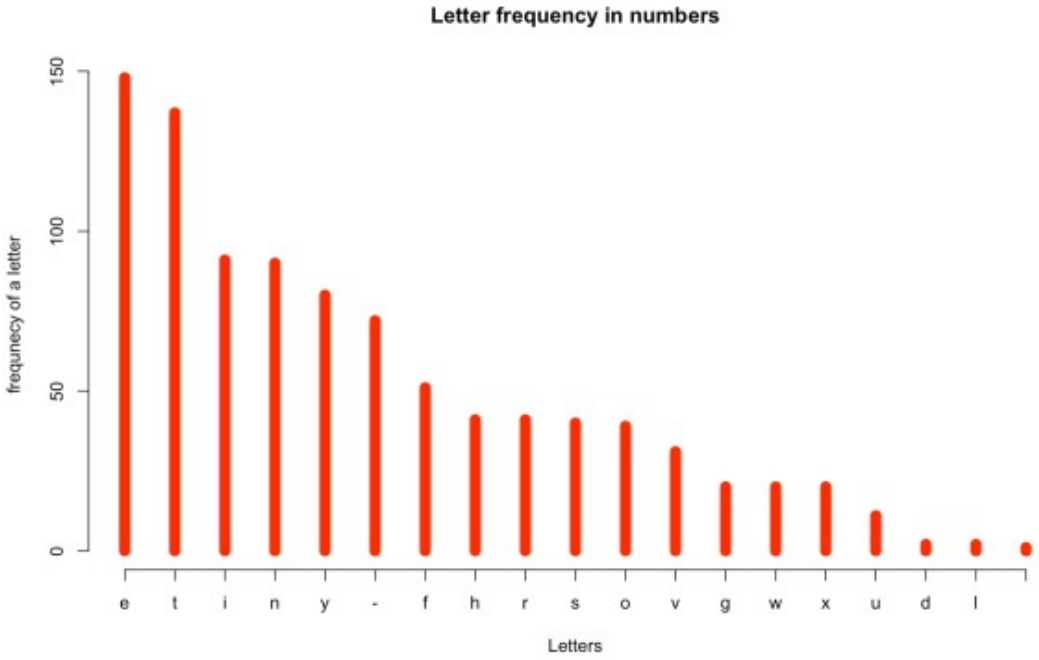
So here is a useless fact. There is no letter A in numbers – written as words – from 1 to 100. And of course, we wanted to put this into the test and check if it holds the water.

And sure, there is the result:



Two sets of functions were created in this case (it can be stuffed in single one and super simplified, but it is all about uselessness.

Getting the words for number:

```
#function
word_a_number <- function(numb){

  basLet <- c('one','two','three','four','five','six','seven','eight','
nine','ten'
              ,'eleven','twelve','thirteen','fourteen','fifteen','sixteen'
,'seventeen','eighteen','nineteen'
              ,'twenty','thirty','forty','fifty','sixty','seventy','
eighty','ninety','one hundred')
  basNum <- c(1:20,30,40,50,60,70,80,90,100)
  df <- data.frame(num = basNum, let = as.character(basLet))

          if (numb <= 20) {
            im <- df[which(df$num == numb),]$let
            print(paste(im, collapse = NULL))
          } else {
                  if (numb %% 10 == 0){
                    e <- df[which(df$num == numb),]$let
                    print(paste0(e, collapse=NULL))
                  } else {
                    sec <- numb %% 10
                    fir <- as.integer(numb/10)*10
                    f_im <- df[which(df$num == fir),]$let
                    s_im <- df[which(df$num == sec),]$let
                    res <- paste0(f_im,"-",s_im, collapse = NULL)
```

```
                    print(res)
                  }
          }
}
```

This functions iterates through a set of numbers, given a boundaries. And since first ten words and and words from eleven to twenty differ significantly, I wrote it in data frame. Next step is to do the increments of ten. And then hundred, thousand, etc.

Once this function is established, it will return the word for given number. Once this is established, a little helper counter would do the job:

```
#function for  count the frequency
freqLet <- function(x) {
  word <- tolower(unlist(strsplit(x,"")))
  word_table <- table(word)
  ans <- word_table[names(word_table)]
}

getFreq <- function(vect) {
  df <- data.frame(word=as.character(), stringsAsFactors = FALSE)
  for (i in 1:length(vect)) {
    df[i,1] <- as.character(word_a_number(i))
    a <<- freqLet(df$word)
  }
  return(a)
}
```

And once this is loaded into environment, we need to run the complete set:

```
######### Let's check the complete set of numbers

# Automate the function, get a vector of first 100 numbers
vect <- c(1:100)

#Is there A in first 100 words?
getFreq(vect)
```

And this returns a table of letter frequencies

```
word
     -  d  e  f  g  h  i  l  n  o  r  s  t  u  v  w  x  y
  1 72  2 148 51 20 41 91  2 90 39 41 40 137 11 31 20 20 80
```

There are some gotcha moments 🙂

It is useless (and that's the catch) to write down the words of all the numbers. If you only create a dictionary of non-repetitive words (like the one from 11 to 19 or from 0 to 10), you may check only the in these subsets for presence of letter A