Colin White
Feb 4, 2021
STAT 330 | INTRODUCTION TO REGRESSION

HOMEWORK ANALYSIS #2 - STOPPING DISTANCE

Problem: "One key component of determining appropriate speed limits is the amount of distance that is required to stop at a given speed. For example, in residential neighborhoods, when pedestrians are commonly in the roadways, it is important to be able to stop in a very short distance to ensure pedestrian safety.

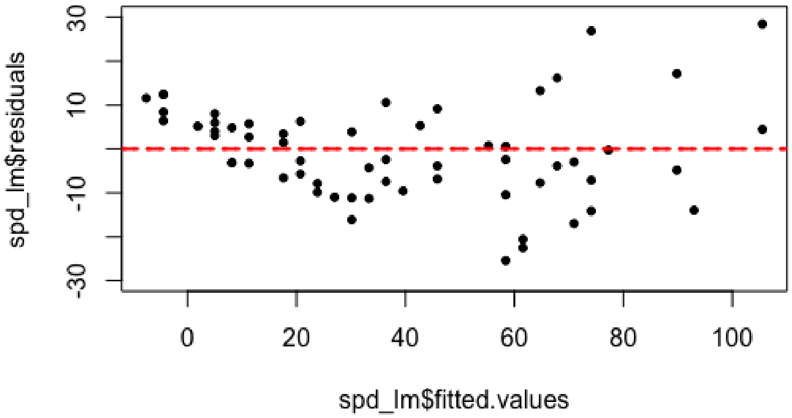
The dataset StoppingDistance.txt compares the distance (in feet) required for a car to stop on a certain rural road against the speed of the car. In each of the following questions, assume that your audience (the people you are writing your answer to) are law enforcement officials who have weak a statistical and mathematical background (be sure to explain things quite simply). Please attach your clearly commented code to the back of your answers as an appendix."

1. In your own words, summarize the overarching problem and any specific questions that need to be answered using the stopping distance data. Discuss how statistical modeling will be able to answer the posed questions.

We need to know what a safe speed limit will be for certain rural road. To figure out what a safe speed limit will be we need to know if speed is an accurate predictor of stopping distance. If the data that compares stopping distance to speed passes all the assumptions for a SLR model, we can make a SLR model that helps us choose a good speed limit based on required stopping distance.

2. Use the data to assess if a simple linear regression model (without doing any transformations) is suitable to analyze the stopping distance data. Justify your answer using any necessary graphics and relevant summary statistics. Provide discussion on why an SLR model on the raw data (not transformed) is or is not appropriate.

By performing a residuals vs fitted scatterplot, a Breusch-Pagan test, a Histogram of standardized residuals, a one-sample Kolmogorov-Smirnov, and a Jarque-Bera test for normality I conclude that the data is suitable for a SLR model without a transform. Please see tests results and interpretations below.



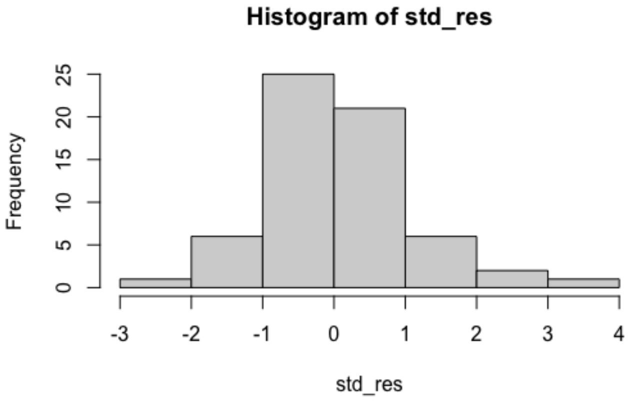
Because there are roughly the same number of points above and below the line on the residual vs fitted the data passes the test for Linearity, Independence and Equal Variance Assumptions.

Breusch-Pagan test (test for Linearity, Independent and Equal Variance Assumptions:)

```
data:  spd_lm
BP = 14.535, df = 1, p-value = 0.0001376
```

Because the p value is low, we reject the null hypothesis and conclude the data are heteroskedastic.

Histogram of standardized residuals (tests normality assumption)



This histogram demonstrates that the errors are normally distributed.

One-Sample Kolmogorov-Smirnov Test: (When assessing model assumptions, what is “close enough” to normal?)
data: std_res
D = 0.082017, p-value = 0.7985
alternative hypothesis: two-sided.
Fail to reject the null hypothesis, and we conclude the data comes from normal distribution of errors.

Jarque-Bera test for normality: for assessing model assumptions, what is “close enough” to normal.
data: std_res
JB = 5.3682, p-value = 0.043
Because the p-value above 0 we fail to reject the null hypothesis, conclude the data comes from normal distribution.

3. Write out (in mathematical form with greek letters) a justifiable (perhaps after a transformation) SLR model that would help answer the questions in problem. Provide an interpretation of each mathematical term (variable or parameter) included in your model. Using the mathematical form, discuss how your model, after fitting it to the data, will be able to answer the questions in this problem.

Sqrt(Y) = $ß_0$ + x * $ß_1$

Y = stopping distance
x = speed
$ß_0$ = The predicted stopping distance if the speed is zero
ß1 = Amount the predicted Sqrt(Y) increases per unit increase x (speed)

This model will be able to predict stopping distance based on speed accurately. This will help the police department set a safe speed limit for the rural road.

4. List, then discuss and justify the assumptions from your model in #3 using appropriate graphics or summary statistics.

By performing a residuals vs fitted scatterplot, a Breusch-Pagan test, a Histogram of standardized residuals, a one-sample Kolmogorov-Smirnov, and a Jarque-Bera test for normality I conclude that the data is suitable for a SLR model without a transform. Please see tests results and interpretations below.

Breusch-Pagan test
data: spd_lm
BP = 3.5597, df = 1, p-value = 0.0592
Because the p value is low, we reject the null hypothesis and conclude the data are

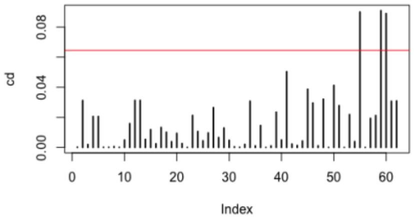
heteroskedastic.

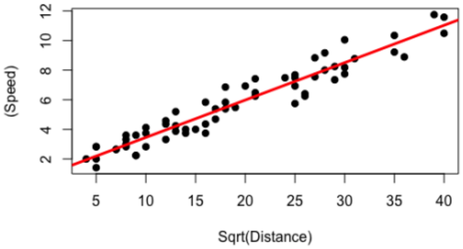
One-sample Kolmogorov-Smirnov test

data: std_res
D = 0.062753, p-value = 0.9676
alternative hypothesis: two-sided
Because the p-value is high, we fail to reject the null hypothesis, conclude that data are heteroskedastic

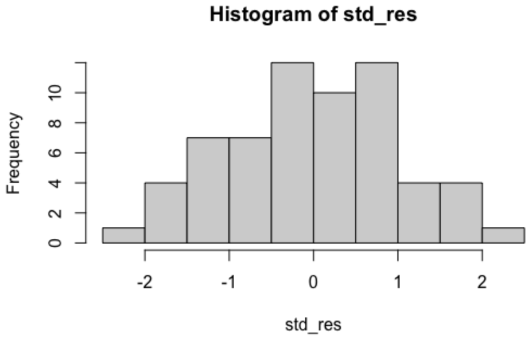
Jarque-Bera test for normality

JB = 1.2723, p-value = 0.418
Because the p-value is above 0 - fail to reject the null hypothesis, conclude the data comes from normal distribution.

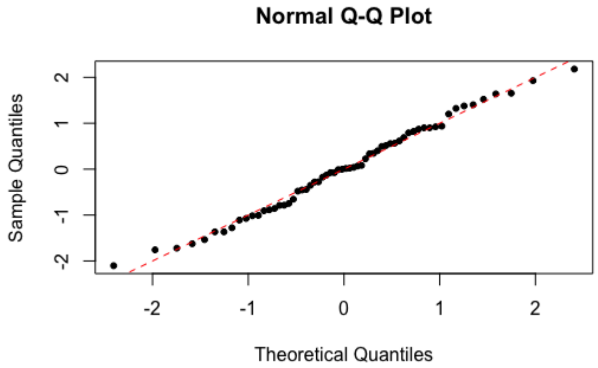


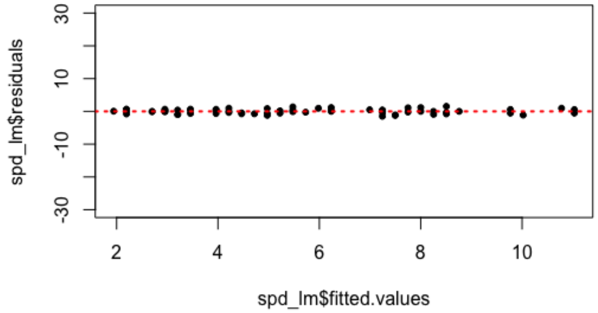


This QQ test assess Normality (and, to some extent, outliers) and the data passes the assumption check.



This histogram demonstrates that the errors are normally distributed.





Because there are roughly the same number of points above and below the line on the residual vs fitted the data passes the test for Linearity, Independence and Equal Variance Assumptions.

After the transform the data passes all the assumption to make a good SLR model.

List of assumptions for reference.
1. Linearity – speed is linearly correlated with the square root of stopping distance
2. Errors are Independent of one another and of covariates (x).

3. Errors are Normally distributed.

4. Equal variance for all ϵi _

5. Assess and interpret the fit and predictive accuracy of your model on the level of your target audience. Make conclusions regarding if your predictive accuracy is "good" relative to the original spread of the response variable.

Goodness of fit:
R squared of untransformed data = 0.8777003
R squared of transformed data = 0.9251019
R squared is bigger for transformed model, therefore its predictive accuracy is better.

Cross validating:
mean(bias) = -0.4173667
mean(rpmse) = 9.398996
range(spd$Distance) = 2, 138
sd(spd$Distance) = 33.37545

The RPMSE spans hardly any of the range and it is much smaller than the standard deviation indicating that the predictive accuracy of the model is "good".

6. Summarize the results of fitting your model in #3 by writing out the fitted model in equation form (do NOT just provide a screen shot of the R or Python output). Interpret each of the estimated parameters in the context of the problem. Provide a plot of the data with a fitted regression line on the original scale of the data.

Sqrt(Y) = 0.9323 + x * 0.2525

Y = stopping distance
x = speed
0.932396 = ß$_0$ = The predicted stopping distance if the speed is zero
0.252466 = ß1 = Amount the Sqrt(Y) increases per unit increase x

7. The local law enforcement is considering implementing a speed limit of 35 MPH. Use your model to obtain a prediction of the distance required by a vehicle to stop when traveling at 35 MPH. How much of a reduction in stopping distance would be achieved by making it a 30 MPH speed limit instead? Given that the road is a rural road with many homes, provide an argument for or against the use of 35 MPH.

Sqrt(Y) = ß$_0$ + x * ß$_1$

Sqrt(Y) = (ß$_0$ + x * ß$_{1)^2}$

Y = (0. 932396 + x * 0. 252466)^2

Y = (0. 932396 + (35) * 0. 252466)^2 = 95.43 ft

Y = (0. 932396 + (30) * 0. 252466)^2 = 72.36 ft

The predicted stopping distance difference between 35 and 30 MPH is 23.07 feet. Based on this prediction and there being many homes on the road I recommend the speed limit be capped at 30 mph so the drivers have enough time to stop if a pedestrian walks on the street unexpectedly.

CODE(R)

```
#Code is derived from examples in class
install.packages("ggplot2")
install.packages(c("ggplot2","lmtest","normtest"),dep = TRUE)
library(MASS)
library(ggplot2)
library(lmtest)
library(normtest)

## Read in the Data
spd = read.csv("speed_dist.txt",sep=" ")

# attach(ad)
head(spd)
summary(spd)

## Exploratory Analyses
## scatterplots
plot(spd$Speed,spd$Distance,pch=19,xlab="Speed",ylab="Dist")
with(spd,plot(Speed,Distance,pch=19,xlab="Speed",ylab="Distance"))
scatter.smooth(spd$Speed,spd$Distance)

## ggplot
ggplot(data = spd) + geom_point(aes(x = P, y = R),col ="red")

ggplot(data = spd) + geom_point(aes(x = P, y = R),col ="red") +
  geom_smooth(aes(P,R)) + theme_classic()

cor(spd$P,spd$R)


## Fit a Linear Model to Data
spd_lm = lm(sqrt(Distance) ~ Speed, data=spd)
summary(spd_lm)
#plot(spd_lm)


## Check Linear & Equal Variance Assumption
plot(spd_lm$fitted.values,spd_lm$residuals,
     pch=20,ylim=c(-30,30))

abline(a=0,b=0,lwd=2,col = "red",lty = 3)

plot(spd$Speed,spd_lm$residuals,
     pch=20,ylim=c(-30,30))
```

```
abline(a=0,b=0,lwd=2,col = "red",lty = 3)
abline(h = 0,lwd=2,col = "red",lty = 2)

bptest(spd_lm)        ## Breusch-Pagan test

## Check Normality Assumption
std_res = MASS::stdres(spd_lm) ## This is accounting for more than just sigma

hist(std_res)


hist(std_res,freq = FALSE)
curve(dnorm,from = -4,to = 4,add = TRUE,
      col = "cornflowerblue",lwd = 2,lty = 4)

hist(std_res,freq = FALSE,breaks = 15)
curve(dnorm,from = -4,to = 4,add = TRUE,
      col = "green")

ggplot() + geom_density(aes(x=std_res))
ggplot() + geom_histogram(aes(x=std_res))

### q-q plot
qqnorm(std_res,pch=20)
abline(a=0,b=1,col = "red",lty = 2)

### not normal (fake data)
set.seed(2)
exp.vars = scale(rt(37,df = 4))
qqnorm(exp.vars,pch=20,main = "Not Normal Q-Q Plot")
abline(a=0,b=1)

### actually normal (fake data)
exp.vars = scale(rnorm(37))
qqnorm(exp.vars,pch=20,main = "Normal Q-Q Plot")
abline(a=0,b=1)


ks.test(std_res,"pnorm") # Kolmogorov-Smirnov test
normtest::jb.norm.test(std_res)  #Jarque-Bera test

## Check to see if there are outliers
cd = cooks.distance(spd_lm)
plot(cd,type="h",lwd = 2)
```

```
abline(h = 4/nrow(spd),col = "red")

outliers = which(cd > 4/nrow(spd))
spd[outliers,]

## Plot sqrt-transformed data
plot(sqrt(spd$Distance),(spd$Speed),pch=19,xlab="Sqrt(Distance)",ylab="(Speed)")
ggplot(spd,aes(x=sqrt(Distance),y=(Speed))) + geom_point()

## Fit a sqrt-transformed SLR Model
trans_lm = lm(sqrt(Distance)~(Speed),data=spd)
summary(trans_lm) ## be careful in interpreting these coefficients since we transformed the
data

summary(trans_lm)$r.squared ## R^2 of original model

std_res = stdres(trans_lm)
ggplot() + geom_density(aes(x=std_res))
ggplot() + geom_histogram(aes(x=std_res))

ks.test(std_res,"pnorm")
normtest::jb.norm.test(std_res)
bptest(trans_lm)

## Plot Fitted Regression line on transformed scale
plot((spd$Speed),sqrt(spd$Distance),pch=19,ylab="Sqrt(Distance)",xlab="(Speed)")
abline(a=trans_lm$coef[1],b=trans_lm$coef[2],lwd=3,col="red")

## Plot the transformed regression model on original scale of the data
plot(spd$Speed,spd$Distance,pch=19,xlab="Speed",ylab="Distance")
#pred_dist = seq(min(spd$Speed),max(spd$Speed),by=1) ## Sequence (seq) of Pages of
Advertising that I'm interested in predicting revenue
preds_trans = trans_lm$coef[1]+trans_lm$coef[2]*spd$Speed ## Prediction of log(Rev)
preds_orig = (preds_trans)^2 ## Predictions of Revenue
lines(spd$Speed,preds_orig,lwd=3,col="blue") ## Draw "line" on original scale of data

preds = data.frame(Speed=spd$Speed,Distance=preds_orig)
ggplot(data=spd,aes(x=Speed,y=Distance))+geom_point()+
  geom_line(data=preds,aes(x=Speed,y=Distance),color="red",inherit.aes=FALSE)

## Check assumptions of transformed model
hist(stdres(trans_lm),freq = FALSE) #Normality OK
curve(dnorm,from = -4,to = 4,add = TRUE)
```

```
plot(trans_lm$fitted.values,trans_lm$residuals,pch=19) ## Equal Variacne OK
abline(a=0,b=0)

## Assess the fit (via R^2)
summary(trans_lm)$r.squared ## R^2 is bigger than untransformed model
og_lm <- lm(Distance ~ Speed, data = spd)
summary(og_lm)$r.squared

## Assess predictive ability of the model (via cross validation)
set.seed(1)
n_test = 4
n_cv = 1000
bias = numeric(n_cv)
rpmse = numeric(n_cv)

for(i in 1:n_cv){

  test_obs = sample(1:nrow(spd),n_test)
  test_spd = spd[test_obs,]
  train_spd = spd[-test_obs,]
  train_lm = lm(sqrt(Distance)~(Speed),data=train_spd)
  test_preds = (predict.lm(train_lm,newdata=test_spd))^2
  bias[i] = mean(test_preds-test_spd$Distance)
  rpmse[i] = sqrt(mean((test_preds-test_spd$Distance)^2))

}

mean(bias)
mean(rpmse)
range(spd$Distance)
sd(spd$Distance)

## Original scale
set.seed(1)
n_test = 4
n_cv = 1000
bias = rep(0,n_cv)
rpmse = rep(0,n_cv)

for(cv in 1:n_cv){
  test_obs = sample(1:nrow(spd), n_test)
  test_spd = spd[test_obs,]
  train_spd = spd[-test_obs,]
  train_lm = lm(R ~ P, data=train_spd)
  test_preds = predict.lm(train_lm,newdata=test_spd)
  bias[cv] = mean(test_preds-test_spd$R)
  rpmse[cv] = sqrt(mean((test_preds-test_spd$R)^2))
}


mean(bias_log)
mean(rpmse_log)

mean(bias)
mean(rpmse)
range(spd$R)
sd(spd$R)
```