Description of my experiment and variables:

I will be presenting a transsaccadic working memory (TSWM) task in a dark room with their heads stabilized using a chin rest and an EyeLink II eye tracker fixed to their head. The stimuli used in the experiment are presented on an LCD monitor which is centred at eye level. This experiment was designed to probe both the upper and lower visual field. A fixation point is presented on the screen. A memory array of three coloured squares would momentarily appear and then disappear around the fixation point in a random position sampled from an (otherwise invisible) octagon's eight vertices (centred around the fixation point). The participants will be instructed to remember the location of the squares presented in the memory array using their peripheral vision, while maintaining fixation on the fixation point. A mask consisting of many coloured squares will pop up momentarily on the screen and disappear. The saccade target would then appear 15 degrees to the left, right, above or below the FP and change colours into one of the remembered squares from the memory array. Using a mouse, participants will click on the remembered location of the cued square.

Therefore, the participants will be indicating on the screen, where the remembered location of the cued stimulus is. The 8 conditions are: Left saccade left visual field (LVF), left saccade right visual field (RVF), right saccade LVF, right saccade RVF, up saccade LVF, up saccade RVF, down saccade LVF and down saccade RVF. The dependent variables will be the mouse clicks of the individual (x and y coordinates).

All in all, I want to look at the systematic and unsystematic errors for the participants. That is, I will need to compute standard deviations per condition (per click cluster) and compare the average x and y standard deviation across my 8 conditions. That will give me the unsystematic errors. I will then conduct a Procrustes analysis to fit the participants responses into the actual location of the stimuli, to look at systematic errors.

The following pages will outline just how I plan to do this.

PseudoCode Outline:

The first thing I want to do is code the vertical and horizontal coordinates from the dataset to explicitly tell me if the individual made a horizontal or vertical saccade.

If the saccade is made vertically, the value is 0 when we subtract the x coordinate of FP1 and FP2. If it is horizontally, you will get a positive or negative value. Speicifically, if it a negative number, it will be a rightward eye movement, and if a positive number it will be a leftward eye movement.

From my TSI.mat file (which contains the data about the saccade, jitter, block type…), I will look at theData.FPx – theData.FPx2 (fixation point 1 and 2). If the result is less than one or greater than one, that would indicate a horizontal saccade (*I know this to be true because the way that the screen is coded is as follows:

If theData.FPx – theData.FPx2 is = to 0, then it is vertical. I will assign horizontal a -1 value and vertical a +1 value. The +1 and -1 values will be saved in a variable HorV (Horizontal or Vertical).

If theData.FPx – theData.FPx2 is > 0, that indicates a left eye movement (denoted 1), and if < 0, a right eye movement (denoted -1). This will be saved in the variable LorR (left or right).

Finally, if theData.FPy – theData.FPy2 < than 0, then it is a downwards eyemovement, and I will denote it a -1 value. And if theData.FPy – theData.FPy2 is > 0, I will denote it a 1 for vertical eye movements.

This will add 3 variables to my theData set; HorV, LorR and UorD. This will help me further segment my data for my analysis.

Function AddSaccDir(subNum,blockNum)

HorV = theData.FPx - theData.FPx2;

HorV(HorV<0) = -1;

```
HorV(HorV>0) = -1;

HorV(HorV==0) = 1;

LorR = theData.FPx - theData.FPx2;

LorR(LorR>0) = 1;

LorR(LorR<0) = -1;

UorD = theData.FPy - theData.FPy2;

UorD(UorD>0) = 1;

UorD(UorD<0) = -1;

theData.HorV = HorV;

theData.LorR = LorR;

theData.UorD = UorD;
```

I will create a function that will do all the above adjusting automatically. I will create a list of all the participants, and a list for number of blocks. Then I will make a loop:

```
Function MassAddSaccade()

For iptcp = 1:length(ptcplist)
        subNum = ptcplist(iptcp);
        for iItem = 1:length(blocklist)
                blockNum = blocklist(iItem);
                AddSaccDir(subNum,blockNum)

        End

End

End
```

<u>** THIS WILL ALLOW ME TO HAVE PRELIMINARY VARIABLES TO MANIPULATE THE DATA **</u>

The next step is to find the average standard deviation for each of the 8 possible clusters (where the stimuli could come up) separately for each participant. These standard deviations will then be inputted into SPSS to conduct a 3-way repeated measures ANOVA, the three factors being: Saccade plane (Horizontal or Vertical), Saccade Direction (L/U, R/D) and Visual field (Left or Right VF).

I will start off my initializing my variables needed:

```
Subs = [2,4,5,7,8,9,11,14,15,23,26,30,32,34,35,36,37,38,39,41,42,43,44]

nSubs = length(subs)
```

outlier_thresh = 3 % This is the amount of standard deviations above the mean that a response has to be to be considered an outlier

Now I want to segment my data into clusters and conditions. To do this, I will need to first remove the horizontal and vertical jitter that I induced so I can align the clusters into 8 possible categories.

```
theData.StimLocNoJitterX = theData.selXingsX - theData.stimJitterX;
theData.StimLocNoJitterY = theData.selXingsY - theData.stimJitterY;
theData.RespLocNoJitterX = theData.responseX - theData.stimJitterX;
theData.RespLocNoJitterY = theData.responseY - theData.stimJitterY;
```

Then I want to implement a line of code that will give each coordinate for every trial a unique value in order to separate out the clusters

```
theData.uniqueLocXY = round(sqrt((theData.StimLocNoJitterX.^2) +
(theData.StimLocNoJitterY.^2)),2);
```

Thenm using the unique() function, I can get a set of 8 unique numbers corresponding to the 8 unique locations of the stimuli:

```
uniqueLocs = unique(theData.uniqueLocXY);
```

Then I want to find the corresponding responses to each of the unique 8 locations

- This will allow me to isolate the responses into each condition. For example, clusters 1, 2, 3 and 5 are for the condition Left saccade LVF (left visual field).
- I want to make sure to do this for each block.

I want to then concatenate the blocks together under the below example categories:

LVFLeftRespX → responses for the left LVF condition, x coordinate

LVFLeftRespY → corresponding y coordinate of above

LVFLeftActX → Actual location of Left LVF condition, x coordinate

LVFLeftActY → corresponding Y coordinate

I will then calculate the Euclidean distances ("ordinary" straight-line distance between two points in Euclidean space) mapping the response and actual location.

- To do this, I need to first create a function to implement Euclidean distances, and then call it. Below is an example

Function:

```
function eucDist = calc_eucDist(x1,x2,y1,y2)

eucDist = sqrt(((x1-x2).^2)+((y1-y2).^2));

 end
```

Calculating Euclidean distance:

eucDistLVFLeft = calc_eucDist(LVFLeftRespX,LVFLeftActX,LVFLeftRespY,LVFLeftActY);

I will then z-transform the data per condition; using zscore(). Then I will remove the outliers based on the pre-determined threshold of 3. Then I will calculate the standard deviation of each condition, x and y coordinates separately.

Then I will average the x and y coordinates to get an average standard deviation for each condition per subject. This will then be submitted into SPSS.

Procrustes Analysis – Procrustes D

1. Procrustes D Analysis will allow me to determine a linera transformation (translation, rotation, scaling) of the points in matrix Y to best conform the points in matrix x. Procrustes returns the minimized value of this dissimilarity measure in D.
2. To do this analysis, I will load a data file consisting of segmented data per participant. The data will be segmented in this way:

	LLVFP			
ptcp #	Average response coordinates		Target Location Coordinates	
	x	y	x	y
Cluster 1				
Cluster 2				
Cluster 3				
Cluster 5				

3. For each condition, I will run the following:

```
for i = 1:nSub

    iSub = DLVF.iSub(:,:,i);
    X = [iSub(:,3),iSub(:,4)];
    Y = [iSub(:,1),iSub(:,2)];
    [D,Z] = procrustes(X, Y);
    listD = [listD;D];

end
```

The above code will give me D values for each condition, for each participant.

This will all then be submitted to test for significant differences between each Condition into SPSS (ANOVA).