ACasaccio - Assignment 11.3 - Final Project Step 3

Alysen Casaccio

2023-11-13

Introduction:

In the healthcare industry today, understanding clinician quality of care key performance metrics and the meaning behind these measurements, which are also being submitted at a nationwide scale, is essential for any Ambulatory Quality Director to draw conclusions about their clinics and clinicians. These conclusions help them select not only individual measure-based interventions, like training in processes and best practice recommendations, but it also helps the Quality Director establish and advocate for a high-level quality strategy to drive the best results and achieve better patient outcomes across their clinicians, clinics, and health systems.

Comparing internal results to those of external, or national feedback can be helpful to someone in this role, but it quickly becomes a data science problem due to the large amount of publicly reported data available, and the complex variables that impact clinician results. All of this can quickly become overwhelming to new Quality Directors, who are fundamentally accountable for constantly improving quality of care.

This specific organization has a series of internal narratives the Quality Director is often hearing, but they don't seem to be connected to any data exploration. These narratives include: NPs and PAs have better performance than MDs because they are more tech savvy and can "document their way around the measures". Some clinics are more "quality-forward" than other clinics and they have the resources and longevity of staff to make the metrics work. The MDs have too many patients on their patient panel, and are required to get through too many visits in a day to be good at all of the quality metrics. Quality metrics are solely an individual thing. A given clinician performs well because they care about their work and they work hard to understand what needs done.

I will be comparing a dataset of de-identified, internal health system results to broader, national results, and examining internal factors such as measure and average performance on six key performance indicators, as well as additional factors such as the clinic environment in which the clinicians work (there are 8 in the dataset), and the number of patients each clinician has seen within this 3 quarter dataset.

Problem Statement

How can performance data help an Ambulatory Quality Director establish their quality improvement approach for the upcoming year that is measurable, attainable, and actionable? ## Project Goal My goal is to develop a clear recommendation for a high-level quality approach, and ideally, arm the Quality Director with some data analyses against false narratives being circulated. We have already decided to set individual organizational goals at the 75th percentile of current performance for each measure, but what we don't yet know is if there are specific clinicians and clinics who could benefit more from increased attention and guidance or if the targets should be rolled out with the same focus organization-wide as all clinics and providers are performing at an expected level.

Research Questions:

1. How does the performance of our clinicians compare to national benchmarks and internal targets from the previous year on our selected KPI?
2. Are there patterns of higher or lower performance associated with specific locations?
3. Are there patterns of higher or lower performance associated with patient panels of different sizes?
4. What is the relationship between the role of a clinician (MD, NP, PA) and their performance?
5. Are there any outlier clinicians or practices that are significantly under- or over-performing?
6. How can the insights gained from this analysis be translated into actionable strategies for quality improvement for the upcoming year to best improve clinician performance?

How I Addressed this Problem Statement:

My initial approach was to first explore the data and analyze the datasets toward the specific narratives expressed by the staff in an attempt to provide the Quality Director with clear answers regarding the truth (or falseness) of these beliefs. Once those ideas were validated and presented in easy-to-understand visual elements, I wanted to highlight some statistical examples and provide some simple recommendations based in the statistical analysis and role-model data-driven decision-making, so the Quality Director could see how I came to these recommendations and why I believe they are good for the organization. Finally, I did want to try my hand at the creation of a regression model with the potential to predict clinician performance based on the variables and relationships found in this dataset.

Analysis and Approach

Data:

I originally obtained three datasets for the purposes of this project, and landed on one file, described below, which essentially had three key elements: provider performance across 6 measures and 8 clinics, national benchmarked average performance, and internally set target goals for performance.

Internal De-Identified Clinician Quality Results - The data I used for this project is a spreadsheet I've compiled of providers and practices along with their KPI results across six CMS-based measures. This spreadsheet is specific to the work I do and a consulting engagement I have with a small health system. Due to the sensitive nature of this customer data, the clinician names and practice locations have been de-identified and I have exchanged provider names with the names of spices and clinic practice locations with the names of culinary schools.

Streamlined CMS 2022 Benchmarking Data - The 2nd worksheet is a simple spreadsheet which includes CMS-established national benchmarks for 2022 specific to the measures listed in the Internal De-Identified Clinician Quality Results spreadsheet. These are also found on cms.gov and are published annually to their QPP online library.

Data Import and Cleaning

I imported two excel worksheets into RStudio and performed some initial cleaning and transformation. During my last iteration of this project, I learned there were still some adjustments to be made to the structure of my data and I've made those adjustments prior to upload this time.

The first worksheet is 3 quarters of performance against quality of care measurements for small primary care practices (less than 15 clinicians each). The second worksheet has the CMS established national benchmarks and decile boundaries for the measurements being performed, the average national rate for each measure, and I have added the internal target goal set by the organizational leaders for last year. Prior to loading and reading the data, I de-identified clinician names and practice locations. Although there were no HIPAA concerns, this internal dataset does contain sensitive clinician information, so while I could have done this as a part of my data cleaning, using 'mutate' to replace the sensitive information, I wanted to complete this prior to obtaining the file on my home computer.

```
file_path <- "C:/Users/alyse/OneDrive/Documents/Bellevue University/DSC 520 - Statistics for Data Scien
performance <- read_excel(file_path, sheet = "performance")
benchmarks <- read_excel(file_path, sheet = "2022_benchmarks")
```

```
head(performance)
```

```
## # A tibble: 6 x 7
##   clinic_name    clinician_name role  measure_id   num   den  rate
##   <chr>          <chr>          <chr>      <dbl> <dbl> <dbl> <dbl>
## 1 Le Cordon Bleu Allspice       MD           236   291   395 0.737
## 2 Le Cordon Bleu Angelica       MD           236   210   274 0.766
## 3 Le Cordon Bleu Anise          PA           236    80   139 0.576
## 4 Le Cordon Bleu Bay            MD           236   211   263 0.802
## 5 Le Cordon Bleu Basil          NP           236    87   115 0.757
## 6 Le Cordon Bleu Barberry       NP           236    85   110 0.773
```

```
head(benchmarks)
```

```
## # A tibble: 6 x 19
##   measure_id measure_title            collection_type measure_type high_priority
##        <dbl> <chr>                    <chr>           <chr>        <chr>
## 1        236 Controlling High Blood ~ eCQM            Intermediat~ Y
## 2        112 Breast Cancer Screening  eCQM            Process      N
## 3        113 Colorectal Cancer Scree~ eCQM            Process      N
## 4        309 Cervical Cancer Screeni~ eCQM            Process      N
## 5         47 Advance Care Plan        Medicare Part ~ Process      Y
## 6        134 Depression Screening an~ eCQM            Process      N
## # i 14 more variables: nat_avg_rate <chr>, benchmark_present <chr>,
## #   decile_1 <chr>, decile_2 <chr>, decile_3 <chr>, decile_4 <chr>,
## #   decile_5 <chr>, decile_6 <chr>, decile_7 <chr>, decile_8 <chr>,
## #   decile_9 <chr>, decile_10 <chr>, topped_out <chr>, int_target <chr>
```

Missing Values

After checking to ensure the data was loaded and read correctly, I needed to check for missing values, as I know some of the clinicians are missing calculated rates, especially when the denominator is zero. I don't believe I need to rename any variables as the measure ID tends to follow the CMS naming convention, but that is another aspect I will check now that my data is present. I will also consider any data consolidation that may be needed utilizing joins and binds on key identifiers.

```
na_counts_performance <- performance %>%
  summarize_all(~ sum(is.na(.)))
```

In this summary, I found one observation in the performance data frame that was empty across all 21 variables. I chose to apply a function that checks all rows again and if all elements are an empty string or NA the row would be removed.

```
performance <- performance[!apply(performance, 1, function(x) all(x == "" | is.na(x))), ]
```

I completed the same action on the benchmark dataframe, and having the same findings, decided that the "empty row" is simply the bottom row of each worksheet and I don't believe it will impact my datasets at all.

```
na_counts_benchmarks <- benchmarks %>%
  summarize_all(~ sum(is.na(.)))
```

Condensed Display of Final Data

Below are a few different views of my dataset:

```
str(performance)
```

```
## tibble [372 x 7] (S3: tbl_df/tbl/data.frame)
##  $ clinic_name   : chr [1:372] "Le Cordon Bleu" "Le Cordon Bleu" "Le Cordon Bleu" "Le Cordon Bleu" .
##  $ clinician_name: chr [1:372] "Allspice" "Angelica" "Anise" "Bay" ...
##  $ role          : chr [1:372] "MD" "MD" "PA" "MD" ...
##  $ measure_id    : num [1:372] 236 236 236 236 236 236 236 236 236 236 ...
##  $ num           : num [1:372] 291 210 80 211 87 85 113 108 308 114 ...
##  $ den           : num [1:372] 395 274 139 263 115 110 162 142 352 160 ...
##  $ rate          : num [1:372] 0.737 0.766 0.576 0.802 0.757 ...
```

```
summary(performance)
```

```
##  clinic_name        clinician_name         role            measure_id
##  Length:372         Length:372         Length:372         Min.   : 47.0
##  Class :character   Class :character   Class :character   1st Qu.:112.0
##  Mode  :character   Mode  :character   Mode  :character   Median :123.5
##                                                           Mean   :158.5
##                                                           3rd Qu.:236.0
##                                                           Max.   :309.0
##       num               den              rate
##  Min.   :   0.00   Min.   :   0.0   Min.   :0.0000
##  1st Qu.:  97.75   1st Qu.: 149.0   1st Qu.:0.6421
##  Median : 211.00   Median : 311.0   Median :0.7178
##  Mean   : 275.12   Mean   : 377.0   Mean   :0.6895
##  3rd Qu.: 407.25   3rd Qu.: 541.8   3rd Qu.:0.7778
##  Max.   :1191.00   Max.   :1407.0   Max.   :1.0000
```

```
str(benchmarks)
```

```
## tibble [8 x 19] (S3: tbl_df/tbl/data.frame)
##  $ measure_id       : num [1:8] 236 112 113 309 47 134 1 488
##  $ measure_title    : chr [1:8] "Controlling High Blood Pressure" "Breast Cancer Screening" "Colorec
```

```
##  $ collection_type  : chr [1:8] "eCQM" "eCQM" "eCQM" "eCQM" ...
##  $ measure_type     : chr [1:8] "Intermediate Outcome" "Process" "Process" "Process" ...
##  $ high_priority    : chr [1:8] "Y" "N" "N" "N" ...
##  $ nat_avg_rate     : chr [1:8] "0.623" "0.50990000000000002" "0.49640000000000001" "0.3644" ...
##  $ benchmark_present: chr [1:8] "Y" "Y" "Y" "Y" ...
##  $ decile_1         : chr [1:8] "2.74 - 41.95" "0.27 - 9.22" "0.18 - 7.21" "0.44 - 7.76" ...
##  $ decile_2         : chr [1:8] "41.96 - 51.35" "9.23 - 27.55" "7.22 - 22.60" "7.77 - 15.58" ...
##  $ decile_3         : chr [1:8] "51.36 - 56.60" "27.56 - 39.41" "22.61 - 34.52" "15.59 - 21.87" ...
##  $ decile_4         : chr [1:8] "56.61 - 60.70" "39.42 - 48.17" "34.53 - 43.89" "21.88 - 27.95" ...
##  $ decile_5         : chr [1:8] "60.71 - 64.23" "48.18 - 54.83" "43.90 - 51.88" "27.96 - 34.03" ...
##  $ decile_6         : chr [1:8] "64.24 - 67.54" "54.84 - 60.56" "51.89 - 59.64" "34.04 - 40.16" ...
##  $ decile_7         : chr [1:8] "67.55 - 71.09" "60.57 - 66.81" "59.65 - 66.97" "40.17 - 46.98" ...
##  $ decile_8         : chr [1:8] "71.10 - 75.27" "66.82 - 73.30" "66.98 - 75.50" "46.99 - 54.54" ...
##  $ decile_9         : chr [1:8] "75.28 - 81.34" "73.31 - 82.04" "75.51 - 85.68" "54.55 - 68.51" ...
##  $ decile_10        : chr [1:8] ">= 81.35" ">= 82.05" ">= 85.69" ">= 68.52" ...
##  $ topped_out       : chr [1:8] "No" "No" "No" "No" ...
##  $ int_target       : chr [1:8] "0.69" "0.61" "0.61" "0.65" ...
```

Exploratory Data Analysis (EDA) and Descriptive Statistics

Let's begin with some basic descriptive statistics for all clinicians and their performance rates.

```
descriptive_stats <- performance %>%
  summarise(
    mean_rate = mean(rate, na.rm = TRUE),
    median_rate = median(rate, na.rm = TRUE),
    min_rate = min(rate, na.rm = TRUE),
    max_rate = max(rate, na.rm = TRUE),
    sd_rate = sd(rate, na.rm = TRUE))
print(descriptive_stats)
```

```
## # A tibble: 1 x 5
##   mean_rate median_rate min_rate max_rate sd_rate
##       <dbl>       <dbl>    <dbl>    <dbl>   <dbl>
## 1     0.690       0.718        0        1   0.147
```

```
# calculating descriptive statistics for each measure ID
performance_stats <- performance %>%
  group_by(measure_id) %>%
  summarise(
    mean_rate = mean(rate, na.rm = TRUE),
    median_rate = median(rate, na.rm = TRUE),
    mode_rate = {modes <- mfv(rate); if(length(modes) > 0) modes[1] else NA}, # Handle multiple modes
    sd_rate = sd(rate, na.rm = TRUE),
    .groups = 'drop')

# joining with benchmarks dataframe to get the measure titles
final_table <- performance_stats %>%
  left_join(benchmarks, by = "measure_id") %>%
  select(measure_id, measure_title, mean_rate, median_rate, mode_rate, sd_rate)

# displaying the table
print(final_table)
```

```
## # A tibble: 6 x 6
##   measure_id measure_title                mean_rate median_rate mode_rate sd_rate
##        <dbl> <chr>                            <dbl>       <dbl>     <dbl>   <dbl>
## 1         47 Advance Care Plan                0.667       0.696     0       0.183
## 2        112 Breast Cancer Screening          0.667       0.682     0       0.133
## 3        113 Colorectal Cancer Screening      0.641       0.667     0.7     0.119
## 4        134 Depression Screening and F~      0.783       0.789     0.5     0.0634
## 5        236 Controlling High Blood Pre~      0.713       0.713     0.667   0.0727
## 6        309 Cervical Cancer Screening        0.666       0.735     0       0.209
```

Before moving on to joins and creating some tables, I would like to also calculate the mean, median, range, and standard deviation across only the MDs, then the NPs, then the PAs for all measures. Having these same calculations based across all clinicians in each clinic will also likely prove useful.

```
# calculating statistics for MDs, grouped by measure_id
md_stats <- performance %>%
  filter(role == "MD") %>%
  group_by(measure_id) %>%
  summarise(
    mean_rate = mean(rate, na.rm = TRUE),
    median_rate = median(rate, na.rm = TRUE),
    min_rate = min(rate, na.rm = TRUE),
    max_rate = max(rate, na.rm = TRUE),
    sd_rate = sd(rate, na.rm = TRUE),
    .groups = "drop")

# calculating statistics for NPs, grouped by measure_id
np_stats <- performance %>%
  filter(role == "NP") %>%
  group_by(measure_id) %>%
  summarise(
    mean_rate = mean(rate, na.rm = TRUE),
    median_rate = median(rate, na.rm = TRUE),
    min_rate = min(rate, na.rm = TRUE),
    max_rate = max(rate, na.rm = TRUE),
    sd_rate = sd(rate, na.rm = TRUE),
    .groups = "drop")

# calculating statistics for PAs, grouped by measure_id
pa_stats <- performance %>%
  filter(role == "PA") %>%
  group_by(measure_id) %>%
  summarise(
    mean_rate = mean(rate, na.rm = TRUE),
    median_rate = median(rate, na.rm = TRUE),
    min_rate = min(rate, na.rm = TRUE),
    max_rate = max(rate, na.rm = TRUE),
    sd_rate = sd(rate, na.rm = TRUE),
    .groups = "drop")

# calculating statistics for all clinicians, grouped by measure_id
all_clinician_stats <- performance %>%
  group_by(measure_id) %>%
  summarise(
```

```
    mean_rate = mean(rate, na.rm = TRUE),
    median_rate = median(rate, na.rm = TRUE),
    min_rate = min(rate, na.rm = TRUE),
    max_rate = max(rate, na.rm = TRUE),
    sd_rate = sd(rate, na.rm = TRUE),
    .groups = "drop")

# for all clinicians within each clinic
clinic_stats <- performance %>%
  group_by(clinic_name) %>%
  summarise(
    mean_rate = mean(rate, na.rm = TRUE),
    median_rate = median(rate, na.rm = TRUE),
    min_rate = min(rate, na.rm = TRUE),
    max_rate = max(rate, na.rm = TRUE),
    sd_rate = sd(rate, na.rm = TRUE),
    .groups = "drop")
```

Now that those are all created, I print the results to check them in each dataframe.

```
print(descriptive_stats)
```

```
## # A tibble: 1 x 5
##   mean_rate median_rate min_rate max_rate sd_rate
##       <dbl>       <dbl>    <dbl>    <dbl>   <dbl>
## 1     0.690       0.718        0        1   0.147
```

```
print(md_stats)
```

```
## # A tibble: 6 x 6
##   measure_id mean_rate median_rate min_rate max_rate sd_rate
##        <dbl>     <dbl>       <dbl>    <dbl>    <dbl>   <dbl>
## 1         47     0.708       0.728    0.182    0.957  0.146
## 2        112     0.682       0.710    0.4      0.843  0.0994
## 3        113     0.667       0.697    0.436    0.819  0.0976
## 4        134     0.792       0.793    0.620    0.901  0.0536
## 5        236     0.720       0.713    0.543    0.875  0.0650
## 6        309     0.681       0.749    0.190    1      0.194
```

```
print(np_stats)
```

```
## # A tibble: 6 x 6
##   measure_id mean_rate median_rate min_rate max_rate sd_rate
##        <dbl>     <dbl>       <dbl>    <dbl>    <dbl>   <dbl>
## 1         47     0.582       0.667    0        0.866  0.263
## 2        112     0.632       0.669    0        1      0.220
## 3        113     0.570       0.618    0.2      0.8    0.173
## 4        134     0.768       0.781    0.5      0.864  0.0897
## 5        236     0.688       0.7      0.483    0.773  0.0931
## 6        309     0.598       0.724    0        0.852  0.282
```

```
print(pa_stats)
```

```
## # A tibble: 6 x 6
##   measure_id mean_rate median_rate min_rate max_rate sd_rate
##        <dbl>     <dbl>       <dbl>    <dbl>    <dbl>   <dbl>
## 1         47     0.536       0.557    0.407    0.673  0.104
## 2        112     0.629       0.657    0.458    0.706  0.0979
## 3        113     0.605       0.603    0.577    0.641  0.0264
## 4        134     0.744       0.735    0.679    0.805  0.0530
## 5        236     0.713       0.736    0.576    0.789  0.0819
## 6        309     0.709       0.721    0.622    0.758  0.0520
```

```
print(all_clinician_stats)
```

```
## # A tibble: 6 x 6
##   measure_id mean_rate median_rate min_rate max_rate sd_rate
##        <dbl>     <dbl>       <dbl>    <dbl>    <dbl>   <dbl>
## 1         47     0.667       0.696    0        0.957  0.183
## 2        112     0.667       0.682    0        1      0.133
## 3        113     0.641       0.667    0.2      0.819  0.119
## 4        134     0.783       0.789    0.5      0.901  0.0634
## 5        236     0.713       0.713    0.483    0.875  0.0727
## 6        309     0.666       0.735    0        1      0.209
```

```
print(clinic_stats)
```

```
## # A tibble: 8 x 6
##   clinic_name           mean_rate median_rate min_rate max_rate sd_rate
##   <chr>                     <dbl>       <dbl>    <dbl>    <dbl>   <dbl>
## 1 Boston University         0.725       0.723    0.330    0.957  0.110
## 2 Culinary Arts Academy     0.598       0.697    0        1      0.271
## 3 Culinary Institute        0.743       0.749    0.6      0.875  0.0629
## 4 Escoffier School          0.731       0.736    0.431    0.906  0.0995
## 5 Hattori Nutrition         0.607       0.623    0.182    1      0.186
## 6 Kendall College           0.617       0.634    0.190    0.784  0.133
## 7 La Cuisine Paris          0.679       0.704    0.2      0.866  0.127
## 8 Le Cordon Bleu            0.714       0.733    0.458    0.879  0.0925
```

Summarizing early patterns

The data has all been sliced and diced appropriately and prepared for my next step of creating data visualizations and joining with the benchmark data to easily bring over measure titles and national performance data ranges. I will want to use some joined tables and basic comparisons and/or visualizations, like histograms or heat maps to display some of the early patterns I am finding in the data.

One example of this includes comparing the mean of each measure across all three roles. Since the measure IDs don't carry a lot of meaning for anyone not working deeply in quality of care, I've decided to bring the measure titles over into my table from the benchmark dataframe.

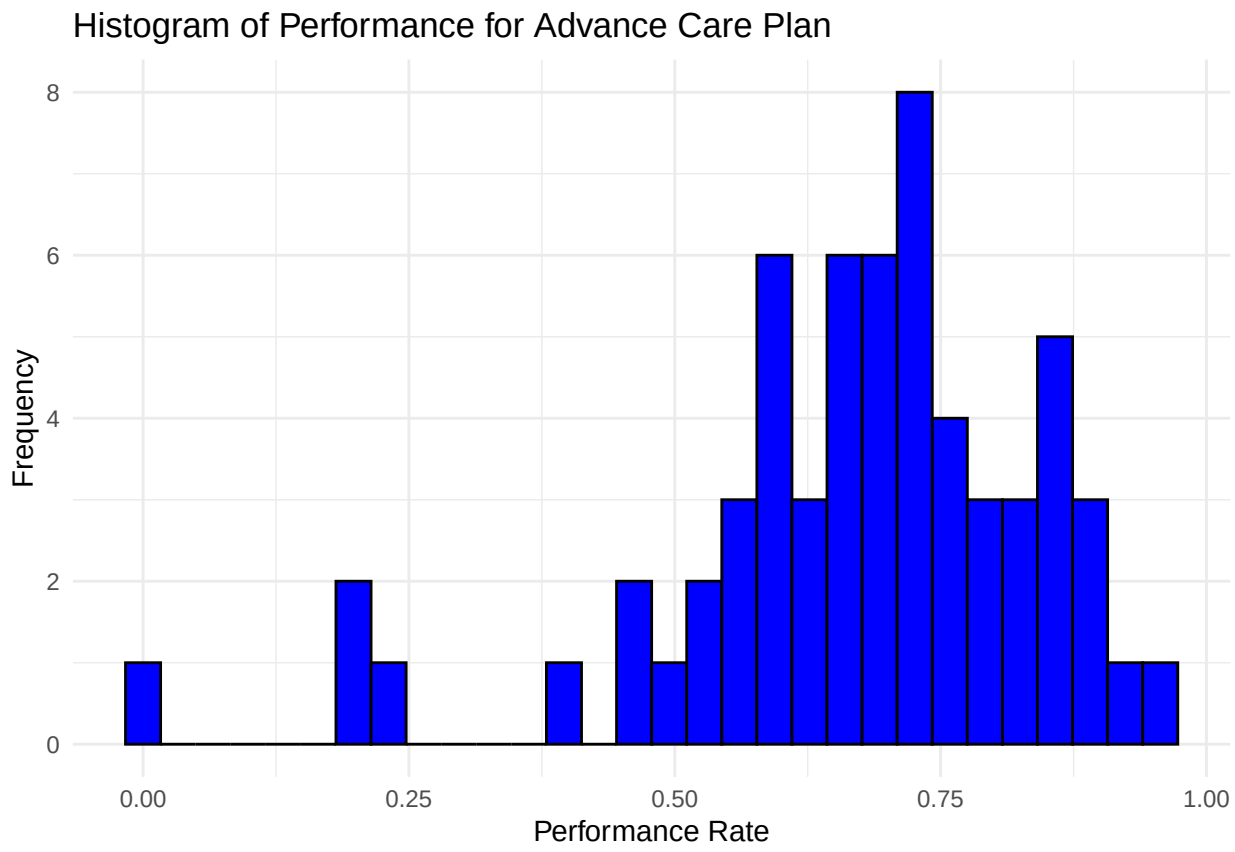
```r
# renaming the mean columns for clarity
md_stats <- rename(md_stats, MD_mean = mean_rate)
np_stats <- rename(np_stats, NP_mean = mean_rate)
pa_stats <- rename(pa_stats, PA_mean = mean_rate)

# joining the dataframes
role_means <- full_join(md_stats, np_stats, by = "measure_id")
role_means <- full_join(role_means, pa_stats, by = "measure_id")

# ensuring 'measure_id' in the benchmarks dataframe is numeric
benchmarks <- benchmarks %>%
  mutate(measure_id = as.numeric(measure_id))

# joining with the benchmarks dataframe to get measure titles
role_comparison_with_titles <- role_means %>%
  left_join(select(benchmarks, measure_id, measure_title), by = "measure_id")

# arranging by measure_id for easier reading and selecting the relevant columns
role_comparison_with_titles <- role_comparison_with_titles %>%
  arrange(measure_id) %>%
  select(measure_id, measure_title, MD_mean, NP_mean, PA_mean)

# viewing the table
print(role_comparison_with_titles)
```

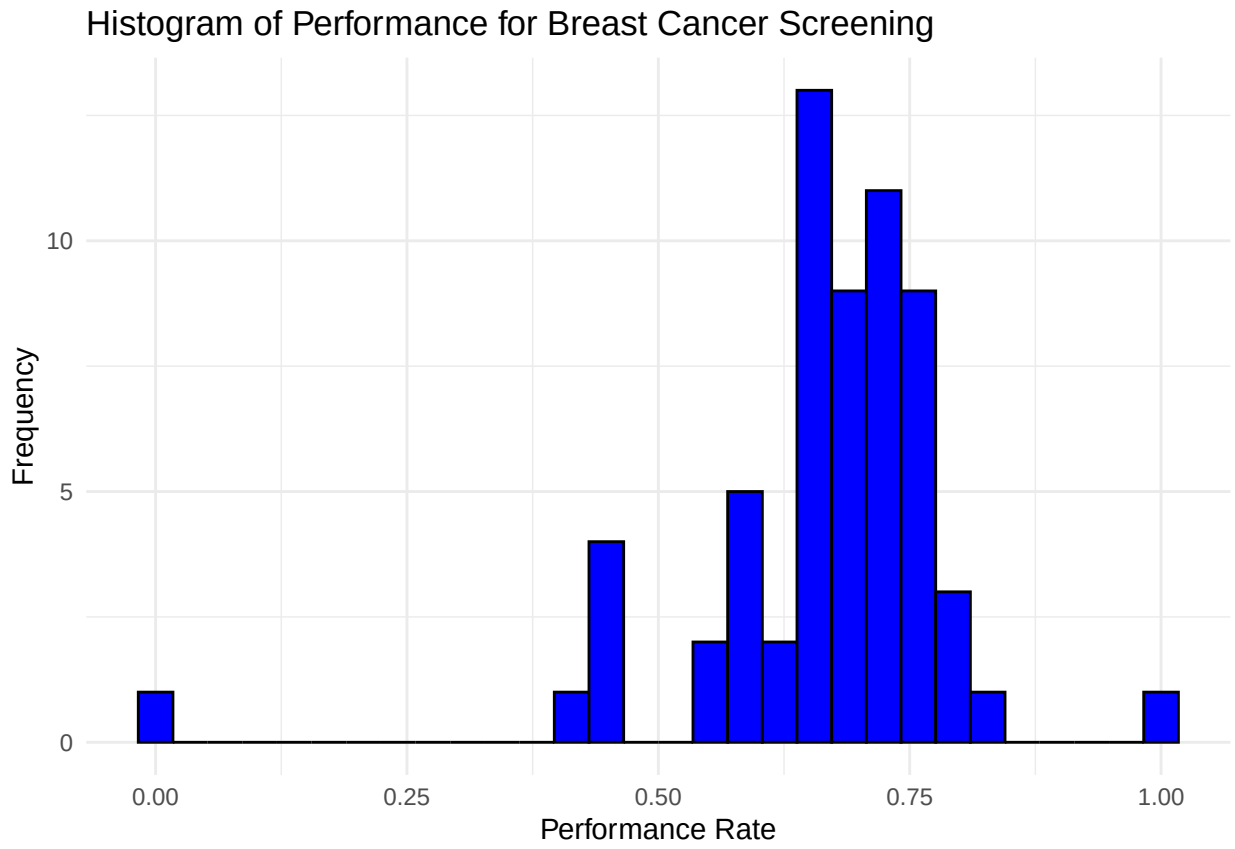
```
## # A tibble: 6 x 5
##   measure_id measure_title                        MD_mean NP_mean PA_mean
##        <dbl> <chr>                                  <dbl>   <dbl>   <dbl>
## 1         47 Advance Care Plan                      0.708   0.582   0.536
## 2        112 Breast Cancer Screening                0.682   0.632   0.629
## 3        113 Colorectal Cancer Screening            0.667   0.570   0.605
## 4        134 Depression Screening and Follow-Up     0.792   0.768   0.744
## 5        236 Controlling High Blood Pressure        0.720   0.688   0.713
## 6        309 Cervical Cancer Screening              0.681   0.598   0.709
```

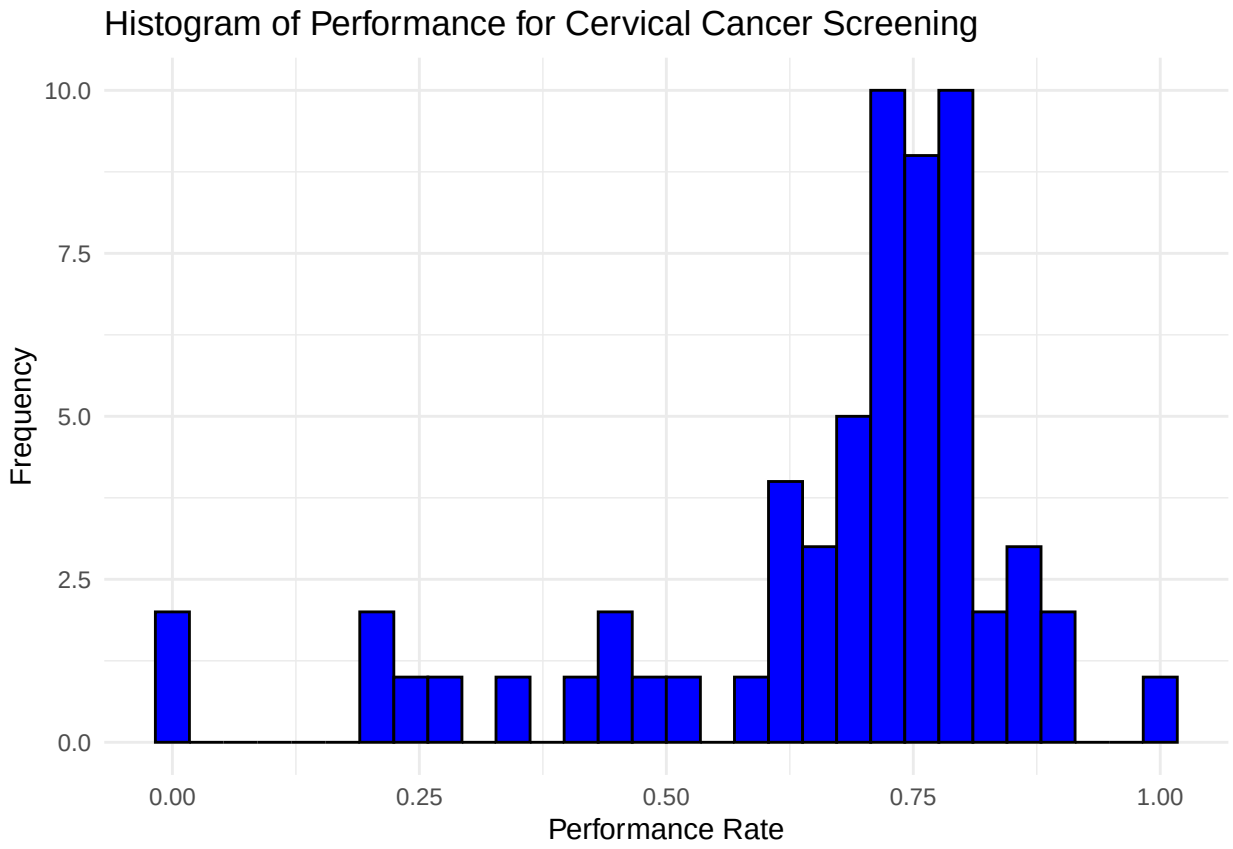
One way to examine measure performance overall is by using histograms. I learned during earlier iterations that it is much faster (and more elegant) to create visualizations for all the measures by looping through them, rather than writing code for each one at a time.

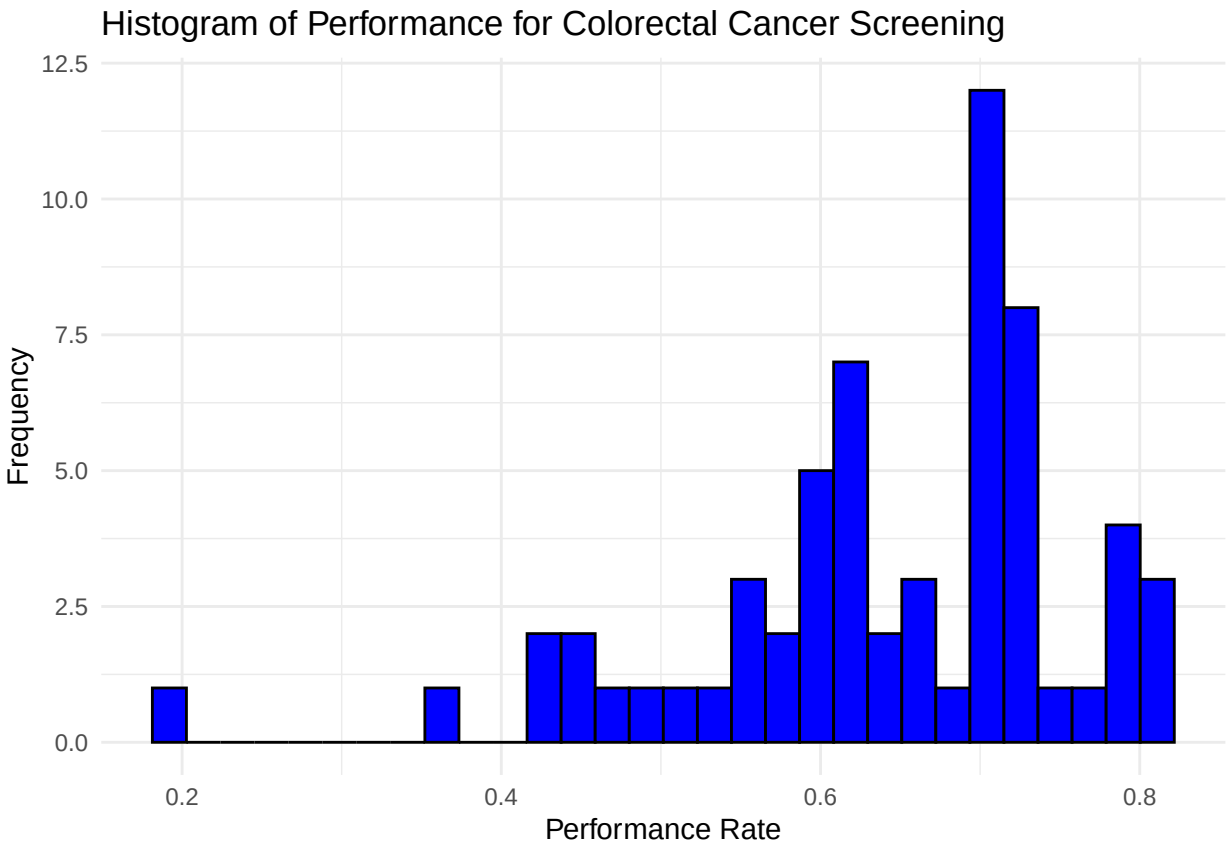
```r
# defining the specific measure IDs
selected_measure_ids <- c(47, 112, 113, 134, 236, 309)

# joining performance dataframe with benchmarks to get measure titles
# and filtering for selected measure IDs
performance_with_titles <- performance %>%
  filter(measure_id %in% selected_measure_ids) %>%
  left_join(benchmarks, by = "measure_id")

# creating a histogram for each selected measure
output_plots <- performance_with_titles %>%
  group_by(measure_title) %>%
  do(
    plot = ggplot(., aes(x = rate)) +
```

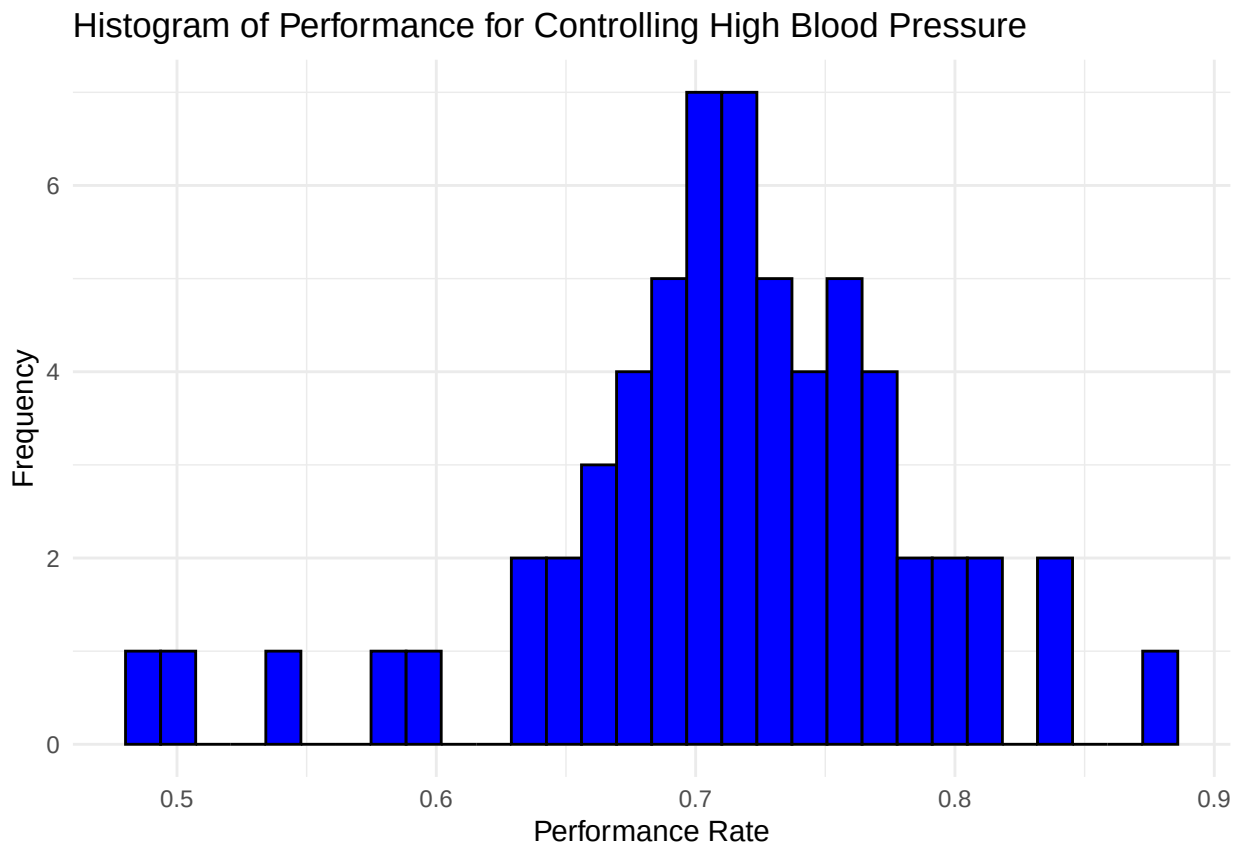
```
      geom_histogram(bins = 30, fill = "blue", color = "black") +
      ggtitle(paste("Histogram of Performance for", .$measure_title[1])) +
      xlab("Performance Rate") +
      ylab("Frequency") +
      theme_minimal())

# displaying the plots
invisible(lapply(output_plots$plot, print))
```

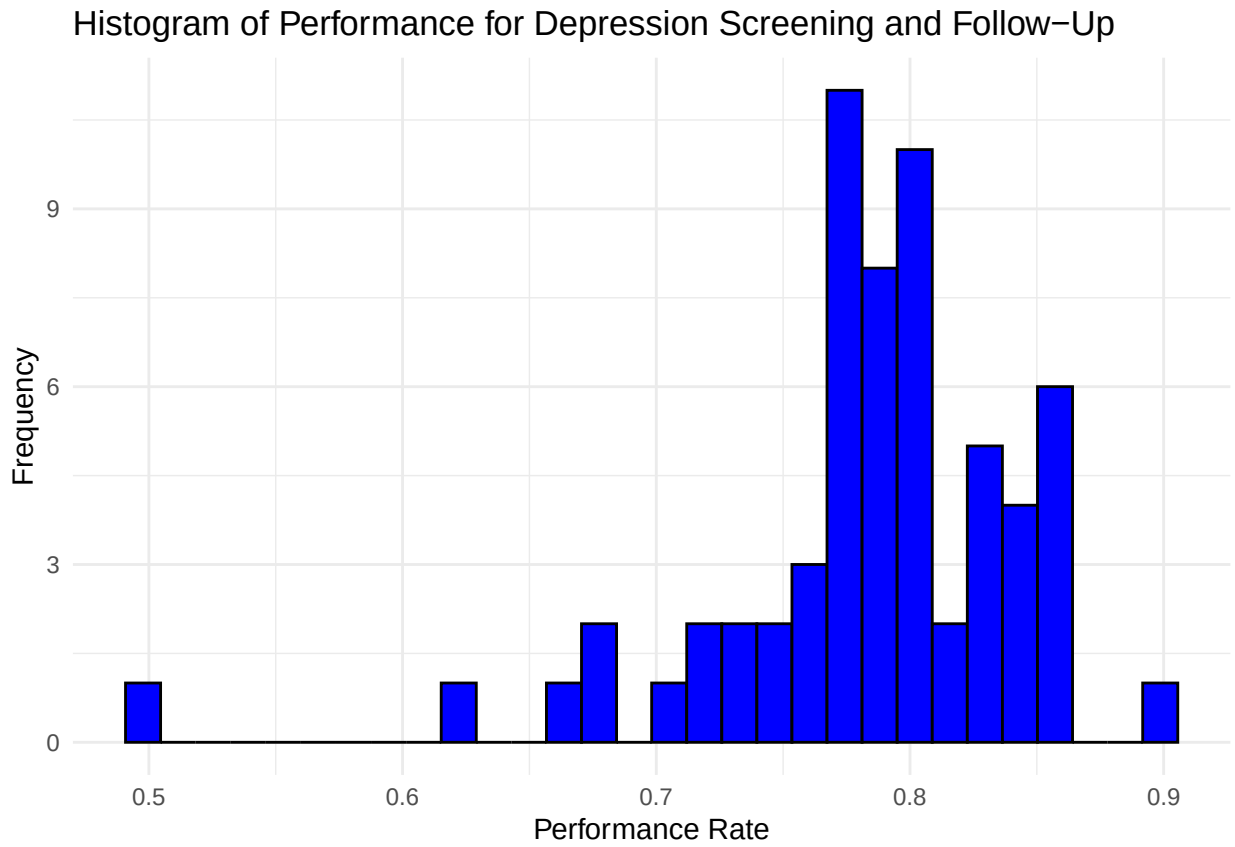



Histogram of Performance for Breast Cancer Screening
Frequency
10
5
0
0.00
0.25
0.50
0.75
1.00
Performance Rate


Histogram of Performance for Cervical Cancer Screening
Frequency
10.0
7.5
5.0
2.5
0.0
0.00
0.25
0.50
0.75
1.00
Performance Rate

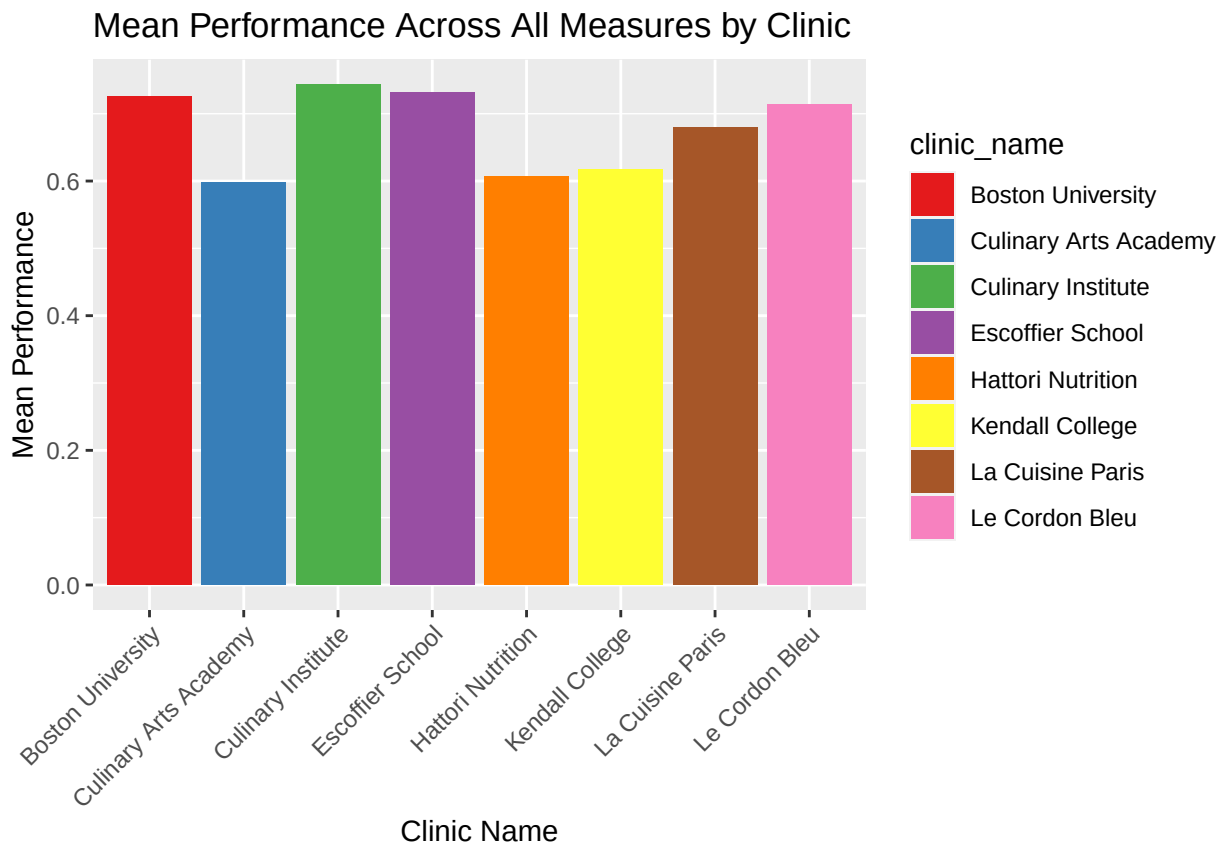

Histogram of Performance for Colorectal Cancer Screening
Frequency
12.5
10.0
7.5
5.0
2.5
0.0
0.2
0.4
0.6
0.8
Performance Rate


Histogram of Performance for Controlling High Blood Pressure
Frequency
0
2
4
6
0.5
0.6
0.7
0.8
0.9
Performance Rate



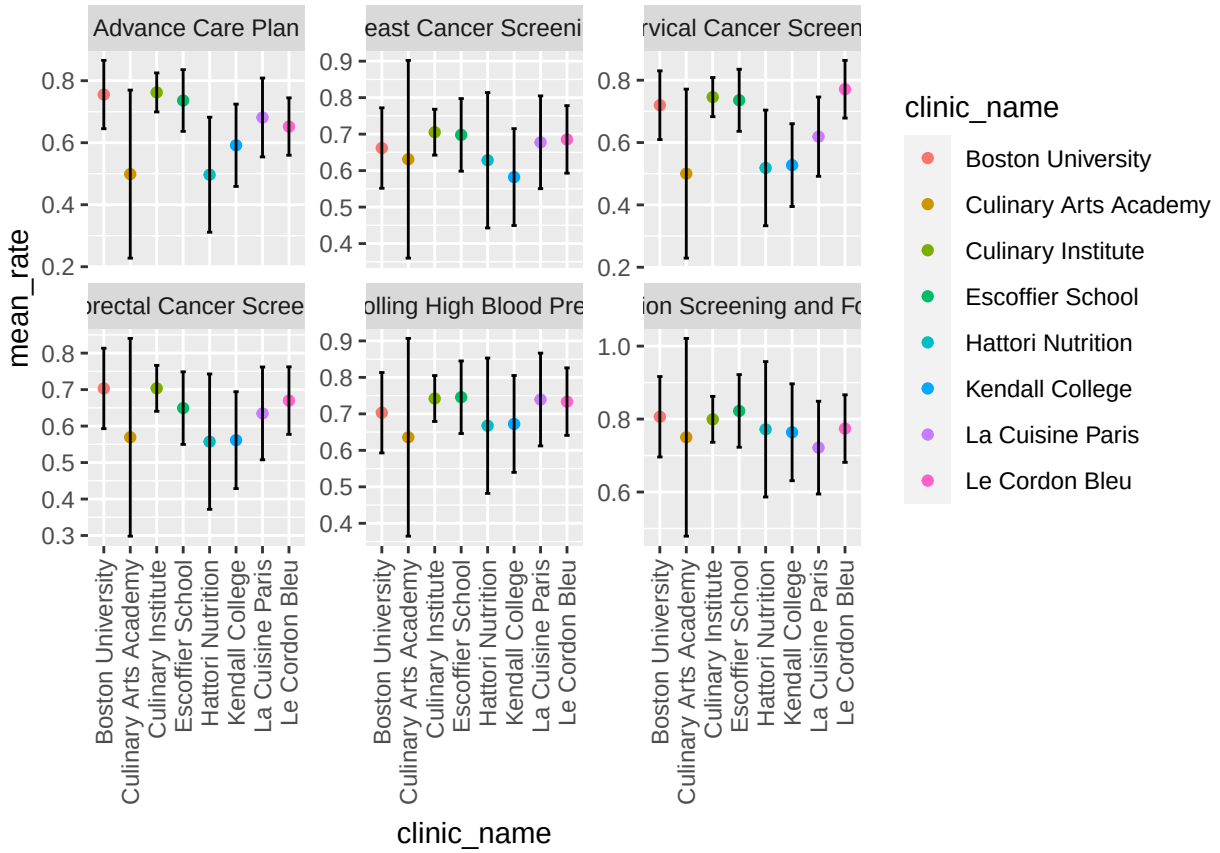
While we are examining some visual ways to look at performance statistics, I think it is also important to show the mean performance per clinic as this is how most quality directors roll up performance for understanding clinic-by-clinic. It does lack the nuance of a deeper data exploration, but I still want to include it here for later comparison.

```
# calculating mean performance for each clinic
clinic_performance <- performance %>%
  group_by(clinic_name) %>%
  summarise(mean_performance = mean(rate, na.rm = TRUE))

# creating a bar chart
ggplot(clinic_performance, aes(x = clinic_name, y = mean_performance, fill = clinic_name)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(y = "Mean Performance", x = "Clinic Name") +
  ggtitle("Mean Performance Across All Measures by Clinic") +
  scale_fill_brewer(palette = "Set1")
```



One last visualization I'd like to present involves the standard deviation.

```
# aggregating performance data
performance_aggregated <- performance %>%
  group_by(clinic_name, measure_id) %>%
  summarise(mean_rate = mean(rate, na.rm = TRUE), .groups = 'drop')

# joining with SD rates for each clinic
performance_with_sd <- performance_aggregated %>%
  left_join(clinic_stats %>% select(clinic_name, sd_rate), by = "clinic_name")

# joining with measure titles for clarity
performance_with_titles <- performance_with_sd %>%
  left_join(benchmarks %>% select(measure_id, measure_title), by = "measure_id")

# creating a faceted error bar plot
ggplot(performance_with_titles, aes(x = clinic_name, y = mean_rate, group = clinic_name)) +
  geom_point(aes(color = clinic_name)) +  # points for mean
  geom_errorbar(aes(ymin = mean_rate - sd_rate, ymax = mean_rate + sd_rate), width = 0.2) +  # error ba
  facet_wrap(~ measure_title, scales = 'free_y') +  # creating a facet for each measure
    theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1),  # Adjust text angle and just
        axis.title.x = element_text(vjust = -0.5))  # Adjust x-axis title position
```




```
  labs(x = "Clinic", y = "Mean with SD", title = "Comparison of Mean and SD across Clinics and Measures
  theme_minimal()
```

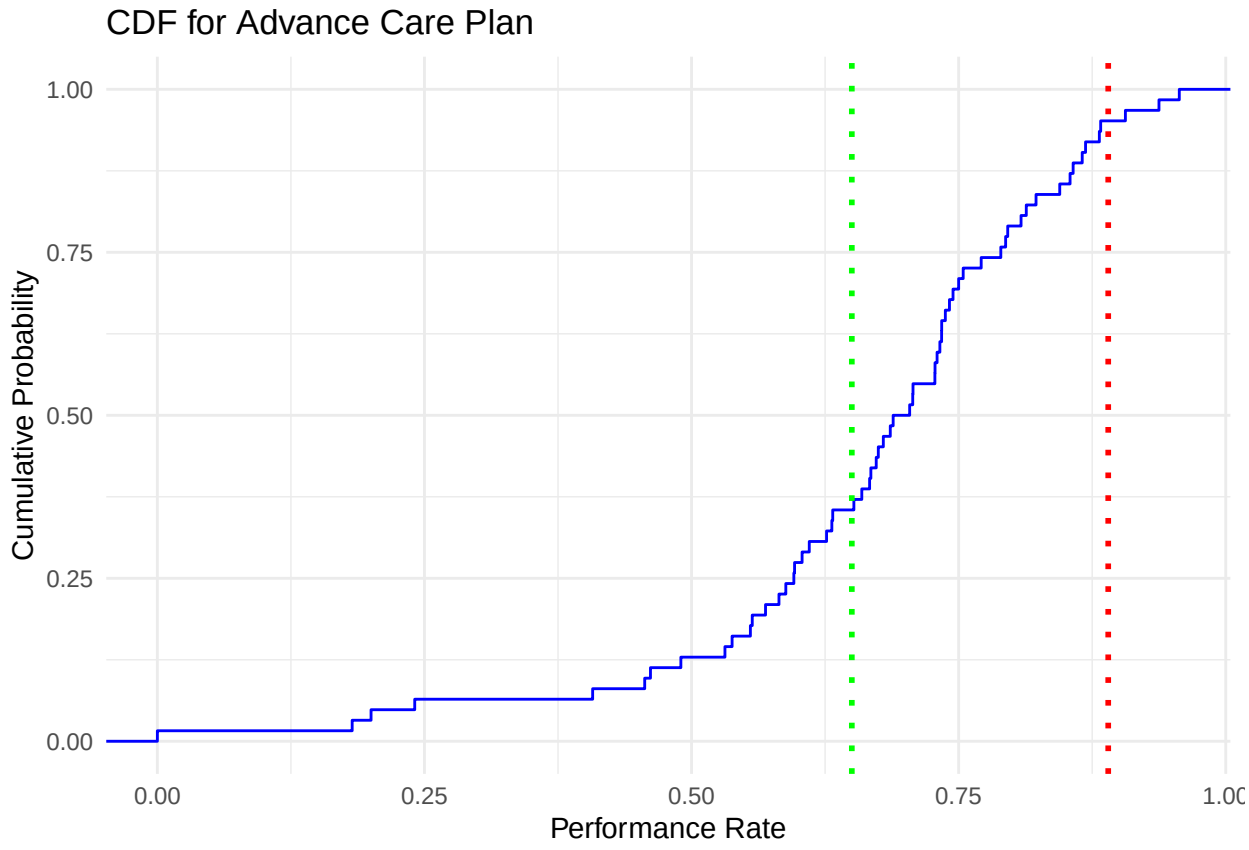
```
## NULL
```

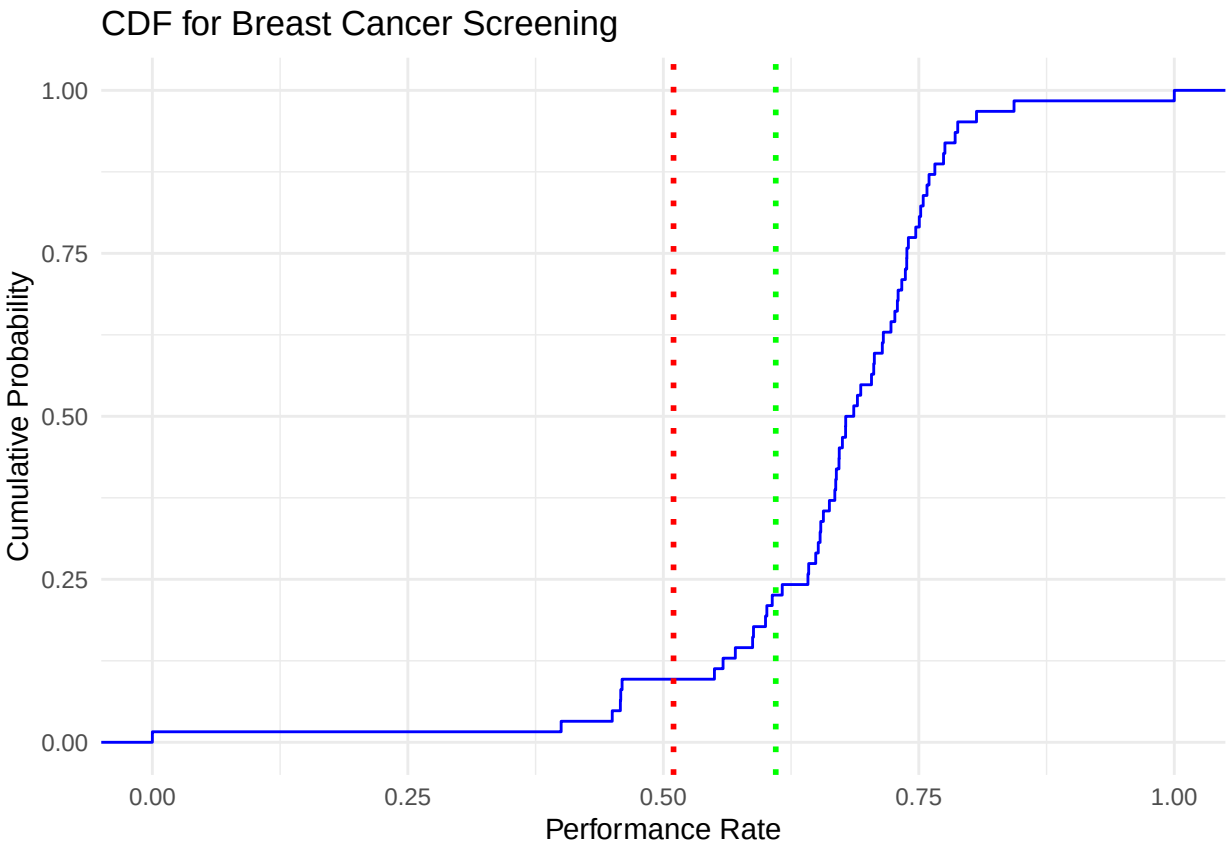
Summarizing Early Patterns

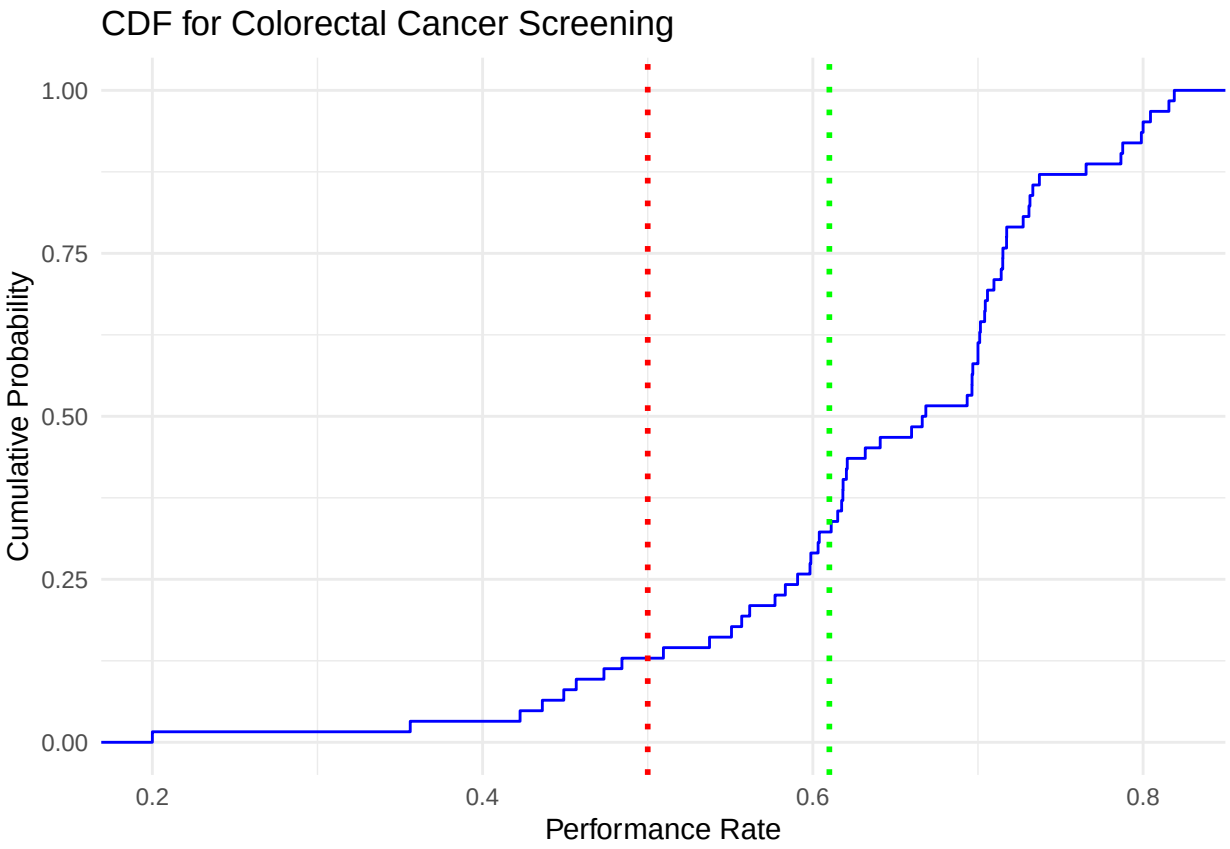
Now that some basic comparative visualizations have been summarized, I can dig a little deeper into the patterns and compare the distributions of each KPI to both national benchmarks and internal goal-setting targets. I performed a number of cumulative distribution functions for my other class and found them to be a highly insightful tool when reviewing clinician performance, so I've looked up how to do this in R and plan to start there. I will say... this piece took multiple iterations, and I needed to use several online resources to get the plots functioning the way I wanted, so it is maybe not something I would do using R again. I also included a check to make sure the appropriate values were numeric and using the same rounding as I was getting some suspicious errors in the first few iterations of this code.

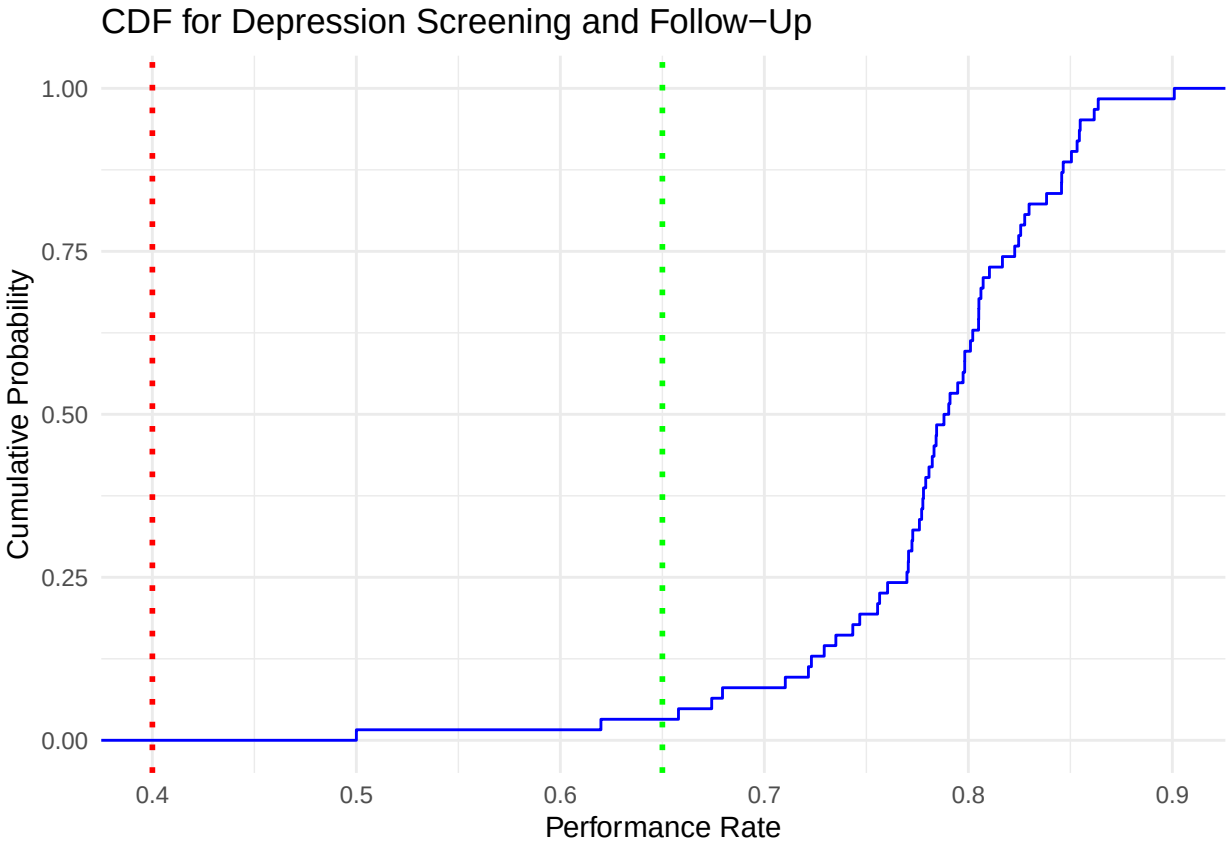
```
# defining the specific measure IDs to display
selected_measure_ids <- c(47, 112, 113, 134, 236, 309)

# merging performance data with benchmark data
# ensure the national average rates and internal targets are numeric and rounded to two decimal places
performance_with_benchmark <- performance %>%
  filter(measure_id %in% selected_measure_ids) %>%
  left_join(benchmarks %>% select(measure_id, measure_title, nat_avg_rate, int_target), by = "measure_i
```

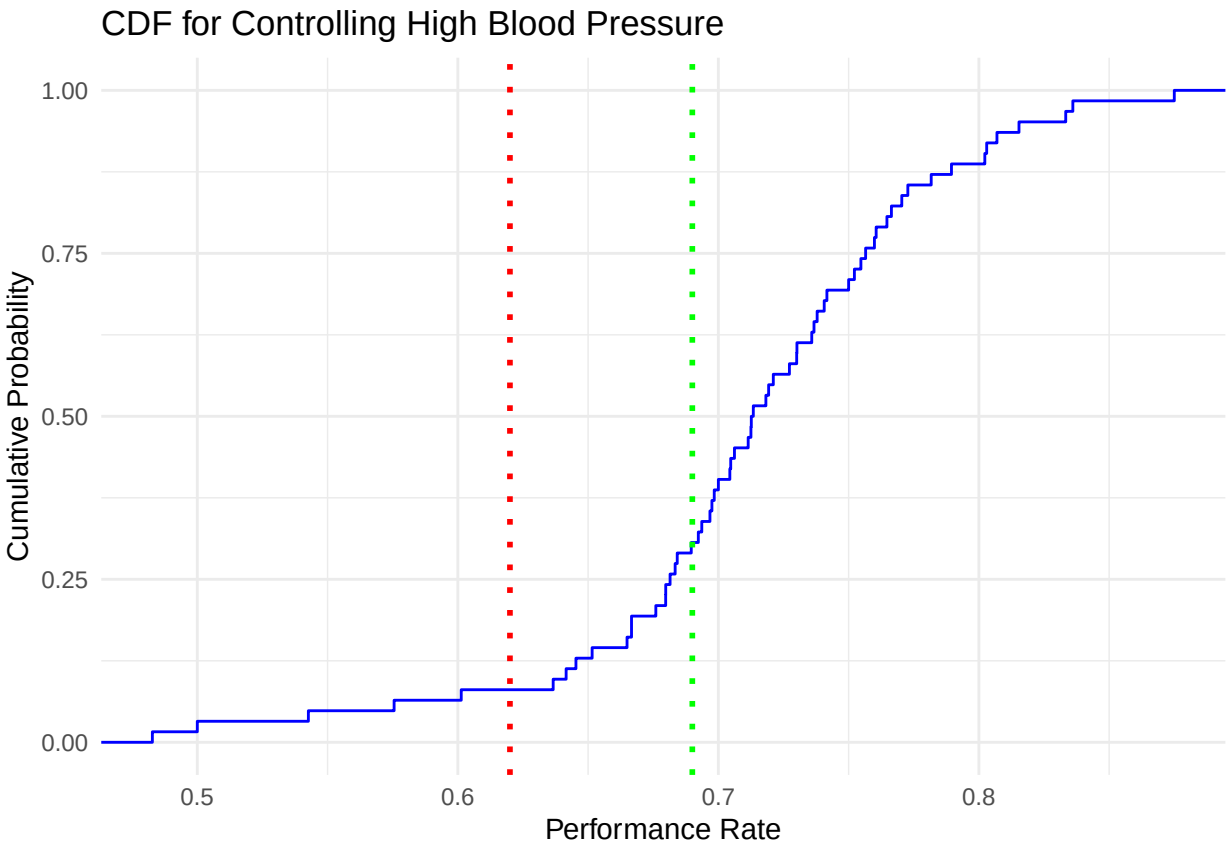
```r
  mutate(nat_avg_rate = round(as.numeric(nat_avg_rate), 2),
         int_target = round(as.numeric(int_target), 2))

# creating a CDF plot for each of the selected measures
performance_with_benchmark %>%
  split(.$measure_id) %>%
  map(~ {
    ggplot(.x, aes(x = rate)) +
      stat_ecdf(geom = "step", color = "blue") +
      geom_vline(aes(xintercept = nat_avg_rate), color = "red", linetype = "dotted", linewidth = 1) +
      geom_vline(aes(xintercept = int_target), color = "green", linetype = "dotted", linewidth = 1) +
      labs(title = paste("CDF for", unique(.x$measure_title)),
           x = "Performance Rate",
           y = "Cumulative Probability") +
      theme_minimal()}) %>%
  walk(print)  # Print each plot
```

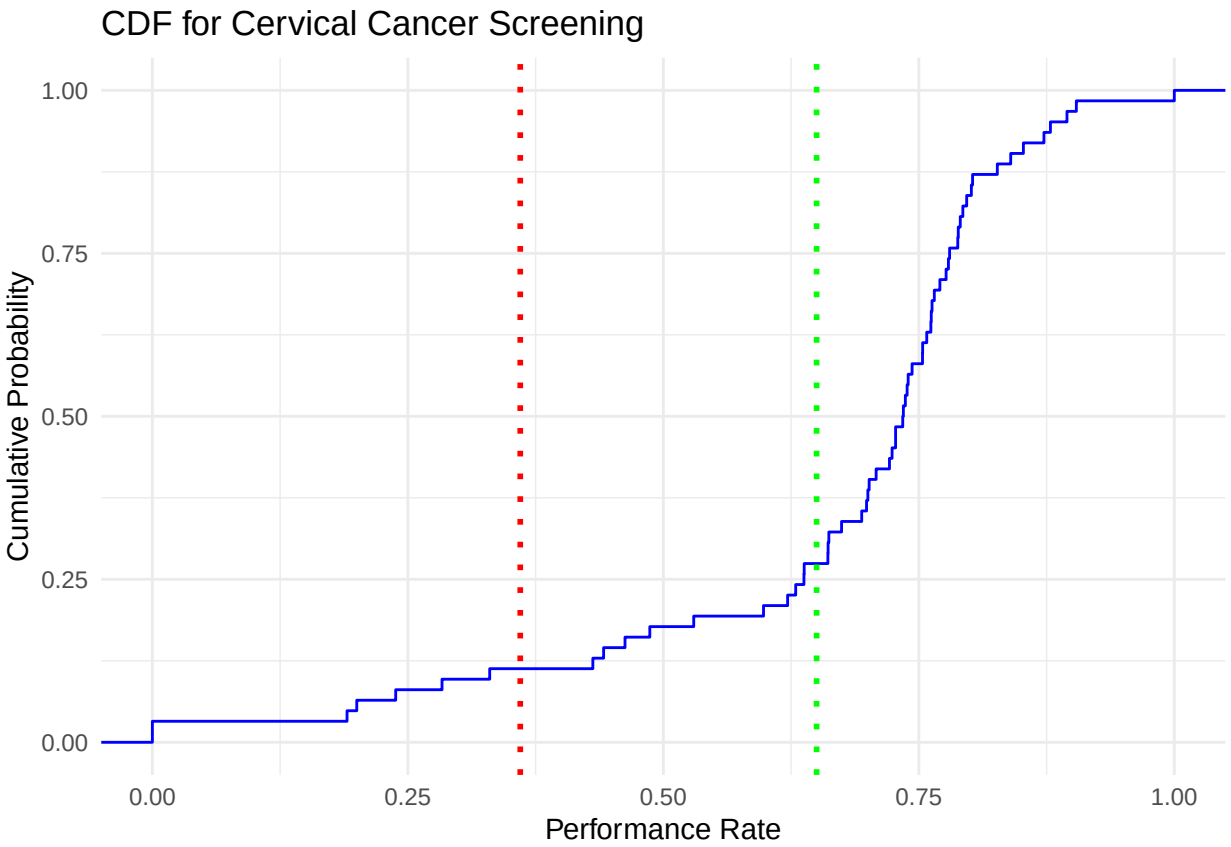



CDF for Breast Cancer Screening
Cumulative Probability
1.00
0.75
0.50
0.25
0.00
0.00
0.25
0.50
0.75
1.00
Performance Rate


CDF for Colorectal Cancer Screening
Cumulative Probability
1.00
0.75
0.50
0.25
0.00
0.2
0.4
0.6
0.8
Performance Rate


CDF for Depression Screening and Follow−Up
Cumulative Probability
1.00
0.75
0.50
0.25
0.00
0.4
0.5
0.6
0.7
0.8
0.9
Performance Rate


CDF for Controlling High Blood Pressure
Cumulative Probability
1.00
0.75
0.50
0.25
0.00
0.5
0.6
0.7
0.8
Performance Rate



That was challenging, but I think it really helps me understand the performance patterns in conjunction with the histograms. I can also very clearly see why some plots are better generated using Python and others are better with R. Before moving to the unsupervised learning algorithms, I would like to do one more visualization and analysis.

First, I want to know the 75th percentile (as a target goal) and 90th percentile (as a stretch goal) of the current performance rates for each measure as this may help our Quality Director compare to the target goals they are considering for 2024. Setting these goals in a place that reassures our top-performers they are already doing well while attempting to shift the entire clinician performance slightly to the right will help us make a data-driven decision regarding future internal targets.

```
# selecting measure IDs to analyze
selected_measure_ids <- c(47, 112, 113, 134, 236, 309)

# merging performance data with benchmark data
performance_with_benchmark <- performance %>%
  filter(measure_id %in% selected_measure_ids) %>%
  left_join(benchmarks %>% select(measure_id, measure_title, nat_avg_rate, int_target), by = "measure_i
  mutate(nat_avg_rate = round(as.numeric(nat_avg_rate), 2),
         int_target = round(as.numeric(int_target), 2))

# calculating 75th and 90th percentiles for each measure
percentile_table <- performance_with_benchmark %>%
  group_by(measure_id, measure_title, int_target) %>%
  summarize(future_target = quantile(rate, 0.75, na.rm = TRUE),
            future_stretch = quantile(rate, 0.90, na.rm = TRUE)) %>%
  ungroup()
```

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

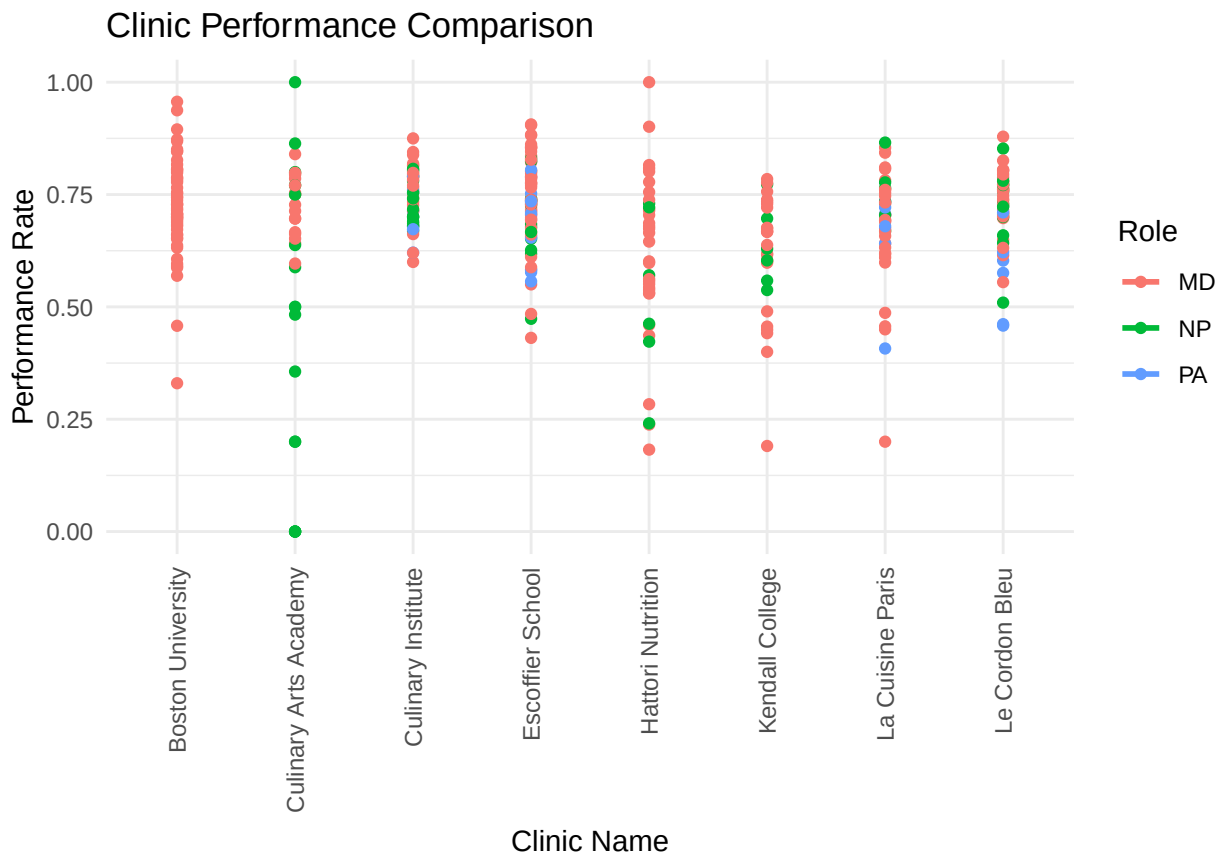
```
# displaying the table
print(percentile_table)
```

```
## # A tibble: 6 x 5
##   measure_id measure_title               int_target future_target future_stretch
##        <dbl> <chr>                            <dbl>         <dbl>          <dbl>
## 1         47 Advance Care Plan                 0.65         0.785          0.865
## 2        112 Breast Cancer Screening           0.61         0.738          0.773
## 3        113 Colorectal Cancer Screening       0.61         0.715          0.784
## 4        134 Depression Screening and F~       0.65         0.821          0.850
## 5        236 Controlling High Blood Pre~       0.69         0.756          0.801
## 6        309 Cervical Cancer Screening         0.65         0.780          0.839
```

Finally, I want to see a scatter plot of performance rates and clinics to see if there is a correlation between specific clinics and higher rates of performance. Since I have also been doing some basic investigation of performance by role, I thought I would try to color-code my dots by role to see if a pattern emerges there, too.

```
# creating a scatter plot comparing clinics and performance

ggplot(performance, aes(x = clinic_name, y = rate, color = role)) +
  geom_point() +  # Add points
  geom_smooth(method = "lm", se = FALSE) +  # Optional: Add a linear trend line
  labs(x = "Clinic Name", y = "Performance Rate", color = "Role") +
  ggtitle("Clinic Performance Comparison") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1),  # Adjust text angle and justif
        axis.title.x = element_text(vjust = -0.5))  # Adjust x-axis title position
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Linear Model: Clinics and Roles

I need to create a linear model of clinician roles and clinic names, then view the p-values and coefficients from the model. My idea here is to assess whether clinic or clinician role are statistically significant factors that impact clinician performance in this dataset, so I can clearly communicate this to the Quality Director.

```
# building the linear model
model <- lm(rate ~ clinic_name + role, data = performance)

# obtaining a summary
model_summary <- summary(model)

# extracting the coefficients table
coefficients_table <- model_summary$coefficients

# displaying the coefficients and p-values in a more readable format
full_table <- coefficients_table %>%
  as.data.frame() %>%
  rownames_to_column(var = "Variable") %>%
  select(Variable, Estimate, `Pr(>|t|)`)

# results
print(full_table)
```

```
##                            Variable    Estimate      Pr(>|t|)
## 1                       (Intercept)  0.72494955 2.004005e-133
## 2  clinic_nameCulinary Arts Academy -0.09431867  2.096592e-03
## 3     clinic_nameCulinary Institute  0.05420406  5.339533e-02
## 4       clinic_nameEscoffier School  0.03029430  2.163419e-01
## 5      clinic_nameHattori Nutrition -0.10716997  2.729968e-04
## 6        clinic_nameKendall College -0.09520512  2.224028e-03
## 7       clinic_nameLa Cuisine Paris -0.02360425  4.026987e-01
## 8         clinic_nameLe Cordon Bleu  0.01722873  5.304019e-01
## 9                           roleNP -0.06614474  4.128766e-04
## 10                          rolePA -0.09054436  8.562692e-04
```

Interpretation

Some of the simple interpretation I would like to point out to the Quality Director here includes: * Clinics statistically associated with a lower performance rate: ** Culinary Arts Academy ** Hattori Nutrition ** Kendall College

In addition, both NPs and PAs are statistically associated with a lower performance rate. These are factors I will include in my recommendation for a quality strategy below.

Evolving Hypothesis

I anticipate there is a lot of information even in the limited dataset I have, hiding patterns or groupings of clinician performance that isn't easily identifiable. Before we move into trying to develop some clusters and modeling, I think I need to further refine my dataframes.

One of the things I'd like to segment out is the number of unique patients seen by each clinician as I anticipate this could be an impactful factor. The quality measure id: 134 is Depression Screening, and the denominator for this measure is a count of unique patients within each provider's entire adult population (age 18+).

I will use this metric to create a field in a new dataframe. I want to use the value found in the denominator of any row identified as measure_id 134 and populate a column for each provider titled "pts_treated". Oonce it is created, I can group clinicians into clusters based on similarities in their performance rates and number of patients treated.

```
# filtering performance data for measure_id 134
depression_screening <- performance %>%
  filter(measure_id == 134) %>%
  select(clinician_name, clinic_name, role, den)

# renaming 'den' column to 'pts_treated'
clinician_data <- depression_screening %>%
  rename(pts_treated = den)
```

Let's check to make sure that worked as intended:

```
print(clinician_data)
```

```
## # A tibble: 62 x 4
##    clinician_name clinic_name        role  pts_treated
##    <chr>          <chr>              <chr>       <dbl>
```

```
##  1 Allspice       Le Cordon Bleu     MD           1365
##  2 Angelica       Le Cordon Bleu     MD            940
##  3 Anise          Le Cordon Bleu     PA            673
##  4 Bay            Le Cordon Bleu     MD           1100
##  5 Basil          Le Cordon Bleu     NP            725
##  6 Barberry       Le Cordon Bleu     NP            538
##  7 Bergamot       Le Cordon Bleu     MD            708
##  8 Borage         Le Cordon Bleu     MD            726
##  9 Caper          Culinary Institute MD            903
## 10 Caraway        Culinary Institute MD            239
## # i 52 more rows
```

Next, in order to include performance rates, I have to decide how they are going to be represented, and I believe the simplest solution is to represent them as an average performance rate per clinician, and then visually checking the dataframe again.

```
# calculating average performance rate for each clinician
average_performance_rates <- performance %>%
  group_by(clinician_name) %>%
  summarise(avg_rate = mean(rate, na.rm = TRUE), .groups = 'drop')

# merging with clinician_data
clinician_data <- clinician_data %>%
  left_join(average_performance_rates, by = "clinician_name")
```

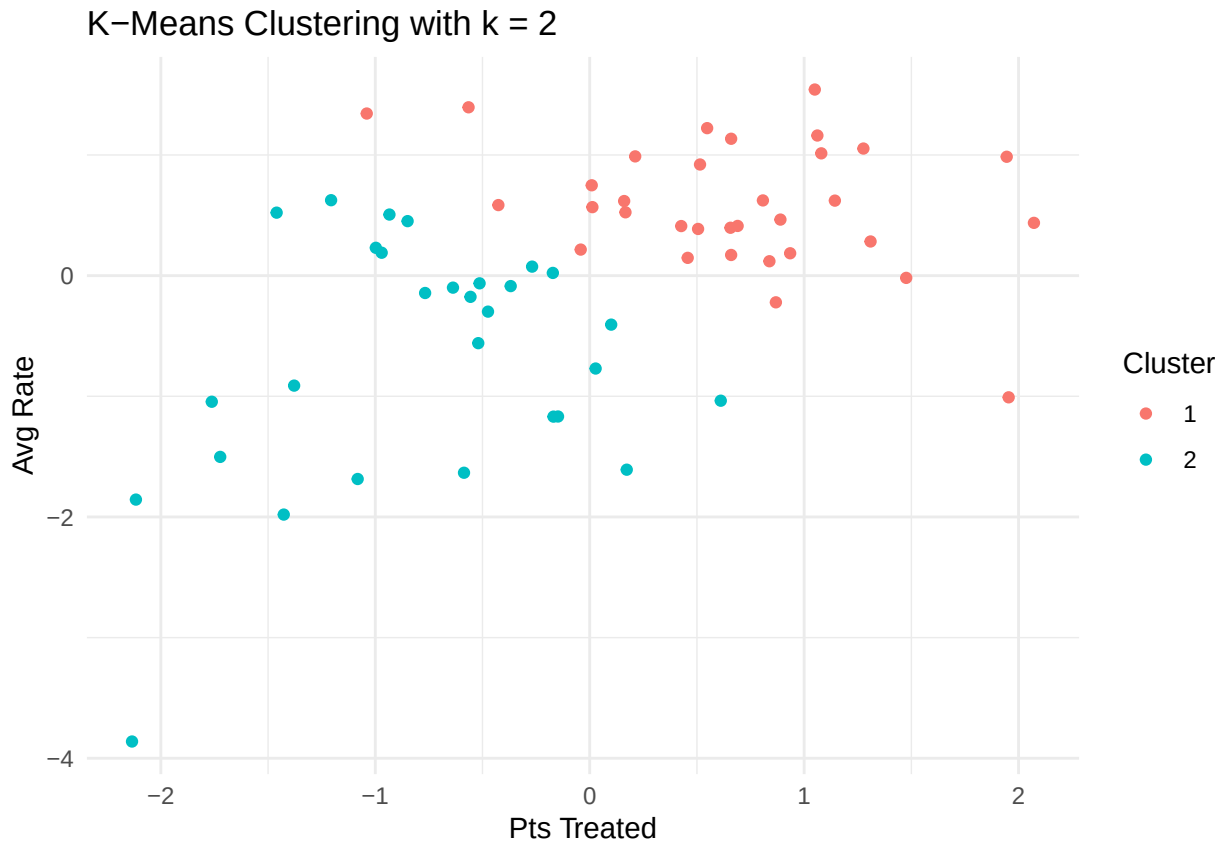
And I am checking the dataframe again.

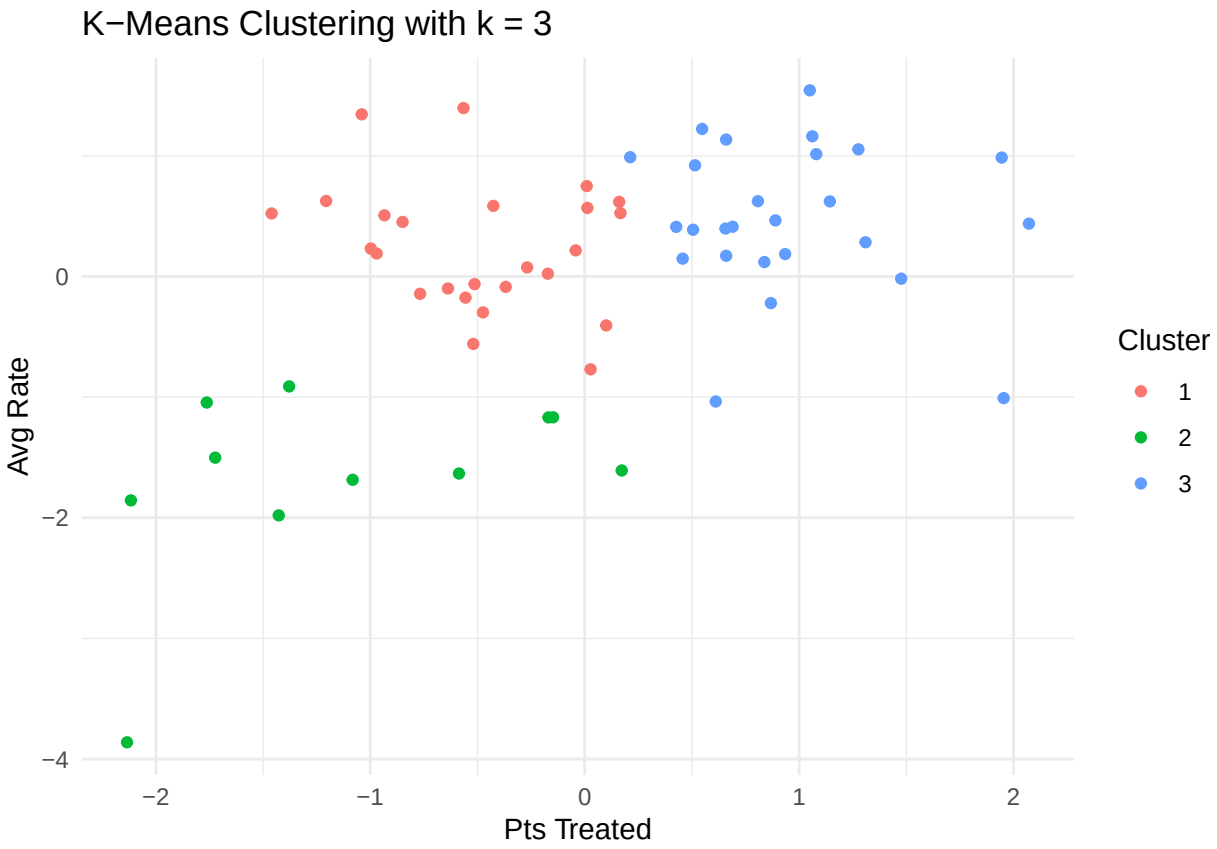
```
head(clinician_data)
```

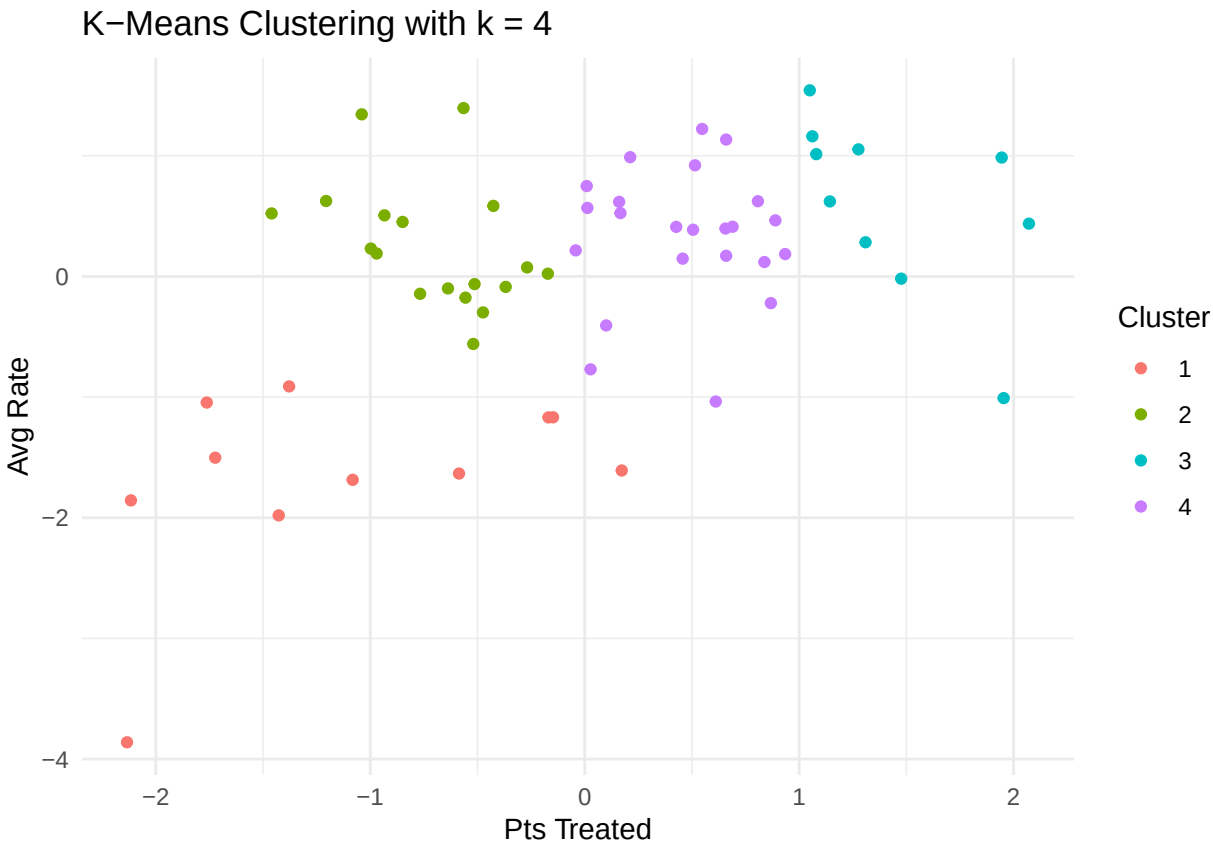
```
## # A tibble: 6 x 5
##   clinician_name clinic_name    role  pts_treated avg_rate
##   <chr>          <chr>          <chr>       <dbl>    <dbl>
## 1 Allspice       Le Cordon Bleu MD           1365    0.789
## 2 Angelica       Le Cordon Bleu MD            940    0.707
## 3 Anise          Le Cordon Bleu PA            673    0.572
## 4 Bay            Le Cordon Bleu MD           1100    0.752
## 5 Basil          Le Cordon Bleu NP            725    0.765
## 6 Barberry       Le Cordon Bleu NP            538    0.672
```

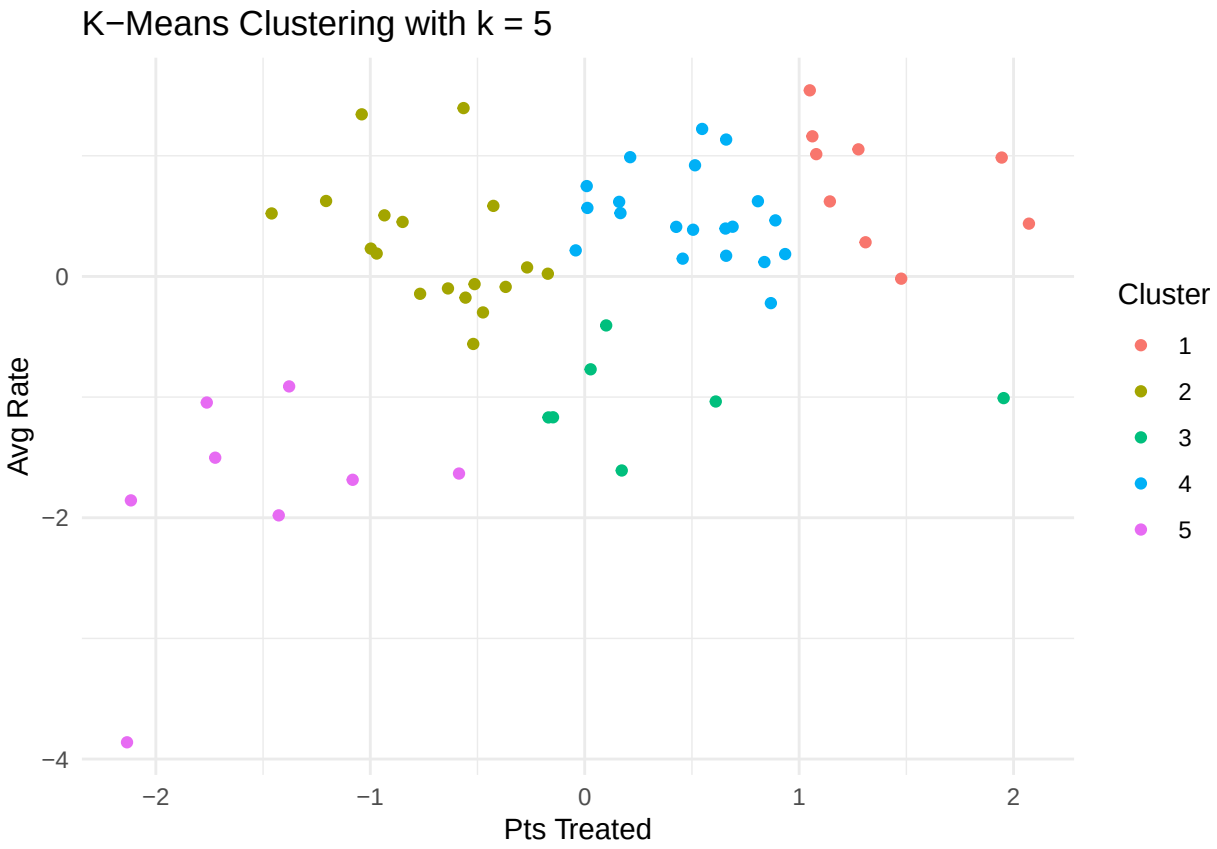
```
# selecting relevant columns for clustering and converting to dataframe
clinician_cluster_data <- clinician_data %>%
  select(pts_treated, avg_rate) %>%
  as.data.frame()

# normalizing data to the appropriate scale
clinician_cluster_data <- scale(clinician_cluster_data)

# converting scaled data back to a dataframe
clinician_cluster_data <- as.data.frame(clinician_cluster_data)

# preparing a list to store the results
kmeans_results <- list()
```

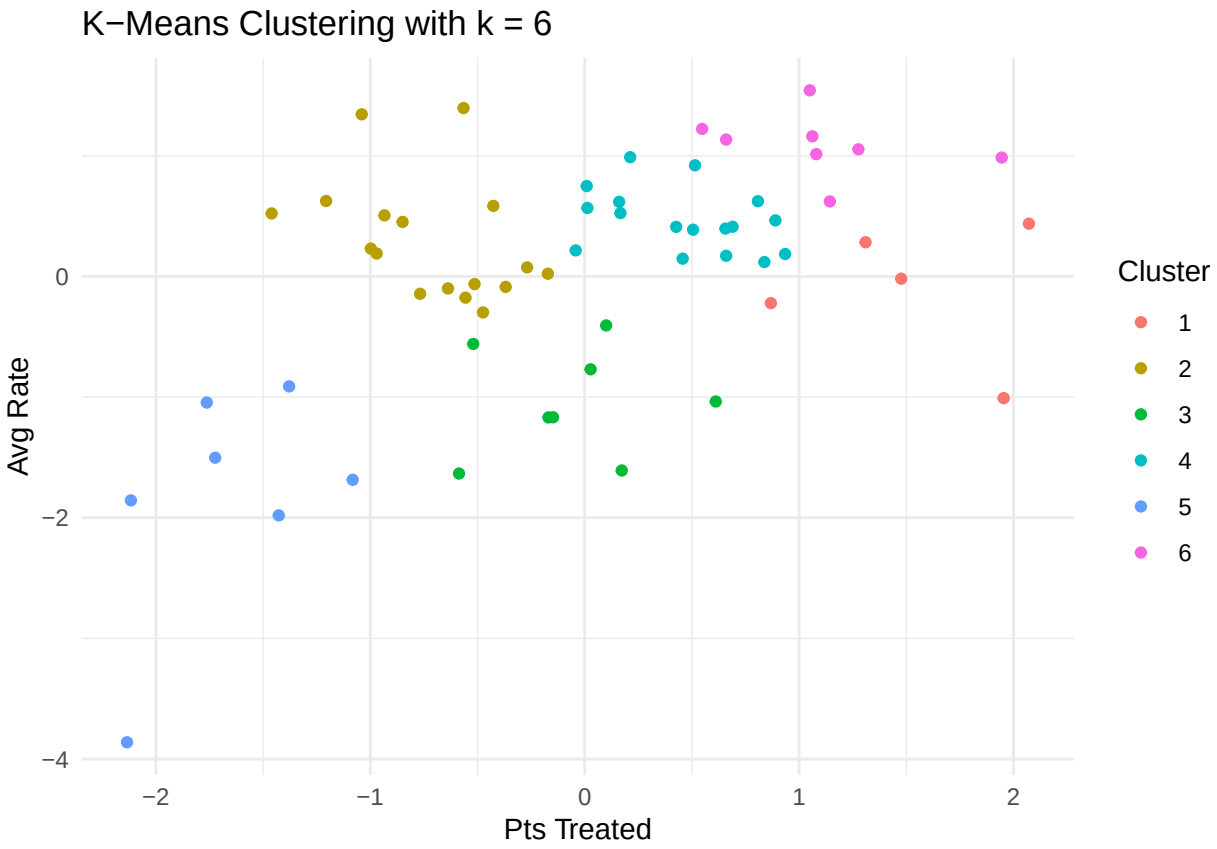
```r
# fitting the k-means and storing results for k = 2 to 12
for (k in 2:12) {set.seed(123) # For reproducible results
  kmeans_result <- kmeans(clinician_cluster_data, centers = k)
  kmeans_results[[as.character(k)]] <- kmeans_result}

# plotting the results
for (k in 2:12) {cluster_data <- clinician_cluster_data
  cluster_data$cluster <- kmeans_results[[as.character(k)]]$cluster

  p <- ggplot(cluster_data, aes(x = pts_treated, y = avg_rate, color = as.factor(cluster))) +
    geom_point() +
    labs(title = paste("K-Means Clustering with k =", k),
         x = "Pts Treated",
         y = "Avg Rate",
         color = "Cluster") +
    theme_minimal()

  print(p)}
```

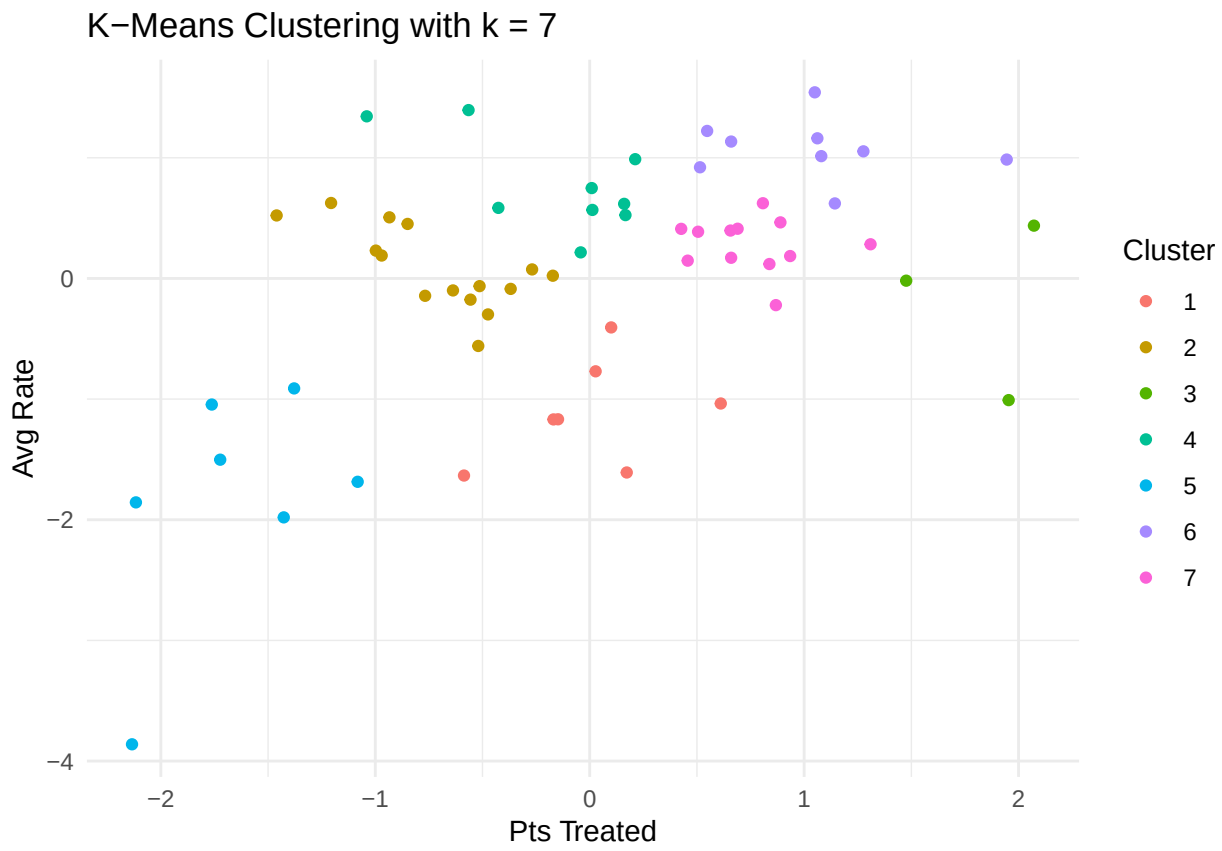


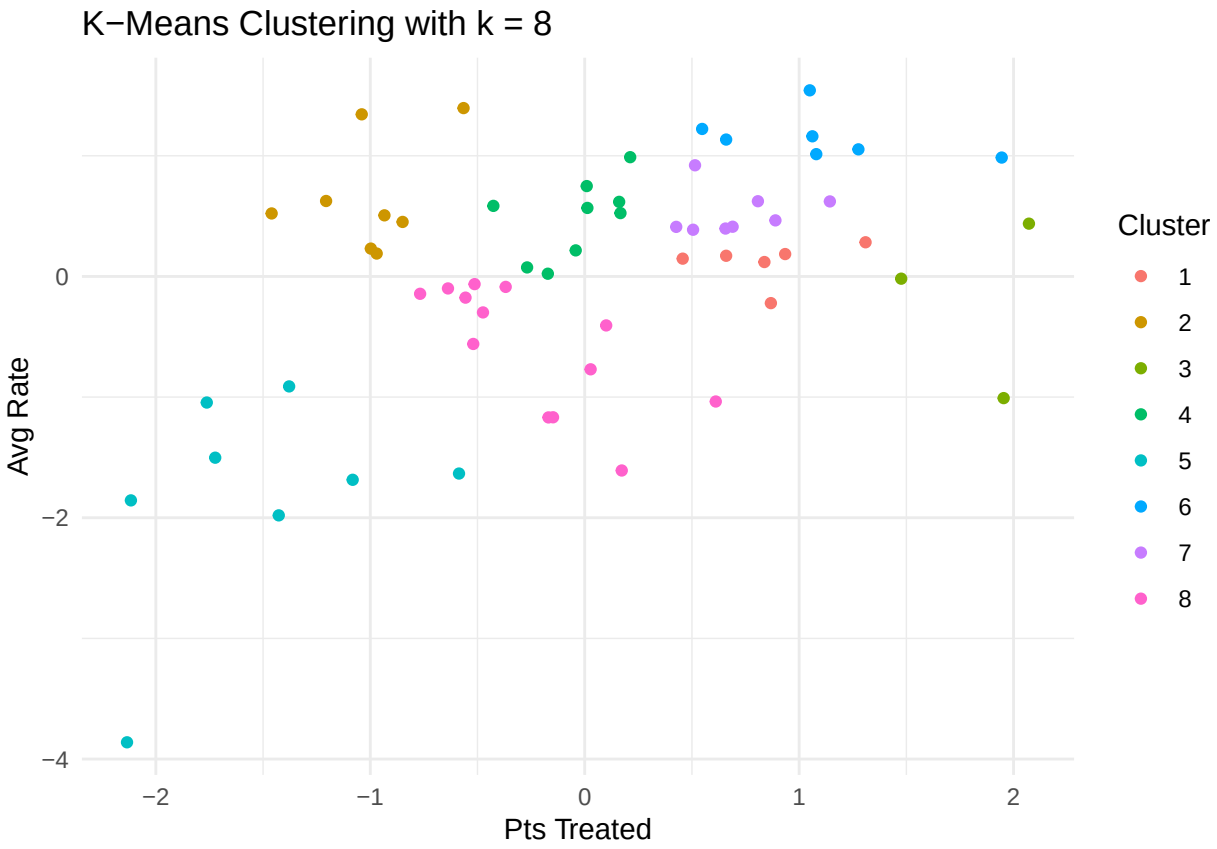

K−Means Clustering with k = 3
Avg Rate
0
−2
−4
−2
−1
0
1
2
Pts Treated
Cluster
1
2
3

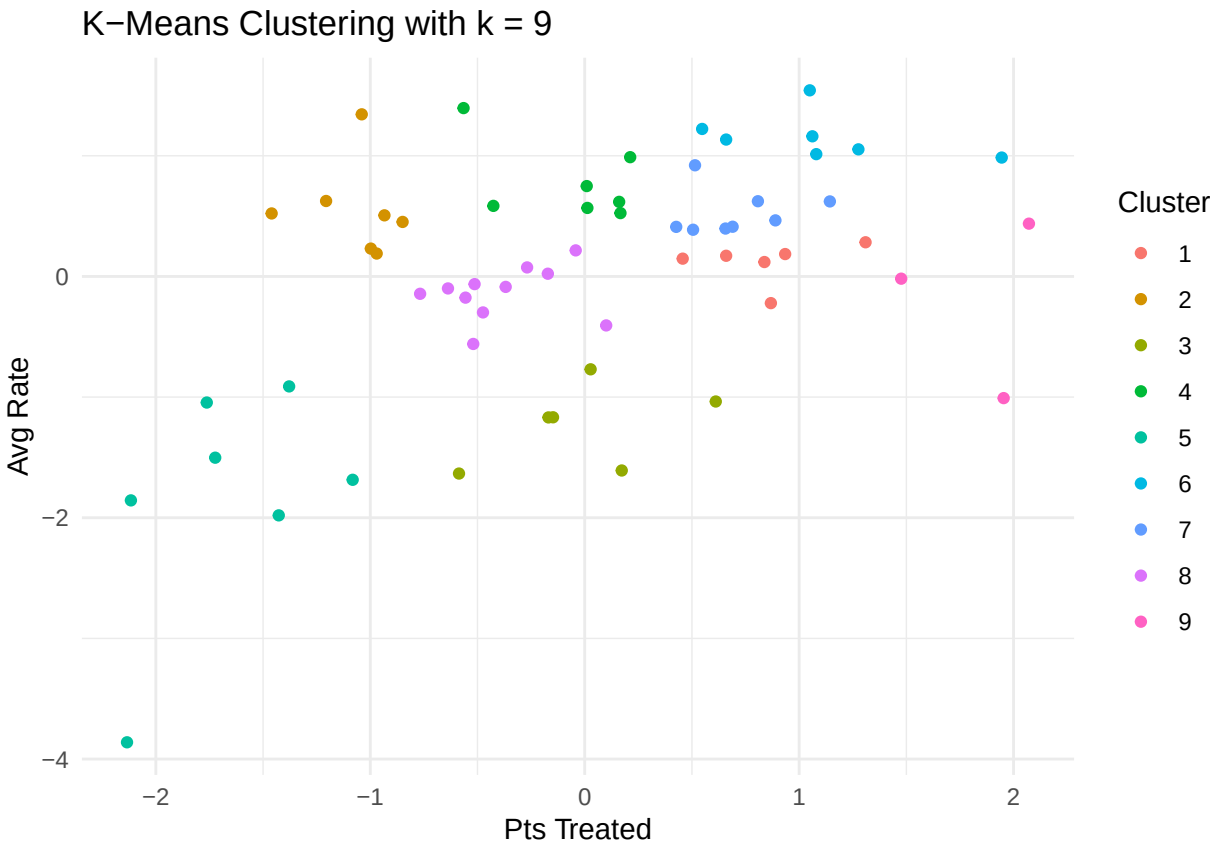

K−Means Clustering with k = 4
Avg Rate
0
−2
−4
−2
−1
0
1
2
Pts Treated
Cluster
1
2
3
4

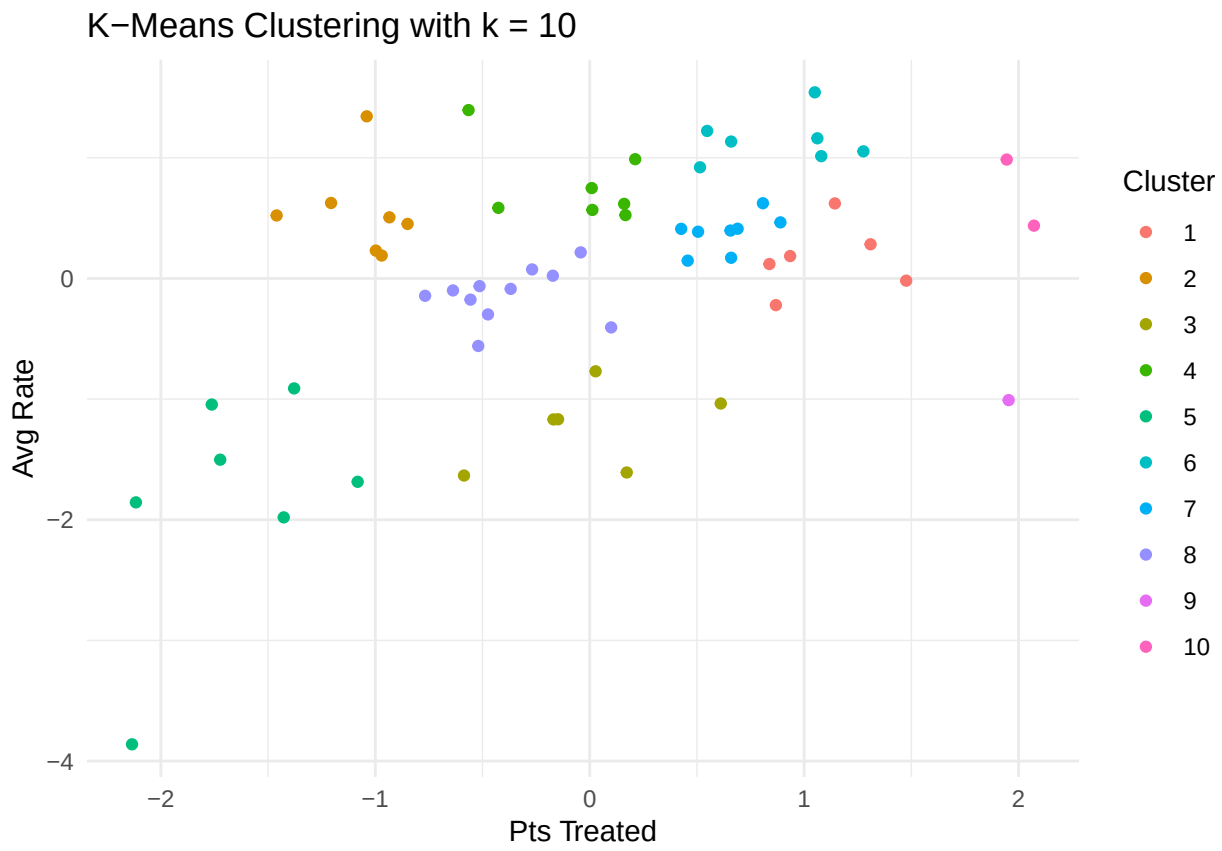

K−Means Clustering with k = 5
Avg Rate
Pts Treated
0
−2
−4
−2
−1
0
1
2
Cluster
1
2
3
4
5

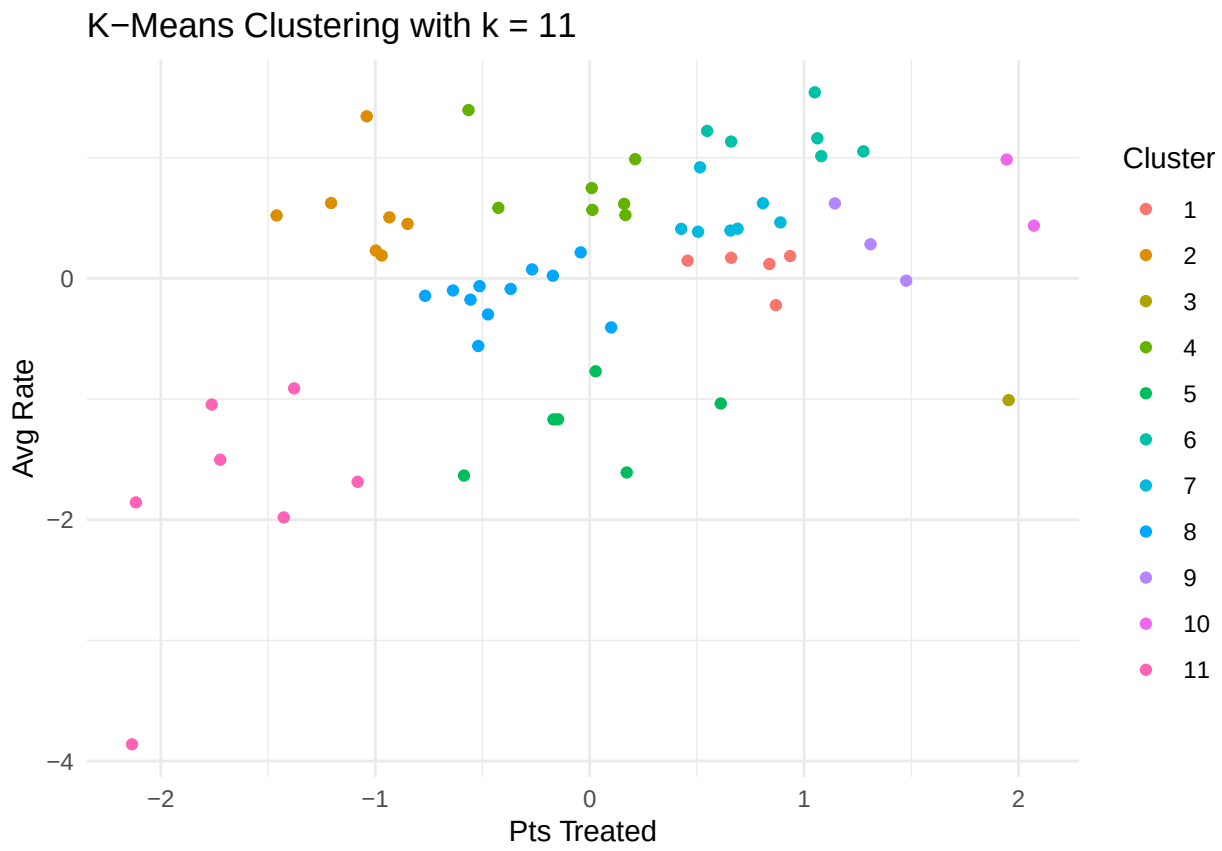

K−Means Clustering with k = 6
Avg Rate
Pts Treated
0
−2
−4
−2
−1
0
1
2
Cluster
1
2
3
4
5
6

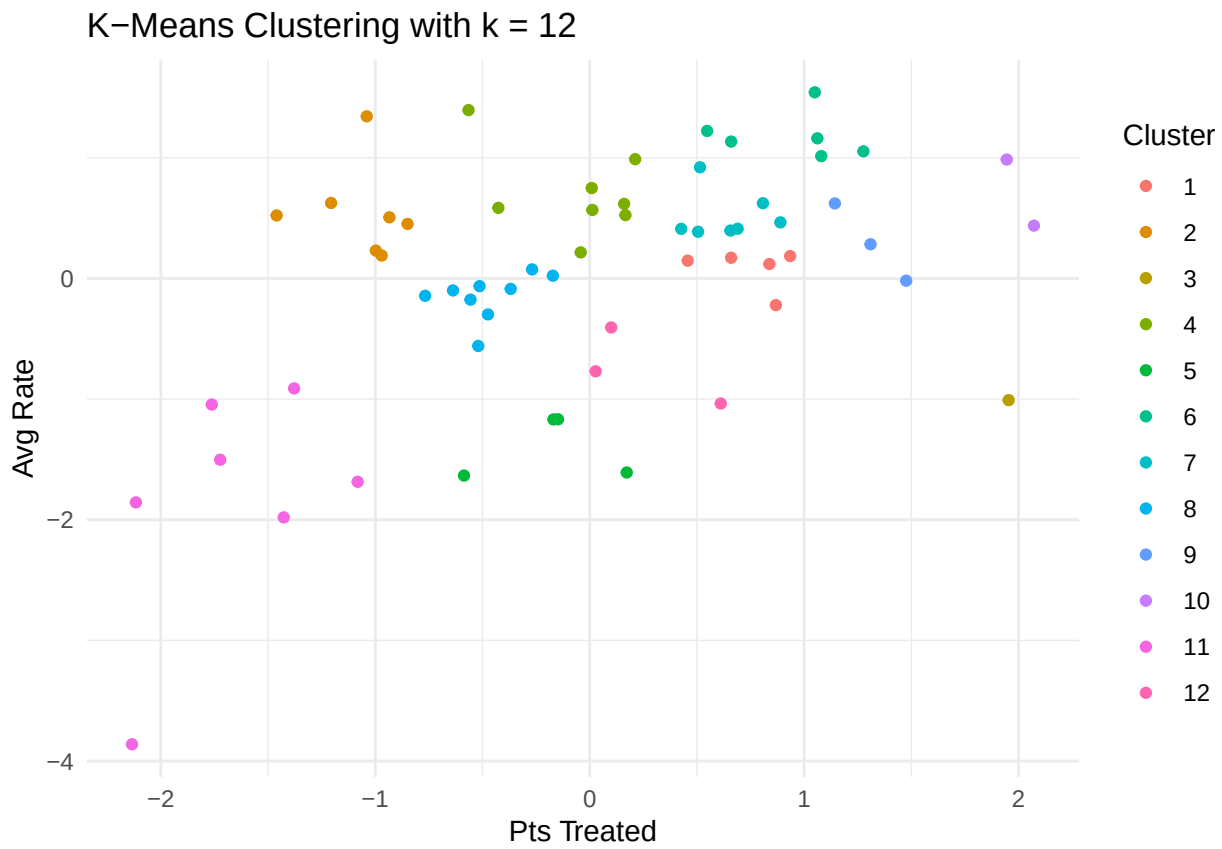

K−Means Clustering with k = 7
Avg Rate
Pts Treated
0
−2
−4
−2
−1
0
1
2
Cluster
1
2
3
4
5
6
7


K−Means Clustering with k = 8
Avg Rate
Pts Treated
0
−2
−4
−2
−1
0
1
2
Cluster
1
2
3
4
5
6
7
8


K−Means Clustering with k = 9
Avg Rate
Pts Treated
0
−2
−4
−2
−1
0
1
2
Cluster
1
2
3
4
5
6
7
8
9


K−Means Clustering with k = 10
Avg Rate
Pts Treated
0
−2
−4
−2
−1
0
1
2
Cluster
1
2
3
4
5
6
7
8
9
10

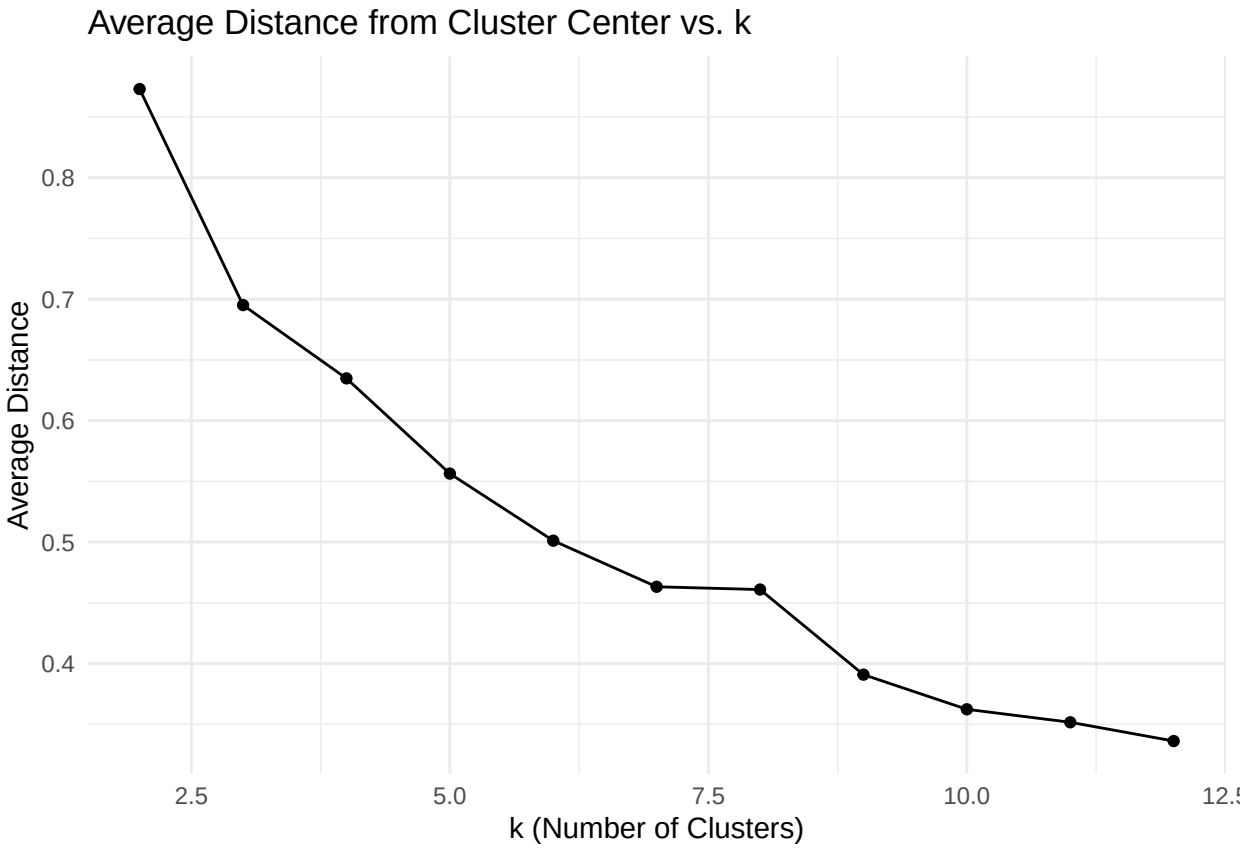

K−Means Clustering with k = 11
Avg Rate
Pts Treated
0
−2
−4
−2
−1
0
1
2
Cluster
1
2
3
4
5
6
7
8
9
10
11



Again, referring to the prior week's exercise, I also want to use the elbow point method to determine the best number of clusters.

```
# creating a dataframe to store the average distances
average_distances <- data.frame(k = integer(), average_distance = numeric())

for (k in 2:12) {
  # extracting the results for each k
  kmeans_result <- kmeans_results[[as.character(k)]]
  centers <- kmeans_result$centers
  clusters <- kmeans_result$cluster

  # calculating the distances and averaging
  distances <- numeric()
  for (i in 1:nrow(clinician_cluster_data)) {
    point <- clinician_cluster_data[i, ]
    center <- centers[clusters[i], ]
    distances <- c(distances, sqrt(sum((point - center)^2)))}
  average_distance <- mean(distances)

  # adding each to the dataframe
  average_distances <- rbind(average_distances, data.frame(k, average_distance))}

# plotting the average distances to examine them visually
ggplot(average_distances, aes(x = k, y = average_distance)) +
  geom_line() +
  geom_point() +
```

```
  labs(title = "Average Distance from Cluster Center vs. k",
       x = "k (Number of Clusters)",
       y = "Average Distance") +
  theme_minimal()
```



Although this graph is less pronounced, I think k = 3 is likely the most appropriate cluster choice.

```
# adding cluster assignment back to the original data
chosen_k <- 3
clinician_data$cluster <- kmeans_results[[as.character(chosen_k)]]$cluster
```

Now... let's take a look at the different clusters:

```
clinician_data %>%
  group_by(cluster) %>%
  summarise(
    avg_pts_treated = mean(pts_treated),
    avg_performance_rate = mean(avg_rate))
```

```
## # A tibble: 3 x 3
##   cluster avg_pts_treated avg_performance_rate
##     <int>           <dbl>                <dbl>
## 1       1            559.                0.714
## 2       2            350.                0.521
## 3       3           1035.                0.738
```

Regression Analysis

I have given this section a lot of thought and have considered both using the clusters and alternatively, creating a separate, but simpler regression on the data points themselves so I could then compare the results.

Knowledge Gap

The piece of knowledge I lack in this arena is how a comparison of the two models would be able to give me more information versus simply providing further statistical nuance that might dilute my own understanding at this early stage.

Because of this knowledge gap, I have decided to try to create a regression analysis of only the clusters and do my best to interpret the results. Essentially, I am trusting the process.

```
# merging dataframes so I have a df to work with
regression_merge <- merge(performance, clinician_data, by = c("clinician_name", "clinic_name", "role"))

# converting clusters to a factor
regression_merge$cluster <- as.factor(regression_merge$cluster)
```

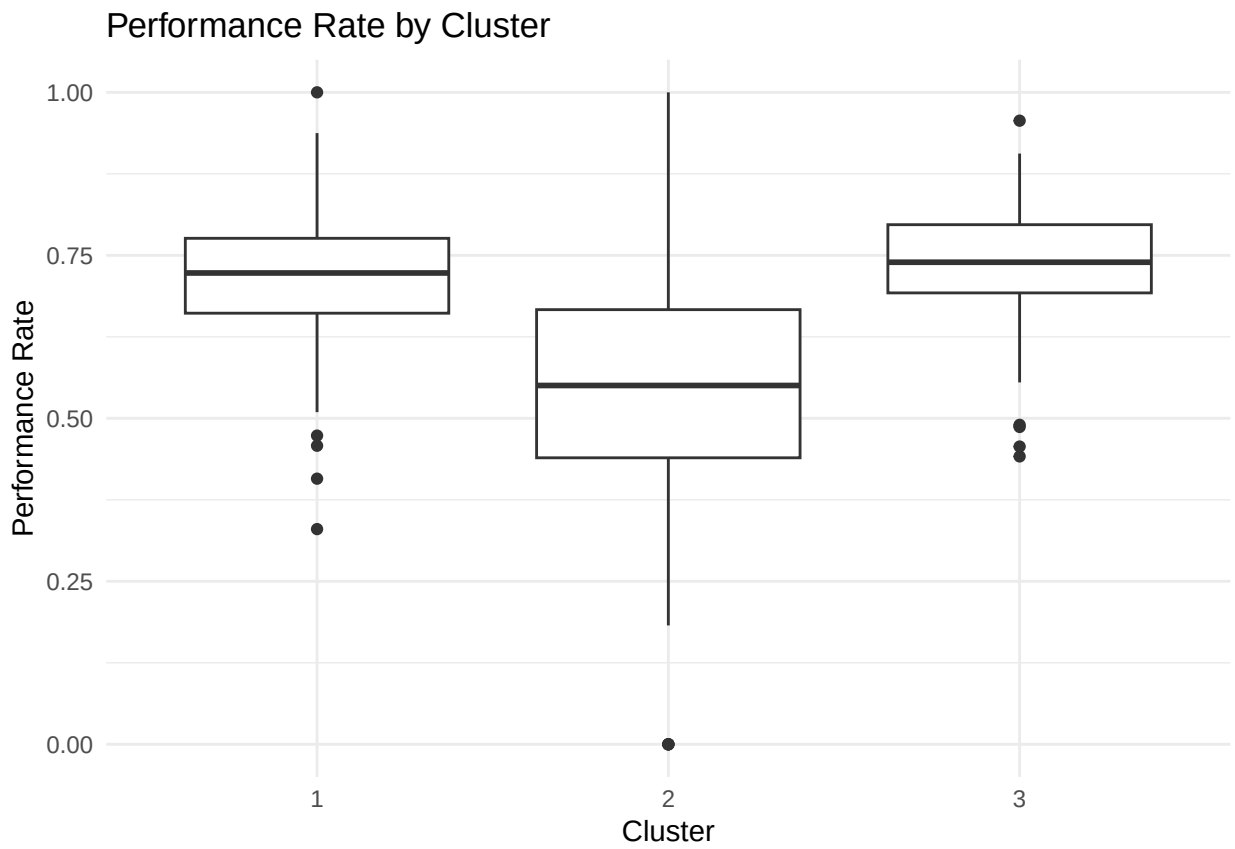
Always a good idea to review a summary of the dataframe I made:

```
summary(regression_merge)
```

```
##  clinician_name     clinic_name           role             measure_id
##  Length:372         Length:372         Length:372         Min.   : 47.0
##  Class :character   Class :character   Class :character   1st Qu.:112.0
##  Mode  :character   Mode  :character   Mode  :character   Median :123.5
##                                                           Mean   :158.5
##                                                           3rd Qu.:236.0
##                                                           Max.   :309.0
##       num               den              rate         pts_treated
##  Min.   :   0.00   Min.   :   0.0   Min.   :0.0000   Min.   :  16.0
##  1st Qu.:  97.75   1st Qu.: 149.0   1st Qu.:0.6421   1st Qu.: 511.0
##  Median : 211.00   Median : 311.0   Median :0.7178   Median : 728.5
##  Mean   : 275.12   Mean   : 377.0   Mean   :0.6895   Mean   : 721.9
##  3rd Qu.: 407.25   3rd Qu.: 541.8   3rd Qu.:0.7778   3rd Qu.: 950.0
##  Max.   :1191.00   Max.   :1407.0   Max.   :1.0000   Max.   :1407.0
##     avg_rate      cluster
##  Min.   :0.3000   1:150
##  1st Qu.:0.6485   2: 66
##  Median :0.7100   3:156
##  Mean   :0.6895
##  3rd Qu.:0.7519
##  Max.   :0.8451
```

Okay - now let's plot measure performance by cluster.

```
ggplot(regression_merge, aes(x = cluster, y = rate)) +
  geom_boxplot() +
  labs(title = "Performance Rate by Cluster", x = "Cluster", y = "Performance Rate") +
  theme_minimal()
```



I also want to check the correlations between my numeric values.

```
# checking correlations between numerical variables
cor(regression_merge[,sapply(regression_merge, is.numeric)])
```

```
##              measure_id         num         den       rate pts_treated
## measure_id   1.00000000 -0.03846039 -0.05297538 0.02044831   0.0000000
## num         -0.03846039  1.00000000  0.98299296 0.44335509   0.5702569
## den         -0.05297538  0.98299296  1.00000000 0.35647502   0.5921168
## rate         0.02044831  0.44335509  0.35647502 1.00000000   0.3345813
## pts_treated  0.00000000  0.57025692  0.59211684 0.33458126   1.0000000
## avg_rate     0.00000000  0.37288665  0.31319132 0.68110547   0.4912327
##              avg_rate
## measure_id  0.0000000
## num         0.3728867
## den         0.3131913
## rate        0.6811055
## pts_treated 0.4912327
## avg_rate    1.0000000
```

Some of these correlations are highly intuitive and expected, like the correlation between average performance rates and individual performance rates, or the lack of correlation to measure ID. If I were doing this again, I might adjust my model to remove the factors that would be intuitive to any Quality Director.

There are some interesting ones, however, like the relationship between the denominators or patients treated and individual performance.

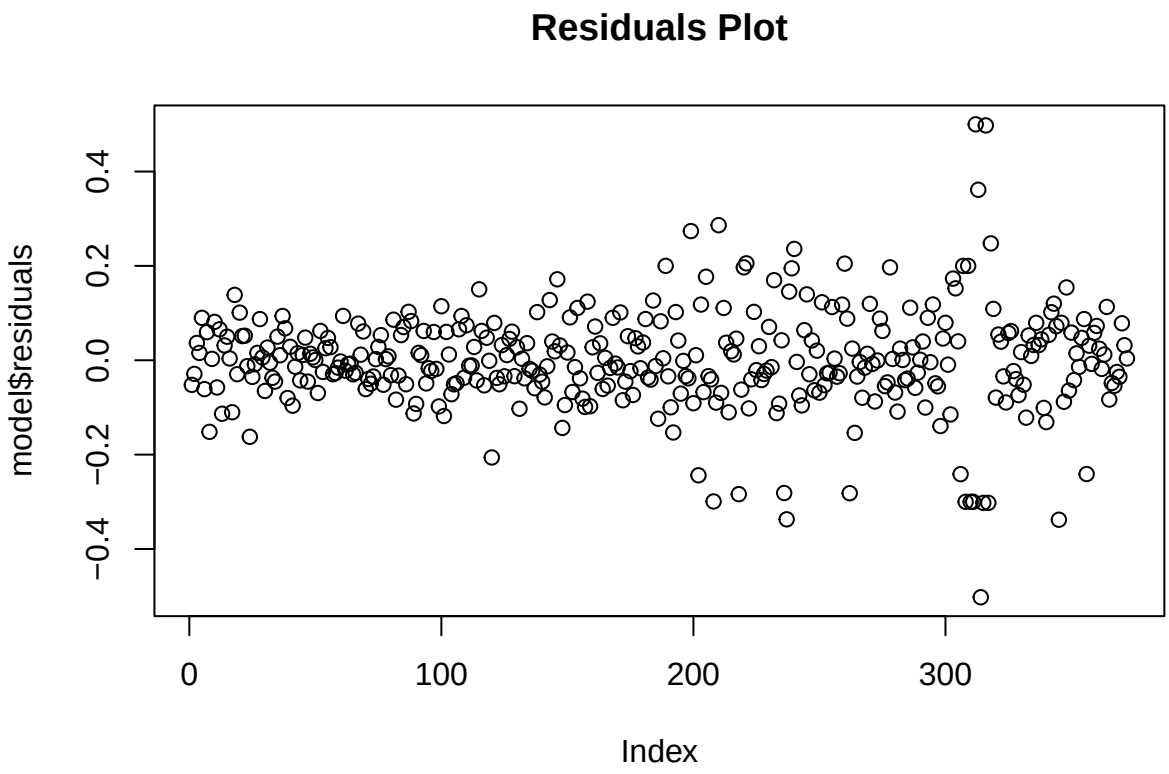
Here is the final linear regression model with the clusters as a factor.

```r
# building the model
model <- lm(rate ~ pts_treated + avg_rate + cluster + clinic_name + role, data = regression_merge)

# viewing the model
summary(model)
```

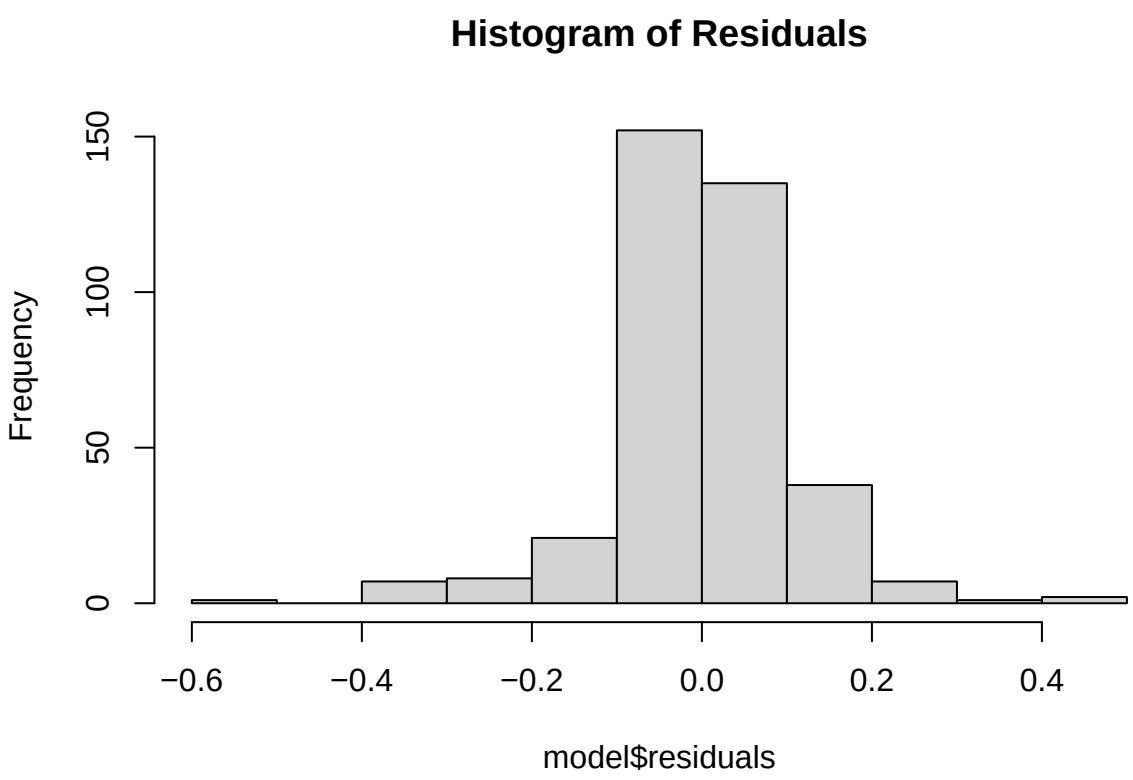
```
##
## Call:
## lm(formula = rate ~ pts_treated + avg_rate + cluster + clinic_name +
##     role, data = regression_merge)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.50227 -0.04927 -0.00197  0.05527  0.50000
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                       9.337e-16  8.211e-02   0.000        1
## pts_treated                      -1.755e-20  3.800e-05   0.000        1
## avg_rate                          1.000e+00  1.081e-01   9.251   <2e-16 ***
## cluster2                         -1.356e-16  2.880e-02   0.000        1
## cluster3                          1.534e-16  2.379e-02   0.000        1
## clinic_nameCulinary Arts Academy -3.071e-17  2.714e-02   0.000        1
## clinic_nameCulinary Institute     2.335e-18  2.415e-02   0.000        1
## clinic_nameEscoffier School       2.398e-17  2.067e-02   0.000        1
## clinic_nameHattori Nutrition      1.224e-16  2.708e-02   0.000        1
## clinic_nameKendall College        2.929e-18  2.636e-02   0.000        1
## clinic_nameLa Cuisine Paris       4.629e-17  2.322e-02   0.000        1
## clinic_nameLe Cordon Bleu         6.044e-17  2.287e-02   0.000        1
## roleNP                            1.769e-17  1.668e-02   0.000        1
## rolePA                            7.479e-18  2.364e-02   0.000        1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1097 on 358 degrees of freedom
## Multiple R-squared:  0.4639, Adjusted R-squared:  0.4444
## F-statistic: 23.83 on 13 and 358 DF,  p-value: < 2.2e-16
```

Okay – and now I want to plot my residuals and check for normality.

```r
# Plotting residuals
plot(model$residuals, main = "Residuals Plot")
```




```r
hist(model$residuals, main = "Histogram of Residuals")
```



How My Approach Addresses the Problem:

This has been a fairly comprehensive approach where I am trying to apply various statistical and machine learning techniques from this term to better understand clinician quality of care and make recommendations to the Quality Director accountable for these 8 clinics. I have been able to assess current performance, have recommendations for both target and stretch goals for the future year, I will be able to clearly provide information to the Quality Director regarding the statistically significant impacts that role, clinic, and number of patients seen has on clinician performance, and will compile these into actionable insights for next steps in driving quality of care forward.

Implications and Safe Recommendations

There was a very wide range of results and implications found here, and I believe the strongest implications actually came from the statistical analyses. My immediate thought is that seeing the standard deviation of performance was compelling, with the higher deviation potentially suggesting there are opportunities for improvement and standardization of processes for some measures. My assumption is that for the measures with a lower deviation, performance is more consistent and closer to the mean, which could indicate well-established procedures and/or guidelines.

Some of the recommendations I would make to the Quality Director now include: * Set the 2024 Target Goals on the 75th Percentile of current performance. * Share the 2024 Stretch Goals, set on the 90th Percentile of current performance with your high-performers and clinicians expressing more interest in becoming leaders or champions of quality in each clinic. * Perform site visits of the clinics associated with a lower performance rate and engage with the healthcare consultants and informaticists on your team to examine the clinic-specific

challenges found there. * Perform site visits of the highest performing clinic to examine what behaviors and factors may be scale-able and replicate-able at other practices. * Review the physician oversight policies regarding NP and PA autonomy, education, and mentorship in the clinics. Ensure that these mid-level clinicians are provided with the appropriate support across the organization. * Use a well-developed and vetted communication plan to "push back" against incorrect narratives throughout the clinics, such as: ** NPs and PAs do not have better performance, and quality of care is more than just electronic documentation. It is bedside manner, patient/provider rapport, and clinial experience paired with helping the patient think critically with us regarding their plan of care. ** Some clinics are not more "quality-forward" than other clinics, and there are a variety of factors which may positively or negatively impact a given clinic's quality scores. We will be delving into those factors during some upcoming site visits, and staff are encouraged to speak openly with the consultants so we can work together to solve our unique pain-points. ** Patient panel size is something that is only recently being analyzed in relation to quality, and we have extensive metrics regarding patient access, patient risk scores, growth, and patient satisfaction. We will be adding quality to this analytic matrix in 2024 and we thank the staff who brought this idea to our attention. ** Finally, strong performance in primary care quality metrics are a group effort. Everyone is here because we care about our patients, our community, and our co-workers. We have the right staff to do our best work and we look forward to hearing your input as our improvement projects re-launch for the 2024 calendar year.

Limitations and Thoughts about My Cluster Model

One major limitation is that my performance dataset only reviews 2023 data. While the quality metrics are focused on each calendar year, patients are considered "active" on a clinician's panel if they've had a visit anytime in the past three years. Seeing three years of performance could certainly lead to a stronger model.

Another limitation I discovered was one of my own knowledge gap. I suspect I had too many factors in my regression, because otherwise I'm not fully certain why my statistical analyses provided significant p-values and insightful coefficients, while the results of the regression analysis that included the clusters, in some cases, provided the opposite result.

The cluster model, in fact, indicated that the number of patients treated, clinician role, and clinics as variables did NOT significantly contribute to variations in the performance rate. My multiple R-squared value suggests that about 46.39% of the variability in the performance rate is explained by the model but the model as a whole IS statistically significant.

Concluding Remarks

This was a fantastic exercise and I only hope I met enough of the requirements to earn a decent grade. I have certainly learned a lot here, and although I would like to take this final model to a professional statistician and ask about a hundred questions, this is a body of work I would love to take additional iterations on and really refine both the models I could use AND my understanding of how they function.

Given enough time and resources, I believe troubleshooting and coding can always be solved. Developing an intuitive understanding of how to apply the coding to advanced statistical models and machine learning, however, is a more nuanced and difficult task.

Thanks so much for your class this term! As always, any comments or further questions are welcomed. :)