Lab 4

Kristina Arevalo

Tue Sep 29 2020

Contents

Problem 1

Estimate the letter occurrence probabilities of all 26 letters in English by measuring a paragraph of English text from wikipedia. (hint use strsplit() to split a paragraph into individual letters) (1 point).

```
my_paragraph <- "Microorganisms include all unicellular organisms and so are extremely diverse, living i

the_letters <- unlist(strsplit(my_paragraph, split = ""))
wiki <- tolower(the_letters)
letter_counts <- table(wiki)
letter_counts <- data.frame(letter_counts)

letters_only <- letter_counts %>%
  filter(wiki %in%letters == TRUE)

total_letters <- sum(letters_only$Freq)

letters_only2 <- letters_only %>%
  mutate(prob = Freq/total_letters)

letters_only2 %>% head()
```

```
##   wiki Freq       prob
## 1    a   58 0.08909370
## 2    b    6 0.00921659
## 3    c   31 0.04761905
## 4    d   29 0.04454685
## 5    e   72 0.11059908
## 6    f    9 0.01382488
```

Confidence: 50 needed to look into the video for some help

Problem 2

Generate "random" strings of letters that are sampled from a distribution where letter occurrence probability is the same as natural English. Use the probabilities for each letter from this wikipedia article, or use your own estimates from the previous question (2 points).

```
my_letters <- sample(letters_only2$wiki,50*5,replace=TRUE, prob = letters_only2$prob)

my_strings <- matrix(my_letters, ncol=5)

paste(my_strings[1,], collapse="")
```

```
## [1] "leaag"
```

```
random_strings <-c()
for(i in 1:dim(my_strings)[1]){
  random_strings[i] <- paste(my_strings[i,], collapse="")
}

random_strings
```

```
##  [1] "leaag" "naaig" "erned" "oussr" "tumte" "amrot" "mlnhe" "tauoh" "eleet"
## [10] "rthcm" "ramra" "lmcns" "racar" "ondcr" "ievrt" "dmrit" "frasi" "ciwsv"
## [19] "neues" "dsrya" "tciva" "riemw" "isoem" "oacio" "tatsb" "elkse" "uhore"
## [28] "noevu" "tcdot" "fcfrc" "nulpn" "umrrr" "ahnno" "ligis" "pitve" "cmoto"
## [37] "neihi" "aihln" "uasvm" "dtell" "aeeis" "muion" "nrnam" "mssoo" "vnngd"
## [46] "reyen" "iepan" "eoelc" "geoee" "rarii"
```

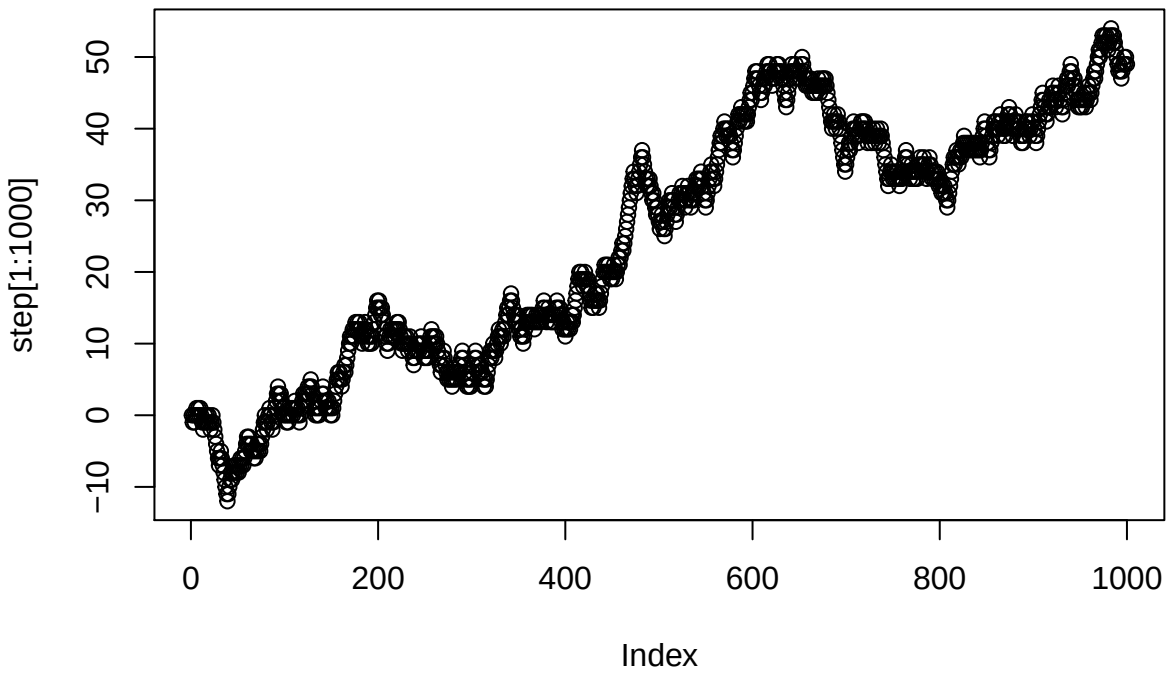
Confidence = 0

Problem 3

Generate a random walk of 10,000 steps. In a random walk, you are simulating the process of randomly taking a step up or down, as if you are on an infinite staircase. At each step you flip a coin. If you get heads you go up one step, if you get tails you go down one step. Start on step 0, then simulate a random walk for 10,000 steps. Your vector should preserve the step number for each step. For example, if the the first three steps were all heads, then the vector would begin with 0,1,2,3, which indicates a single step up each time. Plot the first 1,000 steps. (1 point)

```
step <- c(0)
for(i in 1:10000){
  coin_flip <- sample(c(1,-1),1)
  step[i+1]<- step[i]+ coin_flip
}


plot(step[1:1000])
```



Confidence = 0

Problem 4

What was the most positive and most negative step reached out of 10,000? (1 point)

```
max(step)
```

```
## [1] 231
```

```
min(step)
```

```
## [1] -12
```

confidence = 100

Problem 5

What was the longest run of steps where all steps were positive numbers. For example, in the sequence: 1,2,3,2,1,0,-1,-2,-1,-2,-1,0,1,2,3; the answer is 5 because the first five values were all positive, and this was the longest sequence of positive values. (1 point).

```r
logical_step <- sign(step)

sequence <- c()
counter <- 0
for( i in 1:length(logical_step)){
 if(logical_step[i] == 0){
   sequence <- c(sequence,counter)
   counter <- 0
 } else {
     counter <- counter+logical_step[i]
   }
}

max(sequence)
```

```
## [1] 15
```

confidence = 0