```
yelp.review<- stream_in(file("/project/mssphw1/yelpmssp/review.json"),verbose = F)
yelp.review<-flatten(yelp.review)
yelp.business<- stream_in(file("/project/mssphw1/yelpmssp/business.json"),verbose = F)

business.inf<-yelp.business%>% dplyr::select('business_id', 'state')
```

```
yelp.review<-merge(yelp.review,business.inf,'business_id')
```

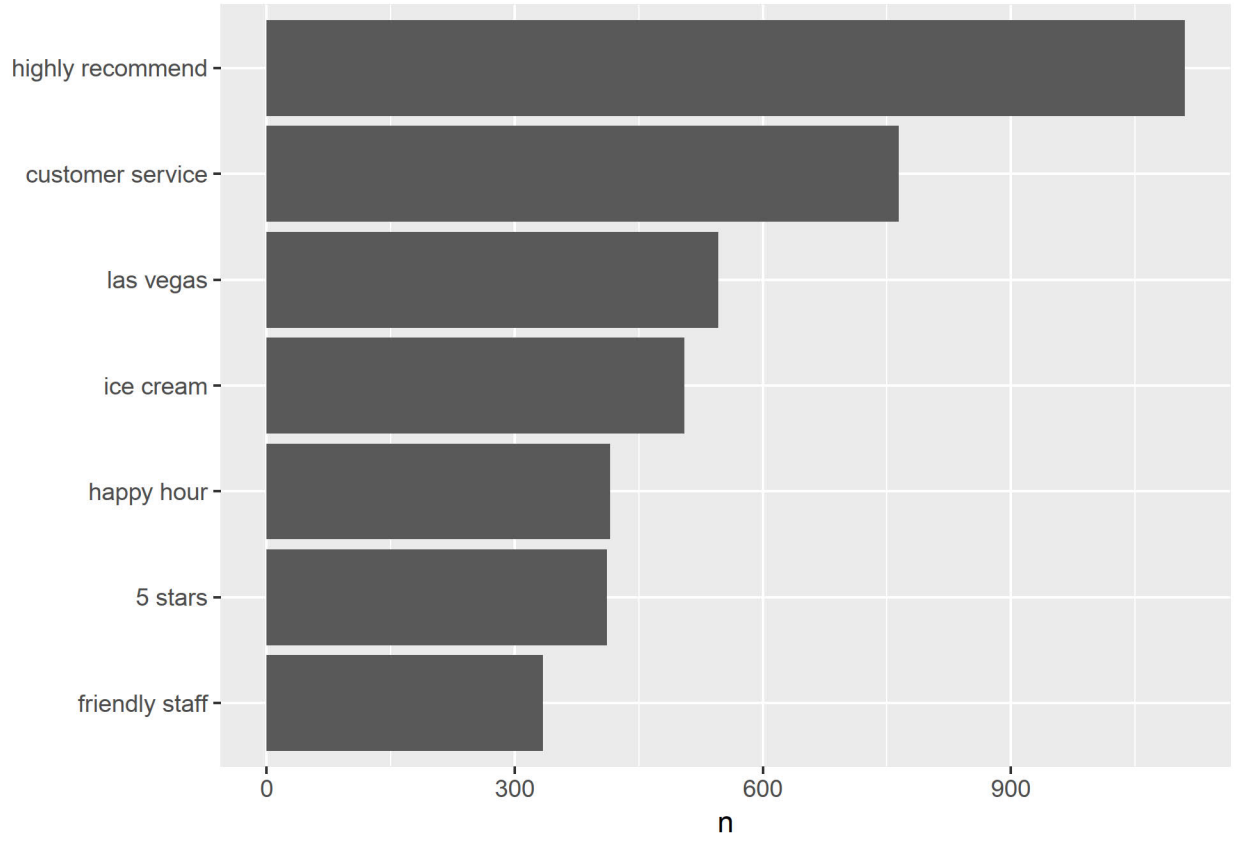
```
review.top<-subset(yelp.review,yelp.review$stars>=4)
review.low<-subset(yelp.review,yelp.review$stars<=2)
```

```
users.review.top<-tibble(useful = review.top$useful,text = review.top$text)
users.review.top$text<-as.character(users.review.top$text)
top.samp<-users.review.top[sample(nrow(users.review.top), 20000, replace=FALSE),]


Count.bigrams <-top.samp %>%
  unnest_tokens(bigram, text, token = "ngrams", n = 2)  %>%
  separate(bigram, c("word1", "word2"), sep = " ") %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word) %>%
  count(word1, word2, sort = TRUE)

united.top.review <- Count.bigrams %>%
  unite(bigram, word1, word2, sep = " ")
```

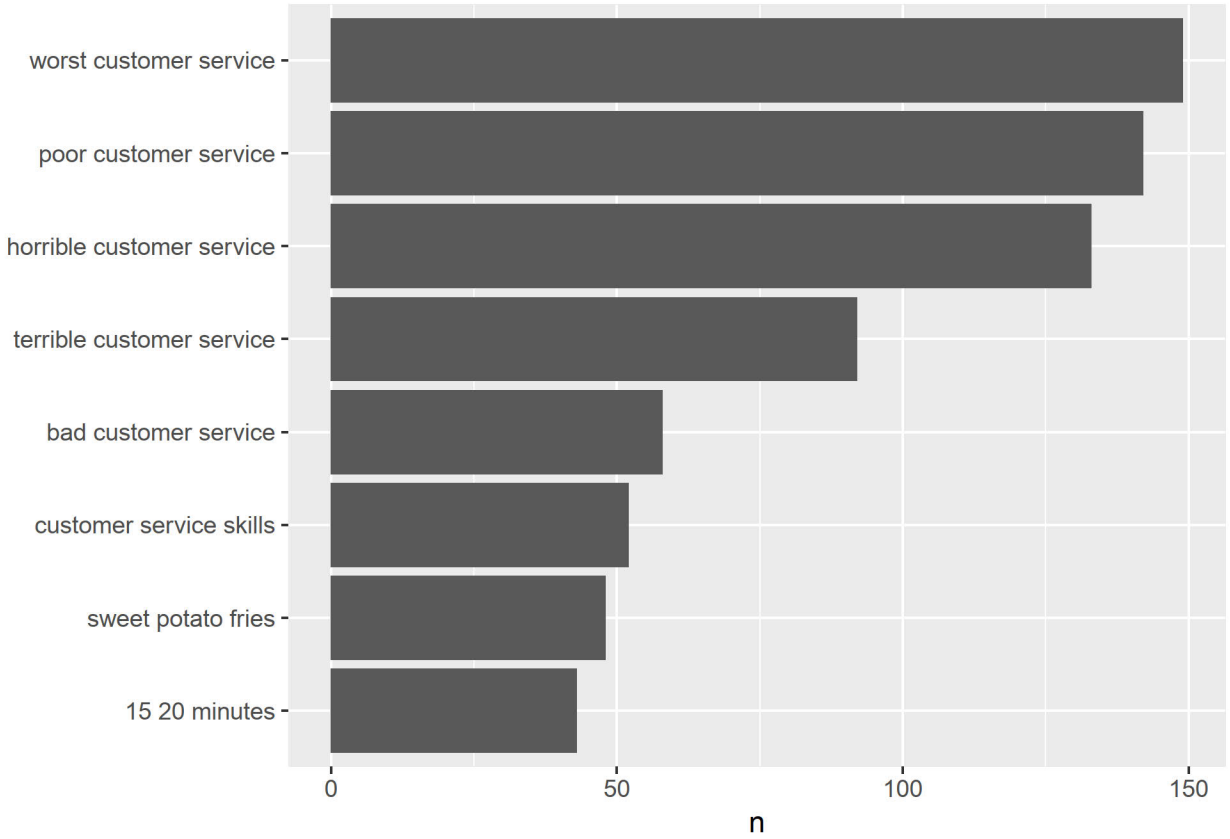
```
united.top.review %>%
  filter(n > 300) %>%
  mutate(bigram = reorder(bigram, n)) %>%
  ggplot(aes(bigram, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip()
```




```
users.review.low<-tibble(useful = review.low$useful,text = review.low$text)
users.review.low$text<-as.character(users.review.low$text)
low.samp<-users.review.low[sample(nrow(users.review.low), 20000, replace=FALSE),]
```

```
united.low.review <-low.samp%>%
  unnest_tokens(trigram, text, token = "ngrams", n = 3) %>%
  separate(trigram, c("word1", "word2", "word3"), sep = " ") %>%
  filter(!word1 %in% stop_words$word,
         !word2 %in% stop_words$word,
         !word3 %in% stop_words$word) %>%
  count(word1, word2, word3, sort = TRUE) %>%
  unite(trigram, word1, word2,word3, sep = " ")
```

```
united.low.review %>%
  filter(n > 40) %>%
  mutate(trigram = reorder(trigram, n)) %>%
  ggplot(aes(trigram, n)) +
  geom_col() +
  xlab(NULL) +
  coord_flip()
```



How to write a useful review

```
phrase.1<-rep(0,nrow(yelp.review))
phrase.2<-rep(0,nrow(yelp.review))
phrase.3<-rep(0,nrow(yelp.review))
phrase.4<-rep(0,nrow(yelp.review))
phrase.1<-str_count(yelp.review$text,"highly recommend")
phrase.2<-str_count(yelp.review$text,"customer service")
phrase.3<-str_count(yelp.review$text,"hours")
phrase.4<-str_count(yelp.review$text,"minutes")
yelp.review$num_phrase<-phrase.1+phrase.2+phrase.3+phrase.4
```

```
yelp.review%>%count(state, sort = T)
```

```
## # A tibble: 36 x 2
##    state       n
##    <chr>   <int>
##  1 NV    2320491
##  2 AZ    2082951
##  3 ON     784461
##  4 NC     408060
##  5 OH     321345
##  6 PA     290097
##  7 QC     179039
##  8 WI     133660
```

```
##  9 AB      99639
## 10 IL      42371
## # ... with 26 more rows
```

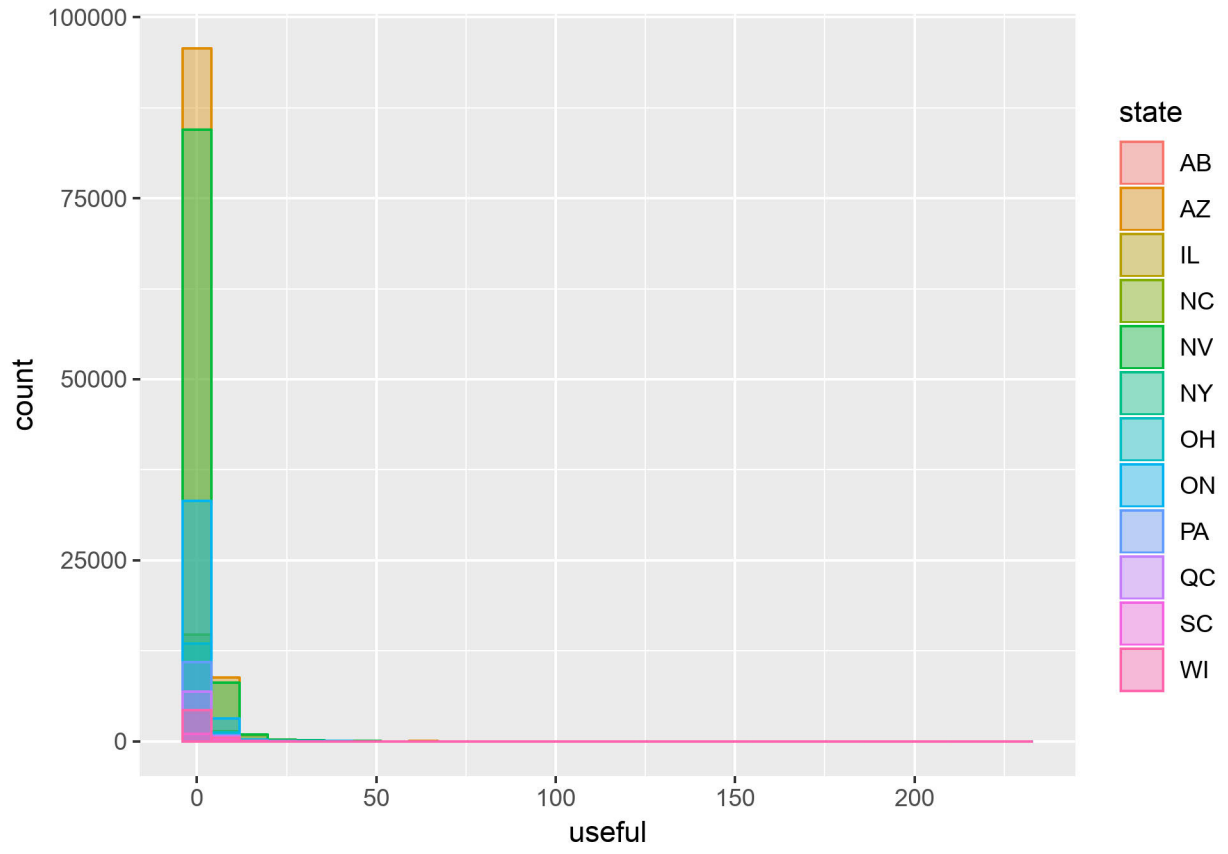
```
yelp.review<-yelp.review %>%
  filter(!grepl("NJ", state))
yelp.review<-yelp.review %>%
  filter(!grepl("VT", state))
yelp.review<-yelp.review %>%
  filter(!grepl("XWY", state))
yelp.review<-yelp.review %>%
  filter(!grepl("AK", state))
yelp.review<-yelp.review %>%
  filter(!grepl("AR", state))
yelp.review<-yelp.review %>%
  filter(!grepl("BAS", state))
yelp.review<-yelp.review %>%
  filter(!grepl("DOW", state))
yelp.review<-yelp.review %>%
  filter(!grepl("UT", state))
yelp.review<-yelp.review %>%
  filter(!grepl("BC", state))
yelp.review<-yelp.review %>%
  filter(!grepl("CON", state))
yelp.review<-yelp.review %>%
  filter(!grepl("DUR", state))
yelp.review<-yelp.review %>%
  filter(!grepl("TN", state))
yelp.review<-yelp.review %>%
  filter(!grepl("XGL", state))

yelp.review<-yelp.review %>%
  filter(!grepl("WA", state))
yelp.review<-yelp.review %>%
  filter(!grepl("CT", state))
yelp.review<-yelp.review %>%
  filter(!grepl("VA", state))
yelp.review<-yelp.review %>%
  filter(!grepl("GA", state))
yelp.review<-yelp.review %>%
  filter(!grepl("NM", state))
yelp.review<-yelp.review %>%
  filter(!grepl("XGM", state))
yelp.review<-yelp.review %>%
  filter(!grepl("AL", state))
yelp.review<-yelp.review %>%
  filter(!grepl("NE", state))
```

```
train.re<-yelp.review[1:300000,]
test.re<-yelp.review[300000:600000,]
```

```
ggplot(train.re, aes(x = useful)) +
  geom_histogram(aes(color = state, fill = state),
```

```
                position = "identity", bins = 30, alpha = 0.4)
```




```
review.mod<-lmer (useful ~  funny + cool + num_phrase + (1 | state), data = train.re)
summary(review.mod)
```

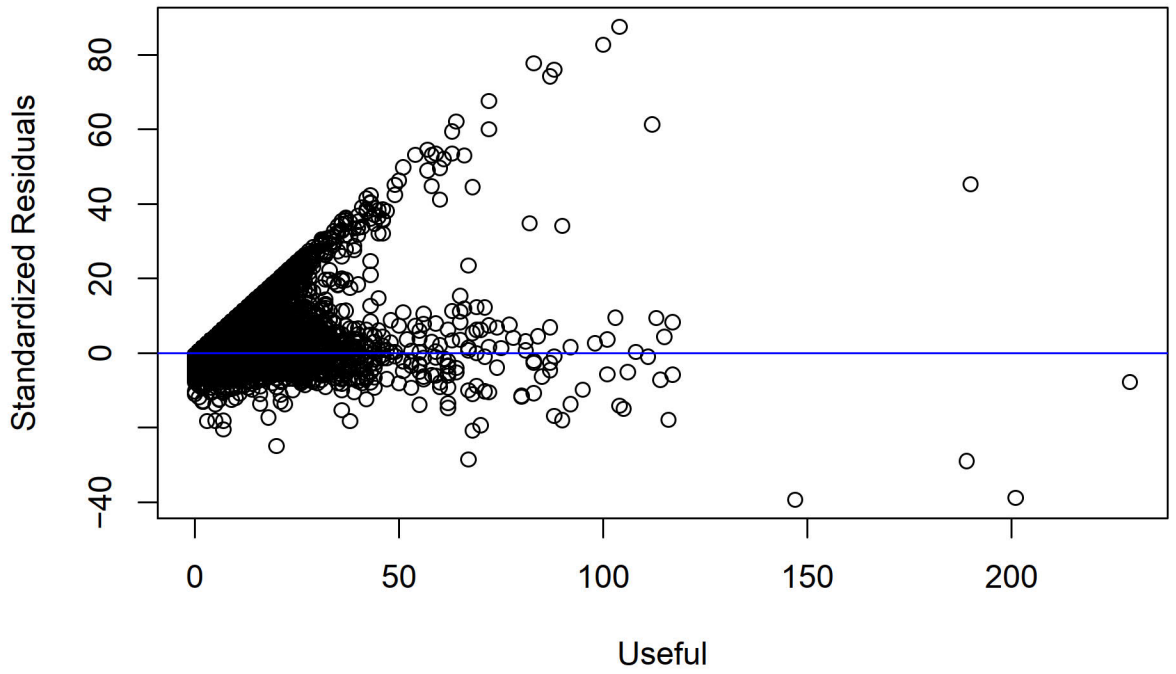
```
## Linear mixed model fit by REML ['lmerMod']
## Formula: useful ~ funny + cool + num_phrase + (1 | state)
##    Data: train.re
##
## REML criterion at convergence: 1223536
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -21.134  -0.339  -0.287   0.206  47.084
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  state    (Intercept) 0.3746   0.612
##  Residual             3.4564   1.859
## Number of obs: 300000, groups:  state, 12
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) 0.813865   0.177433   4.587
## funny       0.451832   0.003288 137.412
```

```
## cool         0.797017   0.002629 303.123
## num_phrase   0.291532   0.005927  49.184
##
## Correlation of Fixed Effects:
##             (Intr) funny  cool
## funny       -0.001
## cool        -0.001 -0.831
## num_phrase -0.007 -0.042  0.030
```

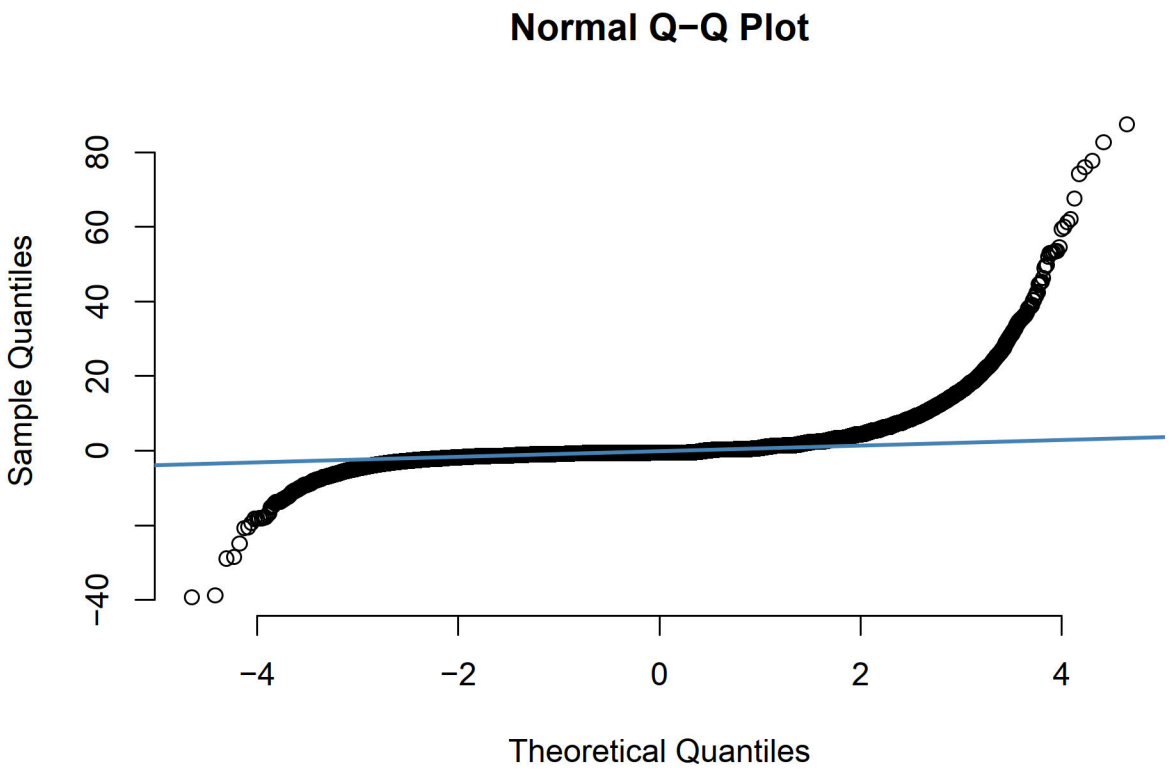
```
AIC(review.mod)
```

```
## [1] 1223548
```

```
train.re$resid.re<-resid(review.mod)
plot(train.re$useful,train.re$resid.re ,ylab="Standardized Residuals", xlab="Useful" )
abline(0,0,col="blue")
```




```
qqnorm(train.re$resid.re, pch = 1, frame = FALSE)
qqline(train.re$resid.re, col = "steelblue", lwd = 2)
```




```
coef(review.mod)
```

```
## $state
##    (Intercept)     funny      cool num_phrase
## AB   0.7240770 0.4518315 0.7970167  0.2915318
## AZ   0.6164706 0.4518315 0.7970167  0.2915318
## IL   0.6398337 0.4518315 0.7970167  0.2915318
## NC   0.6600065 0.4518315 0.7970167  0.2915318
## NV   0.5326716 0.4518315 0.7970167  0.2915318
## NY   2.7139370 0.4518315 0.7970167  0.2915318
## OH   0.5705275 0.4518315 0.7970167  0.2915318
## ON   0.6297510 0.4518315 0.7970167  0.2915318
## PA   0.5156289 0.4518315 0.7970167  0.2915318
## QC   0.5228068 0.4518315 0.7970167  0.2915318
## SC   0.9394186 0.4518315 0.7970167  0.2915318
## WI   0.7012540 0.4518315 0.7970167  0.2915318
##
## attr(,"class")
## [1] "coef.mer"
```

```
predictions <- review.mod %>% predict(test.re)
RMSE(predictions, test.re$useful)
```

```
## [1] 2.143809
```

```r
MAE(predictions, test.re$useful)
```

```
## [1] 1.047956
```