Project1-yz3384

*Yun Zhang-yz3384*

*1/30/2019*

## Loading required package: xml2

##

## Attaching package: 'rvest'

## The following object is masked from 'package:readr':

##

## guess\_encoding

## Warning: package 'gplots' was built under R version 3.5.2

##

## Attaching package: 'gplots'

## The following object is masked from 'package:stats':

##

## lowess

##

## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':

##

## filter, lag

## The following objects are masked from 'package:base':

##

## intersect, setdiff, setequal, union

## Loading required package: NLP

##

## Attaching package: 'syuzhet'

## The following object is masked from 'package:sentimentr':

##

## get\_sentences

## Loading required package: ggplot2

##

## Attaching package: 'ggplot2'

## The following object is masked from 'package:NLP':

##

## annotate

## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at https://goo.gl/13EFCZ

##

## Attaching package: 'scales'

## The following object is masked from 'package:syuzhet':

##

## rescale

## The following object is masked from 'package:readr':

##

## col\_factor

## Warning: package 'RANN' was built under R version 3.5.2

##

## Attaching package: 'wordcloud'

## The following object is masked from 'package:gplots':

##

## textplot

## Parsed with column specification:

## cols(

## hmid = col\_integer(),

## wid = col\_integer(),

## reflection\_period = col\_character(),

## original\_hm = col\_character(),

## cleaned\_hm = col\_character(),

## modified = col\_character(),

## num\_sentence = col\_integer(),

## ground\_truth\_category = col\_character(),

## predicted\_category = col\_character()

## )

## Parsed with column specification:

## cols(

## wid = col\_integer(),

## age = col\_character(),

## country = col\_character(),

## gender = col\_character(),

## marital = col\_character(),

## parenthood = col\_character()

## )

Take a brief look at data

glimpse(cleaned)

## Observations: 100,535

## Variables: 9

## $ hmid <int> 27673, 27674, 27675, 27676, 27677, 27678...

## $ wid <int> 2053, 2, 1936, 206, 6227, 45, 195, 740, ...

## $ reflection\_period <chr> "24h", "24h", "24h", "24h", "24h", "24h"...

## $ original\_hm <chr> "I went on a successful date with someon...

## $ cleaned\_hm <chr> "I went on a successful date with someon...

## $ modified <chr> "True", "True", "True", "True", "True", ...

## $ num\_sentence <int> 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1...

## $ ground\_truth\_category <chr> NA, NA, NA, "bonding", NA, "leisure", NA...

## $ predicted\_category <chr> "affection", "affection", "exercise", "b...

glimpse(demographic)

## Observations: 10,844

## Variables: 6

## $ wid <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, ...

## $ age <chr> "37.0", "29.0", "25", "32", "29", "35", "34", "29",...

## $ country <chr> "USA", "IND", "IND", "USA", "USA", "IND", "USA", "V...

## $ gender <chr> "m", "m", "m", "m", "m", "m", "m", "m", "f", "m", "...

## $ marital <chr> "married", "married", "single", "married", "married...

## $ parenthood <chr> "y", "y", "n", "y", "y", "y", "y", "n", "y", "n", "...

Inspect an overall wordcloud

sentences=cleaned$cleaned\_hm

sentences1=Corpus(VectorSource(sentences))

ff.all<-tm\_map(sentences1, stripWhitespace)

## Warning in tm\_map.SimpleCorpus(sentences1, stripWhitespace): transformation

## drops documents

ff.all<-tm\_map(ff.all, content\_transformer(tolower))

## Warning in tm\_map.SimpleCorpus(ff.all, content\_transformer(tolower)):

## transformation drops documents

ff.all<-tm\_map(ff.all, removeWords, stopwords("english"))

## Warning in tm\_map.SimpleCorpus(ff.all, removeWords, stopwords("english")):

## transformation drops documents

ff.all<-tm\_map(ff.all, removeWords, character(0))

## Warning in tm\_map.SimpleCorpus(ff.all, removeWords, character(0)):

## transformation drops documents

ff.all<-tm\_map(ff.all, removePunctuation)

## Warning in tm\_map.SimpleCorpus(ff.all, removePunctuation): transformation

## drops documents

limit\_words = c("happy","went","time","got","really","nice","yesterday","one","finally","today","life","

came","get","able","great","well","first","year","lot","morning","getting","felt","like","moment","new","week","just","going","long","see","good","back","came","found","hours","month","months","weekend","two","little","days","took","around","people","event","now","big","can","together","made","day","years","much","last","will","best","weeks","happiest","make","fun","friends","favorite","feel","past","started","working","ago","happiness","things")

ff.all <- tm\_map(ff.all, removeWords, limit\_words)

## Warning in tm\_map.SimpleCorpus(ff.all, removeWords, limit\_words):

## transformation drops documents

tdm.all<-TermDocumentMatrix(ff.all)

tdm.tidy=tidy(tdm.all)

tdm.overall=summarise(group\_by(tdm.tidy, term), sum(count))

wordcloud(tdm.overall$term, tdm.overall$`sum(count)`,

scale=c(5,0.5),

max.words=50,

min.freq=1,

random.order=FALSE,

rot.per=0.3,

use.r.layout=T,

random.color=FALSE,

colors=brewer.pal(9,"Blues"))

People mention work, friend ,family, home most, then followed by daughter, son, dinner and birthday.

Basic analysis

Calculate word length

number\_words <- cleaned %>%

unnest\_tokens(word, cleaned\_hm) %>%

group\_by(hmid) %>%

summarise(words\_number = n()) %>%

arrange(desc(words\_number))

cleaned\_1 <- right\_join(cleaned,number\_words, by='hmid')

ggplot(data=cleaned\_1,aes(x=words\_number,fill="red"))+

geom\_histogram(binwidth = 5)+

xlim(-1,100)+

labs(xlab = "Number of words", ylab = "Frequency")+

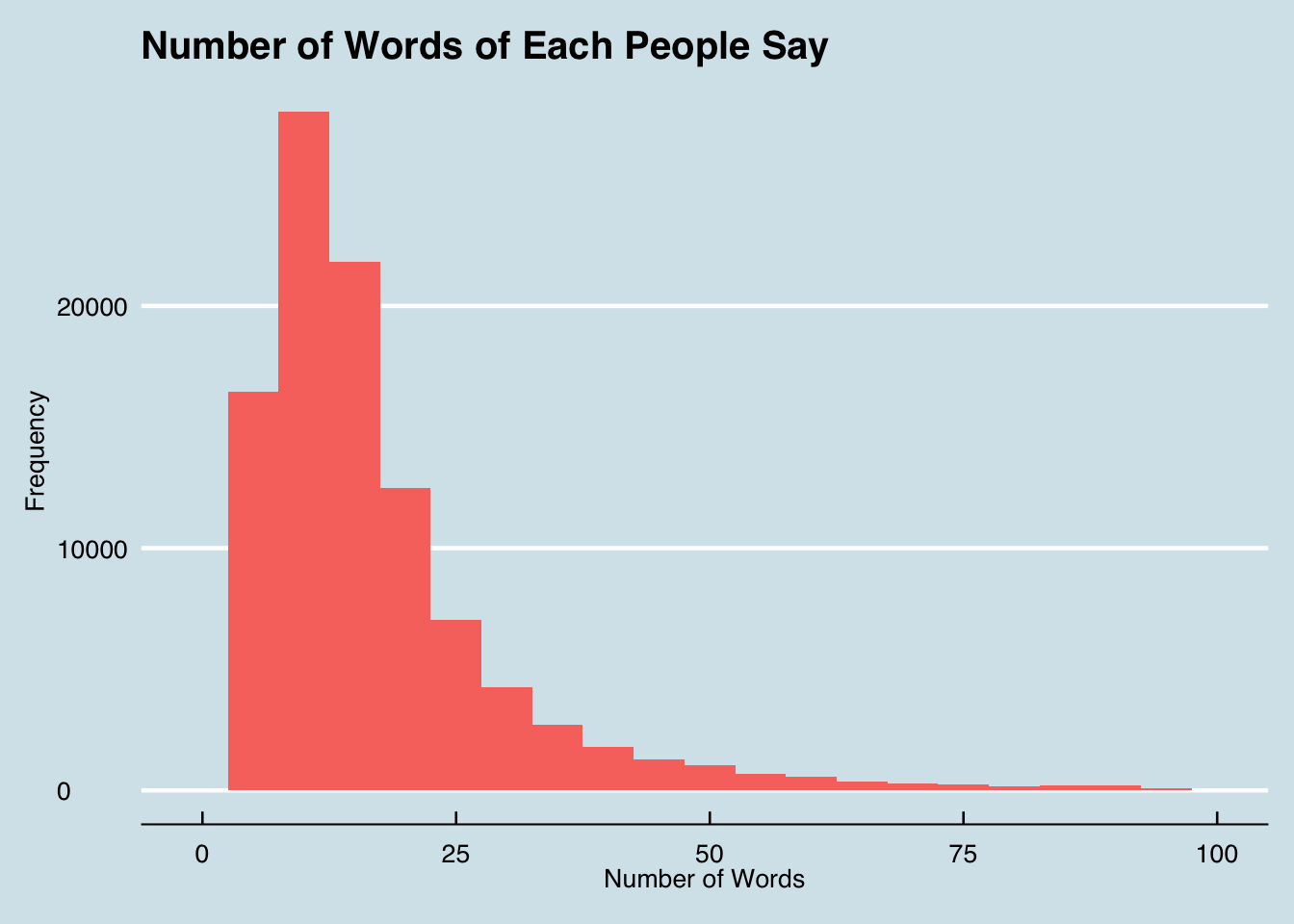
guides(fill=FALSE)+

theme\_economist()+

labs(title="Number of Words of Each People Say",x="Number of Words",y="Frequency")

## Warning: Removed 716 rows containing non-finite values (stat\_bin).

## Warning: Removed 2 rows containing missing values (geom\_bar).

Most people say less than 15 words. And when the number of the words increases, the number of people decreases.

Analyze age distribution

demographic$age <- as.numeric(demographic$age)

## Warning: NAs introduced by coercion

ggplot(data=demographic,aes(age))+

geom\_histogram(binwidth = 5,fill='blue')+

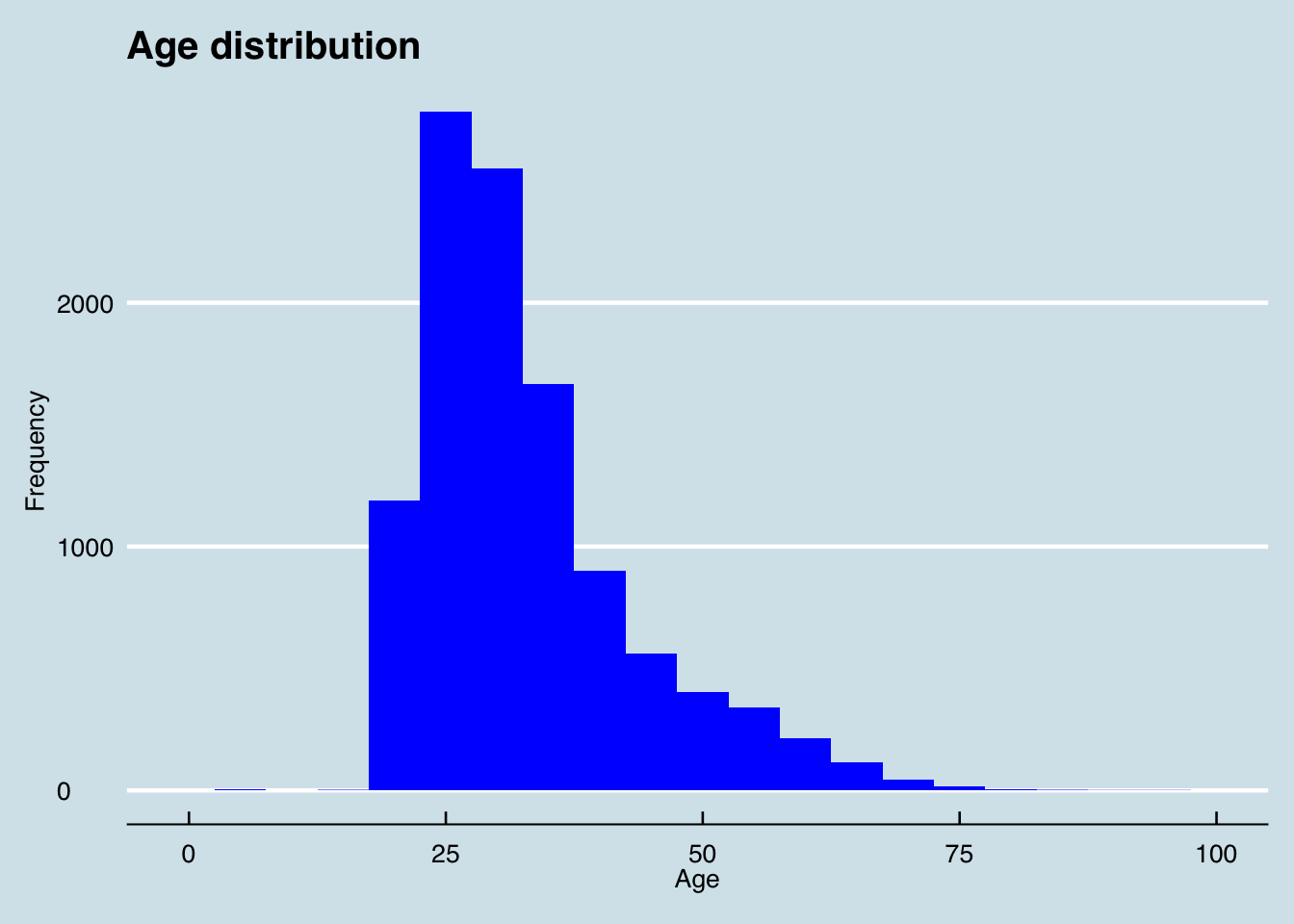
xlim(-1,100)+

theme\_economist()+

labs(title="Age distribution",x="Age",y="Frequency")

## Warning: Removed 40 rows containing non-finite values (stat\_bin).

## Warning: Removed 2 rows containing missing values (geom\_bar).

This histogram graphic shows the distribution of age.  
Most people are about 20 to 30 years old. The number of people decreases when the age of people increases after 20.

Analyze agestage

demographic <- demographic %>%

mutate(agestage =

ifelse(demographic$age %**in**% 11:25, "Teenager",

ifelse(demographic$age %**in**% 26:40, "Young adult",

ifelse(demographic$age %**in**% 41:65, "Middle aged",

ifelse(demographic$age %**in**% 66:90, "Old aged",

"NA")))))

cleaned\_2 <- left\_join(cleaned\_1,select(demographic,wid,agestage), by='wid')

cleaned\_2$agestage <- factor(cleaned\_2$agestage)

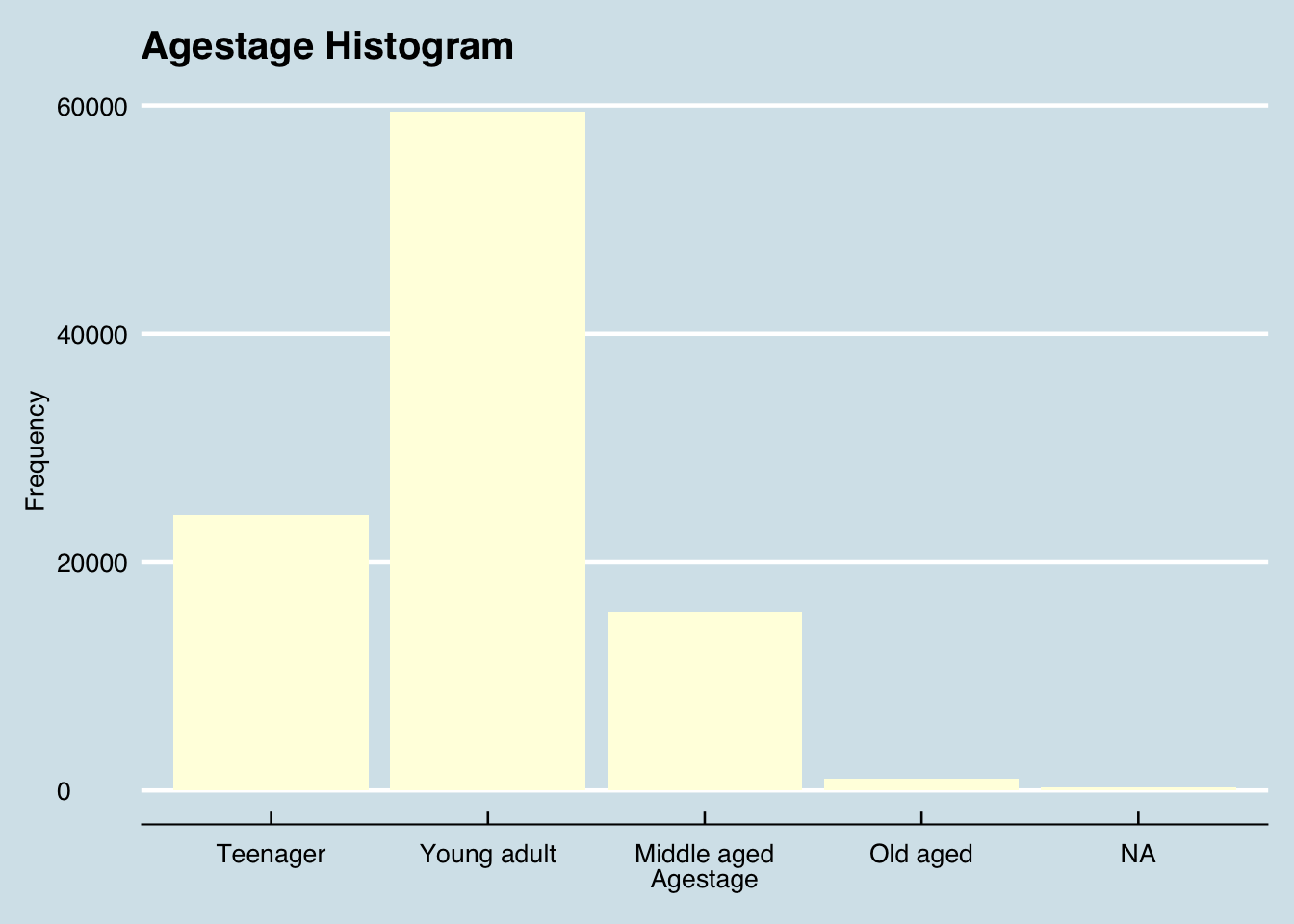
cleaned\_2$agestage <- factor(cleaned\_2$agestage, levels = c('Teenager','Young adult','Middle aged','Old aged','NA'))

ggplot(data=cleaned\_2,aes(agestage))+

geom\_bar(fill='lightyellow')+

theme\_economist()+

labs(title="Agestage Histogram",x="Agestage",y="Frequency")

Young adults account for the main part, followed by teenagers and middle aged. The old aged poeple are the least.

Analyze emotion using NRC sentiment lexion

words<-paste(cleaned$cleaned\_hm)

emotions=get\_nrc\_sentiment(words)

emotions\_sum=apply(emotions,2,sum)

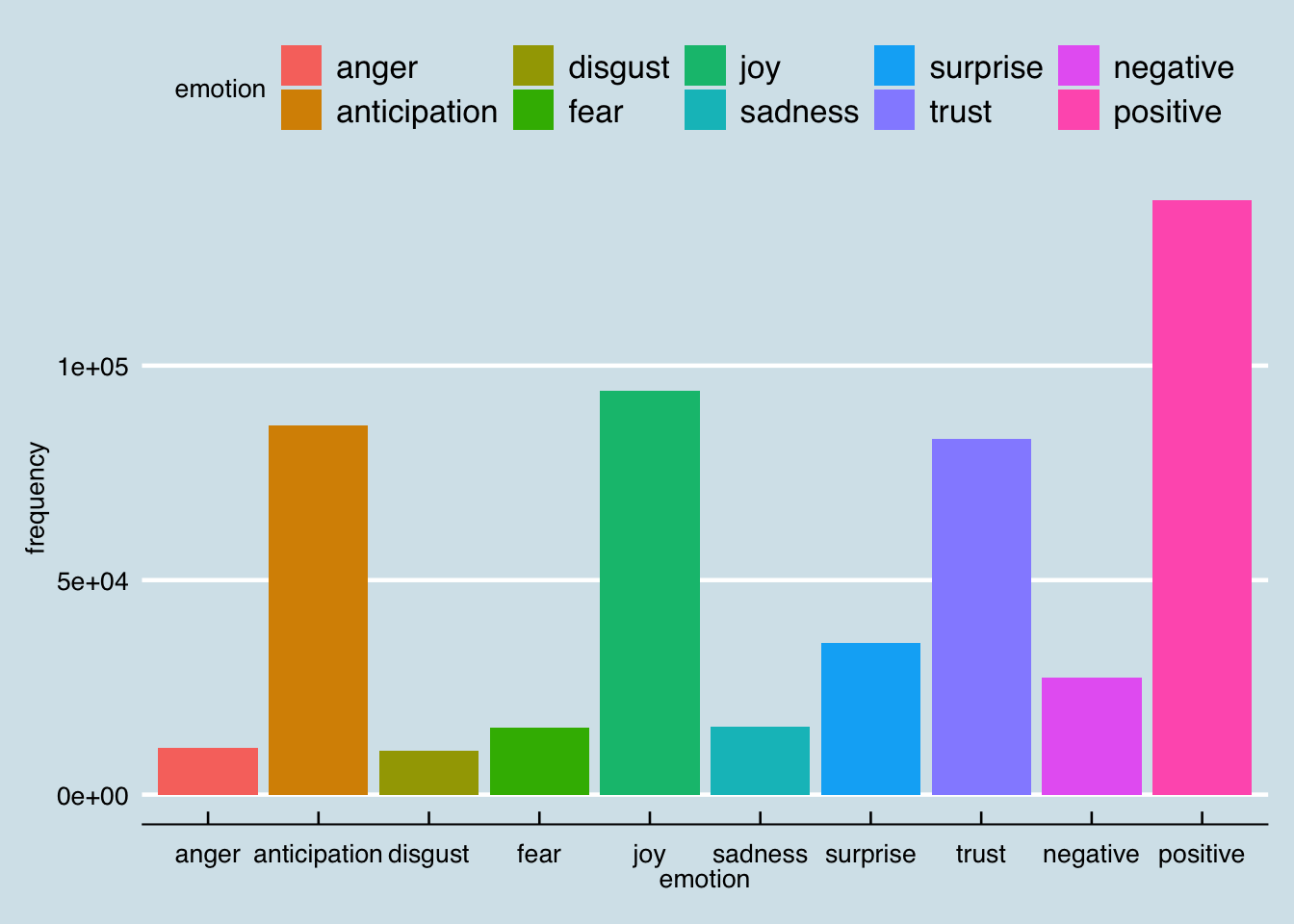
emotions\_df=data.frame(emotion=colnames(emotions),frequency=emotions\_sum)

emotions\_df$emotion = factor(emotions\_df$emotion, levels=c("anger","anticipation","disgust","fear","joy","sadness","surprise","trust","negative","positive"))

ggplot(data=emotions\_df,aes(emotion,y=frequency,fill=emotion))+

geom\_bar(stat = "identity")+

theme\_economist()

The expressions of people are most related to joy, anticipation and trust. Although the topic of sentences people speak is happiness, there are still negative related words which express fear, sadness, disgust and anger.

Analyze most five happiness moments of people at each agestage

agestage\_name=unique(cleaned\_2$agestage)

age\_word=NULL

**for** (age **in** agestage\_name) {

data=as.vector(cleaned\_2[which(cleaned\_2$agestage==age),"cleaned\_hm"])

word=Corpus(VectorSource(data))

ff.all1<-tm\_map(word, stripWhitespace)

ff.all1<-tm\_map(ff.all1, content\_transformer(tolower))

ff.all1<-tm\_map(ff.all1, removeWords, stopwords("english"))

ff.all1<-tm\_map(ff.all1, removeWords, character(0))

ff.all1<-tm\_map(ff.all1, removePunctuation)

ff.all1 <- tm\_map(ff.all1, removeWords, limit\_words)

tdm.all1<-TermDocumentMatrix(ff.all1)

tdm.tidy1=tidy(tdm.all1)

tdm.overall1=as.data.frame(summarise(group\_by(tdm.tidy1, term), sum(count)))

tdm.overall1=tdm.overall1[order(tdm.overall1[,"sum(count)"],decreasing = TRUE),]

age\_top\_word=cbind(tdm.overall1[1:5,],rep(age,5))

age\_word=rbind(age\_word,age\_top\_word)

}

## Warning in tm\_map.SimpleCorpus(word, stripWhitespace): transformation drops

## documents

## Warning in tm\_map.SimpleCorpus(ff.all1, content\_transformer(tolower)):

## transformation drops documents

## Warning in tm\_map.SimpleCorpus(ff.all1, removeWords, stopwords("english")):

## transformation drops documents

## Warning in tm\_map.SimpleCorpus(ff.all1, removeWords, character(0)):

## transformation drops documents

## Warning in tm\_map.SimpleCorpus(ff.all1, removePunctuation): transformation

## drops documents

## Warning in tm\_map.SimpleCorpus(ff.all1, removeWords, limit\_words):

## transformation drops documents

## Warning in tm\_map.SimpleCorpus(word, stripWhitespace): transformation drops

## documents

## Warning in tm\_map.SimpleCorpus(ff.all1, content\_transformer(tolower)):

## transformation drops documents

## Warning in tm\_map.SimpleCorpus(ff.all1, removeWords, stopwords("english")):

## transformation drops documents

## Warning in tm\_map.SimpleCorpus(ff.all1, removeWords, character(0)):

## transformation drops documents

## Warning in tm\_map.SimpleCorpus(ff.all1, removePunctuation): transformation

## drops documents

## Warning in tm\_map.SimpleCorpus(ff.all1, removeWords, limit\_words):

## transformation drops documents

## Warning in tm\_map.SimpleCorpus(word, stripWhitespace): transformation drops

## documents

## Warning in tm\_map.SimpleCorpus(ff.all1, content\_transformer(tolower)):

## transformation drops documents

## Warning in tm\_map.SimpleCorpus(ff.all1, removeWords, stopwords("english")):

## transformation drops documents

## Warning in tm\_map.SimpleCorpus(ff.all1, removeWords, character(0)):

## transformation drops documents

## Warning in tm\_map.SimpleCorpus(ff.all1, removePunctuation): transformation

## drops documents

## Warning in tm\_map.SimpleCorpus(ff.all1, removeWords, limit\_words):

## transformation drops documents

## Warning in tm\_map.SimpleCorpus(word, stripWhitespace): transformation drops

## documents

## Warning in tm\_map.SimpleCorpus(ff.all1, content\_transformer(tolower)):

## transformation drops documents

## Warning in tm\_map.SimpleCorpus(ff.all1, removeWords, stopwords("english")):

## transformation drops documents

## Warning in tm\_map.SimpleCorpus(ff.all1, removeWords, character(0)):

## transformation drops documents

## Warning in tm\_map.SimpleCorpus(ff.all1, removePunctuation): transformation

## drops documents

## Warning in tm\_map.SimpleCorpus(ff.all1, removeWords, limit\_words):

## transformation drops documents

## Warning in tm\_map.SimpleCorpus(word, stripWhitespace): transformation drops

## documents

## Warning in tm\_map.SimpleCorpus(ff.all1, content\_transformer(tolower)):

## transformation drops documents

## Warning in tm\_map.SimpleCorpus(ff.all1, removeWords, stopwords("english")):

## transformation drops documents

## Warning in tm\_map.SimpleCorpus(ff.all1, removeWords, character(0)):

## transformation drops documents

## Warning in tm\_map.SimpleCorpus(ff.all1, removePunctuation): transformation

## drops documents

## Warning in tm\_map.SimpleCorpus(ff.all1, removeWords, limit\_words):

## transformation drops documents

names(age\_word)<-c("word","frequency","agestage")

age\_word[-which(age\_word$agestage=="NA"),]

## word frequency agestage

## 21527 work 4985 Young adult

## 7956 friend 3331 Young adult

## 7190 family 2834 Young adult

## 9318 home 2543 Young adult

## 12988 night 2117 Young adult

## 13109 work 1639 Teenager

## 4842 friend 1573 Teenager

## 4399 family 1048 Teenager

## 5684 home 918 Teenager

## 1380 birthday 849 Teenager

## 11598 work 1186 Middle aged

## 2802 daughter 1010 Middle aged

## 9682 son 896 Middle aged

## 4293 friend 794 Middle aged

## 3889 family 695 Middle aged

## 2808 wife 79 Old aged

## 683 daughter 51 Old aged

## 1236 home 51 Old aged

## 2383 son 50 Old aged

## 1055 friend 42 Old aged

The dataframe shows the top five words people from different agestage mention.  
We can see that young adults and teenagers both mention work, friend, family and home most. But the difference is that young adults aslo consider night creates happiness more while teenagers like birthday more. In addition, middle aged and old aged both mention daughter, son and friend much. However, middle aged people like work best while old aged love wife more.

Analysis about predicted category with other variables

Predict category & agestage

cleaned\_age=cleaned\_2[cleaned\_2$agestage == "Young adult" |cleaned\_2$agestage == "Teenager"|cleaned\_2$agestage == "Middle aged",]

ggplot(cleaned\_age, aes(x=as.factor(cleaned\_age$predicted\_category))) +

facet\_wrap(~cleaned\_age$agestage) +

geom\_bar(aes(y=..count../sum(..count..),fill=cleaned\_age$predicted\_category)) +

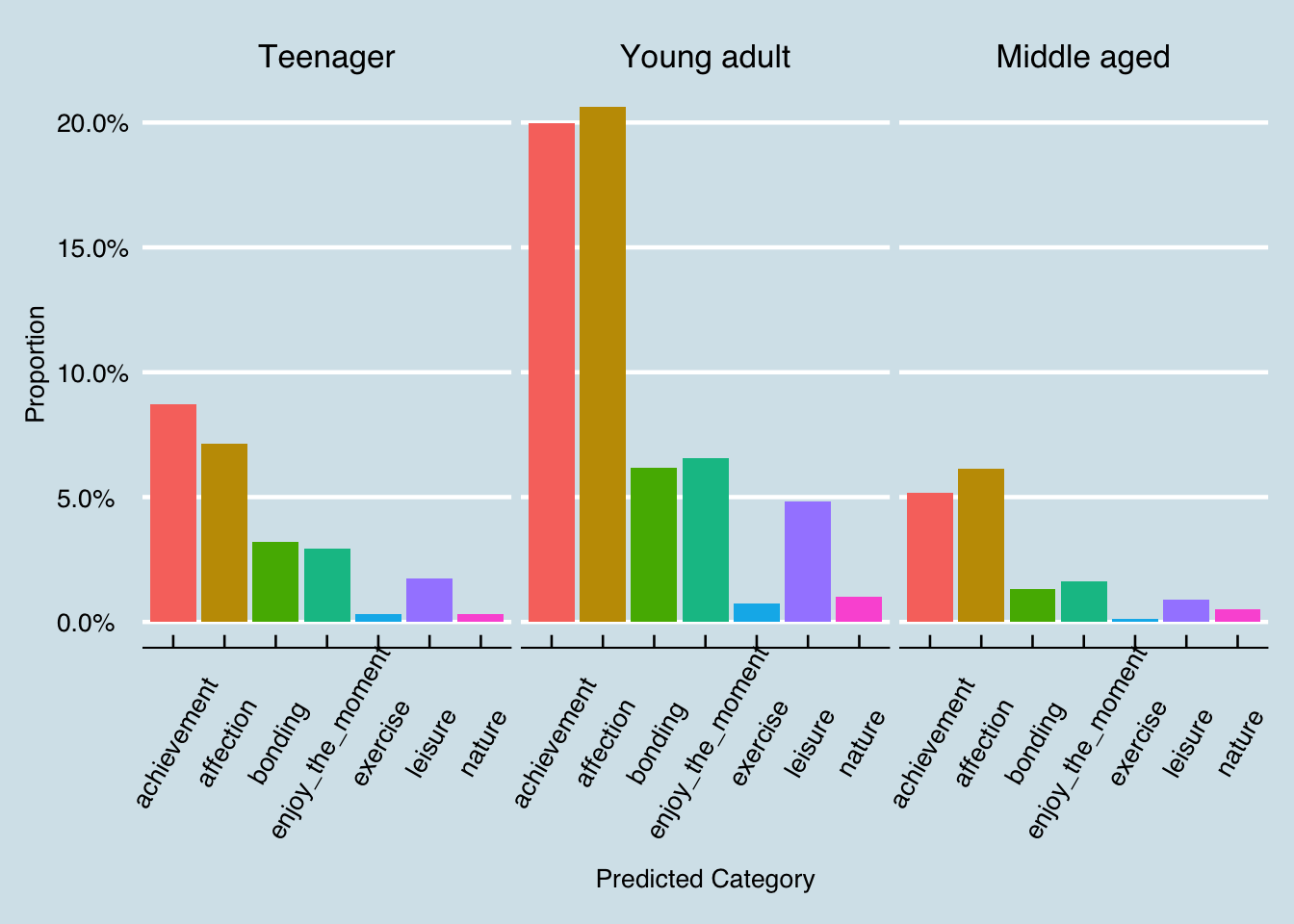
scale\_y\_continuous(labels=percent\_format())+

labs(x="Predicted Category",y="Proportion") +

theme\_economist()+

guides(fill=FALSE)+

theme(axis.text.x = element\_text(angle = 60,vjust=0.6,hjust=0.5))

Teenagers, young adults and middle aged people all like achievement and affection most, but young adults have stronger feelings about that.

Predict category & gender

cleaned\_3 <- left\_join(cleaned\_2,select(demographic,wid,gender), by='wid')

cleaned\_gender <- cleaned\_3[cleaned\_3$gender == "f" |cleaned\_3$gender == "m",]

cleaned\_gender<-cleaned\_gender[-which(is.na(cleaned\_gender$gender)),]

ggplot(cleaned\_gender, aes(x=as.factor(cleaned\_gender$predicted\_category))) +

facet\_wrap(~cleaned\_gender$gender) +

geom\_bar(aes(y=..count../sum(..count..),fill=cleaned\_gender$predicted\_category)) +

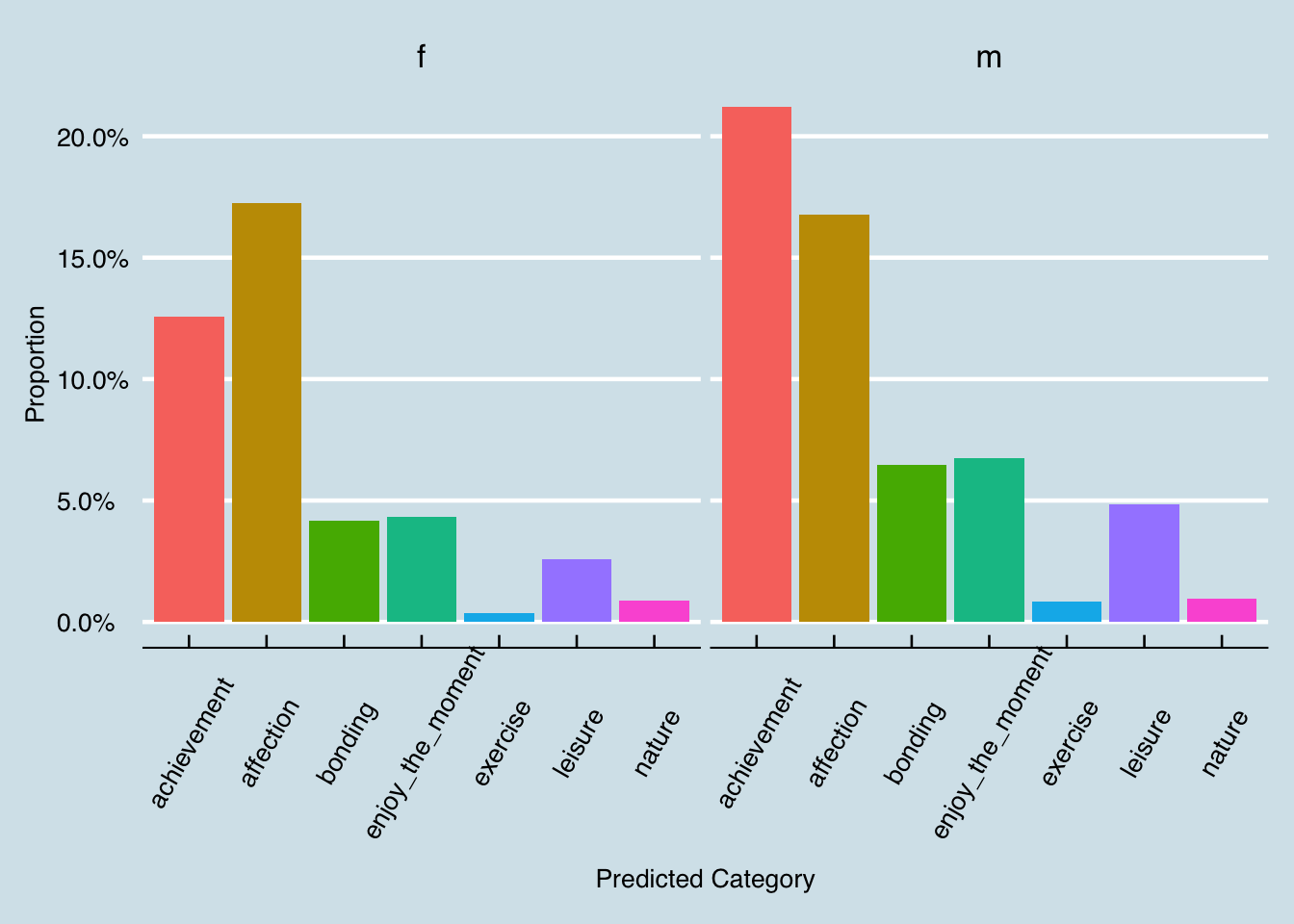
scale\_y\_continuous(labels=percent\_format())+

labs(x="Predicted Category",y="Proportion") +

theme\_economist()+

guides(fill=FALSE)+

theme(axis.text.x = element\_text(angle = 60,vjust=0.6,hjust=0.5))

Female and male both like talking about affection. However, male prefer to discuss achievenment more than affection and female are the opposite.

Predict category & marital

cleaned\_4 <- left\_join(cleaned\_3,select(demographic,wid,marital), by='wid')

cleaned\_marital <- cleaned\_4[cleaned\_4$marital == "married" |cleaned\_4$marital == "single",]

cleaned\_marital<-cleaned\_marital[-which(is.na(cleaned\_marital$marital)),]

ggplot(cleaned\_marital, aes(x=as.factor(cleaned\_marital$predicted\_category))) +

facet\_wrap(~cleaned\_marital$marital) +

geom\_bar(aes(y=..count../sum(..count..),fill=cleaned\_marital$predicted\_category)) +

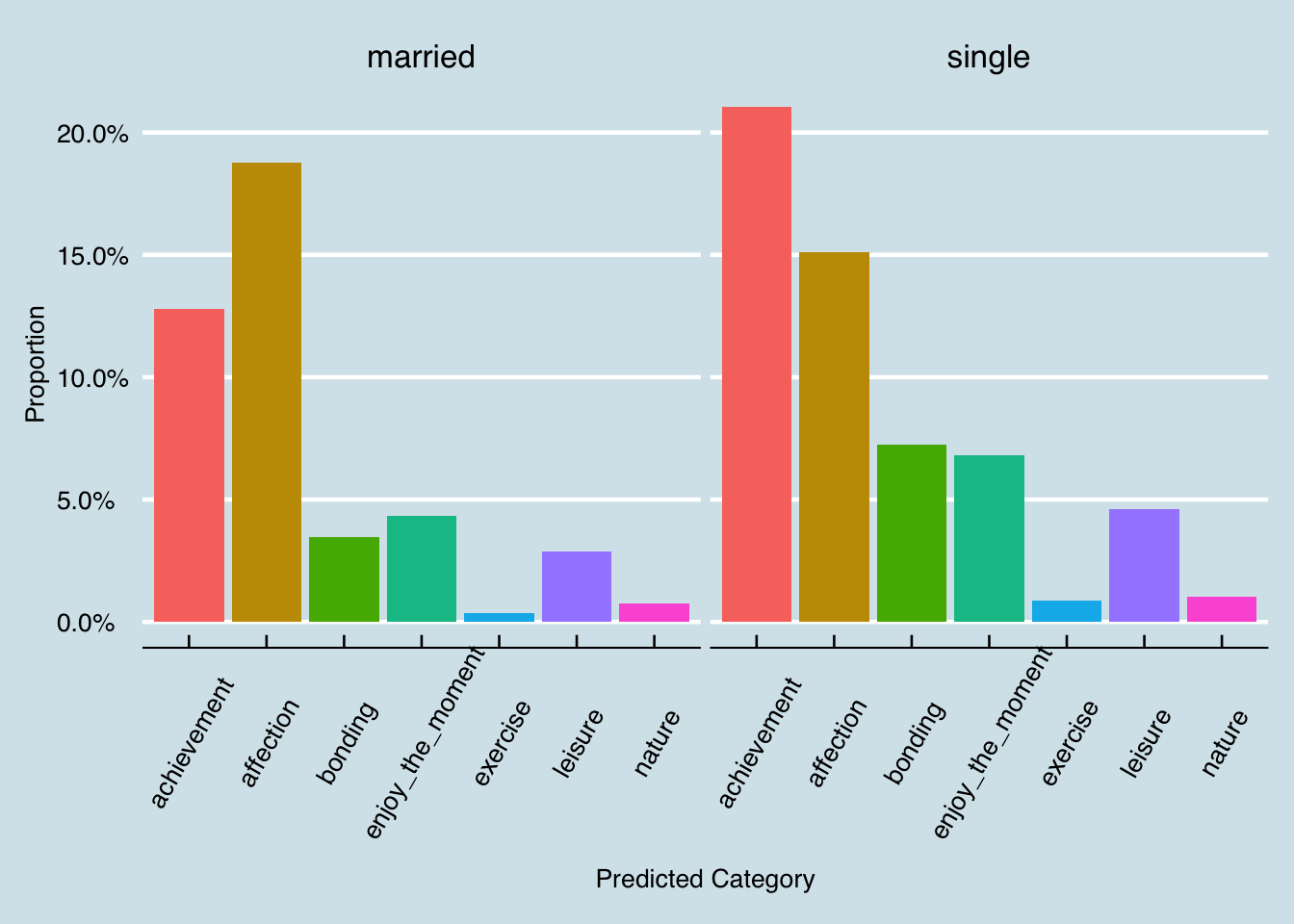
scale\_y\_continuous(labels=percent\_format())+

labs(x="Predicted Category",y="Proportion") +

theme\_economist()+

guides(fill=FALSE)+

theme(axis.text.x = element\_text(angle = 60,vjust=0.6,hjust=0.5))

Married people and single people both like achievement and affection most. However, married people prefer affection better while single people prefer achievement more. In addition, the remaining components of predict category are only small parts of happiness. But single people think bonding, enjoying the moment and leisure more than married people.

Predict category & reflection period

ggplot(cleaned, aes(x=as.factor(cleaned$predicted\_category))) +

facet\_wrap(~cleaned$reflection\_period) +

geom\_bar(aes(y=..count../sum(..count..),fill=cleaned$predicted\_category)) +

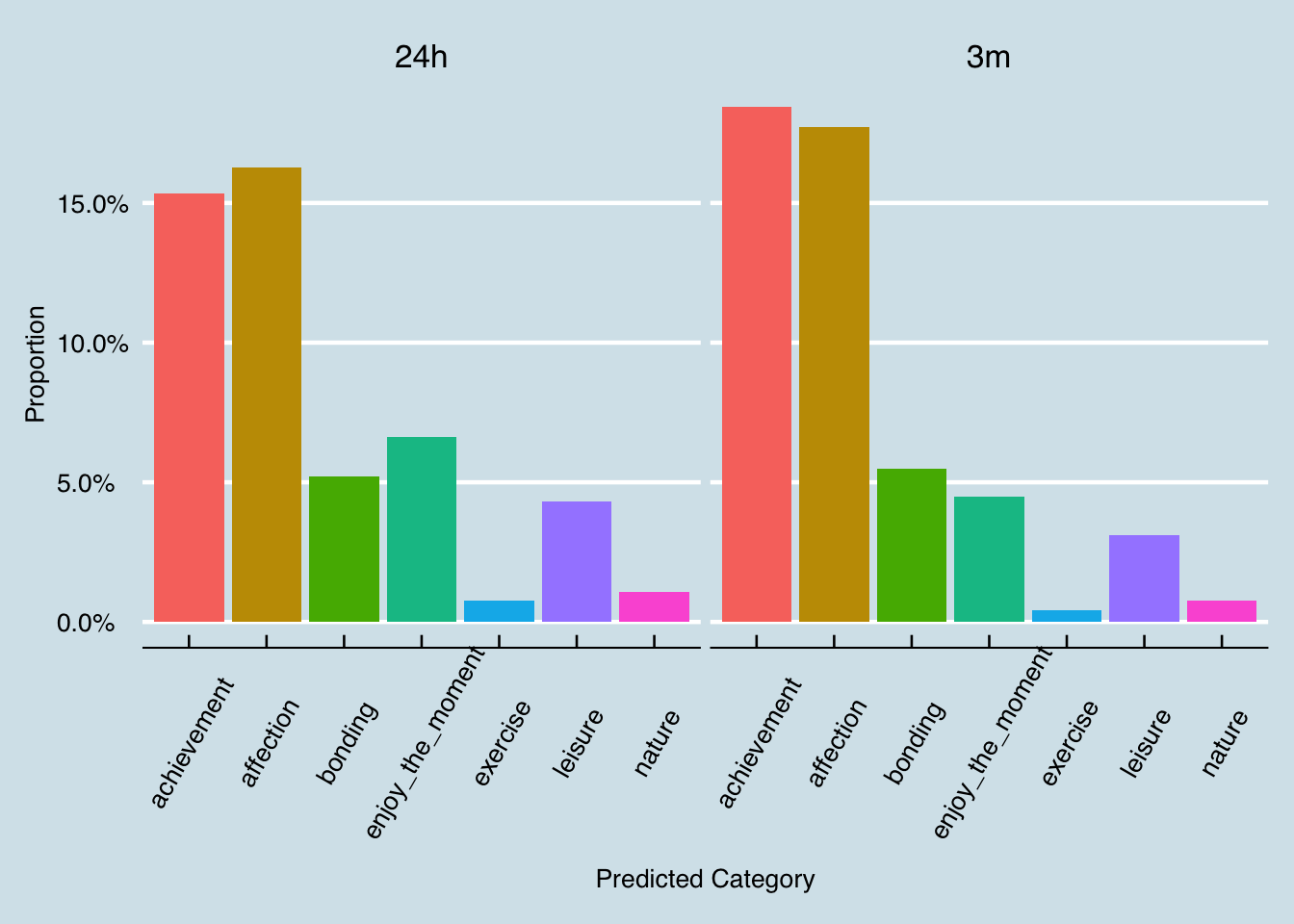
scale\_y\_continuous(labels=percent\_format())+

labs(x="Predicted Category",y="Proportion") +

theme\_economist()+

guides(fill=FALSE)+

theme(axis.text.x = element\_text(angle = 60,vjust=0.6,hjust=0.5))

From the graphic above, there does not exist great differences between reflection period of 24 hours and 3 months. People in two groups all like achievement and affection more. The small difference is that people with 3 months reflection period are more likely to consider achievement and affection as the happiest moment. In contrast, people with one day reflection period think enjoying the moment and leisure matter more than people have 3 months reflection period.

Prediction category & parenthood

cleaned\_5 <- left\_join(cleaned\_4,select(demographic,wid,parenthood), by='wid')

cleaned\_parenthood <- cleaned\_5[cleaned\_5$parenthood == "n" |cleaned\_5$parenthood == "y",]

cleaned\_parenthood<-cleaned\_parenthood[-which(is.na(cleaned\_parenthood$parenthood)),]

ggplot(cleaned\_parenthood, aes(x=as.factor(cleaned\_parenthood$predicted\_category))) +

facet\_wrap(~cleaned\_parenthood$parenthood) +

geom\_bar(aes(y=..count../sum(..count..),fill=cleaned\_parenthood$predicted\_category)) +

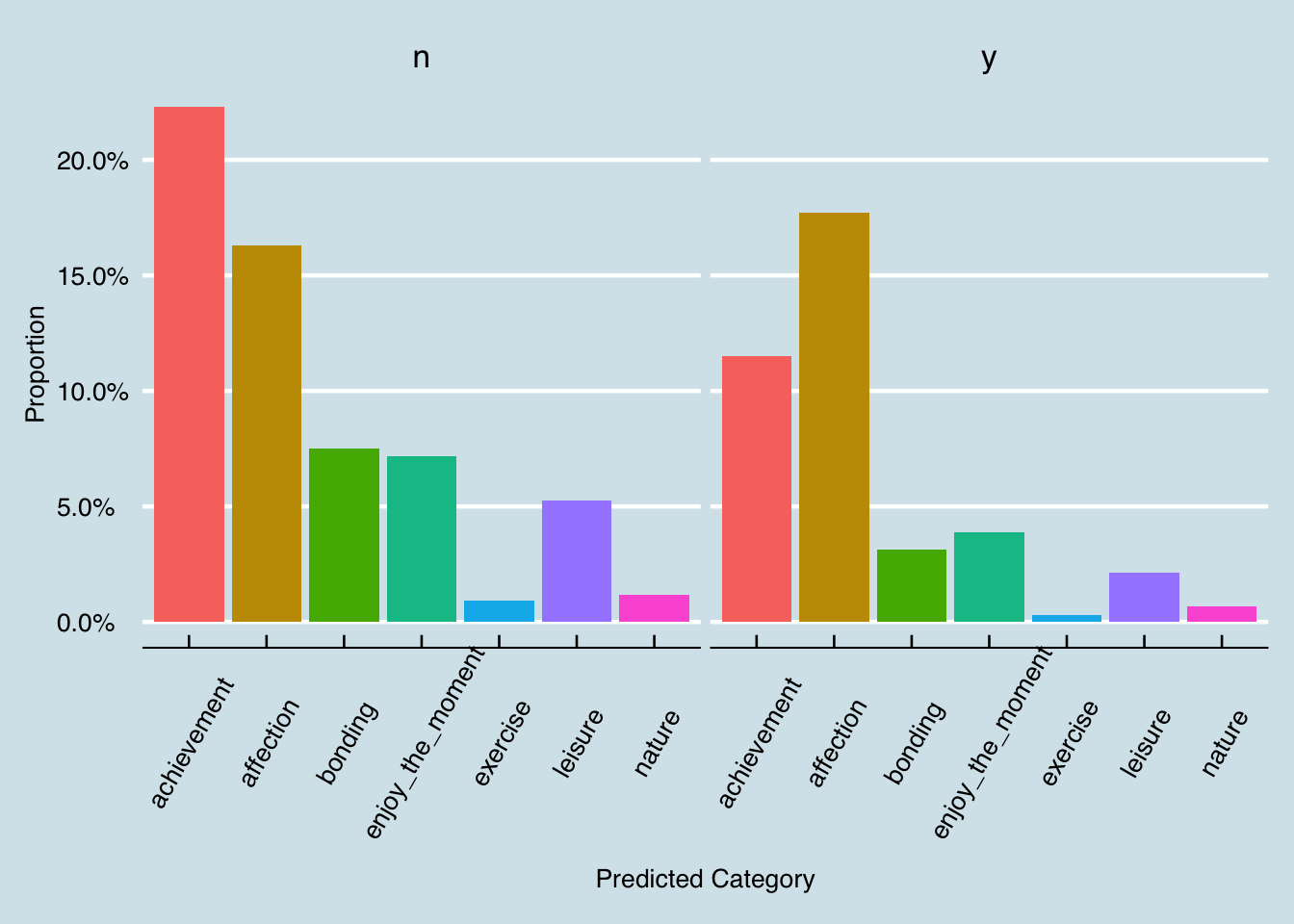
scale\_y\_continuous(labels=percent\_format())+

labs(x="Predicted Category",y="Proportion") +

theme\_economist()+

guides(fill=FALSE)+

theme(axis.text.x = element\_text(angle = 60,vjust=0.6,hjust=0.5))

People who have children like to discuss about affection most, but people without children like achievement more than affection.