

Neurohacking

```
library("oro.dicom")

setwd("C:/Neurohacking/Neurohacking_data/BRAINIX/DICOM/FLAIR")
slice=readDICOM("IM-0001-0011.dcm")
class(slice)
#explore the data
names(slice)
class(slice$hdr)
class(slice$hdr[[1]])
class(slice$img)
class(slice$img[[1]])
dim(slice$img[[1]])

#view the image
d=dim(t(slice$img[[1]]))
image(1:d[1],1:d[2],t(slice$img[[1]]),col=gray(0:64/64))

#look at the actual numbers

slice$img[[1]][101:105,121:125]

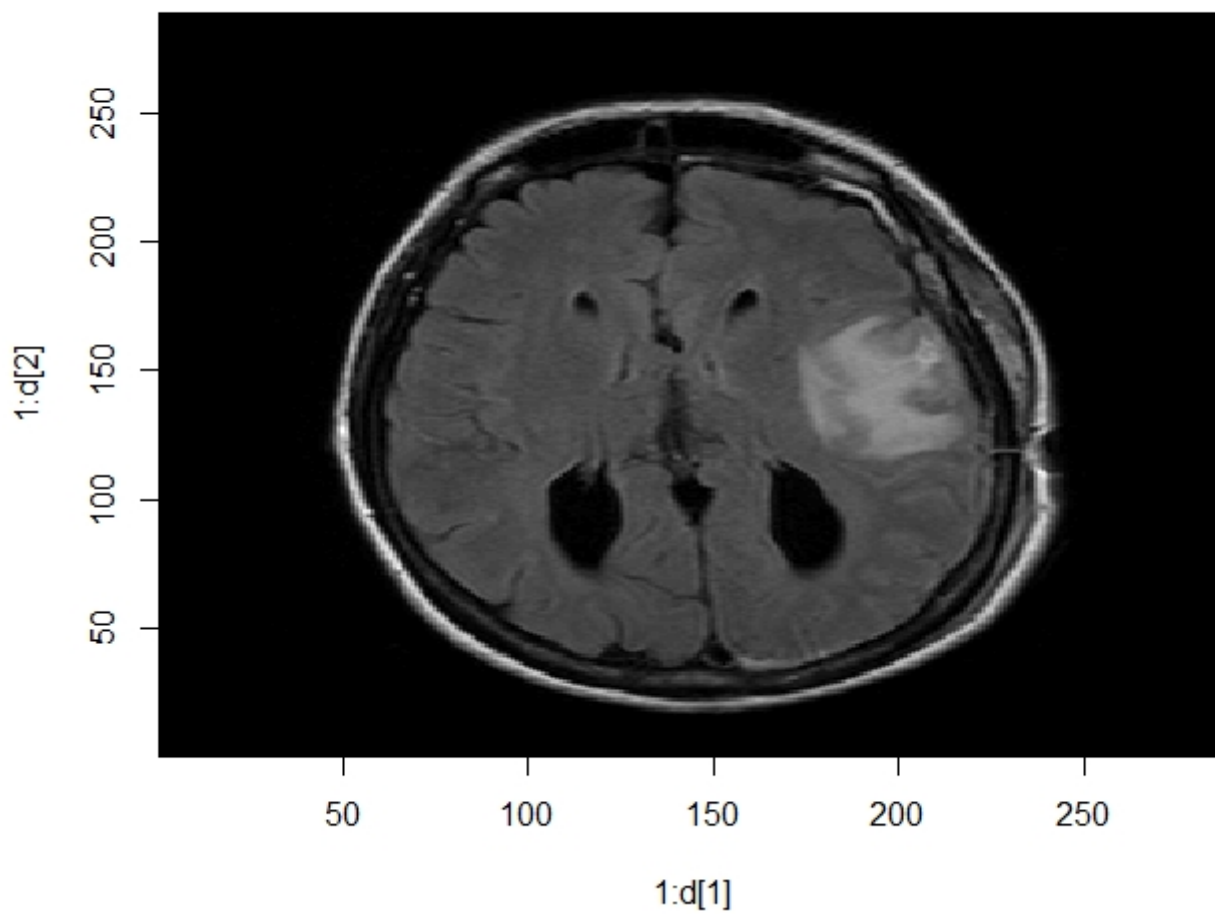
#plot entire thing - make density instead of count
hist(slice$img[[1]][,],breaks=50 ,xlab="FLAIR",prob=T, col=rgb(0,1,1,1/4),main="")

#explore header info

hdr=slice$hdr[[1]]
names(hdr)
hdr$name
#lookk at resoltuion
hdr[hdr$name == "PixelSpacing", "value"]
#flip angle
hdr[hdr$name == "FlipAngle",]

#change directory

setwd("C:/Neurohacking/Neurohacking_data/BRAINIX/DICOM")
all_slices_T1=readDICOM("T1/")
dim(all_slices_T1$img[[1]])
hdr=all_slices_T1$hdr[[1]]
```



dicom slice

```

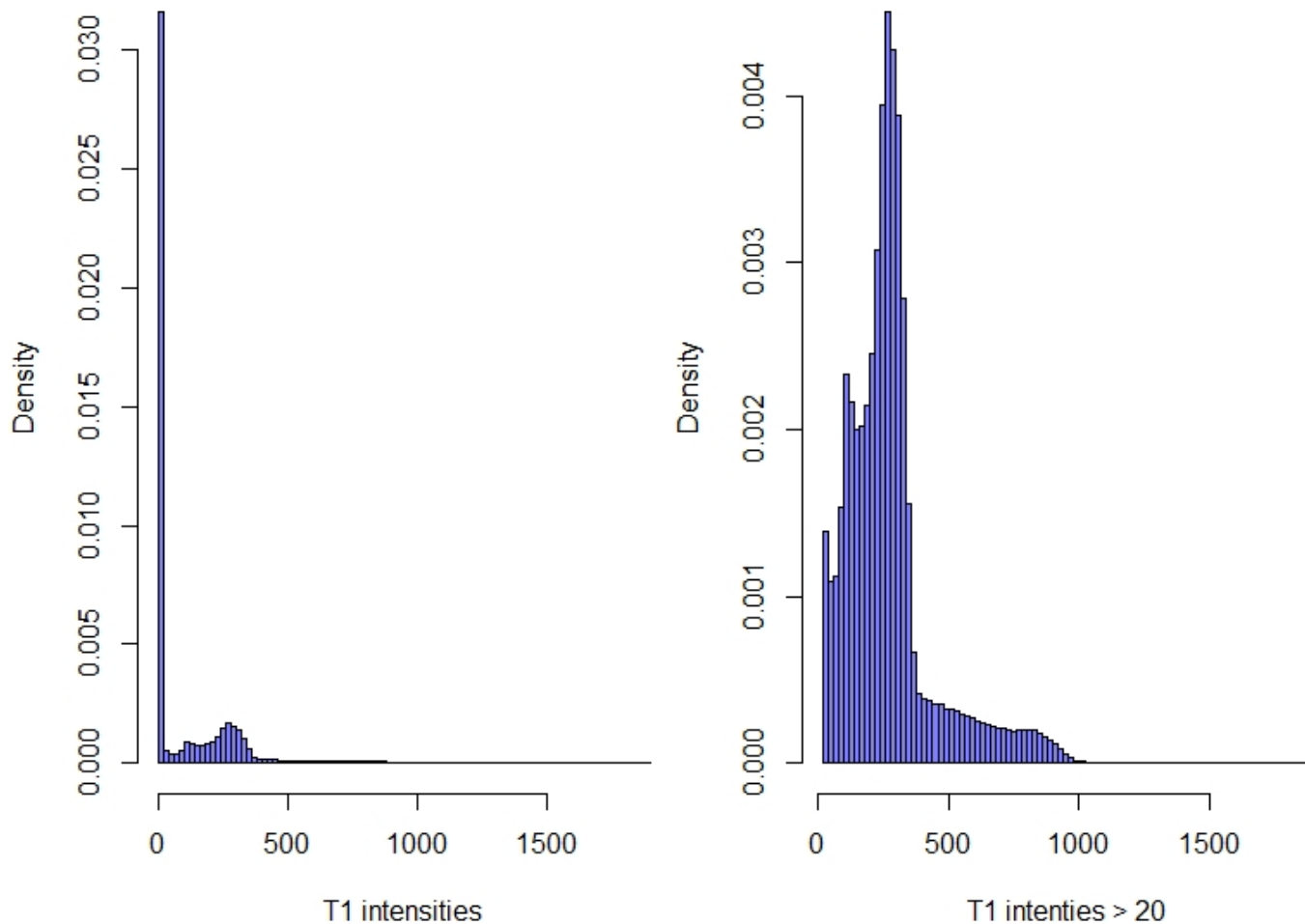
#nifti stuff
setwd("C:/Neurohacking/Neurohacking_data/BRAINIX/DICOM")
all_slices_T1=readDICOM("T1/")
dim(all_slices_T1$img[[11]])
hdr=all_slices_T1$hdr[[11]]

#convert from dicom to nifti using dicom2nifti

nii_T1=dicom2nifti(all_slices_T1)
#check how man slices and pixels
d=dim(nii_T1); d; class(nii_T1)
#pick out 11th slice
image(1:d[1],1:d[2],nii_T1[, ,11],col=gray(0:64/64),xlab="",ylab="")

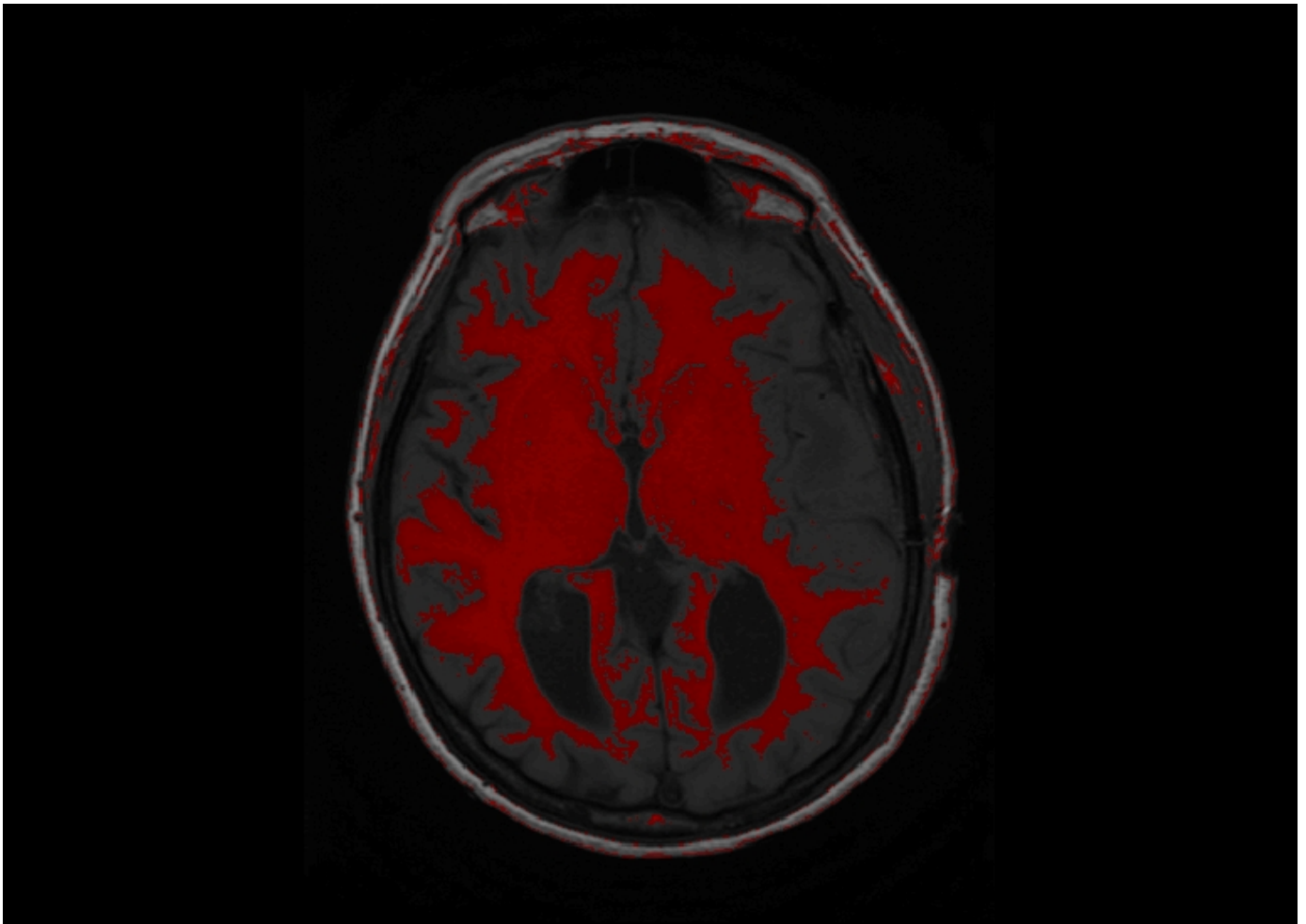
setwd("C:/Neurohacking/Neurohacking_data/BRAINIX/NIFTI")
fname="Output_3D_File"
#look for the files
writeNIfTI(nim=nii_T1,filename=fname)
list.files(getwd(),patter="Output_3D_File")
list.files(getwd(),pattern="T")
#don't reorient
nii_T2=readNIfTI("T2.nii.gz",reorient=FALSE)
dim(nii_T2)
print({nii_T1 = readNIfTI(fname=fname)})
image(1:d[1],1:d[2],nii_T1[, ,11],col=xlab="",ylab="")
#yellowish colours in this plot are where intensities in this plot are very high
graphics::image
heat.colors(12)
#use nifti instead of R , r treats different types of objects differently
image(nii_T1,z=11,plot.type="single")
oro.nifti::image
#plot all slices - but it wont like that
#image(nii_T1)
#all planes coronal,axial
orthographic(nii_T1,xyz=c(200,220,11))
par(mfrow=c(1,2));
o<-par(mar=c(4,4,0,0))
hist(nii_T1,breaks=75,prob=T,xlab="T1 intensities", col=rgb(0,0,1,1/2),main="");
#have to do >20 get rid of all black area which is meaningless
hist(nii_T1[nii_T1 > 20],breaks = 75, prob=T,xlab="T1 intensities > 20", col=rgb(0,0,1,1/2),main="")

```



histograms

```
#create a mask for particular value intensities
#this is the white matter of the brain where the axons are in the brains and we see the skull
is_btw_300_400 <- ((nii_T1>300) & (nii_T1<400))
nii_T1_mask<-nii_T1
nii_T1_mask[!is_btw_300_400]=NA
overlay(nii_T1,nii_T1_mask,z=11,plot.type="single")
#can do all but this breaks my computer
#overlay(nii_T1,nii_T1_mask)
#can also do orthographic
orthographic(nii_T1,nii_T1_mask,xyz=c(200,2,2,11), text="image overlaid with mask",text.cex=
1.5)
```



whitematter

```

#read data

mridir <- "C:/Neurohacking/Neurohacking_data/kirby21/visit_1/113"
T1 <- readNIfTI(file.path(mridir,"/113-01-MPRAGE.nii"),reorient=FALSE)
orthographic(T1)

#read in and view binary mask for the image

mask <- readNIfTI(file.path(mridir,"/113-01-MPRAGE_mask.nii"),reorient=FALSE)
orthographic(mask)
#must be in the same dimensions
#areas not in the mask are now gone
#got rid of bone area and things that arent actually brain
masked.T1 <- T1*mask
orthographic(masked.T1)

#take one image away from another
#also can do multiply, modulus, exponential, log transform etc

T1.follow <- readNIfTI(file.path(mridir,'/113-02-MPRAGE.nii'),reorient=FALSE)
subtract.T1 <- T1.follow - T1
min(subtract.T1)
max(subtract.T1)
orthographic(subtract.T1)

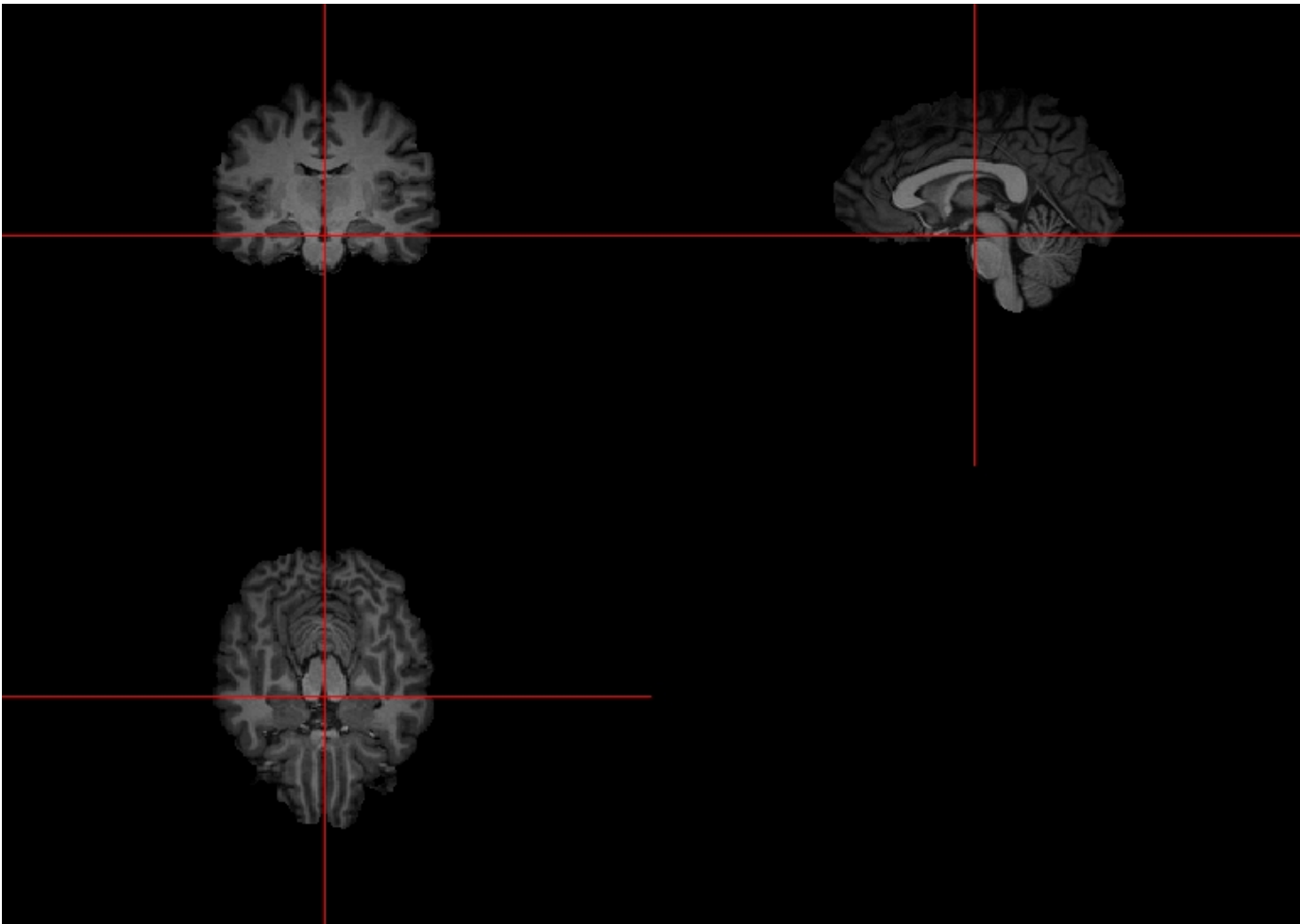
im_hist<-hist(T1,plot=FALSE)
par(mar=c(5,4,4,4) + 0.3)
col1=rgb(0,0,1,1/2)
plot(im_hist$mids,im_hist$count,log="y",type="h",lwd=10,lend=2,col=col1,xlab="intentsity valu
es",ylab="count (log)")
#linear transfomration
par(new=TRUE)
curve(x*1,axes=FALSE,xlab="",ylab="",col=2,lwd=3)
axis(side=4,at=pretty(range(im_hist$mids))/max(T1),labels=pretty(range(im_hist$mids)))
mtext("original intensity",side=4,line=2)
#transform them by differential, define a linear spline, break line in to knots
lin.sp<-function(x,knots,slope)
{
  knots<-c(min(x),knots,max(x))
  slopeS<-slope[1]
  for(j in 2:length(slope)){slopeS<-c(slopeS,slope[j]-sum(slopeS))}

  rvals<-numeric(length(x))
  for(i in 2:length(knots))
    {rvals<-ifelse(x>=knots[i-1],slopeS[i-1]*(x-knots[i-1])+rvals,rvals)}
  return(rvals)}
knots.vals<-c(.3,.6)
slp.vals<-c(1,.5,.25)

```



ortho



takeawaywhitematter