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.

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

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

```r
  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.

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

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

```r
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 = c

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"), le

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))
```

```
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
```

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

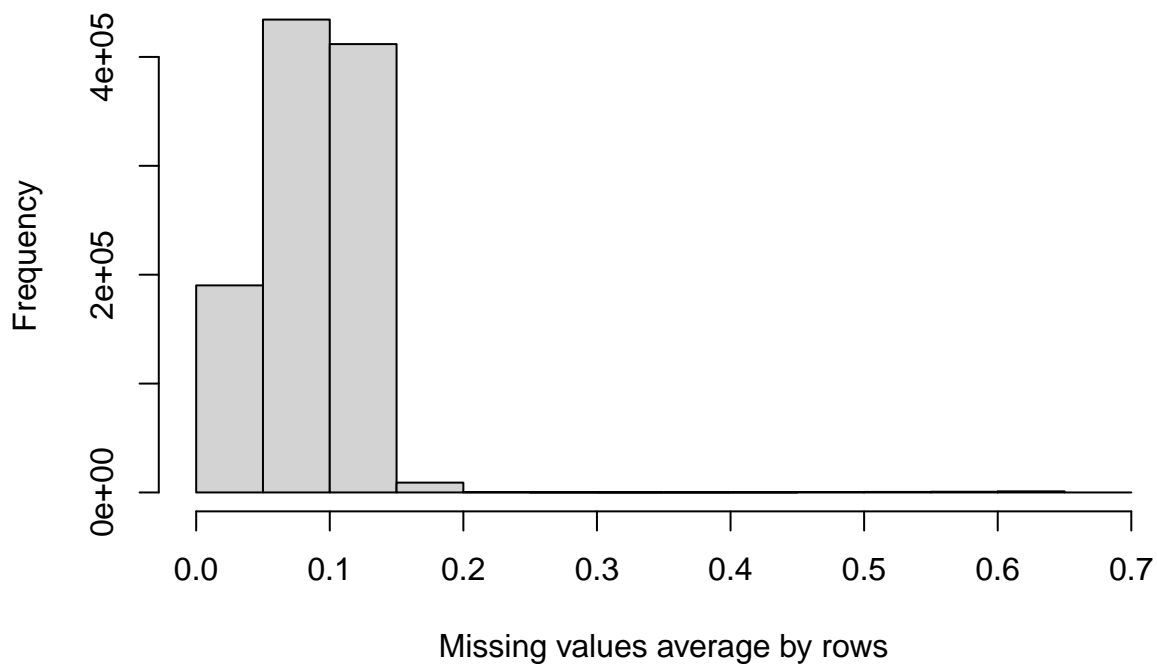Here we see that the data is correctly formated but there are some missing value, so let's fix that.

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())
```
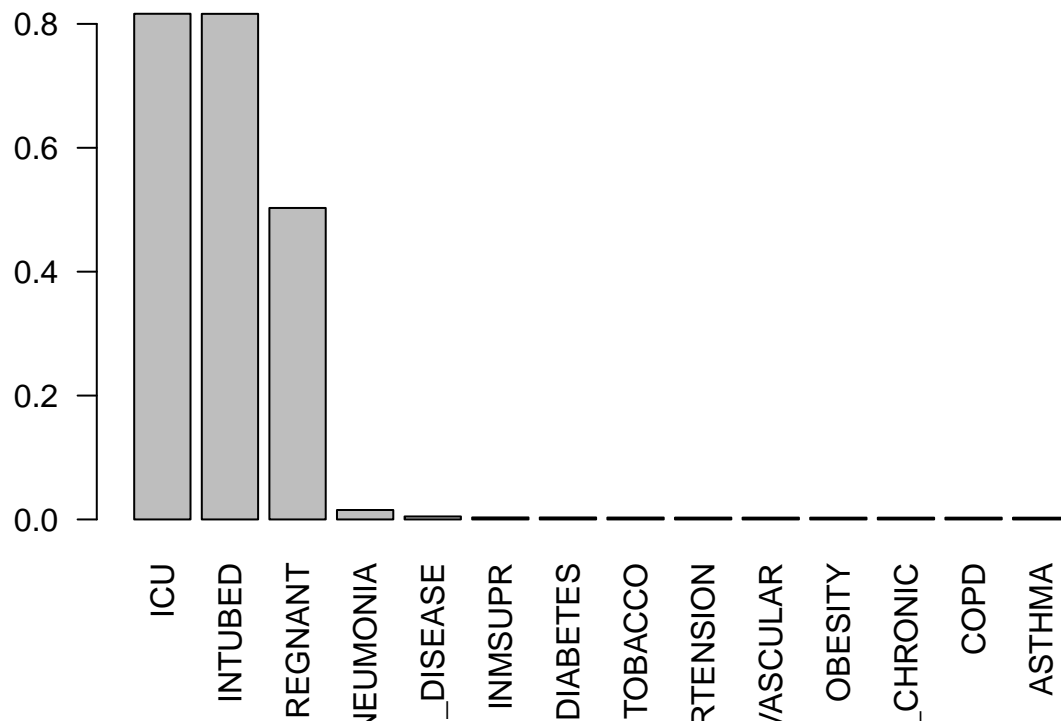

Here we see that there are 3 columns with the most missing values.

```
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.

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

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

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

```r
data = na.omit(data)

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

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

```r
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))
```

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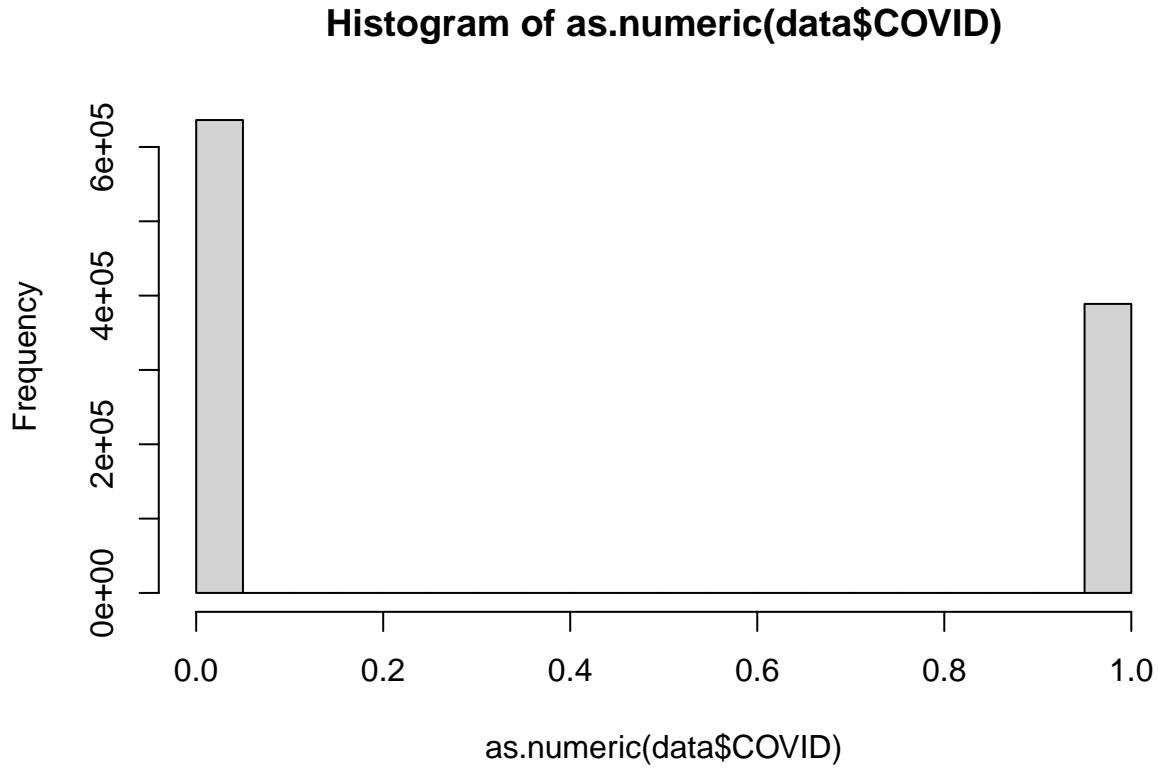
This is as we expected as we are going to be predicting a binary variable. So let's use a Bernoulli distribution to explain this data and see how well it fits. First of all lets compute the analytical posterior distribution of the covid variable.

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] 0.385
```

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] 0.435
```

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%
## 0.3551356 0.4145273
```

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

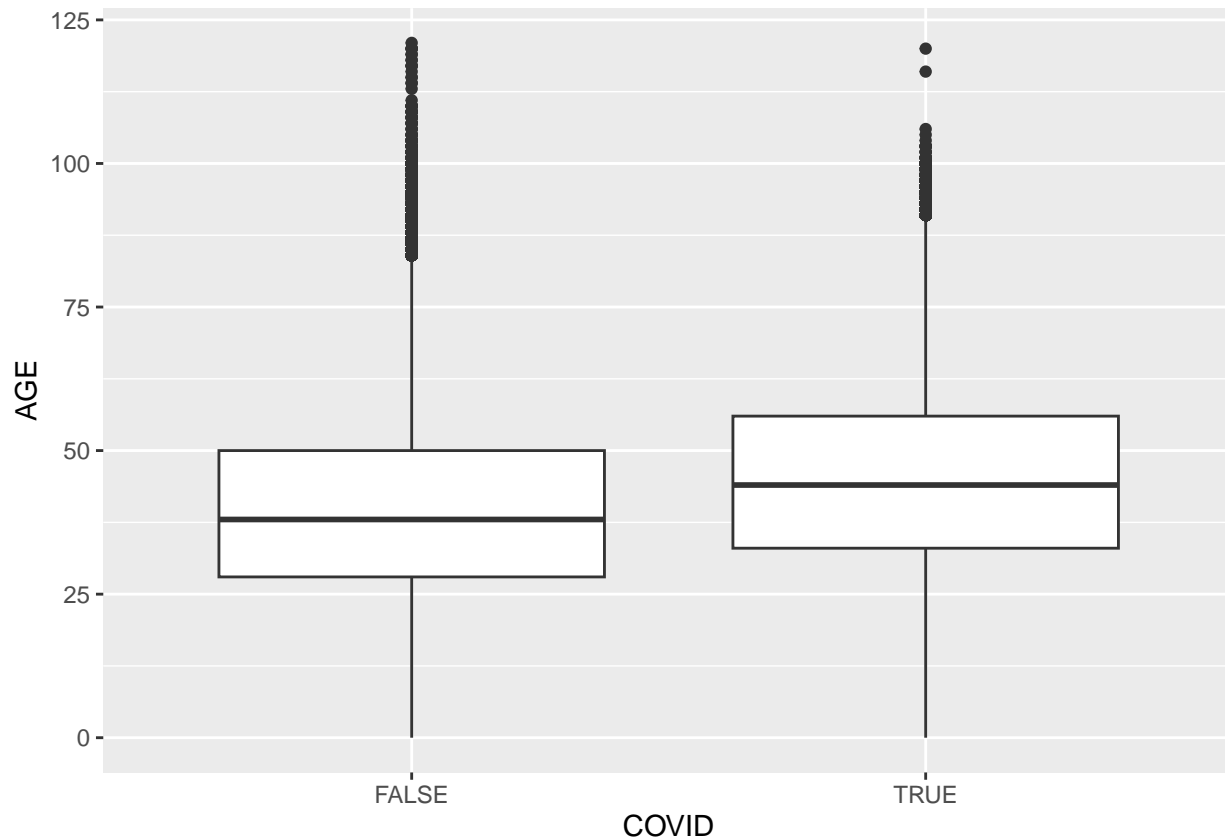Now, we will see if the other variables are useful to predict if a patient has covid or not.

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

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

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

Let's see how the columns distribute with respect to the covid variable.
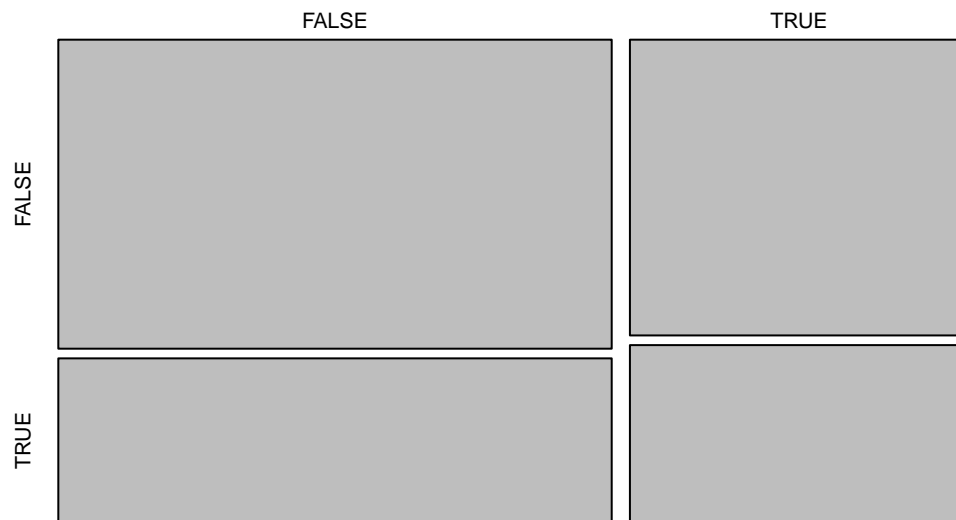
```
ggplot(data, aes(x=COVID, y=AGE)) +
  geom_boxplot()
```


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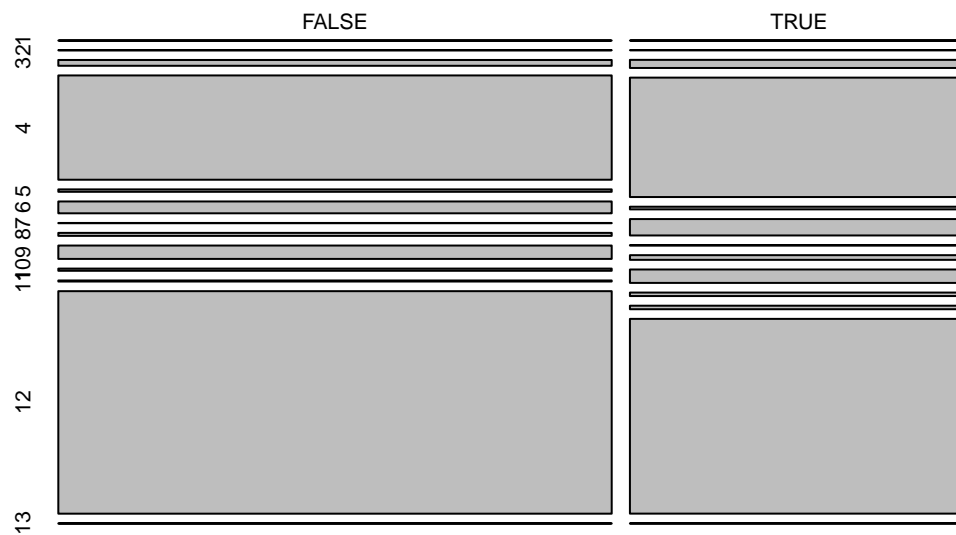
```
plot(table(data$COVID, data$USMER))
```
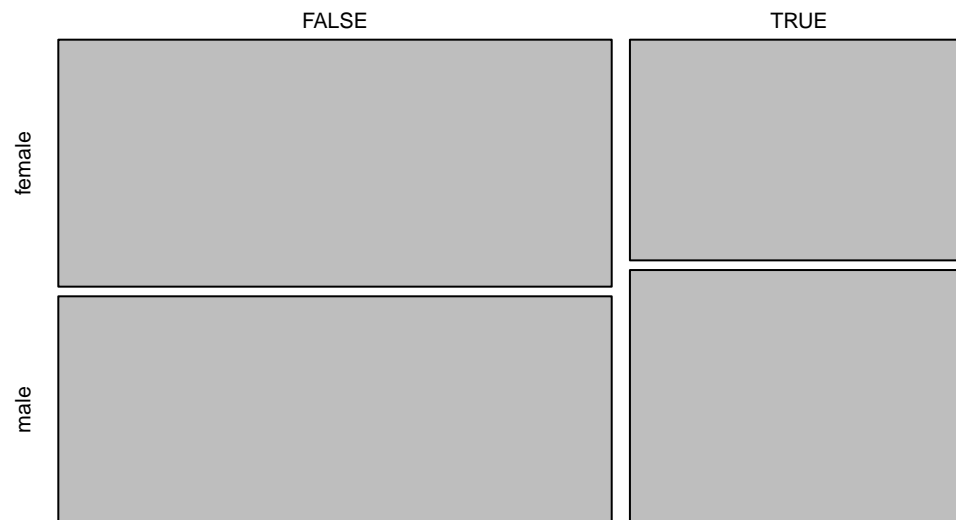
table(data$COVID, data$USMER)

	FALSE	TRUE


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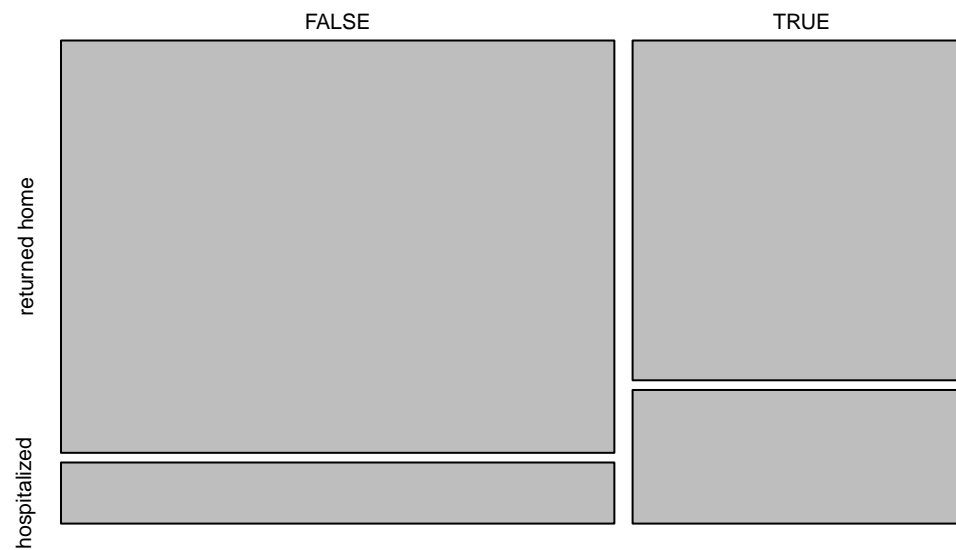
table(data$COVID, data$MEDICAL_UNIT)


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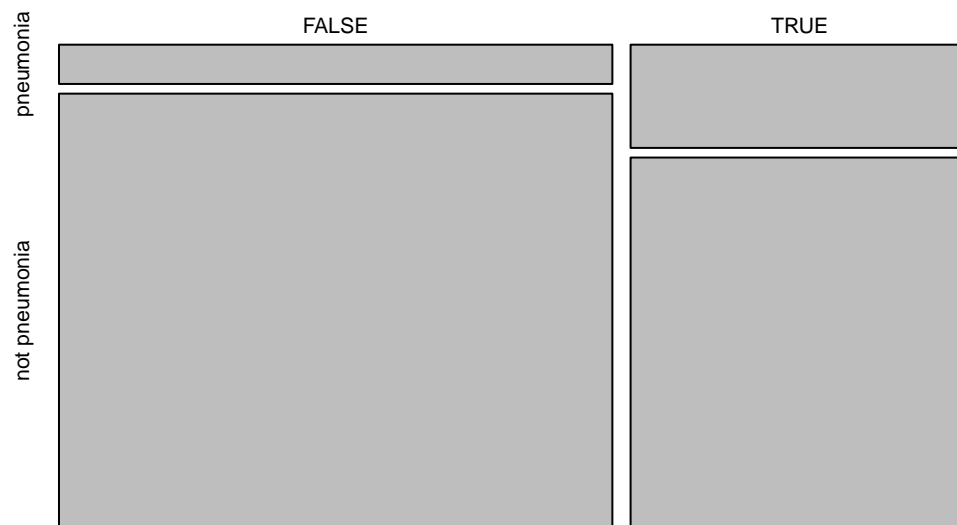
table(data$COVID, data$SEX)


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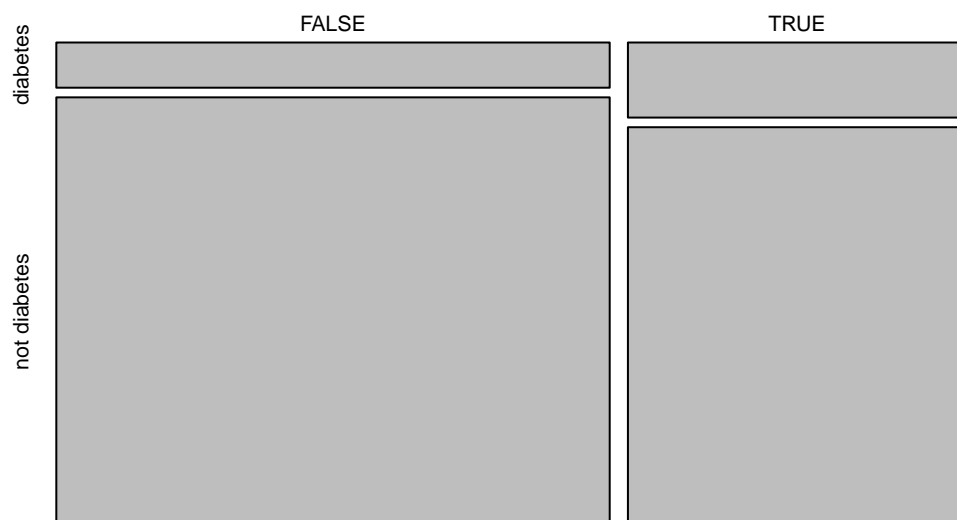
table(data$COVID, data$PATIENT_TYPE)


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

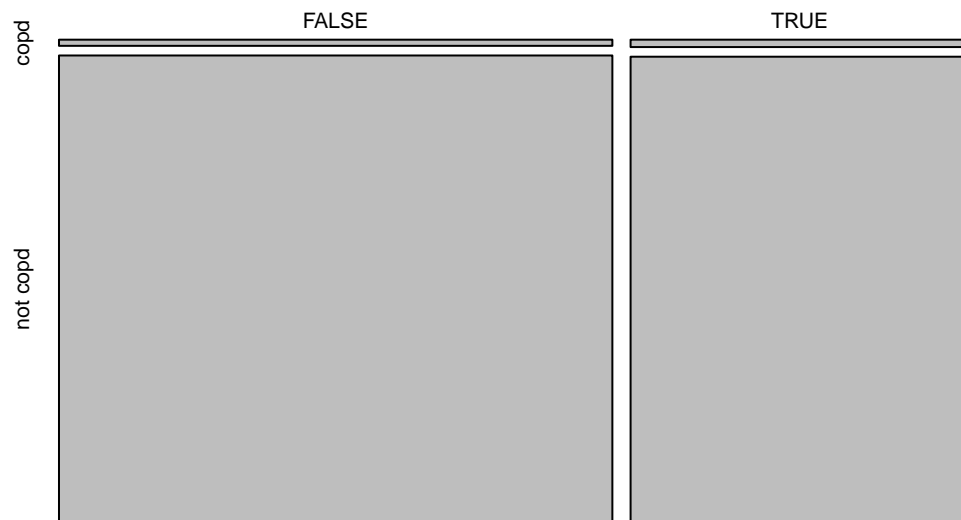# table(data$COVID, data$PNEUMONIA)


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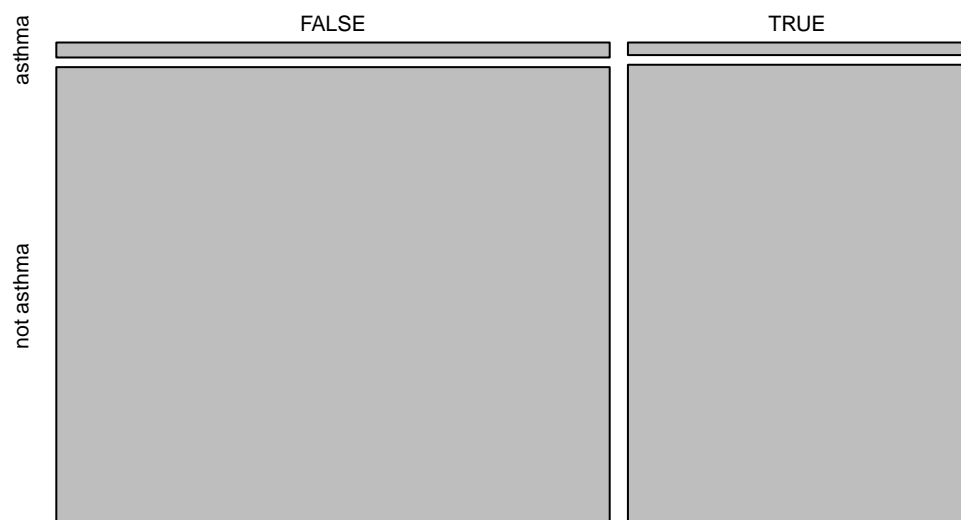
table(data$COVID, data$DIABETES)


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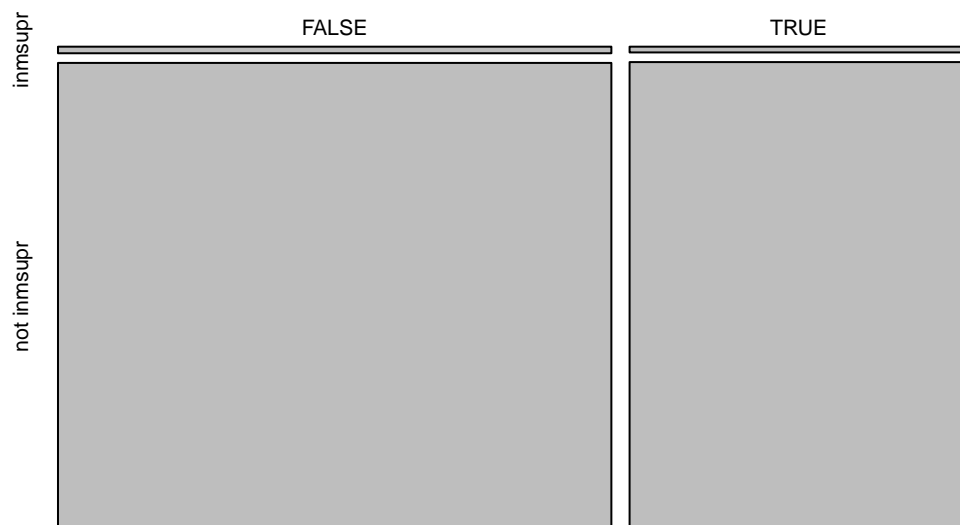
table(data$COVID, data$COPD)


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

table(data$COVID, data$ASTHMA)


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

table(data$COVID, data$INMSUPR)


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

table(data$COVID, data$HIPERTENSION)
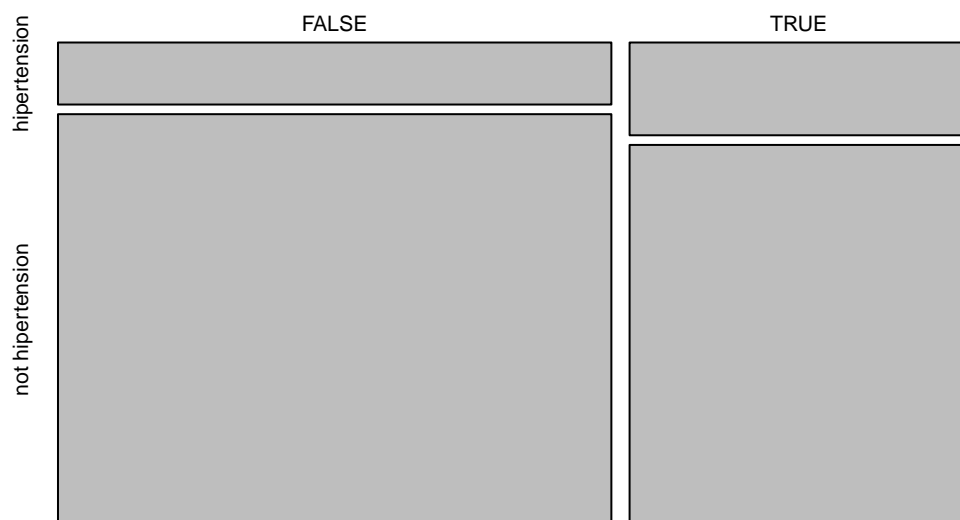

```
plot(table(data$COVID, data$OTHER_DISEASE))
```

table(data$COVID, data$OTHER_DISEASE)

other desease

FALSE	TRUE

not other desease
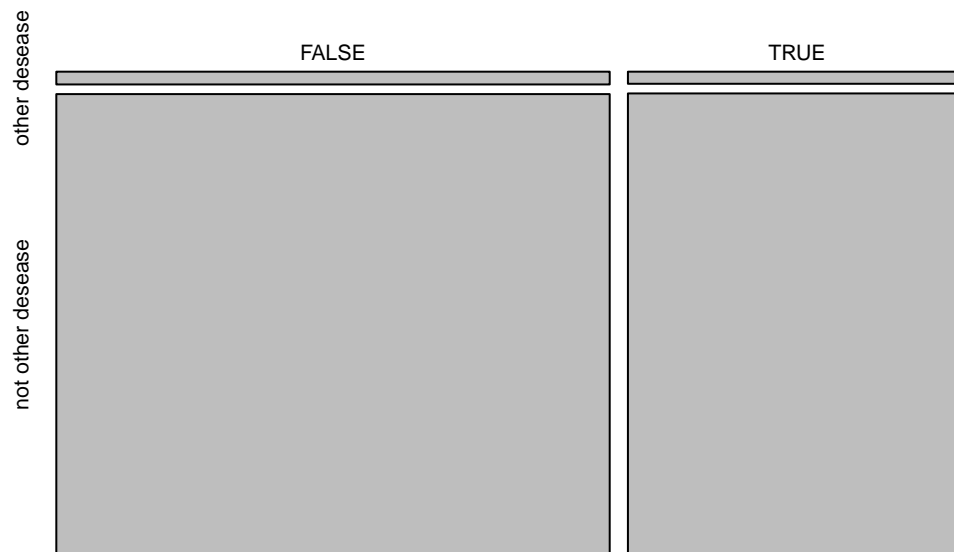
```
plot(table(data$COVID, data$CARDIOVASCULAR))
```

table(data$COVID, data$CARDIOVASCULAR)

cardiovascular

FALSE	TRUE

not cardiovascular

```
plot(table(data$COVID, data$OBESITY))
```
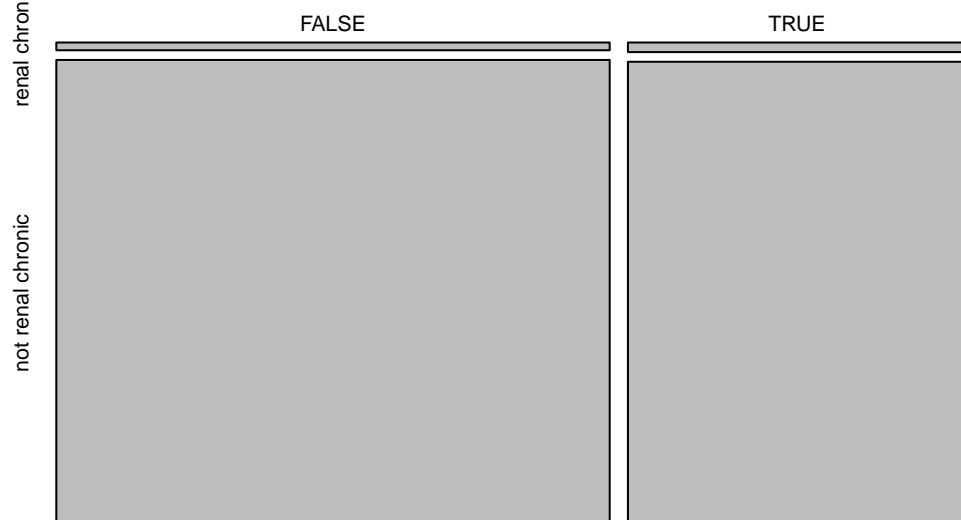
16
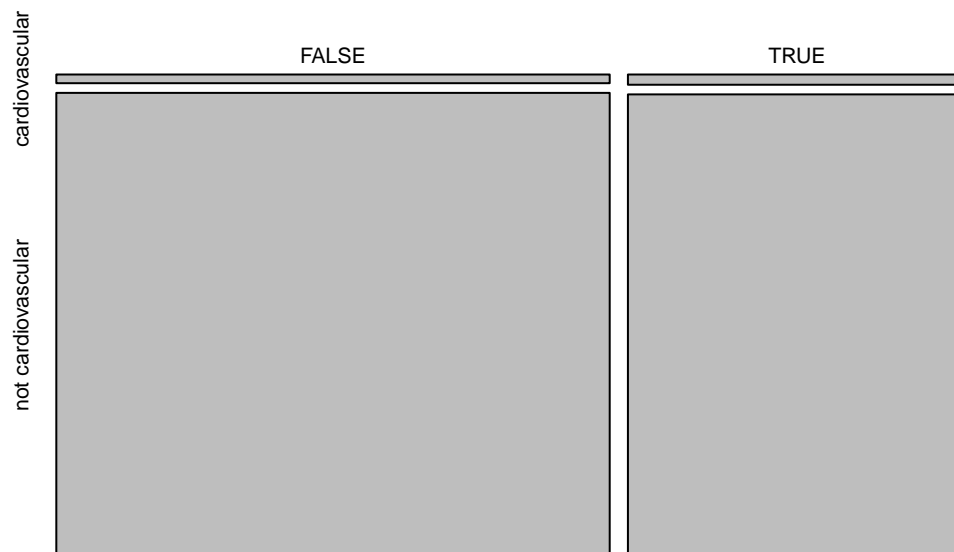
table(data$COVID, data$OBESITY)
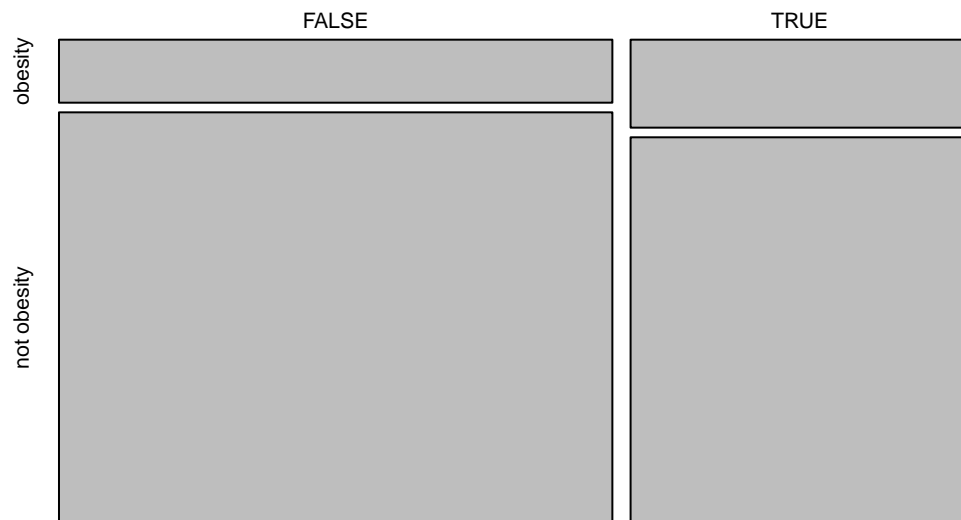

```
plot(table(data$COVID, data$RENAL_CHRONIC))
```

table(data$COVID, data$RENAL_CHRONIC)


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

table(data$COVID, data$TOBACCO)

Now, the columns that show off the most are: MEDICAL_UNIT, SEX, PATITENT_TYPE, and PNEUMO-NIA. 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
```

```
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.

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

summary(fit)
```

```
##
## Call:
## NULL
##
## Coefficients: (2 not defined because of singularities)
##                                     Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)                                -0.480154   1.138193  -0.422  0.67313
## USMERTRUE                                   -0.005030   0.065707  -0.077  0.93898
## MEDICAL_UNIT2                                      NA         NA      NA       NA
## MEDICAL_UNIT3                               -0.211763   1.044873  -0.203  0.83939
## MEDICAL_UNIT4                               -0.356177   1.026262  -0.347  0.72854
## MEDICAL_UNIT5                               -0.383722   1.079604  -0.355  0.72227
## MEDICAL_UNIT6                               -0.597059   1.037737  -0.575  0.56506
## MEDICAL_UNIT7                              -12.453065 228.461441  -0.055  0.95653
## MEDICAL_UNIT8                               -0.012526   1.075717  -0.012  0.99071
## MEDICAL_UNIT9                               -0.685330   1.039360  -0.659  0.50965
## MEDICAL_UNIT10                              -0.100638   1.074661  -0.094  0.92539
## MEDICAL_UNIT11                               0.695247   1.103862   0.630  0.52880
## MEDICAL_UNIT12                              -0.468333   1.025802  -0.457  0.64799
## MEDICAL_UNIT13                                     NA         NA      NA       NA
## SEXmale                                      0.189070   0.060842   3.108  0.00189 **
## PATIENT_TYPEhospitalized                     0.415423   0.100113   4.150 3.33e-05 ***
## `PNEUMONIAnot pneumonia`                    -0.628042   0.110615  -5.678 1.36e-08 ***
## AGE                                          0.009659   0.002056   4.699 2.62e-06 ***
## `DIABETESnot diabetes`                      -0.061639   0.100342  -0.614  0.53902
## `COPDnot copd`                               0.429007   0.280563   1.529  0.12624
## `ASTHMAnot asthma`                          -0.062139   0.177057  -0.351  0.72562
## `INMSUPRnot inmsupr`                         0.573871   0.253999   2.259  0.02386 *
## `HIPERTENSIONnot hipertension`              -0.123096   0.093500  -1.317  0.18799
## `OTHER_DISEASEnot other desease`            -0.142194   0.170069  -0.836  0.40310
## `CARDIOVASCULARnot cardiovascular`          -0.002049   0.213432  -0.010  0.99234
## `OBESITYnot obesity`                        -0.267177   0.083974  -3.182  0.00146 **
## `RENAL_CHRONICnot renal chronic`            -0.014886   0.210786  -0.071  0.94370
## `TOBACCOnot tobacco`                        -0.024984   0.112226  -0.223  0.82383
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6644.6  on 4998  degrees of freedom
## Residual deviance: 6346.9  on 4973  degrees of freedom
## AIC: 6398.9
##
## Number of Fisher Scoring iterations: 11
```

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  2820  275
##      TRUE   1422  484
##
##               Accuracy : 0.6607
##                 95% CI : (0.6474, 0.6738)
##     No Information Rate : 0.8482
##     P-Value [Acc > NIR] : 1
```

```
##
##                 Kappa : 0.1867
##
##   Mcnemar's Test P-Value : <2e-16
##
##           Sensitivity : 0.6648
##           Specificity : 0.6377
##        Pos Pred Value : 0.9111
##        Neg Pred Value : 0.2539
##            Prevalence : 0.8482
##        Detection Rate : 0.5639
##  Detection Prevalence : 0.6189
##      Balanced Accuracy : 0.6512
##
##        '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 + PATIEI

summary(fit)
```

```
##
## Call:
## NULL
##
## Coefficients: (2 not defined because of singularities)
##                                Estimate Std. Error z value Pr(>|z|)
## (Intercept)                     0.212780    1.039192    0.205    0.838
## USMERTRUE                      -0.002279    0.065414   -0.035    0.972
## `PNEUMONIAnot pneumonia`       -0.640955    0.109895   -5.832 5.46e-09 ***
## MEDICAL_UNIT2                         NA          NA       NA       NA
## MEDICAL_UNIT3                  -0.240153    1.038209   -0.231    0.817
## MEDICAL_UNIT4                  -0.371058    1.019501   -0.364    0.716
## MEDICAL_UNIT5                  -0.435579    1.073237   -0.406    0.685
## MEDICAL_UNIT6                  -0.615138    1.031161   -0.597    0.551
## MEDICAL_UNIT7                 -12.346076  228.299334   -0.054    0.957
## MEDICAL_UNIT8                   0.058285    1.068378    0.055    0.956
## MEDICAL_UNIT9                  -0.714162    1.032705   -0.692    0.489
## MEDICAL_UNIT10                 -0.128067    1.068324   -0.120    0.905
## MEDICAL_UNIT11                  0.620423    1.097352    0.565    0.572
## MEDICAL_UNIT12                 -0.486438    1.019184   -0.477    0.633
## MEDICAL_UNIT13                        NA          NA       NA       NA
## `DIABETESnot diabetes`         -0.072495    0.099186   -0.731    0.465
## `HIPERTENSIONnot hipertension` -0.140312    0.091580   -1.532    0.125
## AGE                             0.009169    0.002033    4.511 6.46e-06 ***
## PATIENT_TYPEhospitalized        0.417164    0.098949    4.216 2.49e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 6644.6  on 4998  degrees of freedom
```

```
## Residual deviance: 6375.2  on 4982  degrees of freedom
## AIC: 6409.2
##
## Number of Fisher Scoring iterations: 11
```

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  2809  286
##      TRUE   1432  474
##
##                Accuracy : 0.6565
##                  95% CI : (0.6431, 0.6696)
##     No Information Rate : 0.848
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.1767
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.6623
##             Specificity : 0.6237
##          Pos Pred Value : 0.9076
##          Neg Pred Value : 0.2487
##              Prevalence : 0.8480
##          Detection Rate : 0.5617
##    Detection Prevalence : 0.6189
##       Balanced Accuracy : 0.6430
##
##        '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.
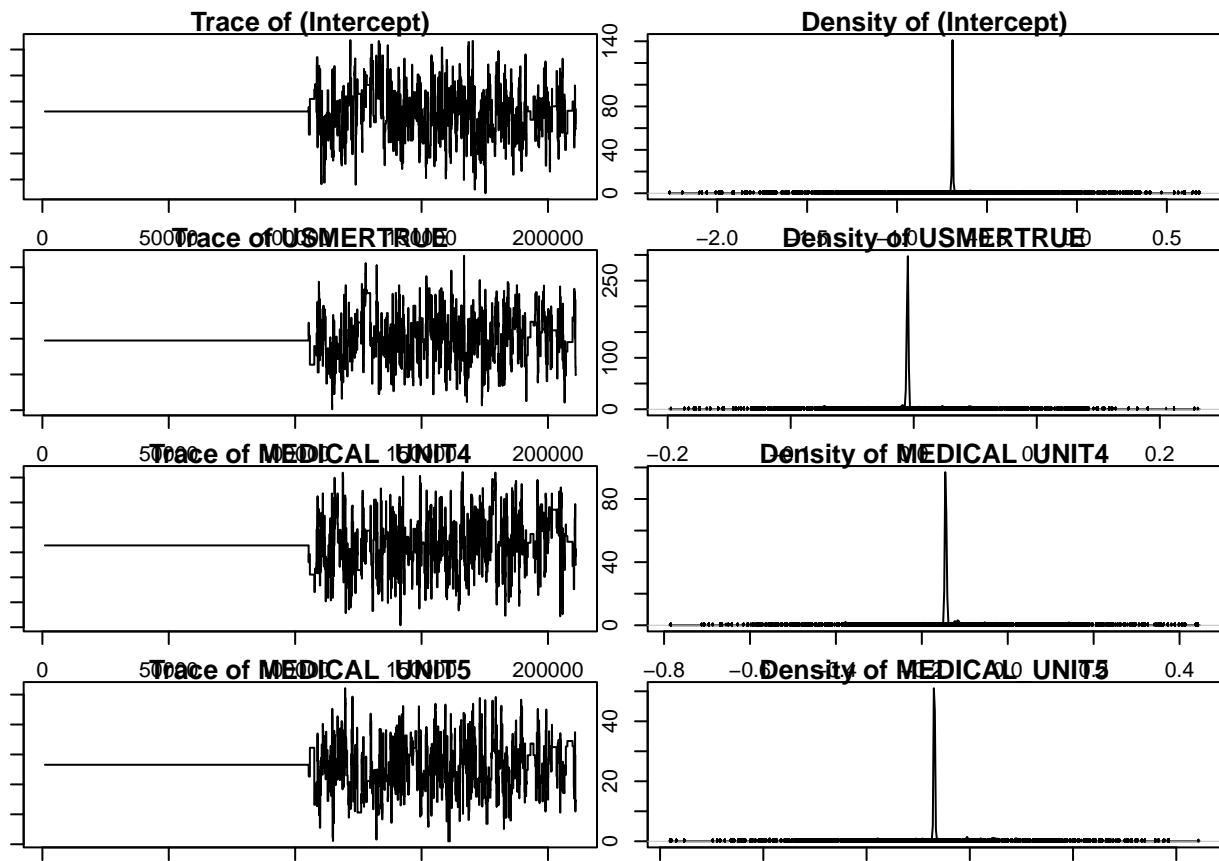
Bayesian LM

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.
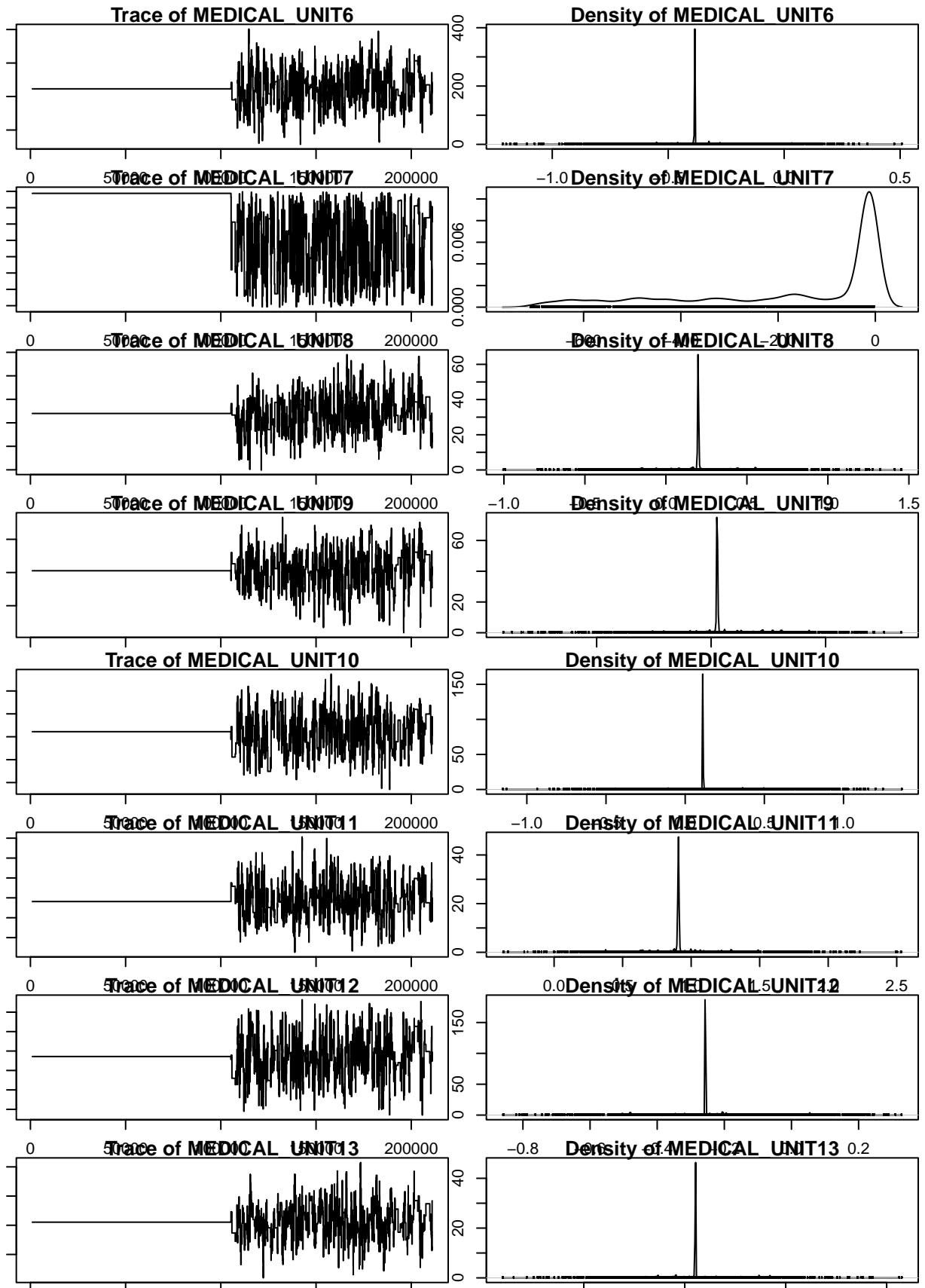
```
library(coda)
library(MASS)
library(MCMCpack)
```
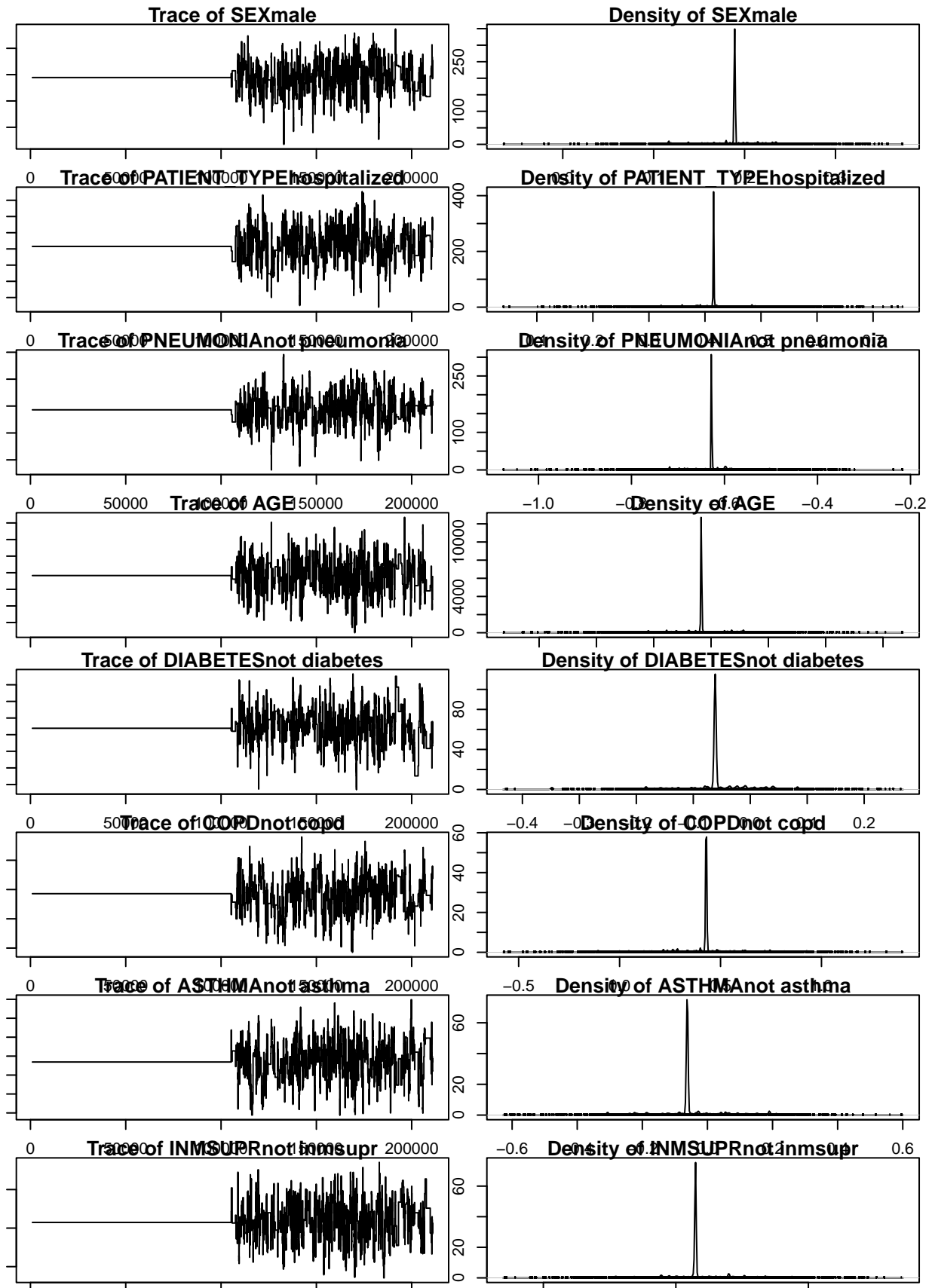
```
rm(list = setdiff(ls(), c("data", "data.small", "data.test", "data.train")))
```
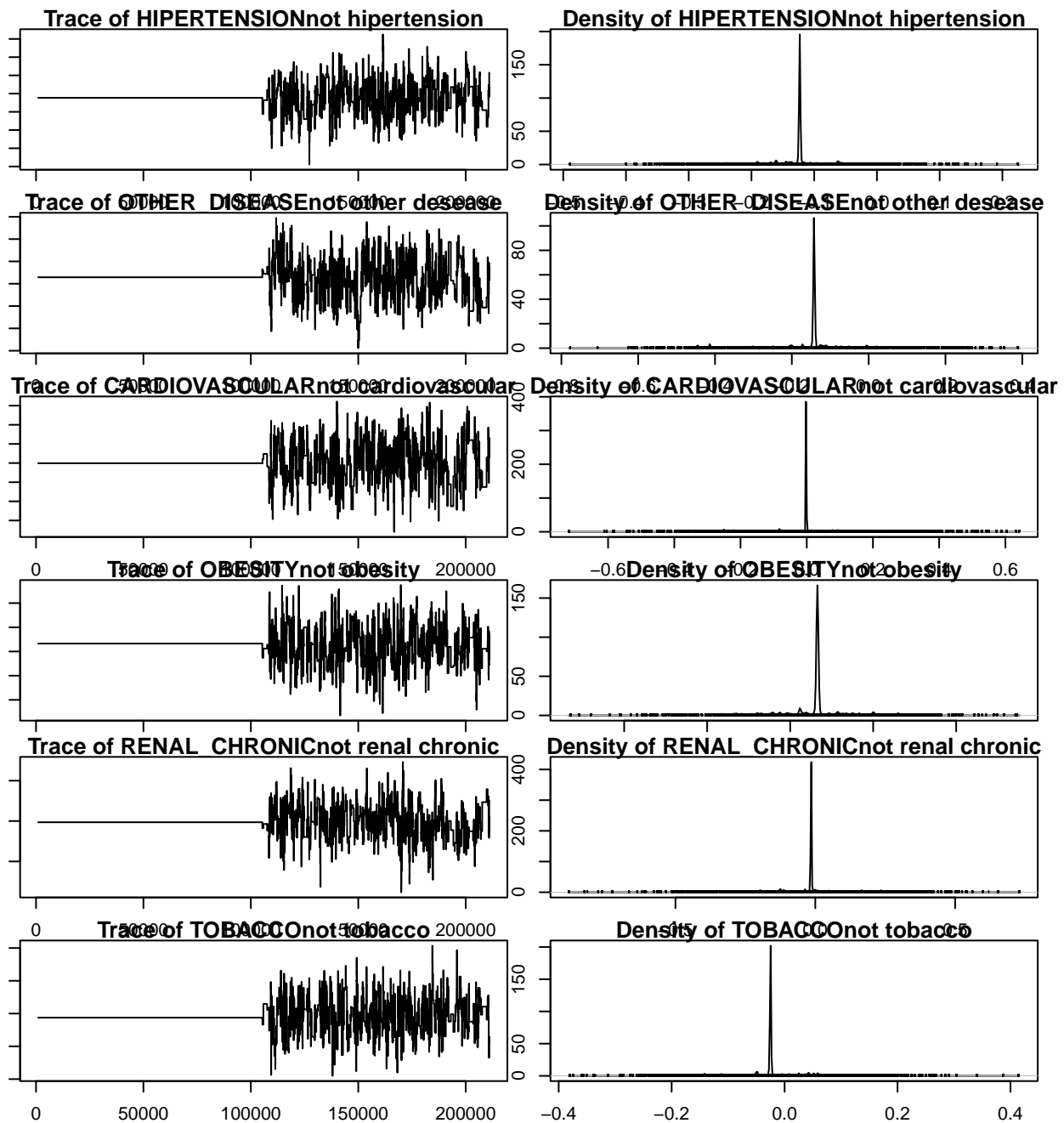
```
fit = MCMClogit(COVID ~ ., data = data.train, burnin=1000, mcmc=210000)
```

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

Trace of MEDICAL_UNIT6

Density of MEDICAL_UNIT6

Trace of MEDICAL_UNIT10

Density of MEDICAL_UNIT10

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

24

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

Trace of OBESITYnot obesity

Density of OBESITYnot obesity

Trace of RENAL_CHRONICnot renal chronic

Density of RENAL_CHRONICnot renal chronic

Trace of TOBACCOnot tobacco

Density of TOBACCOnot tobacco

```
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
```

25

```
## (Intercept)                          -6.998e-01 3.507e-01 7.652e-04    1.582e-02
## USMERTRUE                            -2.014e-03 4.888e-02 1.067e-04    2.415e-03
## MEDICAL_UNIT4                        -1.401e-01 1.520e-01 3.316e-04    7.147e-03
## MEDICAL_UNIT5                        -1.908e-01 3.086e-01 6.734e-04    1.594e-02
## MEDICAL_UNIT6                        -3.831e-01 1.842e-01 4.019e-04    9.073e-03
## MEDICAL_UNIT7                        -1.718e+02 2.116e+02 4.618e-01    1.614e+01
## MEDICAL_UNIT8                         1.917e-01 2.671e-01 5.829e-04    1.258e-02
## MEDICAL_UNIT9                        -4.694e-01 1.975e-01 4.309e-04    9.614e-03
## MEDICAL_UNIT10                        1.046e-01 3.017e-01 6.584e-04    1.614e-02
## MEDICAL_UNIT11                        9.325e-01 3.048e-01 6.650e-04    1.418e-02
## MEDICAL_UNIT12                       -2.571e-01 1.532e-01 3.344e-04    7.306e-03
## MEDICAL_UNIT13                        2.185e-01 7.839e-01 1.711e-03    3.979e-02
## SEXmale                              1.867e-01 4.212e-02 9.192e-05    1.872e-03
## PATIENT_TYPEhospitalized             4.202e-01 7.253e-02 1.583e-04    3.485e-03
## PNEUMONIAnot pneumonia               -6.254e-01 7.648e-02 1.669e-04    3.583e-03
## AGE                                  9.627e-03 1.452e-03 3.168e-06    6.652e-05
## DIABETESnot diabetes                 -5.706e-02 7.986e-02 1.743e-04    4.401e-03
## COPDnot copd                         4.137e-01 2.140e-01 4.671e-04    1.092e-02
## ASTHMAnot asthma                     -5.204e-02 1.399e-01 3.052e-04    7.199e-03
## INMSUPRnot inmsupr                   5.896e-01 1.794e-01 3.915e-04    8.445e-03
## HIPERTENSIONnot hipertension         -1.247e-01 6.771e-02 1.477e-04    3.301e-03
## OTHER_DISEASEnot other desease       -1.537e-01 1.334e-01 2.911e-04    6.837e-03
## CARDIOVASCULARnot cardiovascular     -5.646e-04 1.553e-01 3.388e-04    7.378e-03
## OBESITYnot obesity                   -2.727e-01 5.751e-02 1.255e-04    2.696e-03
## RENAL_CHRONICnot renal chronic       -1.491e-02 1.508e-01 3.291e-04    7.544e-03
## TOBACCOnot tobacco                   -1.963e-02 7.927e-02 1.730e-04    3.538e-03
##
## 2. Quantiles for each variable:
##
##                                          2.5%       25%        50%        75%
## (Intercept)                          -1.513e+00 -7.097e-01  -0.691917  -0.691917
## USMERTRUE                            -1.140e-01 -5.030e-03  -0.005030   0.002826
## MEDICAL_UNIT4                        -4.895e-01 -1.444e-01  -0.144414  -0.123138
## MEDICAL_UNIT5                        -9.499e-01 -2.056e-01  -0.171959  -0.171959
## MEDICAL_UNIT6                        -8.155e-01 -3.922e-01  -0.385296  -0.385296
## MEDICAL_UNIT7                        -6.527e+02 -3.224e+02 -12.241302 -12.241302
## MEDICAL_UNIT8                        -4.164e-01  1.662e-01   0.199237   0.199237
## MEDICAL_UNIT9                        -9.487e-01 -4.736e-01  -0.473567  -0.451139
## MEDICAL_UNIT10                       -5.816e-01  1.111e-01   0.111125   0.127023
## MEDICAL_UNIT11                        2.551e-01  9.070e-01   0.907010   0.950678
## MEDICAL_UNIT12                       -6.073e-01 -2.566e-01  -0.256570  -0.246485
## MEDICAL_UNIT13                       -1.641e+00  1.638e-01   0.211763   0.211763
## SEXmale                              8.943e-02  1.842e-01   0.189070   0.189070
## PATIENT_TYPEhospitalized             2.495e-01  4.154e-01   0.415423   0.421572
## PNEUMONIAnot pneumonia               -7.964e-01 -6.280e-01  -0.628042  -0.620672
## AGE                                  6.225e-03  9.492e-03   0.009659   0.009659
## DIABETESnot diabetes                 -2.261e-01 -6.164e-02  -0.061639  -0.039939
## COPDnot copd                         -1.056e-01  3.998e-01   0.429007   0.429007
## ASTHMAnot asthma                     -3.494e-01 -6.214e-02  -0.062139  -0.034258
## INMSUPRnot inmsupr                   1.962e-01  5.739e-01   0.573871   0.608353
## HIPERTENSIONnot hipertension         -2.826e-01 -1.358e-01  -0.123096  -0.123096
## OTHER_DISEASEnot other desease       -4.616e-01 -1.644e-01  -0.142194  -0.142194
## CARDIOVASCULARnot cardiovascular     -3.526e-01 -2.196e-03  -0.002049   0.004678
## OBESITYnot obesity                   -4.138e-01 -2.808e-01  -0.267177  -0.267177
```

```
## RENAL_CHRONICnot renal chronic   -3.498e-01 -1.489e-02   -0.014886   -0.009515
## TOBACCOnot tobacco                -1.971e-01 -2.498e-02   -0.024984   -0.013233
##                                        97.5%
## (Intercept)                          0.11840
## USMERTRUE                            0.12475
## MEDICAL_UNIT4                        0.22249
## MEDICAL_UNIT5                        0.45819
## MEDICAL_UNIT6                        0.05562
## MEDICAL_UNIT7                      -12.24130
## MEDICAL_UNIT8                        0.82512
## MEDICAL_UNIT9                       -0.05104
## MEDICAL_UNIT10                       0.77907
## MEDICAL_UNIT11                       1.68145
## MEDICAL_UNIT12                       0.11564
## MEDICAL_UNIT13                       1.98000
## SEXmale                              0.27739
## PATIENT_TYPEhospitalized             0.58735
## PNEUMONIAnot pneumonia              -0.42738
## AGE                                  0.01281
## DIABETESnot diabetes                 0.12814
## COPDnot copd                         0.90239
## ASTHMAnot asthma                     0.27950
## INMSUPRnot inmsupr                   1.01206
## HIPERTENSIONnot hipertension         0.03583
## OTHER_DISEASEnot other desease       0.16216
## CARDIOVASCULARnot cardiovascular     0.36474
## OBESITYnot obesity                  -0.14612
## RENAL_CHRONICnot renal chronic       0.34865
## TOBACCOnot tobacco                   0.16941
```

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)

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)
 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.   : 3.000   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 :1.0000   Median :1.0000
##  Mean   :0.3493   Mean   : 8.901   Mean   :0.5049   Mean   :0.8088
##  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 :41.00    Median :0.0000   Median :0.0000
##  Mean   :0.1362   Mean   :42.19    Mean   :0.1236   Mean   :0.0124
##  3rd Qu.:0.0000   3rd Qu.:53.00    3rd Qu.:0.0000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :99.00    Max.   :1.0000   Max.   :1.0000
##     ASTHMA            INMSUPR          HIPERTENSION      OTHER_DISEASE
##  Min.   :0.00000   Min.   :0.000    Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.000    1st Qu.:0.0000   1st Qu.:0.00000
##  Median :0.00000   Median :0.000    Median :0.0000   Median :0.00000
##  Mean   :0.03041   Mean   :0.016    Mean   :0.1616   Mean   :0.03361
##  3rd Qu.:0.00000   3rd Qu.:0.000    3rd Qu.:0.0000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :1.000    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   :0.0212   Mean   :0.153    Mean   :0.0222   Mean   :0.07942
##  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
```

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

```
## t=100, m=7
## t=200, m=7
## t=300, m=13
## t=400, m=9
## t=500, m=7
## t=600, m=8
## t=700, m=10
## t=800, m=9
## t=900, m=7
```

After training the model lets check for stability

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

Histogram of x$m[burnin:x$T]

m chain


```
acf(fit$m)
```
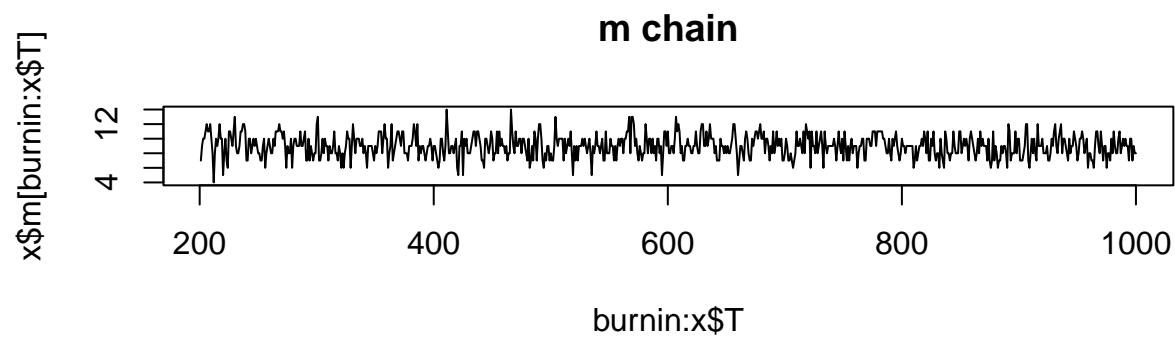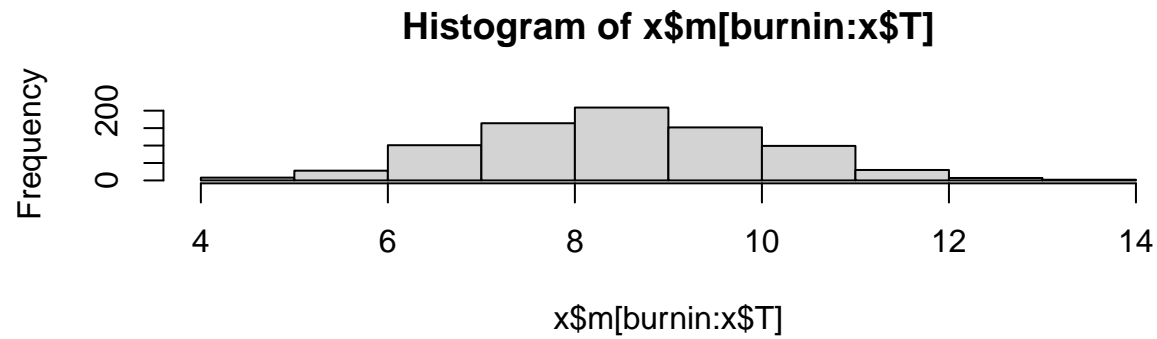
Series fit$m
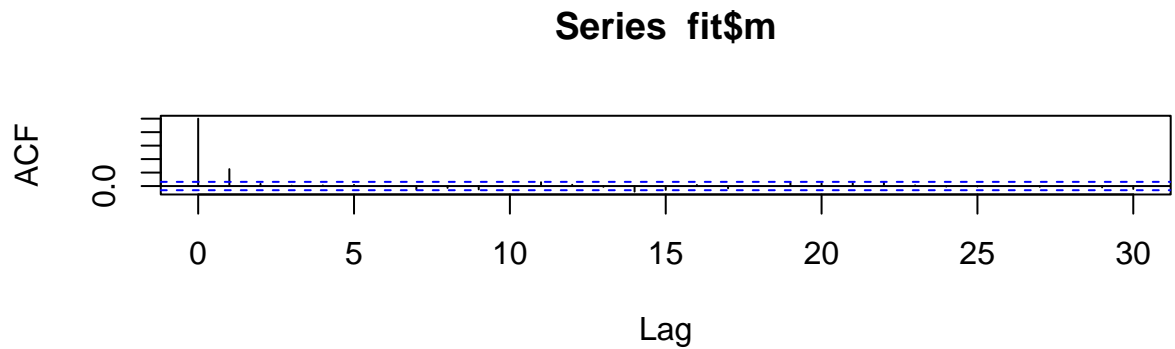
Here we see that there is some autocorrelation and it is not stable enough so to make sure let's add a lot
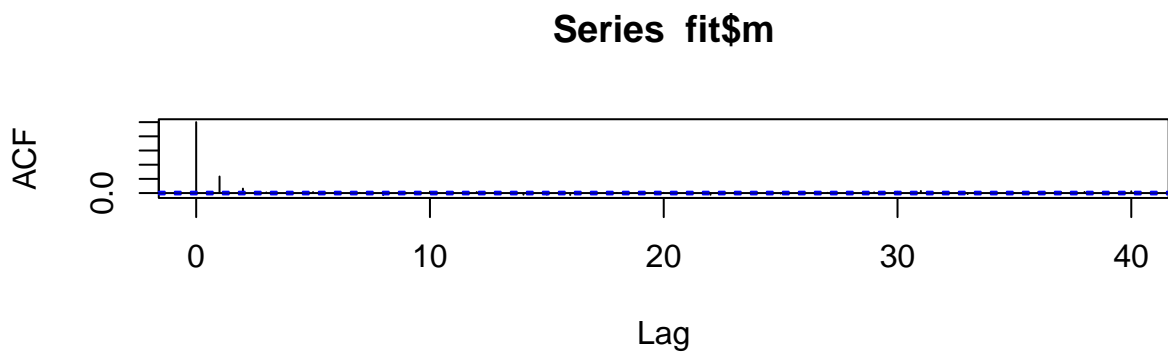more samples and some thinning.

```
set.seed(111)
fit = blasso(x, y, mprior = c(0,1), T = 10000, thin = 20)

## t=100, m=9
## t=200, m=10
## t=300, m=7
## t=400, m=11
## t=500, m=8
## t=600, m=10
## t=700, m=9
## t=800, m=6
## t=900, m=9
```

29

```
## t=1000, m=8
## t=1100, m=6
## t=1200, m=7
## t=1300, m=10
## t=1400, m=10
## t=1500, m=9
## t=1600, m=9
## t=1700, m=8
## t=1800, m=11
## t=1900, m=10
## t=2000, m=8
## t=2100, m=8
## t=2200, m=8
## t=2300, m=9
## t=2400, m=11
## t=2500, m=10
## t=2600, m=10
## t=2700, m=9
## t=2800, m=11
## t=2900, m=10
## t=3000, m=10
## t=3100, m=8
## t=3200, m=10
## t=3300, m=7
## t=3400, m=8
## t=3500, m=7
## t=3600, m=9
## t=3700, m=10
## t=3800, m=8
## t=3900, m=8
## t=4000, m=10
## t=4100, m=8
## t=4200, m=9
## t=4300, m=9
## t=4400, m=8
## t=4500, m=10
## t=4600, m=10
## t=4700, m=9
## t=4800, m=9
## t=4900, m=8
## t=5000, m=8
## t=5100, m=10
## t=5200, m=8
## t=5300, m=10
## t=5400, m=11
## t=5500, m=10
## t=5600, m=8
## t=5700, m=10
## t=5800, m=12
## t=5900, m=8
## t=6000, m=7
## t=6100, m=12
## t=6200, m=11
## t=6300, m=12
```

```
## t=6400, m=9
## t=6500, m=10
## t=6600, m=10
## t=6700, m=9
## t=6800, m=8
## t=6900, m=8
## t=7000, m=8
## t=7100, m=9
## t=7200, m=10
## t=7300, m=8
## t=7400, m=9
## t=7500, m=8
## t=7600, m=7
## t=7700, m=8
## t=7800, m=10
## t=7900, m=10
## t=8000, m=8
## t=8100, m=10
## t=8200, m=7
## t=8300, m=13
## t=8400, m=9
## t=8500, m=8
## t=8600, m=10
## t=8700, m=9
## t=8800, m=7
## t=8900, m=8
## t=9000, m=8
## t=9100, m=8
## t=9200, m=12
## t=9300, m=9
## t=9400, m=10
## t=9500, m=7
## t=9600, m=11
## t=9700, m=9
## t=9800, m=8
## t=9900, m=11
```

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

Histogram of x$m[burnin:x$T]

m chain


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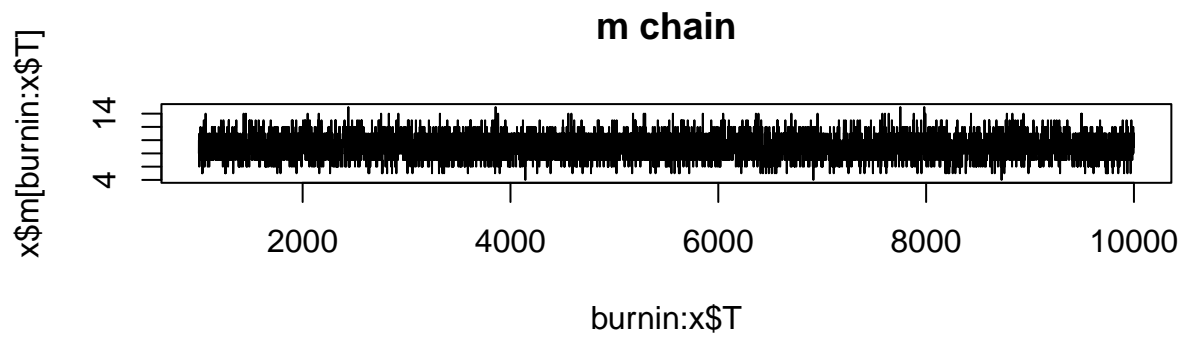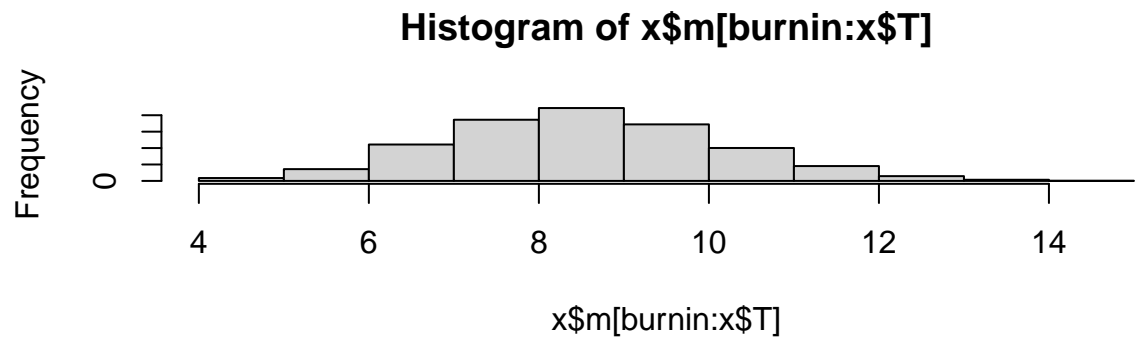
Series fit$m

Now we do not see any periodicity and it is much stable.

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

Histogram of x$s2[burnin:x$T]

s2 chain
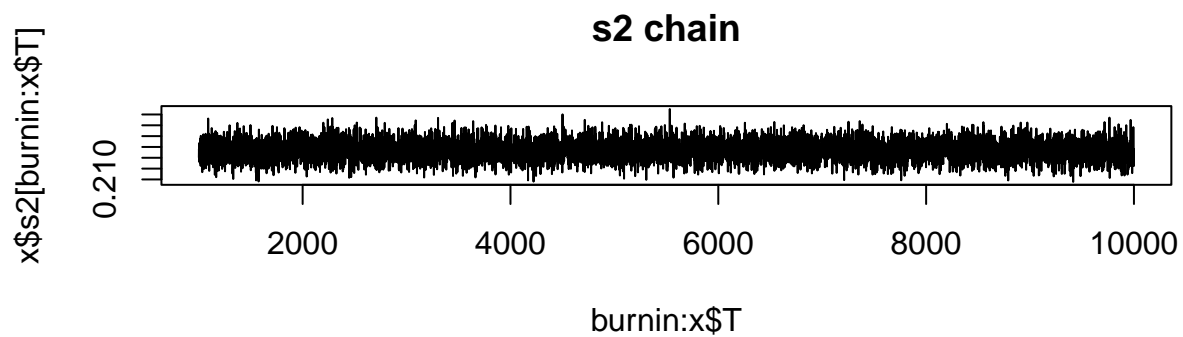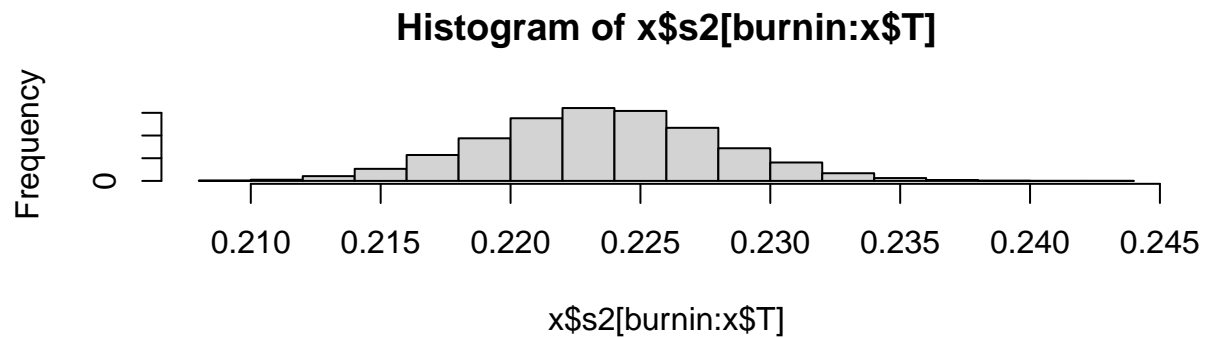

```
plot(fit, burnin=1000, which="lambda2")
```

Histogram of x$lambda2[burnin:x$T]

lambda2 chain
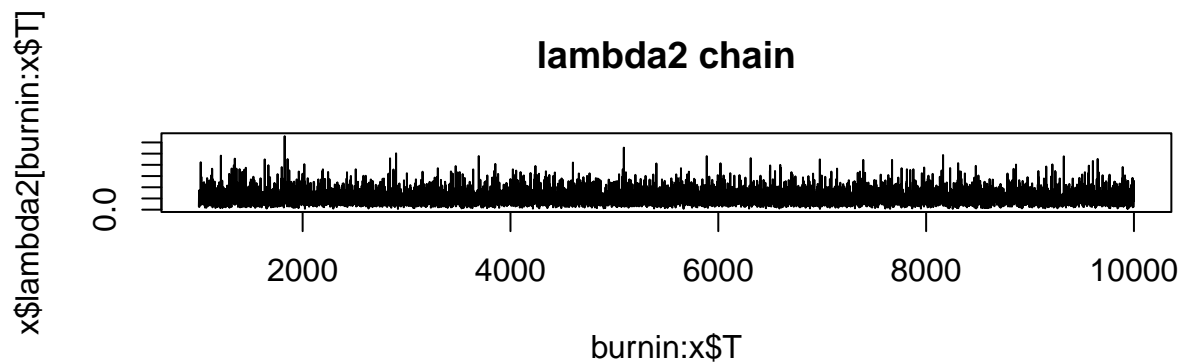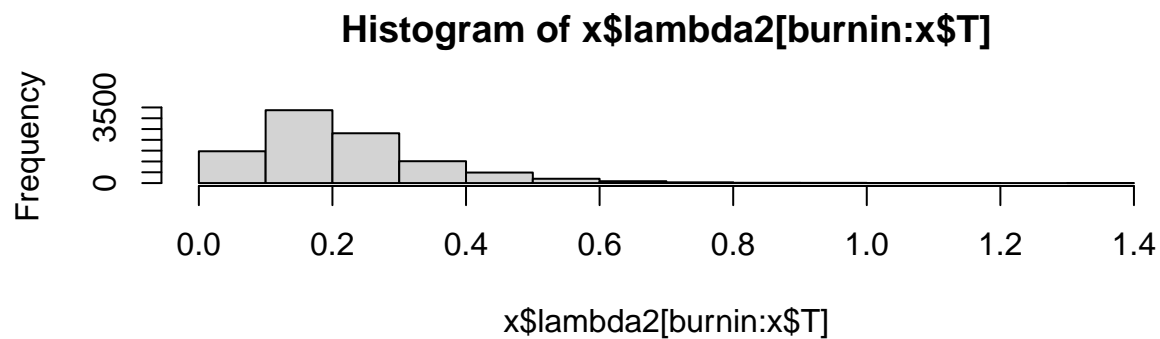
And also we see that it is stable so we can trust that it has converged. So let's see the most important variables.

```
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))
```

Boxplots of regression coefficients
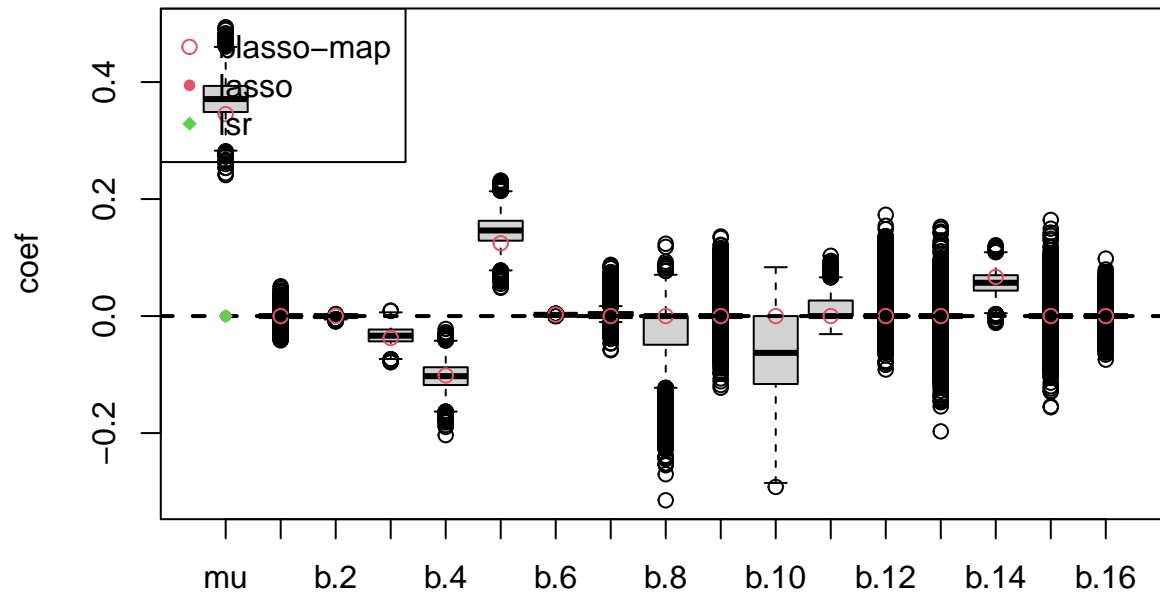

```
s <- summary(fit, burnin=1000)
print(s$bn0) # probability that each beta coef != zero
```

```
##       b.1       b.2       b.3       b.4       b.5       b.6       b.7       b.8
## 0.2672222 0.5098889 0.9303333 1.0000000 1.0000000 0.9997778 0.3573333 0.4225556
##       b.9      b.10      b.11      b.12      b.13      b.14      b.15      b.16
## 0.2615556 0.6800000 0.5247778 0.3250000 0.2566667 0.9750000 0.2667778 0.2620000
```

```
barplot(s$bn0, horiz = TRUE)
```
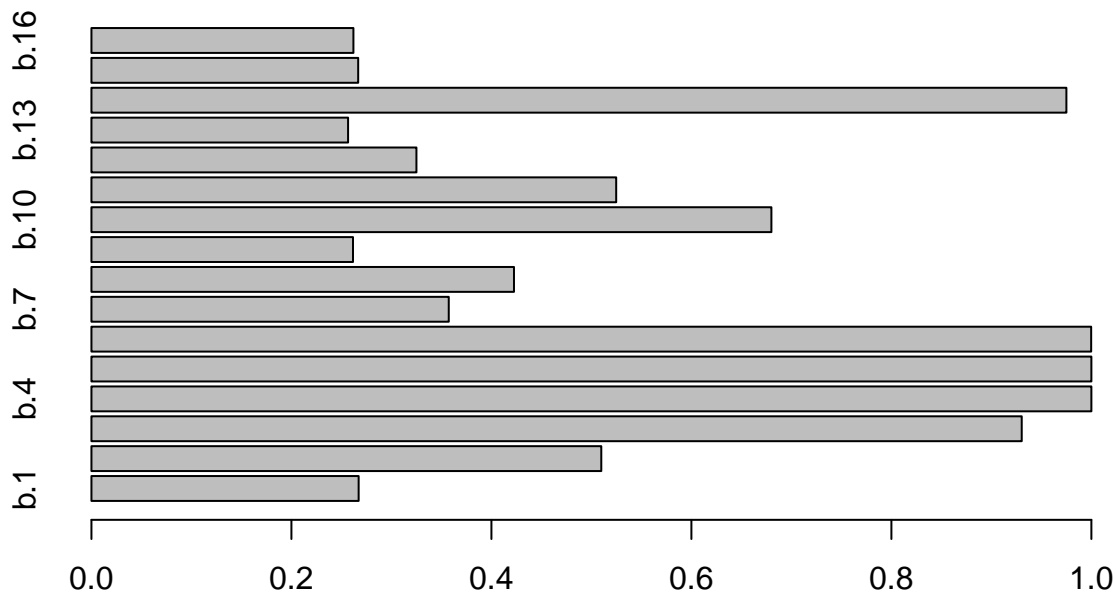
Here we see that the most important variables are SEX, PATIENT_TYPE, PNEUMONIA, AGE, CARDIO-VASCULAR. All in all we can conclude that the post important variables to predict if a patient has covid or not is:

- Sex: we will see but a specific gender is prone to have covid.
- Patient type: this makes sense because if a patient has been hospitalized it is more likely that it will have covid.
- Age: elder people are more likely to have covid.
- Pneumonia: as the covid makes similar symptoms as pneumonia, patients with it are much more likely to have it.
- Cardiovascular: we can see that covid could be more likely to be contracted if you have cardiovascular problems.

Final Model

Finally we will create a model with the variables selected with lasso.

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

library(R2OpenBUGS)

logit.bayes <- function(){
  for( i in 1 : n ) {
    COVID.bin[i] ~ dbern(p[i])
    logit(p[i]) <- b0 + b1 * AGE[i] + b2*SEX.male[i] + b3*PATITENT_TYPE.hospitalized[i] + b4*PNEUMONIA.
  }

  b0 ~ dnorm(0.0, 1.0E-6)
  b1 ~ dnorm(0.0, 1.0E-6)
  b2 ~ dnorm(0.0, 1.0E-6)
  b3 ~ dnorm(0.0, 1.0E-6)
  b4 ~ dnorm(0.0, 1.0E-6)
  b5 ~ dnorm(0.0, 1.0E-6)
}
COVID.bin=ifelse(data.train$COVID,1,0)
```

```
n=length(COVID.bin)
SEX.male = ifelse(data.train$SEX=="male",1,0)
PATITENT_TYPE.hospitalized = ifelse(data.train$PATIENT_TYPE=="hospitalized",1,0)
PNEUMONIA.yes = ifelse(data.train$PNEUMONIA=="pneumonia",1,0)
CARDIOVASCULAR.yes = ifelse(data.train$CARDIOVASCULAR=="cardiovascular",1,0)

data <- list(n=n, COVID.bin=COVID.bin, AGE=data.train$AGE, SEX.male = SEX.male, PATITENT_TYPE.hospitali

inits <- function(){
  list(b0 = 1, b1 = 0, b2 = 0, b3 = 0, b4 = 0, b5 = 0)
}
output <- bugs(data = data, inits = inits, parameters.to.save = c("b0", "b1", "b2", "b3", "b4", "b5"),
```

```
output
```

```
## Inference for Bugs model at "/tmp/RtmpUmGfKp/model154db567ac377.txt",
## Current: 1 chains, each with 1000 iterations (first 100 discarded)
## Cumulative: n.sims = 900 iterations saved
##              mean  sd   2.5%    25%    50%    75%  97.5%
## b0           -1.2 0.1   -1.4   -1.3   -1.2   -1.2   -1.0
## b1            0.0 0.0    0.0    0.0    0.0    0.0    0.0
## b2            0.2 0.1    0.1    0.1    0.2    0.2    0.3
## b3            0.5 0.1    0.3    0.4    0.5    0.5    0.7
## b4            0.6 0.1    0.4    0.6    0.6    0.7    0.9
## b5            0.0 0.2   -0.4   -0.2    0.0    0.1    0.4
## deviance 6396.5 3.6 6392.0 6394.0 6396.0 6398.0 6405.0
##
## DIC info (using the rule, pD = Dbar-Dhat)
## pD = 6.1 and DIC = 6403.0
## DIC is an estimate of expected predictive error (lower deviance is better).
```

```
b0.post <-output$sims.list$b0
b1.post <-output$sims.list$b1
b2.post <-output$sims.list$b2
b3.post <-output$sims.list$b3
b4.post <-output$sims.list$b4
b5.post <-output$sims.list$b5
ts.plot(b0.post)
```
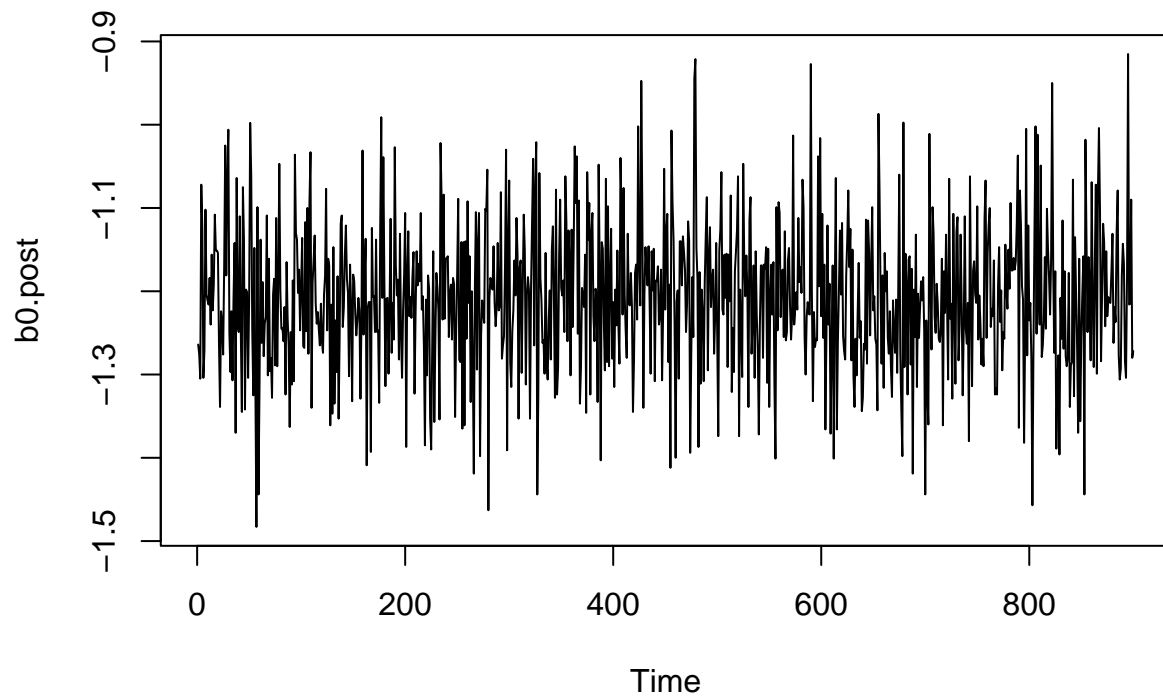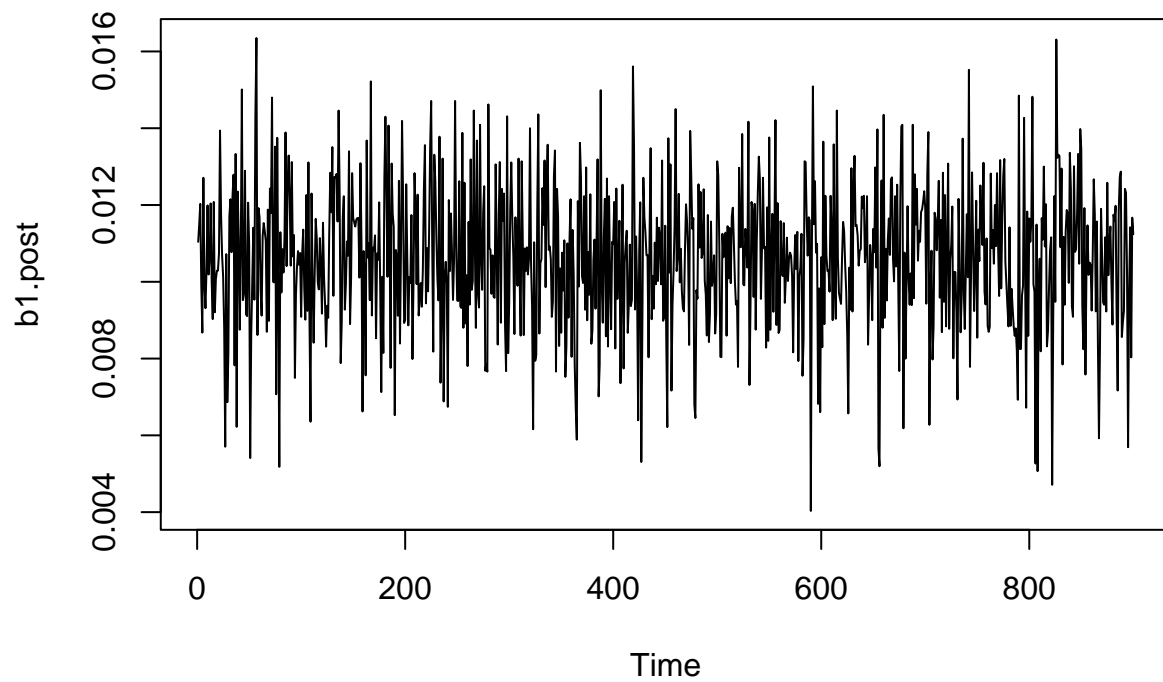
```
acf(b0.post)
```
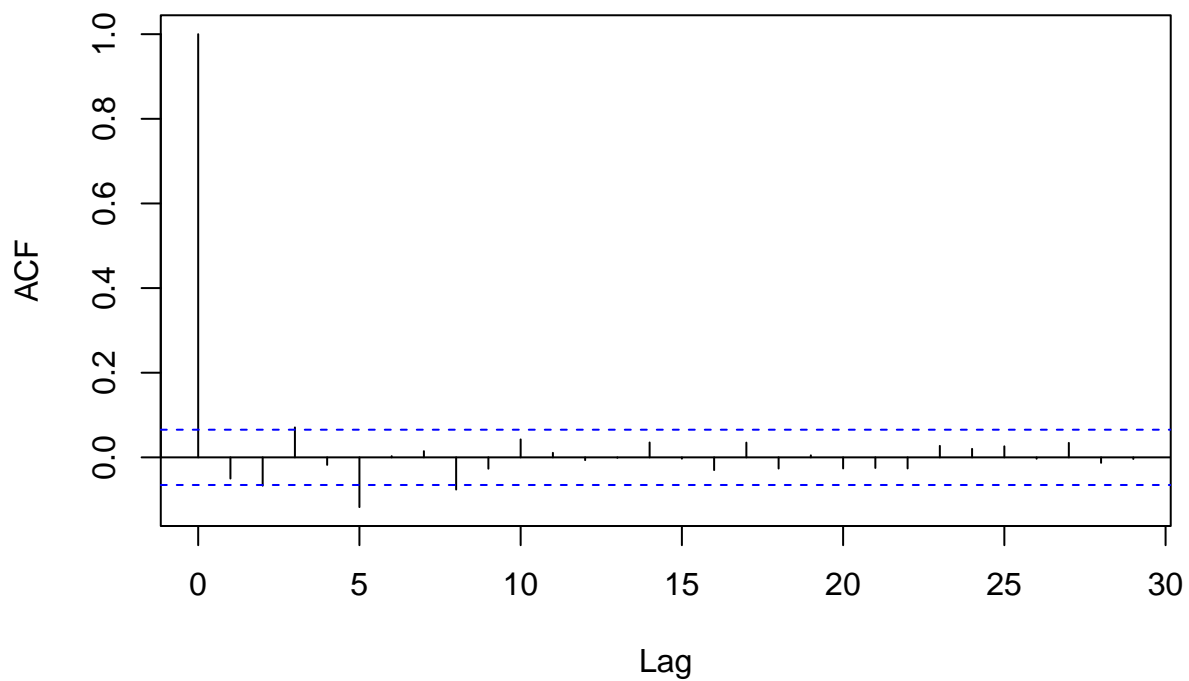
Series b0.post
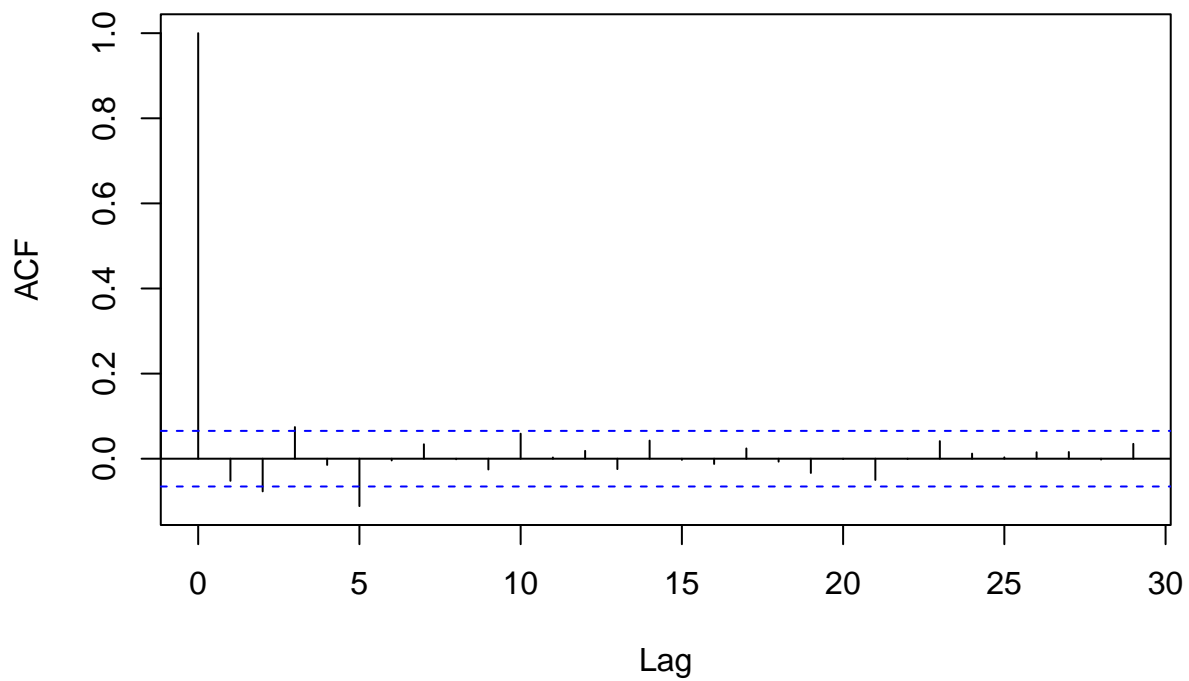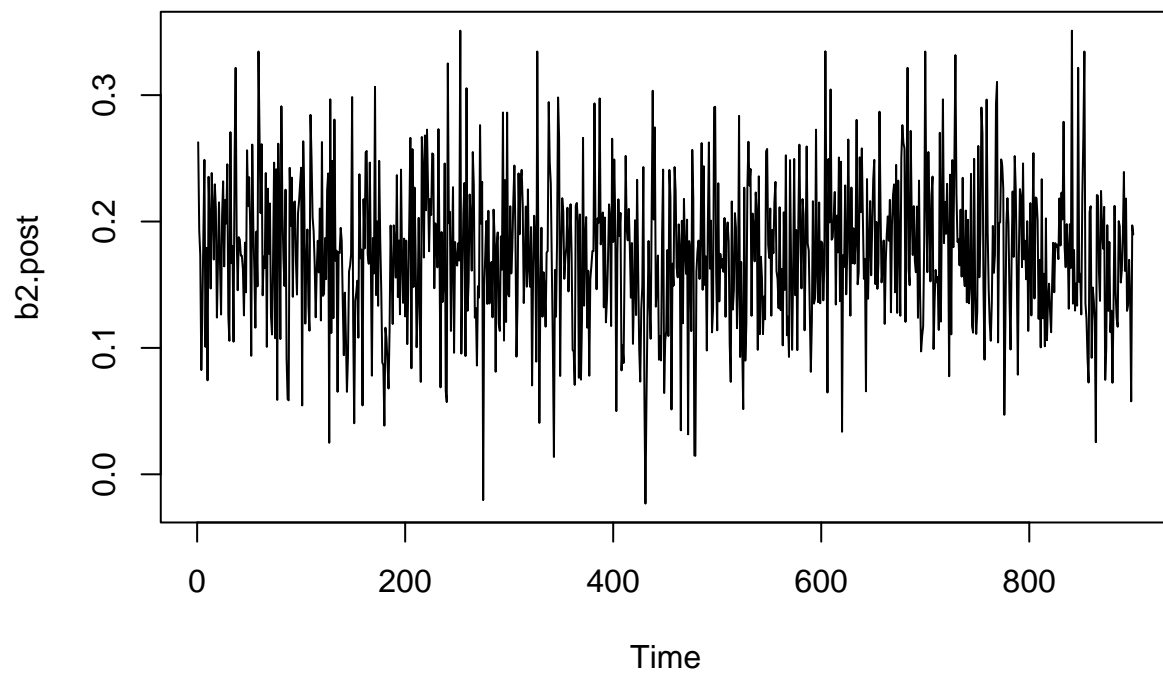


```
ts.plot(b1.post)
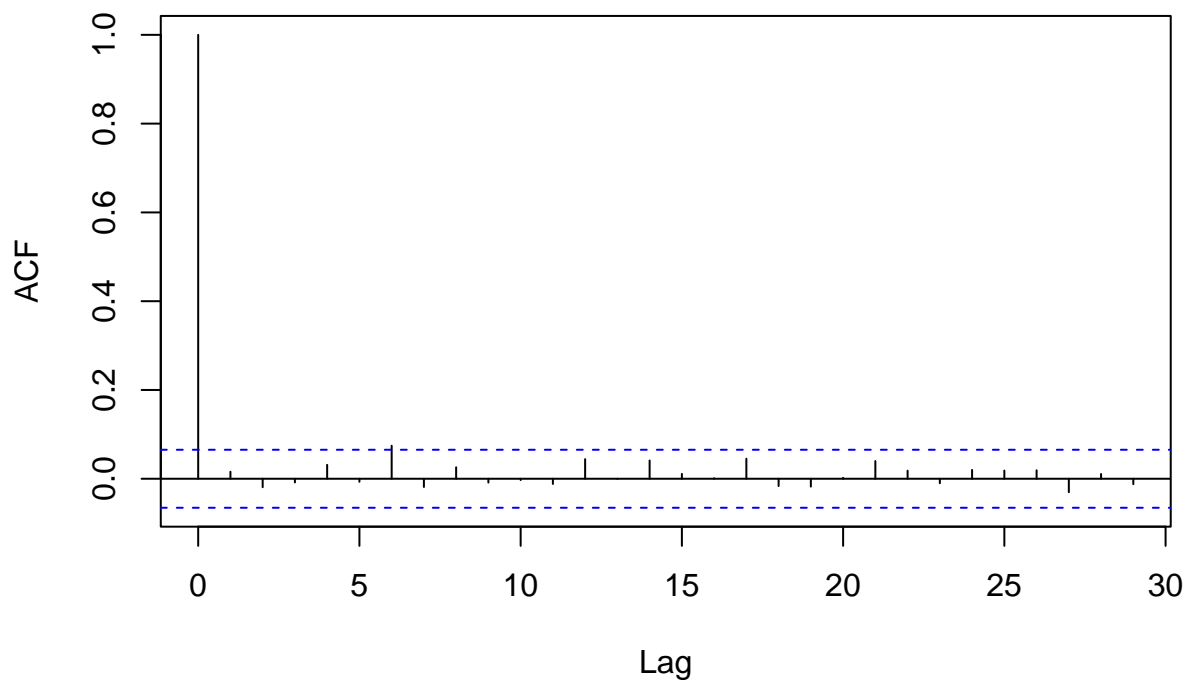```

```
acf(b1.post)
```

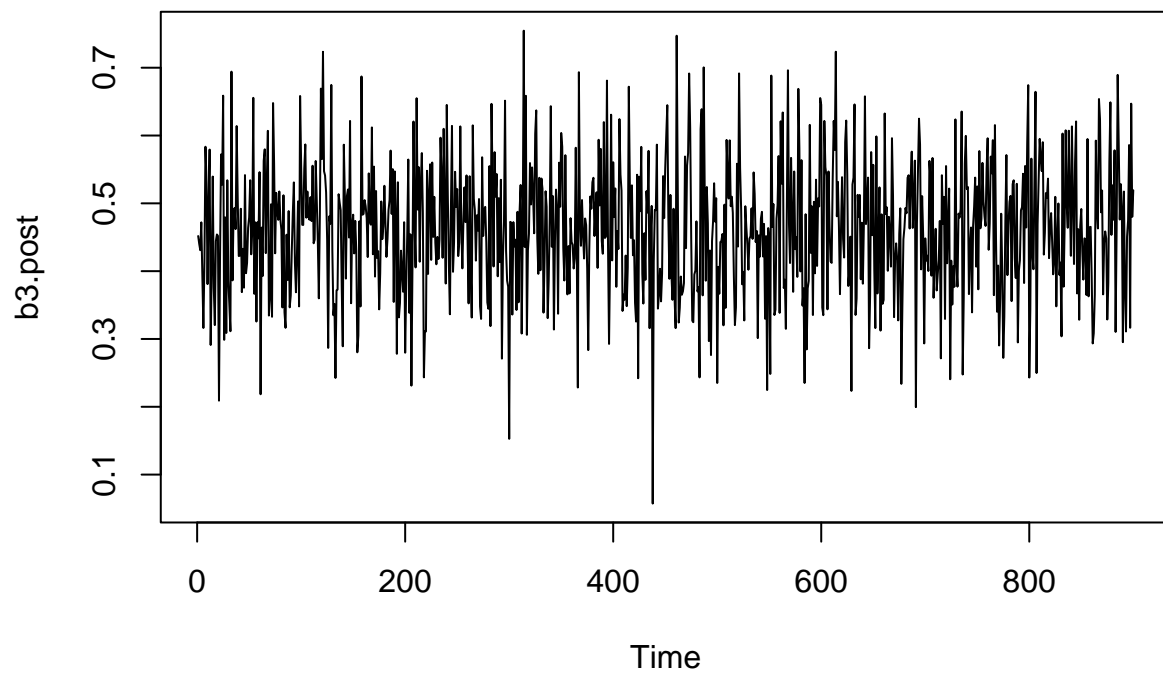**Series b1.post**


```
ts.plot(b2.post)
```

```
acf(b2.post)
```

Series b2.post


```
ts.plot(b3.post)
```
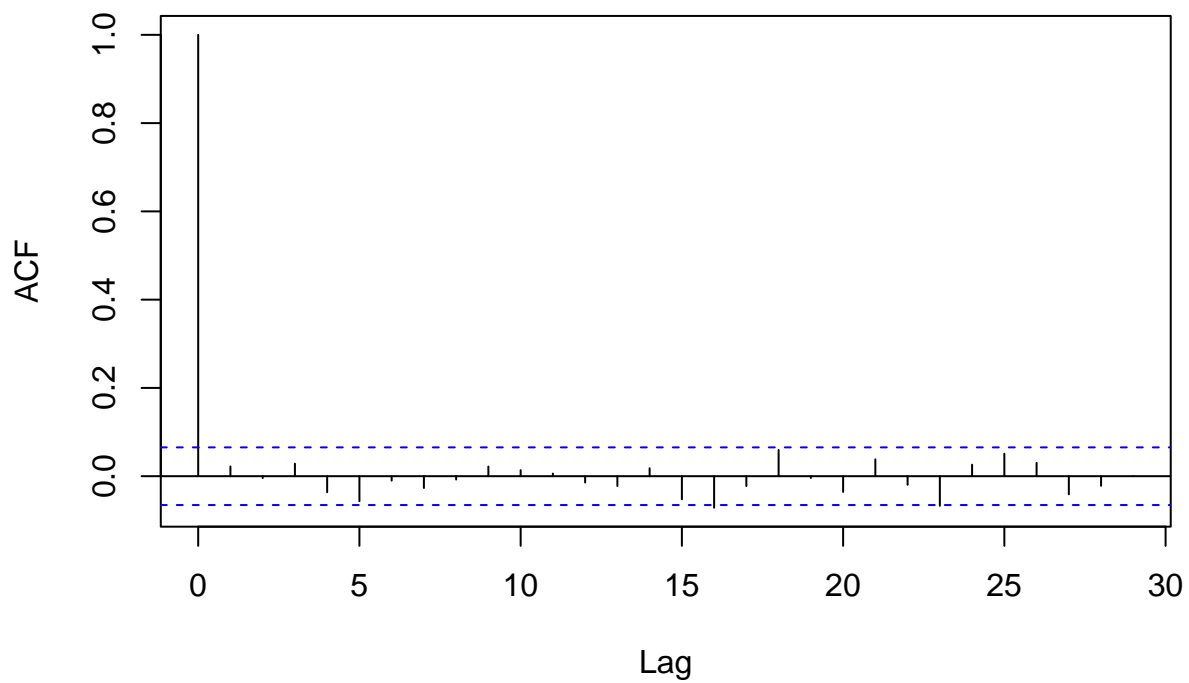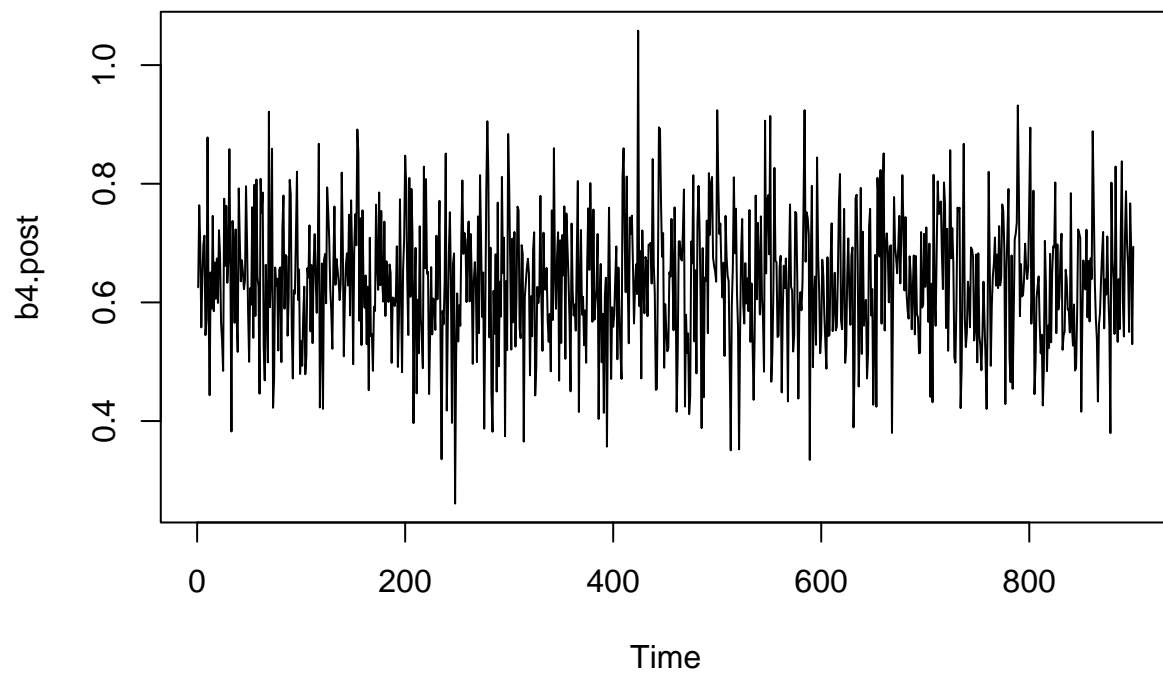
```
acf(b3.post)
```
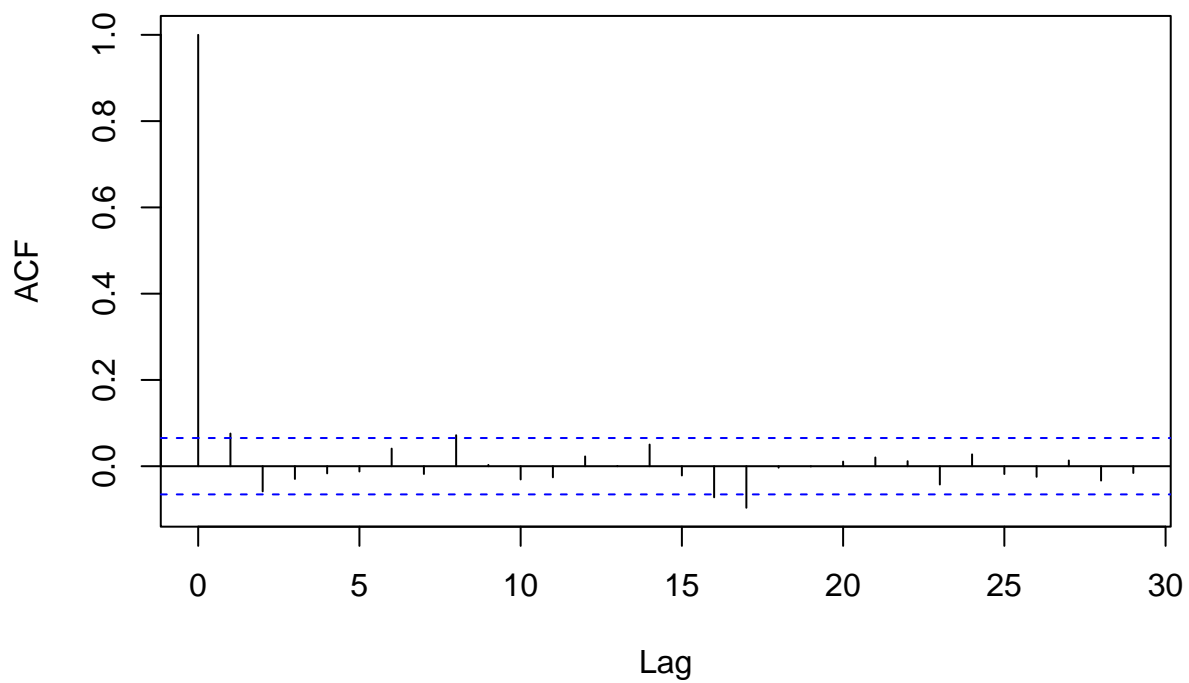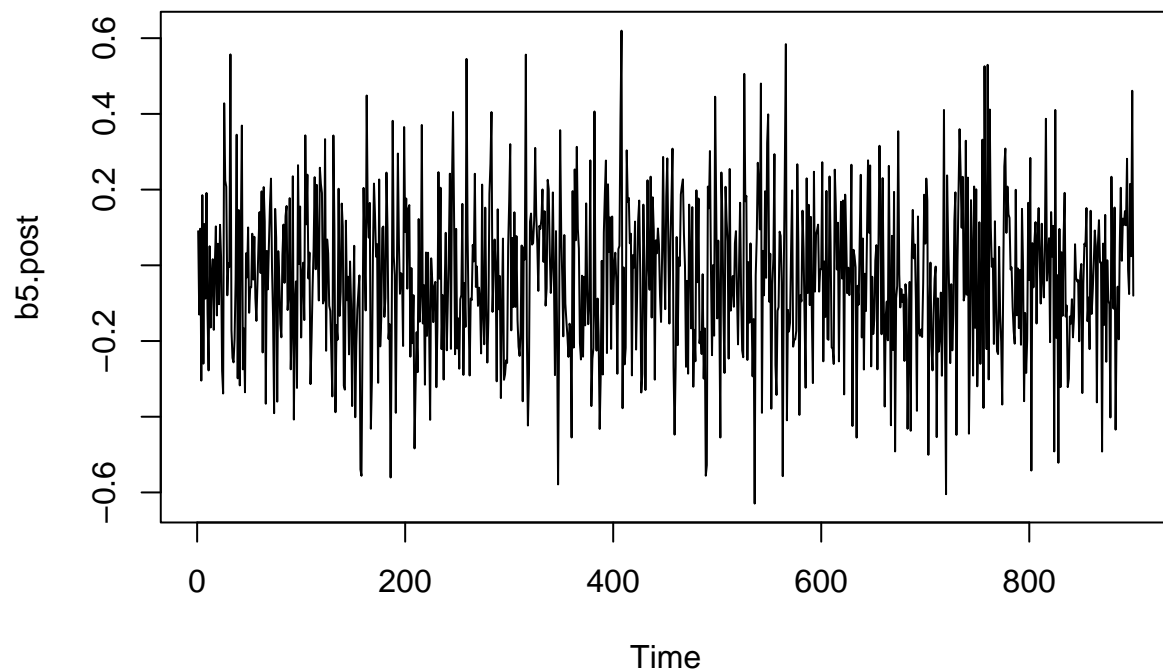
Series b3.post


```
ts.plot(b4.post)
```

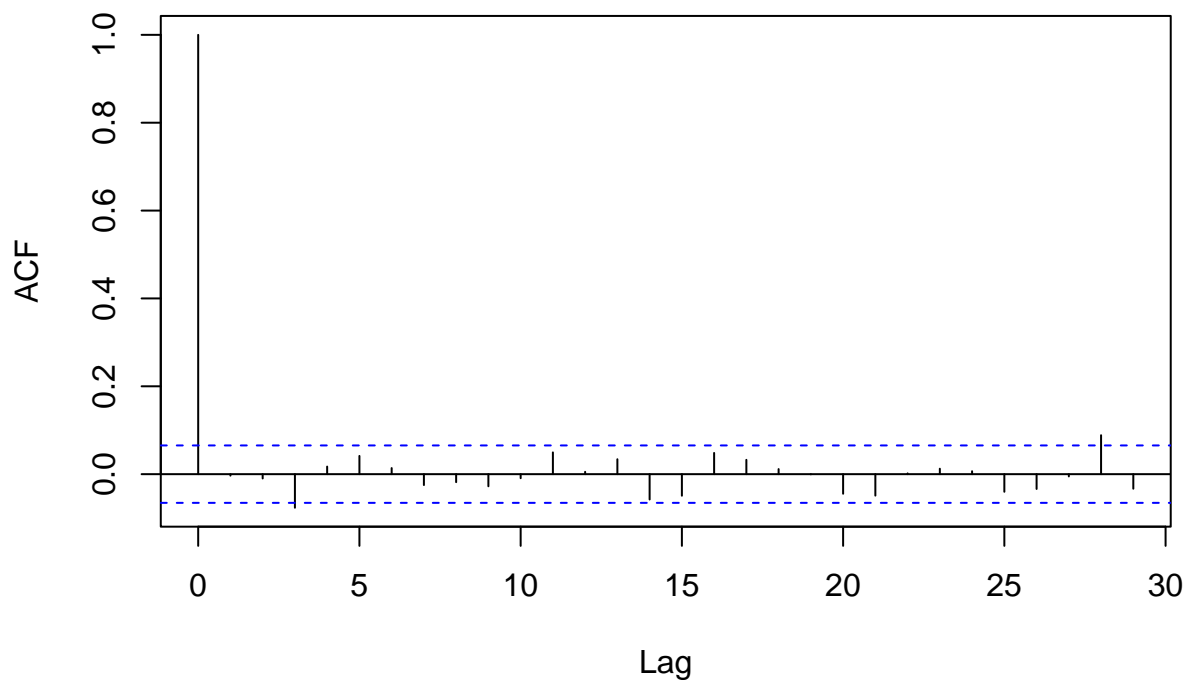```
acf(b4.post)
```

Series b4.post


```
ts.plot(b5.post)
```

```
acf(b5.post)
```

Series b5.post

We see that the model is stable so let's check some assumptions. First let's create the baseline.

```
linear = b0.post
pred.baseline = exp(linear)/(1+exp(linear))

mean(pred.baseline)
```

```
## [1] 0.2303748
```

```
quantile(pred.baseline,c(0.025,0.975))
```

```
##      2.5%     97.5%
## 0.1997193 0.2643609
```

Here we see that a baseline person has a 0.22 probability of having covid. So now let's compare it to other groups.

Age

20 years old

```
linear = b0.post+b1.post * 20
pred.prob = exp(linear)/(1+exp(linear))

mean(pred.prob)
```

```
## [1] 0.2698846
```

```
quantile(pred.prob,c(0.025,0.975))
```

```
##      2.5%     97.5%
## 0.2469362 0.2938482
```

50 years old

```
linear = b0.post+b1.post * 50
pred.prob = exp(linear)/(1+exp(linear))

mean(pred.prob)
```

```
## [1] 0.3368903
```

```
quantile(pred.prob,c(0.025,0.975))
```

```
##      2.5%     97.5%
## 0.3149719 0.3574110
```

80 years old

```
linear = b0.post+b1.post * 80
pred.prob = exp(linear)/(1+exp(linear))

mean(pred.prob)
```

```
## [1] 0.4113386
```

```
quantile(pred.prob,c(0.025,0.975))
```

```
##      2.5%     97.5%
## 0.3692872 0.4539434
```

Here we can clearly see that the higher the age, the more probability people have to have covid.

Sex

Female

```
linear = b0.post
pred.prob = exp(linear)/(1+exp(linear))
```

```
mean(pred.prob)
```

```
## [1] 0.2303748
```

```
quantile(pred.prob,c(0.025,0.975))
```

```
##      2.5%     97.5%
## 0.1997193 0.2643609
```

Male

```
linear = b0.post+b2.post * 1
pred.prob = exp(linear)/(1+exp(linear))

mean(pred.prob)
```

```
## [1] 0.262799
```

```
quantile(pred.prob,c(0.025,0.975))
```

```
##      2.5%     97.5%
## 0.2311018 0.3004539
```

We see some indication that the male population has higher probability to have covid but it is not significant.

Patient Type

Not hospitalized

```
linear = b0.post
pred.prob = exp(linear)/(1+exp(linear))

mean(pred.prob)
```

```
## [1] 0.2303748
```

```
quantile(pred.prob,c(0.025,0.975))
```

```
##      2.5%     97.5%
## 0.1997193 0.2643609
```

Hospitalized

```
linear = b0.post+b3.post * 1
pred.prob = exp(linear)/(1+exp(linear))

mean(pred.prob)
```

```
## [1] 0.322236
```

```
quantile(pred.prob,c(0.025,0.975))
```

```
##      2.5%     97.5%
## 0.2669285 0.3834891
```

Now, this is really significant and it makes sense. If a patient has been hospitazlied, it is really likely that he/she has covid.

Pneumonia

Not pneumonia

```
linear = b0.post
pred.prob = exp(linear)/(1+exp(linear))

mean(pred.prob)
```

```
## [1] 0.2303748
```

```
quantile(pred.prob,c(0.025,0.975))
```

```
##      2.5%     97.5%
## 0.1997193 0.2643609
```

Pneumonia

```
linear = b0.post+b4.post * 1
pred.prob = exp(linear)/(1+exp(linear))

mean(pred.prob)
```

```
## [1] 0.3608047
```

```
quantile(pred.prob,c(0.025,0.975))
```

```
##      2.5%     97.5%
## 0.3014263 0.4228176
```

Same conclusions for pneumonia

Cardiovascular

Not cardiovascular

```
linear = b0.post
pred.prob = exp(linear)/(1+exp(linear))

mean(pred.prob)
```

```
## [1] 0.2303748
```

```
quantile(pred.prob,c(0.025,0.975))
```

```
##      2.5%     97.5%
## 0.1997193 0.2643609
```

Cardiovascular

```
linear = b0.post+b5.post * 1
pred.prob = exp(linear)/(1+exp(linear))

mean(pred.prob)
```

```
## [1] 0.2265248
```

```
quantile(pred.prob,c(0.025,0.975))
```

```
##      2.5%     97.5%
## 0.1544751 0.3139523
```

But for cardiovascular we are not really sure that it plays a big role, so we will not make assumptions.

Conclusions

We have seen that many variables in this data set are not useful to predict if a patient has covid or not, but with the Bayesian approach and lasso, we have found some that are significant (in this case: age, patient type, and pneumonia) this makes sense. But the power of Bayesian approach is that we obtain the posterior distribution so we can see how significant and if we trust the variable. As for example, the sex and the cardiovascular had really big confidence intervals so we discarded them. This shows the usefulness of the Bayesian approach in comparison with the frequentest one.