Risk Models Development Process

Jakub Szotek

November, 2019

1. 1. Data Preparation
2. 2. Analysis of risk parameters
3. 3. Data split
4. 4. Model functional form
5. 5. Multiple Factor Analysis
6. 6. Model selection
7. 7. Model validation

1. Data Preparation

Data Preparation

- Model population
- Observation window
- Observation level
- Review of data systems
- Source data

Load data

```
Data <- read.csv('https://raw.githubusercontent.com/jakubszotek/Present
```

View data

```
print(head(Data,4), digits = 2, row.names = FALSE)
```

```
##  Customer_ID Date_of_data Default_date Country Industry
##            1   01/01/2014   24/11/2014      UK        A
##            2   01/01/2011                   UK        D
##            3   01/01/2018                   FR        A
##            4   01/01/2014   11/08/2014      FR        A
##  Length_of_business Total_assets Financial_leverage
##                 4.3          1.5                1.1
##                 8.7          9.8                1.2
##                 7.1          1.7                0.6
##                 5.6          6.2                1.2
##  Credit_limit   EDF GDP_growth Default
##          0.27 0.013       2.95       1
##          1.27 0.015       1.64       0
##          0.59 0.010       1.72       0
##          0.87 0.013       0.96       1
```

2. Analysis of risk parameters

All headers

```
sapply(Data, class)
```

```
##         Customer_ID        Date_of_data        Default_date
##           "integer"            "factor"            "factor"
##             Country            Industry Length_of_business
##            "factor"            "factor"           "numeric"
##        Total_assets Financial_leverage        Credit_limit
##           "numeric"           "numeric"           "numeric"
##                 EDF          GDP_growth             Default
##           "numeric"           "numeric"           "integer"
```

Default variable

```
head(Data,10) %>% select(Default_date, Default)
```

```
##    Default_date Default
## 1    24/11/2014       1
## 2                     0
## 3                     0
## 4    11/08/2014       1
## 5                     0
## 6    03/04/2012       1
## 7                     0
## 8                     0
## 9                     0
## 10                    0
```

Risk drivers

- Types of drivers:
  - Demographic
  - Financial
  - Behavioural
  - Macroeconomic
- Types of data:
  - Numerical
  - Boolean
  - Categorical

Risk drivers

```
Drivers <- Data %>% select(-Date_of_data,-Default_date)
print(head(Drivers,5), digits = 2, row.names = FALSE)
```

Risk drivers

```
##   Customer_ID Country Industry Length_of_business
## 1           1      UK        A                4.3
## 2           2      UK        D                8.7
## 3           3      FR        A                7.1
## 4           4      FR        A                5.6
## 5           5      UK        A                2.3
##   Total_assets Financial_leverage Credit_limit   EDF
## 1          1.5                1.1         0.27 0.013
## 2          9.8                1.2         1.27 0.015
## 3          1.7                0.6         0.59 0.010
## 4          6.2                1.2         0.87 0.013
## 5         15.9                1.3         1.87 0.018
##   GDP_growth Default
## 1       2.95       1
## 2       1.64       0
## 3       1.72       0
## 4       0.96       1
## 5       1.45       0
```

Single Factor Analysis - Univariate

- We exclude all variables having more than 10% of missing values
- Is there enough variance for each variable?

```
var_summary <- summary(Drivers %>% select(Country, Industry,
                                          Length_of_business,
                                          Total_assets,
                                          Financial_leverage,
                                          Credit_limit,
                                          EDF,
                                          GDP_growth))
```

Single Factor Analysis - Univariate

```
##  Country   Industry Length_of_business  Total_assets
##  DE:164    A:387    Min.    : 0.150     Min.    : 0.020
##  FR:302    B:106    1st Qu.: 2.670     1st Qu.: 1.278
##  PL:157    C:292    Median : 4.280     Median : 2.375
##  UK:377    D:215    Mean    : 4.915     Mean    : 3.088
##                     3rd Qu.: 6.372     3rd Qu.: 4.110
##                     Max.    :20.920     Max.    :15.940
##
##  Financial_leverage
##  Min.   :0.200
##  1st Qu.:0.680
##  Median :1.110
##  Mean   :1.121
##  3rd Qu.:1.587
##  Max.   :2.000
##  NA's   :158
```

Single Factor Analysis - Univariate

```
##   Credit_limit          EDF              GDP_growth
##  Min.   :0.0400   Min.   :0.00990   Min.   :0.310
##  1st Qu.:0.3500   1st Qu.:0.01020   1st Qu.:1.400
##  Median :0.5200   Median :0.01300   Median :1.790
##  Mean   :0.6001   Mean   :0.01405   Mean   :1.954
##  3rd Qu.:0.7800   3rd Qu.:0.01800   3rd Qu.:2.260
##  Max.   :2.2600   Max.   :0.02100   Max.   :5.150
##
```

Exclusions

- Financial_leverage has 158 N/A's out of 1000 observations (15.8%)

```
Drivers_1 = subset(Drivers, select=-c(Financial_leverage))
print(head(Drivers_1,5), digits = 2, row.names = FALSE)
```

```
##  Customer_ID Country Industry Length_of_business
##            1      UK        A                4.3
##            2      UK        D                8.7
##            3      FR        A                7.1
##            4      FR        A                5.6
##            5      UK        A                2.3
##  Total_assets Credit_limit   EDF GDP_growth Default
##           1.5         0.27 0.013       2.95       1
##           9.8         1.27 0.015       1.64       0
##           1.7         0.59 0.010       1.72       0
##           6.2         0.87 0.013       0.96       1
##          15.9         1.87 0.018       1.45       0
```

Further modifications

- Handling outliers
- Dealing with missing values if needed
- Transformations:
    - exponential
    - logarithmic
    - polynomial

Single Factor Analysis - Bivariate - Country

- We check the relationship between risk drivers and default

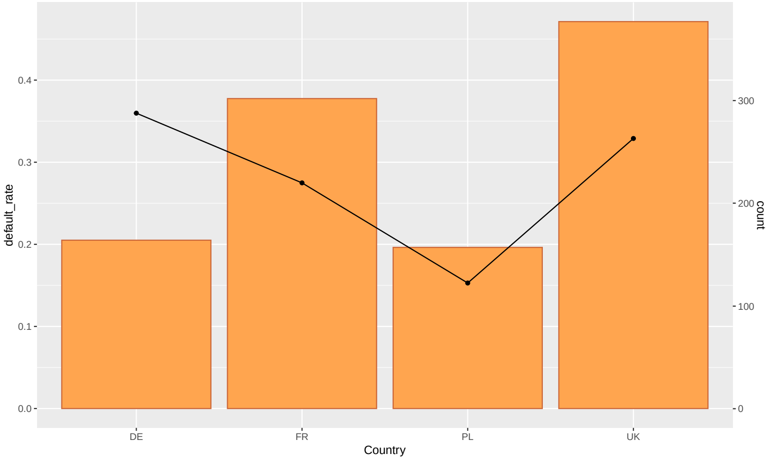
```
country_group <- Data %>% group_by(Country) %>%
  summarise(default_rate = mean(Default),count = n())
print(country_group, digits = 3, row.names = FALSE)
```

```
## # A tibble: 4 x 3
##   Country default_rate count
##   <fct>          <dbl> <int>
## 1 DE             0.360   164
## 2 FR             0.275   302
## 3 PL             0.153   157
## 4 UK             0.329   377
```

Single Factor Analysis - Bivariate - Country

```
ggplot(data=country_group, aes(x=Country, y=default_rate,
                               group=1)) +
    geom_bar(aes(x=Country, y=count/800),stat="identity",
         fill="tan1", colour="sienna3")+
    geom_line() +
    geom_point()+
    scale_y_continuous(name = waiver(),
                  sec.axis = sec_axis(~ . * 800,
                                      name = "count"))
```

Single Factor Analysis - Bivariate - Country



Single Factor Analysis - Bivariate - Country

- Switch from categorical variable Country to boolean Country_PL
- Is this in line with common sense and expectations?
- What is the expected impact of the variable on the final model?

Single Factor Analysis - Bivariate - Country

```
Drivers_2 <- Drivers_1
Drivers_2$Country_PL <- (Drivers_1$Country == "PL")*1
```

Single Factor Analysis - Bivariate - Country

```
##  Country Industry Length_of_business Total_assets
##       UK        A                4.3          1.5
##       UK        D                8.7          9.8
##       FR        A                7.1          1.7
##       FR        A                5.6          6.2
##       UK        A                2.3         15.9
##       UK        B                4.1          7.3
##       PL        A                3.2          4.2
##  Credit_limit   EDF GDP_growth Default Country_PL
##          0.27 0.013       2.95       1          0
##          1.27 0.015       1.64       0          0
##          0.59 0.010       1.72       0          0
##          0.87 0.013       0.96       1          0
##          1.87 0.018       1.45       0          0
##          0.73 0.018       1.45       1          0
##          0.66 0.013       3.32       0          1
```

Single Factor Analysis - Bivariate - Country

We remove the variable Country now

```
Drivers_2 <- subset(Drivers_2, select=-c(Country))
print(head(Drivers_2,7), digits = 2, row.names = FALSE)
```

Single Factor Analysis - Bivariate - Country

We remove the variable Country now

```
##   Industry Length_of_business Total_assets Credit_limit
## 1        A                4.3          1.5         0.27
## 2        D                8.7          9.8         1.27
## 3        A                7.1          1.7         0.59
## 4        A                5.6          6.2         0.87
## 5        A                2.3         15.9         1.87
## 6        B                4.1          7.3         0.73
## 7        A                3.2          4.2         0.66
##     EDF GDP_growth Default Country_PL
## 1 0.013       2.95       1          0
## 2 0.015       1.64       0          0
## 3 0.010       1.72       0          0
## 4 0.013       0.96       1          0
## 5 0.018       1.45       0          0
## 6 0.018       1.45       1          0
## 7 0.013       3.32       0          1
```

Single Factor Analysis - Bivariate - Industry

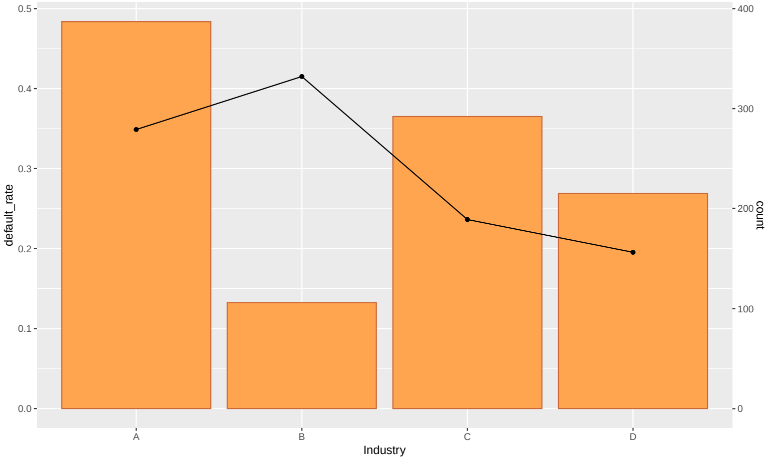
```
industry_group <- Data %>% group_by(Industry) %>%
  summarise(default_rate = mean(Default),count = n())
print(industry_group, digits = 3, row.names = FALSE)
```

```
## # A tibble: 4 x 3
##   Industry default_rate count
##   <fct>           <dbl> <int>
## 1 A               0.349   387
## 2 B               0.415   106
## 3 C               0.236   292
## 4 D               0.195   215
```

Single Factor Analysis - Bivariate - Industry

```r
ggplot(data=industry_group, aes(x=Industry, y=default_rate,
                                group=1)) +
    geom_bar(aes(x=Industry, y=count/800),stat="identity",
             fill="tan1", colour="sienna3")+
    geom_line() +
    geom_point()+
    scale_y_continuous(name = waiver(),
                       sec.axis = sec_axis(~ . * 800,
                                           name = "count"))
```

Single Factor Analysis - Bivariate - Industry



Single Factor Analysis - Bivariate - Industry

- Switch from categorical variables Industry $\in \{A, B\}$ to boolean Industry_AB
- Is this in line with common sense and expectations?
- What is the expected impact of the variable on the final model?

Single Factor Analysis - Bivariate - Industry

```
Drivers_3 <- Drivers_2
Drivers_3$Industry_AB <- (Drivers_2$Industry %in% c("A","B"))*1
print(head(subset(Drivers_3, select=-c(Customer_ID)),7),
      digits = 2, row.names = FALSE)
```

Single Factor Analysis - Bivariate - Industry

```
##   Industry Length_of_business Total_assets Credit_limit
##          A                4.3          1.5         0.27
##          D                8.7          9.8         1.27
##          A                7.1          1.7         0.59
##          A                5.6          6.2         0.87
##          A                2.3         15.9         1.87
##          B                4.1          7.3         0.73
##          A                3.2          4.2         0.66
##     EDF GDP_growth Default Country_PL Industry_AB
##   0.013       2.95       1          0           1
##   0.015       1.64       0          0           0
##   0.010       1.72       0          0           1
##   0.013       0.96       1          0           1
##   0.018       1.45       0          0           1
##   0.018       1.45       1          0           1
##   0.013       3.32       0          1           1
```

Single Factor Analysis - Bivariate - Industry

```
##  Length_of_business Total_assets Credit_limit   EDF
##                 4.3          1.5         0.27 0.013
##                 8.7          9.8         1.27 0.015
##                 7.1          1.7         0.59 0.010
##                 5.6          6.2         0.87 0.013
##                 2.3         15.9         1.87 0.018
##                 4.1          7.3         0.73 0.018
##                 3.2          4.2         0.66 0.013
##  GDP_growth Default Country_PL Industry_AB
##        2.95       1          0           1
##        1.64       0          0           0
##        1.72       0          0           1
##        0.96       1          0           1
##        1.45       0          0           1
##        1.45       1          0           1
##        3.32       0          1           1
```

Single Factor Analysis - Bivariate - Length_of_business

Let's bucket the data by year

```
## # A tibble: 19 x 3
##    Length_of_business_Floor default_rate count
##                       <dbl>        <dbl> <int>
##  1                        0       0.469     32
##  2                        1       0.459    111
##  3                        2       0.377    167
##  4                        3       0.384    151
##  5                        4       0.271    140
##  6                        5       0.266    109
##  7                        6       0.2       85
##  8                        7       0.138     58
##  9                        8       0.1       50
## 10                        9       0.121     33
## 11                       10       0.0714    14
## 12                       11       0         13
## 13                       12       0         12
## 14                       13       0.167      6
```

Single Factor Analysis - Bivariate - Length_of_business

Let's cut the dataset in 11 and put everything longer than that into one group

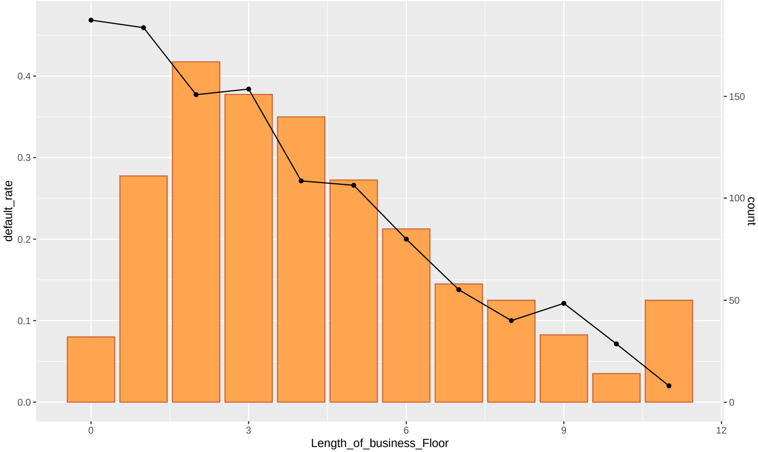
```
## # A tibble: 12 x 3
##    Length_of_business_Floor default_rate count
##                       <dbl>        <dbl> <int>
##  1                        0       0.469     32
##  2                        1       0.459    111
##  3                        2       0.377    167
##  4                        3       0.384    151
##  5                        4       0.271    140
##  6                        5       0.266    109
##  7                        6       0.2       85
##  8                        7       0.138     58
##  9                        8       0.1       50
## 10                        9       0.121     33
## 11                       10       0.0714    14
## 12                       11       0.02      50
```

Single Factor Analysis - Bivariate - Length_of_business

```
ggplot(data=length_group, aes(x=Length_of_business_Floor,
                              y=default_rate, group=1)) +
    geom_bar(aes(x=Length_of_business_Floor, y=count/400),
             stat="identity",
             fill="tan1", colour="sienna3")+
    geom_line() +
    geom_point()+
    scale_y_continuous(name = waiver(),
                       sec.axis = sec_axis(~ . * 400,
                                           name = "count"))
```

Single Factor Analysis - Bivariate - Length_of_business

- Is this relation in line with logic?



Single Factor Analysis - Bivariate - Total_assets

```
Data$Total_assets_Floor <- floor(Data$Total_assets)
assets_group <- Data %>% group_by(
  Total_assets_Floor) %>% summarise(
    default_rate = mean(Default),count = n())
print(assets_group, digits = 3, row.names = FALSE)
```

Single Factor Analysis - Bivariate - Total_assets

```
## # A tibble: 16 x 3
##    Total_assets_Floor default_rate count
##                 <dbl>        <dbl> <int>
##  1                  0        0.320   181
##  2                  1        0.322   236
##  3                  2        0.304   184
##  4                  3        0.321   134
##  5                  4        0.207    87
##  6                  5        0.255    55
##  7                  6        0.268    41
##  8                  7        0.167    30
##  9                  8        0.15     20
## 10                  9        0.167     6
## 11                 10        0.167     6
## 12                 11        0.222     9
## 13                 12        0         4
## 14                 13        0.5       4
## 15                 14        0         1
```

Single Factor Analysis - Bivariate - Total_assets

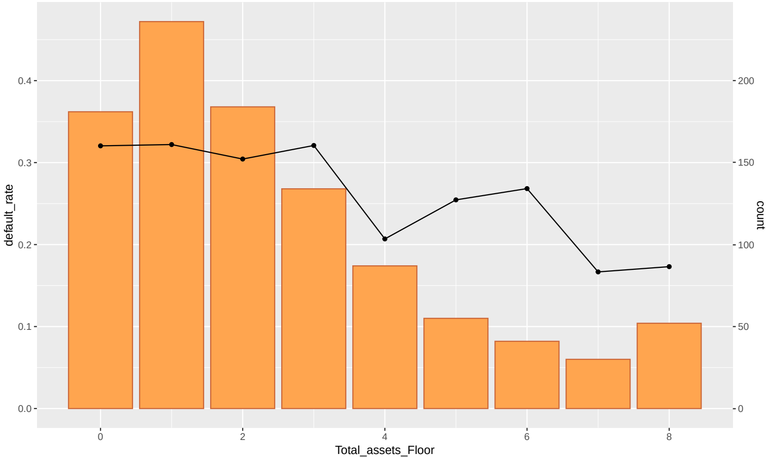
Let's cut the dataset in 8 and put everything longer than that into one group

```
## # A tibble: 9 x 3
##   Total_assets_Floor default_rate count
##                <dbl>        <dbl> <int>
## 1                  0        0.320   181
## 2                  1        0.322   236
## 3                  2        0.304   184
## 4                  3        0.321   134
## 5                  4        0.207    87
## 6                  5        0.255    55
## 7                  6        0.268    41
## 8                  7        0.167    30
## 9                  8        0.173    52
```

Single Factor Analysis - Bivariate - Total_assets

```
ggplot(data=assets_group, aes(x=Total_assets_Floor, y=default_rate,
                              group=1)) +
    geom_bar(aes(x=Total_assets_Floor, y=count/500),
             stat="identity",
             fill="tan1", colour="sienna3")+
    geom_line() +
    geom_point()+
    scale_y_continuous(name = waiver(),
                       sec.axis = sec_axis(~ . * 500,
                                           name = "count"))
```

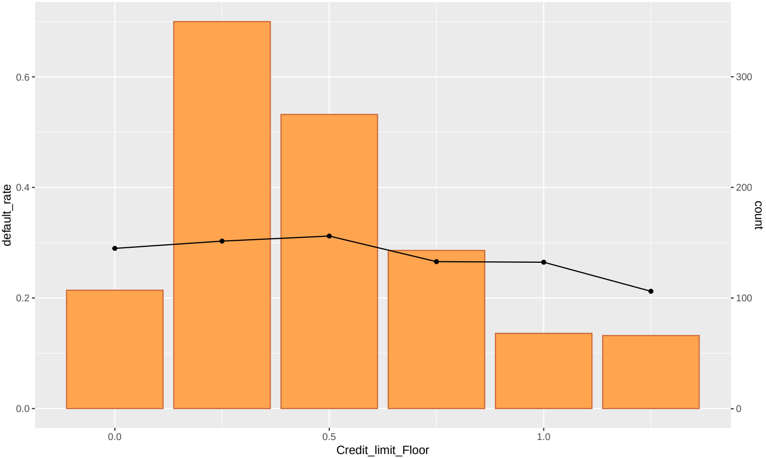
Single Factor Analysis - Bivariate - Total_assets



Single Factor Analysis - Bivariate - Credit_limit

```
## # A tibble: 6 x 3
##   Credit_limit_Floor default_rate count
##                <dbl>        <dbl> <int>
## 1               0           0.290   107
## 2               0.25        0.303   350
## 3               0.5         0.312   266
## 4               0.75        0.266   143
## 5               1           0.265    68
## 6               1.25        0.212    66
```

Single Factor Analysis - Bivariate - Credit_limit



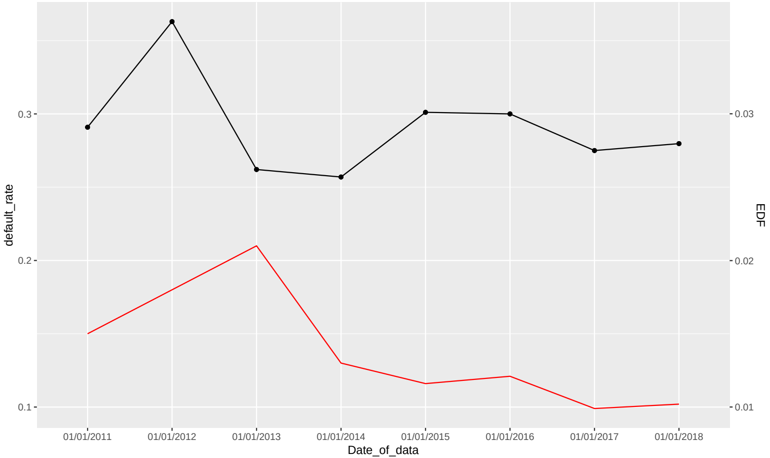
Single Factor Analysis - Bivariate - Expected Default Frequency

EDF is a variable common to all debtors dependent on year

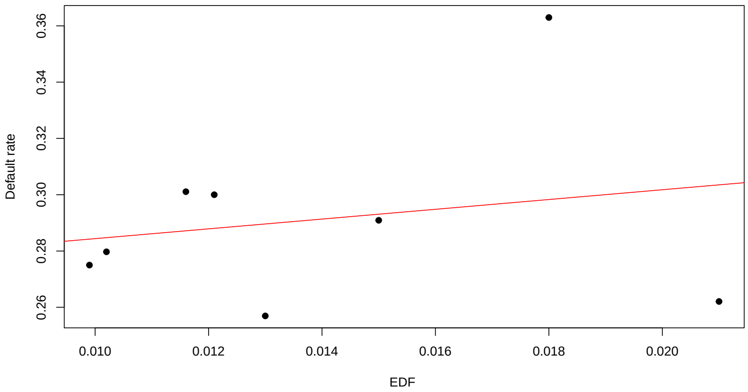
```
EDF_group <- Data %>% group_by(Date_of_data) %>%
  summarise(default_rate = mean(Default), EDF = mean(EDF),
            count = n())
print(EDF_group, digits = 3, row.names = FALSE)
```

```
## # A tibble: 8 x 4
##   Date_of_data default_rate    EDF count
##   <fct>               <dbl>  <dbl> <int>
## 1 01/01/2011          0.291 0.015    110
## 2 01/01/2012          0.363 0.018    135
## 3 01/01/2013          0.262 0.021    145
## 4 01/01/2014          0.257 0.013    144
## 5 01/01/2015          0.301 0.0116    93
## 6 01/01/2016          0.3   0.0121   110
## 7 01/01/2017          0.275 0.0099   120
## 8 01/01/2018          0.280 0.0102   143
```

Single Factor Analysis - Bivariate - Expected Default Frequency



Single Factor Analysis - Bivariate - Expected Default Frequency




```
## integer(0)
```

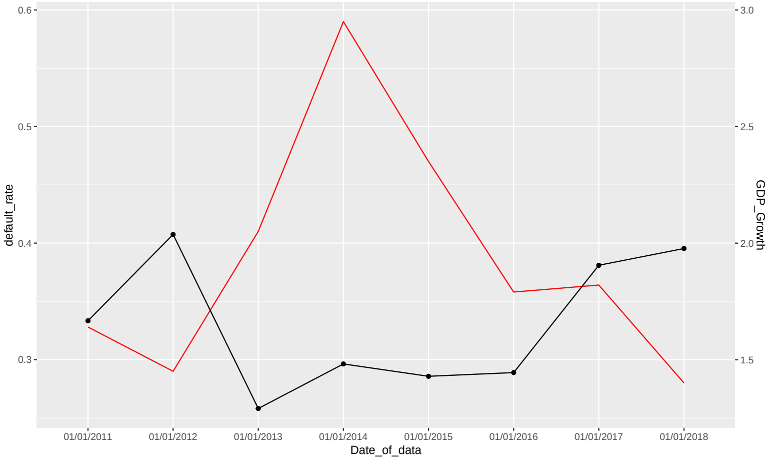
Single Factor Analysis - Bivariate - GDP_growth

GDP_growth is a variable common to all debtors dependent on year and country

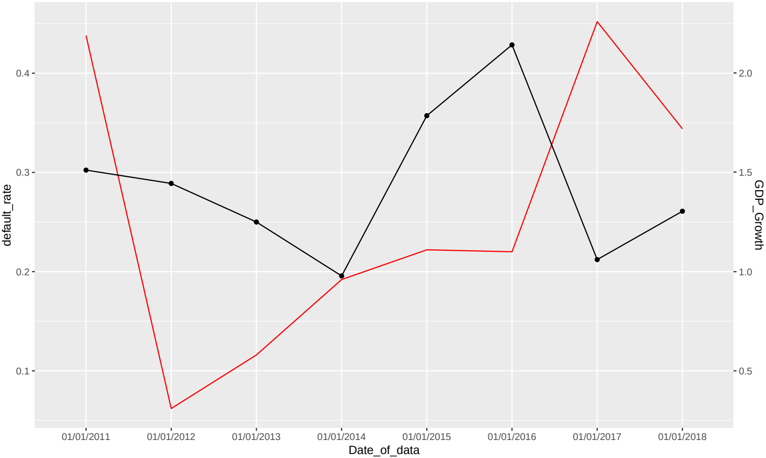
```
GDP_group <- Data %>% group_by(Date_of_data,Country) %>%
  summarise(default_rate = mean(Default),GDP_growth = mean(GDP_growth))
print(GDP_group, digits = 3, row.names = FALSE)
```

```
## # A tibble: 32 x 4
## # Groups:   Date_of_data [8]
##    Date_of_data Country default_rate GDP_growth
##    <fct>        <fct>          <dbl>      <dbl>
##  1 01/01/2011   DE            0.286        3.66
##  2 01/01/2011   FR            0.302        2.19
##  3 01/01/2011   PL            0.0909       5.02
##  4 01/01/2011   UK            0.333        1.64
##  5 01/01/2012   DE            0.474        0.49
##  6 01/01/2012   FR            0.289        0.31
##  7 01/01/2012   PL            0.294        1.61
##  8 01/01/2012   UK            0.407        1.45
```

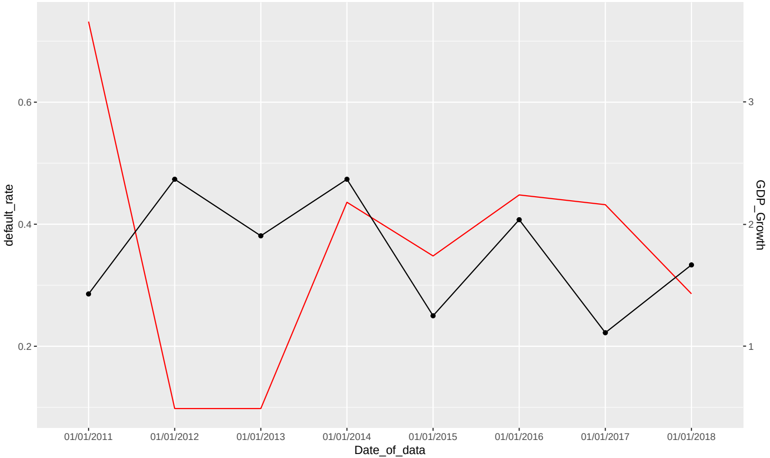
Single Factor Analysis - Bivariate - GDP_growth - UK



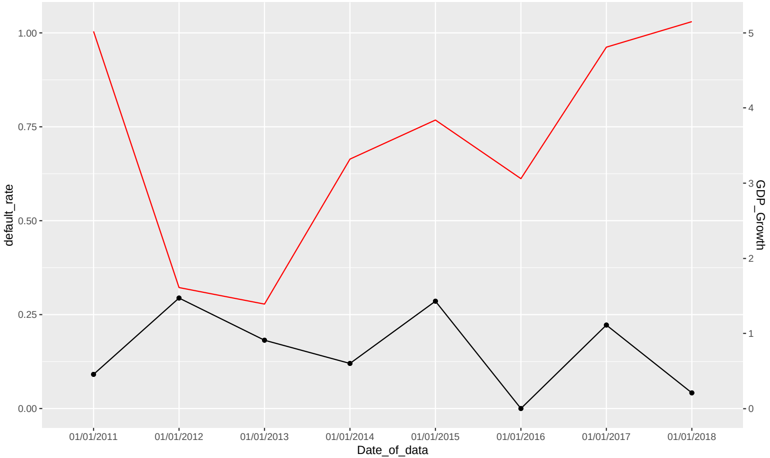
Single Factor Analysis - Bivariate - GDP_growth - FR



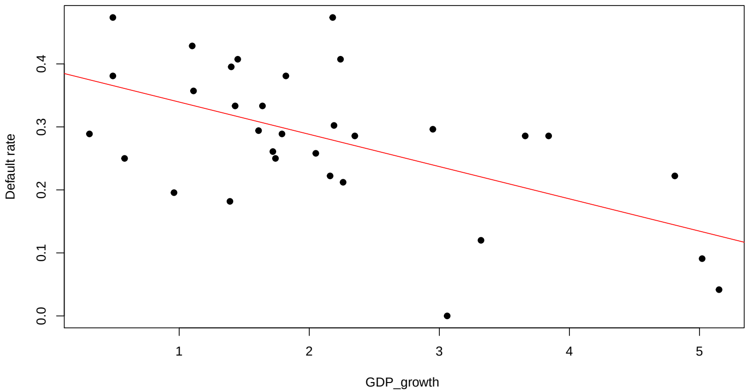
Single Factor Analysis - Bivariate - GDP_growth - DE



Single Factor Analysis - Bivariate - GDP_growth - PL



Single Factor Analysis - Bivariate - GDP_growth - All countries




```
## integer(0)
```

Final dataset

```r
Drivers_final <- Drivers_3[, c("Country_PL","Industry_AB",
                  "Length_of_business","Total_assets","Credit_limit",
                  "EDF","GDP_growth","Default")]
print(head(Drivers_final,5), digits = 2, row.names = FALSE)
```

```
##  Country_PL Industry_AB Length_of_business Total_assets
##           0           1                4.3          1.5
##           0           0                8.7          9.8
##           0           1                7.1          1.7
##           0           1                5.6          6.2
##           0           1                2.3         15.9
##  Credit_limit   EDF GDP_growth Default
##          0.27 0.013       2.95       1
##          1.27 0.015       1.64       0
##          0.59 0.010       1.72       0
##          0.87 0.013       0.96       1
##          1.87 0.018       1.45       0
```

3. Data split

Development sample

- Data that we use to estimate model parameters
- Usually between 75% and 90% of the whole sample

```
set.seed(101)
sample = sample.split(Drivers_final$Default, SplitRatio = .80)
development_sample = subset(Drivers_final, sample == TRUE)
```

Hold-out sample

- Data that we use to evaluate the performance of the model
- Usually between 10% and 25% of the whole sample

```
hold_out_sample  = subset(Drivers_final, sample == FALSE)
```

4. Model functional form

Model functional form

Possible methods for PD modelling:

- Probit model
- Logistic regression
- Scoring models
- Machine learning
- Neural networks

Logistic Regression

$$\ln\left\{\frac{P[Y=1|X]}{P[Y=0|X]}\right\} = \beta_0 + X\beta$$

with $X = (X_1, X_2, \ldots, X_N)$ the set of prognostic factors. Assuming a linear model for f_n, the probability that $Y = 1$ is modelled as:

$$y = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \ldots)}}$$

In R, this regression can be fitted with the function `glm()`.

5. Multiple Factor Analysis

Number of possible models

- We have 7 input variables (risk drivers) and 1 modelled variable
- The number of possible models: $2^7 - 1 = 127$.

```
variables = colnames(Drivers_final)
variables
```

```
## [1] "Country_PL"         "Industry_AB"
## [3] "Length_of_business" "Total_assets"
## [5] "Credit_limit"       "EDF"
## [7] "GDP_growth"         "Default"
```

Exemplary model

```
m0 <- glm(data = development_sample,
          formula = Default ~ Country_PL + Industry_AB +
                                        Total_assets + Credit_limit + EDF,
         family = binomial)
summary(m0)[12]
```

Exemplary model

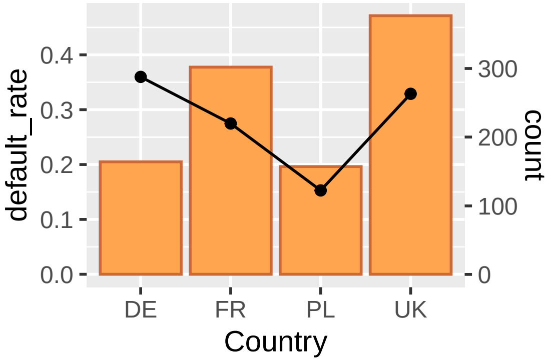
```
## $coefficients
##                  Estimate  Std. Error    z value
## (Intercept)    -0.6016979  0.35144990 -1.7120445
## Country_PL     -0.9552921  0.26030418 -3.6699068
## Industry_AB     0.7913721  0.16323447  4.8480697
## Total_assets   -0.1156390  0.04670397 -2.4759986
## Credit_limit    0.2254393  0.33022570  0.6826825
## EDF           -26.7674014 21.20266772 -1.2624544
##                  Pr(>|z|)
## (Intercept)  8.688847e-02
## Country_PL   2.426389e-04
## Industry_AB  1.246686e-06
## Total_assets 1.328641e-02
## Credit_limit 4.948075e-01
## EDF          2.067853e-01
```

Exemplary model

```
##          Driver Sign Estimate
## 1    Country_PL    -    -0.96
## 2   Industry_AB    +     0.79
## 3  Total_assets    -    -0.12
## 4  Credit_limit   -?     0.23
## 5           EDF   +?   -26.77
```

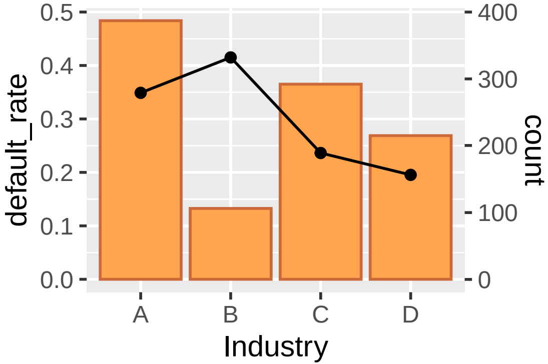
Acceptance criteria - No counterintuitive signs

```
##        Driver Sign Estimate
## 1 Country_PL    -    -0.96
```



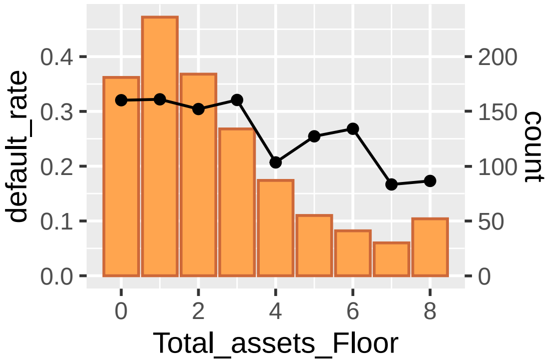
Acceptance criteria - No counterintuitive signs

```
##         Driver Sign Estimate
## 2 Industry_AB    +     0.79
```



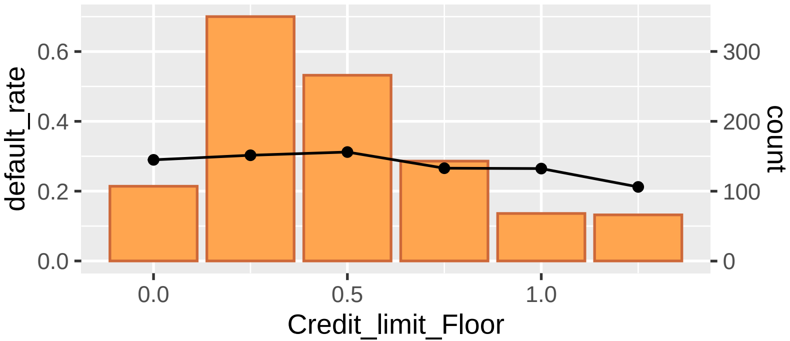
Acceptance criteria - No counterintuitive signs

```
##          Driver Sign Estimate
## 3 Total_assets    -    -0.12
```



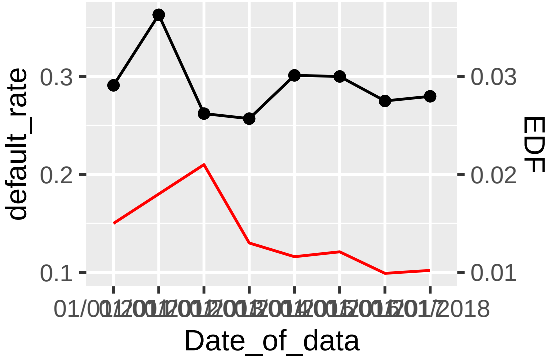
Acceptance criteria - No counterintuitive signs

```
##          Driver Sign Estimate
## 4 Credit_limit   -?     0.23
```



Acceptance criteria - No counterintuitive signs

```
##   Driver Sign Estimate
## 5    EDF   +?      -27
```



Acceptance criteria - p-value

```
summary <- data.frame(coef(summary(m0))[,c(1,4)])
summary$p_val_less_5PRC <- summary[,2] <= 0.05
summary
```

```
##                  Estimate     Pr...z.. p_val_less_5PRC
## (Intercept)    -0.6016979 8.688847e-02           FALSE
## Country_PL     -0.9552921 2.426389e-04            TRUE
## Industry_AB     0.7913721 1.246686e-06            TRUE
## Total_assets   -0.1156390 1.328641e-02            TRUE
## Credit_limit    0.2254393 4.948075e-01           FALSE
## EDF           -26.7674014 2.067853e-01           FALSE
```

Acceptance criteria - correlation

No two variables can be correlated more than 0.50 in absolute terms.

```
corr_data <- subset(development_sample,
                    select = c("Country_PL",
                               "Industry_AB",
                               "Total_assets",
                               "Credit_limit",
                               "EDF"))
correlation_results <- cor(corr_data)
```

Acceptance criteria - correlation

```
##              Country_PL Industry_AB Total_assets
## Country_PL       1.0000      -0.027       0.0053
## Industry_AB     -0.0271       1.000       0.0123
## Total_assets     0.0053       0.012       1.0000
## Credit_limit     0.0193      -0.025       0.7145
## EDF             -0.0395       0.068      -0.0405
##              Credit_limit    EDF
## Country_PL          0.019 -0.040
## Industry_AB        -0.025  0.068
## Total_assets        0.715 -0.040
## Credit_limit        1.000 -0.028
## EDF                -0.028  1.000
```

Acceptance criteria - Summary

- Expected sign
  - Credit_limit and EDF do not meet the criteria
- Significance (p-value)
  - Credit_limit and EDF do not meet the criteria
- Correlation
  - Total_assets and Credit_limit cannot appear in the same model

Result -> model rejected

Model search

An estimation is done for each possible model and only the models that fulfil all the criteria are considered further. In practice:

- models including correlated pairs of variables are not estimated
- regulatory requirements state that some kinds of variables need to be included, eg:
  - customer size or proxy
  - macroeconomic

6. Model selection

Model selection - performance criteria

For all the models that passed the acceptance criteria we calculate some performance metrics eg.:

- Gini coefficient - the higher the better
- Akaike information criterion (AIC) - the lower the better

AIC - Akaike Information Criteria

$$AIC = 2k - 2ln(\hat{L}),$$

where:

k - number of parameters (penalize more parameters)

$\hat{L}$ - likelihood function (promote higher likelihood)

Model selection - performance criteria

Let's compare three models:

- m1: Default ~ Industry_AB + Length_of_business + Total_assets
- m2: Default ~ Country_PL + Length_of_business + Total_assets
- m3: Default ~ Country_PL + Industry_AB + Length_of_business + Total_assets

Model selection - estimation of parameters

```
m1 <- glm(data = development_sample,
          formula = Default ~ Industry_AB + Length_of_business +
                              Total_assets,
          family = binomial)
m2 <- glm(data = development_sample,
          formula = Default ~ Country_PL + Length_of_business +
                              Total_assets,
          family = binomial)
m3 <- glm(data = development_sample,
          formula = Default ~ Country_PL + Industry_AB +
                              Length_of_business + Total_assets,
          family = binomial)
```

Model selection - Gini

We predict the probabilities for each model

```
development_sample$prediction_m1 =
              fitted.values(m1)
development_sample$prediction_m2 =
              fitted.values(m2)
development_sample$prediction_m3 =
              fitted.values(m3)
```

Model selection - Gini

```
print(head(subset(development_sample, select =
                    c(Default,prediction_m1,
                      prediction_m2,prediction_m3)),
           10),digits = 2)
```

```
##    Default prediction_m1 prediction_m2 prediction_m3
## 1        1         0.406         0.357         0.440
## 2        0         0.043         0.071         0.048
## 4        1         0.237         0.202         0.259
## 5        0         0.235         0.207         0.255
## 6        1         0.297         0.259         0.324
## 8        0         0.390         0.343         0.423
## 9        0         0.311         0.269         0.339
## 10       0         0.331         0.463         0.365
## 12       0         0.195         0.293         0.218
## 13       0         0.568         0.283         0.357
```

Model selection - Gini

```
model_summary <- data.frame(
              "Model"= c("m1","m2","m3"),
              "Gini_development" =
                c(Gini(development_sample$prediction_m1,
                        development_sample$Default),
                  Gini(development_sample$prediction_m2,
                        development_sample$Default),
                  Gini(development_sample$prediction_m3,
                        development_sample$Default)))
```

Model selection - Gini

```r
print(model_summary, digits = 3)
```

```
##   Model Gini_development
## 1    m1            0.215
## 2    m2            0.195
## 3    m3            0.228
```

Model selection - AIC

```
model_summary$AIC <- c(AIC(m1),AIC(m2),AIC(m3))
print(model_summary, digits = 3)
```

```
##   Model Gini_development AIC
## 1    m1            0.215 869
## 2    m2            0.195 872
## 3    m3            0.228 854
```

Model selection - Champion and Challenger

After the analysis of all possible models for all functional forms considered we choose:

- Champion model - best model (our m3)
- Challenger model - second best (our m1)

7. Model validation

Model validation

We need to check how our champion and challanger models perform on the hold-out sample

```
hold_out_sample$prediction_m3 <-predict(m3,
                      newdata = hold_out_sample, type = 'response')
hold_out_sample$prediction_m1 <-predict(m1,
                      newdata = hold_out_sample, type = 'response')
```

Validation - Gini

```
validation_summary <- data.frame(
              "Model"= c("m3","m1"),
              "Gini_hold_out" =
                c(Gini(hold_out_sample$prediction_m3,
                      hold_out_sample$Default),
                  Gini(hold_out_sample$prediction_m1,
                      hold_out_sample$Default)))
```

Validation - Gini

```r
print(validation_summary, digits = 3)
```

```
##   Model Gini_hold_out
## 1    m3         0.235
## 2    m1         0.232
```

Summarize

```
summary_final <- merge(x = model_summary,
                       y = validation_summary,
                       by = "Model",
                       all.y = TRUE) %>% subset(select=-c(AIC))
summary_final$Dev_minus_hold_out <-
  summary_final$Gini_development - summary_final$Gini_hold_out
```

Summarize

```
print(summary_final, digits = 3)
```

```
##   Model Gini_development Gini_hold_out Dev_minus_hold_out
## 1    m1            0.215         0.232            -0.0173
## 2    m3            0.228         0.235            -0.0063
```

Conclusions

- Both models seem to perform better on the hold-out sample than on the development sample
- The classification remains the same:
  - Champion: m3 - Default ~ Country_PL + Industry_AB + Length_of_business + Total_assets

```
print(coefficients(m3), digits = 3)
```

```
##         (Intercept)          Country_PL         Industry_AB
##              0.3462             -1.0112              0.7339
## Length_of_business        Total_assets
##             -0.2759             -0.0955
```

Conclusions

- Challenger: m1 - Default ~ Industry_AB + Length_of_business + Total_assets

```r
print(coefficients(m1), digits = 3)
```

```
##         (Intercept)         Industry_AB Length_of_business
##              0.1781              0.7392            -0.2712
##        Total_assets
##             -0.0927
```

End

THANK YOU!!!