**PKDD’99 Financial Data**

# **Code Description**

The PKDD’99 Financial data spreads out to 8 different datasets, which were cleaned and worked through to arrive at an informative Datamart. The different datasets worked on are as follows:

* **AccDat:** The Accounts dataset was read in as ‘AccDat’ which comprised of 4500 observations and 5 variables and contained information regarding the various accounts present.
* **ClnDat:** The Clients dataset was read in as ‘ClnDat’ which comprised of 5369 observations and 4 variables holding information regarding the clients.
* **CrdDat:** The Cards dataset was read in as ‘CrdDat’ which comprised of 892 observations and 4 variables holding information regarding the Card types.
* **DspDat:** The Disposition dataset was read in as ‘DspDat’ which comprised of 5369 observations and 4 variables holding information regarding the Owners and Disponents.
* **DstDat:** The Districts dataset was read in as ‘DstDat’ which comprised of 77 observations and 16 variables holding information regarding the Districts.
* **OrdDat:** The Orders dataset was read in as ‘OrdDat’ which comprised of 6471 observations and 13 variables holding information regarding the Payment Orders.
* **TrnDat:** The Transactions dataset was read in as ‘TrnDat which comprised of 1056320 observations and 16 variables holding information regarding the Transactions Made.

The first step taken after importing the datasets was to have a quick look at the data and understand what the data looked like. Initial Data screening allows you to have a good idea about the path to choose to clean your data and make it presentable. We made use of the ‘.head( )’ function and the ‘.dtypes’ call to have a broad outlook of our data.

On noticing the various inconsistencies over the data, it was decided to build up on robust functions which would be called on these datasets. These functions used are straightforwardly basic and allow us to efficiently clean up the data that we want. We defined multiple functions, which are as follows,

* **NumDaysSD**: This function was built to calculate the number of days passed since the Start date (A date of reference) which was hard-coded to be ’01-01-1993’.
* **NumDaysED**: This function was built to calculate the number of days passed since the End date (A date of reference) which was hard-coded to be ’01-01-2000’.
* **Language\_Conversion**: The Data sets being of Czech Origin were developed in the Czech language, therefore a language converter function was defined to take care of the values of the Frequency variable in the Account Data.

There was a major inconsistency in the Client Data set where we saw the ‘birth\_number’ variable. This variable was interesting as it was coded in a very unique manner. The middle values of the developed number was of great importance as if the number was beyond 50, it referred to a Female customer otherwise a Male customer. This needed to be taken care of to make it simpler and more informative, we made use of the following functions,

* **ExtractMid**: This function was used to extract the middle values of the ‘birth\_number’.
* **ExtractMonth**: Through the extracted middle values, we were able to use this function to extract the exact month values for our clients to build a proper birth date.
* **ExtractDay**: This was the function used to extract the dates of the month from the ‘birth\_number’ variable.
* **ExtractYear**: Here, we extracted the Year values from the ‘birth\_number’ variable and as these were just 2 numbers, we added ‘1900’ to make the Year value more legible and informative.
* **ExtractGender**: The middle values held more than the month here, therefore it was necessary to apply some more data cleaning techniques on the same. We coded the values greater than 50 to return the value ‘F’ for Females and those below 50 to be ‘M’.
* **DateConverter**: This function took the values from the ExtractDay, ExtractMonth, **ExtractYear** to return a proper DateTime value.
* **BirthDateCalculator**: This function used here, calculated the Age of the customer based on the converted date from DateConverter.

As the data is majorly developed in the Czech Language, we needed to make sure we understand each value written in the data set and to do this we needed it to be in English. Therefore, keeping that objective in mind, we created a few more language conversion functions just like the Language\_Conversion function.

* **OrdKSymConverter**: Converts the ‘K Symbol’ values based on the Orders data set from Czech to English.
* **TrnTypeConverter**: Converts the ‘Type’ values based on the Transactions data set from Czech to English.
* **TrnOprConverter**: Converts the ‘Operations’ values based on the Transactions data set from Czech to English.
* **TrnKSymConverter**: Converts the ‘K Symbol’ values based on the Transactions data set from Czech to English.
* **QMarkConverter**: There was something rather interesting observed on the ‘Districts’ dataset. The unknown values were marked as ‘?’. We used this function to convert the ‘?’ to -1, depending upon the variable formats they were observed in they would return a float value or an integer value.

Recency, Frequency and Monetary Variables allow us to draw more robust and definitive insights, keeping this in context, two functions were built to calculate these variables on selected datasets.

* **TrnVariableBuilder**: Builds the RFM Variables for the Transactions Dataset.
* **OrdVariableBuilder**: Builds the RFM Variables for the Orders Dataset.

After the functions were ready to be deployed we commenced with the data cleaning portion of the project. This is where various variables were renamed to be more informative. Following which we deployed our functions using the ‘.apply( )’ call.

The functions served their purpose and we got our data clean and ready to go further for merging them into a final Datamart. However before we merged everything together, we needed to aggregate the Transactions and Orders datasets to ensure data consistency, i.e. One row per client. This was achieved after creating the RFM variables over these datasets.

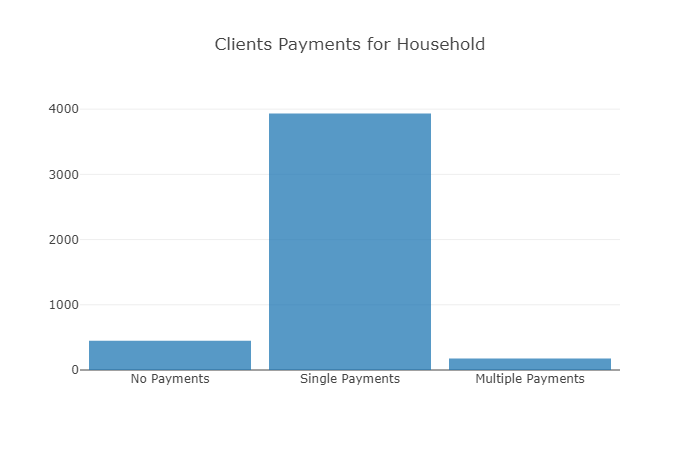
We finally merged all these cleaned and aggregated datasets together using the ‘.merge( )’ call from the Pandas library.

Finally, at the end, we classified the customers based on their loan status to be either Favourable or Non- Favourable. The customers in the ‘LoanStatus’ variable who fell in the A and C categories (i.e. People who have payed their loans) were considered to be favourable while the others were not.

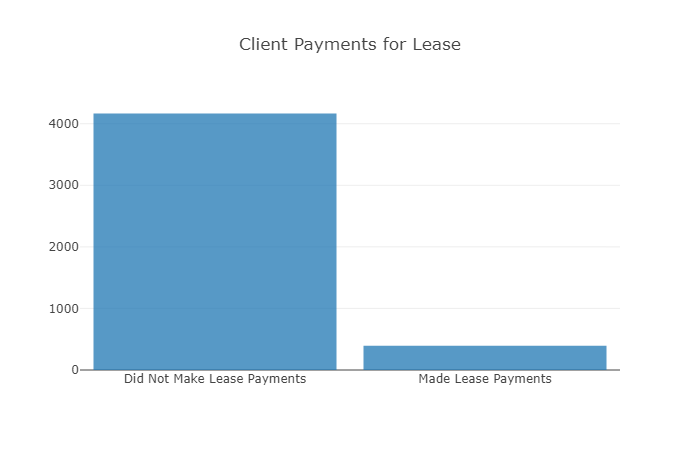
To achieve the final BaseTable we made use of key Python libraries/packages, which were,

* Pandas
* DateTime
* Plotly
* Plotly.Graph\_Objs

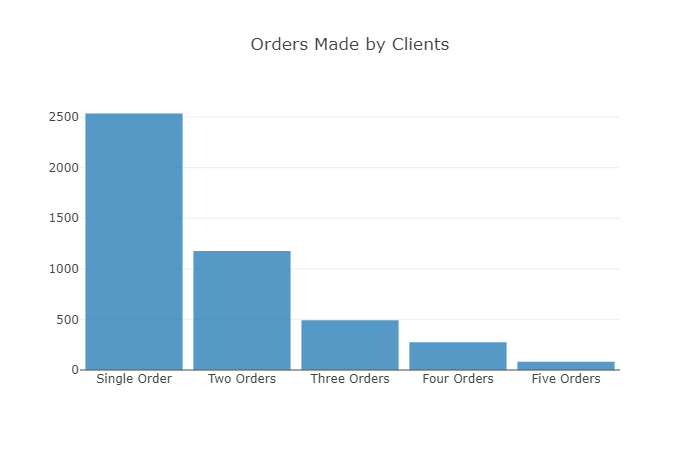
# **Plots Descriptions**



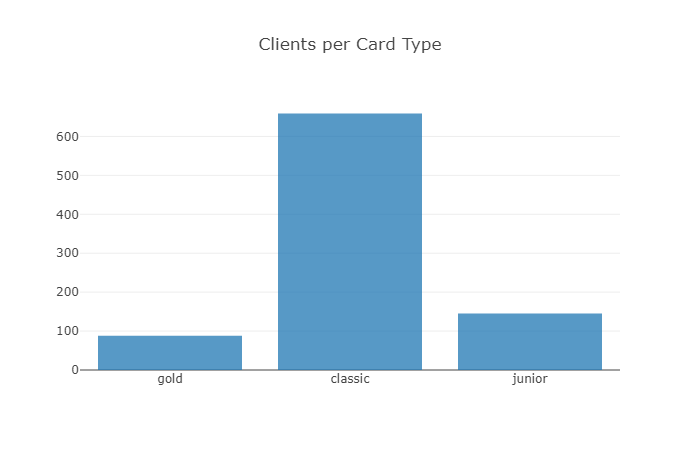
The plot of **Clients Payments for Household** reveals that the majority of the clients just used housing transaction once so these group of customers shouldn’t be the investors, actually the clients who did this kind of transaction more than once can be real investors in real estate industry.



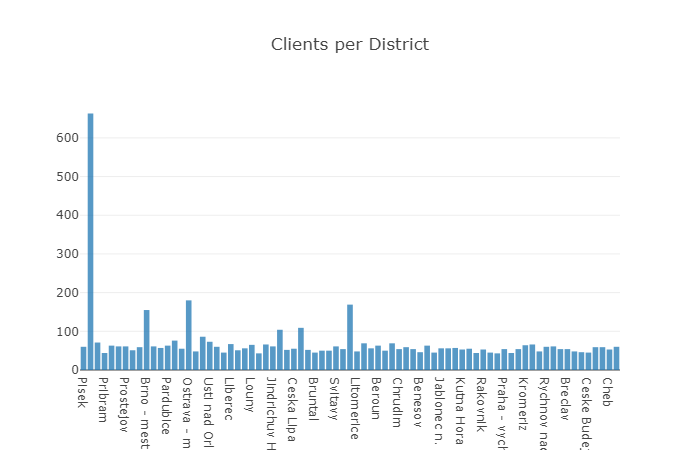
The plot **Client Payments for Lease** shows that most of the clients didn’t make any lease payment during the survey duration.

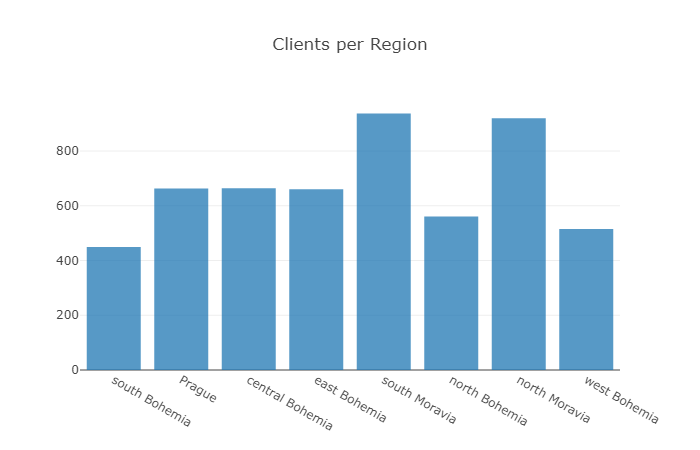


The plot **Order Made by Clients** shows the number of times the customers got other services namely, housing, lease , and insurance .Then based on the plot the about 2500 of whole customers used these services just once and others used more than once. Also, the maximum number of times that clients used these services is 5 times.

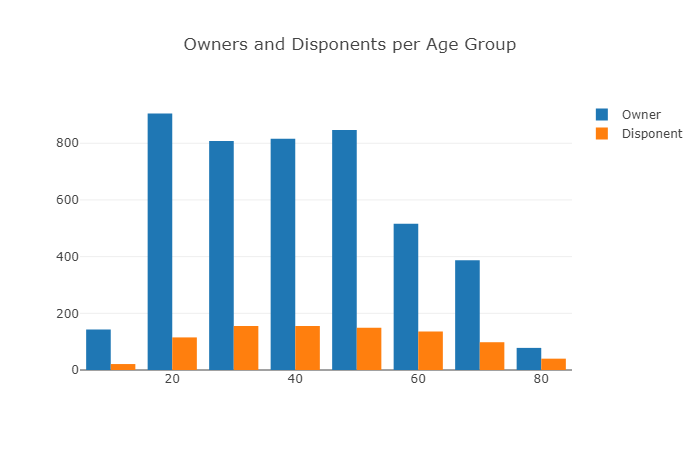


The plot **Clients per Card Type** clearly reveals that most of the customers are interested to classic type of the bank card. Also the 2nd most popular card is Junior one.

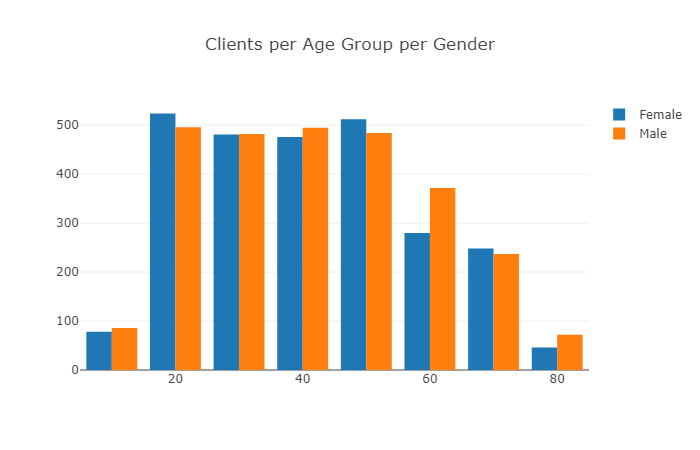




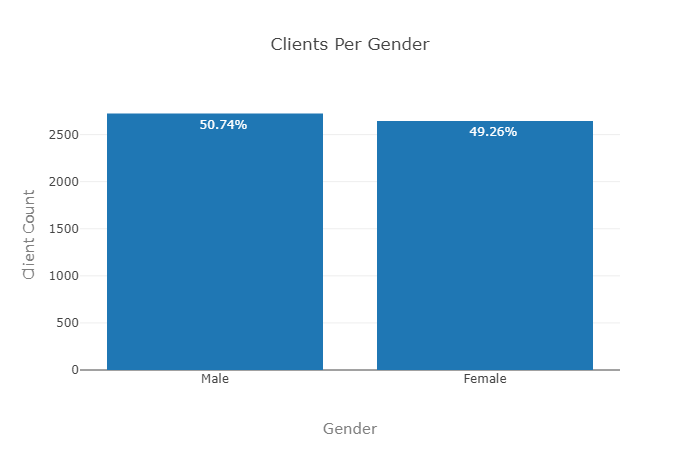
In these 2 plots **Clients per District and Clients per Region**, we aim to show the distribution of the current customers based on the region and the district that they live. Having said that, the highest number of customers come from Moravia. Also the number of clients who live in Prague, central Bohemia, and east Bohemia are almost the same. In addition the lowest number of customers are related to South Bohemia region. The Hl.m.Praha represents the highest number of customers among all districts that has been surveyed. This information will be helpful for segmenting the market.



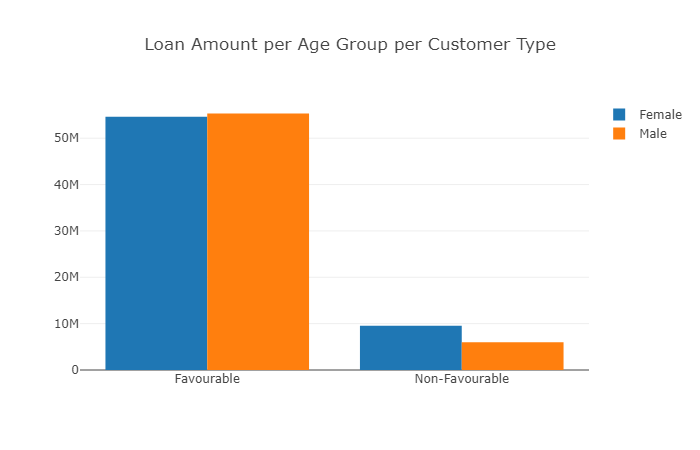
Based on our research (**plot Owners and Disponents per Age group**), Minors above the age of 10 years may be allowed to open and operate savings bank accounts independently, if they so desire. So the ‘Owner and Disponents per age group’ plot shows the number of bank accounts that are being managed by either owners or Disponents. Having information about this two types of bank accounts, bank can segment their market and generate marketing campaigns more effectively because they know that there are some influencers(Disponents) who affect customers'(owner) behavior and sometimes Disponents can even play a role as a decision maker.



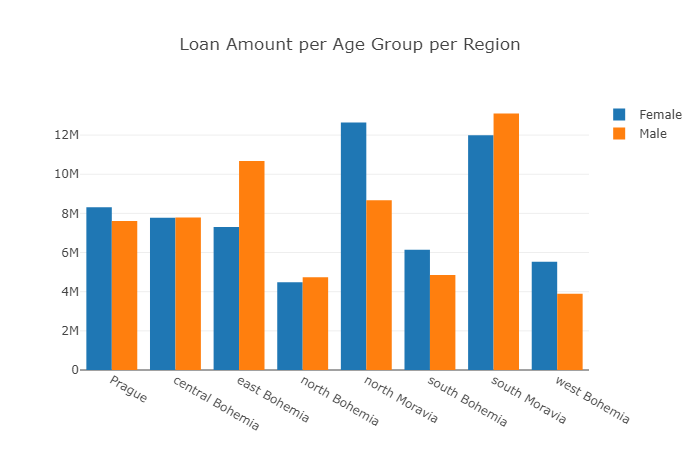
The main objective of creating plot **Client per Age per Gender** is to show that the majority of our current customers are between 20 to 50.Then it can one of the main criteria that the bank needs to segment its market better.



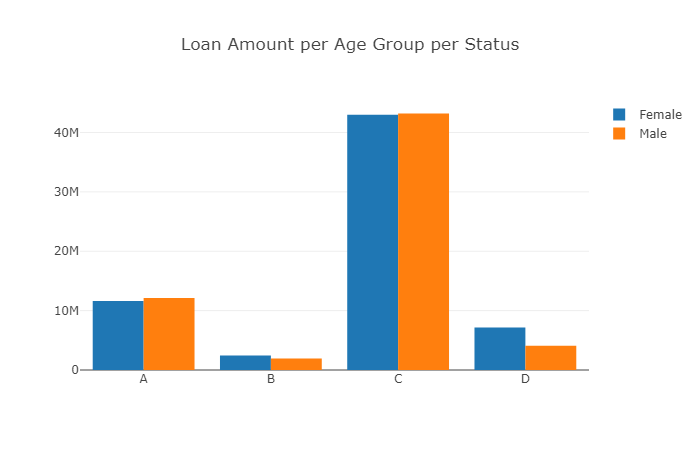
Gaining an understanding of how gender differences influence purchase decisions and recognizing gender-specific tendencies is important for any business that give services to people – and wants to do so more effectively.Having said that, plot number 1 reveals that the bank could successfully attract almost the same number of Male and Female customers with 51 and 49 percent respectively so it will be great to try to preserve this approach and use it as a competitive advantage in the market (Plot **Clients per Gender**).

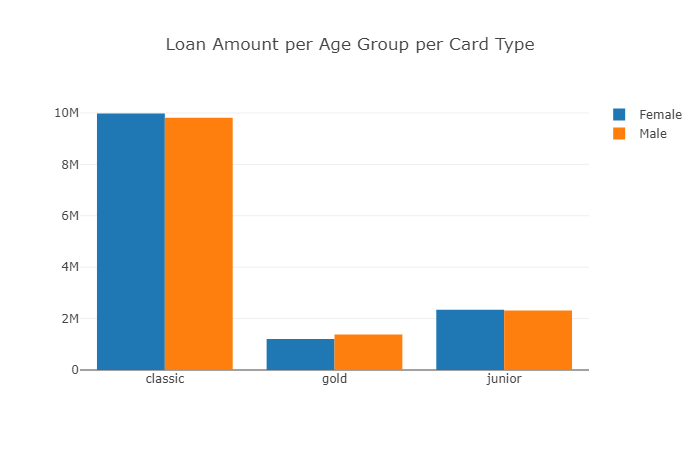


The plot of loan favourability reveals that the majority of the clients paid back their loans .in addition, although there isn’t any meaningful difference between male and female customers who paid back their loans, female are in the majority for those who didn’t pay back. (Plot **Loan Amount per Age Group per Customer Type**).

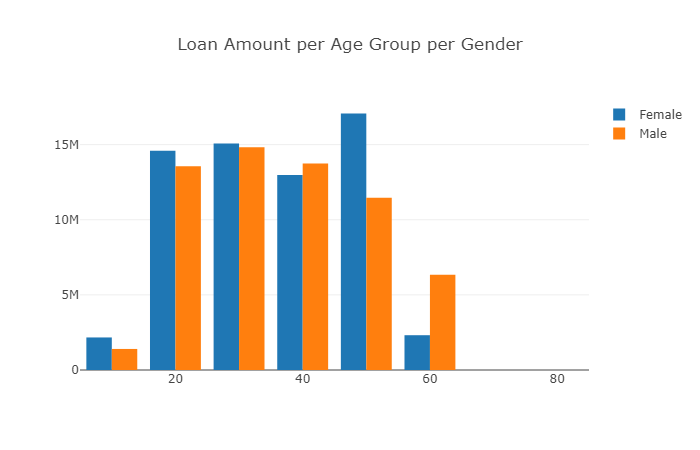


The plots **Loan Amount per Age Group per Region** and **Loan Amount per Age Group per Status** explain 2 main points, firstly except south Moravia and east Bohemia, at the other regions men borrowed more amount of money than women. Secondly, our customers in 2 regions namely, north Moravia and south Moravia got more amount of loans. It may refer to the scope of businesses that are being held there or the number of clients who live in those regions.

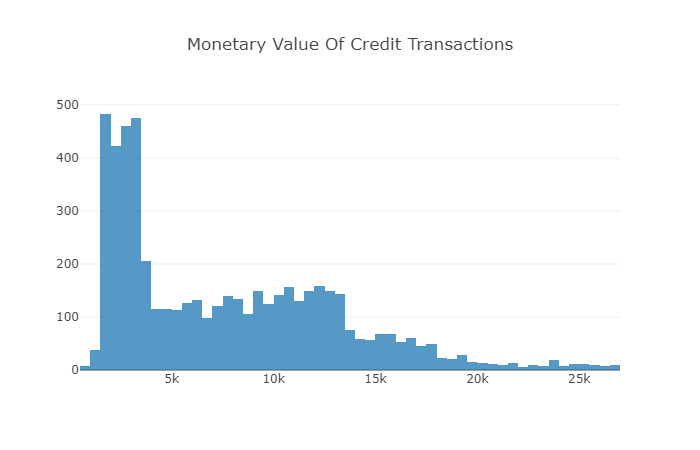




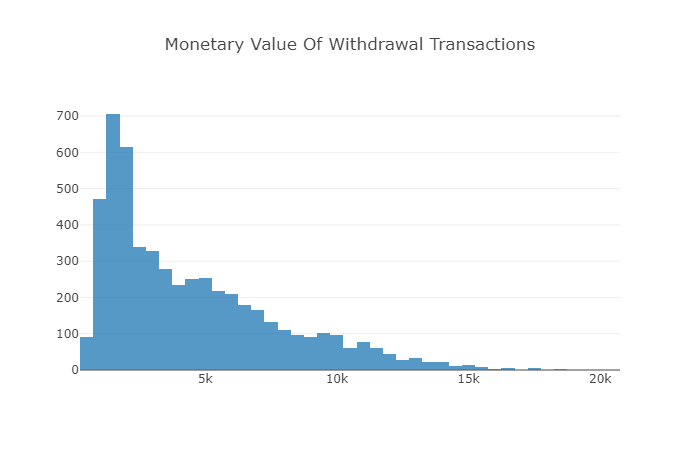
It can be clearly seen that the clients who use classic cards intended to borrow more amount of money in comparison to others (plot **Loan Amount per Age per Card Type**).



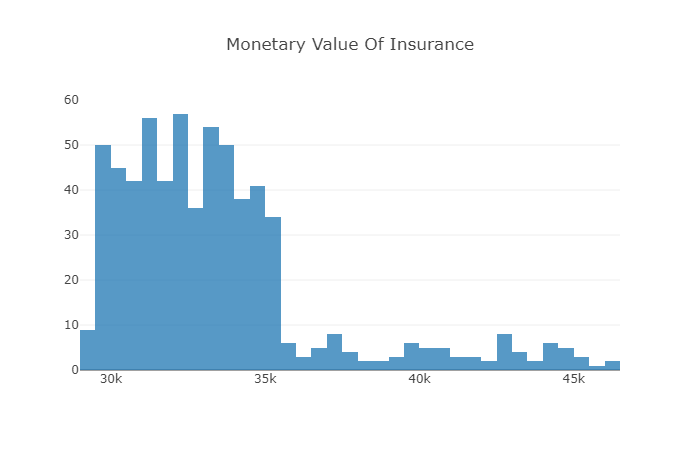
It shows that men who are around 50 got the most amount of money from the bank though 30 years old women borrowed the highest amount in comparison to other female age groups (plot **Loan Amount per Age Group per Gender**).



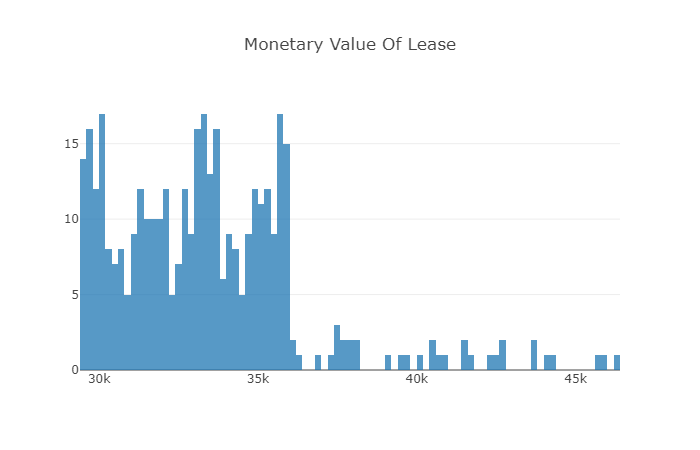
The histogram **Monetary Value of Credit Transactions** shows that most of the bank clients have less than 5000 euro transaction during whole surveyed years.



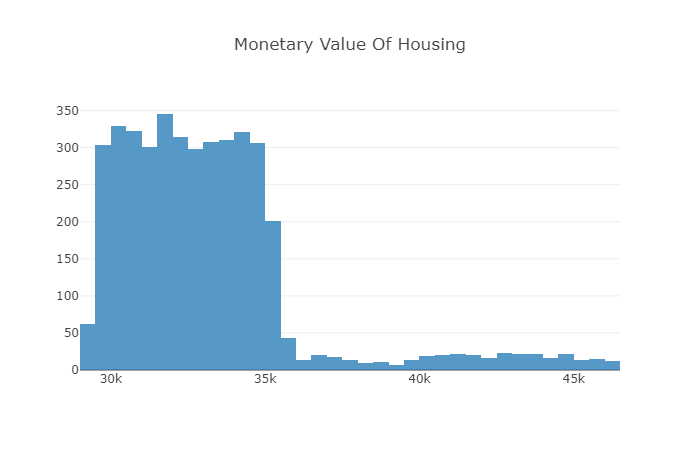
The histogram **Monetary Value of Withdrawal Transactions** obviously shows that most of the customers draw less than 5000 euros during whole years of this survey.it can be affected by country’s law and regulations related to maximum amount of money that people can draw daily and it reveals that people intend to use other kind of transactions through e-banking instead of drawing large amount of money.



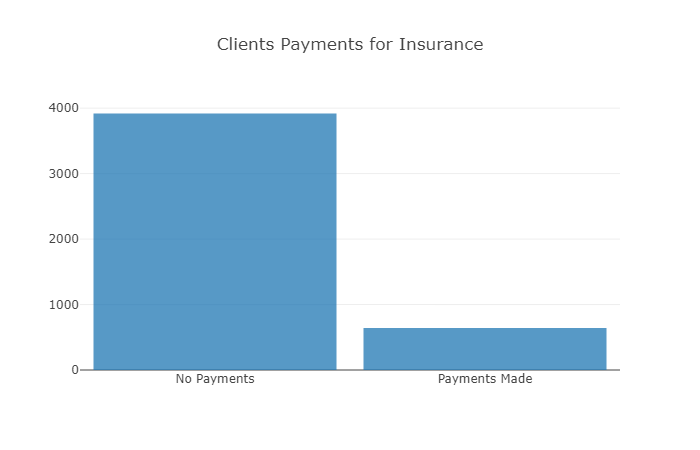
It reveals that most of the customers paid between 30k to 35k just for insurance (plot **Monetary Value of Insurance**)



It reveals that most of the customers paid between 30k to 35k just for lease (**plot Monetary Value of Lease**).



It reveals that most of the customers paid between 30k to 35k just for housing (plot Monetary Value of Housing).



It shows that just less than 1000 of the customers paid for insurance policies (plot **Client Payments for Insurance**).