

FRED

Usando le api di FRED (<https://fred.stlouisfed.org/>) studiare il manuale delle API sul sito: (<https://fred.stlouisfed.org/docs/api/fred/>):

- Scrivere un *package* Python (<https://packaging.python.org/en/latest/>) con funzioni per:
 - Scaricare tutte le categorie a partire da una categoria data (la struttura è ad albero e la categoria root ha id=0)
 - Scaricare tutti i titoli e gli id delle serie da una categoria data e salvarli su un db (es. sqlite)
 - Scaricare le osservabili da una serie data e salvarli su un db
 - Graficare l'andamento di una osservabile (se possibile)
 - [Per gruppi > 3 persone] Se sono presenti dei nan deve essere presente un parametro booleano per interpolarli opzionalmente in maniera lineare
 - Esempio [2, 5, nan, nan, 12, 7] → [2, 5, 7.33, 9.66, 12, 7]
 - Esempio [1, 4, nan, nan, nan, 17] → [1, 4, 7.25, 10.5, 13.75, 17]
 - La formula è $s_{k+j} = s_i + (s_k - s_i)/(k - i + 1) * j \quad \forall j = [1, k - i]$ e s_k, s_i sono i valori ponte ($k > i$)
 - Graficare la media mobile (se possibile) di una osservabile a n giorni (n parametro)
 - [Per gruppi > 3 persone] Se sono presenti dei nan deve essere presente un parametro booleano per interpolarli linearmente
 - Salvare tutto ciò che si scarica su un database sqlite (o altro):
 - Una tabella delle serie scaricate e/o analizzate (record json)
 - Una tabella con l'elenco delle serie scaricate
 - Una tabella con le osservabili delle serie scaricate e/o analizzate (record json)
 - L'applicazione dovrà usare i dati locali qualora già scaricati
 - L'applicazione dovrà permettere di aggiornare una serie o una categoria su richiesta (update del database)
 - Tutte le funzionalità dovranno essere raccolte in funzioni
 - Scrivere un manuale d'uso delle funzioni generate
 - Scrivere un testo di relazione della modalità di lavoro (se in gruppo)
 - Allegare tutto il materiale (codice, database, relazioni) prodotto durante il lavoro
 - Case Study:
 - Usando le funzioni costruite, scaricare 3 serie da una singola (es. Categories > Money, Banking, & Finance > Exchange Rates > Daily Rates) e
 - Graficare le 3 serie su un solo grafico
 - Calcolare la covarianza (attenzione ai nan!)
 - Calcolare e graficare su un solo grafico per ogni serie la serie delle differenze prime ($s_{i+1} - s_i$)
 - Calcolare e graficare su un solo grafico per ogni serie la serie delle differenze prime percentuali $\frac{s_{i+1} - s_i}{s_i}$ fate attenzione agli ∞
 - Per ogni serie calcolare la retta di regressione
 - [Per gruppi > 3 persone] Introdurre due parametri temporali (data_inizio, data_fine) per disegnare tutti i grafici in un periodo temporale dato
- È ammesso l'uso di librerie ad alto livello (come pandas, numpy, matplotlib, ...) ma non è ammesso l'uso di librerie che già si interfacciano con FRED (es. fred 3.1, fred-api-wrapper, ecc.)
- La dimensione del gruppo dovrà influire sulla qualità del materiale prodotto nonché sulla proposta e creazione di ulteriori funzionalità a discrezione del gruppo stesso

WB

Usando le api di WorldBank (<https://datahelpdesk.worldbank.org/knowledgebase/articles/889392-about-the-indicators-api-documentation>):

- Scrivere un package Python (<https://packaging.python.org/en/latest/>) con funzioni per:
 - Scaricare tutti i Topics
 - Scaricare tutti gli indicatori per topic dato e salvarli su un db (es. sqlite)
 - Scaricare le osservabili di un indicatore data e salvarli su un db
 - Graficare l'andamento di una osservabile (se possibile)
 - [Per gruppi > 3 persone] Se sono presenti dei nan deve essere introdotto un parametro booleano per interpolarli opzionalmente in maniera lineare
 - Esempio [2, 5, nan, nan, 12, 7] → [2, 5, 7.33, 9.66, 12, 7]
 - Esempio [1, 4, nan, nan, nan, 17] → [1, 4, 7.25, 10.5, 13.75, 17]
 - La formula è $s_{k+j} = s_i + (s_k - s_i)/(k - i + 1) * j \quad \forall j = [1, k - i]$ e s_k, s_i sono i valori ponte ($k > i$)
 - Graficare la media mobile (se possibile) di una osservabile
 - [Per gruppi > 3 persone] Se sono presenti dei nan deve essere introdotto un parametro booleano per interpolarli linearmente
 - Salvare tutto ciò che si scarica su un database sqlite (o altro):
 - Una tabella delle serie scaricate e/o analizzate (record json)
 - Una tabella con l'elenco delle serie scaricate
 - Una tabella con le osservabili delle serie scaricate e/o analizzate (record json)
 - L'applicazione dovrà usare i dati locali qualora già scaricati
 - L'applicazione dovrà permettere di aggiornare una serie o una categoria su richiesta (update del database)
 - Tutte le funzionalità dovranno essere raccolte in funzioni
 - Scrivere un manuale d'uso di tutte le funzioni generate
 - Scrivere un testo di relazione della modalità di lavoro (se in gruppo)
 - Allegare tutto il materiale (codice, database, relazioni) prodotto durante il lavoro
 - Case Study:
 - Usando le funzioni costruite, scaricare 3 serie da un topic (es. topic Agriculture & Rural Development, indicators: Agricultural machinery, tractors | Fertilizer consumption (% of fertilizer production| ...))
 - Graficare le 3 serie su un solo grafico
 - Calcolare la covarianza (attenzione ai nan!)
 - Calcolare e graficare su un solo grafico per ogni serie la serie delle differenze prime ($s_{i+1} - s_i$)
 - Calcolare e graficare su un solo grafico per ogni serie la serie delle differenze prime percentuali $\frac{s_{i+1} - s_i}{s_i}$
 - Per ogni serie calcolare la retta di regressione
 - [Per gruppi > 3 persone] Introdurre due parametri temporali (data_inizio, data_fine) per disegnare tutti i grafici in un periodo temporale dato
- È ammesso l'uso di librerie ad alto livello (come pandas, numpy, matplotlib, ...) ma non è ammesso l'uso di librerie che già si interfacciano con WB (es. world-bank-data, wbgapi, wbdata, ecc.)
- La dimensione del gruppo dovrà influire sulla qualità del materiale prodotto nonché sulla proposta e creazione di ulteriori funzionalità a discrezione del gruppo stesso

Istruzioni

1. Formare un gruppo da [1] a 5 persone
2. Nominare un referente di gruppo
3. Il referente dovrà per prima cosa:
 - a. Inviare una e-mail a regoli@uniroma2.it con subject: [Progetto CPS]
 - b. Inserire nella e-mail l'elenco degli studenti appartenenti al gruppo (incluso sé stesso)
 - c. Specificare nella e-mail il progetto scelto
 - d. Specificare il nome del gruppo
4. Il referente riceverà una notifica
5. Il referente dovrà successivamente:
 - a. Eseguire l'upload del file zippato contenente tutto il materiale richiesto sulla piattaforma Teams nella sezione attività (Consegna progetto fine corso) entro tre giorni prima della data dell'esame orale scelto
6. Tutti gli elementi di un gruppo dovranno svolgere l'esame nello stesso giorno
7. Durante la prova d'esame a ciascun membro del gruppo verranno fatte domande relative al lavoro svolto, pertanto, anche se il lavoro è in gruppo dovrete essere capaci di rispondere su ciascuna parte del codice prodotto
8. È prevista la possibilità di proporre un proprio progetto con le stesse modalità

Per ogni dubbio: regoli@uniroma2.it con subject [Dubbio CPS]