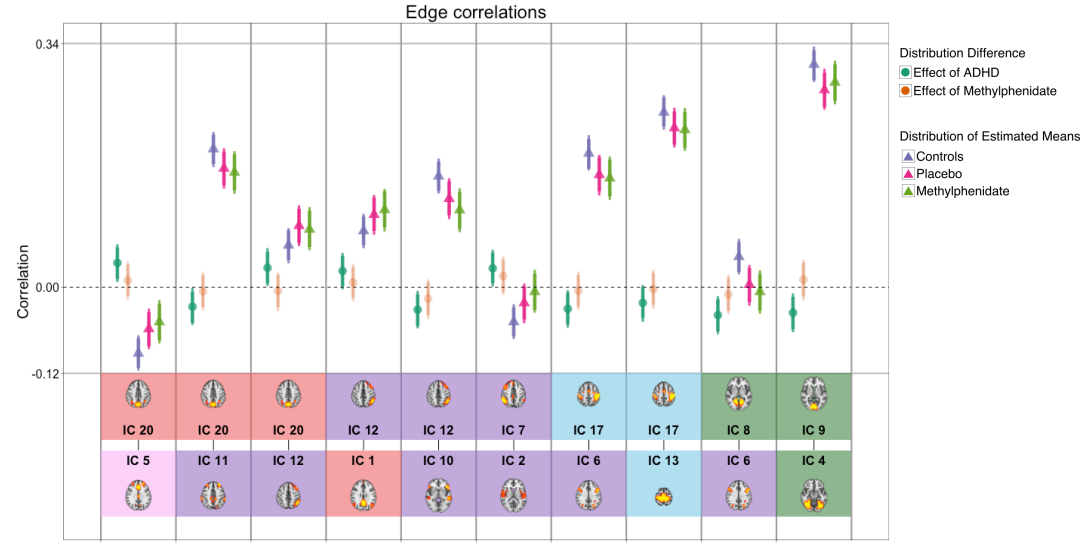
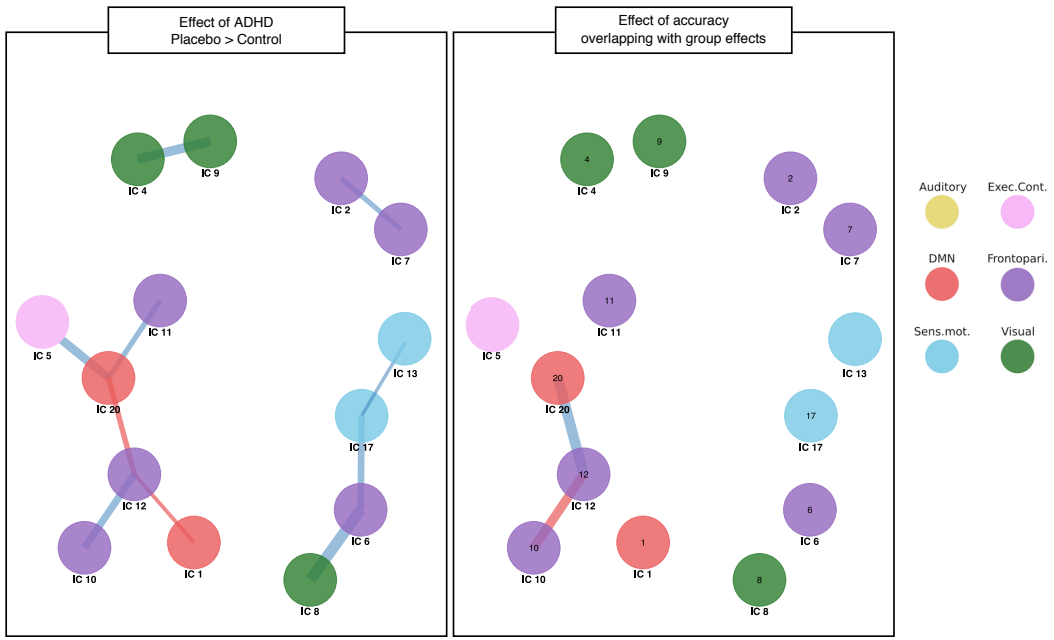
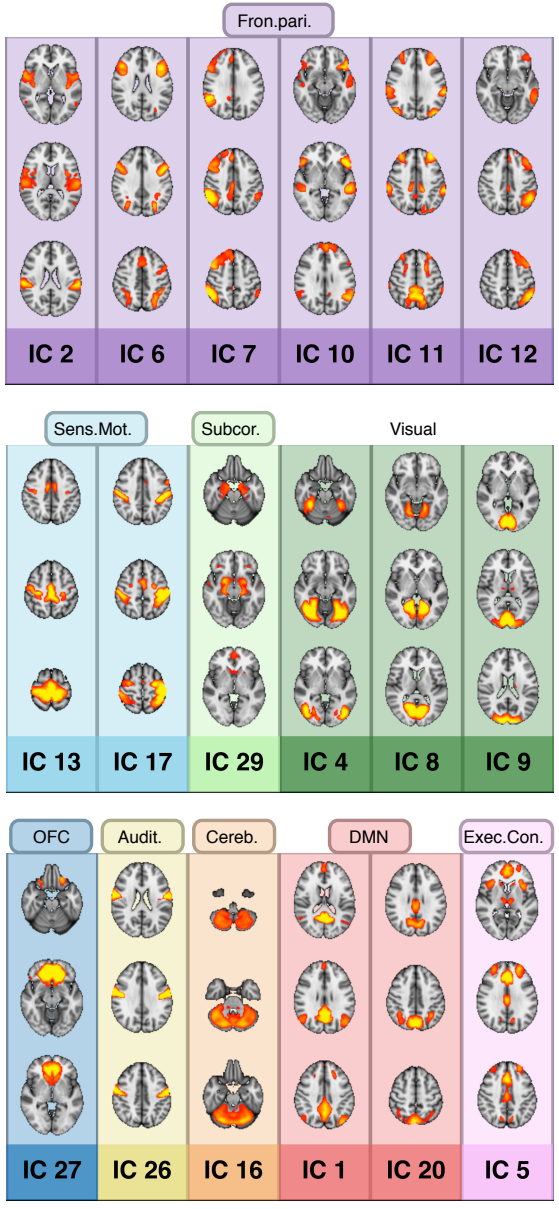
Adding external images to plots

Athanasia Mo Mowinckel

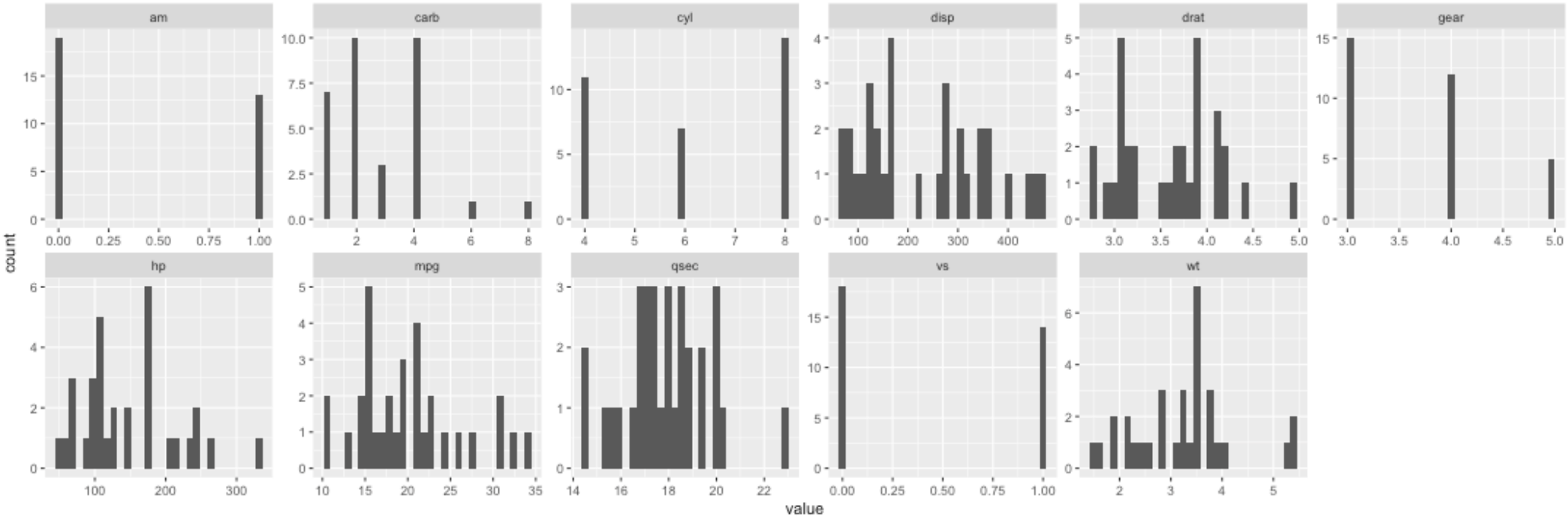
23.04.2018


Edge correlations
0.34
0.00
-0.12
Correlation
IC 20
IC 20
IC 20
IC 12
IC 12
IC 7
IC 17
IC 17
IC 8
IC 9
IC 5
IC 11
IC 12
IC 1
IC 10
IC 2
IC 6
IC 13
IC 6
IC 4
Distribution Difference
Effect of ADHD
Effect of Methylphenidate
Distribution of Estimated Means
Controls
Placebo
Methylphenidate


Effect of ADHD
Placebo > Control
Effect of accuracy
overlapping with group effects
IC 4
IC 9
IC 2
IC 7
IC 5
IC 11
IC 20
IC 13
IC 17
IC 12
IC 10
IC 1
IC 6
IC 8
Auditory
Exec.Cont.
DMN
Frontopari.
Sens.mot.
Visual


Fron.pari.
IC 2
IC 6
IC 7
IC 10
IC 11
IC 12
Sens.Mot.
Subcor.
Visual
IC 13
IC 17
IC 29
IC 4
IC 8
IC 9
OFC
Audit.
Cereb.
DMN
Exec.Con.
IC 27
IC 26
IC 16
IC 1
IC 20
IC 5

```
library(tidyverse)

mtcars %>% gather() %>%
  ggplot(aes(x=value)) + geom_histogram() + facet_wrap(~key, scales="free", nrow=2)
```



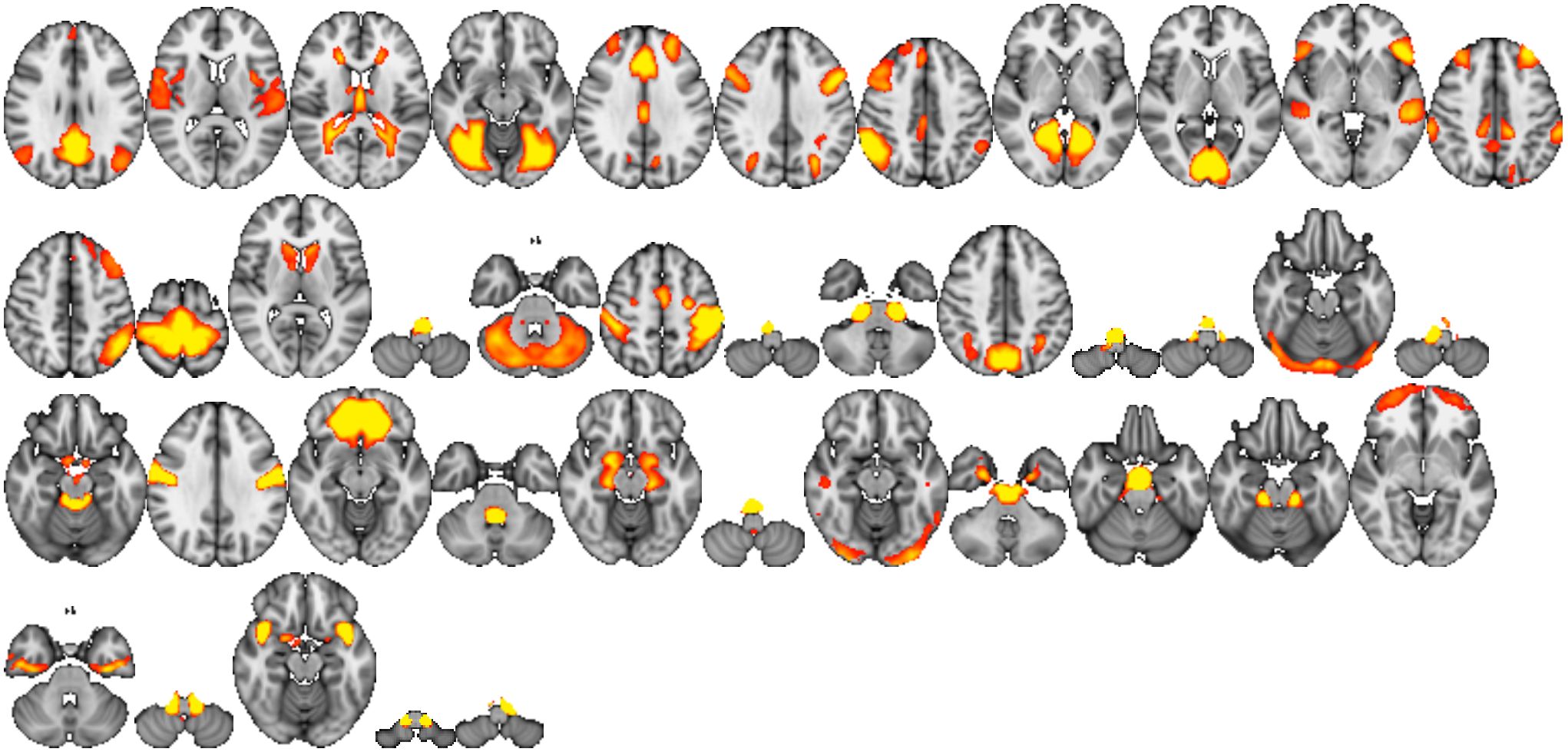
Plotting

When wanting to add external images to plots, there are several necessary steps:

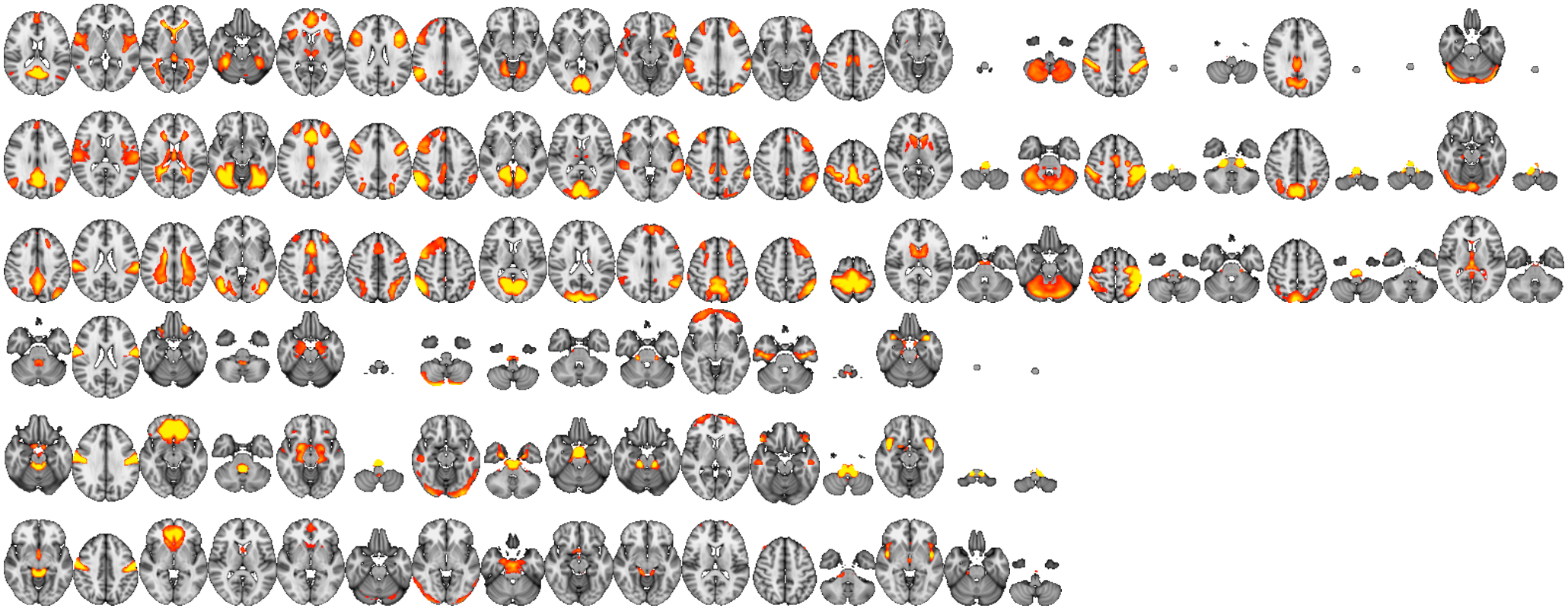
1. You need to create all the images you might want to add, place them in a folder, and give the images names (drop the whitespaces) that make sense according to the data you have.
2. The images **might** need to be processed to make plotting easier (i.e. them being the same size, backgrounds transparent etc.) I use ImageMagick for this, but there are R-packages to help, like imager
3. A clear idea of what you want to plot and why

- **how** comes with trial-and-error and perserverence (and hopefully with the help of these slides)

Single brain network images



Three images per brain network



Get images

Get file-paths for all the images

```
TRIPLE = list.files("img/three", full.names = T)
TRIPLE
```

```
##  [1] "img/three/0000_trans.png" "img/three/0001_trans.png"
##  [3] "img/three/0002_trans.png" "img/three/0003_trans.png"
##  [5] "img/three/0004_trans.png" "img/three/0005_trans.png"
##  [7] "img/three/0006_trans.png" "img/three/0007_trans.png"
##  [9] "img/three/0008_trans.png" "img/three/0009_trans.png"
## [11] "img/three/0010_trans.png" "img/three/0011_trans.png"
## [13] "img/three/0012_trans.png" "img/three/0013_trans.png"
## [15] "img/three/0014_trans.png" "img/three/0015_trans.png"
## [17] "img/three/0016_trans.png" "img/three/0017_trans.png"
## [19] "img/three/0018_trans.png" "img/three/0019_trans.png"
## [21] "img/three/0020_trans.png" "img/three/0021_trans.png"
## [23] "img/three/0022_trans.png" "img/three/0023_trans.png"
## [25] "img/three/0024_trans.png" "img/three/0025_trans.png"
## [27] "img/three/0026_trans.png" "img/three/0027_trans.png"
## [29] "img/three/0028_trans.png" "img/three/0029_trans.png"
## [31] "img/three/0030_trans.png" "img/three/0031_trans.png"
## [33] "img/three/0032_trans.png" "img/three/0033_trans.png"
## [35] "img/three/0034_trans.png" "img/three/0035_trans.png"
## [37] "img/three/0036_trans.png" "img/three/0037_trans.png"
## [39] "img/three/0038_trans.png" "img/three/0039_trans.png"
```

Strip the images of charachters and any non-numeric characters

```
NAMES = gsub("[a-zA-Z]|[[:punct:]]","",TRIPLE)
NAMES
```

```
##  [1] "0000" "0001" "0002" "0003" "0004" "0005" "0006" "0007" "0008" "0009"
## [11] "0010" "0011" "0012" "0013" "0014" "0015" "0016" "0017" "0018" "0019"
## [21] "0020" "0021" "0022" "0023" "0024" "0025" "0026" "0027" "0028" "0029"
## [31] "0030" "0031" "0032" "0033" "0034" "0035" "0036" "0037" "0038" "0039"
```

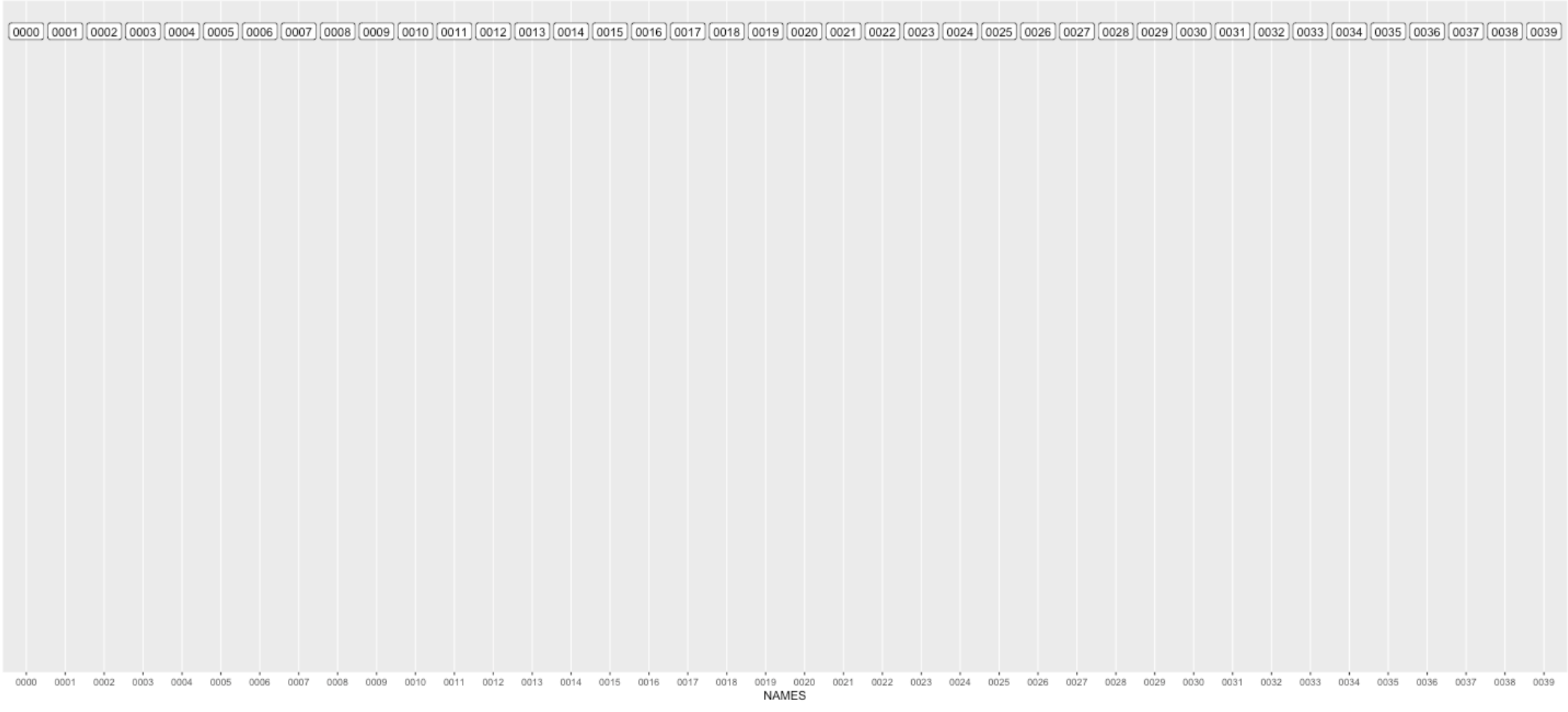
Create a data.frame by combining the two vectors, ggplot will be happy for this!

```
DATA = cbind(TRIPLE,NAMES) %>% as.data.frame(stringsAsFactors=F)
DATA %>% glimpse()
```

```
## Observations: 40
## Variables: 2
## $ TRIPLE <chr> "img/three/0000_trans.png", "img/three/0001_trans.png",...
## $ NAMES  <chr> "0000", "0001", "0002", "0003", "0004", "0005", "0006",...
```

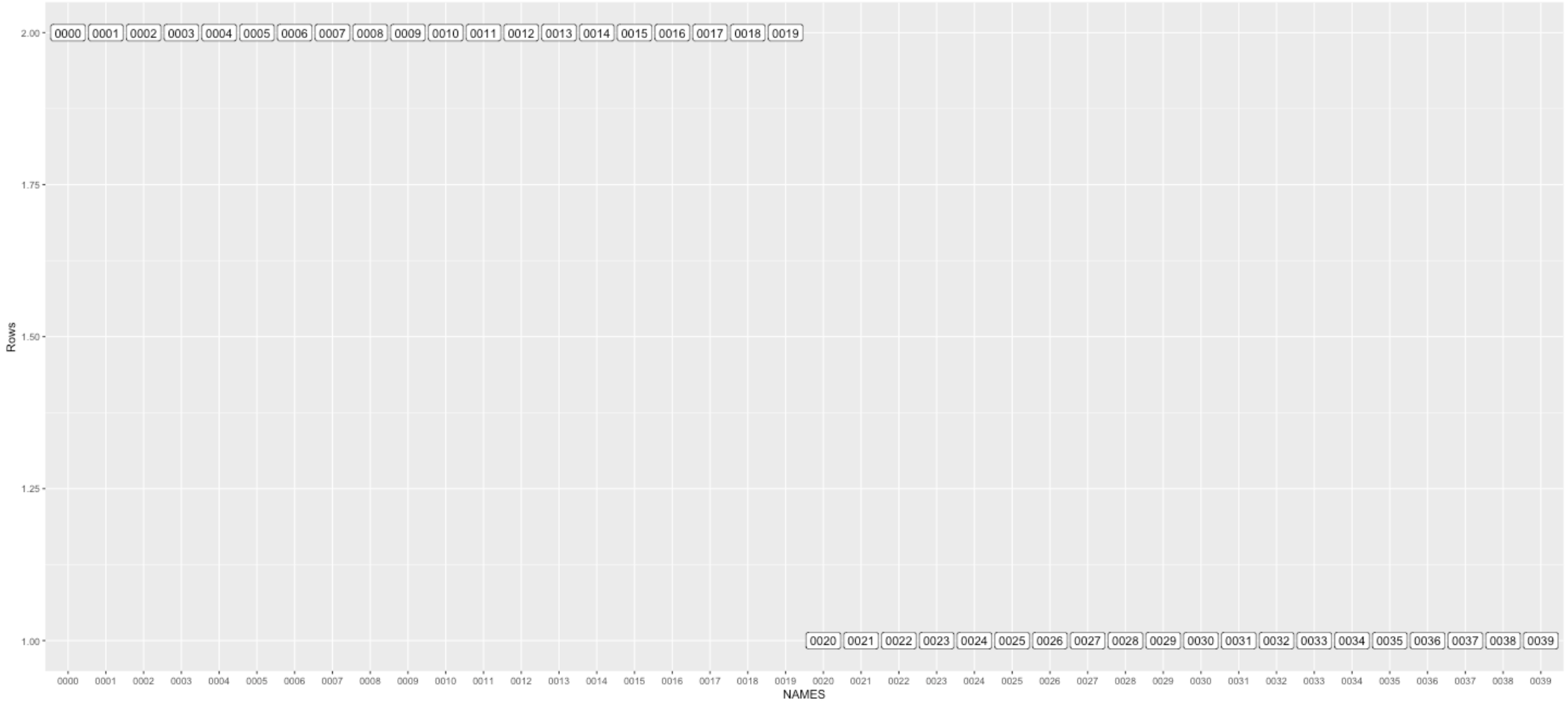
Let's do a very simple plot of the names of the files.

```
ggplot(DATA) + geom_label(aes(x=NAMES), y=1, label=NAMES)
```



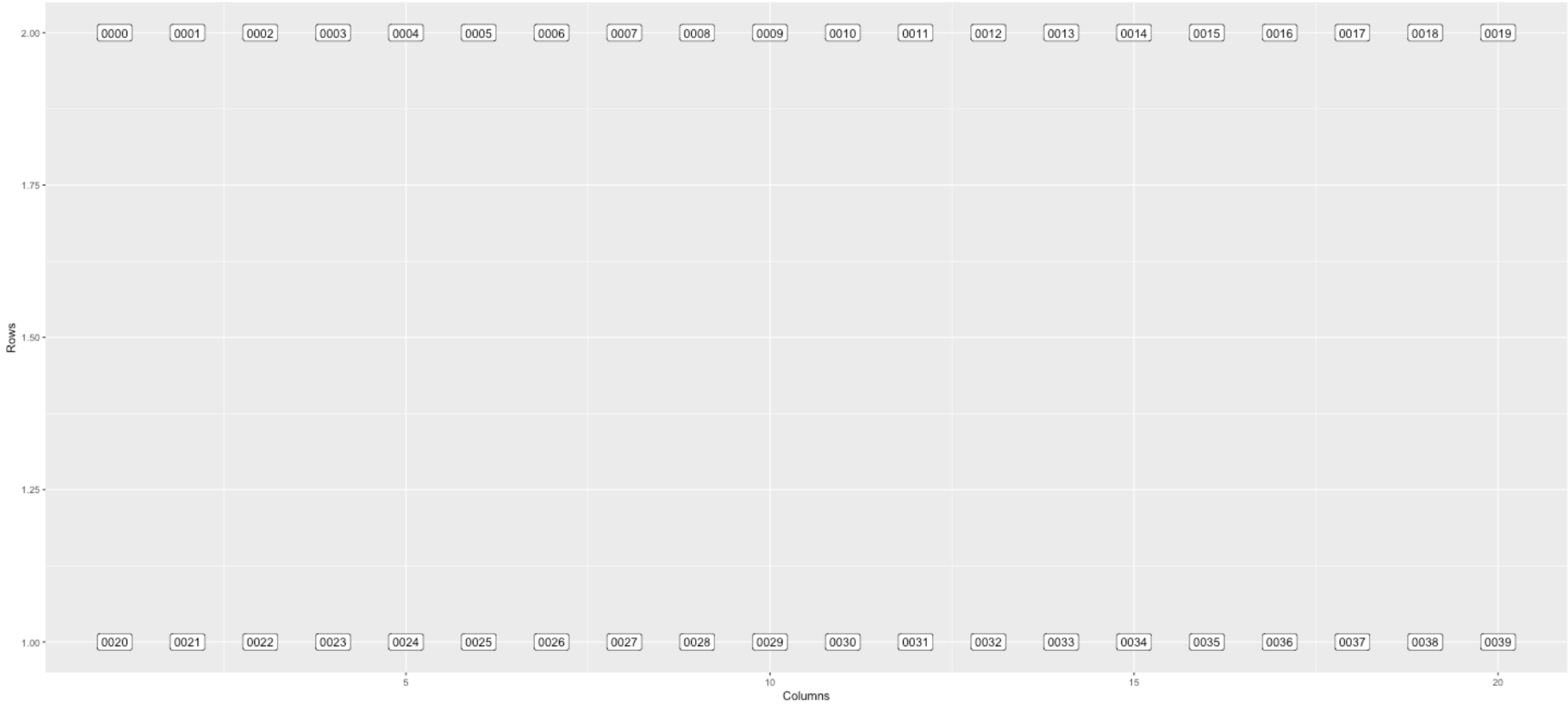
Let's try getting them over several "rows"

```
DATA$Rows = c(rep(2,20), rep(1,20))
ggplot(DATA) + geom_label(aes(x=NAMES, y=Rows), label=NAMES)
```



Ops! We need to adjust the x-coordinates too!

```
DATA$Columns = c(1:20, 1:20)
ggplot(DATA) + geom_label(aes(x=Columns, y=Rows), label=NAMES)
```



Let's have a look at how to get an image into the plot.
Get necessary packages

```
library(png); library(grid)
```

Read in the image

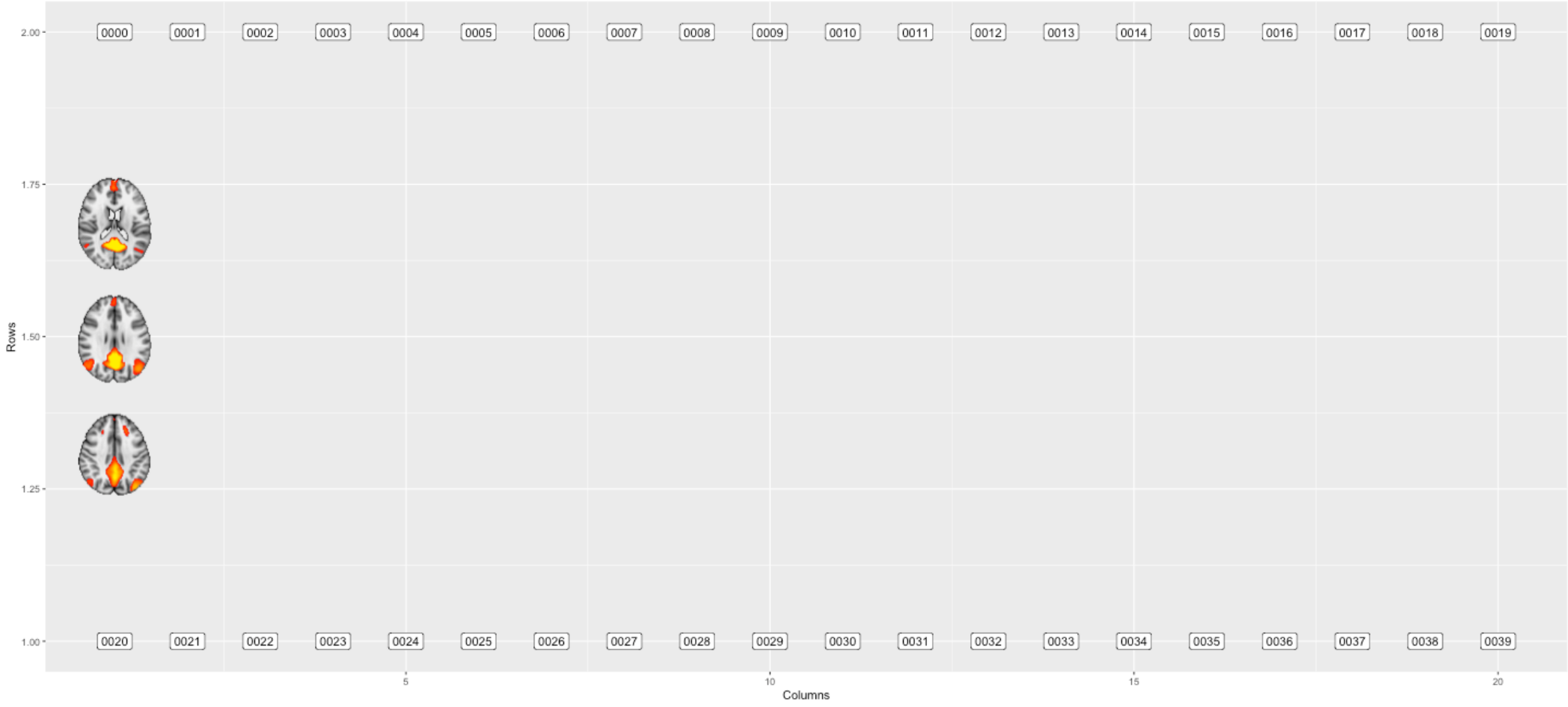
```
img = readPNG(DATA$TRIPLE[1])
```

And transform it into a graphics object (grob)

```
g =  rasterGrob(img, interpolate=TRUE)
```

Then we add it to the plot!

```
ggplot(DATA) +
  geom_label(aes(x=Columns, y=Rows), label=NAMES) +
   annotation_custom(grob = g, ymin = 1,  ymax =2, xmin = .5, xmax = 1.5)
```



To add all the images, we need to loop through the list, and add them incrementally, annotations don't take `aes()` in ggplot.

This requires a couple of steps:

- save the main plot to a global variable

```
PLOT = ggplot(DATA) +
  geom_label(aes(x=Columns, y=Rows), label=NAMES)
```

- create a global variable where the `grob`'s of all the images may be placed.

```
g = list()
```

- create a loop where all the grobs are placed in `g` and are incrementally added to the plot

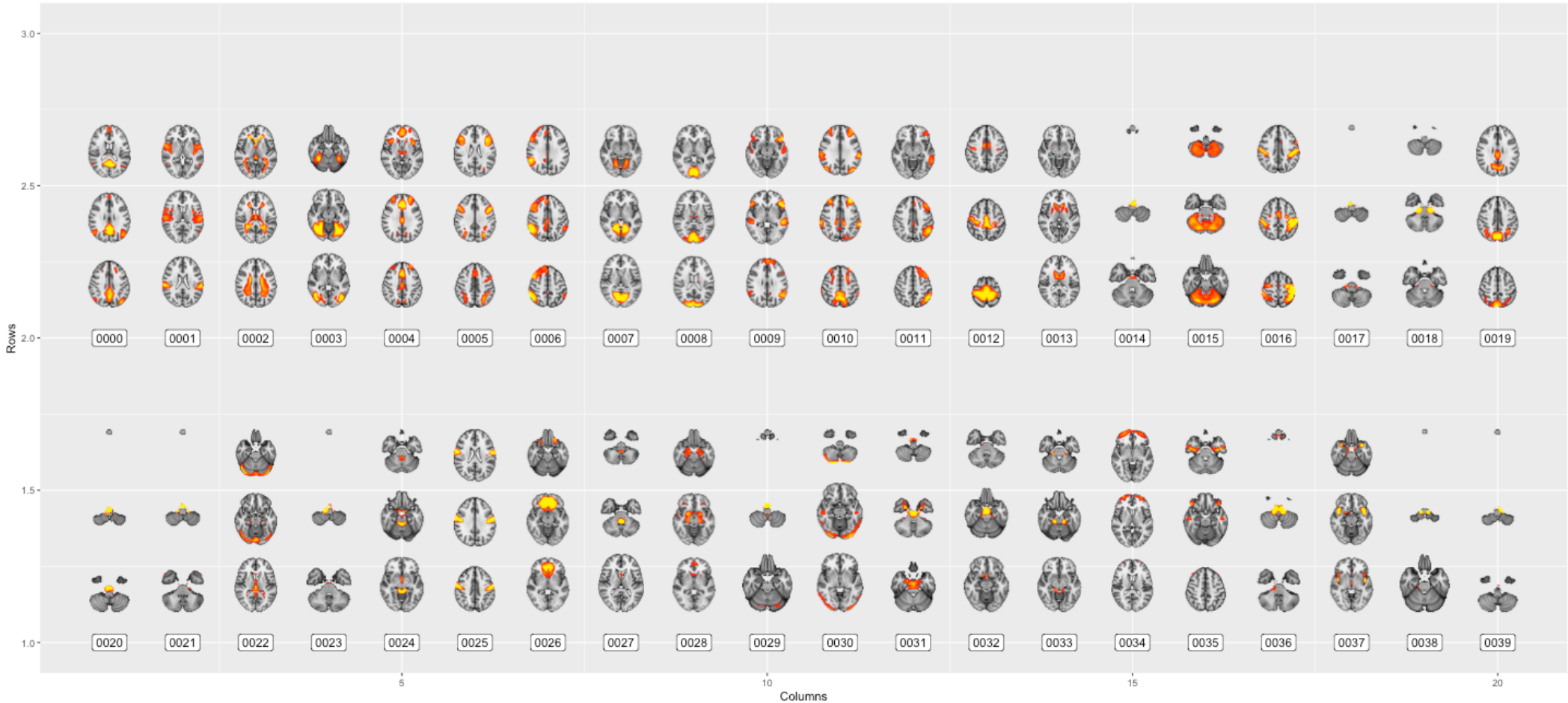
```
for(i in 1:nrow(DATA)){
  img = readPNG(DATA$TRIPLE[i])
  g[[i]] = rasterGrob(img, interpolate=TRUE)

  PLOT = PLOT +
     annotation_custom(
       grob= g[[i]],
       ymin = DATA$Rows[i]+.1,  ymax = DATA$Rows[i]+.7,
       xmin = DATA$Columns[i]-.5, xmax = DATA$Columns[i] +.5)

}
```

Let' have a look at that.

PL0T

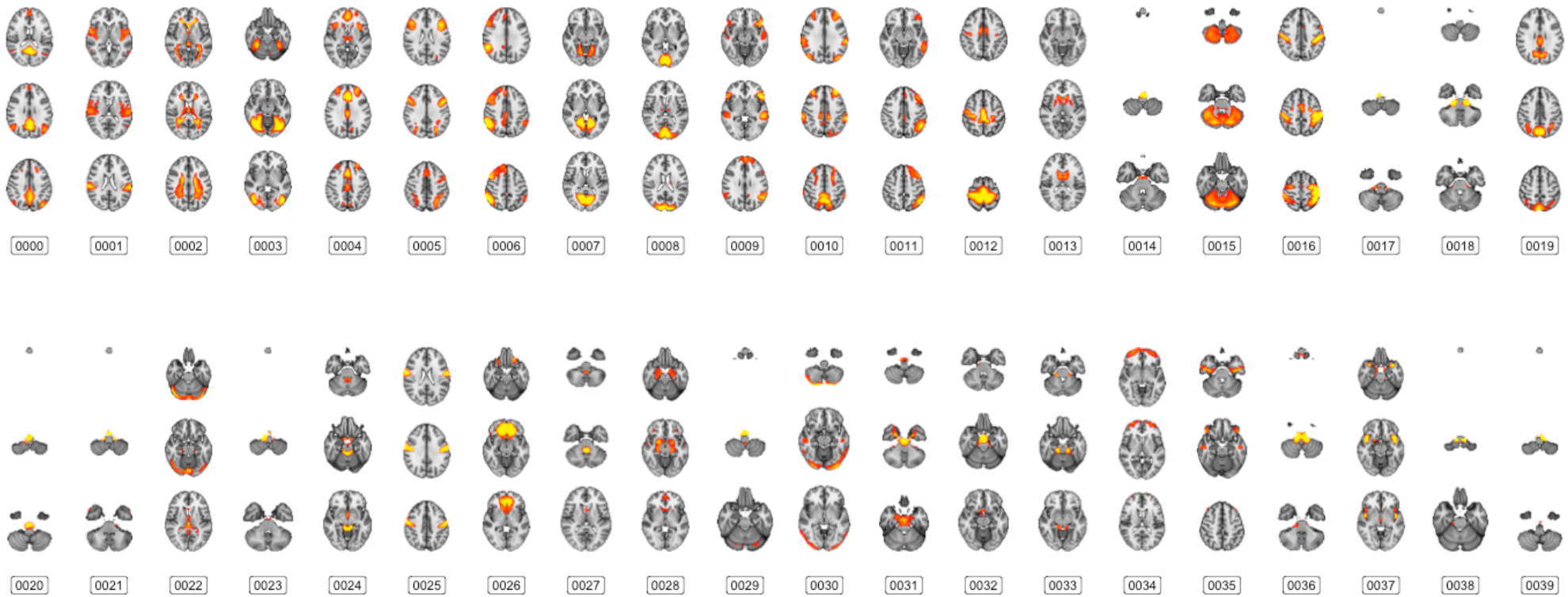
Our top row is "missing". Because `ggplot` sets the plotting area through the first command `geom_label` all the images in the top row is outside the area. We can fix this by chaging the scale!

```
PLOT + scale_y_continuous(limits=c(1,3))
```



Since this isn't really a plot, it's an overview of the images, we may want to strip the plotting area of unecessary information, like axes etc.

```
PLOT + scale_y_continuous(limits=c(1,3)) + theme_void()
```



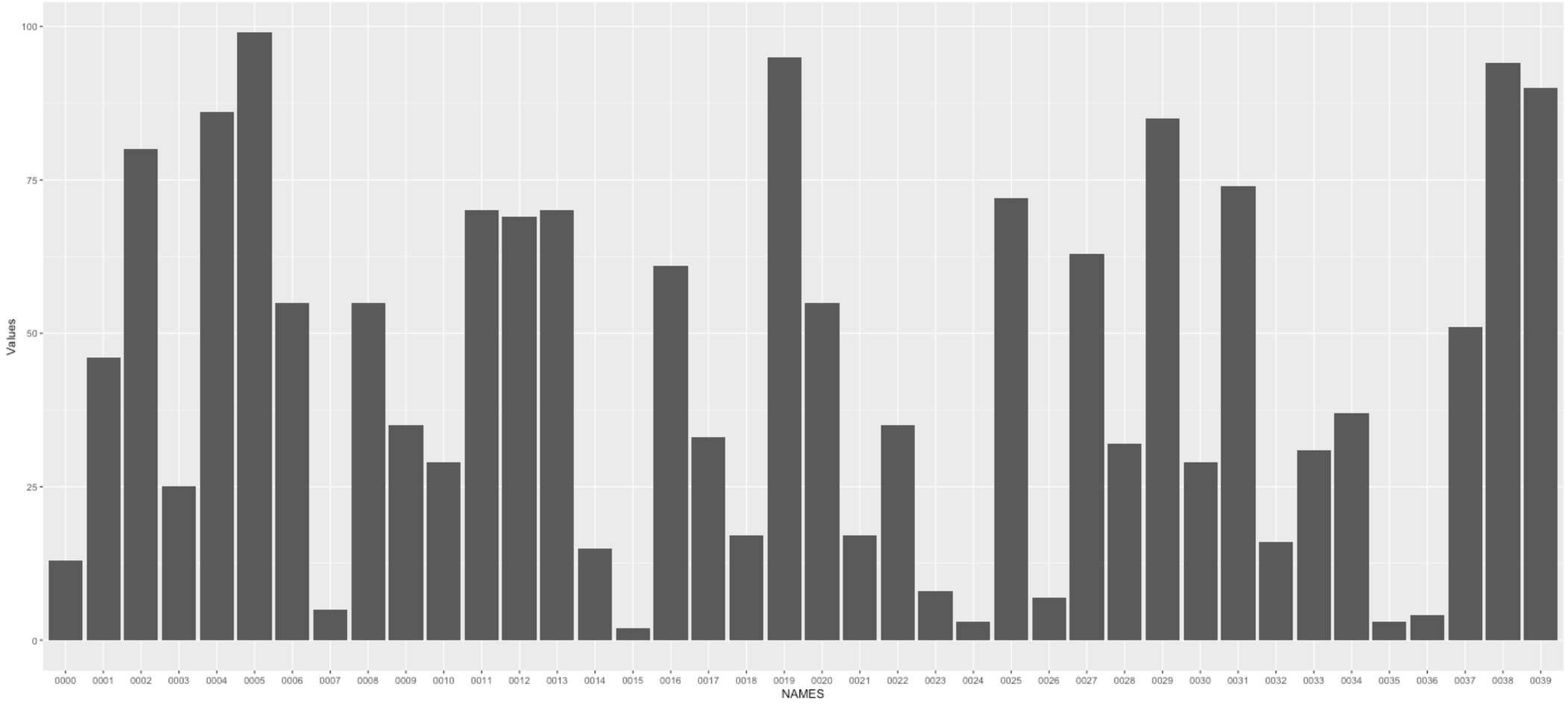
We've successfully added external images to a plot.

Well done, us!

But not a "true" plot, let's give a barchart a go!

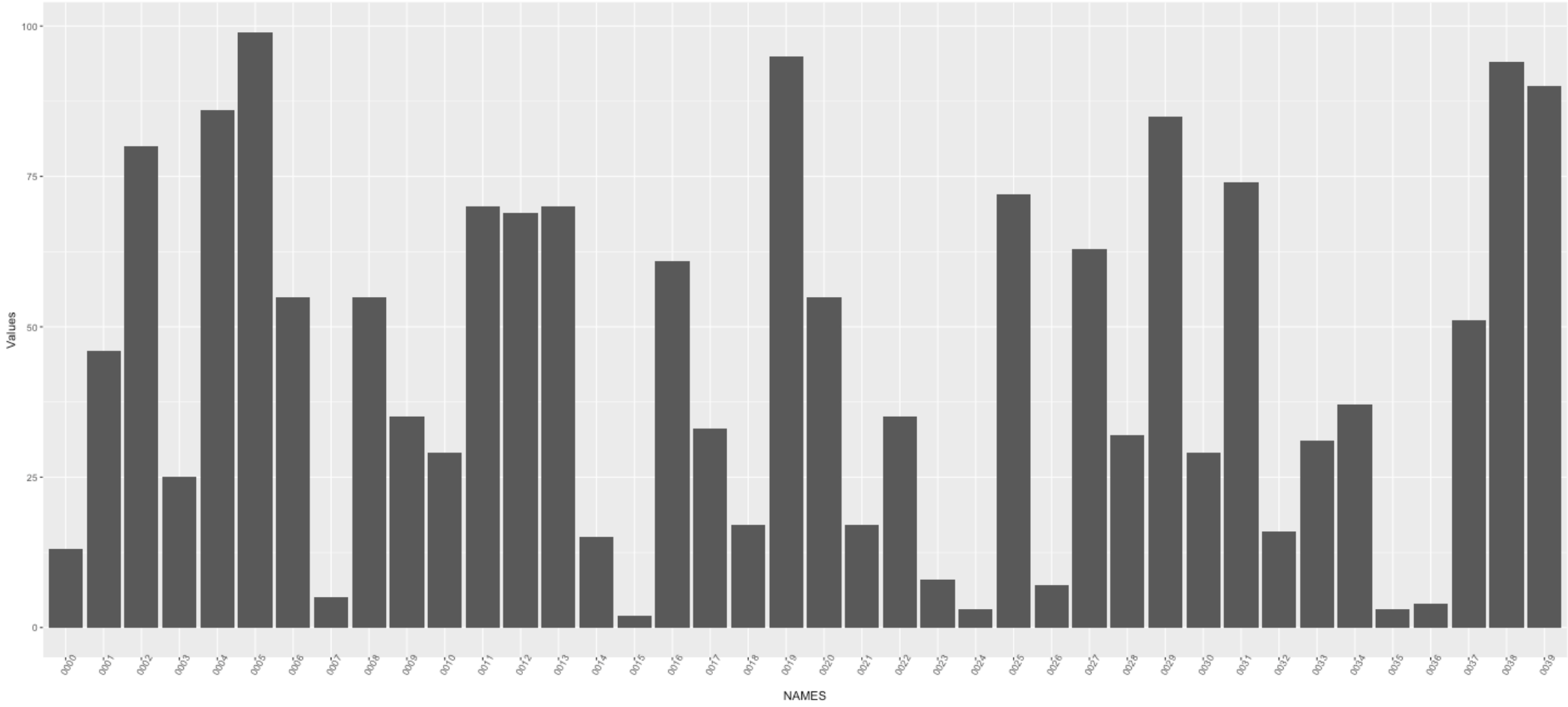
To make a barchart, let's just assing some random values to the various images, and plot them like that.

```
DATA$Values = sample(1:100, size=nrow(DATA), replace=T)

ggplot(DATA) + geom_histogram(aes(x=NAMES, y=Values), stat="identity")
```



First thing's first, the labels are impossible to read like this. Let's rotate them a little.

```
ggplot(DATA) + geom_histogram(aes(x=NAMES, y=Values), stat="identity") +
  theme(axis.text.x = element_text(angle=60))
```



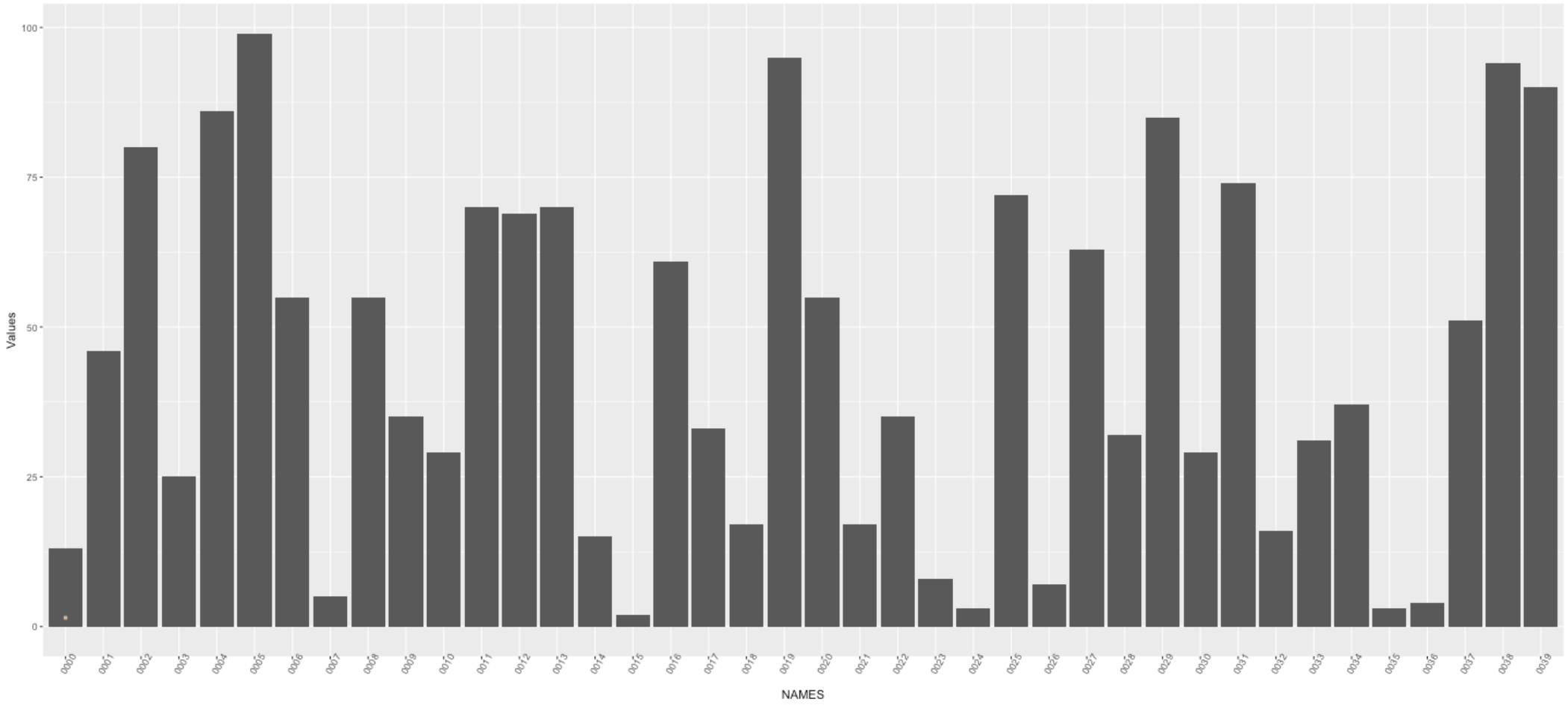
I'm happy with that for now. Now, I'd like to add the brain images at the bottom, to help readers remember what labels goes with which brain. Let's start with a single one, and adjust the plot accordingly.

Let's read in the file paths for all the images in `img/single`, and create a grob from the first image, just like before.

```
PLOT = ggplot(DATA) + geom_histogram(aes(x=NAMES, y=Values), stat="identity") +
  theme(axis.text.x = element_text(angle=60))

DATA$SINGLES = list.files("img/single", full.names = T)

img = readPNG(DATA$SINGLES[1])
g = rasterGrob(img, interpolate = T)


PLOT = PLOT +
  annotation_custom(grob = g, ymin = 1,  ymax = 2,
                    xmin = .5, xmax = 1.5)
```

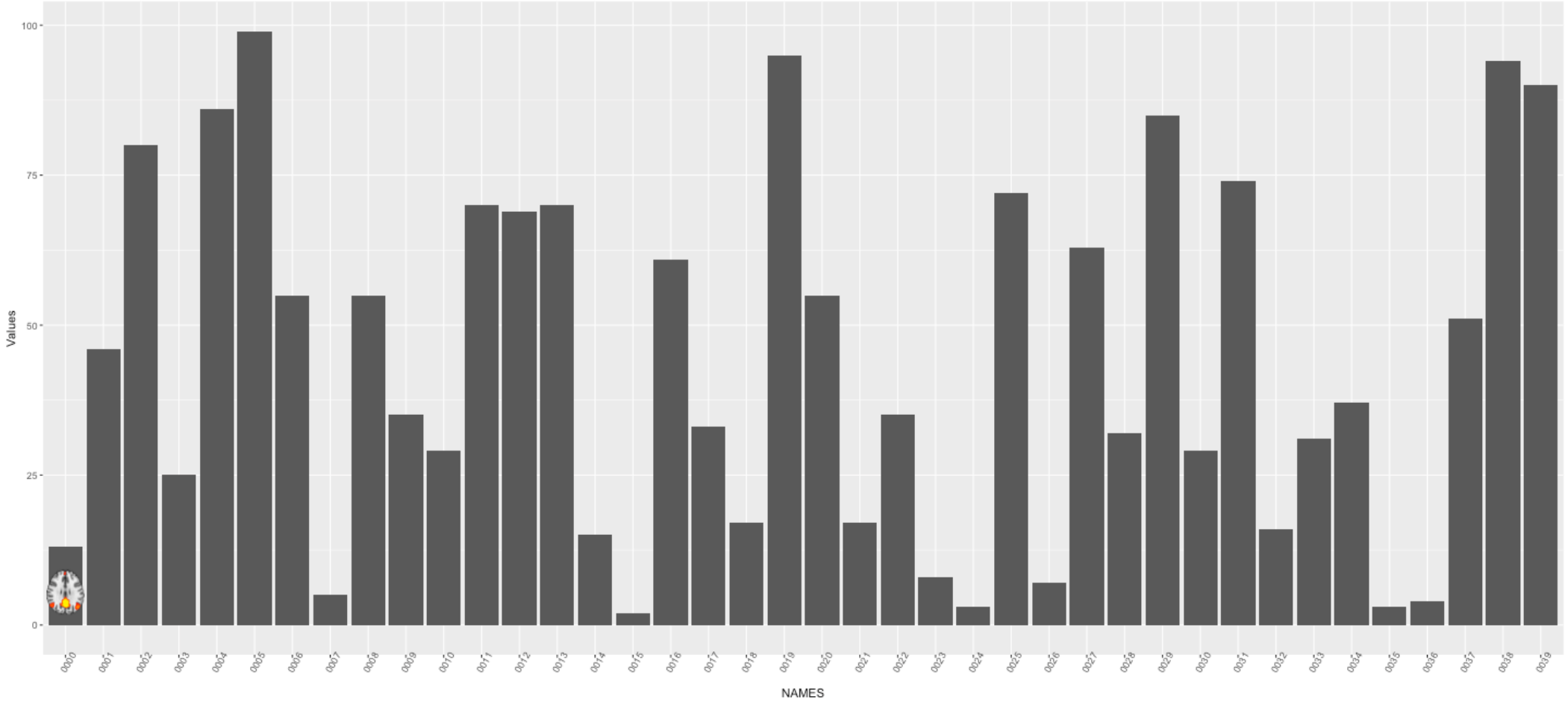
Let's have a look at that.

PL0T



Do you see it? It's that teeny tiny speck at the bottom of the first column. Well, we need to get that sorted for sure, the size is completely off.

```
ggplot(DATA) + geom_histogram(aes(x=NAMES, y=Values), stat="identity") +
  theme(axis.text.x = element_text(angle=60)) +
  annotation_custom(grob = g, ymin = 1,  ymax = 10, xmin = .5, xmax = 1.5)
```

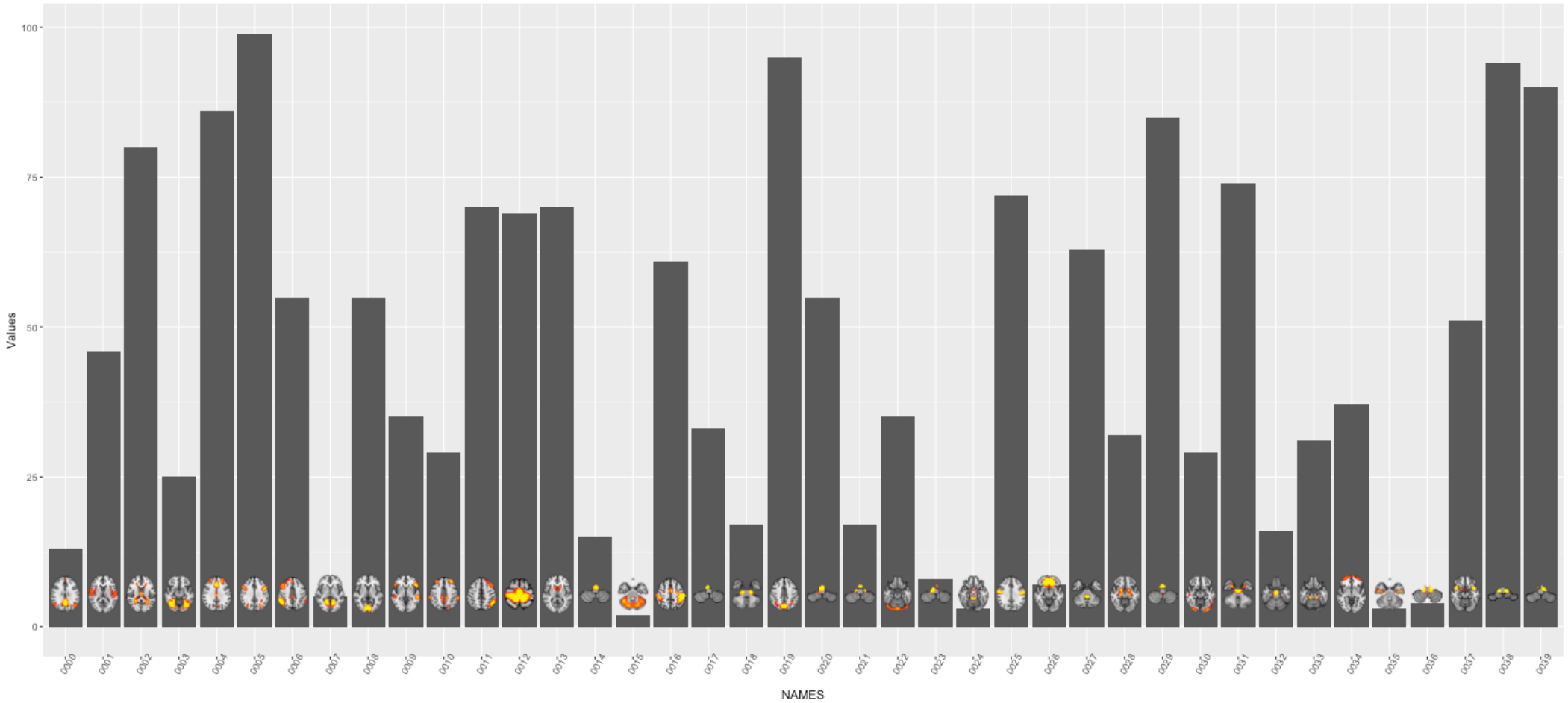


Much better! Let's chuck them all in there, with a loop like last time. Since we also want everything on one row (compared to two before), we add a column with a counter before we start.

```
DATA$N = seq(1,nrow(DATA))
PLOT = ggplot(DATA) + geom_histogram(aes(x=NAMES, y=Values), stat="identity") + theme(axis.text.x = element_text(ang

g = list()
for(i in 1:nrow(DATA)){
  img = readPNG(DATA$SINGLES[i])
  g[[i]] = rasterGrob(img, interpolate = T)

  PLOT = PLOT +
    annotation_custom(
      grob= g[[i]],
      ymin = 1,  ymax = 10,
      xmin = DATA$N[i]-.4, xmax = DATA$N[i] +.4)
}
```

Let's have a look at that.

PL0T

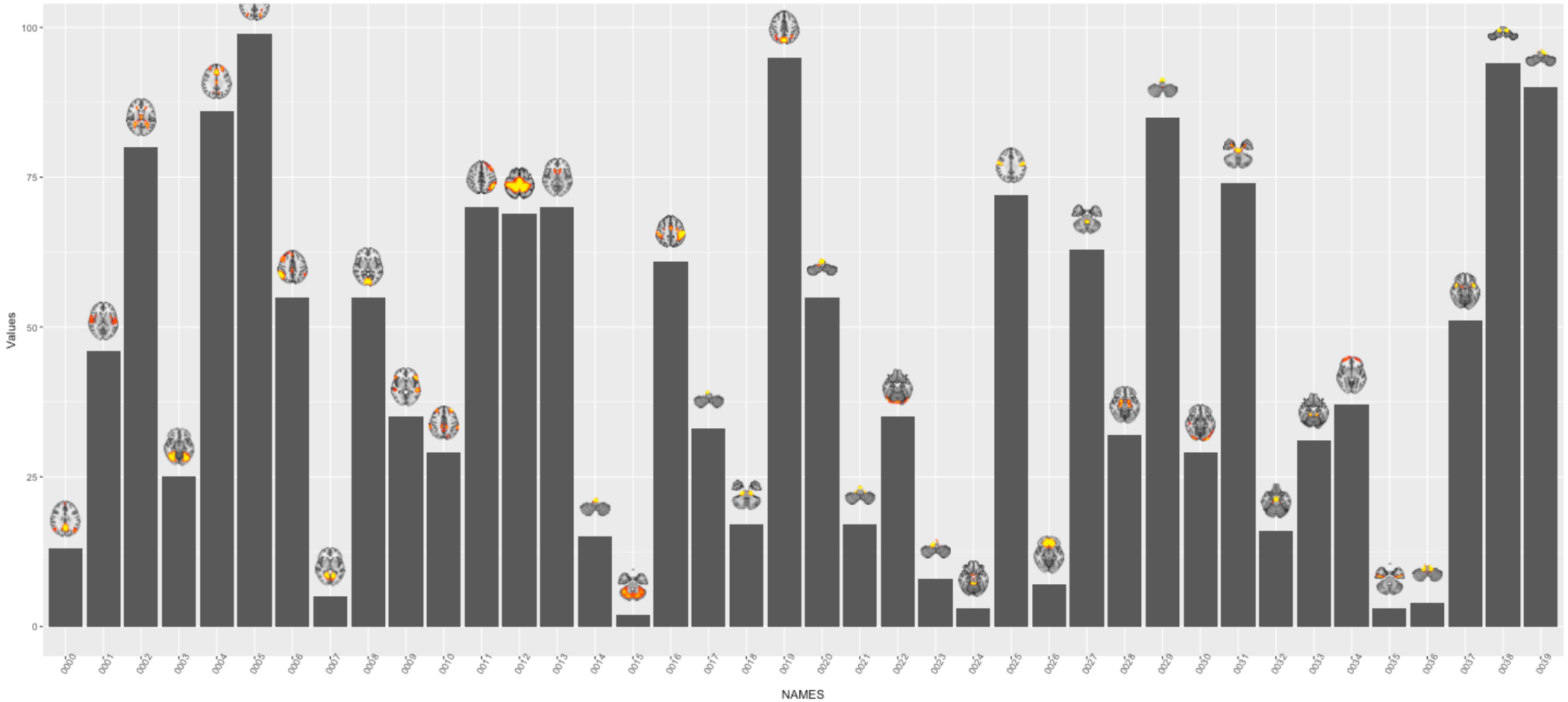


Right'o! They're all there. But their placement is a little inconventient. We can put them at the top of their bars, if we want.

```
PLOT = ggplot(DATA) + geom_histogram(aes(x=NAMES, y=Values), stat="identity") + theme(axis.text.x = element_text(ang

g = list()
for(i in 1:nrow(DATA)){
  img = readPNG(DATA$SINGLES[i])
  g[[i]] = rasterGrob(img, interpolate = T)

  PLOT = PLOT +
    annotation_custom(
      grob= g[[i]],
      ymin = DATA$Values[i],  ymax = DATA$Values[i]+10,
      xmin = DATA$N[i]-.4, xmax = DATA$N[i] +.4)
}
```

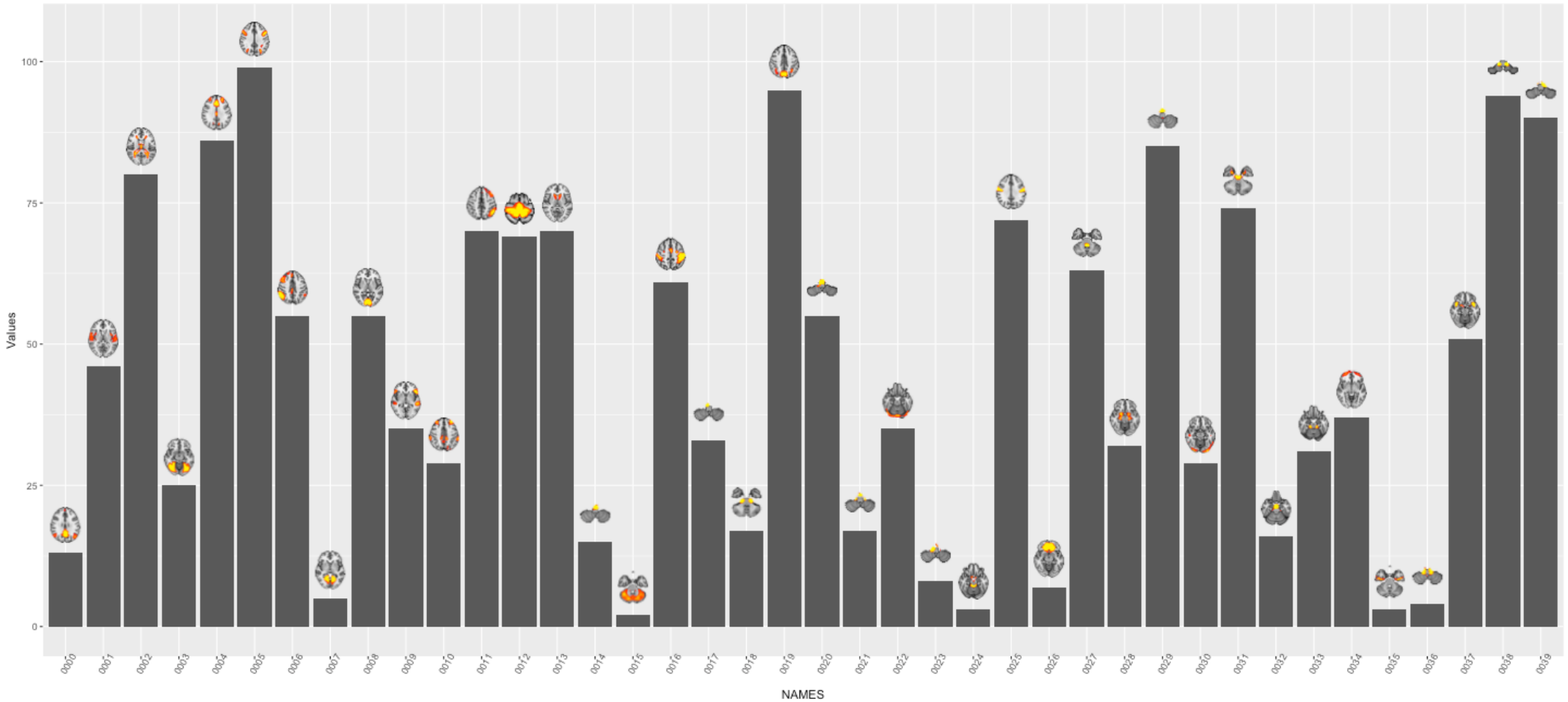
Let's have a look at that.

PL0T



Some of them end up outside the chart limits, but we can solve that again.

```
PLOT + scale_y_continuous(limits=c(0,105))
```

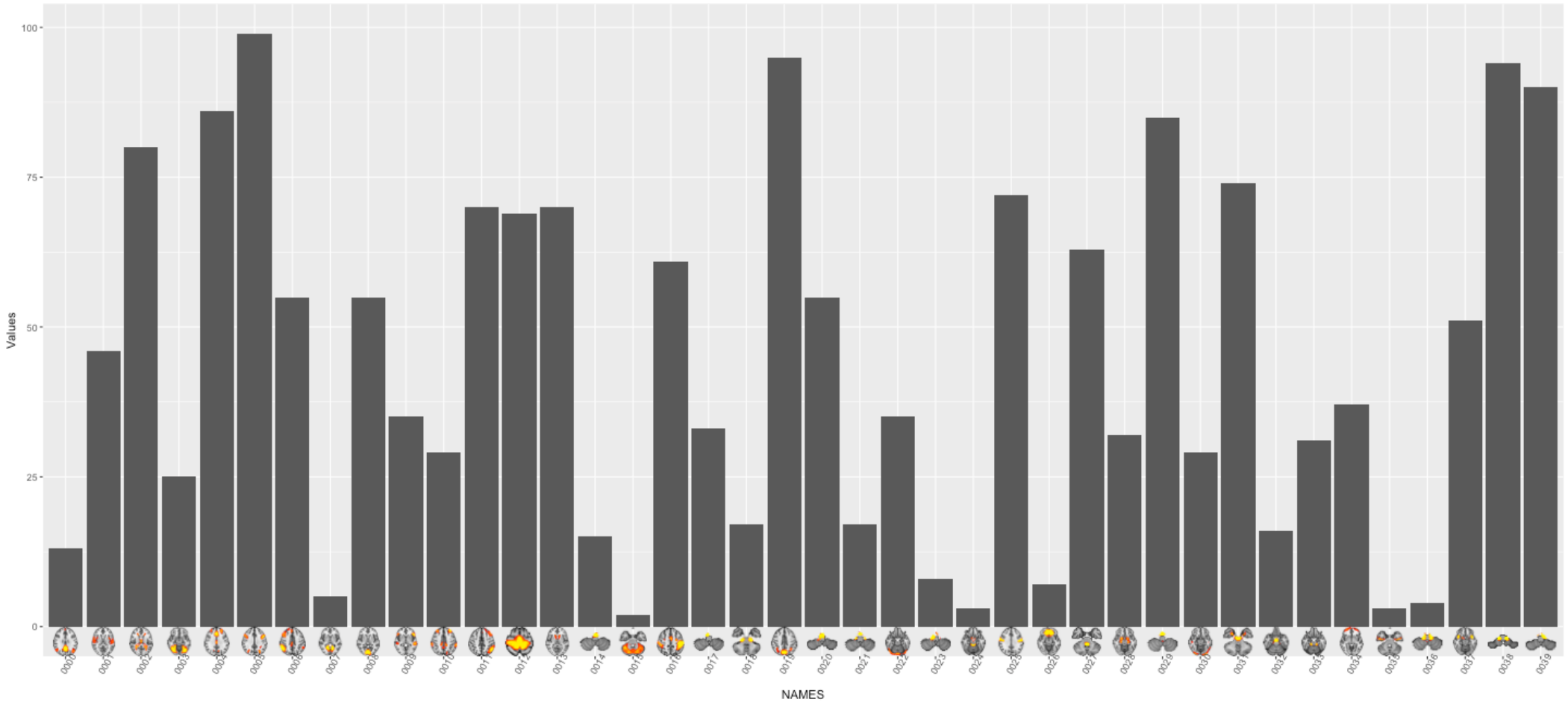


We can also place them below the bars.

```
PLOT = ggplot(DATA) + geom_histogram(aes(x=NAMES, y=Values), stat="identity") + theme(axis.text.x = element_text(ang

g = list()
for(i in 1:nrow(DATA)){
  img = readPNG(DATA$SINGLES[i])
  g[[i]] = rasterGrob(img, interpolate = T)

  PLOT = PLOT +
    annotation_custom(
      grob= g[[i]],
      ymin = -Inf,  ymax = 0,
      xmin = DATA$N[i]-.4, xmax = DATA$N[i] +.4)
}
```

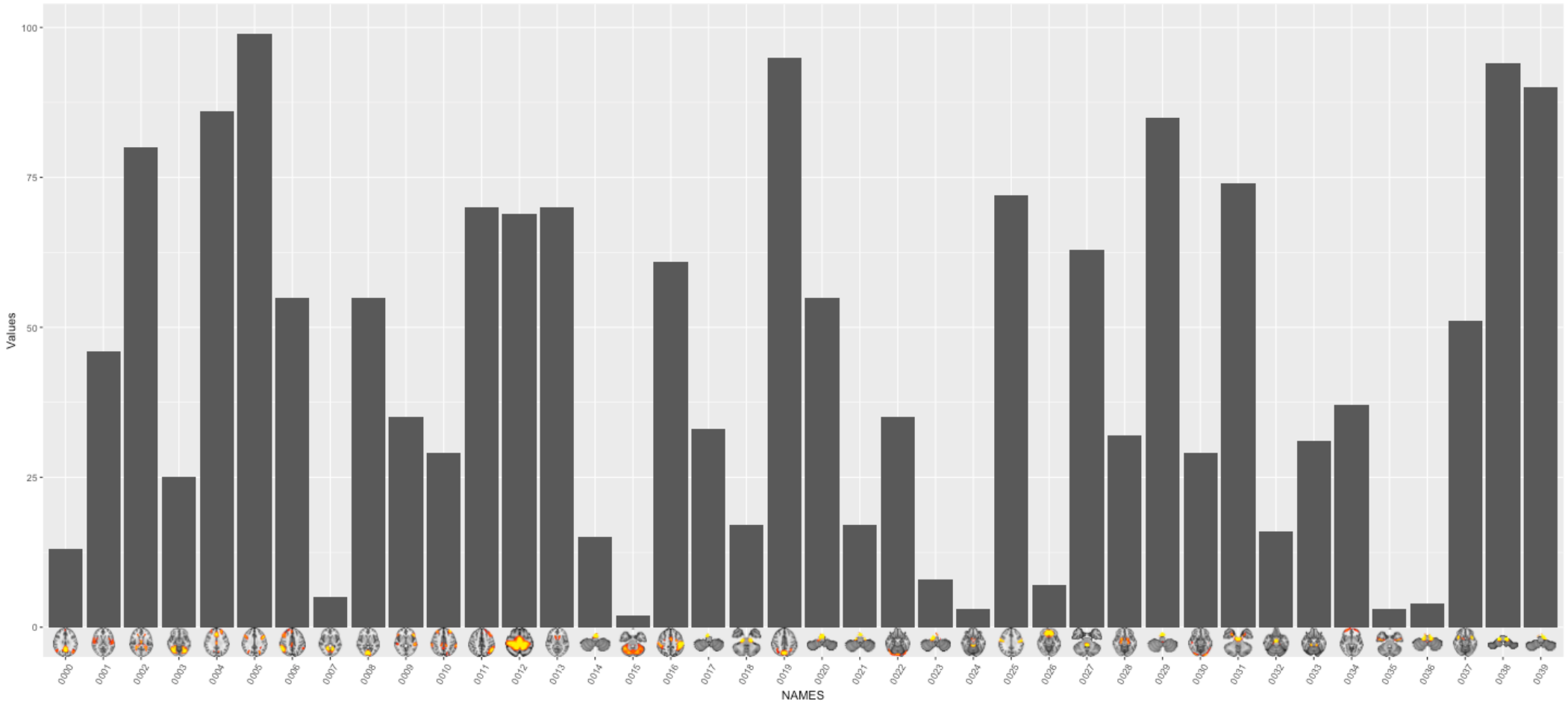
Let's have a look at that.

PL0T



The labels are crashing with the images! We can adjust them to appear slightly further down with `vjust`.

```
PLOT + theme(axis.text.x = element_text(angle=60, vjust=.5))
```



The images are very small. We might consider only to plot the bars and corresponding images that have larger bars. The will provide more space.

We will need to subset the data a little first, keeping only data value above 25.

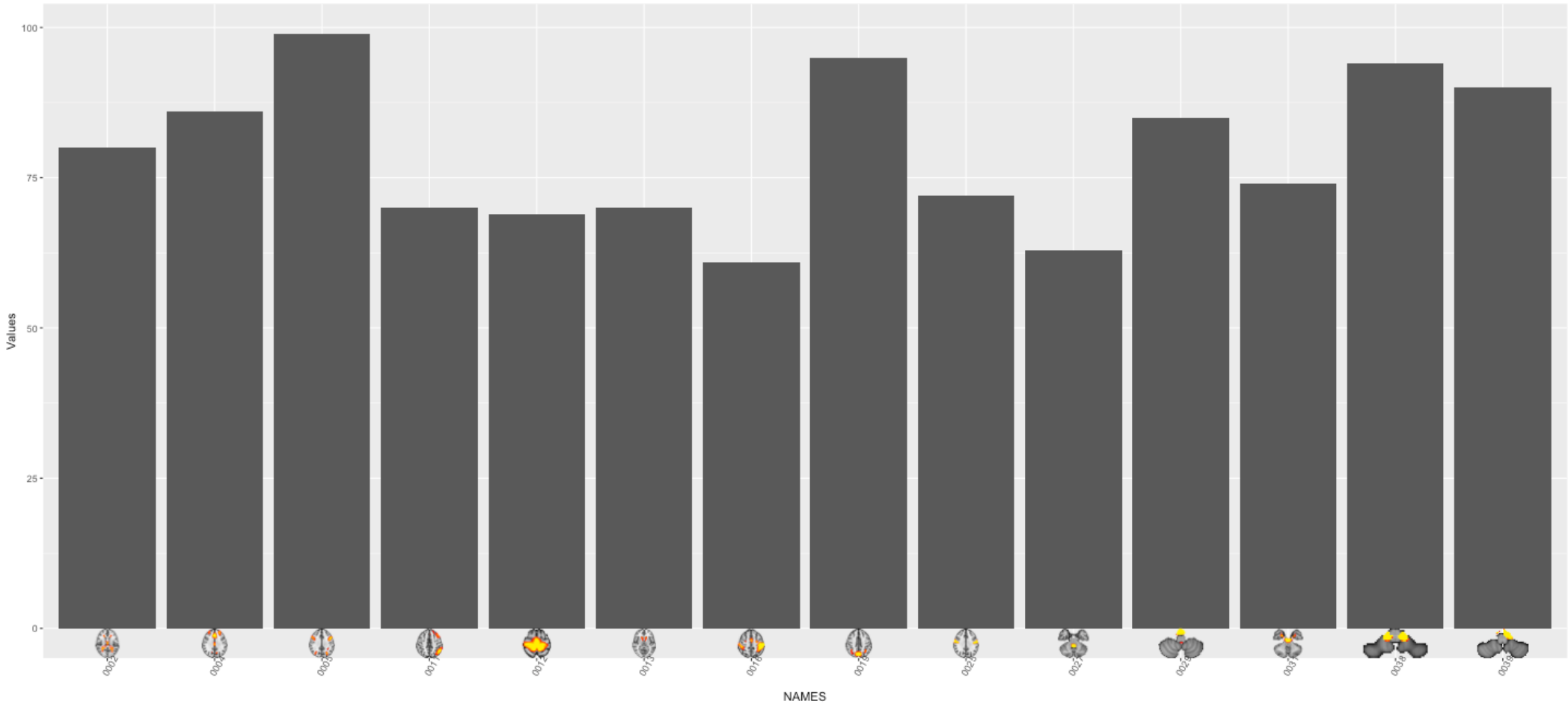
```
DATA_sub = DATA %>% filter(Values > 60)
```

Then we do the same as before, just with the new data

```
PLOT = ggplot(DATA_sub) + geom_histogram(aes(x=NAMES, y=Values), stat="identity") + theme(axis.text.x = element_text

g = list()
for(i in 1:nrow(DATA_sub)){
  img = readPNG(DATA_sub$SINGLES[i])
  g[[i]] = rasterGrob(img, interpolate = T)

  PLOT = PLOT +
    annotation_custom(
      grob= g[[i]],
      ymin = -Inf,  ymax = 0,
      xmin = DATA$N[i]-.4, xmax = DATA$N[i] +.4)
}
```

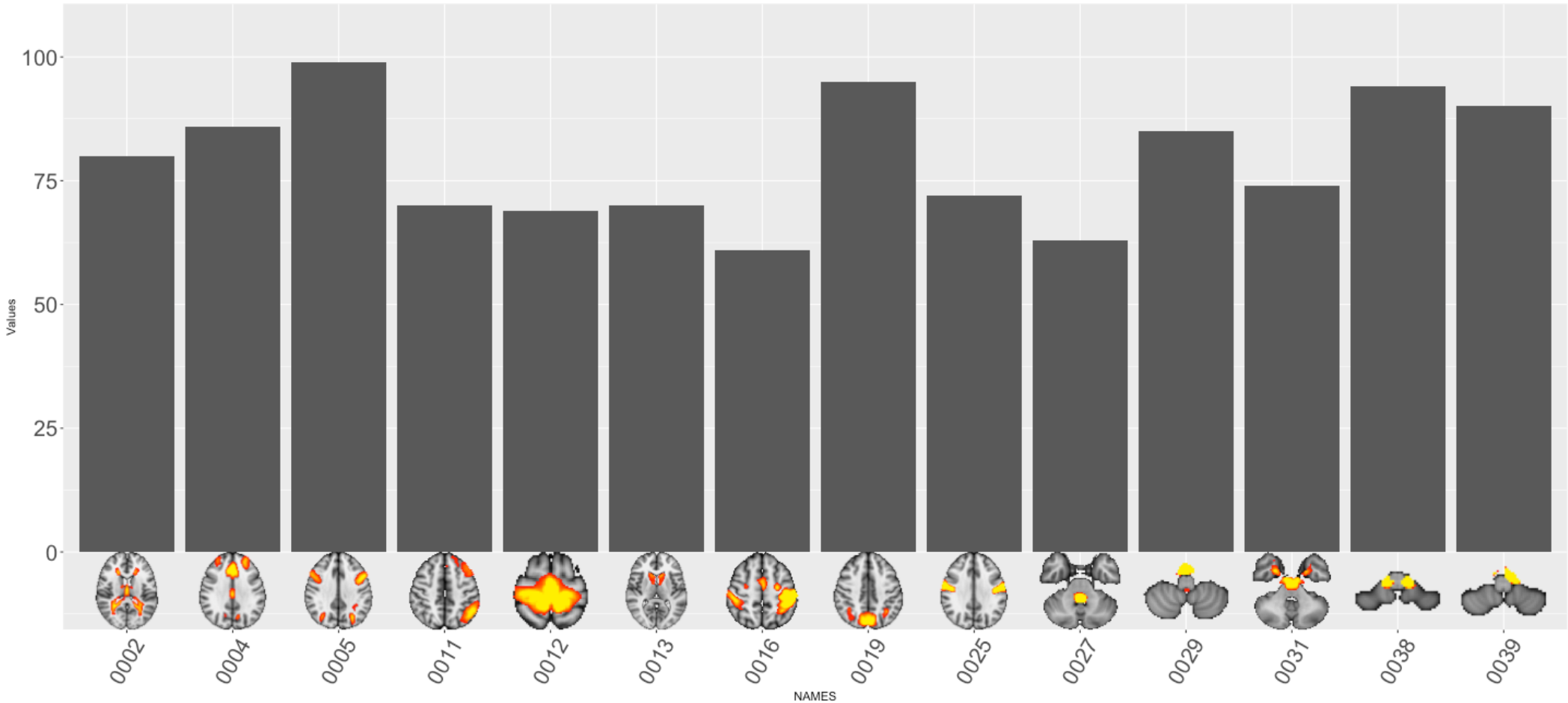
Let's have a look at that.

PL0T



Great! Now we need to adjust the image-sizes, and we only need to fix the scale!

```
PLOT + scale_y_continuous(limits=c(-10,105)) + theme(axis.text.x = element_text(angle=60, vjust=.5), axis.text = ele
```



There you go!

A small intro to adding images directly to your plots!