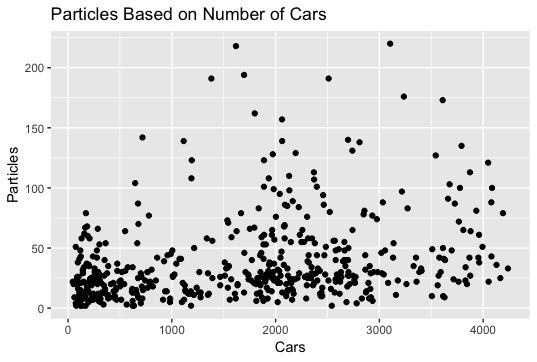
Daniel Witt

**Research Analysis – Pollution**

**Section 1: Introduction and Problem Background**

Particulate matter is a type of pollution that is made up of many chemicals, along with metal, soil, and dust. When a person is exposed to particulate matter (PM) in either the short-term or long-term, negative health affects have been detected. Health scientists know the amount of PM can affect health, but also want to know the affect that the amount of cars in the area has on the amount of PM. We want to study the relationship between the number of cars and the amount of PM particles in order to determine if this relationship exists. If we find that there is a relationship between the number of cars and the number of PM particles, we want to be able to use this relationship to be able to predict PM levels based on the number of cars in the area. This can help us understand how to combat PM levels, since we will know that the number of cars influence the number of PM particles.

In order to determine if there is a relationship between PM particles and the number of cars, we have collected a dataset that contains measurements of the amount of PM particles in the air and the associated number of cars in a certain intersection.

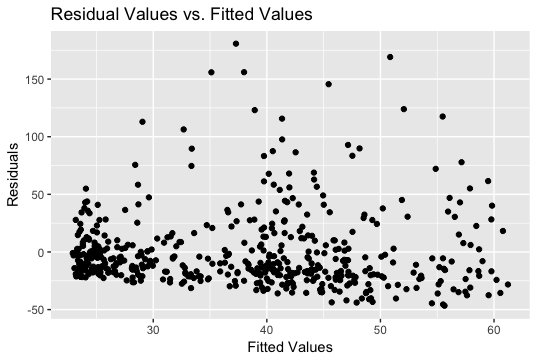


Above is initial graph of the data, with the number of cars along the x-axis and the number of PM particles along the y-axis. The data looks to be somewhat linear with a moderately positive relationship between the number of cars and the number of particles. This means that by looking at the graph, as the number of cars increases, the number of particles also increases.

The number of cars with associated number of PM particles has a covariance of 12174.15 and a correlation of 0.300898. These numbers mean that there is a positive relationship between the number of cars and the number of PM particles, however the relationship is weak, since the scale of correlation is from -1 to 1, with -1 being a perfectly negative correlation and 1 being a perfectly positive correlation.

In order to determine whether there is a relationship between the number of cars and the number of PM particles, we want to use a Simple Linear Regression model. This model will help us to understand the relationship and will also help us to be able to predict the amount of PM particles based on the number of cars after it is fit. In order to perform a Simple Linear Regression model, the assumptions of Linearity, Independence, Normality, and Equal Variance must be met after fitting the model.

First we will check linearity and independence by graphing the residual values with the fitted values of the simple linear regression model:

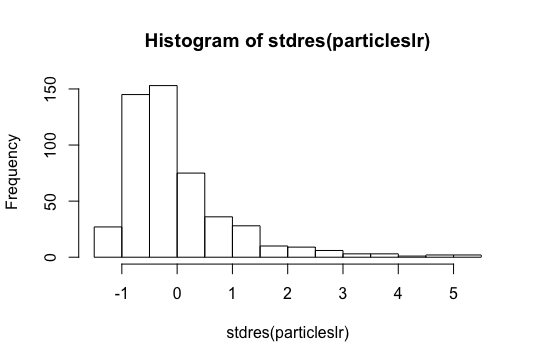


The assumption of linearity seems to be met, since the data does not seem to form any patterns other than linear. In other words, the data does not seem to be curving in one way or the other as a whole, it is mostly continuing in a line.

The data is independent of each other, since there are no patterns in the data. This means that each point does not depend on the points that came before it.

We can also assess the assumption of equal variance using the residual vs. fitted values plot. Equal variance means that the distance between each point and 0 on our graph is relatively equal. Looking at our plot, we can see that this is not the case. Many data points fan out much farther from 0 than other points, so it appears that the assumption of equal variance is not met. In order to confirm this, we conducted a BP test on the residual and fitted data, which helps determine whether the data is equal variance or not. We received a p-value of 0.003089. This value is statistically significant, since it is less than our significant p-value cut-off of 0.05. This means that the data is in fact, not equal in variance, so the equal variance assumption is not met.

We can assess normality by creating a histogram of the residual data:



The histogram of the data is clearly not normal in shape, with many values reaching up over 5 standard deviations from the mean of our data. We also conducted a KS test of normality on the data to confirm this result. We received a p-value of 2.588e-12, which is essentially 0. This is a significant p-value, which means that the data is not normal. Our assumption of normality is not met for this model.

Since two of our assumptions for our Simple Linear Regression model are not met, it would not be appropriate to use Simple Linear Regression on the data as it is. In order to make the data suitable for our model, we must use a transformation. We have chosen to use a natural log transformation to fit our data to the model.

**Section 2: Statistical Modeling**

In order to fit the data to an appropriate SLR model, we must make a transformation using the natural log. Therefore, the new SLR model will be:

*Ln(Yi )= β0 +β1 \*ln( Xi )+ εi*

*εi ~ Ν(0, σ2)*

Where:

Yi = Number of particles for the ith number of cars

Xi = Number of cars for the ith number of cars recorded

i =1, …, n

n = Sample size

β0 = Intercept coefficient = the log of the number of PM particles when the log of the number of cars is equal to 0

β1 = Slope coefficient = when the number of cars increases by ln(x) mph, the number of PM particles increases by β1

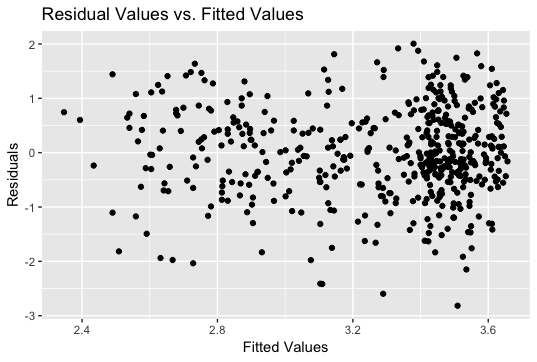
εi = Residuals or distance to the mean

σ = for any number of cars, 99.7% of the PM particles will be within 3 σ of β0 +β1 \* Xi

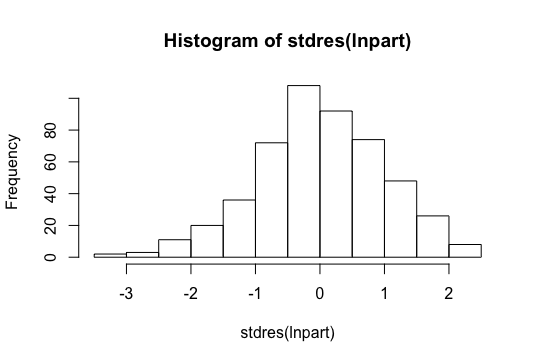
The assumptions we will use for this model are the same as the assumptions we tried to justify earlier on our original data. They are linearity, independence, normality, and equal variance. The assumption of linearity means the data follows a relatively linear pattern. Independence means the data is not influenced by other data points. The assumption of normality means the data has a normal distribution and is not skewed to one side or the other. The new model will have equal variance if the residual vs. fitted data has relatively equal distance from our fitted line to each data point.

**Section 3: Model Verification**

In order to determine if our transformed model is usable on our data, we must justify the assumptions. First we will look at the new residual value vs. fitted value plot to help us determine linearity, independence, and equal variance:



This new plot shows a much more linear looking data, with all the points following along the 0 line, so we can assume our linearity assumption is met. There are no patterns in the data, so we can assume that all the points are independent of eachother, meaning our independence assumption is met as well. Looking at our graph, the variance distance from each point to the 0 line seems to be much more uniform, so our equal variance assumption seems to be met. We again conducted a BP test on the transformed model to determine if there was equal variance. We received a p-value of 0.2008. This is an insignificant p-value, which means that our data is equal in variance, so our equal variance assumption is met.



The histogram of the transformed model shows a much more normal shape, leading us to believe the data is now normal and meets the normality assumption. In order to confirm this, we ran a KS test on the new data to help determine normality. The p-value we received for the KS test was 0.6643. This is also an insignificant p-value, which means the data is normal, which means our normality assumption is met. With our newly transformed model, all of our assumptions are met, so we can use this model on the data.

We performed a hypothesis test on the transformed data to determine if there was a significant relationship between the number of particles and the number of cars in an intersection. The null hypothesis stated that there was no relationship between the number of cars and the amount of PM particles, while the alternative hypothesis stated that there was a relationship between the number of cars and the amount of PM particles. We received a p-value of 2.378e-16, which is essentially 0, meaning we reject the null in favor of the alternative hypothesis. There is a significant relationship between the number of cars and the number of PM particles.

When considering how good of a fit the model is for the data, we can look at the R-squared value, which is the percent of variation in the number of PM particles that is explained away by the number of cars in an intersection. Our R-squared value for our model is 0.1265, which means that only 12.65% of PM particles are explained away by number of cars. This is a low number, so our model isn’t necessarily the best fit for our data. However, the fact that we are 12.65% better off in explaining the number of PM particles when we include information about the number of cars in the intersection is still better than if we had not included the information, so we will continue on in our model for prediction.

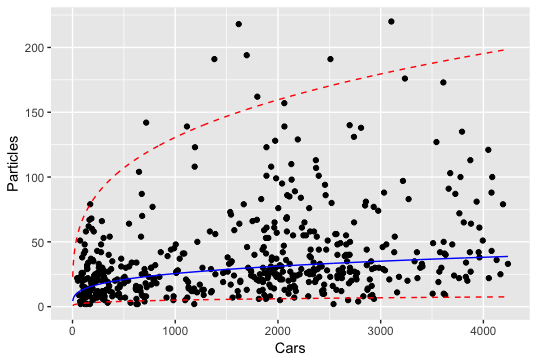
In order to determine if our model has predictive accuracy, we can perform cross validation. This involves removing a percentage of our data, fitting our model to the remaining data, and then using that fitted model to predict the removed test data. We can do this many times to measure how good our model is at predicting.

We performed 100 cross validations in order to measure predictive accuracy. We received a bias of -11.5084, which means that we are under-predicting by 11.5084 on average. Compared with the range of our data (2 to 220), this under-prediction seems to be ok, since we are dealing with such a wide range. We also received a Root Predicted Mean Square Error (RPMSE) of 33.06168, which means that our predictions were off by 33.06168 on average. Comparing this to the PM particles standard deviation of 35.04579, we can see that our RPMSE is about the same, so our predictive accuracy is relatively good. The coverage, which is the percent of our prediction intervals that contain the true value, was calculated to be 0.95. This means that 95% of the true values were contained in our prediction intervals, which shows that our model has good ability to predict PM particles based on cars.

Despite all of this, it is important to understand that our average predictive interval width for our cross validations was 133.5596. This is a very wide prediction interval, which means that although our model is technically good at predicting the value of PM particles, this is because the predicting interval covers a huge portion of the data, so it honestly is hard to tell whether the model is really any good at predicting or not.

**Section 4: Results**

Based on our results of the model, we can conclude that there is a relationship between the number of cars and the number of PM particles. This result is based on our hypothesis test that we conducted earlier in the analysis. We calculated a p-value of essentially 0, which proved to us that there was a relationship between the number of PM particles and the number of cars. We performed a 95% confidence interval on the fitted regression model to help us come up with an estimate of the relationship. The estimate will be in the form of our equation *Yi = β0 +β1 \* Xi,* and this equation will be after we have untransformed the data from our natural log transformation. The fitted model will be Yi = 3.488283 + 1.334089xi with confidence intervals of 95% to help us account for uncertainty. We are 95% confident that if the number of cars in the intersection were 0, the number of PM particles would be between 2.172730 and 5.600383, on average. We are also 95% confident that is the number of cars in the intersection were to increase by 1, then the number of PM particles would increase by between 1.248007 and 1.426109, on average. Below is a graph of the original data with the fitted regression line and the prediction interval (dotted lines):



If we were to use our model to predict the number of PM particles when there are 1800 cars passing through an intersection, we would get a value of 30.265 PM particles. In order to account for uncertainty in our model, we could also construct a confidence interval so that we can estimate the value of PM particles with 95% confidence. We are 95% confident that the number of PM particles when 1800 cars pass through an intersection is between 5.917719 and 154.7843, on average.

**Section 5: Conclusions**

By conducting this analysis, we came to conclude that there is a relationship between the number of PM particles and the number of cars passing through an intersection. We came to this conclusion by fitting a simple linear regression model to log-transformed data that measured the number of PM particles and the associated number of cars driving through an intersection. Although we determined that there was a relationship, it is important to recognize our measure of uncertainty in the relationship. We developed confidence and prediction intervals to help us account for this uncertainty, and the width of our intervals were very large. This means it can be hard to interpret an accurate prediction for the number of PM particles based on the number of cars passing through an intersection.

Some next steps that we would advise you take would be to collect more data in a more isolated area. It could be possible that the number of PM particles is being influenced by another factor, such as a nearby factory. This could account for the large amounts of outliers we have in the data and could skew our model, since outliers affect linear regression greatly. If the data was collected in a more isolated area, we could develop a less skewed model and better understand the relationship between PM particles and the amount of cars. Another aspect of the relationship between cars and PM particles is the type of car and how it affects the number of PM particles. Different cats may affect the amount of PM particles in different ways, so you could do more research into what cars produce the most PM particles to better understand the relationship.

R code:

#Read in data set and check to make sure it was read in correctly

particles <- read.table(file = 'https://mheaton.byu.edu/Courses/Stat330/Exams/Midterm1/Data/PM.txt', header = TRUE)

head(particles)

tail(particles)

names(particles)

#Explore data through graphs and summary statistics

library(ggplot2)

ggplot(data = particles, mapping = aes(x = Cars , y = Particles)) + geom\_point() + ggtitle('Particles Based on Number of Cars') + xlab('Cars') + ylab('Particles')

cov(particles$Cars, particles$Particles)

cor(particles$Cars, particles$Particles)

#fit the data to the model

particleslr <- lm(formula = Particles~Cars, data = particles)

summary(particleslr)

coef(particleslr)

ggplot(particles, aes(x= Cars,y= Particles)) + geom\_point() + geom\_smooth(method="lm",se=FALSE)

part\_pred <- data.frame(Cars=c(2500))

predict.lm(particleslr, newdata=part\_pred)

#check assumptions

names(particleslr)

#linearity and independence

ggplot(particleslr, aes(y= particleslr$residuals,x= particleslr$fitted.values)) + geom\_point() + ggtitle('Residual Values vs. Fitted Values') + xlab('Fitted Values') + ylab('Residuals')

#Normality

library(MASS)

ggplot()+geom\_histogram(mapping=aes(x=stdres(particleslr))) + ggtitle('Std. Residuals Histogram') + xlab('Std. Residuals')

hist(stdres(particleslr))

ks.test(stdres(particleslr), "pnorm")

library(normtest)

jb.norm.test(stdres(particleslr))

#equal variance

library(lmtest)

bptest(particleslr)

#identify outliers

cd <- cooks.distance(particleslr)

cd

plot(cd, type = "h")

which(cd>4/500)

#transform the data

lnpart <- lm(log(Particles)~log(Cars), data = particles)

car.seq <- seq(2, 4239, length=1000)

names(lnpart)

#check assumptions for new model

#linearity and equal variance

ggplot(lnpart, aes(y= lnpart$residuals,x= lnpart$fitted.values)) + geom\_point() + ggtitle('Residual Values vs. Fitted Values') + xlab('Fitted Values') + ylab('Residuals')

bptest(lnpart)

#normality

ggplot()+geom\_histogram(mapping=aes(x=stdres(lnpart))) + ggtitle('Std. Residuals Histogram') + xlab('Std. Residuals')

hist(stdres(lnpart))

ks.test(stdres(lnpart), "pnorm")

#plot original data with new fitted model regression line

preds <- data.frame(Cars=car.seq)

preds$Particles <- predict.lm(lnpart, newdata=preds)#gives prediction on transformed scale

preds$Particles <- exp(preds$Particles)

ggplot(data = particles, mapping = aes(x = Cars , y = Particles)) + geom\_point() + geom\_line(data=preds, aes(x=Cars,y=Particles))

#cross validation

n.cv <- 100

bias <- rep(NA, n.cv)

rpmse <- rep(NA, n.cv)

coverage <- rep(NA, n.cv)

pred.width <- rep(NA, n.cv)

n.test <- 10

for(i in 1:n.cv){

# Choose which obs. to put in test set

test.obs <- sample(1:nrow(particles), n.test)

# Split data into test and training sets

test.set <- particles[test.obs,]

train.set <- particles[-test.obs,]

# Using training data to fit a model

train.lm <- lm(log(Particles)~log(Cars),data=train.set)

# Predict test set

test.preds <- predict.lm(train.lm, newdata=test.set, interval="prediction")

#untransform data

test.preds <- exp(test.preds)

# calculate coverage, bias, rpmse and prediction width

coverage[i] <- mean((test.preds[,2] < test.set$Particles) & (test.preds[,3]>test.set$Particles))

bias[i] <- mean(test.preds[,1]-test.set$Particles)

rpmse[i] <- sqrt(mean((test.preds[,1]- test.set$Particles)^2))

pred.width[i] <- mean(test.preds[,3] - test.preds[,2])

}

mean(bias)

mean(rpmse)

mean(coverage)

mean(pred.width)

sd(particles$Particles)

min(particles$Particles)

max(particles$Particles)

#hypothesis test to determine if there is a realtionship

summary(lnpart)

#confidence and prediction intervals

particle.confidence <- confint(lnpart, level = 0.95)

particle.confidence

particle.confidence <- exp(particle.confidence)

particle.confidence

#predict particles for an intersection with 1800 cars passing through

predictionframe <- data.frame(Cars=c(1800))

partical.prediction <- predict.lm(lnpart, newdata=predictionframe, interval="prediction", level=0.95) #Prediction

partical.prediction <- exp(partical.prediction)

partical.prediction

#graph prediction interval

preds2 <- data.frame(Cars=car.seq)

preds2$Particles <- predict.lm(lnpart, newdata=preds2, interval="prediction", level=0.95) #Prediction

preds2$Particles <- exp(preds2$Particles)

lowerlim <- data.frame(preds2$Particles[,2])

upperlim <- data.frame(preds2$Particles[,3])

linefit <- data.frame(preds2$Particles[,1])

ggplot(data = particles, mapping = aes(x = Cars , y = Particles)) + geom\_point() +

geom\_line(data = upperlim, aes(x=preds2$Cars,y=preds2$Particles[,3]), linetype = "dashed", color = "red")+

geom\_line(data = lowerlim, aes(x=preds2$Cars,y=preds2$Particles[,2]), linetype = "dashed", color = "red")+

geom\_line(data = linefit, aes(x=preds2$Cars,y=preds2$Particles[,1]), color = "blue")