R Notebook

The following is your first chunk to start with. Remember, you can add chunks using the menu above (Insert -> R) or using the keyboard shortcut Ctrl+Alt+I. A good practice is to use different code chunks to answer different questions. You can delete this comment if you like.

Other useful keyboard shortcuts include Alt- for the assignment operator, and Ctrl+Shift+M for the pipe operator. You can delete these reminders if you don't want them in your report.

```
#setwd("C:/") #Don't forget to set your working directory before you start!

library("tidyverse")

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

## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

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

library("tidymodels")

## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

## -- Attaching packages ---------------------------------- tidymodels 0.0.3
--

## v broom     0.5.3     v recipes   0.1.9
## v dials     0.0.4     v rsample   0.0.5
## v infer     0.5.1     v yardstick 0.0.4
## v parsnip   0.0.5

## -- Conflicts ------------------------------------------ tidymodels_conflicts()
--
## x scales::discard()   masks purrr::discard()
## x dplyr::filter()     masks stats::filter()
## x recipes::fixed()    masks stringr::fixed()
## x dplyr::lag()        masks stats::lag()
## x dials::margin()     masks ggplot2::margin()
```

```
## x yardstick::spec()   masks readr::spec()
## x recipes::step()     masks stats::step()
## x recipes::yj_trans() masks scales::yj_trans()
```

```r
library("plotly")
```

```
##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

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

## The following object is masked from 'package:graphics':
##
##     layout
```

```r
library("skimr")
library(car)
```

```
## Loading required package: carData

## Registered S3 methods overwritten by 'car':
##   method                          from
##   influence.merMod                lme4
##   cooks.distance.influence.merMod lme4
##   dfbeta.influence.merMod         lme4
##   dfbetas.influence.merMod        lme4

##
## Attaching package: 'car'

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

## The following object is masked from 'package:purrr':
##
##     some
```

```r
library(modelr)
```

```
##
## Attaching package: 'modelr'

## The following objects are masked from 'package:yardstick':
##
##     mae, mape, rmse
```

```
## The following object is masked from 'package:broom':
##
##     bootstrap
```

```r
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following objects are masked from 'package:yardstick':
##
##     precision, recall

## The following object is masked from 'package:purrr':
##
##     lift
```

```r
dff <-
  read_csv("framingham.csv")
```

```
## Parsed with column specification:
## cols(
##   gender = col_double(),
##   age = col_double(),
##   education = col_double(),
##   currentSmoker = col_double(),
##   cigsPerDay = col_double(),
##   BPMeds = col_double(),
##   prevalentStroke = col_double(),
##   prevalentHyp = col_double(),
##   diabetes = col_double(),
##   totChol = col_double(),
##   sysBP = col_double(),
##   diaBP = col_double(),
##   BMI = col_double(),
##   heartRate = col_double(),
##   glucose = col_double(),
##   TenYearCHD = col_double()
## )
```

```r
dff
```

```
## # A tibble: 3,658 x 16
##    gender   age education currentSmoker cigsPerDay BPMeds prevalentStroke
##     <dbl> <dbl>     <dbl>         <dbl>      <dbl>  <dbl>           <dbl>
##  1      1    39         4             0          0      0               0
##  2      0    46         2             0          0      0               0
##  3      1    48         1             1         20      0               0
##  4      0    61         3             1         30      0               0
##  5      0    46         3             1         23      0               0
```

```
##  6      0    43         2             0          0      0               0
##  7      0    63         1             0          0      0               0
##  8      0    45         2             1         20      0               0
##  9      1    52         1             0          0      0               0
## 10      1    43         1             1         30      0               0
## # ... with 3,648 more rows, and 9 more variables: prevalentHyp <dbl>,
## #   diabetes <dbl>, totChol <dbl>, sysBP <dbl>, diaBP <dbl>, BMI <dbl>,
## #   heartRate <dbl>, glucose <dbl>, TenYearCHD <dbl>
```

```r
head(dff)
```

```
## # A tibble: 6 x 16
##   gender   age education currentSmoker cigsPerDay BPMeds prevalentStroke
##    <dbl> <dbl>     <dbl>         <dbl>      <dbl>  <dbl>           <dbl>
## 1      1    39         4             0          0      0               0
## 2      0    46         2             0          0      0               0
## 3      1    48         1             1         20      0               0
## 4      0    61         3             1         30      0               0
## 5      0    46         3             1         23      0               0
## 6      0    43         2             0          0      0               0
## # ... with 9 more variables: prevalentHyp <dbl>, diabetes <dbl>, totChol <
dbl>,
## #   sysBP <dbl>, diaBP <dbl>, BMI <dbl>, heartRate <dbl>, glucose <dbl>,
## #   TenYearCHD <dbl>
```

```r
nrow(dff)
```

```
## [1] 3658
```

```r
skim(dff)
```

Data summary

Name	dff
Number of rows	3658
Number of columns	16
_____	
Column type frequency:	
numeric	16
_____	
Group variables	None

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist

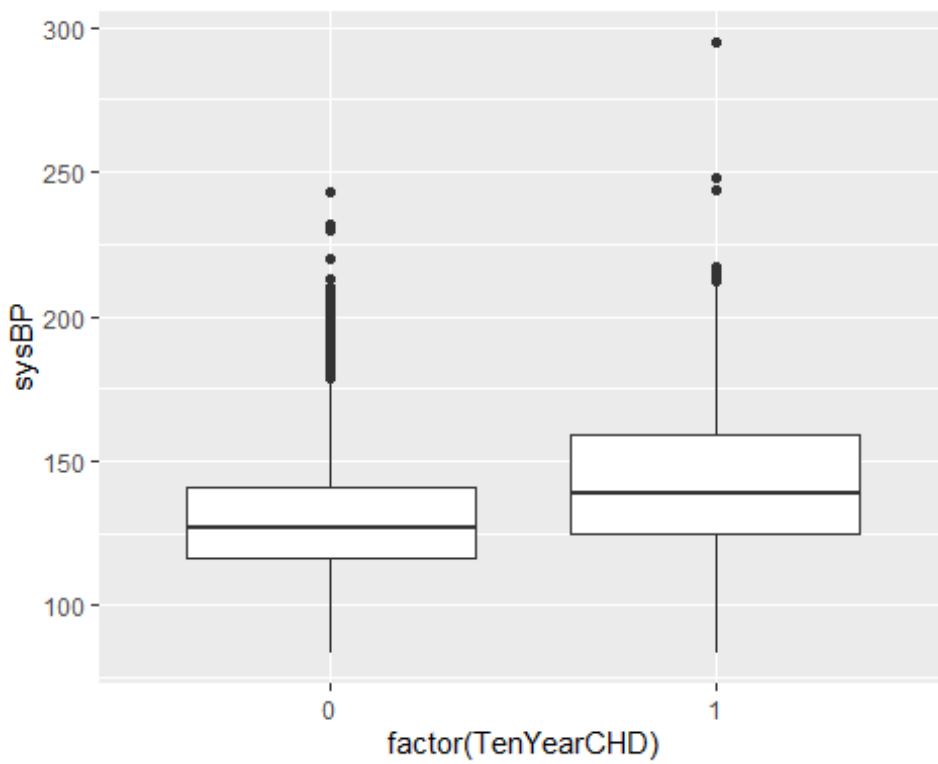
gender	0	1	0.44	0.50	0.00	0.00	0.00	1.00	1.0
age	0	1	49.55	8.56	32.00	42.00	49.00	56.00	70.0
education	0	1	1.98	1.02	1.00	1.00	2.00	3.00	4.0
currentSmoker	0	1	0.49	0.50	0.00	0.00	0.00	1.00	1.0
cigsPerDay	0	1	9.03	11.92	0.00	0.00	0.00	20.00	70.0
BPMeds	0	1	0.03	0.17	0.00	0.00	0.00	0.00	1.0
prevalentStroke	0	1	0.01	0.08	0.00	0.00	0.00	0.00	1.0
prevalentHyp	0	1	0.31	0.46	0.00	0.00	0.00	1.00	1.0
diabetes	0	1	0.03	0.16	0.00	0.00	0.00	0.00	1.0
totChol	0	1	236.85	44.10	113.00	206.00	234.00	263.00	600.0
sysBP	0	1	132.37	22.09	83.50	117.00	128.00	143.88	295.0
diaBP	0	1	82.92	11.97	48.00	75.00	82.00	90.00	142.5
BMI	0	1	25.78	4.07	15.54	23.08	25.38	28.04	56.8
heartRate	0	1	75.73	11.98	44.00	68.00	75.00	82.00	143.0
glucose	0	1	81.85	23.90	40.00	71.00	78.00	87.00	394.0
TenYearCHD	0	1	0.15	0.36	0.00	0.00	0.00	0.00	1.0

```
colsToFactor<- c('gender','education','currentSmoker','BPMeds','prevalentStroke','prevalentHyp','diabetes')

dff<-dff%>%
  mutate_at(colsToFactor, ~factor(.))

str(dff)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 3658 obs. of  16
variables:
##  $ gender         : Factor w/ 2 levels "0","1": 2 1 2 1 1 1 1 1 2 2 ...
##  $ age            : num  39 46 48 61 46 43 63 45 52 43 ...
##  $ education      : Factor w/ 4 levels "1","2","3","4": 4 2 1 3 3 2 1 2 1
1 ...
##  $ currentSmoker  : Factor w/ 2 levels "0","1": 1 1 2 2 2 1 1 2 1 2 ...
##  $ cigsPerDay     : num  0 0 20 30 23 0 0 20 0 30 ...
##  $ BPMeds         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ prevalentStroke: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ prevalentHyp   : Factor w/ 2 levels "0","1": 1 1 1 2 1 2 1 1 2 2 ...
##  $ diabetes       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ totChol        : num  195 250 245 225 285 228 205 313 260 225 ...
##  $ sysBP          : num  106 121 128 150 130 ...
##  $ diaBP          : num  70 81 80 95 84 110 71 71 89 107 ...
##  $ BMI            : num  27 28.7 25.3 28.6 23.1 ...
##  $ heartRate      : num  80 95 75 65 85 77 60 79 76 93 ...
##  $ glucose        : num  77 76 70 103 85 99 85 78 79 88 ...
##  $ TenYearCHD     : num  0 0 0 1 0 0 1 0 0 0 ...
```

Q1

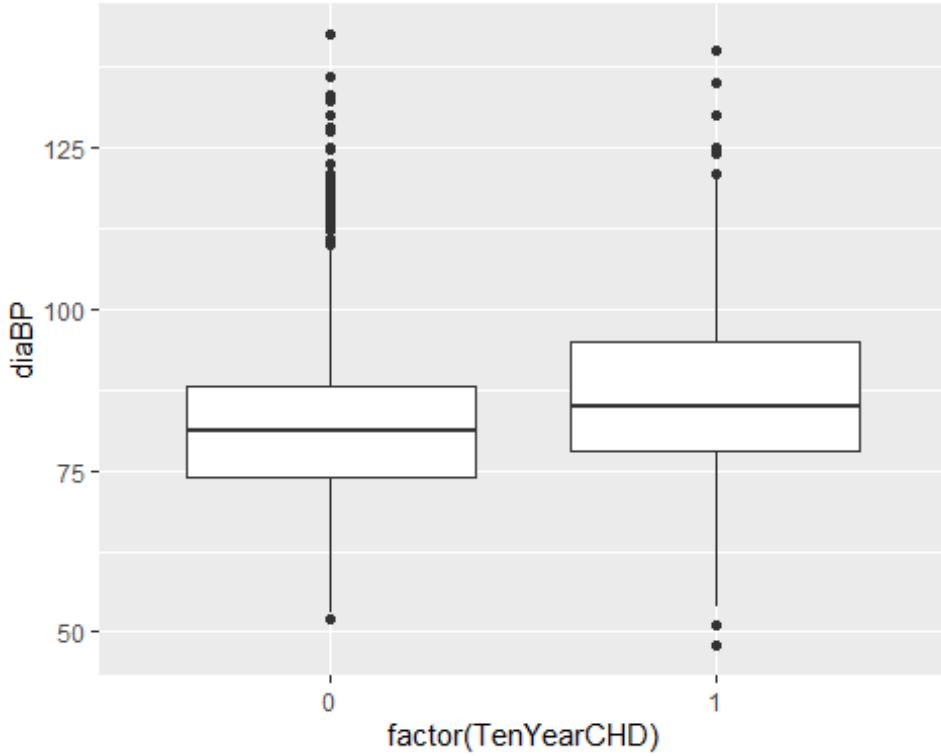
```
ggplot(data=dff)+ geom_boxplot(aes(x=factor(TenYearCHD),y=sysBP))
```




```
ggplotly(ggplot(data=dff)+ geom_boxplot(aes(x=factor(TenYearCHD),y=sysBP)))
```

Q1

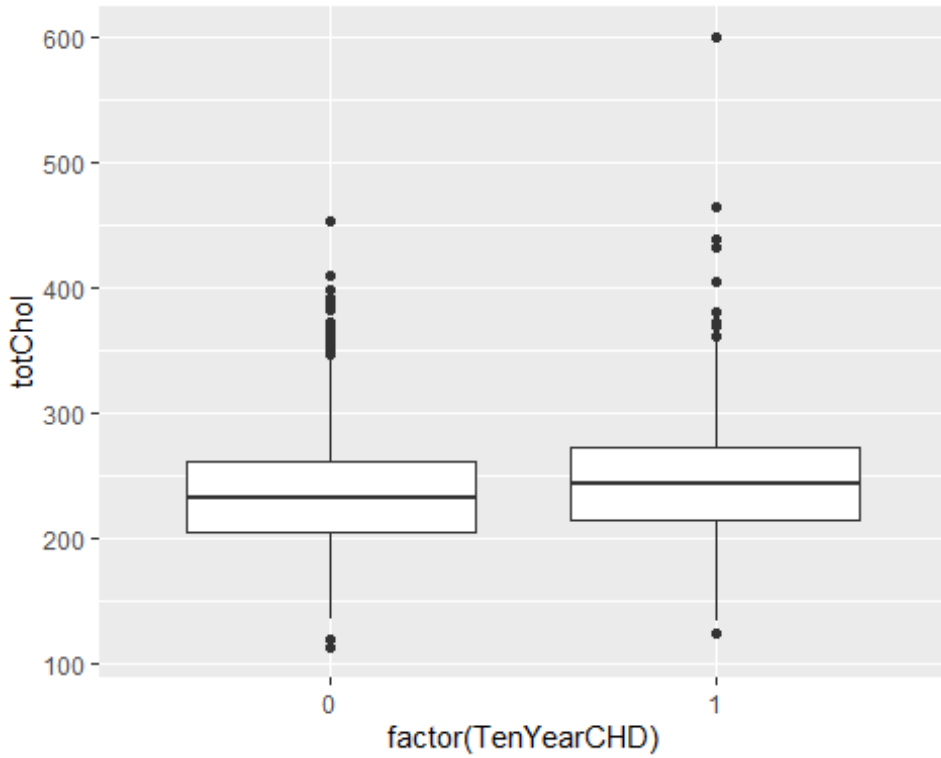
```r
ggplot(data=dff)+ geom_boxplot(aes(x=factor(TenYearCHD),y=diaBP))
```




```r
ggplotly(ggplot(data=dff)+ geom_boxplot(aes(x=factor(TenYearCHD),y=diaBP)))
```

Q1

```r
ggplot(data=dff)+ geom_boxplot(aes(x=factor(TenYearCHD),y=totChol))
```




```
ggplotly(ggplot(data=dff)+ geom_boxplot(aes(x=factor(TenYearCHD),y=totChol)))
```

Q2)i)

```
#Setting the seed
set.seed(123)

#Creating the training dataset by random sampling 80% of the data
dffTrain <- dff %>% sample_frac(0.7)

#Assigning the difference to the test set
dffTest <- dplyr::setdiff(dff, dffTrain)
```

Q2ii)

```
dffTest%>%group_by(gender)%>%tally%>%

  mutate(pct=(100*n)/sum(n))

## # A tibble: 2 x 3
##   gender     n   pct
##   <fct>  <int> <dbl>
## 1 0        616  56.2
## 2 1        481  43.8
```

```r
dffTrain%>%group_by(gender)%>%tally%>%

  mutate(pct=(100*n)/sum(n))
```

```
## # A tibble: 2 x 3
##   gender     n   pct
##   <fct>  <int> <dbl>
## 1 0       1419  55.4
## 2 1       1142  44.6
```

Q2)B)

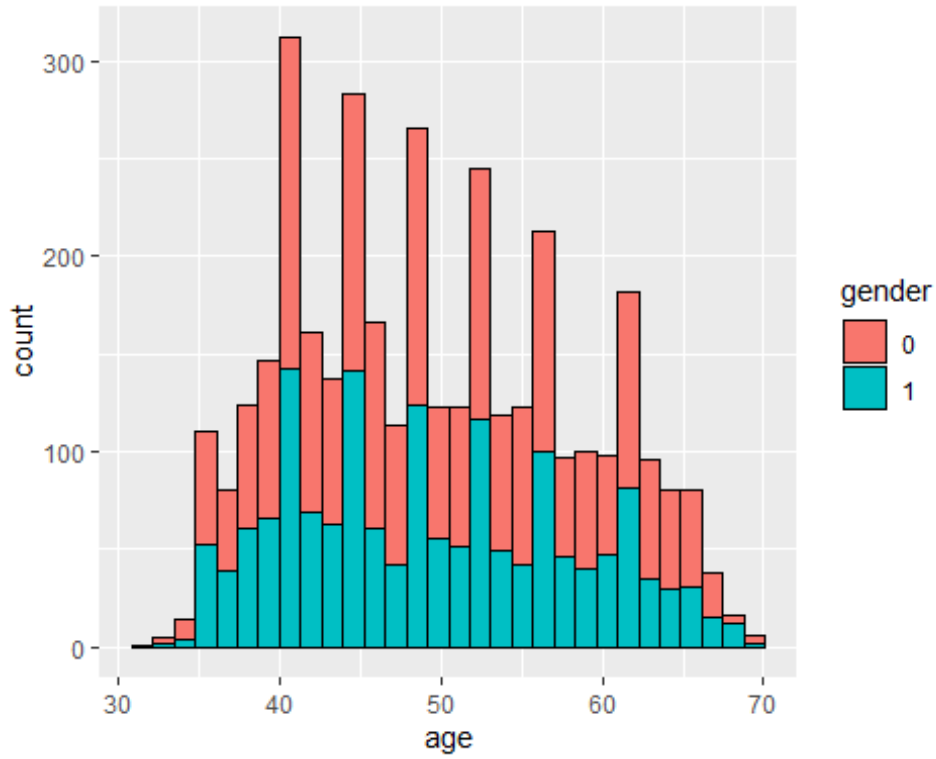
```r
dffTrain %>%
  group_by(ageGroup=cut_interval(age,length=10)) %>%
  tally  %>%
  #group_by(school_number) %>%
  mutate(pct=(100*n)/sum(n))
```

```
## # A tibble: 4 x 3
##   ageGroup     n   pct
##   <fct>    <int> <dbl>
## 1 [30,40]    467  18.2
## 2 (40,50]    973  38.0
## 3 (50,60]    772  30.1
## 4 (60,70]    349  13.6
```

Q2)C)

```r
ggplot(data=dff, aes(age,fill=gender)) + geom_histogram(color='black')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



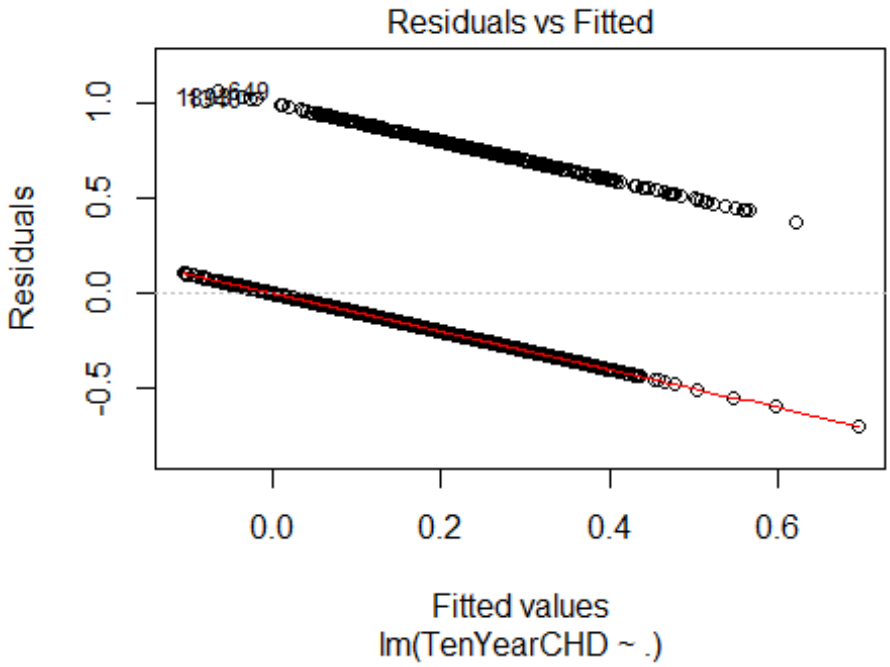

```
ggplotly(ggplot(data=dff, aes(age,fill=gender)) + geom_histogram(color='black
'))

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

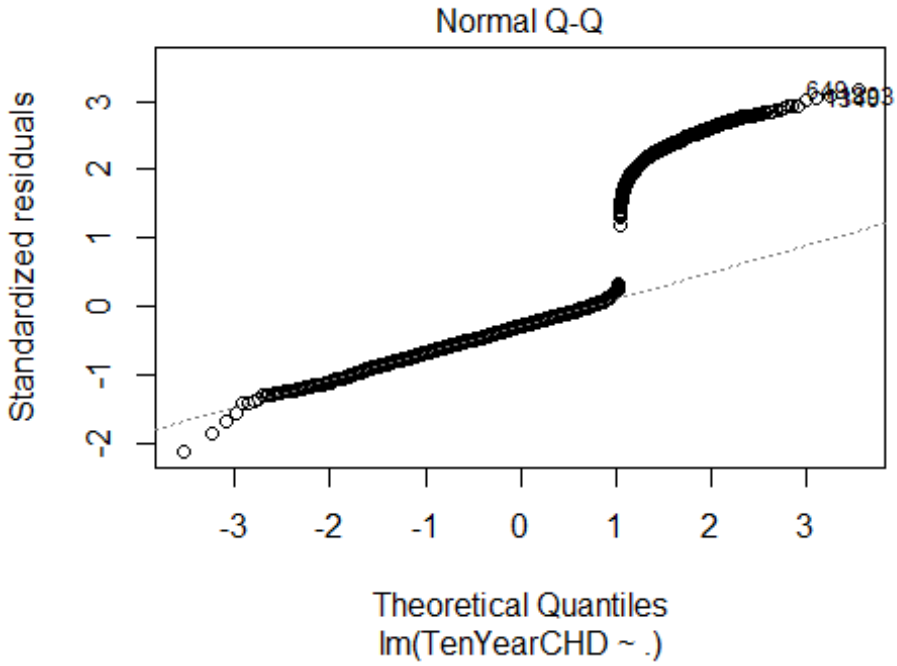
Q3)A]

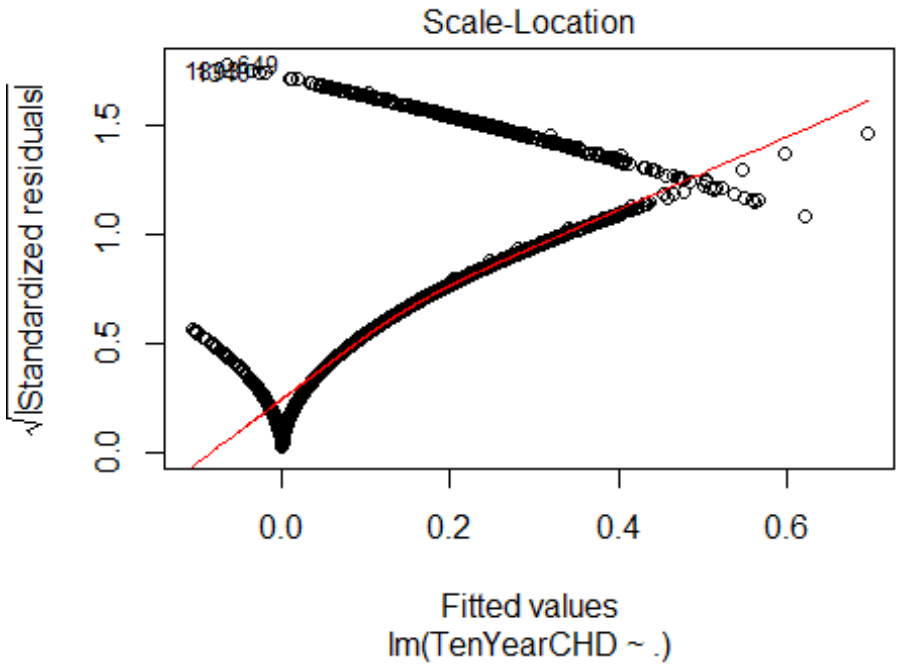
```
fitLPM<-lm(TenYearCHD~ ., data=dffTrain)

summary(fitLPM)

##
## Call:
## lm(formula = TenYearCHD ~ ., data = dffTrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.69588 -0.18760 -0.09864 -0.00854  1.06563
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -0.5193243  0.0939086  -5.530 3.53e-08 ***
## gender1           0.0402834  0.0149552   2.694  0.00711 **
## age               0.0073056  0.0009204   7.938 3.06e-15 ***
## education2       -0.0114841  0.0167200  -0.687  0.49224
## education3       -0.0345910  0.0196551  -1.760  0.07854 .
## education4       -0.0259428  0.0230652  -1.125  0.26080
```

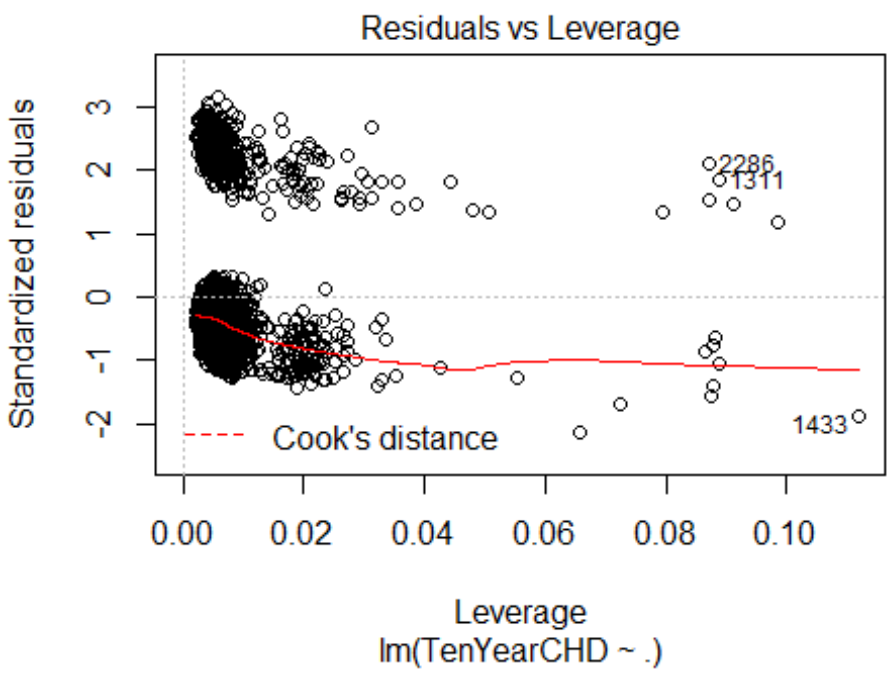
```
## currentSmoker1    0.0143681  0.0216179   0.665  0.50634
## cigsPerDay        0.0018669  0.0009316   2.004  0.04519 *
## BPMeds1           0.0184297  0.0434995   0.424  0.67184
## prevalentStroke1  0.2099878  0.0983542   2.135  0.03285 *
## prevalentHyp1     0.0448001  0.0208879   2.145  0.03206 *
## diabetes1         0.0204464  0.0513727   0.398  0.69066
## totChol           0.0002882  0.0001590   1.813  0.07000 .
## sysBP             0.0023876  0.0005798   4.118 3.95e-05 ***
## diaBP            -0.0016597  0.0009716  -1.708  0.08770 .
## BMI               0.0007242  0.0018265   0.397  0.69175
## heartRate        -0.0013046  0.0005843  -2.233  0.02566 *
## glucose           0.0011775  0.0003608   3.264  0.00111 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3388 on 2543 degrees of freedom
## Multiple R-squared:  0.1077, Adjusted R-squared:  0.1017
## F-statistic: 18.05 on 17 and 2543 DF,  p-value: < 2.2e-16
```

```r
plot(fitLPM)
```


Residuals vs Fitted
Residuals
1.0
0.5
0.0
-0.5
0.0
0.2
0.4
0.6
Fitted values
lm(TenYearCHD ~ .)


Normal Q-Q
Standardized residuals
3
2
1
0
-1
-2
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(TenYearCHD ~ .)






```
vif(fitLPM)

##                      GVIF Df GVIF^(1/(2*Df))
## gender           1.232950  1        1.110383
```

```
## age              1.398367  1        1.182526
## education        1.139817  3        1.022051
## currentSmoker    2.604754  1        1.613925
## cigsPerDay       2.762784  1        1.662163
## BPMeds           1.106826  1        1.052058
## prevalentStroke 1.006585  1        1.003287
## prevalentHyp     2.057398  1        1.434363
## diabetes         1.630615  1        1.276956
## totChol          1.106930  1        1.052107
## sysBP            3.777158  1        1.943491
## diaBP            2.997947  1        1.731458
## BMI              1.227604  1        1.107973
## heartRate        1.095878  1        1.046842
## glucose          1.645722  1        1.282857
```

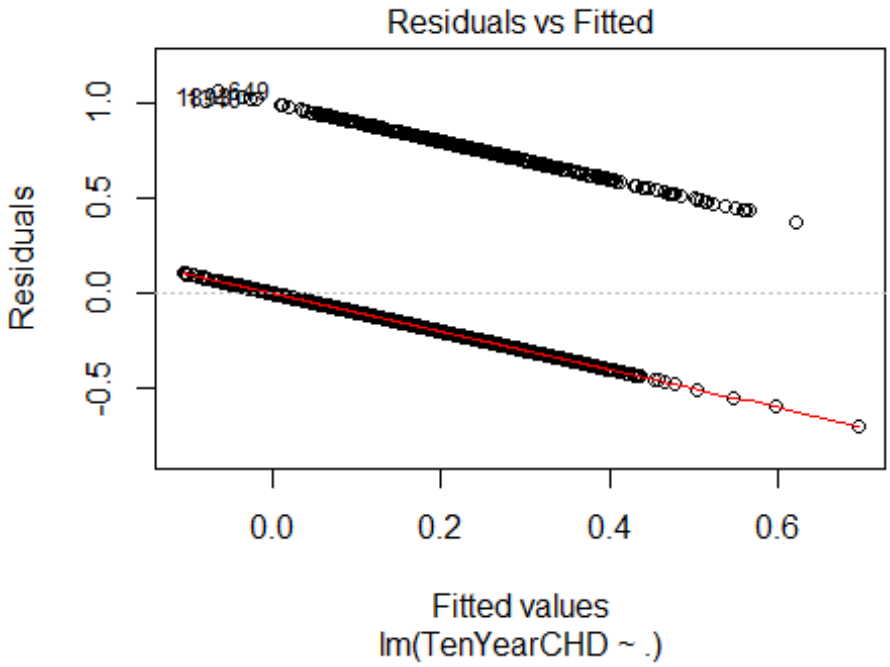
Q3)B)

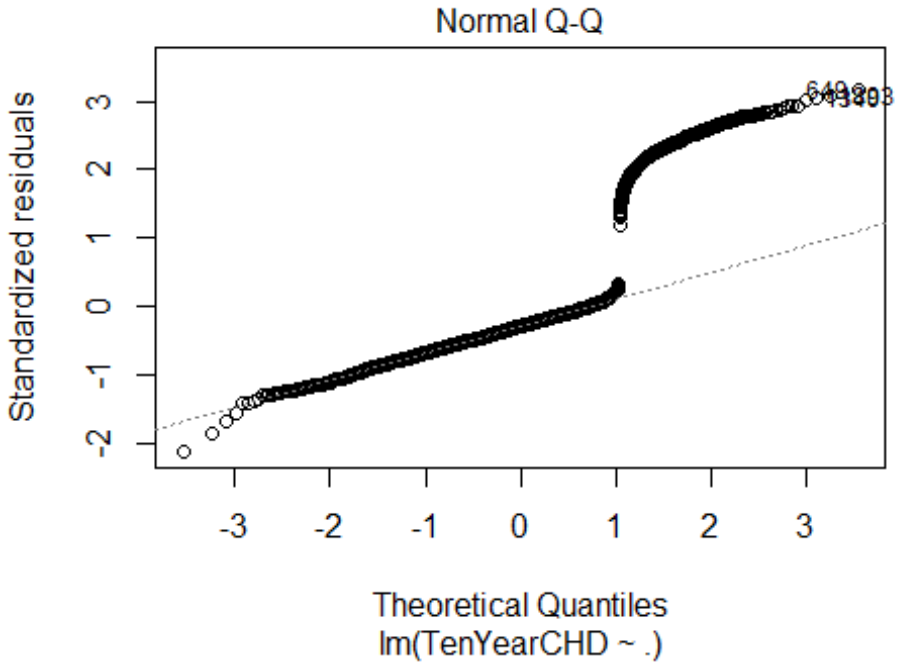
```
library(car)


fitLPM2<-lm(TenYearCHD~ .-currentSmoker, data=dffTrain)

summary(fitLPM2)
```

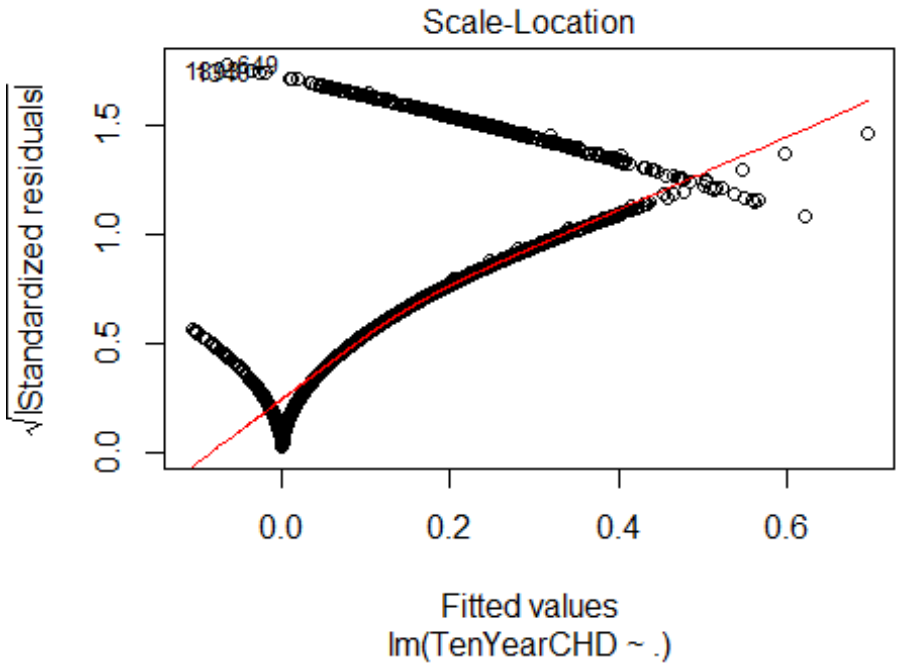
```
##
## Call:
## lm(formula = TenYearCHD ~ . - currentSmoker, data = dffTrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.69721 -0.18848 -0.09967 -0.00937  1.07518
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -0.5092583  0.0926691  -5.495 4.28e-08 ***
## gender1            0.0396262  0.0149208   2.656 0.007962 **
## age                0.0072591  0.0009176   7.911 3.78e-15 ***
## education2        -0.0113009  0.0167159  -0.676 0.499067
## education3        -0.0346151  0.0196529  -1.761 0.078304 .
## education4        -0.0260964  0.0230615  -1.132 0.257909
## cigsPerDay         0.0023323  0.0006145   3.795 0.000151 ***
## BPMeds1            0.0185984  0.0434940   0.428 0.668972
## prevalentStroke1  0.2097097  0.0983425   2.132 0.033066 *
## prevalentHyp1      0.0448426  0.0208855   2.147 0.031882 *
## diabetes1          0.0203925  0.0513670   0.397 0.691403
## totChol            0.0002875  0.0001590   1.809 0.070633 .
## sysBP              0.0023882  0.0005798   4.119 3.92e-05 ***
## diaBP             -0.0016833  0.0009708  -1.734 0.083051 .
## BMI                0.0006191  0.0018194   0.340 0.733670
## heartRate         -0.0013019  0.0005843  -2.228 0.025944 *
```

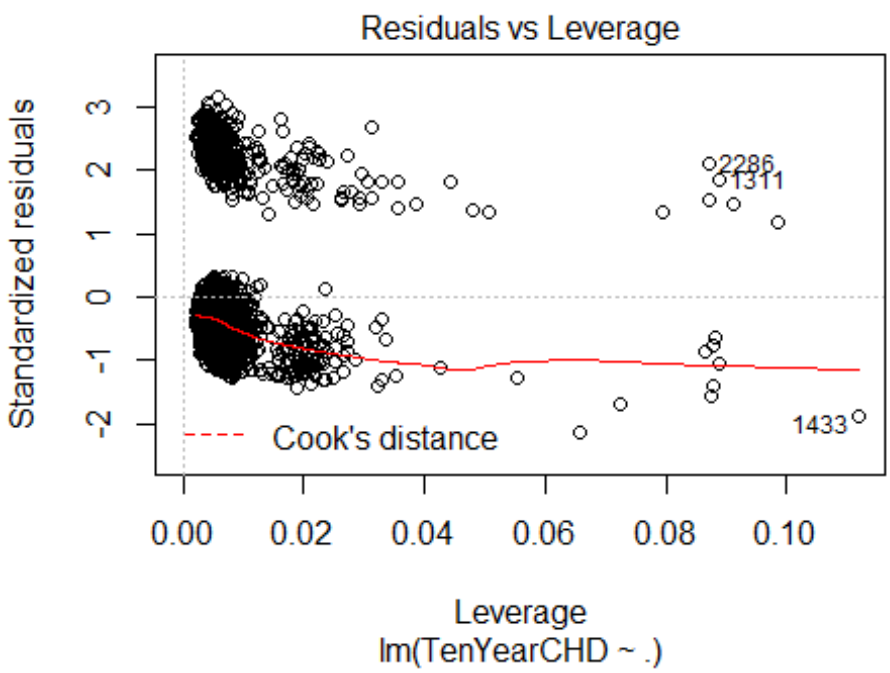
```
## glucose           0.0011752  0.0003607   3.258 0.001138 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3388 on 2544 degrees of freedom
## Multiple R-squared:  0.1075, Adjusted R-squared:  0.1019
## F-statistic: 19.16 on 16 and 2544 DF,  p-value: < 2.2e-16
```

```
plot(fitLPM)
```


Residuals vs Fitted
Residuals
1.0
0.5
0.0
-0.5
0.0
0.2
0.4
0.6
Fitted values
lm(TenYearCHD ~ .)


Normal Q-Q
Standardized residuals
3
2
1
0
-1
-2
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(TenYearCHD ~ .)






```
vif(fitLPM2)

##                      GVIF Df GVIF^(1/(2*Df))
## gender           1.227561  1        1.107954
```

```
## age             1.390293  1        1.179107
## education       1.139163  3        1.021953
## cigsPerDay      1.202282  1        1.096486
## BPMeds          1.106788  1        1.052040
## prevalentStroke 1.006566  1        1.003278
## prevalentHyp    2.057379  1        1.434357
## diabetes        1.630611  1        1.276954
## totChol         1.106882  1        1.052085
## sysBP           3.777149  1        1.943489
## diaBP           2.993948  1        1.730303
## BMI             1.218397  1        1.103810
## heartRate       1.095825  1        1.046817
## glucose         1.645572  1        1.282799
```

Q4)

```
resultsLPM<-
  lm(TenYearCHD~ .-currentSmoker, data=dffTrain)%>%

    predict(dffTest, type='response' ) %>%    #=> Use the option type='respon
se' for probabilities
    bind_cols(dffTest, predictedProb=.) %>%
  mutate(predictedClass = ifelse(predictedProb > 0.5, 1, 0))

#resultsLPM%>%arran




#ge(desc(predictedProb))

#resultsLPM <- subset(resultsLPM, select = -c(predictedClass) )


summary( lm(TenYearCHD~ .-currentSmoker, data=dffTrain))

##
## Call:
## lm(formula = TenYearCHD ~ . - currentSmoker, data = dffTrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.69721 -0.18848 -0.09967 -0.00937  1.07518
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -0.5092583  0.0926691  -5.495 4.28e-08 ***
## gender1           0.0396262  0.0149208   2.656 0.007962 **
## age               0.0072591  0.0009176   7.911 3.78e-15 ***
## education2       -0.0113009  0.0167159  -0.676 0.499067
```

```
## education3       -0.0346151  0.0196529  -1.761 0.078304 .
## education4       -0.0260964  0.0230615  -1.132 0.257909
## cigsPerDay        0.0023323  0.0006145   3.795 0.000151 ***
## BPMeds1           0.0185984  0.0434940   0.428 0.668972
## prevalentStroke1  0.2097097  0.0983425   2.132 0.033066 *
## prevalentHyp1     0.0448426  0.0208855   2.147 0.031882 *
## diabetes1         0.0203925  0.0513670   0.397 0.691403
## totChol           0.0002875  0.0001590   1.809 0.070633 .
## sysBP             0.0023882  0.0005798   4.119 3.92e-05 ***
## diaBP            -0.0016833  0.0009708  -1.734 0.083051 .
## BMI               0.0006191  0.0018194   0.340 0.733670
## heartRate        -0.0013019  0.0005843  -2.228 0.025944 *
## glucose           0.0011752  0.0003607   3.258 0.001138 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3388 on 2544 degrees of freedom
## Multiple R-squared:  0.1075, Adjusted R-squared:  0.1019
## F-statistic: 19.16 on 16 and 2544 DF,  p-value: < 2.2e-16
```

```
dffTest %>%
  group_by((TenYearCHD)) %>%
  tally  %>%
  #group_by(school_number) %>%
  mutate(pct=(100*n)/sum(n))
```

```
## # A tibble: 2 x 3
##   `(TenYearCHD)`     n   pct
##            <dbl> <int> <dbl>
## 1              0   925  84.3
## 2              1   172  15.7
```

```
resultsLPM %>%
  group_by((predictedClass)) %>%
  tally  %>%
  #group_by(school_number) %>%
  mutate(pct=(100*n)/sum(n))
```

```
## # A tibble: 2 x 3
##   `(predictedClass)`     n    pct
##                <dbl> <int>  <dbl>
## 1                  0  1087 99.1
## 2                  1    10  0.912
```

```
dffTest %>%
  group_by(TenYearCHD) %>%
  tally  %>%
  #group_by(school_number) %>%
  mutate(pct=(100*n)/sum(n))
```

```
## # A tibble: 2 x 3
##   TenYearCHD     n   pct
##        <dbl> <int> <dbl>
## 1          0   925  84.3
## 2          1   172  15.7
```

```
resultsLPM %>%
  group_by(predictedClass) %>%
  tally  %>%
  #group_by(school_number) %>%
  mutate(pct=(100*n)/sum(n))
```

```
## # A tibble: 2 x 3
##   predictedClass     n    pct
##            <dbl> <int>  <dbl>
## 1              0  1087 99.1
## 2              1    10  0.912
```

```
colsToFactor<- c('TenYearCHD')
dffTest<-dffTest%>%
  mutate_at(colsToFactor, ~factor(.))


colsToFactor<- c('TenYearCHD')
dffTrain<-dffTrain%>%
  mutate_at(colsToFactor, ~factor(.))

str(dffTrain)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 2561 obs. of  16
variables:
##  $ gender         : Factor w/ 2 levels "0","1": 1 2 2 1 1 2 1 1 2 1 ...
##  $ age            : num  63 43 53 64 57 40 55 57 62 60 ...
##  $ education      : Factor w/ 4 levels "1","2","3","4": 3 4 4 2 2 4 2 2 1
1 ...
##  $ currentSmoker  : Factor w/ 2 levels "0","1": 1 2 1 2 1 2 1 1 2 1 ...
##  $ cigsPerDay     : num  0 25 0 9 0 25 0 0 30 0 ...
##  $ BPMeds         : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ prevalentStroke: Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ prevalentHyp   : Factor w/ 2 levels "0","1": 1 1 1 2 1 1 1 1 1 1 ...
##  $ diabetes       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ totChol        : num  281 296 207 250 175 258 271 239 373 391 ...
##  $ sysBP          : num  125 137 102 145 123 ...
##  $ diaBP          : num  80 90 72.5 79 72 78 80 81 85 64 ...
##  $ BMI            : num  21.4 24 26.5 25.2 22.4 ...
##  $ heartRate      : num  75 72 72 73 77 80 100 75 80 82 ...
##  $ glucose        : num  99 97 95 86 74 70 89 87 67 83 ...
##  $ TenYearCHD     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

Q5)

```r
Logist<-glm(TenYearCHD~.-currentSmoker,family='binomial',data=dffTrain)

summary(Logist)

##
## Call:
## glm(formula = TenYearCHD ~ . - currentSmoker, family = "binomial",
##     data = dffTrain)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.8022  -0.5882  -0.4071  -0.2738   2.8363
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -7.927497   0.846875  -9.361  < 2e-16 ***
## gender1           0.422202   0.133313   3.167 0.001540 **
## age               0.066797   0.008110   8.237  < 2e-16 ***
## education2       -0.079672   0.146967  -0.542 0.587743
## education3       -0.329631   0.183167  -1.800 0.071921 .
## education4       -0.236143   0.213615  -1.105 0.268960
## cigsPerDay        0.020000   0.005146   3.886 0.000102 ***
## BPMeds1          -0.002423   0.294477  -0.008 0.993434
## prevalentStroke1  1.152421   0.659094   1.748 0.080379 .
## prevalentHyp1     0.338398   0.166699   2.030 0.042358 *
## diabetes1        -0.005002   0.374594  -0.013 0.989345
## totChol           0.003606   0.001338   2.696 0.007017 **
## sysBP             0.014442   0.004495   3.213 0.001315 **
## diaBP            -0.007077   0.007813  -0.906 0.365014
## BMI               0.011682   0.015070   0.775 0.438211
## heartRate        -0.011470   0.005157  -2.224 0.026137 *
## glucose           0.007397   0.002634   2.808 0.004983 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2168.1  on 2560  degrees of freedom
## Residual deviance: 1894.3  on 2544  degrees of freedom
## AIC: 1928.3
##
## Number of Fisher Scoring iterations: 5

exp(coef(Logist))

##      (Intercept)          gender1              age       education2
##     0.0003606879     1.5253171095     1.0690784440     0.9234189417
##       education3       education4       cigsPerDay          BPMeds1
##     0.7191887265     0.7896676736     1.0202012574     0.9975796686
## prevalentStroke1    prevalentHyp1        diabetes1          totChol
```

```
##     3.1658488040     1.4026980839     0.9950101842     1.0036127972
##            sysBP            diaBP              BMI        heartRate
##     1.0145465769     0.9929479273     1.0117507851     0.9885958031
##          glucose
##     1.0074239785
```

Q6

```
resultsLog <-
  glm(TenYearCHD~.-currentSmoker, family='binomial', data=dffTrain) %>%
  predict(dffTest, type='response')%>%
  bind_cols(dffTest, predictedProb=.)%>%
  mutate(predictedClass = as.factor(ifelse(predictedProb>0.5, 1, 0)))


resultsLog
```

```
## # A tibble: 1,097 x 18
##    gender   age education currentSmoker cigsPerDay BPMeds prevalentStroke
##    <fct>  <dbl> <fct>     <fct>              <dbl> <fct>  <fct>
##  1 1         48 1         1                     20 0      0
##  2 0         43 2         0                      0 0      0
##  3 0         43 2         0                      0 0      0
##  4 0         41 3         0                      0 1      0
##  5 0         52 3         1                     20 0      0
##  6 0         61 3         0                      0 0      0
##  7 1         46 1         1                     20 0      0
##  8 0         63 2         1                     40 0      0
##  9 0         62 1         0                      0 0      0
## 10 1         49 1         1                      2 0      0
## # ... with 1,087 more rows, and 11 more variables: prevalentHyp <fct>,
## #   diabetes <fct>, totChol <dbl>, sysBP <dbl>, diaBP <dbl>, BMI <dbl>,
## #   heartRate <dbl>, glucose <dbl>, TenYearCHD <fct>, predictedProb <dbl>,
## #   predictedClass <fct>
```

```
summary(resultsLog)
```

```
##  gender       age        education currentSmoker   cigsPerDay     BPMeds
##  0:616   Min.   :32.00   1:461     0:553         Min.   : 0.000   0:1055
##  1:481   1st Qu.:43.00   2:331     1:544         1st Qu.: 0.000   1:  42
##          Median :49.00   3:169                   Median : 0.000
##          Mean   :49.64   4:136                   Mean   : 9.062
##          3rd Qu.:56.00                           3rd Qu.:20.000
##          Max.   :68.00                           Max.   :60.000
##  prevalentStroke prevalentHyp diabetes    totChol          sysBP
##  0:1088          0:734        0:1071   Min.   :133.0   Min.   : 83.5
##  1:   9          1:363        1:  26   1st Qu.:206.0   1st Qu.:118.0
##                                        Median :237.0   Median :129.0
##                                        Mean   :238.2   Mean   :132.5
##                                        3rd Qu.:266.0   3rd Qu.:143.0
##                                        Max.   :392.0   Max.   :215.0
```

```
##      diaBP             BMI          heartRate        glucose       TenYear
CHD
##  Min.   : 48.00   Min.   :16.59   Min.   : 44.0   Min.   : 40.00   0:925
##  1st Qu.: 74.50   1st Qu.:23.05   1st Qu.: 68.0   1st Qu.: 71.00   1:172
##  Median : 82.00   Median :25.45   Median : 75.0   Median : 78.00
##  Mean   : 83.22   Mean   :25.75   Mean   : 75.9   Mean   : 82.14
##  3rd Qu.: 90.00   3rd Qu.:27.93   3rd Qu.: 83.0   3rd Qu.: 87.00
##  Max.   :140.00   Max.   :43.67   Max.   :143.0   Max.   :394.00
##  predictedProb     predictedClass
##  Min.   :0.01444   0:1078
##  1st Qu.:0.06139   1:  19
##  Median :0.11555
##  Mean   :0.15320
##  3rd Qu.:0.21276
##  Max.   :0.92677
```

```
resultsLog %>%
  group_by(predictedClass) %>%
  tally  %>%
  #group_by(school_number) %>%
  mutate(pct=(100*n)/sum(n))
```

```
## # A tibble: 2 x 3
##   predictedClass     n   pct
##   <fct>          <int> <dbl>
## 1 0               1078 98.3
## 2 1                 19  1.73
```

Q7

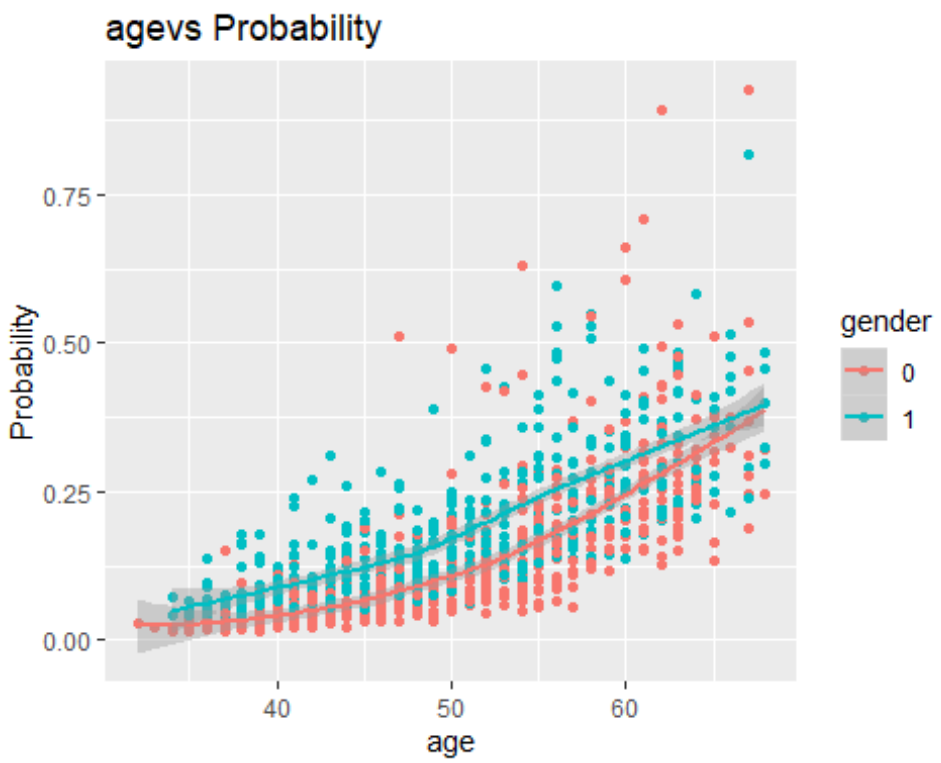
```
resultsLog%>%
  conf_mat(truth=TenYearCHD, estimate=predictedClass)
```

```
##           Truth
## Prediction   0   1
##          0 919 159
##          1   6  13
```

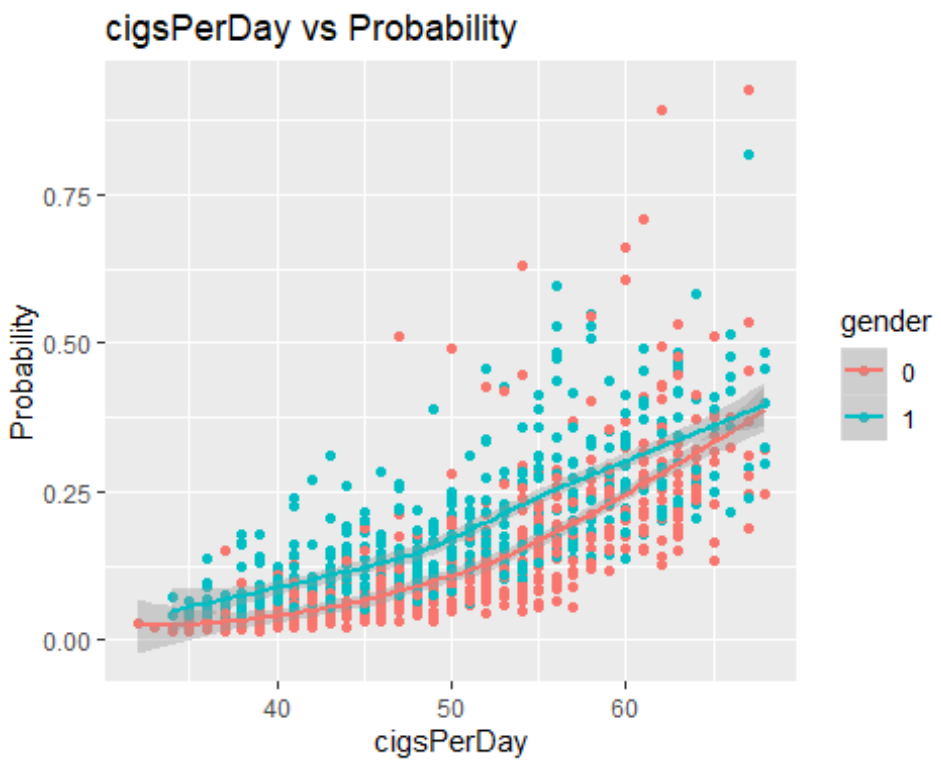
Q8

```
resultsLog%>%ggplot(aes(x=age,y=predictedProb,color=gender))+geom_point()+geo
m_smooth()+labs(title="agevs Probability",x="age",y="Probability")
```

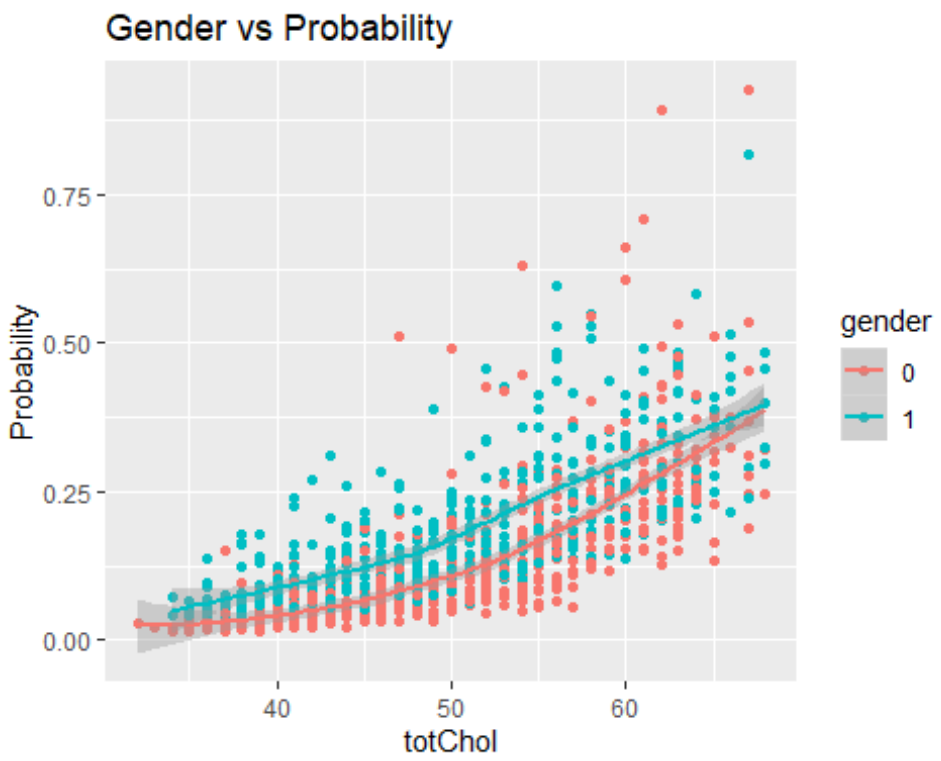
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



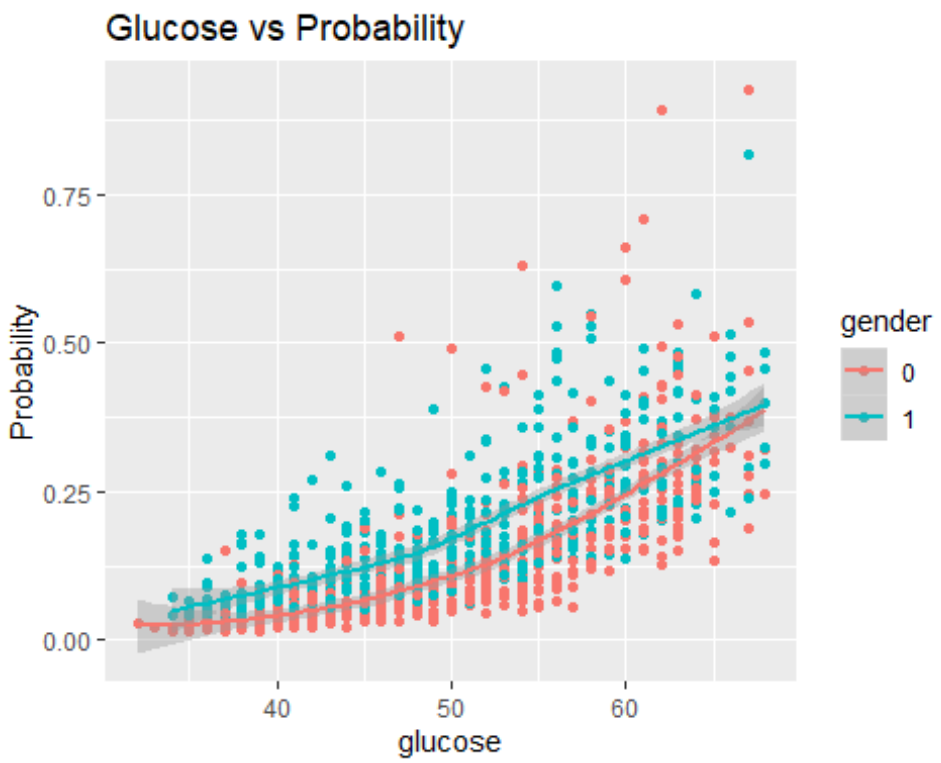

```
ggplotly(resultsLog%>%ggplot(aes(x=age,y=predictedProb,color=gender))+geom_po
int()+geom_smooth()+labs(title="agevs Probability",x="age",y="Probability"))

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

#?labs

resultsLog%>%ggplot(aes(x=age,y=predictedProb,color=gender))+geom_point()+geo
m_smooth()+labs(title="cigsPerDay vs Probability",x="cigsPerDay",y="Probabili
ty")

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
resultsLog%>%ggplot(aes(x=age,y=predictedProb,color=gender))+geom_point()+geo
m_smooth()+labs(title="Gender vs Probability",x="totChol",y="Probability")

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```




```
resultsLog%>%ggplot(aes(x=age,y=predictedProb,color=gender))+geom_point()+geo
m_smooth()+labs(title="Glucose vs Probability",x="glucose",y="Probability")

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Q9

```
library(e1071)
library(caret)
resultsLogCaret<-
  train(TenYearCHD ~ .-currentSmoker, family='binomial', data=dffTrain, metho
d='glm') %>%
  predict(dffTest, type='raw')%>%
  bind_cols(dffTest, predictedClass=.)


resultsLogCaret%>%
  xtabs(~predictedClass+TenYearCHD, .)%>%
  confusionMatrix(positive='1')

## Confusion Matrix and Statistics
##
##                TenYearCHD
## predictedClass   0   1
##              0 919 159
##              1   6  13
##
##                Accuracy : 0.8496
##                  95% CI : (0.827, 0.8702)
##     No Information Rate : 0.8432
##     P-Value [Acc > NIR] : 0.297
##
```

```
##                   Kappa : 0.1083
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.07558
##             Specificity : 0.99351
##          Pos Pred Value : 0.68421
##          Neg Pred Value : 0.85250
##              Prevalence : 0.15679
##          Detection Rate : 0.01185
##    Detection Prevalence : 0.01732
##       Balanced Accuracy : 0.53455
##
##        'Positive' Class : 1
##
```

```
df <-
  read_csv("lab3BancoPortugal.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   job = col_character(),
##   marital = col_character(),
##   education = col_character(),
##   default = col_character(),
##   housing = col_character(),
##   loan = col_character(),
##   contact = col_character(),
##   month = col_character(),
##   day_of_week = col_character(),
##   poutcome = col_character(),
##   agegroup = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
head(df)
```

```
## # A tibble: 6 x 23
##     age job   marital education default housing loan  contact month day_of
_week
##   <dbl> <chr> <chr>   <chr>     <chr>   <chr>   <chr> <chr>   <chr> <chr>
## 1    56 hous~ married basic.4y  no      no      no    teleph~ may   mon
## 2    37 serv~ married high.sch~ no      yes     no    teleph~ may   mon
## 3    40 admi~ married basic.6y  no      no      no    teleph~ may   mon
## 4    56 serv~ married high.sch~ no      no      yes   teleph~ may   mon
## 5    59 admi~ married professi~ no      no      no    teleph~ may   mon
## 6    24 tech~ single  professi~ no      yes     no    teleph~ may   mon
## # ... with 13 more variables: duration <dbl>, campaign <dbl>, pdays <dbl>,
## #   previous <dbl>, poutcome <chr>, emp.var.rate <dbl>, cons.price.idx <db
```

```
l>,
## #   cons.conf.idx <dbl>, euribor3m <dbl>, nr.employed <dbl>,
## #   openedAccount <dbl>, agegroup <chr>, newcustomer <dbl>
```

```r
nrow(df)
```

```
## [1] 30488
```

```r
skim(df)
```

Data summary

Name	df
Number of rows	30488
Number of columns	23

Column type frequency:

character	11
numeric	12

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
job	0	1	6	13	0	11	0
marital	0	1	6	8	0	3	0
education	0	1	8	19	0	7	0
default	0	1	2	3	0	2	0
housing	0	1	2	3	0	2	0
loan	0	1	2	3	0	2	0
contact	0	1	8	9	0	2	0
month	0	1	3	3	0	10	0
day_of_week	0	1	3	3	0	5	0
poutcome	0	1	7	11	0	3	0
agegroup	0	1	6	15	0	4	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist

age	0	1	39.03	10.33	17.00	31.00	37.00	45.00	95.00	▅▇▂▁
duration	0	1	259.48	261.71	0.00	103.00	181.00	321.00	4918.00	▇▁▁▁
campaign	0	1	2.52	2.72	1.00	1.00	2.00	3.00	43.00	▇▁▁▁
pdays	0	1	956.33	201.37	0.00	999.00	999.00	999.00	999.00	▁▁▁▁▇
previous	0	1	0.19	0.52	0.00	0.00	0.00	0.00	7.00	▇▁▁▁
emp.var.rate	0	1	-0.07	1.61	-3.40	-1.80	1.10	1.40	1.40	▁▃▁▂▇
cons.price.idx	0	1	93.52	0.59	92.20	93.08	93.44	93.99	94.77	▂▇▂▇▂
cons.conf.idx	0	1	-40.60	4.79	-50.80	-42.70	-41.80	-36.40	-26.90	▅▇▂▇▁
euribor3m	0	1	3.46	1.78	0.63	1.31	4.86	4.96	5.04	▇▁▁▁▇
nr.employed	0	1	5160.81	75.16	4963.60	5099.10	5191.00	5228.10	5228.10	▁▁▃▁▇
openedAccount	0	1	0.13	0.33	0.00	0.00	0.00	0.00	1.00	▇▁▁▁
newcustomer	0	1	0.85	0.36	0.00	1.00	1.00	1.00	1.00	▂▁▁▁▇

```r
set.seed(123)

#Creating the training dataset by random sampling 80% of the data
dfTrain <- df %>% sample_frac(0.7)

#Assigning the difference to the test set
dfTest <- dplyr::setdiff(df, dfTrain)

colsToFactor<- c('openedAccount')
dfTest<-dfTest%>%
  mutate_at(colsToFactor, ~factor(.))


colsToFactor<- c('openedAccount')
dfTrain<-dfTrain%>%
  mutate_at(colsToFactor, ~factor(.))
```

```r
str(dfTrain)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 21342 obs. of  23
variables:
##  $ age           : num  51 40 35 28 75 34 31 26 32 72 ...
##  $ job           : chr  "technician" "management" "admin." "self-employed"
...
##  $ marital       : chr  "married" "married" "single" "single" ...
##  $ education     : chr  "high.school" "high.school" "high.school" "univers
ity.degree" ...
##  $ default       : chr  "no" "no" "no" "no" ...
##  $ housing       : chr  "yes" "no" "yes" "no" ...
##  $ loan          : chr  "no" "yes" "no" "no" ...
##  $ contact       : chr  "cellular" "cellular" "cellular" "cellular" ...
##  $ month         : chr  "nov" "nov" "jul" "may" ...
##  $ day_of_week   : chr  "fri" "fri" "fri" "fri" ...
##  $ duration      : num  167 105 147 386 153 103 142 291 700 1 ...
##  $ campaign      : num  4 5 2 1 2 2 3 2 3 1 ...
##  $ pdays         : num  999 999 14 999 999 999 999 999 999 999 ...
##  $ previous      : num  0 1 2 0 0 0 0 0 0 1 ...
##  $ poutcome      : chr  "nonexistent" "failure" "failure" "nonexistent" ..
.
##  $ emp.var.rate  : num  -0.1 -0.1 -2.9 -1.8 -1.8 1.1 1.1 -1.8 1.1 -1.1 ...
##  $ cons.price.idx: num  93.2 93.2 92.5 92.9 93.4 ...
##  $ cons.conf.idx : num  -42 -42 -33.6 -46.2 -34.8 -36.4 -36.4 -46.2 -36.4
-37.5 ...
##  $ euribor3m     : num  4.021 4.021 1.059 1.25 0.639 ...
##  $ nr.employed   : num  5196 5196 5076 5099 5009 ...
##  $ openedAccount : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ agegroup      : chr  "Adults" "Adults" "Adults" "Young Adults" ...
##  $ newcustomer   : num  1 0 0 1 1 1 1 1 1 0 ...
```

```r
library(car)
Model1<-
  train(openedAccount~. , family='binomial', data=dfTrain, method='glm')%>%
  predict(dfTest, type='raw')%>%
  bind_cols(dfTest, predictedClass=.)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

Model1%>%
  xtabs(~predictedClass+openedAccount, .)%>%
  confusionMatrix(positive='1')
```

```
## Confusion Matrix and Statistics
##
##                openedAccount
## predictedClass    0    1
##              0 7721  644
##              1  248  529
##
##                Accuracy : 0.9024
##                  95% CI : (0.8962, 0.9084)
##     No Information Rate : 0.8717
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4905
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.45098
##             Specificity : 0.96888
##          Pos Pred Value : 0.68082
##          Neg Pred Value : 0.92301
##              Prevalence : 0.12831
##          Detection Rate : 0.05786
##    Detection Prevalence : 0.08499
##       Balanced Accuracy : 0.70993
##
##        'Positive' Class : 1
##
```

```
Model2<-
  train(openedAccount~.-contact , family='binomial', data=dfTrain, method='gl
m')%>%
  predict(dfTest, type='raw')%>%
  bind_cols(dfTest, predictedClass=.)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
```

```
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading
```

```
Model2%>%
  xtabs(~predictedClass+openedAccount, .)%>%
  confusionMatrix(positive='1')
```

```
## Confusion Matrix and Statistics
##
##               openedAccount
## predictedClass    0    1
##              0 7735  650
##              1  234  523
```

```
##
##                Accuracy : 0.9033
##                  95% CI : (0.8971, 0.9093)
##     No Information Rate : 0.8717
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4907
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.44587
##             Specificity : 0.97064
##          Pos Pred Value : 0.69089
##          Neg Pred Value : 0.92248
##              Prevalence : 0.12831
##          Detection Rate : 0.05721
##    Detection Prevalence : 0.08280
##       Balanced Accuracy : 0.70825
##
##        'Positive' Class : 1
##
```

```
Model108<-
  #ModelNew<-#model 2

  train(openedAccount~. -newcustomer-marital-housing-previous-euribor3m-agegr
oup,data = dfTrain,family = 'binomial',method='glm') %>%
  predict(dfTest,type='raw') %>%
  bind_cols(dfTest,predictedClass=.)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading
```

```
Model108 %>%
  xtabs(~predictedClass+openedAccount,.) %>%
  confusionMatrix(positive='1')
```

```
## Confusion Matrix and Statistics
##
##               openedAccount
## predictedClass    0    1
##              0 7729  644
##              1  240  529
##
##                Accuracy : 0.9033
##                  95% CI : (0.8971, 0.9093)
##     No Information Rate : 0.8717
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4933
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.45098
##             Specificity : 0.96988
##          Pos Pred Value : 0.68791
##          Neg Pred Value : 0.92309
##              Prevalence : 0.12831
##          Detection Rate : 0.05786
##    Detection Prevalence : 0.08412
##       Balanced Accuracy : 0.71043
##
```

```
##        'Positive' Class : 1
##
```

```
Model1080<-
  #ModelNew<-#model 2

  train(openedAccount~. -newcustomer-marital-housing-previous-euribor3m-age,d
ata = dfTrain,family = 'binomial',method='glm') %>%
  predict(dfTest,type='raw') %>%
  bind_cols(dfTest,predictedClass=.)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
```

```
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type
== :
## prediction from a rank-deficient fit may be misleading
```

```
Model1080 %>%
  xtabs(~predictedClass+openedAccount,.) %>%
  confusionMatrix(positive='1')
```

```
## Confusion Matrix and Statistics
##
##               openedAccount
## predictedClass    0    1
##              0 7727  644
##              1  242  529
##
##                Accuracy : 0.9031
##                  95% CI : (0.8968, 0.9091)
##     No Information Rate : 0.8717
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4926
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.45098
##             Specificity : 0.96963
##          Pos Pred Value : 0.68612
##          Neg Pred Value : 0.92307
##              Prevalence : 0.12831
##          Detection Rate : 0.05786
##    Detection Prevalence : 0.08434
##       Balanced Accuracy : 0.71031
##
##        'Positive' Class : 1
##
```