

Normative modeling in Schizophrenia - Analysis

Noemi González Lois

2021-07-30

Packages and libraries

We get a list of all the needed packages, then we install the ones that are not yet and load them all. In the end, we get the number of cores available in PC, which we will be using later on to parallelize.

```
library("easypackages")

list.of.packages <- c("viridis", "tidyverse", "MatchIt", "grid", "png",
                      "gridExtra", "parallel", "nlme", "JMbajes", "plyr",
                      "BiocManager", "Biostrings", "lme4", "ggplot2", "dplyr",
                      "Hmisc", "devtools", "longCombat", "neuroCombat",
                      "tinytex", "knitr", "ggExtra", "variancePartition",
                      "Rmisc", "doParallel", "hrbrthemes", "cowplot",
                      "BiocParallel", "effectsize")

new.packages <- list.of.packages[!(list.of.packages %in%
                                         installed.packages() [, "Package"])]
if(length(new.packages)>0) { install.packages(new.packages)}

libraries(list.of.packages)
ncores <- detectCores()
rm(list.of.packages, new.packages)
```

Set the working directory and load functions

```
setwd("/data_J/Scripts")
source("1_DataPreparation.R")
source("2_RegressionModel.R")
source("3_Statistics.R")
source("4_EDA.R")
```

Data preparation

There are two groups of data, and they both contain the same covariates but differ in the parcellation used when processing the image.

All covariates are stored in two datasets (planU_morphosim_rawdatabase.csv and planU_euler.csv), and the info of each parcellation scheme is organised as follows:

- **68 regions parcellation:** images are processed so that each brain hemisphere is divided into 34 regions. We have the cortical thickness measure for each brain region for each subject stored in two datasets (lh.aparc.thickness.csv, rh.aparc.thickness.csv), one corresponding to each brain hemisphere.
- **308 regions parcellation:** images are processed so that the whole brain is divided into 308 equally sized regions. We do also have the cortical thickness for each brain region for each subject stored, this time, in a single dataset (planU_morphosim_rawdatabase.csv).

We built the DataPreparation() function, which aims to do all the first preparation steps in order to use the data later on. The options when using this function are:

- parc = “parc68” or “parc308” (whether to use the 68 regions parcellation or the 308 regions one)
- harmonization= “lC” or “nC” (whether to apply longCombat or NeuroCombat harmonization)
- match = T or F (whether to use match-it or not)

So, what we first need to do is merge the different datasets to get a final one with all the covariates and features (namely, parc68 and parc308).

Then we can choose whether to apply **data harmonization** or not. This method is recommended in order to minimize inter-site variability, because cortical thickness measurements are highly scanner specific (see Longitudinal ComBat 20 and ComBat CT 18). The aggregation of data from multiple sites is a nice way to increase the statistical power of studies that aim to detect some biological variability of interest. However, this same protocol may introduce some unwanted technical variability due to the acquisition protocols, the scanner software or hardware. This data harmonization seeks to minimize this effects using an empirical Bayesian method (additive effects + multiplicative effects).

With the DataPreparation() function we can also apply the **match-it algorithm**, used for creating patient and control groups balanced on the included covariates (see MatchIt guide and MatchIt docs). By default, the matching method used is nearest neighbor, the distance measure is a generalized linear model (logistic regression). Also, the variable ratio, indicating the number of control units that should be matched to each treated unit, is 1 by default.

In the following, we apply the above-mentioned function in various ways:

```
df_lC_NO_matched_68 <- DataPreparation(parc = "parc68",
                                         harmonization = "lC",
                                         match = F)

## File parc68 already exists, reading file...
## File with longCombat harmonization already exists, reading file...
## Done!

df_lC_matched_68 <- DataPreparation(parc = "parc68",
                                       harmonization = "lC",
                                       match = T)

## File parc68 already exists, reading file...
## File with longCombat harmonization already exists, reading file...
## File with MATCH-IT already exists, reading file...
## Done!

df_lC_NO_matched_308 <- DataPreparation(parc = "parc308",
                                         harmonization = "lC",
                                         match = F)
```

```

## File parc308 already exists, reading file...
## File with longCombat harmonization already exists, reading file...
## Done!

df_lC_matched_308 <- DataPreparation(parc = "parc308",
                                         harmonization = "lC",
                                         match = T)

## File parc308 already exists, reading file...
## File with longCombat harmonization already exists, reading file...
## File with MATCH-IT already exists, reading file...
## Done!

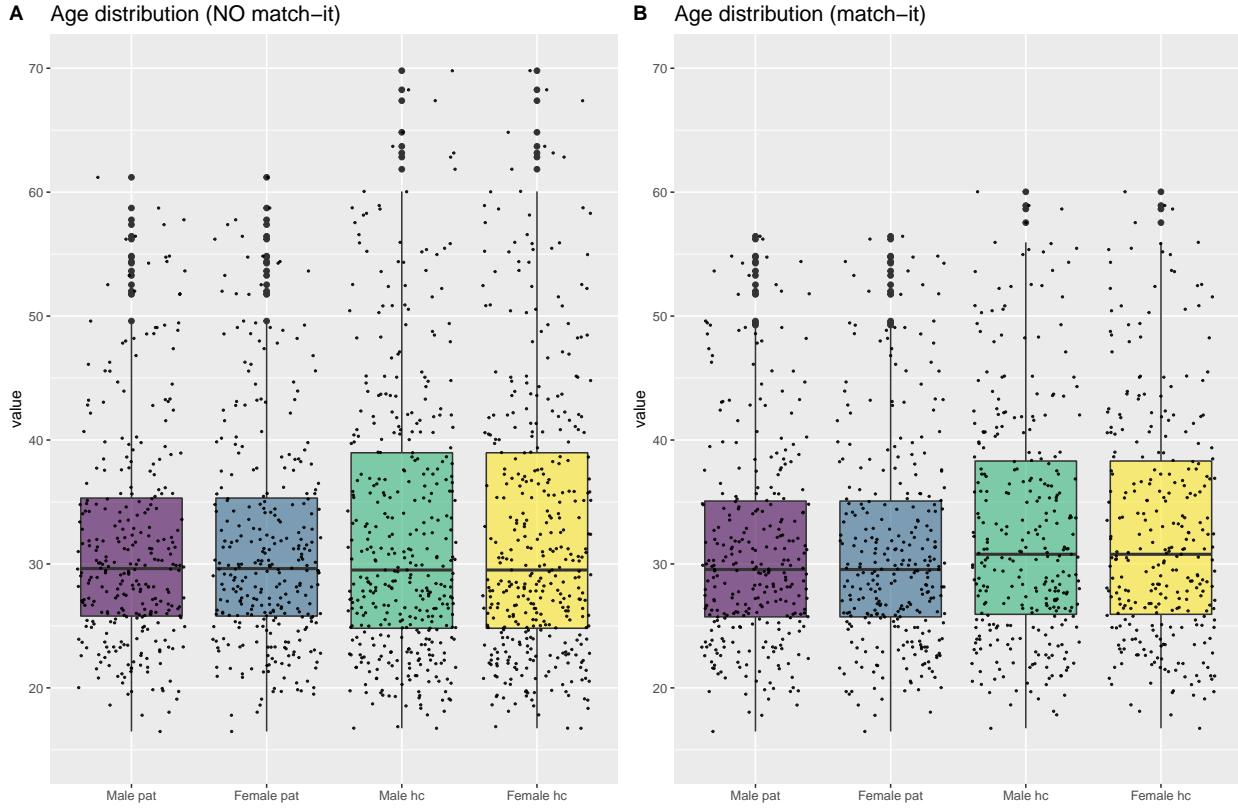
```

Exploratory Data Analysis

The covariates that will be used are:

- age: age of the subject at the moment of the image acquisition. Continuous variable (16.41-69.8).
- scode: biological sex of the subject. Binary variable [0, 1], being 1 for males and 0 for females.
- dcode: diagnostic of the subject. Binary variable [0, 1], being 1 for patients and 0 for controls.
- acode: machine on which the image was taken. Binary variable [0, 1].
- timepoint: number of the visit. Discrete variable [1, 2, 3].
- subID: unique identifier of each subject. Discrete variable [2, ..., 952].
- euler_lh and euler_rh: euler number is a measure that represents the amount of abnormalities encountered in a step of the image preprocessing. There are two values corresponding to the left and right hemisphere of the brain. We computed the mean of this values to obtain a new covariate: euler. It is a negative discrete variable, in this case between (-339, -28).
- ID: unique identifier of each image. Categorical variable.

In the following we explore the effects of the data preparation pipeline applied before, we show relevant changes on **age variable** before and after applying the match-it workflow. To do so, we can use either the 68 regions parcellation or the 308 regions one, results will be the same because we are just looking at the covariates (which are the same for both parcellation maps). First we show a boxplot graph of the age distribution before and after the match-it, differentiating between sex (male-female) and diagnosis (healthy control - patient):



A quick look is taken at the ages than doesn't match between controls and patients (applying the floor function to the age variable, that is, rounding down to the nearest integer) in both datasets:

```
EDA_match_ages(df_1C_NO_matched_68, "NO-match-it")
```

```
## NO-match-it dataset:
## Ages that doesn't match in patients vs controls: 59 60 62 63 64 65 67 68 69
## Min age: 16.41
## Max age: 69.8
```

```
EDA_match_ages(df_1C_matched_68, "Match-it")
```

```
## Match-it dataset:
## Ages that doesn't match in patients vs controls: 46 56 57 58 60
## Min age: 16.48
## Max age: 60.03
```

We show the number of subjects by timepoint and by sex, before and after the match. **The original number of subjects in each category is indicated between parenthesis.**

	timepoint 1	timepoint 2	timepoint 3
# controls	169 (298)	164 (293)	49 (109)
# patients	169 (169)	164 (168)	49 (50)

	sex 0	sex 1
Timepoint 1		
# controls	38 (131)	131 (167)
# patients	38 (38)	131 (131)
Timepoint 2		
# controls	37 (130)	127 (163)
# patients	38 (38)	126 (130)
Timepoint 3		
# controls	8 (50)	41 (59)
# patients	7 (7)	42 (43)

Now we explore the **variance contribution** across measures with the `fitExtractVarPartModel()` function from Bioconductor (see doc). This function fits a linear (mixed) model, taking into account some manually introduced formula, in order to estimate the contribution of multiple sources of variation while simultaneously correcting for all other variables. It reports the fraction of variance attributable to each variable in the model. This will help us to know what covariates can be used and it justifies the model used.

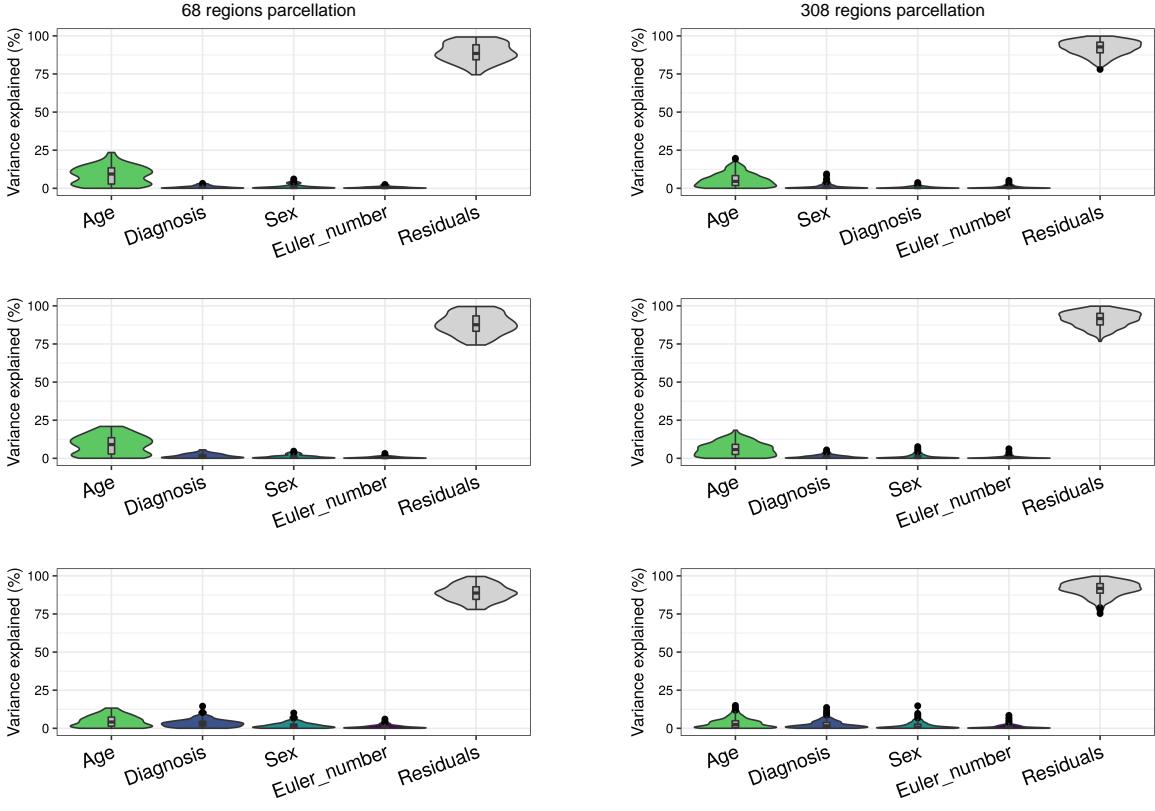
NO Match-it dataframe

At first, we will deal with the NO matched dataset.

The following violin plots show the variance partition based on a general linear model by timepoint (formula $\text{<- } \sim \text{Age} + \text{Sex} + \text{Diagnosis} + \text{Euler_number}$).

```
variancePartition_lm(df = df_1C_NO_matched_68,
                     measure = "CT_freesurfer",
                     lab = "1C_NO_matched_parc68_lm",
                     par = "parc68")

variancePartition_lm(df = df_1C_NO_matched_308,
                     measure = "thickness",
                     lab = "1C_NO_matched_parc308_lm",
                     par = "parc308")
```



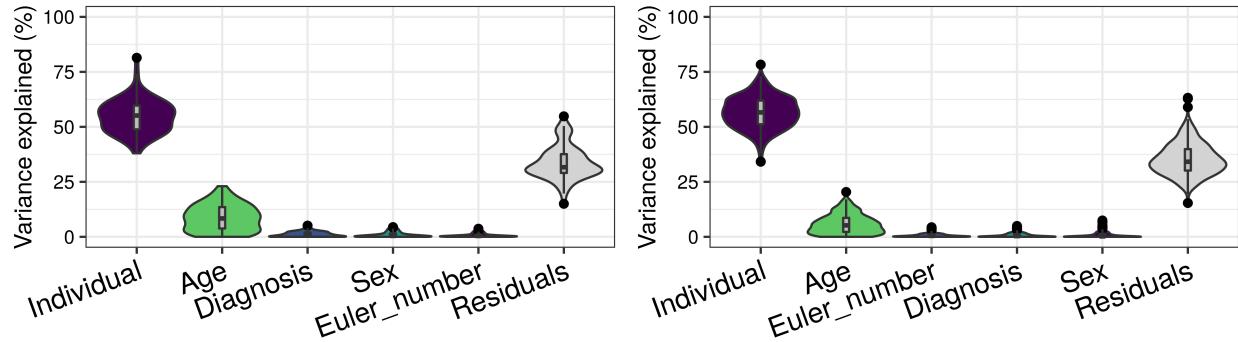
But the `fitExtractVarPartModel()` function can also cope with mixed effects models. This next representation is based on a linear mixed effects model, without explicitly differentiating by timepoint (`formula <- ~ Age + (1|Sex) + (1|Diagnosis) + Euler_number + (1|Individual)`), but implicitly incorporated in the subID covariate.

```
variancePartition_lme(df = df_1C_NO_matched_68,
                      measure = "CT_freesurfer",
                      lab = "1C_NO_matched_parc68_lme",
                      par = "parc68")

variancePartition_lme(df = df_1C_NO_matched_308,
                      measure = "thickness",
                      lab = "1C_NO_matched_parc308_lme",
                      par = "parc308")
```

68 regions parcellation

308 regions parcellation



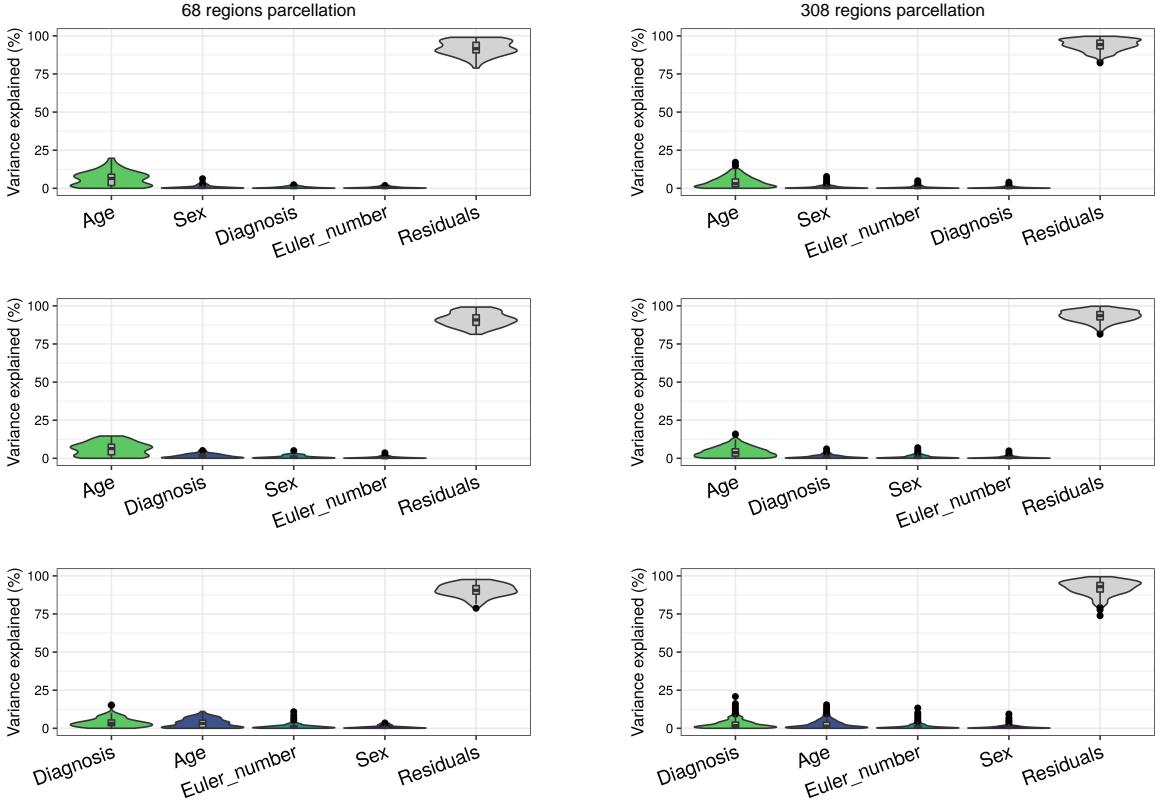
Match-it dataframe

Now, we will apply the same algorithms to the matched dataset.

The following violin plots show the variance partition based on a general linear model by timepoint (formula $\text{<- } \sim \text{Age} + \text{Sex} + \text{Diagnosis} + \text{Euler_number}$).

```
variancePartition_lm(df = df_1C_matched_68,
                     measure = "CT_freesurfer",
                     lab = "1C_matched_parc68_lm",
                     par = "parc68")

variancePartition_lm(df = df_1C_matched_308,
                     measure = "thickness",
                     lab = "1C_matched_parc308_lm",
                     par = "parc308")
```



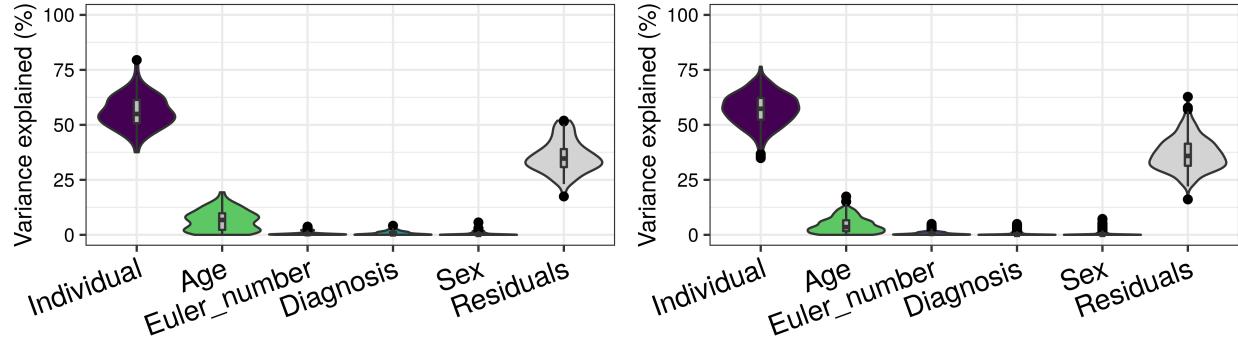
But the `fitExtractVarPartModel()` function can also cope with mixed effects models. This next representation is based on a linear mixed effects model, without explicitly differentiating by timepoint (`formula <- ~ Age + (1|Sex) + (1|Diagnosis) + Euler_number + (1|Individual)`), but implicitly incorporated in the subID covariate.

```
variancePartition_lme(df = df_1C_matched_68,
                      measure = "CT_freesurfer",
                      lab = "1C_matched_parc68_lme",
                      par = "parc68")

variancePartition_lme(df = df_1C_matched_308,
                      measure = "thickness",
                      lab = "1C_matched_parc308_lme",
                      par = "parc308")
```

68 regions parcellation

308 regions parcellation



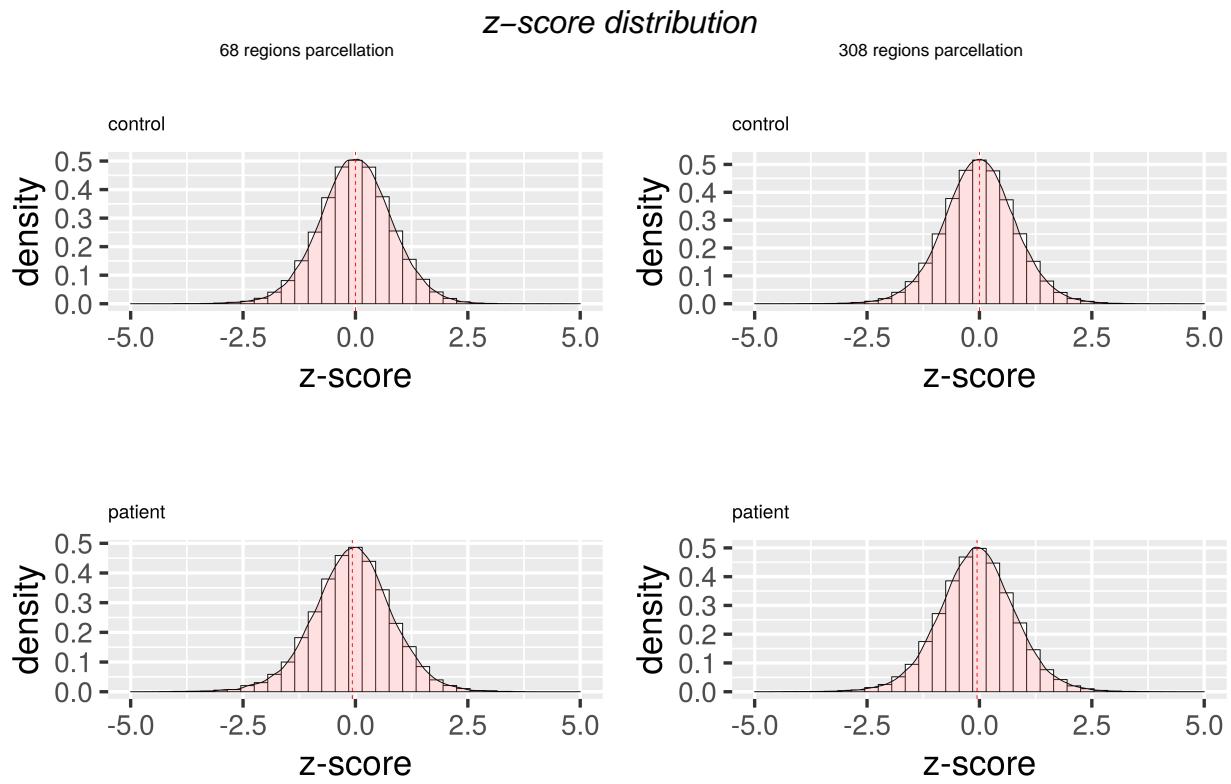
Linear Mixed Effects Model Regression

Calling the `run_NormativeModel` function with different datasets. We build the regression with the data from healthy controls and make individual predictions over controls as well as over patients. The formula used is `region ~ scode + age + euler` as the fixed effects and `~ 1 + age|subID` as the random ones. This function will return the z-scores (one for each region for each timepoint of each subject).

NO Match-it dataframe

```
Zs_NOmatch_68 <- run_NormativeModel(df_1C_NO_matched_68,
                                       measure = "CT_freesurfer",
                                       parc = "parc68",
                                       match = "NOmatch",
                                       harmonization = "1C")

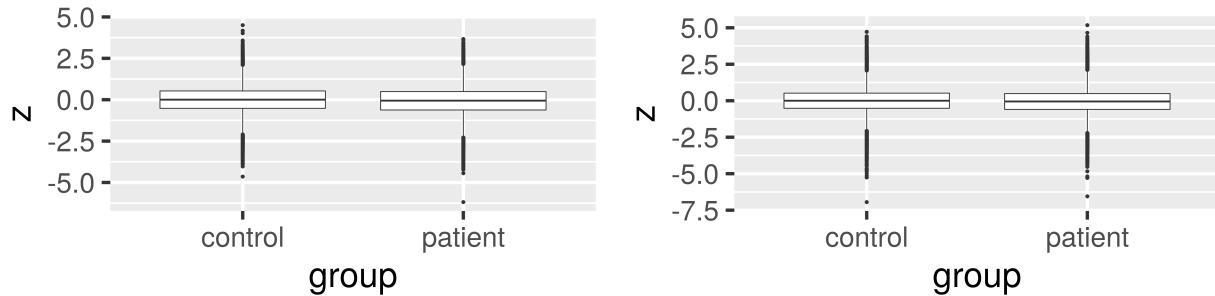
Zs_NOmatch_308 <- run_NormativeModel(df_1C_NO_matched_308,
                                       measure = "thickness",
                                       parc = "parc308",
                                       match = "NOmatch",
                                       harmonization = "1C")
```



```
Z_across_regions_plot(Z = Zs_N0match_308,  
                      parc308",  
                      lab="1C_N0matched_parc308")  
  
Z_across_regions_plot(Z = Zs_N0match_68,  
                      parc68",  
                      lab="1C_N0matched_parc68")
```

z-score distribution

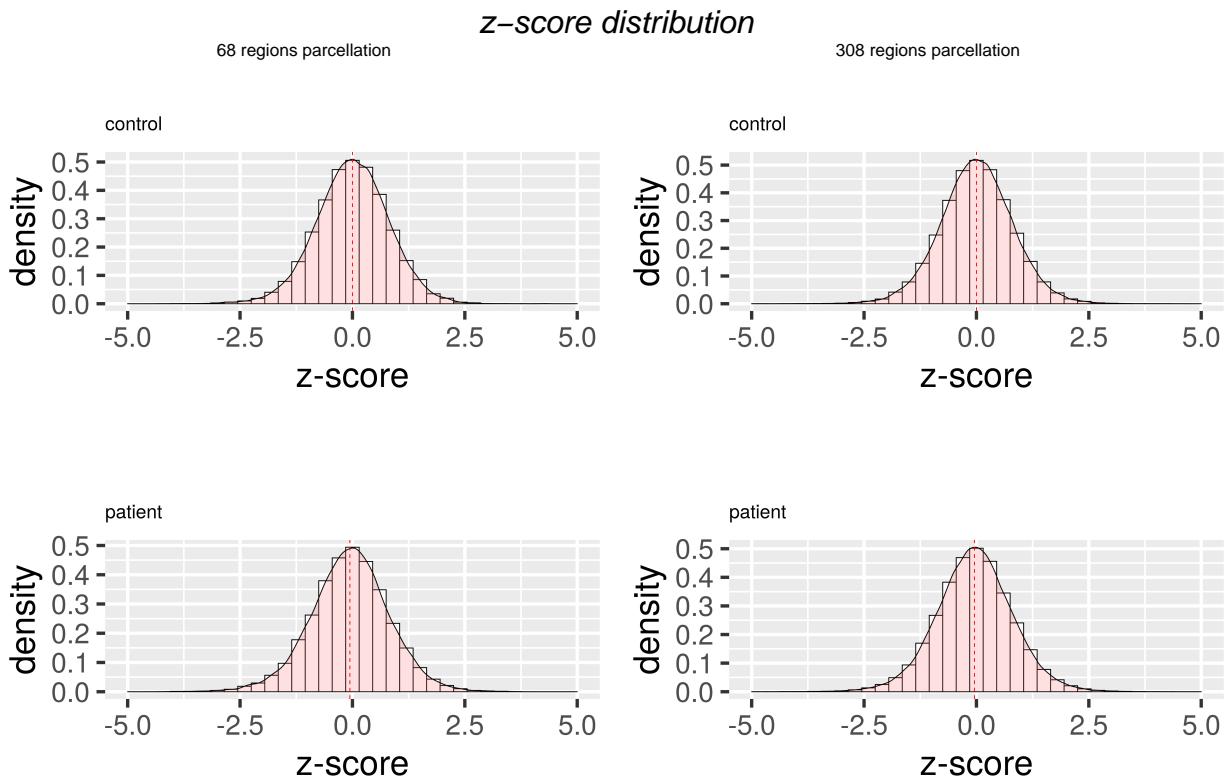
68 regions parcellation 308 regions parcellation



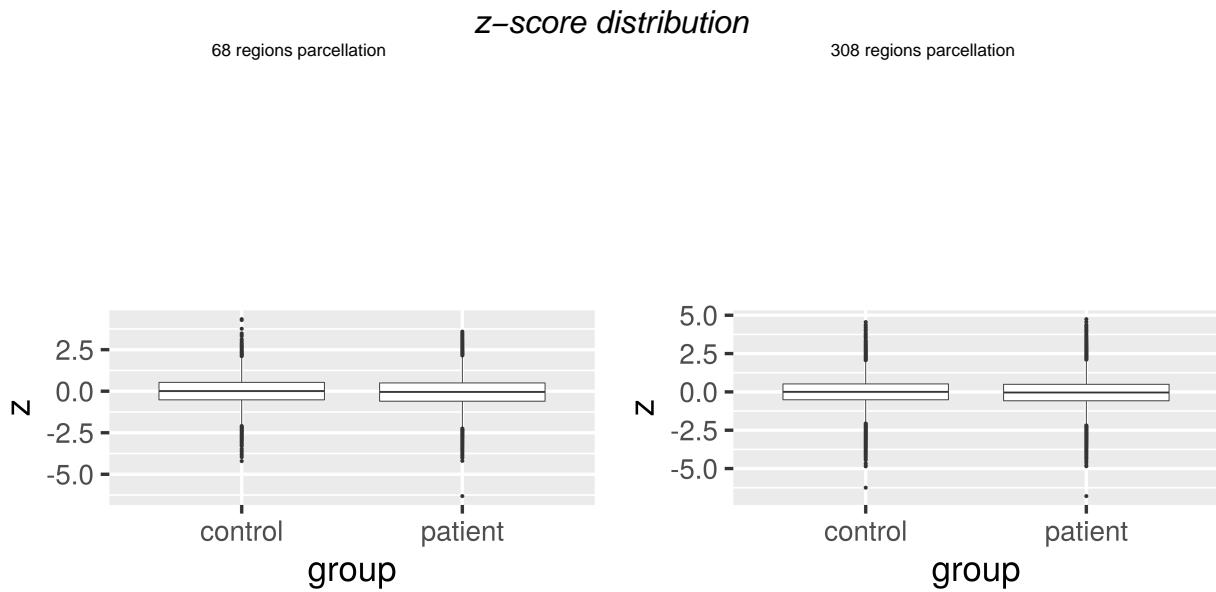
Match-it dataframe

```
Zs_match_68 <- run_NormativeModel(df_lC_matched_68,
                                     measure = "CT_freesurfer",
                                     parc = "parc68",
                                     match = "match",
                                     harmonization = "lC")

Zs_match_308 <- run_NormativeModel(df_lC_matched_308,
                                     measure = "thickness",
                                     parc = "parc308",
                                     match = "match",
                                     harmonization = "lC")
```



```
Z_across_regions_plot(Z = Zs_match_308,  
                      par="parc308",  
                      lab="1C_matched_parc308")  
Z_across_regions_plot(Z = Zs_match_68,  
                      par="parc68",  
                      lab="1C_matched_parc68")
```



Cohen's d

Cohen's d, also known as the standardized mean difference (SMD), is a type of **effect size** between two means. An effect size is a quantitative measure of the magnitude for the difference between two means. Since the values are standardized, it is possible to compare values between different variables. Cohen's d is the number of standard deviations between two groups (for more info see doc).

NO match-it dataframe

```
ds_N0match_68 <- run_NormativeModel_Cohensd(df_1C_N0_matched_68,
                                              measure = "CT_freesurfer",
                                              parc = "parc68",
                                              match = "N0match",
                                              harmonization = "1C")
```

small	very small
1	67

Region lh_lateralorbitofrontal_CT_freesurfer has an small effect size.

```
ds_NOmatch_308 <- run_NormativeModel_Cohensd(df_1C_NO_matched_308,
                                              measure = "thickness",
                                              parc = "parc308",
                                              match = "NOmatch",
                                              harmonization = "1C")
```

very small
308

Match-it dataframe

```
ds_match_68 <- run_NormativeModel_Cohensd(df_1C_matched_68,
                                             measure = "CT_freesurfer",
                                             parc = "parc68",
                                             match = "match",
                                             harmonization = "1C")
```

small very small
1 67

Region lh_lateralorbitofrontal_CT_freesurfer has an small effect size.

```
ds_match_308 <- run_NormativeModel_Cohensd(df_1C_matched_308,
                                              measure = "thickness",
                                              parc = "parc308",
                                              match = "match",
                                              harmonization = "1C")
```

small very small
1 307

Region lh_lateralorbitofrontal_part3_thickness has an small effect size.

Match-it dataframe (Age*Diagnosis) BEFORE EXCLUDING DEVIANTS

Calling the *run_AgeDiagnosisModel* function. In this case, we build the regression with the data from healthy controls as well as patients. Next we exclude the deviants from the model and run again the function

This function will return the p-values (one for each region of each subject)

```
p_val_68 <- run_AgeDiagnosisModel(df_1C_matched_68,
                                      measure = "CT_freesurfer",
                                      Z = NULL,
                                      exclude_deviants = F)

p_val_N068 <- run_AgeDiagnosisModel(df_1C_NO_matched_68,
```

```

    measure = "CT_freesurfer",
    Z = NULL,
    exclude_deviants = F)

p_val_308 <- run_AgeDiagnosisModel(df_1C_matched_308,
                                      measure = "thickness",
                                      Z = NULL,
                                      exclude_deviants = F)

p_val_N0308 <- run_AgeDiagnosisModel(df_1C_N0_matched_308,
                                       measure = "thickness",
                                       Z = NULL,
                                       exclude_deviants = F)

p_val_FDR_68     <- Apply_FDR_Correction(p_val_68)
p_val_FDR_N068   <- Apply_FDR_Correction(p_val_N068)
p_val_FDR_308    <- Apply_FDR_Correction(p_val_308)
p_val_FDR_N0308  <- Apply_FDR_Correction(p_val_N0308)

```

Next we show the number of statistically significant regions before and after FDR correction:

- 68 REGIONS PARCELLATION:

WITHOUT MATCH-IT

Variable dcode: 39 / 68 regions --> 37 / 68 regions
 Variable dcode*age: 15 / 68 regions --> 0 / 68 regions

WITH MATCH-IT

Variable dcode: 28 / 68 regions --> 20 / 68 regions
 Variable dcode*age: 14 / 68 regions --> 3 / 68 regions

- 308 REGIONS PARCELLATION:

WITHOUT MATCH-IT

Variable dcode: 142 / 308 regions --> 112 / 308 regions
 Variable dcode*age: 32 / 308 regions --> 0 / 308 regions

WITH MATCH-IT

Variable dcode: 94 / 308 regions --> 42 / 308 regions
 Variable dcode*age: 41 / 308 regions --> 0 / 308 regions

Match-it dataframe (Age*Diagnosis) AFTER EXCLUDING DEVIANTS

```

p_val_68 <- run_AgeDiagnosisModel(df_1C_matched_68,
                                      measure = "CT_freesurfer",
                                      Z = Zs_match_68,
                                      exclude_deviants = T)

p_val_N068 <- run_AgeDiagnosisModel(df_1C_N0_matched_68,
                                       measure = "CT_freesurfer",
                                       Z = Zs_N0match_68,
                                       exclude_deviants = T)

```

```

p_val_308 <- run_AgeDiagnosisModel(df_1C_matched_308,
                                      measure = "thickness",
                                      Z = Zs_match_308,
                                      exclude_deviants = T)

p_val_N0308 <- run_AgeDiagnosisModel(df_1C_N0_matched_308,
                                         measure = "thickness",
                                         Z = Zs_N0match_308,
                                         exclude_deviants = T)

p_val_FDR_68     <- Apply_FDR_Correction(p_val_68)
p_val_FDR_N068   <- Apply_FDR_Correction(p_val_N068)
p_val_FDR_308    <- Apply_FDR_Correction(p_val_308)
p_val_FDR_N0308  <- Apply_FDR_Correction(p_val_N0308)

```

```

## - 68 REGIONS PARCELLATION:
## WITHOUT MATCH-IT
## Variable dcode: 39 / 68 regions --> 35 / 68 regions
## Variable dcode*age: 15 / 68 regions --> 0 / 68 regions
##
## WITH MATCH-IT
## Variable dcode: 28 / 68 regions --> 22 / 68 regions
## Variable dcode*age: 17 / 68 regions --> 3 / 68 regions
##
## - 308 REGIONS PARCELLATION:
## WITHOUT MATCH-IT
## Variable dcode: 143 / 308 regions --> 106 / 308 regions
## Variable dcode*age: 40 / 308 regions --> 0 / 308 regions
##
## WITH MATCH-IT
## Variable dcode: 95 / 308 regions --> 43 / 308 regions
## Variable dcode*age: 43 / 308 regions --> 0 / 308 regions

```

Analysis of z-scores and computation of global scores

The following tables and results represent, **in terms of samples**, the number of regions deviated ($Z < -1.96$ and $Z > 1.96$).

Again, the number of samples refers to the total number of regions, knowing that each subject can have a total of 68 or 308 regions, calculated for each timepoint ($timepoint = 1, 2, 3$) and differentiated between controls and subjects.

68 REGIONS

NO MATCHED	timepoint 1	timepoint 2	timepoint 3
# total samples	31.688	31.348	10.744
# samples dev (%)	795 (2.51%)	696 (2.22%)	275 (2.56%)
# controls samples	20.264	19.924	7.412
# controls dev (%)	450 (2.22%)	335 (1.68%)	143 (1.93%)
# patients samples	11.424	11.424	3.332
# patients dev (%)	345 (3.02%)	361 (3.16%)	132 (3.96%)

308 REGIONS

NO MATCHED	timepoint 1	timepoint 2	timepoint 3
# total samples	143.528	141.988	48.664
# samples dev (%)	3.421 (2.38%)	3.136 (2.21%)	1.122 (2.31%)
# controls samples	91.784	90.244	33.572
# controls dev (%)	2.042 (2.22%)	1.696 (1.88%)	680 (2.03%)
# patients samples	51.744	51.744	15.092
# patients dev (%)	1.379 (2.67%)	1.440 (2.78%)	442 (2.93%)

68 REGIONS

MATCHED	timepoint 1	timepoint 2	timepoint 3
# total samples	22.916	22.304	6.664
# samples dev (%)	617 (2.69%)	519 (2.33%)	187 (2.81%)
# controls samples	11.492	11.152	3.332
# controls dev (%)	269 (2.34%)	176 (1.58%)	71 (2.13%)
# patients samples	11.424	11.152	3.332
# patients dev (%)	348 (3.05%)	343 (3.08%)	116 (3.48%)

308 REGIONS

MATCHED	timepoint 1	timepoint 2	timepoint 3
# total samples	103.796	101.024	30.184
# samples dev (%)	2.518 (2.42%)	2.318 (2.27%)	760 (2.33%)
# controls samples	52.052	50.512	15.092
# controls dev (%)	1.111 (2.13%)	910 (1.8%)	288 (1.91%)
# patients samples	51.744	50.512	15.092
# patients dev (%)	1.363 (2.63%)	1.389 (2.75%)	440 (2.92%)

In the following, we chose to work with the matched dataset and the Zs derived from its lme model, providing a better statistical support for the analysis.

Taken into account the type of the deviation (infra-normal deviation $Z < -1.96$, supra-normal deviation $Z > 1.96$, or deviant $|Z| > 1.96$):

68 REGIONS

	Infra-normal deviants	Non deviants	Supra-normal deviants	Deviants
tp 1	339 (1.48%)	22.299 (97.31%)	278 (1.21%)	617 (2.69%)
controls	172 (50.74%)	11.223 (50.33%)	97 (34.89%)	269 (43.60%)
patients	167 (49.26%)	11.076 (49.67%)	181 (65.11%)	348 (56.40%)
tp 2	322 (1.44%)	21.785 (97.67%)	197 (0.88%)	519 (2.33%)
controls	83 (25.78%)	10.976 (50.38%)	93 (47.21%)	176 (33.91%)
patients	239 (74.22%)	10.809 (49.62%)	104 (52.79%)	343 (66.09%)
tp 3	133 (2.00%)	6.477 (97.19%)	54 (0.81%)	187 (2.81%)
controls	39 (29.32%)	3.261 (50.35%)	32 (59.26%)	71 (37.97%)

	Infra-normal deviants	Non deviants	Supra-normal deviants	Deviants
patients	94 (70.68%)	3.216 (49.65%)	22 (40.74%)	116 (62.03%)
total	794 (1.53%)	50.561 (97.45%)	529 (1.02%)	1.323 (2.55%)

308 REGIONS

	Infra-normal deviants	Non deviants	Supra-normal deviants	Deviants
tp 1	1.261 (1.21%)	101.321 (97.62%)	1.214 (1.17%)	2.475 (2.38%)
controls	631 (50.04%)	50.941 (50.28%)	480 (39.54%)	1.111 (44.89%)
patients	630 (49.96%)	50.380 (49.72%)	734 (60.46%)	1.364 (55.11%)
tp 2	1.294 (1.28%)	98.725 (97.72%)	1.005 (0.99%)	2.299 (2.28%)
controls	444 (34.31%)	49.602 (50.24%)	466 (46.37%)	910 (39.58%)
patients	850 (65.69%)	49.123 (49.76%)	539 (53.63%)	1.389 (60.42%)
tp 3	440 (1.46%)	29.456 (97.59%)	288 (0.95%)	728 (2.41%)
controls	124 (28.18%)	14.804 (50.26%)	164 (56.94%)	288 (39.56%)
patients	316 (71.82%)	14.652 (49.74%)	124 (43.06%)	440 (60.44%)
total	2.995 (1.27%)	22.9502 (97.66%)	2.507 (1.07%)	5.502 (2.34%)

We computed the number of subjects that are **deviant at timepoint 1** and stay **deviant in subsequent timepoints**, for each region.

68 REGIONS

A maximum of 5 subjects (1.48% of total subjects) fulfill that condition for any region.

	Condition
rh_cuneus_CT_freesurfer	5 (1.48%)
lh_rostralmiddlefrontal_CT_freesurfer	4 (1.19%)
lh_temporalpole_CT_freesurfer	4 (1.19%)

308 REGIONS

A maximum of 6 subjects (1.78% of total subjects) fulfill that condition for any region.

	Condition
lh_cuneus_part2_thickness	6 (1.78%)
lh_precentral_part5_thickness	6 (1.78%)
lh_lateraloccipital_part5_thickness	5 (1.48%)

We computed the number of subjects that are **not deviant at timepoint 1** but become **deviant in subsequent timepoints**, for each region. A maximum of 20 subjects (5.93% of total subjects) fulfill that condition for any region.

	Condition
lh_lateraloccipital_CT_freesurfer	20 (5.93%)
rh_temporalpole_CT_freesurfer	20 (5.93%)
lh_posteriorcingulate_CT_freesurfer	18 (5.34%)

1. Percentage of patients deviated from the normative range for any single cortical region

68 REGIONS

- Timepoint 1: No more than 6.548% of patients deviated from the normative range for any single cortical region.
- Timepoint 2: No more than 6.707% of patients deviated from the normative range for any single cortical region.
- Timepoint 3: No more than 10.204% of patients deviated from the normative range for any single cortical region.

308 REGIONS

- Timepoint 1: No more than 9.52% of patients deviated from the normative range for any single cortical region.
 - Timepoint 2: No more than 9.76% of patients deviated from the normative range for any single cortical region.
 - Timepoint 3: No more than 10.2% of patients deviated from the normative range for any single cortical region.
-

2. Most common regions with infra-normal deviations. Percentage of patients.

68 REGIONS

- Timepoint 1: Infra-normal deviations in CT of subjects were most commonly located in **rh_fusiform_CT_freesurfer** cortices, although only 1.786% of patients showed significant deviations in these regions.
- Timepoint 2: Infra-normal deviations in CT of subjects were most commonly located in **rh_temporalpole_CT_freesurfer** cortices, although only 4.268% of patients showed significant deviations in these regions.
- Timepoint 3: Infra-normal deviations in CT of subjects were most commonly located in **lh_fusiform_CT_freesurfer** cortices, although only 8.163% of patients showed significant deviations in these regions.

308 REGIONS

- Timepoint 1: Infra-normal deviations in CT of subjects were most commonly located in **lh_superiortemporal_part2_thickness** cortices, although only 4.17% of patients showed significant deviations in these regions.

- Timepoint 2: Infra-normal deviations in CT of subjects were most commonly located in **lh_superiorfrontal_part9_thickness** cortices, although only 6.1% of patients showed significant deviations in these regions.
 - Timepoint 3: Infra-normal deviations in CT of subjects were most commonly located in **lh_insula_part1_thickness** cortices, although only 8.16% of patients showed significant deviations in these regions.
-

3. Most common regions with supra-normal deviations. Percentage of individuals.

68 REGIONS

- Timepoint 1: Supra-normal deviations in CT were most common in the **lh_lateraloccipital_CT_freesurfer** regions, 2.967% of individuals.
- Timepoint 2: Supra-normal deviations in CT were most common in the **lh_lateraloccipital_CT_freesurfer** regions, 2.134% of individuals.
- Timepoint 3: Supra-normal deviations in CT were most common in the **lh_parahippocampal_CT_freesurfer** regions, 3.061% of individuals.

308 REGIONS

- Timepoint 1: Supra-normal deviations in CT were most common in the **lh_superiorfrontal_part7_thickness** regions, 3.26% of individuals.
 - Timepoint 2: Supra-normal deviations in CT were most common in the **rh_medialorbitofrontal_part1_thickness** regions, 4.88% of individuals.
 - Timepoint 3: Supra-normal deviations in CT were most common in the **lh_inferiorparietal_part6_thickness** regions, 6.12% of individuals.
-

4. Percentage of subjects with at least one region infra-normal deviated. Patients vs Healthy controls.

68 REGIONS

- Timepoint 1: Infra-normal deviations for at least one region were evident in 29.167% of patients, whereas this was the case for 26.036% of healthy individuals.
- Timepoint 2: Infra-normal deviations for at least one region were evident in 35.366% of patients, whereas this was the case for 27.439% of healthy individuals.
- Timepoint 3: Infra-normal deviations for at least one region were evident in 40.816% of patients, whereas this was the case for only 38.776% of healthy individuals.

308 REGIONS

- Timepoint 1: Infra-normal deviations for at least one region were evident in 60.71% of patients, whereas this was the case for 60.36% of healthy individuals.

- Timepoint 2: Infra-normal deviations for at least one region were evident in 63.41% of patients, whereas this was the case for 62.2% of healthy individuals.
 - Timepoint 3: Infra-normal deviations for at least one region were evident in 77.55% of patients, whereas this was the case for only 73.47% of healthy individuals.
-

5. Percentage of subjects with at least one region supra-normal deviated. Patients vs Healthy controls.

68 REGIONS

- Timepoint 1: Supra-normal deviations for at least one region were evident in 32.143% of patients and 34.32% of healthy individuals.
- Timepoint 2: Supra-normal deviations for at least one region were evident in 28.659% of patients and 29.878% of healthy individuals.
- Timepoint 3: Supra-normal deviations for at least one region were evident in 30.612% of patients and 36.735% of healthy individuals.

308 REGIONS

- Timepoint 1: Supra-normal deviations for at least one region were evident in 76.19% of patients and 71.6% of healthy individuals.
 - Timepoint 2: Supra-normal deviations for at least one region were evident in 67.68% of patients and 69.51% of healthy individuals.
 - Timepoint 3: Supra-normal deviations for at least one region were evident in 67.35% of patients and 81.63% of healthy individuals.
-

Percentage of deviant subjects for number of regions

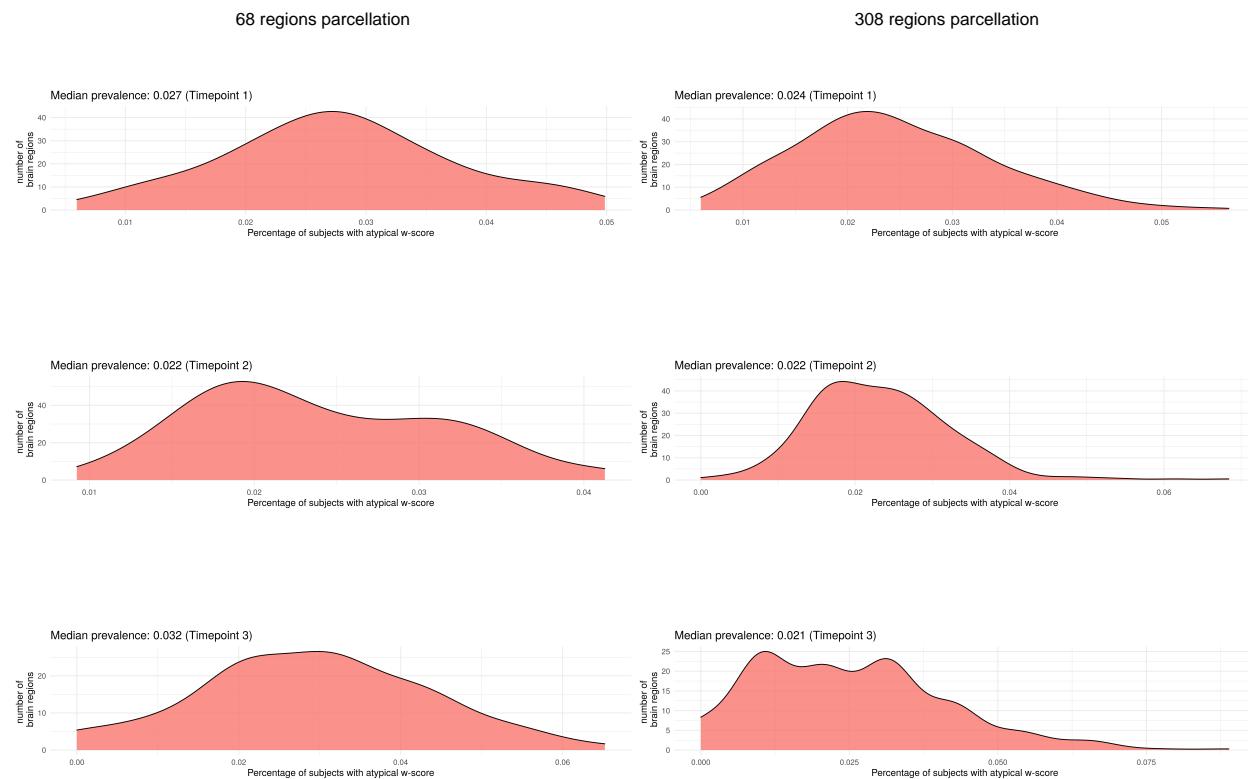
Figura B del paper de Bethlehem. En el eje de las x se representa el porcentaje de sujetos y en el eje y el número de regiones con el mismo ratio $\frac{|Z|>1.96}{|Z|<1.96}$. Se calcula para cada timepoint (*timepoint = 1, 2, 3*).

```
getStatistics(Zs= Zs_match_68, lab= "1C_match_parc68", parc= "parc68")
getStatistics(Zs= Zs_N0match_68, lab= "1C_N0match_parc68", parc= "parc68")

getStatistics(Zs= Zs_match_308, lab= "1C_match_parc308", parc= "parc308")
getStatistics(Zs= Zs_N0match_308, lab= "1C_N0match_parc308", parc= "parc308")
```

Show the results from the global ratios obtained:

Match-it dataframe (longCombat):



NO Match-it dataframe (longCombat):

