

# Cross Selling Credit Union

LISUM12: Data Glacier Final Project Presentation

# Team Details

- ❖ Group Name: Team Coltenback
- ❖ Name: Richard Coltenback
- ❖ Email: [rdcoltenback@gmail.com](mailto:rdcoltenback@gmail.com)
- ❖ Country: United States of America
- ❖ College: Drew University
- ❖ Specialization: Data Analyst
- ❖ GitHub Repository Link: <https://github.com/RColtenback/Cross-Selling-Data-Analysis.git>

# Problem Description

- ❖ XYZ credit union in Latin America is performing very well in selling banking products (e.g.: credit card, deposit account, retirement account, safe deposit box etc.). However, their existing customers are not buying more than one product which means the bank is not performing well in cross-selling (the bank is not able to sell their other offerings to existing customer). XYZ Credit Union decided to approach Team Coltenback to solve their problem.

# Business Understanding

- ❖ We're trying to understand the advantages and disadvantages of cross-selling banking products as opposed to up-selling. In particular, we want to know the financial benefits of cross-selling products to understand which products do well together and which should be sold alone. We also have personal information about each customer for the bank, thus we can analyze their backgrounds and see if there is any correlation with certain personal traits (e.g.: race, age, sex, etc.).

# Project Lifecycle and Deadline

- ❖ 1. Business understanding
  - ❖ 2. Data Understanding
  - ❖ 3. Data Cleansing and Transformation
  - ❖ 4. Exploratory data analysis
  - ❖ 5. EDA Recommendation (PowerPoint)
  - ❖ 6. Dashboard which should capture type of customer their count, segment wise (VIP, student, etc.) customer average age, and other KPIs which gives better business insight in taking decision.
  - ❖ 7. Prepare a final presentation
- 
- ❖ The final deadline is scheduled for October 30<sup>th</sup>, 2022.

# Data Understanding

- ❖ The dataset we will be using will be information given to us from the XYZ Credit Union. It has many different variables and lots of data that can help us solve our problem. The data has to deal with banking and different products that go along with it. Some examples of the variables are loans, pensions, mortgages, etc. Fortunately, due to the data collected being based on each customer, we also have access to personal information like their ages, sex, country of residence, etc. This dataset has a mix of different types of variables. Some variables like `fecha_dato` and `fecha_alta` are dates. Some variables like `indext` and `indfall` are Boolean or dichotomous as they are true/false or yes/no. Some variables like `ind_empleado` are string or nominal as they have letters for each customer. Finally, some variables like `ncodpers` and `age` are integer or scale as they have numbers for each customer.

# Data Understanding

- ❖ More about the dataset:

## Train.csv

<b>Total number of observations (rows)</b>	13647309
<b>Total number of files</b>	1
<b>Total number of features (columns)</b>	48
<b>Base format of the file</b>	.csv
<b>Size of the data</b>	2.29 GB

# Data Cleansing and Transformation

- ❖ My first goal for cleaning up this data was changing the names of the column variables. The original names were in Spanish and didn't make sense to me. Using the column description, I changed the names of each variable to make them more understandable for an English speaking audience (I could change them back to Spanish after since this is a Latin American credit union). Next, I checked the types of variables and how many NA values there are in the dataset to help show me which variables need a work around.

```
df.rename(columns={'fecha_dato':'Partitioned Date',
                  'ncodpers':'Customer Code',
                  'ind_empleado':'Employee Index',
                  'pais_residencia':'Country Residence',
                  'sexo':'Sex',
                  'age':'Age',
                  'fecha_alta':'Date Joined',
                  'ind_nuevo':'New Customer Index',
                  'antiguedad':'Seniority',
                  'indrel':'Primary Customer',
                  'ult_fec_cli_1t':'Last Date as Primary Customer',
                  'indrel_1mes':'Customer Type at Beginning of Month',
                  'tiprel_1mes':'Customer Relation Type at Beginning of Month',
                  'indresi':'Resident Country is Bank Country',
                  'indext':'Birth Country Different Than Bank Country',
                  'conyuemp':'Spouse Index',
                  'canal_entrada':'Channel Used to Join',
                  'indfall':'Deceased',
                  'tipodom':'Address Type',
                  'cod_prov':'Province Code',
                  'nomprov':'Province Name',
                  'ind_actividad_cliente':'Activity Index',
                  'renta':'Gross Household Income',
                  'segmento':'Segmentation',
                  'ind_ahor_fin_ulti':'Savings Account',
                  'indaval_fin_ulti':'Guarantees',
                  'ind_cco_fin_ulti':'Current Accounts',
                  'ind_cder_fin_ulti':'Derivative Account',
                  'ind_cno_fin_ulti':'Payroll Account',
                  'ind_ctju_fin_ulti':'Junior Account',
                  'ind_ctma_fin_ulti':'More Particular Account',
                  'ind_ctop_fin_ulti':'Particular Account',
                  'ind_ctpp_fin_ulti':'Particular Plus Account',
                  'ind_deco_fin_ulti':'Short-Term Deposits',
                  'ind_deme_fin_ulti':'Medium-Term Deposits',
                  'ind_dela_fin_ulti':'Long-Term Deposits',
                  'ind_ecue_fin_ulti':'E-Account',
                  'ind_fond_fin_ulti':'Funds',
                  'ind_hip_fin_ulti':'Mortgage',
                  'ind_plan_fin_ulti':'Pensions',
                  'ind_pres_fin_ulti':'Loans',
                  'ind_reca_fin_ulti':'Taxes',
                  'ind_tjcr_fin_ulti':'Credit Card',
                  'ind_valo_fin_ulti':'Securities',
                  'ind_viv_fin_ulti':'Home Account',
                  'ind_nomina_ulti':'Payroll',
                  'ind_nom_pens_ulti':'Pension',
                  'ind_recibo_ulti':'Direct Debit'}, inplace = True)
```

# Data Cleansing and Transformation

- ❖ Then, I found the mean, median, mode, and other attributes of a couple variables to have experience with them. Using the median, I was able to fill the NA values for each variable, not changing anything about the data and the trends. Also, the amount of NA values for each variable I changed wouldn't have enough weight to statistically significantly impact the data. Finally, I changed some of the numeric variables from objects to integers or float to make exploring the data easier.

```
df['Sex'] = df['Sex'].replace(['H'],[1])
df['Sex'] = df['Sex'].replace(['V'],[0])
df['Sex'] = df['Sex'].astype(float)
df['Sex'] = df['Sex'].fillna(0)

df['Age'] = df['Age'].replace(['NA'],['39'])
df['Age'] = df['Age'].astype(int)

df['Seniority'] = df['Seniority'].replace(['NA'],[50])
df['Seniority'] = df['Seniority'].replace([-999999],[50])
df['Seniority'] = df['Seniority'].astype(int)

df['Resident Country is Bank Country'] = df['Resident Country is Bank Country'].replace(['N'],[0])
df['Resident Country is Bank Country'] = df['Resident Country is Bank Country'].replace(['S'],[1])
df['Resident Country is Bank Country'] = df['Resident Country is Bank Country'].astype(float)
df['Resident Country is Bank Country'] = df['Resident Country is Bank Country'].fillna(1)

df['Birth Country Different Than Bank Country'] = df['Birth Country Different Than Bank Country'].replace(['N'],[0])
df['Birth Country Different Than Bank Country'] = df['Birth Country Different Than Bank Country'].replace(['S'],[1])
df['Birth Country Different Than Bank Country'] = df['Birth Country Different Than Bank Country'].astype(float)
df['Birth Country Different Than Bank Country'] = df['Birth Country Different Than Bank Country'].fillna(0)

df['Spouse Index'] = df['Spouse Index'].replace(['N'],[0])
df['Spouse Index'] = df['Spouse Index'].replace(['S'],[1])
df['Spouse Index'] = df['Spouse Index'].astype(float)

df['Deceased'] = df['Deceased'].replace(['N'],[0])
df['Deceased'] = df['Deceased'].replace(['S'],[1])
df['Deceased'] = df['Deceased'].astype(float)
df['Deceased'] = df['Deceased'].fillna(0)

df['Activity Index'] = df['Activity Index'].fillna(0)
df['Payroll'] = df['Payroll'].fillna(0)
df['Pension'] = df['Pension'].fillna(0)
df['New Customer Index'] = df['New Customer Index'].fillna(0)
df['Primary Customer'] = df['Primary Customer'].fillna(1)
df['Gross Household Income'] = df['Gross Household Income'].fillna(101850)
```

# Data Cleansing and Transformation

Dataframe Variable Types		Dataframe Variable Types		Dataframe NA Count		Dataframe NA Count	
Partitioned Date	object	Partitioned Date	object	Partitioned Date	0	Partitioned Date	0
Customer Code	int64	Customer Code	int64	Customer Code	0	Customer Code	0
Employee Index	object	Employee Index	object	Employee Index	27734	Employee Index	27734
Country Residence	object	Country Residence	object	Country Residence	27734	Country Residence	27734
Sex	object	Sex	float64	Sex	27804	Sex	0
Age	object	Age	int32	Age	0	Age	0
Date Joined	object	Date Joined	object	Date Joined	27734	Date Joined	27734
New Customer Index	float64	New Customer Index	float64	New Customer Index	27734	New Customer Index	0
Seniority	object	Seniority	int32	Seniority	0	Seniority	0
Primary Customer	float64	Primary Customer	float64	Primary Customer	27734	Primary Customer	0
Last Date as Primary Customer	object	Last Date as Primary Customer	object	Last Date as Primary Customer	13622516	Last Date as Primary Customer	0
Customer Type at Beginning of Month	object	Customer Type at Beginning of Month	object	Customer Type at Beginning of Month	149781	Customer Type at Beginning of Month	0
Customer Relation Type at Beginning of Month	object	Customer Relation Type at Beginning of Month	object	Customer Relation Type at Beginning of Month	149781	Customer Relation Type at Beginning of Month	0
Resident Country is Bank Country	object	Resident Country is Bank Country	float64	Resident Country is Bank Country	27734	Resident Country is Bank Country	0
Birth Country Different Than Bank Country	object	Birth Country Different Than Bank Country	float64	Birth Country Different Than Bank Country	27734	Birth Country Different Than Bank Country	0
Spouse Index	object	Spouse Index	float64	Spouse Index	13645501	Spouse Index	0
Channel Used to Join	object	Channel Used to Join	object	Channel Used to Join	186126	Channel Used to Join	0
Deceased	object	Deceased	float64	Deceased	27734	Deceased	0
Address Type	float64	Address Type	float64	Address Type	27735	Address Type	0
Province Code	float64	Province Code	float64	Province Code	93591	Province Code	0
Province Name	object	Province Name	object	Province Name	93591	Province Name	0
Activity Index	float64	Activity Index	float64	Activity Index	27734	Activity Index	0
Gross Household Income	float64	Gross Household Income	float64	Gross Household Income	2794375	Gross Household Income	0
Segmentation	object	Segmentation	object	Segmentation	189368	Segmentation	0
Savings Account	int64	Savings Account	int64	Savings Account	0	Savings Account	0
Guarantees	int64	Guarantees	int64	Guarantees	0	Guarantees	0
Current Accounts	int64	Current Accounts	int64	Current Accounts	0	Current Accounts	0
Derivative Account	int64	Derivative Account	int64	Derivative Account	0	Derivative Account	0
Payroll Account	int64	Payroll Account	int64	Payroll Account	0	Payroll Account	0
Junior Account	int64	Junior Account	int64	Junior Account	0	Junior Account	0
More Particular Account	int64	More Particular Account	int64	More Particular Account	0	More Particular Account	0
Particular Account	int64	Particular Account	int64	Particular Account	0	Particular Account	0
Particular Plus Account	int64	Particular Plus Account	int64	Particular Plus Account	0	Particular Plus Account	0
Short-Term Deposits	int64	Short-Term Deposits	int64	Short-Term Deposits	0	Short-Term Deposits	0
Medium-Term Deposits	int64	Medium-Term Deposits	int64	Medium-Term Deposits	0	Medium-Term Deposits	0
Long-Term Deposits	int64	Long-Term Deposits	int64	Long-Term Deposits	0	Long-Term Deposits	0
E-Account	int64	E-Account	int64	E-Account	0	E-Account	0
Funds	int64	Funds	int64	Funds	0	Funds	0
Mortgage	int64	Mortgage	int64	Mortgage	0	Mortgage	0
Pensions	int64	Pensions	int64	Pensions	0	Pensions	0
Loans	int64	Loans	int64	Loans	0	Loans	0
Taxes	int64	Taxes	int64	Taxes	0	Taxes	0
Credit Card	int64	Credit Card	int64	Credit Card	0	Credit Card	0
Securities	int64	Securities	int64	Securities	0	Securities	0
Home Account	int64	Home Account	int64	Home Account	0	Home Account	0
Payroll	float64	Payroll	float64	Payroll	16063	Payroll	0
Pension	float64	Pension	float64	Pension	16063	Pension	0
Direct Debit	int64	Direct Debit	int64	Direct Debit	0	Direct Debit	0

# Exploratory Data Analysis

- I began the exploratory data analysis (EDA) by taking the little experience I gained from cleaning and transforming the data and applying it by finding all the count values of each variable and the sum of each variable to get a vague sense of how many accounts there were in all the options that the bank offered.

Dataframe Count	
Partitioned Date	13647309
Customer Code	13647309
Employee Index	13619575
Country Residence	13619575
Sex	13647309
Age	13647309
Date Joined	13619575
New Customer Index	13647309
Seniority	13647309
Primary Customer	13647309
Last Date as Primary Customer	24793
Customer Type at Beginning of Month	13497528
Customer Relation Type at Beginning of Month	13497528
Resident Country is Bank Country	13647309
Birth Country Different Than Bank Country	13647309
Spouse Index	1808
Channel Used to Join	13461183
Deceased	13647309
Address Type	13619574
Province Code	13553718
Province Name	13553718
Activity Index	13647309
Gross Household Income	13647309
Segmentation	13457941
Savings Account	13647309
Guarantees	13647309
Current Accounts	13647309
Derivative Account	13647309
Payroll Account	13647309
Junior Account	13647309
More Particular Account	13647309
Particular Account	13647309
Particular Plus Account	13647309
Short-Term Deposits	13647309
Medium-Term Deposits	13647309
Long-Term Deposits	13647309
E-Account	13647309
Funds	13647309
Mortgage	13647309
Pensions	13647309
Loans	13647309
Taxes	13647309
Credit Card	13647309
Securities	13647309
Home Account	13647309
Payroll	13647309
Pension	13647309
Direct Debit	13647309
dtype: int64	

Dataframe Sum	
Sex	6.195253e+06
Age	5.483361e+08
New Customer Index	8.112070e+05
Seniority	1.082538e+09
Primary Customer	1.607702e+07
Resident Country is Bank Country	1.358144e+07
Birth Country Different Than Bank Country	6.447360e+05
Deceased	3.476200e+04
Activity Index	6.235185e+06
Gross Household Income	1.741660e+12
Savings Account	1.396000e+03
Guarantees	3.160000e+02
Current Accounts	8.945588e+06
Derivative Account	5.376000e+03
Payroll Account	1.103620e+06
Junior Account	1.292970e+05
More Particular Account	1.327420e+05
Particular Account	1.760616e+06
Particular Plus Account	5.910080e+05
Short-Term Deposits	2.427500e+04
Medium-Term Deposits	2.266800e+04
Long-Term Deposits	5.863810e+05
E-Account	1.129227e+06
Funds	2.522840e+05
Mortgage	8.033600e+04
Pensions	1.251590e+05
Loans	3.585700e+04
Taxes	7.169800e+05
Credit Card	6.057860e+05
Securities	3.494750e+05
Home Account	5.251100e+04
Payroll	7.459610e+05
Pension	8.100850e+05
Direct Debit	1.745712e+06
dtype: float64	

# Exploratory Data Analysis

- ❖ I then found the means and medians of each variable to see which accounts of the bank were doing better than the others.

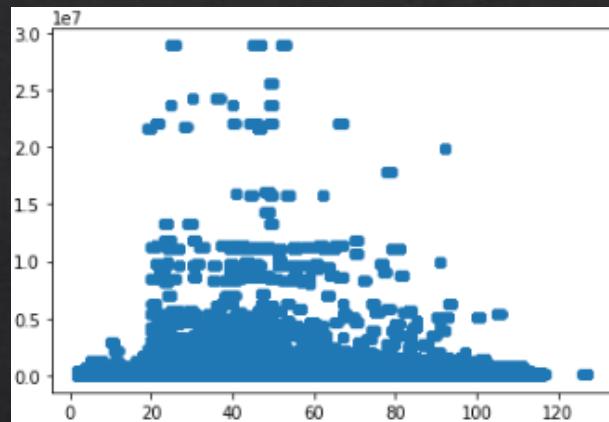
Dataframe Mean	
Sex	0.453954
Age	40.179064
New Customer Index	0.059441
Seniority	79.322469
Primary Customer	1.178036
Resident Country is Bank Country	0.995174
Birth Country Different Than Bank Country	0.047243
Deceased	0.002547
Activity Index	0.456880
Gross Household Income	127619.323986
Savings Account	0.000102
Guarantees	0.000023
Current Accounts	0.655484
Derivative Account	0.000394
Payroll Account	0.080867
Junior Account	0.009474
More Particular Account	0.009727
Particular Account	0.129008
Particular Plus Account	0.043306
Short-Term Deposits	0.001779
Medium-Term Deposits	0.001661
Long-Term Deposits	0.042967
E-Account	0.082744
Funds	0.018486
Mortgage	0.005887
Pensions	0.009171
Loans	0.002627
Taxes	0.052536
Credit Card	0.044389
Securities	0.025608
Home Account	0.003848
Payroll	0.054660
Pension	0.059359
Direct Debit	0.127916
dtype: float64	

Dataframe Median	
Sex	0.0
Age	39.0
New Customer Index	0.0
Seniority	50.0
Primary Customer	1.0
Resident Country is Bank Country	1.0
Birth Country Different Than Bank Country	0.0
Deceased	0.0
Activity Index	0.0
Gross Household Income	101850.0
Savings Account	0.0
Guarantees	0.0
Current Accounts	1.0
Derivative Account	0.0
Payroll Account	0.0
Junior Account	0.0
More Particular Account	0.0
Particular Account	0.0
Particular Plus Account	0.0
Short-Term Deposits	0.0
Medium-Term Deposits	0.0
Long-Term Deposits	0.0
E-Account	0.0
Funds	0.0
Mortgage	0.0
Pensions	0.0
Loans	0.0
Taxes	0.0
Credit Card	0.0
Securities	0.0
Home Account	0.0
Payroll	0.0
Pension	0.0
Direct Debit	0.0
dtype: float64	

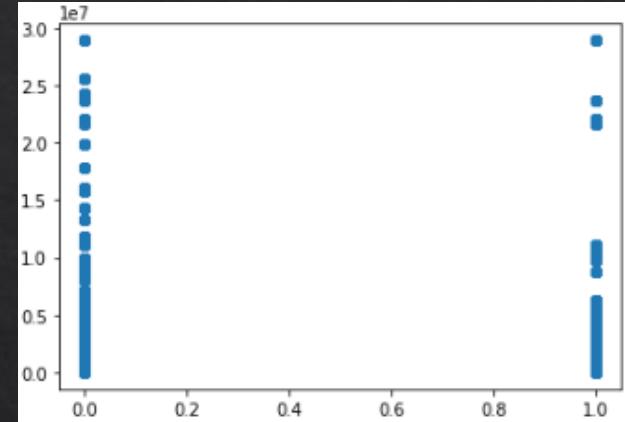
# Exploratory Data Analysis

- ❖ Then, I created some visualizations with the variables and compared them to gross household income. I did this assuming the greater the wealth of a family, the greater the possible amount of money that could be trusted with the bank. Though the visualizations helped a little, they couldn't paint a full picture.

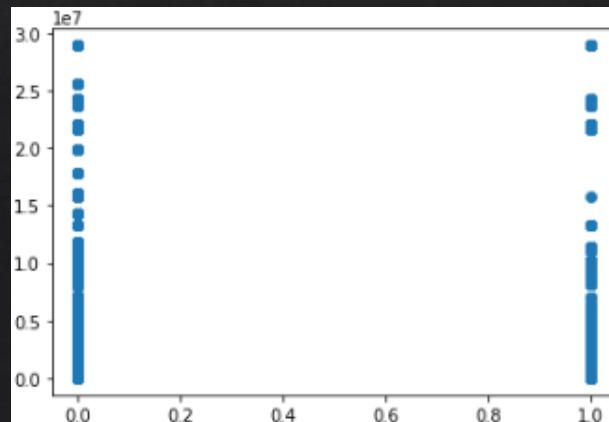
Gross Household Income by Age



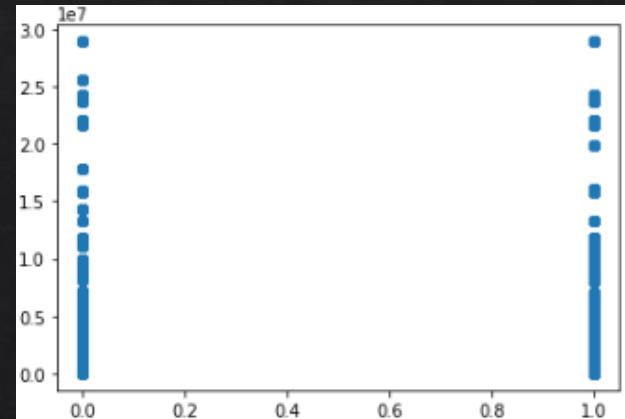
Gross Household Income by E-Account



Gross Household Income by Direct Debit



Gross Household Income by Activity Index



# Exploratory Data Analysis

- ❖ I decided to find the correlation (Pearson's r) values between each of the variables, with special interest in gross household income, to see if I could find any correlations between them. Although it wasn't giving me all the information I wanted, specifically for the variables of Sex and Age, I changed a couple of the variables to make them able to be correlated. I then analyzed all the statistically significant correlations and they helped me come to a final conclusion. Therefore, with the information I received before from the means and medians, and the recent information with the correlation coefficients, I was able to come to a final recommendation.



## Statistically Significant Findings

	Age	Resident Country is Bank Country	Birth Country Different Than Bank Country	Spouse Index	Activity Index	Current Accounts	Payroll Account	Particular Account	Particular Plus Account	Long-Term Deposits	E-Account	Taxes	Credit Card	Payroll	Pension	Direct Debit	
Age	1	-0.02946	-0.01652	-0.08281	0.135249	-0.16767	0.068959	0.317002	0.117488	0.175474	0.142814	0.087455	0.10082	0.043252	0.07142	0.08864	
Resident Country is Bank Country		-0.02946	1	-0.246658	Nan	-0.01376	-0.01251	0.016264	-0.02021	0.00404	0.001642	0.004881	0.007568	0.003258	0.014824	0.014919	0.011704
Birth Country Different Than Bank Country		-0.01652	-0.24666	1	-0.01244	-0.02109	-0.02366	-0.00474	-0.03416	-0.01259	-0.03154	-0.028999	0.007773	-0.001367	-0.00689	-0.00856	-0.003418
Spouse Index		-0.08281	Nan	-0.012439	1	0.041176	-0.11853	0.172047	-0.06007	-0.05426	-0.04302	0.053851	-0.05459	-0.074667	0.227092	0.21249	0.049243
Activity Index		0.135249	-0.01376	-0.021085	0.041176	1	0.170152	0.31544	0.169366	0.156211	0.228881	0.284509	0.222989	0.23232	0.260518	0.272069	0.412218
Current Accounts		-0.16767	-0.01251	-0.023655	-0.11853	0.170152	1	-0.34547	0.003798	-0.05078	0.010251	-0.076882	-0.09158	-0.095949	-0.25773	-0.27089	-0.069308
Payroll Account		0.068959	0.016264	-0.004743	0.172047	0.31544	-0.34547	1	0.021291	0.186533	0.095624	0.306321	0.319811	0.386592	0.757957	0.790128	0.535626
Particular Account		0.317002	-0.02021	-0.034163	-0.06007	0.169366	0.003798	0.021291	1	0.037871	0.077666	-0.0102	0.061071	0.063697	0.027068	0.028114	0.04304
Particular Plus Account		0.117488	0.00404	-0.012588	-0.05426	0.156211	-0.05078	0.186533	0.037871	1	0.074684	0.067205	0.168267	0.167066	0.172543	0.179819	0.201764
Long-Term Deposits		0.175474	0.001642	-0.03154	-0.04302	0.228881	0.010251	0.095624	0.077666	0.074684	1	0.22212	0.063987	0.106513	0.078139	0.090496	0.09392
E-Account		0.142814	0.004881	-0.028999	0.053851	0.284509	-0.07688	0.306321	-0.0102	0.067205	0.22212	1	0.200133	0.255686	0.263043	0.281967	0.296159
Taxes		0.087455	0.007568	0.007773	-0.05459	0.222989	-0.09158	0.319811	0.061071	0.168267	0.063987	0.200133	1	0.269161	0.277961	0.294729	0.343852
Credit Card		0.10082	0.003258	-0.001367	-0.07467	0.23232	-0.09595	0.386592	0.063697	0.167066	0.106513	0.255686	0.269161	1	0.356682	0.370464	0.37466
Payroll		0.043252	0.014824	-0.006887	0.227092	0.260518	-0.25773	0.757957	0.027068	0.172543	0.078139	0.263043	0.277961	0.356682	1	0.957215	0.471138
Pension		0.07142	0.014919	-0.008559	0.21249	0.272069	-0.27089	0.790128	0.028114	0.179819	0.090496	0.281967	0.294729	0.370464	0.957215	1	0.490888
Direct Debit		0.08864	0.011704	-0.003418	0.049243	0.412218	-0.06931	0.535626	0.04304	0.201764	0.09392	0.296159	0.343852	0.37466	0.471138	0.490888	1
# of Greens		1	1	1	2	8	3	8	1	1	2	8	7	7	9	9	8

# Final Recommendation

- ❖ I would like to begin this final recommendation with the saying that, “Correlation does not imply causation”. Although, trends say one thing, you are not always guaranteed the outcome you seek. With that being said, my final recommendation is this. In the prompt, we are told the XYZ Credit Union is performing very well in the selling of their banking products. Although they aren’t where they’d like to be with cross selling to their existing customers, they are still doing well selling individual products. They might not be maximizing their revenue, but their business model is drawing in new customers and keeping them long term. Pushing new accounts on your existing customers could turn them away. I would say you could try cross selling to new customers that enter your credit union, but not the ones who have been loyal for many years. Despite this, there are ways of making more profit by cross selling certain accounts together. So, if you did want to try making more revenue through cross selling, this is how you could do it.

# Final Recommendation

- ❖ Shockingly, from the correlation coefficient findings, gross household income is not a good variable which should be tested with bank success. It wasn't statistically significant with any other of the variables we used. The variables best correlated with other banking variables were activity index, payroll account, e-account, taxes, credit card, payroll, pensions, and direct debit. Looking deeper into a few of those variables, starting with activity index, we can see that it's statistically significantly positively correlated with payroll accounts, long-term deposits, e-accounts, taxes, credit card, payroll, pensions, and direct debit. In short, this means that customers who are active with their account tend to also have multiple types of banking accounts, specifically the accounts listed above. This means that if you were trying to cross sell, do it with active customers. Try getting them signed up for one of these account types, with a priority in direct debit, e-accounts, payroll accounts, pensions, payroll, taxes, and credit card, as these are the most used accounts out of all their customers. Moreover, these accounts are all statistically significantly positively correlated with each other.

# Final Recommendation

- ❖ If you have an older customer, the statistics show that they are more likely to have a particular account with the credit union. Also, particular accounts are one of the most popular accounts to have in this credit union. Thus, I would try pushing those types of accounts to older customers who don't already have one.
- ❖ Customers that already have an account at the credit union tend to not have a payroll account or pension. This probably means that they use other banking competitors to handle those accounts. Maybe with some more open minded customers, you could be able to convince them to switch over to your credit union. This would make those variables more highly correlated, especially if they are an active customer, and could create more usage for your credit union.

# Final Recommendation

- ❖ Particular plus accounts have a statistically significantly positive relationship with direct debit. This makes sense as particular plus accounts deal with checking accounts and so do debit cards. Therefore, if someone is trying to sign up for a checking (particular plus) account, try offering direct debit, as it would line up with the wants of the customer.
- ❖ To keep up with an increasing technologically driven world, the statistics and I highly recommend pushing e-accounts. It is positively correlated with many of the highest used banking accounts, and it provides simplicity and convenience for many, if not all, of your customers.

# Final Recommendation

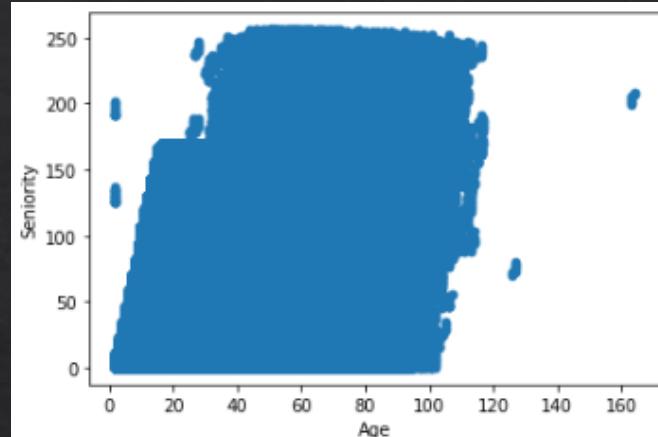
- ❖ There are no statistically significant differences between gender and banking. Although 55% of the customers are male, there is no difference in the amount of accounts that both males and females use. Thus, you shouldn't treat them differently, and you shouldn't offer different accounts based on sex.
- ❖ Finally, using the means of each of the different banking account types, I have found that the top five most used accounts are: particular account, direct debit, e-account, payroll account, and pension; and the five least used accounts are: guarantees, savings account, derivative account, medium-term deposits, and short-term deposits. With this information, you can promote what is working best for the XYZ Credit Union, or try promoting the lesser used accounts in hopes of attracting different types of customers. However, I wouldn't try and deviate too far from what is already being done, since the XYZ Credit Union is doing well already.

# Modeling Technique

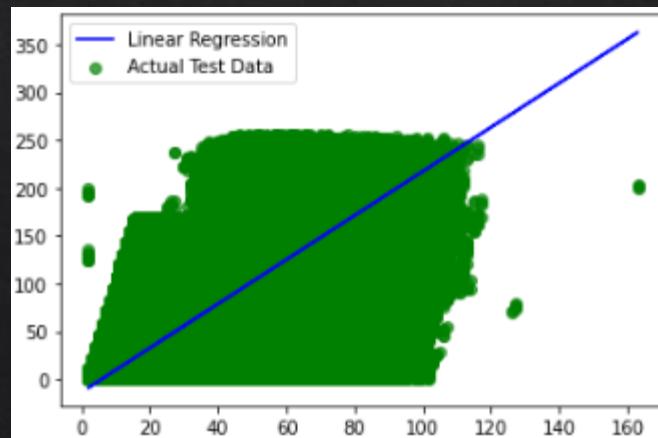
- ❖ The best models for this problem/dataset should be predictive models. Specifically, models that use linear and multiple regression. Using the statistically significant values from the correlation coefficient table, along with means, medians, counts, sums, and regression equations, we can attempt to predict the best combination of cross selling techniques for the XYZ Credit Union.

# Linear Regression Model

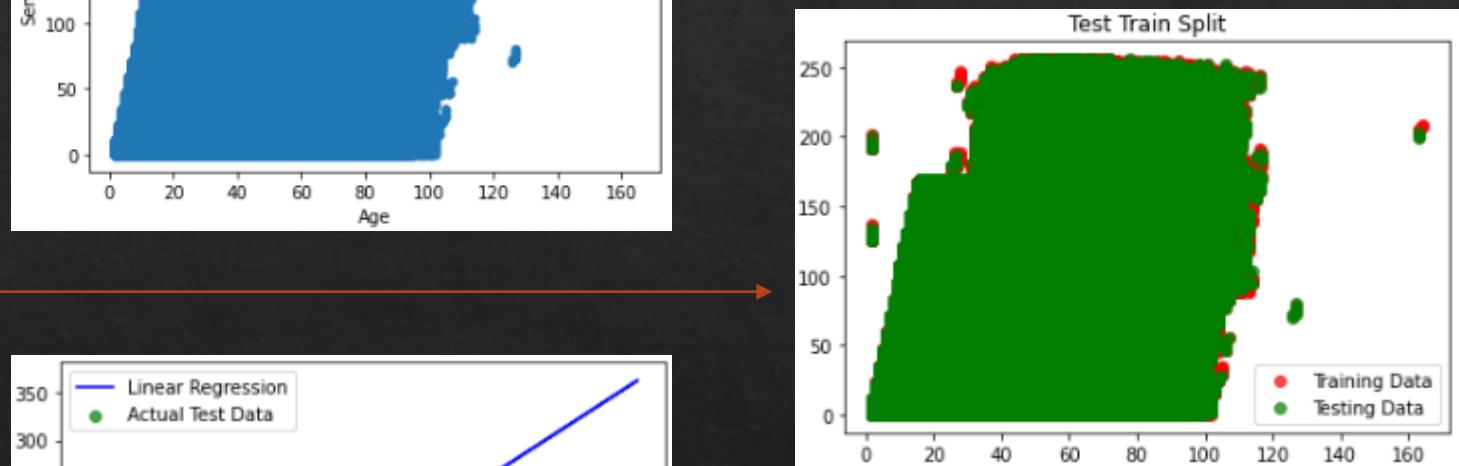
- ◊ The first predictive model I used to attempt to predict the best combination of cross selling techniques was linear regression. I started with a regular scatterplot to get an overall understanding of each of the variables being used.



- ◊ I then used the train, test, split as my method of prediction, and made a graph.



- ◊ Next, I plotted the actual test data with a linear regression line to make an accurate graph.



- ◊ Finally, I created a regression equation, with a linear regression score, and was able to make a prediction based on inputting a number for x.

```
Prediction 0.18993595689775208
Intercept -0.12016394257993027
Slope [0.006202]
Regression Equation: y = [0.006202] x + -0.12016394257993027
Linear Regression Score 0.10090753170178268
```

# Multiple Regression Model

- ◆ The second predictive model I used was multiple regression. I focused less on visualizations for multiple regression as it is similar to linear regression, except it has more regression lines. I took the variables with two or more other variables that were statistically significantly correlated to it and tested the multiple regression. I was able to make a prediction for each dependent variable, along with finding the regression equation with the slope values and an intercept. Here is an example of what the output looks like:

```
Prediction [0.60601549]
Coefficients [ 0.07142447  0.10559897  0.20986961  0.06764832  0.13505848 -0.08433393
               0.13916534  0.07420132]
Intercept 0.009862632917611497
Regression Equation: y = [ 0.07142447  0.10559897  0.20986961  0.06764832  0.13505848 -0.08433393
               0.13916534  0.07420132] x + 0.009862632917611497
```

# Ensemble Model

- ❖ I noticed while using linear regression that the prediction score was very low for most of the variables. Multiple regression would make that number a little better but not enough to feel confident in an accurate prediction. Thus, I decided to use an ensemble model to see if I could increase prediction accuracy. Similar to linear regression, I used a test, train, split model to find a baseline prediction. Then, using a bagging classifier and increasing the number, and therefore accuracy, of parameters like n estimators, I was able to output a better score of prediction. The top number is the model using the train values of the variables, and the bottom number is the model using the test values of the variables. We can see that the outcomes are very good, all producing an accuracy score of 95% or better.

```
bag_model = BaggingClassifier(  
    base_estimator = DecisionTreeClassifier(),  
    n_estimators = 10,  
    max_samples = 0.8,  
    oob_score = True,  
    random_state = 0)  
  
bag_model.fit(A_train, b_train)  
print(bag_model.oob_score_)  
  
print(bag_model.score(A_test, b_test))
```

0.9979651176139157

0.9982795146765898

# Boosted Model (Gradient Boosting Model)

- ◆ Although the numbers of the ensemble model were great, I think it's important to have multiple accurate models to use in case one isn't available to us, or to see if we can make the accuracy score better still. For this model, I used a boosted model (gradient boosting model) to test the accuracy of the model. This one too used a train, test, split approach, then, using a gradient boosting classifier, I was able to make two predictions. The top prediction based on the variables' tested accuracy score, and the bottom prediction testing it against the gradient boosting model's predicted variables. The model generated very accurate scores, much more accurate than the linear regression scores. However, not as accurate as the scores from the ensemble model. I would still have confidence using the boosted model.

```
gbModel = GradientBoostingClassifier(random_state = 42)

gbModel.fit(s_train, t_train)

t_pred_gb = gbModel.predict(s_test)

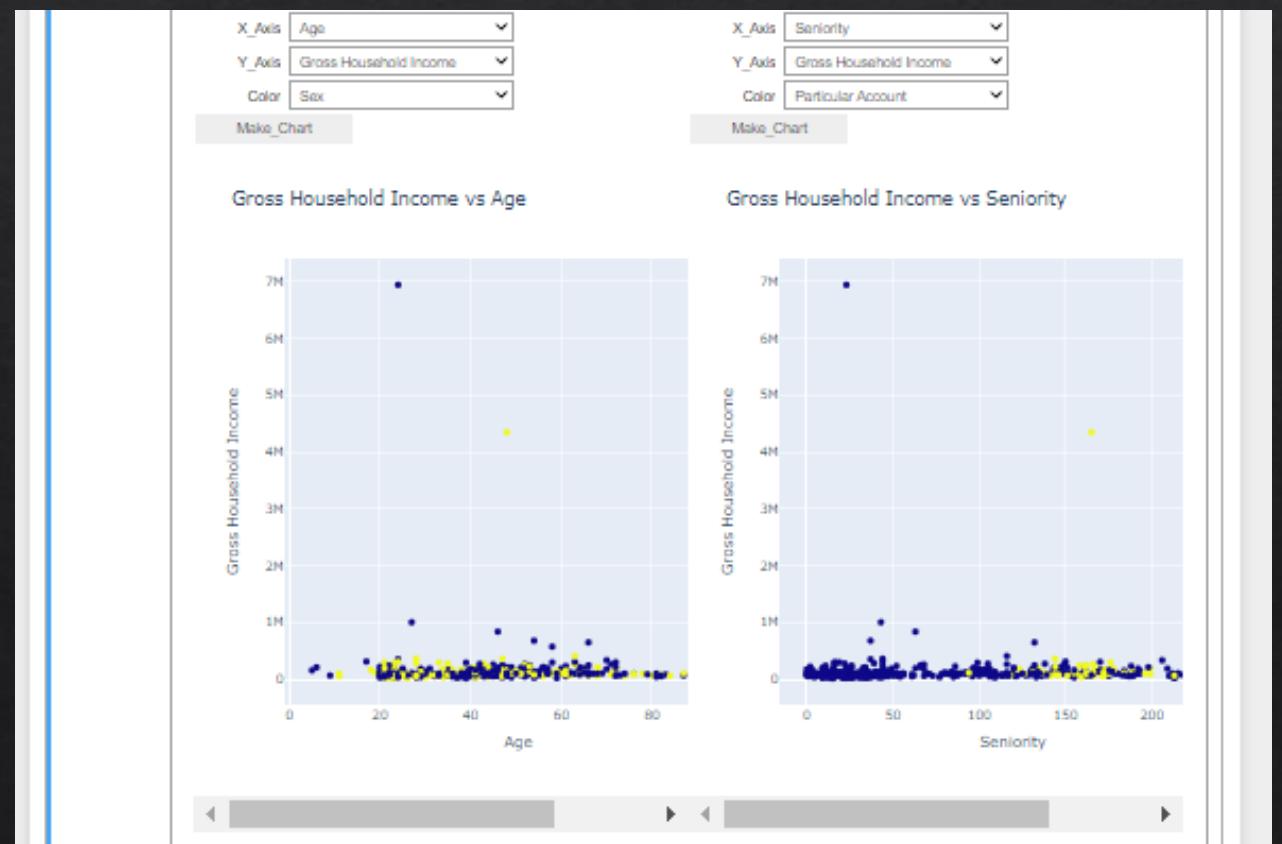
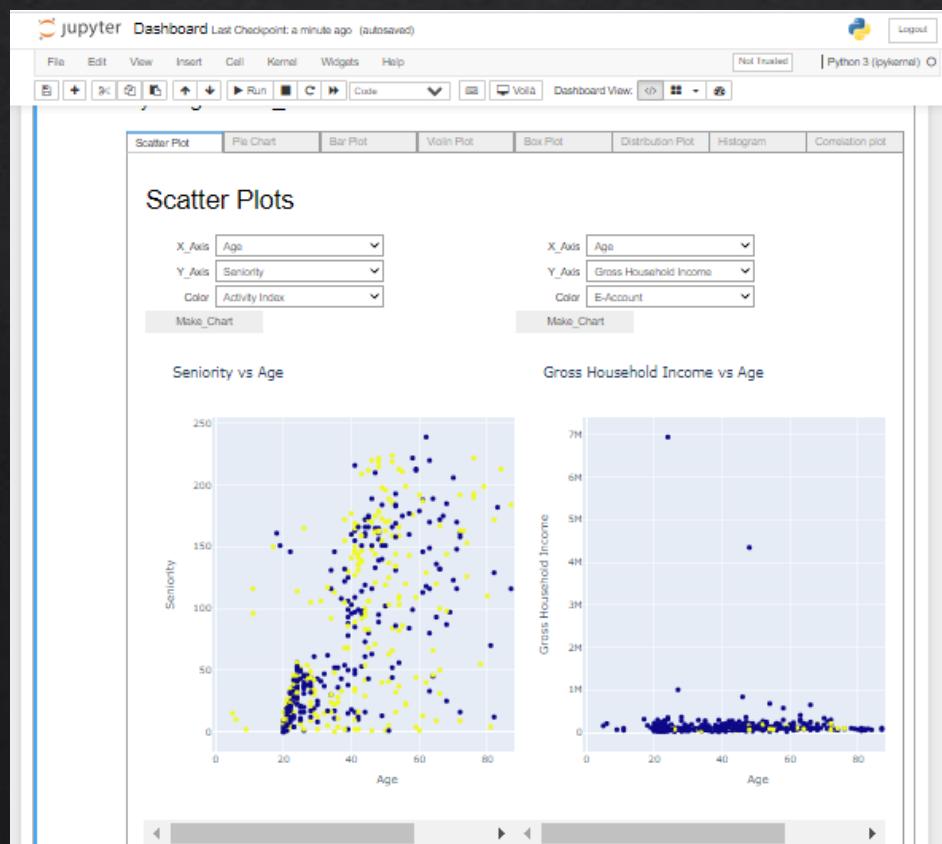
print(gbModel.score(s_test, t_test))

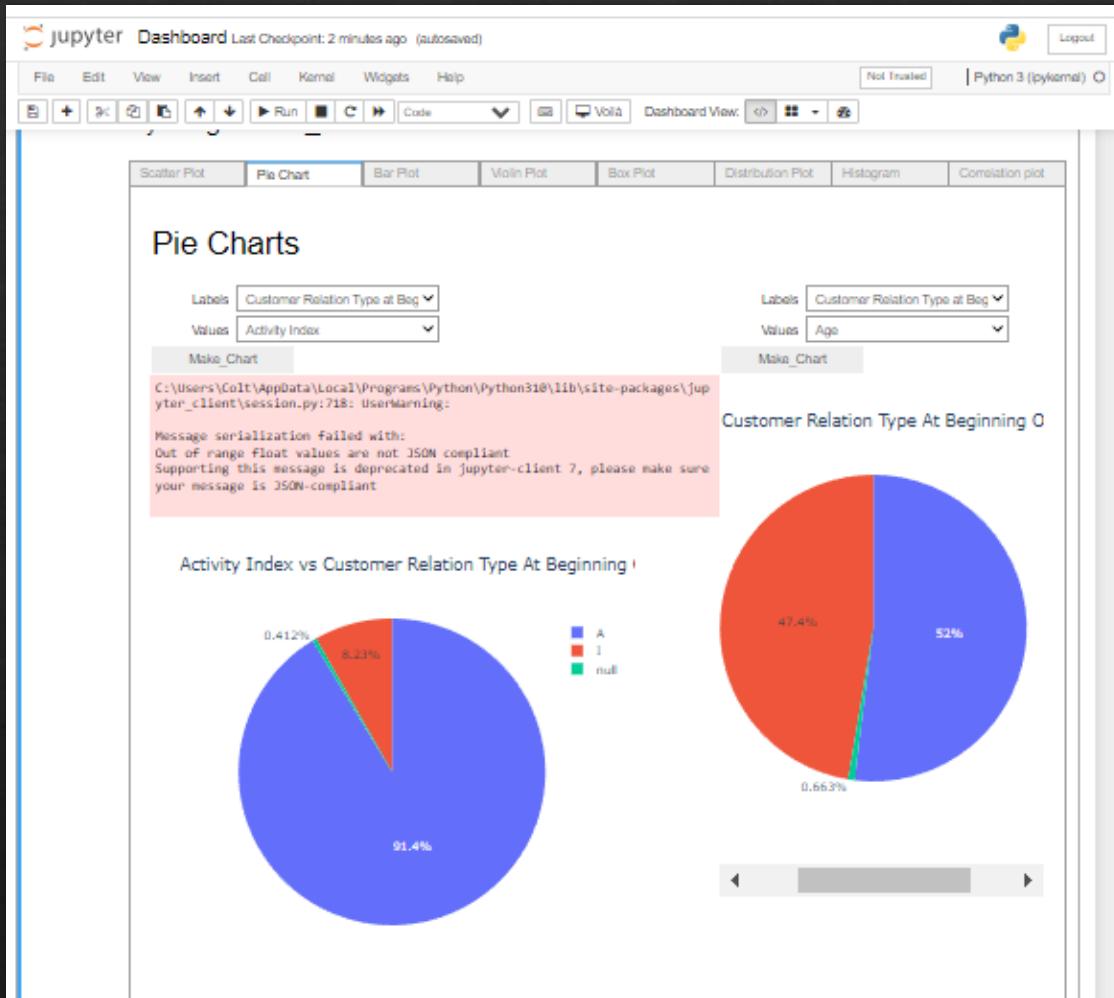
print(accuracy_score(t_test, t_pred_gb))

0.9256824238622849
0.9256824238622849
```

# Dashboard Visualizations

- ❖ The next couple slides will be visualizations I created for my dashboard. I won't be repeat the visualizations seen earlier in the presentation, although they should be considered part of the dashboard. More graphs can be created through Python; however, these were the graphs I chose to use as part of the presentation.





File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3 (ipykernel) O

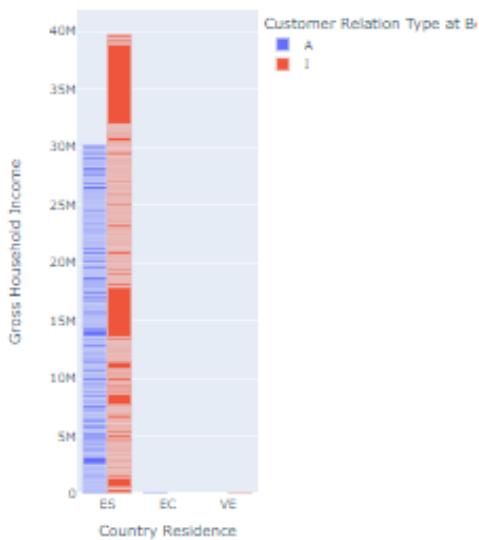
## Bar Plots

X\_Axis Country Residence

Y\_Axis Gross Household Income

Color Customer Relation Type at Beg

Country Residence vs Gross Household Income

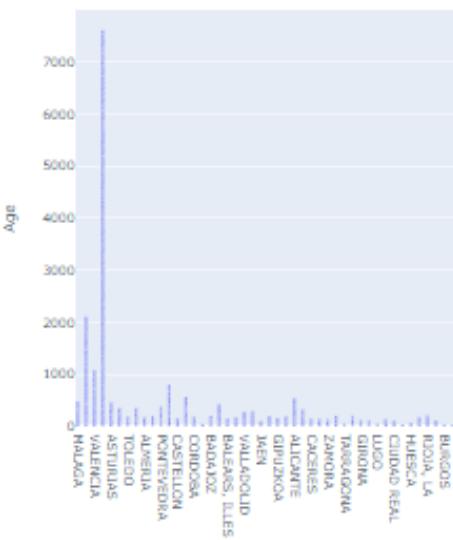


X\_Axis Province Name

Y\_Axis Age

Color Country Residence

Province Name vs Age

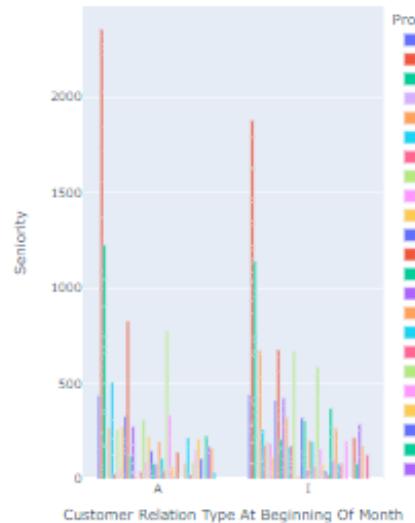


X\_Axis Customer Relation Type at Beg

Y\_Axis Seniority

Color Province Name

Customer Relation Type At Beginning Of Month vs Seniority

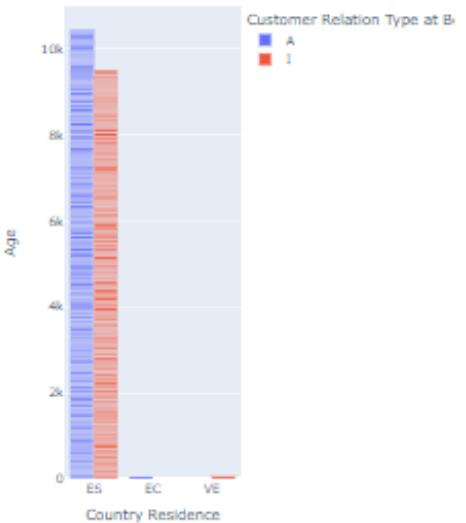


X\_Axis Country Residence

Y\_Axis Age

Color Customer Relation Type at Beg

Country Residence vs Age



Customer Relation Type at Beg

A

I



Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted

Python 3 (ipykernel) O

Code Volta Dashboard View

Scatter Plot Pie Chart Bar Plot Violin Plot Box Plot Distribution Plot Histogram Correlation plot

## Violin Plots

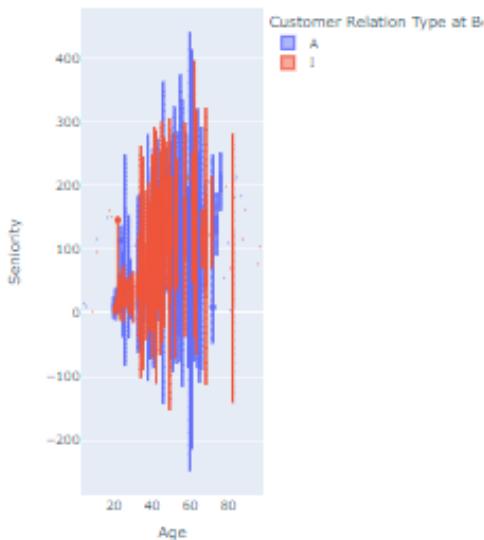
X\_Axis Age

Y\_Axis Seniority

Color Customer Relation Type at B

Make\_Chart

Age vs Seniority



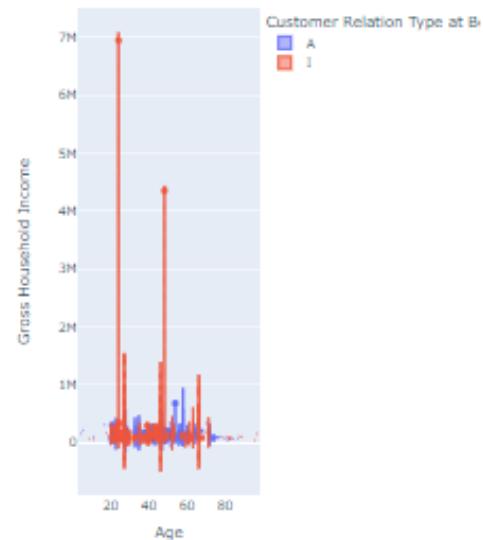
X\_Axis Age

Y\_Axis Gross Household Income

Color Customer Relation Type at B

Make\_Chart

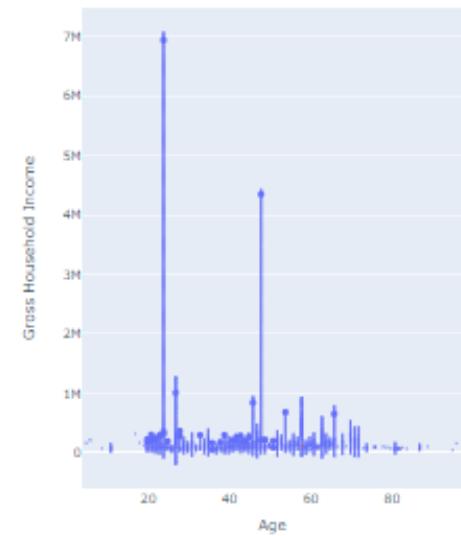
Age vs Gross Household Income



X\_Axis Age  
Y\_Axis Gross Household Income  
Color Country Residence

Make\_Chart

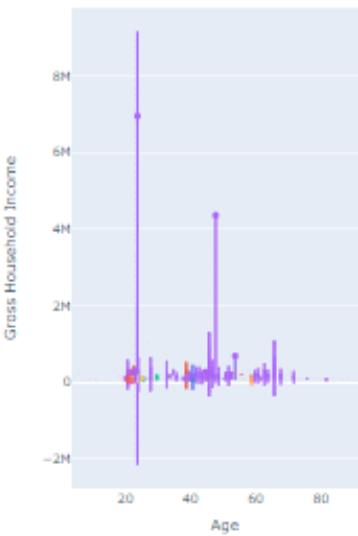
Age vs Gross Household Income



X\_Axis Age  
Y\_Axis Gross Household Income  
Color Province Name

Make\_Chart

Age vs Gross Household Income

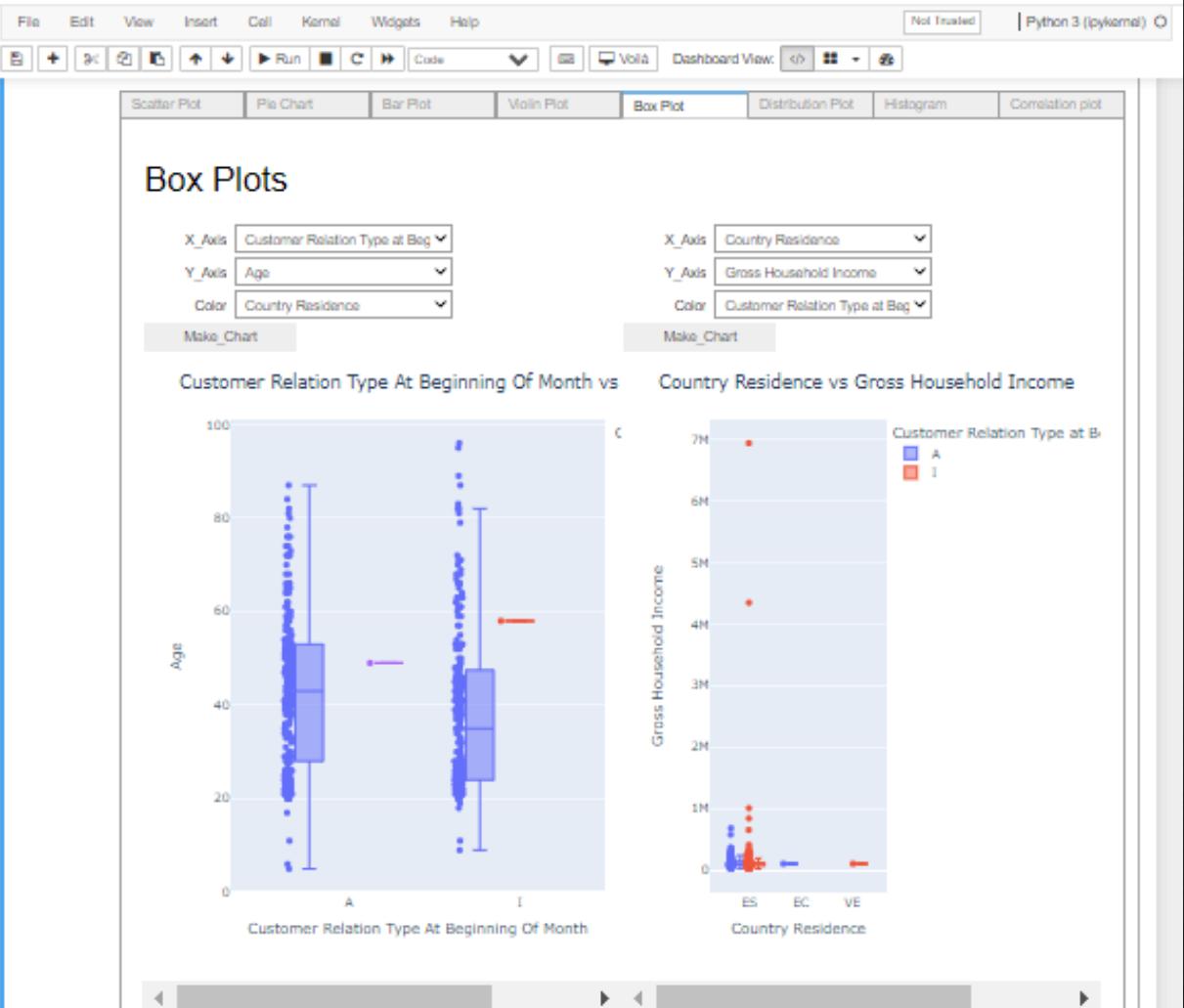


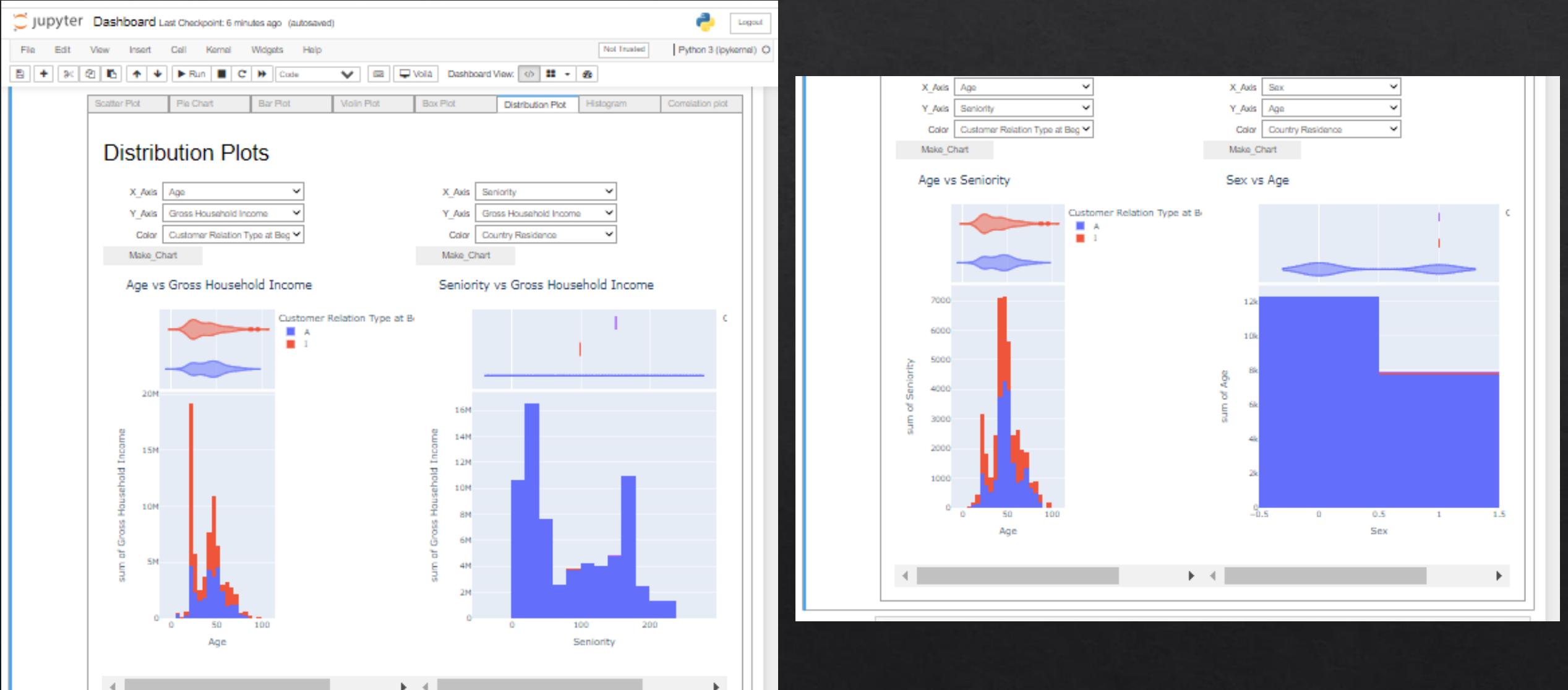
Province N:

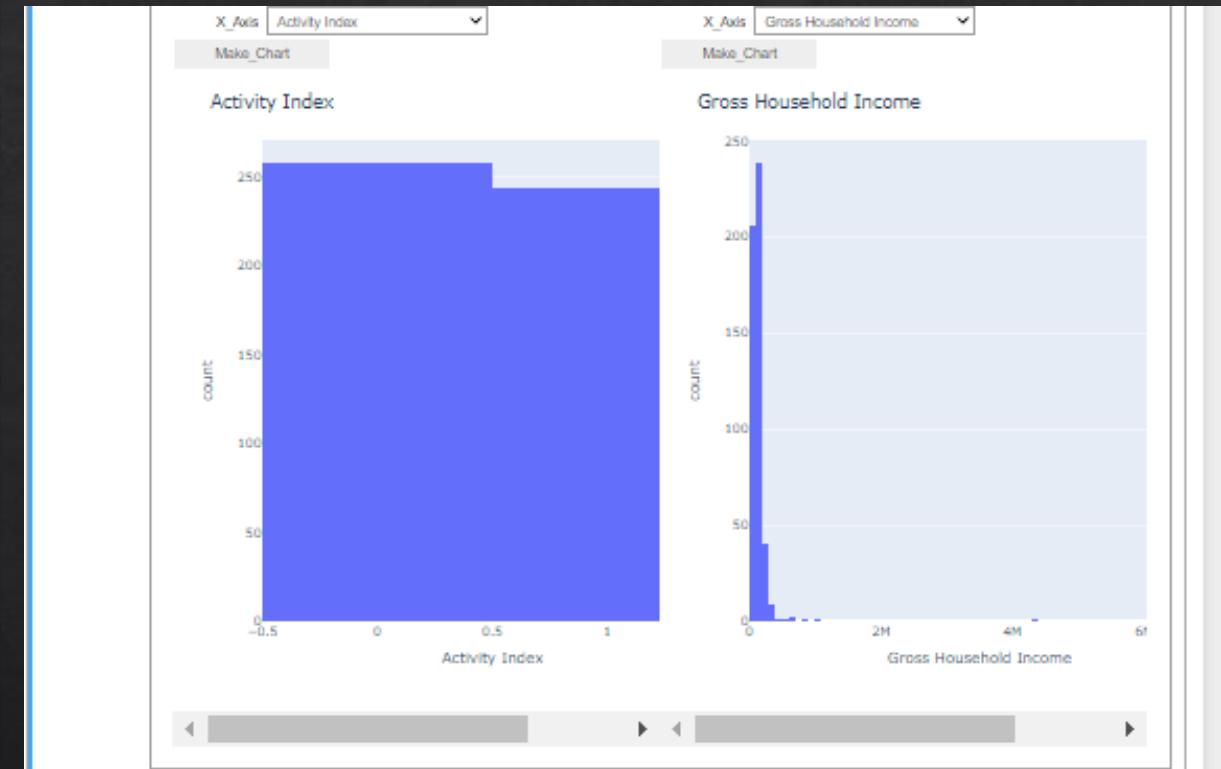
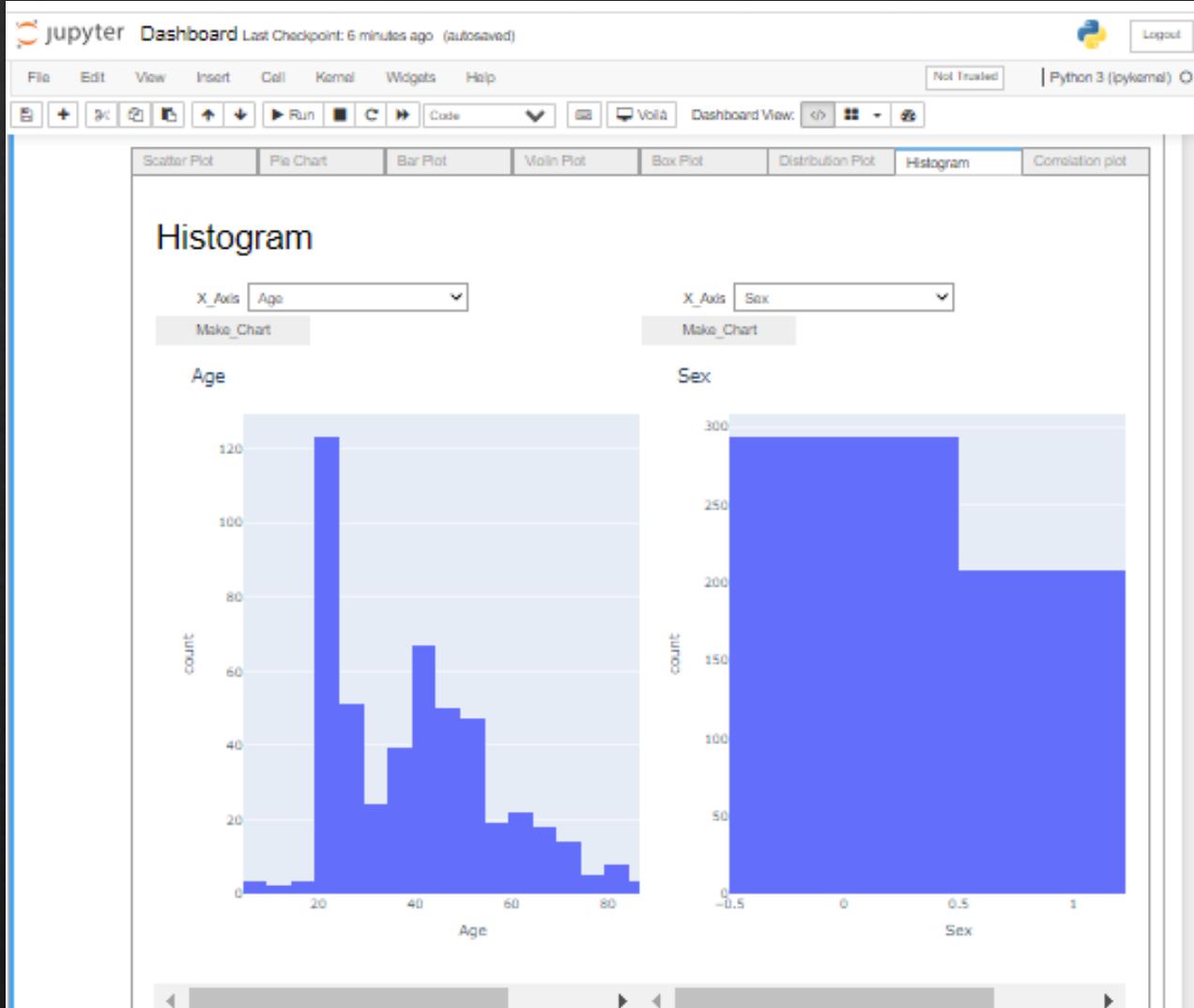
- MALAG
- BARCE
- VALEN
- MADRI
- ASTUR
- BIZKAI
- TOLED
- SALAM
- ALMER
- CANTA
- PONTE
- SEVILL
- CASTE
- CORUN
- CORDC
- SEGOV
- BADAJ
- PALMA
- BALEI
- SANTA
- VALLAI
- GRANA
- JAEN



Logout









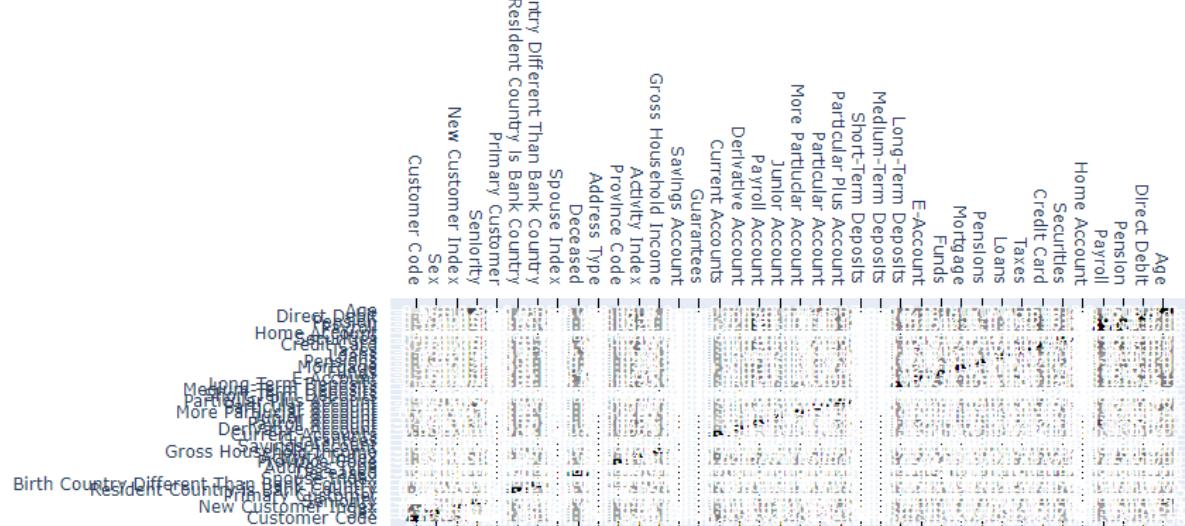
## Pywedge Make\_Charts

Scatter Plot Pie Chart Bar Plot Violin Plot Box Plot Distribution Plot Histogram Correlation plot

### Correlation Plots

colorscale Greys ▾

Make\_Chart



Thank you!