In this module, we'll talk about how we execute data exploration and preparation tasks inside R. To demo the process of data exploration and preparation, we'll be looking at a credit card dataset. Let us, first of all, understand what are the variables present in credit card dataset by looking at the data dictionary for this particular dataset.

This is the data dictionary for the credit card dataset that we'll be using throughout this session. As you can see, since this is a credit card dataset, this is a financial dataset; we have some financial variables like Debt Ratio, monthly income. We also have some demographic variables like gender, region and occupation. Now the variable of interest for us would be NPA status. This variable talks about if a person has been delinquent in the past 180 days or not. Essentially, it gives an idea of who are my good customers and who are my bad customers.

So the person who has been delinquent in the past 180 days would be a bad customer for me and the person who has not been delinquent in the past 180 days would be a good customer for me. Let us now read the dataset inside R.

Since we will need to manipulate the data a lot, I have also loaded the library (dplyer). This command here *options(scipen=999)* switches off the scientific notation in terms of number representation. As far as the process of data exploration and preparation is concerned, we will be touching upon these particular topics. We will

look at how we take a stock of the quality of the data by doing a sanity check. We'll see how we go about doing a sanity check. We'll also see how we identify outliers by generating summary statistics. We'll take a look at some approaches to impute missing values in a data. We'll also learn how to Bin continuous variables in a dataset using the *ntile* function and use the binning process as an approach towards missing value imputation. We'll also see how we partition a data into test and training samples.

The first thing that we will do as far as sanity checks are concerned is to look at the names of the columns that have been read in. So I will use the *names* command to display the names of the column in the dataset that I just read in. Now an interesting thing that I see here is that the 12th column has name called "MonthlyIncome.1" and the 6th column has the name "MonthlyIncome". So probably these two columns have the same values and they are duplicate columns and I might need to verify that. Also if I take a look at the names of the columns, the first column, I see the first column as "NPA.Status" and last column is "Good_Bad" column.

>>

If I go back to my data dictionary, I see that I don't have a column called "Good_Bad" in my dataset. So the original dataset that was given to me did not have the column called "Good_Bad". The dataset that I have just read in might have been manipulated before it reached me and

someone has created a new variable inside my dataset and I need to take a stock of these new variables, which have been created.

>>

One way to do that is to take a look at the dataset that I have just read in and the way to do that is by clicking on the dataset in the environment window. Let us first of all take a stock of NPA.Status and Good_Bad variables. As I can see the NPA.Status variable has values like 1,0,0 and the Good_Bad variable has values like Bad, Good, Good, Bad. Essentially, wherever I have 1 in NPA.Status variable, I have bad in my Good_Bad variable. So both of these columns are giving me same information so I will need to take one of my columns out of my dataset before I proceed further.

Also if I take a look at the MonthlyIncome.1 column, it has these values and if I take a look at the MonthlyIncome column, it has these values which seem to be similar. Let me go back to my Data dictionary and see if I have a column called “MonthlyIncome.1”. As you can see there is just one column which talks about Monthly Income and the name of that column is “MonthlyIncome”. So there is no column called “MonthlyIncome.1” as far as the data dictionary is concerned. So I can conclude that these two are duplicate columns so my data was manipulated in some way before it reached me. So I’ll need take this variable out from my dataset.

The way I would take this variable out is by using this syntax – (-c(1,12)), because NPA.Status is the first column and MonthlyIncome.1 is the 12th column. So I am removing the duplicate variables out of my dataset. Now as you can see, the number of columns has reduced 18 to 16.

Let us look at the summary statistics of all the columns in my dataset by using the ***Summary*** command. Summary command produces 5 point summary statistics for numerical variable and frequency distribution for categorical or factor variables. Now one thing that I notice from the summary statistics is that in my good bad column, I have two missing values. Whenever we have credit card kind of dataset where we have information about people who have defaulted and information about people who have not defaulted, the aim is to build a model to identify good and bad customers. This Good_Bad variable will become a dependent variable if I have to build a model on it.

Now, in my dependent variable, there are two missing values. So I'll need to take these two missing values out of my dataset. The way to do that is first of all identify, for which rows, I have missing values in the Good_Bad column and for that I'll use this command – (which(is.na)) of my dependent variable which has missing values. So the index object here is going to store the row number where I have missing values in the Good_Bad column. I will take these rows out of my data by using the minus symbol in the row index for the data frame cr.

>>

Now as you can see I have two less observations now; I have now 1,50,000 observations. Now what we will do is we will run individual summary for all the variables that we have in our dataset.

Let me first of all clear the console by pressing CTRL + L. Let us first of all take a look at the variable RevolvingUtilizationOfUnsecuredLines. Now let us go back to the dictionary to see what this variable actually signifies. As you can see RevolvingUtilizationOfUnsecuredLines is a percentage variable and this is what it means. If I go back to summary statistics, I see something odd here. Now this is ratio variable or a percentage variable and look at the maximum value that it has. Also take a look at the mean value that it has. I should not have such high numbers for variables, which represent percentages or ratios. I need to investigate further.

I will do a simple check on this variable and figure out how many observations are zero and how many observations have really large magnitude. So I will be using this simple dplyer syntax here, so this syntax here tells me how many rows I have in my data for which this variable has a value of zero. I can see there are around 10,000 rows in my data, which have value of zero for RevolvingUtilizationOfUnsecuredLines. Also since this is ratio variable or percentage variable, ideally its value should not be more than 1 or 2. So Let us take a look at how many observations in my dataset corresponding to this dataset have a

magnitude greater than 0.99. Again 14,000 observations have a value greater than 0.99.

Another thing that we can do it, we can do a percentile wise break up for this variable and take a look at how this percentile break up for this particular variable looks like. I will use a quantile command here. What this command does is, it tells me what percentage of my data has a particular magnitude. Here I see 99% of my data has a magnitude less than equal to 1.09 and there are just 1% observations, which have magnitude greater than this number. What I will do is, I will go back to my stakeholder and have a discussion with my holder and tell him that this is what I found out when I explored your data. So he might give me an input that "Usually this variable never has a magnitude greater than 2". So based on this business knowledge that has been provided by my stakeholder, I will cap the values of this particular variable.

So first of all I will see how many rows in my dataset I have which have a magnitude which less than or equal to 2. There are 1,49,000 rows and the next thing I'm going to do is to filter out those observations from my data whose value is more than 2 for this particular variable. As you can see, I have reduced number of rows in my dataset, because I have taken out the values in my dataset for this particular variable whose magnitude is more than 2.

This is one of the ways we take a look at the quality of a dataset by just observing the summary

statistics. Then we go back to the stakeholder and ask them if the numbers make sense. Basically, they provide us with the business context and tell us where we can cap this variable and where should not cap this variable.

Let us take another example. Let us talk about the age variable. Before we talk about the Age variable, let us clear the console using CTRL + L. Let us take a look at the 5 point summary for the age variable. If I look at the age variable, another odd thing I notice here is that the minimum value for age variable is 0. This would imply that in my dataset, I have some people whose age is 0 and they still have a credit card. Now this does not make sense. It would mean that someone who is not even adult and he still has a credit card, which should not be the case.

So I will first of all find out how many observations in my data have a magnitude of 0. So I will use this dplyer syntax. What it does is it tells me how many rows in my data have a magnitude of 0. I see there is just 1 row. This could have been a data entry error and I can neglect this single observation. But, before I proceed further, I can do quantile wise analysis and see how the distribution of the age variable looks like. As you can see the first 1% of the observations are less than or equal to 24, which makes sense. And the maximum is 109; again I don't see anything really odd here. The only odd thing was that I have one observation in my dataset whose magnitude is 0. So I will filter out those rows where I have a value of 0 for Age

variable in my dataset using this syntax. Now you can see now I have 1 row less in my data.

This is how we proceed ahead with Sanity check as well as identifying if there is something anomalous in a data and treating it by consulting with the stakeholder who has given us the data.

In this section, we have taken a look at how we do sanity checks inside R. The way to do Sanity checks is to produce 5 Point summaries for numeric variables and frequency distributions for categorical variables. If we find something anomalous in our summaries, then what we do is we either go back to a stakeholder or bring in business context to treat those anomalous values.

>>