```
In[•]:= (***Input identifying information***)

In[•]:= date = ToString[Evaluate[Input["Input the date of the experiment"]]]

In[•]:= mouse = ToString[Evaluate[Input["Input the mouse identity (e.g. Mouse123)"]]]

In[•]:= sessionNum = Evaluate[Input["Input the session number"]]

In[•]:= (***Import the frame times for the 2P images and calculate the frame rate***)

In[•]:= tpFrameTimes =
          Drop[Drop[(Import[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date,
                "/", mouse, "/Session", ToString[sessionNum], "/", date, "_", mouse, "_",
                "Session", ToString[sessionNum], "_2PFrameTimes.txt"], "List"]), 16], -1];

In[•]:= tpFrameRate = Round[Length[tpFrameTimes] / (Last[tpFrameTimes] - First[tpFrameTimes])];

In[•]:= (***For each ROI picked for the session, upload the extracted dF/F0 time series***)

In[•]:= numROIs =
          Length[FileNames["*", File[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date,
              "/", mouse, "/Session", ToString[sessionNum], "/dFOverF0TimeSeries/"]]]];

In[•]:= Table[Evaluate@ToExpression[StringJoin["dFFts", ToString[n]]] =
            ToExpression /@ Import[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/",
                date, "/", mouse, "/Session", ToString[sessionNum], "/dFoverF0TimeSeries/",
                date, "_", mouse, "_Session", ToString[sessionNum], "_",
                "dFoverF0ts_ROI", ToString[n], ".txt"], "List"];, {n, 1, numROIs}];

In[•]:= (***Import the Spike2 file indicating onset times of the noise movies***)

In[•]:= vStimOnsets =
          Drop[Drop[(Import[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date,
                "/", mouse, "/Session", ToString[sessionNum], "/", date, "_", mouse, "_",
                "Session", ToString[sessionNum], "_VisStimOnsetTimes.txt"], "List"]), 16], -1];

In[•]:= (***Get interpolation functions for the dF/F0 time series for each ROI***)

In[•]:= Table[Evaluate@ToExpression[StringJoin["dFFInterpFunc", ToString[n]]] =
            Interpolation[ToExpression[StringJoin["dFFts", ToString[n]]]];, {n, 1, numROIs}];

In[•]:= (***For each ROI,
        calculate the raw evoked dF/F0 for around each stimulus presentation onset,
        which includes 1 s prior stimulus onset and 3 s post stimulus offset***)
        Table[Evaluate@ToExpression[StringJoin["evokedRawDFFs", ToString[n]]] =
            Table[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]])[i],
              {i, vStimOnsets[[x]] - 1, vStimOnsets[[x]] + 5, 1 / tpFrameRate}],
             {x, 1, Length[vStimOnsets]}];, {n, 1, numROIs}];
```

```
In[•]:= (***For each ROI, calculate the baseline (0.5 s pre-stim)-
         subtracted evoked dF/F0 for each stimulus presentation***)
        Table[Evaluate@ToExpression[StringJoin["blSubtractEvokedDFFs", ToString[n]]] =
            Table[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]]) [i] -
                (Mean[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]]) [j],
                    {j, vStimOnsets[[x]] - 1, vStimOnsets[[x]]}]]),
               {i, vStimOnsets[[x]] - 1, vStimOnsets[[x]] + 5, 1/tpFrameRate}],
              {x, 1, Length[vStimOnsets]}];, {n, 1, numROIs}];

In[•]:= (***For each ROI,
        calculate the z-scored dF/F0 for around each stimulus presentation onset,
        which includes 1 s prior stimulus onset and 3 s post stimulus offset***)

In[•]:= Table[Evaluate@ToExpression[StringJoin["baselineMeans", ToString[n]]] =
            Table[Mean[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]]) [k],
                {k, vStimOnsets[[x]] - 1, vStimOnsets[[x]], 1/tpFrameRate}]],
              {x, 1, Length[vStimOnsets]}];, {n, 1, numROIs}];

In[•]:= Table[Evaluate@ToExpression[StringJoin["baselineSDs", ToString[n]]] = Table[
             StandardDeviation[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]]) [
                k], {k, vStimOnsets[[x]] - 1, vStimOnsets[[x]], 1/tpFrameRate}]],
              {x, 1, Length[vStimOnsets]}];, {n, 1, numROIs}];

In[•]:= Table[Evaluate@ToExpression[StringJoin["evokedZscoredDFFs", ToString[n]]] =
            Table[Table[(((ToExpression[StringJoin["dFFInterpFunc", ToString[n]]]) [i]) -
                  (ToExpression[StringJoin["baselineMeans", ToString[n]]]) [[x]]) /
                ((ToExpression[StringJoin["baselineSDs", ToString[n]]]) [[x]]),
               {i, vStimOnsets[[x]] - 1, vStimOnsets[[x]] + 5, 1/tpFrameRate}],
              {x, 1, Length[vStimOnsets]}];, {n, 1, numROIs}];

In[•]:= evokedTimeVals = Table[n, {n, -1, 5, 1/tpFrameRate}];

In[•]:= (***Import the csv file containing the stimulus
         properties for each stimulus presentation in the session***)

In[•]:= stimProps = Import[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/",
             date, "/", mouse, "/Session", ToString[sessionNum], "/", date, "_",
             mouse, "_", "Session", ToString[sessionNum], "_StimProps.csv"]];

In[•]:= stimProps = Take[stimProps, Length[vStimOnsets]];

In[•]:= {Length[stimProps], Length[vStimOnsets], Length[stimProps] == Length[vStimOnsets]}

In[•]:= (***Extract grating orientation for each stimulus presented***)

In[•]:= sizes = Round /@ (Part[#, 1] & /@ stimProps);

In[•]:= sizeList = Part[#, 1] & /@ Sort[Tally[sizes]]

In[•]:= (***Find the positions at which each of the 8 stimulus sizes was presented***)

In[•]:= pos10deg = Flatten[Position[sizes, 10]];

In[•]:= pos15deg = Flatten[Position[sizes, 15]];
```

```
In[•]:= pos22deg = Flatten[Position[sizes, 22]];

In[•]:= pos32deg = Flatten[Position[sizes, 32]];

In[•]:= pos46deg = Flatten[Position[sizes, 46]];

In[•]:= pos68deg = Flatten[Position[sizes, 68]];

In[•]:= pos100deg = Flatten[Position[sizes, 100]];

In[•]:= (***For each ROI, make a time series of raw evoked dF/F0,
        baseline-subtracted evoked dF/F0, and mean baseline-subtracted evoked dF/F0***)

In[•]:= Table[Evaluate@ToExpression[StringJoin["evokedRawDFFsTS", ToString[n]]] =
            Table[Partition[Riffle[evokedTimeVals,
                (ToExpression[StringJoin["evokedRawDFFs", ToString[n]]]) [[m]]], 2], {m, 1, Length[
                (ToExpression[StringJoin["evokedRawDFFs", ToString[n]]])]}];, {n, 1, numROIs}];

In[•]:= Table[Evaluate@ToExpression[StringJoin["blSubtractEvokedDFFsTS", ToString[n]]] =
            Table[Partition[Riffle[evokedTimeVals,
                (ToExpression[StringJoin["blSubtractEvokedDFFs", ToString[n]]]) [[m]]], 2],
             {m, 1, Length[(ToExpression[StringJoin["blSubtractEvokedDFFs", ToString[n]]])]}];,
          {n, 1, numROIs}];

In[•]:= Table[Evaluate@ToExpression[StringJoin["meanBLSubtractEvokedDFFsTS", ToString[n]]] =
            Mean[Table[Partition[Riffle[evokedTimeVals,
                 (ToExpression[StringJoin["blSubtractEvokedDFFs", ToString[n]]]) [[m]]], 2],
              {m, 1, Length[(ToExpression[StringJoin["blSubtractEvokedDFFs",
                    ToString[n]]])]}]];, {n, 1, numROIs}];

In[•]:= (***For each ROI,
        calculate the raw evoked dF/F0 for around each stimulus presentation onset,
        which includes 1 s prior stimulus onset and 3 s post stimulus offset,
        but JUST for the smaller stimulus sizes***)
        Table[Evaluate@ToExpression[StringJoin["evokedRawDFFsSmall", ToString[n]]] =
            Table[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]]) [i],
              {i, vStimOnsets[[x]] - 1, vStimOnsets[[x]] + 5, 1 / tpFrameRate}],
             {x, Sort[Join[pos10deg, pos15deg, pos22deg, pos32deg, pos46deg]]}];, {n, 1, numROIs}];

In[•]:= (***For each ROI, calculate the baseline (0.5 s pre-stim)-
         subtracted evoked dF/F0 for each stimulus presentation,
        bjt just for smaller stimuli***)
        Table[Evaluate@ToExpression[StringJoin["blSubtractEvokedDFFsSmall", ToString[n]]] =
            Table[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]]) [i] -
                (Mean[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]]) [j],
                   {j, vStimOnsets[[x]] - 1, vStimOnsets[[x]]}]]),
              {i, vStimOnsets[[x]] - 1, vStimOnsets[[x]] + 5, 1 / tpFrameRate}],
             {x, Sort[Join[pos10deg, pos15deg, pos22deg, pos32deg, pos46deg]]}];, {n, 1, numROIs}];
```

```
In[•]:= (***For each ROI,
        calculate the raw evoked dF/F0 for around each stimulus presentation onset,
        which includes 1 s prior stimulus onset and 3 s post stimulus offset,
        but JUST for the larger stimulus sizes***)
        Table[Evaluate@ToExpression[StringJoin["evokedRawDFFsLarge", ToString[n]]] =
            Table[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]]) [i],
              {i, vStimOnsets[[x]] - 1, vStimOnsets[[x]] + 5, 1 / tpFrameRate}],
             {x, Sort[Join[pos68deg, pos100deg]]}];, {n, 1, numROIs}];
```

```
In[•]:= (***For each ROI, calculate the baseline (0.5 s pre-stim)-
         subtracted evoked dF/F0 for each stimulus presentation,
        bjt just for larger stimuli***)
        Table[Evaluate@ToExpression[StringJoin["blSubtractEvokedDFFsLarge", ToString[n]]] =
            Table[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]]) [i] -
                (Mean[Table[(ToExpression[StringJoin["dFFInterpFunc", ToString[n]]]) [j],
                   {j, vStimOnsets[[x]] - 1, vStimOnsets[[x]]}]]),
              {i, vStimOnsets[[x]] - 1, vStimOnsets[[x]] + 5, 1 / tpFrameRate}],
             {x, Sort[Join[pos68deg, pos100deg]]}];, {n, 1, numROIs}];
```

```
In[•]:= (***For each ROI, make a time series of raw evoked dF/F0,
        baseline-subtracted evoked dF/F0,
        and mean baseline-subtracted evoked dF/F0, but JUST for smaller stimuli***)
```

```
In[•]:= Table[Evaluate@ToExpression[StringJoin["evokedRawDFFsSmallTS", ToString[n]]] =
            Table[Partition[Riffle[evokedTimeVals,
                (ToExpression[StringJoin["evokedRawDFFsSmall", ToString[n]]]) [[m]]], 2],
             {m, 1, Length[(ToExpression[StringJoin["evokedRawDFFsSmall", ToString[n]]])]}];, {n,
          1, numROIs}];
```

```
In[•]:= Table[Evaluate@ToExpression[StringJoin["blSubtractEvokedDFFsSmallTS", ToString[n]]] =
            Table[Partition[Riffle[evokedTimeVals,
                (ToExpression[StringJoin["blSubtractEvokedDFFsSmall", ToString[n]]]) [[m]]], 2],
             {m, 1, Length[(ToExpression[StringJoin["blSubtractEvokedDFFsSmall",
                  ToString[n]]])]}];, {n, 1, numROIs}];
```

```
In[•]:= Table[Evaluate@ToExpression[StringJoin["meanBLSubtractEvokedDFFsSmallTS", ToString[n]]] =
            Mean[Table[Partition[Riffle[evokedTimeVals,
                 (ToExpression[StringJoin["blSubtractEvokedDFFsSmall", ToString[n]]]) [[m]]], 2],
              {m, 1, Length[(ToExpression[StringJoin["blSubtractEvokedDFFsSmall",
                   ToString[n]]])]}]];, {n, 1, numROIs}];
```

```
In[•]:= (***For each ROI, make a time series of raw evoked dF/F0,
        baseline-subtracted evoked dF/F0,
        and mean baseline-subtracted evoked dF/F0, but JUST for larger stimuli***)
```

```
In[•]:= Table[Evaluate@ToExpression[StringJoin["evokedRawDFFsLargeTS", ToString[n]]] =
            Table[Partition[Riffle[evokedTimeVals,
                (ToExpression[StringJoin["evokedRawDFFsLarge", ToString[n]]]) [[m]]], 2],
             {m, 1, Length[(ToExpression[StringJoin["evokedRawDFFsLarge", ToString[n]]])]}];, {n,
          1, numROIs}];
```

```
In[∘]:= Table[Evaluate@ToExpression[StringJoin["blSubtractEvokedDFFsLargeTS", ToString[n]]] =
           Table[Partition[Riffle[evokedTimeVals,
               (ToExpression[StringJoin["blSubtractEvokedDFFsLarge", ToString[n]]])[[m]]], 2],
            {m, 1, Length[(ToExpression[StringJoin["blSubtractEvokedDFFsLarge",
                  ToString[n]]])]}];, {n, 1, numROIs}];

In[∘]:= Table[Evaluate@ToExpression[StringJoin["meanBLSubtractEvokedDFFsLargeTS", ToString[n]]] =
           Mean[Table[Partition[Riffle[evokedTimeVals,
                (ToExpression[StringJoin["blSubtractEvokedDFFsLarge", ToString[n]]])[[m]]], 2],
             {m, 1, Length[(ToExpression[StringJoin["blSubtractEvokedDFFsLarge",
                   ToString[n]]])]}]];, {n, 1, numROIs}];

In[∘]:= (***For each ROI, make a time series of z-scored dF/F0 and mean z-scored dF/F0,***)

In[∘]:= Table[Evaluate@ToExpression[StringJoin["zScoredDFFsTS", ToString[n]]] =
           Table[Partition[Riffle[evokedTimeVals,
               (ToExpression[StringJoin["evokedZscoredDFFs", ToString[n]]])[[m]]], 2],
            {m, 1, Length[(ToExpression[StringJoin["evokedZscoredDFFs", ToString[n]]])]}];, {n,
          1, numROIs}];

In[∘]:= Table[Evaluate@ToExpression[StringJoin["meanZscoredDFFsTS", ToString[n]]] =
           Mean[Table[Partition[Riffle[evokedTimeVals,
                (ToExpression[StringJoin["evokedZscoredDFFs", ToString[n]]])[[m]]], 2],
             {m, 1, Length[(ToExpression[StringJoin["evokedZscoredDFFs",
                   ToString[n]]])]}]];, {n, 1, numROIs}];

In[∘]:= (***To quantify visual responsiveness of each ROI,
        perform a T test comparing the mean dF/F 1 s before stimulus
         onset and the mean dF/F during the 2 s stimulus presentation,
        across all stimulus presentations.  Do the T test for both smaller and larger stimuli***)

In[∘]:= Table[Evaluate@ToExpression[StringJoin["visRespPValSmallROI", ToString[n]]] =
           Quiet[TTest[{Mean /@ (Take[#, tpFrameRate] & /@
                  (ToExpression[StringJoin["evokedRawDFFsSmall", ToString[n]]])),
              Mean /@ (Take[#, {tpFrameRate + 1, (tpFrameRate + 1) + (2 * tpFrameRate)}] & /@
                  (ToExpression[StringJoin["evokedRawDFFsSmall", ToString[n]]]))},
             AlternativeHypothesis → "Less"]];, {n, 1, numROIs}];

In[∘]:= Table[Evaluate@ToExpression[StringJoin["visRespPValLargeROI", ToString[n]]] =
           Quiet[TTest[{Mean /@ (Take[#, tpFrameRate] & /@
                  (ToExpression[StringJoin["evokedRawDFFsLarge", ToString[n]]])),
              Mean /@ (Take[#, {tpFrameRate + 1, (tpFrameRate + 1) + (2 * tpFrameRate)}] & /@
                  (ToExpression[StringJoin["evokedRawDFFsLarge", ToString[n]]]))},
             AlternativeHypothesis → "Less"]];, {n, 1, numROIs}];

In[∘]:= pValListSmall =
          Table[ToExpression[StringJoin["visRespPValSmallROI", ToString[n]]], {n, 1, numROIs}];

In[∘]:= pValListLarge =
          Table[ToExpression[StringJoin["visRespPValLargeROI", ToString[n]]], {n, 1, numROIs}];
```

```
In[∘]:= (***Extract the ROIs with significant responses,
        meaning a p-value below 0.01 for either the smaller stimuli or the larger stimuli***)

In[∘]:= sigRespROIs =
          DeleteCases[Table[If[pValListSmall[[n]] < 0.01 || pValListLarge[[n]] < 0.01, n, Null],
             {n, 1, numROIs}], Null];

In[∘]:= (***For each ROI,
        visualize the p-value and overall evoked responses relative to baseline***)

In[∘]:= Table[{{StringJoin["ROI", ToString[n]],
             ToExpression[StringJoin["visRespPValSmallROI", ToString[n]]], ListLinePlot[
              ToExpression[StringJoin["meanBLSubtractEvokedDFFsSmallTS", ToString[n]]]]},
           {StringJoin["ROI", ToString[n]], ToExpression[StringJoin["visRespPValLargeROI",
               ToString[n]]], ListLinePlot[ToExpression[StringJoin[
                "meanBLSubtractEvokedDFFsLargeTS", ToString[n]]]]}}, {n, numROIs}] // TableForm

In[∘]:= (***For each ROI, calculate a new dF/F value that indicates
         its overall visual responsiveness aross all stimulus presentations,
        where F0 is the mean dF/F 1 s before stimulus onset and F is the mean dF/F during the 2-
         s stimulus presentation***)

In[∘]:= Table[
          Evaluate@ToExpression[StringJoin["overallVisDFF", ToString[n]]] = (Mean[Part[#, 2] & /@
                 Take[Mean[(ToExpression[StringJoin["evokedRawDFFsTS", ToString[n]]])],
                  {tpFrameRate + 1, (tpFrameRate + 1) + (2 * tpFrameRate)}]] - Mean[Part[#, 2] & /@
                 Take[Mean[(ToExpression[StringJoin["evokedRawDFFsTS", ToString[n]]])],
                  tpFrameRate]]) / Mean[Part[#, 2] & /@
               Take[Mean[(ToExpression[StringJoin["evokedRawDFFsTS", ToString[n]]])],
                tpFrameRate]];, {n, 1, numROIs}];

In[∘]:= (***For each ROI, calculate an additional measure of overall visual
         responsiveness across all stimulus presentations, which, instead of a new dF/F,
        is the mean z-scored dF/F during the 2 s of stimulus presentation***)

In[∘]:= Table[Evaluate@ToExpression[StringJoin["overallVisZScore", ToString[n]]] = Mean[
              Part[#, 2] & /@ Take[Mean[(ToExpression[StringJoin["zScoredDFFsTS", ToString[n]]])],
                {tpFrameRate + 1, (tpFrameRate + 1) + (2 * tpFrameRate)}]];, {n, 1, numROIs}];

In[∘]:= (***For each ROI, collect the peri-stimulus dF/F traces for each size***)

In[∘]:= Table[Table[Evaluate@
               ToExpression[StringJoin["periStimDFF", ToString[m], "DegROI", ToString[n]]] =
              (ToExpression[StringJoin["evokedRawDFFsTS", ToString[n]]]) [[ToExpression[
                 StringJoin["pos", ToString[m], "deg"]]]];, {n, 1, numROIs}];, {m, sizeList}];

In[∘]:= Table[Table[Evaluate@
               ToExpression[StringJoin["periStimDFFblSub", ToString[m], "DegROI", ToString[n]]] =
              (ToExpression[StringJoin["blSubtractEvokedDFFsTS", ToString[n]]]) [[ToExpression[
                 StringJoin["pos", ToString[m], "deg"]]]];, {n, 1, numROIs}];, {m, sizeList}];

In[∘]:= (***For each ROI, collect the peri-stimulus z-scored dF/F traces for each size***)
```

```
In[∘]:= Table[Table[Evaluate@ToExpression[
              StringJoin["periStimZscoredDFF", ToString[m], "DegROI", ToString[n]]] =
            (ToExpression[StringJoin["zScoredDFFsTS", ToString[n]]]) [[ToExpression[
              StringJoin["pos", ToString[m], "deg"]]]];, {n, 1, numROIs}];, {m, sizeList}];

In[∘]:= (***For visualization purposes (size):***)

In[∘]:= Table[Evaluate@ToExpression[StringJoin["maxRespValSize", ToString[n]]] =
         Max[Flatten[{Part[#, 2] & /@
              Mean[(ToExpression[StringJoin["periStimDFFblSub10DegROI", ToString[n]]])],
            Part[#, 2] & /@ Mean[(ToExpression[StringJoin["periStimDFFblSub15DegROI",
                 ToString[n]]])], Part[#, 2] & /@ Mean[(ToExpression[
                StringJoin["periStimDFFblSub22DegROI", ToString[n]]])], Part[#, 2] & /@
              Mean[(ToExpression[StringJoin["periStimDFFblSub32DegROI", ToString[n]]])],
            Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
                 "periStimDFFblSub46DegROI", ToString[n]]])], Part[#, 2] & /@
              Mean[(ToExpression[StringJoin["periStimDFFblSub68DegROI", ToString[n]]])],
            Part[#, 2] & /@ Mean[(ToExpression[StringJoin["periStimDFFblSub100DegROI",
                 ToString[n]]])]}]];, {n, 1, numROIs}];

In[∘]:= Table[Evaluate@ToExpression[StringJoin["minRespValSize", ToString[n]]] =
         Min[Flatten[{Part[#, 2] & /@
              Mean[(ToExpression[StringJoin["periStimDFFblSub10DegROI", ToString[n]]])],
            Part[#, 2] & /@ Mean[(ToExpression[StringJoin["periStimDFFblSub15DegROI",
                 ToString[n]]])], Part[#, 2] & /@ Mean[(ToExpression[
                StringJoin["periStimDFFblSub22DegROI", ToString[n]]])], Part[#, 2] & /@
              Mean[(ToExpression[StringJoin["periStimDFFblSub32DegROI", ToString[n]]])],
            Part[#, 2] & /@ Mean[(ToExpression[StringJoin[
                 "periStimDFFblSub46DegROI", ToString[n]]])], Part[#, 2] & /@
              Mean[(ToExpression[StringJoin["periStimDFFblSub68DegROI", ToString[n]]])],
            Part[#, 2] & /@ Mean[(ToExpression[StringJoin["periStimDFFblSub100DegROI",
                 ToString[n]]])]}]];, {n, 1, numROIs}];

In[∘]:= (***For each ROI,
        calculate the mean response to the presentation of each stimulus size,
        which is a new trial-averaged dF/F,
        where F0 for each trial is the mean dF/F 1 s before stimulus
         onset and F is the mean dF/F during the 2 s stimulus presentation***)
```

```
In[•]:= Table[Evaluate@ToExpression[StringJoin["respAt", ToString[m], "DegROI", ToString[n]]] =
          ((Mean[Part[#, 2] & /@ Mean[Table[N@Take[(ToExpression[StringJoin["periStimDFF",
                      ToString[m], "DegROI", ToString[n]]]) [[k]], {tpFrameRate + 1,
                   (tpFrameRate + 1) + (2 * tpFrameRate)}], {k, 1, Length[(ToExpression[
                    StringJoin["periStimDFF", ToString[m], "DegROI", ToString[n]]])]}]]]) -
            (Mean[Part[#, 2] & /@ Mean[Table[N@Take[(ToExpression[StringJoin["periStimDFF",
                      ToString[m], "DegROI", ToString[n]]]) [[k]], tpFrameRate],
                  {k, 1, Length[(ToExpression[StringJoin["periStimDFF", ToString[m],
                      "DegROI", ToString[n]]])]}]]])) /
          (Mean[Part[#, 2] & /@ Mean[Table[N@Take[(ToExpression[StringJoin["periStimDFF",
                    ToString[m], "DegROI", ToString[n]]]) [[k]], tpFrameRate],
                {k, 1, Length[(ToExpression[StringJoin["periStimDFF", ToString[m],
                    "DegROI", ToString[n]]])]}]]]);, {n, 1, numROIs}, {m, sizeList}];

In[•]:= (***For each ROI,
        calculate the z-scored mean response to the presentation of each stimulus size***)

In[•]:= Table[
          Evaluate@ToExpression[StringJoin["zScoredRespAt", ToString[m], "DegROI", ToString[n]]] =
            (Mean[Part[#, 2] & /@ Mean[Table[N@Take[(ToExpression[
                     StringJoin["periStimZscoredDFF", ToString[m], "DegROI", ToString[n]]]) [[
                   k]], {tpFrameRate + 1, (tpFrameRate + 1) + (2 * tpFrameRate)}],
                {k, 1, Length[(ToExpression[StringJoin["periStimZscoredDFF", ToString[m],
                      "DegROI", ToString[n]]])]}]]]);, {n, 1, numROIs}, {m, sizeList}];

In[•]:= (***Make size tuning curves***)

In[•]:= Table[Evaluate@ToExpression[StringJoin["respVsSizeROI", ToString[n]]] =
          {{10, ToExpression[StringJoin["respAt10DegROI", ToString[n]]]},
           {15, ToExpression[StringJoin["respAt15DegROI", ToString[n]]]},
           {22, ToExpression[StringJoin["respAt22DegROI", ToString[n]]]},
           {32, ToExpression[StringJoin["respAt32DegROI", ToString[n]]]},
           {46, ToExpression[StringJoin["respAt46DegROI", ToString[n]]]},
           {68, ToExpression[StringJoin["respAt68DegROI", ToString[n]]]},
           {100, ToExpression[StringJoin["respAt100DegROI", ToString[n]]]}};, {n, 1, numROIs}];

In[•]:= (***Make size tuning curves from z-scored dF/F***)

In[•]:= Table[Evaluate@ToExpression[StringJoin["zScoredRespVsSizeROI", ToString[n]]] =
          {{10, ToExpression[StringJoin["zScoredRespAt10DegROI", ToString[n]]]},
           {15, ToExpression[StringJoin["zScoredRespAt15DegROI", ToString[n]]]},
           {22, ToExpression[StringJoin["zScoredRespAt22DegROI", ToString[n]]]},
           {32, ToExpression[StringJoin["zScoredRespAt32DegROI", ToString[n]]]},
           {46, ToExpression[StringJoin["zScoredRespAt46DegROI", ToString[n]]]},
           {68, ToExpression[StringJoin["zScoredRespAt68DegROI", ToString[n]]]},
           {100, ToExpression[
              StringJoin["zScoredRespAt100DegROI", ToString[n]]]}};, {n, 1, numROIs}];

In[•]:= (**Visualize the size tuning responses and size
         tuning curves for each significantly responsive ROI**)
```

```
In[∘]:= Manipulate[GraphicsGrid[{{ListLinePlot[Mean[ToExpression[
            StringJoin["periStimDFFblSub10DegROI", ToString[ROI]]]], PlotLabel → "10%",
          PlotRange → {All, {ToExpression[StringJoin["minRespValSize", ToString[ROI]]],
             ToExpression[StringJoin["maxRespValSize", ToString[ROI]]]}}], ListLinePlot[
          Mean[ToExpression[StringJoin["periStimDFFblSub15DegROI", ToString[ROI]]]],
          PlotLabel → "15%",
          PlotRange → {All, {ToExpression[StringJoin["minRespValSize", ToString[ROI]]],
             ToExpression[StringJoin["maxRespValSize", ToString[ROI]]]}}], ListLinePlot[
          Mean[ToExpression[StringJoin["periStimDFFblSub22DegROI", ToString[ROI]]]],
          PlotLabel → "22%",
          PlotRange → {All, {ToExpression[StringJoin["minRespValSize", ToString[ROI]]],
             ToExpression[StringJoin["maxRespValSize", ToString[ROI]]]}}], ListLinePlot[
          Mean[ToExpression[StringJoin["periStimDFFblSub32DegROI", ToString[ROI]]]],
          PlotLabel → "32%",
          PlotRange → {All, {ToExpression[StringJoin["minRespValSize", ToString[ROI]]],
             ToExpression[StringJoin["maxRespValSize", ToString[ROI]]]}}], ListLinePlot[
          Mean[ToExpression[StringJoin["periStimDFFblSub46DegROI", ToString[ROI]]]],
          PlotLabel → "46%",
          PlotRange → {All, {ToExpression[StringJoin["minRespValSize", ToString[ROI]]],
             ToExpression[StringJoin["maxRespValSize", ToString[ROI]]]}}], ListLinePlot[
          Mean[ToExpression[StringJoin["periStimDFFblSub68DegROI", ToString[ROI]]]],
          PlotLabel → "68%",
          PlotRange → {All, {ToExpression[StringJoin["minRespValSize", ToString[ROI]]],
             ToExpression[StringJoin["maxRespValSize", ToString[ROI]]]}}], ListLinePlot[
          Mean[ToExpression[StringJoin["periStimDFFblSub100DegROI", ToString[ROI]]]],
          PlotLabel → "100%",
          PlotRange → {All, {ToExpression[StringJoin["minRespValSize", ToString[ROI]]],
             ToExpression[StringJoin["maxRespValSize", ToString[ROI]]]}}]}},
       ImageSize → Full], {ROI, sigRespROIs}]
```

```
In[∘]:= Manipulate[ListLinePlot[ToExpression[StringJoin["zScoredRespVsSizeROI", ToString[ROI]]],
       Joined → True], {ROI, sigRespROIs}]
```

```
In[∘]:= (***Fit a Naka-Rushton function to contrast-
      response curves of each significantly responsive ROI, from z-scored data***)
```

```
In[∘]:= Table[Evaluate@ToExpression[StringJoin["fittedCRFModelZScore", ToString[i]]] =
         NonlinearModelFit[(ToExpression[StringJoin["zScoredRespVsSizeROI", ToString[i]]]),
          {(rMax * x^n) / (c50^n + x^n), 0 ≤ c50 ≤ 100, rMax > 0, n > 0},
          {rMax, n, c50}, x];, {i, 1, numROIs}];
```

```
In[∘]:= (***Compute the adjusted R-Square value for the model
       fit for each significantly responsive ROI, from z-scored data***)
```

```
In[∘]:= Table[Evaluate@ToExpression[StringJoin["rSquaredCRFModelFitZScore", ToString[i]]] =
         (ToExpression[StringJoin["fittedCRFModelZScore", ToString[i]]])[
          "AdjustedRSquared"];, {i, 1, numROIs}];
```

```
In[∘]:= (***Extract the c50 parameter from the contrast-
      response function of each significantly responsive ROI, from z-scored data***)
```

```
In[∘]:= Table[Evaluate@ToExpression[StringJoin["c50from", ToString[i], "zScore"]] =
            Values[(ToExpression[StringJoin["fittedCRFModelZScore", ToString[i]]])[
               "BestFitParameters"]][[3]];, {i, 1, numROIs}];

In[∘]:= (***Extract the rMax parameter from the contrast-
         response function of each significantly responsive ROI, from z-scored data***)

In[∘]:= Table[Evaluate@ToExpression[StringJoin["rMaxfrom", ToString[i], "zScore"]] =
            Values[(ToExpression[StringJoin["fittedCRFModelZScore", ToString[i]]])[
               "BestFitParameters"]][[1]];, {i, 1, numROIs}];

In[∘]:= (***Extract the rMax parameter from the contrast-
         response function of each significantly responsive ROI, from z-scored data***)

In[∘]:= Table[Evaluate@ToExpression[StringJoin["slopefrom", ToString[i], "zScore"]] =
            Values[(ToExpression[StringJoin["fittedCRFModelZScore", ToString[i]]])[
               "BestFitParameters"]][[2]];, {i, 1, numROIs}];

In[∘]:= (***Visualize fits for significantly responsive ROIs***)

In[∘]:= Manipulate[Show[Plot[ToExpression[StringJoin["fittedCRFModelZScore", ToString[n]]][x],
           {x, 0, 100}, PlotStyle → Red],
          ListLinePlot[ToExpression[StringJoin["zScoredRespVsSizeROI", ToString[n]]],
           Joined → True]], {n, sigRespROIs}]

In[∘]:= (***To determine goodness of fit based on the norm of the
         residuals (Keller and Martin, 2015, J Neurosci, 35(27)), fit the N-
         R function to normalized data and calculate the norm of the residuals from this fit***)

In[∘]:= Table[Evaluate@ToExpression[StringJoin["maxResp", ToString[i]]] =
            Max[(ToExpression[StringJoin["zScoredRespVsSizeROI", ToString[i]]])[[
              All, 2]]];, {i, 1, numROIs}];

In[∘]:= Table[Evaluate@ToExpression[StringJoin["normRespVsSizeROI", ToString[i]]] =
            MapAt[Divide[#, ToExpression[StringJoin["maxResp", ToString[i]]]] &, ToExpression[
              StringJoin["zScoredRespVsSizeROI", ToString[i]]], {All, 2}];, {i, 1, numROIs}];

In[∘]:= Table[Evaluate@ToExpression[StringJoin["fittedCRFModelNorm", ToString[i]]] =
            NonlinearModelFit[(ToExpression[StringJoin["normRespVsSizeROI", ToString[i]]]),
             {(rMax * x^n) / (c50^n + x^n), 0 ≤ c50 ≤ 100, rMax > 0, n > 0},
             {rMax, n, c50}, x];, {i, 1, numROIs}];

In[∘]:= Table[Evaluate@ToExpression[StringJoin["normResid", ToString[i]]] = Sqrt[Total[Table[
               ((ToExpression[StringJoin["fittedCRFModelNorm", ToString[i]]])[sizeList[[n]]] -
                  (ToExpression[StringJoin["normRespVsSizeROI", ToString[i]]])[[n, 2]])^2,
               {n, 1, Length[sizeList]}]]];, {i, 1, numROIs}];

In[∘]:= wellFitROIs =
          DeleteCases[Table[If[ToExpression[StringJoin["normResid", ToString[i]]] < 0.3, i, Null],
            {i, sigRespROIs}], Null];

In[∘]:= (****************************Export data***************************)

In[∘]:= (***Export the P-values for visual responsiveness of each ROI***)
```

```
In[∘]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/", mouse,
            "/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse, "_Session",
            ToString[sessionNum], "_", "visResp_pValSmallStim", ToString[n], ".txt"],
          ToExpression[StringJoin["visRespPValSmallROI", ToString[n]]]], {n, 1, numROIs}];

In[∘]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/", mouse,
            "/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse, "_Session",
            ToString[sessionNum], "_", "visResp_pValLargeStim", ToString[n], ".txt"],
          ToExpression[StringJoin["visRespPValLargeROI", ToString[n]]]], {n, 1, numROIs}];

In[∘]:= (***Export the list of the identities of the significantly responsive ROIs***)

In[∘]:= Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/", mouse,
          "/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse,
          "_Session", ToString[sessionNum], "_", "sigResponsiveROIs", ".txt"], sigRespROIs];

In[∘]:= (***Export the list of the identities of the well-fit ROIs***)

In[∘]:= Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/",
        mouse, "/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse,
        "_Session", ToString[sessionNum], "_", "wellFitROIs", ".txt"], wellFitROIs]

Out[∘]= C:/Users/garrett/Desktop/Garrett_Local/031423/Mouse23149/Session2/VisStimResults/031423
        _Mouse23149_Session2_wellFitROIs.txt

In[∘]:= (***Export dF/F values for overall visual responsiveness for each ROI***)

In[∘]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/",
            mouse, "/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse,
            "_Session", ToString[sessionNum], "_", "overallVisDFF_ROI", ToString[n], ".txt"],
          ToExpression[StringJoin["overallVisDFF", ToString[n]]]], {n, 1, numROIs}];

In[∘]:= (***Export z-scored dF/F values for overall visual responsiveness for each ROI***)

In[∘]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/", mouse,
            "/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse, "_Session",
            ToString[sessionNum], "_", "overallVisDFFZScored_ROI", ToString[n], ".txt"],
          ToExpression[StringJoin["overallVisZScore", ToString[n]]]], {n, 1, numROIs}];

In[∘]:= (***Export the mean, baseline subtracted dF/F trace for each ROI***)

In[∘]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/", mouse,
            "/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse, "_Session",
            ToString[sessionNum], "_", "overallVisDFFtrace_ROI", ToString[n], ".txt"],
          N@ToExpression[StringJoin["meanBLSubtractEvokedDFFsTS", ToString[n]]]], {n,
          1, numROIs}];

In[∘]:= (***Export the mean, z-scored dF/F trace for each ROI***)

In[∘]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/", mouse,
            "/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse, "_Session",
            ToString[sessionNum], "_", "overallVisZscoredDFFtrace_ROI", ToString[n], ".txt"],
          N@ToExpression[StringJoin["meanZscoredDFFsTS", ToString[n]]]], {n, 1, numROIs}];

In[∘]:= (***Export the crf data for each responsive ROI***)
```

```
In[∘]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/",
             mouse, "/Session", ToString[sessionNum], "/VisStimResults/", date, "_",
             mouse, "_Session", ToString[sessionNum], "_", "crf_ROI", ToString[n], ".txt"],
           ToExpression[StringJoin["respVsSizeROI", ToString[n]]]], {n, 1, numROIs}];

In[∘]:= (***Export the Z-SCORED crf data for each responsive ROI***)

In[∘]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/",
             mouse, "/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse,
             "_Session", ToString[sessionNum], "_", "crfZscored_ROI", ToString[n], ".txt"],
           ToExpression[StringJoin["zScoredRespVsSizeROI", ToString[n]]]], {n, 1, numROIs}];

In[∘]:= (***Export the RSquare value for the N-R functions fits***)

In[∘]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/",
             mouse, "/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse,
             "_Session", ToString[sessionNum], "_", "crfFitRSquare_ROI", ToString[n], ".txt"],
           ToExpression[StringJoin["rSquaredCRFModelFitZScore", ToString[n]]]], {n, 1, numROIs}];

In[∘]:= (***Export the c50 value for each ROI***)

In[∘]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/",
             mouse, "/Session", ToString[sessionNum], "/VisStimResults/", date, "_",
             mouse, "_Session", ToString[sessionNum], "_", "c50_ROI", ToString[n], ".txt"],
           ToExpression[StringJoin["c50from", ToString[n], "zScore"]]], {n, 1, numROIs}];

In[∘]:= (***Export the rMax value for each ROI***)

In[∘]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/",
             mouse, "/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse,
             "_Session", ToString[sessionNum], "_", "rMax_ROI", ToString[n], ".txt"],
           ToExpression[StringJoin["rMaxfrom", ToString[n], "zScore"]]], {n, 1, numROIs}];

In[∘]:= (***Export the slope value for each ROI***)

In[∘]:= Table[Export[StringJoin["C:/Users/garrett/Desktop/Garrett_Local/", date, "/",
             mouse, "/Session", ToString[sessionNum], "/VisStimResults/", date, "_", mouse,
             "_Session", ToString[sessionNum], "_", "slope_ROI", ToString[n], ".txt"],
           ToExpression[StringJoin["slopefrom", ToString[n], "zScore"]]], {n, 1, numROIs}];
```