

Finanziamenti UE - Cordis & SharePoint management

May 31, 2022

1 Esempio di lettura .csv in Python da SharePoint e salvataggio in file .csv in cartella SharePoint.

Il seguente script descrive alcuni passaggi comuni di gestione di file .csv: dalla lettura da Sharepoint, alla pulizia del dato tramite funzioni della libreria pandas, al salvataggio in cartelle Sharepoint differenti dall'originale.

I file .csv utilizzati sono stati scaricati da Cordis - EU research projects under Horizon 2020 (2014-2020) (<https://data.europa.eu/data/datasets/cordish2020projects?locale=en>).

La cartella .zip contiene i seguenti .csv: - project.csv: informazioni su tutti i progetti UE del programma H2020. - organization.csv: informazioni sulle organizzazioni che hanno partecipato con diversi ruoli ai progetti UE. - legalBasis.csv: titolo e legalBasis dei progetti UE. - topics.csv: argomenti per progetto UE. - euroSciVoc.csv: classificazione sulla base del "Il vocabolario europeo della scienza" (<https://data.europa.eu/data/datasets/euroscivoc-the-european-science-vocabulary?locale=it>).

1.1 Importazione librerie ed accesso a Sharepoint

```
[ ]: import sys
import os
from pathlib import Path
import pandas as pd
import datetime
from pandas import ExcelFile
import numpy as np
import io
import sharepy
import getpass
import csv
pd.options.display.max_columns = None
```

```
[ ]: # Autenticazione a Sharepoint
URL = 'https://areasciencepark.sharepoint.com'
#SHAREPOINT_USER = 'andrea.bincoletto@areasciencepark.it'
SHAREPOINT_USER = input("Inserire l'email aziendale: ")
SHAREPOINT_PASSWORD = getpass.getpass('Inserire la password: ')
s = sharepy.connect(URL, username=SHAREPOINT_USER, password=SHAREPOINT_PASSWORD)
```

1.2 Apertura dei .csv contenuti nella cartella “originali”

Projects

```
[ ]: # aprire il CVS ("project.csv") in un dataframe
SHP_URL = 'https://areasciencepark.sharepoint.com/sites/DataRepository/
↳Documenti condivisi/General/'
SHP_DIR = 'dataset_imprese_fvg/test storicizzazione/originali/'
SHP_FILE = 'project.csv'

r = s.get(SHP_URL + SHP_DIR + SHP_FILE)
f = io.BytesIO(r.content)
df_project = pd.read_csv(f, sep=";", encoding='utf-8-sig', dtype = str)
print(df_project.shape)
assert df_project.shape[1] == 20 # controlla che la dimensione sia quella
↳prevista
print(f'Dimensioni del dataframe: {df_project.shape}. Sample: ')
df_project.sample(5)
```

(35378, 20)

Dimensioni del dataframe: (35378, 20). Sample:

```
[ ]:
```

	id	acronym	status	\
14133	680153	DMAP	CLOSED	
856	950584	DecOmPress	SIGNED	
33647	888726	uHeart	CLOSED	
26388	757038	SIMPLE	SIGNED	
3881	809520	SolidCool	CLOSED	

	title	startDate	\
14133	Data Mining Algorithms in Practice	2016-02-01	
856	Decoding spatio-temporal omics in progressive ...	2021-01-01	
33647	A beating Heart-on-Chip for pre-clinical early...	2019-12-01	
26388	Spacefibre IMPLementation design test Equipmen...	2017-02-01	
3881	Cost-efficient, solid-state refrigeration tech...	2018-02-01	

	endDate	totalCost	ecMaxContribution	legalBasis	\
14133	2021-01-31	1137500	1137500	NaN	
856	2025-12-31	1500000	1500000	NaN	
33647	2020-02-29	71429	50000	NaN	
26388	2019-01-31	912930	639051	NaN	
3881	2018-05-31	71429	50000	H2020-EU.3.3.	

	topics	ecSignatureDate	frameworkProgramme	\
14133	ERC-StG-2015	NaN	H2020	
856	ERC-2020-STG	NaN	H2020	
33647	EIC-SMEInst-2018-2020	NaN	H2020	
26388	SMEInst-04-2016-2017	NaN	H2020	

3881	SMEInst-09-2016-2017	2018-01-31	H2020
------	----------------------	------------	-------

	masterCall	subCall	fundingScheme \
14133	NaN	ERC-2015-STG	ERC-STG
856	NaN	ERC-2020-STG	ERC-STG
33647	NaN	H2020-SMEInst-2018-2020-1	SME-1
26388	NaN	H2020-SMEINST-2-2016-2017	SME-2
3881	H2020-SMEInst-2016-2017	H2020-SMEINST-1-2016-2017	SME-1

	nature	objective \
14133	NaN	Data Mining algorithms are a cornerstone of to...
856	NaN	Multiple sclerosis (MS) is a paradigmatic prog...
33647	NaN	The cost to develop new drugs is huge, mainly ...
26388	NaN	Following the success of SpaceWire, SpaceFibre...
3881	NaN	Vapour compression (VC) technology, used in th...

	contentUpdateDate	rcn	grantDoi
14133	2021-05-31 11:12:06	199559	NaN
856	2020-11-17 17:39:37	232979	NaN
33647	2020-08-07 14:20:28	225930	NaN
26388	2017-01-11 13:54:15	207927	NaN
3881	2022-01-14 17:01:14	213729	NaN

```
[ ]: # copiare il dataframe, per mantenere l'originale se dovesse servire
df_project_elab = df_project.copy(deep=False)
```

```
[ ]: # drop delle colonne del df_project_elab che non servono
df_project_elab = df_project_elab.drop(columns=[
    'legalBasis', 'masterCall', 'grantDoi', 'nature', 'ecSignatureDate', 'contentUpdateDate',
    'ecMaxContribution', 'rcn', 'topics'
])
df_project_elab.shape
```

```
[ ]: (35378, 11)
```

```
[ ]: # rename delle colonne del df_project
df_project_elab = df_project_elab.rename(columns={
    'id': 'projectID',
    'subCall': 'call',
    'title': 'long_title'
})
```

```
[ ]: # UPLOAD df_project_elab in sharepoint
SHP_URL= 'https://areasciencepark.sharepoint.com/Sites/DataRepository/Documenti_
↳condivisi/General/'
SHP_DIR = 'dataset_imprese_fvg/test storicizzazione/elaborati/'
```

```
SHP_FILE = 'projects_elab.csv'

f = df_project_elab.to_csv(sep=';', index=False).encode('utf-8')
#f = df_project_elab.to_csv(sep=';', encoding='utf-8-sig', index=False)
r = s.put(SHP_URL + SHP_DIR + SHP_FILE, f)
print(r.text)
```

Organization

```
[ ]: # Aprire il CSV ("organization.csv") in un dataframe
SHP_URL = 'https://areasciencepark.sharepoint.com/sites/DataRepository/
↳Documenti condivisi/General/'
SHP_DIR = 'dataset_imprese_fvg/test storizzazione/originali/'
SHP_FILE = 'organization.csv'

r = s.get(SHP_URL + SHP_DIR + SHP_FILE)
f = io.BytesIO(r.content)
df_org = pd.read_csv(f, sep=";", dtype = str)
print(df_org.shape)
```

(172730, 25)

```
[ ]: df_org_elab = df_org.copy(deep=False)
```

```
[ ]: # filtro il df per le sole imprese con 'country' = IT
print("Verifica del conteggio delle organizzazioni IT:")
#print(df_org_elab["country"].value_counts().head(7))
print(df_org_elab["country"].value_counts().head())
it_filter = df_org_elab["country"] == "IT"
df_org_elab = df_org_elab.loc[it_filter].copy()
print("Il nuovo dataframe deve avere le stesse righe del conteggio sopra.")
print(f"df_org_elab ora ha dimensioni: {df_org_elab.shape}")
```

Verifica del conteggio delle organizzazioni IT:

```
DE    20249
ES    18161
UK    17073
IT    16523
FR    16339
```

Name: country, dtype: int64

Il nuovo dataframe deve avere le stesse righe del conteggio sopra.

df_org_elab ora ha dimensioni: (16523, 25)

```
[ ]: # drop delle righe di "vatNumber" non valorizzate (vuote)
# NB: si possono salvare le righe senza vatNumber in un altro df!
df_org_elab = df_org_elab[df_org_elab['vatNumber'].notna()]
```

```
df_org_elab.shape
```

```
[ ]: (15703, 25)
```

```
[ ]: # drop delle colonne di df_organization_elab che non servono
df_org_elab = df_org_elab.drop(columns=[
    'nutsCode', 'geolocation', 'order', 'contentUpdateDate',
    'netEcContribution', 'totalCost', 'active', 'SME', 'organizationURL',
    'contactForm', 'contentUpdateDate', 'rcn', 'order',
    'projectAcronym'
])
df_org_elab.shape
```

```
[ ]: (15703, 13)
```

```
[ ]: df_org_elab.columns
```

```
[ ]: Index(['projectID', 'organisationID', 'vatNumber', 'name', 'shortName',
        'activityType', 'street', 'postCode', 'city', 'country', 'role',
        'ecContribution', 'endOfParticipation'],
        dtype='object')
```

```
[ ]: #divido dal "vatNumber" la p.iva e le prime due cifre riferite allo Stato
    ↳d'appartenenza dell'organizzazione
df_org_elab['vatNumber'] = df_org_elab['vatNumber'].astype(str)
# df_org_elab['vatNumber'] = df_org_elab['Country_piva'].str[1]
# df_org_elab['vatNumber'] = df_org_elab['vatNumber'].str[2:]
# df_org_elab[['Country_piva', 'vatNumber']] = df_org_elab['vatNumber'].str.
    ↳split(n=2, expand=True)
df_org_elab["piva"] = df_org_elab["vatNumber"].map(lambda vatNumber:
    ↳str(vatNumber)[2:])
df_org_elab.columns
```

```
[ ]: Index(['projectID', 'organisationID', 'vatNumber', 'name', 'shortName',
        'activityType', 'street', 'postCode', 'city', 'country', 'role',
        'ecContribution', 'endOfParticipation', 'piva'],
        dtype='object')
```

```
[ ]: # count e drop delle righe con una partita iva diversa da 11 caratteri di
    ↳lunghezza
df_org_elab['piva'] = df_org_elab['piva'].astype(str)
#df_org_elab['len_piva'] = df_org_elab['piva'].str.len()
#df_org_elab['len_piva'] = ['']
df_org_elab['len_piva'] = df_org_elab['piva'].apply(lambda x: len(x))
print("Le partite iva presenti hanno le seguenti lunghezze: ")
print(df_org_elab["len_piva"].value_counts().head())
```

Le partite iva presenti hanno le seguenti lunghezze:

```
11    15598
5       51
10      39
12      10
9        4
```

Name: len_piva, dtype: int64

```
[ ]: # drop delle righe con partita iva con lunghezza diversa da 11
df_org_elab = df_org_elab.loc[df_org_elab['len_piva'].isin([11])]
print("Il dataframe deve avere il numero di righe con lunghezza della partita_
    ↪iva pari a 11.")
print(f"{df_org_elab.shape} deve essere uguale al numero del conteggio della_
    ↪cella di output sopra.")
```

Il dataframe deve avere il numero di righe con lunghezza della partita iva pari a 11.

(15598, 15) deve essere uguale al numero del conteggio della cella di output sopra.

```
[ ]: # drop della colonna "vatNumber" e "len_piva", abbiamo "piva" ora
df_org_elab = df_org_elab.drop(columns=[
    'vatNumber', 'len_piva'
])
df_org_elab.columns
```

```
[ ]: Index(['projectID', 'organisationID', 'name', 'shortName', 'activityType',
        'street', 'postCode', 'city', 'country', 'role', 'ecContribution',
        'endOfParticipation', 'piva'],
        dtype='object')
```

```
[ ]: # Filtro dei soli progetti con organizzazioni italiane (basato su projectID)
# due versioni: df_projectID_IT e filter_projectID
df_projectID_IT = df_org_elab.copy(deep=False)
df_projectID_IT = df_projectID_IT.drop(columns={
    'organisationID', 'name', 'shortName', 'activityType',
    'street', 'postCode', 'city', 'country', 'role',
    'ecContribution', 'endOfParticipation', 'piva'
})
filter_projectID = df_org_elab['projectID']
print(f"Il primo filtro ha dimensione: {df_projectID_IT.shape}")
print(f"Il secondo filtro ha dimensione: {filter_projectID.shape}")
```

```
[ ]: projectID
20    787419
39    894921
45    891197
```

```
[ ]: # UPLOAD df_org_elab in sharepoint
SHP_URL= 'https://areasciencepark.sharepoint.com/Sites/DataRepository/Documenti_
↳condivisi/General/'
SHP_DIR = 'dataset_imprese_fvg/test storicizzazione/elaborati/'
SHP_FILE = 'organization_elab.csv'

f = df_org_elab.to_csv(sep=';', index=False).encode('utf-8')
r = s.put(SHP_URL + SHP_DIR + SHP_FILE, f)
print(r.text)
```

Topics

```
[ ]: # Aprire il CSV ("topics.csv") in un dataframe
SHP_URL = 'https://areasciencepark.sharepoint.com/sites/DataRepository/
↳Documenti condivisi/General/'
SHP_DIR = 'dataset_imprese_fvg/test storicizzazione/originali/'
SHP_FILE = 'topics.csv'

r = s.get(SHP_URL + SHP_DIR + SHP_FILE)
f = io.BytesIO(r.content)
df_topics = pd.read_csv(f, sep=";", dtype = str)
print(df_topics.shape)
```

(35378, 3)

```
[ ]: # Copia del df_topics per mantenere l'originale
df_topics_elab = df_topics.copy(deep=False)
```

```
[ ]: # UPLOAD df_topics_elab in sharepoint
SHP_URL= 'https://areasciencepark.sharepoint.com/Sites/DataRepository/Documenti_
↳condivisi/General/'
SHP_DIR = 'dataset_imprese_fvg/test storicizzazione/elaborati/'
SHP_FILE = 'topics_elab.csv'

f = df_topics_elab.to_csv(sep=';', index=False).encode('utf-8')
r = s.put(SHP_URL + SHP_DIR + SHP_FILE, f)
print(r.text)
```

EuroSciVoc

```
[ ]: # Apro il CSV ("euroSciVoc.csv") in un dataframe
SHP_URL = 'https://areasciencepark.sharepoint.com/sites/DataRepository/
↳Documenti condivisi/General/'
SHP_DIR = 'dataset_imprese_fvg/test storicizzazione/originali/'
SHP_FILE = 'euroSciVoc.csv'
```

```

r = s.get(SHP_URL + SHP_DIR + SHP_FILE)
f = io.BytesIO(r.content)
df_esv = pd.read_csv(f, sep=";", dtype = str)
print(df_esv.shape)
print(df_esv.head(5))

```

```

(88930, 4)
  projectID euroSciVocCode euroSciVocPath \
0    750802      /30029 /natural sciences/physical sciences/classical ...
1    750802      /34823946 /natural sciences/physical sciences/classical ...
2    655710      /30048 /social sciences/law/human rights/human rights...
3    701591      /77161814 /natural sciences/biological sciences/genetics...
4    701591      /343 /natural sciences/biological sciences/evolutio...

  euroSciVocTitle
0    solid mechanics
1    microfluidics
2    human rights law
3    viral genomes
4    evolutionary biology

```

```

[ ]: df_esv_elab = df_esv.copy(deep=False)
df_esv_elab.columns

```

```

[ ]: Index(['projectID', 'euroSciVocCode', 'euroSciVocPath', 'euroSciVocTitle'],
dtype='object')

```

```

[ ]: # Per esercizio: uso il groupby per mettere tutti i title in un'unica cella per
    ↳ progetto
    # (che dovrà apparire solo una volta per riga)
df_esv_elab_subset = df_esv_elab[['projectID', 'euroSciVocTitle']]
#df_esv_elab_subset['euroSciVocTitle'] = df_esv_elab_subset['euroSciVocTitle'].
    ↳ astype(str)
df_esv_elab_subset = df_esv_elab_subset.groupby('projectID')['euroSciVocTitle'].
    ↳ apply(';'.join).reset_index()
df_esv_elab_subset = df_esv_elab_subset.rename(columns={
    'euroSciVocTitle': 'esvTitle'
})
print(df_esv_elab_subset.head(10))
print(f"I progetti del dataframe sono: {df_esv_elab_subset.shape}")

```

```

  projectID esvTitle
0  101000020 public policies
1  101000063 didactics
2  101000132 governance
3  101000154 drones;rotorcraft;control systems

```



```

4 101000158 renewable energy;data exchange;internet of thi...
5 101000160 oncology
6 101000162 automation;productivity
7 101000165 deep learning
8 101000169 software;data science
9 101000180 ultrasound

```

I progetti del dataframe sono: (32052, 2)

```

[ ]: # inserisco la nuova colonna nel df_esv_elab
df_esv_elab = pd.merge(df_esv_elab, df_esv_elab_subset, how='left',
↳on='projectID')

```

```

[ ]: df_esv_elab = df_esv_elab.drop(columns=[
    'euroSciVocTitle', 'euroSciVocCode', 'euroSciVocPath'
])

```

```

[ ]: # usare il filtro df_projectID_IT per mantenere solo i progetti italiani (uno
↳dei due creati sul df dell'organizzazione)
# NB: così si perde l'informazione sui diversi temi che lo stesso progetto ha
↳nella colonna euroSciVocPath
#df_esv_elab = pd.merge(df_projectID_IT, df_esv_elab, how='left',
↳on='projectID')
df_esv_elab = df_esv_elab[df_esv_elab['projectID'].isin(filter_projectID)]
# eliminare i doppioni
df_esv_elab = df_esv_elab.drop_duplicates(subset=['projectID'])
print(f"I progetti a cui almeno un'organizzazione italiana ha partecipato e con
↳tema da euroSciVoc sono in totale : {df_esv_elab.shape}")

```

I progetti a cui almeno un'organizzazione italiana ha partecipato e con tema da euroSciVoc sono in totale : (6759, 2)

```

[ ]: # # Creare tante colonne quanti i livelli della colonna "euroSciVocPath"
# df_esv_paths = df_esv_elab['euroSciVocPath'].str.split('/', expand = True).
↳fillna('null')
# print(type(df_esv_paths))
# print(df_esv_paths.head(5))
# # nomi delle colonne di df_esv_paths
# for col in df_esv_paths.columns:
#     print(f"Nome colonna: {col}")

# # splitting 'Name' column into Two columns
# i.e. 'First' and 'Last' respectively and
# # Adding these columns to the existing dataframe.
# df[['First', 'Last']] = df.Name.apply(
#     lambda x: pd.Series(str(x).split("_")))

# # Mantengo solamente la colonna 'path1'

```

```

# df_esv_paths = df_esv_paths.drop(columns=[
#     0, 2, 3, 4, 5, 6, 7
# ], axis=1, inplace=True)

# # Rinominare le colonne dal df_esv_paths
# df_esv_paths = df_esv_paths.rename(columns={
#     1: 'euroSciVocPath1',
#     })
# df_esv_paths.columns

# # Verifico i valori null nelle varie colonne (voglio che tutti i campi siano
#     ↳valorizzati, altrimenti elimino le colonne successive a quella totalmente
#     ↳valorizzata)

# print(df_org_elab["country"].value_counts().head())

# null_value = ['null']
# df_null_path = df_esv_paths.loc[df_esv_paths['path1'].isin(null_value)]
# df_null_path = df_esv_paths[df_esv_paths['path1'].isin(null_value)]
# df_null_path

# # Uniamo i due dataframe
# df_esv_elab = pd.merge(df_esv_elab, df_esv_path, how='left', on='columnmerge')

```

```

[ ]: # UPLOAD df_topics_elab in sharepoint
SHP_URL= 'https://areasciencepark.sharepoint.com/Sites/DataRepository/Documenti
    ↳condivisi/General/'
SHP_DIR = 'dataset_imprese_fvg/test storicizzazione/elaborati/'
SHP_FILE = 'euroSciVoc_elab.csv'

f = df_esv_elab.to_csv(sep=';', index=False).encode('utf-8')
r = s.put(SHP_URL + SHP_DIR + SHP_FILE, f)
print(r.text)

```

1.3 Unione e salvataggio dei .csv elaborati nella cartella “finale”

Salvo tutti i csv della cartella elaborati come un unico file csv nella cartella finale

```

[ ]: # merge projects & topics
df_proj_top = pd.merge(df_project_elab, df_topics_elab, how='left',
    ↳on='projectID').fillna('none')
print(f"Il dataframe sui progetti aveva dimensione: {df_project_elab.shape}")
print(f"Il dataframe sui topics aveva dimensione: {df_topics_elab.shape}")
print(f"Il nuovo dataframe ha dimensioni: {df_proj_top.shape}")
df_proj_top.head(3)

```

Il dataframe sui progetti aveva dimensione: (35378, 11)

Il dataframe sui topics aveva dimensione: (35378, 3)

Il nuovo dataframe ha dimensioni: (35378, 13)

```
[ ]:  projectID      acronym  status  \
0      750802  El_CapiTun  CLOSED
1      655710  CONRICONF  CLOSED
2      701591   EvolVir   CLOSED

                                     long_title  startDate  endDate  \
0  An elastocapillary-enabled self-tunable microf...  2018-01-01  2019-12-31
1  Contentious Rights: A Comparative Study of Int...  2016-01-17  2020-02-08
2  Evolution of virulence in immune-compromised h...  2017-07-01  2019-06-30

    totalCost frameworkProgramme      call  fundingScheme  \
0  175419.6                H2020  H2020-MSCA-IF-2016  MSCA-IF-EF-ST
1  269857.8                H2020  H2020-MSCA-IF-2014    MSCA-IF-GF
2  195454.8                H2020  H2020-MSCA-IF-2015  MSCA-IF-EF-ST

                                     objective      topic  \
0  I am a French citizen with international train...  MSCA-IF-2016
1  Contentious Rights: A Comparative Study of Int...  MSCA-IF-2014-GF
2  Understanding pathogen evolution is key to pre...  MSCA-IF-2015-EF

                                     title
0                                     Individual Fellowships
1  Marie Skłodowska-Curie Individual Fellowships ...
2  Marie Skłodowska-Curie Individual Fellowships ...
```

```
[ ]: # aggiungere le informazioni di euroSciVoc al dataframe precedente
df_proj_top_esv = pd.merge(df_proj_top, df_esv_elab, how='left',
                             on='projectID').fillna('none')
print(f"Il dataframe precedente aveva dimensione: {df_proj_top.shape}")
print(f"Il dataframe su euroSciVoc aveva dimensione: {df_esv_elab.shape}")
print(f"Il nuovo dataframe ha dimensioni: {df_proj_top_esv.shape}")
df_proj_top_esv.head(3)
```

Il dataframe precedente aveva dimensione: (35378, 13)

Il dataframe su euroSciVoc aveva dimensione: (6759, 2)

Il nuovo dataframe ha dimensioni: (35378, 14)

```
[ ]:  projectID      acronym  status  \
0      750802  El_CapiTun  CLOSED
1      655710  CONRICONF  CLOSED
2      701591   EvolVir   CLOSED

                                     long_title  startDate  endDate  \
```

0	An elastocapillary-enabled self-tunable microf...	2018-01-01	2019-12-31
1	Contentious Rights: A Comparative Study of Int...	2016-01-17	2020-02-08
2	Evolution of virulence in immune-compromised h...	2017-07-01	2019-06-30

	totalCost	frameworkProgramme	call	fundingScheme	\
0	175419.6	H2020	H2020-MSCA-IF-2016	MSCA-IF-EF-ST	
1	269857.8	H2020	H2020-MSCA-IF-2014	MSCA-IF-GF	
2	195454.8	H2020	H2020-MSCA-IF-2015	MSCA-IF-EF-ST	

	objective	topic	\
0	I am a French citizen with international train...	MSCA-IF-2016	
1	Contentious Rights: A Comparative Study of Int...	MSCA-IF-2014-GF	
2	Understanding pathogen evolution is key to pre...	MSCA-IF-2015-EF	

	title	esvTitle
0	Individual Fellowships	none
1	Marie Skłodowska-Curie Individual Fellowships ...	none
2	Marie Skłodowska-Curie Individual Fellowships ...	none

```
[ ]: # Verifica dei progetti senza 'esvTitle'
print("Verifica del conteggio di esvTitle non valorizzati (='none'):")
print(df_proj_top_esv["esvTitle"].value_counts().head())
# it_filter = df_org_elab["country"] == "IT"
# df_org_elab = df_org_elab.loc[it_filter].copy()
# print("Il nuovo dataframe deve avere le stesse righe del conteggio sopra.")
# print(f"df_org_elab ora ha dimensioni: {df_org_elab.shape}")
```

Verifica del conteggio di esvTitle non valorizzati (='none'):

```
none                28619
renewable energy     59
ecosystems           41
aircraft             38
transport            27
Name: esvTitle, dtype: int64
```

```
[ ]: # UPLOAD df_proj_top_esv in sharepoint
SHP_URL= 'https://areasciencepark.sharepoint.com/Sites/DataRepository/Documenti_
↳condivisi/General/'
SHP_DIR = 'dataset_imprese_fvg/test storicizzazione/finale/'
SHP_FILE = 'finanziamentiUE_progetti.csv'

f = df_proj_top_esv.to_csv(sep=';', index=False).encode('utf-8')
r = s.put(SHP_URL + SHP_DIR + SHP_FILE, f)
print(r.text)
```

```
[ ]: # UPLOAD df_org_elab in sharepoint
SHP_URL= 'https://areasciencepark.sharepoint.com/Sites/DataRepository/Documenti_
↳condivisi/General/'
SHP_DIR = 'dataset_imprese_fvg/test storicizzazione/finale/'
SHP_FILE = 'finanziamentiUE_org.csv'

f = df_org_elab.to_csv(sep=';', index=False).encode('utf-8')
r = s.put(SHP_URL + SHP_DIR + SHP_FILE, f)
print(r.text)
```