

1. Matemàtica Discreta - Seminari 1 - Probabilitat

1.1. Funcionament dels seminaris

El primer i primordial que heu de saber és:

ÉS MOLT IMPORTANT LLEGIR BÉ ELS ENUNCIATS I CONTESTAR EL QUE ES DEMANA DE LA MANERA QUE ES DEMANA, NO UNA ALTRA COSA.

Aquest és el motiu més habitual de problemes tant als seminaris com als exàmens.

Durant aquests seminaris intentarem aplicar, d'una forma pràctica, aquells coneixements que anem adquirint a teoria i problemes de l'assignatura. Per seguir els seminaris necessiteu un ordinador, un editor de textos i algun tipus de IDE que us permeti programar i executar Python3.

Cada seminari es compondrà d'una sèrie de preguntes i d'algun exercici de programació. Per facilitar la feina, per a cada seminari tindreu disponible un esquelet de programa que heu d'omplir i d'un conjunt de tests per validar que el vostre programa funciona. Per tant, sempre disposareu d'una carpeta anomenada esquelet amb almenys tres fitxers. Un main per fer proves o crear objectes concrets, un o més fitxers que heu d'omplir i on tindreu pistes sobre el que heu de fer i un fitxer de tests.

Per a començar sempre heu de duplicar la carpeta esquelet i dir-li 'solucio' i en aquesta carpeta és on heu de fer els canvis. La carpeta esquelet no l'heu de tocar per res.

El fet que el programa passi el conjunt de tests donat, no vol dir que sigui correcte. Primer, perquè els tests donats no comproven tots els casos és a dir, es valorà la capacitat de tractar casos extrems o no contemplats, i a més, es valorarà que el codi segueixi els conceptes d'Eficiència, Escalabilitat i Llegibilitat. En canvi, no passar el conjunt de tests sí que es considerarà un seminari no correcte.

Els lliuraments es compondran d'un pdf amb les respostes a les preguntes teòriques, d'un arxiu README.md amb les instruccions per a l'execució i els arxius .py necessaris per a la seva execució. Tot comprimit en un zip o un tar.gz.

1.2. Exercici 1 - Probabilitat d'encertar 4 preguntes de tipus test de forma aleatòria

Imagineu que teniu un examen compost de 4 preguntes, cada una de les 4 preguntes té al seu torn 4 possibles respostes. Imagineu que no en teniu ni idea de cap de les preguntes i, per tant, les heu de contestar aleatòriament.

1. **A mà**, calculeu la probabilitat de, donada una única pregunta, encertar-la. I de fallar-la?
2. **A mà**, calculeu la probabilitat de, donades les 4 preguntes juntes: Encertar-ne una, encertar-ne dues, encertar-ne tres i encertar-ne 4.

1.3. Exercici 2 - Probabilitat d'aprovar un examen

Ara tenim un examen de 10 preguntes de 4 respostes cada una, de les quals ens en sabem 4 segures. Les altres 6 no les sabem. Cada pregunta encertada és 1 punt mentre que cada pregunta fallada son $-0,33$. Calculeu, **a mà**:

1. Probabilitat **d'aprovar** si en contesto només 1 aleatòriament.
2. Probabilitat **d'aprovar** si contesto les 4 aleatòriament.

1.4. Exercici 3 - Programem-ho!

Ara volem fer una funció que provi empíricament les probabilitats d'aprovar donat un examen amb *num_preguntes* preguntes, cada una d'elles composta de *num opcions* possibles respostes. D'aquest examen sabem segur, segur, *num_preguntes_resposes* de les preguntes i no en sabem la resta. Posem que contestem aleatòriament *num_preguntes_aleatories*. Per a fer-ho farem una funció que faci exàmens amb les condicions descrites (no que calculi les probabilitats amb la fórmula!). Teniu en compte que:

- L'examen és sobre 10 i, per tant, cada pregunta val $valor_pregunta = 10/(num_preguntes)$.
- Cada pregunta resta $resta_pregunta = (valor_pregunta)/(num_opcions - 1)$.
- Aprovar significa treure un 5 o més.
- El retorn de la funció és les probabilitats (entre 0 i 1) d'aprovar donades les condicions exposades.
- Per a fer-ho estadísticament rellevant, el programa haurà de fer 100,000 exàmens.

Ara programarem aquesta funció. Copieu la carpeta 'esquelet', l'enganxeu a la mateixa ubicació que 'esquelet' i l'anomeneu 'solució'. Obriu la carpeta que heu creat anomenada 'solució'. Dins trobareu la carpeta src amb l'esquelet del codi i una carpeta test amb els testos. Per probar els testos heu de executar la comanda `pytest nom_arxiu` des de la carpeta "solucio" que acbeu de copiar.

Pistes;

- La funció `randint(1, numopcions)` us donarà un nombre entre 1 i numopcions.
- Per fer un for entre dos números poseu usar la estructura `for intent in range(0, 100000):`. Llavors la variable `intent` anirà prenent els valors 0, 1, 2, etc.
- Necessitareu instal·lar `pytest` per executar els testos i mirar que vagi bé. Podeu fer-ho mitjançant `pip`. En concret podeu fer `pip install pytest`. Si useu `pycharm` ho podeu fer des del gestor de `packages` de `pycharm`. <https://www.jetbrains.com/help/pycharm/installing-uninstalling-and-upgrading-packages.html>

1.5. Exercici 4 (opcional): Però jo en puc descartar algunes!

Que passa si algunes de les respostes a les preguntes que no sé, les puc descartar? Canvieu el vostre codi per a implementar el cas en què a cada pregunta que no sàpiga (*num_preguntes_aleatorias*) en realitat estigui dubtant entre x respostes on $x \leq num_opcions$.

Per a fer-ho podeu substituir el paràmetre *num_preguntes_aleatorias* que és un enter, pel paràmetre *preguntes_aleatorias* que és una llista amb el nombre d'opcions en les quals dubteu per a cada pregunta aleatòria. Així, si estiguéssiu entre 2, 3, 4 i 2 possibles respostes a 4 preguntes, llavors *preguntes_aleatorias* = [2, 3, 4, 2].

Lliurament: Heu d'entregar, per grups, un tar.gz o zip amb les preguntes respostes i el codi al CV abans del 19 de març a les 23:59.