Antonio Remiro Azócar - Computational Modelling for Biomedical Imaging: Coursework 2.2

Q1

(a) The random seed is set to eleven for result reproducibility. A t-statistic of -3.63 and a p-value of 0.0034 is obtained via the function `ttest2`. These values are appropriate; the large magnitude of the t-statistic in comparison to its table values (for its corresponding degrees of freedom, 14) corresponds to the sample means being identified as different.

(b) 1D array `D` stores 14 observations; the first six are stored as `group1` and the last eight as `group2`. The valid permutations of `D` for group 1 are stored as `C1`. These are computed via `C1 = combnk(D, 6)`. The corresponding permutations of `D` for group 2 are the elements in `D` not in each `C1`. Iterating over `length(C1)` elements and using the function `setdiff`, the valid permutations of `D` for group 2 are computed. The t-statistic between each permutation for groups 1 and 2 is computed via the function `ttest2` and stored in a list. Figure 1 (in the Appendix) displays the empirical distribution of the t-statistic with a histogram. As expected, the distribution is Gaussian. An exact permutation-based p-value of 0.0037 is obtained. This value is reasonable, being higher than that obtained in **1a** and within one order of magnitude from it.

(c) A p-value of 0.0037 is obtained using the difference between the means as the test statistic. This value is identical as that obtained for **1b**. Figure 2 displays the empirical distribution of the difference between the means with a histogram. As expected, the distribution is Gaussian. Heuristically, I believe the higher p-values for **1b**, **1c** wrt **1a** correspond to our **1a** estimation being overconfident. The central limit theorem assumption for the parametric **1a** triggers this and is not appropriate for so small sample sizes. We can take the p-values for **1b**, **1c** as 'better' estimators than parametric **1a**.

(d) The MATLAB function `randperm` generates a random set of integer permutations. These are utilised to assign samples to different groups. For part (i), a p-value of 0.003 is obtained with 1,000 permutations and using the t-statistic. Figure 3 displays the empirical distribution of the t-statistic with a histogram. As expected, the distribution is Gaussian. Part (i) generates duplicate permutations corresponding to the same elements of a group in different orders. For this specific example, we obtain 178 duplicates for 1,000 permutations. The p-value obtained for (i) is lower than that obtained for **b**, **c**.

We obtain 1,000 unique permutations as follows. We repeat the procedure at (i) for 1,500 permutations. Then sort the rows and use the MATLAB function `unique` to discard the duplicates. Consequently, the first 1,000 rows of our remaining permutations are selected. The new p-value obtained is $p = 0.005$; higher than that accounting for duplicates. Figure 4 displays the empirical distribution of the t-statistic with a histogram. As expected, the distribution is Gaussian.

Q2

(a) Our GLM of choice is $Y = X_1\beta_1 + X_2\beta_2 + e$. Its design matrix has two columns and 16 rows. The first element of the first 8 rows is one; the second element is zero. The second element of the last 8 rows is one; the first element is zero. We compute $\hat{\beta} = (X^TX)^{-1}X^TY$ and $\hat{e} = Y - \beta X^T$. Y is a column vector with the 16 measurements (subjects) for each voxel. The t-statistic for each voxel is stored in a list, being determined via $\frac{\lambda^T\hat{\beta}}{\sqrt{\lambda^T S_{\hat{\beta}}\lambda}}$, where the contrast vector $\lambda = [1 - 1]^T$ and the covariance matrix $S_{\hat{\beta}} = \hat{\sigma}^2(X^TX)^{-1} = \frac{\hat{e}^T\hat{e}(X^TX)^{-1}}{n-\dim(X)}$. The denominator corresponds to $n - p = 16 - 2 = 14$ degrees of freedom.

We only keep the t-statistics in the ROI. (those corresponding to non-zero entries in `wm_mask.img`). The maximum t-statistic among all the voxels is $t_{max} = 6.529$.

(b) We must firstly iterate over all subject combinations. Group indices for these combinations are created via the functions `combnk` and `setdiff` (like in **1b**). Once these combinations have been created, we loop over each permutation of group labels and compute the t-statistic as in **2a** (using X, $\hat{\beta}$,$\hat{e}$,λ, $S_{\hat{\beta}}$). As in **2a**, there is no need to manually loop over each voxel (e.g. using a `for` loop). The problem has been heavily vectorised and takes around 5 minutes to run. The maximum t-statistic for each permutation is stored in a list. The histogram in Figure 5 displays the empirical distribution of the maximum t-statistic over all permutations. As expected, the distribution is Gaussian.

(c) The multiple-comparisons-corrected p-value is determined by finding the percentage of the permutations with a maximum t-statistic greater than that of the original labeling. We obtain an empirical p-value of 0.0918

(d) The maximum t-statistic threshold corresponding to a p-value of 5% is computed using the MATLAB function `prctile`. The threshold is calculated to be 6.94.

Figures (Appendix)

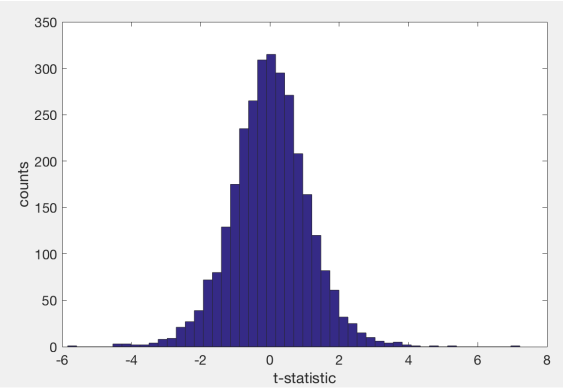

Figure 1: Empirical distribution of t-statistic for **1b**.

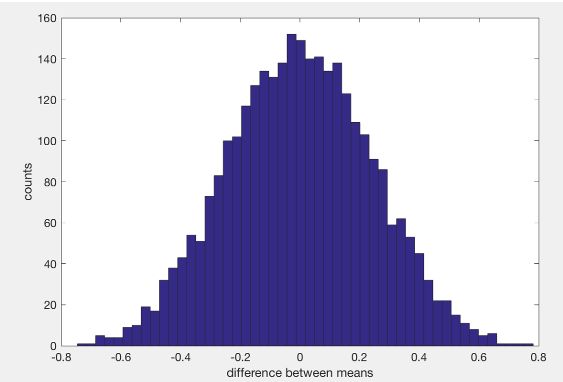

Figure 2: Empirical distribution of mean diff. for **1c**.

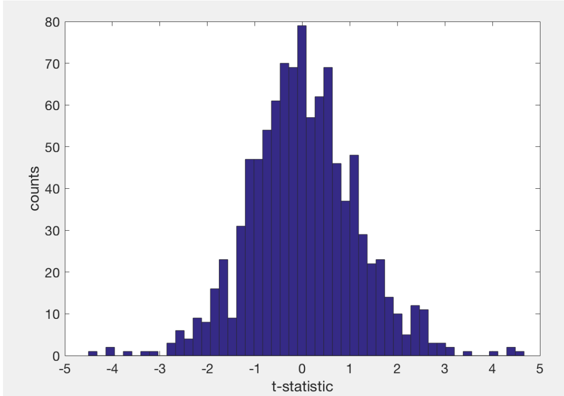

Figure 3: Empirical distribution of t-statistic for **1di**.

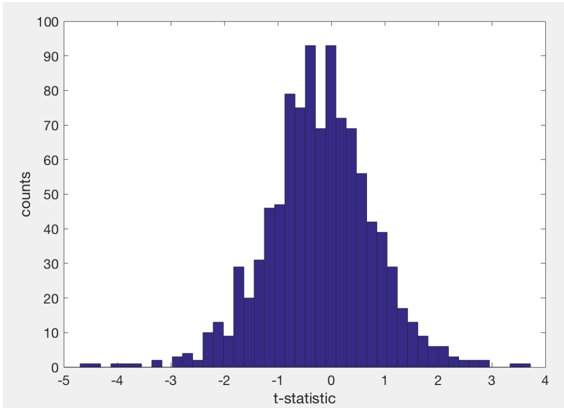

Figure 4: Empirical distribution of t-statistic for **1diii**.

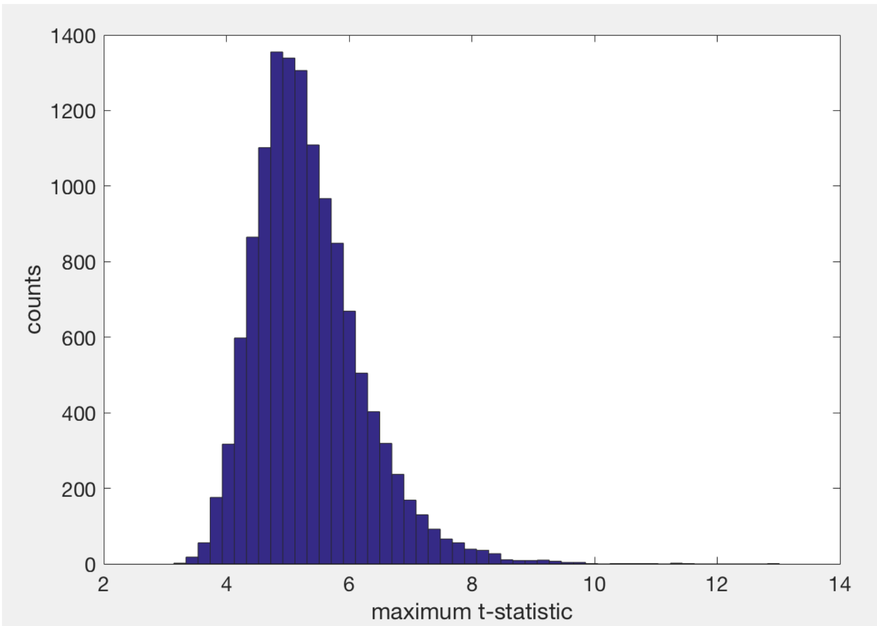

Figure 5: Empirical distribution of max. t-statistic for **2b**.