In Turkey, some parts of society always compare Turkey to Germany and think that we are better than Germany for a lot of issues. The same applies to COVID-19 crisis management; is that reflects to true?

We will use two variables for compared parameters; the number of daily new cases and daily new deaths.First, we will compare the mean of new cases of the two countries. The dataset is attached in the repository.

#load and tidying the dataset

library(readxl)

deu <- read\_excel("covid-data.xlsx",sheet = "deu")

deu$date <- as.Date(deu$date)

tur <- read\_excel("covid-data.xlsx",sheet = "tur")

tur$date <- as.Date(tur$date)

#building the function comparing means on grid table

grid\_comparing <- function(column="new\_cases"){

table <-data.frame(

deu=c(mean=mean(deu[[column]]),sd=sd(deu[[column]]),n=nrow(deu)),

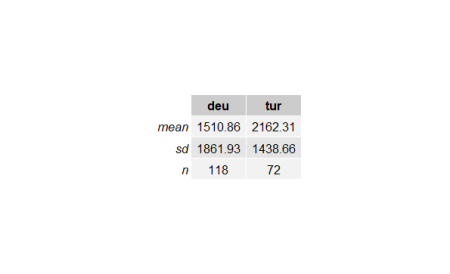
tur=c(mean=mean(tur[[column]]),sd=sd(tur[[column]]),n=nrow(tur))

) %>% round(2)

grid.table(table)

}

grid\_comparing()



Above table shows that the mean of new cases in Turkey is greater than Germany. To check it, we will inference concerning the difference between two means.

In order to make statistical inference for the , the sample distribution must be approximately normal distribution. If it is assumed that the related populations will not be normal, sample distribution is approximately normal only in the volume of relevant samples greater than 30 separately according to the **central limit theorem**. In this case, the distribution is assumed approximately normal.

If the variances of two populations and are known, **z-distribution** would be used for statistical inference. A more common situation, if the variances of population are unknown, we will instead use samples variances , and **distribution**.

When and are unknown, two situation are examined.

* : the assumption they are equal.
* : the assumption they are not equal.

There is a formal test to check whether population variances are equal or not which is a **hypothesis test for the ratio of two population variances**. A two-tailed hypothesis test is used for this as shown below.

The test statistic for :

The sample volumes n_1and n_2, **degrees of freedom of the samples** and . **F-distribution** is used to describe the sample distribution of

var.test(deu$new\_cases,tur$new\_cases)

# F test to compare two variances

#data: deu$new\_cases and tur$new\_cases

#F = 1.675, num df = 117, denom df = 71, p-value = 0.01933

#alternative hypothesis: true ratio of variances is not equal to 1

#95 percent confidence interval:

# 1.088810 2.521096

#sample estimates:

#ratio of variances

# 1.674964

At the %5 significance level, because **p-value(0.01933)** is less than 0.05, the null hypothesis(H_0) is rejected and we assume that variances of the populations are not equal.

Because the variances are not equal we use **Welch’s t-test** to calculate test statistic:

The degree of freedom:

Let’s see whether the mean of new cases per day of Turkey() greater than Germany(); to do that we will build the hypothesis test as shown below:

#default var.equal value is set to FALSE that indicates that the test is Welch's t-test

t.test(tur$new\_cases,deu$new\_cases,alternative = "g")

# Welch Two Sample t-test

#data: tur$new\_cases and deu$new\_cases

#t = 2.7021, df = 177.67, p-value = 0.00378

#alternative hypothesis: true difference in means is greater than 0

#95 percent confidence interval:

# 252.8078 Inf

#sample estimates:

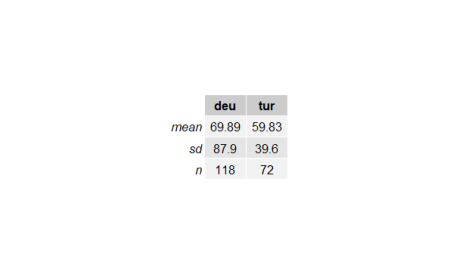
#mean of x mean of y

# 2162.306 1510.856

As shown above , at the %5 significance because the p-value(0.00378) is les than 0.05 the alternative hypothesis is accepted, which means in terms of controlling the spread of the disease, Turkey seems to be less successful than in Germany.

Another common thought in Turkish people that the health system in the country is much better than many European countries including Germany; let’s check that with daily death toll variable (new\_deaths).

grid\_comparing("new\_deaths")



It seems Turkey has much less mean of daily deaths than Germany. Let’s check it.

var.test(deu$new\_deaths,tur$new\_deaths)

# F test to compare two variances

#data: deu$new\_deaths and tur$new\_deaths

#F = 4.9262, num df = 117, denom df = 71, p-value = 1.586e-11

#alternative hypothesis: true ratio of variances is not equal to 1

#95 percent confidence interval:

# 3.202277 7.414748

#sample estimates:

#ratio of variances

# 4.926203

As described before, we will use Welch’s t-test because the variances are not equal as shown above(**p-value = 1.586e-11 < 0.05**).

t.test(deu$new\_deaths,tur$new\_deaths,alternative = "g")

# Welch Two Sample t-test

#data: deu$new\_deaths and tur$new\_deaths

#t = 1.0765, df = 175.74, p-value = 0.1416

#alternative hypothesis: true difference in means is greater than 0

#95 percent confidence interval:

# -5.390404 Inf

#sample estimates:

#mean of x mean of y

# 69.88983 59.83333

At %5 significance level, alternative hypothesis is rejected(**p-value = 0.1416 >0.05**). This indicates that the mean of daily deaths of Germany is not worst than Turkey’s.

**June 1** is set as the day of normalization by the Turkish government therefore many restrictions will be removed after that day. In order to check the decision, first, we will determine fit models for forecasting. To find the fit model we will build a function that compares trend regression models in a plot.

Trend Regression Model

Gasoline prices always is an issue in Turkey; because Turkish people love to drive where they would go but they complain about the prices anyway. I wanted to start digging for the last seven years’ prices and how they went. I have used unleaded gasoline 95 octane prices from Petrol Ofisi which is a fuel distribution company in Turkey. I arranged the prices monthly between 2013 and 2020 as the Turkish currency, Lira (TL). The dataset is [here](https://github.com/mesdi/trend_seasonality/blob/master/gasoline_df.xlsx?raw=true)

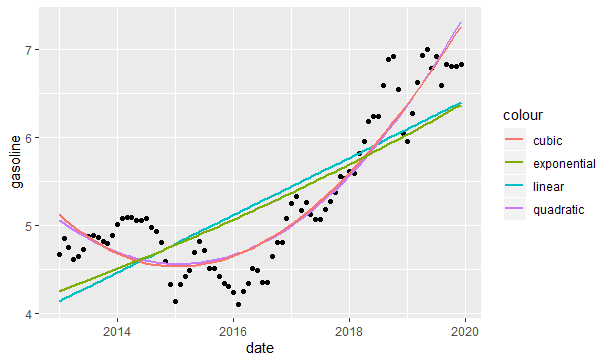
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | head(gasoline\_df)  # date gasoline  #1 2013-01-01 4.67  #2 2013-02-01 4.85  #3 2013-03-01 4.75  #4 2013-04-01 4.61  #5 2013-05-01 4.64  #6 2013-06-01 4.72 |

First of all, I wanted to see how gasoline prices went during the period and analyze the fitted trend lines.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | #Trend lines  library(tidyverse)    ggplot(gasoline\_df, aes(x = date, y = gasoline)) + geom\_point() +    stat\_smooth(method = 'lm', aes(colour = 'linear'), se = **FALSE**) +    stat\_smooth(method = 'lm', formula = y ~ poly(x,2), aes(colour = 'quadratic'), se= **FALSE**) +    stat\_smooth(method = 'lm', formula = y ~ poly(x,3), aes(colour = 'cubic'), se = **FALSE**)+    stat\_smooth(data=exp.model.df, method = 'loess',aes(x,y,colour = 'exponential'), se = **FALSE**) |

Because the response variable is measured by logarithmic in the exponential model, we have to get the exponent of it and create a different data frame for the exponential trend line.

|  |  |
| --- | --- |
| 1  2  3  4 | #exponential trend model  exp.model <- lm(log(gasoline)~date,data = gasoline\_df)  exp.model.df <- data.frame(x=gasoline\_df$date,                             y=exp(fitted(exp.model))) |



As we see above, the cubic and quadratic models almost overlap each other and they seem to be fit better to the data. However, we will analyze each model in detail.  
  
**The Forecasting Trend Models**  
**The linear trend**; y_{t}, the value of the series at given time, t, is described as:  
  
y_{t}=\beta_{0}+\beta_{1}t+\epsilon_{t}  
  
\beta_{0} and \beta_{1} are the coefficients.

|  |  |
| --- | --- |
| 1 | model\_linear <- lm(data = gasoline\_df,gasoline~date) |

Above, we created a model variable for the linear trend model. In order to compare the models, we have to extract the adjusted coefficients of determination, that is used to compare regression models with a different number of explanatory variables, from each trend models.

**Adjusted** R^2=1-(1-R^2)(\frac {n-1} {n-k-1})  
n: sample size  
k: number of variables

R^2: coefficient of determination which is the square of correlation of observation values with predicted values: r^2_{y \hat {y}}

|  |  |
| --- | --- |
| 1  2  3 | adj\_r\_squared\_linear <- summary(model\_linear) %>%    .$adj.r.squared %>%    round(4) |

**The exponential trend**; unlike the linear trend, allows the series to increase at an increasing rate in each period, is described as:  
  
\ln(y_{t})=\beta_{0}+\beta_{1}t+\epsilon_{t}. \ln(y_{t}) is a natural logarithm of the response variable.  
  
To make predictions on the fitted model, we use exponential function as \hat{y_{t}}=exp(b_{0}+b_{1}t+s_{e}^2/2) because the dependent variable was transformed by a natural logarithmic function. In order for \hat{y_{t}} not to be under the expected value of y{t}, we add half of the residual standard error’s square. In order to find s_{e}, we execute the summary function as below. In addition, we can see the level of significance of the coefficients and the model. As we see below, because the p-value of the coefficients and the model are less than 0.05, they are significant at the %5 level of significance.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | summary(model\_exponential)    #Call:  #lm(formula = log(gasoline) ~ date, data = gasoline\_df)    #Residuals:  #      Min        1Q    Median        3Q       Max  #-0.216180 -0.083077  0.009544  0.087953  0.151469    #Coefficients:  #              Estimate Std. Error t value Pr(<|t|)  #(Intercept) -1.0758581  0.2495848  -4.311  4.5e-05 \*\*\*  #date         0.0001606  0.0000147  10.928  < 2e-16 \*\*\*  #---  #Signif. codes:  0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1    #Residual standard error: 0.09939 on 82 degrees of freedom  #Multiple R-squared:  0.5929,    Adjusted R-squared:  0.5879  #F-statistic: 119.4 on 1 and 82 DF,  p-value: < 2.2e-16 |

We create an exponential model variable.

|  |  |
| --- | --- |
| 1 | model\_exponential <- lm(data = gasoline\_df,log(gasoline)~date) |

To get adjusted R^2, we first transform the predicted variable.

|  |  |
| --- | --- |
| 1  2  3  4 | library(purrr)    y\_predicted <- model\_exponential$fitted.values %>%    map\_dbl(~exp(.+0.09939^2/2)) |

And then, we execute the formula shown above to get adjusted R^2

|  |  |
| --- | --- |
| 1  2  3  4 | r\_squared\_exponential <- (cor(y\_predicted,gasoline\_df$gasoline))^2    adj\_r\_squared\_exponential <- 1-(1-r\_squared\_exponential)\*    ((nrow(gasoline\_df)-1)/(nrow(gasoline\_df)-1-1)) |

Sometimes a time-series changes the direction with respect to many reasons. This kind of target variable is fitted to **polynomial trend models**. If there is one change of direction we use **the quadratic trend model.**  
  
y_{t}=\beta_{0}+\beta_{1}t+\beta_{2}t^2+\epsilon  
  
If there are two change of direction we use **the cubic trend models:**  
  
y_{t}=\beta_{0}+\beta_{1}t+\beta_{2}t^2+\beta_{3}t^3+\epsilon  
  
Then again, we execute the same process as we did in the linear model.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | #Model variables  model\_quadratic <- lm(data = df,gasoline~poly(date,2))  model\_cubic <- lm(data = df,gasoline~poly(date,3))    #adjusted coefficients of determination  adj\_r\_squared\_quadratic <- summary(model\_quadratic) %>%    .$adj.r.squared    adj\_r\_squared\_cubic <- summary(model\_cubic) %>%    .$adj.r.squared |

In order to compare all models, we create a variable that gathers all adjusted R^2.

|  |  |
| --- | --- |
| 1  2  3  4 | adj\_r\_squared\_all <- c(linear=round(adj\_r\_squred\_linear,4),                     exponential=round(adj\_r\_squared\_exponential,4),                     quadratic=round(adj\_r\_squared\_quadratic,4),                     cubic=round(adj\_r\_squared\_cubic,4)) |

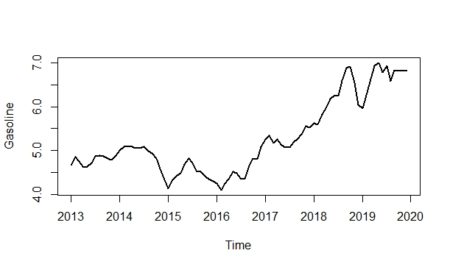
As we can see below, the results justify the graphic we built before; polynomial models are much better than the linear and exponential models. Despite polynomial models look almost the same, the quadratic model is slightly better than the cubic model.

|  |  |
| --- | --- |
| 1  2  3  4 | adj\_r\_squared\_all         #linear exponential   quadratic       cubic       #0.6064      0.6477      0.8660      0.8653 |

**Seasonality**

The seasonality component represents the repeats in a specific period of time. Time series with weekly monthly or quarterly observations tend to show seasonal variations that repeat every year. For example, the sale of retail goods increases every year in the Christmas period or the holiday tours increase in the summer. In order to detect seasonality, we use decomposition analysis.  
  
**Decomposition analysis:** T, S, I, are trend, seasonality and random components of the series respectively. When seasonal variation increases as the time series increase, we’d use the multiplicative model.  
  
y_{t}=T_{t} \times S_{t} \times I_{t}  
  
If the variation looks constant, we should use additive model.  
  
y_{t}=T_{t} + S_{t} + I_{t}  
  
To find which model is fit, we have to look at it on the graph.

|  |  |
| --- | --- |
| 1  2  3  4 | #we create a time series variable for the plot  gasoline\_ts <- ts(gasoline\_df$gasoline,start = c(2013,1),end = c(2019,12),frequency = 12)  #The plot have all the components of the series(T, S, I)  plot(gasoline\_ts,lwd=2,ylab="Gasoline") |

  
  
As we can see from the plot above, it seems that the multiplicative model would be more fit for the data; especially if we look at between 2016-2019. When we analyze the graph, we see some **conjuncture variations** caused by the expansion or contraction of the economy. Unlike seasonality, conjuncture fluctuations can be several months or years in length.  
  
Additionally, the magnitude of conjuncture fluctuation can change in time because the fluctuations differ in magnitude and length so they are hard to reflect with historical data; because of that, we neglected the conjuncture component of the series.  
  
**Extracting the seasonality**Moving averages (**ma**) usually be used to separate the effect of a trend from seasonality.

m\,\,term\,\,moving\,\,averages=\frac {Average\,\,of\,\,the\,\,last\,\,m\,\,observations} {m}  
  
We use the cumulative moving average (**CMA**), which is the average of two consecutive averages, to show the even-order moving average. For example; first two ma terms are \bar y_{6.5} and \bar y_{7.5} but there are no such terms in the original series; therefore we average the two terms to find CMA term matched up in the series:  
  
\bar y_7=\frac {\bar y_{6,5}+\bar y_{7,5}} {2}  
  
The default value of the ‘centre’ argument of the ma function remains TRUE to get CMA value.

|  |  |
| --- | --- |
| 1  2  3 | library(forecast)    gasoline\_trend <- forecast::ma(gasoline\_ts,12) |

If noticed, \bar y_{t} eliminates both seasonal and random variations; hence: \bar y_{t}=T_{t}

Since y_{t}=T_{t} \times S_{t} \times I_{t} and \bar y_{t}=T_{t}, the detrend variable is found by: \frac {\bar y_{t}} {y_{t}}=S_{t} \times I_{t}; it is also called **ratio-to moving averages method**.

|  |  |
| --- | --- |
| 1 | gasoline\_detrend <- gasoline\_ts/gasoline\_trend |

Each month has multiple ratios, where each ratio coincides with a different year; in this sample, each month has seven different ratios. The arithmetic average is used to determine common value for each month; by doing this, we eliminate the random component and subtract the seasonality from the detrend variable. It is called the **unadjusted seasonal index.**

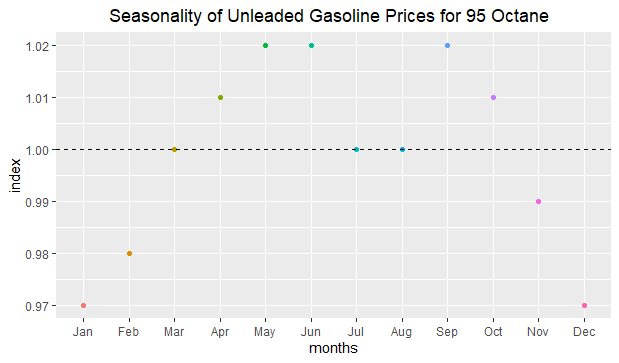
|  |  |
| --- | --- |
| 1  2  3 | unadjusted\_seasonality <- sapply(1:12, **function**(x) mean(window(gasoline\_detrend,  c(2013,x), c(2019,12),deltat=1), na.rm = **TRUE**)) %>%  round(4) |

Seasonal indexes for monthly data should be completed to 12, with an average of 1; in order to do that each unadjusted seasonal index is multiplied by 12 and divided by the sum of 12 unadjusted seasonal indexes.

|  |  |
| --- | --- |
| 1  2 | adjusted\_seasonality <- (unadjusted\_seasonality\*(12/sum(unadjusted\_seasonality))) %>%                          round(4) |

The average of all adjusted seasonal indices is 1; if an index is equal to 1, that means there would be no seasonality. In order to plot seasonality:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | #building a data frame to plot in ggplot  adjusted\_seasonality\_df <- data\_frame(    months=month.abb,    index=adjusted\_seasonality)    #converting char month names to factor sequentially  adjusted\_seasonality\_df$months <- factor(adjusted\_seasonality\_df$months,levels = month.abb)    ggplot(data = adjusted\_seasonality\_df,mapping = aes(x=months,y=index))+   geom\_point(aes(colour=months))+   geom\_hline(yintercept=1,linetype="dashed")+   theme(legend.position = "none")+   ggtitle("Seasonality of Unleaded Gasoline Prices for 95 Octane")+   theme(plot.title = element\_text(h=0.5)) |

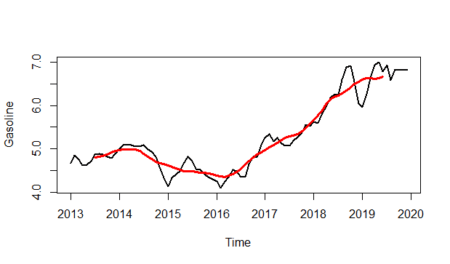


As we can see above, there is approximately a three-percent seasonal decrease in January and December; on the other hand, there is a two-percent seasonal increase in May and June. Seasonality is not seen in March, July, and August; because their index values are approximately equal to 1.

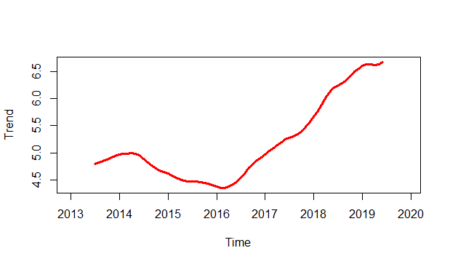
**Decomposing the time series graphically**

We will first show the trend line on the time series.

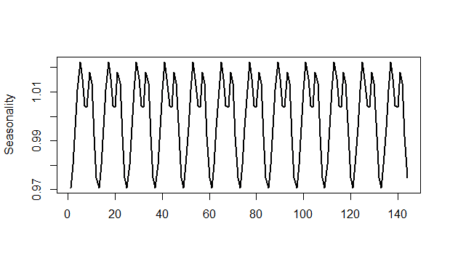
|  |  |
| --- | --- |
| 1  2  3 | #Trend is shown by red line  plot(gasoline\_ts,lwd=2,ylab="Gasoline")+  lines(gasoline\_trend,col="red",lwd=3) |

  
  
And will isolate the trend line from the plot:

|  |  |
| --- | --- |
| 1 | plot(gasoline\_trend,lwd=3,col="red",ylab="Trend") |

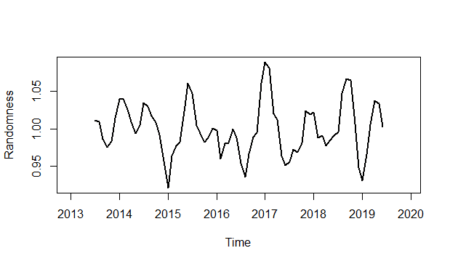
  
  
Let’s see the seasonality line:

|  |  |
| --- | --- |
| 1 | plot(as.ts(rep(unadjusted\_seasonality,12)),lwd=2,ylab="Seasonality",xlab="") |



And finally, we will show the randomness line; in order to do that:  
  
I_{t}=\frac {y_{t}} {S_{t}\times T_{t}}

|  |  |
| --- | --- |
| 1  2 | randomness <- gasoline\_ts/(gasoline\_trend\*unadjusted\_seasonality)  plot(randomness,ylab="Randomness",lwd=2) |



models\_plot <- function(df=tur,column="new\_cases"){

df<- df[!df[[column]]==0,]#remove all 0 rows to calculate the models properly

#exponential trend model data frame

exp\_model <- lm(log(df[[column]])~index,data = df)

exp\_model\_df <- data.frame(index=df$index,column=exp(fitted(exp\_model)))

names(exp\_model\_df)[2] <- column

#comparing the trend plots

ggplot(df,mapping=aes(x=index,y=.data[[column]])) + geom\_point() +

stat\_smooth(method = 'lm', aes(colour = 'linear'), se = FALSE) +

stat\_smooth(method = 'lm', formula = y ~ poly(x,2), aes(colour = 'quadratic'), se= FALSE) +

stat\_smooth(method = 'lm', formula = y ~ poly(x,3), aes(colour = 'cubic'), se = FALSE)+

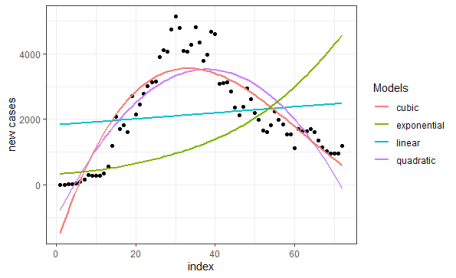
stat\_smooth(data=exp\_model\_df,method = 'loess',mapping=aes(x=index,y=.data[[column]],colour = 'exponential'), se = FALSE)+

labs(color="Models",y=str\_replace(column,"\_"," "))+

theme\_bw()

}

models\_plot()



As we can see from the plot above, the cubic and quadratic regression models seem to fit the data more. To be able to more precise we will create a function that compares **adjusted** .

#comparing model accuracy

trendModels\_accuracy <- function(df=tur,column="new\_cases"){

df<- df[!df[[column]]==0,]#remove all 0 rows to calculate the models properly

model\_quadratic <- lm(data = df,df[[column]]~poly(index,2))

model\_cubic <- lm(data = df,df[[column]]~poly(index,3))

#adjusted coefficients of determination

adj\_r\_squared\_quadratic <- summary(model\_quadratic) %>%

.$adj.r.squared

adj\_r\_squared\_cubic <- summary(model\_cubic) %>%

.$adj.r.squared

c(quadratic=round(adj\_r\_squared\_quadratic,2),cubic=round(adj\_r\_squared\_cubic,2))

}

trendModels\_accuracy()

#quadratic cubic

# 0.73 0.77

**The cubic trend regression model** is much better than the quadratic trend model for Turkeys spread of disease as shown above.

Now, let’s find should the normalization day(June 1) is true. In the following code chunk, we will try some index numbers to find zero new cases.

#forecasting zero point for new cases in Turkey

model\_cubic <- lm(formula = new\_cases ~ poly(index, 3), data = tur)

predict(model\_cubic,newdata=data.frame(index=c(77,78,79,80)))

# 1 2 3 4

#183.92149 111.23894 42.50292 -22.04057

As shown above, index 80 goes to negative, so it can be considered as the day of normalization. If we look at the dataset, we can see that day is June 1. So the government seems to be right about **the normalization calendar**.

You can do the same predictions for Germany using the functions we created before.