Merry\_Happy\_Christmas

Mark Kuether

12/11/2019

## Load Libraries

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(readr)  
library(ggformula)

## Loading required package: ggplot2

## Loading required package: ggstance

##   
## Attaching package: 'ggstance'

## The following objects are masked from 'package:ggplot2':  
##   
## geom\_errorbarh, GeomErrorbarh

##   
## New to ggformula? Try the tutorials:   
## learnr::run\_tutorial("introduction", package = "ggformula")  
## learnr::run\_tutorial("refining", package = "ggformula")

## Constants from external sources:

**Number of UK and US Twitter Users** <https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/>

**Number of Canadian Twitter Users** <https://www.statista.com/statistics/303875/number-of-twitter-users-canada/>

**Number of Australian Twitter Users**

<https://www.socialmedianews.com.au/social-media-statistics-australia-january-2019/>

#Units in Millions of Users  
US <- 48.35  
UK <- 13.7  
CA <- 7.6  
AU <- 4.7  
ALL <- US+UK+CA+AU  
  
#Calculate ratios for use in p-test.  
UK\_US <- UK/(UK+US)  
UK\_ALL <- UK/ALL  
UKAU\_ALL <- (UK+AU)/ALL  
UKAUCA\_ALL <- (UK+AU+CA)/ALL  
  
#Calculate relative percentages for graphs  
UKUSp <- (UK/(UK+US))\*100  
USUKp <- 100-UKUSp  
  
UKALLp <- (UK/ALL)\*100  
ALLUKp <- 100-UKALLp  
  
UKAU\_CAUSp <- ((UK+AU)/ALL)\*100  
CAUS\_UKAUp <- 100-UKAU\_CAUSp  
  
UKAUCA\_USp <- ((UK+AU+CA)/ALL)\*100  
US\_UKAUCAp <- 100-UKAUCA\_USp

## Load and format data from CSV.

#This data set contains 56,265 recordes with 5 variables.  
raw\_tweet\_df <- read.csv("tweet\_summary.csv", header = TRUE,  
 stringsAsFactors = FALSE)  
  
#Set the boolean columns "merry" and "happy" to boolean types. This makes it easier to work with them in R.  
raw\_tweet\_df <- raw\_tweet\_df %>%   
 mutate(merry = as.logical(merry))  
  
raw\_tweet\_df <- raw\_tweet\_df %>%   
 mutate(happy = as.logical(happy))  
  
#We remove duplicate users. If users tweet multiple times, it could throw off the statistics, so we only count one tweet per user.  
  
#This removes 7,350 tweets = 48,915 records.  
tweet\_df <- raw\_tweet\_df[!duplicated(raw\_tweet\_df$author),]  
  
#Remove all records with both "Happy" and "Merry" set the same. This may happen if the tweet includes both sentiments, or if the Python code did not recognize the sentiment.  
  
#This removes 728 tweets = 48,187 records.  
tweet\_df <- tweet\_df[tweet\_df$happy != tweet\_df$merry,]

## Statistical Testing

We use the a p-test to compare the proportions of the counted sentiments to the user proportions set at the top.

H0: p(“happy”) = p(UK users)

H1: p(“happy”) != p(UK users)

We analyze this across four possible variations

* UK vs US only
* UK vs US + CA + AU
* UK + AU vs US + CA
* UK + AU + CA vs US

The p-test uses single values, so we need to extract those first.

#Generate the summary of the two values. Since we have removed all same values between happy and merry columns, we only need to use the happy colunn for analysis.  
s\_summary <- tweet\_df %>%   
 group\_by(happy) %>%   
 summarize(value = n())  
  
#This returns single values of  
happy <- s\_summary$value[s\_summary$happy==TRUE]  
total <- sum(s\_summary$value)  
p\_success <- happy/total  
  
cat("Number of 'Happy' sentiments:",happy)

## Number of 'Happy' sentiments: 5159

cat("\n")

cat("Number of total tweets:", total)

## Number of total tweets: 48187

cat("\n")

cat("Observed propability:",p\_success)

## Observed propability: 0.1070621

cat("\n")

#This ensures our sample size is large enough.   
bell\_shaped\_test <- total\*p\_success \*(1-p\_success)  
cat("Sample Size Test - np(1-p) =",bell\_shaped\_test)

## Sample Size Test - np(1-p) = 4606.667

cat("\n")

test\_uk\_us <- prop.test(happy,total,  
 p=UK\_US,  
 alternative="two.sided",  
 correct=FALSE)  
  
test\_uk\_all <- prop.test(happy,total,  
 p=UK\_ALL,  
 alternative = "two.sided",  
 correct=FALSE)  
  
test\_ukau\_all <- prop.test(happy,total,  
 p=UKAU\_ALL,  
 alternative="two.sided",  
 correct=FALSE)  
  
test\_ukauca\_all <- prop.test(happy,total,  
 p=UKAUCA\_ALL,  
 alternative = "two.sided",  
 correct=FALSE)  
  
#Quick and Dirty - All results indicate Reject Null Hyp:  
cat("test\_uk\_us:\n")

## test\_uk\_us:

test\_uk\_us

##   
## 1-sample proportions test without continuity correction  
##   
## data: happy out of total, null probability UK\_US  
## X-squared = 3622.7, df = 1, p-value < 2.2e-16  
## alternative hypothesis: true p is not equal to 0.2207897  
## 95 percent confidence interval:  
## 0.1043327 0.1098541  
## sample estimates:  
## p   
## 0.1070621

cat("test\_uk\_all:\n")

## test\_uk\_all:

test\_uk\_all

##   
## 1-sample proportions test without continuity correction  
##   
## data: happy out of total, null probability UK\_ALL  
## X-squared = 1910.7, df = 1, p-value < 2.2e-16  
## alternative hypothesis: true p is not equal to 0.1842636  
## 95 percent confidence interval:  
## 0.1043327 0.1098541  
## sample estimates:  
## p   
## 0.1070621

cat("test\_ukau\_all:\n")

## test\_ukau\_all:

test\_ukau\_all

##   
## 1-sample proportions test without continuity correction  
##   
## data: happy out of total, null probability UKAU\_ALL  
## X-squared = 5101.6, df = 1, p-value < 2.2e-16  
## alternative hypothesis: true p is not equal to 0.2474781  
## 95 percent confidence interval:  
## 0.1043327 0.1098541  
## sample estimates:  
## p   
## 0.1070621

cat("test\_ukauca\_all:\n")

## test\_ukauca\_all:

test\_ukauca\_all

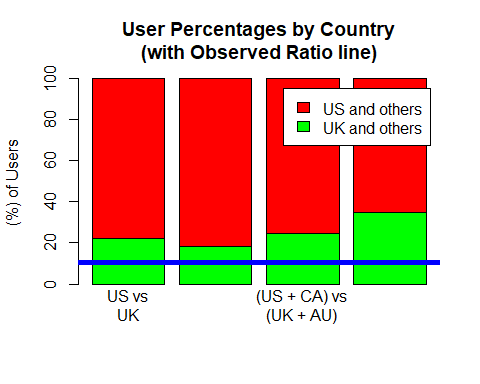
##   
## 1-sample proportions test without continuity correction  
##   
## data: happy out of total, null probability UKAUCA\_ALL  
## X-squared = 12475, df = 1, p-value < 2.2e-16  
## alternative hypothesis: true p is not equal to 0.3496974  
## 95 percent confidence interval:  
## 0.1043327 0.1098541  
## sample estimates:  
## p   
## 0.1070621

## Graphical Review

Graphs illustrate that the observed sentiment values are far below the population percentages of the users. This does not support the hypthosis that the sentiment is cultrually based between the US and UK.

(**NOTE:** labels are hidden with knitted graph, but the labels did show up when run from a stand alone R script.)

#Visual comparison of user proportions and observed proportion.  
prop\_vector <- c(UKUSp,UKALLp,UKAU\_CAUSp,UKAUCA\_USp,  
 USUKp,ALLUKp,CAUS\_UKAUp,US\_UKAUCAp)  
  
dim\_names <- list(c("POPULATION","TOTALS"),c("US vs\nUK","(US + AU + CA) vs\n(UK)","(US + CA) vs\n(UK + AU)","(US) vs\n(UK + AU + CA)"))  
  
data\_matrix <- matrix(prop\_vector,nrow=2,ncol=4,dimnames = dim\_names,byrow=TRUE)  
bar\_colors <- c("Green","Red")  
legend\_text <- c("UK and others","US and others")  
  
mybarplot <- barplot(data\_matrix,col=bar\_colors,  
 ylab="(%) of Users",  
 main="User Percentages by Country\n(with Observed Ratio line)",  
 legend.text=legend\_text) + abline(h=p\_success\*100,col="Blue",lwd=5)

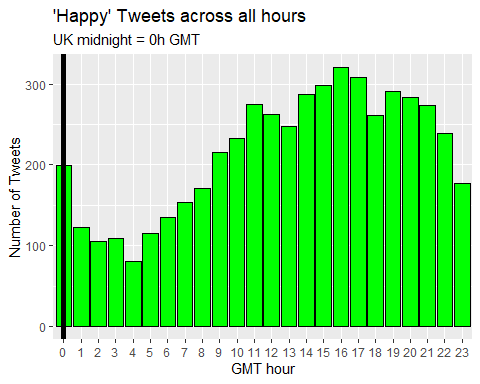


## Additional Analysis

With the negative result, we review the tweet times to see if the times match up geographically with the volume observed. The “Happy” sentiment does coincide with the UK times.

Tweet times are recorded based on GMT.

#From the time column, we extract the hour.  
#I could not figure out a way to extract this using vectors.  
#I had to resort to a loop which is very slow. Sorry. :(  
for(this\_row in (1:dim(tweet\_df)[1])){  
 tweet\_df$hour\_slot[this\_row] = strsplit(  
 strsplit(tweet\_df$date[this\_row]," ",fixed=TRUE)[[1]][2],  
 ":",fixed=TRUE)[[1]][1]  
}  
  
hour\_levels = c("0","1","2","3","4","5","6","7","8","9",  
 "10","11","12","13","14","15","16","17",  
 "18","19","20","21","22","23")  
  
tweet\_df$hour\_slot = factor(tweet\_df$hour\_slot,levels=hour\_levels,ordered=TRUE)  
  
#This is not faceted because the 'Happy' sentiment details are lost when graphed at the same scale as the 'Merry' sentiment.  
  
happy\_chart <- tweet\_df %>%   
 filter(happy==TRUE) %>%   
 gf\_bar(~hour\_slot,col="Black", fill="Green") %>%   
 gf\_vline(xintercept = ~1,size=2) %>%   
 gf\_labs(title="'Happy' Tweets across all hours",  
 subtitle="UK midnight = 0h GMT",  
 y = "Number of Tweets",  
 x = "GMT hour")  
  
merry\_chart <- tweet\_df %>%   
 filter(happy==FALSE) %>%   
 gf\_bar(~hour\_slot,col = "Black",fill="Red") %>%   
 gf\_vline(xintercept = ~7,size=2) %>%   
 gf\_labs(title="'Merry' Tweets across all hours",  
 subtitle="Chicago midnight = -6h GMT",  
 y = "Number of Tweets",  
 x = "GMT hour")  
  
happy\_chart



merry\_chart

