Econ 144 Project 1

Nicholas Cassol-Pawson

April 18, 2024

Contents


```
library(tidyverse)
library(imputeTS)
library(forecast)
```

I. Introduction

This data is the monthly quit rate data for all non-farm industries. The quit rate is the monthly number of quits from a firm divided by the total number of people employed in that firm that month (BLS, Glossary). Quitting is defined as voluntarily leaving a job other than for reasons including retirement and transfers to a separate work site (BLS, Survey). It is important for companies to forecast the monthly percentage change in the quit rate so that they can prepare for unexpected shrinkages in their workforce in the upcoming months by preemptively hiring the correct number of people to replace employees who quit.

II. Results

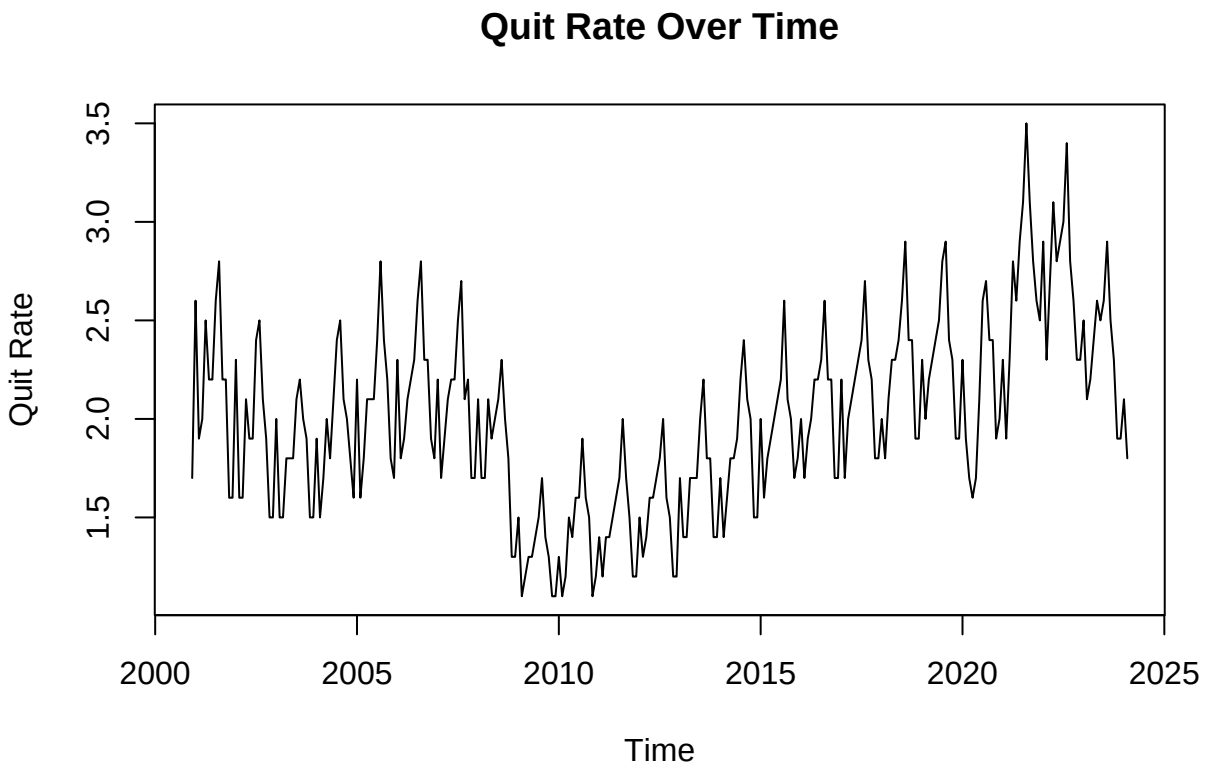
1. Modeling and Forecasting Trend

We first import the data:

```
quits_monthly <- read_csv("QuitsDataMonthlyNonFarm.csv")[[2]] %>% # We read in the data file
  ts(start = c(2000, 12), freq = 12) # We convert the data to a monthly time series
```

(a) Time series plot of data

```
plot(quits_monthly, main = "Quit Rate Over Time", ylab = "Quit Rate") # We plot the quit rate as a func
```



Looking at the plot, we can see that the quit rate appears to display strong seasonality, some levels of cycles, and has a constant trend.

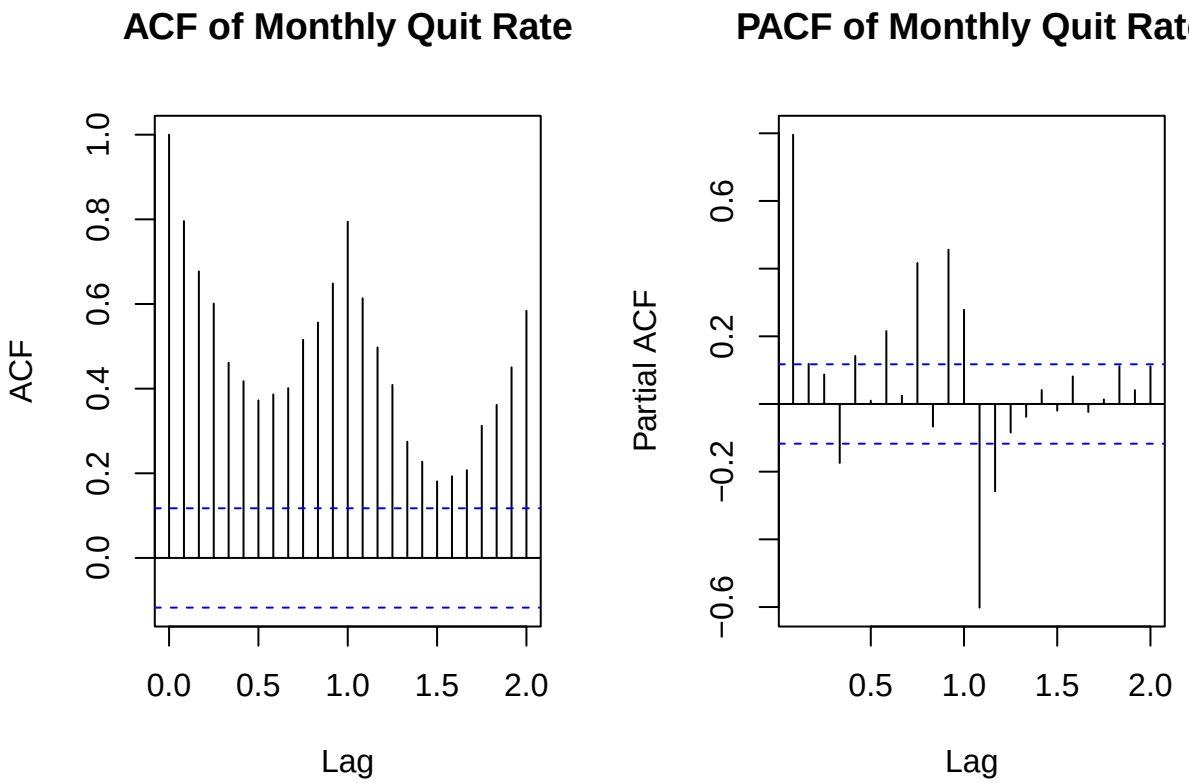
(b) Investigating covariance stationarity

The plot suggests that the data is stationary, but not covariance stationary. The data appear to be mean reverting to a quit rate of about 2% of total employees per month. Even the dip in the quit rate that occurs in late 2010 sees a recovery back to the mean by 2015, which then continues on a cycle past it. However, the variance becomes much larger past 2010, going from a rather small grouping of values close to the mean to a much larger variation in the range of the quit rate.

(c) ACF and PACF

We will now plot the ACF and PACF of the quit rate to get a sense of dynamics in the data.

```
par(mfrow = c(1, 2)) # We set up a graphics environment matrix with one row and two columns
acf(quits_monthly, main = "ACF of Monthly Quit Rate") # We plot the ACF of the quit rate
pacf(quits_monthly, main = "PACF of Monthly Quit Rate") # We plot the PACF of the quit rate
```



Both the ACF and the PACF indicate that there is some level of autocorrelation between the quit rate at time t and that at different lags of it. The PACF in particular suggests some interesting patterns: that there is a strong positive autocorrelation between the quit rate at time t and that at time $t-1$ and that there is a strong negative autocorrelation between it and the quit rate at time $t-13$, i.e., a year and a month prior.

(d) Fitting a Linear and Non-Linear Model to the Data

We first create a time sequence that runs for the same time periods as our data:

```
time <- seq(from = (2000 + 11/12), to = (2024 + 1/12), by= 1/12) # We create a sequence of integers in
```

We now fit a linear model to the data over time:

```
lin_mod <- lm(quits_monthly ~ time) # We run a linear regression of the quit rate over time
summary(lin_mod) # We output the model information
```

```
##
## Call:
## lm(formula = quits_monthly ~ time)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.86229 -0.31344  0.01828  0.27452  1.27598
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -46.995566   7.790542  -6.032 5.16e-09 ***
## time          0.024347   0.003871   6.290 1.24e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.434 on 277 degrees of freedom
## Multiple R-squared:  0.125,  Adjusted R-squared:  0.1218
## F-statistic: 39.56 on 1 and 277 DF,  p-value: 1.237e-09
```

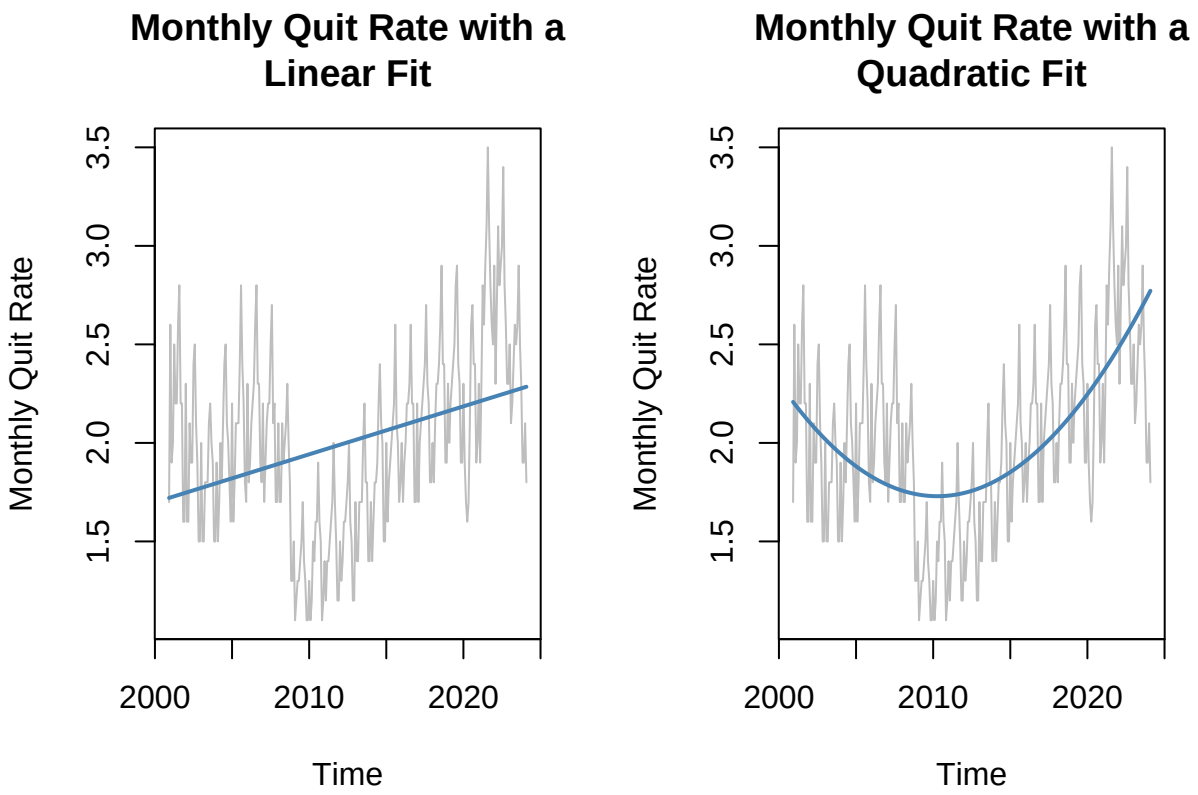
Noticing the curve upwards in 2010, we try a quadratic model as our non-linear model, fitting it to the data over time as well:

```
time_sq <- time^2 # We square the time sequence we created before to have our quadratic variable
quad_mod <- lm(quits_monthly ~ time + time_sq) # We run a linear regression of the quit rate over the q
summary(quad_mod) # We output the model information
```

```
##
## Call:
## lm(formula = quits_monthly ~ time + time_sq)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.97197 -0.26396 -0.01943  0.25869  1.07126
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.209e+04  2.252e+03   9.808   <2e-16 ***
## time        -2.197e+01  2.238e+00  -9.818   <2e-16 ***
## time_sq      5.465e-03  5.560e-04   9.829   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3742 on 276 degrees of freedom
## Multiple R-squared:  0.3518, Adjusted R-squared:  0.3472
## F-statistic: 74.91 on 2 and 276 DF,  p-value: < 2.2e-16
```

We now plot the monthly quit rate with the two linear models fit to it:

```
par(mfrow = c(1, 2)) # We set up the 2x1 matrix environment
plot(quits_monthly, main = "Monthly Quit Rate with a\nLinear Fit", ylab = "Monthly Quit Rate", col = "g
lines(time, lin_mod$fitted.values, col = "steelblue", lwd = 2) # We overlay our linear fit on the quit
plot(quits_monthly, main = "Monthly Quit Rate with a\nQuadratic Fit", ylab = "Monthly Quit Rate", col =
lines(time, quad_mod$fitted.values, col = "steelblue", lwd = 2) # We overlay our quadratic fit on the q
```

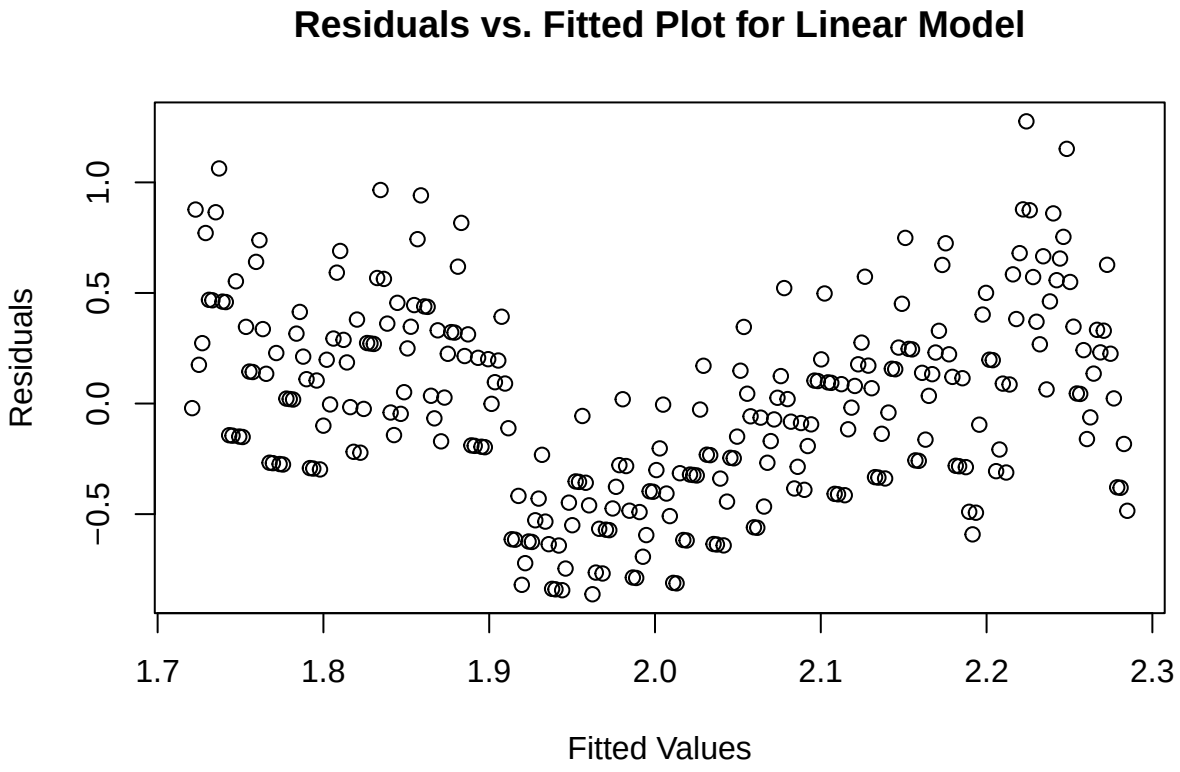


It appears that the quadratic model provides a much closer fit to the data than the linear model does.

(e) Plotting the Residuals vs Fitted Values

We now examine the residuals of each type of fit to see if there are any unexplained dynamics. First the linear model:

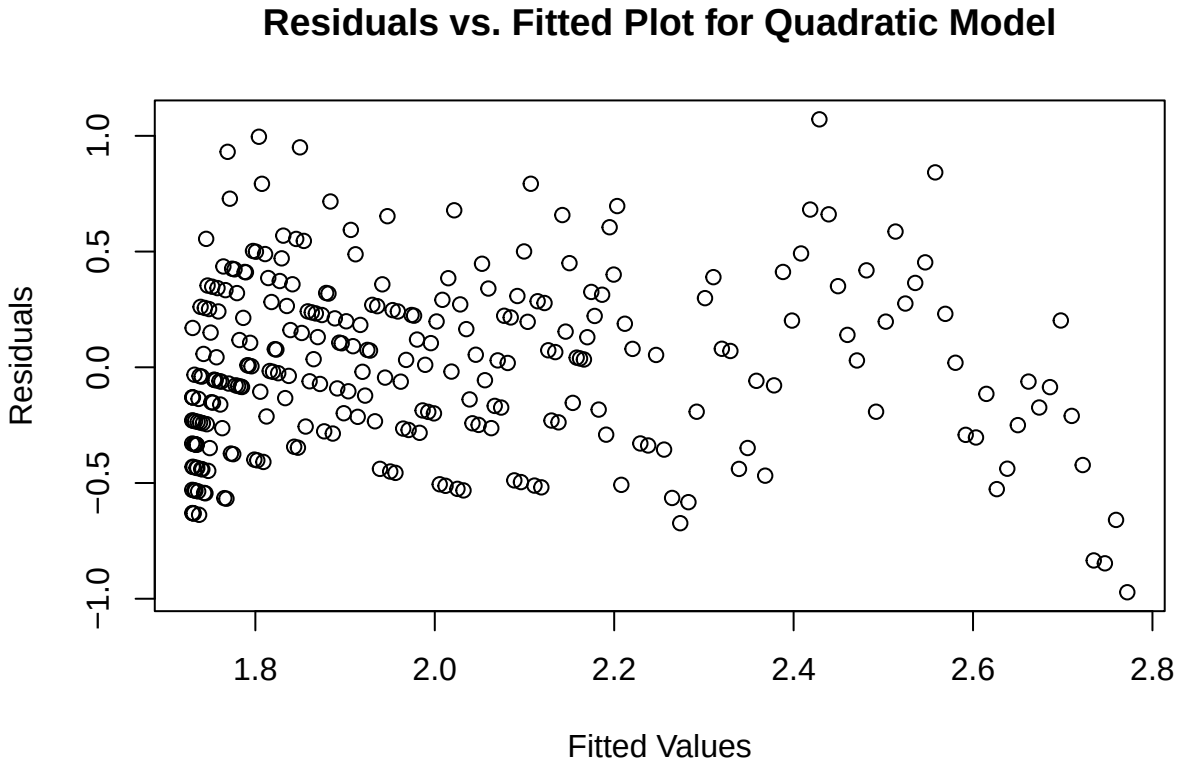
```
plot(lin_mod$residuals ~ lin_mod$fitted.values, main = "Residuals vs. Fitted Plot for Linear Model", xla
```



There is a pretty clear curve in the residuals, implying that there are likely some dynamics to the data that this model is not capturing. The curved nature of the plot suggests a model with higher order terms may do better in explaining the data. This also means that the model assumption that the error terms have constant expectation of being around 0 is violated, as the average value changes based on the fitted value, implying this is not a good model. Since the variance in the points is also somewhat variable, the linear model appears to violate two of the key residual assumptions.

We now examine the residuals versus fitted plot for the quadratic model:

```
plot(quad_mod$residuals ~ quad_mod$fitted.values, main = "Residuals vs. Fitted Plot for Quadratic Model
```

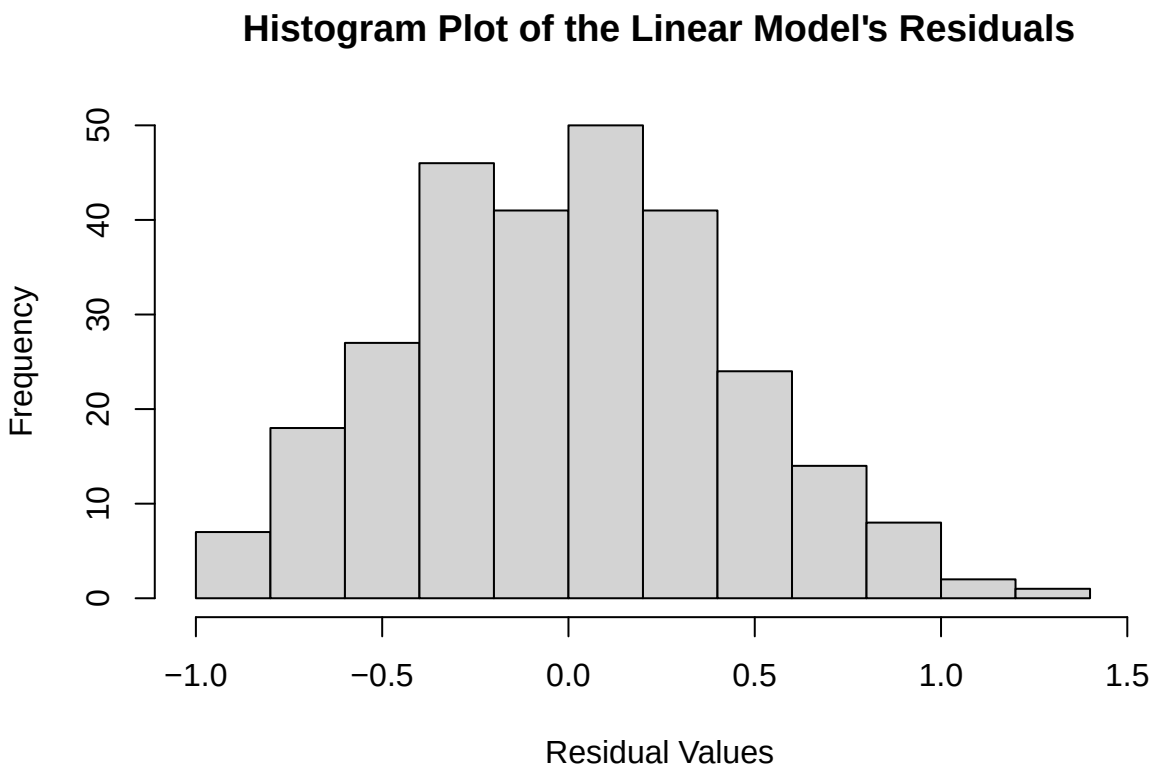


There is less of a dynamic to the residuals graph, but we can now see that there is clumping in the residuals closer to lower fitted values and a higher spread on higher fitted values, implying that there is non-constant variance to these residuals. This makes sense, as lower predicted values are found on the parts of the data before the variance got very severe, causing the estimation to be closer to the true values for earlier, and smaller values. Unfortunately, the change in variance implies that this model violates the assumption that the variance of the errors are constant over time. However, the errors appear to be centered about 0 no matter the fitted value, implying that the assumption of errors being i.i.d. distributed with mean of 0 is satisfied. Overall, the quadratic model so far appears to be satisfying the model assumptions better than the linear one, although there is still room for improvement.

(f) Plotting Residual Histograms

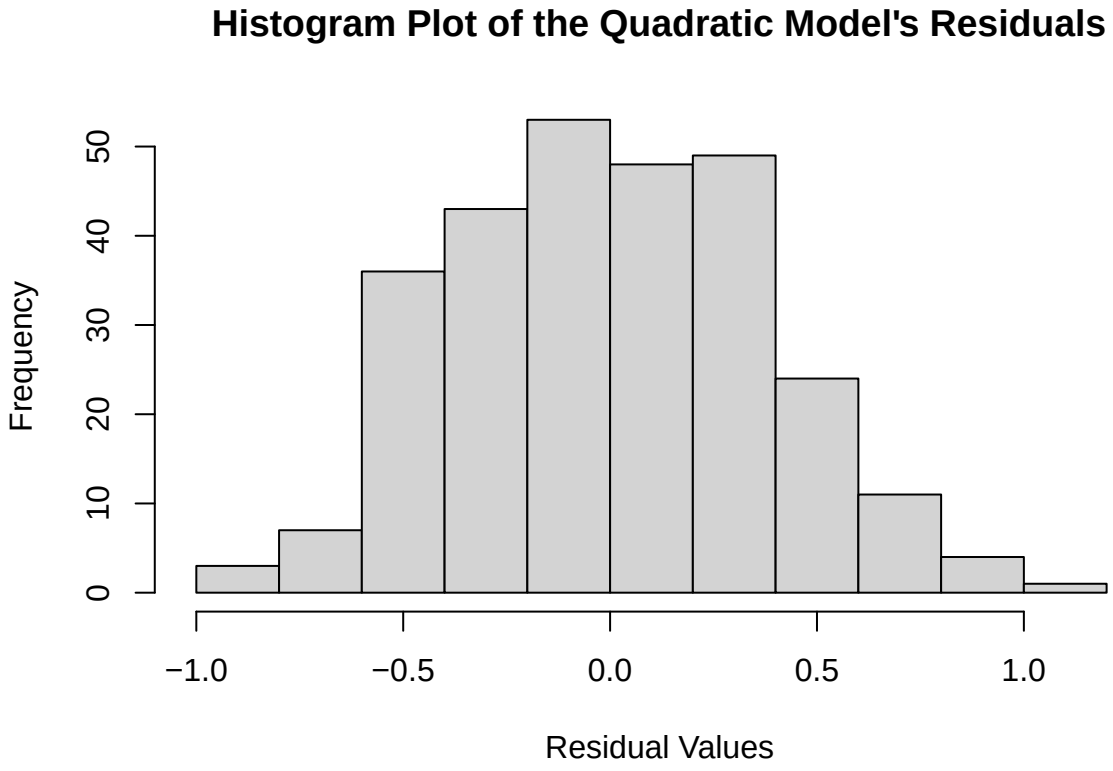
We now examine the histograms of the residuals to see if they are normally distributed. First, again, our linear model:

```
hist(lin_mod$residuals, main = "Histogram Plot of the Linear Model's Residuals", xlab = "Residual Value
```



The residuals of the linear model are somewhat normally distributed, although the tails are somewhat fat and the peak not very well pronounced, there is a decently normal distribution among the residuals. This means that the assumption of normality appears to hold.

```
hist(quad_mod$residuals, main = "Histogram Plot of the Quadratic Model's Residuals", xlab = "Residual Va
```



These residuals appear to be somewhat normally distributed as well, their problem is the opposite of what we saw with the linear model, now the peak is highly pronounced, while the actual distribution is much narrower. However, a normal distribution would still somewhat agree with the residuals, so the assumption of normality among the errors appears to hold.

(g) Looking at the diagnostic statistics for each model

R^2, adj R^2, t and F stats,

```
lin_mod_summ <- summary(lin_mod) # We store the object containing all statistics of interest from the l
lin_mod_f <- unname(lin_mod_summ$fstatistic) # We extract the F-statistic data from the summary
cat("Coef p-val:", paste0(c("Intercept: ", "time: "), lin_mod_summ$coefficients[, "Pr(>|t|)"]), # We ex
"\nF-stat p-val:", pf(lin_mod_f[1], lin_mod_f[2], lin_mod_f[3], lower.tail = FALSE), # We compute the p
"\nR^2:", lin_mod_summ$r.squared, # We extract the R^2 value from the summary
"\nAdj R^2:", lin_mod_summ$adj.r.squared) # We extract the adjusted R^2 value from the summary
```

```
## Coef p-val: Intercept: 5.15749808705995e-09 time: 1.23680099278235e-09
## F-stat p-val: 1.236801e-09
## R^2: 0.1249628
## Adj R^2: 0.1218038
```

For the linear model, both the intercept and time coefficients are statistically different from 0 (making them significant) at the 5% level, with p-values less than 10^{-8}. An F-test that at least one of the slope coefficients is statistically different from 0 comes to the same conclusion, reporting an identical p-value as that for the t-test on the significance of the time coefficient, which makes sense, as that is the only slope coefficient present in the model. However, when we look at the R^2 and adjusted R^2 values, we are disappointed: despite the high significance of the model it only explains about 12.5% of the variation in the data according to the R^2 value and 12.2% of the variation in the data according to the $\bar{R}^2$ value.

```
quad_mod_summ <- summary(quad_mod) # We store the object containing all statistics of interest from the
quad_mod_f <- unname(quad_mod_summ$fstatistic) # We extract the F-statistic data from the summary
cat("Coef p-val:", paste0(c("Intercept: ", "time: ", "time^2: "), quad_mod_summ$coefficients[, "Pr(>|t|
"\nF-stat p-val:", pf(quad_mod_f[1], quad_mod_f[2], quad_mod_f[3], lower.tail = FALSE), # We compute th
"\nR^2:", quad_mod_summ$r.squared, # We extract the R^2 value from the summary
"\nAdj R^2:", quad_mod_summ$adj.r.squared) # We extract the adjusted R^2 value from the summary
```

```
## Coef p-val: Intercept: 1.11752796505136e-19 time: 1.03817573804776e-19 time^2: 9.58292328320048e-20
## F-stat p-val: 1.02639e-26
## R^2: 0.351849
## Adj R^2: 0.3471523
```

For the quadratic model, the intercept, time, and time squared coefficients are all statistically different from 0 (making them significant) at the 5% level, with p-values even lower than in the linear model: less than 10^{-18} for all of them. An F-test that at least one of the slope coefficients is statistically different from 0 comes to the same conclusion, reporting that it is the case that at least one slope parameter is not 0 with a p-value of less than 2×10^{-16}. There is much improvement in the model when we look at the R^2 and adjusted R^2 values, although still not very high, they have almost tripled from the linear model and the quadratic model is now able to explain about 35.2% of the variation in the data according to the R^2 value and 34.7% of the variation in the data according to the $\bar{R}^2$ value.

(h) Selecting a Trend Model with AIC and BIC

```
AIC(lin_mod, quad_mod) # We compute the AIC of the linear and quadratic models
```

```
##          df      AIC
## lin_mod   3 329.9550
## quad_mod  4 248.2152
```

From AIC, we conclude that the quadratic model is a better trend model, with a much lower AIC value.

```
BIC(lin_mod, quad_mod) # We compute the BIC of the linear and quadratic models
```

```
##          df      BIC
## lin_mod   3 340.8486
## quad_mod  4 262.7400
```

We come to the same conclusion with BIC as we did with AIC, that the quadratic model, despite its additional parameter, is a better fit to the data than the linear model, as it has a much lower BIC.

(i) Forecasting with the Trend Model

```
future_dates <- data.frame(time = seq(from = time[279], by = 1/12, length.out = 13)[-1], time_sq = (seq
predict(quad_mod, future_dates, interval = "prediction") # We forecast using the quadratic model along
```

```
##         fit      lwr      upr
## 1  2.784590 2.036041 3.533139
## 2  2.797283 2.048391 3.546176
## 3  2.810052 2.060808 3.559297
## 4  2.822898 2.073292 3.572503
## 5  2.835819 2.085844 3.585793
## 6  2.848816 2.098463 3.599168
## 7  2.861888 2.111149 3.612628
## 8  2.875037 2.123901 3.626173
## 9  2.888262 2.136721 3.639803
## 10 2.901562 2.149607 3.653518
## 11 2.914939 2.162559 3.667319
## 12 2.928391 2.175578 3.681204
```

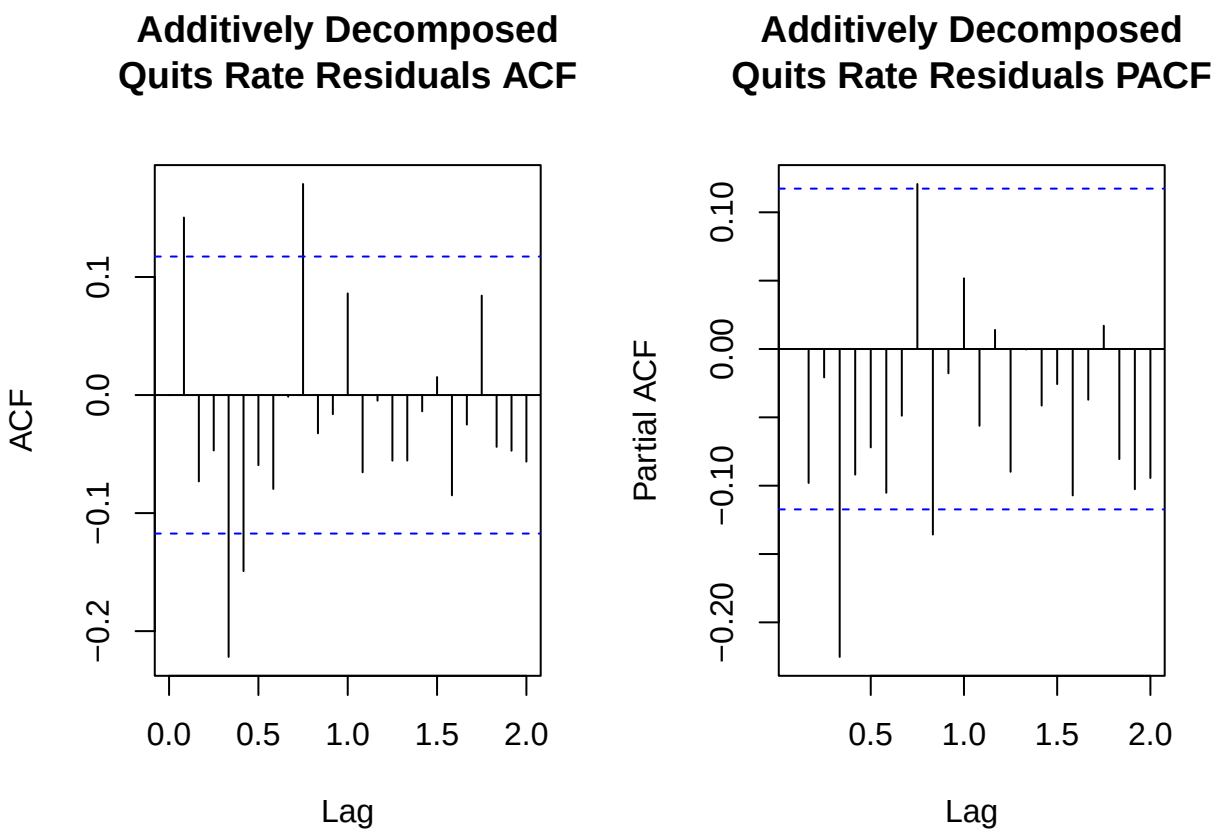
We see that the predicted values for the quits the next year into the future follow the quadratic trend, with the 95% confidence interval suggesting they will likely fall somewhere in the range from a rate of 2.5% to 3% of the paid employees quitting each month.

2. Trend and Seasonal Adjustments

(a) Additively Decomposing the Series

```
quits_add_dcmp <- decompose(quits_monthly, type = "additive") # We conduct a classical moving average d
quits_add_detrnd_sadj <- quits_monthly - quits_add_dcmp$seasonal - quits_add_dcmp$trend %>%
  ts(frequency = 12, start = c(2000, 12)) # We remove the trend and seasonality from the data using the
acf_plot_add <- acf(quits_add_detrnd_sadj, na.action = na.pass, plot = FALSE) # We create the structure
acf_plot_add$acf[1, 1, 1] <- 0 # We remove the first spike from the ACF plot as it will always have a v
pacf_plot_add <- pacf(quits_add_detrnd_sadj, na.action = na.pass, plot = FALSE) # We create the structu
pacf_plot_add$acf[1, 1, 1] <- 0 # We remove the first spike from the PACF plot as it will always have a

par(mfrow = c(1, 2)) # We initialize the graphical environment with two columns for graphs
plot(acf_plot_add, main = "Additively Decomposed\nQuits Rate Residuals ACF") # We plot the ACF
plot(pacf_plot_add, main = "Additively Decomposed\nQuits Rate Residuals PACF") # we plot the PACF
```

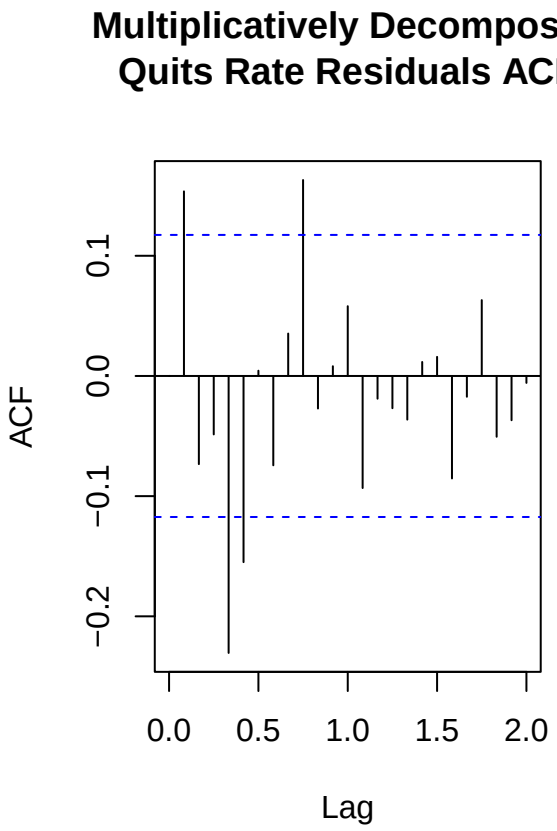


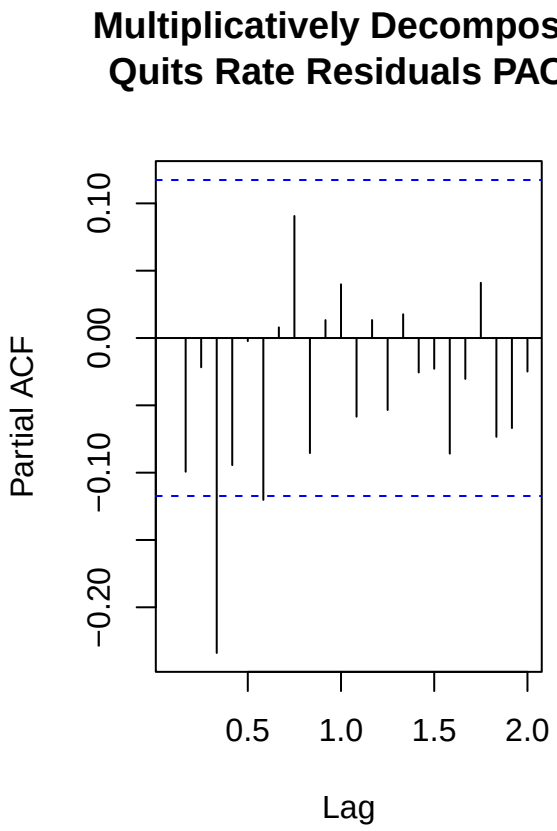
Examining the ACF and PACF plots, we can see that there still exist some significant lags, implying that an additive seasonal decomposition may not fully remove all of the characteristics of the data as we'd like it to.

(b) Multiplicatively Decomposing the Series

```
quits_mult_dcmp <- decompose(quits_monthly, type = "multiplicative") # We conduct a classical moving av
quits_mult_detrnd_sadj <- (quits_monthly / quits_mult_dcmp$seasonal) / quits_mult_dcmp$trend %>%
  ts(frequency = 12, start = c(2000, 12)) # We remove the trend and seasonality from the data using the
acf_plot_mult <- acf(quits_mult_detrnd_sadj, na.action = na.pass, plot = FALSE) # We create the structu
acf_plot_mult$acf[1, 1, 1] <- 0 # We remove the first spike from the ACF plot as it will always have a
pacf_plot_mult <- pacf(quits_mult_detrnd_sadj, na.action = na.pass, plot = FALSE) # We create the struc
pacf_plot_mult$acf[1, 1, 1] <- 0 # We remove the first spike from the PACF plot as it will always have

par(mfrow = c(1, 2)) # We initialize the graphical environment with two columns for graphs
plot(acf_plot_mult, main = "Multiplicatively Decomposed\nQuits Rate Residuals ACF") # We plot the ACF
plot(pacf_plot_mult, main = "Multiplicatively Decomposed\nQuits Rate Residuals PACF") # We plot the PAC
```





The multiplicative decomposition gives similar results on the ACF and PACF plots of the residuals as the additive one did, with slightly smaller magnitudes on the spikes.

(c) Comparing the Additive and Multiplicative Decompositions

Comparing the PACF plots, which shows which decomposition better captures all the dynamics of the data, indicates that the two decompositions are quite similar. The multiplicative decomposition appears to model patterns in the data slightly better than the additive one — it has only one significant PACF spike as opposed to two and the same number of significant ACF spikes — but the differences are very minor.

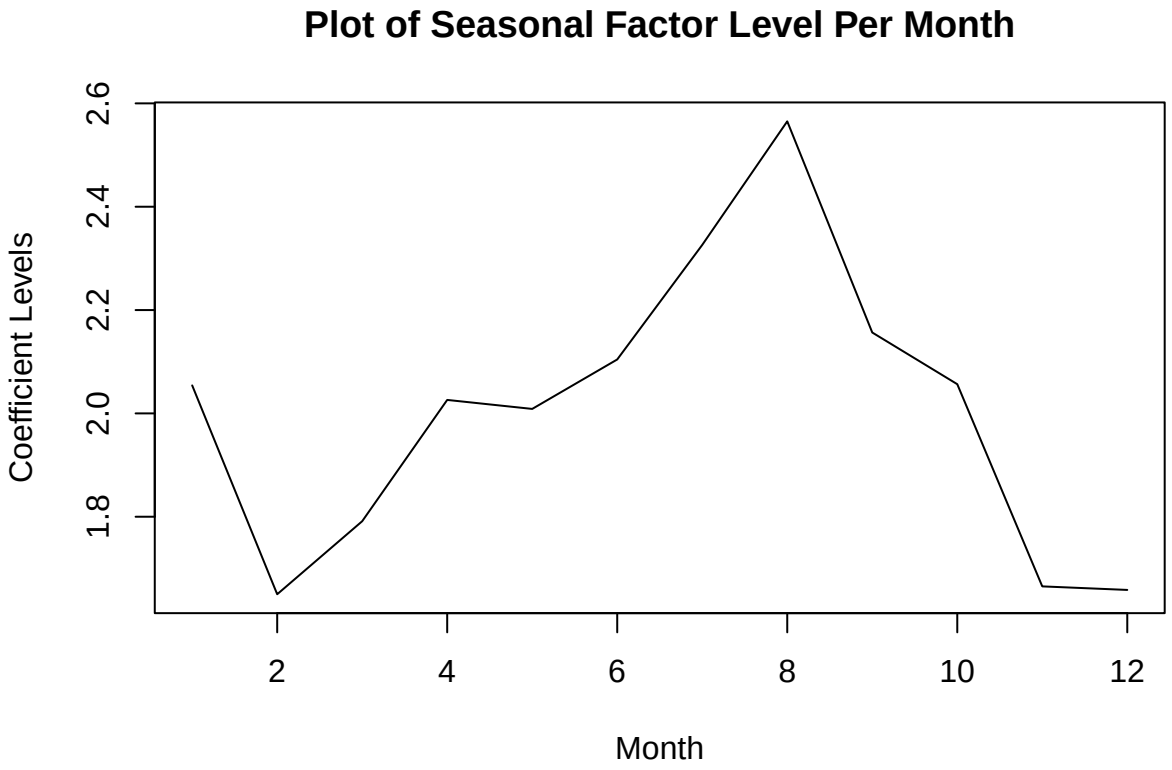
(d) Comparing Cycles Between the Decompositions

Judging from the high level of similarity between the ACF and PACF plots of the residuals of both the additive and multiplicative decompositions, the models for the cycles should be quite similar.

(e) Plotting Seasonal Factors

We now plot the seasonal factors to look for seasonal trends present in the data.

```
plot(tslm(quits_monthly ~ season + 0)$coefficients, type = "l", ylab = "Coefficient Levels", xlab = "Mo
```



Looking at the seasonal factor levels, we see that there is a spike in quit rates in the eighth season, August, with troughs in February and December. These trends could have a couple causes: the peak could be from teenage summer employees quitting a summer job before returning to school, while the troughs could be that people would rather not quit work during the holiday season for December and that since February is a shorter month, it appears to not have as high of a quit rate because there are fewer days over which to collect data.

(f) Choosing a Trend-Seasonal Model and Forecasting with It

In part (c), we were inconclusive about whether a multiplicative or additive decomposition is better for the data, so we will run models of our quadtratic model with both additive and multiplicative decompositions to see which performs bettter. An additive seasonality looks like:

$$y_t = (\beta_0 + \beta_1 TIME_t + \beta_2 TIME_t^2) + \sum_{i=1}^{s} \gamma_i D_{it} + \varepsilon_t$$

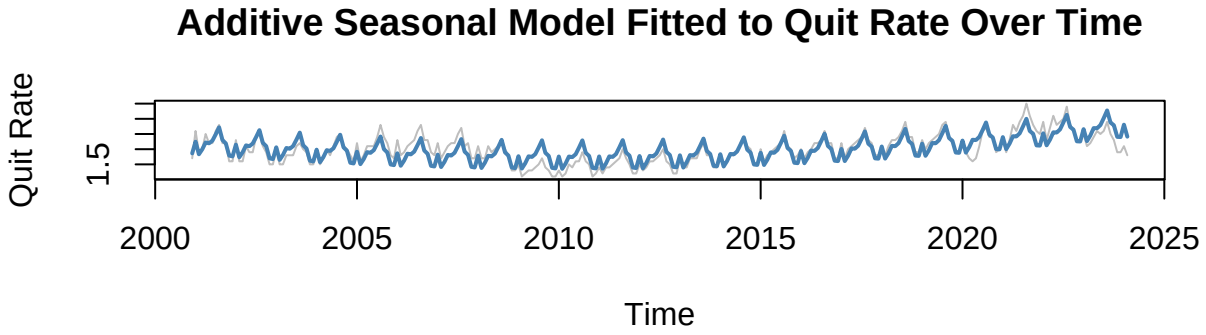
while a multiplicative one is of the form:

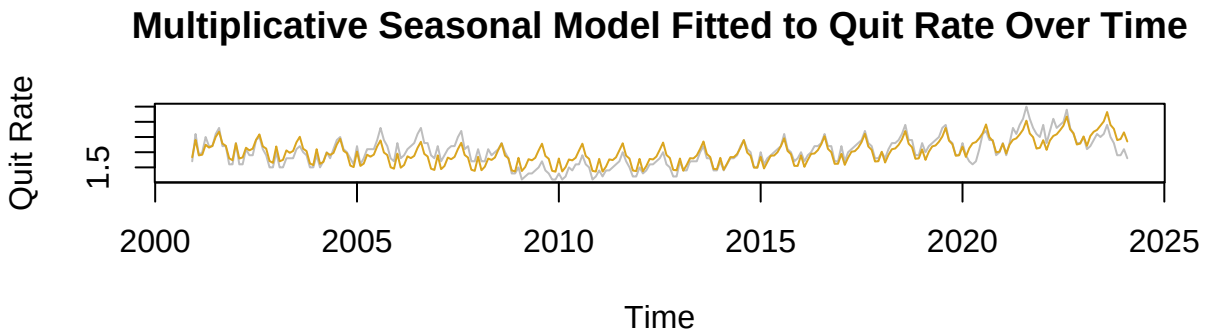
$$y_t = (\beta_0 + \beta_1 TIME_t + \beta_2 TIME_t^2) \times \sum_{i=1}^{s} \gamma_i D_{it} + \varepsilon_t$$

where y_t is the quits rate at time t, $TIME$ is the month at time t, s is the number of seasons in a year, in this case 12, D_{it} is a dummy variable indicating the season at time t, and ε_t is the inherent error present in the model at time t. We run the two models:

```
quad_add_mod <- tslm(quits_monthly ~ time + time_sq + season) # We build a linear model with the quadra
quad_mult_mod <- tslm(quits_monthly ~ (time + time_sq):season) # We build a linear model with the quadr
par(mfrow = c(2, 1)) # We initialize the graphical environment to have 2 rows of graphs and 1 column
plot(quits_monthly, col = "gray", main = "Additive Seasonal Model Fitted to Quit Rate Over Time", ylab =
```

```
lines(quad_add_mod$fitted.values, col = "steelblue", lwd = 2) # We plot the fit of the additive quits r
plot(quits_monthly, col = "gray", main = "Multiplicative Seasonal Model Fitted to Quit Rate Over Time",
lines(quad_mult_mod$fitted.values, col = "goldenrod") # We plot the fit of the multiplicative quits rat
```





We see that the two models look very similar, with only slightly different fitted values over time. We check the AIC and BIC to determine the better model:

```
AIC(quad_add_mod, quad_mult_mod) # We test the AIC of the additive and multiplicative models
```

```
##                df      AIC
## quad_add_mod   15 55.02545
## quad_mult_mod 26 70.68027
```

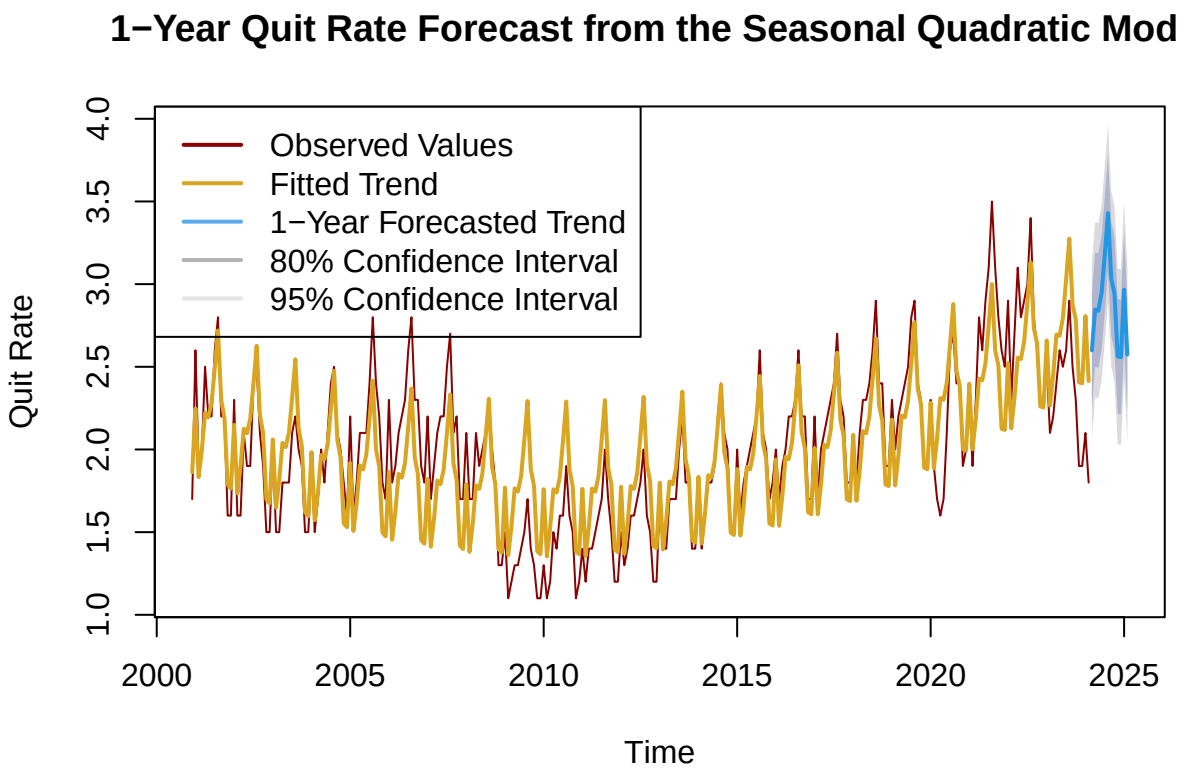
```
BIC(quad_add_mod, quad_mult_mod) # We test the BIC of the additive and multiplicative models
```

```
##                df      BIC
## quad_add_mod   15 109.4936
## quad_mult_mod 26 165.0918
```

By both criteria, the additive model seems to perform slightly better, so we choose to model seasonality with an additive model.

We now forecast from the additive model:

```
future_dates_seasonal <- cbind(future_dates, season = vapply(future_dates$time, function(x) {round(x %%
plot(forecast(quad_add_mod, future_dates, h = 12), main = "1-Year Quit Rate Forecast from the Seasonal
lines(quad_add_mod$fitted.values, col = "goldenrod", lwd = 2, lty = 1) # We plot the forecasted values
legend("topleft", legend = c("Observed Values", "Fitted Trend", "1-Year Forecasted Trend", "80% Confide
```



We see that this model performs better than the purely quadratic one in terms of capturing the seasonal trends to the data, however, because the trend is increasing, it will continue to see an increase in predicted quit rate as the time horizon gets longer.

III. Conclusions and Future Work

The final model indicates that the quit rate can be approximated by a function that contains linear and quadratic terms over time and adds in seasonal components to track monthly changes in the data. Forecasting with it for the 12 months following the end of the data indicates that there will be an increase in the quit rate, following the seasonal pattern observed in the data, with a 95% confidence interval giving a range for the quit rate that is higher than the most recently observed one. This is an important drawback to the model and one that may be able to be addressed in the future: the quadratic trend to the data is caused mostly by the drop in quit rate between 2008 and 2011, followed by an increase in the rate up until 2022. The most recent data indicates a new downwards trend, which is not being captured by the model. This is indicative of the presence of cycles in the data, which could be captured in a future model. The model could also benefit from computing the trend from a moving window of the data, so that it could be used to model more complex patterns.

IV. References

Data:

U.S. Bureau of Labor Statistics, Quits: Total Nonfarm [JTUQUR], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/JTUQUR, April 16, 2024.

Data description:

U.S. Bureau of Labor Statistics (BLS), Job Openings and Labor Turnover Survey, retrieved from BLS; https://www.bls.gov/jlt/jltdef.htm, April 16, 2024.

U.S. Bureau of Labor Statistics (BLS), Glossary, retrieved from BLS; https://www.bls.gov/opub/hom/glossary.htm#R, April 16, 2024.