Metaversalize

www.meta.reapress.com

Metaverse. Vol. 1, No. 4 (2024) 225–239.

Paper Type: Original Article

Advanced Visualization Approaches for Statistical Data Analysis

Md Wahiduzzaman Suva[1] , Md. Imtiaj Alam Sajin[1], Esm E Moula Chowdhury Abha[1], Mushfiqur Rahman Abir[1], Asif Zaman[1,*]

[1] Department of Computer Science, American International University-Bangladesh, Dhaka, Bangladesh; wahedshuvo36@gmail.com; imtiajsajin@gmail.com; esmechowdhuryabha@gmail.com; mushfiqurrohomanabir@gmail.com; zasif4805@gmail.com.

Citation:

Received: 02 May 2024 *Revised: 24 July 2024* *Accepted: 11 September 2024*	Suva, Md. W., Alam Sajin, Md. I., Chowdhury Abha, Esm. E. M., Abir, M. R., & Zaman, A. (2024). Advanced visualization approaches for statistical data analysis. *Metaversalize*, *1*(4), 225-239.

Abstract

In the era of big data, effective data visualization plays a crucial role in presenting and understanding complex datasets. This work investigates sophisticated visualization methods that help researchers spot patterns, comprehend complex relationships within data, and effectively convey findings. R offers several benefits for statistical research and data processing. Many researchers hesitate to use it because of its perceived coding complexity and dependence on proprietary software. This study demonstrates R's powerful data visualization features and uses a dataset named HCV data from the UCI machine learning repository that includes a range of laboratory and demographic characteristics. This study aims to address the challenges associated with data visualization.

Keywords: Data visualisation, R language, Ggplot2, Smplot, Visreg.

1 | Introduction

Data is crucial in businesses, industries, research, and technological development, with the big data era accelerating. Data visualization aids in understanding the significance of data by summarizing and presenting it in an accessible, understandable, and straightforward format [1], [2]. The utilization of the programming language R for data processing and statistical analysis has seen a remarkable surge among researchers in recent years, apparent from the R Core Team's significant increase in citations. This increase highlights the understanding of R's benefits, including its role in enhancing reproducibility, transparency, and offering a broad spectrum of customizable data visualization options. Data visualization in the R programming language

Corresponding Author: zasif4805@gmail.com

https://doi.org/10.22105/metaverse.v1i1.36

is a critical skill across various disciplines, including science, finance, and journalism, due to its ability to graphically represent data and effectively communicate key findings [3], [4].

Despite the obvious benefits, many academics are hesitant to use R because of the perceived complexity of coding skill acquisition. Furthermore, the widespread reliance on proprietary software in academic programs adds to the reluctance to use R. However, it is important to note that, even within these systems, parts of computational thought, such as coding, are frequently used, albeit unnoticed by many researchers. Our lesson seeks to dispel these myths by showing how researchers, regardless of strong coding ability, may use R's powerful data visualization features.

The source of the dataset used for this paper is taken from UCI machine learning repository for data, a renowned repository known for its diverse collection of datasets spanning various domains. Leveraging datasets from such repositories ensures access to high-quality data and promotes reproducibility and research transparency. By utilizing openly available datasets, researchers can facilitate collaboration, validation, and comparison of findings across studies, thereby contributing to the advancement of scientific knowledge. Additionally, the use of publicly accessible datasets aligns with the principles of open science, promoting the sharing of resources and fostering a culture of collaboration within the research community. Thus, by employing datasets from reputable sources like the UCI repository, this study underscores the importance of transparent data sourcing and promotes best practices for reproducible research in the field of data analysis and visualization.

Regardless of existing R knowledge, researchers will develop critical skills in using its visualization tools to improve reproducibility, transparency, and, ultimately, the quality of their research results. Through practical demos and hands-on tutorials, we aim to educate researchers to traverse the complexities of data visualization in R, promoting a culture of informed decision-making and impactful research results. This study improves repeatability and transparency and provides researchers with critical tools for making informed decisions about diabetes care and prevention initiatives.

2 | Literature Review

Data visualization is pivotal in understanding complex relationships within datasets, aiding in feature selection, model interpretation, and decision-making processes. Several studies have explored various visualization techniques to enhance the accuracy and interpretability of data-driven models.

Zoghi and Serpen [5] delved into the issues of class imbalance and overlap in the UNSW-NB15 dataset, highlighting their detrimental impact on classification performance. Traditional visualization methods are found lacking in capturing overlap, prompting the proposal of a novel visualization approach that effectively detects and illustrates overlap, aiding in evaluating data scalers' efficacy.

Kucukler et al. [6] introduced an openly accessible EEG dataset capturing user responses to energy data visualizations. It incorporates empirical and synthetic EEG data, providing a resource for researchers in diverse fields such as computer science, energy conservation, and human-computer interaction to explore user behavior and improve energy consumption strategies.

Pilhöfer and Unwin [7] introduced innovative visualization methods for categorical data. Their techniques, the rmb plot, and CPCP plot offer structured representations and support exploratory analysis.

These approaches address a gap in categorical data visualization, contributing to improved understanding and interpretation of complex datasets. The CPCP plot may not be as effective with high-dimensional data due to visual clutter. It becomes difficult to interpret patterns, and identifying relationships between variables can be challenging.

Nordmann et al. [8] provided a tutorial on data visualization using R aimed at researchers unfamiliar with the language. They highlight R's advantages, such as its open-source nature, which offers customizable visualization options. The tutorial introduces ggplot2 and demonstrates replicating common plots and extending them to less available options. This work aims to empower researchers to utilize R's visualization tools for enhanced transparency and insights in data analysis. Although ggplot2 offers effective visualization methods, it has a steep learning curve, particularly for beginners. Understanding the layering of plot components can be challenging. Another drawback of ggplot2 is that it can be slow with huge datasets or highly detailed plots, with performance degrading as data complexity increases.

Min and Zhou [9] introduced smplot, an R package that simplifies data visualization. They address the steep learning curve associated with R and the time-consuming process of using multiple software programs for plot refinement. The smplot offers visually pleasing default plots, including bar graphs, violin plots, and correlation plots, while adhering to best practices in data visualization. Its modular design allows for further customization, making it accessible to researchers with basic R knowledge. With comprehensive documentation and examples, the smplot presents an intuitive solution for elegant data visualization in R. However, it has fewer visualization options for qualitative data, making visualization of categorical data more challenging.

The paper titled "big data analysis with interactive visualization using R packages" emphasizes the need for advanced visualization techniques for big data [10]. Traditional methods fall short of big data, prompting the development of technologies for massive data processing, such as Hadoop. The authors highlighted the limitations in traditional analytic systems due to the sheer volume of data and the need for more static visualizations. They compare interactive web packages with R visualization packages, proposing an integrated web-based analysis environment. They implemented an interactive environment using Rook, Shiny, GoogleVIS, and RgoogleMaps packages for handling GIS-related big data, showcasing an innovative approach for big data analysis with interactive visualization.

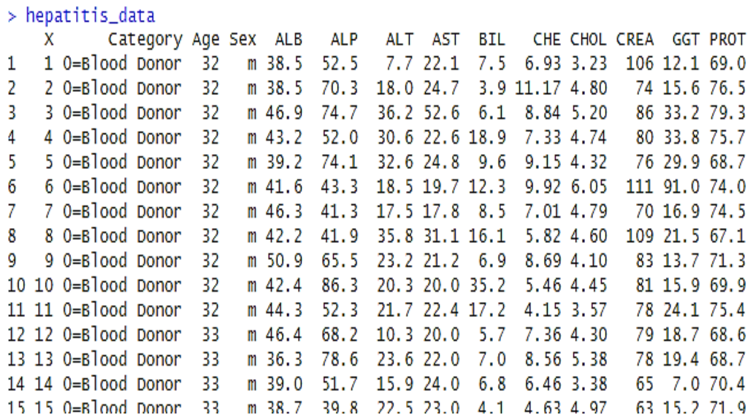
Breheny and Burchett [11] introduce visreg, an R package designed to simplify the visualization of regression models through easy function calls. Traditional methods like plotting regression lines with confidence bands are straightforward for simple models but inadequate for complex multiple regression models with interactions and non-linear relationships. Using R's generic functions, Visreg automates the visualization process for various models, including linear, generalized linear, robust regression, generalized additive, and proportional hazards models. This enables users to generate clear, informative plots with minimal effort. Nonetheless, the visreg package works only on regression models but cannot handle other types of statistical models, such as classification or clustering models.

The paper titled "Kaleidomaps: a new technique for the visualization of multivariate time-series data" introduced Kaleidomaps, designed to improve understanding of complex multivariate time-series data. Kaleidomaps enhance classic cascade plots using line curvature to detect periodic patterns with a design that aids pattern perception. The paper demonstrates its effectiveness with case studies and links it to traditional line graphs and signal processing methods. However, limitations include the need for further research to evaluate scalability, address interpretation complexity, and develop accessible implementation tools [12].

3|Methodology

In this study, we obtained the dataset from the UCI machine learning repository [13]. It comprises laboratory values of blood donors and hepatitis C patients along with demographic variables like age and gender. The dataset contained missing values. To handle the missing values, we initially checked the dataset for missing values, particularly in columns such as Albumin (ALB), Alkaline Phosphatase (ALP), Aspartate Aminotransferase (AST), and Cholesterol (CHOL). Missing values were imputed using established methods such as mean and median imputation to ensure the integrity and completeness of the dataset.

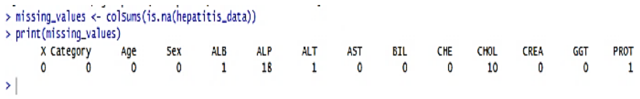
Afterward, the dataset was structured, ensuring variables were appropriately formatted for analysis. Categorical variables were converted to factors, and numeric variables were standardized if necessary to facilitate comparisons.



```
> hepatitis_data
    X      Category Age Sex  ALB   ALP   ALT  AST  BIL   CHE CHOL CREA  GGT PROT
1   1 0=Blood Donor  32   m 38.5  52.5   7.7 22.1  7.5  6.93 3.23  106 12.1 69.0
2   2 0=Blood Donor  32   m 38.5  70.3  18.0 24.7  3.9 11.17 4.80   74 15.6 76.5
3   3 0=Blood Donor  32   m 46.9  74.7  36.2 52.6  6.1  8.84 5.20   86 33.2 79.3
4   4 0=Blood Donor  32   m 43.2  52.0  30.6 22.6 18.9  7.33 4.74   80 33.8 75.7
5   5 0=Blood Donor  32   m 39.2  74.1  32.6 24.8  9.6  9.15 4.32   76 29.9 68.7
6   6 0=Blood Donor  32   m 41.6  43.3  18.5 19.7 12.3  9.92 6.05  111 91.0 74.0
7   7 0=Blood Donor  32   m 46.3  41.3  17.5 17.8  8.5  7.01 4.79   70 16.9 74.5
8   8 0=Blood Donor  32   m 42.2  41.9  35.8 31.1 16.1  5.82 4.60  109 21.5 67.1
9   9 0=Blood Donor  32   m 50.9  65.5  23.2 21.2  6.9  8.69 4.10   83 13.7 71.3
10 10 0=Blood Donor  32   m 42.4  86.3  20.3 20.0 35.2  5.46 4.45   81 15.9 69.9
11 11 0=Blood Donor  32   m 44.3  52.3  21.7 22.4 17.2  4.15 3.57   78 24.1 75.4
12 12 0=Blood Donor  33   m 46.4  68.2  10.3 20.0  5.7  7.36 4.30   79 18.7 68.6
13 13 0=Blood Donor  33   m 36.3  78.6  23.6 22.0  7.0  8.56 5.38   78 19.4 68.7
14 14 0=Blood Donor  33   m 39.0  51.7  15.9 24.0  6.8  6.46 3.38   65  7.0 70.4
15 15 0=Blood Donor  33   m 38.7  39.8  22.5 23.0  4.1  4.63 4.97   63 15.2 71.9
```

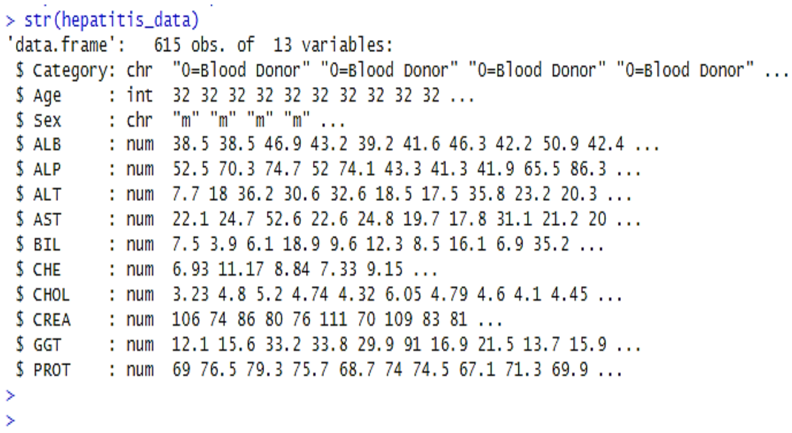
Fig. 1. Numerical visualization of the dataset.

The UCI machine learning repository is a reputable source known for its comprehensive and diverse datasets across various domains. While datasets from such repositories are generally reliable, it is not uncommon to encounter missing values in specific columns, such as ALB, ALP, AST, and CHOL. To address this issue and ensure the integrity of the data, missing values were handled using established methods such as imputation based on mean and median values. By employing these techniques, the dataset was augmented with filled values, maintaining the dataset's completeness for subsequent analysis. In the *Fig. 3*, the processed dataset is shown.



```
> missing_values <- colSums(is.na(hepatitis_data))
> print(missing_values)
       X Category      Age      Sex      ALB      ALP      ALT      AST      BIL      CHE     CHOL     CREA      GGT     PROT 
       0        0        0        0        1       18        1        0        0        0       10        0        0        1 
> |
```

Fig. 2. Visualization of the missing values count.



```
> str(hepatitis_data)
'data.frame':	615 obs. of  13 variables:
 $ Category: chr  "0=Blood Donor" "0=Blood Donor" "0=Blood Donor" "0=Blood Donor" ...
 $ Age     : int  32 32 32 32 32 32 32 32 32 32 ...
 $ Sex     : chr  "m" "m" "m" "m" ...
 $ ALB     : num  38.5 38.5 46.9 43.2 39.2 41.6 46.3 42.2 50.9 42.4 ...
 $ ALP     : num  52.5 70.3 74.7 52 74.1 43.3 41.3 41.9 65.5 86.3 ...
 $ ALT     : num  7.7 18 36.2 30.6 32.6 18.5 17.5 35.8 23.2 20.3 ...
 $ AST     : num  22.1 24.7 52.6 22.6 24.8 19.7 17.8 31.1 21.2 20 ...
 $ BIL     : num  7.5 3.9 6.1 18.9 9.6 12.3 8.5 16.1 6.9 35.2 ...
 $ CHE     : num  6.93 11.17 8.84 7.33 9.15 ...
 $ CHOL    : num  3.23 4.8 5.2 4.74 4.32 6.05 4.79 4.6 4.1 4.45 ...
 $ CREA    : num  106 74 86 80 76 111 70 109 83 81 ...
 $ GGT     : num  12.1 15.6 33.2 33.8 29.9 91 16.9 21.5 13.7 15.9 ...
 $ PROT    : num  69 76.5 79.3 75.7 68.7 74 74.5 67.1 71.3 69.9 ...
>
>
```

Fig. 3. Visualization of the processed dataset.

After the data preparation, various visualization techniques were employed for exploratory data analysis. This included Scatter Matrix, Hexbin, Corrplot, Bubble Chart, Heat Map, Slope Plot, Cross Sectional, Contour Plot. Also 3D Scatter charts for three-dimensional data representation, mosaic plots for visualizing joint distributions of categorical variables, and ridge plots for visualizing density distributions across categories were applied. Below all the applied techniques are listed.

Table 1. Visualization technique for univariate analysis.

Data Type	Visualization Tool
Quantitative	ECDF plot
Qualitative	Pie chart

Table 2. Visualization technique for multivariate analysis.

Data Type	Visualization Tool
Quantitative/Quantitative	Scatter matrix, Hexbin, Corrplot, Bubble chart, Heat map, Slope plot, Cross sectional, Contour plot
Quantitative v/s Qualitative	Box plot, 3D scatter, Density plot, Ridge plot, Violin plot
Qualitative v/s Qualitative	CPCP, Alluvial

4 | Results

Here, the function 'pairs (numerical_vars)' is used to create a matrix of scatterplots. This allows for a quick visual inspection of the relationships between different pairs of numeric variables (e.g., Age, ALB, ALP, etc.). All these numeric attributes create scatter plots with other numeric attributes, representing the correlation between them and any potential patterns or outliers [1]. Some scatter plots may reveal strong linear relationships between variables, such as the Cholinesterase (CHE) and CHOL scatter plots, indicating a possible correlation between these liver function markers. Other scatterplots, such as those comparing Bilirubin (BIL) with variables like ALB or Age, might need a clearer relationship.

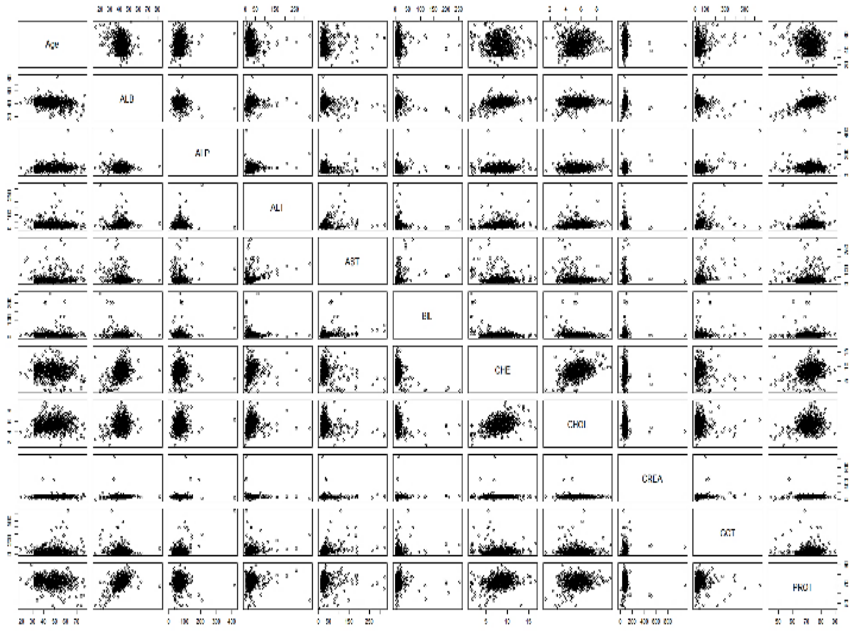


Fig. 4. Hepatitis data displayed in scatter matrix.

```
numerical_vars <- hepatitis_data[, sapply(hepatitis_data, is.numeric)]
pairs(numerical_vars)
```

Hexagonal binning divides the plot area into hexagonal bins, reflecting data density through color intensity. These plots are useful for high-density bivariate data, where conventional scatter plots may fall short [14]. In *Fig. 5*, the hexbin plot shows the relationship between Age and ALB (ALB levels) for hepatitis patients. Ages range from 20 to 75 years, and ALB levels from 20 to 80. A high concentration of patients aged 40 to 60 with ALB levels between 35 and 55 is indicated by yellow hexagons, while sparse hexagons at the edges highlight outliers.

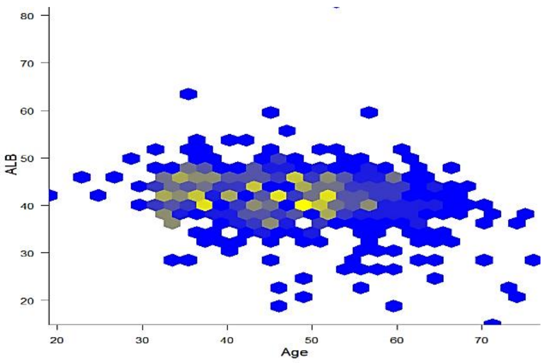


Fig. 5. Hepatitis data displayed in Hexagonal bin plots.

```
plot(hexbin(hepatitis_data$Age ,hepatitis_data$ALB), main="" ,
colramp=colorRampPalette(c("blue", "yellow")) ,  legend=F )
```

A corrplot helps visualize multivariate data, as it represents the correlation matrix of variables, with each cell in the matrix displaying the correlation coefficient between two variables [15]. In *Fig. 6*, the correlation coefficients are color-coded and sized to indicate the strength of the relationships. For instance, ALB shows a strong positive correlation with Protein (PROT), and a strong negative correlation with BIL, indicated by large blue and red circles, respectively.

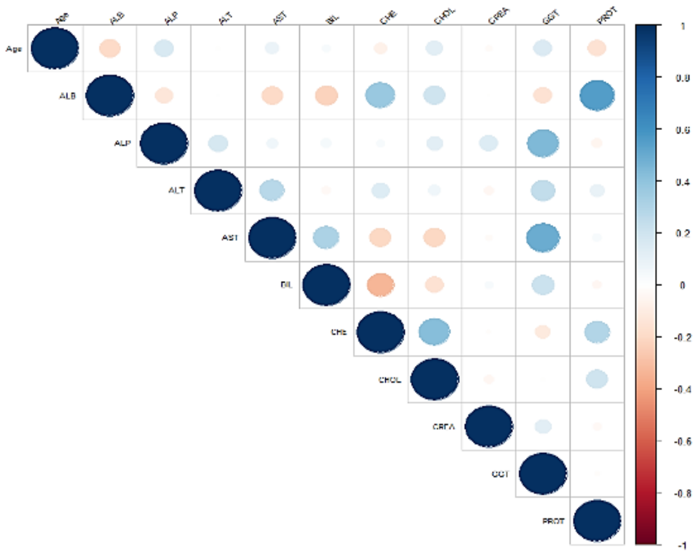


Fig. 6. Hepatitis data displayed in corrplot.

```
correlation_matrix<-cor(hepatitis_data[, sapply(hepatitis_data, is.numeric)])
corrplot(correlation_matrix, method = "circle", type = "upper", tl.col = "black", tl.srt = 45, tl.cex = 0.7)
```

Bubble charts are effective for visualizing three dimensions of data: two numeric variables plotted on the x and y axes, and a third numeric variable represented by the size of the bubbles [16]. *Fig. 7* shows a bubble chart of the hepatitis dataset with Age (x-axis), ALB (ALB levels in g/L on the y-axis), and BIL (BIL levels by bubble size). Patients' ages range from 20 to 60 years, mostly with ALB levels between 35 to 50 g/L. The varying bubble sizes indicate wide differences in BIL levels across different ages and ALB levels.

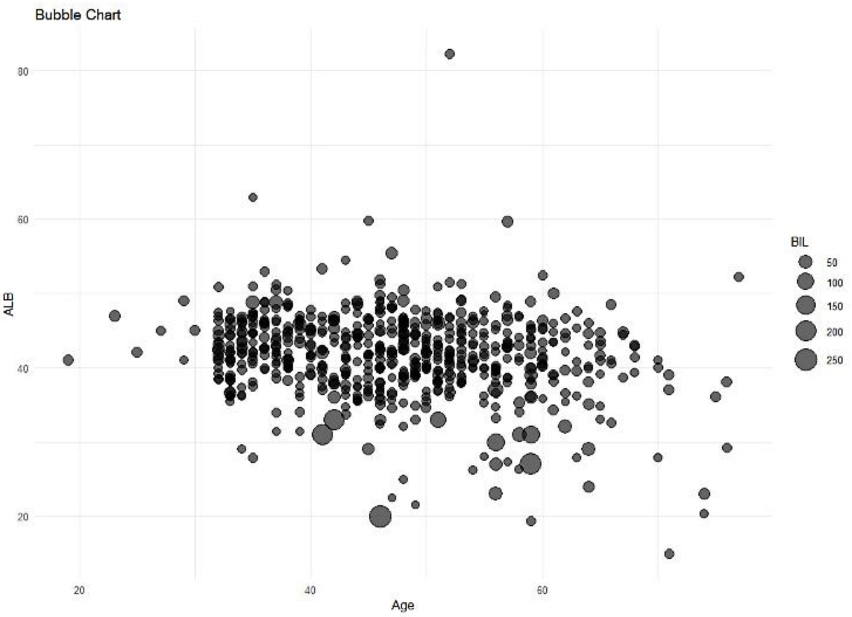


Fig. 7. 3D data visualization using Bubble chart.

```
ggplot(hepatitis_data, aes(x = Age, y = ALB, size = BIL)) + geom_point(alpha = 0.6) +
scale_size(range = c(3, 10)) +  labs(title = "Bubble Chart", x = "Age", y = "ALB", size = "BIL") +
theme_minimal()
```

Mosaic plots represent the joint distribution of two categorical variables by dividing a rectangle into smaller rectangles. *Fig. 8* shows a mosaic plot visualizing the joint distribution of Category and Sex in the hepatitis dataset. The x-axis represents the different categories (Blood Donor, Suspect Blood Donor, Hepatitis, Fibrosis, Cirrhosis), while the y-axis represents Sex (f for female, m for male).

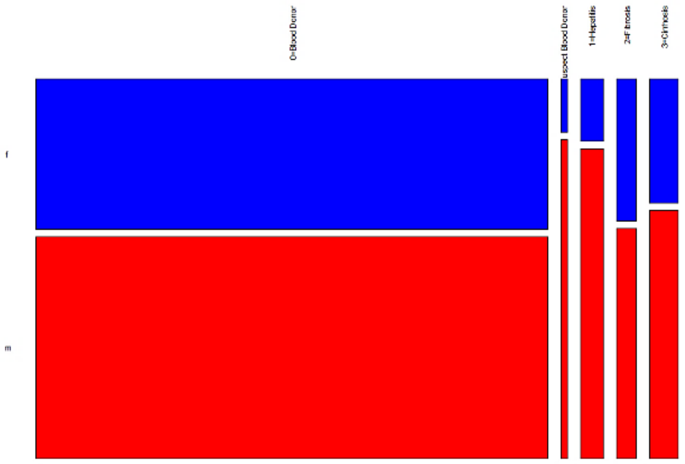


Fig. 8. Joint distribution by mosaicplot.

```
mosaicplot(table(hepatitis_data$Category,
hepatitis_data$Sex), col = c("blue", "red"), las=2)
```

The intensity of the color represents the strength of the correlation. Darker shades of blue or red indicate a stronger correlation, while lighter shades indicate a weaker correlation. Here red indicates positive correlation and blue indicates negative correlation. From *Fig. 9*, we can see that except for their own attributes, there is a stronger correlation between ALB - PROT, ALB - CHE, CHE - CHOL, ALP - GGT, and AST - GGT than other attributes.

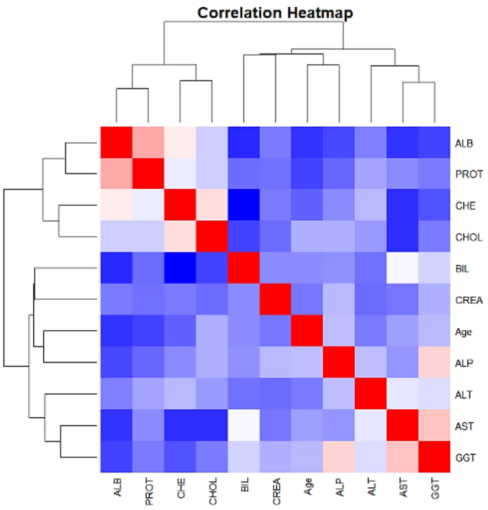


Fig. 9. Correlation shown by color intensity using heatmap.

numeric_data <- hepatitis_data[, sapply(hepatitis_data, is.numeric)] correlation_matrix <- cor(numeric_data)

heatmap(correlation_matrix, col = colorRampPalette(c("blue", "white", "red"))(100), symm = TRUE, main = "Correlation Heatmap")

The box plot summarizes the distribution of Alanine Aminotransferase (ALT) levels across different categories. Each box represents the Interquartile Range (IQR) of ALT values within a specific category, with the median displayed as a horizontal line. Whiskers extend to indicate the range of ALT values, excluding outliers. From *Fig. 10*, we can see that the outliers exist on blood donor and cirrhosis categories. Besides, the IQR values of suspect blood donors and fibrosis are larger than others.

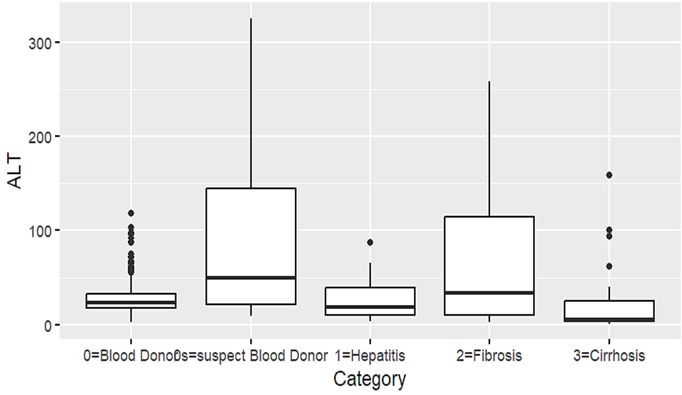


Fig. 10. Summarized ALT categorical distribution by boxplot.

ggplot(hepatitis_data, aes(x = Category, y = ALT)) +

geom_boxplot() +

labs(title = "Distribution of ALT by Category", x = "Category", y = "ALT")

This 3D scatter plot allows for an interactive exploration of the hepatitis dataset, providing a clear visual representation of how Age, ALT, and AST values are distributed across different categories. Using color coding for categories makes it easier to identify trends and groupings within the data. This visualization aids in the comprehensive dataset analysis, revealing potential correlations and outliers that may not be immediately apparent in two-dimensional plots. From *Fig. 11*, we can see some outliers of ALT and AST across categories 2-5.

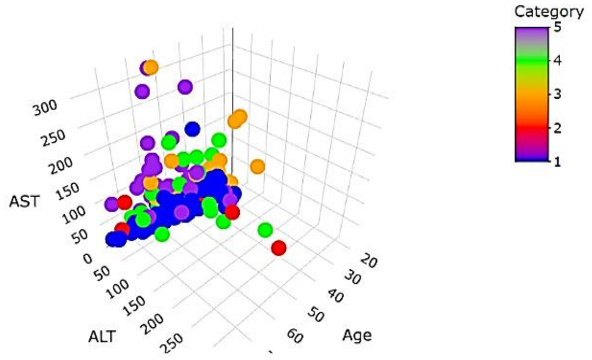


Fig. 11. 3D Alanine aminotransferase data scatter plot.

```
plot_ly(hepatitis_data, x = ~Age,          y = ~ALT,
z = ~AST,
color = ~Category,
colors = c('blue', 'red', 'orange', 'green', 'purple'),
type = 'scatter3d',
mode = 'markers') %>%
layout(title = "3D Scatter Plot of Hepatitis Data",
scene = list(xaxis = list(title = 'Age'),
yaxis = list(title = 'ALT'),
zaxis = list(title = 'AST')))
```

The density plot illustrates the distribution of CHOL levels across genders in the dataset. While the shapes of the distributions appear symmetric for both sexes, there is a notable difference in the peaks. Specifically, the peak of the female distribution appears higher than that of males, suggesting that, on average, females tend to have higher CHOL levels than males in this dataset. This observation indicates a potential gender-based difference in CHOL levels within the population under study.

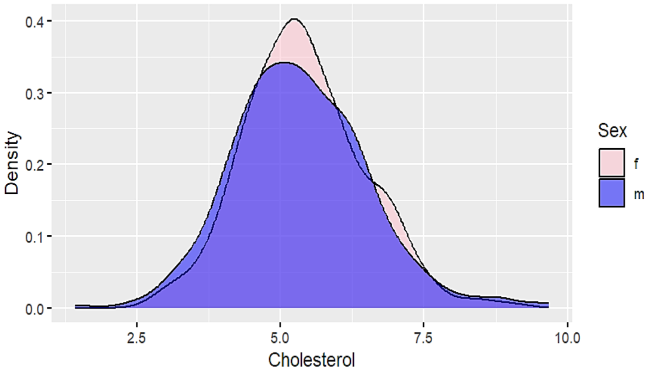


Fig. 12. Density by cholesterol based of sex.

```
ggplot(hepatitis_data, aes(x = CHOL, fill = Sex)) +
geom_density(alpha = 0.5) +
labs(title = "Density Plot of Cholesterol by Sex",
x = "Cholesterol",
y = "Density") +
scale_fill_manual(values = c("pink","blue" ))
```

The ridge plot visualizes the distribution of ALB levels (x-axis) across different categories (y-axis), where each category represents a group defined by the "category" variable in the dataset. Each ridge in the plot represents the density of ALB values within a specific category. This visualization technique allows usto easily compare the distribution of ALB levels across different groups.

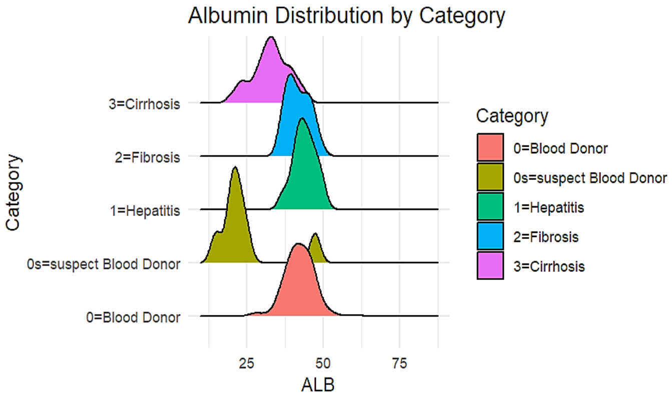


Fig. 13. Albumin distribution by category.

```
ridge_plot <- ggplot(hepatitis_data, aes(x = ALB, y = Category, fill = Category)) +
geom_density_ridges() +
labs(x = "ALB", y = "Category", title = "Albumin Distribution by Category") +
theme_minimal()
print(ridge_plo
```

Categorical parallel coordinate plots are an effective way to visualize multivariate data, allowing the comparison of several quantitative variables and how they interact across different categorical groups. This visualization can be particularly useful in medical research, where observing trends and relationships within patient data is crucial. Here, we describe the creation of a categorical parallel coordinates plot using hepatitis dataset.

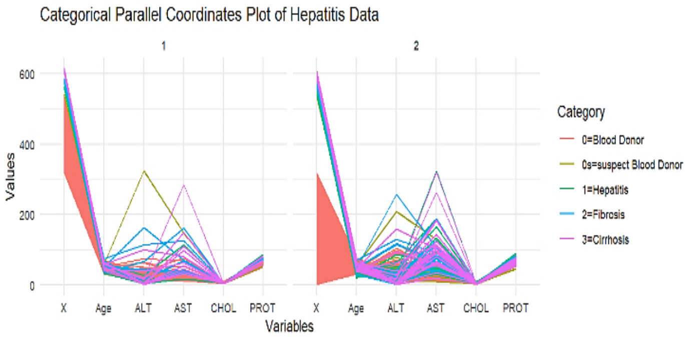


Fig. 14. Multivariate data CPCP plot.

```
cpcp_plot <- ggparcoord(data = hepatitis_data,
columns = c(1, 3, 7, 8, 11, 14),
groupColumn = 2,
scale = "globalminmax") +
labs(title = "Categorical Parallel Coordinates Plot of Hepatitis Data",
x = "Variables",
y = "Values") +
theme_minimal() +
facet_wrap(~ Sex)
print(cpcp_plot)
```

The alluvial plot effectively visualizes the flow and distribution between the Sex and category variables, highlighting patterns and relationships within the hepatitis dataset. By examining the flows and their relative sizes, one can infer the distribution of different categories across sexes, providing valuable insights into the dataset's structure and composition. This visualization method is particularly useful for understanding complex hierarchical relationships in categorical data.

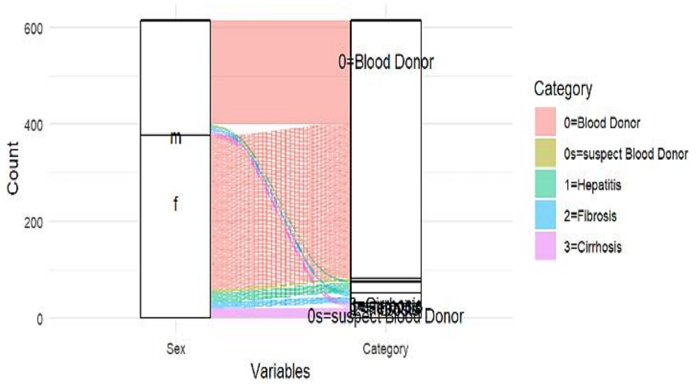


Fig. 15. Alluvial plot of hepatitis data plot.

```
ggplot(hepatitis_data,
aes(axis1 = Sex, axis2 = Category, y = ..count..)) +
geom_alluvium(aes(fill = Category)) +
geom_stratum() +geom_text(stat = "stratum", aes(label = after_stat(stratum))) +
scale_x_discrete(limits = c("Sex", "Category")) + labs(title = "Alluvial Plot of Hepatitis Data",  x =
"Variables", y = "Count") + theme_minimal()
```

This violin plot visualizes the distribution of the ALT levels across different sexes in the hepatitis dataset. The x-axis represents the categorical variable "Sex," indicating the gender of the individuals, while the y-axis displays the numerical variable "ALT," representing the ALT levels. Each violin plot is split by sex and shows the distribution of ALT levels within each group. The width of the violins corresponds to the density of ALT values, with wider sections indicating higher density. The overlaid box plots inside each violin provide additional information about the median, quartiles, and potential outliers within the data.

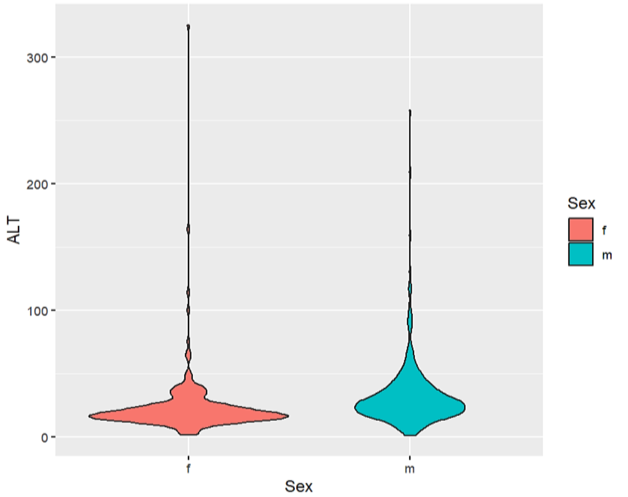


Fig. 16. Violin plot of ALT levels.

```
ggplot(hepatitis_data, aes(x = Sex, y = ALT, fill = Sex)) +geom_violin() +labs(x = "Sex", y =
"ALT", title = "Violin Plot of ALT by Sex")
```

This slope plot visualizes the relationship between the ALP levels and the CHOL levels in the hepatitis dataset. Each point on the plot represents an individual observation, with ALP values plotted on the x-axis and CHOL values plotted on the y-axis. The points are colored according to their CHOL levels for better distinction.

Additionally, a linear regression line is fitted to the data using the geom_smooth() function with the method set to "lm" (linear model), indicating the overall trend between ALP and CHOL levels.

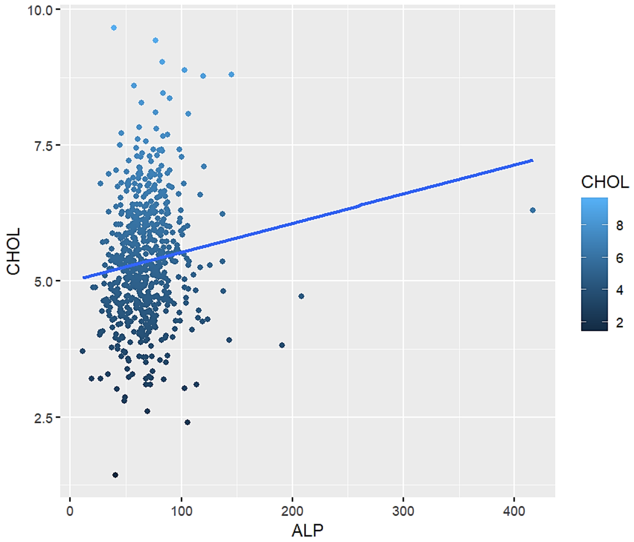


Fig. 17. ALP slope plot.

ggplot(hepatitis_data, aes(x =ALP ,y = CHOL, color = CHOL)) +geom_point() +geom_smooth(method = "lm", se = FALSE) +labs(x = "ALP", y = "CHOL", title = "Slope Plot of ALP by CHOL")

This cross-sectional plot visualizes the relationship between ALT levels and AST levels in the hepatitis dataset. Each point on the plot represents an individual observation, with ALT values plotted on the x-axis and AST values plotted on the y-axis. The plot helps to assess the correlation or association between ALT and AST levels, which are important biomarkers for liver function.

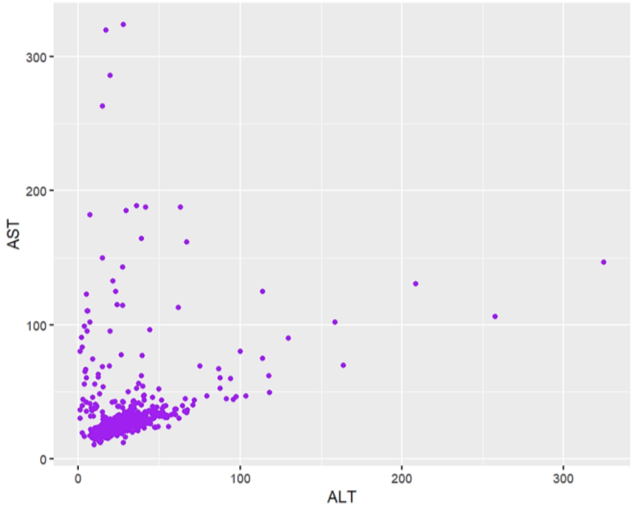


Fig. 18. ALT vs AST cross sectional plot.

ggplot(hepatitis_data, aes(x = ALT, y = AST)) +

geom_point(color = "purple") +labs(x = "ALT", y = "AST", title = "Cross-sectional Plot of ALT vs. AST")

The contour plot visualizes the joint distribution of BIL and CHOL levels in the hepatitis dataset. The filled contours represent regions of similar density, with darker areas indicating higher concentrations of data points. This visualization allows for identifying patterns and associations between BIL and CHOL levels, providing insights into their relationship within the dataset.

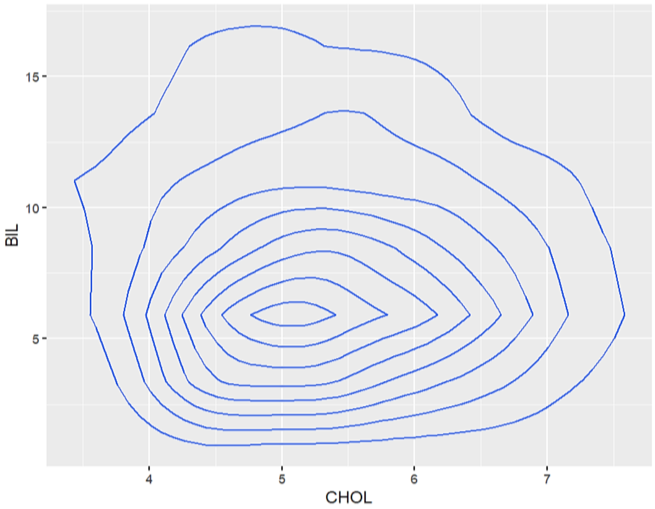


Fig. 19. Bilirubin vs cholesterol contour plot.

ggplot(hepatitis_data, aes(x = CHOL, y = BIL)) +geom_density_2d(aes(fill = ..level..)) +labs(x = "CHOL", y = "BIL", title = "Contour Plot of BIL vs CHOL")

Represents the proportion of each category in a categorical variable as slices of a circle. In this figure (*Fig. 20*), category 1 has the highest proportion of dataset represented by red color.

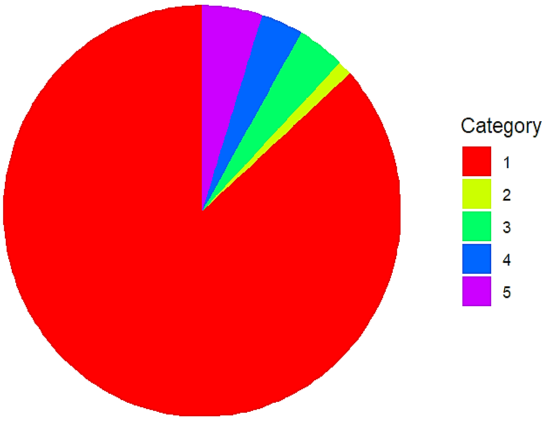


Fig. 20. Category pie chart.

pie_chart <- ggplot(category_df, aes(x = "", y = Count, fill = Category)) + geom_bar(stat = "identity", width = 1) +

coord_polar("y", start = 0) + labs(title = "Pie Chart of Hepatitis Categories")+theme_void()+ scale_fill_manual(values rainbow(length(unique(hepatitis_data$Category)))) print(pie_chart)

It illustrates the distribution of a continuous variable by plotting the proportion of data points less than or equal to each observed value. The ECDF plot shows the cumulative distribution of the AST variable in the hepatitis data. The figure shows that the cumulative probability increases exponentially until it reaches 50 instances, reaching the maximum cumulative probability of 1.

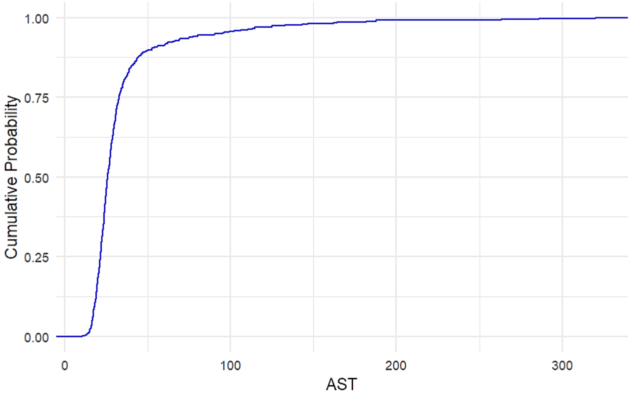


Fig. 21. Cumulative probablity of AST.

```
ecdf_plot <- ggplot(hepatitis_data, aes(x = AST)) +
stat_ecdf(geom = "step", color = "blue") +
labs(title = "ECDF Plot of AST in Hepatitis Data",
x = "AST",
y = "Cumulative Probability") +
theme_minimal()
print(ecdf_plot)
```

5 | Conclusion

This study successfully implemented visualization techniques for both univariate and multivariate datasets. Through these visualizations, we have gained valuable insights into the underlying patterns and relationships within the data. By employing interactive visualization methods and ensuring accessibility, we have endeavored to enhance user experiences and promote inclusivity. These findings lay the groundwork for further exploration and innovation in the field of data visualization, with the potential to drive impactful advancements in understanding and decision-making across various domains.

Future efforts could develop tools that make data visualization accessible to all. By pursuing these avenues, researchers can advance the field of data visualization, fostering a deeper understanding, more effective communication, and better utilization of complex data across various domains.

References

[1] Pant, A., & Rajput Head, R. S. (2019). Introduction to research data and its visualization using R. In *Writing qualitative research paper* (pp. 18–32). https://B2n.ir/jj6884

[2] Hosain, M. T., Zaman, A., Sajid, M. S., Khan, S. S., & Akter, S. (2023). Privacy preserving machine learning model personalization through federated personalized learning. *2023 4th international conference on data analytics for business and industry, ICDABI 2023* (pp. 536–545). IEEE. https://doi.org/10.1109/ICDABI60145.2023.10629638

[3] Brennan, P. (2021). Data visualization with the programming language R. *Biochemist*, *43*(5), 8–14. https://doi.org/10.1042/bio_2021_174

[4] Hosain, M. T., Abir, M. R., Rahat, M. Y., Mridha, M. F., & Mukta, S. H. (2024). Privacy preserving machine learning with federated personalized learning in artificially generated environment. *IEEE open journal of the computer society*. https://doi.org/10.1109/OJCS.2024.3466859

[5] Zoghi, Z., & Serpen, G. (2024). UNSW-NB15 computer security dataset: analysis through visualization. *Security and privacy*, *7*(1), e331. https://doi.org/10.1002/spy2.331

[6] Kucukler, O. F., Amira, A., & Malekmohamadi, H. (2024). EEG dataset for energy data visualizations. *Data in brief*, *52*, 109933. https://doi.org/10.1016/j.dib.2023.109933

[7] Pilhöfer, A., & Unwin, A. (2013). New approaches in visualization of categorical data: R package extracat. *Journal of statistical software*, *53*(7), 1–25. https://doi.org/10.18637/jss.v053.i07

[8] Nordmann, E., McAleer, P., Toivo, W., Paterson, H., & DeBruine, L. M. (2022). Data visualization using R for researchers who do not use R. *Advances in methods and practices in psychological science*, *5*(2), 25152459221074656. https://doi.org/10.1177/25152459221074654

[9] Min, S. H., & Zhou, J. (2021). Smplot: An R package for easy and elegant data visualization. *Frontiers in genetics*, *12*, 802894. https://doi.org/10.3389/fgene.2021.802894

[10] Cho, W., Lim, Y., Lee, H., Varma, M. K., Lee, M., & Choi, E. (2014). *Big data analysis with interactive visualization using R packages* [presentation]. Proceedings of the 2014 international conference on big data science and computing (pp. 1–6). https://doi.org/10.1145/2640087.264416

[11] Breheny, P., & Burchett, W. (2017). Visualization of regression models using visreg. *R journal*, *9*(2), 56–71. https://doi.org/10.32614/rj-2017-046

[12] Bale, K., Chapman, P., Barraclough, N., Purdy, J., Aydin, N., & Dark, P. (2007). Kaleidomaps: A new technique for the visualization of multivariate time-series data. *Information visualization*, *6*(2), 155–167. https://doi.org/10.1057/palgrave.ivs.9500154

[13] Lichtinghagen, R., Klawonn, F., & Hoffmann, G. (2020). *HCV data*. UCI Machine Learning Repository. https://archive.ics.uci.edu/dataset/571/hcv+data

[14] Dupont, W., & Plummer, W. (2002). Sunflower: Stata module to generate density distribution sunflower plots. *Statistical software components s430201*. https://ideas.repec.org/c/boc/bocode/s430201.html

[15] Taiyun. (2021). *Corrplot: visualization of a correlation matrix*. GitHub, Inc. Footer Navigation. https://github.com/taiyun/corrplot

[16] Egoshin, V. L., Ivanov, S. V., Savvina, N. V., Kalmakhanov, S. B., & Grjibovski, A. M. (2018). Visualization of biomedical data using R. *Ekologiya cheloveka (human ecology)*, *25*(8), 52–64. https://doi.org/10.33396/1728-0869-2018-8-52-64