Turkey Labor Force Statistics

Veli Kisa

27/04/2020

Introduction

This data is downloaded from Turkish Statistical Institute website. (Link : http://www.turkstat.gov.tr/PreTabloArama.do?metod=search&araType=vt Labour Force Statistics (2014 and after)(M)) Data shows us the labour force statistics of Turkey. The number of labour force (thousand) is based on year (2014-2019) and there are sociological (gender, age_group,education) and regional variables. We can define these variables as;

gender= Erkek,Kadın (Male, Female)
age_group= illustrates age ranges
education=
Okuma Yazma Bilmeyen = Unalphabet
Lise Altı Eğitim = Lower High-School
Lise ve Dengi Meslek Okulu = High School And Equivalent Technical High School
Yüksek Öğretim = Higher Education

regional =
Akdeniz = Mediterrenean
Batı Anadolu = Western Anatolia
Batı Karadeniz = Western Black Sea
Batı Marmara = Western Marmara
Doğu Karadeniz= Eastern Karadeniz
Doğu Marmara = Eastern Marmara
Ege = Aegean
Güneydoğu Anadolu = Southeastern Anatolia
İstanbul = İstanbul
Kuzeydoğu Anadolu= Northeastern Anatolia
Orta Anadolu= Middle Anatolia
Ortadoğu Anadolu = Middle Eastern Anatolia

The numbers represent thousand result. We can specify the aim of this project as

1. Loading and tidy data (Gathering and separating data)
2. Display some relations between exact variables.
*3. Applying statistical tests to determine whether there is relation between variables.

Loading and Tidy Data

```r
library(readxl)
veri <- read_excel("veri.xlsx")
```

Data need to be arranged

```r
library(tidyr)
colnames(veri)[3:14]<-gsub("-.*", "", colnames(veri)[3:14])
```

```r
#remove all strings after "-" character
veri<-fill(veri,`cinsiyet_egitim.durumu`)
#fill null rows with the lastest full row
head(veri)
```

```
## # A tibble: 6 x 14
##   cinsiyet_egitim~ yil   Akdeniz `Batı Anadolu` `Batı Karadeniz`
##   <chr>            <chr>   <dbl> <chr>          <chr>
## 1 1. (15+) ve Erk~ 2014        2 0              0
## 2 1. (15+) ve Erk~ 2015        1 (6)*           (6)*
## 3 1. (15+) ve Erk~ 2016        1 1              0
## 4 1. (15+) ve Erk~ 2017        2 3              (6)*
## 5 1. (15+) ve Erk~ 2018        4 2              0
## 6 1. (15+) ve Erk~ 2019        2 0              1
## # ... with 9 more variables: `Batı Marmara` <chr>, `Doğu Karadeniz` <chr>,
## #   `Doğu Marmara` <chr>, Ege <chr>, `Güneydoğu Anadolu` <chr>,
## #   İstanbul <chr>, `Kuzeydoğu Anadolu` <chr>, `Orta Anadolu` <chr>,
## #   `Ortadoğu Anadolu` <chr>
```

Now we should separate the first column as “gender”, “age_group” and “education”

```r
library(stringr)
veri$cinsiyet_egitim.durumu<-str_replace_all(veri$'cinsiyet_egitim.durumu', "\\s", "")
#1 remove spaces
veri$cinsiyet_egitim.durumu<-str_split(veri$cinsiyet_egitim.durumu, "ve")
#2 split cells as specific string ("ve")
veri<-separate(veri,cinsiyet_egitim.durumu,c("gen_age","gender","age_group","education"),sep=',')
#3 separate cells to columns with column names
veri<-veri[,-1]
#4 remove unnecessary column
veri$age_group<-gsub(".*\\((.*)\\).*", "\\1", veri$age_group)
veri$education<-gsub(".*\\((.*)\\).*", "\\1", veri$education)
#5 take exact string into paranthesis
veri$gender<-str_replace_all(veri$gender, "[[:punct:]]", " ")
veri$education<-str_replace_all(veri$education, "[[:punct:]]", " ")
#6 remove any characteristic from cells
veri$gender<-str_replace_all(veri$gender, "\\s", "")
#7 take off the spaces
head(veri)
```

```
## # A tibble: 6 x 16
##   gender age_group education yil   Akdeniz `Batı Anadolu` `Batı Karadeniz`
##   <chr>  <chr>     <chr>     <chr>   <dbl> <chr>          <chr>
## 1 Erkek  15-19     "OkumaYa~ 2014        2 0              0
## 2 Erkek  15-19     "OkumaYa~ 2015        1 (6)*           (6)*
## 3 Erkek  15-19     "OkumaYa~ 2016        1 1              0
## 4 Erkek  15-19     "OkumaYa~ 2017        2 3              (6)*
## 5 Erkek  15-19     "OkumaYa~ 2018        4 2              0
## 6 Erkek  15-19     "OkumaYa~ 2019        2 0              1
## # ... with 9 more variables: `Batı Marmara` <chr>, `Doğu Karadeniz` <chr>,
## #   `Doğu Marmara` <chr>, Ege <chr>, `Güneydoğu Anadolu` <chr>,
## #   İstanbul <chr>, `Kuzeydoğu Anadolu` <chr>, `Orta Anadolu` <chr>,
## #   `Ortadoğu Anadolu` <chr>
```

To gather regions (from ‘Akdeniz’ to ‘Orta Doğu Anadolu’) into one column (format should be as ‘region’,‘gender’,‘education’,‘age_group’,‘year’,‘Observation(N)’) ,

```
veri<-gather(veri, region, N, colnames(veri)[5:16])
veri<-veri[,c(5,1,3,2,4,6)]
colnames(veri)[5]<-"year"
head(veri)
```

```
## # A tibble: 6 x 6
##   region  gender education              age_group year  N
##   <chr>   <chr>  <chr>                  <chr>     <chr> <chr>
## 1 Akdeniz Erkek  "OkumaYazmaBilmeyen  " 15-19     2014  2
## 2 Akdeniz Erkek  "OkumaYazmaBilmeyen  " 15-19     2015  1
## 3 Akdeniz Erkek  "OkumaYazmaBilmeyen  " 15-19     2016  1
## 4 Akdeniz Erkek  "OkumaYazmaBilmeyen  " 15-19     2017  2
## 5 Akdeniz Erkek  "OkumaYazmaBilmeyen  " 15-19     2018  4
## 6 Akdeniz Erkek  "OkumaYazmaBilmeyen  " 15-19     2019  2
```

```
str(veri)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    5760 obs. of  6 variables:
##  $ region   : chr  "Akdeniz" "Akdeniz" "Akdeniz" "Akdeniz" ...
##  $ gender   : chr  "Erkek" "Erkek" "Erkek" "Erkek" ...
##  $ education: chr  "OkumaYazmaBilmeyen  " "OkumaYazmaBilmeyen  " "OkumaYazmaBilmeyen  " "OkumaYazmaB
##  $ age_group: chr  "15-19" "15-19" "15-19" "15-19" ...
##  $ year     : chr  "2014" "2015" "2016" "2017" ...
##  $ N        : chr  "2" "1" "1" "2" ...
```

Before analysing, column types should be assigned correctly.

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date
```

```
veri$region<-as.factor(veri$region)
veri$gender<-as.factor(veri$gender)
veri$education<-as.factor(veri$education)
veri$age_group<-as.factor(veri$age_group)
veri$year<-lubridate::year(as.Date(veri$year,format= "%Y"))
veri$N<-as.numeric(veri$N)
str(veri)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    5760 obs. of  6 variables:
##  $ region   : Factor w/ 12 levels "Akdeniz","Batı Anadolu",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ gender   : Factor w/ 2 levels "Erkek","Kadın": 1 1 1 1 1 1 1 1 1 1 ...
##  $ education: Factor w/ 4 levels "LiseAltıEğitimliler  ",..: 3 3 3 3 3 3 1 1 1 1 ...
##  $ age_group: Factor w/ 5 levels "15-19","20-24",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ year     : num  2014 2015 2016 2017 2018 ...
##  $ N        : num  2 1 1 2 4 2 134 130 140 142 ...
```

While the first four (region,gender,education,age_group) columns have labaled as factor, year and N (thousand) has labeled as date and N (the number) has labeled as numeric.

Visualisation of Data

Before plotting the graphs we need to create the frequency tables of variables that we want to plot. Variables "gender", "education" and "age_group" will be tabulated based on years.

```
library(plyr)
```

```
##
## Attaching package: 'plyr'

## The following object is masked from 'package:lubridate':
##
##     here
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
vars <- names(veri)[2:4]
for (i in vars) {
  print(i)
  freq.table <- veri %>% group_by_("year",i)%>%dplyr::summarise(sum. = sum(N,na.rm = T))
  freq.table<-ddply(freq.table, .(year), mutate, per. = round((sum. / sum(sum.,na.rm = T)),3))
  print(freq.table)
}
```

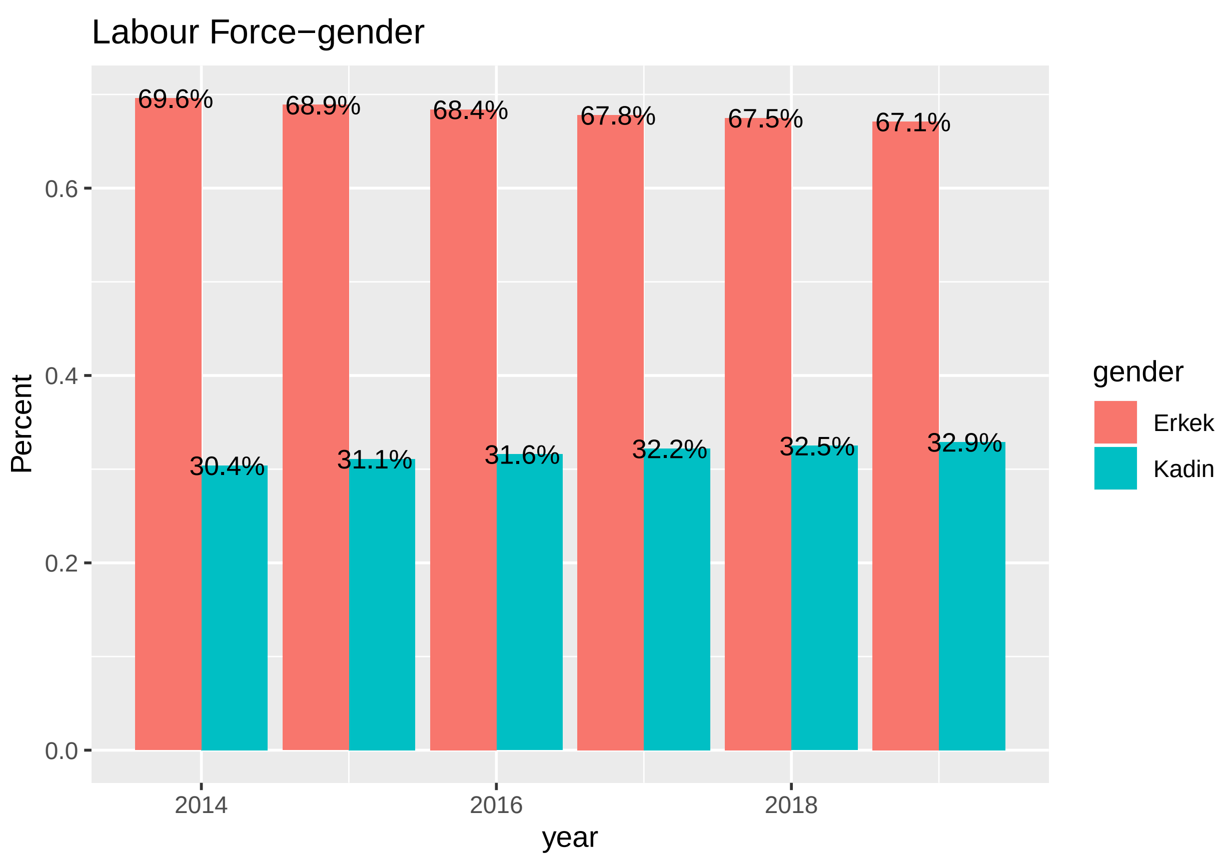
```
## [1] "gender"

## Warning: group_by_() is deprecated.
## Please use group_by() instead
##
## The 'programming' vignette or the tidyeval book can help you
## to program with group_by() : https://tidyeval.tidyverse.org
## This warning is displayed once per session.

##   year gender  sum.  per.
## 1  2014  Erkek 39588 0.696
## 2  2014  Kadın 17267 0.304
## 3  2015  Erkek 40361 0.689
## 4  2015  Kadın 18244 0.311
## 5  2016  Erkek 41227 0.684
## 6  2016  Kadın 19067 0.316
```

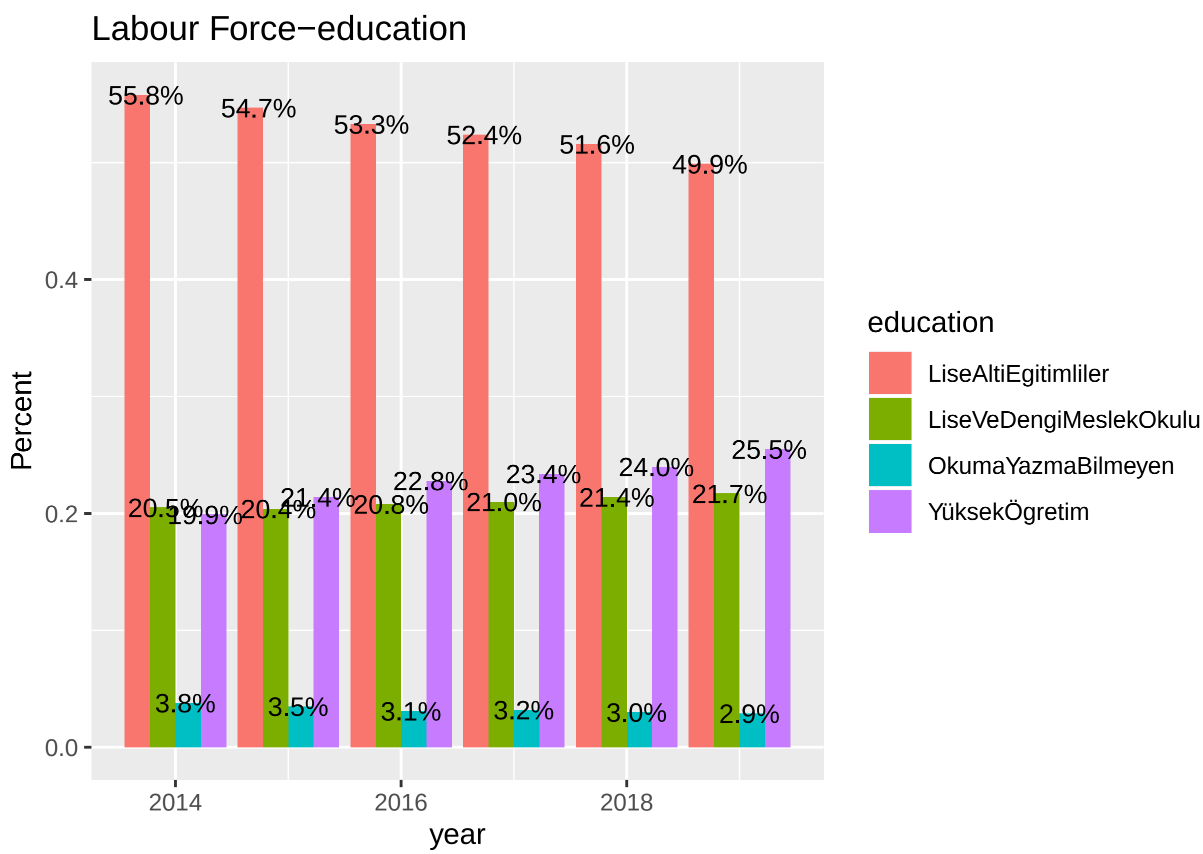
```
## 7  2017  Erkek 42342 0.678
## 8  2017  Kadın 20108 0.322
## 9  2018  Erkek 42939 0.675
## 10 2018  Kadın 20701 0.325
## 11 2019  Erkek 43051 0.671
## 12 2019  Kadın 21142 0.329
## [1] "education"
##    year              education  sum.  per.
## 1  2014    LiseAltıEğitimliler   31728 0.558
## 2  2014 LiseVeDengiMeslekOkulu   11627 0.205
## 3  2014      OkumaYazmaBilmeyen    2169 0.038
## 4  2014           YüksekÖğretim   11331 0.199
## 5  2015    LiseAltıEğitimliler   32061 0.547
## 6  2015 LiseVeDengiMeslekOkulu   11970 0.204
## 7  2015      OkumaYazmaBilmeyen    2049 0.035
## 8  2015           YüksekÖğretim   12525 0.214
## 9  2016    LiseAltıEğitimliler   32155 0.533
## 10 2016 LiseVeDengiMeslekOkulu   12517 0.208
## 11 2016      OkumaYazmaBilmeyen    1898 0.031
## 12 2016           YüksekÖğretim   13724 0.228
## 13 2017    LiseAltıEğitimliler   32737 0.524
## 14 2017 LiseVeDengiMeslekOkulu   13106 0.210
## 15 2017      OkumaYazmaBilmeyen    1972 0.032
## 16 2017           YüksekÖğretim   14635 0.234
## 17 2018    LiseAltıEğitimliler   32869 0.516
## 18 2018 LiseVeDengiMeslekOkulu   13593 0.214
## 19 2018      OkumaYazmaBilmeyen    1906 0.030
## 20 2018           YüksekÖğretim   15272 0.240
## 21 2019    LiseAltıEğitimliler   32035 0.499
## 22 2019 LiseVeDengiMeslekOkulu   13941 0.217
## 23 2019      OkumaYazmaBilmeyen    1832 0.029
## 24 2019           YüksekÖğretim   16385 0.255
## [1] "age_group"
##    year age_group  sum.  per.
## 1  2014     15-19  3470 0.061
## 2  2014     20-24  6096 0.107
## 3  2014     25-34 17126 0.301
## 4  2014     35-54 25208 0.443
## 5  2014       55+  4955 0.087
## 6  2015     15-19  3560 0.061
## 7  2015     20-24  6352 0.108
## 8  2015     25-34 17168 0.293
## 9  2015     35-54 26252 0.448
## 10 2015       55+  5273 0.090
## 11 2016     15-19  3544 0.059
## 12 2016     20-24  6492 0.108
## 13 2016     25-34 17234 0.286
## 14 2016     35-54 27354 0.454
## 15 2016       55+  5670 0.094
## 16 2017     15-19  3572 0.057
## 17 2017     20-24  6704 0.107
## 18 2017     25-34 17372 0.278
## 19 2017     35-54 28696 0.460
## 20 2017       55+  6106 0.098
```

```
## 21 2018      15-19  3596 0.057
## 22 2018      20-24  6754 0.106
## 23 2018      25-34 17318 0.272
## 24 2018      35-54 29466 0.463
## 25 2018        55+  6506 0.102
## 26 2019      15-19  3506 0.055
## 27 2019      20-24  6854 0.107
## 28 2019      25-34 17410 0.271
## 29 2019      35-54 29888 0.466
## 30 2019        55+  6535 0.102
```

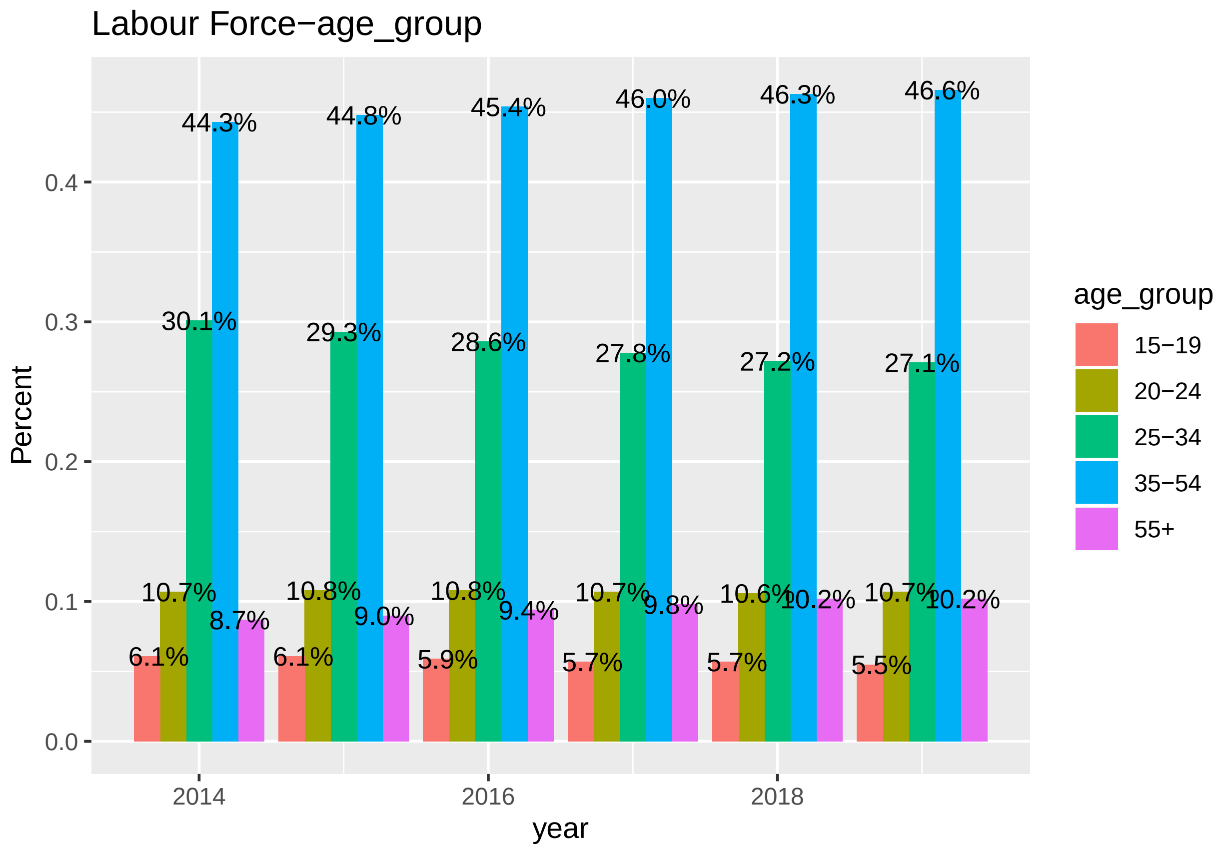
Following chunk indicates the barplot graphs of work force numbers according to gender, education and age_group variables for certain years.

```r
library(ggplot2)
for (i in vars) {

  freq.table <- veri %>% group_by_("year",i)%>%dplyr::summarise(sum. = sum(N,na.rm = T))
  freq.table<-ddply(freq.table, .(year), mutate, per. = round((sum. / sum(sum.,na.rm = T)),3))

  p<-ggplot(freq.table,aes(fill=freq.table[,i],y=freq.table[,"per."],x=freq.table[,"year"]))+
    geom_bar(position="dodge", stat="identity")+
    geom_text(aes(label=scales::percent(freq.table$per.)),position = position_dodge(.7),
              size=3.5)+
    labs(title=paste("Labour Force",i,sep = "-"),y="Percent",x="year",fill=i)
  print(p)
}
```





Labour Force−education
55.8%
54.7%
53.3%
52.4%
51.6%
49.9%
20.5%
19.9%
20.4%
21.4%
20.8%
22.8%
21.0%
23.4%
21.4%
24.0%
21.7%
25.5%
3.8%
3.5%
3.1%
3.2%
3.0%
2.9%
Percent
0.4
0.2
0.0
2014
2016
2018
year
education
LiseAltiEgitimliler
LiseVeDengiMeslekOkulu
OkumaYazmaBilmeyen
YüksekÖgretim



Building Predictive Model

Choosing the best suited technique based on type of predictors and target variable, dimensionality in the data. To select the right regression model belows are key factors on that way; * Data exploration (We have already done on previous sections.) * To compare the goodness of fit for different models, we can analyse with different metrics such as R-square, adjusted R-square, AIC, BIC, significance of parameters and error term. * And last but not least technique is definitely CV (Cross-Validation). This technique is the best way to evaluate models used for prediction. In this technique we need to divide our data set into two group (train and test). A simple mean squared difference between the observed and predicted values give us a measure for the prediction accuracy.

Now, to select the right regression model we will apply both techniques (comparing the GOF and CV).

Comparing the Goodness of Fit Values

To investigate the relationship "gender" , "age_group" , "education" with labor force , we need to apply statistical tests. While these three variables are independent as well as categoric, labor force (N) variable is target (dependent) and numeric. In this part the target variable will be considered as continuous. (We don't choose integer because range is much.)

To select best features and create model "MXM" is a usefull package in R library. Get more information :https://arxiv.org/pdf/1611.03227.pdf

Continuous Target- Mixed Predictors

"test" input describes the conditional independence test to use. Continuous(Target)- Mixed (predictors)= testIndReg (Linear regression) "max_k" implies the maximum conditioning set to use in the conditional independence test. “threshold” shows the threshold (suitable values in [0,1]) for assessing p-values significance. Default value is 0.05.

```
library(MXM)
```

```
## Registered S3 method overwritten by 'sets':
##   method        from
##   print.element ggplot2
```

```
veri_na<-veri[-which(is.na(veri$N)),]
veri_target<-veri_na$N
veri_na2<-as.data.frame(veri_na[,-c(5,6)])
result1<- MXM::SES(target = veri_target,dataset = veri_na2,threshold = 0.1,max_k = 4,
                   test = "testIndReg",ini= NULL,wei= NULL,user_test= NULL,
                   hash       = TRUE, hashObject = NULL,ncores = 1 )
result1@selectedVars
```

```
## [1] 1 2 3 4
```

```
result1@selectedVarsOrder
```

```
## [1] 4 3 1 2
```

```
result1@stats
```

```
## [1]  69.80676 251.86813 341.32139 282.10034
```

```
result1@pvalues
```

```
## [1] -337.2629 -126.1268 -466.6160 -507.3339
```

```
result1@univ
```

```
## $stat
## [1]  69.80676 251.86813 341.32139 282.10034
##
## $pvalue
## [1] -337.2629 -126.1268 -466.6160 -507.3339
```

First display means that four all predictors (region,gender,education, age_group) seleceted as variable for model and second output sorts these variables as significance values. 3rd and 4th results are about association degree between predictors and target variable. (while lower p-values indicate higher association, in test statistics (stats) , higher values indicate higher relation). We can explain such that “age_group”,“education”, “region” and “gender” become predictors for our model in order of importance.

##Creating the Model and Cross Validation Test

Now, we can create a model with selected variable. We apply the linear regression and negative binomial regression. In this chapter we need to “caret” and “MASS” package to use these methods. We will also use Cross validation to evaluate models used for prediction. So we will split the data in training and test data.

```
library(dplyr)
library(caret)
```

```
## Loading required package: lattice
```

```
tra.samples<-veri_na$N %>%
  createDataPartition(p=0.8, list = FALSE)
tra.data<-veri_na[tra.samples,-5]
test.data<-veri_na[-tra.samples,-5]
model.lm <- lm(N ~., data = tra.data)
predictions.lm <- model.lm %>% predict(test.data)
```

```
data.frame( R2 = R2(predictions.lm, test.data$N),
            RMSE = RMSE(predictions.lm, test.data$N),
            MAE = MAE(predictions.lm, test.data$N))
```

```
##          R2     RMSE      MAE
## 1 0.5024948 93.16646 51.43951
```

Because the mean and variance of our target variable is not equal and categorical predictors- continuous (integer actually), we are appliying the neg. binomial regression. Negative binomial regression can be used for over-dispersed count data, that is when the conditional variance exceeds the conditional mean.

```
library(MASS)
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:dplyr':
##
##     select
```

```
tra.samples2<-veri_na$N %>%
  createDataPartition(p=0.8, list = FALSE)
tra.data2<-veri_na[tra.samples2,-5]
test.data2<-veri_na[-tra.samples2,-5]
model.nb <- glm.nb(N ~., data = tra.data2)
predictions.nb <- model.nb%>% predict(test.data2)
data.frame( R2 = R2(predictions.nb, test.data2$N),
            RMSE = RMSE(predictions.nb, test.data2$N),
            MAE = MAE(predictions.nb, test.data2$N))
```

```
##          R2     RMSE      MAE
## 1 0.4665269 131.5707 64.44241
```

When comparing two models linear and negative binom regressions, the one that produces the *lowest test sample RMSE* is the preferred model. So we can say that the linear model can be preferred than negative binomial regression.

Conclusion

In this project we have aimed to select related independent variables and set up models. After modelling we did compare these models and picked the more appropiate model.

For the future projects , converting the categoric variables to numerical using dummy variables can be good idea. So there could be more useful and comparable models we can use.