Lab8

Suixin Jiang

11/3/2019

Exercise 2

1.Load data.

```
CSW <- read.table('Collins Scrabble Words (2015).txt')
```

2.Determine the number of words. There are 942 words that violating the spelling rule, and 169 words satisfied the rule.

```
CSW %>%
  filter(str_detect(V1, 'CIE')) %>%
  select(V1) ->
  violation_word
head(violation_word)
```

```
##               V1
## 1       ABBACIES
## 2      ABBOTCIES
## 3    ABERRANCIES
## 4     ABEYANCIES
## 5   ABHORRENCIES
## 6 ABORTIFACIENT
```

```
CSW %>%
  filter(str_detect(V1, 'CEI')) %>%
  select(V1) ->
  satisfaction_word
head(satisfaction_word)
```

```
##              V1
## 1    APPERCEIVE
## 2   APPERCEIVED
## 3   APPERCEIVES
## 4 APPERCEIVING
## 5       CADUCEI
## 6    CALCEIFORM
```

3.There are 18579 words in Collins Scrabble Words contains an "EI" or "IE" pair. After switching "E" and "I" positions, there are 18433 words still valid. Longest (15 letters) and shortest (4 letters) words show below.

```
switch_1 <- as.data.frame(str_replace_all(CSW$V1, 'EI', 'IE'))
names(switch_1)[1] <- 'V1'
switch_2 <- as.data.frame(str_replace_all(CSW$V1, 'IE', 'EI'))
names(switch_2)[1] <- 'V1'
common_1 <- as.data.frame(intersect(CSW$V1, switch_1$V1))
```

```
names(common_1)[1] <- 'V1'
common_2 <- as.data.frame(intersect(CSW$V1, switch_2$V1))
names(common_2)[1] <- 'V1'
CSW %>%
  filter(str_detect(V1, '.(EI|IE).')) %>%
  select(V1) ->
  switch_word
head(switch_word)
```

```
##            V1
## 1    ABBACIES
## 2   ABBOTCIES
## 3      ABEIGH
## 4 ABERNETHIES
## 5 ABERRANCIES
## 6  ABEYANCIES
```

```
common_1 %>%
  filter(str_detect(V1, '.(EI|IE).')) %>%
  select(V1) ->
  valid_1
head(valid_1)
```

```
##            V1
## 1    ABBACIES
## 2   ABBOTCIES
## 3 ABERNETHIES
## 4 ABERRANCIES
## 5  ABEYANCIES
## 6   ABHENRIES
```

```
common_2 %>%
  filter(str_detect(V1, '.(EI|IE).')) %>%
  select(V1) ->
  valid_2
head(valid_2)
```

```
##         V1
## 1   ABEIGH
## 2  ABLEISM
## 3 ABLEISMS
## 4  ABLEIST
## 5 ABLEISTS
## 6   ABSEIL
```

```
valid <- full_join(valid_1, valid_2, by="V1")
```

```
## Warning: Column `V1` joining factors with different levels, coercing to
## character vector
```

```
length(valid$V1)
```

```
## [1] 18433
```

```
a <- valid$V1
longest_word <- a[nchar(a)==max(nchar(a))]
head(longest_word)
```

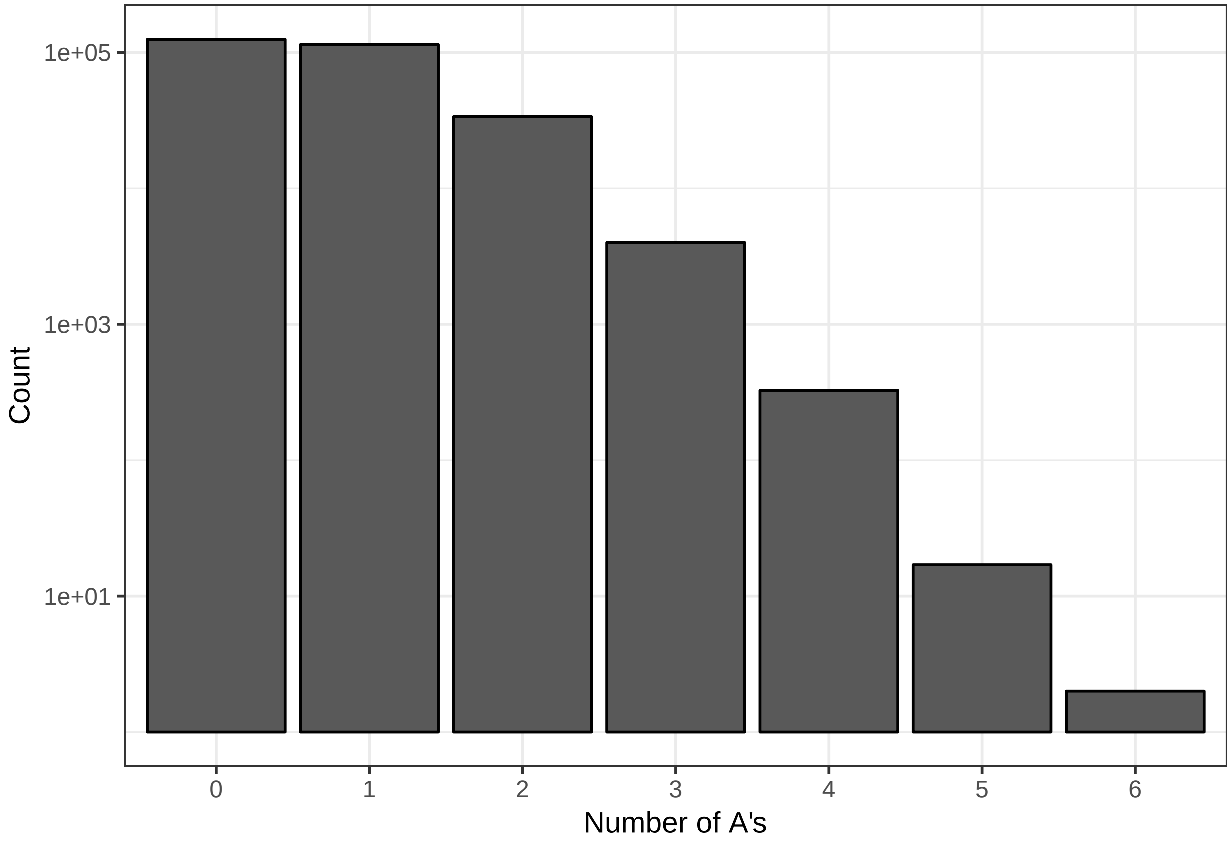
```
## [1] "ABORIGINALITIES" "ABSORBABILITIES" "ABSORBEFACIENTS" "ACCEPTABILITIES"
## [5] "ACCESSIBILITIES" "ACCIDENTALITIES"
```

```
shortest_word <- a[nchar(a)==min(nchar(a))]
head(shortest_word)
```

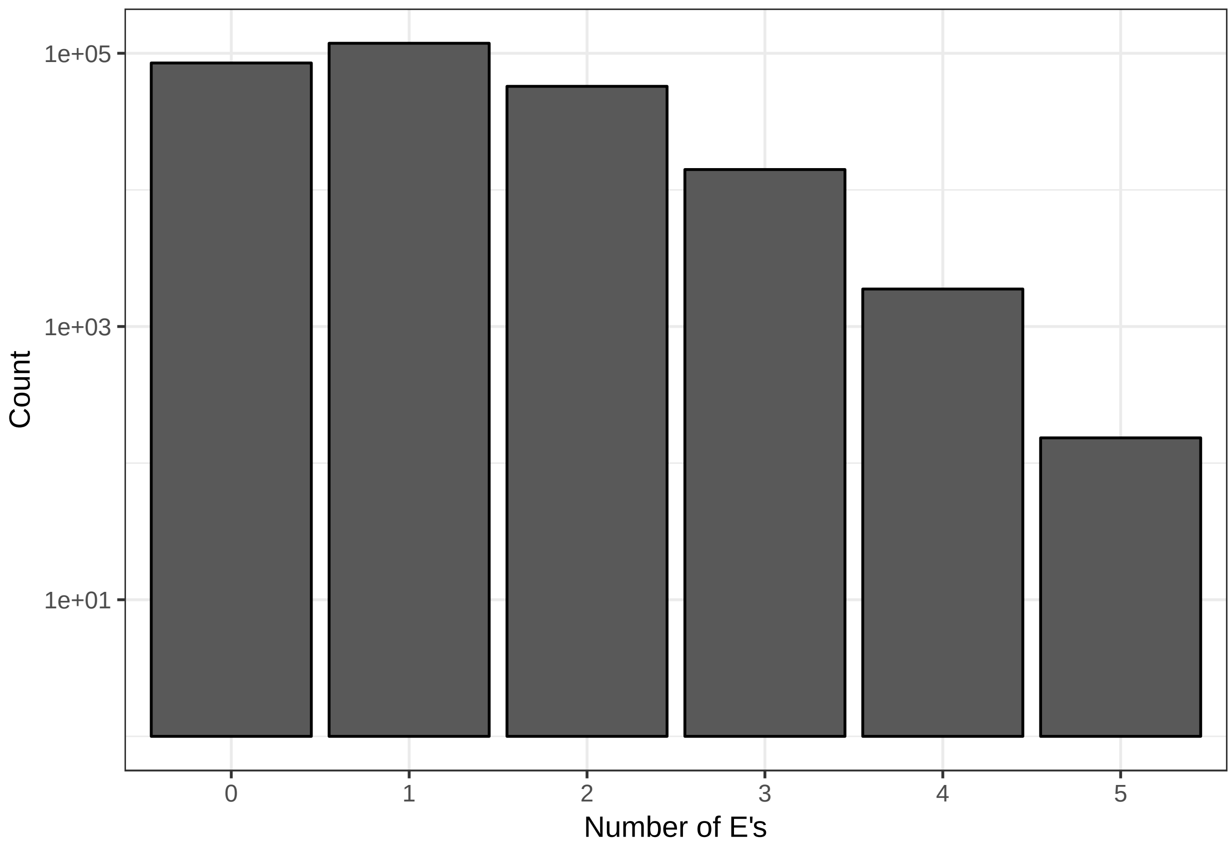
```
## [1] "BIEN" "BIER" "CIEL" "DIEB" "DIED" "DIEL"
```

4.Distribution of each vowel.

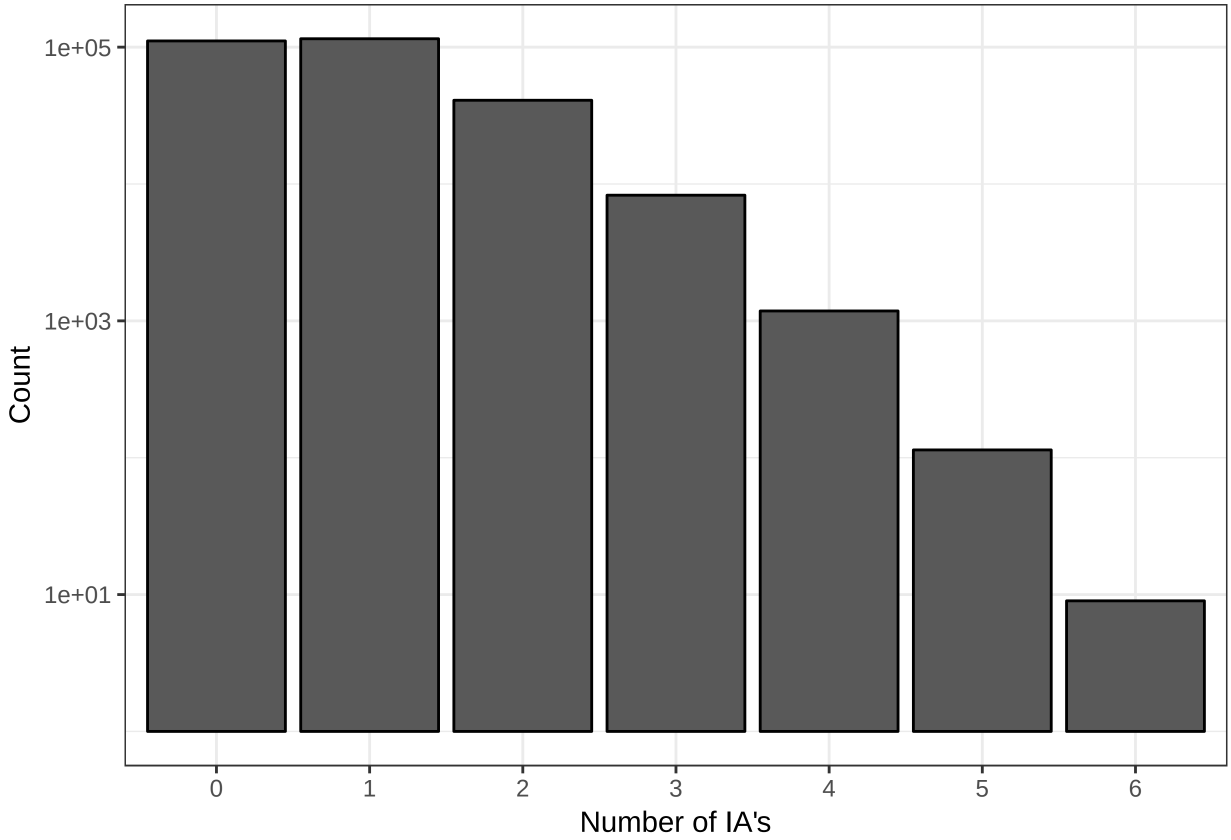
```
A <- as.data.frame(table(str_count(CSW$V1, '[A]')))
ggplot(A, aes(x=Var1, y=Freq)) +
  geom_bar(stat='identity', color='black') +
  theme_bw() +
  scale_x_discrete("Number of A's") +
  scale_y_continuous('Count',  trans = 'log10', breaks = c(10, 1000, 100000),
                     labels = scales::scientific)
```



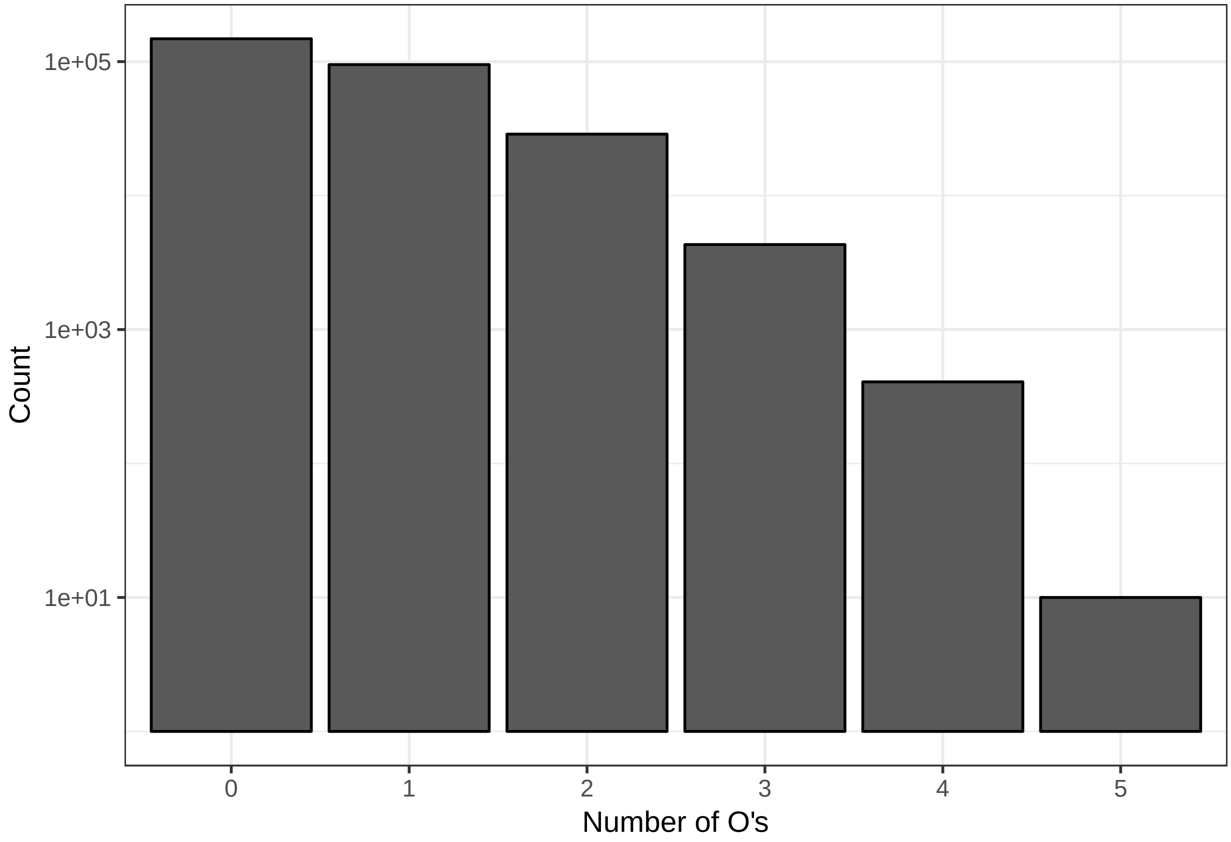
```
E <- as.data.frame(table(str_count(CSW$V1, '[E]')))
ggplot(E, aes(x=Var1, y=Freq)) +
  geom_bar(stat='identity', color='black') +
  theme_bw() +
  scale_x_discrete("Number of E's") +
  scale_y_continuous('Count',  trans = 'log10', breaks = c(10, 1000, 100000),
                     labels = scales::scientific)
```



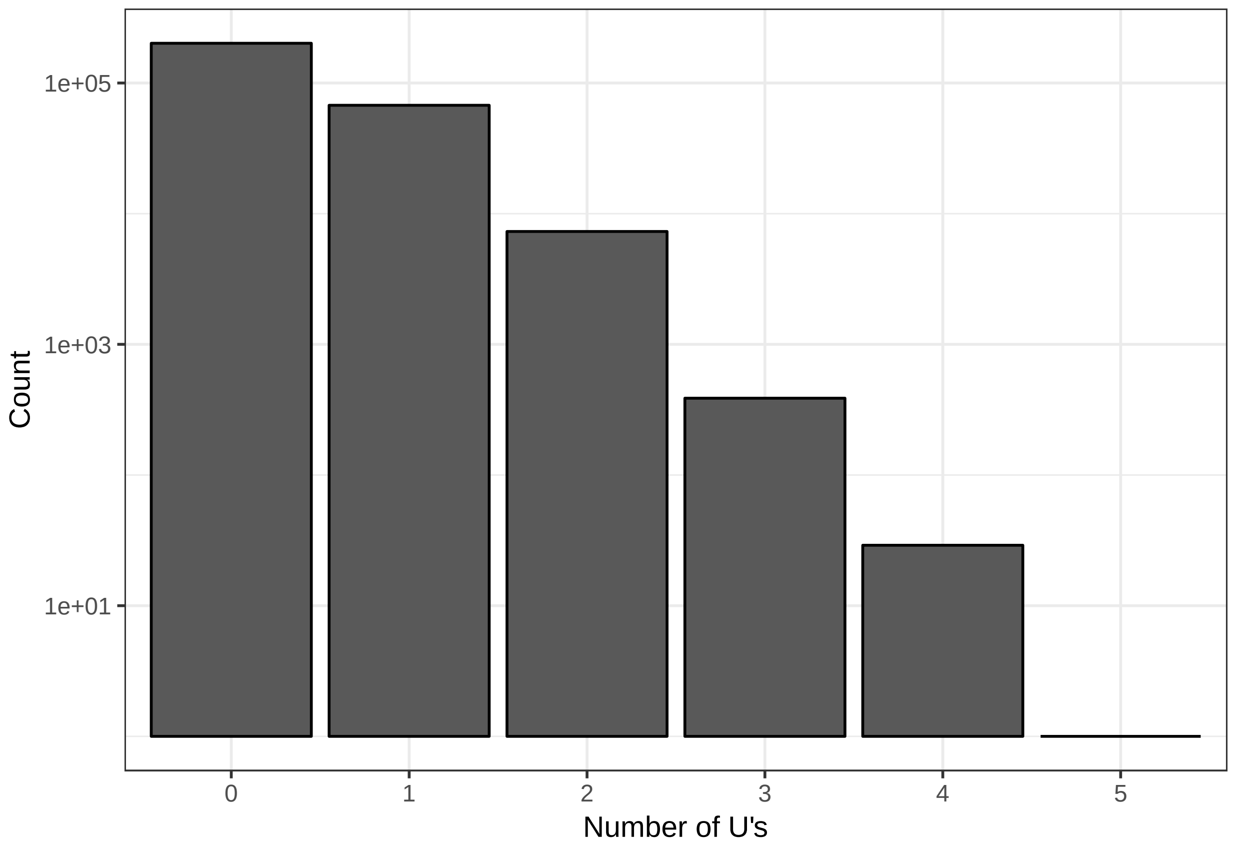
```
I <- as.data.frame(table(str_count(CSW$V1, '[I]')))
ggplot(I, aes(x=Var1, y=Freq)) +
  geom_bar(stat='identity', color='black') +
  theme_bw() +
  scale_x_discrete("Number of IA's") +
  scale_y_continuous('Count',  trans = 'log10', breaks = c(10, 1000, 100000),
                     labels = scales::scientific)
```



```r
O <- as.data.frame(table(str_count(CSW$V1, '[O]')))
ggplot(O, aes(x=Var1, y=Freq)) +
  geom_bar(stat='identity', color='black') +
  theme_bw() +
  scale_x_discrete("Number of O's") +
  scale_y_continuous('Count',  trans = 'log10', breaks = c(10, 1000, 100000),
                     labels = scales::scientific)
```



```
U <- as.data.frame(table(str_count(CSW$V1, '[U]')))
ggplot(U, aes(x=Var1, y=Freq)) +
  geom_bar(stat='identity', color='black') +
  theme_bw() +
  scale_x_discrete("Number of U's") +
  scale_y_continuous('Count',  trans = 'log10', breaks = c(10, 1000, 100000),
                     labels = scales::scientific)
```


1e+05
1e+03
1e+01
Count
0
1
2
3
4
5
Number of U's