607_project2

Samane Khademi

2022-09-30

Link to Github

Link to RPubs

Load libraies:

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(stringr)
library(readr)
```

Data set 1:

Data can be found here

```
df1 <- read.csv("https://raw.githubusercontent.com/Samane86/607_project2/main/agcensus-chapter1-table1-U
df1 <- data.frame(df1)
```

Let's take a look at our data:

```
glimpse(df1)
```

```
## Rows: 129
## Columns: 12
## $ state           <chr> "US TOTAL", "US TOTAL", "US TOTAL", "US TOTAL", "US TO~
## $ state.fips      <int> 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99~
```

```
## $ county          <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ county.code     <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ commodity       <chr> "FARM OPERATIONS", "FARM OPERATIONS", "FARM OPERATIONS~
## $ data.item       <chr> "FARM OPERATIONS - NUMBER OF OPERATIONS", "FARM OPERAT~
## $ domain.category <chr> "", "AREA OPERATED: (1.0 TO 9.9 ACRES)", "AREA OPERATE~
## $ X2017           <chr> "2,042,220", "273,325", "583,001", "564,763", "315,017~
## $ X2012           <chr> "2,109,303", "223,634", "589,549", "634,047", "346,038~
## $ X2007           <chr> "2,204,792", "232,849", "620,283", "660,530", "368,368~
## $ X2002           <chr> "2,128,982", "179,346", "563,772", "658,705", "388,617~
## $ X1997           <chr> "2,215,876", "205,390", "530,902", "694,489", "428,215~
```

```
head(df1)
```

```
##      state state.fips county county.code       commodity
## 1 US TOTAL         99     NA          NA FARM OPERATIONS
## 2 US TOTAL         99     NA          NA FARM OPERATIONS
## 3 US TOTAL         99     NA          NA FARM OPERATIONS
## 4 US TOTAL         99     NA          NA FARM OPERATIONS
## 5 US TOTAL         99     NA          NA FARM OPERATIONS
## 6 US TOTAL         99     NA          NA FARM OPERATIONS
##                                data.item                     domain.category
## 1 FARM OPERATIONS - NUMBER OF OPERATIONS
## 2 FARM OPERATIONS - NUMBER OF OPERATIONS   AREA OPERATED: (1.0 TO 9.9 ACRES)
## 3 FARM OPERATIONS - NUMBER OF OPERATIONS AREA OPERATED: (10.0 TO 49.9 ACRES)
## 4 FARM OPERATIONS - NUMBER OF OPERATIONS    AREA OPERATED: (50 TO 179 ACRES)
## 5 FARM OPERATIONS - NUMBER OF OPERATIONS   AREA OPERATED: (180 TO 499 ACRES)
## 6 FARM OPERATIONS - NUMBER OF OPERATIONS   AREA OPERATED: (500 TO 999 ACRES)
##       X2017     X2012     X2007     X2002     X1997
## 1 2,042,220 2,109,303 2,204,792 2,128,982 2,215,876
## 2   273,325   223,634   232,849   179,346   205,390
## 3   583,001   589,549   620,283   563,772   530,902
## 4   564,763   634,047   660,530   658,705   694,489
## 5   315,017   346,038   368,368   388,617   428,215
## 6   133,321   142,555   149,713   161,552   179,447
```

I want to see how many categories column state has:

```
df1 %>%
  count(state)
```

```
##      state   n
## 1 US TOTAL 129
```

It seems that there is just one value repeated all along the column.

```
df1 %>%
  count(commodity)
```

```
##            commodity  n
## 1            AG LAND  8
## 2      ANIMAL TOTALS  2
## 3             BARLEY  3
```

```
## 4              BEANS  3
## 5             CATTLE  8
## 6    CHEMICAL TOTALS  1
## 7           CHICKENS  4
## 8   COMMODITY TOTALS 11
## 9               CORN  6
## 10            COTTON  3
## 11       CROP TOTALS  1
## 12    EXPENSE TOTALS  1
## 13   FARM OPERATIONS 14
## 14              FEED  1
## 15 FERTILIZER TOTALS  1
## 16              FUELS  1
## 17     HAY & HAYLAGE  3
## 18              HOGS  4
## 19          INTEREST  1
## 20              LABOR  1
## 21  MACHINERY TOTALS  2
## 22              OATS  3
## 23          ORCHARDS  2
## 24           PEANUTS  3
## 25          POTATOES  2
## 26              RICE  3
## 27           SORGHUM  6
## 28          SOYBEANS  3
## 29        SUGARBEETS  3
## 30         SUGARCANE  3
## 31         SUNFLOWER  3
## 32    SWEET POTATOES  2
## 33           TOBACCO  3
## 34  VEGETABLE TOTALS  2
## 35             WHEAT 12
```

This data set has a lot of information, I am going to pare it down to some of the produce items in order to practice the pivot commands more effectively. If we wanted to work with the data set, it would be easy enough to repeat the similar steps for the rest of the data in this file.

```
produce <- df1 %>%
  filter(commodity %in% c("OATS", "BARLEY",
                          "SOYBEANS", "BEANS", "COTTON", "TOBACCO",
                          "HAY & HAYLAGE", "RICE", "SUNFLOWER",
                          "SUGARBEETS", "SUGARCANE", "PEANUTS",
                          "POTATOES", "SWEET POTATOES"))
```

Let's pick up some columns, because some of columns contain just category, So they do not add any knowledge to our analysis.

```
produce <- produce %>%
  select(c("commodity", "data.item", "X2017", "X2012",
           "X2007", "X2002", "X1997"))
```

```
head(produce)
```

```
##   commodity                                data.item       X2017       X2012
## 1      OATS   OATS - OPERATIONS WITH AREA HARVESTED      19,842      35,038
## 2      OATS                OATS - ACRES HARVESTED     814,140   1,078,698
## 3      OATS       OATS - PRODUCTION, MEASURED IN BU  50,406,624  65,646,178
## 4    BARLEY BARLEY - OPERATIONS WITH AREA HARVESTED      11,188      18,667
## 5    BARLEY               BARLEY - ACRES HARVESTED   2,206,808   3,283,905
## 6    BARLEY     BARLEY - PRODUCTION, MEASURED IN BU 161,624,924 215,059,358
##         X2007       X2002       X1997
## 1      42,558      63,763      94,811
## 2   1,509,149   1,996,916   2,739,810
## 3  89,508,669 109,840,449 154,654,269
## 4      19,848      24,747      43,269
## 5   3,521,957   4,015,654   6,108,682
## 6 207,089,232 214,800,035 346,413,080
```

Missing values:

Do we have NULL values?

```
colSums(is.na(produce))
```

```
## commodity data.item     X2017     X2012     X2007     X2002     X1997
##         0         0         0         0         0         0         0
```

So we don't have missing values.

I want to rename some of the columns name to make more sense:

```
produce <- produce %>%
  rename("2017" = "X2017" , "2012"= "X2012" , "2007" = "X2007", "2002" = "X2002",
         "1997" = "X1997")
```

```
produce %>%
  count(data.item)
```

```
##                                                                  data.item
## 1                                                BARLEY - ACRES HARVESTED
## 2                                     BARLEY - OPERATIONS WITH AREA HARVESTED
## 3                                        BARLEY - PRODUCTION, MEASURED IN BU
## 4                    BEANS, DRY EDIBLE, (EXCL CHICKPEAS & LIMA) - ACRES HARVESTED
## 5  BEANS, DRY EDIBLE, (EXCL CHICKPEAS & LIMA) - OPERATIONS WITH AREA HARVESTED
## 6     BEANS, DRY EDIBLE, (EXCL CHICKPEAS & LIMA) - PRODUCTION, MEASURED IN CWT
## 7                                                COTTON - ACRES HARVESTED
## 8                                     COTTON - OPERATIONS WITH AREA HARVESTED
## 9                                      COTTON - PRODUCTION, MEASURED IN BALES
## 10                                            HAY & HAYLAGE - ACRES HARVESTED
## 11                                HAY & HAYLAGE - OPERATIONS WITH AREA HARVESTED
## 12                       HAY & HAYLAGE - PRODUCTION, MEASURED IN TONS, DRY BASIS
## 13                                               OATS - ACRES HARVESTED
```

```
## 14                                    OATS - OPERATIONS WITH AREA HARVESTED
## 15                                        OATS - PRODUCTION, MEASURED IN BU
## 16                                                PEANUTS - ACRES HARVESTED
## 17                                   PEANUTS - OPERATIONS WITH AREA HARVESTED
## 18                                       PEANUTS - PRODUCTION, MEASURED IN LB
## 19                                                POTATOES - ACRES HARVESTED
## 20                                  POTATOES - OPERATIONS WITH AREA HARVESTED
## 21                                                    RICE - ACRES HARVESTED
## 22                                      RICE - OPERATIONS WITH AREA HARVESTED
## 23                                         RICE - PRODUCTION, MEASURED IN CWT
## 24                                                SOYBEANS - ACRES HARVESTED
## 25                                  SOYBEANS - OPERATIONS WITH AREA HARVESTED
## 26                                      SOYBEANS - PRODUCTION, MEASURED IN BU
## 27                                               SUGARBEETS - ACRES HARVESTED
## 28                                  SUGARBEETS - OPERATIONS WITH AREA HARVESTED
## 29                                    SUGARBEETS - PRODUCTION, MEASURED IN TONS
## 30                                          SUGARCANE, SUGAR - ACRES HARVESTED
## 31                                SUGARCANE, SUGAR - OPERATIONS WITH AREA HARVESTED
## 32                                  SUGARCANE, SUGAR - PRODUCTION, MEASURED IN TONS
## 33                                               SUNFLOWER - ACRES HARVESTED
## 34                                 SUNFLOWER - OPERATIONS WITH AREA HARVESTED
## 35                                    SUNFLOWER - PRODUCTION, MEASURED IN LB
## 36                                         SWEET POTATOES - ACRES HARVESTED
## 37                                SWEET POTATOES - OPERATIONS WITH AREA HARVESTED
## 38                                                TOBACCO - ACRES HARVESTED
## 39                                  TOBACCO - OPERATIONS WITH AREA HARVESTED
## 40                                      TOBACCO - PRODUCTION, MEASURED IN LB
##    n
## 1  1
## 2  1
## 3  1
## 4  1
## 5  1
## 6  1
## 7  1
## 8  1
## 9  1
## 10 1
## 11 1
## 12 1
## 13 1
## 14 1
## 15 1
## 16 1
## 17 1
## 18 1
## 19 1
## 20 1
## 21 1
## 22 1
## 23 1
## 24 1
## 25 1
## 26 1
```

```
## 27 1
## 28 1
## 29 1
## 30 1
## 31 1
## 32 1
## 33 1
## 34 1
## 35 1
## 36 1
## 37 1
## 38 1
## 39 1
## 40 1
```

I am interested to make the table wide and have data.item as columns, but this columns have lots of categories, In fact all rows in this column are distinct values, I want to unite them based on three categories: "OPERATIONS WITH AREA HARVESTED" , "ACRES HARVESTED", "PRODUCTION"

```
produce <- mutate_if(
  tibble::as_tibble(produce),
  is.character,
  stringr ::str_replace_all, pattern = ".*OPERATIONS.*",
  replacement = "OPERATIONS_WITH_AREA_HARVESTED")

produce <- mutate_if(
  tibble::as_tibble(produce),
  is.character,
  stringr ::str_replace_all, pattern = ".*ACRES.*",
  replacement = "ACRES_HARVESTED")


produce <- mutate_if(
  tibble::as_tibble(produce),
  is.character,
  stringr ::str_replace_all, pattern = ".*PRODUCTION.*",
  replacement =  "PRODUCTION")
```

Now I want to make all years columns into just one column and name it year:

```
produce <- produce  %>%
  pivot_longer(!c(commodity,data.item), names_to = "year", values_to = "value" )
```

```
produce <- produce %>%
  pivot_wider(names_from = data.item, values_from = "value")
```

```
produce
```

```
## # A tibble: 70 x 5
##    commodity year  OPERATIONS_WITH_AREA_HARVESTED ACRES_HARVESTED PRODUCTION
##    <chr>     <chr> <chr>                         <chr>           <chr>
##  1 OATS      2017  19,842                        814,140         50,406,624
```

```
##  2 OATS      2012  35,038                          1,078,698       65,646,178
##  3 OATS      2007  42,558                          1,509,149       89,508,669
##  4 OATS      2002  63,763                          1,996,916       109,840,449
##  5 OATS      1997  94,811                          2,739,810       154,654,269
##  6 BARLEY    2017  11,188                          2,206,808       161,624,924
##  7 BARLEY    2012  18,667                          3,283,905       215,059,358
##  8 BARLEY    2007  19,848                          3,521,957       207,089,232
##  9 BARLEY    2002  24,747                          4,015,654       214,800,035
## 10 BARLEY    1997  43,269                          6,108,682       346,413,080
## # ... with 60 more rows
```

I tried to convert the numbers in columns 3,4 and 5 to numeric, But I countered a problem, All of the data in these 3 columns converted to NA, Here i noticed that whereas I run the function colSums(is.na(produce)) and the results indicated that there is no NA values, But there was blank cells(Implicit missing values). So I have to mutate the blank cells to NA and then deal with missing values:

```
produce <- produce %>%
  mutate(across(c("OPERATIONS_WITH_AREA_HARVESTED","ACRES_HARVESTED","PRODUCTION"), ~ifelse(.=="", NA, a
```

Also in one column we had a cell with a value of : "(D)" So I tried to mutate it to NA:

```
produce <- produce %>%
  mutate(across(c("OPERATIONS_WITH_AREA_HARVESTED","ACRES_HARVESTED","PRODUCTION"), na_if, "(D)"))
```

Here I tried to convert the last three columns values to numeric, They were comma separated numbers so I used parse_number function from readr package:

```
produce <- produce%>%
  mutate_at(3:5, readr::parse_number)
```

Now we have Na values and we have to decide how we want to treat them. Let's take a look at the distribution of data point for column OPERATIONS_WITH_AREA_HARVESTED:

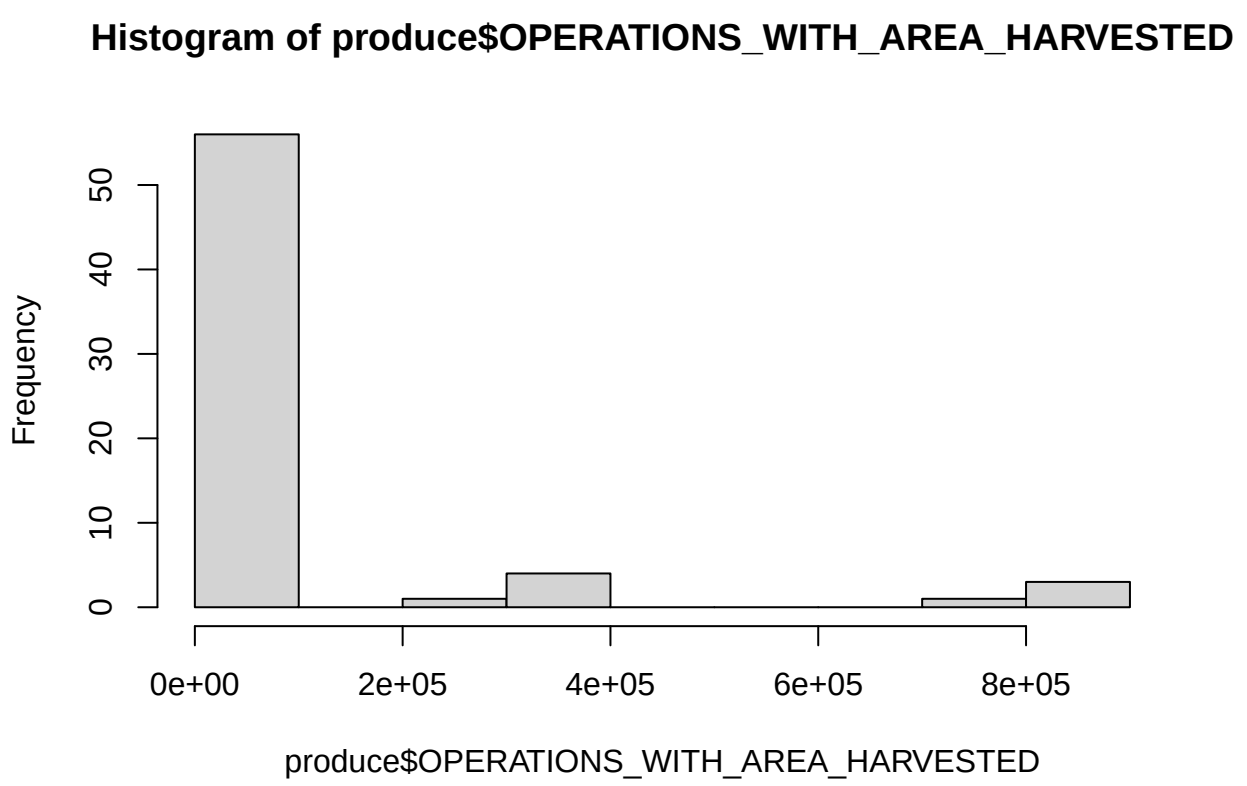
```
hist(produce$OPERATIONS_WITH_AREA_HARVESTED, na.rm=TRUE)
```

```
## Warning in plot.window(xlim, ylim, "", ...): "na.rm" is not a graphical
## parameter

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...): "na.rm"
## is not a graphical parameter

## Warning in axis(1, ...): "na.rm" is not a graphical parameter

## Warning in axis(2, at = yt, ...): "na.rm" is not a graphical parameter
```



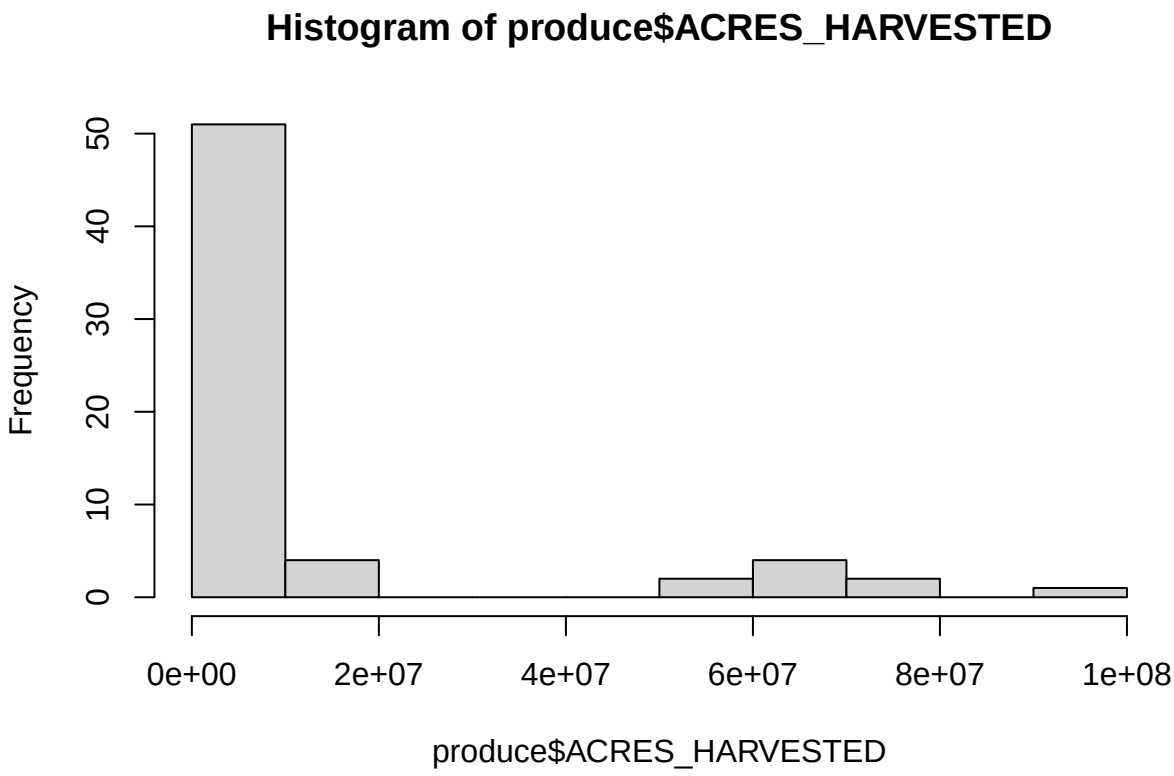

```
hist(produce$ACRES_HARVESTED, na.rm=TRUE)
```

```
## Warning in plot.window(xlim, ylim, "", ...): "na.rm" is not a graphical
## parameter

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...): "na.rm"
## is not a graphical parameter

## Warning in axis(1, ...): "na.rm" is not a graphical parameter

## Warning in axis(2, at = yt, ...): "na.rm" is not a graphical parameter
```



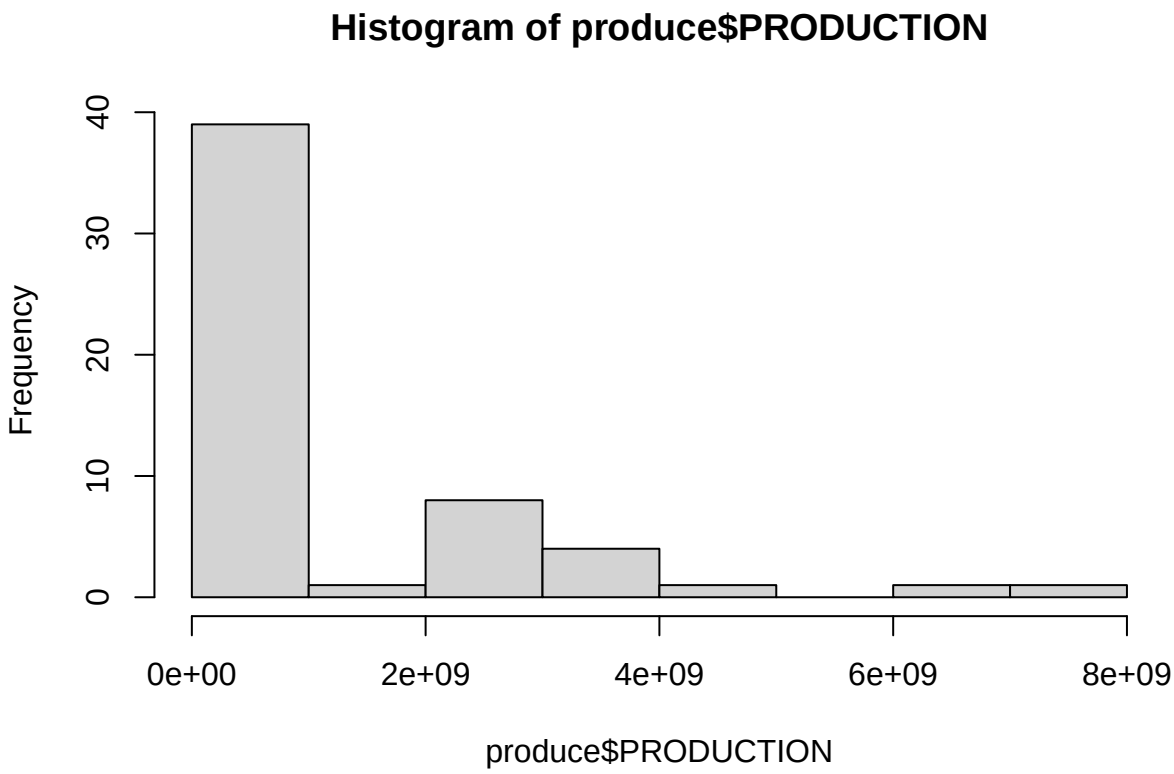

```
hist(produce$PRODUCTION, na.rm=TRUE)
```

```
## Warning in plot.window(xlim, ylim, "", ...): "na.rm" is not a graphical
## parameter

## Warning in title(main = main, sub = sub, xlab = xlab, ylab = ylab, ...): "na.rm"
## is not a graphical parameter

## Warning in axis(1, ...): "na.rm" is not a graphical parameter

## Warning in axis(2, at = yt, ...): "na.rm" is not a graphical parameter
```



As we can see these three distributions are highly right skewed and mean of the values in this columns are going to be very affected by this outlines. So imputing null values with the mean of the column is not going to be a good idea. By eyeballing these columns, I think imputing NA values with the nearest neighbor value using fill function is a good idea:

```
produce <- produce %>%
  fill(3:5, .direction = "up")
```

```
produce <- produce %>%
  fill(PRODUCTION, .direction = "down")
```

```
head(produce)
```

```
## # A tibble: 6 x 5
##   commodity year  OPERATIONS_WITH_AREA_HARVESTED ACRES_HARVESTED PRODUCTION
##   <chr>     <chr>                          <dbl>           <dbl>      <dbl>
## 1 OATS      2017                           19842          814140   50406624
## 2 OATS      2012                           35038         1078698   65646178
## 3 OATS      2007                           42558         1509149   89508669
## 4 OATS      2002                           63763         1996916  109840449
## 5 OATS      1997                           94811         2739810  154654269
## 6 BARLEY    2017                           11188         2206808  161624924
```

We now have the produce data arranged in a “tidy” format, where each row corresponds to a specific measurement and each column corresponds to a single variable.

DATASET 2

Data can be found here

Loading data:

```
df2 <- read.csv("https://raw.githubusercontent.com/Samane86/607_project2/main/StudentsPerformance.csv")
df2 <- data.frame(df2)
```

How our data looks like:

```
glimpse(df1)
```

```
## Rows: 129
## Columns: 12
## $ state           <chr> "US TOTAL", "US TOTAL", "US TOTAL", "US TOTAL", "US TO~
## $ state.fips      <int> 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99, 99~
## $ county          <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ county.code     <lgl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
## $ commodity       <chr> "FARM OPERATIONS", "FARM OPERATIONS", "FARM OPERATIONS~
## $ data.item       <chr> "FARM OPERATIONS - NUMBER OF OPERATIONS", "FARM OPERAT~
## $ domain.category <chr> "", "AREA OPERATED: (1.0 TO 9.9 ACRES)", "AREA OPERATE~
## $ X2017           <chr> "2,042,220", "273,325", "583,001", "564,763", "315,017~
## $ X2012           <chr> "2,109,303", "223,634", "589,549", "634,047", "346,038~
## $ X2007           <chr> "2,204,792", "232,849", "620,283", "660,530", "368,368~
## $ X2002           <chr> "2,128,982", "179,346", "563,772", "658,705", "388,617~
## $ X1997           <chr> "2,215,876", "205,390", "530,902", "694,489", "428,215~
```

```
head(df2)
```

```
##   gender race.ethnicity parental.level.of.education        lunch
## 1 female        group B           bachelor's degree     standard
## 2 female        group C                some college     standard
## 3 female        group B             master's degree     standard
## 4   male        group A          associate's degree free/reduced
## 5   male        group C                some college     standard
## 6 female        group B          associate's degree     standard
##   test.preparation.course math.score reading.score writing.score
## 1                    none         72            72            74
## 2               completed         69            90            88
## 3                    none         90            95            93
## 4                    none         47            57            44
## 5                    none         76            78            75
## 6                    none         71            83            78
```

I prefer to change the columns name:

```
df2 <- df2 %>%
  rename(race = race.ethnicity, parent.LOE = parental.level.of.education, test_prep = test.preparation.
         math = math.score, reading = reading.score, writing = writing.score)
```

Missing values:

Do we have NA values?

```
colSums(is.na(df2))
```

```
##     gender       race parent.LOE      lunch  test_prep       math    reading
##          0          0          0          0          0          0          0
##    writing
##          0
```

How many categories does parental level of education have?

```
table(df2$parent.LOE)
```

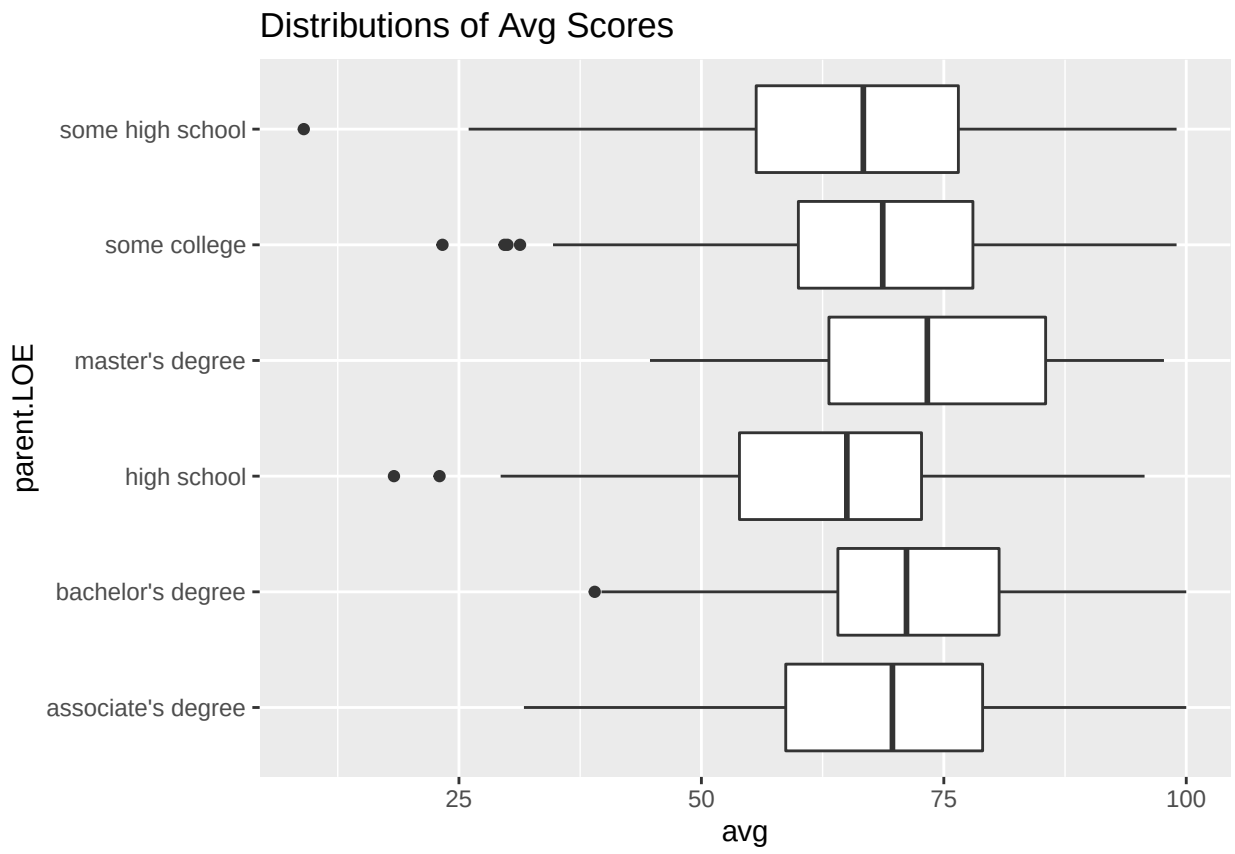
```
##
## associate's degree  bachelor's degree        high school    master's degree
##                222                118                196                 59
##       some college   some high school
##                226                179
```

I want to have all three subject's score as an Average and store it in a new column and arrange them in Descendingorder of average:

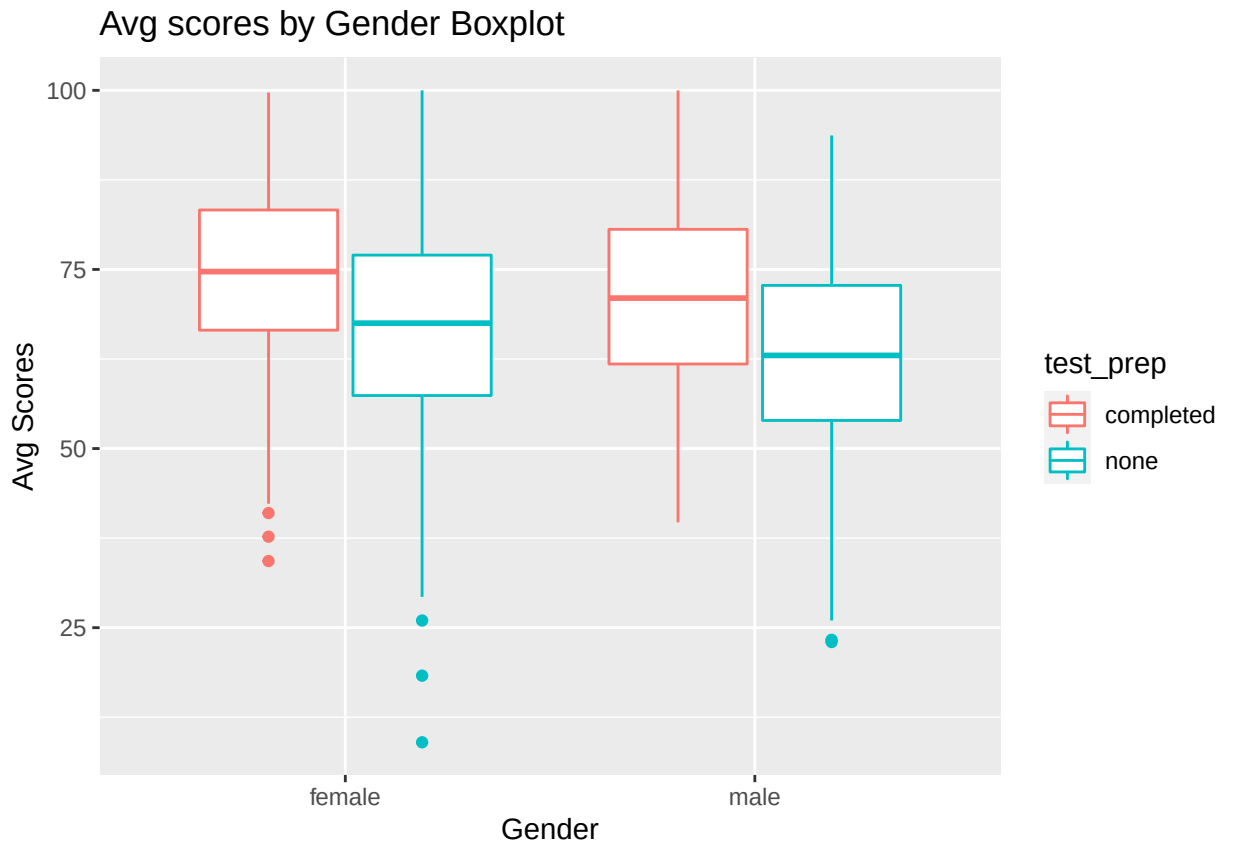
```
df2 <- df2 %>%
  mutate(avg = round((math+reading+writing)/3 , 1)) %>%
  select(-c(math, reading, writing)) %>%
  arrange(desc(avg))
```

plot

```
ggplot(df2, aes(x = parent.LOE, y = avg)) +
  geom_boxplot() +
  coord_flip() +
  ggtitle("Distributions of Avg Scores")
```




```
ggplot(df2, aes(gender, avg, color = test_prep)) +
  geom_boxplot() +
  ggtitle("Avg scores by Gender Boxplot") +
  xlab("Gender") + ylab("Avg Scores")
```



Dataset3:

In 2013, students of the Statistics class at FSEV UK were asked to rate how much they like each one of 39 paintings (on a scale from 1 to 5). These comprise of 13 different art movements (exactly 3 paintings for each art movement).

S1-S48: students' ratings, where one means "don't like at all" (integer) art movement: the art movement the painting belongs to (categorical) artist: the author of the painting (categorical) painting: the name of the painting (categorical)

Data can be found here

Loading data:

```
paintings <- as_tibble (read.csv("https://raw.githubusercontent.com/Samane86/607_project2/main/paintings
```

Taking a look at our data:

```
str(paintings)
```

```
## tibble [39 x 51] (S3: tbl_df/tbl/data.frame)
```

```
##  $ S1          : int [1:39] 3 2 1 5 2 5 4 2 1 3 ...
##  $ S2          : int [1:39] 2 2 3 3 2 4 5 1 2 5 ...
##  $ S3          : int [1:39] 3 2 3 1 2 4 4 1 2 5 ...
##  $ S4          : int [1:39] 1 4 1 1 1 1 3 2 1 5 ...
##  $ S5          : int [1:39] 1 1 1 1 1 1 2 2 1 2 ...
##  $ S6          : int [1:39] 1 1 1 1 1 1 2 3 1 2 ...
##  $ S7          : int [1:39] 2 2 5 3 1 3 5 1 3 5 ...
##  $ S8          : int [1:39] 2 3 2 3 2 5 5 3 3 4 ...
##  $ S9          : int [1:39] 3 4 1 1 1 2 3 2 1 4 ...
##  $ S10         : int [1:39] 1 3 2 1 1 1 2 1 1 4 ...
##  $ S11         : int [1:39] 2 3 2 1 1 2 4 2 1 3 ...
##  $ S12         : int [1:39] 2 2 1 2 1 2 4 2 1 4 ...
##  $ S13         : int [1:39] 2 4 1 2 1 2 3 4 1 5 ...
##  $ S14         : int [1:39] 2 2 1 2 2 3 5 1 2 4 ...
##  $ S15         : int [1:39] 2 5 1 1 2 3 4 2 1 5 ...
##  $ S16         : int [1:39] 4 3 2 3 2 3 2 3 1 5 ...
##  $ S17         : int [1:39] 5 3 5 5 5 5 4 3 5 4 ...
##  $ S18         : int [1:39] 5 5 4 3 5 5 3 3 5 5 ...
##  $ S19         : int [1:39] 4 2 1 5 2 5 5 1 1 5 ...
##  $ S20         : int [1:39] 5 4 4 5 3 4 5 2 2 5 ...
##  $ S21         : int [1:39] 4 3 1 1 1 3 5 1 1 4 ...
##  $ S22         : int [1:39] 3 3 2 2 2 4 4 1 1 3 ...
##  $ S23         : int [1:39] 1 3 3 4 3 4 2 1 2 2 ...
##  $ S24         : int [1:39] 1 1 1 4 2 5 4 2 1 5 ...
##  $ S25         : int [1:39] 5 2 4 1 4 1 4 1 1 1 ...
##  $ S26         : int [1:39] 4 2 4 1 2 4 3 4 1 4 ...
##  $ S27         : int [1:39] 4 3 5 2 2 5 2 4 2 3 ...
##  $ S28         : int [1:39] 2 4 2 1 1 2 3 2 2 2 ...
##  $ S29         : int [1:39] 2 1 1 1 1 2 2 1 1 1 ...
##  $ S30         : int [1:39] 1 3 1 2 1 2 2 2 1 3 ...
##  $ S31         : int [1:39] 1 1 1 1 1 1 3 1 1 1 ...
##  $ S32         : int [1:39] 2 3 1 2 1 4 2 1 2 3 ...
##  $ S33         : int [1:39] 2 5 2 1 2 3 2 1 1 5 ...
##  $ S34         : int [1:39] 5 5 3 4 2 5 5 4 1 5 ...
##  $ S35         : int [1:39] 1 4 2 2 3 3 4 4 1 4 ...
##  $ S36         : int [1:39] 1 4 2 2 3 3 4 4 1 4 ...
##  $ S37         : int [1:39] 3 2 1 2 1 3 4 2 1 4 ...
##  $ S38         : int [1:39] 3 2 1 4 1 1 1 1 1 4 ...
##  $ S39         : int [1:39] 2 3 1 3 1 1 4 3 1 2 ...
##  $ S40         : int [1:39] 2 4 1 3 1 3 4 2 1 3 ...
##  $ S41         : int [1:39] 3 4 2 2 2 1 4 4 2 5 ...
##  $ S42         : int [1:39] 2 4 1 2 1 2 3 3 1 5 ...
##  $ S43         : int [1:39] 1 4 1 3 1 2 4 3 1 1 ...
##  $ S44         : int [1:39] 2 1 1 1 2 2 1 1 2 3 ...
##  $ S45         : int [1:39] 2 4 4 1 1 3 2 3 3 4 ...
##  $ S46         : int [1:39] 4 2 3 3 4 4 5 2 3 5 ...
##  $ S47         : int [1:39] 2 2 1 2 2 1 3 4 1 1 ...
##  $ S48         : int [1:39] 2 1 3 5 4 5 1 1 5 5 ...
##  $ art.movement: chr [1:39] "Renaissance" "Renaissance" "Renaissance" "Baroque" ...
##  $ artist      : chr [1:39] "Sandro Botticelli" "Leonardo da Vinci" "Raphael" "Caravaggio" ...
##  $ painting    : chr [1:39] "The Birth of Venus" "Lady with an Ermine" "Three Graces" "Entombment" .
```

```
head(paintings)
```

```
## # A tibble: 6 x 51
##      S1    S2    S3    S4    S5    S6    S7    S8    S9   S10   S11   S12   S13
##   <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
## 1     3     2     3     1     1     1     2     2     3     1     2     2     2
## 2     2     2     2     4     1     1     2     3     4     3     3     2     4
## 3     1     3     3     1     1     1     5     2     1     2     2     1     1
## 4     5     3     1     1     1     1     3     3     1     1     1     2     2
## 5     2     2     2     1     1     1     1     2     1     1     1     1     1
## 6     5     4     4     1     1     1     3     5     2     1     2     2     2
## # ... with 38 more variables: S14 <int>, S15 <int>, S16 <int>, S17 <int>,
## #   S18 <int>, S19 <int>, S20 <int>, S21 <int>, S22 <int>, S23 <int>,
## #   S24 <int>, S25 <int>, S26 <int>, S27 <int>, S28 <int>, S29 <int>,
## #   S30 <int>, S31 <int>, S32 <int>, S33 <int>, S34 <int>, S35 <int>,
## #   S36 <int>, S37 <int>, S38 <int>, S39 <int>, S40 <int>, S41 <int>,
## #   S42 <int>, S43 <int>, S44 <int>, S45 <int>, S46 <int>, S47 <int>,
## #   S48 <int>, art.movement <chr>, artist <chr>, painting <chr>
```

The paintings data set has 51 columns where the 48 student rankings are each separate columns.

Combine the 48 student rating columns into one column creating a long data set. Combine the 48 student rating columns into one or more summary column, i.e. the average of all rankings, etc.

```
options(dplyr.summarise.inform = FALSE)
```

```
paintings <- paintings%>%
  pivot_longer(starts_with("s"), names_to = "student", values_to = "rating") %>%
  group_by(art.movement, artist, painting) %>%
  summarise(avg = round(mean(rating),2)) %>%
  arrange(desc(avg))
```

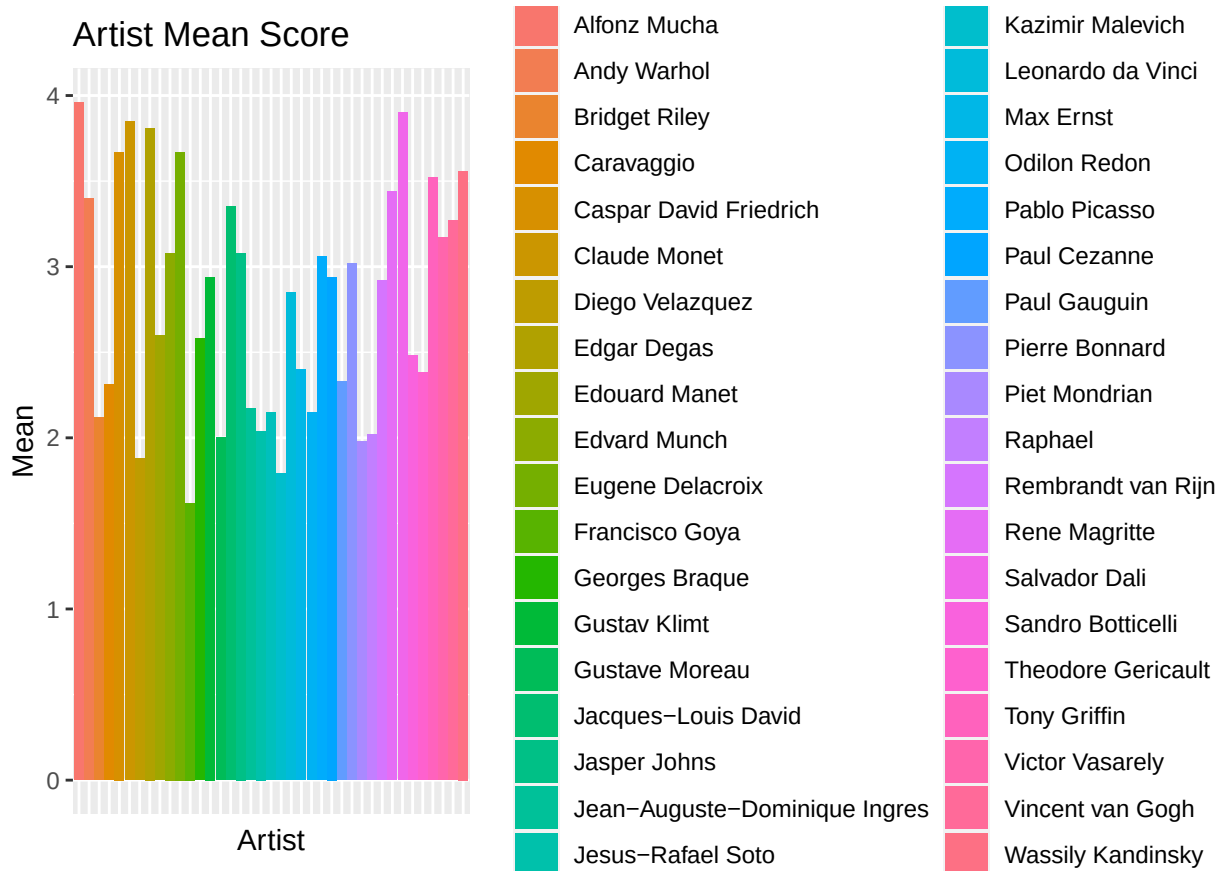
```
head(paintings,10)
```

```
## # A tibble: 10 x 4
## # Groups:   art.movement, artist [10]
##    art.movement  artist                painting                        avg
##    <chr>         <chr>                 <chr>                        <dbl>
##  1 Art Nouveau   Alfonz Mucha          Four Seasons                  3.96
##  2 Surrealism    Salvador Dali         The Temptation of St. Anthony 3.9
##  3 Impressionism Claude Monet          Impression, Sunrise           3.85
##  4 Impressionism Edgar Degas           Ballet Rehearsal              3.81
##  5 Romanticism   Caspar David Friedrich The Cross in the Mountains    3.67
##  6 Romanticism   Eugene Delacroix      Liberty Leading the People    3.67
##  7 Abstract art  Wassily Kandinsky     Moscow. Red Square            3.56
##  8 Pop art       Tony Griffin          Adriana                       3.52
##  9 Surrealism    Rene Magritte         The False Mirror              3.44
## 10 Pop art       Andy Warhol           Campbell's Soup Cans          3.4
```

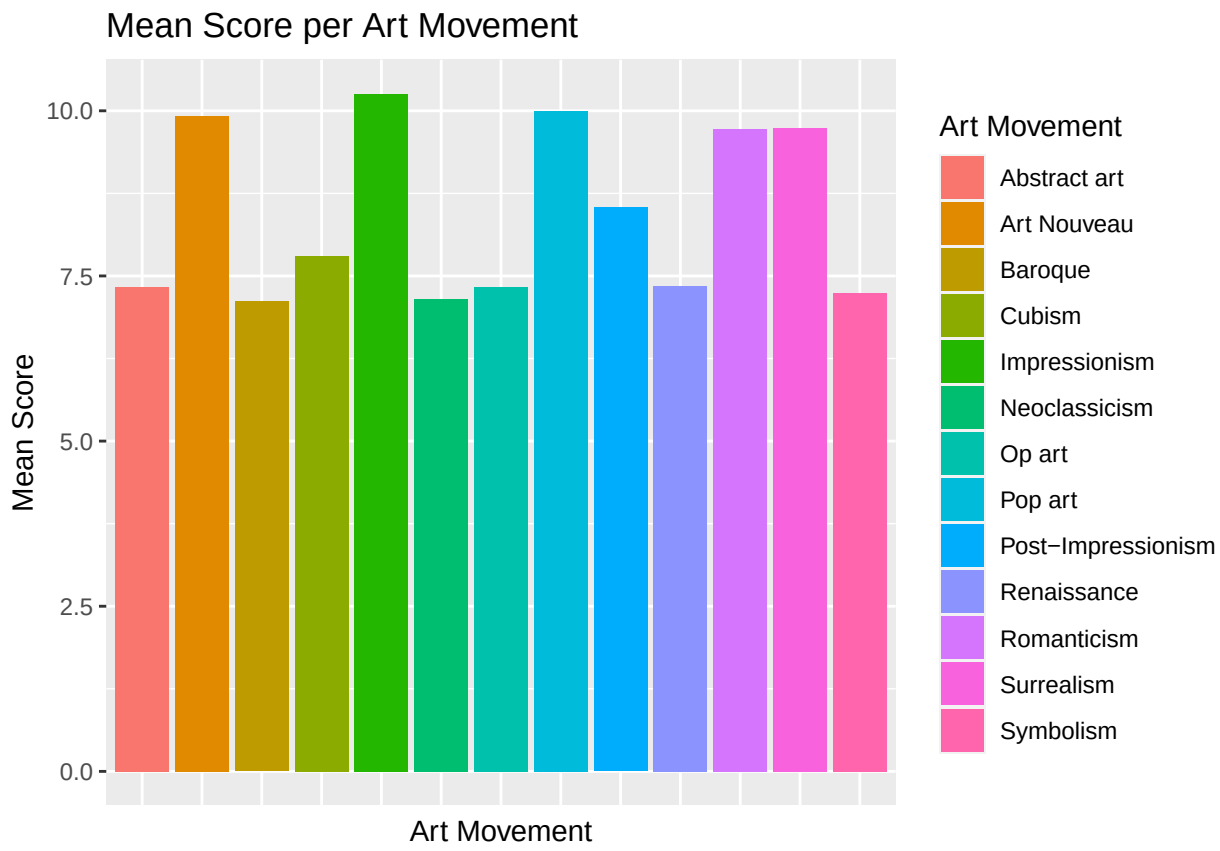
It seems that the painting "Four Seasons" was the most popular piece among students.

Let's see how is the plot of popularity of artists:

```
#Mean by artist
ggplot(paintings , aes(artist, fill = artist, avg)) +
  geom_bar(stat = "identity") +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank()) + labs(x= "Artist",y= "Mean" , title = "Artist Mean Score")
```




```
ggplot(paintings, aes(art.movement, fill = art.movement,avg)) + geom_bar(stat = "identity")+
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank()) + labs(x= "Art Movement",y= "Mean Score" , title = "Mean Score pe
    guides(fill=guide_legend(title="Art Movement"))
```


Mean Score per Art Movement
Mean Score
10.0
7.5
5.0
2.5
0.0
Art Movement
Art Movement
Abstract art
Art Nouveau
Baroque
Cubism
Impressionism
Neoclassicism
Op art
Pop art
Post−Impressionism
Renaissance
Romanticism
Surrealism
Symbolism