UNICEF Project Exploration - STA130 Winter 2024 - Group D

Brad Cho, Tanay Langhe, Christopher Li, Dhruv Patel, Chaoxun Tan

Final Project Overview: Identifying Opportunities to Accelerate Progress on Sustainable Development Goals (SDG)

How do disparities in countries' socio-economic status relate to their progress towards reaching the SDGs?

How do we measure a country's socio-economic status?

To measure a country's socio-economic status, we look at its GDP collected by the World Bank in 2022/2021. In order to determine the socioeconomic brackets, we put all of the countries' gdp onto a log10 scale and use the k-means algorithm to split them into 4 distinct groups: low, medium, high, and very-high.

Cleaning Data

```
# To make sure everything is reproducable
set.seed(69420)

# Cleaning GDP Data
gdp_data <- read_csv("data/country_gdps.csv") %>%
  mutate(log_recent = case_when(!is.na(`2022`) ~ log10(`2022`),
                                .default = log10(`2021`))
                                ) %>%
  mutate(country_code = `Country Code`) %>%
  select(country_code, log_recent) %>%
  filter(!is.na(log_recent))
```

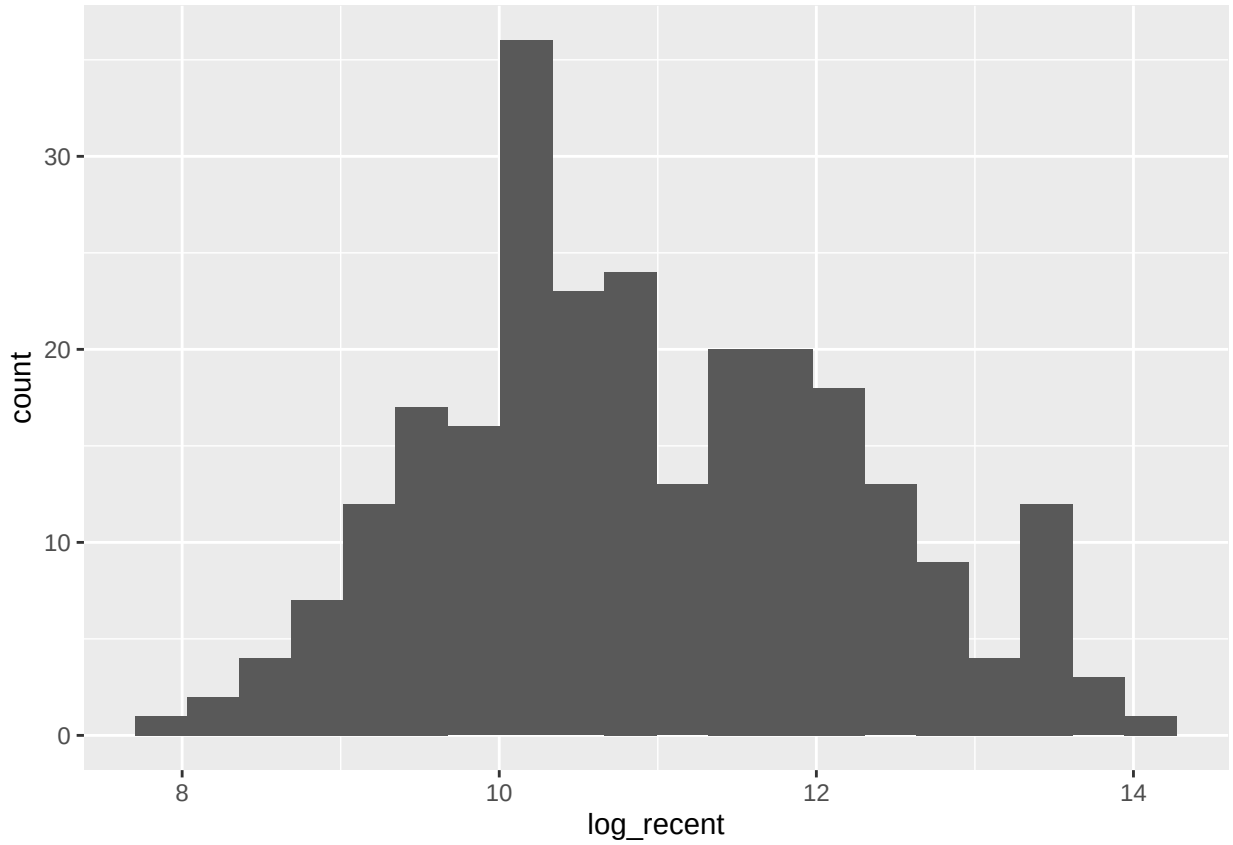
```
## New names:
## Rows: 266 Columns: 68
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (4): Country Name, Country Code, Indicator Name, Indicator Code dbl (63): 1960,
## 1961, 1962, 1963, 1964, 1965, 1966, 1967, 1968, 1969, 1970, ... lgl (1): ...68
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...68`
```

Calculate and remove outliers (1.5 IQR) - This is not used!

```
if (FALSE) {
  median = median(gdp_data$log_recent)
  iqr = IQR(gdp_data$log_recent)

  gdp_data <- gdp_data %>% filter(median - 1.5 * iqr < log_recent & log_recent < median + 1.5 * iqr)
}
```

Plotting GDP Data

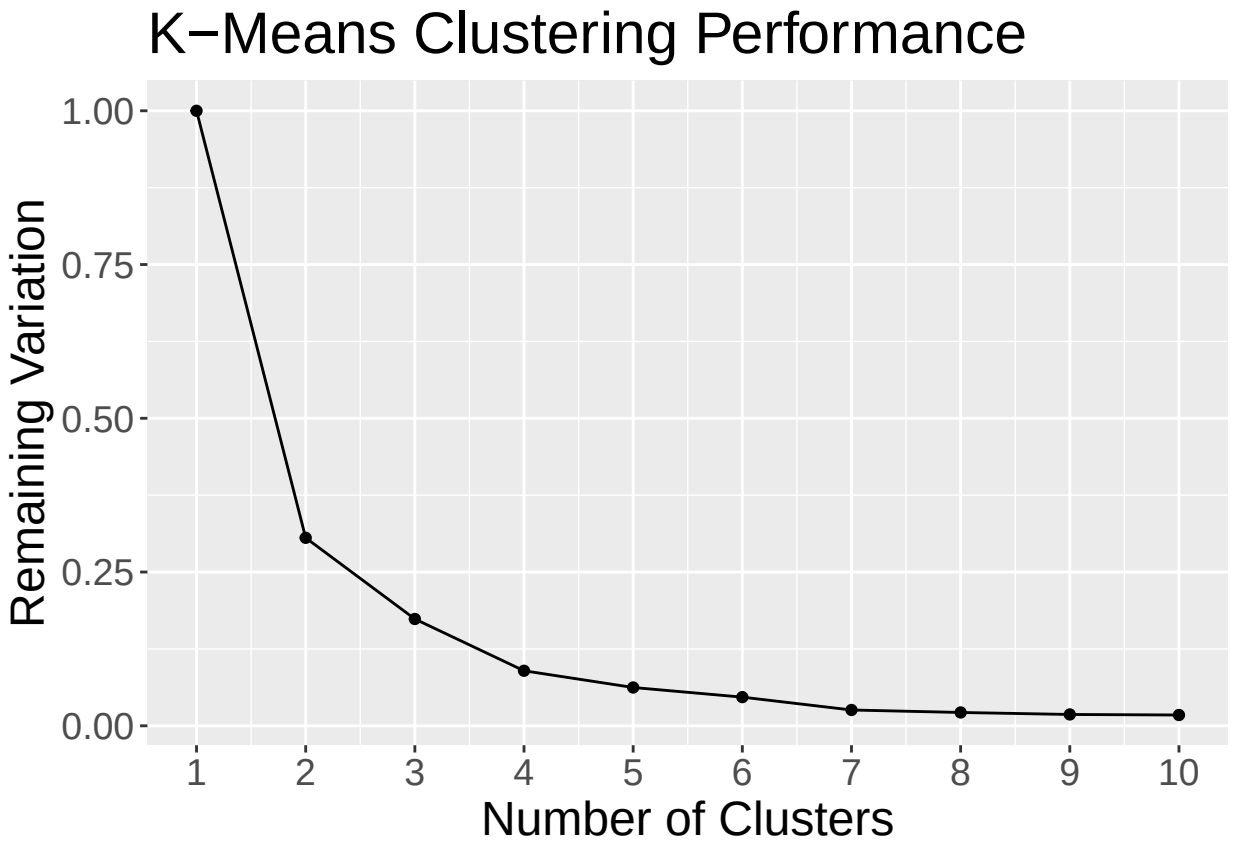
```
gdp_data %>% ggplot(aes(x = log_recent)) + geom_histogram(bins=20)
```



Elbow Method to figure out the best number of clusters, we determined 3 or 4 clusters is best

```
explained_ss <- rep(NA, 10)
for(k in 1:10){
  # run k-means on the data
  clustering <- kmeans(gdp_data$log_recent, k)
  explained_ss[k] <- clustering$betweenss / clustering$totss
}

ggplot() +
  aes(x=1:10, y=1-explained_ss) +
  geom_line() +
  geom_point() +
  labs(x="Number of Clusters",
       y="Remaining Variation",
       title="K-Means Clustering Performance") +
  theme(text=element_text(size=18)) +
  scale_x_continuous(breaks=1:10) +
  scale_x_continuous(breaks=1:10)
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

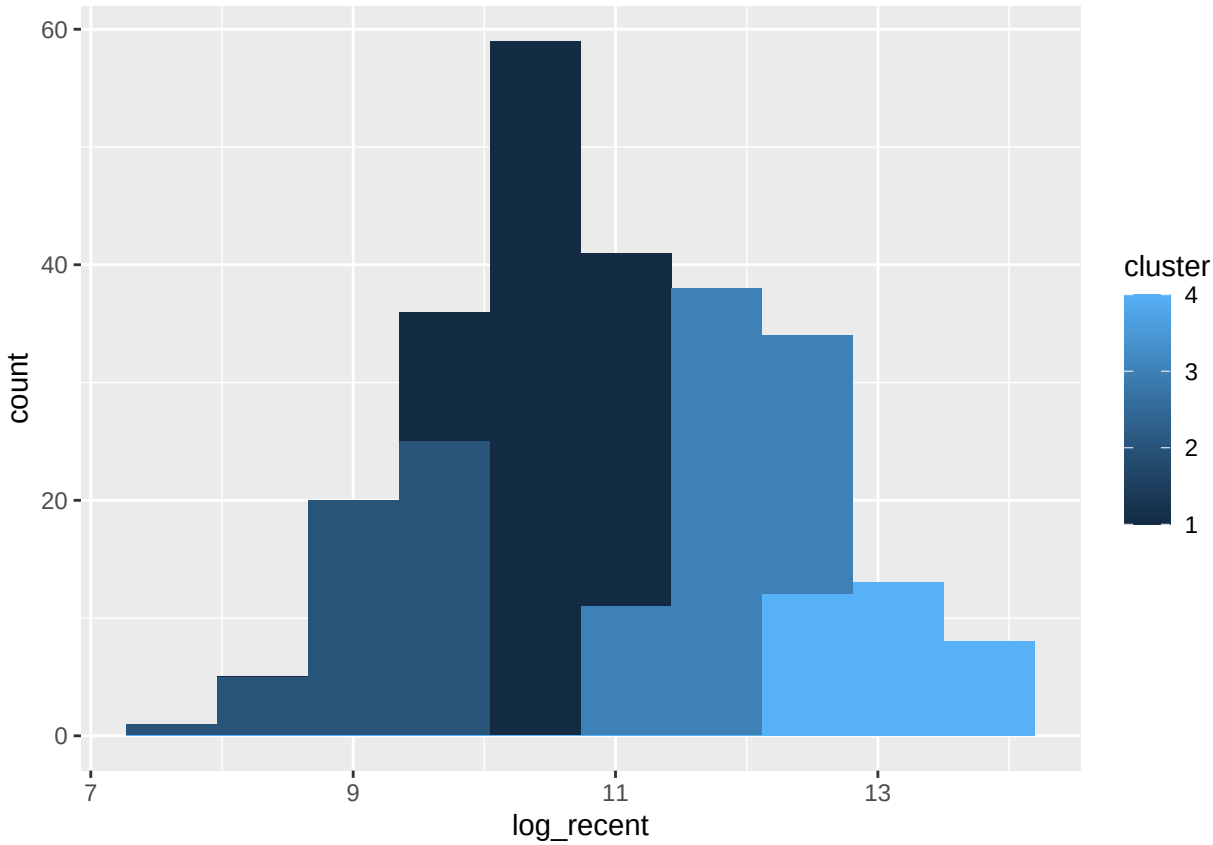


Actually performing k-means on the data, lower number group indicate lower gdp group

```
clustering <- kmeans(gdp_data$log_recent, 4)
gdp_data <- gdp_data %>% mutate(cluster = clustering$cluster) %>% mutate(cluster = case_when(cluster ==
```

Plotting k-means

```
gdp_data %>% ggplot(aes(x=log_recent, group=cluster, fill=cluster)) + geom_histogram(bins = 10)
```



Writing cleaned CSV to a file, for the purpose of this rmd file, we will just be setting it to a variable

```
# write.csv(gdp_data, "data/clean_country_gdps.csv", row.names=FALSE)
clean_country_gdps <- gdp_data
```

Research Question 1

In the second-lowest socio-economic bracket, if we group countries based on location, is the average distance towards the SDG's significant between the groups?

Google's data-set uses Alpha-2 code, I need to convert to Alpha-3 since UNICEF uses Alpha-3

```
country_locations <- read_csv("data/country_locations.csv") %>% select(-name)
```

```
## Rows: 245 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): country, name
## dbl (2): latitude, longitude
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
country_codes <- read_csv("data/country_codes.csv") %>% filter(`Alpha-2 Code` %in% country_locations$co
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
## Rows: 249 Columns: 4
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): Country, Alpha-2 Code, Alpha-3 Code, Numeric
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
country_locations <- country_locations %>% filter(country %in% country_codes$alpha_2) %>% mutate(alpha_

merged_data <- merge(country_locations, country_codes, by="alpha_2") %>% mutate(country_code = alpha_3)
```

Writing cleaned CSV to a file, for the purpose of this rmd file, we will just be setting it to a variable

```
# write.csv(merged_data, "data/clean_country_locations.csv", row.names=FALSE)
clean_country_locations <- merged_data
```

Doing k-means on the location

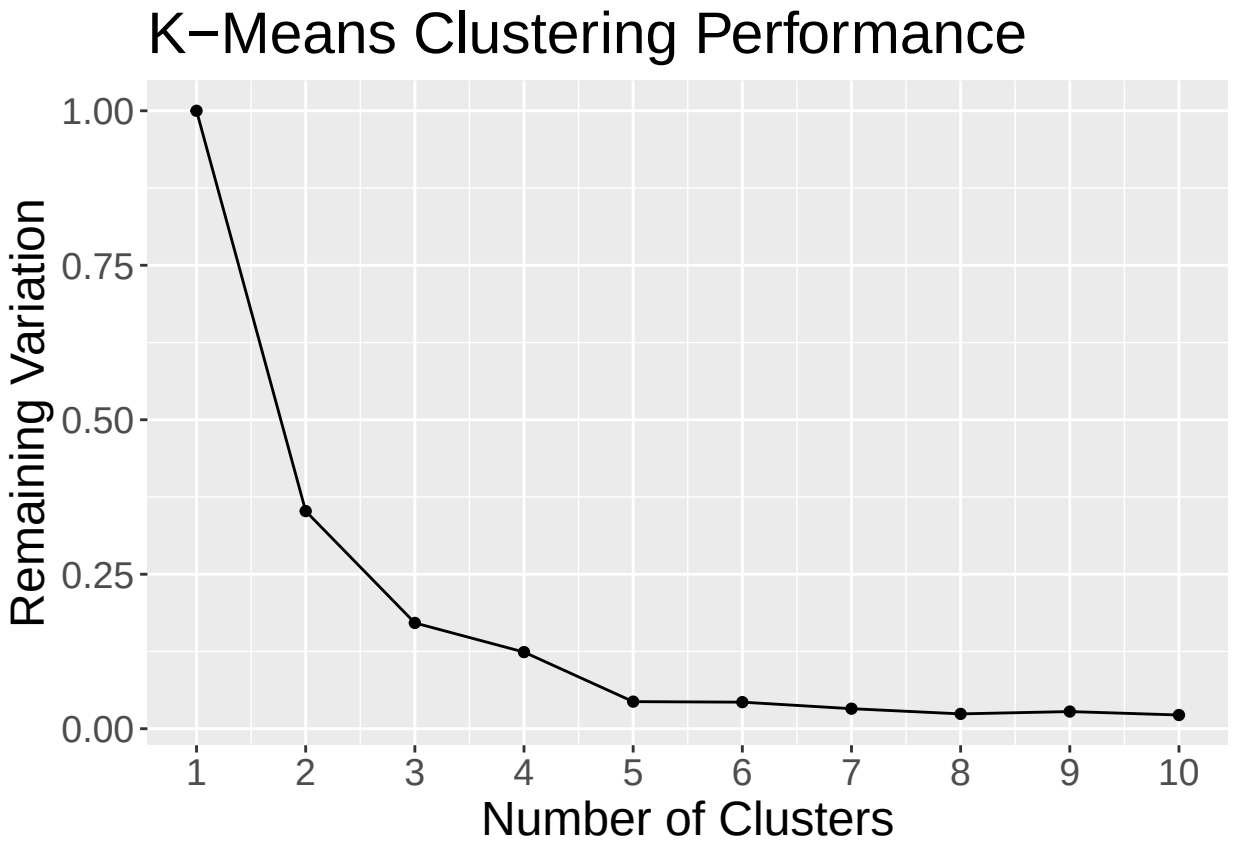
Cleaning datasets

```
set.seed(69420)

country_locations = data.frame(clean_country_locations)
gdp_data = data.frame(clean_country_gdps) %>% filter(cluster==2 & country_code %in% country_locations$c

merged_data = merge(gdp_data, country_locations, by="country_code")
```

Elbow Method to figure out the best number of clusters, we determined 3 or 4 clusters is best

```
explained_ss <- rep(NA, 10)
for(k in 1:10){
  # run k-means on the data
  clustering <- kmeans(merged_data %>% select(longitude, latitude), k)
  explained_ss[k] <- clustering$betweenss / clustering$totss
}

ggplot() +
  aes(x=1:10, y=1-explained_ss) +
  geom_line() +
  geom_point() +
  labs(x="Number of Clusters",
       y="Remaining Variation",
       title="K-Means Clustering Performance") +
  theme(text=element_text(size=18)) +
  scale_x_continuous(breaks=1:10) +
  scale_x_continuous(breaks=1:10)
```

```
## Scale for x is already present.
## Adding another scale for x, which will replace the existing scale.
```

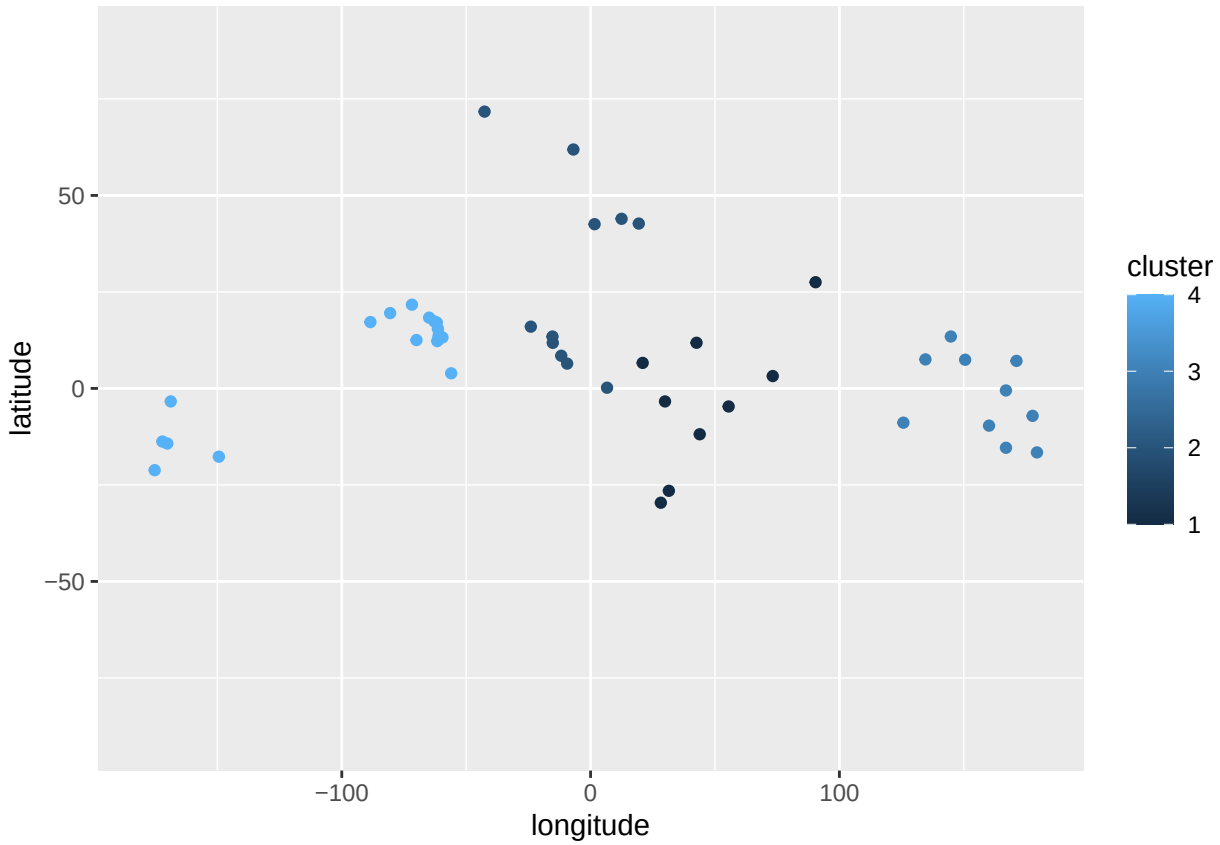


Doing k-means with 4 clusters

```
clustering <- kmeans(merged_data %>% select(longitude, latitude), 4)
merged_data <- merged_data %>% mutate(cluster = clustering$cluster)
```

Plotting k-means

```
# TODO - add a map projection behind this?
merged_data %>% ggplot(aes(x = longitude, y = latitude, color=cluster)) + geom_point() + xlim(-180, 180)
```



Writing cleaned CSV to a file, for the purpose of this rmd file, we will just be setting it to a variable

```
# write.csv(merged_data, "data/country_locations_cluster.csv", row.names=FALSE)
country_locations_cluster <- merged_data
```

Doing hypothesis testing on the different groups

```
country_locations <- data.frame(country_locations_cluster)
sdr_goals <- read_csv("data/sdr_fd5e4b5a.csv") %>% mutate(country_code = `Country Code ISO3`, score = `
```

```
## New names:
## Rows: 206 Columns: 59
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (36): Goal 1 Dash, Goal 1 Trend, Goal 2 Dash, Goal 2 Trend, Goal 3 Dash,... dbl
## (23): ...1, Goal 1 Score, Goal 2 Score, Goal 3 Score, Goal 4 Score, Goal...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```
merged_data <- merge(sdr_goals, country_locations, by="country_code") %>% filter(!is.na(score))
```

Summaries

```
print(merged_data %>% group_by(cluster) %>% summarize(m = mean(score)))
```

```
## # A tibble: 4 x 2
##   cluster     m
##     <int> <dbl>
```

```
## 1       1  56.9
## 2       2  61.1
## 3       3  72.9
## 4       4  67.4
```

```
for (i in 1:3) {
  for (j in (i+1):4) {
    print(paste("Group", i, "with Group", j));
    model1 <- lm(score ~ cluster, data = merged_data %>% filter(cluster == i | cluster == j))
    print(summary(model1)$coefficients)
  }
}
```

```
## [1] "Group 1 with Group 2"
##              Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 52.641667   7.838020 6.7161943 2.145521e-05
## cluster      4.245833   5.184363 0.8189691 4.287691e-01
## [1] "Group 1 with Group 3"
##             Estimate Std. Error  t value    Pr(>|t|)
## (Intercept) 48.88125   7.671442 6.371846 0.000377269
## cluster      8.00625   5.581794 1.434351 0.194597321
## [1] "Group 1 with Group 4"
##              Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 53.383333   4.764193 11.20512 1.377421e-06
## cluster      3.504167   2.111503  1.65956 1.313750e-01
## [1] "Group 2 with Group 3"
##             Estimate Std. Error  t value  Pr(>|t|)
## (Intercept) 37.60000  19.063333 1.972373 0.1056030
## cluster     11.76667   8.779927 1.340178 0.2378611
## [1] "Group 2 with Group 4"
##              Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 54.866667   6.998503 7.839772 0.0001036949
## cluster      3.133333   2.474345 1.266329 0.2459053356
## [1] "Group 3 with Group 4"
##             Estimate Std. Error   t value   Pr(>|t|)
## (Intercept)     89.4  10.888526  8.210478 0.01451204
## cluster         -5.5   2.884441 -1.906782 0.19680208
```

Question 2

For each socio-economic bracket, what is a range of plausible values for the success of education SDG's?

```
set.seed(694201)

cleangdps <- data.frame(clean_country_gdps)
cleanind <- read_csv("data/Very_clean_country_indicators.csv")
```

```
## Rows: 109 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): iso3
## dbl (4): sowc_education__completion_completion-rate-2013-2021-r_primary-educ...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

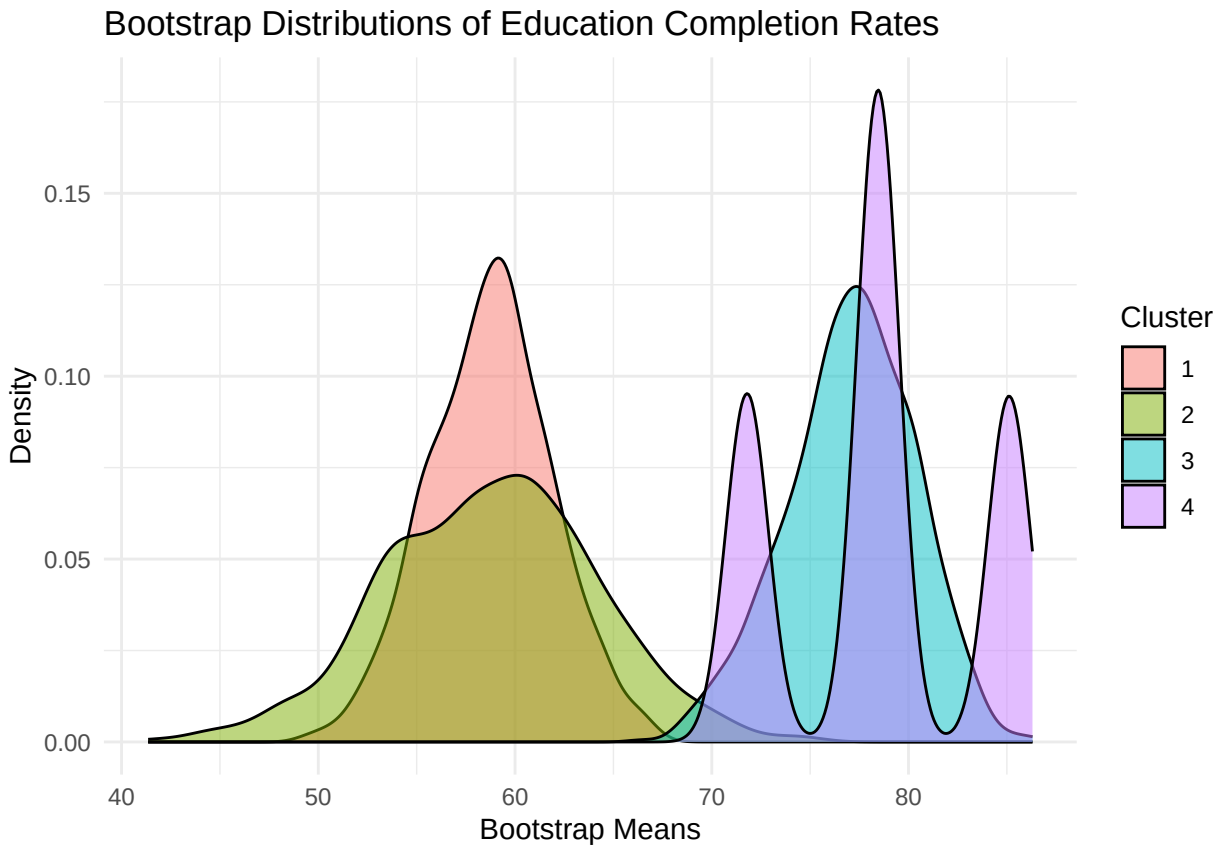
```r
merged_data <- merge(cleangdps, cleanind, by.x = "country_code", by.y = "iso3", all.x = TRUE)

# merging da data
bootstrap_ci <- function(data, n_bootstrap = 1000) {
  bootstrap_means <- replicate(n_bootstrap, mean(sample(data, replace = TRUE), na.rm = TRUE))
  ci <- quantile(bootstrap_means, probs = c(0.025, 0.975))
  list(mean = mean(bootstrap_means), ci_lower = ci[1], ci_upper = ci[2], bootstrap_means = bootstrap_mea
}
```

Doing bootstrapping

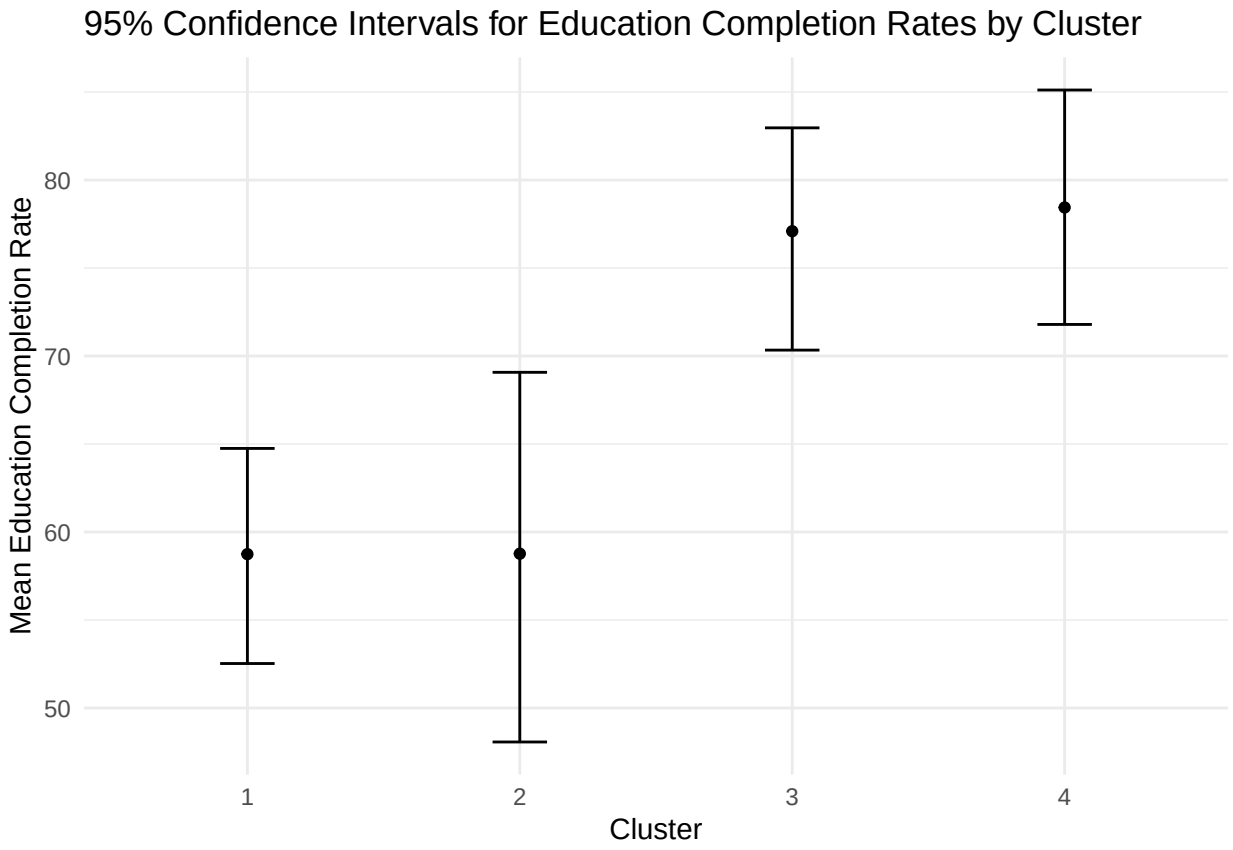
```r
# applying the bootstrap Function and preparing the data for plotting
bootstrap_results <- merged_data %>%
  filter(!is.na(sowc_education__completion_completion_rate_2013_2021_avg)) %>%
  group_by(cluster) %>%
  summarise(bootstrap_data = list(bootstrap_ci(sowc_education__completion_completion_rate_2013_2021_avg
            .groups = 'drop')
```

Plotting the results

```r
# Extracting results for plotting
bootstrap_distributions <- do.call(rbind, lapply(1:nrow(bootstrap_results), function(i) {
  data.frame(cluster = bootstrap_results$cluster[i],
             bootstrap_means = bootstrap_results$bootstrap_data[[i]]$bootstrap_means)
}))

ci_data <- do.call(rbind, lapply(1:nrow(bootstrap_results), function(i) {
  with(bootstrap_results$bootstrap_data[[i]],
       data.frame(cluster = bootstrap_results$cluster[i],
                  mean = mean,
                  ci_lower = ci_lower,
                  ci_upper = ci_upper))
}))
# ploting bootstrap distributions
ggplot(bootstrap_distributions, aes(x = bootstrap_means, fill = as.factor(cluster))) +
  geom_density(alpha = 0.5) +
  labs(title = "Bootstrap Distributions of Education Completion Rates",
       x = "Bootstrap Means",
       y = "Density",
       fill = "Cluster") +
  theme_minimal()
```




```r
# plotting for confidence intervals
ggplot(ci_data, aes(x = as.factor(cluster), y = mean)) +
  geom_point() +
  geom_errorbar(aes(ymin = ci_lower, ymax = ci_upper), width = 0.2) +
  labs(title = "95% Confidence Intervals for Education Completion Rates by Cluster",
       x = "Cluster",
       y = "Mean Education Completion Rate") +
  theme_minimal()
```



Question 3

How well does one's socio-economic status correlate with a country's progress towards the health SDGs?

We are using Goal 1 as a different measure of a country's socio-economic status

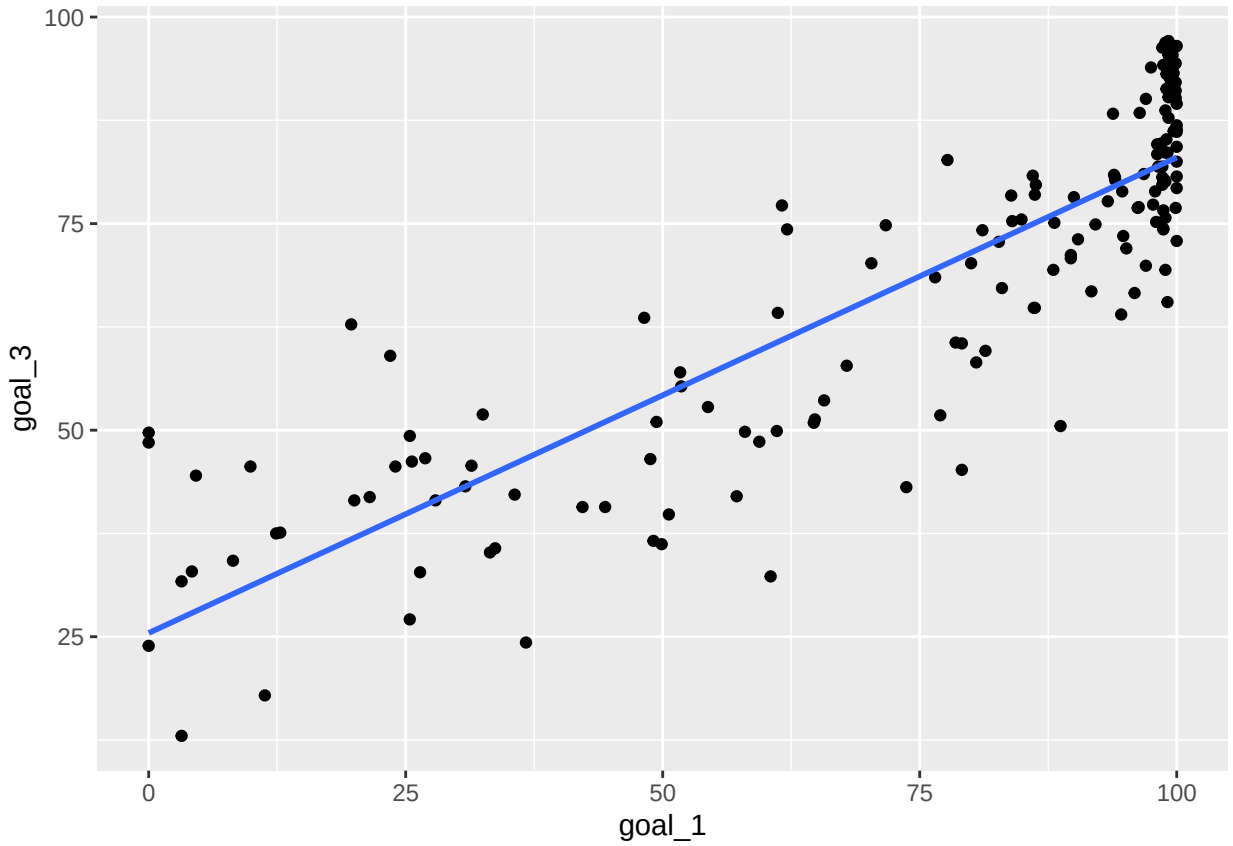
```
sdr_goals <- read_csv("data/sdr_fd5e4b5a.csv") %>% mutate(goal_1 = `Goal 1 Score`, goal_3 = `Goal 3 Sco
```

```
## New names:
## Rows: 206 Columns: 59
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (36): Goal 1 Dash, Goal 1 Trend, Goal 2 Dash, Goal 2 Trend, Goal 3 Dash,... dbl
## (23): ...1, Goal 1 Score, Goal 2 Score, Goal 3 Score, Goal 4 Score, Goal...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

Plotting the result

```
sdr_goals %>% ggplot(aes(x=goal_1, y=goal_3)) + geom_point() + geom_smooth(method="lm", se=FALSE)
```

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



Model Coefficients

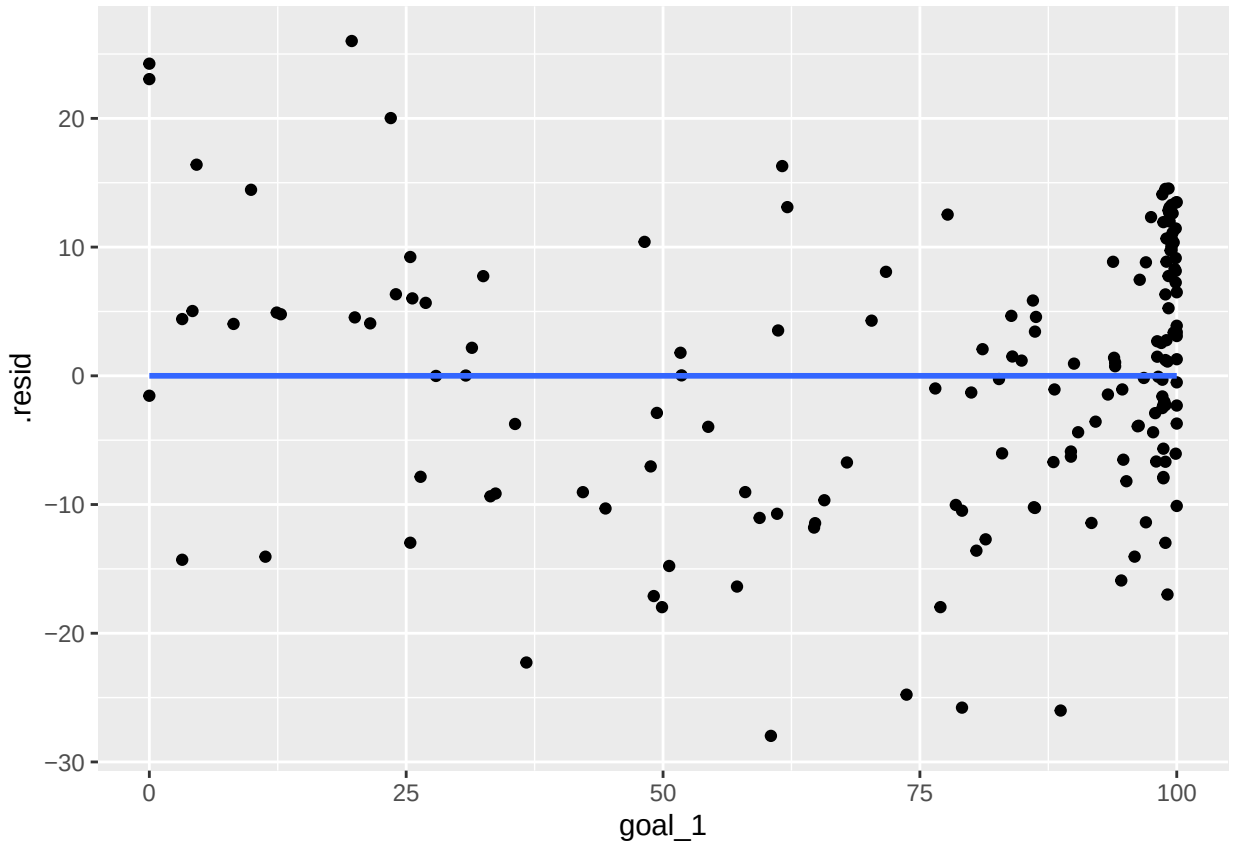
```
model <- lm(goal_3 ~ goal_1, data = sdr_goals)
summary(model)$coefficients
```

```
##               Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 25.4495135 2.12381179 11.98294 4.284726e-24
## goal_1       0.5755994 0.02622236 21.95071 2.082611e-50
```

Plotting residual plot

```
df <- fortify(model)
df %>% ggplot(aes(x = goal_1, y = .resid)) + geom_point() + geom_smooth(method="lm", se=FALSE)
```

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


20
10
0
−10
−20
−30
.resid
0
25
50
75
100
goal_1