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: