```
(***Type the string for the main directory where the data are located***)

mainDir = "";

(***In this example, data from 2 groups (A and B) are being compared.  In each group,
there are a certain number of mice,
and for each session there are a certain number of ROIs.  In this example,
a list of ROIs for a given experiment (a "date, mouse, session" combination)
 is stored as *mainDir/date/mouse/session/Data/ROIs.txt*. The data
 values (e.g. locomotion modulation index) associated with each ROI *n*
 are stored as *mainDir/date/mouse/session/Data/DataVals_ROIn.txt****)

(*****Data for group A*******)

dateMouseSessionListGroupA = {};
(***Type the {date,mouse,session} combinations for all animals in Group A***)

mouseIndexGroupA = dateMouseSessionGroupA[[All, 2]];
(***List of mouse names for each experimental session in Group A***)

roisListGroupA = Table[
   Flatten@(ToExpression /@ Import[StringJoin[mainDir, dateMouseSessionListGroupA[[n, 1]],
        "/", dateMouseSessionListGroupA[[n, 2]], "/", dateMouseSessionListGroupA[[n, 3]],
        "/Data/", dateMouseSessionListGroupA[[n, 1]], "_", dateMouseSessionListGroupA[[n,
         2]], "_", dateMouseSessionListGroupA[[n, 3]], "_", "ROIs", ".txt"], "List"]),
   {n, 1, Length[dateMouseSessionListGroupA]}];(***List
 of ROIs for each experimental
 session
 in
 Group
 A***)

perMouseLengthsGroupA = Length /@ roisListGroupA;
(***Number of ROIs per mouse per session***)

valsGroupA = Flatten[Table[
    Table[ToExpression /@ Import[StringJoin[mainDir, dateMouseSessionListGroupA[[n, 1]],
        "/", dateMouseSessionListGroupA[[n, 2]], "/", dateMouseSessionListGroupA[[n, 3]],
        "/Data/", dateMouseSessionListGroupA[[n, 1]], "_",
        dateMouseSessionListGroupA[[n, 2]], "_", dateMouseSessionListGroupA[[n, 3]],
        "_", "DataVals_ROI", ToString[roi], ".txt"], "List"],
     {roi, roisListGroupA[[n]]}], {n, 1, Length[dateMouseSessionListGroupA]}]];
 (***Data values associated with each
 ROI
 in
 Group
 A***)

valsGroupANested = TakeList[valsGroupA, perMouseLengthsGroupA];
(***Nest the list of data values based on the numbers of ROIs per mouse per session***)

mouseGroupedValsGroupA = Table[
   {mouseIndexGroupA[[n]], valsGroupANested[[n]]}, {n, 1, Length[valsGroupANested]}];
 (***Assign mouse idendity to each list of data values nested in "valsGroupANested"***)
```

```
mouseGroupedValsAllGroupA =
  DeleteCases[#, _String] & /@ (Flatten /@ GatherBy[mouseGroupedValsGroupA, First]);
 (***Group all data lists with the same first element (i.e. mouse identity)
 together and delete the mouse identity string so only the data values remain***)

groupAMouseValList = Flatten[
   Table[Reverse /@ (Append[#, "GroupA"] & /@ Partition[Riffle[mouseGroupedValsGroupA[[
          n, 2]], mouseGroupedValsGroupA[[n, 1]], {2, -1, 2}], 2]),
    {n, 1, Length[mouseGroupedValsGroupA]}], 1];(***Create a list in which
 each element is {group #, mouse identity, data value for an ROI}***)

(*****Data for group B*******)

dateMouseSessionListGroupB = {};

mouseIndexGroupB = dateMouseSessionGroupB[[All, 2]];

roisListGroupB = Table[
   Flatten@(ToExpression /@ Import[StringJoin[mainDir, dateMouseSessionListGroupB[[n, 1]],
        "/", dateMouseSessionListGroupB[[n, 2]], "/", dateMouseSessionListGroupB[[n, 3]],
        "/Data/", dateMouseSessionListGroupB[[n, 1]], "_", dateMouseSessionListGroupB[[n,
         2]], "_", dateMouseSessionListGroupB[[n, 3]], "_", "ROIs", ".txt"], "List"]),
   {n, 1, Length[dateMouseSessionListGroupB]}];

perMouseLengthsGroupB = Length /@ roisListGroupB;

valsGroupB = Flatten[Table[
    Table[ToExpression /@ Import[StringJoin[mainDir, dateMouseSessionListGroupB[[n, 1]],
        "/", dateMouseSessionListGroupB[[n, 2]], "/", dateMouseSessionListGroupB[[n, 3]],
        "/Data/", dateMouseSessionListGroupB[[n, 1]], "_",
        dateMouseSessionListGroupB[[n, 2]], "_", dateMouseSessionListGroupB[[n, 3]],
        "_", "DataVals_ROI", ToString[roi], ".txt"], "List"],
     {roi, roisListGroupB[[n]]}], {n, 1, Length[dateMouseSessionListGroupB]}]];

valsGroupBNested = TakeList[valsGroupB, perMouseLengthsGroupB];

mouseGroupedValsGroupB = Table[
   {mouseIndexGroupB[[n]], valsGroupBNested[[n]]}, {n, 1, Length[valsGroupBNested]}];

mouseGroupedValsAllGroupB =
  DeleteCases[#, _String] & /@ (Flatten /@ GatherBy[mouseGroupedValsGroupB, First]);

medianValsPerMouseGroupB = Median /@ mouseGroupedValsAllGroupB;

groupBMouseValList = Flatten[
   Table[Reverse /@ (Append[#, "GroupB"] & /@ Partition[Riffle[mouseGroupedValsGroupB[[
          n, 2]], mouseGroupedValsGroupB[[n, 1]], {2, -1, 2}], 2]),
    {n, 1, Length[mouseGroupedValsGroupB]}], 1];

(************************************)

groupAvsGroupBMouseValList =
 Prepend[#, {"Group", "Mouse", "Value"}] &@Join[groupAMouseValList, groupBMouseValList];
(***Join the {group,mouse,value} data from group A and
 B and add the heading {"Group","Mouse","Value"}
```

In[•]:=
```
(***)

CreateDirectory[StringJoin[mainDir, "/SemiweightedStats/GroupAvsGroupB"]];

Export[StringJoin[mainDir, "/SemiweightedStats/GroupAvsGroupB/",
   "groupAvsGroupBMouseValList.xlsx"], groupAvsGroupBMouseValList];

(***Put data in format for semi-weighted average analysis in MATLAB***)

groupAvsGroupB = Drop[Flatten[Import[StringJoin[mainDir,
      "/SemiweightedStats/GroupAvsGroupB/groupAvsGroupBMouseValList.xlsx"],
     "Data"], 1], 1];

groupAVsGroupBTypeCategory = (groupAvsGroupB /. "GroupA" → 1) /. "GroupB" → 2;

groupAvsGroupBAllMice = DeleteDuplicates[groupAvsGroupB[[All, 2]]];

groupAvsGroupBTypeMouseCategory = Table[{groupAvsGroupBTypeCategory[[n, 1]],
    Flatten[Position[groupAvsGroupBAllMice, groupAvsGroupBTypeCategory[[n, 2]]]][[1]],
    groupAvsGroupBTypeCategory[[n, 3]]}, {n, 1, Length[groupAvsGroupBTypeCategory]}];

groupAvsGroupBData = groupAvsGroupBTypeMouseCategory[[All, 3]];

groupAvsGroupBConditions = groupAvsGroupBTypeMouseCategory[[All, 1]];

groupAvsGroupBIndividuals = groupAvsGroupBTypeMouseCategory[[All, 2]];

(***)

CreateDirectory[
  StringJoin[mainDir, "/SemiweightedStats/GroupAvsGroupB/SemiweightedStruct"]];

Export[StringJoin[mainDir,
   "/SemiweightedStats/GroupAvsGroupB/SemiweightedStruct/groupAvsGroupB_Data.mat"],
  groupAvsGroupBData];

Export[StringJoin[mainDir,
   "/SemiweightedStats/GroupAvsGroupB/SemiweightedStruct/groupAvsGroupB_Conditions.mat"],
  groupAvsGroupBConditions];

Export[StringJoin[mainDir,
   "/SemiweightedStats/GroupAvsGroupB/SemiweightedStruct/groupAvsGroupB_Individuals"],
  groupAvsGroupBIndividuals];
```