National Institute of Technology Karnataka, Surathkal

Department of Computer Science and Engineering

Big Data Analytics (CS473)

Tutorial

Submitted to

Dr. Asoke Talukder
Adjunct Professor
Department of Computer Science and Engineering
National Institute of Technology Karnataka, Surathkal

Submitted by

Manas Trivedi
B.Tech. Class of 2022
Department of Computer Science and Engineering
National Institute of Technology Karnataka, Surathkal
Roll no. 181CO231

Date: April 4, 2022

Index

Problem Statement

lish_moa_annotated.csv.zip is a big data dataset that contains 27796 rows and 1486 columns. The first few columns are used for identification, columns with prefix 'g-' correspond to gene names, columns with prefix 'c-' correspond to cell lines, and the remaining columns correspond to Mechanisms of Action (MoA). For this dataset, Exploratory Data Analysis (EDA) needs to be performed.

Exploratory Data Analysis

Initial Analysis

The given dataset is a set of rows, where each row corresponds to a cell perturbation experiment. In each experiment (sig_id), a drug (with drug_id), having cp_type as either real (trt_cp) or vehicle (ctl_vehicle), is applied to a sample for cp_time 24/48/72 hrs, in either D1 or D2 dosage. The sample contains multiple genes (g-*) and cell lines (c-*). During the experiment, each gene becomes either up-regulated or down_regulated by a certain value, and each cell line either proliferates or decreases by a certain value. For each experiment, multiple mechanisms of action (MoAs) are either associated (value 1) or not associated (value 0).

We begin with cleaning the environment, noting the execution start time, reading the dataset into a dataframe called data, and noting the number of rows and columns.

```
> rm(list=ls())
> startTime <- Sys.time()
> data <-
readr::read_csv("C:/Users/Manas/Learn/BDA/Tutorial/lish_moa_annot
ated.csv.zip")
> print(nrow(data)) # 27796
[1] 27796
```

```
> print(ncol(data)) # 1486
[1] 1486
```

Next, we check if sig_id can be used to identify each row of the dataset, i.e. whether each sig_id is unique.

```
> print(length(unique(data$sig_id)))
[1] 27796
```

Since the number of unique sig_ids is equal to the number of rows in the dataset, all sig_ids in the dataset are unique.

We now find the number of genes and cell lines. This will give us the indices for the same.

```
> print(sum(grepl("^g-", colnames(data))))
[1] 772
> print(sum(grepl("^c-", colnames(data))))
[1] 100
```

Thus, the columns corresponding to genes are from index 7 to 778, and the columns corresponding to cell lines are from index 779 to 878.

The remaining columns correspond to MoAs, and hence are from index 879 to 1486, i.e. 608 MoAs.

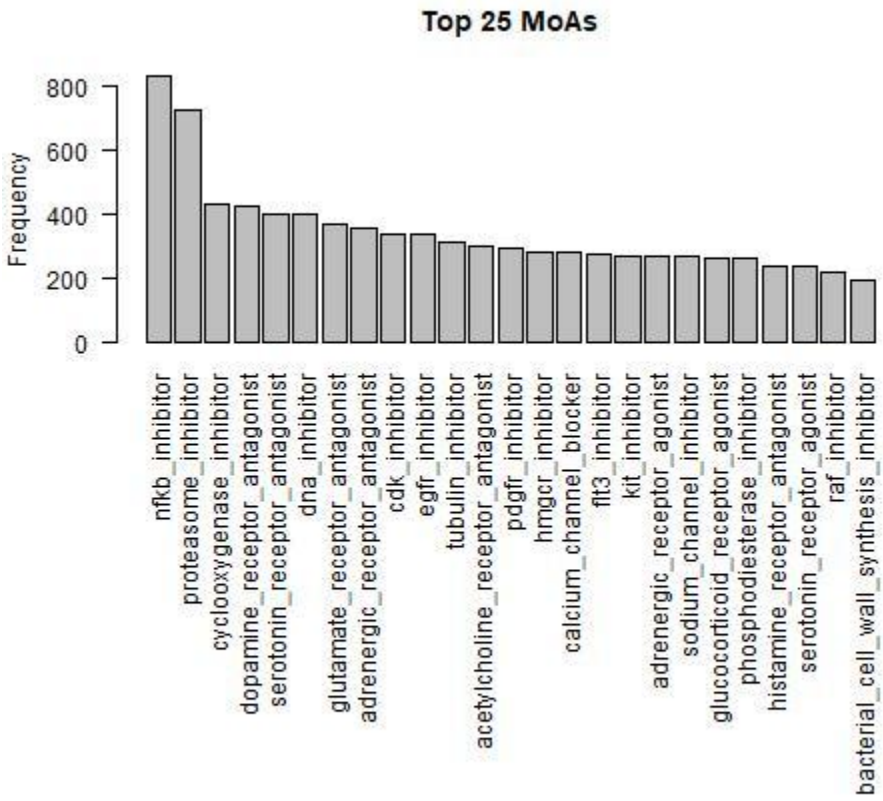
We shall use these indices in further R statements.

Distribution of MoAs

Since the dataset was derived from a challenge where the goal was to predict the MoAs by using the genes, cell lines, and other columns as the input, we can begin our exploratory data analysis by finding the distribution of the MoAs across the dataset.

For this, we will find the sum of all values in each MoA column to get a frequency vector called moaFreqVector for the MoAs. We will then find the top 25 and bottom 25 MoAs and plot them in a bar graph.

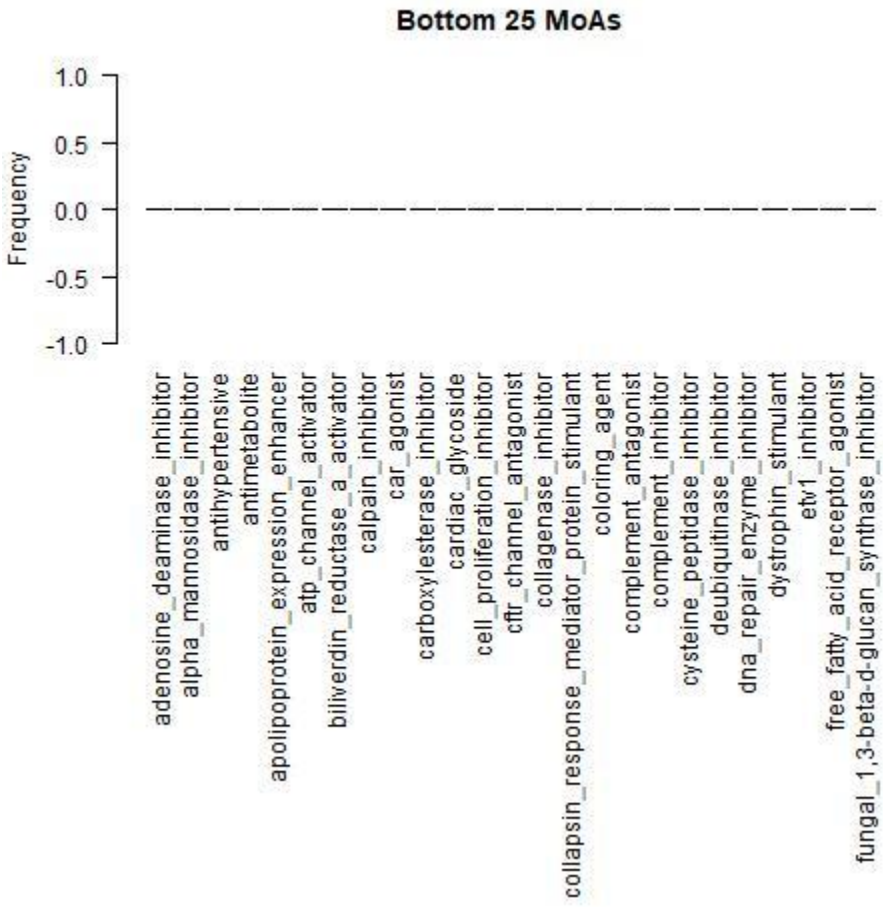
```
> moaFreqVector <- colSums(data[879:1486], na.rm = TRUE)
> jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/top25MoAs.jpeg")
> par(mar=c(20, 4, 4, 2))
> barplot(sort(moaFreqVector, decreasing = TRUE)[1:25], main =
"Top 25 MoAs", ylab = "Frequency", las = 2)
> dev.off()
```



Here, we see that nfkb_inhibitor and proteasome_inhibitor are the two highest-occurring MoAs across the dataset. This means that these two inhibitors are likely to be the most popular mechanisms of action in anti-cancer drugs.

Next, we note the bottom 25 MoAs.

```
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/bottom25MoAs.jpeg")
> par(mar=c(20, 4, 4, 2))
> barplot(sort(moaFreqVector, decreasing = FALSE)[1:25], main =
"Bottom 25 MoAs", ylab = "Frequency", las = 2)
> dev.off()
```



The bottom 25 MoAs have zero occurrences in the dataset. Thus, it is likely that these mechanisms of action do not exist in most, if not all, anti cancer drugs. But there may be more such MoAs in the dataset that have zero occurrences. We find the number of such MoAs through the following command.

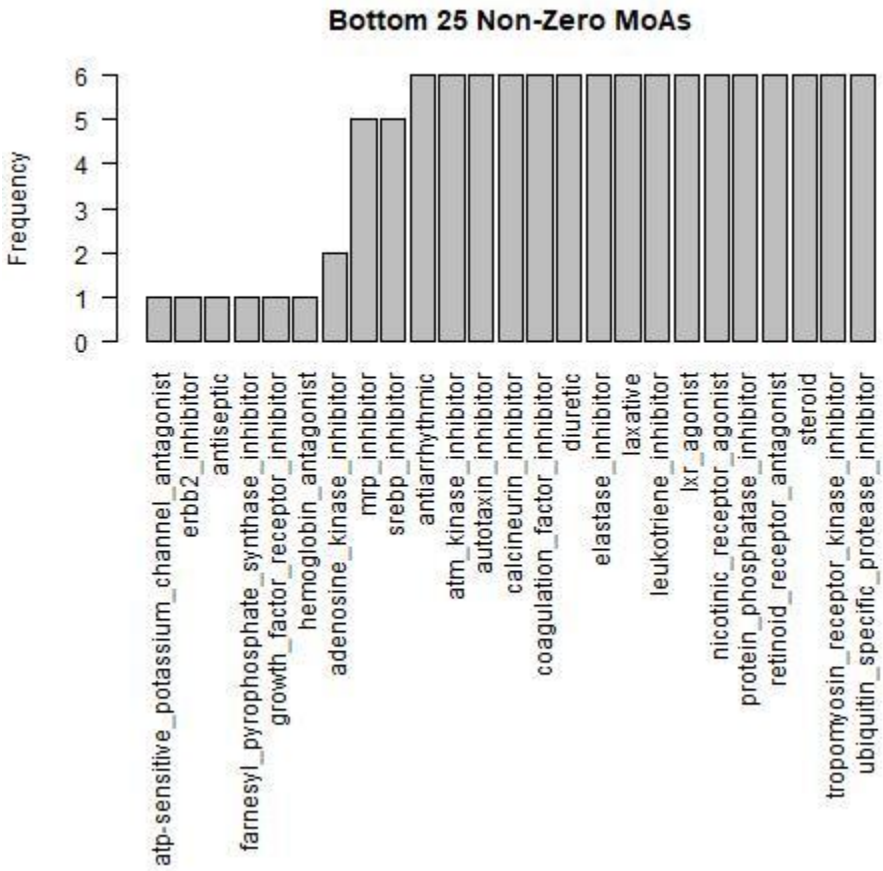
```
> length(moaFreqVector[which(moaFreqVector == 0)])
```

```
[1] 71
```

Thus, 71 out of 608 MoAs have no occurrences in the dataset.

We now find the bottom 25 MoAs that do have occurrences in the dataset.

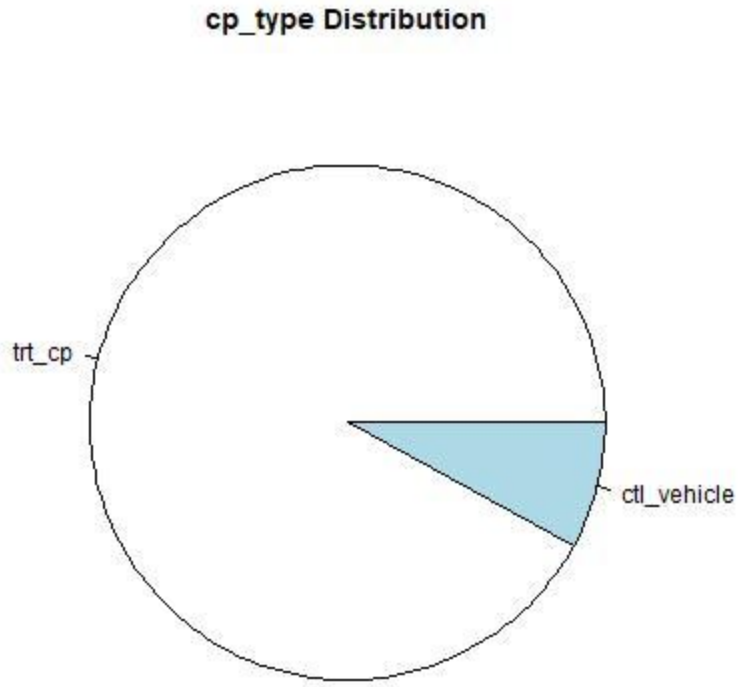
```
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/bottom25NonZeroMoAs.
jpeg")
> par(mar=c(20, 4, 4, 2))
> barplot(sort(moaFreqVector[moaFreqVector != 0], decreasing =
FALSE)[1:25], main = "Bottom 25 Non-Zero MoAs", ylab =
"Frequency", las = 2)
> dev.off()
```



Distribution of drug type, duration, and dosage

Each drug in the dataset is either a real drug (trt_cp) or a control vehicle (ctl_vehicle) with no biological effects. We visualize the distribution of these two drug types across the dataset using a pie chart.

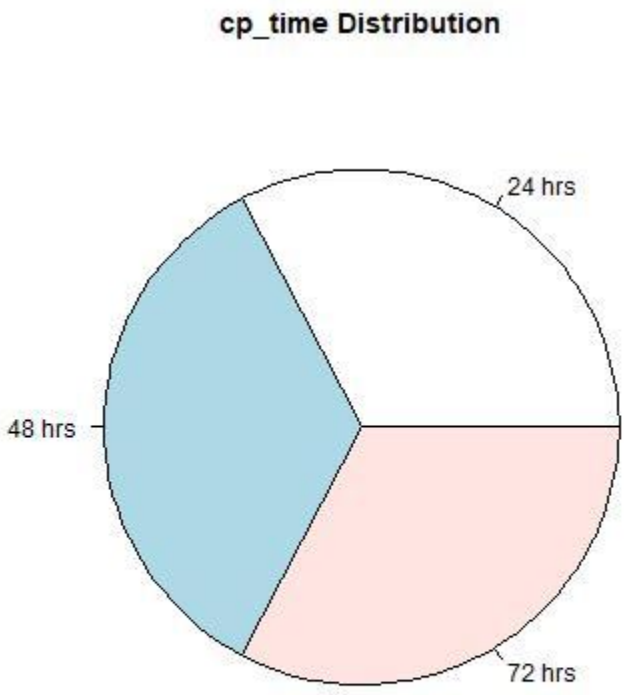
```
> cpTypeVector <- data[["cp_type"]]
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/cpTypeDistribution.j
peg")
> pie(c(length(cpTypeVector[grepl('trt_cp', cpTypeVector)]),
length(cpTypeVector[grepl('ctl_vehicle', cpTypeVector)])), labels
= c("trt_cp", "ctl_vehicle"), main = "cp_type Distribution")
> dev.off()
```



Thus, most of the drugs used in the experiments are real drugs.

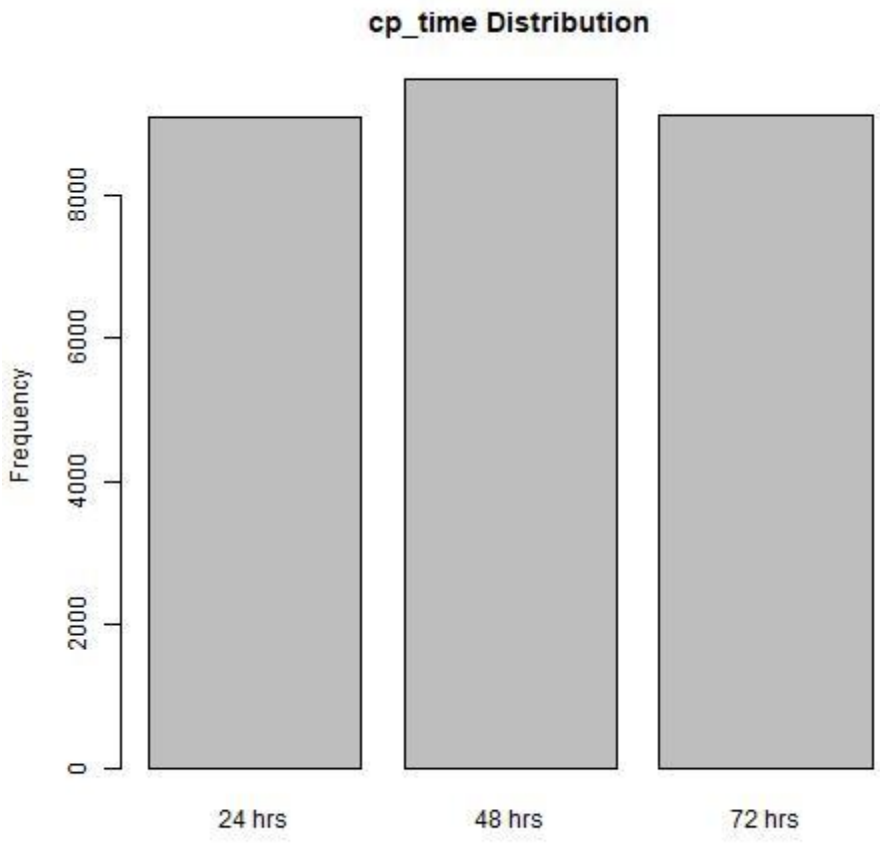
All experiments for constructing the dataset were conducted for either 24, 48, or 72 hours. We visualize the distribution of these durations across the dataset with a pie chart.

```
> cpTimeVector <- data[["cp_time"]]
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/cpTimeDistribution.j
peg")
> pie(c(length(cpTimeVector[grepl('24', cpTimeVector)]),
length(cpTimeVector[grepl('48', cpTimeVector)]),
length(cpTimeVector[grepl('72', cpTimeVector)])), labels = c("24
hrs", "48 hrs", "72 hrs"), main = "cp_time Distribution")
> dev.off()
```



All three durations have nearly the same distribution across the dataset. To note the slight difference in distributions, we visualize the same using a bar plot.

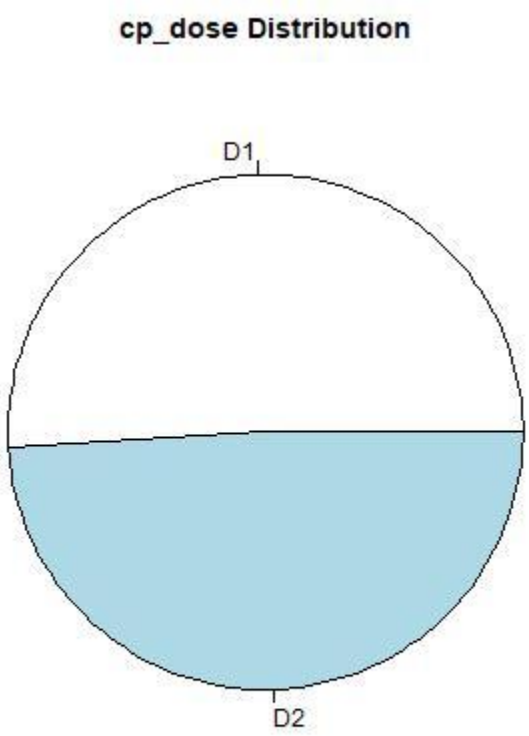
```
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/cpTimeDistributionBa
r.jpeg")
> barplot(c(length(cpTimeVector[grepl('24', cpTimeVector)]),
length(cpTimeVector[grepl('48', cpTimeVector)]),
length(cpTimeVector[grepl('72', cpTimeVector)])), names.arg =
c("24 hrs", "48 hrs", "72 hrs"), main = "cp_time Distribution",
ylab = "Frequency")
> dev.off()
```



Thus, 48 hours is the duration of more than a third of the experiments, followed by 72 hours and 24 hours being the duration of less than a third each.

All drugs in the dataset were given in either D1 or D2 dosage amounts. We now visualize the distribution of these dosages across the dataset.

```
> cpDoseVector <- data[["cp_dose"]]
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/cpDoseDistributionPi
e.jpeg")
> pie(c(length(cpDoseVector[grepl('D1', cpDoseVector)]),
length(cpDoseVector[grepl('D2', cpDoseVector)])), labels =
c("D1", "D2"), main = "cp_dose Distribution")
> dev.off()
```



Dosages D1 and D2 are nearly equal in the distribution, with D1 having slightly more frequency than D2.

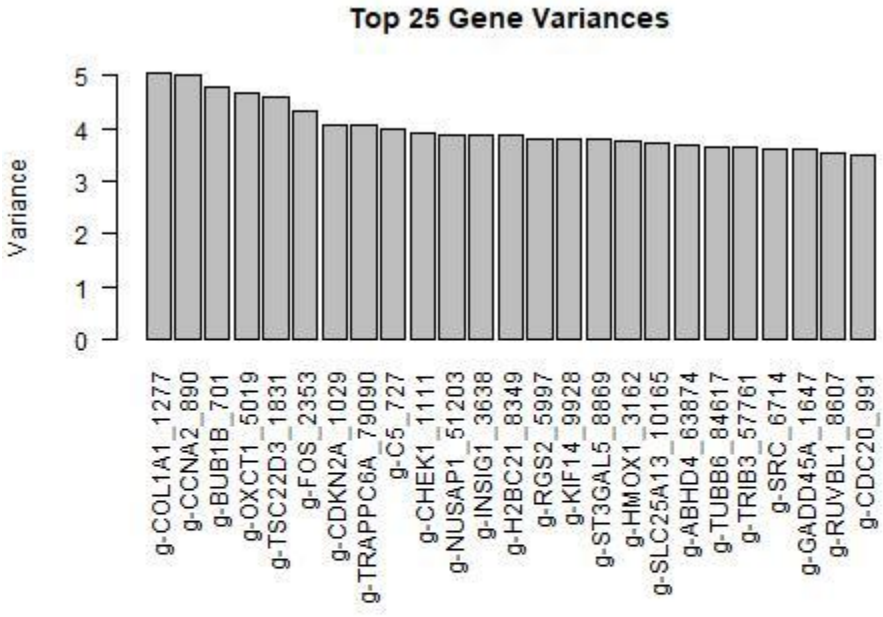
Variances of gene expression

In each experiment, we note a value for each gene. This value corresponds to the gene expression level - a positive value for a particular gene indicates that the gene became up-regulated during the experiment, while negative values indicate

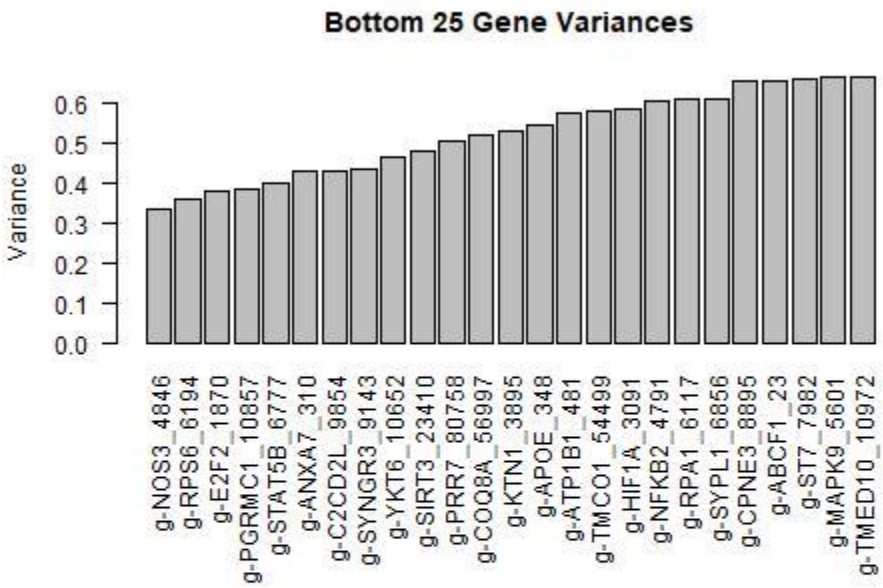
down-regulation. The magnitude of this value indicates the magnitude of regulation.

Some genes have more variance in their regulation values than other genes, which means that those genes are affected more by the drugs used in the experiment. We find the top 25 and bottom 25 gene variances. For this, we shall use the colVars() function present in the matrixStats package.

```
> library("matrixStats")
> geneVarianceVector <- colVars(as.matrix(data[7:778]), na.rm =
TRUE)
> names(geneVarianceVector) <- colnames(data[7:778])
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/top25GeneVariances.j
peg")
> par(mar=c(20, 4, 4, 2))
> barplot(sort(geneVarianceVector, decreasing = TRUE)[1:25], main
= "Top 25 Gene Variances", ylab = "Variance", las = 2)
> dev.off()
```



```
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/bottom25GeneVariance
s.jpeg")
> par(mar=c(20, 4, 4, 2))
> barplot(sort(geneVarianceVector, decreasing = FALSE)[1:25],
main = "Bottom 25 Gene Variances", ylab = "Variance", las = 2)
> dev.off()
```



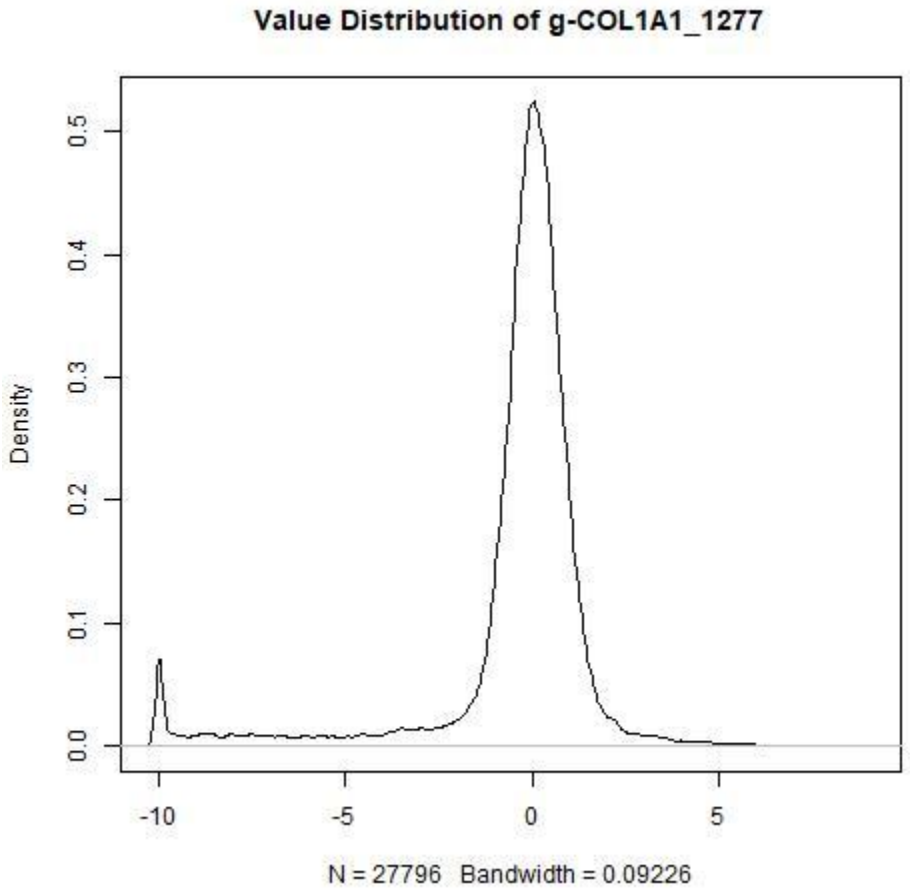
From the above two bar plots, we see that g-COL1A1_1277, g-CCNA2_890, and g-BUB1B_701 are the three genes with highest variance, and g-NOS3_4846, g-RPS6_6194, and g-E2F2_1870 are the three genes with lowest variance.

Distribution of values for top 3 genes with highest variances

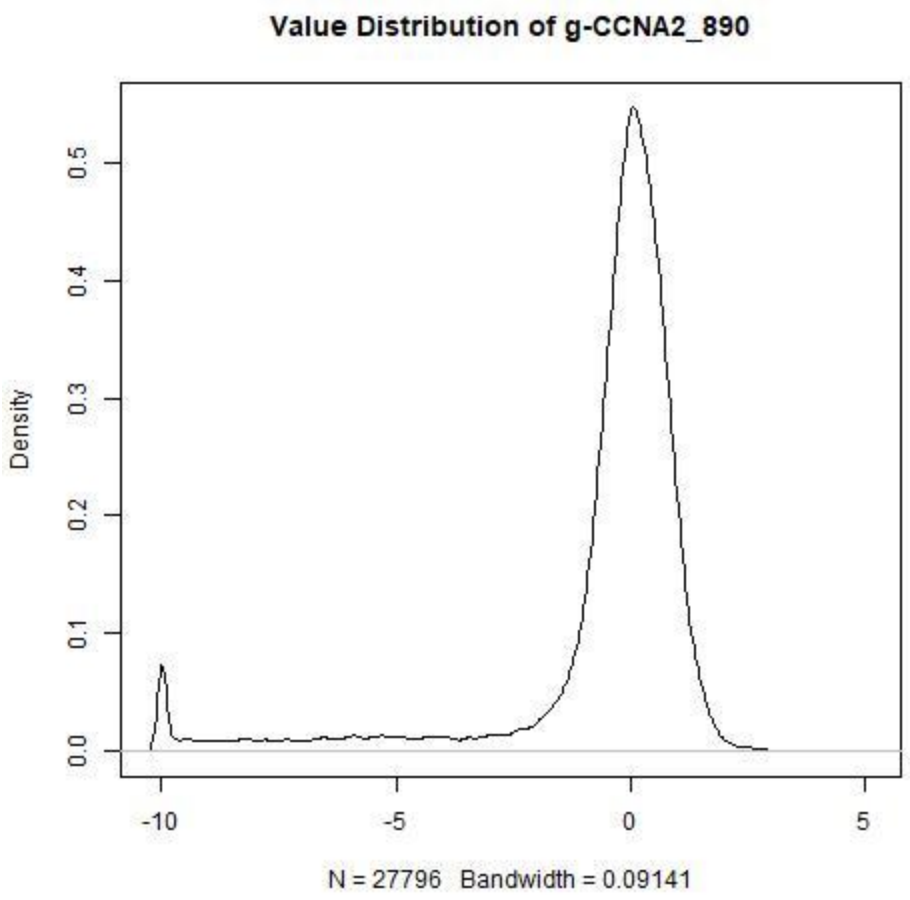
To view the distribution of values for g-COL1A1_1277, g-CCNA2_890, and g-BUB1B_701, we execute the following code.

```
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/valueDistributionOfg
-COL1A1_1277.jpeg")
```

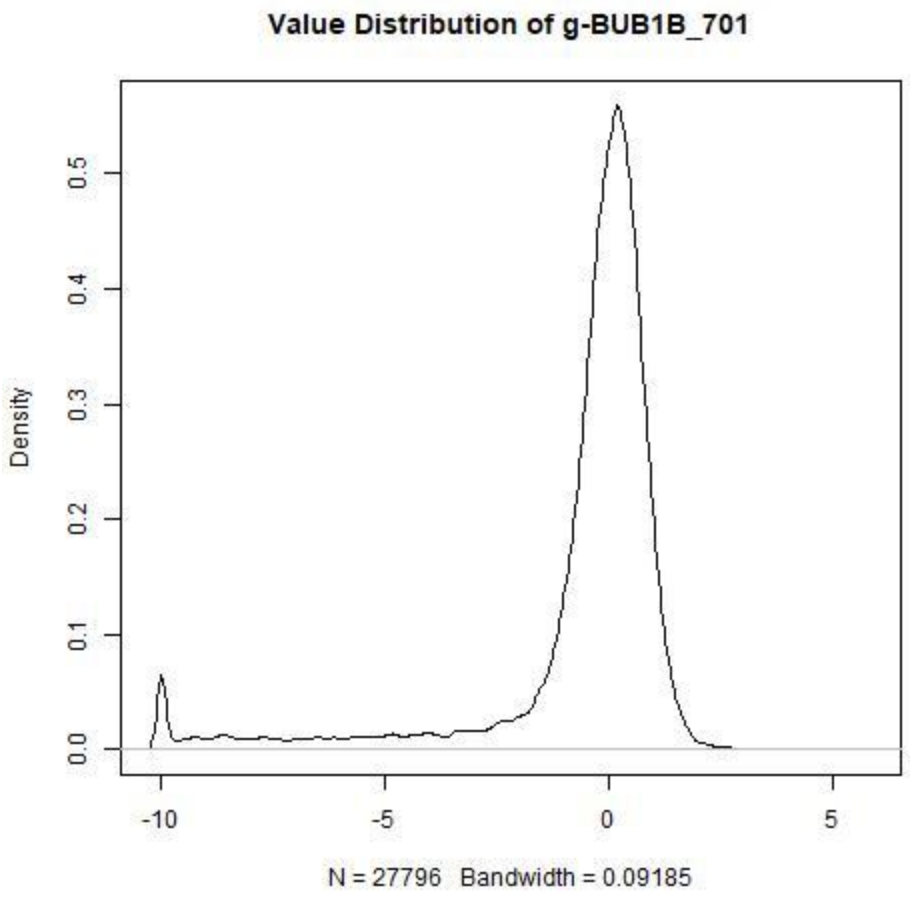
```
> plot(density(data[["g-COL1A1_1277"]]), main = "Value
Distribution of g-COL1A1_1277")
> dev.off()
```




```
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/valueDistributionOfg
-CCNA2_890.jpeg")
> plot(density(data[["g-CCNA2_890"]]), main = "Value Distribution
of g-CCNA2_890")
> dev.off()
```




```
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/valueDistributionOfg
-BUB1B_701.jpeg")
> plot(density(data[["g-BUB1B_701"]]), main = "Value Distribution
of g-BUB1B_701")
> dev.off()
```



From the above three plots, we note that all three distributions, for the three genes with highest variance, are centred at zero. Values close to zero indicate that in most experiments, the genes were up-regulated or down-regulated by a very small amount.

Another notable feature of these plots is that there is a significant bump at the value -10. This indicates that the gene expression values were likely limited to the range -10 to 10, and that any values exceeding -10 or 10 were set as -10 and 10 respectively.

Upon researching the above three genes with the highest variance, all three were found to be mentioned in the paper titled 'Identification of genes associated with osteoarthritis by microarray analysis' by Sun et al. The paper states that the CCNA2 and BUB1B genes are associated with the cell cycle.

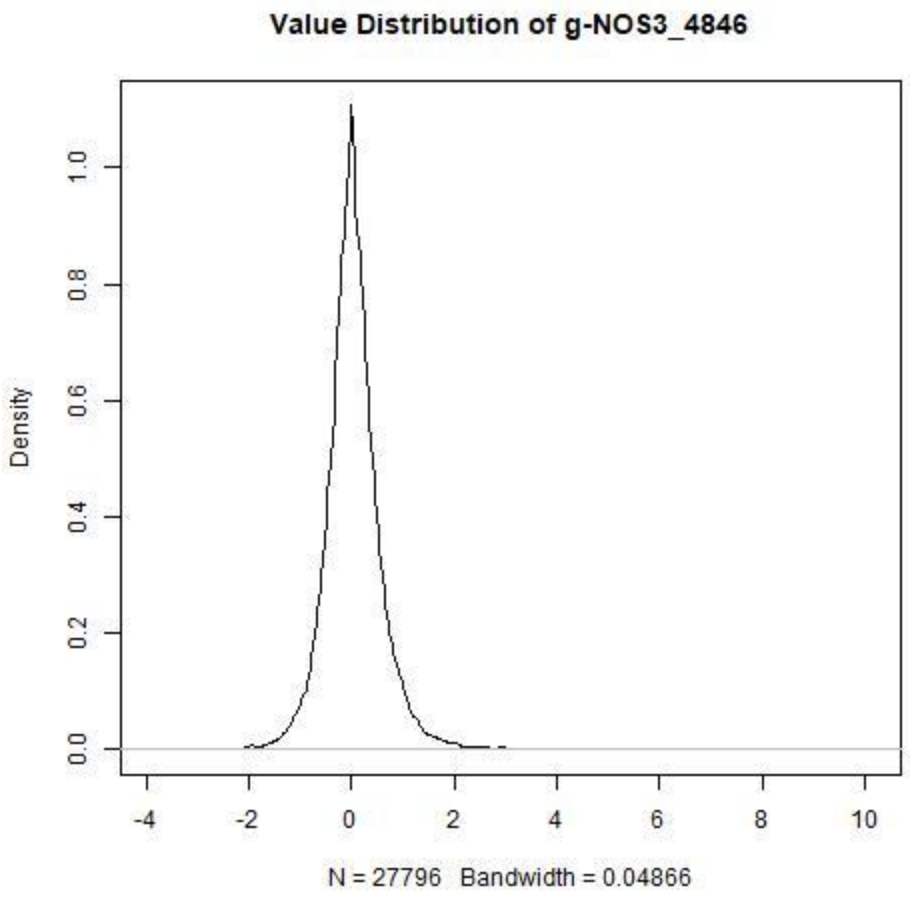
Upon researching the cell cycle, a paper titled 'The cell cycle and cancer' by Collins et al. stated the overlap of cell cycle and cancer.

Thus, it can be inferred that all three aforementioned genes have an important role in the success of anticancer drugs.

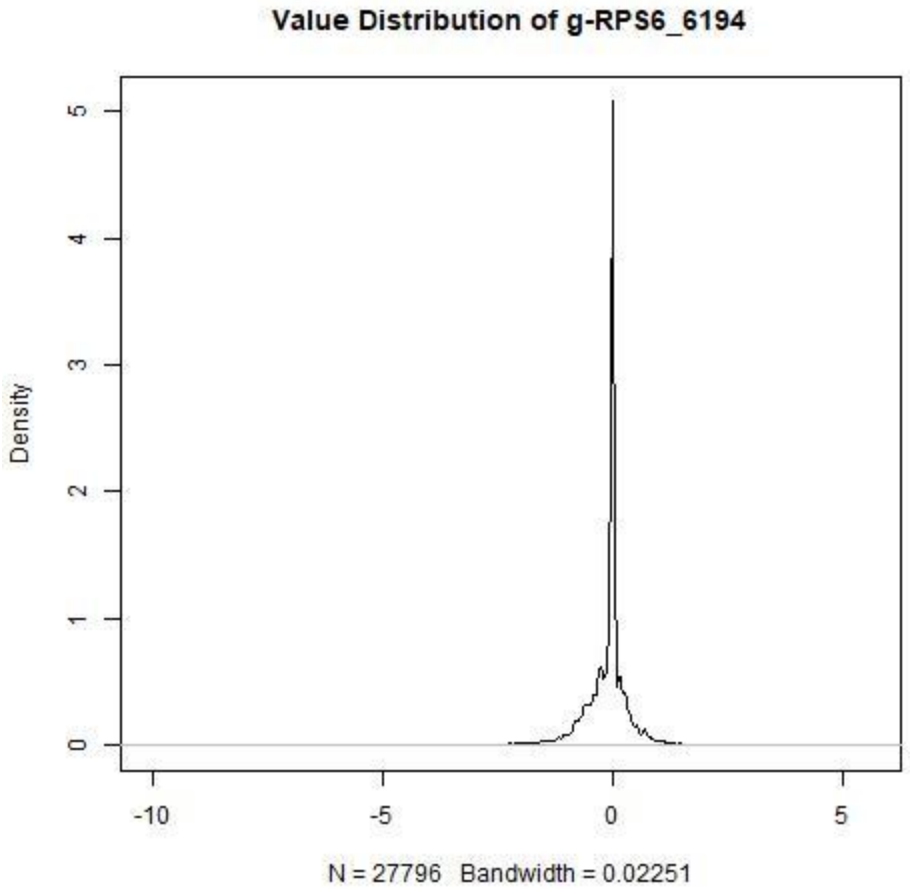
Distribution of values for bottom 3 genes with lowest variances

To view the distribution values for g-NOS3_4846, g-RPS6_6194, and g-E2F2_1870, we execute the following code.

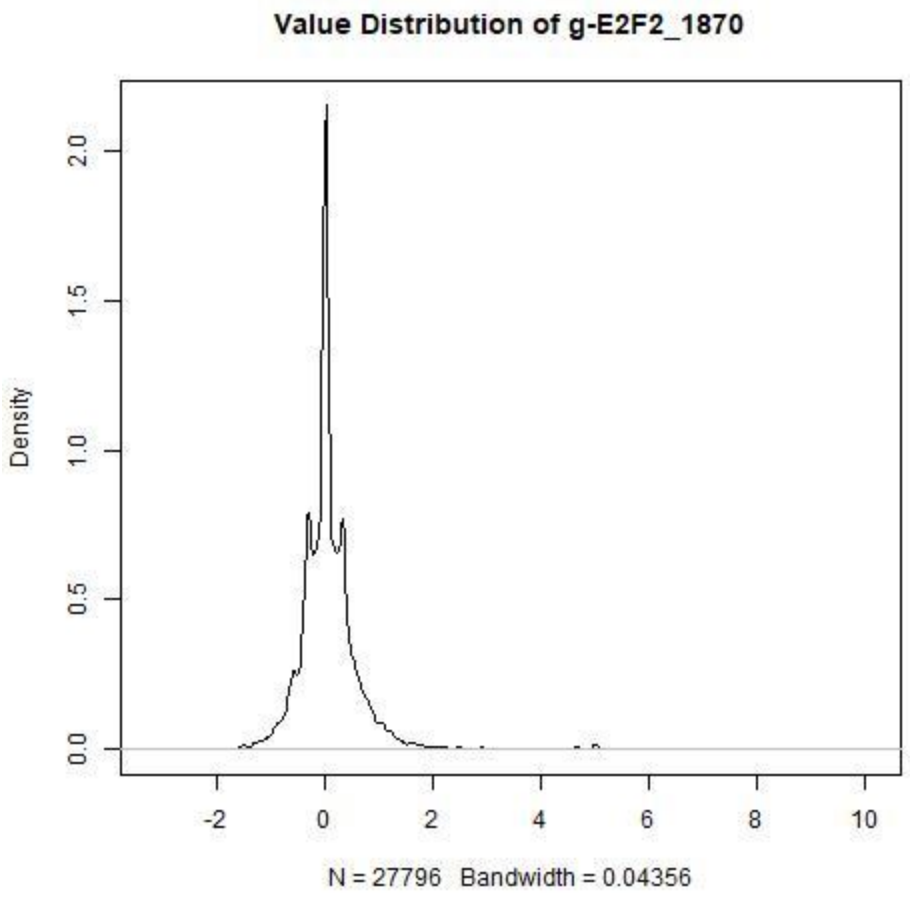
```
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/valueDistributionOfg
-NOS3_4846.jpeg")
> plot(density(data[["g-NOS3_4846"]]), main = "Value Distribution
of g-NOS3_4846")
> dev.off()
```




```
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/valueDistributionOfg
-RPS6_6194.jpeg")
> plot(density(data[["g-RPS6_6194"]]), main = "Value Distribution
of g-RPS6_6194")
> dev.off()
```




```
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/valueDistributionOfg
-E2F2_1870.jpeg")
> plot(density(data[["g-E2F2_1870"]]), main = "Value Distribution
of g-E2F2_1870")
> dev.off()
```



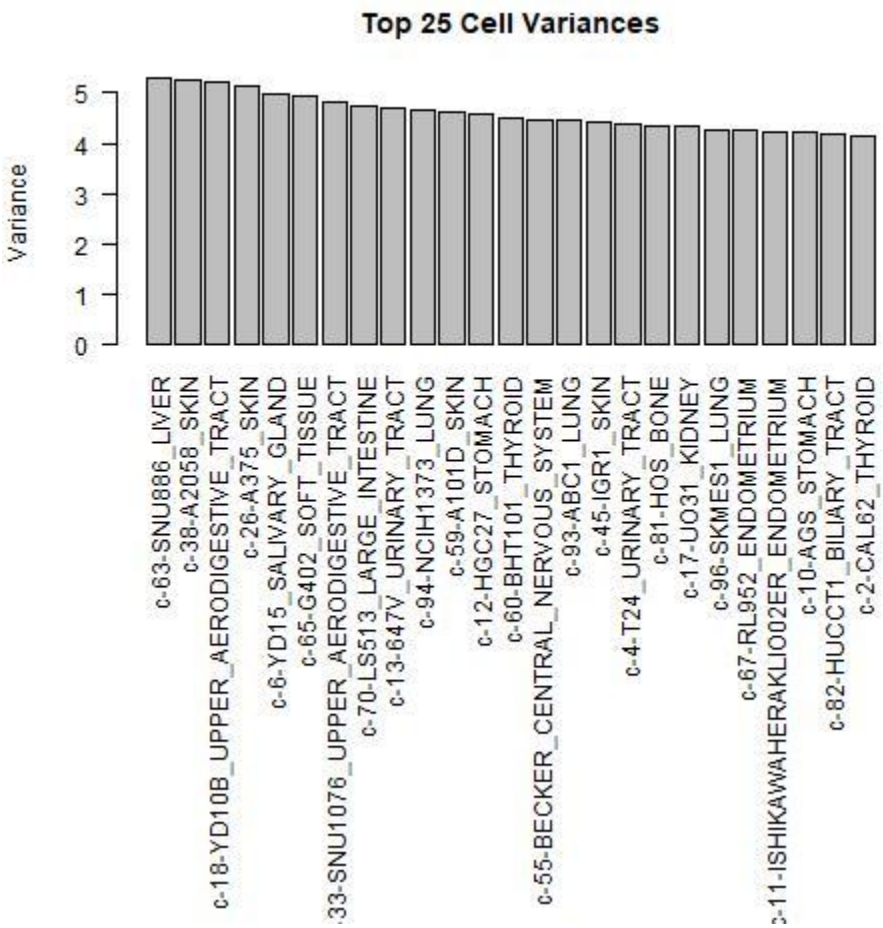
As expected, all three distributions for the bottom three genes with lowest variances have very narrow distributions, centred around zero.

Variances of cell viability

Each cell has a different response to each cell perturbation experiment. A positive value for cell viability indicates that the cell proliferated during the experiment, while a negative value indicates that the concentration of the cell decreased during the same. The magnitude of the value indicates the amount of increase or decrease in proliferation.

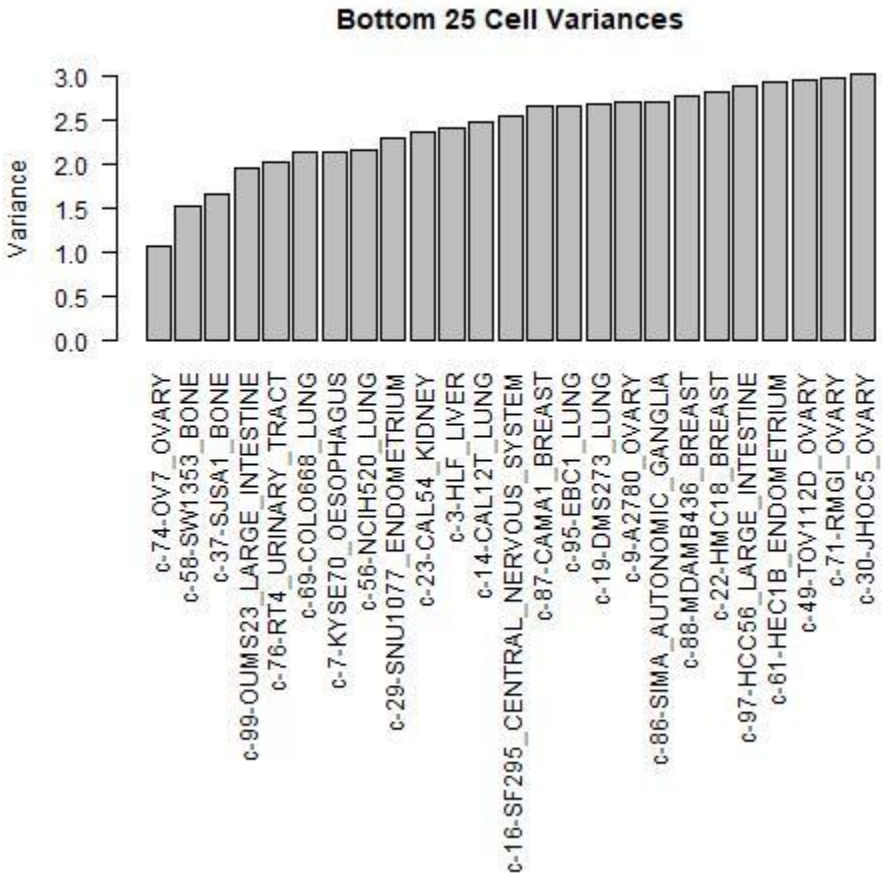
We shall visualize the top 25 and bottom 25 cells having the highest and lowest variance in cell viability respectively.

```
> cellVarianceVector <- colVars(as.matrix(data[779:878]), na.rm =
TRUE)
> names(cellVarianceVector) <- colnames(data[779:878])
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/top25CellVariances.j
peg")
> par(mar=c(20, 4, 4, 2))
> barplot(sort(cellVarianceVector, decreasing = TRUE)[1:25], main
= "Top 25 Cell Variances", ylab = "Variance", las = 2)
> dev.off()
```




```
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/bottom25CellVariance
s.jpeg")
> par(mar=c(20, 4, 4, 2))
> barplot(sort(cellVarianceVector, decreasing = FALSE)[1:25],
main = "Bottom 25 Cell Variances", ylab = "Variance", las = 2)
```

```
> dev.off()
```



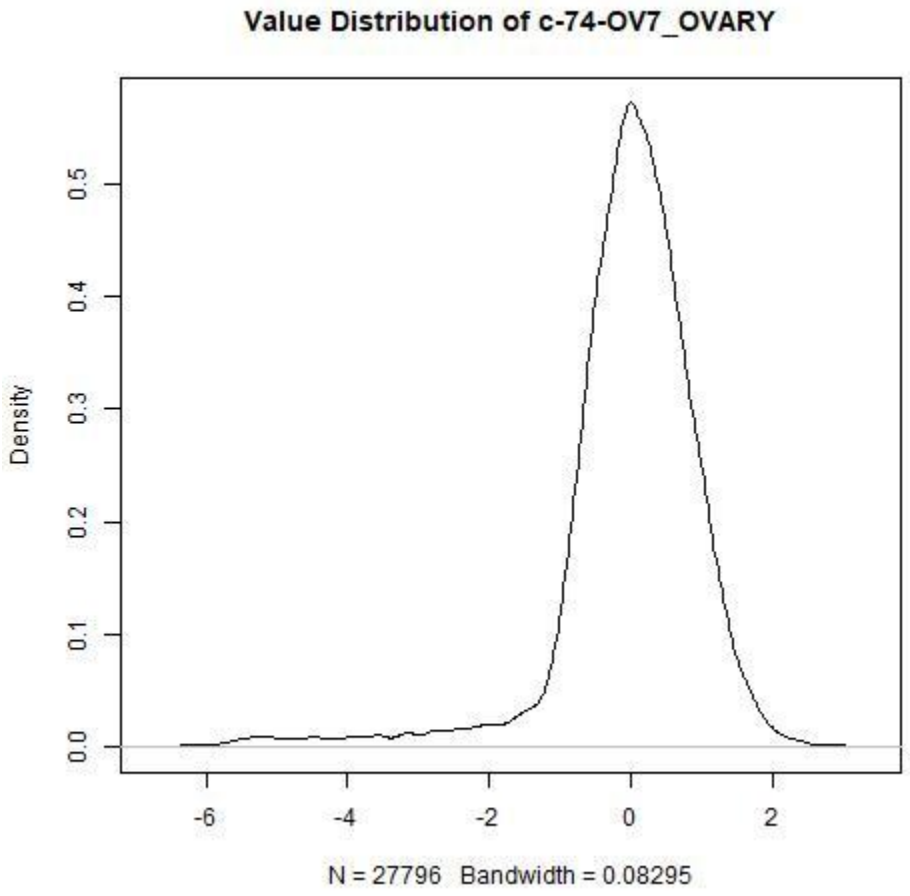
From the above plots, we see that the bottom three cells having lowest variance are c-74-OV7_OVARY, c-58-SW1353_BONE, and c-37-SJSA1_BONE. The top three cells with the highest variance are c-63-SNU886_LIVER, c-38-A2058_SKIN, and c-18-YD10B_UPPER_AERODIGESTIVE_TRACT.

We shall visualize the distribution of values for the cells with bottom 3 and top 3 cell viability variances.

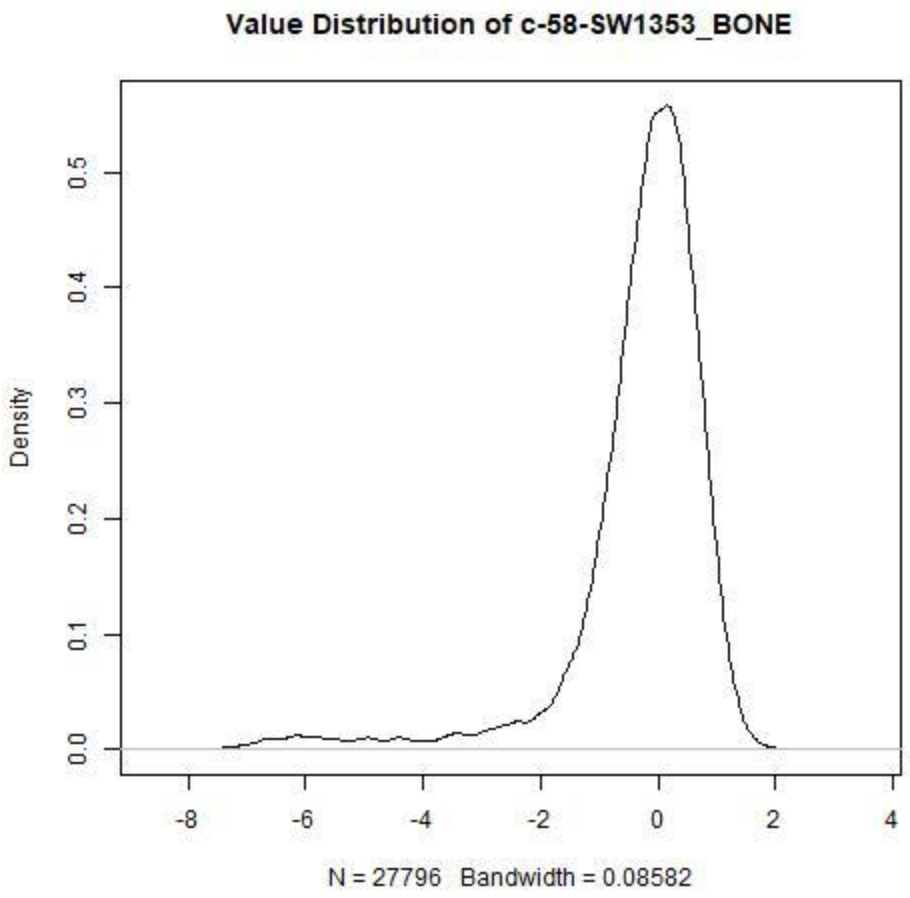
Distribution of values for bottom 3 cells with lowest variances

To view the distribution of values of cell viability for c-74-OV7_OVARY, c-58-SW1353_BONE, and c-37-SJSA1_BONE, we execute the following code:

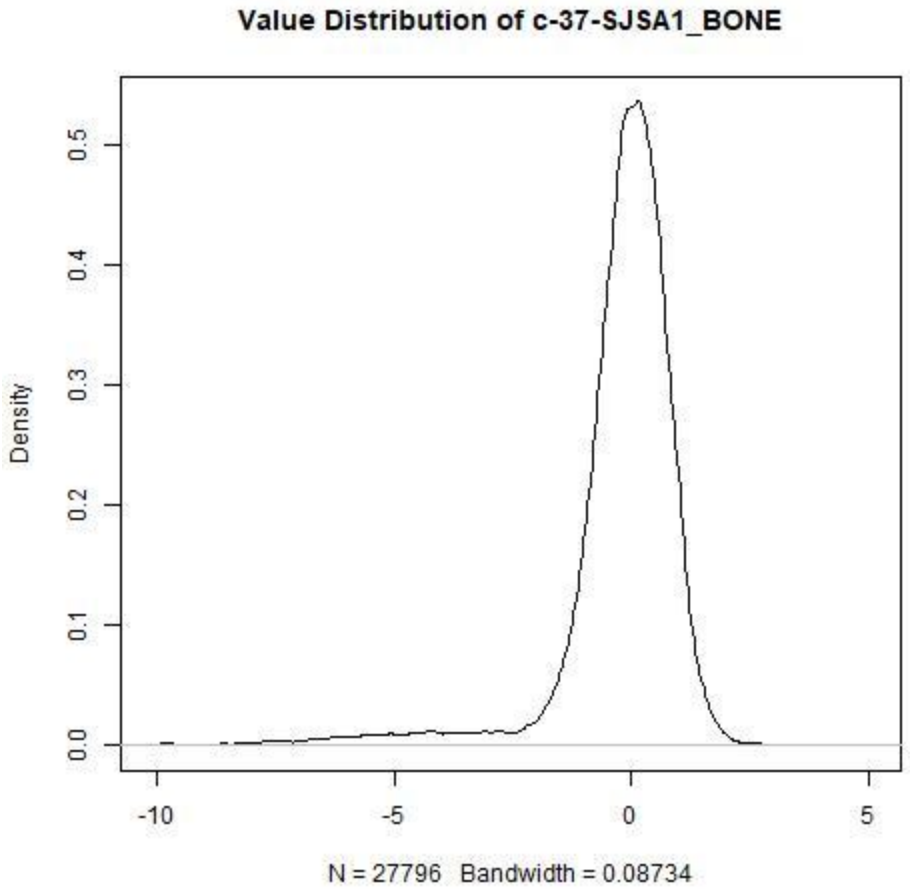
```
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/valueDistributionOfc
-74-OV7_OVARY.jpeg")
> plot(density(data[["c-74-OV7_OVARY"]]), main = "Value
Distribution of c-74-OV7_OVARY")
> dev.off()
```




```
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/valueDistributionOfc
-58-SW1353_BONE.jpeg")
> plot(density(data[["c-58-SW1353_BONE"]]), main = "Value
Distribution of c-58-SW1353_BONE")
> dev.off()
```




```
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/valueDistributionOfc
-37-SJSA1_BONE.jpeg")
> plot(density(data[["c-37-SJSA1_BONE"]]), main = "Value
Distribution of c-37-SJSA1_BONE")
> dev.off()
```



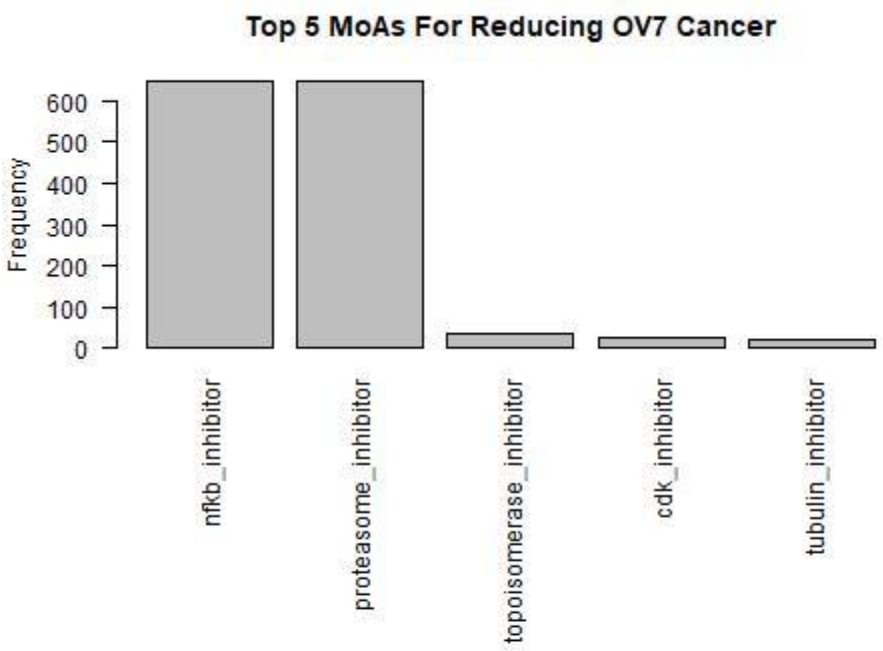
Since these three cells have the lowest variance across the dataset, we can infer that these cells have strong resistance to most anticancer drugs.

Most common MoAs of drugs effective against OV7 cancer

Since OV7 is the cell that has the least variance in cell viability, it is of interest to us to find the mechanisms of actions of drugs that are effective against OV7 cancer. We find the most common MoAs effective against OV7 cancer by executing the following code.

```
> exptsWithReducedOV7 <- data[data$`c-74-OV7_OVARY` <= -2, ]
> moaFreqVector <- colSums(exptsWithReducedOV7[879:1486], na.rm =
TRUE)
```

```
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/top5MoAsForReducingO
V7Cancer.jpeg")
> par(mar=c(20, 4, 4, 2))
> barplot(sort(moaFreqVector, decreasing = TRUE)[1:5], main =
"Top 5 MoAs For Reducing OV7 Cancer", ylab = "Frequency", las =
2)
> dev.off()
```

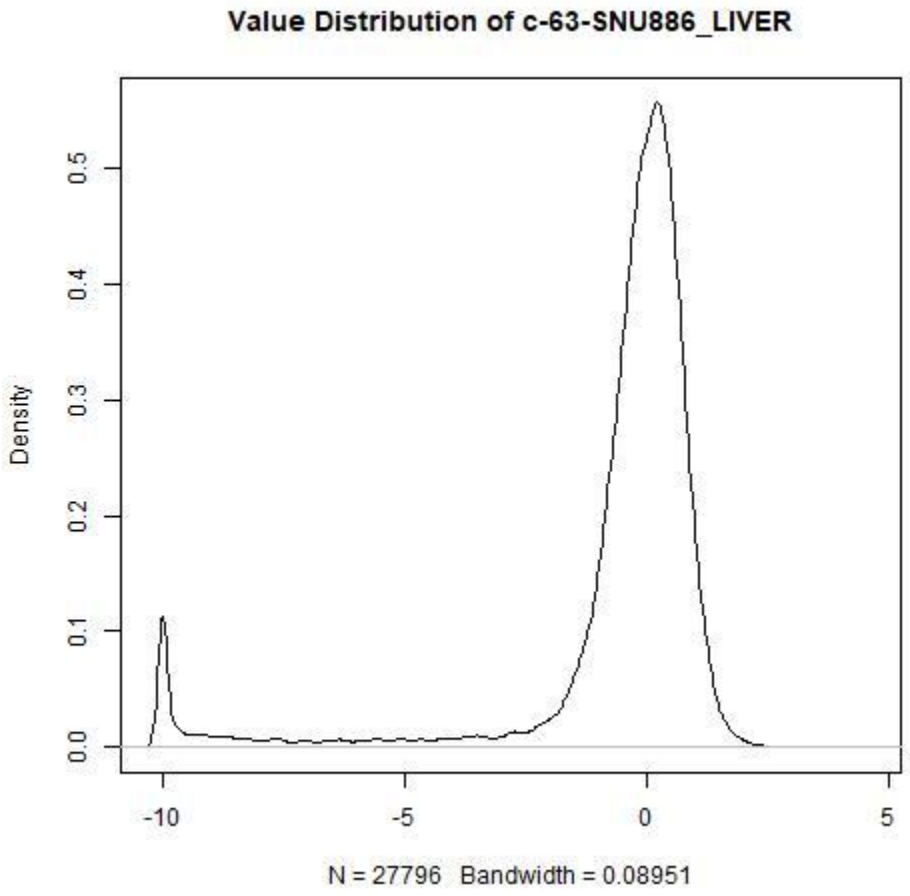


We can immediately see that nfkb_inhibitor and protoeasome_inhibitor are the two most common MoAs for drugs that are effective against OV7 cancer. Thus, drugs that have these two MoAs can be used for the treatment of OV7 cancer.

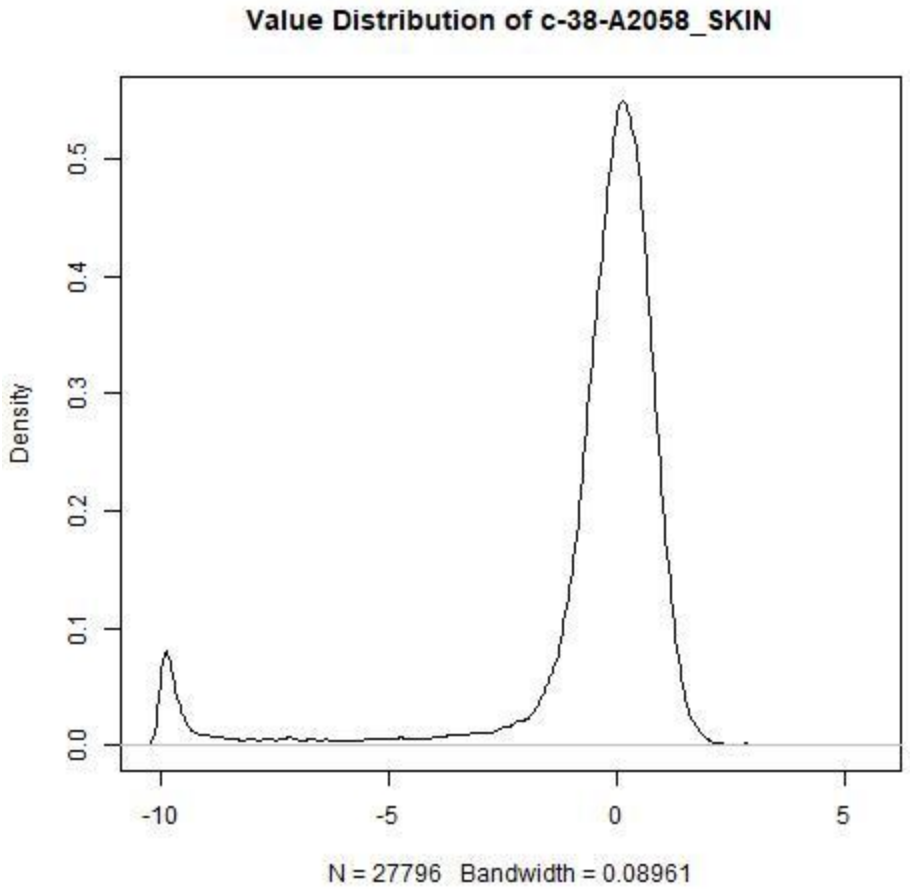
Distribution of values for top 3 cells with highest variances

To view the distribution of cell viability for c-63-SNU886_LIVER, c-38-A2058_SKIN, and c-18-YD10B_UPPER_AERODIGESTIVE_TRACT, we execute the following code:

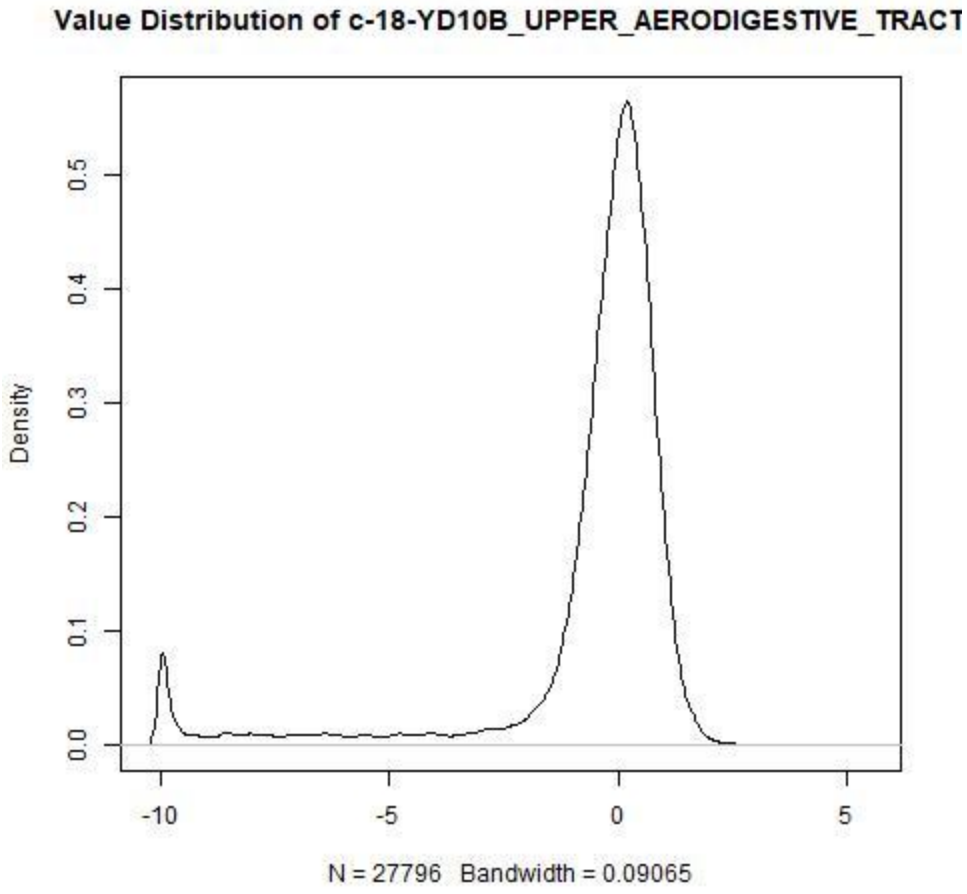
```
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/valueDistributionOfc
-63-SNU886_LIVER.jpeg")
> plot(density(data[["c-63-SNU886_LIVER"]]), main = "Value
Distribution of c-63-SNU886_LIVER")
> dev.off()
```




```
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/valueDistributionOfc
-38-A2058_SKIN.jpeg")
> plot(density(data[["c-38-A2058_SKIN"]]), main = "Value
Distribution of c-38-A2058_SKIN")
> dev.off()
```




```
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/valueDistributionOfc
-18-YD10B_UPPER_AERODIGESTIVE_TRACT.jpeg")
> plot(density(data[["c-18-YD10B_UPPER_AERODIGESTIVE_TRACT"]]),
main = "Value Distribution of
c-18-YD10B_UPPER_AERODIGESTIVE_TRACT")
> dev.off()
```



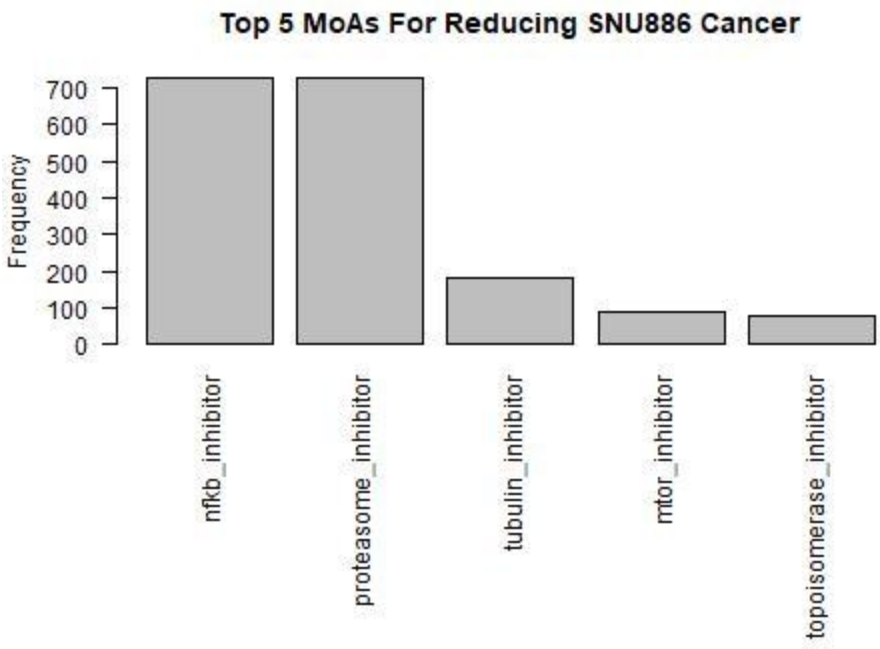
We observe similar plots as for the distribution of values for the top three genes with highest gene expressions. All three plots are centred around 0, and have a bump at -10 due to the lower limit of cell viability set as -10.

Most common MoAs of drugs effective against SNU886 cancer

Since SNU886 is the cell with highest variance in cell viability, we shall note the most common mechanisms of actions of the drugs that are effective against SNU886 cancer, by executing the following code:

```
> exptsWithReducedSNU886 <- data[data$`c-63-SNU886_LIVER` <= -2,
]
> moaFreqVector <- colSums(exptsWithReducedSNU886[879:1486],
na.rm = TRUE)
```

```
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/top5MoAsForReducingS
NU886Cancer.jpeg")
> par(mar=c(20, 4, 4, 2))
> barplot(sort(moaFreqVector, decreasing = TRUE)[1:5], main =
"Top 5 MoAs For Reducing SNU886 Cancer", ylab = "Frequency", las
= 2)
> dev.off()
```



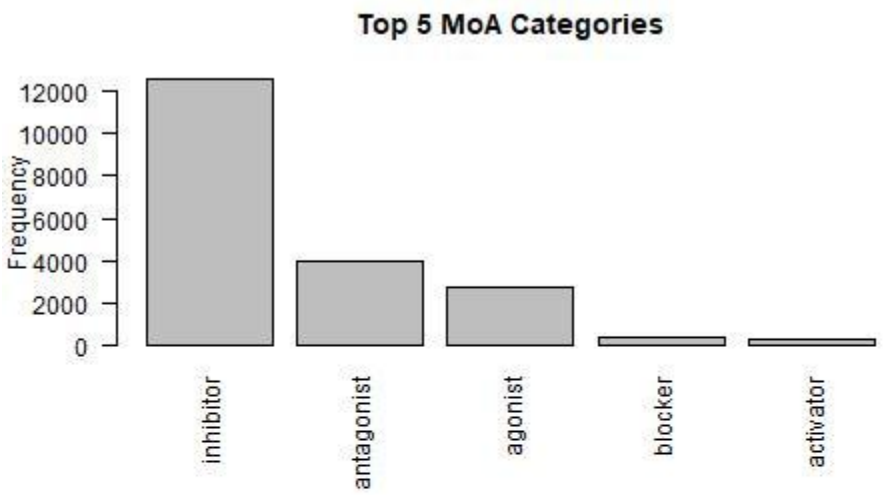
As expected from the plots of top 25 MoAs and top 5 MoAs for reducing OV7 cancer, the same two MoAs, i.e. nfkb_inhibitor and proteasome_inhibitor, are common in drugs that are used for treating SNU886 cancer.

Common MoA Categories

From all three plots that are concerned with common MoAs i.e. top 25 MoAs, top 5 MoAs for reducing OV7 cancer, and top 5 MoAs for reducing SNU886 cancer, we see that nfkb_inhibitor and proteasome_inhibitor are the most common MoAs. Since both MoAs are inhibitors, it is indicative of the possibility of indicators being

the most common category of MoAs. We verify this claim by finding the most common categories of MoAs through the following code:

```
> moaDataFrame <- data[879:1486]
> colnames(moaDataFrame) <- sub("^.*_", "",
colnames(moaDataFrame))
> moaCategoryFreqVector <- colSums(moaDataFrame, na.rm = TRUE)
> moaCategoryFreqVector <- tapply(moaCategoryFreqVector,
names(moaCategoryFreqVector), sum)
>
jpeg(file="C:/Users/Manas/Learn/BDA/Tutorial/top5MoACategories.jp
eg")
> par(mar=c(20, 4, 4, 2))
> barplot(sort(moaCategoryFreqVector, decreasing = TRUE)[1:5],
main = "Top 5 MoA Categories", ylab = "Frequency", las = 2)
> dev.off()
```



Our hypothesis that inhibitors are the most popular category of MoAs is indeed correct. Other common MoAs include antagonist and agonist, as visible from the above plot.

We conclude the exploratory data analysis by noting the system time, and calculating the execution time by finding the difference between the system time and the start time.

```
> executionTime <- Sys.time() - startTime
> print(executionTime)
Time difference of 9.398002 secs
```

Report

Time taken to design the EDA method + write the code: 10-12 hours (spread over 6 days)

Time taken for executing the code: 9.398002 seconds