R PROJECT- MOVIES ON OTT PLATFORM

```
DEVI PRIYA.G     E0320016
```

```
OBJECTIVE  - To analyse and predict which OTT platform has influenced people
PROBLEM STATEMENT- To analyse and find answers for the following questions:
1. To determine which factors decide the runtime
2. Does year affect the Runtime on OTT platforms?
3. Find the type of correlation between Year and Runtime
4. Does the effect of Netflix, Hulu, Prime video and Disney has a positive or negative
correlation on Runtime?
5. Which country has the lowest runtime?
```

In [4]:

```
library(readxl)
```

In [5]:

```
#reading the datas from excel file
df=read.csv('MoviesOnStreamingPlatforms_updated.csv')
df
```

9500	9501	Challenge Accepted! Disney Channel's Epic Holiday Showdown	2020		3.9/10	32/100	0	0	0	1	0
9501	9502	Sharks of the Bermuda Triangle	2020		6.5/10	31/100	0	0	0	1	0
9502	9503	Sharkcano	2020			16/100	0	0	0	1	0
9503	9504	Disney My Music Story: Sukima Switch	2021	16+		16/100	0	0	0	1	0
9504	9505	Big Cat Games	2015			15/100	0	0	0	1	0
9505	9506	Great Shark Chow Down	2019	7+		14/100	0	0	0	1	0

In [6]:

```
#no of rows
nrow(df)
```

9515

In [9]:

```
#no of columns
ncol(df)
```

```
17
```

In [10]:

```
#data type of each column
sapply(df,class)
```

```
X
'integer'
ID
'integer'
Title
'factor'
Year
'integer'
Age
'factor'
IMDb
'factor'
Rotten.Tomatoes
'factor'
Netflix
'integer'
Hulu
'integer'
Prime.Video
'integer'
Disney.
'integer'
Type
'integer'
Directors
'factor'
Genres
'factor'
Country
'factor'
Language
'factor'
Runtime
'numeric'
```

In [12]:

```
#checking whether any NA is available
is.na(df)
```

X	ID	Title	Year	Age	IMDb	Rotten.Tomatoes	Netflix	Hulu	Prime.Video
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
...	...	...	...	...	...	...	...	...	...
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE

X	ID	Title	Year	Age	IMDb	Rotten.Tomatoes	Netflix	Hulu	Prime.Video
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE		FALSE	FALSE	FALSE

In [11]:

```
any(is.na(df))
```

```
TRUE
```

In [8]:

```
#zoo package to update NA
library(zoo)
```

```
Attaching package: 'zoo'

The following objects are masked from 'package:base':

    as.Date, as.Date.numeric
```

In [9]:

```
#updating NA with previous values
df1=na.locf(na.locf(df),fromLast=TRUE)
```

In [10]:

```
df1
```

X	ID	Title	Year	Age	IMDb	Rotten.Tomatoes	Netflix	Hulu	Prime.Video	Disney.	Type	
0	1	The Irishman	2019	18+	7.8/10	98/100	1	0	0	0	0	
1	2	Dangal	2016	7+	8.4/10	97/100	1	0	0	0	0	
2	3	David Attenborough: A Life on Our Planet	2020	7+	9.0/10	95/100	1	0	0	0	0	Fo
3	4	Lagaan: Once Upon a Time in India	2001	7+	8.1/10	94/100	1	0	0	0	0	
4	5	Roma	2018	18+	7.7/10	94/100	1	0	0	0	0	
5	6	To All the Boys I've Loved Before	2018	13+	7.1/10	94/100	1	0	0	0	0	
6	7	The Social										

In [15]:

```
any(is.na(df1))
```

FALSE

In [11]:

```
#summary of dataset
summary(df1)
```

```
       X              ID                                 Title       
 Min.   :   0   Min.   :   1   '71                         :   1  
 1st Qu.:2378   1st Qu.:2380   'Neath Brooklyn Bridge      :   1  
 Median :4757   Median :4758   'Twas the Night             :   1  
 Mean   :4757   Mean   :4758   #Alive                      :   1  
 3rd Qu.:7136   3rd Qu.:7136   #AnneFrank. Parallel Stories:   1  
 Max.   :9514   Max.   :9515   #cats_the_mewvie            :   1  
                               (Other)                     :9509  
      Year          Age             IMDb      Rotten.Tomatoes    Netflix      
 Min.   :1914      :4177   6.5/10 : 373   44/100 : 311    Min.   :0.0000  
 1st Qu.:2006   13+: 998   6.2/10 : 363   46/100 : 298    1st Qu.:0.0000  
 Median :2015   16+: 276   6.4/10 : 352   47/100 : 291    Median :0.0000  
 Mean   :2007   18+:2276   6.6/10 : 325   49/100 : 290    Mean   :0.3883  
 3rd Qu.:2018   7+ :1090   6.3/10 : 320   43/100 : 289    3rd Qu.:1.0000  
 Max.   :2021   all: 698   7.2/10 : 315   48/100 : 288    Max.   :1.0000  
                           (Other):7467   (Other):7748                    
      Hulu        Prime.Video        Disney.           Type  
 Min.   :0.00   Min.   :0.0000   Min.   :0.0000   Min.   :0  
 1st Qu.:0.00   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0  
 Median :0.00   Median :0.0000   Median :0.0000   Median :0  
 Mean   :0.11   Mean   :0.4323   Mean   :0.0969   Mean   :0  
 3rd Qu.:0.00   3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:0  
 Max.   :1.00   Max.   :1.0000   Max.   :1.0000   Max.   :0  
                                                             
                  Directors                         Genres    
                      : 411   Comedy                 : 780  
 Jay Chapman          :  31   Drama                  : 604  
 RaÃºl Campos,Jan Suter:  22   Documentary            : 567  
 Jay Karas            :  20   Comedy,Drama           : 309  
 Manny Rodriguez      :  19   Drama,Romance          : 258  
 Marcus Raboy         :  17   Comedy,Drama,Romance: 247  
 (Other)              :8995   (Other)                :6750  
                Country                    Language        Runtime      
 United States        :4650   English         :5652   Min.   :  1.00  
 India                : 827   Hindi           : 383   1st Qu.: 84.00  
 United Kingdom       : 374                   : 313   Median : 95.00  
                      : 254   Spanish         : 209   Mean   : 94.74  
 Canada               : 233   English,Spanish: 194   3rd Qu.:108.00  
 United States,Canada: 125   English,French : 112   Max.   :566.00  
 (Other)              :3052   (Other)         :2652                   
```

In [12]:

```
#PLOTS USING GGPLOT2
library(ggplot2)
```

In [13]:

```
colnames(df1)
```

'X' 'ID' 'Title' 'Year' 'Age' 'IMDb' 'Rotten.Tomatoes' 'Netflix' 'Hulu' 'Prime.Video' 'Disney.' 'Type' 'Directors' 'Genres' 'Country' 'Language' 'Runtime'

In [ ]:

```

```

STATISTICAL ANALYSIS

In [14]:

```r
colnames(df)
```

```
'X'  'ID'  'Title'  'Year'  'Age'  'IMDb'  'Rotten.Tomatoes'  'Netflix'  'Hulu'  'Prime.Video'
'Disney.'  'Type'  'Directors'  'Genres'  'Country'  'Language'  'Runtime'
```

RUNTIME

In [21]:

```r
#finding mean
mean(df1$Runtime)
```

```
94.7389385181293
```

In [22]:

```r
#finding median
median(df1$Runtime)
```

```
95
```

In [23]:

```r
#finding standard deviation
sd(df1$Runtime)
```

```
29.8438926350055
```

In [24]:

```r
#finding variance
var(df1$Runtime)
```

```
890.657927609736
```

In [25]:

```r
#finding range
range(df1$Runtime)
```

```
1   566
```

In [26]:

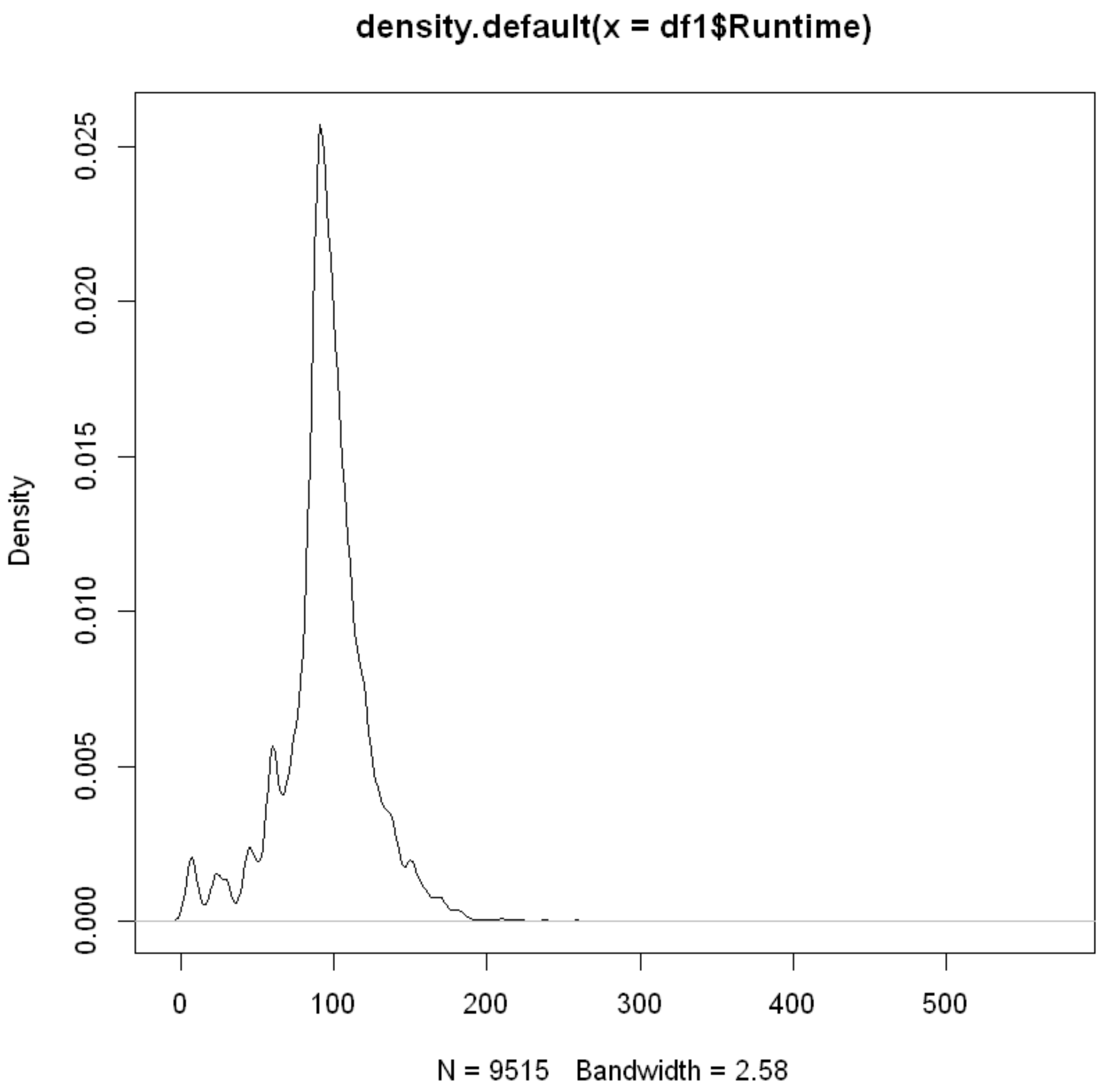
```
#to find skew
library(e1071)
```

```
Warning message:
"package 'e1071' was built under R version 3.6.3"
```

In [27]:

```
paste("skewness: ",skewness(df1$Runtime))
plot(density(df1$Runtime))
```

'skewness: 0.454858195523907'



In [29]:

```
#to find kurtosis
paste( "kurtosis: ",kurtosis( df1$Runtime))
```

'kurtosis: 10.0285252704918'

In [56]:

```
#finding mean
mean(df1$Year)
```

```
2007.42238570678
```

In [57]:

```
median(df1$Year)
```

```
2015
```

In [58]:

```
sd(df1$Year)
```

```
19.1303667085078
```

In [59]:

```
var(df1$Year)
```

```
365.970930401982
```

In [60]:

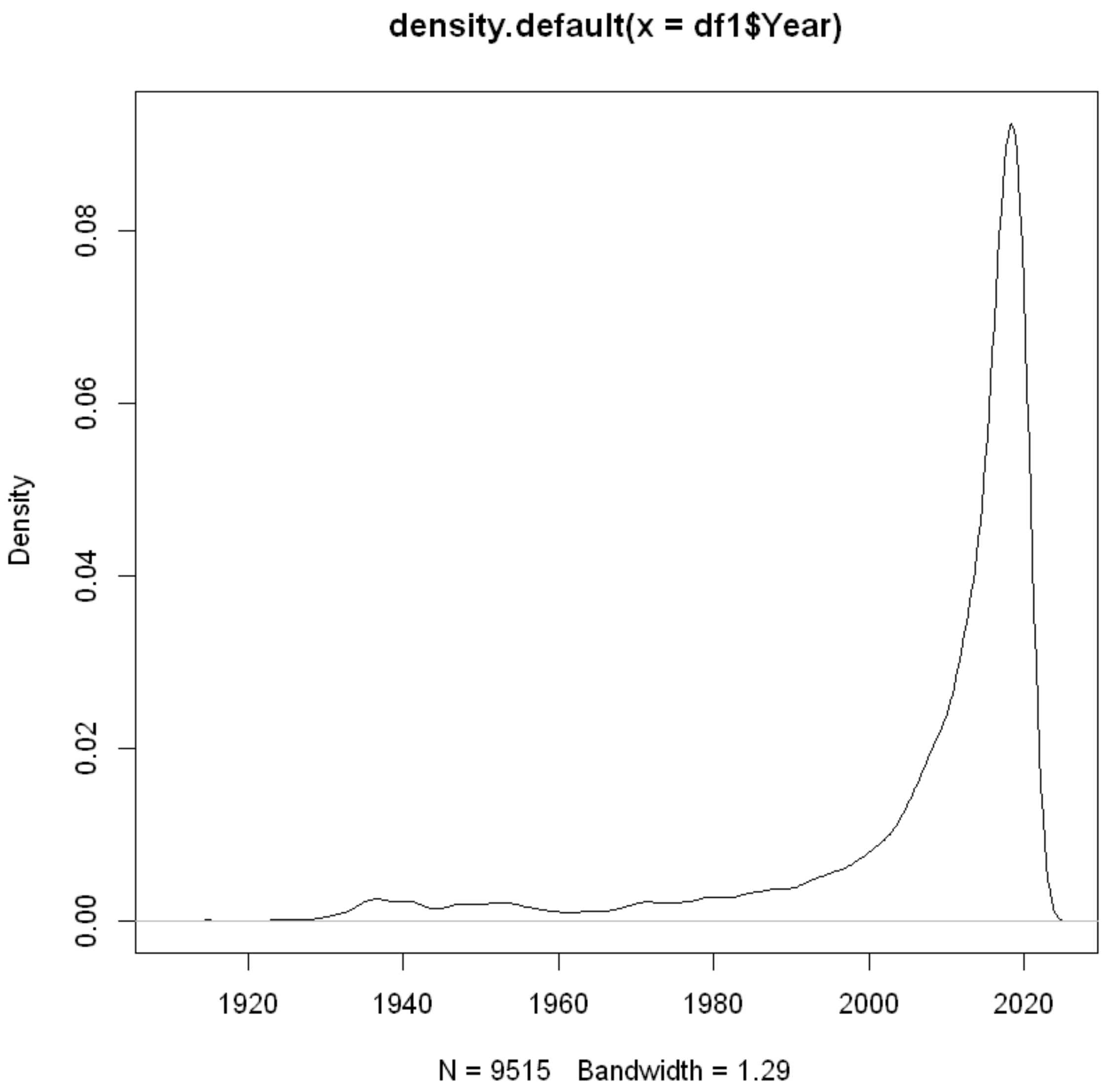
```
range(df1$Year)
```

```
1914   2021
```

In [61]:

```
paste("skewness: ",skewness(df1$Year))
plot(density(df1$Year))
```

'skewness: -2.35910364085927'



In [62]:

```
mean(df1$Netflix)
```

```
0.388334209143458
```

In [63]:

```
median(df1$Netflix)
```

```
0
```

In [64]:

```
sd(df1$Netflix)
```

```
0.487396878936141
```

In [65]:

```
range(df1$Netflix)
```

```
0   1
```

In [66]:

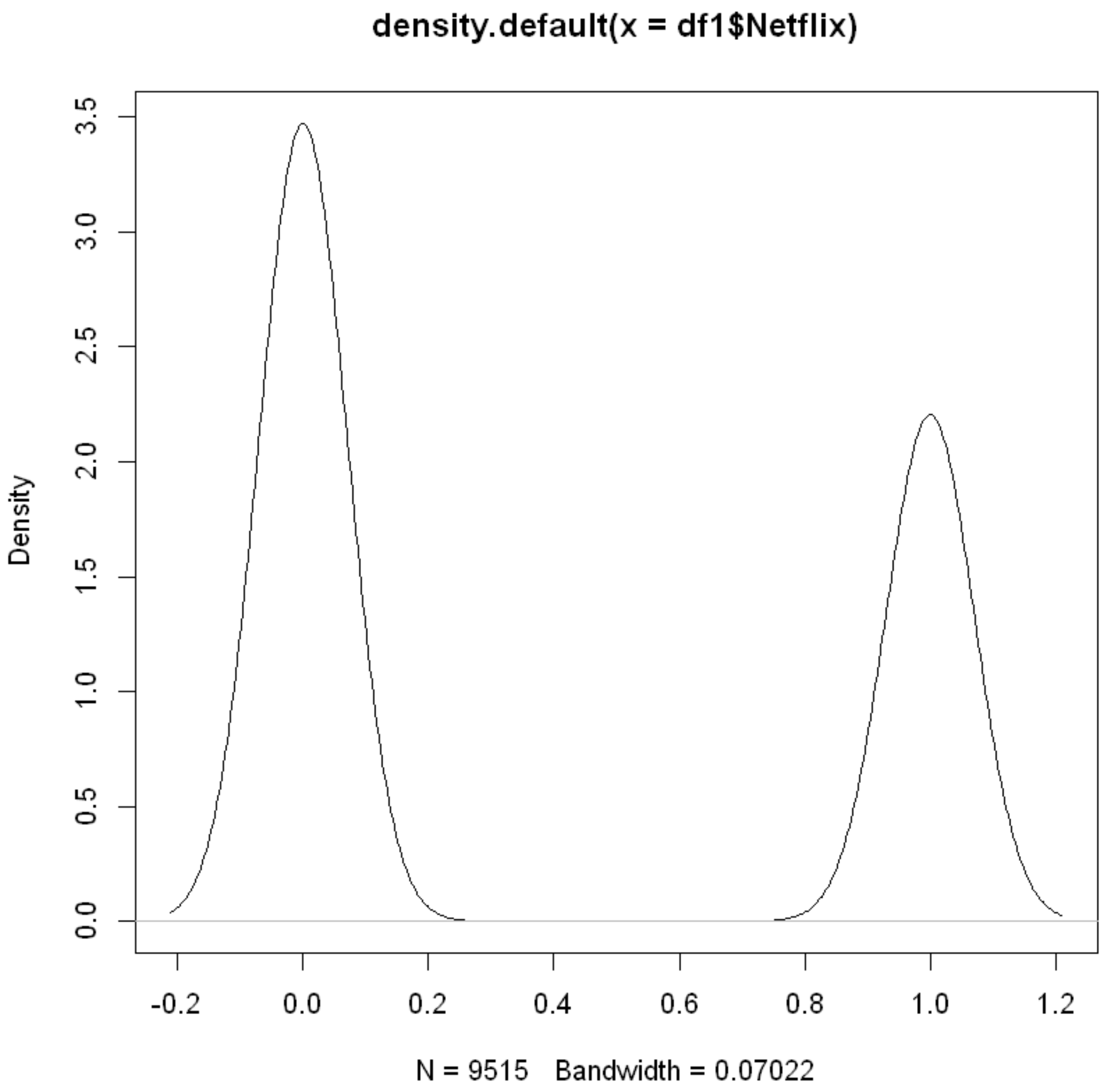
```
var(df1$Netflix)
```

```
0.237555717596692
```

In [67]:

```
paste("skewness: ",skewness(df1$Netflix))
plot(density(df1$Netflix))
```

```
'skewness: 0.458164834114787'
```



In [68]:

```
mean(df1$Hulu)
```

```
0.110036784025223
```

In [69]:

```
median(df1$Hulu)
```

```
0
```

In [70]:

```
sd(df1$Hulu)
```

```
0.312952046328167
```

In [71]:

```
var(df1$Hulu)
```

```
0.0979389833009871
```

In [72]:

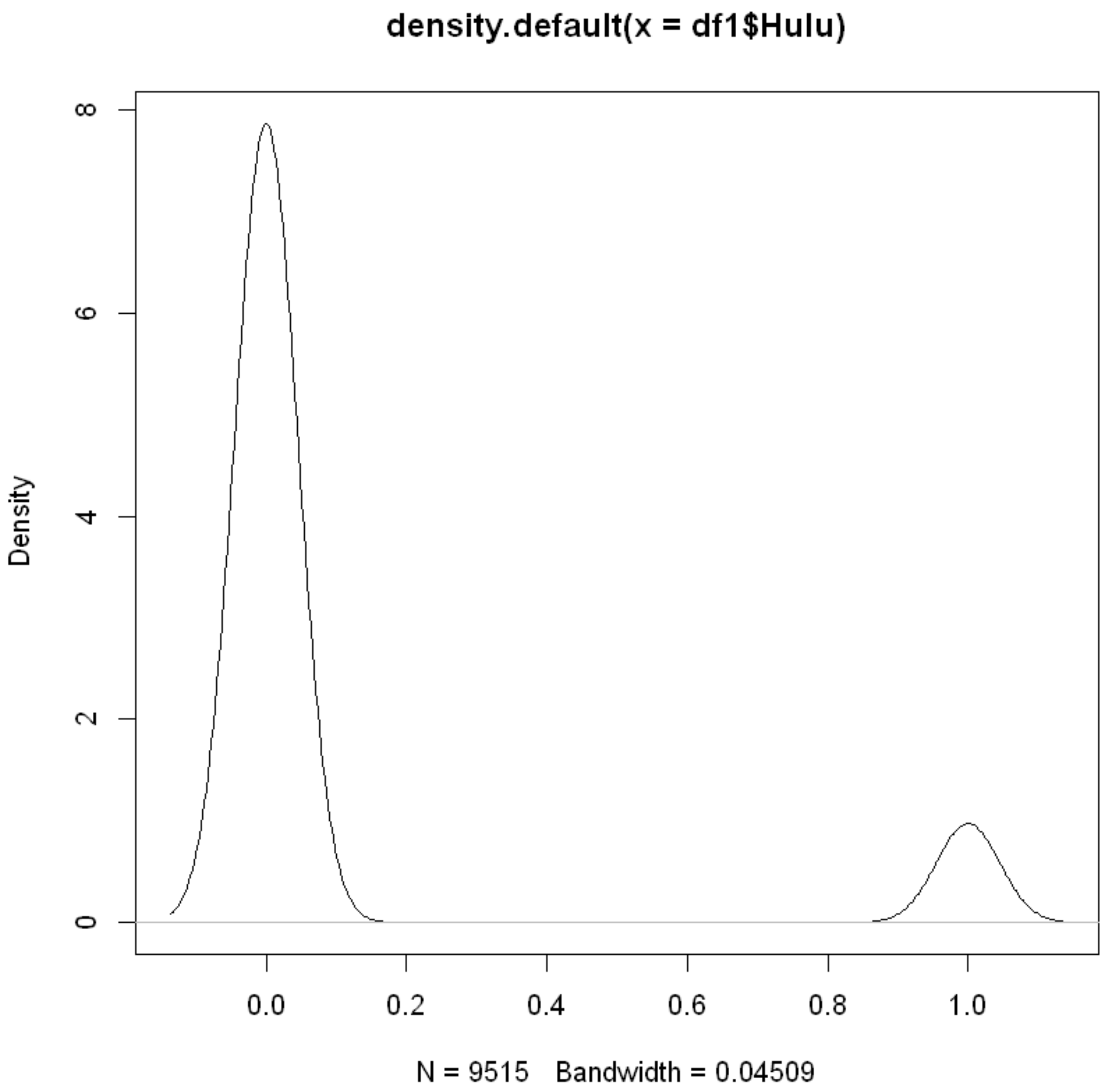
```
range(df1$Hulu)
```

```
0   1
```

In [73]:

```
paste("skewness: ",skewness(df1$Hulu))
plot(density(df1$Hulu))
```

```
'skewness: 2.49189763416965'
```



In [74]:

```
mean(df1$Prime.Video)
```

```
0.432264844981608
```

In [75]:

```
median(df1$Prime.Video)
```

```
0
```

In [76]:

```
sd(df1$Prime.Video)
```

```
0.495416737300955
```

In [77]:

```
var(df1$Prime.Video)
```

```
0.245437743597924
```

In [78]:

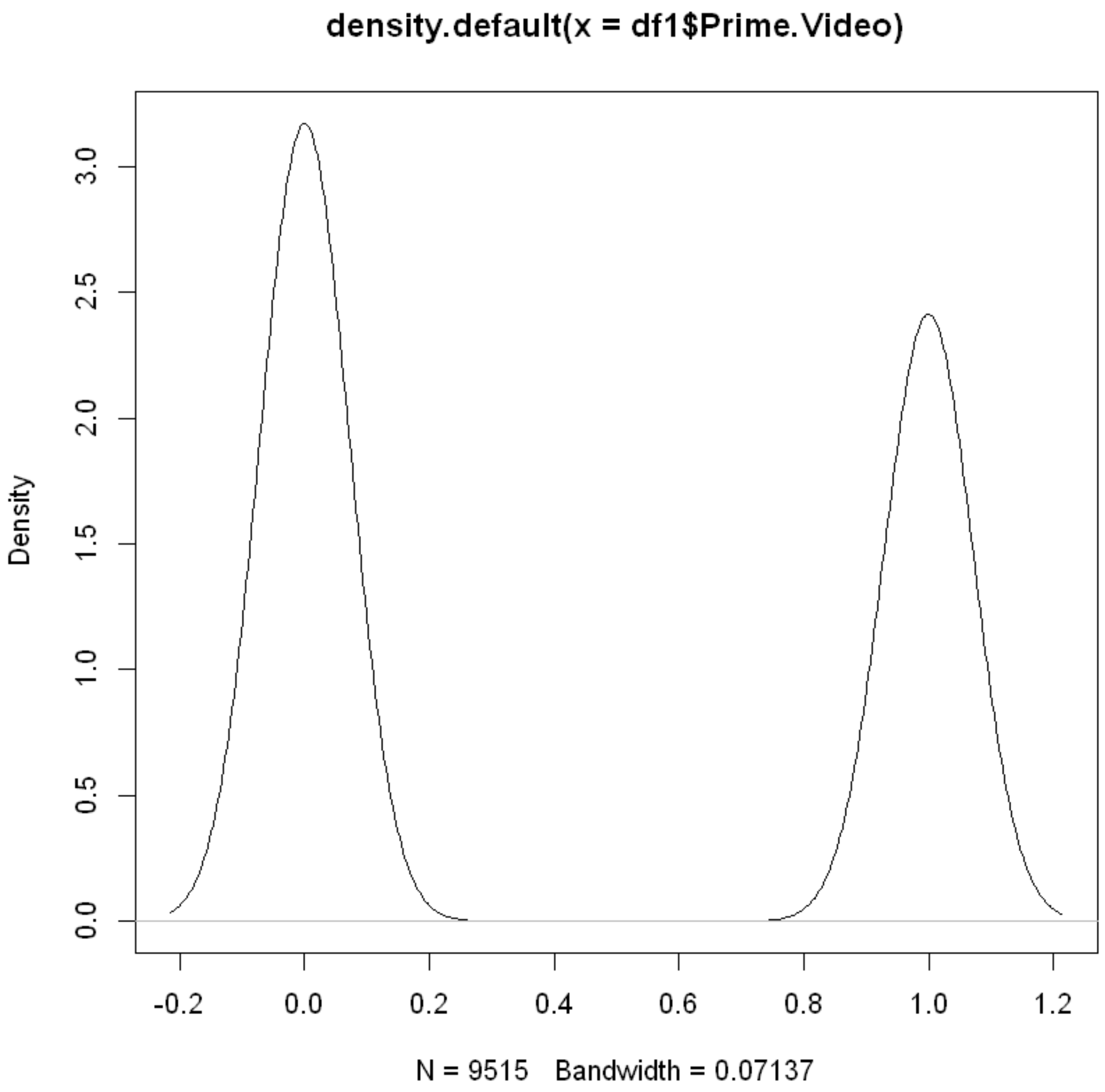
```
range(df1$Prime.Video)
```

```
0   1
```

In [79]:

```
paste("skewness: ",skewness(df1$Prime.Video))
plot(density(df1$Prime.Video))
```

'skewness: 0.27341844206246'



In [80]:

```
mean(df1$Disney)
```

```
0.0968996321597478
```

In [81]:

```
median(df1$Disney)
```

```
0
```

In [82]:

```
sd(df1$Disney)
```

```
0.295836595912068
```

In [83]:

```
var(df1$Disney)
```

```
0.0875192914808401
```

In [84]:

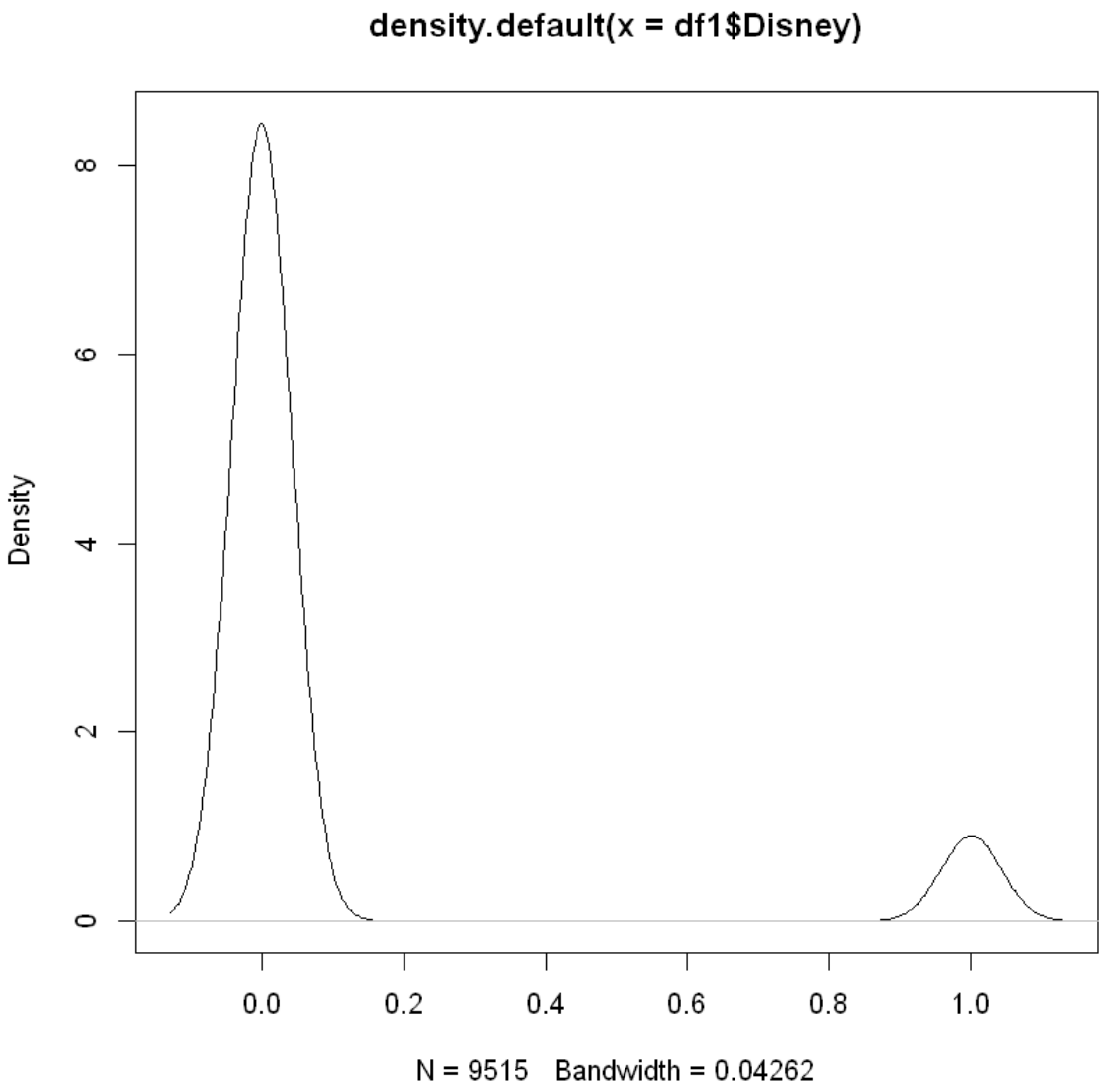
```
range(df1$Disney)
```

```
0   1
```

In [85]:

```
paste("skewness: ",skewness(df1$Disney))
plot(density(df1$Disney))
```

'skewness: 2.72486912494125'



CORRELATION

In [32]:

```
df2=subset(df1,Country== "United States" )
df2
```

	X	ID	Title	Year	Age	IMDb	Rotten.Tomatoes	Netflix	Hulu	Prime.Video	Disney.	Ty
1	0	1	The Irishman	2019	18+	7.8/10	98/100	1	0	0	0	
6	5	6	To All the Boys I've Loved Before	2018	13+	7.1/10	94/100	1	0	0	0	
7	6	7	The Social Dilemma	2020	13+	7.6/10	93/100	1	0	0	0	
9	8	9	The Ballad of Buster Scruggs	2018	16+	7.3/10	92/100	1	0	0	0	
13	12	13	Dolemite Is My Name	2019	18+	7.3/10	92/100	1	0	0	0	
14	13	14	Mudbound	2017	18+	7.4/10	91/100	1	0	0	0	

In [36]:

```
r1=cor.test(df1$Year,df1$Runtime)
r1=r1$estimate
r1
```

cor: 0.114132306185527

In [88]:

```
result=cor.test(df1$Year,df1$Runtime,method="spearman",use=complete.obs)
```

```
Warning message in cor.test.default(df1$Year, df1$Runtime, method = "spearma
n", :
"Cannot compute exact p-value with ties"
```

In [89]:

```
correlation=result$estimate
correlation
```

rho: 0.0261012703827693

In [90]:

```
if(correlation>0 & correlation<2){
    print(paste('positive correlation'))
}else if(correlation == 0){
    print(paste('No correlation'))
}else if(correlation < 0 & correlation==-1){
    print(paste('Negative correlation'))
}else{
    print(paste('invalid'))
}
```

```
[1] "positive correlation"
```

In [45]:

```
r2=cor.test(df1$Netflix,df1$Hulu)
r2=r2$estimate
r2
```

cor: -0.253299992932729

In [47]:

```
r3=cor.test(df1$Netflix,df1$Runtime)
r3=r3$estimate
r3
```

cor: 0.112181096243935

In [48]:

```
pl=ggplot(data=df1,aes( x=Netflix,y=Runtime))
pl=pl+geom_point(col="firebrick")
pl=pl+stat_smooth( col="palegreen" )
ggplotly(pl)
```

```
`geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
Warning message:
"Computation failed in `stat_smooth()`:
x has insufficient unique values to support 10 knots: reduce k."
```

In [50]:

```
r4=cor.test(df1$Hulu,df1$Runtime)
r4=r4$estimate
r4
```

cor: 0.0406528662471413

In [52]:

```
r5=cor.test(df1$Prime.Video,df1$Runtime)
r5=r5$estimate
r5
```

cor: 0.00818780280622094

DATA VISUALISATION USING GGPLOT2 AND PLOTLY

In [18]:

```
library(ggplot2)
```

In [19]:

```
library(plotly)
install.packages("patchwork")
library(patchwork)
```

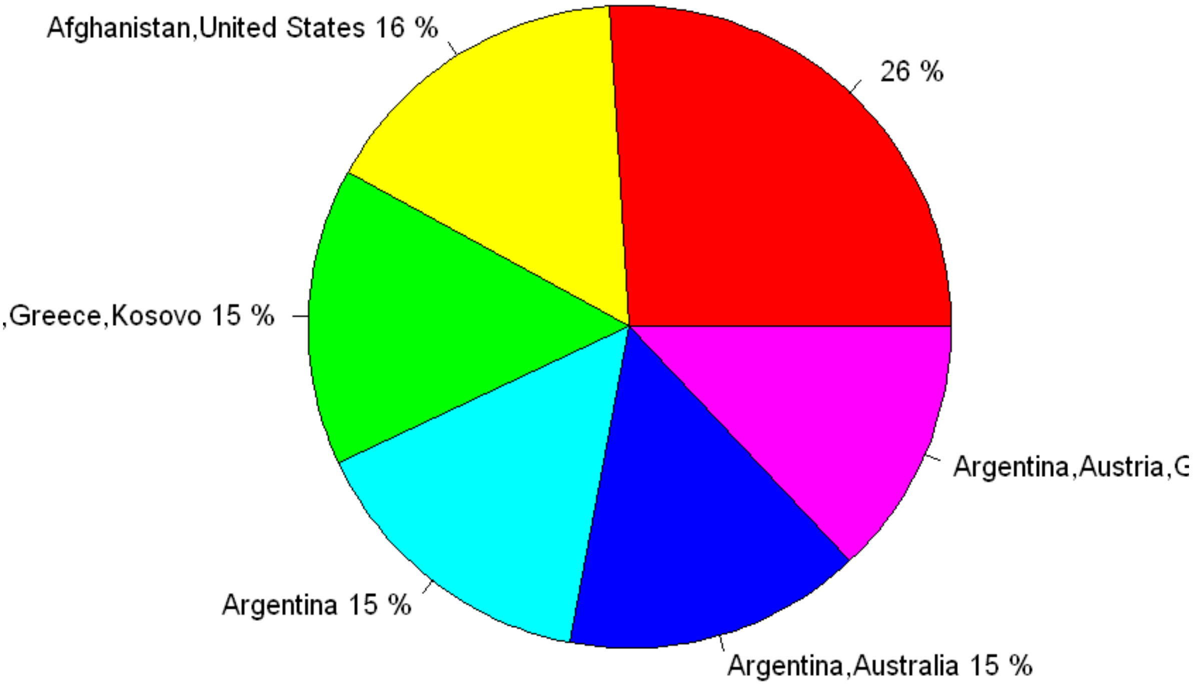
```
Error in library(plotly): there is no package called 'plotly'
Traceback:

1. library(plotly)
```

In [34]:

```
sorted = order(df1['Runtime'],decreasing = TRUE)
a=df1[sorted,]
sort1 = a['Country']
sort11 = a['Runtime']
sorting1 = sort1[c(1:6),]
sorting11 = sort11[c(1:6),]
```

In [35]:

```
percentage=round(100*(sorting11/sum(sorting11)))
pie(percentage,label=paste(levels(df1$Country),percentage,"%"),col=rainbow(length(sorting1)
```



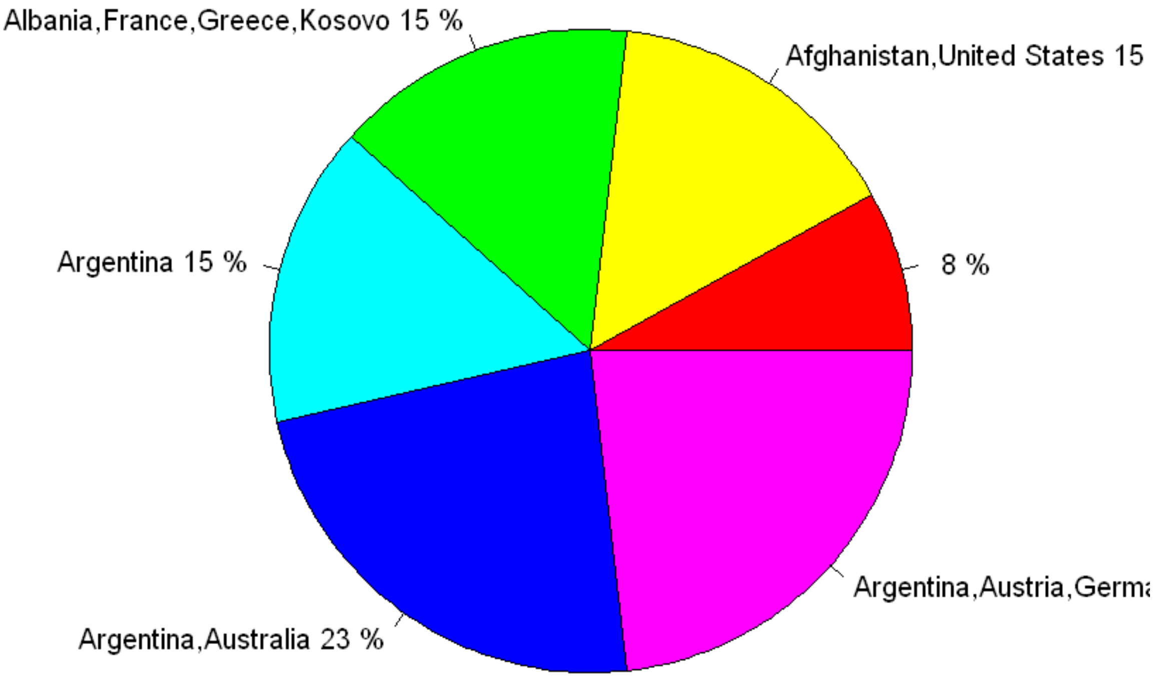
COUNTRY WITH LOWEST RUNTIME

In [20]:

```
sorted2 = order(df1['Runtime'],decreasing = FALSE)
a2=df1[sorted2,]
sort2 = a2['Country']
sort22 = a2['Runtime']
sorting2 = sort2[c(1:6),]
sorting22 = sort22[c(1:6),]
```

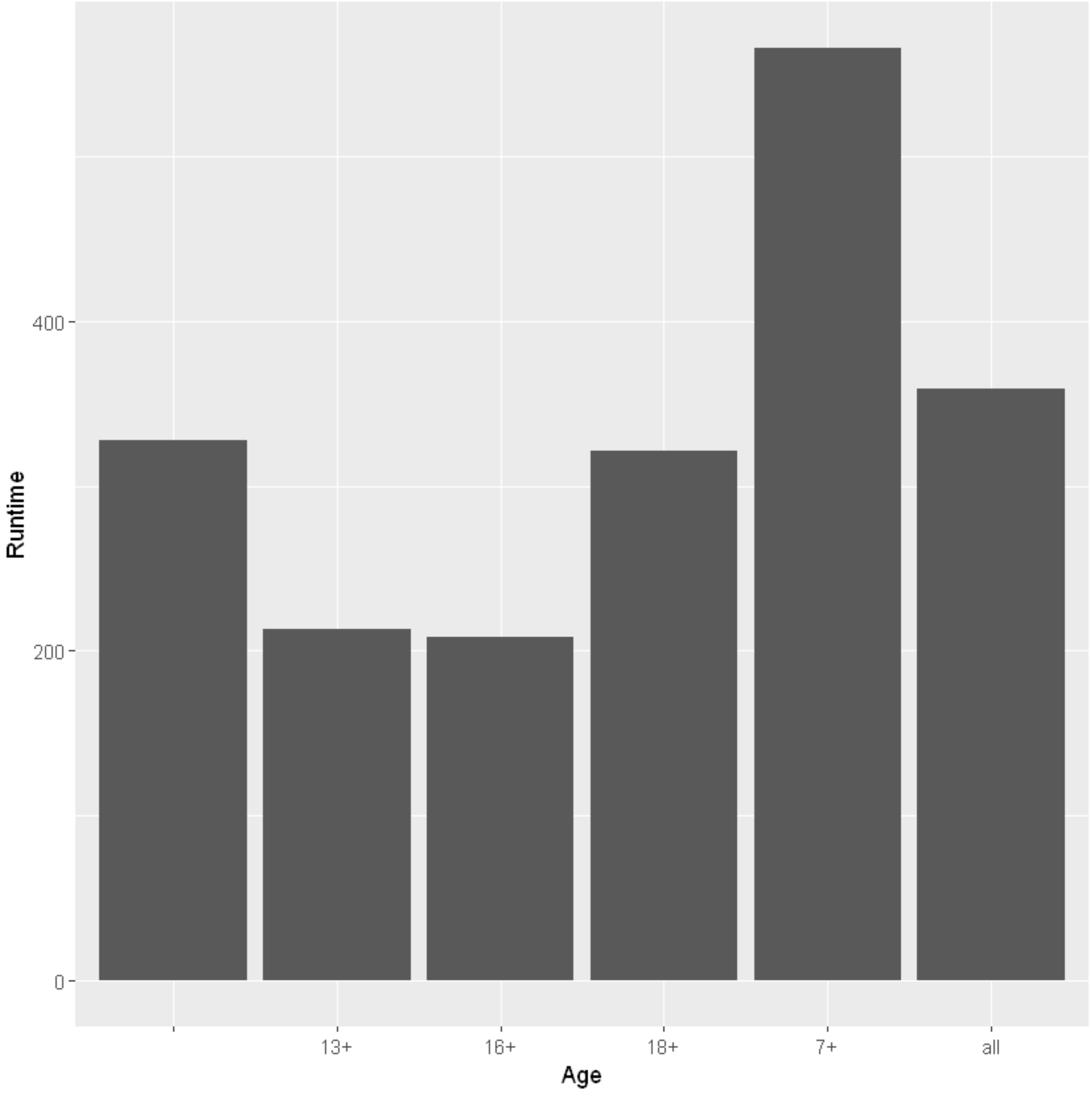
In [34]:

```
percentage=round(100*(sorting22/sum(sorting22)))
pie(percentage,label=paste(levels(df1$Country),percentage,"%"),col=rainbow(length(sorting2)
```



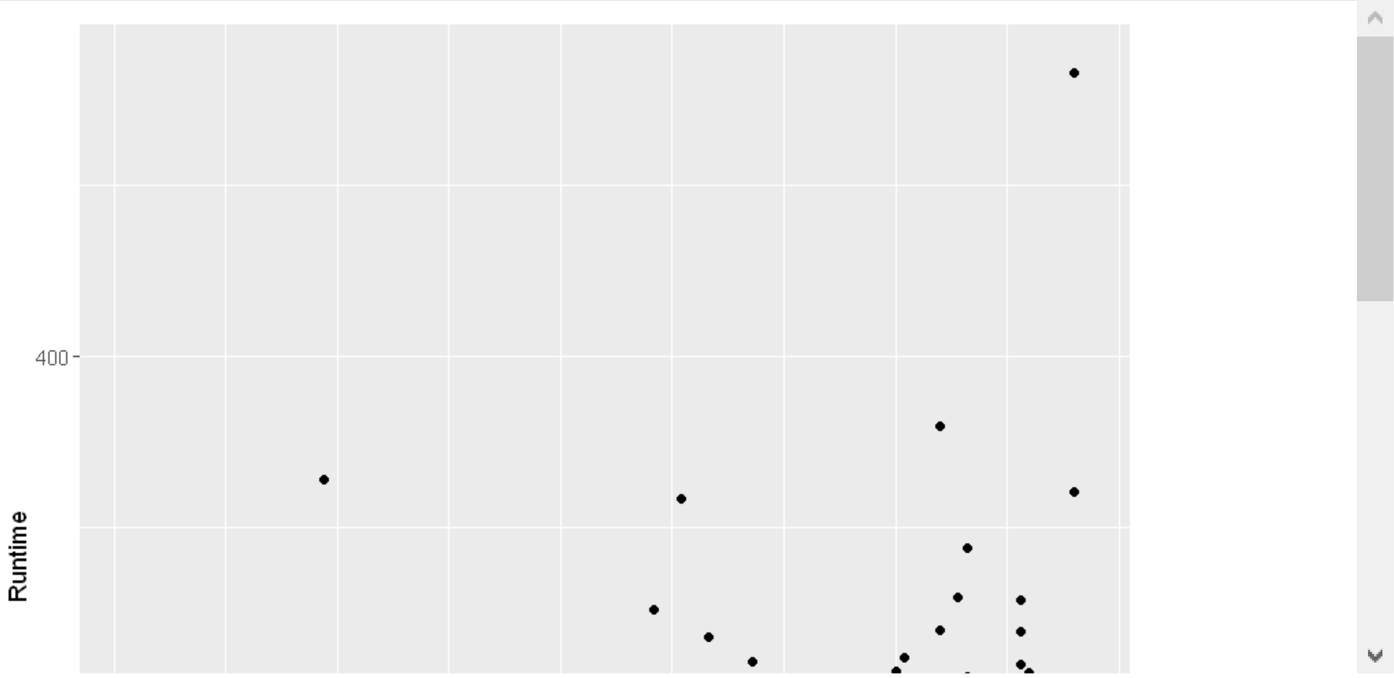
In [95]:

```
p <- ggplot(df1, aes(Age,Runtime))+  geom_bar(stat="identity",position = "dodge")+ scale_fi
 p
```



In [98]:

```
ggplot(df1, aes(Year,Runtime)) +    geom_point()
```



In [122]:

```
install.packages("caTools")
library(caTools)
```

```
Warning message:
"package 'caTools' is in use and will not be installed"
```

Predicting Runtime of Netflix Movies in Minutes

In [121]:

```
s=sample.split(df1,SplitRatio = 0.7)
train = subset(df1,split=T)
test = subset(df1,split=F)

model=lm(df1$Netflix~df1$Runtime,data=test)
summary(model)
p=predict(model,train)
RSE=sigma(model)/mean(df1$Runtime)
RSE
acc=sqrt(mean((df1$Runtime-p)^2))
acc
```

```
Call:
lm(formula = df1$Netflix ~ df1$Runtime, data = test)

Residuals:
    Min      1Q  Median      3Q     Max 
-1.2517 -0.3906 -0.3522  0.5910  0.7797 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 0.2147639  0.0165268   12.99   <2e-16 ***
df1$Runtime 0.0018321  0.0001664   11.01   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4843 on 9513 degrees of freedom
Multiple R-squared:  0.01258,	Adjusted R-squared:  0.01248 
F-statistic: 121.2 on 1 and 9513 DF,  p-value: < 2.2e-16
```

0.00511242562580866

98.9410968987873

In [ ]: