Individual assignment SBD

Linda Ilic

ADS, 2021-2022

Contents

0. Prepare **1**

1. General **3**

1.1. Describe your data 4

1.2. Visualize your data 4

2. Forecasting **6**

2.1. SARIMA modeling 6

2.2. Dynamic regression 8

2.3. Forecasts 10

3. Causal Modeling **11**

3.2 Analysis 12

3.2a Granger Causal analysis 12

3.3 Conclusion and critical reflection 18

0. Prepare

▶ Load the R-packages you will use.

```
library(gridExtra)
library(fpp3)
library(tseries)
```

```
library(expsmooth)
library(data.table)
library(tibble)
library(tsibble)
library(lubridate)
library(dplyr)
library(openair)
library(rts)
```

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

I am working with the Peter de Groot dataset. The participant monitored his momentary experiential states over the course of gradual discontinuation (from 150 to 0 mg over 8 weeks) of his antidepressant and collected reports of momentary states up to 10 times a day (varies per variable) over a period of 239 days. As a first step I chose to average the intraday values so I have only one value per day.

```
#----Loading data------

df <- read.csv("ESMdata.csv")
head(df)


#ADDITIONAL STEPS

colSums(is.na(df)) # check for missing data per colomn


#------ Preprocessing of data ---------

#take the average of the intraday values so that there is only 1 daily value per colum
# do this for all variables that you consider
df_day <- df %>%
  group_by(dayno) %>%
  summarise(
    mood_down_avg=(sum(mood_down)/length(dayno)),
    concentrat_avg=(sum(concentrat)/length(dayno)),
    mood_anxious_avg=(sum(mood_anxious)/length(dayno)),
    se_selfdoub_avg=(sum(se_selfdoub)/length(dayno)),
    phy_tired_avg=(sum(phy_tired)/length(dayno)),
    mood_irritat_avg=(sum(mood_irritat)/length(dayno)),
    mood_lonely_avg=(sum(mood_lonely)/length(dayno)),
    pat_agitate_avg=(sum(pat_agitate)/length(dayno)),
    date=date[1]
    )
```

```
df_day %>%
  mutate(date = dmy(date)) %>% # transform date into correct formate and merge with tim
  as_tsibble(index = date) %>%  # transform to tsibble object
  dplyr::select( # select variables/columns %>%
    date,
    dayno,
    mood_down_avg,
    concentrat_avg,
    se_selfdoub_avg,
    mood_anxious_avg,
    phy_tired_avg,
    mood_irritat_avg,
    mood_lonely_avg,
    pat_agitate_avg)-> df_ts


# check if data is complete
# complete data
has_gaps(df_ts) # check for implicit gaps
df_ts_clean<-tsibble::fill_gaps(df_ts) # make implicit missing data explicit
colSums(is.na(df_ts)) #check for missing data for each column

df_ts_clean %>% replace_na(list(dayno = 350)) -> df_ts_clean # replace one NA in dayno
df_ts_clean<-na.locf(df_ts_clean) # impute rest of NAs

has_gaps(df_ts_clean) # check for implicit gaps again
colSums(is.na(df_ts_clean)) #check again for NAs
```

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.

The tsibble object contains 9 variables (+ date) with 239 rows (observations).

```
print(df_ts_clean)
```

```
## # A tsibble: 239 x 10 [1D]
##    date       dayno mood_down_avg concentrat_avg se_selfdoub_avg
```

```
##    <date>     <dbl>         <dbl>          <dbl>           <dbl>
##  1 2012-08-13   226        -1               150            1
##  2 2012-08-14   227         0               150            1.8
##  3 2012-08-15   228        -0.111           150            2.11
##  4 2012-08-16   229         0.333           150            2.33
##  5 2012-08-17   230         0.222           150            1.78
##  6 2012-08-18   231         0               150            1.67
##  7 2012-08-19   232         0.833           150            2.17
##  8 2012-08-20   233         0               150            2
##  9 2012-08-21   234         0.25            150            1.88
## 10 2012-08-22   235         0.222           150            1.78
## # ... with 229 more rows, and 5 more variables: mood_anxious_avg <dbl>,
## #   phy_tired_avg <dbl>, mood_irritat_avg <dbl>, mood_lonely_avg <dbl>,
## #   pat_agitate_avg <dbl>
```

1.1. Describe your data

Start with answering the following questions:

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

My outcome variable is mood_down. It measures the statement “I feel down” on a scala from -3 (not) until 3 (very) (see Codebook) daily and does so for 239 days.

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

I will be considering different mood states (“feeling lonely”, “feeling anxious”) and selfdoubt as predictors. As I feel that these might contribute to feeling “down”.

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

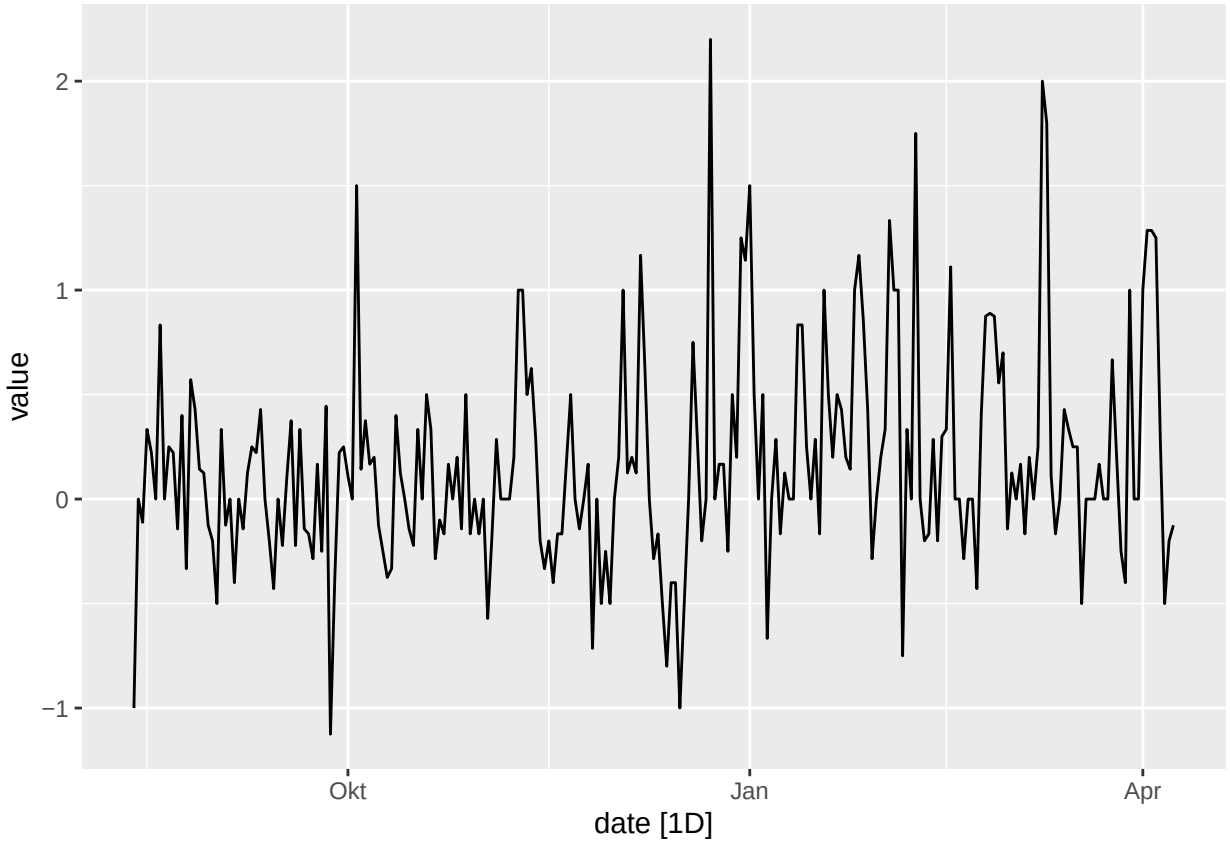
My x (cause) variable is phy_tired. I think that being physically tired and exhausted might cause a person to feel “down”.

1.2. Visualize your data

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

The data seems to be stable and does not show an extreme trend or seasonality although the average does seem to increase a bit at the end.

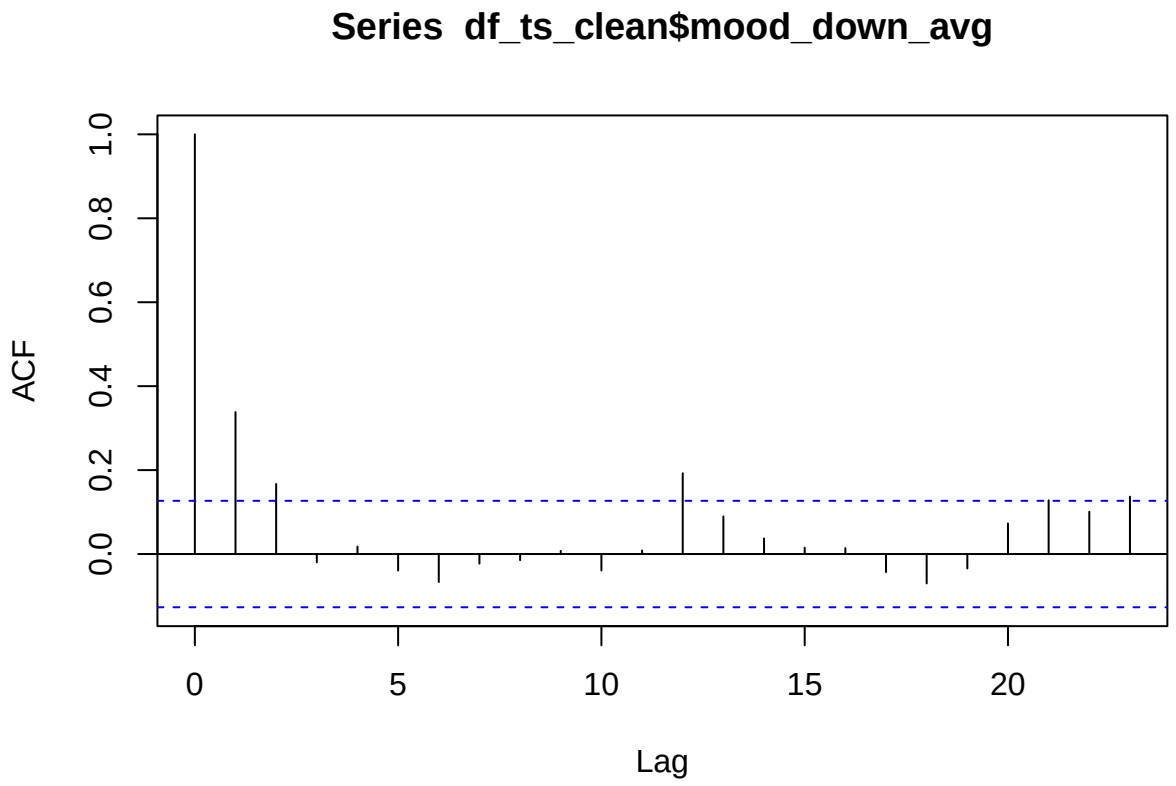
```
# autoplot for outcome variable

autoplot(df_ts_clean, vars(mood_down_avg))
```



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

As expected the correlation at lag 0 is 1. Additionally, there seems to be significant autocorrelation present at lag 1, 2, 12 and 23. Thus there is some autocorrelation present.

```
# autocorrelation of outcome variable
acf(df_ts_clean$mood_down_avg)
```



▶ 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?

There is no strong pattern in the autocorrelation or the plotted data so at the moment I do not see a reason to assume non-stationarity.

2. Forecasting

2.1. SARIMA modeling

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

As the p-value is smaller than 0.01 the H0 (that there is a unit root process) is rejected and the alternative Hypothesis is accepted. Thus the Dickley-Fuller test indicates that the time series is stationary which means that there is no cycle/trend in the data.

```
adf.test(df_ts_clean$mood_down_avg)
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  df_ts_clean$mood_down_avg
## Dickey-Fuller = -6.3345, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

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

Based on the data an AR(1)I(1)MA(3) model was selected. Interestingly, this seems to contradict the results from the Dickey-Fuller test as the selected model is differenced once. This might be due to the different approaches of the two methods.

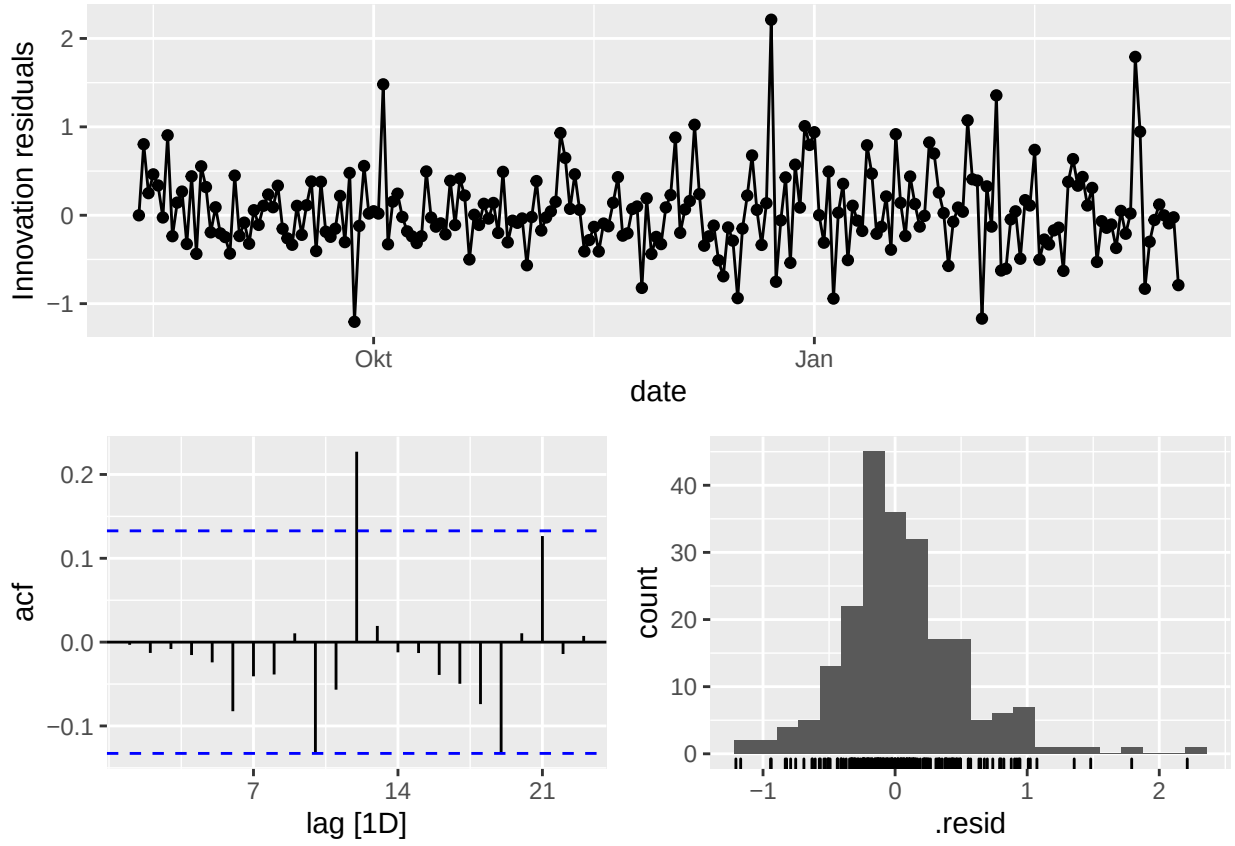
```
#split data into training data(for the fitting) and test data(for the forecasting)
library(forecast)
train <- head(df_ts_clean, 218)#use first 218 time points for the fitting
test <- tail(df_ts_clean, 21 ) # use last 21 time points for forecasting


fit1 <- train %>% model(ARIMA(mood_down_avg)) #fit without using predictors
report(fit1)
```

```
## Series: mood_down_avg
## Model: ARIMA(1,1,3)
##
## Coefficients:
##          ar1      ma1      ma2      ma3
##       -0.3919  -0.2870  -0.3676  -0.2965
## s.e.   0.2448   0.2359   0.1865   0.0781
##
## sigma^2 estimated as 0.2246:  log likelihood=-145.1
## AIC=300.21   AICc=300.49   BIC=317.11
```

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

The plotted residuals seem stable but there are three high peaks that repeat in similar intervals. The ACF does show some significant autocorrelation at lag 12 which indicates that the model should be improved. However, the residuals seem to be distributed normally to some degree.

```
gg_tsresiduals(fit1)
```




```
adf.test(augment(fit1)$.resid)
```

```
## Warning in adf.test(augment(fit1)$.resid): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  augment(fit1)$.resid
## Dickey-Fuller = -6.5286, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

2.2. Dynamic regression

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

I used several predictors as I thought that they might explain the data better. All three predictors have different effect values. Mood_lonely has the strongest effect, self doubt and anxious mood have significantly smaller effects. Also, we can see that the selected model changed to a seasonal ARIMA model with a seasonal period of 7 days i.e. there is an weekly effect.

```
#fitting using 3 predictors
fit2 <- train %>% model(ARIMA(mood_down_avg ~  se_selfdoub_avg  + mood_anxious_avg + moo
report(fit2)
```

```
## Series: mood_down_avg
## Model: LM w/ ARIMA(1,1,1)(2,0,0)[7] errors
##
## Coefficients:
##          ar1      ma1     sar1     sar2  se_selfdoub_avg  mood_anxious_avg
##       0.3086  -0.9530  -0.1220  -0.0682           0.3015           -0.0698
## s.e.  0.0727   0.0259   0.0711   0.0727           0.0361            0.1032
##       mood_lonely_avg
##                0.9411
## s.e.           0.0764
##
## sigma^2 estimated as 0.0495:  log likelihood=20.69
## AIC=-25.38   AICc=-24.69   BIC=1.66
```

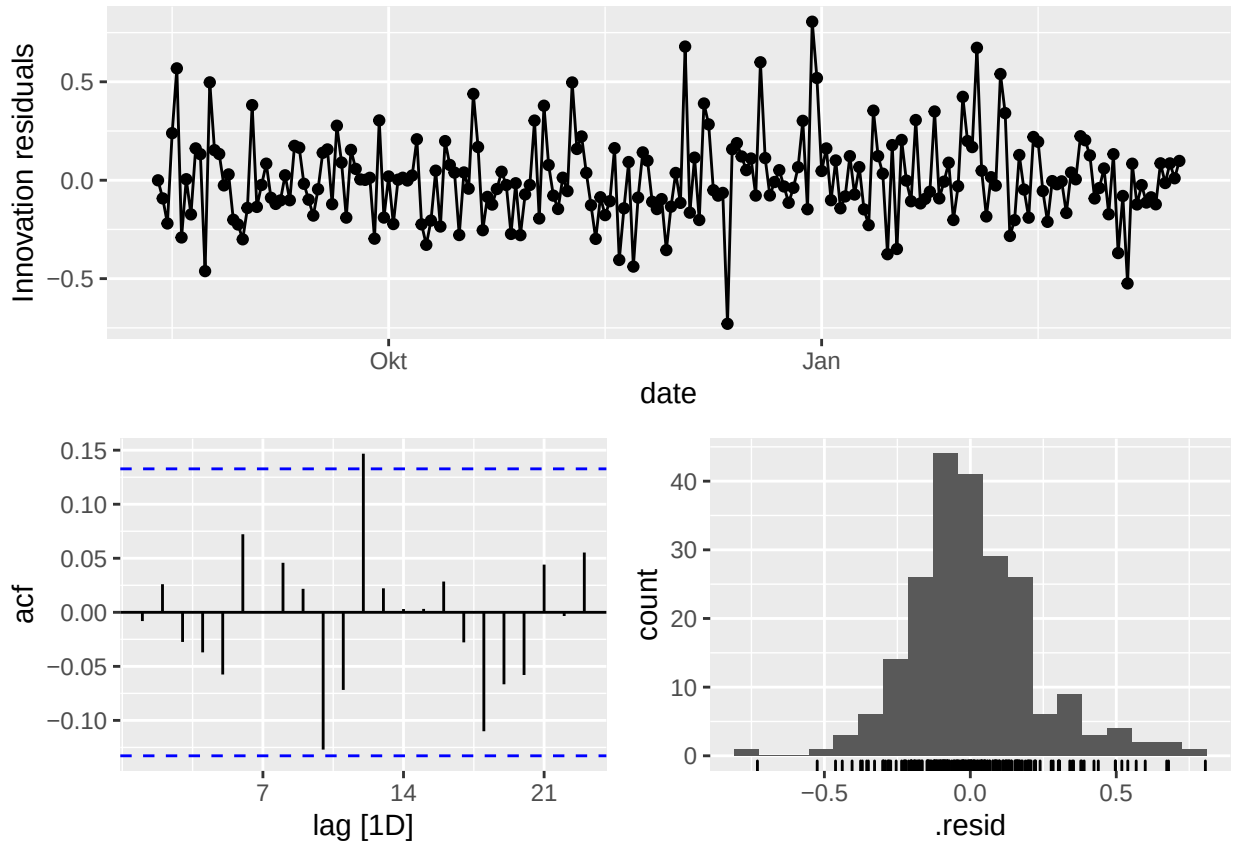
▶ What order is the (S)ARIMA model for the residuals?

The order for the residuals is 1.

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

The residuals over time don't display any obvious seasonality (at least in my opinion). They seem to be normally distributed and they generally have low correlation with lagged versions of itself although they is a significant one at lag 12 which indicates that the model should be improved.

```
gg_tsresiduals(fit2)
```



2.3. Forecasts

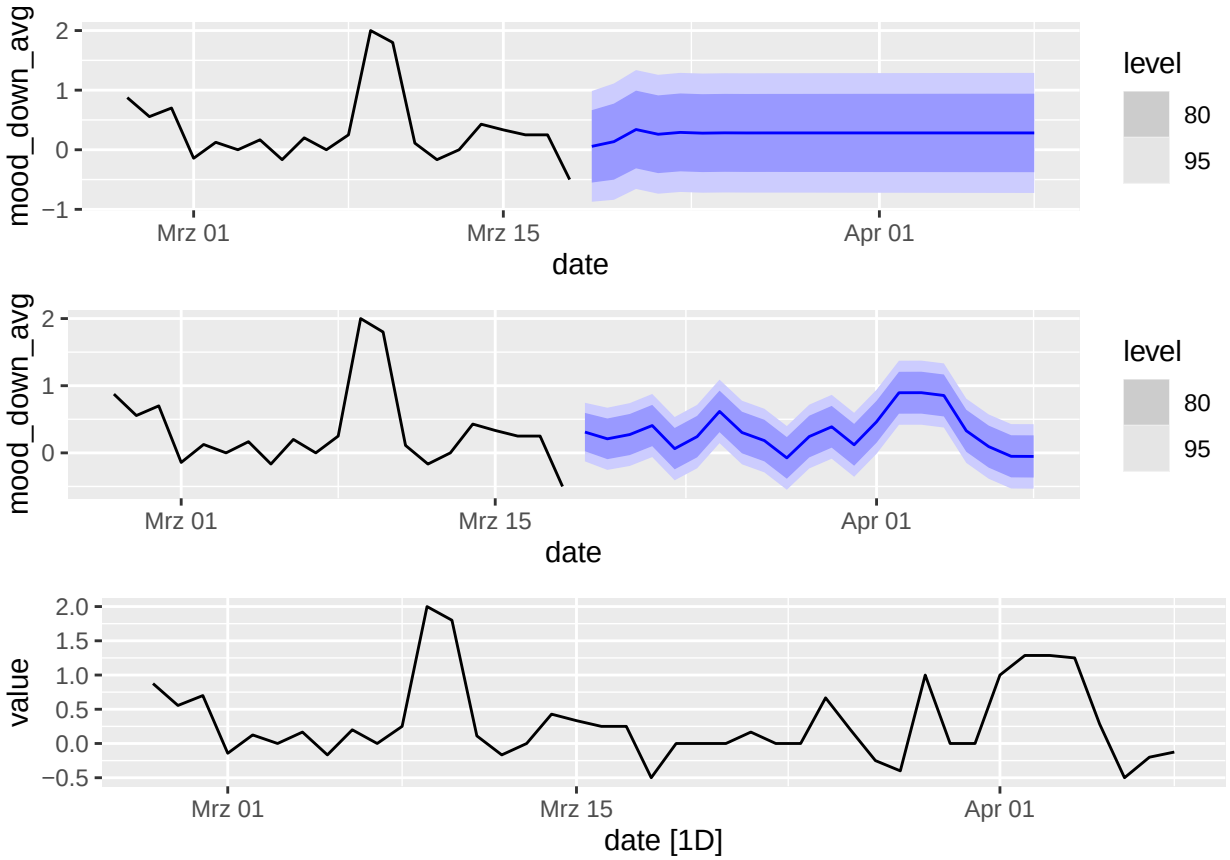
▶ Choose a forecasting horizon, and indicate why this is a reasonable and interesting horizon to consider.

I chose the last 21 values (3 weeks) of the dataset as the forecasting horizon. I did not want to pick to many values as the time series is not that large but I also did not want to pick too few as there should be enough variation in values. That is why I decided that 3 weeks would be a good forecasting horizon.

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

```
# forecasts for outcome variable without predictor, only based on the
# parameters of the model that was fitted
forecast1<- forecast(fit1, new_data=test) %>% autoplot()
forecast1<-forecast1+ autolayer(tail(train,21), vars(mood_down_avg)) # 20 time points f

# forecasts for outcome variable based on future values of the predictors, and on the
# parameters of the model that was fitted
forecast2<- forecast(fit2, new_data=test) %>% autoplot()
forecast2<-forecast2 +autolayer(tail(train,21), vars(mood_down_avg))
```

```
actual_future<-autoplot(tail(df_ts_clean,42), vars(mood_down_avg)) # compare it to actu
grid.arrange( forecast1, forecast2,actual_future, nrow = 3)# plot
```



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

The forecast without predictors is almost a flat line while the forecast with predictors is comparable to the preceding time points from the original dataset and also displays more narrow confidence intervals. Comparing the first two plots with the last one (actual data). We can see that the fit with predictors is quite similar to the actual future and thus the better model for forecasting.

3. Causal Modeling

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

Does physical tiredness lead to feeling “down”?

▶ 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?

Since I am not looking at an intervention and there might be a reciprocal relationship between X and Y i.e. Y might also lead to X. I decided on the Granger Causal approach (although I will only assess Granger causality in one direction).

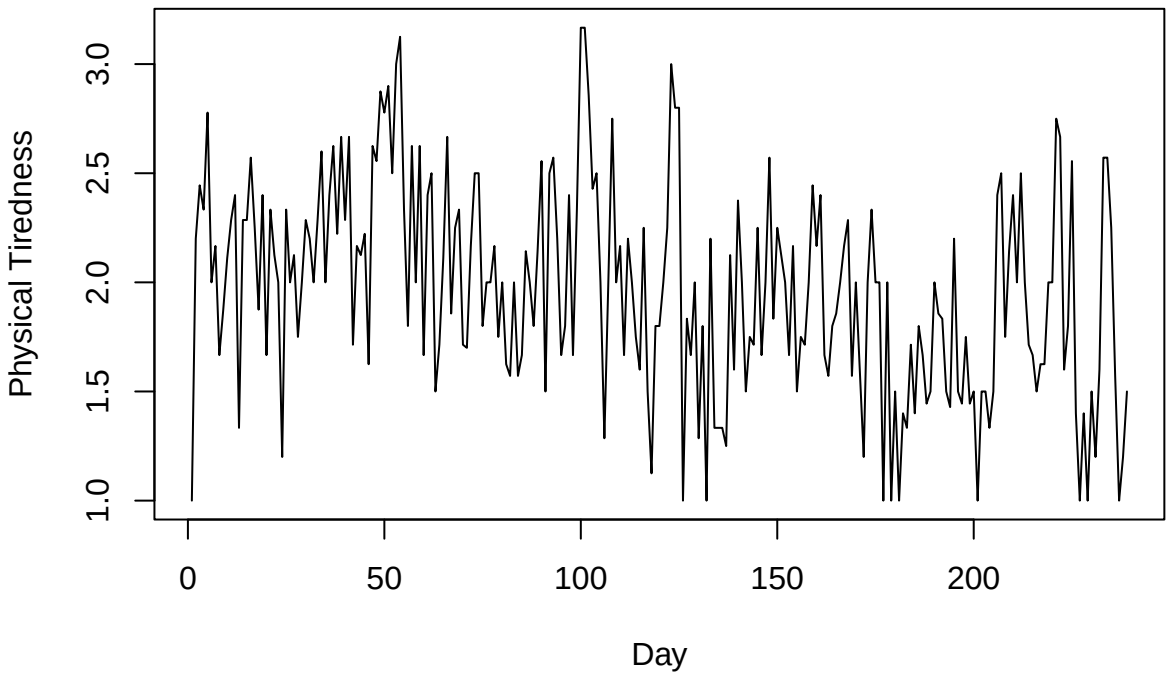
3.2 Analysis

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.

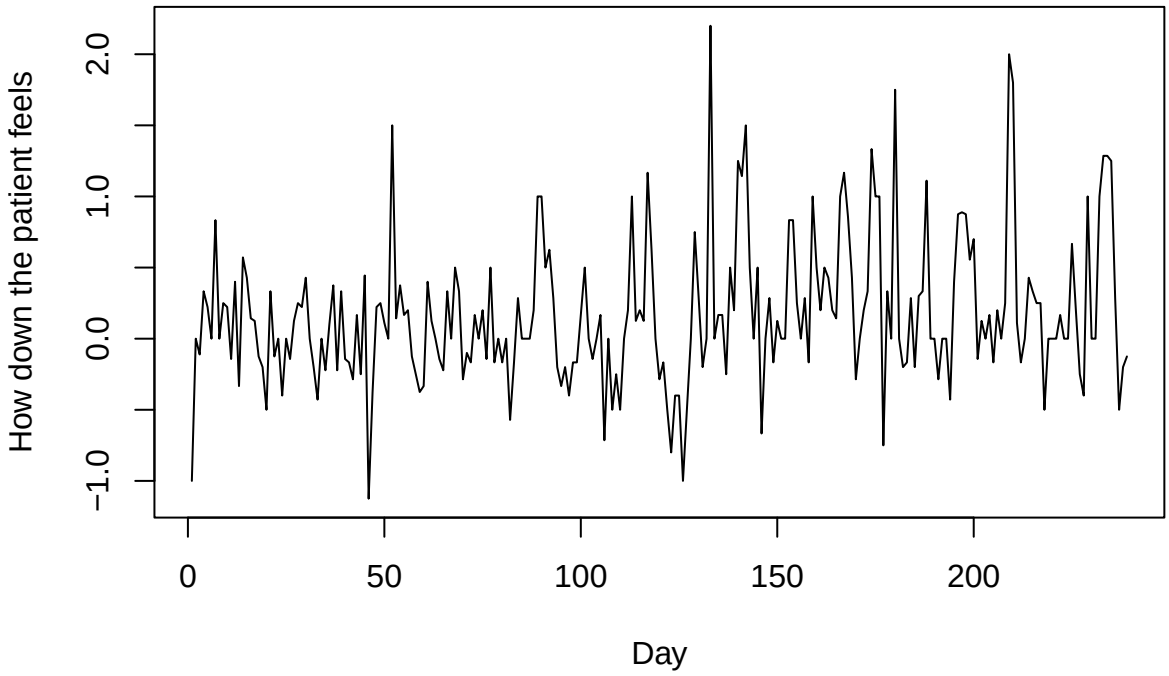
3.2a Granger Causal analysis

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

```
ts.plot(df_ts_clean$phy_tired_avg, ylab="Physical Tiredness", xlab= "Day") # x variable
```



```
ts.plot(df_ts_clean$mood_down_avg, ylab="How down the patient feels", xlab= "Day") # y v
```



▶ 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.

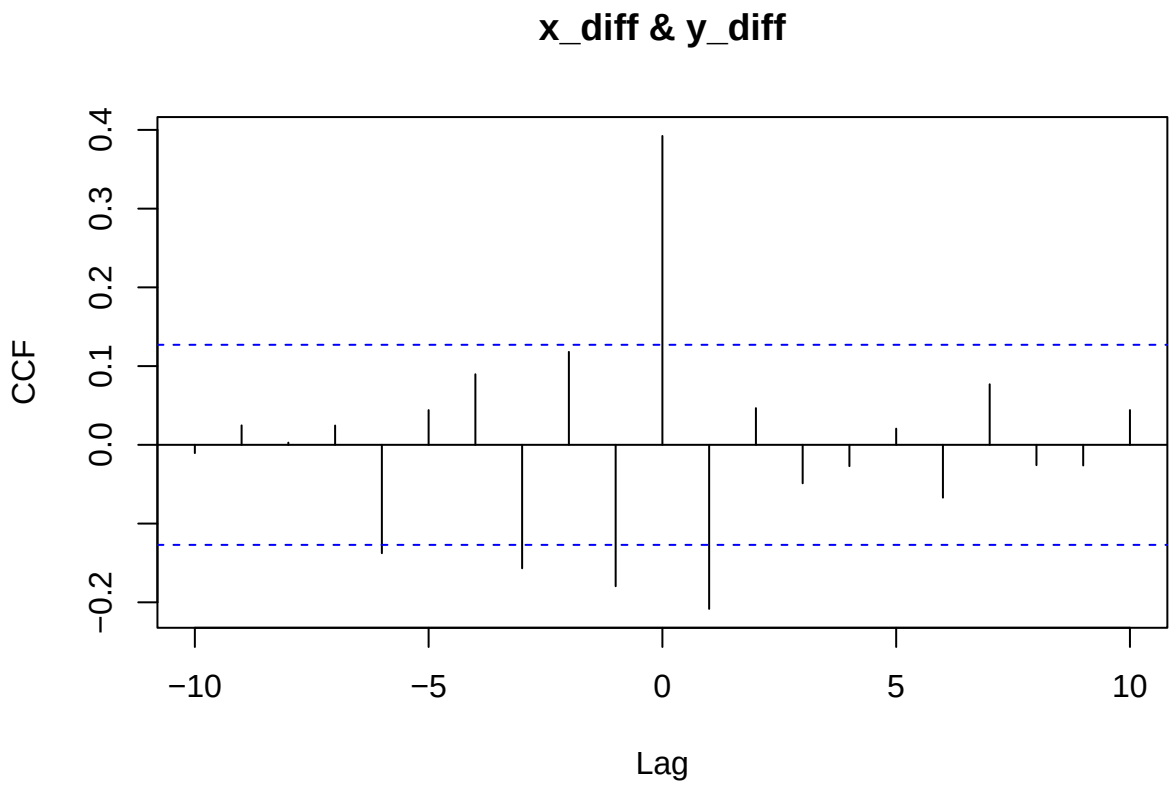
I am using the fit without predictors from part 1. The outcome variable is mood_down and after fitting the selected model was a ARIMA (1,1,3) model. The reason for not using the model with additional covariates is the fact that we were adviced to leave them out for this part of the assignment to keep it simple.

▶ 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.

Looking at the CCF we can see that there is significant correlation from lag -6 until 1, namely at lag 0, -1, -3, and -6. This indicates that the X values might be predictive of future Y values. Moreover we also see a significant correlation at lag = 1 which indicates that past values of Y might also predict X. Considering these findings I decided to use a minimum lag of 1 and a maximum lag of 6.

```
x<-df_ts_clean$phy_tired_avg
y<- df_ts_clean$mood_down_avg
```

```
y_diff<-diff(y)
x_diff<-diff(x)
ccf(x_diff,y_diff,10, ylab="CCF")
```



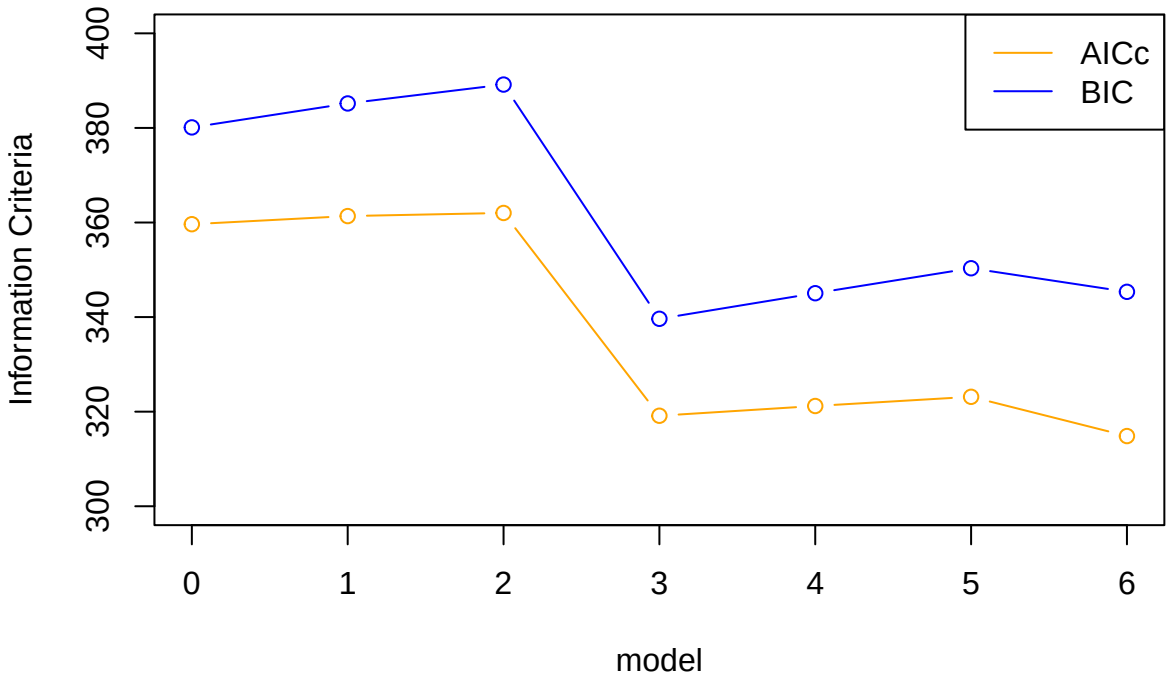
▶ Investigate whether adding your lagged "cause'' variables (X) improve the prediction of your effect variable(s) Y. Use model selection based on information criteria. Describe your final chosen model

We see that the models which include X as a lagged predictor outperform the 'independence' model (that is, the ARIMA model which ignores past values of X) according to both information criteria. This indicates that information about the past of X aids our prediction of future values of Y. We can see that the minimum AICc is reached at lag -6 but lag -3 is only minimally larger. Moreover, the BIC is clearly the lowest at lag -3. That's why I then compared the two models based on the parameter estimates. We can see that the models are actually very similar: In both cases we select an AR(1) model with lagged predictors. Moreover, the ar-1 parameter and standard errors for Y, as well as the lag-1 and lag-3 effect of X are very similar in both cases. In the lag-6 model, the lag-6 is larger compared to the other lag effect which might indicate that the lag-6 model is the better one. On the other hand, the differences between the AICc are minimal (smaller in the lag-6 model) and the BIC is even higher in the lag-6 model. The residuals, distribution and ACF do not differ a lot and are comparable. Based on that and if lower complexity is preferred one might pivot towards the simpler lag-3 model.

```
 fit_causal <- df_ts_clean %>%
  # Restrict data so models use same fitting period
  mutate(y = c(NA,NA,NA,NA,NA,NA, y[7:239])) %>%
  # Estimate models
  model(
    indep = ARIMA(y),
    lag1 = ARIMA(y~ lag(x)),
    lag2 = ARIMA(y~ lag(x) + lag(x,2)),
    lag3 = ARIMA(y~ lag(x) + lag(x,2) + lag(x,3)),
    lag4 = ARIMA(y~ lag(x) + lag(x,2) + lag(x,3) + lag(x,4)),
    lag5 = ARIMA(y~ lag(x) + lag(x,2) + lag(x,3) + lag(x,4) + lag(x,5)),
    lag6 = ARIMA(y~ lag(x) + lag(x,2) + lag(x,3) + lag(x,4) + lag(x,5)+ lag(x,6))
  )

glance(fit_causal)
```

```
## # A tibble: 7 x 8
##   .model sigma2 log_lik   AIC  AICc   BIC ar_roots  ma_roots
##   <chr>   <dbl>   <dbl> <dbl> <dbl> <dbl> <list>    <list>
## 1 indep   0.260   -174.  359.  360.  380. <cpl [5]> <cpl [0]>
## 2 lag1    0.261   -173.  361.  361.  385. <cpl [5]> <cpl [0]>
## 3 lag2    0.260   -173.  361.  362.  389. <cpl [5]> <cpl [0]>
## 4 lag3    0.217   -153.  319.  319.  340. <cpl [1]> <cpl [0]>
## 5 lag4    0.218   -153.  321.  321.  345. <cpl [1]> <cpl [0]>
## 6 lag5    0.219   -153.  323.  323.  350. <cpl [1]> <cpl [0]>
## 7 lag6    0.210   -148.  314.  315.  345. <cpl [1]> <cpl [0]>
```

```
plot(seq(0,6),glance(fit_causal)$AICc,
     col = "orange", type = "b",
     ylab = "Information Criteria", xlab = "model",
     ylim = c(300,400))
lines(seq(0,6),glance(fit_causal)$BIC, col = "blue", type = "b")
legend("topright", c("AICc","BIC"), col = c("orange","blue"), lty = 1)
```




```
fit_best_aic <- df_ts_clean %>% model(ARIMA(y~ lag(x) + lag(x,2) + lag(x,3)))
report(fit_best_aic)
```

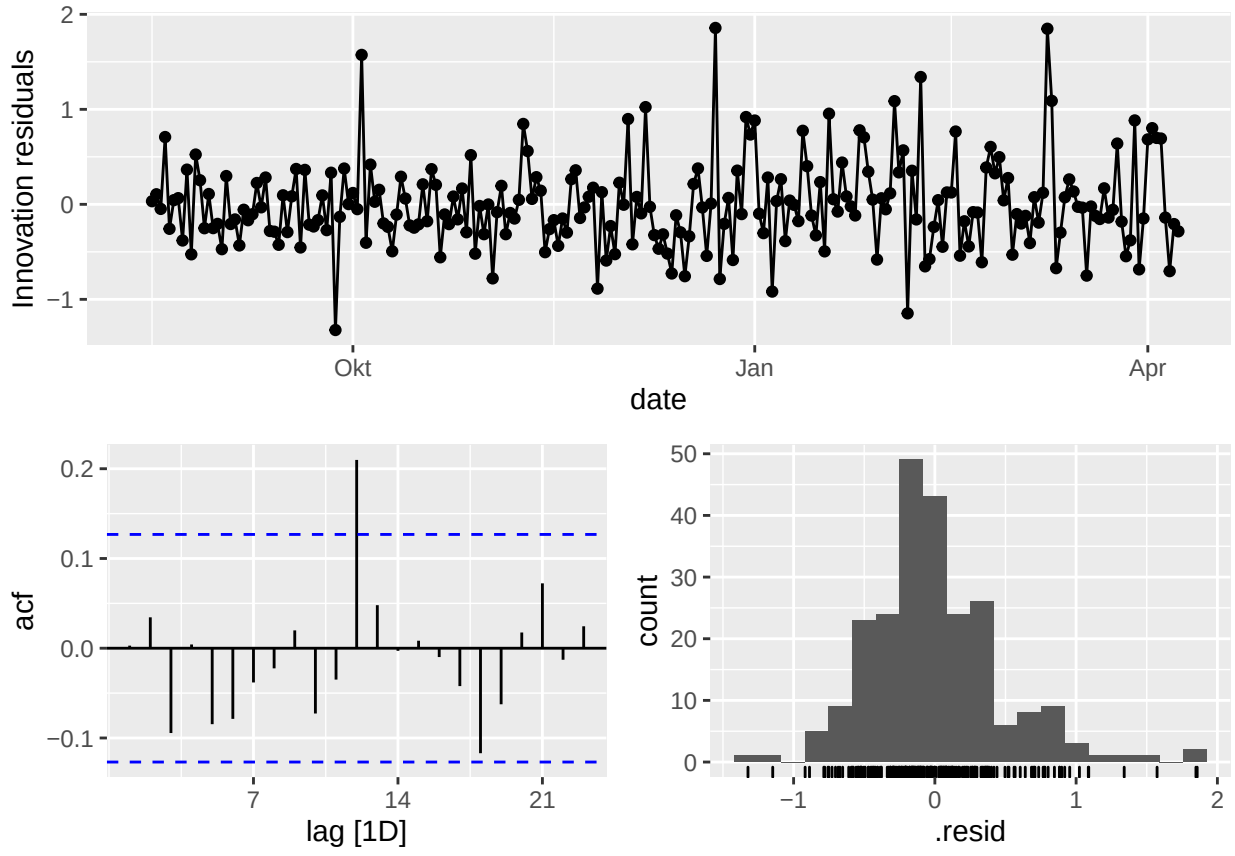
```
## Series: y
## Model: LM w/ ARIMA(1,0,0) errors
##
## Coefficients:
##          ar1   lag(x)  lag(x, 2)  lag(x, 3)  intercept
##       0.3521  -0.1151     0.0256    -0.1778     0.7011
## s.e.  0.0622   0.0724     0.0702     0.0710     0.2342
##
## sigma^2 estimated as 0.2177:  log likelihood=-154.04
## AIC=320.08   AICc=320.44   BIC=340.94
```

```
fit_best_aic2 <- df_ts_clean %>% model(ARIMA(y~ lag(x) + lag(x,2) + lag(x,3)+ lag(x,4) +
report(fit_best_aic2)
```

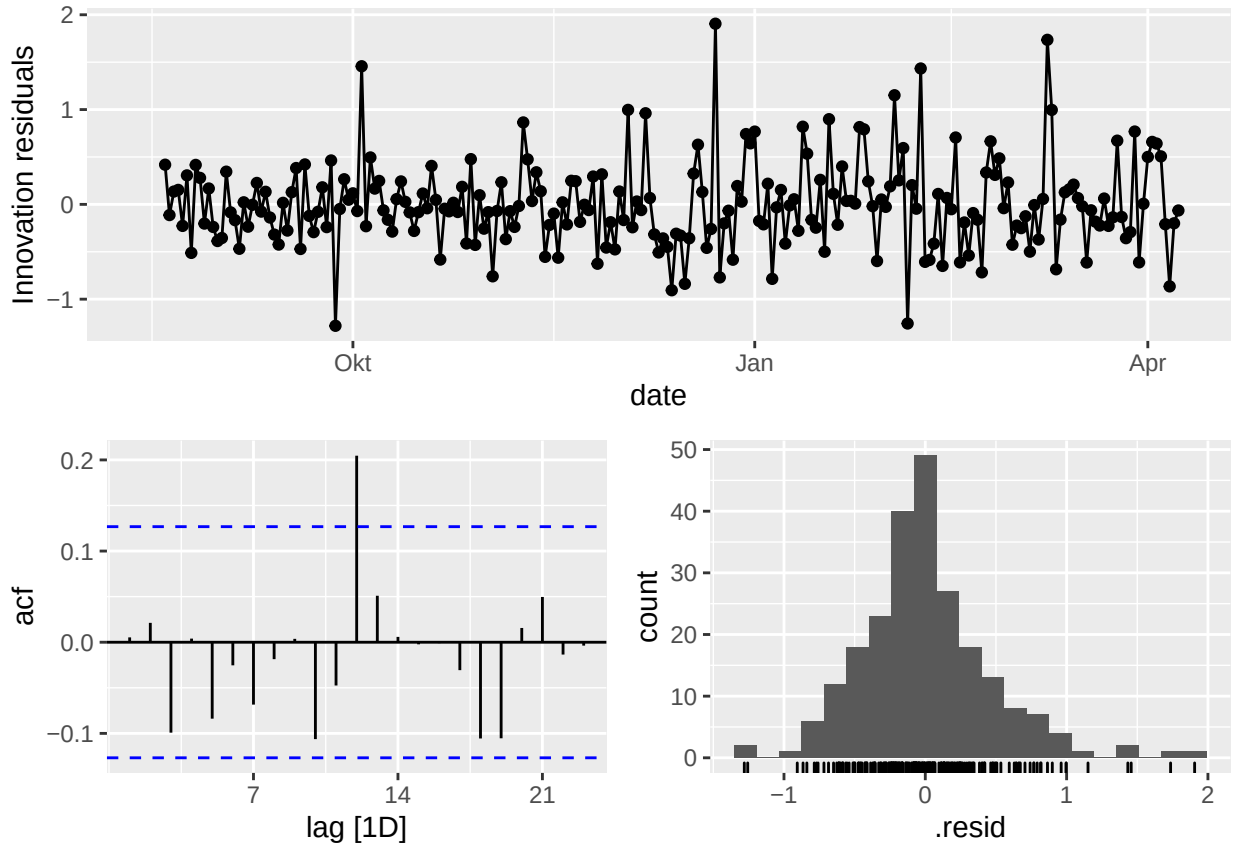
```
## Series: y
## Model: LM w/ ARIMA(1,0,0) errors
##
```

```
## Coefficients:
##          ar1   lag(x)  lag(x, 2)  lag(x, 3)  lag(x, 4)  lag(x, 5)  lag(x, 6)
##       0.3329  -0.1054      0.062    -0.1548     0.0147    -0.0406    -0.2319
## s.e.  0.0632   0.0728      0.071     0.0718     0.0718     0.0718     0.0709
##       intercept
##          1.0747
## s.e.     0.2838
##
## sigma^2 estimated as 0.2103:  log likelihood=-148.03
## AIC=314.05   AICc=314.84   BIC=345.34
```

```
gg_tsresiduals(fit_best_aic)
```




```
gg_tsresiduals(fit_best_aic2)
```



3.3 Conclusion and critical reflection

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

We could say that physical tiredness is a 'prima facie' cause of feeling down as a model with lags was chosen. This means that we fail to find evidence that physical tiredness is not a cause of Y. (Although the model should be improved as there is still significant autocorrelation for the residuals.)

▶ 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 plausability and possible threats to the validity of this assumption in your specific setting (< 75 words)

I assumed no unobserved confounding (sufficiency). This means that the results I got from using the Granger causality method might not be that meaningful as I didn't control for confounding although the other variables showed to have effects on the outcome variable (part 1). This is especially problematic when using Granger causality as the goal is to condition on all other variables at time t that could possibly have an influence on Yt+.