# Happy Moments: A Statistical Analysis

Leslie

2024-01-27

## Motivation

Does the type of happiness change as we age? Or are the types of happiness differ by demographic. This projects aims to find the relationship between happiness and nationality, marital status, and parenthood.

## Data Exploration

Checking the raw data to understand what they are.

```
head(clean)
```

```
##    hmid  wid reflection_period
## 1 27673 2053               24h
## 2 27674    2               24h
## 3 27675 1936               24h
## 4 27676  206               24h
## 5 27677 6227               24h
## 6 27678   45               24h
##
original_hm
## 1                                                      I went on a successful date wi
th someone I felt sympathy and connection with.
## 2                                                                          I was happ
y when my son got 90% marks in his examination
## 3
I went to the gym this morning and did yoga.
## 4 We had a serious talk with some friends of ours who have been flaky lately. They un
derstood and we had a good evening hanging out.
## 5                                                             I went with grandchildr
en to butterfly display at Crohn Conservatory\n
## 6
I meditated last night.
##
cleaned_hm
## 1                                                      I went on a successful date wi
th someone I felt sympathy and connection with.
## 2                                                                          I was happ
y when my son got 90% marks in his examination
## 3
I went to the gym this morning and did yoga.
## 4 We had a serious talk with some friends of ours who have been flaky lately. They un
derstood and we had a good evening hanging out.
## 5                                                             I went with grandchildr
en to butterfly display at Crohn Conservatory\n
## 6
I meditated last night.
##   modified num_sentence ground_truth_category predicted_category
## 1     True            1                                affection
## 2     True            1                                affection
## 3     True            1                                 exercise
## 4     True            2               bonding            bonding
## 5     True            1                                affection
## 6     True            1               leisure            leisure
```

```
head(demo)
```

```
##   wid  age country gender marital parenthood
## 1   1 37.0     USA      m married          y
## 2   2 29.0     IND      m married          y
## 3   3   25     IND      m  single          n
## 4   4   32     USA      m married          y
## 5   5   29     USA      m married          y
## 6   6   35     IND      m married          y
```

```
head(family)
```

```
##             aunt
## 1         auntie
## 2        aunties
## 3          aunts
## 4          aunty
## 5        brother
## 6 brother-in-law
```

Merging the demographic info with cleaned data set, removing duplicates and NAs

```
merged_df = merge(clean, demo, on = "wid", all = T)
duplicate_rows = duplicated(merged_df)
unique_df = merged_df[!duplicate_rows, ]
complete_rows = complete.cases(unique_df)
df = unique_df[complete_rows, ]
```

```
duplicate_rows = duplicated(demo)
demo = demo[!duplicate_rows, ]
complete_rows = complete.cases(demo)
demo = demo[complete_rows, ]
```

Adding the total count to the cleaned data

```
wid_counts = table(df$wid)
df$entries = wid_counts[match(df$wid, names(wid_counts))]
df$entries = as.integer(df$entries)
```

# Creating Variables

Getting the count for the total count of memories by predicted category, duration, but also the combination of both

```
result_sum = df %>%
  group_by(wid) %>%
  summarize(occurrence = n()) %>%
  pivot_wider(names_from = "wid", values_from = "occurrence", values_fill = 0)
result_sum = t(result_sum)
result_sum = data.frame(result_sum)
colnames(result_sum) = "count_total"
result_sum$wid = unique(df$wid)



result_detailed = df %>%
  group_by(reflection_period, predicted_category, wid) %>%
  summarize(occurrence = n()) %>%
  pivot_wider(names_from = c("reflection_period", "predicted_category"), values_from =
"occurrence", values_fill = 0)

result_time = df %>%
  group_by(reflection_period, wid) %>%
  summarize(occurrence = n()) %>%
  pivot_wider(names_from = "reflection_period", values_from = "occurrence", values_fill
= 0)

result_df = df %>%
  group_by(predicted_category, wid) %>%
  summarize(occurrence = n()) %>%
  pivot_wider(names_from = "predicted_category", values_from = "occurrence", values_fill
= 0)
```

Getting the explainatory variables by worker ID

```
dep = df %>%
  select(wid, age, marital, country, parenthood)

unique_marital = unique(df$marital)
unique_parenthood = unique(df$parenthood)
unique_country = unique(df$country)
for (marital_val in unique_marital) {
  dep = dep %>%
    mutate(!!paste0("marital_", marital_val, "_binary") := as.integer(marital == marital
_val))
}

for (parenthood_val in unique_parenthood) {
  dep = dep %>%
    mutate(!!paste0("parenthood_", parenthood_val, "_binary") := as.integer(parenthood =
= parenthood_val))
}

for (country_val in unique_country) {
  dep = dep %>%
    mutate(!!paste0("country_", country_val, "_binary") := as.integer(country == country
_val))
}
```

Remove duplicates and NAs

```
duplicate_rows = duplicated(dep)
dep = dep[!duplicate_rows, ]
complete_rows = complete.cases(dep)
dep = dep[complete_rows, ]
```

Making sure the dataframes have the same number of rows

```
stopifnot(length(dep$wid) == length(result_sum$wid))
```

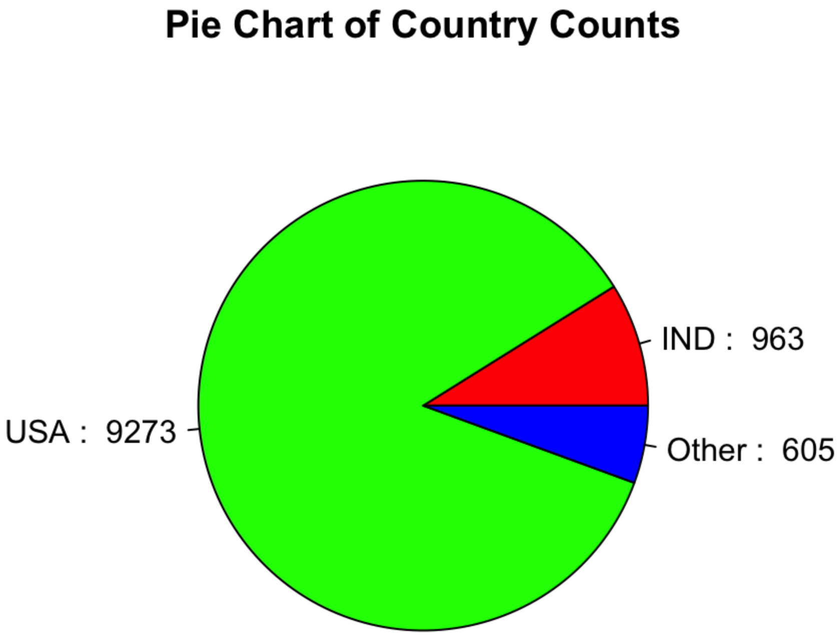
# Data Visualization

Checking the country information. There are quite a few countries but it is dominated by USA and IND. As a result, we group every other country together into “Other” So we do not create variable which only applies to a few data points. Obviously this incorporates a vast number of nations and might include biases that are not accounted for in the scope of the project.

```
country_counts = table(dep$country)
country_counts
```

```
## 
##       AFG  ALB  ARE  ARG  ARM  ASM  AUS  AUT  BEL  BGD  BGR  BHS  BRA  BRB  CAN 
##   72    3    4    6    2    1    3   14    3    4    2    3    1   14    2   66 
##  CHL  COL  CRI  CYP  CZE  DEU  DNK  DOM  DZA  ECU  EGY  ESP  EST  ETH  FIN  FRA 
##    2    6    1    1    2    8    1    2    3    1    5    6    2    1    5   10 
##  GBR  GHA  GMB  GRC  GTM  HKG  HRV  IDN  IND  IRL  IRQ  ISL  ISR  ITA  JAM  JPN 
##   48    1    1    9    2    1    2    6  963    3    1    2    1   11    7    2 
##  KAZ  KEN  KNA  KOR  KWT  LKA  LTU  LVA  MAC  MAR  MDA  MEX  MKD  MLT  MUS  MYS 
##    1    2    1    1    1    2    6    1    1    2    2   22    5    2    1    5 
##  NGA  NIC  NLD  NOR  NPL  NZL  PAK  PER  PHL  POL  PRI  PRT  ROU  RUS  SAU  SGP 
##   12    2    3    1    2    8    5    2   32    4    2    8    5    1    1    6 
##  SLV  SRB  SUR  SVN  SWE  TCA  THA  TTO  TUN  TUR  TWN  UGA  UKR  UMI  URY  USA 
##    1    6    1    1    1    2    3    3    1    8    1    4    1    4    1 9273 
##  VEN  VIR  VNM  ZAF  ZMB 
##   54    1    2    7    1
```

```r
country_counts_processed = country_counts
country_counts_processed[!(names(country_counts_processed) %in% c("USA", "IND"))] <- 0
country_counts_processed = c(country_counts_processed, Other = sum(country_counts[!(names(country_counts) %in% c("USA", "IND"))]))
country_counts_processed = country_counts_processed[country_counts_processed != 0]
pie(country_counts_processed, labels = paste(names(country_counts_processed), ": ", country_counts_processed), main = "Pie Chart of Country Counts", col = rainbow(length(country_counts_processed)))
```



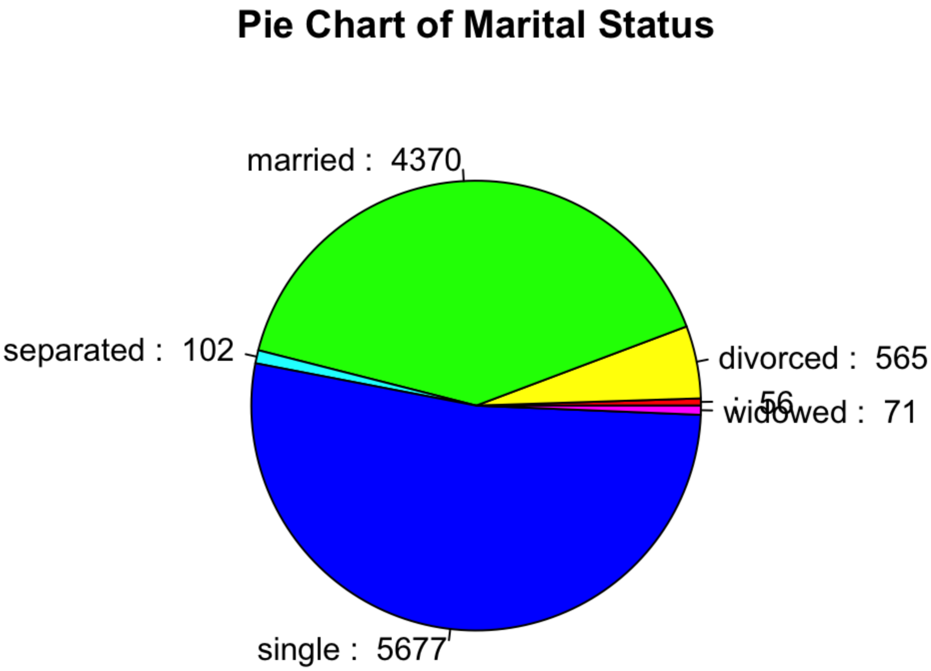
```
#pie(country_counts_processed, main = "Pie Chart of Country Counts", col = rainbow(length(country_counts_processed)))
```

Changing our data to add “other” for country

```
dep = dep %>%
  mutate(country_Other_binary = ifelse(country_USA_binary == 0 & country_IND_binary == 0, 1, 0))
```

Similarly, we can observe that the marital status that is not single or married is quite a bit smaller than the others. Although not to as extreme a degree. Still, we group widowed, divorced, and separated together

```
fam =  table(dep$marital)
pie(fam, labels = paste(names(fam), ": ", fam), main = "Pie Chart of Marital Status", col = rainbow(length(fam)))
```

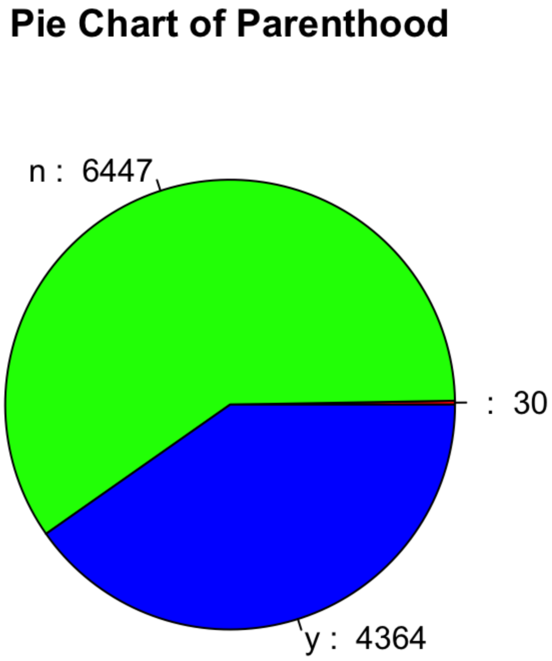


We name the new variable “Lost Partner”

```
dep = dep %>%
  mutate(marital_lostpartner_binary = ifelse(marital_divorced_binary == 1 | marital_separated_binary == 1 | marital_widowed_binary == 1, 1, 0))
```

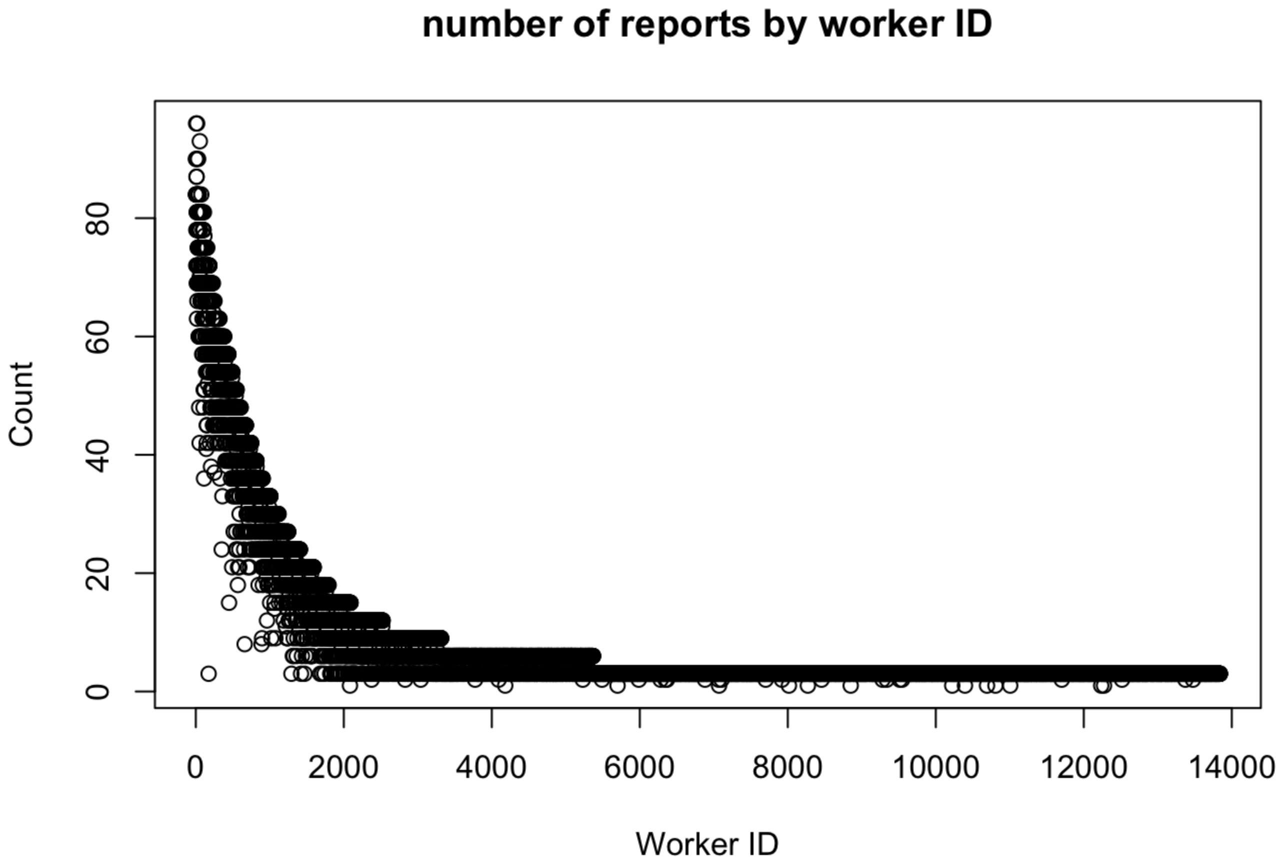
In contrast, parenthood seems to be fine as is, so we keep the variables as is

```
parent = table(dep$parenthood)

pie(parent, labels = paste(names(parent), ": ", parent), main = "Pie Chart of Parenthoo
d", col = rainbow(length(parent)))
```

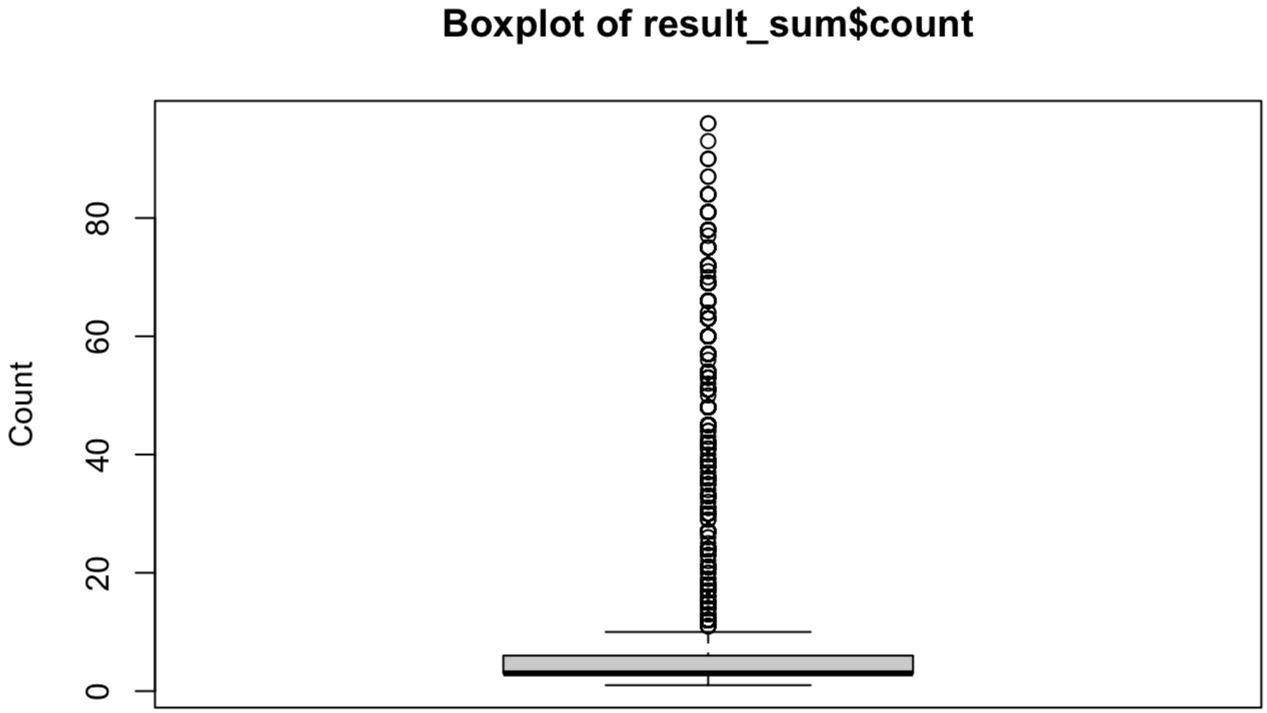


Now we take a look at the interpretations of the response variable. Starting with how many moments each employee reported. It seems there’s a strong trend downward as wid increases, but this should not be an issue as it is not part of the regression. The vast amount of people reported 10 or less instances of happiness.

```
plot(result_sum$wid, result_sum$count, main = "number of reports by worker ID", xlab =
"Worker ID", ylab = "Count")
```

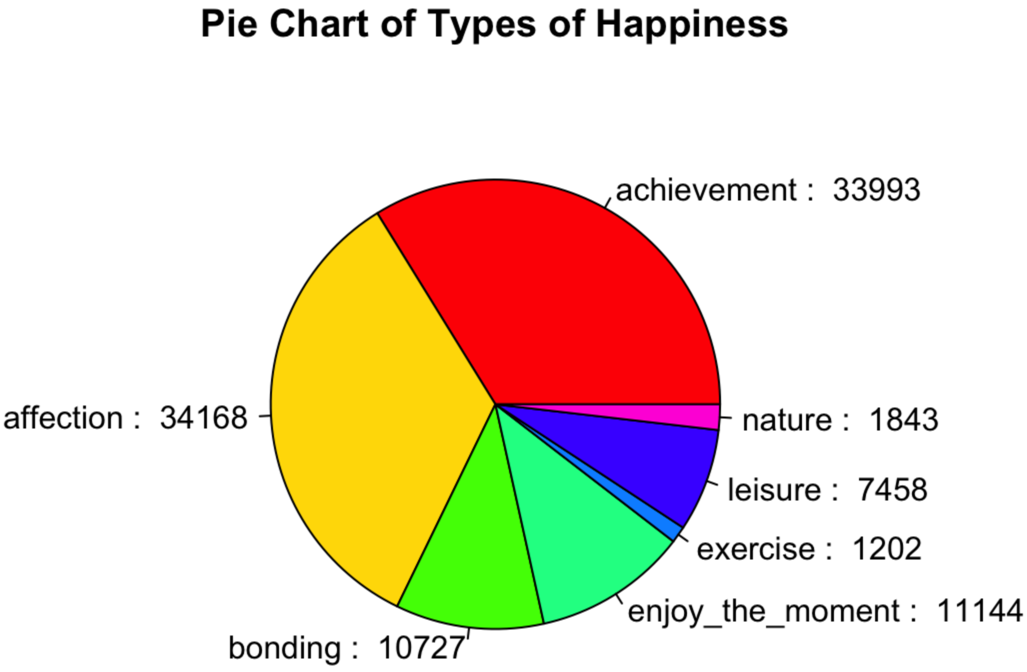


```
boxplot(result_sum$count, main = "Boxplot of result_sum$count", ylab = "Count")
```



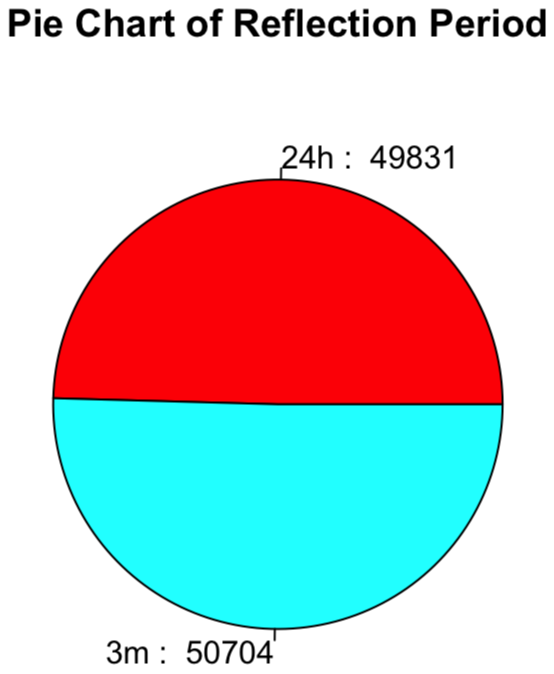
We can also take a look at the distribution of types of happiness. It seems that achievement and affection are the major players, with bonding and enjoy the moment also significant. If we are curious, we can attempt to regress on specific types of happiness, at least the major ones should have enough data

```
happy = table(df$predicted_category)
pie(happy, labels = paste(names(happy), ": ", happy), main = "Pie Chart of Types of Happiness", col = rainbow(length(happy)))
```

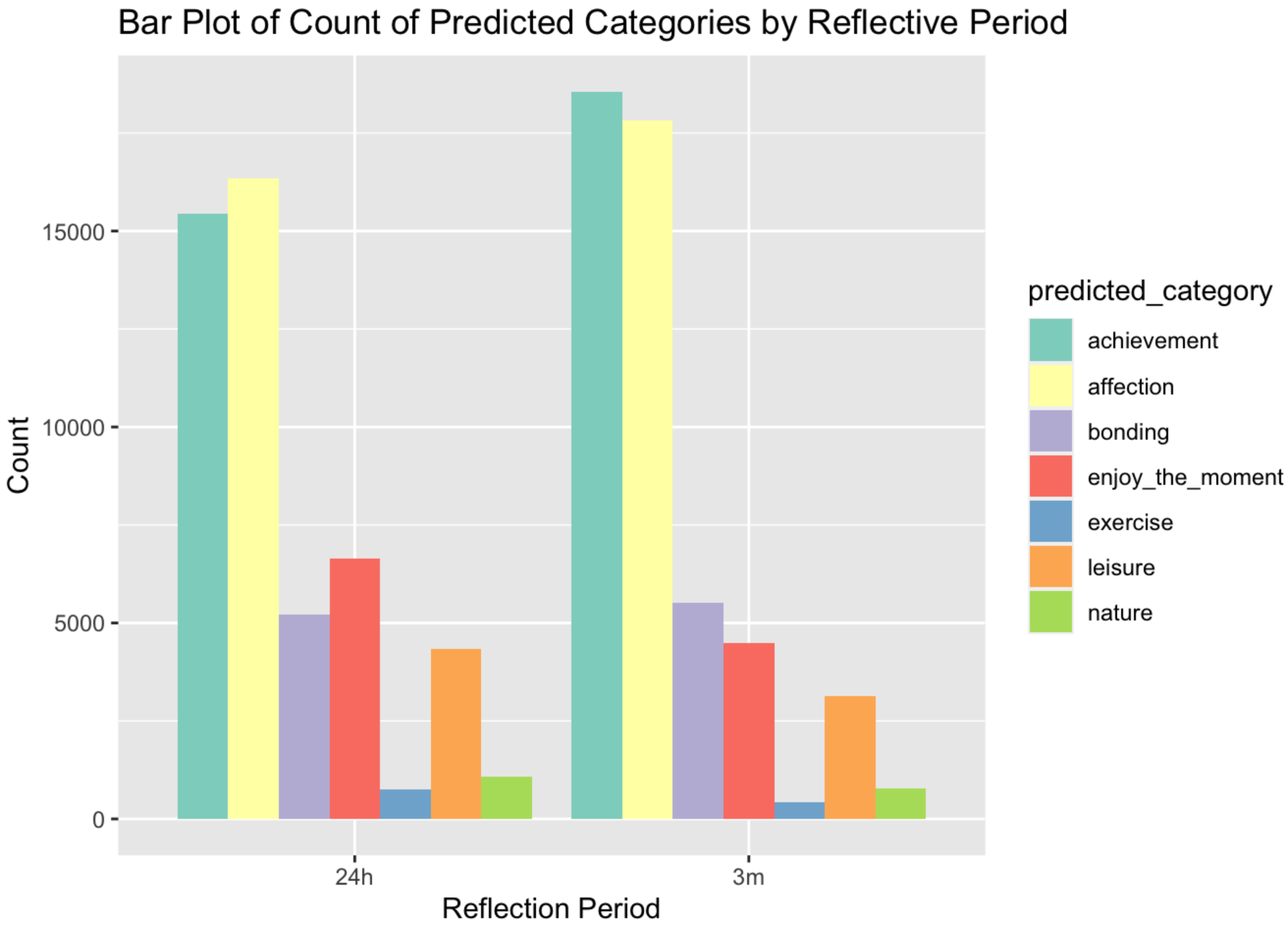


The reflection time seems very even. A side by side comparison shows that there isn’t a ton of difference of the types of happiness, regardless of which reflection period was used.

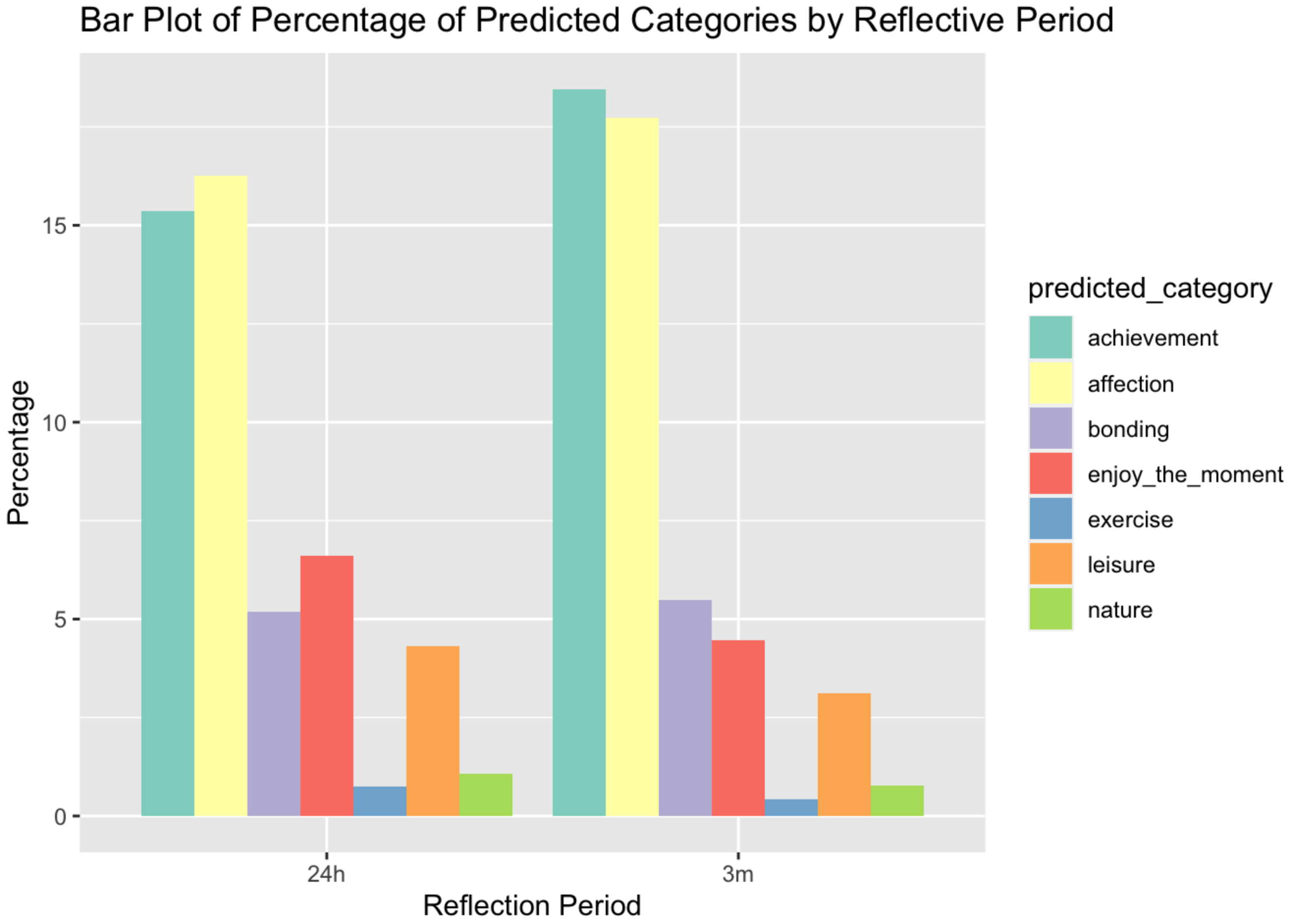
```
time = table(df$reflection_period)
pie(time, labels = paste(names(time), ": ", time), main = "Pie Chart of Reflection Perio
d", col = rainbow(length(time)))
```



```
ggplot(df, aes(x = reflection_period, fill = predicted_category)) +
  geom_bar(position = "dodge") +
  labs(title = "Bar Plot of Count of Predicted Categories by Reflective Period",
       x = "Reflection Period",
       y = "Count") +
  scale_fill_brewer(palette = "Set3")
```



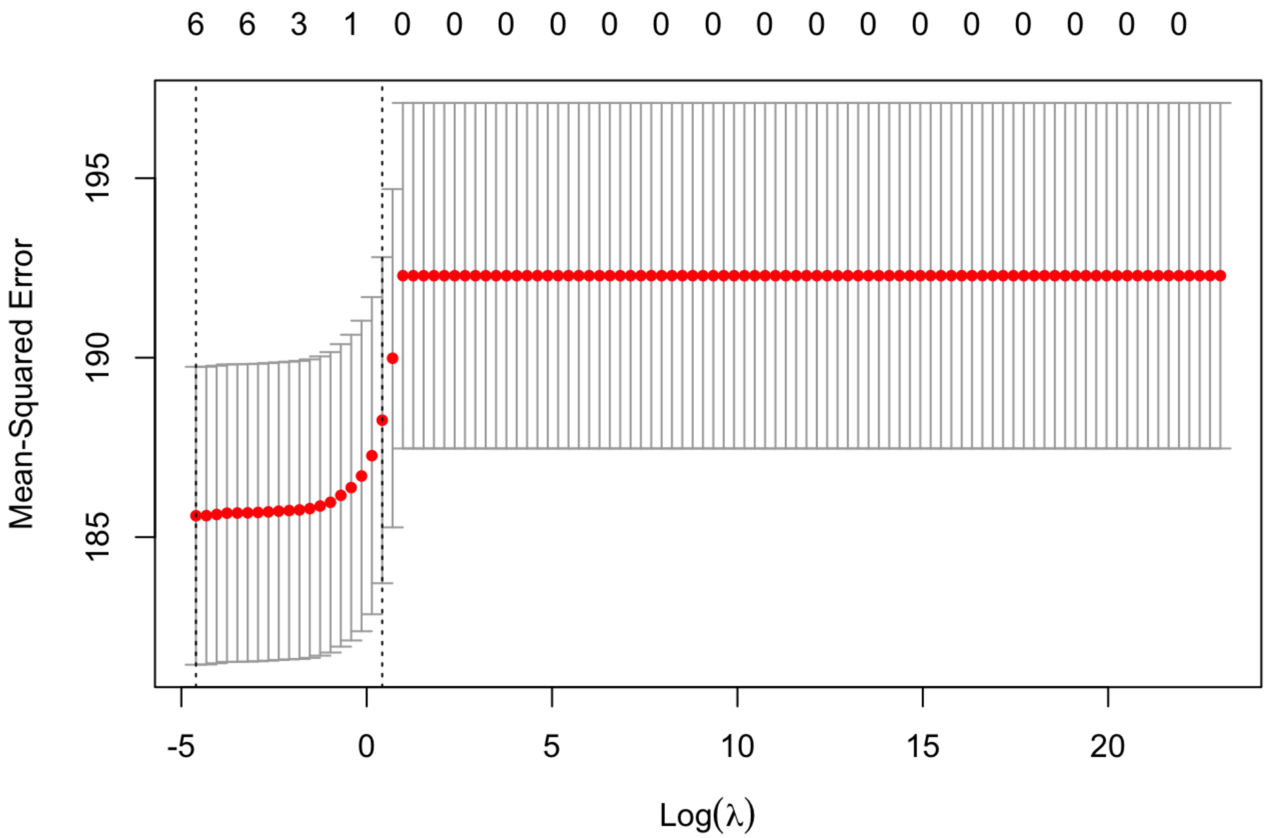
```
df_percent = df %>%
  group_by(reflection_period, predicted_category) %>%
  summarize(percent = n() / nrow(df) * 100)


ggplot(df_percent, aes(x = reflection_period, y = percent, fill = predicted_category)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Bar Plot of Percentage of Predicted Categories by Reflective Period",
       x = "Reflection Period",
       y = "Percentage") +
  scale_fill_brewer(palette = "Set3")
```



# Regression and Interpretation

Now we do some regression. Lasso regression is used to perform feature selection. It seems that losing a partner, being a parent, and being from the US had little impact to happiness. The US part is perhaps not surprising since it is by far the most common nationality. On the other hand, it seems being from India has a strong positive relationship with hapiness, or at least hapiness reported. Single people and married people seem to be happier than those that lost their partner in some way, which is perhaps not surprising. Lastly, as people get older, they seem to be happier, which is good to hear, maybe. Another way to interpret these results, however, could be how people respond to additional projects outside their scope of work. Maybe people are more interested in such a study as they age. It might not be wise to assume that people who responded the most are the happiest. To better understand that, perhaps a rating of their past 24 hours / 3 months can be used in conjunction with the current sentences to paint a more complete picture.

```
X = as.matrix(cbind(as.numeric(dep$age), dep$marital_married_binary, dep$marital_single_binary, dep$marital_lostpartner_binary, dep$parenthood_y_binary, dep$country_USA_binary, dep$country_IND_binary, dep$country_Other_binary))

y = result_sum$count

complete_rows <- complete.cases(X, y)
X <- X[complete_rows, ]
y <- y[complete_rows]

lambda_values = 10^seq(10, -2, length = 100)

lasso_cv = cv.glmnet(X, y, alpha = 1, lambda = lambda_values)

plot(lasso_cv)
```



```
optimal_lambda = lasso_cv$lambda.min


optimal_coefficients = coef(lasso_cv, s = optimal_lambda)

print(optimal_coefficients)
```

```
## 9 x 1 sparse Matrix of class "dgCMatrix"
##                      s1
## (Intercept)  5.65505851
## V1           0.02566815
## V2           2.06391350
## V3           2.56731331
## V4           .
## V5          -0.35271092
## V6           .
## V7           8.87261054
## V8          -0.59217457
```

Lastly, we specify the types of happiness and see if if there are additional pattern not observed with the full data

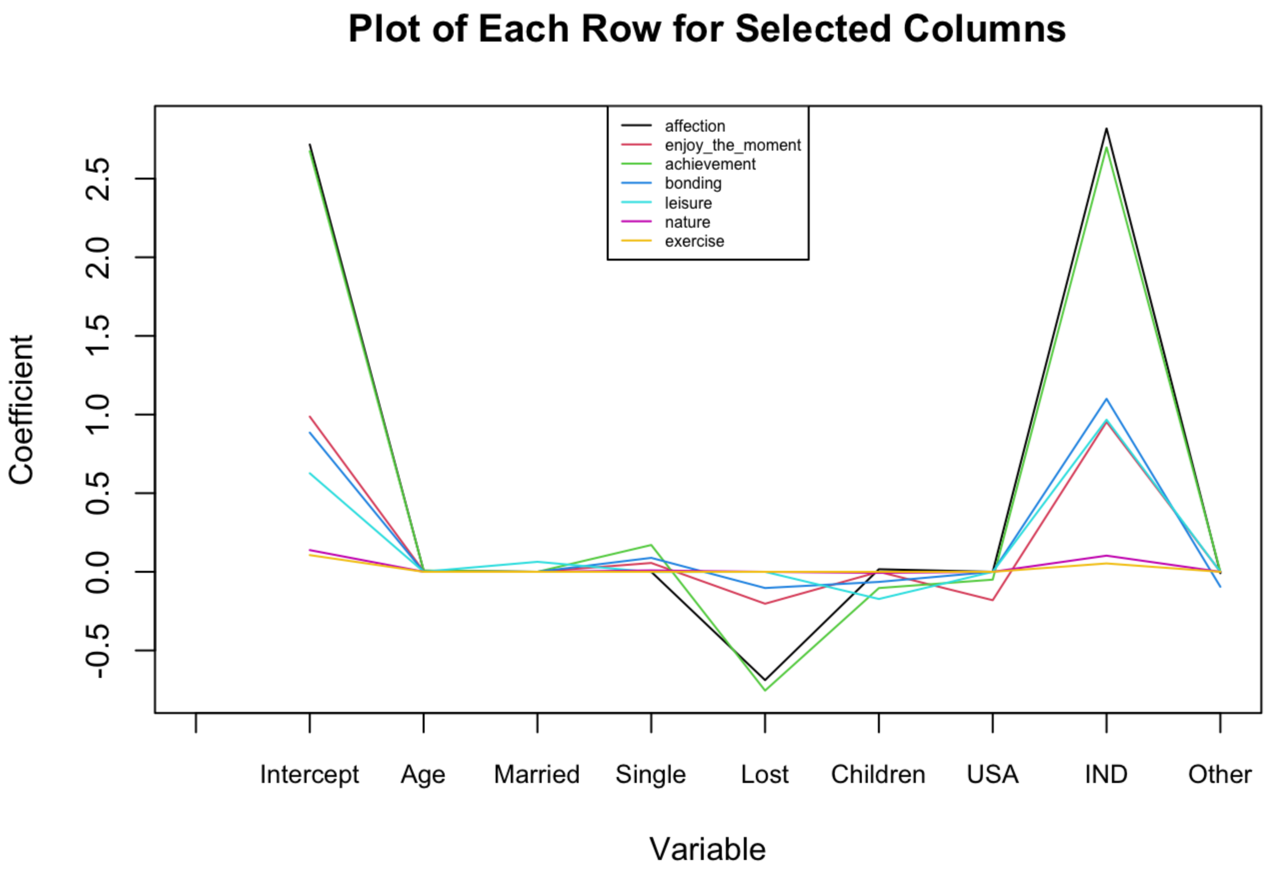
```
happy_type = unique(df$predicted_category)
output = data.frame(matrix(nrow = length(happy_type), ncol = 10))
colnames(output) <- c("Type of Happiness", "Intercept", "Age", "Married", "Single", "Los
t Spouse", "Children", "USA", "IND", "Other")
for (i in 1:length(happy_type)){
  name = happy_type[i]
  output[i,1] = name
  X = as.matrix(cbind(as.numeric(dep$age), dep$marital_married_binary, dep$marital_singl
e_binary, dep$marital_lostpartner_binary, dep$parenthood_y_binary, dep$country_USA_binar
y, dep$country_IND_binary, dep$country_Other_binary))
  ob = result_df[[name]]
  complete_rows <- complete.cases(X, ob)
  X = X[complete_rows, ]
  ob = ob[complete_rows]
  lasso_cv = cv.glmnet(X, ob, alpha = 1, lambda = lambda_values)
  optimal_lambda = lasso_cv$lambda.min
  optimal_coefficients = coef(lasso_cv, s = optimal_lambda)
  for (j in 1 : length(optimal_coefficients)){
    output[i, j+1] = optimal_coefficients[j]
  }
}
```

It seems that when broken down, each predictor's effect on happiness is generally lower. Some happiness metrics that are smaller, such as exercise, has almost no bearing with any of our predictors. We see a greater pull downwards for achievement and affection for those that lost their spouse, while the opposite effect is in place for those from IND. Elsewhere, children is generally a detractor of happiness, but it is positively correlated with affection. Single people cited achievement as their most common happiness, while married people are usually only significant when mentioning leisure. Age's effect is still there, but it is so small that I am surprised that it was not set to 0 in some cases.

```
output
```

```
##   Type of Happiness Intercept          Age    Married       Single Lost Spouse
## 1         affection 2.7169110 0.0069633175 0.00000000  0.000000000  -0.6885169
## 2  enjoy_the_moment 0.9869315 0.0029633428 0.00000000  0.056322604  -0.2028271
## 3       achievement 2.6745147 0.0083170669 0.00000000  0.170546665  -0.7551254
## 4           bonding 0.8849646 0.0000000000 0.00000000  0.088800333  -0.1030936
## 5           leisure 0.6265883 0.0006437781 0.06370302 -0.001425894   0.0000000
## 6            nature 0.1380555 0.0006397312 0.00000000  0.008844456   0.0000000
## 7          exercise 0.1063329 0.0000000000 0.00000000  0.000000000   0.0000000
##       Children         USA        IND        Other
## 1  0.016270900  0.00000000 2.81893239 -0.008733998
## 2  0.000000000 -0.18048430 0.95305582  0.000000000
## 3 -0.103258797 -0.04963065 2.69851471  0.000000000
## 4 -0.064175959  0.00000000 1.09999386 -0.095241316
## 5 -0.172722924  0.00000000 0.96708032  0.000000000
## 6 -0.006680096  0.00000000 0.10268793  0.000000000
## 7  0.000000000  0.00000000 0.05331162  0.000000000
```

```
graph = t(output)
custom_labels = c("", "Intercept", "Age", "Married", "Single", "Lost", "Children", "US
A", "IND", "Other")
#matplot(graph, type = "l", lty = 1, col = 1:length(happy_type), xlab = "Predictor", yla
b = "Value",
#        main = "Plot of Each Row for Selected Columns")
matplot(graph, type = "l", lty = 1, col = 1:nrow(graph), xlab = "Variable", ylab = "Coef
ficient",
        main = "Plot of Each Row for Selected Columns", xaxt = "n")
axis(1, at = 1:length(custom_labels), labels = custom_labels, cex.axis = 0.8)
legend("top", legend = happy_type, col = 1:length(happy_type), lty = 1, cex = 0.5)
```


Plot of Each Row for Selected Columns
affection
enjoy_the_moment
achievement
bonding
leisure
nature
exercise
Coefficient
2.5
2.0
1.5
1.0
0.5
0.0
-0.5
Intercept
Age
Married
Single
Lost
Children
USA
IND
Other
Variable