Allora, essenzialmente monte ha preso gli esercizi nei quali in alcuni punti erano presenti delle imprecisioni o delle cose che davo un risultato senza la motivazione, e mi ha fatto partire da capo riscrivendo i dati e reimpostando i calcoli dandomi solo all'inizio delle piccole puntualizzazioni per vedere se mi rendevo conto dell'errore in fase di svolgimento, poi se vedeva che non mi accorgevo di niente mi dava qualche indizio/contestazione in piú su quello che stavo scrivendo ad esempio:

Nell'esercizio 5 del primo esonero, sulla speranza condizionata, E[X|Z] avevo scritto E[X|Z]=E[X|Z=0]+E[X|Z=1]+E[X|Z=2] ed é partito dicendomi e riscrivimi come é definita Z e dimmi perche viene cosi e riscrivimi come é definita la speranza condizionata per le va uniformi,
ed il senso é perche Z é binomiale (2,p) quindi so che valore assume e con che probabilitá ecc ecc e diciamo alla fine arriviamo al fatto che la speranza condizionata viene 1/2Z
che per l'appunto é anch'essa una variabile aleatoria, io nella scrittura precedente di E[X|Z] avendo saltato le funzioni indicatrici a piedi pari ed avendole messe solo alla fine, quindi sto scrivendo che
una la variabile aleatoria (che é funzione di Z, nel senso E[X|Z]=g(Z)) é uguale ad un numero e questa é l'imprecisione.

Poi sempre sul primo esonero nell'esercizio 3, questo forse é un po piú tricky e ti puo esse utile essenzialmente era sbagliato un dettaglio sul come si imposta il calcolo (che ha detto hanno sbagliato in molti)
e avevo scritto essenzialmente che P(Y<y) per y<0 era 0 invece era 1/2 mo di questo c'é la foto cosi si capisce meglio. Lui mi ha fatto riscrivere X, la sua densitá, g(X) e il grafico di g(X) e mi ha fatto ragionare un po sulle probabilitá.

Poi un'altra cosa nel secondo esonero é che a un certo punto quando ci stava da scrivere la consistenza dello stimatore ho scritto che lo stimatore media campionaria Xn tende ad E[X] in probabilitá, ora di questo non ti so dare troppi dettagli perche lo avevo
fatto direttamente in bella e non ti vorrei dire cose sbagliate, qui la domanda é stata direttamente, quello che hai scritto é giusto peró fammi vedere perché dato che nel compito non lo hai scritto,
e la cosa é che Xn tende in probabilitá alla sua media E[Xn] che per le proprietá dello stimatore media campionaria é E[X] e questo lo si fa vedere con chebicev come in foto.

Regoli invece era ho due urne la prima X palline rosse e Y verdi e la seconda con T rosse e S verdi, pesco dalla prima urna una pallina e la ficco nella seconda urna (senza guardare che pallina ho preso) (io ho fatto un elenco puntato, però puoi fare un po come vuoi flow chart pseudo codice)
poi pesco una pallina dalla seconda urna, calcolare empiricamente la probabilitá di pescare la pallina rossa. La mia risposta é stata letteralmente uno schemino del tipo:
-Creo due array che rappresentano le due urne quindi con X+Y elementi e T+S elementi
-Con una funzione random di qualche tipo pesco un elemento del primo array e lo rimuovo
-Faccio l'append al secondo array
-Ripesco random e vedo di che colore é
-Ripeto tante volte l'esperimento e vedo la media dei risultati ottenuti.

A me ha chiesto di estrarre alcuni dati specifici partendo da un dataset.
Si poteva fare senza alcuna funzione, era semplicemente da capire quale slicing usare
Possibile soluzione: Senza numpy devo fare 'open' del file csv, fare skip della prima riga (i nomi delle colonne), prelevare ogni riga interamente, fare lo split in base ad un carattere particolare (se è un csv si usa ; o , ma non è detto che in un txt si faccia uguale) con line.split(";"), questo mi permette di avere i vari campi separati (come altezza e peso) a cui posso fare append sui rispettivi array. Lo faccio per tutte le linee, poi chiudo il file.
Con numpy è più semplice, perchè ho il metodo genfromtxt, in cui definisco skip header = 1 (da che linea parto,), quale è il delimitatore, e quali sono i nomi delle colonne (es: sesso, altezza peso). faccio quindi dataset = np.genfromtxt(nomefile, dtype, skip_header, delimiter,...) e posso accedere direttamente agli elementi delle colonne specificando dataset[" nomecolonna"].

A me Regoli ha chiesto di scrivere una funzione python che calcoli l'attesa media (anche se poi sostanzialmente si riferiva al numero di pacchetti necessari) per il completamento di un album di 100 figurine dati pacchetti da 5 figurine dove le figurine nello stesso pacchetto non possono essere doppioni e le figurine sono equiprobabili.
Mi ha fatto scrivere una bozza di soluzione su un foglio quindi niente di troppo specifico a livello di funzioni... Gli ho parlato giusto delle librerie random di numpy tramite cui simulare l'apertura di un pacchetto. Poi ovviamente l'album potevi farlo sia come array numpy che semplice lista python in realtà, E poi anche qui alla fine lo stesso esperimento lo dovevi ripetere n volte e fare poi la media.
mia idea: creo vettore di 100 elementi (indice da 0 a 99), tutti settati a 0. Creo array "pacchetto" di size 5, randomicamente estraggo 5 volte un numero da 0 a 99, verificando che non sia già presente nel pacchetto. Inizializzo un counter pacchetti, che mi dice quanti ne ho aperti. Creato il pacchetto, la figurina 'x' andrà a settare l'indice 'x' del vettore album a 1 (true). Ripeto, aprendo nuovi pacchetti (ed incrementando il counter) fino a che tutto il vettore non sia pieno di 1 (o la somma del vettore non sia 100). Quando ciò avviene, metto in un vettore " n°pacchetti" il counter finale. Riparto da 0, e ogni volta che finisco metto il numero di pacchetti necessari in questo vettore "N° pacchetti". Finiti i tentativi, faccio una media su questo vettore, trovando il numero medio di pacchetti necessari per completare l'album.

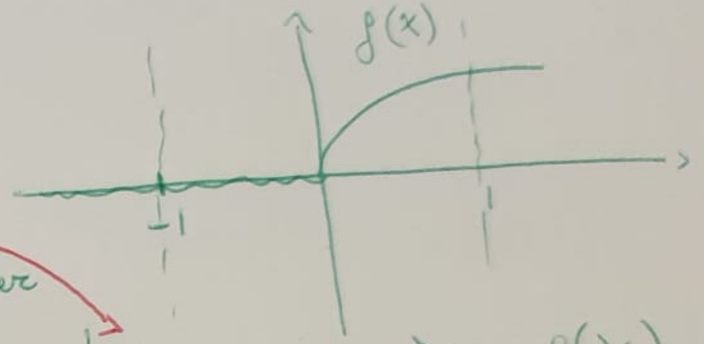
Slicing: Banalmente, mi permette di prendere dei sottoinsiemi di un array, oppure specificare un "salto" (es: prendi elementi di indice pari). Con numpy, posso passargli addirittura come indice un vettore. Se ho un vettore di partenza, e stampo vettore[boolvector], stamperà solo gli elementi con boolvector = true, o comunque quelli che rispettano una certa condizione. Se voglio stampare i valori > determinato valore? creo array di supporto true se bmi>32, false altrimenti. e quindi faccio print vector[supportvector].
Posso plottare con matplotlib.

BeautifulSoup: libreria Python per ottenere dati da HTML, XML etc, permettendo la loro estrazione, rimuovendo i contenuti html. Come? lavorando su scatole nidificate, e passandogli attributi o id delle classi per trovare dati univoci. Esistono metodi .find(), .find_all(), etc...
Normalmente, da una pagina avente tabella preleviamo un url, estraiamo il contenuto grezzo, applichiamo beatifulsoup, e applichiamo find_all (specificando table o class) da mettere in un oggetto table da ritornare.

Selenium: tool per gestione automatizzata dei browser, a noi interessa particolarmente selenium webdriver, il quale simula il comportamento di un utente reale, come i click su un sito. I web browser che lo supportano hanno un web driver specifico (quindi non hanno tutti lo stesso, anzi, anche il cambio di versione può influire). Perchè ci serve? Perchè in molti siti, per avere una vista completa delle informazioni, bisogna avere interazione (espandi recensione, apri spoiler, visualizza tutto). Queste informazioni vengono caricate solo dopo averci cliccato, sfruttando Javascript.

$X \sim \text{Unif}(-1,1)$ $\quad g(x) = \begin{cases} 0 & \text{se } x<0 \\ \sqrt{x} & \text{se } x \geq 0 \end{cases}$

~~$P(X<0) = P(-X)$~~



$\rightarrow P(-1<X<0) = \frac{1}{2}$ $\rightarrow$ perché uniforme

$\rightarrow P(Y \leq y) = P(g(X) \leq y) = \begin{cases} P(X=0) \text{ se } y<0 \\ P(\sqrt{X} \leq y) \text{ se } y \geq 0 \end{cases}$ Questo succede per $X \in (-1,0)$

$\begin{cases} \frac{1}{2} \text{ se } y<0 \\ --- \text{ se } y \geq 0 \end{cases} \rightarrow g(X) = \begin{cases} 0 & \text{se } X<0 \\ \sqrt{X} & \text{se } X \geq 0 \end{cases}$ Cioè per $X \in (-1,0)$

$\bar{X}_n \xrightarrow{P} E[X]$

$P(|\bar{X}_n - E[X]| \geq \varepsilon) = P(|\bar{X}_n - E[\bar{X}_n]| \geq \varepsilon) \leq \frac{D^2[\bar{X}_n]}{\varepsilon^2} = \frac{\frac{1}{n} D^2[X]}{\varepsilon^2} \xrightarrow[n \to \infty]{} 0$

1) $E[X|Z] = \sum_{k \in \{0,1,2\}} E[X|Z=k] = \underbrace{E[X|Z=0]}_{\text{Se scrivo solo questo è un numero ci va } 1_{\{Z=0\}}} + E[X|Z=1] + E[X|Z=2]$

$$E[X|Z=0] = \frac{1}{P(Z=0)} \int_{\{Z=0\}} X \, dP = 0$$

$\{Z=0\} = \{X=0, Y=0\}$

$$E[X|Z=1] = \frac{1}{P(Z=1)} \int_{\{Z=1\}} X \, dP = \frac{1}{2pq} \cdot pq + 1 \cdot pq = = \frac{1}{2}$$

$\{Z=1\} = \{X=0, Y=1\} \cup \{X=1, Y=0\}$

$$E[X|Z=2] = \frac{1}{P(Z=2)} \int_{\{Z=2\}} X \, dP = \frac{1}{p^2}(p^2) = 1$$

$\{Z=2\} = \{X=1, Y=1\}$

$$\rightarrow E[X|Z] = \frac{1}{2} 1_{\{Z=1\}} + 1_{\{Z=2\}} = \frac{1}{2} Z \quad \text{q.c.}$$