Individual assignment SBD

Hagen Aad Fock

ADS, 2021-2022

Contents

0. Prepare

▶ Load the R-packages you will use.

```
library(fpp3)
library(tseries)
library(expsmooth)

library(tidyverse)    # alternatively, this also loads %>%
library(knitr)
```

```
library(mice) # for missing data imputation
library(VIM)
library(zoo)
```

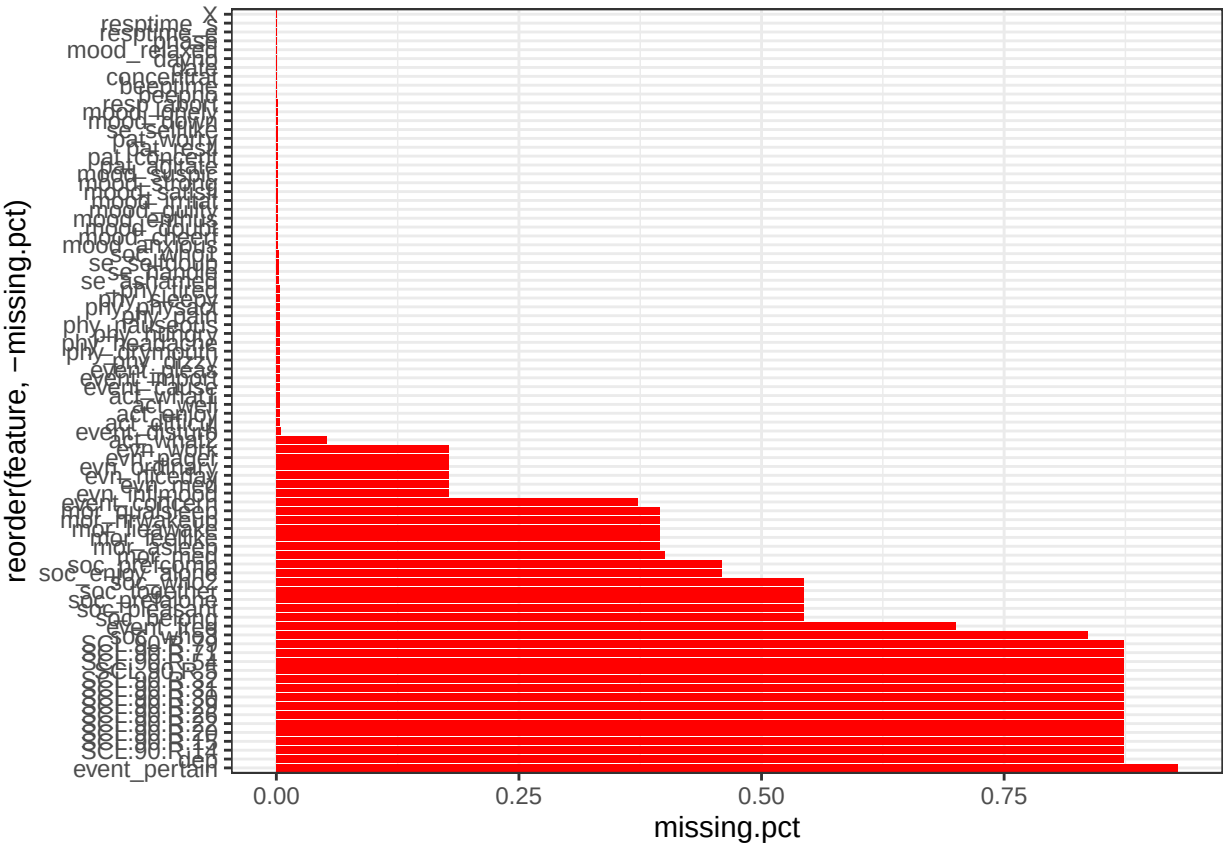
▶ Include R-code you used to load (and prepare) the data.

0.1 Checking for missing data

```
set.seed(666)

data <- read.csv("../ESMdata/ESMdata.csv")
missing.values <- data %>%
  summarize_all(funs(sum(is.na(.))/n())) %>%
  gather(key="feature", value="missing.pct")
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

```
missing.values %>%
  ggplot(aes(x=reorder(feature,-missing.pct),y=missing.pct)) +
  geom_bar(stat="identity",fill="red")+
  coord_flip()+theme_bw()
```

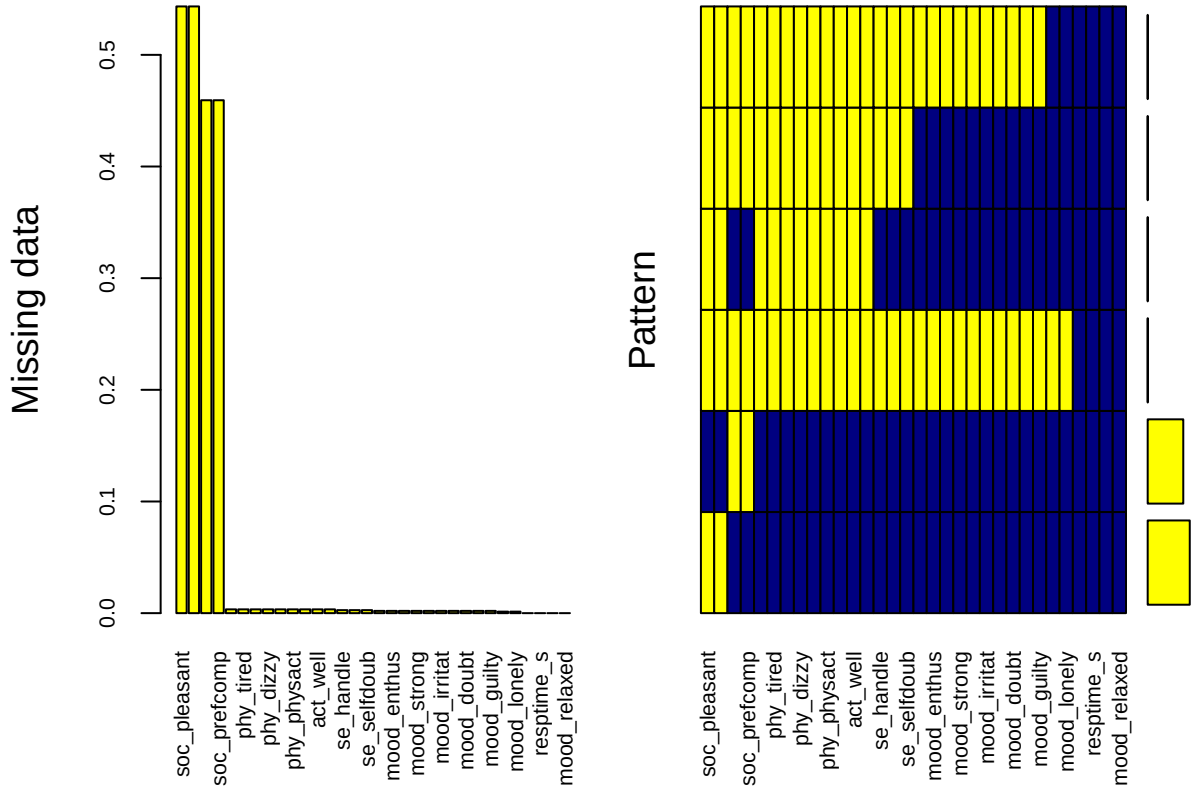


There are several variables that have a high percentage of missing data. Within my analysis I don't consider any variable that has a higher missing percentage then 3.4%. Except attributes with a soc_ prefix. They are displayed with an high percentage of missing data but the missing data has a underlying structure.

```
relevant.variables <- c('date', 'resptime_s',  'concentrat',

                        'mood_relaxed', 'mood_satisfi', 'mood_enthus',
                        'mood_cheerf', 'mood_strong', 'se_selflike',
                        'se_handle',

                        'mood_irritat', 'mood_suspic', 'mood_doubt',
                        'se_ashamed', 'se_selfdoub',

                        'mood_down', 'mood_lonely', 'mood_anxious',
                        'mood_guilty',

                        'phy_hungry', 'phy_tired', 'phy_pain', 'phy_dizzy',
                        'phy_headache',

                        'soc_enjoy_alone', 'soc_prefcomp',
                        'soc_pleasant', 'soc_prefalone',
```

```
                        'phy_physact', 'act_difficul',
                        'act_well', 'act_enjoy')
data <- data[,relevant.variables]

mice.plot <- aggr(data[,relevant.variables], col=c('navyblue','yellow'),
                  numbers=TRUE, sortVars=TRUE,
                  labels=names(relevant.variables), cex.axis=.7,
                  gap=3, ylab=c("Missing data","Pattern"))
```

```
## Warning in plot.aggr(res, ...): not enough horizontal space to display
## frequencies
```




```
##
##  Variables sorted by number of missings:
##         Variable       Count
##     soc_pleasant 0.543360434
##    soc_prefalone 0.543360434
##  soc_enjoy_alone 0.459349593
##     soc_prefcomp 0.459349593
##       phy_hungry 0.003387534
##        phy_tired 0.003387534
##         phy_pain 0.003387534
##        phy_dizzy 0.003387534
```

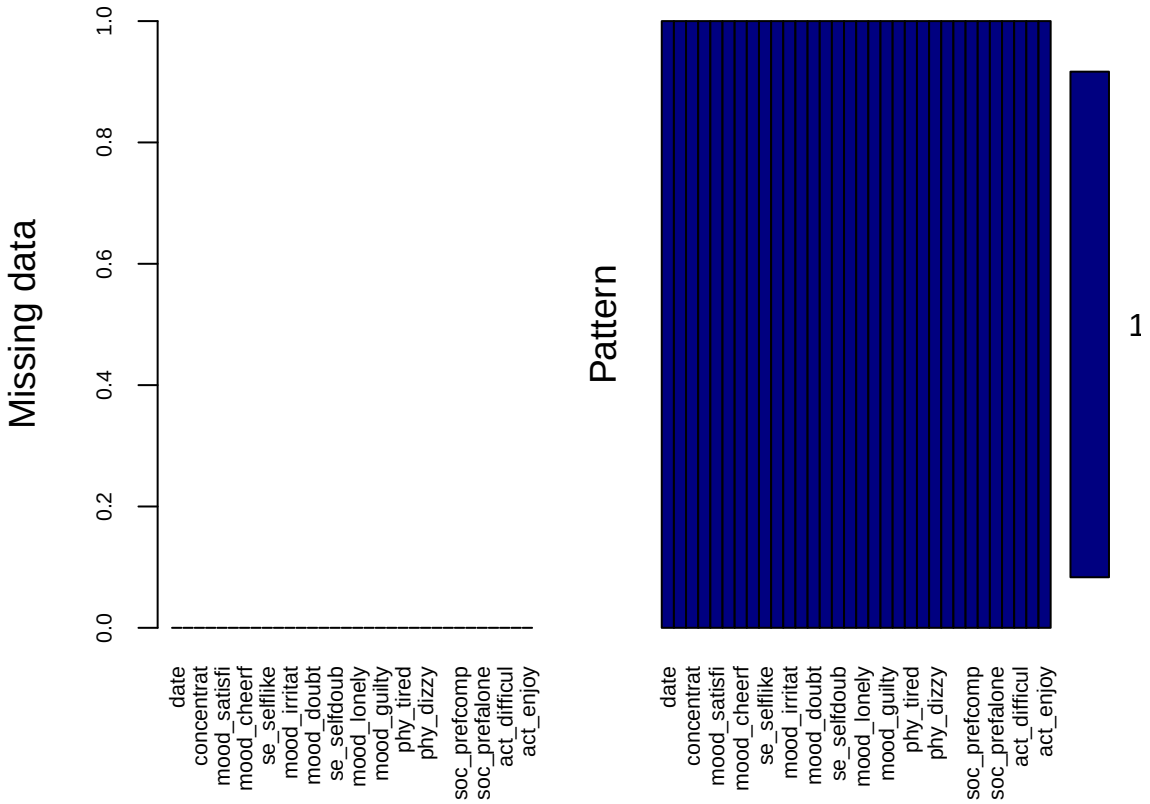
```
##     phy_headache 0.003387534
##      phy_physact 0.003387534
##     act_difficul 0.003387534
##         act_well 0.003387534
##        act_enjoy 0.003387534
##        se_handle 0.002710027
##       se_ashamed 0.002710027
##      se_selfdoub 0.002710027
##     mood_satisfi 0.002032520
##      mood_enthus 0.002032520
##      mood_cheerf 0.002032520
##      mood_strong 0.002032520
##      se_selflike 0.002032520
##     mood_irritat 0.002032520
##      mood_suspic 0.002032520
##       mood_doubt 0.002032520
##     mood_anxious 0.002032520
##      mood_guilty 0.002032520
##        mood_down 0.001355014
##      mood_lonely 0.001355014
##             date 0.000000000
##       resptime_s 0.000000000
##       concentrat 0.000000000
##     mood_relaxed 0.000000000
```

```
# used this command to figure out the exact value of a variable
# missing.values[missing.values$feature == 'soc_prefcomp',]$missing.pct
```

0.2 Imputing missing data

```
## Warning: Number of logged events: 2
```

```
mice.plot <- aggr(data, col=c('navyblue','yellow'),
                  numbers=TRUE, sortVars=TRUE,
                  labels=names(data), cex.axis=.7,
                  gap=3, ylab=c("Missing data","Pattern"))
```




```
##
##  Variables sorted by number of missings:
##        Variable Count
##            date     0
##       resptime_s     0
##       concentrat     0
##     mood_relaxed     0
##     mood_satisfi     0
##      mood_enthus     0
##      mood_cheerf     0
##      mood_strong     0
##      se_selflike     0
##        se_handle     0
##     mood_irritat     0
##      mood_suspic     0
##       mood_doubt     0
##       se_ashamed     0
##      se_selfdoub     0
##        mood_down     0
##      mood_lonely     0
##     mood_anxious     0
##      mood_guilty     0
##       phy_hungry     0
```

```
##        phy_tired     0
##         phy_pain     0
##        phy_dizzy     0
##     phy_headache     0
##  soc_enjoy_alone     0
##     soc_prefcomp     0
##     soc_pleasant     0
##    soc_prefalone     0
##      phy_physact     0
##     act_difficul     0
##         act_well     0
##        act_enjoy     0
```

0.3 Tsibble

Now that every missing data was imputed we can make the tsibble. I create the tsibble by aggregating the mean of every day. The records of a day varied within a range of 1 and 10. I assumed that within such a day with many records the patients mood must have changed a lot otherwise they wouldn't record that much. Deriving from this thought I assumed that having the mean of such a day is more feasible for my analysis then including such a roller coaster ride.

```
data$date <- dmy(data$date)
data <- aggregate(data[, 3:32], list(data$date), mean)
names(data)[1] <- 'date'

data <- data %>%
  as_tsibble(index = date)

# There is no entry for 2012-12-15 therefore I impute the previous values.
data <- tsibble::fill_gaps(data)
data <- na.locf(na.locf(data),fromLast=TRUE)
```

1. General

▶ To be able to use fpp3, the data have to be a tsibble object. If they aren't already, transform them. Describe the structure of this object.

A tsibble is time series optimized tibble. It has in addition an Index that has an inherent ordering from past to present. Also has a key variable so multiple time series are possible. If there are implicit missing values they can be easily converted into explicit missing values with the `fill_gaps()` function. And around the tsibble is again a little tidyverse called `tidyverts` which includes a lot of libraries that are useful for time series analyses.

https://tsibble.tidyverts.org/

1.1. Describe your data

The data is about a man that was reducing his anti depression medication. Every variable that will be stated within this section was measured using a semi-random experience-sampling protocol. “The participant collected reports of momentary states up to 10 times a day over a period of 239 days.” https://doi.org/10.1159/000441458

“Depression is a mood disorder that causes a persistent feeling of sadness and loss of interest. Also called major depressive disorder or clinical depression, it affects how you feel, think and behave and can lead to a variety of emotional and physical problems. You may have trouble doing normal day-to-day activities, and sometimes you may feel as if life isn’t worth living.” https://www.mayoclinic.org/diseases-conditions/depression/symptoms-causes/syc-20356007

▶ What is your outcome variable; how was it measured (how many times, how frequently, etc.)?

The components of my outcome variable were measured at least everyday over 239 days and also sometimes sub daily. The total amount of measurements is approximately 1470 times.

Regarding the information from the article about depression I came up with the assumption that the best way to measure a depression is to observe the patients mood. Because there is such a variety of positive and negative moods I mixed them up in an overall variable called `depression_factor`.

All considered variables are within the following table.

name	description	scale
`mood_relaxed`	I feel relaxed	+(1, 7)
`mood_satisfi`	I feel satisfied	+(1, 7)
`mood_enthus`	I feel enthusiastic	+(1, 7)
`mood_cheerf`	I feel cheerful	+(1, 7)
`mood_strong`	I feel strong	+(1, 7)
`se_selflike`	I like myself	+(1, 7)
`se_handle`	I can handle anything	+(1, 7)
`positive_moods`	Accumulated positive moods. All variables with a ‘positive (1, 7)’ scale combined.	+(-1, 1)
`mood_irritat`	I feel irritated	-(1, 7)
`mood_suspic`	I feel suspicious	-(1, 7)
`mood_doubt`	I feel indecisive	-(1, 7)
`se_ashamed`	I am ashamed of myself	-(1, 7)
`se_selfdoub`	I doubt myself	-(1, 7)
`negative_moods`	Accumulated negative moods. All variables with a ‘negative (1, 7)’ scale combined.	-(-1, 1)
`mood_down`	I feel down	-(-3, 3)
`mood_lonely`	I feel lonely	-(-3, 3)

name	description	scale
`mood_anxious`	I feel anxious	-(-3, 3)
`mood_guilty`	I feel guilty	-(-3, 3)
`depressive_moods`	Accumulated depressive moods. All variables with a 'negative (-3, 3)' scale combined.	-(-1, 1)
`depression_factor`	An accumulated factor that is an approximation to measure the depression. A combination of the variables `positive_moods`, `negative_moods` and `depressive_moods`	+(-1, 1)

Because of the different scales within the variables, I had to transform them to the same base. I decided if the highest score on its scale is towards a positive or a negative mood for every variable. Then I scaled them depending on my assumption towards a scale of -1 and 1, where a positive score represents a positive mood, and a negative score represents a negative mood.

```
# the variable declaration is following
# _positive indicating if the highest value is a good mood
# _negative indicating if the highest value is a depressive mood
# _17       indicating if the scale is (1, 7)
# _33       indicating if the scale is (-3, 3)

# Data Normalization - Min-Max Normalization
# from (-3, 3) to (-1, 1)
Normalize <- function(atr, old_min, old_max)(
  return((atr - (old_min)) / (old_max - (old_min)) * (1 - (-1)) + (-1))
)

# For attributes on a scale (1, 7) where 1 indicates a negative mood and
# 7 a positive mood.
NormalizePositive17 <-  function(atr) (
  return(Normalize((atr - 4), -3, 3))
)

# For attributes on a scale (1, 7) where 1 indicates a negative mood and
# 7 a positive mood.
NormalizeNegative17 <- function(atr) (
  return(Normalize(((atr - 4) * -1), -3, 3))
)

# For attributes on a scale(-3, 3) where 3 is a depressive value.
NormalizeNegative33 <- function(atr) (
  return(Normalize((atr * -1), -3, 3))
)
```

```r
# positive_moods
data$mood_relaxed <- NormalizePositive17(data$mood_relaxed)
data$mood_satisfi <- NormalizePositive17(data$mood_satisfi)
data$mood_enthus <- NormalizePositive17(data$mood_enthus)
data$mood_cheerf <- NormalizePositive17(data$mood_cheerf)
data$mood_strong <- NormalizePositive17(data$mood_strong)
data$se_selflike <- NormalizePositive17(data$se_selflike)
data$se_handle <- NormalizePositive17(data$se_handle)

data$positive_moods <- data$mood_relaxed + data$mood_satisfi +
  data$mood_enthus + data$mood_cheerf + data$mood_strong +
  data$se_selflike + data$se_handle

data$positive_moods <- Normalize(data$positive_moods,
                                 min(data$positive_moods),
                                 max(data$positive_moods))

# negative_moods
data$mood_irritat <- NormalizeNegative17(data$mood_irritat)
data$mood_suspic <- NormalizeNegative17(data$mood_suspic)
data$mood_doubt <- NormalizeNegative17(data$mood_doubt)
data$se_ashamed <- NormalizeNegative17(data$se_ashamed)
data$se_selfdoub <- NormalizeNegative17(data$se_selfdoub)

data$negative_moods <- data$mood_irritat + data$mood_suspic +
  data$mood_doubt + data$se_ashamed + data$se_selfdoub

data$negative_moods <- Normalize(data$negative_moods,
                                 min(data$negative_moods),
                                 max(data$negative_moods))

# depressive_moods
data$mood_down <- NormalizeNegative33(data$mood_down)
data$mood_lonely <- NormalizeNegative33(data$mood_lonely)
data$mood_anxious <- NormalizeNegative33(data$mood_anxious)
data$mood_guilty <- NormalizeNegative33(data$mood_guilty)

data$depressive_moods <- data$mood_down + data$mood_lonely +
  data$mood_anxious + data$mood_guilty

data$depressive_moods <- Normalize(data$depressive_moods,
                                 min(data$depressive_moods),
                                 max(data$depressive_moods))
```

After categorizing three different accumulated moods (`positive_moods`, `negative_moods`, `depressive_moods`) I accumulated and normalized each. Afterwards I added them together to the `depression_factor` and normalized the value again.

```
data$depression_factor <- data$positive_moods + data$negative_moods +
  data$depressive_moods

data$depression_factor <- Normalize(data$depression_factor,
                                    min(data$depression_factor),
                                    max(data$depression_factor))
```

▶ What are the predictor variable(s) you will consider? Why would this make sense as a predictor?

I will chose all the variables that can be found in the next table.

name	description	scale
`phy_hungry`	I am hungry	-(1, 7)
`phy_tired`	I am tired	-(1, 7)
`phy_chanegable`	Physical conditions that can be changed. Variables `phy_hungry` and `phy_tired` combined.	-(-1, 1)
`phy_pain`	I am in pain	-(1, 7)
`phy_dizzy`	I feel dizzy	-(1, 7)
`phy_headache`	I have a headache	-(1, 7)
`phy_complain`	Physical conditions that can be described as complains. Variables `phy_pain`, `phy_dizzy` and `phy_headache` combined.	-(-1, 1)
`phy_physact`	From the last beep on wards I was physically active	+(1, 7)

I assume that the physical variables are suitable to be used as predictor variables. I separated them into three variables the changeable physical conditions (`phy_chanegable`), the physical complains (`phy_complain`) and if the patient was physical active (`phy_physact`). `phy_chanegable` and `phy_complain` will be accumulated in the same manner as the artificial mood variables.

I think it makes sense to use `phy_chanegable` as a predictor because being tired or hungry are feelings that constantly influences the mood. Just as an example, the definition of the word 'hangry' is 'irritable or angry because of hunger'. https://www.merriam-webster.com/dictionary/hangry And having a bad sleep can trigger a bad mood.
The `phy_complain` variable makes also sense to be included as a predictor. Physical complains are obviously mood influential.

Doing sport (`phy_physact`) is also mood influential and thus also suitable to be used as predictor.

At the end I could also account all my predictors as cause variables if I understood the dif-

ference between predictors and cause right. Nevertheless, I chose them to be a predictor.

```
# phy_chanegable
data$phy_hungry <- NormalizeNegative17(data$phy_hungry)
data$phy_tired <- NormalizeNegative17(data$phy_tired)

data$phy_chanegable <- data$phy_hungry + data$phy_tired

data$phy_chanegable <- Normalize(data$phy_chanegable,
                                 min(data$phy_chanegable),
                                 max(data$phy_chanegable))

# phy_complain
data$phy_pain <- NormalizeNegative17(data$phy_pain)
data$phy_dizzy <- NormalizeNegative17(data$phy_dizzy)
data$phy_headache <- NormalizeNegative17(data$phy_headache)

data$phy_complain <- data$phy_pain + data$phy_dizzy + data$phy_headache

data$phy_complain <- Normalize(data$phy_complain,
                               min(data$phy_complain),
                               max(data$phy_complain))

# phy_physact
data$phy_physact <- NormalizePositive17(data$phy_physact)
```

▶ What are the cause(s) you will consider? Why would this make sense as a cause?

A human being is complex. The surrounding factors influence whatever happens within a person. Concluding, I assume all interactions between a human being and its environment must strongly influence its mood. Therefore, I chose social factor as my cause variable.

name	description	scale
`soc_pleasant`	I find this company pleasant.	+(1, 7)
`soc_prefalone`	I prefer to be alone.	-(1, 7)
`soc_enjoy_alone`	I enjoy to be alone.	+(1, 7)
`soc_prefcomp`	I prefer being in company.	-(1, 7)
`soc_factor`	A fact or that approximates the patients social needs. Variables `soc_pleasant`, `soc_prefalone`, `soc_enjoy_alone` and `soc_prefcomp` combined.	+(-1, 1)

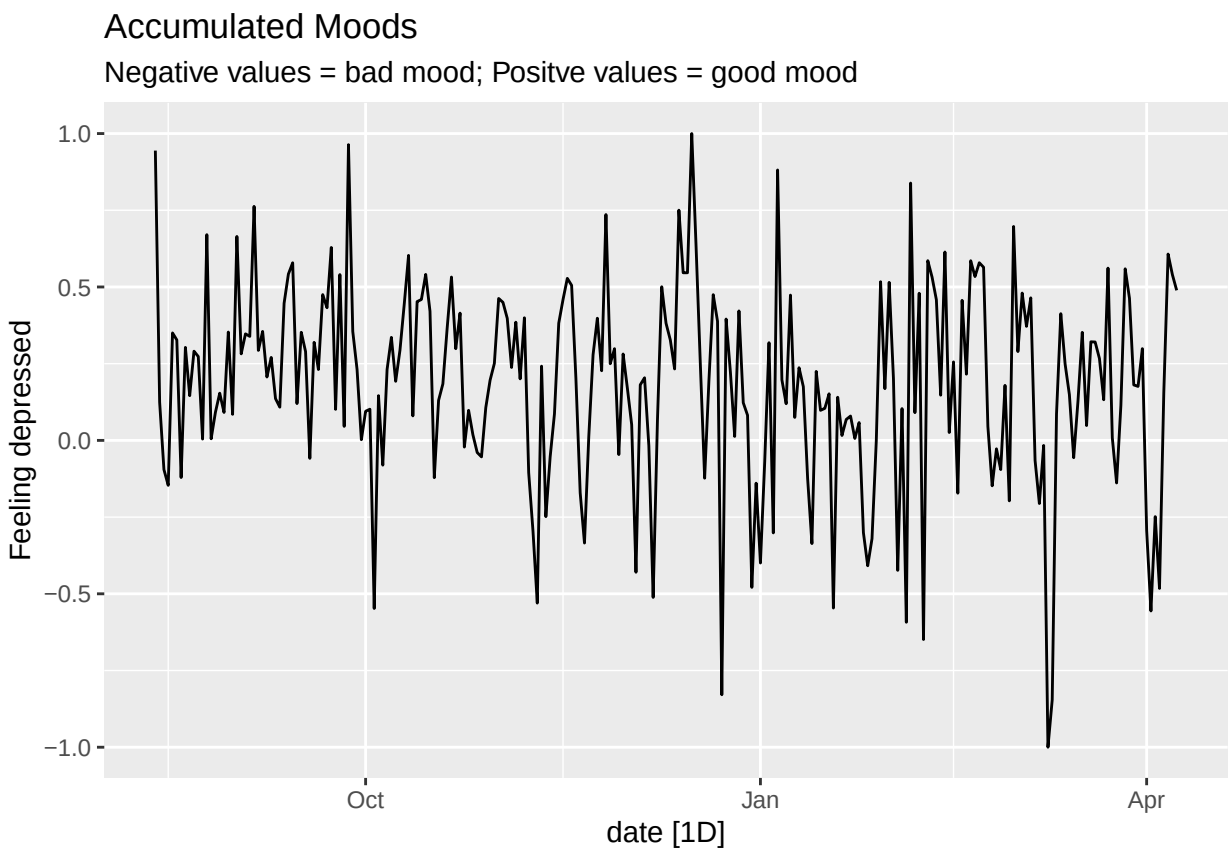
```
# soc_factor
data$soc_pleasant     <- NormalizePositive17(data$soc_pleasant)
data$soc_prefalone    <- NormalizeNegative17(data$soc_prefalone)
data$soc_enjoy_alone  <- NormalizePositive17(data$soc_enjoy_alone)
```

```
data$soc_prefcomp     <- NormalizeNegative17(data$soc_prefcomp)

data$soc_factor       <- data$soc_pleasant + data$soc_prefalone +
  data$soc_enjoy_alone + data$soc_prefcomp

data$soc_factor       <- Normalize(data$soc_factor,
                                   min(data$soc_factor),
                                   max(data$soc_factor))
```

1.2. Visualize your data

▶ Create a sequence plot of the data with the function autoplot(). Interpret the results.

```
autoplot(data, depression_factor) +
  labs(title = "Accumulated Moods",
       subtitle = "Negative values = bad mood; Positve values = good mood",
       y = "Feeling depressed")
```



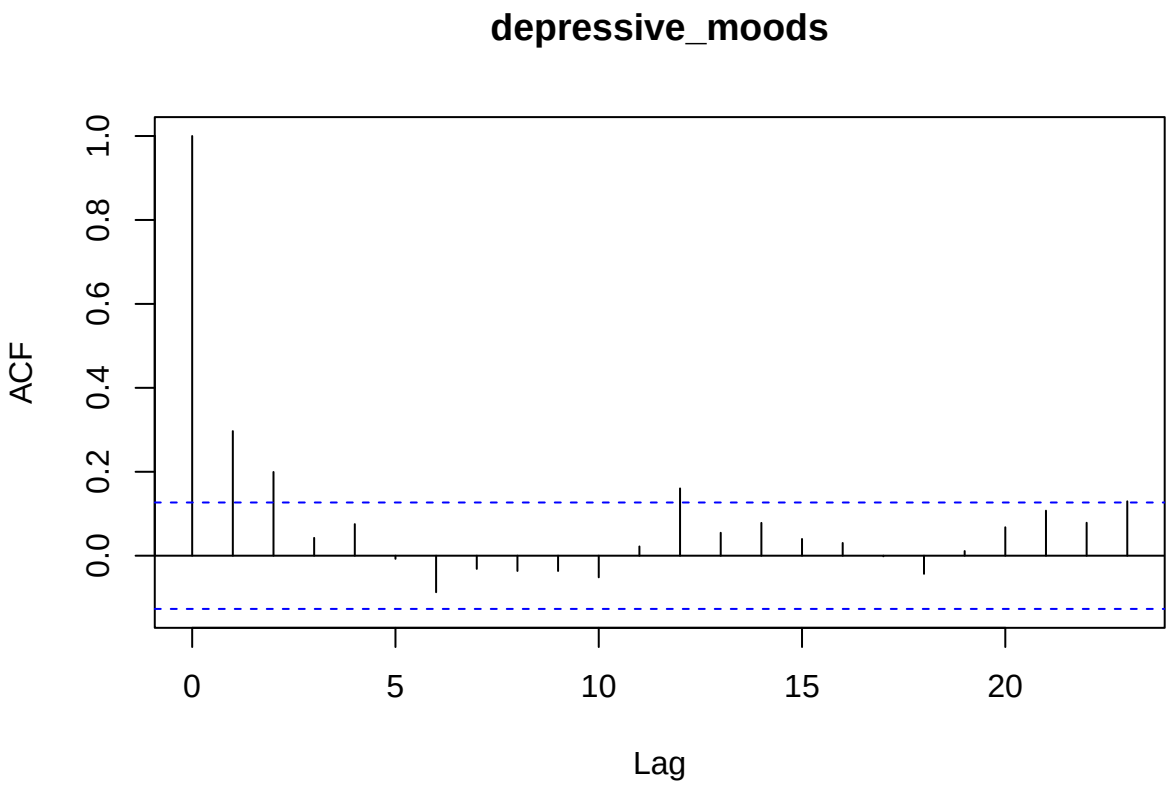
- We can see that there is almost always a fluctuation between feeling and not feeling depressed. It seems like there are few or no existing stable phases.
- On almost every good day (feeling not depressed) a bad day (feeling depressed) is the follow up. The most extreme situations were in October and just before the

January.

- We can also observe a more stable period at the beginning of the time plot regarding the score of depression factor.
- It also seems that the spikes are getting more frequent towards the end of th plot and more towards a depressive mood. This might be due to the reduction of medication.
- The plot indicates that it is near random, so to say like white noise but it seems like there is some structure hidden.

▶ Plot the autocorrelation function with the function acf(). Interpret the results.

```
acf(data[34])
```



Here we see that the autocorrelations at lags 1, 2, 3 and 12 are significantly larger than zero; this implies there are sequential dependencies in these data. which makes z_t somewhat predictable from preceding observations (e.g., z_t-1).

There are no signs of a trend or a unit root process, or of a changing variance; hence, the series seem stationary.

▶ Based on (basic) content knowledge about the variable, and these visualizations, is there reason to assume the data are non-stationary and/or that there is a seasonal component?

I am pretty certain that the data is stationary. Concerning the time plot and the autocorrelation I assume that we will find a weak seasonal component within this data.

2. Forecasting

2.1. SARIMA modeling

▶ Perform the Dickey-Fuller test. What is your conclusion?

```
adf.test(data$depression_factor)
```

```
## Warning in adf.test(data$depression_factor): p-value smaller than printed p-
## value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  data$depression_factor
## Dickey-Fuller = -6.4163, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

The H0 (null=hypothesis) of this test is that there is a unit root process; Here the p-value obtained is very small, which would make us reject H0; hence, we conclude there is no evidence for a unit root process (i.e., there is no need to difference these data).

▶ Fit an (S)ARIMA model to the data; what is the order of the model that was selected?

```
fit.data <- model(data, ARIMA(depression_factor))
report(fit.data)
```

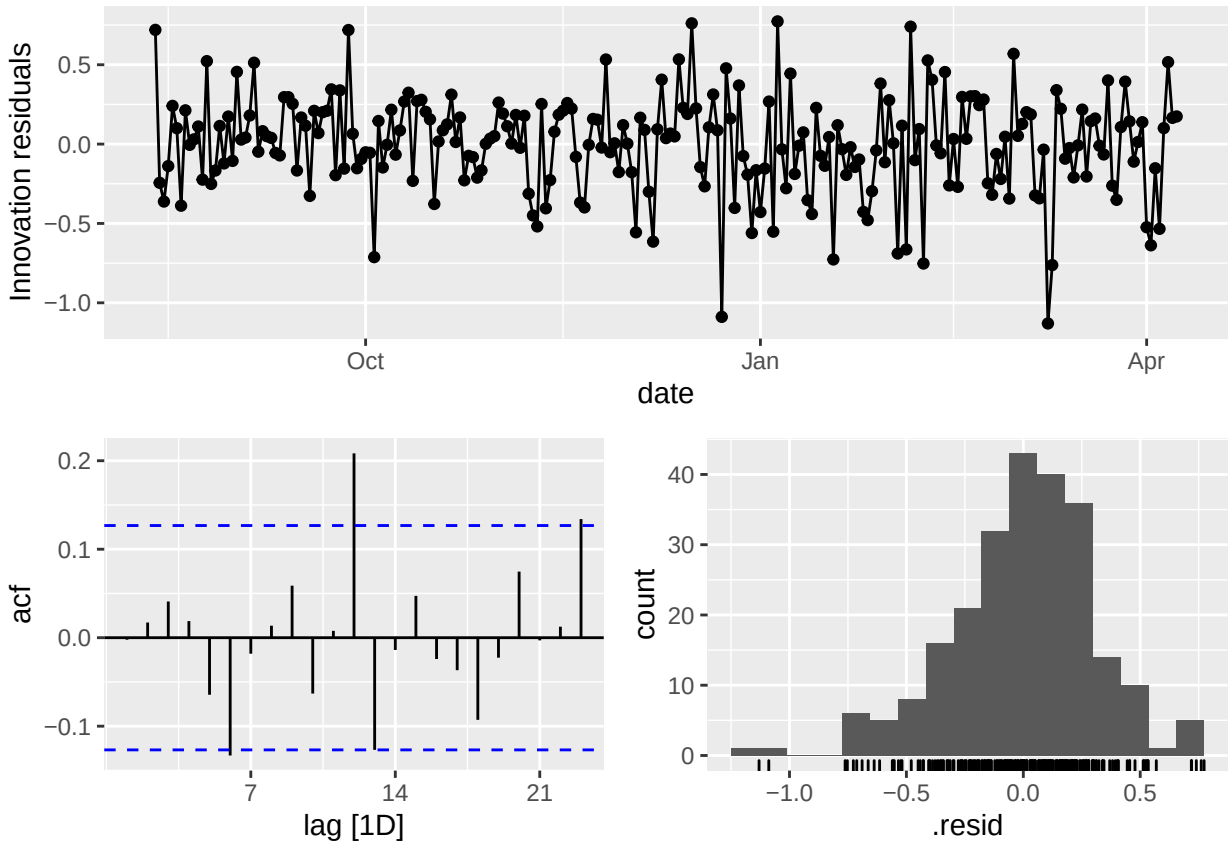
```
## Series: depression_factor
## Model: ARIMA(1,0,2) w/ mean
##
## Coefficients:
##           ar1     ma1     ma2  constant
##       -0.3606  0.6050  0.3082    0.2506
## s.e.   0.2392  0.2314  0.0693    0.0381
##
## sigma^2 estimated as 0.09693:  log likelihood=-58.32
## AIC=126.63   AICc=126.89   BIC=144.01
```

This shows we get an ARIMA(1,0,2)

$$yt = 0.2506 - B0.36y + B0.605e + B^2 0.308e + e$$

▶ Check the residuals of the model using the function gg_tsresiduals(). What is your conclusion?

```
gg_tsresiduals(fit.data)
```



Looking at the innovation residuals it is visible that there are some peaks frequently emerging within the timeline. Nevertheless, the mean seems to be stable but the variation has a slight increase over time.

Concerning the autocorrelation we see one lag #12 with a value above the blue dotted line for significant value threshold. Whereas three other lags #6, #13 and #23 are just touching the blue dotted line.

Regarding the histogram there is a little shift to the right concerning the distribution.

In conclusion I can imagine that there might be transformations or different models that lead to improvements. Nevertheless I assume that the current situation is within a feasible spectrum to be able to proceed without any transformation or change of model.

2.2. Dynamic regression

▶ Include the predictor in an dynamic regression model (i.e., allow for (S)ARIMA residuals); what is the effect of the predictor?

```
fit.data.p <- model(data, ARIMA(depression_factor ~ phy_complain +
                                phy_chanegable +
                                phy_physact +
                                soc_factor))
report(fit.data.p)
```

```
## Series: depression_factor
## Model: LM w/ ARIMA(0,1,3) errors
##
## Coefficients:
##           ma1      ma2      ma3  phy_complain  phy_chanegable  phy_physact
##       -0.6894  -0.0144  -0.2566        0.1631          0.1312       0.3156
## s.e.   0.0627   0.0839   0.0638        0.0538          0.0507       0.1231
##       soc_factor
##           0.2853
## s.e.      0.0469
##
## sigma^2 estimated as 0.06739:  log likelihood=-14.35
## AIC=44.71   AICc=45.34   BIC=72.49
```

We see that `soc_factor`, `phy_complain`, `phy_chanegable` and `phy_physact` are positive predictors for the depression factor.

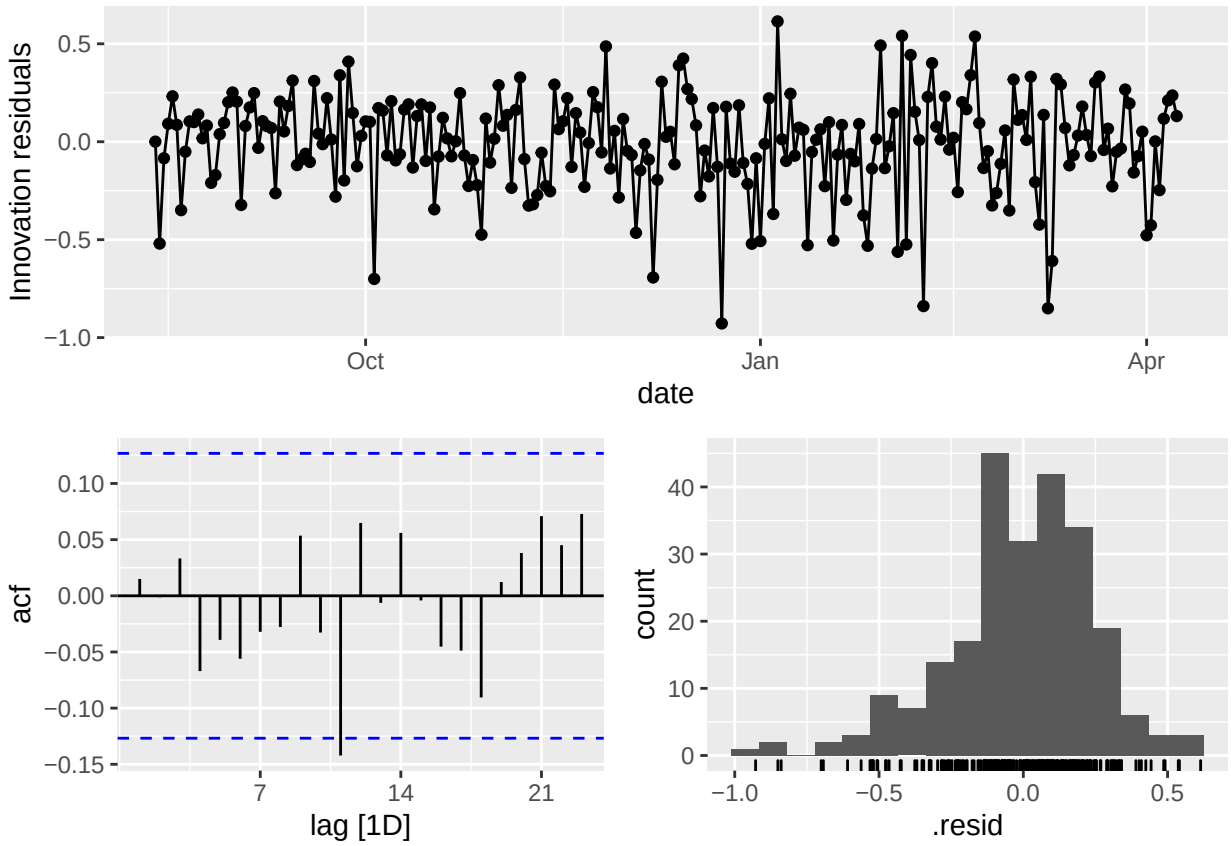
▶ What order is the (S)ARIMA model for the residuals?

The residuals are estimated to be an ARIMA(0,1,3) process;

One remarkable thing is that the ARIMA model including predictors is now without an autoregressive model component and involved one degree of first differencing.

▶ Check the residuals of the model using the function gg_tsresiduals(). What is your conclusion?

```
gg_tsresiduals(fit.data.p)
```



The sequence plot indicates to be as stationary as from the previous model. Mean and variance seem to keep constant over time. Also again a slight increase of variance towards the end.

Concerning the autocorrelation we see not a single lag exceeding the significant threshold.

Regarding the histogram the distribution seems almost perfectly balanced.

In conclusion here is definitely no transformation or adaption of the model required.

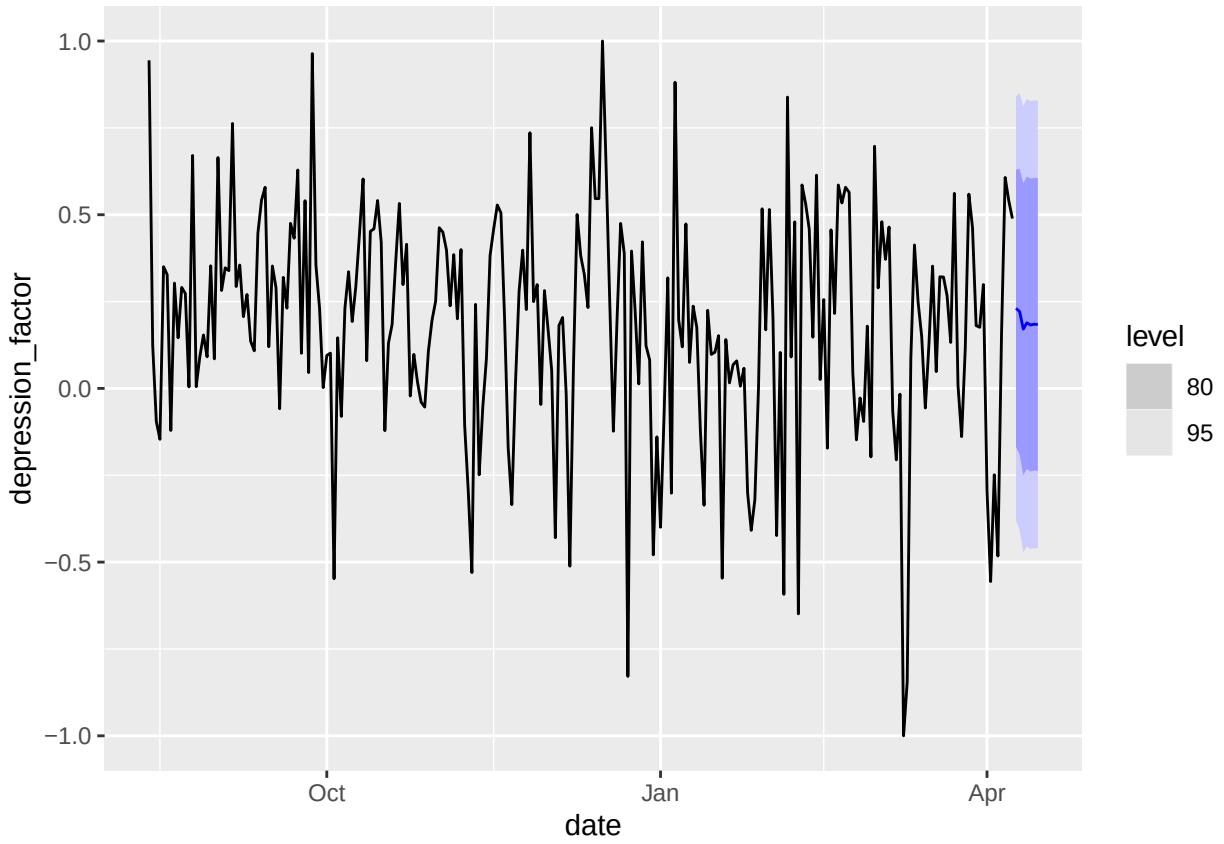
2.3. Forecasts

▶ Choose a forecasting horizon, and indicate why this is a reasonable and interesting horizon to consider.

The forecasting horizon I chose is 1 week. Regarding the high fluctuation of the patents mood predicting one week seems already difficult. Also considering that the data does not even extend over one year. Also forecasting over one week seems to be not reasonable. And in the end one week seems to be an interesting horizon for a patient that has clinical depression.

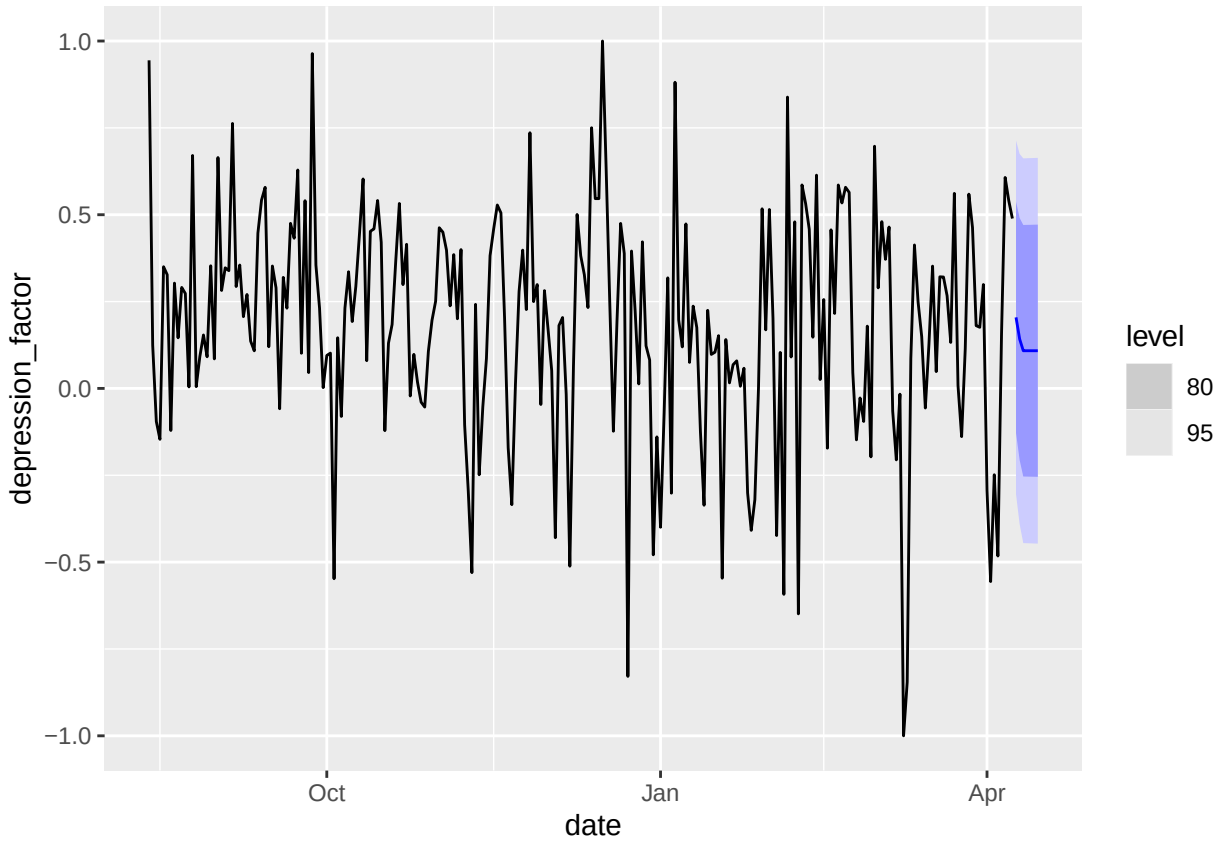
▶ Create forecasts based on the model without the predictor and plot these.

```
fit.data %>% forecast(h=7) %>% autoplot(data)
```



▶ Create forecasts based on the model with the predictor and plot these.

```
new_data_depr <- new_data(data, 7) %>%
  mutate(soc_factor = mean(data$soc_factor),
         phy_complain = mean(data$phy_complain),
         phy_chanegable = mean(data$phy_chanegable),
         phy_physact = mean(data$phy_physact))

forecast(fit.data.p, new_data = new_data_depr) %>%
  autoplot(data)
```



▶ Compare the plots of both forecasts (visually), and discuss how they are similar and/or different.

Comparing both plots lead to the conclusion there is very little difference regarding the line progression. Both visualizations are predicting that the mood was from the beginning of the forecast a little over the average and therefore there is the decline towards a slightly lower level of the depression factor. The first model showed also a little bounce in a negative direction and then spring back to the level where the depression factor keeps stable.

There is additionally one difference. The uncertainty intervals of the second model is more dense then the first model.

3. Causal Modeling

▶ Formulate a causal research question(s) involving the time series variable(s) you have measured.

Does positive social experiences (soc_factor) lead to a better mood (depression_factor) in regards to the patient?

▶ Which method we learned about in class (Granger causal approaches, interrupted time series, synthetic controls) is most appropriate to answer your research question using the

data you have available? Why?

In my case my data is not perfectly suitable to apply interrupted time series nor synthetic controls at all. There would be an possibility with the medication. However, I am afraid that by using one of the two methods mentioned, a result will come out, which in the end would be a confidence interval within the values and at most an indication in the direction of less fluctuation than anything else.

Concluding I choose the Granger causal approach to figure out if there is a Granger Causality between social experience and the mood.

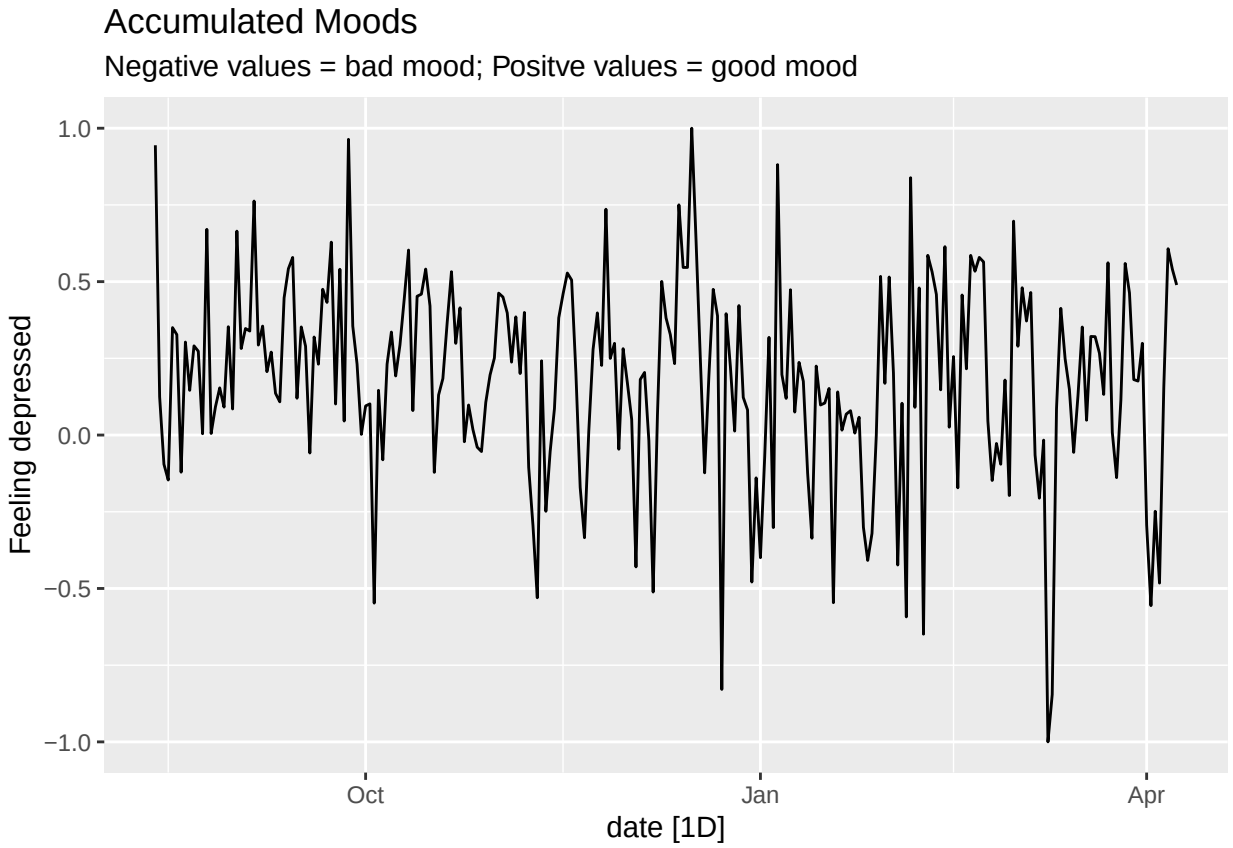
3.2 Analysis

Depending on the choice you made above, follow the questions outlined in 3.2a, 3.2b or 3.2c. If you chose a Granger causal analysis, it is sufficient to assess Granger causality in one direction only: you may evaluate a reciprocal causal relationship, but then answer each question below for both models.

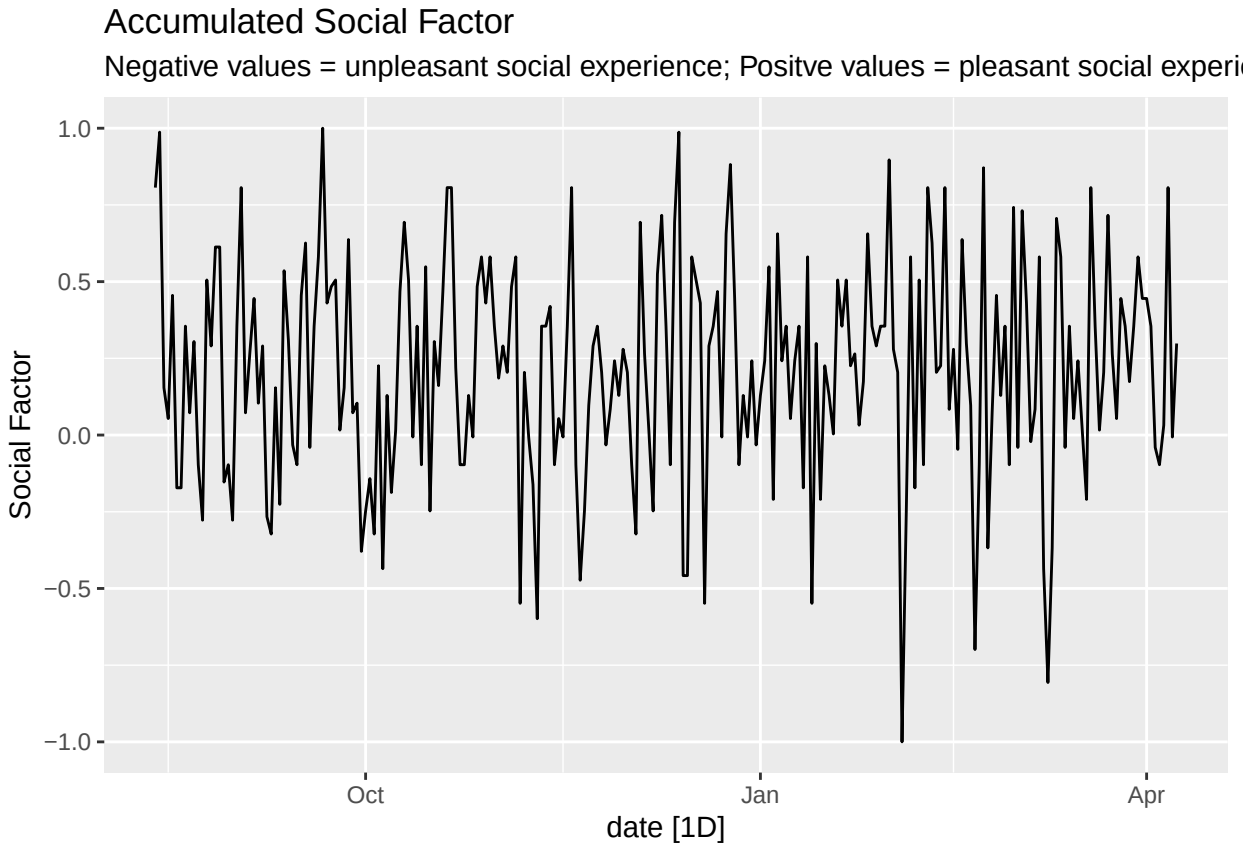
3.2a Granger Causal analysis

▶ Visualize your putative cause variable(s) X and outcome variables Y.

```
autoplot(data, depression_factor) +
  labs(title = "Accumulated Moods",
       subtitle = "Negative values = bad mood; Positve values = good mood",
       y = "Feeling depressed")
```




```
autoplot(data, soc_factor) +
  labs(title = "Accumulated Social Factor",
       subtitle = "Negative values = unpleasant social experience; Positve values = plea
       y = "Social Factor")
```




```
adf.test(data$soc_factor)
```

```
## Warning in adf.test(data$soc_factor): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  data$soc_factor
## Dickey-Fuller = -6.3211, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

Regarding dicky-fuller the soc_factor is also a stationary time series.

▶ Train an appropriate ARIMA model on your outcome variable(s) Y, ignoring the putative cause variable(s) (X) but including, if appropriate, any additional covariates. If using the same model as fit in part 2, briefly describe that model again here.

```
fit.data.soc <- model(data, ARIMA(depression_factor ~ soc_factor))
report(fit.data.soc)
```

```
## Series: depression_factor
## Model: LM w/ ARIMA(0,1,3)(1,0,0)[7] errors
##
## Coefficients:
##          ma1     ma2      ma3     sar1  soc_factor
```

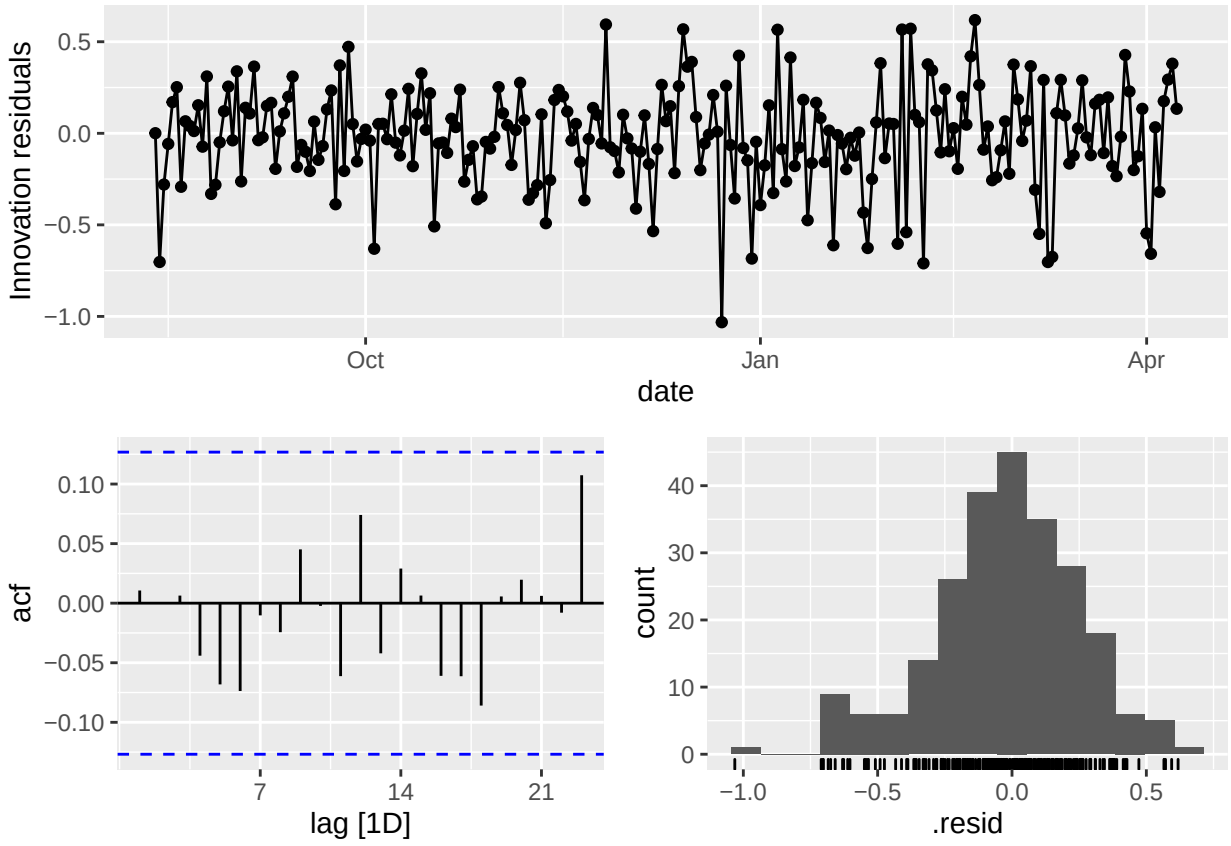
```
##       -0.7312  0.0131  -0.2557  -0.0284      0.3836
## s.e.   0.0635  0.0811   0.0620   0.0679      0.0474
##
## sigma^2 estimated as 0.07768:  log likelihood=-32.52
## AIC=77.03   AICc=77.4   BIC=97.87
```

ARIMA(p,d,q) = ARIMA(0,1,3)(1,0,0)[7] Autoregressive model 0 number of differencing 1 Movingaverage model 3

Due to the seasonal AR(1) part, the model has to regress the differenced score z_t on (z_t-7).

In short the appendix means (1,0,0)[7] is regaring the seasonal part and between brackets the amount of lag between seasons.

```
gg_tsresiduals(fit.data.soc)
```



The sequence plot indicates to be perfectly stationary.

Concerning the autocorrelation we see not a single lag exceeding the significant threshold.

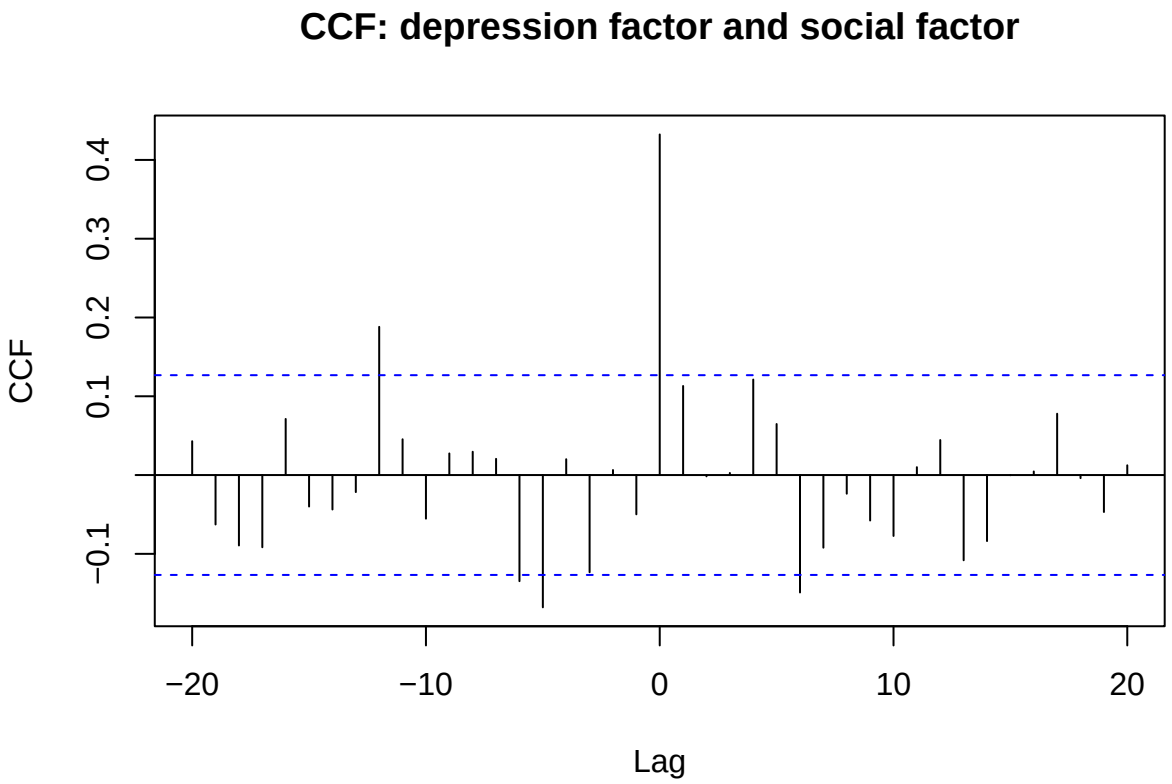
Regarding the histogram the distribution seems almost perfectly balanced.

In conclusion here is definitely no transformation or adaption of the model required.

▶ Justify what range of lags to consider for the lagged predictor(s). Use the CCF, but you

may also justify this based on domain knowledge or substantive theory.

```
ccf(data$depression_factor, data$soc_factor,
    main = "CCF: depression factor and social factor",
    ylab = "CCF")
```



By applying cross-correlation function (CCF) an idea can be generated on which range of lags might be useful by encoding the lagged correlation between X and Y. By moving to the left (starting lag 0) we find three significant correlations at lag -5, -6 and -12 and by moving to the rifght we find only one significant lag at position 6. Regrading also all my information gathered by this plot it makes sense to me to investigate until lag 7. I am not sure if it is feasible to assume that the influence of a social factor correlates on regular basis over 2 weeks.

▶ Investigate whether adding your lagged "cause" variables (X) improve the prediction of your effect variable(s) Y. Use model selection based on information criteria. Describe your final chosen model.

```
fit <- data %>%
  # Restrict data so models use same fitting period
  mutate(depression_factor = c(NA, NA, NA, NA, NA, NA, NA, depression_factor[8:239])) %>
  # Estimate models
  model(
    indep = ARIMA(depression_factor),
    lag1 = ARIMA(depression_factor~ lag(data$soc_factor)),
    lag2 = ARIMA(depression_factor~ lag(data$soc_factor) +
```

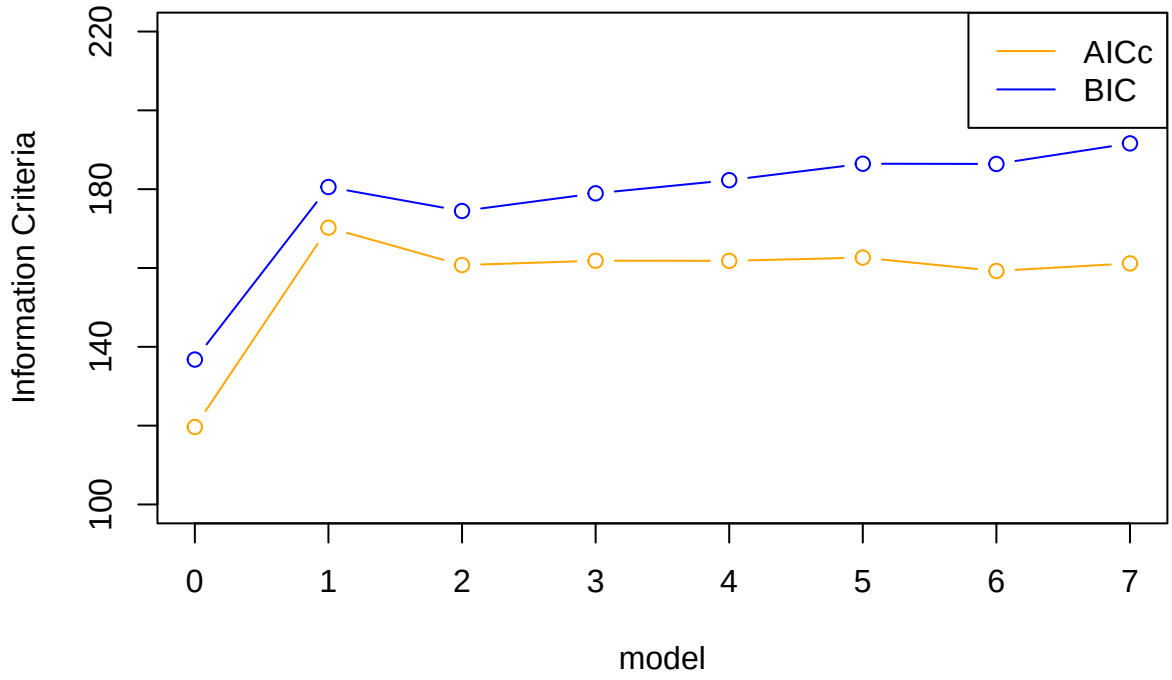
```
                    lag(data$soc_factor,2)),
    lag3 = ARIMA(depression_factor~ lag(data$soc_factor) +
                    lag(data$soc_factor,2) +
                    lag(data$soc_factor,3)),
    lag4 = ARIMA(depression_factor~ lag(data$soc_factor) +
                    lag(data$soc_factor,2) +
                    lag(data$soc_factor,3) +
                    lag(data$soc_factor,4)),
    lag5 = ARIMA(depression_factor~ lag(data$soc_factor) +
                    lag(data$soc_factor,2) +
                    lag(data$soc_factor,3) +
                    lag(data$soc_factor,4) +
                    lag(data$soc_factor,5)),
    lag6 = ARIMA(depression_factor~ lag(data$soc_factor) +
                    lag(data$soc_factor,2) +
                    lag(data$soc_factor,3) +
                    lag(data$soc_factor,4) +
                    lag(data$soc_factor,5) +
                    lag(data$soc_factor,6)),
    lag7 = ARIMA(depression_factor~ lag(data$soc_factor) +
                    lag(data$soc_factor,2) +
                    lag(data$soc_factor,3) +
                    lag(data$soc_factor,4) +
                    lag(data$soc_factor,5) +
                    lag(data$soc_factor,6) +
                    lag(data$soc_factor,7)),
)
```

```
# take a look at the result
glance(fit)
```

```
## # A tibble: 8 x 8
##   .model sigma2 log_lik   AIC  AICc   BIC ar_roots  ma_roots
##   <chr>   <dbl>   <dbl> <dbl> <dbl> <dbl> <list>    <list>
## 1 indep  0.0925   -54.7  119.  120.  137. <cpl [3]> <cpl [0]>
## 2 lag1   0.116    -82.1  170.  170.  181. <cpl [1]> <cpl [0]>
## 3 lag2   0.111    -76.3  161.  161.  174. <cpl [1]> <cpl [0]>
## 4 lag3   0.111    -75.8  162.  162.  179. <cpl [1]> <cpl [0]>
## 5 lag4   0.111    -74.7  161.  162.  182. <cpl [1]> <cpl [0]>
## 6 lag5   0.111    -74.1  162.  163.  186. <cpl [1]> <cpl [0]>
## 7 lag6   0.108    -71.3  159.  159.  186. <cpl [1]> <cpl [0]>
## 8 lag7   0.109    -71.2  160.  161.  192. <cpl [1]> <cpl [0]>
```

```
plot(seq(0,7), glance(fit)$AICc,
     col = "orange", type = "b",
```

```
     ylab = "Information Criteria", xlab = "model",
     ylim = c(100, 220))
lines(seq(0,7), glance(fit)$BIC, col = "blue", type = "b")
legend("topright", c("AICc","BIC"), col = c("orange","blue"), lty = 1)
```



3.3 Conclusion and critical reflection

▶ Based on the result of your analysis, how would you answer your causal research question?

The basic idea behind Granger Causality in this example is the following. Can the patient's overall depression quotient be better predicted by taking into account past values based on social events. Disappointingly, the AIC and BIC values do not show any indication that the prediction is improved by taking the social component into account. Therefore, I have to conclude that in this example all signs show that the social factor does not Granger-cause the depression factor.

▶ Making causal conclusions on the basis of your analysis is reliant on a number of assumptions. Pick a single assumption that is necessary in the approach you chose. Discuss the plausibility and possible threats to the validity of this assumption in your specific setting (< 75 words)

To make a statement as strong as "this is not a Causal Reason", one would have to be sure that all the other variables are observed and controlled for or at least not misleadingly influential - no confounders. Especially in a such a complex domain the human brain, so many hidden processes can be involved. Nevertheless, if I wanted to be confident, I could have assumed Sufficiency.