Assessment 1 - COVID-19 impact on digital learning

Viet Cuong Truong - 47476184

2023-04-06

## Data Cleaning and Wrangling

Import libraries

library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.0 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ ggplot2 3.4.1 ✔ tibble 3.2.0  
## ✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
## ✔ purrr 1.0.1   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all conflicts to become errors

library(dplyr)  
library(ggplot2)  
library(viridis)

## Loading required package: viridisLite

library(highcharter)

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo   
## Highcharts (www.highcharts.com) is a Highsoft software product which is  
## not free for commercial and Governmental use

library(webshot)  
webshot::install\_phantomjs()

## It seems that the version of `phantomjs` installed is greater than or equal to the requested version.To install the requested version or downgrade to another version, use `force = TRUE`.

### Product files

Import the prodict\_info file into the dataset

#import file  
products <- read\_csv("products\_info.csv",na = c("NaN", "Na"))

## Rows: 372 Columns: 6  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (5): URL, Product Name, Provider/Company Name, Sector(s), Primary Essent...  
## 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 message.

#show summary data of tibble  
str(products)

## spc\_tbl\_ [372 × 6] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ LP ID : num [1:372] 13117 66933 50479 92993 73104 ...  
## $ URL : chr [1:372] "https://www.splashmath.com" "https://abcmouse.com" "https://www.abcya.com" "http://www.aleks.com/" ...  
## $ Product Name : chr [1:372] "SplashLearn" "ABCmouse.com" "ABCya!" "ALEKS" ...  
## $ Provider/Company Name : chr [1:372] "StudyPad Inc." "Age of Learning, Inc" "ABCya.com, LLC" "McGraw-Hill PreK-12" ...  
## $ Sector(s) : chr [1:372] "PreK-122" "PreK-12" "PreK-12" "PreK-12; Higher Ed" ...  
## $ Primary Essential Function: chr [1:372] "LC - Digital Learning Platforms" "LC - Digital Learning Platforms" "LC - Sites, Resources & Reference - Games & Simulations" "LC - Digital Learning Platforms" ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. `LP ID` = col\_double(),  
## .. URL = col\_character(),  
## .. `Product Name` = col\_character(),  
## .. `Provider/Company Name` = col\_character(),  
## .. `Sector(s)` = col\_character(),  
## .. `Primary Essential Function` = col\_character()  
## .. )  
## - attr(\*, "problems")=<externalptr>

#inspect the data by printing the first few rows  
head(products)

## # A tibble: 6 × 6  
## `LP ID` URL `Product Name` Provide…¹ Secto…² Prima…³  
## <dbl> <chr> <chr> <chr> <chr> <chr>   
## 1 13117 https://www.splashmath.com SplashLearn StudyPad… PreK-1… LC - D…  
## 2 66933 https://abcmouse.com ABCmouse.com Age of L… PreK-12 LC - D…  
## 3 50479 https://www.abcya.com ABCya! ABCya.co… PreK-12 LC - S…  
## 4 92993 http://www.aleks.com/ ALEKS McGraw-H… PreK-1… LC - D…  
## 5 73104 https://www.achieve3000.com/ Achieve3000 Achieve3… PreK-12 LC - D…  
## 6 37600 http://www.activelylearn.com/ Actively Learn Actively… PreK-12 LC - D…  
## # … with abbreviated variable names ¹​`Provider/Company Name`, ²​`Sector(s)`,  
## # ³​`Primary Essential Function`

Now we will clean the data. First, let’s examine if the dataset has null values

#check if there are any missing values or NAs  
any(is.na(products))

## [1] FALSE

Let see the Sector column

unique(products$`Sector(s)`) #Sectors column

## [1] "PreK-122" "PreK-12"   
## [3] "PreK-12; Higher Ed" "PreK-12; Higher Ed; Corporate"  
## [5] "PreK-112" ""   
## [7] "Corporate" "Higher Ed; Corporate"   
## [9] "PreK-12; Higher; Corporate" "PPreK-12"

There are lots of errors in the data that need cleaning. We will have to change “PreK-122”, “PreK-112”, “PPreK-12” to “PreK-12” and “PreK-12; Higher; Corporate” to “PreK-12; Higher Ed; Corporate”

products\_clean <- products %>%   
 # change "PreK-122", "PreK-112", "PPreK-12" to "PreK-12"  
 mutate(`Sector(s)` = ifelse(grepl("(PreK-122|PreK-112|PPreK-12)", `Sector(s)`, ignore.case = TRUE), "PreK-12", `Sector(s)`),   
 # change "PreK-12; Higher; Corporate" to "PreK-12; Higher Ed; Corporate"  
 `Sector(s)` = ifelse(grepl("PreK-12; Higher; Corporate", `Sector(s)`, ignore.case = TRUE), "PreK-12; Higher Ed; Corporate", `Sector(s)`))   
unique(products\_clean$`Sector(s)`) # check the unique data again in the Sector column

## [1] "PreK-12" "PreK-12; Higher Ed"   
## [3] "PreK-12; Higher Ed; Corporate" ""   
## [5] "Corporate" "Higher Ed; Corporate"

Our data appears to be ready. Let’s move on to Districts data

### District files

Import the prodict\_info file into the dataset

#import file  
districts <- read\_csv("districts\_info.csv",na = c("NaN", "Na", ""))

## Rows: 233 Columns: 7  
## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (6): state, locale, pct\_black/hispanic, pct\_free/reduced, county\_connect...  
## 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 message.

#show summary data of tibble  
str(districts)

## spc\_tbl\_ [233 × 7] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ district\_id : num [1:233] 8815 2685 4921 3188 2238 ...  
## $ state : chr [1:233] "Illinois" NA "UTAH" NA ...  
## $ locale : chr [1:233] "Suburb" NA "Suburb" NA ...  
## $ pct\_black/hispanic : chr [1:233] "[0, 0.2[" NA "[0, 0.2[" NA ...  
## $ pct\_free/reduced : chr [1:233] "[0, 0.2[" NA "[0.2, 0.4[" NA ...  
## $ county\_connections\_ratio: chr [1:233] "[0.18, 1[" NA "[0.18, 1[" NA ...  
## $ pp\_total\_raw : chr [1:233] "[14000, 16000[" NA "[6000, 8000[" NA ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. district\_id = col\_double(),  
## .. state = col\_character(),  
## .. locale = col\_character(),  
## .. `pct\_black/hispanic` = col\_character(),  
## .. `pct\_free/reduced` = col\_character(),  
## .. county\_connections\_ratio = col\_character(),  
## .. pp\_total\_raw = col\_character()  
## .. )  
## - attr(\*, "problems")=<externalptr>

#inspect the data by printing the first few rows  
head(districts)

## # A tibble: 6 × 7  
## district\_id state locale `pct\_black/hispanic` pct\_free/r…¹ count…² pp\_to…³  
## <dbl> <chr> <chr> <chr> <chr> <chr> <chr>   
## 1 8815 Illinois Suburb [0, 0.2[ [0, 0.2[ [0.18,… [14000…  
## 2 2685 <NA> <NA> <NA> <NA> <NA> <NA>   
## 3 4921 UTAH Suburb [0, 0.2[ [0.2, 0.4[ [0.18,… [6000,…  
## 4 3188 <NA> <NA> <NA> <NA> <NA> <NA>   
## 5 2238 <NA> <NA> <NA> <NA> <NA> <NA>   
## 6 5987 Wisconsin Suburb [0, 0.2[ [0, 0.2[ [0.18,… [10000…  
## # … with abbreviated variable names ¹​`pct\_free/reduced`,  
## # ²​county\_connections\_ratio, ³​pp\_total\_raw

There are lots of NA values in the dataset. Let’s see how many incomplete rows are in the districts dataset.

# Calculate the number of rows with missing values in the "districts" data frame  
sum(!complete.cases(districts))

## [1] 145

Remove all the NA, NAN and empty from the data

# Remove null value   
districts\_clean <- na.omit(districts)

Here we have several variables that we will turn into factors.

\* State, district\_id, and locale are fairly self-explanatory.

\* The pct\_black.hispanic and pct\_free.reduced variables appear to be ordered categorical variables that represent a percentage-group.

We will convert the various variables in this situation into factors.

# check the unique data  
unique(districts\_clean$locale)

## [1] "Suburb" "Cit" "Rural" "City" "Town" "Sub"

Change the typo in the districts data

# Locale  
districts\_clean <- districts\_clean %>% mutate(locale = ifelse(locale == "Cit", "City",   
 ifelse(locale == "Sub", "Suburb", locale)))  
unique(districts\_clean$locale) # check the data again

## [1] "Suburb" "City" "Rural" "Town"

Change the format and remove unwanted characters from PCT\_black/hispanic column

# Pct\_black/hispanic  
districts\_clean$`pct\_black/hispanic` <- str\_trim(districts\_clean$`pct\_black/hispanic`) # remove white space  
districts\_clean$`pct\_black/hispanic` <- ifelse(districts\_clean$`pct\_black/hispanic` == "[0, 0.2[" , "0-20%",   
 (ifelse(districts\_clean$`pct\_black/hispanic` == "[0.2, 0.4[" , "20-40%",  
 (ifelse(districts\_clean$`pct\_black/hispanic` == "[0.4, 0.6[" , "40-60%",  
 (ifelse(districts\_clean$`pct\_black/hispanic` == "[0.6, 0.8[" , "60-80%",  
 (ifelse(districts\_clean$`pct\_black/hispanic` == "[0.8, 1[" , "80-100%", NaN) )))))))) # change the data to percentage

Change the format and remove unwanted characters from Pct\_free/reduced column

# Pct\_free/reduced  
districts\_clean$`pct\_free/reduced` <- str\_trim(districts\_clean$`pct\_free/reduced`) # remove white space  
districts\_clean$`pct\_free/reduced` <- ifelse(districts\_clean$`pct\_free/reduced` == "[0, 0.2[" , "0-20%",   
 (ifelse(districts\_clean$`pct\_free/reduced`== "[0.2, 0.4[" , "20-40%",  
 (ifelse(districts\_clean$`pct\_free/reduced` == "[0.4, 0.6[" , "40-60%",  
 (ifelse(districts\_clean$`pct\_free/reduced` == "[0.6, 0.8[" , "60-80%",  
 (ifelse(districts\_clean$`pct\_free/reduced` == "[0.8, 1[" , "80-100%", NaN) )))))))) # change the data to percentage

Change the format and remove unwanted characters from PP\_Total\_raw column

## PP\_Total\_raw  
districts\_clean$pp\_total\_raw <- gsub( "[", "", districts\_clean$pp\_total\_raw, fixed = TRUE ) # replace "[" to " "  
districts\_clean$pp\_total\_raw <- gsub( ", ", "-", districts\_clean$pp\_total\_raw, fixed = TRUE ) # replace "," to "-"

Change the format and remove unwanted characters from county connection ratio column

## county connection ratio   
districts\_clean$county\_connections\_ratio <- gsub( "[", "", districts\_clean$county\_connections\_ratio, fixed = TRUE ) # replace "[" to " "  
districts\_clean$county\_connections\_ratio <- gsub( ", ", "-", districts\_clean$county\_connections\_ratio, fixed = TRUE )# replace "," to "-"

After cleaning the data, let’s see the districts data again

head(districts\_clean)

## # A tibble: 6 × 7  
## district\_id state locale `pct\_black/hispanic` pct\_f…¹ count…² pp\_to…³  
## <dbl> <chr> <chr> <chr> <chr> <chr> <chr>   
## 1 8815 Illinois Suburb 0-20% 0-20% 0.18-1 14000-…  
## 2 4921 UTAH Suburb 0-20% 20-40% 0.18-1 6000-8…  
## 3 5987 Wisconsin Suburb 0-20% 0-20% 0.18-1 10000-…  
## 4 3710 Utah Suburb 0-20% 40-60% 0.18-1 6000-8…  
## 5 7177 North Carolina Suburb 20-40% 20-40% 0.18-1 8000-1…  
## 6 9812 Utah City 0-20% 20-40% 0.18-1 6000-8…  
## # … with abbreviated variable names ¹​`pct\_free/reduced`,  
## # ²​county\_connections\_ratio, ³​pp\_total\_raw

### Engagement files

# import engagement data  
Engagement\_Data <- list.files("Engagement Data/")  
Engagement\_Data\_2 <- paste0("Engagement Data//", Engagement\_Data)  
  
engage <- lapply(Engagement\_Data\_2, read\_csv) %>%  
 bind\_rows(.id = "source")

## 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 message.  
## Rows: 38790 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: 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.

head(engage)

## # A tibble: 6 × 5  
## source time lp\_id pct\_access engagement\_index  
## <chr> <chr> <dbl> <dbl> <dbl>  
## 1 1 1/01/2020 93690 0 NA   
## 2 1 1/01/2020 17941 0.03 0.9  
## 3 1 1/01/2020 65358 0.03 1.2  
## 4 1 1/01/2020 98265 0.57 37.8  
## 5 1 1/01/2020 59257 0 NA   
## 6 1 1/01/2020 90153 0.06 3.9

We will combine the data with the districts\_file where the ID of an engagement matches the ID of a district. Then, we will merge this data with the products file where the lp\_ID in the engagement file matches the Lp\_id in the products file.

First we have to convert the Lp\_id in the products table to lp\_id

names(products\_clean)[1] <- "lp\_id"

then we merge with 2 tables

disctrict\_joiner <- data.frame(district\_id = districts\_clean$district\_id, source = as.character(1:nrow(districts\_clean)))  
  
engage <- engage %>%  
 left\_join(disctrict\_joiner, by = "source")  
  
engage <- engage %>%  
 left\_join(districts\_clean, by = "district\_id")  
  
engage <- engage %>%  
 left\_join(products\_clean, by = "lp\_id")

Let’s see the first few rows of the engage dataset

#inspect the data by printing the first few rows  
head(engage)

## # A tibble: 6 × 17  
## source time lp\_id pct\_access engag…¹ distr…² state locale pct\_b…³ pct\_f…⁴  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl> <chr> <chr> <chr> <chr>   
## 1 1 1/01/2020 93690 0 NA 8815 Illi… Suburb 0-20% 0-20%   
## 2 1 1/01/2020 17941 0.03 0.9 8815 Illi… Suburb 0-20% 0-20%   
## 3 1 1/01/2020 65358 0.03 1.2 8815 Illi… Suburb 0-20% 0-20%   
## 4 1 1/01/2020 98265 0.57 37.8 8815 Illi… Suburb 0-20% 0-20%   
## 5 1 1/01/2020 59257 0 NA 8815 Illi… Suburb 0-20% 0-20%   
## 6 1 1/01/2020 90153 0.06 3.9 8815 Illi… Suburb 0-20% 0-20%   
## # … with 7 more variables: county\_connections\_ratio <chr>, pp\_total\_raw <chr>,  
## # URL <chr>, `Product Name` <chr>, `Provider/Company Name` <chr>,  
## # `Sector(s)` <chr>, `Primary Essential Function` <chr>, and abbreviated  
## # variable names ¹​engagement\_index, ²​district\_id, ³​`pct\_black/hispanic`,  
## # ⁴​`pct\_free/reduced`

Then we drop all the null values

engage\_clean <- na.omit(engage)  
head(engage\_clean)

## # A tibble: 6 × 17  
## source time lp\_id pct\_access engag…¹ distr…² state locale pct\_b…³ pct\_f…⁴  
## <chr> <chr> <dbl> <dbl> <dbl> <dbl> <chr> <chr> <chr> <chr>   
## 1 1 1/01/2020 17941 0.03 0.9 8815 Illi… Suburb 0-20% 0-20%   
## 2 1 1/01/2020 65358 0.03 1.2 8815 Illi… Suburb 0-20% 0-20%   
## 3 1 1/01/2020 98265 0.57 37.8 8815 Illi… Suburb 0-20% 0-20%   
## 4 1 1/01/2020 90153 0.06 3.9 8815 Illi… Suburb 0-20% 0-20%   
## 5 1 1/01/2022 29322 0.06 5.1 8815 Illi… Suburb 0-20% 0-20%   
## 6 1 1/01/2020 51340 0.09 1.2 8815 Illi… Suburb 0-20% 0-20%   
## # … with 7 more variables: county\_connections\_ratio <chr>, pp\_total\_raw <chr>,  
## # URL <chr>, `Product Name` <chr>, `Provider/Company Name` <chr>,  
## # `Sector(s)` <chr>, `Primary Essential Function` <chr>, and abbreviated  
## # variable names ¹​engagement\_index, ²​district\_id, ³​`pct\_black/hispanic`,  
## # ⁴​`pct\_free/reduced`

# check the unique in the time column  
unique(engage\_clean$time)

## [1] "1/01/2020" "1/01/2022" "2/01/2020" "3/01/2020" "4/01/2020"   
## [6] "5/01/2020" "6/01/2020" "7/01/2020" "8/01/2020" "9/01/2020"   
## [11] "10/01/2020" "11/01/2020" "12/01/2020" "13/01/2020" "14/01/2020"  
## [16] "15/01/2020" "16/01/2020" "17/01/2020" "18/01/2020" "19/01/2020"  
## [21] "20/01/2020" "21/01/2020" "22/01/2020" "23/01/2020" "24/01/2020"  
## [26] "25/01/2020" "26/01/2020" "27/01/2020" "28/01/2020" "29/01/2020"  
## [31] "30/01/2020" "31/01/2020" "1/02/2020" "2/02/2020" "3/02/2020"   
## [36] "4/02/2020" "5/02/2020" "6/02/2020" "7/02/2020" "8/02/2020"   
## [41] "9/02/2020" "10/02/2020" "11/02/2020" "12/02/2020" "13/02/2020"  
## [46] "14/02/2020" "15/02/2020" "16/02/2020" "17/02/2020" "18/02/2020"  
## [51] "19/02/2020" "20/02/2020" "21/02/2020" "22/02/2020" "23/02/2020"  
## [56] "24/02/2020" "25/02/2020" "26/02/2020" "27/02/2020" "28/02/2020"  
## [61] "29/02/2020" "1/03/2020" "2/03/2020" "3/03/2020" "4/03/2020"   
## [66] "5/03/2020" "6/03/2020" "7/03/2020" "8/03/2020" "9/03/2020"   
## [71] "10/03/2020" "11/03/2020" "12/03/2020" "13/03/2020" "14/03/2020"  
## [76] "15/03/2020" "16/03/2020" "17/03/2020" "18/03/2020" "19/03/2020"  
## [81] "20/03/2020" "21/03/2020" "22/03/2020" "23/03/2020" "24/03/2020"  
## [86] "25/03/2020" "26/03/2020" "27/03/2020" "28/03/2020" "29/03/2020"  
## [91] "30/03/2020" "31/03/2020" "1/04/2020" "2/04/2020" "3/04/2020"   
## [96] "4/04/2020" "5/04/2020" "6/04/2020" "7/04/2020" "8/04/2020"   
## [101] "9/04/2020" "10/04/2020" "11/04/2020" "12/04/2020" "13/04/2020"  
## [106] "14/04/2020" "15/04/2020" "16/04/2020" "17/04/2020" "18/04/2020"  
## [111] "19/04/2020" "20/04/2020" "21/04/2020" "22/04/2020" "23/04/2020"  
## [116] "24/04/2020" "25/04/2020" "26/04/2020" "27/04/2020" "28/04/2020"  
## [121] "29/04/2020" "30/04/2020" "1/05/2020" "2/05/2020" "3/05/2020"   
## [126] "4/05/2020" "5/05/2020" "6/05/2020" "7/05/2020" "8/05/2020"   
## [131] "9/05/2020" "10/05/2020" "11/05/2020" "12/05/2020" "13/05/2020"  
## [136] "14/05/2020" "15/05/2020" "16/05/2020" "17/05/2020" "18/05/2020"  
## [141] "19/05/2020" "20/05/2020" "21/05/2020" "22/05/2020" "23/05/2020"  
## [146] "24/05/2020" "25/05/2020" "26/05/2020" "27/05/2020" "28/05/2020"  
## [151] "29/05/2020" "30/05/2020" "31/05/2020" "1/06/2020" "2/06/2020"   
## [156] "3/06/2020" "4/06/2020" "5/06/2020" "6/06/2020" "7/06/2020"   
## [161] "8/06/2020" "9/06/2020" "10/06/2020" "11/06/2020" "12/06/2020"  
## [166] "13/06/2020" "14/06/2020" "15/06/2020" "16/06/2020" "17/06/2020"  
## [171] "18/06/2020" "19/06/2020" "20/06/2020" "21/06/2020" "22/06/2020"  
## [176] "23/06/2020" "24/06/2020" "25/06/2020" "26/06/2020" "27/06/2020"  
## [181] "28/06/2020" "29/06/2020" "30/06/2020" "1/07/2020" "2/07/2020"   
## [186] "3/07/2020" "4/07/2020" "5/07/2020" "6/07/2020" "7/07/2020"   
## [191] "8/07/2020" "9/07/2020" "10/07/2020" "11/07/2020" "12/07/2020"  
## [196] "13/07/2020" "14/07/2020" "15/07/2020" "16/07/2020" "17/07/2020"  
## [201] "18/07/2020" "19/07/2020" "20/07/2020" "21/07/2020" "22/07/2020"  
## [206] "23/07/2020" "24/07/2020" "25/07/2020" "26/07/2020" "27/07/2020"  
## [211] "28/07/2020" "29/07/2020" "30/07/2020" "31/07/2020" "1/08/2020"   
## [216] "2/08/2020" "3/08/2020" "4/08/2020" "5/08/2020" "6/08/2020"   
## [221] "7/08/2020" "8/08/2020" "9/08/2020" "10/08/2020" "11/08/2020"  
## [226] "12/08/2020" "13/08/2020" "14/08/2020" "15/08/2020" "16/08/2020"  
## [231] "17/08/2020" "18/08/2020" "19/08/2020" "20/08/2020" "21/08/2020"  
## [236] "22/08/2020" "23/08/2020" "24/08/2020" "25/08/2020" "26/08/2020"  
## [241] "27/08/2020" "28/08/2020" "29/08/2020" "30/08/2020" "31/08/2020"  
## [246] "1/09/2020" "2/09/2020" "3/09/2020" "4/09/2020" "5/09/2020"   
## [251] "6/09/2020" "7/09/2020" "8/09/2020" "9/09/2020" "10/09/2020"  
## [256] "11/09/2020" "12/09/2020" "13/09/2020" "14/09/2020" "15/09/2020"  
## [261] "16/09/2020" "17/09/2020" "18/09/2020" "19/09/2020" "20/09/2020"  
## [266] "21/09/2020" "22/09/2020" "23/09/2020" "24/09/2020" "25/09/2020"  
## [271] "26/09/2020" "27/09/2020" "28/09/2020" "29/09/2020" "30/09/2020"  
## [276] "1/10/2020" "2/10/2020" "3/10/2020" "4/10/2020" "5/10/2020"   
## [281] "6/10/2020" "7/10/2020" "8/10/2020" "9/10/2020" "10/10/2020"  
## [286] "11/10/2020" "12/10/2020" "13/10/2020" "14/10/2020" "15/10/2020"  
## [291] "16/10/2020" "17/10/2020" "18/10/2020" "19/10/2020" "20/10/2020"  
## [296] "21/10/2020" "22/10/2020" "23/10/2020" "24/10/2020" "25/10/2020"  
## [301] "26/10/2020" "27/10/2020" "28/10/2020" "29/10/2020" "30/10/2020"  
## [306] "31/10/2020" "1/11/2020" "2/11/2020" "3/11/2020" "4/11/2020"   
## [311] "5/11/2020" "6/11/2020" "7/11/2020" "8/11/2020" "9/11/2020"   
## [316] "10/11/2020" "11/11/2020" "12/11/2020" "13/11/2020" "14/11/2020"  
## [321] "15/11/2020" "16/11/2020" "17/11/2020" "18/11/2020" "19/11/2020"  
## [326] "20/11/2020" "21/11/2020" "22/11/2020" "23/11/2020" "24/11/2020"  
## [331] "25/11/2020" "26/11/2020" "27/11/2020" "28/11/2020" "29/11/2020"  
## [336] "30/11/2020" "1/12/2020" "2/12/2020" "3/12/2020" "4/12/2020"   
## [341] "5/12/2020" "6/12/2020" "7/12/2020" "8/12/2020" "9/12/2020"   
## [346] "10/12/2020" "11/12/2020" "12/12/2020" "13/12/2020" "14/12/2020"  
## [351] "15/12/2020" "16/12/2020" "17/12/2020" "18/12/2020" "19/12/2020"  
## [356] "20/12/2020" "21/12/2020" "22/12/2020" "23/12/2020" "24/12/2020"  
## [361] "25/12/2020" "26/12/2020" "27/12/2020" "28/12/2020" "29/12/2020"  
## [366] "30/12/2020" "31/12/2020" "31/12/1020" "1/01/2044" "1/01/2050"   
## [371] "1/01/2033"

There are errors in the data collection process. Due to the range of the dataset (“1/01/2020” to “31/12/2020”), we will remove these dates (“31/12/1020”, “1/01/2044”, “1/01/2050”, “1/01/2033”, and “1/01/2024”).

# Create a logical index of rows  
keep\_rows <- !(engage\_clean$time %in% c("31/12/1020", "1/01/2044", "1/01/2050", "1/01/2033","1/01/2022"))  
  
# Subset the data frame to keep only the rows that meet the condition  
engage\_clean <- subset(engage\_clean, keep\_rows)  
  
# check the unique in the time column again   
unique(engage\_clean$time)

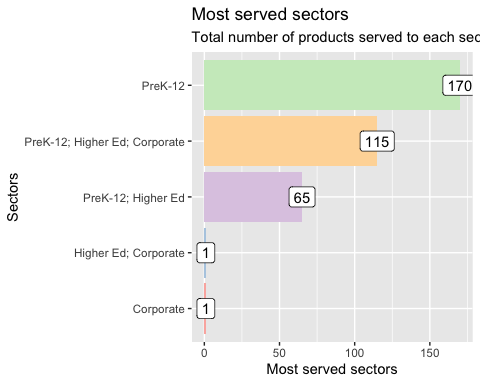
## [1] "1/01/2020" "2/01/2020" "3/01/2020" "4/01/2020" "5/01/2020"   
## [6] "6/01/2020" "7/01/2020" "8/01/2020" "9/01/2020" "10/01/2020"  
## [11] "11/01/2020" "12/01/2020" "13/01/2020" "14/01/2020" "15/01/2020"  
## [16] "16/01/2020" "17/01/2020" "18/01/2020" "19/01/2020" "20/01/2020"  
## [21] "21/01/2020" "22/01/2020" "23/01/2020" "24/01/2020" "25/01/2020"  
## [26] "26/01/2020" "27/01/2020" "28/01/2020" "29/01/2020" "30/01/2020"  
## [31] "31/01/2020" "1/02/2020" "2/02/2020" "3/02/2020" "4/02/2020"   
## [36] "5/02/2020" "6/02/2020" "7/02/2020" "8/02/2020" "9/02/2020"   
## [41] "10/02/2020" "11/02/2020" "12/02/2020" "13/02/2020" "14/02/2020"  
## [46] "15/02/2020" "16/02/2020" "17/02/2020" "18/02/2020" "19/02/2020"  
## [51] "20/02/2020" "21/02/2020" "22/02/2020" "23/02/2020" "24/02/2020"  
## [56] "25/02/2020" "26/02/2020" "27/02/2020" "28/02/2020" "29/02/2020"  
## [61] "1/03/2020" "2/03/2020" "3/03/2020" "4/03/2020" "5/03/2020"   
## [66] "6/03/2020" "7/03/2020" "8/03/2020" "9/03/2020" "10/03/2020"  
## [71] "11/03/2020" "12/03/2020" "13/03/2020" "14/03/2020" "15/03/2020"  
## [76] "16/03/2020" "17/03/2020" "18/03/2020" "19/03/2020" "20/03/2020"  
## [81] "21/03/2020" "22/03/2020" "23/03/2020" "24/03/2020" "25/03/2020"  
## [86] "26/03/2020" "27/03/2020" "28/03/2020" "29/03/2020" "30/03/2020"  
## [91] "31/03/2020" "1/04/2020" "2/04/2020" "3/04/2020" "4/04/2020"   
## [96] "5/04/2020" "6/04/2020" "7/04/2020" "8/04/2020" "9/04/2020"   
## [101] "10/04/2020" "11/04/2020" "12/04/2020" "13/04/2020" "14/04/2020"  
## [106] "15/04/2020" "16/04/2020" "17/04/2020" "18/04/2020" "19/04/2020"  
## [111] "20/04/2020" "21/04/2020" "22/04/2020" "23/04/2020" "24/04/2020"  
## [116] "25/04/2020" "26/04/2020" "27/04/2020" "28/04/2020" "29/04/2020"  
## [121] "30/04/2020" "1/05/2020" "2/05/2020" "3/05/2020" "4/05/2020"   
## [126] "5/05/2020" "6/05/2020" "7/05/2020" "8/05/2020" "9/05/2020"   
## [131] "10/05/2020" "11/05/2020" "12/05/2020" "13/05/2020" "14/05/2020"  
## [136] "15/05/2020" "16/05/2020" "17/05/2020" "18/05/2020" "19/05/2020"  
## [141] "20/05/2020" "21/05/2020" "22/05/2020" "23/05/2020" "24/05/2020"  
## [146] "25/05/2020" "26/05/2020" "27/05/2020" "28/05/2020" "29/05/2020"  
## [151] "30/05/2020" "31/05/2020" "1/06/2020" "2/06/2020" "3/06/2020"   
## [156] "4/06/2020" "5/06/2020" "6/06/2020" "7/06/2020" "8/06/2020"   
## [161] "9/06/2020" "10/06/2020" "11/06/2020" "12/06/2020" "13/06/2020"  
## [166] "14/06/2020" "15/06/2020" "16/06/2020" "17/06/2020" "18/06/2020"  
## [171] "19/06/2020" "20/06/2020" "21/06/2020" "22/06/2020" "23/06/2020"  
## [176] "24/06/2020" "25/06/2020" "26/06/2020" "27/06/2020" "28/06/2020"  
## [181] "29/06/2020" "30/06/2020" "1/07/2020" "2/07/2020" "3/07/2020"   
## [186] "4/07/2020" "5/07/2020" "6/07/2020" "7/07/2020" "8/07/2020"   
## [191] "9/07/2020" "10/07/2020" "11/07/2020" "12/07/2020" "13/07/2020"  
## [196] "14/07/2020" "15/07/2020" "16/07/2020" "17/07/2020" "18/07/2020"  
## [201] "19/07/2020" "20/07/2020" "21/07/2020" "22/07/2020" "23/07/2020"  
## [206] "24/07/2020" "25/07/2020" "26/07/2020" "27/07/2020" "28/07/2020"  
## [211] "29/07/2020" "30/07/2020" "31/07/2020" "1/08/2020" "2/08/2020"   
## [216] "3/08/2020" "4/08/2020" "5/08/2020" "6/08/2020" "7/08/2020"   
## [221] "8/08/2020" "9/08/2020" "10/08/2020" "11/08/2020" "12/08/2020"  
## [226] "13/08/2020" "14/08/2020" "15/08/2020" "16/08/2020" "17/08/2020"  
## [231] "18/08/2020" "19/08/2020" "20/08/2020" "21/08/2020" "22/08/2020"  
## [236] "23/08/2020" "24/08/2020" "25/08/2020" "26/08/2020" "27/08/2020"  
## [241] "28/08/2020" "29/08/2020" "30/08/2020" "31/08/2020" "1/09/2020"   
## [246] "2/09/2020" "3/09/2020" "4/09/2020" "5/09/2020" "6/09/2020"   
## [251] "7/09/2020" "8/09/2020" "9/09/2020" "10/09/2020" "11/09/2020"  
## [256] "12/09/2020" "13/09/2020" "14/09/2020" "15/09/2020" "16/09/2020"  
## [261] "17/09/2020" "18/09/2020" "19/09/2020" "20/09/2020" "21/09/2020"  
## [266] "22/09/2020" "23/09/2020" "24/09/2020" "25/09/2020" "26/09/2020"  
## [271] "27/09/2020" "28/09/2020" "29/09/2020" "30/09/2020" "1/10/2020"   
## [276] "2/10/2020" "3/10/2020" "4/10/2020" "5/10/2020" "6/10/2020"   
## [281] "7/10/2020" "8/10/2020" "9/10/2020" "10/10/2020" "11/10/2020"  
## [286] "12/10/2020" "13/10/2020" "14/10/2020" "15/10/2020" "16/10/2020"  
## [291] "17/10/2020" "18/10/2020" "19/10/2020" "20/10/2020" "21/10/2020"  
## [296] "22/10/2020" "23/10/2020" "24/10/2020" "25/10/2020" "26/10/2020"  
## [301] "27/10/2020" "28/10/2020" "29/10/2020" "30/10/2020" "31/10/2020"  
## [306] "1/11/2020" "2/11/2020" "3/11/2020" "4/11/2020" "5/11/2020"   
## [311] "6/11/2020" "7/11/2020" "8/11/2020" "9/11/2020" "10/11/2020"  
## [316] "11/11/2020" "12/11/2020" "13/11/2020" "14/11/2020" "15/11/2020"  
## [321] "16/11/2020" "17/11/2020" "18/11/2020" "19/11/2020" "20/11/2020"  
## [326] "21/11/2020" "22/11/2020" "23/11/2020" "24/11/2020" "25/11/2020"  
## [331] "26/11/2020" "27/11/2020" "28/11/2020" "29/11/2020" "30/11/2020"  
## [336] "1/12/2020" "2/12/2020" "3/12/2020" "4/12/2020" "5/12/2020"   
## [341] "6/12/2020" "7/12/2020" "8/12/2020" "9/12/2020" "10/12/2020"  
## [346] "11/12/2020" "12/12/2020" "13/12/2020" "14/12/2020" "15/12/2020"  
## [351] "16/12/2020" "17/12/2020" "18/12/2020" "19/12/2020" "20/12/2020"  
## [356] "21/12/2020" "22/12/2020" "23/12/2020" "24/12/2020" "25/12/2020"  
## [361] "26/12/2020" "27/12/2020" "28/12/2020" "29/12/2020" "30/12/2020"  
## [366] "31/12/2020"

## Data Visualization

### Product files

First let’s see what is the most served sector in the Products

The bar plot is a suitable choice for visualizing the most served sectors because it can effectively display the total number of products served to each sector. With the bar plot, it is easy to compare the number of products served in each sector by simply comparing the height of the bars.



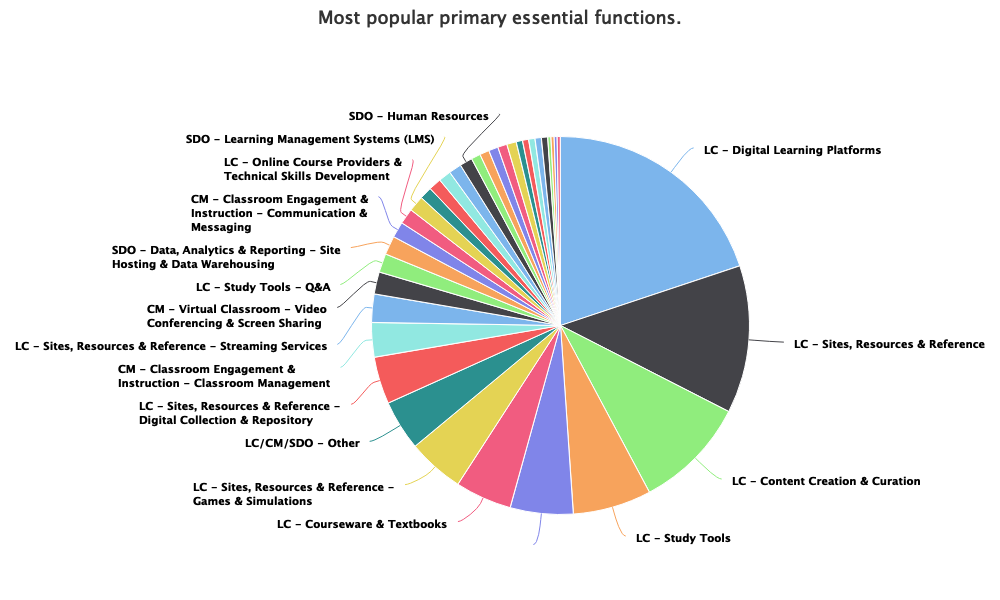
We can see that “PreK–12” appears 170 times in the data file, which shows that a sizable portion of digital learning materials are targeted towards K–12 students. This could be because most kids in the US have to go to school from K–12, and there is a high demand for digital learning resources to meet that need.

Similarly, “PreK-12; Higher Ed; Corporate” also appears 140 times, indicating that a significant number of digital learning products have been developed for multiple sectors, including K-12, higher education, and corporate training. This could be because digital learning products can be customised to meet the needs of different audiences, and many providers offer products that can be used across multiple sectors.

As the phrase “PreK–12; Higher Ed” only appears 65 times in the data file, the market for digital learning products aimed at K–12 and higher education is not very big. Because K–12 and higher education are two different markets with different needs and requirements, it may be harder to make products that work well in both.

Next we will see about the Primary Essential Functions

The pie chart is a useful visualization to show the proportion of each category in a dataset. In this case, we are showing the proportion of each primary essential function in the products\_clean dataset. The pie chart is particularly useful in this case because we are interested in understanding the relative popularity of each primary essential function. By using a pie chart, we can easily see which functions are more popular and which are less popular, and we can also see the proportion of each function relative to the others.

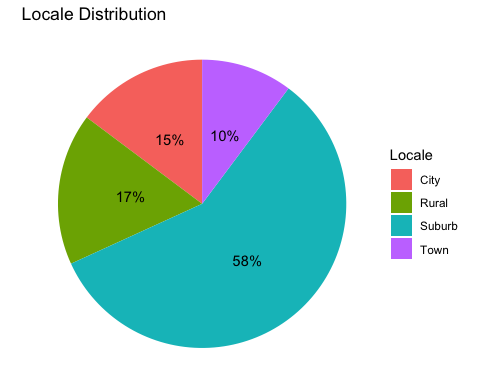


* The most popular product function is “LC - Digital Learning Platforms”, accounting for 28.5% of the total products.
* The second most popular function is “LC - Sites, Resources & Reference”, accounting for 18% of the total products.
* “LC - Content Creation & Curation” and “LC - Study Tools” are also popular, accounting for 13.8% and 9.5% of the total products, respectively.
* Other functions have a relatively smaller market share, with most accounting for less than 5% of the total products.

Overall, the data suggests that digital learning platforms and resources are the most in-demand product functions, with a significant proportion of products falling under these categories.

### District files

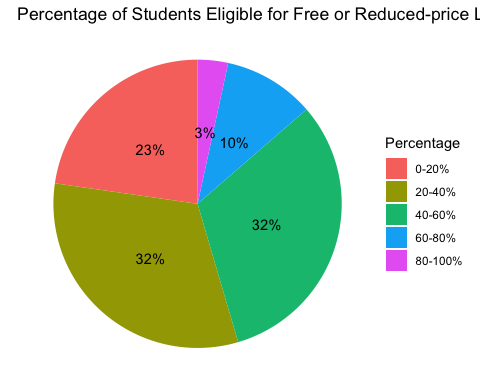
**Locale distribution**



The chart shows that the majority of the dataset is located in suburban areas, with 58% falling into this category. This suggests that suburban areas are disproportionately represented in the dataset compared to other types of locations.

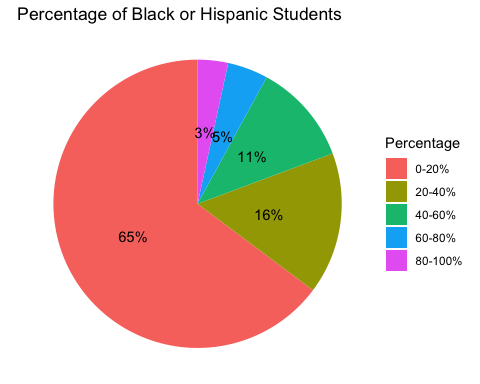
Moreover, the graph shows that only 17% of schools are located in rural areas, 15% in the city, and 10% in town. This could potentially be significant if there are important differences in digital learning outcomes or experiences between rural, city, town, and suburban areas.

**Percentage of Students Eligible for Free or Reduced-price Lunch**



The graph depicts the percentage of students who qualify for free or reduced-cost lunches across various categories. This plot could potentially be useful for identifying schools or districts with higher or lower levels of need. The majority of students qualify for the reduced-price lunch range of 20 to 60%, while only a small percentage qualify for the highest and lowest ranges. This could suggest that the majority of students come from low- to middle-income families.

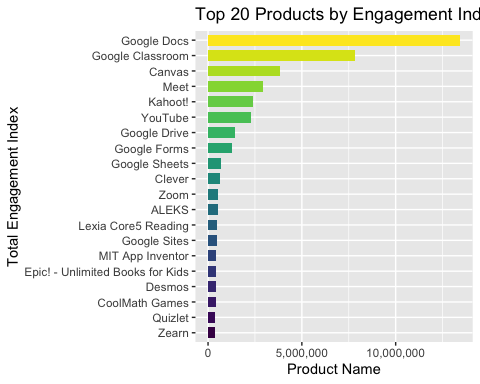
**Percentage of Black or Hispanic Students**



The third graph illustrates the percentage of Black and Hispanic students in each category. The majority of Black and Hispanic students fall within the 0-20% range, indicating a higher level of need compared to other groups.

### Engagement files

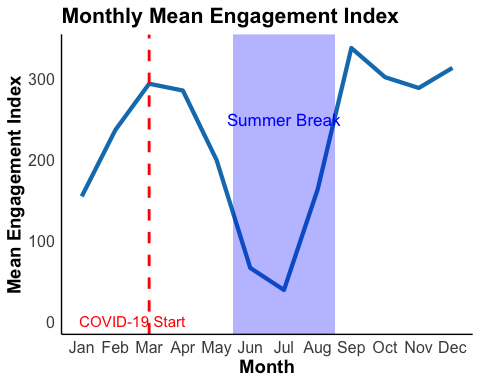
For the engagement files, we aim to determine the most popular products that students have been using throughout the year 2020. To achieve this, we will need to analyze the data and identify the products that have been accessed the most by students during this time period.



According to the engagement index in the chart, these are the top 20 digital learning tools that students will use the most in 2020. The engagement index is a metric that looks at how often and for how long students use a product to figure out how interested they are in it.

Most of the items on the list are made by Google. Google Docs and Google Classroom are at the top, followed by Canvas and Meet. Kahoot!, YouTube, and Zoom are some other popular products. Some learning tools, like ALEKS and Lexia Core5 Reading, and game-based learning platforms, like CoolMath Games, are also on the list.

In the LearnPlatform COVID-19 Impact on Digital Learning we will analyze the engagement\_index of students using various educational products throughout the year 2020. We will examine the changes in the engagement\_index over the course of the month and identify any patterns or trends that may emerge.



Based on the data given, it looks like the average level of engagement on digital learning platforms changes over the course of the year. The level of engagement is low in January and February, but it steadily goes up in March and reaches its peak in April.

In May and June, however, the level of engagement drops sharply, and in July it hits its lowest point. From there, it starts to get better again slowly in August and September, and then it peaks again in October. In November and December, the level of engagement stays pretty high, but at the end of the year, it goes down a little.

## Insights

* Google products dominate the list: Out of the top 10 most popular products used by students, 7 are Google products. This shows that Google products are popular and widely used in the education field.
* People use video conferencing tools a lot. Tools like Meet, Zoom, and Zearn are in the top 20 most popular products, which shows that online video conferencing is being used more and more for remote learning.
* Personalized learning platforms are also popular: ALEKS, Lexia Core5 Reading, and Zearn are some of the most popular products that students use to learn in a way that fits their needs.
* People want free resources for learning: Epic! - Unlimited Books for Kids is a popular product that lets kids read digital books for free from a huge library. This shows that people want free educational materials, especially during the COVID-19 pandemic.
* The engagement index varies a lot. It is a way to measure how much students are using a certain product. Based on the data, we can see that the engagement index for different products is very different, with some having a much higher engagement index than others. This could be because of many things, like the quality of the product, how easy it is to use, and how interested students are.
* After the pandemic started, there is a sharp drop in the amount of engagement. It begins to rise again during the middle of summer. This indicates that the majority of students did not attend spring semester classes following the outbreak of the pandemic.
* Digital learning has changed a lot because of the COVID-19 pandemic. With schools closed and students learning from home, there has been a surge in the use of digital learning platforms.
* Digital learning platforms are used in different ways at different times of the year and during different seasons. Engagement is usually at its highest when students are in school and at its lowest when school is out for the summer.
* There may be differences in how different kinds of online learning platforms are used (e.g. those focused on specific subjects, those used by different age groups).

## Conclution

In conclusion, some intriguing new information about the US digital learning landscape has emerged from the data analysis of the LearnPlatform dataset. The dataset includes information on the products, districts, and student engagement, allowing us to gain a comprehensive understanding of the trends and patterns in digital learning.

From analyzing the product files, we’ve learned that digital learning products are most popular in the K-12 market and that digital learning platforms and resources are the most-requested product functions. In comparison to other types of locations, suburban areas are disproportionately represented in the dataset, and the majority of students come from low- to middle-income families, according to the analysis of the district files.

From the district files dataset suburbs are overrepresented in the dataset when compared to other kinds of locations. Only 17% of schools are in rural areas, 15% are in cities, and 10% are in towns; the majority are in suburban areas. Additionally, a sizable portion of students qualify for free or reduced-price lunch, indicating that socioeconomic status should be taken into account when analysing the outcomes and experiences of digital learning.

The top 20 digital learning tools that students used the most in 2020, with Google products being the most popular, were identified through analysis of the engagement files. Additionally, we’ve noticed that the degree of engagement on digital learning platforms varies throughout the year, with a notable rise in March and April and a decline in May and June.

Overall, The LearnPlatform dataset provides important insight into the American digital learning landscape. However, with only five engagement files available for analysis, it has limited ability to inform policy development and decision-making towards improving digital learning outcomes for all students.