OpenfMRI Voxel Activity Prediction

Matt Galloway

December 11, 2016

Introduction

In this short project we will be modeling neural activity in an `fmri` study using polynomial kernel estimation – a non-parametric density estimation method. This data was provided by the OpenfMRI organization (openfmri.org) which is a project dedicated to open-sourcing and sharing MRI (magnetic resonance imaging) data sets.

Having access to this dataset allows us to use statistics and inference methods to track brain activity. Specifically, we are interested in detecting changes in blood flow throughout the brain – a proxy for brain activity. This is done by monitoring the hemodynamic response (blood-oxygen-level dependent (BOLD)) for each voxel in the data set.

Below I will briefly outline the procedure and offer a more formal interpretation of the results.

Procedure

The dataset of interest is a NIFTI (bold.nii) file. This is a 4-dimension object $D[x, y, z, t]$ where $x, y \in \{1, ..., 64\}$, $z \in \{1, ..., 33\}$, and $t \in \{1, ..., 210\}$. Each (x, y, z) pair is a voxel – we can think of a voxel as a 3-dimension pixel. Recall that a pixel is each color "dot" on your TV or computer screen. The dimensionality of the dataset implies that there are 135,168 voxels in total.

The data set can be found on the following website: https://openfmri.org/dataset/ds000117/ . We will be analyzing the first run of subject 11.

Below I outline how to load .nii files into R and load the necessary packages/modules required for this project:

```
# load the packages into R
library(fmri)
library(locfit)

# load the data
data = read.NIFTI(path)

# extract voxel intensities
voxels = extract.data(data)

# extract mask from NIFTI file
mask = data$mask

# denote activity = mask, arbitrarily (used later)
activity = mask
```

Note that in the code above, we define a *mask*. A mask in the fMRI environment denotes voxels that should be ignored in the study. An example of this would be the area outside of the patients head but inside the MRI machine. This is clearly an area that we can ignore. Thus, for each voxel, if mask == TRUE then we would like to model its activity – otherwise the voxel is ignored. It can be shown that of the 135,168 voxels

in the data file, nearly 101,473 will be masked from the study. Reducing the dimensionality in this way is crucial and will help to reduce the overall computation time of the study.

As mentioned in the introduction, the response of each voxel is the hemodynamic response. Given the time t, we wish to create a model that will predict this hemodynamic response reliably for each voxel. To do this, we will fit a kernel non-parametric regression of $D[x, y, z, t]$ over t:

$$D[x, y, z, t] = \mu_{x,y,z}(t) + \epsilon_{x,y,z}(t)$$

where μ is the function we wish to approximate. The estimator for $\mu(t_i)$ is of the following form:

$$\hat{\mu}(t_i) = \frac{\sum D_{x,y,z}(t_i) K(\frac{t_i - t}{h})}{\sum K(\frac{t_i - t}{h})}$$

where K is the kernel (similarity measure) and h is the bandwidth of the kernel (tuning parameter). We will be utilizing the `locfit` package in R to do so. In order to get an estimate for the optimal bandwidth h, we will perform an 80/20 train/test split and use cross validation. The average mean squared error (MSE) over all voxels will be our performance metric.

The code is displayed below:

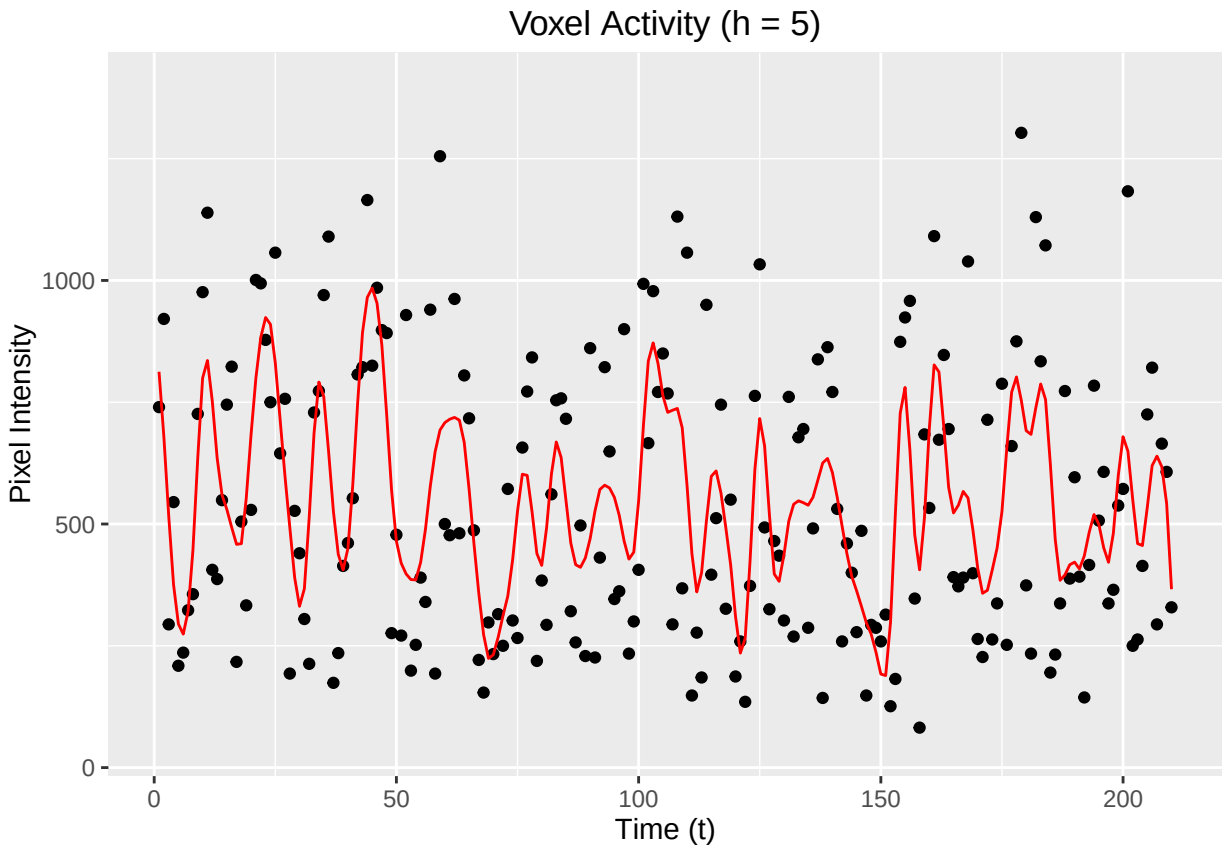
```
# define various parameters
I = 64
J = 64
K = 33
L = 210

# we will be using CV to choose a bandwidth from:
H = c(5, 10, 15, 20, 25, 30, 35)

# randomly select 80% as training data the other 20% will be validation data
index = 1:210
per = round(210 * 0.8)

index_train = sample(index, per)
index_valid = index[-index_train]


# iterate over all bandwidths
for (h in H) {

    num = 0
    MSEs_train = c()
    MSEs_valid = c()


    for (i in 1:I) {
        for (j in 1:J) {
            for (k in 1:K) {

                # if mask == TRUE
                if (mask[i, j, k]) {
```

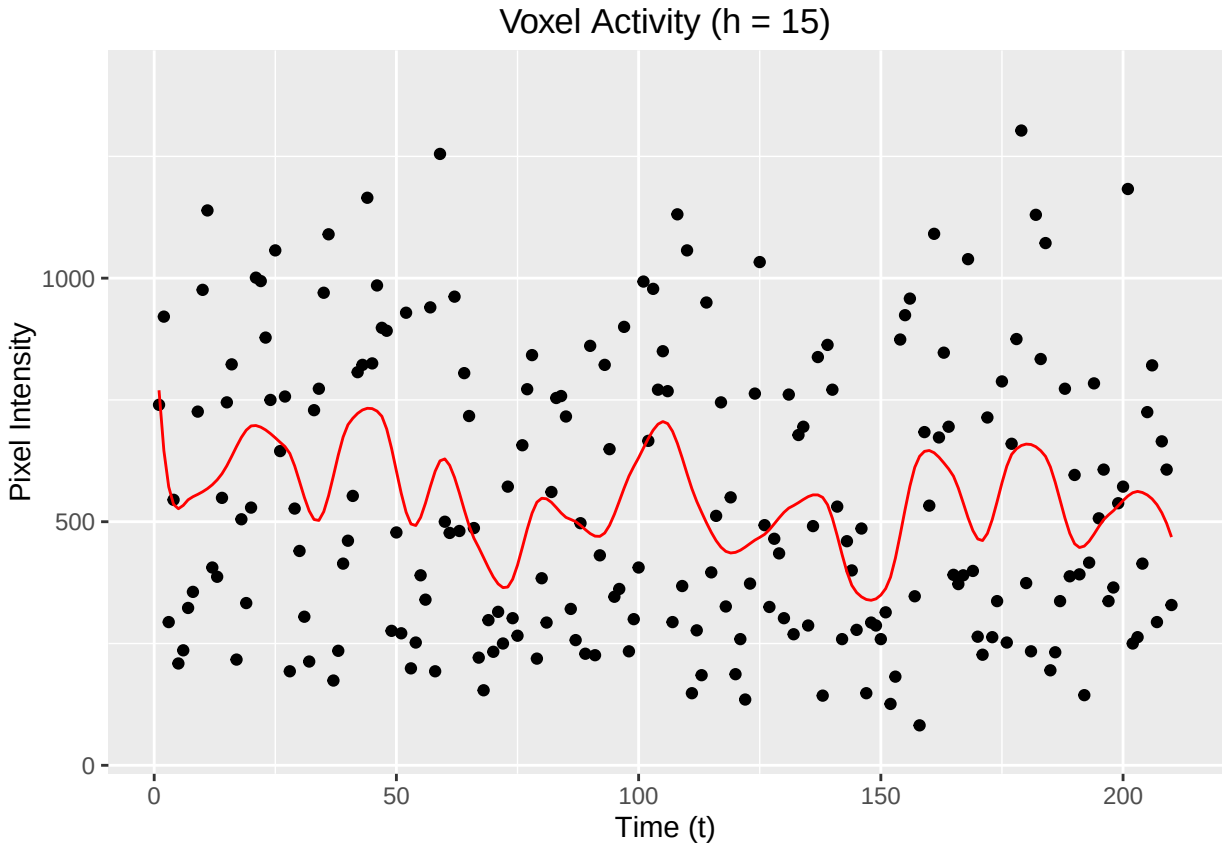
```
                # 210 observations to train from
                y = voxels[i, j, k, ]
                x = (1:L)

                # fit local polynomial model with specified bandwidth
                model = locfit(y[index_train] ~ lp(x[index_train], deg = 2,
                  h = h))

                # calculate training and validation error
                MSE_train = mean((predict(model, x[index_train]) - y[index_train])^2)
                MSE_valid = mean((predict(model, x[index_valid]) - y[index_valid])^2)

                MSEs_train = c(MSEs_train, MSE_train)
                MSEs_valid = c(MSEs_valid, MSE_valid)


            }
        }
    }

    # print out MSE's
    print(c(h, mean(MSEs_train), mean(MSEs_valid)))

}
```

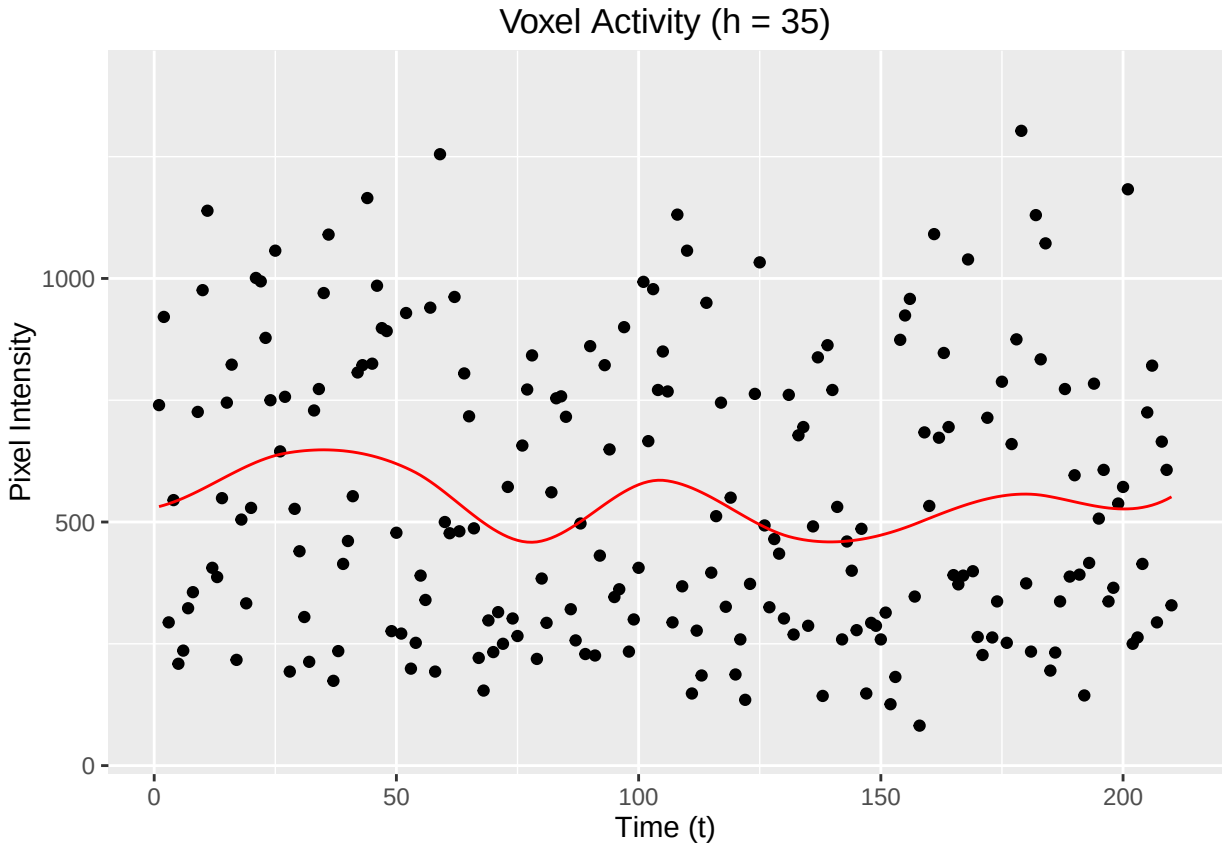
```
## [1]   5.0000 124.5537 328.9312
## [1]  10.0000 175.5725 270.0186
## [1]  15.0000 191.9506 256.8318
## [1]  20.0000 208.6300 255.5616
## [1]  25.0000 214.8675 254.8472
## [1]  30.0000 218.9134 253.6436
## [1]  35.0000 223.4867 252.2881
```

The results of our cross validation are presented above. For each bandwith, the average MSE was calculated for both the training and validation set. It is clear that we are overfitting to the training set quite heavily regrardless of bandwidth.

Consider the examples below:


Voxel Activity (h = 5)
Pixel Intensity
1000
500
0
0
50
100
150
200
Time (t)


Voxel Activity (h = 15)
Pixel Intensity
1000
500
0
0
50
100
150
200
Time (t)

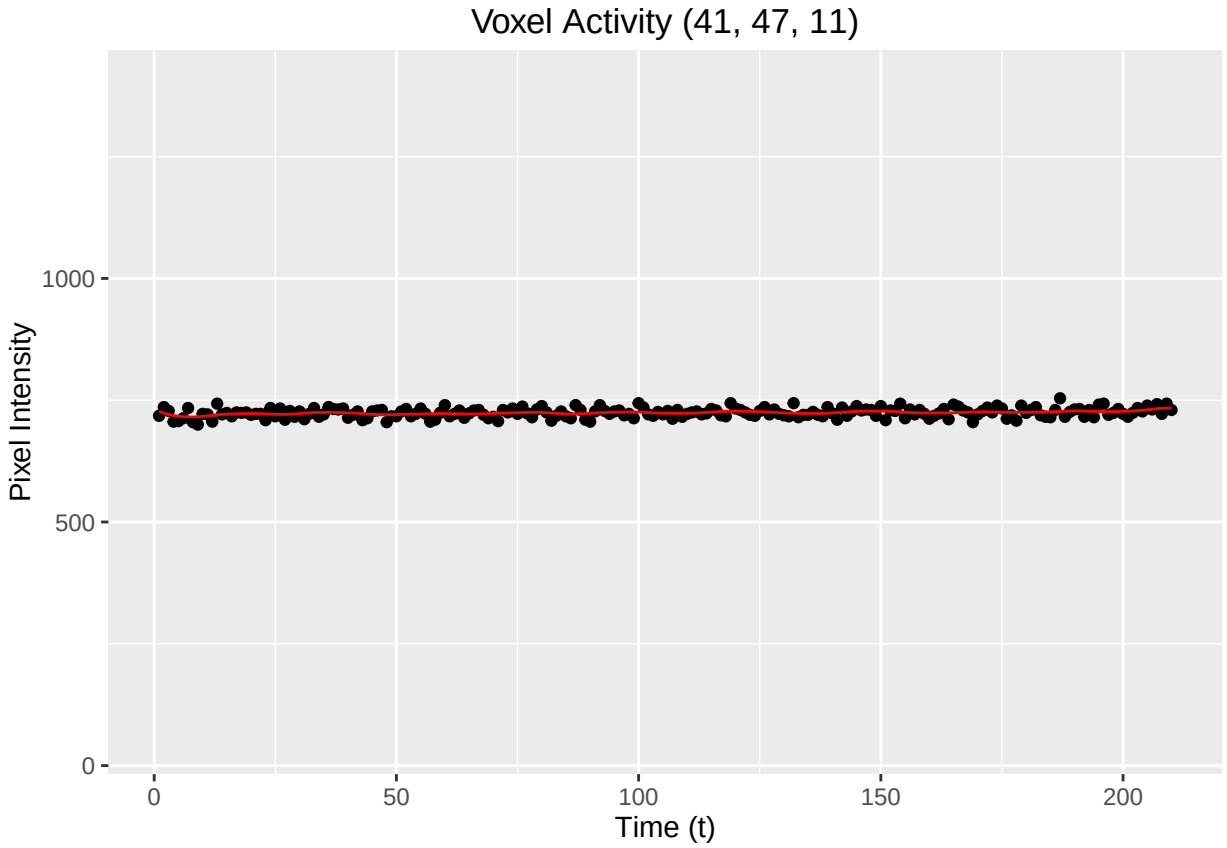


The plots above are created with bandwidths 5, 15, and 35, respectively.

It appears that the model with $h = 5$ is capturing a lot of noise where as the model with $h = 35$ is drowning out a lot of the signal. Moving forward, we will proceed with a bandwidth $h = 15$ because it offers a balance between predictive power and variance.

Now that we have selected a bandwidth, we can proceed to fit each voxel with our local polynomial model and measure the activity. In voxels with a large amount of activity, we expect that the hemodynamic response will oscillate with large fluctuations for each stimulus that the patient is shown (this is clear in the previous plots).

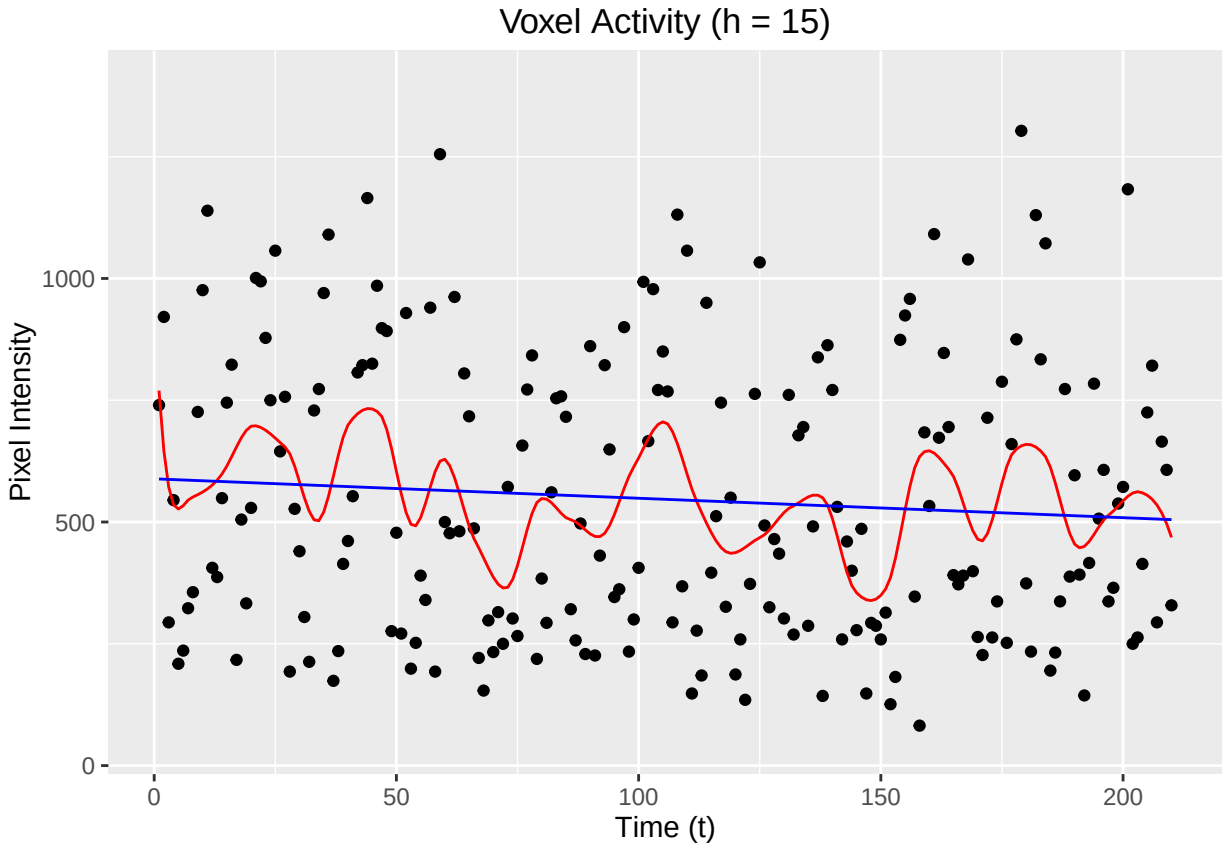
This is in stark contrast to a non-active (or less active) voxel. In these areas we expect the activity to remain relatively flat (see plot below - note that this plot was created using a bandwidth of $h = 15$).



Now the important question remains: how do we reliably measure brain activity and identify the voxels whose activity is particularly large? Is there a way to test whether the model fit has large versus small fluctuations?

One approach is to use two models! We can fit a local polynomial model and then based on those predicted values we can fit a linear model. This method allows us to calculate the MSE (or "distance") between the two. If the MSE is sufficiently large then we can conclude that the voxel is active. If the opposite is true (the MSE is small), that would suggest that there is minimal brain activity as a result of the stimulus.

An example of this method is presented below along with the relevant code for performing this procedure on all of the voxels.




```
# we choose bandwidth 15
h = 15

# iterate over all voxels
for (i in 1:I) {
    for (j in 1:J) {
        for (k in 1:K) {

            # if mask == TRUE
            if (mask[i, j, k]) {

                # 210 observations to train from
                y = voxels[i, j, k, ]
                x = (1:L)

                # fit local polynomial model with specified bandwidth
                model_poly = locfit(y ~ lp(x, deg = 2, h = h))

                # generate predictions for all 210 time points
                predictions = predict(model_poly, x)

                # fit linear model
                model_linear = lm(predictions ~ x)

                # we measure activity as the MSE between local polynomial and linear model
                activity[i, j, k] = mean((predictions - predict(model_linear,
                  data.frame(x)))^2)
```

```
            }
        }
    }
}
```

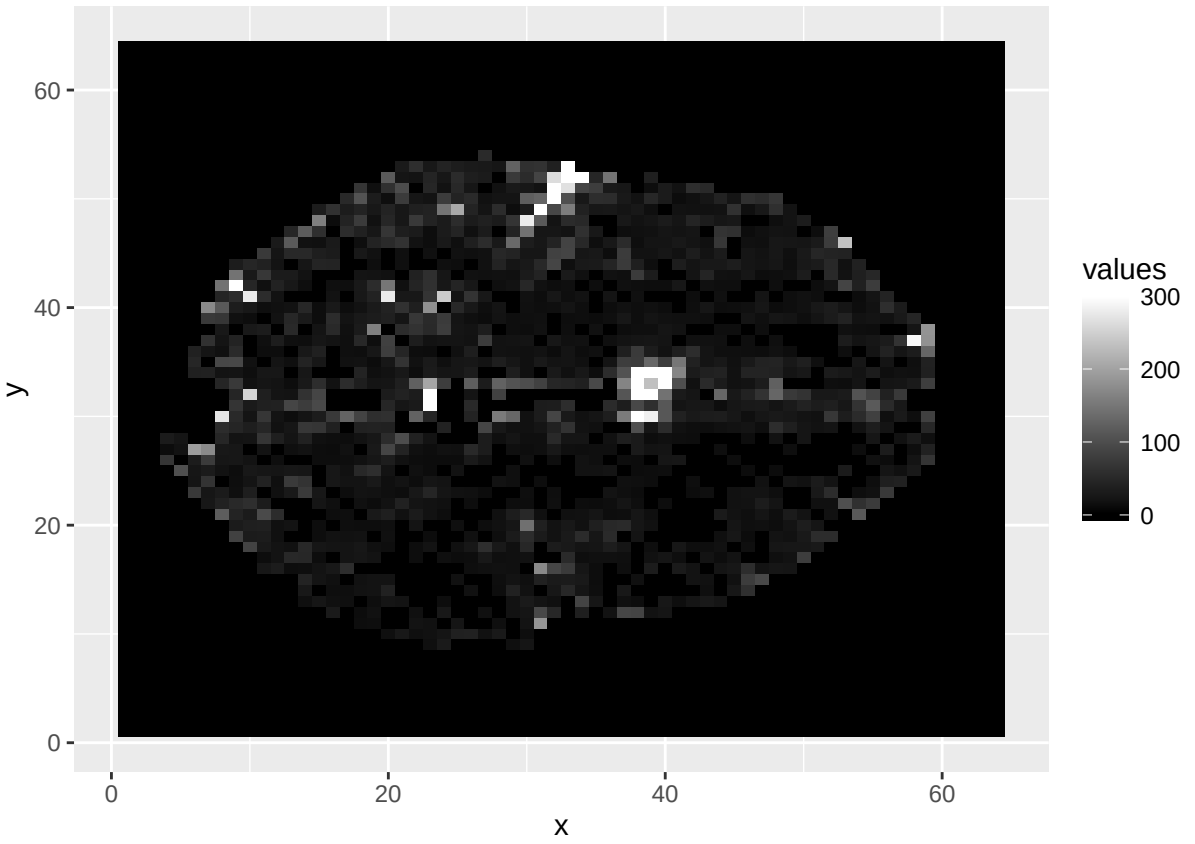
Now that we have a way of measure the activity for each voxel, we can analyze the results and draw conclusions about the overall affect of the stimulus!

Results

To illustrate the results, we have displayed a number of heatmaps. These are visuals that make it extremely easy to identify variations in the data (in this case variations in neural activity). Each heatmap represents a different horizontal "slice" of the brain. In medical literature this is referred to as an textit{axial} section of the brain.

In the plots, a darker color represents minimal activity whereas a bright, white color represents a large amount of activity. Note that the front of the brain is facing the left vertical edge of the image.

```
# load package into R
library(NeatMap)

#'slice' the dataset along to z-axis (for 2-D image)
slice = activity[, , 16]

# eliminate noise and cap the maximum MSE per voxel this is simply to make
# the plots look cleaner
slice[slice < 10] = 0
slice[slice > 300] = 300

# generate a heat map
heatmap1(slice) + scale_fill_gradient(low = "black", high = "white")
```

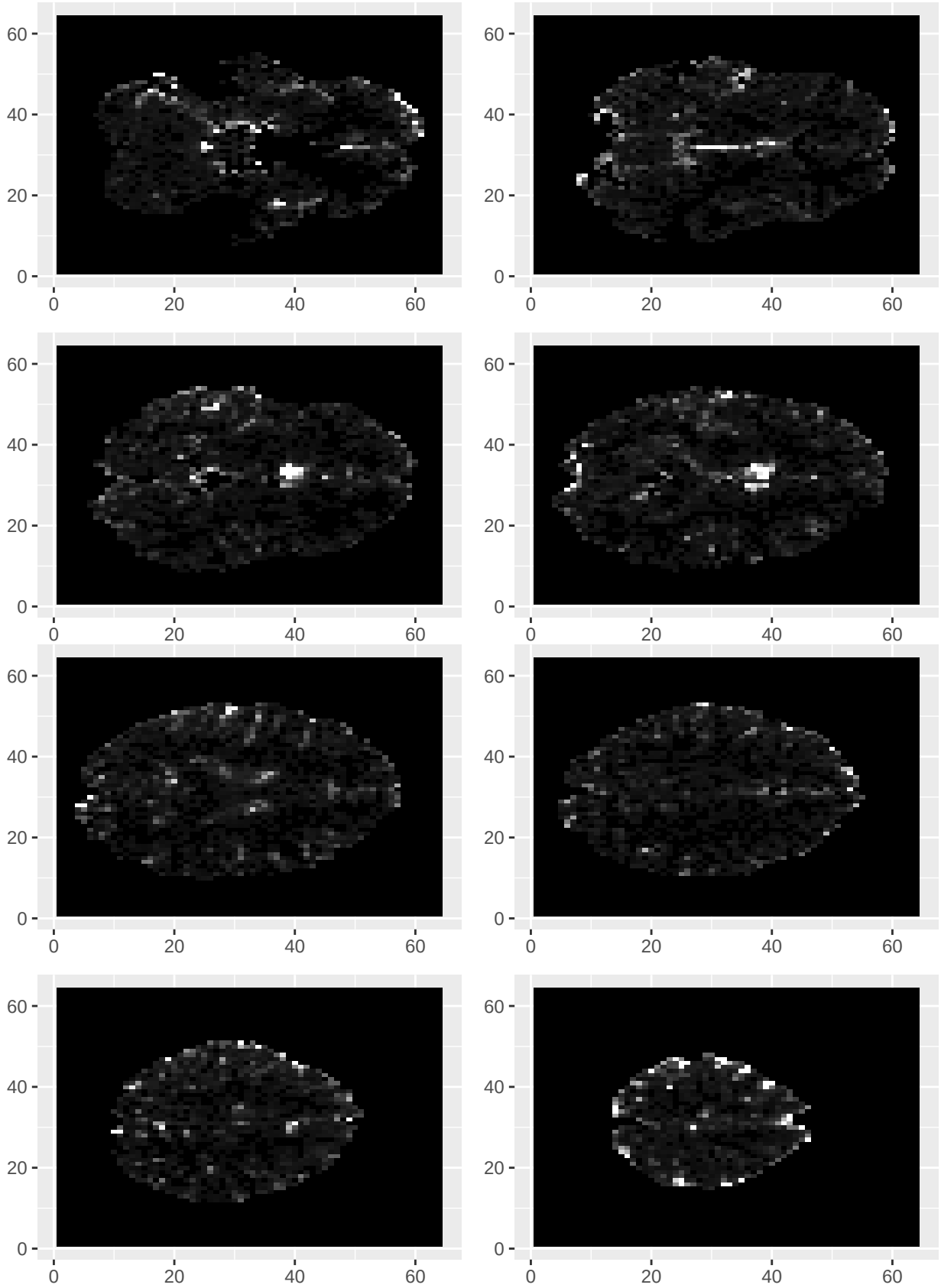


In this particular image we are looking at an axial section with a vertical z-axis equal to 16 – that is, approximately halfway betweeen the uppermost and lowermost portion of the brain. The most obvious observation to note is the large white ball directly in the middle of the brain. This is where we observed the most-concentrated activity throughout the study. Additionally, we can see a large amount of activity in the top and right left edge of the plot (right and front side of the brain).

The image below is a grid of heatmaps displaying axial sections at a vertical axis of 10, 13, 15, 17, 19, 22, 25, and 28, respectively. They represent the transition from the lower part of the brain to the top.

There are many observations we could note as we make the transition from the bottom to the top. One of these observations is that as we progress towards the top of the brain, much more of the activity appears to be concentrated on the perimeter. For instance, in axial sections 3 and 4 (row 2), there appears to be substantial activity in the center near the corpus callosum (the structure that connects the two hemispheres of the brain). In rows 3 and 4, more activity is registered on the edges – possibly because we are now capturing portions of the cerebral cortex (the outer layer/membrane of the brain). This is believed to be an area that contributes to thought, memory, attention, and consciousness [6].

Due to my lack of knowledge on the anatomy of the brain, I will refrain from offering any further conjectures about which portions of brain are activated and why that might be true. The goal of this project was to attempt to identify regions of activity through contemporary statistical models and I believe we achieved exactly that!


60
40
20
0
0
20
40
60

References

[1] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3938835/

[2] https://cran.r-project.org/web/packages/locfit/locfit.pdf

[3] https://cran.r-project.org/web/packages/fmri/fmri.pdf

[4] https://cran.r-project.org/web/packages/NeatMap/NeatMap.pdf

[5] https://openfmri.org/dataset/ds000117/

[6] https://en.wikipedia.org/wiki/Cerebral.cortex