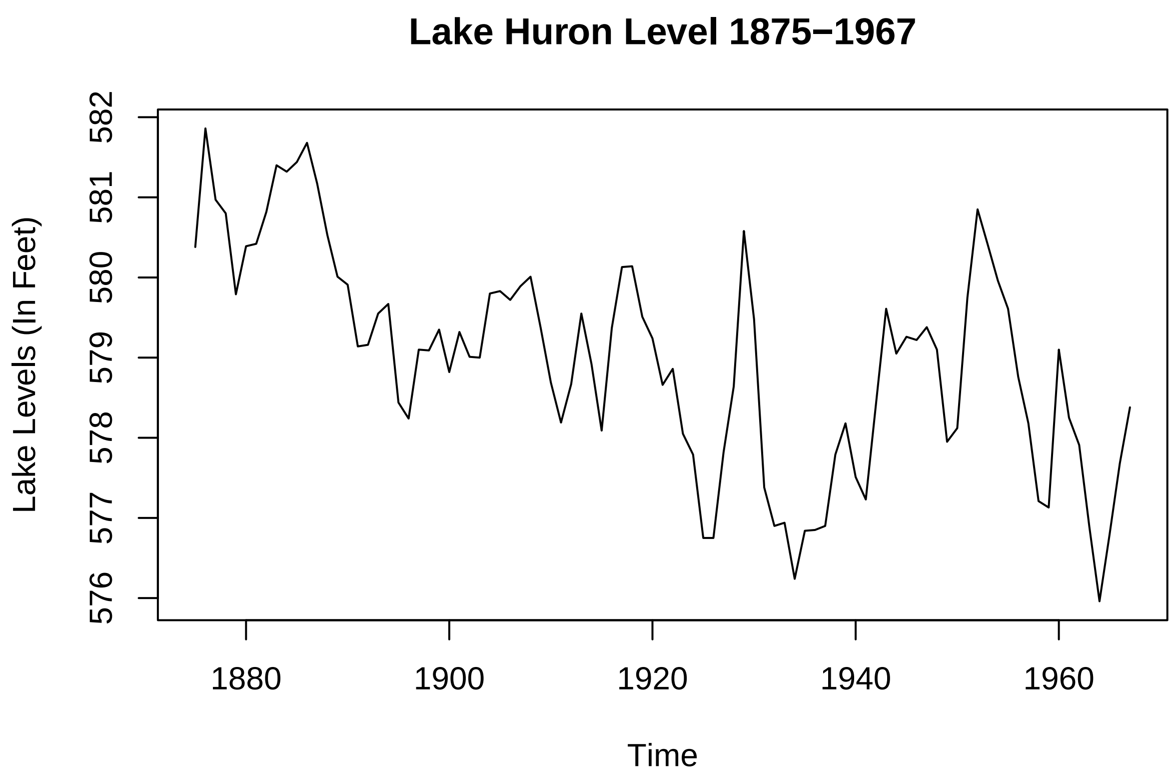
STAT 443: Lab 9

Saksham Sudershan (Student #31339427)

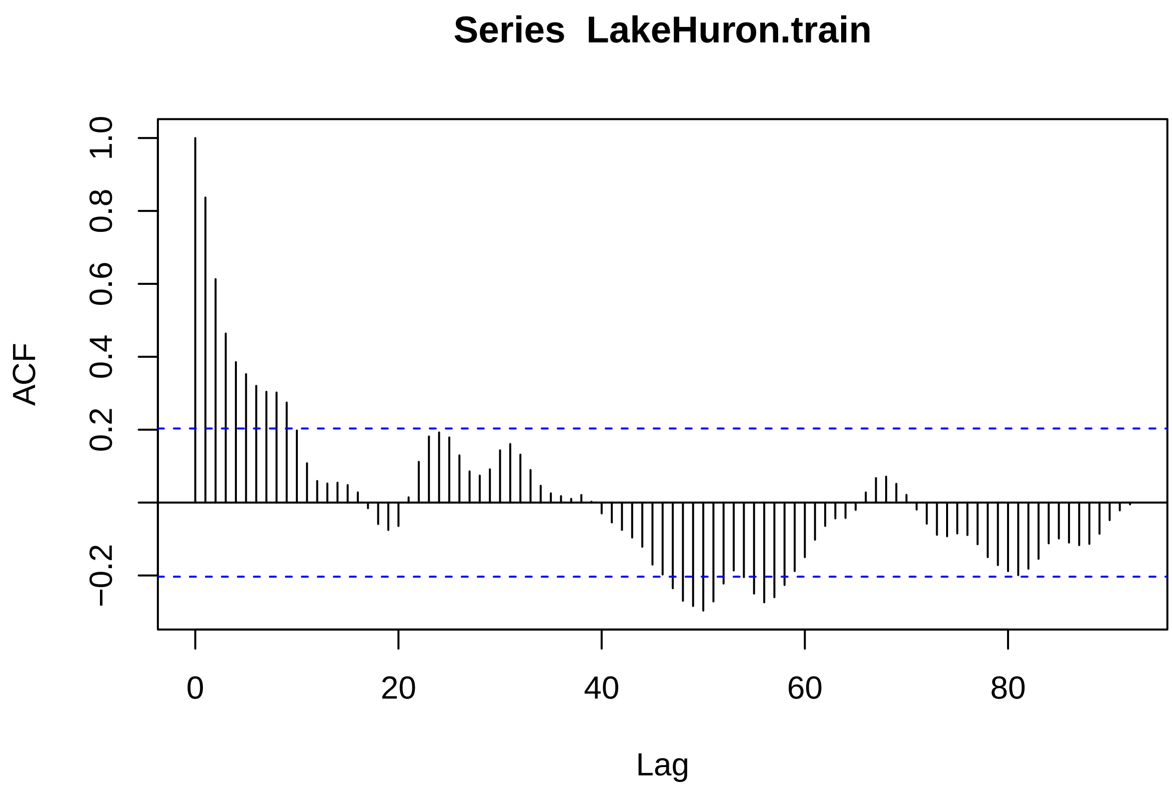
21 March, 2022

Problem 1

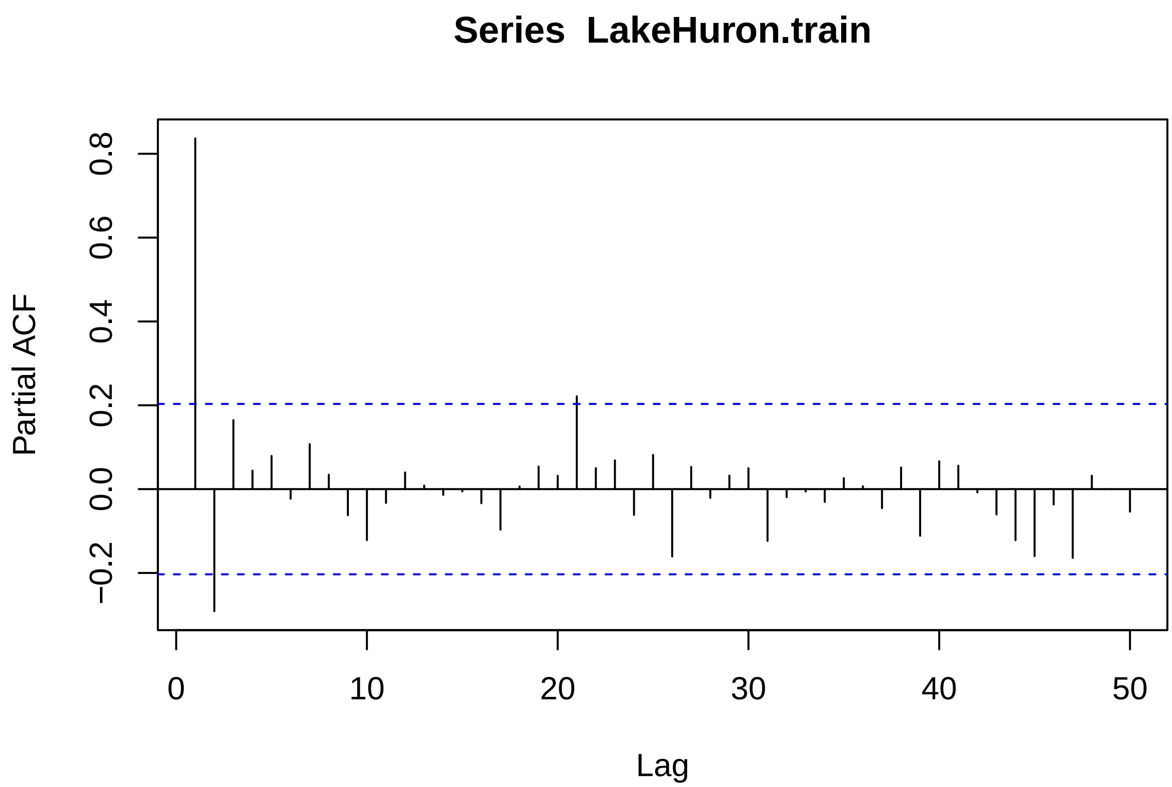
```r
# Reading data and creating datasets
data("LakeHuron")
LakeHuron.train <- window(LakeHuron, start = 1875, end = 1967)
LakeHuron.test <- window(LakeHuron, start = 1968, end = 1972)

# Plotting training data
plot(LakeHuron.train, ylab = "Lake Levels (In Feet)",
     main = "Lake Huron Level 1875-1967")
```



```
acf(LakeHuron.train, lag.max = 100)
```




```
pacf(LakeHuron.train, lag.max = 50)
```



The ACF tails off. The PACF cuts of lag $h = 2$. This resembles an AR(2) process as the ACF tails off and the PACF cuts of at lag p.

Problem 2

```r
# Fitting an AR(2) process
LakeHuron.model <- arima(LakeHuron.train, order = c(2,0,0))
LakeHuron.model
```

```
##
## Call:
## arima(x = LakeHuron.train, order = c(2, 0, 0))
##
## Coefficients:
##          ar1      ar2  intercept
##       1.0627  -0.2691   578.9888
## s.e.  0.1006   0.1035     0.3396
##
## sigma^2 estimated as 0.4815:  log likelihood = -98.65,  aic = 205.31
```

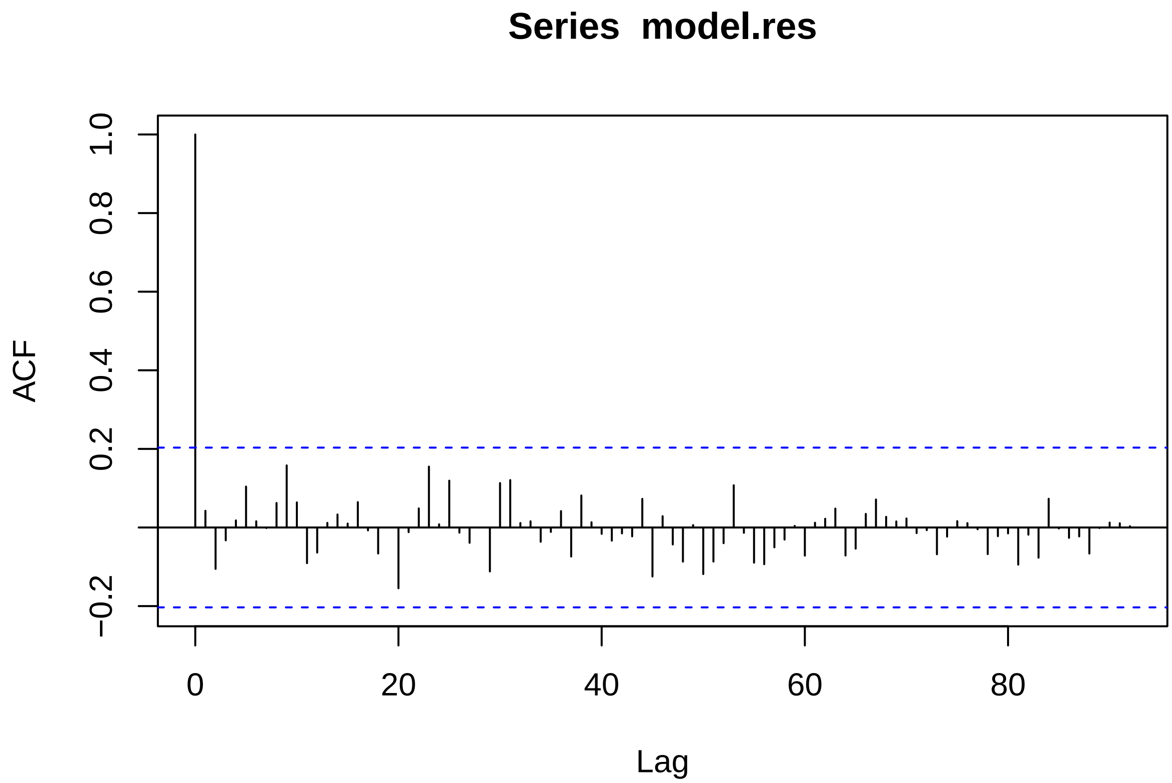
The fitted model is:

$$X_t - \mu = \alpha_1(X_{t-1} - \mu) + \alpha_2(X_{t-2} - \mu) + Z_t$$

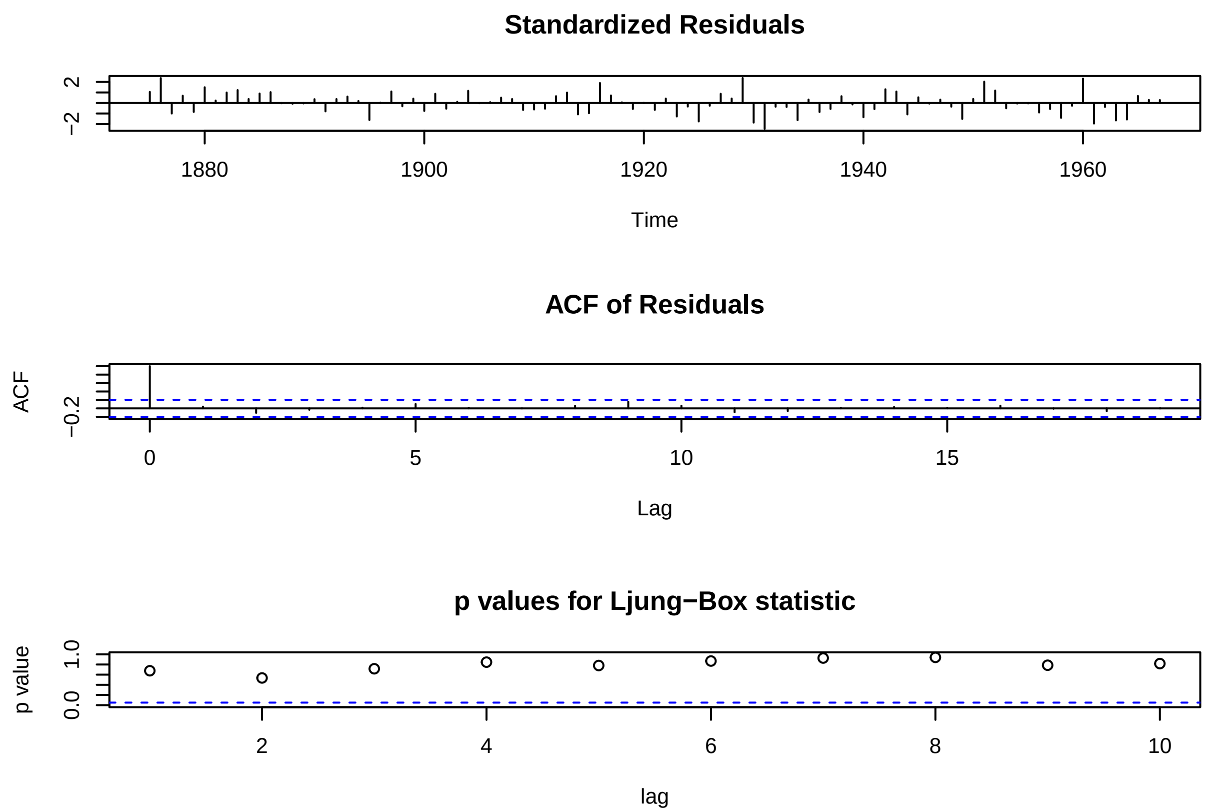
$$\Rightarrow X_t = 578.9888 + 1.0627(X_{t-1} - 578.9888) - 0.2691(X_{t-2} - 578.9888) + Z_t \qquad where\ Z_t \sim N(0, 0.4815)$$

Problem 3

```
# ACF of residuals
model.res <- ts(residuals(LakeHuron.model))
acf(model.res, lag.max = 100)
```




```
# Model Diagnostics
tsdiag(LakeHuron.model)
```



The residuals seem to have no significant autocorrelations between the lags.

The standardized residuals also seem to not follow any pattern and appear to resemble white noise. The p-values for Ljung-Box statistic is higher than 0.6 for most lags, thus we can comfortably say that this model fits fairly well.

Problem 4

```
# Finding 95% CI
predicted.val <- data.frame(predict(LakeHuron.model, n.ahead = 3,
                                    level = 0.95, prediction.interval = T))
for(i in 1:3){
  predicted.val[i, 3] <- predicted.val[i,1]-1.96*predicted.val[i,2]
  predicted.val[i, 4] <- predicted.val[i,1]+1.96*predicted.val[i,2]
}
colnames(predicted.val) <- c("pred", "se", "lwr", "upr")
predicted.val
```

```
##       pred        se      lwr      upr
## 1 578.6940 0.6938894 577.3340 580.0540
## 2 578.8394 1.0125382 576.8548 580.8239
## 3 578.9093 1.1753820 576.6056 581.2131
```

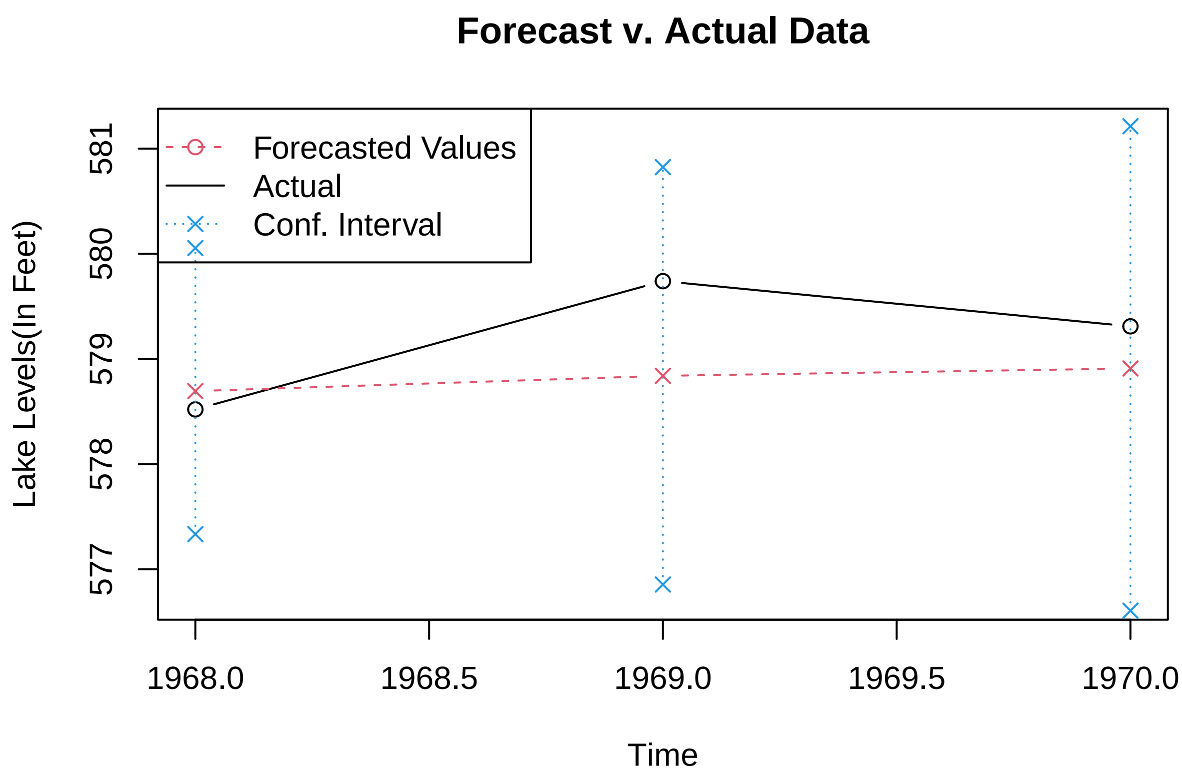
The forecast value for 1968 is 578.694, and the 95% CI is (577.334,580.054).

The forecast value for 1969 is 578.8394, and the 95% CI is (576.8548,580.8239).

The forecast value for 1970 is 578.9093, and the 95% CI is (576.6056,581.2131).

Problem 5

```
# Making plot of CI, predicted and actual values
plot(c(1968, 1969, 1970),LakeHuron.test[1:3], type = "b", lty = 1,
     col = 1, ylim = c(576.7,581.2), ylab = "Lake Levels(In Feet)",
     xlab = "Time", main = "Forecast v. Actual Data")
points(c(1968, 1969, 1970), predicted.val[1:3,3], pch = 4, col = 4)
segments(x0=1968 ,y0=predicted.val[1,3],x1=1968,y1=predicted.val[1,4], lty = 3, col = 4)
segments(x0=1969,y0=predicted.val[2,3],x1=1969,y1=predicted.val[2,4], lty = 3, col = 4)
segments(x0=1970,y0=predicted.val[3,3],x1=1970,y1=predicted.val[3,4], lty = 3, col = 4)
points(c(1968, 1969, 1970), predicted.val[1:3,4], pch = 4, col = 4)
lines(c(1968, 1969, 1970),predicted.val[1:3,1], type = "b",lty = 2, col = 2, pch = 4)
legend("topleft", legend = c("Forecasted Values", "Actual","Conf. Interval"),
       lty = c(2,1, 3), col = c(2,1, 4), pch = c(1, NA, 4))
```



The graph above shows that the predicted values are very close to the actual lake levels. This actual values also fall within the 95% confidence interval produced using our forecast. Thus, the forecast seems to work well.