DATA 606 Data Project Proposal

Michael Y.

Contents

Brief Intro

The “Oxford Comma” is the subject of endless debate among grammarians. It’s the comma that precedes the final conjunction in a listing of 3 or more items. For example, the sentence

“For breakfast I like to eat cereal, toast, and juice”

contains the Oxford Comma, while

“I am planning to invite Tim, Dick and Harry”

does not.

In cases like the above, there is really no difference in interpretation, but there are notable examples where the lack of an Oxford Comma can produce humorous results, for example:

Some additional humorous examples include:

On a more serious note, ambiguities in contracts because of the presence or absence of the Oxford Comma have resulted in costly legal disputes, such as

https://www.nytimes.com/2018/02/09/us/oxford-comma-maine.html

Data Preparation

```r
# Load libraries
library(tidyr)
library(dplyr)
library(kableExtra)
library(ggplot2)
library(psych)
library(forcats)   ## for releveling of factors
```

```r
# load data from fivethirtyeight
commadata <- read.csv("https://raw.githubusercontent.com/fivethirtyeight/data/master/comma-survey/comma-
summary(commadata)
```

```
##   RespondentID
##  Min.   :3.288e+09
##  1st Qu.:3.289e+09
##  Median :3.290e+09
##  Mean   :3.290e+09
##  3rd Qu.:3.291e+09
##  Max.   :3.293e+09
##
##                              In.your.opinion..which.sentence.is.more.gramatically.correct.
##  It's important for a person to be honest, kind and loyal. :488
##  It's important for a person to be honest, kind, and loyal.:641
##
```

```
##
##
##
##
## Prior.to.reading.about.it.above..had.you.heard.of.the.serial..or.Oxford..comma.
## No  :444
## Yes :655
## NA's: 30
##
##
##
##
## How.much..if.at.all..do.you.care.about.the.use..or.lack.thereof..of.the.serial..or.Oxford..comma.in
## A lot     :291
## Not at all:126
## Not much  :268
## Some      :414
## NA's      : 30
##
##
##                                                        How.would.you.write.the.following.sentence.
## Some experts say it's important to drink milk, but the data are inconclusive.:228
## Some experts say it's important to drink milk, but the data is inconclusive. :865
## NA's                                                                         : 36
##
##
##
##
## When.faced.with.using.the.word..data...have.you.ever.spent.time.considering.if.the.word.was.a.singul
## No  :547
## Yes :544
## NA's: 38
##
##
##
##
## How.much..if.at.all..do.you.care.about.the.debate.over.the.use.of.the.word..data..as.a.singluar.or.
## A lot     :133
## Not at all:203
## Not much  :403
## Some      :352
## NA's      : 38
##
##
##        In.your.opinion..how.important.or.unimportant.is.proper.use.of.grammar.
## Neither important nor unimportant (neutral): 26
## Somewhat important                         :333
## Somewhat unimportant                       :  7
## Very important                             :688
## Very unimportant                           :  5
## NA's                                       : 70
##
##    Gender       Age                 Household.Income
## Female:548   > 60 :272   $0 - $24,999       :121
##  Male  :489   18-29:221   $100,000 - $149,999:164
##  NA's  : 92   30-44:254   $150,000+          :103
##                45-60:290   $25,000 - $49,999  :158
##                NA's : 92   $50,000 - $99,999  :290
##                            NA's               :293
##
##                               Education         Location..Census.Region.
##  Bachelor degree                   :344   Pacific           :180
##  Graduate degree                   :276   East North Central:170
##  High school degree                :100   South Atlantic    :164
##  Less than high school degree      : 11   Middle Atlantic   :140
##  Some college or Associate degree:295   West South Central: 88
##  NA's                              :103   (Other)           :285
##                                          NA's              :102
```

Clean up the names

```
initial_comma_headers <- names(commadata)
print(initial_comma_headers)
```

```
##  [1] "RespondentID"
##  [2] "In.your.opinion..which.sentence.is.more.gramatically.correct."
##  [3] "Prior.to.reading.about.it.above..had.you.heard.of.the.serial..or.Oxford..comma."
##  [4] "How.much..if.at.all..do.you.care.about.the.use..or.lack.thereof..of.the.serial..or.Oxford..comm
##  [5] "How.would.you.write.the.following.sentence."
##  [6] "When.faced.with.using.the.word..data...have.you.ever.spent.time.considering.if.the.word.was.a.s
##  [7] "How.much..if.at.all..do.you.care.about.the.debate.over.the.use.of.the.word..data..as.a.singluar
##  [8] "In.your.opinion..how.important.or.unimportant.is.proper.use.of.grammar."
##  [9] "Gender"
## [10] "Age"
## [11] "Household.Income"
## [12] "Education"
## [13] "Location..Census.Region."
```

```
###### Clean up the names
newnames <- c("RespondentID","USES_Oxford","HEARD_Oxford","CARE_Oxford","DATA_Sentence","DATA_Plural","
names(commadata) <- newnames

print("Newnames:")
```

```
## [1] "Newnames:"
```

```
print(t(t(names(commadata))))
```

```
##       [,1]
##  [1,] "RespondentID"
##  [2,] "USES_Oxford"
##  [3,] "HEARD_Oxford"
##  [4,] "CARE_Oxford"
##  [5,] "DATA_Sentence"
##  [6,] "DATA_Plural"
##  [7,] "DATA_Care"
##  [8,] "Grammar_Important"
##  [9,] "Gender"
## [10,] "Age"
## [11,] "Income"
## [12,] "Education"
## [13,] "Location"
```

[1] `RespondentID` should not impact the results – it is just an identifier, so drop it

```r
# dimension before dropping
print(dim(commadata))
```

```
## [1] 1129   13
```

```r
# drop the column
commadata$RespondentID <- NULL
# dimension after dropping
print(dim(commadata))
```

```
## [1] 1129   12
```

[2] Restate Oxford Comma usage response to `True` or `False`

```r
## [2] The question posed to participants was whether they preferred a sentence using the oxford comma
t(t(table(commadata$USES_Oxford)))
```

```
##
##                                                                  [,1]
##   It's important for a person to be honest, kind and loyal.   488
##   It's important for a person to be honest, kind, and loyal.  641
```

```r
#  It's important for a person to be honest, kind and loyal.   488
#  It's important for a person to be honest, kind, and loyal.  641
#### Replace the response with False / True:
levels(commadata$USES_Oxford) <- c(F,T)
print("Does the respondent prefer to use the Oxford Comma?")
```

```
## [1] "Does the respondent prefer to use the Oxford Comma?"
```

```r
t(t(summary(commadata$USES_Oxford)))
```

```
##       [,1]
## FALSE  488
## TRUE   641
```

```r
#FALSE  TRUE
#  488   641
```

[4] Resequence the levels for the responses to reflect how much does the participant *care about* the Oxford Comma?

```r
# First ten observations:
t(t(head(commadata$CARE_Oxford,10)))
```

```
##       [,1]
##  [1,] Some
##  [2,] Not much
##  [3,] Some
##  [4,] Some
##  [5,] Not much
##  [6,] A lot
##  [7,] A lot
##  [8,] A lot
##  [9,] A lot
## [10,] Not at all
## Levels: A lot Not at all Not much Some
```

```
# Summary table (sequence is not ordinal):
t(t(table(commadata$CARE_Oxford,useNA = "always")))
```

```
## 
##               [,1]
##   A lot       291
##   Not at all  126
##   Not much    268
##   Some        414
##   <NA>         30
```

```
# What are the original levels on the CARE_Oxford variable?
t(t(levels(commadata$CARE_Oxford)))
```

```
##      [,1]
## [1,] "A lot"
## [2,] "Not at all"
## [3,] "Not much"
## [4,] "Some"
```

```
### use fct_relevel from library `forcats` to sort the CARE_Oxford levels ordinally
commadata$CARE_Oxford = fct_relevel(commadata$CARE_Oxford, levels(commadata$CARE_Oxford)[c(2,3,4,1)])

t(t(levels(commadata$CARE_Oxford)))
```

```
##      [,1]
## [1,] "Not at all"
## [2,] "Not much"
## [3,] "Some"
## [4,] "A lot"
```

```
t(t(summary(commadata$CARE_Oxford)))      # resequenced to reflect ordering from  "Not at all" to "A lo
```

```
##             [,1]
## Not at all  126
## Not much    268
## Some        414
## A lot       291
## NA's         30
```

```
### Make sure the results are still same

print("CARE_oxford levels:")
```

```
## [1] "CARE_oxford levels:"
```

```
t(t(table(commadata$CARE_Oxford,useNA = "always")))
```

```
## 
##               [,1]
##   Not at all  126
##   Not much    268
##   Some        414
##   A lot       291
##   <NA>         30
```

```
t(t(head(commadata$CARE_Oxford,10)))
```

```
##       [,1]
##  [1,] Some
##  [2,] Not much
##  [3,] Some
##  [4,] Some
##  [5,] Not much
##  [6,] A lot
##  [7,] A lot
##  [8,] A lot
##  [9,] A lot
## [10,] Not at all
## Levels: Not at all Not much Some A lot
```

[5] Grammatical questions on usage of word "Data"

```
### [5] Another question asked users whether they think the word "data" should be considered singular o

t(t(table(commadata$DATA_Sentence,useNA = "always")))
```

```
##
##                                                                                  [,1]
##   Some experts say it's important to drink milk, but the data are inconclusive.  228
##   Some experts say it's important to drink milk, but the data is inconclusive.   865
##   <NA>                                                                            36
```

```
#  Some experts say it's important to drink milk, but the data are inconclusive.  228
#  Some experts say it's important to drink milk, but the data is inconclusive.   865

### Replace the above sentences with the word "PLURAL" or "SINGULAR" to reflect user preference
levels(commadata$DATA_Sentence) <- c("PLURAL","SINGULAR")
t(t(summary(commadata$DATA_Sentence)))
```

```
##          [,1]
## PLURAL    228
## SINGULAR  865
## NA's       36
```

```
# PLURAL    228
# SINGULAR  865
# NA's       36
```

[7] Resequence the levels for the responses to reflect how much does the participant care about whether "Data" is considered Singular or Plural

```
# First ten observations:
t(t(head(commadata$DATA_Care,10)))
```

```
##       [,1]
##  [1,] Not much
##  [2,] Not much
##  [3,] Not at all
##  [4,] Some
##  [5,] Not much
##  [6,] Some
##  [7,] Some
##  [8,] A lot
##  [9,] Not much
## [10,] Some
## Levels: A lot Not at all Not much Some
```

```
# Summary table (sequence is not ordinal):
t(t(table(commadata$DATA_Care,useNA = "always")))
```

```
##
##              [,1]
##   A lot       133
##   Not at all  203
##   Not much    403
##   Some        352
##   <NA>         38
```

```
# What are the original levels on the DATA_Care variable?
t(t(levels(commadata$DATA_Care)))
```

```
##      [,1]
## [1,] "A lot"
## [2,] "Not at all"
## [3,] "Not much"
## [4,] "Some"
```

```
### use fct_relevel from library `forcats` to sort the DATA_Care levels ordinally
commadata$DATA_Care = fct_relevel(commadata$DATA_Care, levels(commadata$DATA_Care)[c(2,3,4,1)])

t(t(levels(commadata$DATA_Care)))
```

```
##      [,1]
## [1,] "Not at all"
## [2,] "Not much"
## [3,] "Some"
## [4,] "A lot"
```

```
t(t(summary(commadata$DATA_Care)))      # resequenced to reflect ordering from  "Not at all" to "A lot"
```

```
##            [,1]
## Not at all  203
## Not much    403
## Some        352
## A lot       133
## NA's         38
```

```
### Make sure the results are still same

print("DATA_Care levels:")
```

```
## [1] "DATA_Care levels:"
```

```
t(t(table(commadata$DATA_Care,useNA = "always")))
```

```
##
##              [,1]
##   Not at all  203
##   Not much    403
##   Some        352
##   A lot       133
##   <NA>         38
```

```
t(t(head(commadata$DATA_Care,10)))
```

```
##       [,1]
##   [1,] Not much
##   [2,] Not much
##   [3,] Not at all
##   [4,] Some
##   [5,] Not much
##   [6,] Some
##   [7,] Some
##   [8,] A lot
##   [9,] Not much
##  [10,] Some
## Levels: Not at all Not much Some A lot
```

[8] Resequence the levels for the the responses to *Importance of Proper Use of Grammar*?

```
# Incoming table of Grammar_Important responses:
t(t(table(commadata$Grammar_Important,useNA = "always")))
```

```
##
##                                                 [,1]
##   Neither important nor unimportant (neutral)   26
##   Somewhat important                            333
##   Somewhat unimportant                            7
##   Very important                                688
##   Very unimportant                                5
##   <NA>                                           70
```

```
# First 10 responses:
t(t(head(commadata$Grammar_Important,10)))
```

```
##       [,1]
##   [1,] Somewhat important
##   [2,] Somewhat unimportant
##   [3,] Very important
##   [4,] Somewhat important
##   [5,] <NA>
##   [6,] Very important
##   [7,] Very important
##   [8,] Very important
##   [9,] Very important
##  [10,] Very important
## 5 Levels: Neither important nor unimportant (neutral) ...
```

```
# What are the original levels on the Grammar_Important variable?
t(t(levels(commadata$Grammar_Important)))
```

```
##      [,1]
## [1,] "Neither important nor unimportant (neutral)"
## [2,] "Somewhat important"
## [3,] "Somewhat unimportant"
## [4,] "Very important"
## [5,] "Very unimportant"
```

```
### use fct_relevel from library `forcats` to sort the Grammar_Important levels ordinally
commadata$Grammar_Important = fct_relevel(commadata$Grammar_Important, levels(commadata$Grammar_Importan

# Resequenced levels:
t(t(levels(commadata$Grammar_Important)))
```

```
##      [,1]
## [1,] "Very unimportant"
## [2,] "Somewhat unimportant"
## [3,] "Neither important nor unimportant (neutral)"
## [4,] "Somewhat important"
## [5,] "Very important"
```

```
# Summary table:
t(t(summary(commadata$Grammar_Important)))  # resequenced to reflect ordering from "very unimportant" t
```

```
##                                              [,1]
## Very unimportant                                5
## Somewhat unimportant                            7
## Neither important nor unimportant (neutral)    26
## Somewhat important                            333
## Very important                                688
## NA's                                           70
```

```
### Make sure the results are still same

# Grammar_Important levels:
t(t(table(commadata$Grammar_Important,useNA = "always")))
```

```
##
##                                                [,1]
##   Very unimportant                                5
##   Somewhat unimportant                            7
##   Neither important nor unimportant (neutral)    26
##   Somewhat important                            333
##   Very important                                688
##   <NA>                                           70
```

```
# First ten entries:
t(t(head(commadata$Grammar_Important,10)))
```

```
##       [,1]
##  [1,] Somewhat important
##  [2,] Somewhat unimportant
##  [3,] Very important
##  [4,] Somewhat important
##  [5,] <NA>
##  [6,] Very important
##  [7,] Very important
##  [8,] Very important
##  [9,] Very important
## [10,] Very important
## 5 Levels: Very unimportant ... Very important
```

[10] Fix the `Age` variable - resequence the levels, and create a numeric equivalent, based upon the midpoints

```
# First ten entries from the Age variable:
t(t(head(commadata$Age,10)))
```

```
##       [,1]
##  [1,] 30-44
##  [2,] 30-44
##  [3,] 30-44
##  [4,] 18-29
##  [5,] <NA>
##  [6,] 18-29
##  [7,] 18-29
##  [8,] 18-29
##  [9,] 30-44
## [10,] 30-44
## Levels: > 60 18-29 30-44 45-60
```

```
# Save the Age bands
commadata$AgeBands <- commadata$Age

# What are the original levels on the Age variable?
t(t(levels(commadata$Age)))
```

```
##      [,1]
## [1,] "> 60"
## [2,] "18-29"
## [3,] "30-44"
## [4,] "45-60"
```

```
### use fct_relevel from library `forcats` to sort the age levels ordinally
commadata$AgeBands = fct_relevel(commadata$AgeBands, levels(commadata$Age)[c(2,3,4,1)])

# What are the resequenced levels on the Age variable?
t(t(levels(commadata$AgeBands)))
```

```
##      [,1]
## [1,] "18-29"
## [2,] "30-44"
## [3,] "45-60"
## [4,] "> 60"
```

```
### Make sure the results are still same

# Age bands:
t(t(table(commadata$AgeBands,useNA = "always")))
```

```
##
##         [,1]
##   18-29  221
##   30-44  254
##   45-60  290
##   > 60   272
##   <NA>    92
```

```r
# First ten entries:
t(t(head(commadata$AgeBands,10)))
```

```
##       [,1]
##  [1,] 30-44
##  [2,] 30-44
##  [3,] 30-44
##  [4,] 18-29
##  [5,] <NA>
##  [6,] 18-29
##  [7,] 18-29
##  [8,] 18-29
##  [9,] 30-44
## [10,] 30-44
## Levels: 18-29 30-44 45-60 > 60
```

```r
#### Let's replace the above Age ranges with their midpoints, so we can treat Age as a numeric variable

commadata$AgeNumeric <- commadata$AgeBands
levels(commadata$AgeNumeric) <- c(23.5,37,52.5,65)   ## 18-29 ; 30-44 ; 45-60 ; Over 60

print("Age table (B):")
```

```
## [1] "Age table (B):"
```

```r
t(t(table(commadata$AgeNumeric,useNA = "always")))
```

```
##
##        [,1]
##   23.5  221
##   37    254
##   52.5  290
##   65    272
##   <NA>   92
```

```r
str(commadata$AgeNumeric)
```

```
##  Factor w/ 4 levels "23.5","37","52.5",..: 2 2 2 1 NA 1 1 1 2 2 ...
```

```r
# First ten entries:
t(t(head(commadata$AgeNumeric,10)))
```

```
##       [,1]
##  [1,] 37
##  [2,] 37
##  [3,] 37
##  [4,] 23.5
##  [5,] <NA>
##  [6,] 23.5
##  [7,] 23.5
##  [8,] 23.5
##  [9,] 37
## [10,] 37
## Levels: 23.5 37 52.5 65
```

```r
# Replace the above string values with their numeric equivalents
commadata$AgeNumeric <- as.numeric(levels(commadata$AgeNumeric))[commadata$AgeNumeric]
```

```
print("Age table (C):")
```

```
## [1] "Age table (C):"
```

```
t(t(table(commadata$AgeNumeric,useNA = "always")))
```

```
##
##         [,1]
##   23.5  221
##   37    254
##   52.5  290
##   65    272
##   <NA>   92
```

```
str(commadata$AgeNumeric)
```

```
##  num [1:1129] 37 37 37 23.5 NA 23.5 23.5 23.5 37 37 ...
```

```
# First ten entries:
t(t(head(commadata$AgeNumeric,10)))
```

```
##       [,1]
##  [1,] 37.0
##  [2,] 37.0
##  [3,] 37.0
##  [4,] 23.5
##  [5,]   NA
##  [6,] 23.5
##  [7,] 23.5
##  [8,] 23.5
##  [9,] 37.0
## [10,] 37.0
```

[11] Fix the `Income` variable - resequence the levels, and create a numeric equivalent, based upon the midpoints

```
t(t(head(commadata$Income,10)))
```

```
##       [,1]
##  [1,] $50,000 - $99,999
##  [2,] $50,000 - $99,999
##  [3,] <NA>
##  [4,] <NA>
##  [5,] <NA>
##  [6,] $25,000 - $49,999
##  [7,] $0 - $24,999
##  [8,] $25,000 - $49,999
##  [9,] $50,000 - $99,999
## [10,] $150,000+
## 5 Levels: $0 - $24,999 $100,000 - $149,999 ... $50,000 - $99,999
```

```
# Save the income bands (for later)
commadata$IncomeBands <- commadata$Income

# What are the original levels on the income variable?
t(t(levels(commadata$Income)))
```

```
##      [,1]
## [1,] "$0 - $24,999"
## [2,] "$100,000 - $149,999"
## [3,] "$150,000+"
## [4,] "$25,000 - $49,999"
## [5,] "$50,000 - $99,999"
```

```
### use fct_relevel from library forcats to sort the income levels ordinally
commadata$IncomeBands = fct_relevel(commadata$IncomeBands, levels(commadata$IncomeBands)[c(1,4,5,2,3)])

t(t(levels(commadata$IncomeBands)))
```

```
##      [,1]
## [1,] "$0 - $24,999"
## [2,] "$25,000 - $49,999"
## [3,] "$50,000 - $99,999"
## [4,] "$100,000 - $149,999"
## [5,] "$150,000+"
```

```
### Make sure the results are still same

print("Income bands:")
```

```
## [1] "Income bands:"
```

```
t(t(table(commadata$IncomeBands,useNA = "always")))
```

```
##
##                        [,1]
##   $0 - $24,999          121
##   $25,000 - $49,999     158
##   $50,000 - $99,999     290
##   $100,000 - $149,999   164
##   $150,000+             103
##   <NA>                  293
```

```
t(t(head(commadata$IncomeBands,10)))
```

```
##       [,1]
##  [1,] $50,000 - $99,999
##  [2,] $50,000 - $99,999
##  [3,] <NA>
##  [4,] <NA>
##  [5,] <NA>
##  [6,] $25,000 - $49,999
##  [7,] $0 - $24,999
##  [8,] $25,000 - $49,999
##  [9,] $50,000 - $99,999
## [10,] $150,000+
## 5 Levels: $0 - $24,999 $25,000 - $49,999 ... $150,000+
```

```
#### Let's replace the above income ranges with their midpoints, so we can treat income as a numeric va

commadata$IncomeNumeric <- commadata$IncomeBands
levels(commadata$IncomeNumeric) <- c(12500,37500,75000,125000,160000)

print("Income table (B):")
```

```
## [1] "Income table (B):"
```

```
t(t(table(commadata$IncomeNumeric,useNA = "always")))
```

```
## 
##          [,1]
##   12500   121
##   37500   158
##   75000   290
##   125000  164
##   160000  103
##   <NA>    293
```

```
str(commadata$IncomeNumeric)
```

```
##  Factor w/ 5 levels "12500","37500",..: 3 3 NA NA NA 2 1 2 3 5 ...
```

```
t(t(head(commadata$IncomeNumeric,10)))
```

```
##       [,1]  
##  [1,] 75000 
##  [2,] 75000 
##  [3,] <NA>  
##  [4,] <NA>  
##  [5,] <NA>  
##  [6,] 37500 
##  [7,] 12500 
##  [8,] 37500 
##  [9,] 75000 
## [10,] 160000
## Levels: 12500 37500 75000 125000 160000
```

```
# Replace the values with their numeric equivalents
commadata$IncomeNumeric <- as.numeric(levels(commadata$IncomeNumeric))[commadata$IncomeNumeric]
print("Income table (C):")
```

```
## [1] "Income table (C):"
```

```
t(t(table(commadata$IncomeNumeric,useNA = "always")))
```

```
## 
##          [,1]
##   12500   121
##   37500   158
##   75000   290
##   125000  164
##   160000  103
##   <NA>    293
```

```
str(commadata$IncomeNumeric)
```

```
##  num [1:1129] 75000 75000 NA NA NA 37500 12500 37500 75000 160000 ...
```

```
t(t(head(commadata$IncomeNumeric,10)))
```

```
##         [,1]
##  [1,]  75000
##  [2,]  75000
##  [3,]     NA
##  [4,]     NA
##  [5,]     NA
##  [6,]  37500
##  [7,]  12500
##  [8,]  37500
##  [9,]  75000
## [10,] 160000
```

[12] Resequence the levels for the *Education* variable

```r
print("Incoming table of Education responses:")
```

```
## [1] "Incoming table of Education responses:"
```

```r
t(t(table(commadata$Education,useNA = "always")))
```

```
##                                       
##                                       [,1]
##   Bachelor degree                      344
##   Graduate degree                      276
##   High school degree                   100
##   Less than high school degree          11
##   Some college or Associate degree     295
##   <NA>                                 103
```

```r
print("First 10 responses:")
```

```
## [1] "First 10 responses:"
```

```r
t(t(head(commadata$Education,10)))
```

```
##       [,1]
##  [1,] Bachelor degree
##  [2,] Graduate degree
##  [3,] <NA>
##  [4,] Less than high school degree
##  [5,] <NA>
##  [6,] Some college or Associate degree
##  [7,] Some college or Associate degree
##  [8,] Some college or Associate degree
##  [9,] Graduate degree
## [10,] Bachelor degree
## 5 Levels: Bachelor degree Graduate degree ... Some college or Associate degree
```

```r
# What are the original levels on the Education variable?
t(t(levels(commadata$Education)))
```

```
##      [,1]
## [1,] "Bachelor degree"
## [2,] "Graduate degree"
## [3,] "High school degree"
## [4,] "Less than high school degree"
## [5,] "Some college or Associate degree"
```

```r
#[1,] "Bachelor degree"
#[2,] "Graduate degree"
#[3,] "High school degree"
#[4,] "Less than high school degree"
```

```
#[5,] "Some college or Associate degree"


### use fct_relevel from library forcats to sort the Education levels ordinally
commadata$Education = fct_relevel(commadata$Education, levels(commadata$Education)[c(4,3,5,1,2)])

t(t(levels(commadata$Education)))
```

```
##      [,1]
## [1,] "Less than high school degree"
## [2,] "High school degree"
## [3,] "Some college or Associate degree"
## [4,] "Bachelor degree"
## [5,] "Graduate degree"
```

```
t(t(summary(commadata$Education)))            # resequenced to reflect ordering from "Less than high
```

```
##                                   [,1]
## Less than high school degree       11
## High school degree                100
## Some college or Associate degree  295
## Bachelor degree                   344
## Graduate degree                   276
## NA's                              103
```

```
### Make sure the results are still same

print("Education levels:")
```

```
## [1] "Education levels:"
```

```
t(t(table(commadata$Education,useNA = "always")))
```

```
##
##                                     [,1]
##   Less than high school degree       11
##   High school degree                100
##   Some college or Associate degree  295
##   Bachelor degree                   344
##   Graduate degree                   276
##   <NA>                              103
```

```
t(t(head(commadata$Education,10)))
```

```
##       [,1]
##  [1,] Bachelor degree
##  [2,] Graduate degree
##  [3,] <NA>
##  [4,] Less than high school degree
##  [5,] <NA>
##  [6,] Some college or Associate degree
##  [7,] Some college or Associate degree
##  [8,] Some college or Associate degree
##  [9,] Graduate degree
## [10,] Bachelor degree
## 5 Levels: Less than high school degree ... Graduate degree
```

[12] Resequence the levels for the *Location* variable to reflect geography (east to west)

```
print("Incoming table of Location responses:")
```

```
## [1] "Incoming table of Location responses:"
```

```
t(t(table(commadata$Location,useNA = "always")))
```

```
##
##                         [,1]
##   East North Central  170
##   East South Central   43
##   Middle Atlantic     140
##   Mountain             87
##   New England          73
##   Pacific             180
##   South Atlantic      164
##   West North Central   82
##   West South Central   88
##   <NA>                102
```

```
print("First 10 responses:")
```

```
## [1] "First 10 responses:"
```

```
t(t(head(commadata$Location,10)))
```

```
##       [,1]
##  [1,] South Atlantic
##  [2,] Mountain
##  [3,] East North Central
##  [4,] Middle Atlantic
##  [5,] <NA>
##  [6,] New England
##  [7,] Pacific
##  [8,] East North Central
##  [9,] Mountain
## [10,] Pacific
## 9 Levels: East North Central East South Central ... West South Central
```

```
# What are the original levels on the Location variable?
t(t(levels(commadata$Location)))
```

```
##       [,1]
##  [1,] "East North Central"
##  [2,] "East South Central"
##  [3,] "Middle Atlantic"
##  [4,] "Mountain"
##  [5,] "New England"
##  [6,] "Pacific"
##  [7,] "South Atlantic"
##  [8,] "West North Central"
##  [9,] "West South Central"
```

```
# [1,] "East North Central"
# [2,] "East South Central"
# [3,] "Middle Atlantic"
# [4,] "Mountain"
# [5,] "New England"
# [6,] "Pacific"
# [7,] "South Atlantic"
# [8,] "West North Central"
# [9,] "West South Central"


### use fct_relevel from library forcats to sort the Location levels so they reflect geography from Eas
commadata$Location = fct_relevel(commadata$Location, levels(commadata$Location)[c(5,3,7,1,2,8,9,4,6)])

t(t(levels(commadata$Location)))
```

```
##       [,1]
##  [1,] "New England"
##  [2,] "Middle Atlantic"
##  [3,] "South Atlantic"
##  [4,] "East North Central"
##  [5,] "East South Central"
##  [6,] "West North Central"
##  [7,] "West South Central"
##  [8,] "Mountain"
##  [9,] "Pacific"
```

```
t(t(summary(commadata$Location)))              # resequenced to reflect ordering from east cost to west
```

```
##                     [,1]
## New England           73
## Middle Atlantic      140
## South Atlantic       164
## East North Central   170
## East South Central    43
## West North Central    82
## West South Central    88
## Mountain              87
## Pacific              180
## NA's                 102
```

```
### Make sure the results are still same

print("Location levels:")
```

```
## [1] "Location levels:"
```

```
t(t(table(commadata$Location,useNA = "always")))
```

```
##
##                       [,1]
##   New England           73
##   Middle Atlantic      140
##   South Atlantic       164
##   East North Central   170
##   East South Central    43
##   West North Central    82
##   West South Central    88
##   Mountain              87
##   Pacific              180
##   <NA>                 102
```

```
t(t(head(commadata$Location,10)))
```

```
##       [,1]
##  [1,] South Atlantic
##  [2,] Mountain
##  [3,] East North Central
##  [4,] Middle Atlantic
##  [5,] <NA>
##  [6,] New England
##  [7,] Pacific
##  [8,] East North Central
##  [9,] Mountain
## [10,] Pacific
## 9 Levels: New England Middle Atlantic ... Pacific
```

Let's drop the "DATA (singular vs. plural)" questions from the analysis:

```
commadata$DATA_Sentence <- NULL
commadata$DATA_Plural <- NULL
commadata$DATA_Care <- NULL

# Dimension after dropping the above variables
dim(commadata)
```

```
## [1] 1129   13
```

Drop cases containing NAs

```
summary(commadata)
```

```
##  USES_Oxford HEARD_Oxford     CARE_Oxford
##  FALSE:488   No  :444     Not at all:126
##  TRUE :641   Yes :655     Not much  :268
##              NA's: 30     Some      :414
##                           A lot     :291
##                           NA's      : 30
##
##
##                                    Grammar_Important    Gender
##  Very unimportant                           :  5     Female:548
##  Somewhat unimportant                       :  7     Male  :489
##  Neither important nor unimportant (neutral): 26     NA's  : 92
##  Somewhat important                         :333
##  Very important                             :688
##  NA's                                       : 70
##
##     Age                      Income
##  > 60 :272   $0 - $24,999       :121
##  18-29:221   $100,000 - $149,999:164
##  30-44:254   $150,000+          :103
##  45-60:290   $25,000 - $49,999  :158
##  NA's : 92   $50,000 - $99,999  :290
##              NA's               :293
```

```
##
##                             Education                 Location
##  Less than high school degree    : 11   Pacific           :180
##  High school degree              :100   East North Central:170
##  Some college or Associate degree:295   South Atlantic    :164
##  Bachelor degree                 :344   Middle Atlantic   :140
##  Graduate degree                 :276   West South Central: 88
##  NA's                            :103   (Other)           :285
##                                         NA's              :102
##   AgeBands     AgeNumeric                IncomeBands  IncomeNumeric
##  18-29:221   Min.   :23.5   $0 - $24,999       :121   Min.   : 12500
##  30-44:254   1st Qu.:37.0   $25,000 - $49,999  :158   1st Qu.: 37500
##  45-60:290   Median :52.5   $50,000 - $99,999  :290   Median : 75000
##  > 60 :272   Mean   :45.8   $100,000 - $149,999:164   Mean   : 79148
##  NA's : 92   3rd Qu.:65.0   $150,000+          :103   3rd Qu.:125000
##              Max.   :65.0   NA's               :293   Max.   :160000
##              NA's   :92                               NA's   :293
```

```
# Size of dataframe BEFORE dropping rows containing NAs:
dim(commadata)
```

```
## [1] 1129   13
```

```
## drop rows which contain any NA values
comma2 <- commadata[complete.cases(commadata),]
summary(comma2)
```

```
##  USES_Oxford HEARD_Oxford     CARE_Oxford
##  FALSE:358   No :329      Not at all: 86
##  TRUE :470   Yes:499      Not much  :208
##                           Some      :311
##                           A lot     :223
##
##
##
##                                   Grammar_Important    Gender
##  Very unimportant                           :  2     Female:437
##  Somewhat unimportant                       :  5     Male  :391
##  Neither important nor unimportant (neutral): 21
##  Somewhat important                         :267
##  Very important                             :533
##
##
##     Age                        Income
##  > 60 :201   $0 - $24,999       :120
##  18-29:174   $100,000 - $149,999:163
##  30-44:205   $150,000+          :102
##  45-60:248   $25,000 - $49,999  :157
##              $50,000 - $99,999  :286
##
##
##                             Education                 Location
##  Less than high school degree    :  8   Pacific           :152
##  High school degree              : 82   South Atlantic    :132
##  Some college or Associate degree:228   East North Central:131
##  Bachelor degree                 :281   Middle Atlantic   :108
##  Graduate degree                 :229   West South Central: 73  
##                                         West North Central: 68  
##                                         (Other)           :164  
##   AgeBands     AgeNumeric                IncomeBands  IncomeNumeric
##  18-29:174   Min.   :23.5   $0 - $24,999       :120   Min.   : 12500  
##  30-44:205   1st Qu.:37.0   $25,000 - $49,999  :157   1st Qu.: 37500  
##  45-60:248   Median :52.5   $50,000 - $99,999  :286   Median : 75000  
##  > 60 :201   Mean   :45.6   $100,000 - $149,999:163   Mean   : 79146  
##              3rd Qu.:52.5   $150,000+          :102   3rd Qu.:125000  
##              Max.   :65.0                             Max.   :160000  
##
```

```
# Size of dataframe "comma2" AFTER dropping rows containing NAs:
dim(comma2)
```

```
## [1] 828  13
```

Research question

You should phrase your research question in a way that matches up with the scope of inference your dataset allows for.

Is there any association between gender, age, income, educational attainment, and geographic region among individuals who choose to use the Oxford Comma? Which variables provide the strongest support for predicting whether an individual will or will not prefer to use the Oxford Comma?

Cases

What are the cases, and how many are there?

The dataset contains 1129 cases, each of which represents a response to an online poll conducted in June 2014, where participants were asked various questions, including:

1) whether they knew what the Oxford Comma is,
2) which of two sentences (one with the serial comma, and one without) they preferred, and
3) whether they believed the use of proper grammar was important.

Additionally, participants were asked questions regarding their gender, age, income, educational attainment, and geographic region.

Data collection

Describe the method of data collection.

Data was collected using an online poll on the SurveyMonkey.com platform. The survey was open for three days (June 3-5, 2014). There were 1129 participants, however not everyone answered all of the questions. A total of 825 respondents answered all questions. Excluding the questions pertaining to whether “Data” is singular or plural, 828 respondents answered all remaining questions.

Type of study

What type of study is this (observational/experiment)?

This is an observational study.

Data Source

If you collected the data, state self-collected. If not, provide a citation/link.

In June of 2014, `FiveThirtyEight.com` ran an online poll using “surveymonkey.com” asking Americans whether they preferred the serial comma (also known as the `Oxford Comma`.)

Additional questions were posed regarding the respondents’ educational level, income level, age, and what part of the country each was from.

Additional grammatical questions which were part of the same poll concerned usage of the word “data”: respondents were asked whether they considered “data” to be *singular* or *plural*.

Information on the study is here: https://fivethirtyeight.com/features/elitist-superfluous-or-popular-we-polled-americans-on-th

The data can be sourced from here: https://github.com/fivethirtyeight/data/tree/master/comma-survey

Dependent Variable

What is the response variable? Is it quantitative or qualitative?

The response variable is `USES_Oxford` - it is a True/False variable which indicates whether a subject prefers the use of the Oxford Comma, or not.

Independent Variable

You should have two independent variables, one quantitative and one qualitative.

I am going to implement stepwise logistic regression on all of the variables to determine which are most indicative as to whether an individual prefers the use of the Oxford Comma. (I will omit the variables pertaining to whether “Data” is singular or plural, as those represent a side question.)

Most significantly, I’ll explore whether there is any association favoring the Oxford Comma based upon participant `Age`, `Income`, `Gender`, `Education`, and geographic `Location` .

The variables `Age` and `Income` are quantitative, but not continuous (as the results are given in bands rather than exact values.)

`Gender` and `Location` are qualitative, non-ordinal variables.

`Education` is a qualitative variable, but it is ordinal as the various levels of education can be ranked.

My conjecture would be that users of the Oxford Comma would tend to be younger, higher in income, and higher in education that non-users.

I do not have any prior expectation that `Gender` would impact the results.

It would be interesting to see whether `Location` has any impact on the results, as regional dialects can impact usage of English in various parts of the country.

Relevant summary statistics

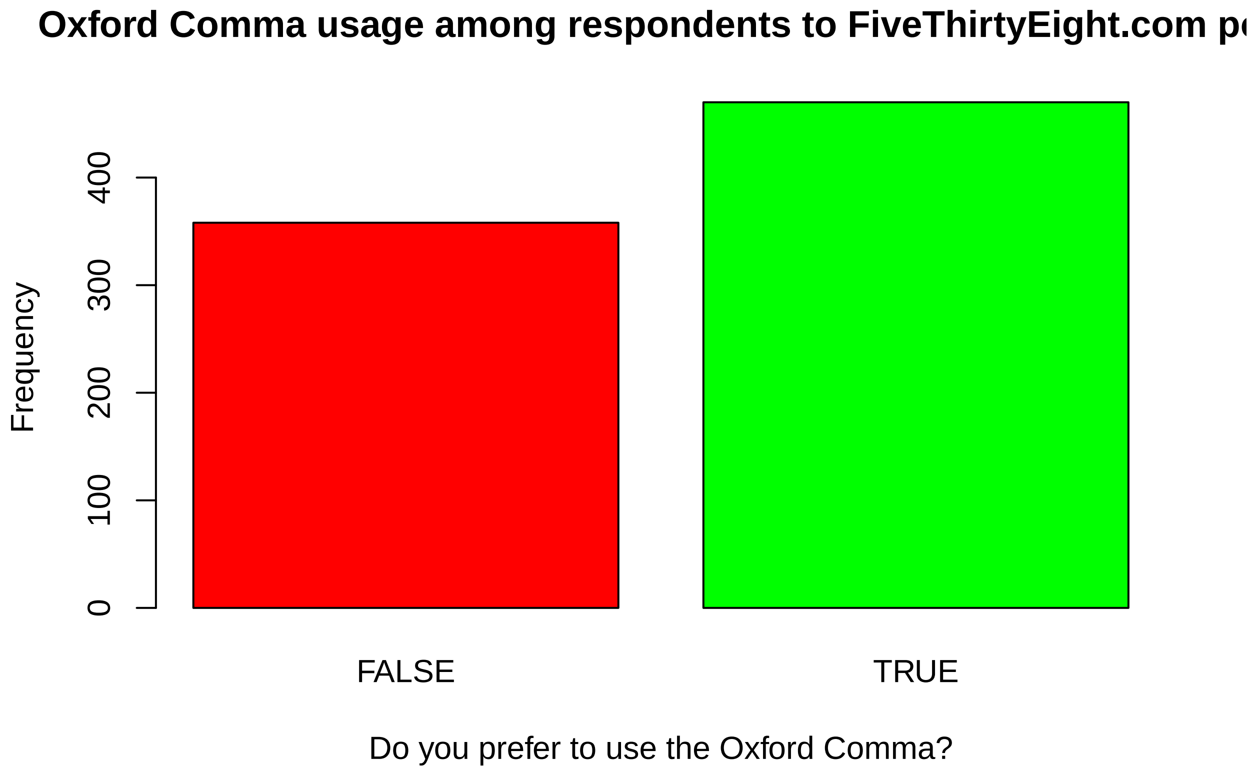
Provide summary statistics for each the variables. Also include appropriate visualizations related to your research question (e.g. scatter plot, boxplots, etc). This step requires the use of R, hence a code chunk is provided below. Insert more code chunks as needed.

[2] **Does the respondent *prefer to use* the oxford comma?**

```
t(t(summary(comma2$USES_Oxford)))
```

```
##       [,1]
## FALSE  358
## TRUE   470
```

```
barplot(table(comma2$USES_Oxford),col=c("red","green"),
        xlab = "Do you prefer to use the Oxford Comma?", ylab = "Frequency",
        main = "Oxford Comma usage among respondents to FiveThirtyEight.com poll")
```

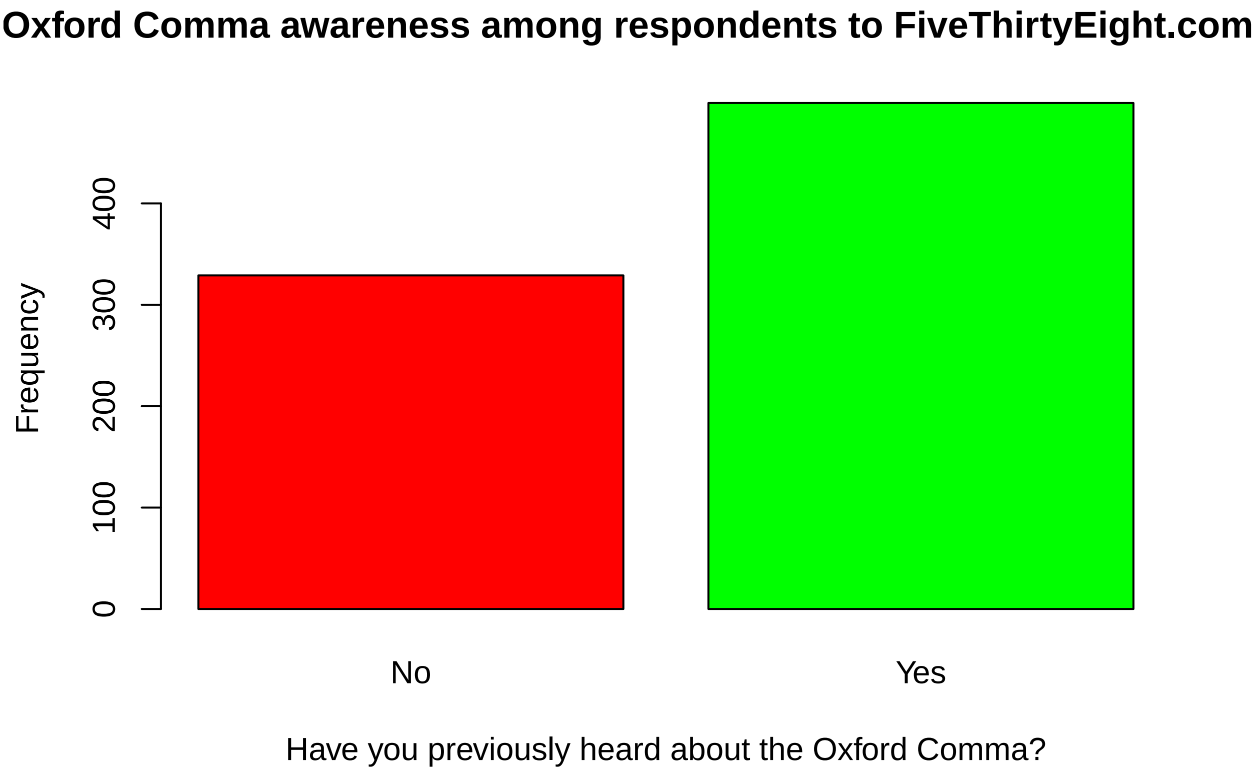


[3] **Has the respondent *previously heard about* the Oxford Comma?**

```
t(t(summary(comma2$HEARD_Oxford)))
```

```
##     [,1]
## No   329
## Yes  499
```

```
barplot(table(comma2$HEARD_Oxford),col=c("red","green"),
        xlab = "Have you previously heard about the Oxford Comma?", ylab = "Frequency",
        main = "Oxford Comma awareness among respondents to FiveThirtyEight.com poll")
```

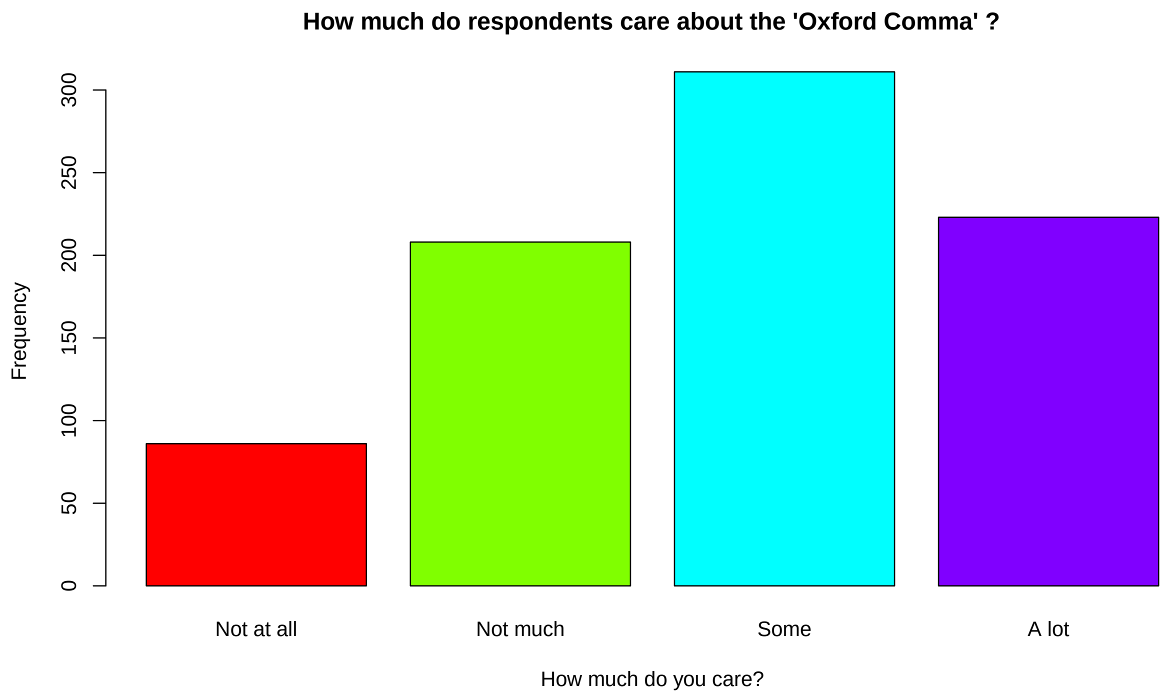


[4] Does the respondent *care about* the Oxford Comma?

```
t(t(summary(comma2$CARE_Oxford))) # already resequenced above to reflect ordering from "A lot" to "Not
```

```
##             [,1]
## Not at all    86
## Not much     208
## Some         311
## A lot        223
```

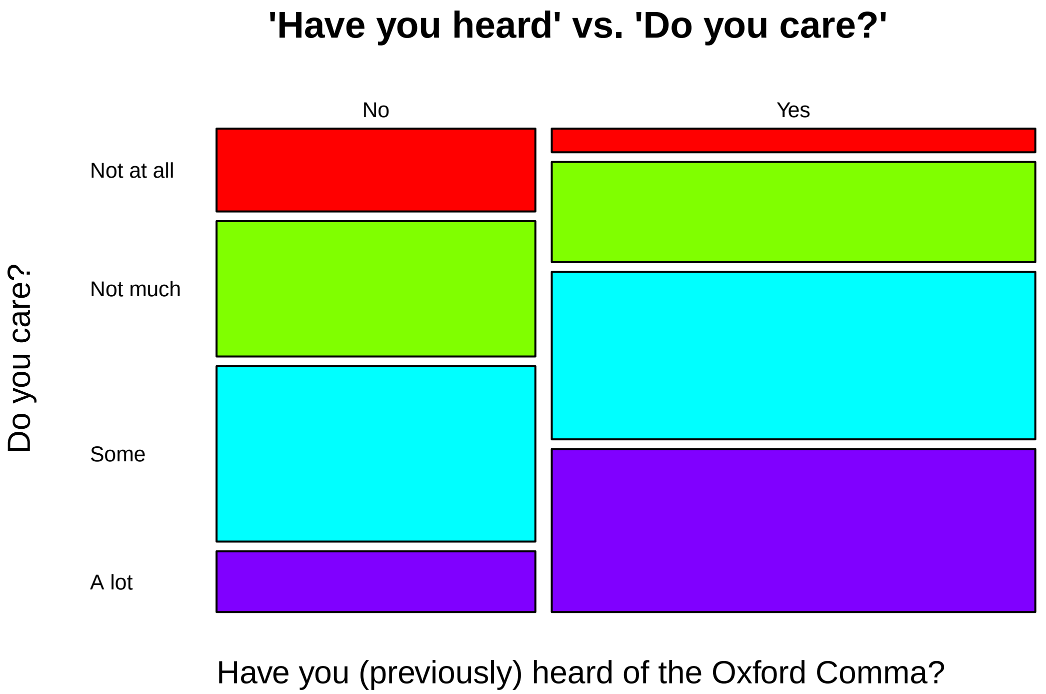
```
### Barplot of degree to which respondent cares about the Oxford Comma
barplot(table(comma2$CARE_Oxford),col=rainbow(4),
        xlab = "How much do you care?", ylab = "Frequency",
        main = "How much do respondents care about the 'Oxford Comma' ?")
```



```
# Have you previously heard about the Oxford Comma vs. how much do you care about it?
table(comma2$CARE_Oxford,comma2$HEARD_Oxford)
```

```
##
##                No Yes
##   Not at all   60  26
##   Not much     98 110
##   Some        127 184
##   A lot        44 179
```

```
mosaicplot(comma2$HEARD_Oxford ~ comma2$CARE_Oxford,col=rainbow(4),las=1,
           xlab="Have you (previously) heard of the Oxford Comma?",
           ylab="Do you care?",
           main="'Have you heard' vs. 'Do you care?'")
```



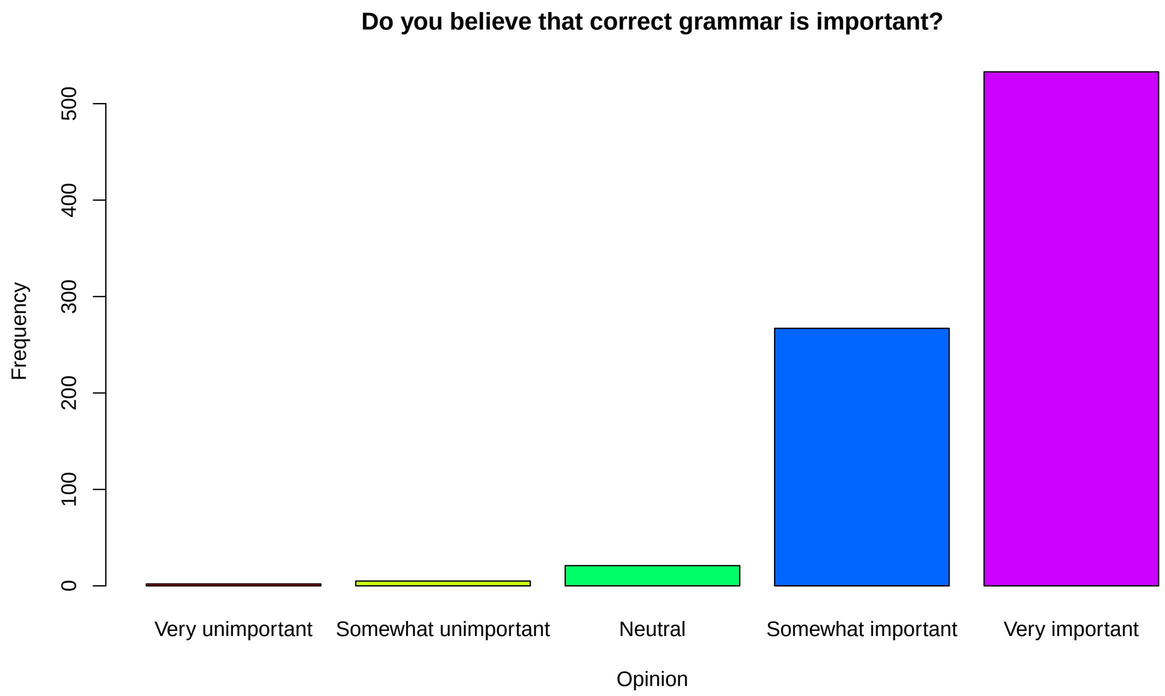
Respondents who have previously heard of the Oxford Comma are more likely to care about it, while those who have not previously heard about it are less likely.

[8] Does the respondent believe that *correct grammar* is important?

```
t(t(summary(comma2$Grammar_Important)))      # already resequenced above to reflect ordering
```

```
##                                              [,1]
## Very unimportant                               2
## Somewhat unimportant                           5
## Neither important nor unimportant (neutral)   21
## Somewhat important                           267
## Very important                               533
```

```
### Barplot of importance of correct grammar
barplot_names = levels(comma2$Grammar_Important)
barplot_names[3]="Neutral"                         # otherwise this entry is too long
barplot(table(comma2$Grammar_Important),col=rainbow(5),names.arg = barplot_names,
        xlab = "Opinion", ylab = "Frequency",
        main = "Do you believe that correct grammar is important?")
```



[9] What is the respondent's *gender* ?

```
t(t(summary(comma2$Gender)))
```

```
##        [,1]
## Female  437
## Male    391
```

Usage of Oxford Comma by gender: Is there any obvious difference?

```
# Number of each gender using Oxford Comma
table(comma2$USES_Oxford, comma2$Gender)
```

```
##
##         Female Male
##   FALSE    187  171
##   TRUE     250  220
```

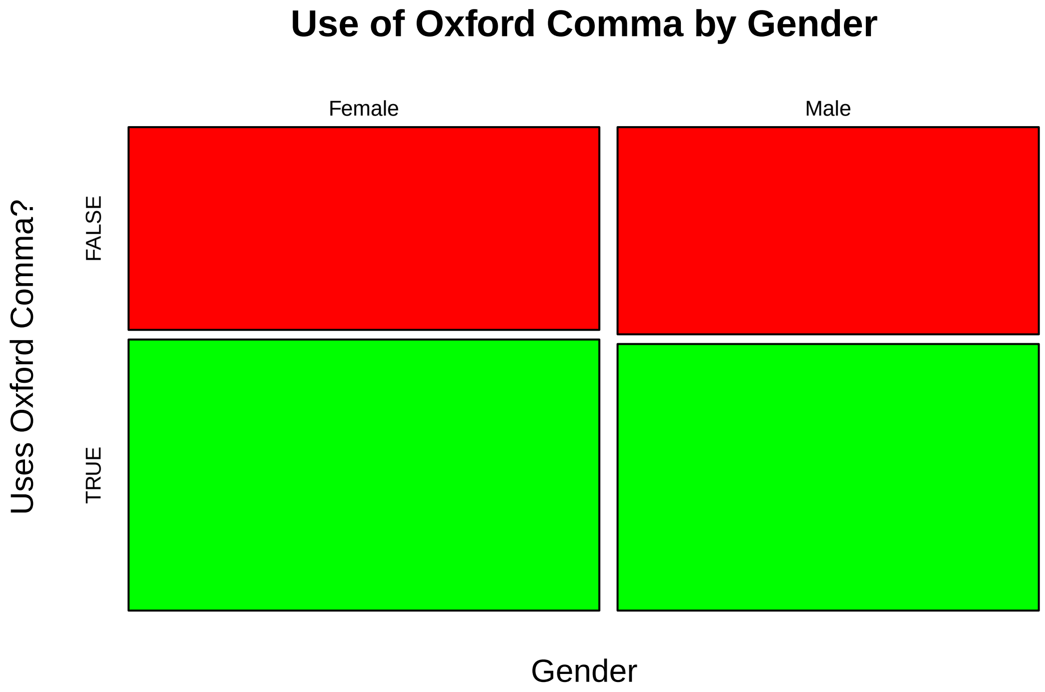
```
# Percentage within each row using the Oxford Comma
table(comma2$USES_Oxford, comma2$Gender)/rowSums(table(comma2$USES_Oxford, comma2$Gender))
```

```
##
##            Female      Male
##   FALSE 0.5223464 0.4776536
##   TRUE  0.5319149 0.4680851
```

```
# Percentage within each column using the Oxford Comma
t(t(table(comma2$USES_Oxford, comma2$Gender))/colSums(table(comma2$USES_Oxford, comma2$Gender)))
```

```
##
##            Female      Male
##   FALSE 0.4279176 0.4373402
##   TRUE  0.5720824 0.5626598
```

```
mosaicplot(comma2$Gender~ comma2$USES_Oxford, col=c("Red","Green"),
           xlab="Gender", ylab="Uses Oxford Comma?",
           main="Use of Oxford Comma by Gender")
```



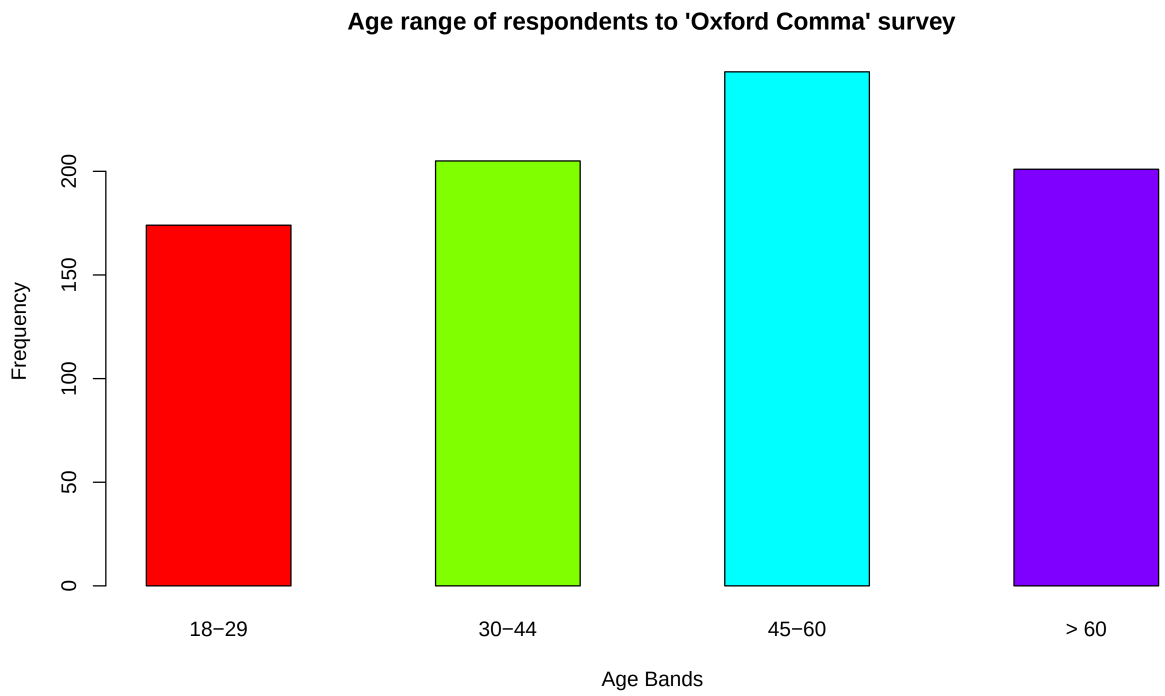
The percentage of each gender using the Oxford Comma is about the same.

[10] What is the respondent's *age band* ?

```
t(t(table(comma2$AgeBands,useNA = "ifany")))      # already resequenced to reflect ordering of age band
```

```
##
##         [,1]
##   18-29  174
##   30-44  205
##   45-60  248
##   > 60   201
```

```
### Barplot of age bands
barplot(table(comma2$AgeBands),col=rainbow(4),space = 1,
        xlab = "Age Bands", ylab = "Frequency",
        main = "Age range of respondents to 'Oxford Comma' survey")
```

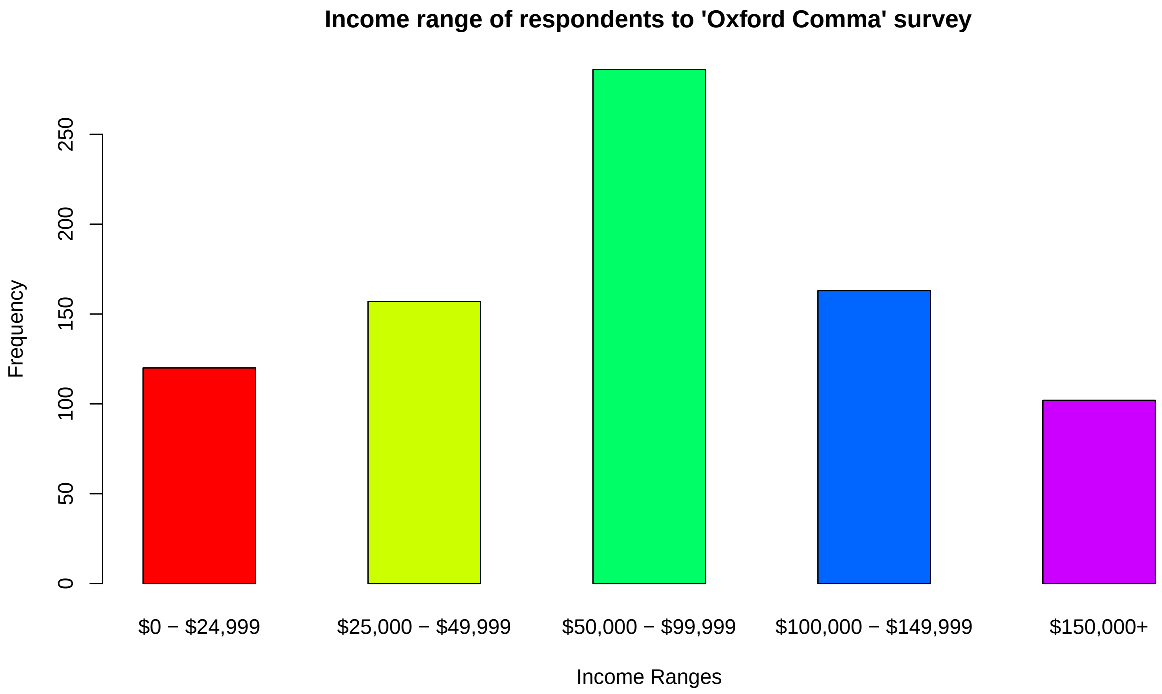


[11] What is the respondent's *income band* ?

```
t(t(table(comma2$IncomeBands,useNA = "ifany"))) # resequenced to reflect ordering of income bands
```

```
##
##                        [,1]
##   $0 - $24,999          120
##   $25,000 - $49,999     157
##   $50,000 - $99,999     286
##   $100,000 - $149,999   163
##   $150,000+             102
```

```
### Barplot of income bands
barplot(table(comma2$IncomeBands),col=rainbow(5),space = 1,
        xlab = "Income Ranges", ylab = "Frequency",
        main = "Income range of respondents to 'Oxford Comma' survey")
```



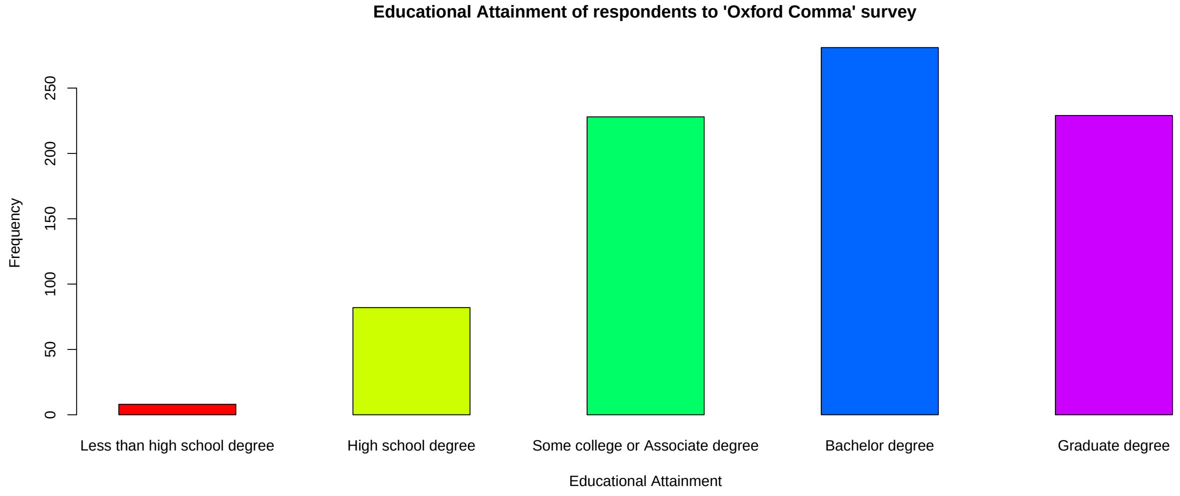
The above results appear to resemble a normal distribution.

[12] What is the respondent's level of *education*?

```
t(t(summary(comma2$Education)))              # already resequenced to reflect ordering of educational a
```

```
##                                   [,1]
## Less than high school degree        8
## High school degree                 82
## Some college or Associate degree  228
## Bachelor degree                   281
## Graduate degree                   229
```

```
### Barplot of educational attainment
barplot(table(comma2$Education),col=rainbow(5),space = 1,
        xlab = "Educational Attainment", ylab = "Frequency",
        main = "Educational Attainment of respondents to 'Oxford Comma' survey")
```



The data is left-skewed, with the median respondent having a bachelor's degree.

[13] Geography

```
# Where is the respondent located?
t(t(summary(comma2$Location)))      # already resequenced to reflect geography (east to west)
```

```
##                      [,1]
## New England            59
## Middle Atlantic       108
## South Atlantic        132
## East North Central    131
## East South Central     38
## West North Central     68
## West South Central     73
## Mountain               67
## Pacific               152
```

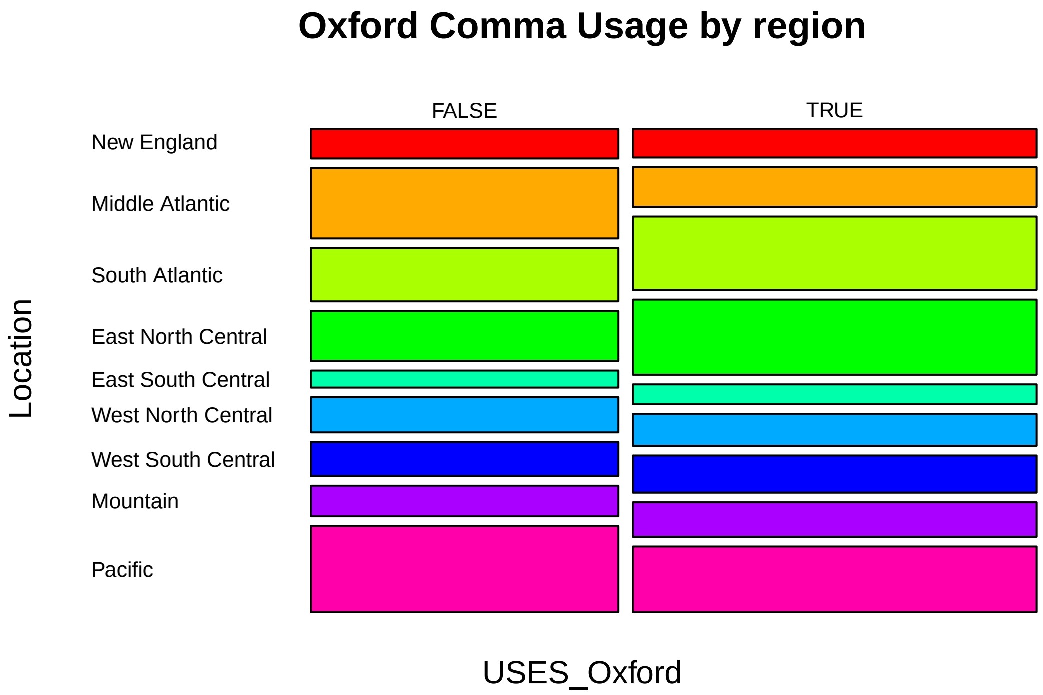
```
EastCoast = summary(comma2$Location)[c(1,2,3)]
MiddleUSA = summary(comma2$Location)[c(4,5,6,7)]
WesternUSA = summary(comma2$Location)[c(8,9)]
```

The respondents are geographically well-distributed across the USA, with 299 from the **East Coast**, 310 from the **Central** portion of the country, and 219 from the **West**.

```
# Table of Oxford Comma Usage by region
table(comma2$Location, comma2$USES_Oxford)
```

```
##
##                       FALSE TRUE
##   New England            26   33
##   Middle Atlantic        62   46
##   South Atlantic         47   85
##   East North Central     44   87
##   East South Central     15   23
##   West North Central     31   37
##   West South Central     30   43
##   Mountain               27   40
##   Pacific                76   76
```

```
# Mosaic Plot
mosaicplot(data=comma2, USES_Oxford ~ Location, col=rainbow(9), las=1, main = "Oxford Comma Usage by reg
```


Oxford Comma Usage by region
FALSE
TRUE
New England
Middle Atlantic
South Atlantic
East North Central
East South Central
West North Central
West South Central
Mountain
Pacific
Location
USES_Oxford