Stat 153 HW1

September 16, 2020

1 Homework 1, Stat 153

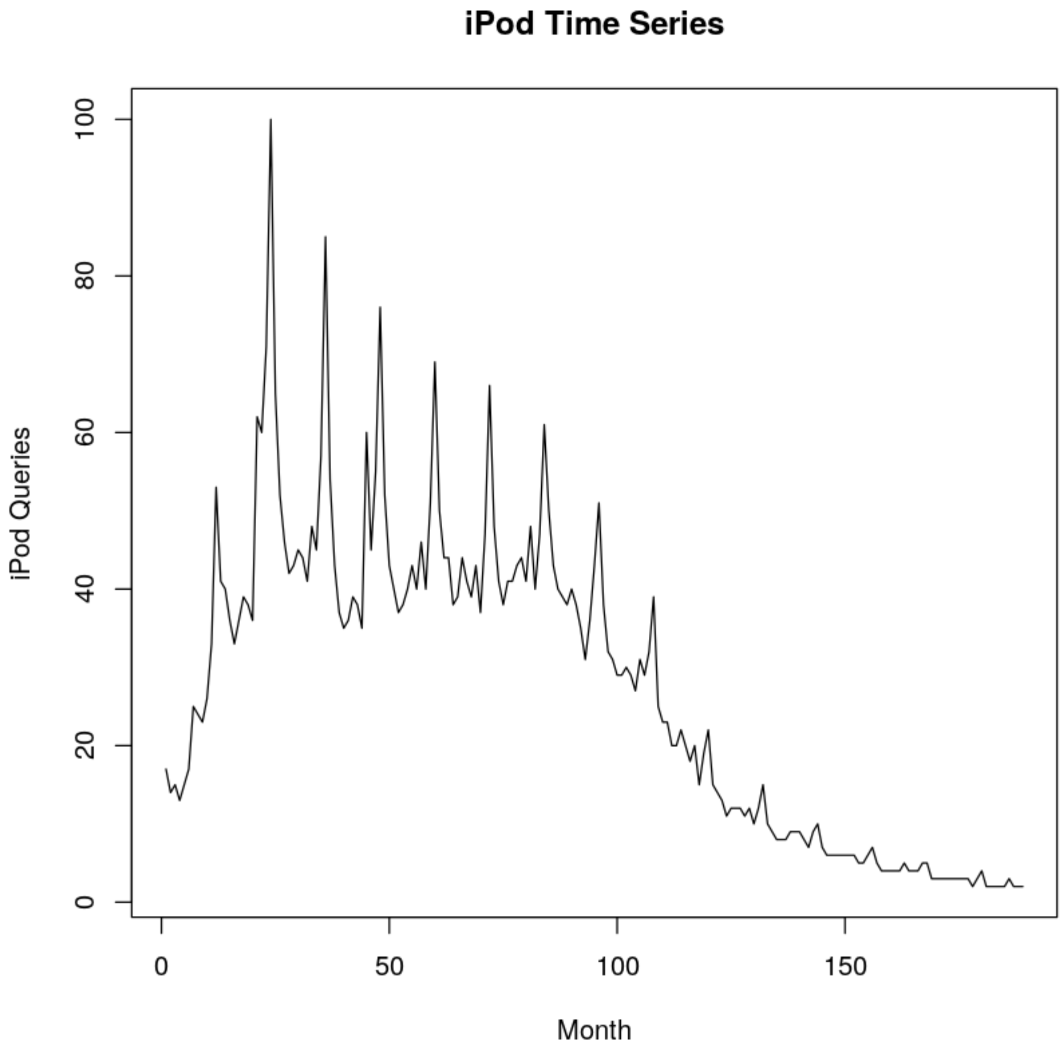
1.0.1 Robbie Netzke

[1]:
```
ipod_data <- read.csv("iPod.csv")
```

1.1 1A

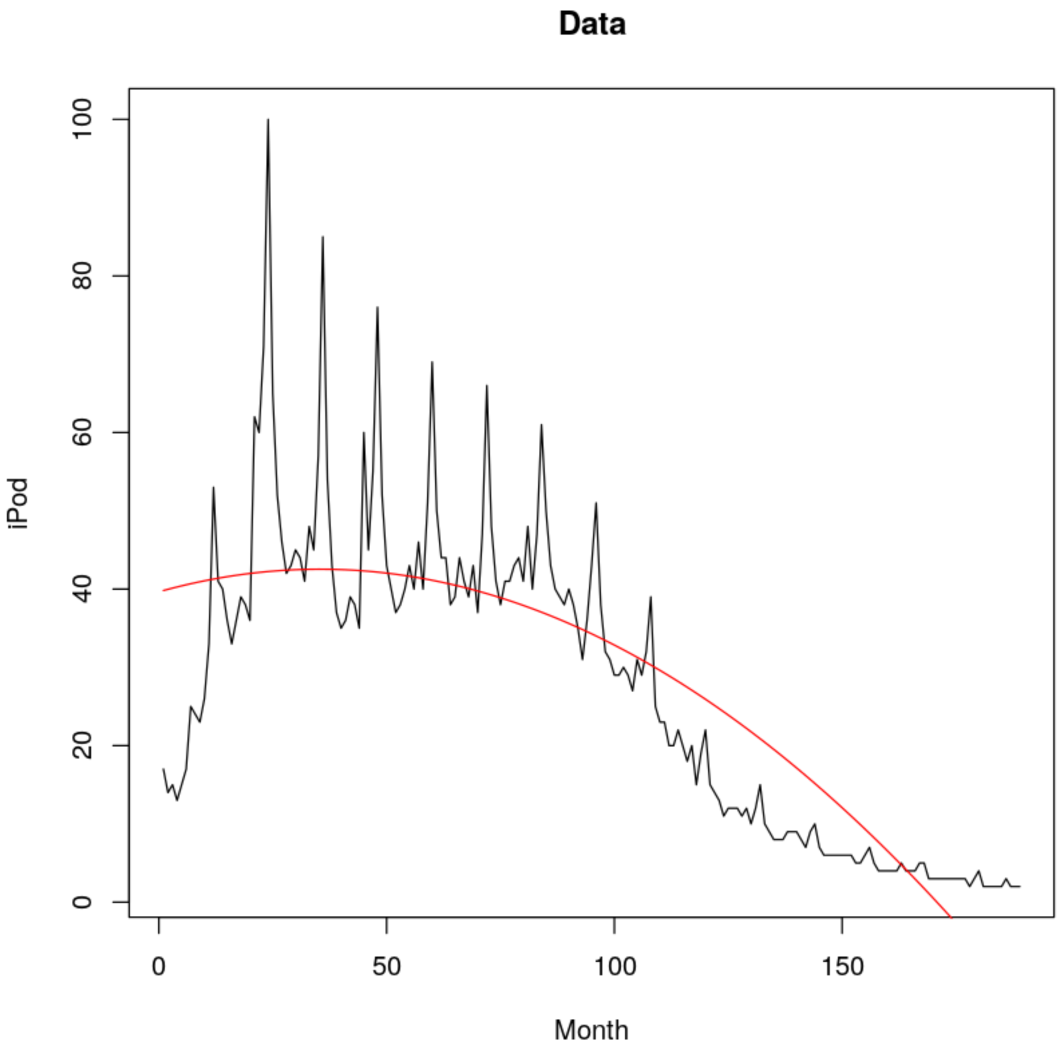
[2]:
```
# 1
# A
month_data <- 1:189
ipod_d <- as.vector(ipod_data$iPod)
plot(month_data, ipod_d, xlab = "Month", ylab = "iPod Queries", main = "iPod␣
 ↪Time Series", type="l")

## I will begin by plotting the time series to become more familiar with it.
## Months are represented as integers 1-189
```



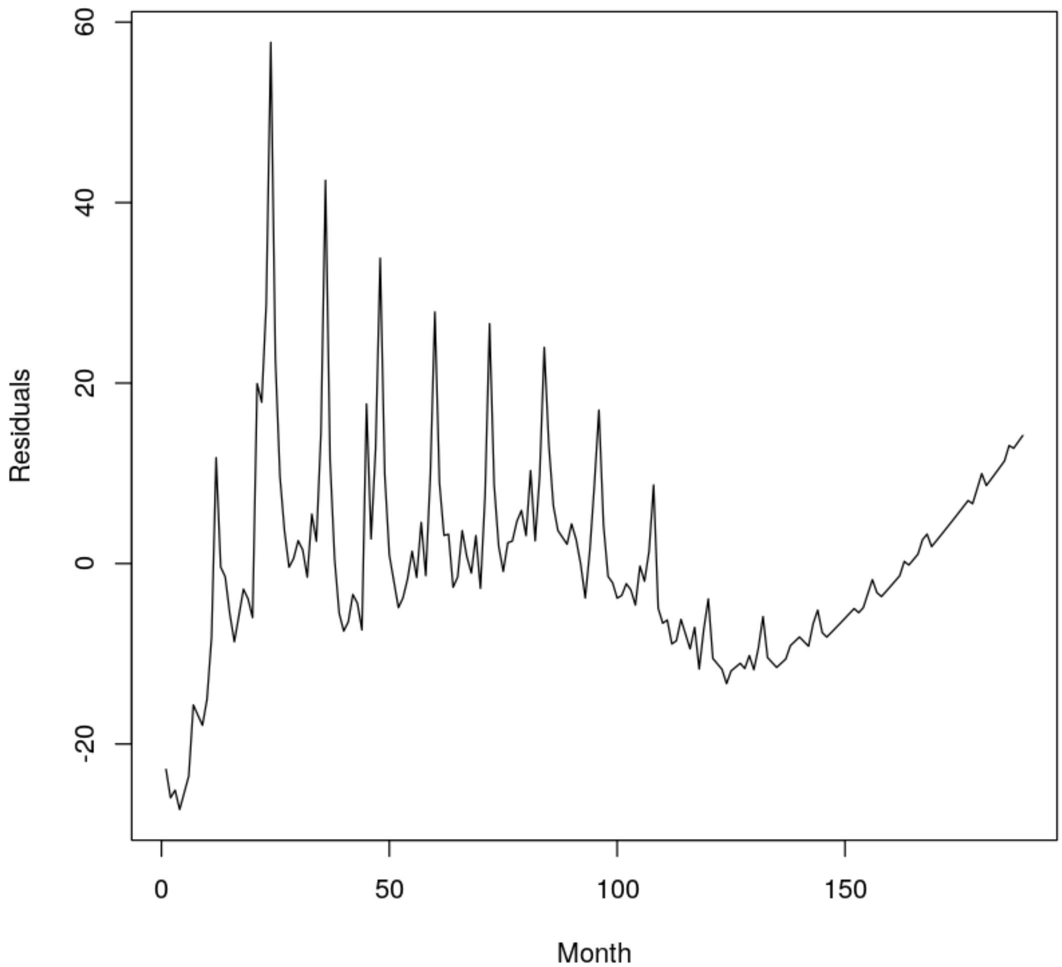
Here I fit the parametric estimation function: t^2 - log(5t) + c, for constant c

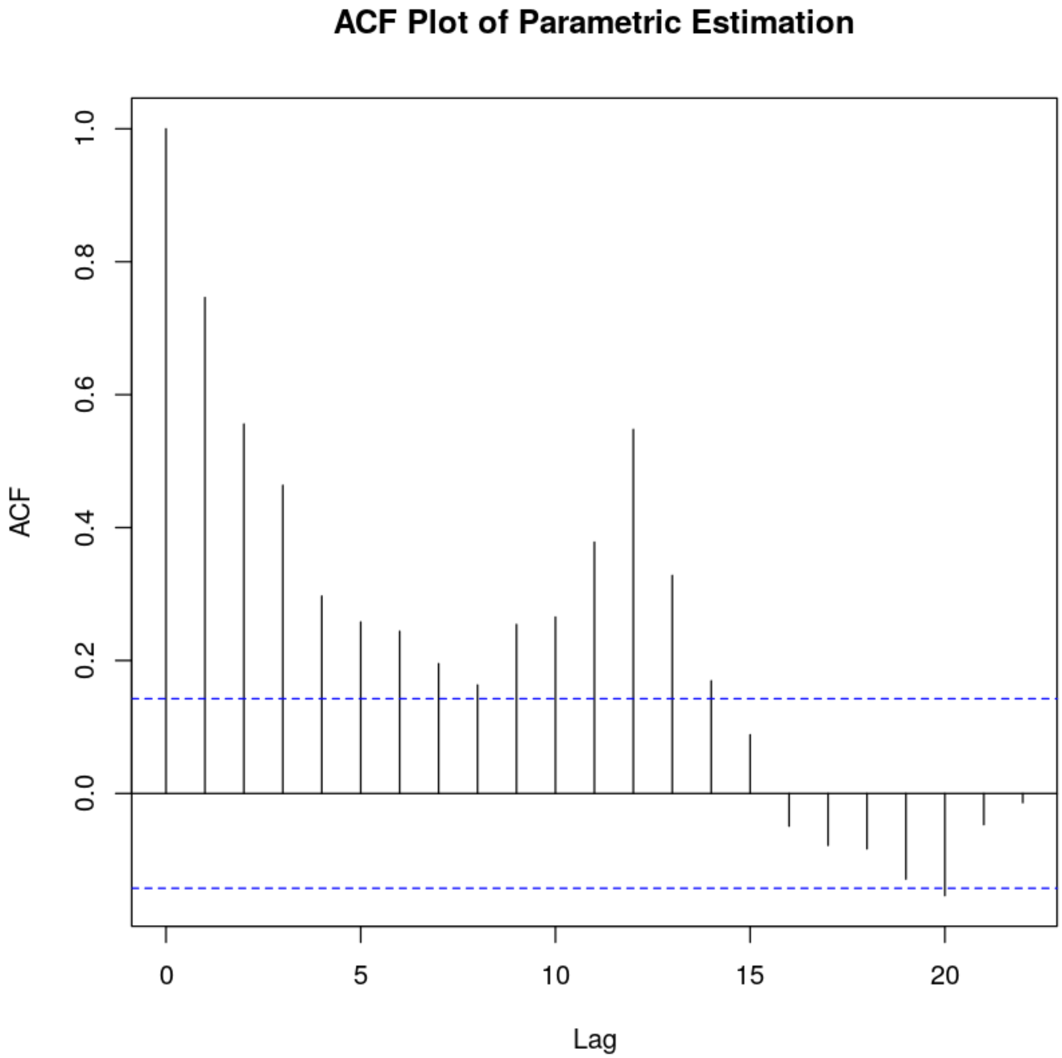
[3]:
```
func <- - month_data^2 - log(5*month_data) + 50
model1 = lm(ipod_d ~ func + month_data + 1)
plot(month_data, ipod_d, xlab = "Month", ylab = "iPod", main = "Data", type="l")
lines(month_data, model1$fitted.values, col = "red")
```




```
[4]: plot(month_data, model1$residuals, xlab = "Month", ylab = "Residuals", main =␣
      ↪"Residual Plot of Parametric Estimation Function", type = "l")
     acf(model1$residuals, main = "ACF Plot of Parametric Estimation")
```

Residual Plot of Parametric Estimation Function

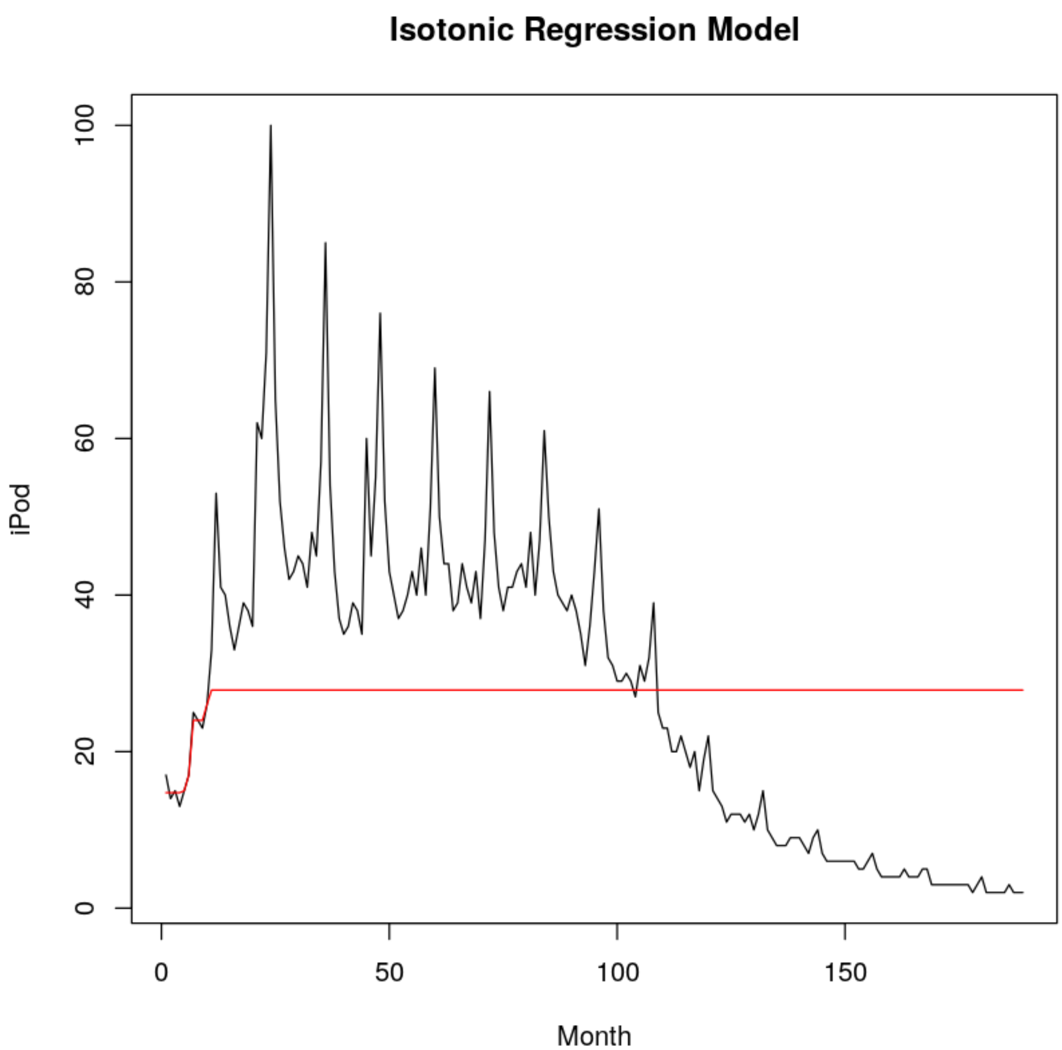




Comments: In the plot of the residuals, we see a non-zero covariance between residuals, especially for t on the interval 150-189. From the ACF, the sample autocorrelation is not approximately 0 for the majority of t, so it seems the residuals of our model do not possess properties of white noise. I would assume that the expected value of the residuals depends on time t, and the covariance varies for different lags, so I expect the residuals are not a realization of a weakly stationary time series. Visibly, from the residual plot we can see a clear change in the pattern of residuals over time, and the ACF demonstrates a trend not captured by our signal function.

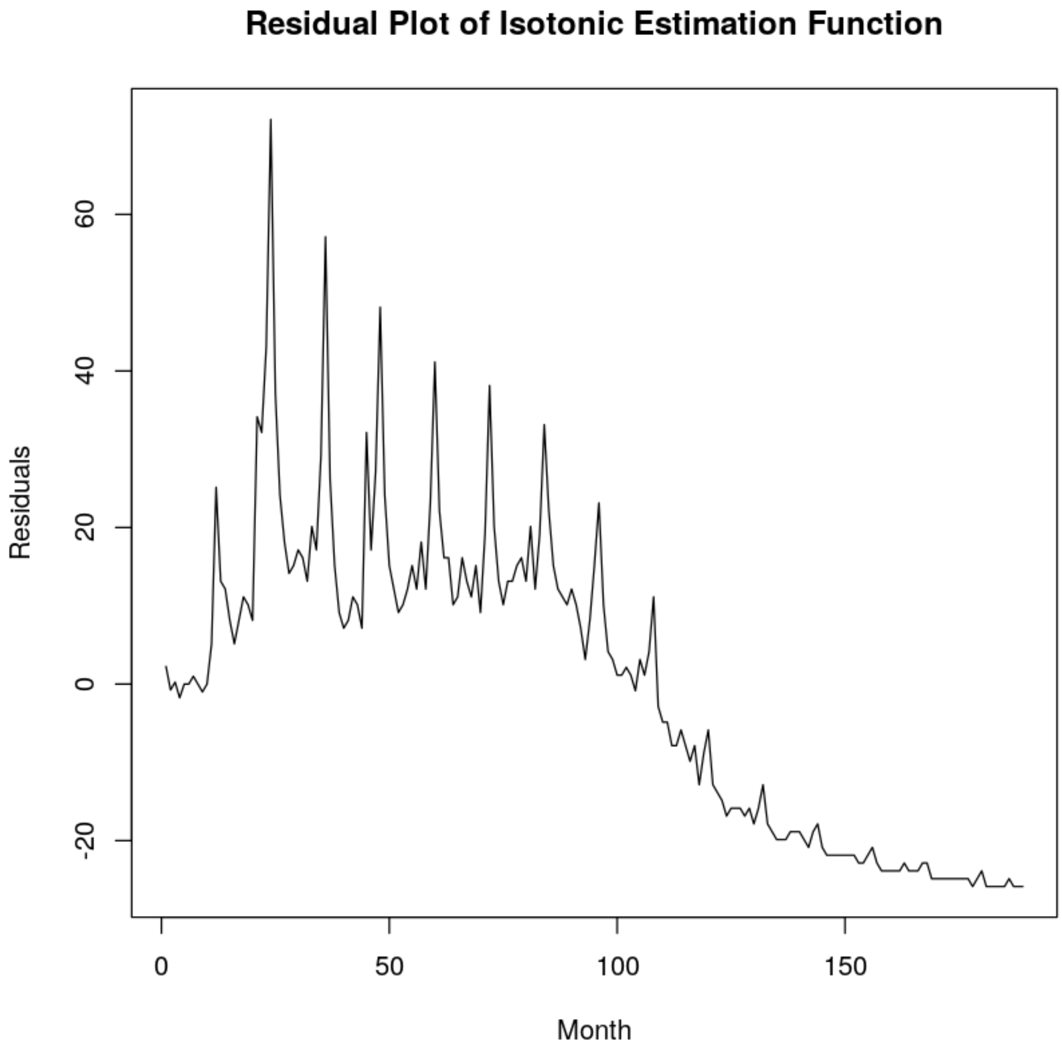
1.2 1B

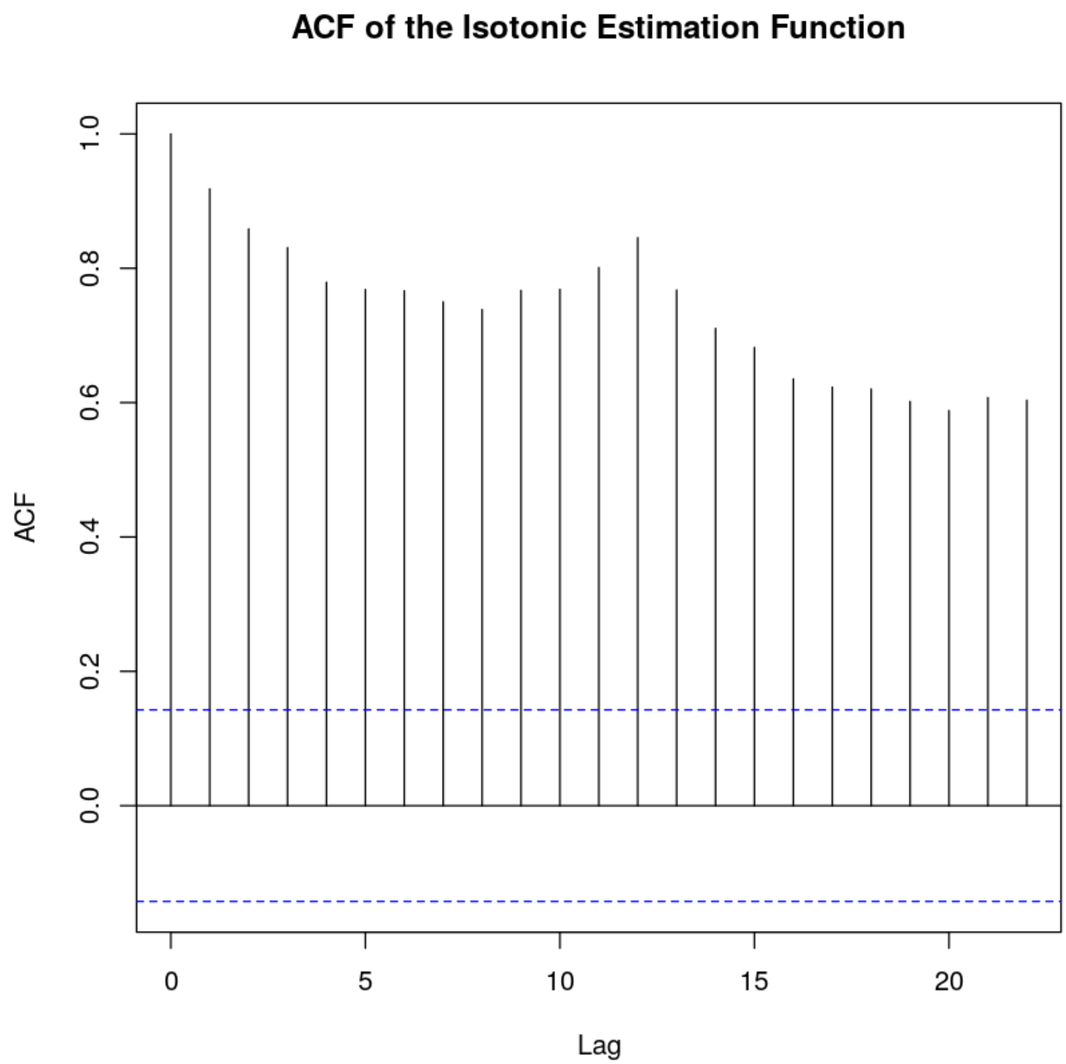
[5]:
```
model1b <- isoreg(month_data, ipod_d)
plot(month_data, ipod_d, xlab = "Month", ylab = "iPod", main = "Isotonic␣
↪Regression Model", type="l")
lines(month_data, model1b$yf, col = "red")
```



Plotting the residuals and ACF for the isoregressive model.

[6]:
```
bresiduals <- ipod_d - model1b$yf
plot(month_data, bresiduals, xlab = "Month", ylab = "Residuals", main =␣
↪"Residual Plot of Isotonic Estimation Function", type = "l")
acf(bresiduals, main = "ACF of the Isotonic Estimation Function")
```


Residual Plot of Isotonic Estimation Function
Residuals
Month
60
40
20
0
-20
0
50
100
150

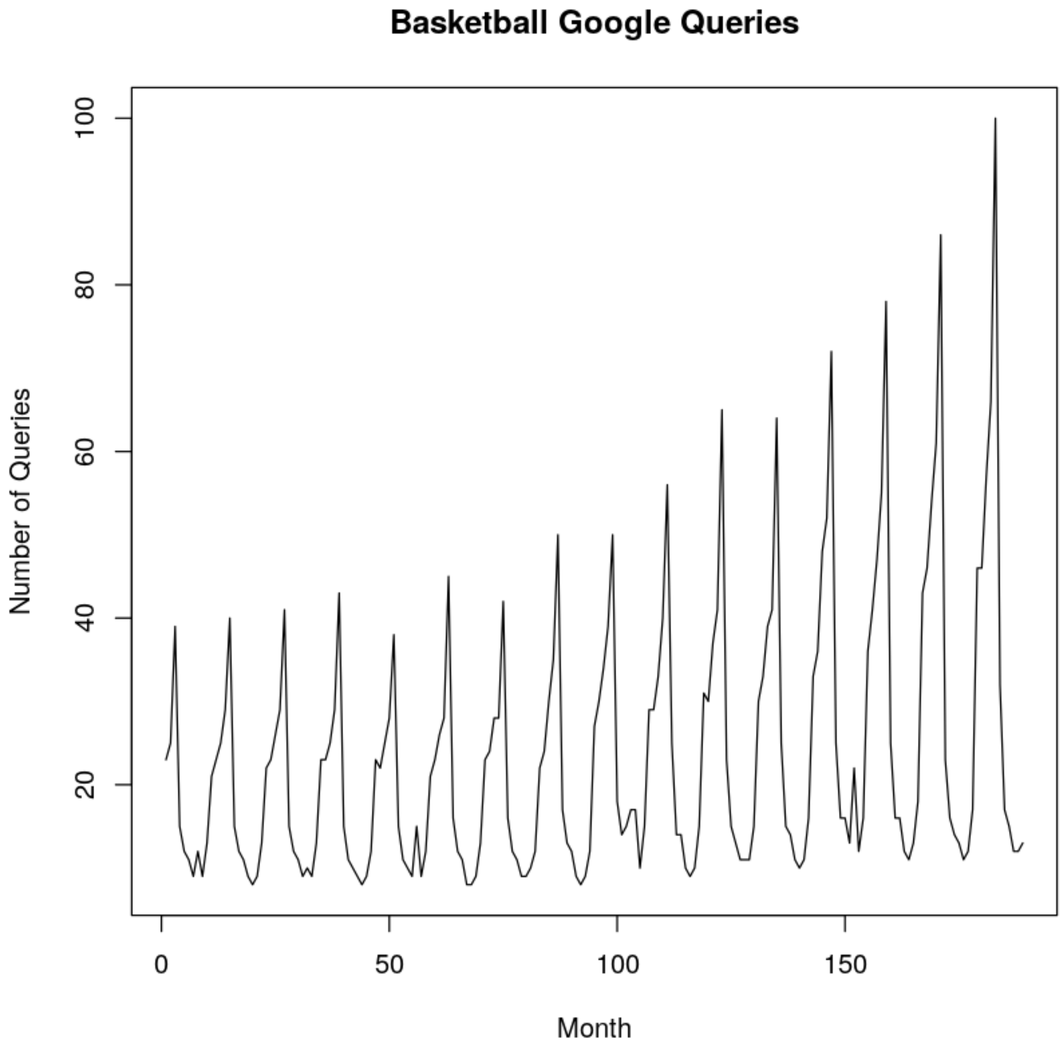


Comment: The isotonic regression model does not capture trend or seasonality as demonstrated by the residual plot. When the number of queries decreases, we have no chance at capturing this trend under the definition of isotonic trend estimators, since we may not decrease our number of queries estimated over time. Our noise has no chance of being stationary, as the sample expected value of the residuals varies over time and the residuals, I suspect, have varying covariances per choice of lag. To further reinforce the notion of non-stationary noise, the correlogram shows non-zero autocorrelation for all times t, so the noise cannot possibly be white noise.

1.3 2A

[7]:
```
bball_data <- read.csv("Basketball.csv")
```

[8]:
```
plot(month_data, bball_data$Basketball, xlab = "Month", ylab = "Number of␣
↪Queries", main = "Basketball Google Queries", type="l")
```

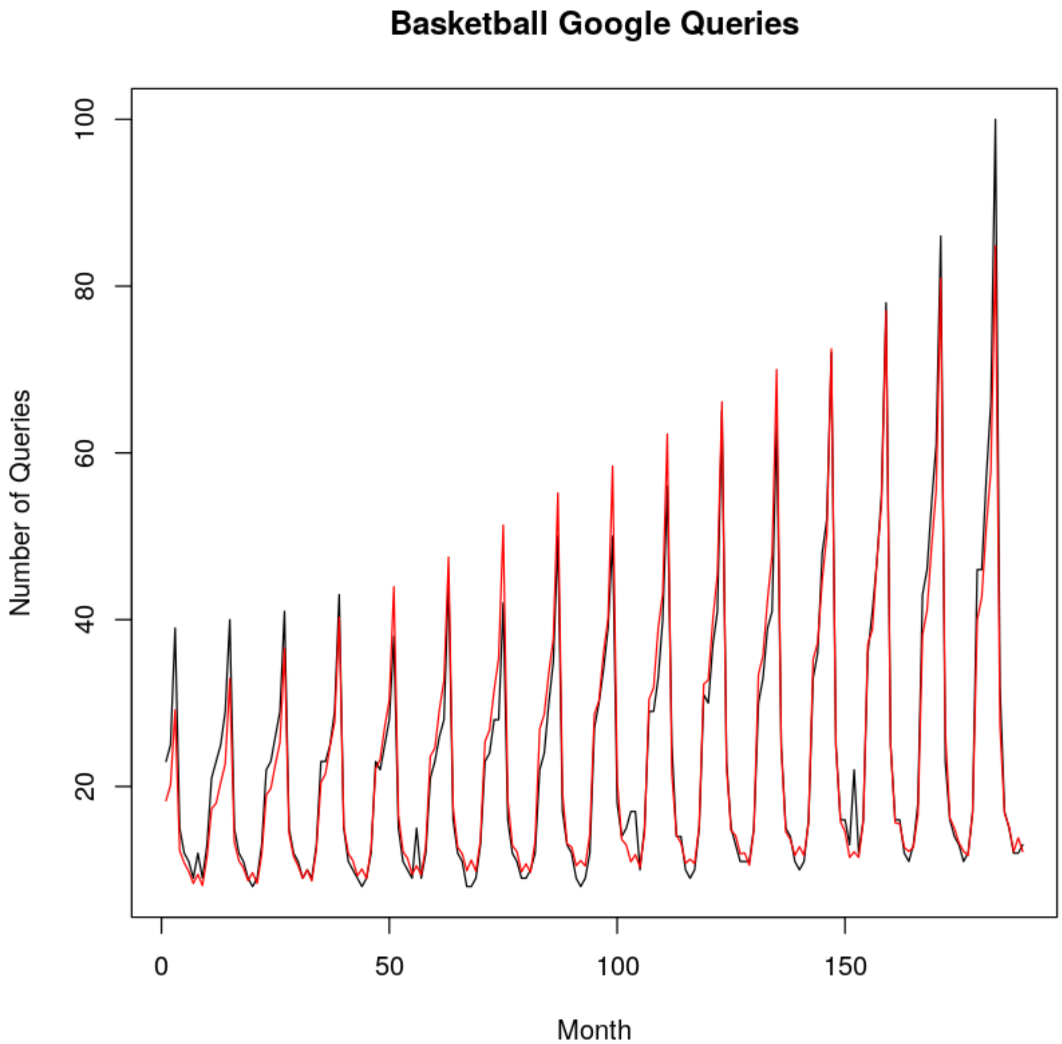


Comment: We see seaonality roughly every year, or, in terms of sinusoid functions, a seasonality trend for a period of 12 months. I assume that many use this query to update themselves on games throughout the season, and not very often when basketball is not in season. We see that seasonality holds for the minimum queries of the periods, but the maximums of each period do not follow the strict definition of seasonality. Furthmore, we see an upward trend in maximum number of queries per period, and this trend appears exponential or quadratic; however, the smallest amount of queries per period remains around the same, following our definition of seasonality.

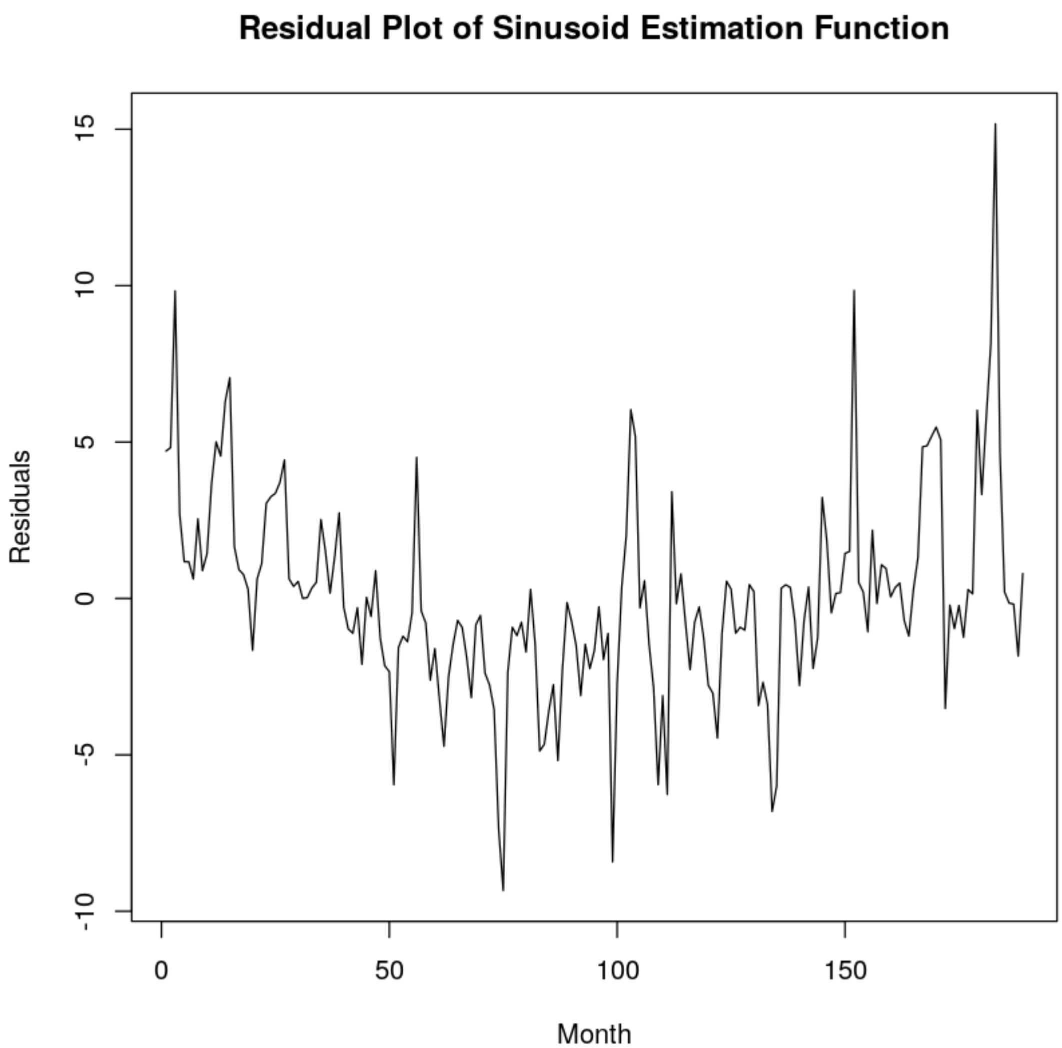
1.4 2B

I sought to improve the model by having time ”interact” with the sinusoid. I preferred the fit using this feature of R

[10]:
```
plot(month_data, bball_data$Basketball, xlab = "Month", ylab = "Number of␣
↪Queries", main = "Basketball Google Queries", type = "l")
model2b <- lm(bball_data$Basketball ~ 1 + month_data + month_data*(␣
↪cos(2*pi*month_data/12) + sin(2*pi*month_data/12) + cos(2*pi*month_data*2/
↪12) + sin(2*pi*month_data*2/12)+ cos(2*pi*month_data*3/12) +␣
↪sin(2*pi*month_data*3/12) + cos(2*pi*month_data*4/12) +␣
↪sin(2*pi*month_data*4/12) + cos(2*pi*month_data*5/12) +␣
↪sin(2*pi*month_data*5/12)+ cos(2*pi*month_data*6/12) + sin(2*pi*month_data*6/
↪12)))
lines(month_data, model2b$fitted.values, col = "red")
plot(month_data, model2b$residuals, xlab = "Month", ylab = "Residuals", main =␣
↪"Residual Plot of Sinusoid Estimation Function", type = "l")
```



Basketball Google Queries
Number of Queries
Month
0
50
100
150
20
40
60
80
100



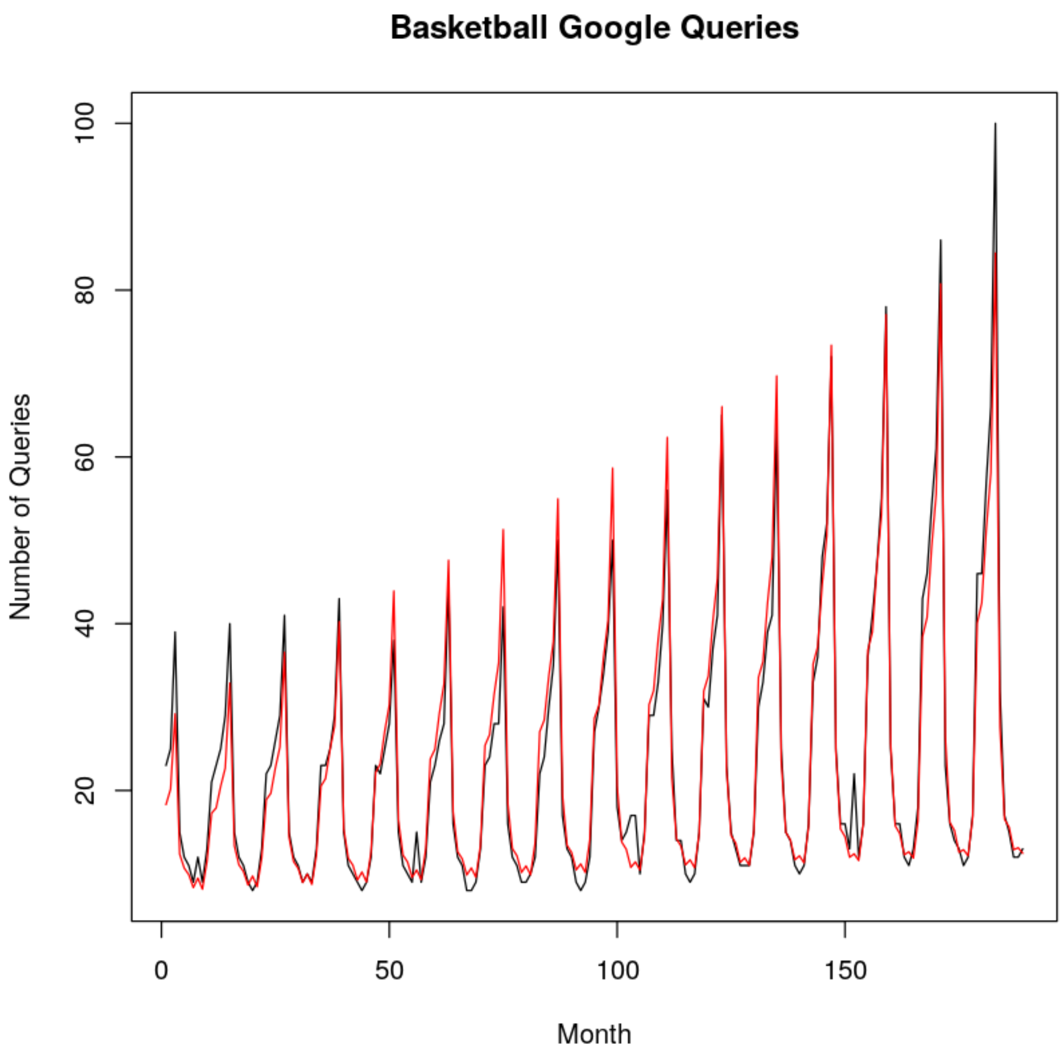
1.5 2C

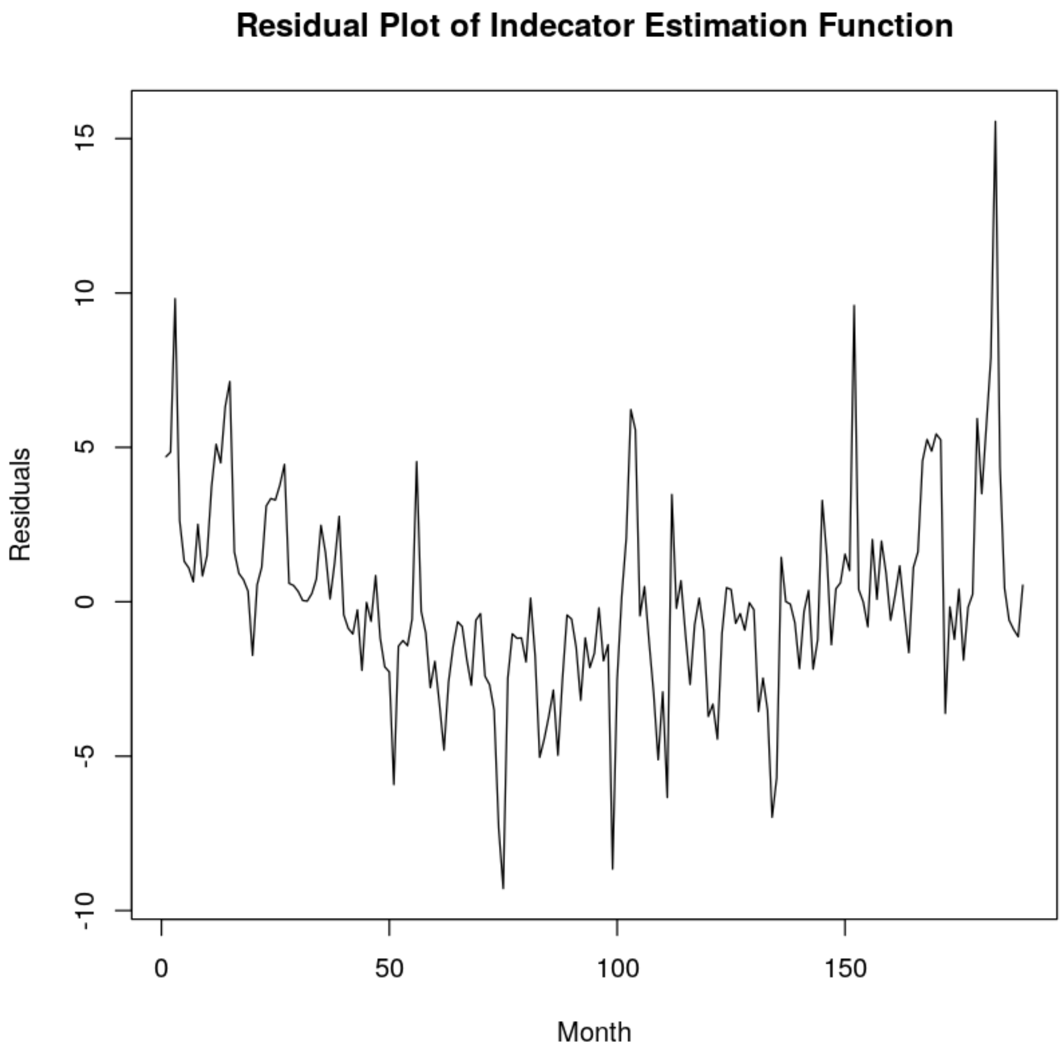
Again, I sought to improve the model by having time "interact" with the month indicators.

[9]:
```
plot(month_data, bball_data$Basketball, xlab = "Month", ylab = "Number of␣
↪Queries", main = "Basketball Google Queries", type = "l")
months <- bball_data$month
jan <- ifelse(months == "January",1,0)
feb <- ifelse(months == "February",1,0)
mar <- ifelse(months == "March",1,0)
apl <- ifelse(months == "April",1,0)
may <- ifelse(months == "May",1,0)
```

```r
jun <- ifelse(months == "June",1,0)
jul <- ifelse(months == "July",1,0)
aug <- ifelse(months == "August",1,0)
sep <- ifelse(months == "September",1,0)
oct <- ifelse(months == "October",1,0)
nov <- ifelse(months == "November",1,0)
dec <- ifelse(months == "December",1,0)
model2c = lm(bball_data$Basketball ~ 1 + month_data + month_data*( jan + feb +
 mar+ apl+ may+ jun+ jul+ aug+ sep+ oct+ nov+ dec))
lines(month_data, model2c$fitted.values, xlab = "Month", ylab = "Number of
 Queries", main = "Basketball Google Queries", type = "l", col="red")

plot(month_data, model2c$residuals, xlab = "Month", ylab = "Residuals", main =
 "Residual Plot of Indecator Estimation Function", type = "l")
```





1.6 2D

Comment: The fitted values are precisely the same as far as I can tell. The most prominant period of seasonality for the sinusoid function is 12 months, and this is exactly the number of indicators. Although I might be wrong, this seems to draw a relationship between the parametric model and indicator model for seasonality.

Theoretical questions on the next page! Sorry for any unnecessary scrolling