Data Wrangling Final Project: Disease Divas

Afsha Tabasum, Jessica Serrao, Parinitha Kompala, Alisa Tvorun Dunn, Ayush Chakraborty

11/16/2021

Introduction

Prostate cancer is among one of the leading causes of death in males in the United States. The genomic study of metastatic prostate cancer has been limited due to tumor samples being obtained from autopsies as well as limited cohort pre-clinical models. Prospective studies are few and far between due to limitations in collecting adequate tumor tissue through biopsy. A paper published in 2015 prospectively analyzed tumor biopsies among affected individuals. Subsequently as of 2016, genomic data for prostate cancer has become more widely available due to the National Cancer Institutes (NCI) Genomics Data Commons (GDC).

In this project we will explore data from both the 2015 prospective cohort study along with data available from the NCI's Genomic Data Commons. This project aims to investigate three data sets, through mapping and transformation of the raw data through multiple platforms such as Excel, SQL and R, with the ultimate goal of optimizing downstream analysis of the prostate cancer cohorts. We will discuss various methods by which we wrangled the data to investigate underlying associations between individual case presentations and manifestation of disease.

A team of 73 researchers conducted a multi-institutional study that investigated clinical genomics of advanced prostate cancer specifically metastatic, castration resistant prostate cancer (mCRPC). mCRPC occurs in individuals who develop resistance to androgen deprivation therapies (ADT). The aim of this research was to facilitate precision medicine by developing a prospective whole-exome and transcriptome sequencing of tumor biopsies. Data was obtained from living affected individuals to compile a set of genomic alterations. The study was published in the journal "Cell" in May 2015.

The link to the publication can be found here.

Data was extracted from the cBioPortal, a public cancer genomics site hosted and mainted by the Memorial Sloan Kettering Cancer Center. The sample included 150 individuals with metastatic prostate cancer along with 17 data points per individual, described below:

- **Patient ID**: unique patient identifier (number)
- **Sample ID**: unique biopsy sample identifier from clinical trial(number)
- **Mutation Count**: mutation rate for mCRPC defined as mutations per megabase or mutations/Mb (number)
- **Fraction Genome Altered**: fraction of mutant allele variants in biopsy (number)
- **Diagnosis Age**: patient's age when they were diagnosed (number)
- **Tumor Site**: physical site of tumor on human body from which biopsy was taken (string)
- **Abiraterzone (ABI) and Enzalutamide (ENZA) Exposure Status**: exposure to second-generation androgen deprivation therapies (binary: yes/no)
- **Center of Sequencing**: the international academic medical center where exome sequencing occurred (string)
- **Prior Treatment**: receipt of previous androgen deprivation therapy (binary:yes/no)
- **Site Sequenced**: site of exome sequencing: Broad or UM (string)
- **Tumor content**: percentage of tumor captured in the biopsy (number)

The National Cancer Institute (NCI) maintains a unified repository of genomic data known as the Genomic Data Commons(GDC). Groundwork for this repository began in June 2015 with nearly 50,000 raw data sequences analyzed by June 2016. This data sharing platform uses an Open-Stack-based private cloud to unite several large-scale cancer genome research programs such as TCGA and OCG. The purpose of this repository is to standardize the data processing of these multi-dimensional data sets by using common bioinformatics pipelines that facilitate direct comparison of the data.

More about the National Cancer Institute's Genomic Data Commons can be found here.

Data was extracted from GDC Portal's Exploration page. The "cleaned" version of the data sample included 493 individuals with metastatic prostrate cancer along with 24 data points per individual as described below:

- **CaseUUID**: universally unique identifier assigned to every entity in the data model (string)
- **CaseID**:specific individual cases using the submitter ID from the genomic cancer research program (string)
- **Project**:The cancer research program from which the case originated from (string)
- **Primary Site**: The primary site of the cancer (string)
- **Gender**:"Gender is defined as a set of characteristics that distinguish persons based on their social roles" (string)
- **Files**: the quantity of files available for that case (number)
- **Seq**:Number of the sequencing reads(an inferred sequence of basepairs) (number)
- **Exp**:The transcriptome profiling (number)
- **SNV**:simple nucleotide variation (number)
- **CNV**:Copy number variation,the number of copies of a particular gene varies from one individual to the next (number)
- **Meth**:Number of DNA methylation that took place (number)
- **Clinical**: The number of clinical alterations (number)
- **Bio**:biospecimen (number)
- **Mutations**:The number of simple somatic mutations detected for that case(number)
- **Genes**: The quantity of genes affected by mutations for each case (number)
- **Slides**: the quantity of slides available for each case (number)
- **Program**: The type of program the subject is enrolled in "The Cancer Genome Atlas" or the "Count Me In" (string)
- **Disease Type**: This describes if it is Adenomas and Adenocarcinomas, the location where the cancer is formed (string)
- **Age at diagnosis**:Age at the time of diagnosis calculated from their day of birth to the day they were diagnosed with cancer (string)
- **Days to death**:The date from when they were diagnosed, and they died (string)
- **Vital Status**: the survival status of the person (string)
- **Primary Diagnosis**:Their initial pathological diagnoses of cancer (string)
- **Ethnicity**: "An individual's self-identified social and cultural category, particularly whether they identify as Hispanic or Latino" (string)
- **Race**: "A taxonomic group that is a division of a species that is classified arbitrarily. It is characterized by shared heredity, physical attributes, and behavior, and in the case of humans, by common history, nationality, or geographic distribution" (string)

Using patient data from the 2015 prospective cohort study, we centered our project around three main aims:

First, we wanted to look at possible associations between geographic location and key characteristics of prostate cancer. More specifically, our analysis revolved around looking at links between patients' sequencing site location and severity of tumor content, along with whether sequencing site region had any link to prior treatment status and exposure to second-generation androgen deprivation therapies. Through this aim, we sought to answer the question "Does geographic region affect severity of prostate cancer characteristics?" We specifically chose to focus on geographic region for this aim so we could draw conclusions on if patient location had any clinical implications when considering future treatments for prostate cancer, and answer questions such as “Would it be beneficial to allocate more resources to certain regions of the country?” or “Are men in certain regions of the country who have had exposure to second-generation androgen deprivation therapies at more risk for developing prostate cancer?”

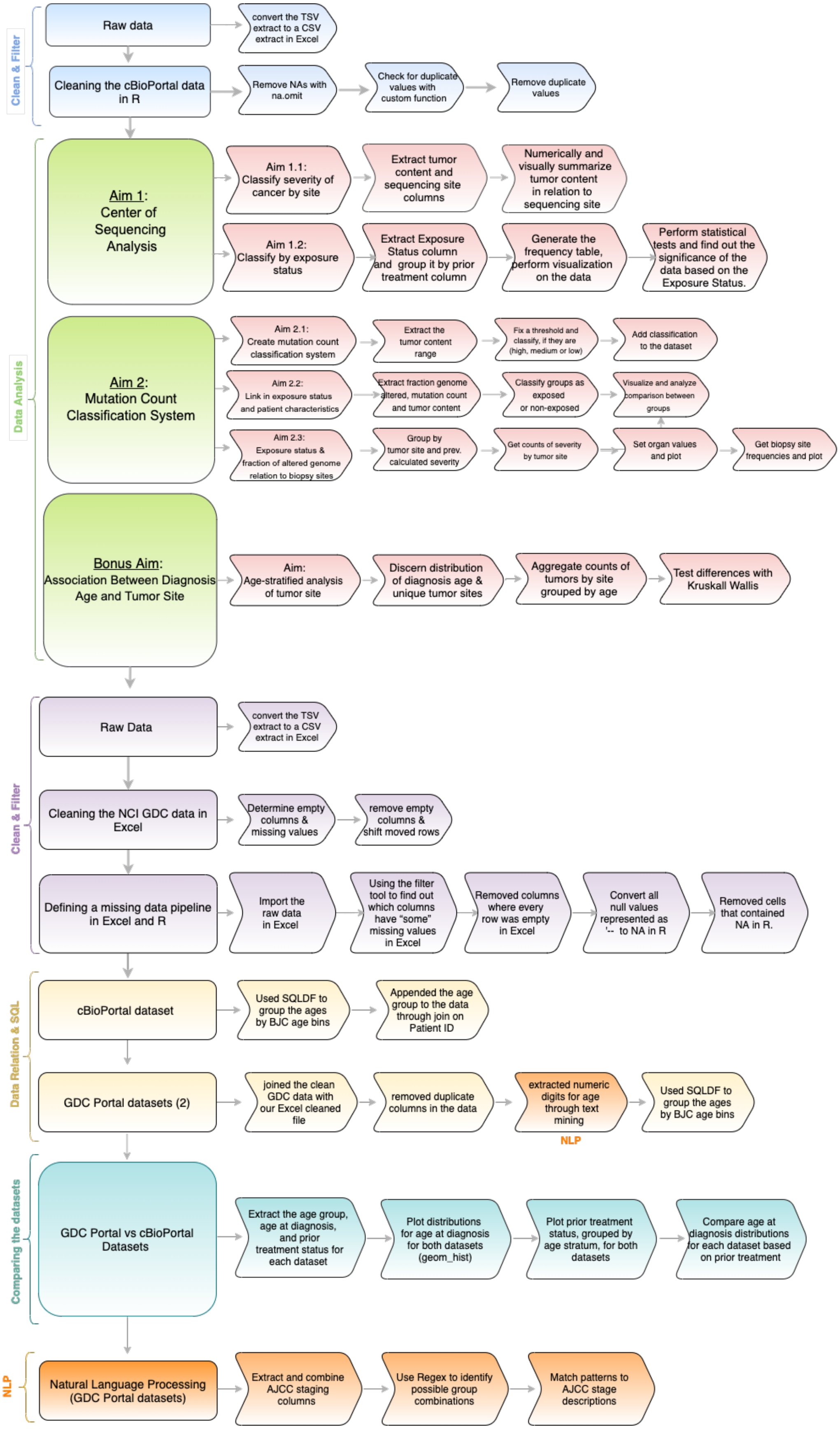
Our second aim focused on patient mutation counts and their relationship to biopsy sites, exposure to second-generation androgen deprivation therapies, and fraction of patient’s genome that was altered. According to a 2017 publication from Cell Martincorena, et al. , cancer will usually emerge in a patient after 1-10 mutations. Thus, we created a classification system to categorize mutation level as “high”, “medium”, or “low” based on the mutation count variable from the data set, and compare severity of mutation to other patient characteristics. Since the 2015 prospective cohort study was specifically interested in patients who developed metastatic, castration resistant prostate cancer (mCRPC), which occurs in individuals who develop resistance to androgen deprivation therapies (ADT), our second aim sought to look at correlations between mutation count and exposure to second-generation androgen deprivation therapies, along with fraction of patient’s genome altered and biopsy sites.

Our last aim, titled the “Bonus Aim” as we hadn’t originally planned to perform this analysis, studied possible associations between patient age at diagnosis and tumor site. We wanted to study this specific association to see if we could identify if earlier age at diagnosis corresponds to specific tumor site (likewise for older ages of diagnosis), since finding such an association could have clinical implications for when considering treatments for different age groups. To study this association, we performed an age stratified analysis based on age groups that were used by a population based study published in the British Journal of Cancer in 2013 read here.

We also wanted to utilize the lesson on Natural Language Processing from class in our data sets, so we applied the principles of NLP to the second data set in order to better understand information that was given about cancer stages for each patient. In order to do this, we used Regex and information on cancer stages descriptions found here.

PIPELINE

Below is the pipeline for our project. It covers our cleaning process for both data sets, our process for working with relational data (with GDC data), how we performed Natural Language Processing on the GDC portal data sets, and what we did for each of our aims. We will be covering each of these points in detail over the course of this paper.


Clean & Filter
Raw data
convert the TSV extract to a CSV extract in Excel
Cleaning the cBioPortal data in R
Remove NAs with na.omit
Check for duplicate values with custom function
Remove duplicate values
Data Analysis
Aim 1: Center of Sequencing Analysis
Aim 1.1: Classify severity of cancer by site
Extract tumor content and sequencing site columns
Numerically and visually summarize tumor content in relation to sequencing site
Aim 1.2: Classify by exposure status
Extract Exposure Status column and group it by prior treatment column
Generate the frequency table, perform visualization on the data
Perform statistical tests and find out the significance of the data based on the Exposure Status.
Aim 2: Mutation Count Classification System
Aim 2.1: Create mutation count classification system
Extract the tumor content range
Fix a threshold and classify, if they are (high, medium or low)
Add classification to the dataset
Aim 2.2: Link in exposure status and patient characteristics
Extract fraction genome altered, mutation count and tumor content
Classify groups as exposed or non-exposed
Visualize and analyze comparison between groups
Aim 2.3: Exposure status & fraction of altered genome relation to biopsy sites
Group by tumor site and prev. calculated severity
Get counts of severity by tumor site
Set organ values and plot
Get biopsy site frequencies and plot
Bonus Aim: Association Between Diagnosis Age and Tumor Site
Aim: Age-stratified analysis of tumor site
Discern distribution of diagnosis age & unique tumor sites
Aggregate counts of tumors by site grouped by age
Test differences with Kruskall Wallis
Clean & Filter
Raw Data
convert the TSV extract to a CSV extract in Excel
Cleaning the NCI GDC data in Excel
Determine empty columns & missing values
remove empty columns & shift moved rows
Defining a missing data pipeline in Excel and R
Import the raw data in Excel
Using the filter tool to find out which columns have "some" missing values in Excel
Removed columns where every row was empty in Excel
Convert all null values represented as '-- to NA in R
Removed cells that contained NA in R.
Data Relation & SQL
cBioPortal dataset
Used SQLDF to group the ages by BJC age bins
Appended the age group to the data through join on Patient ID
GDC Portal datasets (2)
joined the clean GDC data with our Excel cleaned file
removed duplicate columns in the data
extracted numeric digits for age through text mining
NLP
Used SQLDF to group the ages by BJC age bins
Comparing the datasets
GDC Portal vs cBioPortal Datasets
Extract the age group, age at diagnosis, and prior treatment status for each dataset
Plot distributions for age at diagnosis for both datasets (geom_hist)
Plot prior treatment status, grouped by age stratum, for both datasets
Compare age at diagnosis distributions for each dataset based on prior treatment
NLP
Natural Language Processing (GDC Portal datasets)
Extract and combine AJCC staging columns
Use Regex to identify possible group combinations
Match patterns to AJCC stage descriptions

Methods

GITHUB

We used Github, a version-control software readily available on the web, to collaborate with each other throughout the project and efficiently add our individual contributions to our final paper. We created a repository titled “Data Wrangling Project”, which included our final Rmd file (this paper), a copy of our project pipeline image, our custom library package, and all relevant data sets throughout our analysis. We followed a standard protocol for adding and updating changes to our files – we used the git pull command for updating our files and we used the add, commit, and push commands for adding our individual changes to the repository. Utilizing Github allowed all of our group members to work on our individual aims at our own time and pace while keeping everyone else updated on our individual and overall progress. A link to our finals repository can be found here.

CUSTOM FUNCTIONS

Below are our custom functions that will be used throughout our data cleaning and joining processes.

Our first function, `string_NA`, checks for values in a data frame that are equal to the string " ’– “, and changes the value to”NA".

Our second function, `carbon.copy`, checks for columns that have the exact same values in a data frame. If two columns have the exact same values (case insensitive), the function will remove one of those columns.

Our third and final function `find_nonuniq`, prints out columns in a given data frame that do not contain a single unique value.

Our functions are included below:

```
library(DiseaseDivas)

## Convert to NAs function
string_NA <- function(data){
  data[data == "'--"] <- NA
  return(data)
}

# Remove a column if two columns are the exact same
carbon.copy <- function(df){
  return(df[!duplicated(as.list(df))])
}

# Print out all columns that have the same value for all rows
find_nonuniq <- function(dataset){
  cols_nonunique <- c()
  for(i in 1:ncol(dataset)){
    if(length(unique(dataset[,i])) <= 1){
      #append the vector with the new columns
      cols_nonunique[length(cols_nonunique) + 1] <- i
    }
  }
  print(cols_nonunique) #print the vector
}
```

We also compiled our functions into a simple library, titled `DiseaseDivas`, which is available on our GitHub repository. We followed methods outlined by Dr. Joshua Levy from class on Thursday, October 14th, 2021. Link to the lecture can be found here.

Below is an example of how to use our library:

```
rawdat<- read.csv("~/Desktop/Data-Wrangling-Project/diseasedivadata.csv")
dim(rawdat) # Check raw data dimensions
```

```
## [1] 150  20
```

```
# Use library to get rid of duplicate functions
example_dat = DiseaseDivas::carbon.copy(rawdat)
dim(example_dat) # Check new non-duplicate data frame dimensions
```

```
## [1] 150  18
```

Extracting the Data from the cBioPortal

The first data set was downloaded from the cBioPortal site using the "download clinical data for the selected cases" button extracting all columns.

The link to the data set can be found here.

Cleaning Our Data

The first data set was downloaded as a TSV file which required conversion in Excel to a CSV file. After importing the data into Excel using the Import Data From Text tool, the worksheet was saved as a csv file. The raw data contained 150 rows and 20 columns. As a team we worked through the exploratory investigation and cleaning of the data by utilizing GitHub to collaborate on an R Markdown file.

Please refer to our GitHub repository for the link to our csv data for the first dataset. It is labelled `diseaseddivadata`.

```
#Load data downloaded from
rawdat<- read.csv("~/Desktop/Data-Wrangling-Project/diseasedivadata.csv")
```

To gain an understanding for the types of variables we would be analyzing along with some of the numeric summaries for the continuous variables, we employed the summary function.

```
##    Study.ID          Patient.ID         Sample.ID
##  Length:150         Length:150         Length:150
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  Abiraterone..ABI..and.Enzalutamide..ENZA..Exposure.Status Diagnosis.Age
##  Length:150                                                 Min.   :40.00
##  Class :character                                           1st Qu.:62.00
##  Mode  :character                                           Median :68.00
##                                                             Mean   :66.84
##                                                             3rd Qu.:72.00
##                                                             Max.   :85.00
##                                                             NA’s   :3
##  Cancer.Type        Cancer.Type.Detailed Center.of.sequencing
##  Length:150         Length:150           Length:150
##  Class :character   Class :character     Class :character
##  Mode  :character   Mode  :character     Mode  :character
##
##
##
##
##  Fraction.Genome.Altered Mutation.Count    Oncotree.Code
##  Min.   :0.0072          Min.   :  26.00   Length:150
##  1st Qu.:0.2285          1st Qu.:  55.25   Class :character
##  Median :0.3570          Median :  71.50   Mode  :character
##  Mean   :0.3767          Mean   : 114.81
##  3rd Qu.:0.5432          3rd Qu.: 120.75
##  Max.   :0.8466          Max.   :1255.00
##
##  Prior.Treatment    Number.of.Samples.Per.Patient Sample.Type
##  Length:150         Min.   :1                     Length:150
##  Class :character   1st Qu.:1                     Class :character
##  Mode  :character   Median :1                     Mode  :character
##                     Mean   :1
##                     3rd Qu.:1
##                     Max.   :1
##
##      Sex            Site.Sequenced     Somatic.Status     tumor.content
##  Length:150         Length:150         Length:150         Min.   :20.00
##  Class :character   Class :character   Class :character   1st Qu.:44.00
##  Mode  :character   Mode  :character   Mode  :character   Median :62.50
##                                                           Mean   :60.53
##                                                           3rd Qu.:75.00
##                                                           Max.   :98.00
##
##   Tumor.Site         Tumor.Type
##  Length:150         Length:150
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
##
```

Preliminary findings indicate, mean age of diagnosis for prostate cancer is approximately 67 years of age. Average tumor content found in these biopsies appears to be 60%. Initial analysis revealed that there were several cells for which NA values were present. The following command was used to eliminate the NAs saving the new data to a data frame called complete_dat.

```
#Remove NAs
complete_dat <- na.omit(rawdat)
```

Upon removal of the NAs, the data set reduced to 147 rows and 20 columns. Three patients from the original extract contained in rows 71 to 73 were therefore excluded from subsequent analyses due to the NAs being present in both the Diagnosis.Age column along with the Abiraterone (ABI) and Enzalutamide (ENZA) Exposure Status. These two columns are crucial in further analyses being as age is a related factor for severity of disease. Furthermore, without information as to prior exposure status to the androgen deprivation therapies, assessment of mCRPC is incomplete.

Further analysis of the data revealed that several columns were found to have duplicated values. To assess whether the individual cases included in the sample were duplicated, we utilized the duplicated command to check the Sample ID column for repeat values. As shown below, none were found.

```
#Check for duplicates
duplicated(complete_dat$Sample.ID)
```

```
##   [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [37] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [49] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [61] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [73] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [85] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
##  [97] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [109] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [121] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [133] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [145] FALSE FALSE FALSE
```

Upon cursory glance the Patient ID and Sample ID column appeared to be populated with the exact same values per each row. To confirm this assumption we used the following syntax to check if the values between the two columns were unique. The syntax reads as follows, from the complete_dat data frame check if Patient Id is not contained in Sample Id then output the results.

```
#Check if sample id is dup
complete_dat[!complete_dat$Patient.ID %in% complete_dat$Sample.ID, ]
```

```
##  [1] Study.ID
##  [2] Patient.ID
##  [3] Sample.ID
##  [4] Abiraterone..ABI..and.Enzalutamide..ENZA..Exposure.Status
##  [5] Diagnosis.Age
##  [6] Cancer.Type
##  [7] Cancer.Type.Detailed
##  [8] Center.of.sequencing
##  [9] Fraction.Genome.Altered
## [10] Mutation.Count
## [11] Oncotree.Code
## [12] Prior.Treatment
## [13] Number.of.Samples.Per.Patient
## [14] Sample.Type
## [15] Sex
## [16] Site.Sequenced
## [17] Somatic.Status
## [18] tumor.content
## [19] Tumor.Site
## [20] Tumor.Type
## <0 rows> (or 0-length row.names)
```

The output above is empty indicating that all the Patient Ids are contained within the Sample Id. In other words the two columns are the same.

Several columns appeared to be populated by the same value for each row. For example, the Cancer.Type column was entirely populated by the value "Prostate Adenocarcinoma". To determine whether this assumption was indeed true we created the following for loop that cycled through each column to check that the values contained in the column were unique.

```
cols_nonunique=DiseaseDivas::find_nonuniq(complete_dat)
```

```
## [1]  1  6  7 11 13 14 15 17 20
```

```
remove_nonuniques <- complete_dat[,-cols_nonunique]
```

Consequently, after removal of the data columns with repeated values we were left with the "remove_nonuniques" dataframe which contained 147 rows and 11 columns.

Aim 1

Aim 1.1: Classify severity of cancer by site.

We used the dplyr library for most of the data manipulation for Aim 1.1. To begin our analysis, we extracted the two columns of interest: "Center of sequencing", which has the locations of where each patient sample was sequenced, and "Tumor content", which contains the tumor content of each patient sample. Tumor content was classified as percentages that were defined by computational analysis, based on mutant allele variant fractions and zygosity shifts. We made a separate table using just these two columns from the filtered data and named it "aim_1_1".

```
# Load dplyr library for data manipulation.
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# Extract columns needed for Aim 1.1 and make subtable
aim_1_1 = complete_dat %>%
  select(Center.of.sequencing, tumor.content)
```

The next step in our analysis was to create a classification system for tumor content and categorize each patient sample as either low, medium, or high tumor content level. The study we downloaded our raw data from only included patient samples whose tumor content was greater than 20%, so we made a simple classification system in which 20-45% tumor content is considered low, 46-70% tumor content is considered medium, 71-100% tumor content is considered high. To add this classification to our data frame, we created a new column named "TC_Level" and added each classification for patient sample as a factor using if-else statements.

```
# 20 - 45 is low
# 46 - 70 is medium
# 71 - 98 is high

# Create and add classification system to data frame
aim_1_1$TC_level <- as.factor(ifelse(aim_1_1$tumor.content < 46, 'Low',
                     ifelse(aim_1_1$tumor.content < 71, 'Medium', 'High')))
```

The next step in our analysis was to group sequencing centers (aka the "Center of sequencing" column) into geographic regions. Sequencing centers in our data set included Dana-Farber Cancer Institute, Memorial Sloan Kettering Cancer Center, Karmanos Cancer Center, University of Michigan, University of Washington, and the Insitute of Cancer Research & The Royal Marsden. Geographic regions were grouped as shown below:

EAST COAST LOCATIONS: DFCI (Boston, MA) MSKCC (New York, NY)

MIDWEST LOCATIONS: Karmanos (Detroit, MI) U Michigan (Ann Arbor, MI)

WEST COAST LOCATIONS: U Washington (Seattle, WA)

INTERNATIONAL LOCATIONS: RM-ICR (London, UK)

To add the geographic regions to our data frame, we followed the same step used to add the tumor content classifications to the data frame – we created a factor using if-else statements and added the region for each patient sample, as shown below

```
# Add geographic region based on sequencing center
aim_1_1$Site_region <- as.factor(
  ifelse(aim_1_1$Center.of.sequencing == 'DFCI' |
           aim_1_1$Center.of.sequencing == 'MSKCC', 'East Coast',
         ifelse(aim_1_1$Center.of.sequencing == 'Karmanos Cancer Insitute' |
                  aim_1_1$Center.of.sequencing == 'U Michigan', 'Midwest',
                     ifelse(aim_1_1$Center.of.sequencing == 'U Washington',
                            'West Coast', 'International'))))
```

Now that we have created a classification system for tumor content and grouped sequencing centers by geographic region, we started to look at the data numerically and visually. To get a better understanding of the distribution of the tumor content levels across the geographic region, we created a simple table looking at the mean tumor content across each sequencing region. We made this table using the dplyr and gtsummary packages. The gtsummary package contains the tbl_summary() function which is very helpful when trying to create tables with different types of statistical information. We used piping to select the proper data columns from our original aim_1_1 subtable, and then we used the gtsummary package to design the table as shown below.

```
library(gtsummary)
# Create numeric summary of mean tumor content by sequencing region
raw_tc_tab = aim_1_1 %>%
  select(Site_region, tumor.content) %>%
  tbl_summary(by=Site_region, statistic =
                list(all_continuous() ~ "{mean} ({sd})"),
              label = tumor.content ~ "Mean Tumor Content")  %>%
  modify_spanning_header( all_stat_cols() ~ "**Sequencing Region**") %>%
  modify_header(label = "")
raw_tc_tab
```

```
## Table printed with ‘knitr::kable()‘, not {gt}. Learn why at
## http://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include ‘message = FALSE‘ in code chunk header.
```

	East Coast, N = 28	**International**, N = 30	**Midwest**, N = 48	**West Coast**, N = 41
Mean Tumor Content	63 (18)	58 (19)	62 (21)	58 (20)

From the table created above, we can see that the mean tumor content seems to be fairly equal among the geographic regions covered in the data. The mean tumor content ranges from 58% in the West Coast/internationally to 63% on the East Coast. Additionally, the standard deviations are also relatively similar across the regions, ranging from 18% to 21%.

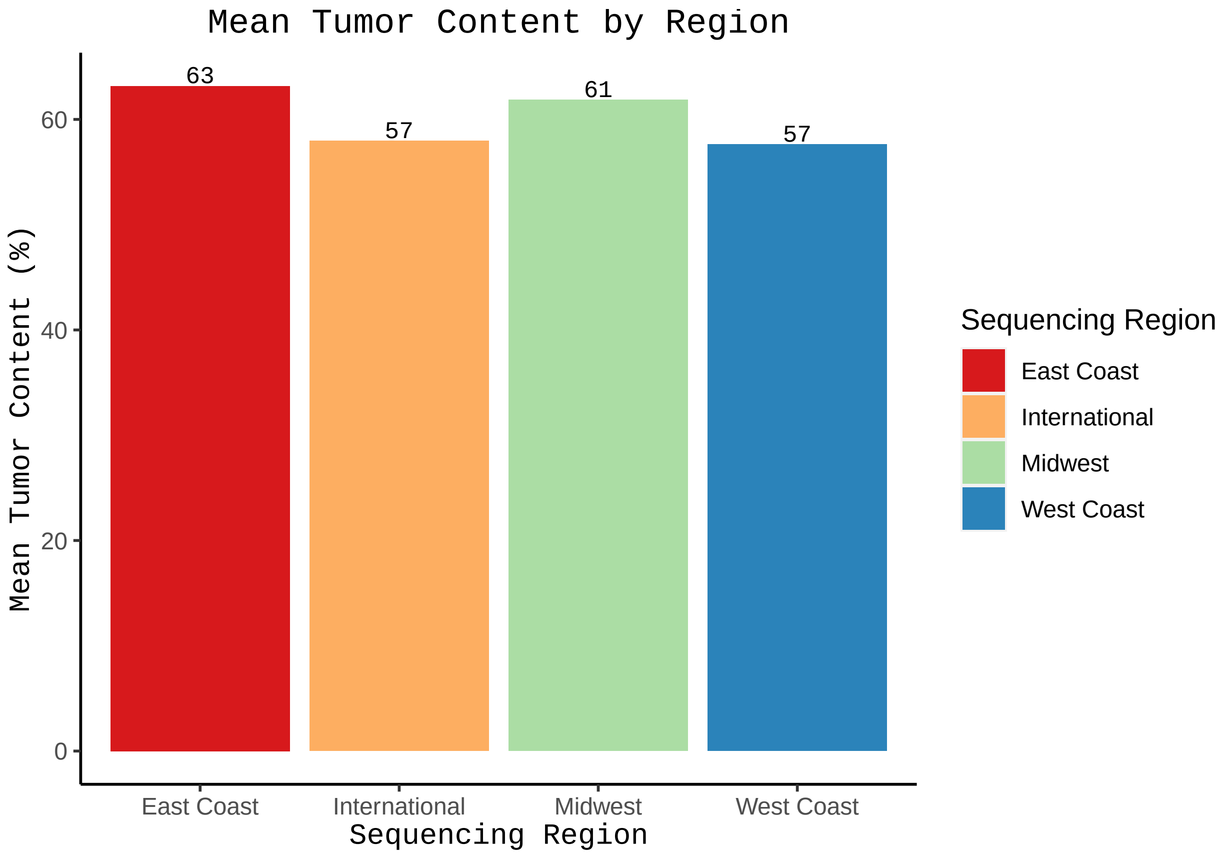
Next, we tried to look at the data visually. We decided to make a bar plot to visually assess the distribution of the mean tumor content percentages across sequencing regions. In order to create my plot, we first used the tidyverse package to group my data by sequencing region, and then get the mean tumor content levels for each group. We made a new table, “raw_tc_df”, which contains the mean tumor content level by sequencing region that we could feed into ggplot to create our visualization.

Then, we used ggplot to create our boxplot. Using the data from “raw_tc_df”, we plotted the mean tumor content on the y axis against the sequencing regions on the x axis. We then used several features (geom_text, scale_fill_brewer, theme, just to name a few) to improve the aesthetics of the plot.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.5     v stringr 1.4.0
## v tidyr   1.1.4     v forcats 0.5.1
## v readr   2.0.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# Create subtable which contains mean tumor content levels by sequencing region
raw_tc_df = aim_1_1 %>%
  group_by(Site_region) %>%
  summarise_at(vars(tumor.content), list(name = mean))
raw_tc_df <- raw_tc_df %>%
  rename(mean.tumor.content=name)

# Plot mean tumor content levels for each sequencing region
ggplot(data=raw_tc_df, mapping=aes(x=Site_region, y=mean.tumor.content,
                                    fill=Site_region)) +
  ggtitle("Mean Tumor Content by Region") + # Set title
  # Add mean tumor content values to the bars
  geom_text(aes(label = as.integer(mean.tumor.content)), vjust = -0.2, size = 3,
            position = position_dodge(0.9), colour="black", family="Courier") +
  geom_bar(stat='identity')+
  # Set color palette
  scale_fill_brewer(palette = "Spectral") +
  # Set legend title
  guides(fill=guide_legend(title="Sequencing Region")) +
  # Remove background grid, change fonts, make it pretty
  theme(panel.background = element_blank(),
        axis.line = element_line(colour="black"),
        plot.title = element_text(hjust=0.5, family="Courier"),
        axis.title.x = element_text(family="Courier"),
        axis.title.y = element_text(family="Courier")) +
  xlab("Sequencing Region") + ylab("Mean Tumor Content (%)")
```



After visually assessing the mean tumor content levels by region, we can see that the tumor content levels do seem to be fairly equal among the geographic regions. Next, we decided to look at the distributions of tumor content levels (using my classification system) by center of sequencing. We made this table using the gtsummary package again. We also ran a Fisher's exact test and obtained a p-value of 0.9, suggesting that there is not a significant association between Sequencing Site and tumor content level.

```
# Make a table to look at tumor content levels across sequencing site
p1 = aim_1_1 %>%
  select(Center.of.sequencing, TC_level) %>%
  tbl_summary(by = TC_level,
                        label = Center.of.sequencing ~ "Sequencing Site") %>%
  add_p() %>%
  modify_header(label = "") %>%
  modify_spanning_header(all_stat_cols() ~ "Tumor Content Level") %>%
  bold_labels()

p1
```

```
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## http://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in code chunk header.
```

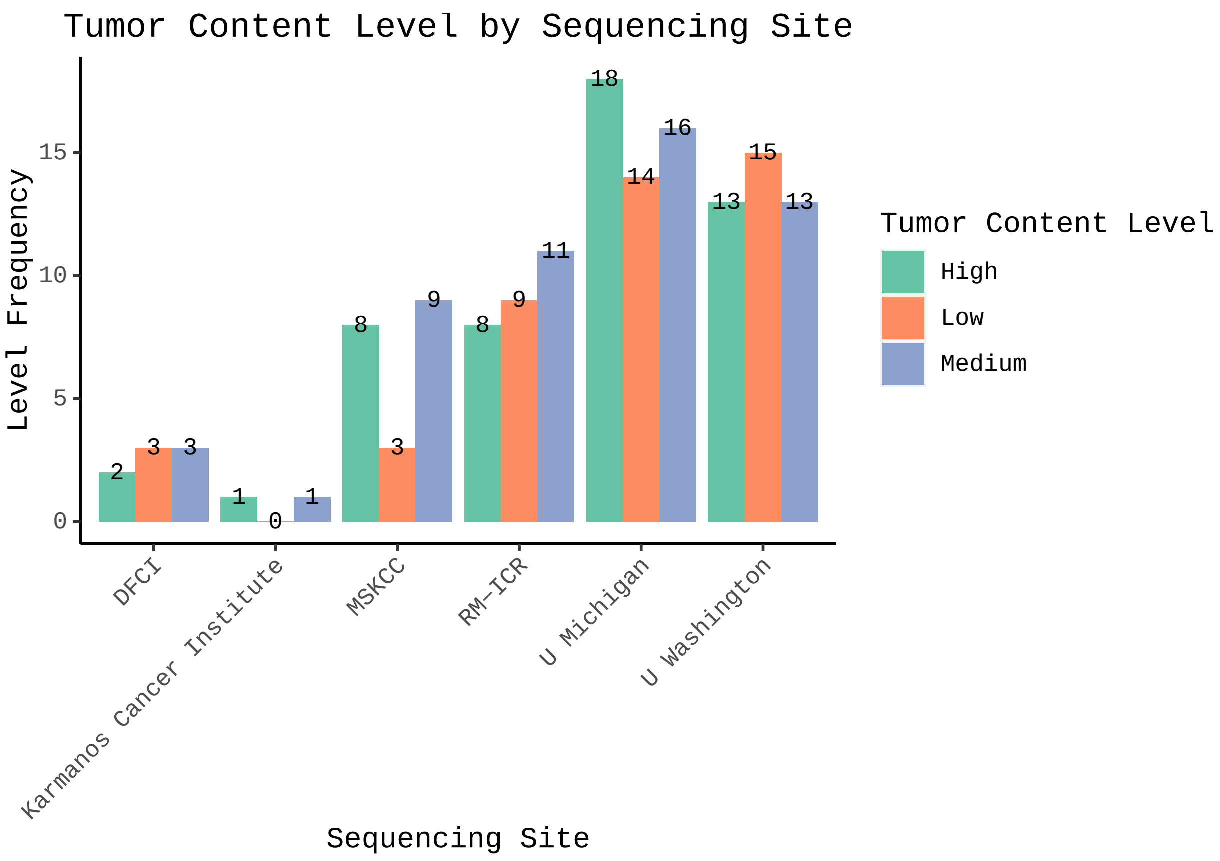
	High, N = 50	**Low**, N = 44	**Medium**, N = 53	**p-value**
Sequencing Site				0.9
DFCI	2 (4.0%)	3 (6.8%)	3 (5.7%)	
Karmanos Cancer Institute	1 (2.0%)	0 (0%)	1 (1.9%)	
MSKCC	8 (16%)	3 (6.8%)	9 (17%)	
RM-ICR	8 (16%)	9 (20%)	11 (21%)	
U Michigan	18 (36%)	14 (32%)	16 (30%)	
U Washington	13 (26%)	15 (34%)	13 (25%)	

To get a better visual assessment of the tumor content levels by sequencing sites, we made a grouped bar plot using ggplot:

```
mid_tab=table(aim_1_1$TC_level, aim_1_1$Center.of.sequencing)
mid_df = data.frame(mid_tab)
mid_df
```

```
##      Var1                      Var2 Freq
## 1    High                      DFCI    2
## 2     Low                      DFCI    3
## 3  Medium                      DFCI    3
## 4    High Karmanos Cancer Institute    1
## 5     Low Karmanos Cancer Institute    0
## 6  Medium Karmanos Cancer Institute    1
## 7    High                     MSKCC    8
## 8     Low                     MSKCC    3
## 9  Medium                     MSKCC    9
## 10   High                    RM-ICR    8
## 11    Low                    RM-ICR    9
## 12 Medium                    RM-ICR   11
## 13   High                U Michigan   18
## 14    Low                U Michigan   14
## 15 Medium                U Michigan   16
## 16   High              U Washington   13
## 17    Low              U Washington   15
## 18 Medium              U Washington   13
```

```r
grouped_plot_a = ggplot(data = mid_df,
                        mapping = aes(x=Var2, y=Freq, fill=Var1)) +
  geom_bar(stat="identity", position = "dodge") +
  geom_text(aes(label = Freq), vjust = 0.5, size = 3,
            position = position_dodge(0.9), colour="black", family="Courier") +
  scale_fill_brewer(palette = "Set2") +
  ggtitle("Tumor Content Level by Sequencing Site") +
  theme(panel.background = element_blank(),
        axis.line = element_line(colour="black"),
        plot.title = element_text(hjust=0.5, family="Courier"),
        axis.title.x = element_text(family="Courier"),
        axis.title.y = element_text(family="Courier"),
        axis.text.x = element_text(family="Courier",angle = 45, vjust = 1,
                                   hjust = 1),
        axis.text.y = element_text(family="Courier"),
        legend.text=element_text(family="Courier"),
        legend.title=element_text(family="Courier")) +
  guides(fill=guide_legend(title="Tumor Content Level")) +
  xlab("Sequencing Site") + ylab("Level Frequency")
grouped_plot_a
```



We then realized that this table might be misleading since some sequencing sites have a much larger sample size than others, so we decided to look at the tumor content levels by sequencing site region instead. This table was also made using gtsummary.

```r
# Create a table to compare tumor content levels across sequencing site regions
library(dplyr)
aim_1_1 %>%
  tbl_cross(
    row = TC_level,
    col = Site_region,
    percent = "cell", label = TC_level ~ "Tumor Content"
  ) %>% modify_header(label = "") %>%
  modify_spanning_header(all_stat_cols() ~ "Sequencing Region") %>%
  add_p() %>% bold_labels()
```

```
## Table printed with ‘knitr::kable()‘, not {gt}. Learn why at
## http://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include ‘message = FALSE‘ in code chunk header.
```

	East Coast	International	Midwest	West Coast	**Total**	**p-value**
Tumor Content						0.9
High	10 (6.8%)	9 (6.1%)	18 (12%)	13 (8.8%)	50 (34%)	
Low	6 (4.1%)	9 (6.1%)	14 (9.5%)	15 (10%)	44 (30%)	
Medium	12 (8.2%)	12 (8.2%)	16 (11%)	13 (8.8%)	53 (36%)	
Total	28 (19%)	30 (20%)	48 (33%)	41 (28%)	147 (100%)	

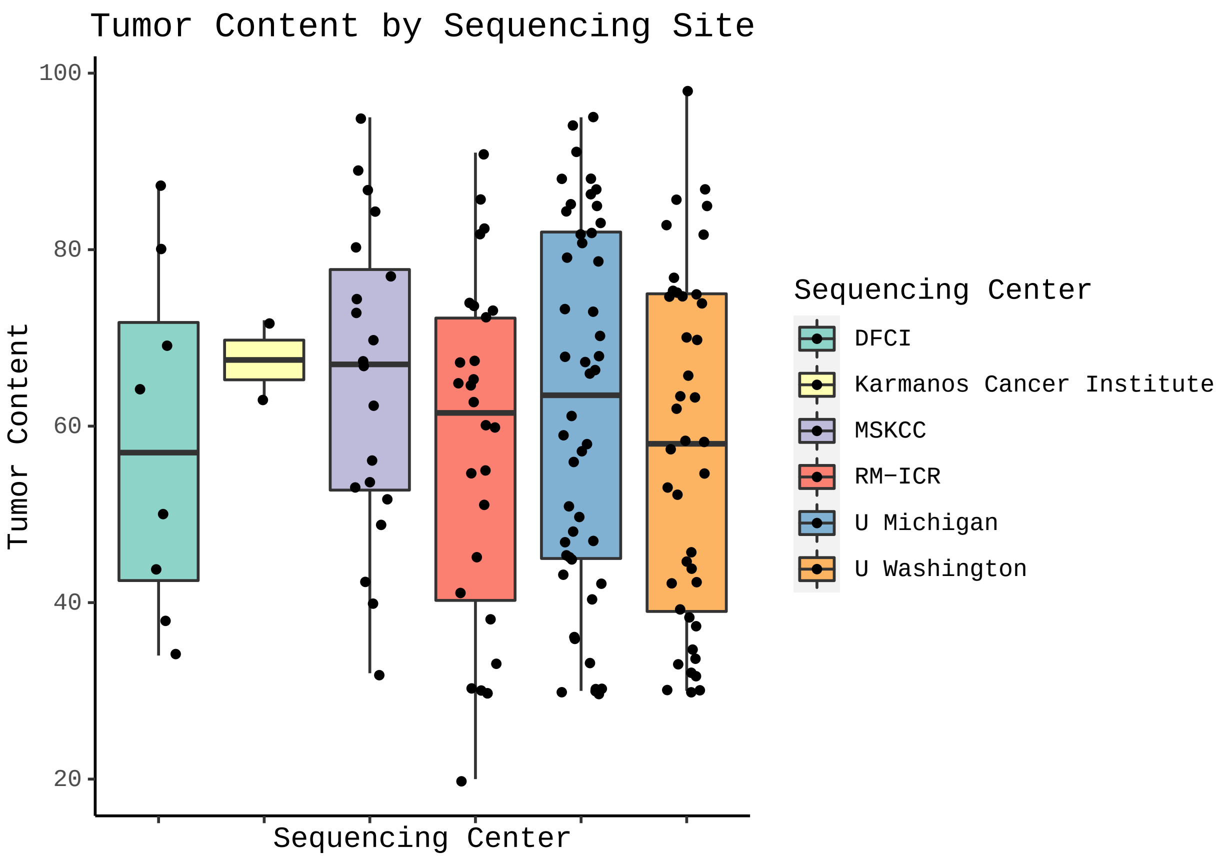
Our next visualization is a boxplot comparing tumor content by sequencing site. We used the theme feature in ggplot to change the fonts and colors in the plot.

```r
# Plot the tumor content percentages across sequencing site
aim_1_1 %>% select(Center.of.sequencing, tumor.content) %>%
  tbl_summary(by = Center.of.sequencing)
```

```
## Table printed with ‘knitr::kable()‘, not {gt}. Learn why at
## http://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include ‘message = FALSE‘ in code chunk header.
```

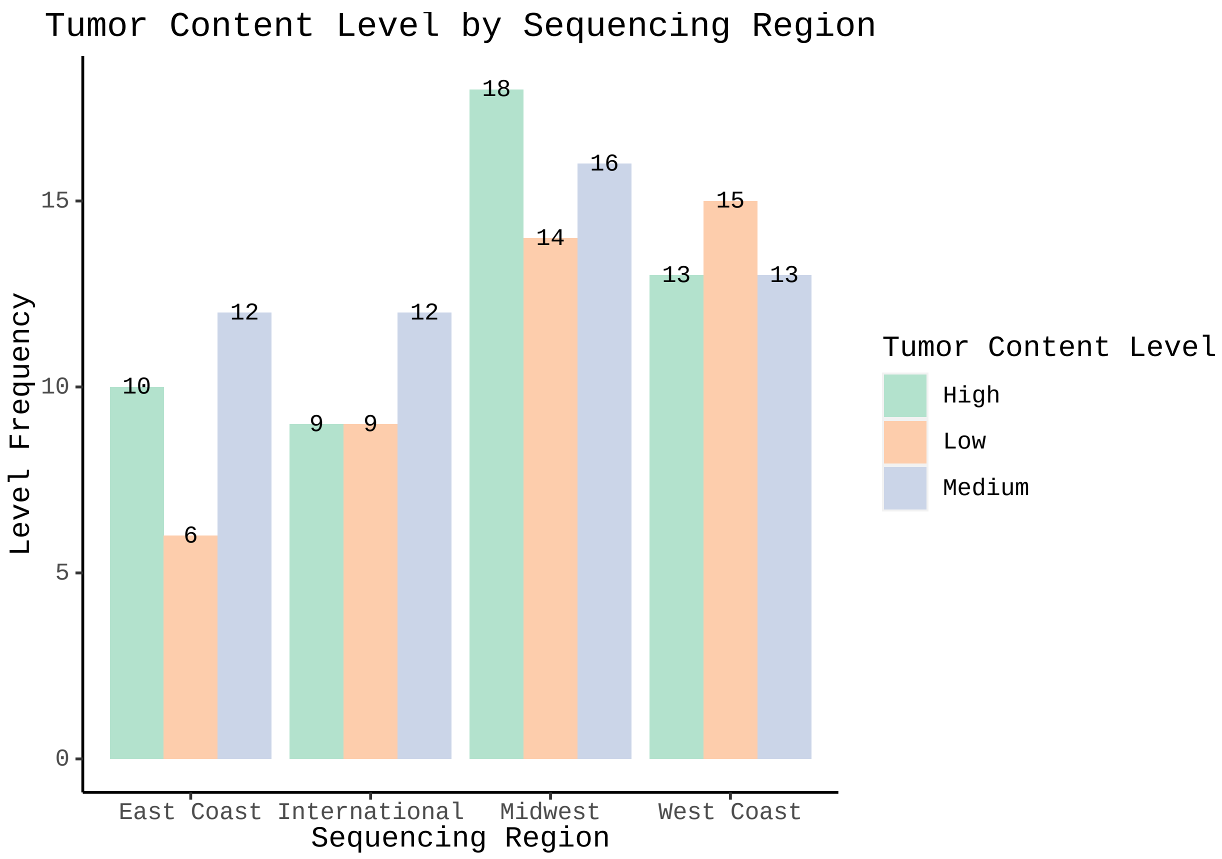
Characteristic	**DFCI**, N = 8	**Karmanos Cancer Institute**, N = 2	**MSKCC**, N = 20	**RM-ICR**, N = 28	**U Michigan**, N = 48	**U Washington**, N = 41
tumor.content	57 (42, 72)	68 (65, 70)	67 (53, 78)	62 (40, 72)	64 (45, 82)	58 (39, 75)

```r
xlabs <- paste(levels(aim_1_1$Center.of.sequencing),
               "\n(N=",table(aim_1_1$Center.of.sequencing),")",sep="")

raw_plot = ggplot(data = aim_1_1, mapping = aes(x=Center.of.sequencing,
                                                y=tumor.content,
                                                fill=Center.of.sequencing)) +
  theme(panel.background = element_blank(),
        axis.line = element_line(colour="black"),
        plot.title = element_text(hjust=0.5, family="Courier"),
        axis.title.x = element_text(family="Courier"),
        axis.title.y = element_text(family="Courier"),
        axis.text.x=element_blank(),
        axis.text.y = element_text(family="Courier"),
        legend.text=element_text(family="Courier"),
        legend.title=element_text(family="Courier")) +
  scale_fill_brewer(palette = "Set3") +
  guides(fill=guide_legend(title="Sequencing Center")) +
  xlab("Sequencing Center") + ylab("Tumor Content") +
  ggtitle("Tumor Content by Sequencing Site") +geom_boxplot() +
  geom_jitter(shape=16, position=position_jitter(0.2))

raw_plot
```



Lastly, we plotted the tumor content levels by sequencing site region. We first made a subtable to count the frequencies of tumor content levels by site region, and then fed this table into ggplot. Once again, we utilized the theme feature to add labels to each bar and customize the fonts and colors in the graph.

```
# Make table to count frequencies of tumor content levels by site region
tab = table(aim_1_1$TC_level, aim_1_1$Site_region)
tab1 = data.frame(tab)

# Stacked barplot
#barplot(tab, legend = TRUE, args.legend = list(x = "topleft",
#                              inset = c(0.07, -0.1)))
# Grouped Barplot
grouped_plot = ggplot(data = tab1, mapping = aes(x=Var2, y=Freq, fill=Var1)) +
  geom_bar(stat="identity", position = "dodge") +
  geom_text(aes(label = Freq), vjust = 0.5, size = 3,
            position = position_dodge(0.9), colour="black", family="Courier") +
  scale_fill_brewer(palette = "Pastel2") +
  ggtitle("Tumor Content Level by Sequencing Region") +
  theme(panel.background = element_blank(),
        axis.line = element_line(colour="black"),
        plot.title = element_text(hjust=0.5, family="Courier"),
        axis.title.x = element_text(family="Courier"),
        axis.title.y = element_text(family="Courier"),
        axis.text.x = element_text(family="Courier"),
        axis.text.y = element_text(family="Courier"),
        legend.text=element_text(family="Courier"),
        legend.title=element_text(family="Courier")) +
  guides(fill=guide_legend(title="Tumor Content Level")) +
  xlab("Sequencing Region") + ylab("Level Frequency")
grouped_plot
```



And that completes the analysis for aim 1.1!

AIM 1.2: Classify by exposure status (grouped by prior treatment) by site

Just like in Aim 1.1, we are going to use the library `dplyer` to perform data manipulation for this Aim. In order to know more about the exposure status, we select two columns: `Abiraterone..ABI..and.Enzalutamide..ENZA..Exposu` which is a binary column for a patient's exposure to Abiraterone and Enzalutamide (contains Yes's and No's) and `Prior.Treatment` stores the information if the patient has ever received any prior Taxane treatment or not.

```
# Load dplyr library for data manipulation.
library(dplyr)
# Extract columns needed for Aim 1.2 and make sub-table
aim_1_2 = complete_dat %>%
  select(Center.of.sequencing,Exposure.Status =
           Abiraterone..ABI..and.Enzalutamide..ENZA..Exposure.Status,
         Prior.Treatment) %>%
  group_by(Prior.Treatment)

head(aim_1_2,10)
```

```
## # A tibble: 10 x 3
## # Groups:   Prior.Treatment [2]
##    Center.of.sequencing Exposure.Status Prior.Treatment
##    <chr>                <chr>           <chr>
##  1 DFCI                 Yes             No Prior Taxane Treatment
##  2 DFCI                 No              No Prior Taxane Treatment
##  3 DFCI                 No              No Prior Taxane Treatment
##  4 DFCI                 Yes             No Prior Taxane Treatment
##  5 DFCI                 No              No Prior Taxane Treatment
##  6 DFCI                 No              No Prior Taxane Treatment
##  7 DFCI                 No              No Prior Taxane Treatment
##  8 DFCI                 Yes             No Prior Taxane Treatment
##  9 MSKCC                Yes             Had prior Taxane treatment
## 10 MSKCC                Yes             No Prior Taxane Treatment
```

In order to have few categories for finding the exposure status based on prior treatment found in different sequencing institutes, we need to club institutes into 4 main regions based on the data we have: East Coast, Midwest, West Coast and International. Similarly in Aim 1.1, we run the same if-else code for Aim 1.2.

```
aim_1_2$Site_region <- as.factor(ifelse(aim_1_2$Center.of.sequencing == 'DFCI' |
                                        aim_1_2$Center.of.sequencing == 'MSKCC',
                                        'East Coast',
                     ifelse(aim_1_2$Center.of.sequencing ==
                              'Karmanos Cancer Insitute' |
                              aim_1_2$Center.of.sequencing == 'U Michigan',
                            'Midwest',
                     ifelse(aim_1_2$Center.of.sequencing == 'U Washington',
                            'West Coast', 'International'))))
```

We now have grouped the Exposure status by Prior Treatment and have added the Sequencing region column into the data frame `aim_1_2`. Next, we calculate the mean of the all `Exposure.Status` counts based on the Sequencing Region and grouped by `Prior.Treatment`

```
library(gtsummary)
# Create numeric summary of mean tumor content by sequencing region
temp = aim_1_2 %>%
  select(Exposure.Status,Prior.Treatment,Site_region) %>%
  tbl_summary(by=Site_region, statistic =
                list(all_continuous() ~ "{mean} ({sd})"),
              label = Exposure.Status ~ "Mean of Exposure Status")  %>%
  modify_spanning_header( all_stat_cols() ~ "**Sequencing Region**") %>%
  modify_header(label = "Exposure Status, grouped by Prior Treatment")
temp
```

```
## Table printed with ‘knitr::kable()‘, not {gt}. Learn why at
## http://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include ‘message = FALSE‘ in code chunk header.
```

Exposure Status, grouped by Prior Treatment	**East Coast**, N = 28	**International**, N = 30	**Midwest**, N = 48	**West Coast**, N = 41
Mean of Exposure Status	17 (61%)	25 (83%)	16 (33%)	14 (34%)
Prior.Treatment				
Had prior Taxane treatment	9 (32%)	23 (77%)	19 (40%)	10 (24%)
No Prior Taxane Treatment	19 (68%)	7 (23%)	29 (60%)	31 (76%)

Next we perform visualizations on the same numbers we just got above and present the data in form of bar graphs with the help of `ggplot2` package and used `geom_bar` to plot the graph grouped by Prior.Treatment and have six graphs for six different `Center.of.sequencing` institutes. Furthermore, to plot the graph, we use the `janitor` and `dplyr` packages to store the tables based on these three variables.

```
library(janitor)
```

```
##
## Attaching package: ’janitor’

## The following objects are masked from ’package:stats’:
##
##     chisq.test, fisher.test
```

```
summary_table = tabyl(aim_1_2, Exposure.Status,Prior.Treatment,
                      Center.of.sequencing)
summary_table
```

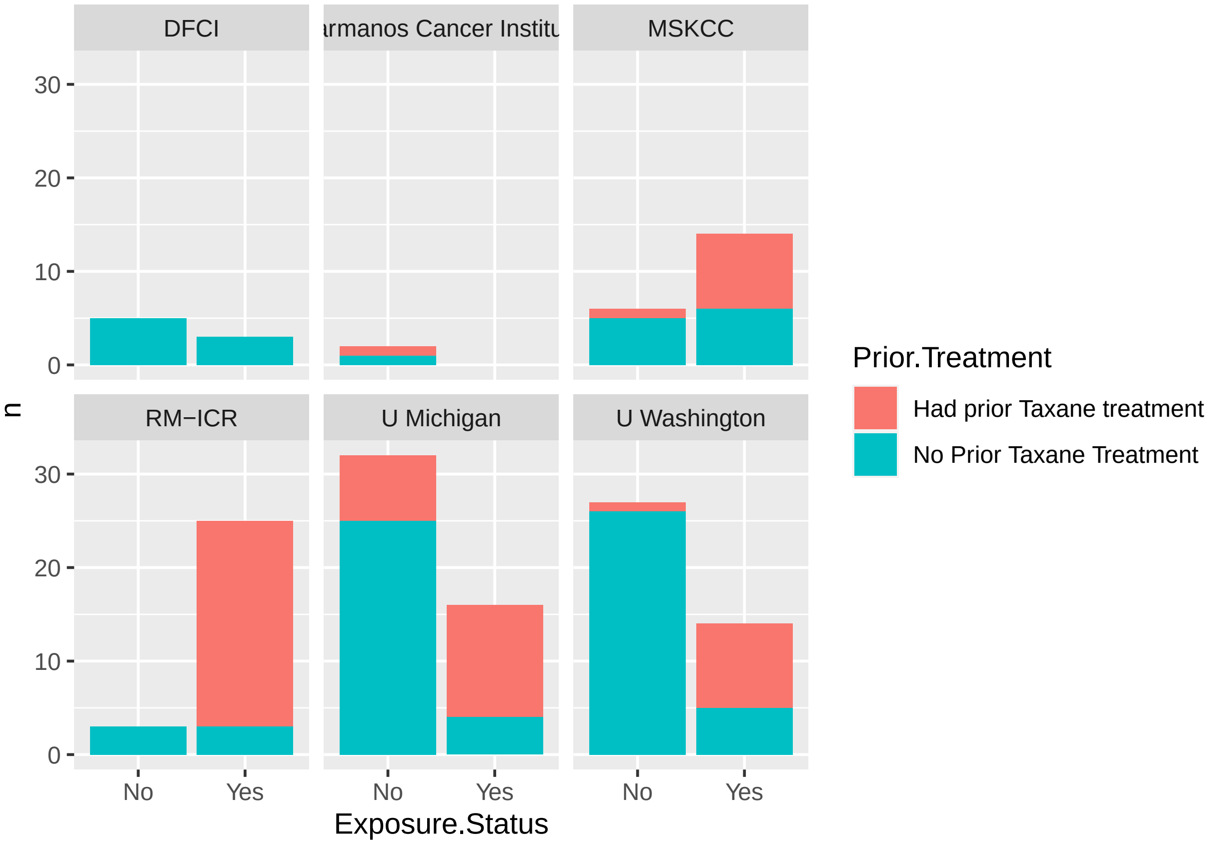
```
## $DFCI
##  Exposure.Status Had prior Taxane treatment No Prior Taxane Treatment
##               No                          0                         5
##              Yes                          0                         3
##
## $‘Karmanos Cancer Institute‘
##  Exposure.Status Had prior Taxane treatment No Prior Taxane Treatment
##               No                          1                         1
##              Yes                          0                         0
##
## $MSKCC
##  Exposure.Status Had prior Taxane treatment No Prior Taxane Treatment
##               No                          1                         5
##              Yes                          8                         6
```

```
##
## $‘RM-ICR‘
##  Exposure.Status Had prior Taxane treatment No Prior Taxane Treatment
##               No                          0                         3
##              Yes                         22                         3
##
## $‘U Michigan‘
##  Exposure.Status Had prior Taxane treatment No Prior Taxane Treatment
##               No                          7                        25
##              Yes                         12                         4
##
## $‘U Washington‘
##  Exposure.Status Had prior Taxane treatment No Prior Taxane Treatment
##               No                          1                        26
##              Yes                          9                         5
```

```
library(dplyr)
my_summary <- aim_1_2 %>%
  count(Center.of.sequencing,Exposure.Status,Prior.Treatment, sort = FALSE)
my_summary
```

```
## # A tibble: 19 x 4
## # Groups:   Prior.Treatment [2]
##    Prior.Treatment            Center.of.sequencing      Exposure.Status     n
##    <chr>                      <chr>                     <chr>           <int>
##  1 Had prior Taxane treatment Karmanos Cancer Institute No                  1
##  2 Had prior Taxane treatment MSKCC                     No                  1
##  3 Had prior Taxane treatment MSKCC                     Yes                 8
##  4 Had prior Taxane treatment RM-ICR                    Yes                22
##  5 Had prior Taxane treatment U Michigan                No                  7
##  6 Had prior Taxane treatment U Michigan                Yes                12
##  7 Had prior Taxane treatment U Washington              No                  1
##  8 Had prior Taxane treatment U Washington              Yes                 9
##  9 No Prior Taxane Treatment  DFCI                      No                  5
## 10 No Prior Taxane Treatment  DFCI                      Yes                 3
## 11 No Prior Taxane Treatment  Karmanos Cancer Institute No                  1
## 12 No Prior Taxane Treatment  MSKCC                     No                  5
## 13 No Prior Taxane Treatment  MSKCC                     Yes                 6
## 14 No Prior Taxane Treatment  RM-ICR                    No                  3
## 15 No Prior Taxane Treatment  RM-ICR                    Yes                 3
## 16 No Prior Taxane Treatment  U Michigan                No                 25
## 17 No Prior Taxane Treatment  U Michigan                Yes                 4
## 18 No Prior Taxane Treatment  U Washington              No                 26
## 19 No Prior Taxane Treatment  U Washington              Yes                 5
```

```
library(ggplot2)
ggplot(my_summary, aes(Exposure.Status, n, fill = Prior.Treatment)) +
  geom_bar(stat = "identity") +
  facet_wrap(facets = vars(Center.of.sequencing))
```



In order to get better visualizations, we decided to congregate the institutes into region, and find the `Exposure.Status` grouped by the `Prior.Treatment` column based on the `Site_region`.

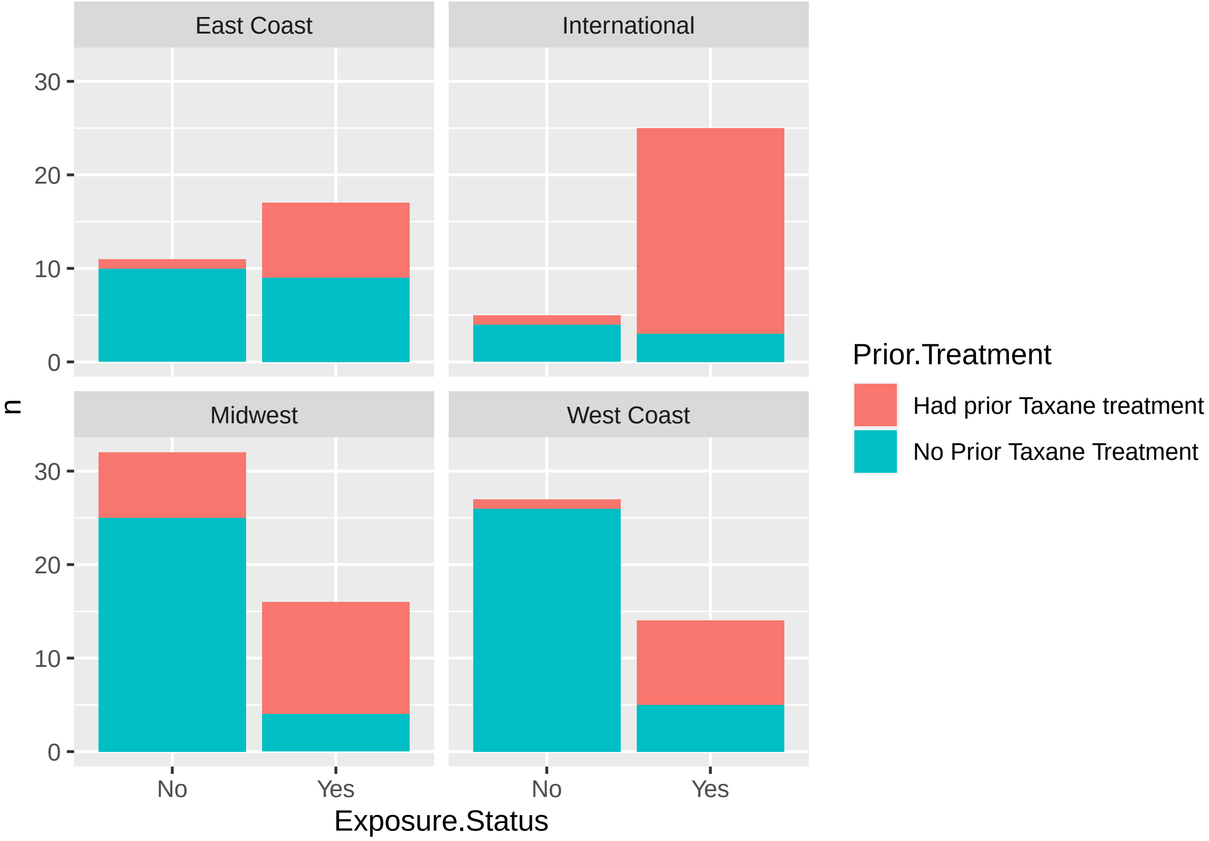
```
library(janitor)
summary.table.region = tabyl(aim_1_2, Exposure.Status,Prior.Treatment,
                             Site_region)
summary.table.region
```

```
## $`East Coast`
##  Exposure.Status Had prior Taxane treatment No Prior Taxane Treatment
##               No                          1                        10
##              Yes                          8                         9
##
## $International
##  Exposure.Status Had prior Taxane treatment No Prior Taxane Treatment
##               No                          1                         4
##              Yes                         22                         3
##
## $Midwest
##  Exposure.Status Had prior Taxane treatment No Prior Taxane Treatment
##               No                          7                        25
##              Yes                         12                         4
##
## $`West Coast`
##  Exposure.Status Had prior Taxane treatment No Prior Taxane Treatment
##               No                          1                        26
##              Yes                          9                         5
```

```
library(dplyr)
my.summary.region <- aim_1_2 %>%
  count(Exposure.Status,Prior.Treatment, Site_region, sort = TRUE)
my.summary.region
```

```
## # A tibble: 16 x 4
## # Groups:   Prior.Treatment [2]
##    Prior.Treatment             Exposure.Status Site_region       n
##    <chr>                       <chr>           <fct>         <int>
##  1 No Prior Taxane Treatment   No              West Coast       26
##  2 No Prior Taxane Treatment   No              Midwest          25
##  3 Had prior Taxane treatment Yes             International    22
##  4 Had prior Taxane treatment Yes             Midwest          12
##  5 No Prior Taxane Treatment   No              East Coast       10
##  6 Had prior Taxane treatment Yes             West Coast        9
##  7 No Prior Taxane Treatment   Yes             East Coast        9
##  8 Had prior Taxane treatment Yes             East Coast        8
##  9 Had prior Taxane treatment No              Midwest           7
## 10 No Prior Taxane Treatment   Yes             West Coast        5
## 11 No Prior Taxane Treatment   No              International     4
## 12 No Prior Taxane Treatment   Yes             Midwest           4
## 13 No Prior Taxane Treatment   Yes             International     3
## 14 Had prior Taxane treatment No              East Coast        1
## 15 Had prior Taxane treatment No              International     1
## 16 Had prior Taxane treatment No              West Coast        1
```

```
library(ggplot2)
ggplot(my.summary.region, aes(Exposure.Status, n, fill = Prior.Treatment)) +
  geom_bar(stat = "identity") +
  facet_wrap(facets = vars(Site_region))
```



As seen in the graphs and tables, the distribution seems to be not normal and hence we decided to perform Fisher's exact test to calculate the p-value. We see that the p-value is less than 0.001, which means it is statistically significant across all confidence levels and hence there is an significant association between Sequencing Site and Exposure Status. Similarly we also get p-value less than 0.001 for Prior Treatment status and Exposure Status (using Pearson's Chi-Squared test), which means there is a statistically significant association between exposure status/sequencing site and exposure status/prior treatment status.

```
# Create numeric summary comparing exposure status to sequencing site and prior
# treatment status
# Run Fisher's exact test and Chi-squared for sequencing site and prior
# treatment status, respectively
p.test = aim_1_2 %>%
  select(Center.of.sequencing, Exposure.Status, Prior.Treatment) %>%
  tbl_summary(by = Exposure.Status, label =
              Center.of.sequencing ~ "Sequencing Site") %>%
  add_p() %>%
  modify_header(label = "") %>%
  modify_spanning_header(all_stat_cols() ~ "Exposure Status") %>%
  bold_labels()
p.test
```

```
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## http://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in code chunk header.
```

	No, N = 75	**Yes**, N = 72	**p-value**
Sequencing Site			<0.001
DFCI	5 (6.7%)	3 (4.2%)	
Karmanos Cancer Institute	2 (2.7%)	0 (0%)	
MSKCC	6 (8.0%)	14 (19%)	
RM-ICR	3 (4.0%)	25 (35%)	
U Michigan	32 (43%)	16 (22%)	
U Washington	27 (36%)	14 (19%)	
Prior.Treatment			<0.001
Had prior Taxane treatment	10 (13%)	51 (71%)	
No Prior Taxane Treatment	65 (87%)	21 (29%)	

Now that we have completed our analysis for Aim 1, we can conclude that exposure status has shown evidence of associaiton with prior treatment status (Taxane treatment) and sequencing site, and that sequencing site does not seem to be associated with severity of tumor content.

Aim 2

Aim 2.1 - Assign classifications of "high", "medium" or "low" based on the intensity of mutation count

For Aim 2, we will be creating a mutation count classification system similar to the one made and utilized in Aim 1. To assess how we should divide the counts into "high", "medium", and "low", lets first look at the distribution of Mutation Count:

```
finaldata<-remove_nonuniques
summary(finaldata$Mutation.Count)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    26.0    55.0    71.0   107.4   120.5  1071.0
```

Based off of the summary above, we can divide the mutation count into the following levels of severity:

Low Mutation Count: 0-55 mutations Medium Mutation Count: 56-110 mutations High Mutation Count: >110 mutations

Now, lets add this severity classification to our data frame using our tried and true if-else statement and summarize the frequencies of mutation count severity levels:

```
# low = 0 - 55
# med = 56 - 110
# high = >110
finaldata%>%add_column(mu_class="Low")->finaldata
finaldata%>%mutate(mu_class=ifelse(`Mutation.Count`>55 &
                                     `Mutation.Count`<=110,
                                   "Moderate",mu_class))->finaldata
finaldata%>%mutate(mu_class=ifelse(`Mutation.Count`>110,"High",
                                   mu_class))->finaldata
# Summarize frequencies of mutation count severity levels
table(finaldata$mu_class)
```

```
##
##     High      Low Moderate
##       41       38       68
```

Now that we have summarized the frequencies of mutation count severity levels, we can plot the frequencies in relation to biopsy sites.

For plotting the severity on a human graph we can use the package **gganatogram**, which is a package based on ggplot2 and allows for better visualization of organs and tissues. Here, we will be displaying mutation count data onto anatomical structures from the package. Essentially, we will be grouping our data set by biopsy site, and then calculating the mean mutation count for each site, and then use our mutation count classification system to set an appropriate value for visualizing the spread of severity across biopsy sites.

First we need to construct a data frame by renaming the biopsy sites exactly as built in the package **gganatogram**. Then, we can assign each tumor site a value corresponding to the severity level that matches the mutation count levels.

```
# Download all our gganatogram packages
#source("https://neuroconductor.org/neurocLite.R")
#neuro_install("gganatogram")
library(gganatogram)
```

```
## Loading required package: ggpolypath
```

```
library(gridExtra)
```

```
##
## Attaching package: ’gridExtra’
```

```
## The following object is masked from ’package:dplyr’:
##
##     combine
```

```
# Calculate mean mutation count levels for each biopsy site
finaldata %>%
  group_by(Tumor.Site) %>%
  summarise(mean(Mutation.Count))
```

```
## # A tibble: 20 x 2
##    Tumor.Site          ‘mean(Mutation.Count)‘
##    <chr>                                <dbl>
##  1 Abdominal wall mass                 1071
##  2 Bladder                               79
##  3 Bone                                 112.
##  4 Chest Wall                           216
##  5 Dura                                  71
##  6 glans penis                          100
##  7 Liver                                 75.2
##  8 Lymph node                           102.
##  9 Muscle                               113
## 10 Neck Mass                             58
## 11 Pelvic mass                           82
## 12 Penile                               134
## 13 Perirectal                           154
## 14 Peritoneal nodule                     86
## 15 Prostate                              50
## 16 ProstateTURP                         120
## 17 Psoas Mass                            78
## 18 Retroperitonium                       61
## 19 Soft Tissue                          101.
## 20 Thoracic epidural                     55
```

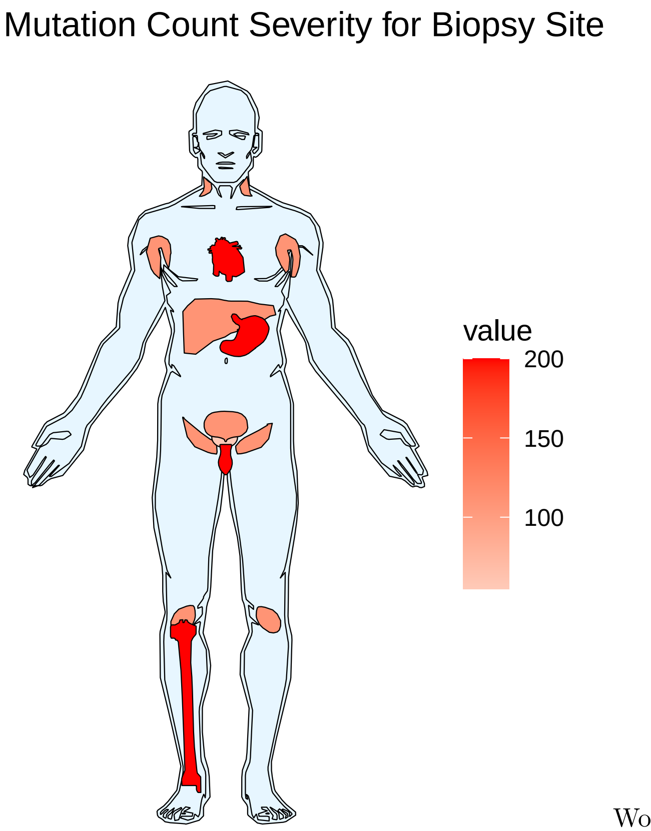
We've successfully calculated the avergae mutation count for each biopsy site! Now, we can start the process to plot these on a human graph. First, note that the ranges for mutation count are much more variable that tumor content. Good thing we created a classification system for this! To ensure that our color scale doesn't range from 0-1000 mutations, which would cause most of the biopsy sites to be colored the same shade since majority of them would be on the lower end of the scale, we can utilize our classification system to normalize our mutation count values to a certain extent.

In the section below, we create a data frame that is formatted for the gganatogram package. It contains two columns – organ and value. Organ is the biopsy site, and value is the mutation count (note that the columns must be named like this in order to be fed into gganatogram).

To normalize the mutation count values, we will manually check which level of mutation each relevant biopsy site matches to, and then set that biopsy site's value to the maximum value of that level's range. For example, lymph node's mutation count average is 101, which falls into the medium severity level, which ranges from 56 to 110, so lymph node's value in our data frame will be 110.

It is also worth noting that we can only plot the biopsy sites which are included in gganatogram's `hgMale_key` data frame due to the way the package works, but for the purposes of data wrangling we figured including the package and all it's visually helpful plots is worth the loss of a couple of sites not included in the library.

```
# For reference, here is our mutation count classification system
# low = 0 - 55
# med = 56 - 110
# high = >110


biopsies <- data.frame(organ = c("lymph_node", "prostate", "liver", "penis",
                                 "bone","heart","stomach","urinary_bladder"),
                       value = c(110, 55, 110, 200, 200, 200, 200, 110),
                       stringsAsFactors = TRUE)


temp_works <-gganatogram(data = biopsies, organism="human", sex="male",
 fill="value",  fillOutline = "#E7F6FF") + theme_void() +
 ggtitle("Mutation Count Severity for Biopsy Site") + coord_fixed() +
  scale_fill_gradient2(low = "grey", mid = "white",
                       high = "red", midpoint = .02)
temp_works
```



Wow! We just made our first human plot that looks at mutation count severity levels by biopsy site! Thanks, `gganatogram`! As we can see from above the penis, bone, heart, and stomach biopsy sites seem to have the highest mutation count. This could be useful to look into in further studies to see if there could be any clinical implications in this area.

Aim 2.2

This data set included a recording of patient exposure to Abiraterone and Enzalutamide (ABI and ENZA). ABI and ENZA are drugs commonly used in the treatment of prostate cancer. Both medications work as an anti-androgen drugs; ABI is an androgen synthase inhibitor, and ENZA inhibits androgen processes in the cell. Close to half of the participants in the study were either exposed or not exposed to ABI and ENZA in the fashion of a randomized control trial. In this aim we are trying to see if the participants in each arm of the study (exposed or not exposed) share similar characteristics, particularly among the Mutation Count, Fraction Genome Altered, and Tumor Content variables. To explore these characteristics of the data set, the data must first be split into those rows corresponding to participants who were exposed to ABI and ENZA and those who were not.

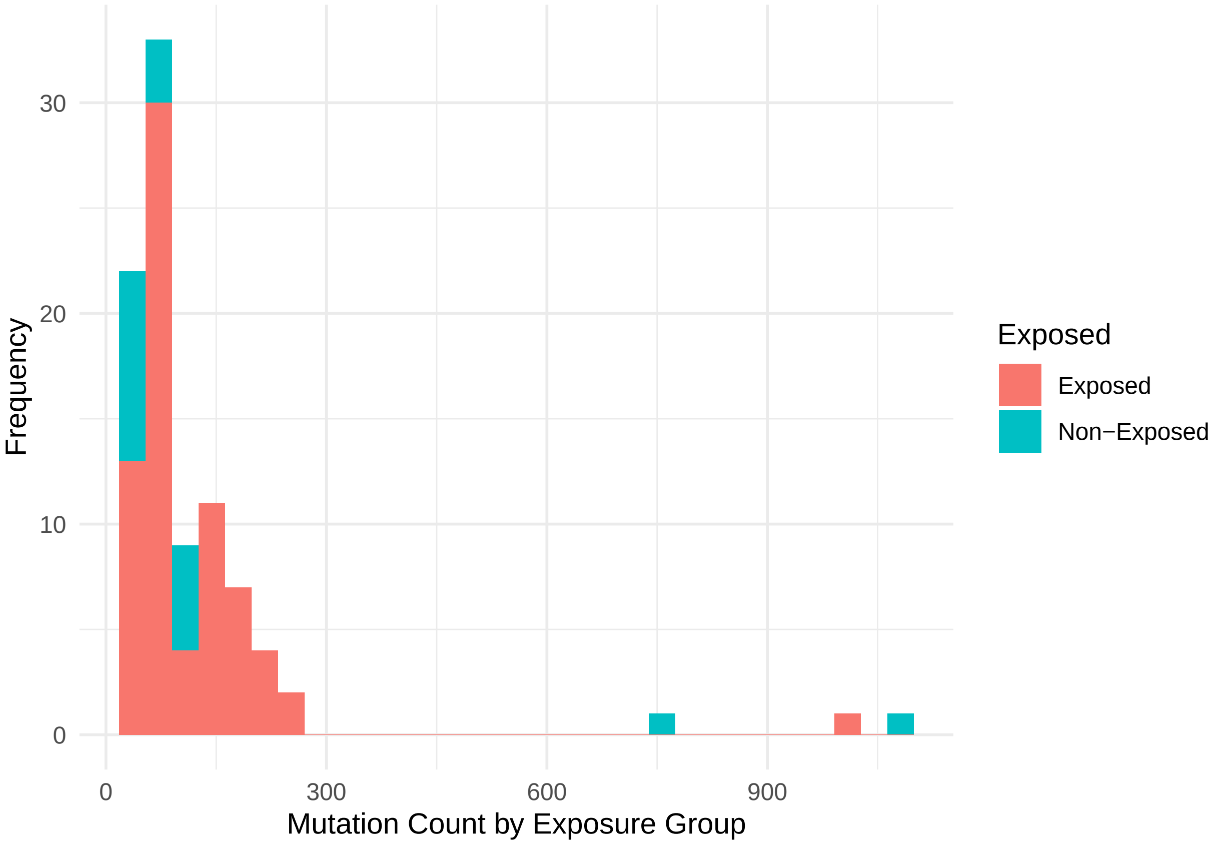
```
#Convert exposure status to numeric binary variable.
exposure_data <- remove_nonuniques
binary_exposure <-
  remove_nonuniques$Abiraterone..ABI..and.Enzalutamide..ENZA..Exposure.Status

binary_exposure[binary_exposure == "Yes"] = 1
binary_exposure[binary_exposure == "No"] = 0

# Add column for binary exposure
exposure_data$Binary.Exposure <- binary_exposure
```

```r
# Make separate tables for exposed vs non-exposed patients
exposed <- which(binary_exposure == 1)
non_exposed <- which(binary_exposure == 0)

exposed_dat <- exposure_data[exposed,]
nonexposed_dat <- exposure_data[non_exposed,]
```

Next, the participants can be plotted by the two groups – exposure vs. no exposure – for comparing mutation counts, fraction of genome that was altered, and tumor content.

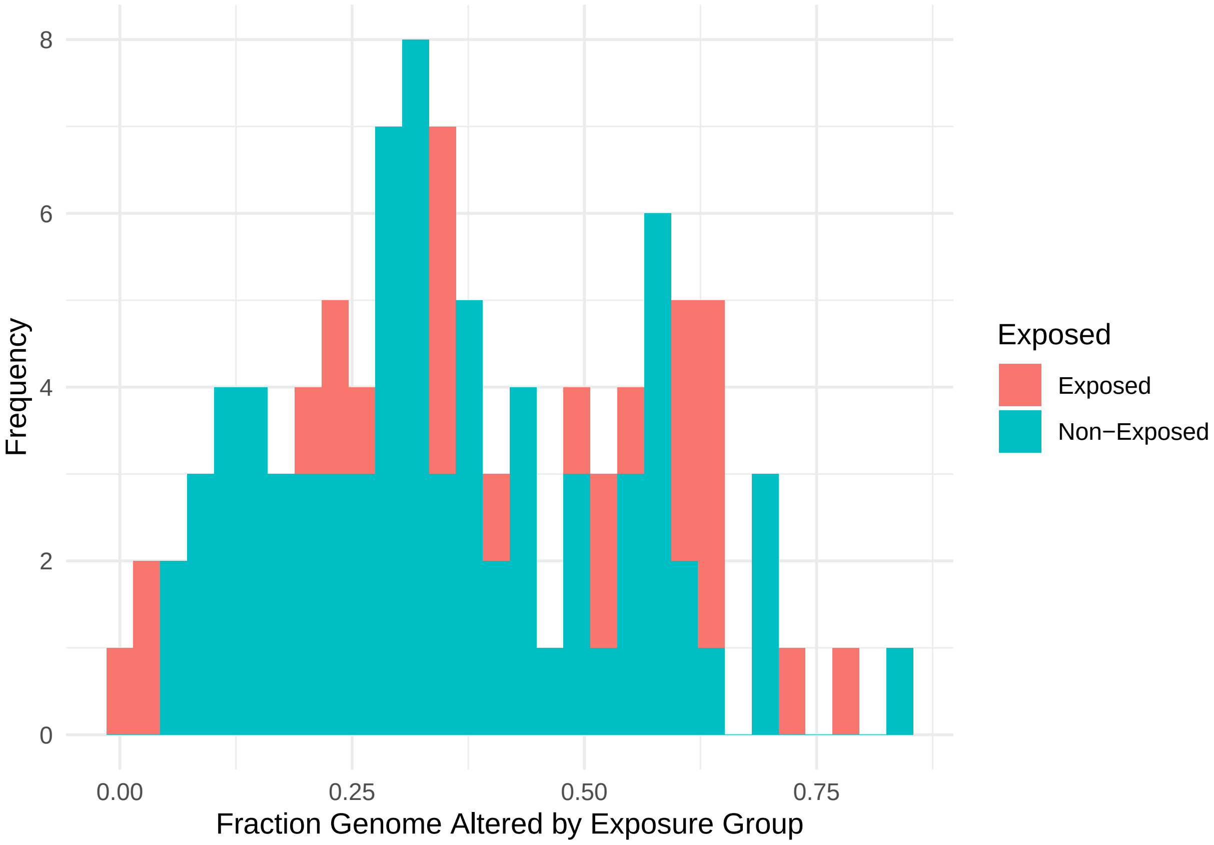
```r
# Construct table used for plotting mutation count, fraction of genome altered,
# and tumor content by exposure status
emc <- exposed_dat %>% select(c("Mutation.Count", "Fraction.Genome.Altered",
                                "tumor.content"))
nemc <- nonexposed_dat %>% select(c("Mutation.Count", "Fraction.Genome.Altered",
                                    "tumor.content"))

emc$Exposed <- "Exposed"
nemc$Exposed <- "Non-Exposed"

amc <- rbind(emc, nemc)

# Plot distributions of mutation count by exposure status
ggplot(amc,aes(x=Mutation.Count, fill=Exposed)) +
  geom_histogram(data=subset(amc,Exposed == 'Non-Exposed')) +
  geom_histogram(data=subset(amc,Exposed == 'Exposed')) +
  xlab("Mutation Count by Exposure Group") +
  ylab("Frequency")+
  theme_minimal()
```

```
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
## ‘stat_bin()‘ using ‘bins = 30‘. Pick better value with ‘binwidth‘.
```



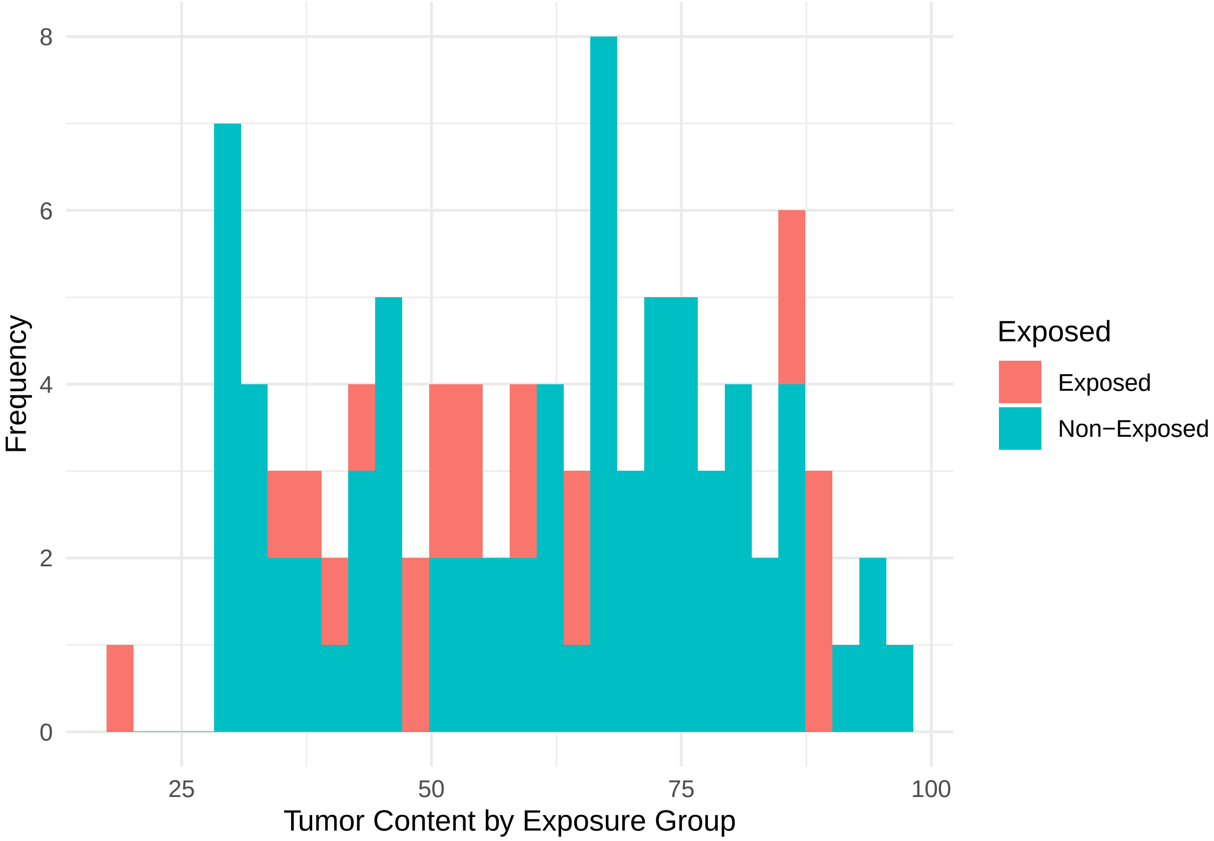
```
# Plot distributions of fraction genome altered by exposure status
ggplot(amc,aes(x=Fraction.Genome.Altered, fill=Exposed)) +
    geom_histogram(data=subset(amc,Exposed == 'Exposed')) +
    geom_histogram(data=subset(amc,Exposed == 'Non-Exposed')) +
    xlab("Fraction Genome Altered by Exposure Group") +
    ylab("Frequency")+
    theme_minimal()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
# Plot distributions of tumor content by exposure status
ggplot(amc,aes(x=tumor.content, fill=Exposed)) +
    geom_histogram(data=subset(amc,Exposed == 'Exposed')) +
    geom_histogram(data=subset(amc,Exposed == 'Non-Exposed')) +
    xlab("Tumor Content by Exposure Group") +
    ylab("Frequency")+
    theme_minimal()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
# Create numeric summary of characteristics by exposure status
colnames(amc) <- c("Mutation Count", "Fraction Genome Altered", "Tumor Content",
                   "Exposed")

amc %>%
  tbl_summary(by = Exposed, label = Exposed ~ "Exposed") %>%
  add_p(test=everything()~"t.test")
```

```
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## http://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in code chunk header.
```

Characteristic	**Exposed**, N = 72	**Non-Exposed**, N = 75	**p-value**
Mutation Count	80 (60, 150)	64 (54, 96)	0.5
Fraction Genome Altered	0.39 (0.23, 0.56)	0.32 (0.22, 0.49)	0.12
Tumor Content	58 (42, 76)	66 (44, 75)	0.6

These visualizations and results show that the groups differ by values that are reasonably expected under random chance. We performed a Welch's two sample t-test (which assumes unequal variances between exposure arms), and obtained p-values of 0.5 (mutation count), 0.12 (fraction genome altered), and 0.6 (tumor content). These values all suggest that there is no association between exposure status and mutation count, fraction genome altered, and tumor content.

Aim 2.3- Is Severity of Mutation Count Related to Frequency of Biopsy Sites?

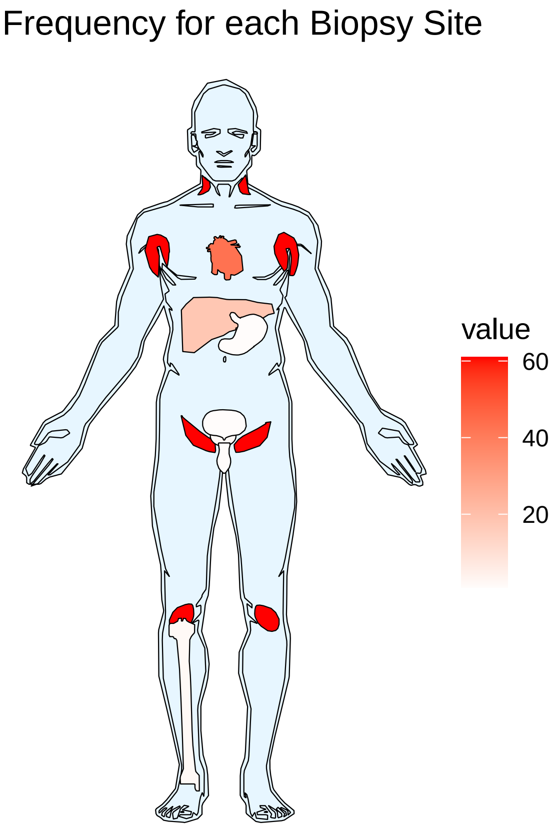
To finish off aim 2, we will be checking if severity of Mutation Count is associated with frequency of biopsy sites sampled in the first data set. Let's first create a data frame that stores the frequency of each biopsy site from our data set:

```
# Frequencies of biopsy site in first data set
aim_2_3 = data.frame(table(finaldata$Tumor.Site))
aim_2_3
```

```
##                    Var1 Freq
## 1  Abdominal wall mass    1
## 2              Bladder    1
## 3                 Bone   43
## 4           Chest Wall    1
## 5                 Dura    1
## 6          glans penis    1
## 7                Liver   18
## 8           Lymph node   61
## 9               Muscle    2
## 10           Neck Mass    1
## 11         Pelvic mass    1
## 12              Penile    1
## 13          Perirectal    1
## 14   Peritoneal nodule    1
## 15            Prostate    2
## 16        ProstateTURP    1
## 17          Psoas Mass    1
## 18     Retroperitonium    1
## 19         Soft Tissue    7
## 20   Thoracic epidural    1
```

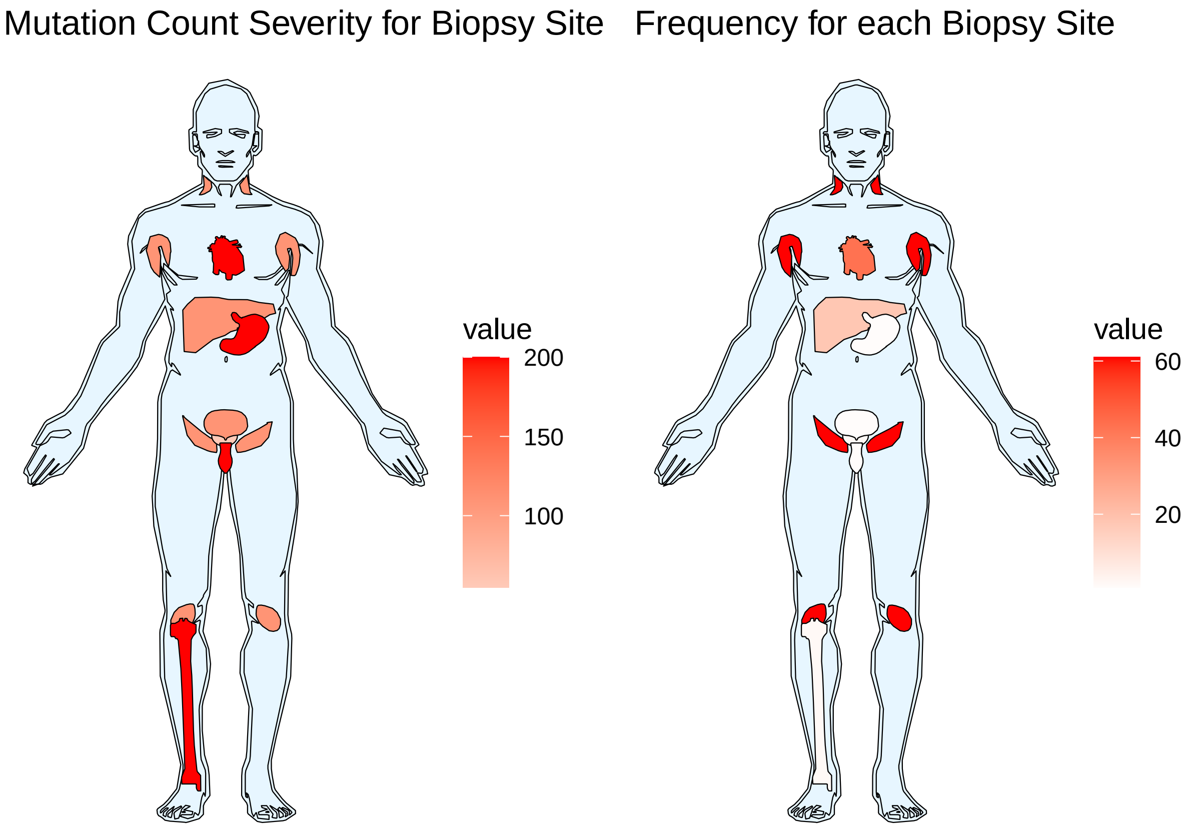
Now we can once again use the `gganatogram` package to plot the frequencies of each biopsy site on a human outline. Note, that since the package is picky about which organs we can use, we must once again manually create a data frame with relevant "organs" for the package's plotting function and manually include the values (in this case, the frequency of the biopsy site):

```
# Create data frame for plotting frequencies of biopsy site
biopsies <- data.frame(organ = c("lymph_node", "prostate", "liver", "penis",
                                 "bone","heart","stomach","urinary_bladder"),
 value = c(61, 2, 18, 1, 2, 43, 1, 1), stringsAsFactors = TRUE)

# Plot frequency of biopsy sites
p2 <-gganatogram(data = biopsies, organism="human", sex="male",
 fill="value",  fillOutline = "#E7F6FF") + theme_void() +
 ggtitle("Frequency for each Biopsy Site") + coord_fixed() +
  scale_fill_gradient2(low = "grey", mid = "white", high = "red",
                       midpoint = .02)
p2
```



Now that we have created two human plots – one for mutation count severity by biopsy site, and one for frequency of each biopsy site from our data set, we can compare the two side-by-side:

```
#Compare tumor content severity for biopsy site to frequency for each biopsy site
lay <- rbind(c(1,2), c(1,2))
grid.arrange(temp_works, p2, layout_matrix = lay)
```



Now that we can compare severity of tumor content directly next to frequency of biopsy site, we can visually assess any possible correlation between the two. After comparing the two side-by-side, we can see that there does not seem to be any association between tumor content severity at a biopsy site and how frequently that biopsy site was sampled in this data set. And with that, we can complete our analysis for Aim 2!

BONUS AIM

The last aim checks for an association between diagnosis age and tumor site. According to the Cleveland Clinic 1 in 6 men will be diagnosed with prostate cancer during the course of their lifetime. Research from the American Cancer Society suggests that 6 out of 10 cases of prostate cancer are diagnosed among men 65 years or older. As mentioned previously the mean age of diagnosis among this prospective cohort was 67 years of age. Prior research suggests that as men get older their risk of being diagnosed with prostate cancer increases. In addition, the publication in Cell suggests that bone tumors are the most common site of metastatic disease. This aim therefore intends to determine whether these two prior notions hold true among the data in question.

To conduct the analysis we subset the data columns Diagnosis.Age and Tumor.Site into a dataframe called age_tumor. The numeric index of these columns was 4 and 11 respectively. Before proceeding with any analysis we conducted an exploratory investigation to understand the quantity of unique tumor sites present in the data set along with the spread of age at diagnosis.

```r
age_tumor <- remove_nonuniques[c(4,11)] #subset the data
head(age_tumor)
```

```
##   Diagnosis.Age Tumor.Site
## 1            54       Bone
## 2            49 Lymph node
## 3              78 Lymph node
## 4              71 Lymph node
## 5              67 Lymph node
## 6              63       Bone
```

```
summary(unique(age_tumor$Tumor.Site)) #look at unique values for tumor.site
```

```
##    Length     Class      Mode
##        20 character character
```

According to the aforementioned summary and unique command, 20 unique tumor sites are present within this data. Next, we explored the distribution of diagnosis age to determine min, max, mean and median ages for this prospective prostate cohort.

```
summary(age_tumor$Diagnosis.Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   40.00   62.00   68.00   66.84   72.00   85.00
```

Given the minimum age at diagnosis was 40 and the maximum age at diagnosis was 85 we needed to come up with a way to age-stratify the data in order to compare the various tumor sites with age at diagnosis. TAn article published in the British Journal of Cancer described a methodology for stratifying age within a population based study investigating prostate cancer. The researchers stratified age according to the following groups "40-59", "60-69", "70-79" and "80+ years". Since this was a validated study we employed the same age-stratified bins for our analysis.

Using the `tidyverse` package we employed several `ifelse` statements to group the ages into the various aforementioned age strata. All values were grouped into their respective bins with the final clause indicating that values that did not fall into these specified ranges should be classified as `NA`.

```
library(tidyverse) #load tidyverse to assist in analysis
#grouping the age by the BJC age-stratified bins using ifelse statements
age_tumor$agegrp <- ifelse(age_tumor$Diagnosis.Age >= 40 &
                              age_tumor$Diagnosis.Age <=59, "40-59",
                  ifelse(age_tumor$Diagnosis.Age > 59 &
                            age_tumor$Diagnosis.Age <= 69, "60-69",
                         ifelse(age_tumor$Diagnosis.Age > 69 &
                                   age_tumor$Diagnosis.Age <= 79, "70-79",
                                 ifelse(age_tumor$Diagnosis.Age >79,"80+", NA))))
```

After stratifying the data, we created a summary table utilizing the packages `dplyr` and `gtsummary` to assess the aggregated counts of tumors by site grouped by age.

```
library(dplyr) #load dplyr to use gtsummary
#install.packages("gtsummary") #in order to create the graph you may need to
# install the gtsummary pkg
library(gtsummary) #load gtsummary to create the summary table
age_tumor %>%
  select(Tumor.Site, agegrp) %>%
  #develop the table based on the selected input
  tbl_summary(by = agegrp, label = Tumor.Site ~ "Tumor Site") %>%
  add_p(test=everything()~"kruskal.test") #test for the differences among ranks
```

```
## Table printed with `knitr::kable()`, not {gt}. Learn why at
## http://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html
## To suppress this message, include `message = FALSE` in code chunk header.
```

Characteristic	**40-59**, N = 27	**60-69**, N = 54	**70-79**, N = 59	**80+**, N = 7	**p-value**
Tumor Site					0.9
Abdominal wall mass	0 (0%)	0 (0%)	1 (1.7%)	0 (0%)	
Bladder	0 (0%)	1 (1.9%)	0 (0%)	0 (0%)	
Bone	8 (30%)	17 (31%)	16 (27%)	2 (29%)	
Chest Wall	0 (0%)	1 (1.9%)	0 (0%)	0 (0%)	
Dura	1 (3.7%)	0 (0%)	0 (0%)	0 (0%)	
glans penis	1 (3.7%)	0 (0%)	0 (0%)	0 (0%)	
Liver	3 (11%)	8 (15%)	6 (10%)	1 (14%)	
Lymph node	9 (33%)	19 (35%)	31 (53%)	2 (29%)	
Muscle	0 (0%)	1 (1.9%)	1 (1.7%)	0 (0%)	
Neck Mass	0 (0%)	1 (1.9%)	0 (0%)	0 (0%)	
Pelvic mass	0 (0%)	1 (1.9%)	0 (0%)	0 (0%)	
Penile	0 (0%)	1 (1.9%)	0 (0%)	0 (0%)	
Perirectal	0 (0%)	1 (1.9%)	0 (0%)	0 (0%)	
Peritoneal nodule	1 (3.7%)	0 (0%)	0 (0%)	0 (0%)	
Prostate	2 (7.4%)	0 (0%)	0 (0%)	0 (0%)	
ProstateTURP	0 (0%)	0 (0%)	1 (1.7%)	0 (0%)	
Psoas Mass	0 (0%)	1 (1.9%)	0 (0%)	0 (0%)	
Retroperitonium	1 (3.7%)	0 (0%)	0 (0%)	0 (0%)	
Soft Tissue	1 (3.7%)	2 (3.7%)	2 (3.4%)	2 (29%)	
Thoracic epidural	0 (0%)	0 (0%)	1 (1.7%)	0 (0%)	

As evidenced by the output of the table, several of the aggregate counts of tumors by site were either 0 or 1 in most cases, which is not that significant. Five sites however did demonstrate counts greater than 1 and those include, Bone, Liver, Lymph Node, Prostate and Soft Tissue. To assess whether the differences among the age stratified ranks were significant by tumor site we conducted a Kruskal-Wallis Rank Sum Test. The output of which can be found in the last column of this table. The value of p was reported as 0.9 and assuming an alpha significance level of 0.05 this value is not significant. Thus, we conclude that there is no significant difference exhibited among the age stratified ranks by tumor site.

We found the quantities of the aggregate counts for the following tumor sites: Bone, Liver, Lymph Node, Prostate and Soft Tissue to be interesting so we further subsetted the data to explore the distribution of tumors. Additionally, we wanted to further assess whether bone, as mentioned earlier, was indeed the site of greatest metastatic disease.

To do this we grouped the data by age group and tumor site to create a count of tumors by age. Then we filtered for those cases for which the frequency of tumors was greater than 1. Using the `ggplot2` package we created a clustered bar plot to compare the distribution of those five tumors by age group. The graph was formatted with the `viridis` package to create the color scheme displayed below. Axes labels and legend titles were added as appropriate.

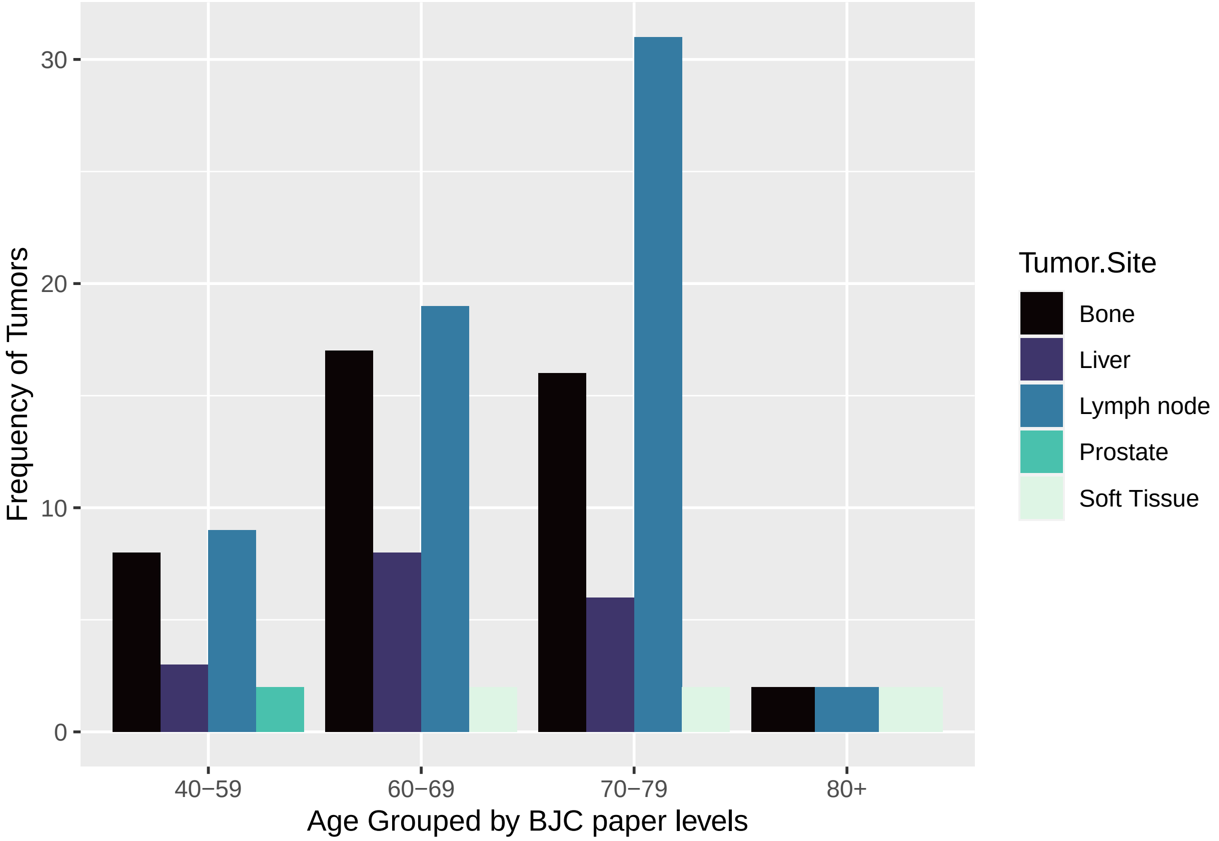
```
#calculate the frequency of tumors by age group
freq <- age_tumor %>% group_by(agegrp,Tumor.Site) %>% summarise(Freq=n())
```

```
## `summarise()` has grouped output by 'agegrp'. You can override using the `.groups` argument.
```

```
freq <- rename(freq, "Count of Tumors"=Freq) #rename the frequency column
#subset the data to include the five columns of interest
freq_gt1 <-freq[which(freq$`Count of Tumors`>1),]
library(ggplot2) #load in ggplot to create graphical summary
library(viridis) #load in viridis for pretty color palette
```

```
## Loading required package: viridisLite
```

```
#create a bar chart of the data
ggplot(freq_gt1, aes(agegrp,`Count of Tumors`)) +
  labs(x="Age Grouped by BJC paper levels", y="Frequency of Tumors") +
  geom_bar(aes(fill=`Tumor.Site`),stat="identity",position="dodge") +
  scale_fill_viridis_d(option  = "mako")
```



Curiously enough, bone was not the most common site of metastatic disease among this prospective prostate cohort. In fact, the lymph nodes were shown to have a greater quantity of tumors amounting to 61 tumors across the entire cohort as opposed to 43 among bone. Here from this graph we can also see that the "60-69" and "70-79" groups experience the most tumors proportionally, as would be expected given that as men age the likelihood of developing prostate cancer increases. If time permitted we would have further calculated the incident rate ratios per age group for the five tumor sites where frequency was greater than 1 to understand the proportional burden of disease experienced by each age group.

Overall, this aim confirms that as men age incidence of prostate cancer increases with metastasis of disease occurring more frequently in the lymph nodes and bone. This data was limited given that the population was a bit skewed toward the older age groups (60-79) with very few cases representing those older than 80. More information as to prognosis and vital status beyond age 80 would be beneficial in determining whether the assumption of age and incidence is valid.

Natural Language Processing

The opportunities to employ natural language processing (NLP) in this dataset were limited, but the principles and tools of NLP were utilized to extract more information from the data. The second data, the GDC portal clinical data, had four columns corresponding to the AJCC staging classification of each patient in the data set. To help better understand what these classifications meant, we added a new column that included a written description of each of the AJCC staging classifications provided. To do this, we used Regex to identify AJCC stage patterns to match them to their written description. The written descriptions of each stage come from this webpage.

In order to analyze the data with the NLP principles we used the package `stringr`. First we subsetted the data into the four relevant columns, `ajcc_clinical_m`, `ajcc_clinical_t`, `ajcc_pathologic_n`, and `ajcc_pathologic_t`. This was saved to the dataframe `ajcc cols`. Each column corresponds to a clinical assessment of a particular site in relation to the clinical stage of cancer or prognosis.`ajcc_clinical_m` relates to the extent of the distant metastases for the cancer while `ajcc_clinical_t` relates to the extent of the primary cancer. `ajcc_pathologic_n` describes the stage of the cancer based on the nodes and `ajcc_pathologic_t` describes the stage of the cancer based on the primary tumor.

```r
library(stringr)
library(tidyverse)
sample <- read.csv("~/Desktop/Data-Wrangling-Project/cleanedsample.csv")
SampleRevised<-read.csv("~/Desktop/Data-Wrangling-Project/SampleRevised.csv")
staging_dat <- sample
ajcc_cols <- staging_dat %>% select(c("ajcc_clinical_m", "ajcc_clinical_t",
                                      "ajcc_pathologic_n", "ajcc_pathologic_t"))
```

Next we combined the four aforementioned classifier columns into one string. We removed mention of the string placeholder, '–, used to indicate missing values by employing the `str_remove` function. This caused some of combined strings to only contain three values rather than four.

```r
#Combine classifiers into one string
ajcc_cols$combined <- str_c(ajcc_cols$ajcc_clinical_m,
                            ajcc_cols$ajcc_clinical_t,
                            ajcc_cols$ajcc_pathologic_n,
                            ajcc_cols$ajcc_pathologic_t, sep = " ")

#Take out effective NAs
ajcc_cols$combined <- str_remove(ajcc_cols$combined, "('--)++")
```

After combining the strings we created a vector of the `classificiations` as described in the AJCC staging classifications website link provided above. Each classification accounts for how the cancer was presented in the lymph nodes or elsewhere in the body. These values `N0` and `M1` shown in the descriptions below are available in our dataset from the `ajcc_clinical_m` and the `ajcc_pathologic_n` columns.

```r
#Classification descriptions (from website given above)
classifications = c("The cancer has not spread to nearby lymph nodes or
                    elsewhere in the body. The Grade Group is 1 , and the PSA
                    level is less than 10.",
                  "The cancer has not spread to nearby lymph nodes [N0] or
                  elsewhere in the body [M0]. The Grade Group is 1.
                  The PSA level is at least 10 but less than 20.",
                  "The cancer has not spread to nearby lymph nodes [N0] or
                  elsewhere in the body [M0]. The Grade Group is 2. The PSA
                  level is less than 20.",
                  "It has not spread to nearby lymph nodes [N0] or elsewhere in
                  the body [M0]. The Grade Group is 1 to 4, and the PSA can be
                  any value.",
                  " The cancer has spread to nearby lymph nodes [N1] but has not
                  spread elsewhere in the body [M0]. The Grade Group can be any
                  value, and the PSA can be any value.",
                  "It has spread to other parts of the body, such as distant
                  lymph nodes, bones, or other organs [M1]. The Grade Group can
                  be any value, and the PSA can be any value.",
                  "Not classified")
```

We then created a vector of the classifications for the clinical and pathologic information contained in our dataset that corresponded to each unique classification given above.

```
#Patterns to identify classification from AJCC info
regs = c("M0.T[0|2a].*N0", "M0.T[12][abc].*N0", "M0.T[12].N0", "M0.T[34].N0",
         "M0.T.*N1", "M1.T.*N.*", ".*")
```

Next, we grouped the data according to each of the numeric classifications assigned in the previous step. To do this we used the `which` function along with the `grepl` function to find the position or index for which the classification of interest was true. For example, the first group `M0.T[0|2a].*N0` defined in the `regs` vector above, was used to classify 222 of the individuals. The logical positions from the `ajcc cols` combined data for the first five individuals were 1,2,13,14,15.

```
#Label row indices with groups
group1psa10 <- which(grepl(regs[1], ajcc_cols$combined))
group1psa20 <- which(grepl(regs[2], ajcc_cols$combined))
group2psa20 <- which(grepl(regs[3], ajcc_cols$combined))
groups14 <- which(grepl(regs[4], ajcc_cols$combined))
nodegroup <- which(grepl(regs[5], ajcc_cols$combined))
metastizedgroup <- which(grepl(regs[6], ajcc_cols$combined))
nogroup <- which(grepl(regs[7], ajcc_cols$combined))
```

Once grouping of the data based on the numeric classifications was complete we merged the text description of the AJCC clinical stage from `classifications` with the `ajcc cols` data frame. This added a sentence description that interpreted the numeric clinical stage. In the output below notice how the sentences are structured.

```
#Create a column containing group description
ajcc_cols$AJCC_Stage_Description <- ajcc_cols$combined

ajcc_cols$AJCC_Stage_Description[nogroup] <- classifications[7]
ajcc_cols$AJCC_Stage_Description[metastizedgroup] <- classifications[6]
ajcc_cols$AJCC_Stage_Description[nodegroup] <- classifications[5]
ajcc_cols$AJCC_Stage_Description[groups14] <- classifications[4]
ajcc_cols$AJCC_Stage_Description[group2psa20] <- classifications[3]
ajcc_cols$AJCC_Stage_Description[group1psa20] <- classifications[2]
ajcc_cols$AJCC_Stage_Description[group1psa10] <- classifications[1]

#Lets view the first 6 classification descriptions.
head(ajcc_cols$AJCC_Stage_Description)
```

```
## [1] "The cancer has not spread to nearby lymph nodes or \n                    elsewhere in the body.
## [2] "The cancer has not spread to nearby lymph nodes or \n                    elsewhere in the body.
## [3] "Not classified"
## [4] "Not classified"
## [5] " The cancer has spread to nearby lymph nodes [N1] but has not\n                  spread elsewhe
## [6] " The cancer has spread to nearby lymph nodes [N1] but has not\n                  spread elsewhe
```

```
#Add this column to rest of data
staging_dat$AJCC_Stage_Description <- ajcc_cols$AJCC_Stage_Description
```

This type of NLP processing methodology described above is very useful in automating the translation of exome and transcriptome numeric sequencing of the cancer genes into something digestible for the general public. Very few people understand what `M0` means with regards to a prognosis of prostrate cancer disease. As evidenced by this analysis, `M0` indicates that the cancer had no spread elsewhere in body. Practical applications of this particular NLP methodology may include sharing sequencing results with affected individuals through a patient portal, informing the care team of results through the electronic medical record or automating documentation.

Extract the Data from National Cancer Institute

We downloaded the second cancer data set from the National Cancer Institute GDC Data Portal for all the clinical cases including all the columns.

The link can be found here.

Excel Cleaning for Second Data Set

The GDC portal website provided both a cleaned and uncleaned version of their prostrate cancer data. The cleaned data mirrors what can be found on their visualization dashboard describing 493 patients with 24 data points while the uncleaned version contains many more clinical parameters such as clinical stages of the tumor, days to last follow up, and treatment. Both GDC datasets are more robust in their capture of the clinical parameters related to prostate cancer in contrast with the cBioPortal data previously analyzed. This indicates that since 2015 when the Cell paper was published, the observation and recording of tumor data related to prostrate cancer has significantly increased allowing researchers to provide more insight into the manifestation of disease. For this portion of our analysis we downloaded both datasets as we were interested in wrangling the data and discerning the overlap between the two datasets. We first started by cleaning the clinical data set with the abundance of parameters.

The data was downloaded as a TSV file so it needed to be imported into Excel to change the format to a CSV or .XLSX file. The raw data contained 960 rows and 156 columns. Upon further investigation it was evident that several of the columns were empty or had values missing. In order to determine the columns with no data present along with those which had missing values we used the Filter tool in Excel. Missing values were those marked as ’– in the cells. We developed a key for the missing values called “Key for Missing Values Columns”.

Based on the assessment where we developed the key of the missing values we determined that the raw data contained 112 empty columns for which no data was available and 33 columns with missing values. The 112 empty columns were then removed from the analysis by simple deletion in Excel and a new sheet was created, “Sample”. This new sheet “Sample” contained 44 columns.

Further investigation of the new “Sample” data indicated that some rows were shifted over particularly rows 141, 142, 206, 207, 284, 285, 494 and 495. We pinpointed the origin of the shift for the aforementioned rows to column Y. In order to account for this shift we copied the data from the “Sample” worksheet to a new sheet called “Sample Revised”. Here we inserted a cell in column Y for each of the aforementioned rows to shift the values of the cells to the the right so that the cells would be associated with the correct column.

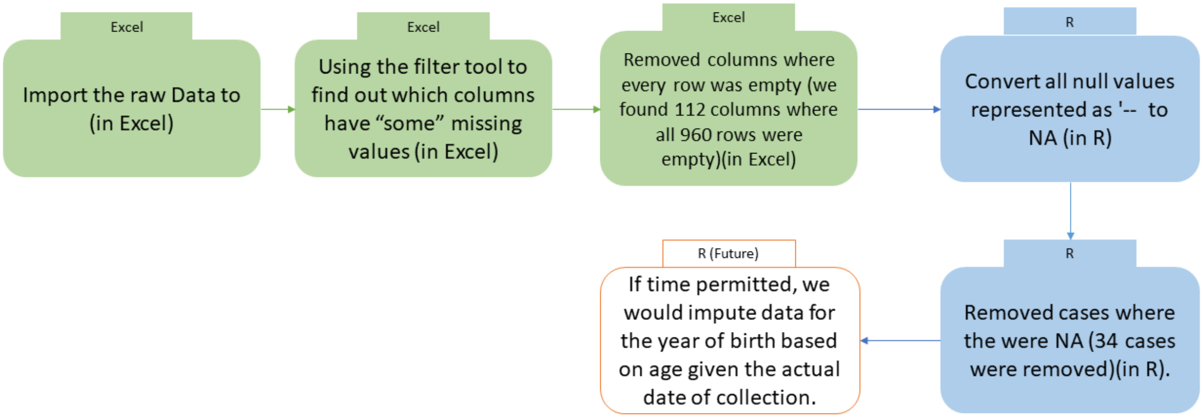
Similarly the “treatment_type” column values were shifted to the right but also broken up into values of “Radiation Therapy/Pharmaceutical Therapy” and “NOS”. To fix this issue we concatenated the values for the therapy and NOS in a new column, column AS. We then named this column “treatment_type”.

After concatenating the values we copied the data from “Sample Revised” to a new worksheet called “cleaned_sample”. In this dataset we noticed that after shifting the rows five columns were shown to be empty. Thus we removed the five columns: secondary_gleason_grade, synchronous_malignancy, tumor_grade, year_of_diagnosis and treatment_or_therapy. We exported this data as a CSV file to then continue our analysis of the missing values in R.

For our final Excel file with the metadata and the various sheets please refer to our GitHub repository. You will find the .xlsx file under the title `clinical data from NCI`.

Missing Value Pipeline

The following flowchart details the “Missing Value Pipeline” which outlines the sequence of steps taken to address the missing values present in the “uncleaned” version of the GDC portal data. Some of the steps have been previously discussed as they were done as part of the cleaning of the dataset in Excel. The remainder relating to “R” will be discussed in further detail below.


Excel
Import the raw Data to (in Excel)
Excel
Using the filter tool to find out which columns have “some” missing values (in Excel)
Excel
Removed columns where every row was empty (we found 112 columns where all 960 rows were empty)(in Excel)
R
Convert all null values represented as '-- to NA (in R)
R
Removed cases where the were NA (34 cases were removed)(in R).
R (Future)
If time permitted, we would impute data for the year of birth based on age given the actual date of collection.

Dealing with Missing Values

As mentioned previously, the GDC portal “uncleaned” data set contained 33 columns with missing values. To be able to adequately analyze the data we need to deal with these missing value strings. First we began by exporting our cleaned data sample as a csv file from Excel. Then subsequently we imported the data into R.

```r
sample <- read.csv("~/Desktop/Data-Wrangling-Project/cleanedsample.csv")
```

Next, we recorded the string values representative of missing values to NAs using a custom function that searched for each value of ’– and replaced it with `NA`. The function was applied to the sample data set imported above and saved to a new data frame called `temp.NA`.

```r
#changing '-- Null values to NA
string_NA <- function(data){
data[data == "'--"] <- NA
return(data)
}

temp.NA <- string_NA(sample)
```

A cursory scan revealed that the strings were indeed changed to NA values within the dataset. However it was apparent that across several rows there were `NA` values present concurrently across various columns. To address this issue we therefore subsetted the data conditionally using the subset function based on whether `NA` values were present simultaneously across all three of the following columns:`classification of tumor`, `last known disease status`and `tumor largest dimension diameter`.

```r
#subsetting data to remove specific rows
temp.NA1 <- subset(temp.NA, temp.NA$classification_of_tumor != "NA" &
                     temp.NA$last_known_disease_status !="NA" &
                     temp.NA$tumor_largest_dimension_diameter!="NA")
dim(temp.NA)
```

```
## [1] 960  40
```

```r
dim(temp.NA1)
```

```
## [1] 934  40
```

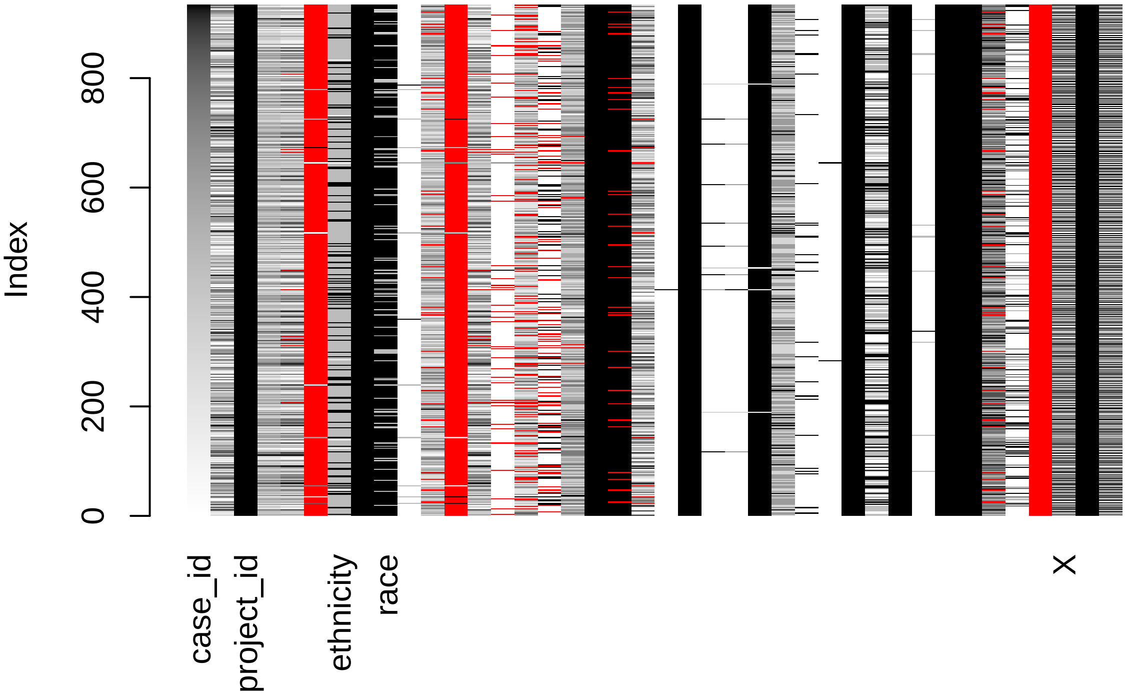
BY subsetting the rows from the `temp.NA` dataframe we removed 26 rows in which `NA` values were present across the three afroementioned columns. The `NA` values that are now left are column dependent rather than case dependent. Several `NA` values still remain however simply removing the columns is not effective given that those values that are populated within the column are important. For instance the majority of the `days to death` column is populated with `NA` values however lines 131 and 132 have the value `146` for the days of death. This is important information in determining how many days elapsed between diagnosis and death.

To further visually explore the relationship between the missing variables and their respective CaseIDs we created a matrixplot using the package `VIM`. This graph shows black and white variables indicating the values present in each column followed by red demarcations indicating the missing variables. Notice the three red columns highlighted here. They correspond to `days to death`, `year of death`, and `treatment type`. Many columns towards the middle of the graph also seem to highlight several missing values. Those columns correspond to ratings for the clinical cancer stage of tumors, regional nodes and metastases. As demonstrated by this graph those three columns `days to death`, `year of death`, and `treatment type` uniformly are shown to have missing values however the clinical cancer stage columns intermittently illustrate missing values relative to each respective CaseID.

```r
library(VIM)
```

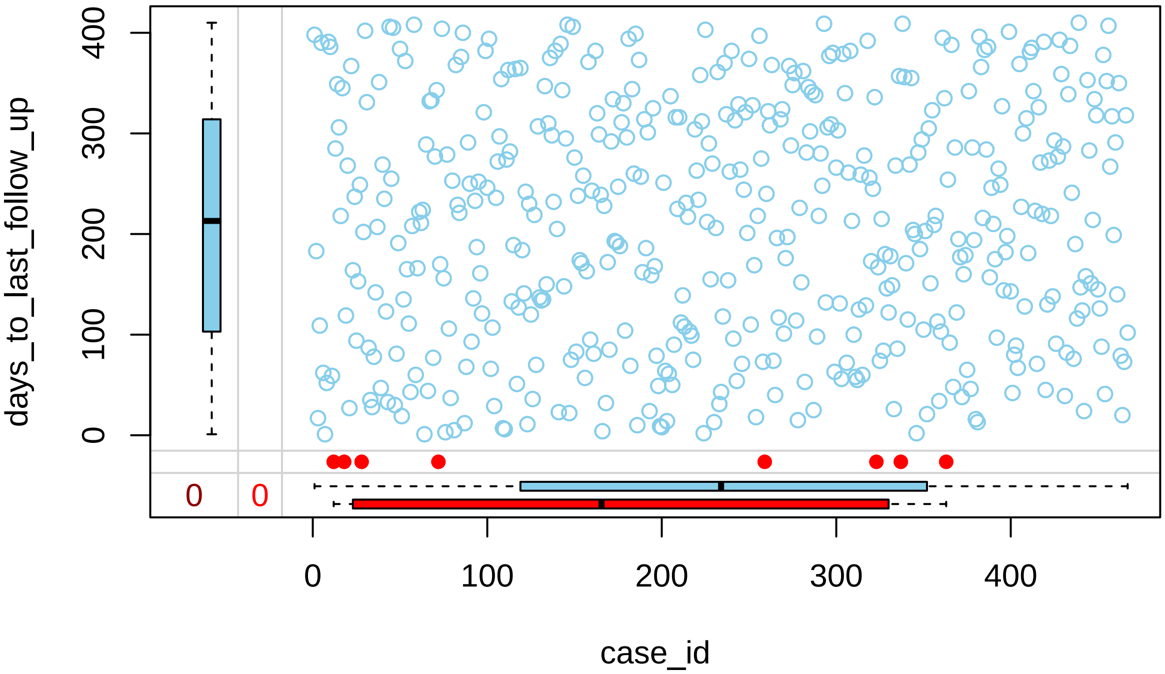
```
## Loading required package: colorspace
## Loading required package: grid

## VIM is ready to use.

## Suggestions and bug-reports can be submitted at: https://github.com/statistikat/VIM/issues

##
## Attaching package: 'VIM'

## The following object is masked from 'package:datasets':
##
##     sleep
```

```
matrixplot(temp.NA1,sortby=1)
```



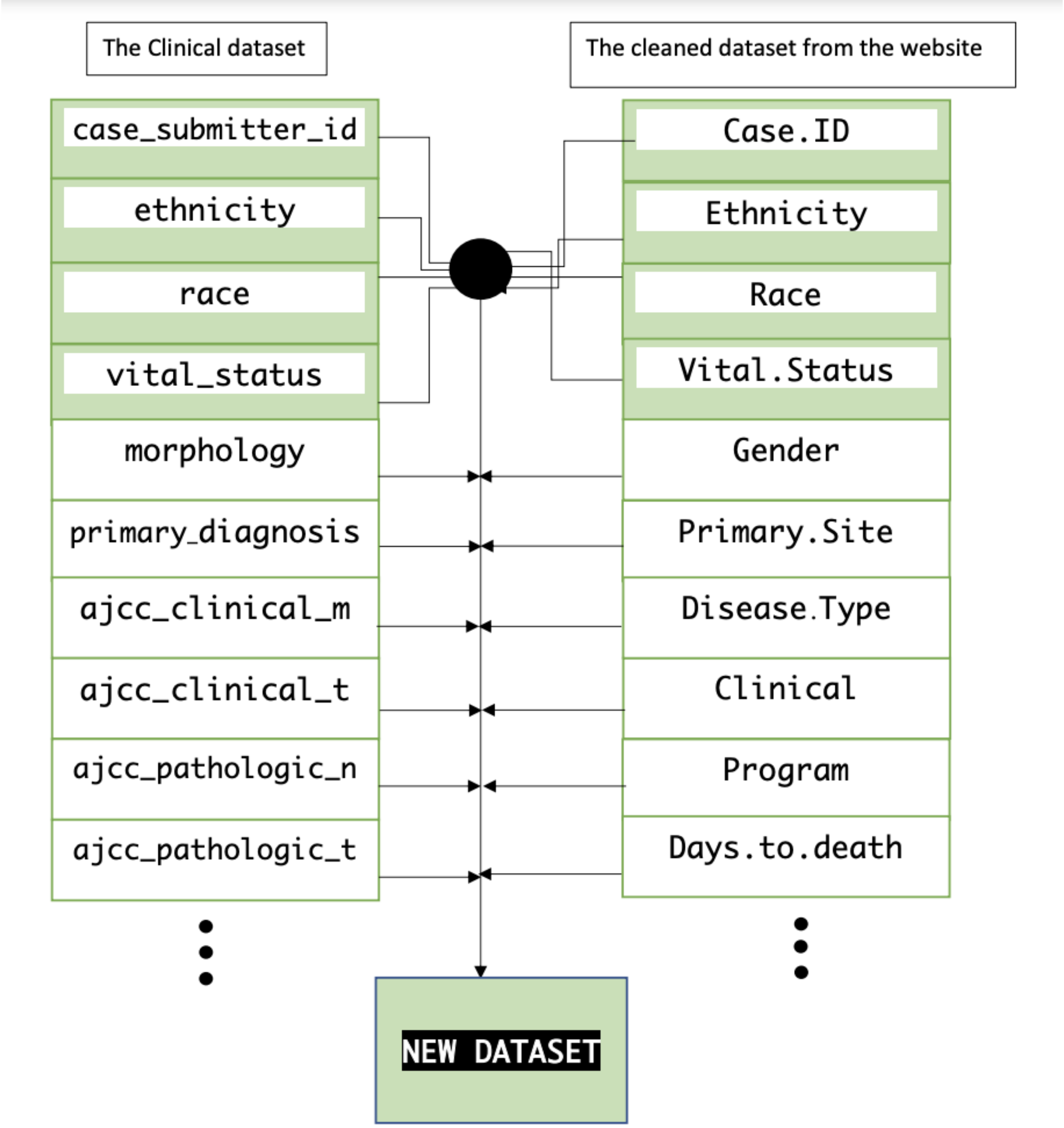
We further investigated how one of the columns illustrated above with the missing values compared to CaseID. Using the `VIM` package again we created a margin plot to observe how the missing values in `days to last folluw up` (column 20) compared with each case. As depicted below, the column had 16 missing values with a fairly random scatter for the observed values shown in blue. The red values however seem to be concentrated towards the earlier and latter cases with the distribution being skewed for the red boxplot, in contrast to the blue boxplot for CaseID.

```
marginplot(temp.NA1[,c(1,20)])
```



Relational Data

With the data for the GDC portal clinical data set cleaned, we wanted to compare apples to apples and determine just how different our newly cleaned clinical data set was from the already clean website version. The following diagram illustrates how the clinical data set is interconnected with the already clean version from the website. Highlighted below it is apparent that four of the columns present in the clinical data set serve as foreign keys for the cleaned data set. These columns include `case submitter id`, `ethnicity`, `race`, and `vital status`. We will utilize this relationship to join the data in subsequent steps.



SQL to JOIN and APPEND DATA

Although the two data sets do share similarities one stark difference was the number of observations contained in each. The clinical data set cleaned by the DiseaseDivas contained nearly 960 observations while the cleaned version taken directly from the GDC portal website only contained 493. `What might be contributing to this discrepancy?`

After sorting the data by `Case Submitter ID`, we learned that several of the `case submitter Id's` in the cleaned clinical dataset were duplicates due to columns AJ and AN corresponding to `treatment outcome` and `treatment type`. For our analysis treatment outcome and treatment type were not relevant. Thus, to account for this discrepancy in the data observations we copied the data from the `cleaned sample` sheet in

Excel and created a new sheet called `removed dups`. In this sheet we removed the columns that contained the split version of the therapy and NOS which corresponded to columns AK-AN. Additionally we removed columns AJ and AN then proceeded to use the `Remove Duplicates` tool in Excel.

Once this cleaning in Excel was complete we were left with 493 observations of 39 variables for the clinical dataset. The lengths of the two datasets therefore aligned. To confirm that this was indeed true we imported the data into R and compared both data sets using the `dim` function.

```
#install.packages("sqldf")
library(sqldf)
```

```
## Loading required package: gsubfn

## Loading required package: proto

## Loading required package: RSQLite
```

```
library(tidyverse)
new_sample <- read.csv("~/Desktop/Data-Wrangling-Project/removed_dups.csv")
new_sample <- subset(new_sample, select=-c(36:39))
dim(new_sample)
```

```
## [1] 493  35
```

```
GDC <- read.csv("~/Desktop/Data-Wrangling-Project/Dataset2DD.csv")
GDC <- rename(GDC, case_id=Case.UUID)
dim(GDC)
```

```
## [1] 493  24
```

The clinical data that we cleaned was loaded into R as `new sample`. The columns which we had removed in Excel,AJ-AN, unfortunately still appeared but with `NA` values contained in each row. We removed these columns using the subset function. The clinical dataset then contained 493 observations of 35 data points while the cleaned data contained 493 observations of 24 data points. The lengths match!

After confirmation that the lengths of the two datasets matched we began to conduct our SQL relational data comparative analysis. Ultimately the goal of this analysis was to relate findings across the cBioPortal data against the GDC portal data. Although the variables greatly differed between the two data sources two things that both datasets did have in common were the `Diagnosis Age` and the `Prior Treatment`. We therefore decided to group the diagnosis age by the age stratified bins previously used as per the BJC paper.

We started with the first dataset utilizing the dataframe `remove nonuniques` for which the repeat values had been removed. Next we renamed the `Patient Id` and the `Diagnosis Age` column so that they were easier to call in the SQL syntax. Using the `sqldf` package we subsetted just the `Patient Id` and the `Diagnosis Age` and employed a `case when` statement similar to an ifelse statement in R to group the ages by their respective bins.This was saved to the dataframe `dataset 1 sql`. The name given to this newly created age group column was the case when statement therefore we changed the name to `agegrp` using the `names` function. We then appended this data in `dataset 1 sql` to the original dataframe using the `JOIN` syntax. This appended version of the data was saved to `join data1 sql`. To ensure that the ages did indeed match the new groupings we selected the `Age` column and the `agegrp` column and conducted a quick investigation utilizing the `head` function. Voila! As the output demonstrates, the age stratification in SQL was indeed successful.

```
dataset_1 <- remove_nonuniques
dataset_1 <- rename(dataset_1, ID=Patient.ID)
dataset_1 <- rename(dataset_1,Age=Diagnosis.Age)
dataset_1sql <- sqldf("SELECT ID, case when Age >=40 and Age <= 59 then
'40-59' when Age >59 and Age <= 69 then '60-69'when Age >69 and Age <= 79 then
'70-79' else '80+' end  FROM dataset_1 GROUP BY ID,
                      case when Age >=40 and Age <= 59 then '40-59' when
                      Age >59 and Age <= 69 then '60-69'when
                      Age >69 and Age <= 79 then '70-79'
                      else '80+' end")
names(dataset_1sql)[2]<-"agegrp"
join_data1_sql <- sqldf("SELECT * FROM dataset_1 JOIN dataset_1sql on
                        dataset_1.ID=dataset_1sql.ID")

#age check
age_ds1 <- select(join_data1_sql,"Age","agegrp")
```

With the first data source formatting complete we then moved on to the second data source, the GDC portal data. Here a few more steps were employed given that we had to join the clinical dataset to the cleaned website version of the data in order to compare it as a whole to the cBioPortal data. Thus, we first joined the clinical dataset to the cleaned website data based on the foreign key for the `Case Id`. The combination of the two datasets produced a large dataset of 493 observations with 59 data points.

```
joined_data <- sqldf("SELECT *
                     FROM new_sample JOIN GDC on
                     new_sample.case_id=GDC.case_id")
dim(joined_data)
```

```
## [1] 493  59
```

Fifty-nine data points is A LOT of data. Before proceeding with our SQL additions we perused this new data sample. We noticed many columns for which the same value was contained within every row of the column so we employed the same `find nonuniques` custom function from our earlier cleaning of cBioPortal to remove the columns where the data was repeated.

```
repeat_cols <- find_nonuniq(joined_data)
```

```
## [1]  8 31 33 39 40
```

```
remove_dups <- joined_data[,-repeat_cols]
dim(remove_dups)
```

```
## [1] 493  54
```

From 59 columns we then reduced to 54 columns. Still yet we had duplicate columns present in the data but of a different kind. These columns were the foreign keys present in the two datsets: `case submitter id`, `ethnicity`, `race`, and `vital status`. To account for this duplication we employed the use of a new custom function called `carbon copy`. This function looks through the dataframe and removes the columns that are duplicates of each other. It employs base R syntax using the `!` to facilitate the match.

```
new_updatedremovedups <- carbon.copy(remove_dups)
dim(new_updatedremovedups)
```

```
## [1] 493  45
```

After employing the use of the `carbon copy` function we were left with 45 columns in the new dataframe `new updated remove dups`. Investigation of the data confirms that the duplicate foreign keys were indeed removed from the second half the data, from columns 33 onwards in `removed dups`. The first occurrence of the foreign key still remains so no information was lost as part of this function.

Before we could add the age group stratification we needed to perform a bit of natural language processing, specifically text mining. `Age at Diagnosis` as per the cleaned website version of the GDC portal data was presented as a `character` or string value, `63 years 262 days`. In order for us to be able to use this data to create our age stratification we needed to extract the 2 numeric digits at the beginning of the string relating to age in years. To do this we used the `substr` function which extracts substrings in a character vector. We set the `start` argument as the first position and the `stop` argument as the second position in order to capture the two digits necessary. Then, we merged this vector using the `cbind` function to the `new updated remove dups` dataframe. Unfortunately the value was still saved as a character so we had to change the format of the variable to numeric using the `as.numeric` function. Some `NA` values were introduced as part of this function which were applicable given that the data for `Age at Diagnosis` was missing in those cases.

```
z <- substr(new_updatedremovedups$Age.at.diagnosis, 1,2)
head(z)
```

```
## [1] "63" "58" "47" "70" "61" "68"
```

```
new_updatedremovedups$Age <- cbind(z)
new_updatedremovedups$Age <- as.numeric(new_updatedremovedups$Age)
```

```
## Warning: NAs introduced by coercion
```

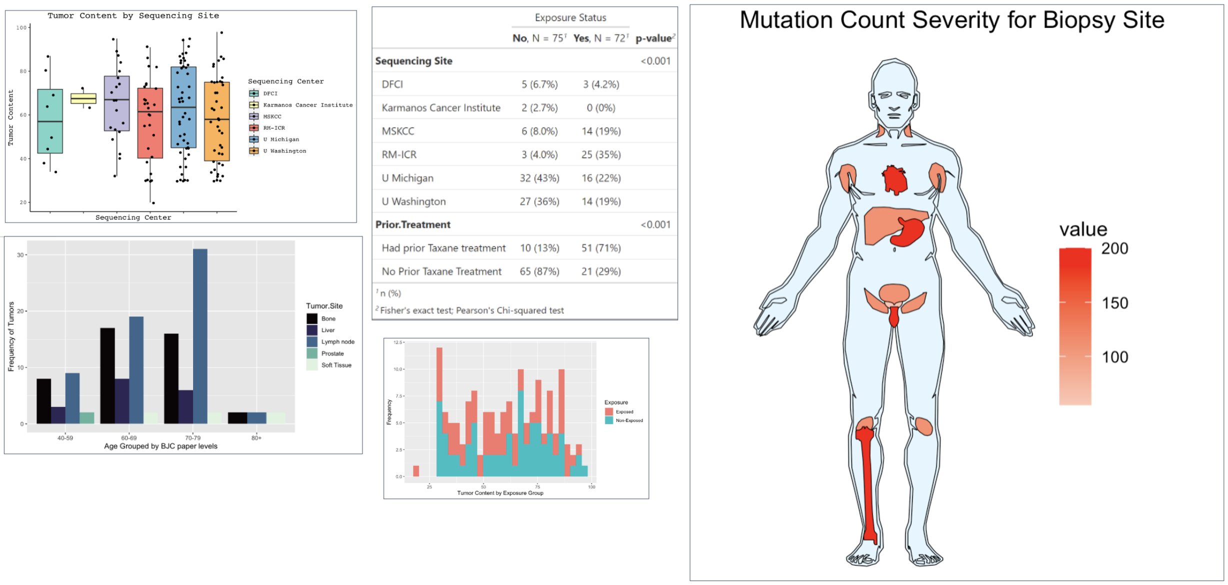
Lastly, we performed the age stratification. In order to run the SQL syntax as done previously we renamed the `Case ID` column to `ID` for ease of use in the arguments. We employed the same methodology as before using `case when` statements to group the ages by the BJC paper age bins. Finally we merged the data from the subset for age stratification to the combined GDC `dataset 2`.

```
new_updatedremovedups <- rename(new_updatedremovedups, ID=case_id)
dataset_2 <- new_updatedremovedups
dataset_2sql <- sqldf("SELECT ID, case when Age >=40 and Age <= 59 then
'40-59' when Age >59 and Age <= 69 then '60-69'when Age >69 and Age <= 79
then '70-79' else '80+' end  FROM dataset_2 GROUP BY ID,
                      case when Age >=40 and Age <= 59 then '40-59' when
                      Age >59 and Age <= 69 then '60-69'when
                      Age >69 and Age <= 79 then '70-79'
                      else '80+' end")
names(dataset_2sql)[2]<-"agegrp"
join_data2_sql <- sqldf("SELECT * FROM dataset_2 JOIN dataset_2sql ON
                        dataset_2.ID=dataset_2sql.ID")
```

At this point both the cBioPortal and GDC datasets have been outfitted with age stratified bins facilitating greater analysis of the clinical parameters in relation to age. Furthermore the clinical dataset along with the website version of the GDC data have been merged to create one cohesive dataset. This relational data comparative analysis provides great insight into the discrepancies apparent in the data and the similarities exhibited by the datasets. In addition the methods employed can aid in subsequent research down the line in comparing each dataset to each other to garner insights into the burden and progression of metastatic prostate cancer.

Dashboard

Lastly, we couldn't let our beautiful analyses go to waste so we created a simple dashboard in Excel of some of our more notable and significant graphs from this prostate cancer analysis.



	Exposure Status		
	No, N = 75[1]	**Yes**, N = 72[1]	**p-value**[2]
Sequencing Site			<0.001
DFCI	5 (6.7%)	3 (4.2%)	
Karmanos Cancer Institute	2 (2.7%)	0 (0%)	
MSKCC	6 (8.0%)	14 (19%)	
RM-ICR	3 (4.0%)	25 (35%)	
U Michigan	32 (43%)	16 (22%)	
U Washington	27 (36%)	14 (19%)	
Prior.Treatment			<0.001
Had prior Taxane treatment	10 (13%)	51 (71%)	
No Prior Taxane Treatment	65 (87%)	21 (29%)	

[1] n (%)

[2] Fisher's exact test; Pearson's Chi-squared test

CONCLUSION

Summary

Our analysis aimed to understand the associations between patient characteristics and severity of prostate cancer through various data wrangling procedures we learned throughout the quarter. After utilizing three platforms for our cleaning and wrangling processes (R, SQL, and Excel), and creating numerical summaries along with visual representations of our data through R packages such as gtsummary and ggplot, we have come to the following conclusions:

1. After performing a Chi-squared test on the relationships between geographic location and severity of tumor content (%), the data set provides no evidence of association between geographic location and severity of tumor content (%).
2. After running a Fisher's exact test (for Sequencing Site) and a Pearson's Chi Squared test (Prior Treatment Status), there is evidence of a statistically significant association between Sequencing Site and Exposure Status along with Prior Treatment Status and Exposure Status.
3. As men age, incidence of prostate cancer increases with metastasis of disease occurring more frequently in the lymph nodes and bone.
4. From visual assessments, there does not seem to be an association between exposure status and mutation group, fraction of genome altered, or tumor content.

Next Steps

After joining the two data sets from GDC via SQL, we noticed that our initial data set from cBioPortal and the new data set both had data on patient's age at diagnosis as well as whether the patient had prior treatment. As a basic step to check similarities between patients across both data sets, we plotted the distributions of age at diagnosis for both data sets.

```
# Plot age at diagnosis distributions for both data sets
conc_plot1 = ggplot(data.frame(dataset_2), aes(x=Age)) +
  geom_histogram(colour="black",fill="lightskyblue") +theme_classic()

summary(dataset_2$Age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   41.00   56.00   61.00   61.05   66.00   85.00       9
```

```
conc_plot2 = ggplot(data.frame(dataset_1), aes(x=Age)) +
  geom_histogram(colour="black",fill="lightsalmon") + theme_classic()
summary(dataset_1$Age)
```

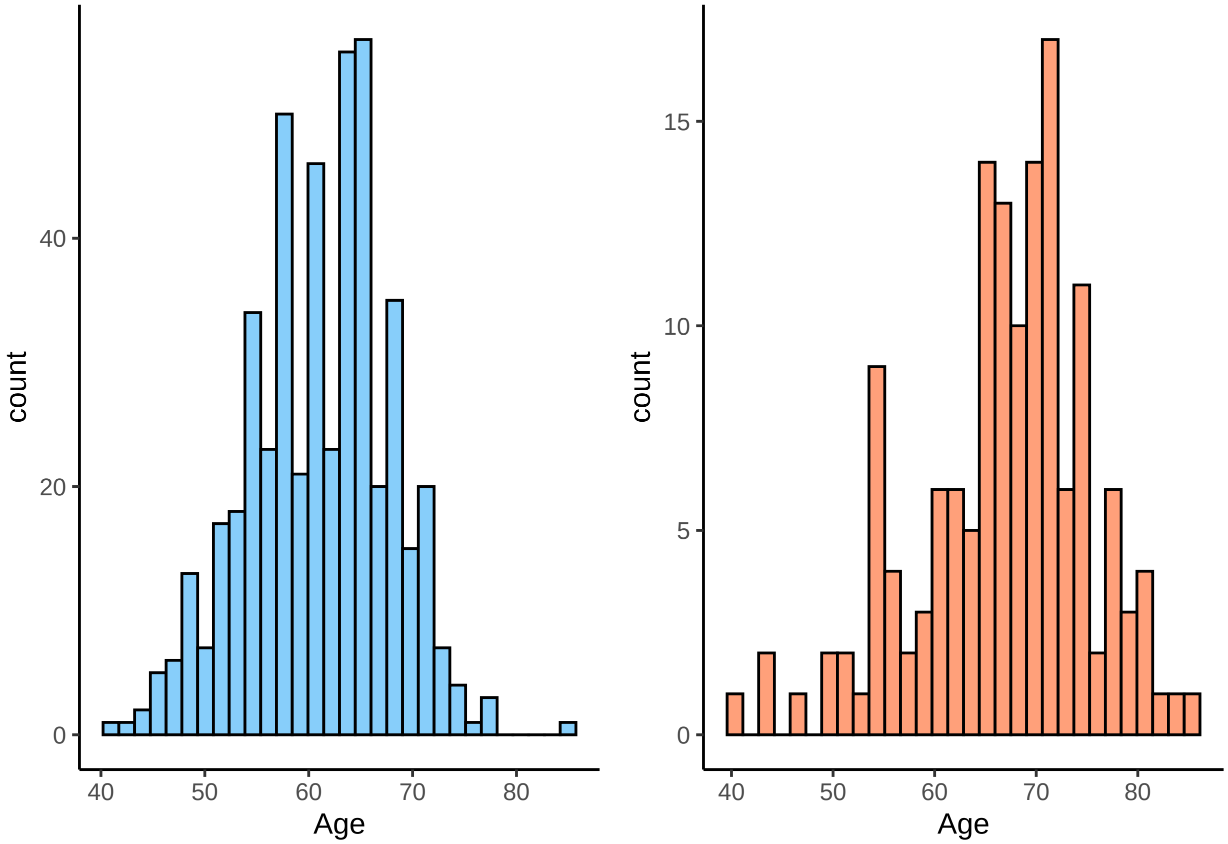
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   40.00   62.00   68.00   66.84   72.00   85.00
```

```
lay <- rbind(c(1,2), c(1,2))
grid.arrange(conc_plot1, conc_plot2, layout_matrix = lay)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 9 rows containing non-finite values (stat_bin).
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



After assessing the plots above, we noticed that the mean age at diagnosis for our first data set (cBioPortal) was slightly later than the mean age at diagnosis for our second data set (GDC). After realizing this, we wanted to see if patients from the cBioPortal had higher rates of prior treatment status than patients from the GDC data. In order to check this while stratifying for age groups, we plotted prior treatment status for patients from each data set based on our age groups.

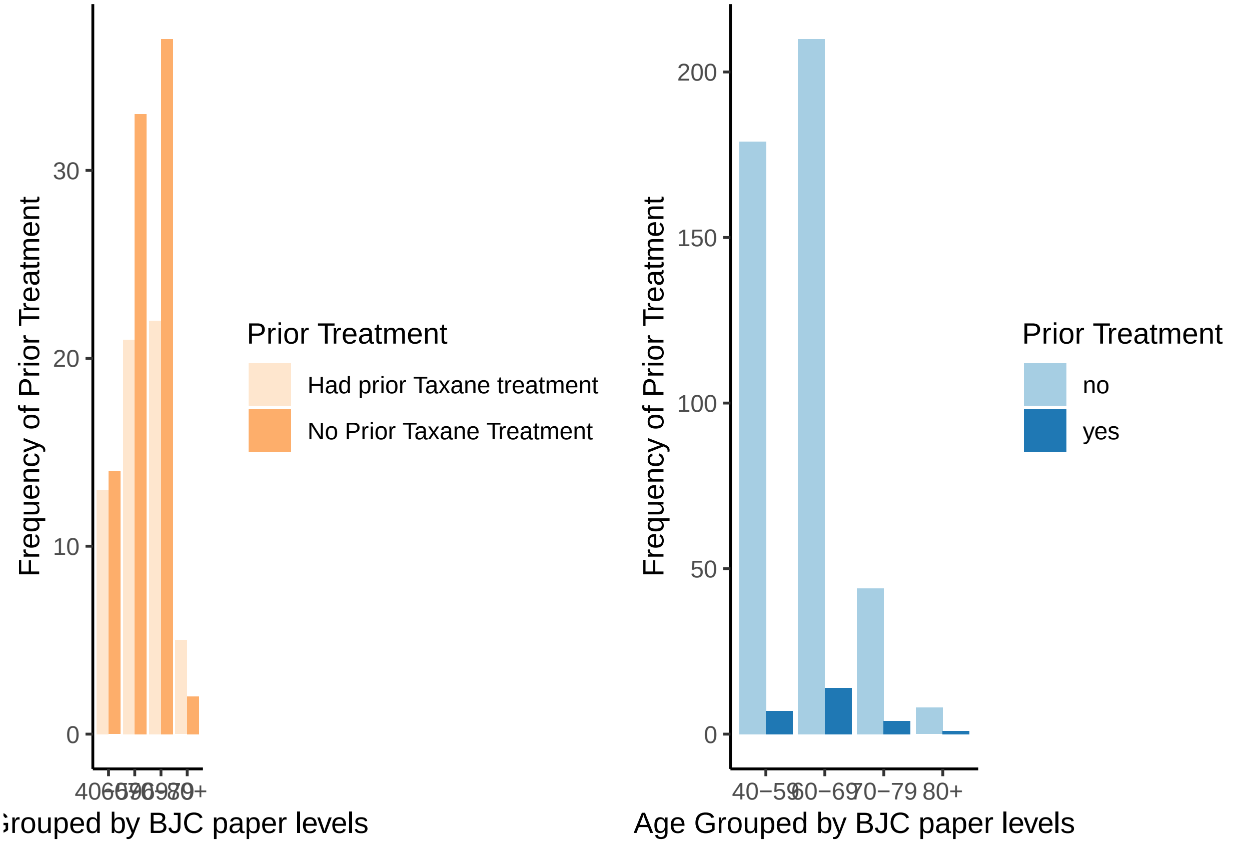
```
# Plot prior treatment status for both data sets
join_data1_sql[1] <- NULL
freq_1treat <-join_data1_sql %>% group_by(agegrp,Prior.Treatment) %>%
  summarise(Freq=n())
```

```
## ‘summarise()‘ has grouped output by ’agegrp’. You can override using the ‘.groups‘ argument.
```

```
dat2 <- dataset_2
join_data2_sql[join_data2_sql == "'--"] <- NA
join_data2_sql <- join_data2_sql[!is.na(join_data2_sql$prior_treatment),]
join_data2_sql[1] <- NULL
freq_2treat <- join_data2_sql %>% group_by(agegrp,prior_treatment) %>%
  summarise(Freq=n())
```

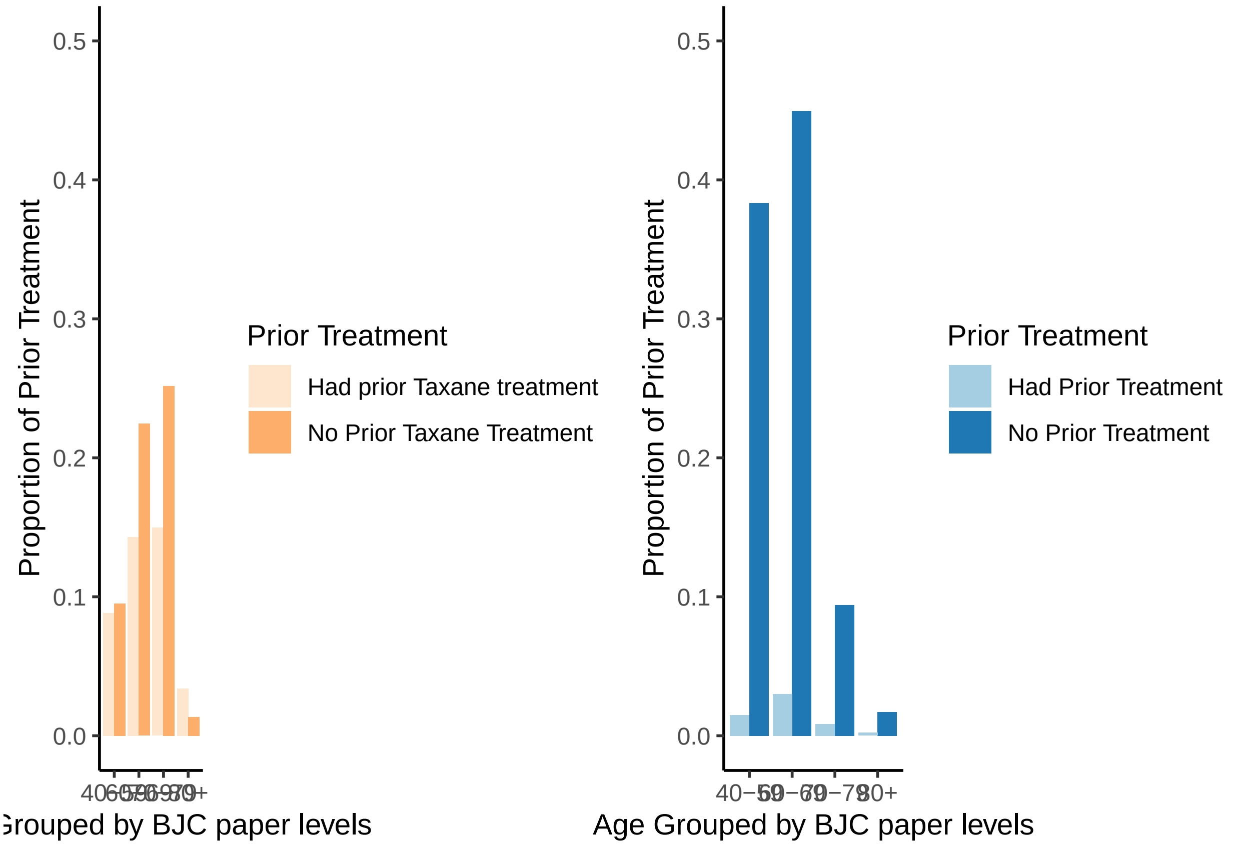
```
## ‘summarise()‘ has grouped output by ’agegrp’. You can override using the ‘.groups‘ argument.
```

```
conc_hist1 = ggplot(data.frame(freq_1treat), aes(x=agegrp, y=Freq)) +
  labs(x="Age Grouped by BJC paper levels", y="Frequency of Prior Treatment") +
  geom_bar(aes(fill=Prior.Treatment),stat="identity",position="dodge") +
  guides(fill=guide_legend(title="Prior Treatment"))+
  scale_fill_brewer(palette="Oranges") +
  theme_classic()


conc_hist2 = ggplot(data.frame(freq_2treat), aes(x=agegrp, y=Freq)) +
  labs(x="Age Grouped by BJC paper levels", y="Frequency of Prior Treatment") +
  geom_bar(aes(fill=prior_treatment),stat="identity",position="dodge") +
  guides(fill=guide_legend(title="Prior Treatment"))+
  scale_fill_brewer(palette="Paired") +
  theme_classic()

lay <- rbind(c(1,2), c(1,2))
grid.arrange(conc_hist1, conc_hist2, layout_matrix = lay)
```



The first thing we noticed is that these graphs are slightly misleading since we are looking at frequencies instead of proportions. The cBioPortal data set frequencies range from 0-40, while the GDC data set frequencies range from 0 to >200. Let's try to make these graphs more comparable.

```
# Make proportions
freq_1treat$Prop = freq_1treat$Freq / sum(freq_1treat$Freq)

# Make proportions
freq_2treat$Prop = freq_2treat$Freq / sum(freq_2treat$Freq)

# Fix colors
freq_2treat$UpdatedPT <- ifelse(freq_2treat$prior_treatment == 'yes',
                                'Had Prior Treatment', 'No Prior Treatment')



conc_hist3 = ggplot(data.frame(freq_1treat), aes(x=agegrp, y=Prop)) +
  labs(x="Age Grouped by BJC paper levels", y="Proportion of Prior Treatment") +
  geom_bar(aes(fill=Prior.Treatment),stat="identity",position="dodge") +
  guides(fill=guide_legend(title="Prior Treatment"))+
  scale_fill_brewer(palette="Oranges") +
  theme_classic() + ylim(0, 0.5)

conc_hist4 = ggplot(data.frame(freq_2treat), aes(x=agegrp, y=Prop)) +
  labs(x="Age Grouped by BJC paper levels", y="Proportion of Prior Treatment") +
  geom_bar(aes(fill=UpdatedPT),stat="identity",position="dodge") +
  guides(fill=guide_legend(title="Prior Treatment"))+
  scale_fill_brewer(palette="Paired") +
  theme_classic() + ylim(0, 0.5)

lay <- rbind(c(1,2), c(1,2))
grid.arrange(conc_hist3, conc_hist4, layout_matrix = lay)
```



Great! Now that we have the proportions and equal y axis ranges, we can see that the gap between having prior treatment versus no prior treatment is relatively smaller in the patients from the first data set compared to patients from the second data set. Additionally, the proportion of patients who received prior treatment in the first data set seems to be higher across all age groups compared to the second data set.

This would be a great place to start further analysis if we had more time. We could look more into how prior treatment affects diagnosis age, if prior treatment has been linked to later diagnosis age, and where we would be able to find more data to test this association. One thing to note is that the first data set's patients were specifically studied on prior Taxane treatment, while the patients in the second data set did not have a specific treatment mentioned. So, we could also look at Taxane treatment effectiveness in preventing prostate cancer versus other treatments. In addition to investigating diagnosis age's relationship to prior treatment status, we had also noticed that the second data set had a "vital status" column which gives us information on the survival status of the patient. This information was missing from our first data set, so another great next step would be to track the incidence of disease in our second data set (from GDC), and by using our vital status column, we would calculate the patient mortality ratio for the GDC data set.

Limitations

Though we were able to derive several conclusions from our data sets, there were a few limitations we faced. For starters, the data set we got from cBioPortal did not have well-defined variables or information on certain variables were calculated, what the units of measurements were, etc. For example, we initially struggled to find information on how tumor content and mutation count were measured, what the units of measurements were, and what ranges of each measurement were considered significant. When we tried to do some background research through search engines such as Google, we weren't able to find any helpful information, which leads us to our second limitation: prostate cancer data is very niche, and there were not many resources readily available for us to use to get a better understanding on some of the variables we were dealing with. Nevertheless, after reading through the data set's attached paper for understanding tumor content and looking at other academic papers to better understand how mutation count works, we were able to successfully interpret our variables of interest.

Looking Forward - Clinical Implications

If we were to use our analysis to make conclusions about the clinical implications of the prostate cancer data we analyzed throughout this project, our first piece of advice would be to further study associations between sequencing site and exposure status, along with prior treatment status and exposure status since we obtained significant results when studying these associations. Additionally, it would be worth looking into how age plays into certain factors such as prior treatment status and tumor site.