Laboratory 2 Written Report

ANA 535 Forecasting

Periodicity in Time Series Data

Name KOHEI NISHITANI Student ID# 1122867

Date April 13, 2025

# Table of Contents

* Introduction - 3
* Background - 4
* Data - 4
* Methods and Procedures - 5
* Results - 7
* Conclusions - 10
* References - 10
* Attachment or Appendices - 10

# Introduction

Spectral methods used for periodicity identification and management within data sets receive attention through the second laboratory session which extends basic time-series principles. The laboratory starts with creation of basic sine waves at specific frequencies and sampling rates to produce composed signals through wave combination. Through this procedure the Fast Fourier Transform (FFT) transforms time-domain waves into frequency-domain power spectral which produce specific frequency peaks. As part of the laboratory instruction researchers explain how moving a wave through π radians results in total destructive interference when combined with its original form. The evaluation of dominant frequencies in artificial and natural datasets makes use of periodograms for both verification and measurement purposes.

The real-world application takes the educated signals to explore genuine data sets that showcase:

* The recorded hourly CO along with NOx data shows clearly observable daily patterns in their measurement results.
* The monthly samples of Amtrak passenger information demonstrate either yearly or longer-term repeating patterns.

The educational content reveals time-series decomposition which breaks a sequence into Trend, Seasonal and Remainder components by using decompose() and lag-based differencing and X-11 approach. Using seasonal wave subtraction at particular lag intervals reduces regular cycles to make data more stationary and better fit the model. Lag=12 functions for yearly waves serve as an example in this method. The exercise ends with a demonstration of time regression analysis which acts as an alternative way to model continuing trends following seasonal adjustments.

This analysis platform includes several complementary statistical methods running on R software including FFT for finding cyclic patterns together with periodograms for validating frequency information along with differencing and decomposition for seasonal adjustments and polynomial regression for trending estimation. The discussed methods receive real dataset demonstration at the end of this laboratory through monthly Amtrak passenger miles and CO and Nox emission data which serves as base information for developing advanced forecasting solutions.

**Background**

The past laboratory session focused on preparing time-series data for analysis by addressing core steps like cleaning, sort and filtering data. The Amtrak passenger dataset is now ready for analysis but needed researchers to find and process seasonal patterns found in time-series data. Signals from actual measurements like pollution and passengers follow regular repeating periodic behaviour. The problem of unaccounted cycles affects our ability to treat the series as stationary for traditional forecasting tools. The lab activities focus on multiple main subject areas which described method and procedure section.

**Data**

Data from two sources enables the exploration of time-series analysis periodicity finding and adjustment through this exercise. The Amtrak dataset covers a time span from January 1991 to June 2024 holding about four hundred monthly records. The records contain a month field along with Ridership, PassengerMiles and RidersReported fields. Because monthly data shows recurring seasonal behaviour on annual cycles decomposition and differencing methods would make suitable choices for analysis. The other dataset Hourly COandNOx2 measurements from one year were recorded air quality dataset. The dataset consists of 8,760 rows that includes carbon monoxide measurements along with other measurements. Analysis starts with daily cycles which last 24 hours because this dataset rovides one-hour measurements. When using subsequent modeling methods which require stational data such as ARIMA it becomes necessary to identify and eliminate repeating patterns because doing so improving the achievement of stationary.

**Methods and Procedures**

All of exercises were conducted within Rstudio and using provided R script and libraries such as lubridate, tsbble, feast, forecast, and TSA. These packages provide essential functionality for time-series handling, spectral analysis, and decomposition. The R script for this laboratory, located in the appendix, is organized into four main segments reflecting the steps described below.

Initially, the script generates and visualize synthetic sine waves to illustrate fundamental concept of periodicity. A sampling frequency of 1,000HZ is used for a simple 120Hz wave, verifying that a FFT finds periodicities from sine wave. This reveals FFT can detect hidden frequency (Figure1).

A diagram of a waveform

AI-generated content may be incorrect.

Figure 1

There are several methods to look at the energy in the frequency spectrum. One of the method is Periodogram from TSA library. It enables to find higher estimated spectral power (Figure. 2)

A blue screen with white text

AI-generated content may be incorrect.

Figure 2

As next, this exercise introduces phase shifts by adding pi to sine waves (Figure. 3) and demonstrates destructive interference in situation where the original and shifted signals align out of phase. This portion relies on time-domain plots to show how a shifted wave cancels the original wave if it is exactly half a cycle behind (Figure N).

A diagram of waves and waves

AI-generated content may be incorrect.A graph of waves with text

AI-generated content may be incorrect.

Figure. 3 Figure. 4

Lastly, this exercise also introduces the concept of superposition of waves and demonstrate how FFT identifies the frequency from combined and complex wave. As an example, this exercise generated three types of waves with different frequency and amplitude, then combine it to one wave (Figure. 5). Because this complex and combined wave is composed by three different frequencies, FFT can detect the frequencies by showing power spectral(Figure 6).

A group of diagrams showing different waves

AI-generated content may be incorrect.A diagram of a power spectrum

AI-generated content may be incorrect.

Figure. 5 Figure. 6

**Results**

One objective of this exercise was to examine the COandNOx2 dataset, which consists of hourly measurements of carbon monoxide over the course of a full year. The goal was to investigate what kind of periodicities exist on this dataset. An FFT applied to this hourly time series revealed a prominent spike near the 24-hour mark (Figure 7), indicating a pronounced daily repeat. Closer inspection of frequencies around 1.16 × 10⁻⁵ Hz (the reciprocal of 86 400 seconds) confirms that the dominant periodic component aligns closely with a one-day cycle. A periodogram from the TSA package further supports this finding by showing maximum spectral power in precisely those intervals(Figure 8). These observations suggest that hourly pollution measurements exhibit a strong daily pattern, implying that additional processing—such as subtracting a fitted 24-hour seasonal component—would be required to approach stationarity and facilitate accurate forecasting.

**A graph of a power spectrum

AI-generated content may be incorrect.A graph of a number of people

AI-generated content may be incorrect.**

Figure. 7 Figure. 8

In contrast to the hourly CO data, the Amtrak dataset is sampled monthly, covering period from 1991 to 2024. The decomposition plot(Figure 9), which segregates trend, seasonality, and reminder reveals an annual pattern in the seasonal component.

**A graph of a number of different miles

AI-generated content may be incorrect.**

Figure. 9

This exercise also introduced two methods how to reduce or remove this seasonal effect. First, using diff() command from forecast package, differencing at lag=12, and optionally at shorter lags like 6 or 3, demonstrated how subtracting the previous year’s readings stabilizes the data and reduces annual variation(Figure 10).

**A graph of a waveform

AI-generated content may be incorrect.**

Figure 10

Secondly, classical decomposition approach calculates separate trend and seasonal components using decompose() function. By subtracting the seasonal wave from original time-series data, this output shows seasonally adjusted version whose recurring peak subtracted. Bottom left Zoomed-in plot(Figure 11) shows seasonally adjusted data and it reveals no strong periodicity, that suggesting that differencing or direct subtraction effectively removes periodic cycle.

A graph of different types of time

AI-generated content may be incorrect.

Figure 11

Lastly, X-11 method from seasonal package converts time-series to tsibble format(Figure 12). And this package generates components of raw data, season-adjusted series, and trend, easily extract those components and visualize in one plot(Figure 13).

A graph of different lines

AI-generated content may be incorrect.

Figure 12

A graph of a passenger

AI-generated content may be incorrect.

Figure 13

**Conclusions**

This exercise demonstrated several methods for detecting and removing periodicities in time-series data. Synthetic waves confirmed the effectiveness and usefulness of FFT and periodograms for revealing hidden cycle. This exercise also engages with real-world dataset, it showed strong daily and annual patterns respectively, it enables to achieve stationary to run forecasting accordingly.

**References**

**Hyndman, R.J., & Athanasopoulos, G. (2021) Forecasting: principles and practice, 3rd edition, OTexts: Melbourne, Australia. OTexts.com/fpp3. Accessed on April 8 2025.**

**Peter J. Brockwell and Richard A. Davis (1996)** Introduction to time series and forecasting

**U.S. Department of Transportation, Bureau of Transportation Statistics** (2025, April 7).

*Monthly Transportation Statistics*. [https://data.bts.gov/Research-and-Statistics/Monthly-Transportation-Statistics/crem-w557/about\_data](https://nam04.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdata.bts.gov%2FResearch-and-Statistics%2FMonthly-Transportation-Statistics%2Fcrem-w557%2Fabout_data&data=05%7C02%7Ckn0117%40mcdaniel.edu%7C685a3463f92143d3981308dd76ed7880%7C5db4271773af42439ab0e4402b2add7a%7C0%7C0%7C638797484532942557%7CUnknown%7CTWFpbGZsb3d8eyJFbXB0eU1hcGkiOnRydWUsIlYiOiIwLjAuMDAwMCIsIlAiOiJXaW4zMiIsIkFOIjoiTWFpbCIsIldUIjoyfQ%3D%3D%7C0%7C%7C%7C&sdata=bIQQ7oDFx%2FN0XwcBs5jWsb8VWEIrXyp1CNrCyTLl508%3D&reserved=0)

**S. De Vito, E. Massera, M. Piga, L. Martinotto, G. Di Francia**. (2008).

On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario, *Sensors and Actuators B: Chemical,*129(2), 750-757.

**References**

library(xlsx)

library(fpp3)

library(dplyr)

library(tidyverse)

library(ggplot2)

library(tsibble)

library(tsibbledata)

library(fable)

library(feasts)

library(lubridate)

library(zoo)

library(forecast)

library(seasonal)

#

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Step 1 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#

#

#First complete the analysis of a simple sine wave at a specified freq.

#Setup the parameters. You are given that the data are collected at

#a sampling frequency (Fs) of 1,000 Hertz. Hertz are cycles per second.

#That is am important distinction. Later we will compute frequencies and

#need to put time in the denominator, i.e. 1/time. The period, T, is

#1/Fs.There are 1,500 data points available. The independent variable is

#time (t).

#

Fs <- 1000

T = 1/Fs

L = 1500

t = (0:L-1)\*T

#

#Generate the data and plot it

#

Sig1 <- sin(2\*pi\*120\*t)

plot(1000\*t(1:100), Sig1[1:100], type = "l")

#

#Conduct the FFT and plot the resulting power spectrum

#

Y <- fft(Sig1)

P2 <- abs(Y/L)

m <- (length(Y)/2)+1

P1 <- P2[1:m]

P1 <- 2\*P1

f = Fs\*(0:(L/2)/L)

plot(f, P1, type = "l")

#

#There are several ways you can look at the energy in the frequency

#spectrum. One way is to generate a perriodogram. Another way

#is to generate the frequency spectrum. First, we'll look at

#the periodogram.

#

library(TSA)

p = periodogram(Sig1)

dd=data.frame(freq=p$freq, spec=p$spec)

order=dd[order(-dd$spec),]

top1=head(order, 1)

top1

#

#If you multiply the frequency output by the TSA periodogram by

#the sampling frequency you'll get 119.7917 or about 120 Hz

#which is the original signal frequency. The periodogram is great

#for a univariate signal. It does not work well for complex signals

#composed of many frequencies.

#

#

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Step 2 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#

#Now work on the combined, complex wave problem

#wave1 has a freq of 100 and amplitude of 5

#wave2 has a freq of 200 and amplitude of 10

#wave3 has a freq of 400 and amplitude of 15

#

#

#Setup the parameters

#

Fs <- 1000

T = 1/Fs

L = 1500

t = (0:L-1)\*T

#

#Generate the data and plot it

#

wave1 <- 5\*sin(2\*pi\*100\*t)

wave2 <- 10\*sin(2\*pi\*200\*t)

wave3 <- 15\*sin(2\*pi\*400\*t)

par(mfrow = c(2,2))

plot(1000\*t(1:100), wave1[1:100], type = "l")

plot(1000\*t(1:100), wave2[1:100], type = "l")

plot(1000\*t(1:100), wave3[1:100], type = "l")

#

#Combine the three waves into one complex wave and plot

#

par(mfrow = c(1,1))

wave4 <- wave1 + wave2 + wave3

plot(1000\*t(1:100), wave4[1:100], type = "l",

main = "Complex, Combined Wave Time Plot")

#

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Step 3 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#

#

#Now let's look at what a phase shift does. A phase shift by pi will setup

#destructive interference. First, setup a sequence of points over 180 degrees.

#

xs <- seq(-100\*pi, 100\*pi, pi/100)

#

#Then, set up the sine waves. And, just to make things interesting

#let's combine them into a complex wave.

#

wavea <- sin(0.1\*xs)

waveb <- sin(0.333\*xs)

wavec <- wavea + waveb

par(mfrow = c(3,1))

plot(xs,wavea,type="l", ylim=c(-1,1)); abline(h=0,lty=3)

plot(xs,waveb,type="l", ylim=c(-1,1)); abline(h=0,lty=3)

plot(xs,wavec,type="l", ylim=c(-1,1)); abline(h=0,lty=3)

#

#Next setup a phase shift by 180 degrees or pi.

#

wavea2 <- sin(0.1\*xs + pi)

waveb2 <- sin(0.333\*xs + pi)

wavec2 <- wavea2 + waveb2

#

#Now let's plot the original complex wave, the wave shifted by pi

#or 180 degrees, and the combination of these two waves. As you

#can see, the combination of the original wave and the phase

#shifted wave is zero, total destructive interference.

#

par(mfrow = c(3,1))

plot(xs,wavec,type="l", ylim=c(-1,1)); abline(h=0,lty=3)

plot(xs,wavec2,type="l", ylim=c(-1,1)); abline(h=0,lty=3)

waved <- wavec + wavec2

plot(xs,waved,type="l", ylim=c(-1,1)); abline(h=0,lty=3)

#

#Conduct the fft analysis. But now, since we have several waves

#combined, we'll plot the resulting power spectrum rather than a periodogram.

#

Y <- fft(wave4)

P2 <- abs(Y/L)

m <- (length(Y)/2)+1

P1 <- P2[1:m]

P1 <- 2\*P1

f = Fs\*(0:(L/2)/L)

par(mfrow = c(1,1))

plot(f, P1, type = "l", main = "Combined Wave Power Spectrum",

xlab = "Frequency", ylab = "Strength")

#

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Additional example from real-world

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* signal acquisition for air pollution

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* due to street traffic.

#

#

#Second complete the analysis of a CO signal from an air monitoring

#station by a roadway in Italy.

#

#

#Setup the parameters. Note that the data were acquired hourly so the sampling

#frequency is (60 sec \* 60 min = 3600 or 0.000278 Hertz).

#

Fs <- 0.000278

T = 1/Fs

L = 8760

t = (0:L-1)\*T

#

#Conduct the fft analysis and plot the resulting power spectrum

#

#Read in the data and add variable names

#

COandNOx2 <- read.csv("I:/My Passport Documents/McDaniel/DataAnalytics/ANA535/NewANA535/Laboratories/COandNOx2.txt", header=FALSE)

colnames(COandNOx2) <- c("CO.GT", "NOx.GT", "T", "RH", "AH")

Y <- fft(COandNOx2$CO.GT)

P2 <- abs(Y/L)

m <- (length(Y)/2)+1

P1 <- P2[1:m]

P1 <- 2\*P1

f = Fs\*(0:(L/2)/L)

par(mfrow = c(1,1))

plot(f, P1, type = "l", ylim = c(0, 1),

main = "CO Power Spectrum", xlab = "Frequency", ylab = "Strength")

#

#Zoom into power spectrum

#

plot(f, P1, type = "l", xlim = c(0, 0.00002), ylim = c(0, 1),

main = "CO Power Spectrum", xlab = "Frequency", ylab = "Strength")

#

#If there is a daily (24-hour) periodicity we would expect a spike at

#24\*60\*60 or 1.157 x e-005 which we see when we zoom into the power

#spectrum.

#

#

#\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Step 4 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#

#

#Now work on the Amtrak data. If you do not still have it in

#R/RStudio you can use the following lines to read it in again. You'll

#need to rewrite the path in this command to the directory/folder where

#you have stored the Amtrak data. For example, the commands and the

#path to my directory is

#setwd("I:/My Passport Documents/McDaniel/DataAnalytics/ANA535/NewANA535/Laboratories")

#Amtrak <- read.csv("Amtrak1991-2024.csv")

#colnames(Amtrak) <- c("Month", "Ridership", "PassengerMiles", "RidersReported")

#

#If you have set up different working directories for each of the

#laboratories you might need to change from where you stored the

#dataset in the "base" directory to the individual laboratory as

#follows.

#

#setwd("I:/My Passport Documents/McDaniel/DataAnalytics/ANA535/NewANA535/Laboratories")

#

#and then have sub-directories or sub-folders under this, e.g. where

#Laboratory2 is this particular laboratory.

#

setwd("I:/My Passport Documents/McDaniel/DataAnalytics/ANA535/NewANA535/Laboratories")

Amtrak <- read.csv("Amtrak1991-2024.csv")

colnames(Amtrak) <- c("Month", "Ridership", "PassengerMiles", "RidersReported")

View(Amtrak)

Amtrak$Month <- mdy(Amtrak$Month)

Amtrak

str(Amtrak)

#Amtrak$Month <- zoo::as.yearmon(Amtrak$Month)

#Amtrak

#Amtrak |>

# mutate(Month = yearmonth(Month)) |>

# as\_tsibble(index = Month)

#

#Setup the parameters

#

Fs <- 1/(30.4\*24\*60\*60)

T = 1/Fs

L = 402

t = (0:L-1)\*T

#

#Conduct the fft analysis and plot the resulting power spectrum

#

Y <- fft(Amtrak$PassengerMiles)

P2 <- abs(Y/L)

m <- (length(Y)/2)+1

P1 <- P2[1:m]

P1 <- 2\*P1

f = Fs\*(0:(L/2)/L)

par(mfrow = c(1,1))

plot(f, P1, type = "l", xlim = c(0,0.0000002), ylim = c(0,90000000),

main = "Amtrak time series power spectrum", xlab = "freq in Hz")

#

#To look for annual periodicity we need to look for a spike at

#1/(365 \* 24 \* 60 \* 60 = 3.1536 x e07) or 3.1710 x e-08 and there is a nice

#large spike at about 3 x e-08.

#

#Because months have different numbers of days and so on, looking at plots on

#this scale is a really crude approximation. The main point is that there are

#multiple periods in the Amtrak data as we will see in the time series

#decomposition.

#

#

#Code to analyze Amtrak example/case from Practical Time Series Forecasting

# with R: A Hands-on Guide (G. Shmueli and K.C. Lichtendahl Jr.) and fpp3

#

#

#Plot Amtrak data in time plot

#

ggplot(data = Amtrak) +

geom\_line(mapping = aes(x = Month, y = PassengerMiles)) +

labs(title = "Amtrak Passenger Miles by Month")

#

#The data is rough. First, the drop due to the Covid lockdowns is very obvious.

#There is also a small dip from the 2008 financial meltdown. It looks like there

#may be a seasonal component... First, split out the desired variable (Ridership

#or PassengerMiles) numbers as a time series.

#

PassengerMiles.ts <- ts(Amtrak$PassengerMiles, start = c(1991,1), end = c(2024, 6), frequency = 12)

Amtrak.comp <- decompose(PassengerMiles.ts)

par(mfrow = c(4,1))

autoplot(Amtrak.comp) +

labs(title = 'Amtrak Passenger Miles Decomposition')

#rect(0, 1.1, 1, 1.7, xpd=TRUE, col="white", border="white")

#title("Amtrak Passenger Miles Decomposition")

#

#Let's look at just the seasonal component of the Amtrak data by

#zooming into two years from that single plot

#

par(mfrow = c(1,1))

plot(Amtrak.comp$seasonal, xlim = c(1994, 1997))

#

#As we saw when we looked at the power spectrum of the Amtrak data,

#there are a number of periods in the data, e.g. annual or every year periodicity

#is pretty obvious but the data also repeat or have periodicity every 6-months,

#every roughly 3 months, etc.

#

#

#Let's look at what a seasonal adjustment would be for the Amtrak data.

#Keep in mind that there are many ways to do this. I'll show you three ways

#here. Let's start with a tried and true method by using lags. I know we

#haven't covered lags yet but we will. Just keep this in mind for later

#when you'll use it more extensively.

#

PassengerMiles.ts <- ts(Amtrak$PassengerMiles, start = c(1991,1), end = c(2024, 6), frequency = 12)

par(mfrow = c(1,1))

#

#The first way is to use the diff() command from the forecast package in R

#You can use either the syntax to find values of components of a time series

#with a lag = 12 first, or just put all the commands into the plot() command

#directly.

#

#The best webpage I've found with examples of the diff() command is at

#https://atsa-es.github.io/atsa-labs/sec-tslab-differencing-to-remove-a-trend-or-seasonal-effects.html

#

par(mfrow = c(1,2))

d12 <- diff(PassengerMiles.ts, lag = 12)

plot(d12, ylab = "Lag-12,

then Lag-1", xlab = "Time", bty = "l", xlim = c(1991,2024.25),

main = "Differenced (Lag-12)")

plot(diff(PassengerMiles.ts, lag = 12), ylab = "Lag-12,

then Lag-1", xlab = "Time", bty = "l", xlim = c(1991,2024.25),

main = "Differenced (Lag-12)")

#

#You can see from the side-by-side plots that either syntax returns the same

#result.

#

par(mfrow=c(1,2))

plot(diff(diff(PassengerMiles.ts, lag = 12), lag = 6), ylab = "Lag-12,

then Lag-1", xlab = "Time", bty = "l", xlim = c(1991,2024.25),

main = "Twice-Differenced (Lag-12, Lag-1)")

plot(diff(diff(diff(PassengerMiles.ts, lag = 12), lag = 6), lag = 3), ylab = "Lag-12,

then Lag-1", xlab = "Time", bty = "l", xlim = c(1991,2024.25),

main = "Three times-Differenced (Lag-12, Lag-1, Lag-6)")

#

#Here is the 2nd way to make a "seasonal adjustment" to time series data. It uses

#data from the decompose() command.

#

PassengerMiles.SeasAdj <- PassengerMiles.ts - Amtrak.comp$seasonal

par(mfrow = c(2,2))

plot(PassengerMiles.ts, xlab = "Time", ylab = "Passenger Miles", type = "l")

plot(PassengerMiles.SeasAdj, xlab = "Time", ylab = "Passenger Miles", type = "l")

PassengerMiles.seas.zoom <- window(PassengerMiles.SeasAdj, start = c(1997, 1), end = c(2000, 12))

plot(PassengerMiles.seas.zoom, xlab = "Time", ylab = "Seasonally adjusted passenger miles", type = "l")

par(mfrow = c(1,1))

#

#Again, there doesn't appear to be a obvious periodicity in this zoomed in plot.

#

#

#Here is the 3rd way to make a seasonal adjustment. It is from your textbook,

#Section 3.5. To use this you will need to install the "seasonal" package in R/

#RStudio. Also, to use the "seasonal" package we need to transform out time

#series data from .ts into a tsibble.

#

PassengerMiles.tsb <- as\_tsibble(PassengerMiles.ts)

PassengerMiles.tsb

x11\_dcmp <- PassengerMiles.tsb |>

model(x11 = X\_13ARIMA\_SEATS(value ~ x11())) |>

components()

autoplot(x11\_dcmp) +

labs(title =

"Decomposition of total Amtrak Passenger Miles using X-11.")

#

#This decomposition looks a little different than the decomposition using the

#decompose() command. Notice the spike in the trend data just before the year

#2015? It isn't nearly as sharp of a spike in the trend using the decompose()

#command from the "forecast" package.

#

#Let's finish this off with a nice way to illustrate the decomposition of a

#time series.

#

x11\_dcmp |>

ggplot(aes(x = index)) +

geom\_line(aes(y = value, colour = "Data")) +

geom\_line(aes(y = season\_adjust,

colour = "Seasonally Adjusted")) +

geom\_line(aes(y = trend, colour = "Trend")) +

labs(y = "Passenger Miles",

title = "Total Amtrak Passenger Miles") +

scale\_colour\_manual(

values = c("gray", "#0072B2", "#D55E00"),

breaks = c("Data", "Seasonally Adjusted", "Trend")

)

#

#The zoomed in figure isn't a very pretty figure but it is hard to see any

#repeating pattern in it that would suggest any residual periodicity. You might

#have to make more than one adjustment to the data. What you are doing is

#removing the trend and seasonality to make the data stationary. The textbook

#does a good job of covering what this means but you may have to read and reread

#a couple sections to get it clearly set in your mind.This is an important

#part of time series analysis so you want to make sure you really do

#understand what it means for your data to be stationary and what you have to

#do to achieve that!

#

PassengerMiles.lm <- tslm(PassengerMiles.ts ~ trend + I(trend^2))

summary(PassengerMiles.lm)

plot(PassengerMiles.ts, xlab = "Time", ylab = "Passenger Miles", type = "l")

lines(PassengerMiles.lm$fitted, lwd = 2)

#

#From this plot it doesn't look like a quadratic model fits the data fairly well.

#Remember the last plot from the first laboratory?

#It was pretty obvious that the quadratic was not really working. So, let's add

#a cubic term and see if we get the same result as the last lab.

#

PassengerMiles.lm <- tslm(PassengerMiles.ts ~ trend + I(trend^2) + I(trend^3))

summary(PassengerMiles.lm)

plot(PassengerMiles.ts, xlab = "Time", ylab = "Passenger Miles", type = "l")

lines(PassengerMiles.lm$fitted, lwd = 2)

#

#It doesn't capture everything but it does look better than the quadratic fit.

#