BUSINESS INTELLIGENCE 381
GROUP D

CONTENTS

EXECUTIVE SUMMARY

LangaSat currently determines customer eligibility for its satellite internet service based on a fixed annual salary threshold. However, this method oversimplifies credit risk assessment, as it does not consider other key factors that may provide a more accurate view of a customer's financial capacity. To address this, LangaSat plans to develop an intelligent recommender model that incorporates additional variables such as occupation, education, and demographic factors to better assess credit risk and customer eligibility.

This program is a foundational step towards building that intelligent model by performing exploration, and analysis of a customer dataset (`CustData2`). The dataset includes various fields such as demographic information (e.g., year of birth, marital status, education) and financial data (e.g., annual salary). The program's main objectives are to ensure the integrity and quality of the dataset by addressing common data issues such as missing values, duplicates, outliers, and inconsistencies in formatting.

BACKGROUND

LangaSat is a satellite service provider that determines customer eligibility for its services based on the current yearly earnings. However, they currently use a fixed annual salary threshold of R50 000+ to determine the eligibility which may exclude the customers who can afford the services based on other factors such as occupation, education, years of residence and other customer demographics.

The goal of this project is to develop an intelligent recommender model that can accurately identify customers who are good or poor credit risks, and therefore eligible or ineligible for the service. This model will consider various customer attributes beyond just salary.

The overall aim of LangaSat is to create a more accurate and sophisticated method of determining the customer's eligibility for the company.

BUSINESS OBJECTIVES AND SUCCESS CRITERIA

Diversifying Customer Eligibility Criteria

- The business aims to enhance its customer eligibility criteria by looking beyond the fixed annual salary threshold of R50 000+. The objective is to develop a more comprehensive model that considers different customer demographics, such as occupation, education level, and years of residence, to better assess their ability to pay for the services that the business provides.

Implementation of an intelligent recommender model

- The business wants to create an intelligent recommender model that can accurately assess customer eligibility and credit risk. This model must be able to utilize multiple customer attributes to ensure a more accurate determination of whether customers can afford and sustain the services offered by the business, thereby improving customer inclusion.

Minimizing financial risk

- The business objective is to maintain a balance between expanding customer eligibility and minimizing financial risk. By implementing the model, the business should aim to ensure that the

customers that are able to get the service can meet their financial obligations, leading to sustained revenue without compromising profitability.

Success criteria

- The success of this initiative will be measured by the development of a model that expands service eligibility to customers based on multiple demographic factors, beyond just their annual salary. Additionally, the model must ensure that the business continues to receive full payment from their customers, safeguarding against financial losses while broadening the customer base.

INVENTORY OF RESOURCES

DATA RESOURCES

- Customer dataset: "CustData2.csv"
- Contains information on job titles, departments, salaries, year of birth, marital status, city of residence, years of residence, level of education, occupation, and household size.

SOFTWARE RESOURCES

- R programming environment (implied from R Markdown requirement)
- Power BI (optional, for data visualization)

HARDWARE RESOURCES

- Computers capable of running R and Power BI (specifics not provided, but assumed)

HUMAN RESOURCES

- Project team: Groups of up to four students
- Project supervisor: Course lecturer

METHODOLOGICAL RESOURCES

- CRISP-DM methodology guide.

TIME RESOURCES

- Project duration: Not explicitly stated, but spans several weeks or months given the complexity.

KNOWLEDGE RESOURCES

- Project outline document
- Project milestone documents
- Additional research materials

CONSTRAINTS

- A team of four people
- Seven completed Project Milestones

RISKS

1. Data Privacy and Security
   Regarding data privacy and security, the machine learning model will be handling sensitive customer data such as occupation, income, and other demographics. Any data breaches could lead to financial and reputational damage for LangaSat.

2. Model bias and fairness.
   Unfair decisions made by the model will result in the model unintentionally favoring or discriminating against certain demographic groups.

3. Inaccurate predictions
   No matter how sophisticated the model could be, there will always be a risk of false positives, like approving ineligible customers or false negatives like denying eligible customers, which can lead to loss of revenue or customer dissatisfaction.

4. Regulatory compliance
   The system will need to comply with financial regulatory bodies such as the National Credit Act 34 of 2005 and failure to align with the legal requirements may lead to fines or legal action.

5. Data quality issues
   If the training data used is incomplete, outdated, inaccurate or is not well prepared, the model's predictions could be compromised, which will lead to incorrect eligibility assessments.

ASSUMPTIONS

1. Availability of relevant data
   One of the assumptions is that LangaSat has access to a variety of customer dimensions beyond salary, such as occupation, education, and residential history and that this data is accurate and comprehensive.

2. Stability of customer demographics

It may not always be the case, but the model makes the assumptions that income and employment status are stable and do not vary regularly.

3. Predictive value of customer attributes
   The hypothesis that years of residency, education and employment status have a major influence on credit risk and eligibility to be validated through data analysis.

4. No drastic changes in market conditions
   The model assumes that the market and economic environments will be mostly steady. For example, a significant monetary crisis might have a significant impact on consumers' financial behavior.

CONSTRAINTS

1. Limited access to data
   It is possible that LangaSat might not have access to all the relevant client data required to create a reliable model. For instance, information on prior credit histories or spending patterns might not be accessible.

2. Model interpretability
   To justify a customer's eligibility or ineligibility, the business might require the model to be sensible. Transparency may be hampered by complex models like neural networks, which may be more difficult to understand.

3. Time and resource constraints
   Developing a sophisticated model requires time and resources, including data collection, model training and validation. LangaSat may face limitations in terms of budget and skilled personnel for this project.

4. Regulatory and ethical constraints
   Financial rules, privacy laws, and ethical issues must all be complied with by the model, which may restrict the kinds of data that can be used or the methods that can be employed.

5. Integration with current systems
   The integration of the new model with LangaSat's current decision-making processes may prove to be technically difficult, especially if the systems are rigid or outdated.

DATA MINING GOALS AND SUCCESS CRITERIA

PRIMARY DATA MINING GOAL

Develop an intelligent recommender model capable of accurately identifying customers who are good or poor credit risks, and therefore eligible or ineligible for LangaSat's satellite internet service.

SPECIFIC DATA MINING OBJECTIVES

1. Classification: Build a classification model to predict customer eligibility based on various attributes beyond just annual salary.

2. Feature Importance: Identify and rank the most significant variables that influence customer eligibility and credit risk.

3. Pattern Discovery: Uncover patterns and relationships in the customer data that may impact eligibility.

SUCCESS CRITERIA

BUSINESS SUCCESS CRITERIA

1. Improved accuracy in identifying eligible customers compared to the current salary-only method.

2. Potential increase in the customer base without significantly increasing credit risk.

3. Better understanding of factors influencing customer eligibility and credit risk.

DATA MINING SUCCESS CRITERIA

1. Model Performance:

- Achieve higher accuracy than the baseline model (using only salary for eligibility).
- Attain a minimum accuracy of 80% in predicting customer eligibility.
- Achieve balanced precision and recall scores, with a minimum F1-score of 0.75.

2. Feature Importance:

- Identify at least 3-5 significant variables besides annual salary that influence eligibility.
- Quantify the relative importance of these variables in the model.

3. Model Interpretability:

- Develop a model that can provide clear explanations for its eligibility decisions.
- Create visualizations that effectively communicate the model's decision-making process.

4. Validation:

- Demonstrate consistent performance across different subsets of the data.
- Show resilience to slight variations in input data.

5. Deployment Readiness:

- Develop a user-friendly interface for the model.
- Create a clear plan for integrating the model into LangaSat's existing systems.

6. Ethical Considerations:

- Ensure the model does not introduce or amplify biases against protected characteristics.
- Provide transparency in how customer data is used and how decisions are made.

EVALUATION METHODS

- Use appropriate metrics such as accuracy, precision, recall, and F1-score.
- Employ cross-validation techniques to ensure model robustness.
- Conduct statistical tests to validate the significance of findings.
- Perform A/B testing comparing the new model against the current salary-only method.

DATA DESCRIPTION

The dataset consists of several important variables that capture customer demographics, job details, and financial information. Key columns include variables such as Marital Status, which indicates whether the customer is single, married, or in another relationship status. Additionally, Street Address, Postal Code, City, State/Province, and Country detail the customer's residential location while Phone Number and Email provide contact information. Other critical columns include Education and Occupation which represent the customer's highest level of education and current job title. The dataset also includes Household Size indicating the number of people living in the customer's household and Years of Residence indicating how long the customer has lived in their current residence.

The dataset contains both categorical and numeric variables. Categorical variables include Marital Status, Street Address, City, State/Province, Country, Phone Number, Email, Education, and Occupation. On the other hand, Household Size and Years of Residence are numeric variables. Each row in the dataset represents a unique customer with approximately 191,000 rows in total making this a large and detailed dataset. These attributes provide a comprehensive overview of the customer profiles which will be important for building a predictive model such as a recommender system for customer eligibility.

Upon initial inspection, some columns such as Phone Number, Email, and Household Size contain missing data. Handling this missing data through imputation is essential as these gaps could impact the accuracy of the model if left untreated. For example, numeric columns like Household Size can be imputed with the median

while missing categorical data in columns like Marital Status can be replaced with "Unknown" to maintain consistency in the dataset.

Outliers may exist in variables like Years of Residence, where some customers with extremely prolonged periods of residence may skew the analysis. These outliers should be capped at the 99th percentile or removed entirely to improve the model's performance and accuracy.

Certain variables such as Street Address, Postal Code, and Phone Number may be unnecessary for the predictive model and could be dropped during the feature selection process. These columns offer little value in predicting customer eligibility for satellite services. Instead, focusing on variables such as Education, Occupation, and Years of Residence which could have strong correlations with customer eligibility will improve the model's predictive power. Exploring correlations between Years of Residence, Household Size, and Education will also help identify key drivers that influence customer eligibility for the satellite service.

DATA EXPLORATION AND VISUALIZATION

LOADING THE DATASET

Loading the dataset

```r
{r}
CustData2 <- read.csv("CustData2.csv", sep = " ", header = FALSE)
```

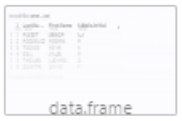
CLEANING THE DATA SET BY SPLITTING COLUMNS

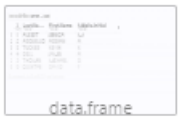
Cleaning the Last Column so that we can have separate columns to be able to do visualizations and outlier analysis

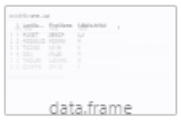
```r
{r}
#  Split the combined column into separate columns using commas as the delimiter
split_data <- strsplit(CustData2$year_of_birth.marital_status.street_address
.postal_code.city.state_province.Country_id.phone_number.email.Education.Occupation
.household_size.yrs_residence, ",")

#  Convert the list of split values into a data frame
split_data_df <- do.call(rbind, split_data)

#  Convert the split data to a data frame with the correct column names
split_data_df <- as.data.frame(split_data_df, stringsAsFactors = FALSE)
colnames(split_data_df) <- c("year_of_birth", "marital_status", "street_address",
"postal_code", "city",
                             "state_province", "Country_id", "phone_number",
"email",
                             "Education", "Occupation", "household_size",
"yrs_residence")

#  Combine the split columns with the original dataset
# Make sure that the 'Annual.Salary' column is preserved and not overwritten
CustData2_cleaned <- cbind(CustData2, split_data_df)

# Preview the cleaned dataset to check the structure and ensure 'Annual Salary' is
not overwritten
head(CustData2_cleaned)
str(CustData2_cleaned)

# Remove commas from 'Annual.Salary' and 'yrs_residence' columns before conversion
CustData2_cleaned$Annual.Salary <- gsub(",", "", CustData2_cleaned$Annual.Salary)
CustData2_cleaned$Gross.Pay.Last.Paycheck <- gsub(",", "", CustData2_cleaned$Gross
.Pay.Last.Paycheck)
CustData2_cleaned$Gross.Year.To.Date <- gsub(",", "", CustData2_cleaned$Gross.Year
.To.Date)
CustData2_cleaned$Gross.Year.To.Date...FRS.Contribution <- gsub(",", "",
CustData2_cleaned$Gross.Year.To.Date...FRS.Contribution)
CustData2_cleaned$yrs_residence <- gsub(",", "", CustData2_cleaned$yrs_residence)

#  Convert 'Annual.Salary' and 'yrs_residence' to numeric, ensuring correct
conversion
CustData2_cleaned$Annual.Salary <- as.numeric(CustData2_cleaned$Annual.Salary)
CustData2_cleaned$Gross.Pay.Last.Paycheck <- as.numeric(CustData2_cleaned$Gross.Pay
.Last.Paycheck)
CustData2_cleaned$Gross.Year.To.Date <- as.numeric(CustData2_cleaned$Gross.Year.To
.Date)
CustData2_cleaned$Gross.Year.To.Date...FRS.Contribution <- as.numeric
(CustData2_cleaned$Gross.Year.To.Date...FRS.Contribution)
```

```
head(CustData2_cleaned)
str(CustData2_cleaned)

# Remove commas from 'Annual.Salary' and 'yrs_residence' columns before conversion
CustData2_cleaned$Annual.Salary <- gsub(",", "", CustData2_cleaned$Annual.Salary)
CustData2_cleaned$Gross.Pay.Last.Paycheck <- gsub(",", "", CustData2_cleaned$Gross
.Pay.Last.Paycheck)
CustData2_cleaned$Gross.Year.To.Date <- gsub(",", "", CustData2_cleaned$Gross.Year
.To.Date)
CustData2_cleaned$Gross.Year.To.Date...FRS.Contribution <- gsub(",", "",
CustData2_cleaned$Gross.Year.To.Date...FRS.Contribution)
CustData2_cleaned$yrs_residence <- gsub(",", "", CustData2_cleaned$yrs_residence)

#  Convert 'Annual.Salary' and 'yrs_residence' to numeric, ensuring correct
conversion
CustData2_cleaned$Annual.Salary <- as.numeric(CustData2_cleaned$Annual.Salary)
CustData2_cleaned$Gross.Pay.Last.Paycheck <- as.numeric(CustData2_cleaned$Gross.Pay
.Last.Paycheck)
CustData2_cleaned$Gross.Year.To.Date <- as.numeric(CustData2_cleaned$Gross.Year.To
.Date)
CustData2_cleaned$Gross.Year.To.Date...FRS.Contribution <- as.numeric
(CustData2_cleaned$Gross.Year.To.Date...FRS.Contribution)
CustData2_cleaned$yrs_residence <- as.numeric(CustData2_cleaned$yrs_residence)
CustData2_cleaned$household_size <- as.numeric(CustData2_cleaned$household_size)



#  Preview the cleaned dataset to check the structure
head(CustData2_cleaned)
str(CustData2_cleaned)

#  Removing the last column from the dataset(added a column by mistake at the end)
CustData2_cleaned <- CustData2_cleaned[, -ncol(CustData2_cleaned)]

# Verifying that the last column has been removed
print(head(CustData2_cleaned))  # Display the first few rows to verify the deletion
```

Output:



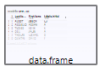



```
: chr  "married" "" "single" "married" ...
 $ street_address
: chr  "27 North Sagadahoc Boulevard" "37 West Geneva Street" "47 Toa Alta Road" "47
South Kanabec Road" ...
 $ postal_code
: chr  "60332" "55406" "34077" "72996" ...
 $ city
: chr  "Ede" "Hoofddorp" "Schimmert" "Scheveningen" ...
 $ state_province
: chr  "Gelderland" "Noord-Holland" "Limburg" "Zuid-Holland" ...
 $ Country_id
: chr  "52770" "52770" "52770" "52770" ...
 $ phone_number
: chr  "519-236-6123" "327-194-5008" "288-613-9676" "222-269-1259" ...
 $ email
: chr  "Ruddy@company.com" "Ruddy@company.com" "Ruddy@company.com"
"Ruddy@company.com" ...
 $ Education
: chr  "Masters" "Masters" "Masters" "Masters" ...
 $ Occupation
: chr  "Prof." "Prof." "Prof." "Prof." ...
 $ household_size
: num  2 2 2 2 2 2 2 2 2 ...
 $ yrs_residence
: num  4 4 4 4 4 4 4 4 4 ...
 $ NA
: chr  "1976" "1964" "1942" "1977" ...
```


Description: df [6 × 25]

	X <int>	Last.Na... <chr>	First.Name <chr>	Middle.Initial <chr>	Title <chr>	Department.Name <chr>	Annual.Salary <chr>	Gross.Pay.Last.Paycheck <chr>
1	1	ALBERT	JESSICA	M	CORRECTIONAL OFFICER	CORRECTIONS & REHABILITATION	54,619.76	2,501.62
2	2	ARGUELLO	ADRIAN	A	POLICE OFFICER	POLICE	65,250.38	3,467.63
3	3	TUCKER	KEVIN	K	CORRECTIONAL OFFICER	CORRECTIONS & REHABILITATION	62,393.76	4,513.71
4	4	DELL	JAMES	A	WASTE SCALE OPERATOR	SOLID WASTE MANAGEMENT	37,735.10	1,561.67
5	5	THOMAS	MICHAEL	D	RAIL VEHICLE ELECTRONIC TECH	TRANSPORTATION AND PUBLIC WORKS	64,386.40	6,665.66
6	6	QUINTAS	DAVID	F	POLICE SERGEANT	POLICE	89,621.22	3,802.71

6 rows | 1-9 of 25 columns



Description: df [6 × 25]

	X <int>	Last.Na... <chr>	First.Name <chr>	Middle.Initial <chr>	Title <chr>	Department.Name <chr>	Annual.Salary <dbl>	Gross.Pay.Last.Paycheck <dbl>
1	1	ALBERT	JESSICA	M	CORRECTIONAL OFFICER	CORRECTIONS & REHABILITATION	54619.76	2501.62
2	2	ARGUELLO	ADRIAN	A	POLICE OFFICER	POLICE	65250.38	3467.63
3	3	TUCKER	KEVIN	K	CORRECTIONAL OFFICER	CORRECTIONS & REHABILITATION	62393.76	4513.71
4	4	DELL	JAMES	A	WASTE SCALE OPERATOR	SOLID WASTE MANAGEMENT	37735.10	1561.67
5	5	THOMAS	MICHAEL	D	RAIL VEHICLE ELECTRONIC TECH	TRANSPORTATION AND PUBLIC WORKS	64386.40	6665.66
6	6	QUINTAS	DAVID	F	POLICE SERGEANT	POLICE	89621.22	3802.71

6 rows | 1-9 of 25 columns

Description: df [6 × 24]

year_of_birth <chr>	marital_status <chr>	street_address <chr>	postal_code <chr>	city <chr>	state_province <chr>	Country_id <chr>	phone_num... <chr>	email <chr>	Education <chr>
1976	married	27 North Sagadahoc Boulevard	60332	Ede	Gelderland	52770	519-236-6123	Ruddy@company.c...	Masters
1964		37 West Geneva Street	55406	Hoofddorp	Noord-Holland	52770	327-194-5008	Ruddy@company.c...	Masters
1942	single	47 Toa Alta Road	34077	Schimmert	Limburg	52770	288-613-9676	Ruddy@company.c...	Masters
1977	married	47 South Kanabec Road	72996	Scheveningen	Zuid-Holland	52770	222-269-1259	Ruddy@company.c...	Masters
1949		57 North 3rd Drive	67644	Joinville	Santa Catarina	52775	675-133-2226	Ruddy@company.c...	Masters
1950	single	67 East Mcintosh Avenue	83786	Nagoya	Aichi	52782	183-207-2933	Ruddy@company.c...	Masters

6 rows | 13-22 of 24 columns

Description: df [6 × 24]

postal_code <chr>	city <chr>	state_province <chr>	Country_id <chr>	phone_num... <chr>	email <chr>	Education <chr>	Occupation <chr>	household_size <dbl>	yrs_residence <dbl>
60332	Ede	Gelderland	52770	519-236-6123	Ruddy@company.c...	Masters	Prof.	2	4
55406	Hoofddorp	Noord-Holland	52770	327-194-5008	Ruddy@company.c...	Masters	Prof.	2	4
34077	Schimmert	Limburg	52770	288-613-9676	Ruddy@company.c...	Masters	Prof.	2	4
72996	Scheveningen	Zuid-Holland	52770	222-269-1259	Ruddy@company.c...	Masters	Prof.	2	4
67644	Joinville	Santa Catarina	52775	675-133-2226	Ruddy@company.c...	Masters	Prof.	2	4
83786	Nagoya	Aichi	52782	183-207-2933	Ruddy@company.c...	Masters	Prof.	2	4

6 rows | 16-25 of 24 columns

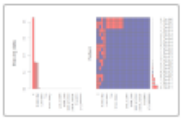
\

MISSING DATA ANALYSIS

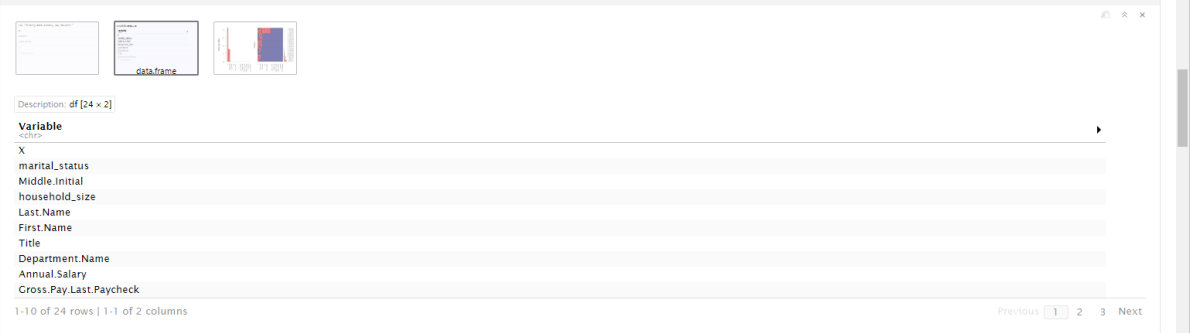
Missing Data Analysis

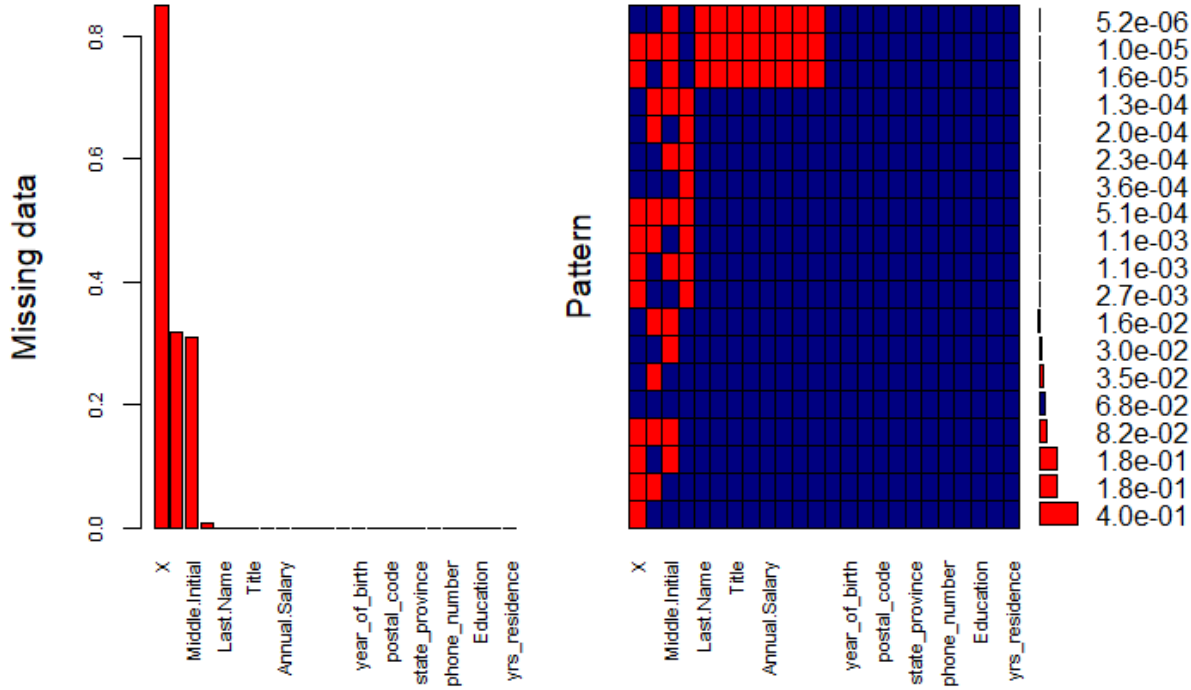
```r
# Replace empty strings with NA in the entire dataset
CustData2_cleaned[CustData2_cleaned == ""] <- NA
# Now check for missing data again
missing_data_summary <- colSums(is.na(CustData2_cleaned))
print("Missing Data Summary (by column):")
print(missing_data_summary)

# Visualization of the missing data distribution
library(VIM)
aggr_plot <- aggr(CustData2_cleaned, col = c('navyblue', 'red'), numbers = TRUE,
sortVars = TRUE, labels = names(CustData2_cleaned), cex.axis = 0.7, gap = 3, ylab =
c("Missing data", "Pattern"))
```

Output:

```
Occupation

0

household_size

1228

yrs_residence

0
Loading required package: colorspace
Loading required package: grid
VIM is ready to use.

Suggestions and bug-reports can be submitted at:
https://github.com/statistikat/VIM/issues

Attaching package: 'VIM'

The following object is masked from 'package:datasets':

    sleep


 Variables sorted by number of missings:
```





EXPLORING DUPLICATES

Exploring Duplicate Rows

```r
{r}
#  Checking for duplicate rows in the dataset
duplicate_rows <- duplicated(CustData2_cleaned)

#  Print the number of duplicate rows
print(paste("Number of duplicate rows:", sum(duplicate_rows)))

# Checking actual duplicate rows
if (sum(duplicate_rows) > 0) {
  duplicates_data <- CustData2_cleaned[duplicate_rows, ]
  print("Duplicate Rows:")
  print(head(duplicates_data))  # Display the first few duplicate rows
}

#View duplicates in RStudio
if (sum(duplicate_rows) > 0) {
  view(duplicates_data)
}
```

Output:

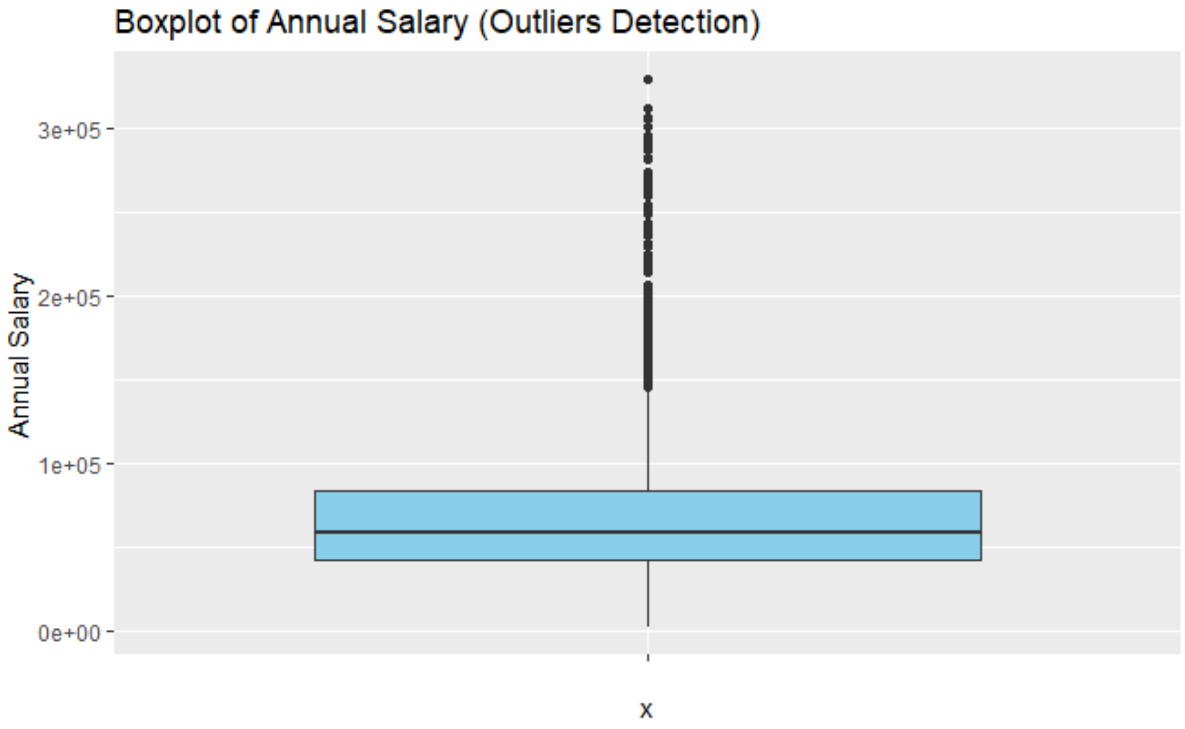
```
[1] "Number of duplicate rows: 0"
```

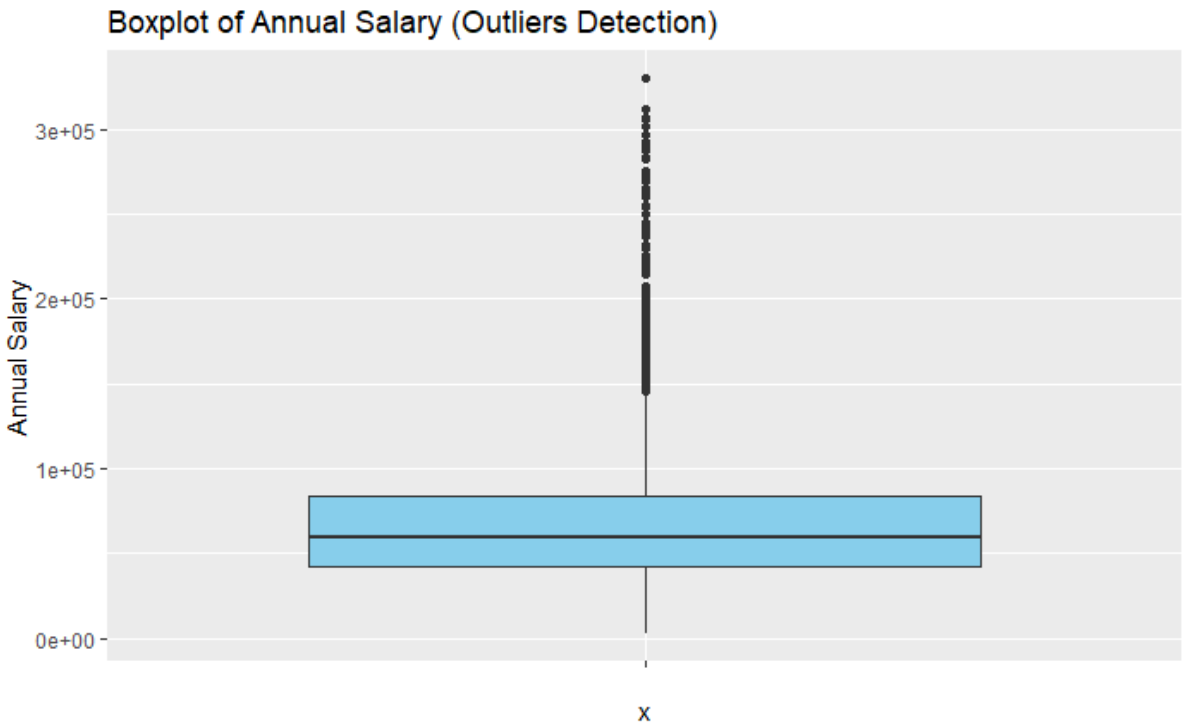
EXPLORING OUTLIERS (VISUALIZATION)

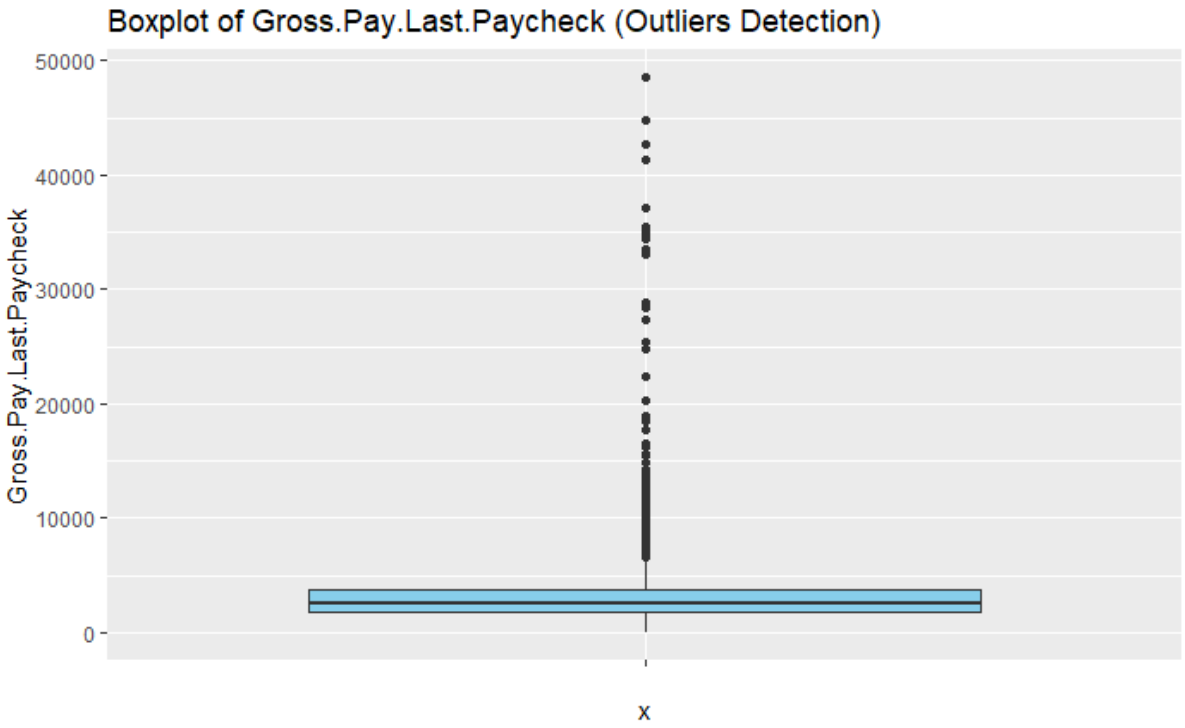
Exploring Outliers(Visualization)

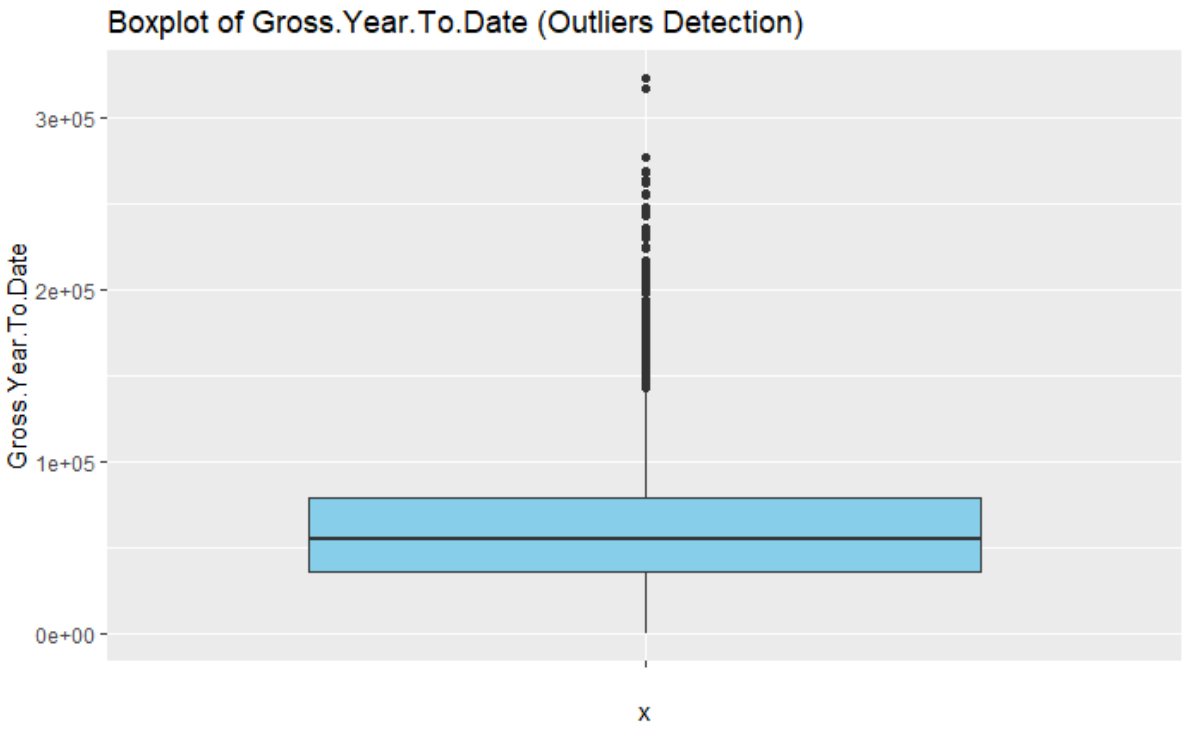
```r
{r}
#  Boxplot for 'Annual Salary' to visually identify outliers
ggplot(CustData2_cleaned, aes(x = "", y = Annual.Salary)) +
  geom_boxplot(fill = "skyblue") +
  ggtitle("Boxplot of Annual Salary (Outliers Detection)") +
  ylab("Annual Salary")

#  Boxplot for 'Gross.Pay.Last.Paycheck' to visually identify outliers
ggplot(CustData2_cleaned, aes(x = "", y = Gross.Pay.Last.Paycheck)) +
  geom_boxplot(fill = "skyblue") +
  ggtitle("Boxplot of Gross.Pay.Last.Paycheck (Outliers Detection)") +
  ylab("Gross.Pay.Last.Paycheck")

#  Boxplot for 'Gross.Year.To.Date' to visually identify outliers
ggplot(CustData2_cleaned, aes(x = "", y = Gross.Year.To.Date)) +
  geom_boxplot(fill = "skyblue") +
  ggtitle("Boxplot of Gross.Year.To.Date (Outliers Detection)") +
  ylab("Gross.Year.To.Date")

#  Boxplot for 'Gross.Year.To.Date...FRS.Contribution' to visually identify outliers
ggplot(CustData2_cleaned, aes(x = "", y = Gross.Year.To.Date...FRS.Contribution)) +
  geom_boxplot(fill = "skyblue") +
  ggtitle("Boxplot of Gross.Year.To.Date...FRS.Contribution (Outliers Detection)") +
  ylab("Gross.Year.To.Date...FRS.Contribution")

#  Boxplot for 'household_size' to visually identify outliers
ggplot(CustData2_cleaned, aes(x = "", y = household_size)) +
  geom_boxplot(fill = "skyblue") +
  ggtitle("Boxplot of household_size (Outliers Detection)") +
  ylab("household_size")

# Boxplot for 'Years of Residence'
ggplot(CustData2_cleaned, aes(x = "", y = yrs_residence)) +
  geom_boxplot(fill = "lightblue") +
  ggtitle("Boxplot of Years of Residence (Outliers Detection)") +
  ylab("Years of Residence")

#  Identifying outliers using IQR

# For Annual Salary
Q1_salary <- quantile(CustData2_cleaned$Annual.Salary, 0.25, na.rm = TRUE)
Q3_salary <- quantile(CustData2_cleaned$Annual.Salary, 0.75, na.rm = TRUE)
IQR_salary <- Q3_salary - Q1_salary
```

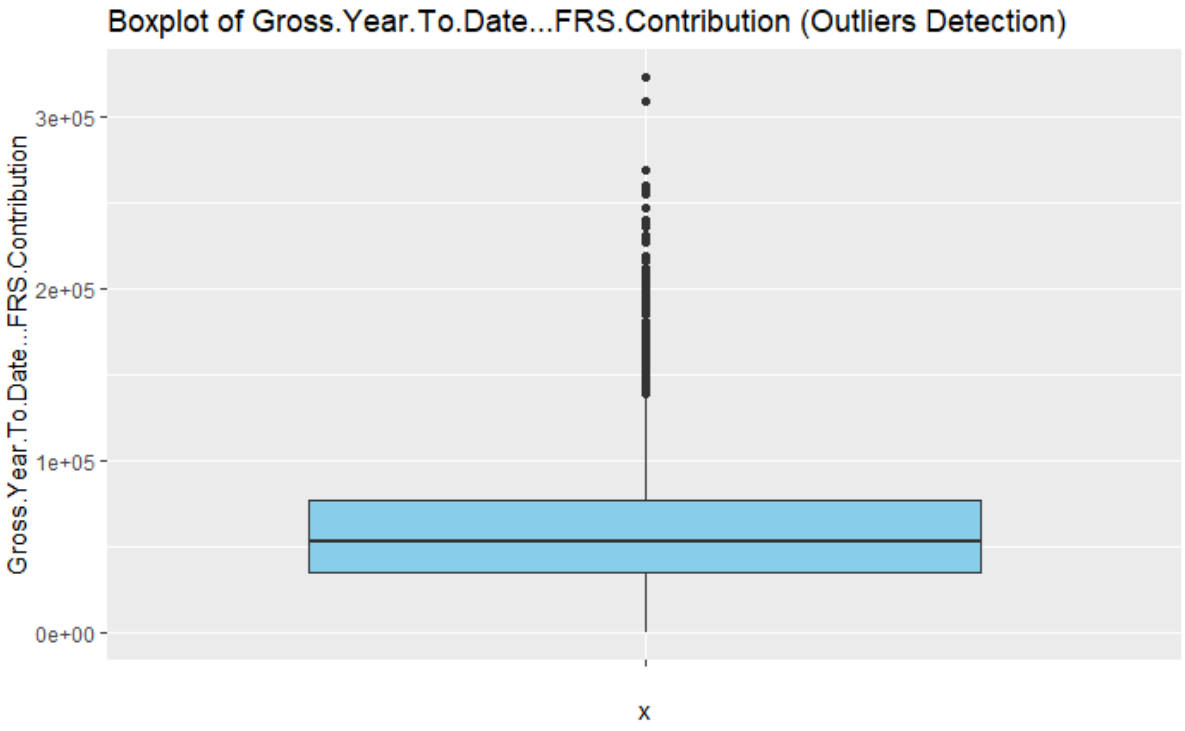
```r
#  Boxplot for 'Gross.Year.To.Date' to visually identify outliers
ggplot(CustData2_cleaned, aes(x = "", y = Gross.Year.To.Date)) +
  geom_boxplot(fill = "skyblue") +
  ggtitle("Boxplot of Gross.Year.To.Date (Outliers Detection)") +
  ylab("Gross.Year.To.Date")

#  Boxplot for 'Gross.Year.To.Date...FRS.Contribution' to visually identify outliers
ggplot(CustData2_cleaned, aes(x = "", y = Gross.Year.To.Date...FRS.Contribution)) +
  geom_boxplot(fill = "skyblue") +
  ggtitle("Boxplot of Gross.Year.To.Date...FRS.Contribution (Outliers Detection)") +
  ylab("Gross.Year.To.Date...FRS.Contribution")

#  Boxplot for 'household_size' to visually identify outliers
ggplot(CustData2_cleaned, aes(x = "", y = household_size)) +
  geom_boxplot(fill = "skyblue") +
  ggtitle("Boxplot of household_size (Outliers Detection)") +
  ylab("household_size")

# Boxplot for 'Years of Residence'
ggplot(CustData2_cleaned, aes(x = "", y = yrs_residence)) +
  geom_boxplot(fill = "lightblue") +
  ggtitle("Boxplot of Years of Residence (Outliers Detection)") +
  ylab("Years of Residence")

#  Identifying outliers using IQR

# For Annual Salary
Q1_salary <- quantile(CustData2_cleaned$Annual.Salary, 0.25, na.rm = TRUE)
Q3_salary <- quantile(CustData2_cleaned$Annual.Salary, 0.75, na.rm = TRUE)
IQR_salary <- Q3_salary - Q1_salary

lower_bound_salary <- Q1_salary - 1.5 * IQR_salary
upper_bound_salary <- Q3_salary + 1.5 * IQR_salary

salary_outliers <- CustData2_cleaned$Annual.Salary[CustData2_cleaned$Annual.Salary < lower_bound_salary | CustData2_cleaned$Annual.Salary > upper_bound_salary]
print(paste("Number of outliers in Annual Salary:", length(salary_outliers)))

# For Years of Residence
Q1_residence <- quantile(CustData2_cleaned$yrs_residence, 0.25, na.rm = TRUE)
Q3_residence <- quantile(CustData2_cleaned$yrs_residence, 0.75, na.rm = TRUE)
IQR_residence <- Q3_residence - Q1_residence

lower_bound_residence <- Q1_residence - 1.5 * IQR_residence
upper_bound_residence <- Q3_residence + 1.5 * IQR_residence

residence_outliers <- CustData2_cleaned$yrs_residence[CustData2_cleaned$yrs_residence < lower_bound_residence | CustData2_cleaned$yrs_residence > upper_bound_residence]
print(paste("Number of outliers in Years of Residence:", length(residence_outliers)))
```

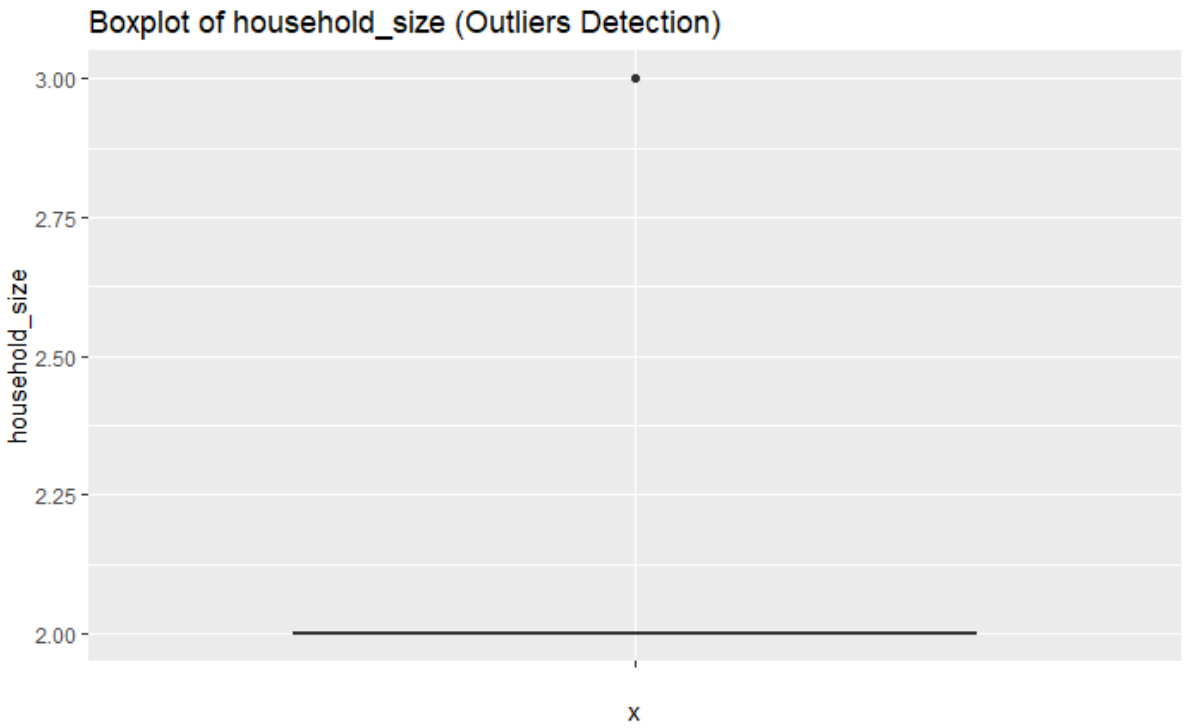

Boxplot of Annual Salary (Outliers Detection)
Annual Salary
3e+05
2e+05
1e+05
0e+00
x

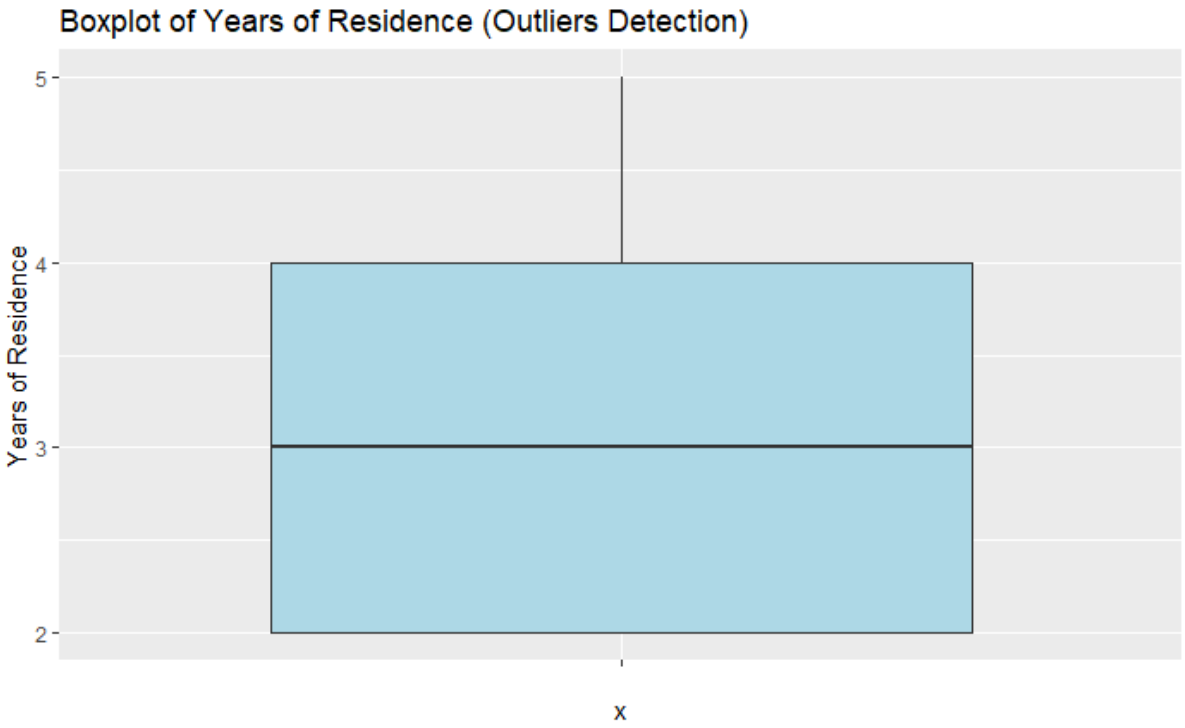

Boxplot of Annual Salary (Outliers Detection)
Annual Salary
3e+05
2e+05
1e+05
0e+00
x


Boxplot of Gross.Pay.Last.Paycheck (Outliers Detection)
Gross.Pay.Last.Paycheck
50000
40000
30000
20000
10000
0
x


Boxplot of Gross.Year.To.Date (Outliers Detection)
Gross.Year.To.Date
3e+05
2e+05
1e+05
0e+00
x


Boxplot of Gross.Year.To.Date...FRS.Contribution (Outliers Detection)
Gross.Year.To.Date...FRS.Contribution
3e+05
2e+05
1e+05
0e+00
x


Boxplot of household_size (Outliers Detection)
household_size
3.00
2.75
2.50
2.25
2.00
x


Boxplot of Years of Residence (Outliers Detection)
Years of Residence
5
4
3
2
x

OUTLIER ANALYSIS

Outlier Analysis

```r
{r}
#  Function to calculate outliers using IQR method
detect_outliers <- function(column) {
  Q1 <- quantile(column, 0.25, na.rm = TRUE)
  Q3 <- quantile(column, 0.75, na.rm = TRUE)
  IQR <- Q3 - Q1

  # Define the lower and upper bounds for outliers
  lower_bound <- Q1 - 1.5 * IQR
  upper_bound <- Q3 + 1.5 * IQR

  # Identifying the outliers
  outliers <- column[column < lower_bound | column > upper_bound]
  return(outliers)
}

# Count outliers for each column

# Annual Salary
annual_salary_outliers <- detect_outliers(CustData2_cleaned$Annual.Salary)
annual_salary_outliers_count <- length(annual_salary_outliers)
cat("Number of outliers in Annual Salary:", annual_salary_outliers_count, "\n")

# Gross Pay Last Paycheck
gross_pay_last_paycheck_outliers <- detect_outliers(CustData2_cleaned$Gross.Pay
.Last.Paycheck)
gross_pay_last_paycheck_outliers_count <- length(gross_pay_last_paycheck_outliers)
cat("Number of outliers in Gross Pay Last Paycheck:",
gross_pay_last_paycheck_outliers_count, "\n")

# Gross Year to Date
gross_year_to_date_outliers <- detect_outliers(CustData2_cleaned$Gross.Year.To.Date
)
gross_year_to_date_outliers_count <- length(gross_year_to_date_outliers)
cat("Number of outliers in Gross Year to Date:", gross_year_to_date_outliers_count,
"\n")

# Gross Year to Date FRS Contribution
gross_year_to_date_frs_outliers <- detect_outliers(CustData2_cleaned$Gross.Year.To
.Date...FRS.Contribution)
gross_year_to_date_frs_outliers_count <- length(gross_year_to_date_frs_outliers)
cat("Number of outliers in Gross Year to Date FRS Contribution:",
gross_year_to_date_frs_outliers_count, "\n")

# Household Size
household_size_outliers <- detect_outliers(CustData2_cleaned$household_size)
household_size_outliers_count <- length(household_size_outliers)
cat("Number of outliers in Household Size:", household_size_outliers_count, "\n")

# Years of Residence
```

Output

```
Number of outliers in Annual Salary: 2204
Number of outliers in Gross Pay Last Paycheck: 5952
Number of outliers in Gross Year to Date: 2114
Number of outliers in Gross Year to Date FRS Contribution: 2154
Number of outliers in Household Size: 25906
Number of outliers in Years of Residence: 0
```

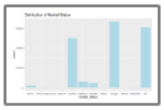
SUMMARY FOR STATISTICS FOR CATEGORICAL AND NUMERICAL DATA

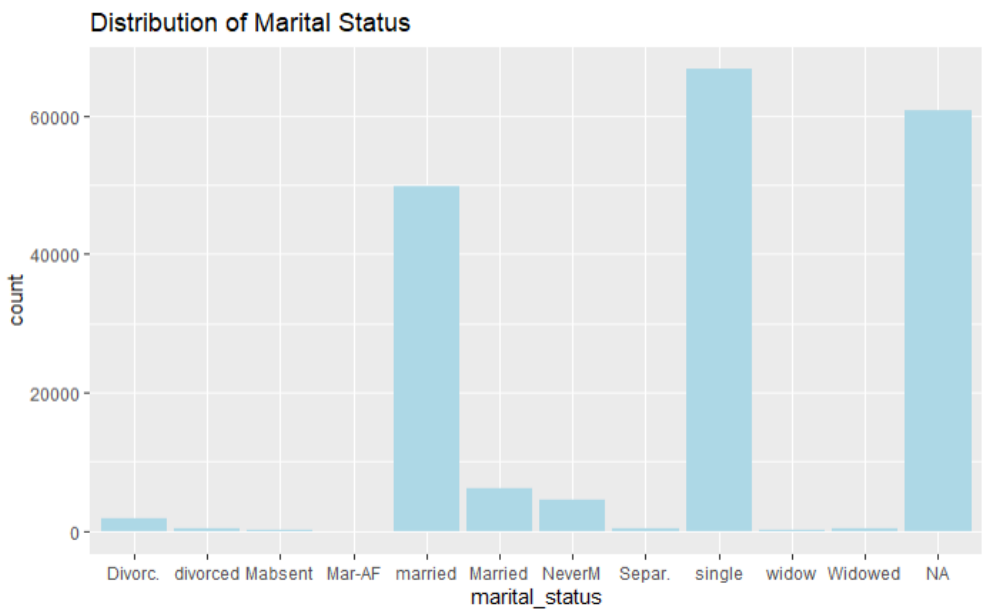
Summary Statistics for Categorical and Numeric Data

```r
{r}

#  Summary of statistics for numeric columns
library(ggplot2)
numeric_summary <- summary(CustData2_cleaned[, sapply(CustData2_cleaned, is.numeric
)])
print("Summary statistics for numeric data:")
print(numeric_summary)

# Frequency distribution of categorical columns (Marital Status)
print("Frequency distribution for Marital Status:")
print(table(CustData2_cleaned$marital_status))

# Visualization of the distribution of a categorical variable ( Marital Status)
#install.packages("ggplot2")
ggplot(CustData2_cleaned, aes(x = marital_status)) +
  geom_bar(fill = "lightblue") +
  ggtitle("Distribution of Marital Status")
```

Output:


```
[1] "Summary statistics for numeric data:"
       X          Annual.Salary    Gross.Pay.Last.Paycheck Gross.Year.To.Date
Gross.Year.To.Date...FRS.Contribution household_size yrs_residence
 Min.   :    1    Min.   :  2756   Min.   :  -11.33        Min.   :     0     Min.   :
0                        Min.   :2.00   Min.   :2.000
 1st Qu.: 7187    1st Qu.: 42537   1st Qu.: 1740.11        1st Qu.: 35984     1st Qu.:
35030                      1st Qu.:2.00   1st Qu.:2.000
 Median :14374    Median : 58987   Median : 2581.56        Median : 54703     Median :
53170                      Median :2.00   Median :3.000
 Mean   :14374    Mean   : 63933   Mean   : 2868.06        Mean   : 57923     Mean   :
56379                      Mean   :2.13   Mean   :3.252
 3rd Qu.:21560    3rd Qu.: 83850   3rd Qu.: 3682.00        3rd Qu.: 78555     3rd Qu.:
76446                      3rd Qu.:2.00   3rd Qu.:4.000
 Max.   :28746    Max.   :329680   Max.   :48530.27        Max.   :322713     Max.
:322713                       Max.   :3.00   Max.   :5.000
 NA's   :162577   NA's   :6        NA's   :6               NA's   :6          NA's   :6
NA's   :1228
[1] "Frequency distribution for Marital Status:"

 Divorc. divorced  Mabsent   Mar-AF  married  Married   NeverM   Separ.   single
widow  Widowed
    1845      450      225        9    49678     6102     4509      402    66675
225      408
```



Exploring Negative or Inconsistent Values

```r
# Checking for negative values in 'Annual Salary'
negative_salaries <- sum(CustData2_cleaned$Annual.Salary < 0, na.rm = TRUE)
print(paste("Number of negative values in Annual Salary:", sum
(negative_salaries)))

gross_pay_last <- sum(CustData2_cleaned$Gross.Pay.Last.Paycheck < 0, na.rm =
TRUE)
print(paste("Number of negative values in Gross.Pay.Last.Paycheck:", sum
(gross_pay_last)))

gross_year_to_date <- sum(CustData2_cleaned$Gross.Year.To.Date < 0, na.rm =
TRUE)
print(paste("Number of negative values in Gross.Year.To.Date:", sum
(gross_year_to_date)))

FRS <- sum(CustData2_cleaned$Gross.Year.To.Date...FRS.Contribution < 0, na.rm
= TRUE)
print(paste("Number of negative values in Gross.Year.To.Date...FRS
.Contribution:", sum(FRS)))

house <- sum(CustData2_cleaned$household_size < 0, na.rm = TRUE)
print(paste("Number of negative values in household_size:", sum(house)))

# Checking for negative values in 'Years of Residence'
negative_residence <- sum(CustData2_cleaned$yrs_residence < 0, na.rm = TRUE)
print(paste("Number of negative values in Years of Residence:", sum
(negative_residence)))
```

Output:

```
[1] "Number of negative values in Annual Salary: 0"
[1] "Number of negative values in Gross.Pay.Last.Paycheck: 6"
[1] "Number of negative values in Gross.Year.To.Date: 0"
[1] "Number of negative values in Gross.Year.To.Date...FRS.Contribution: 0"
[1] "Number of negative values in household_size: 0"
[1] "Number of negative values in Years of Residence: 0"
```

DATA QUALITY ASSESSMENT

A **Data Quality Assessment** involves evaluating the dataset's overall integrity and reliability to ensure that it can be used for accurate data analysis and modelling. This process typically includes assessing various factors such as:

COMPLETENESS

Completeness refers to the extent to which data is missing from the dataset. The dataset exhibits significant missing values, especially in key financial columns:

- **Annual Salary** has **162,577** missing values, which is a substantial portion of the dataset.
- Other columns such as **Gross Pay Last Paycheck** and **Household Size** also have missing values, although in smaller numbers.
- **Marital Status** contains some blank values represented as "", indicating incomplete records.

To address completeness, missing data needs to be handled through methods like imputation (filling missing values with averages, medians, or placeholders) or dropping rows or columns based on the significance of the missing data.

CONSISTENCY

Consistency ensures that the data follows standardized formats across the dataset. In this case, categorical variables such as **Marital Status** show inconsistent formats, with variations like "married" and "Married." Consistency is crucial for accurate analysis, especially in categorical data.
Numeric fields like **Annual Salary** and **Household Size** are consistent in terms of their data types after conversion, but the presence of outliers may suggest inconsistencies in data entry.

ACCURACY

Accuracy assesses whether the data entries correctly reflect the real-world values they are supposed to represent. Based on the analysis:

- **Outliers**: There are a large number of outliers in several key columns, including **2204** outliers in **Annual Salary** and **25906** in **Household Size**. These extreme values might not be accurate reflections of real data and could distort the analysis.
- Negative values were detected in **Gross Pay Last Paycheck**, which may indicate erroneous entries or unusual cases like refunds.

OUTLIERS

Outliers are data points that significantly differ from other observations, and their presence can skew analysis, leading to inaccurate results or biased modeling. Outlier detection is critical for ensuring the integrity of the dataset, particularly in columns with numeric data.
In this dataset, the following columns have been evaluated for outliers:

- **Annual Salary**: With **2204 outliers** detected, the presence of these extreme values can distort the analysis of employee earnings. These outliers may represent unusually high or low salaries, which can

negatively affect any salary-related insights or predictions. Strategies to handle these outliers may include capping them to a reasonable maximum or removing them from the dataset.

- **Gross Pay Last Paycheck**: A substantial number of **5952 outliers** were found. This suggests discrepancies in payroll data, such as exceptionally high or low paycheck amounts. These values may require investigation to ensure they are valid or can be addressed through capping or normalization methods.
- **Gross Year to Date**: **2114 outliers** in this column indicate variability in the total earnings over the year. Such outliers might be the result of bonus payments, overtime, or data entry errors. Handling these values is important to maintain accuracy in financial reporting and modeling.
- **Gross Year to Date FRS Contribution**: **2154 outliers** were detected in FRS contributions, which may signify unusual contribution amounts. Outliers here could impact predictions related to employee benefits or tax calculations and may require further examination.
- **Household Size**: The column shows an exceptionally high number of **25906 outliers**, suggesting data entry errors or highly irregular household sizes. This is a critical area for cleaning, as these extreme values are likely to distort any analysis related to customer demographics or household metrics.
- **Years of Residence**: No outliers were detected in this column, indicating a consistent and reasonable range for the length of residence. This consistency adds confidence in the accuracy of this variable for modeling or analysis.

DUPLICATES

Checking for duplicate records ensures that each data entry represents a unique customer. Duplicates can lead to incorrect analysis or biased model predictions, so it's essential to identify and remove them from the dataset. In this data set there is no duplicate rows.

VALIDITY

Validity refers to whether the data adheres to expected formats, types, and ranges. In this dataset:

- **Annual Salary** and **Years of Residence** were converted successfully to numeric types, which is essential for further numeric analysis.
- Categorical variables like **Marital Status** show some inconsistencies in representation and missing values.

Further validation should ensure that all categorical variables are cleaned for consistency and that all numeric variables remain within reasonable, valid ranges.

DATA DISTRIBUTION AND VISUAL INSPECTION

Visualization of the data reveals important insights:

- The distribution of **Marital Status** shows a high number of **NA** values and an unusual concentration of responses in categories like **single** and **married**, which may indicate data entry patterns or issues.
- Boxplots and summaries of **Annual Salary**, Gross **Pay** and **Years of Residence** reveal skewed distributions, influenced by outliers.

Visualizing data distributions (using histograms, boxplots, or bar charts) allows the detection of unusual patterns or anomalies in the data. For example, plotting the distribution of "Years of Residence" or "Annual Salary" helps to identify skewness or unusual peaks in the data.

HANDLING MISSING DATA

Handling missing data is critical, especially in this dataset where many missing values were detected. Potential strategies include:

- For numeric fields like **Annual Salary**, impute missing values using the median (to minimize the impact of outliers).
- For categorical fields like **Marital Status**, replace missing values with a placeholder like **"Unknown"** or the most frequent value in the dataset.

Effective handling of missing data will ensure that the dataset is more dependable and complete for analysis.

CONCLUSION

The dataset faces several data quality challenges, including missing values, outliers, and inconsistent categorical variables. Proper handling of missing data, and outliers. Ensuring consistency and accuracy will significantly improve the quality of the data, making it more reliable for predictive modeling and analysis. This assessment provides a framework for improving data quality, which is essential for generating accurate insights from the dataset.