Tyler Cann

Stat 330 – Heaton

Particulate Matter Level Report

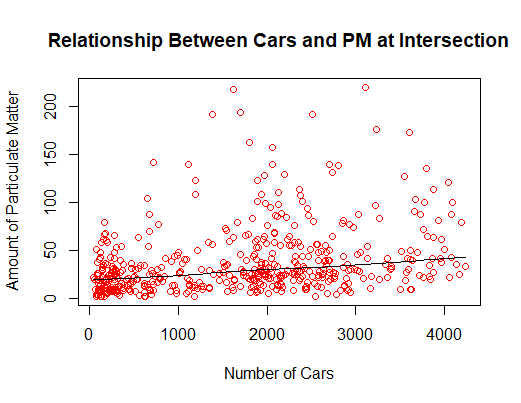
**Section 1: Introduction and Problem Background**

Our problem is that people are experiencing health problems because of exposure to particulate matter pollution. In order to measure the relationship between human activities and PM concentration in the air, we’re going to analyze data we’ve collected from cars passing through an intersection. As a result, we hope to find out whether there’s a relationship between human activities and PM concentration, as well as whether traffic levels can be used to predict PM levels. After the analysis we will have the data necessary to guide people on what actions to take to lower PM levels.

We are going to use the PM.txt data we collected from the intersection traffic to fulfill our goals stated above.

We want to look at the relationship between the number of cars passing through an intersection and the number of PM particles at the same intersection. First we want to analyze the data using some simple graphics and summaries.

First we want to check how linear the data is, so we’ll run a scatterplot of the data:



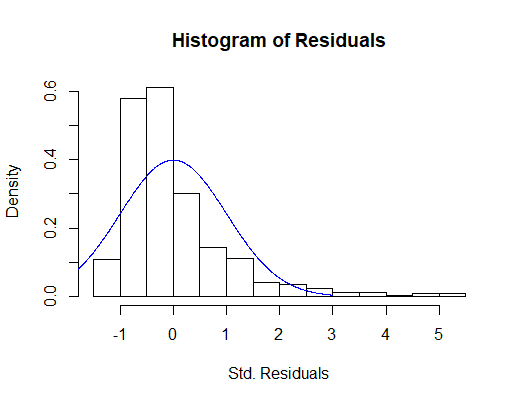
The above plot is a summary of the data we were given. We can that there’s a linear relationship between the number of cars and particles, but it’s a very weak one. While it IS weak, it’s still straight and can be considered linear. We may have to transform the data to observe a stronger relationship.

The correlation coefficient, a numerical representation of the relationship, is only 0.3 (a 0 means there’s no relationship, 0.5 means there’s a moderate relationship, and 1 means there’s direct causation). Given that 0.3 is lower than 0.5, we know it’s weak.

Next, we want to check whether the data is independent.

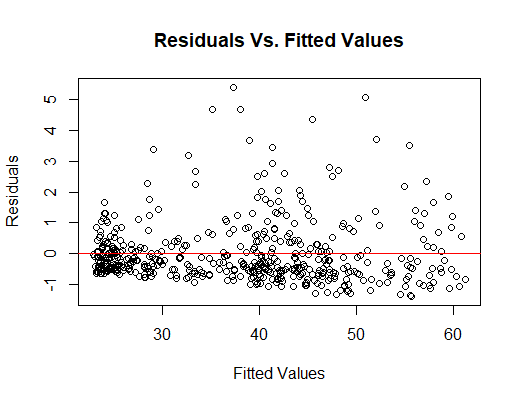
Given that each dot represents different moments when cars had passed through the intersection, they wouldn’t have affected each other. Therefore, the data is independent.

Third, we want to check normality, so we’ll run a histogram of the residuals of the plot:



The histogram above shows that the data isn’t normal. It’s right skewed, and doesn’t follow the bell curve imposed on it very well, even having outliers at the end of the tail. A normal histogram would have its highest part lie more towards the middle and spread out evenly.

Next, we want to check if the data has equal variance, so we’ll plot the residuals vs fitted values:



This plot shows that the data isn’t scattered equally. Most of it falls beneath the line we drew through the middle, so there isn’t equal variance. If there *was* equal variance, then there would be an even distribution of circles on both sides of the line.

With the data in its current state, a simple linear regression (SLR) model would NOT be appropriate. The histogram is skewed, and the data isn’t equally variant. In order to run linear regression, we’re going to have to transform this data. A suitable transformation would be a logarithmic transformation. That means that we’re going to use a natural log function like you’d find in a calculator and run the data through it to see how the data looks after.

**Section 2: Statistical Modeling**

The Linear Regression Model that we’ll use is **log(yi) = β0 + β1\*log(xi) + ϵi**

**yi** represents our ith observed amount of PM Particles. *(This is the amount of PM we can expect after measuring the number of cars in a certain area.)*

**β0** represents our intercept coefficient. When our log(number of cars) is 0, our log(amount of PM Particles) would be this value on average. *(When there are no cars, we can expect this to be the amount of PM to occur.)*

**β1** represents our slope coefficient. For every 1 more car going the intersection [log(number of cars)], our

log(amount of PM Particles) would increase by this value on average. *(For each car going through a certain area, we can expect an increase in PM represented by this* ***β1*** *value)*

**xi** represents our explanatory variable, or rather, the ith number of cars in the intersection. *(This represents how many cars are in the area)*

**ϵi** is the error for the ith observation, which are distributed with equal variance and a mean of zero,

given by the equation: **ϵi ~iid N(0,σ2)** *(This simply helps us measure how far off our model is from what’s the “true value”)*

We must follow the following assumptions to follow this model:

1 – The relationship between the number of cars and pm particles is linear.

2 – Our observations are independent of each other.

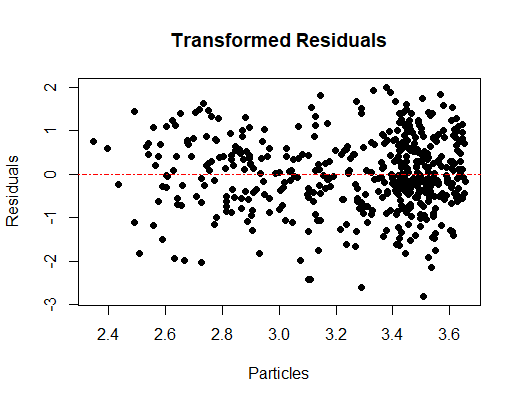
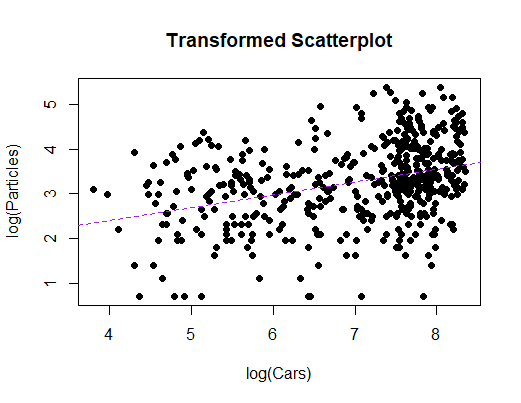
3 – Our residuals are normally distributed.

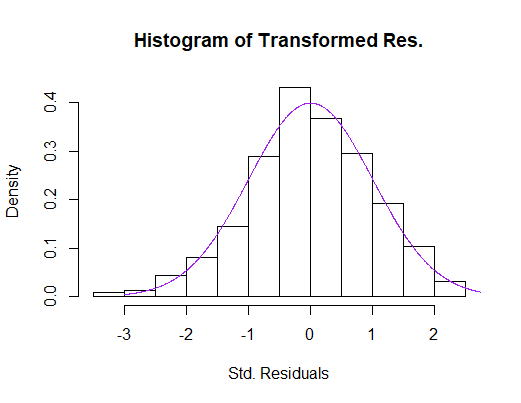
4 – There’s equal variance among our predicted values.

(*Our logarithmic transformations allows us to make these assumptions)*

**Section 3: Model Verification**

As mentioned previously, we fitted our assumed model using various transformations.





As seen in the two plots and the histogram above, our assumptions are justified.

Our linearity assumption is still met with the transformed scatterplot. We also got a correlation coefficient (r) of 0.36. This shows that there’s still weak correlation, but it’s stronger than in the original model, and supports a linear relationship.

The data was independent, also mentioned previously. Nothing changed in that regard.

Normality is met because of our transformation. It follows a bell curve, unlike previously, and has no outliers. Also, all of the data lies within 3 standard deviations.

It looks like the data is equally variant this time. Both sides of plot look to have about the same amount of data. There appear to be some outliers near the bottom of the plot, and the data tends to bunch up more on the right side, but these things are ignorable.

We also calculated an R2 of about 0.12. This means that about 12% of the variation in the amount of PM particles can be explained by the number of cars going through the intersection. Because this number is closer to 0, we learn that there isn’t a very strong relationship between the number of cars and the amount of PM particles.

*(it’s important to note here that I transformed the data by square rooting it, and got an R2 of about 0.11 the residual plot didn’t become equally variant, and the histogram had an outlier, so I stuck with the logarithmic transformation)*

We ran a cross validation procedure to help us know the predictability of our model. This means we took a sample of data points from our original data set and used the remaining data as our “training data set.” We fit a model to our training data and then looked to see how far off our predictions would be from our sample of data points we took. We repeated this process 100 times to get a good idea of how well our model predict on average. This gave us an average bias of -5.818 pollution particles. We can interpret this as meaning that our model tends to underpredict by this value on average. Our root predictive mean square error(RPMSE), how much our model is off on average, was 34.13 pollution particles. So, on average, our model is off by this number. We’re dealing with pollution particles going into the hundreds, so our even though the 34.13 value is large, we can still predict with it and not be too far off.

The model fits the data well, as evidenced in the previous paragraphs of section 3, along with the graphics of the transformed data. However, this data we’re analyzing just doesn’t appear to have very strong correlation, so it’ll be difficult to make predictions.

Honestly, our model doesn’t do very well at predicting accurately though. According to our model, our R2 is only 0.12, so only about 12% of the pm particles can be explained by the number of cars. 12% won’t tell you very much. Our RMPSE was fairly large too. Most of our data lies within the 100 or less PM particle range, so being off by 34.13 means that you’re potentially off by quite a bit.

**Section 4: Results**

There’s a relationship between PM and the number of cars. It’s just a weak one. To verify this, we performed a test of significance on our slope, to determine if our there was a relationship.

H0: β1 =0 (Meaning there is no linear relationship between cars and PM particles)

Ha: β1 0 (Meaning there is a linear relationship between cars and PM particles)

We calculated a p-value of 4.76E-16, or essentially 0. Our p-value is less than alpha of .05, so we would reject the null hypothesis and conclude that there is a linear relationship between snowfall and water runoff.

Here’s an estimate of what that relationship would be:

A 95% confidence interval for the intercept () is (0.7759, 1.7228). Thus, we are 95% confident that if there were no cars driving through the intersection, the amount of PM particles would be between (0.7759, 1.7228) on average.

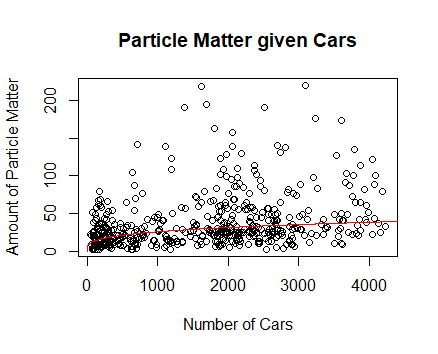
A 95% confidence interval for the slope () is (0.2215, 0.3549). Meaning, we are 95% confident that the change in cars in the intersection by one would be between (0.2215, 0.3549) on average.

Our model is able to make specific predictions. After finding the coefficients of our transformed model, we get the following equation: **Log(PM Particles) = 1.249 + 0.2882 \* log(number of cars)**

1.249 is the intercept coefficient **β0**. So when there are no cars going through the intersection, there would be 1.25 particles on average.

0.2882 is the slope coefficient **β1**. For every extra car going through the intersection, there would an increase of .2882 PM particles on average.

Below is our fitted regression line given the original scale of the data:



If we were to predict the PM concentrations at an intersection where there are 1800 cars passing through, we would get a prediction interval of (1.778, 5.042). This means we are 95% confident that the PM concentrations for a intersection with 1800 cars passing would fall between 1.778 and 5.042 on average.

**Section 5: Conclusions**

We’ve found that there IS a relationship between the number of cars passing through an intersection and the amount of PM concentrations there. It was a weak relationship that required mathematical transformations to fully analyze, but we we’re able to draw certain conclusions from it. We also created a mathematical model that will let us predict the levels of PM concentrations of similar intersections in the future. The model’s predictions won’t always be accurate, but will give us a ballpark idea of what to expect.

1) Environmental scientists should gather more data with multiple factors to better determine the relationship between cars and PM. Measuring at different times of the day, with different intersections may help determine this better. If we were able to get a stronger linear relationship between the two, we could build a stronger model.

2) Until then, scientists should use this data to help educate people about the need to carpool more and lower the amounts of PM concentrations in the air.

R CODE:

### THIS SHOULD READ LIKE A REPORT

### NOT NUMBERED LIKE THE HW

### USE THE SECTION HEADINGS IN THE EXPLANATION

setwd("C:/Users/thety/Desktop/330/exam1")

pm <- read.table(file="PM.txt",header=TRUE)

#1b

scatter.smooth(pm$Cars,pm$Particles,col="red",

xlab="Number of Cars",ylab="Amount of Particulate Matter",main="Relationship Between Cars and PM at Intersection")

cor(pm$Cars,pm$Particles)

#lmpm <- lm(pm$Particles~pm$Cars)#put the y first, then the x

lmpm <- lm(Particles~Cars, data=pm)

coef(lmpm)

summary(lmpm)

abline(lmpm,col="darkgreen",lty=2,lwd=1)

#install.packages("lmtest")

library(lmtest)

bptest(lmpm)

library(MASS)

stres <- stdres(lmpm)

#cooks.Particles(lmpm)

#which(cooks.Particles(lmpm)>3) #if you have greater than 3 then you probs have outliers

hist(stres,freq=FALSE,main="Histogram of Residuals",xlab="Std. Residuals")

ressy <- seq(-3,3,length=1000)

lines(ressy,dnorm(ressy),col="blue")

plot(lmpm$fitted.values,stres,main="Residuals Vs. Fitted Values",xlab = "Fitted Values", ylab = "Residuals")

abline(0,0,lty=1,col="red")

#1c and 3a

##log

#

#

logpm <- lm(log(Particles)~log(Cars),data=pm)

logpm

summary(logpm) #0.12 r squared

plot(log(pm$Cars),log(pm$Particles),xlab="log(Cars)",

ylab="log(Particles)",main="Transformed Scatterplot",pch=19)

abline(logpm,col="purple",lty=2,lwd=1)

cor(log(pm$Cars),log(pm$Particles)) #0.356 cor

plot(logpm$fitted.values,logpm$residuals,pch=19, ylab = "Residuals", xlab = "Particles",main="Transformed Residuals")

abline(0,0,lty=4,col="red")

logstres <- stdres(logpm)

hist(logstres, freq=FALSE, main="Histogram of Transformed Res.", xlab="Std. Residuals")

resso <- seq(-3,3,length=1000)

lines(resso,dnorm(resso),col="purple")

##sqrt

#

#

#

#sqpm <- lm(sqrt(Particles)~sqrt(Cars),data=pm)

#sqpm

#summary(sqpm) #0.11 r squared

#plot(sqrt(pm$Cars),sqrt(pm$Particles),xlab="log(Cars)",

#ylab="log(Particles)",main="Transformed Scatterplot",pch=19)

#abline(sqpm,col="purple",lty=2,lwd=1)

#cor(sqrt(pm$Cars),sqrt(pm$Particles)) #0.346 cor

#plot(sqpm$fitted.values,sqpm$residuals,pch=19, ylab = "Residuals", xlab = "Particles",main="Transformed Residuals")

#abline(0,0,lty=4,col="red")

#sqstres <- stdres(sqpm)

#hist(sqstres, freq=FALSE, main="Histogram of Transformed Res.", xlab="Std. Residuals")

#resso <- seq(-3,3,length=1000)

#lines(resso,dnorm(resso),col="purple")

#more 3a

n.cv <- 100

bias <- rep(NA,n.cv)

rpmse <- rep(NA,n.cv)

for(i in 1:n.cv){

## Step 1 - split into test and training sets

obs.test <- sample(1:nrow(pm),round(.1\*nrow(pm)))

obs.test

test.data <- pm[obs.test,] #putting a blank after a comma will give you all of teh columns

test.data

training.data <- pm[-obs.test,]

training.data

## Step 2 - fit model to training data

my.model <- lm(sqrt(Particles)~Cars,data=training.data)

## Step 3 - predict for test data

test.predict <- predict.lm(my.model,newdata=test.data)^2

test.predict

## Step 4 - calculate bias and RMPSE

bias[i] <- mean(test.predict-test.data$Particles)

rpmse[i] <- sqrt(mean((test.predict-test.data$Particles)^2))

}

mean(bias) #-5.818

mean(rpmse) #34.13

#4a

summary(logpm) #4.76E-16

confint(logpm, level = .95)

coef(logpm)

hey <- seq(0,4500,length=500)

points <- predict.lm(logpm,newdata=data.frame(Cars=hey,Particles=1))

predx <- exp(points)

plot(pm$Cars,pm$Particles, main="Particle Matter given Cars",xlab="Number of Cars",ylab="Amount of Particle Matter")

lines(hey,predx,pch=19,col="red") #doing the exp undoes the log transformation that we made

#4b

predict.lm(logpm, newdata = data.frame(Cars=1800), interval = "prediction", level = .95)