Tutorial 1a: Exploring Pharmacological Data with the `rawPharmacoData` Dataset

Introduction

Probably the most important step of analyzing datasets is to actually understand the data. This process is crucial to know what kind of questions we can answer with it.

This tutorial has code that will help guiding you through this process with the `rawPharmacoData` dataset.

Make sure you understand the experimental design of the two studies well and try to link each variable to this experimental design. Also, make sure you understand what each *R* command is doing. Feel free to hack the code!

When it makes sense, we include examples for answering the question using both base R and the tidyverse packages. There's usually more than one way of doing things in R!

If you have any question about the code, ask one of the mentors. Also remember that Google search and ChatGPT can aid you in data science tasks.

Setup Workspace

We start by loading the tidyverse family of packages.

```
library(tidyverse)
```

There are several pre-defined themes for plotting with `ggplot2`. While the default "`theme_gray`" is nice, we will set the default to "`theme_bw`" using the `theme_set` function.

```
theme_set(theme_bw())
```

Load Raw Dataset

Let's start by loading the RDS file containing the raw pharmacological data.

```
pharmacoData <- readRDS(file.path("..", "data", "rawPharmacoData.rds"))
```

Exploratory Analysis

We can take a quick peek at the data using the `head` and `str` functions.

- What kind of variables are in the data?
- Are these variables numerical and/or categorical?
- What does each column represent?

```
1 head(pharmacoData)
```

```
  cellLine   drug doseID concentration viability study
1    22RV1 17-AAG doses1        0.0025    94.100  CCLE
2    22RV1 17-AAG doses2        0.0080    86.000  CCLE
3    22RV1 17-AAG doses3        0.0250    99.932  CCLE
4    22RV1 17-AAG doses4        0.0800    85.000  CCLE
5    22RV1 17-AAG doses5        0.2500    62.000  CCLE
6    22RV1 17-AAG doses6        0.8000    29.000  CCLE
```

```
1 str(pharmacoData)
```

```
'data.frame':   43427 obs. of  6 variables:
 $ cellLine     : chr  "22RV1" "22RV1" "22RV1" "22RV1" ...
 $ drug         : chr  "17-AAG" "17-AAG" "17-AAG" "17-AAG" ...
 $ doseID       : chr  "doses1" "doses2" "doses3" "doses4" ...
 $ concentration: num  0.0025 0.008 0.025 0.08 0.25 0.8 2.53 8 0.0025 0.008 ...
 $ viability    : num  94.1 86 99.9 85 62 ...
 $ study        : chr  "CCLE" "CCLE" "CCLE" "CCLE" ...
```

Next, we can count the number of drugs and cell lines in the dataset.

```r
## using base R
length(unique(pharmacoData$cellLine))
```

```
[1] 288
```

```r
length(unique(pharmacoData$drug))
```

```
[1] 15
```

```r
## with the tidyverse
pharmacoData |>
    summarize(nCellLines = n_distinct(cellLine),
              nDrugs     = n_distinct(drug))
```

```
  nCellLines nDrugs
1        288     15
```

Let's also try something a little more complex. We can also count the number of unique drug concentrations **in each study** separately.

```r
## with base R
tapply(pharmacoData$concentration, pharmacoData$study,
       function(x) { length(unique(x)) })
```

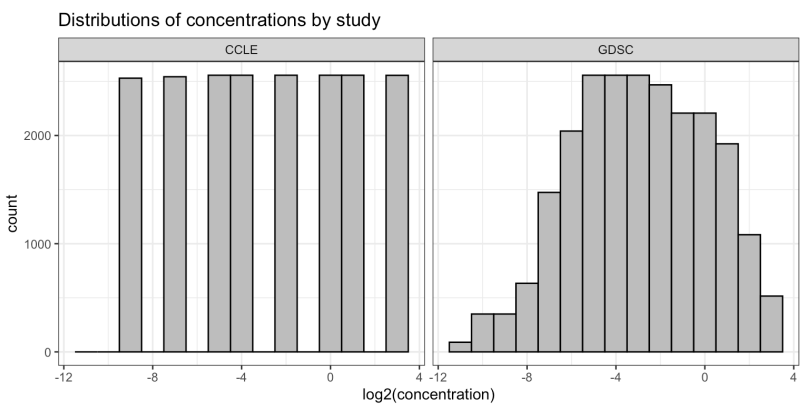
```
CCLE GDSC
   8   32
```

```r
## with the tidyverse
pharmacoData |>
    group_by(study) |>
    summarize(n = n_distinct(concentration))
```

```
# A tibble: 2 × 2
  study     n
  <chr> <int>
1 CCLE      8
2 GDSC     32
```

One of the first things data scientists do when digging into new data is to explore their distributions. Histograms visualize the data distributions and can also point us towards statistical models to use. The code below transforms the concentration values to the logarithmic scale and plots a histogram separately for each study.

```
pharmacoData |>
    ggplot(aes(x = log2(concentration))) +
    geom_histogram(fill = "gray", color = "black", binwidth = 1) +
    facet_wrap(~ study) +
    ggtitle("Distributions of concentrations by study")
```



Based on these plots, which study would you say has the most consistent experimental protocol?

> Place your answer here

Viability scores are the percentage of cells that survive upon exposure to a certain drug. Below, we will explore the range of the data and calculate how many data points are below 0 and above 100.

```r
## with base R
range(pharmacoData$viability)
```

```
[1] -20.0000 319.4919
```

```r
sum(pharmacoData$viability < 0)
```

```
[1] 23
```

```r
sum(pharmacoData$viability > 100)
```

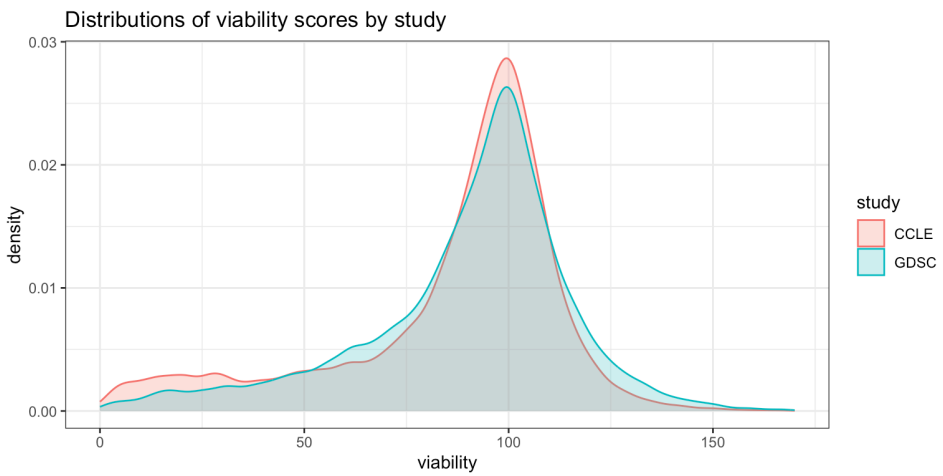
```
[1] 15778
```

```r
## with the tidyverse
pharmacoData |>
    summarize(min_viability = min(viability),
              max_viability = max(viability),
              n_too_small   = sum(viability < 0),
              n_too_big     = sum(viability > 100))
```

```
  min_viability max_viability n_too_small n_too_big
1           -20      319.4919          23     15778
```

We can also compare the distribution of viability scores between the two studies using density plots.

```
pharmacoData |>
    ggplot(aes(x = viability, group = study, fill = study, color = study)) +
    geom_density(alpha = 1/4) +
    xlim(0, 170) +
    ggtitle("Distributions of viability scores by study")
```

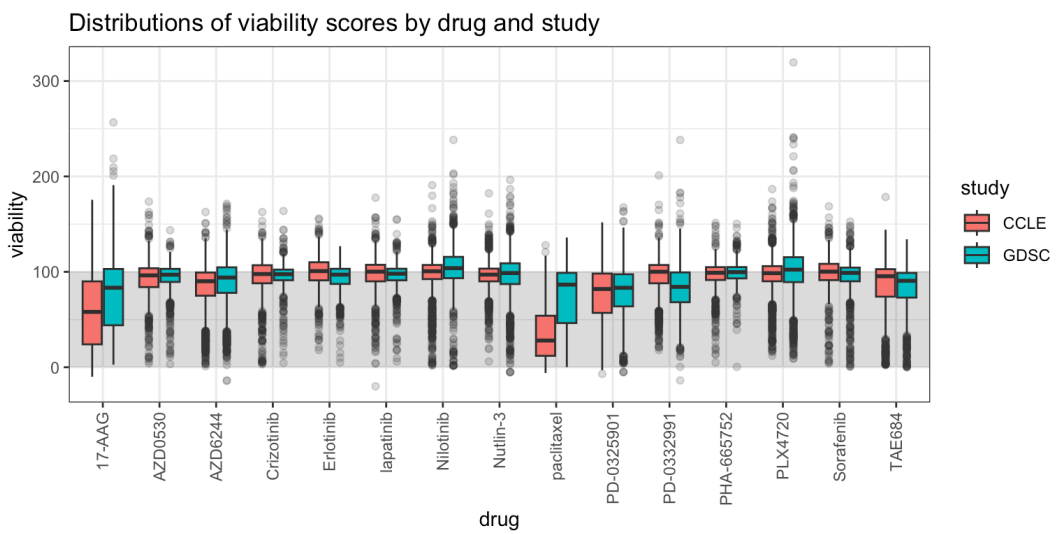


Based on the distribution of the viability scores, would you say there are obvious differences between the two studies?

> Place your answer here

The code below plots the viability scores as box-plots for each drug, stratified by the two studies. We highlight the region of the plot where viability scores should fall (between 0 and 100).

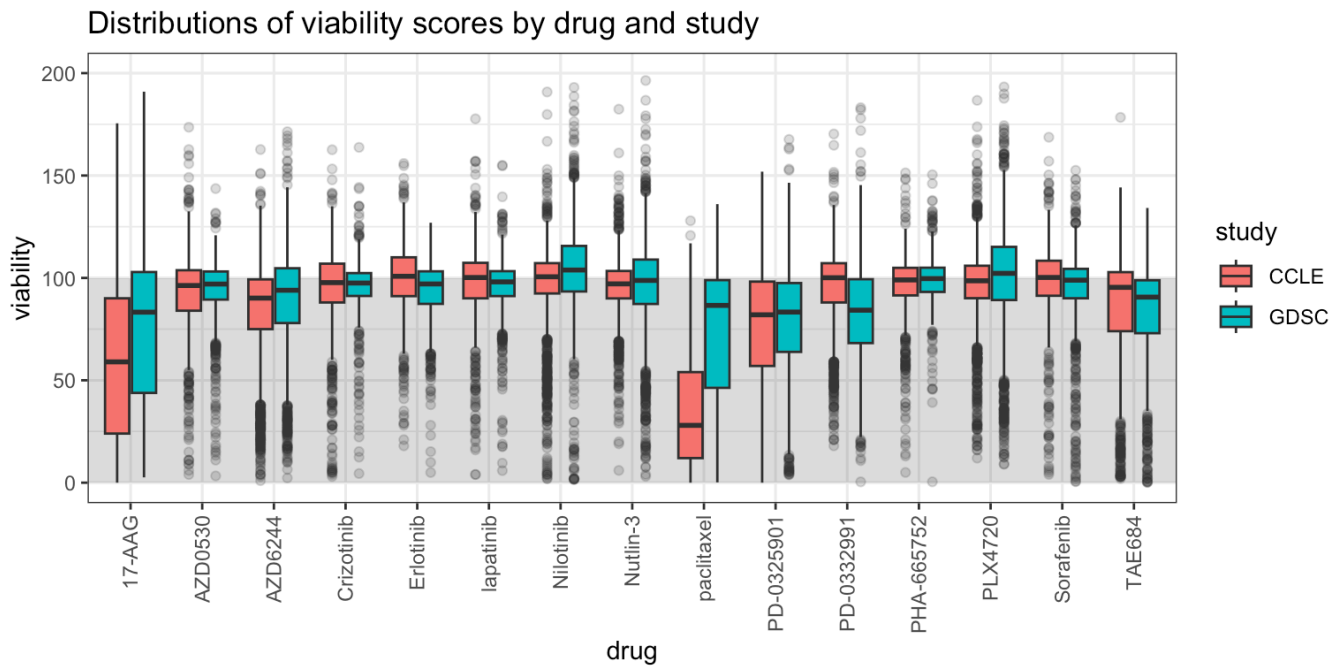
```
gp <- pharmacoData |>
    ggplot(aes(y = viability, x = drug, fill = study)) +
    scale_x_discrete() +
    annotate(geom = "rect", ymin = 0, ymax = 100, xmin = -Inf, xmax = Inf,
             fill = 'black', alpha = 1/6) +
    geom_boxplot(outlier.alpha = 1/5) +
    theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 1/2)) +
    ggtitle("Distributions of viability scores by drug and study")
gp
```



There appear to be a few outliers with incredibly high viability scores!

We should keep this in mind, but to get a better look at the majority of the data, we can limit the y-axis of the plot.

```
1  gp + ylim(0, 200)
```



Can you tell something about the toxic properties of the different drugs? Are these properties consistent across studies?

> Place your answer here

Confirmatory Analysis

So far, we have visually inspected plots of the data to answer scientific questions. This is typically referred to as "*Exploratory* Data Analysis" (EDA).

- This type of analysis is useful for getting a sense of what the data looks like and getting a personal sense of what scientific questions might be worth further investigation.
- However, visual inspection is imprecise; what looks like a large difference to one person might be small to another

Confirmatory analysis allows to quantify whether the differences we might see in a plot are actually "significant" (whether they actually might mean something) or whether they really just result from the randomness of experimentation.

A **statistical hypothesis test** is a procedure to tell us whether our results might be "statistically significant" – meaning, not just due to random experimental error alone.

- To perform a hypothesis test, we first formulate a *null hypothesis*: a condition under which we consider absolutely no effect to have occurred.
- For example, if we want to know whether the two studies in this data differed in terms of viability across drugs and cell lines, we might compare the *mean viability* score.
- In this case, our null hypothesis is that

$$\text{mean viability in CCLE} - \text{mean viability in GDSC} = 0$$

- To test this hypothesis statistically, we choose a test that compares means. The most common is the *t-test*.
- There are other types of hypotheses and ways of testing them too; we won't go into the mathematical details here, but **you can read more about such procedures in the Supplementary Tutorial, Supplement: Statistical Hypothesis Testing**

We can test whether the difference between the mean viability across studies is "statistically significant" using the `t.test` function below:

```
1  t.test(viability ~ study, data = pharmacoData)
```

```
    Welch Two Sample t-test

data:  viability by study
t = -15.77, df = 41956, p-value < 2.2e-16
alternative hypothesis: true difference in means between group CCLE and group GDSC is not equal to 0
95 percent confidence interval:
 -4.698422 -3.659608
sample estimates:
mean in group CCLE mean in group GDSC
          85.90825           90.08727
```

The **p-value** tells us the *probability* of sampling data with a difference in means as large as our own sample had the "true" difference been equal to 0. If the p-value is below some low threshold – commonly 0.05 or 0.01 – then we can say we've "rejected the null hypothesis", meaning that the difference in means is probably *not* just due to randomness.

The p-value of the above is 2.2e-16. Does this indicate that the difference in mean viability between CCLE and GDSC is more than just random experimental error?

> Place your answer here