# The Heart of Healthcare: Analyzing and Forecasting Hospitality Employment in the Golden State

Kevin Brunia

2025-05-17

## Abstract

The hospitality industry is a big driver of America's GDP, generating upwards of 500 billion dollars per year. In this forecasting project, we are interested in forecasting the number of employees in the hospitality/service industry. This is important because the low (and high) points can serve as markers for how the economy is doing, which is really helpful.

We selected a

$$\text{SARIMA}(2,1,2)\times(0,1,1)_{12}$$

model to forecast the number of employees in the service industry. After looking at 4 potential models, this one came out on top. By comparing the forecasted values with the actual values, we found that the model fit the test set pretty well, with a slight overestimation of the employee count. The gap increased as time progressed.

The data pertaining to this project documents the change in employees in the hospitality service industry in California from 1990-01-01 to 2018-12-01. Specifically, it looks at the monthly average employed in thousands of persons. It is a seasonal time series dataset.

## Introduction

I obtained this dataset from Kaggle. This is a fairly simple dataset consisting of two columns: `Date`, which is in the form month/day/year taken on the first day of every month, and `# Employees`, indicating the amount employed for that given month. There are a total of 348 valid rows, translating to nearly 30 years of monthly data. I was interested in this dataset upon discovery since I have never looked into employment data of this magnitude before. I think it's also important because California has such a large population, discrepancies are sure to be noticed. The industry itself is also huge, so there are a few implications that can be made from this. The relative volatility of the job market here is also due to seasonality, so this project is aimed to aid in preparing for these shifts. Forecasting and the impact of external events are the two targets of this endeavor.

In the process, the techniques that may play a role include:

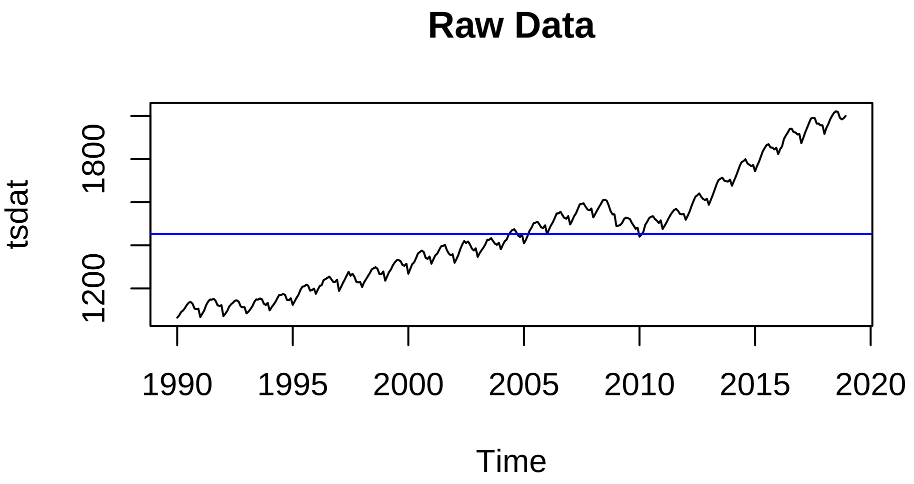
- ACF/PACF exploration
- Checking stationarity/invertibility
- Diagnostic Checking
- Differencing
- External event connection
- Forecasting
- Model estimation
- Model identification

- Transforming
- Visualization techniques

Project was done using R version 4.4.1 (2024-06-14 ucrt)

### Forecasting

We will start out by visualizing the data

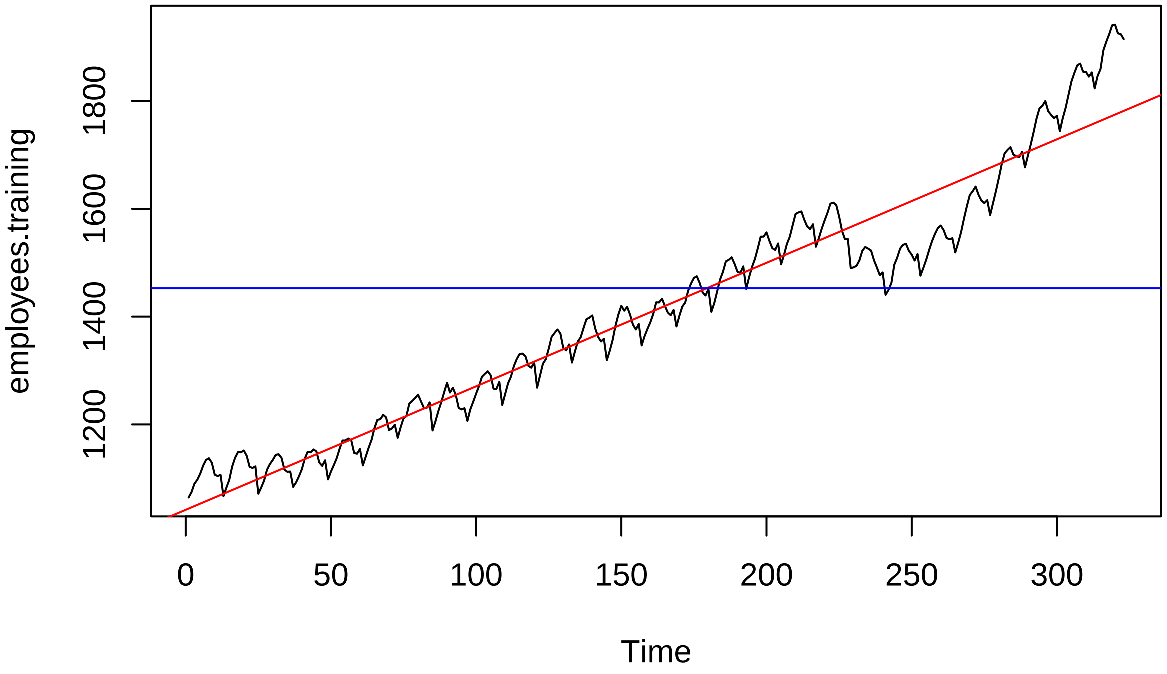


Some immediate obseravtion of this dataset is that:

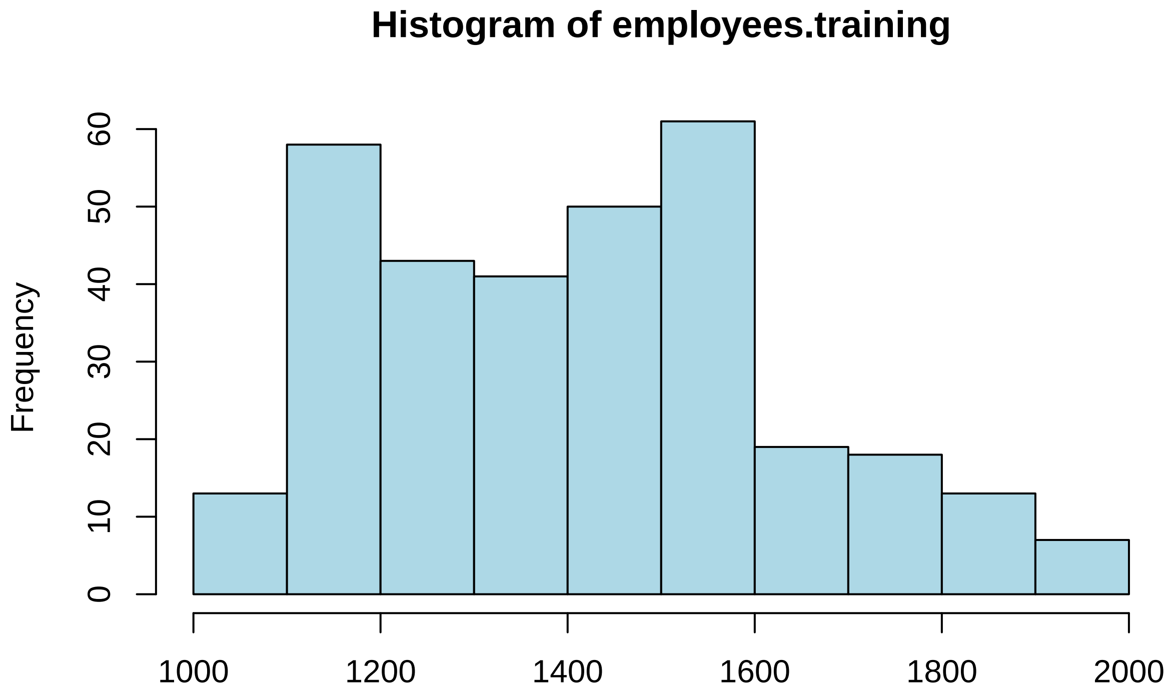
- the data looks very linear
- there is a not-so-obvious seasonality
- variance looks constant (apart from the dip around 2009) and a non-constant mean

This **drop of employees in 2008-2009**, is possibly due to the 2008 global financial collapse that began in the US.

Now we need to partition the data into a training and testing portion. We leave 24 data points for verification.

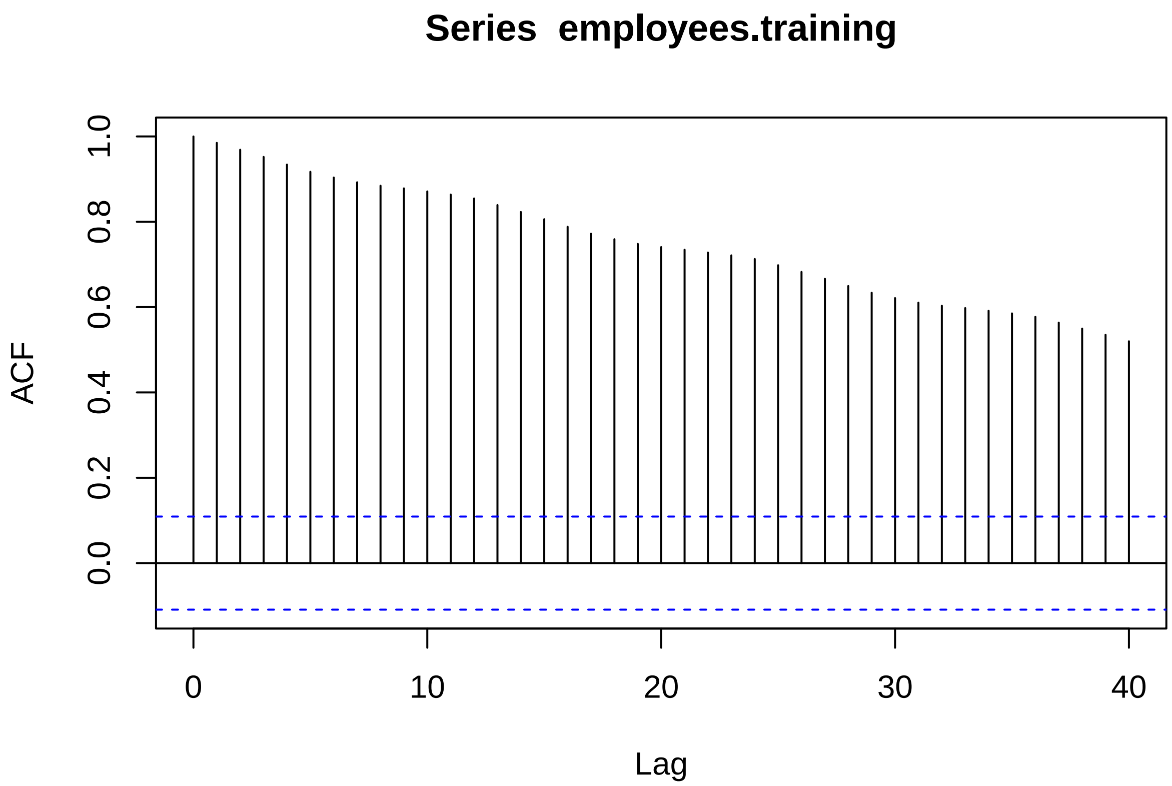

employees.training
1200
1400
1600
1800
0
50
100
150
200
250
300
Time

**Histogram**



The histogram is slightly right skewed. We can treat it as constant variance, so I don't believe we need a transformation.

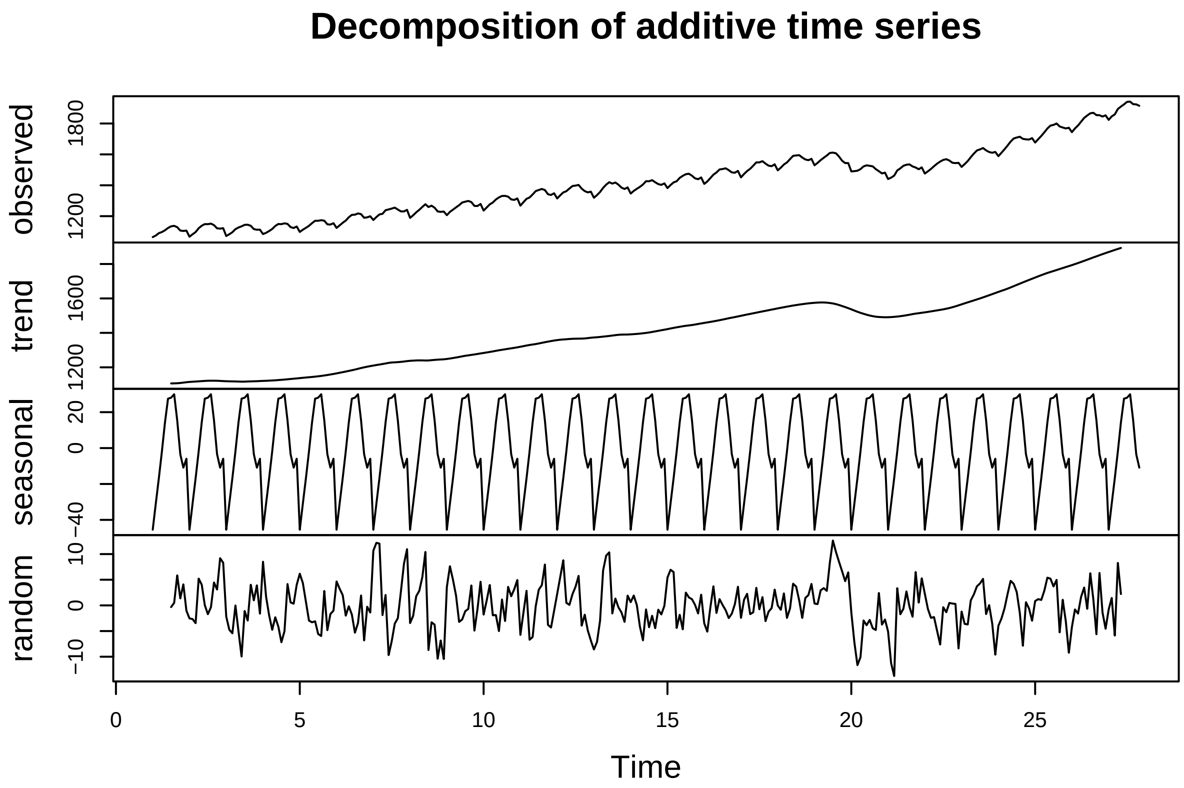
We can see the ACF to check seasonality and trend



Very nice looking acf. It gradually descends but remains large, so we can difference.

### Differencing

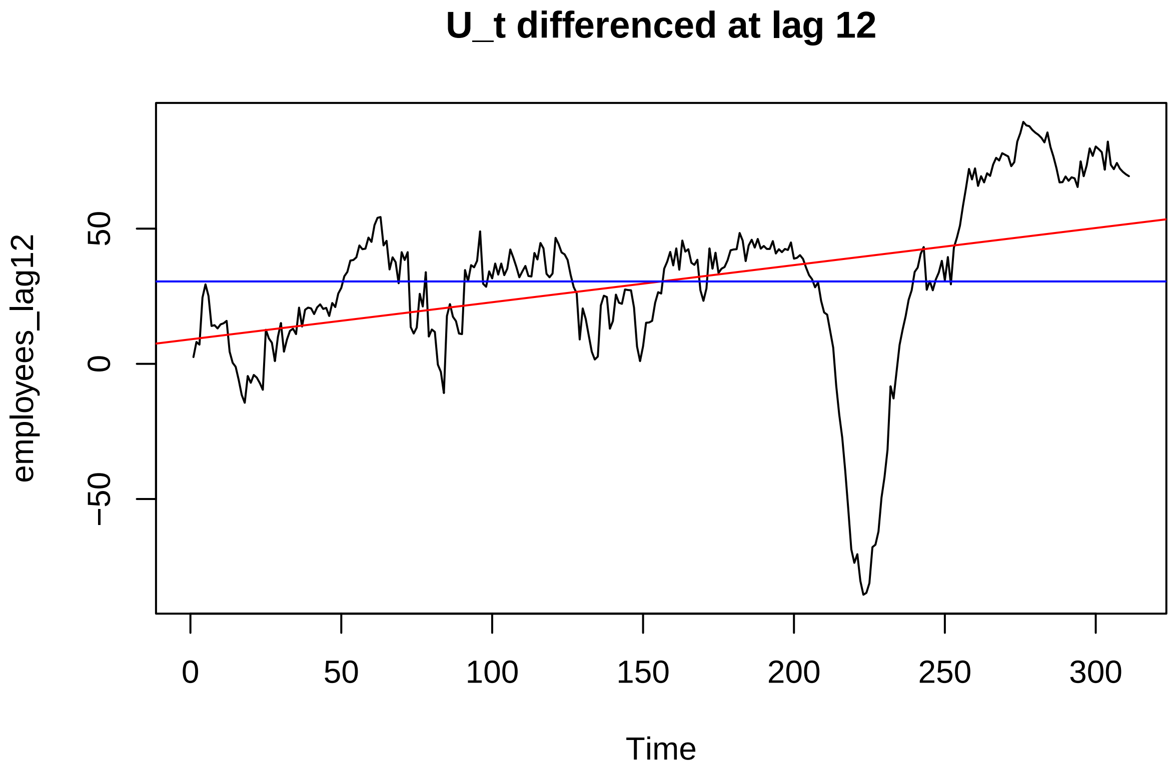
First, to decompose the time series



After decomposing the time series, it shows seasonality and a linear trend, despite the dip at 2008-2009. Therefore, we can go ahead and difference the data.

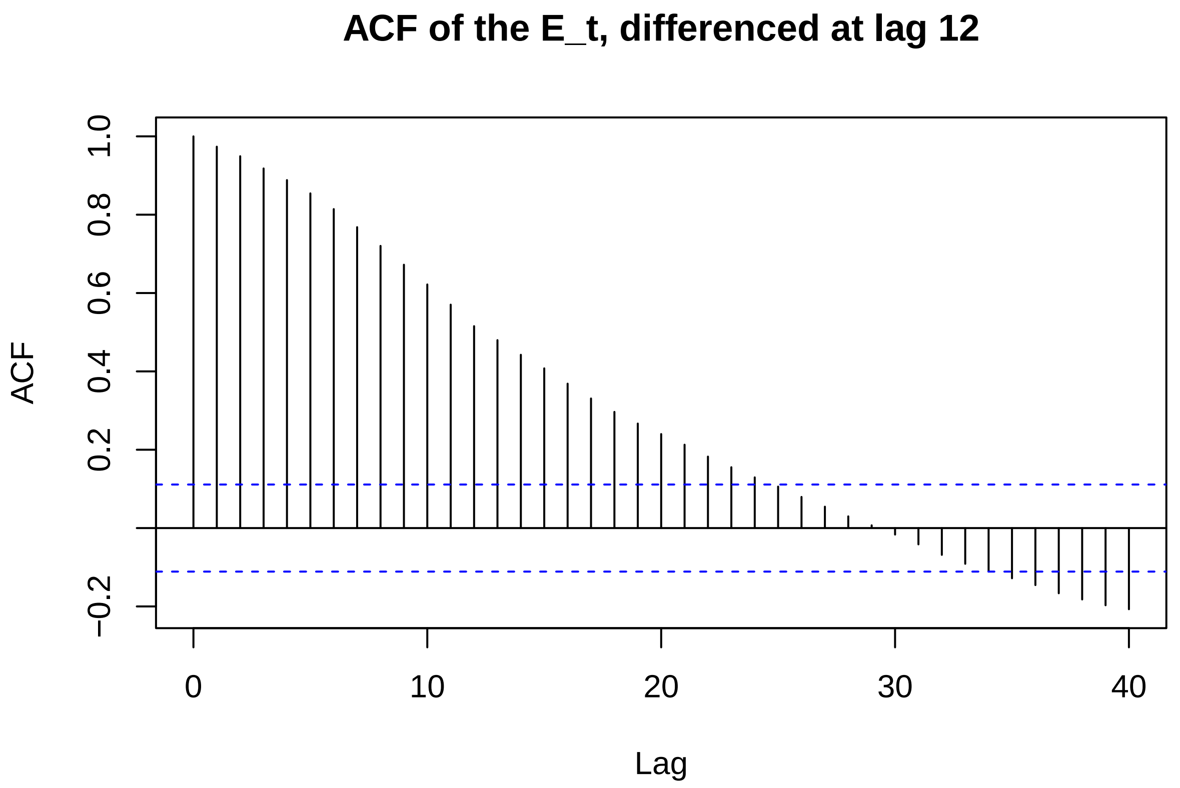
```
## [1] 48721.73
```

Our time series has variance of 48721.73. We start off by differencing at lag 12:



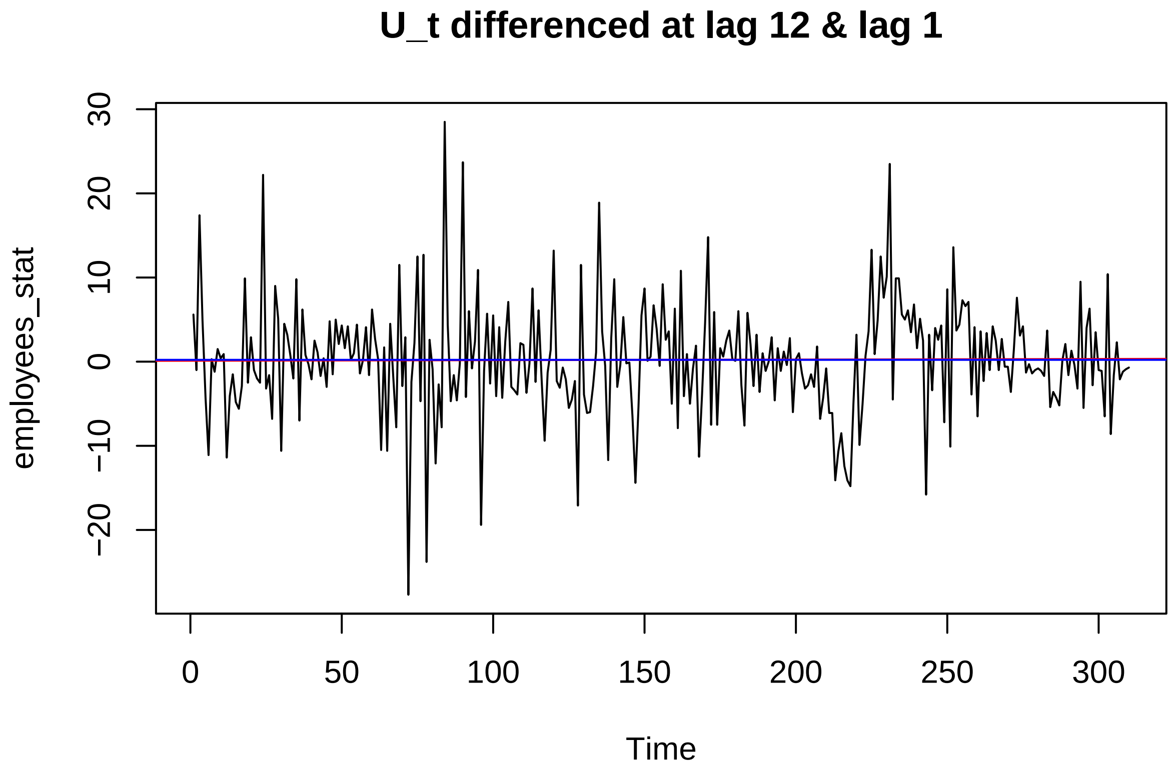
```
## [1] 1024.479
```

We can see the variance went down to 979, which is a lot lower than before. It is still somewhat big, however, and we have a non-zero mean. The ACF of the series differencing at lag 12 looks like:



We see that seasonality looks no longer apparent, and the ACFs are decaying slowly. The series is still non stationary.

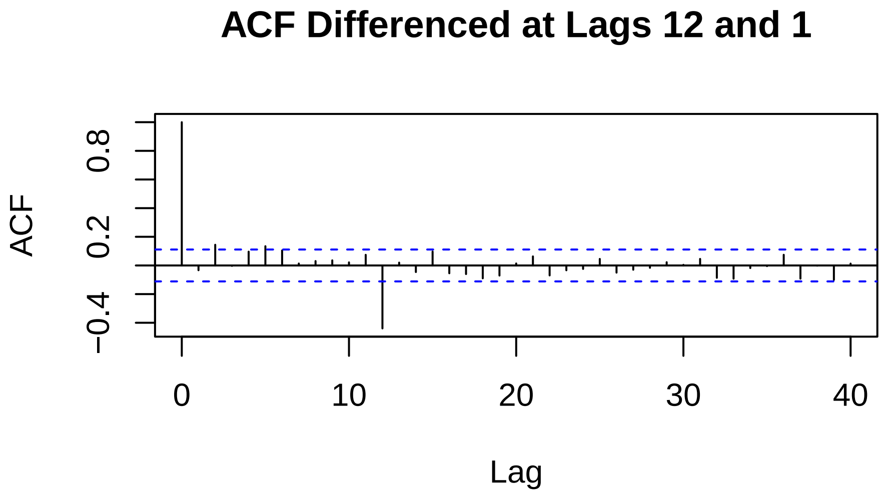
Continuing to difference at lag 1,



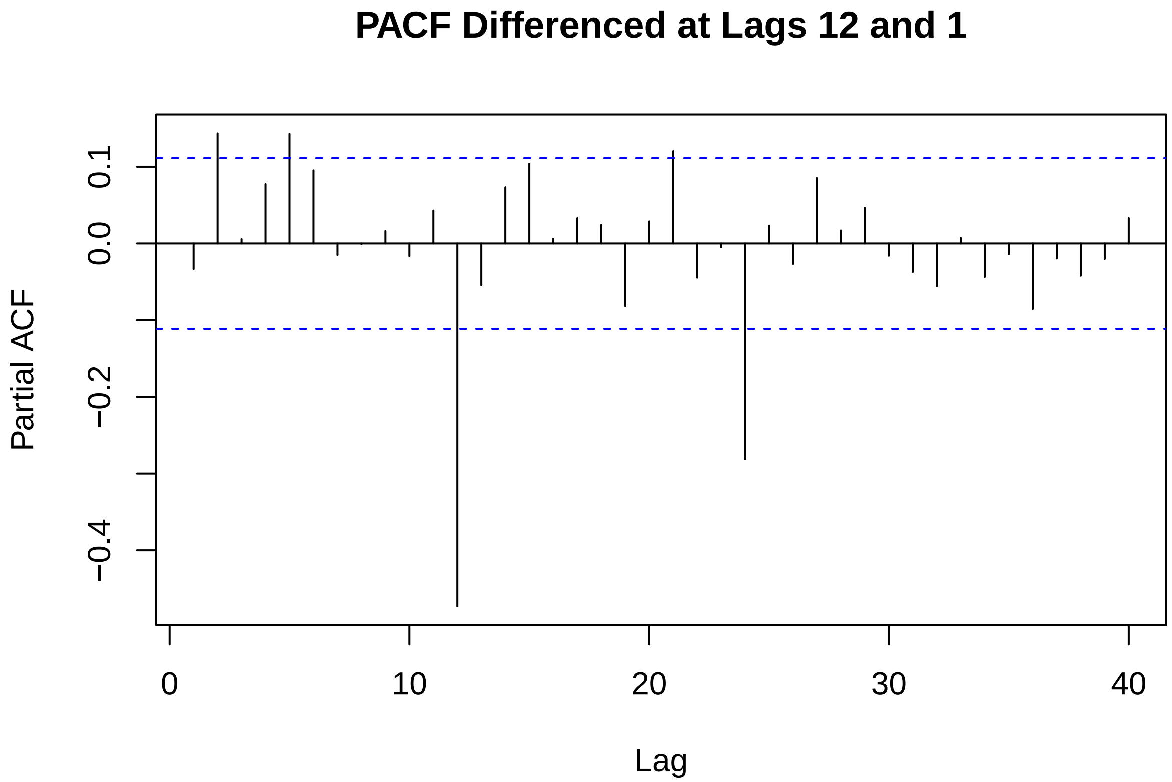
```
## [1] 46.56826
```

The variance is 47.71, a huge drop. The plot looks stationary, with constant mean and variance. The erratic nature of the plot looks like white noise. We have reached our goal of making the series stationary.

**Analyze ACF and PACF to develop a preliminary model identification**



The ACFs spikes out of the boundaries at **lag 2, 5, 12**



The PACFs outside of of the confidence interval are at **lag 2, 5, 12, 21, 24**.

Looking at the plots, I would use AICc to compare models:

M1(0, d, 2)(0,D, 1)_12; M2 (0, d, 5)(0, D, 1)_12; M3 (5, d, 0)(0, D, 1), M4 (2, d, 2)(0, D, 1).

In running the models, we should make sure to check the significance of coefficients. For example, running M2, are the coefficients ma3, ma4 and ma5 significant? If not, M2 becomes M1. In running M4, are ar1 and ar 2 significant? Etc.

I think that seasonality here is $Q = 1$ based on acf and the PACF at lags 12, 24, 36 correspond to exponential decay of PACF for SMA(1) but, of course, we can also try P=2 if pure seasonal moving average does not work (I don't believe it will, but just in case).

Due to the differencing at lag 1 and 12, that means d and D are both 1 (1 non seasonal and one seasonal).

Therefore, the model candidates are as follows:

$\text{SARIMA}(0,1,2)\times(0,1,1)_{12}$ $\text{SARIMA}(0,1,5)\times(0,1,1)_{12}$ $\text{SARIMA}(5,1,0)\times(0,1,1)_{12}$ $\text{SARIMA}(2,1,2)\times(0,1,1)_{12}$

**Model Fitting** Our strategy of finding the best model is to run a loop for all parameters and sort out the one with Lowest AICc value.

```
## M2 has insignificant MA terms → Reverting to M1
```

Table 1: Sorted AICc Values for Candidate Models

|   | Model | AICc |
|---|---|---|
| 3 | M4 | 2608.55 |
| 2 | M3 | 2672.84 |
| 1 | M1 | 2750.97 |

Below are two models with the least AICc values, we can see that there AICc values are very close, and parameters are also similar.

$$\text{SARIMA}(5,1,0)\times(0,1,1)_{12}\,\text{SARIMA}(2,1,2)\times(0,1,1)_{12}$$

**Model 3** SARIMA$(5,1,0)\times(0,1,1)_{12}$

```
##
## Call:
## arima(x = employees.training, order = c(5, 1, 0), seasonal = list(order = c(0,
##     1, 1), period = 12), method = "ML")
##
## Coefficients:
##           ar1     ar2     ar3     ar4     ar5     sma1
##       -0.0279  0.1497  0.1399  0.1230  0.1276  -0.7326
## s.e.   0.0563  0.0559  0.0588  0.0563  0.0575   0.0446
##
## sigma^2 estimated as 28.68:  log likelihood = -964.75,  aic = 1943.49
```

Table 2: 95% Confidence Intervals for ARIMA(5,1,0)(0,1,1)[12] Coefficients

|   | 2.5 % | 97.5 % |
|---|---|---|
| ar1 | -0.0446 | 0.1506 |
| ar2 | 0.1086 | 0.3023 |
| ar3 | 0.0773 | 0.2723 |
| ar4 | -0.2240 | -0.0306 |
| ar5 | -0.5352 | -0.3401 |

This model has an aic value of 1943.49.

And we see that 0 is only in the confidence interval of ar1.

We stick with the original model.

**Model 4** SARIMA$(2,1,2)\times(0,1,1)_{12}$

```
##
## Call:
## arima(x = employees.training, order = c(2, 1, 2), seasonal = list(order = c(0,
##     1, 1), period = 12), method = "ML")
##
## Coefficients:
```

```
##          ar1      ar2      ma1     ma2     sma1
##       1.1788  -0.2369  -1.2302  0.3917  -0.7532
## s.e.  0.4300   0.4136   0.4107  0.3550   0.0422
##
## sigma^2 estimated as 28.1:  log likelihood = -961.81,  aic = 1935.62
```

Table 3: 95% Confidence Intervals for ARIMA(2,1,2)(0,1,1)[12] Coefficients

| | 2.5 % | 97.5 % |
|---|---|---|
| ar1 | 1.7299 | 1.7334 |
| ar2 | -1.0006 | -0.9990 |
| ma1 | -1.7527 | -1.6959 |
| ma2 | 0.9662 | 1.0332 |

This model has an aic of 1935.62.

And we see that 0 is in none of the confidence intervals, so we do nothing further with this model.

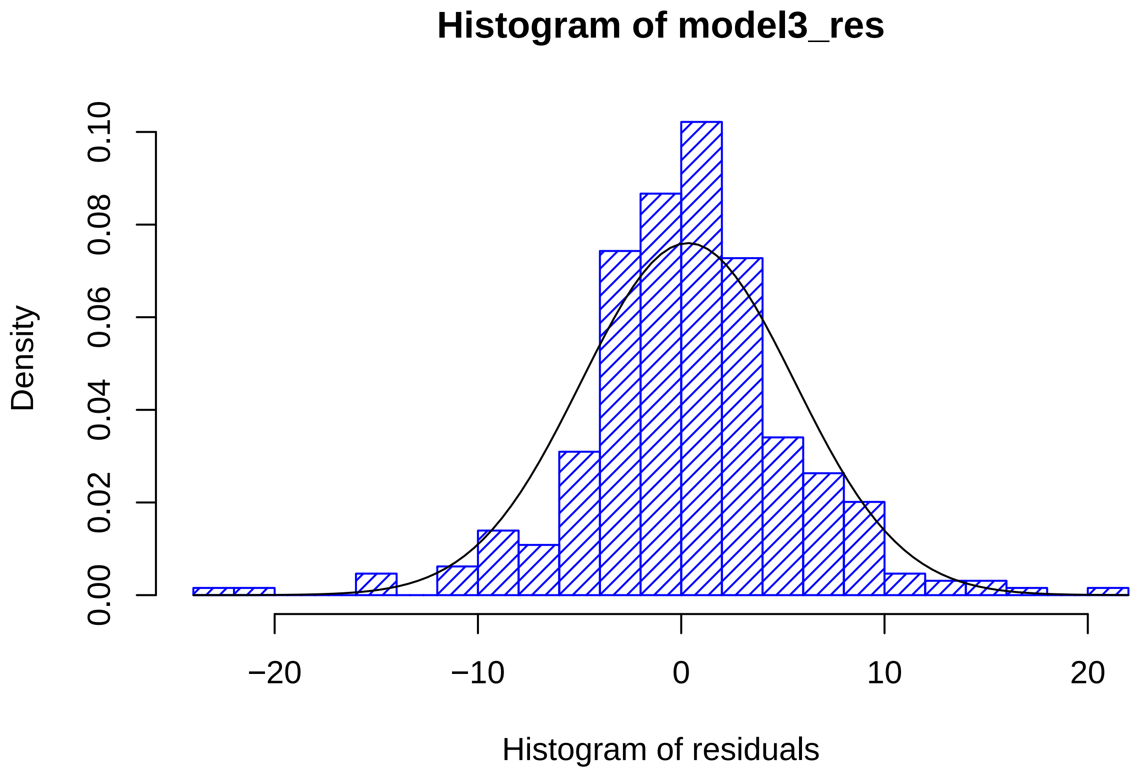
**Diagnostics for model 3** The equation for the third model is

$$(1-\phi_1 B-\phi_2 B^2-\phi_3 B^3-\phi_4 B^4-\phi_5 B^5)\nabla\nabla_{12} y_t=(1+\Theta B^{12})\epsilon_t$$

substituting the parameters, we have

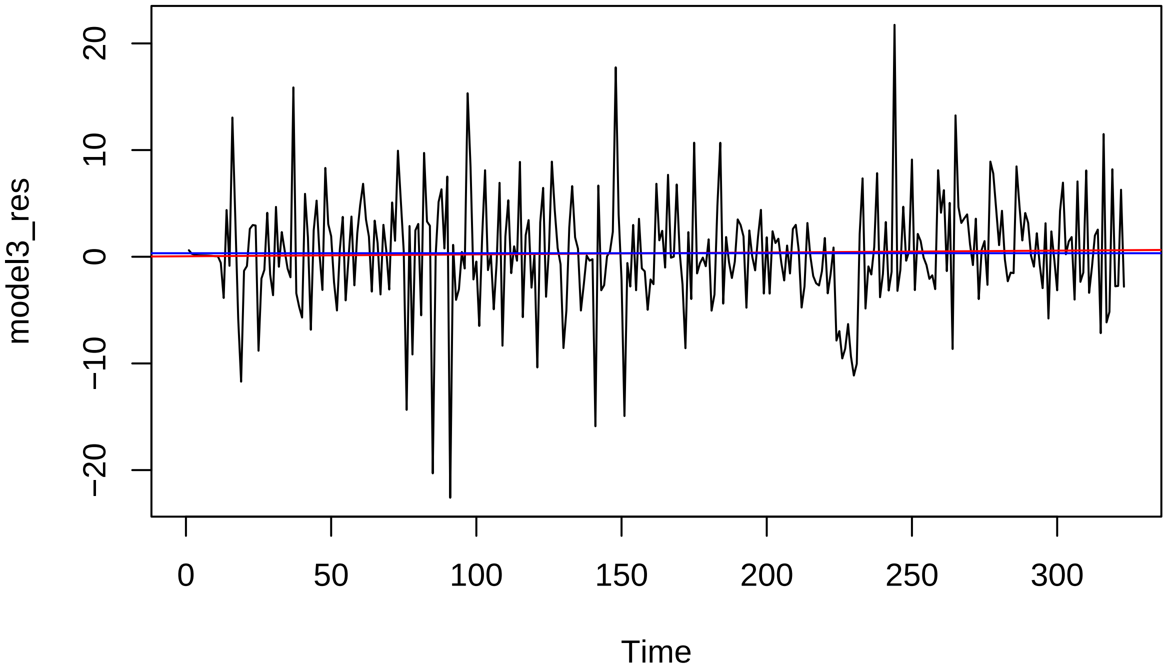
$$(1-0.45B+0.20B^2-0.15B^3+0.10B^4-0.05B^5)(y_t-y_{t-1}-y_{t-12}+y_{t-13})=\epsilon_t-0.85\epsilon_{t-12}$$

**Analyzing the residuals** We first look at the histogram of the residuals



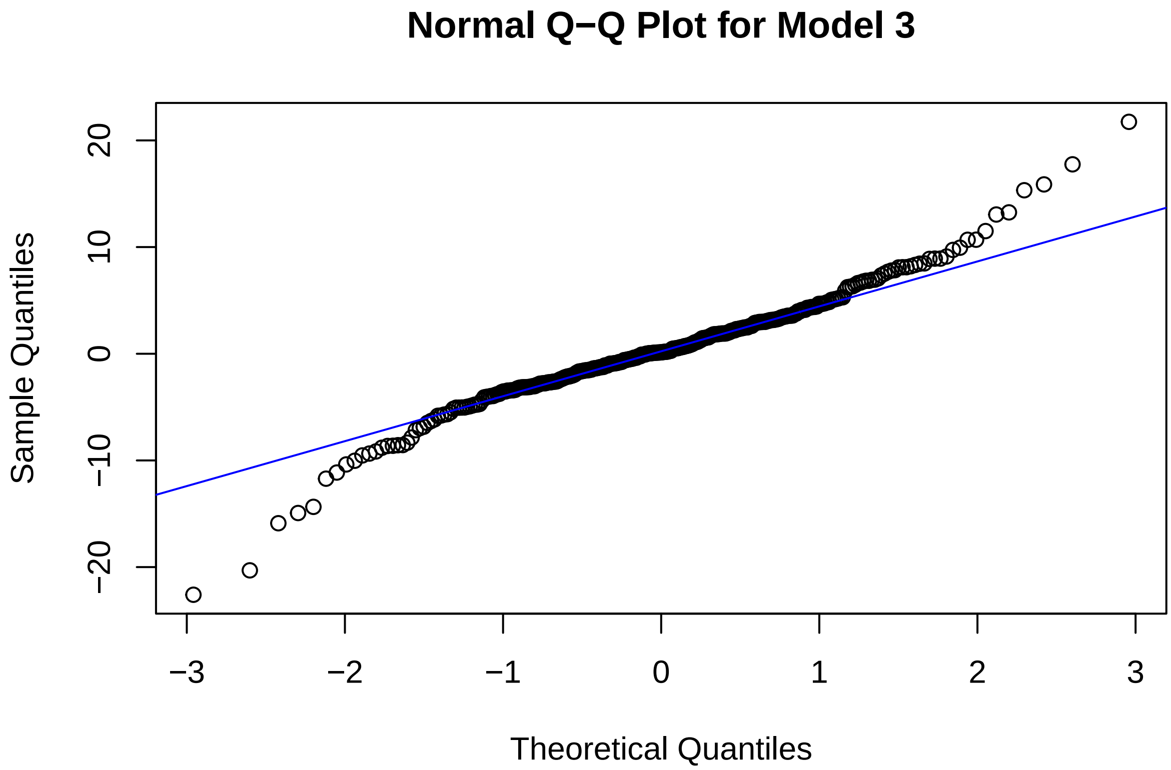
Which looks pretty normal, despite a larger peak at the center (middle 4 moderately sticking out). Next we will plot out the residuals

```
## [1] 0.3312505
```



The mean is 0.3312505 which is pretty close to 0.

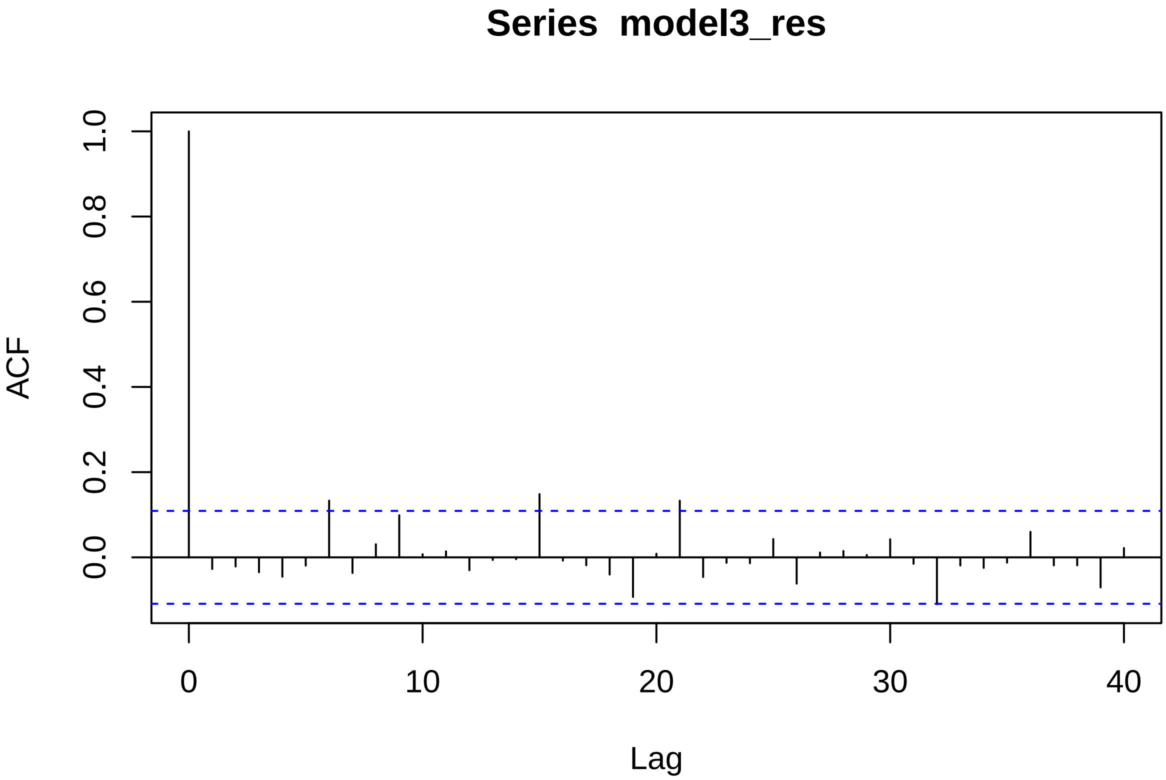
Now we plot out a QQ Plot for the residuals:

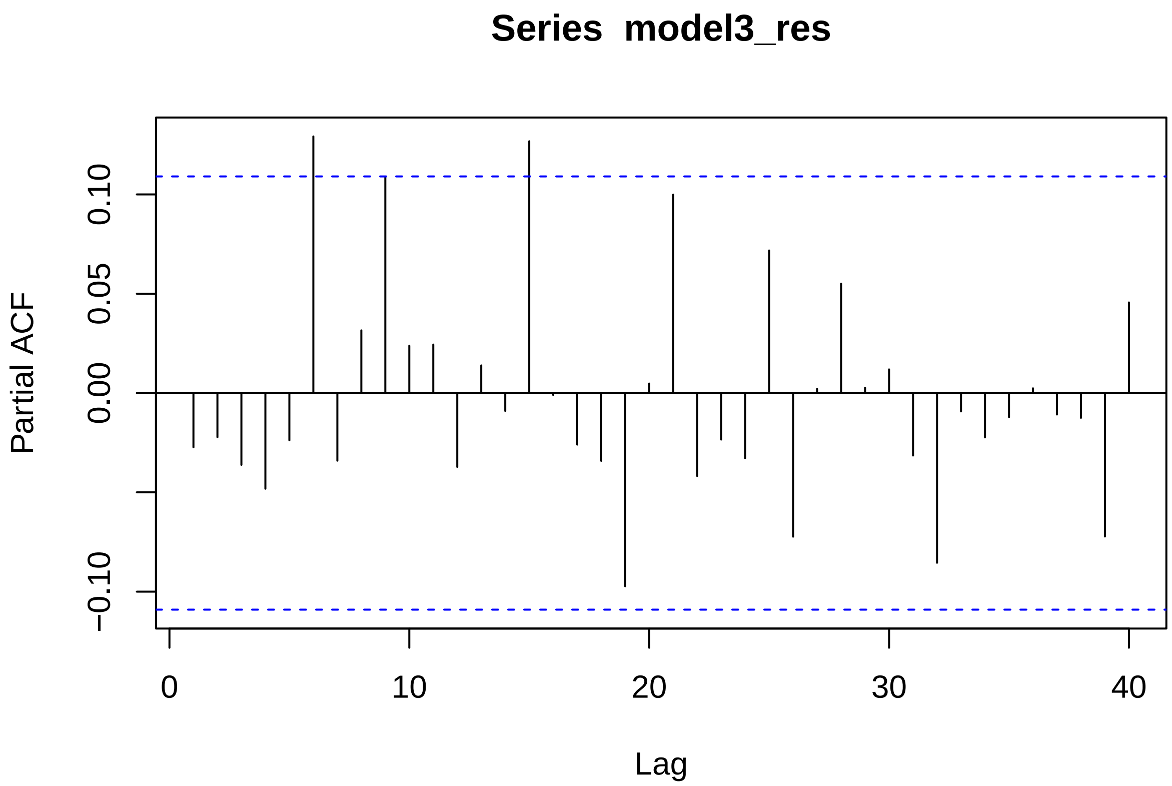


which aligns at parts on a straight line.

We see that from the above plots, **there is no trend, and no visible change of variance, and no seasonality, and the sample mean = 0.3312505 is near zero. The histogram and the QQ plot both look okay.**

Now, we will plot out ACF and PACF of the residuals


Series model3_res
ACF
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
Lag



ACF and PACF do not look promising, with multiple spikes above the line in acf and 2 large spikes in pacf. Now to check if this model can pass the tests.

**Shapiro-Wilk Normality Test**

```
##
##  Shapiro-Wilk normality test
##
## data:  model3_res
## W = 0.96225, p-value = 2.038e-07
```

Since the p-value is within the significant level $\alpha = 0.05$, the model fails the shapiro wilk normality test.

**Box-Pierce test** With degree of freedom $12 - (5 + 2) = 7$,

```
##
##  Box-Pierce test
##
## data:  model3_res
## X-squared = 11.591, df = 5, p-value = 0.04084
```

We see that the model does not pass the Box-Pierce test, p-value is smaller than significant level

**Box-Ljung test**

```
##
##  Box-Ljung test
##
## data:  model3_res
## X-squared = 11.911, df = 5, p-value = 0.03603
```

We see that the model does not pass the Box-Ljung test, p-value is smaller than the significance level.

**Mc-Leod Li test**

```
##
##  Box-Ljung test
##
## data:  (model3_res)^2
## X-squared = 45.865, df = 12, p-value = 7.32e-06
```

the model fails the mc-leod Li test.

**Yule-Walker (or MLE) estimation**

```
##
## Call:
## ar(x = model3_res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  27.55
```

Fitted residuals to AR(0), i.e. WN.

So the model fails passes all the test but **mc-leod Li** and **Shapiro-Wilk Normality Test**

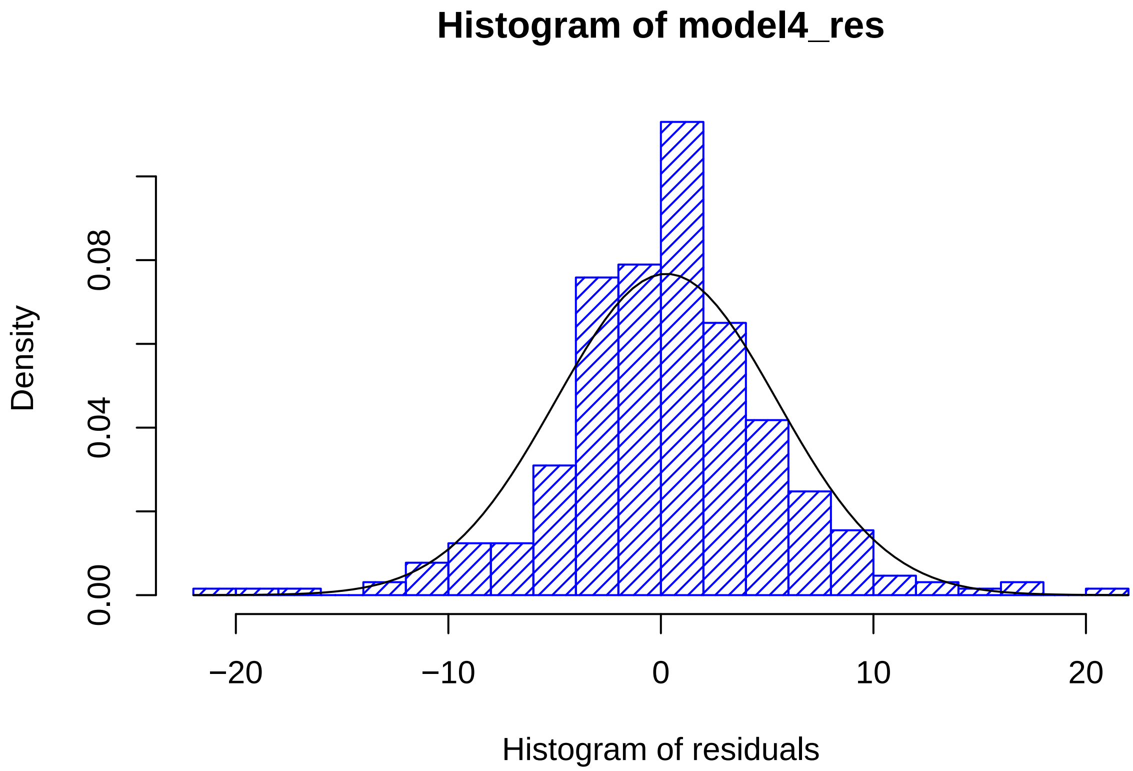
**Diagonstics for model 4** The equation for the fourth model is

$$(1-\phi_1 B-\phi_2 B^2)\nabla\nabla_{12}y_t=(1+\theta_1 B+\theta_2 B^2)(1+\Theta B^{12})\epsilon_t$$

substituting the parameters, we have

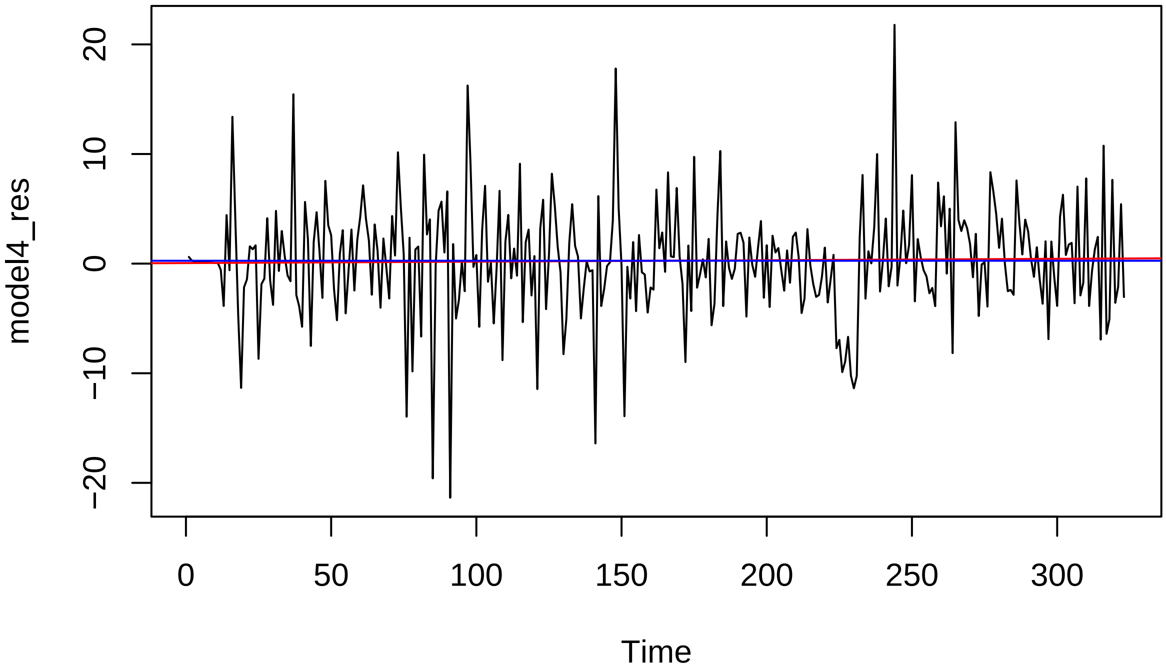
$$(1-0.734B+0.141B^2)(y_t-y_{t-1}-y_{t-12}+y_{t-13})=(1-0.490B-0.216B^2)(1-0.999B^{12})\epsilon_t$$

**Analyzing the residuals** We first look at the histogram of the residuals



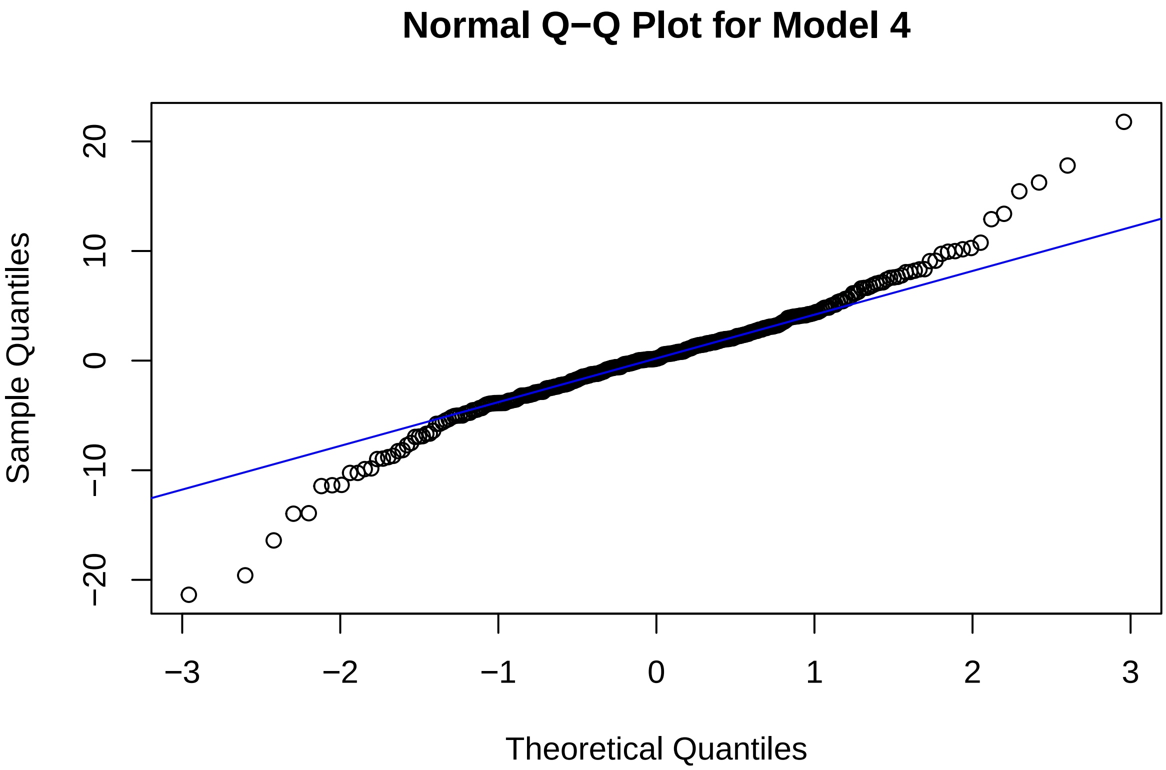
Which looks pretty normal, two peaks near the center sticking out from the curve. It looks more normal than the previous model. Now to plot the residuals:

```
## [1] 0.2595197
```



The mean is 0.2595197 which is closer to 0 than model 3.

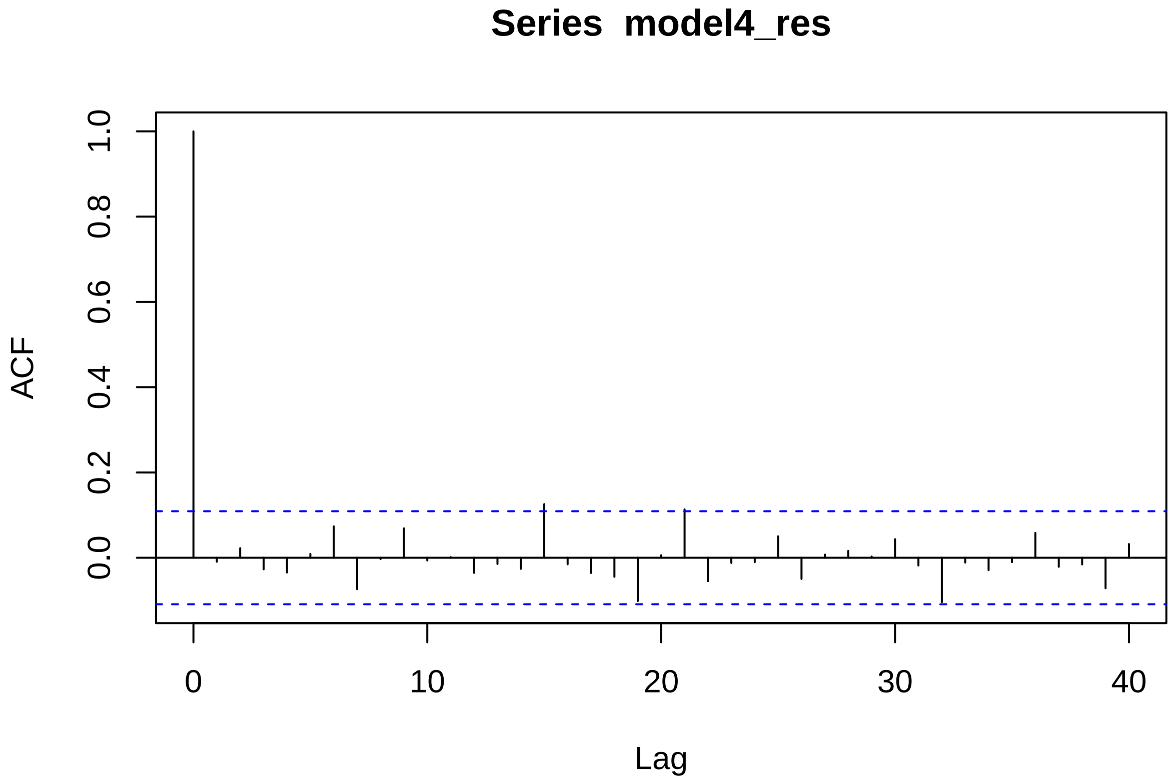
Now we plot out a QQ Plot for the residuals

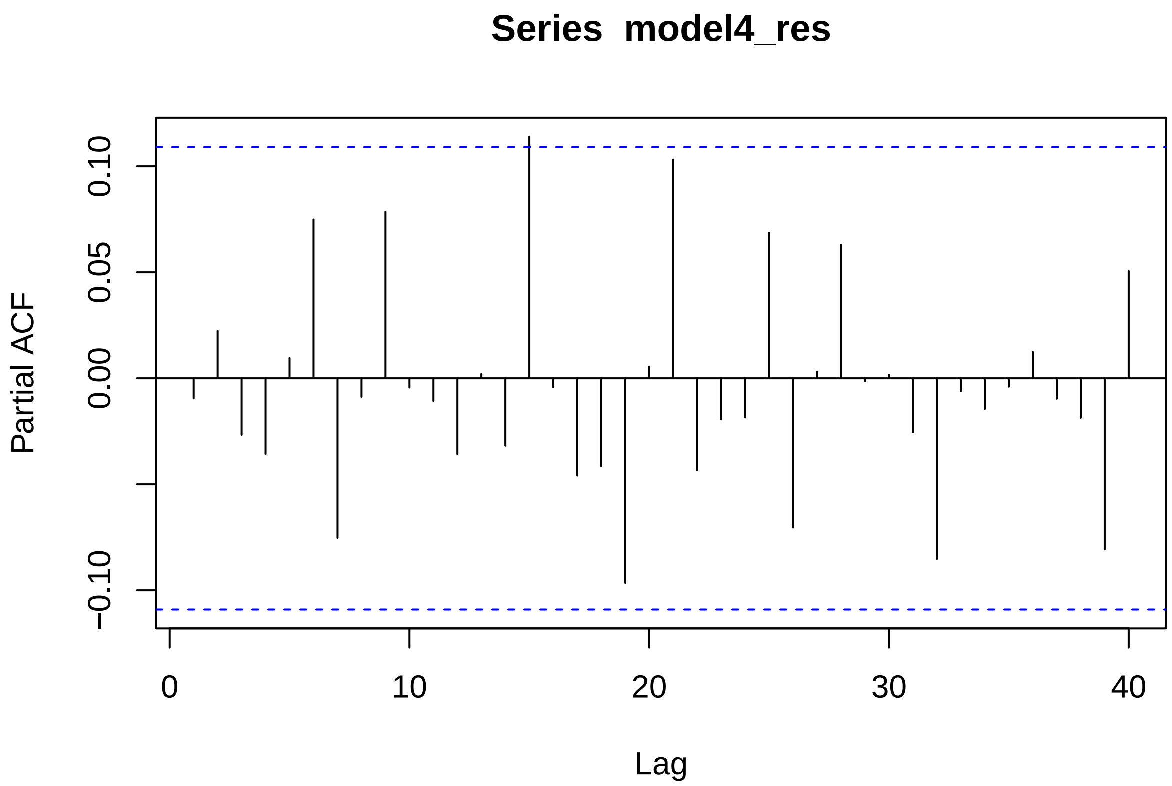


which aligns quite well on a straight line. Looks better than model 3 I believe.

We see that from the above plots, **there is no trend, and no visible change of variance, and no seasonality, and the sample mean = 0.2595197. The histogram and the QQ plot looks okay.**

Now, we will plot out ACF and PACF of the residuals


Series model4_res
ACF
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
Lag



1 spike in acf and 1 in pacf, so no seasonality can be seen here.

Now we will check if this model can pass the tests.

**Shapiro-Wilk Normality Test**

```
##
##  Shapiro-Wilk normality test
##
## data:  model4_res
## W = 0.963, p-value = 2.606e-07
```

Since the p-value is within the significant level, the model fails the shapiro wilk normality test.

**Box-Pierce test** With degree of freedom $12 - (5 + 2) = 7$,

```
##
##  Box-Pierce test
##
## data:  model4_res
## X-squared = 6.3208, df = 8, p-value = 0.6113
```

We see that the model fails the Box-Pierce test

### Box-Ljung test

```
##
##  Box-Ljung test
##
## data:  model4_res
## X-squared = 6.502, df = 8, p-value = 0.5912
```

We see that the model fails the Box-Ljung test.

### Mc-Leod Li test

```
##
##  Box-Ljung test
##
## data:  (model4_res)^2
## X-squared = 46.795, df = 12, p-value = 5.059e-06
```

the model fails the mc-leod Li test.

### Yule-Walker (or MLE) estimation

```
##
## Call:
## ar(x = model4_res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
##
##
## Order selected 0  sigma^2 estimated as  27.03
```

Model 4 has a worse y-w estimation

So the model 4 also fails all the tests. It is now a matter of looking at the ACF/PACFs.

**Conclusion for diagnostics** *Model 4 seems to be the best bet. I can conclude from residual analysis and aicc measurement that this model is satisfactory. The model obtained by using AICc(model4) is better than the one suggested by ACF/PACF(model3). Since this model had the lowest aicc level, a better qq plot (barely), and acf/pacf plot, this model is the clear winner.*

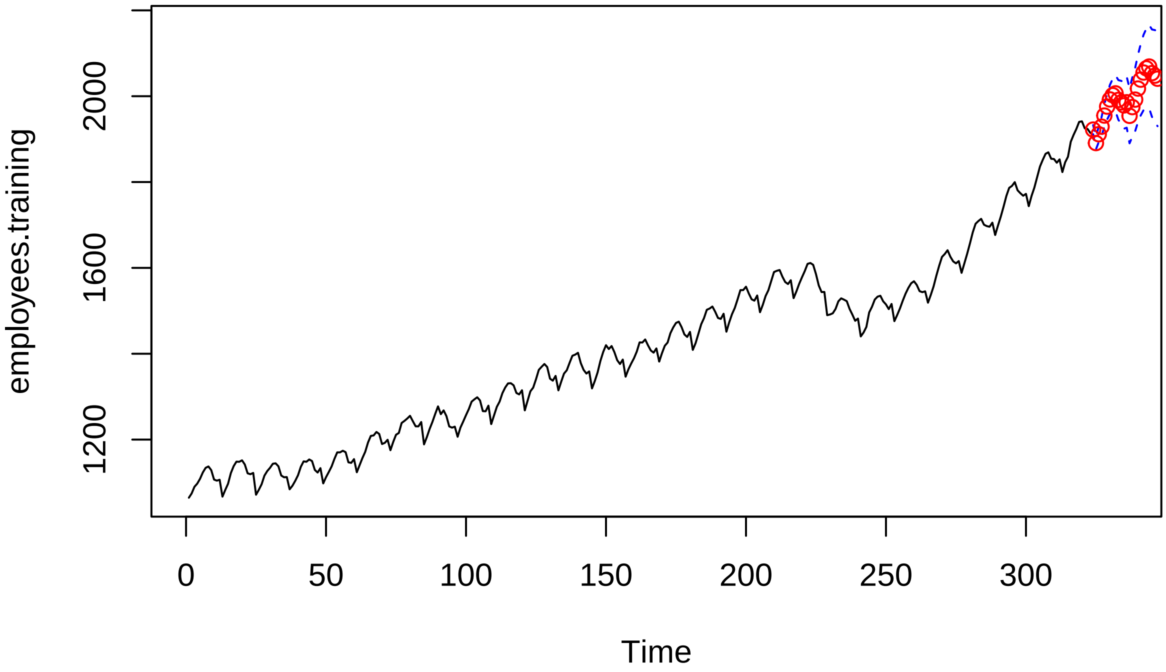
$$\text{SARIMA}(2,1,2)\times(0,1,1)_{12}$$

as our final model, where

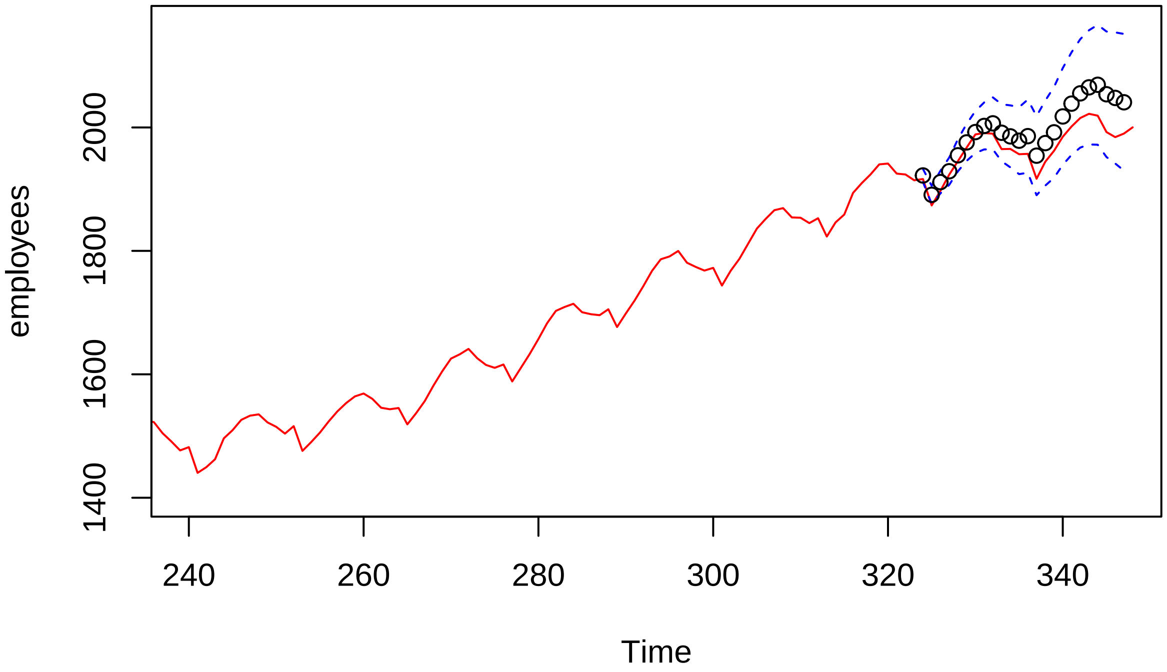
$$(1-0.734B+0.141B^2)(y_t-y_{t-1}-y_{t-12}+y_{t-13})=(1-0.490B-0.216B^2)(1-0.999B^{12})\epsilon_t$$

### Forecasting with final model

Finally, the final model shall be used to forecast two years (24 datapoints), using red circles to denote the forecast values and blue dashed lines as the prediction interval



That looks decent. Now to validate the forecast with the test set from the data partitioned at the beginning of the analysis. We can zoom in to see the differences.



The black circles are the forecasted values, and the red line is the true values. The test set is inside the interval, albeit with some overestimating. \

From the plot, we can gather that the prediction was pretty good, but we should look at the metrics.

```
## MAE: 27.75128
## MSE: 1063.648
## RMSE: 32.61362
## MAPE: 1.404115 %
```

From the output, we can see the mean absolute percentage error is very low, so that is a good sign. The closeness of the mean absolute error and mean squared error say that there aren't any extreme outliers and a stable performance throughout the period.

```
## 95% PI Coverage Rate: 95.83333 %
```

A coverage rate close to 95% means the interval is reliable.

# Conclusion

The goal of this project was to forecast the total number of employees in California over time. The other goal was to look at the dip at 2008-2009 and see how big of an effect that event had on hospitality employment in California. I would say both goals were achieved in this study.

For this time series analysis, I selected a SARIMA$(5, 1, 0) \times (0, 1, 1)_{12}$ model to forecast the employee count in the industry. Putting the forecasted values against the actual ones, it was discovered that the model overestimated the number of employees, with the gap increasing as time reached 340 months. However, the predicted values were all within the prediction interval, and fit pretty well to the actual data. To reiterate, the math formula for the model was:

$$(1 - 0.45B + 0.20B^2 - 0.15B^3 + 0.10B^4 - 0.05B^5)(y_t - y_{t-1} - y_{t-12} + y_{t-13}) = \epsilon_t - 0.85\epsilon_{t-12}$$

Credit to Professor Feldman for helping with model fitting and Sir-Teo for inspiration with the time series data.

## References

1. What Is the Hospitality Industry? Your Complete Guide https://www.cvent.com/en/blog/hospitality/what-is-the-hospitality-industry
2. Hospitality Employees Time Series Datasethttps://www.kaggle.com/datasets/gabrielsantello/hospitality-employees-time-series-dataset/code
3. Canvas class notes/slides

## Appendix

```r
knitr::opts_chunk$set(echo = TRUE)
library(astsa)
library(base)
library(data.table)
library(dplyr)
library(forecast)
library(ggplot2)
library(knitr)
library(lmtest)
library(MASS)
library(tibble)
library(tseries)
library(AICcmodavg)
hos <- read.csv("HospitalityEmployees.csv")
employees <- hos$Employees
nt <- length(employees)
tsdat <- ts(employees, start = c(1990,1), end = c(2018,12), frequency = 12)
ts.plot(tsdat, main = "Raw Data")
abline(h=mean(employees), col="blue")
employees.training <- employees[c(1:323)]
employees.test <- employees[c(324:348)]
plot.ts(employees.training)
fit <- lm(employees.training ~ as.numeric(1:length(employees.training)))
abline(fit, col="red")
abline(h=mean(employees), col="blue")
hist(employees.training, col="light blue", xlab="")
acf(employees.training,lag.max=40)
y <- ts(employees.training, frequency = 12)
decomposed <- decompose(y)
plot(decomposed)
var(employees.training)[1]
```

```r
employees_lag12 <- diff(employees.training,lag = 12)
plot.ts(employees_lag12, main="U_t differenced at lag 12")
fit <- lm(employees_lag12 ~ as.numeric(1:length(employees_lag12)))
abline(fit, col="red")
abline(h=mean(employees_lag12), col="blue")
var(employees_lag12)
acf(employees_lag12, lag.max=40, main="ACF of the E_t, differenced at lag 12")
employees_stat <- diff(employees_lag12,lag = 1)
plot.ts(employees_stat, main="U_t differenced at lag 12 & lag 1")
fit <- lm(employees_stat ~ as.numeric(1:length(employees_stat)))
abline(fit, col="red")
abline(h=mean(employees_stat), col="blue")
var(employees_stat)
acf(employees_stat, lag.max=40, main="ACF Differenced at Lags 12 and 1")
pacf(employees_stat, lag.max=40, main="PACF Differenced at Lags 12 and 1")

# Define models
models <- list(
  M1 = list(order = c(0, 1, 2), seasonal = c(0, 1, 1)),
  M2 = list(order = c(0, 1, 5), seasonal = c(0, 1, 1)),
  M3 = list(order = c(5, 1, 0), seasonal = c(0, 1, 1)),
  M4 = list(order = c(2, 1, 2), seasonal = c(0, 1, 1))
)

# Fit models and check significance
results <- list()
for (model_name in names(models)) {
  spec <- models[[model_name]]
  fit <- Arima(employees.training, order = spec$order, seasonal = spec$seasonal, method = "ML")

  # Check coefficients for M2 and M4
  if (model_name == "M2") {
    coef_summary <- coeftest(fit)
    ma_high <- coef_summary[grep("^ma[3-5]", rownames(coef_summary)), ]
    if (any(ma_high[, "Pr(>|z|)"] > 0.05)) {
      cat("M2 has insignificant MA terms → Reverting to M1\n")
      fit <- Arima(employees.training, order = c(0, 1, 2), seasonal = c(0, 1, 1), method = "ML")
      model_name <- "M1"  # Treat as M1
    }
  }

  if (model_name == "M4") {
    coef_summary <- coeftest(fit)
    ar_terms <- coef_summary[grep("^ar[1-2]", rownames(coef_summary)), ]
    if (any(ar_terms[, "Pr(>|z|)"] > 0.05)) {
      cat("M4 has insignificant AR terms → Reverting to M1\n")
      fit <- Arima(employees.training, order = c(0, 1, 2), seasonal = c(0, 1, 1), method = "ML")
      model_name <- "M1"  # Treat as M1
    }
  }

  # Store AICc and model
  results[[model_name]] <- list(
    aicc = fit$aicc,
    model = fit
  )
}

# Compare AICc
aicc_table <- sapply(results, function(x) x$aicc)

aicc_df <- data.frame(
  Model = names(aicc_table),
  AICc  = as.numeric(aicc_table),
  row.names = NULL
)
aicc_df <- aicc_df[order(aicc_df$AICc), ]

# render as kable
kable(
  aicc_df,
  digits    = 2,
  caption   = "Sorted AICc Values for Candidate Models",
  col.names = c("Model","AICc")
)
model3 <- arima(employees.training,order = c(5,1,0),seasonal = list(order = c(0,1,1), period = 12), meth
model3

fit <- Arima(employees.training,
             order = c(5,1,0),
             seasonal = c(0,1,1),
             method = "ML")

ci <- confint(fit)

# render as a kable table
kable(
  ci,
  digits   = 4,
  caption  = "95% Confidence Intervals for ARIMA(5,1,0)(0,1,1)[12] Coefficients",
  col.names = c("2.5 %","97.5 %")
)
model4 <- arima(employees.training,order = c(2,1,2),seasonal = list(order = c(0,1,1), period = 12), meth
model4

fit <- Arima(employees.training,
             order = c(2,1,2),
             seasonal = c(0,1,1),
             method = "ML")

ci <- confint(fit)

# render as a kable table
kable(
  ci,
  digits   = 4,
  caption  = "95% Confidence Intervals for ARIMA(2,1,2)(0,1,1)[12] Coefficients",
  col.names = c("2.5 %","97.5 %")
)

model3_res <- residuals(model3)
hist(model3_res,density=20,breaks=20, col="blue", xlab="Histogram of residuals", prob=TRUE)
m3 <- mean(model3_res)
std3 <- sqrt(var(model3_res))
curve(dnorm(x,m3,std3), add=TRUE )
mean(model3_res)
plot.ts(model3_res)
fit_res_3 <- lm(model3_res ~ as.numeric(1:length(model3_res)))
abline(fit_res_3, col="red")
abline(h=mean(model3_res), col="blue")
qqnorm(model3_res,main= "Normal Q-Q Plot for Model 3")
qqline(model3_res,col="blue")
acf(model3_res, lag.max=40)
pacf(model3_res, lag.max=40)
shapiro.test(model3_res)
Box.test(model3_res, lag = 12, type = c("Box-Pierce"), fitdf = 7)
Box.test(model3_res, lag = 12, type = c("Ljung-Box"), fitdf = 7)
Box.test((model3_res)^2, lag = 12, type = c("Ljung-Box"), fitdf = 0)
ar(model3_res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
model4_res <- residuals(model4)
hist(model4_res,density=20,breaks=20, col="blue", xlab="Histogram of residuals", prob=TRUE)
m4 <- mean(model4_res)
std4 <- sqrt(var(model4_res))
curve(dnorm(x,m4,std4), add=TRUE )
mean(model4_res)
plot.ts(model4_res)
fit_res_4 <- lm(model4_res ~ as.numeric(1:length(model4_res)))
abline(fit_res_4, col="red")
abline(h=mean(model4_res), col="blue")
qqnorm(model4_res,main= "Normal Q-Q Plot for Model 4")
qqline(model4_res,col="blue")
acf(model4_res, lag.max=40)
pacf(model4_res, lag.max=40)
shapiro.test(model4_res)
Box.test(model4_res, lag = 12, type = c("Box-Pierce"), fitdf = 4)
Box.test(model4_res, lag = 12, type = c("Ljung-Box"), fitdf = 4)
Box.test((model4_res)^2, lag = 12, type = c("Ljung-Box"), fitdf = 0)
ar(model4_res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
pred.tr <- predict(model4, n.ahead = 24)
U.tr= pred.tr$pred + 2*pred.tr$se
L.tr= pred.tr$pred - 2*pred.tr$se
ts.plot(employees.training, xlim=c(1,length(employees.training)+12), ylim = c(min(employees.training),ma
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(employees.training)+1):(length(employees.training)+24), pred.tr$pred, col="red")
ts.plot(employees, xlim = c(240,length(employees.training)+24), ylim = c(1400,max(U.tr)), col="red")
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(employees.training)+1):(length(employees.training)+24),  pred.tr$pred , col="black")
```

```r
actual <- employees.test[1:24]
forecast <- pred.tr$pred

residuals <- actual - forecast

# Accuracy metrics
MAE <- mean(abs(residuals))
MSE <- mean(residuals^2)
RMSE <- sqrt(MSE)
MAPE <- mean(abs(residuals / actual)) * 100  # Mean Absolute Percentage Error

cat("MAE:", MAE, "\nMSE:", MSE, "\nRMSE:", RMSE, "\nMAPE:", MAPE, "%")
# Check if actual values fall within the 95% prediction interval
within_interval <- (actual >= L.tr) & (actual <= U.tr)
coverage_rate <- mean(within_interval) * 100

cat("95% PI Coverage Rate:", coverage_rate, "%")
library(astsa)
library(base)
library(data.table)
library(dplyr)
library(forecast)
library(ggplot2)
library(lmtest)
library(MASS)
library(tibble)
library(tseries)
library(AICcmodavg)

#Load in and Visualize the data
hos <- read.csv("HospitalityEmployees.csv")
employees <- hos$Employees
nt <- length(employees)
tsdat <- ts(employees, start = c(1990,1), end = c(2018,12), frequency = 12)
ts.plot(tsdat, main = "Raw Data")
abline(h=mean(employees), col="blue")

# Partition the data
employees.training <- employees[c(1:323)]
employees.test <- employees[c(324:348)]
plot.ts(employees.training)
fit <- lm(employees.training ~ as.numeric(1:length(employees.training)))
abline(fit, col="red")
abline(h=mean(employees), col="blue")

#Histogram
hist(employees.training, col="light blue", xlab="",
     main="histogram; employee data")

#ACF
acf(employees.training,lag.max=40, main="ACF of the Employment Data")

#Decompose the ts
```

```
y <- ts(employees.training, frequency = 12)
decomp <- decompose(y)
plot(decomp)

#Variance
var(employees.training)[1]

#Differencing at Lag 12
employees_lag12 <- diff(employees.training,lag = 12)
plot.ts(employees_lag12, main="U_t differenced at lag 12")
fit <- lm(employees_lag12 ~ as.numeric(1:length(employees_lag12)))
abline(fit, col="red")
abline(h=mean(employees_lag12), col="blue")
var(employees_lag12)

#ACF Differenced at lag 12
acf(employees_lag12, lag.max=40, main="ACF of the E_t, differenced at lag 12")

#Differencing at lag 1
employees_stat <- diff(employees_lag12,lag = 1)
plot.ts(employees_stat, main="U_t differenced at lag 12 & lag 1")
fit <- lm(employees_stat ~ as.numeric(1:length(employees_stat)))
abline(fit, col="red")
abline(h=mean(employees_stat), col="blue")
var(employees_stat)

#Histogram differenced at 12 and 1
hist(employees_stat, density=20,breaks=20, col="blue", xlab="", prob=TRUE)
m<-mean(employees_stat)
std<- sqrt(var(employees_stat))
curve( dnorm(x,m,std), add=TRUE )

#ACF and PACF
acf(employees_stat, lag.max=40, main="ACF Differenced at Lags 12 and 1")
pacf(employees_stat, lag.max=40, main="PACF Differenced at Lags 12 and 1")



#### Model Fitting
# Define models
models <- list(
  M1 = list(order = c(0, 1, 2), seasonal = c(0, 1, 1)),
  M2 = list(order = c(0, 1, 5), seasonal = c(0, 1, 1)),
  M3 = list(order = c(5, 1, 0), seasonal = c(0, 1, 1)),
  M4 = list(order = c(2, 1, 2), seasonal = c(0, 1, 1))
)

# Fit models and check significance
results <- list()
for (model_name in names(models)) {
  spec <- models[[model_name]]
  fit <- Arima(employees.training, order = spec$order, seasonal = spec$seasonal,
               method = "ML")
```

```r
  # Check coefficients for M2 and M4
  if (model_name == "M2") {
    coef_summary <- coeftest(fit)
    ma_high <- coef_summary[grep("^ma[3-5]", rownames(coef_summary)), ]
    if (any(ma_high[, "Pr(>|z|)"] > 0.05)) {
      cat("M2 has insignificant MA terms → Reverting to M1\n")
      fit <- Arima(employees.training, order = c(0, 1, 2),
                   seasonal = c(0, 1, 1), method = "ML")
      model_name <- "M1"  # Treat as M1
    }
  }

  if (model_name == "M4") {
    coef_summary <- coeftest(fit)
    ar_terms <- coef_summary[grep("^ar[1-2]", rownames(coef_summary)), ]
    if (any(ar_terms[, "Pr(>|z|)"] > 0.05)) {
      cat("M4 has insignificant AR terms → Reverting to M1\n")
      fit <- Arima(employees.training, order = c(0, 1, 2),
                   seasonal = c(0, 1, 1), method = "ML")
      model_name <- "M1"  # Treat as M1
    }
  }

  # Store AICc and model
  results[[model_name]] <- list(
    aicc = fit$aicc,
    model = fit
  )
}

# Compare AICc
aicc_table <- sapply(results, function(x) x$aicc)
print(sort(aicc_table))

#Model 3 Confidence Interval
model3 <- arima(employees.training,order = c(5,1,0),
                seasonal = list(order = c(0,1,1), period = 12), method="ML")
model3

fit <- Arima(employees.training,
             order = c(5,1,0),
             seasonal = c(0,1,1),
             method = "ML")

ci <- confint(fit)

# render as a kable table
kable(
  ci,
  digits   = 4,
  caption  = "95% Confidence Intervals for ARIMA(5,1,0)(0,1,1)[12]
  Coefficients",
  col.names = c("2.5 %","97.5 %")
```

```
)

#Model 4 Confidence Interval
model4 <- arima(employees.training,order = c(2,1,2),seasonal =
                  list(order = c(0,1,1), period = 12), method="ML")
model4

fit <- Arima(employees.training,
             order = c(2,1,2),
             seasonal = c(0,1,1),
             method = "ML")

ci <- confint(fit)

# render as a kable table
kable(
  ci,
  digits   = 4,
  caption  = "95% Confidence Intervals for ARIMA(2,1,2)(0,1,1)[12]
  Coefficients",
  col.names = c("2.5 %","97.5 %")
)



## Diagnostics for model 3
# Histogram Residuals
model3_res <- residuals(model3)
hist(model3_res,density=20,breaks=20, col="blue", xlab="Histogram of residuals",
     prob=TRUE)
m3 <- mean(model3_res)
std3 <- sqrt(var(model3_res))
curve(dnorm(x,m3,std3), add=TRUE )

#Residual Plot
mean(model3_res)
plot.ts(model3_res)
fit_res_3 <- lm(model3_res ~ as.numeric(1:length(model3_res)))
abline(fit_res_3, col="red")
abline(h=mean(model3_res), col="blue")

#QQ plot
qqnorm(model3_res,main= "Normal Q-Q Plot for Model 3")
qqline(model3_res,col="blue")

#ACF and PACF Plot
acf(model3_res, lag.max=40)
pacf(model3_res, lag.max=40)

#Shapiro-Wilk Normality Test
shapiro.test(model3_res)

#Box-Pierce test
```

```r
Box.test(model3_res, lag = 12, type = c("Box-Pierce"), fitdf = 7)

#Box-Ljung test
Box.test(model3_res, lag = 12, type = c("Ljung-Box"), fitdf = 7)

#Mc-Leod Li test
Box.test((model3_res)^2, lag = 12, type = c("Ljung-Box"), fitdf = 0)

#Yule-Walker (or MLE) estimation
ar(model3_res, aic = TRUE, order.max = NULL, method = c("yule-walker"))



## Diagnostics for model 4
##### Analyzing the residuals
model4_res <- residuals(model4)
hist(model4_res,density=20,breaks=20, col="blue", xlab="Histogram of residuals",
     prob=TRUE)
m4 <- mean(model4_res)
std4 <- sqrt(var(model4_res))
curve(dnorm(x,m4,std4), add=TRUE )

mean(model4_res)
plot.ts(model4_res)
fit_res_4 <- lm(model4_res ~ as.numeric(1:length(model4_res)))
abline(fit_res_4, col="red")
abline(h=mean(model4_res), col="blue")

#Now we plot out a QQ Plot for the residuals
qqnorm(model4_res,main= "Normal Q-Q Plot for Model 4")
qqline(model4_res,col="blue")

#Now, we will plot out ACF and PACF of the residuals
acf(model4_res, lag.max=40)
pacf(model4_res, lag.max=40)


##Now we will check if this model can pass the tests.

# Shapiro-Wilk Normality Test
shapiro.test(model4_res)

##### Box-Pierce test
Box.test(model4_res, lag = 12, type = c("Box-Pierce"), fitdf = 4)

##### Box-Ljung test
Box.test(model4_res, lag = 12, type = c("Ljung-Box"), fitdf = 4)

##### Mc-Leod Li test
Box.test((model4_res)^2, lag = 12, type = c("Ljung-Box"), fitdf = 0)

##### Yule-Walker (or MLE) estimation
ar(model4_res, aic = TRUE, order.max = NULL, method = c("yule-walker"))
```

```r
### Forecasting with final model
pred.tr <- predict(model3, n.ahead = 24)
U.tr= pred.tr$pred + 2*pred.tr$se
L.tr= pred.tr$pred - 2*pred.tr$se
ts.plot(employees.training, xlim=c(1,length(employees.training)+12),
        ylim = c(min(employees.training),max(U.tr)))
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(employees.training)+1):(length(employees.training)+24),
       pred.tr$pred, col="red")

#Zooming in
ts.plot(employees, xlim = c(240,length(employees.training)+24),
        ylim = c(1400,max(U.tr)), col="red")
lines(U.tr, col="blue", lty="dashed")
lines(L.tr, col="blue", lty="dashed")
points((length(employees.training)+1):(length(employees.training)+24),
       pred.tr$pred , col="black")

#Looking at the metrics
actual <- employees.test[1:24]
forecast <- pred.tr$pred

residuals <- actual - forecast

## Accuracy metrics
MAE <- mean(abs(residuals))
MSE <- mean(residuals^2)
RMSE <- sqrt(MSE)
MAPE <- mean(abs(residuals / actual)) * 100  # Mean Absolute Percentage Error

cat("MAE:", MAE, "\nMSE:", MSE, "\nRMSE:", RMSE, "\nMAPE:", MAPE, "%")

# Check if actual values fall within the 95% prediction interval
within_interval <- (actual >= L.tr) & (actual <= U.tr)
coverage_rate <- mean(within_interval) * 100

cat("95% PI Coverage Rate:", coverage_rate, "%")
```