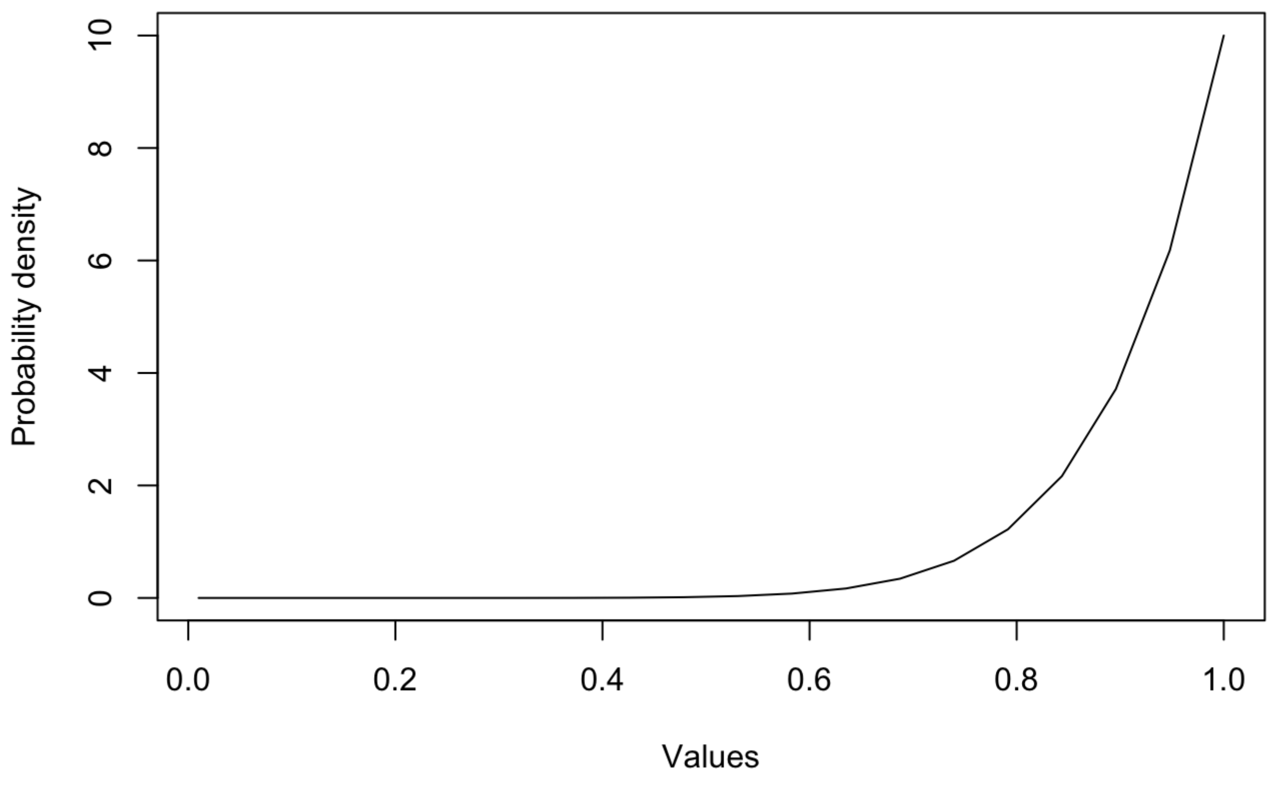
Figure 1A

This is an additional script aiming to illustrate beta distribution in 2D or 3D as a series of lines.

#by L.Bystrykh Feb 2022.

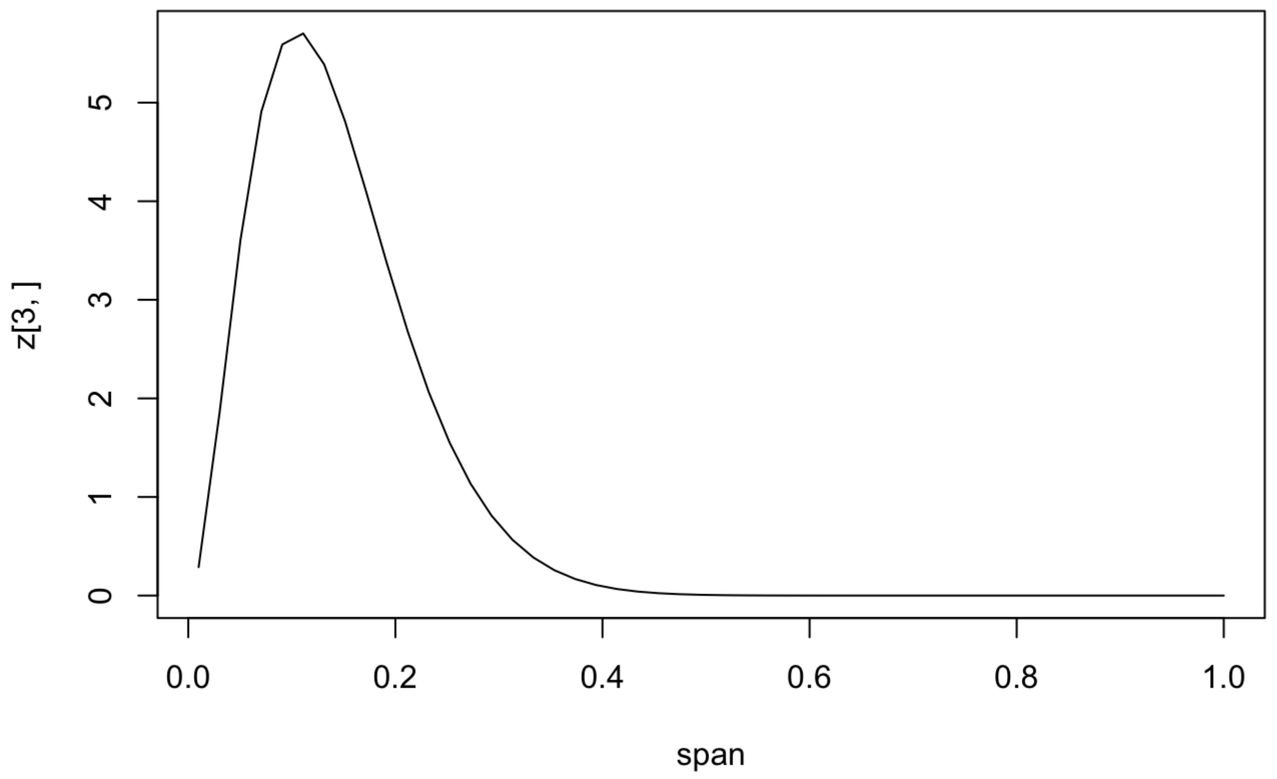
First we test how a single beta-distribution probability function works

```
#for a single line the script would be like this
span<-seq(0.01,1,length.out=20) #number of points in the series
alpha=10
beta=1
plot(span,dbeta(span,alpha,beta), "l",
     xlab="Values",
     ylab="Probability density") #span, alpha and beta
```

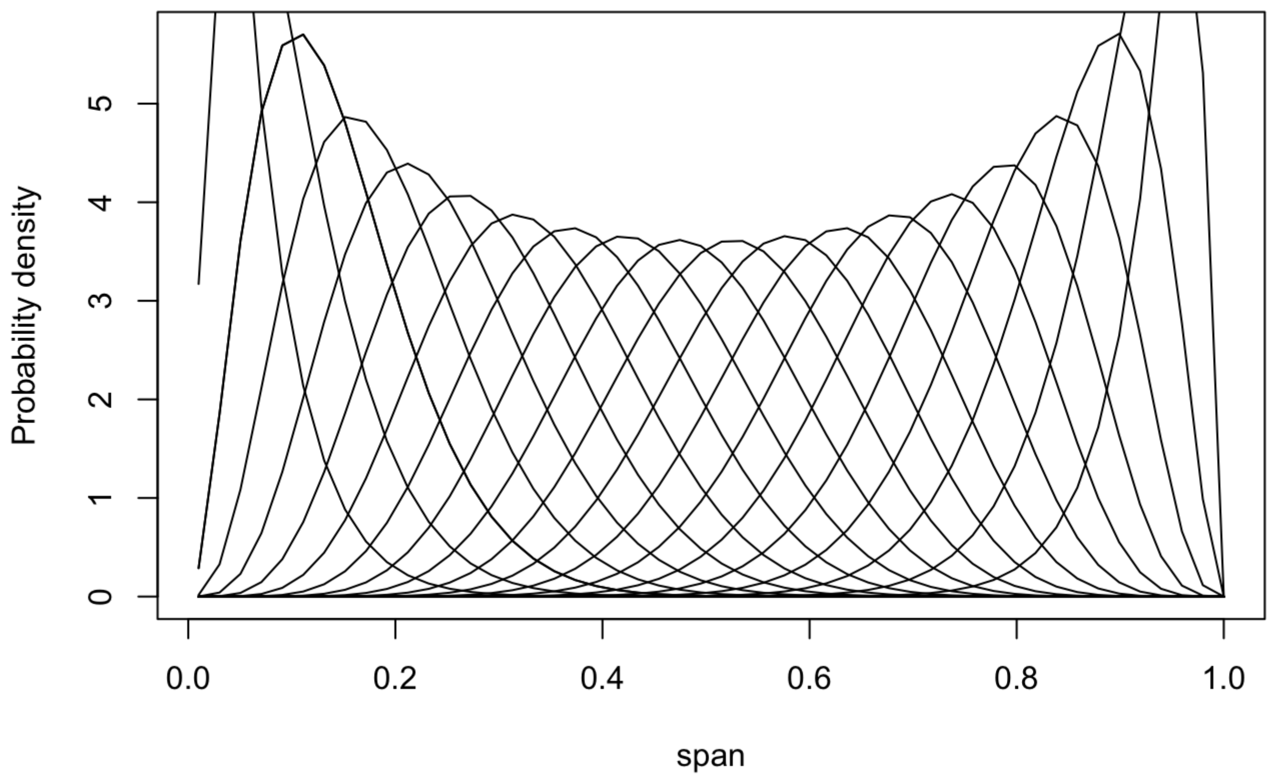


Now we do the same with the series of distributions where alpha is gradually increasung and beta is gradually decreasing

```
size=50 #length of the density distribution series
series=20
span=seq(0.01,1,length.out=size) #size series in scale 0 to 1
z=c() #probabioity density itself
#generate series of probabioity densities with increasing alpha (i) and
#decreasing beta (21-i) at the same time
for (i in 1:series){
  z=rbind(z,dbeta(span,i,series+1-i))
}
#plot one result as a single probability density in 2D
plot(span, z[3,],"l")
```



```
#add all series to the same plot
plot(span, z[3,],"l", ylab="Probability density")
for (i in 1:series){
  points(span, z[i,], "l")
}
```



Next we try 3D interactive figure as a series of lines

```
library(plotly)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

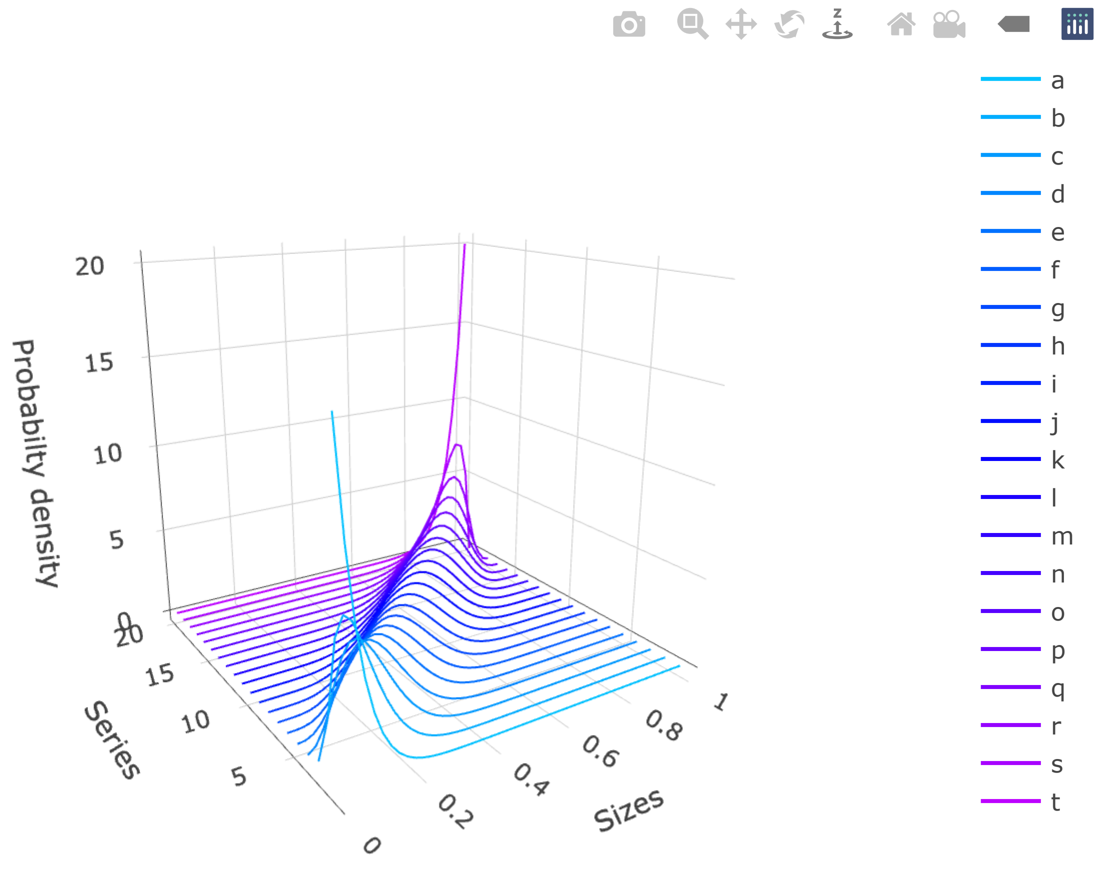
```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
#this loop will stitch coordinates for series (x), sizes (y) and
#make probability density distribution in zet (taken from the previous calculation)
Threedee=c()
for (i in 1:series){
  x=span
  y=rep(i,size)
  zet=z[i,]
  character=letters[i]
  block=cbind(x,y,zet,character)
  Threedee=rbind(Threedee, block)
}
```

Data are stitched together Now we show it in the plot. I choose for the shortest options. Plotly is a bit awkward in details. I need more time to find it all.

```
#this is an interactive 3D picture if you run the script
D3D=data.frame(Threedee)
plot_ly(D3D, x = ~D3D[,1], y = ~D3D[,2], z = ~D3D[,3],
        type='scatter3d', mode='lines',
        color= ~character, #discrete color disconnects lines
        colors=rainbow(100)[55:80]
                    ) %>%
layout(scene = list(xaxis = list(title = 'Sizes'),
                      yaxis = list(title = 'Series'),
                      zaxis = list(title = 'Probabilty density'
                      ))
         )
```



It feels like the first view is from the wrong angle. But good luck it is rotating picture. You can find a right angle for sure. The plot clearly shows smooth transition from left-biased distribution to the right biased distribution. Note that the left biased distribution in real life is very biased (many small clones and a few big), whereas right biased distribution is close to equal or flat (many big clones and a few small). Ths is the end of the illustration. L.B. The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.