BL5229: Data Analysis with Matlab
Lab: data analysis and data modeling

Problem 1: Checking if a method is statistically better than another.

Finding efficient algorithms to describe, measure and compare shapes is a central problem in image processing. This problem arises in numerous disciplines that generate extensive quantitative and visual information. Among these, biology occupies a central place. For example, registration of brain anatomy is essential to many studies in neurobiology.

Here we consider the problem of aligning 38 brains (defined by their cortical surface) onto a template brain. Each brain consists of two independent hemispheres, Left, and Right. We have two methods for aligning the hemisphere of a brain onto the corresponding hemisphere of the template, Method1 and Method2. For each method, we align the hemisphere onto the template and check the quality of the alignment by comparing the positions of 35 regions between the aligned brain and the template. Figure 1 illustrates the positions of these regions, while Figure 2 shows one example of an alignment.

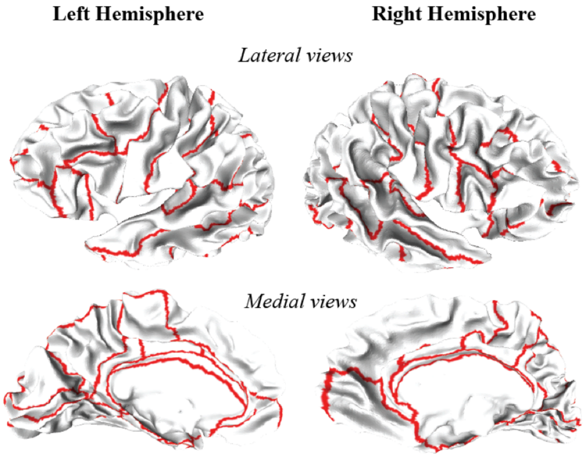


Figure 1: The 35 regions of the brain

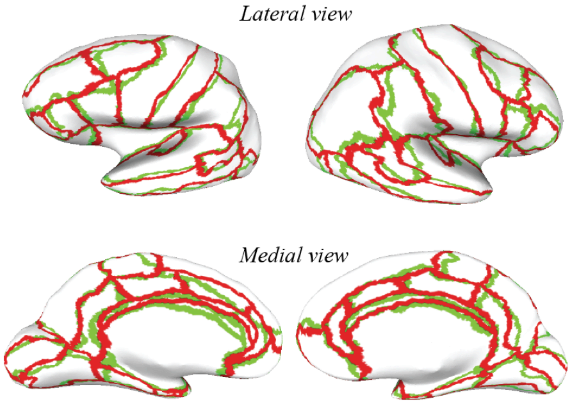


Figure 2: One alignment between a brain (green) and the template (red), shown on the inflated brain.

We will only consider the left hemisphere in this exercise. The files Left1.dat and Left2.dat (available on the web page) give the results of the alignments for Method1 and Method2, respectively. Each file includes a large table with 35 rows, for the 35 regions of interest, and 38 columns, for the 38 brains considered. The value for a row I and a column J gives the distance between the region I in brain J and the same region in the template brain. Note that the lower the distance, the better.

For each region, compare the two distributions of distance values obtained over all 38 brains for the two alignment methods, indicate if these distributions are significantly different (at 95% confidence level), and which of the two methods works best. How would you present these results?

Script for generating results:

```
%
% Read in data from both methods
%
load Left1.dat
load Left2.dat
%
% Find number of regions
%
s = size(Left1);
nregions = s(1);
nbrains  = s(2);
%
% Build rtable table
%
rtable = cell(36,6);
%
rtable{1,1} = 'Region';
rtable{1,2} = 'Mean (method 1)';
rtable{1,3} = 'Mean (method 2)';
rtable{1,4} = 'Best method';
rtable{1,5} = 'Significance';
rtable{1,6} = 'P-value';
%
% Analyse all regions
%
for i = 1:nregions
        val1 = Left1(i,:);
        val2 = Left2(i,:);
        m1   = mean(val1);
        m2   = mean(val2);
        rtable{i+1,1}=i;
        rtable{i+1,2}=m1;
        rtable{i+1,3}=m2;
        if m1 > m2
                rtable{i+1,4} = 'Method 2';
        else
                rtable{i+1,4} = 'Method 1';
        end
        [h,p]=ttest2(val1,val2);
        rtable{i+1,6} = p;
```

```
    if p < 0.01
        rtable{i+1,5} = 'Y';
    else
        rtable{i+1,5} = 'N';
    end
end

%
% Generate plot as a bar graph that shows the difference of the mean, with significant
% differences in blue, and non significant differences in red
%
n_significant = 0
n_nonsignificant = 0
for i = 1:35
     if rtable{i+1,6} < 0.01
          n_significant = n_significant + 1;
          region_sig(n_significant) = i;
          diff_sig(n_significant) = rtable{i+1,2}-rtable{i+1,3};
     else
          n_nonsignificant = n_nonsignificant + 1;
          region_nonsig(n_nonsignificant) = i;
          diff_nonsig(n_nonsignificant) = rtable{i+1,2}-rtable{i+1,3};
      end
end
%
bar(region_sig,diff_sig,’b’);
hold on
bar(region_nonsig,diff_nonsig,’r’)
%
xlabel(‘Region #’);
ylabel(‘Mean (method1) – Mean(method2)’);
legend(‘Significant’,’Not significant’);
title(‘Comparing two methods for brain surface matching’);
```

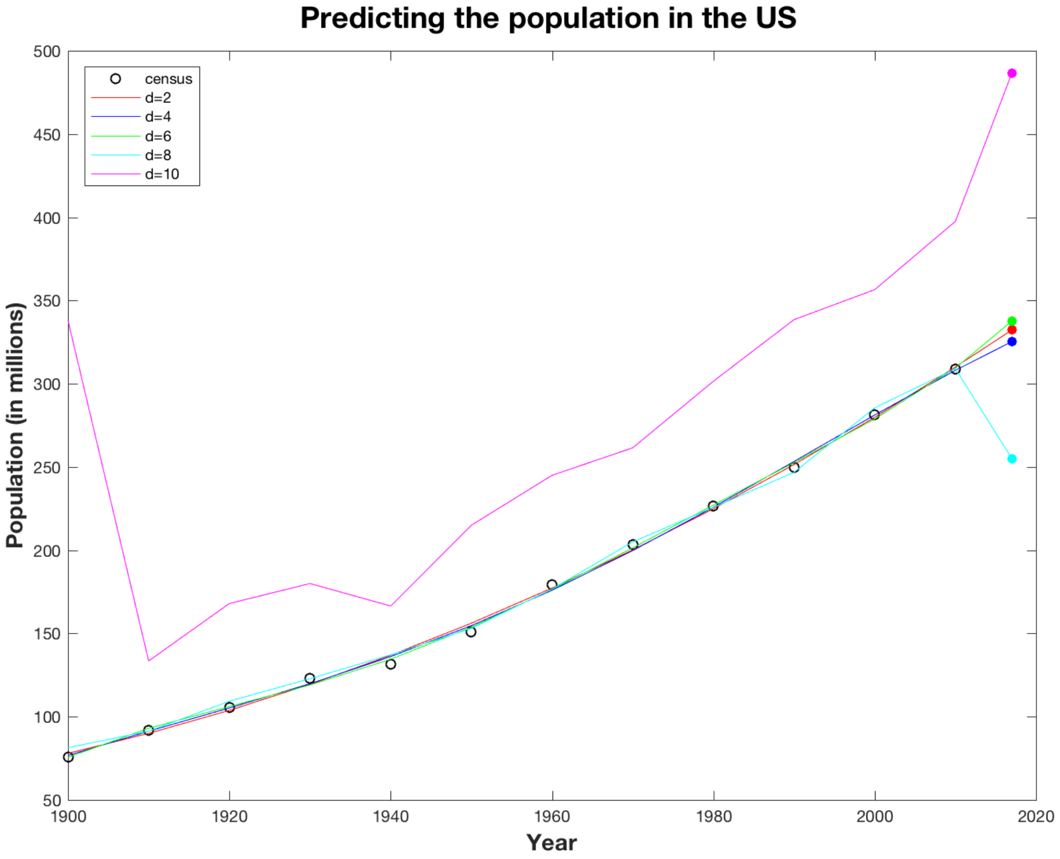
Problem 2: Polynomial fitting

In the file census.dat (available on the web site), I have stored the census data for the population in the US between 1900 and 2010, by step. Our goal is to try to predict the population in the US in the year 2017. This population is 324.8 millions. We will use and discuss a naïve approach to this problem: find a polynomial fit to the data, where the polynomial may be of order 2 to 10, and compute the value of that polynomial for the year 2020. Do read carefully the help for the two main MATLAB functions you will need:
polyfit
and
polyval.

Once you understood those functions, write a MATLAB script that allows you to fill in this table:

Polynomial order	Predicted population in 2020
2	**332.2**
4	**325.3**
6	**337.6**
8	**255.1**
10	**486.5**

Draw on the same plot the different fits obtained for polynomials of degree 2, 4, 6, 8, and 10. Comment your results.


Predicting the population in the US
Population (in millions)
Year
census
d=2
d=4
d=6
d=8
d=10