BSRP Problem Set 2

your-name-here

Introduction

The main goal of this problem set is to implement the Figure 1 analysis from Stephens et al. paper on KRAS mutation and gene expression using CCLE data. The author used The Cancer Genome Atlas (TCGA) data in their analysis, which is a genomics dataset similar to CCLE but profiled using patient's tumor biopsies. Cell lines are the workhorses of cancer research, and it would be interesting to see whether the genomics of cell line data for KRAS is similar to that of TCGA's. You will be working with CCLE's metadata, mutation, and gene expression data.

Biological Question <-> Computational Analysis

Figure 1 answers the following biological question:

> *Is KRAS expression higher in KRAS mutated cell lines compared to KRAS wild-type cell lines? Does this trend hold in lung, pancreas, and colon cancer cell lines?*

It was answered with the following computational analysis:

> *T-test to compare means of two groups, visualized with box plots.*

We will go over the details behind a T-test next week. For now, think of the analysis as comparing the KRAS mean expression for KRAS mutant cell lines against KRAS mean expression for KRAS wild-type cell lines. Instead, we will focus mainly on wrangling multiple datasets so that they are are ready for the computational analysis.

Data Wrangling

Here is the proposed strategy for data wrangling:

> *Our target dataframe is going to have cell lines as observations, and KRAS mutation, KRAS expression, and disease type as variables.*

The **target dataframe** is the dataframe where you can run the computational analysis easily. We will show you exactly how the run the computational analysis once the target dataframe is in the right format.

Exploratory Data Analysis

In the process of data wrangling, you will look at each part of the data carefully to learn what the data is like. After looking at the data, does the data help answer your question, or are adjustments needed? What other hypothesis might come up as you explore the data? This process in data science/statistics is called Exploratory Data Analysis.

A wetlab analogy is to check that you have the samples, reagents, and materials before you run your experiments. As you make these checks and get materials in order, you might find unexpected things: missing items, new kits, etc. that might make you change experiments. We do the same thing in computational work: we think we have the right data to answer our question but we might find something else along the way that make us change plans.

Assignment Formatting

Write your code in the designated code chunks provided, and write your written answers in the text region of the document. If you want to incorporate both text and code output, use the function `cat()` . For example, `cat("The number of rows in the iris dataframe is", nrow(iris))`.

Warming up to the problem

Problem 0.

In your own words, explain the scientific significance of the Biological Question (see introduction)

Solution: The RAS family proteins are a important molecular switch that regulate cell growth and differentiation. When mutated at a hotspot, the switch becomes unregulated, leading to abnormal proliferation and growth. A better understanding of how KRAS mutation affects its expression gives better understanding of how an mutant, oncogenetic KRAS has impact on its molecular pathway. Namely, we don't know if mutation is associated with increased expression.

In your own words, explain why the target dataframe format is useful for the computational analysis.

Solution: We can quickly subset to the disease type of interest, and then use the variables KRAS mutation and KRAS expression to compare differences in gene expression. The observations are in cell lines, which is the what is needed in the analysis.

Loading in the data

Problem 1.

We first load in the needed library and data, and this is done in the code chunk below.

```
## Fetching https://cds.team/taiga/api/dataset/public-21q1-4b39/33
## Status 200
## loading cached data version from  /home/chris/.taiga/public-21q1-4b39_33.toc
```

```
## Fetching https://cds.team/taiga/api/dataset/public-21q1-4b39/33
## Status 200
## loading cached data version from  /home/chris/.taiga/public-21q1-4b39_33.toc
## Fetching https://cds.team/taiga/api/dataset/public-21q1-4b39/33
## Status 200
## loading cached data version from  /home/chris/.taiga/public-21q1-4b39_33.toc
```

First, let's take a look at the number of cell line observations we have for each dataset. This is important to get a sense on how many cell lines can be involved in the analysis. If we make any `join()` on the data, we need to be mindful how we join.

How many mutations are profiled in the mutation data?

How many celllines are profiled in the expression data?

How many genes are profiled in the expression data?

How many celllines have metadata description?

```
## Mutation has 1288288 mutations.
```

```
## expression has 1376 cell lines.
```

```
## expression has 19178 genes.
```

```
## Metdata has 1811 cell lines.
```

Why is it difficult to figure out the number of celllines in the mutation data using the vocabulary observations, variable, and values?

Using the `table()` function on the column "DepMap_ID", figure out the number of cell lines profiled for mutation. Do the same for the number of genes using column "Hugo_Symbol".

Solution: Mutation is difficult because the observations are for individual mutations. Each mutation relates to a gene of a cellline. The corresponding cellline is in one of variables, "DepMap_ID".

```
## Mutation has 1747 cell lines.
```

```
## Mutation has 19541 genes.
```

Looking at KRAS mutations

Let's look at the mutation data to make sure that we are using the appropriate mutations for the analysis.

Problem 2

One important column in the mutation dataset is "Variant_Classification". It classifies mutations on the way it affect amino acid. Take a look at what kind of mutations are profiled: *what kind of mutations would have a functional impact on KRAS protein?*

Solution: Frame shifts, Missense, Nonsense, Nonstop, Start/stop codons, Splice Sites.

Subset the data to contain only missense AND KRAS mutations, as we know that the most well-known gain-of-function mutations for KRAS are missense mutations. Store this in a new dataframe variable.

Take a look at the "Protein_Change" column. *Does the most common codon change reflect the most common codon changes described in the Introduction of Stephens et al. paper? Given your answer, why is it a reasonable answer?*

```
##
## p.A130T p.A146P p.A146T p.A146V  p.A18D  p.A59G  p.A59T p.D119N  p.D33E  p.G12A
##       1       3       7       3       1       1       3       1       1      16
##  p.G12C  p.G12D  p.G12F  p.G12R  p.G12S  p.G12V p.G138V  p.G13C  p.G13D p.I171M
##      27      70       1       8       7      46       1       6      15       1
## p.I187V p.K117N  p.L19F  p.L23R p.P110H p.P121H p.P140H  p.Q61H  p.Q61K  p.Q61L
##       2       3       1       1       1       1       1       9       2       2
##  p.Q61R  p.T20A  p.T58I  p.T74P  p.V14I  p.V14L p.V160A   p.V9I
##       2       1       1       1       2       1       1       1
```

Solution: Yes, the top mutations correspond to codons 12, 13, and 61. Cell lines should reflect genetically to the primary tumor sample, so we would hopefully expect the top mutations described in TCGA match with that of CCLE's.

Problem 3

Transform the mutation dataframe so that the observations are cell lines, and the variable (call it `N_KRAS_mutation`) is the number of KRAS mutation observed per cell line. There can be more than one mutation per cell line.

Examine the number of cell lines in the dataframe you just transformed. What happened? Is this expected? How might this affect downstream analysis?

Solution: When we subsetted the dataset to KRAS missense mutations, we also subset the number of cell lines. We are left with 240 cell lines. The variable we created is the number of KRAS mutations, not whether a cell line has a KRAS mutation. When we join this the metadata, we want to make sure we use join_all.

Looking at KRAS expression

Problem 4

Locate KRAS gene expression in the expression dataframe, and summarize the minimum, maximum, mean, and medium of KRAS expression.

```
## Min: 1.070389
```

```
## Max: 10.8844
```

```
## Mean: 4.018991
```

```
## Median: 3.947666
```

The unit of gene expression is log_2 transformed. That means each increased unit means double amount of mRNA transcript observed, and you can see it on the y-axis of Figure 1 in the paper. In most RNA analysis, the data is log_2 transformed, for statistical properties.

To get a sense of the original scale, *transform the expression back to its raw mRNA counts, and recalculate the min, max, mean, and median.* This is the mRNA count, normalized to gene length and sequencing depth.

```
## Min: 2.1
## Max: 1890.3

## Mean: 23.43565

## Median: 15.43
```

Integrating metadata, mutation, and expression data together

Problem 5

Using the `*_join()` functions carefully, create a dataframe with the following:

- Observations: cell lines (all of them from metadata)
- Variables: all variables from metadata dataframe, KRAS expression, and number of KRAS missense mutations.

Hint 1: Work with the dataframe you created in Problem 3.

Hint 2: It may be helpful to subset the expression data to something smaller before you use `*_join()`

Then, create a new variable "KRAS_mutated" that has the value True if the number of KRAS missense is not `NA` and False if it is `NA`.

Hint: Use `is.na()` function on `N_KRAS_mutation` column.

Perform some checks yourself to make sure that what you did makes sense. Are there missing NA values that should be flagged?

```
##
## FALSE  TRUE
##  1375   436
```

Solution: There are 500 NAs in the KRAS dataset. This will limit our analysis sample size.

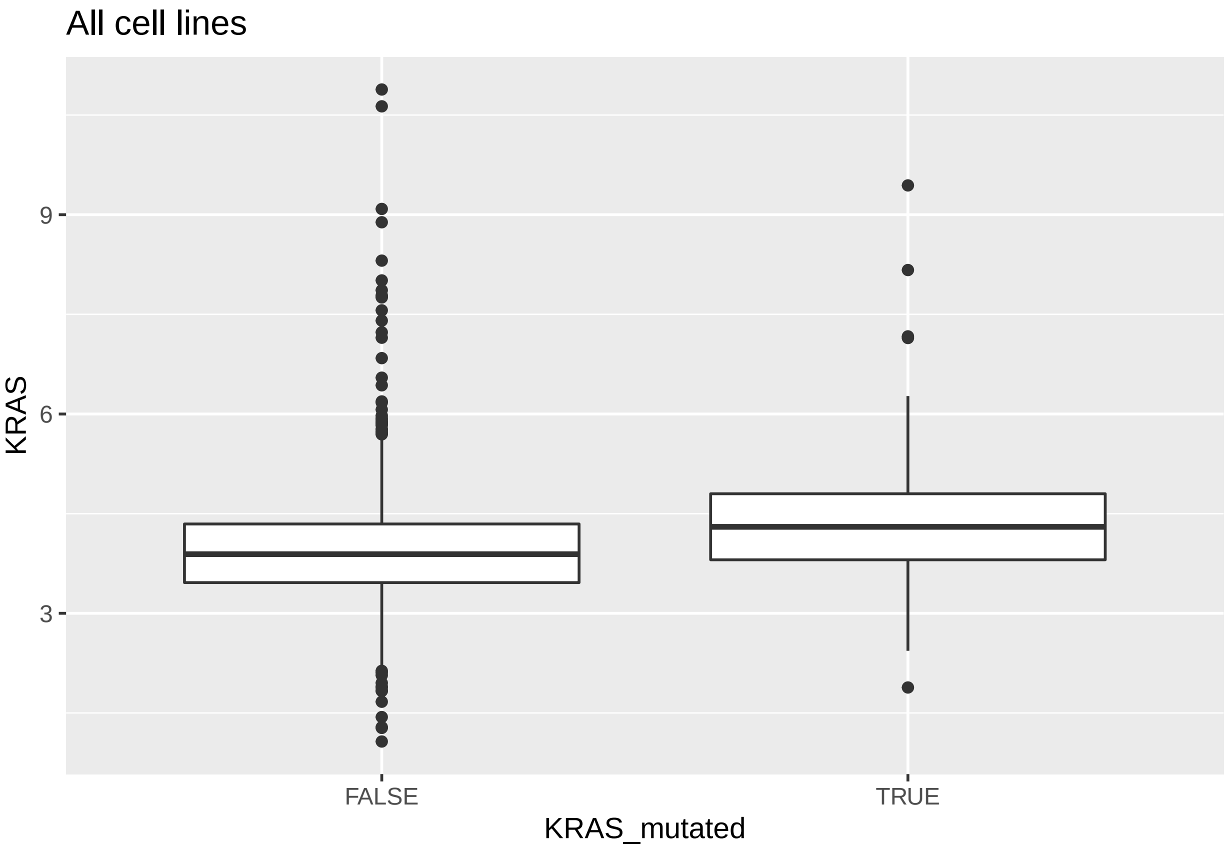
Recreating Figure 1

Problem 6

Make a boxplot of KRAS expression by whether the cell line has KRAS mutation or not.

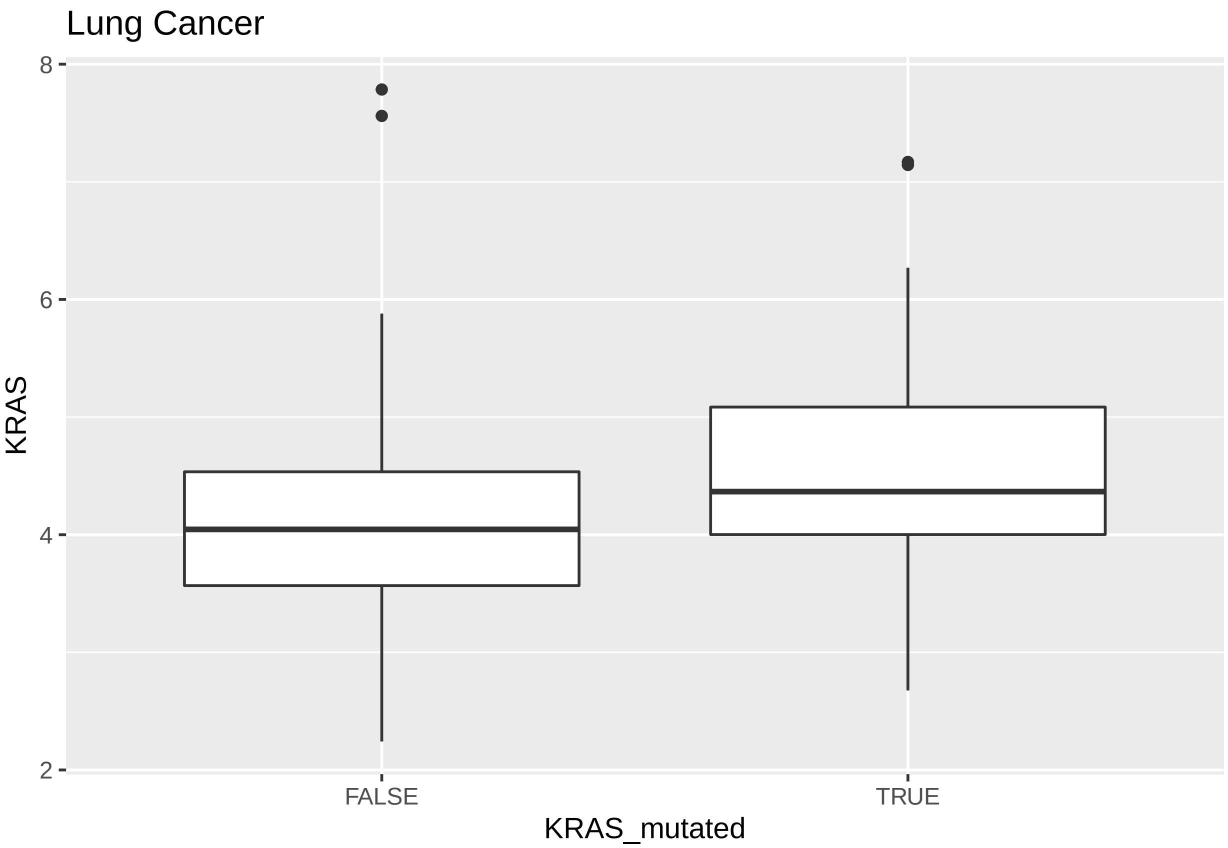
The boxplot is created using ggplot, a plotting function we will go in depth next week. For now, use the commented starter code to put in your dataframe from Problem 5 and the data variables associated with the dataframe to make the plot.

```
## Warning: Removed 436 rows containing non-finite values (stat_boxplot).
```

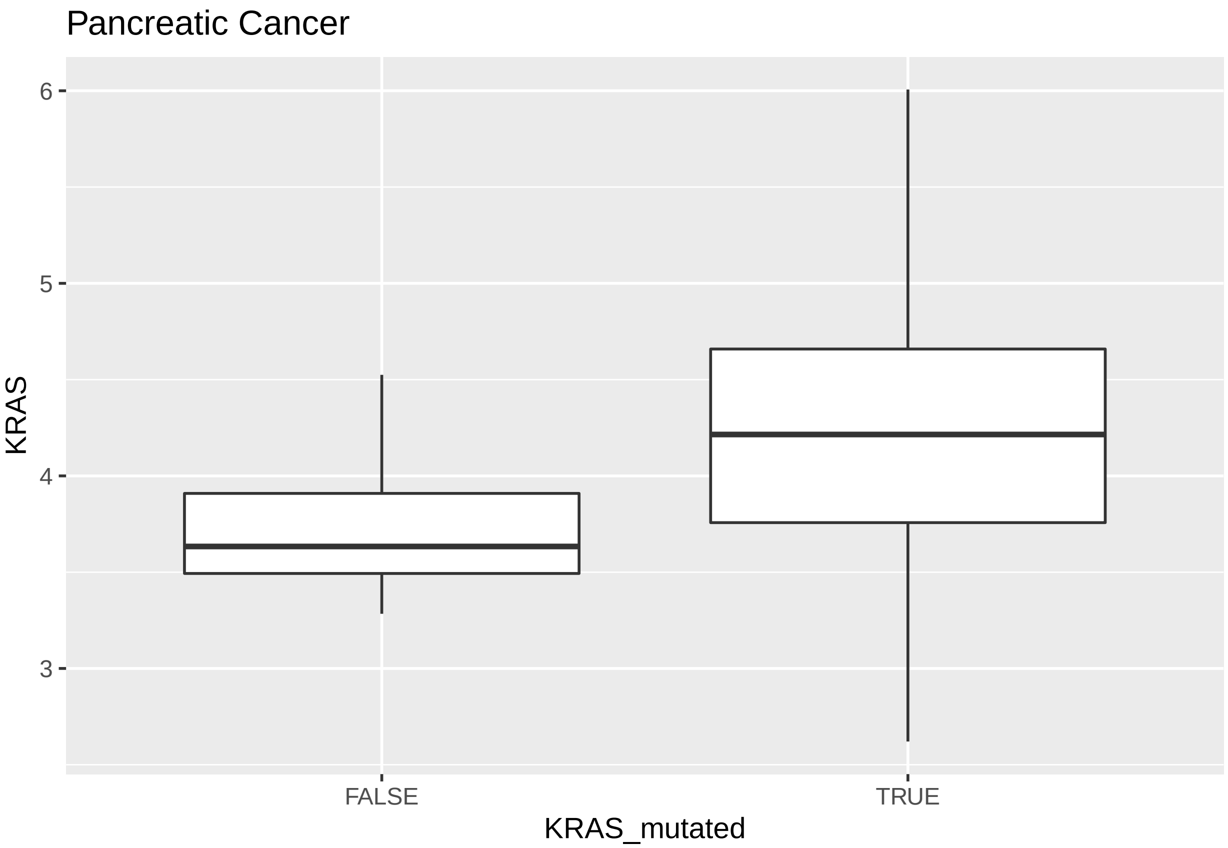


Then, subset to lung, pancreas, and colon cancers and make three more boxplots. How do you interpret the boxplot? How do they compare to all cancer cell lines and Figure 1 of Stephens et al. paper? What could be some reasons behind the difference in the comparison?

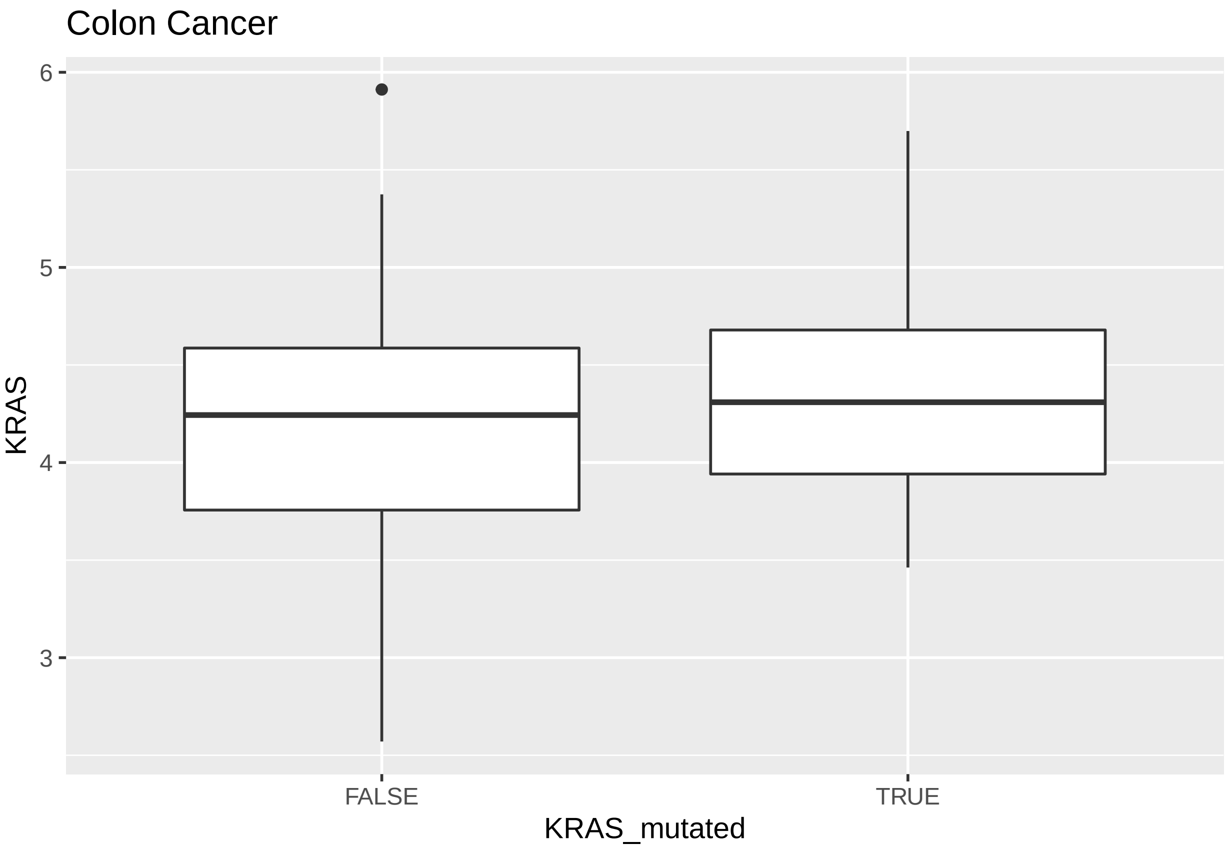
```
## Warning: Removed 67 rows containing non-finite values (stat_boxplot).
```



```
## Warning: Removed 7 rows containing non-finite values (stat_boxplot).
```



```
## Warning: Removed 12 rows containing non-finite values (stat_boxplot).
```



Solution: The boxplot shows the range, mean, Q1, and Q3 of gene expression data. All of the cell lines, and specific subsets have a similar trend as the paper - that mutation is associated with higher gene expression There seems to be barely a difference in the colon cancer in CCLE compared to a significant difference in TCGA.

The range and difference between WT/Mutant is different in the two analysis. CCLE analysis has log2 ratios of 4-6, while the paper's TCGA analysis has a range of 10-12.

All of these differences could be due to different computational pipelines, difference between patient and cell line, and sample size differences.

Problem 7

We know that association is far from causation. *Using ideas from the paper and/or your ideas, what are some other factors that might explain gene expression changes between KRAS wildtype and mutant?*

Solution: Copy number changes has a strong relationship with gene expression. Epigenetic or gene regulatory relationships, such as enhancers, silencers, methylation, histone modifications, may also play a role. It is also possible that the majority of the KRAS transcript are wildtype encoding, not mutant!