

University of Greenwich ID Number: 001312184

FPT Student ID Number: GCD220076

Module Code: COMP1810

Module Assessment Title: Data and Web Analytics

Lecturer Name: Trần Trọng Minh

Submission Date: 13/08/2024

TASK A

This section will work primarily with e-shop clothing 2008 and shop clothing infor 2008 files with the aim of analyzing the provided datasets using various statistical, visualization and data manipulation techniques in R. The tasks involve extracting relevant data, performing sales and revenue analysis, conducting descriptive statistics and working with the datasets

A screenshot of a computer program

Description automatically generated

The code loads necessary libraries, imports data from an Excel file, and processes it by splitting the first column into multiple columns using semicolons. It then removes unwanted columns, renames others for clarity, and adds a new column to indicate if a product was sold based on the highest click stream value within each session. And it displays the first few rows of the modified data and opens it in a viewer for inspection.  
  
As a result, we will have a file with separated columns and format like this  
  
A screenshot of a computer

Description automatically generated

After that, we will visualize the data to understand the correlations between them and the relationships between different fields.  
  
A screenshot of a computer code

Description automatically generated  
  
The code converts all columns in the dataframe to numeric values, calculates the correlation matrix to show the relationships between variables, and then reshapes the matrix into a long format for plotting. Using ggplot2, it creates a heatmap of the correlation matrix, where each tile represents the correlation between variables, colored from purple (low correlation) to green (high correlation). The plot includes a color bar for correlation values, a title, rotated x-axis labels for readability, and text labels showing the exact correlation values.  
  
A green and purple squares with black text

Description automatically generated

The correlation matrix provides insights into the relationships between different variables:

1. **Colour, Month, Day, Location**: These variables exhibit very weak or no correlation with other variables, suggesting that they may be relatively independent or not significantly related to numerical values such as sales, clicks, or prices.
2. **Strong Negative Correlation between Price and Price 2 (-0.74)**: This strong negative correlation indicates that as the primary price (price) increases, the secondary price (price 2) tends to decrease. This could imply a discounting strategy where one price reflects a promotional or reduced rate.
3. **Moderate Negative Correlation between Page 1 (Main Category) and Price (-0.36)**: This moderate negative correlation suggests that certain product categories are associated with lower prices. This may reflect different pricing strategies across various product lines, where specific categories may be priced lower, possibly to appeal to a broader customer base or to meet competitive pricing in the market.

TASK B

The modified code creates a bar chart to visualize total sales by month. It first summarizes the sales data by grouping it by month and calculating the total sales for each month. The ggplot function is then used to generate the bar chart, with geom\_bar(stat = "identity") creating bars where the height directly represents the total sales values. The bars are colored dark orange (fill = "darkorange"), and the chart is labeled with appropriate titles for clarity. The theme\_minimal() function is applied to give the chart a clean, simple look.

A screenshot of a graph

Description automatically generated

In the table of Total Sales by month, April is the month with the highest sales, which could be due to increased customer demand during the summer season, with a total of 7,000 products sold. People have more time to relax and shop. From May to July, sales remain stable, fluctuating around 5,000. August has the lowest sales, with a total of nearly 2,000 products sold. This may be because, at the end of the summer break, customers are preparing for a busy work season, so their shopping time is limited, leading to a significant decrease in sales.

Next comes the total sale by location section, here is the code to compare revenue based on each different location  
  
A screenshot of a computer program

Description automatically generated

And when visualized, we will get a bar chart graph showing total sales based on local  
  
A graph of a graph

Description automatically generated

Based on the graph, we can see that the left area of ​​the graph has a higher total number of products than the center and right areas. Location 2 has the highest total product sales with more than 5000 products sold, followed by location 1 which is about 500 products less than location 2 and the total products sold in location 1 is about 4800 to 4900. The total number of products sold from the center area 3 increases gradually to Location 5. In which Location 3 is 3000 lowest equal to location 6, location 4 is about 3900 products and location 5 is 4200 to 4300 products.

Below is a chart showing the number of products based on color:

A screenshot of a computer code

Description automatically generated

Here is the chart:  
  
A green bar graph with white text

Description automatically generated

Based on the chart, we see that the number of products is shown to increase gradually with color 2 being the most sold color and color 5 being the least sold and the difference is very large with color 2 being around 4500 and color 5 being 200-300 products. With the product having color 3 having a sales volume of around 4000, we see that customers really like color 3 and 2 and the product having color 5 8 is underrated with a total sales volume of less than 1000.

Here is the monthly product sales by category chart:  
  
A computer code with text

Description automatically generated with medium confidence

The code summarizes total products sold by category and month, then creates a line plot to visualize these sales trends. It groups the data by month and product category, calculates the total products sold, and plots this information with lines connecting the points and different colors for each category. The plot includes a title, labeled axes, a color legend for product categories, and rotated x-axis labels for better readability.

A graph with colored dots

Description automatically generated

Based on the graph, we observe that product sales are declining over time. Product Category 1 initially met user demand well, with sales nearing 2000 units, but then experienced a sharp decline. Product Category 2, while not decreasing as quickly as Product 1, showed a steady decline from April to July, with a significant drop in August. Products 3 and 4, despite being popular in April, also suffered a significant drop in sales.

Next is the chart showing sales by price:

A computer screen shot of a program

Description automatically generated

A graph of a product sold by price

Description automatically generated

The chart indicates that most users are willing to spend between 28 and 48, with total product sales rising from 2500 to a peak of 5000 at a price of 38, and then gradually decreasing to around 2800 at a price of 48. Additionally, some higher-priced products, such as those priced at 57 and 62, still attract significant purchases, with a total of 2000 units sold. This suggests that while the ideal price range for products is between 28 and 48, indicating a strong value perception, higher-priced items can also be popular if they offer perceived value or benefits.  
  
Discussion:  
The total sales graph reflects various key insights from the data tables. Location 2 stands out with the highest sales, exceeding 5,000 products, significantly impacting the total sales graph with a prominent peak. Location 1 follows closely, contributing substantially with around 4,800 to 4,900 products sold. Sales from Locations 3 to 5 show a gradual increase, with Location 5 achieving the highest in this range, while Location 6’s lower sales dampen the overall performance. Color preferences also play a significant role: Color 2 leads with around 4,500 products sold, while Color 5 lags behind with only 200-300 products. This disparity in color sales affects the total sales graph, with peaks driven by Color 2 and dips corresponding to less popular colors. Additionally, sales trends by product category show an initial high for Category 1, which then sharply declines, and a steady decrease for Category 2. Categories 3 and 4 also experience significant drops in sales. These declining trends are visible as downward slopes on the graph. Lastly, the chart highlights a peak in sales at a price of 38, with a total of 5,000 units sold, while higher prices show decreased sales. Despite the overall decline at higher price points, the graph reflects continued significant sales for premium-priced products, indicating varied consumer behavior across price ranges.

TASK C

A screenshot of a computer program

Description automatically generated

The code first ensures that the price and product\_sold columns in the dataframe are numeric and then calculates the Revenue for each record by multiplying these columns. It then summarizes the total revenue for each month by grouping the data by month and summing up the Revenue. Finally, it creates a bar chart of the total monthly revenue using ggplot2, with bars filled in dark red, a title, labeled axes, and rotated x-axis labels for better readability.

A graph of a graph

Description automatically generated with medium confidence

Below is the revenue chart from April to August for the store. Revenue in April is the highest of the five months, while August has the lowest. Revenue in May and July are nearly identical, and June's revenue is lower than both May and July. The high revenue in April may be due to new products attracting a wide range of customers, whereas the lower revenue in subsequent months could be because older products lost their appeal, and new products did not meet the needs of most customers.

TASK D

A computer screen shot of a code

Description automatically generated

The code reads data from an Excel file into a dataframe named dframe, then splits the first column into six new columns (id, id2, name, salary, start\_date, dept) using whitespace as the separator. It removes the unnecessary id2 column and converts the salary column to numeric type. Finally, it displays the transformed dataframe in a viewer, though the correct name for the dataframe in the View function should be dframe.

And this is a result after transformed data:  
  
A screenshot of a computer

Description automatically generated

Below is the code to find the mean, mode, standard deviation and variance of salary column:A screenshot of a computer program

Description automatically generated

The salary data for the company yields the following statistics:

* **Mean Salary:** 656.88
* **Median Salary:** 628.05
* **Mode Salary:** 515.20
* **Standard Deviation of Salary:** 103.06
* **Variance of Salary:** 10621.25

These metrics provide a comprehensive view of the salary distribution within the company, indicating the average, central tendency, and dispersion of salaries.

To gain deeper insights into salary issues, we can examine the salary density plot:  
A screenshot of a computer screen

Description automatically generated

This code calculates key statistical measures for salary data—mean, median, mode, and standard deviation—and visualizes them on a density plot. The density plot shows the distribution of salaries, with vertical lines added to represent each calculated statistic. A red dashed line marks the mean salary, a green dotted line shows the median, an orange solid line indicates the mode, and purple dot-dashed lines highlight one standard deviation above and below the mean. This visualization allows for easy comparison of these central tendency measures and the overall salary distribution.

A graph of distribution density

Description automatically generated

 **Salary Distribution**: The curve represents the density of salary values, showing how common different salary ranges are. The highest peak of the curve indicates the salary range where most employees are concentrated.

 **Mean Salary (Red dashed line)**: The mean salary is marked by the red dashed line. It falls towards the higher end of the distribution, indicating that the average salary is somewhat higher than the most common salaries (the mode).

 **Median Salary (Green dotted line)**: The median salary, represented by the green dotted line, lies slightly to the left of the mean. This suggests that the distribution is right-skewed, meaning that there are more lower salary values, but a few high salaries are pulling the mean upward.

 **Mode Salary (Orange solid line)**: The mode is represented by the orange solid line, located at the peak of the distribution curve. This is the most frequently occurring salary value, and it is lower than both the median and the mean, reinforcing the skewness observed.

 **Standard Deviation (Purple dot-dashed lines)**: The purple dot-dashed lines mark one standard deviation above and below the mean. These lines enclose the central portion of the data, showing where the bulk of salary values lie. The placement of these lines indicates the range within which most employees' salaries fall.

The right-skewed distribution indicates that while most salaries are clustered around a certain value, there are some higher salaries that pull the average up. The fact that the mode is lower than the median and mean suggests that more employees earn less than the average, with fewer employees earning significantly more, contributing to the skew. Additionally, the spread of the standard deviation shows that although most employees' salaries are close to the average, there is noticeable variability in pay within the company.

To explore salaries across different departments in the company, we can visualize this using a scatter plot. The scatter plot will show how salaries are distributed among various departments, helping to identify patterns or discrepancies:  
A screenshot of a computer screen

Description automatically generated

The code converts the dept column to a factor to ensure it's treated as a categorical variable and then creates a scatter plot to visualize the relationship between department and salary. Points are plotted in red with some horizontal jitter to prevent overlap, and the plot includes a title and axis labels. The minimal theme is used for a clean look, and x-axis labels are rotated 45 degrees for improved readability.  
  
The graph highlights salary variations across departments. It shows that the Finance department generally offers higher salaries compared to others. In contrast, the IT department exhibits a broader salary range, with some employees earning significantly less and others earning comparable salaries to those in Finance. The Operations department has a more limited salary range, with most salaries clustered around the mid-level.

Task E

A computer screen shot of a code

Description automatically generated

The code replaces "none" values in the starwars dataset with NA to handle missing data and then displays the cleaned dataframe.

Và nó sẽ làm file trở nên như thế này  
  
A screenshot of a computer

Description automatically generated

The code filters the star\_wars\_df dataframe to include only those rows where eye\_color is not "black" and height is greater than 150. This results in a subset of the data that meets both criteria.

A black and white text

Description automatically generated

And this is the result of that code  
  
A screenshot of a computer

Description automatically generated

And here is the visualization code to create the graph after filtering

The code uses the ggplot2 and dplyr libraries to filter and visualize data from the star\_wars\_df dataframe. It first filters the data to exclude rows where eye\_color is "black" and height is 150 cm or less. Then, it creates a histogram of the heights, grouping and filling the bars by eye\_color. The histogram bins are set to 10 cm in width and are positioned side by side for each eye color. The plot is titled "Distribution of Heights by Eye Color," with axes labeled "Height (cm)" and "Count," and uses a minimal theme for a clean appearance. Note that the comment refers to creating a scatter plot, but the code actually generates a histogram.

A screen shot of a graph

Description automatically generated

Looking at the chart, we see that brown eye color is the most common eye color with a count of 8, followed by blue and yellow with an equal count of 6.

E2

A screenshot of a computer code

Description automatically generated

Here is how to calculate BMI and round to 2 decimal places

And after calculating BMI, the data table will have a bmi column like this  
  
A screenshot of a computer

Description automatically generated

Here is plot a graph of height against BMI, and provide an explanation for your graph

A screenshot of a computer

Description automatically generated

The bubble plot visualizes the relationship between height and BMI for characters in the star\_wars\_df dataframe, with bubble size representing mass. The X-axis shows height in centimeters, and the Y-axis displays BMI. Larger bubbles indicate higher mass, while smaller bubbles denote lower mass, with semi-transparent blue bubbles allowing overlap visibility. The plot reveals how BMI varies with height, potentially indicating trends where taller characters may have higher or lower BMI. Large bubbles at various heights and BMIs help illustrate how mass influences BMI, while smaller bubbles suggest a diverse mass range among characters. The plot also highlights clusters of characters with similar height, BMI, and mass, and outliers with atypical mass values, offering insights into unique or exceptional cases.