

# ADNI data R package

Los datos de ADNI estan empaquetados en formato R por Michel Donahue. El paquete se llama ADNIMERGE y está instalado en el servidor. Puede descargarse de la pagina de ADNI. Hay además un [grupo de google de adnimerge](#) que puede ser util.

## Instalación

Copiando del README:

```
ADNI Data Package for R
```

```
For users of R, we have developed a data package "ADNIMERGE" which contains coded data, documentation, and analysis vignettes. It depends on Frank Harrell's Hmisc package which can be installed from the R package repository (CRAN) by:
```

```
R prompt> install.packages("Hmisc")
```

After downloading the compressed ADNIMERGE\_0.0.1.tar.gz file, it can be installed to your R system by entering the following at an R prompt:

```
R prompt> install.packages("/path/to/file/ADNIMERGE_0.0.1.tar.gz", repos = NULL, type = "source")
```

If you have trouble installing, more information can be found at <http://cran.r-project.org/doc/manuals/R-admin.html#Installing-packages>.

To browse document and analysis vignettes,

```
R prompt> help(package = "ADNIMERGE")
```

To view, for example, documentation for the adas:

```
R prompt> ?adas
```

## Cargar la biblioteca

```
library("ADNIMERGE")
```

## MCI due to AD

Los MCI por AD se sacan:

```
dxmci <- dxsum[dxsum[, "DXMDUE"] == "MCI due to Alzheimer's Disease" &
```

```
!is.na(dxsum$DXMDUE),]
```

y además amnesicos

```
dxmci=dxsum[dxsum[, "DXMDUE"] == "MCI due to Alzheimer's Disease" &
!is.na(dxsum$DXMDUE) & dxsum[, "DXMDES"] == "MCI - Memory features
(amnestic)" & !is.na(dxsum$DXMDES),]
```

## First try

```
mcimerged0 <- merge(dxmci, adnimerge, by=c("RID", "VISCODE"))
mciwav <- mcimerged0[!is.na(mcimerged0$AV45) ,]
```

ahora voy a añadir las variables de delay real que me ha calculado sergi,

```
comp_delay_recall <- read.csv("adni_delay_recall.csv", sep=",")
mci_dr <- merge(adnimerge, comp_delay_recall, by=c("RID", "VISCODE"))
```

## Merging from several tables

Merging and calculating data

```
mt1 <- merge(adas, adnimerge, by=c("RID", "VISCODE") )
mt1$vAGE = mt1$AGE + mt1$Years
a <- lm(mt1$Hippocampus ~ mt1$ICV)
b=a$coefficients[[2]]
mt1$aHV = mt1$Hippocampus - b*(mt1$ICV - mean(mt1$ICV, na.rm = TRUE))
```

Preparing data to fit

```
data <- data.frame(mt1$aHV, mt1$vAGE, mt1$PTGENDER, mt1$PTEDUCAT,
mt1$Q4SCORE)
datac <- data[complete.cases(data),]
```

Just fitting

```
fit <- cusp(y ~ mt1.Q4SCORE, alpha ~ mt1.aHV +mt1.vAGE + mt1.PTGENDER
+mt1.PTEDUCAT, beta ~ mt1.aHV +mt1.vAGE + mt1.PTGENDER +mt1.PTEDUCAT, datac)
```

A working example, for FDG data and NEUROBAT ADNI table

```
mt1 <- merge(neurobat, adnimerge, by=c("RID", "VISCODE") )
mt1$vAGE = mt1$AGE + mt1$Years
data <- data.frame(mt1$FDG, mt1$vAGE, mt1$PTGENDER, mt1$PTEDUCAT,
mt1$AVDEL30MIN)
datac <- data[complete.cases(data),]
fit <- cusp(y ~ mt1.AVDEL30MIN, alpha ~ mt1.FDG +mt1.vAGE + mt1.PTGENDER
```

```
+mt1.PTEDUCAT, beta ~ mt1.FDG +mt1.vAGE + mt1.PTGENDER +mt1.PTEDUCAT, datac)
```

y lo miramos a ver que tal queda:

```
> summary(fit)

Call:
cusp(formula = y ~ mt1.AVDEL30MIN, alpha = alpha ~ mt1.FDG +
      mt1.vAGE + mt1.PTGENDER + mt1.PTEDUCAT, beta = beta ~ mt1.FDG +
      mt1.vAGE + mt1.PTGENDER + mt1.PTEDUCAT, data = datac)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.087409 -0.279966 -0.002222  0.592400  3.437777

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
a[(Intercept)]   -3.481541   0.293956  -11.844 < 2e-16 ***
a[mt1.FDG]        0.418469   0.025365   16.498 < 2e-16 ***
a[mt1.vAGE]       -0.010894   0.002552   -4.269 1.96e-05 ***
a[mt1.PTGENDERFemale] 0.282399   0.036690    7.697 1.39e-14 ***
a[mt1.PTEDUCAT]   0.069970   0.007042    9.936 < 2e-16 ***
b[(Intercept)]    5.548930   0.399315   13.896 < 2e-16 ***
b[mt1.FDG]       -0.555766   0.004995  -111.262 < 2e-16 ***
b[mt1.vAGE]       -0.011760   0.004163   -2.825 0.00473 **
b[mt1.PTGENDERFemale] 0.308486   0.061716    4.998 5.78e-07 ***
b[mt1.PTEDUCAT]   0.027550   0.011087    2.485 0.01296 *
w[(Intercept)]   -1.827114   0.016373  -111.594 < 2e-16 ***
w[mt1.AVDEL30MIN] 0.262274   0.002336  112.278 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

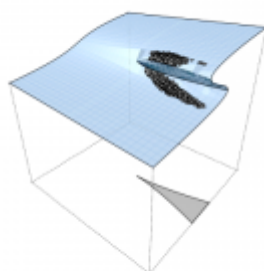
Null deviance: 4464.2 on 3319 degrees of freedom
Linear deviance: 50944.0 on 3314 degrees of freedom
Logist deviance: NA on NA degrees of freedom
Delay deviance: 1690.2 on 3308 degrees of freedom

                R.Squared   logLik npar      AIC      AICc      BIC
Linear model 0.2150188 -9243.943   6 18499.885 18499.910 18536.531
Cusp model  0.6407748 -3427.447  12  6878.895  6878.989  6952.188
---
Note: R.Squared for cusp model is Cobb's pseudo-R^2. This value
      can become negative.

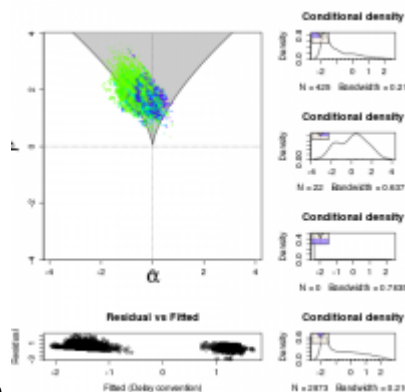
Chi-square test of linear vs. cusp model

X-squared = 1.163e+04, df = 6, p-value = 0

Number of optimization iterations: 82
```



*cusp3d(fit)* give the 3D graph,



*plot(fit)* give the proyection on control plane and more info

## How to do this with Composite Scores

```

tmp_np <- merge(adas, neurobat, by=c("RID", "VISCODE") )
mt2fa <- merge(tmp_np, adnimerge, by=c("RID", "VISCODE") )
rm(tmp_np)
mt2fa$vAGE = mt2fa$AGE + mt2fa$Years
data <- data.frame(mt2fa$FDG, mt2fa$vAGE, mt2fa$PTGENDER, mt2fa$PTEDUCAT,
mt2fa$Q4SCORE, mt2fa$AVDEL30MIN)
datac <- data[complete.cases(data),]
datac$zDR = (datac$mt2fa.Q4SCORE -
mean(datac$mt2fa.Q4SCORE))/sd(datac$mt2fa.Q4SCORE)
datac$zAVD = (mean(datac$mt2fa.AVDEL30MIN) -
datac$mt2fa.AVDEL30MIN)/sd(datac$mt2fa.AVDEL30MIN)
gfam <- data.frame(datac$zDR, datac$zAVD)
famod <- fa(gfam, scores="regression")
datac$DRCS <- famod$scores
fitdr <- cusp(y ~ DRCS, alpha ~ mt2fa.FDG +mt2fa.vAGE + mt2fa.PTGENDER
+mt2fa.PTEDUCAT, beta ~ mt2fa.FDG +mt2fa.vAGE + mt2fa.PTGENDER
+mt2fa.PTEDUCAT, datac)

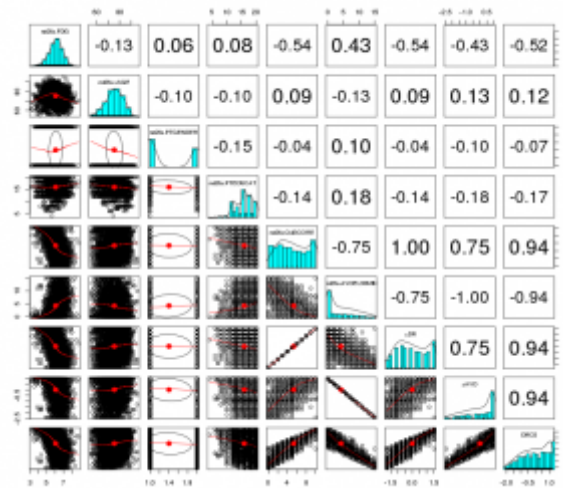
```

The composites can be seen here:

```

pairs.panels(datac, pch=21)

```



Now, notice that your results will be as good as your former worst (or even worse!).

```
> summary(fitdr)

Call:
cusp(formula = y ~ DRCS, alpha = alpha ~ mt2fa.FDG + mt2fa.vAGE +
      mt2fa.PTGENDER + mt2fa.PTEDUCAT, beta = beta ~ mt2fa.FDG +
      mt2fa.vAGE + mt2fa.PTGENDER + mt2fa.PTEDUCAT, data = datac)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.60924 -0.63803 -0.05687  0.35020  3.02185

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
a[(Intercept)]    4.861677   0.302969  16.047 < 2e-16 ***
a[mt2fa.FDG]     -0.658838   0.018241 -36.119 < 2e-16 ***
a[mt2fa.vAGE]      0.010701   0.002784   3.844 0.000121 ***
a[mt2fa.PTGENDERFemale] -0.266786   0.007268 -36.705 < 2e-16 ***
a[mt2fa.PTEDUCAT] -0.073632   0.007562  -9.737 < 2e-16 ***
b[(Intercept)]    5.945053   0.365065  16.285 < 2e-16 ***
b[mt2fa.FDG]     -0.741062           NA         NA         NA
b[mt2fa.vAGE]    -0.017238   0.004638  -3.717 0.000202 ***
b[mt2fa.PTGENDERFemale] 0.600601   0.040893  14.687 < 2e-16 ***
b[mt2fa.PTEDUCAT] 0.042803   0.012205   3.507 0.000453 ***
w[(Intercept)]   0.426748   0.014152  30.154 < 2e-16 ***
w[DRCS]          1.169788   0.008142 143.666 < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Null deviance: 3895.6 on 3317 degrees of freedom
Linear deviance: 2026.9 on 3312 degrees of freedom
Logist deviance: NA on NA degrees of freedom
Delay deviance: 2475.7 on 3306 degrees of freedom

R.Squared   logLik npar   AIC   AICc   BIC
```

```
Linear model 0.2880251 -3890.378 6 7792.755 7792.781 7829.398
Cusp model 0.3869597 -3622.986 12 7269.971 7270.066 7343.257
---
```

Note: R.Squared for cusp model is Cobb's pseudo-R<sup>2</sup>. This value can become negative.

Chi-square test of linear vs. cusp model

X-squared = 534.8, df = 6, p-value = 0

Number of optimization iterations: 43

From:

<http://imagen.fundacioace.com/wiki/> - **Detritus Wiki**

Permanent link:

[http://imagen.fundacioace.com/wiki/doku.php?id=neuroimagen:adni\\_data](http://imagen.fundacioace.com/wiki/doku.php?id=neuroimagen:adni_data)

Last update: **2020/08/04 10:58**

