

Egészségügyi előrejelzés logisztikus regresszióval

– avagy egy adattábla vizsgálatának, tisztításának története, benne R-szkriptek

Ez az elemzés arra szándékozik rámutatni, hogy az adattudomány módszerei (jelen esetben a logisztikus regressziós modell – **R nyelvi környezetben**) alkalmasak egészségügyi adatokból kiindulva azonosítani a magas kockázatú eseteket, előre jelezve egy jövőbeli betegséget, illetve nem kívánt kimenetelt. Így lehetővé téve a megelőzést, az időben való életmódváltást, gyógykezelést. Az adattábla része annak az adathalmaznak, mely Framingham városának (Massachusetts állam) lakóin végzett rendszeres **szív és érrendszeri** vizsgálatokból származik.

Forrás: Framingham heart study dataset

<https://www.kaggle.com/datasets/aasheesh200/framingham-heart-study-dataset/>

Osztályok, sorok és oszlopok

Fájl beolvasás:

```
szdn <- read.csv("framingham.csv")
```

A "sysBP", "diaBP", "BMI" változók "numeric" a többi egyelőre "integer" osztályú:

```
> sapply(szdn, class)
```

```
  male      age  education currentSmoker  cigsPerDay
"integer" "integer"  "integer"  "integer"  "integer"
BPMeds prevalentStroke prevalentHyp  diabetes  totChol
"integer" "integer"  "integer"  "integer"  "integer"
  sysBP    diaBP      BMI  heartRate  glucose
"numeric" "numeric" "numeric"  "integer"  "integer"
TenYearCHD
"integer"
```

```
> length(row.names(szdn))
```

```
[1] 4240
```

```
> sapply(szdn, typeof)
```

```
  male      age  education currentSmoker  cigsPerDay
"integer" "integer"  "integer"  "integer"  "integer"
BPMeds prevalentStroke prevalentHyp  diabetes  totChol
"integer" "integer"  "integer"  "integer"  "integer"
  sysBP    diaBP      BMI  heartRate  glucose
"double"  "double"  "double"  "integer"  "integer"
TenYearCHD
"integer"
```

Mint látható **4240 adatsora** van a táblának, illetve 16 változója.

A **célváltozó** a "TenYearCHD" (hogyan áll e a szívkoszorúér-betegség kockázata a következő 10 évben: 1 a veszélyeztetett, 0 a nem veszélyeztetett).

A **független változók**: férfi (1 igen, 0 nő); kor; iskolai végzettség; dohányos (1 igen, 0 nem); napi cigaretta szám; vérnyomás csökkentő gyógyszer szedése; stroke előfordulás; magas vérnyomás; cukorbetegség megléte (1 igen, 0 nem); koleszterinszint; vérnyomás felső értéke; vérnyomás alsó értéke; testtömeg; pulzusszám; glükóz szint.

Három változó (sysBP, diaBP, BMI) lebegőpontos, a többi egész típusú.

Változók, értékek, nem kitöltött cellák (NA), diagramok

```
> sum(is.na(szdn))
```

```
[1] 645
```

Ezek szerint 645 cellában szerepel NA – azaz ennyi helyen nincs használható érték.

Jó áttekinteni a változók alapparamétereit.

```

> for (i in colnames(szdn)) {
  print(i)
  print(summary(szdn[,i]))
  print("_____")
}
[1] "male"
0      1
2420   1820
[1] "_____ "
[1] "age"
  Min. 1st Qu. Median   Mean 3rd Qu.  Max.
32.00  42.00  49.00  49.58  56.00  70.00
[1] "_____ "
[1] "education"
1      2      3      4      NA's
1720   1253   689   473   105
[1] "_____ "
[1] "currentSmoker"
0      1
2145   2095
[1] "_____ "
[1] "cigsPerDay"
  Min. 1st Qu. Median   Mean 3rd Qu.  Max.   NA's
0.000  0.000  0.000  9.006  20.000  70.000  29
[1] "_____ "
[1] "BPMeds"
0      1      NA's
4063   124   53
[1] "_____ "
[1] "prevalentStroke"
0      1
4215   25
[1] "_____ "
[1] "prevalentHyp"
0      1
2923   1317
[1] "_____ "
[1] "diabetes"
0      1
4131   109
[1] "_____ "
[1] "totChol"
  Min. 1st Qu. Median   Mean 3rd Qu.  Max.   NA's
107.0  206.0  234.0  236.7  263.0  696.0   50
[1] "_____ "
[1] "sysBP"
  Min. 1st Qu. Median   Mean 3rd Qu.  Max.
83.5  117.0  128.0  132.4  144.0  295.0
[1] "_____ "
[1] "diaBP"
  Min. 1st Qu. Median   Mean 3rd Qu.  Max.
48.0  75.0  82.0  82.9  90.0  142.5
[1] "_____ "
[1] "BMI"
  Min. 1st Qu. Median   Mean 3rd Qu.  Max.   NA's
15.54  23.07  25.40  25.80  28.04  56.80   19
[1] "_____ "
[1] "heartRate"
  Min. 1st Qu. Median   Mean 3rd Qu.  Max.   NA's
44.00  68.00  75.00  75.88  83.00  143.00   1
[1] "_____ "
[1] "glucose"
  Min. 1st Qu. Median   Mean 3rd Qu.  Max.   NA's

```

```
40.00 71.00 78.00 81.96 87.00 394.00 388
[1] "_____"
[1] "TenYearCHD"
0      1
3596 644
[1] "_____"
```

A kiírtak egy része nem tartalmaz lényegi információt – mint a **"male" változó** adatai – hacsak azt nem, hogy minden értéke valódi (nincs benne NA). Illetve, hogy 1820 férfi és 2420 nő adatai szerepelnek a táblában – illetve más módon:

```
> sum(szdn[szdn$male==1,"male"])
[1] 1820
> sum((szdn[szdn$male==0,"male"]+1))
[1] 2420
```

Vagy gyakorisági táblázatban:

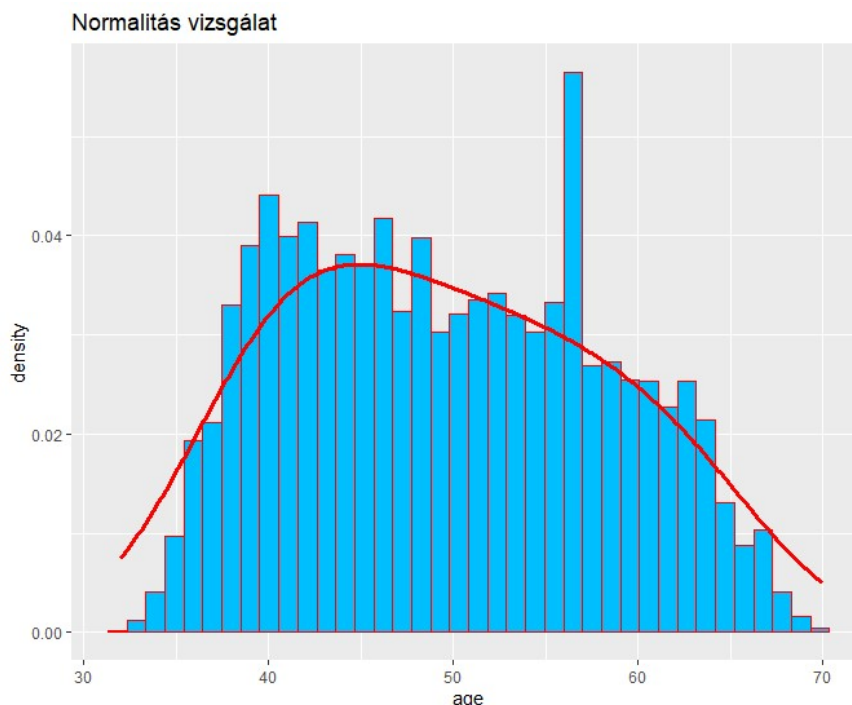
```
> table(szdn$male)
```

```
0      1
2420 1820
```

Az **"age" változó** értékei kimondottan szimmetrikusan helyezkednek el a minimális 32 és a maximális 70 éves kor között és lényeges, hogy nem tartalmaz NA-t.

Hisztogramja elég "normális" (lásd az alábbi ábrát).

```
> library(ggplot2)
> ggplot(szdn, aes(x=age)) +
  geom_histogram(aes(y=after_stat(density)), fill="deepskyblue", color="red", bins=38) +
  labs(title = "Normalitás vizsgálat") +
  geom_density(adjust=3, color="red", size=1.4)
```



Az "age" változó hisztogramja

Az **"education" változó** problémásabb. Mint a fenti áttekintő sorában látható, 105 db NA-t tartalmaz, ami így is megjeleníthető:

```
> sum(is.na(szdn$education))  
[1] 105
```

Adataiból látható, hogy 1-4 közötti (zárt értelemben) egész értékeket (1,2,3,4) vehet fel, melyek jelentései: 1 – középiskola; 2 – felsőoktatás; 3 – posztgraduális; 4 – PhD

Mivel 105 sor elvesztése nem tűnik jó megoldásnak, így az "education" változó által tartalmazott NA-k helyét a lehetséges értékekkel (1,2,3,4) töltöm ki, azok arányában.

A lehetséges értékek gyakoriságai (az NA-k nélkül)

```
> table(szdn$education)
```

```
 1     2     3     4  
1720 1253  689  473
```

```
> sum(table(szdn$education))  
[1] 4135
```

Láthatóan éppen az NA-k száma (105) hiányzik.

```
> sum(is.na(szdn$education))  
[1] 105
```

```
> xx<-c(1:sum(is.na(szdn$education))) # csak a vektor def. miatt
```

A függvényt, ami véletlenszerű pozíciókra elhelyezni a megfelelő számú lehetséges értéket (1,2,3,4), **osztó()** névre kereszteltem.

```
osztó <- function(zz) { # véletlenszerűen kiosztja az NA-k helyére  
  # a nem NA-k gyakorisági arányában a lehetséges  
  # értékeket
```

```
  edugyakor<-table(szdn$education) # gyakorisági tábla - NA-k nélkül
```

```
  # az NA-k számának szétosztása a többi "education" érték arányában
```

```
  aranyos<-round((edugyakor[1:4]/sum(edugyakor))*sum(is.na(szdn$education)), digits=0)
```

```
  aranyos[4]<-sum(is.na(szdn$education))-sum(aranyos[1:3])
```

```
  rx<- sample(1:sum(is.na(szdn$education)), sum(is.na(szdn$education))) # egész véletlen számok
```

```
  ry<-c(1:sum(is.na(szdn$education))) # csak a vektor def. miatt
```

```
  # az "education" lehetséges értékeinek véletlenszerű besorolása az 'ry' vektorba - azok arányában  
  x<-0
```

```
  for (jj in 1:4) {  
    for (ii in 1:aranyos[jj]) {  
      ry[ii+x]<-jj  
      # print(c(x,jj,ii))  
    }  
    x<-x+aranyos[jj]  
  }
```

```
  # az "education" lehetséges értékeinek kiosztása az NA-k helyére véletlen sorrendben
```

```
  for (ii in 1:length(rx)) {  
    zz[rx[ii]]<-ry[ii]  
  }  
  return(zz)  
} # osztó fgg vége
```

Ezt követően:

```
> qq<-oszto(xx)
```

Elől a besorolt értékek, illetve utána azok száma:

```
> for (jj in 1:4) {  
  print(c(jj,sum(qq==jj)))  
}  
[1] 1 44  
[1] 2 32  
[1] 3 17  
[1] 4 12
```

Az összegük:

```
> length(qq)  
[1] 105
```

Épen annyi, mint az NA-k száma az "education" változóban.

```
> sum(is.na(szdn$education))  
[1] 105
```

A "qq"-ban lévő értékek elhelyezése az "education" változóba.

```
> szdn[is.na(szdn$education),"education"]<-qq  
> sum(is.na(szdn$education)) # az NA-k eltűntek.  
[1] 0
```

A kiigazítás utáni gyakoriságok és összegük:

```
> table(szdn$education)  
1      2      3      4  
1764 1285  706  485  
> sum(table(szdn$education))  
[1] 4240
```

A **"currentSmoker"** változó rendben van. Gyakorisági táblája:

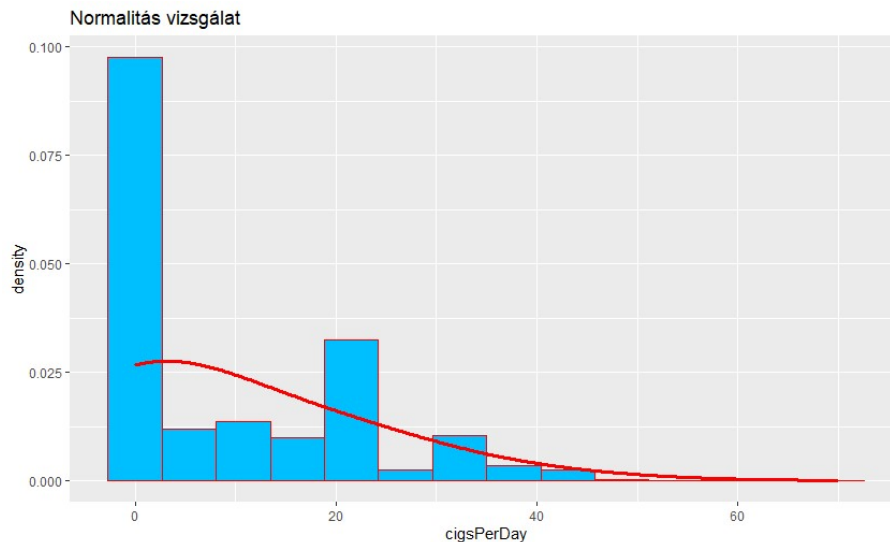
```
> table(szdn$currentSmoker)  
0      1  
2145 2095
```

A **"cigsPerDay"** esetében van 29 NA.

```
> sum(is.na(szdn$cigsPerDay))  
[1] 29
```

Vizsgáljuk meg, milyen értéket vesz fel a "currentSmoker", azokban a sorokban, ahol a "cigsPerDay" értéke NA (29 db sor).

```
> szdn[is.na(szdn$cigsPerDay),c("cigsPerDay", "currentSmoker")]  
  cigsPerDay currentSmoker  
132      NA           1
```

Az "cigsPerDay" változó hisztogramja

A **"BPMeds"** változó 53 NA-t tartalmaz.

```
> sum(is.na(szdn$BPMeds))
[1] 53
```

Minden bizonnyal feltételezhető, hogy ezek a páciensek nem szednek vérnyomás csökkentő gyógyszer, így ezeket a cellákat 0 értékre lehet állítani.

```
> szdn[is.na(szdn$BPMeds),"BPMeds"]
[1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
NA NA NA NA NA NA
[30] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
NA
> szdn[is.na(szdn$BPMeds),"BPMeds"]<-0
> sum(is.na(szdn$BPMeds))
[1] 0
```

Gyakorisági tábla:

```
> table(szdn$BPMeds)

0      1
4116  124
```

A **"prevalentStroke"**, **"prevalentHyp"**, **"diabetes"** változók rendben vannak. A gyakorisági táblák:

```
> table(szdn$prevalentStroke)
```

```
0      1
4215  25
```

```
> table(szdn$prevalentHyp)
```

```
0      1
2923 1317
```

```
> table(szdn$diabetes)
```

```
0      1
4131 109
```

Megjegyzem, hogy a diabéteszesek átlagéletkora 55 év.

```
> round(mean(szdn[(szdn$diabetes==1),"age"]), digits=0)
[1] 55
```

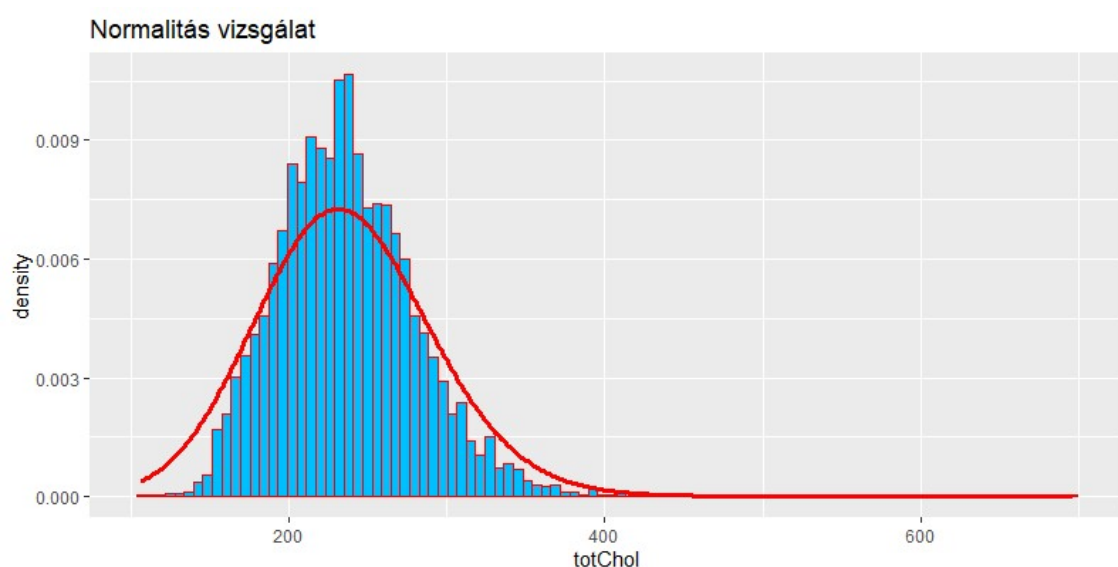
A **"totChol"** változó (koleszterinszint) esetén 50 NA észlelhető.

```
> sum(is.na(szdn$totChol))
[1] 50
```

Itt nyilván a többi érték egészre kerekített átlagával cserélhetők le az NA-k.

```
> szdn[(is.na(szdn$totChol)), "totChol"]
[1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
NA NA NA NA NA NA
[30] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
> szdn[(is.na(szdn$totChol)), "totChol"] <-
  round(mean(szdn[!(is.na(szdn$totChol)), "totChol"]), digits=0)
> sum(is.na(szdn$totChol))
[1] 0
```

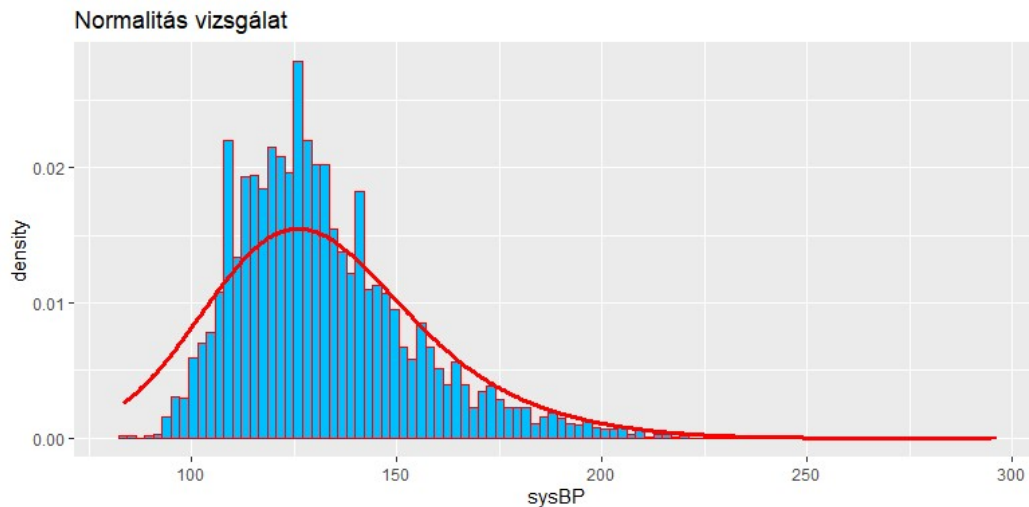
A hisztogramja elég normális, némileg balra tolódva.



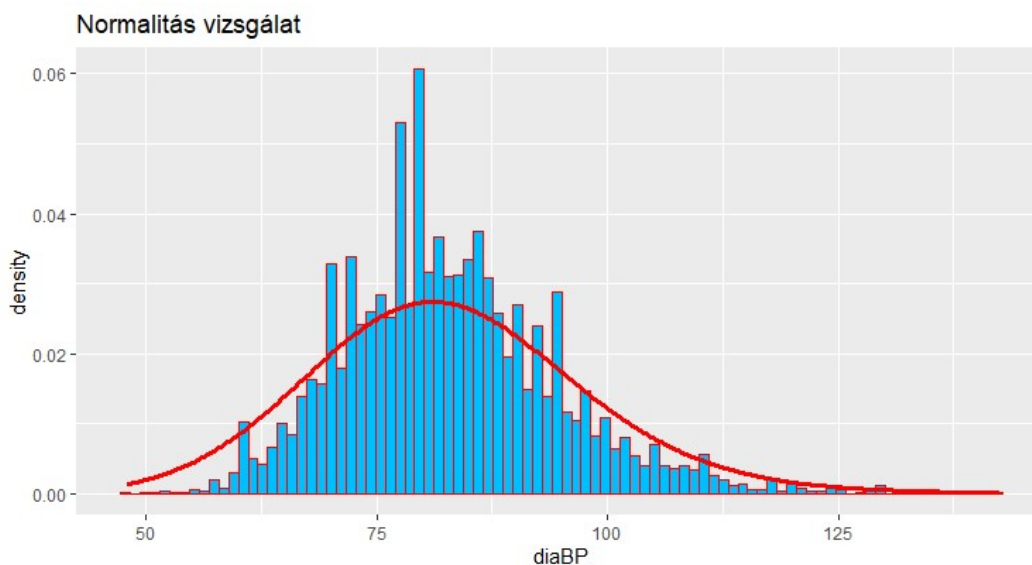
A "totChol" változó hisztogramja

A "sysBP", "diaBP" változók rendben vannak.

A hisztogramjaik normálisnak tekinthetők.



A "sysBP" változó hisztogramja



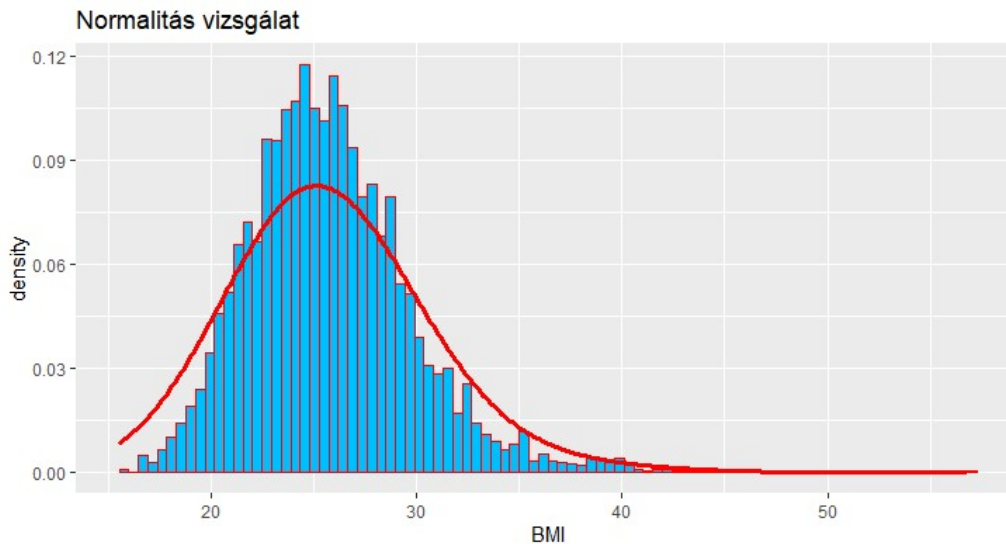
A "diaBP" változó hisztogramja

A "BMI" (testtömeg), "heartRate" (pulzusszám) változók esetén is a többiek átlagával helyettesíthetők az NA-k, az utóbbinál egészre kerekítve.

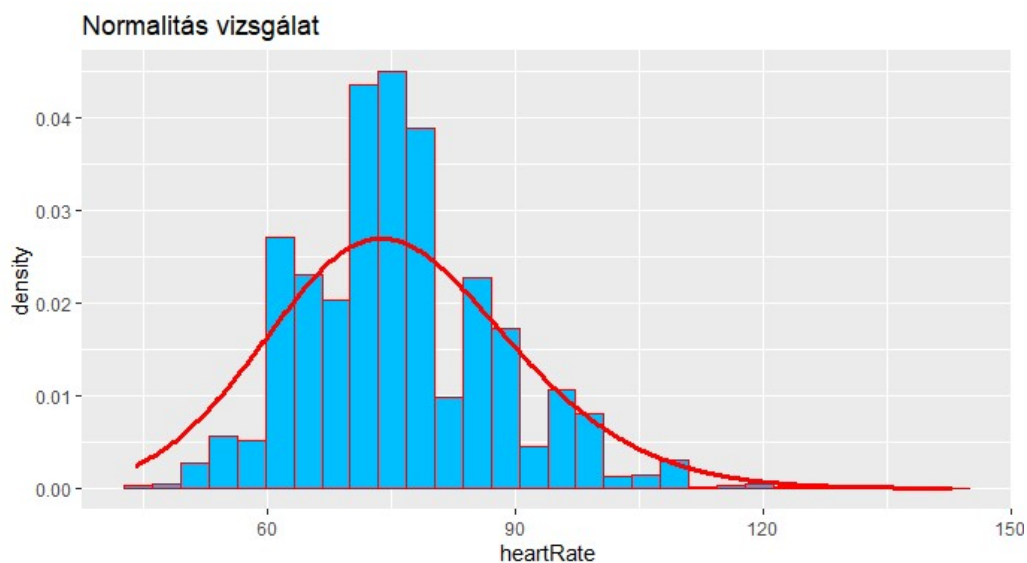
```
> sum(is.na(szdn$BMI))
[1] 19
> szdn[(is.na(szdn$BMI)), "BMI"]
[1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
> szdn[(is.na(szdn$BMI)), "BMI"] <-
+ mean(szdn[!(is.na(szdn$BMI)), "BMI"])
> sum(is.na(szdn$BMI))
[1] 0
> sum(is.na(szdn$heartRate))
[1] 1
> szdn[(is.na(szdn$heartRate)), "heartRate"]
[1] NA
> szdn[(is.na(szdn$heartRate)), "heartRate"] <-
```

```
+ round(mean(szdn[!(is.na(szdn$heartRate)),"heartRate"]), digits=0)
> sum(is.na(szdn$heartRate))
[1] 0
```

A hisztogramjaik normálisak.



A "BMI" változó hisztogramja



A "heartRate" változó hisztogramja

Mivel a célváltozó ("TenYearCHD") rendben van, így már csak a **"glucose"** (glükózsztint, mg/dl) van hátra.

Feltűnő, hogy nagyon sok (388 db) NA szerepel a "glucose" változóban – ez nem lehet véletlen. Megint segít, ha megnézzük, milyen értéket vesz fel a "diabetes", azokban a sorokban, ahol a "glucose" értéke NA. Nos – nem meglepően – ezen esetekben mindegyik a "diabetes" érték 0.

```
> szdn[is.na(szdn$glucose),c("glucose", "diabetes")]
glucose diabetes
```

```
15    NA      0
22    NA      0
27    NA      0
43    NA      0
```

```
55    NA    0
71    NA    0
112   NA    0
stb
```

Azaz előfordult, hogy bizonyos nem diabéteszes páciensek esetén nem jegyezték fel a vércukorszintet (vagy meg sem mérték).

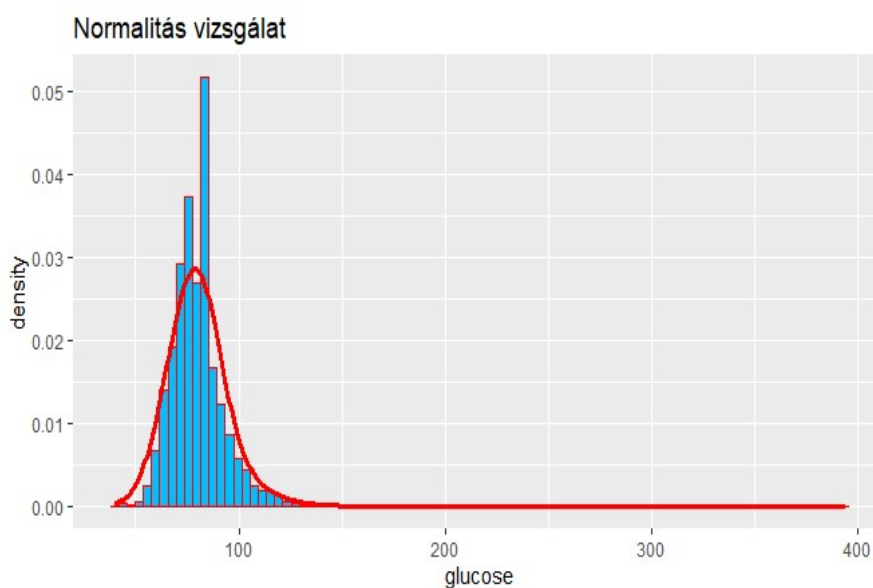
Ezért ezeket az NA értékeket egy **átlagos** normál értékre 85 mg/dl értékre állítom be.

```
> szdn[(is.na(szdn$glucose)), "glucose"] <- 85
> sum(is.na(szdn$glucose))
[1] 0
```

Így minden NA-t eltávolítottunk a táblából.

```
> sum(is.na(szdn))
[1] 0
```

A "glucose" balra tolódott normális eloszlású.



A "glucose" változó hisztogramja

A célváltozó ("TenYearCHD") gyakorisági táblája:

```
> table(szdn$TenYearCHD)

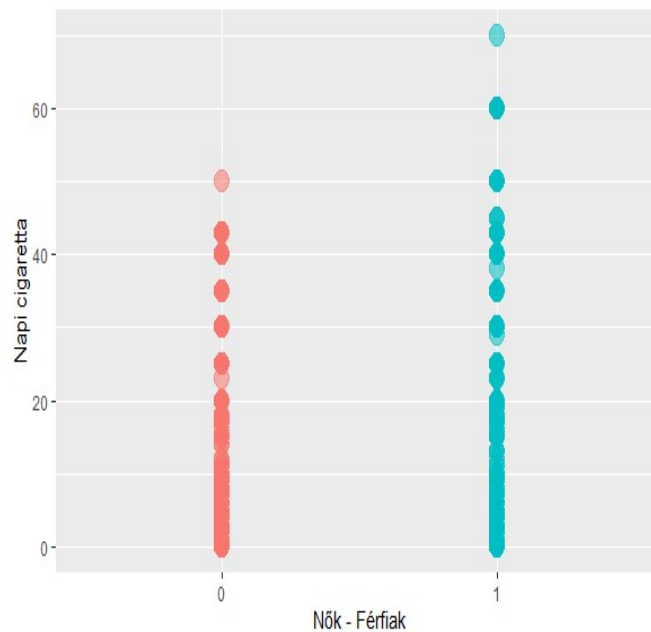
0      1
3596  644
```

A tisztított fájl mentése

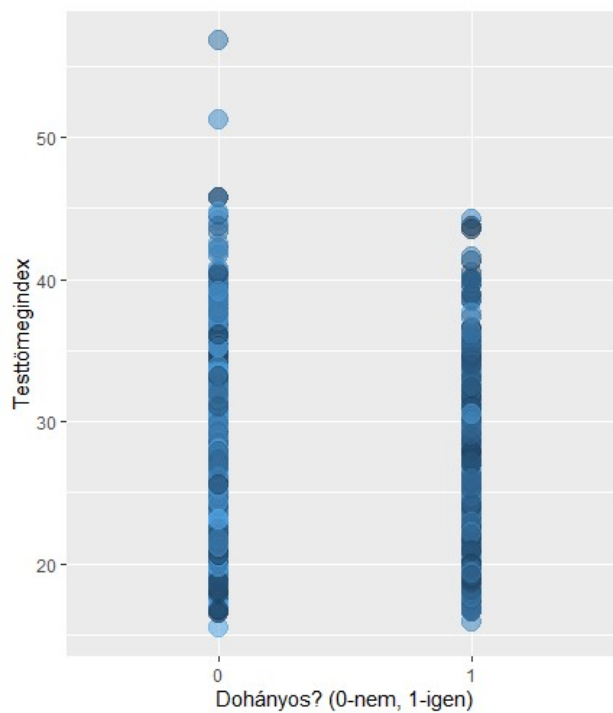
```
> write.csv(szdn, file=" framingham_To.csv", row.names = F)
```

Áttekintő diagramok

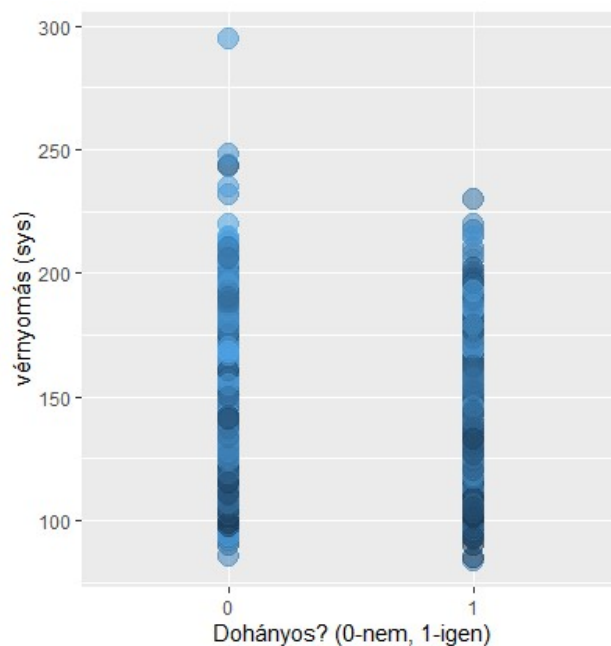
A férfiak (az adatok rögzítése idején) még többet dohányoztak.



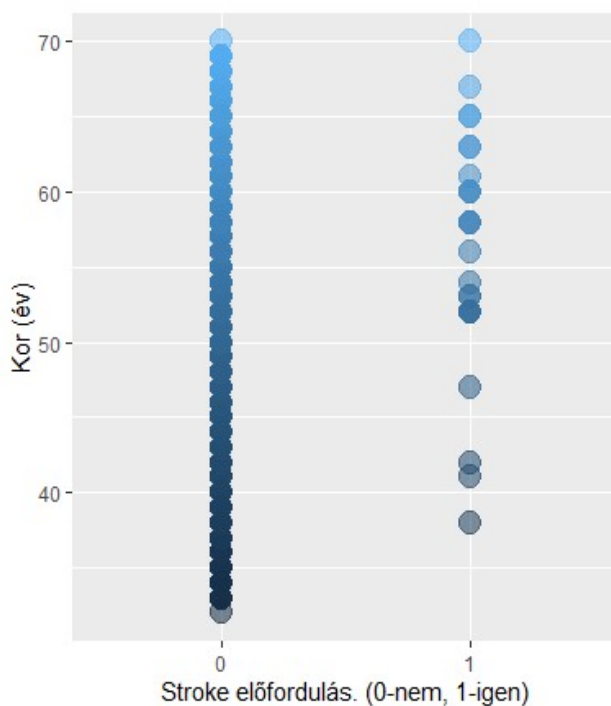
A dohányosok kissé szikárabbak.



Hümm... Ez érdekes. Dohányosoknak alacsonyabb a vérnyomása??



Nos, igen. A szívinfarktus esélye a korral nő.



Kiugró értékek szűrése

A kiugró adatsorok kiszűrésére a lineáris regresszió reziduüm táblázatát használom. Amelyik reziduális abszolútértéke nagyobb mint 2.9 (szűk 3 szórás), azt tekintem kiugróértéknek.

Az NA-t már nem tartalmazó tábla beolvasása, sztenderdizálása, illetve konvertálás dataframe formátumra, majd a reziduüm tábla összeállítása.

```
> szdn <- read.csv("df/mas_beadandok/eg/framingham_To.csv")
> szdnSZ<-as.data.frame(scale(szdn))
> model_sz <- lm(formula=TenYearCHD ~ ., data=szdnSZ)
```

```
> modelResiduals <- as.data.frame(residuals(model_sz))
> colnames(modelResiduals) <- "rezidumok"
> modelResiduals[abs(modelResiduals$rezidumok)>2.9,]
[1] 2.997584 2.967966
> rownames(modelResiduals)[modelResiduals$rezidumok>2.9]
[1] "963" "3488"
```

Két kiszűrt reziduális mutatkozik, ezek sorszámai (mint látható) a "963" és a "3488". Ezeket a sorokat törölöm, így 4238 soros tábla adódik.

```
> szdn<-szdn[c(-963,-3488),]
> length(row.names(szdn))
[1] 4238
```

Ezt a táblát is lementem:

```
> write.csv(szdn, file="df/mas_beadandok/eg/framingham_T.csv", row.names = F)
```

Regresszió

A próba kedvéért elsőre minden függetlenváltozót bevontam a regresszióba – a végeredmény alapján tisztábban lehet látni...

A cél változót leválasztottam, majd sztenderdizálás és a faktorváltozók konvertálása után, következett tábla felosztása "train" és "test" részre.

```
> szdn <- read.csv("framingham_T.csv")
> torzs<-szdn[,-16]
> cel<-szdn$TenYearCHD
> torzs_s<-as.data.frame(scale(torzs, center = TRUE, scale = TRUE))

> library(caret)
> split <- createDataPartition(cel, p = 0.8, list = FALSE)
> X_train <- torzs_s[split, ]
> X_test <- torzs_s[-split, ]
> Y_train <- cel[split]
> Y_test <- cel[-split]

> length(row.names(X_train))
[1] 3391
> length(row.names(X_test))
[1] 847
> length(Y_train)
[1] 3391
> length(Y_test)
[1] 847
> 3391+847
[1] 4238
```

Jöhetett a **tényleges regresszió** szakasza.

```
library(stats)
> train_data <- as.data.frame(cbind(target = Y_train, X_train))
> model <- glm(target ~ ., data = train_data, family = "binomial")
> predictions <- predict(model, newdata = as.data.frame(X_test), type = "response")
> binary_predictions <- ifelse(predictions >= 0.5, 1, 0)
> tar <- data.frame(megf = Y_test, pred = binary_predictions)
```

Zavaró volt (és elbizonytalanító), hogy a confusionMatrix() funkció éppen fordítva írja az értékeket a confusion matrix szélére mint szokásos (előbb a 0, utána az 1), így külön is kiszámoltam az alap mutatókat – az eligazodás kedvéért.

Konvertáltam a típusokat azonos (numerikus) alakra, mellékhátas ként a 0-ból 1 lett, az 1-ből 2.

```
> megf<-as.numeric(tar$megef)
> pred<-as.numeric(as.factor(tar$pred))

> sum((as.integer((megf==2)))&(as.integer((pred==2)))) # TP
[1] 11
> sum((as.integer(!(megf==2)))&(as.integer((pred==2)))) # FP
[1] 6
> sum((as.integer((megf==2)))&(as.integer(!(pred==2)))) # FN
[1] 117
> sum((as.integer(!(megf==2)))&(as.integer(!(pred==2)))) # TN
[1] 713
> sensitivity<-TP/(TP+FN)
[1] 0.0859375
> specificity<-TN/(TN+FP)
[1] 0.9916551
> PPV<-TP/(TP+FP)
[1] 0.6470588
> NPV<-TN/(TN+FN)
[1] 0.8590361
> f1_score<-2*PPV*sensitivity/(PPV+sensitivity)
[1] 0.1517241
```

Majd a "caret" könyvtár onfusionMatrix függvénye következik.

```
> confmx<-confusionMatrix(as.factor(tar$pred), as.factor(tar$megef), positive = "1")
```

Confusion Matrix and Statistics

Reference

Prediction	0	1
0	713	117
1	6	11

Accuracy : 0.8548

95% CI : (0.8292, 0.8778)

No Information Rate : 0.8489

P-Value [Acc > NIR] : 0.3363

Kappa : 0.1206

Mcnemar's Test P-Value : <2e-16

Sensitivity : 0.08594

Specificity : 0.99166

Pos Pred Value : 0.64706

Neg Pred Value : 0.85904

Prevalence : 0.15112

Detection Rate : 0.01299

Detection Prevalence : 0.02007

Balanced Accuracy : 0.53880

'Positive' Class : 1

Nos, ez nem valami jó eredmény.

Érdekes lehet megnézni az együtthatókat (coefficients):

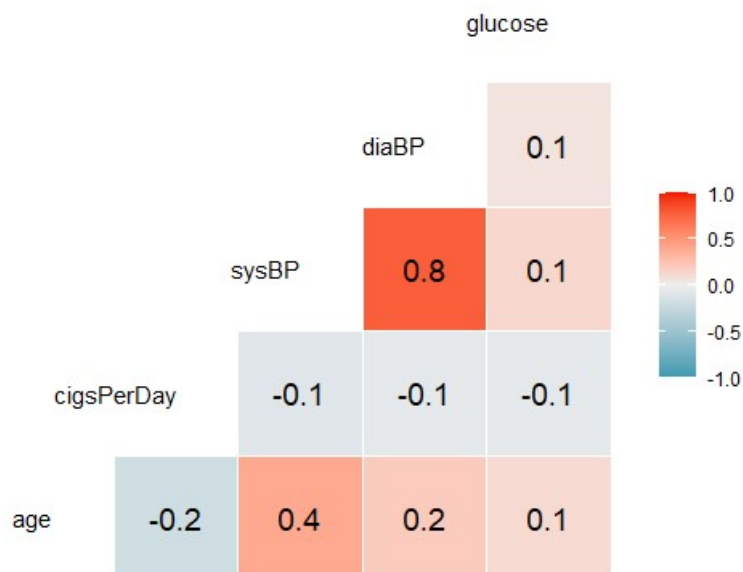
"male":	0.5365427
"age":	0.5458093
"education"1:	-0.1871551
"education"2:	-0.06053849
"education"3:	0.1480964
"currentSmoker":	-0.009582588
"cigsPerDay":	0.2745735
"BPMeds":	0.2401977
"prevalentStroke":	1.175277
"prevalentHyp":	0.1618606
"diabetes":	0.08020376
"totChol":	0.1242981
"sysBP":	0.3294597
"diaBP":	-0.02696032
"BMI" :	-0.008373888
"heartRate":	0.02494553
"glucose":	0.2005193

A "magyarázóerő" tekintetében a nagyobb együtthatójú változók inkább a mérvadóak.

A negatívak jelen esetben elég kétséges értelműek, pl. hogy dohányosnak lenni csökkentené (?) szívinfarktus kialakulásának esélyét, vagy a testtömeg index... Külön érdekes a vérnyomás második értékének negatív együtthatója. Az iskolai végzettség ügye nem egyértelmű és kicsinyek is a vonatkozó együtthatók. A stroke előfordulás pedig egyértelműen függ össze a további szívinfarktus kialakulásának esélyével.

Ezek alapján megismétlem a procedúrát. A kiválasztott változók: "male", "age", "cigsPerDay", "sysBP", "diaBP", "glucose". A célváltozó persze továbbra is a "TenYearCHD".

A : "male" nominális, a többi független változó skála típus. Így előbb ez utóbbiak összefüggéseit tekintem át egy **Pearson korrelációs mátrix** révén.



Láthatóan a "sysBP", és a "diaBP" erősen korrelál – ami nem meglepő. Úgy döntöttem a **"diaBP"-t is kihagyom**. Bár az "age"-sysBP" négyzet is erősebben virít (0,4) – amin ugyan csak nem lehet meglepődni – de összességében a maradék változók elegendően függetlennek tekinthetők.

A "male" bináris lévén Point-biserial korrelációval vizsgálható a többi független változóhoz való viszonya.

```
> for (i in 2:5) {  
  corv<-cor.test(alap$male, alap[,i], method = "pearson")  
  print(paste("Point Biserial Correlation(", "male", " ~ ", vvar[i-1], ") = ",  
    as.character(round(corv[["estimate"]][["cor"]], digits=6)), sep=""))  
}  
[1] "Point Biserial Correlation(male ~ age) = -0.02971"  
[1] "Point Biserial Correlation(male ~ cigsPerDay) = 0.316234"  
[1] "Point Biserial Correlation(male ~ sysBP) = -0.03669"  
[1] "Point Biserial Correlation(male ~ glucose) = 0.002195"
```

Mindegyik érték nullához közeli, elmondható, hogy függetlenek.

Az ismételt regressziós

A metódus lefutása után külön is kiszámoltam a fő mutatókat:

```
TP:    11  
FP:    10  
FN:    114  
TN:    712  
  
sensitivity: 0.0880000  
specificity: 0.9861496  
PPV:        0.5238095  
NPV:        0.8619855  
f1_score:   0.1506849
```

Illetve a metódus eredményei:

Confusion Matrix and Statistics

```
Reference  
Prediction    0     1  
0         712   114  
1         10     11
```

```
Accuracy : 0.8536  
95% CI : (0.828, 0.8767)  
No Information Rate : 0.8524  
P-Value [Acc > NIR] : 0.4852  
Kappa : 0.113  
Mcnemar's Test P-Value : <2e-16  
Sensitivity : 0.08800  
Specificity : 0.98615  
Pos Pred Value : 0.52381  
Neg Pred Value : 0.86199  
Prevalence : 0.14758  
Detection Rate : 0.01299
```

Detection Prevalence : 0.02479
Balanced Accuracy : 0.53707
'Positive' Class : 1

... hasonló eredmény, mint az előbb...

Túl sok az FN...

Ezt követően felvettem még változókat többféle variációban, sőt bizonyos változók másfelek hatványával is próbálkoztam – de lényegileg jobbnak mondható eredményt nem kaptam.

A döntés határszámának állítása

Végül jóslás határszámának állításával próbálkoztam (hogyan milyen határtól legyen a döntés 1 és egyébként 0). A 0,2-es határszámmal lefolytatott regressziót fogadtam el – ez hozott lényeges változást.

```
> binary_predictions <- ifelse(predictions >= 0.2, 1, 0)
```

Confusion Matrix and Statistics

Reference

Prediction	0	1
0	582	57
1	142	66

Accuracy : 0.7651
95% CI : (0.735, 0.7932)
No Information Rate : 0.8548
P-Value [Acc > NIR] : 1
Kappa : 0.2646
McNemar's Test P-Value : 2.607e-09
Sensitivity : 0.53659
Specificity : 0.80387
Pos Pred Value : 0.31731
Neg Pred Value : 0.91080
Prevalence : 0.14522
Detection Rate : 0.07792
Detection Prevalence : 0.24557
Balanced Accuracy : 0.67023
'Positive' Class : 1