

Analyzing time series data using `permu.test`

Cesko C. Voeten

14 September 2021

This vignette models ERP data, an example of *time series data*, using permutation testing. The `permutest` package implements two approaches, one based on package `lmPerm`, which is shown here, and the other based on package `buildmer`, which is shown in the other vignette. The `lmPerm` approach can do multivariate responses, but cannot run generalized linear models and only has very limited support for random effects (i.e.: the `Error()` functionality works to run ANOVA with random effects, but only using Type II SS). The `buildmer` approach can fit any model that function `buildmer` from package `buildmer` can (i.e.: `lm`, `glm`, `lmer`, and `glmer` models), but cannot fit multiple responses. Also, the `buildmer` approach can use the cluster-mass test by Maris & Oostenveld (2007), using the relevant machinery in package `permuco`.

Sample data

The dataset we will be working with is called `MMN`, which is an excerpt from a passive-oddball task conducted by Jager (in prep.) A proper method section with more details should eventually be found in that paper, but a brief and incomplete summary follows here in order to make it easier to work with the data. Jager (in prep.) presented Dutch learners of English with a stream of isolated vowels from English or from Dutch created by means of formant synthesis. Each of her six conditions (summarized in the below table; the package supplies data from two of these conditions) each having used four vowels: three of the same phonetic category (e.g. three realizations of the DRESS vowel with slightly different formant frequencies) and one of a different phonetic category (e.g. the TRAP vowel). The first set of vowels, termed ‘standards’, were presented frequently; the fourth vowel, the ‘deviant’, infrequently interrupted the stream of standards. Participants were instructed not to pay attention to this stream of sounds, and watched a silent movie to keep them occupied. EEG recordings of 30 electrodes (T7 and T8 were excluded) were recorded while the participants were exposed to the vowel stimuli. At the presentation of one of the deviant vowels, we expect to observe a negative deflection in the EEG signal about 200 milliseconds after stimulus onset, originating from frontal and central electrode sites. This effect is called the ‘mismatch negativity’. A second effect called the ‘P300’ may also occur, but is ignored here. The data supplied with `permutest` are a subset of the vowel pairs tested by Jager (in prep.) consisting of the English vowel in DRESS presented as a standard vs. as a deviant, in both cases against the vowel from the Dutch word ZET.

The first few rows of the data look as follows:

```
library(permutest)
head(MMN)

##           Fp1      AF3      F7      F3      FC1      FC5      C3      CP1      CP5      P7
## 1 -0.0002 -1.7087  0.1357  0.3822  0.1538  1.1556  0.7552  0.7864 -0.0222  0.3399
## 2  0.2871 -2.2155  0.0740  0.3518  0.1051  1.4625  0.6329  0.7107 -0.0411  0.3134
## 3  0.4573 -2.4930  0.0070  0.2775  0.0810  1.6963  0.5497  0.6228 -0.0651  0.2984
## 4  0.5319 -2.4990 -0.0517  0.1462  0.0860  1.8356  0.5089  0.5332 -0.0914  0.2901
## 5  0.5321 -2.2408 -0.0890 -0.0346  0.1177  1.9058  0.4975  0.4489 -0.1169  0.2790
## 6  0.4875 -1.7814 -0.0951 -0.2321  0.1620  1.9712  0.4907  0.3710 -0.1400  0.2545
##           P3      Pz      P03      O1      Oz      O2      P04      P4      P8      CP6
## 1  0.4392  0.6500 -0.1648 -0.5426 -0.5382 -0.9072 -0.0193  0.7312  0.0102  1.0790
## 2  0.4209  0.5860 -0.1576 -0.5430 -0.5308 -0.8392  0.0938  0.7830  0.0822  1.1723
```

```

## 3 0.3585 0.5030 -0.1631 -0.5533 -0.5394 -0.7937 0.1809 0.7927 0.1162 1.1724
## 4 0.2612 0.4114 -0.1817 -0.5750 -0.5648 -0.7628 0.2506 0.7774 0.1246 1.1051
## 5 0.1417 0.3183 -0.2126 -0.6071 -0.6043 -0.7330 0.3125 0.7553 0.1226 1.0042
## 6 0.0138 0.2272 -0.2520 -0.6417 -0.6443 -0.6848 0.3767 0.7424 0.1276 0.9066
##      CP2      C4      FC6      FC2      F4      F8      AF4      Fp2      Fz      Cz
## 1 0.8142 1.3876 0.4432 0.7724 -0.1918 0.5416 0.0526 -0.1283 0.1105 0.9059
## 2 0.7216 1.4579 0.6214 0.7768 -0.1749 0.4926 -0.1035 -0.6480 0.0827 0.7886
## 3 0.5969 1.4392 0.7589 0.7753 -0.1024 0.3913 -0.2377 -1.2004 0.0861 0.6604
## 4 0.4643 1.3590 0.8526 0.7678 0.0054 0.2513 -0.3246 -1.7130 0.1086 0.5332
## 5 0.3446 1.2564 0.9184 0.7586 0.1225 0.1030 -0.3462 -2.1017 0.1352 0.4168
## 6 0.2484 1.1672 0.9760 0.7520 0.2162 -0.0195 -0.2996 -2.2827 0.1481 0.3172
##      Time Subject Session Deviant
## 1 1.171875      1      0      1
## 2 3.125000      1      0      1
## 3 5.078125      1      0      1
## 4 7.031250      1      0      1
## 5 8.984375      1      0      1
## 6 10.937500     1      0      1

```

```
nrow(MMN) #how many observations?
```

```
## [1] 22407
```

```
length(unique(MMN$Time)) #how many timepoints?
```

```
## [1] 231
```

The first 30 columns are the 30 sampled EEG electrodes. The **Time** column denotes the moment from stimulus onset, in milliseconds, of each measurement type. **Subject** is the participant, and **Session** is the session of the experiment minus one, to make it a proper dummy indicating whether the datapoint is from the first (0) or the second session (1). Finally, **Deviant** is a dummy indicating whether the stimulus was a standard (0) or a deviant (1), explained in the next paragraph. Note that, contrary to the results and recommendations in Brysbaert (2007), Jager (in prep.) averaged over all her items belonging to the same condition in this experiment.

Time series data such as these are special because the data consist of multiple, *strongly correlated*, measurements of the same signal. This generates two problems. The first is a research problem: which portion of this time series do we want to analyze? The **permutes** package was designed to help researchers answer precisely this question. The second problem is of a statistical nature: how do we handle the strong autocorrelation present throughout this sampled window? Note that in the case of ERP data, these same two problems are additionally encountered in the *spatial* domain: what combination of electrodes ('region of interest') do we want to analyze, and how do we deal with the spatial correlation present in data measured from neighboring sites? This issue is dealt with in the **permutes** package by running separate analyses for each site, as is shown below.

Determining the window and ROI

The first problem equates to what is known in the ERP literature as determining the *window*. The normal way to do this is to run a MANOVA (with the 30 electrodes as the dependent variables) on every individual timepoint, and take as the window the point where a large sequence of significant *p*-values occurs. If we plot time horizontally and the electrode site vertically, we can use this approach to select both a time window and an ROI at the same time.

It should be noted that this approach suffers from an extreme multiple-comparisons problem: we will be running 30×231 ANOVAs! When compared to an asymptotic null distribution, the *p*-values will be spuriously significant in, expectedly, 347 random combinations. The permutation testing approach to time series data helps with this problem in two ways. Firstly, and most importantly, the results from a permutation analysis

are *never* interpreted as actual findings; rather, they only serve as a guideline to empirically determine the analysis window and ROI. Secondly, the null distribution is not taken as the asymptotic F distribution, but is rather inferred from the data itself by *permutation testing*: the null distribution is obtained by randomly permuting the entries of Y and assessing how badly this perturbs the model fit; if a permutation incurs a large deterioration of the fit, then the portions of the design matrix associated with that permutation apparently contributed rather significantly to obtaining the proper fit.

The below code runs a permutation test series on the MMN data and plots the results as a heatmap. Note that permutation testing is not deterministic (the permutations are chosen randomly), so your results may be slightly different from mine.

```
perms <- permu.test(cbind(Fp1,AF3,F7,F3,FC1,FC5,C3,CP1,CP5,P7,P3,Pz,P03,O1,Oz,O2,P04,
  P4,P8,CP6,CP2,C4,FC6,FC2,F4,F8,AF4,Fp2,Fz,Cz) ~ Deviant * Session | Time,
  data=MMN)
## (output not shown)
```

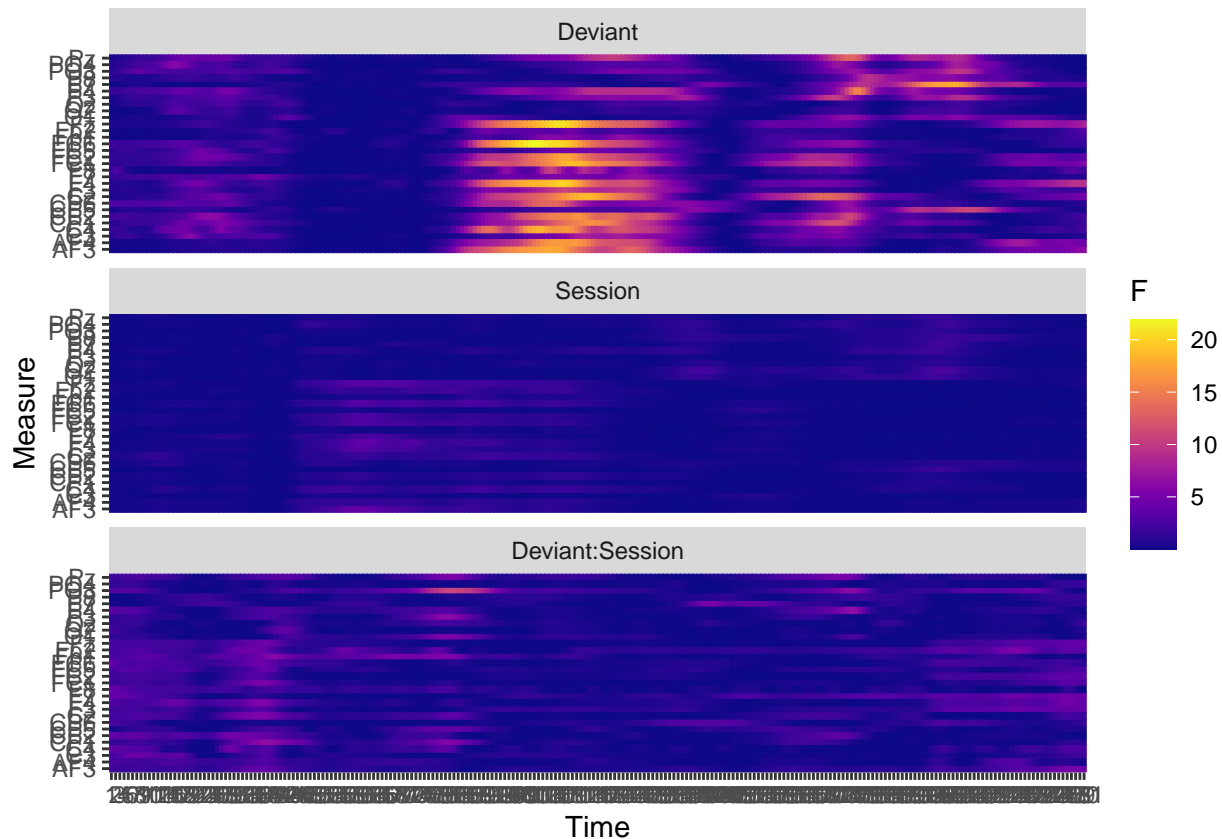
This takes a few seconds to run. We can speed it up by parallelizing the testing of the individual timepoints:

```
library(doParallel)
cl <- makeCluster(2) #or more
registerDoParallel(cl)
perms <- permu.test(cbind(Fp1,AF3,F7,F3,FC1,FC5,C3,CP1,CP5,P7,P3,Pz,P03,O1,Oz,O2,P04,
  P4,P8,CP6,CP2,C4,FC6,FC2,F4,F8,AF4,Fp2,Fz,Cz) ~ Deviant * Session | Time,data=MMN,
  parallel=TRUE)
## (output not shown)
```

```
## Loading required namespace: lmPerm
```

We can then plot the results:

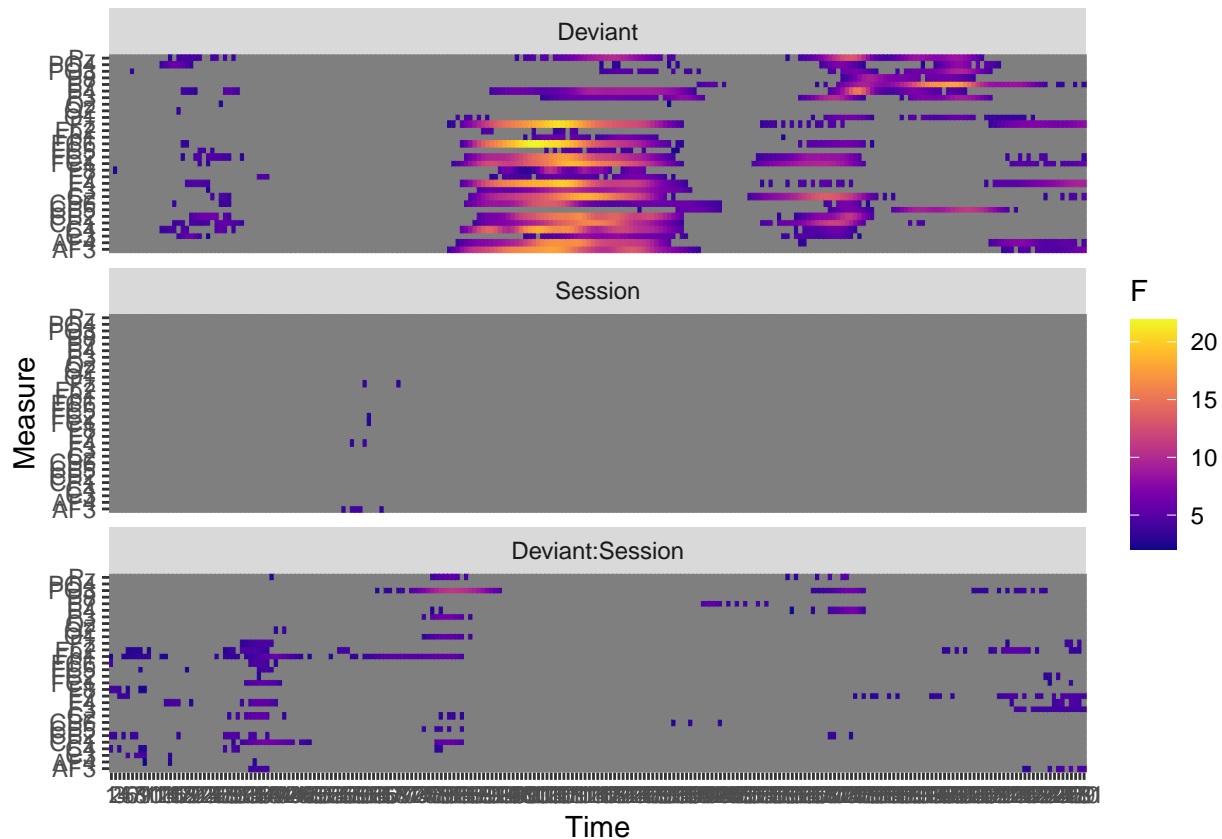
```
plot(perms)
```



Following Maris & Oostenveld (2007), the `plot` method from `permutes` by default plots F -values. Maris & Oostenveld (2007) then compute a cluster statistic over a range of temporally and/or spatially adjacent F -values that all match an inclusion criterion (e.g. $p < .05$). This cluster statistic is not explicitly calculated by `permu.test` (in the general case, `permu.test` does not know which responses in any multivariate design should be considered adjacent), but the clusters can be identified by the researcher based on their own cut-off criteria, and the researcher can then run any test they want on the subset of data falling within the cluster. If we want to stick to Maris & Oostenveld (2007), this test should be another permutation ANOVA over the whole window, but we will see below that generalized additive models offer a much better option than ANOVA, by virtue of being able to take into account the temporal and spatial correlations present in the data.

Based on visual inspection, the plot suggests one windows for the factor ‘Deviant’, located around 200 ms post-stimulus-onset. This window corresponds nicely to the window in which one usually finds MMN effects. The region of interest can be tentatively estimated as frontal and central (which is good news for (oddball?), as this is in line with prior literature on the MMN component), but we would like some verification. One option is to follow Maris & Oostenveld (2007) in discarding those results that have failed to achieve significance; we can then check again for contiguous bands that achieve large effect sizes, and base our ROI on their positions. We can do this using the `sig` option to the `plot` method.

```
plot(perms, sig='p')
```

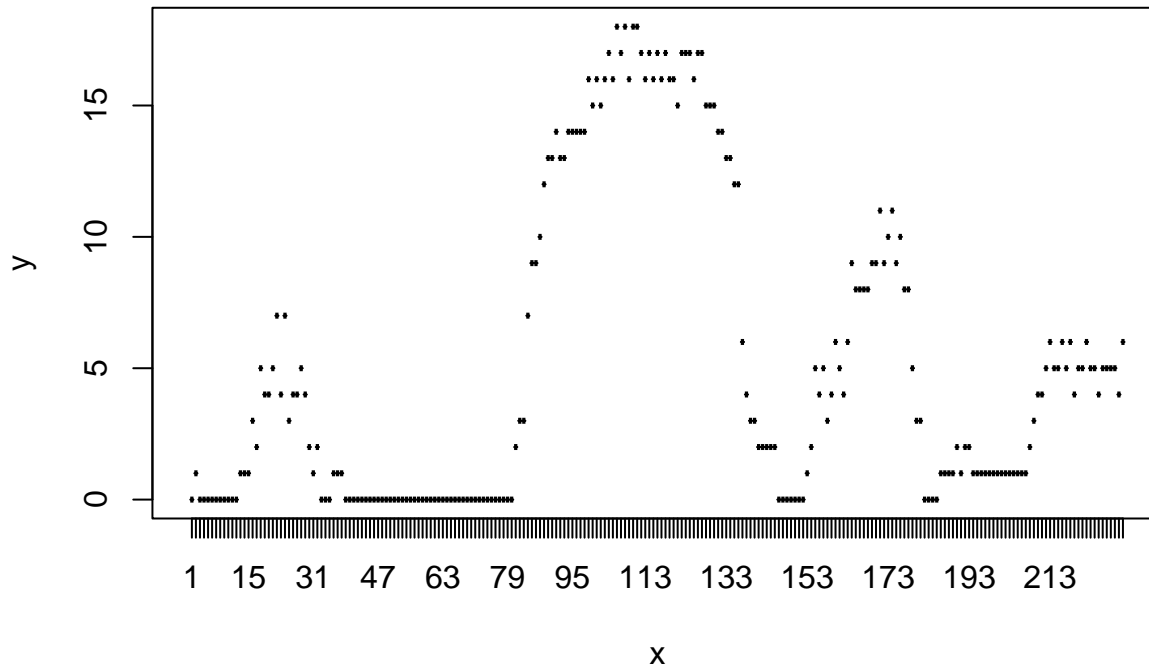


From this plot, the effect looks to be relatively robust at, indeed, the frontal and central regions. As a next step, we should determine the temporal window with more precision. In absence of a formal criterion, we simply take the narrowest window in which at least ten of the frontal and central electrodes achieve significance. Because `permutes` objects are also normal data frames, this is easily achieved.

```
ROI <- c('Fp1', 'AF3', 'F7', 'F3', 'FC1', 'FC5', 'C3', 'CP1', 'CP5', 'CP6', 'CP2', 'C4',
        'FC6', 'FC2', 'F4', 'F8', 'AF4', 'Fp2', 'Fz')
head(perms) #what do we have?

##   Time Measure      Factor      F      p      w2
## 1     1     Fp1      Deviant 0.09310588 0.66666667 0.00000000
## 2     1     Fp1      Session 0.12830533 0.70588235 0.00000000
## 3     1     Fp1 Deviant:Session 2.52786031 0.04118404 0.02100173
## 4     1     AF3      Deviant 0.18156362 1.00000000 0.00000000
## 5     1     AF3      Session 0.29049007 1.00000000 0.00000000
## 6     1     AF3 Deviant:Session 2.21753404 0.06988848 0.01674977

perms2 <- perms[perms$Factor == 'Deviant' & perms$Measure %in% ROI,]
perms2$sig <- perms2$p < .05
perms2 <- aggregate(sig ~ Time, perms2, sum)
plot(perms2$Time, perms2$sig) #look at all windows
```



```
print(perms2[perms2$sig > 10,'Time']) #our arbitrary criterion
```

```
## [1] 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106
## [20] 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
## [39] 126 127 128 129 130 131 132 133 134 135 136 171 174
## 231 Levels: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 ... 231
```

For technical reasons, the Time variable was coerced to a factor. We see that *contiguous* permutation significance was achieved from the 88th to the 136th level of this factor. The below code shows that this corresponds to a window from 171 ms to 265 ms post-stimulus-onset.

```
print(unique(MMN$Time)[c(88,136)])
```

```
## [1] 171.0938 264.8438
```

Having determined a window and an ROI, we can now proceed to the actual modeling step.

Running the actual analysis

We can now run the actual analysis that we want. We will average our data along the aforementioned window and ROI, and then run a *permutation linear mixed-effects model* to analyze the data, so that we can easily obtain robust *p*-values. The advantage of these permutation *p*-values is, again, that we could run multiple such models without requiring any Bonferroni correction. For the present single model, however, an ordinary linear mixed-effects model would also suffice.

```
data <- MMN[MMN$Time > 171 & MMN$Time < 265,]
data$amplitude <- rowMeans(data[,ROI])
data <- aggregate(amplitude ~ Deviant + Session + Subject,data,mean)
model <- perm.lmer(amplitude ~ Deviant * Session + (Deviant + Session | Subject),data)
```

Finally, we look at the resulting summary:

```
print(summary(model))
```

```
##      Factor          LRT          beta          t
## Length:4      Min.    : 0.01038      Min.    :-1.6561      Min.    :-6.2755
```

```

## Class :character  1st Qu.: 5.62154  1st Qu.: -0.7193  1st Qu.: -2.2638
## Mode  :character  Median :17.18740  Median : -0.2138  Median : -0.4856
##                               Mean  :33.58537  Mean   : -0.2036  Mean   : -0.7995
##                               3rd Qu.:45.15123  3rd Qu.: 0.3019  3rd Qu.: 0.9787
##                               Max.   :99.95630  Max.    : 1.2694  Max.    : 4.0489
##                               p
## Min.    :0.000000
## 1st Qu.:0.000000
## Median :0.007493
## Mean   :0.234515
## 3rd Qu.:0.242008
## Max.   :0.923077

```

We first observe that the highest random-effects structure that this model could support was (1 + **Session** | **Subject**). Given that the data only contain 81 observations—due to Jager’s averaging over items—this is not surprising. The main conclusions to be drawn from these results is that there is a significant mismatch negativity of -1.65 microvolts. No evidence is found for a change in this MMN over the two sessions of the experiment.

This concludes our expository foray into these data.

References

- Brybaert, M. (2007). The language-as-fixed-effect-fallacy: Some simple SPSS solutions to a complex problem. Jager, L. (in prep.).
- Maris, E., & Oostenveld, R. (2007). Nonparametric statistical testing of EEG- and MEG-data. *Journal of Neuroscience Methods*, *164*(1), 177–190. <https://doi.org/10.1016/j.jneumeth.2007.03.024>