Universidad del Desarrollo
Facultad de Ingeniería

Taller de Programación

Tarea 2

25 de Noviembre de 2020

Instrucciones Generales

Resuelva con un compañero/a, la siguiente lista de ejercicios. Regístrese en uno de los grupos correspondientes a la evaluación `Tarea2` disponibles en la sección `Personas` de canvas. Su código debe estar contenido en un único archivo `t2_apellido1_apellido2.py`. Este y los archivos utilizados o creados en la tarea deben ser entregados en un archivo comprimido `.zip` con el formato `t2_apellido1_apellido2.zip` en canvas.

Detalles de la Entrega

- La fecha de entrega es el 25 de Noviembre del 2020 a las 23:59 hrs.
- Si su código tiene errores de sintaxis (no se ejecuta), su tarea no se considerará entregada.
- Sus soluciones pueden incluir todas funciones adicionales que usted estime necesarias
- Si utiliza métodos de librerías no vistas en clases, comente brevemente su funcionalidad y agregue un link a la documentación oficial.
- Recuerde que solo puede comentar sus soluciones con su profesor, los ayudantes y su compañero(a) de equipo.
- Su solución, junto con las soluciones de todas las secciones, será evaluada por un software detector de plagio. De detectarse copia, su certamen será evaluado con la **NOTA MÍNIMA** y la situación será informada al comité de ética de la Facultad, lo que puede derivar en una causal de eliminación de la universidad.

Contexto

Ustedes han sido contratados como analistas de datos para el Instituto de Ciencia de los datos para analizar los resultados del último plebiscito en Chile. Se les solicita programar algunas funciones y elaborar indicadores específicos para estudiar con más profundidad los resultados.

Archivos

Los archivos `resultados_comuna.csv` y `resultados_mesa.csv` contienen los resultados a nivel comunal y a nivel de mesa para el plebiscito celebrado en octubre de 2020. El archivo `resultados_participacion_17_20.csv` por otro lado, contiene los resultados de la primera vuelta de las elecciones presidenciales del año 2017. Con estos archivos y los aprendido en clases, resuelva los problemas descritos en la sección **Problemas**.

Problemas

1. Participación en el plebiscito a nivel de comuna [0.5 pts.] Programe la función `participacion_comuna(file)` que recibe el archivo `resultados_comuna.csv` y retorna una lista de tuplas con las diez comunas con más participación electoral en el plebiscito y el porcentaje de participación para cada una. Esto se calcula dividiendo el total de votos realizados por el total de votantes habilitados.

```
import operator
def participacion_comuna(file):
    participacion=list()
    with open("resultados_comuna.csv", "r") as f:
        next(f)
        for line in f:
            s=line.split(';')
            p=int(s[11])/int(s[10])*100
            participacion.append((s[0],p))
    a=sorted(participacion, reverse=True, key=operator.itemgetter(1))

    return a[:10]
```

```
participacion_comuna('plebiscito.csv')

[('VITACURA', 67.9817888817517),
 ('LO BARNECHEA', 67.91583190963729),
 ('LA REINA', 64.55665830515869),
 ('ÑUÑOA', 64.06075736956811),
 ('CONCON', 63.89602222105759),
 ('HUECHURABA', 63.14118423222826),
 ('MAIPU', 62.37528715717604),
 ('LAS CONDES', 62.146523565095656),
 ('PAPUDO', 61.98830409356725),
 ('ZAPALLAR', 61.904161412358135)]
```

2. Plebiscito en la Región Metropolitana [0.5 pts.] Programe la función `resultados_comuna(file)` que recibe el archivo `resultados_comuna.csv` , y que retorna un archivo `resultados_comuna_rm.csv` con los datos del plebiscito y el porcentaje de participación solo para la región Metropolitana. Considere que todas las comunas cuyo código de referencia comienzan con 13 pertenecen a la Region Metropolitana.

```
def resultados_comuna_rm(file):
    with open(file, "r") as f:
        with open("resultados_comuna_rm.csv", 'w') as g:
            for line in f:
                s=line.split(';')
                if s[0]=='Comuna'.strip():
                    header=s.copy()
                    header_new= [e.replace('\n', '') for e in header]
                    header_new.append('Participacion')
                    print(header_new)
                    g.write(';'.join(header_new) + '\n')
                elif s[1].startswith('13')==True:
                    p=str(round(int(s[-1])/int(s[-2])*100,2))
                    lines_rm=s.copy()
                    lines_rm.append(p)
                    lines_rm_new = [e.replace('\n', '') for e in lines_rm]
                    g.write(';'.join(lines_rm_new) + '\n')   ```
```

3. Participación a nivel país [0.5 pts.] Programe la función `participacion_pais(file)` que reciba el archivo `resultados_comuna.csv` y retorne el porcentaje de participación a nivel país.

```python
def participacion_pais(file):
    votos=list()
    votantes=list()
    with open(file, "r") as f:
        next(f)
        for line in f:
            s=line.split(';')
            votos.append(int(s[11]))
            votantes.append(int(s[10]))
    return round((sum(votos)/sum(votantes))*100,2)
```

```python
participacion_pais('resultados_comuna.csv')
50.9
```

4. Apruebo o Rechazo [1.0 pts.] Programe la función `apruebo_rechazo(file, opcion)` que recibe el archivo `resultados_comuna.csv` y el string `apruebo` o `rechazo` y retorna una lista de comunas en la que ganó la opción `apruebo` o `rechazo` según corresponda.

```python
def apruebo_rechazo(file,opcion):
    comunas_apruebo=list()
    comunas_rechazo=list()
    with open(file, "r") as f:
            next(f)
            for line in f:
                s=line.split(';')
                if int(s[2])>int(s[3]):
                    comunas_apruebo.append(s[0])
                else:
                    comunas_rechazo.append(s[0])
            if opcion=='apruebo':
                return comunas_apruebo
            else:
                return comunas_rechazo
```

```python
apruebo_rechazo('resultados_comuna_rm.csv','apruebo')
['LAS CONDES', 'LO BARNECHEA', 'VITACURA']
```

5. Comparación Participación [1.0 pts.] Programe la función `participación_17_20(file1, file2)` que recibe al archivo

`resultados_comuna.csv` y el archivo `resultados_presidenciales_2017.csv` y retorna el archivo `resultados_participacion_17_20.csv` con la comuna, la fracción de participación para en el plebiscito del 2020, la fracción de participación para las elecciones presidenciales del 2017 en primera vuelta y la diferencia entre la participación electoral entre ambas votaciones.

```
def participacion_17_20(file1, file2):
    r_2017 = []
    r_2020 = []

    comunas = []
    diff = []

    with open(file1, 'r') as f:
        next(f)
        with open(file2, 'r') as g:
            next(g)
            for line in f:
                datos_2020 = line.split(';')

                p=int(datos_2020[11])/int(datos_2020[10])
                r_2020.append(round(p,2))
                comunas.append(datos_2020[0])

            for line in g:
                datos_2017 = line.split(';')
                r_2017.append(round(float(datos_2017[2]),2))


            with open('resultados_participacion_17_20.csv', 'w') as g:
                g.write('comuna;part_2020;part_2017; diff')
                g.write('\n')
                for i in range(len(comunas)):
                    data_escritura = [comunas[i], str(r_2020[i]), str(r_2017[i]),
                    g.write(';'.join(data_escritura))
                    g.write('\n')
```

6. Proporción de mujeres por colegio y circunscripción [1.0 pts.] Programe la función `participacion_mujeres(file, circunscripción, colegio)` que reciba el archivo `resultados_mesa.csv`, una circunscripción electoral y el nombre de un colegio, y retorne el promedio de participación de las mujeres para de todas las mesas.

Solución parcialmente correcta (no es del correcto promediar porcentajes):

```
def participacion_mujeres(file, circunscripcion, colegio):
    with open(file, 'r') as f:
        next(f)
        part_mujeres=[]
        for line in f:
            s = line.split(';')
            if s[3] == circunscripcion and s[4]==colegio:
                part_mujeres.append(float(s[11]))
        prom = sum(part_mujeres) / len(part_mujeres)
        print(circunscripcion, colegio, round(prom*100,2), '%')
```

Solución correcta:

```
def participacion_mujeres(file, circunscripcion, colegio):
    with open(file, 'r') as f:
        next(f)
        votantes_tot=list()
        votantes_m_tot=list()
        for line in f:
            s = line.split(';')
            if s[3] == circunscripcion and s[4]==colegio:
                votantes=float(s[6])
                prop_m=float(s[11])
                votantes_m=votantes*prop_m
                votantes_tot.append(votantes)
                votantes_m_tot.append(votantes_m)
        prom = round(sum(votantes_m_tot)/sum(votantes_tot)*100,2))


     return (circunscripcion, colegio,prom)
```

7. Proporción de mujeres a nivel de comuna [1.0 pts.] Programe un código que permita crear el archivo `participación_mujeres_comuna.csv`, el cual debe contener el listado de todas las regiones y comunas con su nivel promedio de proporción de mujeres en el plebiscito. El archivo debe tener la forma `region;comuna;porcentaje_mujeres`.

Solución parcialmente correcta (no es del todo correcto promediar porcentajes):

```python
def participacion_mujeres_comuna(file):
    with open(file, 'r') as f:

        next(f)
        regiones = []
        for line in f:
            s = line.split(';')
            regiones.append(s[2])
            regiones_set = set(regiones)


    with open('participacion_mujeres_comuna.csv', 'w') as j:
        j.write('region;comuna;porcentaje_mujeres')
        j.write('\n')
        for region in regiones_set:
            comunas_in_region = []

            with open(file, 'r') as g:
                next(g)
                for lines in g:
                    s2 = lines.split(';')
                    if s2[2] == region:
                        comunas_in_region.append(s2[3])

                for comuna in set(comunas_in_region):
                    with open(file, 'r') as h:
                        next(h)
                        valores_part = []
                        for lines in h:
                            s3 = lines.split(';')
                            if s3[3] == comuna:
                                valores_part.append(float(s3[-1]))
                        if sum(valores_part) >= 0 and len(valores_part) > 0:
                            prom = sum(valores_part) / len(valores_part)
                        else:
                            prom = 0
                        j.write(';'.join([region, comuna, str(prom)]))
                        j.write('\n')
```

Solución correcta:

```python
def participacion_mujeres_comuna(file):
    file_dicc=dict()
    with open(file, 'r') as f:#abro archivo
        next(f)
        for line in f:
            s = line.split(';')
            votantes=float(s[6])#votantes
            prop_m=float(s[11])#proporcion mujeres que votaron
            votantes_m=votantes*prop_m#prop mujeres que votaron
            region=s[1]

            #agrupo utilizando un dicc
            #para una region, si la comuna esta en el diccionario,
            #sumo los votantes y #las mujeres votantes al valor actual
            #si no está entonces los guardo como valor

            if s[2] in file_dicc:
                votantes_m= file_dicc[s[2]][1] + votantes_m
                votantes   = file_dicc[s[2]][2] + votantes
                file_dicc[s[2]]=(region,votantes_m,votantes)
            else:
                file_dicc[s[2]]=(region,votantes_m, votantes)

    #escribo el archivo
    with open('participacion_mujeres_comuna.csv', 'w') as t:
        t.write('region;comuna;porcentaje_mujeres\n')
        for k,v in file_dicc.items():
            #creo una lista que me ayudará a escribir una linea
            linea=[v[0],k,str(round(100*v[1]/v[2],2))]
            t.write(';'.join(linea))#transformo la lista en una linea y escribo
            t.write('\n')
```