Start-Up Global

A comprehensive analysis on the Ease of Doing Business using Deep Learning

https://neonflux56.shinyapps.io/EODB/

Project by -

Xianya Xiong

Qinyi Wu

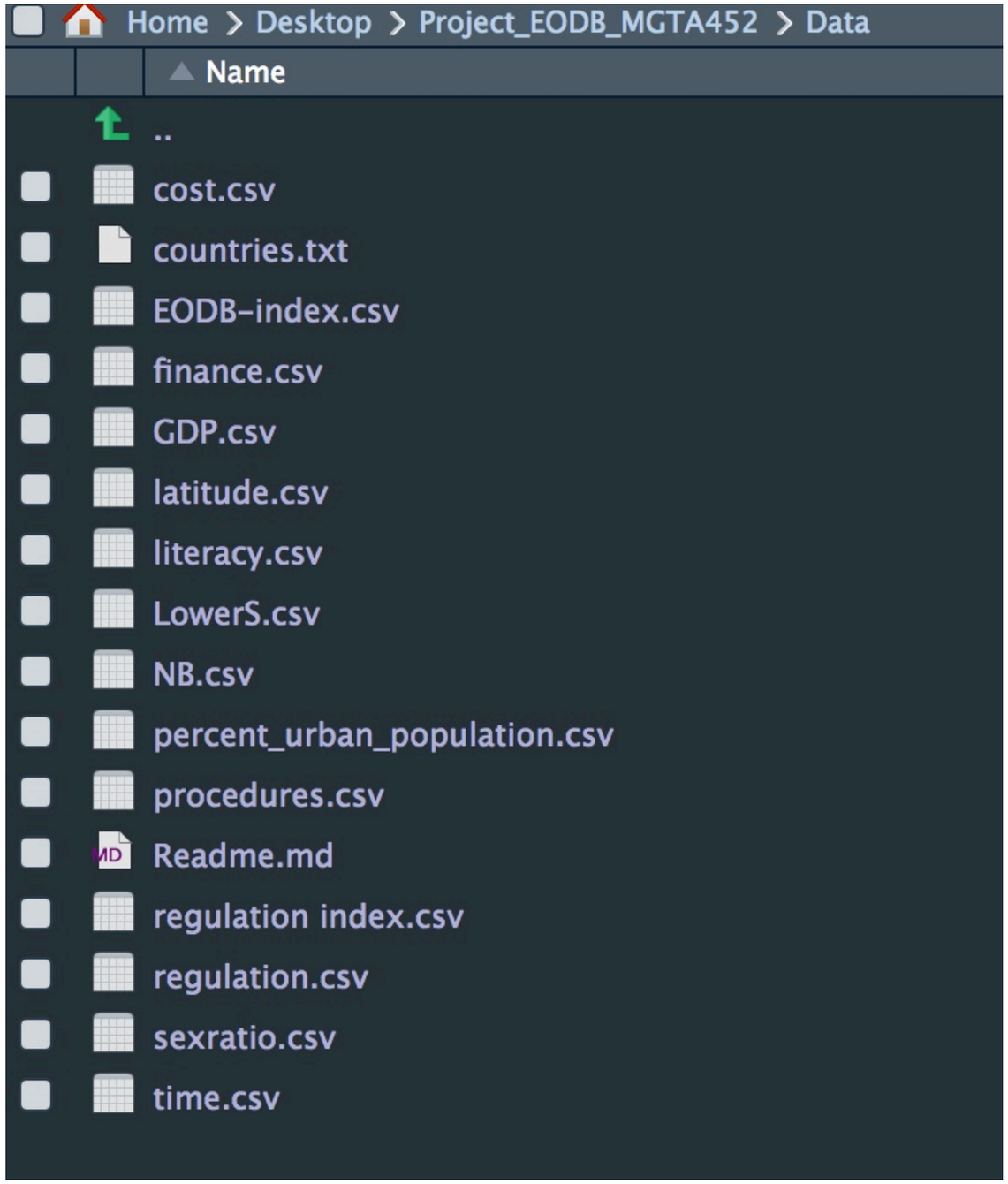
John Shen

Ashish Gupta

Ease of Doing Business

Features Extracted for years 2006 to 2019

- **GDP per capita** : Gross domestic product divided by midyear population.
- **NB** : Number of new limited liability corporations registered in the calendar year.
- **Cost** : Cost to register a business normalized by presenting it as a percentage of gross national income (GNI) per capita
- **LowerSecondary** : Ratio of people entering last grade of lower secondary education to the actual population age entering same grade.
- **Percent_Urban_Population** : Urban population ratio with respect to the whole population.
- **Procedures** : Number of start-up procedures required to start a business
- **Sexratio** : Male to female ratio.
- **Time_days** : Number of days required to start the business



https://data.worldbank.org/indicator

http://www.countries-list.info/Download-List

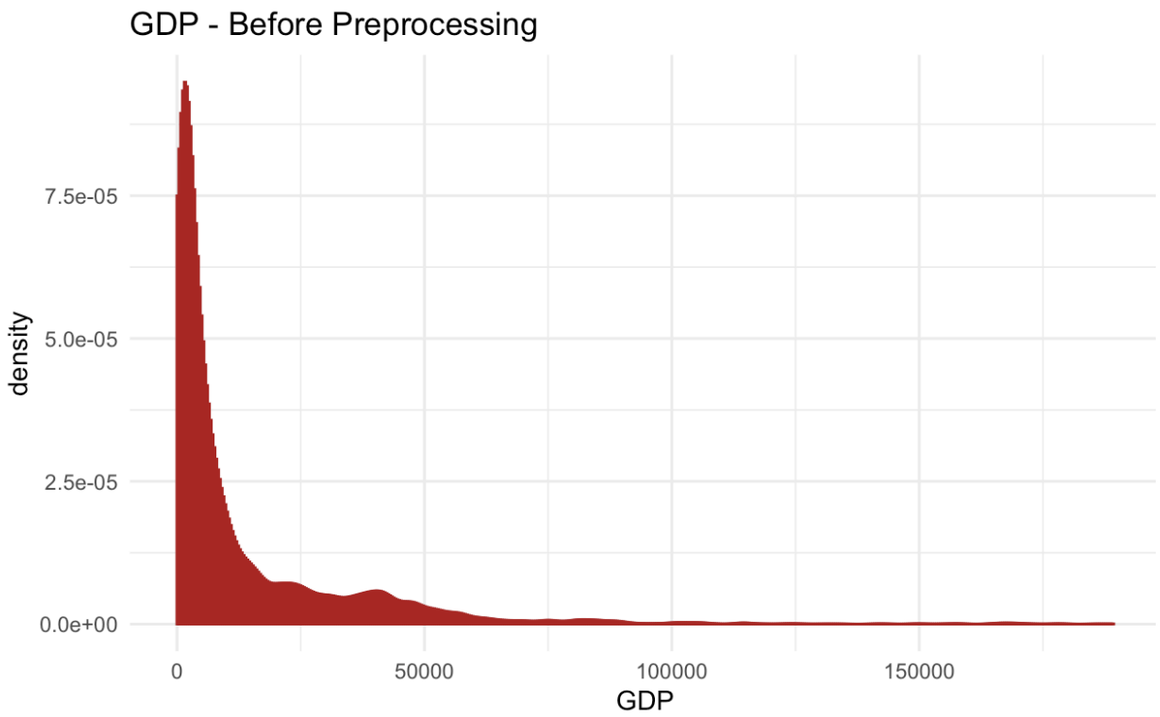
https://www.doingbusiness.org/en/custom-query

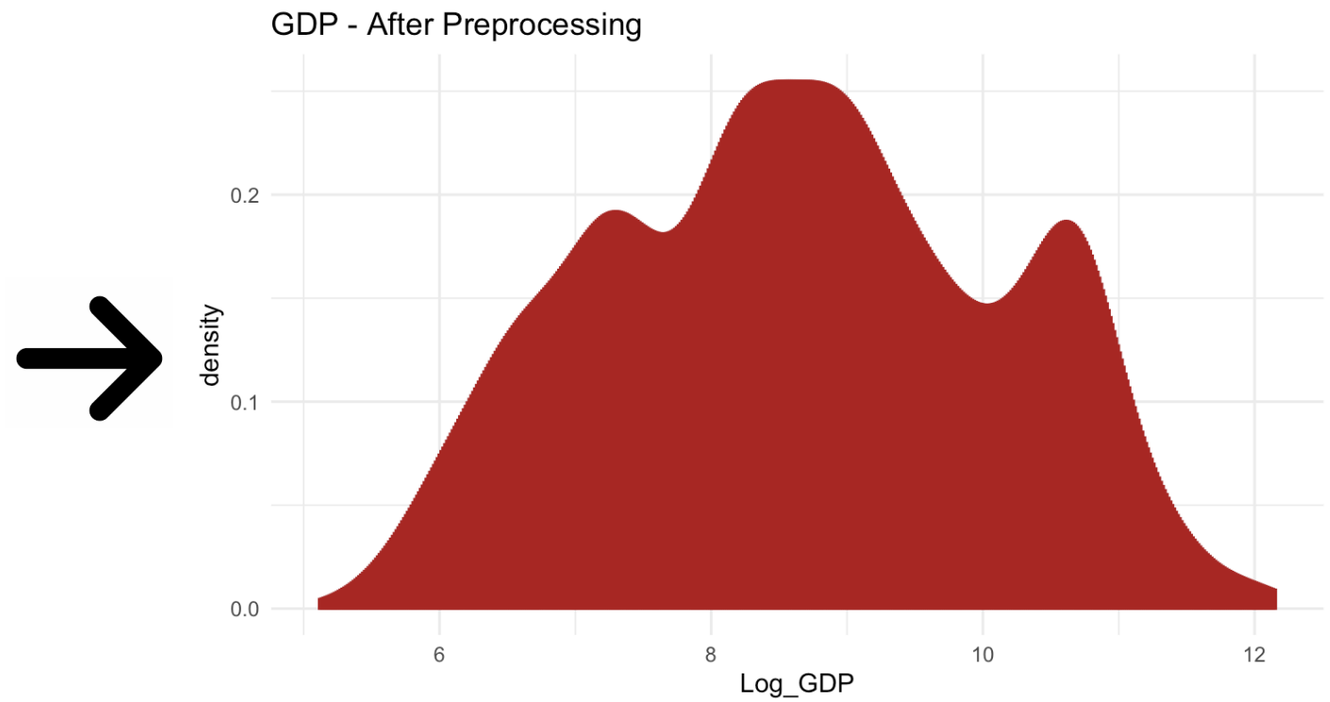
The Question now is …

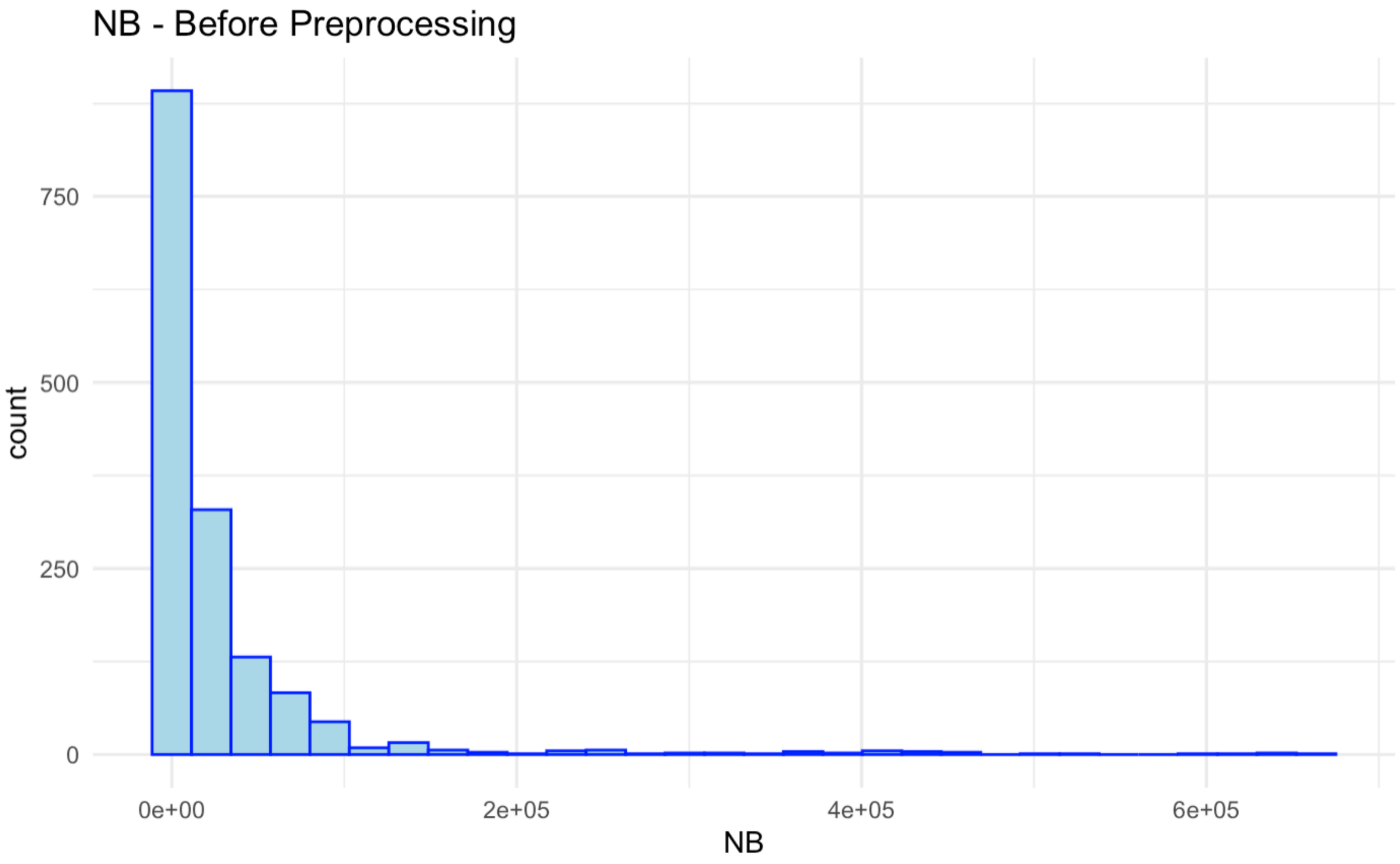
- How do we build a model?
- Response Variable?
- Train model for years 2006-2018 and predict for 2019

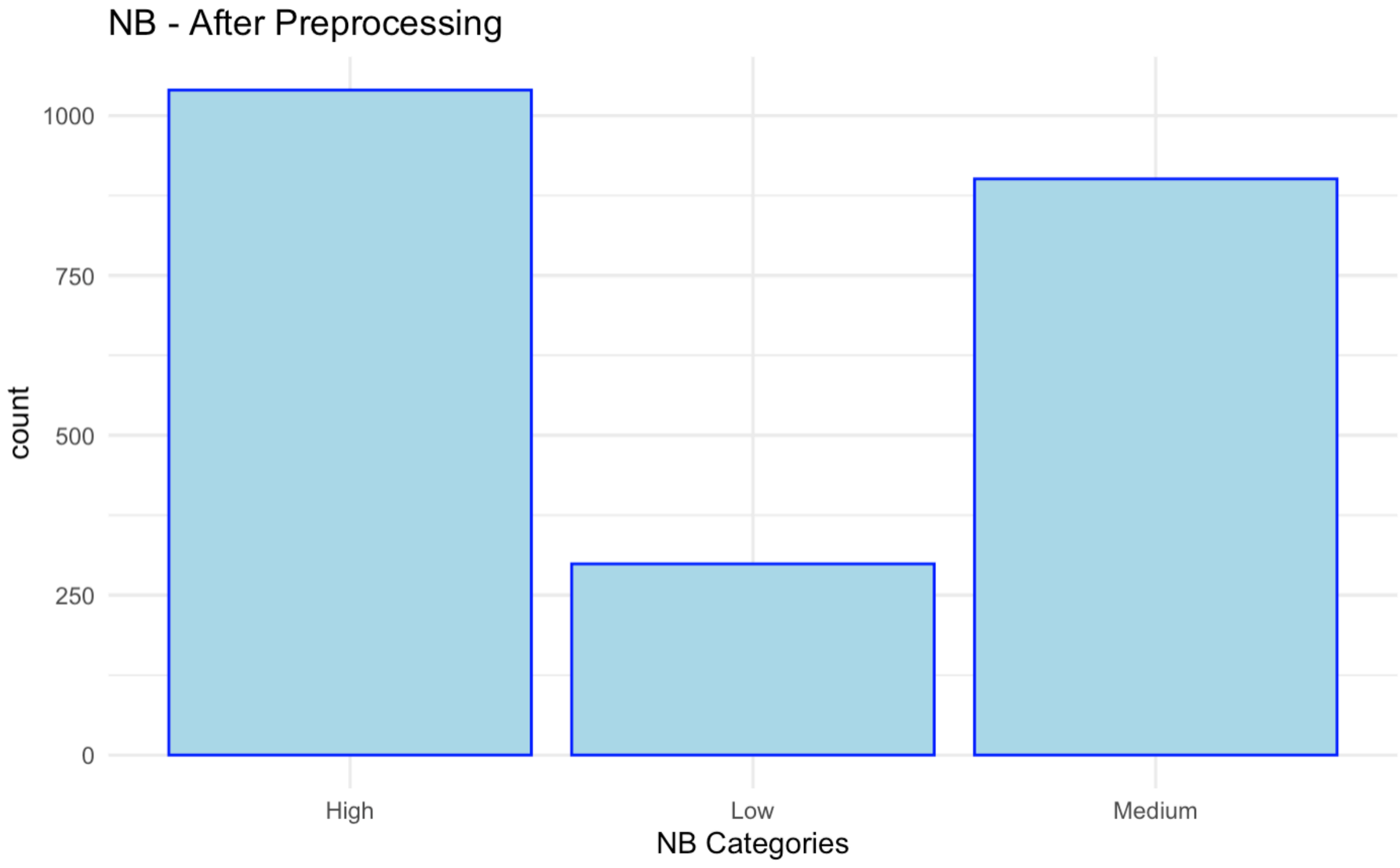
Classic Classification problem!

Transformations

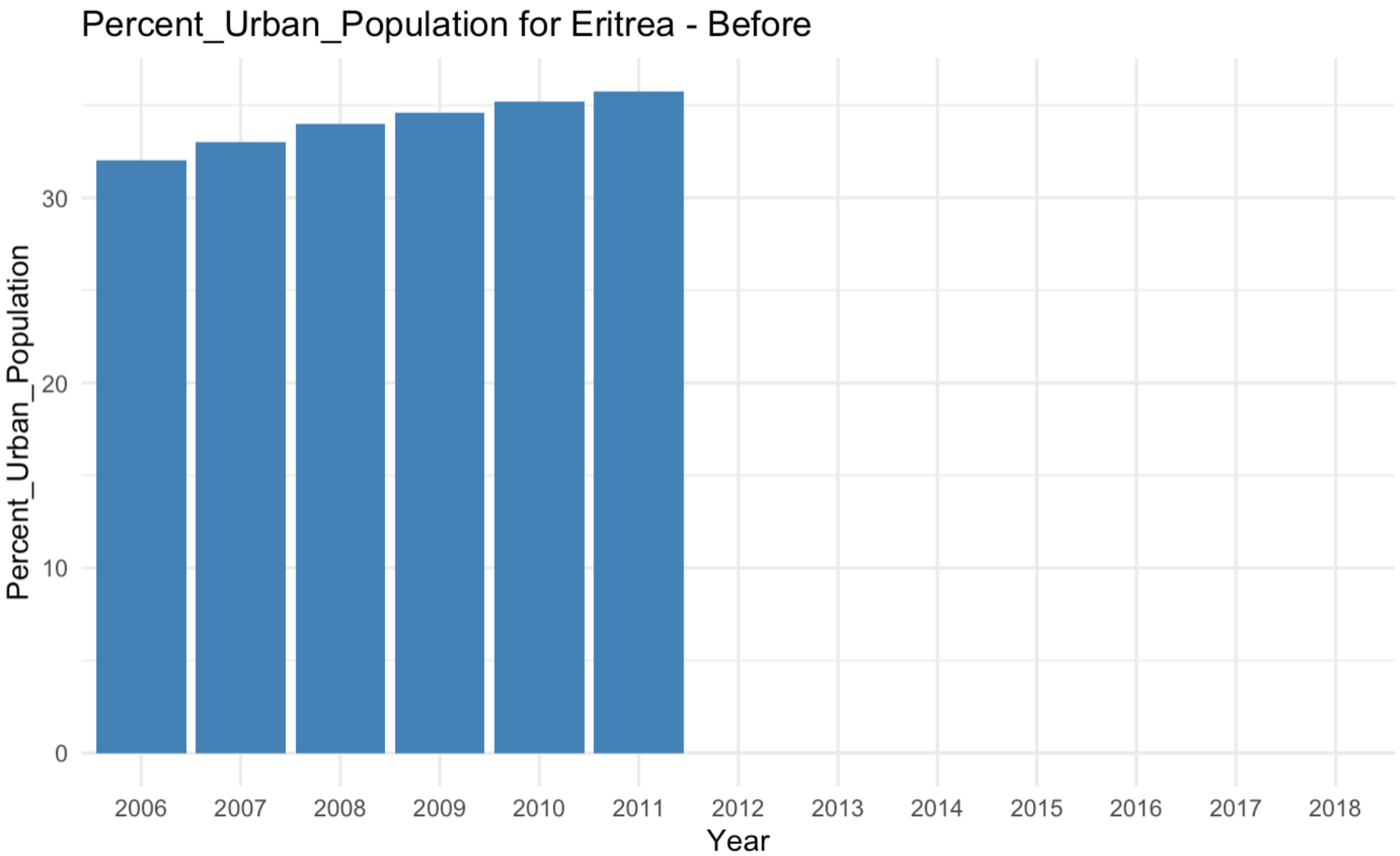


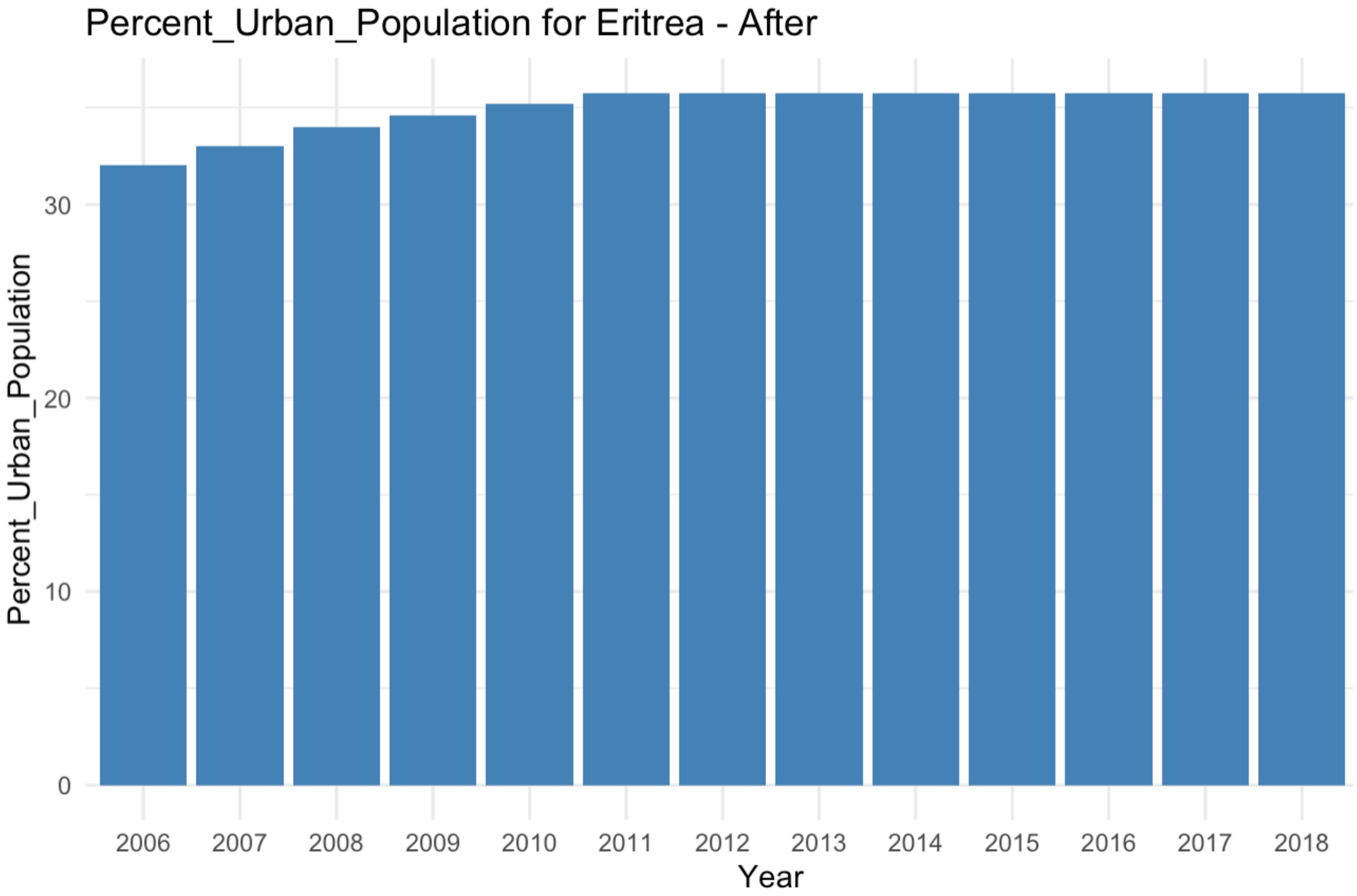


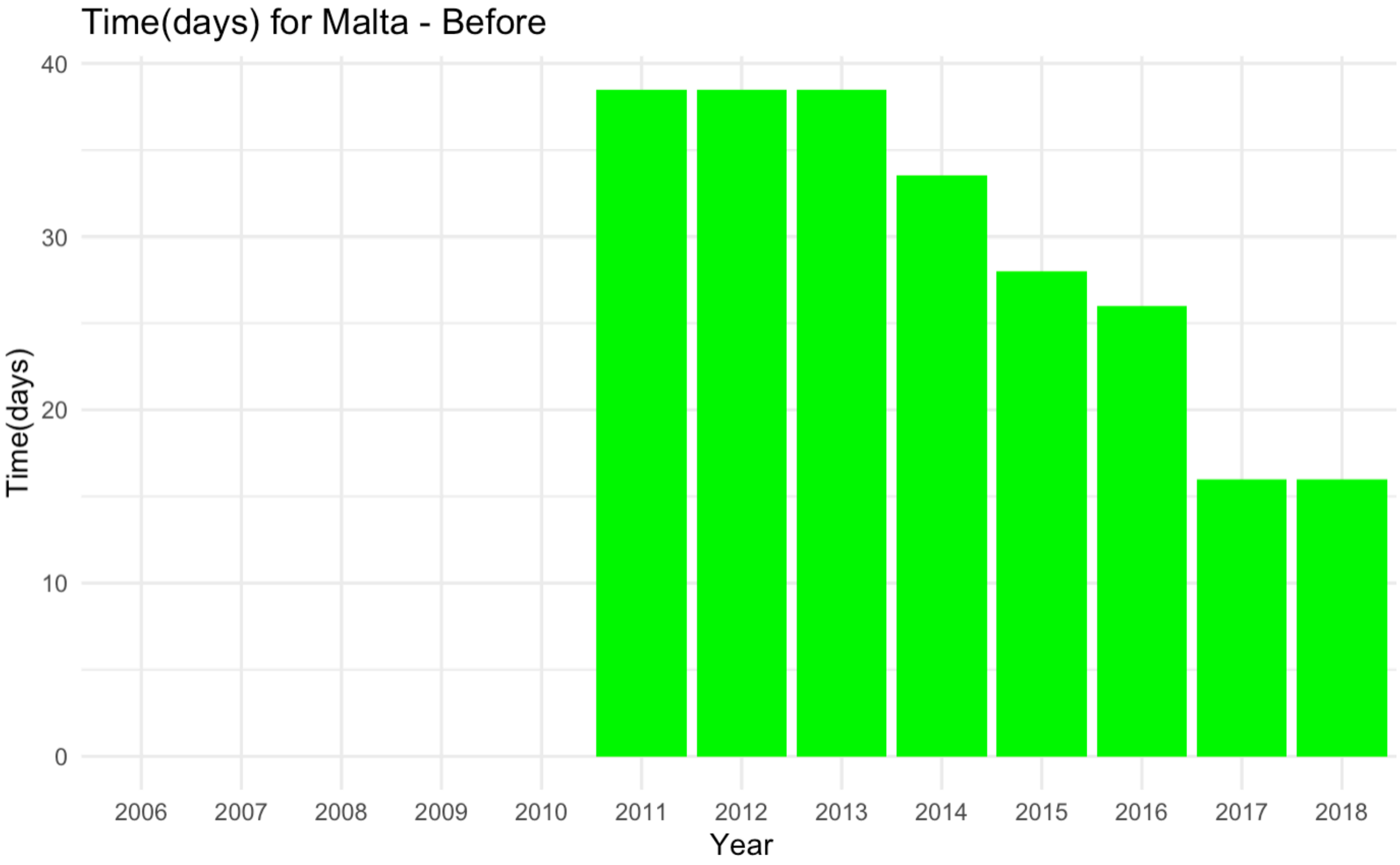

NB - Before Preprocessing
count
750
500
250
0
0e+00
2e+05
4e+05
6e+05
NB

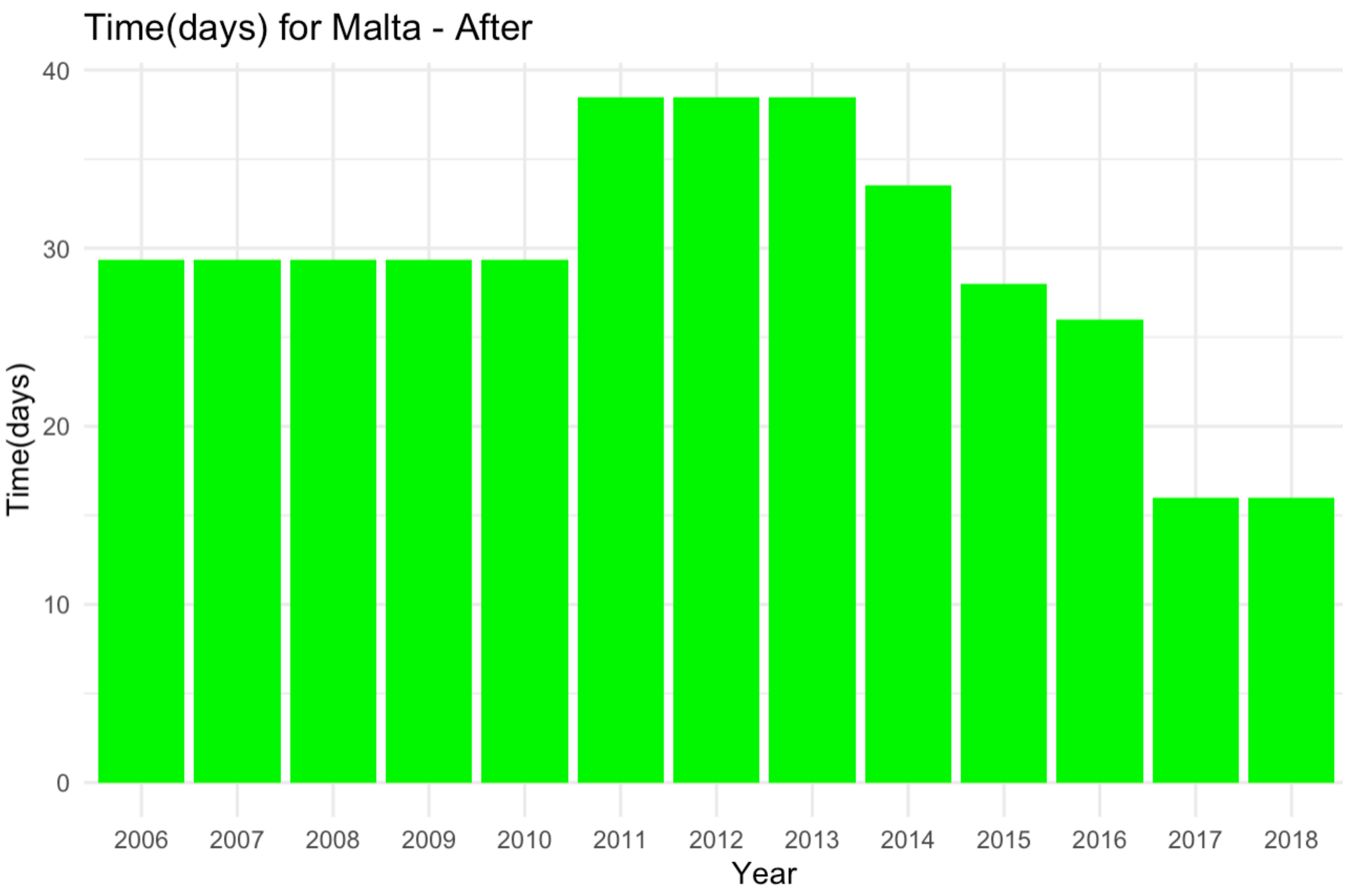

NB - After Preprocessing
1000
750
500
250
0
count
High
Low
Medium
NB Categories

Dealing with NA values




Percent_Urban_Population for Eritrea - After
Percent_Urban_Population
30
20
10
0
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
Year


Time(days) for Malta - Before
40
30
20
10
0
Time(days)
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
Year


Time(days) for Malta - After
40
30
20
10
0
Time(days)
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
Year

Updated dataset

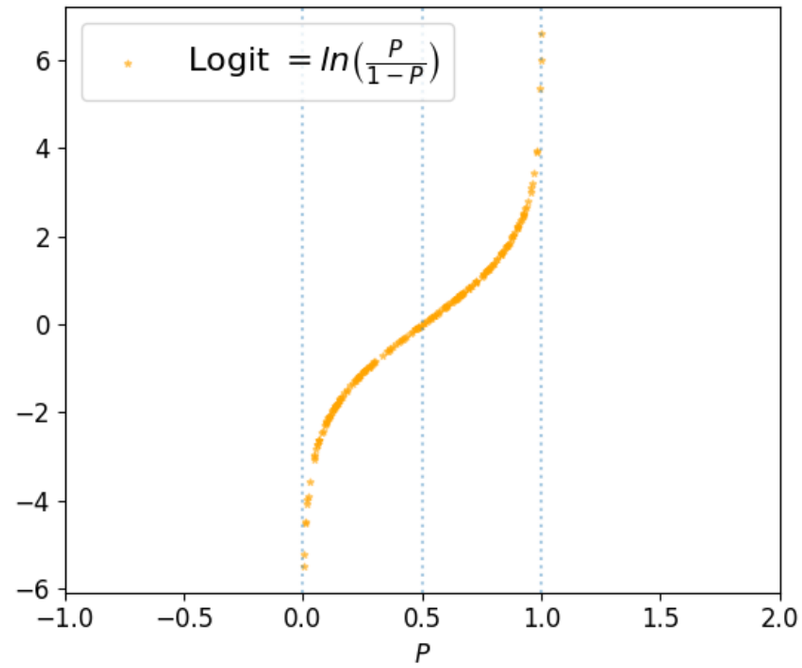
```
'data.frame':   2240 obs. of  14 variables:
 $ index                     : num  1 2 3 4 5 6 7 8 9 10 ...
 $ Country                   : Factor w/ 291 levels "","Afghanistan",..: 2 2 2 2 2 2 2 2 2
2 ...
 $ Code                      : Factor w/ 264 levels "ABW","AFG","AGO",..: 2 2 2 2 2 2 2 2 2
2 ...
 $ Year                      : chr  "2019" "2018" "2017" "2016" ...
 $ Cost                      : chr  "High" "High" "Medium Low" "Medium Low" ...
 $ Log_GDP                   : num  6.26 6.26 6.32 6.3 6.36 ...
 $ procedures                : num  4 4 4 4 4 4 4 5 5 5 ...
 $ time_days                 : num  8.5 8.5 8.5 8.5 8.5 8.5 6.5 9.5 9.5 9.5 ...
 $ sexratio                  : num  1.06 1.06 1.06 1.06 1.06 1.06 1.06 1.06 1.06 1.06 ...
 $ NB                        : chr  "Medium" "Medium" "Medium" "Medium" ...
 $ LowerSecondary            : num  53.2 53.2 53.2 49.7 53.2 ...
 $ Percent_Urban_Population: num  25.5 25.5 25.2 25 24.8 ...
 $ EODB_index                : num  183 183 183 177 183 164 168 160 167 160 ...
 $ Y                         : num  2 2 2 2 2 2 2 2 2 2 ...
```

Logit Regression

- library(glmnet)
- Using the binomial link **logit** function

$$Ln\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k$$

- Training set: Years 2006-2017
- Validation set: Year 2018

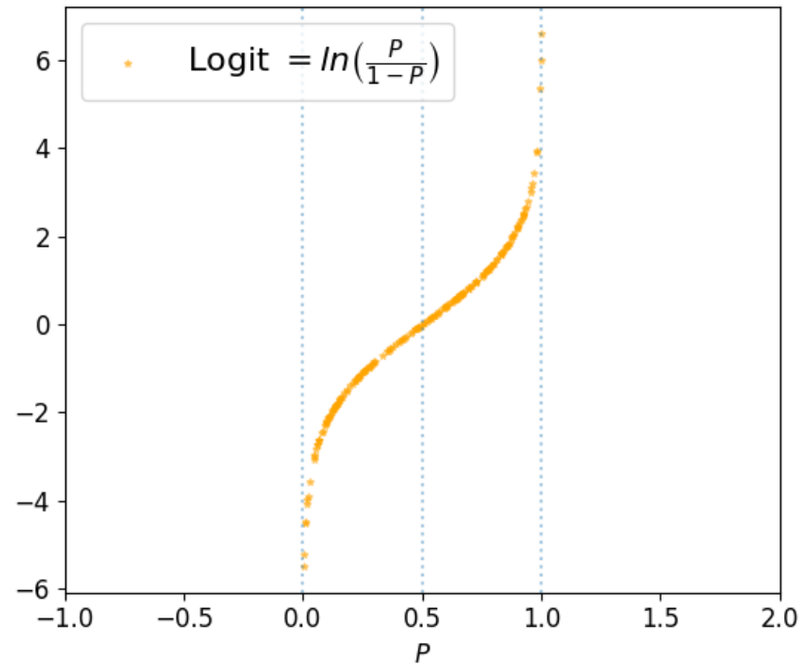


Logit Regression

- library(glmnet)
- Using the binomial link **logit** function

$$Ln\left(\frac{P}{1-P}\right) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k$$

- Training set: Years 2006-2017
- Validation set: Year 2018



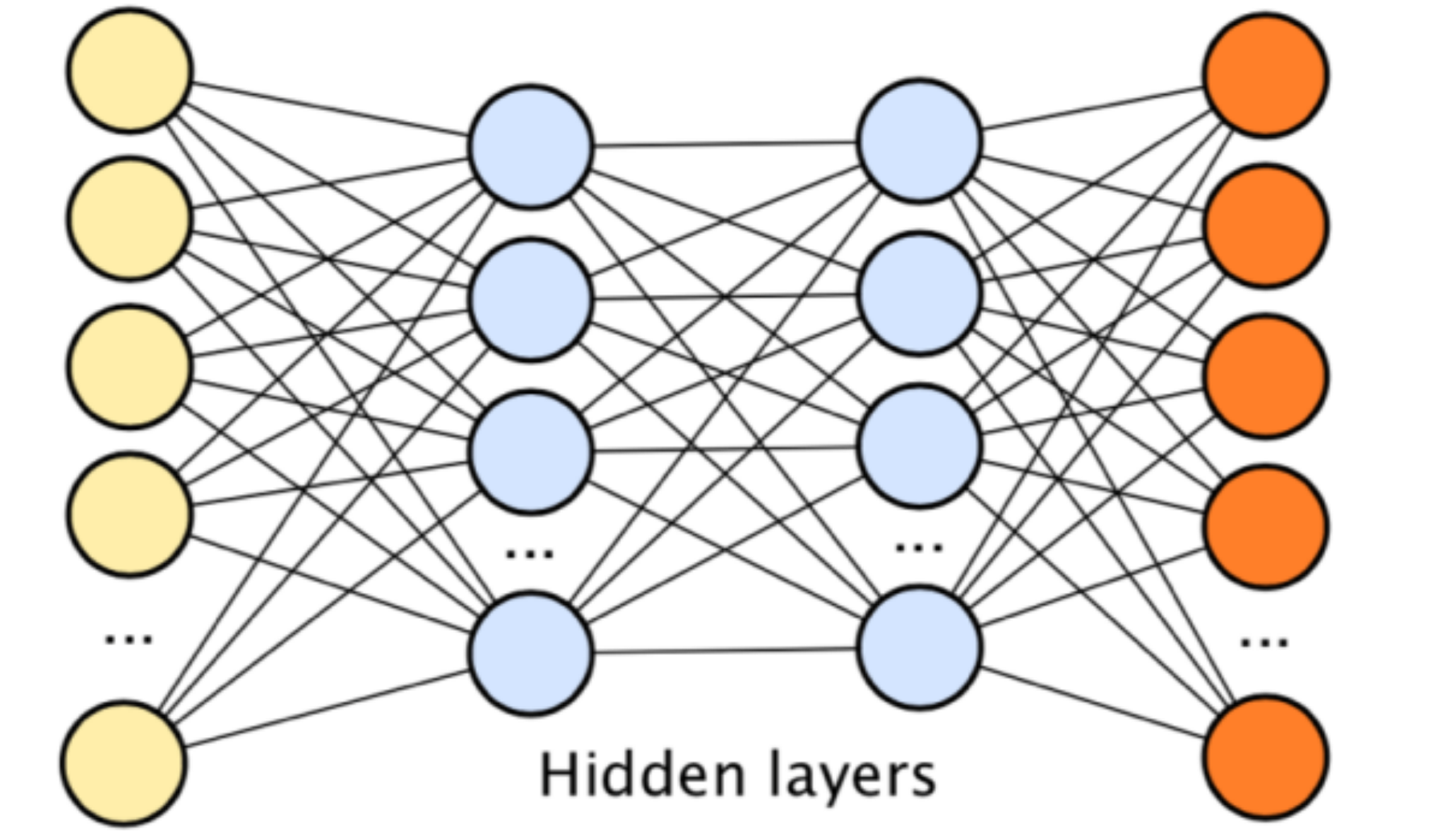
Result:

```
       PredStatusLogit
actual  1  2
     1 80  6
     2 31 43
```

```
[1] "Accuracy using logit regression:  0.76875"
```

Deep Learning

Keras



- library(recipe)
- Convert categorical predictors into one hot encoding
- Center and scale all predictors

```
Classes ‘tbl_df’, ‘tbl’ and 'data.frame':       2080 obs. of  173 variables:
 $ Log_GDP                          : num  -1.52 -1.48 -1.49 -1.45 -1.41 ...
 $ procedures                       : num  -1.26 -1.26 -1.26 -1.26 -1.26 ...
 $ time_days                        : num  -0.443 -0.443 -0.443 -0.443 -0.443 ...
 $ sexratio                         : num  0.353 0.353 0.353 0.353 0.353 ...
 $ LowerSecondary                   : num  -0.899 -0.899 -1.034 -0.899 -0.899 ...
 $ Percent_Urban_Population         : num  -1.35 -1.36 -1.37 -1.38 -1.39 ...
 $ Country_Afghanistan              : num  12.6 12.6 12.6 12.6 12.6 ...
 $ Country_Albania                  : num  -0.0793 -0.0793 -0.0793 -0.0793 -0.0793 ...
 $ Country_Algeria                  : num  -0.0793 -0.0793 -0.0793 -0.0793 -0.0793 ...
 $ Country_Angola                   : num  -0.0793 -0.0793 -0.0793 -0.0793 -0.0793 ...
 $ Country_Antigua.and.Barbuda      : num  -0.0793 -0.0793 -0.0793 -0.0793 -0.0793 ...
 $ Country_Argentina                : num  -0.0793 -0.0793 -0.0793 -0.0793 -0.0793 ...
 $ Country_Armenia                  : num  -0.0793 -0.0793 -0.0793 -0.0793 -0.0793 ...
 $ Country_Australia                : num  -0.0793 -0.0793 -0.0793 -0.0793 -0.0793 ...
 $ Country_Austria                  : num  -0.0793 -0.0793 -0.0793 -0.0793 -0.0793 ...
 $ Country_Azerbaijan               : num  -0.0793 -0.0793 -0.0793 -0.0793 -0.0793 ...
 $ Country_Bahrain                  : num  -0.0793 -0.0793 -0.0793 -0.0793 -0.0793 ...
 $ Country_Bangladesh               : num  -0.0793 -0.0793 -0.0793 -0.0793 -0.0793 ...
 $ Country_Barbados                 : num  -0.0793 -0.0793 -0.0793 -0.0793 -0.0793 ...
 $ Country_Belarus                  : num  -0.0793 -0.0793 -0.0793 -0.0793 -0.0793 ...
 $ Country_Belgium                  : num  -0.0793 -0.0793 -0.0793 -0.0793 -0.0793 ...
```

- One hidden layer with **80 units** and one output layer with **3 units**
- 80*173 + 3*80 + 80 + 3 = **14,163** parameters to train
- The hidden layer uses **ReLU** activation and the output layer uses **Softmax** activation

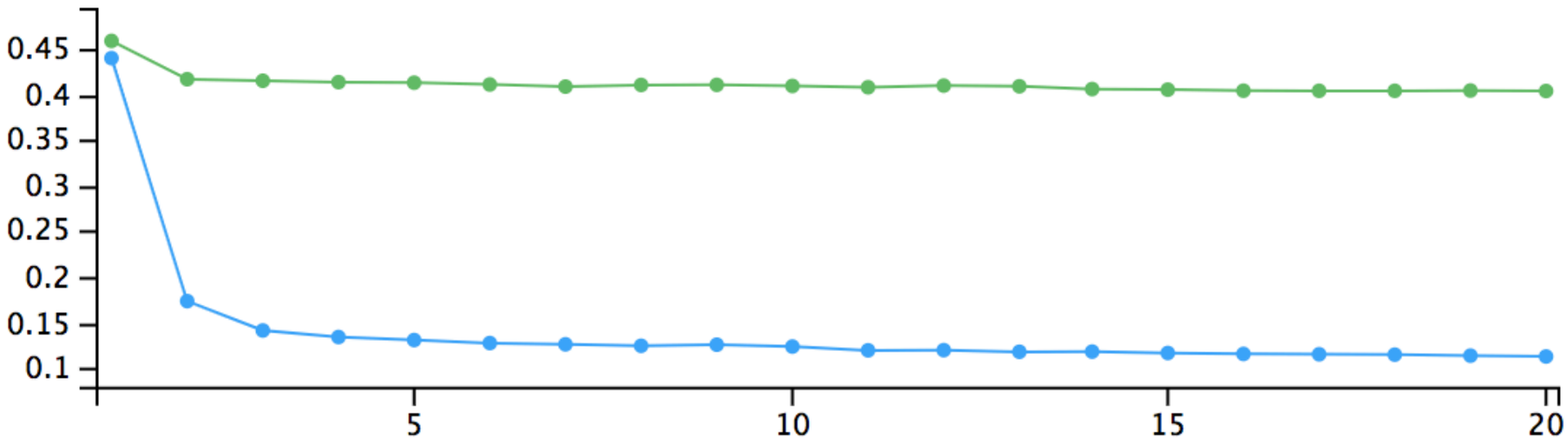
$$g(z) = \max\{0, z\}$$

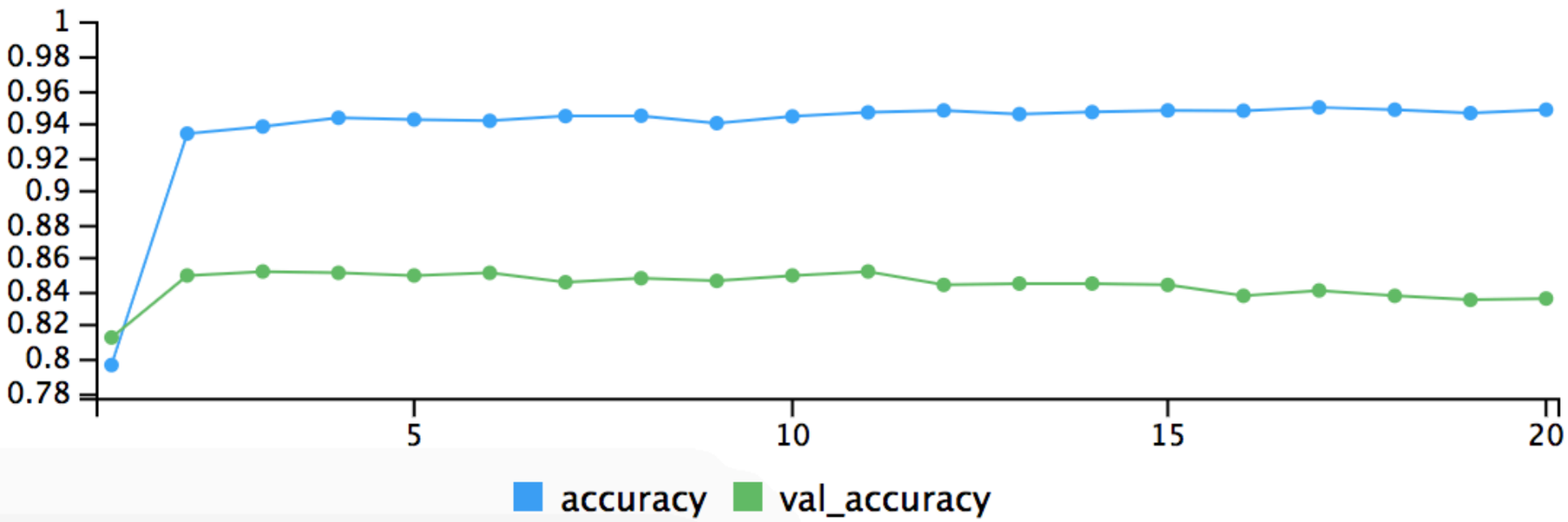
```
Model: "sequential_15"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_30 (Dense)             (None, 80)                13920
_________________________________________________________________
dense_31 (Dense)             (None, 3)                 243
=================================================================
Total params: 14,163
Trainable params: 14,163
Non-trainable params: 0
```

- Compile model with **binary cross-entropy** loss and **adam** optimiser

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log(p(y_i)) + (1 - y_i) \cdot log(1 - p(y_i))$$

- Train the model for training data 2006 - 2017 with 20% validation set





- Predict for the year 2018…

- Predict for the year 2018…

Result:

```
       pred
actual  1  2
     1 83  3
     2 15 59
```

```
[1] "Accuracy using deep learning:  0.8875"
```

- Predict for the year 2018…

Result:

```
       pred
actual   1  2
     1 83  3
     2 15 59
```

```
[1] "Accuracy using deep learning:  0.8875"
```

- Run the model for year 2019 and visualise results using **Shiny**