A4_youth_risk_behaviour_survey

A4_youth_risk_behaviour_survey

library packages

```
options(warn = -1)
library(dplyr)

##
## 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

library(xgboost)

##
## Attaching package: 'xgboost'

## The following object is masked from 'package:dplyr':
##
##     slice

library(tidyverse)

## -- Attaching packages ---------------------------------------------
------------- tidyverse 1.3.0 --

## √ ggplot2 3.3.0     √ purrr   0.3.4
## √ tibble  3.0.1     √ stringr 1.4.0
## √ tidyr   1.1.0     √ forcats 0.5.0
## √ readr   1.3.1

## -- Conflicts ------------------------------------------------------
------- tidyverse_conflicts() --
## x dplyr::filter()  masks stats::filter()
## x dplyr::lag()     masks stats::lag()
## x xgboost::slice() masks dplyr::slice()

library(ggplot2)
library(naniar)
library(visdat)
```

load data

```
load("./datasets/yrbs-1.rda")
nrow(x)
```

```
## [1] 15624
```

```
length(r)
```

```
## [1] 15624
```

```
#ensure that r and x have same length
```

Task 1: Build a classifier to predict labels `r` from `x` with xgboost, and show the confusion matrix

clean data, clean NAs from ethnicity not elsewhere

```
x.df = data.frame(x)
xr.df <- x.df %>% mutate(r = r)
summary(xr.df)
```

```
##        q6              q7               q8              q9
##  Min.   :1.000   Min.   : 33.00   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.000   1st Qu.: 55.00   1st Qu.:1.000   1st Qu.:4.000
##  Median :1.000   Median : 63.00   Median :2.000   Median :5.000
##  Mean   :1.002   Mean   : 67.27   Mean   :2.038   Mean   :4.373
##  3rd Qu.:1.000   3rd Qu.: 75.00   3rd Qu.:2.000   3rd Qu.:5.000
##  Max.   :2.000   Max.   :180.00   Max.   :6.000   Max.   :5.000
##  NA's   :1266    NA's   :1266     NA's   :2462    NA's   :1554
##       q10             q11             q12             q13
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :1.000   Median :2.000   Median :2.000   Median :1.000
##  Mean   :1.452   Mean   :1.713   Mean   :2.344   Mean   :1.503
##  3rd Qu.:1.000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:1.000
##  Max.   :5.000   Max.   :6.000   Max.   :8.000   Max.   :5.000
##  NA's   :69      NA's   :1709    NA's   :967     NA's   :1201
##       q14             q15             q16             q17
 q18
##  Min.   :1.000   Min.   :1.000   Min.   :1.00   Min.   :1.000   Min.
   :1.000
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.00   1st Qu.:1.000   1st
Qu.:1.000
##  Median :1.000   Median :1.000   Median :1.00   Median :1.000   Medi
an :1.000
##  Mean   :1.128   Mean   :1.125   Mean   :1.12   Mean   :1.163   Mean
   :1.483
##  3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.00   3rd Qu.:1.000   3rd
Qu.:1.000
##  Max.   :5.000   Max.   :5.000   Max.   :5.00   Max.   :8.000   Max.
   :8.000
```

```
##  NA's   :2361    NA's   :156     NA's   :61     NA's   :631     NA's
   :2500
##       q19             q20             q21             q22
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:1.000
##  Median :1.000   Median :1.000   Median :2.000   Median :2.000
##  Mean   :1.048   Mean   :1.157   Mean   :1.925   Mean   :1.842
##  3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:2.000   3rd Qu.:2.000
##  Max.   :5.000   Max.   :8.000   Max.   :2.000   Max.   :6.000
##  NA's   :2284    NA's   :292     NA's   :728     NA's   :540
##       q23             q24             q25             q26
 q27
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.00   Min.
   :1.000
##  1st Qu.:1.000   1st Qu.:2.000   1st Qu.:2.000   1st Qu.:1.00   1st
Qu.:2.000
##  Median :2.000   Median :2.000   Median :2.000   Median :2.00   Medi
an :2.000
##  Mean   :1.851   Mean   :1.809   Mean   :1.853   Mean   :1.69   Mean
   :1.818
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:2.00   3rd
Qu.:2.000
##  Max.   :6.000   Max.   :2.000   Max.   :2.000   Max.   :2.00   Max.
   :2.000
##  NA's   :1075    NA's   :176     NA's   :159     NA's   :169    NA's
   :190
##       q28             q29             q30             q31
 q32
##  Min.   :1.000   Min.   :1.000   Min.   :1.00   Min.   :1.000   Min.
   :1.000
##  1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.00   1st Qu.:1.000   1st
Qu.:1.000
##  Median :2.000   Median :1.000   Median :1.00   Median :2.000   Medi
an :1.000
##  Mean   :1.846   Mean   :1.167   Mean   :1.16   Mean   :1.663   Mean
   :1.985
##  3rd Qu.:2.000   3rd Qu.:1.000   3rd Qu.:1.00   3rd Qu.:2.000   3rd
Qu.:2.000
##  Max.   :2.000   Max.   :5.000   Max.   :3.00   Max.   :2.000   Max.
   :7.000
##  NA's   :483     NA's   :3057    NA's   :3293   NA's   :1825    NA's
   :805
##       q33             q34             q35             q36
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :1.000   Median :1.000   Median :1.000   Median :1.000
##  Mean   :1.337   Mean   :1.277   Mean   :1.388   Mean   :1.217
##  3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.000
##  Max.   :7.000   Max.   :7.000   Max.   :8.000   Max.   :3.000
##  NA's   :635     NA's   :827     NA's   :1198    NA's   :2336
```

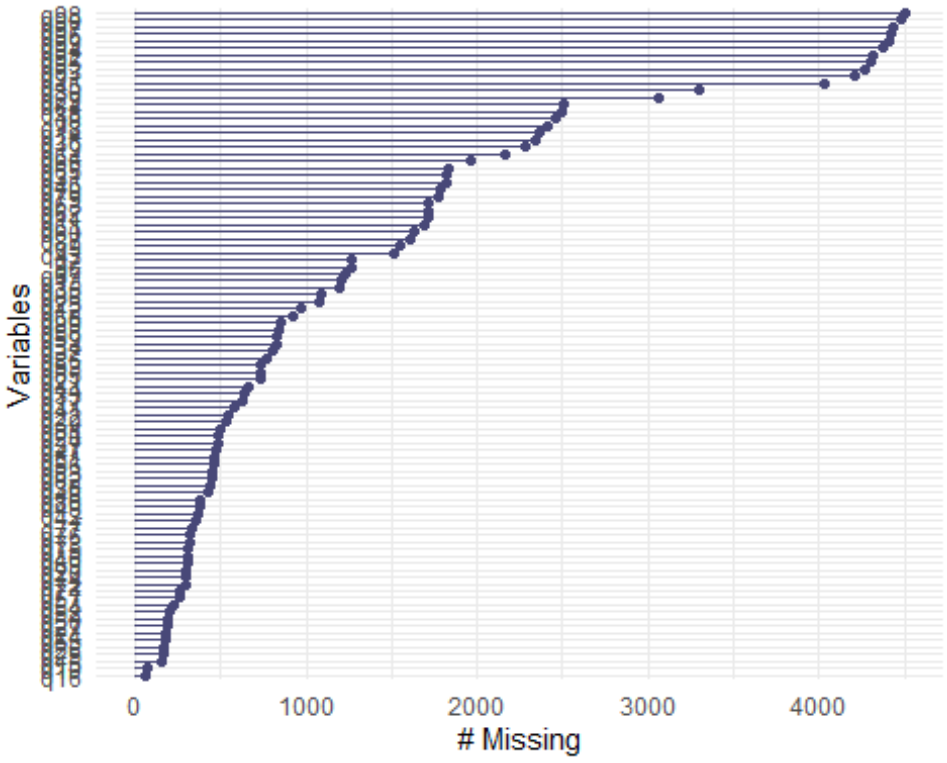
```
##       q37             q38             q39            q40
 q41
##  Min.   :1.000   Min.   :1.000   Min.   :1.00   Min.   :1.000   Min.
   :1.000
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.00   1st Qu.:1.000   1st
Qu.:1.000
##  Median :1.000   Median :1.000   Median :2.00   Median :1.000   Medi
an :2.000
##  Mean   :1.243   Mean   :1.241   Mean   :1.53   Mean   :1.604   Mean
   :2.778
##  3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:2.00   3rd Qu.:2.000   3rd
Qu.:4.000
##  Max.   :7.000   Max.   :7.000   Max.   :2.00   Max.   :7.000   Max.
   :7.000
##  NA's   :478     NA's   :439     NA's   :533    NA's   :306     NA's
   :575
##       q42            q43             q44             q45
 q46
##  Min.   :1.00   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.
   :1.000
##  1st Qu.:1.00   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000   1st
Qu.:1.000
##  Median :4.00   Median :1.000   Median :1.000   Median :1.000   Medi
an :1.000
##  Mean   :3.42   Mean   :1.625   Mean   :1.405   Mean   :2.084   Mean
   :2.677
##  3rd Qu.:5.00   3rd Qu.:2.000   3rd Qu.:1.000   3rd Qu.:2.000   3rd
Qu.:5.000
##  Max.   :7.00   Max.   :7.000   Max.   :7.000   Max.   :8.000   Max.
   :8.000
##  NA's   :367    NA's   :1510    NA's   :657     NA's   :4027    NA's
   :1819
##       q47             q48             q49             q50
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :1.000   Median :1.000   Median :1.000   Median :1.000
##  Mean   :2.366   Mean   :2.661   Mean   :1.577   Mean   :1.135
##  3rd Qu.:3.000   3rd Qu.:5.000   3rd Qu.:1.000   3rd Qu.:1.000
##  Max.   :7.000   Max.   :7.000   Max.   :6.000   Max.   :6.000
##  NA's   :467     NA's   :426     NA's   :374     NA's   :192
##       q51             q52             q53             q54
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :1.000   Median :1.000   Median :1.000   Median :1.000
##  Mean   :1.154   Mean   :1.069   Mean   :1.089   Mean   :1.117
##  3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.000
##  Max.   :6.000   Max.   :6.000   Max.   :6.000   Max.   :6.000
##  NA's   :466     NA's   :200     NA's   :735     NA's   :227
##       q55             q56             q57             q58
 q59
```

```
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.00   Min.
   :1.000
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.00   1st
Qu.:2.000
##  Median :1.000   Median :1.000   Median :1.000   Median :1.00   Medi
an :2.000
##  Mean   :1.202   Mean   :1.098   Mean   :1.404   Mean   :1.03   Mean
   :1.764
##  3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.000   3rd Qu.:1.00   3rd
Qu.:2.000
##  Max.   :6.000   Max.   :6.000   Max.   :6.000   Max.   :3.00   Max.
   :2.000
##  NA's   :174     NA's   :459     NA's   :257     NA's   :838    NA's
   :828
##       q60             q61             q62             q63
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :2.000   Median :1.000   Median :1.000   Median :1.000
##  Mean   :1.574   Mean   :2.935   Mean   :2.105   Mean   :1.883
##  3rd Qu.:2.000   3rd Qu.:5.000   3rd Qu.:3.000   3rd Qu.:3.000
##  Max.   :2.000   Max.   :8.000   Max.   :7.000   Max.   :8.000
##  NA's   :1636    NA's   :1692    NA's   :1714    NA's   :1714
##       q64             q65             q66             q67
  q68
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Mi
n.   :1.00
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000   1st
 Qu.:1.00
##  Median :1.000   Median :1.000   Median :1.000   Median :2.000   Med
ian :1.00
##  Mean   :1.766   Mean   :1.592   Mean   :2.287   Mean   :1.875   Mea
n   :1.25
##  3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:4.000   3rd Qu.:3.000   3rd
 Qu.:1.00
##  Max.   :3.000   Max.   :3.000   Max.   :8.000   Max.   :4.000   Ma
x.   :4.00
##  NA's   :2167    NA's   :1833    NA's   :1967    NA's   :1229    NA'
s   :921
##       q69             q70             q71             q72
  q73
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Mi
n.   :1.00
##  1st Qu.:3.000   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:2.000   1st
 Qu.:1.00
##  Median :3.000   Median :2.000   Median :2.000   Median :3.000   Med
ian :2.00
##  Mean   :3.203   Mean   :2.031   Mean   :2.667   Mean   :3.185   Mea
n   :2.03
##  3rd Qu.:4.000   3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:4.000   3rd
 Qu.:2.00
```

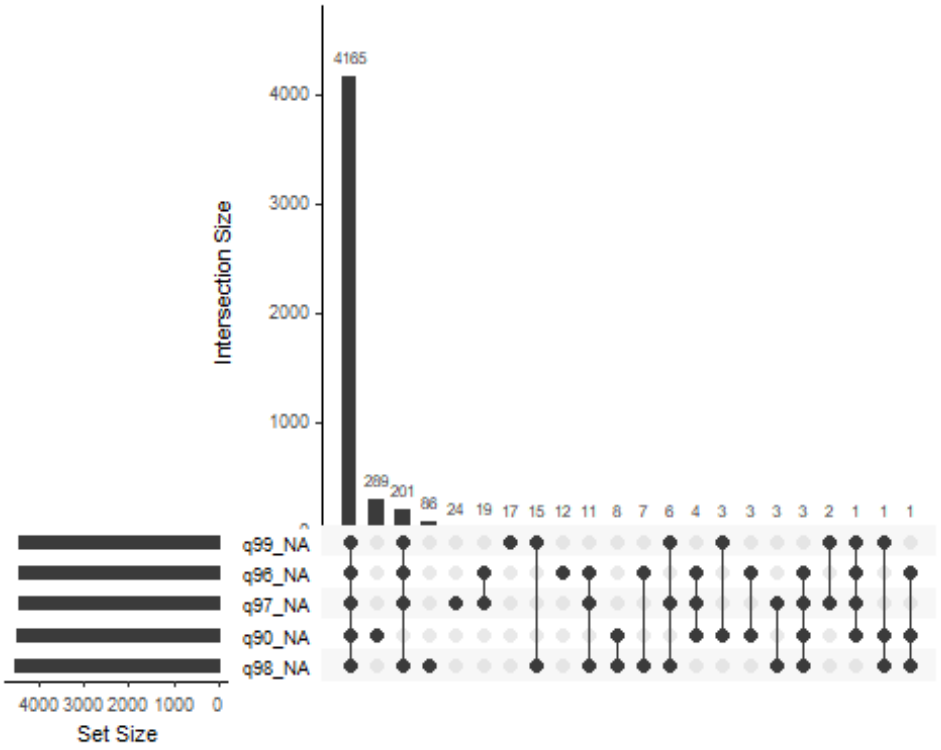
```
##  Max.   :5.000   Max.   :4.000   Max.   :7.000   Max.   :7.000   Ma
x.   :7.00
##  NA's   :304     NA's   :1780    NA's   :263     NA's   :290     NA'
s   :315
##       q74             q75             q76             q77
 q78
##  Min.   :1.000   Min.   :1.000   Min.   :1.00   Min.   :1.000   Min.
   :1.000
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:2.00   1st Qu.:2.000   1st
Qu.:2.000
##  Median :2.000   Median :1.000   Median :2.00   Median :2.000   Medi
an :3.000
##  Mean   :2.041   Mean   :1.773   Mean   :2.64   Mean   :2.656   Mean
   :3.042
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:3.00   3rd Qu.:3.000   3rd
Qu.:4.000
##  Max.   :7.000   Max.   :7.000   Max.   :7.00   Max.   :7.000   Max.
   :7.000
##  NA's   :291     NA's   :310     NA's   :319    NA's   :332     NA's
   :2409
##       q79             q80             q81             q82
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:3.000   1st Qu.:3.000   1st Qu.:2.000   1st Qu.:2.000
##  Median :5.000   Median :5.000   Median :3.000   Median :4.000
##  Mean   :5.125   Mean   :4.897   Mean   :3.358   Mean   :4.017
##  3rd Qu.:8.000   3rd Qu.:8.000   3rd Qu.:5.000   3rd Qu.:6.000
##  Max.   :8.000   Max.   :8.000   Max.   :7.000   Max.   :7.000
##  NA's   :1768    NA's   :379     NA's   :500     NA's   :446
##       q83             q84             q85             q86
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:1.000
##  Median :2.000   Median :2.000   Median :2.000   Median :1.000
##  Mean   :3.129   Mean   :1.932   Mean   :2.006   Mean   :1.624
##  3rd Qu.:6.000   3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:2.000
##  Max.   :6.000   Max.   :4.000   Max.   :3.000   Max.   :5.000
##  NA's   :452     NA's   :2502    NA's   :728     NA's   :769
##       q87             q88             q89             q90
  q91
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Mi
n.   :1.00
##  1st Qu.:2.000   1st Qu.:3.000   1st Qu.:1.000   1st Qu.:1.000   1st
 Qu.:1.00
##  Median :2.000   Median :4.000   Median :2.000   Median :1.000   Med
ian :1.00
##  Mean   :1.816   Mean   :3.656   Mean   :2.215   Mean   :1.289   Mea
n   :1.13
##  3rd Qu.:2.000   3rd Qu.:5.000   3rd Qu.:3.000   3rd Qu.:1.000   3rd
 Qu.:1.00
##  Max.   :3.000   Max.   :7.000   Max.   :7.000   Max.   :6.000   Ma
x.   :6.00
```

```
##  NA's   :1612    NA's   :1090    NA's   :855     NA's   :4478    NA'
s   :4212
##       q92             q93             q94             q95
 q96
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.00   Min.
   :1.000
##  1st Qu.:1.000   1st Qu.:3.000   1st Qu.:2.000   1st Qu.:1.00   1st
Qu.:1.000
##  Median :2.000   Median :5.000   Median :2.000   Median :4.00   Medi
an :1.000
##  Mean   :2.209   Mean   :5.009   Mean   :1.951   Mean   :3.95   Mean
   :1.169
##  3rd Qu.:3.000   3rd Qu.:7.000   3rd Qu.:2.000   3rd Qu.:6.00   3rd
Qu.:1.000
##  Max.   :7.000   Max.   :7.000   Max.   :3.000   Max.   :8.00   Max.
   :6.000
##  NA's   :4319    NA's   :4274    NA's   :4375    NA's   :4304   NA's
   :4427
##       q97             q98             q99              r
##  Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :0.00
##  1st Qu.:1.000   1st Qu.:1.000   1st Qu.:1.000   1st Qu.:4.00
##  Median :2.000   Median :2.000   Median :1.000   Median :4.00
##  Mean   :2.319   Mean   :1.685   Mean   :1.202   Mean   :4.27
##  3rd Qu.:3.000   3rd Qu.:2.000   3rd Qu.:1.000   3rd Qu.:5.00
##  Max.   :6.000   Max.   :2.000   Max.   :4.000   Max.   :7.00
##  NA's   :4439    NA's   :4507    NA's   :4411    NA's   :358
```

```
gg_miss_var(xr.df)
```




```
gg_miss_upset(xr.df)
```




```
as_shadow(xr.df)
```

```
## # A tibble: 15,624 x 95
##    q6_NA q7_NA q8_NA q9_NA q10_NA q11_NA q12_NA q13_NA q14_NA q15_NA
 q16_NA
##    <fct> <fct> <fct> <fct> <fct>  <fct>  <fct>  <fct>  <fct>  <fct>
 <fct>
##  1 !NA   !NA   !NA   !NA   !NA    !NA    !NA    !NA    !NA    !NA
 !NA
##  2 !NA   !NA   !NA   !NA   !NA    !NA    !NA    !NA    !NA    !NA
 !NA
##  3 !NA   !NA   !NA   !NA   !NA    !NA    !NA    !NA    !NA    !NA
 !NA
##  4 NA    NA    !NA   !NA   !NA    !NA    !NA    !NA    !NA    !NA
 !NA
##  5 !NA   !NA   !NA   !NA   !NA    !NA    !NA    !NA    !NA    !NA
 !NA
##  6 !NA   !NA   !NA   !NA   !NA    !NA    !NA    !NA    !NA    !NA
 !NA
##  7 !NA   !NA   !NA   !NA   !NA    !NA    !NA    !NA    !NA    !NA
 !NA
##  8 !NA   !NA   NA    !NA   !NA    NA     NA     !NA    !NA    !NA
 !NA
##  9 !NA   !NA   !NA   !NA   !NA    !NA    !NA    !NA    !NA    !NA
 !NA
## 10 !NA   !NA   !NA   !NA   !NA    !NA    !NA    !NA    !NA    !NA
 !NA
## # ... with 15,614 more rows, and 84 more variables: q17_NA <fct>, q1
8_NA <fct>,
## #   q19_NA <fct>, q20_NA <fct>, q21_NA <fct>, q22_NA <fct>, q23_NA <
fct>,
## #   q24_NA <fct>, q25_NA <fct>, q26_NA <fct>, q27_NA <fct>, q28_NA <
fct>,
## #   q29_NA <fct>, q30_NA <fct>, q31_NA <fct>, q32_NA <fct>, q33_NA <
fct>,
## #   q34_NA <fct>, q35_NA <fct>, q36_NA <fct>, q37_NA <fct>, q38_NA <
fct>,
## #   q39_NA <fct>, q40_NA <fct>, q41_NA <fct>, q42_NA <fct>, q43_NA <
fct>,
## #   q44_NA <fct>, q45_NA <fct>, q46_NA <fct>, q47_NA <fct>, q48_NA <
fct>,
## #   q49_NA <fct>, q50_NA <fct>, q51_NA <fct>, q52_NA <fct>, q53_NA <
fct>,
## #   q54_NA <fct>, q55_NA <fct>, q56_NA <fct>, q57_NA <fct>, q58_NA <
fct>,
## #   q59_NA <fct>, q60_NA <fct>, q61_NA <fct>, q62_NA <fct>, q63_NA <
fct>,
## #   q64_NA <fct>, q65_NA <fct>, q66_NA <fct>, q67_NA <fct>, q68_NA <
fct>,
## #   q69_NA <fct>, q70_NA <fct>, q71_NA <fct>, q72_NA <fct>, q73_NA <
fct>,
## #   q74_NA <fct>, q75_NA <fct>, q76_NA <fct>, q77_NA <fct>, q78_NA <
```

```
fct>,
## #   q79_NA <fct>, q80_NA <fct>, q81_NA <fct>, q82_NA <fct>, q83_NA <
fct>,
## #   q84_NA <fct>, q85_NA <fct>, q86_NA <fct>, q87_NA <fct>, q88_NA <
fct>,
## #   q89_NA <fct>, q90_NA <fct>, q91_NA <fct>, q92_NA <fct>, q93_NA <
fct>,
## #   q94_NA <fct>, q95_NA <fct>, q96_NA <fct>, q97_NA <fct>, q98_NA <
fct>,
## #   q99_NA <fct>, r_NA <fct>

(aq_shadow <- bind_shadow(xr.df))

## # A tibble: 15,624 x 190
##       q6    q7    q8    q9   q10   q11   q12   q13   q14   q15   q16
   q17   q18
##    <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
 <int> <int>
##  1     1    54     2     5     5     1     1     1     1     1     1
     2     3
##  2     1    51     2     5     1     1     1     1     1     1     1
     1     1
##  3     1    66     1     5     1     1     1     1     1     1     3
    NA    NA
##  4    NA    NA     2     5     1     2     4     4     1     1     1
     1     2
##  5     1    68     1     4     3     1     1     1     1     1     1
     1     1
##  6     1    59     2     5     1     1     1     1     1     1     1
     1     1
##  7     1    70     2     5     1     2     8     1     1     1     1
     1     1
##  8     1    90    NA     5     1    NA    NA     1     1     1     1
     2     1
##  9     1    40     2     4     2     1     1     3     1     1     1
     1     1
## 10     1    49     1     5     1     1     1     1     1     1     1
     1     1
## # ... with 15,614 more rows, and 177 more variables: q19 <int>, q20
<int>,
## #   q21 <int>, q22 <int>, q23 <int>, q24 <int>, q25 <int>, q26 <in
t>,
## #   q27 <int>, q28 <int>, q29 <int>, q30 <int>, q31 <int>, q32 <in
t>,
## #   q33 <int>, q34 <int>, q35 <int>, q36 <int>, q37 <int>, q38 <in
t>,
## #   q39 <int>, q40 <int>, q41 <int>, q42 <int>, q43 <int>, q44 <in
t>,
## #   q45 <int>, q46 <int>, q47 <int>, q48 <int>, q49 <int>, q50 <in
t>,
```

```
## #   q51 <int>, q52 <int>, q53 <int>, q54 <int>, q55 <int>, q56 <in
t>,
## #   q57 <int>, q58 <int>, q59 <int>, q60 <int>, q61 <int>, q62 <in
t>,
## #   q63 <int>, q64 <int>, q65 <int>, q66 <int>, q67 <int>, q68 <in
t>,
## #   q69 <int>, q70 <int>, q71 <int>, q72 <int>, q73 <int>, q74 <in
t>,
## #   q75 <int>, q76 <int>, q77 <int>, q78 <int>, q79 <int>, q80 <in
t>,
## #   q81 <int>, q82 <int>, q83 <int>, q84 <int>, q85 <int>, q86 <in
t>,
## #   q87 <int>, q88 <int>, q89 <int>, q90 <int>, q91 <int>, q92 <in
t>,
## #   q93 <int>, q94 <int>, q95 <int>, q96 <int>, q97 <int>, q98 <in
t>,
## #   q99 <int>, r <dbl>, q6_NA <fct>, q7_NA <fct>, q8_NA <fct>, q9_NA
 <fct>,
## #   q10_NA <fct>, q11_NA <fct>, q12_NA <fct>, q13_NA <fct>, q14_NA <
fct>,
## #   q15_NA <fct>, q16_NA <fct>, q17_NA <fct>, q18_NA <fct>, q19_NA <
fct>,
## #   q20_NA <fct>, q21_NA <fct>, q22_NA <fct>, q23_NA <fct>, ...
```

We find that there are many missing value in the data.

An acceptable way is to clean NAs form ethnicity (r) and *xgboost* package will deal with the other NA values. The other NA values are considered as 'missing' by the algorithm of *xgboost*.

Another way is to use *missRanger* package to replace the NA values but it is not very useful in this case, and it takes so much time to get a result.

```
xr_new=subset(xr.df,r!="NA")

#library(missRanger)
#xr_imp <- xr_new %>%
  #missRanger(verbose = 1, returnOOB = TRUE)
```

Use cross validation to find best value of nrounds (and possibly eta)

```
#Data setup
train_test = sample(2,nrow(xr_new),replace = T,prob = c(0.75,0.25))
xr.train<-xr_new[train_test==1,]
xr.test<-xr_new[train_test==2,]
#cross validation, begin from nrounds=30, eta =1
xgb.cv(data=as.matrix(xr.train[,-(95)]), label=xr.train$r, missing = N
A,
       num_class=8, nrounds=40, nfold=5,eta =1, objective="multi:softma
x")
```

```
## [1]  train-merror:0.457065+0.007881  test-merror:0.500694+0.012206
## [2]  train-merror:0.400191+0.006290  test-merror:0.485492+0.008006
## [3]  train-merror:0.365058+0.006435  test-merror:0.484004+0.009255
## [4]  train-merror:0.331017+0.003057  test-merror:0.483829+0.008027
## [5]  train-merror:0.300952+0.004127  test-merror:0.483304+0.009452
## [6]  train-merror:0.274886+0.004441  test-merror:0.483214+0.011620
## [7]  train-merror:0.252469+0.005067  test-merror:0.478582+0.012828
## [8]  train-merror:0.233219+0.005004  test-merror:0.480503+0.015929
## [9]  train-merror:0.212463+0.003180  test-merror:0.481642+0.010496
## [10] train-merror:0.196273+0.003136  test-merror:0.480855+0.009802
## [11] train-merror:0.178465+0.003881  test-merror:0.482603+0.008962
## [12] train-merror:0.166164+0.002860  test-merror:0.483128+0.009807
## [13] train-merror:0.153054+0.002864  test-merror:0.481906+0.008338
## [14] train-merror:0.140448+0.002349  test-merror:0.481468+0.011240
## [15] train-merror:0.130550+0.002412  test-merror:0.480855+0.010918
## [16] train-merror:0.119865+0.004315  test-merror:0.480681+0.011260
## [17] train-merror:0.110251+0.003833  test-merror:0.482167+0.010857
## [18] train-merror:0.103172+0.001713  test-merror:0.482080+0.010881
## [19] train-merror:0.096814+0.001337  test-merror:0.481641+0.013936
## [20] train-merror:0.090675+0.001682  test-merror:0.481903+0.013766
## [21] train-merror:0.081760+0.003799  test-merror:0.484437+0.016379
## [22] train-merror:0.075052+0.002772  test-merror:0.483214+0.013254
## [23] train-merror:0.068192+0.002743  test-merror:0.482342+0.011284
## [24] train-merror:0.063101+0.003125  test-merror:0.482518+0.012449
## [25] train-merror:0.056065+0.002999  test-merror:0.484091+0.013536
## [26] train-merror:0.051149+0.002794  test-merror:0.483128+0.013512
## [27] train-merror:0.047042+0.001992  test-merror:0.482256+0.013169
## [28] train-merror:0.040836+0.001880  test-merror:0.483654+0.012554
## [29] train-merror:0.036925+0.002402  test-merror:0.481468+0.012379
## [30] train-merror:0.033713+0.002759  test-merror:0.482691+0.012217
## [31] train-merror:0.030065+0.002457  test-merror:0.482865+0.013556
## [32] train-merror:0.026525+0.002421  test-merror:0.484963+0.013391
## [33] train-merror:0.023772+0.002118  test-merror:0.485749+0.012823
## [34] train-merror:0.021062+0.002406  test-merror:0.482866+0.010960
## [35] train-merror:0.019096+0.002380  test-merror:0.482342+0.011717
## [36] train-merror:0.017326+0.002686  test-merror:0.482779+0.011251
## [37] train-merror:0.015688+0.002454  test-merror:0.483477+0.012756
## [38] train-merror:0.013699+0.002655  test-merror:0.481641+0.014991
## [39] train-merror:0.012323+0.002184  test-merror:0.481903+0.013041
## [40] train-merror:0.010553+0.002164  test-merror:0.480681+0.012272
```

```
#A bit of L2 regularisation
xgb.cv(data=as.matrix(xr.train[,-(95)]),label=xr.train$r, missing = NA,
       num_class=8, nrounds=30, nfold=5, eta =1, objective="multi:softm
ax",lambda=1)
```

```
## [1]  train-merror:0.459032+0.004410  test-merror:0.501048+0.007329
## [2]  train-merror:0.401350+0.003263  test-merror:0.490735+0.005336
## [3]  train-merror:0.363966+0.005677  test-merror:0.484267+0.007615
## [4]  train-merror:0.333551+0.005053  test-merror:0.481296+0.008289
```

```
## [5]  train-merror:0.307048+0.005015  test-merror:0.482345+0.008681
## [6]  train-merror:0.282358+0.003799  test-merror:0.482082+0.009894
## [7]  train-merror:0.260072+0.004995  test-merror:0.483743+0.007190
## [8]  train-merror:0.240058+0.005062  test-merror:0.480249+0.004832
## [9]  train-merror:0.219608+0.003213  test-merror:0.481036+0.009174
## [10] train-merror:0.202892+0.003837  test-merror:0.479811+0.008693
## [11] train-merror:0.187773+0.003951  test-merror:0.476143+0.009437
## [12] train-merror:0.171648+0.003250  test-merror:0.478152+0.006143
## [13] train-merror:0.157424+0.001771  test-merror:0.475180+0.006303
## [14] train-merror:0.144817+0.003603  test-merror:0.473170+0.008165
## [15] train-merror:0.133739+0.004601  test-merror:0.473693+0.006817
## [16] train-merror:0.123580+0.002195  test-merror:0.472907+0.007411
## [17] train-merror:0.113857+0.002988  test-merror:0.474480+0.006755
## [18] train-merror:0.104854+0.003910  test-merror:0.474742+0.007890
## [19] train-merror:0.096989+0.003750  test-merror:0.473869+0.007641
## [20] train-merror:0.089320+0.003214  test-merror:0.474567+0.007635
## [21] train-merror:0.082087+0.003522  test-merror:0.474741+0.007802
## [22] train-merror:0.075139+0.005193  test-merror:0.476752+0.007122
## [23] train-merror:0.067558+0.004117  test-merror:0.475963+0.009098
## [24] train-merror:0.061833+0.003956  test-merror:0.474564+0.009609
## [25] train-merror:0.057551+0.003765  test-merror:0.478149+0.008369
## [26] train-merror:0.050777+0.003699  test-merror:0.474827+0.007927
## [27] train-merror:0.045577+0.003714  test-merror:0.475440+0.006526
## [28] train-merror:0.041316+0.003913  test-merror:0.474217+0.006768
## [29] train-merror:0.037318+0.003525  test-merror:0.475265+0.008438
## [30] train-merror:0.033276+0.003579  test-merror:0.475614+0.010027
```

Add L_2 regularisation is not bad. It seems to plateau at about 7 rounds. Similar performance after varying the learning rate and penalty. Let's set the nrounds = 7.

Fit the full model

```
model<-xgboost(data=as.matrix(xr.train[,-(95)]),label=xr.train$r,
               num_class=8, nrounds=7,eta=1,objective="multi:softmax",m
issing = NA, lamda = 1)
```

```
## [1]  train-merror:0.468013
## [2]  train-merror:0.414700
## [3]  train-merror:0.381926
## [4]  train-merror:0.347579
## [5]  train-merror:0.324856
## [6]  train-merror:0.300909
## [7]  train-merror:0.280808
```

Confusion matrix.

```
res = table(predict(model,newdata=as.matrix(xr.test[,-(95)]))==xr.test
$r)
res
```

```
##
## FALSE  TRUE
##  1838  1986

accuracy_T=res[2]/(res[1]+res[2])
accuracy_T

##      TRUE
## 0.5193515

table(predict(model,newdata=as.matrix(xr.test[,-(95)])),xr.test$r, dnn=
c("actual","predict"))

##       predict
## actual    0    1    2    3    4    5    6    7
##      0    0    0    4    0    1    0    3    0
##      1    1   30    8    2   12   14   15    5
##      2    2    8  163    5   73   66   66   23
##      3    0    0    0    0    0    1    1    0
##      4   22   53   89    7 1404  173  253  102
##      5    5   23   54    5   90  210  161   20
##      6   11   25   72    5  131  150  176   39
##      7    1    0    4    0   11    6   16    3
```

The confusion matrix is not very good. The accuracy is only about 50%.

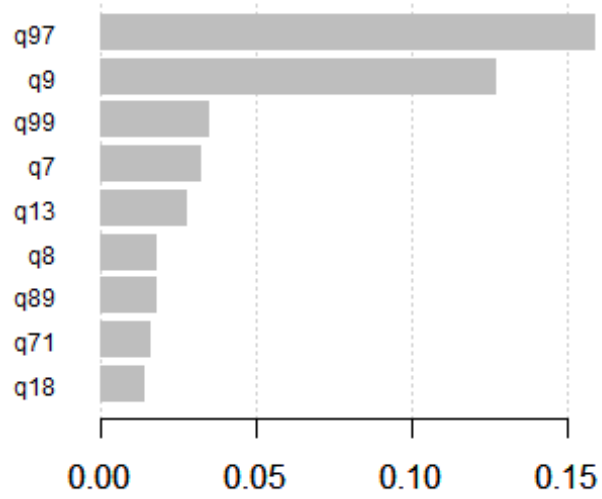
Task 2: Describe and visualise which variables are most important in the prediction.

Variables importance.

```
xgb.importance(model=model)

##     Feature         Gain        Cover   Frequency
##  1:     q97 0.1593773476 0.0537374129 0.017345704
##  2:      q9 0.1275726702 0.0244681553 0.018959258
##  3:     q99 0.0354471107 0.0188914437 0.007664381
##  4:      q7 0.0324945146 0.0575754336 0.058894716
##  5:     q13 0.0277776422 0.0221218271 0.012908431
##  6:      q8 0.0180503925 0.0198850702 0.015328762
##  7:     q89 0.0180313512 0.0370039427 0.025010085
##  8:     q71 0.0162735192 0.0224293487 0.020169423
##  9:     q18 0.0145867011 0.0137227617 0.013715208
## 10:     q78 0.0145311174 0.0248574566 0.018152481
## 11:     q81 0.0143338864 0.0162137651 0.019766035
## 12:     q82 0.0141473179 0.0090303079 0.020976200
## 13:     q76 0.0133646309 0.0274748937 0.019766035
## 14:     q84 0.0125917580 0.0093756474 0.018152481
## 15:     q37 0.0122740342 0.0167341979 0.008067769
## 16:     q92 0.0119415780 0.0155992569 0.013715208
## 17:     q83 0.0112000732 0.0127434362 0.016538927
```

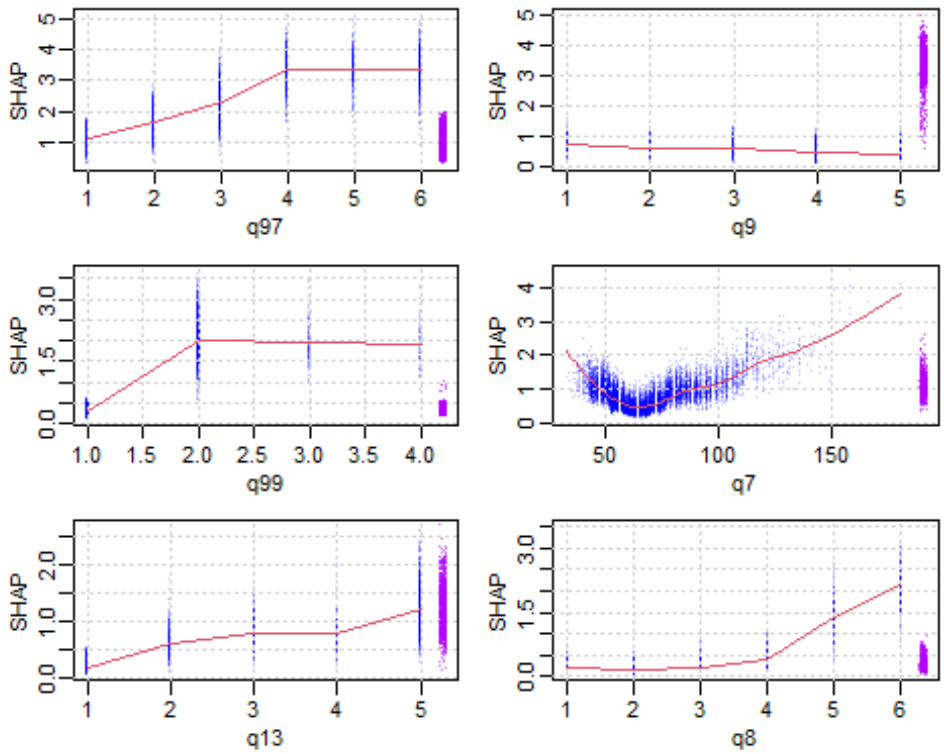
```
## 18:     q79 0.0111144759 0.0054585617 0.016942315
## 19:     q70 0.0111143403 0.0160180061 0.012505042
## 20:     q72 0.0109613540 0.0250122277 0.018152481
## 21:     q14 0.0109603832 0.0096156988 0.005244050
## 22:     q42 0.0109179423 0.0156303655 0.019766035
## 23:     q75 0.0108911393 0.0063940164 0.012908431
## 24:     q48 0.0105543751 0.0145492615 0.012101654
## 25:     q88 0.0105353524 0.0148871255 0.019362646
## 26:     q80 0.0101687516 0.0098601250 0.020169423
## 27:     q41 0.0100354120 0.0112503826 0.018959258
## 28:     q77 0.0099989431 0.0119292895 0.017749092
## 29:     q12 0.0099780537 0.0156983287 0.014118596
## 30:     q11 0.0097780479 0.0124507461 0.011698265
## 31:     q74 0.0096728470 0.0230591332 0.018152481
## 32:     q45 0.0092491972 0.0096112277 0.010084712
## 33:     q28 0.0091727043 0.0071357361 0.005647438
## 34:     q93 0.0091285303 0.0055409583 0.016538927
## 35:     q86 0.0090027676 0.0122009990 0.013311819
## 36:     q36 0.0088897929 0.0022185843 0.009277935
## 37:     q69 0.0084789609 0.0168320574 0.013715208
## 38:     q10 0.0083351457 0.0139423691 0.010891489
## 39:     q20 0.0077960750 0.0047424841 0.008471158
## 40:     q47 0.0077287436 0.0031104967 0.012101654
## 41:     q32 0.0076921116 0.0044746427 0.014521985
## 42:     q62 0.0075190133 0.0089928031 0.006857604
## 43:     q22 0.0073337050 0.0137650643 0.012505042
## 44:     q40 0.0071602362 0.0096085547 0.012101654
## 45:     q51 0.0071284422 0.0049608670 0.006050827
## 46:     q33 0.0068006893 0.0079475244 0.003227108
## 47:     q95 0.0067122218 0.0052320381 0.014925373
## 48:     q39 0.0066588138 0.0094554639 0.010084712
## 49:     q61 0.0064551418 0.0044935635 0.010891489
## 50:     q30 0.0063446429 0.0059542540 0.005647438
## 51:     q38 0.0063346114 0.0070277234 0.008471158
## 52:     q46 0.0058350546 0.0122630542 0.011698265
## 53:     q23 0.0058131869 0.0043714463 0.011294877
## 54:     q25 0.0053063316 0.0147126818 0.005244050
## 55:     q94 0.0051789317 0.0052163745 0.009681323
## 56:     q53 0.0050955162 0.0040592740 0.004033885
## 57:     q87 0.0050841576 0.0079702844 0.008874546
## 58:     q29 0.0050676634 0.0060295920 0.007664381
## 59:     q59 0.0050058131 0.0106982360 0.006857604
## 60:     q26 0.0048933157 0.0030653702 0.008067769
## 61:     q85 0.0048140230 0.0069850308 0.009277935
## 62:     q90 0.0046725966 0.0091943643 0.008067769
## 63:     q66 0.0046193664 0.0034528360 0.008874546
## 64:     q67 0.0042177806 0.0039267152 0.009277935
## 65:     q96 0.0041483300 0.0171951650 0.006050827
## 66:     q21 0.0040594156 0.0038752564 0.007260992
## 67:     q35 0.0038823339 0.0053200482 0.005647438
```

```
## 68:     q91 0.0038143057 0.0169172159 0.006454215
## 69:     q73 0.0037525340 0.0079464927 0.007664381
## 70:     q34 0.0035925401 0.0089662655 0.005647438
## 71:     q63 0.0034233024 0.0037715746 0.006857604
## 72:     q60 0.0033926378 0.0014661558 0.004033885
## 73:     q24 0.0032839198 0.0110006677 0.005647438
## 74:     q49 0.0032198219 0.0040993243 0.007260992
## 75:     q57 0.0032182959 0.0060315044 0.008471158
## 76:     q17 0.0031873795 0.0072049777 0.005244050
## 77:     q50 0.0031755707 0.0019484427 0.004437273
## 78:      q6 0.0031632218 0.0009170851 0.003227108
## 79:     q64 0.0031539713 0.0059573555 0.004437273
## 80:     q15 0.0030107208 0.0055569700 0.007664381
## 81:     q98 0.0028149868 0.0008306501 0.003227108
## 82:     q44 0.0026557851 0.0056275990 0.004033885
## 83:     q16 0.0025815535 0.0031135408 0.005244050
## 84:     q55 0.0025494586 0.0108251305 0.005647438
## 85:     q68 0.0022516334 0.0048542077 0.005244050
## 86:     q65 0.0021127900 0.0039393160 0.004437273
## 87:     q31 0.0020125602 0.0021013140 0.002823719
## 88:     q43 0.0018134817 0.0005879708 0.003630496
## 89:     q56 0.0017054895 0.0041603512 0.002016942
## 90:     q27 0.0016171499 0.0026034831 0.003227108
## 91:     q58 0.0008259720 0.0015994481 0.001210165
## 92:     q19 0.0005643353 0.0069618302 0.003227108
## 93:     q54 0.0005573254 0.0007702576 0.001613554
## 94:     q52 0.0002568302 0.0009801019 0.001210165
##     Feature         Gain        Cover   Frequency

names <- dimnames(data.matrix(xr.train[,c(1:94)]))[[2]]
importance_matrix <- xgb.importance(names,model=model)
xgb.plot.importance(importance_matrix[1:9,])
```



q97 is the most important variable, then the q7 and q9. The top six variables' importance seems significant. Let's draw SHAP plot for these 6 variables.

```
xgb.plot.shap(model=model, data=as.matrix(xr_new[,1:94]),top_n=6,n_col=
2)
```



Obviously, q97 is the most important variable.

Task 3: Describe and display the relationships between the most important variables and the label categories.

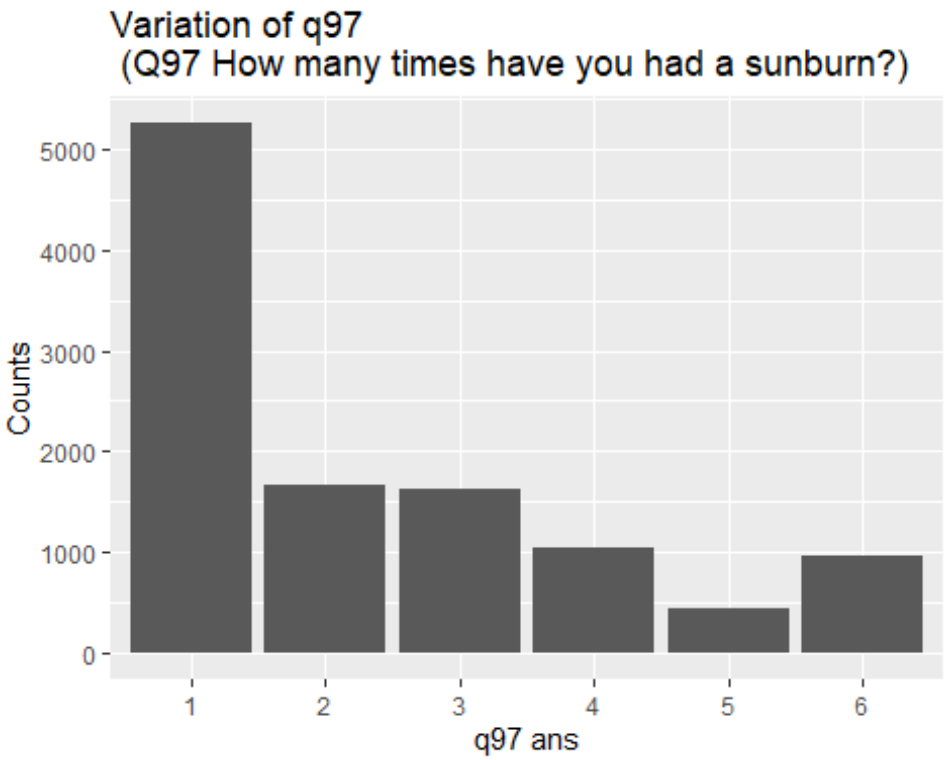
visualisation of variation across classes for top 3 variables

```
#q97

q97clean <- xr_new %>% subset(q97!="NA"&r!="NA")
head(q97clean)
```

```
##   q6 q7 q8 q9 q10 q11 q12 q13 q14 q15 q16 q17 q18 q19 q20 q21 q22 q2
3 q24 q25
## 1  1 54  2  5   5   1   1   1   1   1   1   2   3   2   2   2   2
2   1   1
## 2  1 51  2  5   1   1   1   1   1   1   1   1   1   1   1   2   1
1   2   2
## 3  1 66  1  5   1   1   1   1   1   1   3  NA  NA  NA  NA  NA   1  N
A   1   1
## 4 NA NA  2  5   1   2   4   4   1   1   1   1   2   1   2   2   1
1   2   2
## 5  1 68  1  4   3   1   1   1   1   1   1   1   1   1   1   1   4
5   1   2
## 6  1 59  2  5   1   1   1   1   1   1   1   1   1   1   1   2   2
2   1   1
##   q26 q27 q28 q29 q30 q31 q32 q33 q34 q35 q36 q37 q38 q39 q40 q41 q4
```

```
2 q43 q44
## 1   2   2   2   1   1   1   6  NA  NA  NA   2   1   6   1   3   6
5   1   1
## 2   2   2   2   1   1   2   1   1   1   1   1   1   1   2   1   1
1   1   1
## 3   1   1   2  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA   2  NA   2
5   1   1
## 4   1   2   2   1   1   1   6   2   2   5   3   1   4   1   2   5
5   2   2
## 5   1   1   1   1   1   1   6   5   4   5   3   1   1   1   7   6
5   2   2
## 6   1   1   1   3   3   1   1   1   1   1   1   1   1   2   1   3
6   2  NA
##   q45 q46 q47 q48 q49 q50 q51 q52 q53 q54 q55 q56 q57 q58 q59 q60 q6
1 q62 q63
## 1   1   1   7   6   4   1   1   1   1   1   1   1   1   1   1   1
7   3   3
## 2   1   1   1   1   1   1   1   1   1   1   1   1   1   1   2   2
1   1   1
## 3   1   1  NA   7  NA   1   1   1   1   1   1   1   1   1   2   1
2   3   3
## 4   5   6   7   5   6   2   1   1   2   1   2   1   4   1   2   1
5   5   2
## 5   7   5   7   5   1   3   1   1   3   2   1   1   4   1   1   1
6   7   4
## 6   4   7   6   6   4   1   1   1   1   1   1   1   2   1   1   1
6   5   5
##   q64 q65 q66 q67 q68 q69 q70 q71 q72 q73 q74 q75 q76 q77 q78 q79 q8
0 q81 q82
## 1   3   2   4   2   1   2   2   3   4   1   2   2   7   3   1   4
6   5   7
## 2   1   1   1   3   1   3   4   2   4   2   2   2   2   3   2   8
4  NA   1
## 3   3   2  NA   3   2   3   4   3   4   1   3   2   2   4   4   6
4   1   6
## 4   3   2   4   2   1   2   2   1   2   1   2   2   2   4   2   3
3   4   5
## 5   3   2   4   3   1   3   1   3   3   1   2   2   2   2   1   1
1   2   2
## 6   2   3   2   2   1   3   1   2   2   2   2   1   1   7   3   2
4   1   7
##   q83 q84 q85 q86 q87 q88 q89 q90 q91 q92 q93 q94 q95 q96 q97 q98 q9
9 r
## 1   1   1   2   1   2   4   4   2   1   1   4   3   1   1   1   1
1 2
## 2   1   1   2   1   2   5   2   1   1   1   5   3   4   1   1   2
1 6
## 3   1   1   1  NA  NA   5   2  NA   1   1   7   2   1   1   3   2
1 4
## 4   6   2   2   1   2   3   4  NA   3   2   4   2   1   1   1   1
```

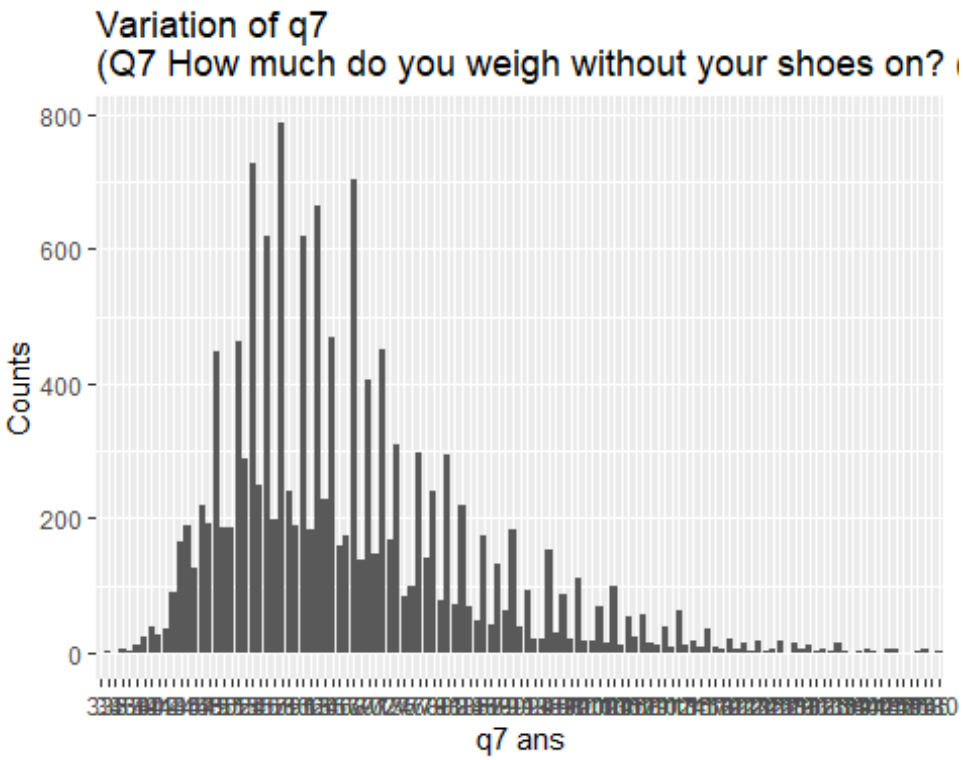
```
1 4
## 5   1   1   2   1   2   1   2   1   3   2   7   3   1   1   2   1
1 4
## 6   6   2   2   1   1   2   4   2   1   5   2   2   4   1   3   1
2 4

q97=as.data.frame(cbind(q97clean$q97, q97clean$r))
head(q97)

##   V1 V2
## 1  1  2
## 2  1  6
## 3  3  4
## 4  1  4
## 5  2  4
## 6  3  4

colnames(q97) = c("ans", "r")
plot_97=ggplot(data=q97,mapping=aes(x=factor(ans)))+geom_bar(stat= 'cou
nt')+labs(title="Variation of q97\n (Q97 How many times have you had a
sunburn?)",x = "q97 ans", y = "Counts")
plot_97
```



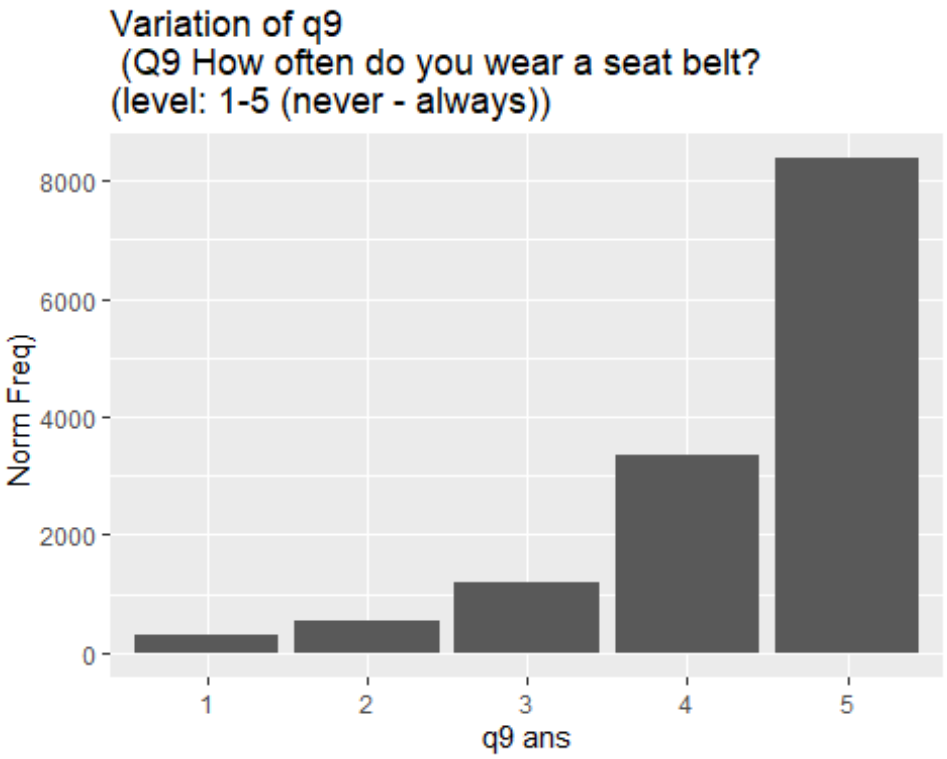

```
## Warning: Use of `q97$q97_1` is discouraged. Use `q97_1` instead.
#top2 q9=subset(data,q9!="NA"&r!="NA") q9_1=q9$q9 r_na_9=q9$r q9_trans=
cbind(q9_1,r_na_9) q9=as.data.frame(q9_trans)
```

Most of them don't like sunburn much.

```
q7clean <- xr_new %>% subset(q7!="NA"&r!="NA")
q7=as.data.frame(cbind(q7clean$q7, q7clean$r))
colnames(q7) = c("ans", "r")
plot_7=ggplot(data=q7,mapping=aes(x=factor(ans)))+geom_bar(stat= 'count
')+labs(title="Variation of q7 \n(Q7 How much do you weigh without your
 shoes on? (kilograms.)) ",x = "q7 ans", y = "Counts")
plot_7
```

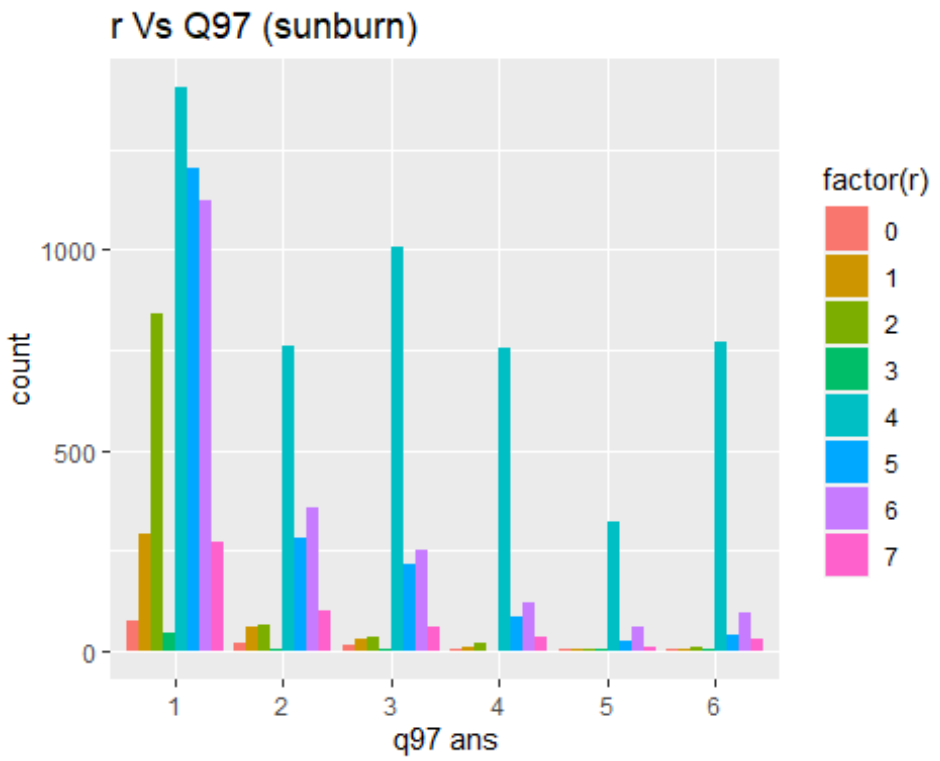



```
q9clean <- xr_new %>% subset(q9!="NA"&r!="NA")
q9=as.data.frame(cbind(q9clean$q9, q9clean$r))
colnames(q9) = c("ans", "r")
plot_9=ggplot(data=q9,mapping=aes(x=factor(ans)), fill = factor(q9[,
1]))+geom_bar(stat= 'count')+labs(title="Variation of q9 \n (Q9 How oft
en do you wear a seat belt? \n(level: 1-5 (never - always))", x = "q9 a
ns", y = "Norm Freq) ",x = "q9 ans", y = "Counts")
plot_9
```

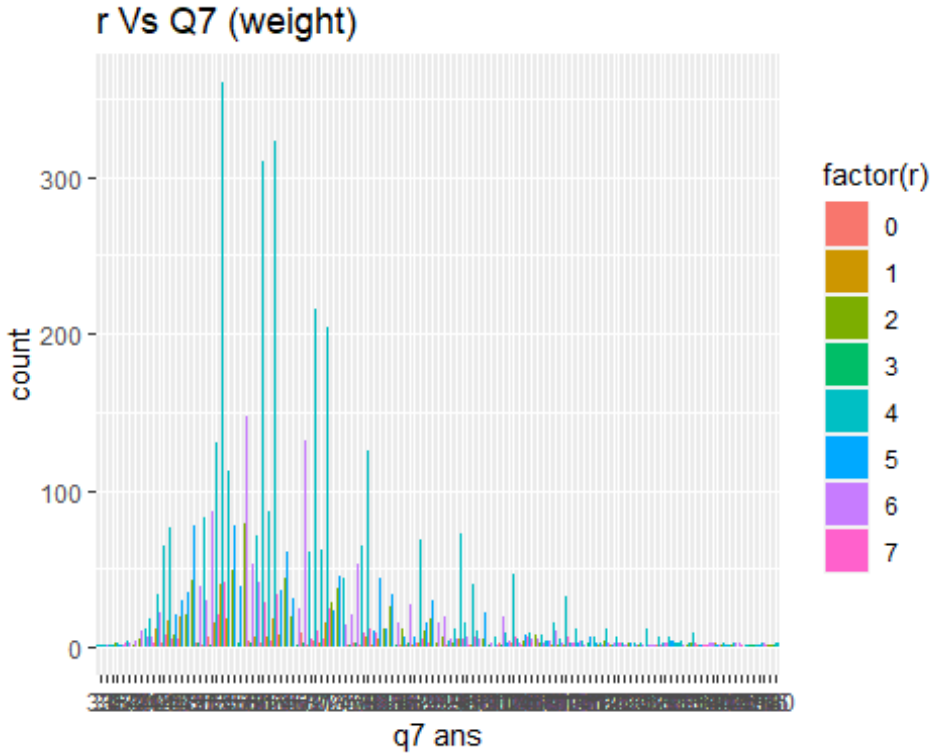


Most of them always wear a seat belt.

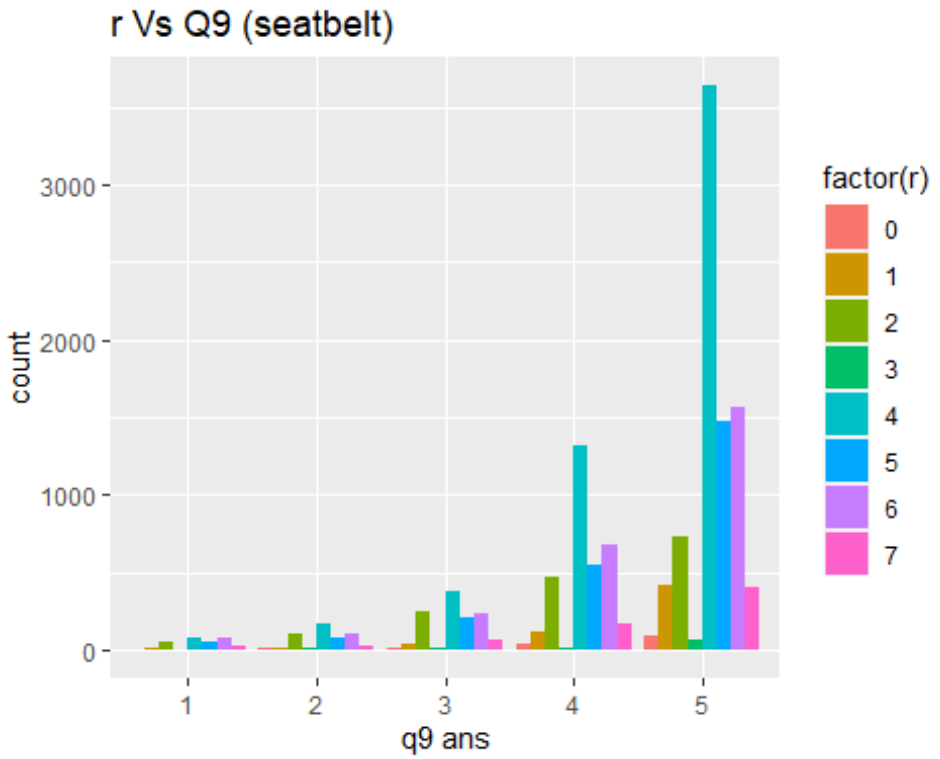
```
r_q97=ggplot(data=q97,aes(x=factor(ans)))+geom_bar(aes(fill=factor(r)),
position="dodge")+labs(title="r Vs Q97 (sunburn)", x = "q97 ans")
r_q97
```




```
r_q7=ggplot(data=q7,aes(x=factor(ans)))+geom_bar(aes(fill=factor(r)),po
sition="dodge")+labs(title="r Vs Q7 (weight)", x = "q7 ans")
r_q7
```



```
r_q9=ggplot(data=q9,aes(x=factor(ans)))+geom_bar(aes(fill=factor(r)),po
sition="dodge")+labs(title="r Vs Q9 (seatbelt)",x = "q9 ans")
r_q9
```



From plots above, we can see race 4 (white) like sunburn most and they seems heavier (greater weight) than other races. And generally speaking, every race wear a seat belt often.

Inspect all top variables to identify best for discriminating classes or fit model for each class and get top variables.

```
get_top = function(v){
  for(i in c(0:7)){
    train.label_ri = case_when(xr.train$r == i ~ TRUE, xr.train$r !=i ~
 FALSE)
    train.xgbmat <- xgb.DMatrix(data = as.matrix(xr.train[,-(95)]), lab
el = train.label_ri)
    xgb_races = xgboost(data = train.xgbmat, max_depth=6, eta=1,  objec
tive='binary:logistic', nround=30, lamda = 1)
    v[i+1] = xgb.importance(model = xgb_races)[1]
  }
  return(v)
}
v= rep(" ",7)
topv = get_top(v)

topv
```

```
## [[1]]
## [1] "q7"
##
## [[2]]
## [1] "q7"
##
## [[3]]
## [1] "q97"
##
## [[4]]
## [1] "q7"
##
## [[5]]
## [1] "q97"
##
## [[6]]
## [1] "q7"
##
## [[7]]
## [1] "q7"
##
## [[8]]
## [1] "q7"
```

So for race 0-7, the best discriminating variable is “q7” “q7” “q97” “q7” “q97” “q99” “q7” “q7”, respectively.

Task 4: Comment on whether (or not) task 3 would be ethically problematic if intended to be published, and for what reasons.

- First, the accuracy of this model is poor (about 50%). It is not good enough to be published. If we publish it and people believe it, it could be a problem.
- Second, the prediction of this model is about race and ethnicity which is one of the most sensitive topic in the world. Many people from different races could feel offended.
- Third, even though the model is good, if we get the model published, it will reinforce stereotypes of different race. The personality do exists everywhere, and we should respect it. In conclusion, publish this model is not a good idea.