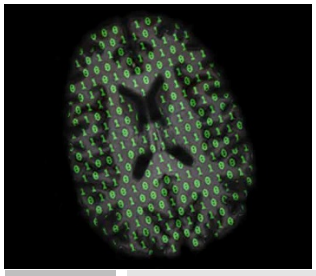
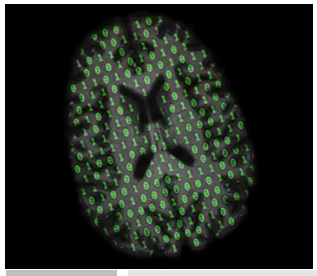
THEME 2/LECTURE 5: TRANSFORMATIONS AND SMOOTHING



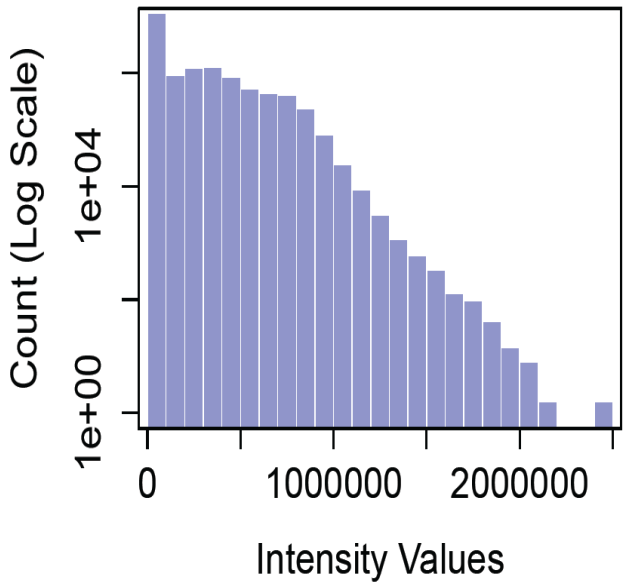
Transformations

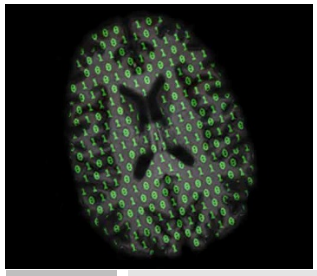
- Transformations
  - Allows to transform intensity values
  - Affects how light or dark regions of the brain appear, depending on their intensity value and the transfer function used



Log-Scale Histogram

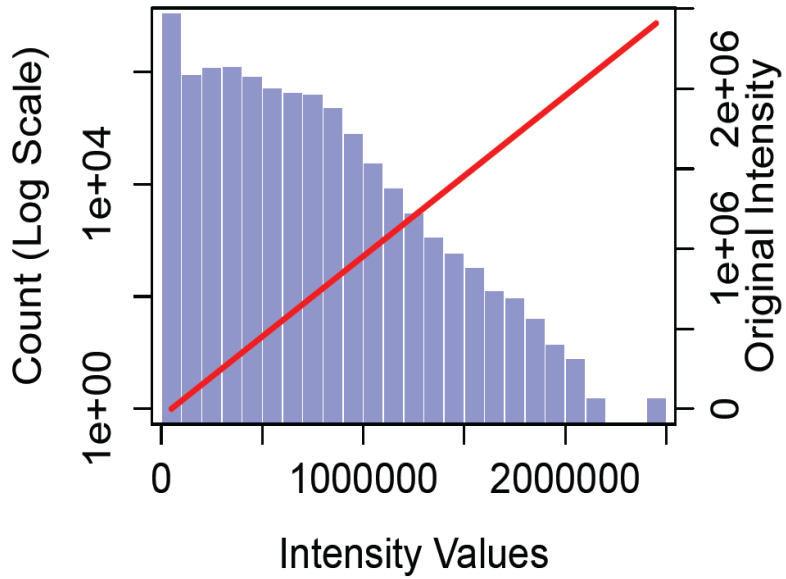
```
im_hist<-hist(T1,plot=FALSE)
par(mar = c(5, 4, 4, 4) + 0.3)
col1=rgb(0,0,1,1/2)
plot(im_hist$mids,im_hist
$count,log="y",type='h',lwd=10, lend=2,
col=col1,xlab="Intensity Values",ylab="Count
(Log Scale)" )
```

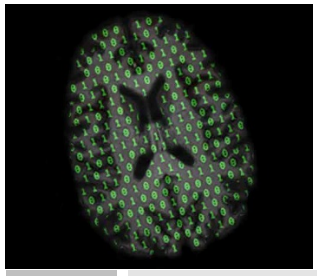




Log-Scale Histogram with Linear Transfer Function

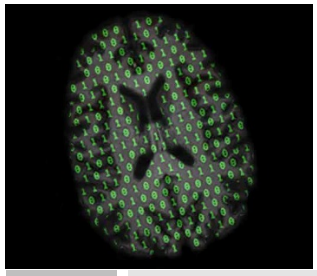
```
par(new = TRUE)
curve(x*1, axes = FALSE,xlab = "",ylab= "",
col=2, lwd=3)
axis(side=4,at = pretty(range(im_hist$mids))/
max(T1), labels=pretty(range(im_hist$mids)))
mtext("Original Intensity", side=4, line=2)
```





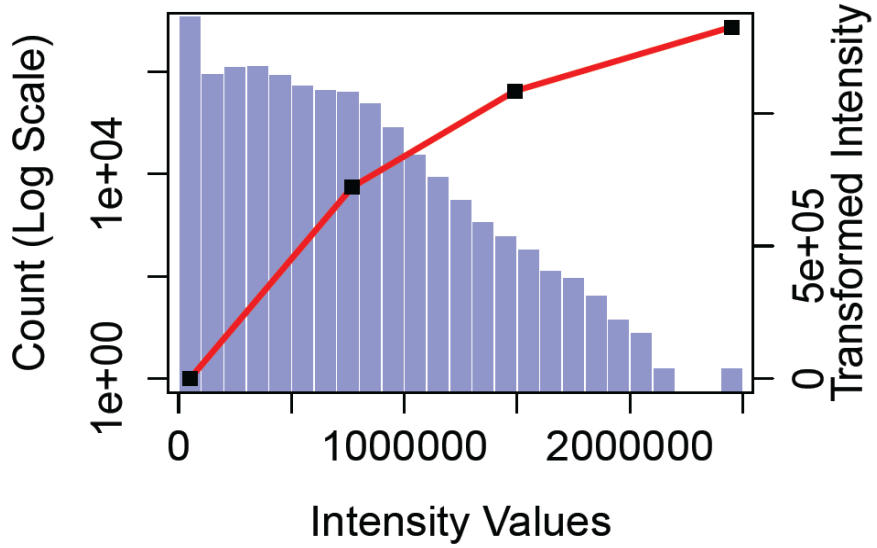
Define a different transfer function

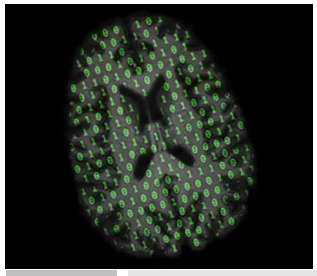
```
#This defines a linear spline. Other definitions are
possible
lin.sp<-function(x,knots,slope)
    {knots<-c(min(x),knots,max(x))
     slopeS<-slope[1]
     for(j in 2:length(slope)){slopeS<-c(slopeS,slope[j]-
sum(slopeS))}

     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)}

#Define a spline with two knots and three slopes
knot.vals<-c(.3,.6)
slp.vals<-c(1,.5,.25)
```



Plot the spline transfer function

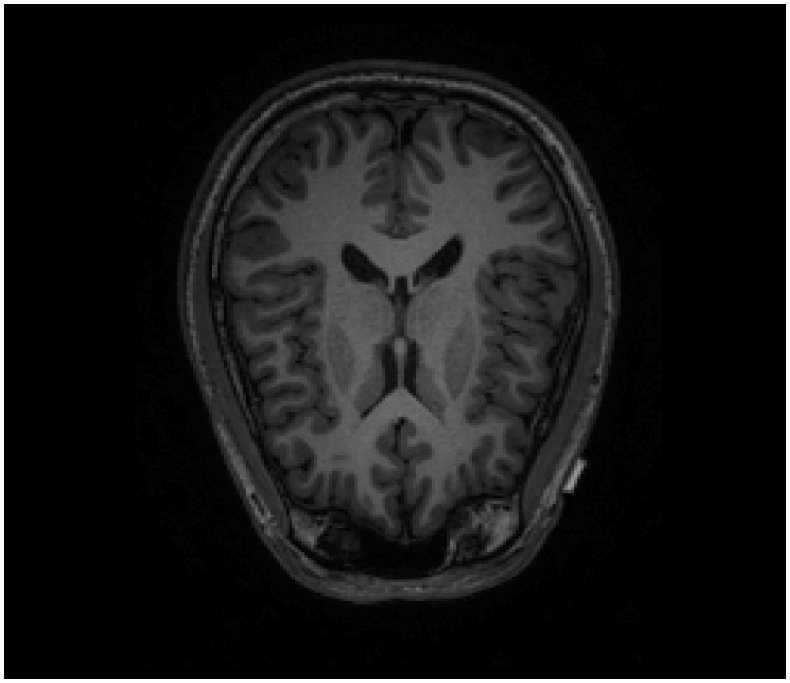
```
par(new = TRUE)
curve(lin.sp(x,knot.vals,slp.vals),axes=FALSE,xlab="",yl
ab="",col=2,lwd=3)
axis(side=4,at = pretty(range(im_hist$mids))/
max(T1),labels=pretty(range(im_hist$mids)))
mtext("Transformed Intensity", side=4, line=2)
```

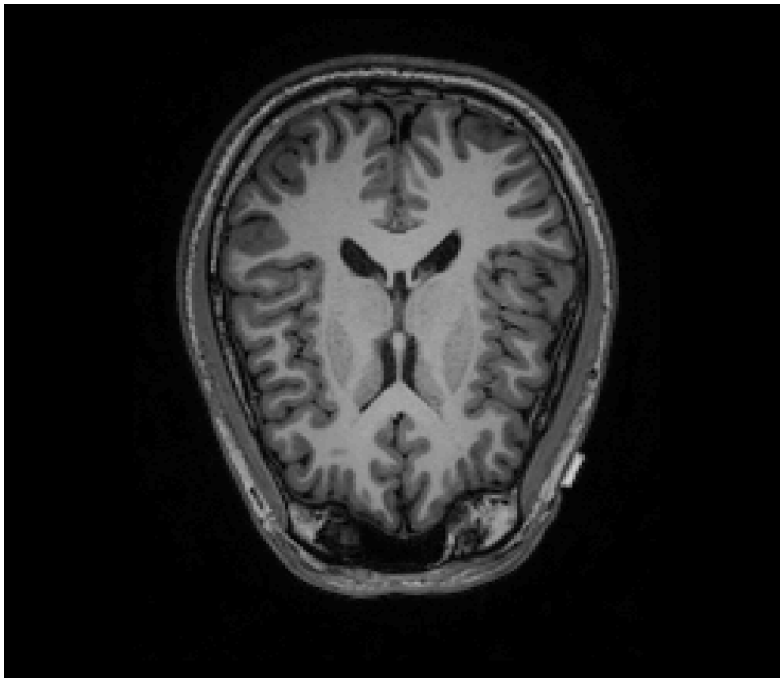


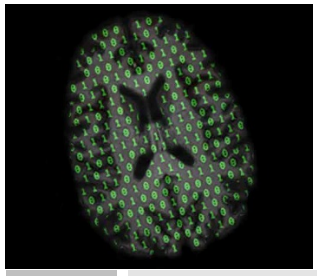


Plot the spline transfer function

```
trans_T1<-lin.sp(T1, knot.vals*max(T1), slp.vals)
image(T1,z=150,plot.type='single', main="Original
Image")
image(trans_T1,z=150,plot.type='single',
main="Transformed Image")
```

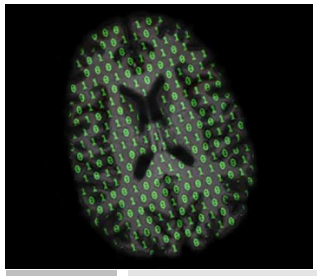






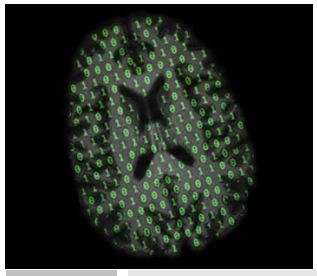
Some notes about the transfer function

- Knots rescaled to the scale of intensities: `knot.vals*max(T1)`
- The transfer function can be any function, not necessarily increasing
- Used for better
  - `visualization`
  - `prediction`
  - `input into standard software`



Smoothing

- `AnalyzeFMRI::GaussSmoothArray:`
  apply a Gaussian smooth to the image



Smoothing

Smooth the image with a Gaussian smoother (~ 1 minute)

```
smooth.T1 <- GaussSmoothArray(T1,
voxdim=c(1,1,1),
ksize=11,sigma=diag(3,3),mask=NULL,
var.norm=FALSE)
orthographic(smooth.T1)
```