Home Loan Approval Data Cleansing and Exploratory Data Analysis

DSM110 – R For Data Science - CW1

Aims

This project aims to prepare a dataset for future modelling. To do this the dataset will be explored, analysed and cleaned. This analysis will identify potential issues with the dataset that could hinder future modelling. It will explore incomplete data and use appropriate methods to rectify these. It will check datatypes to ensure the correct types are used for each variable. And outliers will be flagged and explored. To ensure the data is formatted correctly, the principles of tidy data will be adhered to, stating each column is a single variable, each row is a single observation, and each variable follows the same unit [1]. To align with the coursework requirements my project will run using a base install of R and libraries from the tidyverse.

Through the exploration of my data, I have identified several key variables that I hypothesise can be used to create a machine learning model to predict the outcome of the target variable (Loan_Status) given the dataset features. In CW2 I aim to build multiple machine learning classifier models (for example logistic regression, random forest). I will then use evaluation metrics such as accuracy F1-Score and AUC-ROC to compare models and find which is best suited to the data. From the analysis, some of the key variables identified are applicant and coapplicant income, credit history, education and marital status. With these in mind and as highlighted throughout the report, the dataset I have chosen is suitable for further modelling in CW2.

I've taken some specific approaches during my analysis that I will briefly cover before delving further. I made the choice to use tibbles over standard data fames as this will help make code more robust, clean and error free [2]. For categorical variables I chose to use Factors over Character datatypes this is best practice as factors are designed to store a fixed set of values and are more memory efficient as they store data as integers rather than strings [3]. Finally, I chose to use vectorised operations such as apply, sapply, mutate and various mathematical functions rather than loops. R is designed to work with vectors; therefore, vectorised operations are much faster than loops and usually involve less code which makes for better memory efficiency [4].

Data Gathering

The chosen data set is from Kaggle. It was retrieved by downloading it in .csv format from here [5]. The dataset was chosen for two reasons, firstly the automation of loan approvals saves financial institutions time and money by taking away any manual intervention. Therefore, this is an important subject to study. Secondly, this dataset met the requirements for CW1 and CW2 as it presents a dataset that has both numeric and categorical features, it has missing values, and it has inconsistent datatypes. It will also be suitable for training machine learning models which I believe will be the subject of CW2. The dataset consists of 614 observations and 13 variables. The target variable is Loan_Status with 422 records approved ("Y") and 192 rejected ("N"). This means the dataset is imbalanced.

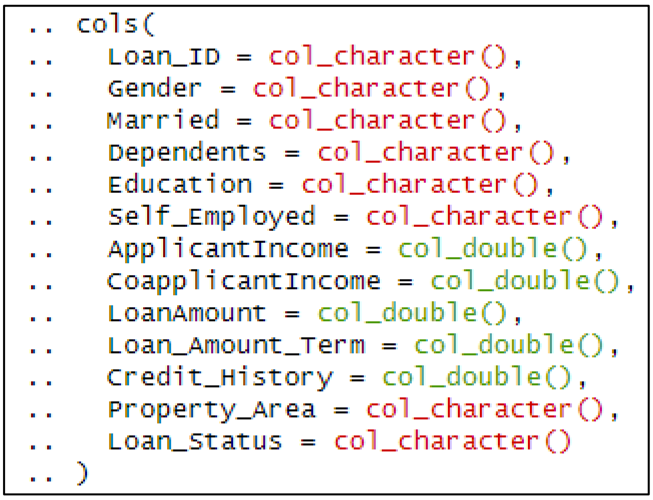
Running the code

To run the code, the files will first need to be extracted from the zip file and saved in an appropriate destination. Then open R studio and go File, Open File and navigate to where the .R file is saved. In the code replace the string assigned to the folder variable in code line 18 with the folder where the .csv file is saved. Finally highlight all the code and hit Run.

Data checking

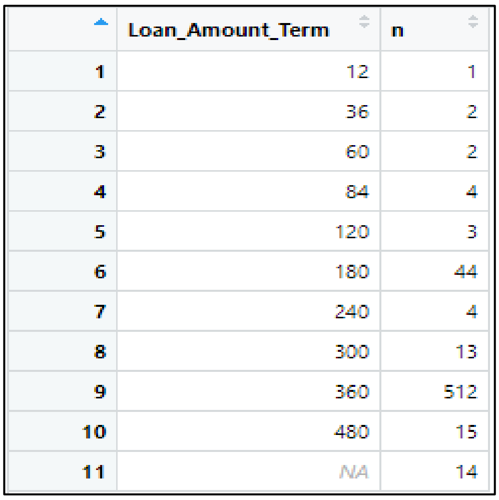
Firstly to achieve the R1 objective I loaded my data into R via the .csv file downloaded from Kaggle, to do this I used the read_csv() function which is part of the readr package from the tidyverse. Then to achieve the R2 objective I went on to check the data and confirm whether it was clean and consistent by using a variety of methods. The datatypes of each variable were checked using the str() function. This is a useful function that creates a summary of the structure of an R object. This showed the data set consisted of 2 datatypes – character and numeric (see figure 1). Further analysis using pairs() to generate pairplots of the numerical data and getting a count of the values it showed that the Loan_Amount_Term was actually categorical. Further inspection of several of the character datatypes confirmed these were also categorical (see figure 2). I will go into more detail below, but I decided these categorical datatypes would be better placed as Factor datatypes. I also used the summarise() function and the dplyr library and summarise_all() to check for missing values and then visualised this with ggplot2 (see figure 3) (code lines 28-135). Outliers in the numerical columns were checked too, these will be explained in more detail in the EDA section.

Figure 1



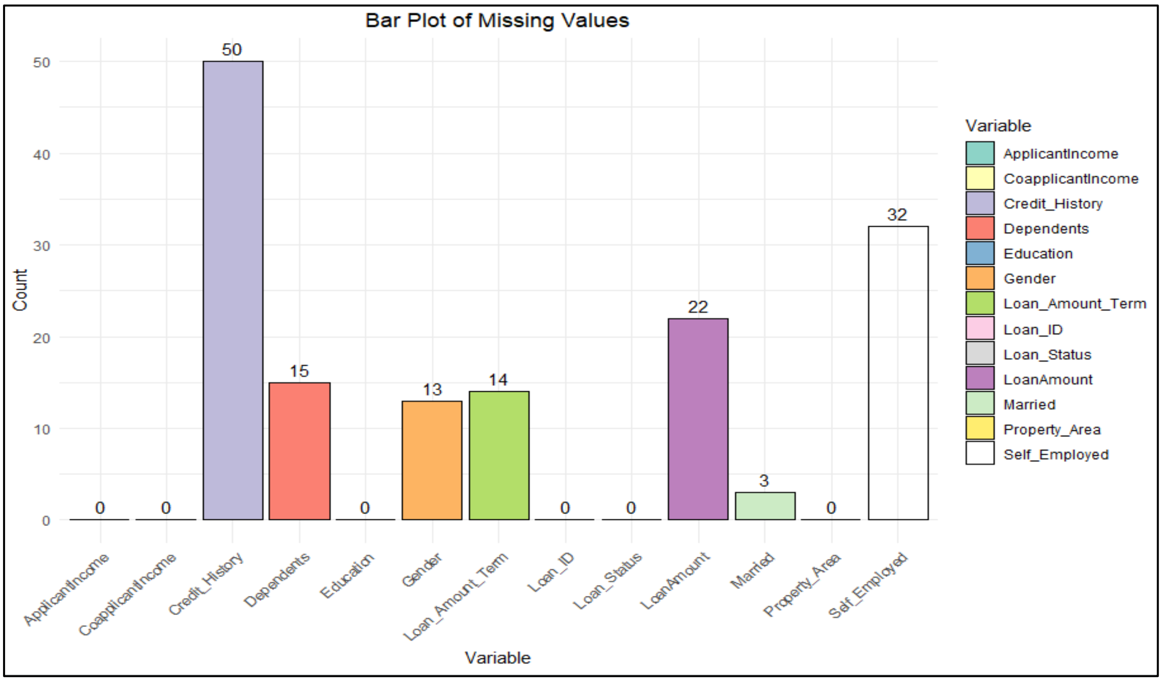
```
.. cols(
..   Loan_ID = col_character(),
..   Gender = col_character(),
..   Married = col_character(),
..   Dependents = col_character(),
..   Education = col_character(),
..   Self_Employed = col_character(),
..   ApplicantIncome = col_double(),
..   CoapplicantIncome = col_double(),
..   LoanAmount = col_double(),
..   Loan_Amount_Term = col_double(),
..   Credit_History = col_double(),
..   Property_Area = col_character(),
..   Loan_Status = col_character()
.. )
```

Figure 2



	Loan_Amount_Term	n
1	12	1
2	36	2
3	60	2
4	84	4
5	120	3
6	180	44
7	240	4
8	300	13
9	360	512
10	480	15
11	NA	14

Figure 3



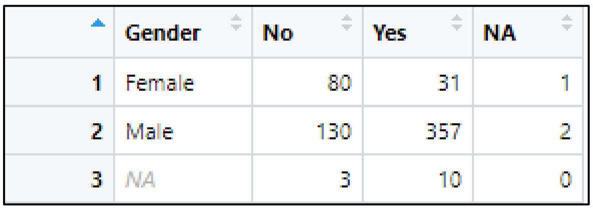
Dataset manipulations and data cleaning

This section describes how I achieved the R3 objective. Before making any changes, I began by creating a new data frame called clean_data_tb. Whilst creating this dataframe I also changed the categorical variables to Factors as highlighted in the aims section (code lines 148-159). I then deleted Loan_ID as it is just a reference number and meaningless for analysis. Finally, I updated the column headers of the income and loan amount variables, so the principles of tidy data are better adhered to (code lines 174-179).

My approach was to fill the missing values rather than remove any as the dataset is not particular large so removing any could impact later modelling. I used a variety of methods depending on the variable. For the Gender and Married variables, I noted there was a correlation between the variables – when performing a chi-square test the p-value was close to 0 and viewed as a table (Figure 4) showed that 73% of male applicants were married compared to only 28% of female applicants. Therefore, rather than just using the mode of each variable I decided to impute NA's in the Gender column based on Married and vice versa (code lines 196-236).

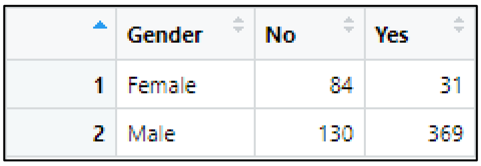
Figure 4

Raw



	Gender	No	Yes	NA
1	Female	80	31	1
2	Male	130	357	2
3	NA	3	10	0

Clean



	Gender	No	Yes
1	Female	84	31
2	Male	130	369

For the missing values in the Loan_Amount variable I used the median based on the Property_Area variable to fill the nulls (code lines 21-258). My rationale is house prices will be different dependant on location, which in turns means the loan amount required will change – median loan amounts:

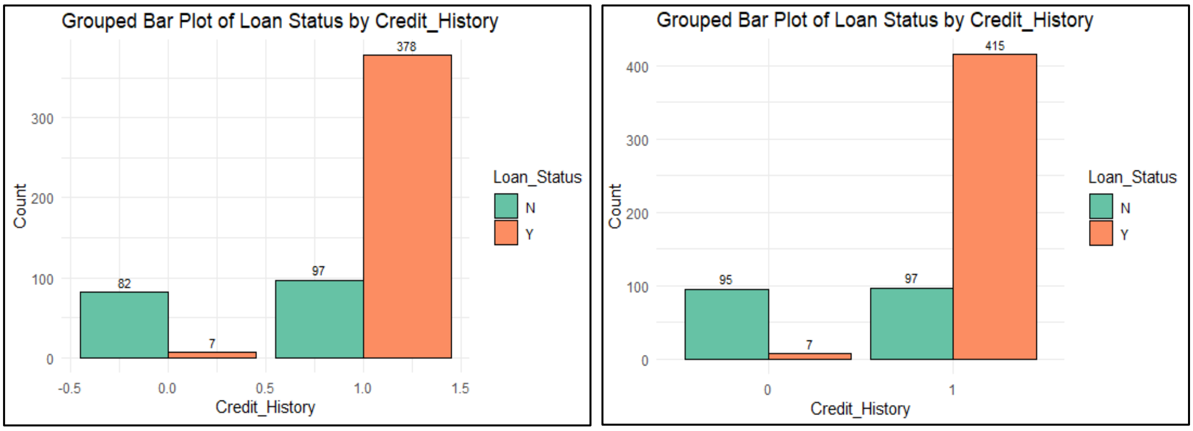
Rural – 133

Semiurban – 128

Urban – 120

For Credit_History, I interpolated nulls with the mode dependant on Loan_Status. They are highly dependent – credit histories of 0 or None are much more likely to have a loan status of No. If I filled on the mode, all would go to 1 which means they have a credit history (code lines 262-267). See Figure 5 below, this shows bar charts of the raw data compared with the clean data – as you can see the raw data reflects the incorrect datatype for Credit_History compared against the data type changed to Factor in the clean data and the change in numbers between the two after the interpolating of the missing values.

Figure 5



Raw

Clean

For the remaining variables; Dependents, Self_Employed, Loan_Amount_Term I filled these with the mode value as these didn't seem to be dependant on anything else (code lines 283-284)

Post cleaning, the data is in much better order and adheres to tidy data principles, there is no missing data, and the data types are appropriate for further exploration and analysis.

At the end of the code some further cleaning was done to remove the applicant and coapplicant income and the original Loan_Amount column as new features were created that will produce multicollinearity which we will want to avoid in the modelling stage.

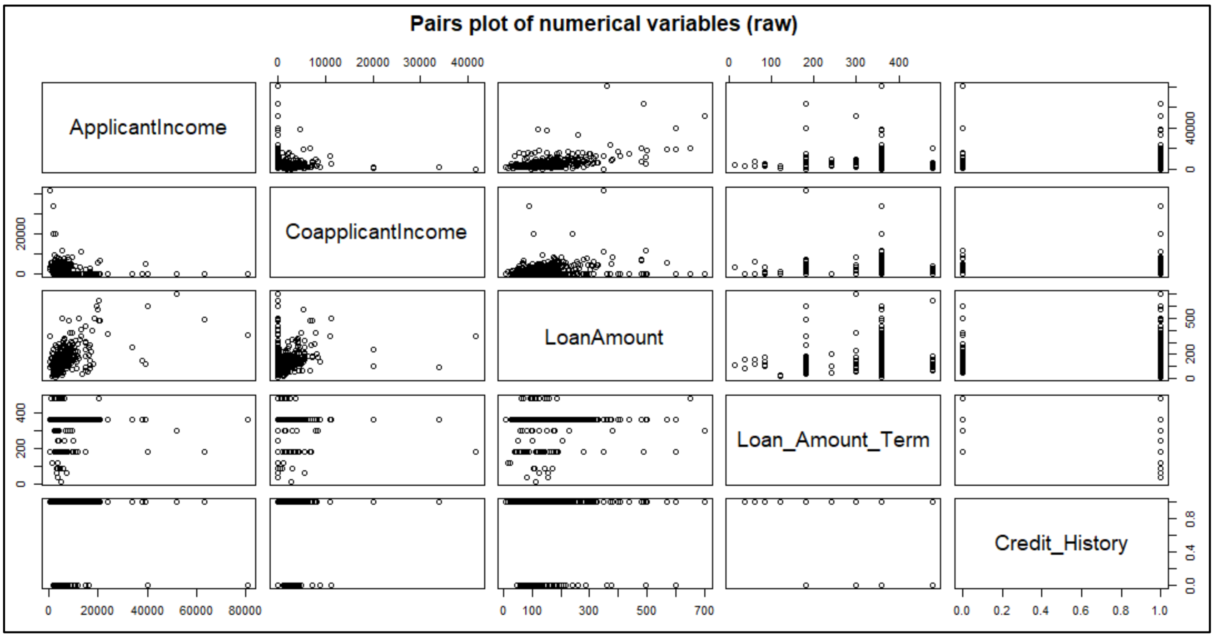
Exploratory analysis

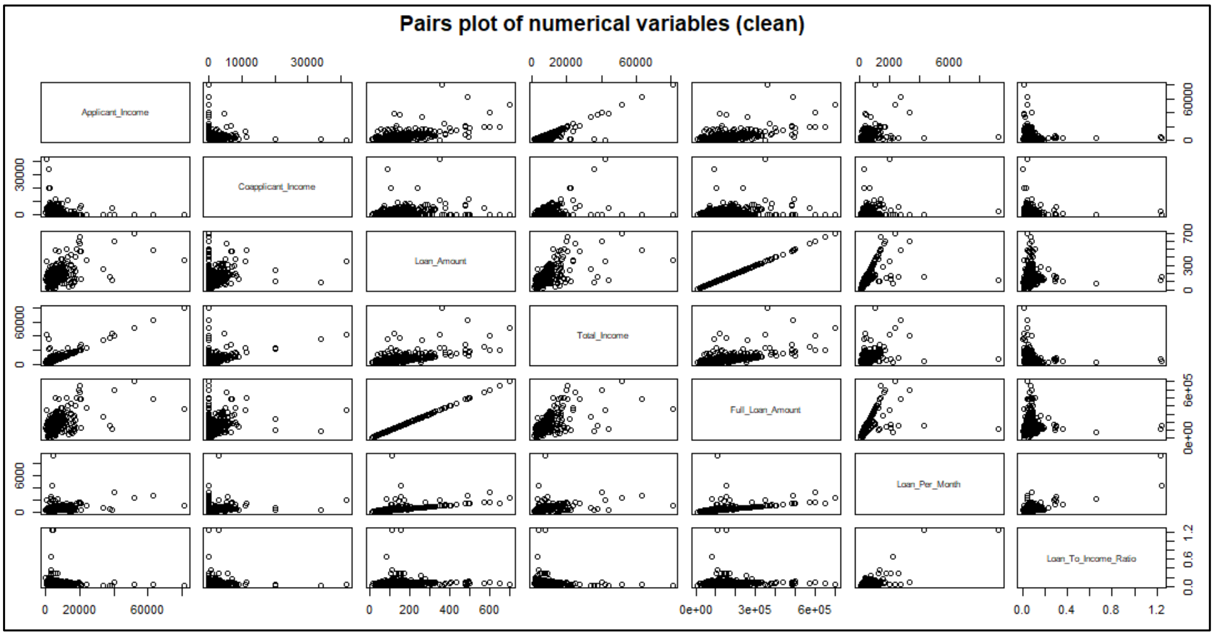
This section describes how I achieved the R4 objective. Before starting the exploration, I did some feature engineering to create new variables that will help with analysis and modelling. Firstly, Total_Income – this combines the Applicant and Coapplicant Income to create a total. My understanding is it will be this figure that will be the basis of any decision rather than individually. Secondly, I created new loan amount variables. The documentation states that

Loan_Amount is in thousands so firstly I created a Full_Loan_Amount variable that multiples Loan_Amount by 1000. I then created a Loan_Per_Month variable by dividing Full_Loan_Amount by Loan_Amount_Term. Then finally, Loan_To_Income_Ratio that divides Loan_Per_Month by Total_Income (total income is a monthly figure in the data) to get the ratio of the household income that is used for the loan each month. My hypothesis is the higher the loan to income ratio is the less likely it is a loan will be approved.

The exploration started with the numerical variables. Firstly, I ran a clean version of the pairs plot. This now shows the new engineered features. The plot highlights a relationship between loan per month and total income suggesting that a higher incomes are associated with higher loan amounts. Loan to income ratio shows a wide spread across income levels which is one to explore further. Figure 6 below shows both the raw and clean pairplots for context. This also highlights the decision to change the loan amount term and credit history variables from numeric to factors.

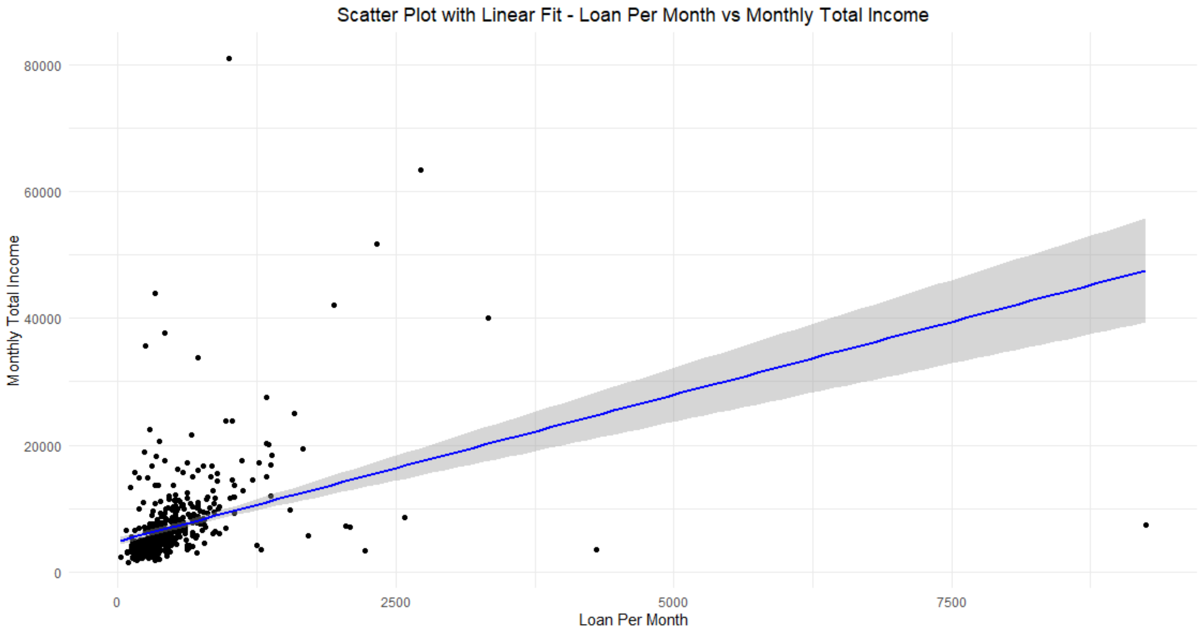
Figure 6





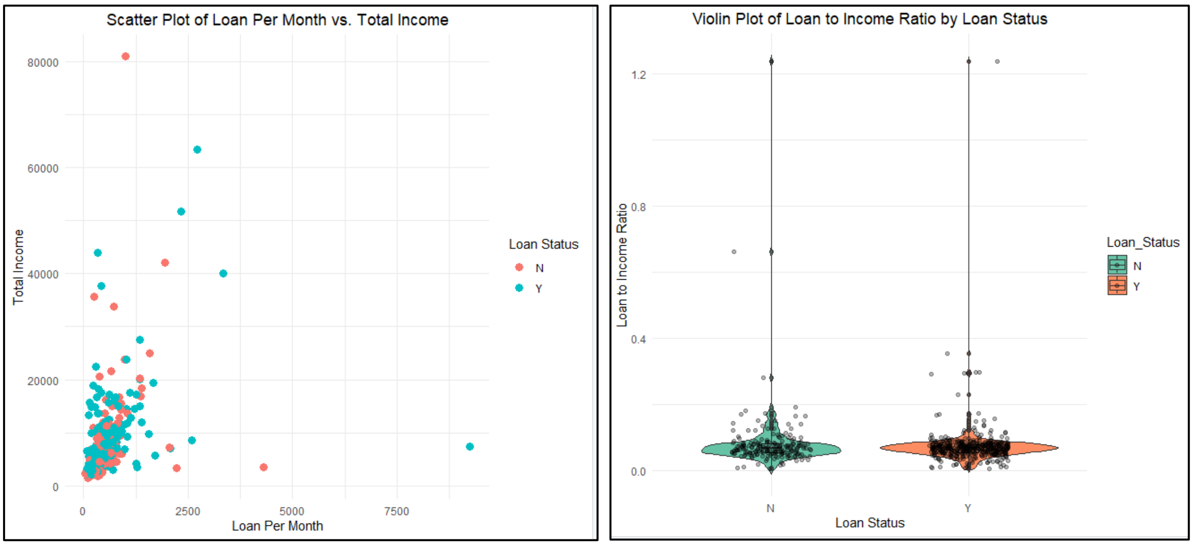
I next created some histograms and performed Shapiro Wilk tests to look at the distribution (code lines 337-362) these show the variables do not follow a normal distribution. I then created a scatter plot with liner fit to show the relationship between Total Income and Loan per month. This shows a linear relationship (Figure 7). The Pearson correlation is 0.37, this shows a moderate positive correlation where one increases the other does too (code lines 365-375).

Figure 7



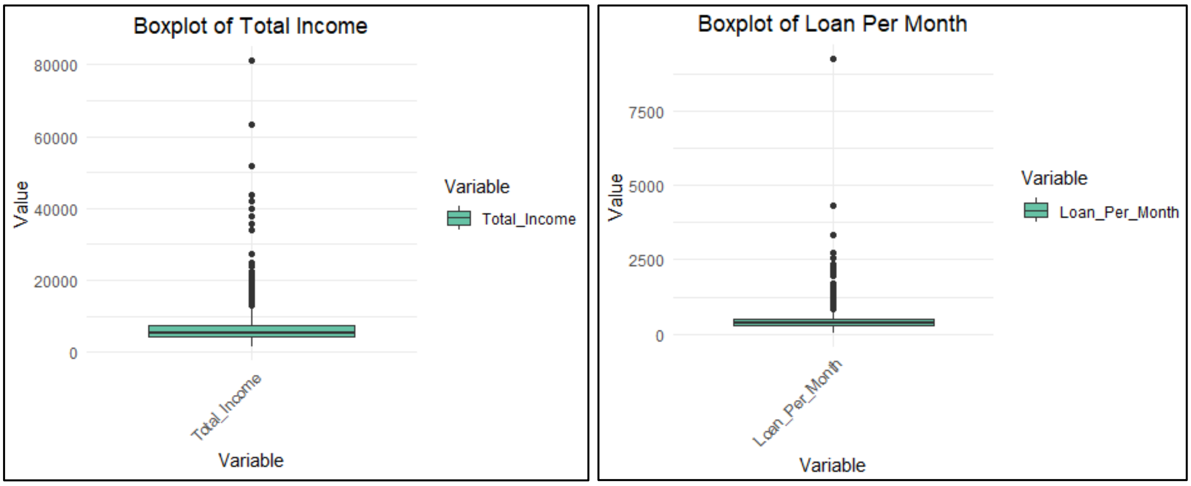
I then looked at loan per month in comparison with total income. See Figure 8. I was surprised here to see these showing loan amount and total income do not seem to have a big influence on loan status. The violin plots show a similar distribution of loan to income ratio with approved and rejected loans. I expected to see a higher loan to income ratio associated with rejected loans. Performing a t-test confirmed the difference in loan to income ratio between approved and rejected loans is not significant (p-value 0.28) although the mean of the rejected group is slightly higher (0.08 vs 0.75). (code lines 389-409 and 451-459).

Figure 8



Finally for the numeric variables I looked at outliers (Figure 9). Both Total Income and Loan Amounts have outliers. These will need to be considered in the modelling stage as some models such as regression models are affected by outliers. However, I don’t believe there is any erroneous data so I did not want to drop any of this data as it could hold some valuable insights (Code lines 418-448)

Figure 9



I next moved onto the categorical variables. I first created functions to return bar plots to show the number of records in each variable and then bar plots to show the distribution between loan status (code lines 469-512). I then performed chi-squared tests to find if there were significant relationships between any of the variables in relation to the target variable. The analysis showed 3 variables with statistical significance which I will focus on here (figures 10,11,12). Firstly, Credit History, there were 102 records with no credit history and 512 with credit history. When compared with Loan Status you can see that no history largely has rejected loans and with history has a much higher proportion of approved loans. The chi-squared test shows a p-value very close to zero which shows it is a highly significant association. The Married variable shows there is a higher proportion of approvals for people that are married than people not married (72% to 63% respectively) and the chi squared test had a p value of 0.034 showing a significant association. Lastly the education variable showed there is a higher proportion of approvals for graduates than non graduates (71% to 61% respectively) and a p value of 0.043 showing a significant association. (Code lines 525-531)

Figure 10

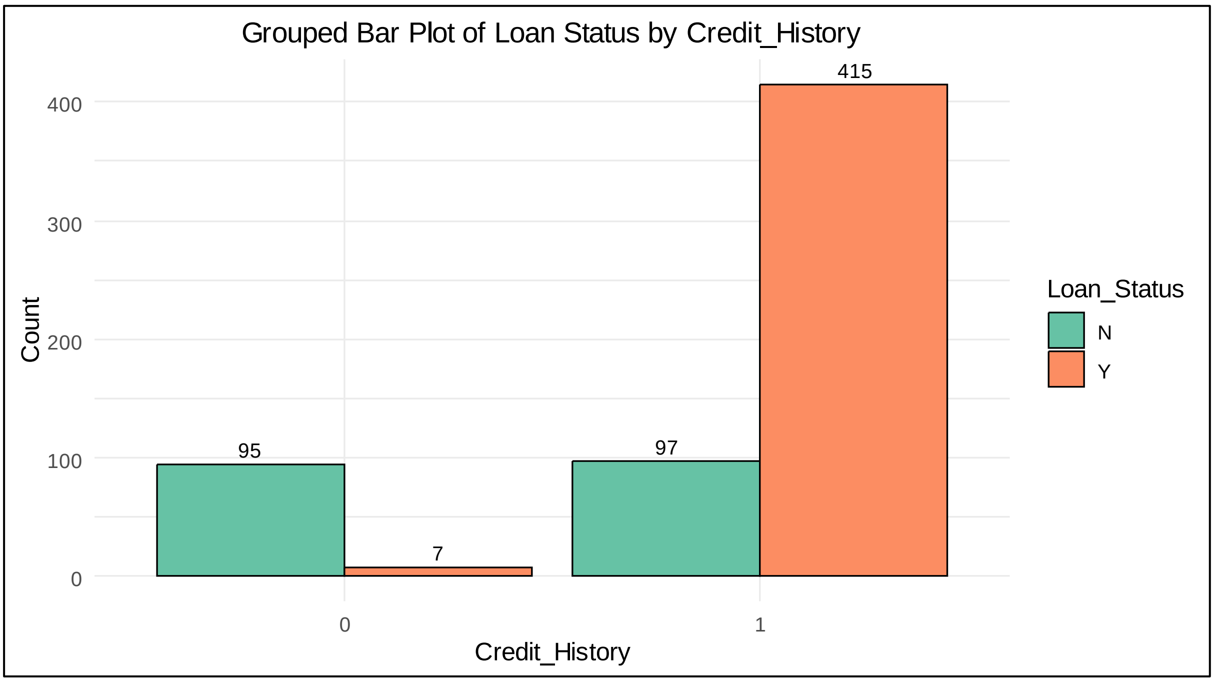


Figure 11

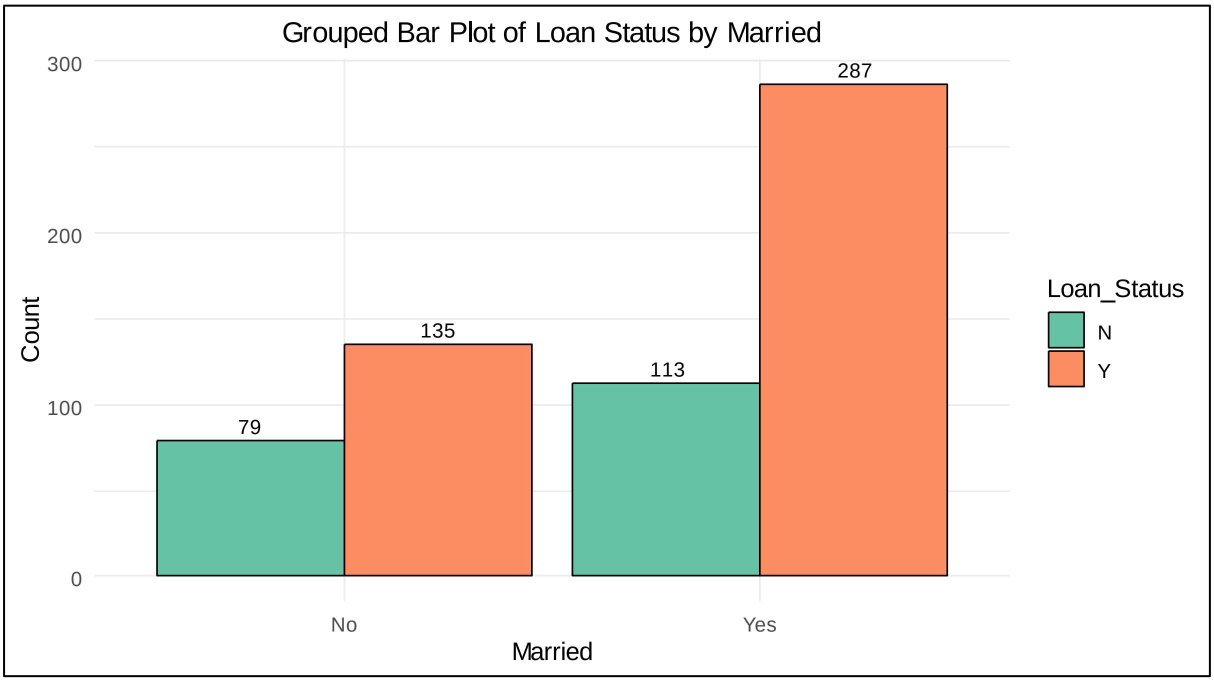
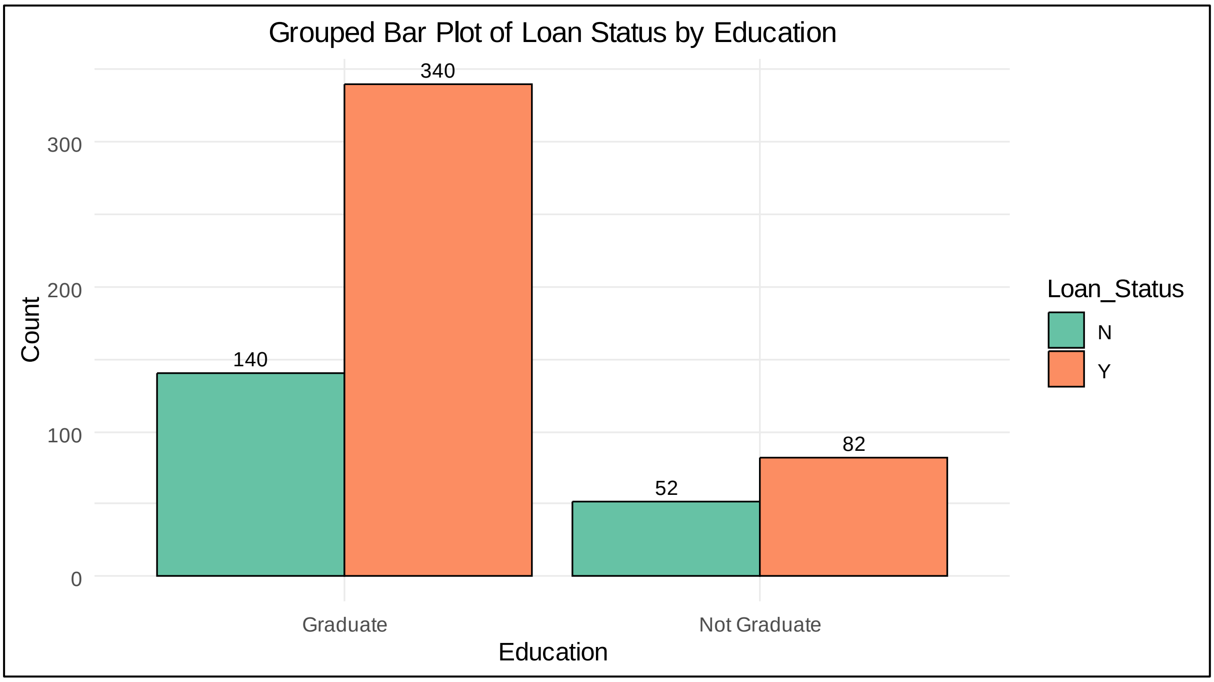


Figure 12



Conclusion

The dataset is now clean with no missing values and data points in the correct scales. In the last bit of my code, I removed some of the fields that could cause multicollinearity in the modelling stage. When it comes to modelling in CW2 depending on the task and which models I choose I may need to encode the categorical variables as numeric to feed into machine learning models but at this stage I decided to leave this as CW2 has not yet been announced and the dataset in its current form is easier for a human to interpret. I have decided to leave the outliers in for a similar reason. When it comes to modelling these may need to be handled dependant on chosen models however, I don't believe these are erroneous outliers so could prove to hold significant information. At the modelling stage the imbalanced nature of the dataset may also need to be addressed with means such as over/under sampling or smote.

From my exploration I discovered there are several key variables that have relationships with the target variable, the top 3 being Credit History, Married and Education. Although there is a relationship between Loan Amount and Total Income, I was surprised to see there wasn't a significant relationship between this and Loan Status. On reflection, more could have been done to explore these and I believe when combined with some of the other key variables it could lead to further key predictors. During modelling I will explore these further.

Word Count: 2279

References

Report

1. Wickham H. Tidy data. J Stat Softw [Internet]. 2014;59(10). Available from: http://dx.doi.org/10.18637/jss.v059.i10

2. What is the difference between a factor and a character data type in R, and why might you choose to use a factor over a character data type? [Internet]. Quora. [cited 2024 Jun 26]. Available from: https://www.quora.com/What-is-the-difference-between-a-factor-and-a-character-data-type-in-R-and-why-might-you-choose-to-use-a-factor-over-a-character-data-type

3. Buchanan D. 5.10 Lecture 3: Tibble [Internet]. Available from: https://learn.london.ac.uk/mod/page/view.php?id=158670

4. Spanton R. Make your R code 10x faster: Vectorization explained in 3 minutes [Internet]. Towards Data Science. 2022 [cited 2024 Jun 26]. Available from: https://towardsdatascience.com/make-your-r-code-10x-faster-vectorization-explained-in-3-minutes-9eb4cdd7a49e

5. Konapure R. Home Loan Approval [Internet]. 2023 [cited 2024 Jun 26]. Available from: https://www.kaggle.com/datasets/rishikeshkonapure/home-loan-approval/data

Code

1. Elegant way to check for missing packages and install them? [Internet]. Stack Overflow. [cited 2024 Jun 26]. Available from: https://stackoverflow.com/questions/4090169/elegant-way-to-check-for-missing-packages-and-install-them

2. Holtz Y. Basic barplot with ggplot2 [Internet]. R-graph-gallery.com. [cited 2024 Jun 26]. Available from: https://r-graph-gallery.com/218-basic-barplots-with-ggplot2.html

3. Bobbitt Z. How to return value from function in R (with examples) [Internet]. Statology. 2022 [cited 2024 Jun 26]. Available from: https://www.statology.org/r-function-return/

4. How to find the statistical mode? [Internet]. Stack Overflow. [cited 2024 Jun 26]. Available from: https://stackoverflow.com/questions/2547402/how-to-find-the-statistical-mode

5. Datacamp.com. [cited 2024 Jun 26]. Available from: https://www.datacamp.com/tutorial/contingency-analysis-r

6. How can I fill missing values based on a condition in R [Internet]. Stack Overflow. [cited 2024 Jun 26]. Available from: https://stackoverflow.com/questions/68804434/how-can-i-fill-missing-values-based-on-a-condition-in-r

7. Getting Median of a Column where value of another Column is 1 in R [Internet]. Stack Overflow. [cited 2024 Jun 26]. Available from: https://stackoverflow.com/questions/17435810/getting-median-of-a-column-where-value-of-another-column-is-1-in-r

8. Clemons A. Finding the mode in R: A step-by-step guide [Internet]. DEV Community. 2024 [cited 2024 Jun 26]. Available from: https://dev.to/rapp2043/finding-the-mode-in-r-a-step-by-step-guide-49h1

9. Bobbitt Z. How to replace NA with mean in dplyr [Internet]. Statology. 2022 [cited 2024 Jun 26]. Available from: https://www.statology.org/dplyr-replace-na-with-mean/

10. Holtz Y. An overview of color names in R [Internet]. R-graph-gallery.com. [cited 2024 Jun 26]. Available from: https://r-graph-gallery.com/42-colors-names.html

11. Holtz Y. Linear model and confidence interval in ggplot2 [Internet]. R-graph-gallery.com. [cited 2024 Jun 26]. Available from: https://r-graph-gallery.com/50-51-52-scatter-plot-with-ggplot2.html

12. Holtz Y. Basic ggplot2 boxplot [Internet]. R-graph-gallery.com. [cited 2024 Jun 26]. Available from: https://r-graph-gallery.com/262-basic-boxplot-with-ggplot2.html