

Assessment 1 - COVID-19 impact on digital learning

Do Minh Huu Nguyen_47772360

2024-04-12

```
library(tidyverse)

## — Attaching core tidyverse packages — tidyverse 2.
0.0 —
## ✓ dplyr      1.1.4      ✓ readr      2.1.5
## ✓ forcats   1.0.0      ✓ stringr   1.5.1
## ✓ ggplot2   3.5.0      ✓ tibble    3.2.1
## ✓ lubridate 1.9.3      ✓ tidyr     1.3.1
## ✓ purrr     1.0.2
## — Conflicts — tidyverse_conflict
s() —
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()     masks stats::lag()
## ⓘ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors

library(plyr)

## Warning: package 'plyr' was built under R version 4.3.3

## -----
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, th
en dplyr:
## library(plyr); library(dplyr)
## -----
##
## Attaching package: 'plyr'
##
## The following objects are masked from 'package:dplyr':
##
##   arrange, count, desc, failwith, id, mutate, rename, summarise,
##   summarize
##
## The following object is masked from 'package:purrr':
##
##   compact

library(dplyr)
library(janitor)
```

```
## Warning: package 'janitor' was built under R version 4.3.3

##
## Attaching package: 'janitor'
##
## The following objects are masked from 'package:stats':
##
##   chisq.test, fisher.test

library(stringr)
library(readr)
library(lubridate)
library(ggplot2)

## Rows: 372 Columns: 6
## — Column specification —————
## Delimiter: ","
## chr (5): URL, Product Name, Provider/Company Name, Sector(s), Primary Esse
nt...
## dbl (1): LP ID
##
## 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 m
essage.
## Rows: 233 Columns: 7
## — Column specification —————
## Delimiter: ","
## chr (6): state, locale, pct_black/hispanic, pct_free/reduced, county_conne
ct...
## dbl (1): district_id
##
## 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 m
essage.

spec(products_info)
spec(districts_info)
```

1. Data Cleaning and Wrangling

1.1 Districts_info Cleaning

```
districts_info <- districts_info %>% clean_names()
```

Initially, we need to change all column names into a convenient format, and they should be suitable to be used in the function. Before the replacement of NAs with appropriate data, we need to convert all data formats into a shared and readable standard. After that, for the “pct” column and the “pp” columns, for better understandings, we can change the format from “[x, y]” to “x%-y%”.

District_id

We need to check the unique value in the state to see if there is any missing value to be eliminated, or any inappropriate data form should be fixed. Then, we double check with the `unique()` function to ensure that there is no further cleaning conduct is needed.

```
unique(districts_info$`district_id`)
```

State

We need to check the unique value in the state to see if there is any missing value to be eliminated, or any inappropriate data form should be fixed. Then, we double check with the `unique()` function.

```
unique(districts_info$`state`)  
cleaned1_districts_info = district_info %>%  
  mutate(state = str_trim(state)) %>%  
  mutate(state = replace(state, state == "don't know", NA)) %>%  
  mutate(state = replace(state, state == "whereabouts", NA)) %>%  
  mutate(state = replace(state, state == "Ohio", "Ohio")) %>%  
  mutate(state = replace(state, state == "NY City", "New York")) %>%  
  mutate(state = replace(state, state == "New York", "New York")) %>%  
  mutate(state = str_replace_all(state, regex("uttah", ignore_case = TRUE), "  
Utah")) %>%  
  mutate(state = replace(state, state == "Utaah", "Utah")) %>%  
  mutate(state = replace(state, state == "NaN", NA)) %>%  
  mutate(state = str_to_title(state))  
cleaned1_districts_info <- cleaned1_districts_info[!is.na(cleaned1_districts_  
info$state), ]
```

Locale

```
unique(cleaned1_districts_info$`locale`)  
cleaned2_districts_info = cleaned1_districts_info %>%  
  mutate(locale = str_trim(locale)) %>%  
  mutate(locale = replace(locale, locale == "Sub", "Suburb")) %>%  
  mutate(locale = replace(locale, locale == "Cit", "City")) %>%  
  mutate(locale = replace(locale, locale == "C1ty", "City")) %>%  
  mutate(locale = replace(locale, locale == "NaN", NA)) %>%  
  mutate(locale = str_to_title(locale))  
unique(cleaned2_districts_info$`locale`)
```

pct_black_hispanic

```
unique(cleaned2_districts_info$`pct_black_hispanic`)  
cleaned3_districts_info <- cleaned2_districts_info %>%  
  mutate(pct_black_hispanic = replace(pct_black_hispanic, pct_black_hispanic  
== "NaN", NA))  
unique(cleaned3_districts_info$`pct_black_hispanic`)  
  
cleaned3_districts_info <- cleaned3_districts_info %>%  
  mutate(pct_black_hispanic = case_when(pct_black_hispanic == "[0, 0.2[" ~ "0  
-20%",
```

```
pct_black_hispanic == "[0.2, 0.4[" ~ "20-40%",
pct_black_hispanic == "[0.4, 0.6[" ~ "40-60%",
pct_black_hispanic == "[0.6, 0.8[" ~ "60-80%",
pct_black_hispanic == "[0.8, 1[" ~ "80-100%"))
```

pct_free_reduced

```
unique(cleaned3_districts_info$pct_free_reduced`)

cleaned4_districts_info <- cleaned3_districts_info %>%
  mutate(pct_free_reduced = case_when(pct_free_reduced == "[0, 0.2[" ~ "0-20%",
    pct_free_reduced == "[0.2, 0.4[" ~ "20-40%",
    pct_free_reduced == "[0.4, 0.6[" ~ "40-60%",
    pct_free_reduced == "[0.6, 0.8[" ~ "60-80%",
    pct_free_reduced == "[0.8, 1[" ~ "80-100%"))
```

county_connections_ratio

```
unique(cleaned4_districts_info$county_connections_ratio`)

cleaned5_districts_info <- cleaned4_districts_info %>%
  mutate(county_connections_ratio = replace(county_connections_ratio, county_
connections_ratio == "NaN", NA))

cleaned5_districts_info <- cleaned5_districts_info %>%
  mutate(county_connections_ratio = case_when(county_connections_ratio == "[
0.18, 1[" ~ "0.18-1",
    county_connections_ratio == "[1, 2[" ~ "1-2"))
unique(cleaned5_districts_info$county_connections_ratio`)
```

pp_total_raw

```
unique(cleaned5_districts_info$pp_total_raw`)

cleaned_districts_info <- cleaned5_districts_info %>%
  mutate(pp_total_raw = replace(pp_total_raw, pp_total_raw == "NaN", NA))

cleaned_districts_info <- cleaned_districts_info %>% mutate(pp_total_raw =
case_when(pp_total_raw == "[10000, 12000[" ~ "10000-12000",
  pp_total_raw == "[12000, 14000[" ~ "12000-14000",
  pp_total_raw == "[14000, 16000[" ~ "14000-16000",
  pp_total_raw == "[16000, 18000[" ~ "16000-18000",
  pp_total_raw == "[18000, 20000[" ~ "18000-20000",
  pp_total_raw == "[20000, 22000[" ~ "20000-22000",
  pp_total_raw == "[22000, 24000[" ~ "22000-24000",
  pp_total_raw == "[32000, 34000[" ~ "32000-34000",
  pp_total_raw == "[4000, 6000[" ~ "4000-6000",
  pp_total_raw == "[6000, 8000[" ~ "6000-8000",
  pp_total_raw == "[8000, 10000[" ~ "8000-10000"))
unique(cleaned_districts_info$pp_total_raw`)
```

Replace NA with appropriate value

The idea is to use the state as a reference for the substitution of the missing values.

```
counts_state <- table(cleaned_districts_info$state)
counts_state
```

Based on the shared characteristics, we can replace the NA in 3 following columns including “pct_free_reduced”, county_connections_ratio”, and “pp_total_raw”.

```
pct_free_reduced
table(cleaned_districts_info$state[is.na(cleaned_districts_info$pct_free_reduced)])

counts_pct_free_reduced <- table(cleaned_districts_info$pct_free_reduced)
counts_pct_free_reduced
sum(is.na(cleaned_districts_info$pct_free_reduced))
```

Based on the observations, the missing values are coming from a whole state such as “Massachusetts”, “District of Columbia”, “Tennessee”, and this figure does not seem to have any relation to the others figure. Hence, the use of the median should be applied in this case, as it indicates the mid-point in the value of this category.

```
pct_free_reduced_data <- factor(cleaned_districts_info$pct_free_reduced, ordered = TRUE)
pct_free_reduced_data
median(as.numeric(pct_free_reduced_data), na.rm = TRUE)
```

According to the result, we can recognize that the median for the pct_free_reduced after encoding is “20-40%”. Hence, we should use this value for the NAs.

```
cleaned_districts_info <- cleaned_districts_info %>%
  mutate(pct_free_reduced = replace(pct_free_reduced, which(is.na(pct_free_reduced)), "20-40%"))
```

county_connections_ratio

As there are NA under “county_connections_ratio”, we need to replace this NA with another possible value.

```
cleaned_districts_info <- cleaned_districts_info %>%
  mutate(county_connections_ratio = replace(county_connections_ratio, which(is.na(county_connections_ratio)), "0.18-1"))
```

Among 176 observations, there are 161 “[0.18, 1[” and 1 “[1, 2[”, indicating in average, the county_connections_ratio will be “0.18-1”. Hence, we can assume that the replaced value for missing value of district “1000” is “0.18-1”.

```
cleaned_districts_info <- cleaned_districts_info %>%
  mutate(county_connections_ratio = replace(county_connections_ratio, which(is.na(county_connections_ratio)), "0.18-1"))
```

pp_total_raw

```
table(cleaned_districts_info$state[is.na(cleaned_districts_info$pp_total_raw)])
```

In accordance with the observations, the missing values in the “pp_total_raw” are coming from the same state, which is difficult for us to use the similar districts in the same state as reference. As a consequence, we can take into account of the median of “pp_total_raw” in U.S to fill in the missing values.

```
counts_pp_total_raw <- table(cleaned_districts_info$pp_total_raw)
counts_pp_total_raw
sum(is.na(cleaned_districts_info$pp_total_raw))

pp_total_raw_data <- factor(cleaned_districts_info$pp_total_raw, c("4000-6000", "6000-8000", "8000-10000", "10000-12000", "12000-14000", "14000-16000", "16000-18000", "18000-20000", "20000-22000", "22000-24000", "24000-26000", "26000-28000", "28000-30000", "30000-32000", "32000-34000"), ordered = TRUE)
levels(pp_total_raw_data)
median(as.numeric(pct_free_reduced_data), na.rm = TRUE)
```

According to the result, we can recognize that the median for the pp_total_raw after encoding is “6000-8000”. Hence, we should use this value for the NAs.

```
cleaned_districts_info <- cleaned_districts_info %>%
  mutate(pp_total_raw = replace(pp_total_raw, which(is.na(pp_total_raw)), "6000-8000"))
cleaned_districts_info
```

1.2 Products_info cleaning

We need to check the unique value in the state to see if there is any missing value to be eliminated, or any inappropriate data form should be fixed. Then, we double check with the unique() function to ensure that there is no further cleaning conduct is needed.

LP ID

```
unique(products_info$`LP ID`)
sum(is.na(products_info$`LP ID`))
```

Product Name

```
unique(products_info$`Product Name`)
sum(is.na(products_info$`Product Name`))
```

Provider/Company Name

```
unique(products_info$`Provider/Company Name`)
sum(is.na(products_info$`Provider/Company Name`))
which(is.na(products_info$`Provider/Company Name`))
```

Sector(s)

```
unique(products_info$`Sector(s)`)

products_info <- products_info %>% clean_names()
products_info <- products_info %>%
  mutate(sector_s = na_if(sector_s, "not sure")) %>%
  mutate(sector_s = case_when(
    sector_s == "Corporate" ~ " ; ;Corporate",
```

```

sector_s == "Higher Ed; Corporate" ~ " ;Higher Ed; Corporate",
TRUE ~ sector_s))

products_info <- products_info %>%
  separate(sector_s, c("pre_k_12", "higher_ed", "corporate"), sep=";")

## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 236 rows [1
, 2, 3, 4, 5,
## 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 19, 20, 21, 22, 23, ...].

products_info <- products_info %>%
  mutate(`pre_k_12` = replace(`pre_k_12`, `pre_k_12` == " ", NA)) %>%
  mutate(`higher_ed` = replace(`higher_ed`, `higher_ed` == " ", NA))

products_info <- products_info %>%
  mutate(pre_k_12=ifelse(is.na(pre_k_12), FALSE, TRUE),
         higher_ed=ifelse(is.na(higher_ed), FALSE, TRUE
),
         corporate=ifelse(is.na(corporate), FALSE, TRUE
))
head(products_info)

```

Primary Essential Function

```

cleaned_products_info <- products_info %>%
  separate(primary_essential_function, c("primary", "sub_category_1", "sub_ca
tegory_2"), sep="-")

## Warning: Expected 3 pieces. Missing pieces filled with `NA` in 240 rows [1
, 2, 4, 5, 6,
## 7, 8, 11, 13, 14, 15, 16, 17, 18, 19, 22, 23, 24, 25, 28, ...].

unique(cleaned_products_info$primary)
unique(cleaned_products_info$sub_category_1)

```

Because there is misspelling “CL” instead of “CM”, then we should change it to the correct spelling.

```

cleaned_products_info <- cleaned_products_info %>%
  mutate(primary = replace(primary, primary == " ", NA)) %>%
  mutate(primary = replace(primary, primary == "CL ", "CM ")) %>%
  mutate(primary = str_trim(primary)) %>%
  mutate(sub_category_1 = str_trim(sub_category_1)) %>%
  mutate(sub_category_1 = replace(sub_category_1, sub_category_1 == "Sites, R
esources & Reference", "Sites, Resources & References"))

unique(cleaned_products_info$primary)
unique(cleaned_products_info$sub_category_1)

```

1.3 Engagement data cleaning and handling

Check if whether the following district info. exists in the cleaned_districts_info. data set

```

CheckDistrict_1000 <- "1000" %in% cleaned_districts_info$district_id
CheckDistrict_1000
CheckDistrict_1039 <- "1039" %in% cleaned_districts_info$district_id
CheckDistrict_1039
CheckDistrict_1044 <- "1044" %in% cleaned_districts_info$district_id
CheckDistrict_1044
CheckDistrict_1052 <- "1052" %in% cleaned_districts_info$district_id
CheckDistrict_1052
CheckDistrict_1131 <- "1131" %in% cleaned_districts_info$district_id
CheckDistrict_1131

print(cleaned_districts_info[cleaned_districts_info$district_id %in% c("1000"
,"1039","1044","1052", "1131"), ])

```

Due to the fact that there is no information about district “1039” and “1131”, we should eliminate this from the analysis as this would not be critical and represented for U.S.

```

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

## Rows: 104003 Columns: 4
## — Column specification —————
##
## Delimiter: ","
## chr (1): time
## dbl (3): lp_id, pct_access, engagement_index
##
## 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 m
essage.
## Rows: 38791 Columns: 4
## — Column specification —————
##
## Delimiter: ","
## chr (2): time, lp_id
## dbl (2): pct_access, engagement_index
##
## 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 m
essage.

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

```



```
## Rows: 255722 Columns: 4
## — Column specification —————
## Delimiter: ","
## chr (1): time
## dbl (3): lp_id, pct_access, engagement_index
##
## 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.
## Rows: 91977 Columns: 4
## — Column specification —————
## Delimiter: ","
## chr (1): time
## dbl (3): lp_id, pct_access, engagement_index
##
## 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.
## Rows: 32963 Columns: 4
## — Column specification —————
## Delimiter: ","
## chr (1): time
## dbl (3): lp_id, pct_access, engagement_index
##
## 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.
```

1.3.1 Data cleaning and combining

For each engagement data, we need to ensure all of the value in the “time” column makes sense like year should be 2020, all of the NA in “lp id” should be removed. Furthermore, for the NA in “engagement_index”, this should be converted to 0.

After that, We need to combine these 3 files together including all engagement_data, cleaned_districts_info, and cleaned_products_info. The result can be checked by comparing the number of observations between final_merge and engagement_data

In order to combine the engagement_data and the district_id and distinguish between different districts in the combined file, we need to create a district_id column in each district file.

```
district_1000$district_id <- "1000"
district_1039$district_id <- "1039"
district_1044$district_id <- "1044"
district_1052$district_id <- "1052"
district_1131$district_id <- "1131"
```

District_1000

```
#check the NAs in the "lp_id" and "time" column
sum(is.na(district_1000$`lp_id`))
sum(is.na(district_1000$`time`))

#remove the NAs in the "lp_id" column
district_1000 <- district_1000 %>%
  filter(!is.na(`lp_id`))
sum(is.na(district_1000$`lp_id`))

#Change the format of the value "time" to date in order to clean the value with year is not 2020.
unique(district_1000$`time`)
district_1000$`time` <- as.Date(district_1000$`time`, format="%d/%m/%Y")

district_1000$`time` <- update(district_1000$`time`, year = 2020)

unique(district_1000$`time`)

#The use of right_join in this case will only merge the lp_id appears in the "cleaned_products_info".
merge_1000 <- merge(district_1000, cleaned_districts_info, by="district_id", all.x=TRUE)
final_merge_1000 <- right_join(merge_1000, cleaned_products_info, by="lp_id")
#After merging, there will be some products have not been used and we need to eliminate those from the data set.
final_merge_1000 <- final_merge_1000 %>%
  filter(!is.na(`time`))
final_merge_1000
```

District_1044

```
#check the NAs in the "lp_id" and "time" column
sum(is.na(district_1044$lp_id))
sum(is.na(district_1044$time))

#Change the format of the value "time" to date in order to clean the value with year is not 2020.
unique(district_1044$time)
district_1044$time <- as.Date(district_1044$time, format="%d/%m/%Y")

# Then, update the year to 2020 for all entries
district_1044$time <- update(district_1044$time, year = 2020)
unique(district_1044$time)

#The use of right_join in this case will only merge the lp_id appears in the "cleaned_products_info".
merge_1044 <- merge(district_1044, cleaned_districts_info, by="district_id", all.x=TRUE)
final_merge_1044 <- right_join(merge_1044, cleaned_products_info, by="lp_id")
```

#After merging, there will be some products have not been used and we need to eliminate those from the data set.

```
final_merge_1044 <- final_merge_1044 %>%  
  filter(!is.na(`time`))  
final_merge_1044
```

District_1052

#check the NAs in the "lp_id" and "time" column

```
sum(is.na(district_1052$`lp_id`))  
sum(is.na(district_1052$`time`))
```

#Change the format of the value "time" to date in order to clean the value with year is not 2020.

```
unique(district_1052$time)  
district_1052$time <- as.Date(district_1052$time, format="%d/%m/%Y")
```

Then, update the year to 2020 for all entries

```
district_1052$time <- update(district_1052$time, year = 2020)  
unique(district_1052$time)
```

#The use of right_join in this case will only merge the lp_id appears in the "cleaned_products_info".

```
merge_1052 <- merge(district_1052, cleaned_districts_info, by="district_id",  
all.x=TRUE)  
final_merge_1052 <- right_join(merge_1052, cleaned_products_info, by="lp_id")
```

#After merging, there will be some products have not been used and we need to eliminate those from the data set.

```
final_merge_1052 <- final_merge_1052 %>%  
  filter(!is.na(`time`))  
final_merge_1052
```

Districts combinations and cleaning

The reason why we only combine 3 districts instead of 5 districts because the district 1039 and 1131 don't have any value in the districts_info.

```
combined_districts <- rbind(final_merge_1000, final_merge_1044, final_merge_1052)
```

#Change the NA in the "engagement_index" and "pct_access" to zero

```
combined_districts <- combined_districts %>%  
  mutate(pct_access = replace(pct_access, is.na(pct_access), 0)) %>%  
  mutate(engagement_index = replace(engagement_index, is.na(engagement_index), 0))
```

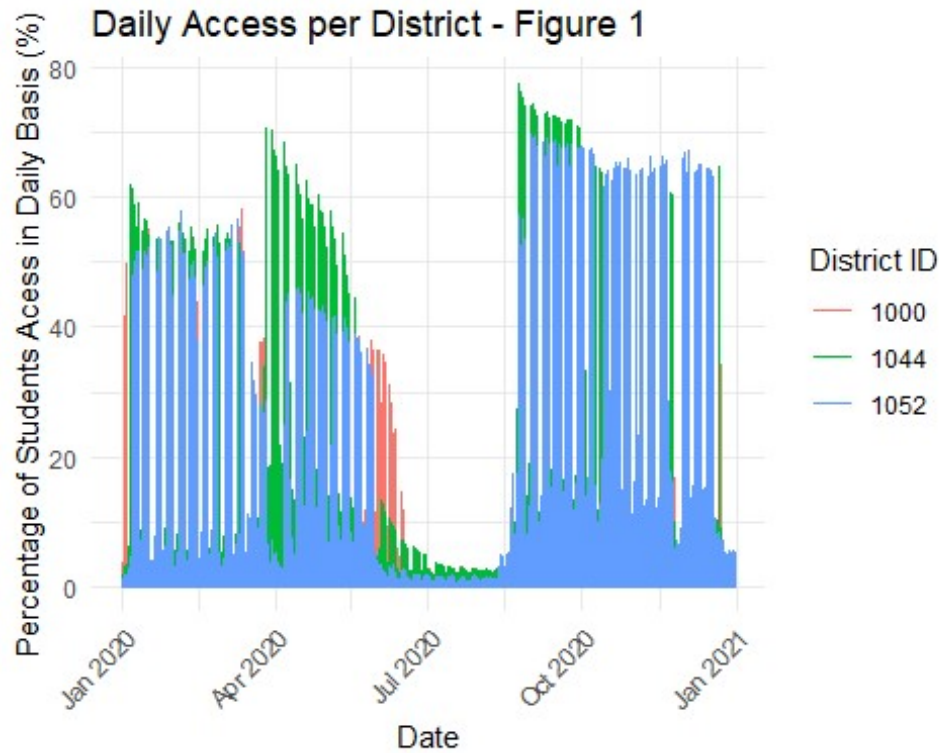
#Double check if any further modification is needed

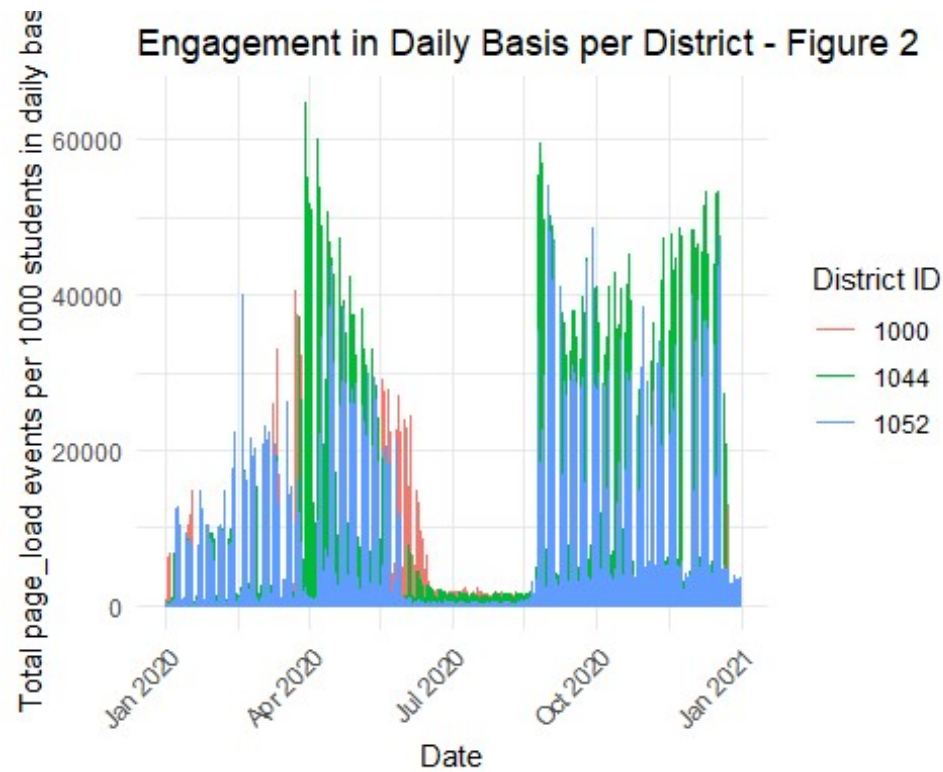
```
sum(is.na(combined_districts$time))  
sum(is.na(combined_districts$lp_id))  
sum(is.na(combined_districts$pct_access))  
sum(is.na(combined_districts$engagement_index))
```

2. Data Visualisation

2.1 Line Chart

I have used the below visualization technique because we want to investigate the changes in the engagement measurement overtime during the period of COVID-19 pandemic classified by district_id.

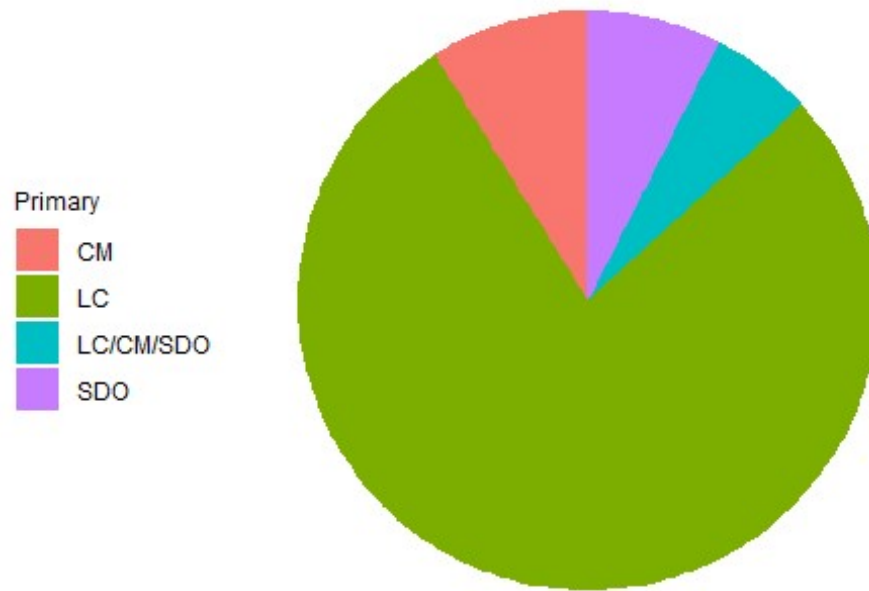




2.2 Pie Chart

I have used the below visualization technique because we want to investigate the distributions in the primary functions and sub-categories.

Distribution of Primary-Function - Figure 3



Sub-Category

Distribution of Sub-Category of Primary Functions - F

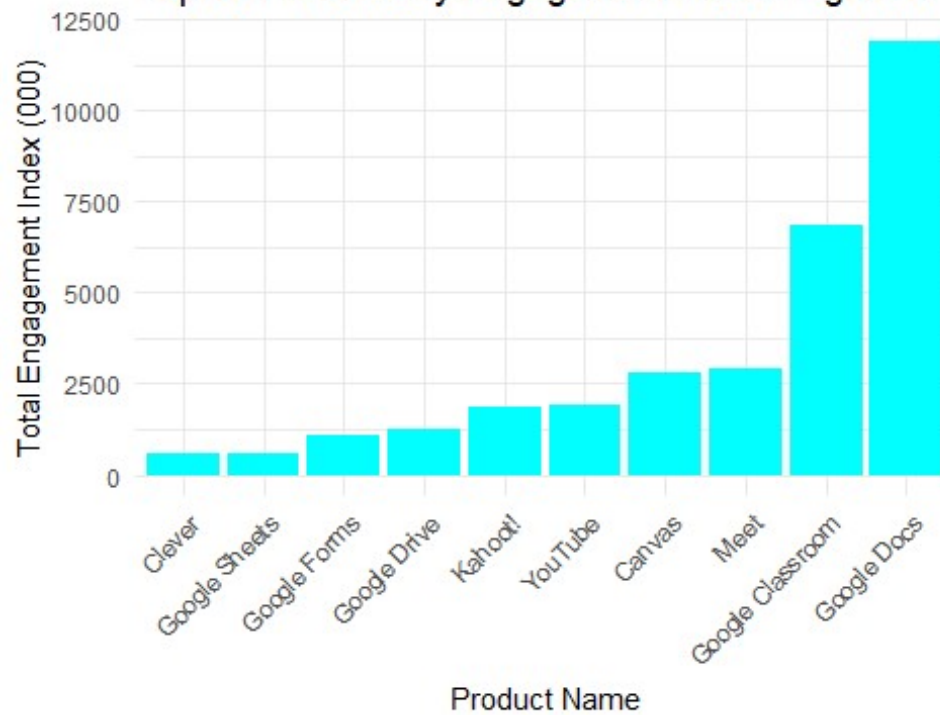


2.3 Bar Chart

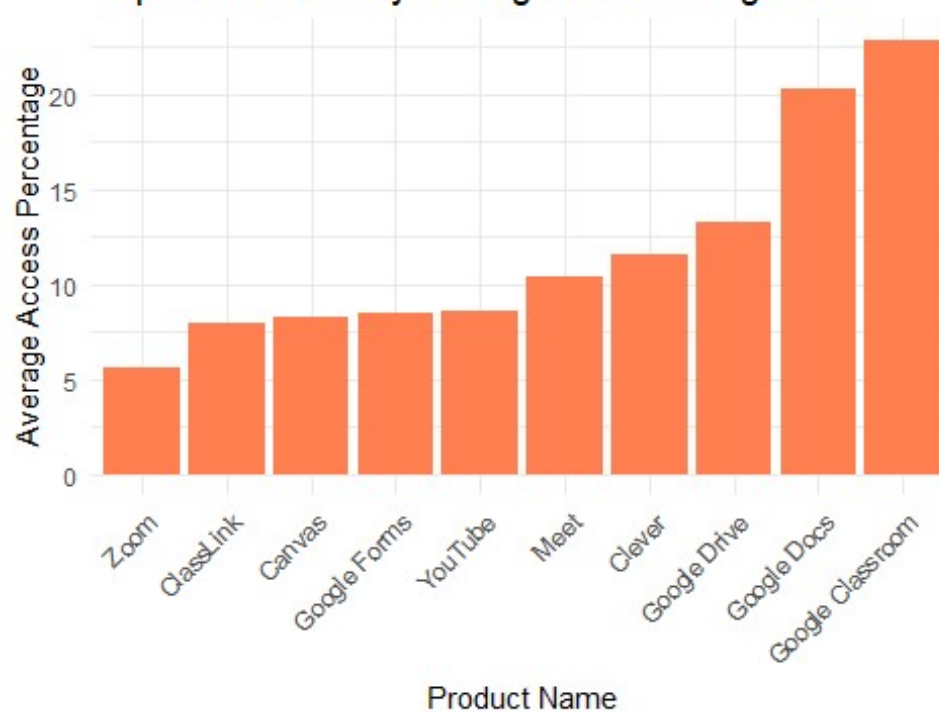
2.3.1 *Top Products measured by the engagement measurement*

I have used the below visualization technique because we want to investigate the top digital products measured by engagement_index and pct_access.

Top 10 Products by Engagement Index - Figure 5



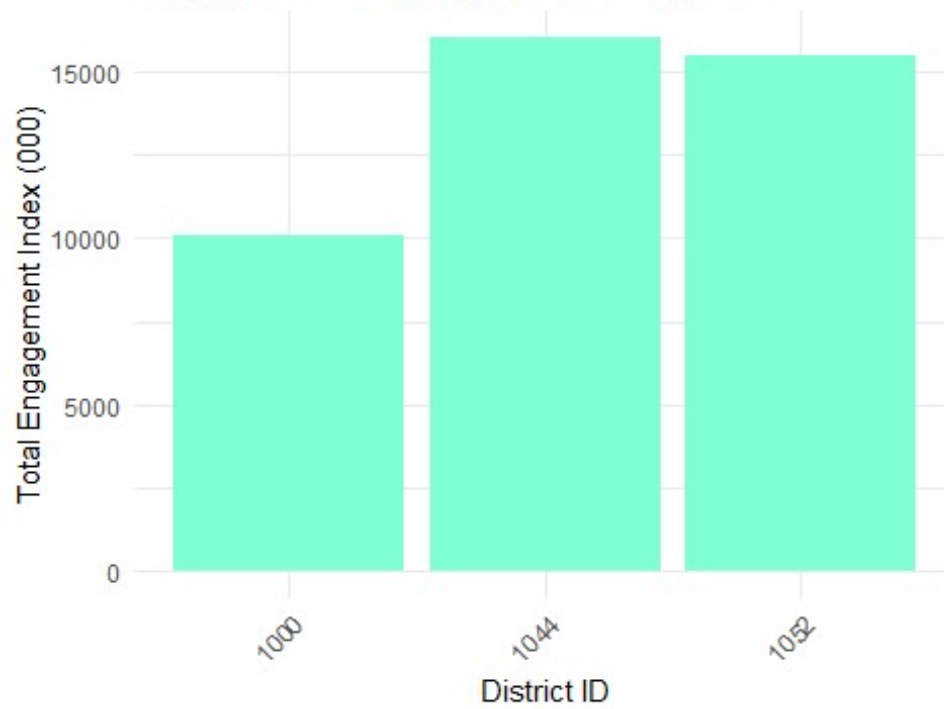
Top 10 Products by Average Access - Figure 6



2.3.2 The engagement measurement classified by District

I have used the below visualization technique because we want to investigate the different in engagement measurement categorized by district_id, in order to further investigate the differences in characteristics.

Engagement Index by District - Figure 7



Average Access by District - Figure 8



3. Findings

The observed trends from figure 1 and 2 in the “engagement index” and “pct_access” measurements from early 2020 show a distinct pattern correlating with the COVID-19 pandemic’s impact on education in the U.S. Initially, there was a noticeable increase in these metrics from the beginning of the year until around April 2020, followed by a decrease from June to August 2020. This decline can be attributed to the typical summer break period. However, engagement levels surged to new heights in the subsequent months and stabilized at a high level through the end of the year.

This pattern reflects the shift from traditional to digital learning methods prompted by the widespread closure of public schools in late March 2020, as the pandemic intensified. Before the pandemic, online learning was often supplementary, focused more on enriching students’ education outside of the core curriculum. The rise in digital engagement in the early part of 2020 highlights how swiftly educational institutions adapted to online platforms (Hollister et al., 2020).

To further analyze the efficacy of these digital tools, it’s important to examine the distribution of digital products based on their primary functions and sub-functions as shown in figure 3 and 4. The majority of these products were geared towards Learning and Curriculum (LC), then Classroom Management (CM) and School & District Operations (SDO). The three sub-categories with the highest usage were “Sites, Resources, and References,” “Digital Learning Platforms,” and “Study Tools,” all of which were well-suited to support isolated, home-based learning environments.

Regarding Figure 5 and 6, the engagement indices show that “Google Docs,” “Google Classroom,” and “Google Meet” were the most utilized platforms, reflecting high engagement, while “Google Drive” was notable for its accessibility, albeit ranking last among the products measured by “pct_access.” These Google products cater to a diverse range of educational needs across various educational sectors.

A comparative analysis of three specific districts, as shown in Figures 7 and 8 — districts “1000,” “1044,” and “1052” — provides additional insights. Although these districts do not represent the entire U.S., they illustrate how demographic and financial factors can influence digital connectivity. District 1000, characterized by the highest percentages of Black and Hispanic students and the highest eligibility for free or reduced meals — coupled with the lowest per-pupil spending — had the lowest digital engagement levels. Conversely, Districts 1044 and 1052, with lower proportions of Black and Hispanic students and higher educational investment, exhibited significantly higher engagement levels, suggesting that students in these districts had better access to remote learning resources during the pandemic. In particular, although district 1000 had high level of supports indicated through pct_free_reduced at about 20% to 40%, this may yield weak beneficial influence on the assessment to digital platforms.

4. References List

Hollister, B., Nair, P., Hill-Lindsay, S., & Chukoskie, L. (2022). Engagement in Online Learning: Student Attitudes and Behavior During COVID-19. *Frontiers in Education (Lausanne)*, 7. <https://doi.org/10.3389/feduc.2022.851019>