assessmentPap

Question 1

There are 39644 record in the data set. I checked different ways each giving me the same result, indicating that the records are unique. For example:

```
NewsData <- read.csv("~/Assessment/Data/OnlineNewsPopularity/OnlineNewsPopularity.csv", stringsAsFactor
dim(NewsData)
```

```
## [1] 39644    61
```

```
sum(duplicated(NewsData))
```

```
## [1] 0
```

There are 39644 urls in the data, and time frame is “2013-01-07” “2014-12-27”.

```
library(stringr)
urls <- NewsData$url
length(urls)
```

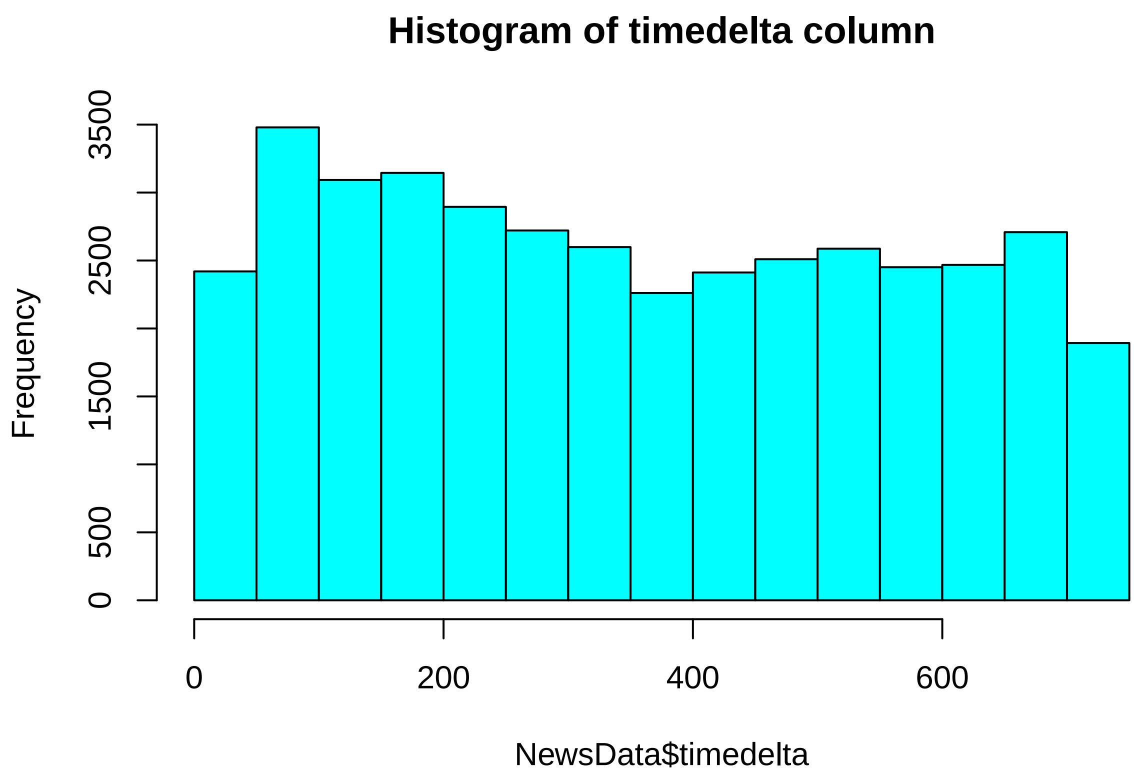
```
## [1] 39644
```

```
x <-as.Date(str_extract(urls[1:length(urls)],"[0-9]{4}/[0-9]{2}/[0-9]{2}"),"%Y/%m/%d")
x<-sort(x)
x[c(1,length(urls))]
```

```
## [1] "2013-01-07" "2014-12-27"
```

Question 2

```
hist(NewsData$timedelta, right = FALSE, col = "cyan", main = "Histogram of timedelta column")
```



The histogram indicates an almost uniform distribution of timedelta.

A part of the question asks *does it changes overtime*. It is not very obvious what "it" refers to. Does the histogram, distribution of the acquisition time (timedeta) change? Does the acquisition time itself changes? Here I have the monthly average of timedeta:

```
library(tidyverse)
```

```
## Loading tidyverse: ggplot2
## Loading tidyverse: tibble
## Loading tidyverse: tidyr
## Loading tidyverse: readr
## Loading tidyverse: purrr
## Loading tidyverse: dplyr

## Conflicts with tidy packages ----------------------------------------------

## filter(): dplyr, stats
## lag():    dplyr, stats
```

```
library(knitr)
Date <-as.Date(str_extract(urls[1:length(urls)],"[0-9]{4}/[0-9]{2}/[0-9]{2}"),"%Y/%m/%d")
NewsDataD <- cbind.data.frame(NewsData,Date,year = as.numeric(format(Date, format = "%Y")),
                 month = as.numeric(format(Date, format = "%m")),
                 day = as.numeric(format(Date, format = "%d")))
by_month <- group_by(NewsDataD,year,month)
  res <-summarise(by_month, Avetime = mean(timedelta))
  kable(res)
```

year	month	Avetime
2013	1	718.51550
2013	2	692.20176

year	month	Avetime
2013	3	663.36225
2013	4	632.81532
2013	5	601.98256
2013	6	571.58038
2013	7	541.12647
2013	8	510.01286
2013	9	479.09647
2013	10	448.74000
2013	11	418.83086
2013	12	389.74340
2014	1	356.08268
2014	2	327.10067
2014	3	297.56064
2014	4	267.39670
2014	5	236.69589
2014	6	206.19504
2014	7	175.35789
2014	8	144.42529
2014	9	114.12629
2014	10	82.60632
2014	11	52.75907
2014	12	24.21988
and we	can plot	this data:

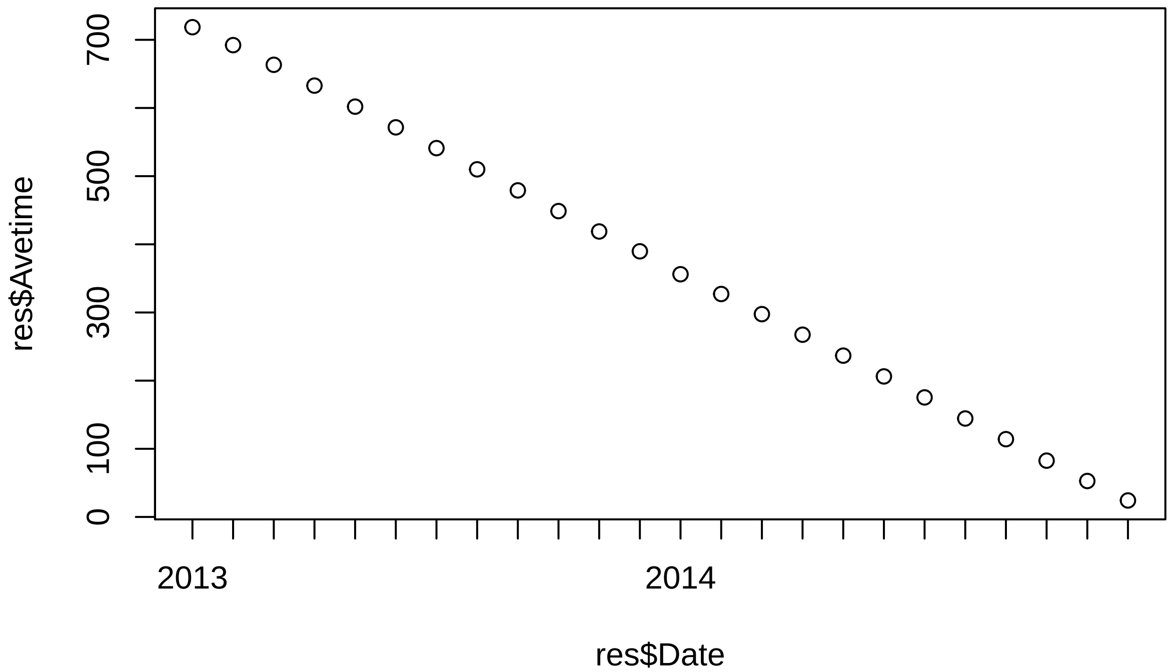
```
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
res$Date <- as.yearmon(paste(res$year,res$month, sep = "-"))
head(res,2)
```

```
## Source: local data frame [2 x 4]
## Groups: year [1]
##
##    year month  Avetime          Date
##   <dbl> <dbl>    <dbl> <S3: yearmon>
## 1  2013     1 718.5155      Jan 2013
## 2  2013     2 692.2018      Feb 2013
```

```
plot(res$Avetime~res$Date)
```



There is a steep drop in the days until acqusition as time has gone by. This makes sense since the publisher has gotten more mature and gained more experience and the delay between publishing time and acqusition time has decreased.

Question 3

The function *topic_extract* is written to extract the topic, as it is defined in question 3, from a given url:

```
ExtractedTopics <- lapply(urls,topic_extract) %>% unlist()
head(ExtractedTopics,4)
```

```
## [1] "amazon-instant-video-browser"   "ap-samsung-sponsored-tweets"
## [3] "apple-40-billion-app-downloads" "astronaut-notre-dame-bcs"
```

```
sum(duplicated(ExtractedTopics))
```

```
## [1] 0
```

I built a frequency table, used loops to detect and count any possible multiple occurances of a topic, and checked with duplicated function of R. All indicate that there is no multi-occurance and the frequency of each topic is exactly 1.

Question 4

The little one liner function *is_it_there* returns TRUE if a substring is in a given string, if not it returns FALSE. We use this function to answer this question.

```
s <- c("elon-musk", "facebook", "ebola", "ipad", "iphone", "tornado", "sharknado", "taylor-swift")
res <- lapply(ExtractedTopics,is_in_there,s)
x <- setNames(do.call(rbind.data.frame,res),s)
t <- apply(x,2,sum)
kable(t)
```

elon-musk	37
facebook	1109
ebola	261
ipad	286
iphone	578
tornado	51
sharknado	25
taylor-swift	77

I think the results makes sense. For example, Facebook is a very popular webpage and one should think there would be a lot of news worthy events which make Facebook appear in the news very often. Adding to this, is the period which the data covers. This is the time period in which facebook filed for IPO and went public. The debut was a little bumpy and stock prices gyrated which all contributed to facebook being in the news quite a bit.

Another example is *ebola*. This was the period in which there was a severe outbreak of the desease that made the word very news worthy.

Question 5

First the groupping and calculations:

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

```
NewsDataDT <- cbind.data.frame(NewsDataD,ExtractedTopics)
x <- with(NewsDataDT,split(ExtractedTopics,list(year,month)))
sn <- paste(names(x),"1",sep = ".") %>% ymd() %>% sort()
sn <- sub("\\.0",".",sub("-",".",sub("-01$","",sn)))
```

We can build monthly tables. The following snippet produces the table that can be later be extracted and presented if neccessary.

```
tables <- list()
s <- c("elon-musk", "facebook", "ebola", "ipad", "iphone", "tornado", "sharknado", "taylor-swift")
for(n in sn){
  res <- lapply(as.vector(x[[n]]),is_in_there,s)
  df <- setNames(do.call(rbind.data.frame,res),s)
  t <- apply(df,2,sum)
  t <- as.data.frame(t)
  tables[[n]] <- t
}
```

Here are the a few of the tables:

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
for(i in 1:4)
grid.arrange(tableGrob(tables[[i]],cols = sn[i]),nrow = 1)
```

	2013.1
elon−musk	0
facebook	79
ebola	0
ipad	12
iphone	37
tornado	0
sharknado	0
taylor−swift	1

	2013.2
elon−musk	2
facebook	57
ebola	0
ipad	14
iphone	22
tornado	0
sharknado	0
taylor−swift	2

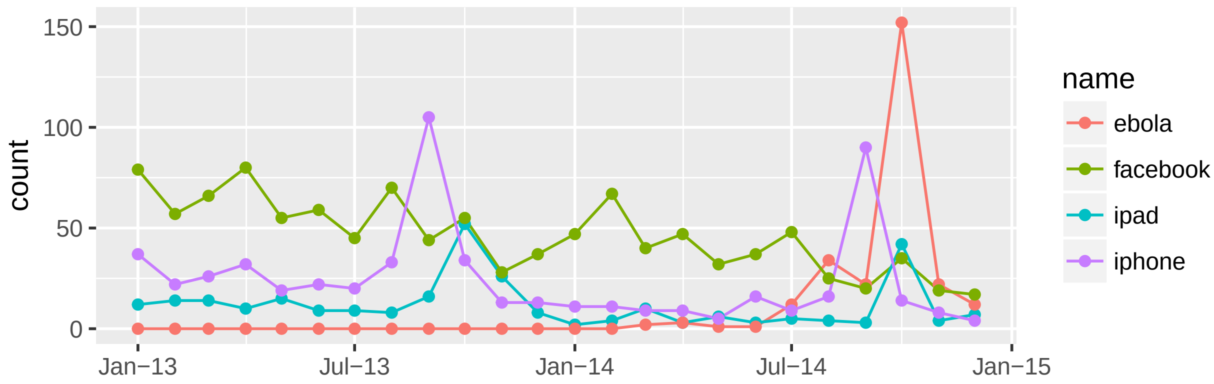
	2013.3
elon−musk	2
facebook	66
ebola	0
ipad	14
iphone	26
tornado	0
sharknado	0
taylor−swift	2

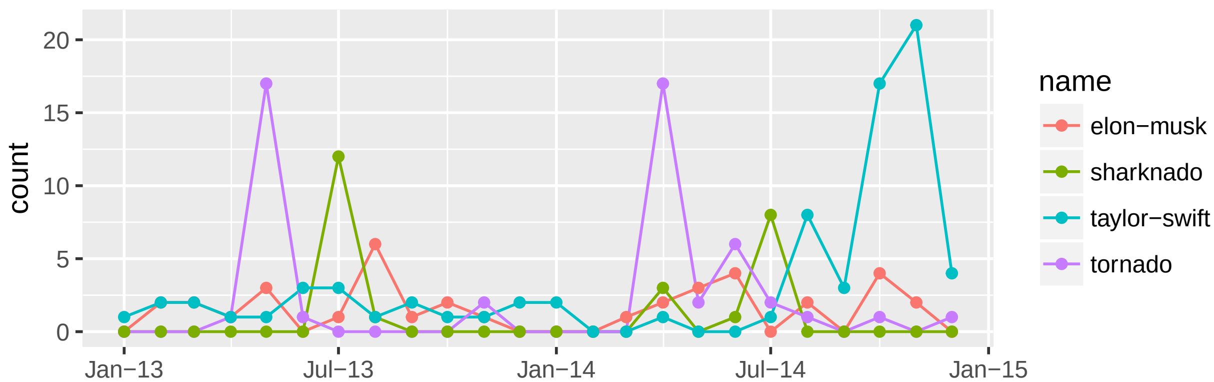
	2013.4
elon−musk	1
facebook	80
ebola	0
ipad	10
iphone	32
tornado	1
sharknado	0
taylor−swift	1

The frequency changes from month-to-month based on events that happened in that month. For example a month in which a new iphone has been released shows a spike in the iphone frequency. Here is two graphs showing the changes in frequency over time. Again, we can see from this graph that there are jumps in popularity around the times of major events. The noticable exception is facebook that enjoys a high ans stable popularity over time.

```
x <- with(NewsDataDT,split(ExtractedTopics,list(year,month)))
sd <- paste(names(x),"1",sep = ".") %>% ymd() %>% sort()
sd <- as.Date(sd)

p1 <- c("ipad","ebola","iphone","facebook")
Numof_ph <- lapply(p1,phrase_num)
df <- data.frame(Numof_ph[[1]]$Date)
for(i in 1:length(Numof_ph)){
  df <- cbind.data.frame(df,Numof_ph[[i]][,2])
}
n <- c("Date",p1)
colnames(df) <- n
df$Date <- sd
xx <- df %>% gather(name,count,ipad:facebook)
g1 <- ggplot(data = xx, aes(x = Date, y = count, group = name, colour = name)) +geom_line()+geom_point(


p2 <- c("elon-musk","tornado","sharknado","taylor-swift")
Numof_ph2 <- lapply(p2,phrase_num)
df2 <- data.frame(Numof_ph2[[1]]$Date)
for(i in 1:length(Numof_ph2)){
  df2 <- cbind.data.frame(df2,Numof_ph2[[i]][,2])
}
n2 <- c("Date",p2)
colnames(df2) <- n2
df2$Date <- sd
```

```
xx2 <- df2 %>% gather(name,count,2:5)
g2 <- ggplot(data = xx2, aes(x = Date, y = count, group = name, colour = name)) +geom_line()+geom_point

#putting two graphs together
grid.arrange(g1, g2, ncol=1)
```





Question 6

We first perform that calculations and then follow that with a discussion.

```
dayN <- c("weekday_is_sunday", "weekday_is_monday", "weekday_is_tuesday","weekday_is_wednesday","weekda
dayData <- NewsData[,dayN]
df <- apply(dayData,2,sum)
#sum(df/7)
kable(df, caption = "Total urls")
```

Table 3: Total urls

weekday_is_sunday	2737
weekday_is_monday	6661
weekday_is_tuesday	7390
weekday_is_wednesday	7435
weekday_is_thursday	7267
weekday_is_friday	5701
weekday_is_saturday	2453

From the data one can observe that number of shared urls are almost the same during the weekdays, but start to go down on friday. During the weekend activities are less than half of a typical weekday. This finding is what one normally expects.

Another way to see this result is to look at the averages. On average 5663 URLs are shared each day. The weekend averge of 2595 is well below this number and the weekday average of 6891, is well above it.

For others we use a little function in FunRs.R, *Forq6_8* to summerize the results and answer the questions.

```
results <- Forq6_8("num_videos")
Daily_Ave <- results$aved
Weekday_Ave <- results$avweekd
Weekend_Ave <- results$avweekendd
kable(results$dayofweek,caption = "num_videos")
```

Table 4: num_videos

	dayNum
weekday_is_sunday	2819
weekday_is_monday	8901
weekday_is_tuesday	9664
weekday_is_wednesday	9204
weekday_is_thursday	8852
weekday_is_friday	7324
weekday_is_saturday	2786

```
Daily_Ave
```

```
## [1] 7078.571
```

```
Weekday_Ave
```

```
## [1] 8789
```

```
Weekend_Ave
```

```
## [1] 2802.5
```

For Average num_images:

```
results <- Forq6_8("num_imgs")
Daily_Ave <- results$aved
Weekday_Ave <- results$avweekd
Weekend_Ave <- results$avweekendd
kable(results$dayofweek,caption = "num_imgs")
```

Table 5: num_imgs

	dayNum
weekday_is_sunday	16054
weekday_is_monday	29622
weekday_is_tuesday	33098
weekday_is_wednesday	30613
weekday_is_thursday	32280
weekday_is_friday	25035
weekday_is_saturday	13446

```
Daily_Ave
```

```
## [1] 25735.43
```

```
Weekday_Ave
```

```
## [1] 30129.6
```

```
Weekend_Ave
```

```
## [1] 14750
```

For Average abs_title_subjectivity:

```
results <- Forq6_8("abs_title_subjectivity")
Daily_Ave <- results$aved
Weekday_Ave <- results$avweekd
Weekend_Ave <- results$avweekendd
kable(results$dayofweek,caption = "abs_title_subjectivity")
```

Table 6: abs_title_subjectivity

	dayNum
weekday_is_sunday	883.1936
weekday_is_monday	2270.3192
weekday_is_tuesday	2559.9802
weekday_is_wednesday	2565.1790
weekday_is_thursday	2496.8950
weekday_is_friday	1973.8736
weekday_is_saturday	802.5737

```
Daily_Ave
```

```
## [1] 1936.002
```

```
Weekday_Ave
```

```
## [1] 2373.249
```

```
Weekend_Ave
```

```
## [1] 842.8837
```

Average abs_title_sentiment_polarity:

```
results <- Forq6_8("abs_title_sentiment_polarity")
Daily_Ave <- results$aved
Weekday_Ave <- results$avweekd
Weekend_Ave <- results$avweekendd
kable(results$dayofweek,caption = "abs_title_sentiment_polarity")
```

Table 7: abs_title_sentiment_polarity

	dayNum
weekday_is_sunday	504.1440
weekday_is_monday	1005.0527
weekday_is_tuesday	1143.2529
weekday_is_wednesday	1118.3505

	dayNum
weekday_is_thursday	1118.6678
weekday_is_friday	881.2746
weekday_is_saturday	416.2453

```
Daily_Ave
```

```
## [1] 883.8554
```

```
Weekday_Ave
```

```
## [1] 1053.32
```

```
Weekend_Ave
```

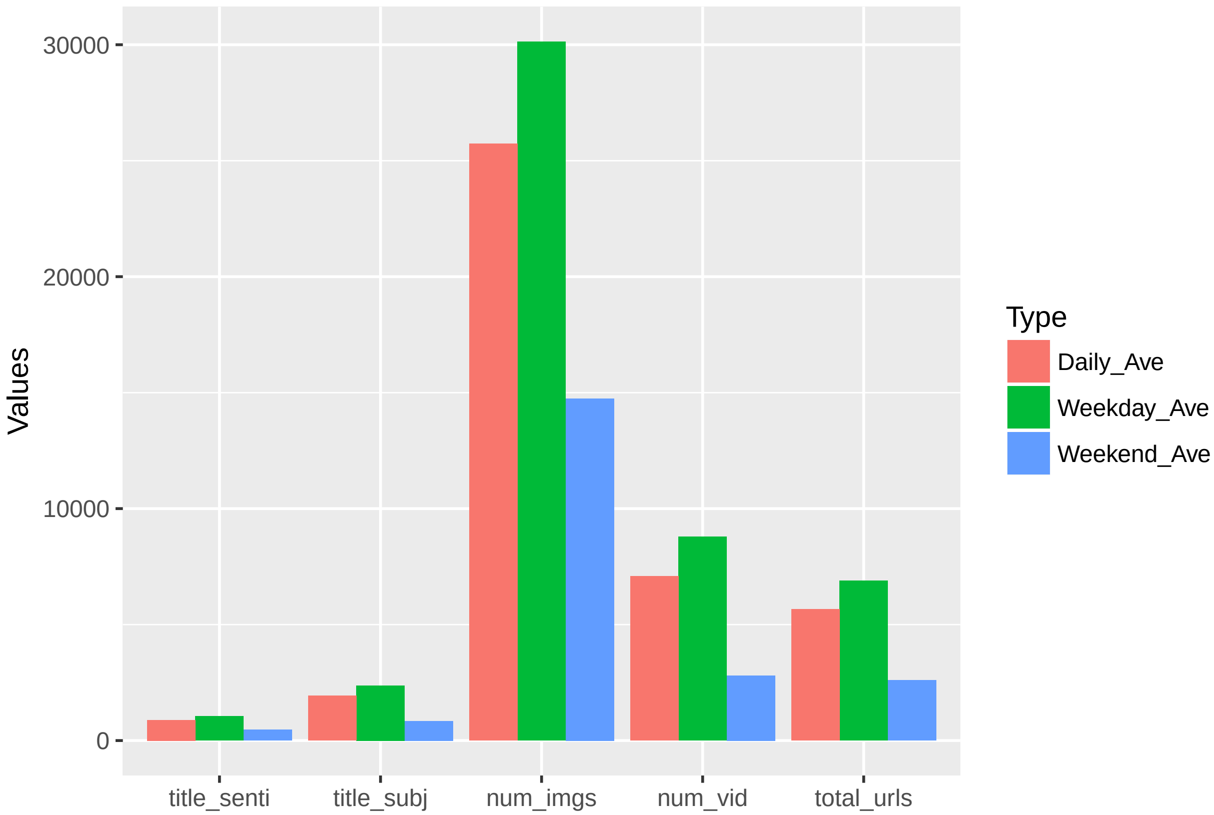
```
## [1] 460.1947
```

Looking at the results we realize that overall activity on the weekend is well below the activity on the weekdays for all these categories. Here is summaraized results:

```
Type_of_Ave <-c("Daily_Ave","Weekday_Ave","Weekend_Ave")
Total_urls <- c(5663,6891,2595)
num_images <- c(25735.43,30129.6,14750)
num_videos <- c(7078.571,8789,2802.5)
abs_title_subjectivity <- c(1936,2373.25,842.88)
abs_title_sentiment<- c(883.86,1053.32,460.19)
Averages <- rbind.data.frame(Total_urls,num_images,num_videos,abs_title_subjectivity,abs_title_sentiment
colnames(Averages)<-Type_of_Ave
Averages <- cbind.data.frame(Parameter = c('Total_urls','num_images','num_videos','abs_title_subjectivi
kable(Averages)
```

Parameter	Daily_Ave	Weekday_Ave	Weekend_Ave
Total_urls	5663.000	6891.00	2595.00
num_images	25735.430	30129.60	14750.00
num_videos	7078.571	8789.00	2802.50
abs_title_subjectivity	1936.000	2373.25	842.88
abs_title_sentiment	883.860	1053.32	460.19

```
meltedData <- Averages %>% gather(Type,Values,2:4)
g3 <- ggplot(meltedData
             ,aes(Parameter, Values, fill = Type))+geom_bar(stat ='identity', position = 'dodge')
g3 <- g3+scale_x_discrete(labels = c("title_senti","title_subj","num_imgs","num_vid","total_urls"))+xlal
g3
```



Question 7

First we perform the calculations and then will analyze the results. These are for data channels of Entertainment, Lifestyle, Tech and World. Since they all use the same code, I suppress the codes to be printed.

Table 9: data_channel_is_lifestyle

	dayNum
weekday_is_sunday	210
weekday_is_monday	322
weekday_is_tuesday	334
weekday_is_wednesday	388
weekday_is_thursday	358
weekday_is_friday	305
weekday_is_saturday	182

```
## [1] 299.8571
```

```
## [1] 341.4
```

```
## [1] 196
```

Table 10: data_channel_is_entertainment

	dayNum
weekday_is_sunday	536

	dayNum
weekday_is_monday	1358
weekday_is_tuesday	1285
weekday_is_wednesday	1295
weekday_is_thursday	1231
weekday_is_friday	972
weekday_is_saturday	380

```
## [1] 1008.143
```

```
## [1] 1228.2
```

```
## [1] 458
```

Table 11: data_channel_is_tech

	dayNum
weekday_is_sunday	396
weekday_is_monday	1235
weekday_is_tuesday	1474
weekday_is_wednesday	1417
weekday_is_thursday	1310
weekday_is_friday	989
weekday_is_saturday	525

```
## [1] 1049.429
```

```
## [1] 1285
```

```
## [1] 460.5
```

Table 12: data_channel_is_world

	dayNum
weekday_is_sunday	567
weekday_is_monday	1356
weekday_is_tuesday	1546
weekday_is_wednesday	1565
weekday_is_thursday	1569
weekday_is_friday	1305
weekday_is_saturday	519

```
## [1] 1203.857
```

```
## [1] 1468.2
```

```
## [1] 543
```

Again the observation is the same. The activities in the weeked days are substantially below those of weekdays.

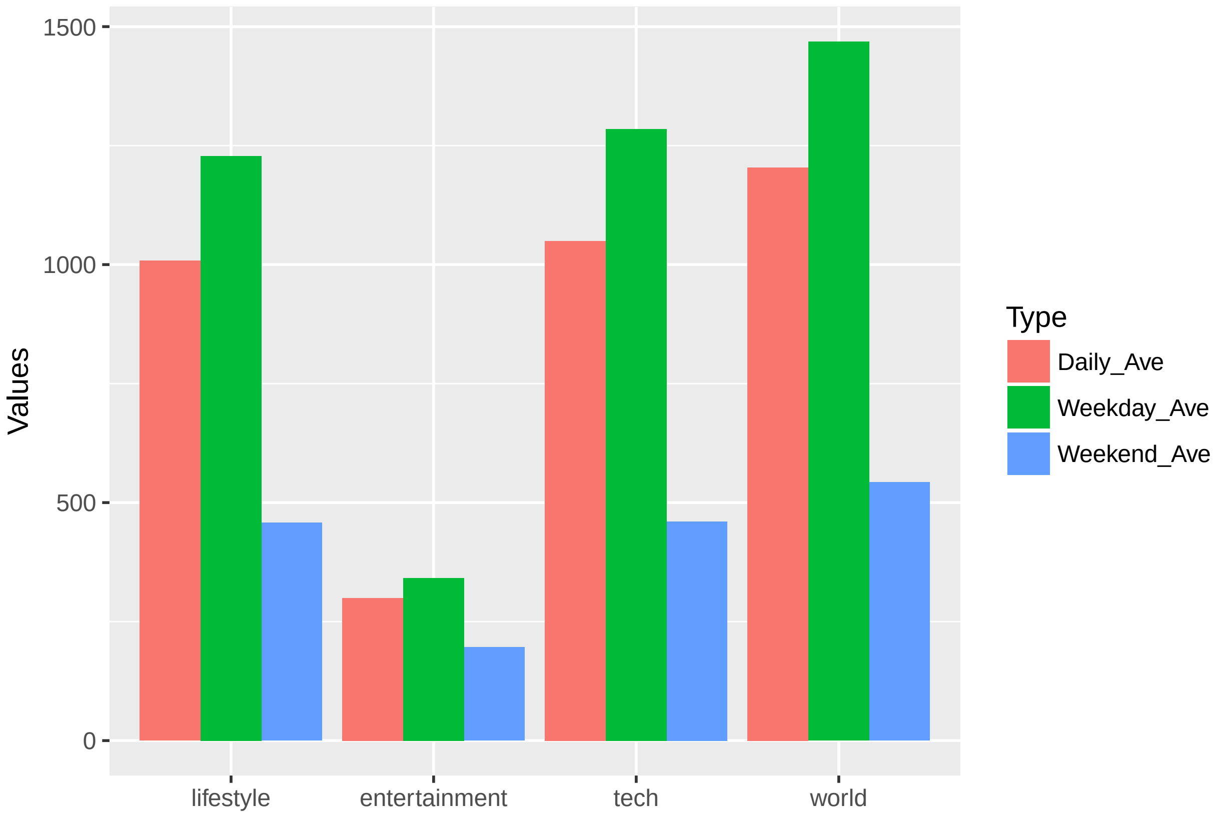
Here is summary of the results:

```
data_channel_is_lifestyle <-c(299.86,341.4,196)
data_channel_is_entertainment <-c(1008.14,1228.2,458)
data_channel_is_tech<-c(1049.43,1285,460.5)
data_channel_is_world<-c(1203.86,1468.2,543)
```

```
Averages <- rbind.data.frame(data_channel_is_lifestyle,data_channel_is_entertainment,data_channel_is_te
colnames(Averages)<-Type_of_Ave
Averages <- cbind.data.frame(Parameter = c('data_channel_is_lifestyle','data_channel_is_entertainment',
kable(Averages)
```

Parameter	Daily_Ave	Weekday_Ave	Weekend_Ave
data_channel_is_lifestyle	299.86	341.4	196.0
data_channel_is_entertainment	1008.14	1228.2	458.0
data_channel_is_tech	1049.43	1285.0	460.5
data_channel_is_world	1203.86	1468.2	543.0

```
meltedData <- Averages %>% gather(Type,Values,2:4)
g4 <- ggplot(meltedData
             ,aes(Parameter, Values, fill = Type))+geom_bar(stat ='identity', position = 'dodge')
g4 <- g4+scale_x_discrete(labels = c("lifestyle","entertainment","tech","world"))+xlab("")
g4
```



Question 8

I was expecting to see a difference between results of some of the question 7 and 6. One thinks entertaiment and lifestyle are type of news that people pay more attention in the weekend, and therefore should be a higher sharing of them.

I think after all most people, including media and news works, are off on weekend are they rather to spend time with family and things like that. Therefore the overal activity derops in the weekends. One can guess that some people shift and prepare what should be consumed for the weekends in the weekdays.