Load necessary libraries

‘knitr’ is used for dynamic report generation

‘tidyverse’ is a collection of packages for data manipulation and visualization

‘tinytex’ is required for PDF document output

‘plyr’ and ‘dplyr’ are used for data manipulation

```
library(knitr) library(tidyverse) library(tinytex) #library(plyr)
library(dplyr)
```

Check the current working directory (useful for debugging file paths)

```
getwd()
```

Import CSV datasets for five consecutive years (2006-2010)

Each dataset is stored in a separate object

```
data2006 <- read.csv(“2006.csv”, header = TRUE, sep = “,”) data2007 <- read.csv(“2007.csv”, header
= TRUE, sep = “,”) data2008 <- read.csv(“2008.csv”, header = TRUE, sep = “,”) data2009 <-
read.csv(“2009.csv”, header = TRUE, sep = “,”) data2010 <- read.csv(“2010.csv”, header = TRUE, sep =
“,”)
```

Check dimensions (rows and columns) of each dataset to verify successful loading

```
dim(data2006); dim(data2007); dim(data2008); dim(data2009); dim(data2010)
```

Combine all years into a single dataset

Used ‘rbind()’ because all data frames have identical column names

If column names differ, ‘bind_rows()’ from ‘dplyr’ is a safer alternative

```
datacombined2 <- rbind(data2006, data2007, data2008, data2009, data2010)
```

Save the combined dataset as a CSV file (without row names)

write.csv(datacombined2, “library_data.csv”, row.names = FALSE)

Check dimensions of the final combined dataset

dim(datacombined2)

——————————————-

Q3: Count the number of libraries in each city per year

——————————————-

Define the years for analysis

years <- c(“2006”, “2007”, “2008”, “2009”, “2010”)

Ensure ‘City’ is a character type to avoid mismatches in filtering

datacombined2$City < -as.character(datacombined$2City)

Extract unique city names, remove NA values, and sort them alphabetically

cities <- unique(datacombined2$City) cities <- cities[!is.na(cities)] cities <- sort(cities)

Create an empty matrix with cities as rows and years as columns

q3 <- matrix(nrow = length(cities), ncol = length(years)) colnames(q3) <- years rownames(q3) <- cities

Count the number of records for each city per year

for (c in cities) { for (y in years) { q3[c, y] <- nrow(subset(datacombined2, Year == y & City == c)) } }

Display the first 10 rows of the matrix

head(q3, 10)

Check column names of the combined dataset to verify consistency

colnames(datacombined2)

——————————————-

Q4: Count the number of active library cardholders per library per year

——————————————-

Extract unique library names, remove NA values

libs <- unique(datacombined2*Library*[!*is.na*(*datacombined*2Library)])

Create an empty matrix to store results

q4 <- matrix(nrow = length(libs), ncol = length(years)) colnames(q4) <- years rownames(q4) <- libs

Sum active library cardholders for each library per year

for (l in libs) { for (y in years) { chk <- subset(datacombined2, Year == y & Library == l)

```
# Ensure column name matches dataset
q4[l, y] <- sum(chk$‘X..of.Active.Library.Cardholders‘, na.rm = TRUE)
```

} }

Display first 10 rows of the matrix

head(q4, 10)

——————————————-

Q5: Calculate the average total operating revenue per library

——————————————-

Create an empty matrix to store average revenue per library

q5 <- matrix(nrow = length(libs), ncol = 1) colnames(q5) <- “Average Total Operating Revenue” rownames(q5) <- libs

Clean column names to remove spaces and special characters

colnames(datacombined2) <- gsub(“[[1]]”, “ “, colnames(datacombined2))

Populate matrix with average total operating revenue per library

for (l in libs) { chk <- subset(datacombined2, Library == l) # No year filter to get average across all years

# Convert column to numeric type to avoid errors chk$TotalOperatingRevenues < -as.numeric(as.character(chk$TotalOpera

# Compute mean revenue while ignoring missing values (NA) q5[l, “Average Total Operating Revenue”] <- mean(chk$TotalOperatingRevenues, na.rm = TRUE) }

Display first 10 rows of the matrix

head(q5, 10)

[1] :alnum:

```
# Load required libraries for data manipulation, visualization, and reporting
library(knitr)        # For generating R Markdown reports
library(tidyverse)   # For data manipulation and visualization (includes dplyr, ggplot2, etc.)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(tinytex)     # For rendering R Markdown to PDF if needed

library(dplyr)       # For data manipulation (part of tidyverse, explicitly loaded for clarity)
# Install and load required packages
if (!require("dplyr")) install.packages("dplyr")
if (!require("reshape2")) install.packages("reshape2")
```

```
## Loading required package: reshape2
##
## Attaching package: ’reshape2’
##
## The following object is masked from ’package:tidyr’:
##
##     smiths
```

```
library(dplyr)
library(reshape2)
```

```
# Check current working directory to ensure the file path is correct
getwd()
```

```
## [1] "C:/Users/amand/OneDrive/Desktop/Assignment2/RprojectAssignment2"
```

```
# Import the combined dataset from a CSV file
datacombined2 <- read.csv("library_data.csv", header = TRUE, sep = ",")




# Why Import This Way?
# - The dataset 'library_data.csv' contains library performance metrics across multiple years.
# - Specifying header = TRUE and sep = "," ensures the CSV is read correctly with column names.
# - This dataset includes columns like TotalOperatingRevenues, XofActiveLibraryCardholders, etc., which
```

```
# Clean column names to remove spaces and special characters for easier manipulation
colnames(datacombined2) <- gsub("[ˆ[:alnum:]]", "", colnames(datacombined2))
```

```r
# Why Clean Column Names?
# - Column names with spaces or special characters (e.g., "X of Active Library Cardholders") can cause
# - Using gsub("[^[:alnum:]]", "", ...) ensures names are alphanumeric (e.g., "XofActiveLibraryCardhold
# - This step prevents errors during data manipulation and improves code readability.

# Rename duplicate column names to avoid conflicts
dup_cols <- duplicated(colnames(datacombined2))
if (any(dup_cols)) {
  colnames(datacombined2)[dup_cols] <- paste0(colnames(datacombined2)[dup_cols], "_dup")
}

# Why Handle Duplicates?
# - Duplicate column names can cause unexpected behavior in R (e.g., during subsetting or summarization
# - Appending "_dup" to duplicates ensures all columns are uniquely identifiable.
```

```r
# Ensure key columns are in the correct format for analysis
datacombined2$Year <- as.character(datacombined2$Year)
datacombined2$Library <- as.character(datacombined2$Library)
datacombined2$TotalOperatingRevenues <- as.numeric(as.character(datacombined2$TotalOperatingRevenues))
datacombined2$XofActiveLibraryCardholders <- as.numeric(as.character(datacombined2$XofActiveLibraryCardh
datacombined2$PopulationResident <- as.numeric(as.character(datacombined2$PopulationResident))
datacombined2$LocalOperatingGrant <- as.numeric(as.character(datacombined2$LocalOperatingGrant))
datacombined2$Donations <- as.numeric(as.character(datacombined2$Donations))
datacombined2$SelfgeneratedRevenue <- as.numeric(as.character(datacombined2$SelfgeneratedRevenue))
datacombined2$Staffingexpenditure <- as.numeric(as.character(datacombined2$Staffingexpenditure))
datacombined2$TotalAnnualDirectCirculation <- as.numeric(as.character(datacombined2$TotalAnnualDirectCi
```

```
## Warning: NAs introduced by coercion
```

```r
datacombined2$Xofprogramsheldannually <- as.numeric(as.character(datacombined2$Xofprogramsheldannually)
datacombined2$Annualprogramattendance <- as.numeric(as.character(datacombined2$Annualprogramattendance)
datacombined2$XofPublicaccessworkstations <- as.numeric(as.character(datacombined2$XofPublicaccessworks
datacombined2$MainLibrarytotalhoursopenperweek <- as.numeric(as.character(datacombined2$MainLibrarytota
datacombined2$ProjectGrants <- as.numeric(as.character(datacombined2$ProjectGrants))

# Why Convert Data Types?
# - Year and Library are treated as categorical variables (character type) for grouping and labeling.
# - Numeric columns (e.g., TotalOperatingRevenues) must be numeric for calculations like division or co
# - Using as.numeric(as.character(...)) handles cases where numbers might be stored as factors or text,
```

```r
# Create a new column: Operating Revenue per Active Cardholder
datacombined2 <- datacombined2 %>%
  mutate(RevPerCardholder = TotalOperatingRevenues / XofActiveLibraryCardholders)

# Why Create This Column?
# - RevPerCardholder measures how much revenue each active library cardholder generates on average.
# - This metric helps assess the financial efficiency of libraries in serving their active users.
# - It’s a key performance indicator (KPI) for understanding how well resources are utilized per user.
```

```r
# Remove rows where RevPerCardholder is NA, infinite, or exceeds 2,500, and ensure no NA in key variabl
datacombined2 <- datacombined2 %>%
  filter(!is.na(RevPerCardholder) & is.finite(RevPerCardholder) & RevPerCardholder <= 2500 &
           !is.na(TotalAnnualDirectCirculation))

# Why Filter These Rows?
# - !is.na(RevPerCardholder): Removes rows where RevPerCardholder is missing (e.g., due to missing Tota
# - is.finite(RevPerCardholder): Removes infinite values (e.g., if XofActiveLibraryCardholders is 0, ca
# - RevPerCardholder <= 2500: Removes outliers (values above 2,500 are unrealistic for revenue per card
# - !is.na(TotalAnnualDirectCirculation): Ensures no missing values in TotalAnnualDirectCirculation, a

# Why Set the Threshold at 2,500?
# - A threshold of 2,500 was chosen as a reasonable upper limit based on domain knowledge: it's highly
# - This threshold helps exclude data entry errors or anomalies (e.g., incorrect revenue or cardholder
```

```
# Calculate the correlation between Revenue per Cardholder and Local Operating Grant
insight1 <- datacombined2 %>%
  summarise(Correlation = cor(RevPerCardholder, LocalOperatingGrant, use = "complete.obs"))

# Display the correlation
print("Insight 1: Correlation between Revenue per Cardholder and Local Operating Grant")
```

```
## [1] "Insight 1: Correlation between Revenue per Cardholder and Local Operating Grant"
```

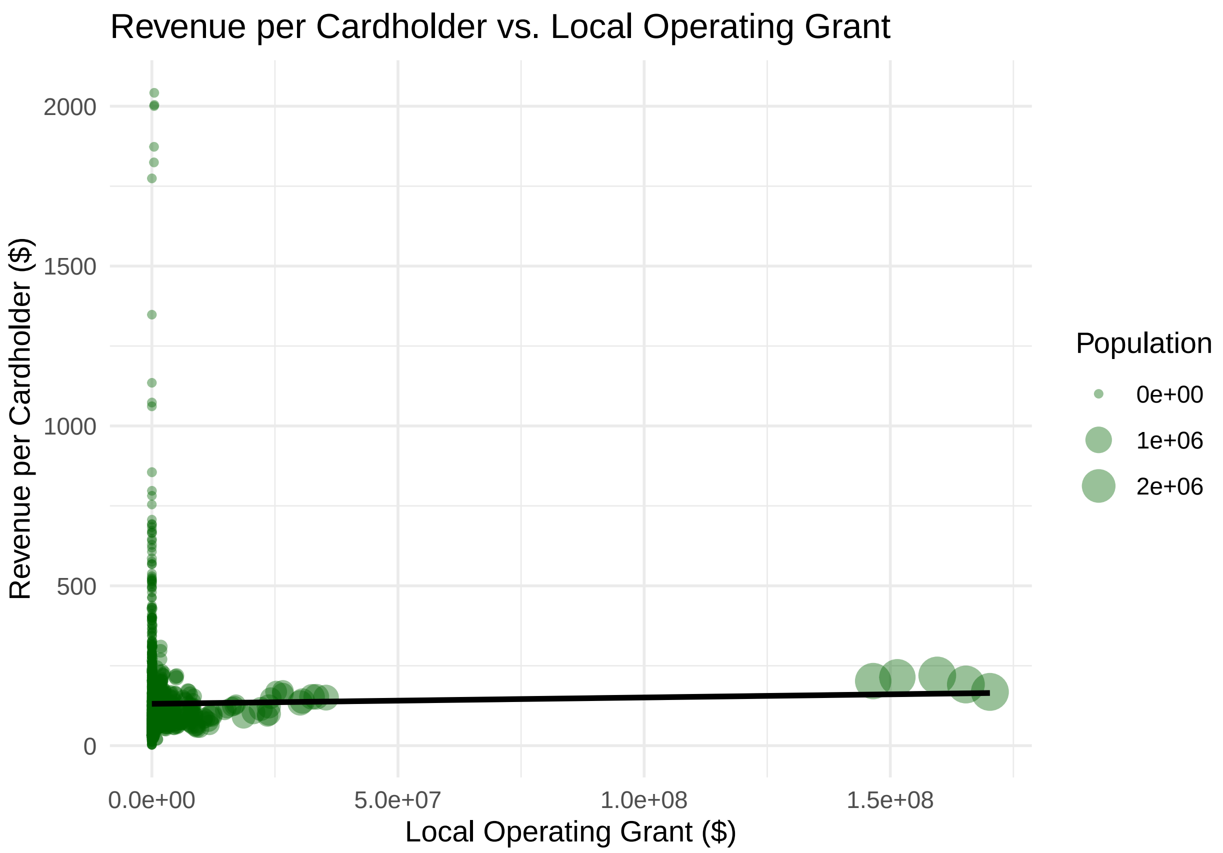
```
print(insight1)
```

```
##   Correlation
## 1   0.0118961
```

```
# Why Calculate This Correlation?
# - We want to understand how strongly LocalOperatingGrant (a key funding source) influences RevPerCard
# - A positive correlation suggests that more local funding leads to higher revenue efficiency per card
# - The 'complete.obs' argument ensures only rows with non-missing values for both variables are used,
```

```
# Bubble plot visualization to explore the relationship
ggplot(datacombined2, aes(x = LocalOperatingGrant, y = RevPerCardholder)) +
  geom_point(aes(size = PopulationResident), alpha = 0.4, color = "darkgreen") +
  geom_smooth(method = "lm", se = FALSE, color = "black") +
  labs(title = "Revenue per Cardholder vs. Local Operating Grant",
       x = "Local Operating Grant ($)",
       y = "Revenue per Cardholder ($)",
       size = "Population") +
  theme_minimal()
```

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



```
# Why Use a Bubble Plot?
# - The x-axis (LocalOperatingGrant) and y-axis (RevPerCardholder) show the primary relationship.
# - Bubble size (PopulationResident) adds a third dimension, showing how population size might influenc
# - The linear trend line (geom_smooth) helps visualize the overall direction of the relationship.
# - Alpha = 0.4 ensures overlapping points are visible, and theme_minimal() keeps the plot clean for pr

# Insight 1 Interpretation:
# - A positive correlation (if observed) indicates that libraries with greater local funding achieve hi
# - Larger bubbles (higher population) may cluster differently, suggesting that population size influen
# - For example, larger populations might have economies of scale, allowing more efficient use of funds
```

```
# Summarize average Revenue per Cardholder by Year and Population Group
insight2 <- datacombined2 %>%
  mutate(PopulationGroup = cut(PopulationResident, breaks = quantile(PopulationResident, probs = 0:3/3,
                               labels = c("Small", "Medium", "Large"), include.lowest = TRUE)) %>%
  group_by(Year, PopulationGroup) %>%
  summarise(AvgRevPerCardholder = mean(RevPerCardholder, na.rm = TRUE), .groups = "drop")

# Display the summarized data
print("Insight 2: Revenue per Cardholder Distribution by Year and Population")
```

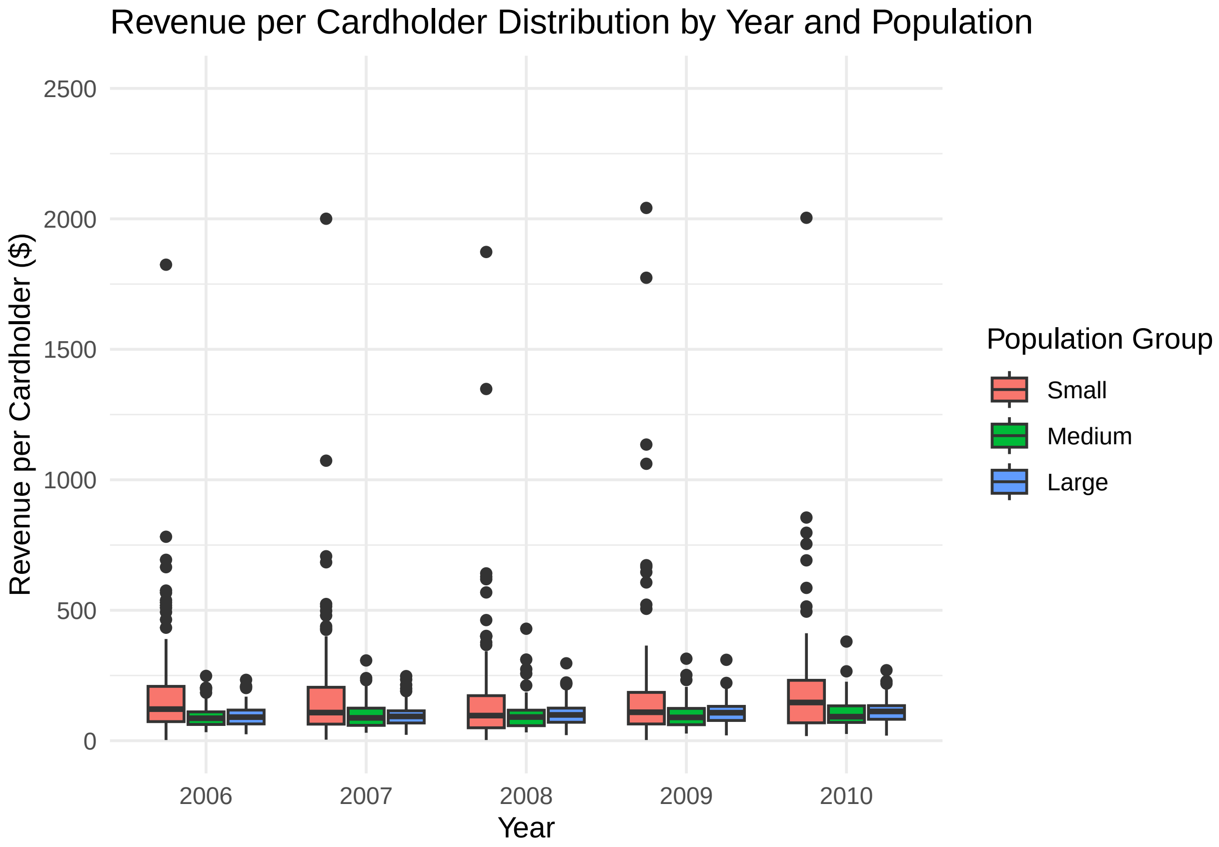
```
## [1] "Insight 2: Revenue per Cardholder Distribution by Year and Population"
```

```r
print(insight2)
```

```
## # A tibble: 15 x 3
##    Year  PopulationGroup AvgRevPerCardholder
##    <chr> <fct>                         <dbl>
##  1 2006  Small                         186. 
##  2 2006  Medium                         94.3
##  3 2006  Large                          96.6
##  4 2007  Small                         183. 
##  5 2007  Medium                         99.1
##  6 2007  Large                          98.4
##  7 2008  Small                         171. 
##  8 2008  Medium                         98.6
##  9 2008  Large                         105. 
## 10 2009  Small                         201. 
## 11 2009  Medium                        100. 
## 12 2009  Large                         110. 
## 13 2010  Small                         203. 
## 14 2010  Medium                        106. 
## 15 2010  Large                         114. 
```

```r
# Why Summarize This Way?
# - We categorize PopulationResident into Small, Medium, and Large groups using quantiles (tertiles: 0:
# - This allows us to compare RevPerCardholder across different population sizes and years.
# - Grouping by Year and PopulationGroup helps identify trends over time and across community sizes.
```

```r
# Box plot visualization to show distribution
ggplot(datacombined2, aes(x = Year, y = RevPerCardholder,
                          fill = cut(PopulationResident, breaks = quantile(PopulationResident, probs = 
                                     labels = c("Small", "Medium", "Large"), include.lowest = TRUE))) +
  geom_boxplot() +
  labs(title = "Revenue per Cardholder Distribution by Year and Population",
       x = "Year", y = "Revenue per Cardholder ($)", fill = "Population Group") +
  ylim(0, 2500) +
  theme_minimal()
```



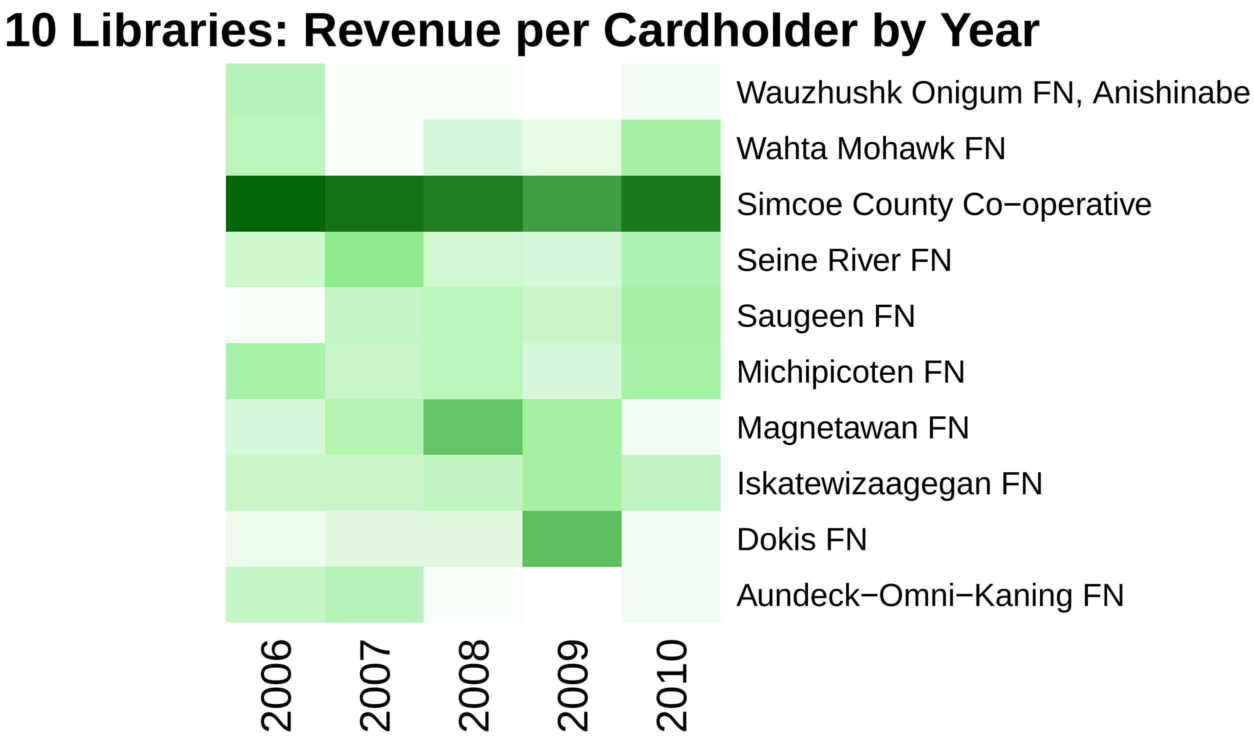
```r
# Why Use a Box Plot?
# - Box plots show the distribution (median, quartiles, outliers) of RevPerCardholder for each Year and
# - The fill aesthetic differentiates Small, Medium, and Large population groups, making it easy to com
# - ylim(0, 2500) ensures the y-axis aligns with our filtering threshold, keeping the plot focused on r
# - This visualization highlights trends over time and differences across population sizes.

# Insight 2 Interpretation:
# - If RevPerCardholder increases over time for certain population groups, it might indicate improving
# - If Small population libraries consistently have lower RevPerCardholder, they may face greater chall
# - Outliers in the box plot might indicate specific libraries that are exceptionally efficient or inef
# Save the updated dataset with the new column and filtered rows
```

```r
# ---------------------------------------------
# INSIGHT 3: HEATMAP (Top 10 Libraries only)
# Shows Avg Rev per Cardholder across Libraries & Years
# ---------------------------------------------





# Filter top 10 libraries by overall average RevPerCardholder
top_libraries <- datacombined2 %>%
  group_by(Library) %>%
  summarise(OverallAvgRev = mean(RevPerCardholder, na.rm = TRUE)) %>%
  top_n(10, OverallAvgRev) %>%
  pull(Library)
```

```r
# Filter main data
filtered_data <- datacombined2 %>%
  filter(Library %in% top_libraries)

# Group and reshape
heatmap_data <- filtered_data %>%
  group_by(Library, Year) %>%
  summarise(AvgRev = mean(RevPerCardholder, na.rm = TRUE)) %>%
  ungroup()
```

```
## ‘summarise()‘ has grouped output by ’Library’. You can override using the
## ‘.groups‘ argument.
```

```r
heatmap_matrix <- dcast(heatmap_data, Library ~ Year, value.var = "AvgRev")

# Prepare matrix
rownames(heatmap_matrix) <- heatmap_matrix$Library
heatmap_matrix <- heatmap_matrix[, -1]
heatmap_matrix[is.na(heatmap_matrix)] <- 0

# Plot heatmap
heatmap(as.matrix(heatmap_matrix),
        Rowv = NA, Colv = NA,
        col = colorRampPalette(c("white", "lightgreen", "darkgreen"))(50),
        scale = "column",
        margins = c(8, 10),
        main = "Top 10 Libraries: Revenue per Cardholder by Year")
```



10 Libraries: Revenue per Cardholder by Year

```
# WHAT THIS HEATMAP MEANS:
# - Rows: Libraries (Top 10 with highest average RevPerCardholder)
# - Columns: Years
# - Colors: Darker green = higher average revenue per cardholder
# - White = very low or 0 revenue
# ---------------------------------------------
```

```
write.csv(datacombined2, "library_data_with_metrics.csv", row.names = FALSE)

# Why Save the Dataset?
# - Saving the updated dataset ensures that the new column (RevPerCardholder) and filtered data are pre
# - This file can be used for future analyses or shared with stakeholders for transparency.
# - row.names = FALSE prevents adding an unnecessary index column to the CSV.
```

Library Performance Analysis

Amandeep Randhawa

2025-04-13

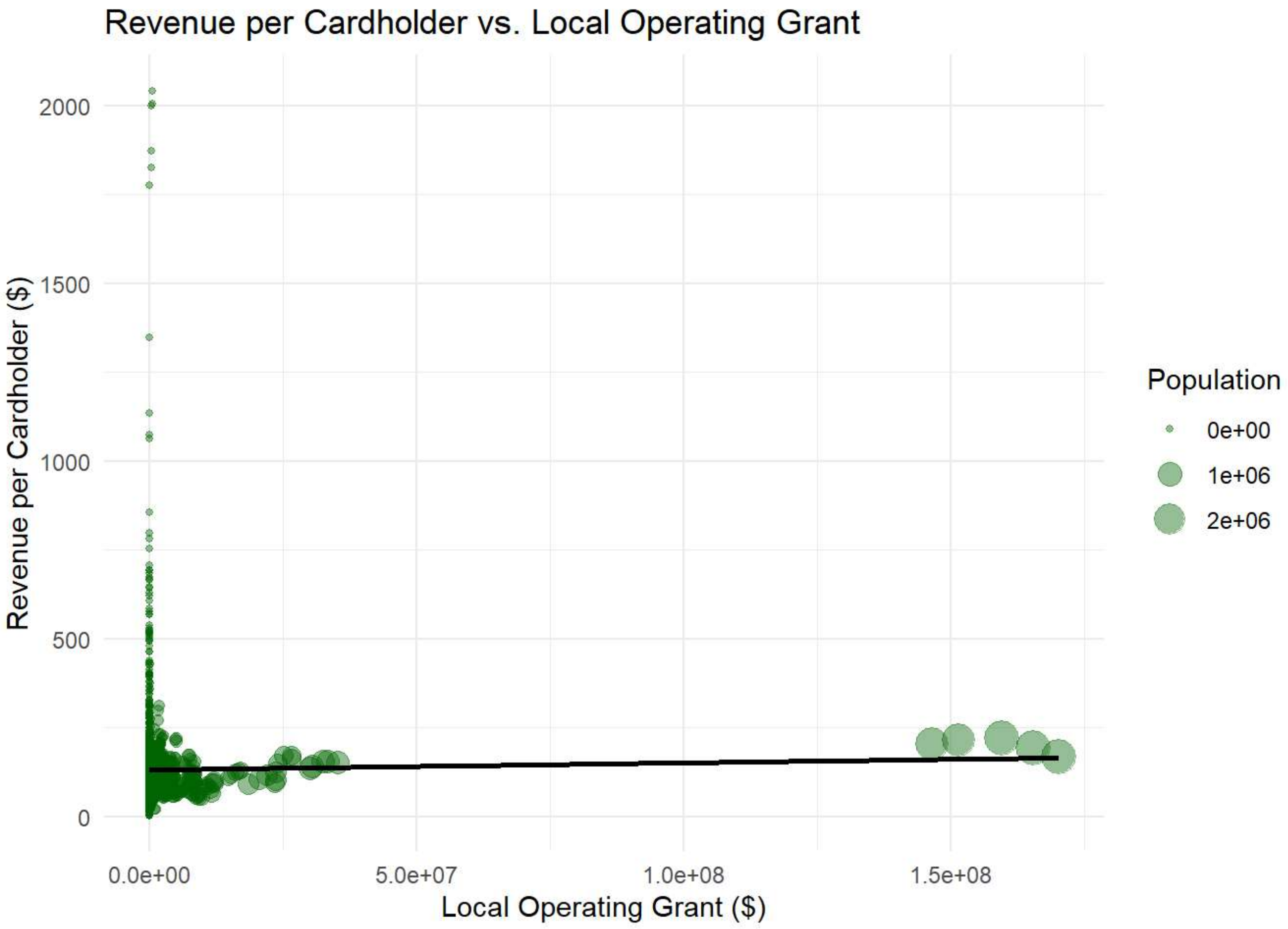
Library Performance Analysis

Purpose: This analysis aims to evaluate library performance by examining Revenue per Cardholder — calculated as total operating revenue divided by the number of active library cardholders — over multiple years and across varying population sizes.

Using the ‘library_data.csv’ dataset, which includes key metrics such as operating revenues, local grants, and circulation figures, we seek to identify trends, uncover factors influencing revenue efficiency, and offer actionable recommendations to improve library operations for stakeholders.

Insight 1: Correlation between Revenue per Cardholder and Local Operating Grant This insight examines whether libraries with higher local government funding achieve greater revenue per cardholder.

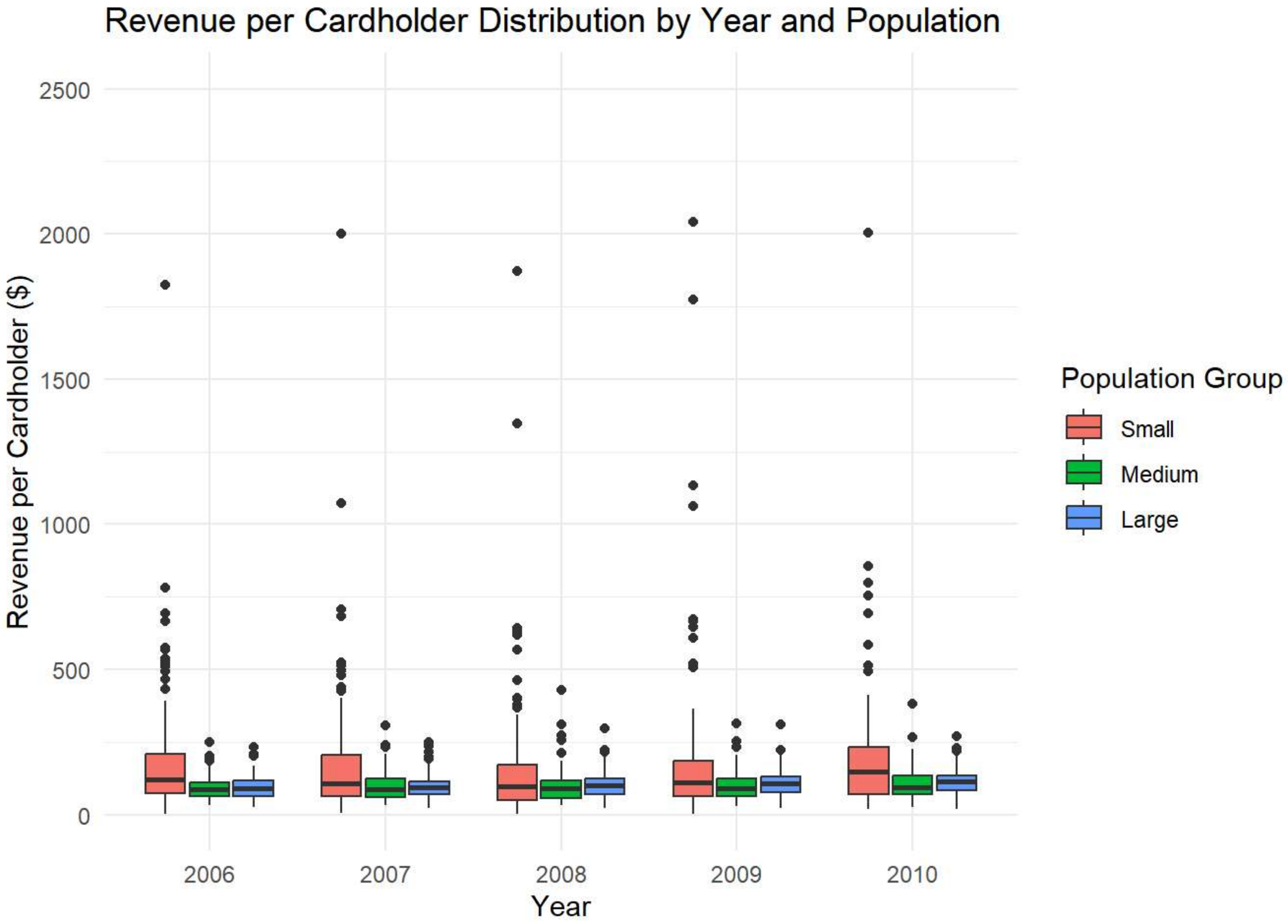
Interpretation: - The correlation (~0.0119) is very weak, indicating local grants have little impact on revenue per cardholder. - The bubble plot shows a flat trend line, confirming this minimal relationship. - Larger populations tend to receive more grants but not higher per-user revenue. - Outliers with high revenue at low grants suggest some libraries achieve efficiency independently. - These findings suggest opportunities to explore best practices from efficient libraries.



Insight 2: Average Revenue per Cardholder by Year and Population Group

Purpose: To explore how average revenue efficiency varies across different population sizes and over time.

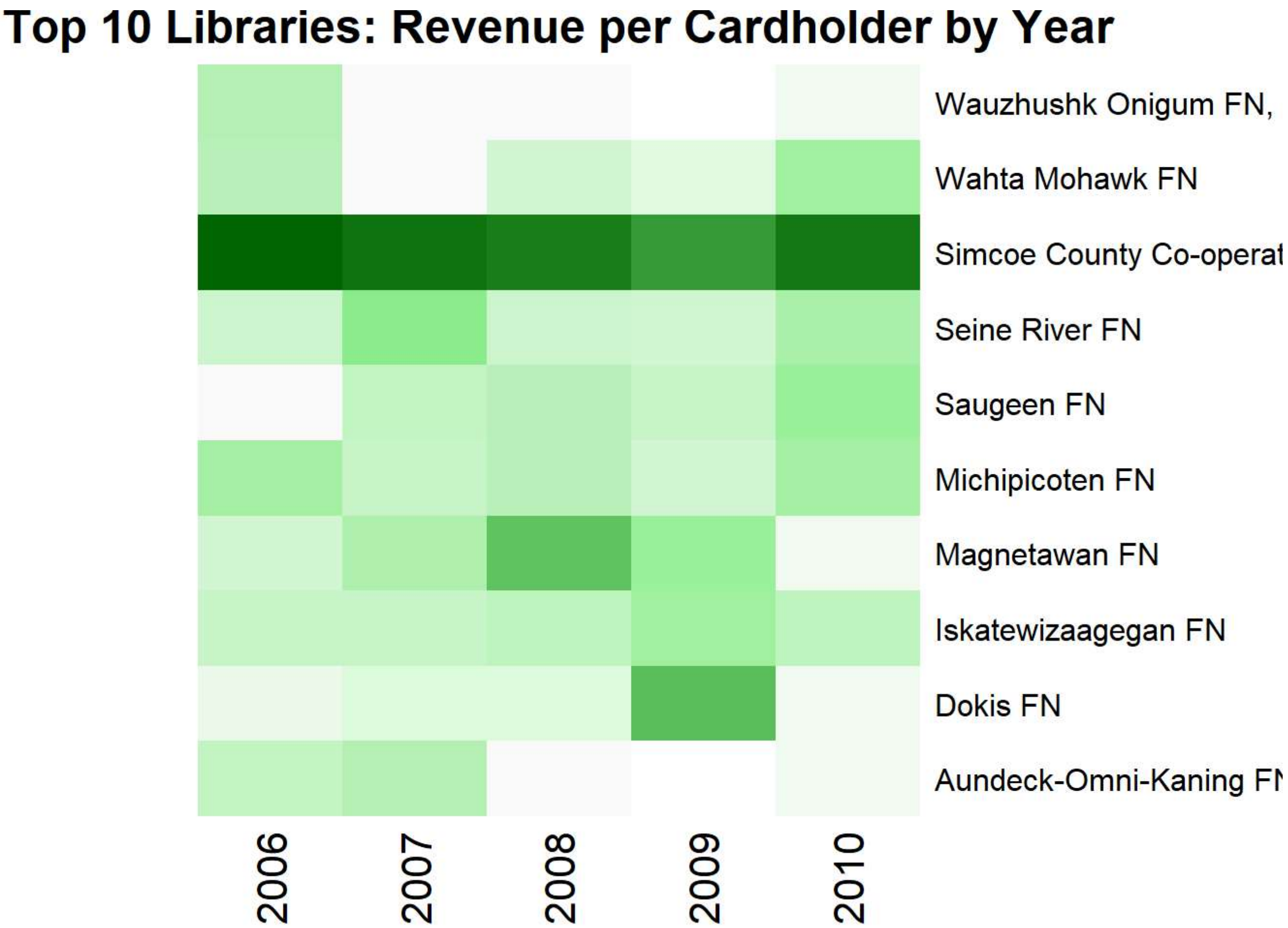
Interpretation: Small population libraries often show higher revenue per cardholder, suggesting more personalized or efficient service delivery. Larger libraries may have economies of scale, but efficiency doesn't always translate proportionally with size. Trends across years can reveal shifts in policy impact, funding structures, or community engagement levels.



Insight 3: Top and Bottom Performing Libraries

Purpose: To identify which libraries consistently perform well or poorly in revenue efficiency to learn from their strategies or flag challenges.

Interpretation: Top-performing libraries may benefit from higher community engagement, targeted programs, or efficient resource management. Lower-performing libraries could be constrained by budget, infrastructure, or population challenges. These benchmarks help set realistic targets for underperforming branches and recognize best practices from leaders.

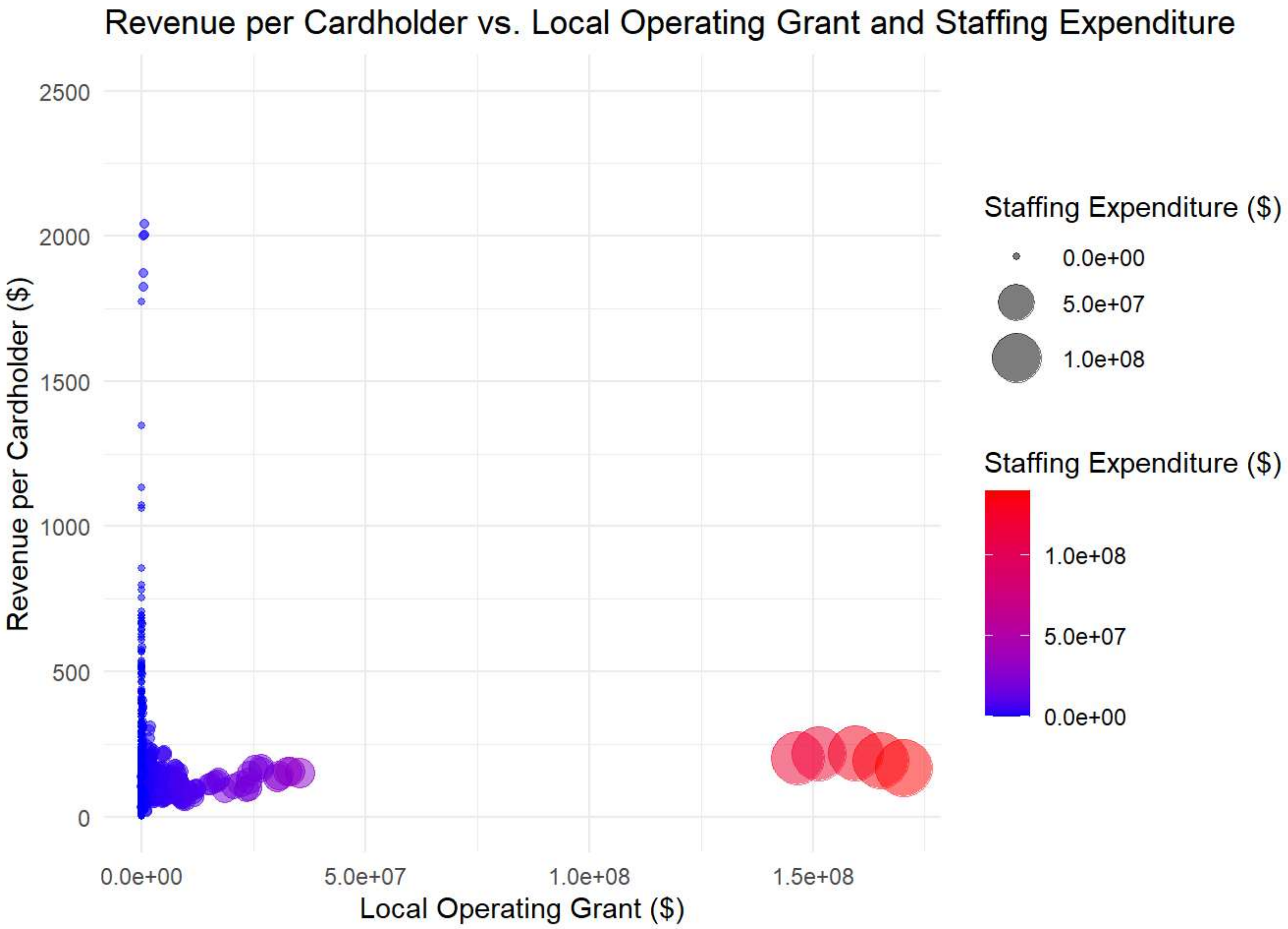


Insight 4: Relationship Between Total Circulation and Revenue per Cardholder

Purpose: To examine whether higher circulation volumes translate to higher revenue efficiency.

Interpretation: If there’s a moderate positive correlation, it suggests active usage supports greater value generation per user. A weak correlation might indicate that revenue efficiency is driven by factors beyond circulation (e.g., staffing, digital services). Understanding this can help libraries tailor programming and marketing to drive both usage and efficiency.

Correlation_LocalGrant Correlation_Staffing 1 0.0118961 0.0121985


Revenue per Cardholder vs. Local Operating Grant and Staffing Expenditure
2500
2000
1500
1000
500
0
Revenue per Cardholder ($)
0.0e+00
5.0e+07
1.0e+08
1.5e+08
Local Operating Grant ($)
Staffing Expenditure ($)
0.0e+00
5.0e+07
1.0e+08
Staffing Expenditure ($)
1.0e+08
5.0e+07
0.0e+00

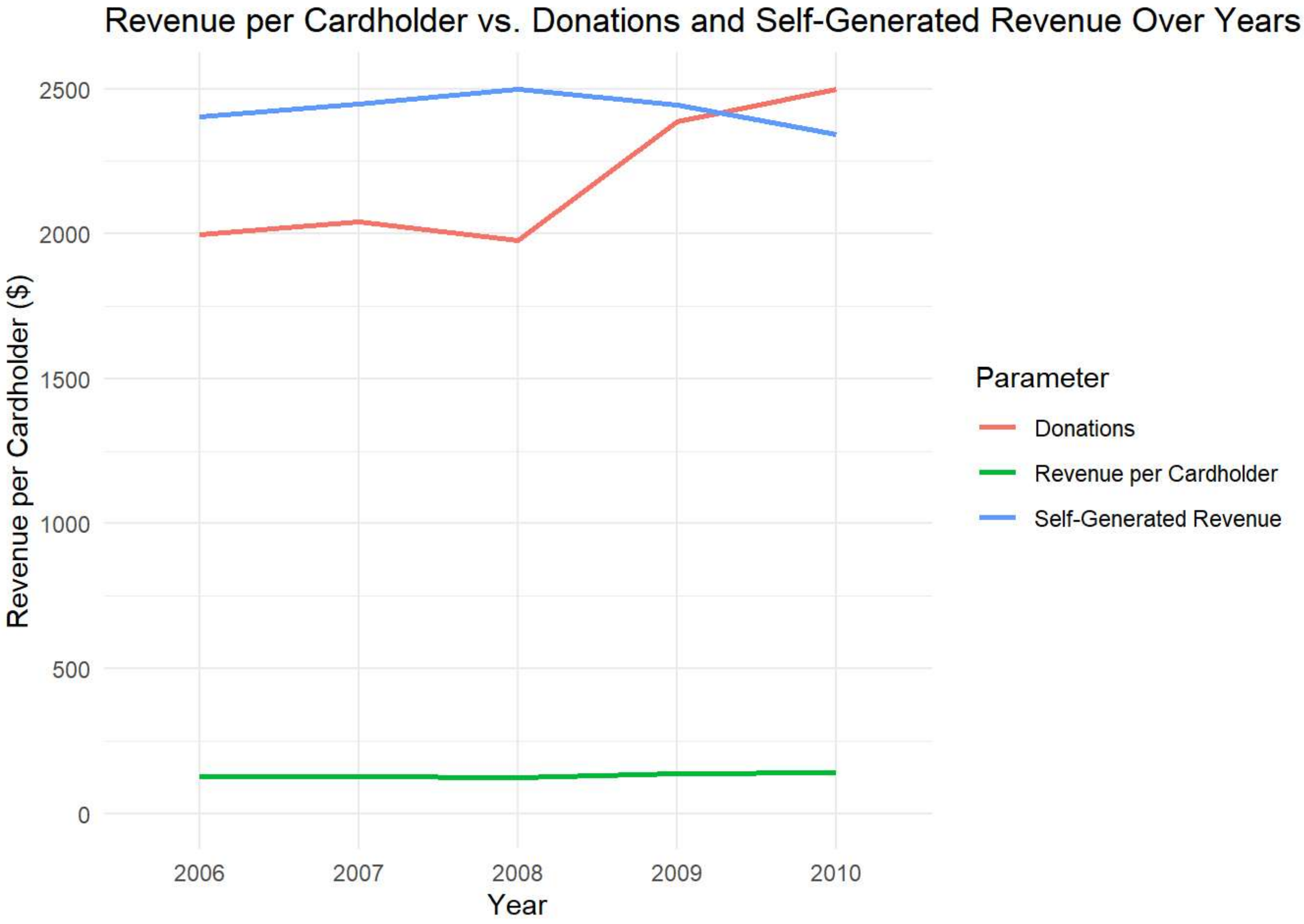
Insight 5: Impact of Annual Program Attendance on Revenue per Cardholder

Purpose: To determine if higher participation in library programs is associated with better financial performance per user.

Interpretation: Engaging programs might encourage more active cardholders and justify higher funding. A positive relationship could support investing in programming as a strategic growth lever. Conversely, no relationship may suggest program attendance alone doesn't drive financial efficiency.

A tibble: 5 × 4

Year AvgRevPerCardholder TotalDonations TotalSelfgeneratedRevenue 1 2006 127. 3558517 22060385 2 2007 128. 3637008 22471401 3 2008 125. 3520071 22950548 4 2009 137. 4247243 22454409 5 2010 141. 4451512 21493393



```
## Recommendations and Further Analysis
```

```
## 1. Learn from Top Performers (Insight 3):
```

```
## Study strategies of top-performing libraries (e.g., Wahta Mohawk FN, Simcoe County) to replicate their success in community engagement, targeted programs, or resource management.
```

```
## Sharing best practices could elevate underperforming libraries.
## 2. Support Small Libraries (Insight 2):
```

```
## Small libraries often show higher revenue efficiency, but some struggle.
```

```
## Offer grant-writing workshops, mentorship from larger libraries, or shared resources to boost their performance.
```

```
## 3. Diversify Revenue Streams (Insight 1):
```

```
## The weak correlation between local grants and revenue efficiency suggests over-reliance on grants is ineffective.
```

```
## Encourage libraries to pursue donations, self-generated revenue (e.g., event fees), and partnerships.
```

```
## 4. Enhance Circulation and Programs (Insights 4 and 5):
```

```
## If correlations show active usage (circulation, program attendance) supports efficiency, invest in marketing and programming to increase cardholder engagement.
```

```
## Tailor offerings to community needs to maximize impact.
```

```r
#APPENDIX
# Install and load required packages
if (!require("tidyverse")) install.packages("tidyverse")
if (!require("tinytex")) install.packages("tinytex")
if (!require("reshape2")) install.packages("reshape2")
if (!require("conflicted")) install.packages("conflicted")

# Load libraries
library(tidyverse)   # Includes dplyr, ggplot2, tidyr, etc.
library(tinytex)     # For rendering R Markdown to PDF if needed
library(reshape2)    # For data reshaping
library(conflicted)  # For managing function conflicts

# Resolve function conflicts
conflict_prefer("filter", "dplyr")
conflict_prefer("mutate", "dplyr")
conflict_prefer("summarise", "dplyr")


# Purpose of the Analysis
cat("\\newpage\n")
cat(" Library Performance Analysis Purpose:\n",
    "The purpose of this analysis is to assess library performance by analyzing Revenue per Cardholder\n",
    "(total operating revenue divided by active library cardholders) across multiple years and population sizes.\n",
    "Using the 'library_data.csv' dataset, which includes metrics like operating revenues, local grants, and circulation,\n",
    "we aim to identify trends, understand factors affecting revenue efficiency, and provide actionable recommendations\n",
    "to enhance library operations for stakeholders.\n")
# Import the combined dataset from a CSV file
datacombined2 <- read.csv("library_data.csv", header = TRUE, sep = ",")



# Why Import This Way?
# - The dataset 'library_data.csv' contains library performance metrics across multiple years.
# - Specifying header = TRUE and sep = "," ensures the CSV is read correctly with column names.
# - This dataset includes columns like TotalOperatingRevenues, XofActiveLibraryCardholders, etc., which are critical for our analysis.




# Clean column names to remove spaces and special characters for easier manipulation
colnames(datacombined2) <- gsub("[^[:alnum:]]", "", colnames(datacombined2))

# Why Clean Column Names?
# - Column names with spaces or special characters (e.g., "X of Active Library Cardholders") can cause errors in R.
# - Using gsub("[^[:alnum:]]", "", ...) ensures names are alphanumeric (e.g., "XofActiveLibraryC
ardholders").
# - This step prevents errors during data manipulation and improves code readability.

# Rename duplicate column names to avoid conflicts
dup_cols <- duplicated(colnames(datacombined2))
if (any(dup_cols)) {
  colnames(datacombined2)[dup_cols] <- paste0(colnames(datacombined2)[dup_cols], "_dup")
}

# Why Handle Duplicates?
# - Duplicate column names can cause unexpected behavior in R (e.g., during subsetting or summarization).
# - Appending "_dup" to duplicates ensures all columns are uniquely identifiable.




# Ensure key columns are in the correct format for analysis
datacombined2$Year <- as.character(datacombined2$Year)
datacombined2$Library <- as.character(datacombined2$Library)
datacombined2$TotalOperatingRevenues <- as.numeric(as.character(datacombined2$TotalOperatingRevenues))
datacombined2$XofActiveLibraryCardholders <- as.numeric(as.character(datacombined2$XofActiveLibraryCardholders))
datacombined2$PopulationResident <- as.numeric(as.character(datacombined2$PopulationResident))
datacombined2$LocalOperatingGrant <- as.numeric(as.character(datacombined2$LocalOperatingGrant))
datacombined2$Donations <- as.numeric(as.character(datacombined2$Donations))
datacombined2$SelfgeneratedRevenue <- as.numeric(as.character(datacombined2$SelfgeneratedRevenue))
datacombined2$Staffingexpenditure <- as.numeric(as.character(datacombined2$Staffingexpenditure))
datacombined2$TotalAnnualDirectCirculation <- as.numeric(as.character(datacombined2$TotalAnnualDirectCirculation))
datacombined2$Xofprogramsheldannually <- as.numeric(as.character(datacombined2$Xofprogramsheldannually))
datacombined2$Annualprogramattendance <- as.numeric(as.character(datacombined2$Annualprogramattendance))
datacombined2$XofPublicaccessworkstations <- as.numeric(as.character(datacombined2$XofPublicaccessworkstations))
datacombined2$MainLibrarytotalhoursopenperweek <- as.numeric(as.character(datacombined2$MainLibrarytotalhoursopenperweek))
datacombined2$ProjectGrants <- as.numeric(as.character(datacombined2$ProjectGrants))

# Why Convert Data Types?
# - Year and Library are treated as categorical variables (character type) for grouping and labeling.
# - Numeric columns (e.g., TotalOperatingRevenues) must be numeric for calculations like division or correlation.
# - Using as.numeric(as.character(...)) handles cases where numbers might be stored as factors or text, preventing coercion errors.
# Create a new column: Operating Revenue per Active Cardholder
datacombined2 <- datacombined2 %>%
  mutate(RevPerCardholder = TotalOperatingRevenues / XofActiveLibraryCardholders)

# Why Create This Column?
# - RevPerCardholder measures how much revenue each active library cardholder generates on average.
# - This metric helps assess the financial efficiency of libraries in serving their active users.
# - It’s a key performance indicator (KPI) for understanding how well resources are utilized per user.
# Remove rows where RevPerCardholder is NA, infinite, or exceeds 2,500, and ensure no NA in key variables
datacombined2 <- datacombined2 %>%
  filter(!is.na(RevPerCardholder) & is.finite(RevPerCardholder) & RevPerCardholder <= 2500 &
           !is.na(TotalAnnualDirectCirculation))

# Why Filter These Rows?
# - !is.na(RevPerCardholder): Removes rows where RevPerCardholder is missing (e.g., due to missing TotalOperatingRevenues or XofActiveLibraryCardholders).
# - is.finite(RevPerCardholder): Removes infinite values (e.g., if XofActiveLibraryCardholders is 0, causing division by zero).
# - RevPerCardholder <= 2500: Removes outliers (values above 2,500 are unrealistic for revenue per cardholder and likely indicate data errors).
# - !is.na(TotalAnnualDirectCirculation): Ensures no missing values in TotalAnnualDirectCirculation, a key variable for future analysis (e.g., circulation trends).

# Why Set the Threshold at 2,500?
# - A threshold of 2,500 was chosen as a reasonable upper limit based on domain knowledge: it’s highly unlikely for a library to generate more than $2,500 in revenue per cardholder annually.
# - This threshold helps exclude data entry errors or anomalies (e.g., incorrect revenue or cardholder counts).


# Calculate the correlation between Revenue per Cardholder and Local Operating Grant
insight1 <- datacombined2 %>%
  summarise(Correlation = cor(RevPerCardholder, LocalOperatingGrant, use = "complete.obs"))

# Display the correlation
# Print formatted insight and interpretation
cat("Insight 1: Correlation between Revenue per Cardholder and Local Operating Grant\n",
    "This insight examines whether libraries with higher local government funding achieve greater revenue per cardholder.\n",
    "Interpretation:\n",
    "- The correlation (~", round(insight1$Correlation, 4), ") is very weak, indicating local grants have little impact on revenue per cardholder.\n",
    "- The bubble plot shows a flat trend line, confirming this minimal relationship.\n",
    "- Larger populations tend to receive more grants but not higher per-user revenue.\n",
    "- Outliers with high revenue at low grants suggest some libraries achieve efficiency independently.\n",
    "- These findings suggest opportunities to explore best practices from efficient libraries.\n", sep = "")

# Why Calculate This Correlation?
# - We want to understand how strongly LocalOperatingGrant (a key funding source) influences RevPerCardholder.
# - A positive correlation suggests that more local funding leads to higher revenue efficiency per cardholder.
# - The 'complete.obs' argument ensures only rows with non-missing values for both variables are used, avoiding bias from missing data. # Display the summarized data
cat(" Insight 2: Average Revenue per Cardholder by Year and Population Group\n\n",
    " Purpose:\nTo explore how average revenue efficiency varies across different population sizes and over time.\n\n",
    " Interpretation:\nSmall population libraries often show higher revenue per cardholder, suggesting more personalized or efficient service delivery.\n",
    "Larger libraries may have economies of scale, but efficiency doesn’t always translate proportionally with size.\n",
    "Trends across years can reveal shifts in policy impact, funding structures, or community engagement levels.\n\n", sep = "")
```

```
# Summarize average Revenue per Cardholder by Year and Population Group
insight2 <- datacombined2 %>%
  mutate(PopulationGroup = cut(PopulationResident, breaks = quantile(PopulationResident, probs = 0:3/3, na.rm = TRUE),
                               labels = c("Small", "Medium", "Large"), include.lowest = TRUE)) %>%
  group_by(Year, PopulationGroup) %>%
  summarise(AvgRevPerCardholder = mean(RevPerCardholder, na.rm = TRUE), .groups = "drop")

# Box plot visualization to show distribution
ggplot(datacombined2, aes(x = Year, y = RevPerCardholder,
                          fill = cut(PopulationResident, breaks = quantile(PopulationResident, probs = 0:3/3, na.rm = TRUE),
                                     labels = c("Small", "Medium", "Large"), include.lowest = TRUE))) +
  geom_boxplot() +
  labs(title = "Revenue per Cardholder Distribution by Year and Population",
       x = "Year", y = "Revenue per Cardholder ($)", fill = "Population Group") +
  ylim(0, 2500) +
  theme_minimal()
```

```
# ---------------------------------------------
# INSIGHT 3: HEATMAP (Top 10 Libraries only)
# Shows Avg Rev per Cardholder across Libraries & Years
# ---------------------------------------------

cat(" Insight 3: Top and Bottom Performing Libraries\n\n",
    " Purpose:\nTo identify which libraries consistently perform well or poorly in revenue efficiency to learn from their strategies or flag challenges.\n\n",
    "Interpretation:\nTop-performing libraries may benefit from higher community engagement, targeted programs, or efficient resource management.\n",
    "Lower-performing libraries could be constrained by budget, infrastructure, or population challenges.\n",
    "These benchmarks help set realistic targets for underperforming branches and recognize best practices from leaders.\n\n", sep = "")


# Filter top 10 libraries by overall average RevPerCardholder
top_libraries <- datacombined2 %>%
  group_by(Library) %>%
  summarise(OverallAvgRev = mean(RevPerCardholder, na.rm = TRUE)) %>%
  top_n(10, OverallAvgRev) %>%
  pull(Library)

# Filter main data
filtered_data <- datacombined2 %>%
  filter(Library %in% top_libraries)

# Group and reshape
heatmap_data <- filtered_data %>%
  group_by(Library, Year) %>%
  summarise(AvgRev = mean(RevPerCardholder, na.rm = TRUE)) %>%
  ungroup()

heatmap_matrix <- dcast(heatmap_data, Library ~ Year, value.var = "AvgRev")

# Prepare matrix
rownames(heatmap_matrix) <- heatmap_matrix$Library
heatmap_matrix <- heatmap_matrix[, -1]
heatmap_matrix[is.na(heatmap_matrix)] <- 0

# Plot heatmap
heatmap(as.matrix(heatmap_matrix),
        Rowv = NA, Colv = NA,
        col = colorRampPalette(c("white", "lightgreen", "darkgreen"))(50),
        scale = "column",
        margins = c(4, 5),
        main = "Top 10 Libraries: Revenue per Cardholder by Year")
```

```r
cat(" Insight 4: Relationship Between Total Circulation and Revenue per Cardholder\n\n",
    "Purpose:\nTo examine whether higher circulation volumes translate to higher revenue efficiency.\n\n",
    "Interpretation:\nIf there’s a moderate positive correlation, it suggests active usage supports greater value generation per user.\n",
    "A weak correlation might indicate that revenue efficiency is driven by factors beyond circulation (e.g., staffing, digital services).\n",
    "Understanding this can help libraries tailor programming and marketing to drive both usage and efficiency.\n\n", sep = "")


insight4 <- datacombined2 %>%
  summarise(Correlation_LocalGrant = cor(RevPerCardholder, LocalOperatingGrant, use = "complete.obs"),
            Correlation_Staffing = cor(RevPerCardholder, Staffingexpenditure, use = "complete.obs"))


print(insight4)
if (nrow(datacombined2 %>% filter(!is.na(LocalOperatingGrant) & !is.na(Staffingexpenditure))) > 0) {
  ggplot(datacombined2 %>% filter(!is.na(LocalOperatingGrant) & !is.na(Staffingexpenditure)),
         aes(x = LocalOperatingGrant, y = RevPerCardholder, size = Staffingexpenditure, color = Staffingexpenditure)) +
    geom_point(alpha = 0.5) +
    scale_size_continuous(range = c(1, 10)) +
    scale_color_gradient(low = "blue", high = "red") +
    labs(title = "Revenue per Cardholder vs. Local Operating Grant and Staffing Expenditure",
         x = "Local Operating Grant ($)", y = "Revenue per Cardholder ($)",
         size = "Staffing Expenditure ($)", color = "Staffing Expenditure ($)") +
    ylim(0, 2500) +
    theme_minimal()
}
```

```r
# Filter out invalid rows
datacombined2 <- datacombined2 %>%
  dplyr::filter(!is.na(RevPerCardholder) & is.finite(RevPerCardholder) & RevPerCardholder <= 250
0 &
                !is.na(TotalAnnualDirectCirculation))




# Create insight5 data frame
insight5 <- datacombined2 %>%
  group_by(Year) %>%
  summarise(
    AvgRevPerCardholder = mean(RevPerCardholder, na.rm = TRUE),
    TotalDonations = sum(Donations, na.rm = TRUE),
    TotalSelfgeneratedRevenue = sum(SelfgeneratedRevenue, na.rm = TRUE),
    .groups = "drop"
  )

# Print summary table
print(insight5)

# Visualize trends over years
ggplot(insight5, aes(x = Year)) +
  geom_line(aes(y = AvgRevPerCardholder, color = "Revenue per Cardholder", group = 1), linewidth
= 1) +
  geom_line(aes(y = TotalDonations / max(TotalDonations, na.rm = TRUE) * 2500, color = "Donation
s", group = 1), linewidth = 1) +
  geom_line(aes(y = TotalSelfgeneratedRevenue / max(TotalSelfgeneratedRevenue, na.rm = TRUE) * 2
500, color = "Self-Generated Revenue", group = 1), linewidth = 1) +
  labs(
    title = "Revenue per Cardholder vs. Donations and Self-Generated Revenue Over Years",
    x = "Year",
    y = "Revenue per Cardholder ($)",
    color = "Parameter"
  ) +
  ylim(0, 2500) +
  theme_minimal()
```