Group Project

BUAN 6356
Members: Anshumali Shrivastava, Durga Rao, Praval Godre, Tushar Kumar, Dennis Black
November 29, 2018

Table of Contents

Table of Figures

Table of Tables

Executive Summary

Group 4 met multiple times in September 2018 examining the various websites to see which data may work for the group project, and some of the websites that group members examined were KDnuggets, Kaggle, and data.world are some of the sites, and in this document are embedded files that show some of the data sets examined, for example Telecom data and Right Whale data as example. The impetus for the project was to enhance the group members resume where the project would resonate with a hiring manager in a model development, data science or data analysis department, and to this end, given in the DFW area, Capital One, JP Morgan, Citi and Comerica Bank, have strong analytics department group 4 decided on searching for bank data which, and with some effort Universal Bank data was found on a machine learning website, but unfortunately, it is data used in the course! With a strong effort, members found bank data from the Czech Republic, that is real bank data, disguised for privacy purposes, and was gathered and organized by Petra Berka of the University of Economics, and in the spirt of the exercise, the group decided on a proof of concept for an assumed company North Texas Econometrics that had time series expertise, but lacked data mining experience and had an opportunity to contract with a bank marketing department to run marketing campaigns and possibly build a probability of default model.

Group 4 analysts downloaded the data, organized the data, and verified the data, and the verification process turned on a maker-checker program where the analysts input the data separately, and compared their results, and followed the rubric of the Shmuelit data mining book.

After the data was organized, described, verified, analysts examined visual and tabular aspects of the data, looking at histograms, boxplots, and tables, the transaction data balance histogram shows a distribution skewed to the right with a large frequency at the median of 38,291 K

Now analysts examined cluster analysis looking at the mean number of inhabitants, mean number of municipalities in the district, and the bad-good dummy variable, and discovered that Prague separated from the other regions, and this is associated with the fact the small municipalities are juxtaposed to the mean number of inhabitants. Management should look at Prague from a population point of view as different than other regions. Analyst also looked at the bad-good dummy variable cluster over regions and found that North Bohemia is a high-risk area and underwriting procedures need to be reviewed, and in the intermediate range the south Moravia, Prague, central Bohemia and east Bohemia all cluster together in an intermediate range of risk, and for the low risk area analysts discovered west Bohemia, north Moravia and south Bohemia.

Next to discover any associative rules in the transaction data analysts looked at the Apriori method to discover an association in the 1 million plus transaction entries, and found, with a lift over 2.25 for the first four rules.

	lhs <fctr>	<fctr>	rhs <fctr>	support <dbl>	confidence <dbl>	lift <dbl>	count <dbl>
[1]	{type.xVYDAJ,bankNone,typeNone}	=>	{operationVYBER}	0.3259479	0.9969266	2.431194	329240
[2]	{type.xVYDAJ,bankNone}	=>	{operationVYBER}	0.3945867	0.9815473	2.393688	398572
[3]	{type.xVYDAJ,k_symbolNone,bankNone}	=>	{operationVYBER}	0.2425473	0.9703274	2.366327	244997
[4]	{type.xVYDAJ,k_symbolNone}	=>	{operationVYBER}	0.2425473	0.9383766	2.288409	244997

Notice that these four rules VYDAJ (transaction withdrawal) implies VYBER (which is a cash withdrawal) and this is an opportunity for the bank to explore these customers and move them to a credit card position to and obtain the associated fees.

Next analysts examined CART, which is a distribution free procedure, that allows for missing values and outliers, and with the first run the sensitivity value was low (note the procedure models 0), and modelers went with oversampling that yielded a reasonable sensitivity value of 0.825, and consequently the confusion matrix was adjusted for oversampling, and this model is a candidate model to recommend to North Texas Econometrics.

The logistic regression procedure was accomplished using a variable selection methodology of all possible regression, and the cp value was employed to select the number of variables. The p-value was examined and used to further select variable, along with r- square and adjusted r-square graphs with also guided analysts to the proper number of model variables. The lift chart, decile chart and confusion matrix all show reasonable model fit, and modelers felt the logistic regression played well with the data.

LDA was examined but the sensitive result was only approximately 6%, and was not pursued at this time.

Finally, Group 4 modeler feel that CART and Logistic Regression are competitors for best fit on the data with each procedure having positive characteristics the model the data. The CART methodology can be employed to classify obligors as to good and bad loan prospects, and the logistic regression had material lift that showed promise for analyzing a binary response variable.

Group 4 recommends to North Texas Econometric to contract with the bank to further explore CART and logistic regression as method for loan classification and marketing campaign response lift.

Introduction

The objective of this project is to create a product that resonates with the BUAN 6356 professor teaching the course, and a project that group members are able to list on their resumes which exhibits their knowledge, skills and abilities to prospective employers. Team members will be able to describe the process of obtaining data, documenting the data, organizing the data, preparing the data for analysis, performing exploratory data analysis, choosing the appropriate methodology to model the data, documenting the model building process and creating a presentation to communicate the results to senior management.

Group 4 team members are following the model building process as shown in Shmueli, Bruce, Yahav, Patel, and Lichtendahl Jr., Data Mining for Business Analytics: Concepts, Techniques, and Applications in R, Wiley, 2018, p. 39.

1. Determine the purpose.
2. Obtain the data.
3. Explore, clean, and preprocess the data.
4. Reduce the data dimension (if needed).
5. Determine the data mining task.
6. Partition the data (for supervised tasks)
7. Choose the technique.
8. Use the algorithm to perform the task.
9. Interpret the results.

Determine the Purpose

The purpose of this project is twofold, first, to fulfill the grading requirements enumerated in the *Grading Rubric for Group Project* directions sheet, and second, to showcase the knowledge, skills and abilities of the group members such that a potential employer would obtain a favorable light of a project members application for employment.

The statement of purpose then from a BUAN 6356 perspective is as follows:

a. Write a written report containing:
    i. Executive summary
    ii. Background
    iii. Objectives
    iv. Data exploration
    v. Predictive model or classification algorithm
    vi. Results section
    vii. Conclusion
    viii. Marketing take away
    ix. References

b. Initial data exploration is reported to identify unusual observations and patterns. Data and dimension reduction considerations.

c. A thorough explanation of the algorithm including a description for senior management.

d. A well thought out results section.

e. A professional quality report.

Obtain the data.

Group 4 members have, individually and corporately, searched the Internet for a suitable data mining dataset, and created a list of potential data sets from the Kaggle website, and a few of the datasets are listed in the following embedded files.

Kaggle 17 data mining competitions.c

Group project data search.docx

Insurance data, education data, KDD competitions data, employee churn data, Right Whale identification data among other datasets were downloaded and examined.

Group 4 met a number of times in September 2018 and October 2018 to discuss data selection, and explored and decided on Universal Bank, located in Culver City, California, however, Group 4 discovered in class that the Universal Bank data was actually an example in the course's data mining book, and discussion with the course professor, Group 4, decided on another round for data exploration.

Bank data prepared by Professor Petra Berka of the University of Economics in Prague for a KDD 1999 competition, and the data had characteristics Group 4 members felt would be an excellent dataset for the group project, such as, bank data with over a 1,000,000 records and loan characteristic information including default. A description of the data is at the following website:

https://sorry.vse.cz/~berka/challenge/pkdd1999/berka.htm

Financial Data description_Petra_Berl

The description of the data from Professor Berka follows follows:

The data about the clients and their accounts consist of following relations:

- relation account (**4500** objects in the file ACCOUNT.ASC) - each record describes static characteristics of an account,
- relation client **(5369** objects in the file CLIENT.ASC) - each record describes characteristics of a client,
- relation disposition **(5369** objects in the file DISP.ASC) - each record relates together a client with an account,
- relation permanent order (**6471** objects in the file ORDER.ASC) - each record describes characteristics of a payment order,
- relation transaction (**1056320** objects in the file TRANS.ASC) - each record describes one transaction on an account,
- relation loan (**682** objects in the file LOAN.ASC) - each record describes a loan granted for a given account,
- relation credit card (**892** objects in the file CARD.ASC) - each record describes a credit card issued to an account,
- relation demographic data (**77** objects in the file DISTRICT.ASC) - each record describes demographic characteristics of a district.

Each account has both static characteristics (e.g. date of creation, address of the branch) given in relation "account" and dynamic characteristics (e.g. payments debited or credited, balances) given in relations "permanent order" and "transaction". Relation "client" describes characteristics of persons who can manipulate with the accounts. One client can have more accounts, more clients can manipulate with single account; clients and accounts are related together in relation "disposition". Relations "loan" and "credit card" describe some services which the bank offers to its clients; more credit cards can be issued to an account, at most one loan can be granted for an account. Relation "demographic data" gives some publicly available information about the districts (e.g. the unemployment rate); additional information about the clients can be deduced from this.

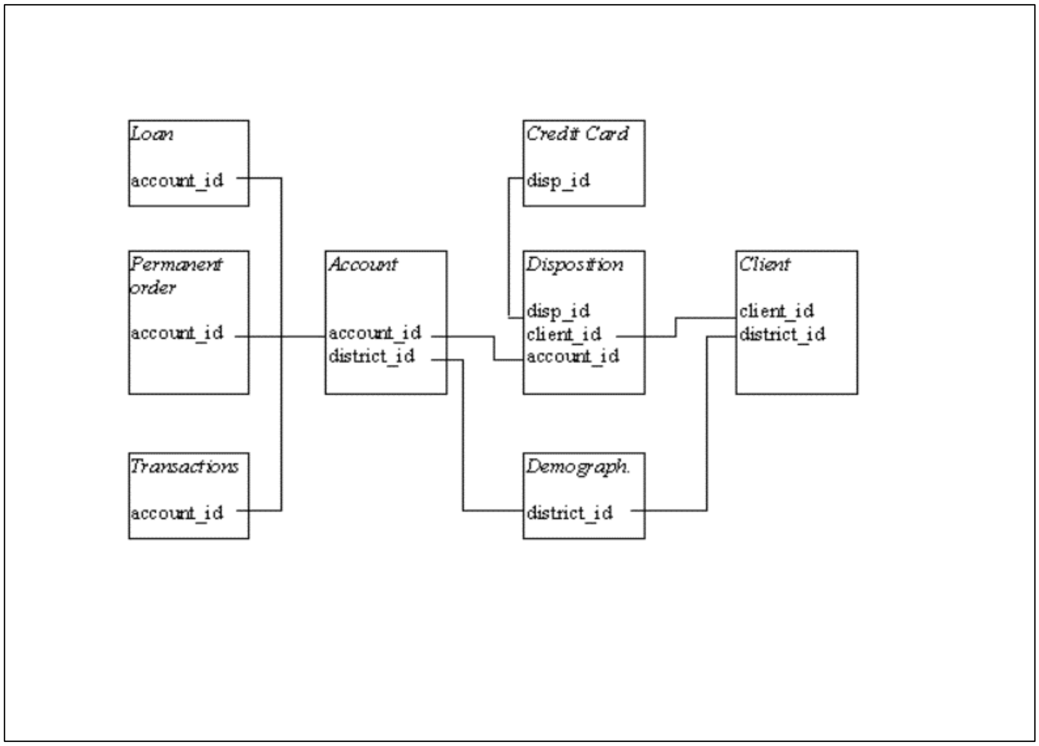
And, analysts downloaded the data from data.world

https://data.world/lpetrocelli/czech-financial-dataset-real-anonymized-transactions

and obtained a gif file of the data structure, and verification the downloaded the data is in a dedicated folder.

account	143 KB	Microsoft Excel Comma Separated Values File
card	30 KB	Microsoft Excel Comma Separated Values File
client	83 KB	Microsoft Excel Comma Separated Values File
disp	117 KB	Microsoft Excel Comma Separated Values File
district	7 KB	Microsoft Excel Comma Separated Values File
loan	24 KB	Microsoft Excel Comma Separated Values File
order	212 KB	Microsoft Excel Comma Separated Values File
trans	56,324 KB	Microsoft Excel Comma Separated Values File

Figure 1 - Data Structure



Data and primary keys

- ✓ Loan data

Primary key: account_id

- ✓ Transaction data

Primary key: account_id

- ✓ Account data

Primary Key: account_id
secondary Key: district_id

- ✓ Demographic data

Primary Key: district_id

- ✓ Disposition data

Primary Key: account_id
secondary Key: client_id
tertiary Key: disp_id

- ✓ Credit Card data

Primary Key: disp_id

- ✓ Client data

Primary Key: client_id
secondary Key: district_id

<u>Data integration:</u>

From the above graphic analysts accomplished the following:

Step 1. The data was left joined on account_id for loan data, transaction data, account_data.
Step 2. The left joined data result of step 1 was left joined to the district data on district_id.
Step 3. The result of step 1 and step 2 was left joined with disposition data by disp_id.
Step 4. The result of step 1, step 2 and step 3 was left joined with card data by disp_id.
Step 5. The result of step 1, step 2, step 3 and step 4 was left joined with the client data by client_id

The final data was validated by three team members in the following table.

Table 1 Data Table matches Berka data from data.world website

Data	Count
Account	4500
Card	892
Client	5369
Disp	5369
District	77
Loan	682
Order	6471
Trans	1,056,320

The final data frame has 1,262,625 records.

Analysts used the rubric in figure one using left joins, and the data inventory appears in the table below.

Each account has both static characteristics (e.g. date of creation, address of the branch) given in relation "account" and dynamic characteristics (e.g. payments debited or credited, balances) given in relations "permanent order" and "transaction". Relation "client" describes characteristics of persons who can manipulate with the accounts. One client can have more accounts, more clients can manipulate with single account; clients and accounts are related together in relation "disposition". Relations "loan" and "credit card" describe some services which the bank offers to its clients; more credit cards can be issued to an account, at most one loan can be granted for an account. Relation "demographic data" gives some publicly available information about the districts (e.g. the unemployment rate); additional information about the clients can be deduced from this.

Table 2 Relation account data frame

Relation account

item	meaning	remark
account_id	identification of the account	
district_id	location of the branch	
date	date of creating of the account	in the form YYMMDD
frequency	frequency of issuance of statements	"POPLATEK MESICNE" stands for monthly issuance "POPLATEK TYDNE" stands for weekly issuance "POPLATEK PO OBRATU" stands for issuance after transaction

<u>Note:</u>
account_id = identification of the account
district_id= location of the branch
date = date of creating of the account in the form YYMMDD
frequency = frequency of issuance of statements

"POPLATEK MESICNE" stands for monthly issuance
"POPLATEK TYDNE" stands for weekly issuance
"POPLATEK PO OBRATU" stands for issuance after transaction

Table 3 Relation client data frame

Relation client

item	meaning	remark
client_id	record identifier	
birth number	identification of client	the number is in the form YYMMDD for men, the number is in the form YYMM+50DD for women, where YYMMDD is the date of birth
district_id	address of the client	

Note:

client_id =record identifier
birth number = identification of client the number is in the form YYMMDD for men,
the number is in the form YYMM+50DD for women,
where YYMMDD is the date of birth
district_id = address of the client

Table 4 Relation disposition data frame

Relation disposition

item	meaning	remark
disp_id	record identifier	
client_id	identification of a client	
account_id	identification of an account	
type	type of disposition (owner/user)	only owner can issue permanent orders and ask for a loan

Note:

disp_id = record identifier
client_id= identification of a client
account_id = identification of an account
type = type of disposition (owner/user) only owner can issue permanent orders and ask for a loan

Table 5 Relation permanent order data frame

item	meaning	remark
order_id	record identifier	
account_id	account, the order is issued for	
bank_to	bank of the recipient	each bank has unique two-letter code
account_to	account of the recipient	
amount	debited amount	
K_symbol	characterization of the payment	"POJISTNE" stands for insurrance payment "SIPO" stands for household "LEASING" stands for leasing "UVER" stands for loan payment

This data frame was found to double the amount of records, and did not add significant information to the overall database, and was not used by analysts.

Table 6 Relation transaction data frame

item	meaning	remark
trans_id	record identifier	
account_id	account, the transation deals with	
date	date of transaction	in the form YYMMDD
type	+/- transaction	"PRIJEM" stands for credit "VYDAJ" stands for withdrawal
operation	mode of transaction	"VYBER KARTOU" credit card withdrawal "VKLAD" credit in cash "PREVOD Z UCTU" collection from another bank "VYBER" withdrawal in cash "PREVOD NA UCET" remittance to another bank
amount	amount of money	
balance	balance after transaction	
k_symbol	characterization of the transaction	"POJISTNE" stands for insurrance payment "SLUZBY" stands for payment for statement "UROK" stands for interest credited "SANKC. UROK" sanction interest if negative balance "SIPO" stands for household "DUCHOD" stands for old-age pension "UVER" stands for loan payment
bank	bank of the partner	each bank has unique two-letter code
account	account of the partner	

Note:

trans_id = record identifier
account_id = account, the transation deals with
date = date of transaction in the form YYMMDD
type = +/- transaction "PRIJEM" stands for credit
"VYDAJ" stands for withdrawal
operation=bmode of transaction
"VYBER KARTOU" credit card withdrawal
"VKLAD" credit in cash
"PREVOD Z UCTU" collection from another bank
"VYBER" withdrawal in cash
"PREVOD NA UCET" remittance to another bank
amount= amount of money
balance = balance after transaction
k_symbol =characterization of the transaction "POJISTNE" stands for insurrance payment
"SLUZBY"= stands for payment for statement
"UROK" = stands for interest credited
"SANKC. UROK" = sanction interest if negative balance
"SIPO" = stands for household
"DUCHOD"= stands for old-age pension
"UVER" = stands for loan payment
bank = bank of the partner each bank has unique two-letter code
account= account of the partner

Table 7 Relation loan data frame

item	meaning	remark
loan_id	record identifier	
account_id	identification of the account	
date	date when the loan was granted	in the form YYMMDD
amount	amount of money	
duration	duration of the loan	
payments	monthly payments	
status	status of paying off the loan	'A' stands for contract finished, no problems, 'B' stands for contract finished, loan not payed, 'C' stands for running contract, OK so far, 'D' stands for running contract, client in debt

Note:

loan_id = record identifier
account_id = identification of the account
date = date when the loan was granted in the form YYMMDD
amount = amount of money
duration = duration of the loan
payments = monthly payments
status = status of paying off the loan
'A' stands for contract finished, no problems,
'B' stands for contract finished, loan not payed,
'C' stands for running contract, OK so far,
'D' stands for running contract, client in debt

Table 8 Relation credi card data frame

item	meaning	remark
card_id	record identifier	
disp_id	disposition to an account	
type	type of card	possible values are "junior", "classic", "gold"
issued	issue date	in the form YYMMDD

Note:

card_id = record identifier
disp_id = disposition to an account
type = type of card possible values are "junior", "classic", "gold"
issued = issue date in the form YYMMDD

Table 9 Relation demographic data frame

item	meaning	remark
A1 = district_id	district code	
A2	district name	
A3	region	
A4	no. of inhabitants	
A5	no. of municipalities with inhabitants < 499	
A6	no. of municipalities with inhabitants 500-1999	
A7	no. of municipalities with inhabitants 2000-9999	
A8	no. of municipalities with inhabitants >10000	
A9	no. of cities	
A10	ratio of urban inhabitants	
A11	average salary	
A12	unemploymant rate '95	
A13	unemploymant rate '96	
A14	no. of enterpreneurs per 1000 inhabitants	
A15	no. of commited crimes '95	
A16	no. of commited crimes '96	

<u>Note:</u>

A1 = district_id district code
A2 = district name
A3 = region
A4 = no. of inhabitants
A5 = no. of municipalities with inhabitants < 499
A6 = no. of municipalities with inhabitants 500-1999
A7 = no. of municipalities with inhabitants 2000-9999
A8 = no. of municipalities with inhabitants >10000
A9 = no. of cities
A10 = ratio of urban inhabitants
A11 = average salary
A12 = unemployment rate '95
A13 = unemployment rate '96
A14 = no. of entrepreneurs per 1000 inhabitants
A15 = no. of committed crimes '95
A16 = no. of committed crimes '96

Next is the import data log

Table 10 Data Processing

DATA LOG (Note the order database was left out it did not contribute information but doubled the number of records)			
Data	**obs**	**Unique ID**	**Comment**
loans	682	682	account_id used for join
trans	1056320	4500	account_id used for join
order	6471	3758	account_id used for join (note that there are 6471 unique order IDs)
account	4500	4500	account_id used for join
district	77	77	district_id used for join
disp	5369	4500	account_id used for join
card	892	892	disp_id used for join
client	5369	5369	client_id used for join
leftjoindat	1056320		Same as trans data and loans
unique_leftjoindat	682	682	Same as loan data
leftjoindat_account	1056320		number of row in database with is the same as trans and loan data
leftjoindat_account_check	62625		omit loan accounts that are missing
unique_leftjoindat_account	682	682	unique account_ ids in this data 682 exactly the loan count
leftjoindat_account_district	1056320		integrate district data
leftjoindat_account_district_check	62625		omit loan accounts that are missing
unique_leftjoindat_account_district	682	682	Same as loan data
leftjoindat_account_district_disp	1262625		
leftjoindat_account_district_disp_check	77073		
unique_leftjoindat_account_district_disp	682	682	Same as loan data
leftjoindat_account_district_disp_card	1262625		
leftjoindat_account_district_disp_card_check	15185		
unique_leftjoindat_account_district_disp_card	170		Looks like not every indivdual with a loan has a credit card
leftjoindat_order_account_district_disp_card_client	1262625		
leftjoindat_order_account_district_disp_card_client_check	15185		
unique_leftjoindat_order_account_district_disp_card_client	170		
leftjoindat_order_account_district_disp_card_client1	1056320	1056320	This is all the joined data with unique trans_id
leftjoindat_order_account_district_disp_card_client2	4500	4500	this is all the joined data with unique account_id
leftjoindat_order_account_district_disp_card_client3	683	682 using na.omit	this is all the joined data with unique loan_id -- picked up one NA in the loan ID

Explore, Visualize and Preprocess the Data

The first table examines the transactions data frame operations variable which has five categories.

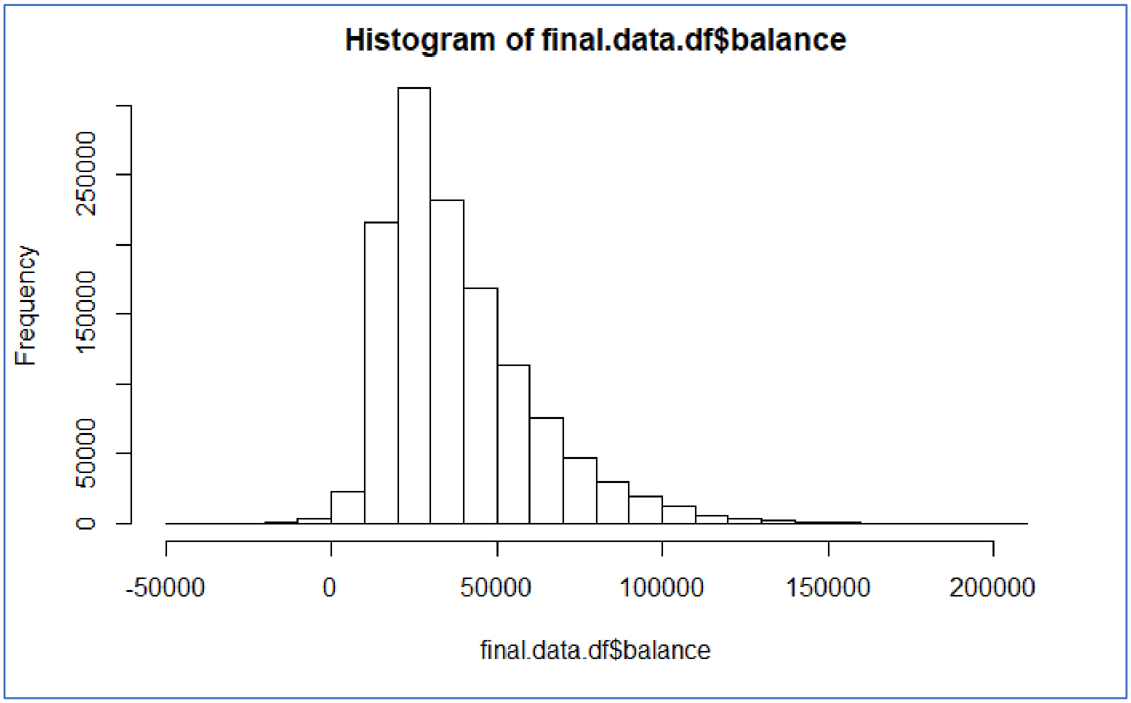
1. 'VYBER KARTOU' stands for Credit Card Withdrawal
2. 'VKLAD' stands for Credit in Cash
3. 'PREVOD Z UCTU' stands for Collection from Another Bank
4. 'VYBER' stands for Withdrawal in Cash
5. 'PREVOD NA UCET' stands for Remittance to Another Bank

Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
	0	0.0000000
PREVOD NA UCET	254449	20.1523810
PREVOD Z UCTU	81601	6.4628057
VKLAD	181962	14.4114048
VYBER	517031	40.9488961
VYBER KARTOU	9271	0.7342639
None	218311	17.2902485

This variable examines the withdrawal characteristics of an obligor looking at credit card or cash withdrawals which may be a profit opportunity for the bank because cash withdrawal may be substituted for credit card withdrawals where a higher profit margin resides, and the clear majority, as the table indicates, are cash withdrawals.

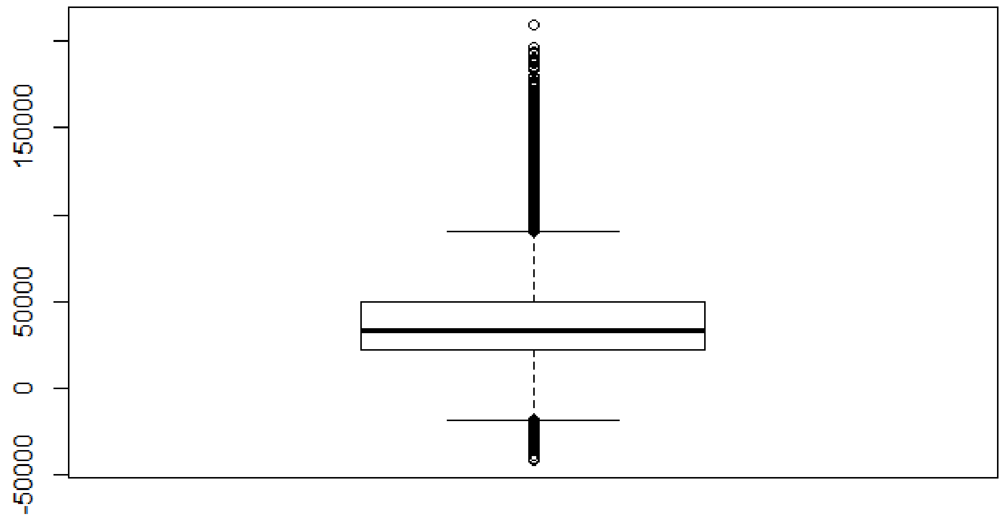
Next is an examination of balance.

Figure 2 Transaction Balance



There are negative balances which the bank my want to explore further to determine why a negative transaction balance would be on the books, and also note balances are skewed to the right, and the boxplot below gives lots off outliers with larger accounts, senior management might want to explore these larger accounts.

Figure 3 Transaction balance boxplot



Summary statistics below indicate the median transaction balance is 38,421.

```
Min.     1st Qu.  Median     Mean 3rd Qu.     Max.
-41126    22256    32961    38421    49436   209637
```

Below are the frequencies of partner banks that interact with the bank and it appears that our bank interacts with partner banks in a consistent manner the numbers being very similar.

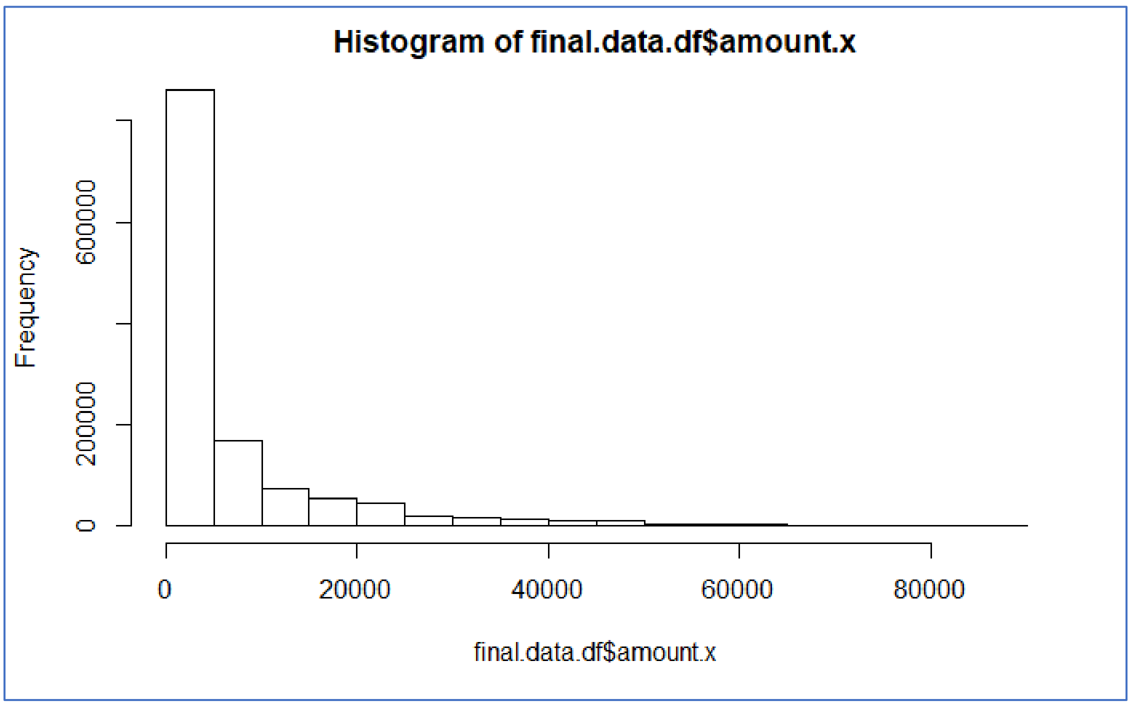
Banks

Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
	0	0.000000
AB	26521	2.100465
CD	24119	1.910227
EF	26216	2.076309
GH	26292	2.082328
IJ	25650	2.031482
KL	26130	2.069498
MN	24027	1.902940
OP	25510	2.020394
QR	27074	2.144263

Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
ST	26784	2.121295
UV	26227	2.077180
WX	25210	1.996634
YZ	26289	2.082091
None	926576	73.384893

Next is the transaction amount.

Figure 4 Transaction amount



The transaction amounts are small and this is an opportunity for the bank's marketing department to advertise encouraging customers to make larger purchases.

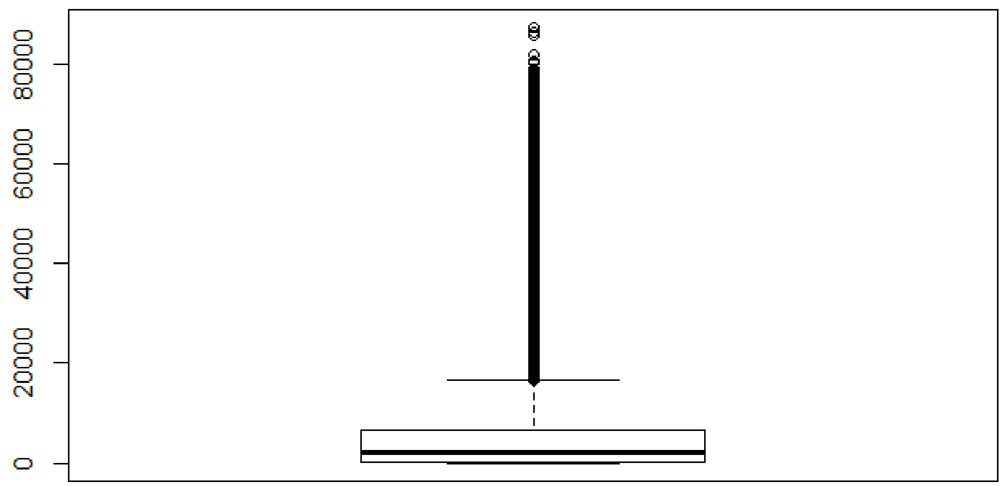
The median transaction as shown below is 2100 koruna.

```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  0.0   134.9  2100.0  5873.2  6694.0 87400.0
```

Now the boxplot below show a large number of outliers that might be explored to understand the motivation of the customer to make larger transactions, and encourage other customers to do the same.

Figure 5 Transaction amount boxplot

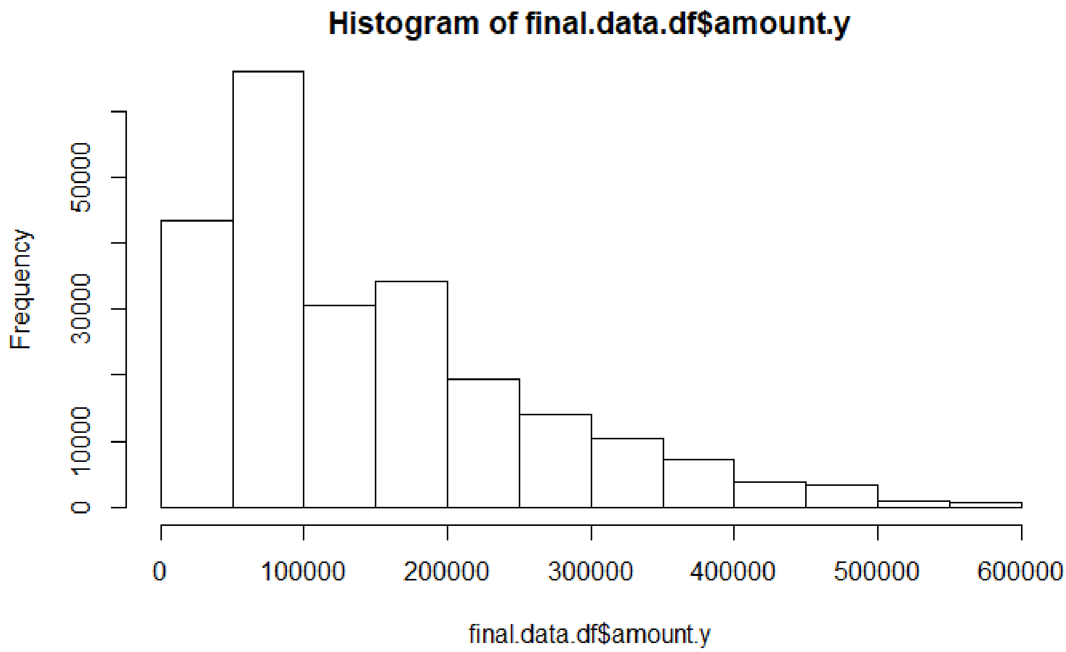
Amount.x



Notice the large number of outliers here.

Next examine loans.

Figure 6 Loan amount histogram

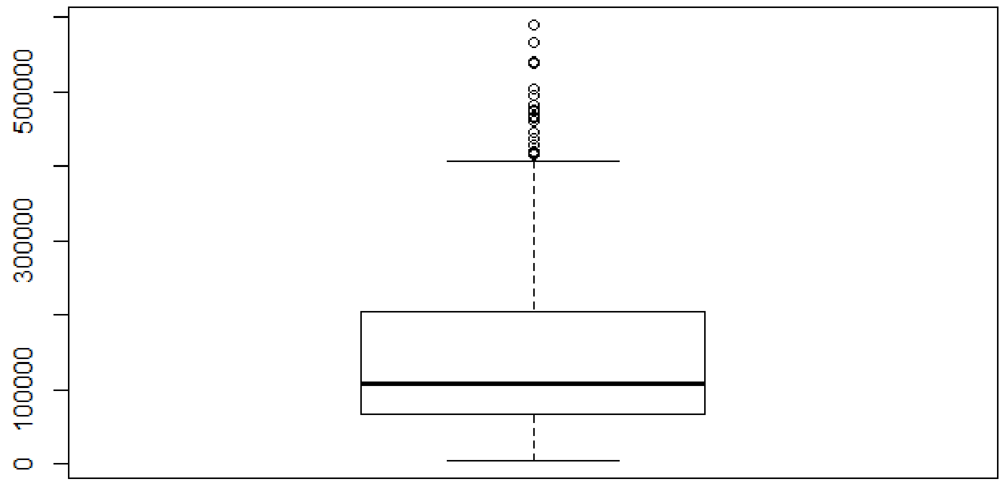


Notice the skewed distribution as expected, that is smaller loans.

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
4980   67464  108144  147130  203940  590820 1028998
```

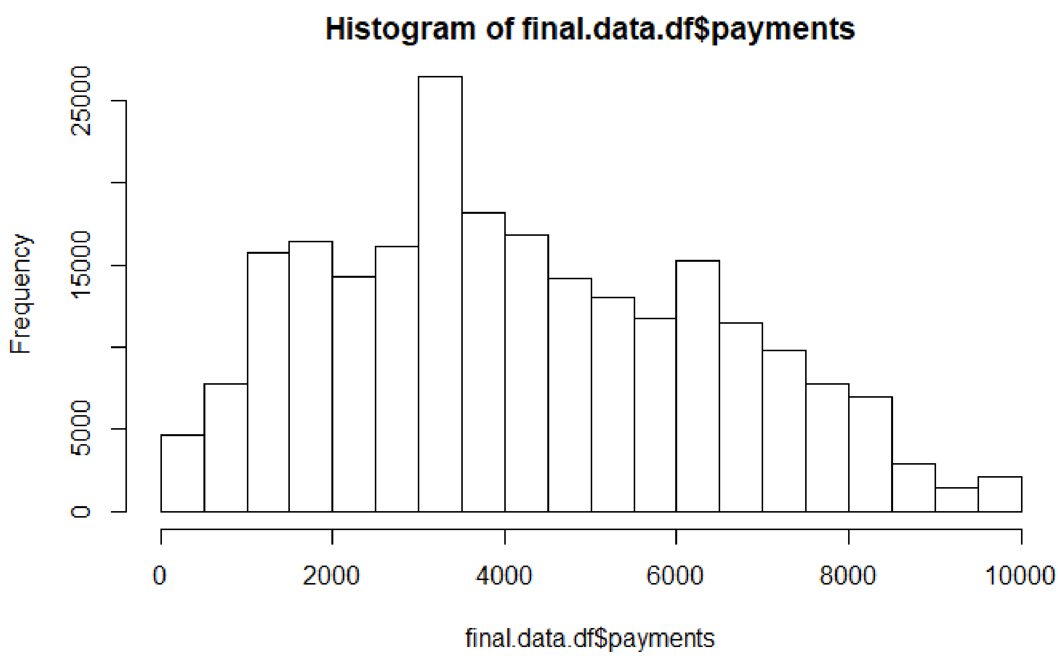
The median loan is 108,144 koruna, and the boxplot below shows there are outliers the bank might explore to understand what motivates customers to get larger loans.

Figure 7 Loan amount boxplot



There are outliers in the loan amount data that might need to be examined.

Figure 8 Loan payment histogram



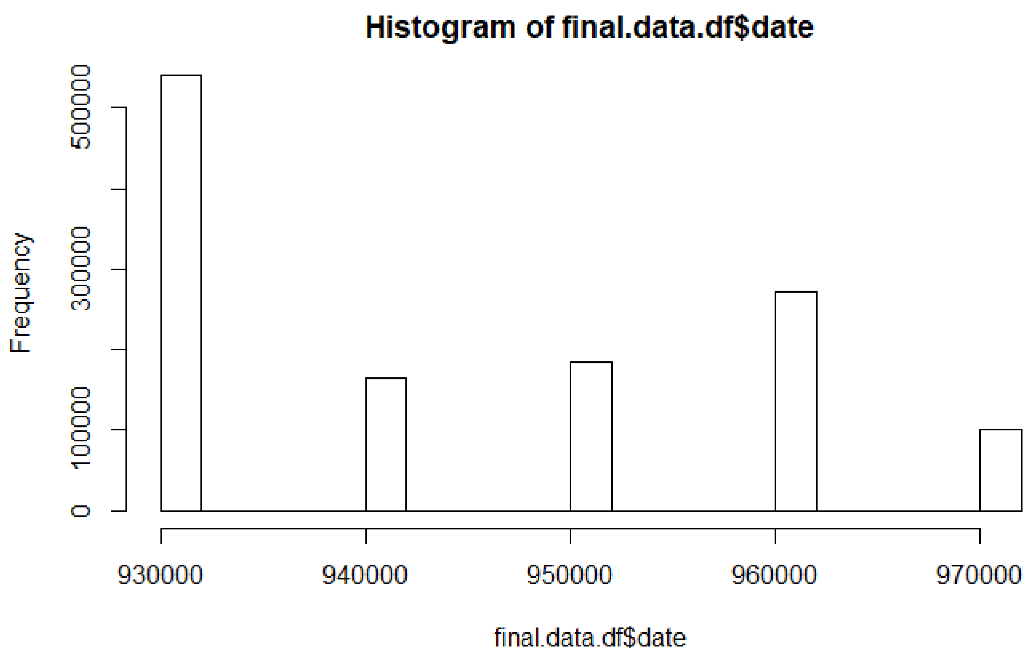

```
Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
```

304 2490 3900 4217 5996 9910 1028998

The median payment is 3900 koruna.

The next group of histograms show date activity.

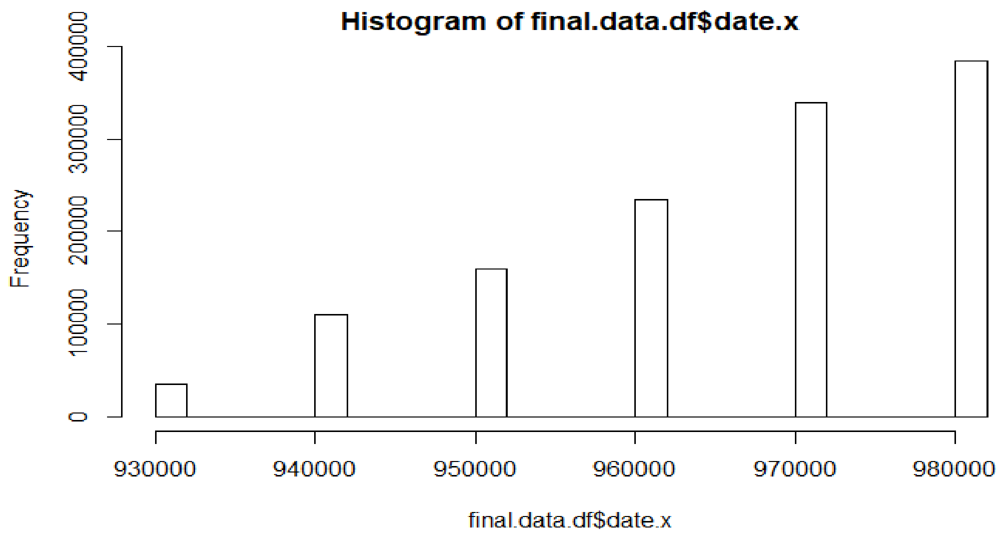
Figure 9 Loan origination date in the account data frame



Date is from account.df and the date (yymmdd) the account was created, and notice a lot more activity in 1993 which may be due to macroeconomic circumstances or local idiosyncratic behavior of the bank, and this needs to be investigated.

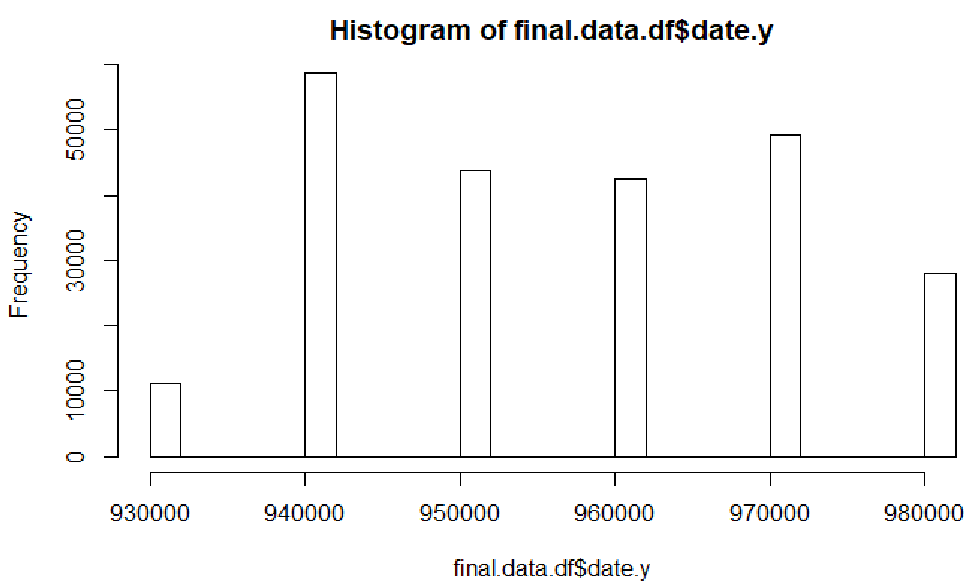
Next is date.x is from the transaction.df data and is the date of the transaction.

Figure 10 Transaction date



Notice many more transactions in 1998, and again this could be a macroeconomic effect, or local idiosyncratic effect. More transaction in 1998, but less loan origination activity in the above graphs needs to be investigated.

Figure 11 Loan origination date



date.y is from the loan.df data and is the date the loan was granted, and less activity in 1998 is observed and again, if senior management cannot explain the decline, an investigation needs to be undertaken.

Next is the region frequency in the database.

Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
central_Bohemia	158588	12.560182
east_Bohemia	143148	11.337333
north_Bohemia	130837	10.362301
north_Moravia	228096	18.065221
Prague	160444	12.707178
south_Bohemia	101212	8.015998
south_Moravia	219395	17.376101
west_Bohemia	120905	9.575686

Notice above that North Moravia has the most activity, but Prague is third on the list for activity

There are 77 districts in the database and each of the 77 has demographic statistics and for each district enumerated an entry for the muni's with less that 499 inhabitants is recorded, so that the last entry read 151 municipalities had less that 499 inhabitants for one of the 77 districts and that occurs for 17345 records in the database. Other categories are read similarly.

final.data.df$num_of_municipalities_with_inhabitants_LT_499

Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
0	290910	23.0400950
4	12047	0.9541234
5	11327	0.8970993
8	24600	1.9483219
9	13445	1.0648451
10	12706	1.0063162
11	12847	1.0174834
15	35484	2.8103356
17	27526	2.1800614
21	24836	1.9670132

Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
22	26299	2.0828829
24	14964	1.1851500
25	13850	1.0969211
28	9234	0.7313335
29	42841	3.3930106
31	24790	1.9633700
32	38443	3.0446886
34	15173	1.2017028
35	12799	1.0136818
37	12491	0.9892882

Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
38	36917	2.9238293
41	31787	2.5175329
48	12784	1.0124938
49	31359	2.4836353
50	31355	2.4833185
52	12086	0.9572122
55	27355	2.1665182
59	16830	1.3329373
60	33920	2.6864667
61	25243	1.9992476

Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
63	12385	0.9808930
65	28042	2.2209286
66	12649	1.0018018
67	15755	1.2477972
69	10765	0.8525889
71	21570	1.7083457
73	13157	1.0420354
74	12687	1.0048114
75	14251	1.1286803
77	11738	0.9296505

Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
80	11410	0.9036729
83	15386	1.2185724
84	11322	0.8967033
85	11388	0.9019305
87	25162	1.9928324
88	13556	1.0736363
94	11067	0.8765073
95	14577	1.1544996
98	9217	0.7299871
99	10981	0.8696961

Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
101	8780	0.6953767
139	13178	1.0436986
151	17354	1.3744382

The way to read the above table is that, for example, there are 290910 records with 0 muni's LT 499, and there are lots of small communities in the database.

$num_of_municipalities_with_inhabitants_2000_9999

Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
0	248771	19.7026829
1	33669	2.6665875
2	42246	3.3458865
3	64310	5.0933571
4	162737	12.8887833
5	110688	8.7664984
6	132850	10.5217305
7	148702	11.7772102
8	100005	7.9204039
9	12799	1.0136818

Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
10	73758	5.8416394
11	11327	0.8970993
12	27975	2.2156222
13	27412	2.1710326
14	12847	1.0174834
18	40818	3.2327888
20	11711	0.9275121

Also, there are a lot of communities in the database that are not necessarily small, but have somewhere between 2000 and 10000 inhabitants.

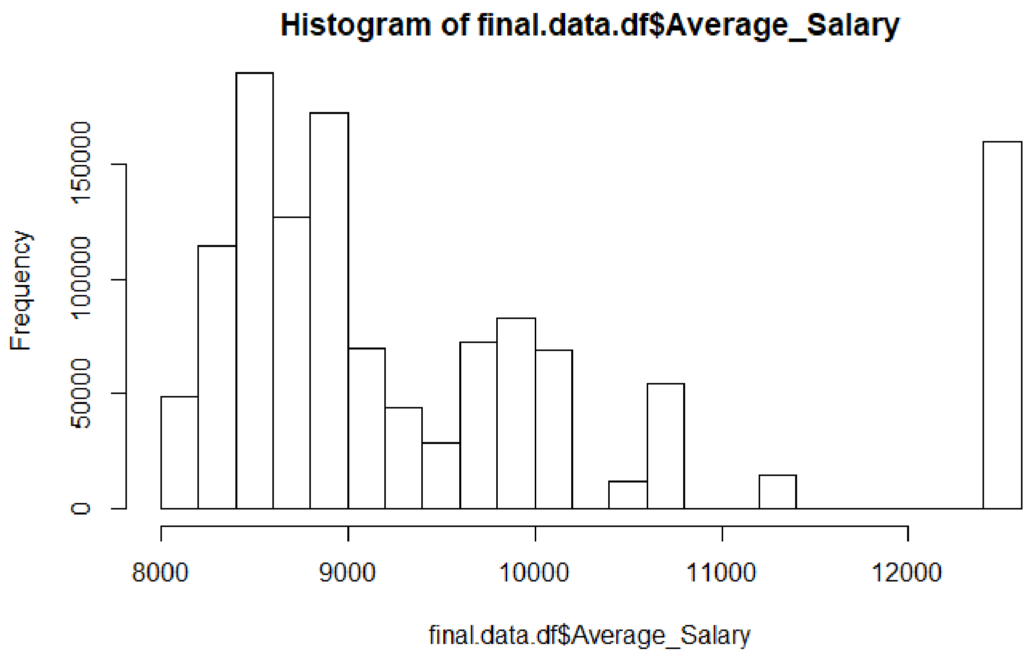
The following is the number of cities.

Var1 <fctr>	Freq <int>
1	248771
2	11882
3	24334
4	166406
5	131497
6	197116
7	170331
8	91766
9	91032
10	102035
11	27455

Next is the average salary.

In each of the 77 districts an average salary is given, and the histogram appears below.

Figure 12 Histogram of average salary



Notice there is an outlier in the mix and one district has a large average salary which might want to be explored.

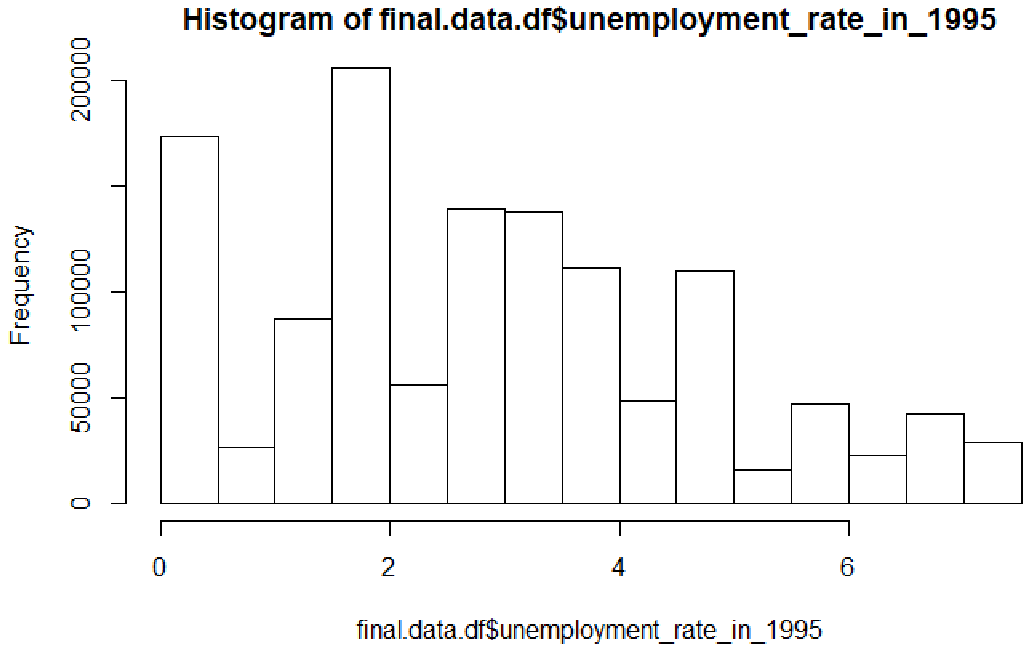
Here are the summary statistics for average salary.

```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 8110    8554    8994    9540    9920   12541
```

Notice the outlier, the median average salary for the 77 districts is 8994 koruna. Senior management should investigate average salaries greater than 9920, which is the third quartile, and represents an opportunity for wealth management.

Figure 13 Unemployment rate 1995

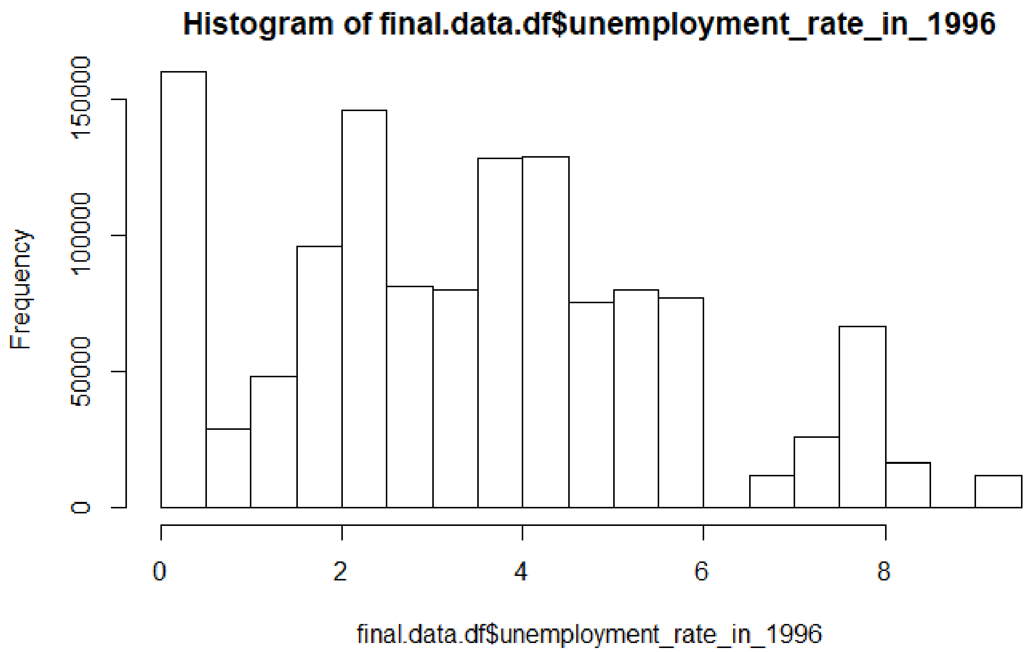
The unemployment rate for the 77 districts looks to have a fair dispersion with close to 2% having the most frequency, but some districts have more than 6%.




```
 Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.290   1.510   2.770   2.885   4.010   7.340
```

The median unemployment for the 77 districts is 2.77

Figure 14 Unemployment rate 1996



The unemployment rate for the 77 districts looks to have a much different distribution in 1996, and is bimodal with one distribution up to 6%, and the other distribution centered around 7.8%.

```
Min.    1st Qu.  Median    Mean 3rd Qu.    Max.
0.430    1.960   3.490    3.507   4.790   9.400
```

Now the median is 3.49% and increase of 26% over 1995.

Figure 15 Number of crimes in 1995

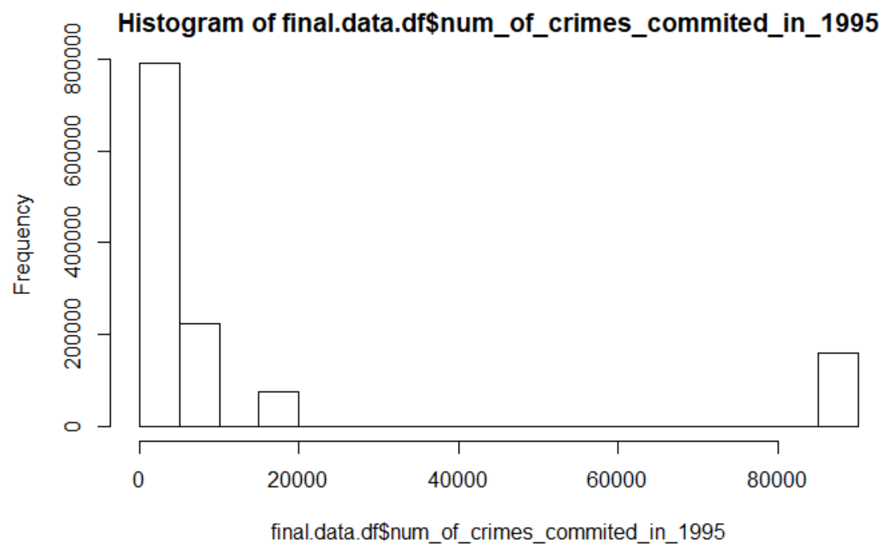
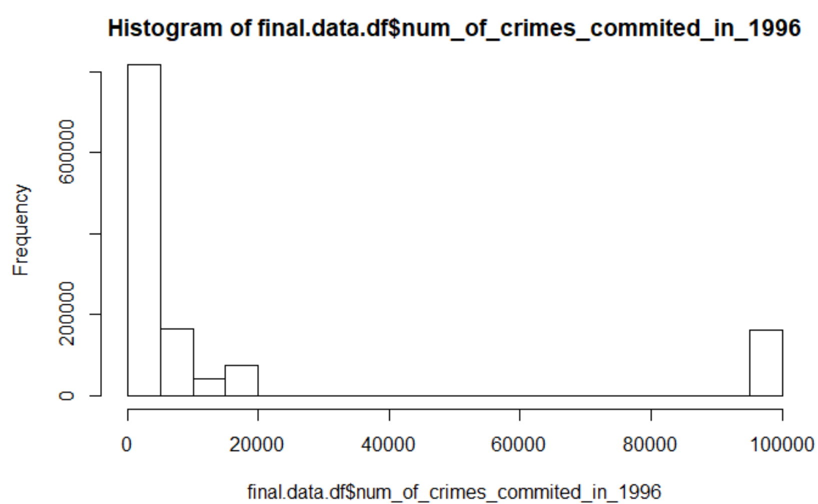


Figure 16 Number of crimes in 1996



Notice the difference between the two histograms the distribution in 1996 is filling out somewhat at the lower tail, and senior management should investigate the districts with higher crime rates against loans and transactions to understand if crime is impacting sales.

type.x (transaction data base)
type.y (loan data)
type (disposition)

table(final.data.df$type.x)
table(final.data.df$type.y)
table(final.data.df$type)

The following table comes from the transaction database where:

"PRIJEM" stands for credit
VYBER' stands for Withdrawal in Cash
"VYDAJ" stands for withdrawal

Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
PRIJEM	481874	38.164459
VYBER	19487	1.543372
VYDAJ	761264	60.292169

Observe that cash withdrawals represent 19487 records, and credit withdrawals represent 481874 records, and there is an opportunity for senior management to better understand these 19487 records, and possibly turn them from cash to credit.

Next are class credit cards, gold credit cards and junior credit cards.

Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
classic	159859	12.660845
gold	25740	2.038610
junior	36339	2.878052
None	1040687	82.422493

There is opportunity for management to possibly move classic card holders to the gold edition for a premium, and consequently increase revenue.

This data is from the transaction data frame (k_symbol)

Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
	0	0.0000000
	64868	5.1375507
DUCHOD	39322	3.1143055
POJISTNE	22229	1.7605386
SANKC. UROK	1578	0.1249777
SIPO	143452	11.3614098
SLUZBY	186440	14.7660628
UROK	218311	17.2902485
UVER	16608	1.3153549
None	569817	45.1295515

"POJISTNE" stands for insurance payment
"SLUZBY" stands for payment for statement
"UROK" stands for interest credited
"SANKC. UROK" sanction interest if negative balance
"SIPO" stands for household
"DUCHOD" stands for old-age pension
"UVER" stands for loan payment

Notice UVER is a loan payment and pojistne is an insurance payment, and there might be a cross sell opportunity loans by insurance contacting those household who have loans with the bank, but insurance with another bank.

Duration of the loan is when the loan is due 12 month, 24 months, 36 months, 48 months, and 60 months, and these are standard loan terms.

Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
12	48869	20.91753
24	51352	21.98034
36	42301	18.10621
48	42599	18.23377
60	48506	20.76216

The status field is associate with good and bad loans

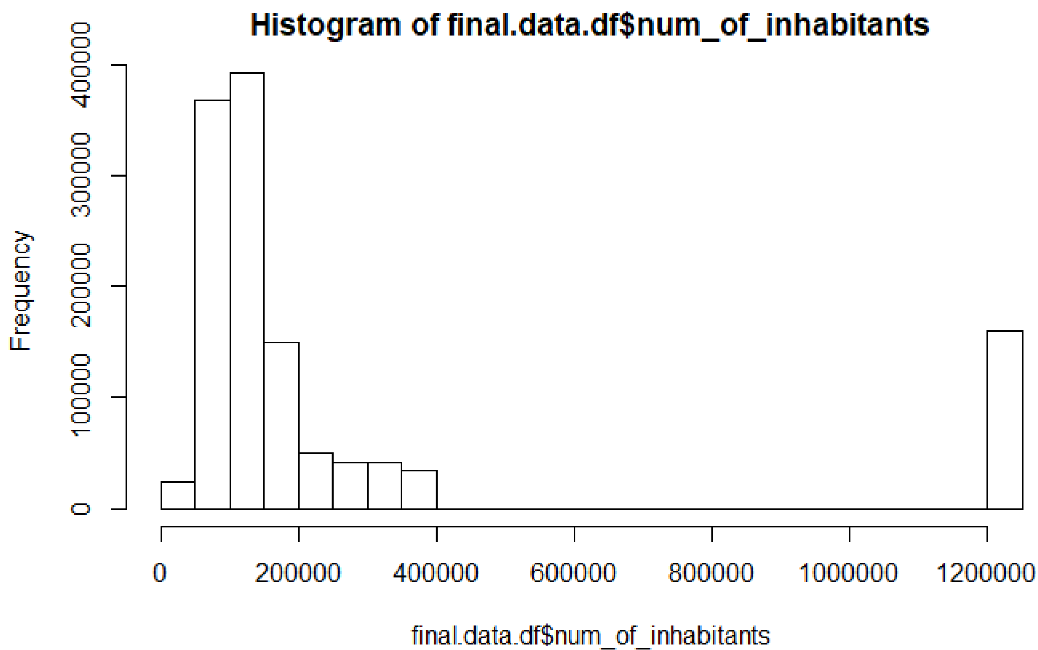
Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
A	100329	42.944095
B	11469	4.909107
C	110538	47.313881
D	11291	4.832917

```
'A' stands for contract finished, no problems,
'B' stands for contract finished, loan not payed,
'C' stands for running contract, OK so far,
'D' stands for running contract, client in debt
```

Analysts pick B and D, contract finished – loan not paid off, and client in debt as problematic loans which are then model 1 (bad), 0 (good) loans.

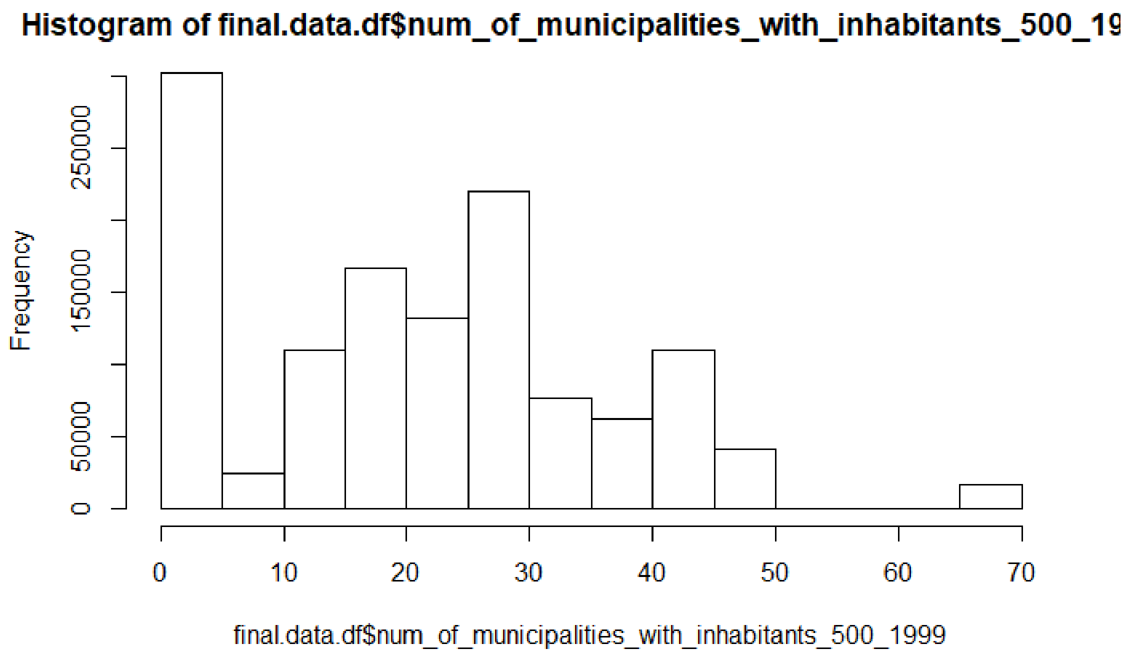
Now examine the number of inhabitants in the 77 districts.

Figure 17 Number of inhabitants in the districts



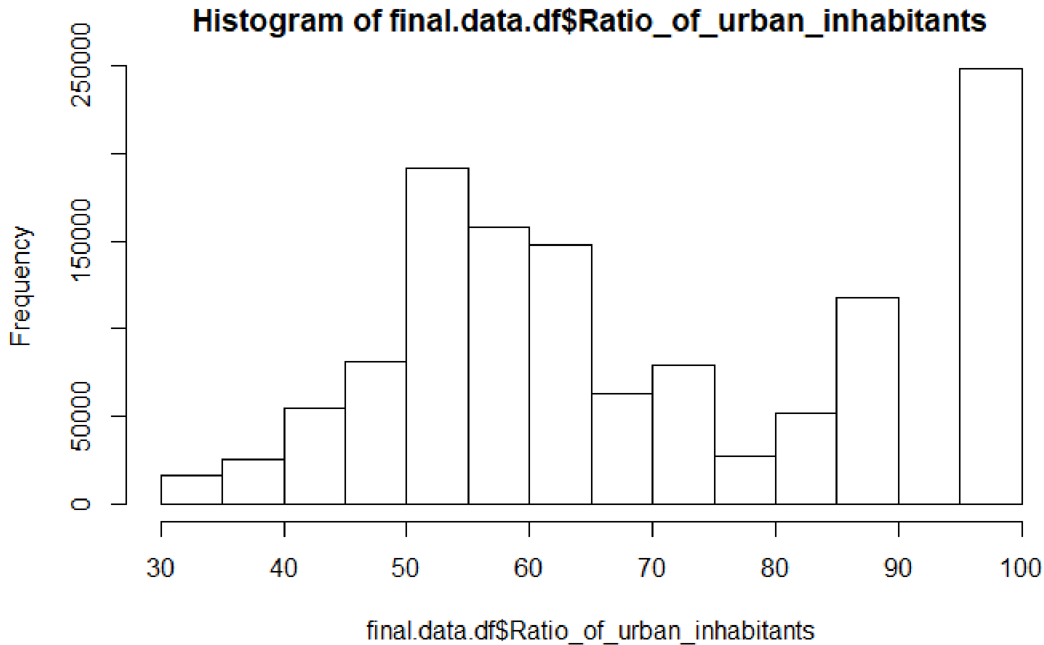
The largest district is not Prague (120,000) note that, Ostrava – mesto and Brno – mesto districts have 323,000 and 387,000 respectively

Figure 18 Municipalities with 500 - 1999 inhabitants



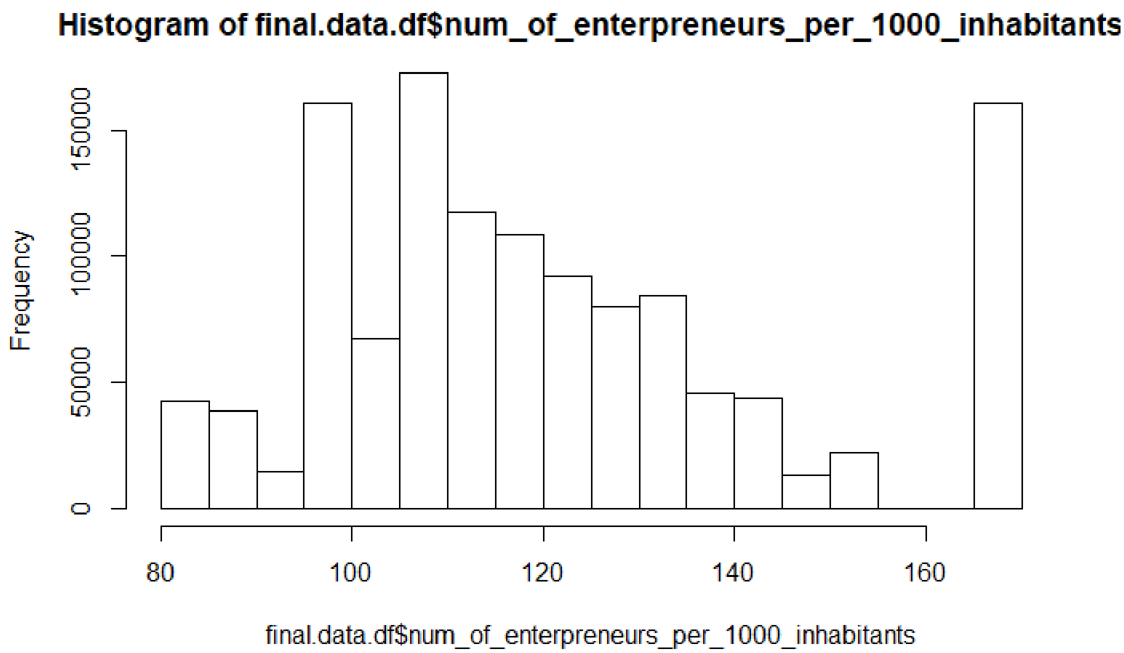
The x-axis in the graph is the 77 districts represented as the frequency of municipalities with 500-1999 inhabitants, so for example one of the districts has 70 municipalities that have population between 500-1999.

Figure 19 Ratio of urban inhabitants



There is one district, as shown in the graph that is all urban (100%) and there are districts that have as low as 30% urban inhabitants, and it may very well be that urban dwellers are more educated than their country counterparts, and management may want to market to each segment (urban versus rural) differently.

Figure 20 Number of entrepreneurs per 1000



Prague has the most entrepreneurs per 1000 inhabitants at 167 which is the outlier in the histogram, and there are a number of records associated with 110 entrepreneurs per 1000 inhabitants, and the assumption would be the more entrepreneurs the more demand for banking services.

Analysts created a binary dummy variable that separated the data into customers that contracted for a bank loan and those that did not.

Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
0	1028998	81.49672
1	233627	18.50328

Eight-one percent of the database records do not have a loan contract, and 19% of the database records are associated with a loan contract. This dummy variable can be used to profile the no loan customers from the loan customers, and allow the marketing department to understand the differences and bridge the gap between loan and no loan customers, hopefully, moving some of the no loan to the loan category.

Analysts created another dummy variable that separates good loans from bad loan, and this was facilitated using the Status field.

The status field is associate with good and bad loans

Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
A	100329	42.944095
B	11469	4.909107
C	110538	47.313881
D	11291	4.832917

```
'A' stands for contract finished, no problems,
'B' stands for contract finished, loan not payed,
'C' stands for running contract, OK so far,
'D' stands for running contract, client in debt
```

A good loan was defined at either "A" contracted finished no problems, or "C" contract running smoothly so far, and a bad loan as "B" contract finished and the loan is not paid or "D" the contract is running but the client is in debt where 0 is a good loan and 1 is a bad loan.

bad.good.loan.data.df <fctr>	Freq <int>	percentage_Freq <dbl>
0	210867	90.257975
1	22760	9.742025

Analyst also oversampled the data by taking all the 1's and sampling 10% of the 0's, and in this way, analysts are able to build a model on a data sample that "should" produce a more accurate model.

Oversampled Dummy

Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
0	22759	49.9989
1	22760	50.0011

Analysts also created dummy variables for transaction amount where the 3rd quartile is 6694 and greater than that number is 1 and below is 0. Similarly, transaction balance was partitioned into data that was above the 3rd quartile (49436) as 1 and below as 0. Also, an average salary dummy was created where the 3rd quartile was 9920, and was flag a 1, and below the 3rd quartile as 0.

The data is partitioned into geographic regions

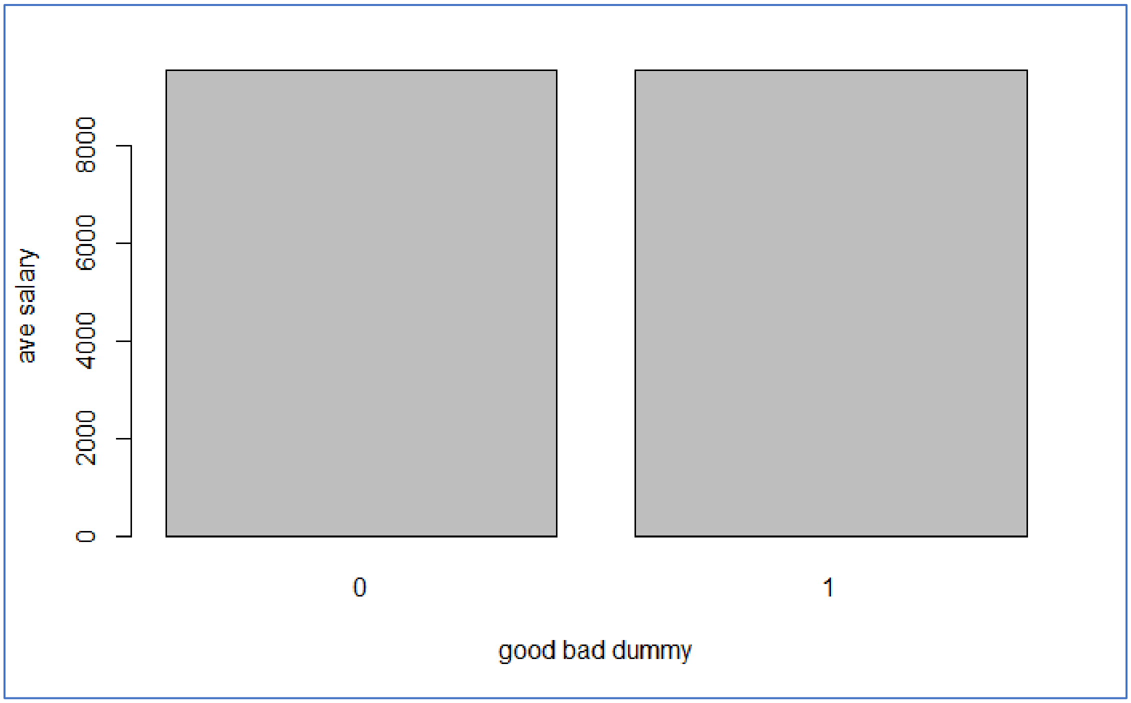
Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
central_Bohemia	158588	12.560182
east_Bohemia	143148	11.337333
north_Bohemia	130837	10.362301
north_Moravia	228096	18.065221
Prague	160444	12.707178
south_Bohemia	101212	8.015998
south_Moravia	219395	17.376101
west_Bohemia	120905	9.575686

With the most records in North Moravia.

Also, the credit card data is partitioned into 3 grades: the classic, gold a junior.

Var1 <fctr>	Freq <int>	percentage_Freq <dbl>
classic	159859	12.660845
gold	25740	2.038610
junior	36339	2.878052
None	1040687	82.422493

Profile of a Good Loan versus a Bad Loan

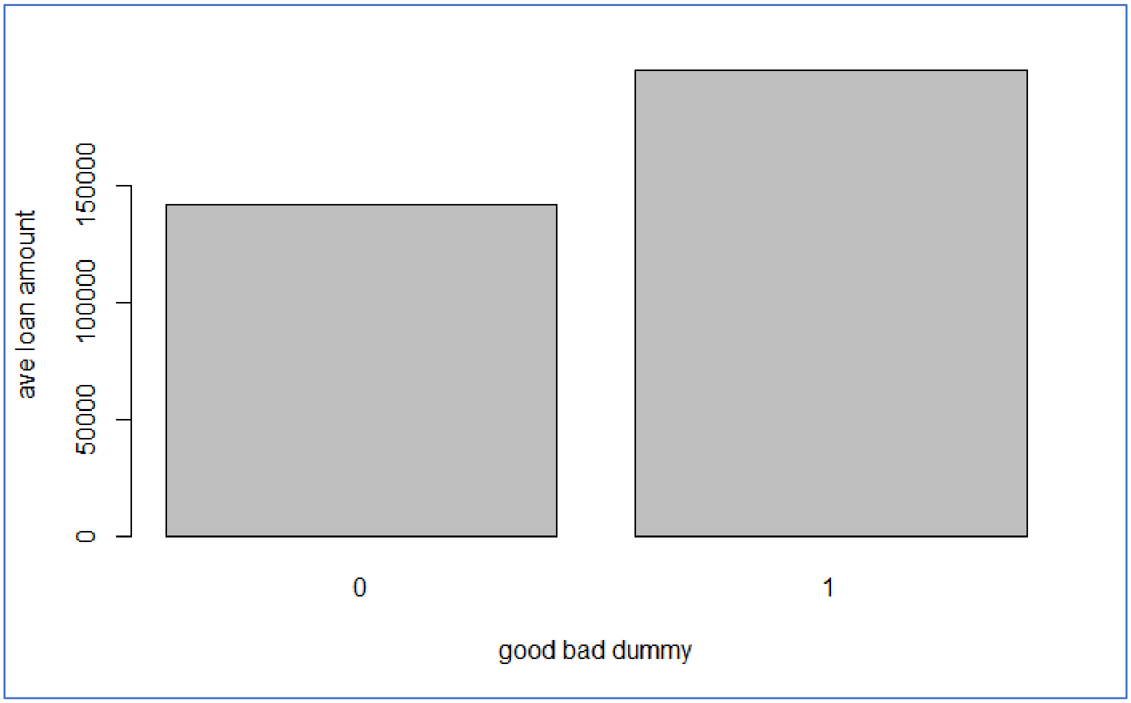


This graph show there is no difference between average salary for good loans versus bad loans.

	bad_good	meanSalary
1	0	9551.856
2	1	9543.570

Note there is a slight difference between the good (0) loans and bad (1) loans.

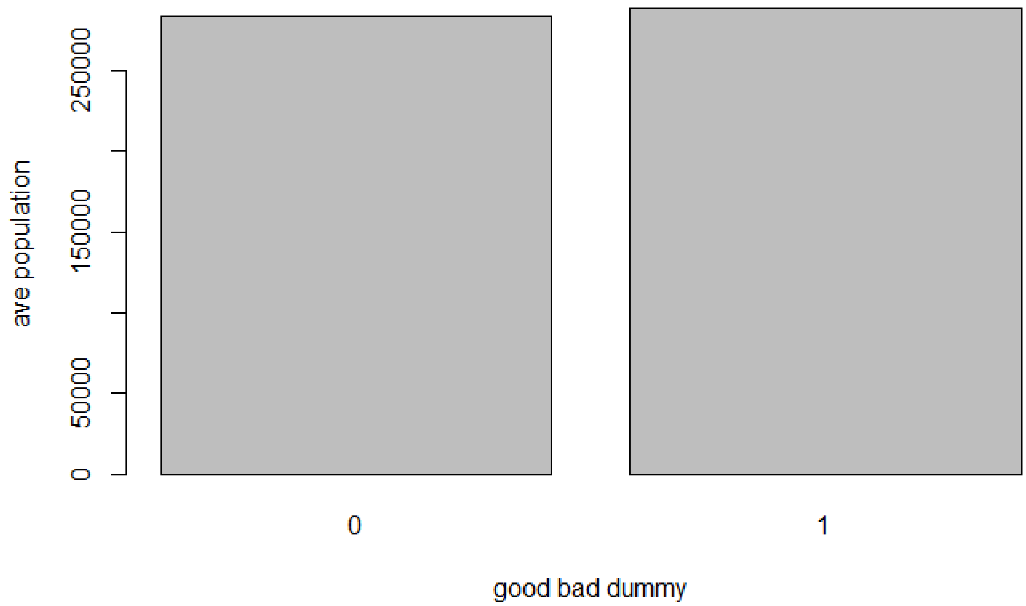
Mean loan amount



	bad_good	meanLoanAmount
1	0	141482.1
2	1	199457.2

Note mean salary is the same, but defaulters are taking more on the loan and unable to keep up payments.

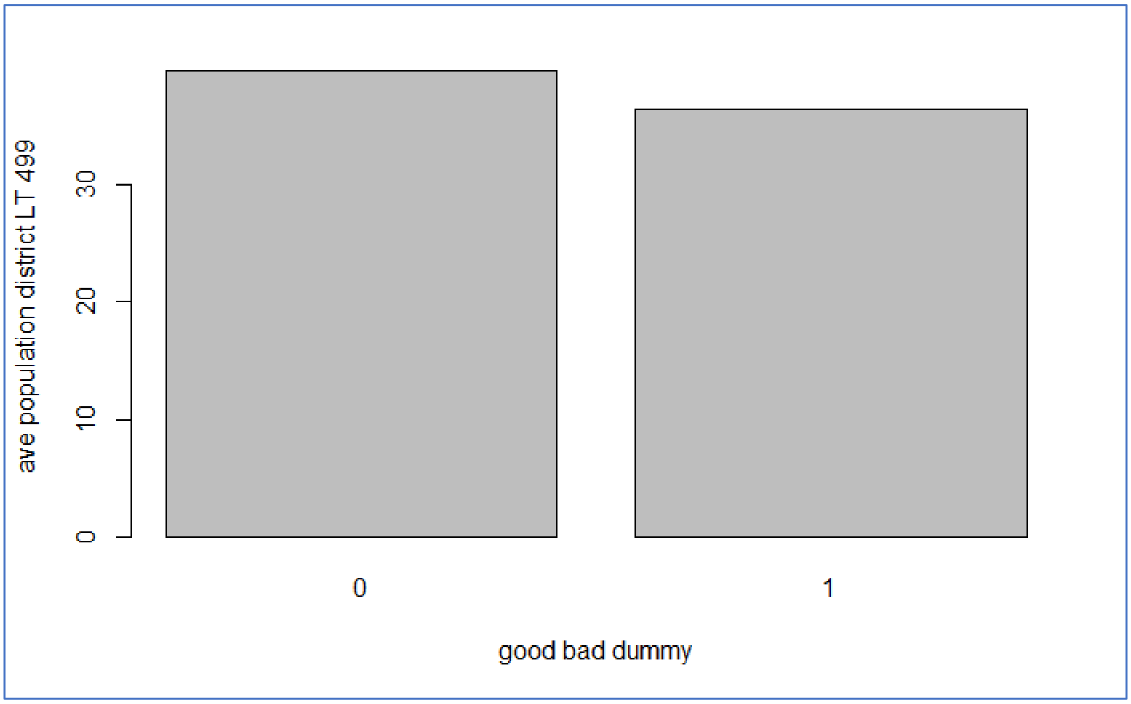
Average population



	bad_good	meanPop
1	0	284365.4
2	1	289103.8

More inhabitants with default population, but does not look significant.

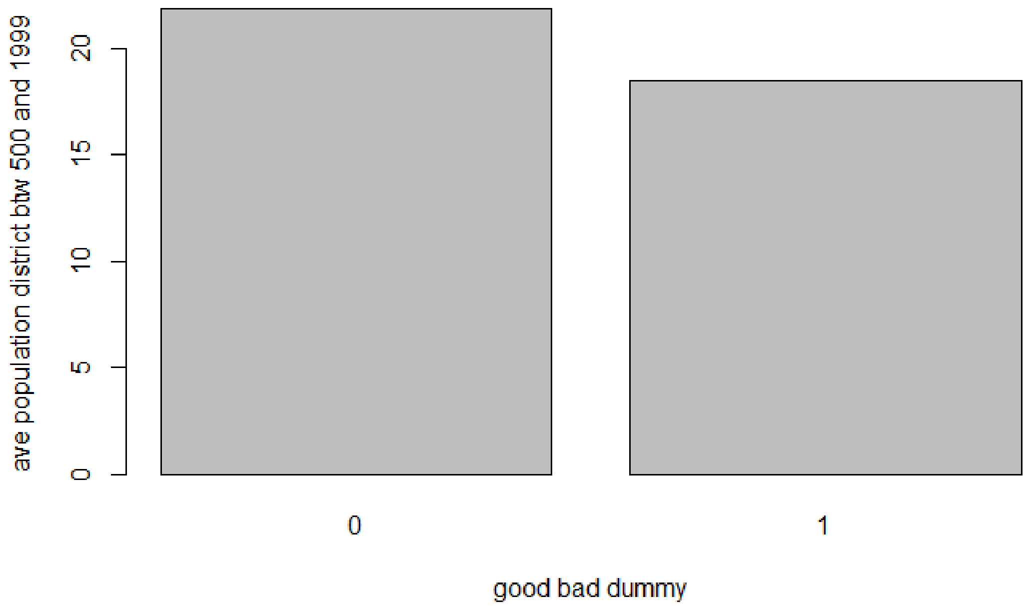
District municipalities with less than 499 inhabitants.



	bad_good	meanPopLT499
1	0	39.75092
2	1	36.47518

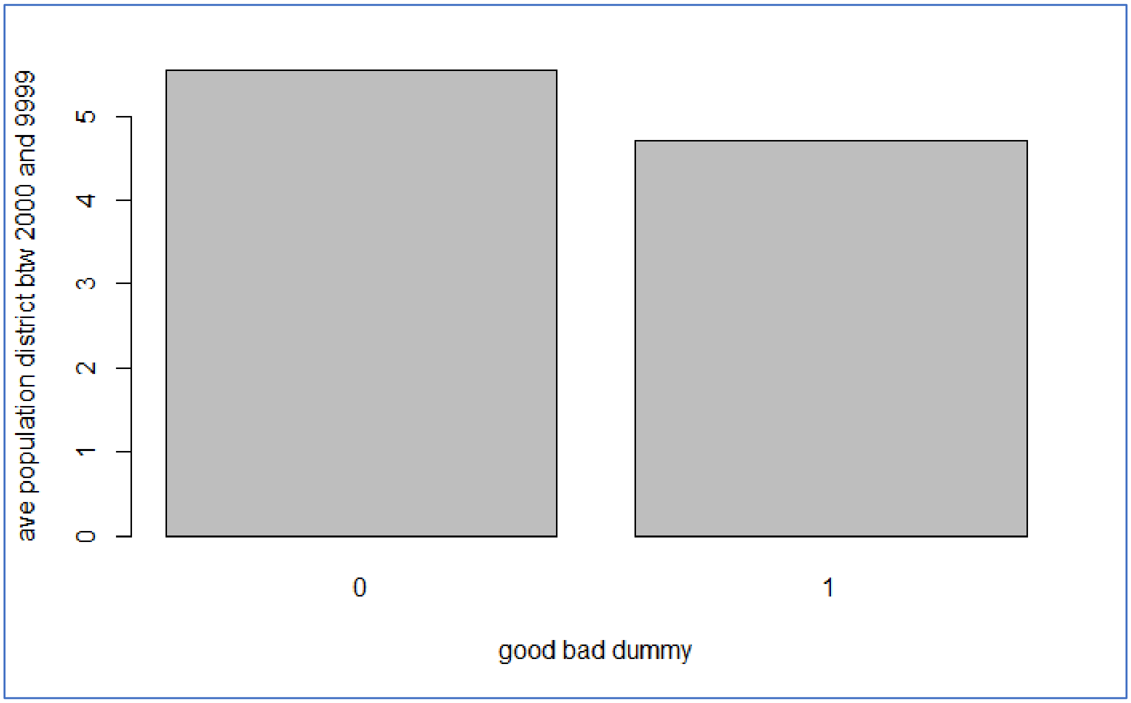
Districts with municipalities having more than 499 inhabitants fair better with loan default than districts with fewer than 499 inhabitants.

Districts with municipalities that have inhabitants between 500 and 1999 inhabitants.



	bad_good	meanPop500_1999
1	0	21.86236
2	1	18.48533

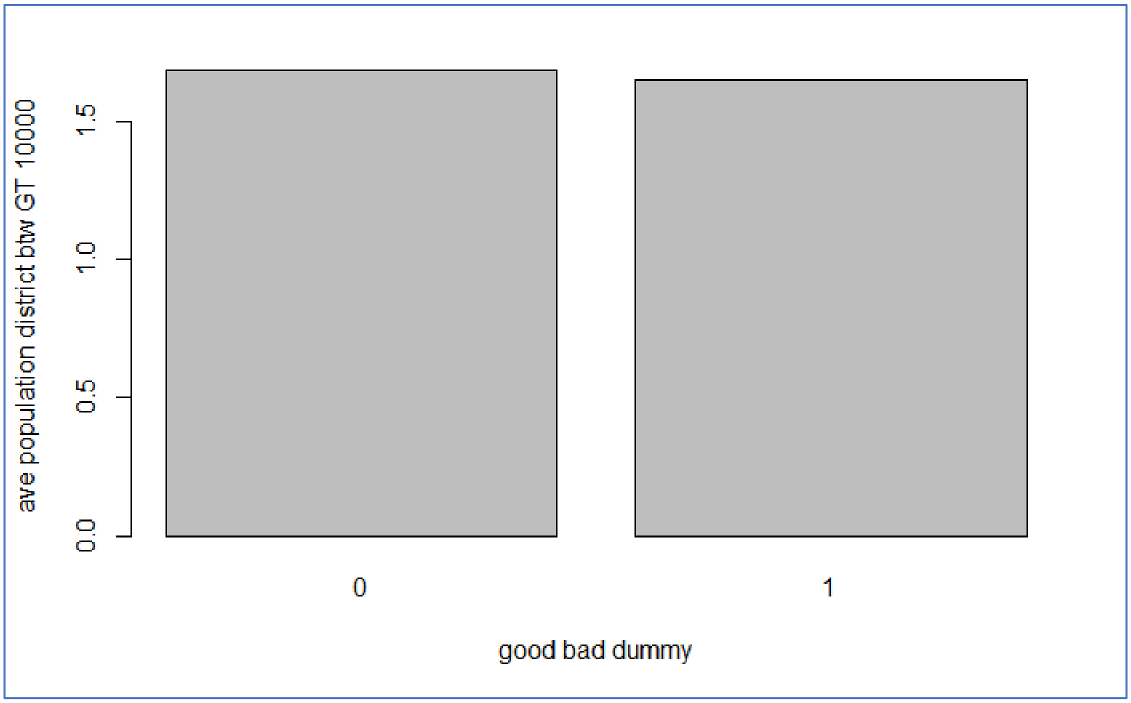
More municipalities with population from 500 to 1999 have more good loans than bad.



	bad_good	meanPop2000_9999
1	0	5.550873
2	1	4.711863

More municipalities with population from 2000 to 9999 have more good loans than bad.

Districts with municipalities greater than 10000



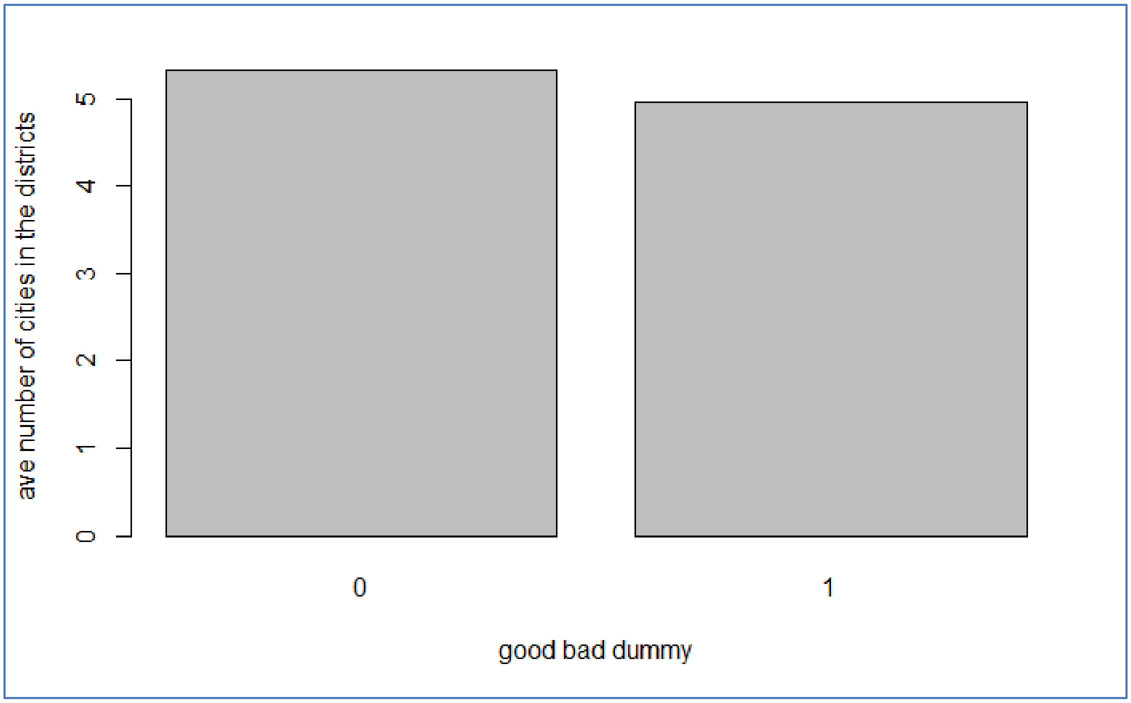
Not many municipalities greater than 10000 and they split fairly evenly with good and bad loans.

	bad_good	meanPopGT_10000
1	0	1.686063
2	1	1.652417

Districts with municipalities that have inhabitants GT 10000 are split evenly between good and bad loans.

Management should enforce stricter underwriting policies in metropolitan areas.

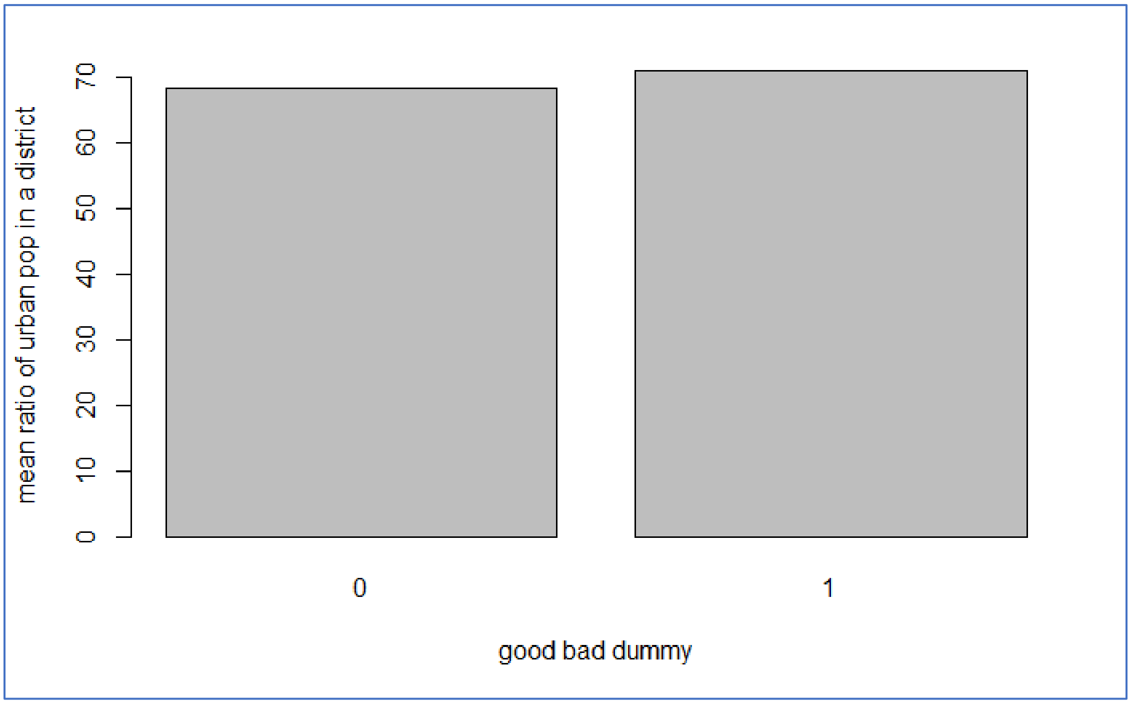
Mean number of cities in a district.



	bad_good	meanNumCities
1	0	5.328325
2	1	4.953339

Less defaults are seen in districts with more cities.

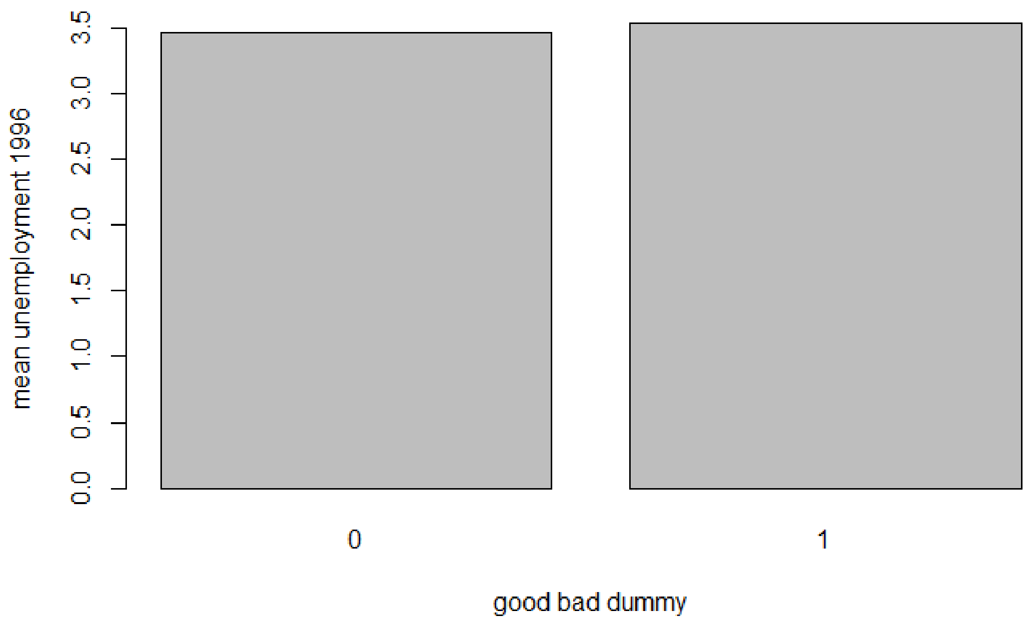
The higher the urban population the more defaults.



	bad_good	meanRatioUbanPop
1	0	68.27870
2	1	71.12029

Districts with higher urban population have more defaults

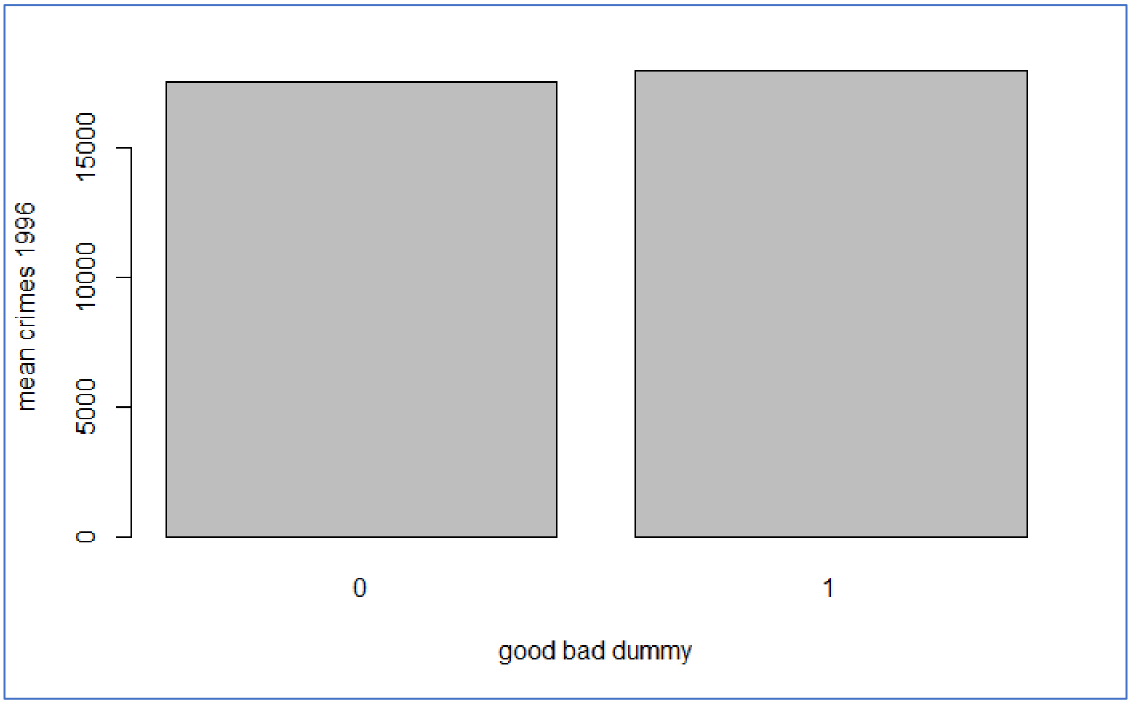
Unemployment rate in 1996



	bad_good	meanunemp1996
1	0	3.461161
2	1	3.537966

There is a slight difference with more defaults in higher unemployment areas.

Crimes in 1996



	bad_good	meancrimes1996
1	0	17539.25
2	1	17984.95

There is a higher number of defaults in districts where there is more crime.

Summary of Data Exploration

A data search was conducted by Group 4, and the websites and data descriptions are attached in the document, and Group 4 met through September 2018 and October 2018, and decided on a database that, unfortunately, was a textbook example. Group 4 renewed the search, and decided on a data set that was real bank data (masked for confidentiality), that contained over a million transactions, and had eight component data frames.

The data was downloaded, organized, input into R, and validated by members of the Group 4 team with a total of 1262625 records without the order data frame which inflated the records to twice its current size without adding information.

The data is described in the document and 77 districts were reported with demographic data to describe individuals within a region, and this is standard procedure for most data used for loss modeling and marketing purposes, in fact, some companies (Claritas) provide demographic data by zipcode to augment the model results.

Determine the data mining task.

The objective given Group 4 analysts is to determine the an appropriate lift for a marketing department to examine response to accommodate both the marketing department, and possibly a probability of default model for problematic loans for the Risk Group at the bank, and to demonstrate the potential for data mining to augment the companies marketing plan, and to this end analyst examine logistic regression and LDA as potential methodologies for developing the probability of default model. In support of the default modeling analyst employ clustering analysis to examine potential default clusters.

To supplement the banks marketing effort, analysts use associative rules to examine potential marketing lift.

Analyst also examine dimension reduction tools to collapse the data while retaining a significant amount of the variation in the variance-covariance matrix.

First up is data reduction.

Reduce the data dimension (if needed)

The typical data reduction is associated with principal components, however, clustering and factor analysis also are used industry wide as variable reduction tool, for example clustering variable data can produce cluster where one variable dominates, or a distance (similarity) matrix is used to cluster the variables, and a subject matter expert picks a preferred variable from each cluster for use in the analysis. Factor analysis uses correlation to model a reduced number of variables in a similar fashion as PCA, but PCA is a mathematical technique, and factor analysis is a modeling technique. For purposes of the proof of concept exercise PCA will be used where the maximum variation is extracted from the variance – covariance matrix, and orthogonal vectors (eigenvectors) associated with an eigen value.

Typically, behavior variables or demographic variables are clustered to reduce the mathematical burden on a procedure, or when looking at big data with a large number of variables let's say 300 to 1000 variables reduction will allow a procedure to explore more salient relationships.

The variables will be used from the demographic data for exposition purposes.

[1] "num_of_inhabitants"
[2] "num_of_municipalities_with_inhabitants_LT_499"
[3] "num_of_municipalities_with_inhabitants_500_1999"
[4] "num_of_municipalities_with_inhabitants_2000_9999"
[5] "num_of_municipalities_with_inhabitants_GT_10000"
[6] "num_of_cities"
[7] "Ratio_of_urban_inhabitants"
[8] "unemployment_rate_in_1995"
[9] "num_of_crimes_commited_in_1995"
[10] "Average_Salary_demographic"
[11] "bad_good_dummy"

Now the principal component procedure will examine linear combination of the above 11 variables.
What is Principal Components Analysis (PCA)?

It is a useful component in order to reduce the dimension in the case of larger number of variables. It creates new variables which are weighted linear combinations of the original variables, and that retain the majority of the information of the full original set. PCA is intended to use with numerical variables.

In order to get an effective result all, the variables need to be in the same unit. To produce variables with the same unit, it has to be normalized. Using normalization each variable is replaced by the standardized version of the variable with the unit as variance.

In our analysis we found our variables to be in different units such as dollars, average salary, percentages and other units.

We performed the normalization along with PCA so that units of measurements do not affect the PCA.

PC1-PC4 constitutes 86% of total variation associated with all 10 of the original variables. This suggests that we can capture most of the variability in the data with less than 50% of the original dimensions in the data.

- The first principal component (PC1) is measuring the variable 'num_of_crimes_commited_in_1995' tuition from the dataset as it is dominated in that group.
- The second principal component (PC2) is measuring the variable 'num_of_municipalities_with_inhabitants_GT_10000' tuition from the dataset as it is dominated in that group.
- The third principal component (PC3) is measuring the variable 'num_of_municipalities_with_inhabitants_2000_9999' tuition from the dataset as it is dominated in that group.
- The fourth principal component (PC4) is measuring the variable 'num_of_municipalities_with_inhabitants_500_1999' tuition from the dataset as it is dominated in that group.

Analysts eventually decided to use all the variable without principle components, since PC1-PC4 did not significantly reduce the number of variables compared with the information from all the variables.

Now the results are presented below where each PC1 – PC11 represents a vector used in the calculation determining the linear combination, so for example, if PC1 and PC2 are taken 69% of the variation in the variance – covariance matrix is explained.

	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
Standard deviation	2.2963	1.2796	1.0198	0.77740	0.73915	0.5966	0.51033	0.40795	0.33841	0.03846
Proportion of Variance	0.5273	0.1638	0.1040	0.06044	0.05463	0.0356	0.02604	0.01664	0.01145	0.00015
Cumulative Proportion	0.5273	0.6911	0.7951	0.85548	0.91012	0.9457	0.97176	0.98840	0.99985	1.00000

The first 4 principal components are listed below, and explain 86% of the data.

	PC1	PC2	PC3	PC4
num_of_inhabitants	0.3927166	0.06336165	-0.286757959	0.2286966
num_of_municipalities_with_inhabitants_LT_499	-0.2590971	0.35506781	0.403286946	0.5772649
num_of_municipalities_with_inhabitants_500_1999	-0.3508814	0.12765689	-0.325312967	0.2192766
num_of_municipalities_with_inhabitants_2000_9999	-0.3027946	-0.08485293	-0.606298865	-0.1856322
num_of_municipalities_with_inhabitants_GT_10000	-0.1080394	-0.63999295	-0.048529583	0.5677564
num_of_cities	-0.3312169	-0.12615794	-0.300468473	0.1184768
Ratio_of_urban_inhabitants	0.3615169	-0.28263712	0.005192786	-0.1321177
unemployment_rate_in_1995	-0.1839076	-0.55099715	0.345339797	-0.1971316
num_of_crimes_commited_in_1995	0.3958067	0.09025229	-0.256262212	0.2239573
Average_Salary_demographic	0.3482577	-0.17111976	-0.064104489	0.2984369

To form the PCA, analysts look at the values of each variable and multiply by the weight in the principal component.

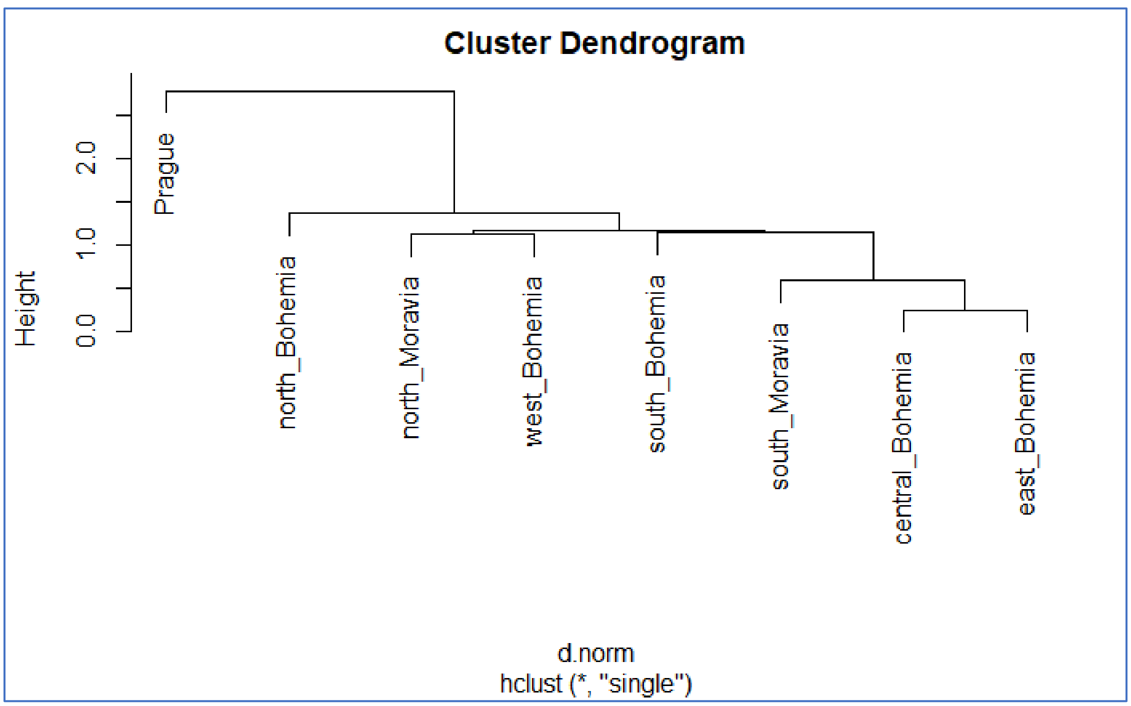
The number of variables is not so large as to warrant variable reduction this exercise just shows how the PCA is accomplished and satisfies the proof of concept assumption.

Probability of Default Support.

Cluster analysis gives insight to analysts as to how similar aspects of the data group together, the data was organized around region for an understanding how the mean number of inhabitants of the districts and the number of small municipalities in the district, and finally the default (bad_good_dummy) cluster the data.

	mean_num_of_inhabitants	mean_num_of_municipalities_with_inhabitants_LT_499	bad_good_dummy
central_Bohemia	90645.77	60.33553	0.08834638
east_Bohemia	114017.77	65.38302	0.08345298
north_Bohemia	114115.07	32.79823	0.02602504
north_Moravia	225036.56	13.84455	0.13538023
Prague	1204953.00	0.00000	0.09623602
south_Bohemia	83458.14	64.89809	0.13455058
south_Moravia	178825.45	51.00212	0.07230762
west_Bohemia	85704.70	38.26048	0.14999310

The data table is shown above (before scaling) for each of the regions, and after scaling the data and running the single linkage cluster procedure (minimum distance) the dendrogram is presented below.

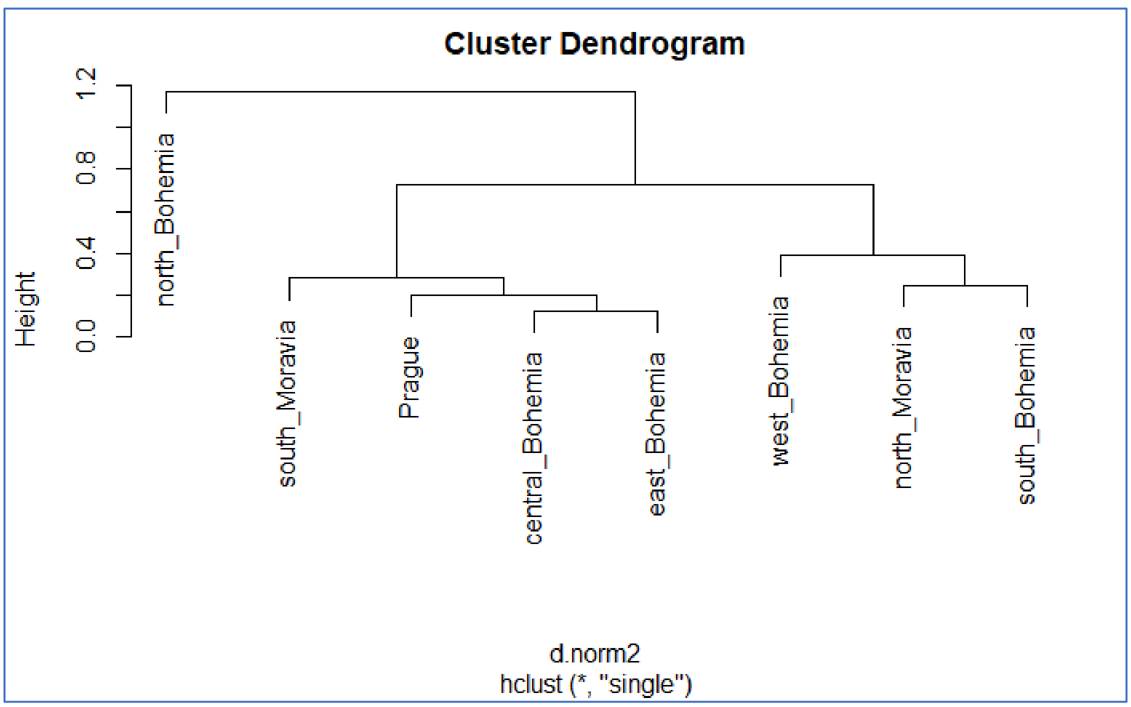


Prague is different than the other regions, and even though analysts are able to break out further cluster the dominant relationship is Prague against the other regions.

Now analysts examined looking at the just the default data and when taking the average this is the relative frequency approach to estimating the default probability in the regions, and notice the lowest default probability is North Bohemia (PD = 0.026), and the highest default area is South Bohemia (PD = 0.135).

	good_bad
central_Bohemia	0.08834638
east_Bohemia	0.08345298
north_Bohemia	0.02602504
north_Moravia	0.12483355
Prague	0.09623602
south_Bohemia	0.13455058
south_Moravia	0.07230762
west_Bohemia	0.14999310

Now scaling the default estimate in the single column and again using single linkage analysts obtain the following dendrogram.



This is very interesting analysts observe the North Bohemia is separate from the other regions, and South Moravia, Prague, Central Bohemia, and East Bohemia cluster together and West Bohemia, North Moravia and South Bohemia form the final cluster (height approximately 0.5). Looking at the default probabilities, analysts see that North Bohemia has the least risk, South Moravia, Prague, Central Bohemia, and East Bohemia (form a mid-range risk) and West Bohemia, North Moravia and South Bohemia form a cluster of extreme risk (West Bohemia is 15% default rate) this is not a cluster where weak underwriting terms and conditions are allowed. Management needs strict and strong underwriting.

Associative Rules for Marketing Support.

Associative rules play an important part in analyzing transaction data, and are used by companies to market items purchased together where a vendor would present items to a customer checking out or exploring products to purchase that are frequently purchased or to offer discounts for other purchases. The bank data has transaction data, and analyst used the Apriori program to ferret out relationships between the various field both for products and consumer behavior.

The evaluation of rules actually follows a mathematical construct that examines support (frequency), confidence, and lift. Support is the number of times the product or service appears in the database, so for example if loan is purchased then a credit card is purchased the frequency of the itemset (loan, credit card in the database, for example, if there are 10 records and loan and credit card appear in 2 times then the support is 20%. The mathematical definition is associated with what is called an *if then statement*, i.e., If A then B, and in the association literature (and mathematical logic) A is called the antecedent and B is called the consequent, and analysts confidence is increased if a customer has purchased B, and the frequency of A is high in the database, mathematically we have P(A|B), read the probability of A given B, and in the association literature the relationship becomes P(consequent | antecedent), read given the antecedent what is the frequency of the consequent.

Lastly, lift is the scaled confidence where the scaling factor is when the antecedent and the consequent are independent which mathematically is defined as P(A and B) = P(A) P(B) so that P(A|B) is just the P(A), and in the associative literature, P(consequent|antecedent) = P(consequent), that is, the frequency of the consequent in the database, and the popular name for this condition is *benchmark confidence.*

So, the lift ratio is defined as follows:

$$Lift\ ratio = \frac{confidence}{benchmark\ confidence}$$

In the following table, analyst use the support and lift to create associative rules.

The categories for the associative rules are as follows.

k_symbol
"POJISTNE" stands for insurance payment
"SIPO" stands for household
"LEASING" stands for leasing
"UVER" stands for loan payment

transaction type.x
"PRIJEM" stands for credit
"VYDAJ" stands for withdrawal

transaction operation
"VYBER KARTOU" credit card withdrawal
"VKLAD" credit in cash
"PREVOD Z UCTU" collection from another bank
"VYBER" withdrawal in cash
"PREVOD NA UCET" remittance to another bank

Categorical variables were created for the above variables, and loaded into the Apriori algorithm which looks at the frequency of the item sets with just one item, then with all the one item sets in mind, look at the two item sets, and so forth, and this substantially decreases the combinations the algorithm needs to try.

The data was split into training and validation sets on an 80-20 basis, and the as the table indicates 39 itemsets have a lift greater than 1.25, and looking at the first four association rules the lift is greater than 2.25 which is a reasonable standard.

The first four rules in the training set are shown below:

	lhs <fctr>	<fctr>	rhs <fctr>	support <dbl>	confidence <dbl>	lift <dbl>	count <dbl>
[1]	{type.xVYDAJ,bankNone,typeNone}	=>	{operationVYBER}	0.3259479	0.9969266	2.431194	329240
[2]	{type.xVYDAJ,bankNone}	=>	{operationVYBER}	0.3945867	0.9815473	2.393688	398572
[3]	{type.xVYDAJ,k_symbolNone,bankNone}	=>	{operationVYBER}	0.2425473	0.9703274	2.366327	244997
[4]	{type.xVYDAJ,k_symbolNone}	=>	{operationVYBER}	0.2425473	0.9383766	2.288409	244997

Note: Type.x is from the transactional database, bank is a two-letter code for 15 partner banks, the k_symbol is from the transactional database and characterizes the transaction.

The first rule indicates that if the transaction is in cash, there in no entry for bank partner, and no entry for transaction type that the operation is VYBER, withdrawal in cash, and this represents an opportunity for the bank, and the relation hold in the validation set which follows the 50 training rules, and the motivation for the opportunity is enumerated below.

	lhs <fctr>	<fctr>	rhs <fctr>	support <dbl>	confidence <dbl>	lift <dbl>	count <dbl>
[1]	{type.xVYDAJ,bankNone,typeNone}	=>	{operationVYBER}	0.3259479	0.9969266	2.431194	329240
[2]	{type.xVYDAJ,bankNone}	=>	{operationVYBER}	0.3945867	0.9815473	2.393688	398572
[3]	{type.xVYDAJ,k_symbolNone,bankNone}	=>	{operationVYBER}	0.2425473	0.9703274	2.366327	244997
[4]	{type.xVYDAJ,k_symbolNone}	=>	{operationVYBER}	0.2425473	0.9383766	2.288409	244997
[5]	{amount_trans,bankNone}	=>	{k_symbolNone}	0.2042897	0.9935577	2.201300	206353
[6]	{amount_trans}	=>	{k_symbolNone}	0.2210890	0.8848606	1.960474	223322
[7]	{operationPREVOD NA UCET}	=>	{type.xVYDAJ}	0.2013177	1.0000000	1.657491	203351
[8]	{operationVYBER,typeNone}	=>	{type.xVYDAJ}	0.3259479	0.9670758	1.602920	329240
[9]	{operationVYBER,bankNone,typeNone}	=>	{type.xVYDAJ}	0.3259479	0.9670758	1.602920	329240
[10]	{type.xVYDAJ,typeNone}	=>	{operationVYBER}	0.3259479	0.6551531	1.597715	329240
[11]	{operationVYBER}	=>	{type.xVYDAJ}	0.3945867	0.9622741	1.594961	398572
[12]	{operationVYBER,bankNone}	=>	{type.xVYDAJ}	0.3945867	0.9622741	1.594961	398572
[13]	{type.xVYDAJ}	=>	{operationVYBER}	0.3945867	0.6540239	1.594961	398572
[14]	{k_symbolNone,bankNone,typeNone}	=>	{operationVYBER}	0.2102109	0.6477488	1.579658	212334
[15]	{operationVYBER,k_symbolNone}	=>	{type.xVYDAJ}	0.2425473	0.9400437	1.558114	244997
[16]	{operationVYBER,k_symbolNone,bankNone}	=>	{type.xVYDAJ}	0.2425473	0.9400437	1.558114	244997
[17]	{k_symbolNone,bankNone}	=>	{operationVYBER}	0.2580170	0.6302869	1.537074	260623
[18]	{k_symbolNone,typeNone}	=>	{operationVYBER}	0.2102109	0.5811972	1.417359	212334
[19]	{operationVYBER}	=>	{k_symbolNone}	0.2580170	0.6292232	1.394090	260623
[20]	{operationVYBER,bankNone}	=>	{k_symbolNone}	0.2580170	0.6292232	1.394090	260623
[21]	{k_symbolNone}	=>	{operationVYBER}	0.2580170	0.5716557	1.394090	260623
[22]	{bankNone,typeNone}	=>	{operationVYBER}	0.3370448	0.5668604	1.382396	340449
[23]	{operationVYBER,typeNone}	=>	{k_symbolNone}	0.2102109	0.6236881	1.381827	212334
[24]	{operationVYBER,bankNone,typeNone}	=>	{k_symbolNone}	0.2102109	0.6236881	1.381827	212334
[25]	{type.xVYDAJ,bankNone}	=>	{k_symbolNone}	0.2499644	0.6217945	1.377632	252489
[26]	{operationVYBER}	=>	{bankNone}	0.4100564	1.0000000	1.362358	414198
[27]	{bankNone}	=>	{operationVYBER}	0.4100564	0.5586437	1.362358	414198
[28]	{operationVYBER,k_symbolNone}	=>	{bankNone}	0.2580170	1.0000000	1.362358	260623
[29]	{type.xVYDAJ,operationVYBER}	=>	{bankNone}	0.3945867	1.0000000	1.362358	398572
[30]	{operationVYBER,typeNone}	=>	{bankNone}	0.3370448	1.0000000	1.362358	340449
[31]	{type.xVYDAJ,operationVYBER,k_symbolNone}	=>	{bankNone}	0.2425473	1.0000000	1.362358	244997
[32]	{operationVYBER,k_symbolNone,typeNone}	=>	{bankNone}	0.2102109	1.0000000	1.362358	212334
[33]	{type.xVYDAJ,operationVYBER,typeNone}	=>	{bankNone}	0.3259479	1.0000000	1.362358	329240
[34]	{type.xVYDAJ,operationVYBER}	=>	{k_symbolNone}	0.2425473	0.6146869	1.361884	244997
[35]	{type.xVYDAJ,operationVYBER,bankNone}	=>	{k_symbolNone}	0.2425473	0.6146869	1.361884	244997
[36]	{type.xVYDAJ,bankNone,typeNone}	=>	{k_symbolNone}	0.2001178	0.6120695	1.356085	202139
[37]	{type.xVYDAJ,k_symbolNone}	=>	{bankNone}	0.2499644	0.9670722	1.317499	252489
[38]	{type.xVYDAJ,k_symbolNone,typeNone}	=>	{bankNone}	0.2001178	0.9592049	1.306781	202139
[39]	{amount_trans,k_symbolNone}	=>	{bankNone}	0.2042897	0.9240155	1.258840	206353
[40]	{k_symbolNone}	=>	{bankNone}	0.4093644	0.9069771	1.235628	413499
[41]	{bankNone}	=>	{k_symbolNone}	0.4093644	0.5577009	1.235628	413499
[42]	{k_symbolNone,typeNone}	=>	{bankNone}	0.3245253	0.8972571	1.222385	327803
[43]	{bankNone,typeNone}	=>	{k_symbolNone}	0.3245253	0.5458044	1.209270	327803
[44]	{type.xPRIJEM}	=>	{bankNone}	0.3165469	0.8303766	1.131270	319744
[45]	{amount_trans}	=>	{bankNone}	0.2056143	0.8229264	1.121121	207691
[46]	{type.xPRIJEM,typeNone}	=>	{bankNone}	0.2565320	0.8125398	1.106970	259123
[47]	{k_symbolNone,bankNone,typeNone}	=>	{type.xVYDAJ}	0.2001178	0.6166478	1.022088	202139
[48]	{k_symbolNone,bankNone}	=>	{type.xVYDAJ}	0.2499644	0.6106157	1.012090	252489
[49]	{type.xPRIJEM}	=>	{typeNone}	0.3157163	0.8281978	1.004695	318905
[50]	{type.xVYDAJ,operationVYBER}	=>	{typeNone}	0.3259479	0.8260490	1.002089	329240

The validation data yields something similar to the training data above.

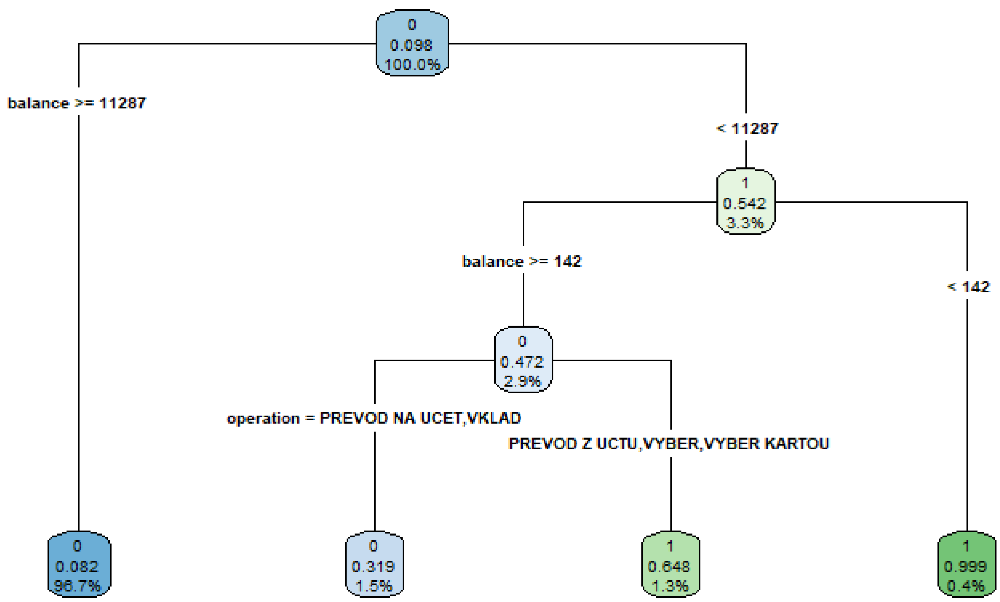
	lhs <fctr>	<fctr>	rhs <fctr>	support <dbl>	confidence <dbl>	lift <dbl>
[1]	{type.xVYDAJ,bankNone,typeNone}	=>	{operationVYBER}	0.3241224	0.9973072	2.449068
[2]	{type.xVYDAJ,bankNone}	=>	{operationVYBER}	0.3919295	0.9823426	2.412320
[3]	{type.xVYDAJ,k_symbolNone,bankNone}	=>	{operationVYBER}	0.2416553	0.9716733	2.386119
[4]	{type.xVYDAJ,k_symbolNone}	=>	{operationVYBER}	0.2416553	0.9387874	2.305362
[5]	{amount_trans,bankNone}	=>	{k_symbolNone}	0.2048708	0.9935664	2.202658

The top four consequents are VYBER, that is, withdraw in cash. There are three reasons in Keynesian economics for holding cash the precautionary motive for individuals that are expecting to have a large number of transactions, the speculative motive, for individuals that are going to potentially invest in a better market, and transaction motive where individuals are going to make a large number of purchases. The fact that VYDAJ stands for transaction withdrawal, but individuals are looking to take out cash, and this represents an opportunity for the bank to identify these individuals and market a credit card with first a loss leader then bring the customer on slowly to be a valued credit card holder. The lift on all the transactions is greater than 2.25 which indicates compared to the benchmark confidence (when antecedent and consequent are independent) the confidence P(consequent|antecedent) is 2.25 or greater than the benchmark confidence.

CART

The Classification and regression trees algorithm was created in 1984 to give an alternative classification model that was distribution free, and appropriate for large sample data.

This tree is reasonably simple where we have:



If the transaction balance is greater than 11287 then it is a good loan, and if the balance is less than 11287 and balance is less than 142 it 1 with the other categories enumerated.

The confusion matrix follows which is an evaluation of the tree.

```
          0      1
 0 167772  15811
 1    880   2438

               Accuracy : 0.9107
                 95% CI : (0.9094, 0.912)
    No Information Rate : 0.9024
    P-Value [Acc > NIR] : < 0.00000000000000022

                  Kappa : 0.2021
 Mcnemar's Test P-Value : < 0.00000000000000022

            Sensitivity : 0.9948
            Specificity : 0.1336
         Pos Pred Value : 0.9139
         Neg Pred Value : 0.7348
             Prevalence : 0.9024
         Detection Rate : 0.8977
   Detection Prevalence : 0.9822
      Balanced Accuracy : 0.5642

       'Positive' Class : 0
```

Notice that the 'positive' class is 0 which means sensitivity is for the 0 class with is quite accurate but the 1 class of interest is the specificity at 0.1336 which is not satisfactory.

Analysts oversampled looking at all the 1's and then a matching sample of 0's
On 80% of the training data. The 1's total 18122 and the 0's total 16954

The analysis is in chunk 41 where the zeros are samples at 10% to match the 1's which are 10% with cp=0.02 the following statistics are obtained. Even with oversampling the statistics are not good for a tree that is easy to read.

```
Confusion Matrix and Statistics


          0     1
  0 14667  8170
  1  2287  9952

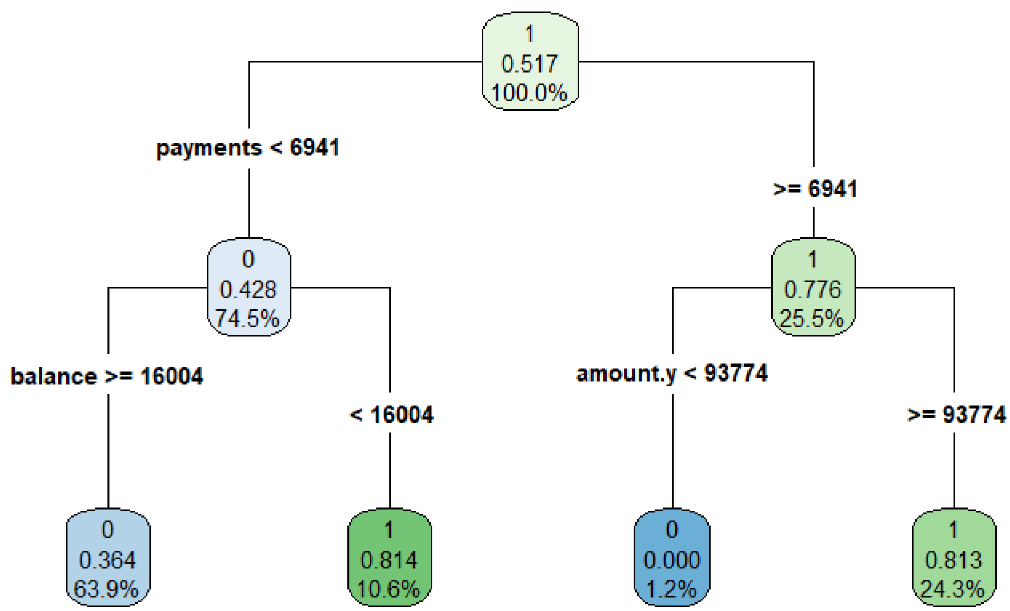
               Accuracy : 0.7019
                 95% CI : (0.6971, 0.7067)
    No Information Rate : 0.5166
    P-Value [Acc > NIR] : < 0.00000000000000022

                  Kappa : 0.4097
 Mcnemar's Test P-Value : < 0.00000000000000022

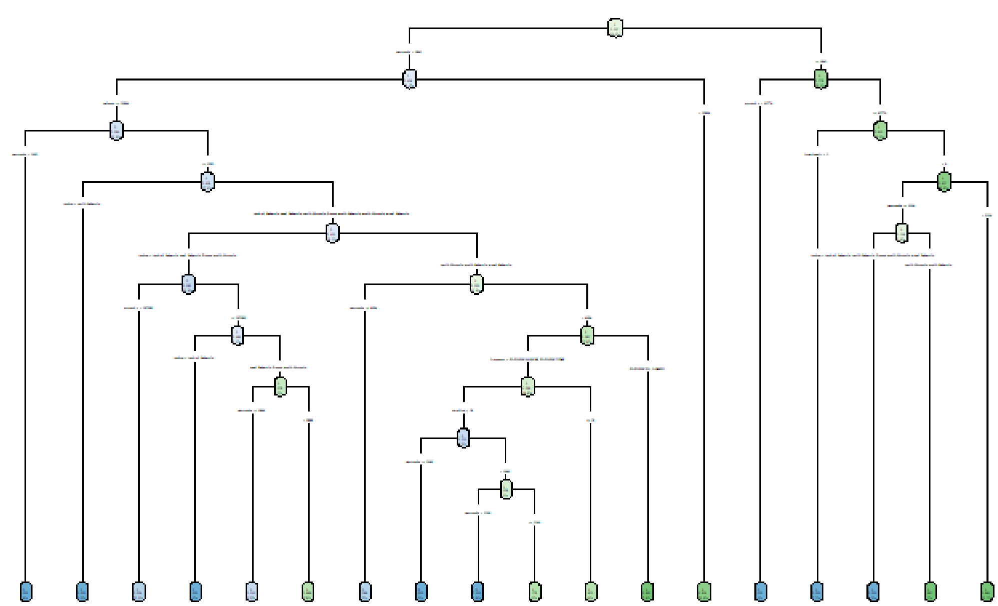
            Sensitivity : 0.8651
            Specificity : 0.5492
         Pos Pred Value : 0.6422
         Neg Pred Value : 0.8131
             Prevalence : 0.4834
         Detection Rate : 0.4181
   Detection Prevalence : 0.6511
      Balanced Accuracy : 0.7071

       'Positive' Class : 0
```

This is the tree associated with the oversampling.



At a cp = 0.01 the sensitivity and specificity improve, but tree complexity increases.



The tree statistics improve.

```
Confusion Matrix and Statistics


       0     1
  0 13381  3164
  1  3573 14958

               Accuracy : 0.8079
                 95% CI : (0.8038, 0.812)
    No Information Rate : 0.5166
    P-Value [Acc > NIR] : < 0.00000000000000022

                  Kappa : 0.6151
 Mcnemar's Test P-Value : 0.0000006667

            Sensitivity : 0.7893
            Specificity : 0.8254
         Pos Pred Value : 0.8088
         Neg Pred Value : 0.8072
             Prevalence : 0.4834
         Detection Rate : 0.3815
   Detection Prevalence : 0.4717
      Balanced Accuracy : 0.8073

       'Positive' Class : 0
```

Responders are 10% → 50% of sample → 5 responders

Nonresponders are 90% → 50% of sample → 0.556 nonresponders

Nonresponders
0 →13381/0.556 = 24086

1 → 3573/0.556 = 6426

Responders
0 → 3164/5 = 633

1 → 14958/5 = 2992

So, sensitivity is 2992/3625 = 82.5% with target 1
And, specificity 24086/ 3625 = 79%
Accuracy = 27078 /34137 = 79%
Not unreasonable, but the tree is difficult to interpret, and analysts will not look at the validation set.

Logistics Regression

Lots of variables but easier to interpret than the tree.

The regression formulation has the log odd as a function of the X's and Beta's and can be estimate with Ordinary Least Squares or Maximum Likelihood.

$$ln\left(\frac{p}{1-p}\right) = X\beta$$

Note the following interpretation for the odds ratio.

$$\frac{odds(x_1{+}1, .., x_n)}{odds(x_1, .., x_n)} = e^{\beta_1}$$

```
Call:
glm(formula = bad_good_dummy ~ ., family = "binomial", data = target.train.data.df)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4429  -0.4595  -0.3036  -0.1754   3.4610

Coefficients:
                                                 Estimate     Std. Error z value                Pr(>|z|)
(Intercept)                                 -1.61286989476  0.31816004319  -5.069  0.00000039914178615 ***
amount.x                                     0.00002239285  0.00000120874  18.526 < 0.0000000000000002 ***
balance                                     -0.00003694452  0.00000069175 -53.408 < 0.0000000000000002 ***
amount.y                                     0.00000119747  0.00000021987   5.446  0.00000005140790799 ***
duration                                     0.00168136872  0.00123310158   1.364             0.172716
payments                                     0.00021844715  0.00000901063  24.243 < 0.0000000000000002 ***
num_of_inhabitants                          -0.00000007534  0.00000072785  -0.104             0.917556
num_of_municipalities_with_inhabitants_LT_499    -0.00584988965  0.00052369882 -11.170 < 0.0000000000000002 ***
num_of_municipalities_with_inhabitants_500_1999   0.00674808437  0.00135591588   4.977  0.00000064653360645 ***
num_of_municipalities_with_inhabitants_2000_9999 -0.04696984519  0.00643436381  -7.300  0.00000000000028810 ***
num_of_municipalities_with_inhabitants_GT_10000  -0.19248731722  0.01607706689 -11.973 < 0.0000000000000002 ***
num_of_cities                                0.02296818108  0.00893305646   2.571             0.010136 *
Ratio_of_urban_inhabitants                   0.03509432902  0.00186763734  18.791 < 0.0000000000000002 ***
Average_Salary                              -0.00007140773  0.00003464454  -2.061             0.039288 *
unemployment_rate_in_1995                    0.69961698104  0.03713031014  18.842 < 0.0000000000000002 ***
unemployment_rate_in_1996                   -0.53601779571  0.03655450737 -14.664 < 0.0000000000000002 ***
num_of_enterpreneurs_per_1000_inhabitants   -0.01364539470  0.00097997786 -13.924 < 0.0000000000000002 ***
num_of_crimes_commited_in_1995              -0.00009470299  0.00003951926  -2.396             0.016558 *
num_of_crimes_commited_in_1996               0.00007815629  0.00004440763   1.760             0.078412 .
amount_trans                                 0.06890772966  0.02826174502   2.438             0.014761 *
balance_trans                                0.83371918890  0.03185900213  26.169 < 0.0000000000000002 ***
Average_Salary_demographic                  -1.20434546346  0.05496267211 -21.912 < 0.0000000000000002 ***
type.xPRIJEM                                 0.23886833583  0.02592030023   9.215 < 0.0000000000000002 ***
type.xVYBER                                  0.18624421994  0.04575338723   4.071  0.00004689008635813 ***
`k_symbolSANKC. UROK`                        3.64021819222  0.15865972821  22.944 < 0.0000000000000002 ***
k_symbolSIPO                                 0.52430698137  0.04468146996  11.734 < 0.0000000000000002 ***
k_symbolSLUZBY                               0.26186631680  0.03095891217   8.459 < 0.0000000000000002 ***
k_symbolUROK                                 0.14174262761  0.03399488640   4.170  0.00003052312384741 ***
```

```
k_symbolUVER                           1.01127591582  0.04776586295  21.172 < 0.0000000000000002 ***
bankAB                                -1.03019260904  0.08193112785 -12.574 < 0.0000000000000002 ***
bankCD                                -1.33115549380  0.09299319217 -14.315 < 0.0000000000000002 ***
bankEF                                -0.93336415096  0.07562136992 -12.343 < 0.0000000000000002 ***
bankGH                                -0.87116562872  0.07000750019 -12.444 < 0.0000000000000002 ***
bankIJ                                -1.00829811532  0.08144235819 -12.381 < 0.0000000000000002 ***
bankKL                                -1.17654549791  0.07693534188 -15.293 < 0.0000000000000002 ***
bankMN                                -0.29520778612  0.06392054189  -4.618  0.0000038679318883 ***
bankOP                                -0.03732453049  0.05972587471  -0.625           0.532017    
bankQR                                -1.48953407592  0.08604211526 -17.312 < 0.0000000000000002 ***
bankST                                -1.39476558220  0.08497235576 -16.414 < 0.0000000000000002 ***
bankUV                                -1.02053386143  0.07572343805 -13.477 < 0.0000000000000002 ***
bankWX                                -2.26046688065  0.13446463065 -16.811 < 0.0000000000000002 ***
bankYZ                                -1.12081363122  0.07923710912 -14.145 < 0.0000000000000002 ***
regioncentral_Bohemia                  0.15315957792  0.04912581047   3.118           0.001823 ** 
regioneast_Bohemia                    -0.21423611652  0.03928938248  -5.453  0.0000000495901253 8***
regionnorth_Bohemia                   -2.28526798560  0.06542246713 -34.931 < 0.0000000000000002 ***
regionPrague                           0.65785062956  0.84588368249   0.778           0.436741    
regionsouth_Bohemia                    0.42555985294  0.04068834738  10.459 < 0.0000000000000002 ***
regionsouth_Moravia                   -0.68091609174  0.04194541935 -16.233 < 0.0000000000000002 ***
typeclassic                           -1.17748297383  0.03638343204 -32.363 < 0.0000000000000002 ***
typegold                              -0.29567643148  0.08350097539  -3.541           0.000399 ***
typejunior                            -0.42145832325  0.05376252722  -7.839  0.00000000000000453 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 116851  on 185396  degrees of freedom
Residual deviance:  96493  on 185346  degrees of freedom
(1504 observations deleted due to missingness)
AIC: 96595

Number of Fisher Scoring iterations: 6
```

From the above output of the logistic regression model we observe that the below variables are not significant.

1. duration
2. num_of_inhabitants
3. num_of_crimes_commited_in_1996
4. bankOP
5. regionPrague

We see that the other coefficients are significant and interpret the results. Let us look at the variable type classic, the odds of classifying a customer loan status as bad who possess classic card is 30 percent less compared to customer who doesn't have any credit card.

We see that the coefficient of unemployment rate in 1995 is 0.6996. From this we can conclude that, if the unemployment rate increases by 1 percent, the odds of classifying the loan status as bad doubles as compared to classifying loan as good.

The coefficient of regionnorth_Bohemia is -2.2852, We may interpret that odds of classifying a customer loan status as bad who lives in region north Bohemia is 10 percent less compared to customer who lives in north Moravia.

The coefficient of bank YZ is -2.2604. So, the odds of classifying a customer loan status as bad who transferred money to bank YZ is 10 percent less who doesn't transfer money to any bank.

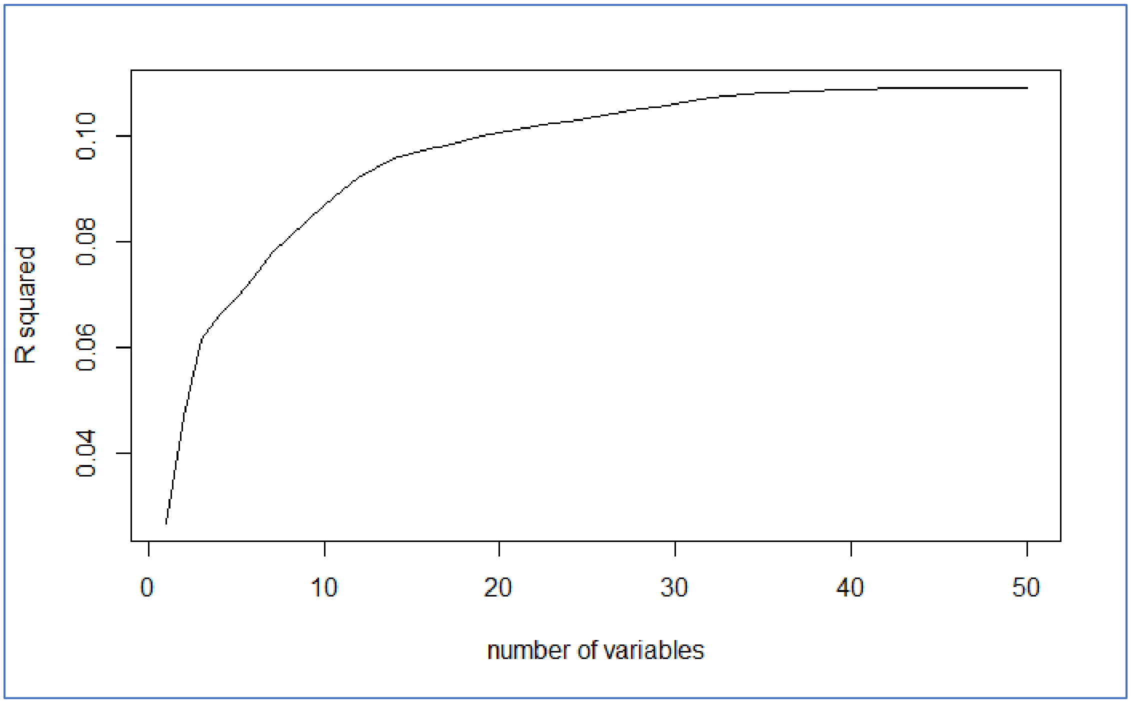
We ran the exhaustive search algorithm to get the best subset. The output of this below. We interpret this result by looking at the cp value. For selecting the best subset, cp value has to be equal to p+1 (= number of predictors + 1) and have small p value. From the output, the cp value that is equal to

```
> sum$rsq
 [1] 0.02678688 0.04704010 0.06169031 0.06608355 0.06943708 0.07339499 0.07785955 0.08106468 0.08412294 0.08701082 0.08975329 0.09220957 0.09416893
[14] 0.09575763 0.09681123 0.09759187 0.09840536 0.09919098 0.09998850 0.10072190 0.10140097 0.10193683 0.10244613 0.10293962 0.10349002 0.10411731
[27] 0.10466422 0.10514705 0.10560127 0.10612301 0.10675337 0.10722805 0.10753806 0.10804977 0.10824584 0.10842123 0.10857599 0.10871407 0.10882813
[40] 0.10891722 0.10901476 0.10908015 0.10913411 0.10918103 0.10921503 0.10923731 0.10925488 0.10926615 0.10927203 0.10927360
> sum$adjr2
 [1] 0.02678163 0.04702982 0.06167513 0.06606340 0.06941198 0.07336500 0.07782474 0.08102503 0.08407848 0.08696157 0.08969928 0.09215081 0.09410541
[14] 0.09568935 0.09673815 0.09751399 0.09832268 0.09910351 0.09989626 0.10062488 0.10129918 0.10183025 0.10233476 0.10282348 0.10336911 0.10399165
[27] 0.10453381 0.10501189 0.10546134 0.10597834 0.10660399 0.10707392 0.10737918 0.10788617 0.10807746 0.10824807 0.10839805 0.10853135 0.10864063
[40] 0.10872493 0.10881767 0.10887828 0.10892744 0.10896956 0.10899877 0.10901624 0.10902901 0.10903547 0.10903655 0.10903331
> sum$cp
 [1] 17117.17387 12904.80234  9858.32609  8946.16306  8250.34619  7428.76832  6501.76341  5836.82756  5202.45224  4603.53116  4034.86610  3525.75296
[13]  3120.04222  2791.45763  2574.22038  2413.78148  2246.50779  2085.03270  1921.08073  1770.47245  1631.16815  1521.66420  1417.68893  1316.99983
[25]  1204.47070  1075.94171   964.13857   865.66847   773.15341   666.58735   537.41964   440.64734   378.13809   273.65932   234.86056   200.36447
[37]   170.16123   143.42868   121.69459   105.15608    86.86086    75.25291    66.02462    58.26270    53.18676    50.55224    48.89576    48.55022
[49]    49.32750    51.00000
```

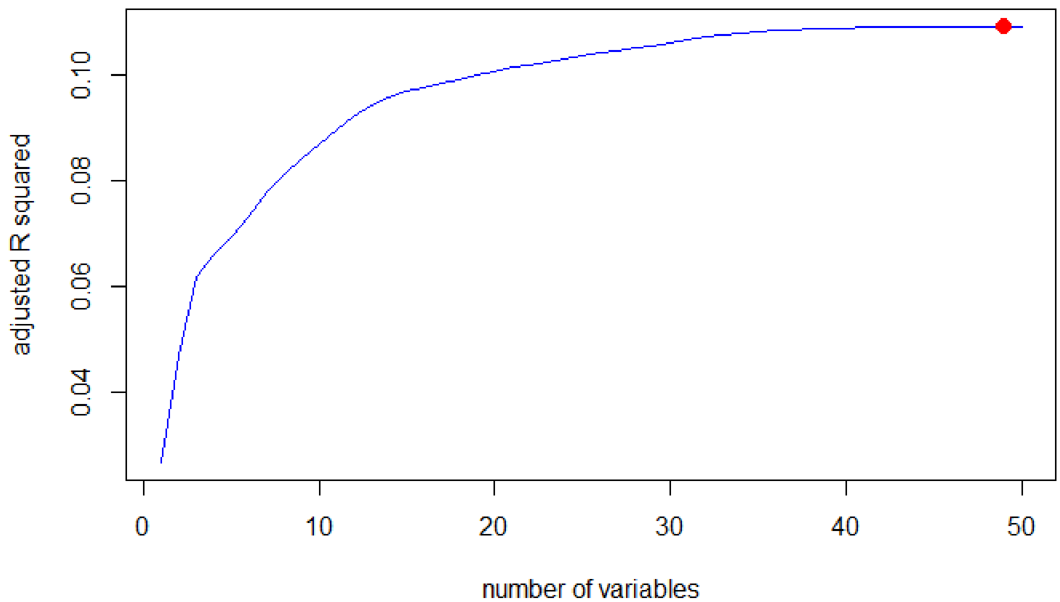
Notice the cp value moves into 48 as the number of variables that is optional for the analysis.

Next, observe the R square graph by the number of variables.

*



Notice the star above the graph for the number of variables using the R – Square goodness of fit criteria, analyst observe that the graph flattens out at about 40 variables. Next is the adjusted R-Square.



Again, notice the graph flattens at approximately 40 variables.

The consensus of the group was to take the cp value and put 48 variables in the model.

The confusion matrix is next.

```
Confusion Matrix and Statistics

          Reference
Prediction     0     1
         0 37901  2602
         1  4037  1823

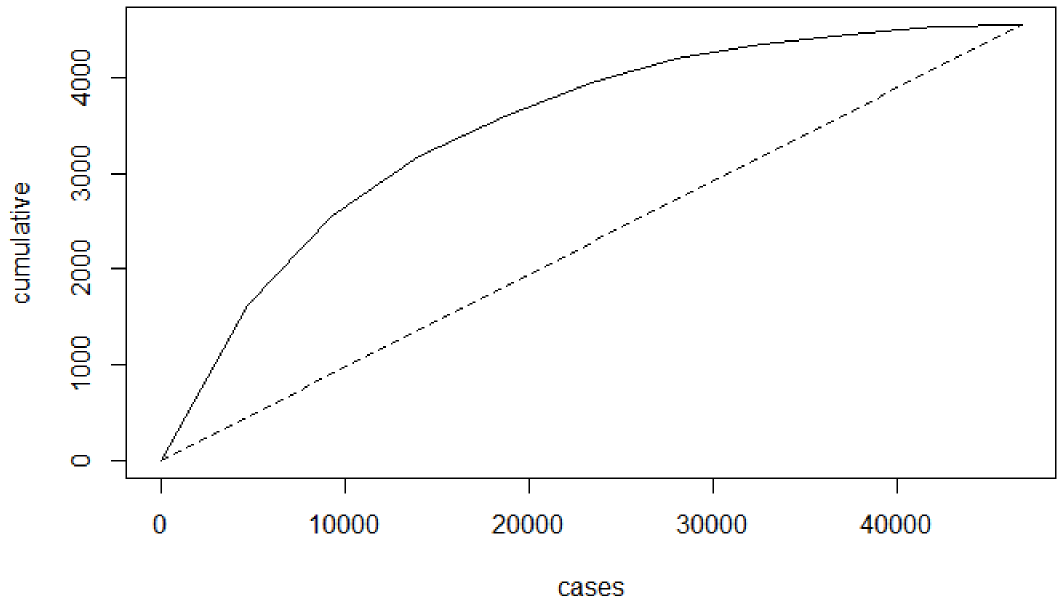
               Accuracy : 0.8568
                 95% CI : (0.8536, 0.86)
    No Information Rate : 0.9046
    P-Value [Acc > NIR] : 1

                  Kappa : 0.2757
 Mcnemar's Test P-Value : <0.0000000000000002

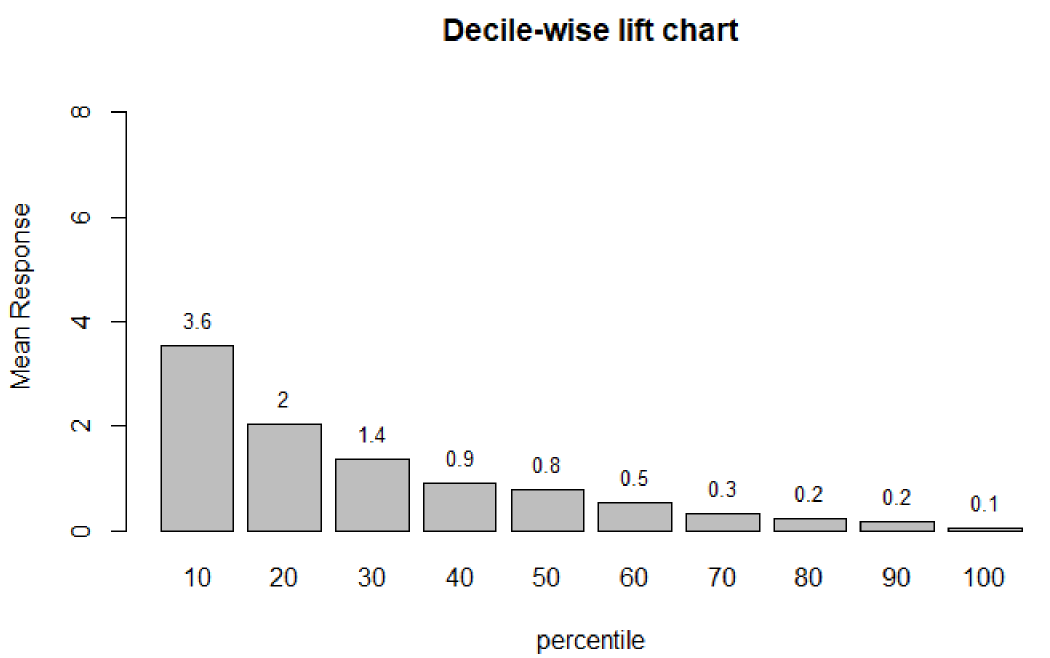
            Sensitivity : 0.9037
            Specificity : 0.4120
         Pos Pred Value : 0.9358
         Neg Pred Value : 0.3111
             Prevalence : 0.9046
         Detection Rate : 0.8175
   Detection Prevalence : 0.8736
      Balanced Accuracy : 0.6579

       'Positive' Class : 0
```

Notice the confusion matrix has an accuracy rate of 86% and a (positive 0 is modeled) so specificity is 90% while sensitivity is 41%. Next is the Lift chart.



This lift is reasonable since is bows out from the random model dotted line, and not the more the graph bows to the left the better the model.



Now the decile chart looks good coming in at 3.6, that is 3.6 in the first decile with a multiplier of 3.6 above the average mean response for the entire dataset, and analyst feel this is a very reasonable lift.

LDA

The linear discriminant analysis procedure developed by R.A. Fisher was also employed since it is well documented that if the data is multivariate normal LDA preforms better than Logistic regression. However, running lda in R did not yield a confusion matrix that did better than CART or Logistic Regression (notice the sensitivity is 0.06772, since positive is 0), and modelers did not pursue this avenue of analysis.

```
Confusion Matrix and Statistics


          0      1
  0 168283  16920
  1    469   1229

               Accuracy : 0.907
                 95% CI : (0.9056, 0.9083)
    No Information Rate : 0.9029
    P-Value [Acc > NIR] : 0.000000001189

                  Kappa : 0.109
 Mcnemar's Test P-Value : < 0.0000000000000022

            Sensitivity : 0.99722
            Specificity : 0.06772
         Pos Pred Value : 0.90864
         Neg Pred Value : 0.72379
             Prevalence : 0.90290
         Detection Rate : 0.90039
   Detection Prevalence : 0.99091
      Balanced Accuracy : 0.53247

       'Positive' Class : 0

[1] 0.9069614
```

Bibliography

This section remains to be completed.

Note the Czech bank data has been analyze in various scenarios, and here is one by Lija Mohan and Sudheep Elayidom M.

A Novel Big Data Approach to Classify Bank Customers – Solution by Combining PIG, R and Hadoop.

IJITCS-V8-N9-10_Mohan_Sudheep.pdf

Appendix (Rmarkdown code)

```
```{r, echo=FALSE}
library(dplyr)
account.df <- read.csv("account.csv")
card.df <- read.csv("card.csv")
client.df <- read.csv("client.csv")
disp.df <- read.csv("disp.csv")
district.df <- read.csv("district.csv")
str(district.df)
loans.df <- read.csv("loan.csv")
order.df <- read.csv("order.csv")
trans.df <- read.csv("trans.csv")

names(district.df) <- c("district_id","district_name",
                "region","num_of_inhabitants",
                "num_of_municipalities_with_inhabitants_LT_499",
                "num_of_municipalities_with_inhabitants_500_1999",
                "num_of_municipalities_with_inhabitants_2000_9999",
                "num_of_municipalities_with_inhabitants_GT_10000",
                "num_of_cities",
                "Ratio_of_urban_inhabitants",
                "Average_Salary",
                "unemployment_rate_in_1995",
                "unemployment_rate_in_1996",
```
```

```
                "num_of_enterpreneurs_per_1000_inhabitants",
                "num_of_crimes_commited_in_1995",
                "num_of_crimes_commited_in_1996")

head(district.df)

leftjoindat_trans_loans <- left_join(trans.df, loans.df, by = "account_id")

leftjoindat_trans_loans_account <- left_join(leftjoindat_trans_loans, account.df, by = "account_id")

leftjoindat_trans_loans_account_district <- left_join(leftjoindat_trans_loans_account,district.df, by =
"district_id")

leftjoindat_trans_loans_account_district_disp <-
left_join(leftjoindat_trans_loans_account_district,disp.df, by = "account_id")

leftjoindat_trans_loans_account_district_disp_card <-
left_join(leftjoindat_trans_loans_account_district_disp,card.df, by = "disp_id")

leftjoindat_trans_loans_account_district_disp_card_client <-
left_join(leftjoindat_trans_loans_account_district_disp_card,client.df, by = "client_id")

str(leftjoindat_trans_loans_account_district_disp_card_client)

names(leftjoindat_trans_loans_account_district_disp_card_client)

final.data.df = leftjoindat_trans_loans_account_district_disp_card_client
str(final.data.df)
```

```{r, echo=FALSE}

final.data.df$k_symbol[which(final.data.df$k_symbol=="")] = NA

final.data.df$bank[which(final.data.df$bank=="")] = NA

final.data.df$operation[which(final.data.df$operation=="")] = NA


NA_population <- as.data.frame(colSums(is.na(final.data.df)))

names(NA_population) <- "NA population"

#NA_population


Number_Populated <- as.data.frame(colSums(!is.na(final.data.df)))

names(Number_Populated) <- "Number populated"

Number_Populated


#table(final.data.df$operation)

#table(as.data.frame(colSums(is.na(final.data.df))))

#names(which(colSums(is.na(final.data.df))!=0))




```

Now Exploratory Data Analysis

The following is a list of fields in the final.data.df

#trans_id

#date.x

operation

balance

bank

#loan_id

amount.y

payments

district_id.x

date

region

num_of_municipalities_with_inhabitants_LT_499

num_of_municipalities_with_inhabitants_2000_9999

num_of_cities

Average_Salary

unemployment_rate_in_1996

num_of_crimes_commited_in_1995

#disp_id

type.y

type

birth_number

#account_id

type.x

amount.x

k_symbol

account (transaction data account of partner)

date.y

duration

status

frequency

district_name

num_of_inhabitants

num_of_municipalities_with_inhabitants_500_1999

num_of_municipalities_with_inhabitants_GT_10000

Ratio_of_urban_inhabitants

unemployment_rate_in_1995

num_of_enterpreneurs_per_1000_inhabitants

num_of_crimes_commited_in_1996

#client_id

#card_id

issued

district_id.y

```{r, echo=FALSE}

table(final.data.df$operation)

```

Mode of Transaction

'VYBER KARTOU' stands for Credit Card Withdrawal

'VKLAD' stands for Credit in Cash

'PREVOD Z UCTU' stands for Collection from Another Bank

'VYBER' stands for Withdrawal in Cash

'PREVOD NA UCET' stands for Remittance to Another Bank

```{r, echo=FALSE}

hist(final.data.df$balance,main ="Histogram of account balance",xlab= "Account balance",col = "orange")
```

boxplot(final.data.df$balance,ylab="Account balance")

summary(final.data.df$balance)

```

There are negative balances and balances are skewed to the right and the boxplot gives lots off outliers with larger accounts managment might want to explore these larger accounts

```{r, echo=FALSE}

table(final.data.df$bank)

```

We observe that bank labels have lots of missing values

These are the amount of the loan data, transactions are x and loans are y

```{r, echo=FALSE}

options(scipen=999)

hist(final.data.df$amount.x,main = "Histogram of transaction amount", xlab = "transaction amount", ylim = c(0,1000000),col = "orange")

boxplot(final.data.df$amount.x,ylab="transaction amount")

summary(final.data.df$amount.x)

hist(final.data.df$amount.y,main = "Histogram of total loan amount", xlab = "total loan amount",ylim = c(0,80000),col = "violet")

boxplot(final.data.df$amount.y, ylab = "total loan amount")

summary(final.data.df$amount.y)

```

Boxplots are amount.x and amount.y

These are monthly payments in the loan.df file

```{r, echo=FALSE}

hist(final.data.df$payments,main = "Histogram of monthly loan amount payments",xlab = "Monthly Payment Amount",ylim = c(0,35000),col = "orange") summary(final.data.df$payments)

```

date is from account.df and the the date the account was created

date.x is from the transaction.df data and is the date of the transaction

date.y is from the loan.df data and is the date the loan was granted

We will use hist to see if there is a peak in the date distribution

```{r, echo=FALSE}

hist(as.numeric(paste("19",substr(final.data.df$date,1,2),sep = "")),main="Histogram of yearly accounts opened",xlab="Year",ylim=c(0,700000),col = "orange")

hist(as.numeric(paste("19",substr(final.data.df$date.x,1,2),sep = "")),main="Histogram of Yearly transactions",xlab="Year")

hist(as.numeric(paste("19",substr(final.data.df$date.y,1,2),sep = "")),main="Histogram of Yearly loans granted",xlab="Year",ylim=c(0,70000),col = "orange")

# hist(final.data.df$date)

# hist(final.data.df$date.x)

# hist(final.data.df$date.y)

# note the dates are yymmdd  for the date.y the first class in the histogram is yymmdd so the year is 93 the mm dd do not count this is a general class so notice 94 had a lot of transactions.

```

```{r, echo=FALSE}

table(final.data.df$region)

```
```

```{r, echo=FALSE}

table(final.data.df$num_of_municipalities_with_inhabitants_LT_499)

```

The way to read the above table is that, for example, there are 290910 records with 0 municipalities LT 499

```{r, echo=FALSE}

table(final.data.df$num_of_municipalities_with_inhabitants_2000_9999 )

```

We observe that there are 248771 records with no muni's with population 2000 - 9999

```{r, echo=FALSE}

table(final.data.df$num_of_cities)

```

```{r, echo=FALSE}

hist(final.data.df$Average_Salary,main = "Histogram of average salary",xlab = "Salary",xlim = c(8000,13000),col="yellow",ylim = c(0,200000))

boxplot(final.data.df$Average_Salary, ylab="Average Salary")

summary(final.data.df$Average_Salary)

```

```{r, echo=FALSE}

hist(final.data.df$unemployment_rate_in_1995,main = "unemployment in 1995",xlab = "district no",xlim = c(0,10),col="blue",ylim = c(0,200000))

summary(final.data.df$unemployment_rate_in_1995)

hist(final.data.df$unemployment_rate_in_1996,main = "unemployment in 1996",xlab = "district no",xlim = c(0,10),col="blue",ylim = c(0,200000))

summary(final.data.df$unemployment_rate_in_1996)
```

```

As the unemployment rate increases there is a statistical relationship with a higher number of cities

```{r, echo=FALSE}

hist(as.numeric(final.data.df$num_of_crimes_commited_in_1995),main = "Histogram of num of crimes commited in 1995",xlab = "No. of crimes in 1995",col = "red",xlim = c(0,100000))

hist(as.numeric(final.data.df$num_of_crimes_commited_in_1996),main = "Histogram of num of crimes commited in 1996",xlab = "No. of crimes in 1996",col = "red",xlim = c(0,100000))

table(final.data.df$num_of_crimes_commited_in_1995,final.data.df$num_of_cities)

```

type.x (tranaction data base)

type.y (loan data)

type (dispostion)

```{r, echo=FALSE}

table(final.data.df$type.x)

table(final.data.df$type.y)

table(final.data.df$type)

```

"PRIJEM" stands for credit

VYBER' stands for Withdrawal in Cash

"VYDAJ" stands for withdrawal

birth number is client data

birth number

identification of client
```

the number is in the form YYMMDD for men,

the number is in the form YYMM+50DD for women,

where YYMMDD is the date of birth

```{r, echo=FALSE}
hist(as.numeric(paste("19",substr(final.data.df$birth_number,1,2),sep = "")),xlim=c(1900,2000),main = "birth",xlab= "Year",col="green",ylim = c(0,140000))
```

```{r, echo=FALSE}
k_symbol <- as.data.frame(table(final.data.df$k_symbol))
print(k_symbol)
```

k_symbol is from the tranaction data

"POJISTNE" stands for insurrance payment

"SLUZBY" stands for payment for statement

"UROK" stands for interest credited

"SANKC. UROK" sanction interest if negative balance

"SIPO" stands for household

"DUCHOD" stands for old-age pension

"UVER" stands for loan payment

Next: transaction data account of partner

no information here

```{r, echo=FALSE}
```

summary(final.data.df$account)

```

duration is the life of the loan

this is the loan term

```{r, echo=FALSE}
table(final.data.df$duration)
```

status is A B C D

```{r, echo=FALSE}
table(final.data.df$status)

```

'A' stands for contract finished, no problems,

'B' stands for contract finished, loan not payed,

'C' stands for running contract, OK so far,

'D' stands for running contract, client in debt

frequency is in the account database

```{r, echo=FALSE}
table(final.data.df$frequency)
```
```

frequency of issuance of statements

"POPLATEK MESICNE" stands for monthly issuance

"POPLATEK TYDNE" stands for weekly issuance

"POPLATEK PO OBRATU" stands for issuance after transaction

~~~
```{r, echo=FALSE}

district.name.df <- as.data.frame(table(final.data.df$district_name))

#district.name.df

```
~~~

~~~
```{r, echo=FALSE}

hist(final.data.df$num_of_inhabitants,main="Histogram of No. of inhabitants",xlab = "No. of inhabitants",col = "orange")

```
~~~

~~~
```{r, echo=FALSE}

hist(final.data.df$num_of_municipalities_with_inhabitants_500_1999,main = "number of municipalitites with inhabitants 500-1999",xlab = "No. of municipalitites",col = "orange")



```
~~~

num_of_municipalities_with_inhabitants_500_1999)

```{r, echo=FALSE}
table(final.data.df$num_of_municipalities_with_inhabitants_GT_10000 )
#table(unique(final.data.df$num_of_municipalities_with_inhabitants_GT_10000))
```

num_of_municipalities_with_inhabitants_GT_10000

```{r, echo=FALSE}
hist(final.data.df$Ratio_of_urban_inhabitants,main = "Histogram of the ratio of urban to rural inhabitants",xlab = "% of urban inhabitants",col = "yellow")
```

```{r, echo=FALSE}
hist(final.data.df$num_of_enterpreneurs_per_1000_inhabitants,main = "Histogram of num of entrepreneurs per 1000 inhabitants",col = "blue",xlim = c(80,180))
```

```{r, echo=FALSE}
hist(final.data.df$num_of_crimes_commited_in_1996,main = "Histogram of number of crimes commited in 1996",xlab = "No. of crimes",col = "red",ylim = c(0,800000))
#table(final.data.df$num_of_crimes_commited_in_1996)
```

checking loan data

```
```{r, echo=FALSE}
loan.data.df <- subset(final.data.df,subset=(!is.na(loan_id)))
table(loan.data.df$status)
```
```

issued   issue date   in the form YYMMDD (credit card data)

```
```{r, echo=FALSE}
#table(final.data.df$issued)
```
```

```
```{r, echo=FALSE}
#table(final.data.df$district_id.x)
#table(final.data.df$district_id.y)
```
```

look at loan versus non loan obligor

```
```{r, echo=FALSE}
loan.nonloan.data.df <- as.data.frame(ifelse(!is.na(final.data.df$loan_id), 1, 0))
names(loan.nonloan.data.df) <- "loan_dummy"
```
```

#str(loan.nonloan.data.df)

table(loan.nonloan.data.df$loan_dummy)

final.data.loan.dummy.df <- cbind(final.data.df,loan.nonloan.data.df)

#str(final.data.loan.dummy.df)

#lm.reg <- lm(loan_dummy ~ Average_Salary+region + num_of_crimes_commited_in_1996, final.data.loan.dummy.df)

#summary(lm.reg)

```

good loan verus bad loan

```{r, echo=FALSE}

loan.data.df <- subset(final.data.df,subset=(!is.na(loan_id)))

table(loan.data.df$status)

bad.good.loan.data.df <- as.data.frame(ifelse(loan.data.df$status == "B"|loan.data.df$status == "D", 1, 0))

names(bad.good.loan.data.df) <- "bad_good_dummy"

table(bad.good.loan.data.df)

final.bad.good.data.df <- cbind(loan.data.df,bad.good.loan.data.df)

#str(final.bad.good.data.df)

one.data.df <- subset(final.bad.good.data.df,bad_good_dummy == 1)

zero.data.df <- subset(final.bad.good.data.df,bad_good_dummy == 0 )

set.seed(5)

```
sample_size <- c(22760/210867)

#print(sample_size)

zero.rows<-sample(rownames(zero.data.df), 0.1079353*dim(zero.data.df)[1])

zero.rand.df <- zero.data.df[zero.rows,]

final.zero.one.df <- rbind(one.data.df,zero.rand.df)

table(final.zero.one.df$bad_good_dummy)

table(final.bad.good.data.df$bad_good_dummy)

now put bad_good_dummy back onto the loan.data.df data

```
```
```{r}
# do pca here and profile good versus bad obligor


str(final.bad.good.data.df)

data.for.plot1 <- aggregate(final.bad.good.data.df$Average_Salary, by =
list(final.bad.good.data.df$bad_good_dummy) ,FUN=mean)

names(data.for.plot1) <- c("bad_good", "meanSalary")

barplot(data.for.plot1$meanSalary, names.arg = data.for.plot1$bad_good, xlab ="good bad dummy",
ylab = "ave salary" )


data.for.plot2 <- aggregate(final.bad.good.data.df$amount.y , by =
list(final.bad.good.data.df$bad_good_dummy) ,FUN=mean)

names(data.for.plot2) <- c("bad_good", "meanLoanAmount")

barplot(data.for.plot2$meanLoanAmount, names.arg = data.for.plot2$bad_good, xlab ="good bad
dummy", ylab = "ave loan amount" )
```
```

```
data.for.plot3 <- aggregate(final.bad.good.data.df$num_of_inhabitants , by = list(final.bad.good.data.df$bad_good_dummy) ,FUN=mean)

names(data.for.plot3) <- c("bad_good", "meanPop")

barplot(data.for.plot3$meanPop, names.arg = data.for.plot3$bad_good, xlab ="good bad dummy", ylab = "ave population" )


data.for.plot4 <- aggregate(final.bad.good.data.df$num_of_municipalities_with_inhabitants_LT_499 , by = list(final.bad.good.data.df$bad_good_dummy) ,FUN=mean)

names(data.for.plot4) <- c("bad_good", "meanPopLT499")

barplot(data.for.plot4$meanPopLT499, names.arg = data.for.plot4$bad_good, xlab ="good bad dummy", ylab = "sum number of muni LT 499" )


data.for.plot5 <- aggregate(final.bad.good.data.df$num_of_municipalities_with_inhabitants_500_1999 , by = list(final.bad.good.data.df$bad_good_dummy) ,FUN=mean)

names(data.for.plot5) <- c("bad_good", "meanPop500_1999")

barplot(data.for.plot5$meanPop500_1999, names.arg = data.for.plot5$bad_good, xlab ="good bad dummy", ylab = "ave population district btw 500 and 1999" )


data.for.plot6 <- aggregate(final.bad.good.data.df$num_of_municipalities_with_inhabitants_2000_9999 , by = list(final.bad.good.data.df$bad_good_dummy) ,FUN=mean)

names(data.for.plot6) <- c("bad_good", "meanPop2000_9999")

barplot(data.for.plot6$meanPop2000_9999, names.arg = data.for.plot6$bad_good, xlab ="good bad dummy", ylab = "ave population district btw 2000 and 9999" )


data.for.plot7 <- aggregate(final.bad.good.data.df$num_of_municipalities_with_inhabitants_GT_10000 , by = list(final.bad.good.data.df$bad_good_dummy) ,FUN=mean)

names(data.for.plot7) <- c("bad_good", "meanPopGT_10000")

barplot(data.for.plot7$meanPopGT_10000, names.arg = data.for.plot7$bad_good, xlab ="good bad dummy", ylab = "ave population district btw GT 10000" )
```

data.for.plot8 <- aggregate(final.bad.good.data.df$num_of_cities, by = list(final.bad.good.data.df$bad_good_dummy) ,FUN=mean)

names(data.for.plot8) <- c("bad_good", "meanNumCities")

barplot(data.for.plot8$meanNumCities, names.arg = data.for.plot8$bad_good, xlab ="good bad dummy", ylab = "ave number of cities in the districts")

data.for.plot9 <- aggregate(final.bad.good.data.df$Ratio_of_urban_inhabitants , by = list(final.bad.good.data.df$bad_good_dummy) ,FUN=mean)

names(data.for.plot9) <- c("bad_good", "meanRatioUbanPop")

barplot(data.for.plot9$meanRatioUbanPop, names.arg = data.for.plot9$bad_good, xlab ="good bad dummy", ylab = "mean ratio of urban pop in a district")

data.for.plot10 <- aggregate(final.bad.good.data.df$unemployment_rate_in_1996 , by = list(final.bad.good.data.df$bad_good_dummy) ,FUN=mean)

names(data.for.plot10) <- c("bad_good", "meanunemp1996")

barplot(data.for.plot10$meanunemp1996, names.arg = data.for.plot10$bad_good, xlab ="good bad dummy", ylab = "mean unemployment 1996")

data.for.plot11 <- aggregate(final.bad.good.data.df$num_of_crimes_commited_in_1996, by = list(final.bad.good.data.df$bad_good_dummy) ,FUN=mean)

names(data.for.plot11) <- c("bad_good", "meancrimes1996")

barplot(data.for.plot11$meancrimes1996, names.arg = data.for.plot11$bad_good, xlab ="good bad dummy", ylab = "mean crimes 1996")

```
```

creating one zero data so analysts can get

create categorical varialbes of continous variables using percentiles in the summary function

```{r, echo=FALSE}

summary(final.data.df$amount.x)

final.data.df$amount_trans <- as.numeric(ifelse(final.data.df$amount.x < 6694, 0, 1))

table(final.data.df$amount_trans)

summary(final.data.df$balance)

final.data.df$balance_trans <- as.numeric(ifelse(final.data.df$balance < 49436, 0, 1))

table(final.data.df$balance_trans)

summary(final.data.df$Average_Salary)

final.data.df$Average_Salary_demographic <- as.numeric(ifelse(final.data.df$Average_Salary < 9920, 0, 1))

table(final.data.df$Average_Salary_demographic)

# Get levels and add "None"

levels <- levels(final.data.df$operation)

levels[length(levels) + 1] <- "None"
```

```
# refactor operation to include "None" as a factor level
# and replace NA with "None"
final.data.df$operation <- factor(final.data.df$operation, levels = levels)
final.data.df$operation[is.na(final.data.df$operation)] <- "None"
table(final.data.df$operation)

# Get levels and add "None"
levels <- levels(final.data.df$k_symbol)
levels[length(levels) + 1] <- "None"


final.data.df$k_symbol <- factor(final.data.df$k_symbol, levels = levels)
final.data.df$k_symbol[is.na(final.data.df$k_symbol)] <- "None"
table(final.data.df$k_symbol)

# Get levels and add "None"
levels <- levels(final.data.df$bank)
levels[length(levels) + 1] <- "None"


final.data.df$bank <- factor(final.data.df$bank, levels = levels)
final.data.df$bank[is.na(final.data.df$bank)] <- "None"
table(final.data.df$bank)

# Get levels and add "None"
levels <- levels(final.data.df$region)
levels[length(levels) + 1] <- "None"
```

```
final.data.df$region <- factor(final.data.df$region, levels = levels)
final.data.df$bank[is.na(final.data.df$region)] <- "None"
table(final.data.df$region)

# Get levels and add "None"
levels <- levels(final.data.df$type)
levels[length(levels) + 1] <- "None"


final.data.df$type <- factor(final.data.df$type, levels = levels)
final.data.df$type[is.na(final.data.df$type)] <- "None"
table(final.data.df$type)

type.x.dummy <- as.data.frame(model.matrix(~ 0 + type.x, data=final.data.df))
#str(type.x.dummy)
operation.dummy <- as.data.frame(model.matrix(~ 0 + operation, data=final.data.df))
#str(operation.dummy)
k_symbol.dummy <- as.data.frame(model.matrix(~ 0 +  k_symbol, data=final.data.df))
#str(k_symbol.dummy)
bank.dummy <- as.data.frame(model.matrix(~ 0 + bank , data=final.data.df))
#str(bank.dummy)
type.dummy <- as.data.frame(model.matrix(~ 0 + type , data=final.data.df))
#str(type.dummy)


final.data.df$region <- as.factor(gsub(" ","_",final.data.df$region))
```

````
table(final.data.df$region)

region.dummy <- as.data.frame(model.matrix(~ 0 + region , data=final.data.df))

#str(region.dummy)

final.data.rules.df <-
cbind(final.data.df,type.x.dummy,operation.dummy,k_symbol.dummy,bank.dummy,region.dummy,type
.dummy)

#str(final.data.rules.df)

which(colSums(is.na(final.data.rules.df))!=0)

```
````

Now look at association rules and collaborative filtering

````
```{r, echo=FALSE}

library(arules)

#str(final.data.rules.df)

set.seed(72)

train.rows <- sample(rownames(final.data.rules.df), 0.80*dim(final.data.rules.df)[1])

train.df<- final.data.rules.df[train.rows,]

valid.rows <- setdiff(rownames(final.data.rules.df), train.rows)

valid.df <- final.data.rules.df[valid.rows,]

#table(train.df$amount_trans)

#table(train.df$balance_trans)#44

#table(train.df$Average_Salary_demographic)#45

#table(train.df$type.xPRIJEM) #46
````

```
#table(train.df$type.xVYBER) #47
#table(train.df$type.xVYDAJ) #48
#str(train.df)
#rules.data.df <- train.df[,c(43:92)]
rules.data.df <- valid.df[,c(43:92)]
rules.matrix <- as.matrix(rules.data.df)
rules.trans <- as(rules.matrix,"transactions")
rules <- apriori(rules.trans, parameter=list(supp=0.2,conf=0.5, target = "rules"))
inspect(head(sort(rules,by = "lift"), n=50))


```

k_symbol

"POJISTNE" stands for insurrance payment

"SIPO" stands for household

"LEASING" stands for leasing

"UVER" stands for loan payment

transaction type.x

"PRIJEM" stands for credit

"VYDAJ" stands for withdrawal

transaction operation

"VYBER KARTOU" credit card withdrawal

"VKLAD" credit in cash

"PREVOD Z UCTU" collection from another bank

"VYBER" withdrawal in cash

"PREVOD NA UCET" remittance to another bank

bank of the partner

CLUSTERING

```
```{r, echo=FALSE}

data.loan.df <- as.data.frame(ifelse(final.data.df$status == "B"|final.data.df$status == "D", 1, 0))

#str(data.loan.df)

names(data.loan.df) <- "bad_good_dummy"

#str(data.loan.df)

table(data.loan.df)

final.data.loan.df <- cbind(final.data.df,data.loan.df)

str(final.data.loan.df)

cluster.df <- as.data.frame(final.data.loan.df[,c(21:28,30,33,45,46)])

# 21:28 are the demographic variables 30 is average salary , 33 is number of entreprenures, 45 is transaction balance and 46 is bad good dummy

#str(cluster.df)

table(cluster.df$bad_good_dummy)

demographics.df <- na.omit(cluster.df)

#str(demographics.df)

demographics.df1 <- demographics.df[,c(1:3,12)]
```
```

```
for.cluster1 <-as.data.frame(aggregate(demographics.df1$num_of_inhabitants, by =
list(demographics.df1$region),FUN=mean))

for.cluster2 <-
as.data.frame(aggregate(demographics.df1$num_of_municipalities_with_inhabitants_LT_499 , by =
list(demographics.df1$region),FUN=mean))

for.cluster3 <-as.data.frame(aggregate(demographics.df1$bad_good_dummy , by =
list(demographics.df1$region),FUN=mean))


colnames(for.cluster1)<- c("region","mean_num_of_inhabitants")

colnames(for.cluster2)<- c("region","mean_num_of_municipalities_with_inhabitants_LT_499")

colnames(for.cluster3)<- c("region","bad_good_dummy")

str(for.cluster1)

str(for.cluster2)

str(for.cluster3)


final.cluster.data1 <- merge(for.cluster1,for.cluster2,by.for.cluster1 ="region", by.for.cluster2 ="region")

final.cluster.data2 <- merge(final.cluster.data1, for.cluster3,by.final.cluster.data1 = "region",
by.for.cluster3 ="region")


row.names(final.cluster.data2) <- final.cluster.data2[,1]

final.cluster.data <- final.cluster.data2[,-1]


final.cluster.data.norm <- sapply(final.cluster.data,scale)

row.names(final.cluster.data.norm) <- row.names(final.cluster.data)


d.norm <- dist(final.cluster.data.norm , method = "euclidean")

hc1 <-hclust(d.norm, method="single")

plot(hc1)
```

#now just good bad

final.cluster.good.bad <- as.data.frame(final.cluster.data2[,-c(1:3)])

colnames(final.cluster.good.bad) <- "good_bad"

row.names(final.cluster.good.bad) <- row.names(final.cluster.data2)

final.cluster.good.bad.norm <- sapply(final.cluster.good.bad ,scale)

row.names(final.cluster.good.bad.norm) <- row.names(final.cluster.data)

d.norm2 <- dist(final.cluster.good.bad.norm , method = "euclidean")

hc2 <-hclust(d.norm2, method="single")

plot(hc2)

```

Now look at PCA

final.cluster.data

```{r}

pca <- prcomp(na.omit(final.cluster.data), scale=T)

summary(pca)

str(cluster.df)

pca.data.df <- cluster.df[,-c(1,12)]

str(pca.data.df)

names(pca.data.df)

pca1 <- prcomp(na.omit(pca.data.df), scale=T)

summary(pca1)

pca1$rotation[,1:4]

```
```

now look at CART

looking at all the 233000 records not oversampled.

```{r, echo=FALSE}
# turn this on and off for all or oversampled data
final.zero.one.df <- final.bad.good.data.df #not oversampled here
#str(final.zero.one.df)
set.seed(271)
train.cart.rows <-sample(rownames(final.zero.one.df), 0.8*dim(final.zero.one.df)[1])
train.cart.df <- final.zero.one.df[train.cart.rows,]
valid.cart.rows <- setdiff(rownames(final.zero.one.df),train.cart.rows)
valid.cart.df <- final.zero.one.df[valid.cart.rows,]
str(train.cart.df)

table(train.cart.df$operation)


#str(train.cart.df)


target.data.df <- train.cart.df[,c(4,5,6,7,8:9,22:29,31,32,34,43)]
#target.data.df <- train.cart.df[,c(4,43)]
str(target.data.df)
library(rpart)
```

```
library(rpart.plot)

library(caret)



class.tree <- rpart(bad_good_dummy ~ ., data = target.data.df, method = "class")

options(scipen = 999)

prp(class.tree, type=1, extra=1, split.font = 1,varlen = -5)

rpart.plot(class.tree, type=4, digits=-3)


predict.train <- as.data.frame(as.numeric(predict(class.tree, target.data.df, type = "class")))

#str(predict.train)

colnames(predict.train) <- "predicted"

#str(predict.train)

table(predict.train)

predict.train$predicted <- ifelse(predict.train$predicted == 2,1,0)

table(predict.train$predicted)

table(predict.train$predicted,target.data.df$bad_good_dummy )

confusionMatrix(data= table(predict.train$predicted, target.data.df$bad_good_dummy))



#deeper.tree <-  rpart(bad_good_dummy ~ ., data = target.data.df, method = "class", cp=0.1, minsplit=1)


#length(deeper.tree$frame$var[deeper.tree$frame$var == "<leaf>"])


#prp(deeper.tree, type = 1, extra = 1, under = TRUE, split.font =1, varlen = -10,
box.col=ifelse(deeper.tree$frame$var == "<leaf>", 'gray', 'white'))
```

```

looking at CART with the rules data oversampling (option for no oversampling)

```{r, echo=FALSE}

loan.data.rules.df <- subset(final.data.rules.df,subset=(!is.na(loan_id)))
#str(loan.data.rules.df)
table(loan.data.rules.df$status)

bad.good.loan.data.rules.df <- as.data.frame(ifelse(loan.data.rules.df$status == "B"|loan.data.rules.df$status == "D", 1, 0))
names(bad.good.loan.data.rules.df) <- "bad_good_dummy"
table(bad.good.loan.data.rules.df)

final.bad.good.data.rules.df <- cbind(loan.data.rules.df,bad.good.loan.data.rules.df)
#str(final.bad.good.data.rules.df)

one.data.rules.df <- subset(final.bad.good.data.rules.df,bad_good_dummy == 1)
zero.data.rules.df <- subset(final.bad.good.data.rules.df,bad_good_dummy == 0 )
#str(one.data.rules.df)
#str(zero.data.rules.df)

set.seed(5)
sample_size <- c(22760/210867)
```
```

```
#print(sample_size)

zero.rows.rules<-sample(rownames(zero.data.rules.df), 0.1*dim(zero.data.rules.df)[1])

zero.rand.rules.df <- zero.data.rules.df[zero.rows.rules,]

final.zero.one.rules.df <- rbind(one.data.rules.df,zero.rand.rules.df)


table(final.zero.one.rules.df$bad_good_dummy)

str(final.zero.one.rules.df)




#########################################

# no oversampling here

#set.seed(275)

#train.cart.rows.rules <-sample(rownames(final.bad.good.data.rules.df),
0.8*dim(final.bad.good.data.rules.df)[1])

#train.cart.rules.df <- final.bad.good.data.rules.df[train.cart.rows.rules,]

#valid.cart.rows.rules <- setdiff(rownames(final.bad.good.data.rules.df),train.cart.rows.rules)

#valid.cart.rules.df <- final.bad.good.data.rules.df[valid.cart.rows.rules,]

#str(train.cart.rules.df)

#################################################################

# oversampling

set.seed(271)

train.cart.rows.rules <-sample(rownames(final.zero.one.rules.df), 0.8*dim(final.zero.one.rules.df)[1])

train.cart.rules.df <- final.zero.one.rules.df[train.cart.rows.rules,]

valid.cart.rows.rules <- setdiff(rownames(final.zero.one.rules.df),train.cart.rows.rules)

valid.cart.rules.df <- final.zero.one.rules.df[valid.cart.rows.rules,]
```

```
#str(train.cart.rules.df)

table(train.cart.rules.df$bad_good_dummy)

###### this tree works
target.data.df <- train.cart.rules.df[,c(4,5,6,7,8,9,13,14,15,18,21,43:92,93)]
# oversample here
#target.data.df <- train.cart.df[,c(4,5,6,7,8:9,22:29,31,32,34,43)]
#target.data.df <- train.cart.rules.df[,c(86:92,93)] # this works accuracy 40 percent
#target.data.df <- train.cart.rules.df[,c(92,93)]
t#arget.data.df <- train.cart.rules.df[,c(4,5,6,7,8,9,13,14,15,18,20,21,93)]
#target.data.df <- train.cart.rules.df[,c(4,93)] 47 percent
#target.data.df <- train.cart.rules.df[,c(5,93)] #43 percent
#target.data.df <- train.cart.rules.df[,c(6,93)] #43 percent
#target.data.df <- train.cart.rules.df[,c(7,93)] #38 percent
#target.data.df <- train.cart.rules.df[,c(8,93)] #43 percent
#target.data.df <- train.cart.rules.df[,c(9,93)] #43 percent
#target.data.df <- train.cart.rules.df[,c(13,93)] #43 percent
#target.data.df <- train.cart.rules.df[,c(14,93)] #45 percent
#target.data.df <- train.cart.rules.df[,c(15,93)] #45 percent
#target.data.df <- train.cart.rules.df[,c(18,93)] #did not work
#target.data.df <- train.cart.rules.df[,c(20,93)] #29 percent
#target.data.df <- train.cart.rules.df[,c(21,93)] #42 percent
#target.data.df <- train.cart.rules.df[,c(22,93)] #35 percent
#target.data.df <- train.cart.rules.df[,c(23,24,93)] #38 percent
#target.data.df <- train.cart.rules.df[,c(23,24,25,93)] #38 percent
#target.data.df <- train.cart.rules.df[,c(23,24,25,26,93)] #35 percent
#target.data.df <- train.cart.rules.df[,c(28,29,93)] #31 percent
```

```
#target.data.df <- train.cart.rules.df[,c(23,24,25,26,93)] #35 percent
#target.data.df <- train.cart.rules.df[,c(30,93)] #32 percent
#target.data.df <- train.cart.rules.df[,c(31,32,34,93)] #30 percent
#target.data.df <- train.cart.rules.df[,c(37,93)] #30 percent GETS ZERO
#target.data.df <- train.cart.rules.df[,c(39,93)] #42 percent
#target.data.df <- train.cart.rules.df[,c(42,93)] #42 percent
#target.data.df <- train.cart.rules.df[,c(43,93)] #does not work
#target.data.df <- train.cart.rules.df[,c(44,93)] #44 percent specificy up
#target.data.df <- train.cart.rules.df[,c(45,93)] # does not work
#target.data.df <- train.cart.rules.df[,c(46,93)] # does not work
#target.data.df <- train.cart.rules.df[,c(50:52,93)] #43
#target.data.df <- train.cart.rules.df[,c(55,93)] #does not work
#target.data.df <- train.cart.rules.df[,c(62,93)] #does not work
#target.data.df <- train.cart.rules.df[,c(65,93)] #does not work
#target.data.df <- train.cart.rules.df[,c(81,93)] #does not work
#target.data.df <- train.cart.rules.df[,c(82,93)] #47 percent
#target.data.df <- train.cart.rules.df[,c(83,93)] #45 percent
#target.data.df <- train.cart.rules.df[,c(84,93)] #78 specificity
#target.data.df <- train.cart.rules.df[,c(85,93)] #does not work
#target.data.df <- train.cart.rules.df[,c(86,93)] #46 percent
#target.data.df <- train.cart.rules.df[,c(87,93)] #46 percent
#target.data.df <- train.cart.rules.df[,c(88,93)] #does not work
#target.data.df <- train.cart.rules.df[,c(89:91,93)] #does not work

###############################
# best model by hand
#target.data.df <- train.cart.rules.df[,c(37,93)] #sensitivity
#target.data.df <- train.cart.rules.df[,c(84,93)] #specificity
```

```
subset_1 <-subset(train.cart.df,bad_good_dummy ==1)

table(subset_1$region)


table(train.cart.df$bad_good_dummy,train.cart.df$region)

table(train.cart.df$bad_good_dummy,train.cart.df$type.y)


#target.data.df <- train.cart.rules.df[,c(37,84,87,88,93)]



#str(target.data.df)

library(rpart)

library(rpart.plot)

library(caret)


class.tree <- rpart(bad_good_dummy ~ ., data = target.data.df, method = "class", cp=0.01)

options(scipen = 999)

prp(class.tree, type=1, extra=1, split.font = 1,varlen = -10)

rpart.plot(class.tree, type=4, digits=-3)


predict.train <- as.data.frame(as.numeric(predict(class.tree, target.data.df, type = "class")))

colnames(predict.train) <- "predicted"

#str(predict.train)

table(predict.train)

predict.train$predicted <- ifelse(predict.train$predicted == 2,1,0)

table(predict.train)


table(predict.train$predicted,target.data.df$bad_good_dummy )
```

~~~
confusionMatrix(data= table(predict.train$predicted, target.data.df$bad_good_dummy))


#table(target.data.df$regioncentral_Bohemia,target.data.df$bad_good_dummy)
#table(target.data.df$regioneast_Bohemia,target.data.df$bad_good_dummy)
#table(target.data.df$regionnorth_Bohemia,target.data.df$bad_good_dummy)
#table(target.data.df$regionnorth_Moravia,target.data.df$bad_good_dummy)
#table(target.data.df$regionPrague,target.data.df$bad_good_dummy)
#table(target.data.df$regionwest_Bohemia,target.data.df$bad_good_dummy)
#table(target.data.df$regionsouth_Moravia,target.data.df$bad_good_dummy)
#table(target.data.df$regionsouth_Bohemia,target.data.df$bad_good_dummy)


#deeper.tree <-  rpart(bad_good_dummy ~ ., data = target.data.df, method = "class", cp=0.001, 
minsplit=1)

#length(deeper.tree$frame$var[deeper.tree$frame$var == "<leaf>"])

#prp(deeper.tree, type = 1, extra = 1, under = TRUE, split.font =1, varlen = -10, 
box.col=ifelse(deeper.tree$frame$var == "<leaf>", 'gray', 'white'))


#final.data.rules.df

```
~~~

Here is the validation of the tree determined above

no oversampling here

```{r, echo=FALSE}

set.seed(275)

train.cart.rows.rules <-sample(rownames(final.bad.good.data.rules.df), 0.8*dim(final.bad.good.data.rules.df)[1])

train.cart.rules.df <- final.bad.good.data.rules.df[train.cart.rows.rules,]

valid.cart.rows.rules <- setdiff(rownames(final.bad.good.data.rules.df),train.cart.rows.rules)

valid.cart.rules.df <- final.bad.good.data.rules.df[valid.cart.rows.rules,]

str(train.cart.rules.df)

##################################################################

###### this tree works

target.data.df <- valid.cart.rules.df[,c(4,5,6,7,8,9,13,14,15,18,21,43:92,93)]

library(rpart)

library(rpart.plot)

library(caret)

class.tree <- rpart(bad_good_dummy ~ ., data = target.data.df, method = "class", cp=0.005)

options(scipen = 999)

prp(class.tree, type=1, extra=1, split.font = 1,varlen = -10)

predict.train <- as.data.frame(as.numeric(predict(class.tree, target.data.df, type = "class")))
```

```
colnames(predict.train) <- "predicted"

#str(predict.train)

table(predict.train)

predict.train$predicted <- ifelse(predict.train$predicted == 2,1,0)

table(predict.train)


table(predict.train$predicted,target.data.df$bad_good_dummy )

confusionMatrix(data= table(predict.train$predicted, target.data.df$bad_good_dummy))
```

```{r, echo=FALSE}

#install.packages("adabag")


#library(adabag)

#library(rpart)

#library(caret)



train.cart.rules.df$bad_good_dummy <- as.factor(train.cart.rules.df$bad_good_dummy)

target.data.df <- train.cart.rules.df[,c(52,84,93)]

#boost <-boosting(bad_good_dummy ~ ., data = target.data.df)
```

```
#pred <- predict(boost,target.data.df)

#nstall.packages("randomForest")
library(randomForest)
rf <- randomForest(bad_good_dummy ~ ., data = target.data.df, ntree =100,mtry=1, nodesize=5, importance=TRUE)
varImpPlot(rf,type=1)

predict.rf <- as.data.frame(predict(rf,target.data.df))
colnames(predict.rf) <- c("predicted")
table(predict.rf)
#str(predict.rf)

predict.rf$predicted <- ifelse(predict.rf$predicted == 1,0,1)
table(predict.rf)
table(predict.rf$predicted,target.data.df$bad_good_dummy )
#confusionMatrix(data= table(predict.rf$predicted,target.data.df$bad_good_dummy ))
```

logistic

look at the full sample

```{r, echo=FALSE}
```

```
target.data.df <-
final.bad.good.data.rules.df[,c(6,7,13,14,15,22:34,43:47,60:64,67:79,81:83,85:87,89:91,93)]

str(target.data.df)

logit.reg <- glm(bad_good_dummy ~ . , data=target.data.df, family = "binomial")

options(scipen=999)

summary(logit.reg)


logit.reg.pred <- predict(logit.reg, target.data.df, type = "response")


library(gains)

target.data.df$bad_good_dummy <- as.numeric(target.data.df$bad_good_dummy)

gain <- gains(target.data.df$bad_good_dummy,logit.reg.pred, groups=10)

plot(c(0,gain$cume.pct.of.total*sum(target.data.df$bad_good_dummy))~c(0,gain$cume.obs),xlab =
"cases", ylab="cumulative", main="", type="l")

lines(c(0,sum(target.data.df$bad_good_dummy))~c(0,dim(target.data.df)[1]), lty=2)


target.data.df$bad_good_dummy<- as.numeric(target.data.df$bad_good_dummy)

heights <- gain$mean.resp/mean(target.data.df$bad_good_dummy)

midpoints <- barplot(heights,names.arg=gain$depth, ylim=c(0,9), xlab="percentile", ylab="Mean
Response", main="Decile-wise lift chart")


text(midpoints, heights + 0.5, labels=round(heights,1), cex=0.8)
```

now look at logistic regression on full sample with training and validation data

```{r, echo=FALSE}
set.seed(127)
```

all.train.cart.rows.rules <-sample(rownames(final.bad.good.data.rules.df), 0.8*dim(final.bad.good.data.rules.df)[1])

all.train.cart.rules.df <- final.bad.good.data.rules.df[all.train.cart.rows.rules,]

all.valid.cart.rows.rules <- setdiff(rownames(final.bad.good.data.rules.df),all.train.cart.rows.rules)

all.valid.cart.rules.df <- final.bad.good.data.rules.df[all.valid.cart.rows.rules,]

#str(all.train.cart.rules.df)

table(all.train.cart.rules.df$bad_good_dummy)

```

Here is the logistic regresss with all possible subset selection.

```{r}

target.train.data.df <- all.train.cart.rules.df[,c(6,7,13,14,15,22:34,43:47,60:64,67:79,81:83,85:87,89:91,93)]

str(target.train.data.df)

library(leaps)

search <- regsubsets(bad_good_dummy ~ ., data = target.train.data.df, nbest = 1,  nvmax = dim(target.train.data.df)[2],

    method = "exhaustive", really.big = T)

sum = summary(search)

sum$which

sum$rsq

sum$adjr2

sum$cp

plot(sum$rsq, xlab = "number of variables", ylab="R squared", type="l")

```
plot(sum$rss, type = "l")

which.max(sum$adjr2)

plot(sum$adjr2, type = "l",xlab = "number of variables", ylab="adjusted R squared",col="blue")

points(49,sum$adjr2[49], col="red", cex=2, pch=20)

logit.reg.sam <- glm(bad_good_dummy ~ . , data=target.train.data.df, family = "binomial")

options(scipen=999)

summary(logit.reg.sam)

########################

target.valid.data.df <-
all.valid.cart.rules.df[,c(6,7,13,14,15,22:34,43:47,60:64,67:79,81:83,85:87,89:91,93)]

#c(6,7,13,15,23:33,43:47,60:64,67:73,75:79,81:83,86:87,89:91,93)

str(target.valid.data.df)

logit.reg.pred.valid <- predict(logit.reg.sam, target.valid.data.df, type = "response")

pred = factor(ifelse(logit.reg.pred.valid>0.2,1,0), levels=c(0,1))

library(caret)

confusionMatrix(as.factor(pred), as.factor(target.valid.data.df$bad_good_dummy))

library(gains)

target.valid.data.df$bad_good_dummy <- as.numeric(target.valid.data.df$bad_good_dummy)

gain <- gains(target.valid.data.df$bad_good_dummy,logit.reg.pred.valid, groups=10)

plot(c(0,gain$cume.pct.of.total*sum(target.valid.data.df$bad_good_dummy))~c(0,gain$cume.obs),xlab
= "cases", ylab="cumulative", main="", type="l")
```

lines(c(0,sum(target.valid.data.df$bad_good_dummy))~c(0,dim(target.valid.data.df)[1]), lty=2)

target.valid.data.df$bad_good_dummy<- as.numeric(target.valid.data.df$bad_good_dummy)

heights <- gain$mean.resp/mean(target.valid.data.df$bad_good_dummy)

midpoints <- barplot(heights,names.arg=gain$depth, ylim=c(0,9), xlab="percentile", ylab="Mean Response", main="Decile-wise lift chart")

text(midpoints, heights + 0.5, labels=round(heights,1), cex=0.8)

```

LDA

```{r, echo=FALSE}

library(MASS)

library(caret)

library(ggplot2)

library(standardize)

train.cart.rules.n <- as.data.frame(scale(train.cart.rules.df[,c(6,7,13,14,15,43:47,60:64,67:79,81:83,85:87,89:91)]))

dep_var <- as.data.frame(train.cart.rules.df[,c(93)])

names(dep_var) <- c("bad_good_dummy")

#str(dep_var)

#str(train.cart.rules.n)

train.cart.rules.norm <- cbind(train.cart.rules.n,dep_var)

#str(train.cart.rules.norm)

target.data.df <- train.cart.rules.norm

lda1 <- lda(bad_good_dummy ~ . , data=target.data.df)

pred1 <- predict(lda1,target.data.df)

#str(pred1)

table(pred1$class, target.data.df$bad_good_dummy)  # pred v actual

confusionMatrix(data=table(pred1$class, target.data.df$bad_good_dummy))

mean(pred1$class ==target.data.df$bad_good_dummy)  # percent accurate

#sum(pred1$posterior[, 1] >=.5)

#sum(pred1$posterior[, 1] >=.75)  # increase the cut-off from .5 to .75

```

now look at the validation set for LDA

```{r, echo=FALSE}

#with MASS

library(MASS)

library(caret)

library(ggplot2)

```
valid.cart.rules.n <-
as.data.frame(scale(valid.cart.rules.df[,c(6,7,13,14,15,43:47,60:64,67:79,81:83,85:87,89:91)]))

dep_var <- as.data.frame(valid.cart.rules.df[,c(93)])

names(dep_var) <- c("bad_good_dummy")

#str(dep_var)

#str(valid.cart.rules.n)

valid.cart.rules.norm <- cbind(valid.cart.rules.n,dep_var)

#str(valid.cart.rules.norm)

target.data.df <- valid.cart.rules.norm

##

now adjust for priors

prior <- c(0.9, 0.1)

##

lda1v <- lda(bad_good_dummy ~ . , data=target.data.df,prior = prior)

pred1v <- predict(lda1v,target.data.df)

#str(pred1v)

table(pred1v$class, target.data.df$bad_good_dummy) # pred v actual

mean(pred1v$class ==target.data.df$bad_good_dummy) # percent accurate

confusionMatrix(data=table(pred1v$class, target.data.df$bad_good_dummy) , prior = c(0.9,0.1))

#with Discrimant

library(DiscriMiner)
```

```
target.data.df <- valid.cart.rules.df[,c(6,7,13,14,15,43:47,60:64,67:79,81:83,85:87,89:91,93)]
#str(target.data.df)

lda2v <- linDA(target.data.df[,1:37],target.data.df[,38])
#str(lda2v)

table(lda2v$classification, target.data.df$bad_good_dummy)  # pred v actual

confusionMatrix(data=table(lda2v$classification, target.data.df$bad_good_dummy))
```

now look at full sample for training

```{r, echo=FALSE}
library(MASS)
library(caret)
library(ggplot2)
library(standardize)
train.cart.rules.n <-
as.data.frame(scale(all.train.cart.rules.df[,c(6,7,13,14,15,43:47,60:64,67:79,81:83,85:87,89:91)]))
dep_var <- as.data.frame(all.train.cart.rules.df[,c(93)])
names(dep_var) <- c("bad_good_dummy")
#str(dep_var)
#str(train.cart.rules.n)

train.cart.rules.norm <- cbind(train.cart.rules.n,dep_var)
#str(train.cart.rules.norm)
```

```
target.data.df <- train.cart.rules.norm

lda1 <- lda(bad_good_dummy ~ . , data=target.data.df)



pred1 <- predict(lda1,target.data.df)

#str(pred1)


table(pred1$class, target.data.df$bad_good_dummy)  # pred v actual

confusionMatrix(data=table(pred1$class, target.data.df$bad_good_dummy) )


mean(pred1$class ==target.data.df$bad_good_dummy)  # percent accurate
```

 now look at full sample for validation

```{r, echo=FALSE}
library(MASS)

library(caret)

library(ggplot2)

library(standardize)

valid.cart.rules.n <-
as.data.frame(scale(all.valid.cart.rules.df[,c(6,7,13,14,15,43:47,60:64,67:79,81:83,85:87,89:91)]))

dep_var <- as.data.frame(all.valid.cart.rules.df[,c(93)])

names(dep_var) <- c("bad_good_dummy")

#str(dep_var)

#str(train.cart.rules.n)
```

```
valid.cart.rules.norm <- cbind(valid.cart.rules.n,dep_var)

#str(valid.cart.rules.norm)



target.data.df <- valid.cart.rules.norm

lda1 <- lda(bad_good_dummy ~ . , data=target.data.df)



pred1 <- predict(lda1,target.data.df)

#str(pred1)


table(pred1$class, target.data.df$bad_good_dummy)  # pred v actual

confusionMatrix(data=table(pred1$class, target.data.df$bad_good_dummy) )


mean(pred1$class ==target.data.df$bad_good_dummy)  # percent accurate

```
```
```

Individual Writeup

It was a tremendous opportunity provided by Prof. Sourav Chatterjee to work on this project. Its been fun and challenging at the same time. I learnt not only about R but I think I improved my interpersonal as well as critical thinking skills while working with the team. The team was energetic and enthusiastic to take on the challenges.

Some of the challenges we faced in the group project were: As expected, most of the challenges

The project selection: We had initially finalized on working on bank dataset taken from Kaggle, but later found out that the same dataset was used by you in the class for practice. After multiple meetings and discussions, we decided to work on Czech Bank loan data by Prof. Petra Berka.

Data Gathering: We were provided with multiple files instead of single dataset. So, we had to combine all the files that were in single column format. First thing we did was to use excel to delimit the columns of each file individually then saved the files. Our task was to combine all the files into single one so we decided to take two approach and cross verify the records. In first approach, we created a database in SQL server with data from each file in separate tables and then did left joins on each table to create a single view. We exported the combined data into a csv file.
In the second approach, we loaded the each excel delimited csv files separately in R directly and then did SQL join using dplyr package. After doing all the left joins we compared the number of records from both the approaches (deciding which columns to use for join itself was a challenge).

But when we later realized that excel only supports 1,048,576 rows. Earlier, when we delimited the files in excel and saved it, our records got missing because the original data had more than 1,048,576 rows. The solution, what we decided was, directly importing the original files in R and cleaning the data in R itself.

Some other challenges, we faced were while coding, for which we got the solution from internet.

All the group member provided support and input based on their skills which made this project a success.