PCA of Launched Drug Data

Andrew Wang

10/25/2020

Introduction

In this report, I will be using Principal Component Analysis to analyze a set of drug data, where I'm to answer the research question: "Of the 15 different variables in this dataset, which contribute to the commercial success of these drugs?". However, before I can begin conducting PCA on this data, I need to explain some background information regarding drugs and their creation/testing.

Parmacokinetics

Parmacokinetics (PK), is the study of the effect that a drug has on the body. In PK, questions such as "How does the drug enter the body", or "Where does the drug go once entering the body", are answered. Often, the effects of a drug on the body are classified as ADME(T), Absorption, Distribution, Metabolism, Excretion, and Toxicity. These five areas are commonly used to analyze the impacts of a drug on the body. In these next paragraphs, I will go over what each of these terms means.

Absorption: Absorption answers the question of "How does the drug enter the body?". There are a variety of ways drugs can enter the body such as:

Topical: Applied to the skin, medicines like eye drops, ear drops, and skin creams

Patenteral: Taken through a needle

Enteral: Taken by pill. Regarding oral drugs, a rule known as "Lipinski's rule of five", is applied. In Lipinski's rule of five, five parameters for drugs are laid out which oral drugs must follow. They are:

- Molecular weight: Less than 500 g/mol.
- Hydrogen donors: Less than five.
- Hydrogen acceptors: Less than ten.
- LogP: Less than five.

This is why it is called Lipinski's rule of five, as the paramters have maximums which are all multiples of five.

Metabolism The metabolism refers to how the drug is process by the body, such as the organs involved in breaking it down. The liver is usually involved in breaking down drugs, specifically the CYP (Cytochrome P450) enzymes, which are responsible for 75% of total metabolism.

Excretion Excretion refers to how the drug leaves the body. Usually, drugs leave the body through urinary and fecal excretion, or through sweat pores in the skin.

Toxicity While in the past toxicity was not considered a analyzed factor, toxicity has become more important recently due to the large impacts of things like unintended side effects. In analyzing toxicity, researchers want to answer the question "Do the benefits of this drug outweigh its negative impacts?". Two common terms in measuring the toxicity include the ED50 and the LD50. These stand for effective-dose 50% and lethal-dose 50% (respectively). Effective-dose 50% is the drug dosage at which 50% of the test animals exhibit the intended benefits. Lethal-dose 50% is the drug dosage at which 50% of the test animals die.

Term definitions

Here, I am going to go over a few of the terms (columns) present in our data set.

Molecular weight: Measured in grams per mol, molecular weight is simply the sum of all atoms in a molecule.

LogP: LogP is a measure of lipophilicity, the “water-hating” property of certain molecules. LogP indicates the permeability of drugs to reach their target tissue.

LogS: LogS is a measure of the drug’s solubility. It is a 10-based logarithm of the solubility measure through mol / L.

Rotatable bonds: The number of bonds on a molecule that are free to rotate around themselves. These bonds must be single bonds, non-terminal (not at the end), and not including carbon-nitrogen bonds. The rotatable bonds in a molecule are particularly important for drug interactions with target proteins (docking).

Topological Polar Surface Area: The Topological Polar Surface Area (TPSA), is the sum of all polar surface area on the molecule. TPSA is measured in angstroms.

What is PCA?

I said I would be conducting PCA, Principal Component Analysis, on drug data at the beginning of this report, but what exactly is PCA and why is it useful? In this section I will discuss that.

PCA is a dimension reduction technique, where it takes a set of variables and looks for the relationships between them. It then uses the relationships between variables to produce components for analysis. With fewer dimensions, we can usually determine the relationship between variables easily through the use of scatterplots and 3D graphs, but at higher dimensions (columns), it becomes harder - this is why PCA is used. PCA will create a set of Principal Components, the number of which is equal to the number of dimensions your data contains. The Principal Components are also sorted in order of importance, with Principal Component 1 (PC1), being the most important, PC2 being the second most important, and so on and so forth.

Furthermore, by analyzing the Principal Components which account for the most variance in the data, we can determine what dimensions (columns) have the most impact on the data, thereby determining what columns have the most influence. While the exact specifics of PCA calculations are beyond the scope of this paper, interested readers can watch this video (https://www.youtube.com/watch?v=FgakZw6K1QQ&feature=emb_logo).

PCA analysis through R

The following sections will contain the R code I use to conduct a PCA analysis, which ultimately answered the research question.

Data setup and cleanup

In this section, I am going to review basic environment clean-up and imported packages, along with cleaning the data and making sure everything is ready for analysis. First, I clean up and set up the R environment with the working directory. Keep in mind that your directory structure is likely different from mine, and adjust accordingly.

```
# Andrew Wang
# October 25
# PCA of Launched Drug Data
#
# clean up and setup
rm(list=ls()) # clean up any old stuff in R
setwd("C:/Users/hyper/OneDrive/Desktop/Desktop Folders/Programming/R/Assignments/Week 8") # go t
o this folder
#load up myfunctions.R
source("C:/Users/hyper/OneDrive/Desktop/Desktop Folders/Programming/R/myfunctions.R")
```

After this is done, I import the packages I'm using. Keep in mind that you will likely need to install these packages first before using them.

```
#library import
library(tidyverse)
```

Finally, we can import our .csv file, and assign it to a variable. Notice that I've also added some optional arguments to my import, which specify the location of the row names in the importation. I then run some basic observations on the dataset as a starting point, in order to get a basic overview of the data we have.

```
drugs <- read.csv("drugs.csv", header=TRUE, row.names = 1)
str(drugs)
```

```
## 'data.frame':    1270 obs. of  14 variables:
##  $ logS          : num  3.23 2.15 6.36 5.4 3.49 ...
##  $ logSpH7       : num  1.93 3.89 3.81 5.13 2.95 ...
##  $ logP          : num  1.39 1.39 -1.92 -2.5 1.71 ...
##  $ logD          : num  0.41 4.289 -1.844 -0.917 -0.09 ...
##  $ X2C9pKi       : num  4.71 5.03 3.87 5.22 4.4 ...
##  $ hERGpIC50     : num  5.55 1.69 3.55 2.63 4.7 ...
##  $ BBB           : num  -0.441 -1.072 -0.47 -1.586 -0.15 ...
##  $ Pgpcategory   : int  1 1 0 1 1 0 0 0 0 0 ...
##  $ MW            : num  286 1416 181 646 336 ...
##  $ HBD           : int  3 13 2 14 3 2 2 1 2 2 ...
##  $ HBA           : int  7 28 5 19 6 3 7 2 6 3 ...
##  $ TPSA          : num  101.9 425 83.5 321.2 87.7 ...
##  $ Flexibility   : num  0.167 0.453 0.5 0.192 0.458 ...
##  $ RotatableBonds: int  4 48 5 9 11 2 3 0 6 1 ...
```

```
print(sum(is.na(drugs)))
```

```
## [1] 0
```

PCA analysis

I am now going to preform to run a PCA on our data. This is very simply executed through one line of code, which conducts the PCA. After that, I wish to look at the names of the new dataset, and I use the names() function to do so.

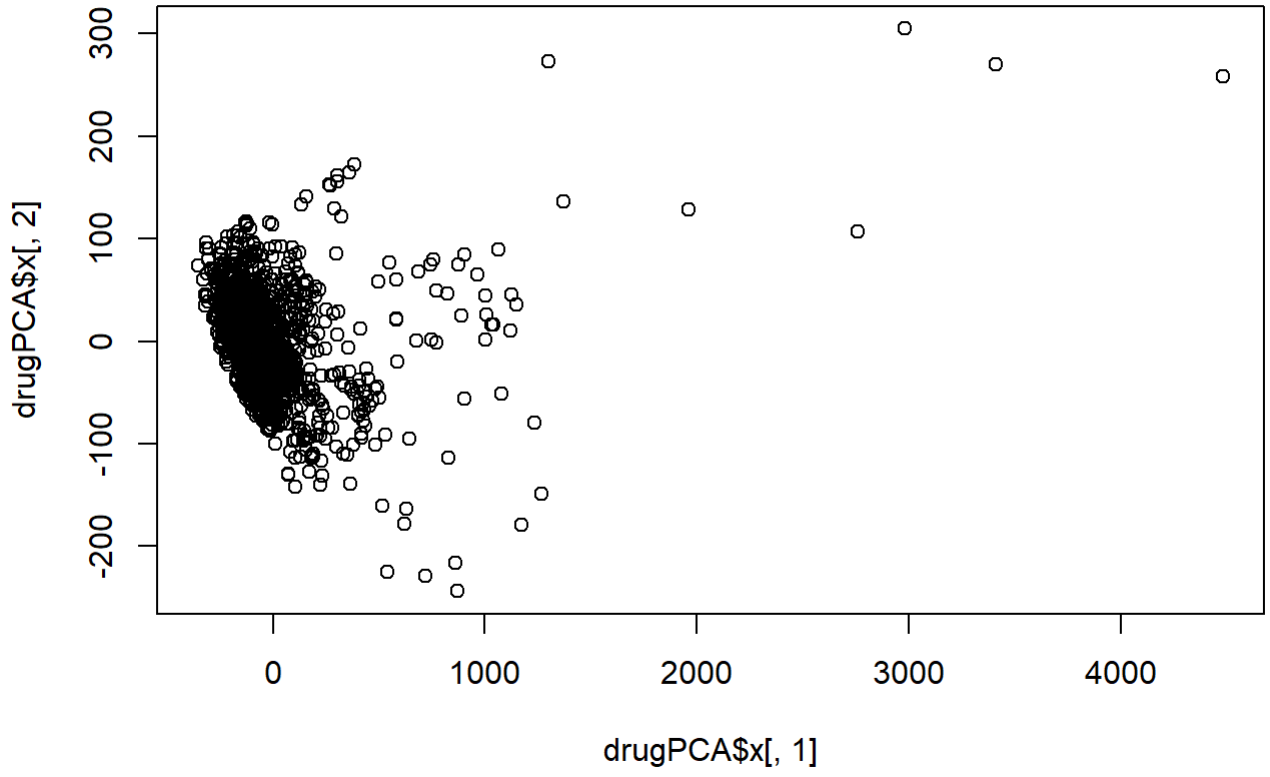
```
drugPCA <- prcomp(drugs)
names(drugPCA)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

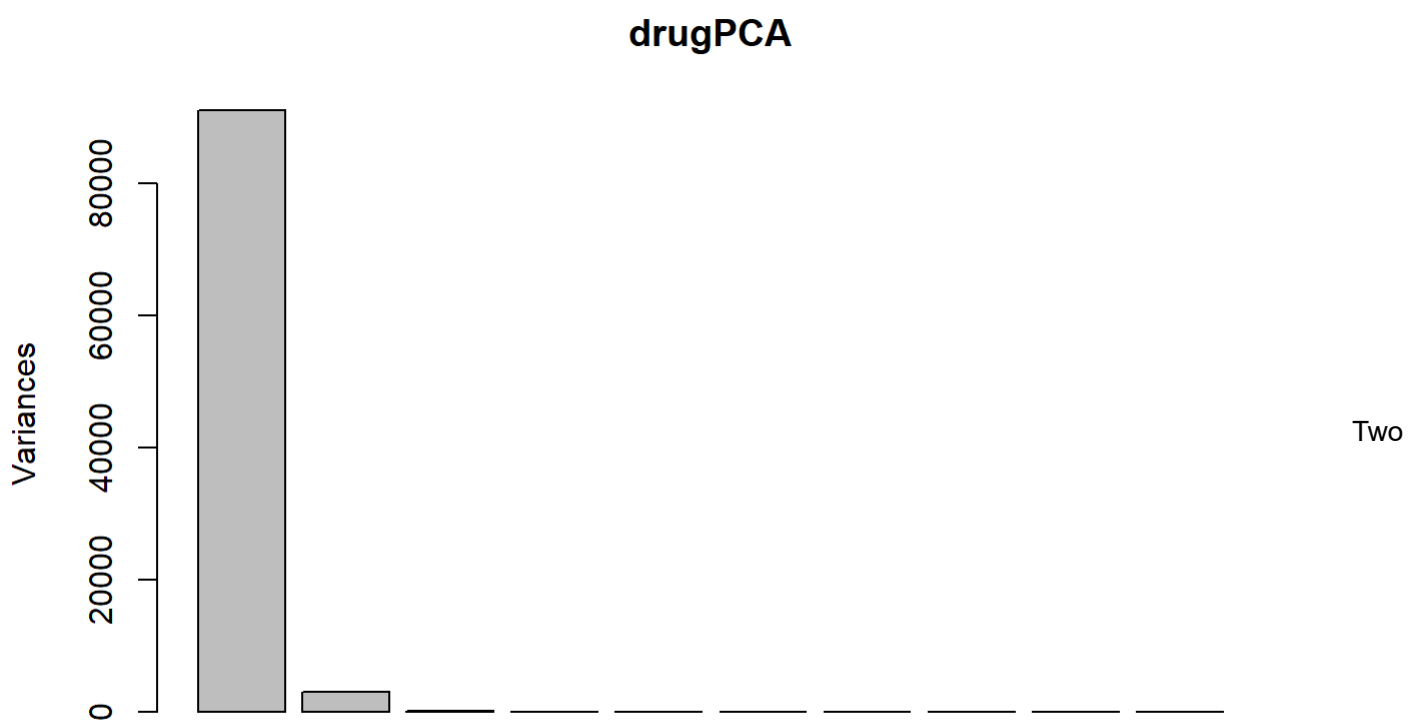
Basic plotting

I now want to get an overview of my PCA data. I want to do this so that I can make a comparison (in the next section) to transposed data, before deciding on a data version to use. First, I plot PC1 and PC2 on a simple scatterplot, and I follow it up with a screeplot, which shows all Principal Components in the form of a bar graph

```
#screeplot + plot
plot(drugPCA$x[,1],drugPCA$x[,2])
```




```
screeplot(drugPCA)
```



things to notice here. Firstly, the data from my scatterplot is extremely varied. This is not a good thing, as you generally want datapoints which are clustered, and therefore are more similar to each other. Secondly, notice in my screeplot the prevalence of PC1. This is expected, as PC1 is supposed to be the most important component.

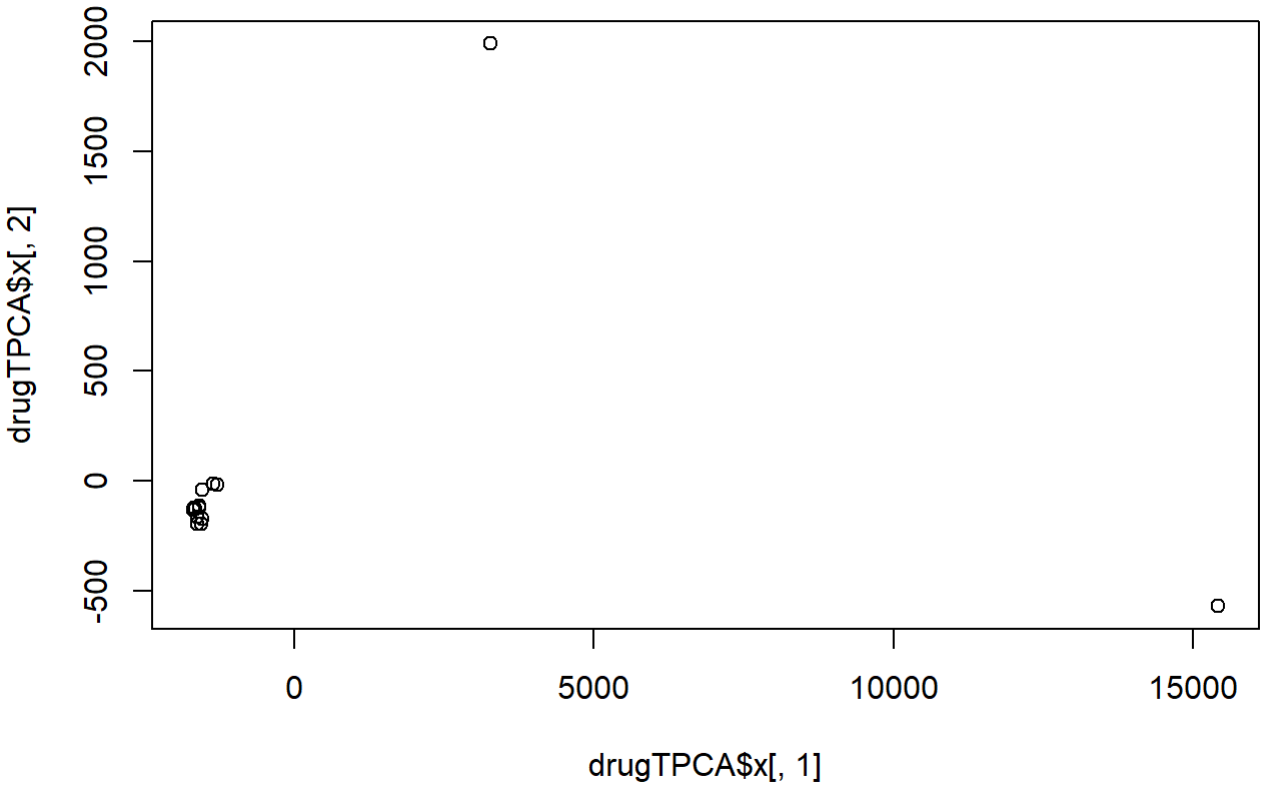
Transposing and comparing

Next, I want to transpose the original data, make new graphs for the transposed data, and make a decision on which data style is better for the PCA. In transposing the data, I merely flip the columns and the rows. Next, I conduct a PCA on the newly transposed data, and perform the same plotting that I did with the original data.

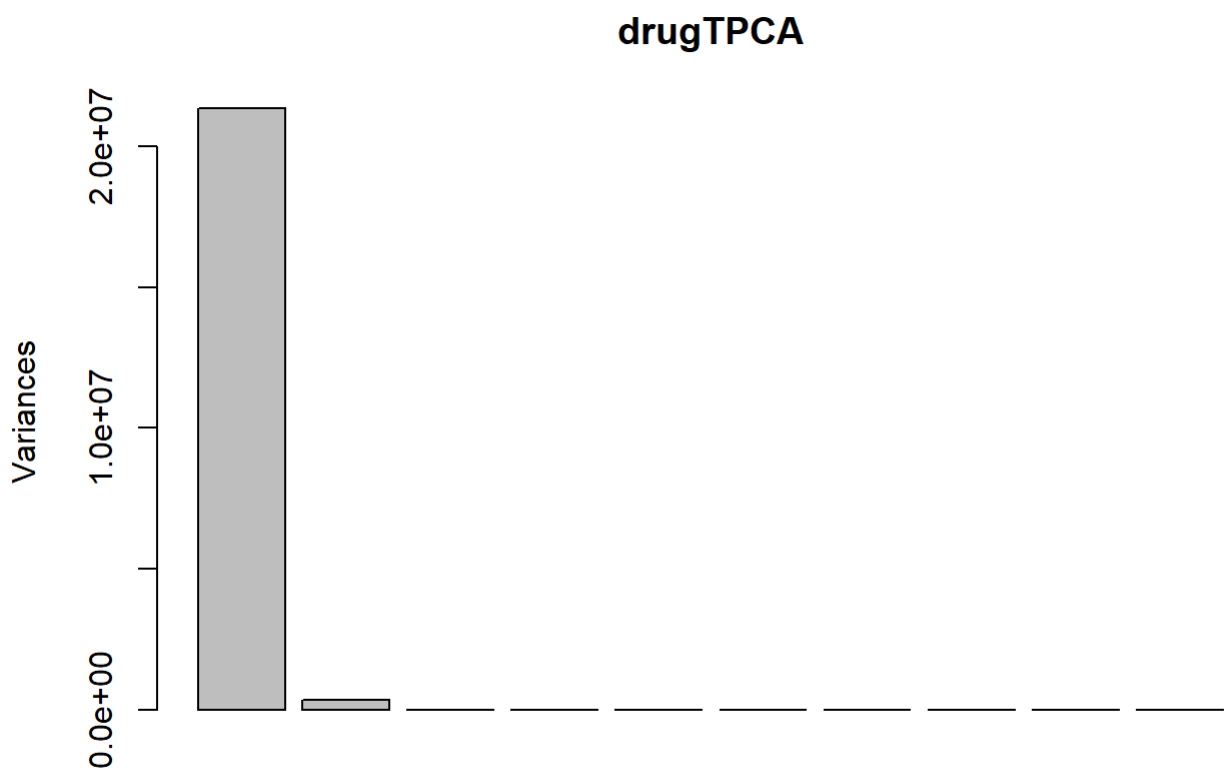
```
#transposed data + screeplot + plot
drugsT <- t(drugs)
drugTPCA <- prcomp(drugsT)
names(drugTPCA)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

```
plot(drugTPCA$x[,1],drugTPCA$x[,2])
```




```
screeplot(drugTPCA)
```



Notice two main things in these two new plots. Firstly, the datapoints are much more clustered. As mentioned earlier, we want data which is more clustered, as opposed to dispersed, because it will help make relationships between variables more clear. Secondly, notice that in our second screeplot, the distance between PC1 and PC2 has increased. This is also a good thing, as it means most of the variation comes from PC1, it decreases the total amount of analysis we need to do, because a smaller PC means it impacts the data less. Because of these two reasons, I decided that the transposed data would be better to conduct the PCA on, and I make our original PCA data the new transposed data.

```
#we like the transposed data better - lets keep it
drugPCA <- drugTPCA
```

Variance calculation

Next, I conduct a variance calculation, which determines the percentages that each factor is responsible for. First, I create a new variable to store the square of the standard deviation (the variance). Next, I divide each variance value by the total to calculate the percentage of variation that each factor is responsible for. Furthermore, I round these values (mostly for the aesthetics of the plots). I then make a barplot out of the variance data.

```
#variance calculation
drugVariance <- drugPCA$sdev^2
drugVariance <- round(drugVariance/sum(drugVariance)*100,1)
drugVariance
```

```
##  [1] 98.4  1.6  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
```

```
barplot(drugVariance)
```

Based on this bar graph, I can see that PC1 contributes an overwhelming amount to the variance in the data, 98.4% of variance to be exact.

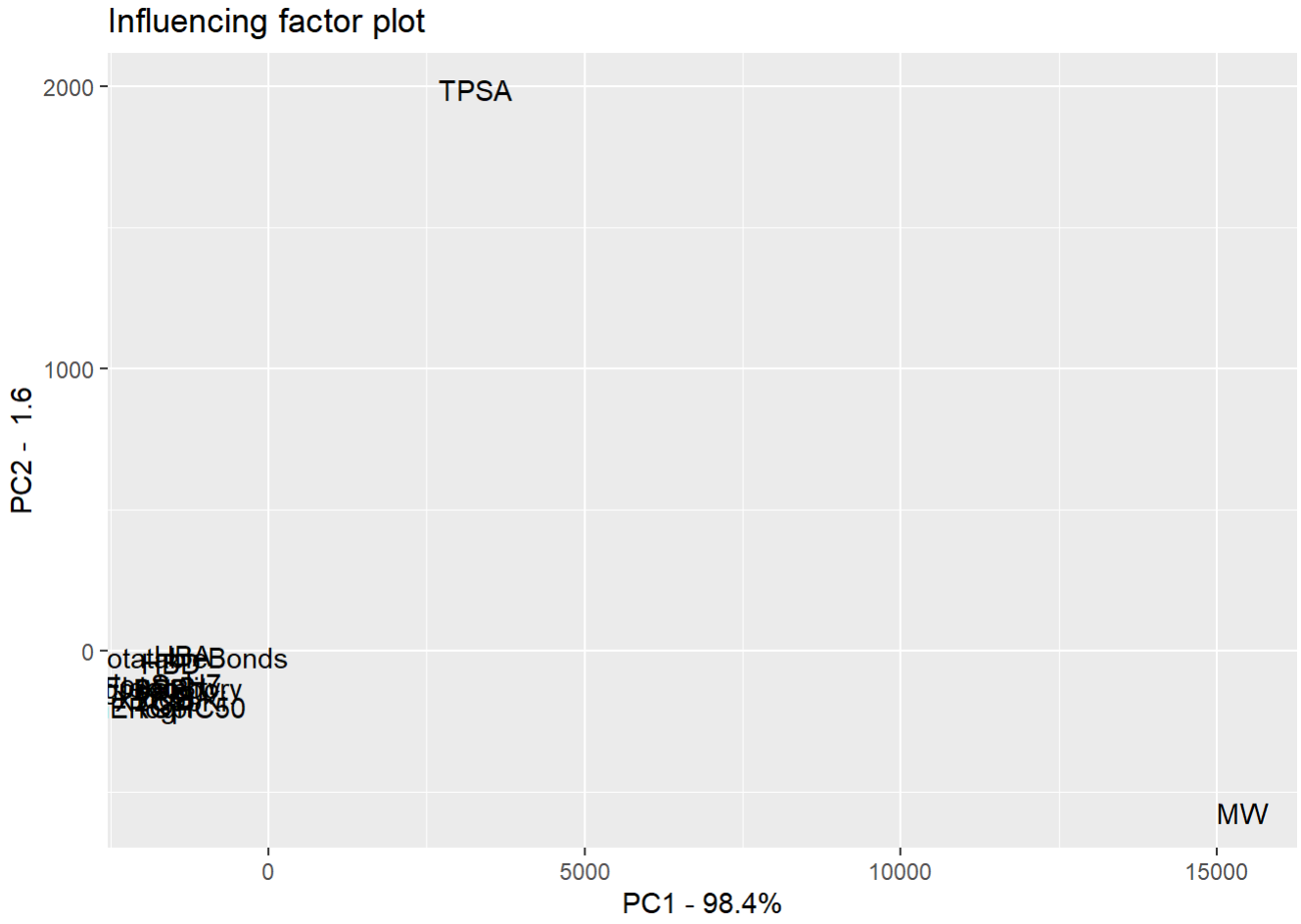
Contributing factor plot

For the final step, I use the ggplot package (built in with tidyverse) to plot the factors, so that I may determine which factor is the cause of the variation and therefore answer the research question. First, I create a dataset based off of the PC column in my PCA data. I put PC1 and PC2 into the data. While PC1 is my main target, A second axis is required to plot, and PC2 is a good second choice. Next, I create the ggplot, parsing in my columns into the plot, adding labels to the axes, and a title.

```
#contributing variable plot
drugPD <- data.frame(Sample=rownames(drugPCA$x), X=drugPCA$x[,1], Y=drugPCA$x[,2])
drugPD
```

```
##                         Sample         X          Y
## logS                       logS -1595.874 -122.65631
## logSpH7                 logSpH7 -1587.401 -115.04553
## logP                       logP -1615.724 -196.58613
## logD                       logD -1624.835 -163.57657
## X2C9pKi                 X2C9pKi -1534.006 -172.28378
## hERGpIC50             hERGpIC50 -1552.506 -197.79457
## BBB                         BBB -1691.092 -129.84905
## Pgpcategory         Pgpcategory -1655.722 -126.00836
## MW                           MW 15414.245 -570.24446
## HBD                         HBD -1530.992  -39.98425
## HBA                         HBA -1347.245  -14.65678
## TPSA                       TPSA  3273.529 1991.03211
## Flexibility         Flexibility -1665.673 -123.23515
## RotatableBonds RotatableBonds -1286.704  -19.11118
```

```
drugP <- ggplot(data=drugPD, aes(x=X, y=Y, label=Sample)) +
  geom_text() +
  xlab(paste("PC1 - ", drugVariance[1], "%", sep="")) +
  ylab(paste("PC2 - ", drugVariance[2]))+
  ggtitle("Influencing factor plot")
drugP
```



Looking at our plot, we finally see the source of the variation: MW - Molecular Weight.

Conclusion

At the start of this paper, I defined a reserach question to be answered through PCA, which was “Of the 15 different variables in this dataset, which contribute to the commercial success of these drugs?”. Through my PCA analysis of the data, and secondary factor analysis, I have reached the conclusion that molecular weight is the most important factor to consider in the commercial success of drugs. I was able to reach such a conclusion through a series of steps: transposing the data, conducing PCA on the data, and finally outputting a factor plot. This final step showed that molecular weight (MW), was the most important factor in PC1, which was responsible for the most variation in the dataset (98.4%).