Does the ranking of the msnbc pages viewed by visitors follow Zipf's law?

The pages follow this order

```
frontpage news tech local opinion on-air misc weather
msn-news health living business msn-sports sports
summary bbs travel
```

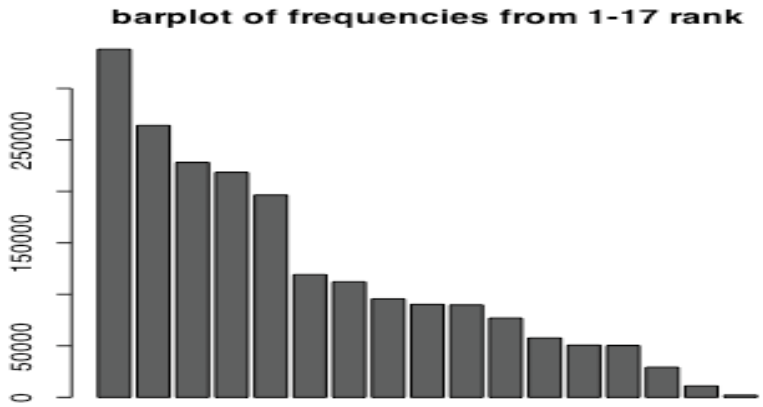
The frequency (popularity) of those pages and their rank is given in the next table. Got the information from the popularity and sorted command in R (see code attached in Appendix 2).

Page	popularity(f)	rank(after sorting)
Frontpage	338056	1
News	264016	2
Tech	196461	5
Local	228143	3
Opinion	50326	14
On-air	218560	4
Misc	89708	10
Weather	95615	8
msn-new	90192	9
health	50606	13
living	57597	12
business	112183	7
msn-sports	76948	11
sports	119138	6
summary	29200	15
bbs	2082	17
travel	11006	16

When the table is arranged in order of popularity, from most popular to less popular, we get the following table.

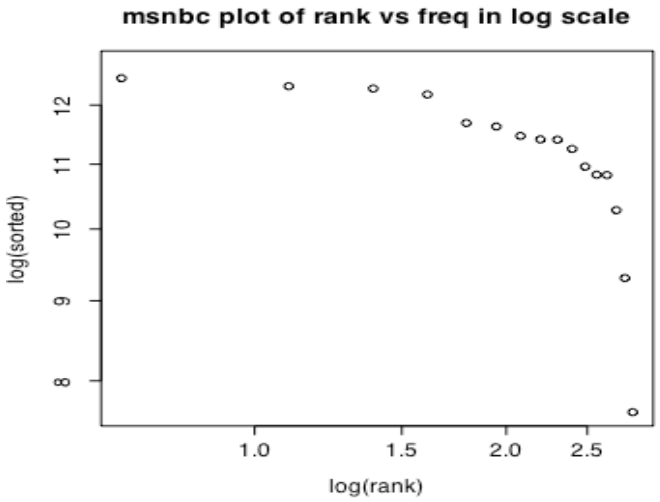
Page	Frequency (f)	Rank (r)	fxr
Frontpage	338056	1	338056
News	264016	2	528032
Local	228143	3	684429
On-air	218560	4	874240
Tech	196461	5	982305
Sports	119138	6	714828
business	112183	7	785281
Weather	95615	8	764920
msn-news	90192	9	811728
Misc	89708	10	897080
msn-sports	76948	11	846428
Living	57597	12	691164
Health	50606	13	657878

Opinion	50326	14	704564
summary	29200	15	438000
travel	11006	16	176096
bbs	2082	17	35394



Looking at the table, we can see that one of the implications of Zipf's law, that C=fxr, doesn't seem to hold very well for ranks 1 to 2 and ranks 15-18. We see in the middle some fxr around the 800000's or not too far but the upper tail (last 3 frequencies) are too far from that. One would not be surprised to see the first 3 (rank 1,2,3) to depart and a few random ones, but the upper tail is what Zipf's law predicts… so we can be skeptical.

By doing the regression of log frequency against log rank we can find the power law coefficient. We look at the plot in log scale too.

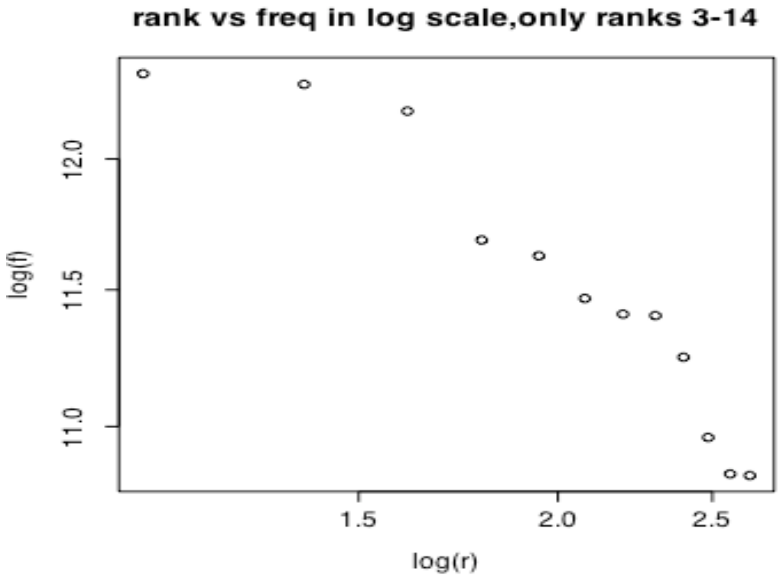


The plot doesn't look too linear, as we can see, due mostly to the r=1, 2, and 3..

So fitting a regression is not even adequate to do…
The regression equation is

Log(frequency) = 13.638 –1.227 log(rank)

The slope is not –1. It is not too far from it, though. It is –1.227. We should try removing the ranks that seem to be out of the C level for fxr and that make the plot nonlinear. I removed the first 2 and the last 3. Doing that, the plot in log scale and the regression equation is



Log(f) = 13.692 –1.062 log(r)

So for those ranges, Zipf's law seems to hold better.

Attachments

Attachment 1.- R code for problem 1 on words in Tom Sawyer.

```
>f=c(3332,2972,1775,877,410,294,222,172,158,138,124,116,104,51,30,21,16,13,11,10,9,
8,4,2,2,1)
>r=c(1,2,3,10,20,30,40,50,60,70,80,90,100,200,300,400,500,600,700,800,900,1000,2000,
3000,4000,8000)
> logf=log(f)
> logr=log(r)
```

> lm(logf~logr)
Call:
lm(formula = logf ~ logr)
Coefficients:
(Intercept) logr
8.7990 -0.9623

- plot(r,f,log="xy",main="relation rank, frequency in log scale")

Attachment 2.- R code for problem 2 on the msnbc dataset

```
>zipfmsnbc=matrix(scan(file="http://www.stat.ucla.edu/~jsanchez/oid03/datasets/msnbc.txt"),nc
ol=19,byrow=T)            #type it all in one line. Don't cut and paste.
> zipfmsnbc[1:5,]
> popularity=matrix(rep(0,17),nrow=1)
>  for(i in 2:18){
     popularity[i-1]=sum(zipfmsnbc[,i])
     }
>  popularity          # This helps you see the frequencies for  pages 1-17
> sorted=sort(popularity,decreasing=T)  #sort from most frequent to least frequent
> sorted        # This helps you rank the 17 pages in increasing order of popularity
> barplot(sorted, main="barplot of frequencies from 1-17 rank")  #does bar plot

> freq=(sorted/989818)
> rank=seq(1:17)
> plot(rank,freq, main="bar graph of frequencies")

>  rank=seq(1,17)
regfit=lm(log(sorted[1,])~log(rank))  #notice we need logs
regfit       #  this gives you the slope  when you include all pages.
> f=sorted[1,3:14]     #exclude observations
> r=rank[3:14]       #exclude observations
> plot(log(r),log(f),log="xy",main="rank vs freq in log scale,only ranks 3-14")
> lm(log(f)~log(r) )         #regression with the reduced data set
```