Summary - Data Wrangling with Panda

Conda Environment	`conda env export > environment.yml : existing environment`
	
	`conda env create -f environment.yml : create from file`

Imported Packages	`import pandas as pd`
	`import numpy as np`
	`%matplotlib inline`
	`import matplotlib.pyplot as plt`
	`from pandas.plotting import scatter_matrix`
	
	`%load_ext line_profiler`
	`import seaborn as sns`
	`import missingno as msno`
	
	`from sklearn.preprocessing import Imputer`
	`from sklearn.preprocessing import LabelBinarizer`

Read CSV	`df = pd.read_csv('application_train.csv')`
Glimpse Data in Dataframe	`df.head(6)`
Dataframe Informtion	`df.info(null_counts=True, verbose=True)`
Distinct observations	`df.nunique().sort_values()`
Variables that have 2 Values	`inter_cat = df.loc[:,df.nunique(axis=0).apply(lambda x: x < 3)]`
Sort values	`inter_cat.nunique().sort_values()`
Value of the Binary Variables	`for c in inter_cat.columns:` `  print("{}: \t {} ".format(c, df[c].unique()))`

Separate Categorical and Numeric Variables	`cat_vars = [*df.select_dtypes('object').columns.values.tolist(),`
	`*inter_cat.columns.values.tolist()]`
	
	`num_vars = df.columns.difference(cat_vars).values.tolist()`

Class imbalance	`df['TARGET'].value_counts()`
Overview of missing data	`import missingno as msno` `msno.matrix(df)`

Percentage of missing data	`total = df.isnull().sum().sort_values(ascending = False)`
	
	`percent = (df.isnull().sum()/df.isnull().count()*100).sort_values(ascending = False)`
	
	`missing_application_train_data = pd.concat([total, percent], axis=1, keys=['Total', 'Percent'])`
	
	`missing_application_train_data.query('Percent > 50')`

Imputer Class Median	`from sklearn.preprocessing import Imputer`
	`imputer = Imputer(strategy="median", verbose= True)`
	`imputer.fit(df[num_vars])`
	`X = imputer.transform(df[num_vars].copy())`
	`temp = pd.DataFrame(X, columns=num_vars)`

Filling Missing Categorical Attributes with Special Value	`temp = df[cat_vars].fillna(999,inplace=False)`

Encode Dummy Variables	*Option 1:*
	`pd.get_dummies(df[cat_vars], dummy_na=True) : Option 1`
	
	Option 2:
	`encoder = LabelBinarizer()`
	`temp_cat_1hot = encoder.fit_transform(df['FLAG_DOCUMENT_4'])`

Data Distributions	`df[num_vars].hist(figsize=(40,30))` `plt.show()`
Descriptive Statistics	`df[num_vars].describe()`
Visualize Correlations	`plt.figure(figsize = (20,20))` `ax = sns.heatmap(df[num_vars].corr(), square= True , linewidths=.5, cmap="Blues")`
Feature Selection	
Outliers	
Conversation Cafe	Build a Checklist for Exploratory Data Analysis Scaling the Analysis: How would you organize your analysis - extending it to include the other files in the data set? Class Imbalance: In debt collection, data is often imbalance. How would you handle the imbalanced data? Missing Data: How do you decide on an appropriate Strategy to impute missing values for Numerical and Categorical data?