High School Graduation Prediction

Nelson Marcelo Ferreira Berg

2022

Contents

1 Preface

The objective of this capstone project is to create a predictive report to achieve the Professional Certificate Program of Data Science from HarvardX.

2 Introduction

Governance and corruption is a big issue that greatly affects education. Education has a weak system and the young people who attend them do not always come from a stable economic and social background. So, there are good chances of not finishing high school. This project is trying to predict high-school graduates in Paraguay using the *House's Permanent Poll* or *Encuesta Permanente de Hogares Continua* (EPHC) database from the *General Directorate of Statistics* from Paraguay. The EPHC collects information on different dimensions of the well-being of Paraguayan households, such as education, health, employment and income, among others. This report focuses on education.

We are going to use the EPHC database from 2019 for training our models and the EPHC from 2020 to test the models.

3 Data Preparation

The first step in the project is to download the necessary packages and load the EPHC databases.

```
# Loading packages
if(!require(caret)) install.packages("caret", repos="http://cran.us.r-project.org")
if(!require(e1071)) install.packages("e1071", repos="http://cran.us.r-project.org")
if(!require(haven)) install.packages("haven", repos="http://cran.us.r-project.org")
if(!require(ggplot2)) install.packages("ggplot2", repos="http://cran.us.r-project.org")
if(!require(ggthemes)) install.packages("ggthemes", repos="http://cran.us.r-project.org")
if(!require(ggpmisc)) install.packages("ggpmisc", repos="http://cran.us.r-project.org")
if(!require(kableExtra)) install.packages("kableExtra", repos="http://cran.us.r-project.org")
if(!require(MLmetrics)) install.packages("MLmetrics", repos="http://cran.us.r-project.org")
if(!require(rpart)) install.packages("rpart", repos="http://cran.us.r-project.org")
if(!require(stringr)) install.packages("stringr", repos="http://cran.us.r-project.org")
if(!require(surveyr)) install.packages("survey", repos="http://cran.us.r-project.org")
```

```
## package 'survey' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Nelson Jr\AppData\Local\Temp\RtmpSe3Lvo\downloaded_packages
```

```
if(!require(srvyr)) install.packages("srvyr", repos="http://cran.us.r-project.org")
if(!require(tidyverse)) install.packages("tidyverse", repos="http://cran.us.r-project.org")
```

##Load database

```
set.seed(1, sample.kind = "Rounding")
#2019
ephc2019 <- read_sav("reg02_ephc2019.sav")

#2020
ephc2020 <- read_sav("reg02_ephc2020.sav")
```

3.1 Variables in the data set

We explore the variables in the EPHC data set.

```
glimpse(ephc2019)
```

```
## Rows: 18,233
## Columns: 260
## $ UPM      <dbl> 106, 106, 106, 106, 106, 106, 106, 106, 106, 106, 106, 106, 1~
## $ NVIVI    <dbl> 4, 19, 19, 19, 19, 19, 19, 19, 19, 38, 38, 38, 38, 43, 43, 43~
## $ NHOGA    <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ DPTOREP  <dbl+lbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ AREA     <dbl+lbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ L02      <dbl> 1, 1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 1, 2, 3, 1, 2, 1, 2, 3~
## $ P02      <dbl+lbl> 29, 51, 62, 30, 27, 22, 21, 18, 14, 33, 32,  6,  3, 47, 4~
## $ P03      <dbl+lbl> 1, 1, 2, 3, 3, 3, 3, 3, 3, 1, 2, 3, 3, 1, 2, 8, 1, 2, 1, ~
## $ P04      <dbl+lbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ P04A     <dbl+lbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
## $ P04B     <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ P05C     <dbl+lbl> 0, 2, 1, 0, 0, 0, 0, 0, 0, 2, 1, 0, 0, 2, 1, 0, 2, 1, 0, ~
## $ P05P     <dbl+lbl> 0, 0, 0, 2, 2, 2, 2, 2, 2, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, ~
## $ P05M     <dbl+lbl> 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 2, 2, 0, 3, 0, 0, 0, 0, ~
## $ P06      <dbl+lbl> 1, 6, 1, 1, 1, 6, 6, 1, 6, 1, 6, 1, 6, 1, 6, 6, 1, 6, 6, ~
## $ P08D     <dbl+lbl> 21,  4, 18, 31,  3,  3, 31, 10, 14, 20, 18, 11, 10, 30,  ~
## $ P08M     <dbl+lbl>  3,  7,  7,  5,  7,  4,  8,  5,  3, 10, 10, 10,  7,  5,  ~
## $ P08A     <dbl+lbl> 1990, 1968, 1957, 1989, 1992, 1997, 1998, 2001, 2005, 198~
## $ P09      <dbl+lbl> 5, 1, 1, 5, 5, 5, 5, 5, 5, 2, 2, 5, 5, 2, 2, 5, 1, 1, 2, ~
## $ P10A     <dbl+lbl>  0,  9,  7,  0,  0,  0,  0,  0,  0, 11,  0,  0,  0,  0,  ~
## $ P10AB    <dbl+lbl>    0,  916,  701,    0,    0,    0,    0,    0,    0, 110~
## $ P10Z     <dbl> 1, 6, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 6, 6, 1, 1, 1~
## $ P11A     <dbl+lbl>  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, NA,  0,  ~
## $ P11AB    <dbl+lbl>  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, NA,  0,  ~
## $ P11Z     <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, NA, 1, 1, 1, 1, 1, 1, 1, ~
## $ P12      <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ A01      <dbl+lbl>  1,  1,  1,  6,  6,  6,  1,  6,  6,  1,  1, NA, NA,  6,  ~
## $ A01A     <dbl+lbl> NA, NA, NA,  1,  1,  1, NA,  1,  1, NA, NA, NA, NA,  2, N~
## $ A02      <dbl+lbl>  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1, NA, NA,  1,  ~
## $ A03      <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ A04      <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ A04A     <dbl+lbl>  1,  2,  1,  2,  1,  1,  1,  1,  1,  1,  1, NA, NA,  1,  ~
## $ A05      <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ A07      <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ A08      <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ A10      <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ A11A     <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ A11M     <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ A11S     <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ A12      <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ A13REC   <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ A14REC   <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ A15      <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ A16      <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ A17A     <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ A17M     <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
```

```
## $ A17S     <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ A18      <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ B01REC   <dbl+lbl>  7,  1,  7,  1,  4,  3,  5,  5,  5,  3,  3, NA, NA,  7,  ~
## $ B02REC   <dbl+lbl>  5,  5,  8,  7,  8,  7,  5,  5,  5,  7,  7, NA, NA,  2,  ~
## $ B03LU    <dbl+lbl> 11, 13,  8,  8, 10,  7, 10,  8,  4,  6,  6, NA, NA,  0,  ~
## $ B03MA    <dbl+lbl> 11, 13,  8,  8, 10,  7, 10,  8,  4,  6,  6, NA, NA,  0,  ~
## $ B03MI    <dbl+lbl> 11, 13,  8,  8, 10,  7, 10,  0,  4,  6,  6, NA, NA,  0,  ~
## $ B03JU    <dbl+lbl> 11, 13,  8,  8, 10,  7, 10,  8,  4,  6,  6, NA, NA,  0,  ~
## $ B03VI    <dbl+lbl> 11, 13,  8,  8,  0,  7, 10,  0,  4,  6,  6, NA, NA,  4,  ~
## $ B03SA    <dbl+lbl>  5,  0,  0,  0,  0,  5, 14,  8,  0,  9,  9, NA, NA,  4,  ~
## $ B03DO    <dbl+lbl>  0,  0,  0,  0,  4,  0,  5,  0,  0,  9,  9, NA, NA,  0,  ~
## $ B04      <dbl+lbl>  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1, NA, NA,  3,  ~
## $ B05      <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,  9, N~
## $ B06      <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 30, N~
## $ B07A     <dbl+lbl> 15, 13, 40,  1,  8,  2, 10,  8,  4, 14, 12, NA, NA,  0, 2~
## $ B07M     <dbl+lbl>  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, NA, NA, 10,  ~
## $ B07S     <dbl+lbl>  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, NA, NA,  0,  ~
## $ B08      <dbl+lbl>  3,  2,  4,  4,  5,  5,  2,  2,  2,  2,  2, NA, NA,  1,  ~
## $ B09A     <dbl+lbl>  0, 13, 40,  2,  8,  2, 10,  8,  4,  9,  9, NA, NA,  0, 2~
## $ B09M     <dbl+lbl>  6,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, NA, NA, 10,  ~
## $ B09S     <dbl+lbl>  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, NA, NA,  0,  ~
## $ B10      <dbl+lbl>  6,  6,  1,  1,  1,  1,  6,  6,  6,  6,  6, NA, NA,  6,  ~
## $ B11      <dbl+lbl> NA, NA,  2,  1,  1,  1, NA, NA, NA, NA, NA, NA, NA, NA,  ~
## $ B12      <dbl+lbl>  2,  3,  1,  2,  2,  2,  5,  5,  5,  3,  3, NA, NA,  4,  ~
## $ B12A     <dbl+lbl>  6, NA,  6,  6,  6,  6, NA, NA, NA, NA, NA, NA, NA, NA,  ~
## $ B12B     <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ B12C     <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ B13      <dbl+lbl> 88, NA, 45,  7, 18, 12, NA, NA, NA, NA, NA, NA, NA, NA, 3~
## $ B14      <dbl+lbl>  1, NA,  2,  2,  2,  1, NA, NA, NA, NA, NA, NA, NA, NA,  ~
## $ B15      <dbl+lbl>  4, NA,  1,  4,  4,  4, NA, NA, NA, NA, NA, NA, NA, NA,  ~
## $ B16G     <dbl+lbl>  500000,      NA, 2800000, 3000000, 3500000, 1800000,    ~
## $ B16U     <dbl+lbl>  3, NA,  5,  5,  5,  5, NA, NA, NA, NA, NA, NA, NA, NA,  ~
## $ B16D     <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ B16T     <dbl+lbl> 2000000,      NA, 2800000, 3000000, 3500000, 1800000,    ~
## $ B17      <dbl+lbl>  6, NA,  6,  6,  6,  6, NA, NA, NA, NA, NA, NA, NA, NA,  ~
## $ B18AG    <dbl+lbl>    NA,    NA,    NA,    NA,    NA,    NA,    NA,    NA,  ~
## $ B18AU    <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ B18BG    <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ B18BU    <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ B19      <dbl+lbl>  6, NA,  6,  6,  1,  6, NA, NA, NA, NA, NA, NA, NA, NA,  ~
## $ B20G     <dbl+lbl>    NA,    NA,    NA,    NA, 20000,    NA,    NA,    NA,  ~
## $ B20U     <dbl+lbl> NA, NA, NA, NA,  2, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ B20D     <dbl+lbl> NA, NA, NA, NA, 22, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ B20T     <dbl+lbl>     NA,     NA,     NA,     NA, 440000,     NA,     NA,  ~
## $ B21      <dbl+lbl>  6, NA,  6,  6,  6,  6, NA, NA, NA, NA, NA, NA, NA, NA,  ~
## $ B22      <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ B23      <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ B24      <dbl+lbl>  6, NA,  6,  1,  1,  1, NA, NA, NA, NA, NA, NA, NA, NA,  ~
## $ B25      <dbl+lbl>    NA,    NA,    NA, 6e+05, 3e+05, 3e+05,    NA,    NA,  ~
## $ B26      <dbl+lbl>  4, NA,  1,  1,  1,  1, NA, NA, NA, NA, NA, NA, NA, NA,  ~
## $ B271     <dbl+lbl> NA,  1, NA, NA, NA, NA, NA, NA, NA,  1,  1, NA, NA,  1, N~
## $ B272     <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA,  2,  2, NA, NA, NA, N~
## $ B28      <dbl+lbl>  1,  1, NA,  1,  1,  1, NA, NA, NA,  1,  1, NA, NA,  1, N~
## $ B29      <dbl+lbl>  1,  1, NA,  5,  5,  5, NA, NA, NA,  1,  1, NA, NA,  1, N~
```

```
## $ B30      <dbl+lbl>  1,  1, NA,  1,  1,  1, NA, NA, NA,  1,  1, NA, NA,  1, N~
## $ B31      <dbl+lbl>  6,  1,  6,  1,  6,  6,  6,  6,  6,  6,  6, NA, NA,  6,  ~
## $ C01REC   <dbl+lbl> NA,  5, NA,  2, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,  ~
## $ C02REC   <dbl+lbl> NA,  8, NA,  8, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,  ~
## $ C03      <dbl+lbl> NA, 24, NA, 26, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 3~
## $ C04      <dbl+lbl> NA,  1, NA,  1, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,  ~
## $ C05      <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ C06      <dbl+lbl> NA,  8, NA,  4, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,  ~
## $ C07      <dbl+lbl> NA,  1, NA,  6, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,  ~
## $ C08      <dbl+lbl> NA,  2, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,  ~
## $ C09      <dbl+lbl> NA,  1, NA,  2, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,  ~
## $ C101     <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ C102     <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ C11G     <dbl+lbl>      NA, 2800000,      NA, 2000000,      NA,      NA,    ~
## $ C11U     <dbl+lbl> NA,  5, NA,  5, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,  ~
## $ C11D     <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ C11T     <dbl+lbl>      NA, 2800000,      NA, 2000000,      NA,      NA,    ~
## $ C12      <dbl+lbl> NA,  6, NA,  6, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,  ~
## $ C13AG    <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ C13AU    <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ C13BG    <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ C13BU    <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ C14      <dbl+lbl> NA,  1, NA,  3, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,  ~
## $ C14A     <dbl+lbl> NA, NA, NA,  1, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ C14B     <dbl+lbl> NA, NA, NA,  5, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ C14C     <dbl+lbl> NA, NA, NA,  1, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ C15      <dbl+lbl> NA,  0, NA,  0, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,  ~
## $ C16REC   <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ C17REC   <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ C18      <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ C18A     <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ C18B     <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ C19      <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ D01      <dbl+lbl>  6,  6,  6,  6,  6,  6,  6,  6,  6,  6,  6, NA, NA,  1,  ~
## $ D02      <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,  8, N~
## $ D03      <dbl+lbl>  6,  6,  1,  1,  1,  1,  6,  6,  6,  1,  1, NA, NA,  2,  ~
## $ D04      <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,  1, N~
## $ D05      <dbl+lbl> NA, NA, 11, 11, 11,  7, NA, NA, NA, 12, 12, NA, NA,  1, N~
## $ E01A     <dbl+lbl> 2000000, 3000000, 2800000, 3000000, 3940000, 1800000,    ~
## $ E01B     <dbl+lbl>       0, 2800000,       0, 2000000,       0,       0,    ~
## $ E01C     <dbl+lbl>  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, NA, NA,  0,  ~
## $ E01D     <dbl+lbl>  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, NA, NA,  0,  ~
## $ E01E     <dbl+lbl>  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, NA, NA,  0,  ~
## $ E01F     <dbl+lbl> 0e+00, 0e+00, 0e+00, 0e+00, 0e+00, 0e+00, 0e+00, 0e+00, 0~
## $ E01G     <dbl+lbl>  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, NA, NA,  0,  ~
## $ E01H     <dbl+lbl>  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, NA, NA,  0,  ~
## $ E01I     <dbl+lbl>  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, NA, NA,  0,  ~
## $ E01J     <dbl+lbl>  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, NA, NA,  0,  ~
## $ E01K     <dbl+lbl>     0,      0,      0,      0,      0,      0,      0,  ~
## $ E01L     <dbl+lbl>  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0, NA, NA,  0,  ~
## $ E01M     <dbl+lbl> 6e+05, 0e+00, 0e+00, 0e+00, 0e+00, 0e+00, 0e+00, 0e+00, 0~
## $ ED01     <dbl+lbl>  2,  2,  2,  2,  2,  2,  2,  2,  2,  2,  2,  2, NA,  2,  ~
## $ ED02     <dbl+lbl>  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1, NA,  1,  ~
## $ ED03     <dbl+lbl>  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1, NA,  1,  ~
```

```
## $ ED0504   <dbl+lbl>  903,  306,  306, 2405,  903,  903, 2401,  902,  408, 240~
## $ ED06C    <dbl+lbl> NA, NA, NA,  1, NA, NA, 14, NA, NA, 14, 14, NA, NA,  1,  ~
## $ ED08     <dbl+lbl> 19, NA, NA, 18, 16, 19, 16,  3,  2, 19, 19,  2, NA, 19, 1~
## $ ED09     <dbl+lbl> NA, NA, NA,  2,  2, NA,  2,  3,  3, NA, NA,  3, NA, NA,  ~
## $ ED10     <dbl+lbl>  1, NA, NA, NA, NA, 18, NA, NA, NA,  2, 14, NA, NA,  2, N~
## $ ED11B1   <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, 12, 12, NA, NA,  8, NA, NA, N~
## $ ED11B2   <dbl+lbl> NA, NA, NA, NA, NA, NA, NA,  3,  4, NA, NA,  2, NA, NA, N~
## $ ED11B3   <dbl+lbl> NA, NA, NA, NA, NA, NA, NA,  0,  0, NA, NA, 12, NA, NA, N~
## $ ED11B4   <dbl+lbl> NA, NA, NA, NA, NA, NA, NA,  4,  4, NA, NA,  0, NA, NA, N~
## $ ED11B5   <dbl+lbl> NA, NA, NA, NA, NA, NA, NA,  1,  1, NA, NA,  0, NA, NA, N~
## $ ED11B6   <dbl+lbl> NA, NA, NA, NA, NA, NA, NA,  2,  2, NA, NA,  1, NA, NA, N~
## $ ED11B7   <dbl+lbl> NA, NA, NA, NA, NA, NA, NA,  2,  2, NA, NA,  1, NA, NA, N~
## $ ED11B8   <dbl+lbl> NA, NA, NA, NA, NA, NA, NA,  1,  1, NA, NA,  0, NA, NA, N~
## $ ED11B9   <dbl+lbl> NA, NA, NA, NA, NA, NA, NA,  6,  6, NA, NA,  6, NA, NA, N~
## $ ED11C1   <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ ED11D1   <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ ED11E1   <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ ED11F1   <dbl+lbl> NA, NA, NA, NA, NA, NA, NA,  6,  6, NA, NA,  1, NA, NA, N~
## $ ED11F1A  <dbl+lbl> NA, NA, NA, NA, NA, NA, NA,  6,  6, NA, NA,  1, NA, NA, N~
## $ ED11F1B  <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, 20, NA, NA, N~
## $ ED11G1   <dbl+lbl> NA, NA, NA, NA, NA, NA, NA,  6,  6, NA, NA,  6, NA, NA, N~
## $ ED11G1A  <dbl+lbl> NA, NA, NA, NA, NA, NA, NA,  6,  6, NA, NA,  6, NA, NA, N~
## $ ED11G1B  <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ ED11H1   <dbl+lbl> NA, NA, NA, NA, NA, NA, NA,  6,  6, NA, NA,  6, NA, NA, N~
## $ ED11H1A  <dbl+lbl> NA, NA, NA, NA, NA, NA, NA,  6,  6, NA, NA,  6, NA, NA, N~
## $ ED11H1B  <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ ED12     <dbl+lbl>  1,  1,  6,  1,  6,  1,  6,  6, NA,  1,  1, NA, NA,  1,  ~
## $ ED13     <dbl+lbl>  9,  5, NA,  2, NA, 10, NA, NA, NA, 11, 11, NA, NA,  2,  ~
## $ ED14     <dbl+lbl> 2018, 1988,   NA, 2019,   NA, 2019,   NA,   NA,   NA, 201~
## $ ED14A    <dbl+lbl>  1,  1, NA,  1, NA,  1, NA, NA, NA,  1,  1, NA, NA,  1,  ~
## $ ED15     <dbl+lbl>  1,  3, NA,  1, NA,  3, NA, NA, NA,  3,  3, NA, NA,  1,  ~
## $ S01A     <dbl+lbl> 7, 5, 5, 1, 1, 1, 7, 5, 5, 7, 4, 4, 4, 7, 5, 5, 7, 7, 1, ~
## $ S01B     <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ S02      <dbl+lbl> NA, NA, NA,  1,  1,  1, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ S03      <dbl+lbl> 3, 3, 1, 1, 3, 1, 2, 2, 3, 1, 1, 1, 1, 1, 1, 1, 3, 3, 1, ~
## $ S04      <dbl+lbl> NA, NA,  6,  6, NA,  6,  1,  1, NA,  6,  1,  1,  1,  1,  ~
## $ S05      <dbl+lbl> NA, NA,  5,  5, NA,  5, NA, NA, NA,  1, NA, NA, NA, NA,  ~
## $ S06      <dbl+lbl> NA, NA, NA, NA, NA, NA,  1,  1, NA, NA,  1,  1,  1,  1, N~
## $ S07      <dbl+lbl> NA, NA, NA, NA, NA, NA,  7,  9, NA, NA,  9,  9,  9,  7, N~
## $ S08      <dbl+lbl> NA, NA, NA, NA, NA, NA,  6,  6, NA, NA,  6,  6,  6,  6, N~
## $ S09      <dbl+lbl> NA, NA, NA, NA, NA, NA,  6,  6, NA, NA,  6,  6,  6,  6, N~
## $ CATE_PEA <dbl+lbl>  2,  3,  1,  2,  2,  2,  5,  5,  5,  3,  3, NA, NA,  4,  ~
## $ TAMA_PEA <dbl+lbl>  3,  2,  4,  4,  5,  5,  2,  2,  2,  2,  2, NA, NA,  1,  ~
## $ OCUP_PEA <dbl+lbl>  7,  1,  7,  1,  4,  3,  5,  5,  5,  3,  3, NA, NA,  7,  ~
## $ RAMA_PEA <dbl+lbl>  5,  5,  8,  7,  8,  7,  5,  5,  5,  7,  7, NA, NA,  2,  ~
## $ HORAB    <dbl> 60, 65, 40, 40, 44, 40, 69, 32, 20, 48, 48, NA, NA, 30, 30, N~
## $ HORABC   <dbl> 60, 89, 40, 66, 44, 40, 69, 32, 20, 48, 48, NA, NA, 30, 60, N~
## $ HORABCO  <dbl> 60, 89, 40, 66, 44, 40, 69, 32, 20, 48, 48, NA, NA, 30, 60, N~
## $ PEAD     <dbl+lbl>  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1, NA, NA,  1,  ~
## $ PEAA     <dbl+lbl>  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1, NA, NA,  1,  ~
## $ TIPOHOGA <dbl+lbl> 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 4, 4, 4, 2, 2, 4, ~
## $ FEX      <dbl> 708, 504, 504, 504, 504, 504, 504, 504, 504, 525, 525, 525, 5~
## $ NJEF     <dbl+lbl>  903,  306,  306,  306,  306,  306,  306,  306,  306, 240~
## $ NCON     <dbl+lbl> 8888,  306,  306, 8888, 8888, 8888, 8888, 8888, 8888, 240~
```

```
## $ NPAD     <dbl+lbl> 8888, 8888, 8888,  306,  306,  306,  306,  306,  306, 888~
## $ NMAD     <dbl+lbl> 8888, 8888, 8888,  306,  306,  306,  306,  306,  306, 888~
## $ TIC01    <dbl+lbl>  1,  1,  1,  1,  1,  1,  1,  1,  6,  1,  1, NA, NA,  1,  ~
## $ TIC02    <dbl+lbl>  6,  6,  6,  1,  1,  1,  1,  1,  1,  6,  1, NA, NA,  1,  ~
## $ TIC03    <dbl+lbl>  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1, NA, NA,  1,  ~
## $ TIC0401  <dbl+lbl>  6,  6,  6,  1,  1,  1,  1,  1,  1,  1,  1, NA, NA,  1,  ~
## $ TIC0402  <dbl+lbl>  6,  6,  6,  1,  1,  1,  6,  6,  6,  1,  1, NA, NA,  6,  ~
## $ TIC0403  <dbl+lbl>  6,  6,  6,  6,  6,  6,  6,  6,  6,  6,  6, NA, NA,  1,  ~
## $ TIC0404  <dbl+lbl>  6,  6,  6,  6,  6,  6,  6,  6,  6,  1,  1, NA, NA,  1,  ~
## $ TIC0405  <dbl+lbl>  6,  6,  6,  6,  6,  6,  6,  6,  6,  6,  6, NA, NA,  1,  ~
## $ TIC0406  <dbl+lbl>  6,  6,  6,  6,  6,  6,  6,  6,  6,  1,  6, NA, NA,  1,  ~
## $ TIC0407  <dbl+lbl>  1,  1,  1,  1,  1,  1,  1,  1,  6,  1,  6, NA, NA,  1,  ~
## $ TIC0408  <dbl+lbl>  6,  6,  6,  6,  6,  6,  6,  6,  6,  6,  1, NA, NA,  6,  ~
## $ TIC0409  <dbl+lbl>  6,  6,  6,  6,  6,  6,  6,  6,  6,  6,  6, NA, NA,  6,  ~
## $ TIC0501  <dbl+lbl>  1,  6,  6,  1,  1,  1,  1,  1,  6,  1,  1, NA, NA,  1,  ~
## $ TIC0502  <dbl+lbl>  6,  6,  1,  1,  1,  1,  1,  1,  6,  1,  1, NA, NA,  1,  ~
## $ TIC0503  <dbl+lbl>  1,  1,  6,  1,  1,  1,  1,  1,  6,  1,  1, NA, NA,  1,  ~
## $ TIC0504  <dbl+lbl>  1,  6,  6,  1,  1,  1,  1,  1,  6,  1,  1, NA, NA,  1,  ~
## $ TIC0505  <dbl+lbl>  1,  6,  6,  1,  1,  1,  1,  6,  6,  1,  1, NA, NA,  1,  ~
## $ TIC0506  <dbl+lbl>  6,  6,  6,  1,  1,  1,  1,  6,  6,  1,  1, NA, NA,  1,  ~
## $ TIC0507  <dbl+lbl>  6,  6,  6,  1,  1,  1,  1,  6,  6,  1,  1, NA, NA,  1,  ~
## $ TIC0508  <dbl+lbl>  6,  6,  6,  1,  1,  6,  1,  6,  6,  1,  1, NA, NA,  1,  ~
## $ TIC0509  <dbl+lbl>  6,  6,  6,  1,  1,  6,  6,  6,  6,  1,  1, NA, NA,  1,  ~
## $ TIC0510  <dbl+lbl>  6,  6,  6,  1,  6,  1,  1,  1,  1,  6,  6, NA, NA,  6,  ~
## $ TIC0511  <dbl+lbl>  6,  6,  6,  1,  1,  1,  1,  6,  6,  1,  6, NA, NA,  1,  ~
## $ TIC0512  <dbl+lbl>  1,  6,  1,  1,  1,  1,  1,  1,  1,  1,  6, NA, NA,  1,  ~
## $ TIC0513  <dbl+lbl>  6,  6,  6,  6,  6,  6,  6,  6,  6,  6,  6, NA, NA,  6,  ~
## $ TIC06    <dbl+lbl>  1,  1,  1,  1,  1,  1,  1,  1,  1,  1,  1, NA, NA,  1,  ~
## $ añoest   <dbl+lbl> 12,  6,  6, 17, 12, 12, 13, 11,  8, 14, 16,  0, NA, 17, 1~
## $ ra06ya09 <dbl+lbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ e01aimde <dbl+lbl> 2007626.3, 3011439.5, 2810676.8, 3011439.5, 3955023.8, 18~
## $ e01bimde <dbl+lbl>       0, 2810677,       0, 2007626,       0,       0,    ~
## $ e01cimde <dbl+lbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ e01dde   <dbl+lbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ e01ede   <dbl+lbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ e01fde   <dbl+lbl>      0.0,      0.0,      0.0,      0.0,      0.0,      0.~
## $ e01gde   <dbl+lbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ e01hde   <dbl+lbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ e01ide   <dbl+lbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ e01jde   <dbl+lbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ e01kde   <dbl+lbl>      0.0,      0.0,      0.0,      0.0,      0.0,      0.~
## $ e01lde   <dbl+lbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ e01mde   <dbl> 602287.9, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0~
## $ e01kjde  <dbl+lbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ e02bde   <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ ipcm     <dbl> 2911058.2, 2489456.6, 2489456.6, 2489456.6, 2489456.6, 248945~
## $ pobrezai <dbl+lbl> 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 3, ~
## $ pobnopoi <dbl+lbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, ~
## $ quintili <dbl> 5, 5, 5, 5, 5, 5, 5, 5, 5, 3, 3, 3, 3, 5, 5, 5, 2, 2, 3, 3, 3~
## $ decili   <dbl> 9, 9, 9, 9, 9, 9, 9, 9, 9, 5, 5, 5, 5, 9, 9, 9, 3, 3, 6, 6, 6~
## $ quintiai <dbl> 5, 4, 4, 4, 4, 4, 4, 4, 4, 2, 2, 2, 2, 5, 5, 5, 1, 1, 3, 3, 3~
## $ decilai  <dbl> 9, 8, 8, 8, 8, 8, 8, 8, 8, 4, 4, 4, 4, 9, 9, 9, 2, 2, 5, 5, 5~
```

As we can see here, there are many variables on our database. Now, we will select the ones we will be using

for the data exploration and modeling. There's also many NA, we'll deal with them.

3.2 Wrangling

```
# The data from 2019 will be our train set
eph_2019 <- ephc2019 %>%
  select(UPM, NVIVI, NHOGA, DPTOREP, AREA, P02, P03, P04A, P06, P08A,
         E01A, ED01, ED02, ED0504, añoest, ED09, S01A, S01B, FEX, NMAD, NPAD, pobnopoi, TIC0510, quintil

# The data from 2020 will be our test set
eph_2020 <- ephc2020 %>% select(UPM, NVIVI, NHOGA, DPTOREP, AREA, P02, P03, P04A, P06, P08A,
                                E01A, ED01, ED02, ED0504,añoest, ED09, S01A, S01B, FEX, NMAD, NPAD, pobn


# Remove the NA values
eph_2019 <- eph_2019 %>%
  filter(!is.na(AREA)) %>%
  filter(!is.na(añoest)) %>%
  filter(!is.na(P02)) %>%
  filter(!is.na(pobnopoi)) %>%
  filter(!is.na(ED01))

eph_2020 <- eph_2020 %>%
  filter(!is.na(AREA)) %>%
  filter(!is.na(añoest)) %>%
  filter(!is.na(P02)) %>%
  filter(!is.na(pobnopoi)) %>%
  filter(!is.na(ED01))

rm(ephc2019, ephc2020)
```

Selected variables for our databases:

UPM + NVIVI + NHOGA = Household Identification DPTOREP = Departament AREA = Residence Area P02 = Household member's age P03 = Family relationship P04A = Do you have an identity card? P06 = Sex P08A = Birth year E01A = Monthly income Main occupation declared ED01 = Language spoken in the home most of the time ED02 = Can you read and write? ED0504 = Highest level and grade passed añoest = Years of study ED09 = The institution or program where you attend is in the sector S01A = Do you currently have any health insurance in force in the country? S01B = Do you have insurance and what type of insurance? FEX = Expansion factor NMAD = Mother's level and degree of education NPAD = Father's level and degree of education pobnopoi = poverty level (constructed variable) TIC0510 = If you have an Internet connection (constructed variable)

```
# Create 'glyst_hs' variable
eph_2019<- eph_2019 %>%
  filter(!is.na(añoest)) %>%
  filter(!añoest %in% c("99")) %>%
  mutate(glyst_hs = ifelse(añoest %in%
                             c("1","2","3","4","5","6", "7", "8", "9", "10", "11"), "<12",
                           ifelse(añoest %in%
                                    c("12", "13", "14", "15", "16", "17", "18"), ">=12", "<12")))
eph_2019 <- eph_2019 %>% mutate(glyst_hs = as.factor(glyst_hs))
```

```
eph_2020<- eph_2020 %>%
  filter(!is.na(añoest)) %>%
  filter(!añoest %in% c("99")) %>%
  mutate(glyst_hs = ifelse(añoest %in%
                             c("1","2","3","4","5","6", "7", "8", "9", "10", "11"), "<12",
                           ifelse(añoest %in%
                                    c("12", "13", "14", "15", "16", "17", "18"), ">=12", "<12")))
eph_2020 <- eph_2020 %>% mutate(glyst_hs = as.factor(glyst_hs))

#Creating the 'graduate' variable
eph_2019 <- eph_2019 %>%
  filter(!is.na(añoest)) %>%
  mutate(graduate = ifelse(añoest %in% c("12"), "HSgrad",
                           ifelse(añoest %in% c("1","2","3","4","5","6"), "EEB_1_2",
                                  ifelse(añoest %in% c("7", "8", "9"), "EEB3",
                                         ifelse(añoest %in% c("10", "11"), "EM",
                                                ifelse(añoest %in%
                                                         c("13", "14", "15","16", "17", "18"), "H_ED",

eph_2020 <- eph_2020 %>%
  filter(!is.na(añoest)) %>%
  mutate(graduate = ifelse(añoest %in%
                             c("12"), "HSgrad",ifelse(añoest %in%
                                                        c("1","2","3","4","5","6"), "EEB_1_2",
                                                      ifelse(añoest %in% c("7", "8", "9"), "EEB3",
                                                             ifelse(añoest %in% c("10", "11"), "EM",
                                                                    ifelse(añoest %in% c("13", "14", "15
```

We created two new variables:

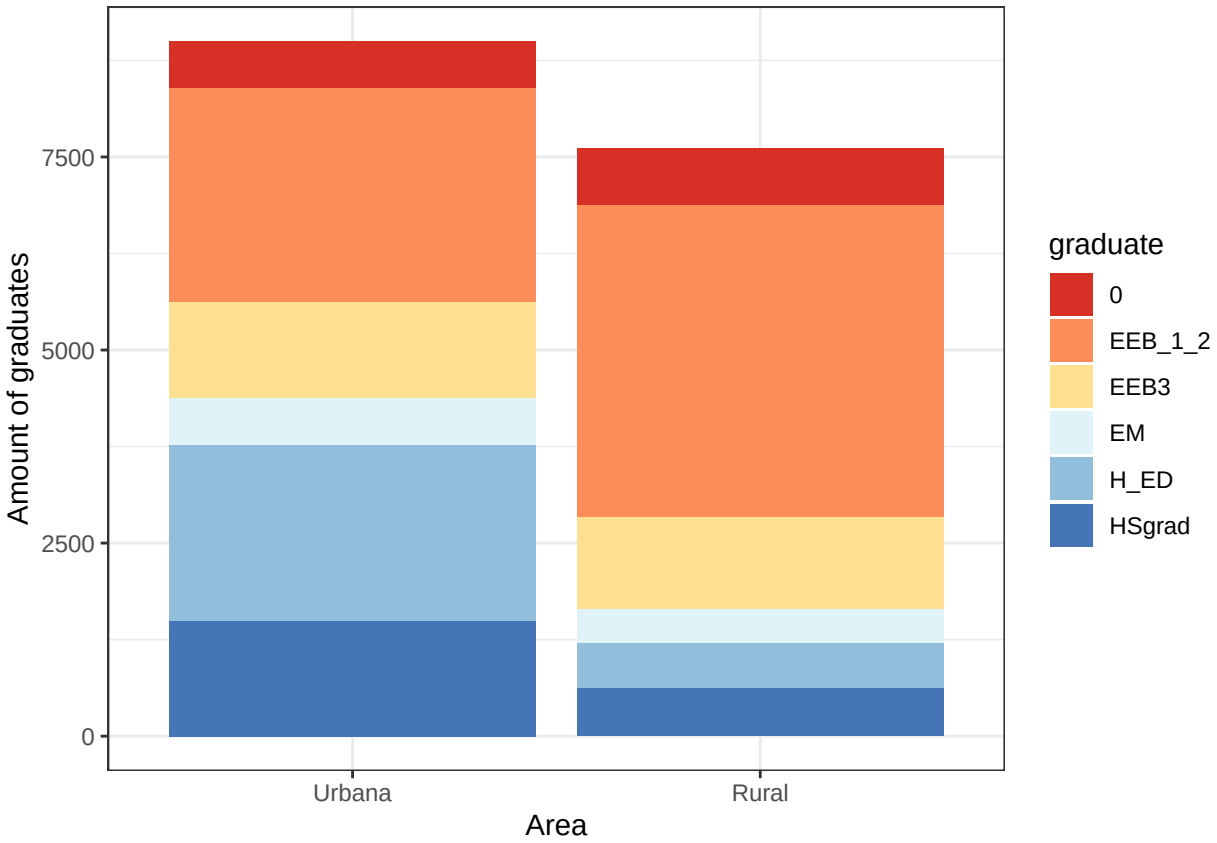
glyst_hs: describes the goal of years studied a person need to finish high-school in Paraguay. In Paraguay, 12 years is the normal duration to finish school and high-school. graduate: describes the title of the last academic year achieved in Paraguay.

4 Data exploration

In this phase, after loading the databases and preparing the data, we can begin with the data exploration.

4.1 Characteristics of high-school graduates.

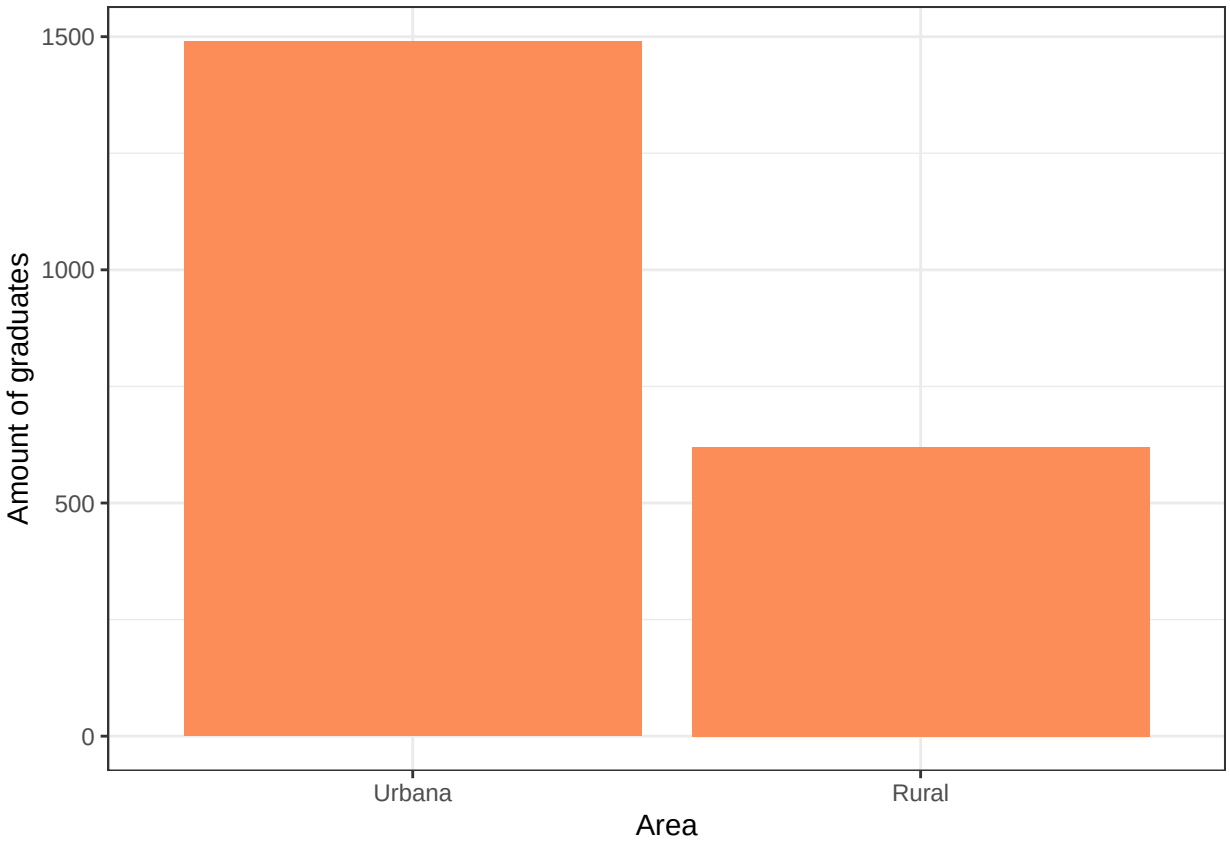
```
#Graduation per area
eph_2019 %>% group_by(graduate) %>% ggplot(aes(as_factor(AREA)))+ geom_bar(aes(fill=graduate)) + labs(x
```



We can observe the distribution of the different titles of the last academic year achieved classified by urban and rural areas.

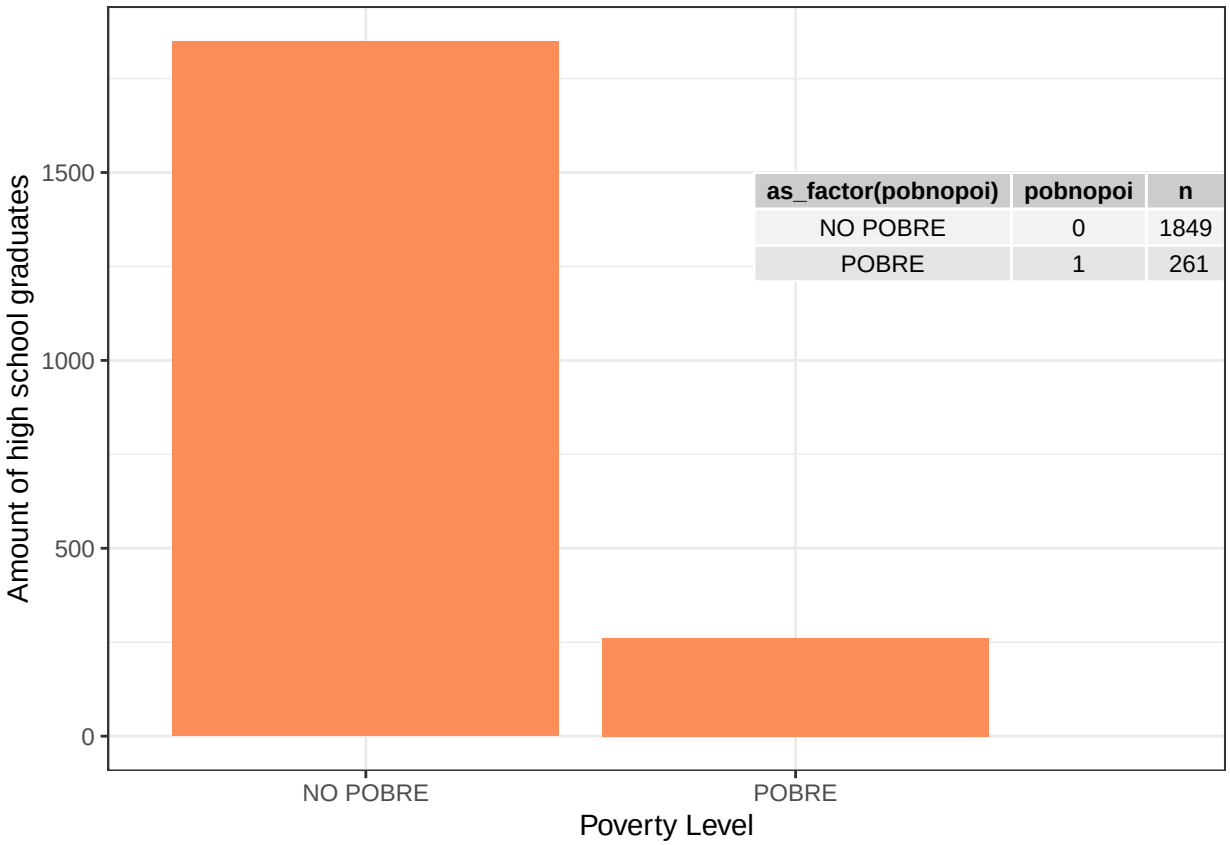
4.2 Characteristics of high-school graduates.

```
#High School graduation per area
eph_2019 %>% group_by(graduate) %>% filter(añoest %in% c("12")) %>% ggplot(aes(as_factor(AREA)))+ geom_
```



In this graph we can observe that there is a larger amount of high-school graduates in urban areas than in rural areas. People from urban areas have higher chances of graduating from highschool.

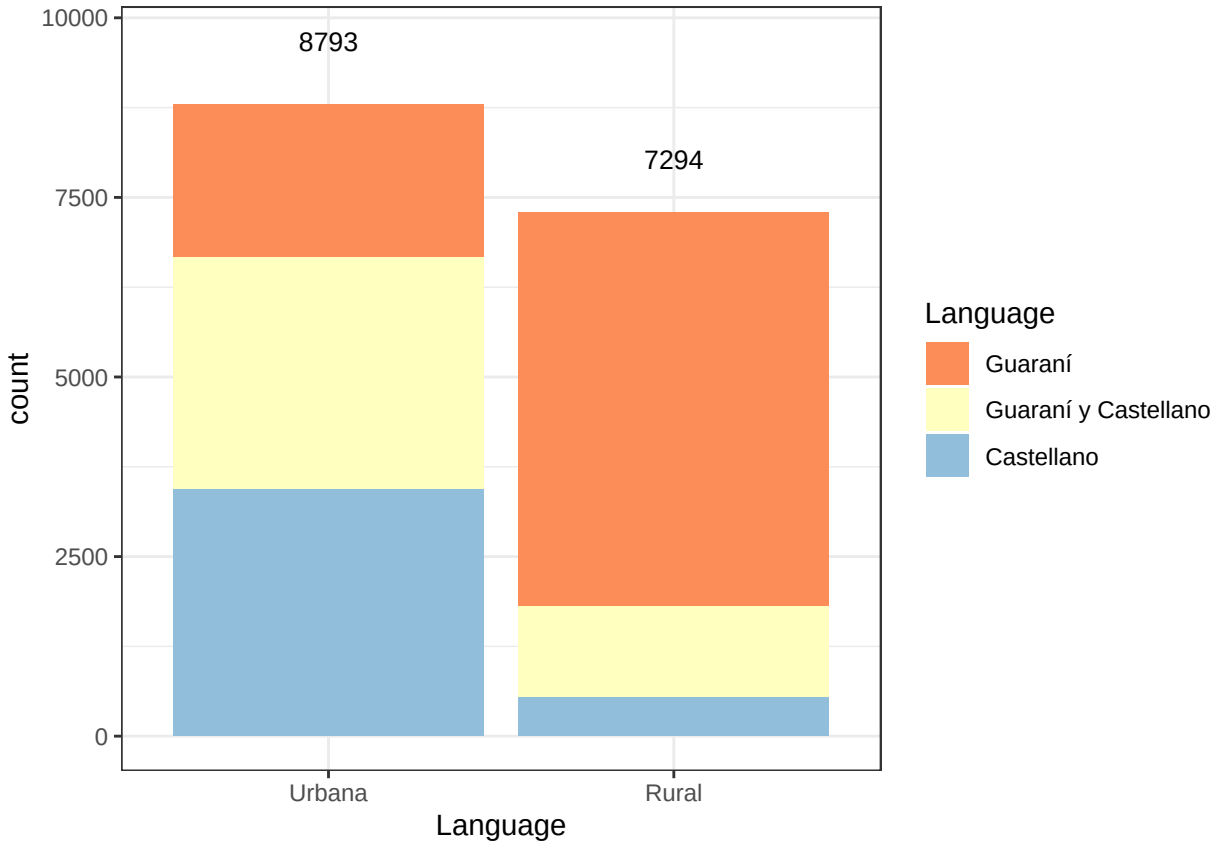
```
#Poverty Level
poverty <- eph_2019 %>% group_by(as_factor(pobnopoi)) %>% filter(añoest ==  12) %>% count(pobnopoi)
pvrty <-eph_2019 %>% group_by(as_factor(pobnopoi)) %>% filter(añoest %in% c("12")) %>% ggplot(aes(as_fa
pvrty + annotate(geom = "table", x = 3, y = 1500, label = list(poverty))
```



as_factor(pobnopoi)	pobnopoi	n
NO POBRE	0	1849
POBRE	1	261

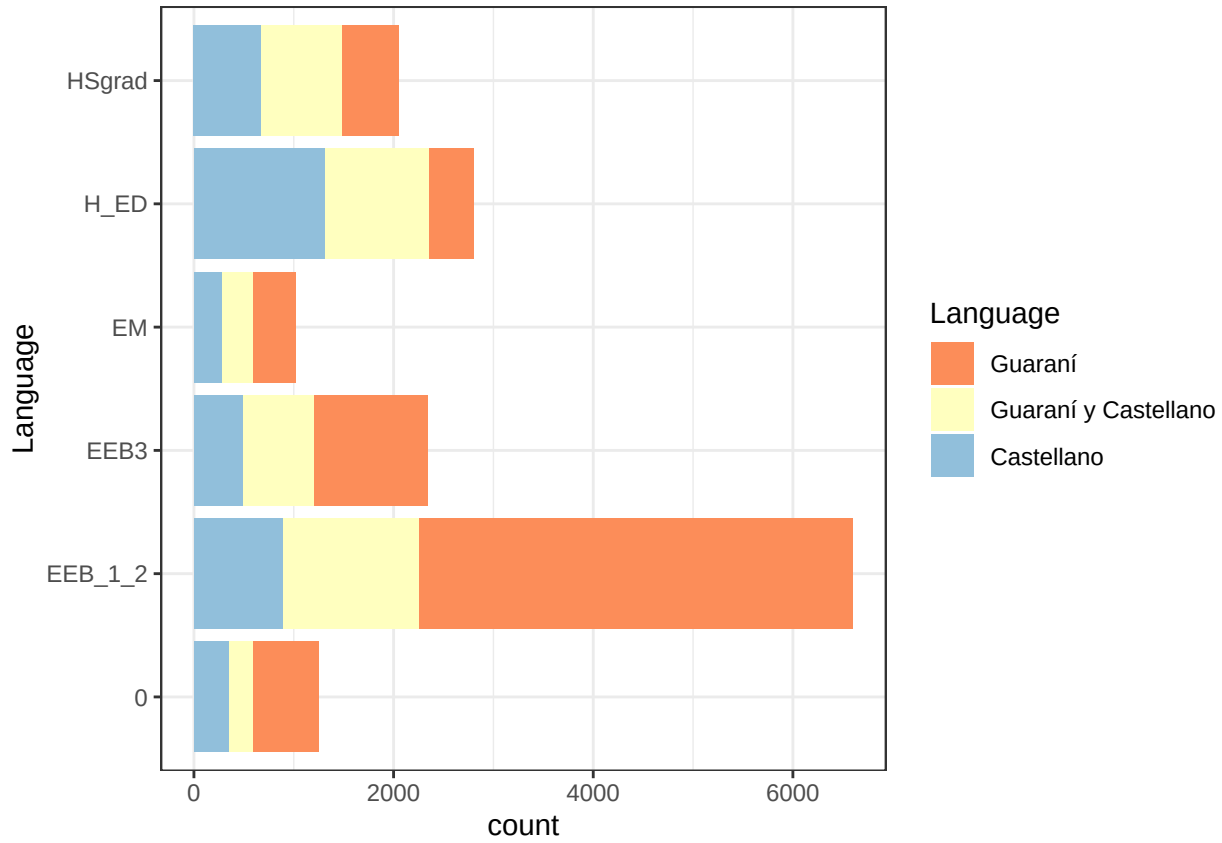
Most of our high-school graduates are considered "not poor" by the Poll.

```
#Area and language spoken
eph_2019 %>% filter(ED01 %in% c("1","2","3")) %>% ggplot(aes(x=as_factor(AREA)))+ geom_bar(aes(fill=as_
```



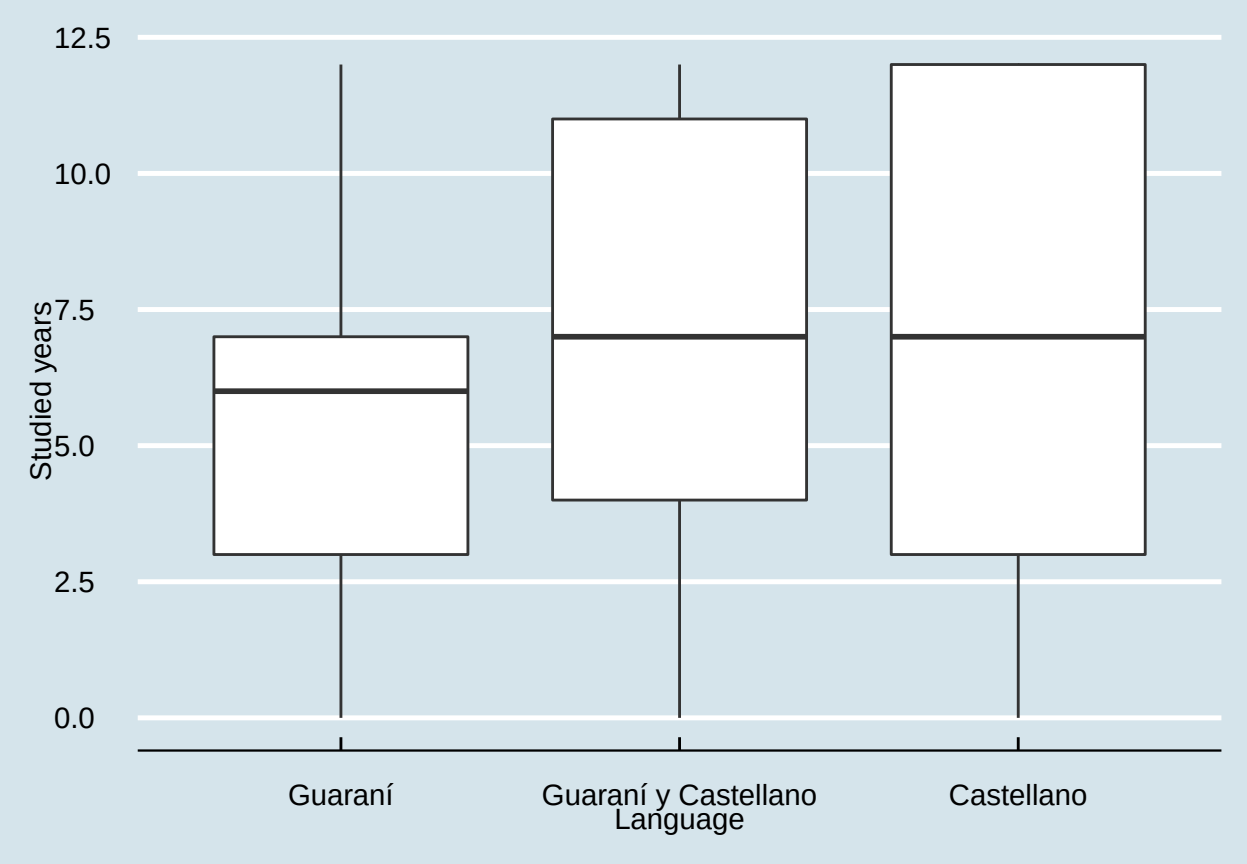
In Paraguay there are two official languages: spanish and guaraní. We can notice that guaraní is the common language in rural areas, while spanish has a greater role in urban areas.

```
#Last degree obtained and language spoken
eph_2019 %>% filter(ED01%in%c("1","2","3")) %>% ggplot(aes(graduate)) + geom_bar(aes(fill=as_factor(ED0
```



This graph show us the distribution of last academic titles achieved by language. For high-school graduates there is not a significant relevance shown.

```
#Years of study by language spoken
eph_2019 %>% filter(ED01 %in% c("1","2","3")) %>%
filter(añoest%in%c("0","1","2","3","4","5","6","7","8","9","10", "11","12")) %>%
group_by(ED01)%>%
ggplot(aes(as_factor(ED01), as.numeric(añoest)))+
geom_boxplot()+
ylab("Studied years")+
xlab("Language")+
theme_economist()
```



Unlike the last graph shown, this box-plot is more informative. We can observe that guaraní-speakers are less likely to have the 12 years or finishing high-school.

5 Modeling

In this data science project, we are working in a classification problem. As it was vaguely explained before, we are going to use the EPHC 2019 data set to train our models. We divide the data set in *train_set* (to train the models) and *test_set* (to test the models). Later, we use the EPHC 2020 data set for validation.

5.1 Preparation

```
set.seed(1, sample.kind = "Rounding")
eph_2019 <- eph_2019 %>% select(AREA, añoest, P02, pobnopoi, ED01, glyst_hs)
eph_2019 <- eph_2019 %>% mutate_at(
  vars("AREA", "añoest", "pobnopoi", "ED01", "glyst_hs"),
  funs(as_factor(.))
)
for (i in 1:length(eph_2019$añoest))
  {
  eph_2019$añoest[i]<-ifelse(eph_2019$añoest[i]=="Sin instrucción", 0, eph_2019$añoest[i])
}
eph_2019$añoest<-as.numeric(eph_2019$añoest)
eph_2019$P02<-as.numeric(eph_2019$P02)
```

```r
eph_2019<- eph_2019 %>%  # Create glyst_hs variable
  filter(!is.na(añoest))

eph_2019 <- eph_2019 %>% select(AREA, añoest, P02, pobnopoi, ED01, glyst_hs)
test_index <- createDataPartition(eph_2019$glyst_hs, times=1, p=0.2, list = F)
train_set <- eph_2019[-test_index,]
test_set <- eph_2019[test_index,]
model_weights <- ifelse(train_set$glyst_hs == "<12",
                        (1/table(train_set$glyst_hs)[1]) * 0.5,
                        (1/table(train_set$glyst_hs)[2]) * 0.5)
sum(model_weights)#The sum must equal 1
```

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

```r
rm(test_index)
```

5.2 SVM model (Support Vector Machine)

The SVM is a supervised model for classification algorithm that produces significant accuracy with less computation power. The objective of the SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points, using the support vectors to maximize the margin of the classifier.

The dependent variable we are going to use is glyst_hs.

The independent variables that we are going to use for the models are: *AREA* P02 *pobnopoi* ED01

```r
# SVM Model
svm.eph = svm(glyst_hs ~AREA+P02+pobnopoi+ED01, data = train_set)
test_set$pred.value = predict(svm.eph, newdata = test_set,type="response")
confusionMatrix(test_set$glyst_hs, test_set$pred.value)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  <12 >=12
##       <12  2109  217
##       >=12  346  590
##
##                Accuracy : 0.8274
##                  95% CI : (0.814, 0.8402)
##     No Information Rate : 0.7526
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5601
##
##  Mcnemar's Test P-Value : 6.869e-08
##
##             Sensitivity : 0.8591
##             Specificity : 0.7311
##          Pos Pred Value : 0.9067
##          Neg Pred Value : 0.6303
##              Prevalence : 0.7526
```

```
##          Detection Rate : 0.6465
##    Detection Prevalence : 0.7131
##       Balanced Accuracy : 0.7951
##
##        'Positive' Class : <12
##
```

```
results <- data.frame(
  Model="SVM (Support Vector Machine)",
  Accuracy=
    Accuracy(test_set$glyst_hs, test_set$pred.value),
  F1Score=
    F1_Score(test_set$glyst_hs, test_set$pred.value),
  Specificity=
   specificity(test_set$glyst_hs, test_set$pred.value),
  Sensitivity=
    sensitivity(test_set$glyst_hs, test_set$pred.value))
results
```

```
##                          Model  Accuracy   F1Score Specificity Sensitivity
## 1 SVM (Support Vector Machine) 0.8274065 0.8822422   0.7311029   0.8590631
```

We can observe that the SVM model has a good accuracy in general for all the scores. The F1 score is the highest. ## Decision Tree

```
# Applying Decision Tree Model
detree <- rpart(glyst_hs ~
                  AREA + P02 + pobnopoi + ED01,
                data = train_set)
# Prediction of data and Confusion Matrix
test_set$pred.value2 = predict(detree, newdata = test_set, type="class")
confusionMatrix(test_set$glyst_hs, test_set$pred.value2)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  <12 >=12
##       <12  2082  244
##       >=12  336  600
##
##                Accuracy : 0.8222
##                  95% CI : (0.8086, 0.8352)
##     No Information Rate : 0.7413
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.5523
##
##  Mcnemar's Test P-Value : 0.0001577
##
##             Sensitivity : 0.8610
##             Specificity : 0.7109
##          Pos Pred Value : 0.8951
##          Neg Pred Value : 0.6410
```

```
##              Prevalence : 0.7413
##          Detection Rate : 0.6383
##    Detection Prevalence : 0.7131
##       Balanced Accuracy : 0.7860
##
##        'Positive' Class : <12
##
```

```
results<- bind_rows(results,
  data.frame(Model="Decision Tree",
             Accuracy=
               Accuracy(test_set$glyst_hs, test_set$pred.value2),
             F1Score=
               F1_Score(test_set$glyst_hs, test_set$pred.value2),
             Specificity=
               specificity(test_set$glyst_hs, test_set$pred.value),
             Sensitivity=
               sensitivity(test_set$glyst_hs, test_set$pred.value)))
results
```

```
##                          Model  Accuracy   F1Score Specificity Sensitivity
## 1 SVM (Support Vector Machine) 0.8274065 0.8822422   0.7311029   0.8590631
## 2                Decision Tree 0.8221950 0.8777403   0.7311029   0.8590631
```

With the Decision Tree model we can observe good accuracy in general in the scores. The F1 score is the highest in this model too.

6 Validation

We have train two models and the SVM model is the most accurate among the two. So, in the validation phase we are going to use the SVM model.

6.1 Preparation

```
set.seed(1, sample.kind = "Rounding")
eph_2020 <- eph_2020 %>% select(AREA, añoest, P02, pobnopoi, ED01, glyst_hs)
eph_2020 <- eph_2020 %>% mutate_at(
  vars("AREA", "añoest", "pobnopoi", "ED01", "glyst_hs"),
  funs(as_factor(.))
)
for (i in 1:length(eph_2020$añoest))
  {
  eph_2020$añoest[i]<-ifelse(eph_2020$añoest[i]=="Sin instrucción", 0, eph_2020$añoest[i])
}
eph_2020$añoest<-as.numeric(eph_2020$añoest)
eph_2020$P02<-as.numeric(eph_2020$P02)
eph_2020<- eph_2020 %>%  # Create glyst_hs variable
  filter(!is.na(añoest))

eph_2020 <- eph_2020 %>% select(AREA, añoest, P02, pobnopoi, ED01, glyst_hs)
```

```
model_weights <- ifelse(eph_2020$glyst_hs == "<12",
                        (1/table(eph_2020$glyst_hs)[1]) * 0.5,
                        (1/table(eph_2020$glyst_hs)[2]) * 0.5)
sum(model_weights)#The sum must equal 1
```

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

6.2 SVM Validation

```
svm.eph = svm(glyst_hs ~
                  AREA +  P02 + pobnopoi + ED01,
                data = eph_2020)
eph_2020$pred.value = predict(svm.eph, newdata = eph_2020,type="response")
ConfusionMatrix(eph_2020$glyst_hs, eph_2020$pred.value)
```

```
##       y_pred
## y_true   <12  >=12
##   <12  10398  2046
##   >=12   921  2490
```

```
results<- bind_rows(
  results,
  data.frame(Model="Validation - SVM",
             Accuracy=
               Accuracy(eph_2020$glyst_hs, eph_2020$pred.value),
             F1Score=
               F1_Score(eph_2020$glyst_hs, eph_2020$pred.value),
             Specificity=
               specificity(test_set$glyst_hs, test_set$pred.value),
             Sensitivity=
               sensitivity(test_set$glyst_hs, test_set$pred.value)))
results
```

```
##                          Model  Accuracy   F1Score Specificity Sensitivity
## 1 SVM (Support Vector Machine) 0.8274065 0.8822422   0.7311029   0.8590631
## 2                Decision Tree 0.8221950 0.8777403   0.7311029   0.8590631
## 3             Validation - SVM 0.8128666 0.8751420   0.7311029   0.8590631
```

We see that for the SVM Validation model the accuracy is a little smaller than in the train_set data set. However, it is a good score and the F1 score is the highest again. # Results The results table is a summary of all models we have done so far. As it was described before, the Support Vector Machine model is the most appropriate model for our data science model. The decision tree has a big score, but is not better than the SVM model.

```
results
```

```
##                          Model  Accuracy   F1Score Specificity Sensitivity
## 1 SVM (Support Vector Machine) 0.8274065 0.8822422   0.7311029   0.8590631
## 2                Decision Tree 0.8221950 0.8777403   0.7311029   0.8590631
## 3             Validation - SVM 0.8128666 0.8751420   0.7311029   0.8590631
```

The **SVM Model** is better in the accuracy and F1 Score. The **Decision Tree** has a minor score, but the diference is not very large. Using the **SVM Model** for validation has less score than in the train_Set.

7 Conclusion

At first, we loaded the EPHC datasets from 2019 and 2020. We prepared the EPHC 2019 dataset for training purposes and then we used the EHPC 2020 for validation. We explored the data and selected a few variables that were more significant for visualising. We did some wrangling for better data visualization and observed some relevant information. Finally, we proceeded to the modeling phase to train the models and figure out which model is the best for this project.

7.0.1 Limitations

The two models used for the project had satisfactory results, using more models could overtrain the data set and give wrong results.

7.0.2 Future work

The EPHC data base is huge and has a lot of variables. Some variables could be use for modeling from a different perception and generate interesting results. There are variables such as access to Wi-Fi connection, genre, income that after some wrangling could be use in the models. Some variables I could not use because of the limitations from my notebook's features. My Specificity and Sensitivity score did not change, so I encourage for future work to figure out why the results are the same in the different models.