Homework 2: Health Economics

Nikhita Gandhe

2025-02-18

Load necessary libraries

library(dplyr) library(ggplot2) library(readr) ## Question 1: Number of Hospitals Filing Multiple Reports ### How many hospitals filed more than one report in the same year? Show your answer as a line graph of the number of hospitals over time.

Load dataset

data_all <- readRDS('C:/Users/Nikhita Gandhe/Documents/GitHub/ECON-470-HW2/data/output/HCRIS_DATA.rds')

hcris_Data <- readRDS("C:/Users/Nikhita Gandhe/Documents/GitHub/ECON-470-HW2/data/output/HCRIS_DATA.rds") hcris_data <- as.data.frame(HCRIS.numeric) print(colnames(HCRIS.numeric))

Question 1: Number of Hospitals Filing Multiple Reports

How many hospitals filed more than one report in the same year? Show your answer as a line graph of the number of hospitals over time.

fig.dup <- dup.count %>% ggplot(aes(x = as.factor(fyear), y = duplicates, group = 1)) + geom_line(color = "blue", size = 1) + geom_point(color = "red", size = 2) + labs(x = "Year", y = "Number of Hospitals", title = "Number of Hospitals Filing More Than One Report Per Year") + theme_bw() + scale_y_continuous(limits = c(0, 300)) + theme(axis.text.x = element_text(angle = 70, hjust = 1))

Print the plot

print(fig.dup)

Question 2: Unique Hospital IDs

After removing/combining multiple reports, how many unique hospital IDs (Medicare provider numbers) exist in the data?

Remove duplicate reports by selecting unique combinations of provider_number and year

unique_hospitals <- final.hcris %>% distinct(provider_number, .keep_all = TRUE) # Keep only unique provider_number

Count the number of unique hospital IDs (Medicare provider numbers)

num_unique_hospitals <- unique_hospitals %>% summarize(total_unique_hospitals = n_distinct(provider_number))

Print the result

print(num_unique_hospitals)

Question 3: Distribution of Total Charges by Year

What is the distribution of total charges in each year? Show your results with a violin plot.

Question 3:

Load necessary libraries

library(ggplot2) library(dplyr)

Step 1: Calculate the IQR for each year and remove outliers

filtered_hcris <- final.hcris %>% filter(!is.na(tot_charges) & tot_charges > 0) %>% group_by(fyear) %>% mutate(Q1 = quantile(tot_charges, 0.25), Q3 = quantile(tot_charges, 0.75), IQR = Q3 - Q1, lower_bound = Q1 - 1.5 * IQR, upper_bound = Q3 + 1.5 * IQR) %>% filter(tot_charges >= lower_bound & tot_charges <= upper_bound) # Remove outliers

Step 2: Create the violin plot

violin_plot <- ggplot(filtered_hcris, aes(x = as.factor(fyear), y = tot_charges)) + geom_violin(fill = "lightblue", color = "darkblue") + scale_y_continuous(labels = scales::comma) + # Format y-axis with commas labs(title = "Distribution of Total Charges by Year (Outliers Removed)", x = "Year", y = "Total Charges") + theme_minimal() + theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotate x-axis labels for readability

Step 3: Display the plot

print(violin_plot)

Question 4: Distribution of Estimated Prices by Year

What is the distribution of estimated prices in each year? Show your results with a violin plot.

Load necessary libraries

library(dplyr) library(ggplot2)

Step 1: Calculate the discount factor

final.hcris <- final.hcris %>% mutate(discount_factor = 1 - (tot_discounts / tot_charges))

Step 2: Calculate price numerator and denominator

final.hcris <- final.hcris %>% mutate(price_num = (ip_charges + icu_charges + ancillary_charges) * discount_factor - tot_mcare_payment, price_denom = tot_discharges - mcare_discharges, price = price_num / price_denom)

Step 3: Remove rows with negative or missing prices

final.hcris_clean <- final.hcris %>% filter(!is.na(price) & price > 0)

Step 4: Remove outliers using IQR method for each year

final.hcris_filtered <- final.hcris_clean %>% group_by(fyear) %>% mutate(Q1 = quantile(price, 0.25), Q3 = quantile(price, 0.75), IQR = Q3 - Q1, lower_bound = Q1 - 1.5 * IQR, upper_bound = Q3 + 1.5 * IQR) %>% filter(price >= lower_bound & price <= upper_bound) # Remove outliers

Step 5: Create the violin plot

violin_plot_prices <- ggplot(final.hcris_filtered, aes(x = as.factor(fyear), y = price)) + geom_violin(fill = "lightgreen", color = "darkgreen") + scale_y_continuous(labels = scales::comma) + # Format y-axis with commas labs(title = "Distribution of Estimated Prices by Year (Outliers Removed)", x = "Year", y = "Estimated Price") + theme_minimal() + theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotate x-axis labels for readability

Step 6: Display the plot

print(violin_plot_prices)