eman ta zabal zazu
Universidad del País Vasco
Euskal Herriko Unibertsitatea
GASTEIZKO INGENIARITZAKO UNIBERTSITATE ESKOLA
ESCUELA UNIVERSITARIA DE INGENIERÍA DE VITORIA-GASTEIZ

ARAU OROKORRAK

- Taldeek bere kabuz lan egin behar dute, beste taldeen inolako laguntzarik gabe. Era susmagarri batean berdina edo antzekoa den kodea aurkezten dute taldeek ez dute proiektua bakarrik suspenditu, irakasgaia ere bai.

- Kodearen azkeneko bertsioa eGela-ra igoko da gutxienez astero behin. Kontutan denbora gutxian egindako aldaketa handiak susmagarriak direla irakaslearen ikuspuntutik. Arazoak ekiditeko, zuen kodea ahal den aldi kopuru handienean igo, zuen lanaren eboluzioa jarrai dezagun.

PROIEKTUAREN HELBURU OROKORRA

Proiektu honetan logistika konpainia batentzat lan egingo dugu, beraien datuak erabaki onak hartu ahal izateko analizatuz. Konpainiak bi gidari ezberdin ditu Donostia-San Sebastián eta Vitoria-Gasteiz artean produktuak banatzeko. Bi bide ezberdin erabili daitezke toki batetik bestera bidaiatzeko: errepide nazionala (N1) eta autopista (A1). Hainbat Matlab funtzio eta script garatuko ditugu konpainiari ze gidari eta ze bide erabili behar diren erabakitzeko.

TALDEAREN EBALUAKETA

Talde bakoitzak talde nota bat edukiko du, ikasle bakoitzaren nota kalkulatzeko oinarri bezala erabiliko dena (taldeko nota baino handiagoa edo txikiagoa izan daiteke). Taldearen nota maximoa ondo burututako araza kopuruaren araberakoa izango da:

- 100%: Ataza guztiak

- 75%: 1, 2, 3 eta 4 atazak

PROGRAMAZIORAKO GIDAK

- Funtzio baten barruan ez informaziorik pantailan idatzi edo grafikoki poteatu.
- Posible denean, funtzioetan garatutako kodea berrerabili. Kasu gehienetan geuk esango dizuegu nola diseinatu funtzioak.
- Posible denean, begiztak erabili (*for*/*while*) balio bildumen gainean iteratzeko, existitzen den kodea errepikatu ordez.
- Aldagai/script/funtzioentzako izen esanguratsuak erabili, edonork kodea ulertu dezan berau irakurtzen denbora gehiegirik pasa gabe.
- Programan iruzkinak idatzi programen atal bakoitzaren helburuak zeintzuk diren azalduz.

DATU SARRERA FITXATEGIAK

Ondorengo fitxategiekin egingo dugu lan:

1. Bi gidarien log-ak bi bide ezberdinetan:
   - `a1-driver1-log.csv`
   - `a1-driver2-log.csv`
   - `n1-driver1-log.csv`
   - `n1-driver2-log.csv`

   Fitxategi hauetan jatorri puntuarekiko distantzia *(km)* eta ibilgailuak bidean zeharreko puntu batzuetan daukan abiadura (*km/orduko*) ditugu. Lehenengo zutabean jatorrarekiko distantziak daude, eta bigarrenean abiadurak:

   Adibidea: n1-driver1-log.csv

```
0,0
0.10013,8.5168
0.20027,16.901
0.3004,24.103
0.40053,30.048
0.50066,35.717
...
```

2. Bide bakoitzaren altuera profila:
   - `a1-height.csv`
   - `n1-height.csv`

   Fitxaregi hauetan bidea bakoitzaren informazioa daukagu:

Example: n1-height.csv

```
Latitude,Longitude,Elevation(metres),Distance (KMs),Gradient
43.31839,-1.98127,5.7,0,
43.31821,-1.98169,6.2,0.039,
43.31876,-1.98207,6,0.108,
43.31887,-1.98214,6,0.121,
43.31936,-1.98249,6.1,0.183,
43.31939,-1.98252,6.1,0.187,
43.31945,-1.98256,6.1,0.195,0
43.31917,-1.9833,6.3,0.262,0
...
```

Zutabe goiburukoa duen fitxategiko lehenengo lerroa ekidin beharko dugu datuak irakurtzeko garaian.

3. Bide bakoitzerako abiadura limiteak :
   - `a1-speed-limit.csv`
   - `n1-speed-limit.csv`

Adibidea: a1-speed-limit.csv

```
0;90
5;120
95;90
```

Fitxategi hauetan abiadura limitei buruzko informazioa daukagu. Lehenengo zutabean jatorriatik hasita distantzia jakin baterarte (*km*-etan), eta, bigarren zutabean, puntu horretan baimendutako abiadura maximoa (*km/orduko*). Goiko adibidean, gidariek lehenengo 5 kilometroetan 90 *km*/*orduko* abiaduran joan daitezke gehienez, ondoren 5 . kilometroatik 95 .erarte , 120 *km*/*orduko* abiaduran joan daitezke.

ATAZAK

1. Ataza:

Script bat idatzi bi bideetan elebazio profila analizatzen duena:

1. Bi subplot sortuko ditu:
    - Bi bideen altuerak (*m*) jatorriko puntuarekiko distantziaren funtzioan (*km*).
    - Bideak (longitudea latitudearen aurka)

    Subplot bakoitza titulu baten bidez identifikatuta geratuko da, eta biek *x/y* ardatzak era egokian etiketatuta edukiko dituzte.Adibidez, denbora adierazteko erabiltzen den ardatz batentzako etiketa egoki bat: *denbora (s)* izan daiteke. Plot-a *'route-elevations.png'* izenarekin gordeko da script-ak dauden karpeta berdinean.
2. Script-ak kontsola leihotik ondorengo estatistikak ere adieraziko ditu: elebazioaren media, desbiderazio estandarra, minimoa eta maximoa. Irteera honakoa izan beharko litzateke:

```
n1 route statistics:
Mean height: 318.37 (sd: 235.54)
Height range: [4.20, 656.10]

a1 route statistics:
Mean height: 189.71 (sd: 181.70)
Height range: [0.00, 825.80]
```

2. Ataza:

Idatzi script bat bide bakoitzean gidari bakoitzaren log-ak analizatzen dituena:

1. Script-ak plot bat sortuko du abiadurak (*km/orduko*) jatorriarekiko distantzien aurka (*km*) aurka adieraziz. Subplot ezberdin bat erabiliko da bide bakoitzera, bakoitzean bi gidarien abiadurak egongo direlarik. Subplot bakoitza titulu batekin era egokian identifikatuko da, eta subplot bakoitzak etiketa bat edukiko du bi gidarien artean bereizteko. Etiketak testu egoki batekin adierazita egongo dira.
2. Plot-a eta batazbestekoa, desbiderazio estandarra ,minimoa eta maximoa kalkulatuko ditu gidari/bide bakoitzeko. Irteera honakoa izan beharko litzateke:

```
driver1 statistics in route n1
Mean speed: 95.99 km/h (sd. 10.14)
Min-Max speed: [0.00,118.44]

driver2 statistics in route n1
Mean speed: 98.76 km/h (sd. 10.78)
Min-Max speed: [0.00,121.43]

driver1 statistics in route a1
Mean speed: 101.29 km/h (sd. 14.82)
Min-Max speed: [0.00,119.00]

driver2 statistics in route a1
Mean speed: 104.79 km/h (sd. 15.62)
Min-Max speed: [0.00,122.17]
```

3. Ataza:

Idatzi *mainMenu.m* script-a, erabiltzaileari honako menua erakusten diona:

```
#######  MENU  ########:
1. Show route plots/statistics
2. Show driver plots/statistics
3. Time calculations for each driver/route
4. Check speed limits
5. Fuel consumption calculations for each driver/route
6. Exit
Choose an option:
```

Erabiltzaileak orduan aukeretako bat aukeratuko du eta, aukeratutako aukeraren arabera (1etik 6ra), dagokion script-a deituko du. Menuaren sarrera bakoitza proiektuko ataza bati dagokio. Sarrera zenbakia zuzena ez bada, errore mezu bat erakutsiko da honakoa adieraziz: "*Aukera ezegokia: 1 eta 6 artean egon behar da*".

Menua erakutsi aurretik, kontsola garbitu behar da. Menu hay aukeretako bat aukeratu ondoren erakutsiko da, aukeratutako aukera *6* (*Exit*) izan arte, script-a mezu gehiago erakutsi gabe bukatuko duen aukera.

Emaitzak errazago irakurri ahal izateko, script hauen emaitzak aldatu beharko dira '*Sakatu edozein tekla jarraitzeko*' testua daikan mezu bat adieraziz, eta ez da amaituko erabiltzailean zerbait tekleatu arte. Era honetan, script-aren emaitza pantailan adieraziko da erabiltzaileak edozein tekla sakatu eta menu nagusia berriro azaldu arte.s

4. Ataza:

Idatzi script bat gidari bakoitzaren bide bakoitza burutzeko behar izan den denbora kalkulatzeko. Hainbat pausotan egingo dugu:

1. Idatzi funtzio bat x eta y balio bilduma bat linearki interpolatzen dituena:

```
function [ interpolatedY ] = interpolateLinearly( xVector, yVector , x)

%This function gets a vector with x-values and a vector with y-values
%and returns the interpolated value of y for the given x.
```

   Funtzioaren egokitasuna honako balioekin frogatu:

```
interpolateLinearly([10 20 40],[120 150 130],10) => 120
interpolateLinearly([10 20 40],[120 150 130],15) => 135
interpolateLinearly([10 20 40],[120 150 130],25) => 145
interpolateLinearly([10 20 40],[120 150 130],40) => 130
```

2. Idatzi funtzio bat non, kilometro kopuru bat emanda, dagokion metro kopurua bueltatzen duena:

```
function [ m ] = toMeters( km )
```

   Funtzioaren egokitasuna honako balioekin frogatu:

```
toMeters(2.3) => 2300
toMeters(0.3) => 300
```

3. Idatzi funtzio bat abiadura bat *km/orduko* formatuan emanda, bere *m/segunduko* abiadura baliokidea ematen duena:

```
function [ msSpeed ] = toMetersPerSecond( speedKmH )
```

   Funtzioaren egokitasuna honako balioekin frogatu:

```
toMetersPerSecond(120) => 33.3333
toMetersPerSecond(15) => 4.1667
```

4. Idatzi funtzio bat bide bakoitzean emandako denbora kalkulatzen duena:

```
function [ estimatedTime ] = estimateTime( kms, speedKmH, numSlices)
```

```
%This function is given a vector with a vehicle's speed (speedKmH) at
%different points (kms) and the number of integration slices we want to
%use (numSlices)
```

Bide-gidari log-ak erreferentzia bezala edukita, *kms*-k lehenengo zutabeko balioak edukiko lituzke, eta *speedKmH*-ek bigarren zutabeko balioak edukiko lituzke.

Funtzio honek emandako bektorean dauden abiadurak linearki interpolatuko ditu erabiliko ditugun *numSlices* integrazio zatientzako. Abiadura aurreko azpiatazan garatutako funtzioa erabiliz interpolatuko dugu aati bakoitzarentzat, eta zatiaren luzera errekorritzeko behar izan den denbora kalkulatuko dugu. Emaitza bidea errekorritzeko behar izan den segundu kopurua izango da.

Funtzioaren egokitasuna honako balioekin frogatu:

```
estimateTime([0 1 2 3],[40 50 40 30],1000) => 264.2797
estimateTime([0 1 2 3],[40 50 40 20],1000) => 285.5495
```

5. Idatzi funtzio bat non, segundu kopuru bat emanda, karaktere kate bat itzultzen duen OO:MM:SS formatuan (orduak, minutuak and segunduak, bakoitza bi digituekin). Segundu frakzioak ez ditugu kontutan edukiko.

```
function [ hms ] = toHMS( seconds )
```

Funtzioaren egokitasuna honako balioekin frogatu:

```
toHMS(234.4) => 00:03:54
toHMS(3601) => 01:00:01
```

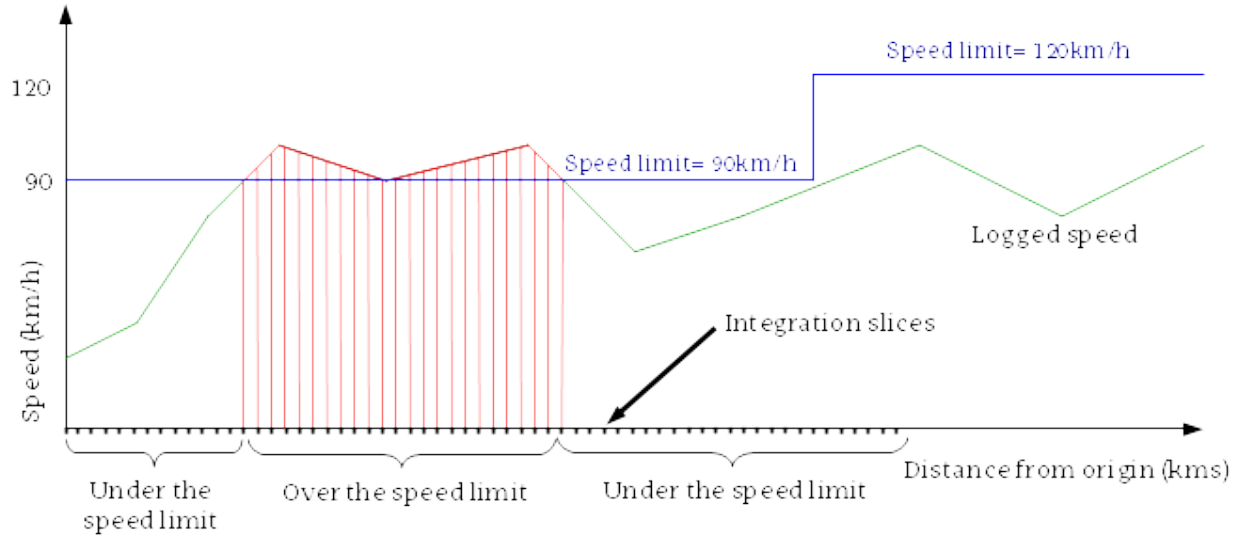
6. *estimateTime* funtzioa erabilita, script bat idatzi denbora estimatzen duena bide-gidari konbinazio guztientzako *ordu:minutu:segundu* formatuan.

```
Estimated time for driver1 in route n1: 1:58:54
Estimated time for driver2 in route n1: 1:56:13
```

```
Estimated time for driver1 in route a1: 1:54:09
Estimated time for driver2 in route a1: 1:51:12
```

5. Ataza:

Script bat idatzi gidari bakoitzak tarte bakoitzeko abiadura limitearen gainetik zenbat denbora gidatu duen analizatzeko.



4 log fitxategiak eta abiadura limitak dituzten beste bi fitxategi erabiliko ditu. Bakoitzak bereizmen ezberdina dauka, beraz interpolatu beharra edukiko dugu.

Oinarrizko ideia: Bide-gidari log bat emanda, gidaria abiadura limitetik gora ibili den bidearen zatiak integratuko ditugu. Hori egin ahal izateko, zati txikiak erabiliko ditugu eta hoietako bakoitzarentzat interpolatutako abiadura kalkulatuko dugu zatiaren hasieran, eta puntu berdinean dagoen abiadura limitearekin alderatu. Abiadura limitea baino altuagoa bada, zatiaren azalera integratuko dugu eta gidaria limitearen gainetik ibilitako kilometro kopuruari gehituko diogu.

Ondoko pausuak jarraitzea gomendatzen dugu:

1. Abiadura limiteak, log-ean dauden abiardurekin alderatuta,era ezberdin batean interpolatu behar dira. Adibidez, abiadura limiteak honakoarekin ematen badira:

```
kms=         [0   10   40];
speedLimit=  [90  120  90];
```

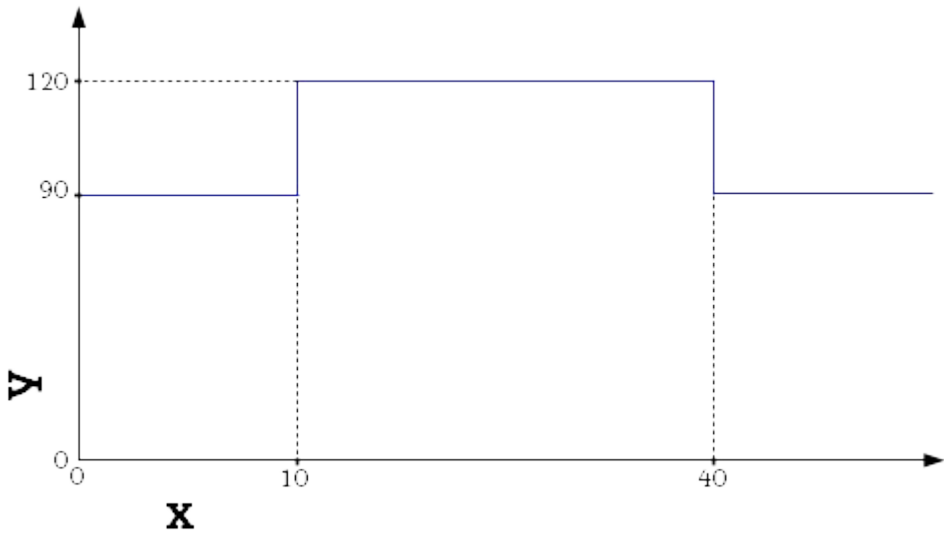
honek adierazten du abiadura limitea lehenengo 10 kilometroetan 90km/orduko dela, ondoren 120 km/orduko 10 kilometrotik 40ra, eta 90 km/orduko 40 kilometrotik aurrera.

Garatu ondoreko funtzioa, emandako x balioarentzat interpolatutako y balioa bueltatzen duena:

```
function [ interpolatedY ] = interpolateToTheLeft( xVector, yVector , x)

% This function gets a vector with x-values and a vector with y-values
% and returns the interpolated value of y for the given x.
```

Adibideko balioak hartuta, funtzioak yVector-en xVector-en **x-rekiko ezkerraldetik hurbilen dagoen balioa** bueltatu beharko luke. Bisualki, funtzioaren emaitza honela adieraziko zen grafiko baten bidez:



Funtzioaren egokitasuna honako balioekin frogatu:

```
interpolateToTheLeft([0 10 40],[90 120 90],3) => 90
interpolateToTheLeft([0 10 40],[90 120 90],15) => 120
interpolateToTheLeft([0 10 40],[90 120 90],30) => 120
interpolateToTheLeft([0 10 40],[90 120 90],100) => 90
```

2. Funtzio bat idatzi non, bide batean zehar hainbat puntu ezberdinetan jasotako abiadura balio ezberdinak (*driverLogKm* eta *driverLogSpeed*), eta kalkuluetan erabiliko ditugun zati kopura (*numSlices*) emanda, gidaria abiadura limite gainetik (*kmsAboveSpeedLimit*) egondako kilometro kopurua bueltatzen duena, eta bidean abiadura limite gainetik egondako kilometro portzentaia (*percentAboveSpeedLimit*).

```
function [ kmsAboveSpeedLimit,percentAboveSpeedLimit ] =
    checkSpeedLimits( driverLogKm, driverLogSpeed, limitKms, limitSpeeds, numSlices)
```

3. Idatzi script bat, aurreko funtzioa erabiltzen duena, abiadura limite bat dela eta multa bat jasotzeko arriskua neurtzen duena. Bide-gidari bakoitzerako mezu bat idatziko dugu gidaria abiadura limitearen gainetik egon den portzentaiaren arabera. %10etik gorakoa bada, *arrisku altua* dagoela kontsideratuko dugu,eta *arrisku moderatua* zenbaki hori baino txikiago bada. %0ko portzentaia badago, noski, *arriskurik ez*.

   Irteera honakoa izan beharko litzateke:

```
Analyzing: Driver= driver1, Route= n1
Mild infraction risk: Kms above the speed limit= 10.43 (5.64% of
the route)

Analyzing: Driver= driver2, Route= n1
HIGH INFRACTION RISK: Kms above the speed limit= 149.39 (80.75% of
the route)

Analyzing: Driver= driver1, Route= a1
No risk of infraction

Analyzing: Driver= driver2, Route= a1
HIGH INFRACTION RISK: Kms above the speed limit= 131.29 (70.97% of
the route)
```

6. Ataza:

Script bat idatzi kamioi batek bide-gidari log bakoitzean zenbat fuel kontsumitu duen estimatzeko. Honako formula erabiliko dugu fuel kontsumoa estimatzeko:

$$f = \frac{d * (0.00009\, v + (0.021\, a + 0.087\theta)^2)}{33} \quad ,$$

non f gidatzeko kontsumitu den fuel kantitatea den (litroetan), eta d $\theta\ (rad)$ angelua daukan aldapa batean $v\ (m/s)$ abiadura eta $a\ (m/s^2)$ azelerazioarekin egindako metro kopurua den. Formula honek abiadura, azelerazio eta aldapa angelu konstanteak direla onartzen du, beraz bide-gidari log-aren distantzian zehar integratu beharko dugu.

Ondorengo pausuak jarraitzea gomendatzen dugu:

1. Idatzi funtzio bat distantzia motzetan fuel kontsumoa kalkulatzen duena abiadura, azelerazioa eta aldapa angelua konstanteak direla suposatuz:

   ```
   function [ fuelExpenditure ] = calculateFuelExpenditure(v,a,theta,d)
   ```

   Funtzioaren egokitasuna honako balioekin frogatu (*33.33 m/s = 120 km/orduko*):

   ```
   calculateFuelExpenditure(33.33,0,0,1000) => 0.0909
   calculateFuelExpenditure(33.33,1,0.3,1000) => 0.1581
   calculateFuelExpenditure(33.33,1,-0.3,1000) => 0.0917
   ```

2. Idatzi funtzio bat bide oso bateako fuel kontsumoa kalkulatzen duena:

   ```
   function [ fuelExpenditure ] =
   calculateRouteFuelExpenditure(routeKms,routeHeights,logKms,logSpeeds,num
   Slices )
   ```

   , non bidearen elebazio profila *routeKms* eta *routeHeights* bektoreetan ematen den, puntu bakoitzeko abiadura *logKms* eta *logSpeeds* bektoreetan, eta integrazio zatiak *numSlices*-en. Funtzio honek bidean zehar kontsumitutako fuel kopurua bueltatu behar du.

Bidean zehar betetako distantziaren arabera integratzea aholkatzen dugu. Zati bakoitzaren hasieran abiadura, azelerazioa eta aldapa angelua kalkulatu ahal izateko:

- Kontutan eduki aldagai/parametro bakoitzean erabilitako unitateak. Ez nahastu metroak kilometroekin edota km/orduko m/s-rekin.
- Abiadura d_i distantziarekin batera edozein puntutan kalkulatu daiteke abiaduraren interpolatutako balioa d_i puntuan eskuratuz.
- Azelerazioa kalkulatzeko, d_i puntuko abiadura eta balio horatik gertu dagoen balio bat besterik ez dira behar, esaterako $d_{i-1}=d_i-0.01$.
- $p_0=\langle x_0 y_0 \rangle$ eta $p_1=\langle x_1 y_1 \rangle$ puntuen areko segmentu lerroko aldapa angelua kalkulatzeko (α *radianetan*), honako formula erabili dezakezu:

$$\alpha=\arctan\frac{(y_1-y_0)}{(x_1-x_0)} \quad .$$

3. Idatzi log bat kontsumitutako fuel kopurua kalkulatu eta gordetzen duena bide-gidari log bakoitzeko. Irteera honakoaren antzeko izan beharko litzateke:

```
Analyzing: Driver= driver1, Route= n1
Estimated fuel consumption: 8.383010 liters of fuel

Analyzing: Driver= driver2, Route= n1
Estimated fuel consumption: 8.611214 liters of fuel

Analyzing: Driver= driver1, Route= a1
Estimated fuel consumption: 9.009569 liters of fuel

Analyzing: Driver= driver2, Route= a1
```

```
Estimated fuel consumption: 9.324132 liters of fuel
```