gspread

connect to gspread api_json file

gc = gspread.service_account(filename = 'hangvtkfd_api.json')

connect to googlesheet file

'sh = gc.open_by_url('https://docs.google.com/spreadsheets/d/1d-dDVYIh2isksM7PLvLEg_Rza6OqnYvgt1iEdanY8ODw/edit#gid=20-82271837')

update dataframe to a specific google sheet

df.fillna(fill_values, inplace=True)

values = [df.columns.tolist()] + df.values.tolist()

raw = sh.worksheet("raw")

raw.clear()

raw.update(values,1) `insert data at the first row`

append dataframe to a specific googlesheet

values = df_new.values.tolist()

sh.values_append("collection_pfm_metrics_raw", {'valueInputOption': 'USER_ENTERED'}, {'values': values})

load data from googlesheet to dataframe

sh = gc.open_by_url('https://docs.google.com/spreadsheets/d/1H-hT2eENDaffNKz5vZVKcpZR0LZJbMagg0AyCOKuMIsw/edit#gid=0')

raw = sh.worksheet("raw")

values = raw.get_all_values()

product_group = pd.DataFrame(values[1:], columns=values[0])

handling data before updating to googlesheet

`fill  na`

df.fillna(fill_values, inplace=True)

`convert date column to str format`

df_new['week'] = df_new['week'].apply(lambda x: x.strftime("%Y-%m-%d"))

pandas

fill na, missing values

fill_values = {'A': 0, 'B': -1} # Specify the columns and their fill values

df.fillna(fill_values, inplace=True)

pivot dataframe

pivoted_df = df.pivot(index='MERCHANT_NAME', columns='WEEK', values='TOTAL_BOOKING').fillna(0)

define function for multiple aggregations within dataframe

def category_quantity_calculation(df, cl):

def DISBURSE_AMOUNT(df):

return df['DISBURSE_AMOUNT'].sum()

ranking_functions = { 'DISBURSE_AMOUNT': DISBURSE_-AMOUNT}

return_df = df.groupby(cl).apply(lambda group: pd.Series({key: func(group) for key, func in ranking_functions})).reset_index()

return return_df

filter rows that contain any na values

df_filtered = df[df.isna().any(axis=1)]

filter rows that contain na value in a specific column

rows_with_null = df[df['COLUMN_NAME'].isnull()]

rows_with_null = df[df['COLUMN_NAME'].isna()]

filter rows that contain len of specific string in column < 1

def filter_columns_by_length(col):

return all(len(str(cell)) < 1 for cell in col)

filtered_columns = df.loc[:, df.apply(filter_columns_by_length)]

Xử lý ngày tháng trong Python

Convert the 'date_string' column to a date

datetime column df['date_column'] = pd.to_datetime(df['date_string'])