LEGOS.R

hadeelmusallam

2023-09-25

##stakeholder: For my LEGO business research I need to plot the average number of parts in LEGO sets over   
##the years (starting with whatever year this dataset starts and ending in whatever year this dataset ends)  
  
##I need this to see if the sets have gotten more complex or less complex over the years.  
setwd("/Users/hadeelmusallam/Library/Mobile Documents/com~apple~CloudDocs/PROJECTS/Legos")  
library(readr)  
library(ggplot2)  
library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

##Load the data  
sets <- read\_csv("sets.csv")

## Rows: 11673 Columns: 5

## ── Column specification ────────────────────────────────────────────────────────  
## Delimiter: ","  
## chr (2): set\_num, name  
## dbl (3): year, theme\_id, num\_parts  
##   
## 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.

##Lets take a look at the structure  
str(sets)

## spc\_tbl\_ [11,673 × 5] (S3: spec\_tbl\_df/tbl\_df/tbl/data.frame)  
## $ set\_num : chr [1:11673] "00-1" "0011-2" "0011-3" "0012-1" ...  
## $ name : chr [1:11673] "Weetabix Castle" "Town Mini-Figures" "Castle 2 for 1 Bonus Offer" "Space Mini-Figures" ...  
## $ year : num [1:11673] 1970 1978 1987 1979 1979 ...  
## $ theme\_id : num [1:11673] 414 84 199 143 143 143 143 186 413 413 ...  
## $ num\_parts: num [1:11673] 471 12 2 12 12 12 18 15 147 149 ...  
## - attr(\*, "spec")=  
## .. cols(  
## .. set\_num = col\_character(),  
## .. name = col\_character(),  
## .. year = col\_double(),  
## .. theme\_id = col\_double(),  
## .. num\_parts = col\_double()  
## .. )  
## - attr(\*, "problems")=<externalptr>

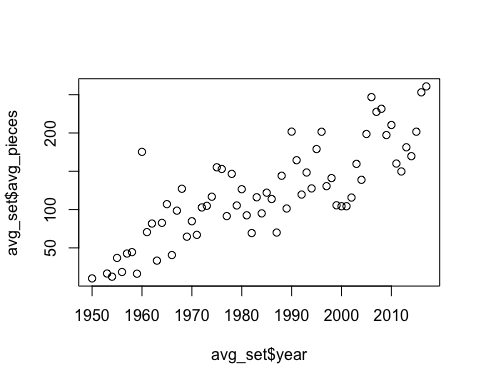
colnames(sets)

## [1] "set\_num" "name" "year" "theme\_id" "num\_parts"

##Lets order the years from oldest to newest   
sets <- sets[order(sets$year),]  
  
##Look at first five rows   
head(sets)

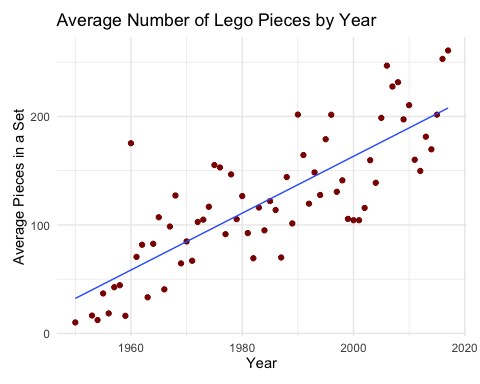
## # A tibble: 6 × 5  
## set\_num name year theme\_id num\_parts  
## <chr> <chr> <dbl> <dbl> <dbl>  
## 1 700.1.1-1 Individual 2 x 4 Bricks 1950 371 10  
## 2 700.1.2-1 Individual 2 x 2 Bricks 1950 371 9  
## 3 700.A-1 Automatic Binding Bricks Small Brick Set (… 1950 366 24  
## 4 700.B.1-1 Individual 1 x 4 x 2 Window (without glass) 1950 371 7  
## 5 700.B.2-1 Individual 1 x 2 x 3 Window (without glass) 1950 371 7  
## 6 700.B.3-1 Individual 1 x 2 x 2 Window (without glass) 1950 371 7

##Group by year, average the number of pieces in a set  
avg\_set <- sets %>%   
 group\_by(year) %>%   
 summarise(avg\_pieces = mean(num\_parts, na.rm = T))  
  
##Basic plot   
plot(avg\_set$year, avg\_set$avg\_pieces)



##ggplot  
ggplot(avg\_set, aes(x = year, y = avg\_pieces)) +   
 geom\_point(color = 'darkred') +   
 geom\_smooth(method = lm, se = FALSE, linewidth = 0.5)+   
 theme\_minimal() +   
 labs(title = 'Average Number of Lego Pieces by Year', x = 'Year',   
 y = 'Average Pieces in a Set') +   
 theme(plot.margin = margin(10,10,10,10, 'pt'))

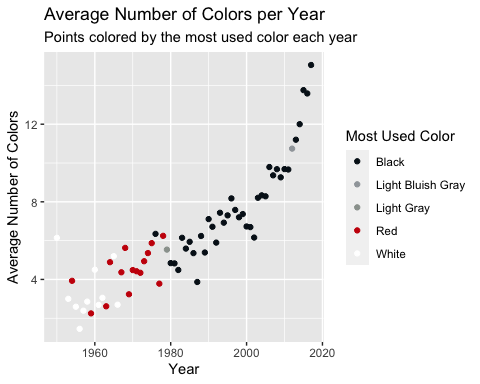
## `geom\_smooth()` using formula = 'y ~ x'



##On average, how many colors were in a set at the beginning? As the sets got slowly bigger over time,   
##how did that impact the colors in sets as time progressed? Bonus points. Make this data visualization   
##colorful and beautiful.  
inventory\_parts <- read.csv('inventory\_parts.csv')  
inventories <- read.csv('inventories.csv')  
colors <- read.csv('colors.csv')  
  
# Merge datasets, remove duplicates, and compute number of colors  
result <- inventory\_parts %>%  
 left\_join(inventories, by = c("inventory\_id" = "id")) %>%  
 left\_join(sets, by = "set\_num") %>%  
 distinct(set\_num, name, color\_id, year) %>%  
 group\_by(set\_num, name, year) %>%  
 summarise(Number\_of\_Colors = n(), .groups = "drop")  
  
##Order by year  
result <- result[order(result$year),]  
  
##Group number of colors by year  
avg\_color\_year <- result %>%   
 group\_by(year) %>%   
 summarise(avg\_color = mean(Number\_of\_Colors, na.rm = T))  
  
######Lets add the most used color per year to the dataset  
# Merge the datasets  
merged\_data <- inventory\_parts %>%  
 left\_join(inventories, by = c("inventory\_id" = "id")) %>%  
 left\_join(sets, by = "set\_num") %>%  
 left\_join(colors, by = c("color\_id" = "id"))  
  
# Group by year and color, summing up the quantities  
color\_counts\_per\_year <- merged\_data %>%  
 group\_by(year, name.y) %>%  
 summarise(total\_quantity = sum(quantity, na.rm = TRUE), .groups = "drop")  
  
# Determine the most used color for each year  
most\_used\_color\_per\_year <- color\_counts\_per\_year %>%  
 arrange(year, desc(total\_quantity)) %>%  
 group\_by(year) %>%  
 slice\_head(n = 1) %>%  
 ungroup()  
  
# Renaming columns for clarity  
colnames(most\_used\_color\_per\_year) <- c('Year', 'Most Used Color', 'Total Quantity')  
  
head(most\_used\_color\_per\_year)

## # A tibble: 6 × 3  
## Year `Most Used Color` `Total Quantity`  
## <dbl> <chr> <int>  
## 1 1950 White 23  
## 2 1953 White 23  
## 3 1954 Red 69  
## 4 1955 White 368  
## 5 1956 White 143  
## 6 1957 White 474

###Add to avg\_color\_year dataset  
avg\_color\_year$most\_used\_color <- most\_used\_color\_per\_year$`Most Used Color`  
  
# Create a named vector of RGB values  
color\_mapping <- setNames(paste0("#", colors$rgb), colors$name)  
  
# Plot with custom color mapping  
ggplot(avg\_color\_year, aes(x = year, y = avg\_color, color = most\_used\_color)) +   
 geom\_point() +  
 labs(title = "Average Number of Colors per Year",  
 subtitle = "Points colored by the most used color each year",  
 x = "Year",  
 y = "Average Number of Colors") +  
 scale\_color\_manual(values = color\_mapping, name = "Most Used Color")



##Stakeholder: Which are the top 3 rarest colors?   
# Count occurrences of each color in inventory\_parts  
color\_counts <- as.data.frame(table(inventory\_parts$color\_id))  
colnames(color\_counts) <- c('id', 'count')  
  
# Merge with colors dataframe to get color names  
merged\_df <- merge(color\_counts, colors, by = 'id', all.x = TRUE)  
  
# Sort by count to get the rarest colors  
rarest\_colors <- head(merged\_df[order(merged\_df$count),], 3)  
  
print(rarest\_colors[,c('name', 'count')])

## name count  
## 10 Trans Light Royal Blue 1  
## 7 Medium Violet 2  
## 98 Very Light Gray 2

##Stakeholder: Which lego set is the most complex? aka has the most steps in the manual  
  
##Stakeholder: which set has the highest number of pieces? what is its name, theme, year of release, cost?  
themes <- read.csv('themes.csv')  
  
# Finding the set with the highest number of pieces  
most\_complex\_set <- sets[which.max(sets$num\_parts),]  
  
# Merging with themes to get the theme name  
most\_complex\_set\_theme <- merge(most\_complex\_set, themes, by.x = 'theme\_id', by.y = 'id')  
  
# Extracting relevant information  
result <- most\_complex\_set\_theme[, c('name.x', 'name.y', 'year', 'num\_parts')]  
colnames(result) <- c('Set Name', 'Theme', 'Year of Release', 'Number of Pieces')  
  
# Display the result  
print(result)

## Set Name Theme Year of Release Number of Pieces  
## 1 Taj Mahal Sculptures 2008 5922