0.7582) and Mcleod-Li Test (p-value = 0.5272) also return p-values > 0.05. Therefore, the null hypotheses of these tests fail to be rejected and it is evident that the model does not show a lack of fit. The model is also fitted to AR of order 0 ~ White Noise. Thus, from analysis of the residuals we can conclude that this model passes diagnostic checking.

Similarly, model 2's histogram of residuals appears symmetric and normal. There is also no visible trend or change of variance, and the QQ-plot of residuals is indicated as normal. All ACF and PACF of the residuals are within the confidence intervals and can be counted as 0. The Shapiro-Wilk test for normality yields a p-value of $0.6067 > 0.05$. Other Portmanteau tests such as the Box-Pierce Test (p-value = 0.6896), Ljung-Box Test (p-value = 0.6205) and Mcleod-Li Test (p-value = 0.4418) also return p-values > 0.05. Therefore, the null hypotheses of these tests fail to be rejected and it is evident that the model does not show a lack of fit. The model is also fitted to AR of order 0 ~ White Noise. Thus, from analysis of the residuals we can conclude that this model also passes diagnostic checking.

Diagnostic checking of both models ARIMA(0,1,1) and ARIMA(2,1,1) show that either model is satisfactory for the given data. However, as stated earlier, the AIC of ARIMA(0,1,1) is 224.08 while the AIC of ARIMA(2,1,1) is 227.9, implying that ARIMA(0,1,1) is the better model as suggested in the initial ACF/PACF graphs. It's also important to note that because the AR estimates $\phi_1 = -0.0154$ and $\phi_2 = -0.0535$ are so close to 0, these estimates can be fixed to 0 so that model 2 also becomes an ARIMA(0,1,1). Still, if not fixed, the Principle of Parsimony says to choose the model with the fewest parameters which in this case is model 1.

Final Model - ARIMA(0,1,1): $(1 - B)X_t = \theta(B)Z_t$

$X_t - X_{t-1} = Z_t - 0.8479Z_{t-1}$

```r
arima(temp, order=c(2,1,1), method="ML") # SECOND best model
```

```
##
## Call:
## arima(x = temp, order = c(2, 1, 1), method = "ML")
##
## Coefficients:
##           ar1      ar2      ma1
##       -0.0154  -0.0535  -0.8298
## s.e.   0.1364   0.1273   0.0896
##
## sigma^2 estimated as 0.6829:  log likelihood = -109.95,  aic = 227.9
```

```r
arima(temp, order=c(2,1,0), method="ML")
```

```
##
## Call:
## arima(x = temp, order = c(2, 1, 0), method = "ML")
```

```
##
## Coefficients:
##           ar1      ar2
##       -0.6447  -0.3549
## s.e.   0.1000   0.0991
##
## sigma^2 estimated as 0.7959:  log likelihood = -116.39,  aic = 238.78
```

```
arima(temp, order=c(0,1,1), method="ML") # BEST model
```

```
##
## Call:
## arima(x = temp, order = c(0, 1, 1), method = "ML")
##
## Coefficients:
##           ma1
##       -0.8479
## s.e.   0.0577
##
## sigma^2 estimated as 0.6843:  log likelihood = -110.04,  aic = 224.08
```

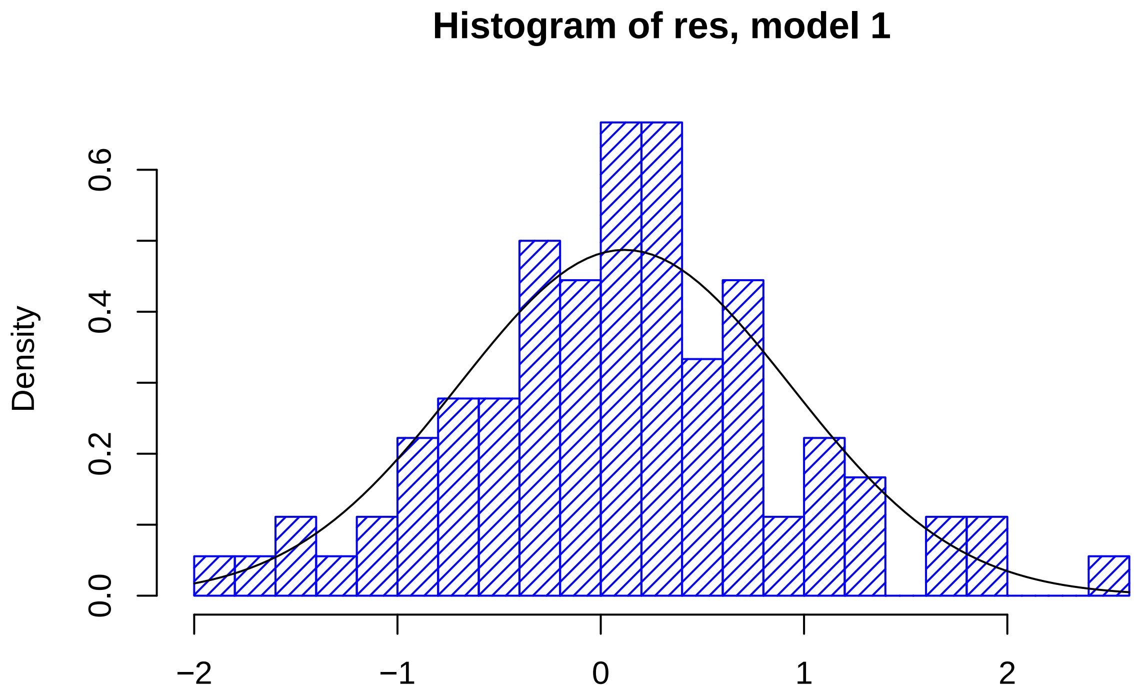
```
# check stationarity/invertibility of the best models - stationary is phi(z) outside unit cir, # invert
# model 1: ARIMA(0,1,1)
# stationary because this is a moving average process
# invertible because |theta1| < 1

# model 2: ARIMA(2,1,1)
# stationarity
polyroot(c(1,-0.0154,-0.0535))
```

```
## [1]  4.181847+0i -4.469697-0i
```

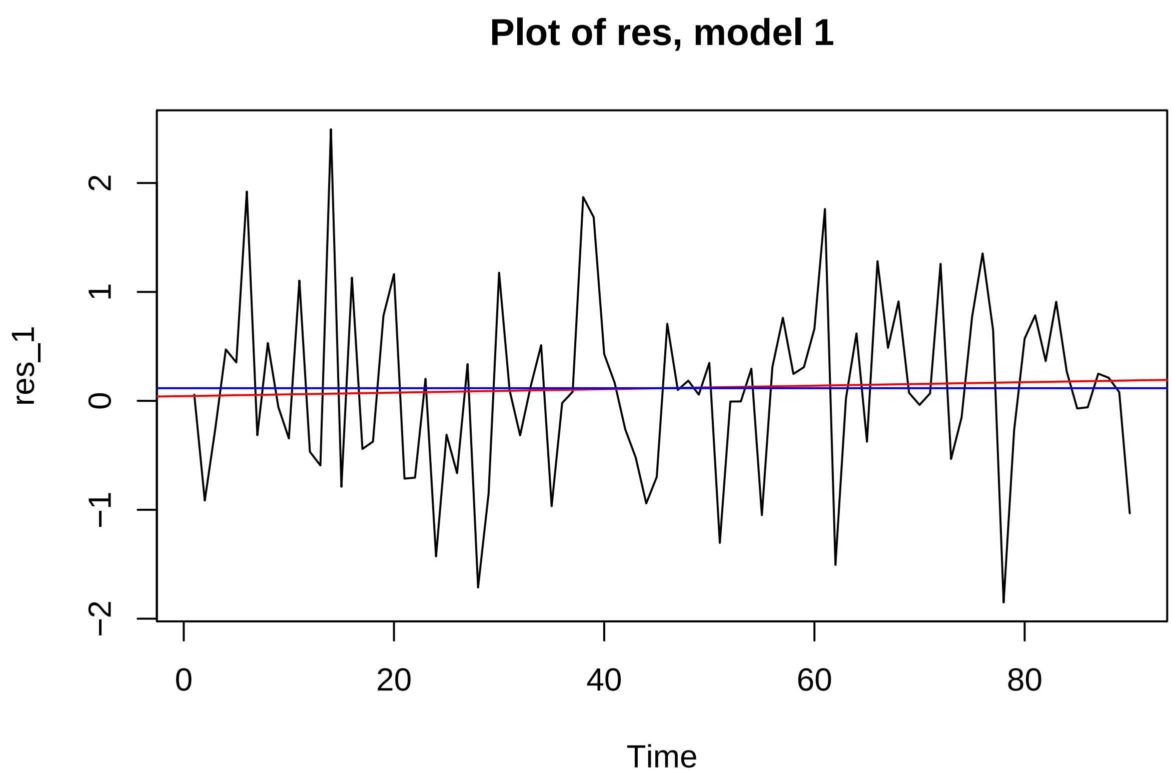
```
# roots are 4.181847 and -4.469697, both outside unit circle --> stationary!
polyroot(c(1,-0.8298))
```

```
## [1] 1.20511+0i
```

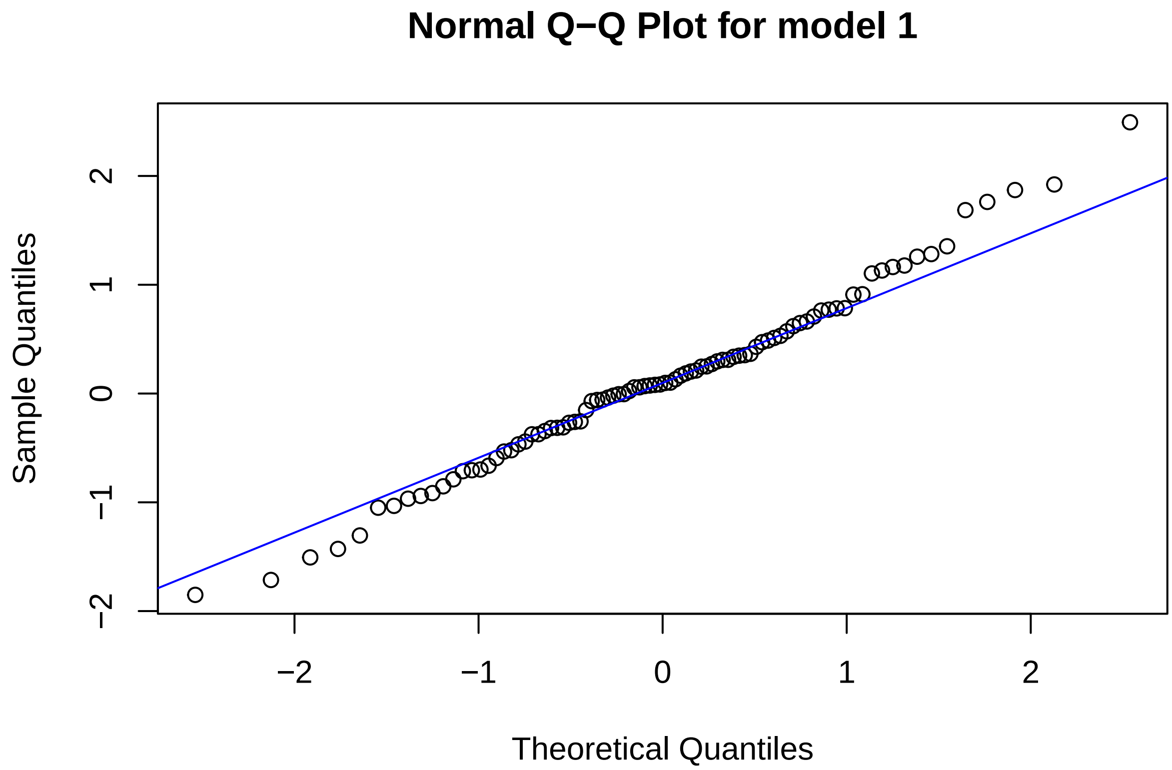
```
# root is 1.20511, outside unit circle --> invertible!

# Perform diagnostic checking
# Model 1
fit_mod1 = arima(temp, order=c(0,1,1), method="ML")
res_1 = residuals(fit_mod1)
hist(res_1, density=20, breaks=20, col="blue", xlab="", prob=TRUE, main="Histogram of res, model 1")
m = mean(res_1)
std = sqrt(var(res_1))
curve(dnorm(x,m,std), add=TRUE)
```



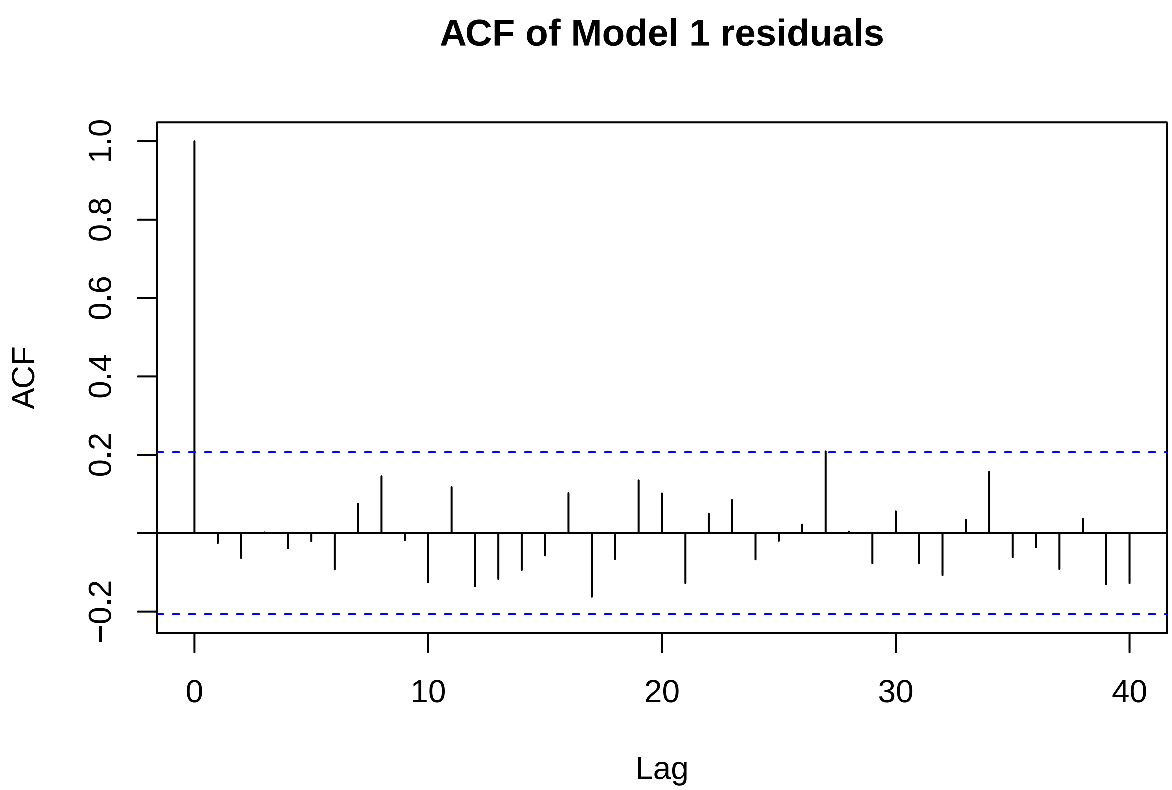

```
plot.ts(res_1, main="Plot of res, model 1")
fit_mod1_res = lm(res_1 ~ as.numeric(1:length(res_1))); abline(fit_mod1_res, col="red")
abline(h=mean(res_1), col="blue")
```



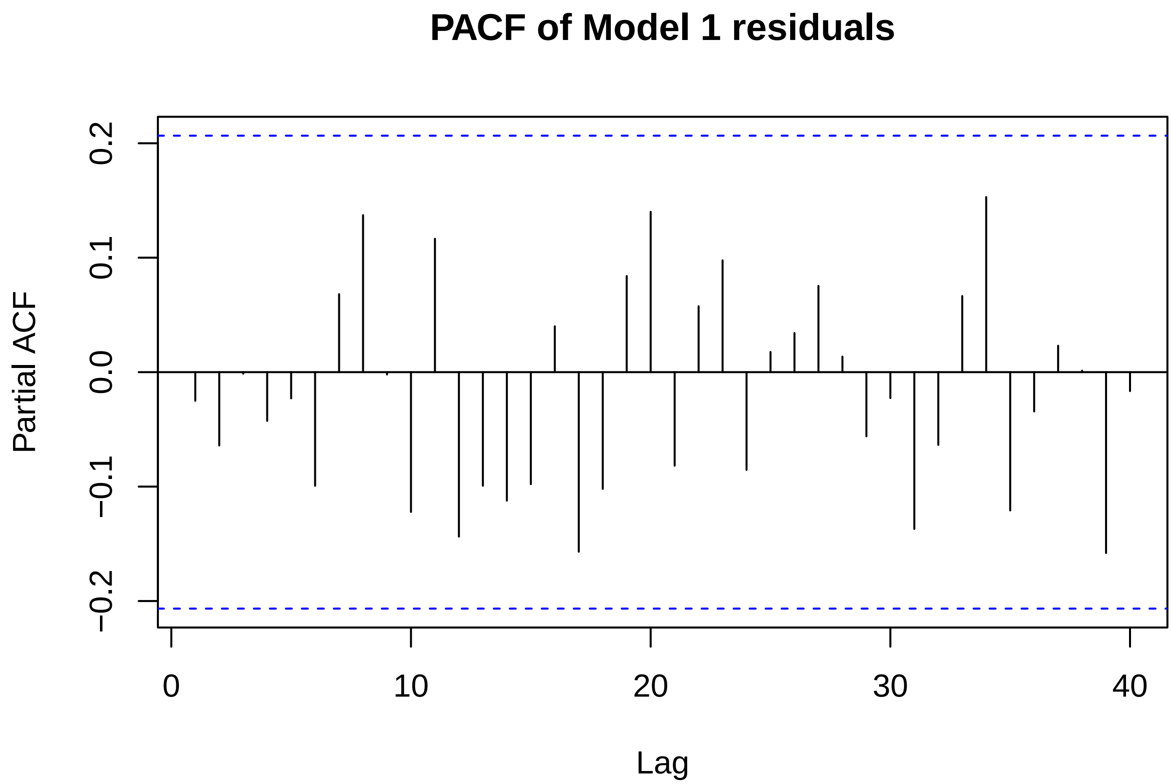

```
qqnorm(res_1, main= "Normal Q-Q Plot for model 1")
qqline(res_1, col="blue")
```




```
acf(res_1, lag.max=40, main="ACF of Model 1 residuals")
```




```
pacf(res_1, lag.max=40, main="PACF of Model 1 residuals")
```




```r
shapiro.test(res_1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res_1
## W = 0.98954, p-value = 0.6976
```

```r
# rule of thumb: take h = sqrt(n) = sqrt(100) = 10
Box.test(res_1, lag=10, type = c("Box-Pierce"), fitdf = 1) # df = 10-1 = 9
```

```
##
##  Box-Pierce test
##
## data:  res_1
## X-squared = 5.2091, df = 9, p-value = 0.8157
```

```r
Box.test(res_1, lag=10, type = c("Ljung-Box"), fitdf = 1) # df = 10-1 = 9
```

```
##
##  Box-Ljung test
##
## data:  res_1
## X-squared = 5.816, df = 9, p-value = 0.7582
```

```r
Box.test((res_1)^2, lag=10, type = c("Ljung-Box"), fitdf = 0)
```

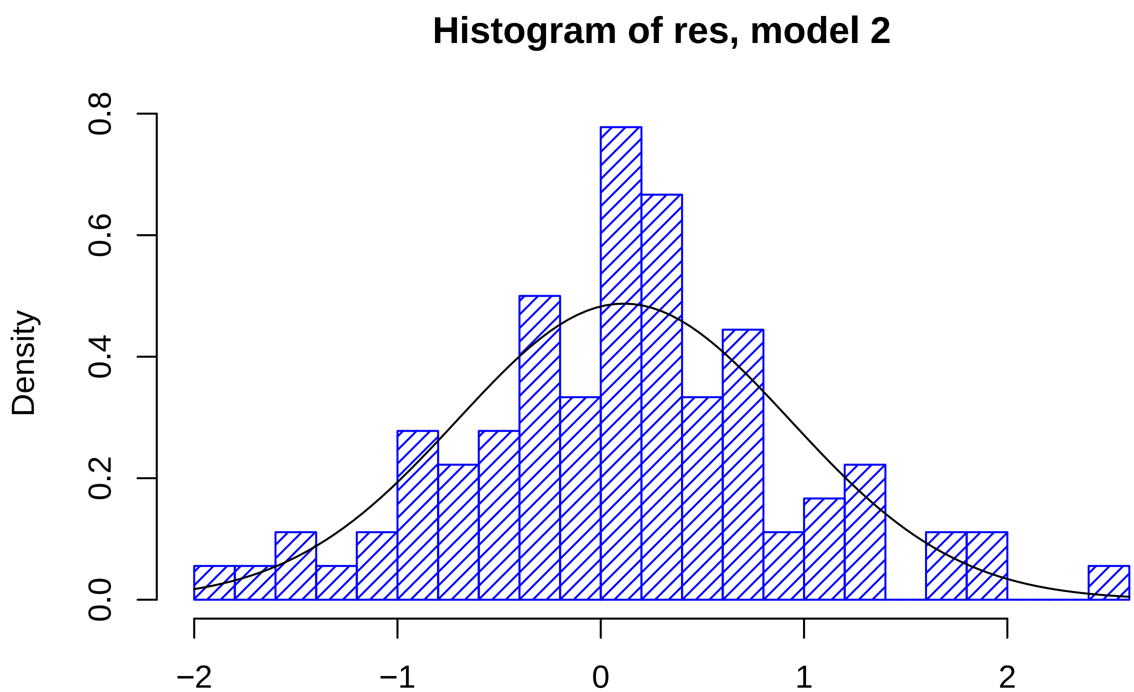
```
##
##  Box-Ljung test
##
## data:  (res_1)^2
```

```
## X-squared = 9.0517, df = 10, p-value = 0.5272
```

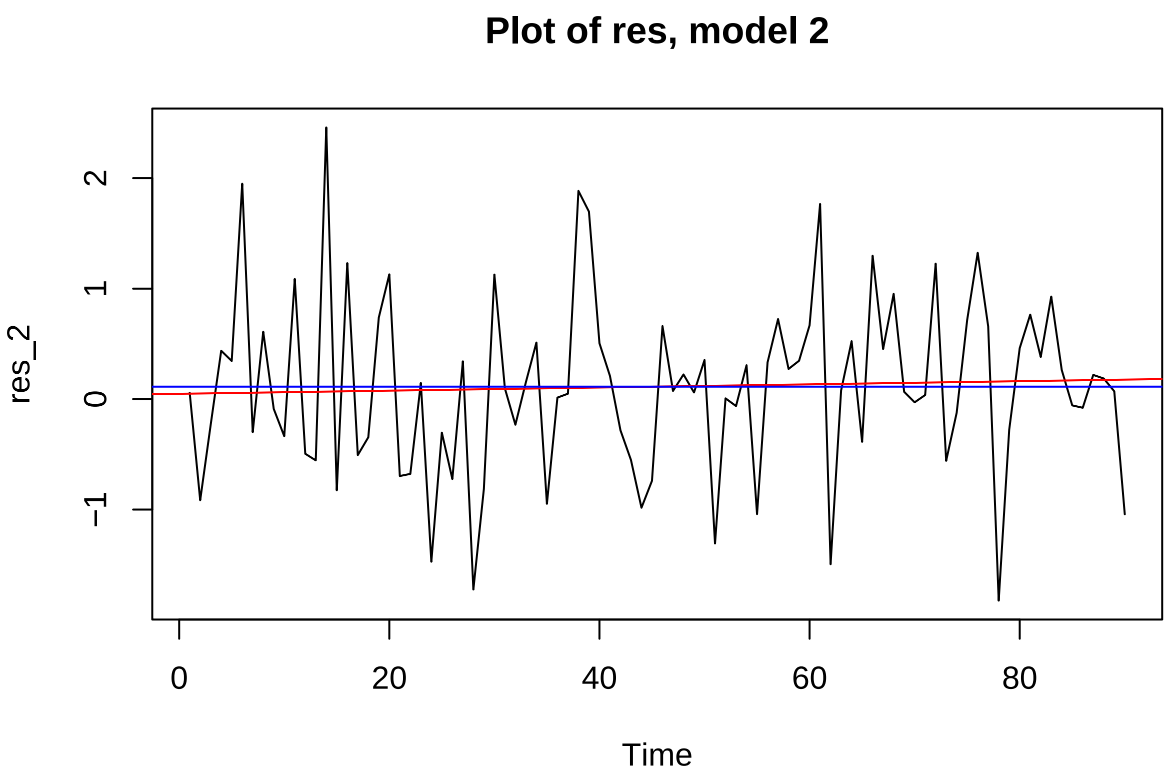
```
ar(res_1, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```
##
## Call:
## ar(x = res_1, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  0.6707
```

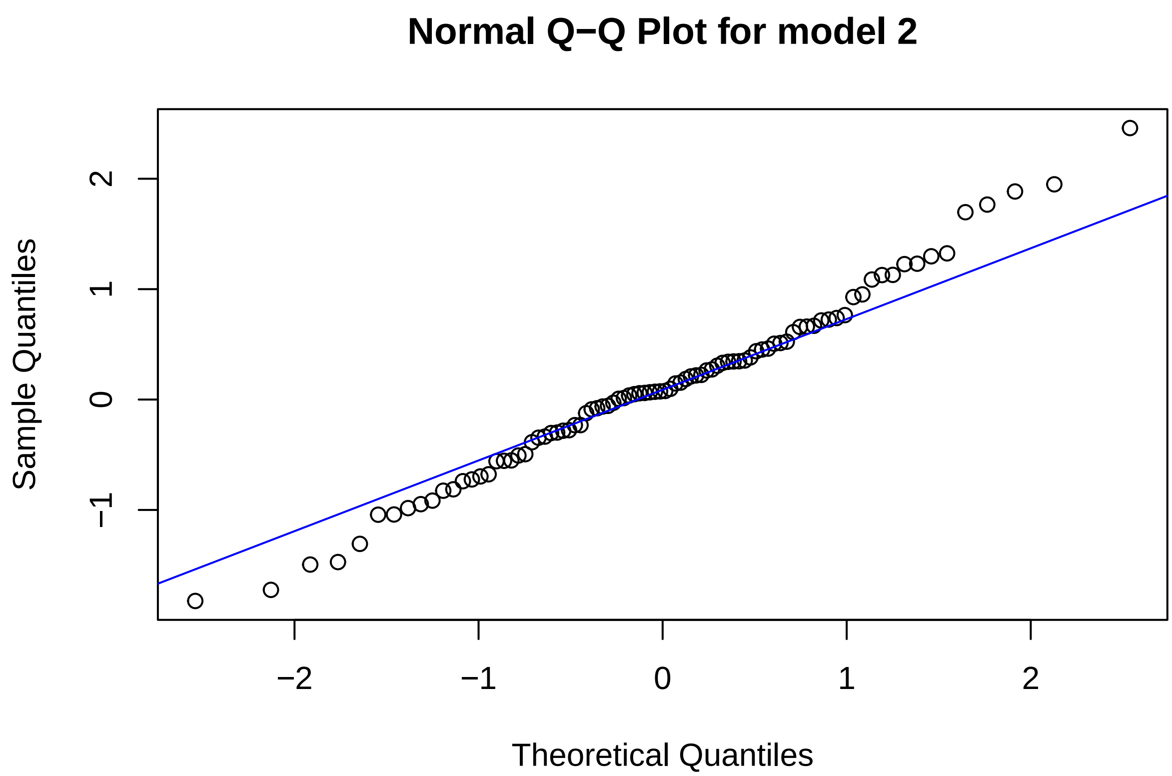
```
# Model 2
fit_mod2 = arima(temp, order=c(2,1,1), method="ML")
res_2 = residuals(fit_mod2)
hist(res_2, density=20, breaks=20, col="blue", xlab="", prob=TRUE, main="Histogram of res, model 2")
m = mean(res_2)
std = sqrt(var(res_2))
curve(dnorm(x,m,std), add=TRUE)
```



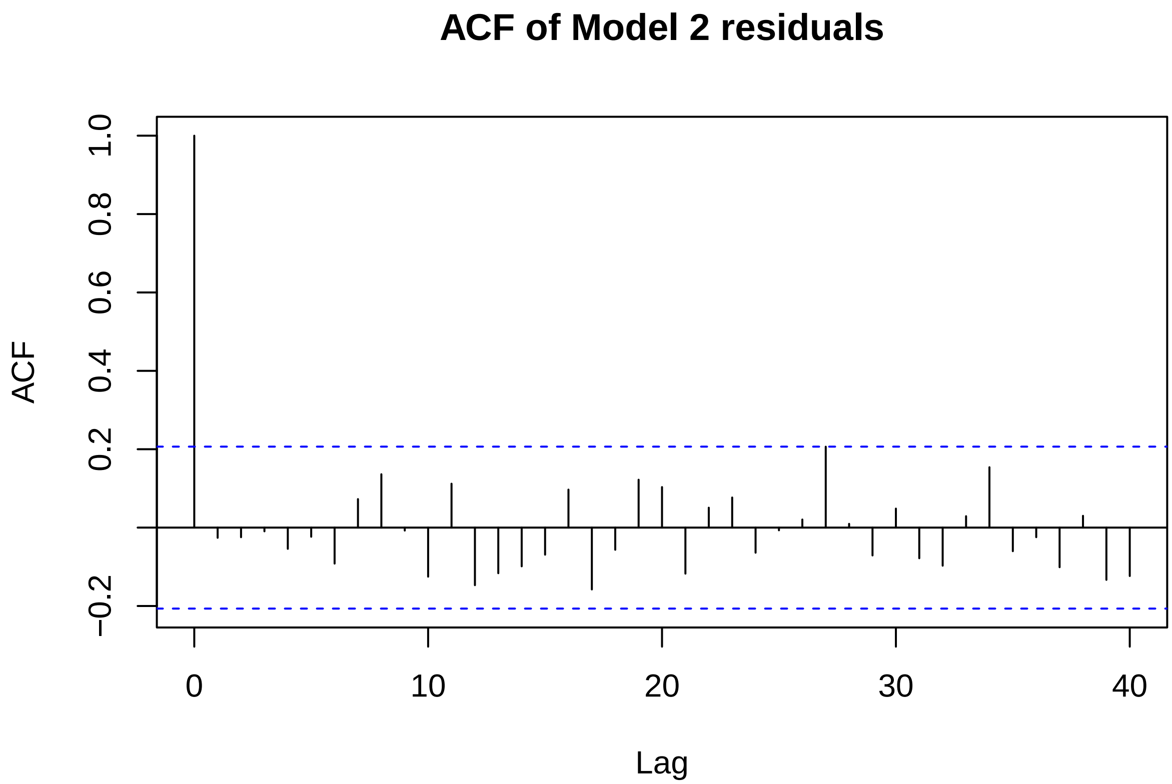

```
plot.ts(res_2, main="Plot of res, model 2")
fit_mod2_res = lm(res_2 ~ as.numeric(1:length(res_2))); abline(fit_mod2_res, col="red")
abline(h=mean(res_2), col="blue")
```



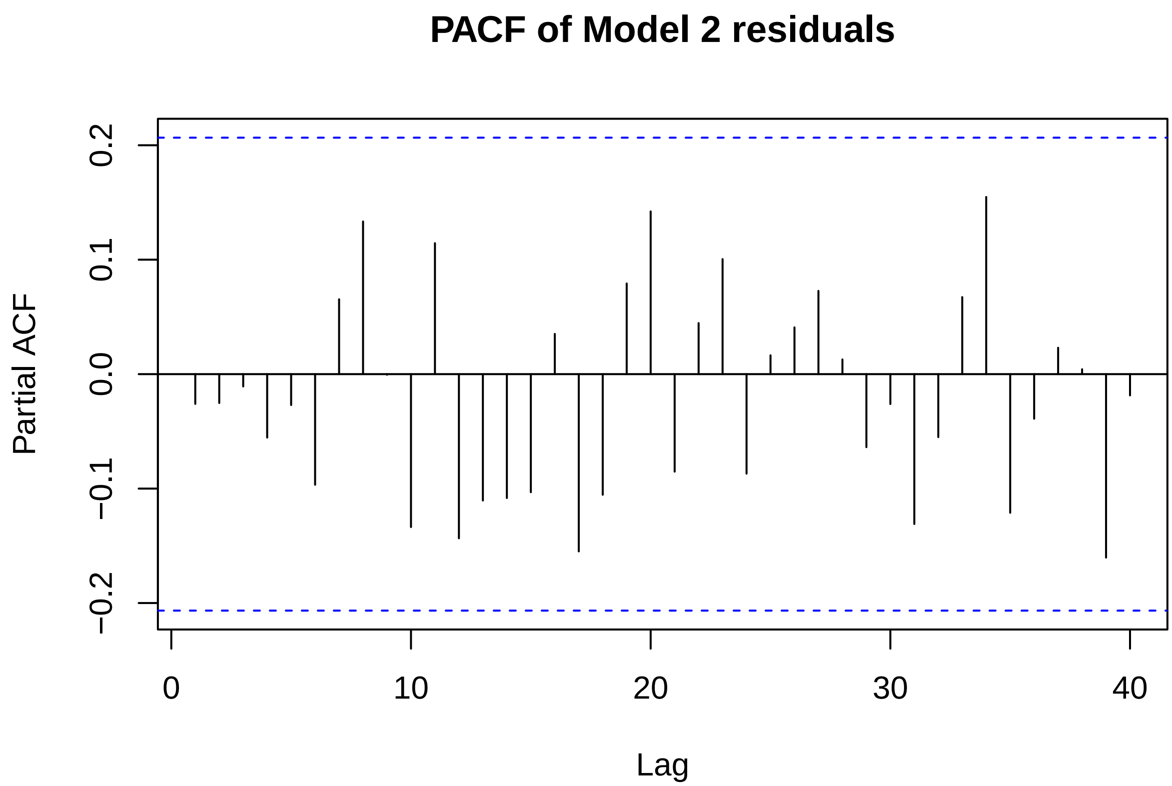

```
qqnorm(res_2, main= "Normal Q-Q Plot for model 2")
qqline(res_2, col="blue")
```




```
acf(res_2, lag.max=40, main="ACF of Model 2 residuals")
```




```
pacf(res_2, lag.max=40, main="PACF of Model 2 residuals")
```




```
shapiro.test(res_2)
```

```
##
```

```
##  Shapiro-Wilk normality test
##
## data:  res_2
## W = 0.98831, p-value = 0.6067
```

```
# rule of thumb: take h = sqrt(n) = sqrt(100) = 10
Box.test(res_2, lag=10, type = c("Box-Pierce"), fitdf = 3) # df = 10-3 = 7
```

```
##
##  Box-Pierce test
##
## data:  res_2
## X-squared = 4.7567, df = 7, p-value = 0.6896
```

```
Box.test(res_2, lag=10, type = c("Ljung-Box"), fitdf = 3) # df = 10-3 = 7
```

```
##
##  Box-Ljung test
##
## data:  res_2
## X-squared = 5.324, df = 7, p-value = 0.6205
```

```
Box.test((res_2)^2, lag=10, type = c("Ljung-Box"), fitdf = 0)
```

```
##
##  Box-Ljung test
##
## data:  (res_2)^2
## X-squared = 9.9853, df = 10, p-value = 0.4418
```

```
ar(res_2, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```
##
## Call:
## ar(x = res_2, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  0.6701
```

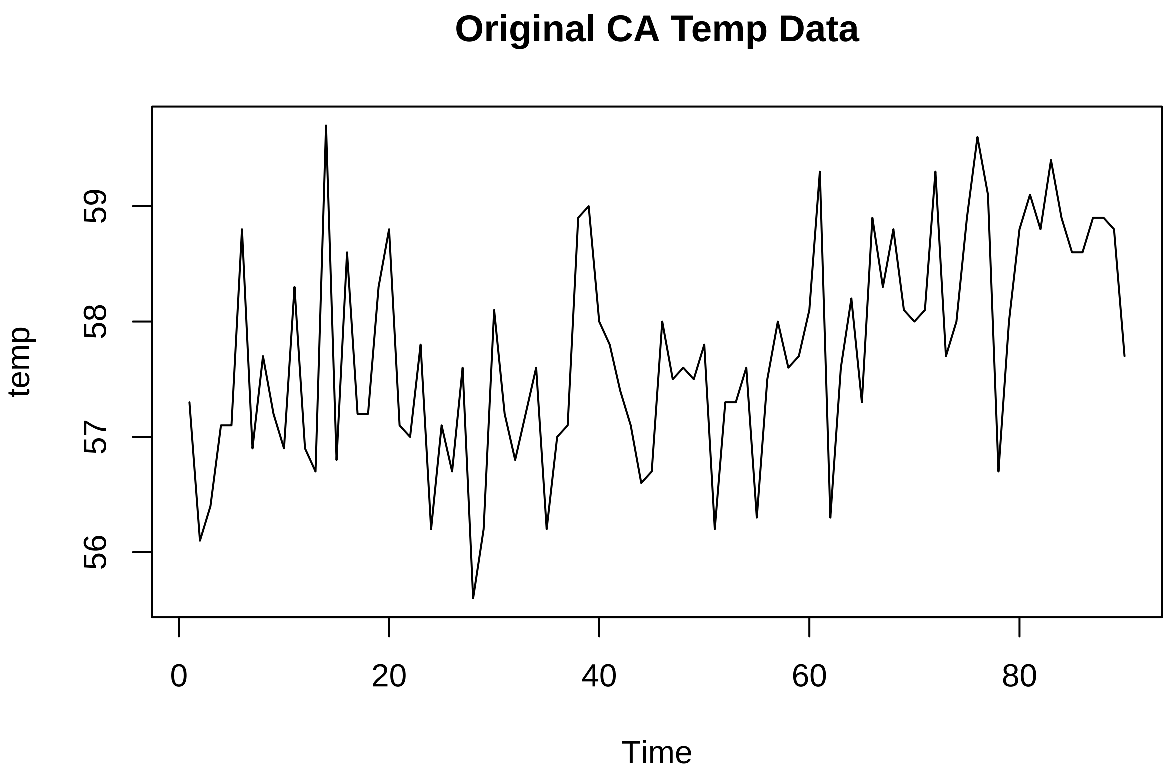
```
# Finally, check to see if auto.arima() agrees with choice of p,d,q
library("forecast")
auto.arima(temp) # ARIMA(0,1,1)
```

```
## Series: temp
## ARIMA(0,1,1) with drift
##
## Coefficients:
##           ma1   drift
##       -0.8968  0.0180
## s.e.   0.0590  0.0099
##
## sigma^2 estimated as 0.6803:  log likelihood=-108.95
## AIC=223.89   AICc=224.17   BIC=231.36
```

V: **Forecasting**

Forecasts for the next 10 observations display a very positive linear trend, with an average temperature of 58.57548 degrees. The lower bound for the confidence interval has a mean of 57.1 degrees while the upper bound has a mean of 60.5 degrees.

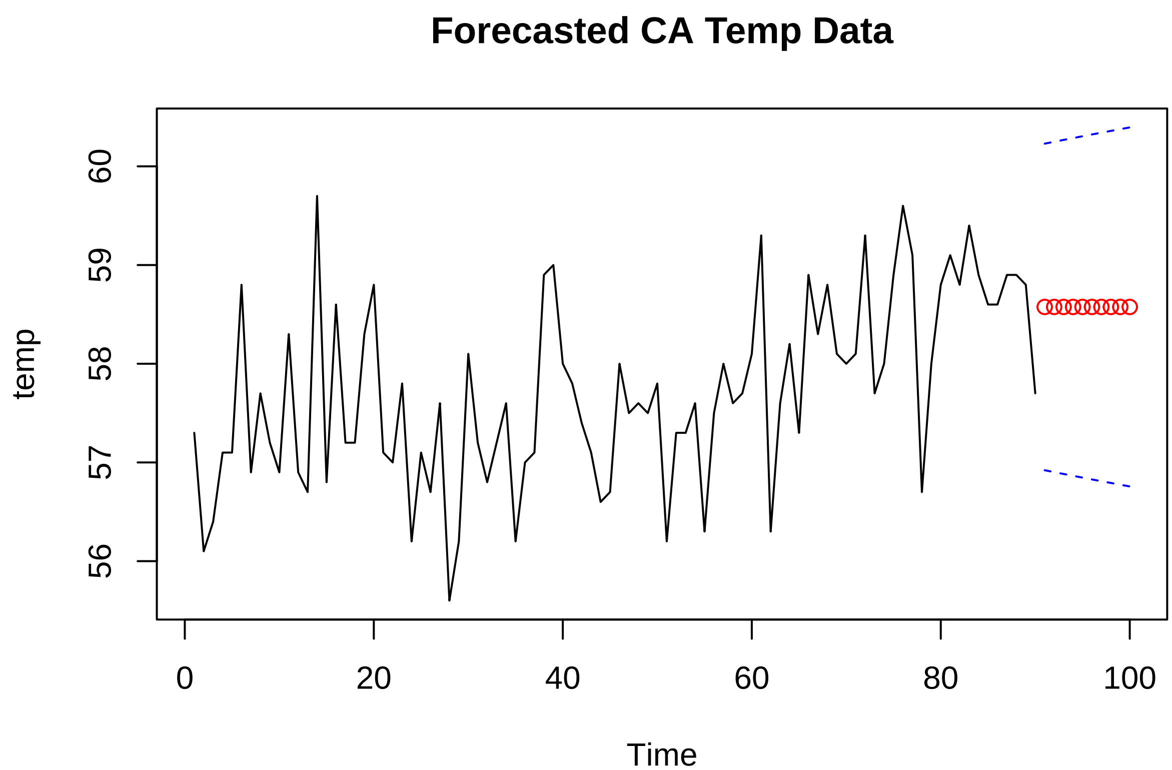
```r
# Plot original data
plot.ts(temp, main="Original CA Temp Data")
```




```r
library(forecast)
fit.fc = arima(temp, order=c(0,1,1), method="ML")
forecast(fit.fc)
```

```
##     Point Forecast    Lo 80    Hi 80    Lo 95    Hi 95
##  91       58.57548 57.51536 59.63560 56.95416 60.19680
##  92       58.57548 57.50316 59.64780 56.93551 60.21546
##  93       58.57548 57.49110 59.65987 56.91706 60.23390
##  94       58.57548 57.47917 59.67180 56.89881 60.25215
##  95       58.57548 57.46737 59.68360 56.88077 60.27020
##  96       58.57548 57.45569 59.69527 56.86291 60.28806
##  97       58.57548 57.44413 59.70683 56.84523 60.30573
##  98       58.57548 57.43269 59.71827 56.82774 60.32322
##  99       58.57548 57.42137 59.72960 56.81042 60.34055
## 100       58.57548 57.41015 59.74081 56.79326 60.35770
```

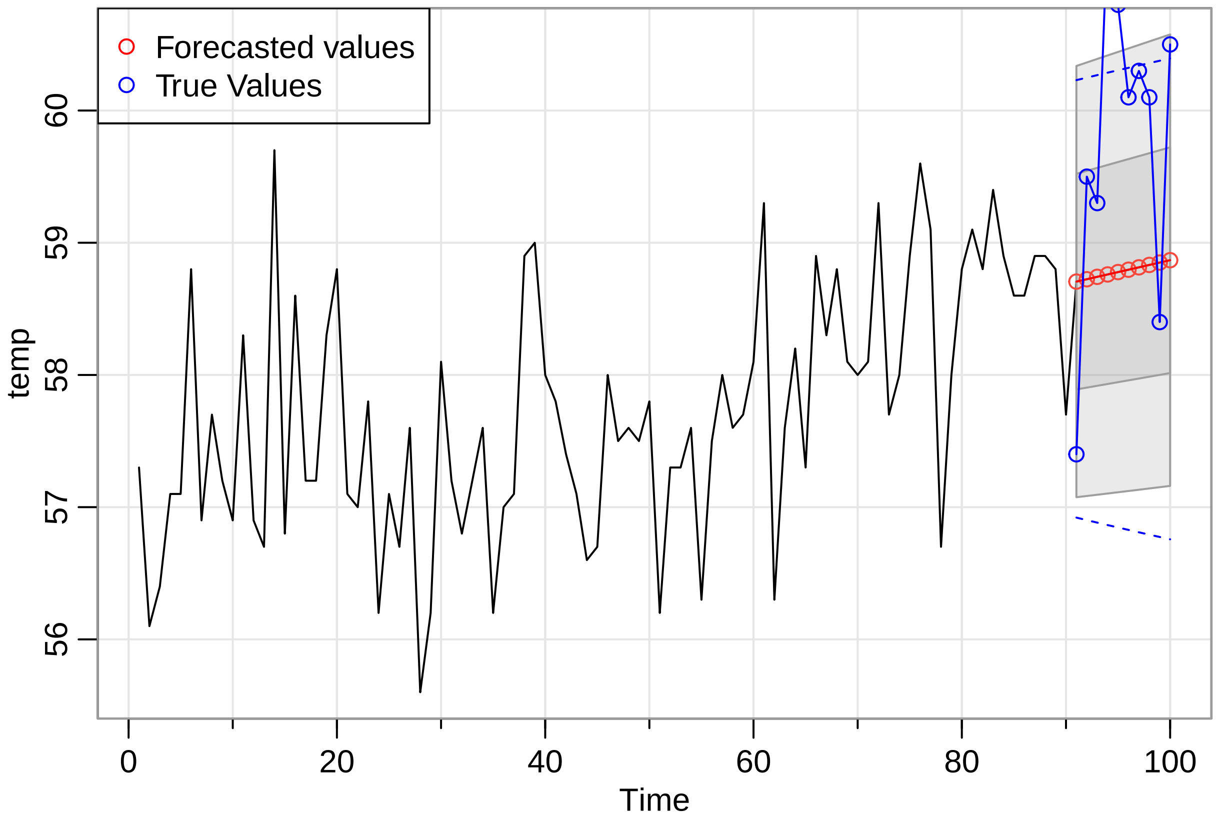
```r
pred = predict(fit.fc, n.ahead = 10)
U = pred$pred + 2*pred$se
L = pred$pred - 2*pred$se
ts.plot(temp, xlim=c(1,length(temp)+10), ylim = c(min(temp),max(U)), main="Forecasted CA Temp Data")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points(c(91:100), pred$pred, col="red")
```




```
# Another method of forecasting the data
library(astsa)
```

```
##
## Attaching package: 'astsa'

## The following object is masked from 'package:forecast':
##
##     gas
```

```
pred.tr <- sarima.for(temp, n.ahead=10, plot.all=T, p=0, d=1, q=1, P=0, D=0, Q=0, S=1)
lines(91:100, pred.tr$pred, col="red")
lines(91:100, temp.test, col="blue")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points(91:100, temp.test, col="blue")
legend("topleft", pch=1, col=c("red", "blue"), legend=c("Forecasted values", "True Values"))
```



CONCLUSION:

To recall, the questions I posed before the analysis regarded how drastically has the temperature in California increased in the last 100 years, and to what extent California's climate is expected to warm in the coming 10 years. I also considered the global problems that are caused by rising temperatures. A thorough analysis of each proposed model followed by diagnostic checking resulted in an ARIMA(0,1,1) model for the data: $X_t - X_{t-1} = Z_t - 0.8479Z_{t-1}$. In short, the conclusions of this project showcased the very real threat of global warming CA faces. In the next 10 years, average temperature is expected to rise to more than 3 degrees warmer than it was a century ago. If California residents do not try to reduce their carbon footprint, they can continue to expect hotter weather and worsening climate hazards. Finally, I would like to acknowledge and thank Professor Feldman as well as teachings assistants Sunpeng Duan and Jasmine Li for all their great help and mentorship this quarter! I learned a lot and look forward to potentially taking classes with them in the future.

REFERENCES:

<https://climate.nasa.gov/resources/global-warming-vs-climate-change/>

<https://www.energyupgradeca.org/climate-change/> <https://www.epa.gov/sites/default/files/2016-09/documents/climate-change-ca.pdf>

https://gauchospace.ucsb.edu/courses/pluginfile.php/18494621/mod_resource/content/1/Lecture%2015-AirPass%20slides.pdf

https://gauchospace.ucsb.edu/courses/pluginfile.php/18494621/mod_resource/content/1/Lecture%2015-AirPass%20slides.pdf

https://gauchospace.ucsb.edu/courses/pluginfile.php/18467560/mod_resource/content/1/week7-F21%20%20slides.pdf

https://gauchospace.ucsb.edu/courses/pluginfile.php/18429381/mod_resource/content/1/week6-Lecture%2012%20slides.pdf https://gauchospace.ucsb.edu/courses/pluginfile.php/18429353/mod_resource/content/1/week6-F21Lecture%2011-diagnostic%20checking.pdf https://www.ncdc.noaa.gov/cag/statewide/time-series/4/tavg/ytd/12/1921-2021

https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/arima

https://rpruim.github.io/s341/S19/from-class/MathinRmd.html https://rpubs.com/richkt/269797

https://statisticsglobe.com/convert-character-to-numeric-in-r/ https://www.statology.org/dickey-fuller-test-in-r/ https://wildaid.org/programs/climate/?gclid=Cj0KCQiA-qGNBhD3ARIsAO_o7ynwDC-jkZVpRxaRRF4LU7mqc_dha9SypKGjRN4DZGSe261TkJK7MOoaAoWTEALw_wcB

APPENDIX:

load the Data

temp_data = read.csv("CATemp.csv")

temp = as.numeric(temp_data$Average.Temperature

$$4:93$$

)

temp.test = as.numeric(temp_data$Average.Temperature

$$94:103$$

) # leave 10 points for model validation

plot.ts(temp, main="CA Temp Data") # stable variance, no apparent seasonality, linear trend

nt = length(temp)

fit = lm(temp ~ as.numeric(1:nt)); abline(fit, col="red")

mean(temp) # 57.71889

abline(h=mean(temp), col="blue")

hist(temp, col="light blue", xlab="", main="Histogram; CA temp data") # slightly skewed right, but somewhat symmetric

acf(temp,lag.max=40, main="ACF of the CA Temp Data") # outside at lags 1,2,3,4,5,7,8, maybe 11

Try log or BC transform to improve variance, although variance already looks stable

log transform

temp.log = log(temp)

plot.ts(temp.log, main="Log Transform")

hist(temp.log, col="light blue", xlab="", main="Histogram; ln(U_t)")

BC transform

bcTransform <- boxcox(temp ~ as.numeric(1:length(temp)))

lambda = bcTransform$x

$$which(bcTransform\$y == max(bcTransform\$y))$$

lambda # -1.43

temp.bc = (1/lambda)*(temp^lambda-1)

plot.ts(temp.bc, main="BC Transform") # slightly less variant

hist(temp.bc, col="light blue", xlab="", main="Histogram; bc(U_t)")

Not much change for either, histograms become more skewed and imply no transformation

is necessary.

Try differencing to remove linear trend.

temp.diff1 = diff(temp, lag=1)

plot.ts(temp.diff1, main="Differenced at lag 1")

fit_diff = lm(temp.diff1 ~ as.numeric(1:length(temp.diff1))); abline(fit_diff, col="red") # differencing eliminated the trend

abline(h=mean(temp.diff1), col="blue")

hist(temp.diff1, density=20, breaks=20, col="blue", xlab="", prob=TRUE, main="Differenced at lag 1")

m = mean(temp.diff1)

std = sqrt(var(temp.diff1))

curve(dnorm(x,m,std), add=TRUE)

var(temp) # 0.87301

temp.diff11 = diff(temp.diff1, lag=1) # difference again

var(temp.diff11) # 3.49 = overfitting

library(tseries) # perform ADF test for unit root/stationarity

adf.test(temp) # p-value of 0.09 = not stationary

adf.test(temp.diff1) # p-value of 0.01 = stationary!

acf(temp.diff1,lag.max=40, main="ACF of the CA Temp Data (Diff at lag 1)") # ACF outside at lags 1

pacf(temp.diff1,lag.max=40, main="PACF of the CA Temp Data (Diff at lag 1)") # PACF outside at lag 1 and 2

Proposed models to try: ARIMA(2,1,1) or ARIMA(2,1,0) or ARIMA(0,1,1) –> look at lowest AIC

arima(temp, order=c(2,1,1), method="ML") # SECOND best model

arima(temp, order=c(2,1,0), method="ML")

arima(temp, order=c(0,1,1), method="ML") # BEST model

check stationarity/invertibility of the best models - stationary is phi(z) outside unit cir, # invertible is theta(z) outside unit cir

model 1: ARIMA(0,1,1)

stationary because this is a moving average process

invertible because |theta1| < 1

model 2: ARIMA(2,1,1)

stationarity

polyroot(c(1,-0.0154,-0.0535))

roots are 4.181847 and -4.469697, both outside unit circle –> stationary!

polyroot(c(1,-0.8298))

```
# root is 1.20511, outside unit circle –> invertible!
# Perform diagnostic checking
# Model 1
fit_mod1 = arima(temp, order=c(0,1,1), method="ML")
res_1 = residuals(fit_mod1)
hist(res_1, density=20, breaks=20, col="blue", xlab="", prob=TRUE, main="Histogram of res, model 1")
m = mean(res_1)
std = sqrt(var(res_1))
curve(dnorm(x,m,std), add=TRUE)
plot.ts(res_1, main="Plot of res, model 1")
fit_mod1_res = lm(res_1 ~ as.numeric(1:length(res_1))); abline(fit_mod1_res, col="red")
abline(h=mean(res_1), col="blue")
qqnorm(res_1, main= "Normal Q-Q Plot for model 1")
qqline(res_1, col="blue")
acf(res_1, lag.max=40, main="ACF of Model 1 residuals")
pacf(res_1, lag.max=40, main="PACF of Model 1 residuals")
shapiro.test(res_1)
# rule of thumb: take h = sqrt(n) = sqrt(100) = 10
Box.test(res_1, lag=10, type = c("Box-Pierce"), fitdf = 1) # df = 10-1 = 9
Box.test(res_1, lag=10, type = c("Ljung-Box"), fitdf = 1) # df = 10-1 = 9
Box.test((res_1)^2, lag=10, type = c("Ljung-Box"), fitdf = 0)
ar(res_1, aic = TRUE, order.max = NULL, method = c("yule-walker"))
# Model 2
fit_mod2 = arima(temp, order=c(2,1,1), method="ML")
res_2 = residuals(fit_mod2)
hist(res_2, density=20, breaks=20, col="blue", xlab="", prob=TRUE, main="Histogram of res, model 2")
m = mean(res_2)
std = sqrt(var(res_2))
curve(dnorm(x,m,std), add=TRUE)
plot.ts(res_2, main="Plot of res, model 2")
fit_mod2_res = lm(res_2 ~ as.numeric(1:length(res_2))); abline(fit_mod2_res, col="red")
abline(h=mean(res_2), col="blue")
qqnorm(res_2, main= "Normal Q-Q Plot for model 2")
qqline(res_2, col="blue")
acf(res_2, lag.max=40, main="ACF of Model 2 residuals")
pacf(res_2, lag.max=40, main="PACF of Model 2 residuals")
```

```
shapiro.test(res_2)
# rule of thumb: take h = sqrt(n) = sqrt(100) = 10
Box.test(res_2, lag=10, type = c("Box-Pierce"), fitdf = 3) # df = 10-3 = 7
Box.test(res_2, lag=10, type = c("Ljung-Box"), fitdf = 3) # df = 10-3 = 7
Box.test((res_2)^2, lag=10, type = c("Ljung-Box"), fitdf = 0)
ar(res_2, aic = TRUE, order.max = NULL, method = c("yule-walker"))
# Finally, check to see if auto.arima() agrees with choice of p,d,q
auto.arima(temp) # ARIMA(0,1,1)
# Plot original data
plot.ts(temp, main="CA Temp Data")
library(forecast)
fit.fc = arima(temp, order=c(0,1,1), method="ML")
forecast(fit.fc)
pred = predict(fit.fc, n.ahead = 10)
U = pred.tr$pred + 2*pred.tr$se
L = pred.tr$pred - 2*pred.tr$se
ts.plot(temp, xlim=c(1,length(temp)+10), ylim = c(min(temp),max(U)), main="Forecasted CA Temp Data")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points(c(91:100), pred$pred, col="red")
# Another method of forecasting the data
library(astsa)
pred.tr <- sarima.for(temp, n.ahead=10, plot.all=T, p=0, d=1, q=1, P=0, D=0, Q=0, S=1)
lines(91:100, pred.tr$pred, col="red")
lines(91:100, temp.test, col="blue")
lines(U, col="blue", lty="dashed")
lines(L, col="blue", lty="dashed")
points(91:100, temp.test, col="blue")
legend("topleft", pch=1, col=c("red", "blue"), legend=c("Forecasted values", "True Values"))
```