Lab 2- Solution Stat 230 Transformations and Outliers

P.B. Matheson adapted from A.S. Wagaman

Transformations / Re-expressions

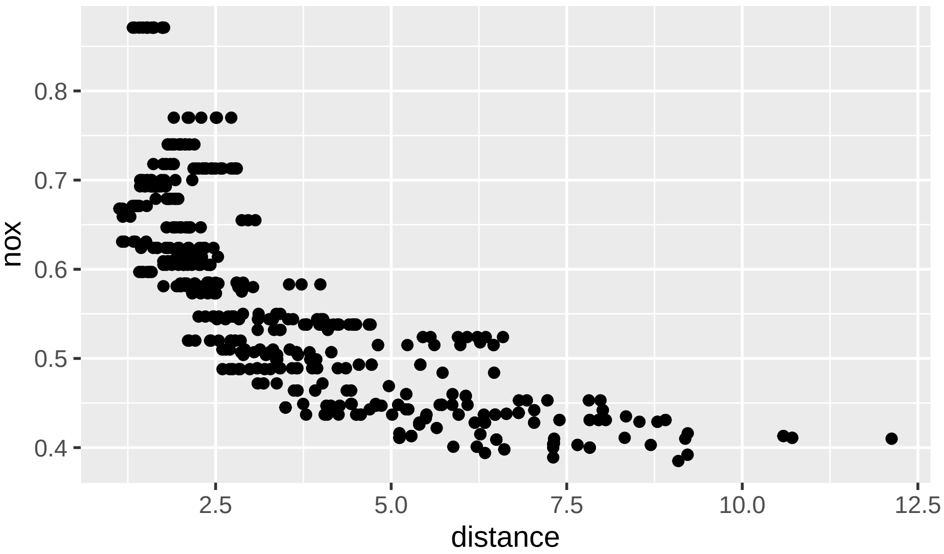
For this exercise, we first load the **Boston** dataset.

```
boston <- read.table("https://pmatheson.people.amherst.edu/stat230/boston.txt", header = TRUE)
```

This is a data set on 14 variables on housing values in the suburbs of Boston. Variables include crime, zone (deals with residential proportion of housing), indust (deals with industry proportion), charles (on river or not), nox (amount of nitric oxides), rooms (avg. number per dwelling), age (proportion built before 1940), distance (weighted to 5 employment centers), radial (index of access to highways), tax (property tax per $10,000), ptratio (pupil-teacher ratio), minor (index of proportion of minorities), lstat (% lower status of population), and medv (median value of homes). Zone is always between 0 and 100. Charles is a binary variable. Radial is an index from 1- 24 (integers). The oddest variable is perhaps minor. Values from 196 and below indicate a large proportion of minorities. Values around 396 indicate a completely non-minority area, and values below 396 down to 196 indicate areas that have decreasing proportions of minorities.

Suppose we want to try to predict *nox* as a function of *distance*.

```
gf_point(nox ~ distance, data = boston)
```



Based on the plot, does it appear that this is reasonable for a linear regression?

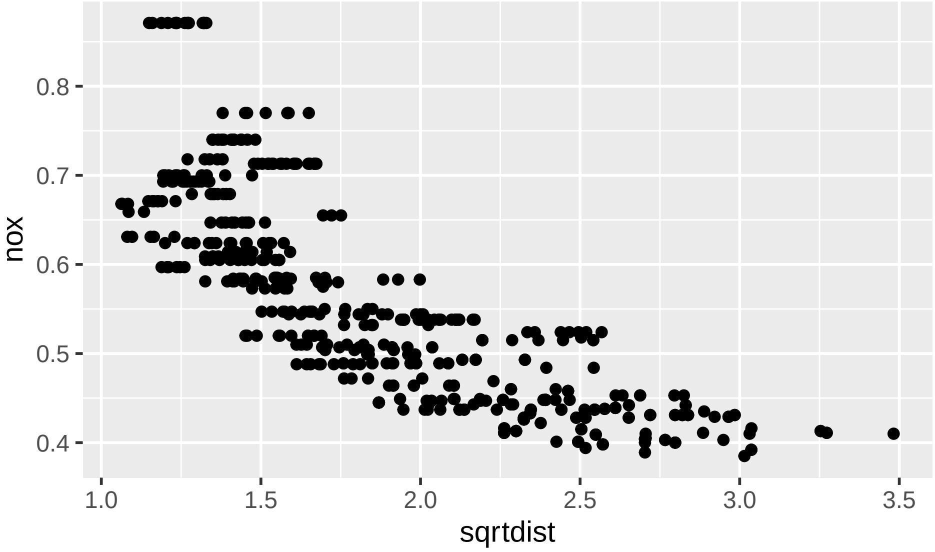
> This relationship is curved. A transformation is likely to be useful.

There are issues here so we should explore some re-expressions. Here, I will demo how to add a new variable to the data set to try out. I'll add sqrtdist as the square root of distance to the data set. (Note that log and sqrt can be interpreted inline, meaning you don't have to add these to the data set when exploring via graphs, but you will want to before running the lm command to fit the model, so you may as well add them anyway.)

```
boston <- mutate(boston, sqrtdist = sqrt(distance))
```

Now I can see what the relationship between nox and my new distance variable is.

```
gf_point(nox ~ sqrtdist, data = boston)
```



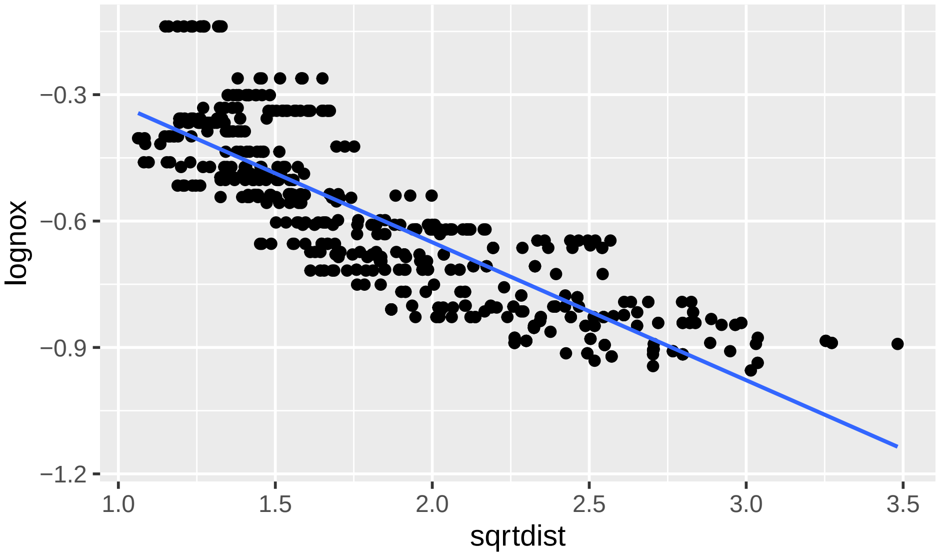
Based on the plot, does it appear that this is reasonable for a linear regression? Is it improved over the original?

ANSWER

It looks to be an improvement, but I'd prefer the relationship to be more linear, so this still needs work.

We can keep exploring - maybe we should try working with *nox*. After exploring a bit (by adding variables, making scatterplots, etc.), we decided on the following re-expressions - square root of distance and log of nox. It's not perfect, but is more linear than the original. Automated ways of searching for re-expressions do exist - you can explore some R packages to learn more, but for our class, you'd just want to try a few powers (square, cube, square root, log) up or down, so you don't go too crazy exploring options.

```
boston <- mutate(boston, lognox = log(nox))
gf_point(lognox ~ sqrtdist, data = boston) %>% gf_lm()
```



Fit a regression line with these 2 re-expressed variables and report your fitted regression line in terms of the new variables you used.

```
#you need lm and another command - should be two lines
bostonmod <- lm(lognox ~ sqrtdist, data = boston)
msummary(bostonmod)
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.004261   0.018335   0.232    0.816
## sqrtdist    -0.327321   0.009403 -34.809   <2e-16 ***
##
## Residual standard error: 0.1086 on 498 degrees of freedom
## Multiple R-squared:  0.7087, Adjusted R-squared:  0.7081
## F-statistic:  1212 on 1 and 498 DF,  p-value: < 2.2e-16
```

The fitted line here is: predicted log of nox (natural log!) = 0.0043 - 0.3273 sqrtdist.

What about outliers? Good question!

For most of our analyses, we will leave outliers in the model, and if really concerned, run the model a second time without the outliers to see their impact. This means we need ways to easily find them (plots are often used) and temporarily remove them from the dataset (filter).

Do you see any points that might be considered outliers based on chosen fit above? You probably do. First, you might just want to identify the points in question - are there points particularly far away from the overall pattern?

ANSWER

There are a few points that could be considered outliers in the X direction , with sqrtdistance > 3.2. There are also some potential outliers in the Y direction (and also off the overall pattern), with lognox > -0.2.

Assuming you said yes and had similar criteria, to get just the observation numbers of these points, you can do commands like this:

```
with(boston, which(lognox > -0.2)) #IDs points with lognox > -0.2
```

```
##  [1] 142 143 144 145 146 147 148 149 150 151 152 153 154 155 158
```

```
with(boston, which(sqrtdist > 3.2)) #IDs points with sqrtdist > 3.2
```

```
## [1] 348 349 350 351 352
```

```r
with(boston, which(lognox > -0.2 & sqrtdist < 1.2)) #IDs points with lognox > -0.2 and sqrtdist < 1.2
```

```
## [1] 142 143 144
```

Recall that we also could just look for residuals larger than 2 in absolute value when standardized. abs() does absolute value. There are several ways to get the residuals. One way is to use the *rstandard* and *rstudent* functions. So you could do something like:

```r
#model refit because not sure what you may have called yours above
fm <- lm(lognox ~ sqrtdist, data = boston)
msummary(fm)
```

```
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.004261   0.018335   0.232    0.816
## sqrtdist     -0.327321   0.009403 -34.809   <2e-16 ***
##
## Residual standard error: 0.1086 on 498 degrees of freedom
## Multiple R-squared:  0.7087, Adjusted R-squared:  0.7081
## F-statistic:  1212 on 1 and 498 DF,  p-value: < 2.2e-16
```

```r
standres <- rstandard(fm) #computes standardized residuals
studres <- rstudent(fm) #computes studentized residuals
which(abs(standres) > 2) #which standardized residuals are > 2 in absolute value
```

```
## 142 143 144 145 146 147 148 149 150 151 152 153 154 155 158 350 354 355 356 357
## 142 143 144 145 146 147 148 149 150 151 152 153 154 155 158 350 354 355 356 357
## 358
## 358
```

```r
which(abs(studres) > 2) #which studentized residuals are > 2 in absolute value
```

```
## 142 143 144 145 146 147 148 149 150 151 152 153 154 155 158 350 354 355 356 357
## 142 143 144 145 146 147 148 149 150 151 152 153 154 155 158 350 354 355 356 357
## 358
## 358
```

assuming you had a fitted model called fm. You can SAVE the vector of unusual points, to then aid in their temporary removal from the dataset. The example below saves ones with high values of lognox - you could adjust it to just save ones with standardized or studentized residuals greater than 2 in absolute value.

```r
highnox <- with(boston, which(lognox > -0.2)) #IDs points with lognox > -0.2
boston2 <- boston[-highnox, ]
```

This allows you to filter data points with fairly simple R code. But, what if you wanted to string together several of these? The indexing idea gets a little old. Luckily, the dplyr package (which is loaded with mosaic) has a simple *verb* to help us with this. Do you remember which one? This is the preferred way to remove outliers temporarily - make a new data set with FILTER.

```r
boston3 <- filter(boston, (lognox) < (-0.2)) #keep values with lognox < (-0.2), same as above two lines
boston4 <- filter(boston, (lognox) < (-0.2), sqrtdist < 1.8, charles < 1)
```

What observations does the boston4 data set include? How many observations meet that criteria?

> It is all observations from the boston data set with (log(nox) < -0.2, with the sqrt of distance < 1.8, and a charles value of 0 (because that can only be 0 or 1, and < 1 is specified)).

Another way that you can access the residuals (and some other values of interest) is to look at the augmented data set. This is obtainable via the *broom* package command *augment*.

```
augboston <- augment(fm)
names(augboston)
```

```
## [1] "lognox"      "sqrtdist"    ".fitted"     ".resid"      ".hat"
## [6] ".sigma"      ".cooksd"     ".std.resid"
```

What variables do you recognize in the augboston dataset?

ANSWER

The response and predictor are included as the first two variables. We also get predicted y values as .fitted, and residuals as .resid. There are some other values, including standardized residuals.

We will learn more about these variables as the semester progresses. You can filter this dataset using the same ideas as above, with the variables present. Note that the augmented dataset does not contain the studentized residuals. You can use *rstudent* and *mutate* to add those values to the augmented dataset though.

```
augboston <-mutate(augboston, studres=rstudent(fm)) #computes studentized residuals
```

Remove any points you consider outliers from your boston dataset, using the filter command, and refit your model. Compare it to the original model.

COMPARISON:

boston2 is my data set with the outliers removed. Basically, just removed the few points at the top of the scatterplot that are off the overall pattern pretty severely.

```
fm2 <- lm(lognox ~ sqrtdist, data = boston2)
msummary(fm2)
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.042574   0.017225  -2.472   0.0138 *
## sqrtdist    -0.307019   0.008755 -35.068   <2e-16 ***
##
## Residual standard error: 0.09875 on 483 degrees of freedom
## Multiple R-squared:  0.718,  Adjusted R-squared:  0.7174
## F-statistic:  1230 on 1 and 483 DF,  p-value: < 2.2e-16
```

```
msummary(fm)
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.004261   0.018335   0.232    0.816
## sqrtdist    -0.327321   0.009403 -34.809   <2e-16 ***
##
## Residual standard error: 0.1086 on 498 degrees of freedom
## Multiple R-squared:  0.7087, Adjusted R-squared:  0.7081
## F-statistic:  1212 on 1 and 498 DF,  p-value: < 2.2e-16
```

The fitted lines are very similar whether the outlying points are included or not in the model. The original model was that predicted lognox = 0.0043 - 0.3273(sqrtdist), and the new model is that predicted lognox = -0.0426 - 0.3070(sqrtdist). The R-squared for the original model was 0.7087, and the R-squared for the outliers-removed model is 0.718, which is a slight improvement. Not seeing major changes here indicates these points are not too unusual, and we may choose to keep them in the model (easier at times than explaining why they were removed).

Summary

When looking at potential re-expressions, create new variables and save them to your data set using *mutate*. Make scatterplots with the new variables and check for improvements.

For outliers, identify them on scatterplots, and then figure out ways to *filter* them out of the data set if you want to run the analysis without them (and also run it with them to assess their impact).