גלאי חריגות פשוט ל time-series data
אבן דרך 1

הקדמה

סטודנטים יקרים שלום רב,

בפרויקט זה נממש אלגוריתם פשוט לגילוי חריגות ב time-series data. המטרה היא להגיע לתוצר הנדרש תוך כדי שמירה על עקרונות התכנות הפונקציונאלי.

במסמך זה נתאר את אבן הדרך הראשונה.

ההגשה היא לבודדים במערכת הבדיקות האוטומטית בקורס FP ומטלה ex1.

עליכם להגיש את Util.scala ו Line.scala

תאריך ההגשה הוא 04.04.21

בהצלחה!

אבן דרך 1 – ספריית קוד סטטיסטי לצורך גילוי חריגות

בפרויקט זה נממש גלאי חריגות (Anomaly Detector) פשוט המבוסס על שיטות סטטיסטיות פשוטות. לשם כך נצטרך להיעזר בספריית קוד שאותה נממש באבן דרך זו. זוהי בעיקר מטלת "חימום" שנועדה להרגיל אתכם בחשיבה ובתכנות פונקציונאלי. אלגוריתם לגילוי החריגות יעשה בהמשך שימוש **בחלק** מהפונקציות שמימשתם כאן.

חלק א':

1.

צרו object בשם Util ובתוכו הפונקציה max, כך שבהינתן רשימה של כל טיפוס פרמטרי A, ובהינתן פונקציה משני A-ים ל Int (בדומה ל comparator של Java), אז max תחזיר את ה A הגדול ביותר ברשימה.

לדוגמה:

```
val nums: List[Int] = List(1, 2, 3, 4)
if(Util.max(nums,(x:Int,y:Int)=>x-y)!=4)
  println("max does not return the max value  for list of ints(-10)")
```

.2

צרו ב Util פונקציה בשם map, כך שבהינתן רשימה של A, ובהינתן פונקציה מ A ל B, ובהינתן פונקציה מ B ל C, היא תמפה כל A ל C ותחזיר רשימה של C-ים. (C,B,A הם טיפוסים פרמטריים)

לדוגמה:

```
Util.map(nums,(x:Int)=>x*2,(y:Int)=>"student "+y).foreach(s=>println(s))
```

פלט:

```
student 2
student 4
student 6
student 8
```

.3

כתבו ב Util פונקציה רקורסיבית isSorted כך שבהינתן רשימה של A, ובהינתן פונקציה שבהינתן שני A-ים מחזירה בוליאני, אז isSorted תחזיר אמת או שקר בהתאמה להאם הרשימה ממוינת ע"פ הגדרות הפונקציה שקבלה כפרמטר.

לדוגמה:

```
if(!Util.isSorted(nums,(x:Int,y:Int)=>x<=y))
  println("wrong result for isSorted (-10)")
```

.4

כתבו ב Util פונקציה בשם probs כך שבהינתן מערך של double בשם xs, אז probs תחזיר מערך של double בו כל תא במקום ה i מציין את ההסתברות לראות את האיבר ה i ב xs.

לדוגמה:

```
val vs=Array(14.0,14.0,1.0,2.0) // values
val ps=Array(0.5,0.5,0.25,0.25) // probabilities
if(!Util.probs(vs).sameElements(ps))
  println("wrong probabilities returned (-5)")
```

.5

נתונה הפונקציה הבאה הנקראת אנטרופיה

$$H = -\sum_{v_i \in S} p_i \log_2 p_i$$

לכל ערך $v_i \in S$ נסכום את p_i ההסתברות לראות את v_i בתוך S, כפול $\log_2 p_i$. הוספת המינוס לפני הסכום תחזיר ערך חיובי. ערך זה מהווה מדד ל"אי הסדר" בקבוצה S.

עליכם לממש ב Util את הפונקציה entropy כך שבהינתן מערך של double, היא תחזיר את האנטרופיה שלו.

לדוגמה:

```
var xs = Array(1.0,2.0,3.0,4.0,5.0,6.0)
if(x<2.584 || x>2.585)
  println("wrong result for entropy (-5)")
```

.6

נתונה הגדרת הפונקציה לשונות סטטיסטית עבור משתנה בדיד:

בהינתן פונקציית הסתברות בדידה $x_1 \mapsto p_1, \ldots, x_n \mapsto p_n$ ניתן לחשב את ערך השונות לפי הנוסחה

$$\mathrm{Var}(X) = \sum_{i=1}^{n} p_i \cdot (x_i - \mu)^2$$

כאשר μ הוא ערך התוחלת

$$\mu = \sum_{i=1}^{n} p_i \cdot x_i$$

עליכם לממש ב Util את הפונקציות mu ו variance עבור התוחלת והשונות בהתאמה אשר בהינתן מערך של double הן מחזירות double.

לדוגמה:

```
xs=Array(1.0, 1.0, 3.0, 4.0, 4.0)
val m=Util.mu(xs)
val vari=Util.variance(xs)
if(m<4.59 || m>4.61)
  println("wrong result for mu function (-6)")
else{
  if(vari<11.167 || vari>11.169)
    println("wrong result for variance function (-3)")
}
```

.7

נתונה ההגדרה לציון Z(x): (z score)

$$Z_x = \frac{x - \mu}{\sigma}$$

כאשר μ היא התוחלת ו σ היא סטיית התקן (שורש ה variance) אז ציון ה z מודד את המרחק בין ערך x לסדרה של ערכים X ביחידות של סטיות תקן.

עליכם לממש ב Util את zscore עבור ציון Z.

לדוגמה:

```
xs=Array(1.0,1.0,3.0,4.0,4.0)
val z=Util.zscore(xs,3.0)
if(z<(-0.478)-0.001 || z>(-0.478)+0.001)
  println("wrong result for z score function (-5)")
```

.8

נתונה הפונקציה cov שצריכה להחזיר את השונות המשותפת (covariance) של המשתנים X ו Y.

$$cov(X,Y) = E(XY) - E(X)E(Y) = E\big((X - E(X))(Y - E(Y))\big)$$

כאשר $E(X)$ היא התוחלת המחושבת כמו μ לעיל.

הפונקציה Pearson היא מדד לקורלציה ליניארית. ככל שהערך קרוב יותר ל 1 כך שני המשתנים מתנהגים דומה יותר (כשהאחד עולה או יורד השני עולה או יורד בהתאמה). ככל שהערך קרוב יותר ל 1- כך המשתנים מתנהגים הפוך זה מזה (כשהאחד עולה, השני יורד). בשני המקרים מדובר בקורלציה מאד חזקה. לעומת זאת, ככל שהערך קרוב יותר ל 0 כך גובר חוסר הקשר ביו המשתנים.

יש לממש ב Util את Pearson כך:

$$\frac{cov(X,Y)}{\sigma_x \sigma_y}$$

כאשר σ היא סטיית התקן (השורש של variance)

לדוגמה:

```
xs=Array(1.0,2.0,3.0,4.0,5.0)
ys=Array(3.0,6.0,9.0,12.0,15.0)
if(Util.pearson(xs,ys)<1.0-0.0001 )// result should be 1 or very close to 1
  println("wrong result for pearson function (-5)")
```

אילוצים:

- על כל הפונקציות לעיל להיות כתובות בצורה פונקציונאלית טהורה – כלומר ללא תופעות לוואי
- אין להשתמש בלולאות!
  - מותר להשתמש ברקורסיה
  - מותר ואף מומלץ להשתמש בפונקציות של הספרייה הסטנדרטית של סקאלה.

טיפ: מהפונקציה probs ואילך המימוש של כל פונקציה יכול להיעשות בשורת קוד אחת.

חלק ב':

כעת בקובץ Point.scala נתונה לכם המחלקה Point המייצגת נקודה דו-ממדית.

עליכם לממש בקובץ Line.scala את המחלקה Line, כך שבהינתן (בבנאי) מערך של נקודות, היא תחזיק את משוואת הישר שחושבה ע"פ רגרסיה ליניארית:

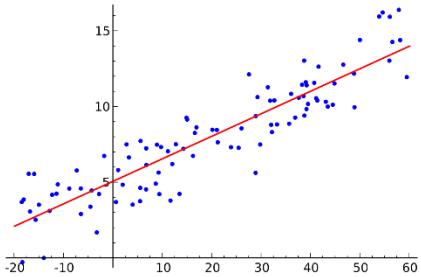
משוואת ישר מוגדרת ע"י $Y = a \cdot X + b$.

רגרסיה לינארית:

$a = \frac{COV(x,y)}{VAR(x)}$ ואילו $b = \bar{y} - a\bar{x}$ ($\bar{x}$ ו $\bar{y}$ הם הממוצעים של X ו Y)

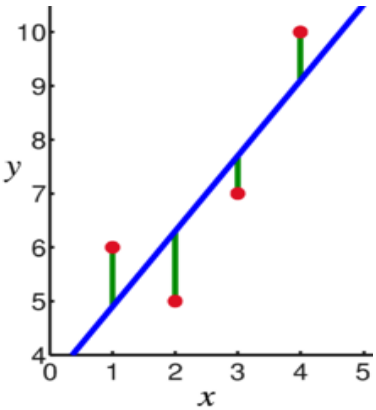
כך מתקבלת משוואת ישר $Y = a \cdot X + b$

למשל:



לקריאה נוספת (ולאפשרויות חישוב נוספות):

https://en.wikipedia.org/wiki/Simple_linear_regression



כעת, שתי הפונקציות האחרונות בספרייה יעזרו לנו למדוד את הסטייה בין נק' כלשהי, לבין שאר הנקודות בהתפלגות. יש המון דרכים למדוד זאת. אנו נבחר בדרך פשוטה והיא המרחק בין הנק' לבין הישר שיצרו הנקודות. ראו את התרשים הבא. לכל נק' *(x,y)* יש את את הנק' המתארת היכן ציפינו שהיא תהיה על הישר *(x,f(x))* ואילו ההפרש בערך מוחלט $|f(x)-y|$ מתאר לנו את המרחק מהצפייה הזו. בתרשים אלו הקווים הירוקים. ככל שאורך הקו הירוק גדול יותר כך הנק' נראת לנו חריגה יותר ביחס לכל האחרות.

(הערת אגב: אין צורך למדוד את אורך הקו בין נק' אדומה וניצב לקו הכחול כי בגלל כלל פיתגורס החישוב הפשוט שלנו מספיק כדי לתאר עד כמה רחוקה הנק' האדומה מהקו הכחול)

לשם כך עליכם לממש את

- המשתנים *a,b* עבור משוואת הישר. המשתנים צריכים להיות *read only*.
- המתודה *f(x)* אשר בהינתן *x* היא תחשב את ערך ה *y* ע"פ משוואת הישר.
- המתודה *dist* אשר בהינתן נקודה היא תחשב את המרחק שלה מהישר.

דוגמה:

```
val pnts=Array(new Point(0,0.1),new Point(1,2.01),new Point(5.1,10))
val l=new Line(pnts)
println("A="+l.a+" B="+l.b) // A=1.94 B=0.085
println(l.f(4)) // 7.85
println(l.dist(new Point(4,8))) // 0.14
```

בהצלחה!