

הדגמות אינטראקטיביות של מודלים סטטיסטיים סטוכסטיים
החוג לסטטיסטיקה, אוניברסיטת חיפה,
סמסטר חורף – תשס"ט
מרצה: יוני נצרתי

דוגמא למבחן ביניים מס' 1:

משך המבחן, שעתיים וחצי.
יש לענות על כל השאלות.
חומר עזר אסור – מצורף עזר לפקודות.

חלק א: עבור כל סעיף כתוב מהו הפלט.

(1א)

```
Fold[Plus, 0, NestList[(#+1) &, 0, 5]]
```

פתרון:

(2א)

```
Map[Apply[Plus, #] &, Table[Table[i, {i, j}], {j, 5}]]//Last
```

פתרון:

(3א)

```
Select[Map[2 # &, Table[i, {i, 1000}]], OddQ]~Join~{"lala"}
```

פתרון:

חלק ב: עבור כל סעיף רשום תשובה קצרה (רק את מה שיש להזין ל Mathematica)
(1)

הגדר את הפונקציה

`index[lst_, element_]`

פונקציה זו מקבלת רשימה של איברים (lst) ומחזירה את האינדקס הראשון אשר בו מופיע האיבר element. במידה וelement אינו ב lst אז הפונקציה מחזירה -1.

לדוגמא:

מחזירה 2 `index[{a,b, {}, 12}, b]`

מחזירה -1 `index[{a,b, {}, 12}, {12}]`

פתרון:

(2ב)

הגדר את הפונקציה

`equalLength[lst_]`

פונקציה זו מקבלת רשימה מקוננת ומחזירה `True` במידה וכל האיברים ברשימה הינם בעלי אותו אורך ומחזירה `False` אחרת.

לדוגמא:

`equalLength[{{1}, {1, 2}}]` מחזירה `False`.

`equalLength[{{1}, {2}, {3}}]` מחזירה `True`.

פתרון:

(3ב)

נתונה רשימה של מספרים xx ורשימה של מספרים yy , שתיהן בעלות אורך זהה.
ברצוננו תצייר "ענן" של הנקודות,
 $\{xx[[1]], yy[[1]]\}$
 $\{xx[[2]], yy[[2]]\}$ וכו'.
כתוב קוד אשר מבצע זאת.

פתרון:

In[28]:= ? **Fold**

`Fold[f, x, list]` gives the last element of `FoldList[f, x, list]`. >>

In[29]:= ? **FoldList**

`FoldList[f, x, {a, b, ...}]` gives `{x, f[x, a], f[f[x, a], b], ...}`. >>

In[30]:= ? **Nest**

`Nest[f, expr, n]` gives an expression with `f` applied `n` times to `expr`. >>

In[31]:= ? **NestList**

`NestList[f, expr, n]` gives a list of the results of applying `f` to `expr` 0 through `n` times. >>

In[32]:= ? **If**

`If[condition, t, f]` gives `t` if `condition` evaluates to True, and `f` if it evaluates to False.

`If[condition, t, f, u]` gives `u` if `condition` evaluates to neither True nor False. >>

In[33]:= ? **Select**

`Select[list, crit]` picks out all elements e_i of `list` for which `crit[ei]` is True.

`Select[list, crit, n]` picks out the first `n` elements for which `crit[ei]` is True. >>

In[34]:= ? Plus

$x + y + z$ represents a sum of terms. >>

In[35]:= ? Apply

Apply[f , $expr$] or $f @@ expr$ replaces the head of $expr$ by f .

Apply[f , $expr$, $levelspec$] replaces heads in parts of $expr$ specified by $levelspec$. >>

In[36]:= ? Map

Map[f , $expr$] or $f /@ expr$ applies f to each element on the first level in $expr$.

Map[f , $expr$, $levelspec$] applies f to parts of $expr$ specified by $levelspec$. >>

In[37]:= ? Table

Table[$expr$, { i_{max} }] generates a list of i_{max} copies of $expr$.

Table[$expr$, { i , i_{max} }] generates a list of the values of $expr$ when i runs from 1 to i_{max} .

Table[$expr$, { i , i_{min} , i_{max} }] starts with $i = i_{min}$.

Table[$expr$, { i , i_{min} , i_{max} , di }] uses steps di .

Table[$expr$, { i , { i_1 , i_2 , ...}}] uses the successive values i_1 , i_2 , ...

Table[$expr$, { i , i_{min} , i_{max} }, { j , j_{min} , j_{max} }, ...]

gives a nested list. The list associated with i is outermost. >>

In[38]:= ? OddQ

OddQ[$expr$] gives True if $expr$ is an odd integer, and False otherwise. >>

In[39]:= ? Last

Last[$expr$] gives the last element in $expr$. >>

In[40]:= ? Join

Join[$list_1$, $list_2$, ...] concatenates lists or other expressions that share the same head.

Join[$list_1$, $list_2$, ..., n] joins the objects at level n in each of the $list_i$. >>

In[41]:= ? ListPlot

ListPlot[{ y_1 , y_2 , ...}] plots points corresponding

to a list of values, assumed to correspond to x coordinates 1, 2, ...

ListPlot[{{ x_1 , y_1 }, { x_2 , y_2 }, ...}] plots a list of points with specified x and y coordinates.

ListPlot[{ $list_1$, $list_2$, ...}] plots several lists of points. >>