Lösungsvorschlag Punkte 2 und 3

Das Skript ermittelt die Anzahl Records im csv und schreibt für jede Record-Art ein CSV-File:

Fragen zu beantworten

1. Wie viele Records sind im CSV enthalten?
2. Erstellen Sie aus dem einzelnen csv je ein csv pro Record-Art. (=9 CSVs)

Lösungsvorschlag:

```
"""
File: read_swisspost_csv.py
Author: Thomas Jäggi
Date: 25.09.2024
Description:
Read the official Swiss-Post csv with street information and create for each table (REC_ART) a seperate csv-File with the name of the table.
Table Names are defined by the column REC_ART and are described in "Anleitung Strassenverzeichnisse.p
"""

import pandas as pd

# Input-File is a ANSI-File, Windows 1252 encoded, separated with Semicolon
csv_file = "Post_Adressdaten20240

# Delimiter
delimiter = ';'

# Since the csv-File does not have the same number of columns for each row, we have to
# determine the maximum number of columns  and include them in the dataframe as names.
# The name will be a number from 0 to max_columns-1.
largest_column_count = 0

# Loop the data lines and determine the maximum number of columns per row. Name the column with a number from 0 to largest_column_count
with open(csv_file, 'r', encoding="cp1252") as temp_f:
    # get No of columns in each line
    col_count = [ len(row.split(delimiter)) for row in temp_f.readlines() ]

### Generate column names  (names will be 0, 1, 2, ..., maximum columns - 1)
column_names = [i for i in range(0, max(col_count))]


# Read csv into dataframe df and print. Set Encoding to cp1252 for windows.
df = pd.read_csv(csv_file, header=None, delimiter=delimiter, names=column_names,
encoding="cp1252")
print('First 20 rows in csv')
print(df.head(20))
print('Last 20 rows in csv')
print(df.tail(20))
print('Number of lines in csv-File:')
print(df.shape[0])

# Generate for each REC_ART one csv-File. groupby([0]): Group by REC_ART which is column 0
for (rec_art), group in df.groupby([0]):

     if rec_art == (0,):
         print('Writing csv NEW_HEA ..')
         tablename = 'NEW_HEA'
     if rec_art == (1,):
         print('Writing csv NEW_PLZ ..')
         tablename = 'NEW_PLZ1'
     if rec_art == (2,):
         print('Writing csv NEW_PLZ2 ..')
         tablename = 'NEW_PLZ2'
     if rec_art == (3,):
         print('Writing csv NEW_COM ..')
         tablename = 'NEW_COM'
     if rec_art == (4,):
         print('Writing csv NEW_STR ..')
         tablename = 'NEW_STR'
     if rec_art == (5,):
         print('Writing csv NEW_STRA ..')
         tablename = 'NEW_STRA'
     if rec_art == (6,):
         print('Writing csv NEW_GEB ..')
         tablename = 'NEW_GEB'
     if rec_art == (7,):
         print('Writing csv NEW_GEBA ..')
         tablename = 'NEW_GEBA'
     if rec_art == (8,):
         print('Writing csv NEW_BOT_B ..')
         tablename = 'NEW_BOT_B'
     if rec_art == (12,):
         print('Writing csv NEW_GEB_COM ..')
         tablename = 'NEW_GEB_COM'
     if rec_art == (10,):
         print('Writing csv NEW_GEO ..')
         tablename = 'NEW_GEO'
     if rec_art == (11,):
         print('Writing csv NEW_HH ..')
         tablename = 'NEW_HH'

     # Using groupby() method of Pandas we can create multiple CSV files. To create a file we
can use the to_csv() method of  Pandas.
     group.to_csv(f'{tablename}.csv', index=False, encoding="cp1252", sep=delimiter)
```

Lösungsvorschlag Punkt 4

Fragen zu beantworten

1. Beantworten Sie mit dem erstellten CSV, welches REC_ART=1 (=NEW_PLZ1) enthält, folgende Fragen:
    a. Wie viele verschiedene Postleitzahlen gibt es in der Schweiz und im Fürstentum Lichtenstein?
    b. Wie viele Gemeinden tragen die Postleitzahl 4566, 4556 und 1580 und in welchen Kantonen befinden sich diese?
    c. Wie viele Gemeinden haben dieselbe PLZ wie ihre Wohngemeinde?

Lösungsvorschlag:

```python
"""
File: abfragen1file.py
Author: Thomas Jäggi
Date: 25.09.2024
Description: Perform some queries on the PLZ-Datafile.
"""

import pandas as pd

# Input-File is a ANSI-File, Windows 1252 encoded, separated with Semicolon
csv_file = "NEW_PLZ1.csv"

# Delimiter
delimiter = ';'

df = pd.read_csv(csv_file, delimiter=delimiter, encoding="cp1252")
# Print PLZ (Column 4), Ort (Column 7) und Kanton (Column 9)
print(df[['4', '7', '9']])

# Number of unique PLZ's in Switzerland
n = len(pd.unique(df['4']))
print("Nr of unique PLZ in Switzerland:", n)



# Number of Ortschaften with PLZ 3000
# To count the number of rows that satisfy the condition, you should use first df[] to filter the rows and
# then use the len() to count the rows after the rows are filtered with the condition.
# You need to select  column 4 in DataFrame to check if any value of the '4' column is equal to PLZ.
# When condition matched, len() function counts the number of rows that contain it.
nr1 = len(df[df['4']==3000])
print("No of recors with PLZ 3000:",nr1)

# Number of Records with PLZ 1000
nr2 = len(df[df['4']==1000])
print("No of recors with PLZ 1000:",nr2)

# Number of Records with PLZ 4000
nr3 = len(df[df['4']==4000])
print("No of recors with PLZ 4000:",nr3)

# Number of Records with PLZ 2500
```

```
nr4 = len(df[df['4']==2500])
print("No of recors with PLZ 2500:",nr4)

# Number of Records with PLZ 4556
nr4 = len(df[df['4']==4556])
print("No of recors with PLZ 4556:",nr4)

# Welche Ortsbezeichnungen gehören zu PLZ 4566
# Remember: df[['4', '7', '9']]: This line prints a subset of the DataFrame that includes only the column with name
# '4' (PLZ) and '7' (Ort) and '9' (Kanton)
df2=(df[df['4']==4566])
print('Gemeinden 4566:\n',df2[['4', '7', '9']])

# Welche Ortsbezeichnungen gehören zu PLZ 1580
df3=(df[df['4']==1580])
print('Gemeinden 1580:\n',df3[['4', '7', '9']])

# Welche Ortsbezeichnungen gehören zu PLZ 3000
df3=(df[df['4']==3000])
print('Gemeinden 3000:\n',df3[['4', '7', '9']])

# Welche Ortsbezeichnungen gehören zu PLZ 2500
df3=(df[df['4']==2500])
print('Gemeinden 2500:\n',df3[['4', '7', '9']])


# Welche Ortsbezeichnungen gehören zu PLZ 4556
df3=(df[df['4']==4556])
print('Gemeinden 4556:\n',df3[['4', '7', '9']])
```

Lösungsvorschlag Punkt 5

Fragen zu beantworten

1. Unter Einbezug des CSV-Files mit Record_Art=4 (NEW_STR) können Sie auch folgende Fragen beantworten:
    a. In welchen Gemeinden mit Kanton existiert dieselbe Strasse ihrer Wohnadresse?
    b. In welchen Gemeinden gibt es ebenfalls eine "Quellgasse"?

Lösungsvorschlag:

```
""
Fi
ab
Au
Th
Jä
Da
25
De
Pe
so
qu
on
th
PL
an
ST
Da
An
th
qu
In
ho
ma
vi
ex
st
=
Qu
""

im
pa
as
pd

#
In
Fi
is
a
AN
Fi
Wi
12
en
se
wi
Se
Ha
be
wr
in
th
pr
ta
cs
=
"N
cs
=
"N
```

```
# Delimiter
delimiter = ';'

#Now we will import the records with the column names specified in "Anleitung
Strassenverzeichnisse.pdf", page 8 (NEW_PLZ1) and 11 (NEW_STR). Both csv's have 16 columns, so
we name the additional columns just with numbers.
#in order to be consistent with the number of columns in the source file. The values will be
NaN-Values, since they are empty.
colnamesPLZ1=['REC_ART', 'ONRP', 'BFSNR', 'PLZ_TYP', 'POSTLEITZAHL', 'PLZ_ZZ', 'GPLZ',
'ORTBEZ18', 'ORTBEZ27', 'KANTON', 'SPRACHCODE', 'SPRACHCODE_ABW', 'BRIEFZ_DURCH',
'GILT_AB_DAT', 'PLZ_BRIEFZUST', 'PLZ_COFF']
colnamesSTR=
['REC_ART','STRID','ONRP','STRBEZK','STRBEZL','STRBEZ2K','STRBEZ2L','STR_LOK_TYP','STRBEZ_SPC','ST


#Read both csv-Files into a dataframe with the defined column names above.
dfPLZ1 = pd.read_csv(csv_file1, delimiter=delimiter, encoding="cp1252", names=colnamesPLZ1)
dfSTR = pd.read_csv(csv_file2, delimiter=delimiter, encoding="cp1252", names=colnamesSTR)

# Some checks to be sure there is data..

#print('First 20 rows in PLZ1-csv')
#print(dfPLZ1.head(20))

#print('Last 20 rows in PLZ1-csv')
#print(dfPLZ1.tail(20))

#print('Number of lines in PLZ1-csv-File:')
#print(dfPLZ1.shape[0])

#print('First 20 rows in STR-csv')
#print(dfSTR.head(20))

#print('Last 20 rows in STR-csv')
#print(dfSTR.tail(20))

#print('Number of lines in STR-csv-File:')
#print(dfSTR.shape[0])

# Merge dataframes with column name "ONRP". ONRP is the "Link" between the 2 files.
df = pd.merge(dfSTR, dfPLZ1, on='ONRP')
print(df)


# Query: In how many villages exists the street X
qresult = df.query("STRBEZ2L == 'Chemin des Prés'")
print(qresult)

qresult = df.query("STRBEZ2L == 'Höheweg'")
print(qresult)

qresult = df.query("STRBEZ2L == 'Quellgasse'")
print(qresult)

qresult = df.query("STRBEZ2L == 'Ringstrasse'")
print(qresult)
```