מבוא

במסגרת מטלה זו יצרנו ניסוי חיפוש חזותי בסיסי בפסיכולוגיה קוגנטיבית אשר הוצג לראשונה על ידי החוקרת אן טריזמן. מטרת הניסוי היא בחינת ההבדלים באופן החיפוש החזותי עבור תנאים שונים (גודל סט , סוג גירוי וקיום/היעדר מטרה). על מנת לבחון נושא זה, יצרנו ניסוי בתוכנת Matlab המדמה את המטלה המקורית של טריזמן. בהמשך ניתחנו את התוצאות שהתקבלו מנבדק אחד והסברנו כיצד מתיישבות עם התיאוריות בנושא זה. נציין כי בדיקת זמן התגובה במטלה זו מתבססת על ההנחה כי הבדלים בזמן תגובה משקפים פעילות קשבית שונה.

שיטה:

על מנת ליצור מטלת חיפוש חזותית כפי שהיתבקשנו, כתבנו קוד מתאים בתוכנת Matlab. בשורות הבאות נפרט ונסביר קטעים עיקריים בקוד זה.

```
Set_Size = [4 8 12 16];
Color = {'r' 'b'};
Elements =  { 'X' 'O' };
Exp_Type = ['f' 'c'];                                   % 1- feature search, 2-
conjunction search
keys = {'A' 'L'};                                       % A - target present, L -
target absent
N_Repeats = 30;

%Results settings
T_Max_filter = 3.75;                                    %max response time in seconds
Min_Trials_per_Block = 20;
```

בהגדרות הניסוי דאגנו לייצר משתנים כך שניתן יהיה להגדיר אותם באופן שונה עבור ניסוים שונים. המשך הקוד לוקח את מרבית המידע שלו מחלק זה ובנוי כך שגודל הסט יכול להיות שונה, הצבעים שונים (גם יותר מ2), הצורות והמקשים.

כמו כן, הגדרנו גם מספר חזרות רצוי עבור כל בלוק (N_Reapeats).

בנוסף, על מנת למנוע P-Hacking, דאגנו להכניס את קריטריון הניפוי לתוצאות טרם הפעלת הניסוי (שורה 18). T_Max_Filter - קובע את הזמן בשניות אשר מעליו תוצאות ינופו (זמנים ארוכים מדי) וMin_Trails_per_Block - קובע את מספר החזרות הנכונות והמהירות מספיק המינימלי על מנת להמשיך לביצוע ניתוח. בהמשך הקוד, לאחר הרצת הניסוי, במידה ולא יהיו מספיק תוצאות תקינות תצא הודעת שגיאה.

```
run_mode = 0;                                           %0- human, 1-robot
time_frame = 0.3 + 2;                                   %set average time response
```

בהמשך, יצרנו משתנה בשם run_mode שיקבע האם הקוד יופעל על ידי משתמש אנושי או האם הקוד ירוץ באופן רובוטי. בהמשך הקוד חלקים רלוונטים יופעלו ע״פ הגדרת משתנה זה. כמו כן, הגדרנו את המשתנה time_frame ע״מ להגדיר את טווח זמן התגובה במצב הרובוט.

בשלב הבא מוגדר הfigure הראשי של הניסוי ומופיעים מסך הפתיחה וההוראות. מצב החוזר על עצמו במטלה הוא לחיצה על מקש רווח על מנת להתקדם למסך הבא. בכדי לגשת לבעיה זו באופן יעיל יצרנו את הפונקציה Press2Continue

```
function [] = Press2Continue(Screen,Key2Continue,run_mode)
%this function moves to next screen when the correct key is pressed.
%Screen - the screen that we record the key on
%Key2Continue - the only key that we want to make us move to the next
%screen.

if run_mode == 0
    pause(); Pressed_key = get(Screen,'CurrentCharacter');

    while strcmpi(Pressed_key,Key2Continue) == 0
        pause(); Pressed_key = get(Screen,'CurrentCharacter');
    end

    clf;
    axis off

elseif run_mode == 1
    time_sti = 2*rand();                        %random time according to settings.
    pause(time_sti);

    clf;
    axis off
end
```

פונקציה זו שיצרנו מאפשרת שימוש יעיל ונוח במקש ״המשך״ לאורך הניסוי.

הפונקציה מחולקת לשני מצבים (אנושי/רובוטי) ועל המשתמש להכניס את המקש שעבור לחיצה עליו הניסוי ימשיך. במצב האנושי - הפונקציה משווה את המקש שלחץ המשתמש למקש הרצוי. במידה והמקש לא מתאים הפונקציה חוזרת להתחלה (while). במצב הרובוטי - הפונקציה בוחרת זמן השהיה אקראי (כדי שהמסך לא יעבור מהר מדי) ופשוט עוברת הלאה.

הכנות לניסוי

```
function [RandBlocks] = rand_comb(Set_Size,Exp_Type,Elements)
%This function preduce all the possible blocks without reapets in a random
%sort.
%exp_type and Target_Elment are converted to numbers according to their
%index
%Set size - vector with set size as numbers, each number represents the number
%of stimulus in the current experiment.
%Exp_Type - the input can be numbers or strings, each number represents a diffrent
type of
%experiment
%Elements - the input can be numbers or strings, each number represents a diffrent
target
%elment.

Exp_Type = 1:length(Exp_Type);

All_Blocks = combvec(Set_Size,Exp_Type);
cols = size(All_Blocks,2);
P = randperm(cols);
RandBlocks = All_Blocks(:,P);

Elements = 1:length(Elements);

TRG = [];                                                   %TRG - Target Element
Vector
for i = 1:floor(length(RandBlocks)/length(Elements))
    TRG = [TRG Elements];
end

TRG = TRG(randperm(length(TRG)));

RandBlocks(3,:) = TRG;
end
```

בשלב הבא, ועל מנת שתתבצע חלוקה רנדומלית של מאפיינים לכל בלוק בניסוי, יצרנו פונקציה בשם rand_comb. פונקציה זו מקבלת כקלט שלושה משתנים שיצרנו בתחילת הקוד : Set_Size (גודל הסט), Exp_Type (סוג המטלה) ו-Elements (צורת המטרה שעל הנבדק לחפש). ראשית, הפונקציה יוצרת את כל הקומבינציות האפשריות בין גדלי הסט האפשריים וסוגי המטלות האפשריות (על ידי שימוש בפונקציה combvec), ומכניסה את הקומבינציות למשתנה All_Blocks (כל עמודה במטריצה זו מכילה מאפיינים של בלוק אחד). בהמשך, השתמשנו בפונקציות size ו- randperm ע״מ לערבב את עמודותיו של המשתנה All_Blocks, ובאופן זה ליצור רנדומליות בחלוקת המאפיינים לבלוקים. הגדרנו את המטריצה המעורבבת תחת השם RandBlocks. בשלב הבא, השתמשנו בוקטור Target_Element (המכיל את צורות המטרות האפשריות), אותו הגדרנו בתחילת הקוד, על מנת להוסיף למשתנה RandBlocks את צורת המטרה שתהיה בכל בלוק. תחילה המרנו את וקטור Target_Element, המכיל אותיות (X ו- O), לכזה שיכיל מספרים (1 ו-2) שייצגו את האותיות. לאחר מכן יצרנו משתנה ריק בשם TRG. בשלב

הבא מתבצע שירשור של הוקטור Target_Element אל תוך המשתנה TRG, כך שלבסוף המשתנה TRG יהיה וקטור באורך מספר הבלוקים במטלה (8 במקרה זה), המכיל את שני סוגי המטרות האפשריות באופן מאוזן. לאחר מכן, יתבצע ערבוב של וקטור זה (שימוש בפונקציה randperm), על מנת שהחלוקה תהיה רנדומלית. לבסוף הוקטור ייכנס למשתנה RandBlocks. מטריצה זו היא הפלט של הפונקציה, ומכילה מספר עמודות כמספר הבלוקים בניסוי. בכל עמודה מאפייני הבלוק: סוג , מספר הפריטים , וסוג המטרה. בקוד הראשי, משתנה זה נקרא Rand_Blocks.

אופן אחסון המידע:

כלל הניסוי נשמר תחת המשתנה data. כבר בשורה 49, טרם תחילת הניסוי, אנו מכינים את הזיכרון לשמירת התוצאות ומגדירים משתנה זה כcell array. בחרנו במבנה של 8 תאים (תא עבור כל בלוק) ובכל תא תימצא מטריצה בגודל מספר החזרות(בכל בלוק) * 5.

1x8 cell

	1	2	3	4	5	6	7	8
1	30x5 double	30x5 double	30x5 double	30x5 double	30x5 double	30x5 double	30x5 double	30x5 double

כל שורה בכל מטריצה מייצגת חזרה אחת והטורים מייצגים את כלל המידע שנזדקק לו בשלב ניתוח הנתונים. המידע המאוחסן לפי טורים הוא כדלקמן: זמן תגובה (בשניות), האם התשובה נכונה? (1/0), מה גודל הסט? (4/8/12/16), האם המטרה הופיעה? (1/0) והאם מדובר בניסוי מסוג feature search-1 או conjunction search-2.

data{1, 2}

	1	2	3	4	5
1	1.1843	0	8	0	2
2	10.4030	1	8	1	2
3	2.4093	1	8	0	2
4	2.4062	1	8	0	2
5	2.3188	1	8	1	2
6	2.5371	1	8	1	2

לדוגמא, עבור בלוק מספר 2, בחזרה הראשונה (שורה ראשונה), ניתן לראות כי זמן התגובה היה 1.1 שניות, הנבדק טעה, גודל הסט הוא 8 צורות, המטרה איננה הופיעה וסוג הניסוי הוא conjunction search.

הניסוי

```
for N_Block = 1 :length(Rand_Blocks)
    data{N_Block} = zeros(N_Repeats,5);                    %sets the matrix in the cell
for block data.
                                                           %5 = RT, ACC,Setsize,Target
present,Exp_Type.

    %Preparing Block
    N_Symbols = Rand_Blocks(1,N_Block);                                %set number of
symbols.
    Target = Elements(Rand_Blocks(3,N_Block));                         %choose target
symbol.

    %chosing distractor (choose one symbol from elments not included
    %target symbol.
    Block_Elements = Elements;
    Block_Elements((Rand_Blocks(3,N_Block))) = [];
    Distractor = Block_Elements(randi(length(Block_Elements)));

    %storing data
    data{N_Block}(:,5) = Rand_Blocks(2,N_Block);
    data{N_Block}(:,3) = N_Symbols;

    Trials = Trials_gen(N_Repeats,N_Symbols,Color);                    %generate
trials for the block.
```

בשלב זה יצרנו את הלוף המרכזי שמפעיל את המטלה. ראשית יצרנו לולאת for שתרוץ על מספרי הבלוקים. בשלב הבא תיווצר ב-cell המתאים במשתנה data מטריצת אפסים שתשמש לשמירת המידע הנחוץ אודות ביצוע הניסוי לטובת ניתוח הנתונים.

לאחר מכן יוגדרו המשתנים N_Symbols (מספר הפריטים בבלוק) ו- Target (צורת המטרה בבלוק) על ידי שליפתם מהמקום המתאים במטריצה Rand_Blocks. בנוסף, יוגדר המשתנה Distractor - משתנה זה יכיל את צורת המסיחים בבלוק הנוכחי. את הגדרתו ביצענו על ידי מחיקת המשתנה Target מהוקטור Elements (כאמור וקטור זה מכיל את צורות הגירויים האפשריות בניסוי). לאחר המחיקה מתבצעת הגרלה בוקטור Elements (ע"י שימוש בפוקנציה randi שורה 67) שבה תקבע צורתם של המסיחים (מצב זה מותאם למקרה בו וקטור Elements מכיל יותר מ-2 צורות).

בשלב הבא מאפייני הבלוק הרלוונטים ישמרו במקומות המתאימים במשתנה data. כך למעשה טיפלנו במאפיינים שקבועים לכל אורך הבלוק. כעת נסביר כיצד התמודדנו עם המאפיינים המשתנים בין trails.

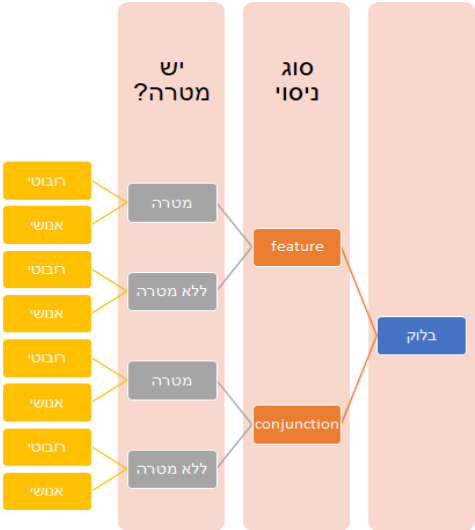
```
function [Trials] = Trials_gen(N_Repeats,N_Symbols,Color)
%this function creats a cell array with the locations of all symbols for
%all repeations. also included color randomization and equal number of
%trials with and without target.
```

```
Trials = cell(4, N_Repeats);

%locations without overlap
for i = 1:2
    for j = 1: N_Repeats
    Trials{i,j} = [0.001 * randperm(1000, N_Symbols)];
    end
end

%color randomly
for j = 1:N_Repeats
    Trials{3,j} = Color(randi(length(Color)));
end

%with or without target equaly.
Trials(4,:) = {0};
replace_position = randperm (N_Repeats,N_Repeats/2);
Trials(4,replace_position) = {1};

end
```

הפונקציה Trails_gen מייצרת את החזרות עבור כל בלוק. ראשית, הפונקצייה מכינה זיכרון בגודל מתאים למספר החזרות בבלוק ובעל 4 שורות וזאת באמצעות cell array על מנת להכיל רכיבים מסוגים שונים עליהם נרחיב ונפרט. בלולאה בשורות 8-13 מתבצעת הקצאה של מיקומים עבור הגירוים על גבי המסך. השורה הראשונה (בTrails) מייצגת את וקטור הקורדינאטות לפי ציר הX והשורה השניה את וקטור הקורדינאטות לפי ציר Y. באמצעות הפונקציה randperm אנו מוודאים כי שני גירוים לא יקבלו את אותם הקורדינאטות. בשורות 15-18 הפונקציה מקצה צבע רנדומלי מתוך רשימת הצבעים המוגדרת בתחילת הניסוי (שורה 11 בקוד הראשי) - חשוב לציין כי מבנה זה שיצרנו מאפשר הכנסת מספר צבעים גדול מ2 וזאת על מנת לייצר גמישות בקוד. בשורות 20-23 (ובשורה ה4 במשתנה Trails) מתרחשת הקצאה אקראית אך שווה עבור חזרות עם וללא מטרה. עשינו זאת באמצעות: ראשית, הקצאה של "0" (ללא מטרה) עבור כל החזרות ואז החלפה של מחצית מהתאים במספר "1" (יש מטרה) וזאת באופן אקראי.

5	6
[0.8690,0.2780,0.1760,0.4200]	[0.5490,0.8610,0.4350,0.1340]
[0.7120,0.3870,0.3030,0.2250]	[0.3710,0.4550,0.7800,0.0300]
'r'	'b'
0	1

כאן ניתן לראות דוגמא מתוך המטריצה המייצרת Trail_gen - עבור החזרה החמישית (טור חמישי), ניתן לראות את מיקומי הגירוים לפי ציר X (שורה 1), את מיקומי הגיורים לפי ציר Y (שורה 2) (נשים לב גם כי אורך כל וקטור הוא 4 וזאת מאחר וגודל הסט כאן הוא 4) צבע המטרה המוקצה - אדום (שורה 3) וכי מדובר בחזרה ללא הופעת מטרה.



בתרשים זה מתואר באופן סכמטי הלופ המרכזי של המטלה. מאחר והתנאים נבדלים זה מזה בפרטים בודדים, נסביר בפירוט על ענף אחד ונפרט על ההבדלים בין הענפים.

נציין כי הבחירה בענפים מוגדרת בהתאם לRand_Blocks, Trails & run_mode המפורטות מוקדם יותר במסמך.

```
%(1)Feature Search
if Rand_Blocks(2,N_Block) == 1

    for t = 1:N_Repeats

        Set_Color = cell2mat(Trials{3,t});

        %No Target

        if Trials{4,t} == 0
            text (Trials{1,t},Trials{2,t},Distractor,'Color',Set_Color);
            tic;
```

נפרט עבור ניסוי מסוג feature, ללא מטרה: עבור חזרה "ללא מטרה" - מופיעים הגיורים בהתאם לקורדניאטות הX והY מהמשתנה Trails עליו הסברנו קודם לכן, כלל הגיורים מקבלים את הelement של המסיח, והצבע המוגדר הוא בהתאם לצבע בSet_Color אשר מנגיש את המידע מהמשתנה Trails.

```
        %with Target
        elseif Trials{4,t} == 1

            %the (n-1) first symbols location will be the distractors,
            %and the (n) symbol will be the target
            text (Trials{1,t}(1:(N_Symbols-1)),Trials{2,t}(1:(N_Symbols-1)),...
            Distractor,'Color',Set_Color);
            text (Trials{1,t}(N_Symbols),Trials{2,t}(N_Symbols),Target,...
```

```
                'Color',Set_Color);

                tic;
```

עבור חזרה "עם מטרה" - נשתמש ב2 פונקציות text: הראשונה מתייחסת לכל הצורות מהראשונה עד לאחת לפני האחרונה ומייצרת מהם מסיחים בהתאם להגדרות הראשוניות והשניה מייצרת את הגירוי האחרון כגירוי מטרה.

```
                if run_mode == 0                                    %Human mode
                    pause(); key = get(h,'CurrentCharacter');

                elseif run_mode == 1                                %Robot mode
                    time_sti = time_frame*rand();                   %random time
according to settings.
                    key = keys(randi(length(keys)));                %random key.
                    pause(time_sti);
                end
```

עבור מצב מענה "אנושי" - הקוד מחכה לתגובת הנבדק ושומר אותה.

עבור מצב מענה "רובוטי" - הקוד בוחר באקראי זמן השהייה ותשובה.

```
                %store data
                data{N_Block}(t,1) = toc;
                data{N_Block}(t,2) = strcmpi(key,keys{1});
                data{N_Block}(t,4) = Trials{4,t};
```

בהמשך (עבור כל אחד מהתנאים), המידע נשמר ומאוחסן בdata בהתאם להגדרות. נשים לב כי המידע המאוחסן בשורה 106 מתייחס לנכונות התשובה ולא למקש שנלחץ ובהתאם לכך משתנה במעט עבור תנאי עם מטרה וללא מטרה (הרכיב האחרון בשורה 106 ניגש לוקטור המקשים ובמקרה זה בוחר את המקש במקום השני אשר מתייחס להיעדר מטרה. במצב של מטרה במקום 2 יופיע 1).

כעת עבור conjunction search נציין רק את החלקים השונים:

```
        %(2) Conjunction Search
    elseif Rand_Blocks(2,N_Block) == 2

        for t = 1:N_Repeats

            %setting colors
            Set_Color = cell2mat(Trials{3,t});
```

```
            dis_Color = cell2mat(Color(randi(length(Color))));          %choose random
color.

            %if the color that have been chosen is the same as the first one the code
will try to choose
            %color once again until a different one is chosen.
            while strcmp(Set_Color,dis_Color) == 1
                dis_Color = cell2mat(Color(randi(length(Color))));
            end
```

ראשית הקוד מנגיש את צבע המטרה מתוך Trials והפעם ממשיך בבחירת צבע שונה עבור המסיח (שורות 150-156) -נשים לב כי הקוד מוגדר בצורה זו על מנת לייצר גמישות עתידית להכנסת מספר צבעים שונים.

נדגים עבור תנאי עם מטרה :

```
                %with target
            elseif Trials{4,t} == 1
                Dis_num1 = [1:(N_Symbols/2)];
                Dis_num2 = [(N_Symbols/2)+1:(N_Symbols-1)];
                target_num3 = [N_Symbols];

                %1st color distractors
                text
(Trials{1,t}(Dis_num1),Trials{2,t}(Dis_num1),Distractor,'Color',Set_Color);

                %2nd color distractors
                text
(Trials{1,t}(Dis_num2),Trials{2,t}(Dis_num2),Target,'Color',dis_Color);

                %target
                text
(Trials{1,t}(target_num3),Trials{2,t}(target_num3),Target,'Color',Set_Color);
```

בשורות 193-195 מוגדרים האינדקסים של הגירויים המיועדים להופיע בשלושת פונקציות הטקסט(197-204) בהתאמה. הקבוצה הראשונה מוגדרת כחצי הראשון של הקבוצה (193) והיא תופיע כמסיח בצבע התואם לצבע המטרה (198). בצורה דומה מוגדרת הקבוצה השנייה של הגירויים אשר תהווה את החצי הנותר של הקבוצה לא כולל הגירוי האחרון, תוגדר להופיע בהתאם לצורת המטרה אך בצבע שאינו תואם (dis_Color - בהתאם להגדרה בקטע הקוד הקודם). הקבוצה השלישית מוגדרת כגירוי האחרון בקבוצה ומאופיינת על ידי צורת וצבע המטרה המוגדרים.

ההבדל המהותי היחידי ב conjunction search ללא מטרה הוא צמצום קבוצות ההגדרה ל2 קבוצות שוות בגודלן. האחת מכילה צורות של מסיחים עם צבעים שאינם תואמים לצבע המטרה והקבוצה השניה מכילה צורות של מטרות עם צבעי "מסיחים" - במצב כזה אין גירוי יוצא דופן ולכן אין מטרה.

אופן שמירת הנתונים והמעבר בין מצבי עבודה (אנושי/רובוטי) זהים בשני סוגי הניסוי ולכן לא נחזור עליהם.

נציין כי בסיום הניסוי שמרנו את כלל הנתונים הגולמים שהצטברו בdata לקובץ בשם "data" מסוג mat.

שלב סינון התוצאות:

בשלב סינון התוצאות, יצרנו את המשתנה filter_data. משתנה זה מכיל את תוצאות הניסוי ועליו מתבצעים הניתוחים.

בשלב זה נמחקות החזרות אשר במסגרתן תשובת הנבדק הייתה שגויה ו/או ארוכה מזמן המקסימום שהגדרנו וזאת באמצעות לולאות פשוטות עליהן לא נפרט.

מספר החזרות התקינות מופיע ב Valid Trails ובמידה ומספר החזרות התקינות באחד מהבלוקים קטן מהמספר שהגדרנו כמינימלי לניתוח בתחילת הקוד תצא הודעת שגיאה מתאימה.

ניתוח התוצאות וגרפים:

PLOT1

4x8 double

	1	2	3	4	5	6	7	8
1	2.2667	2.5392	3.0122	1.9097	1.7044	3.1743	0.9986	1.8527
2	0.3378	0.4639	0.5304	0.6085	0.5169	0.4217	0.2525	0.7236
3	16	8	12	12	8	16	4	4
4	1	2	2	1	1	2	1	2

לאחר סינון התוצאות, התחלנו בניתוח התוצאות ע"פ מאפייניהן. מטרתנו הייתה לסווג את התוצאות לפי המאפיינים הרלוונטיים (עם/בלי מטרה, conjunction/feature), ע"מ שנוכל ליצור גרף נפרד לכל מאפיין. לשם כך, פיצלנו את המשתנה filter_data לשני משתנים- plot1, שמכיל את הנתונים השייכים ל-trails ללא מטרה, ו- plot2, שמכיל את הנתונים השייכים לכאלה עם מטרה. ניתן לראות לדוגמא בטבלה המצורפת למעלה, את המטריצה plot1: השורה הראשונה מכילה את זמן התגובה הממוצע (שימוש בפונקציית mean) של הנבדק בכל בלוק, בtrails בהם לא היתה מטרה בלבד. השורה השנייה מכילה את סטיות התקן של זמני התגובה (שימוש בפונקציית std) בכל בלוק ב-trails בהם לא הופיעה מטרה , השורה השלישית את גודל הסט באותו בלוק, והשורה הרביעית את סוג הסט (conjunction/feature). המשתנה plot2 בעל מבנה זהה, ומכיל את אותם נתונים עבור ה-trails שבהם הופיעה מטרה.

Plot_C_A

4x4 double

	1	2	3	4
1	1.8527	0.7236	4	2
2	2.5392	0.4639	8	2
3	3.0122	0.5304	12	2
4	3.1743	0.4217	16	2

לאחר מכן נשאר לנו לברור בין התוצאות שהתקבלו ע״פ סוג הניסוי (conjunction/feature). לשם כך פיצלנו את המשתנים plot1 ו-plot2 על פי מאפיין זה. לאחר ביצוע הפיצול נוצרו 4 מטריצות: plot_F_A , plot_C_A , plot_F_P , plot_C_P (מקרא: F=feature, C=conjunction, A=absence, P=present). משתנים אלה מכילים את המידע המסווג, ע״פ סוג הסט וע״פ קיומה/העדרה של מטרה. לבסוף השתמשו בפונקציה sortrows ע״מ לסדר את המטריצות שהתקבלו בסדר עולה ע״פ העמודה השלישית שמכילה את המידע על גודל הסט, זאת ע״מ לאפשר עבודה קלה יותר בתהליך יצירת הגרפים. מצורף למעלה המשתנה plot_C_A כדוגמא: העמודה הראשונה מכילה את ממוצע זמני התגובה, השניה את סטיית התקן של זמני התגובה, השלישית את גודל הסט, והרביעית את סוג הניסוי. כל המידע מתייחס למקרים בהם סוג הניסוי היה conjunction ולא הוצגה מטרה.

אופן יצירת הגרפים זהה בעיקרו עבור כלל התנאים ולכן לא נחזור עליו. בחרנו להציג את הגרפים בחלונית אחת המחולקת לשתי מערכות צירים באמצעות הפונקציה subplot, אחד עבור תנאי מטרה והשני עבור תנאי ללא מטרה. בכל מערכת יופיעו שני גרפים עבור כל סוג ניסוי: גרף תוצאות וקו ניבוי (סה״כ 8 עקומות, 4 בכל מערכת צירים ו2 עבור כל תנאי ניסוי).

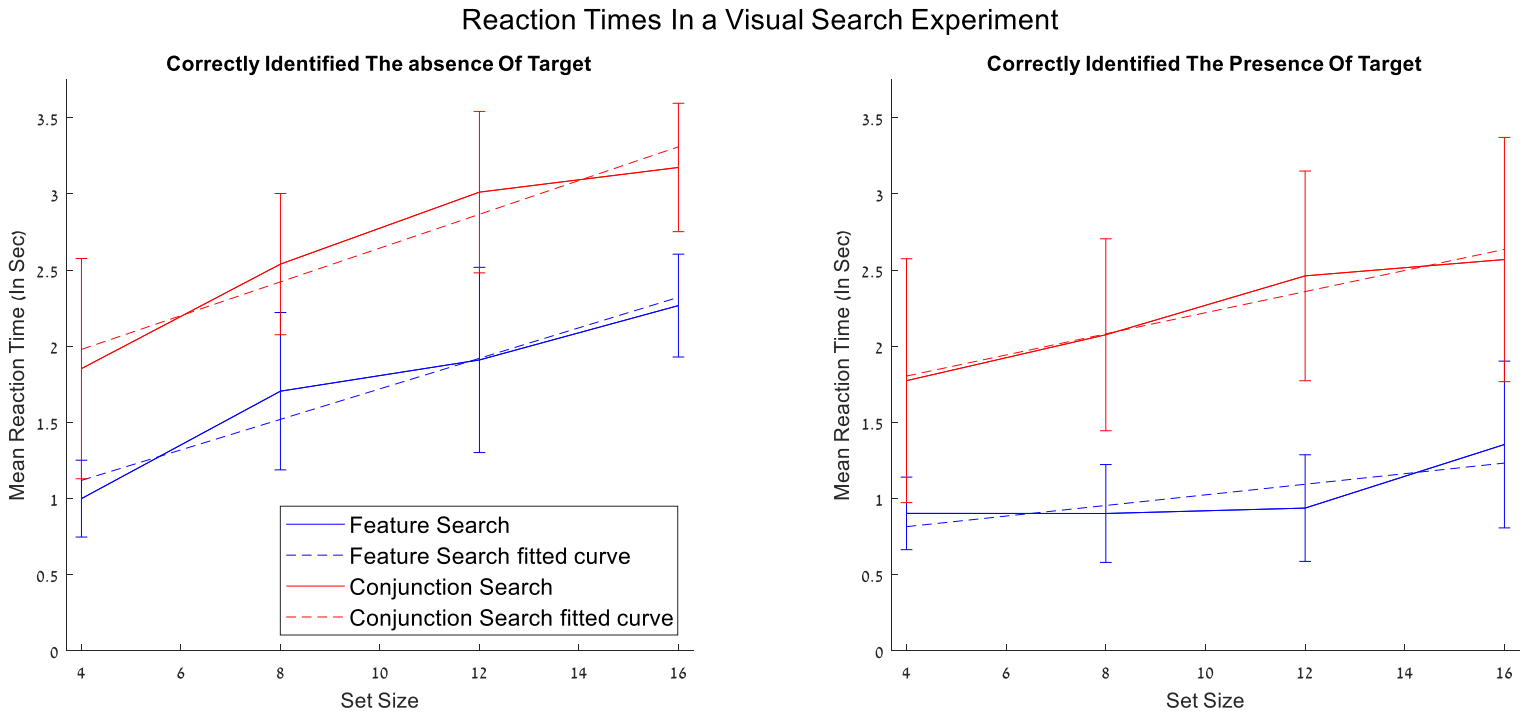
תוצאות:

תחילה, נציג מספר נתונים בסיסיים לגבי מאפייניו של כל בלוק. בטבלה המצורפת למטה, ניתן לראות את סוג הסט וגודלו בכל בלוק, ובנוסף את מספר ה-trails שנשארו בכל בלוק לאחר סינון התוצאות, ועליהם בוצעו הניתוחים. ניתן לראות שגדלי הסטים בין הבלוקים מאוזנים (כל גודל סט נבחן פעמיים, פעם אחת לכל סוג סט) כמו גם סוגי הסטים (4 פעמים כל סוג סט). בנוסף ניתן לראות שמספר הtrails התקינים בכל בלוק גבוה מ-20 (Min_Trials_per_Bloc).

מספר בלוק	1	2	3	4	5	6	7	8
סוג הסט	feature	conjunction	conjunction	feature	feature	conjunction	feature	conjunction
גודל הסט	16	8	12	12	8	16	4	4
מספר trails תקינים	29	25	21	30	28	24	30	20

בשתי מערכות הצירים המצורפות למטה מוצגות תוצאות ניסוי החיפוש הויזואלי. בשתיהן ציר ה-X מייצג את גודל הסט, כלומר, מספר הפריטים שהוצגו בסט. נבדקו זמני התגובה לסטים בגודל 4,8,12,16. ציר ה-Y מייצג את זמן התגובה הממוצע של הנבדק בשניות. כל מערכות הצירים

מתייחסות למקרים בהם התשובות של הנבדק היו נכונות בלבד, ובטווח זמן של 0 עד 3.75 שניות (T_Max_filter). כמו כן, בכל בלוק יש בוודאות יותר מ- 20 trails שעומדים בקריטריונים הללו.



מערכת הצירים השמאלית מתייחסת למקרים בהם לא הוצגה מטרה. מערכת הצירים הימנית מתייחסת למקרים בהם הוצגה מטרה. בכל אחת ממערכות הצירים מוצגים 4 גרפים בסך הכל. הצבע הכחול מייצג סט מסוג feature. הצבע האדום מייצג סט מסוג conjunction.

מטרתנו בהצגה הנ"ל היא לאפשר בחינה ויזואלית נוחה לשוני שנוצר בזמן תגובתו של הנבדק כפונקציה של מאפייני סט הגירויים שהוא קיבל (גודל הסט, סוג הסט, וקיומה/ העדרה של מטרה). לשם כך יצרנו שני גרפים (קו רצוף וקו מקווקו) לכל אחד מהמקרים. הגרף הרצוף מייצג את ממוצע זמני התגובה (בשניות) של הנבדק בגודל הסט, סוג הסט ומצב המטרה הרלוונטיים. הקווים האנכיים מייצגים את סטיות התקן של כל אחד מהממוצעים האלה. על ידי שימוש בפונקציה corr, חושב מתאם פירסון בין גודל הסט לבין זמן התגובה בכל אחד מהתנאים (R). בנוסף חושבה מובהקותם של כל אחד ממתאמי הפירסון (P-value). הנתונים מרוכזים בטבלה הבאה:

	Feature Absence	Conjunction Absence	Feature Present	Conjunction Present
R	{[0.9693]}	{[0.9666]}	{[0.8130]}	{[0.9798]}
P-Value	{[0.0307]}	{[0.0334]}	{[0.1870]}	{[0.0202]}
significance level	{'*'}	{'*'}	{'n.s'}	{'*'}

(טבלה 1)

על ידי שימוש בפונקציות polyfit ו- polyval יצרנו לכל אחד מהגרפים הרצופים את עקומת הניבוי הטובה ביותר האפשרית ממעלה ראשונה. עקומות הניבוי מוצגות בגרפים המקווקוים. הצגת הגרפים המקווקוים מאפשרת הערכה לזמן התגובה הממוצע המנובא לגודלי סטים שלא

נבחנו. כמו כן, עקומת הניבוי מאפשרת בחינה מדויקת של שיפוע העקומה בכל אחד מהמקרים, ובכך הערכה להשפעת גודל הסט על זמן התגובה כתלות במאפייני הסט (סוגו של הסט ומצב המטרה). השיפוע ונקודת החיתוך עם ציר ה-Y של כל אחת מעקומות הניבוי מצרופות בטבלה הבאה.

	Feature Absence	Conjunction Absence	Feature Present	Conjunction Present
Slope	0.10025	0.11095	0.034825	0.069366
Coefficient	0.7174	1.5351	0.675	1.5263

(טבלה 2)

דיון ומסקנות

באופן כללי ניתן לומר כי התוצאות שמצאנו בניסוי מתיישבות באופן די מדויק עם התיאוריה של אן טריזמן בנוגע לחיפוש חזותי. נציין כי נתייחס בניתוח זה לזמן תגובה כאינדיקטור לתהליכים שונים בחשיבה ובסריקה החזותית.

ראשית נתייחס לתוצאה היוצאת מהכלל. בתנאי Feature Present ניתן לשים לב כבר מגרף התוצאות כי הקשר בין גודל הסט וזמן התגובה אינו לינארי. חיזוק משמעותי לשיפוט הראשוני ניתן לקבל בטבלאות 1 ו2 המציגות כי תנאי זה בעל המתאם הנמוך ביותר (R=0.81) מבין כלל התנאים ואף היחיד שאינו מובהק (P-value = 0.187) והשיפוע עבורו בקו הניבוי הוא המתון ביותר וקרוב במיוחד לשיפוע 0 (0.034). מסקנה מתבקשת מנתונים אלו היא כי אין קשר לינארי בין גודל הסט וזמן התגובה עבור תנאי זה.

מצב זה מתיישב באופן נפלא עם התיאוריה של אן טריזמן. מצב זה מתאר את אפקט ה"POP-OUT" אשר לפיו, כאשר גירוי המטרה שונה רק במאפיין אחד משאר המסיחים אין צורך בחיפוש קשבי על מנת לאתרו ובאמצעות עיבוד מקבילי הנבדק מזהה את הגירוי יוצא הדופן ומאתרו ללא קשר במספר המסיחים.

בשאר התנאים נראה כי מתקיים קשר לינארי חזק מאוד בין גודל הסט למהירות התגובה (כלל התנאים מובהקים והR המינימלי שווה ל0.96). במבט ראשון ניתן לחשוב כי מאחר ובכלל התנאים קשר לינארי חזק ודומה הרי שמתרחש תהליך דומה בכולם. אך במבט עמוק יותר בתוצאות, בעיקר בטבלה 2, ניתן להבחין באופן דק יותר בין המקרים, להצביע על ההבדלים ולראות כיצד התיאוריה של טריזמן שוב מתיישבת באופן מדויק עם התוצאות.

שנית, נתייחס למצב Feature Absence, נשים לב כי בדומה לתנאי בו המטרה הופיעה עבור חיפוש מסוג זה, נקודות החיתוך של גרף הניבוי עם ציר הY דומות (יחס של 1.02 בין ללא מטרה לעם מטרה). ההבדל המשמעותי בין התנאים הוא בשיפוע קו הניבוי. בעוד בתנאי הכולל מטרה השיפוע הוא 0.03, בתנאי ללא מטרה השיפוע הוא 0.1. מבחינת יחס בין השיפועים, השיפוע בתנאי ללא מטרה גדול פי 3.3 מהתנאי עם מטרה. הבדל חריג זה, יחד עם המובהקות בקשר הלינארי, מדגים כי אכן קיים הבדל מהותי בין התהליכים ובתנאי הFeature Absence אכן מתרחש תהליך סריקה כלשהו התלוי בגודל הסט. בהתאם לתיאוריה של טריזמן, מאחר ובשלב זה אין גירוי יוצא דופן בתכונית אחת (כולם זהים), לא מתרחש אפקט pop-out והנבדק מבצע סריקה קשבית (נציין כי

מהירה יחסית) וזאת במטרה לוודא כי אכן לא קיים גירוי יוצא דופן. התיאוריה המתייחסת לסריקה שיטתית מקבלת חיזוק מהתוצאות בשלב זה בדיוק בגלל הקשר החזק בין גודל הסט וזמן התגובה. ככל שיש יותר גירוים לבחון בצורה שיטתית, כך זמן התגובה עולה.

נעבור כעת לדיון מעמיק יותר ב Conjunction search. גם כאן, ניתן לראות כי שני התנאים מקיימים קשר לינארי גדול ומובהק בין גודל הסט וזמן התגובה. גם כאן, נשאלת השאלה האם מדובר בתהליך דומה או זהה. נשים לב כי ההבדל המהותי בין התנאים נעוץ פעם נוספת בקווי הניבוי שלהם אשר פרטיהם מופיעים בטבלה 2. בעוד נקודות החיתוך של הגרפים עם ציר הY דומות (1.5), השיפועים מדגימים שוני מהותי - יחס של 1.6 בין שיפוע תנאי ללא מטרה לתנאי עם מטרה.

הבדל מהותי זה יכול להצביע על תהליכים שונים המתרחשים בזמן ביצוע המטלה. מסקנה סבירה בשלב זה, אשר מתיישבת עם התיאוריה של טריזמן בנושא, יכולה להיות כי במצב של שילוב תכוניות, הנבדק משתמש ב"פנס קשבי" אשר באמצעותו מבצע סריקה מדוקדקת של כל גירוי ומפרק אותו לפי התכונויות שלו. מאחר ומדובר בתנאי המשלב מספר תכוניות (2), לא מתרחש אפקט הpop-out (שלפי טריזמן קיים רק עבור תכונית אחת).

כעת נשאלת השאלה, מדוע קיים הבדל כה גדול בין השיפועים. אם כך, בהנחה ולנבדק שיטת סריקה קבועה יחסית וסדר הופעת הגירויים הינו אקראי לחלוטין, ניתן להסיק כי בעוד הנבדק מבצע סריקה שיטתית באמצעות "הפנס הקשבי" בתנאי "עם מטרה", לעיתים יגיע למטרה בשלב מוקדם של החיפוש (ויוכל להכריע מיד כי אכן קיימת מטרה) ולעיתים יגיע אלה בשלבי הסיום (ואז יוכל להכריע רק לקראת סוף החיפוש כי אכן קיימת מטרה). אם לדייק, הנבדק מוצא את גירוי המטרה בממוצע באמצע החיפוש (מבחינת מספר פריטים) ובהתאם זמן התגובה מושפע (דוגמא לטובת הבהרה - אם מדובר ב8 פריטים, הנבדק מבצע סריקה שיטתית של כל ה8 ובממוצע מגיע לגירוי המטרה בגירוי ה4).

עבור תנאי conjunction absence, המצב שונה בתכלית. בעוד בתנאי הקודם יש סיבה טובה להעריך כי הנבדק פוגש במטרה בממוצע באמצע גודל הסט, כאן, על הנבדק לסרוק את כלל הפריטים באופן שיטתי על מנת לקבוע כי לא קיימת מטרה. לכן במצב זה אם מדובר בסט של 8 פריטים, הנבדק יסרוק בממוצע את כל 8 הפריטים לפני שיכריע כי מדובר בחזרה ללא מטרה וזמני התגובה יושפעו בהתאם. הבדל מהותי זה מודגם לנו באמצעות שיפועי קו הניבוי.