<!DOCTYPE html>

Analysis of NHANES 2018 Health Data

0. Data Processing in python

```
import pandas as pd import numpy as np import matplotlib.pyplot as
plt import seaborn as sns

demo = pd.read_sas("DEMO_J.XPT") diabetes = pd.read_sas("DIQ_J.XPT")
bld_pr = pd.read_sas("BPX_J.XPT") bmi_m = pd.read_sas("BMX_J.XPT")


demo.head()
```

```
##       SEQN  SDDSRVYR  RIDSTATR  ...  INDHHIN2  INDFMIN2  INDFMPIR
## 0  93703.0      10.0       2.0  ...      15.0      15.0      5.00
## 1  93704.0      10.0       2.0  ...      15.0      15.0      5.00
## 2  93705.0      10.0       2.0  ...       3.0       3.0      0.82
## 3  93706.0      10.0       2.0  ...       NaN       NaN       NaN
## 4  93707.0      10.0       2.0  ...      10.0      10.0      1.88
##
## [5 rows x 46 columns]
```

```
bld_pr.head()
```

```
##       SEQN  PEASCCT1  BPXCHR  BPAARM  ...  BPAEN3  BPXSY4  BPXDI4  BPAEN4
## 0  93703.0       NaN   120.0     NaN  ...     NaN     NaN     NaN     NaN
## 1  93704.0       NaN   114.0     NaN  ...     NaN     NaN     NaN     NaN
## 2  93705.0       NaN     NaN     1.0  ...     2.0   198.0    74.0     2.0
## 3  93706.0       NaN     NaN     1.0  ...     2.0     NaN     NaN     NaN
## 4  93707.0       NaN     NaN     1.0  ...     2.0     NaN     NaN     NaN
##
## [5 rows x 21 columns]
```

```
bmi_m.head()
```

```
##       SEQN  BMDSTATS  BMXWT  BMIWT  ...  BMXWAIST  BMIWAIST  BMXHIP  BMIHIP
## 0  93703.0       1.0   13.7    3.0  ...      48.2       NaN     NaN     NaN
## 1  93704.0       1.0   13.9    NaN  ...      50.0       NaN     NaN     NaN
## 2  93705.0       1.0   79.5    NaN  ...     101.8       NaN   110.0     NaN
## 3  93706.0       1.0   66.3    NaN  ...      79.3       NaN    94.4     NaN
## 4  93707.0       1.0   45.4    NaN  ...      64.1       NaN    83.0     NaN
##
## [5 rows x 21 columns]
```

```
merged_inner_1 = pd.merge(left=demo, right=bmi_m, left_on="SEQN", right_on="SEQN")
merged_inner_2 = pd.merge(left=merged_inner_1, right=bld_pr, left_on="SEQN", right_on="SEQN")
combined = pd.merge(left=merged_inner_2, right = diabetes, left_on="SEQN", right_on = "SEQN")

combined.head()
```

```
##       SEQN  SDDSRVYR  RIDSTATR  RIAGENDR  ...  DID350  DIQ350U  DIQ360  DIQ080
## 0  93703.0      10.0       2.0       2.0  ...     NaN      NaN     NaN     NaN
## 1  93704.0      10.0       2.0       1.0  ...     NaN      NaN     NaN     NaN
## 2  93705.0      10.0       2.0       2.0  ...     NaN      NaN     NaN     NaN
## 3  93706.0      10.0       2.0       1.0  ...     NaN      NaN     NaN     NaN
## 4  93707.0      10.0       2.0       1.0  ...     NaN      NaN     NaN     NaN
##
## [5 rows x 139 columns]
```

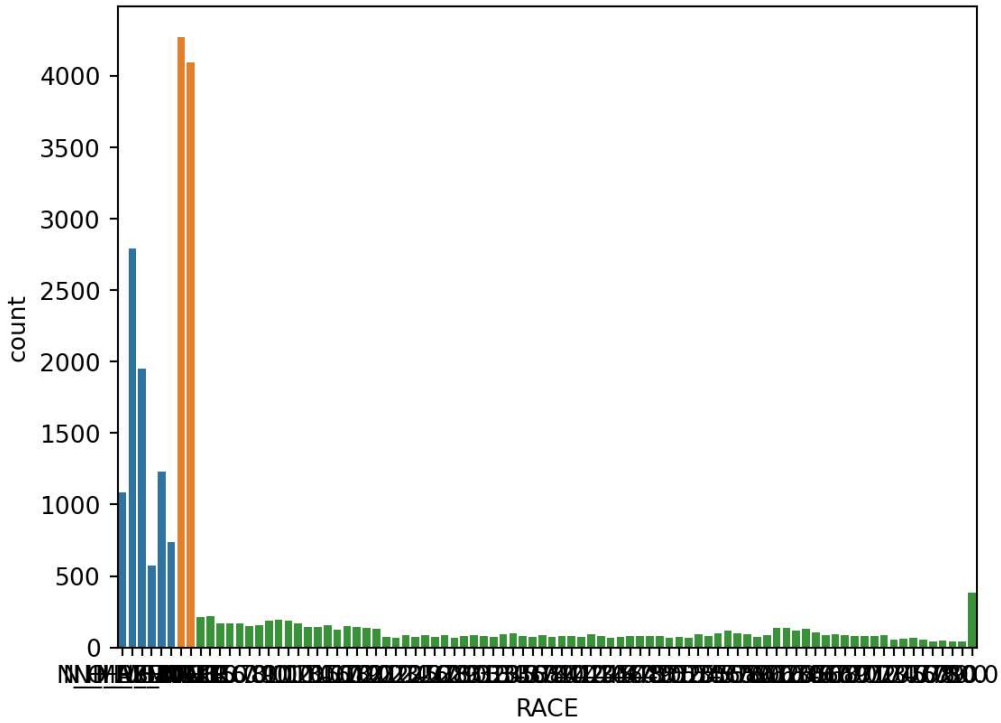
```
health = combined[["RIDRETH3","RIAGENDR","RIDAGEYR","BMXBMI","BPXPLS","BPXPULS","BPXSY2", "BPXDI2",
"DIQ010"]]

health = health.rename(columns={"RIDRETH3": "RACE","RIAGENDR": "SEX", "RIDAGEYR" : "AGE", "BMXBMI":
"BMI", "BPXPLS":"PULSE_P/M", "BPXPULS":"PULSE_TYPE", "BPXSY2":"BLD_PRE_S", "BPXDI2":"BLD_PRE_D", "DI
Q010":"DIABETES" })
health.head()
```

```
##    RACE  SEX   AGE   BMI  PULSE_P/M  PULSE_TYPE  BLD_PRE_S  BLD_PRE_D  DIABETES
## 0   6.0  2.0   2.0  17.5        NaN         1.0        NaN        NaN       2.0
## 1   3.0  1.0   2.0  15.7        NaN         1.0        NaN        NaN       2.0
## 2   4.0  2.0  66.0  31.7       52.0         1.0        NaN        NaN       2.0
## 3   6.0  1.0  18.0  21.5       82.0         1.0      114.0       70.0       2.0
## 4   7.0  1.0  13.0  18.1      100.0         1.0      128.0       46.0       2.0
```

```
for i in range(0,len(health["RACE"])):
    if health["RACE"][i] == 1.0:
        health["RACE"][i] = "MEX_AMER"
    if health["RACE"][i] == 2.0:
        health["RACE"][i] = "HISPANIC"
    if health["RACE"][i] == 3.0:
        health["RACE"][i] = "N_H_WHITE"
    if health["RACE"][i] == 4.0:
        health["RACE"][i] = "N_H_BLACK"
    if health["RACE"][i] == 6.0:
        health["RACE"][i] = "N_H_ASIAN"
    if health["RACE"][i] == 7.0:
        health["RACE"][i] = "O_R_MULTI"

for j in range(0,len(health["SEX"])):
    if health["SEX"][j] == 1.0:
        health["SEX"][j] = "MALE"
    if health["SEX"][j] == 2.0:
        health["SEX"][j] = "FEMALE"

for k in range(0,len(health["PULSE_TYPE"])):
    if health["PULSE_TYPE"][k] == 1.0:
        health["PULSE_TYPE"][k] = "REGULAR"
    if health["PULSE_TYPE"][k] == 2.0:
        health["PULSE_TYPE"][k] = "IRREGULAR"

for z in range(0,len(health["DIABETES"])):
    if health["DIABETES"][z] == 1.0:
        health["DIABETES"][z] = "YES"
    if health["DIABETES"][z] == 2.0:
        health["DIABETES"][z] = "NO"
    if health["DIABETES"][z] == 3.0:
        health["DIABETES"][z] = "BORDERLINE"

ax = sns.countplot(x="RACE",data=health)

ax = sns.countplot(x="SEX",data=health)

ax = sns.countplot(x="AGE",data=health)

health.tail()
```

```
##            RACE     SEX   AGE  ...  BLD_PRE_S  BLD_PRE_D    DIABETES
## 8361  N_H_ASIAN  FEMALE  70.0  ...      142.0       78.0  BORDERLINE
## 8362   MEX_AMER    MALE  42.0  ...      122.0       76.0          NO
## 8363  N_H_BLACK  FEMALE  41.0  ...      118.0       72.0          NO
## 8364  N_H_BLACK  FEMALE  14.0  ...      114.0       60.0          NO
## 8365  N_H_WHITE    MALE  38.0  ...      146.0       92.0          NO
##
## [5 rows x 9 columns]
```

```
health.to_csv("health_data.csv", sep=',', index=False, encoding='utf-8')
```



1. Data description

1.1 Data import in R

```
# import the data in R
mydata <- read.csv("health_data.csv")
dim(mydata)
```

```
# [1] 8366    9
```

```
# rename columns
names(mydata) <- c("Race", "Sex", "Age", "BMI", "Pulse", "Pulse_type", "BPS", "BPD",
 "Diabetes")
names(mydata)
```

```
# [1] "Race"       "Sex"        "Age"        "BMI"        "Pulse"
# [6] "Pulse_type" "BPS"        "BPD"        "Diabetes"
```

```
# variable names
colnames(mydata)
```

```
# [1] "Race"       "Sex"        "Age"        "BMI"        "Pulse"
# [6] "Pulse_type" "BPS"        "BPD"        "Diabetes"
```

```
# Race
table(mydata$Race, useNA = "always")
```

```
#
#  HISPANIC  MEX_AMER N_H_ASIAN N_H_BLACK N_H_WHITE O_R_MULTI      <NA>
#       738      1229      1083      1949      2792       575         0
```

```
mydata$Race <- case_match(mydata$Race, "HISPANIC" ~ "Hispanic", "MEX_AMER" ~ "Mex_Amer",
 "N_H_ASIAN" ~ "Asian", "N_H_BLACK" ~ "Black", "N_H_WHITE" ~ "White", "O_R_MULTI" ~
     "Other")
print(prop.table(table(mydata$Race, useNA = "always")), digits = 1)
```

```
#
#    Asian    Black Hispanic Mex_Amer    Other    White     <NA>
#     0.13     0.23     0.09     0.15     0.07     0.33     0.00
```

```
mydata$Race <- factor(mydata$Race)
mydata$Race <- relevel(mydata$Race, ref = "White")
table(mydata$Race, useNA = "always")
```

```
#
#    White    Asian    Black Hispanic Mex_Amer    Other     <NA>
#     2792     1083     1949      738     1229      575        0
```

```
# Sex
table(mydata$Sex, useNA = "always")
```

```
#
# FEMALE   MALE   <NA>
#   4272   4094      0
```

```
mydata$Sex <- case_match(mydata$Sex, "FEMALE" ~ "Female", "MALE" ~ "Male")
print(prop.table(table(mydata$Sex, useNA = "always")), digits = 3)
```

```
#
# Female   Male   <NA>
#  0.511  0.489  0.000
```

```
# Diabetes
table(mydata$Diabetes, useNA = "always")
```

```
#
#        9.0 BORDERLINE         NO        YES       <NA>
#          4        175       7334        853          0
```

```
mydata$Diabetes <- case_match(mydata$Diabetes, "9.0" ~ NA, "BORDERLINE" ~ "Yes",
 "NO" ~ "No", "YES" ~ "Yes")
print(prop.table(table(mydata$Diabetes, useNA = "always")), digits = 5)
```

```
#
#         No        Yes       <NA>
# 0.87664356 0.12287832 0.00047813
```

```
# create binary Diabetes mydata\(Diabetes.b &lt;-
ifelse(mydata\)Diabetes == "Yes", 1, 0)
```

Pulse_type

```
table(mydata\(Pulse_type, useNA =
&amp;quot;always&amp;quot;)</code></pre>
<pre><code># # IRREGULAR REGULAR &amp;lt;NA&amp;gt; # 422
226 7718 0</code></pre> <pre
class="r"><code>mydata\)Pulse_type <-
case_match(mydata\(Pulse_type,
&amp;quot;&amp;quot; ~ NA, &amp;quot;IRREGULAR&amp;quot; ~
&amp;quot;Irregular&amp;quot;,  &amp;quot;REGULAR&amp;quot; ~
&amp;quot;Regular&amp;quot;)
print(prop.table(table(mydata\)Pulse_type, useNA = "always")),
digits = 5)
```

```
#
# Irregular   Regular      <NA>
#  0.027014  0.922544  0.050442
```

```
# Age
summary(mydata$Age)
```

```
#    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
#    1.00   12.00   33.00   35.83   59.00   80.00
```

```
# BMI
summary(mydata$BMI)
```

```
#    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
#   12.30   20.40   25.80   26.58   31.30   86.20     361
```

```
# Pulse
summary(mydata$Pulse)
```

```
#    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
#   34.00   66.00   72.00   73.75   82.00  136.00    1624
```

```
# BPS
summary(mydata$BPS)
```

```
#    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
#    72.0   106.0   118.0   121.6   132.0   236.0    1803
```

```
# BPD
summary(mydata$BPD)
```

```
#    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
#     0.0    60.0    70.0    68.3    78.0   136.0    1803
```

```
# subset participants with complete data
mydata <- subset(mydata, subset = complete.cases(mydata))
dim(mydata)
```

```
# [1] 6483   10
```

Elapsed time: 0.234 sec.

1.2 Table 1 stratified by Sex

```
# table one
allvars <- names(mydata)
contvars <- c("Age", "BMI", "Pulse", "BPS", "BPD")
catvars <- setdiff(allvars, contvars)
catvars0 <- setdiff(catvars, "Sex")

tableone <- CreateTableOne(vars = allvars, strata = "Sex", addOverall = TRUE, factorVars = catvars0,
    data = mydata)
kableone(tableone)
```

	level	Overall	Female	Male	p	test
n		6483	3315	3168		
Race (%)	White	2168 (33.4)	1071 (32.3)	1097 (34.6)	0.004	
	Asian	867 (13.4)	450 (13.6)	417 (13.2)		
	Black	1521 (23.5)	815 (24.6)	706 (22.3)		
	Hispanic	568 (8.8)	297 (9.0)	271 (8.6)		
	Mex_Amer	937 (14.5)	497 (15.0)	440 (13.9)		
	Other	422 (6.5)	185 (5.6)	237 (7.5)		
Sex (%)	Female	3315 (51.1)	3315 (100.0)	0 (0.0)	<0.001	
	Male	3168 (48.9)	0 (0.0)	3168 (100.0)		
Age (mean (SD))		41.50 (22.59)	41.46 (22.29)	41.54 (22.90)	0.897	
BMI (mean (SD))		28.01 (7.75)	28.45 (8.30)	27.54 (7.11)	<0.001	
Pulse (mean (SD))		73.66 (12.27)	74.90 (12.05)	72.37 (12.37)	<0.001	
Pulse_type (%)	Irregular	196 (3.0)	79 (2.4)	117 (3.7)	0.003	
	Regular	6287 (97.0)	3236 (97.6)	3051 (96.3)		
BPS (mean (SD))		121.45 (20.26)	120.50 (21.24)	122.44 (19.13)	<0.001	
BPD (mean (SD))		68.25 (16.13)	67.49 (15.13)	69.05 (17.07)	<0.001	
Diabetes (%)	No	5542 (85.5)	2887 (87.1)	2655 (83.8)	<0.001	
	Yes	941 (14.5)	428 (12.9)	513 (16.2)		
Diabetes.b (%)	0	5542 (85.5)	2887 (87.1)	2655 (83.8)	<0.001	
	1	941 (14.5)	428 (12.9)	513 (16.2)		

Elapsed time: 0.444 sec.

1.3 Table One stratified by Sex

```
catvars0 <- setdiff(catvars, "Race")

tableone <- CreateTableOne(vars = allvars, strata = "Race", addOverall = TRUE, factorVars = catvars
0,
    data = mydata)
kableone(tableone)
```

	level	Overall	White	Asian	Black	Hispanic	Mex_Amer	Other	p	test
n		6483	2168	867	1521	568	937	422		
Race (%)	White	2168 (33.4)	2168 (100.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	<0.001	
	Asian	867 (13.4)	0 (0.0)	867 (100.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)		
	Black	1521 (23.5)	0 (0.0)	0 (0.0)	1521 (100.0)	0 (0.0)	0 (0.0)	0 (0.0)		
	Hispanic	568 (8.8)	0 (0.0)	0 (0.0)	0 (0.0)	568 (100.0)	0 (0.0)	0 (0.0)		
	Mex_Amer	937 (14.5)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	937 (100.0)	0 (0.0)		
	Other	422 (6.5)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	0 (0.0)	422 (100.0)		
Sex (%)	Female	3315 (51.1)	1071 (49.4)	450 (51.9)	815 (53.6)	297 (52.3)	497 (53.0)	185 (43.8)	0.004	
	Male	3168 (48.9)	1097 (50.6)	417 (48.1)	706 (46.4)	271 (47.7)	440 (47.0)	237 (56.2)		
Age (mean (SD))		41.50 (22.59)	45.13 (23.99)	41.89 (20.03)	41.34 (22.37)	42.20 (21.41)	36.23 (21.04)	33.40 (21.56)	<0.001	
BMI (mean (SD))		28.01 (7.75)	28.25 (7.96)	25.14 (5.37)	29.13 (8.78)	28.08 (6.49)	28.56 (7.09)	27.25 (8.37)	<0.001	
Pulse (mean (SD))		73.66 (12.27)	73.61 (12.31)	72.79 (11.74)	73.02 (12.14)	73.25 (12.25)	74.29 (12.08)	77.16 (13.42)	<0.001	
Pulse_type (%)	Irregular	196 (3.0)	94 (4.3)	14 (1.6)	56 (3.7)	14 (2.5)	7 (0.7)	11 (2.6)	<0.001	
	Regular	6287 (97.0)	2074 (95.7)	853 (98.4)	1465 (96.3)	554 (97.5)	930 (99.3)	411 (97.4)		
BPS (mean (SD))		121.45 (20.26)	121.49 (19.43)	120.64 (19.47)	125.02 (22.28)	120.93 (21.12)	118.44 (18.58)	117.37 (18.80)	<0.001	
BPD (mean (SD))		68.25 (16.13)	67.69 (15.09)	70.17 (14.72)	69.49 (17.82)	68.14 (14.75)	66.61 (16.30)	66.52 (18.26)	<0.001	
Diabetes (%)	No	5542 (85.5)	1866 (86.1)	722 (83.3)	1282 (84.3)	493 (86.8)	805 (85.9)	374 (88.6)	0.074	

	level	Overall	White	Asian	Black	Hispanic	Mex_Amer	Other	p	test
	Yes	941 (14.5)	302 (13.9)	145 (16.7)	239 (15.7)	75 (13.2)	132 (14.1)	48 (11.4)		
Diabetes.b (%)	0	5542 (85.5)	1866 (86.1)	722 (83.3)	1282 (84.3)	493 (86.8)	805 (85.9)	374 (88.6)	0.074	
	1	941 (14.5)	302 (13.9)	145 (16.7)	239 (15.7)	75 (13.2)	132 (14.1)	48 (11.4)		

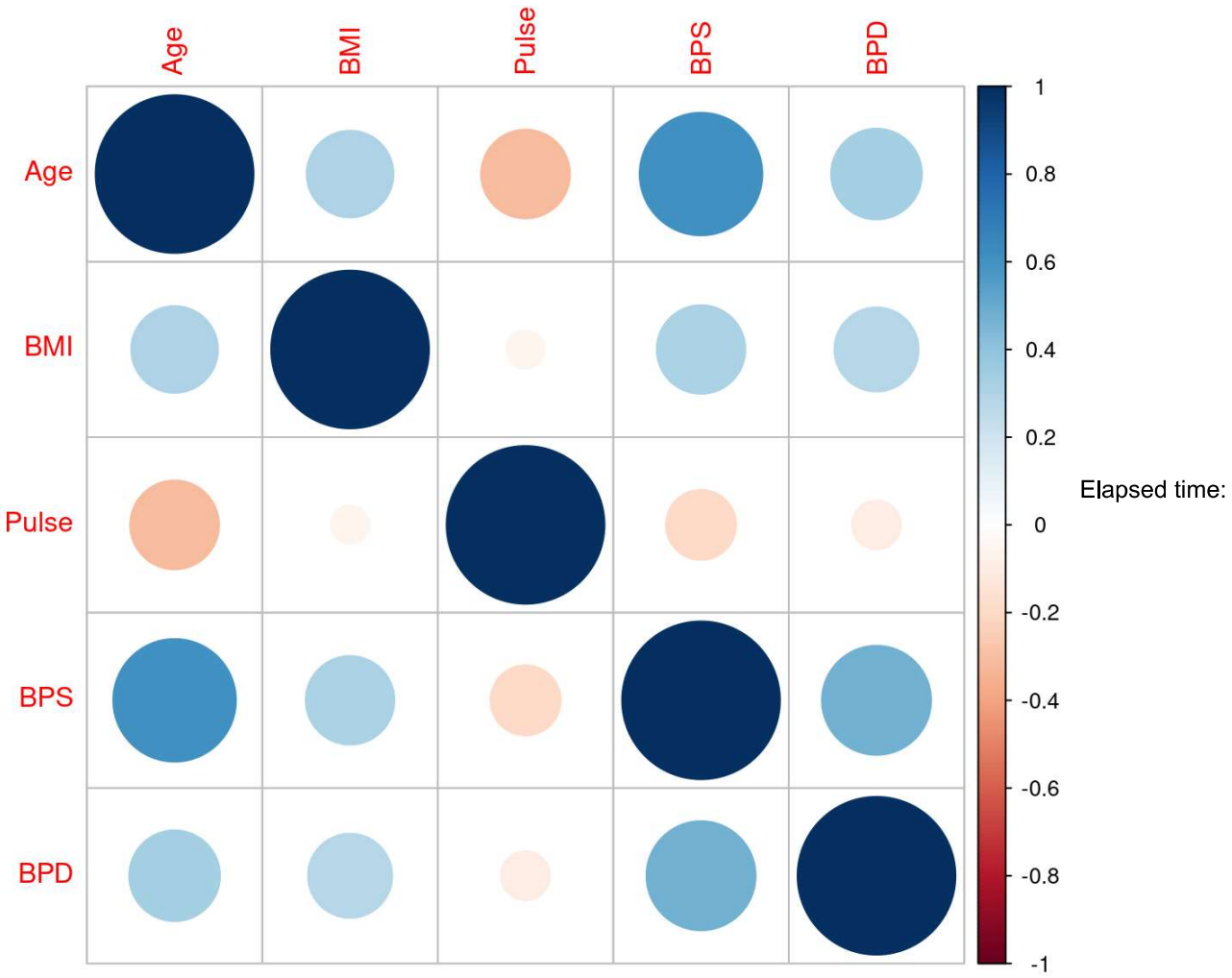
Elapsed time: 0.195 sec.

1.4 Correlation analysis

```
# overall correlation matrix
rcorr(as.matrix(mydata[, contvars]))
```

```
#         Age   BMI Pulse   BPS   BPD
# Age    1.00  0.30 -0.32  0.60  0.33
# BMI    0.30  1.00 -0.06  0.32  0.29
# Pulse -0.32 -0.06  1.00 -0.20 -0.10
# BPS    0.60  0.32 -0.20  1.00  0.48
# BPD    0.33  0.29 -0.10  0.48  1.00
#
# n= 6483
#
#
# P
#       Age BMI Pulse BPS BPD
# Age         0   0     0   0
# BMI    0        0     0   0
# Pulse  0   0         0   0
# BPS    0   0   0         0
# BPD    0   0   0     0
```

```
corrplot(rcorr(as.matrix(mydata[, contvars]))$r)
```



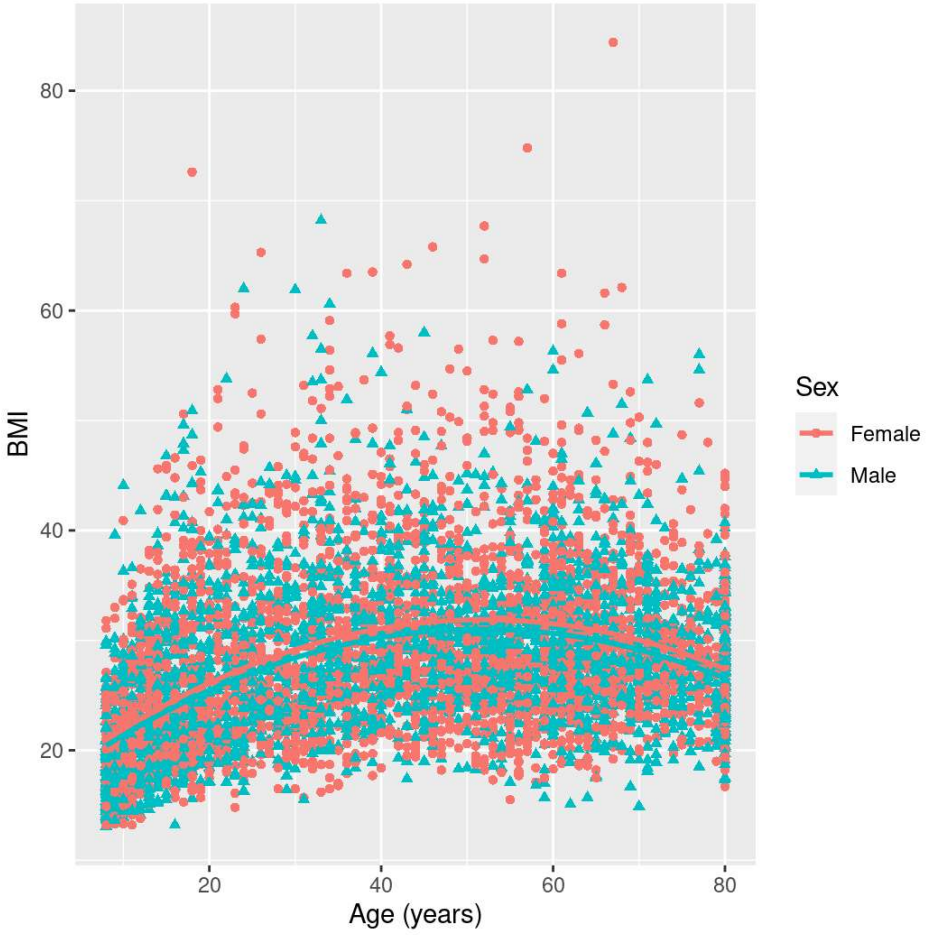
Elapsed time:

0.153 sec.

1.5 Graphical summaries

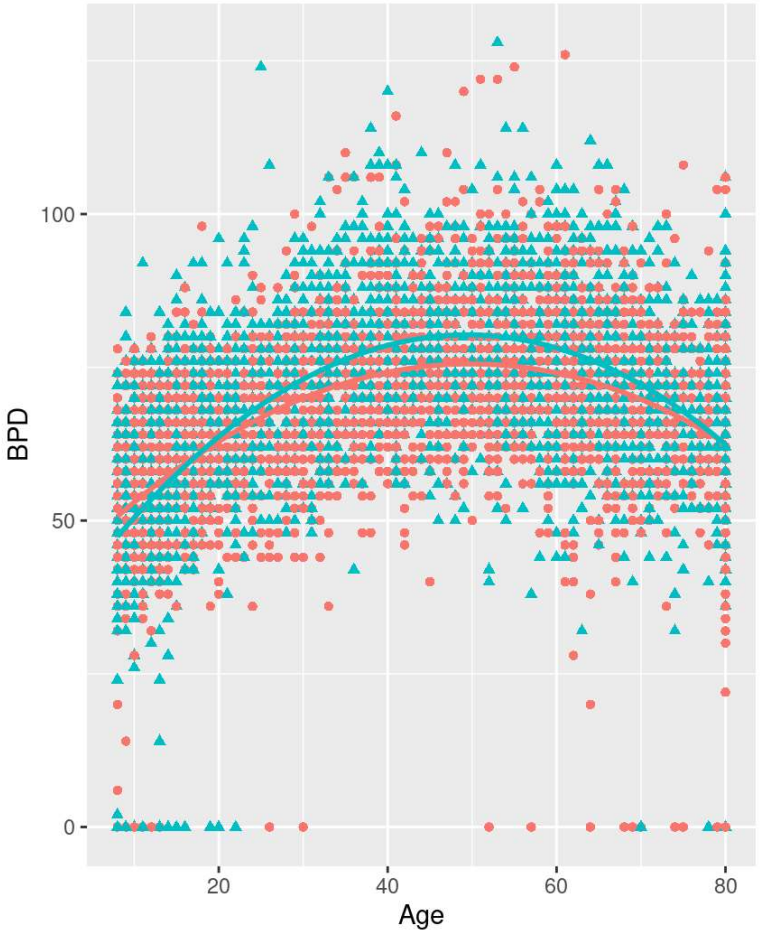
1.5a Graphical summaries for BPD vs covariates stratified by sex

```
# plot BMI vs Age
ggplot(mydata, aes(y = BMI, x = Age, color = Sex, shape = Sex)) + geom_point() +
    geom_smooth(method = "lm", fill = NA, formula = y ~ x + I(x^2)) + xlab("Age (years)") +
    ylab("BMI")
```




```
# plot BPS vs Age
ggplot(mydata, aes(y = BPS, x = Age, color = Sex, shape = Sex)) + geom_point() +
    geom_smooth(method = "lm", fill = NA, formula = y ~ x + I(x^2)) + xlab("Age") +
    ylab("BPS")
```


```
# plot BPD vs Age
ggplot(mydata, aes(y = BPD, x = Age, color = Sex, shape = Sex)) + geom_point() +
    geom_smooth(method = "lm", fill = NA, formula = y ~ x + I(x^2)) + xlab("Age") +
    ylab("BPD")
```



Elapsed time: 1.335 sec.

2. Linear Regression Models

2.1 Regression models for BPS vs BMI, Sex, Race, Pulse

```
# center the predictors to interpret the intercepts
mydata$Age.c <- scale(mydata$Age, scale = FALSE)
summary(mydata$Age.c)
```

```
#        V1
#  Min.   :-33.5002
#  1st Qu.:-22.5002
#  Median : -0.5002
#  Mean   :  0.0000
#  3rd Qu.: 19.4998
#  Max.   : 38.4998
```

```
mydata$BMI.c <- scale(mydata$BMI, scale = FALSE)
summary(mydata$BMI.c)
```

```
#        V1
#  Min.   :-14.906
#  1st Qu.: -5.406
#  Median : -1.006
#  Mean   :  0.000
#  3rd Qu.:  4.194
#  Max.   : 56.394
```

```
mydata$Pulse.c <- scale(mydata$Pulse, scale = FALSE)
summary(mydata$Pulse.c)
```

```
#        V1
#  Min.   :-39.662
#  1st Qu.: -7.662
#  Median : -1.662
#  Mean   :  0.000
#  3rd Qu.:  8.338
#  Max.   : 60.338
```

```
# null model
lm.out <- lm(BPS ~ Age.c + BMI.c + Sex + Race + Pulse.c, data = mydata)
summary(lm.out)
```

```
#
# Call:
# lm(formula = BPS ~ Age.c + BMI.c + Sex + Race + Pulse.c, data = mydata)
#
# Residuals:
#     Min      1Q  Median      3Q     Max
# -62.941  -9.764  -1.291   7.923 107.498
#
# Coefficients:
#                Estimate Std. Error t value Pr(>|t|)
# (Intercept)  118.378893   0.394642 299.965  < 2e-16 ***
# Age.c          0.506055   0.009706  52.139  < 2e-16 ***
# BMI.c          0.376889   0.026933  13.993  < 2e-16 ***
# SexMale        2.339408   0.394887   5.924  3.3e-09 ***
# RaceAsian      2.017654   0.638278   3.161  0.00158 **
# RaceBlack      5.207978   0.529609   9.834  < 2e-16 ***
# RaceHispanic   1.052296   0.742961   1.416  0.15672
# RaceMex_Amer   1.424473   0.621855   2.291  0.02201 *
# RaceOther      2.083685   0.844925   2.466  0.01368 *
# Pulse.c       -0.004538   0.016962  -0.268  0.78906
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 15.75 on 6473 degrees of freedom
# Multiple R-squared:  0.3968,  Adjusted R-squared:  0.3959
# F-statistic: 473.1 on 9 and 6473 DF,  p-value: < 2.2e-16
```

```
# fit best subset regression
lm.out <- regsubsets(BPS ~ Age.c + I(Age.c * Age.c) + Pulse.c + I(Pulse.c * Pulse.c) +
 I(Age.c * Pulse.c) + BMI.c + I(BMI.c * BMI.c) + I(BMI.c * Age.c) + I(BMI.c *
 Pulse.c) + Race + Race:BMI.c + Race:Pulse.c + Race:Age.c + Sex + Sex:BMI.c +
 Sex:Pulse.c + Sex:Age.c, data = mydata)
summary(lm.out)$adjr2
```

```
# [1] 0.3647317 0.3840893 0.3918056 0.3997172 0.4033164 0.4076977 0.4111785
# [8] 0.4126856
```

```
summary(lm.out)$which[summary(lm.out)$adjr2 == max(summary(lm.out)$adjr2), ]
```

```
#          (Intercept)                Age.c     I(Age.c * Age.c)
#                 TRUE                 TRUE                FALSE
#              Pulse.c I(Pulse.c * Pulse.c)   I(Age.c * Pulse.c)
#                FALSE                FALSE                 TRUE
#                BMI.c     I(BMI.c * BMI.c)     I(BMI.c * Age.c)
#                 TRUE                FALSE                 TRUE
#   I(BMI.c * Pulse.c)            RaceAsian            RaceBlack
#                FALSE                FALSE                 TRUE
#         RaceHispanic         RaceMex_Amer            RaceOther
#                FALSE                FALSE                FALSE
#              SexMale      BMI.c:RaceAsian      BMI.c:RaceBlack
#                 TRUE                FALSE                FALSE
#   BMI.c:RaceHispanic   BMI.c:RaceMex_Amer      BMI.c:RaceOther
#                FALSE                FALSE                FALSE
#    Pulse.c:RaceAsian    Pulse.c:RaceBlack Pulse.c:RaceHispanic
#                FALSE                FALSE                FALSE
# Pulse.c:RaceMex_Amer    Pulse.c:RaceOther      Age.c:RaceAsian
#                FALSE                FALSE                FALSE
#      Age.c:RaceBlack   Age.c:RaceHispanic   Age.c:RaceMex_Amer
#                 TRUE                FALSE                FALSE
#      Age.c:RaceOther        BMI.c:SexMale      Pulse.c:SexMale
#                FALSE                FALSE                FALSE
#        Age.c:SexMale
#                 TRUE
```

```
myind <- summary(lm.out)$which[summary(lm.out)$adjr2 == max(summary(lm.out)$adjr2),
 ][-1]
names(summary(lm.out)$which[summary(lm.out)$adjr2 == max(summary(lm.out)$adjr2),
 ])[-1][myind]
```

```
# [1] "Age.c"              "I(Age.c * Pulse.c)" "BMI.c"
# [4] "I(BMI.c * Age.c)"   "RaceBlack"          "SexMale"
# [7] "Age.c:RaceBlack"    "Age.c:SexMale"
```

```
# final model
lm.out <- lm(BPS ~ Age.c + Pulse.c + I(Age.c * Pulse.c) + BMI.c + I(BMI.c * Age.c) +
 Race + Sex + Sex:Age.c + Race:Age.c, data = mydata)
summary(lm.out)
```

```
#
# Call:
# lm(formula = BPS ~ Age.c + Pulse.c + I(Age.c * Pulse.c) + BMI.c +
#     I(BMI.c * Age.c) + Race + Sex + Sex:Age.c + Race:Age.c, data = mydata)
#
# Residuals:
#     Min      1Q  Median      3Q     Max
# -61.210  -9.499  -1.217   8.022 108.930
#
# Coefficients:
#                       Estimate Std. Error t value Pr(>|t|)
# (Intercept)        118.579367   0.402613 294.524  < 2e-16 ***
# Age.c                0.527104   0.017014  30.982  < 2e-16 ***
# Pulse.c             -0.003573   0.016877  -0.212  0.83234
# I(Age.c * Pulse.c)  -0.005192   0.000725  -7.161 8.88e-13 ***
# BMI.c                0.373652   0.027003  13.837  < 2e-16 ***
# I(BMI.c * Age.c)    -0.008479   0.001265  -6.701 2.25e-11 ***
# RaceAsian            1.490792   0.632167   2.358  0.01839 *
# RaceBlack            5.175962   0.523009   9.897  < 2e-16 ***
# RaceHispanic         0.696200   0.733323   0.949  0.34246
# RaceMex_Amer         1.125420   0.623170   1.806  0.07097 .
# RaceOther            1.470021   0.875062   1.680  0.09302 .
# SexMale              2.439103   0.389245   6.266 3.94e-10 ***
# Age.c:SexMale       -0.176397   0.017262 -10.219  < 2e-16 ***
# Age.c:RaceAsian      0.087813   0.030002   2.927  0.00344 **
# Age.c:RaceBlack      0.117328   0.022605   5.190 2.16e-07 ***
# Age.c:RaceHispanic   0.099898   0.033465   2.985  0.00284 **
# Age.c:RaceMex_Amer   0.062889   0.027888   2.255  0.02416 *
# Age.c:RaceOther      0.036707   0.037875   0.969  0.33250
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 15.51 on 6465 degrees of freedom
# Multiple R-squared:  0.4157,  Adjusted R-squared:  0.4142
# F-statistic: 270.6 on 17 and 6465 DF,  p-value: < 2.2e-16
```

```
# train data
tprob <- 0.7
tind <- rbinom(nrow(mydata), 1, tprob)
tdata <- subset(mydata, subset = tind == 1)
dim(tdata)
```

```
# [1] 4549   13
```

```
vdata <- subset(mydata, subset = tind == 0)
dim(vdata)
```

```
# [1] 1934   13
```

```
# fit the model on the training data
lm.out <- lm(BPS ~ Age.c + Pulse.c + I(Age.c * Pulse.c) + BMI.c + I(BMI.c * Age.c) +
 Race + Sex + Sex:Age.c + Race:Age.c, data = tdata)
summary(lm.out)
```

```
#
# Call:
# lm(formula = BPS ~ Age.c + Pulse.c + I(Age.c * Pulse.c) + BMI.c +
#     I(BMI.c * Age.c) + Race + Sex + Sex:Age.c + Race:Age.c, data = tdata)
#
# Residuals:
#     Min      1Q  Median      3Q     Max
# -61.461  -9.267  -1.129   7.706 108.970
#
# Coefficients:
#                       Estimate Std. Error t value Pr(>|t|)
# (Intercept)         1.184e+02  4.713e-01 251.209  < 2e-16 ***
# Age.c               5.079e-01  2.005e-02  25.324  < 2e-16 ***
# Pulse.c             5.738e-04  1.974e-02   0.029  0.97681
# I(Age.c * Pulse.c) -5.414e-03  8.496e-04  -6.372 2.05e-10 ***
# BMI.c               3.814e-01  3.118e-02  12.230  < 2e-16 ***
# I(BMI.c * Age.c)   -7.762e-03  1.462e-03  -5.309 1.15e-07 ***
# RaceAsian           1.838e+00  7.394e-01   2.485  0.01298 *
# RaceBlack           5.111e+00  6.069e-01   8.421  < 2e-16 ***
# RaceHispanic        1.108e+00  8.400e-01   1.319  0.18729
# RaceMex_Amer        1.061e+00  7.294e-01   1.454  0.14596
# RaceOther           1.574e+00  1.029e+00   1.530  0.12618
# SexMale             2.566e+00  4.528e-01   5.667 1.54e-08 ***
# Age.c:SexMale      -1.609e-01  2.017e-02  -7.977 1.89e-15 ***
# Age.c:RaceAsian     8.299e-02  3.523e-02   2.355  0.01854 *
# Age.c:RaceBlack     1.403e-01  2.620e-02   5.354 9.01e-08 ***
# Age.c:RaceHispanic  1.251e-01  3.865e-02   3.237  0.00122 **
# Age.c:RaceMex_Amer  5.751e-02  3.272e-02   1.758  0.07888 .
# Age.c:RaceOther     5.892e-02  4.598e-02   1.282  0.20003
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 15.11 on 4531 degrees of freedom
# Multiple R-squared:  0.4244,  Adjusted R-squared:  0.4223
# F-statistic: 196.5 on 17 and 4531 DF,  p-value: < 2.2e-16
```

```
# predict BPS on vdata
vpred <- predict(lm.out, newdata = vdata)
1 - sum((vpred - vdata$BPS)^2)/sum((vdata$BPS - mean(vdata$BPS))^2)
```

```
# [1] 0.3962873
```

Elapsed time: 0.401 sec.

2.2 Regression models for BPD vs BMI, Sex, Race, Pulse

```
# null model
lm.out <- lm(BPD ~ Age.c + BMI.c + Sex + Race + Pulse.c, data = mydata)
summary(lm.out)
```

```
#
# Call:
# lm(formula = BPD ~ Age.c + BMI.c + Sex + Race + Pulse.c, data = mydata)
#
# Residuals:
#     Min      1Q  Median      3Q     Max
# -80.907  -7.186   1.287   8.758  58.385
#
# Coefficients:
#               Estimate Std. Error t value Pr(>|t|)
# (Intercept)  65.837534   0.370762 177.574  < 2e-16 ***
# Age.c         0.194272   0.009119  21.305  < 2e-16 ***
# BMI.c         0.450530   0.025304  17.805  < 2e-16 ***
# SexMale       2.051641   0.370992   5.530 3.32e-08 ***
# RaceAsian     4.570411   0.599654   7.622 2.86e-14 ***
# RaceBlack     2.230553   0.497561   4.483 7.49e-06 ***
# RaceHispanic  1.159264   0.698003   1.661   0.0968 .
# RaceMex_Amer  0.578018   0.584225   0.989   0.3225
# RaceOther     1.389544   0.793797   1.751   0.0801 .
# Pulse.c       0.014950   0.015935   0.938   0.3482
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 14.79 on 6473 degrees of freedom
# Multiple R-squared:  0.1598,  Adjusted R-squared:  0.1586
# F-statistic: 136.8 on 9 and 6473 DF,  p-value: < 2.2e-16
```

```
# fit best subset regression
lm.out <- regsubsets(BPD ~ Age.c + I(Age.c * Age.c) + Pulse.c + I(Pulse.c * Pulse.c) +
 I(Age.c * Pulse.c) + BMI.c + I(BMI.c * BMI.c) + I(BMI.c * Age.c) + I(BMI.c *
 Pulse.c) + Race + Race:BMI.c + Race:Pulse.c + Race:Age.c + Sex + Sex:BMI.c +
 Sex:Pulse.c + Sex:Age.c, data = mydata)
summary(lm.out)$adjr2
```

```
# [1] 0.1568685 0.2985545 0.3044754 0.3086108 0.3118663 0.3145876 0.3166483
# [8] 0.3186444
```

```
summary(lm.out)$which[summary(lm.out)$adjr2 == max(summary(lm.out)$adjr2), ]
```

```
#          (Intercept)                Age.c     I(Age.c * Age.c)
#                 TRUE                 TRUE                 TRUE
#              Pulse.c I(Pulse.c * Pulse.c)   I(Age.c * Pulse.c)
#                 TRUE                 TRUE                FALSE
#                BMI.c     I(BMI.c * BMI.c)     I(BMI.c * Age.c)
#                FALSE                FALSE                 TRUE
#   I(BMI.c * Pulse.c)            RaceAsian            RaceBlack
#                FALSE                FALSE                FALSE
#         RaceHispanic         RaceMex_Amer            RaceOther
#                FALSE                FALSE                FALSE
#              SexMale      BMI.c:RaceAsian      BMI.c:RaceBlack
#                 TRUE                FALSE                FALSE
#   BMI.c:RaceHispanic   BMI.c:RaceMex_Amer      BMI.c:RaceOther
#                FALSE                FALSE                FALSE
#    Pulse.c:RaceAsian    Pulse.c:RaceBlack Pulse.c:RaceHispanic
#                FALSE                FALSE                FALSE
# Pulse.c:RaceMex_Amer    Pulse.c:RaceOther      Age.c:RaceAsian
#                FALSE                FALSE                FALSE
#      Age.c:RaceBlack   Age.c:RaceHispanic   Age.c:RaceMex_Amer
#                 TRUE                FALSE                FALSE
#      Age.c:RaceOther        BMI.c:SexMale      Pulse.c:SexMale
#                FALSE                 TRUE                FALSE
#        Age.c:SexMale
#                FALSE
```

```
myind <- summary(lm.out)$which[summary(lm.out)$adjr2 == max(summary(lm.out)$adjr2),
 ][-1]
names(summary(lm.out)$which[summary(lm.out)$adjr2 == max(summary(lm.out)$adjr2),
 ])[-1][myind]
```

```
# [1] "Age.c"                "I(Age.c * Age.c)"     "Pulse.c"
# [4] "I(Pulse.c * Pulse.c)" "I(BMI.c * Age.c)"     "SexMale"
# [7] "Age.c:RaceBlack"      "BMI.c:SexMale"
```

```
# final model
lm.out <- lm(BPD ~ Age.c + I(Age.c * Age.c) + Pulse.c + I(Pulse.c * Pulse.c) + BMI.c +
 I(BMI.c * Age.c) + Race + Sex + Sex:BMI.c + Race:Age.c, data = mydata)
summary(lm.out)
```

```
#
# Call:
# lm(formula = BPD ~ Age.c + I(Age.c * Age.c) + Pulse.c + I(Pulse.c *
#     Pulse.c) + BMI.c + I(BMI.c * Age.c) + Race + Sex + Sex:BMI.c +
#     Race:Age.c, data = mydata)
#
# Residuals:
#     Min      1Q  Median      3Q     Max
# -77.292  -6.465   0.594   7.811  48.921
#
# Coefficients:
#                        Estimate Std. Error t value Pr(>|t|)
# (Intercept)          75.4255525  0.4231247 178.258  < 2e-16 ***
# Age.c                 0.2367206  0.0131487  18.003  < 2e-16 ***
# I(Age.c * Age.c)     -0.0151107  0.0004494 -33.624  < 2e-16 ***
# Pulse.c               0.0971719  0.0153718   6.321 2.76e-10 ***
# I(Pulse.c * Pulse.c) -0.0033763  0.0007332  -4.605 4.21e-06 ***
# BMI.c                 0.0315252  0.0298350   1.057 0.290710
# I(BMI.c * Age.c)     -0.0071074  0.0011218  -6.336 2.52e-10 ***
# RaceAsian             0.6678406  0.5483204   1.218 0.223278
# RaceBlack             1.5748381  0.4491586   3.506 0.000458 ***
# RaceHispanic         -0.7057529  0.6301687  -1.120 0.262780
# RaceMex_Amer         -0.7626223  0.5354189  -1.424 0.154394
# RaceOther             0.4634316  0.7506855   0.617 0.537029
# SexMale               2.3589981  0.3341226   7.060 1.84e-12 ***
# BMI.c:SexMale         0.2226910  0.0437811   5.086 3.75e-07 ***
# Age.c:RaceAsian      -0.0193001  0.0257882  -0.748 0.454242
# Age.c:RaceBlack       0.0778880  0.0195608   3.982 6.91e-05 ***
# Age.c:RaceHispanic    0.0159378  0.0288332   0.553 0.580446
# Age.c:RaceMex_Amer    0.0067622  0.0243268   0.278 0.781041
# Age.c:RaceOther      -0.0158530  0.0326638  -0.485 0.627453
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 13.29 on 6464 degrees of freedom
# Multiple R-squared:  0.3225,  Adjusted R-squared:  0.3206
# F-statistic: 170.9 on 18 and 6464 DF,  p-value: < 2.2e-16
```

```
# fit the model on the training data
lm.out <- lm(BPD ~ Age.c + I(Age.c * Age.c) + Pulse.c + I(Pulse.c * Pulse.c) + BMI.c +
 I(BMI.c * Age.c) + Race + Sex + Sex:BMI.c + Race:Age.c, data = tdata)
summary(lm.out)
```

```
#
# Call:
# lm(formula = BPD ~ Age.c + I(Age.c * Age.c) + Pulse.c + I(Pulse.c *
#     Pulse.c) + BMI.c + I(BMI.c * Age.c) + Race + Sex + Sex:BMI.c +
#     Race:Age.c, data = tdata)
#
# Residuals:
#     Min      1Q  Median      3Q     Max
# -76.927  -6.233   0.642   7.770  48.799
#
# Coefficients:
#                        Estimate Std. Error t value Pr(>|t|)
# (Intercept)          75.6413224  0.5090609 148.590  < 2e-16 ***
# Age.c                 0.2370207  0.0157241  15.074  < 2e-16 ***
# I(Age.c * Age.c)     -0.0155606  0.0005390 -28.870  < 2e-16 ***
# Pulse.c               0.1026763  0.0185030   5.549 3.03e-08 ***
# I(Pulse.c * Pulse.c) -0.0036199  0.0008715  -4.154 3.33e-05 ***
# BMI.c                 0.0157094  0.0353571   0.444    0.657
# I(BMI.c * Age.c)     -0.0064249  0.0013305  -4.829 1.42e-06 ***
# RaceAsian             0.6314357  0.6593782   0.958    0.338
# RaceBlack             1.3790735  0.5353629   2.576    0.010 *
# RaceHispanic         -0.2435242  0.7423448  -0.328    0.743
# RaceMex_Amer         -0.7607116  0.6440205  -1.181    0.238
# RaceOther            -0.4734782  0.9076483  -0.522    0.602
# SexMale               2.6093772  0.3992710   6.535 7.05e-11 ***
# BMI.c:SexMale         0.2200481  0.0517517   4.252 2.16e-05 ***
# Age.c:RaceAsian      -0.0328777  0.0311299  -1.056    0.291
# Age.c:RaceBlack       0.1066730  0.0232655   4.585 4.66e-06 ***
# Age.c:RaceHispanic    0.0404095  0.0342232   1.181    0.238
# Age.c:RaceMex_Amer    0.0235303  0.0293413   0.802    0.423
# Age.c:RaceOther      -0.0088336  0.0408053  -0.216    0.829
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# Residual standard error: 13.31 on 4530 degrees of freedom
# Multiple R-squared:  0.3312,  Adjusted R-squared:  0.3286
# F-statistic: 124.6 on 18 and 4530 DF,  p-value: < 2.2e-16
```

```
# predict BPS on vdata
vpred <- predict(lm.out, newdata = vdata)
1 - sum((vpred - vdata$BPD)^2)/sum((vdata$BPD - mean(vdata$BPD))^2)
```

```
# [1] 0.2971578
```

Elapsed time: 0.077 sec.

2.3 Regression models for Diabetes vs BMI, Sex, Race, Pulse

```
# null model
glm.out <- glm(Diabetes.b ~ Age.c + BMI.c + Sex + Race + Pulse.c + BPS + BPD, data = mydata,
 family = binomial)
summary(glm.out)
```

```
#
# Call:
# glm(formula = Diabetes.b ~ Age.c + BMI.c + Sex + Race + Pulse.c +
#     BPS + BPD, family = binomial, data = mydata)
#
# Coefficients:
#               Estimate Std. Error z value Pr(>|z|)
# (Intercept)  -2.448673   0.290605  -8.426  < 2e-16 ***
# Age.c         0.063556   0.002754  23.079  < 2e-16 ***
# BMI.c         0.074735   0.005449  13.715  < 2e-16 ***
# SexMale       0.505816   0.081317   6.220 4.96e-10 ***
# RaceAsian     1.068375   0.129198   8.269  < 2e-16 ***
# RaceBlack     0.474226   0.108726   4.362 1.29e-05 ***
# RaceHispanic  0.323439   0.154182   2.098   0.0359 *
# RaceMex_Amer  0.722571   0.130151   5.552 2.83e-08 ***
# RaceOther     0.452073   0.189834   2.381   0.0172 *
# Pulse.c       0.029373   0.003491   8.415  < 2e-16 ***
# BPS           0.002250   0.002295   0.981   0.3268
# BPD          -0.014810   0.003048  -4.859 1.18e-06 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# (Dispersion parameter for binomial family taken to be 1)
#
#     Null deviance: 5370.5  on 6482  degrees of freedom
# Residual deviance: 4121.8  on 6471  degrees of freedom
# AIC: 4145.8
#
# Number of Fisher Scoring iterations: 6
```

```
mytab <- data.frame(OR = exp(summary(glm.out)$coef[, 1]), SE = exp(summary(glm.out)$coef[,
 1]) * summary(glm.out)$coef[, 2], LCL = exp(summary(glm.out)$coef[, 1]) - 1.96 *
 exp(summary(glm.out)$coef[, 1]) * summary(glm.out)$coef[, 2], UCL = exp(summary(glm.out)$coef[,
 1]) - 1.96 * exp(summary(glm.out)$coef[, 1]) * summary(glm.out)$coef[, 2])
kable(mytab, digits = 3)
```

	OR	SE	LCL	UCL
(Intercept)	0.086	0.025	0.037	0.037
Age.c	1.066	0.003	1.060	1.060
BMI.c	1.078	0.006	1.066	1.066
SexMale	1.658	0.135	1.394	1.394
RaceAsian	2.911	0.376	2.174	2.174
RaceBlack	1.607	0.175	1.264	1.264

	OR	SE	LCL	UCL
RaceHispanic	1.382	0.213	0.964	0.964
RaceMex_Amer	2.060	0.268	1.534	1.534
RaceOther	1.572	0.298	0.987	0.987
Pulse.c	1.030	0.004	1.023	1.023
BPS	1.002	0.002	0.998	0.998
BPD	0.985	0.003	0.979	0.979

```
# accuracy
glm.pred <- predict(glm.out, type = "response")
glm.predb <- ifelse(glm.pred >= 0.13, 1, 0)
table(glm.predb, mydata$Diabetes.b)
```

```
#
# glm.predb    0    1
#         0 3788  143
#         1 1754  798
```

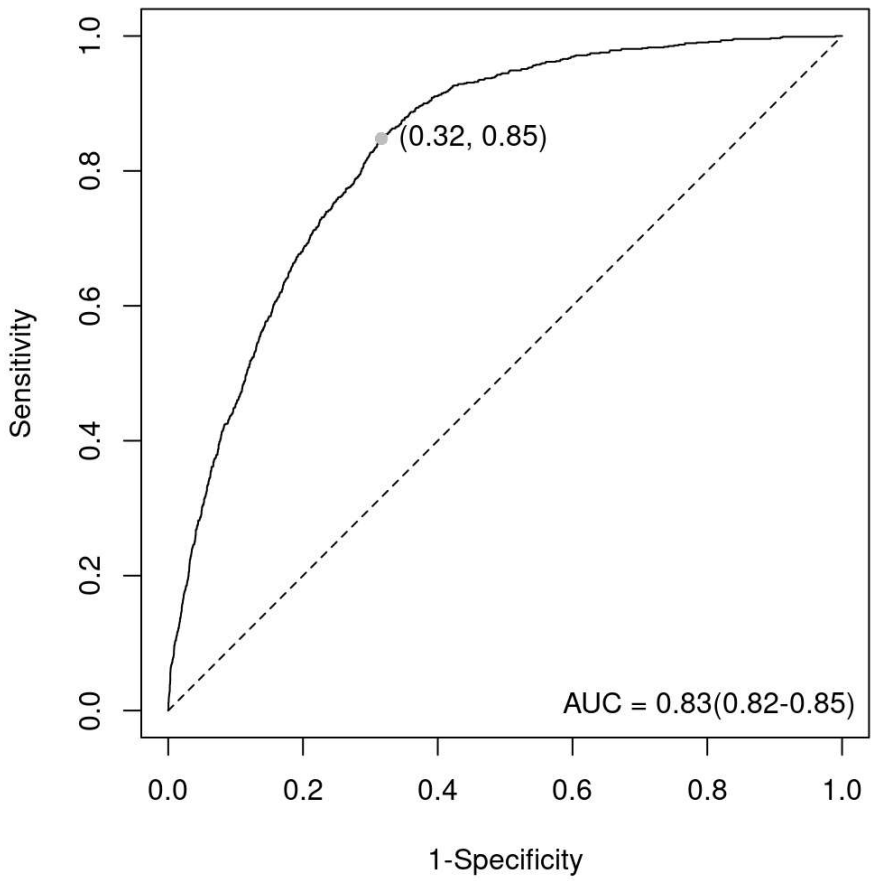
```
prop.table(table(glm.predb, mydata$Diabetes.b))
```

```
#
# glm.predb          0          1
#         0 0.58429739 0.02205769
#         1 0.27055376 0.12309116
```

```
prop.table(table(glm.predb, mydata$Diabetes.b), margin = 2)
```

```
#
# glm.predb         0         1
#         0 0.6835078 0.1519660
#         1 0.3164922 0.8480340
```

```
# ROC
reportROC(mydata$Diabetes.b, glm.pred)
```




```
#  Cutoff   AUC AUC.SE AUC.low AUC.up     P   ACC ACC.low ACC.up   SEN SEN.low
#   0.130 0.833  0.006   0.821  0.846 0.000 0.707   0.707  0.707 0.848   0.825
#  SEN.up   SPE SPE.low SPE.up   PLR PLR.low PLR.up   NLR NLR.low NLR.up   PPV
#   0.871 0.684   0.671  0.696 2.679   2.556  2.809 0.222   0.191  0.259 0.313
#  PPV.low PPV.up   NPV NPV.low NPV.up   PPA PPA.low PPA.up   NPA NPA.low NPA.up
#    0.295  0.331 0.964   0.958  0.969 0.848   0.825  0.871 0.684   0.671  0.696
#    TPA TPA.low TPA.up KAPPA KAPPA.low KAPPA.up
#  0.707   0.696  0.718 0.311     0.290    0.331
```

```
# fit the model on the training data
glm.out <- glm(Diabetes.b ~ Age.c + BMI.c + Sex + Race + Pulse.c + BPS + BPD, data = tdata,
 family = binomial)
summary(glm.out)
```

```
#
# Call:
# glm(formula = Diabetes.b ~ Age.c + BMI.c + Sex + Race + Pulse.c +
#     BPS + BPD, family = binomial, data = tdata)
#
# Coefficients:
#               Estimate Std. Error z value Pr(>|z|)
# (Intercept)  -2.335129   0.350601  -6.660 2.73e-11 ***
# Age.c         0.064334   0.003291  19.548  < 2e-16 ***
# BMI.c         0.071425   0.006454  11.067  < 2e-16 ***
# SexMale       0.412869   0.096915   4.260 2.04e-05 ***
# RaceAsian     0.957175   0.156929   6.099 1.06e-09 ***
# RaceBlack     0.395837   0.128812   3.073 0.002119 **
# RaceHispanic  0.182828   0.186099   0.982 0.325893
# RaceMex_Amer  0.650191   0.156062   4.166 3.10e-05 ***
# RaceOther     0.415180   0.224180   1.852 0.064027 .
# Pulse.c       0.031917   0.004189   7.619 2.55e-14 ***
# BPS           0.001772   0.002803   0.632 0.527272
# BPD          -0.013930   0.003664  -3.802 0.000143 ***
# ---
# Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#
# (Dispersion parameter for binomial family taken to be 1)
#
#     Null deviance: 3756.7  on 4548  degrees of freedom
# Residual deviance: 2887.5  on 4537  degrees of freedom
# AIC: 2911.5
#
# Number of Fisher Scoring iterations: 6
```

```
# predict BPS on vdata
vpred <- predict(glm.out, newdata = vdata, type = "response")
glm.predb <- ifelse(vpred >= 0.13, 1, 0)
table(glm.predb, vdata$Diabetes.b)
```

```
#
# glm.predb    0    1
#         0 1116   49
#         1  534  235
```

```
prop.table(table(glm.predb, vdata$Diabetes.b), margin = 2)
```

```
#
# glm.predb         0         1
#         0 0.6763636 0.1725352
#         1 0.3236364 0.8274648
```

```
prop.table(table(glm.predb, vdata$Diabetes.b), margin = 1)
```

```
#
# glm.predb          0          1
#         0 0.95793991 0.04206009
#         1 0.69440832 0.30559168
```

Elapsed time: 0.234 sec.

3. Deep Learning Models

3.1 BPS

```
x_train <- model.matrix(~0 + Race:Sex + Age.c + BMI.c +
Pulse.c, data = tdata) x_test <- model.matrix(~0 + Race:Sex + Age.c +
BMI.c + Pulse.c, data = vdata) y_test <- vdata\(BPS y_train &lt;- tdata\)BPS
```

set up the model for deep neural network

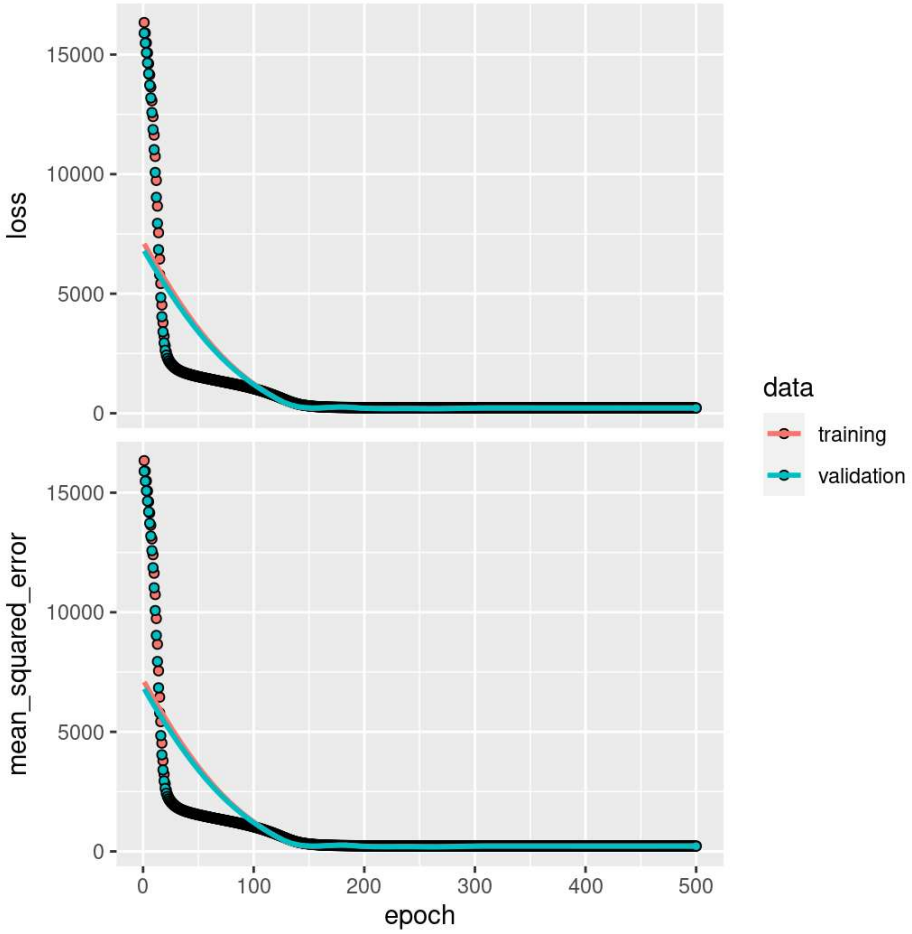
```
mymodel <- keras_model_sequential() mymodel %>%
layer_dense(name = "Layer1", units = 15, activation = "relu",
input_shape = c(15)) %>% layer_dense(name = "Layer2", units = 15,
activation = "relu") %>% layer_dense(name = "OutputLayer", units = 1,
activation = "linear")
```

summary model

```
summary(mymodel)
```

```
# Model: "sequential"
# _________________________________________________________________________
#  Layer (type)                       Output Shape                    Param #
# =========================================================================
#  Layer1 (Dense)                     (None, 15)                      240
#  Layer2 (Dense)                     (None, 15)                      240
#  OutputLayer (Dense)                (None, 1)                       16
# =========================================================================
# Total params: 496 (1.94 KB)
# Trainable params: 496 (1.94 KB)
# Non-trainable params: 0 (0.00 Byte)
# _________________________________________________________________________
```

```
# compile the model
mymodel %>%
    compile(loss = "mean_squared_error", optimizer = "adam", metrics = "mean_squared_error")

# fit the model
history <- mymodel %>%
    fit(x_train, y_train, epoch = 500, batch_size = 256, validation_split = 0.3,
        verbose = 0)

plot(history)
```




```
# prediction for training data
mypred <- predict(mymodel, x_train)
```

```
# 143/143 - 0s - 136ms/epoch - 950us/step
```

```
# r-squared for train data
1 - sum((y_train - mypred)^2)/sum((y_train - mean(y_train))^2)
```

```
# [1] 0.4311886
```

```
# prediction for test data
mypred <- predict(mymodel, x_test)
```

```
# 61/61 - 0s - 40ms/epoch - 657us/step
```

```
# r-squared for test data
1 - sum((y_test - mypred)^2)/sum((y_test - mean(y_test))^2)
```

```
# [1] 0.3931495
```

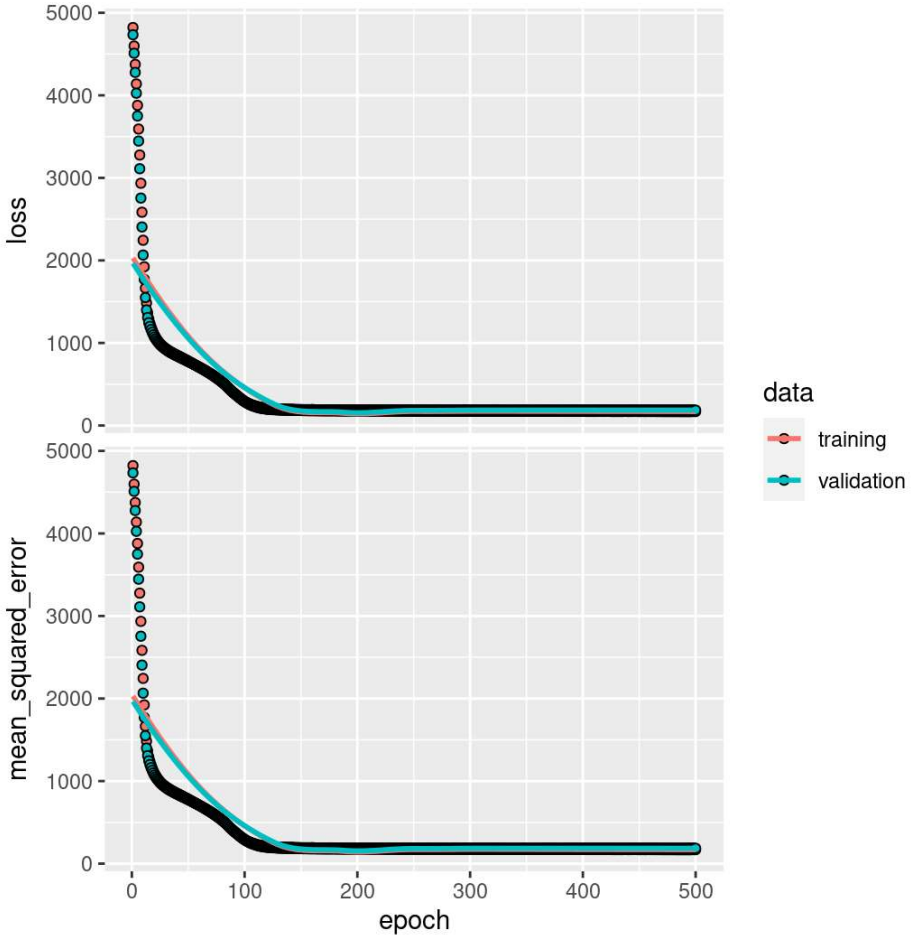
Elapsed time: 57.686 sec.

3.2 BPD

```
y_test <- vdata$BPD
y_train <- tdata$BPD

# set up the model for deep neural network
mymodel <- keras_model_sequential()
mymodel %>%
    layer_dense(name = "Layer1", units = 15, activation = "relu", input_shape = c(15)) %>%
    layer_dense(name = "Layer2", units = 15, activation = "relu") %>%
    layer_dense(name = "OutputLayer", units = 1, activation = "linear")

# summary model
summary(mymodel)
```

```
# Model: "sequential_1"
# ________________________________________________________________________________
#  Layer (type)                       Output Shape                    Param #
# ================================================================================
#  Layer1 (Dense)                     (None, 15)                      240
#  Layer2 (Dense)                     (None, 15)                      240
#  OutputLayer (Dense)                (None, 1)                       16
# ================================================================================
# Total params: 496 (1.94 KB)
# Trainable params: 496 (1.94 KB)
# Non-trainable params: 0 (0.00 Byte)
# ________________________________________________________________________________
```

```
# compile the model
mymodel %>%
    compile(loss = "mean_squared_error", optimizer = "adam", metrics = "mean_squared_error")

# fit the model
history <- mymodel %>%
    fit(x_train, y_train, epoch = 500, batch_size = 256, validation_split = 0.3,
        verbose = 0)
plot(history)
```




```
# prediction for training data
mypred <- predict(mymodel, x_train)
```

```
# 143/143 - 0s - 125ms/epoch - 873us/step
```

```
# r-squared for train data
1 - sum((y_train - mypred)^2)/sum((y_train - mean(y_train))^2)
```

```
# [1] 0.3420803
```

```
# prediction for test data
mypred <- predict(mymodel, x_test)
```

```
# 61/61 - 0s - 49ms/epoch - 800us/step
```

```
# r-squared for train data
1 - sum((y_test - mypred)^2)/sum((y_test - mean(y_test))^2)
```

```
# [1] 0.2893855
```

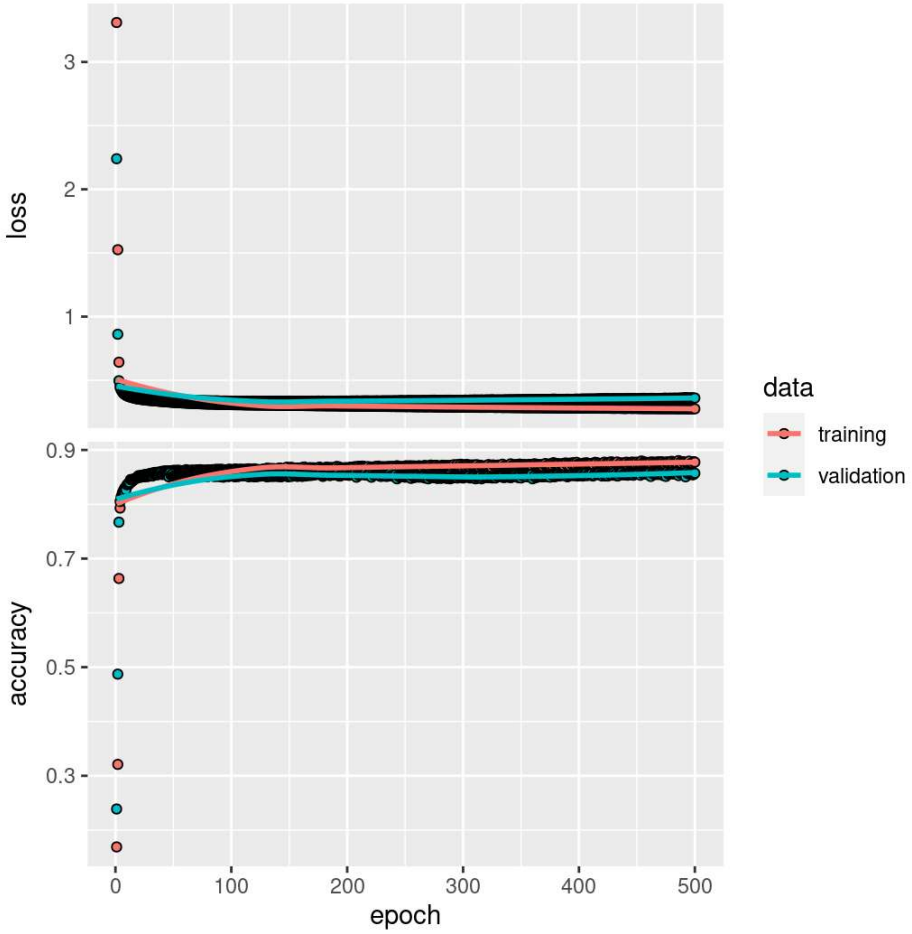
Elapsed time: 23.289 sec.

3.3 Diabetes

```
y_test <- to_categorical(vdata$Diabetes.b)
y_train <- to_categorical(tdata$Diabetes.b)

# set up the model for deep neural network
mymodel <- keras_model_sequential()
mymodel %>%
    layer_dense(name = "Layer1", units = 15, activation = "relu", input_shape = c(15)) %>%
    layer_dense(name = "Layer2", units = 15, activation = "relu") %>%
    layer_dense(name = "OutputLayer", units = 2, activation = "softmax")

# summary model
summary(mymodel)
```

```
# Model: "sequential_2"
# ________________________________________________________________________________
#  Layer (type)                       Output Shape                    Param #
# ================================================================================
#  Layer1 (Dense)                     (None, 15)                      240
#  Layer2 (Dense)                     (None, 15)                      240
#  OutputLayer (Dense)                (None, 2)                       32
# ================================================================================
# Total params: 512 (2.00 KB)
# Trainable params: 512 (2.00 KB)
# Non-trainable params: 0 (0.00 Byte)
# ________________________________________________________________________________
```

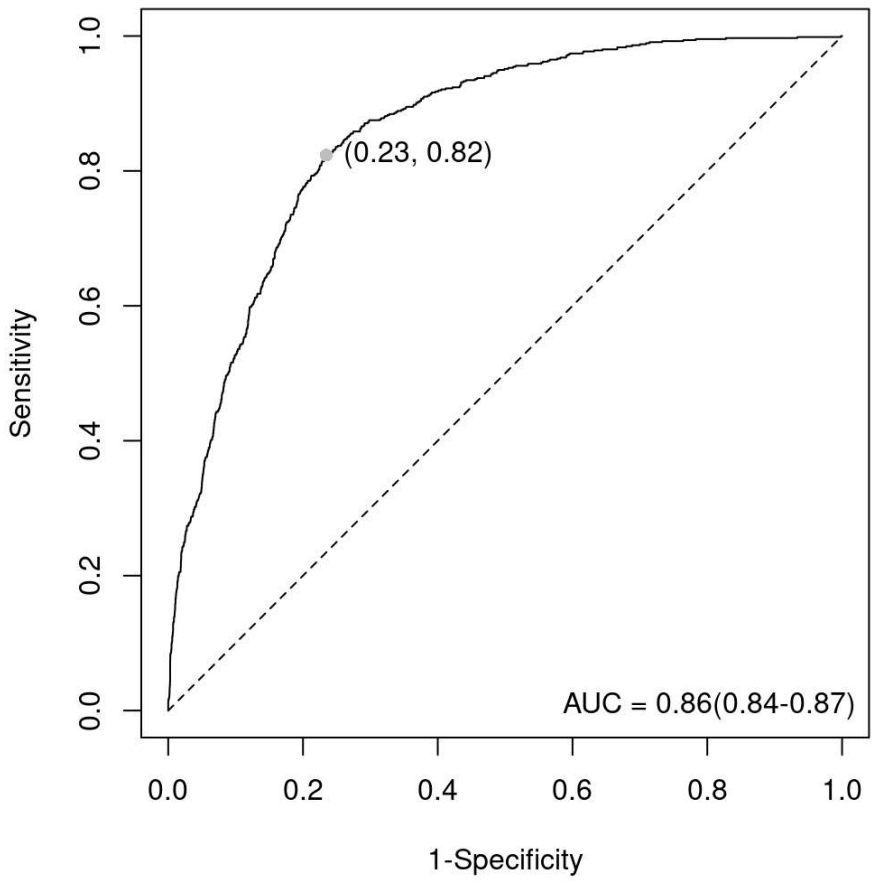
```
# compile the model
mymodel %>%
    compile(loss = "categorical_crossentropy", optimizer = "adam", metrics = "accuracy")

# fit the model
history <- mymodel %>%
    fit(x_train, y_train, epoch = 500, batch_size = 256, validation_split = 0.3,
        verbose = 0)
plot(history)
```




```
# prediction for training data
mypred <- predict(mymodel, x_train)[, 2]
```

```
# 143/143 - 0s - 112ms/epoch - 787us/step
```

```
# ROC
reportROC(tdata$Diabetes.b, mypred)
```




```
#  Cutoff   AUC AUC.SE AUC.low AUC.up     P   ACC ACC.low ACC.up   SEN SEN.low
#   0.161 0.858  0.007   0.844  0.872 0.000 0.774   0.774  0.774 0.823   0.794
#  SEN.up   SPE SPE.low SPE.up   PLR PLR.low PLR.up   NLR NLR.low NLR.up   PPV
#   0.853 0.765   0.752  0.779 3.510   3.283  3.753 0.231   0.195  0.272 0.372
#  PPV.low PPV.up   NPV NPV.low NPV.up   PPA PPA.low PPA.up   NPA NPA.low NPA.up
#    0.347  0.397 0.963   0.956  0.969 0.823   0.794  0.853 0.765   0.752  0.779
#    TPA TPA.low TPA.up KAPPA KAPPA.low KAPPA.up
#  0.774   0.762  0.786 0.391     0.364    0.419
```

```
# confusion matrix training data
predb <- ifelse(mypred >= 0.125, 1, 0)
table(predb, tdata$Diabetes.b)
```

```
#
# predb    0    1
#     0 2789   93
#     1 1103  564
```

```
prop.table(table(predb, tdata$Diabetes.b))
```

```
#
# predb          0          1
#     0 0.61310178 0.02044405
#     1 0.24247087 0.12398329
```

```
prop.table(table(predb, tdata$Diabetes.b), margin = 2)
```

```
#
# predb         0         1
#     0 0.7165982 0.1415525
#     1 0.2834018 0.8584475
```

```
# confusion matrix test data
mypred <- predict(mymodel, x_test)[, 2]
```

```
# 61/61 - 0s - 44ms/epoch - 729us/step
```

```
predb <- ifelse(mypred >= 0.125, 1, 0)
table(predb, vdata$Diabetes.b)
```

```
#
# predb    0    1
#     0 1150   66
#     1  500  218
```

```
prop.table(table(predb, vdata$Diabetes.b))
```

```
#
# predb          0          1
#     0 0.59462254 0.03412616
#     1 0.25853154 0.11271975
```

```
prop.table(table(predb, vdata$Diabetes.b), margin = 2)
```

```
#
# predb         0         1
#     0 0.6969697 0.2323944
#     1 0.3030303 0.7676056
```

Elapsed time: 24.266 sec.