Published in The Startup

Kay Chansiri Follow
Dec 12, 2020 · 11 min read

Data Screening

The first important step before analyzing your data is getting to know the data. This tutorial uses R and focuses on basic data screening principles, such as checking outliers and testing general linear model assumptions before beginning data analysis.

Credit: https://www.toppr.com/guides/economics/presentation-of-data/textual-and-tabular-presentation-of-data/

Case study: The data for this tutorial is fictitious and about psychosocial traits and abusive relationships. Let's say I am interested in examining the roles of adverse

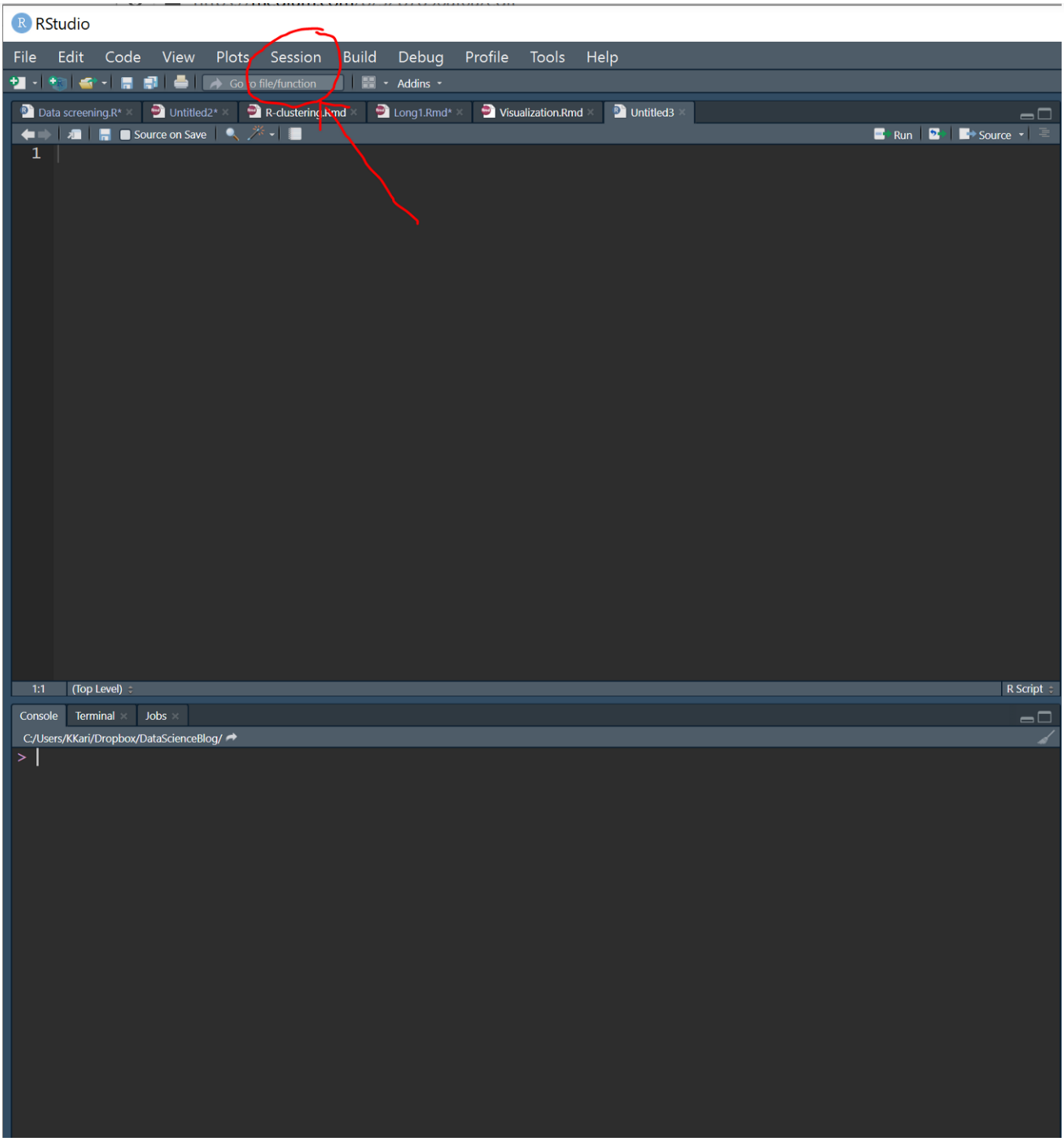
income, because those factors may impact one's intimate relationships. According to the scenario, the independent variables are two continuous variables (childhood trauma exposure and BPT), and the outcomes are the tendency to be an abuser and the tendency to be abused. Race and gender are categorical variables, whereas other confounding variables are continuous. Now that you know my fictitious data let's begin the data screening process.

Step 1: Set the working directory in R. To do so, use the code:

```
> setwd("C:/indicate where your dataset is stored")
```

If you use R Studio, you can set the working directory by go to Session →Set Working Directory → Choose Directory and then identify the folder where you store your data set.





Step 2: Import your data set. The *foreign* package needs to be installed to enable R to read your data.

```
> install.packages("foreign)

> library(foreign)
```

Import your data set:

```
> data <- read.spss("the name of your data set.sav",
use.value.label=TRUE, to.data.frame=TRUE)
```

Don't forget to put *.sav* at the end of the data file name to prevent an error. You now should see your imported data in Global Environment. I store my data as the object' data.'

Step 3: Check basic descriptive statistics of the data.

```
> summary(data)
```

Here is what I get from using the code *summary*:

```
     AltID          Gender           Race           Income          Edu             Age       
       :   1   Min.   :0.0000   Min.   :0.0000   Min.   : 1.000   Min.   :1.000   Min.   :18.00  
A1001  :   1   1st Qu.:0.0000   1st Qu.:1.0000   1st Qu.: 4.000   1st Qu.:3.000   1st Qu.:31.00  
A1002  :   1   Median :1.0000   Median :1.0000   Median : 6.000   Median :4.000   Median :37.00  
A1003  :   1   Mean   :0.5086   Mean   :0.7911   Mean   : 6.533   Mean   :3.623   Mean   :40.57  
A1004  :   1   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:10.000   3rd Qu.:4.000   3rd Qu.:49.00  
A1005  :   1   Max.   :1.0000   Max.   :1.0000   Max.   :12.000   Max.   :5.000   Max.   :91.00  
(Other) :1278   NA's   :12       NA's   :1        NA's   :1        NA's   :1                      
 Intelligence       Abused          Abuser           ACE              BPT        
Min.   :0.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :  0.00  
1st Qu.:2.000   1st Qu.:5.000   1st Qu.:4.800   1st Qu.:2.667   1st Qu.: 17.67  
Median :3.000   Median :5.533   Median :5.400   Median :3.500   Median : 35.00  
Mean   :2.907   Mean   :5.335   Mean   :5.217   Mean   :3.446   Mean   : 36.19  
3rd Qu.:4.000   3rd Qu.:5.867   3rd Qu.:5.800   3rd Qu.:4.000   3rd Qu.: 50.33  
Max.   :4.000   Max.   :6.000   Max.   :6.000   Max.   :6.000   Max.   :100.00  
                NA's   :171     NA's   :171     NA's   :171     NA's   :171     
>
```

According to the summary of my data set, the variables containing missing values show *NA* and the number of missing cases. You should also check whether each variable's minimum and maximum values in your data are accurate, looking at *min* and *max*.

Step 4: Check outliers using the Mahalanobis distance. If you don't know what a Mahalanobis distance is, a simple explanation is a distance between a single data point plotted on a plane of given variables and the means of those variables. Mahalanobis also accounts for the covariance of the given variables, as shown in the formula:

$$D^2=(\mathbf{x}-\mathbf{m})^{\mathrm{T}}\mathbf{C}^{-1}(\mathbf{x}-\mathbf{m})$$

D squared = Mahalanobis distance

x = Vector of data points plotted on a plane of given variables

m =Vector of mean values of the given variables

C-1 = Inverse Covariance matrix of the given variables

and T = Transposed

For example, according to my data set, let's say I would like to explore outliers of the relationship between ACE and one's tendency to be abused using Mahalanobis distance. If Mr. A, a participant in my study, has the childhood trauma score at 4 and the tendency score to be abused at 5, Mr. A's data point on the ACE-abused plane is (4,5). The mean values of ACE and abuse can be obtained using the following codes.

```
> mean(data$ACE, na.rm = TRUE)
#3.445792
> mean(data$Abused, na.rm = TRUE)
#5.335234
```

The means of ACE and Abused are 3.45 and 5.33, respectively. Thus the vector of the mean values is (3.45, 4.33). According to the above mean code, make sure that *na.rm = TRUE*; if *FALSE*, NA will be returned. Now that we have the vector of Mr. A's data

```
> cov(data$ACE, data$Abused, use = “pairwise.complete.obs”, method =
c(“pearson”))

#[1] 18.47806

> var(data$ACE, data$ACE, na.rm = TRUE)

#[1] 0.9536436

> var(data$Abused, data$Abused, na.rm = TRUE)

#[1] 1.3481234
```

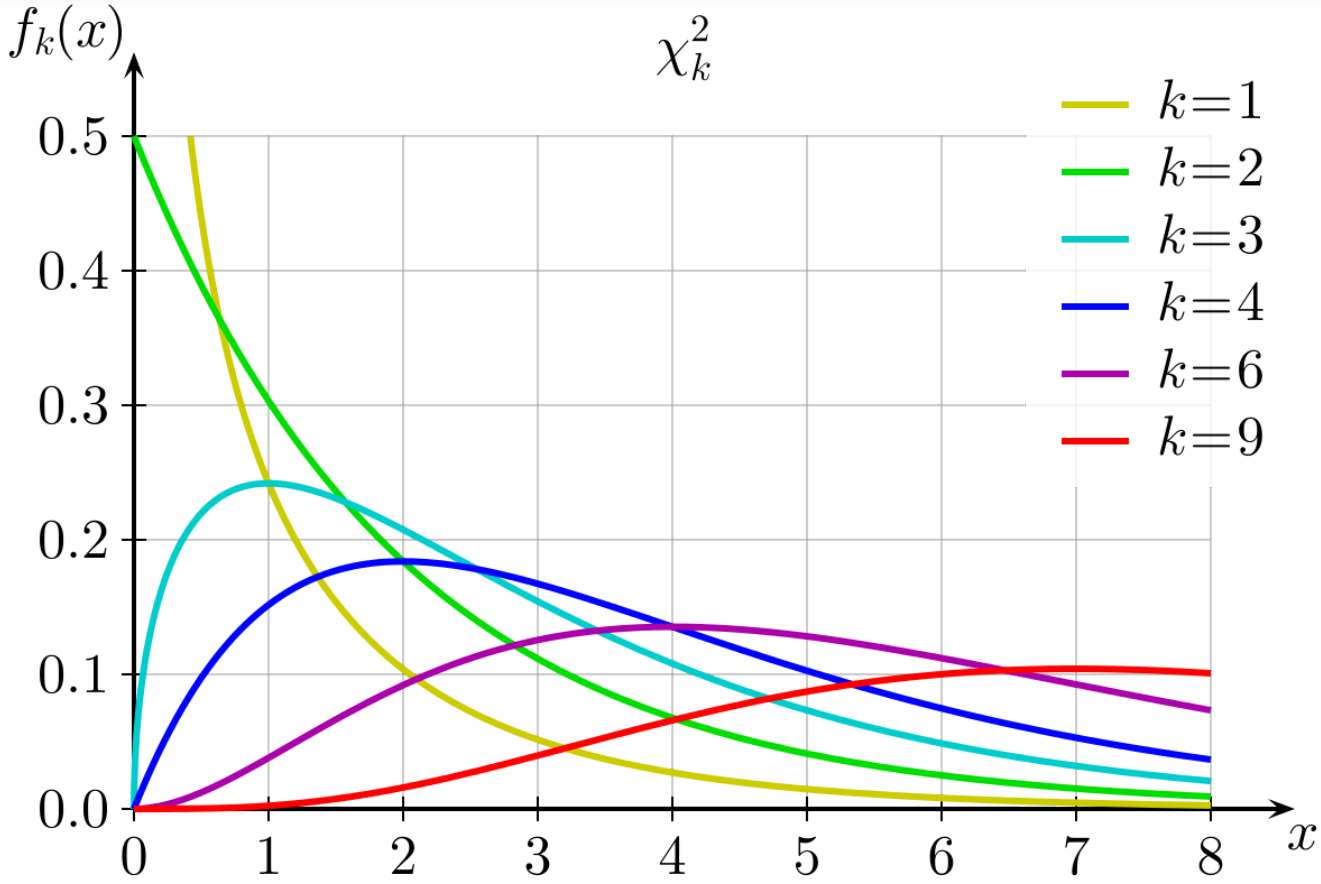
Now that you got the variance/covariance matrix, the mean vector, and the data point vector, plug-in those parameters in the Mahalanobis formula, and you will get the distance of Mr. A from other participants in the dataset.

The example of the Mahalanobis distance calculation that you just learned is about only two variables. In my data set, I have more than two variables of interest. Thus, the vectors of the means and data points will be multidimensional. Don’t worry, though; you don’t need to manually calculate Mahalanobis distances of a multidimensional dataset. R will do the work for you within just a few lines of the below code!

```
> mahal = mahalanobis(data[ , -c(1:3)], colMeans(data[ ,-c(1:3)],
na.rm = TRUE), cov(data[ , -c(1:3)], use = “pairwise.complete.obs”))
```

The code above calculated Mahalanobis distances of every variable in my data set except categorical variables (column 1–3) and then stored the output as the object *mahal*.

Mahalanobis distance needs to be interpreted using Chi-sqaure statistics. Thus, the next step is to convert the whole data set to a Chi-square p-value distribution, which is always right skewed if you have a small sample size. When a sample size increases, the chi-square distribution will be closed to a normal distribution.



Credit: Wikimedia Commons, link to the license: https://commons.wikimedia.org/wiki/File:Chi-square_pdf.svg

As you see from the distribution figure, the higher degree of freedom (i.e., the number of sample size), the more the curve is normally distributed. The area under a Chi-square distribution graph is always 1, and a data point can fall somewhere under the graph from 0 to 1.

Chi-square statistics are used for two purposes: 1) to compare whether a sample's distribution matches their population's distribution (AKA' goodness of fit test') and 2) to compare two sets of data from a contingency table (e.g., to compare whether males tend to hold an engineering degree more than females). The first purpose is what I would consider checking outliers using Mahalanobis. I will see whether the Mahalanobis distance of any participants in my dataset is significantly different from the rest of the data set.

To do so, I will convert my dataset into a Chi-square distribution and set the cutoff point at .999. Remember, the area under a chi-square distribution curve is always 1. Any data

```
> cutoff = qchisq(.999, ncol(data[ , -c(1:3)]))

> summary(mahal < cutoff)
```

Here is my results from *summary*.

```
Mode   FALSE    TRUE    NA's
logical      17    1095     172
```

You see that 1,095 cases fall somewhere less than .999 of my Chi-square distribution curve. Only 17 cases exceed the cutoff, which perhaps could be considered unduly influential outliers. Whether to exclude those outliers, you may test your hypotheses/research questions with and without those outliers to see whether the results change. Then, make a decision based on the theory that you apply and prior research.

On a side note, if you still don't get Mahalanobis' concept, I highly recommend reviewing linear algebra to get a sense of matrices' and vectors' operations because such knowledge is the key to data science. A good quick review of linear algebra can be found in the series "The Essence of Linear Algebra" by 3Blue1Brown on YouTube. This website also provides a good explanation of Mahalanobis distance and a chi-square p-value distribution.

Step 4: Check correlations and multicollinearity. You do not want to use variables that share an extremely overlapping variance as you want each variable to have its predictive value. Use the following code to check correlations.

```
> correlation = cor(data[ , -c(1:3)], use = "pairwise.complete.obs")

> symnum(correlation)
```

A cool thing about *symnum* is that the command uses symbols and letters to indicate



your field to support your decision whether you should drop one of those variables. Here is my *symnum* output.

```
             Inc E Ag Int Absd Absr AC B
Income       1
Edu          .   1
Age                1
Intelligence           1
Abused                     1
Abuser                     +    1
ACE                                 1
BPT                                 +  1
attr(,"legend")
[1] 0 ‘ ’ 0.3 ‘.’ 0.6 ‘,’ 0.8 ‘+’ 0.9 ‘*’ 0.95 ‘B’ 1
```

My fictitious data told me that the correlation between the tendency to be an abuser and being abused is quite high (.9). The correlation between ACE and BPD is quite high as well (.9). Those correlations are theoretically supported by childhood trauma literature, and the variables are key variables in my data. Therefore, I did not drop any of them.

Step 5: Assumption check. This is the most important step that you should carefully consider because statistical models do not tolerate assumption violation.

For General Linear Model (GLM), you need to meet major assumptions: LINE — Linerarity, Independence of error, Normality, and Error homogeneity. For a quick review, GLM represents ‘data +error’ modeling (Rutherford, 2001), and GLM’s dependent variables are continuous. Basic GLM models are ANOVA, regression, and ANCOVA. As my independent and outcome variables are continuous, let’s do the assumption check for a regression model.

To do so, I need to build a model of interest first. If you remember, in the beginning, I would like to explore the roles of ACE and BPT in predicting one’s tendency to experience an intimate abusive relationship, controlling for demographic variables. I will select the tendency to be abused as my outcome and ignore the tendency to be an abuser for now. Thus, my model is,

One thing that people often misunderstand is that assumption checking is all about dependent variables. Assumption checking is centered around residuals. Before we move forward, I highly recommend you understand what residuals are. This site provides a good and simple explanation.

Now let's continue with our assumption check. Once I got my model, I use *fitted.values* to generate the vector of my dependent variable(Abused) and then use *scale* to center the vector's data points such that zero represents the mean of Abused. Data centering enables an easier interpretation, including the interpretation of assumption checks.

I also use *rstudent* to standardize residuals of my dependent variable. Studentized residuals are the division of a residual by its standard deviation. A greater studentized residual value indicates that the data point has the potential to be an outlier.

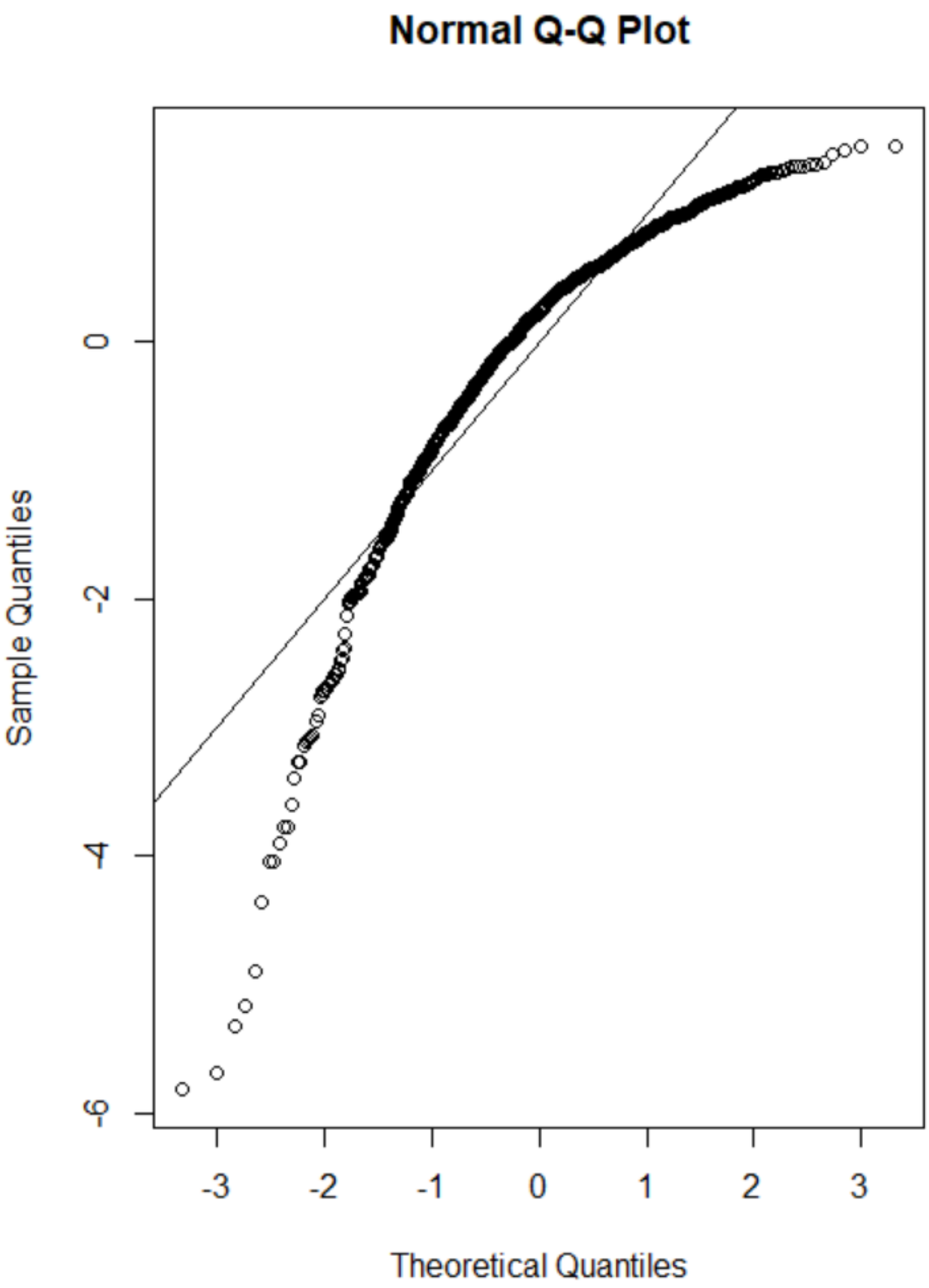
```
> fitted = scale(model$fitted.values)
> standardized = rstudent(model)
```

Now that we fitted our model and standardized our residuals let's begin our first assumption check.

1. Linearity. This assumption check will use a Q-Q plot to see the relationship between our observed and predicted values.

```
> qqnorm(standardized)
> abline(0,1)
```

Here is what my output looks like:


Normal Q-Q Plot
Sample Quantiles
0
-2
-4
-6
-3
-2
-1
0
1
2
3
Theoretical Quantiles

independent variables' planes. If our observed values are linear, the plane of observed values should be in line with the plane of predicted values, resulting in a linear trend in our Q-Q plot.

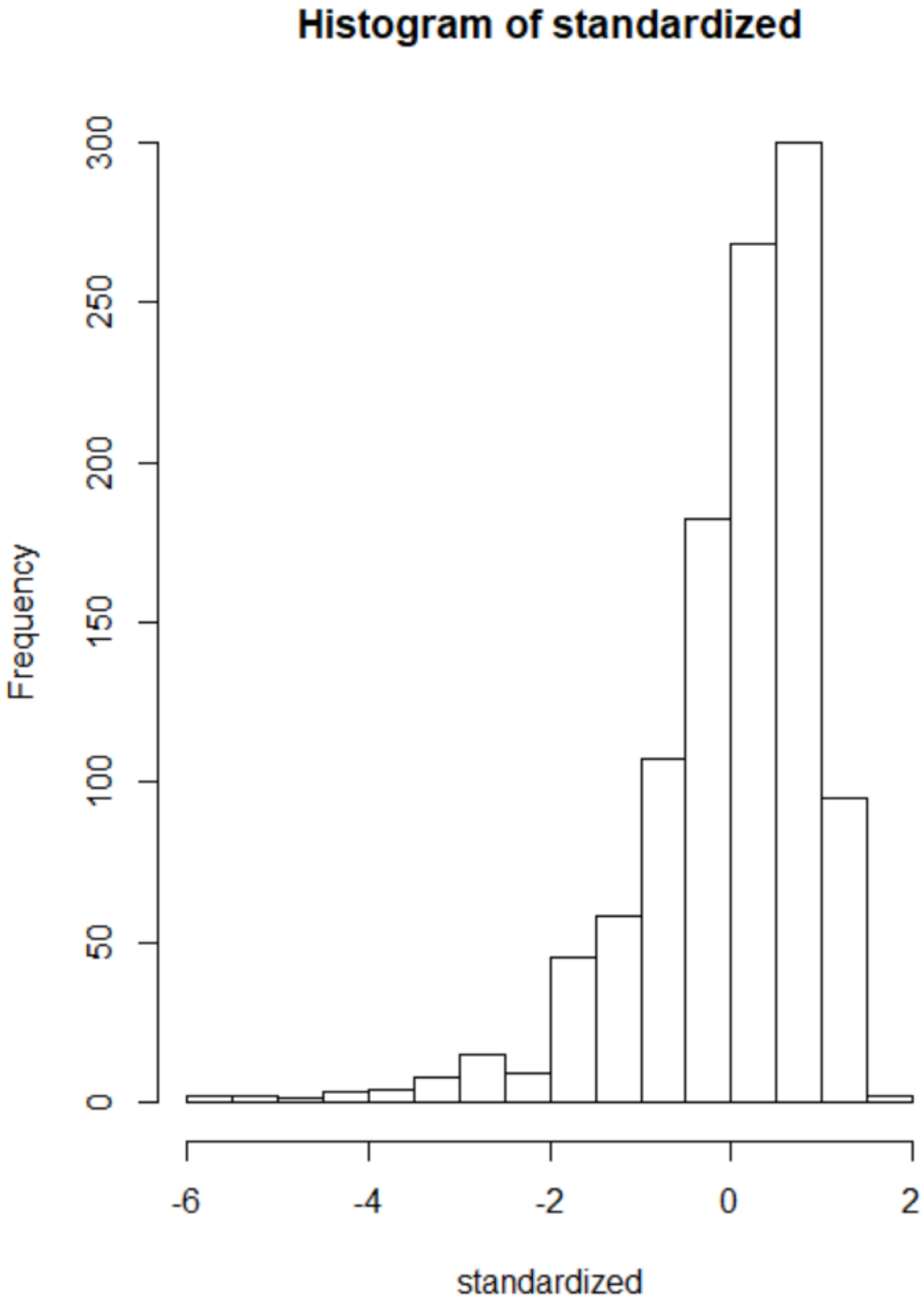
According to my Q-Q plot, the line does not look that linear, meaning that some other omitted predictors could influence my dependent variable's data points. Thus the plane of my observed values is not in alignment with the plane of my predicted values. Perhaps my data is not linearly distributed but curvilinearly. However, as the data is fictitious, I will ignore the distribution pattern and keep going with the next assumption check.

2. Normality. One good tool to check normality is a histogram.

```
> hist(standardized)
```

This is my histogram.




Histogram of standardized
Frequency
0
50
100
150
200
250
300
-6
-4
-2
0
2
standardized

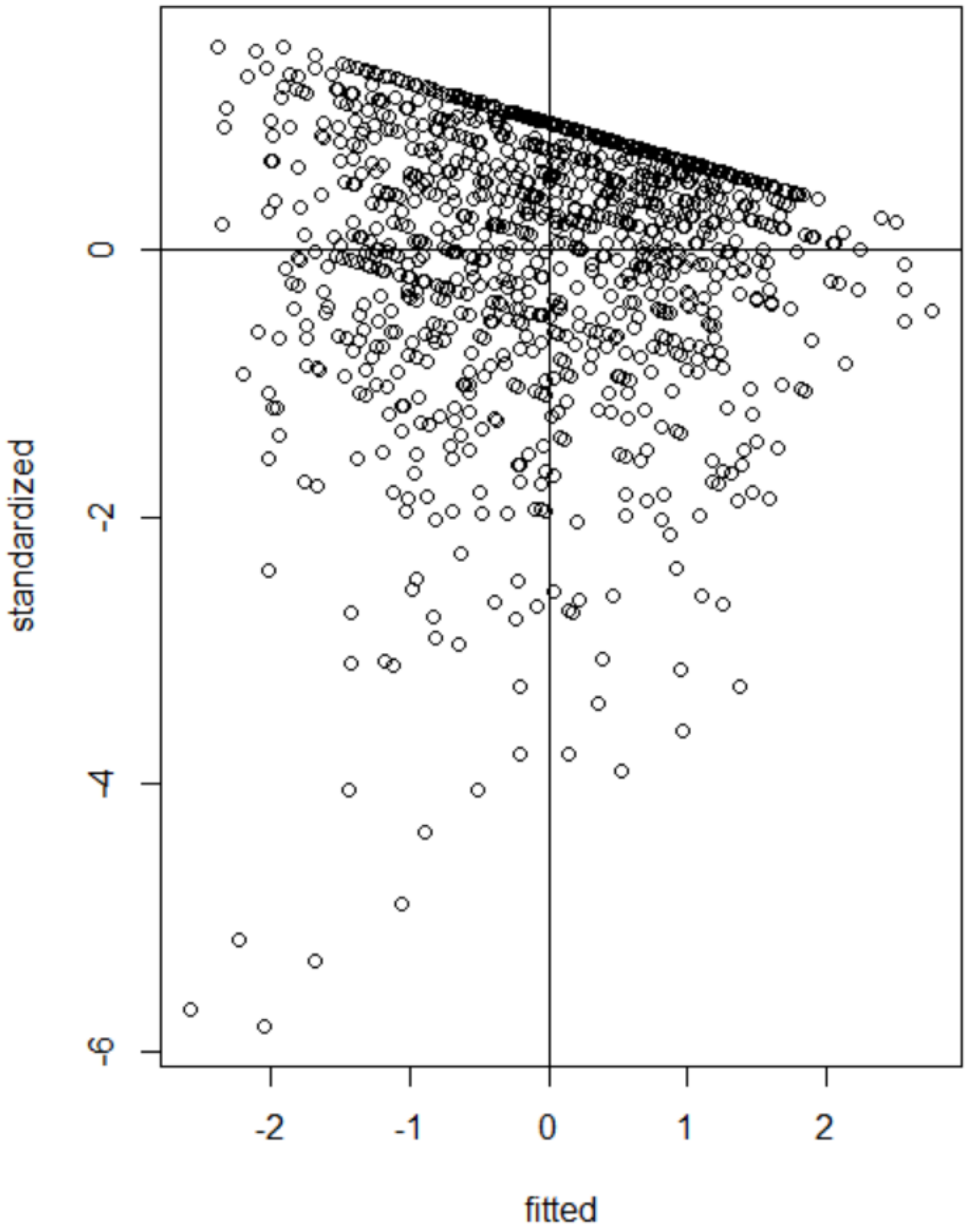
lower than the mean of the abused scale, considering the predictive role of ACE, BPT, and demographic variables. It could be that those participants are outliers, or perhaps some omitted protective factors have interfered here.

3. Error homogeneity (AKA Homogeneity of variance, homoscedasticity). This assumption requires that residuals are equally distributed throughout the dependent variable's data points (AKA fitted values). To check this assumption, we will use a scatter plot.

```
> plot(fitted, standardized)
> abline(0,0)
> abline(v = 0)
```

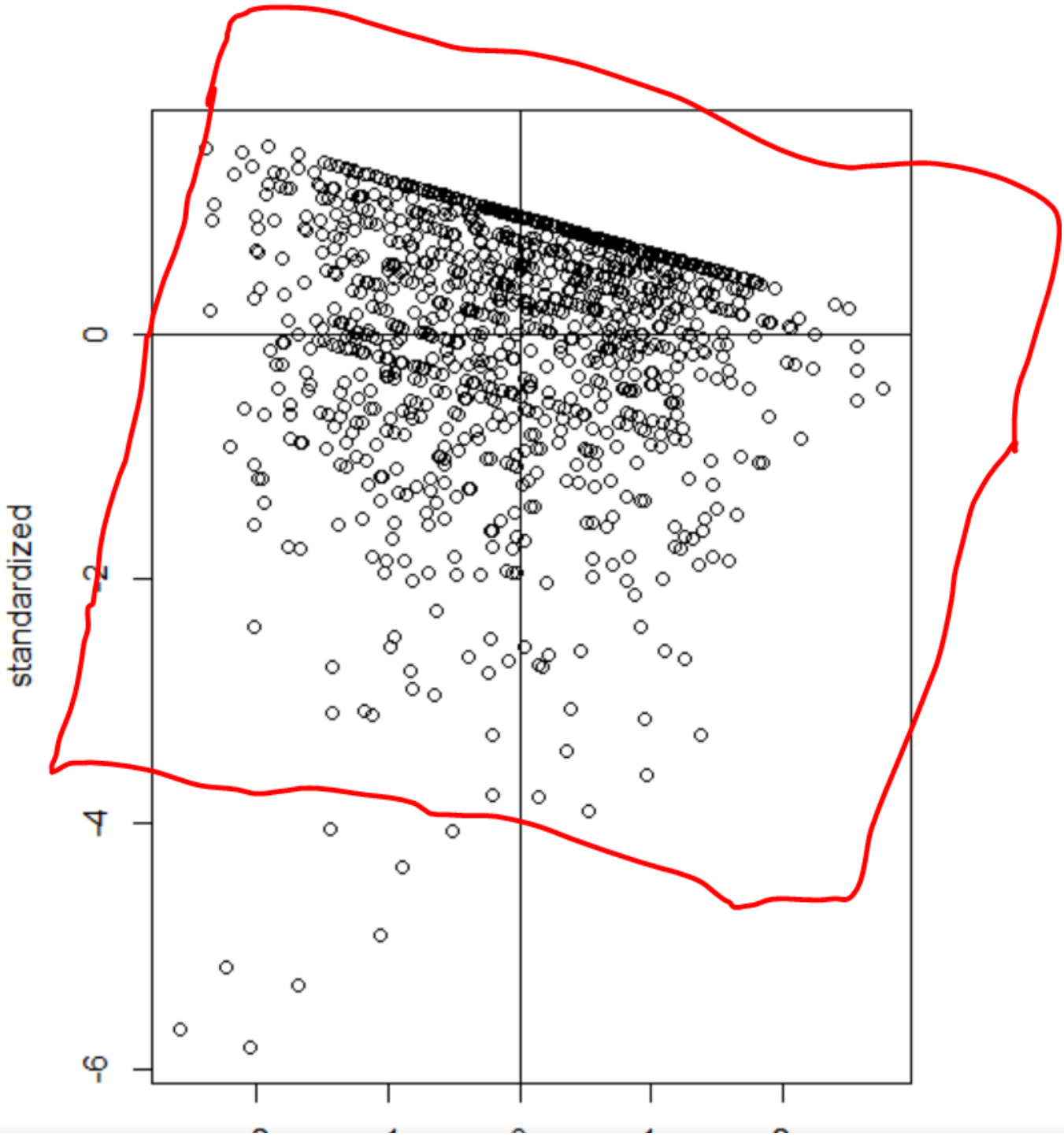
If my model is accurate(i.e., the abused tendency is predicted by ACE and BPT, account for control variables), residuals should be equally distributed throughout the abused scores and around zero. This is my output.



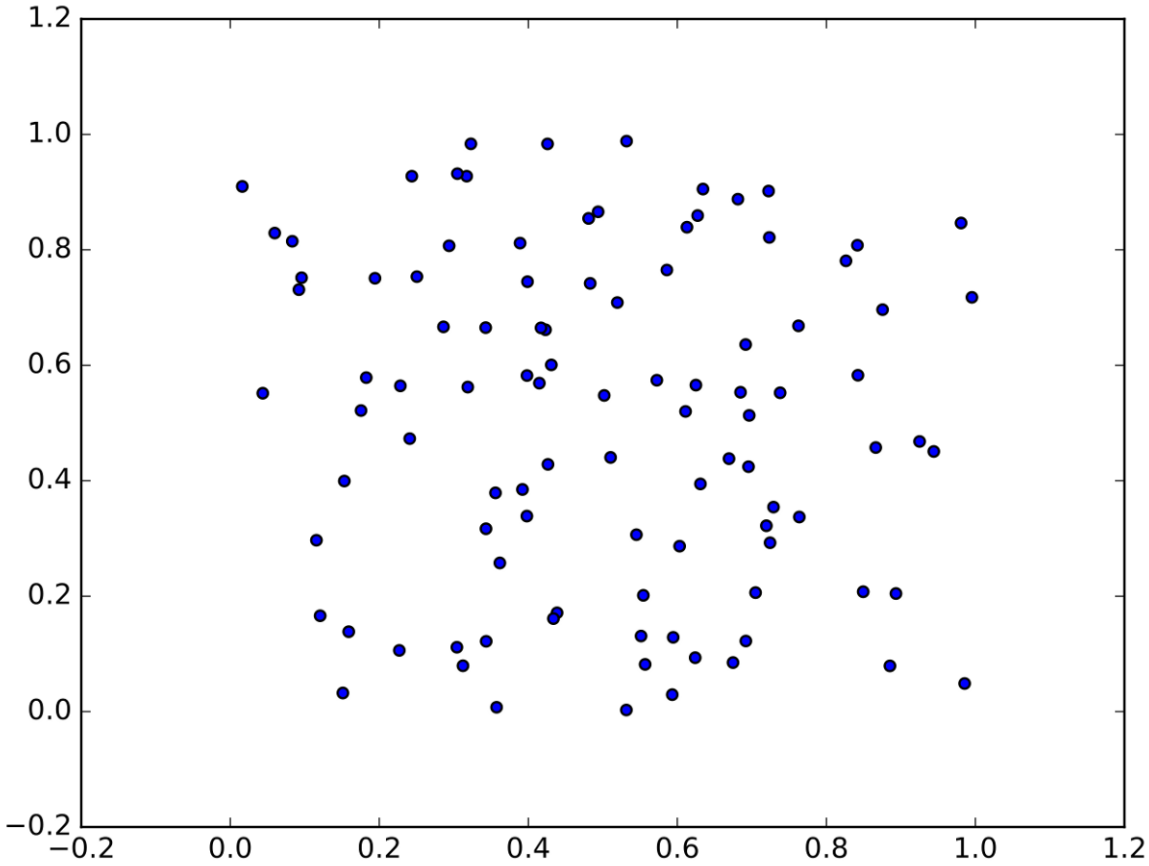

standardized
0
-2
-4
-6
-2
-1
0
1
2
fitted

factors may play roles in predicting one's tendency to being abused but have been omitted.

4. Independence of errors. This assumption refers to a random distribution of residuals. Residuals should be randomly distributed and do not form a specific shape. To examine whether the assumption is met, we can see from the scatter plot we generated earlier.



My scatter plot looks like a tilted rectangle with more data points below the zero value of the standardized axis and on the fitted axis' negative side, indicating a specific form or shape of errors. The results are consistent with my assumption checks of normality and homogeneity, confirming that some omitted predictors may explain my dependent variable's error variance. If the independence of errors is met, a scatter plot should look completely at random like the below. It should be noted that fitted values of the below plot are likely not standardized and thus data points are not centered around zero, although they look are randomly distributed.



Credit: Wikimedia Commons. A link to license:
https://commons.wikimedia.org/wiki/File:Mpl_example_scatter_plot.svg

I hope this tutorial is useful for your next data screening. For the full list of codes used in this tutorial, please see below.

```
1  #Import data
```

```
mahal = mahalanobis(data[ , -c(1:3)#categorical variables columns)], colMeans(data[ ,-c(
summary(mahal)
cutoff = qchisq(.999, ncol(data[ , -c(1:3)]))
summary(mahal < cutoff)

#Multicolinearity
correlation = cor(data[ , -c(1:3)], use = "pairwise.complete.obs")
symnum(correlation)

#Assumption check
#Fit a model
model = lm(DV ~ IVs, data = data)
fitted = scale(model$fitted.values)
standardized = rstudent(model)

#Alternative for model fitting in case that IVs are categorical
random = rchisq(nrow(data), 10#any number works fine)
fakemodel = lm(random ~ ., data = data)
fitted = scale(fakemodel$fitted.values)
standardized = rstudent(fakemodel)

#1.linearity
qqnorm(standardized)
abline(0,1)

#2. normality
hist(standardized)

#3. & 4. Independence of errors and homogeneity
plot(fitted, standardized)
abline(0,0)
abline(v = 0)
```