7-26-2019

Low-Rank Multivariate General Linear Model and One-Way Random Effect Models for Brain Response Analysis

Yidan Yang
yy2513@rit.edu

Low-Rank Multivariate General Linear Model and One-Way Random Effect Models for Brain Response Analysis

By

Yidan Yang

A thesis submitted in partial fulfillment of the requirement for the Degree of Master of Science in Applied Statistics

Department of Applied Statistics

College of Science

Rochester Institute of Technology

Rochester, NY

July 26, 2019

Committee Approval:

Dr. Peter Bajorski — Date

Thesis Advisor/Committee Member

Dr. Minh Pham — Date

Thesis Advisor/Committee Member

Dr. Robert Parody — Date

Committee Member

Abstract

Human brain is the central organ of human nervous system, the activity in brain becomes a significant topic in neuroscience and medical science field. New techniques for detecting brain regions activity has been developed very fast in recent year, a basic method is functional MRIs which can measure brain activities based on oxygen level in bloodstream. This work will introduce a new approach to analyze brain region relationships through low-rank multivariate general linear model and one-way random effect model. By using fMRI and low-rank multivariate general liner model, this model contains a new penalized optimization function, which can lead to smooth HRF (Hemodynamic response functions) temporally and spatially. Also, this new model is flexible to characterize variation across different regions and stimulus types, moreover, it enables information across voxels and use fewer parameters. After analysis our fMRI data through low-rank multivariate general linear model, we apply one-way random effect model to analyze the brain regions connection via multiple subjects.

Contents

Acknowledgement

First of all, I would like to thank Dr. Peter Bajorski and Dr. Minh Pham for being my thesis advisor who always giving me guidance and invaluable research advises. Also, thank Dr. Robert Parody for giving me academic advises, willing to make time for me answering every single question, the two years in Applied Statistics program will be a precious experience to me.

In addition, I want to express my gratitude to my parents for supporting me chase my dream and being my close friends. I appreciate everything you did in our family including taking care of my grandparents, I'm so proud of being your child.

Thank my brother Jing Xu who always had his door open to give me coding advises whenever I needed and tell me what is the most important thing in people's life.

I sincerely thank Miss Yilun Zhou and Miss Ge Song for accompanying me to get through my life struggle and encouraging me when I was lost in my research. I appreciate your friendship and always being my side whenever I need you.

Most importantly, I would like to express my biggest thank to Mr. Rukawa Kaede, Mr. Sakuragi Hanamichi and Mr. Mitsui Hisashi for making me proud of being a Shohoku girl, your passion lighted my life and inspired me a lot.

Introduction

Since brain contains a very complex neuro network system, so understanding the connection and correlation among it becomes a popular and important topic in neuroscience. The nodes and their links between regions in nervous system are keys to interpret how the neuro system works.

fMRI has been an important technique to detect human brain activities by measure changes associated with brain blood flow. This method based on the fact that cerebral blood flow and neuronal activation are related, if brain blood flow to region increased, that means an area of the brain is in use.

Our fMRI data contains 4 dimensions, each of them represents time series, scans, brain regions and subjects ID. We subset 25 subjects from 820 subjects to do computation because whole 820 subject data are 3.4G which is too big. We use the original 25 subject data to do the Low-Rank Multivariate General Linear Model, checked the smooth by plot $B \cdot u$ and extracted drift coefficients Dd_i and perform random effect model on d_i to see how drift coefficient influence the whole model. Also, in random effect model, we choose subject as a random factor and approaches variance analysis on consistency between Scans.

Related Work

fMRI

Functional magnetic resonance imaging (fMRI) can give us a deep understanding of human brain activity, also, fMRI is widely used to detect neural signals from fMRI responses. But the main disadvantage of fMRI is that it can't present exact relationship between the measured fMRI signal and the underlying neural activity.

Also, fMRI provides more sophisticated brain activity data in order to extract meaningful information from it. This procedure gives us a better understanding of how brain region connected and related, analysis and interpretation of fMRI data can help us identify areas of significance on a threshold statistical map of entire brain volume.

fMRI is one type of MRI, comparing to standard MRI procedure which can only detect the physical changes from region to region in brain, fMRI procedure detects blood oxygen levels to determine brain activities associated with blood flow, can give us a better understanding of how brain regions work respect to time.

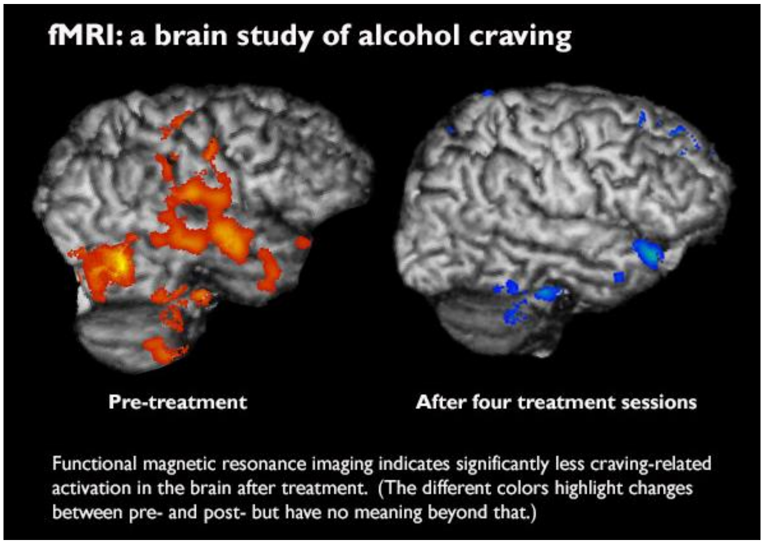


Figure 1: fMRI for a brain study of alcohol craving

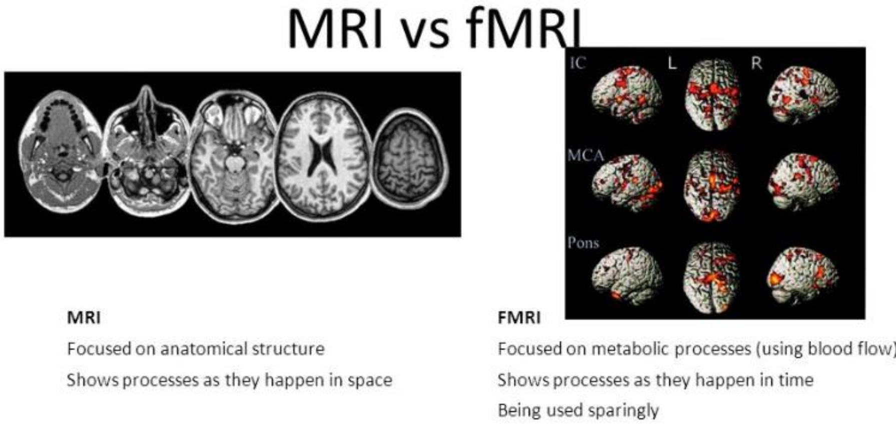


Figure 2: MRI vs fMRI

Low-Rank Multivariate General Linear Model

Given the observed fMRI time series data at brain region j of subject i, $y_j^i(t), t = \delta, 2\delta, 3\delta, \ldots, T\delta, i = 1,2,3 \ldots, n, and\ j = 1,2,3 \ldots, J$, where δ is the experiment unit time for each brain image to be captured. J represents brain regions under study either J voxels in a ROI (Regions of Interest) which defined anatomically by some brain parcellation technique in terms of their functions in brain. And i is the subject here.

The General Linear Model connects the observed fMRI data to the brain response to a stimulus through a convolution between a stimulus function and a HRF (Hemodynamic Response Functions). In this case, if we implement an fMRI experiment with K different stimuli, the GLM is

$$y_j^i(t) = D^i(t)d_j^i + \sum_{k=1}^{K} \int_0^m h_{j,k}^i(\mu) \ \cdot \nu_k^i(t-\nu)du + \epsilon_j^i(t)$$

Where $D^i(t) \in$ R is a vector at time t, $\mathrm{d}_\mathrm{j}^\mathrm{i}$ is a column vector of r coefficients, here $\nu_k^i(t-\nu), k = 1, \ldots, K$ is a known function, m is a fixed constant, and $\epsilon_j^i(t)$ is our error term. Here $\nu_k^i(t)$ represents stimulus function and describes the evoked time of the kth stimulus, more specifically, when stimulus is evoked at time t in experiment for subject i then we have $\nu_k^i(t) = 1$, otherwise it equals to 0. The function $h^i_{j,k}(\mu)$ is HRF defined on domain $[0, m]$ which characterizes subject i's brain response to the kth stimulus at brain voxel j.

Since fMRI can also detect subjects' heartbeat, motion, respiration and machine noise here we regard them as low frequency drift and we use the term $D^i(t)d^i_j$ in the GLM to characterize. From past studies, different approaches have been developed to correct for this drift effect including high-pass filter application and modeling the drift by a low-order polynomial function of time or spline functions. Here we let the covariates $D^i(t)$ be discrete cosine transform basis functions.

Standard approaches implement the GLM to one voxel's fMRI time series at one time, and independently estimate each voxel's HRFs. Here is a joint model for all voxel's fMRI data within the GLM framework. $h^i_{j,k}\ (t)$ is forth-order B-spline bases

$$h^i_{j,k}(t) = \sum_{l=1}^{L} \omega^i_{jl,k} \cdot b_{l(t)}$$

where $b_l(t)$ is basis functions.

More detailed information and explanation for how to choose basis functions is in [10].

In order to address parameter dimensions issue, we rewrite our GLM into a new form

$$Y^i = Dd_i + \sum_{k=1}^{K} X^i_k\, U^i_k V^i_k + E^i$$

where U^i_k and V^i_k are $L \times P$ and $P \times L$ matrices, here P is a given positive constant much smaller than J, in practice, we choose $P = 2$ in order to capture main differences in HRFs across subjects and voxels, it's a better way to keep model simple. This form leads to a bilinear regression model of Y^i versus X^i.

Parameter	Description
Y^i	fMRI data in a $T \times J$ matrix form for J voxels of the ith subject.
X^i_k	The $T \times L$ design matrix of the ith subject.

D	A $T \times \gamma$ matrix with the ith row equaling *D(t)*.
d^i	A $T \times J$ matrix of drift coefficients with the jth column equaling d_j^i.
V_k^i	The ith subject's $\mathrm{P} \times J$ spatial matrix characterizes voxel-specific responses of *J* voxels to the *k*th stimulus.
U_k^i	The ith subject's $\mathrm{L} \times P$ spatial matrix characterizes common HRF shapes shared across *J* voxels in response to the *k*th stimulus.

Random Effect Model

Comparing to fixed effects model, random effect model which also called variance components model is a statistical model where model parameters are random variables.

In our case, the variance consists of with-person and between-person variability.

There are two common assumptions in random effect model in order to make individual specific effect: the fixed effects assumptions and random effects assumption. It assumes that the studies are just a sample from a population of studies, the main difference between random effect model and fixed effect model is specific to the data we used in experiment.

$$Total\ var. = within\ person\ var. + between\ person\ var.$$

Statistical Analysis

Exploratory Data Analysis

The fMRI data we used to analyze comes from the Human Connectome Project. The main objective of this project is providing a better understanding of human brain connectivity, in order to study this anatomical and functional connection, the researchers provide an unparalleled compilation of neural data, an interface to graphically navigate this data, the data consist of the neuro activities sequences coming from a fMRIs.

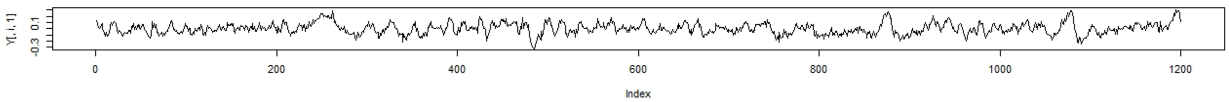
The original fMRI data is a large array with 4 dimensions:

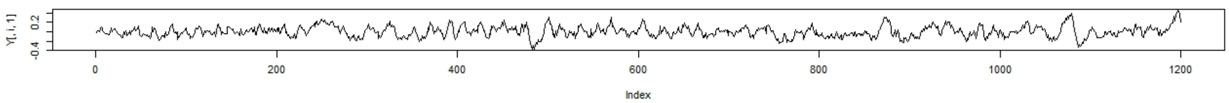
```
load("Scans.arr")
dim(Scans.arr)

## [1] 1200  116    4  820
```

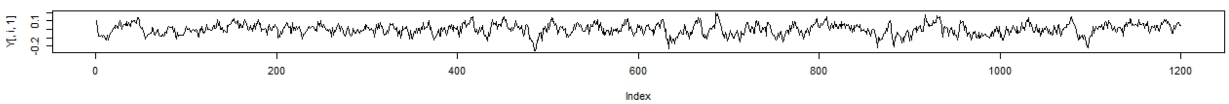
- Time Series for Brain Activity: 1200 time points.
- Regions of Interest: fMRI data contains 116 brain regions for study.
- Number of Scan: 4 scans.
- Number of Subjects: 820 subjects participated in this experiment.

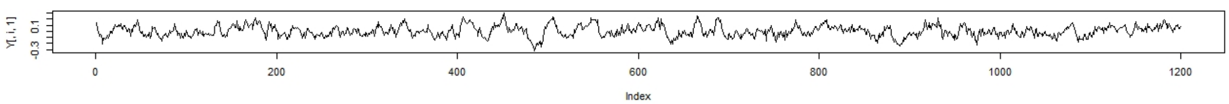
The following plot is for subject 1 in Scan 1 the brain time series activities from region 1 to region 4, different region shows different brain activities.

```
Y <- Scans.arr[,,1,]

x11()
par(mfrow=c(4,1))
for (i in 1:4) {
  plot(Y[,i,1], type = "l")
}
```









In order to perform Low-Rank Multivariate GLM efficiently, we split 1200 time series into windows with length equals to 30, so for each time series we have 1171 windows. Then the original fMRI dimension changed from *1200·116·4·820 to 30·135836·4·820* where $135836 = 1171 \cdot 116$.

MATLAB Code Comparison for Low-Rank Multivariate GLM

We process a comparison between R code and MATLAB code for Low-Rank Multivariate GLM since the original algorithm is from MATLAB code. In order to check code quickly, the data we used here is subject 1 for one scan, the dimension of *Y* train is *1200·116·1*.

The following results are first 6 output of u is for 1 iteration:

MATLAB		**R**	
Output for u		**Output for u**	
0.2027	-0.1614	0.2033418	-0.160371
0.1292	-0.1035	0.1298786	-0.102463
-0.0945	0.1122	-0.093768	0.1132575
0.1162	-0.1693	0.1168484	-0.168264
-0.1819	0.1013	-0.181278	0.1022935
-0.2162	0.228	-0.215604	0.2289555

The following is first 6 results for 20 iteration, the final output:

MATLAB		R	
Output for u		**Output for u**	
0.1058	-0.1341	0.0988771	-0.133891
-0.0228	-0.0216	-0.029603	-0.021416
0.0212	0.1834	0.0142061	0.1838667
0.0299	-0.1329	0.0237278	-0.132708
-0.0429	0.0003	-0.048315	0.0005034
-0.0214	0.0991	-0.026133	0.0994259

First 10 objective function values at the end of 20 iterations

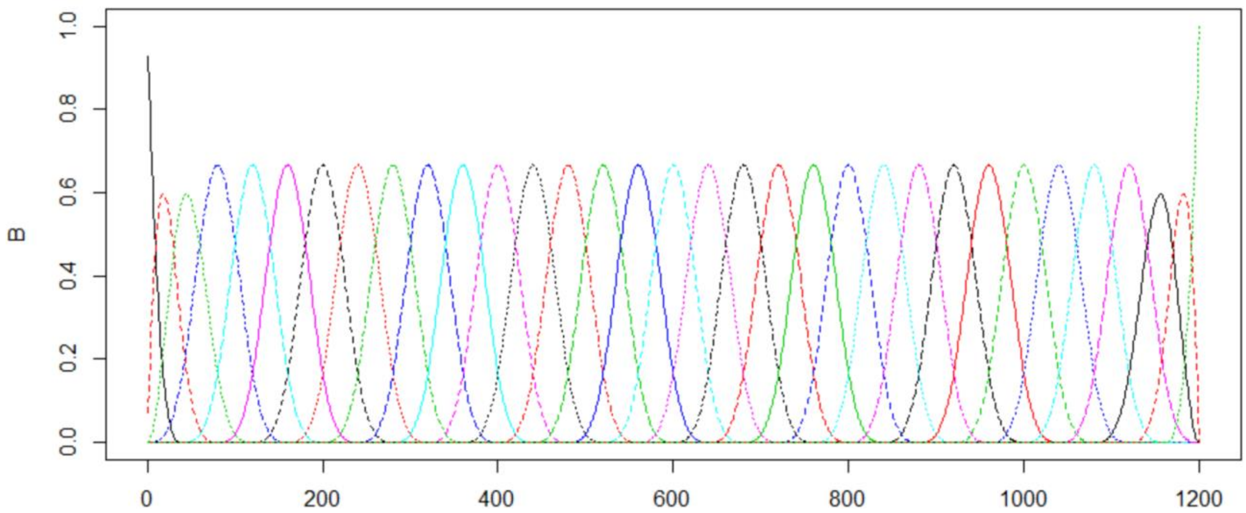
MATLAB	R
1.37E+03	[1] 1426.617
1.35E+03	[1] 1365.408
1.35E+03	[1]1356.86
1.35E+03	[1] 1353.557
1.35E+03	[1] 1351.575
1.35E+03	[1] 1350.143
1.35E+03	[1] 1349.034
1.34E+03	[1] 1348.147
1.34E+03	[1] 1347.421
1.34E+03	[1] 1346.818

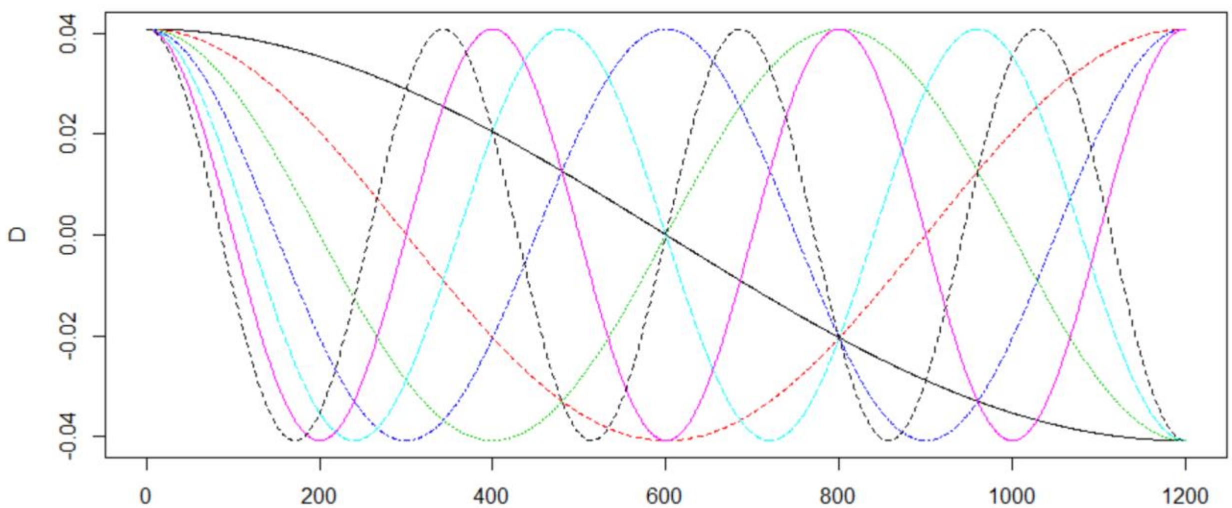
Given the table above the R output is similar to MATLAB output so the results we got from R code works.

Low-Rank Multivariate General Linear Model

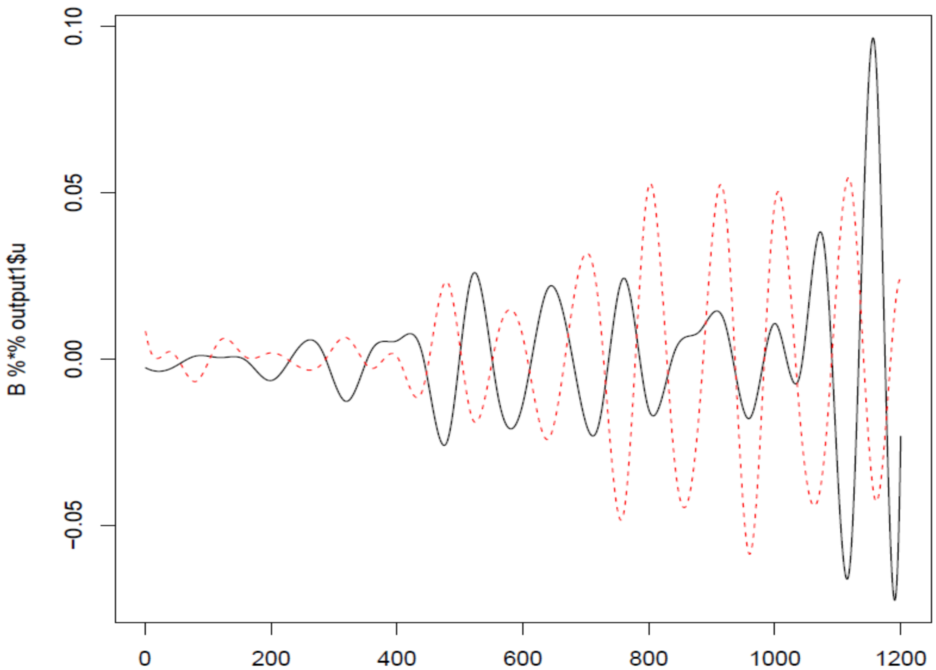
Original fMRI Data

The original fMRI data has 4 dimensions each of them is (1200, 116, 4, 820), before we perform GLM we check basis *B* and *D*.





We combined the fMRI data into 3 dimensions by doing multiplication with Scans and Subjects so it becomes (1200, 116, 4·820), we plot the results for $B \cdot u$:



Reformat fMRI Data

Original fMRI data contain 1200 time series points, we split it into 1171 windows each one has length of 30. Since we doing so, the data set will become too large and it's difficult to implement algorithm on PC or even server. In this case, we subset 25 subjects from the reformatted fMRI data, so the dimension becomes (30, 135836, 4, 25). Since Low-Rank Multivariate General Linear Model prefers third-dimensional data then we combine we data by multiplying Scans and Subjects, the data dimensions switched to (30, 135836, 100).

The change of data dimension leads basis in Low-Rank Multivariate GLM changed, therefore, we listed the dimension of each variable and parameter in our model.

	Original fMRI Dimension	**Reformat fMRI Dimension**
Y	1200*116	30*135836
D	1200*7	30*7
d	7*116	7*135836
B	1200*33	30*33
u	33*2	33*2
v	2*116	2*135836

Random Effect Model

We extracted drift coefficient d_i from output of Low-Rank Multivariate GLM and perform random effect model on it. In this model predictor X is each subject's scan, since we have 25 subjects here so the length of predictor is 100. For response d_i, it's a vector of length 100, where $Y_1 = [1,1,\cdot]$.

Predictor	Person
X	Y
1	Y1
1	Y2
1	Y3
1	Y4
2	Y5
2	Y6
2	Y7
2	Y8
…	…
25	Y97
25	Y98
25	Y99
25	Y100

The random effect model for $Y_1 = Y[1,1,\cdot]$ shows as following:

Analysis of Variance

Source	DF	Seq SS	Contribution	Adj SS	Adj MS	F-Value	P-Value
person	24	0.9316	24.46%	0.9316	0.03882	1.01	0.463
Error	75	2.8771	75.54%	2.8771	0.03836		
Total	99	3.8086	100.00%				

The variance within and between person is following:

Variance Components, using Adjusted SS

Source	Variance	% of Total	StDev	% of Total
person	0.0001137	0.30%	0.010661	5.44%

Error	0.0383610	99.70%	0.195860	99.85%
Total	0.0384746		0.196150	

Using R to perform the whole random effect model calculated variances between person, we used d_i which dimension is (7, 1171·116, 4·25) and we picked every element in 7 and 116 to see how the variances change. Also, we changed all negative values to 0, the results showing as below.

```
# output.d

setwd('C:\\Users\\yancx\\Desktop\\Thesis\\Peter fMRI data\\')
load("output.d")


dim(output.d)

## [1]      7 135836    100

Y1 <- output.d[1,1,]
Y1 <- matrix(data = Y1 , ncol =1)
Y2 <- output.d[2,1,]
Y2 <- matrix(data = Y2 , ncol =1)
Y3 <- output.d[3,1,]
Y3 <- matrix(data = Y3 , ncol =1)
Y4 <- output.d[4,1,]
Y4 <- matrix(data = Y4 , ncol =1)
Y5 <- output.d[5,1,]
Y5 <- matrix(data = Y5 , ncol =1)
Y6 <- output.d[6,1,]
Y6 <- matrix(data = Y6 , ncol =1)
Y7 <- output.d[7,1,]
Y7 <- matrix(data = Y7 , ncol =1)


extract.116.test <- function(Y){
  out.arr <- array(data = 0, dim = c(100,1,7,116))
  for (j in 1:116) {
    for (i in 1:7) {
      out.arr[,,i,j] <- Y[i,(1171*(j-1)+1),]
    }
  }
  out.arr
}
out.array <- extract.116.test(output.d)


out.df <- data.frame(data=out.array)
person <- paste(rep(1:25, each=4), sep = "")
```

```r
out.df <- cbind(person, out.df)


colnames(out.df) <- c("person", 1:812)


adjsq.func.d <- function(data){
  out.var <- matrix(data = 0, nrow = 812)
  for (i in 2:813) {
  out.var[(i-1),] <- summary(lm(out.df[,i] ~ person, data = out.df))$adj.r.sq
uared
  }
  out.var <- matrix(data = out.var, nrow = 7, ncol = 116)
  out.var
}
adjsq.out.d <- adjsq.func.d(out.df)

adjsq.out.d [adjsq.out.d  < 0] <- 0
adjsq.out.d
```

```
##             [,1]        [,2]       [,3] [,4]       [,5]        [,6]
## [1,] 0.002864811 0.051109726 0.00000000    0 0.08395585 0.000000000
## [2,] 0.000000000 0.006932619 0.06460687    0 0.00000000 0.003343756
## [3,] 0.000000000 0.000000000 0.00000000    0 0.00000000 0.094422974
## [4,] 0.039023094 0.041259533 0.01321886    0 0.03330971 0.000000000
## [5,] 0.042609648 0.000000000 0.14117112    0 0.01824489 0.000000000
## [6,] 0.066406178 0.000000000 0.07525265    0 0.00000000 0.004734788
## [7,] 0.000000000 0.000000000 0.03503993    0 0.00000000 0.037442966
##            [,7]        [,8]       [,9]      [,10]      [,11]      [,12]
## [1,] 0.06029930 0.000000000 0.00000000 0.03838504 0.00000000 0.04888318
## [2,] 0.09565393 0.000000000 0.00000000 0.00000000 0.07114471 0.04454667
## [3,] 0.01115500 0.035092709 0.00000000 0.00000000 0.00000000 0.00000000
## [4,] 0.00000000 0.030354570 0.00000000 0.00000000 0.08755001 0.09586695
## [5,] 0.00000000 0.000000000 0.00000000 0.00000000 0.04983520 0.01061622
## [6,] 0.00000000 0.002793398 0.05255703 0.00000000 0.00000000 0.00000000
## [7,] 0.03646037 0.000000000 0.00000000 0.00000000 0.04225066 0.07786538
##           [,13]      [,14]      [,15]      [,16]      [,17]      [,18]
## [1,] 0.03381266 0.00000000 0.00000000 0.00000000 0.00000000 0.03099220
## [2,] 0.00000000 0.00000000 0.00000000 0.04780182 0.00000000 0.00000000
## [3,] 0.08924529 0.01477926 0.00000000 0.00000000 0.03903037 0.07165699
## [4,] 0.00000000 0.00000000 0.05308367 0.05262786 0.01037373 0.00000000
## [5,] 0.01977072 0.00000000 0.00000000 0.00000000 0.02203492 0.04548164
## [6,] 0.17170676 0.19629084 0.00000000 0.00000000 0.00000000 0.06033811
## [7,] 0.00000000 0.00000000 0.00000000 0.00000000 0.05955933 0.00000000
##           [,19]      [,20]      [,21]     [,22]      [,23]      [,24]
## [1,] 0.02602517 0.00000000 0.00000000 0.0000000 0.04721299 0.00000000
## [2,] 0.10574114 0.10580253 0.01523745 0.0000000 0.00000000 0.00000000
## [3,] 0.00000000 0.00000000 0.00000000 0.0000000 0.02595492 0.00000000
## [4,] 0.06832325 0.00000000 0.00000000 0.0000000 0.00000000 0.09907724
## [5,] 0.06014204 0.00000000 0.00000000 0.0256542 0.00000000 0.01942983
```

```
## [6,] 0.00000000 0.08615232 0.14158095 0.0000000 0.16180978 0.00000000
## [7,] 0.00000000 0.00000000 0.03888415 0.0000000 0.00000000 0.00000000
##            [,25]       [,26]       [,27]       [,28]      [,29]     [,30]
## [1,] 0.00000000 0.000000000 0.000000000 0.000000000 0.00000000 0.0000000
## [2,] 0.00000000 0.000000000 0.000000000 0.041893592 0.00000000 0.0000000
## [3,] 0.00000000 0.056764486 0.004009404 0.000000000 0.00000000 0.1739786
## [4,] 0.02819383 0.019394993 0.038512353 0.051900352 0.07244422 0.0000000
## [5,] 0.00000000 0.001191753 0.000000000 0.007060772 0.12564854 0.1546246
## [6,] 0.00000000 0.000000000 0.000000000 0.045972190 0.20091704 0.0000000
## [7,] 0.00000000 0.000000000 0.016653902 0.047365398 0.00000000 0.0000000
##            [,31]       [,32]      [,33]       [,34]       [,35]       [,36]
## [1,] 0.12801584 0.000000000 0.00000000 0.002249931 0.000000000 0.000000000
## [2,] 0.10011492 0.074054560 0.02386692 0.006246562 0.003546431 0.014726505
## [3,] 0.01154893 0.000000000 0.00000000 0.004888754 0.000000000 0.206164321
## [4,] 0.00000000 0.071944003 0.00000000 0.019695595 0.000000000 0.000000000
## [5,] 0.00000000 0.025624316 0.00000000 0.126962377 0.000000000 0.014127579
## [6,] 0.00000000 0.000000000 0.00000000 0.000000000 0.083960188 0.021755114
## [7,] 0.00000000 0.009471564 0.05202625 0.000000000 0.183750604 0.008544853
##              [,37]       [,38]      [,39]      [,40]       [,41]      [,42]
## [1,] 0.0860245912 0.000000000 0.00000000 0.15741932 0.082430839 0.00000000
## [2,] 0.0000000000 0.042445182 0.00000000 0.00000000 0.000000000 0.11094985
## [3,] 0.0008639233 0.000000000 0.19770906 0.07836757 0.000000000 0.09770056
## [4,] 0.0000000000 0.000000000 0.00000000 0.00000000 0.004593663 0.06309648
## [5,] 0.0000000000 0.013320127 0.05619886 0.00000000 0.000000000 0.04538828
## [6,] 0.0000000000 0.004167038 0.00000000 0.00000000 0.038554375 0.00000000
## [7,] 0.0000000000 0.037255221 0.05193283 0.09302802 0.000000000 0.00000000
##             [,43]       [,44]       [,45]       [,46]      [,47]      [,48]
## [1,] 0.008019067 0.039412380 0.003348899 0.037856923 0.00000000 0.08455425
## [2,] 0.000000000 0.000000000 0.077383160 0.039914197 0.00000000 0.03658319
## [3,] 0.000000000 0.091232407 0.000000000 0.055117799 0.00000000 0.00000000
## [4,] 0.000000000 0.122299996 0.000000000 0.058607572 0.05367664 0.00000000
## [5,] 0.025102891 0.073933452 0.121383217 0.008786756 0.00000000 0.00000000
## [6,] 0.044603522 0.003990717 0.008739612 0.061321979 0.10332840 0.00000000
## [7,] 0.000000000 0.021993993 0.005465797 0.000000000 0.15728904 0.00000000
##            [,49]      [,50]      [,51]      [,52]       [,53]       [,54]
## [1,] 0.00000000 0.06980252 0.00000000 0.05980106 0.189347567 0.102261779
## [2,] 0.00000000 0.13534993 0.00000000 0.07414676 0.000000000 0.003815112
## [3,] 0.04816867 0.00000000 0.00000000 0.11373938 0.000000000 0.000000000
## [4,] 0.00000000 0.02009737 0.02597275 0.00000000 0.000000000 0.004248239
## [5,] 0.00000000 0.06920988 0.21891979 0.00000000 0.009581625 0.000000000
## [6,] 0.02911011 0.01366715 0.00000000 0.02208593 0.000000000 0.173953208
## [7,] 0.14809645 0.00000000 0.00000000 0.00000000 0.003839101 0.000000000
##             [,55]       [,56]      [,57]      [,58]      [,59]      [,60]
## [1,] 0.000000000 0.068258225 0.17680080 0.00000000 0.02914794 0.00000000
## [2,] 0.000000000 0.029281896 0.00000000 0.00000000 0.01600500 0.00000000
## [3,] 0.000000000 0.042924423 0.16910370 0.09198879 0.14155941 0.04320880
## [4,] 0.000000000 0.000000000 0.00000000 0.00000000 0.00000000 0.01283095
## [5,] 0.000000000 0.000000000 0.04322730 0.00000000 0.07394329 0.00000000
## [6,] 0.001359142 0.000000000 0.00167188 0.13424147 0.00000000 0.00000000
## [7,] 0.000000000 0.009367426 0.00000000 0.00000000 0.14579679 0.08194326
```

```
##            [,61]      [,62]       [,63]      [,64]        [,65]       [,66]
## [1,] 0.00000000 0.00000000 0.000000000 0.06393752 0.0006091796 0.000000000
## [2,] 0.00000000 0.09119744 0.107296725 0.00000000 0.0000000000 0.054364879
## [3,] 0.00000000 0.07643028 0.000000000 0.00000000 0.0882008413 0.007144671
## [4,] 0.00000000 0.00000000 0.068660967 0.00000000 0.0000000000 0.036154022
## [5,] 0.00000000 0.08073361 0.064993789 0.10854150 0.0064555243 0.107405286
## [6,] 0.00000000 0.00000000 0.007199566 0.00000000 0.0080870424 0.000000000
## [7,] 0.03000737 0.11161749 0.000000000 0.03245929 0.0000000000 0.000000000
##            [,67]      [,68]      [,69]      [,70]       [,71]      [,72]
## [1,] 0.00000000 0.00000000 0.00000000 0.00000000 0.100887022 0.00000000
## [2,] 0.00000000 0.16125414 0.03096020 0.01713514 0.000000000 0.00000000
## [3,] 0.00000000 0.00000000 0.04096226 0.00000000 0.004165561 0.01166727
## [4,] 0.00000000 0.00000000 0.06833315 0.06360833 0.048475512 0.00000000
## [5,] 0.09768125 0.01980067 0.00000000 0.00000000 0.123271304 0.01021297
## [6,] 0.16659372 0.10244558 0.11261862 0.00000000 0.000000000 0.11303919
## [7,] 0.00000000 0.00000000 0.09517219 0.10220143 0.000000000 0.00000000
##            [,73]      [,74]      [,75]     [,76]      [,77]    [,78]
## [1,] 0.00000000 0.00000000 0.06835379 0.1207871 0.06945157 0.000000
## [2,] 0.18143603 0.05450577 0.16484677 0.0000000 0.00000000 0.144655
## [3,] 0.02705203 0.00000000 0.00000000 0.0000000 0.00000000 0.000000
## [4,] 0.00000000 0.00000000 0.00000000 0.0000000 0.00000000 0.000000
## [5,] 0.05524901 0.08793475 0.00000000 0.0000000 0.17383369 0.000000
## [6,] 0.22749010 0.00000000 0.03235672 0.0000000 0.13991216 0.000000
## [7,] 0.13510356 0.00000000 0.04340497 0.1203103 0.01512268 0.000000
##            [,79]      [,80]      [,81]      [,82]       [,83]      [,84]
## [1,] 0.03627100 0.00000000 0.00000000 0.00000000 0.000000000 0.00000000
## [2,] 0.00000000 0.03448429 0.00000000 0.00000000 0.081571452 0.00000000
## [3,] 0.02940218 0.00000000 0.00000000 0.00000000 0.228059635 0.06016105
## [4,] 0.00000000 0.00000000 0.01447423 0.00000000 0.112585299 0.00000000
## [5,] 0.12823724 0.01862143 0.00000000 0.06322224 0.003944239 0.03797253
## [6,] 0.00000000 0.00000000 0.11915822 0.12597163 0.040676400 0.00000000
## [7,] 0.00000000 0.04712014 0.00000000 0.03935247 0.015113309 0.00000000
##            [,85]      [,86]      [,87]      [,88]       [,89]      [,90]
## [1,] 0.00000000 0.00000000 0.00000000 0.14954958 0.018090545 0.00000000
## [2,] 0.00000000 0.00000000 0.03073461 0.00000000 0.000000000 0.04613243
## [3,] 0.02274665 0.00000000 0.00000000 0.00000000 0.000000000 0.05809589
## [4,] 0.04431817 0.00000000 0.01363610 0.00000000 0.000000000 0.01950323
## [5,] 0.22426885 0.00000000 0.00000000 0.05319235 0.000000000 0.08116166
## [6,] 0.01867640 0.09946475 0.13585924 0.05643614 0.000000000 0.10349871
## [7,] 0.00000000 0.00000000 0.07125544 0.01625181 0.001462811 0.02697592
##              [,91]     [,92]      [,93]       [,94]      [,95]      [,96]
## [1,] 0.0494611079 0.0000000 0.08351265 0.214037391 0.00914790 0.00000000
## [2,] 0.0000000000 0.0415978 0.00000000 0.000000000 0.00000000 0.00000000
## [3,] 0.0903181152 0.0000000 0.00000000 0.000000000 0.03158618 0.07238027
## [4,] 0.0008199938 0.1063050 0.09751430 0.007683116 0.10745849 0.01889672
## [5,] 0.0000000000 0.0000000 0.05876960 0.000000000 0.11438855 0.07258774
## [6,] 0.0000000000 0.1201576 0.00000000 0.000000000 0.00000000 0.03099058
## [7,] 0.1371742170 0.0000000 0.05899140 0.000000000 0.00000000 0.02868906
##             [,97]      [,98]       [,99]     [,100]     [,101]     [,102]
## [1,] 0.000000000 0.00000000 0.009161775 0.06093865 0.00000000 0.00000000
```

```
## [2,] 0.000000000 0.00000000 0.017525953 0.06423971 0.00000000 0.11534202
## [3,] 0.000000000 0.00000000 0.071850230 0.07639097 0.00000000 0.00000000
## [4,] 0.000000000 0.00000000 0.027167486 0.01252082 0.00000000 0.00000000
## [5,] 0.008937282 0.04412763 0.000000000 0.00000000 0.00000000 0.00000000
## [6,] 0.000000000 0.00000000 0.000000000 0.06540994 0.04242772 0.08955566
## [7,] 0.000000000 0.00000000 0.000000000 0.00000000 0.00000000 0.00000000
##          [,103]     [,104]     [,105]      [,106]     [,107]      [,108]
## [1,] 0.20527598 0.00000000 0.00000000 0.000000000 0.05706452 0.005671669
## [2,] 0.10035639 0.11176257 0.00000000 0.000000000 0.16966485 0.000000000
## [3,] 0.00000000 0.18468242 0.00000000 0.000000000 0.04503687 0.000000000
## [4,] 0.00000000 0.00000000 0.00000000 0.000000000 0.00000000 0.010368458
## [5,] 0.05937177 0.00000000 0.03679596 0.003712446 0.19231352 0.000000000
## [6,] 0.12274669 0.00000000 0.09088649 0.004845472 0.21772917 0.000000000
## [7,] 0.00000000 0.07483901 0.12082691 0.117219498 0.00000000 0.000000000
##           [,109]     [,110]     [,111]     [,112]      [,113]       [,114]
## [1,] 0.009110803 0.00000000 0.00000000 0.00000000 0.000000000 0.0000000000
## [2,] 0.024674370 0.00000000 0.00000000 0.00000000 0.000000000 0.0460288514
## [3,] 0.000000000 0.07552638 0.00000000 0.08566843 0.000000000 0.0000000000
## [4,] 0.064905410 0.06169280 0.04030871 0.00000000 0.010404594 0.0006019767
## [5,] 0.000000000 0.00000000 0.03274447 0.00000000 0.037689555 0.0525831787
## [6,] 0.003620807 0.17325126 0.02904073 0.02610422 0.000000000 0.0000000000
## [7,] 0.000000000 0.00000000 0.00000000 0.03213279 0.000153408 0.0816243435
##          [,115]     [,116]
## [1,] 0.00000000 0.09867671
## [2,] 0.04890933 0.00000000
## [3,] 0.00000000 0.00000000
## [4,] 0.00000000 0.00000000
## [5,] 0.02059910 0.00000000
## [6,] 0.01028657 0.05148423
## [7,] 0.06612941 0.02849868
```

```
p.value.func <- function(data){
  out.pvalue <- matrix(data = 0, nrow = 812)
  for (i in 2:813) {
    out.pvalue[(i-1),] <- anova(lm(out.df[,i] ~ person, data = out.df))$'Pr(>
F)'[1]
  }
  out.pvalue <- matrix(data = out.pvalue, nrow = 7, ncol = 116)
  out.pvalue
}
p.value.d <- p.value.func(out.df)
p.value.d
```

```
##           [,1]      [,2]       [,3]      [,4]      [,5]      [,6]
## [1,] 0.4631472 0.2518422 0.78795461 0.8584634 0.1482707 0.6665094
## [2,] 0.8641300 0.4433573 0.20487908 0.9312179 0.7912815 0.4608056
## [3,] 0.4928441 0.5864707 0.76394218 0.6891010 0.8462222 0.1228032
## [4,] 0.2989404 0.2898819 0.41327843 0.5642688 0.3227551 0.9099433
## [5,] 0.2844873 0.6703872 0.04712949 0.9652749 0.3897567 0.7014719
## [6,] 0.1990817 0.9882254 0.17217470 0.8254022 0.6828591 0.4540216
```

```
## [7,] 0.7171802 0.7976702 0.31544324 0.8302486 0.5599562 0.3054313
##           [,7]      [,8]      [,9]     [,10]     [,11]     [,12]
## [1,] 0.2192033 0.5479120 0.6616093 0.3015525 0.7700879 0.2601696
## [2,] 0.1200366 0.6225575 0.6408167 0.8524885 0.1843397 0.2768463
## [3,] 0.4230789 0.3152216 0.5039007 0.8628914 0.6228148 0.7654918
## [4,] 0.5529301 0.3354372 0.9782893 0.4996553 0.1391286 0.1195626
## [5,] 0.8551583 0.7449486 0.4989386 0.9475570 0.2565893 0.4256499
## [6,] 0.9638294 0.4634965 0.2465160 0.7120539 0.8539728 0.5047709
## [7,] 0.3095048 0.5166885 0.5133843 0.6022832 0.2859162 0.1647333
##           [,13]     [,14]     [,15]     [,16]     [,17]     [,18]
## [1,] 0.32062089 0.5304461 0.8646450 0.5717341 0.8760235 0.3326805
## [2,] 0.95559525 0.9930970 0.6249132 0.2642719 0.5048706 0.6214664
## [3,] 0.13496183 0.4059216 0.5364419 0.7261277 0.2989107 0.1827916
## [4,] 0.96995792 0.8362228 0.2445950 0.2462571 0.4268087 0.9874610
## [5,] 0.38272051 0.7798097 0.5477439 0.8165043 0.3723757 0.2732001
## [6,] 0.02274995 0.0119239 0.5216390 0.5750363 0.8330557 0.2190715
## [7,] 0.98743997 0.7065355 0.7718847 0.6516806 0.2217270 0.5360936
##           [,19]      [,20]      [,21]     [,22]      [,23]     [,24]
## [1,] 0.35444075 0.53707961 0.78347200 0.9007253 0.26652151 0.8994180
## [2,] 0.09909898 0.09898077 0.40377037 0.7729331 0.73009639 0.8550219
## [3,] 0.98894730 0.71588138 0.91533442 0.8562280 0.35475312 0.6244636
## [4,] 0.19302582 0.63734507 0.94579957 0.7682407 0.85476344 0.1125880
## [5,] 0.21973811 0.83956426 0.59718692 0.3560916 0.80323029 0.3842880
## [6,] 0.56218929 0.14263376 0.04669622 0.5432360 0.02906608 0.6012990
## [7,] 0.59866823 0.53239475 0.29950819 0.6137298 0.70489440 0.9322269
##          [,25]     [,26]     [,27]     [,28]      [,29]      [,30]
## [1,] 0.5563692 0.6098682 0.8281682 0.9777131 0.63383264 0.53019135
## [2,] 0.5382771 0.6109307 0.8256946 0.2873414 0.58798226 0.52043614
## [3,] 0.6103766 0.2314250 0.4575561 0.7980839 0.86604373 0.02147962
## [4,] 0.3448603 0.3844484 0.3010303 0.2489241 0.18042976 0.57789374
## [5,] 0.8227906 0.4713486 0.5061428 0.4427376 0.06615116 0.03453812
## [6,] 0.9952779 0.6873688 0.5396619 0.2712982 0.01049575 0.56144165
## [7,] 0.6923038 0.7997132 0.3971468 0.2659382 0.69752061 0.80674897
##           [,31]     [,32]     [,33]      [,34]      [,35]       [,36]
## [1,] 0.06290277 0.7551371 0.5952418 0.46615749 0.65070256 0.764779055
## [2,] 0.11040071 0.1756638 0.3640933 0.44667868 0.45981557 0.406169555
## [3,] 0.42120232 0.7955439 0.7434496 0.45327228 0.57655015 0.009060735
## [4,] 0.96677931 0.1819280 0.5712128 0.38306572 0.75445705 0.679987098
## [5,] 0.79365242 0.3562248 0.7932059 0.06433201 0.69341372 0.408988372
## [6,] 0.59429030 0.8795474 0.8692145 0.87284034 0.14825943 0.373647697
## [7,] 0.63357724 0.4311286 0.2484614 0.94996359 0.01668929 0.435580194
##          [,37]     [,38]      [,39]      [,40]     [,41]      [,42]
## [1,] 0.1429573 0.7035551 0.48200088 0.03231428 0.1522775 0.63680515
## [2,] 0.6889130 0.2851414 0.83272906 0.60299895 0.7085200 0.08945085
## [3,] 0.4729595 0.8880002 0.01146889 0.16332917 0.6696581 0.11554033
## [4,] 0.9200581 0.5095226 0.49370762 0.89049549 0.4547087 0.20983020
## [5,] 0.9318317 0.4127996 0.23341954 0.56185923 0.9800809 0.27356299
## [6,] 0.9684917 0.4567874 0.98578822 0.50113638 0.3008581 0.47764184
## [7,] 0.8663686 0.3062074 0.24880470 0.12599564 0.7321170 0.83930100
##          [,43]      [,44]      [,45]     [,46]      [,47]     [,48]
```

```
## [1,] 0.4381120 0.29735279 0.46078045 0.3037236 0.48980289 0.1467194
## [2,] 0.9689428 0.61410539 0.16608945 0.2953129 0.72884340 0.3089941
## [3,] 0.6330939 0.13019510 0.71577530 0.2372612 0.90767547 0.6834939
## [4,] 0.9183534 0.07097544 0.47839570 0.2250002 0.24244301 0.5405539
## [5,] 0.3585514 0.17601918 0.07234427 0.4344168 0.86440366 0.5093069
## [6,] 0.2766238 0.45764721 0.43464349 0.2157458 0.10383143 0.8395525
## [7,] 0.6856412 0.37256164 0.45046694 0.6374335 0.03241518 0.7968369
##          [,49]      [,50]       [,51]      [,52]      [,53]      [,54]
## [1,] 0.6804293 0.18843812 0.917237518 0.22090057 0.01438853 0.10597790
## [2,] 0.7942268 0.05365111 0.695386821 0.17539358 0.66292620 0.45850395
## [3,] 0.2628760 0.58866502 0.959568203 0.08459415 0.89127935 0.76591347
## [4,] 0.5257434 0.38122086 0.354673808 0.52854417 0.68981138 0.45639154
## [5,] 0.6415112 0.19026717 0.006272811 0.80152428 0.43060088 0.95557359
## [6,] 0.3408492 0.41116007 0.924567102 0.37214399 0.89541255 0.02149349
## [7,] 0.0402402 0.66746327 0.845771480 0.61169010 0.45838689 0.79300240
##          [,55]     [,56]      [,57]      [,58]      [,59]     [,60]
## [1,] 0.7134764 0.1932292 0.01998726 0.52797895 0.34068407 0.6893990
## [2,] 0.9206923 0.3400996 0.84381868 0.61554909 0.40017612 0.7325604
## [3,] 0.7073774 0.2832376 0.02428451 0.12841370 0.04671891 0.2821113
## [4,] 0.7134155 0.5013753 0.71950483 0.88043076 0.57102211 0.4151145
## [5,] 0.8055791 0.9878513 0.28203809 0.79439990 0.17599029 0.8183322
## [6,] 0.4705266 0.8944656 0.46899165 0.05497325 0.66808120 0.5673481
## [7,] 0.5990577 0.4316282 0.99358339 0.96646924 0.04242796 0.1535748
##          [,61]      [,62]      [,63]      [,64]     [,65]      [,66]
## [1,] 0.8862816 0.70249836 0.51139344 0.20706371 0.4742120 0.62099584
## [2,] 0.4925258 0.13027791 0.09613705 0.73403712 0.6086673 0.23995982
## [3,] 0.6723115 0.16879225 0.99513377 0.90356042 0.1375180 0.44233209
## [4,] 0.8650474 0.50704018 0.19197192 0.53780902 0.6889704 0.31078059
## [5,] 0.6911400 0.15682712 0.20362316 0.09381663 0.4456663 0.09593293
## [6,] 0.5549301 0.68242947 0.44206681 0.62961465 0.4377845 0.53575861
## [7,] 0.3369430 0.08826907 0.78599891 0.32637994 0.6587664 0.58110264
##           [,67]      [,68]      [,69]     [,70]      [,71]      [,72]
## [1,] 0.61661061 0.61201050 0.92403834 0.7207636 0.10879422 0.93121086
## [2,] 0.47996629 0.02946115 0.33281858 0.3949058 0.72481062 0.99486476
## [3,] 0.75634057 0.80231492 0.29107722 0.8969714 0.45679458 0.42063908
## [4,] 0.49887058 0.84461261 0.19299489 0.2081437 0.26171177 0.55612971
## [5,] 0.11558214 0.38258291 0.48369971 0.4990610 0.06954791 0.42757744
## [6,] 0.02584754 0.10560562 0.08651994 0.9218350 0.58153842 0.08579328
## [7,] 0.80828183 0.62468448 0.12111370 0.1061004 0.97428191 0.59777585
##            [,73]     [,74]     [,75]      [,76]      [,77]      [,78]
## [1,] 0.667058010 0.7072024 0.1929304 0.07324564 0.18951982 0.78234136
## [2,] 0.017730710 0.2394534 0.0269856 0.50969265 0.63704180 0.04355053
## [3,] 0.349889306 0.9944343 0.7097333 0.63474446 0.72566125 0.48557158
## [4,] 0.774193006 0.9254148 0.6527349 0.74196685 0.99882244 0.71884893
## [5,] 0.236792902 0.1381748 0.5259841 0.89964404 0.02155880 0.72330424
## [6,] 0.004859163 0.7121216 0.3268185 0.86727404 0.04848132 0.89822607
## [7,] 0.053942697 0.9891554 0.2813358 0.07397298 0.40430883 0.64862291
##          [,79]     [,80]      [,81]      [,82]       [,83]     [,84]
## [1,] 0.3102931 0.8969315 0.83219465 0.92566621 0.991810375 0.7681596
## [2,] 0.7180196 0.3177821 0.91567702 0.62688096 0.154569279 0.5324568
```

```
## [3,] 0.3395753 0.8907586 0.56068699 0.78590627 0.004776266 0.2196734
## [4,] 0.6023597 0.7453633 0.40735603 0.99322113 0.086577715 0.7648168
## [5,] 0.0626057 0.3880156 0.65893980 0.20941501 0.457873928 0.3032477
## [6,] 0.7031731 0.5211729 0.07575416 0.06570001 0.292229206 0.6979886
## [7,] 0.9148319 0.2668772 0.96250301 0.29759685 0.404352822 0.8414569
##           [,85]     [,86]      [,87]      [,88]     [,89]     [,90]
## [1,] 0.99817984 0.5919962 0.90242787 0.03890705 0.3904713 0.9153404
## [2,] 0.75204623 0.6157334 0.33379282 0.73650352 0.9594037 0.2706786
## [3,] 0.36914846 0.5879811 0.83888282 0.53355567 0.9128614 0.2267724
## [4,] 0.27774153 0.9693068 0.41130669 0.75240228 0.9972134 0.3839503
## [5,] 0.00535286 0.7944734 0.91840102 0.24419971 0.9998173 0.1556707
## [6,] 0.38776166 0.1117674 0.05305243 0.23258153 0.6185764 0.1034918
## [7,] 0.81447545 0.9916051 0.18400432 0.39902291 0.4700177 0.3502257
##           [,91]      [,92]     [,93]       [,94]      [,95]     [,96]
## [1,] 0.25799270 0.76462069 0.1494273 0.007233441 0.43268179 0.8440924
## [2,] 0.55808128 0.28852505 0.7434820 0.639143466 0.90893671 0.9307423
## [3,] 0.13237267 0.75999810 0.8893595 0.483877485 0.33012246 0.1806208
## [4,] 0.47317544 0.09801729 0.1159442 0.439732011 0.09583301 0.3867446
## [5,] 0.80093894 0.95966613 0.2244408 0.786738058 0.08349419 0.1800014
## [6,] 0.55552623 0.07420703 0.6276363 0.658580460 0.65422058 0.3326875
## [7,] 0.05153204 0.73587936 0.2236765 0.508881353 0.75025979 0.3426896
##          [,97]     [,98]     [,99]    [,100]    [,101]     [,102]
## [1,] 0.8611910 0.6736368 0.4326152 0.2170376 0.8633164 0.69443390
## [2,] 0.9187846 0.5980011 0.3930895 0.2060755 0.5048547 0.08189908
## [3,] 0.6827469 0.7497609 0.1822099 0.1689044 0.5163440 0.84092669
## [4,] 0.7988146 0.6830547 0.3493793 0.4165844 0.6547881 0.75559309
## [5,] 0.4336934 0.2784893 0.6886105 0.8253391 0.5549053 0.71880810
## [6,] 0.6681592 0.9408279 0.9593975 0.2022780 0.2852109 0.13420901
## [7,] 0.5199679 0.8131354 0.4990536 0.8477770 0.7018382 0.73622748
##          [,103]     [,104]     [,105]     [,106]      [,107]    [,108]
## [1,] 0.00929070 0.54129864 0.64630479 0.48141790 0.230371349 0.4494672
## [2,] 0.10989637 0.08801390 0.49462861 0.89616316 0.023946382 0.5312700
## [3,] 0.58412134 0.01628532 0.73303190 0.98090826 0.274931159 0.9823788
## [4,] 0.99127731 0.97470720 0.51709077 0.52971328 0.643916116 0.4268339
## [5,] 0.22236965 0.74921922 0.30811034 0.45900502 0.013285816 0.7863120
## [6,] 0.07031604 0.58687219 0.13101585 0.45348288 0.006495902 0.5837720
## [7,] 0.88515810 0.17337383 0.07318512 0.07882848 0.715869990 0.7995420
##         [,109]    [,110]    [,111]    [,112]    [,113]    [,114]    [,115]
## [1,] 0.4328599 0.9682626 0.9190493 0.5731611 0.6708728 0.5890649 0.5927029
## [2,] 0.3604686 0.7107097 0.9059142 0.9113703 0.7722951 0.2710790 0.2600709
## [3,] 0.4854962 0.1713843 0.6473648 0.1438621 0.5863838 0.7866704 0.9020901
## [4,] 0.2039096 0.2145008 0.2937146 0.5199002 0.4266611 0.4742475 0.7831325
## [5,] 0.7348667 0.8386832 0.3251621 0.4893448 0.3044134 0.2464204 0.3789222
## [6,] 0.4594524 0.0218795 0.3411521 0.3540894 0.6635306 0.9151334 0.4272254
## [7,] 0.8220139 0.9012593 0.4988079 0.3277770 0.4764548 0.1544275 0.1999663
##         [,116]
## [1,] 0.1134410
## [2,] 0.6692852
## [3,] 0.7328469
## [4,] 0.9702275
```

```
## [5,] 0.9527144
## [6,] 0.2504574
## [7,] 0.3435233
```

Conclusion

- Human brain activities varied from individual to individual.

- fMRI data contains 4 Scans, each scan on each subject gives very similar result.

- In our random effect model we combined Scans and Regions and taking subjects as random factor, the results of adjusted sum of square are kind of large and P-value is high, so the drift coefficient from Low-Rank Multivariate General Linear Model can be looked as a constant in this case. Which means subjects' heartbeat, motion, respiration and machine noise are not significant.

Future Work

This analysis highly based on Low-Rank Multivariate General Linear Model, after splitting time series data into multiple windows the dataset becomes very large. The calculation almost did on RIT server and it's still difficult to do whole 820 subjects so we just subset 25 subjects from fMRI data to implement algorithm.

The main objective of this thesis is to provide a new perspective on statistical research on human brain activity, in future, there is still large space to explore more effective calculation method including reduce computing time and memory.

Reference

[1] Zhang, T., Pham, M., Sun, J., Yan, G., Li, H., Sun, Y., Gonzalez, M.Z. and Coan, J.A., 2018. A low-rank multivariate general linear model for multi-subject fMRI data and a non-convex optimization algorithm for brain response comparison. NeuroImage, 173, pp.580-591.

[2] R. Casanova, S. Ryali, J. Serences, L. Yang, R. Kraft, P. Laurienti, J. Maldjian, The impact of temporal regularization on estimates of the BOLD hemodynamic response function: a comparative analysis, NeuroImage 40 (4) (2008) 1606–1618.

[3] K. Friston, P. Jezzard, R. Turner, Analysis of functional MRI time-series, Human Brain Mapping 1 (1994) 153–171.

[5] Finn, E.S., Shen, X., Scheinost, D., Rosenberg, M.D., Huang, J., Chun, M.M., Papademetris, X. and Constable, R.T., 2015. Functional connectome fingerprinting: identifying individuals using patterns of brain connectivity. Nature neuroscience, 18(11), p.1664.

[6] Friston, K.J., Fletcher, P., Josephs, O., Holmes, A.N.D.R.E.W., Rugg, M.D. and Turner, R., 1998. Event-related fMRI: characterizing differential responses. Neuroimage, 7(1), pp.30-40.

[7] Laird, N.M. and Ware, J.H., 1982. Random-effects models for longitudinal data. Biometrics, 38(4), pp.963-974.

[8] Friston, K.J., Holmes, A.P., Poline, J.B., Grasby, P.J., Williams, S.C.R., Frackowiak, R.S. and Turner, R., 1995. Analysis of fMRI time-series revisited. Neuroimage, 2(1), pp.45-53.

[9] Maldjian, J.A., Laurienti, P.J., Kraft, R.A. and Burdette, J.H., 2003. An automated method for neuroanatomic and cytoarchitectonic atlas-based interrogation of fMRI data sets. Neuroimage, 19(3), pp.1233-1239.

[10] T. Zhang, F. Li, L. Beckes, J. Coan, A semi-parametric model of the hemodynamicresponse for multi-subject fmri data, NeuroImage 75 (2013) 136–145.

Appendix

MATLAB and R Code Comparison

The following results are output of u is for 1 iteration:

Matlab output for u	R output for u
U_cur = 0.2027 -0.1614 0.1292 -0.1035 -0.0945 0.1122 0.1162 -0.1693 -0.1819 0.1013 -0.2162 0.2280 -0.0865 0.1199 0.6477 -0.4682 -0.1293 0.0270 -0.2669 0.0832 -0.1780 0.1018 0.0667 0.0232 0.3397 -0.0925 -0.1737 0.0474 -0.0209 -0.0519 0.0456 -0.0822 -0.1727 0.1469 0.1842 -0.1771 -0.4209 0.3613 0.5696 -0.3948 0.0024 -0.0001 -0.1754 0.1028 -0.5029 0.3816 0.5990 -0.5936 -0.3325 0.2862 0.2452 -0.1051 0.0324 0.0185 -0.1006 0.0887 0.3117 -0.3592 -0.9124 0.5870 0.2125 -0.1112 0.4214 -0.2174 0.7555 -0.4730	> output1$u [,1] [,2] [1,] 0.203341817 -0.1603714444 [2,] 0.129878581 -0.1024625540 [3,] -0.093767810 0.1132575152 [4,] 0.116848387 -0.1682639006 [5,] -0.181278204 0.1022934687 [6,] -0.215603994 0.2289555378 [7,] -0.085834535 0.1207780760 [8,] 0.648265265 -0.4673481096 [9,] -0.128801096 0.0278041850 [10,] -0.266471992 0.0840054686 [11,] -0.177590385 0.1024378674 [12,] 0.067041063 0.0238734569 [13,] 0.339957389 -0.0920450738 [14,] -0.173467889 0.0477693578 [15,] -0.020783490 -0.0517475293 [16,] 0.045521017 -0.0823042488 [17,] -0.172825963 0.1466703374 [18,] 0.183985537 -0.1776061024 [19,] -0.421252842 0.3606691721 [20,] 0.569233281 -0.3955750200 [21,] 0.001795273 -0.0009590812 [22,] -0.175988424 0.1019005224 [23,] -0.503514959 0.3806468214 [24,] 0.598323115 -0.5946203614 [25,] -0.333170933 0.2852225414 [26,] 0.244580774 -0.1060312112 [27,] 0.031727460 0.0175890244 [28,] -0.101250756 0.0878163395 [29,] 0.311124082 -0.3599776955 [30,] -0.912967765 0.5862383923 [31,] 0.211983938 -0.1119938805 [32,] 0.420833907 -0.2181584295 [33,] 0.754918295 -0.4737421277

The following is for 20 iteration, the final output:

Matlab output for u	R output for u
U_cur =	output1$u [,1] [,2] [1,] 0.098877112 -0.1338910771

0.1058 -0.1341 -0.0228 -0.0216 0.0212 0.1834 0.0299 -0.1329 -0.0429 0.0003 -0.0214 0.0991 -0.0515 -0.0127 0.2287 -0.0769 -0.0687 0.0060 -0.0617 0.0170 -0.0469 0.0634 0.0065 -0.0832 0.1635 0.1278 -0.1241 -0.1507 0.0244 0.0665 0.0146 -0.0497 -0.0462 0.0961 0.0836 -0.1092 -0.1987 0.1241 0.2327 -0.0769 -0.0558 -0.0462 -0.0100 0.0331 -0.1735 0.1565 0.1983 -0.2987 -0.1552 0.2175 0.1005 -0.0434 0.0049 -0.0453 -0.0718 0.0830 0.2006 -0.1167 -0.4194 0.0750 0.2576 0.0204 -0.1459 -0.0709 0.4124 0.0054	[2,] -0.029603465 -0.0214156177 [3,] 0.014206085 0.1838666888 [4,] 0.023727835 -0.1327081979 [5,] -0.048314953 0.0005033566 [6,] -0.026133216 0.0994259350 [7,] -0.055297649 -0.0124743000 [8,] 0.225177646 -0.0765368685 [9,] -0.071350978 0.0062533462 [10,] -0.063887518 0.0173174103 [11,] -0.048764219 0.0637671384 [12,] 0.005198155 -0.0829973941 [13,] 0.162116177 0.1282373924 [14,] -0.124193535 -0.1507619999 [15,] 0.024254045 0.0667092947 [16,] 0.014875417 -0.0497178374 [17,] -0.045678244 0.0961935025 [18,] 0.084528594 -0.1092481533 [19,] -0.197314448 0.1239920863 [20,] 0.233966563 -0.0769609162 [21,] -0.053709113 -0.0464550131 [22,] -0.007718939 0.0328530910 [23,] -0.170810051 0.1561283682 [24,] 0.201170846 -0.2992735443 [25,] -0.152132654 0.2170755929 [26,] 0.103629540 -0.0438629727 [27,] 0.008447988 -0.0457304613 [28,] -0.068025518 0.0827281304 [29,] 0.204544726 -0.1167862920 [30,] -0.414344031 0.0747376876 [31,] 0.261938477 0.0206689067 [32,] -0.140965590 -0.0709760237 [33,] 0.416549699 0.0057666369

Objective function values at the end of 20 iterations

Matlab Code	R Code
F_cur = 1.4266e+03 F_cur = 1.3654e+03 F_cur = 1.3569e+03	[1] 1 [1] 1426.617 [1] 2 [1] 1365.408 [1] 3 [1] 1356.86 [1] 4 [1] 1353.557 [1] 5 [1] 1351.575 [1] 6 [1] 1350.143 [1] 7 [1] 1349.034 [1] 8

F_cur = 1.3536e+03 F_cur = 1.3516e+03 F_cur = 1.3501e+03 F_cur = 1.3490e+03 F_cur = 1.3481e+03 F_cur = 1.3474e+03 F_cur = 1.3468e+03 F_cur = 1.3463e+03 F_cur = 1.3459e+03 F_cur = 1.3455e+03 F_cur = 1.3452e+03 F_cur = 1.3449e+03 F_cur = 1.3446e+03	[1] 1348.147 [1] 9 [1] 1347.421 [1] 10 [1] 1346.818 [1] 11 [1] 1346.31 [1] 12 [1] 1345.877 [1] 13 [1] 1345.504 [1] 14 [1] 1345.179 [1] 15 [1] 1344.894 [1] 16 [1] 1344.643 [1] 17 [1] 1344.419 [1] 18 [1] 1344.219 [1] 19 [1] 1344.038 [1] 20 [1] 1343.875

F_cur = 1.3444e+03 F_cur = 1.3442e+03 F_cur = 1.3440e+03 F_cur = 1.3439e+03	

R code

Read in data

```
# lr_lsq

library('R.matlab')

## R.matlab v3.6.2 (2018-09-26) successfully loaded. See ?R.matlab for help.

##
## Attaching package: 'R.matlab'

## The following objects are masked from 'package:base':
##
##     getOption, isOpen

library('matrixcalc')
setwd('C:\\Users\\yancx\\Desktop\\Thesis\\Peter fMRI data\\')
basis <- readMat("basis.mat")
load("Scans.arr")
```

```r
Y <- Scans.arr[,,1,]

# d is random array with dimension (7,116,820)
# u is random matrix with dimension (33,p) where p=2 in this case
# v is random array with dimension (p,116,820) where p=2

p <- 2
v <- array(data = runif(p*116*820), dim = c(p,116,820))

set.seed(10)
d <- array(data = runif(7*116*820), dim = c(7,116,820))
u <- matrix(data = runif(33*p), nrow = 33, ncol = p)


D <- basis$D
B <- basis$B
R <- basis$R



###################################################
#       Obj.val
###################################################
obj_val <- function(Y,D,d, B,u,v, lambda, R){
  n = dim(Y)[3]
  val=0;
  for (i in 1:n) {
    val = val + frobenius.norm(Y[,,i]-D%*%d[,,i]-B%*%u%*%v[,,i])^2
  }
  frob.val = val/n +lambda*matrix.trace(t(u)%*%(R%*%u))
  frob.val
}

###################################################

# step1: fix U and d, solve v(v is unknown here)
# step2: fix v(at solution of previous step), u and d are unknown here

library("Matrix") # sparse matrix
library("pcg")
library("matlib")

##
## Attaching package: 'matlib'

## The following object is masked from 'package:matrixcalc':
##
##     vec
```

```r
lr.func.stp12 <- function(Y, D, d, B, u, v, lambda, R){
  # [T,L]=size(X)
  mat.TL <- matrix(c(dim(B)[1], dim(B)[2]), nrow = 1)
  Tn = mat.TL[1]
  Ln = mat.TL[2]
  P = dim(u)[2]
  Iden_T = Matrix(diag(Tn), sparse = T)
  Iden_T <- as.matrix(Iden_T)
  H2 = inv(t(D)%*%D)%*%t(D)
  H = D%*%H2
  ITH = Iden_T - H
  totalR = kronecker(diag(P), R)
  F_cur=obj_val(Y,D,d,B,u,v,lambda,R)
  maxIter=20
  N = dim(Y)[3] # Y_train
  ########################################
  # May 6
  # the following for loop is for rest of function
  for (iter in 1:maxIter)
  {
    print(iter)
    # U and d are fixed, find V
    W = B%*%u
    Y_new <- array(data = NA, dim = c(dim(Y)[1],dim(Y)[2],dim(Y)[3]))
    for (i in 1:dim(Y)[3]) {
      Y_new[,,i] = Y[,,i]-D%*%d[,,i]
    }
    Y_new
    Q = t(W)%*%W
    inv_Q = inv(Q)
    for (i in 1:dim(Y)[3]) {
      v[,,i] = inv_Q%*%(t(W)%*%Y_new[,,i])
    }
    v
    #}
    ################################# iter for loop should not end here

    # When V are fixed,
    # min_{U} 1/n ||Y -Dd - X*U*V||^2 + lambda U' R U
    # A bit more complicated but fairly easy.
    # Calculate sum of V_iV_i^T and XY_iV_i^T
    totalW = matrix(0, nrow = Ln*P, ncol = Ln*P)
    f = matrix(0, nrow = Ln*P, ncol = 1)
    for (n in 1:N) {
      temp = t(B)%*%ITH
      VVT = v[,,n]%*%t(v[,,n])
      XTX = temp%*%B
      totalW = totalW+ kronecker(VVT,XTX)
      f = f + matrix(temp%*%Y[,,n]%*%t(v[,,n]), nrow = P*Ln, ncol = 1)
    }
```

```
    Q = totalW/N + lambda*totalR
    f=f/N
    # pcg part
    # [sol,flag]=pcg(Q,f,1e-5,2000)
    sol <- pcg(Q, f, maxiter = 5000, tol = 1e-06)
    u = matrix(sol, nrow = Ln, ncol = P)
    # in Matlab d_cur=cell(N,1)
    # R has to set up dimension
    # For previous, d is random array with dimension (7,116,820)
    d = array(data = sol, dim = c(7,116,820))
    for (n in 1:N) {
      d[,,n] = H2%*%Y[,,n]
      d[,,n] = d[,,n] - H2%*%B%*%u%*%v[,,n]
    }
    F_new=obj_val(Y,D,d,B,u,v,lambda,R)
    print(F_new)
    if (F_new - F_cur > 1e-5)
    {F_cur = F_new
    break}
    F_cur =  F_new
  }
  list(d=d,v=v,u=u)
}

# output1 <- lr.func.stp12(Y, D, d, B, u, v, lambda=100, R)

# matplot(B%*%output1$u, type = "l", ylab = "B*u")

# d is random array with dimension (7,116,820)
# out.d is matrix with dimension (7,116*820)
# out.d <- matrix(data = output1$d, ncol = 116*820)


# matplot(B,type = "l")
# matplot(D,type = "l")
```

Predict Curve

```
library('R.matlab')
library('matrixcalc')
lr_lsq <- function(Y_test, D, B , U_cur){
  N <- dim(Y_test)[3]
  P <- ncol(U_cur)
  W <- B%*%U_cur
  A <- cbind(W,D)
  inv_A = solve(t(A)%*%A)
  # in Matlab d_test=cell(N,1), V_test=cell(N,1)
  d_test <- array(data = 0, dim = c(7,116,N))
  V_test <- array(data = 0, dim = c(2,116,N))
  for (n in 1:N) {
```

```
    sol = inv_A%*%(t(A)%*%Y_test[,,n])
    V_test[,,n] = sol[1:P,]
    D_test[,,n] = sol[-c(1:P),]
  }
  for (i in 1:dim(Y_test)[3]) {
    temp = temp + frobenius.norm(Y_test[,,i]-D%*%D_test[,,i])^2
  }
  error = temp/N
}
```

Generate Basis

```
library(fda)

## Loading required package: splines

##
## Attaching package: 'fda'

## The following object is masked from 'package:matlib':
##
##     Eigen

## The following object is masked from 'package:graphics':
##
##     matplot

time_length=30
Breaks=0:time_length
ttime <- 1:time_length

B<-bsplineS(ttime,breaks=Breaks,norder=4)
r <- 7

D<-matrix(0,nrow=time_length,ncol=r)

for( r in 1:7){
  D[,r] <- sqrt(2/time_length)*cos(r*pi/time_length*(1:time_length))
}
```

Reformat data

```
# Y_25 <- Scans.arr[,,,1:25]


reformat4scan <- function(Y){
  h <- 30
  k <- dim(Y)[2]
  t <- dim(Y)[1]-h+1
  sn <- dim(Y)[3]
  subn <- dim(Y)[4]
  fin.out.arr <- array(data = 0, dim = c(h,t*k,sn,subn))
```

```r
  for (index in 1:sn) {
    # aa1 is for one patient MA time series
    # aa1 will eventually convert into matrix then store in an array aa2
    # finally it will convert into matrix w/ dim(h,t*k)
    aa1 <- array(data = 0,dim = c(h,k,t))
    # aa2 is for all 820 patients
    # also the final output array w/ dim 30*(1171*116)*820
    aa2 <- array(data = 0,dim = c(h,t*k,subn))
    # extract one scan
    Y1s <- Y[,,index,]
    for (ind in 1:subn) {
      for (i in 1:t) {
        aa1[,,i] <- Y1s[i:(i+29),,ind]
        aa2[,,ind] <- matrix(aa1,ncol = t*k)
      }
    }
    fin.out.arr[,,index,] <- aa2
  }
  fin.out.arr <- array(data = fin.out.arr, dim = c(h,t*k,sn*subn))
  fin.out.arr
}

# Scan30_25.arr <- reformat4scan(Y_25)
```

Random Effect

```r
# output.v
setwd('C:\\Users\\yancx\\Desktop\\Thesis\\Peter fMRI data\\')
load("output.v")

class(output.v)

## [1] "array"

dim(output.v)

## [1]      2 135836    100

extract.116.v <- function(Y){
  out.arr <- array(data = 0, dim = c(100,1,2,116))
  for (j in 1:116) {
    for (i in 1:2) {
      out.arr[,,i,j] <- Y[i,(1171*(j-1)+1),]
    }
  }
  out.arr
}
out.array <- extract.116.v(output.v)


out.df <- data.frame(data=out.array)
```

```
person <- paste(rep(1:25, each=4), sep = "")
out.df <- cbind(person, out.df)


colnames(out.df) <- c("person", 1:232)


adjsq.func.v <- function(data){
  out.var <- matrix(data = 0, nrow = 232)
  for (i in 2:233) {
    out.var[(i-1),] <- summary(lm(out.df[,i] ~ person, data = out.df))$adj.r.
squared
  }
  out.var <- matrix(data = out.var, nrow = 2, ncol = 116)
  out.var
}
adjsq.out.v <- adjsq.func.v(out.df)

adjsq.out.v [adjsq.out.v  < 0] <- 0
adjsq.out.v

##      [,1]       [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
## [1,]    0 0.05899422    0    0    0    0    0    0    0     0     0
## [2,]    0 0.04772655    0    0    0    0    0    0    0     0     0
##            [,12] [,13]      [,14]      [,15]     [,16] [,17]     [,18]
## [1,] 0.05256526     0 0.04059682 0.05785718 0.1231110     0 0.2663078
## [2,] 0.05497705     0 0.03777781 0.03891055 0.1249247     0 0.2712387
##      [,19] [,20] [,21]      [,22] [,23] [,24]     [,25]     [,26]
## [1,]     0     0     0 0.09141593     0     0 0.1709059 0.1174849
## [2,]     0     0     0 0.09189232     0     0 0.1809692 0.1196714
##          [,27]      [,28] [,29] [,30] [,31] [,32] [,33]     [,34] [,35]
## [1,] 0.1229084 0.06691678     0     0     0     0     0 0.2250841     0
## [2,] 0.1342421 0.09397375     0     0     0     0     0 0.2387908     0
##           [,36]      [,37]     [,38]      [,39] [,40]     [,41]     [,42]
## [1,] 0.09229229 0.09478323 0.1158448 0.07680260     0 0.1217537 0.1027655
## [2,] 0.10449860 0.04904605 0.1189993 0.06608807     0 0.1036391 0.1005818
##      [,43] [,44] [,45]      [,46] [,47] [,48]       [,49]      [,50] [,51]
## [1,]     0     0     0 0.07344790     0     0 0.008577619 0.04301075     0
## [2,]     0     0     0 0.08037808     0     0 0.032912386 0.04705502     0
##      [,52] [,53] [,54]      [,55]     [,56] [,57]      [,58]     [,59]
## [1,]     0     0     0 0.04374575 0.1022766     0 0.00395331 0.1760955
## [2,]     0     0     0 0.05099243 0.1155410     0 0.00000000 0.1589436
##            [,60]      [,61]      [,62]      [,63] [,64]       [,65] [,66]
## [1,] 0.000000000 0.08387791 0.02893927 0.04747483     0 0.001396705     0
## [2,] 0.008005893 0.10324925 0.04185982 0.03415152     0 0.000000000     0
##      [,67] [,68]      [,69]      [,70] [,71] [,72] [,73]      [,74]
## [1,]     0     0 0.02303322 0.10888311     0     0     0 0.07791799
## [2,]     0     0 0.02542621 0.05017896     0     0     0 0.07866825
##           [,75]      [,76]      [,77] [,78]      [,79] [,80] [,81]
## [1,] 0.00000000 0.01232109 0.08229854     0 0.03797791     0     0
```

```
## [2,] 0.02813087 0.03018651 0.05077243     0 0.06338682     0     0
##            [,82] [,83] [,84] [,85] [,86] [,87]      [,88]     [,89]
## [1,] 0.03571153     0     0     0     0     0 0.09466044 0.1129366
## [2,] 0.04698150     0     0     0     0     0 0.09104430 0.1236580
##            [,90] [,91] [,92]      [,93]      [,94]     [,95] [,96]
## [1,] 0.08891020     0     0 0.06596561 0.09973603 0.1376509     0
## [2,] 0.07844016     0     0 0.08884245 0.10185582 0.1277669     0
##            [,97]      [,98]      [,99] [,100] [,101]     [,102]
## [1,] 0.04797241 0.05081409 0.08865547      0      0 0.09453402
## [2,] 0.06232223 0.05811943 0.09942280      0      0 0.09971610
##             [,103]      [,104] [,105] [,106] [,107] [,108] [,109]    [,110]
## [1,] 0.0000000000 0.014953397      0      0      0      0      0 0.1125626
## [2,] 0.0001183567 0.008788233      0      0      0      0      0 0.1024753
##      [,111]    [,112]     [,113] [,114]    [,115] [,116]
## [1,]      0 0.1651118 0.06657843      0 0.2537711      0
## [2,]      0 0.1196668 0.06156965      0 0.2503594      0

p.value.func <- function(data){
  out.pvalue <- matrix(data = 0, nrow = 232)
  for (i in 2:233) {
    out.pvalue[(i-1),] <- anova(lm(out.df[,i] ~ person, data = out.df))$'Pr(>
F)'[1]
  }
  out.pvalue <- matrix(data = out.pvalue, nrow = 2, ncol = 116)
  out.pvalue
}
p.value.v <- p.value.func(out.df)
p.value.v

##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
## [1,] 0.6676980 0.2236668 0.8499273 0.8149259 0.7466525 0.8526446 0.6687835
## [2,] 0.6459225 0.2645588 0.6358224 0.8160406 0.6286101 0.9018339 0.5447495
##           [,8]      [,9]     [,10]     [,11]     [,12]     [,13]     [,14]
## [1,] 0.8042677 0.8834027 0.8837883 0.8736588 0.2464859 0.5003875 0.2925503
## [2,] 0.8417301 0.9297368 0.8993601 0.8574925 0.2377643 0.6452338 0.3040496
##          [,15]      [,16]     [,17]       [,18]     [,19]     [,20]
## [1,] 0.2276022 0.06978194 0.6543363 0.001403947 0.8280825 0.9273250
## [2,] 0.2994003 0.06717094 0.7107979 0.001186985 0.8521785 0.9503277
##          [,21]     [,22]     [,23]     [,24]      [,25]      [,26]
## [1,] 0.8875990 0.1297612 0.6532619 0.8837550 0.02321293 0.07840181
## [2,] 0.9242247 0.1286399 0.7776898 0.8517602 0.01794746 0.07495654
##           [,27]     [,28]     [,29]     [,30]     [,31]     [,32]
## [1,] 0.07007859 0.1974566 0.4859939 0.8501569 0.9047278 0.7209244
## [2,] 0.05497244 0.1238246 0.5812980 0.8528644 0.8445208 0.7322964
##          [,33]       [,34]     [,35]    [,36]     [,37]      [,38]
## [1,] 0.5159547 0.005223830 0.9607078 0.127704 0.1219887 0.08106766
## [2,] 0.5460359 0.003434518 0.9477822 0.101515 0.2595551 0.07600260
##          [,39]     [,40]      [,41]     [,42]     [,43]     [,44]
## [1,] 0.1677326 0.7562823 0.07178863 0.1049601 0.4843698 0.9211074
## [2,] 0.2000987 0.7396352 0.10321253 0.1094272 0.5083220 0.8512261
```

```
##           [,45]     [,46]     [,47]    [,48]     [,49]     [,50]     [,51]
## [1,] 0.6198414 0.1774490 0.6023524 0.596603 0.4354226 0.2828955 0.4952916
## [2,] 0.5663802 0.1577922 0.6372934 0.692554 0.3244461 0.2671269 0.5352031
##           [,52]     [,53]     [,54]     [,55]      [,56]     [,57]
## [1,] 0.7093333 0.9622590 0.5603764 0.2799914 0.10594791 0.5374747
## [2,] 0.7022282 0.9544999 0.6327866 0.2522769 0.08156922 0.5087715
##           [,58]      [,59]     [,60]     [,61]     [,62]     [,63]
## [1,] 0.4578297 0.02035158 0.8372634 0.1484737 0.3415954 0.2655198
## [2,] 0.5299946 0.03115330 0.4381755 0.1039895 0.2874764 0.3191870
##           [,64]     [,65]     [,66]    [,67]     [,68]     [,69]     [,70]
## [1,] 0.6704383 0.4703422 0.5103445 0.928555 0.5570350 0.3678525 0.0931875
## [2,] 0.6188021 0.5464804 0.5152975 0.928246 0.6123286 0.3571079 0.2553037
##           [,71]     [,72]     [,73]     [,74]     [,75]     [,76]     [,77]
## [1,] 0.9106303 0.5227067 0.8060774 0.1645858 0.5381231 0.4175320 0.1526287
## [2,] 0.8905947 0.6636065 0.7596478 0.1624925 0.3451367 0.3361657 0.2530934
##           [,78]     [,79]     [,80]     [,81]     [,82]     [,83]     [,84]
## [1,] 0.6078016 0.3032255 0.9887231 0.7241353 0.3126282 0.9839123 0.4922712
## [2,] 0.7013690 0.2088725 0.9548922 0.8359900 0.2674090 0.9810762 0.6444104
##           [,85]     [,86]     [,87]     [,88]      [,89]     [,90]
## [1,] 0.5706196 0.4836476 0.6482661 0.1222658 0.08597007 0.1357781
## [2,] 0.5252147 0.4912883 0.6361806 0.1306409 0.06898597 0.1631269
##           [,91]     [,92]     [,93]     [,94]      [,95]     [,96]
## [1,] 0.7344377 0.9780551 0.2004911 0.1111956 0.05098980 0.7990916
## [2,] 0.6425919 0.9780086 0.1359436 0.1068037 0.06323818 0.9482011
##           [,97]     [,98]    [,99]    [,100]    [,101]    [,102]    [,103]
## [1,] 0.2636223 0.2529386 0.136401 0.6895028 0.8263361 0.1225517 0.5734368
## [2,] 0.2123982 0.2266907 0.111856 0.5887478 0.8874761 0.1112375 0.4766274
##          [,104]    [,105]    [,106]    [,107]    [,108]    [,109]
## [1,] 0.4051035 0.9580460 0.7960052 0.6427722 0.4842246 0.8935441
## [2,] 0.4344097 0.9490778 0.7803167 0.5500522 0.5351016 0.9295218
##           [,110]    [,111]     [,112]    [,113]    [,114]      [,115]
## [1,] 0.08661704 0.8681257 0.02681023 0.1985325 0.9130464 0.002129056
## [2,] 0.10554552 0.9758605 0.07496362 0.2149137 0.8794311 0.002378397
##          [,116]
## [1,] 0.6654877
## [2,] 0.6426594

# output.d

setwd('C:\\Users\\yancx\\Desktop\\Thesis\\Peter fMRI data\\')
load("output.d")

class(output.d)

## [1] "array"

dim(output.d)

## [1]      7 135836    100
```

```
Y1 <- output.d[1,1,]
Y1 <- matrix(data = Y1 , ncol =1)
Y2 <- output.d[2,1,]
Y2 <- matrix(data = Y2 , ncol =1)
Y3 <- output.d[3,1,]
Y3 <- matrix(data = Y3 , ncol =1)
Y4 <- output.d[4,1,]
Y4 <- matrix(data = Y4 , ncol =1)
Y5 <- output.d[5,1,]
Y5 <- matrix(data = Y5 , ncol =1)
Y6 <- output.d[6,1,]
Y6 <- matrix(data = Y6 , ncol =1)
Y7 <- output.d[7,1,]
Y7 <- matrix(data = Y7 , ncol =1)


extract.116.test <- function(Y){
  out.arr <- array(data = 0, dim = c(100,1,7,116))
  for (j in 1:116) {
    for (i in 1:7) {
      out.arr[,,i,j] <- Y[i,(1171*(j-1)+1),]
    }
  }
  out.arr
}
out.array <- extract.116.test(output.d)


out.df <- data.frame(data=out.array)
person <- paste(rep(1:25, each=4), sep = "")
out.df <- cbind(person, out.df)


colnames(out.df) <- c("person", 1:812)


adjsq.func.d <- function(data){
  out.var <- matrix(data = 0, nrow = 812)
  for (i in 2:813) {
  out.var[(i-1),] <- summary(lm(out.df[,i] ~ person, data = out.df))$adj.r.sq
uared
  }
  out.var <- matrix(data = out.var, nrow = 7, ncol = 116)
  out.var
}
adjsq.out.d <- adjsq.func.d(out.df)

adjsq.out.d [adjsq.out.d  < 0] <- 0
adjsq.out.d
```

```
##              [,1]        [,2]       [,3] [,4]       [,5]        [,6]
## [1,] 0.002864811 0.051109726 0.00000000    0 0.08395585 0.000000000
## [2,] 0.000000000 0.006932619 0.06460687    0 0.00000000 0.003343756
## [3,] 0.000000000 0.000000000 0.00000000    0 0.00000000 0.094422974
## [4,] 0.039023094 0.041259533 0.01321886    0 0.03330971 0.000000000
## [5,] 0.042609648 0.000000000 0.14117112    0 0.01824489 0.000000000
## [6,] 0.066406178 0.000000000 0.07525265    0 0.00000000 0.004734788
## [7,] 0.000000000 0.000000000 0.03503993    0 0.00000000 0.037442966
##             [,7]        [,8]       [,9]      [,10]      [,11]      [,12]
## [1,] 0.06029930 0.000000000 0.00000000 0.03838504 0.00000000 0.04888318
## [2,] 0.09565393 0.000000000 0.00000000 0.00000000 0.07114471 0.04454667
## [3,] 0.01115500 0.035092709 0.00000000 0.00000000 0.00000000 0.00000000
## [4,] 0.00000000 0.030354570 0.00000000 0.00000000 0.08755001 0.09586695
## [5,] 0.00000000 0.000000000 0.00000000 0.00000000 0.04983520 0.01061622
## [6,] 0.00000000 0.002793398 0.05255703 0.00000000 0.00000000 0.00000000
## [7,] 0.03646037 0.000000000 0.00000000 0.00000000 0.04225066 0.07786538
##           [,13]      [,14]      [,15]      [,16]      [,17]      [,18]
## [1,] 0.03381266 0.00000000 0.00000000 0.00000000 0.00000000 0.03099220
## [2,] 0.00000000 0.00000000 0.00000000 0.04780182 0.00000000 0.00000000
## [3,] 0.08924529 0.01477926 0.00000000 0.00000000 0.03903037 0.07165699
## [4,] 0.00000000 0.00000000 0.05308367 0.05262786 0.01037373 0.00000000
## [5,] 0.01977072 0.00000000 0.00000000 0.00000000 0.02203492 0.04548164
## [6,] 0.17170676 0.19629084 0.00000000 0.00000000 0.00000000 0.06033811
## [7,] 0.00000000 0.00000000 0.00000000 0.00000000 0.05955933 0.00000000
##           [,19]      [,20]      [,21]     [,22]      [,23]      [,24]
## [1,] 0.02602517 0.00000000 0.00000000 0.0000000 0.04721299 0.00000000
## [2,] 0.10574114 0.10580253 0.01523745 0.0000000 0.00000000 0.00000000
## [3,] 0.00000000 0.00000000 0.00000000 0.0000000 0.02595492 0.00000000
## [4,] 0.06832325 0.00000000 0.00000000 0.0000000 0.00000000 0.09907724
## [5,] 0.06014204 0.00000000 0.00000000 0.0256542 0.00000000 0.01942983
## [6,] 0.00000000 0.08615232 0.14158095 0.0000000 0.16180978 0.00000000
## [7,] 0.00000000 0.00000000 0.03888415 0.0000000 0.00000000 0.00000000
##           [,25]       [,26]       [,27]       [,28]      [,29]     [,30]
## [1,] 0.00000000 0.000000000 0.000000000 0.000000000 0.00000000 0.0000000
## [2,] 0.00000000 0.000000000 0.000000000 0.041893592 0.00000000 0.0000000
## [3,] 0.00000000 0.056764486 0.004009404 0.000000000 0.00000000 0.1739786
## [4,] 0.02819383 0.019394993 0.038512353 0.051900352 0.07244422 0.0000000
## [5,] 0.00000000 0.001191753 0.000000000 0.007060772 0.12564854 0.1546246
## [6,] 0.00000000 0.000000000 0.000000000 0.045972190 0.20091704 0.0000000
## [7,] 0.00000000 0.000000000 0.016653902 0.047365398 0.00000000 0.0000000
##           [,31]       [,32]      [,33]       [,34]       [,35]       [,36]
## [1,] 0.12801584 0.000000000 0.00000000 0.002249931 0.000000000 0.000000000
## [2,] 0.10011492 0.074054560 0.02386692 0.006246562 0.003546431 0.014726505
## [3,] 0.01154893 0.000000000 0.00000000 0.004888754 0.000000000 0.206164321
## [4,] 0.00000000 0.071944003 0.00000000 0.019695595 0.000000000 0.000000000
## [5,] 0.00000000 0.025624316 0.00000000 0.126962377 0.000000000 0.014127579
## [6,] 0.00000000 0.000000000 0.00000000 0.000000000 0.083960188 0.021755114
## [7,] 0.00000000 0.009471564 0.05202625 0.000000000 0.183750604 0.008544853
##              [,37]       [,38]      [,39]      [,40]       [,41]      [,42]
## [1,] 0.0860245912 0.000000000 0.00000000 0.15741932 0.082430839 0.00000000
```

```
## [2,] 0.0000000000 0.042445182 0.00000000 0.00000000 0.000000000 0.11094985
## [3,] 0.0008639233 0.000000000 0.19770906 0.07836757 0.000000000 0.09770056
## [4,] 0.0000000000 0.000000000 0.00000000 0.00000000 0.004593663 0.06309648
## [5,] 0.0000000000 0.013320127 0.05619886 0.00000000 0.000000000 0.04538828
## [6,] 0.0000000000 0.004167038 0.00000000 0.00000000 0.038554375 0.00000000
## [7,] 0.0000000000 0.037255221 0.05193283 0.09302802 0.000000000 0.00000000
##            [,43]       [,44]       [,45]       [,46]      [,47]      [,48]
## [1,] 0.008019067 0.039412380 0.003348899 0.037856923 0.00000000 0.08455425
## [2,] 0.000000000 0.000000000 0.077383160 0.039914197 0.00000000 0.03658319
## [3,] 0.000000000 0.091232407 0.000000000 0.055117799 0.00000000 0.00000000
## [4,] 0.000000000 0.122299996 0.000000000 0.058607572 0.05367664 0.00000000
## [5,] 0.025102891 0.073933452 0.121383217 0.008786756 0.00000000 0.00000000
## [6,] 0.044603522 0.003990717 0.008739612 0.061321979 0.10332840 0.00000000
## [7,] 0.000000000 0.021993993 0.005465797 0.000000000 0.15728904 0.00000000
##           [,49]      [,50]      [,51]      [,52]       [,53]       [,54]
## [1,] 0.00000000 0.06980252 0.00000000 0.05980106 0.189347567 0.102261779
## [2,] 0.00000000 0.13534993 0.00000000 0.07414676 0.000000000 0.003815112
## [3,] 0.04816867 0.00000000 0.00000000 0.11373938 0.000000000 0.000000000
## [4,] 0.00000000 0.02009737 0.02597275 0.00000000 0.000000000 0.004248239
## [5,] 0.00000000 0.06920988 0.21891979 0.00000000 0.009581625 0.000000000
## [6,] 0.02911011 0.01366715 0.00000000 0.02208593 0.000000000 0.173953208
## [7,] 0.14809645 0.00000000 0.00000000 0.00000000 0.003839101 0.000000000
##            [,55]       [,56]      [,57]      [,58]      [,59]      [,60]
## [1,] 0.000000000 0.068258225 0.17680080 0.00000000 0.02914794 0.00000000
## [2,] 0.000000000 0.029281896 0.00000000 0.00000000 0.01600500 0.00000000
## [3,] 0.000000000 0.042924423 0.16910370 0.09198879 0.14155941 0.04320880
## [4,] 0.000000000 0.000000000 0.00000000 0.00000000 0.00000000 0.01283095
## [5,] 0.000000000 0.000000000 0.04322730 0.00000000 0.07394329 0.00000000
## [6,] 0.001359142 0.000000000 0.00167188 0.13424147 0.00000000 0.00000000
## [7,] 0.000000000 0.009367426 0.00000000 0.00000000 0.14579679 0.08194326
##           [,61]      [,62]       [,63]      [,64]        [,65]       [,66]
## [1,] 0.00000000 0.00000000 0.000000000 0.06393752 0.0006091796 0.000000000
## [2,] 0.00000000 0.09119744 0.107296725 0.00000000 0.0000000000 0.054364879
## [3,] 0.00000000 0.07643028 0.000000000 0.00000000 0.0882008413 0.007144671
## [4,] 0.00000000 0.00000000 0.068660967 0.00000000 0.0000000000 0.036154022
## [5,] 0.00000000 0.08073361 0.064993789 0.10854150 0.0064555243 0.107405286
## [6,] 0.00000000 0.00000000 0.007199566 0.00000000 0.0080870424 0.000000000
## [7,] 0.03000737 0.11161749 0.000000000 0.03245929 0.0000000000 0.000000000
##           [,67]      [,68]      [,69]      [,70]       [,71]      [,72]
## [1,] 0.00000000 0.00000000 0.00000000 0.00000000 0.100887022 0.00000000
## [2,] 0.00000000 0.16125414 0.03096020 0.01713514 0.000000000 0.00000000
## [3,] 0.00000000 0.00000000 0.04096226 0.00000000 0.004165561 0.01166727
## [4,] 0.00000000 0.00000000 0.06833315 0.06360833 0.048475512 0.00000000
## [5,] 0.09768125 0.01980067 0.00000000 0.00000000 0.123271304 0.01021297
## [6,] 0.16659372 0.10244558 0.11261862 0.00000000 0.000000000 0.11303919
## [7,] 0.00000000 0.00000000 0.09517219 0.10220143 0.000000000 0.00000000
##           [,73]      [,74]      [,75]     [,76]      [,77]    [,78]
## [1,] 0.00000000 0.00000000 0.06835379 0.1207871 0.06945157 0.000000
## [2,] 0.18143603 0.05450577 0.16484677 0.0000000 0.00000000 0.144655
## [3,] 0.02705203 0.00000000 0.00000000 0.0000000 0.00000000 0.000000
```

```
## [4,] 0.00000000 0.00000000 0.00000000 0.0000000 0.00000000 0.000000
## [5,] 0.05524901 0.08793475 0.00000000 0.0000000 0.17383369 0.000000
## [6,] 0.22749010 0.00000000 0.03235672 0.0000000 0.13991216 0.000000
## [7,] 0.13510356 0.00000000 0.04340497 0.1203103 0.01512268 0.000000
##            [,79]      [,80]      [,81]      [,82]       [,83]      [,84]
## [1,] 0.03627100 0.00000000 0.00000000 0.00000000 0.000000000 0.00000000
## [2,] 0.00000000 0.03448429 0.00000000 0.00000000 0.081571452 0.00000000
## [3,] 0.02940218 0.00000000 0.00000000 0.00000000 0.228059635 0.06016105
## [4,] 0.00000000 0.00000000 0.01447423 0.00000000 0.112585299 0.00000000
## [5,] 0.12823724 0.01862143 0.00000000 0.06322224 0.003944239 0.03797253
## [6,] 0.00000000 0.00000000 0.11915822 0.12597163 0.040676400 0.00000000
## [7,] 0.00000000 0.04712014 0.00000000 0.03935247 0.015113309 0.00000000
##            [,85]      [,86]      [,87]      [,88]       [,89]      [,90]
## [1,] 0.00000000 0.00000000 0.00000000 0.14954958 0.018090545 0.00000000
## [2,] 0.00000000 0.00000000 0.03073461 0.00000000 0.000000000 0.04613243
## [3,] 0.02274665 0.00000000 0.00000000 0.00000000 0.000000000 0.05809589
## [4,] 0.04431817 0.00000000 0.01363610 0.00000000 0.000000000 0.01950323
## [5,] 0.22426885 0.00000000 0.00000000 0.05319235 0.000000000 0.08116166
## [6,] 0.01867640 0.09946475 0.13585924 0.05643614 0.000000000 0.10349871
## [7,] 0.00000000 0.00000000 0.07125544 0.01625181 0.001462811 0.02697592
##              [,91]     [,92]      [,93]       [,94]      [,95]      [,96]
## [1,] 0.0494611079 0.0000000 0.08351265 0.214037391 0.00914790 0.00000000
## [2,] 0.0000000000 0.0415978 0.00000000 0.000000000 0.00000000 0.00000000
## [3,] 0.0903181152 0.0000000 0.00000000 0.000000000 0.03158618 0.07238027
## [4,] 0.0008199938 0.1063050 0.09751430 0.007683116 0.10745849 0.01889672
## [5,] 0.0000000000 0.0000000 0.05876960 0.000000000 0.11438855 0.07258774
## [6,] 0.0000000000 0.1201576 0.00000000 0.000000000 0.00000000 0.03099058
## [7,] 0.1371742170 0.0000000 0.05899140 0.000000000 0.00000000 0.02868906
##             [,97]      [,98]       [,99]     [,100]     [,101]     [,102]
## [1,] 0.000000000 0.00000000 0.009161775 0.06093865 0.00000000 0.00000000
## [2,] 0.000000000 0.00000000 0.017525953 0.06423971 0.00000000 0.11534202
## [3,] 0.000000000 0.00000000 0.071850230 0.07639097 0.00000000 0.00000000
## [4,] 0.000000000 0.00000000 0.027167486 0.01252082 0.00000000 0.00000000
## [5,] 0.008937282 0.04412763 0.000000000 0.00000000 0.00000000 0.00000000
## [6,] 0.000000000 0.00000000 0.000000000 0.06540994 0.04242772 0.08955566
## [7,] 0.000000000 0.00000000 0.000000000 0.00000000 0.00000000 0.00000000
##           [,103]     [,104]     [,105]      [,106]     [,107]      [,108]
## [1,] 0.20527598 0.00000000 0.00000000 0.000000000 0.05706452 0.005671669
## [2,] 0.10035639 0.11176257 0.00000000 0.000000000 0.16966485 0.000000000
## [3,] 0.00000000 0.18468242 0.00000000 0.000000000 0.04503687 0.000000000
## [4,] 0.00000000 0.00000000 0.00000000 0.000000000 0.00000000 0.010368458
## [5,] 0.05937177 0.00000000 0.03679596 0.003712446 0.19231352 0.000000000
## [6,] 0.12274669 0.00000000 0.09088649 0.004845472 0.21772917 0.000000000
## [7,] 0.00000000 0.07483901 0.12082691 0.117219498 0.00000000 0.000000000
##            [,109]     [,110]     [,111]     [,112]      [,113]       [,114]
## [1,] 0.009110803 0.00000000 0.00000000 0.00000000 0.000000000 0.0000000000
## [2,] 0.024674370 0.00000000 0.00000000 0.00000000 0.000000000 0.0460288514
## [3,] 0.000000000 0.07552638 0.00000000 0.08566843 0.000000000 0.0000000000
## [4,] 0.064905410 0.06169280 0.04030871 0.00000000 0.010404594 0.0006019767
## [5,] 0.000000000 0.00000000 0.03274447 0.00000000 0.037689555 0.0525831787
```

```
## [6,] 0.003620807 0.17325126 0.02904073 0.02610422 0.000000000 0.0000000000
## [7,] 0.000000000 0.00000000 0.00000000 0.03213279 0.000153408 0.0816243435
##           [,115]     [,116]
## [1,] 0.00000000 0.09867671
## [2,] 0.04890933 0.00000000
## [3,] 0.00000000 0.00000000
## [4,] 0.00000000 0.00000000
## [5,] 0.02059910 0.00000000
## [6,] 0.01028657 0.05148423
## [7,] 0.06612941 0.02849868
```

```
p.value.func <- function(data){
  out.pvalue <- matrix(data = 0, nrow = 812)
  for (i in 2:813) {
    out.pvalue[(i-1),] <- anova(lm(out.df[,i] ~ person, data = out.df))$'Pr(>
F)'[1]
  }
  out.pvalue <- matrix(data = out.pvalue, nrow = 7, ncol = 116)
  out.pvalue
}
p.value.d <- p.value.func(out.df)
p.value.d
```

```
##            [,1]      [,2]        [,3]      [,4]      [,5]      [,6]
## [1,] 0.4631472 0.2518422 0.78795461 0.8584634 0.1482707 0.6665094
## [2,] 0.8641300 0.4433573 0.20487908 0.9312179 0.7912815 0.4608056
## [3,] 0.4928441 0.5864707 0.76394218 0.6891010 0.8462222 0.1228032
## [4,] 0.2989404 0.2898819 0.41327843 0.5642688 0.3227551 0.9099433
## [5,] 0.2844873 0.6703872 0.04712949 0.9652749 0.3897567 0.7014719
## [6,] 0.1990817 0.9882254 0.17217470 0.8254022 0.6828591 0.4540216
## [7,] 0.7171802 0.7976702 0.31544324 0.8302486 0.5599562 0.3054313
##            [,7]      [,8]      [,9]     [,10]     [,11]     [,12]
## [1,] 0.2192033 0.5479120 0.6616093 0.3015525 0.7700879 0.2601696
## [2,] 0.1200366 0.6225575 0.6408167 0.8524885 0.1843397 0.2768463
## [3,] 0.4230789 0.3152216 0.5039007 0.8628914 0.6228148 0.7654918
## [4,] 0.5529301 0.3354372 0.9782893 0.4996553 0.1391286 0.1195626
## [5,] 0.8551583 0.7449486 0.4989386 0.9475570 0.2565893 0.4256499
## [6,] 0.9638294 0.4634965 0.2465160 0.7120539 0.8539728 0.5047709
## [7,] 0.3095048 0.5166885 0.5133843 0.6022832 0.2859162 0.1647333
##           [,13]     [,14]     [,15]     [,16]     [,17]     [,18]
## [1,] 0.32062089 0.5304461 0.8646450 0.5717341 0.8760235 0.3326805
## [2,] 0.95559525 0.9930970 0.6249132 0.2642719 0.5048706 0.6214664
## [3,] 0.13496183 0.4059216 0.5364419 0.7261277 0.2989107 0.1827916
## [4,] 0.96995792 0.8362228 0.2445950 0.2462571 0.4268087 0.9874610
## [5,] 0.38272051 0.7798097 0.5477439 0.8165043 0.3723757 0.2732001
## [6,] 0.02274995 0.0119239 0.5216390 0.5750363 0.8330557 0.2190715
## [7,] 0.98743997 0.7065355 0.7718847 0.6516806 0.2217270 0.5360936
##           [,19]      [,20]      [,21]     [,22]      [,23]     [,24]
## [1,] 0.35444075 0.53707961 0.78347200 0.9007253 0.26652151 0.8994180
## [2,] 0.09909898 0.09898077 0.40377037 0.7729331 0.73009639 0.8550219
```

```
## [3,] 0.98894730 0.71588138 0.91533442 0.8562280 0.35475312 0.6244636
## [4,] 0.19302582 0.63734507 0.94579957 0.7682407 0.85476344 0.1125880
## [5,] 0.21973811 0.83956426 0.59718692 0.3560916 0.80323029 0.3842880
## [6,] 0.56218929 0.14263376 0.04669622 0.5432360 0.02906608 0.6012990
## [7,] 0.59866823 0.53239475 0.29950819 0.6137298 0.70489440 0.9322269
##          [,25]     [,26]     [,27]     [,28]      [,29]      [,30]
## [1,] 0.5563692 0.6098682 0.8281682 0.9777131 0.63383264 0.53019135
## [2,] 0.5382771 0.6109307 0.8256946 0.2873414 0.58798226 0.52043614
## [3,] 0.6103766 0.2314250 0.4575561 0.7980839 0.86604373 0.02147962
## [4,] 0.3448603 0.3844484 0.3010303 0.2489241 0.18042976 0.57789374
## [5,] 0.8227906 0.4713486 0.5061428 0.4427376 0.06615116 0.03453812
## [6,] 0.9952779 0.6873688 0.5396619 0.2712982 0.01049575 0.56144165
## [7,] 0.6923038 0.7997132 0.3971468 0.2659382 0.69752061 0.80674897
##           [,31]     [,32]     [,33]      [,34]      [,35]       [,36]
## [1,] 0.06290277 0.7551371 0.5952418 0.46615749 0.65070256 0.764779055
## [2,] 0.11040071 0.1756638 0.3640933 0.44667868 0.45981557 0.406169555
## [3,] 0.42120232 0.7955439 0.7434496 0.45327228 0.57655015 0.009060735
## [4,] 0.96677931 0.1819280 0.5712128 0.38306572 0.75445705 0.679987098
## [5,] 0.79365242 0.3562248 0.7932059 0.06433201 0.69341372 0.408988372
## [6,] 0.59429030 0.8795474 0.8692145 0.87284034 0.14825943 0.373647697
## [7,] 0.63357724 0.4311286 0.2484614 0.94996359 0.01668929 0.435580194
##          [,37]     [,38]      [,39]      [,40]     [,41]      [,42]
## [1,] 0.1429573 0.7035551 0.48200088 0.03231428 0.1522775 0.63680515
## [2,] 0.6889130 0.2851414 0.83272906 0.60299895 0.7085200 0.08945085
## [3,] 0.4729595 0.8880002 0.01146889 0.16332917 0.6696581 0.11554033
## [4,] 0.9200581 0.5095226 0.49370762 0.89049549 0.4547087 0.20983020
## [5,] 0.9318317 0.4127996 0.23341954 0.56185923 0.9800809 0.27356299
## [6,] 0.9684917 0.4567874 0.98578822 0.50113638 0.3008581 0.47764184
## [7,] 0.8663686 0.3062074 0.24880470 0.12599564 0.7321170 0.83930100
##          [,43]      [,44]      [,45]     [,46]      [,47]     [,48]
## [1,] 0.4381120 0.29735279 0.46078045 0.3037236 0.48980289 0.1467194
## [2,] 0.9689428 0.61410539 0.16608945 0.2953129 0.72884340 0.3089941
## [3,] 0.6330939 0.13019510 0.71577530 0.2372612 0.90767547 0.6834939
## [4,] 0.9183534 0.07097544 0.47839570 0.2250002 0.24244301 0.5405539
## [5,] 0.3585514 0.17601918 0.07234427 0.4344168 0.86440366 0.5093069
## [6,] 0.2766238 0.45764721 0.43464349 0.2157458 0.10383143 0.8395525
## [7,] 0.6856412 0.37256164 0.45046694 0.6374335 0.03241518 0.7968369
##          [,49]      [,50]       [,51]      [,52]      [,53]      [,54]
## [1,] 0.6804293 0.18843812 0.917237518 0.22090057 0.01438853 0.10597790
## [2,] 0.7942268 0.05365111 0.695386821 0.17539358 0.66292620 0.45850395
## [3,] 0.2628760 0.58866502 0.959568203 0.08459415 0.89127935 0.76591347
## [4,] 0.5257434 0.38122086 0.354673808 0.52854417 0.68981138 0.45639154
## [5,] 0.6415112 0.19026717 0.006272811 0.80152428 0.43060088 0.95557359
## [6,] 0.3408492 0.41116007 0.924567102 0.37214399 0.89541255 0.02149349
## [7,] 0.0402402 0.66746327 0.845771480 0.61169010 0.45838689 0.79300240
##          [,55]     [,56]      [,57]      [,58]      [,59]     [,60]
## [1,] 0.7134764 0.1932292 0.01998726 0.52797895 0.34068407 0.6893990
## [2,] 0.9206923 0.3400996 0.84381868 0.61554909 0.40017612 0.7325604
## [3,] 0.7073774 0.2832376 0.02428451 0.12841370 0.04671891 0.2821113
## [4,] 0.7134155 0.5013753 0.71950483 0.88043076 0.57102211 0.4151145
```

```
## [5,] 0.8055791 0.9878513 0.28203809 0.79439990 0.17599029 0.8183322
## [6,] 0.4705266 0.8944656 0.46899165 0.05497325 0.66808120 0.5673481
## [7,] 0.5990577 0.4316282 0.99358339 0.96646924 0.04242796 0.1535748
##          [,61]      [,62]      [,63]      [,64]     [,65]      [,66]
## [1,] 0.8862816 0.70249836 0.51139344 0.20706371 0.4742120 0.62099584
## [2,] 0.4925258 0.13027791 0.09613705 0.73403712 0.6086673 0.23995982
## [3,] 0.6723115 0.16879225 0.99513377 0.90356042 0.1375180 0.44233209
## [4,] 0.8650474 0.50704018 0.19197192 0.53780902 0.6889704 0.31078059
## [5,] 0.6911400 0.15682712 0.20362316 0.09381663 0.4456663 0.09593293
## [6,] 0.5549301 0.68242947 0.44206681 0.62961465 0.4377845 0.53575861
## [7,] 0.3369430 0.08826907 0.78599891 0.32637994 0.6587664 0.58110264
##           [,67]      [,68]      [,69]     [,70]      [,71]      [,72]
## [1,] 0.61661061 0.61201050 0.92403834 0.7207636 0.10879422 0.93121086
## [2,] 0.47996629 0.02946115 0.33281858 0.3949058 0.72481062 0.99486476
## [3,] 0.75634057 0.80231492 0.29107722 0.8969714 0.45679458 0.42063908
## [4,] 0.49887058 0.84461261 0.19299489 0.2081437 0.26171177 0.55612971
## [5,] 0.11558214 0.38258291 0.48369971 0.4990610 0.06954791 0.42757744
## [6,] 0.02584754 0.10560562 0.08651994 0.9218350 0.58153842 0.08579328
## [7,] 0.80828183 0.62468448 0.12111370 0.1061004 0.97428191 0.59777585
##            [,73]     [,74]     [,75]      [,76]      [,77]      [,78]
## [1,] 0.667058010 0.7072024 0.1929304 0.07324564 0.18951982 0.78234136
## [2,] 0.017730710 0.2394534 0.0269856 0.50969265 0.63704180 0.04355053
## [3,] 0.349889306 0.9944343 0.7097333 0.63474446 0.72566125 0.48557158
## [4,] 0.774193006 0.9254148 0.6527349 0.74196685 0.99882244 0.71884893
## [5,] 0.236792902 0.1381748 0.5259841 0.89964404 0.02155880 0.72330424
## [6,] 0.004859163 0.7121216 0.3268185 0.86727404 0.04848132 0.89822607
## [7,] 0.053942697 0.9891554 0.2813358 0.07397298 0.40430883 0.64862291
##          [,79]     [,80]      [,81]      [,82]       [,83]     [,84]
## [1,] 0.3102931 0.8969315 0.83219465 0.92566621 0.991810375 0.7681596
## [2,] 0.7180196 0.3177821 0.91567702 0.62688096 0.154569279 0.5324568
## [3,] 0.3395753 0.8907586 0.56068699 0.78590627 0.004776266 0.2196734
## [4,] 0.6023597 0.7453633 0.40735603 0.99322113 0.086577715 0.7648168
## [5,] 0.0626057 0.3880156 0.65893980 0.20941501 0.457873928 0.3032477
## [6,] 0.7031731 0.5211729 0.07575416 0.06570001 0.292229206 0.6979886
## [7,] 0.9148319 0.2668772 0.96250301 0.29759685 0.404352822 0.8414569
##           [,85]     [,86]      [,87]      [,88]     [,89]     [,90]
## [1,] 0.99817984 0.5919962 0.90242787 0.03890705 0.3904713 0.9153404
## [2,] 0.75204623 0.6157334 0.33379282 0.73650352 0.9594037 0.2706786
## [3,] 0.36914846 0.5879811 0.83888282 0.53355567 0.9128614 0.2267724
## [4,] 0.27774153 0.9693068 0.41130669 0.75240228 0.9972134 0.3839503
## [5,] 0.00535286 0.7944734 0.91840102 0.24419971 0.9998173 0.1556707
## [6,] 0.38776166 0.1117674 0.05305243 0.23258153 0.6185764 0.1034918
## [7,] 0.81447545 0.9916051 0.18400432 0.39902291 0.4700177 0.3502257
##           [,91]      [,92]     [,93]       [,94]      [,95]     [,96]
## [1,] 0.25799270 0.76462069 0.1494273 0.007233441 0.43268179 0.8440924
## [2,] 0.55808128 0.28852505 0.7434820 0.639143466 0.90893671 0.9307423
## [3,] 0.13237267 0.75999810 0.8893595 0.483877485 0.33012246 0.1806208
## [4,] 0.47317544 0.09801729 0.1159442 0.439732011 0.09583301 0.3867446
## [5,] 0.80093894 0.95966613 0.2244408 0.786738058 0.08349419 0.1800014
## [6,] 0.55552623 0.07420703 0.6276363 0.658580460 0.65422058 0.3326875
```

```
## [7,] 0.05153204 0.73587936 0.2236765 0.508881353 0.75025979 0.3426896
##          [,97]     [,98]     [,99]    [,100]    [,101]     [,102]
## [1,] 0.8611910 0.6736368 0.4326152 0.2170376 0.8633164 0.69443390
## [2,] 0.9187846 0.5980011 0.3930895 0.2060755 0.5048547 0.08189908
## [3,] 0.6827469 0.7497609 0.1822099 0.1689044 0.5163440 0.84092669
## [4,] 0.7988146 0.6830547 0.3493793 0.4165844 0.6547881 0.75559309
## [5,] 0.4336934 0.2784893 0.6886105 0.8253391 0.5549053 0.71880810
## [6,] 0.6681592 0.9408279 0.9593975 0.2022780 0.2852109 0.13420901
## [7,] 0.5199679 0.8131354 0.4990536 0.8477770 0.7018382 0.73622748
##          [,103]     [,104]     [,105]     [,106]      [,107]    [,108]
## [1,] 0.00929070 0.54129864 0.64630479 0.48141790 0.230371349 0.4494672
## [2,] 0.10989637 0.08801390 0.49462861 0.89616316 0.023946382 0.5312700
## [3,] 0.58412134 0.01628532 0.73303190 0.98090826 0.274931159 0.9823788
## [4,] 0.99127731 0.97470720 0.51709077 0.52971328 0.643916116 0.4268339
## [5,] 0.22236965 0.74921922 0.30811034 0.45900502 0.013285816 0.7863120
## [6,] 0.07031604 0.58687219 0.13101585 0.45348288 0.006495902 0.5837720
## [7,] 0.88515810 0.17337383 0.07318512 0.07882848 0.715869990 0.7995420
##         [,109]    [,110]    [,111]    [,112]    [,113]    [,114]    [,115]
## [1,] 0.4328599 0.9682626 0.9190493 0.5731611 0.6708728 0.5890649 0.5927029
## [2,] 0.3604686 0.7107097 0.9059142 0.9113703 0.7722951 0.2710790 0.2600709
## [3,] 0.4854962 0.1713843 0.6473648 0.1438621 0.5863838 0.7866704 0.9020901
## [4,] 0.2039096 0.2145008 0.2937146 0.5199002 0.4266611 0.4742475 0.7831325
## [5,] 0.7348667 0.8386832 0.3251621 0.4893448 0.3044134 0.2464204 0.3789222
## [6,] 0.4594524 0.0218795 0.3411521 0.3540894 0.6635306 0.9151334 0.4272254
## [7,] 0.8220139 0.9012593 0.4988079 0.3277770 0.4764548 0.1544275 0.1999663
##         [,116]
## [1,] 0.1134410
## [2,] 0.6692852
## [3,] 0.7328469
## [4,] 0.9702275
## [5,] 0.9527144
## [6,] 0.2504574
## [7,] 0.3435233


# output.v
setwd('C:\\Users\\yancx\\Desktop\\Thesis\\Peter fMRI data\\')
load("output.v")

class(output.v)

## [1] "array"

dim(output.v)

## [1]      2 135836    100

extract.116.v <- function(Y){
  out.arr <- array(data = 0, dim = c(100,1,2,116))
  for (j in 1:116) {
```

```
    for (i in 1:2) {
      out.arr[,,i,j] <- Y[i,(1171*(j-1)+1),]
    }
  }
  out.arr
}
out.array <- extract.116.v(output.v)


out.df <- data.frame(data=out.array)
person <- paste(rep(1:25, each=4), sep = "")
out.df <- cbind(person, out.df)


colnames(out.df) <- c("person", 1:232)


adjsq.func.v <- function(data){
  out.var <- matrix(data = 0, nrow = 232)
  for (i in 2:233) {
    out.var[(i-1),] <- summary(lm(out.df[,i] ~ person, data = out.df))$adj.r.
squared
  }
  out.var <- matrix(data = out.var, nrow = 2, ncol = 116)
  out.var
}
adjsq.out.v <- adjsq.func.v(out.df)

adjsq.out.v [adjsq.out.v  < 0] <- 0
adjsq.out.v

##      [,1]       [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11]
## [1,]    0 0.05899422    0    0    0    0    0    0    0     0     0
## [2,]    0 0.04772655    0    0    0    0    0    0    0     0     0
##           [,12] [,13]      [,14]      [,15]     [,16] [,17]     [,18]
## [1,] 0.05256526     0 0.04059682 0.05785718 0.1231110     0 0.2663078
## [2,] 0.05497705     0 0.03777781 0.03891055 0.1249247     0 0.2712387
##      [,19] [,20] [,21]      [,22] [,23] [,24]     [,25]     [,26]
## [1,]     0     0     0 0.09141593     0     0 0.1709059 0.1174849
## [2,]     0     0     0 0.09189232     0     0 0.1809692 0.1196714
##          [,27]      [,28] [,29] [,30] [,31] [,32] [,33]     [,34] [,35]
## [1,] 0.1229084 0.06691678     0     0     0     0     0 0.2250841     0
## [2,] 0.1342421 0.09397375     0     0     0     0     0 0.2387908     0
##           [,36]      [,37]     [,38]      [,39] [,40]     [,41]     [,42]
## [1,] 0.09229229 0.09478323 0.1158448 0.07680260     0 0.1217537 0.1027655
## [2,] 0.10449860 0.04904605 0.1189993 0.06608807     0 0.1036391 0.1005818
##      [,43] [,44] [,45]      [,46] [,47] [,48]       [,49]      [,50] [,51]
## [1,]     0     0     0 0.07344790     0     0 0.008577619 0.04301075     0
## [2,]     0     0     0 0.08037808     0     0 0.032912386 0.04705502     0
##      [,52] [,53] [,54]      [,55]     [,56] [,57]      [,58]     [,59]
```

```
## [1,]     0     0     0 0.04374575 0.1022766     0 0.00395331 0.1760955
## [2,]     0     0     0 0.05099243 0.1155410     0 0.00000000 0.1589436
##            [,60]      [,61]      [,62]      [,63] [,64]       [,65] [,66]
## [1,] 0.000000000 0.08387791 0.02893927 0.04747483     0 0.001396705     0
## [2,] 0.008005893 0.10324925 0.04185982 0.03415152     0 0.000000000     0
##      [,67] [,68]      [,69]      [,70] [,71] [,72] [,73]      [,74]
## [1,]     0     0 0.02303322 0.10888311     0     0     0 0.07791799
## [2,]     0     0 0.02542621 0.05017896     0     0     0 0.07866825
##           [,75]      [,76]      [,77] [,78]      [,79] [,80] [,81]
## [1,] 0.00000000 0.01232109 0.08229854     0 0.03797791     0     0
## [2,] 0.02813087 0.03018651 0.05077243     0 0.06338682     0     0
##           [,82] [,83] [,84] [,85] [,86] [,87]      [,88]     [,89]
## [1,] 0.03571153     0     0     0     0     0 0.09466044 0.1129366
## [2,] 0.04698150     0     0     0     0     0 0.09104430 0.1236580
##           [,90] [,91] [,92]      [,93]      [,94]     [,95] [,96]
## [1,] 0.08891020     0     0 0.06596561 0.09973603 0.1376509     0
## [2,] 0.07844016     0     0 0.08884245 0.10185582 0.1277669     0
##           [,97]      [,98]      [,99] [,100] [,101]     [,102]
## [1,] 0.04797241 0.05081409 0.08865547      0      0 0.09453402
## [2,] 0.06232223 0.05811943 0.09942280      0      0 0.09971610
##             [,103]      [,104] [,105] [,106] [,107] [,108] [,109]    [,110]
## [1,] 0.0000000000 0.014953397      0      0      0      0      0 0.1125626
## [2,] 0.0001183567 0.008788233      0      0      0      0      0 0.1024753
##      [,111]    [,112]     [,113] [,114]    [,115] [,116]
## [1,]      0 0.1651118 0.06657843      0 0.2537711      0
## [2,]      0 0.1196668 0.06156965      0 0.2503594      0
```

```
p.value.func <- function(data){
  out.pvalue <- matrix(data = 0, nrow = 232)
  for (i in 2:233) {
    out.pvalue[(i-1),] <- anova(lm(out.df[,i] ~ person, data = out.df))$'Pr(>
F)'[1]
  }
  out.pvalue <- matrix(data = out.pvalue, nrow = 2, ncol = 116)
  out.pvalue
}
p.value.v <- p.value.func(out.df)
p.value.v
```

```
##           [,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
## [1,] 0.6676980 0.2236668 0.8499273 0.8149259 0.7466525 0.8526446 0.6687835
## [2,] 0.6459225 0.2645588 0.6358224 0.8160406 0.6286101 0.9018339 0.5447495
##           [,8]      [,9]     [,10]     [,11]     [,12]     [,13]     [,14]
## [1,] 0.8042677 0.8834027 0.8837883 0.8736588 0.2464859 0.5003875 0.2925503
## [2,] 0.8417301 0.9297368 0.8993601 0.8574925 0.2377643 0.6452338 0.3040496
##          [,15]      [,16]     [,17]       [,18]     [,19]     [,20]
## [1,] 0.2276022 0.06978194 0.6543363 0.001403947 0.8280825 0.9273250
## [2,] 0.2994003 0.06717094 0.7107979 0.001186985 0.8521785 0.9503277
##          [,21]     [,22]     [,23]     [,24]      [,25]      [,26]
## [1,] 0.8875990 0.1297612 0.6532619 0.8837550 0.02321293 0.07840181
```

```
## [2,] 0.9242247 0.1286399 0.7776898 0.8517602 0.01794746 0.07495654
##           [,27]     [,28]     [,29]     [,30]     [,31]     [,32]
## [1,] 0.07007859 0.1974566 0.4859939 0.8501569 0.9047278 0.7209244
## [2,] 0.05497244 0.1238246 0.5812980 0.8528644 0.8445208 0.7322964
##          [,33]       [,34]     [,35]    [,36]     [,37]      [,38]
## [1,] 0.5159547 0.005223830 0.9607078 0.127704 0.1219887 0.08106766
## [2,] 0.5460359 0.003434518 0.9477822 0.101515 0.2595551 0.07600260
##          [,39]     [,40]      [,41]     [,42]     [,43]     [,44]
## [1,] 0.1677326 0.7562823 0.07178863 0.1049601 0.4843698 0.9211074
## [2,] 0.2000987 0.7396352 0.10321253 0.1094272 0.5083220 0.8512261
##          [,45]     [,46]     [,47]    [,48]     [,49]     [,50]     [,51]
## [1,] 0.6198414 0.1774490 0.6023524 0.596603 0.4354226 0.2828955 0.4952916
## [2,] 0.5663802 0.1577922 0.6372934 0.692554 0.3244461 0.2671269 0.5352031
##          [,52]     [,53]     [,54]     [,55]      [,56]     [,57]
## [1,] 0.7093333 0.9622590 0.5603764 0.2799914 0.10594791 0.5374747
## [2,] 0.7022282 0.9544999 0.6327866 0.2522769 0.08156922 0.5087715
##          [,58]      [,59]     [,60]     [,61]     [,62]     [,63]
## [1,] 0.4578297 0.02035158 0.8372634 0.1484737 0.3415954 0.2655198
## [2,] 0.5299946 0.03115330 0.4381755 0.1039895 0.2874764 0.3191870
##          [,64]     [,65]     [,66]    [,67]     [,68]     [,69]     [,70]
## [1,] 0.6704383 0.4703422 0.5103445 0.928555 0.5570350 0.3678525 0.0931875
## [2,] 0.6188021 0.5464804 0.5152975 0.928246 0.6123286 0.3571079 0.2553037
##          [,71]     [,72]     [,73]     [,74]     [,75]     [,76]     [,77]
## [1,] 0.9106303 0.5227067 0.8060774 0.1645858 0.5381231 0.4175320 0.1526287
## [2,] 0.8905947 0.6636065 0.7596478 0.1624925 0.3451367 0.3361657 0.2530934
##          [,78]     [,79]     [,80]     [,81]     [,82]     [,83]     [,84]
## [1,] 0.6078016 0.3032255 0.9887231 0.7241353 0.3126282 0.9839123 0.4922712
## [2,] 0.7013690 0.2088725 0.9548922 0.8359900 0.2674090 0.9810762 0.6444104
##          [,85]     [,86]     [,87]     [,88]      [,89]     [,90]
## [1,] 0.5706196 0.4836476 0.6482661 0.1222658 0.08597007 0.1357781
## [2,] 0.5252147 0.4912883 0.6361806 0.1306409 0.06898597 0.1631269
##          [,91]     [,92]     [,93]     [,94]      [,95]     [,96]
## [1,] 0.7344377 0.9780551 0.2004911 0.1111956 0.05098980 0.7990916
## [2,] 0.6425919 0.9780086 0.1359436 0.1068037 0.06323818 0.9482011
##          [,97]     [,98]    [,99]    [,100]    [,101]    [,102]    [,103]
## [1,] 0.2636223 0.2529386 0.136401 0.6895028 0.8263361 0.1225517 0.5734368
## [2,] 0.2123982 0.2266907 0.111856 0.5887478 0.8874761 0.1112375 0.4766274
##         [,104]    [,105]    [,106]    [,107]    [,108]    [,109]
## [1,] 0.4051035 0.9580460 0.7960052 0.6427722 0.4842246 0.8935441
## [2,] 0.4344097 0.9490778 0.7803167 0.5500522 0.5351016 0.9295218
##          [,110]    [,111]     [,112]    [,113]    [,114]      [,115]
## [1,] 0.08661704 0.8681257 0.02681023 0.1985325 0.9130464 0.002129056
## [2,] 0.10554552 0.9758605 0.07496362 0.2149137 0.8794311 0.002378397
##         [,116]
## [1,] 0.6654877
## [2,] 0.6426594
```