Geographical Analysis of Patient Data in North America for the Testing of Protein PD-L1

Niall Tyndall

Introduction

Loading the Data

We start this report by loading and preparing the data set.

```r
# Load the data set
my.data <- pdata::data_reader()

# Get some map data...
map.data <- my.data %>%
  subset(my.data$Client.Zip %in% c("", "Moroc") %>% `!`())

# Convert ZIP codes to integers
map.data$Client.Zip %<>% as.integer
```

[1]This report outlines a brief investigation of 100581 rows of patient data for the testing of protein PD-L1, spanning over a date range of 384 days. The majority of this data set has been collected from Northern America with a few exceptions (after some pre-analysis), such as the country of Peurto Rico, and other unknown zip codes from the **Client.Zip** column header. With this in mind, the format of this report is designed to group all the data geographically to investigate further and drill down specific metrics by this specific *grouping* method. A minor assumption is considered in order to cast the data into such a form, therefore data that has **no** well defined region from the ordering physician is neglected. It is possible to go through a process of estimating the error due to these undefined regions by considering the ratio of tests to states and during the analysis we could incorporate this error, although it should be negligible enough to avoid this task. This leads to the question, how many records are *thrown* away from the analysis? In total, there are 286 ZIP codes that could not be determined, and as such, have been provided a tag of *unknown* and are not considered during the analysis. Finally, 17 could not be determined at all, resulting in a total of 303 missing entries (or 0.3%).

Preparing the Data

Below is a quick summary of some of the numerical data involved in the full dataset.

```r
# Generate some summary statistics of numerical data
summary.data <- my.data %>% subset(
  select = c(
    "Test.Ordered.Days.From.Specimen.Collected",
    "Test.Reported.Days.From.Specimen.Collected",
    "Tat"
  )
)
```

[1] Throughout this report, data columns are represented in **bold face**, R variables are represented as `camelCase verbatim`, functions are represented as `snake_case() verbatim`, and data sets are represented as `dot.separated verbatim`.

```r
names(summary.data) <- c("Ordered.Days", "Reported.Days", "Tat")
summary(summary.data)
```

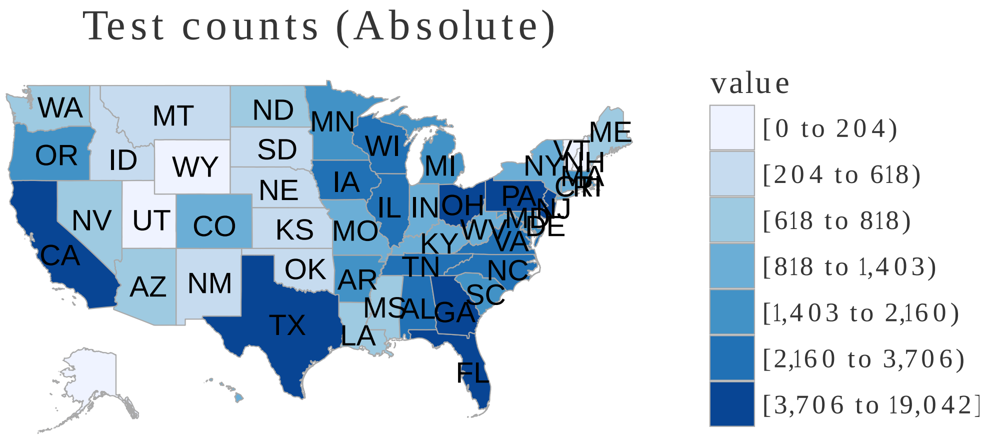
```
##   Ordered.Days       Reported.Days          Tat
##  Min.   :-67537.0   Min.   :-67534.0   Min.   :-308.000
##  1st Qu.:      5.0   1st Qu.:      8.0   1st Qu.:   2.000
##  Median :     13.0   Median :     17.0   Median :   3.000
##  Mean   :    134.3   Mean   :    137.8   Mean   :   3.519
##  3rd Qu.:     46.0   3rd Qu.:     50.0   3rd Qu.:   4.000
##  Max.   : 42687.0   Max.   : 42690.0   Max.   : 176.000
##  NA's   :3          NA's   :1367       NA's   :1367
```

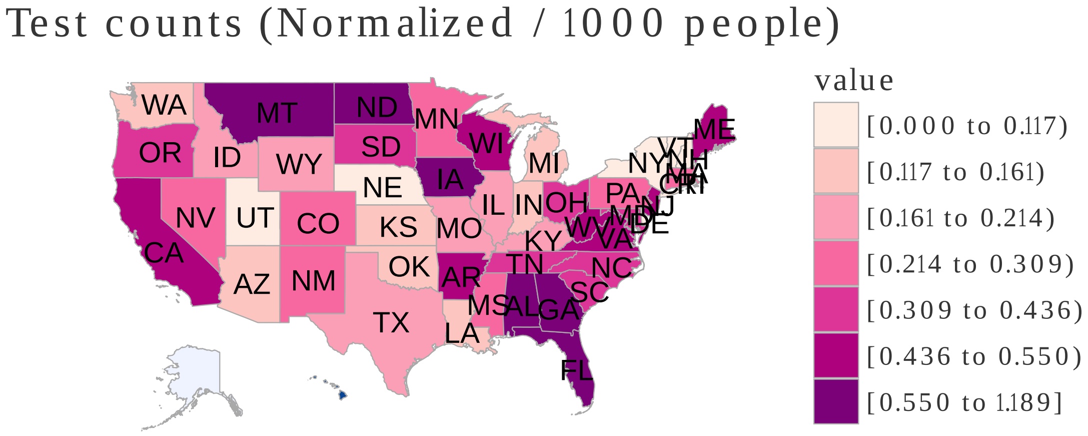
```r
# Compute unique rows containing NA's
uniqueNA <- summary.data$Reported.Days %>% is.na %>%
  `|`(summary.data$Tat %>% is.na) %>%
  `|`(summary.data$Ordered.Days %>% is.na) %>%
  sum
```

This leads to a total of 1367 unique rows that contain `NA` values as provided by `uniqueNA` above.

The data set that is used in the analysis is therefore almost identical to the original set, with every record having an additional piece of information, a mapping of the zip code to a US State. As shown below, the zip codes are provided as input to the function `create_map()`, which maps them to a US state through the open source package *zipcode*.

```r
# Get ZIP code hash and create map
mapInfo <- map.data$Client.Zip %>%
  pdata::create_map()
```





```r
# Append actual states to the main data set
map.data$state <- sapply(
  X = map.data$Client.Zip,
  FUN = function(x) x %>% mapInfo$find()
  )
```

From here on in, I will now use the new data set of `map.data` which has is an augmented version of `my.data`. A hashmap lookup of zip codes (keys) to states (values) are also saved, and stored in `zipInfo`, which is then used to append this information through the vectorized *sapply* approach. This function also generates the American map of individual test counts, where I have generated one for absolute counts and also counts that are normalized per state population per 1000 people. Generally, the two maps are quite similar, however the most stark difference is the flip between the central American states after normalization, such as Texas, Montana, and North Dakota. While the coastal counties of California, Florida, and Georgia (generally the entire east coast) remaining quite dominant even after normalization.

The remainder of this report will now consider investigations into the data set provided, using this geographical approach. There are four subsections that follow in the Results section, and a brief summary is provided in the final section, called Conclusions.

Results

In this section I will outline various ideas and provide some basic functionality in the form of code snippets, utilizing the augmented data set, called `map.data`. All functionality will accept this data frame object as input and manipulate it to consider analysis into individual states.

1) Methodologies

After some basic analysis, we can see clearly that the four antibodies used as a diagnostic all use the same methodology; *IHC*, representing 92.2% of the total data set.

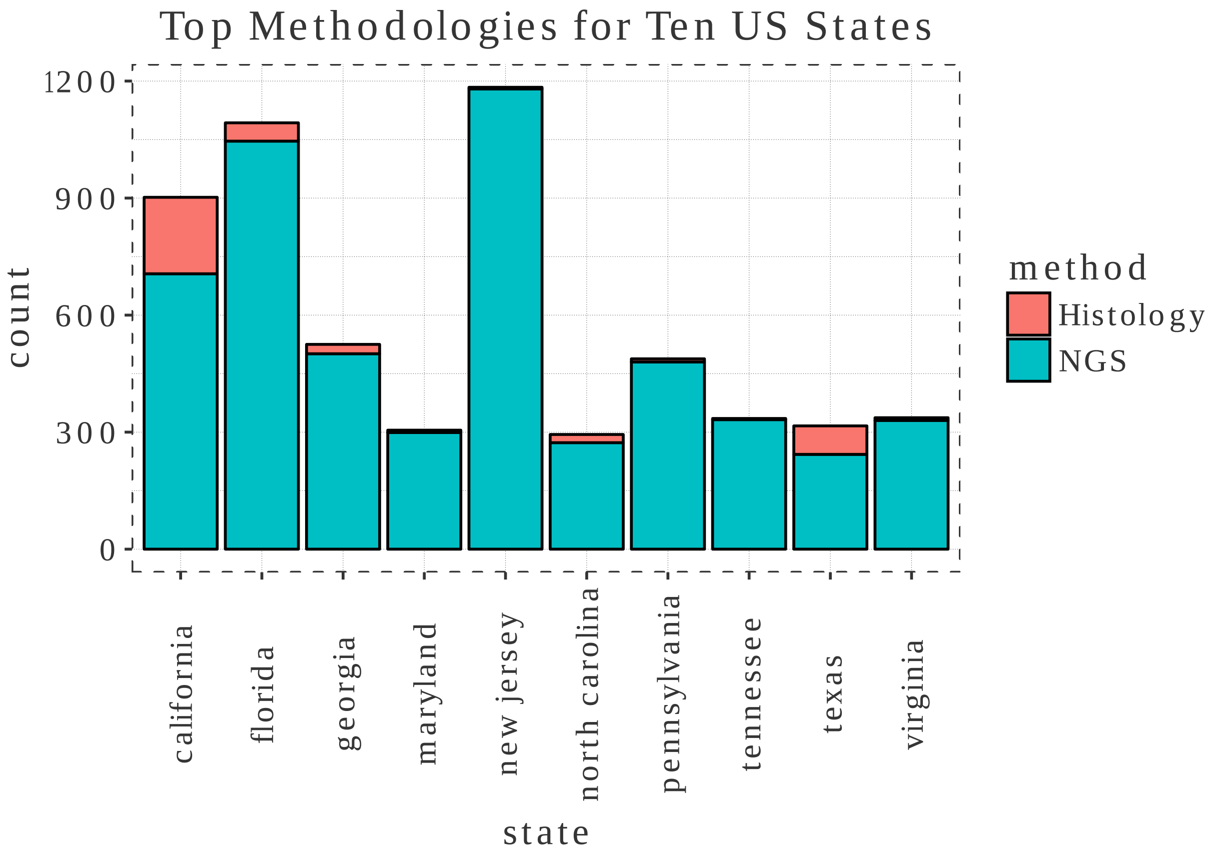
```r
# Create a summary of the Methodologies
summaryTable <- map.data$Methodology %>%
  table %>%
  data.frame

# Get percentage
summaryTable$Per <- summaryTable$Freq %>%
  `/`(map.data %>% nrow) %>%
  `*`(100) %>%
  round(2)

names(summaryTable) <- c("Methodology", "Count", "Per")

knitr::kable(
  x = summaryTable,
  align = "c",
  caption = "A table containing the Methodology types and percentages of the total data set."
)
```

Table 1: A table containing the Methodology types and percentages of the total data set.

Methodology	Count	Per
Histology	779	0.77
IHC	92761	92.24
NGS	7024	6.98

PD-L1 is also tested using 2 other methodologies, defined here by Histology and Next-generation sequencing (NGS). The snippet below provides the function call `method_plot()` to generate this stacked bar chart and is stored in `mResults`. This bar chart depicts the remaining two methodologies for the ten highest state counts.

```r
# Return results from top methodologies per state
mResults <- map.data %>%
  pdata::method_plot()

mResults
```



In taking this approach we can see the differences and magnitudes between these methodology uses, without being swamped by the remaining 92.2% IHC method. It is clear from the chart that New Jersey and Florida (East coast) have the highest usage of NGS methods, whereas California and Texas have a higher percentage use of Histology (West coast). It is possible to further this analysis by considering test specifics for these methods as opposed to IHC by looking at the anatomical site the sample was removed from - as an example. This approach provides a layer, in general, to look at specific features within the context of method being carried out in the test.

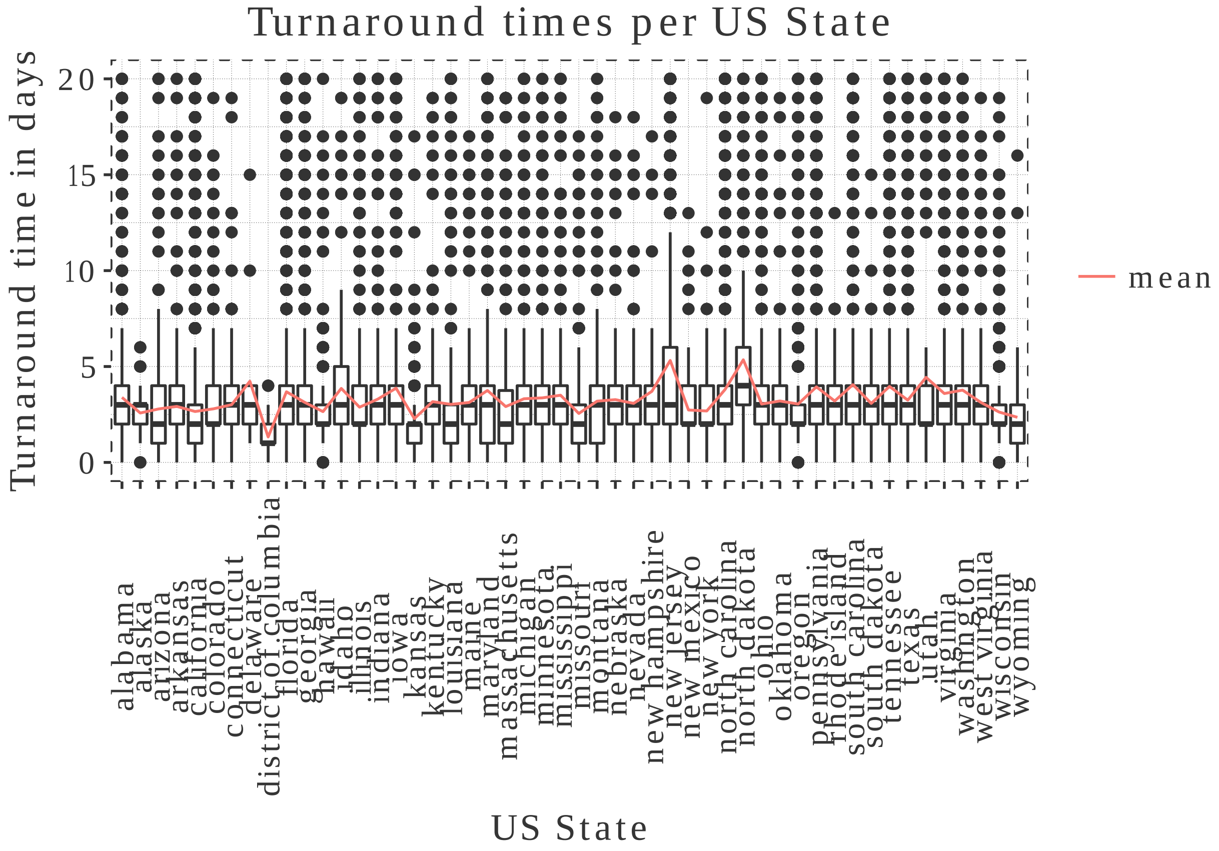
2) Turnaround times

The next investigation revolved around looking at the turnaround times recorded for each test on a US state basis. Nothing overly complex was carried out with the data here as the times have been recorded under the column **Tat**, although more effort could be spent to provide some error on the results of the boxplots. No data cleansing of such was carried out, as no preliminary evidence to support this would have enhanced the results. One such solution could have involved analysing each state in turn, and then predicting the missing `NA` values between the number of days between result reported and number of days between test order date from the specimen collection date. In other words, either calculating averages on the data set available and updating `NA`'s with the median/mean value, or looking at a time series approach and investigating the surrounding x days and fitting a polynomial spline to estimate that value. However, due to time constraints, none of this was implemented and more than likely, would have made little difference to the results. Below is a snippet for the state of Texas where we plot the time series data and their turnaround times to estimate the single `NA` value in this data set as 15.96 (which turns out to be approximately the value of μ).

```r
# Look at the time series of Texas
texas <- map.data %>%
  subset(map.data$state %>% `==`('texas'))

# Get x-axis values (timestamps)
allDates <- texas$Month..Day..Year.of.Estimated.Ordered.Date %>%
  as.Date(format = '%d-%b-%y') %>%
  as.integer

# Get the time indexes for the ordered time series
indexes <- allDates %>%
  order

# Get ordered days (ordered / sorted)!
orderedDays <- texas$Test.Ordered.Days.From.Specimen.Collected %>%
  `[`(indexes)

# Pick up a missing value (only 1 [= 1888])
missing <- orderedDays %>%
  is.na %>%
  which

# Now get the surrounding timestamp
surroundingInd <- c(
  (missing - 100):(missing - 1), c(missing + 1):(missing + 100)
)
orderedDays %<>% `[`(surroundingInd)

# Define x axis with one free in the middle where NA is
xAxis <- c(c(1:100), c(102:(surroundingInd %>% length %>% `+`(1))))

# Remove possible outliers (set to 100 also)
chopVals <- orderedDays %>%
  `>`(100)

# Calculate mean without the outliers
unChoppedMean <- orderedDays %>%
  `[`(chopVals %>% `!`()) %>%
  mean

# Replace outliers with calculated mean
orderedDays[chopVals] <- unChoppedMean

# Set up data frame for linear regression + prediction
new.data <- data.frame(
  x = xAxis,
  y = orderedDays,
  stringsAsFactors = FALSE
)

# Make the prediction
linearPred <- predict(lm(y ~ x, data = new.data), data.frame(x = 101)) %>%
  as.double
```

The next snippet below shows a call to `reporting_time()` with the `map.data` set, returning a boxplot for every US state, recording the aggregated statistics of turnaround time in days.

```r
# Create turnaround time boxplots
rResults <- map.data %>%
  pdata::reporting_time()

rResults
```



The extent of the whiskers here is not the typical 100%, but instead is set to 95% to visualise some of the outliers from the quartiles. The plot has also been cut at twenty days to inspect the boxplots closer. Most of the data is in fact relatively similar across each state, with nothing really catching the eye. I have included a summary function of the mean overlaid with the boxplots in red to analyse the typical averages per state. While most median values lie around the 3 or 4 day turnaround period, the states of North Dakota and New Jersey are slightly higher, around 6 days turnaround and also a much larger deviation from the widths of their respective boxplots. In contrast, the District Of Columbia has the lowest avergae turnaround time, at just 2 days with a compact boxplot, which emphasises that the results vary little.

```r
# Return large turnaround times (over 50)
tat.anom <- map.data %>% subset(
  map.data$Tat %>% `>`(50),
  select = c(
    "Test.Order.Id", "Body.Site", "Specimen.Type", "Methodology", "Tat", "state"
  )
)

# Return JUST the order ID
tat.anom$Test.Order.Id %<>%
```

```
  strsplit(split = ',') %>%
  purrr::map(1) %>%
  as.character

knitr::kable(
  x = tat.anom[tat.anom$Tat %>% order %>% rev, ],
  row.names = FALSE,
  align = "c",
  caption = "Possible anomalous values of turnaround times that are above 50 days."
)
```

Table 2: Possible anomalous values of turnaround times that are above 50 days.

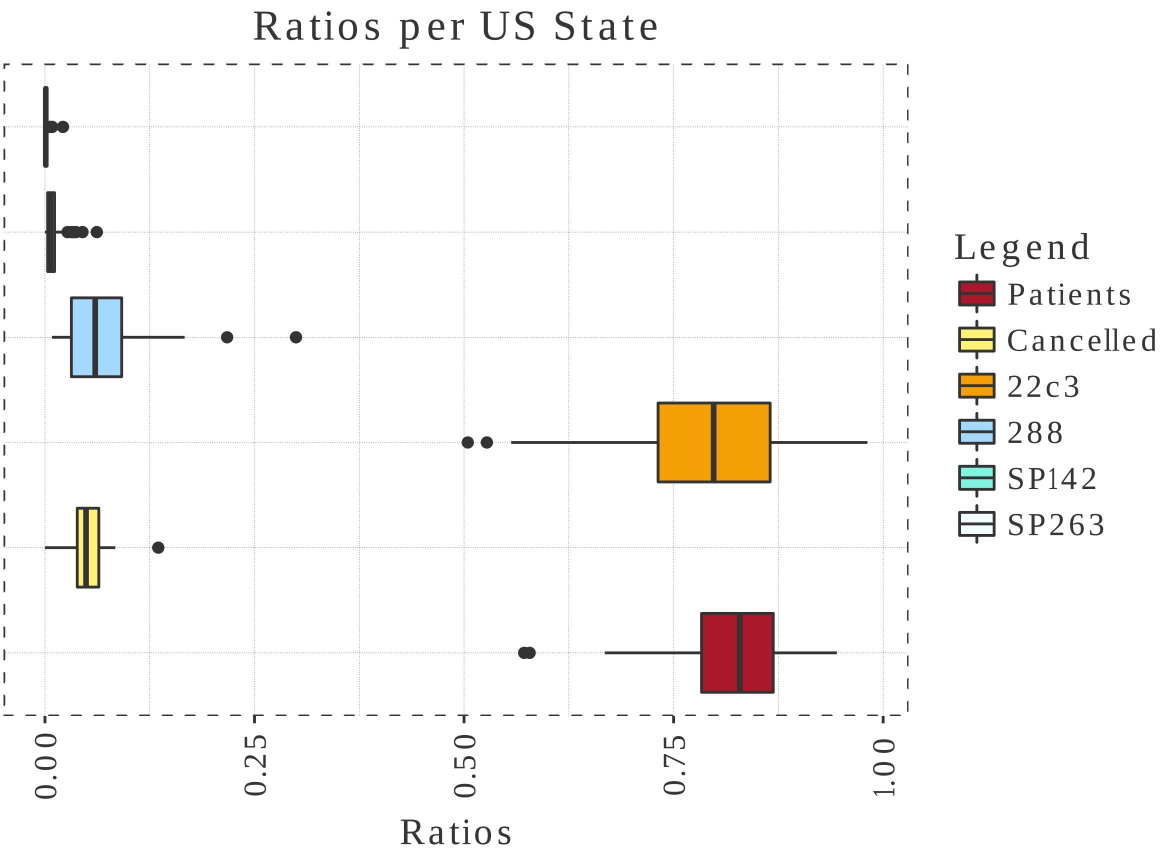
Test.Order.Id	Body.Site	Specimen.Type	Methodology	Tat	state
N124687953265X	Pre Carina FNA	Paraffin Tissue	IHC	176	florida
N127265913265X	Left Pleural Fluid	Paraffin Tissue	IHC	155	california
N137501223249X	Left Mediastinal / Lung Mass	Paraffin Tissue	IHC	94	ohio
C16020725PD-L1	Right Iliac Bone	Tissue	IHC	88	new york
N141937683266X	Right Flank Melanoma	Paraffin Tissue	IHC	70	california
N128941133265X	Right Station 4R Lymph Node	Paraffin Tissue	IHC	64	connecticut
N132875233265X	Liver	Paraffin Tissue	IHC	56	tennessee
N125169733266X	Lymph Node	Paraffin Tissue	IHC	54	texas
N144534783266X	Liver	Paraffin Tissue	IHC	53	california
N144534773265X	Liver	Paraffin Tissue	IHC	53	california

As this plot has been sliced at approximately twenty days, there are still values that exceed this. The above snippet provides some possible *anomalous* behaviour in turnaround times for tests reported, and could shed some light into erronous results obtained or issues concerning specific tests. For example, digging into the underlying reason why some of the largest turnaround times originate from California. The *select* argument above controls the return items for the table and helps to pinpoint common trends.

3) State Data Spreads

This third section is useful in providing a high level overview of some of the features or combination of features in the `map.data`, again on a state by state basis. This is achieved by looking at well defined results and reporting on states that deviate largely from the norm. I have chosen to use boxplots again to display the deviation of results, and will take simple ratios of insightful data to help us build a better picture of particlar features. The snippet below provides the function call to `ratio_features()` for this operation, and the resulting graphic is stored in the list object `ratioResults$plot`.

```
# Create the ratio data of various features
ratioResults <- map.data %>%
  pdata::ratio_features()

ratioResults$plot
```



This function also returns a data frame object as a list of the individual ratios and is accessible through `ratioResults$data`, which will be shown shortly. The plot above displays boxplots of defined ratios per US state. Looking bottom to top; the first boxplot (red) represents the number of patients actually carrying out tests in each state, and the second boxplot (yellow) is the fraction of tests that have been cancelled. The remaining four boxplots represent the antibody being used as a diagnostic in each test. I will expand on these results below using the tabulated form. The following snippet provides the details to cast the results into a human readable form

```
# Construct table from the output
ratioData <- ratioResults$data
tabNames <- ratioData %>% names

blnk <- data.frame(
  uniqueState = "...",
  patientToTest = "...",
  stringsAsFactors = FALSE
)

for (i in 1:(ratioData %>% length)) {
  ratioData[[i]]$type <- NULL
  ratioData[[i]]$patientToTest %<>% signif(digits = 2)
  ratioData[[i]]$uniqueState %<>% openintro::state2abbr()
  ratioData[[i]] <- rbind(
    ratioData[[i]][1:2, ],
    blnk,
    ratioData[[i]][3, ],
    blnk,
    ratioData[[i]][4:5, ]
  )

  names(ratioData[[i]]) <- c(tabNames[i], "Ratio")
}

knitr::kable(
  x = ratioData %>% purrr::reduce(cbind),
  row.names = FALSE,
  align = "c",
  caption = paste0(
    "A table of the maximum two states, the middle ",
    "value state, and the two minimum states of ",
    "ratio data from the boxplot information"
  )
)
```

Table 3: A table of the maximum two states, the middle value state, and the two minimum states of ratio data from the boxplot information

Patients	Ratio	Cancelled	Ratio	22c3	Ratio	288	Ratio	SP142	Ratio	SP263	Ratio
SD	0.94	MD	0.14	DC	0.98	AZ	0.3	AZ	0.062	NH	0.022
WY	0.94	SC	0.084	SD	0.97	NV	0.22	NM	0.045	AZ	0.0087
...	...	...	...	...	...	...	...	...	...	...	...
LA	0.83	TX	0.049	CA	0.8	KY	0.061	IL	0.0063	MO	0.001
...	...	...	...	...	...	...	...	...	...	...	...
NJ	0.58	WY	0.011	AZ	0.53	KS	0.01	ND	0	NE	0
AZ	0.57	DE	0	NJ	0.5	AR	0.0084	SD	0	AR	0

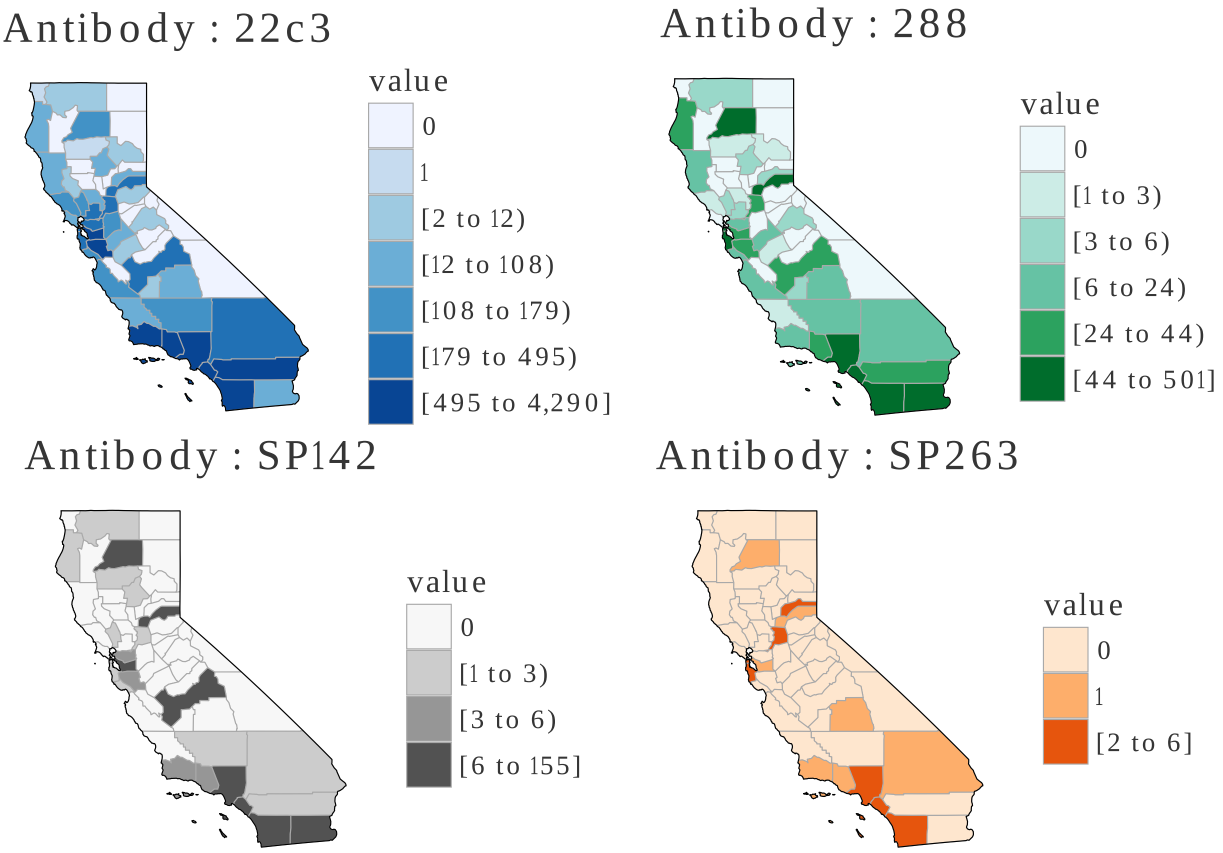
This table above represents a fraction of the results from the boxplots, and can provide some interpretation of the data. For example, the ratio of patients to tests in the first two columns for South Dakota is high, meaning that very few patients are re-tested, in comparison to Arizona, where nearly only 1 in every 2 patients are being tested for the first time (all constrained within the time period of `my.data`). The number of cancellations is also useful, as it is possible to identify locations where tests are being cancelled at a higher rate than others, which could be indicative of multiple issues. Surprisingly, looking at the third and fourth columns, Maryland has the highest cancellation ratio of any state, and Delaware has zero in total, despite geographically the two states being neighbours. Again, these are just some of the many results that can be obtained from this analysis, and the statistics on antibodies used in testing are also provided in the remaining columns. We can also show or report on states that do not use the antibody *SP142* or *SP263*, denoted by zero's in their respective columns.

4) Specific State Information (California)

This final section is useful to help visualise these larger aggregated data sets on a more granular state level. The data set that we have been working with (`map.data`) is now not enough for this section, and we must involve individual county information in order to plot meaningful results. This is probably a little more intricate than the previous investigations, as a number of edge cases are required to be handled for the general case. A default state of *California* is currently employed, as this state has the highest absolute test count from the initial world map shown in the introduction. However, it is possible to supply the input argument `currentState` to this function with any of the states in `map.data` (excluding *unknown* of course). To do so, I have mapped the initial ZIP code, as handled earlier when calculating state data, to a US standard

FIPS county code. Further checks to look for missing FIPS information must also be considered to create the data set required for the package *choroplethr*. I have written this function to look at the antibodies used in each test geographically across any state, however this could easily be extended to look at other features investigated in the previous sections of this report. The snippet below creates the 2 × 2 state plots.

```r
# Look at the most populous counts - california
california <- map.data %>%
  pdata::state_plot()
```



This plot represents each of the 4 antibodies considered in the data set for the state of *California*. It is clear to see from the boxplot of ratios across all states also conforms to the results found here, where Antibody *22c3* is vastly more used in testing compared with the likes of *SP263*, where only single figure values are obtained across the counties.

I have provided a further snippet to access the data and represent it in tabular form for the four most frequent counties in *California* for each antibody.

```r
# Create table for state metrics of California
tabNames <- california %>% names
for (i in 1:4) {
  california[[i]]$region <- NULL
  california[[i]] <- california[[i]][ , 2:1]
  california[[i]]$county %<>% strsplit(' County') %>% purrr::flatten_chr()
  names(california[[i]]) <- c(tabNames[i], "Value")
}

knitr::kable(
  x = california %>% purrr::reduce(cbind),
  row.names = FALSE,
  align = "c",
  caption = paste0(
    "The table of highest frequency counts for each ",
    "of the 4 antibodies considered during testing."
  )
)
```

Table 4: The table of highest frequency counts for each of the 4 antibodies considered during testing.

22c3	Value	28-8	Value	SP142	Value	SP263	Value
Los Angeles	4290	Orange	501	Shasta	155	Los Angeles	6
Orange	2404	Los Angeles	311	Orange	61	San Francisco	3
San Diego	1336	Shasta	160	Los Angeles	48	San Diego	3
Santa Clara	818	San Francisco	159	Imperial	20	Orange	3

The patterns across the antibodies used in testing across California are quite consistent, and even from the table you can see the most frequent occur in Los Angeles and Orange County. This helps provide a more granular vision of the data, right down to the county level. It is also possible to create a more general approach to look at not only Antibodies used in testing, but other useful features.

Conclusions

This final section is simply a brief summary of the analysis carried out. I have taken the route of analysing this data based on geographic location, showing some of the possibilities that can be achieved through a high level perspective. The final results in section 4 outlines an approach to further delve and group the data on the county level, which can provide details right down to this level of detail (and even city specific results if required). The investigation can be carried through further for most of these sections, incorporating other features from the data set and testing hypotheses and trying to identify correlated data.

I have touched a little on carrying out some time series analysis based on section 3 to try and fill `NA` values with something more meaningful. A similar approach could be taken to apply this to other features and metrics of the `map.data`. I have categorized the majority of this data as input features, however, provided some output of test results it could be possible to utilize feature sets such as **Body.Site** and **state** to name a few to try and analyse and predict test outcomes or results. Other grouping parameters could have also been performed to specify key regions where samples are removed by categorizing **Body.Site** even further. With more time, I would have liked to investigate correlations with **Body.Site** and **Icd.Codes**, using dates and other deciding factors.