Student Assistant UC Merced

Coding Examples from time as Student Assistant for the GIS Center at UC Merced.

Automated File Cleaning

Examples of Data Processing and Cleaning. Mixture of improved and non-improved examples. Output data not shown for privacy purposes.

In [2]:
```
import pandas as pd
import csv
import xlsxwriter
import os
```

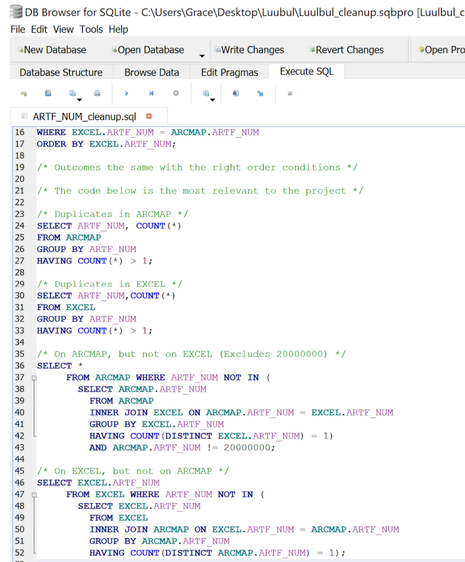
In [ ]:
```
def processData(read, create):
    df = pd.read_csv(read)
    pro = df.loc[(
        df['clientId'] == 'arcgispro'),
        ['clientId', 'id', 'actor', 'actorFullName', 'created_utc']].drop_duplicates(
        ['actorFullName'])
    desktop = df.loc[
        (df['clientId'] == 'arcgisprodesktop'),
        ['clientId', 'id', 'actor', 'actorFullName', 'created_utc']].drop_duplicates(
        ['actorFullName'])

    # self.list.append(create)

    writer = pd.ExcelWriter(create)
    pro.to_excel(writer, 'pro')
    desktop.to_excel(writer, 'desktop')
    writer.save()


def mergeData(self):
    df_total = pd.DataFrame()
    for file in self:
        if file.endswith('.xlsx'):
            excel_file = pd.ExcelFile(file)
            sheets = excel_file.sheet_names
            for sheet in sheets:  # loop through sheets inside an Excel file
                df = excel_file.parse(sheet_name=sheet)
                df_total = df_total.append(df)
        df_total.to_excel('ArcProOrganization.xlsx')


# Month Year
# processData('fileName.csv', 'fileName.xlsx')

rawData = ['OrganizationActivityMonthly_2020-01.csv',
           'OrganizationActivityMonthly_2020-04.csv',
           'OrganizationActivityMonthly_2020-07.csv',
           'OrganizationActivityMonthly_2020-10.csv',
           'OrganizationActivityMonthly_2021-01.csv']

xlsxMonth = []

for month in rawData:
    createMonth = str(month).replace('csv', 'xlsx')
    xlsxMonth.append(createMonth)
    processData(month, createMonth)
    # org.mergeData()
mergeData(xlsxMonth)
```

In [ ]:
```
all_files = ["Spring2021_W1_GISEssentials.csv", "Spring2021_W2_StoryMaps.csv", "Spring2021_W3_DesktopGIS
             "Spring2021_W4_PythonGIS.csv", "Spring2021_W5_WebMapping.csv", "Spring2021_W6_OpenSourceGIS
combined_csv = pd.concat([pd.read_csv(f,header=None) for f in all_files])
combined_csv.head()
combined_csv.to_csv("s21Merge.csv", quotechar='"', quoting=csv.QUOTE_ALL, index=False, encoding='utf-8')

df = pd.read_csv("s21Merge.csv")
s21merged = df # .drop_duplicates(["Email (@ucmerced.edu)"])
writer = pd.ExcelWriter(str("s21MergeDuplicates.xlsx"))
s21merged.to_excel(writer, 'Merge w/ Duplicates')
writer.save()
```

In [ ]:
```
# improved example
def workshop_read(csv_list):
    for i_file, csv_file in enumerate(csv_list):
        ws = pd.read_csv(csv_file)
        # email_column = ws["Email"].str.replace("@ucmerced.edu", "").tolist()
        string = str(ws["Email"]).replace("@ucmerced.edu", " ").split()
        email_col = string[1::2]
        for row_num, data in enumerate(email_col):
            worksheet.write(row_num, len(i_file), data)

workbook = xlsxwriter.Workbook('Fall2020_users.xlsx')
worksheet = workbook.add_worksheet()

path = os.listdir("/path-to-dir")
files = list(filter(lambda f: f.endswith('.csv'), path))

workshop_read(path)

workbook.close()
```

Luulbul Data Clean up

Artifact data extracted from ArcMap and compared to an identifier excel spreadsheet. The code compares the datasets and removes non-matching, non-unique, and duplicate entries.




```
# SQL code adjusted for R

library("DBI")
library("RSQLite")
library("sqldf")
install.packages("DBI")
install.packages("RSQLite")
install.packages("sqldf")

# Outline of the steps I took

# 1  Create csv file (using excel, save as csv)

# 2  Input data from excel reference sheet.
#    Name the column ARCMAP
#    (The name may be different as long as the code has corresponding changes)

# 3  (From Arcmap) Select attribute table. Save as .txt file.
#    Input and convert into existing .csv file.
#    (alternative approaches to this acceptable as long as end result same)
#    Name the column EXCEL
#    Column entries do not have to match

# 4  Save file under desired file name
#    Upload file to database (actions below)

# creates database
db <- dbConnect(RSQLite::SQLite(),"my_db.sqlite")
# checks directory
getwd()

# creates artifacts table
# When applicable, change file name to desired file name
# artifacts variable may change as long as corresponding code changes
artifacts <- read.csv('~/Luulbul_ARTIFACT_ID.csv')
dbWriteTable(db, "artifacts", artifacts)

# use to overwrite table if necessary
dbRemoveTable(db, "artifacts", artifacts)

# returns duplicates IP
duplicates<-dbGetQuery(db, 'SELECT ARCMAP, EXCEL, COUNT(*) FROM artifacts
GROUP BY ARCMAP, EXCEL HAVING COUNT(*) > 1')

# entries in excel, not in arcmap IP
unique_xl_entries<-dbGetQuery(db, 'SELECT EXCEL FROM artifacts WHERE ARCMAP != EXCEL')

# entries in arcmap, not excel IP
unique_arcmap_entries<-dbGetQuery(db, 'SELECT ARCMAP FROM artifacts WHERE ARCMAP != EXCEL')

# table of result
dbGetQuery(db, 'SELECT DISTINCT ARTF_NUM_ARCMAP FROM artifacts
           WHERE ARTF_NUM_EXCEL IS NULL')

dbCreateTable(db, 'new_table AS (SELECT column_1, column2, ... column_n
    FROM old_table) ' )

# excess code/notes:

# chooses arc map entries without excel entry
dbGetQuery(db, 'SELECT DISTINCT ARTF_NUM_ARCMAP FROM artifacts
           WHERE ARTF_NUM_EXCEL IS NULL')

# removes duplicates (ARTF_NUM_ARCMAP)
dbGetQuery(db, 'SELECT DISTINCT ARTF_NUM_ARCMAP FROM artifacts')

# selects ARTF_NUM with matching rows (NOT matching values)
dbGetQuery(db, 'SELECT DISTINCT ARCMAP, EXCEL FROM artifacts
           WHERE ARCMAP = EXCEL')

dbGetQuery(db,'CREATE TABLE artifacts_results (
  Excel int,
  ArcMap int,
  Duplicates int
)')

dbReadTable(db, artifacts_results)
```