Lab Bayesian

Andres Navarro

2023-04-25

Dataset Lookup

First of all, the objective of this Case Study will be to show the advantages of Bayesian Statistics for small data sets and the ability to estimated better the posterior parameters.. As it is known, Bayesian Statistics allows as to set up prior believes of our predictors with specific probability distributions. This is really useful when we do not have a lot of data and we have some insights on the data. For this reason I have decided to use a data set with 21 variables and I will be reducing the number of observations to simulate what we are trying to show. This data is about the COVID cases in Mexico and the goal is to predict if a patient has COVID or not.

[[https://www.kaggle.com/datasets/meirnizri/covid19-dataset][Dataset]]

```
rm(list = ls())
data = read.csv("data.csv", header = TRUE)
dim(data)
```

```
## [1] 1048575      21
```

```
summary(data)
```

```
##      USMER        MEDICAL_UNIT         SEX         PATIENT_TYPE
##  Min.   :1.000   Min.   : 1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:1.000   1st Qu.: 4.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :2.000   Median :12.000   Median :1.000   Median :1.000
##  Mean   :1.632   Mean   : 8.981   Mean   :1.499   Mean   :1.191
##  3rd Qu.:2.000   3rd Qu.:12.000   3rd Qu.:2.000   3rd Qu.:1.000
##  Max.   :2.000   Max.   :13.000   Max.   :2.000   Max.   :2.000
##   DATE_DIED            INTUBED        PNEUMONIA           AGE
##  Length:1048575     Min.   : 1.00   Min.   : 1.000   Min.   :  0.00
##  Class :character   1st Qu.:97.00   1st Qu.: 2.000   1st Qu.: 30.00
##  Mode  :character   Median :97.00   Median : 2.000   Median : 40.00
##                     Mean   :79.52   Mean   : 3.347   Mean   : 41.79
##                     3rd Qu.:97.00   3rd Qu.: 2.000   3rd Qu.: 53.00
##                     Max.   :99.00   Max.   :99.000   Max.   :121.00
##     PREGNANT        DIABETES           COPD            ASTHMA
##  Min.   : 1.00   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 2.00   1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.: 2.000
##  Median :97.00   Median : 2.000   Median : 2.000   Median : 2.000
##  Mean   :49.77   Mean   : 2.186   Mean   : 2.261   Mean   : 2.243
##  3rd Qu.:97.00   3rd Qu.: 2.000   3rd Qu.: 2.000   3rd Qu.: 2.000
##  Max.   :98.00   Max.   :98.000   Max.   :98.000   Max.   :98.000
##     INMSUPR        HIPERTENSION     OTHER_DISEASE    CARDIOVASCULAR
##  Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.: 2.000
```

```
##  Median : 2.000   Median : 2.000   Median : 2.000   Median : 2.000
##  Mean   : 2.298   Mean   : 2.129   Mean   : 2.435   Mean   : 2.262
##  3rd Qu.: 2.000   3rd Qu.: 2.000   3rd Qu.: 2.000   3rd Qu.: 2.000
##  Max.   :98.000   Max.   :98.000   Max.   :98.000   Max.   :98.000
##     OBESITY       RENAL_CHRONIC       TOBACCO       CLASIFFICATION_FINAL
##  Min.   : 1.000   Min.   : 1.000   Min.   : 1.000   Min.   :1.000
##  1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.: 2.000   1st Qu.:3.000
##  Median : 2.000   Median : 2.000   Median : 2.000   Median :6.000
##  Mean   : 2.125   Mean   : 2.257   Mean   : 2.214   Mean   :5.306
##  3rd Qu.: 2.000   3rd Qu.: 2.000   3rd Qu.: 2.000   3rd Qu.:7.000
##  Max.   :98.000   Max.   :98.000   Max.   :98.000   Max.   :7.000
##       ICU
##  Min.   : 1.00
##  1st Qu.:97.00
##  Median :97.00
##  Mean   :79.55
##  3rd Qu.:97.00
##  Max.   :99.00
```

The raw data set consists of 21 unique features and 1,048,576 unique patients. In the Boolean features, 1 means "yes" and 2 means "no". values as 97 and 99 are missing data.

- sex: 1 for female and 2 for male.
- age: of the patient.
- classification: covid test findings. Values 1-3 mean that the patient was diagnosed with covid in different
- degrees. 4 or higher means that the patient is not a carrier of covid or that the test is inconclusive.
- patient type: type of care the patient received in the unit. 1 for returned home and 2 for hospitalization.
- pneumonia: whether the patient already have air sacs inflammation or not.
- pregnancy: whether the patient is pregnant or not.
- diabetes: whether the patient has diabetes or not.
- copd: Indicates whether the patient has Chronic obstructive pulmonary disease or not.
- asthma: whether the patient has asthma or not.
- inmsupr: whether the patient is immunosuppressed or not.
- hypertension: whether the patient has hypertension or not.
- cardiovascular: whether the patient has heart or blood vessels related disease.
- renal chronic: whether the patient has chronic renal disease or not.
- other disease: whether the patient has other disease or not.
- obesity: whether the patient is obese or not.
- tobacco: whether the patient is a tobacco user.
- usmr: Indicates whether the patient treated medical units of the first, second or third level.
- medical unit: type of institution of the National Health System that provided the care.
- intubed: whether the patient was connected to the ventilator.
- icu: Indicates whether the patient had been admitted to an Intensive Care Unit.
- date died: If the patient died indicate the date of death, and 9999-99-99 otherwise.

Here we can see a summary of the data, first we have to clean and adapt the data so we can work on it. First of all, I will create the variable that we want to predict that is if a patient has been diagnosed with COVID or not.

```r
data$COVID = ifelse(data$CLASIFFICATION_FINAL <= 3, 1, 2)
data = subset(data, select = -c(CLASIFFICATION_FINAL))
```

```r
convertToLogic = function(col.name, df) {
  index = which(names(df) == col.name)
  print(index)
```

```
  if (length(index) != 0) {
    df[, index] = ifelse(df[, index] == 2, 0, df[, index])
    df[, index] = as.logical(df[, index])

  }

  return(df)
}
```

This column will tell us if a patient has been diagnosed with COVID or not. Then, I will factor and format all the other variables to adapt them properly.

```
data = convertToLogic("COVID", data)
```

```
## [1] 21
```

```
data$USMER = ifelse(data$USMER == 2, 0, data$USMER)
data$USMER = as.logical(data$USMER)

data$MEDICAL_UNIT = factor(data$MEDICAL_UNIT)

data$SEX = factor(data$SEX, labels = c("female", "male"), levels = c(1, 2))

data$PATIENT_TYPE = factor(data$PATIENT_TYPE, labels = c("returned home", "hospitalized"), levels = c(1

data$INTUBED = factor(data$INTUBED, labels = c("intubed", "not intubed"), levels = c(1, 2))

data$PNEUMONIA = factor(data$PNEUMONIA, labels = c("pneumonia", "not pneumonia"), levels = c(1, 2))

data$PREGNANT = factor(data$PREGNANT, labels = c("pregnant", "not pregnant"), levels = c(1, 2))

data$DIABETES = factor(data$DIABETES, labels = c("diabetes", "not diabetes"), levels = c(1, 2))

data$COPD = factor(data$COPD, labels = c("copd", "not copd"), levels = c(1, 2))

data$ASTHMA = factor(data$ASTHMA, labels = c("asthma", "not asthma"), levels = c(1, 2))

data$INMSUPR = factor(data$INMSUPR, labels = c("inmsupr", "not inmsupr"), levels = c(1, 2))

data$HIPERTENSION = factor(data$HIPERTENSION, labels = c("hipertension", "not hipertension"), levels = 

data$OTHER_DISEASE = factor(data$OTHER_DISEASE, labels = c("other desease", "not other desease"), level

data$CARDIOVASCULAR = factor(data$CARDIOVASCULAR, labels = c("cardiovascular", "not cardiovascular"), l

data$OBESITY = factor(data$OBESITY, labels = c("obesity", "not obesity"), levels = c(1, 2))

data$RENAL_CHRONIC = factor(data$RENAL_CHRONIC, labels = c("renal chronic", "not renal chronic"), level

data$TOBACCO = factor(data$TOBACCO, labels = c("tobacco", "not tobacco"), levels = c(1, 2))

data$ICU = factor(data$ICU, labels = c("icu", "not icu"), levels = c(1, 2))

data = subset(data, select = -c(DATE_DIED))
```

```r
summary(data)
```

```
##    USMER          MEDICAL_UNIT        SEX               PATIENT_TYPE
##  Mode :logical   12     :602995   female:525064   returned home:848544
##  FALSE:662903    4      :314405   male  :523511   hospitalized :200031
##  TRUE :385672    6      : 40584
##                  9      : 38116
##                  3      : 19175
##                  8      : 10399
##                  (Other): 22901
##         INTUBED               PNEUMONIA           AGE
##  intubed     : 33656   pneumonia    :140038   Min.   :  0.00
##  not intubed:159050   not pneumonia:892534   1st Qu.: 30.00
##  NA's       :855869   NA's         : 16003   Median : 40.00
##                                               Mean   : 41.79
##                                               3rd Qu.: 53.00
##                                               Max.   :121.00
##
##          PREGNANT              DIABETES            COPD
##  pregnant    :  8131   diabetes    :124989   copd    :  15062
##  not pregnant:513179   not diabetes:920248   not copd:1030510
##  NA's        :527265   NA's        :  3338   NA's    :   3003
##
##
##
##
##         ASTHMA               INMSUPR                HIPERTENSION
##  asthma    :  31572   inmsupr    :  14170   hipertension    :162729
##  not asthma:1014024   not inmsupr:1031001   not hipertension:882742
##  NA's      :   2979   NA's       :   3404   NA's            :  3104
##
##
##
##
##            OTHER_DISEASE               CARDIOVASCULAR           OBESITY
##  other desease    :  28040   cardiovascular    :  20769   obesity    :159816
##  not other desease:1015490   not cardiovascular:1024730   not obesity:885727
##  NA's             :   5045   NA's              :   3076   NA's       :  3032
##
##
##
##
##            RENAL_CHRONIC           TOBACCO           ICU
##  renal chronic    :  18904   tobacco    : 84376   icu    : 16858
##  not renal chronic:1026665   not tobacco:960979   not icu:175685
##  NA's             :   3006   NA's       :  3220   NA's   :856032
##
##
##
##
##    COVID
##  Mode :logical
##  FALSE:656596
##  TRUE :391979
```

```
##
##
##
##
```

Here we see that the data is correctly formated but there are some missing value, so let's fix that.

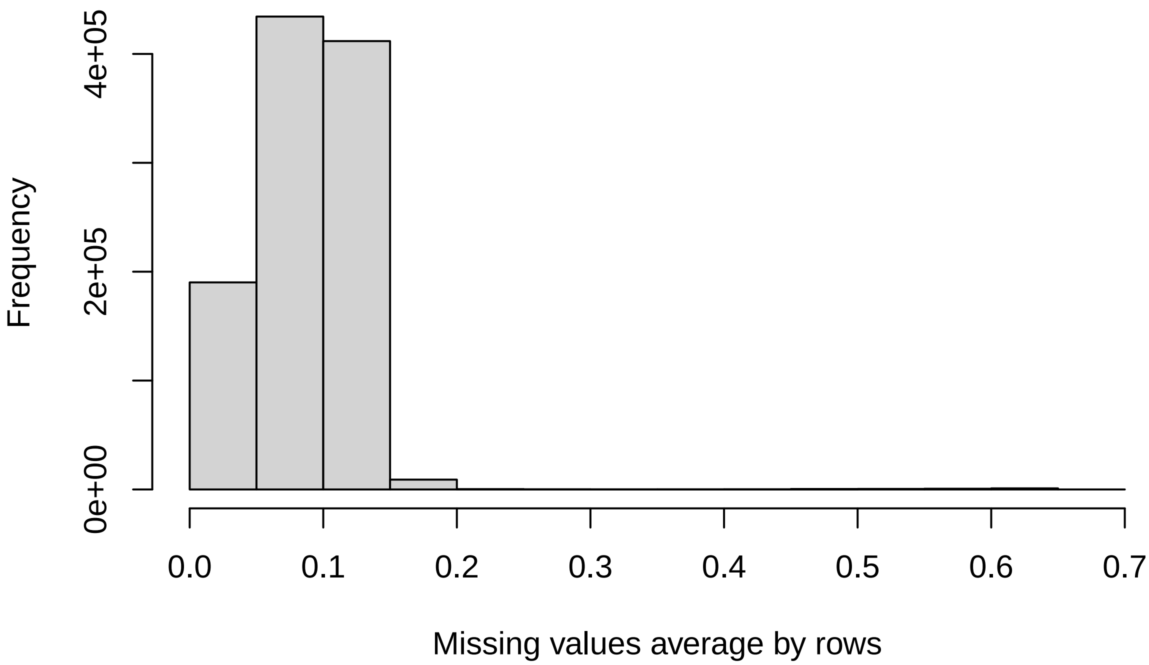
Data Cleaning

First we will see how manu missing values there are by rows so we can remove some columns that have a lot of missing values.

```
print(length(which(is.na(data))))
```

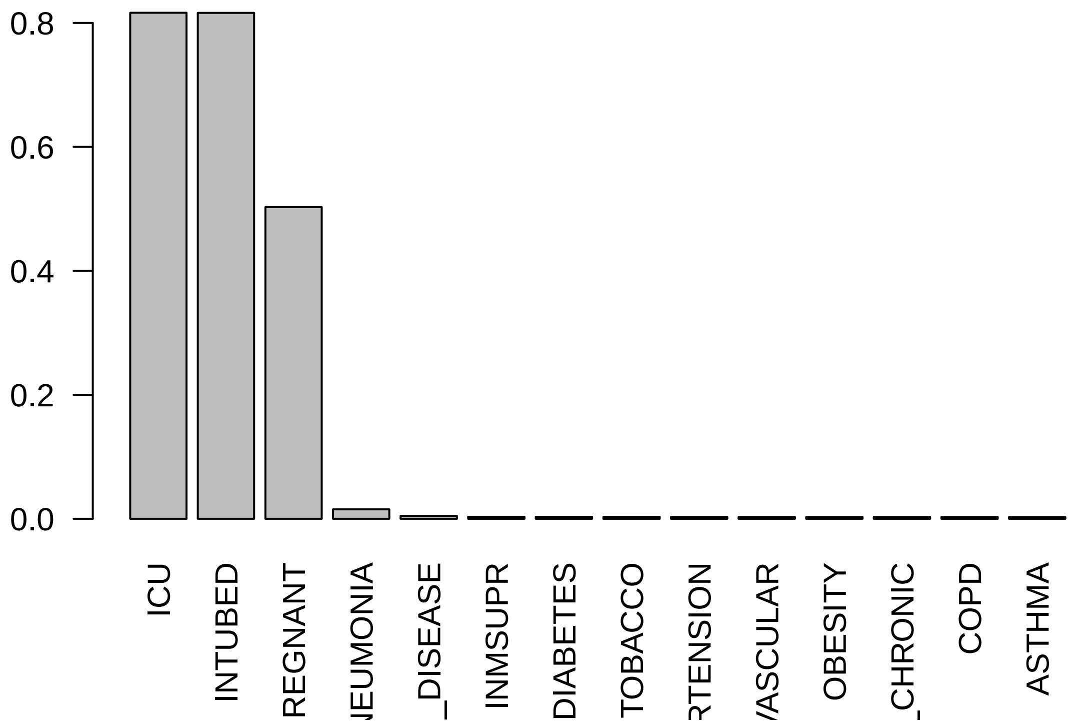
```
## [1] 2288376
```

```
hist(rowMeans(is.na(data)), xlab = c("Missing values average by rows"), main = c())
```



Here we see that there are 3 columns with the most missing values.

```
indexesEmptyCols = which(colMeans(is.na(data)) != 0)

colsWithNA = sort(colMeans(is.na(data[, indexesEmptyCols])),
                  decreasing = TRUE)

barplot(colsWithNA, las=2)
```



And the columns that have the most missing values are ICU, INTUBED, and PREGNANT, so let's remove them.

```
data = subset(data, select = -c(ICU, INTUBED, PREGNANT))

print(length(which(is.na(data))))
```

```
## [1] 49210
```

```
data = na.omit(data)

length(unique(which(is.na(data))))
```

```
## [1] 0
```

```
summary(data)
```

```
##    USMER           MEDICAL_UNIT        SEX                 PATIENT_TYPE
##  Mode :logical    12      :591811   female:513216   returned home:833253
##  FALSE:658255     4       :307177   male  :511936   hospitalized :191899
##  TRUE :366897     6       : 37868
##                   9       : 37384
##                   3       : 18660
##                   8       : 10097
##                   (Other): 22155
##          PNEUMONIA            AGE                  DIABETES
##  pneumonia    :137599   Min.   :  0.00    diabetes    :122415
##  not pneumonia:887553   1st Qu.: 30.00    not diabetes:902737
##                         Median : 40.00
##                         Mean   : 41.89
##                         3rd Qu.: 53.00
##                         Max.   :121.00
##
```

```
##       COPD                ASTHMA              INMSUPR
##  copd    :  14376   asthma    : 30497   inmsupr    :  13588
##  not copd:1010776   not asthma:994655   not inmsupr:1011564
##
##
##
##
##
##           HIPERTENSION             OTHER_DISEASE
##  hipertension    :159577   other desease    : 27131
##  not hipertension:865575   not other desease:998021
##
##
##
##
##
##            CARDIOVASCULAR           OBESITY                RENAL_CHRONIC
##  cardiovascular    :  20126   obesity    :156961   renal chronic    :  18351
##  not cardiovascular:1005026   not obesity:868191   not renal chronic:1006801
##
##
##
##
##
##        TOBACCO         COVID
##  tobacco    : 82675   Mode :logical
##  not tobacco:942477   FALSE:636274
##                       TRUE :388878
##
##
##
##
```

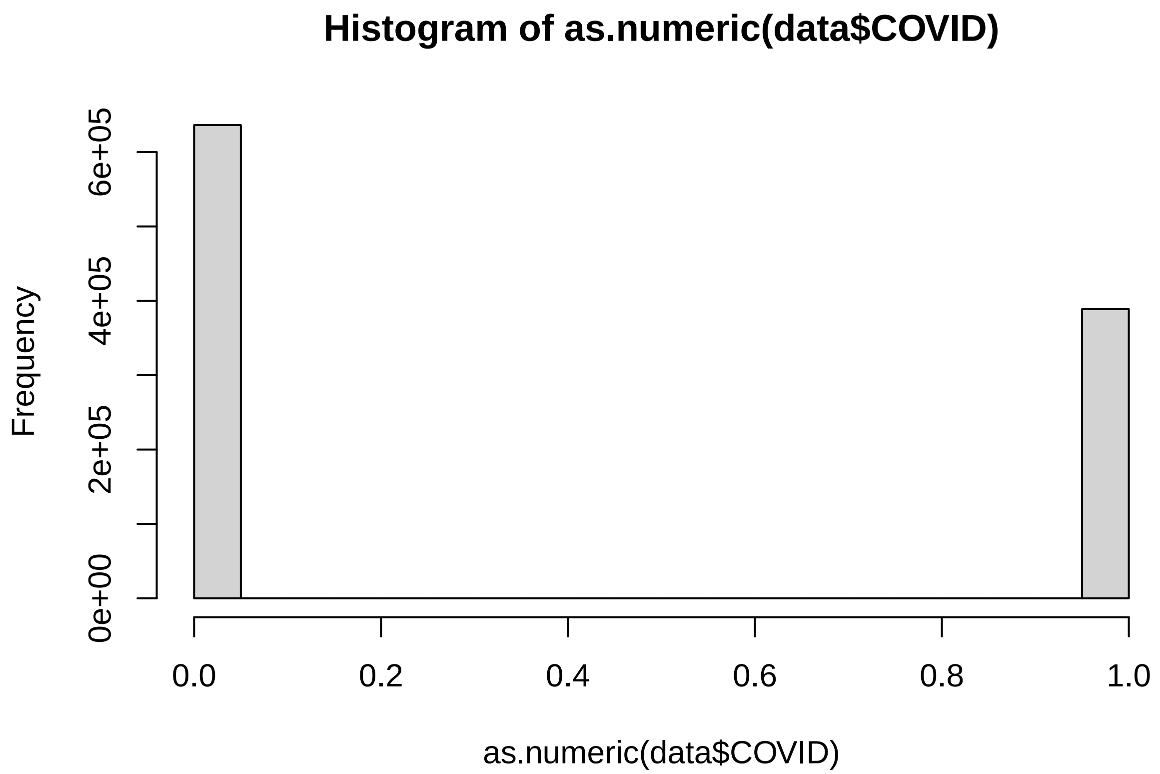
Now the data set is clean so let's start working on it. But first we will shrink it to 1000 observations to work with.

```
data.small = data[sample(nrow(data), size=1000),]
```

Bayesian Analysis of the covid variable

First of all, let's plot a histogram of the COVID variable (the one we want to predict) and see.

```
rm(list = setdiff(ls(), c("data", "data.small")))

hist(as.numeric(data$COVID))
```



This is as we expected as we are going to be predicting a binary variable. So let's use a Bernoulli distribution to explain this data and see how well it fits. First of all lets compute the analytical posterior distribution of the covid variable.

Analytical Study

1. We assume a Bernoulli distribution for COVID, we will use X to denote that variable.

$$X \mid \theta \sim Bernoulli(\theta)$$

$$f(x \mid \theta) = \theta^x \cdot (1-\theta)^x$$

2. As we do not have any prior knowledge on the probability of a patient of having covid, we will define the prior distribution as an improper prior. Moreover, we will be using a Beta distribution as in the end we will get a posterior conjugate which will be much easier to work with.

$$\theta \sim Beta(0,0)$$

$$f(\theta \mid 0,0) = \frac{\theta^{0-1} \cdot (1-\theta)^{0-1}}{B(0,0)}$$

3. Now we get the likelihood

$$f(data \mid \theta) \propto \theta^k \cdot (1-\theta)^{n-k}$$

Being n the total number of observations and k the positive ones.

4. And finally the posterior distribution

$$f(\theta \mid data) = \frac{\theta^{k-1} \cdot (1-\theta)^{n-k-1}}{B(k, n-k)}$$

$$\theta \mid data \sim Beta(k, n-k)$$

So now that we have the posterior distribution let's obtain the prediction of the next value called Y given the data

$$Y \mid \theta \sim Bernuilli(\theta)$$
$$P(Y=1|data) = \int_{-\infty}^{\infty} P(Y=1|\theta) \cdot P(\theta|data) d\theta = \frac{B(k+1, n-k)}{B(k, n-k)}$$

```
n = as.numeric(length(data.small$COVID))
k = as.numeric(length(which(data.small$COVID)))
print(beta(k+1,n-k)/beta(k, n-k))
```

```
## [1] 0.387
```

And here we can see that the probability of a new patient of having covid is 0.362 that is really close to the ML estimator of 0.38

And finally let's try to obtain the same result numerically

Numerical Study

As we know the distribution of the new observation we will get a random sample and compare.

$$Y \mid \theta \sim Bernuilli(Beta(k, n-k))$$

```
y.sample = rbinom(n, 1, rbeta(1, k, n - k))

mean(y.sample)
```

```
## [1] 0.397
```

Here we see that the estimated probability is almost the same as previously.

```
covid.prob = rbeta(n, k, n - k)
quantile(covid.prob, probs = c(0.025, 0.975))
```

```
##      2.5%     97.5%
## 0.3566339 0.4157329
```

And also we see that the confidence interval for the probability of having covid is pretty narrow, so we can be sure that it is correct.

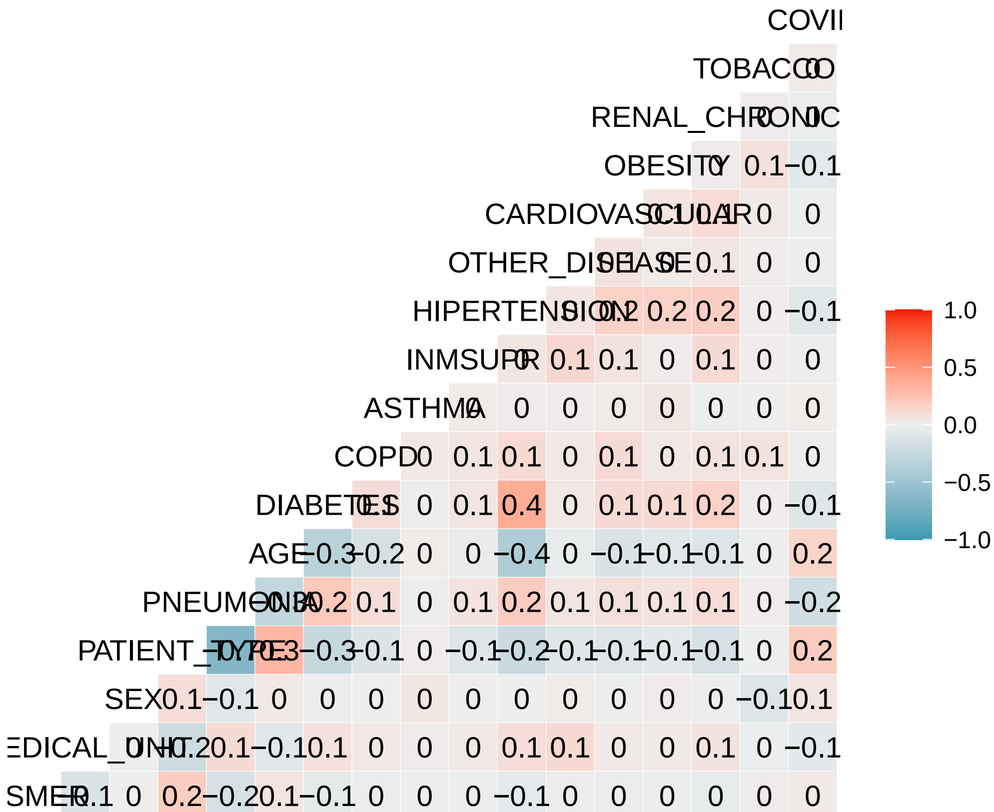
Data Exploration

Now, we will see if the other variables are useful to predict if a patient has covid or not.

```
rm(list = setdiff(ls(), c("data", "data.small")))

library(ggplot2) # GGally
library(GGally)
```

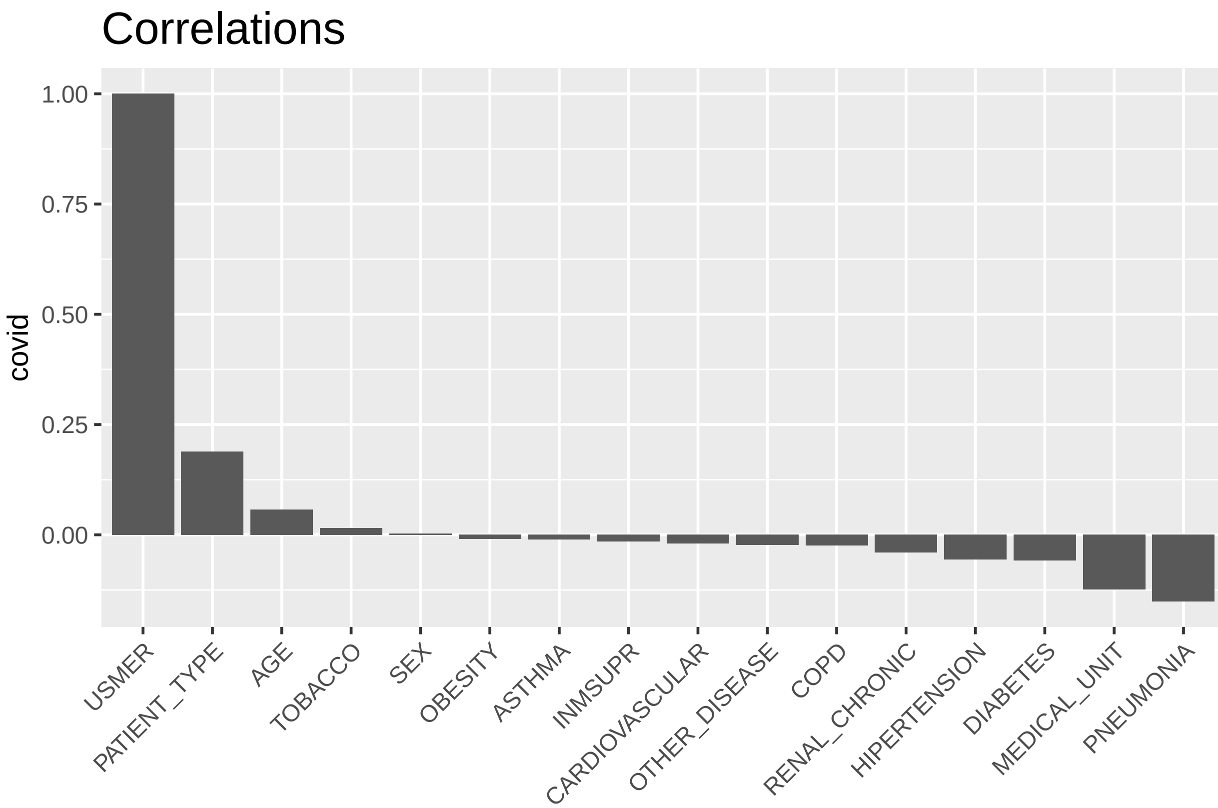
```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
ggcorr(data, cor_matrix = cor(sapply(data, as.numeric)), label = TRUE)
```



Here we see that there is some correlation between the columns but nothing strong with respect to the COVID variable so lets see if we can identify better which columns have more correlation with the covid column.

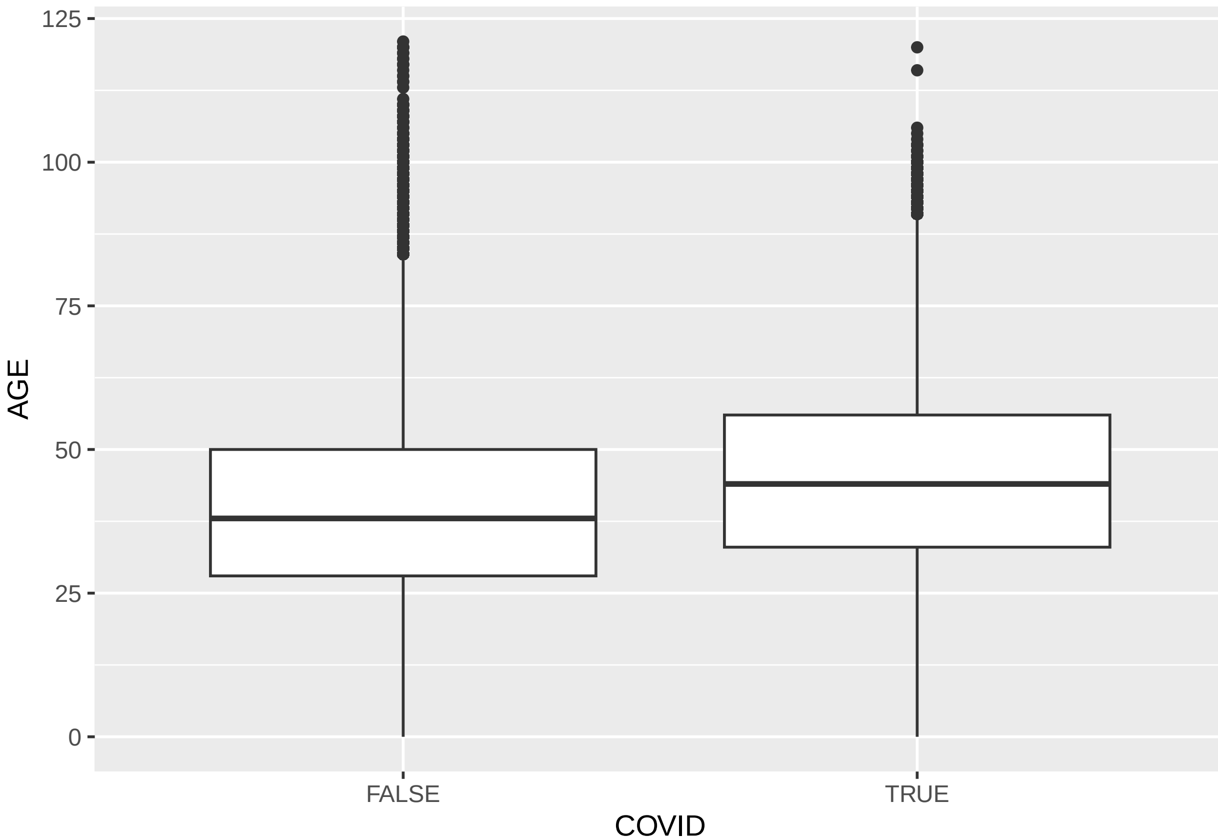
```
corr_covid = sort(cor(sapply(subset(data, select = -c(COVID)), as.numeric))[1,], decreasing = T)

corr = data.frame(corr_covid)

ggplot(corr,aes(x = row.names(corr), y = corr_covid)) + geom_bar(stat = "identity") +
  scale_x_discrete(limits= row.names(corr)) +  labs(x = "", y = "covid", title = "Correlations") +
  theme(plot.title = element_text(hjust = 0, size = rel(1.5)), axis.text.x = element_text(angle = 45, h
```



Here we see that USMER has the most correlation and this makes sense as it indicated if the pacient has received medication.

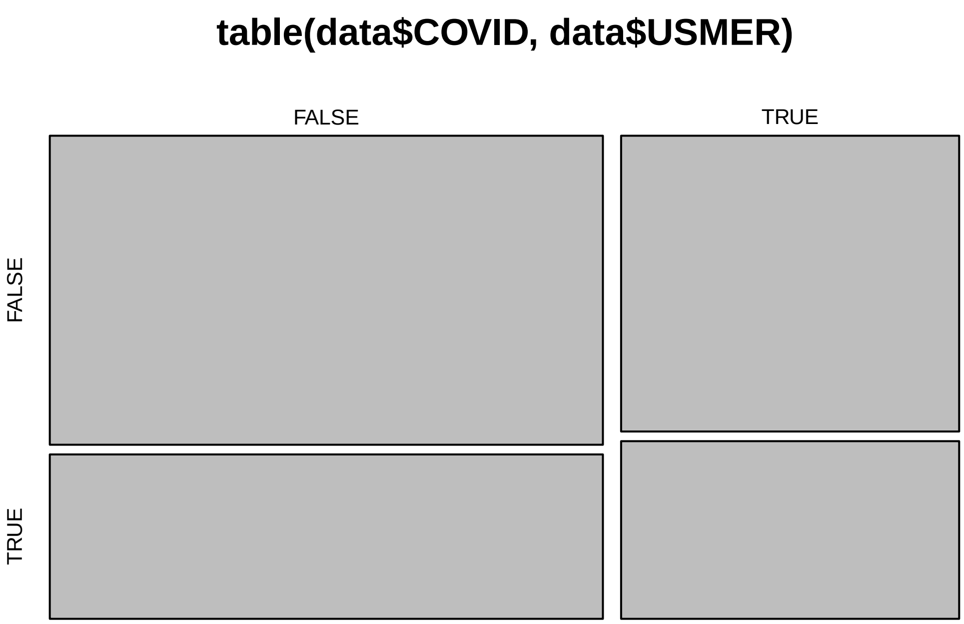
Now let's see how the columns distribute with respect to the covid variable.

```
ggplot(data, aes(x=COVID, y=AGE)) +
  geom_boxplot()
```

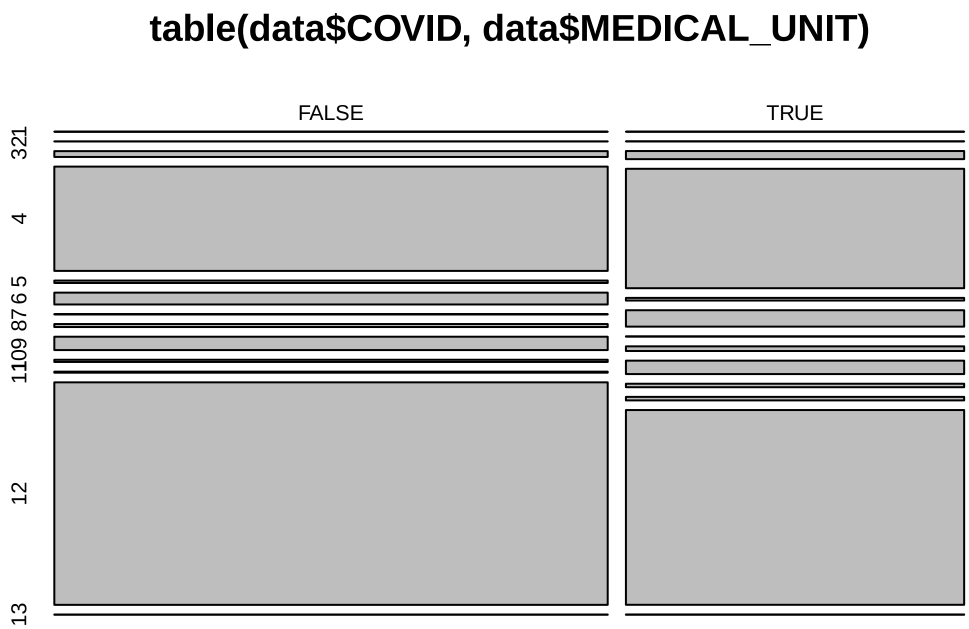


Here we see that there is a visible difference between the mean of the covid, so this can be a useful variable to use in our model.

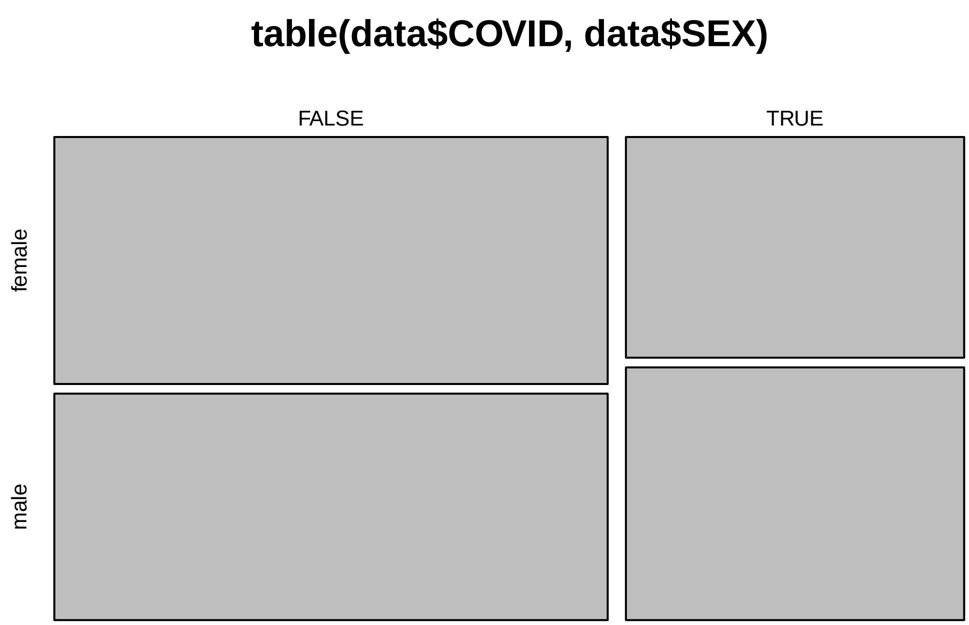
```
plot(table(data$COVID, data$USMER))
```



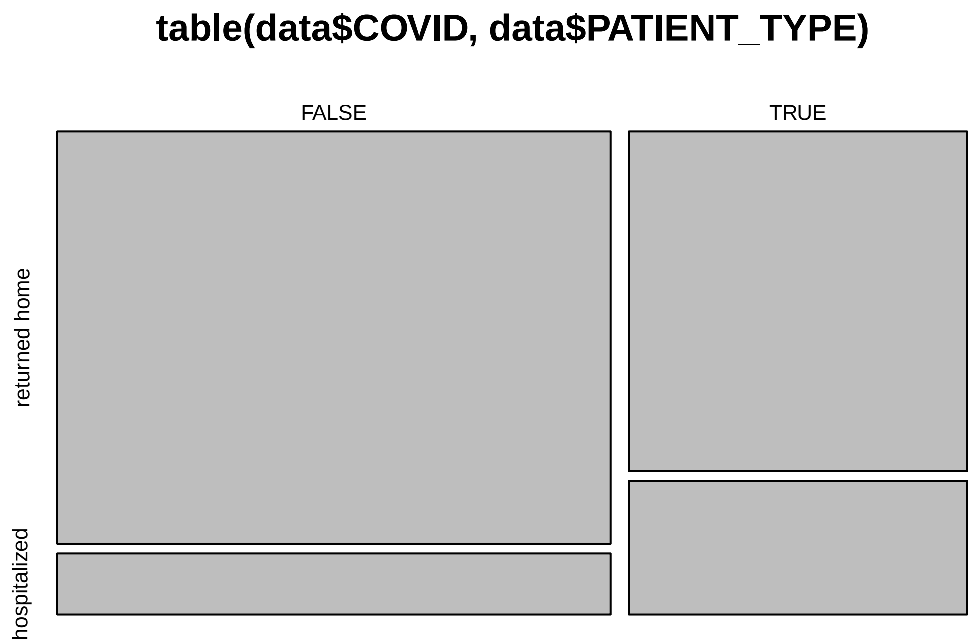

```
plot(table(data$COVID, data$MEDICAL_UNIT))
```



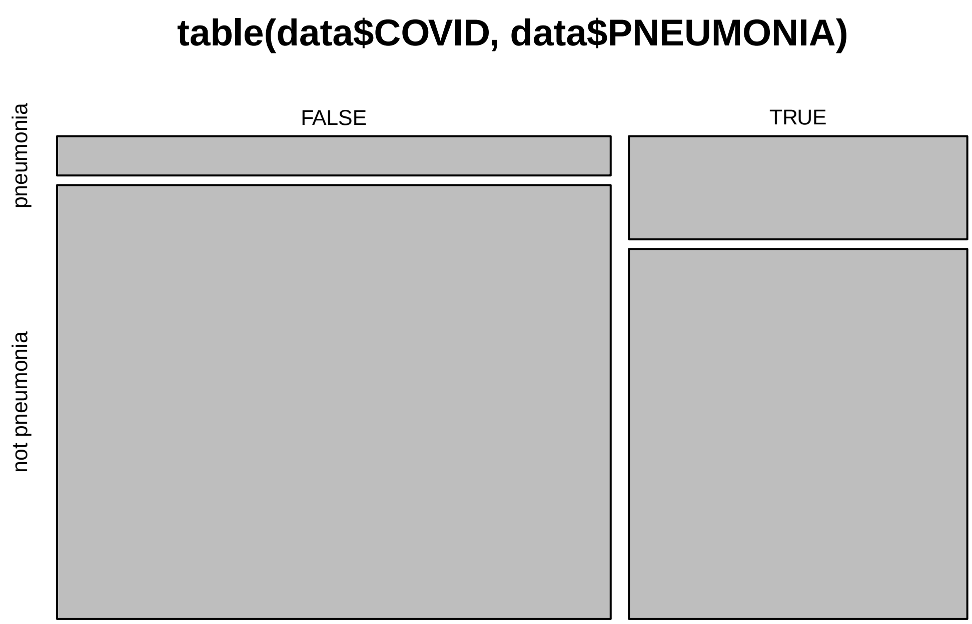

```
plot(table(data$COVID, data$SEX))
```



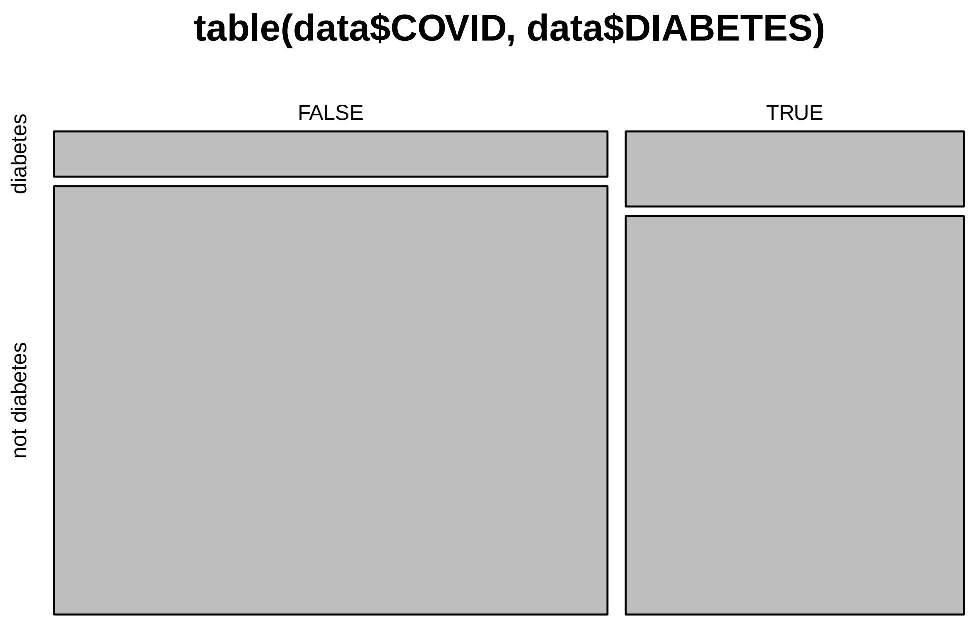

```
plot(table(data$COVID, data$PATIENT_TYPE))
```



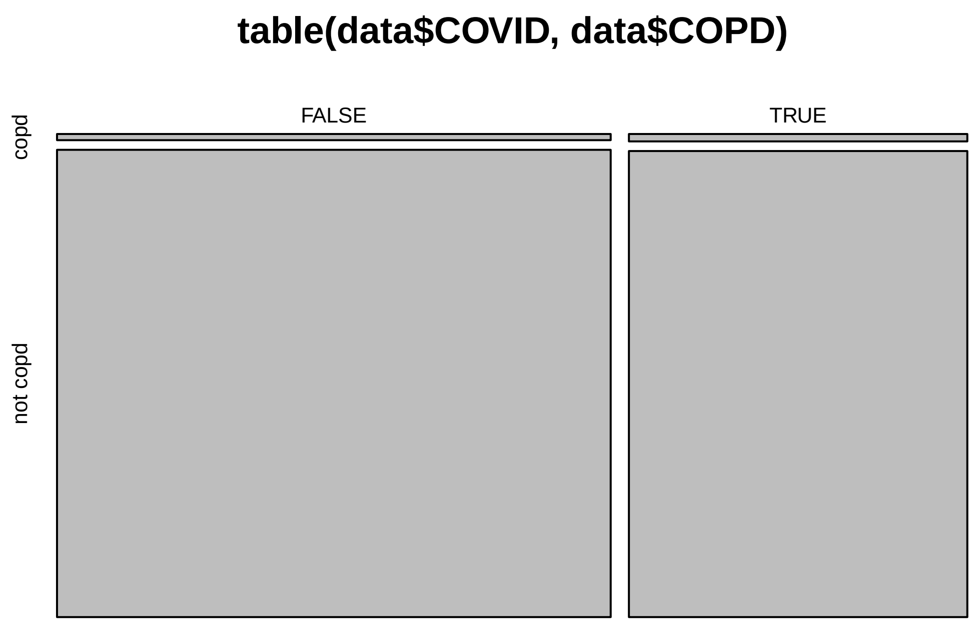

```
plot(table(data$COVID, data$PNEUMONIA))
```



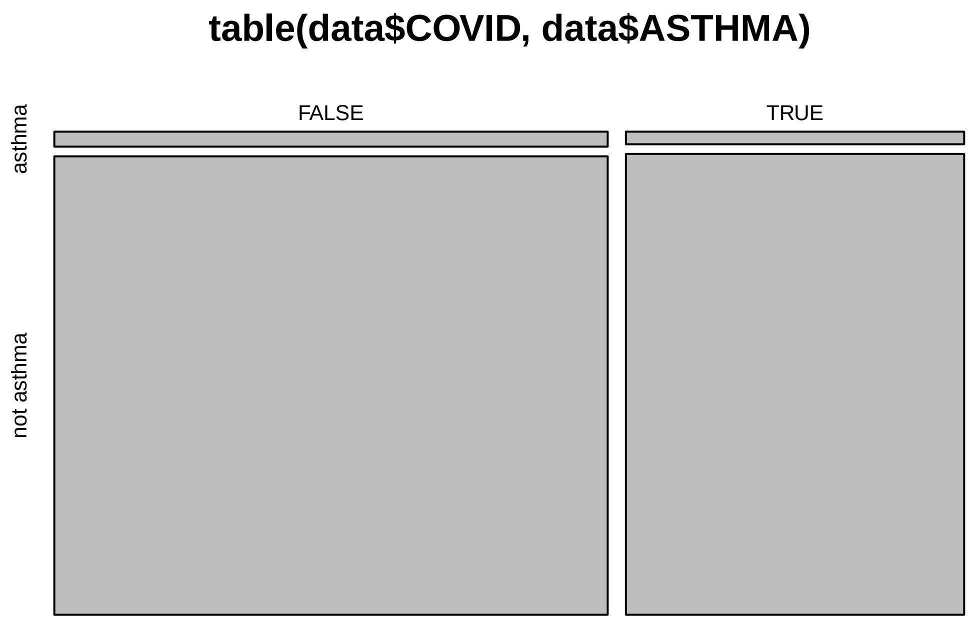

```
plot(table(data$COVID, data$DIABETES))
```



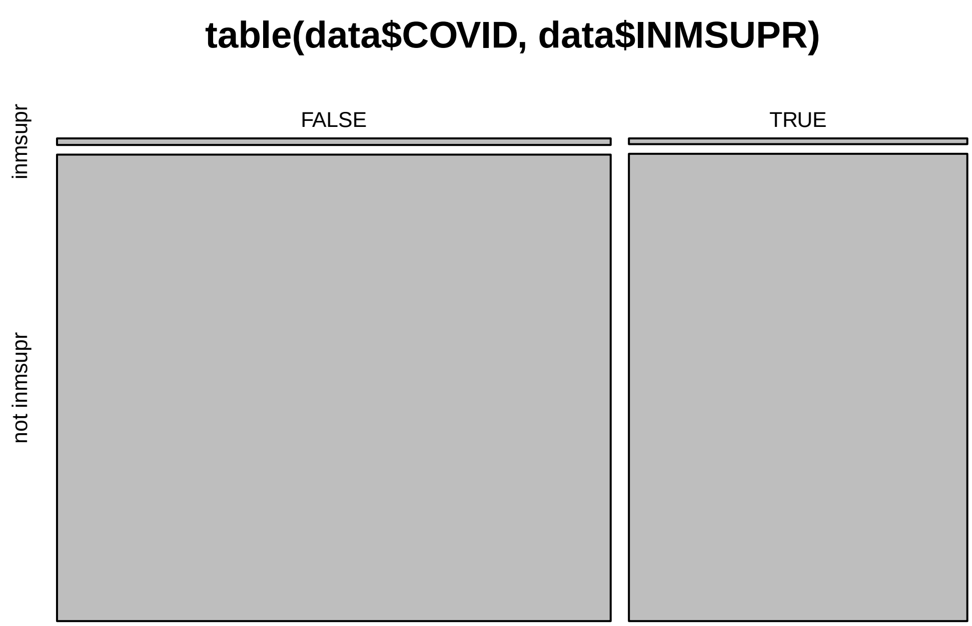

```
plot(table(data$COVID, data$COPD))
```



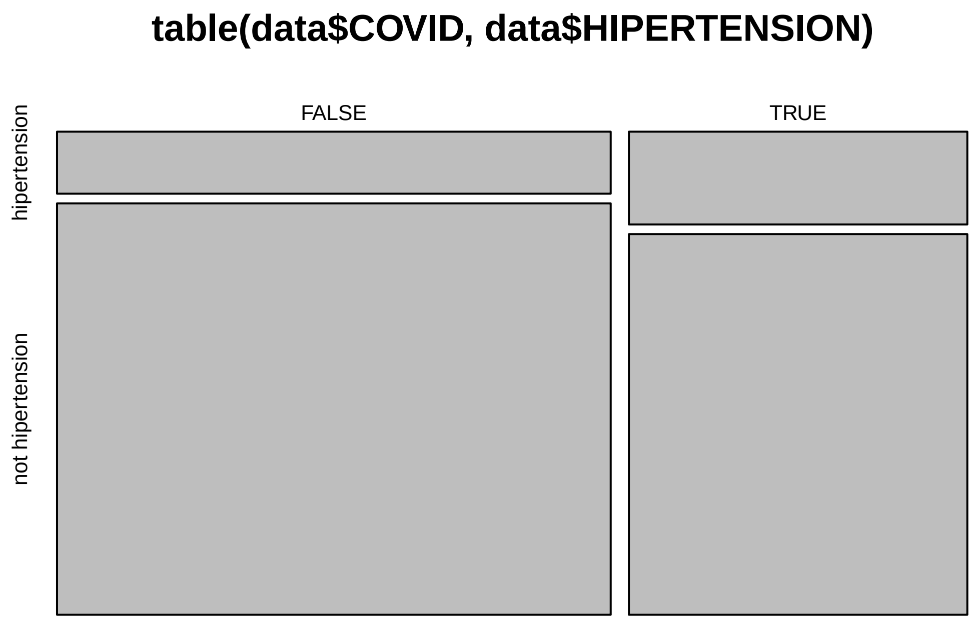

```
plot(table(data$COVID, data$ASTHMA))
```



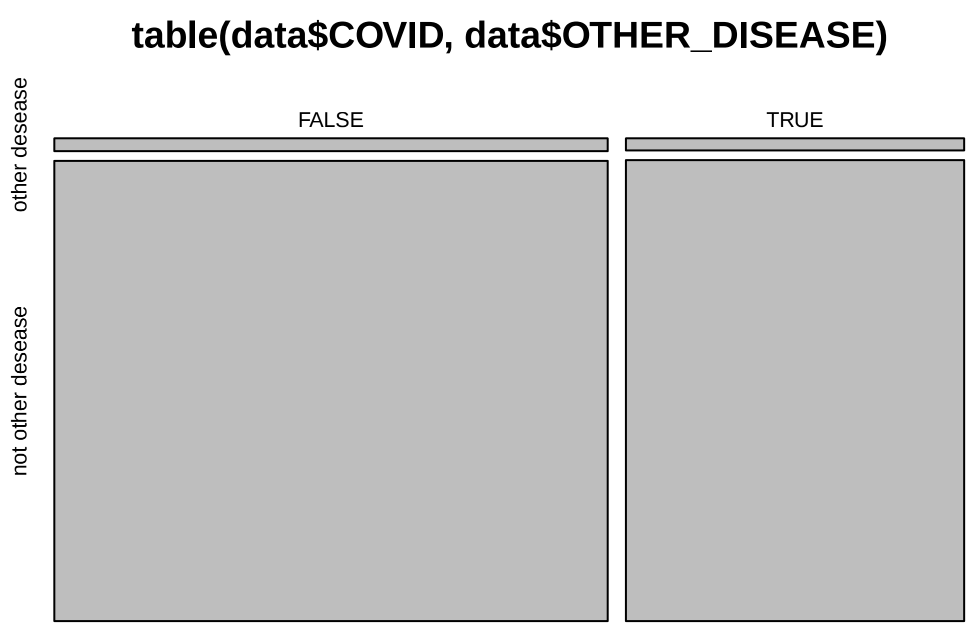

```
plot(table(data$COVID, data$INMSUPR))
```



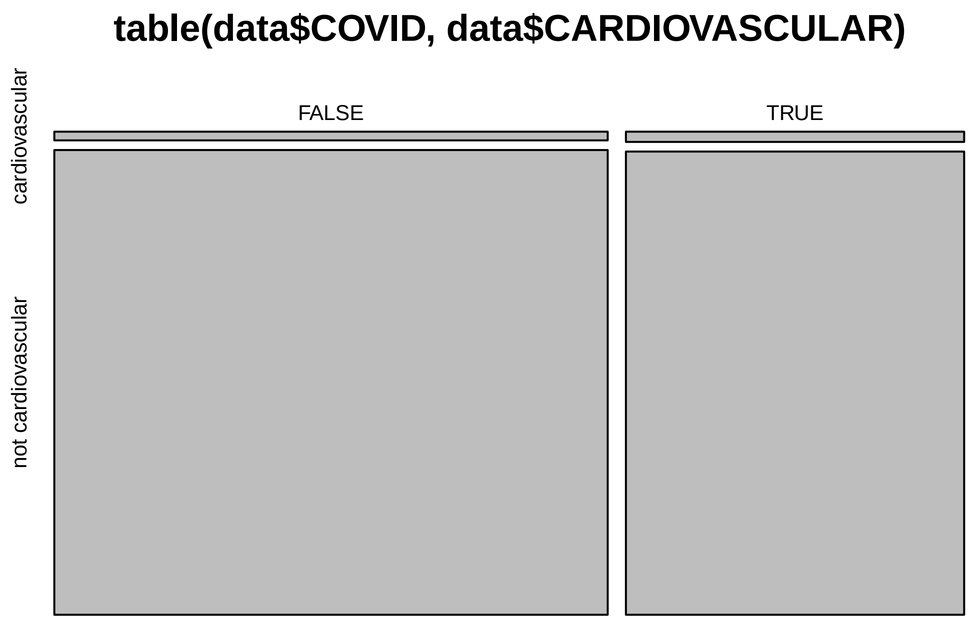

```
plot(table(data$COVID, data$HIPERTENSION))
```



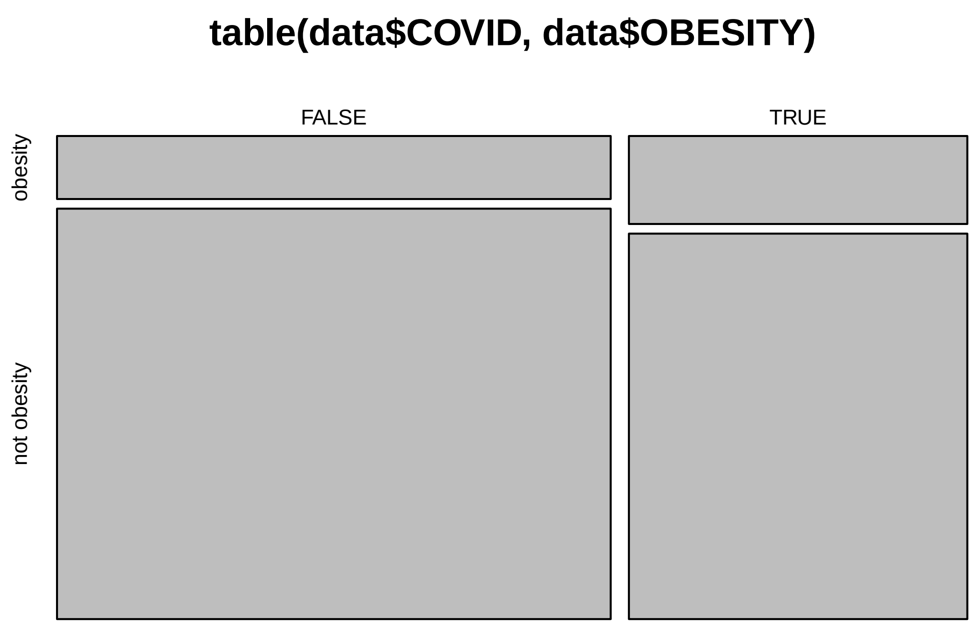

```
plot(table(data$COVID, data$OTHER_DISEASE))
```



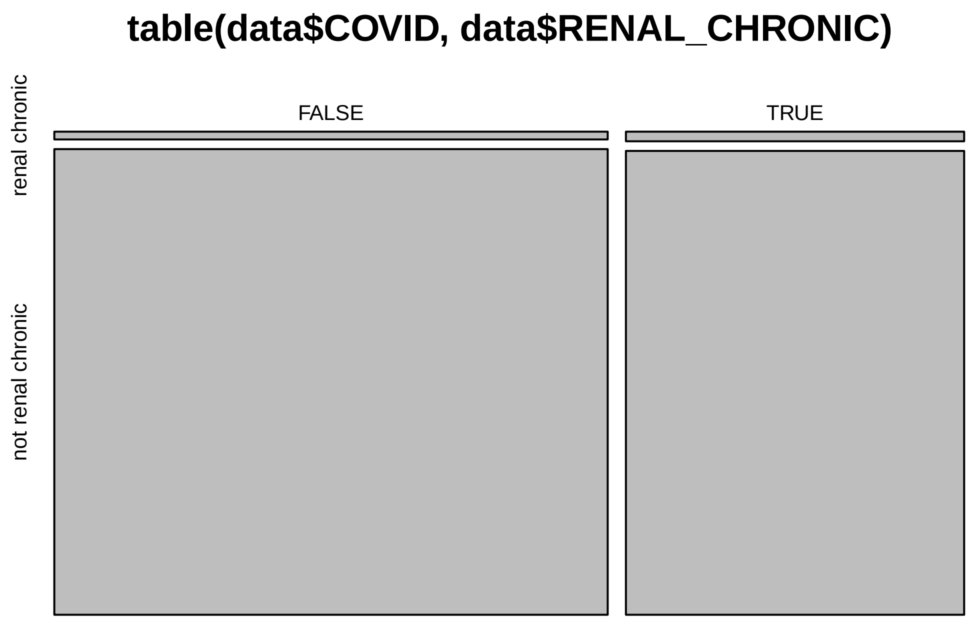

```
plot(table(data$COVID, data$CARDIOVASCULAR))
```



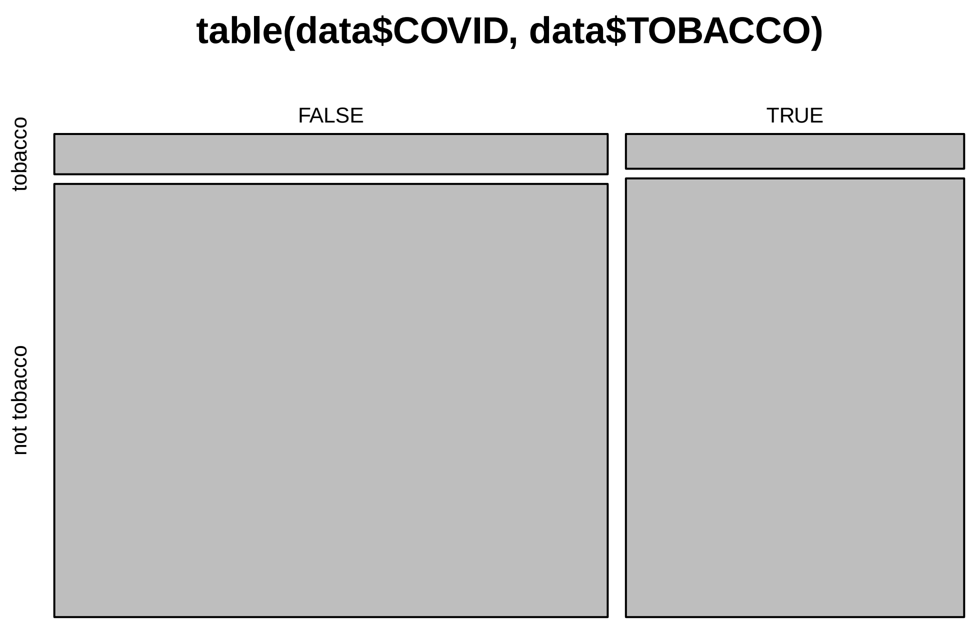

```
plot(table(data$COVID, data$OBESITY))
```




```
plot(table(data$COVID, data$RENAL_CHRONIC))
```




```
plot(table(data$COVID, data$TOBACCO))
```



Now, the columns that show off the most are: MEDICAL_UNIT, SEX, PATITENT_TYPE, and PNEUMONIA. This makes sense and we will see after if we are confident that there is a visible difference.

Frequentist LM

Now let's implement a simple LM model to see how well we can predict a patient to have covid.

```
rm(list = setdiff(ls(), c("data")))

library(caret)
```

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

```r
library(lattice)

data.small = data[sample(nrow(data), size=10000),]

index.test = createDataPartition(data.small$COVID, p = 0.5, list = FALSE)

data.test = data.small[index.test,]
data.train = data.small[-index.test,]

rm(index.test)
```

Now first of all let's try to use all the variables to try to predict if a patient has covid or not.

```r
fit = train(as.factor(COVID) ~ ., data = data.train, method = "glm", family = "binomial")

summary(fit)
```

```
##
## Call:
## NULL
##
## Coefficients: (1 not defined because of singularities)
##                                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)                          -1.549007   1.040384  -1.489 0.136519
## USMERTRUE                            -0.006100   0.065788  -0.093 0.926121
## MEDICAL_UNIT2                       -11.032253 196.969748  -0.056 0.955334
## MEDICAL_UNIT3                         0.661634   0.925289   0.715 0.474574
## MEDICAL_UNIT4                         0.372106   0.896533   0.415 0.678106
## MEDICAL_UNIT5                         0.579578   0.968971   0.598 0.549748
## MEDICAL_UNIT6                        -0.083100   0.909352  -0.091 0.927188
## MEDICAL_UNIT7                        -0.885892   1.491755  -0.594 0.552606
## MEDICAL_UNIT8                         0.403198   0.954278   0.423 0.672648
## MEDICAL_UNIT9                         0.563616   0.909098   0.620 0.535276
## MEDICAL_UNIT10                        1.173119   0.962655   1.219 0.222985
## MEDICAL_UNIT11                        0.685636   0.975880   0.703 0.482316
## MEDICAL_UNIT12                        0.295849   0.896320   0.330 0.741347
## MEDICAL_UNIT13                              NA         NA      NA       NA
## SEXmale                               0.225557   0.061613   3.661 0.000251 ***
## PATIENT_TYPEhospitalized              0.517222   0.105990   4.880 1.06e-06 ***
## `PNEUMONIAnot pneumonia`             -0.487553   0.116301  -4.192 2.76e-05 ***
## AGE                                   0.009648   0.002100   4.593 4.36e-06 ***
## `DIABETESnot diabetes`               -0.283205   0.104088  -2.721 0.006512 **
## `COPDnot copd`                        0.230603   0.276268   0.835 0.403882
## `ASTHMAnot asthma`                    0.151850   0.185813   0.817 0.413805
## `INMSUPRnot inmsupr`                  0.016963   0.265770   0.064 0.949110
## `HIPERTENSIONnot hipertension`       -0.103733   0.096202  -1.078 0.280911
## `OTHER_DISEASEnot other desease`      0.234106   0.179897   1.301 0.193144
## `CARDIOVASCULARnot cardiovascular`    0.268790   0.225357   1.193 0.232977
## `OBESITYnot obesity`                 -0.425641   0.086943  -4.896 9.80e-07 ***
## `RENAL_CHRONICnot renal chronic`      0.243150   0.245701   0.990 0.322361
## `TOBACCOnot tobacco`                  0.056725   0.112339   0.505 0.613596
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6562.6  on 4998  degrees of freedom
## Residual deviance: 6247.4  on 4972  degrees of freedom
## AIC: 6301.4
##
## Number of Fisher Scoring iterations: 10
```

Here we see that there are a lot of variables that are useless. As the p value of the betas is really high for most of them.

```
confusionMatrix(as.factor(data.test$COVID), predict(fit, newdata = data.test))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  2851  323
##      TRUE   1340  487
##
##                Accuracy : 0.6675
##                  95% CI : (0.6542, 0.6805)
##     No Information Rate : 0.838
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1869
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.6803
##             Specificity : 0.6012
##          Pos Pred Value : 0.8982
##          Neg Pred Value : 0.2666
##              Prevalence : 0.8380
##          Detection Rate : 0.5701
##    Detection Prevalence : 0.6347
##       Balanced Accuracy : 0.6408
##
##        'Positive' Class : FALSE
##
```

Here, we see that we get an accuracy of 0.6578 so it is not that bad, probably it is because we only have a few significant variables as we saw in the correlation graph. So let's try a simpler model.

```
fit = train(as.factor(COVID) ~ USMER + PNEUMONIA + MEDICAL_UNIT + DIABETES + HIPERTENSION + AGE + PATIE

summary(fit)
```

```
##
## Call:
## NULL
##
## Coefficients: (1 not defined because of singularities)
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     -0.693216   0.930881  -0.745   0.4565
## USMERTRUE                       -0.014868   0.065410  -0.227   0.8202
```

```
## `PNEUMONIAnot pneumonia`        -0.521051   0.115291  -4.519 6.20e-06 ***
## MEDICAL_UNIT2                 -10.807228 196.969791  -0.055   0.9562
## MEDICAL_UNIT3                   0.807632   0.934440   0.864   0.3874
## MEDICAL_UNIT4                   0.491514   0.906614   0.542   0.5877
## MEDICAL_UNIT5                   0.741058   0.977544   0.758   0.4484
## MEDICAL_UNIT6                   0.030421   0.919609   0.033   0.9736
## MEDICAL_UNIT7                  -0.568439   1.468192  -0.387   0.6986
## MEDICAL_UNIT8                   0.721489   0.961007   0.751   0.4528
## MEDICAL_UNIT9                   0.704366   0.918767   0.767   0.4433
## MEDICAL_UNIT10                  1.296947   0.972232   1.334   0.1822
## MEDICAL_UNIT11                  0.855189   0.986170   0.867   0.3858
## MEDICAL_UNIT12                  0.426837   0.906500   0.471   0.6377
## MEDICAL_UNIT13                        NA         NA      NA       NA
## `DIABETESnot diabetes`         -0.260212   0.102917  -2.528   0.0115 *
## `HIPERTENSIONnot hipertension`  -0.140968   0.094053  -1.499   0.1339
## AGE                             0.009311   0.002071   4.496 6.91e-06 ***
## PATIENT_TYPEhospitalized        0.518198   0.104440   4.962 6.99e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6562.6  on 4998  degrees of freedom
## Residual deviance: 6290.2  on 4981  degrees of freedom
## AIC: 6326.2
##
## Number of Fisher Scoring iterations: 10
```

Now it is better but the medical unit for example, it is only relevant the level 2 and also for other.

```
confusionMatrix(as.factor(data.test$COVID), predict(fit, newdata = data.test))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  2857  317
##      TRUE   1356  471
##
##                Accuracy : 0.6655
##                  95% CI : (0.6522, 0.6785)
##     No Information Rate : 0.8424
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1796
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.6781
##             Specificity : 0.5977
##          Pos Pred Value : 0.9001
##          Neg Pred Value : 0.2578
##              Prevalence : 0.8424
##          Detection Rate : 0.5713
##    Detection Prevalence : 0.6347
```

```
##        Balanced Accuracy : 0.6379
##
##         'Positive' Class : FALSE
##
```

Here we see that the accuracy is almost the same and the kappa so we have not lost a lot of info.

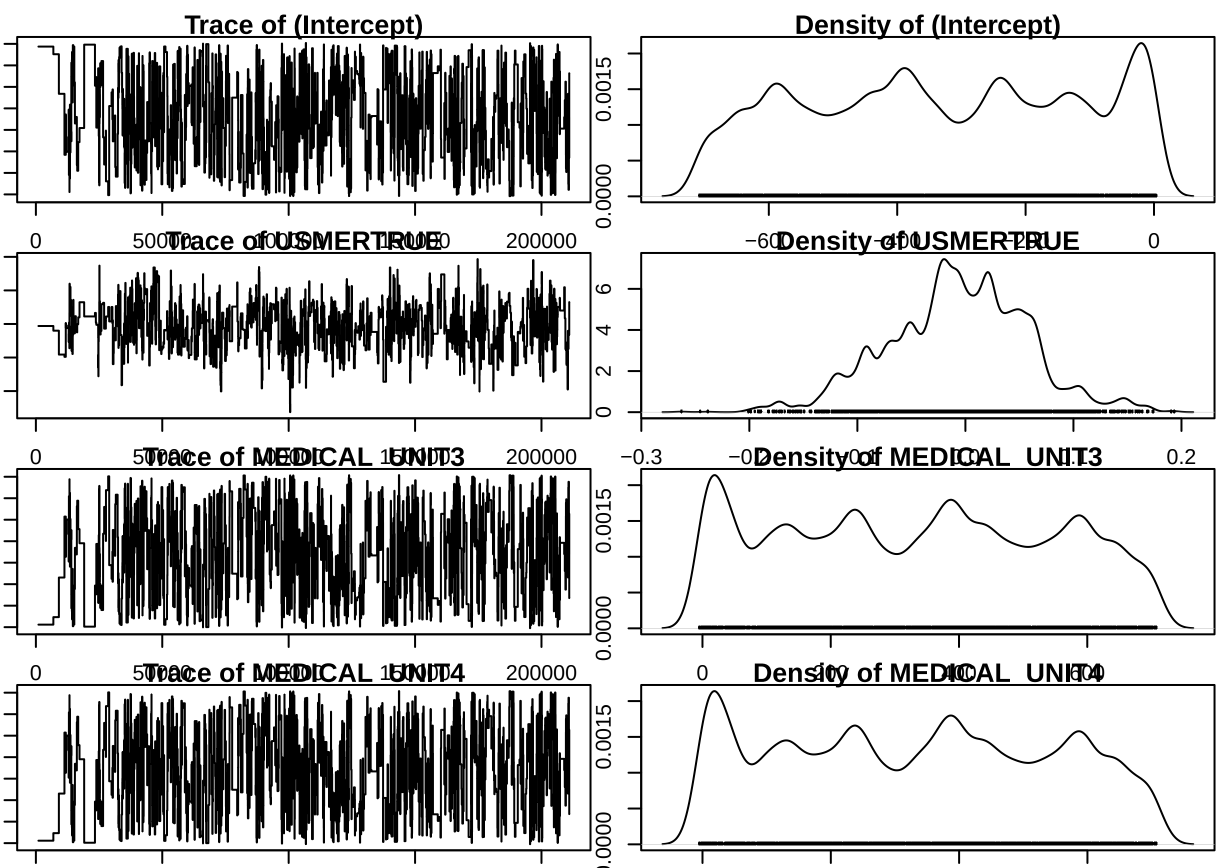
Bayesian LM

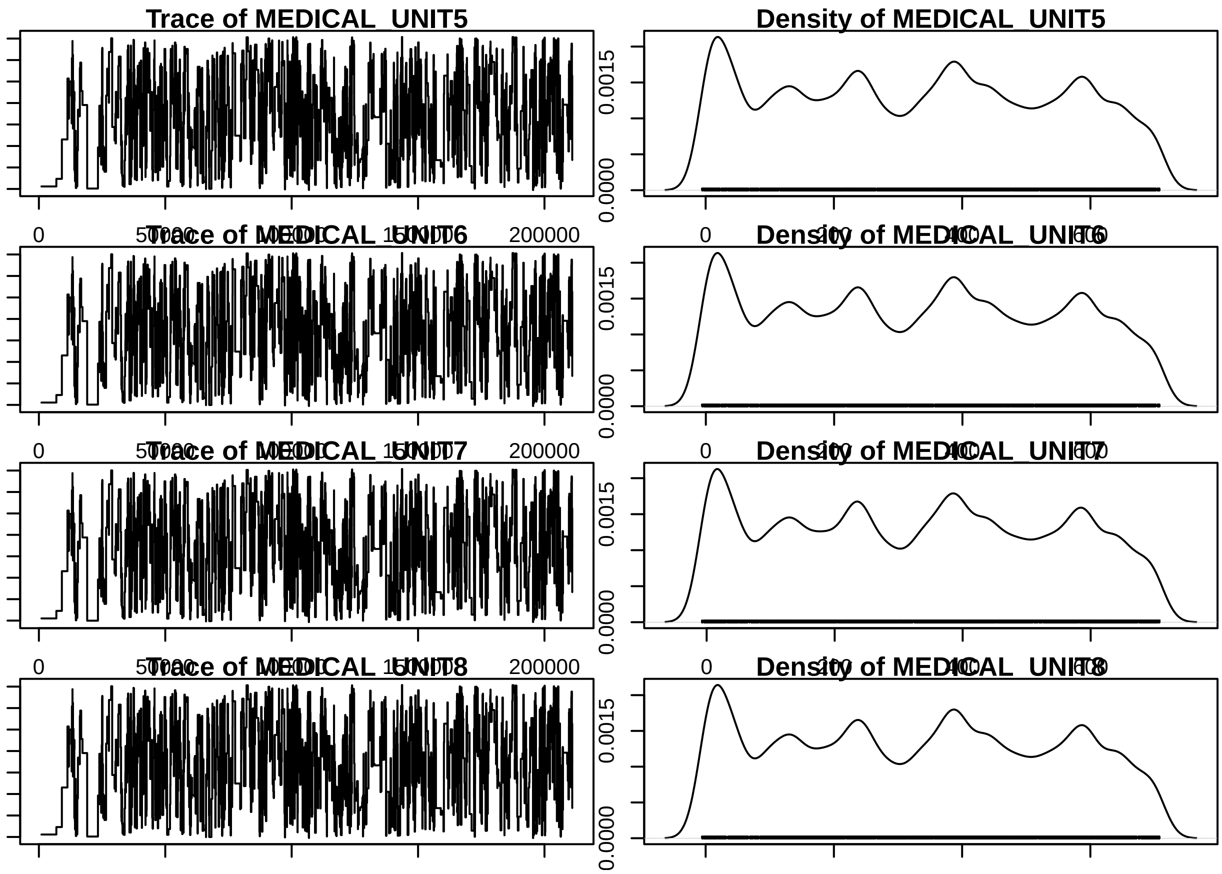
The frequentest approach is easier but we if we want to compute confidence intervals for the parameters or predictive intervals we cannot do them. That is why we will be using the Bayesian approach to better study the effects of each variable with covid and get more conclusions. The power of the Bayesian approach is that we obtain the posterior distribution of the parameters so we can study better the relation and the significance. So let's start.

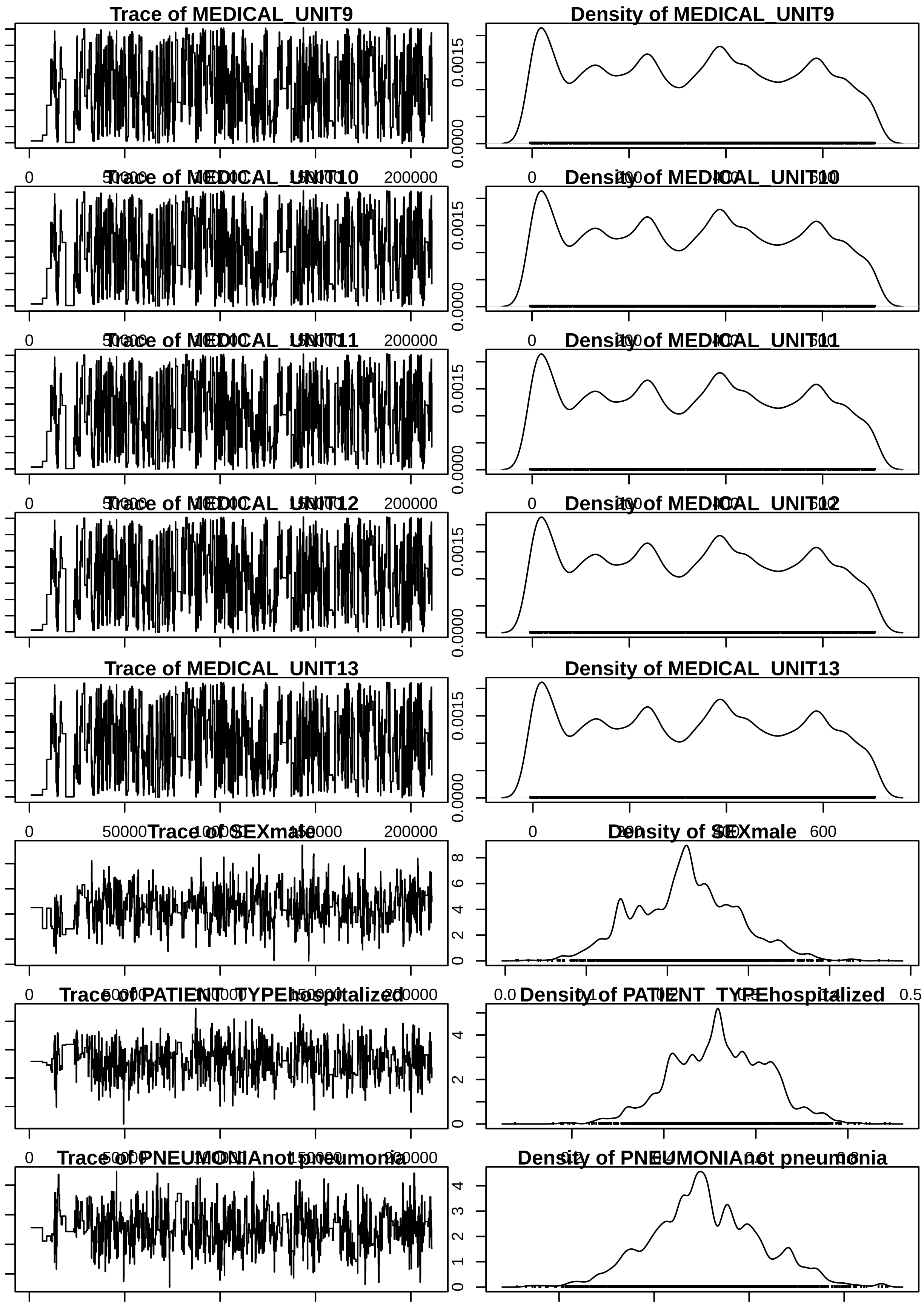
```
library(coda)
library(MASS)
library(MCMCpack)
```

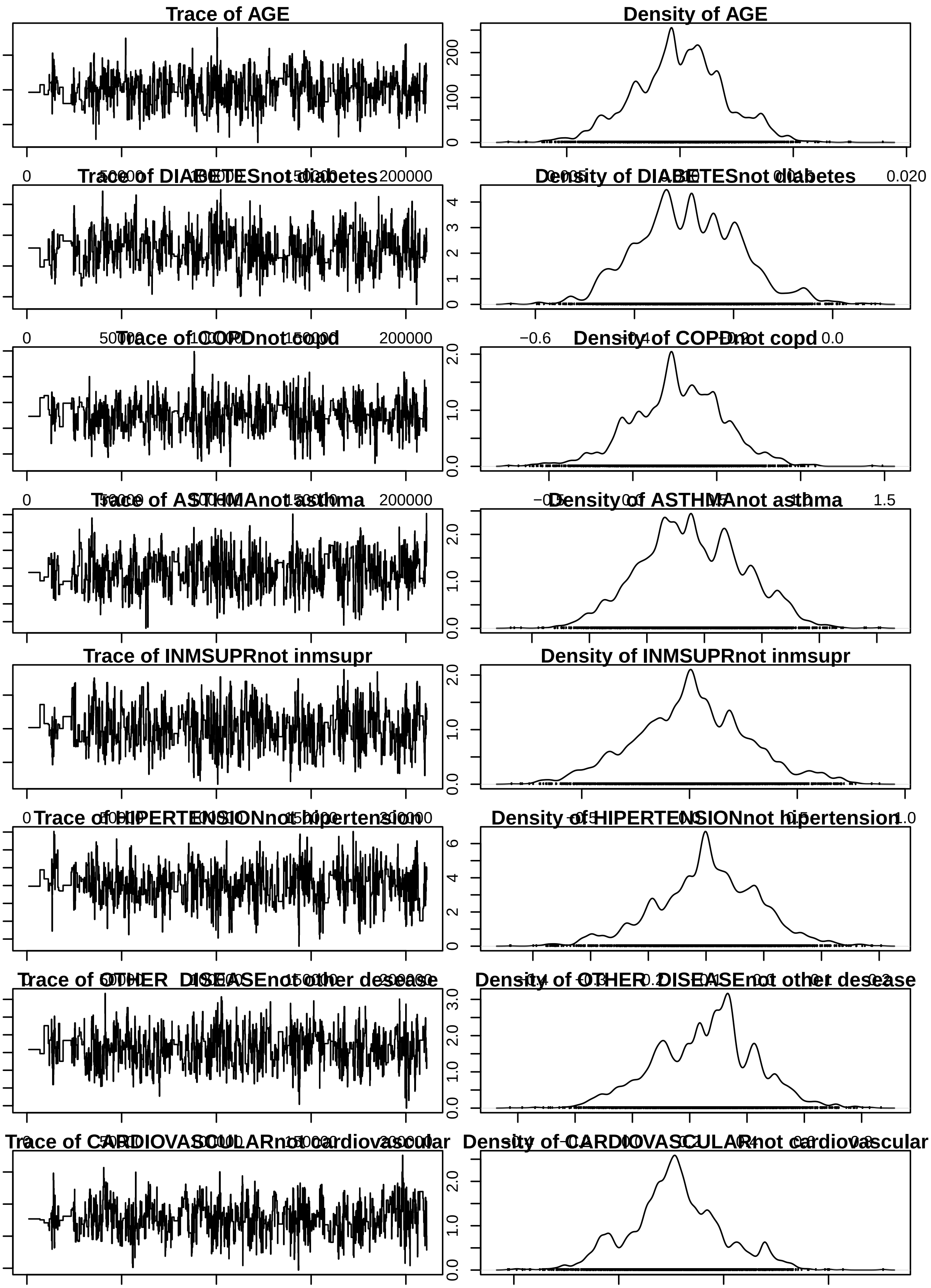
```
rm(list = setdiff(ls(), c("data", "data.small", "data.test", "data.train")))

fit = MCMClogit(COVID ~ ., data = data.train, burnin=1000, mcmc=210000)
```

```
par(mar=c(1, 1, 1, 1))
plot(fit)
```

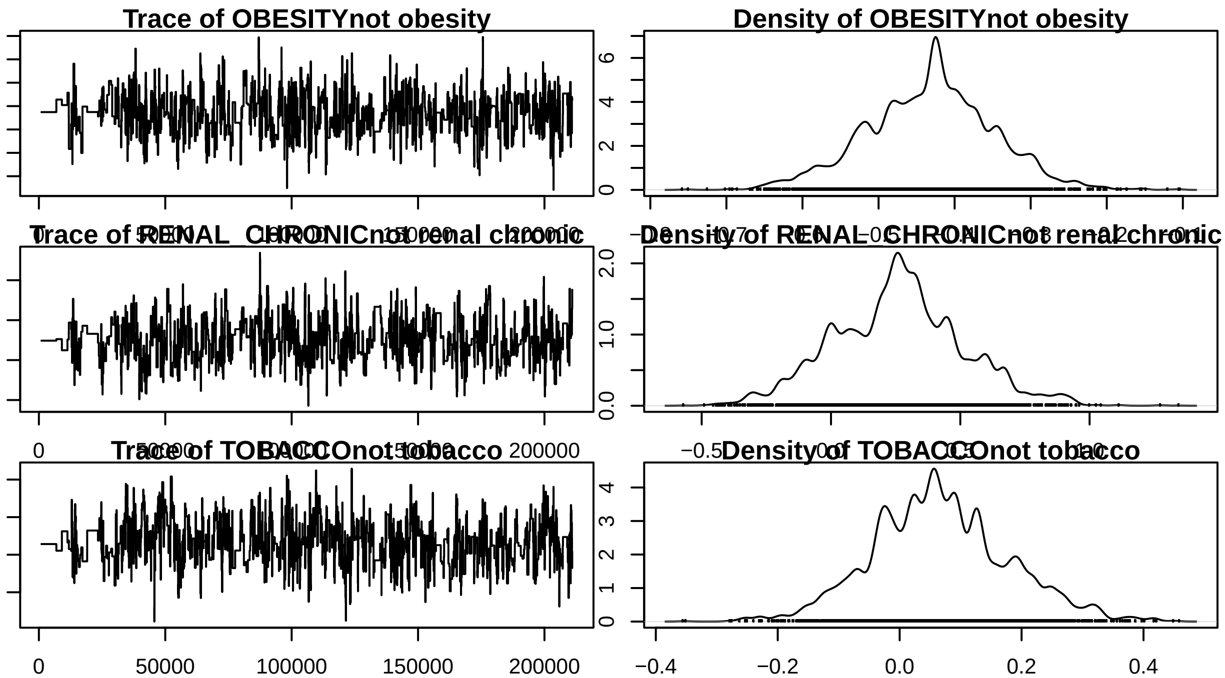



Trace of MEDICAL_UNIT5
Density of MEDICAL_UNIT5
Trace of MEDICAL_UNIT6
Density of MEDICAL_UNIT6
Trace of MEDICAL_UNIT7
Density of MEDICAL_UNIT7
Trace of MEDICAL_UNIT8
Density of MEDICAL_UNIT8
0
200000
0.0000
0.0015


Trace of MEDICAL_UNIT9
Density of MEDICAL_UNIT9
Trace of MEDICAL_UNIT10
Density of MEDICAL_UNIT10
Trace of MEDICAL_UNIT11
Density of MEDICAL_UNIT11
Trace of MEDICAL_UNIT12
Density of MEDICAL_UNIT12
Trace of MEDICAL_UNIT13
Density of MEDICAL_UNIT13
Trace of SEXmale
Density of SEXmale
Trace of PATIENT_TYPEhospitalized
Density of PATIENT_TYPEhospitalized
Trace of PNEUMONIAnot pneumonia
Density of PNEUMONIAnot pneumonia



Trace of AGE
Density of AGE
Trace of DIABETESnot diabetes
Density of DIABETESnot diabetes
Trace of COPDnot copd
Density of COPDnot copd
Trace of ASTHMAnot asthma
Density of ASTHMAnot asthma
Trace of INMSUPRnot inmsupr
Density of INMSUPRnot inmsupr
Trace of HIPERTENSIONnot hipertension
Density of HIPERTENSIONnot hipertension
Trace of OTHER_DISEASEnot other desease
Density of OTHER_DISEASEnot other desease
Trace of CARDIOVASCULARnot cardiovascular
Density of CARDIOVASCULARnot cardiovascular




```
summary(fit)
```

```
##
## Iterations = 1001:211000
## Thinning interval = 1
## Number of chains = 1
## Sample size per chain = 210000
##
## 1. Empirical mean and standard deviation for each variable,
##    plus standard error of the mean:
##
##                                        Mean        SD  Naive SE Time-series SE
## (Intercept)                      -3.315e+02 2.102e+02 4.586e-01      1.237e+01
## USMERTRUE                        -7.082e-03 6.432e-02 1.404e-04      3.512e-03
## MEDICAL_UNIT3                     3.305e+02 2.102e+02 4.587e-01      1.237e+01
## MEDICAL_UNIT4                     3.302e+02 2.102e+02 4.587e-01      1.237e+01
## MEDICAL_UNIT5                     3.304e+02 2.102e+02 4.587e-01      1.237e+01
## MEDICAL_UNIT6                     3.297e+02 2.102e+02 4.587e-01      1.237e+01
## MEDICAL_UNIT7                     3.285e+02 2.102e+02 4.587e-01      1.237e+01
## MEDICAL_UNIT8                     3.302e+02 2.102e+02 4.588e-01      1.238e+01
## MEDICAL_UNIT9                     3.304e+02 2.102e+02 4.587e-01      1.237e+01
## MEDICAL_UNIT10                    3.310e+02 2.102e+02 4.588e-01      1.237e+01
## MEDICAL_UNIT11                    3.305e+02 2.102e+02 4.587e-01      1.237e+01
## MEDICAL_UNIT12                    3.301e+02 2.102e+02 4.587e-01      1.237e+01
## MEDICAL_UNIT13                    3.297e+02 2.102e+02 4.588e-01      1.237e+01
## SEXmale                           2.232e-01 6.345e-02 1.385e-04      3.556e-03
## PATIENT_TYPEhospitalized          5.210e-01 1.025e-01 2.237e-04      5.237e-03
## PNEUMONIAnot pneumonia           -4.900e-01 1.167e-01 2.547e-04      6.279e-03
## AGE                               9.984e-03 2.024e-03 4.416e-06      1.055e-04
## DIABETESnot diabetes             -2.905e-01 1.046e-01 2.284e-04      5.764e-03
## COPDnot copd                      2.674e-01 2.751e-01 6.003e-04      1.438e-02
## ASTHMAnot asthma                  1.584e-01 1.792e-01 3.912e-04      9.317e-03
```

```
## INMSUPRnot inmsupr              1.221e-02 2.655e-01 5.794e-04      1.455e-02
## HIPERTENSIONnot hipertension   -9.758e-02 8.905e-02 1.943e-04      4.602e-03
## OTHER_DISEASEnot other desease  2.542e-01 1.752e-01 3.823e-04      9.469e-03
## CARDIOVASCULARnot cardiovascular 2.721e-01 2.248e-01 4.905e-04      1.171e-02
## OBESITYnot obesity             -4.323e-01 8.318e-02 1.815e-04      4.179e-03
## RENAL_CHRONICnot renal chronic  2.617e-01 2.511e-01 5.480e-04      1.377e-02
## TOBACCOnot tobacco              6.633e-02 1.117e-01 2.437e-04      5.882e-03
##
## 2. Quantiles for each variable:
##
##                                      2.5%       25%       50%       75%
## (Intercept)                    -6.928e+02 -509.58594 -335.57127 -141.04887
## USMERTRUE                      -1.320e-01   -0.04972   -0.00610    0.03678
## MEDICAL_UNIT3                   2.253e+00  139.79847  333.95952  508.19720
## MEDICAL_UNIT4                   1.856e+00  139.42718  333.82917  508.12804
## MEDICAL_UNIT5                   1.842e+00  139.84156  334.02260  508.53996
## MEDICAL_UNIT6                   1.153e+00  139.06455  333.30264  507.57020
## MEDICAL_UNIT7                  -4.698e-01  137.34562  334.23637  505.86906
## MEDICAL_UNIT8                   2.030e+00  139.18505  333.50224  508.73075
## MEDICAL_UNIT9                   2.033e+00  139.51839  333.93265  508.35535
## MEDICAL_UNIT10                  2.444e+00  140.31565  334.39394  508.75437
## MEDICAL_UNIT11                  2.335e+00  139.80875  334.21112  508.13608
## MEDICAL_UNIT12                  1.771e+00  139.38199  333.72562  508.08403
## MEDICAL_UNIT13                  2.485e-01  139.38551  334.27532  509.23532
## SEXmale                         1.051e-01    0.17978    0.22426    0.26490
## PATIENT_TYPEhospitalized        3.179e-01    0.45051    0.51730    0.59470
## PNEUMONIAnot pneumonia         -7.166e-01   -0.56573   -0.49585   -0.41229
## AGE                             5.949e-03    0.00874    0.01001    0.01127
## DIABETESnot diabetes           -4.776e-01   -0.36044   -0.29044   -0.21722
## COPDnot copd                   -2.929e-01    0.09774    0.26009    0.44847
## ASTHMAnot asthma               -1.814e-01    0.03977    0.15185    0.28185
## INMSUPRnot inmsupr             -5.218e-01   -0.15912    0.01011    0.18092
## HIPERTENSIONnot hipertension   -2.976e-01   -0.15084   -0.09844   -0.03817
## OTHER_DISEASEnot other desease -1.019e-01    0.12654    0.27452    0.35631
## CARDIOVASCULARnot cardiovascular -1.511e-01    0.14171    0.26879    0.40693
## OBESITYnot obesity             -6.006e-01   -0.48557   -0.42567   -0.37930
## RENAL_CHRONICnot renal chronic -2.090e-01    0.08979    0.26231    0.42746
## TOBACCOnot tobacco             -1.453e-01   -0.00921    0.05861    0.12871
##                                    97.5%
## (Intercept)                     -3.59546
## USMERTRUE                        0.12214
## MEDICAL_UNIT3                  691.46183
## MEDICAL_UNIT4                  690.92586
## MEDICAL_UNIT5                  690.62031
## MEDICAL_UNIT6                  690.19355
## MEDICAL_UNIT7                  688.96992
## MEDICAL_UNIT8                  691.17591
## MEDICAL_UNIT9                  691.06858
## MEDICAL_UNIT10                 691.53135
## MEDICAL_UNIT11                 691.04387
## MEDICAL_UNIT12                 690.85454
## MEDICAL_UNIT13                 689.60722
## SEXmale                          0.35069
## PATIENT_TYPEhospitalized         0.72167
```

```
## PNEUMONIAnot pneumonia            -0.25679
## AGE                                0.01387
## DIABETESnot diabetes              -0.06078
## COPDnot copd                       0.80971
## ASTHMAnot asthma                   0.50358
## INMSUPRnot inmsupr                 0.57217
## HIPERTENSIONnot hipertension       0.07425
## OTHER_DISEASEnot other desease     0.57825
## CARDIOVASCULARnot cardiovascular   0.73216
## OBESITYnot obesity                -0.27201
## RENAL_CHRONICnot renal chronic     0.80040
## TOBACCOnot tobacco                 0.30335
```

From the Bayesian point of view, we see that the CI for all the parameters does not contain 0 so theoretically all of the predictors are significant with an alpha = 5%.

Lasso

```r
rm(list = setdiff(ls(), c("data", "data.small", "data.test", "data.train")))

library(monomvn)
```

```
## Loading required package: pls

##
## Attaching package: 'pls'

## The following object is masked from 'package:caret':
##
##     R2

## The following object is masked from 'package:stats':
##
##     loadings

## Loading required package: lars

## Loaded lars 1.3

##
## Attaching package: 'monomvn'

## The following object is masked from 'package:MCMCpack':
##
##     rwish
```

```r
x = data.frame(lapply(subset(data.train, select = -c(COVID)), function(x) as.numeric((x))))

adaptTo0And1 = function(col.name, df) {
  index = which(names(df) == col.name)

  if (length(index) != 0) {
    df[, index] = ifelse(df[, index] == 2, 0, df[, index])
  }

  return(df)
}
 x = adaptTo0And1("SEX", x)
```

```r
x = adaptTo0And1("PATIENT_TYPE", x)
x = adaptTo0And1("PNEUMONIA", x)
x = adaptTo0And1("DIABETES", x)
x = adaptTo0And1("COPD", x)
x = adaptTo0And1("ASTHMA", x)
x = adaptTo0And1("INMSUPR", x)
x = adaptTo0And1("HIPERTENSION", x)
x = adaptTo0And1("OTHER_DISEASE", x)
x = adaptTo0And1("CARDIOVASCULAR", x)
x = adaptTo0And1("OBESITY", x)
x = adaptTo0And1("RENAL_CHRONIC", x)
x = adaptTo0And1("TOBACCO", x)

summary(x)
```

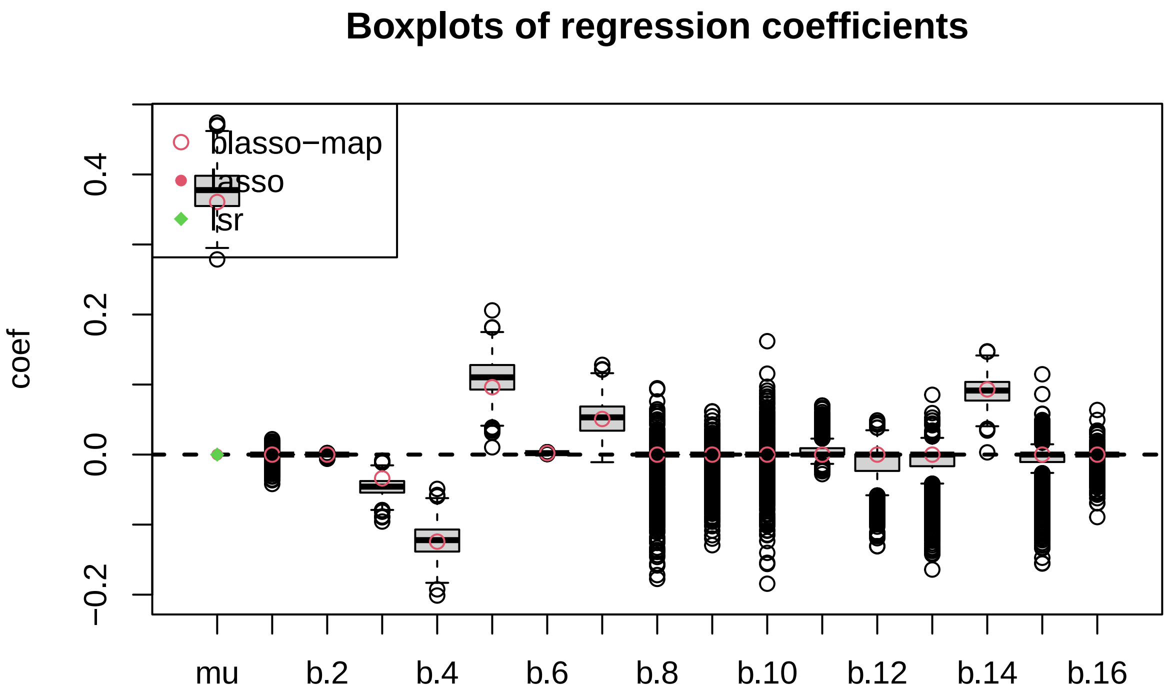
```
##      USMER         MEDICAL_UNIT         SEX          PATIENT_TYPE
##  Min.   :0.0000   Min.   : 2.000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.: 4.000   1st Qu.:0.0000   1st Qu.:1.0000
##  Median :0.0000   Median :12.000   Median :0.0000   Median :1.0000
##  Mean   :0.3535   Mean   : 9.017   Mean   :0.4893   Mean   :0.8286
##  3rd Qu.:1.0000   3rd Qu.:12.000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :13.000   Max.   :1.0000   Max.   :1.0000
##    PNEUMONIA           AGE           DIABETES           COPD
##  Min.   :0.0000   Min.   : 0.00   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:30.00   1st Qu.:0.0000   1st Qu.:0.0000
##  Median :0.0000   Median :40.00   Median :0.0000   Median :0.0000
##  Mean   :0.1258   Mean   :41.53   Mean   :0.1116   Mean   :0.0124
##  3rd Qu.:0.0000   3rd Qu.:52.00   3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :98.00   Max.   :1.0000   Max.   :1.0000
##      ASTHMA            INMSUPR        HIPERTENSION     OTHER_DISEASE
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median :0.00000   Median :0.0000   Median :0.0000   Median :0.00000
##  Mean   :0.02941   Mean   :0.0136   Mean   :0.1436   Mean   :0.03181
##  3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.0000   Max.   :1.00000
##  CARDIOVASCULAR      OBESITY       RENAL_CHRONIC        TOBACCO
##  Min.   :0.0000   Min.   :0.000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median :0.0000   Median :0.000   Median :0.0000   Median :0.00000
##  Mean   :0.0194   Mean   :0.141   Mean   :0.0162   Mean   :0.08102
##  3rd Qu.:0.0000   3rd Qu.:0.000   3rd Qu.:0.0000   3rd Qu.:0.00000
##  Max.   :1.0000   Max.   :1.000   Max.   :1.0000   Max.   :1.00000
```

```r
y = data.train$COVID
fit = blasso(x, y, mprior = c(0,1))
```

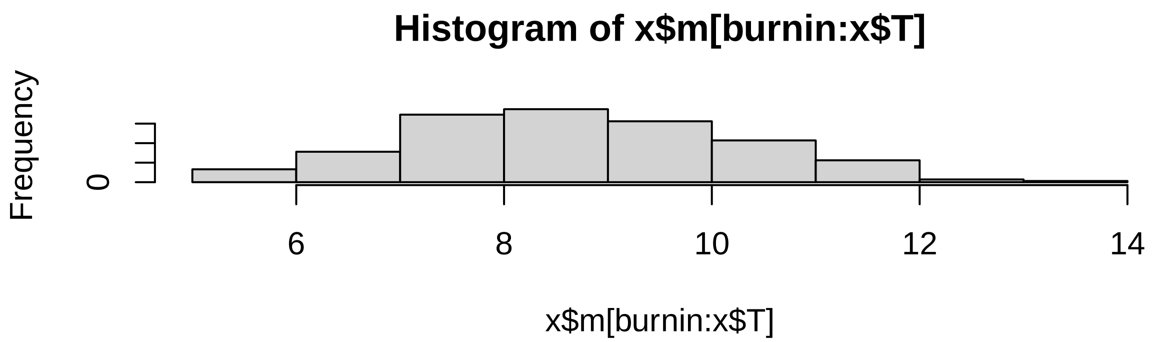
```
## t=100, m=8
## t=200, m=11
## t=300, m=9
## t=400, m=9
## t=500, m=9
## t=600, m=11
## t=700, m=9
```

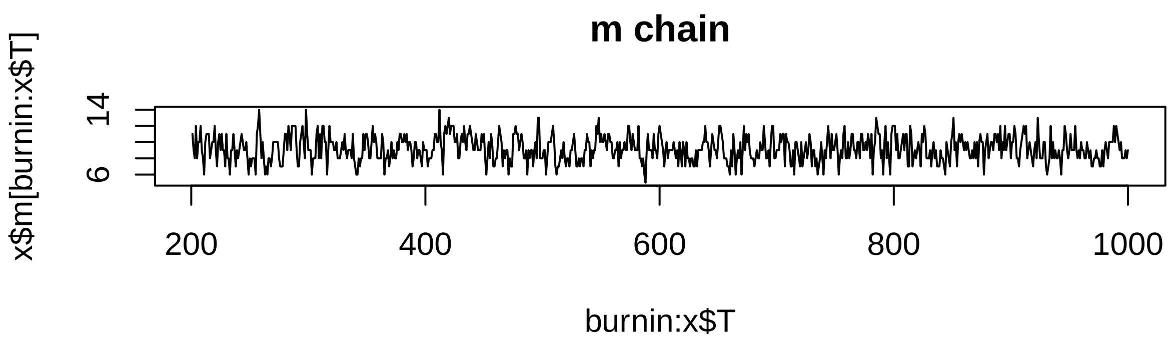
```
## t=800, m=12
## t=900, m=10
```

```
plot(fit, burnin=200)
points(drop(fit$b), col=2, pch=20)
points(drop(fit$b), col=3, pch=18)
legend("topleft", c("blasso-map", "lasso", "lsr"),
       col=c(2,2,3), pch=c(21,20,18))
```

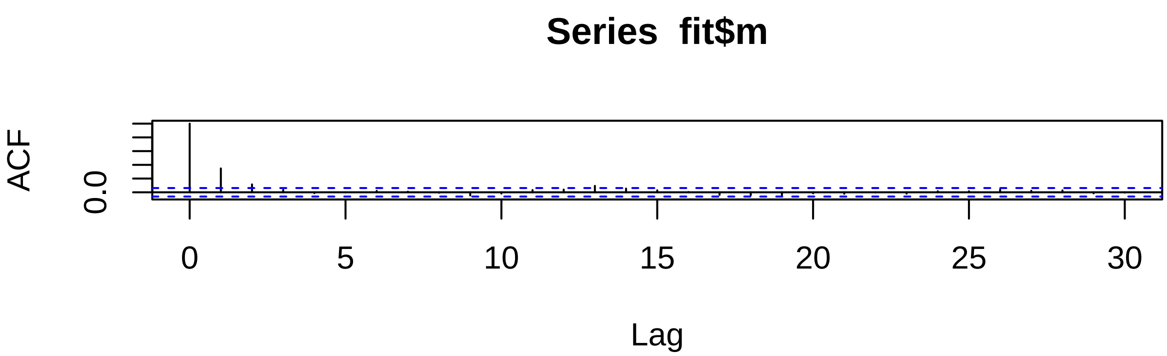



```
plot(fit, burnin=200, which="m")
```






```
acf(fit$m)
```



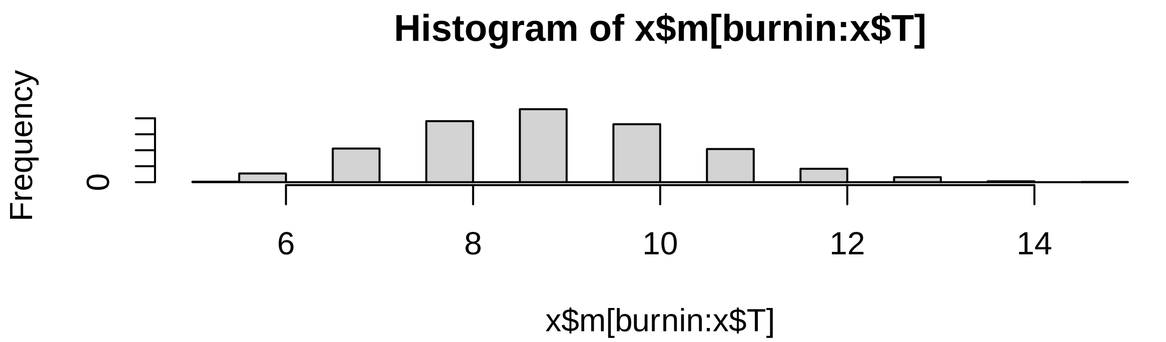

```
fit = blasso(x, y, mprior = c(0,1), T = 10000, thin = 20)
```

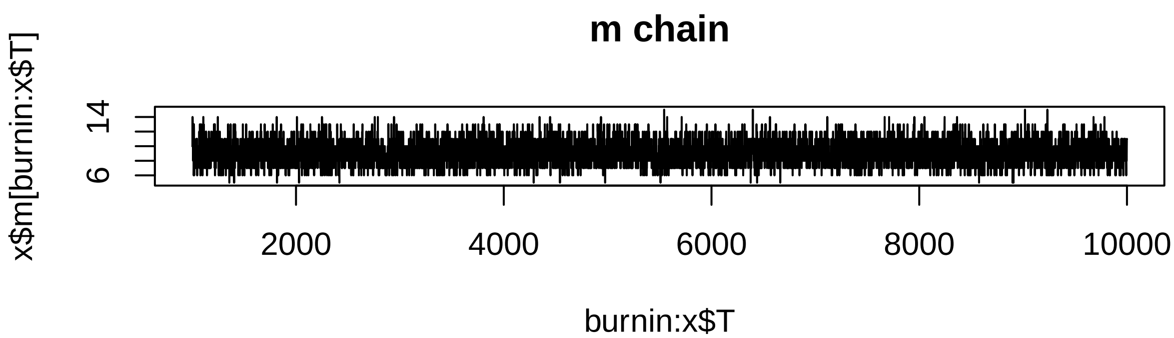
```
## t=100, m=9
## t=200, m=10
## t=300, m=11
## t=400, m=13
## t=500, m=9
## t=600, m=10
## t=700, m=11
## t=800, m=10
## t=900, m=9
## t=1000, m=10
## t=1100, m=9
## t=1200, m=8
## t=1300, m=9
## t=1400, m=10
## t=1500, m=8
```

```
## t=1600, m=9
## t=1700, m=10
## t=1800, m=8
## t=1900, m=8
## t=2000, m=9
## t=2100, m=8
## t=2200, m=11
## t=2300, m=10
## t=2400, m=9
## t=2500, m=10
## t=2600, m=8
## t=2700, m=7
## t=2800, m=6
## t=2900, m=8
## t=3000, m=10
## t=3100, m=8
## t=3200, m=10
## t=3300, m=8
## t=3400, m=8
## t=3500, m=11
## t=3600, m=9
## t=3700, m=8
## t=3800, m=8
## t=3900, m=10
## t=4000, m=10
## t=4100, m=8
## t=4200, m=8
## t=4300, m=11
## t=4400, m=8
## t=4500, m=9
## t=4600, m=9
## t=4700, m=7
## t=4800, m=11
## t=4900, m=9
## t=5000, m=10
## t=5100, m=10
## t=5200, m=12
## t=5300, m=8
## t=5400, m=8
## t=5500, m=9
## t=5600, m=10
## t=5700, m=12
## t=5800, m=7
## t=5900, m=11
## t=6000, m=11
## t=6100, m=11
## t=6200, m=7
## t=6300, m=8
## t=6400, m=9
## t=6500, m=11
## t=6600, m=10
## t=6700, m=11
## t=6800, m=9
## t=6900, m=11
```

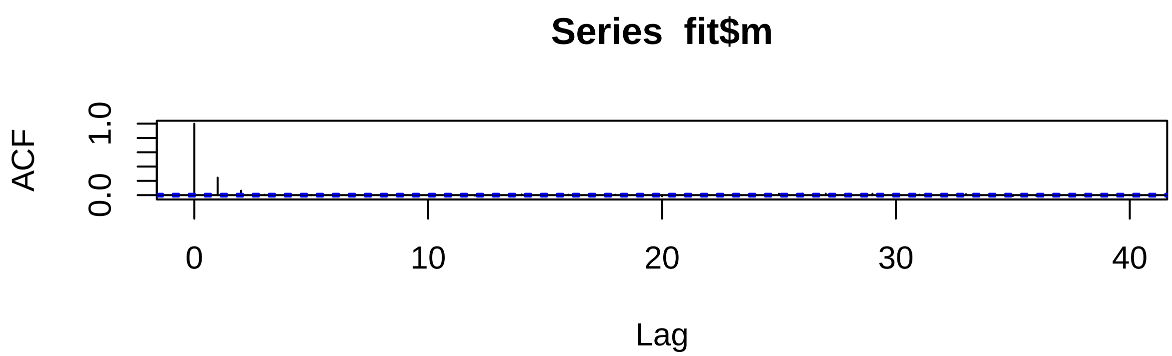
```
## t=7000, m=10
## t=7100, m=8
## t=7200, m=11
## t=7300, m=9
## t=7400, m=6
## t=7500, m=12
## t=7600, m=8
## t=7700, m=10
## t=7800, m=10
## t=7900, m=8
## t=8000, m=10
## t=8100, m=10
## t=8200, m=9
## t=8300, m=6
## t=8400, m=10
## t=8500, m=12
## t=8600, m=9
## t=8700, m=8
## t=8800, m=10
## t=8900, m=11
## t=9000, m=10
## t=9100, m=8
## t=9200, m=8
## t=9300, m=9
## t=9400, m=9
## t=9500, m=11
## t=9600, m=9
## t=9700, m=8
## t=9800, m=10
## t=9900, m=12
```

```
plot(fit, burnin=1000, which="m")
```

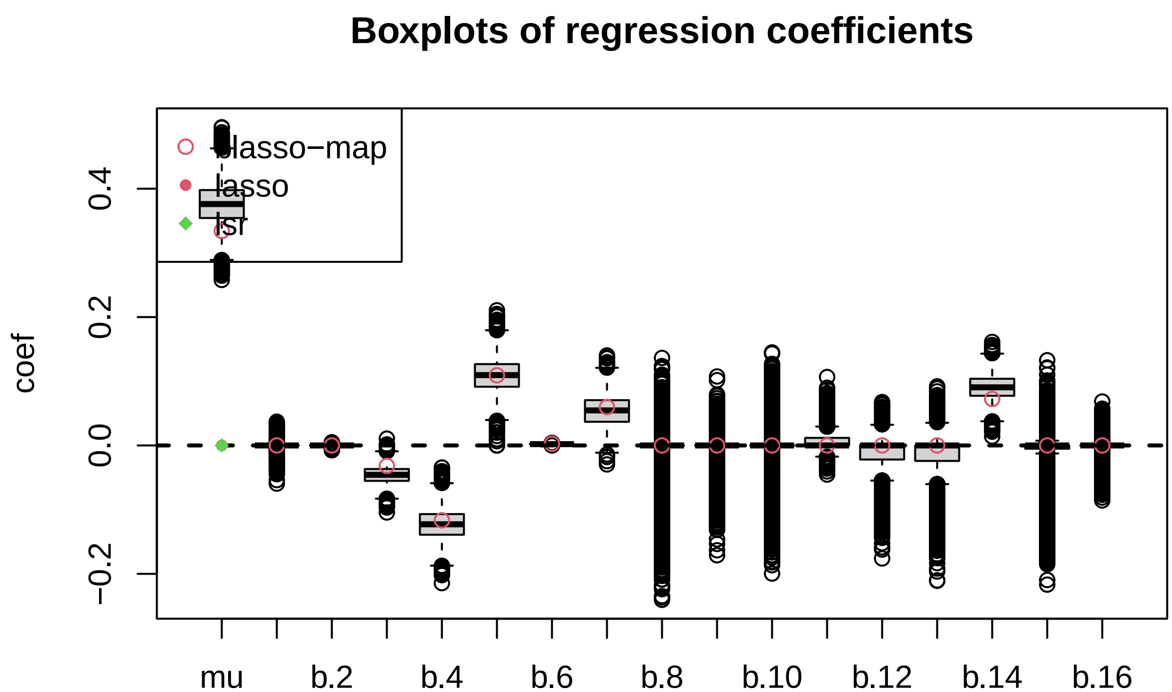





```
acf(fit$m)
```



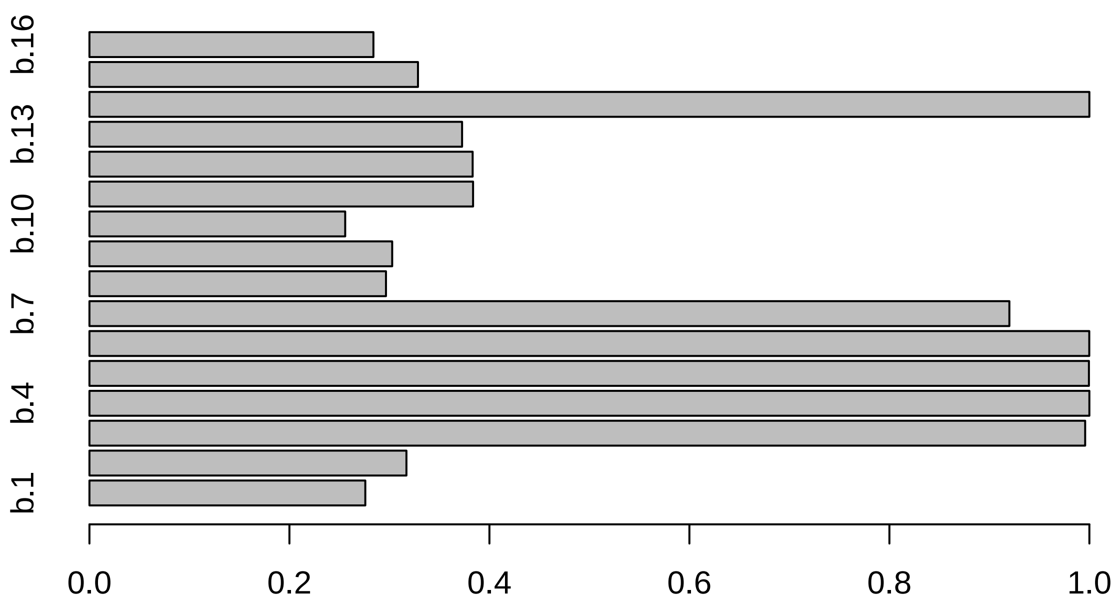

```
plot(fit, burnin=1000)
points(drop(fit$b), col=2, pch=20)
points(drop(fit$b), col=3, pch=18)
legend("topleft", c("blasso-map", "lasso", "lsr"),
       col=c(2,2,3), pch=c(21,20,18))
```




```
s <- summary(fit, burnin=1000)
print(s$bn0) # probability that each beta coef != zero
```

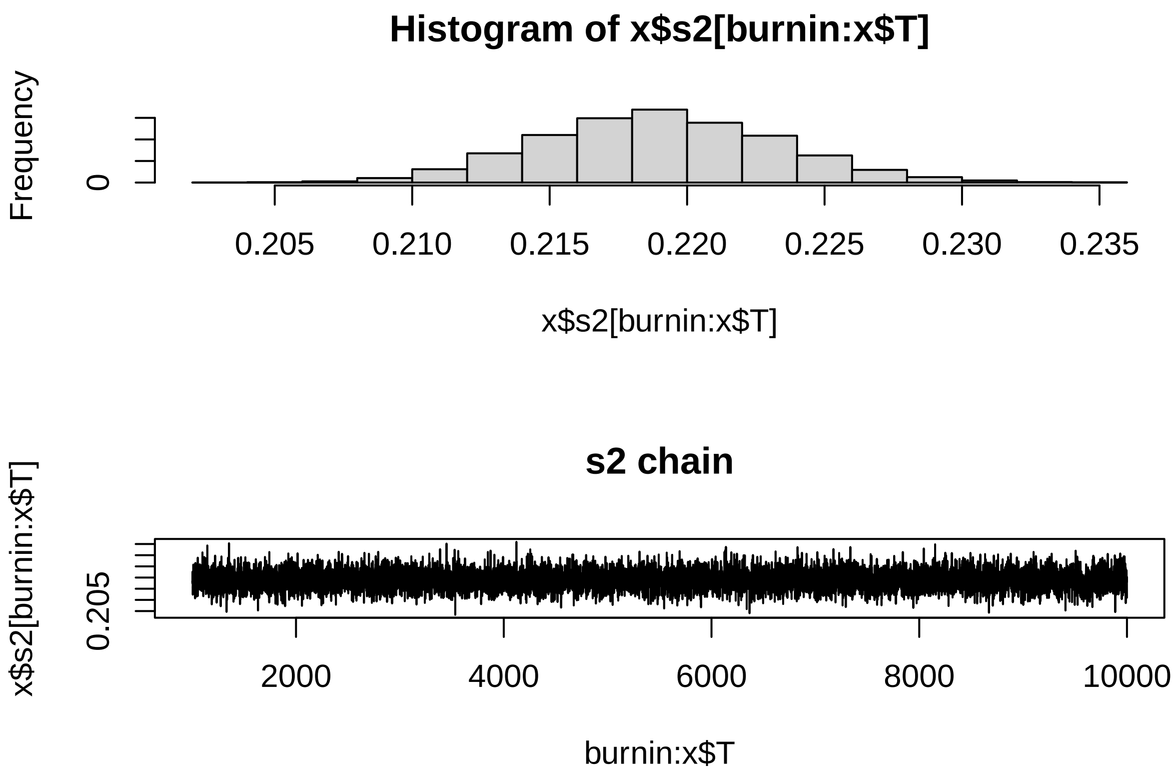
```
##       b.1       b.2       b.3       b.4       b.5       b.6       b.7       b.8
## 0.2758889 0.3170000 0.9957778 1.0000000 0.9995556 0.9998889 0.9200000 0.2965556
##       b.9      b.10      b.11      b.12      b.13      b.14      b.15      b.16
## 0.3027778 0.2557778 0.3836667 0.3832222 0.3726667 1.0000000 0.3285556 0.2840000
```

```
barplot(s$bn0, horiz = TRUE)
```

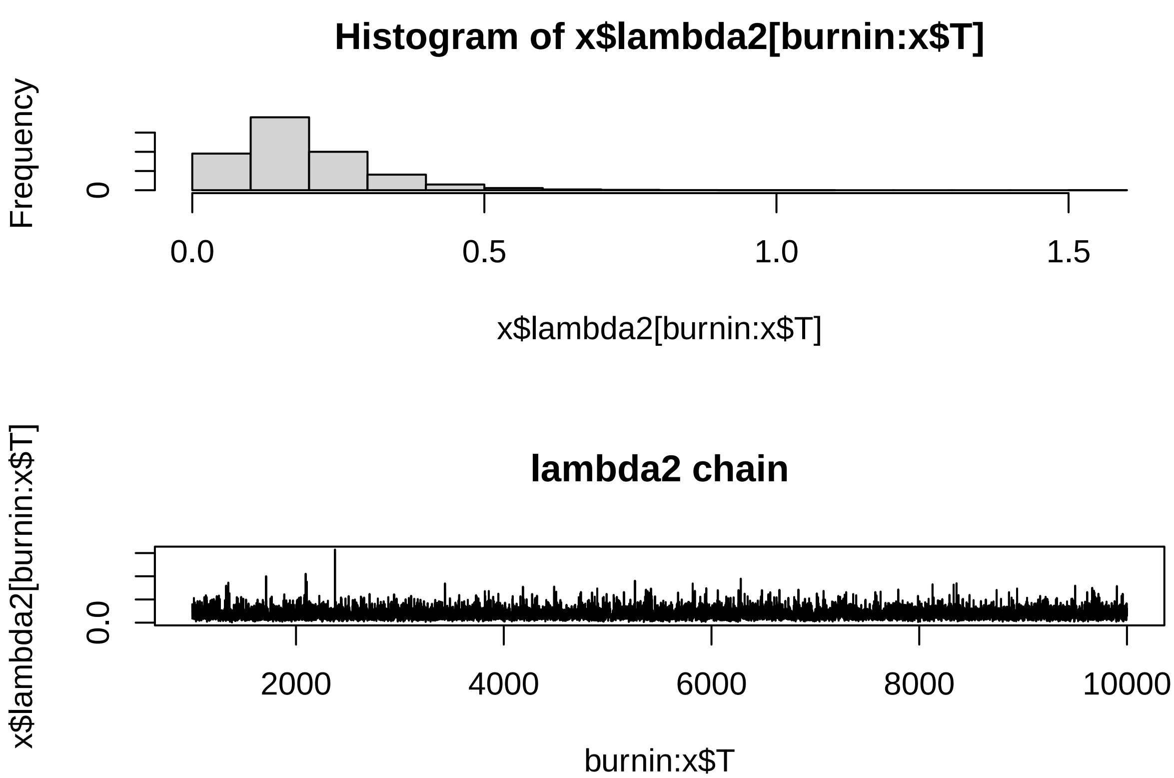


here we see that the

```r
plot(fit, burnin=1000, which="s2")
```




```r
plot(fit, burnin=1000, which="lambda2")
```



Final Model

```
rm(list = setdiff(ls(), c("data", "data.small", "data.test", "data.train")))

library(R2OpenBUGS)

model = function() {

}
```