Week 6 – First-level fMRI data analysis

L06-08. Doing GLM with custom regressors (regress.m)

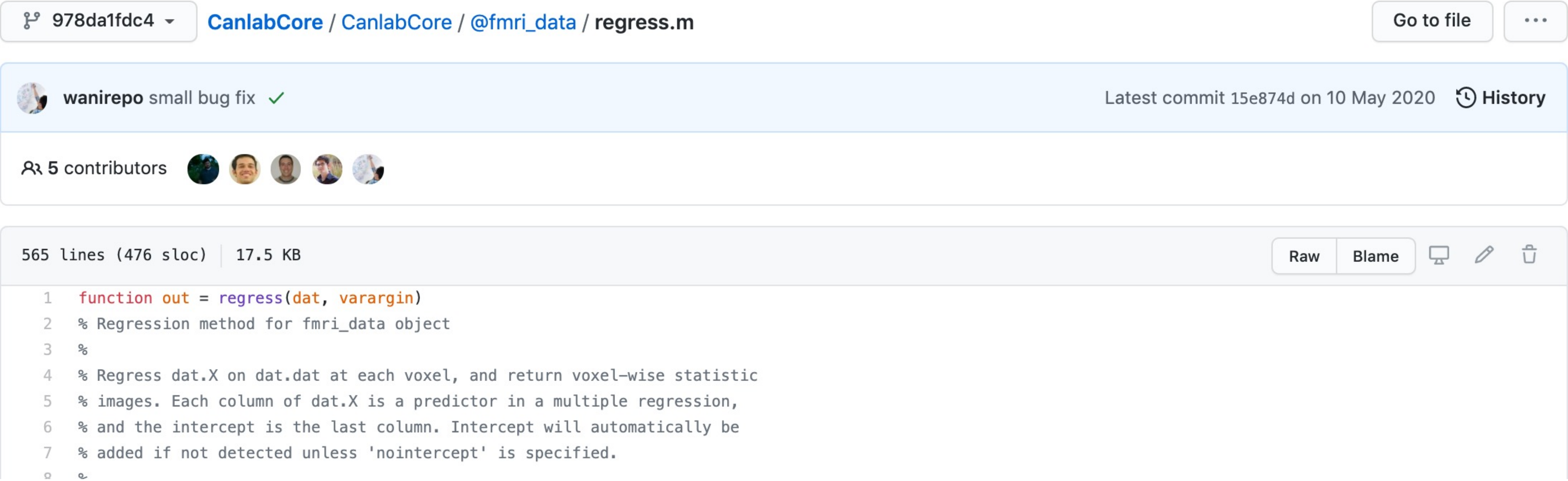
Hongji Kim

Ph. D. Student in the Cocoan lab

2 April 2021

L06-08. Doing GLM with custom regressors (regress.m)

- regress.m function in CanlabCore



CanlabCore / CanlabCore / @fmri_data / regress.m

wanirepo small bug fix ✓ Latest commit 15e874d on 10 May 2020

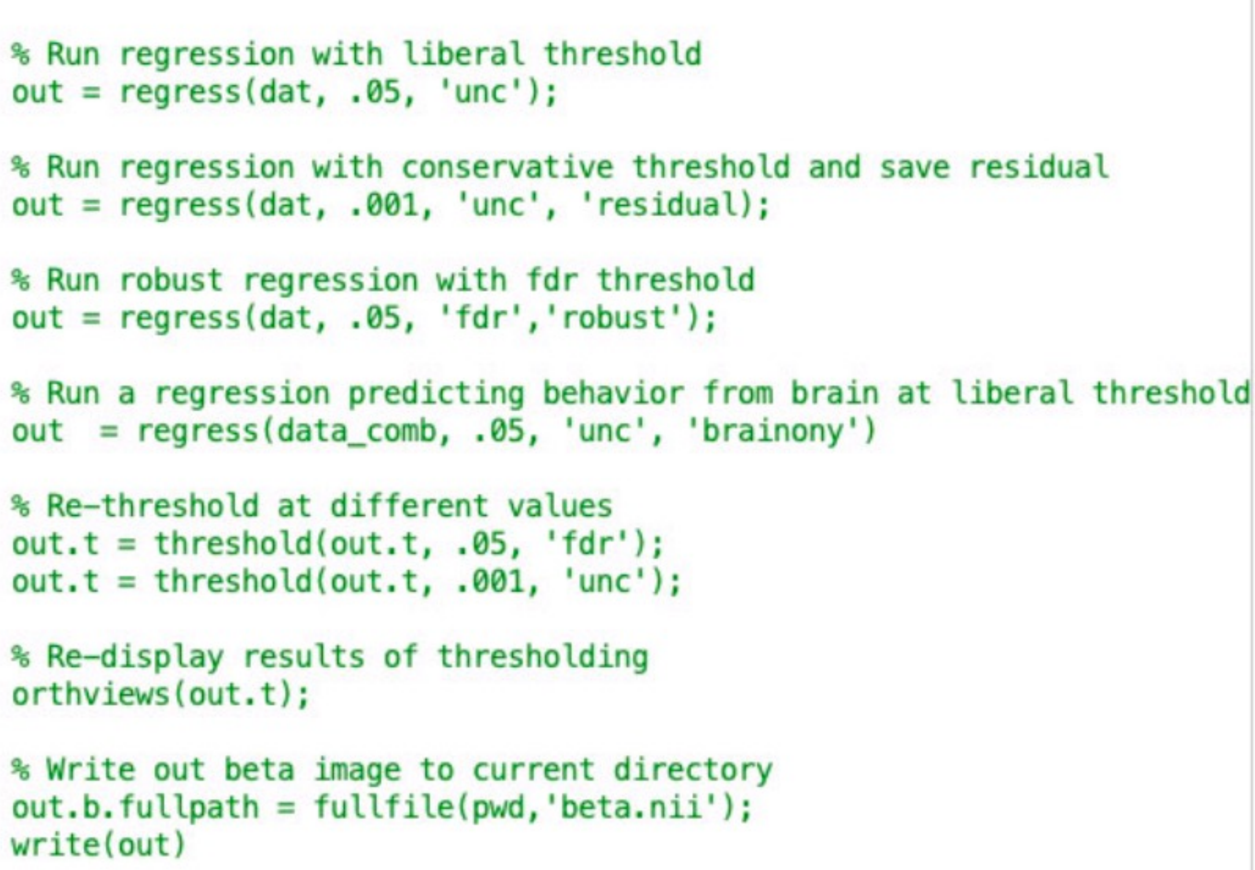
5 contributors

565 lines (476 sloc) | 17.5 KB

```
function out = regress(dat, varargin)
% Regression method for fmri_data object
%
% Regress dat.X on dat.dat at each voxel, and return voxel-wise statistic
% images. Each column of dat.X is a predictor in a multiple regression,
% and the intercept is the last column. Intercept will automatically be
% added if not detected unless 'nointercept' is specified.
%
```

L06-08. Doing GLM with custom regressors (regress.m)

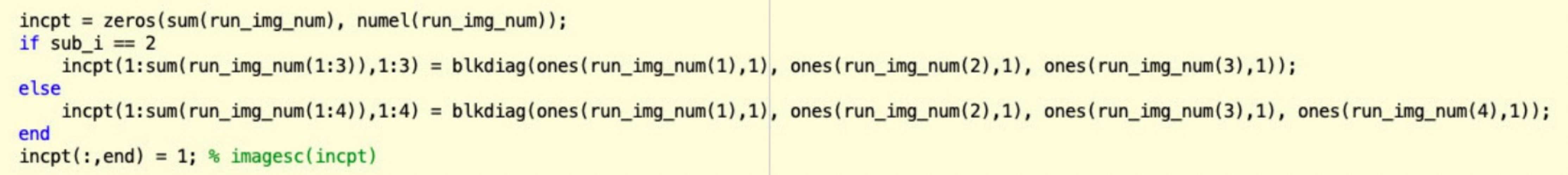
- regress.m function in CanlabCore



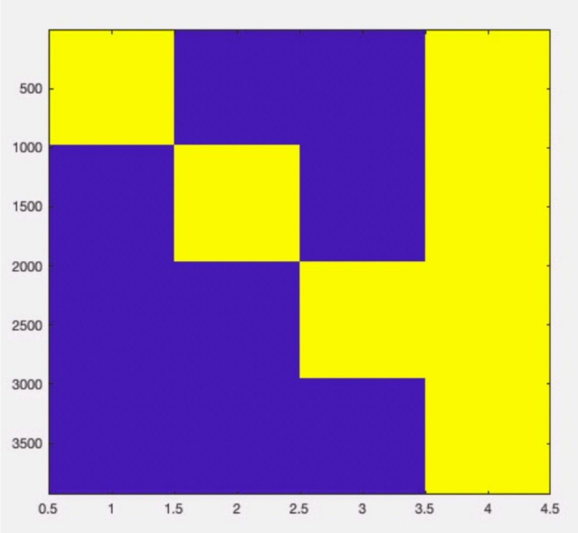
```
% Run regression with liberal threshold
out = regress(dat, .05, 'unc');

% Run regression with conservative threshold and save residual
out = regress(dat, .001, 'unc', 'residual);

% Run robust regression with fdr threshold
out = regress(dat, .05, 'fdr','robust');

% Run a regression predicting behavior from brain at liberal threshold
out  = regress(data_comb, .05, 'unc', 'brainony')

% Re-threshold at different values
out.t = threshold(out.t, .05, 'fdr');
out.t = threshold(out.t, .001, 'unc');

% Re-display results of thresholding
orthviews(out.t);

% Write out beta image to current directory
out.b.fullpath = fullfile(pwd,'beta.nii');
write(out)
```

L06-08. Doing GLM with custom regressors (regress.m)

- Actual codes
  - Caution: don’t use “/Applications/MATLAB_R2017b.app/toolbox/stats/stats/regress.m” !
  - After loading x variables, make intercept variable (run indicator)

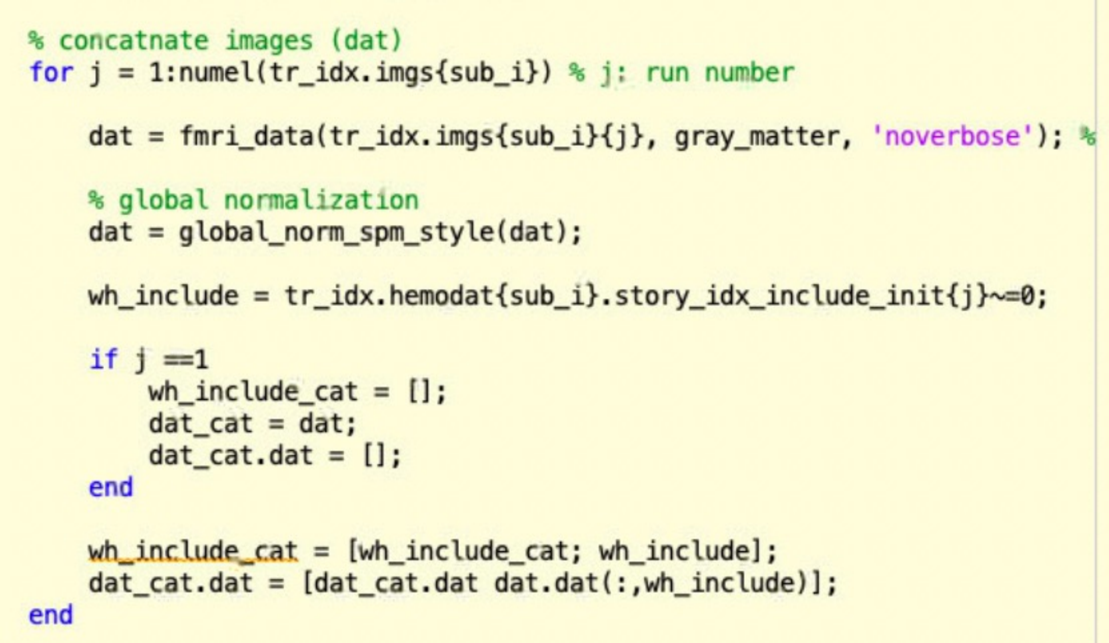


```
incpt = zeros(sum(run_img_num), numel(run_img_num));
if sub_i == 2
    incpt(1:sum(run_img_num(1:3)),1:3) = blkdiag(ones(run_img_num(1),1), ones(run_img_num(2),1), ones(run_img_num(3),1));
else
    incpt(1:sum(run_img_num(1:4)),1:4) = blkdiag(ones(run_img_num(1),1), ones(run_img_num(2),1), ones(run_img_num(3),1), ones(run_img_num(4),1));
end
incpt(:,end) = 1; % imagesc(incpt)
```

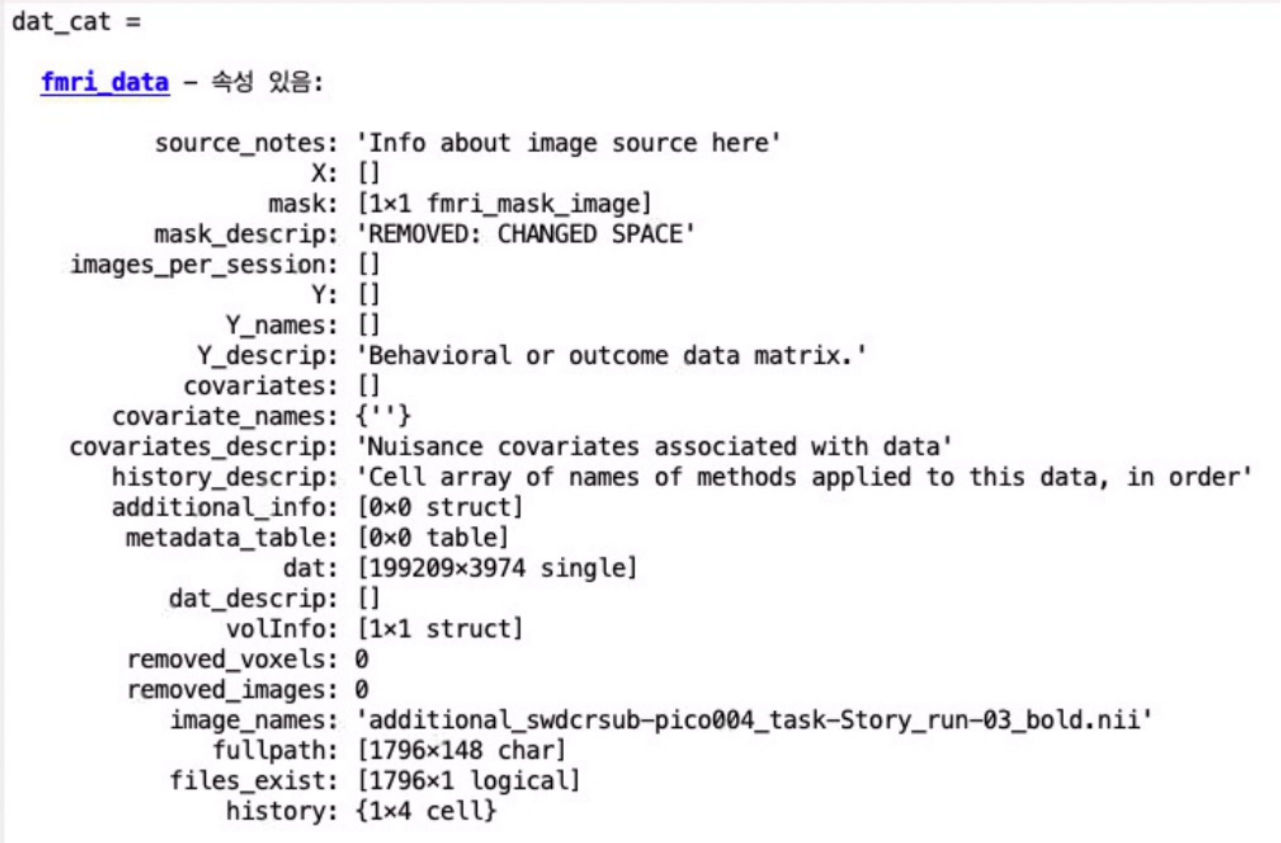


L06-08. Doing GLM with custom regressors (regress.m)

- Actual codes
  - Concatenate all runs in one participant



```
% concatnate images (dat)
for j = 1:numel(tr_idx.imgs{sub_i}) % j: run number

    dat = fmri_data(tr_idx.imgs{sub_i}{j}, gray_matter, 'noverbose'); %

    % global normalization
    dat = global_norm_spm_style(dat);

    wh_include = tr_idx.hemodat{sub_i}.story_idx_include_init{j}~=0;

    if j ==1
        wh_include_cat = [];
        dat_cat = dat;
        dat_cat.dat = [];
    end

    wh_include_cat = [wh_include_cat; wh_include];
    dat_cat.dat = [dat_cat.dat dat.dat(:,wh_include)];
end
```

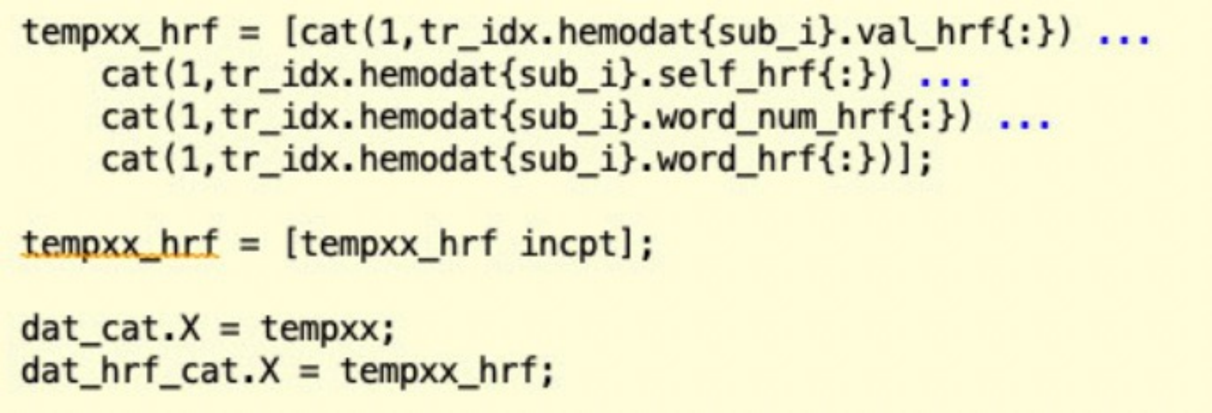


```
dat_cat =

  fmri_data - 속성 있음:

         source_notes: 'Info about image source here'
                    X: []
                 mask: [1×1 fmri_mask_image]
          mask_descrip: 'REMOVED: CHANGED SPACE'
    images_per_session: []
                    Y: []
              Y_names: []
             Y_descrip: 'Behavioral or outcome data matrix.'
            covariates: []
       covariate_names: {''}
    covariates_descrip: 'Nuisance covariates associated with data'
       history_descrip: 'Cell array of names of methods applied to this data, in order'
       additional_info: [0×0 struct]
        metadata_table: [0×0 table]
                   dat: [199209×3974 single]
           dat_descrip: []
               volInfo: [1×1 struct]
       removed_voxels: 0
       removed_images: 0
           image_names: 'additional_swdcrsub-pico004_task-Story_run-03_bold.nii'
              fullpath: [1796×148 char]
           files_exist: [1796×1 logical]
               history: {1×4 cell}
```

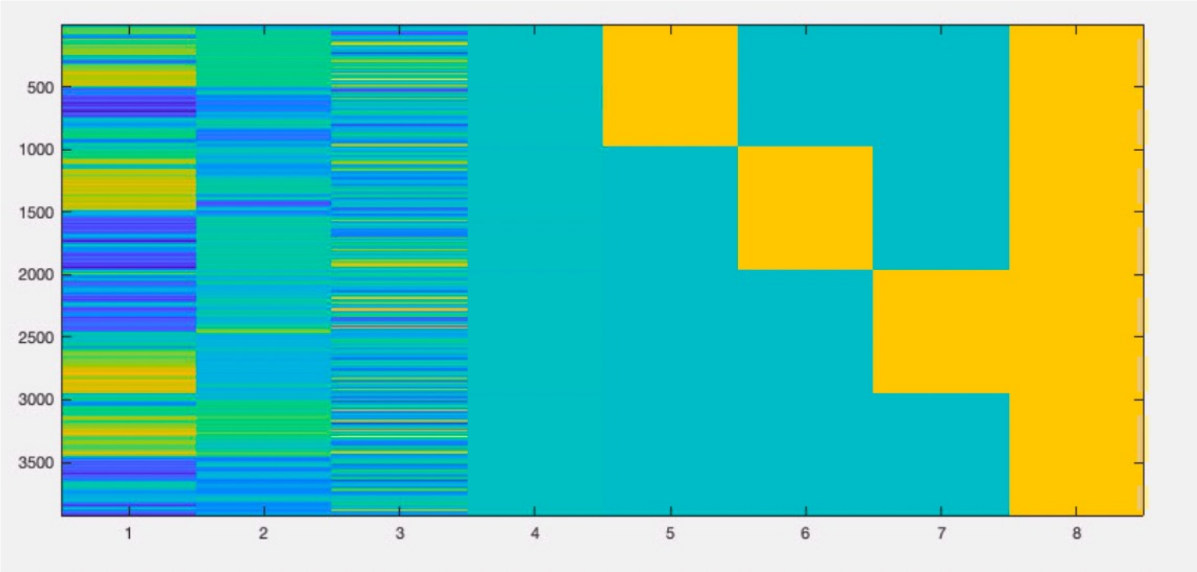
L06-08. Doing GLM with custom regressors (regress.m)

❖ Actual codes

- Make regressors into whole-run level as well (concatnate)



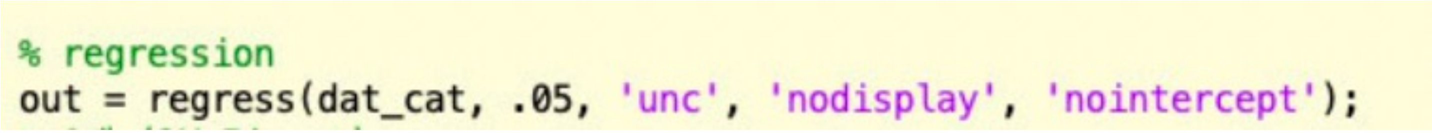
```
tempxx_hrf = [cat(1,tr_idx.hemodat{sub_i}.val_hrf{:}) ...
    cat(1,tr_idx.hemodat{sub_i}.self_hrf{:}) ...
    cat(1,tr_idx.hemodat{sub_i}.word_num_hrf{:}) ...
    cat(1,tr_idx.hemodat{sub_i}.word_hrf{:})];

tempxx_hrf = [tempxx_hrf incpt];

dat_cat.X = tempxx;
dat_hrf_cat.X = tempxx_hrf;
```



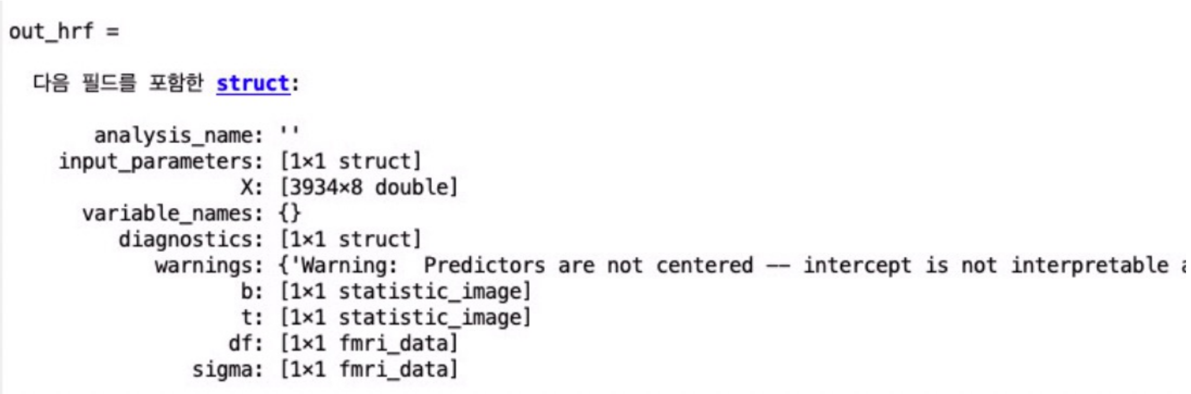
L06-08. Doing GLM with custom regressors (regress.m)

❖ Actual codes

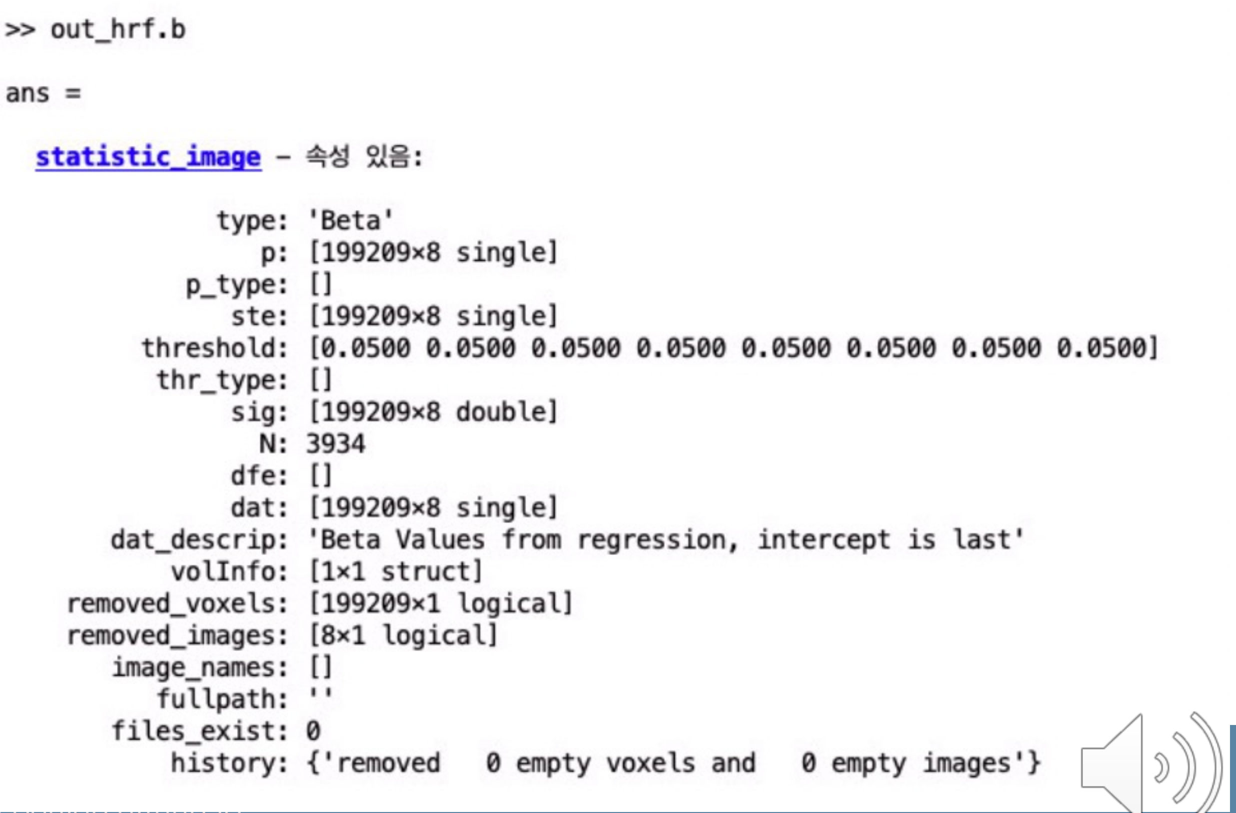
- Do the regression!



```
% regression
out = regress(dat_cat, .05, 'unc', 'nodisplay', 'nointercept');
```



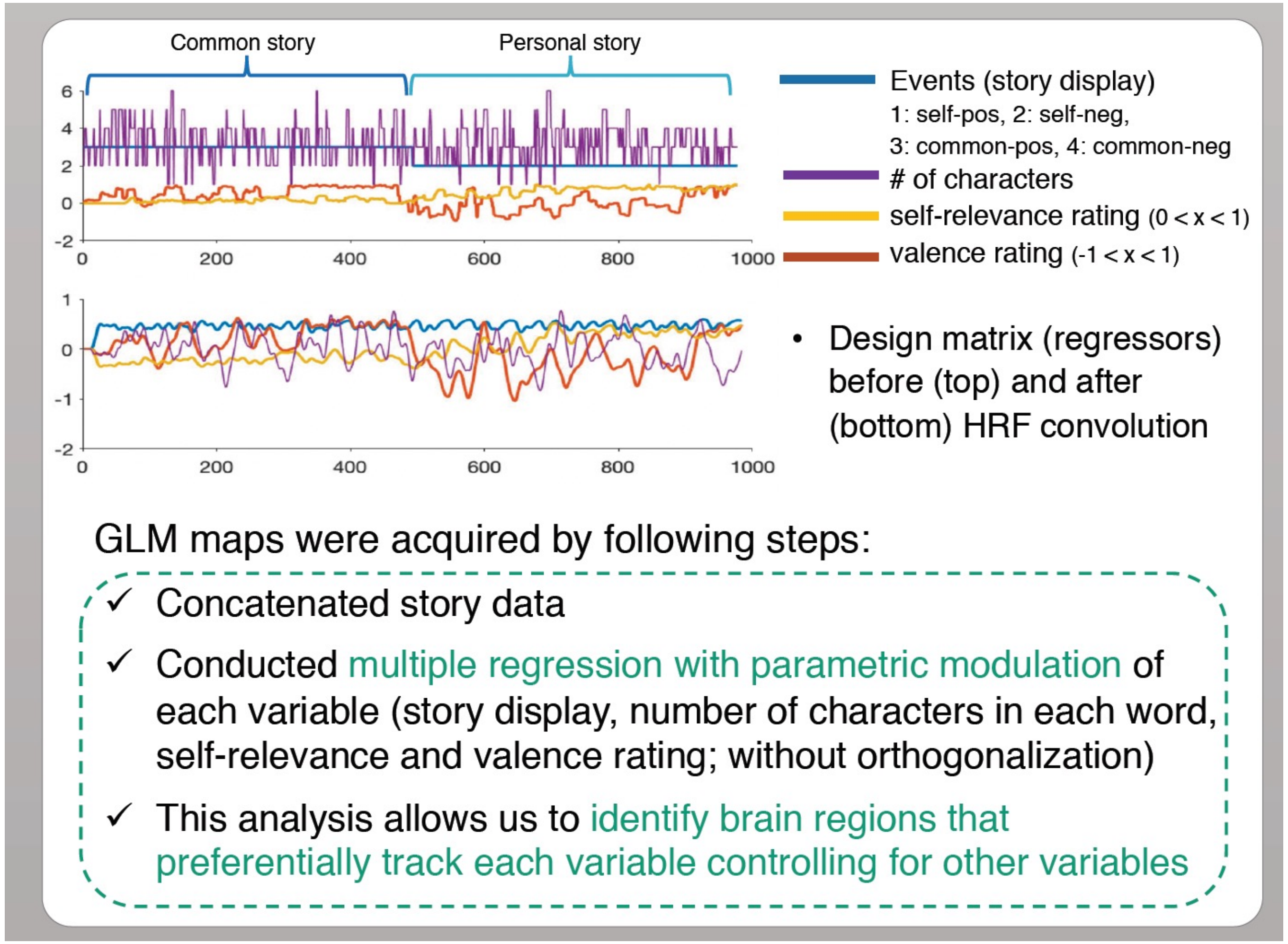
```
out_hrf =

  다음 필드를 포함한 struct:

      analysis_name: ''
   input_parameters: [1×1 struct]
                  X: [3934×8 double]
     variable_names: {}
        diagnostics: [1×1 struct]
           warnings: {'Warning:  Predictors are not centered -- intercept is not interpretable
                  b: [1×1 statistic_image]
                  t: [1×1 statistic_image]
                 df: [1×1 fmri_data]
              sigma: [1×1 fmri_data]
```



```
>> out_hrf.b

ans =

  statistic_image - 속성 있음:

               type: 'Beta'
                  p: [199209×8 single]
             p_type: []
                ste: [199209×8 single]
          threshold: [0.0500 0.0500 0.0500 0.0500 0.0500 0.0500 0.0500 0.0500]
           thr_type: []
                sig: [199209×8 double]
                  N: 3934
                dfe: []
                dat: [199209×8 single]
        dat_descrip: 'Beta Values from regression, intercept is last'
            volInfo: [1×1 struct]
    removed_voxels: [199209×1 logical]
    removed_images: [8×1 logical]
        image_names: []
           fullpath: ''
        files_exist: 0
            history: {'removed    0 empty voxels and    0 empty images'}
```

L06-08. Doing GLM with custom regressors (regress.m)

- ❖ Example result



Events (story display)
1: self-pos, 2: self-neg,
3: common-pos, 4: common-neg
# of characters
self-relevance rating (0 < x < 1)
valence rating (-1 < x < 1)

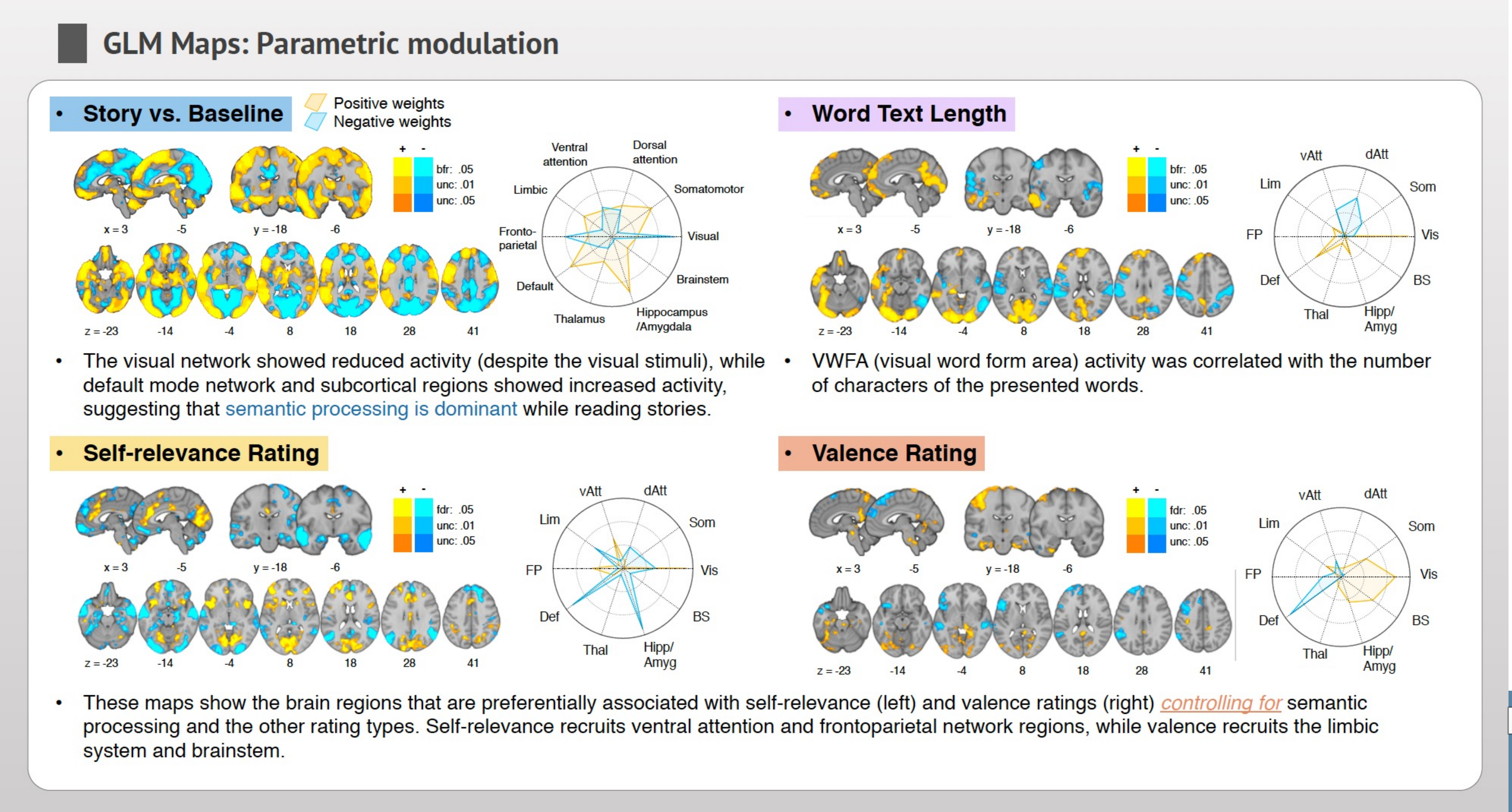
- Design matrix (regressors) before (top) and after (bottom) HRF convolution

GLM maps were acquired by following steps:

- ✓ Concatenated story data
- ✓ Conducted multiple regression with parametric modulation of each variable (story display, number of characters in each word, self-relevance and valence rating; without orthogonalization)
- ✓ This analysis allows us to identify brain regions that preferentially track each variable controlling for other variables

L06-08. Doing GLM with custom regressors (regress.m)

- Example result



GLM Maps: Parametric modulation

- **Story vs. Baseline**

- The visual network showed reduced activity (despite the visual stimuli), while default mode network and subcortical regions showed increased activity, suggesting that semantic processing is dominant while reading stories.

- **Word Text Length**

- VWFA (visual word form area) activity was correlated with the number of characters of the presented words.

- **Self-relevance Rating**

- **Valence Rating**

- These maps show the brain regions that are preferentially associated with self-relevance (left) and valence ratings (right) controlling for semantic processing and the other rating types. Self-relevance recruits ventral attention and frontoparietal network regions, while valence recruits the limbic system and brainstem.

Cocoan 101

https://cocoanlab.github.io