Seasonal adjustment of weekly data by discounted least squares in 

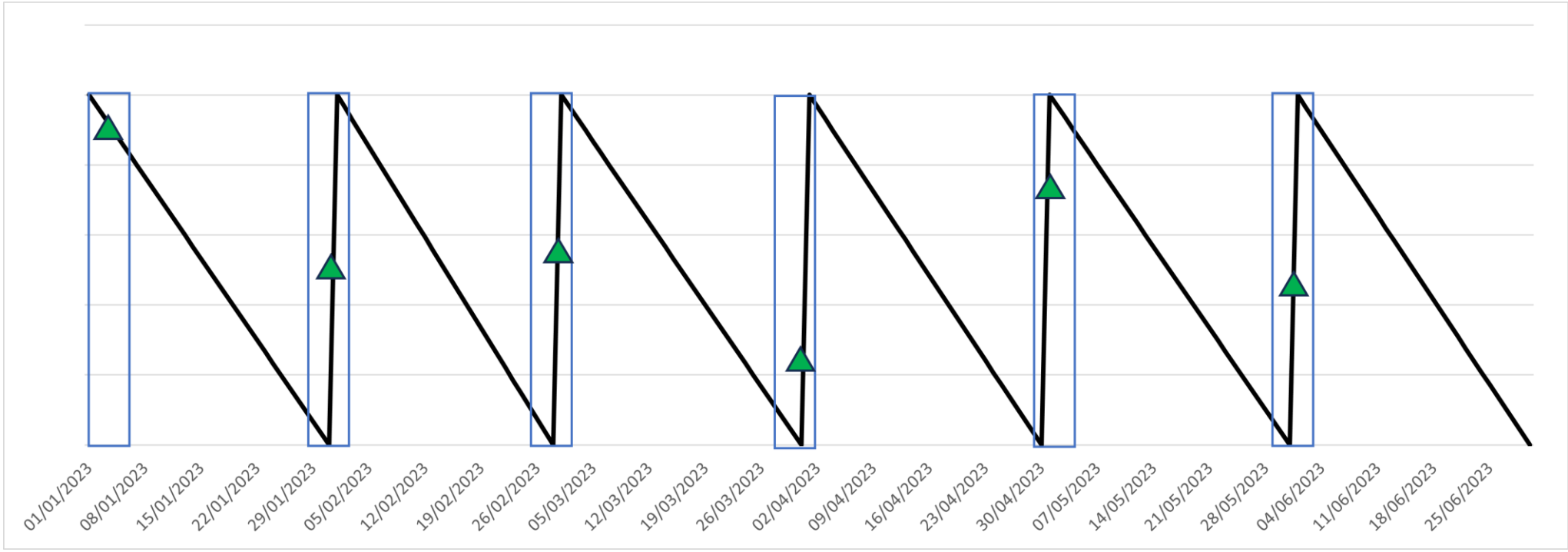
Tim Ginker

Bank of Israel

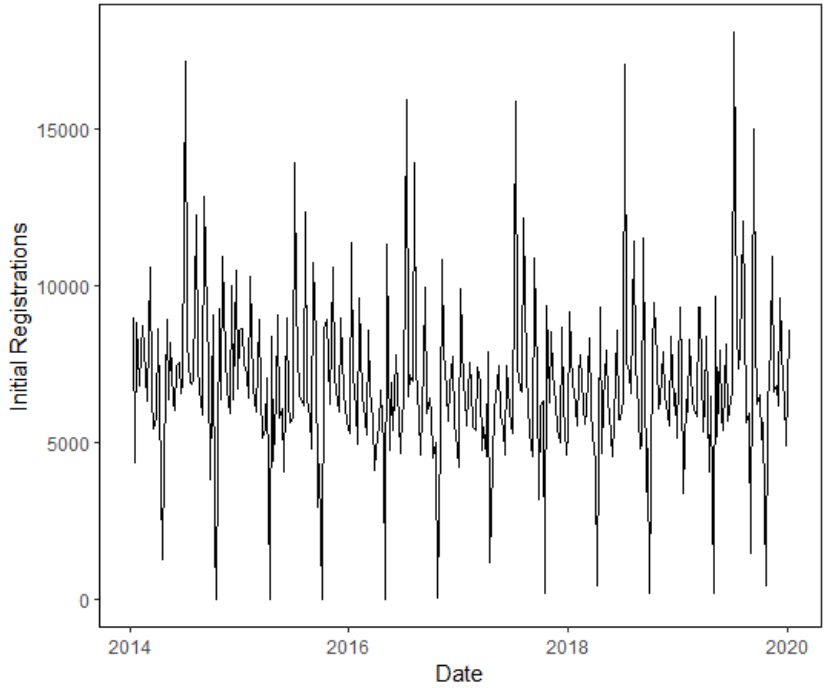
Statistical Methods and Data Science Unit

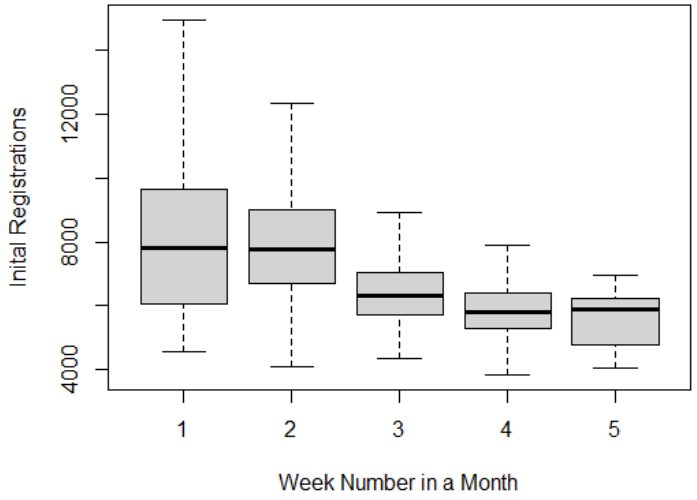
Why can’t we implement conventional statistical methods on weekly data?

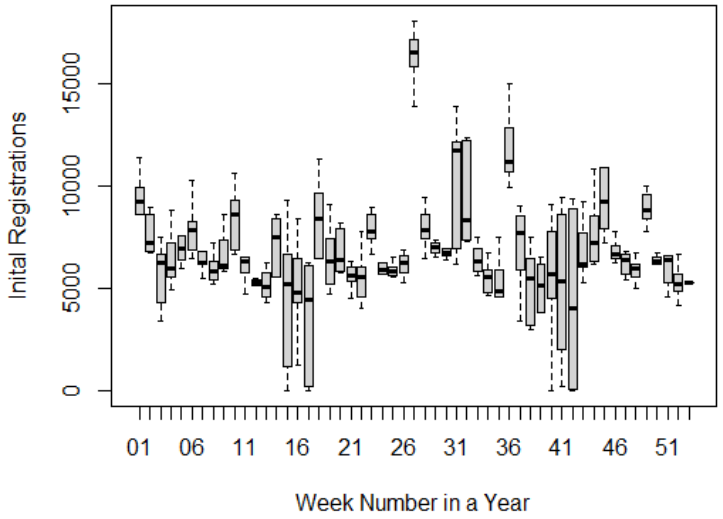
- Varying periodicity (number of weeks in a month/year)
- Multiple seasonal cycles – intra-monthly and intra-yearly cycles.
- Moving window - for weekly data, each week ends at a different position within a month.

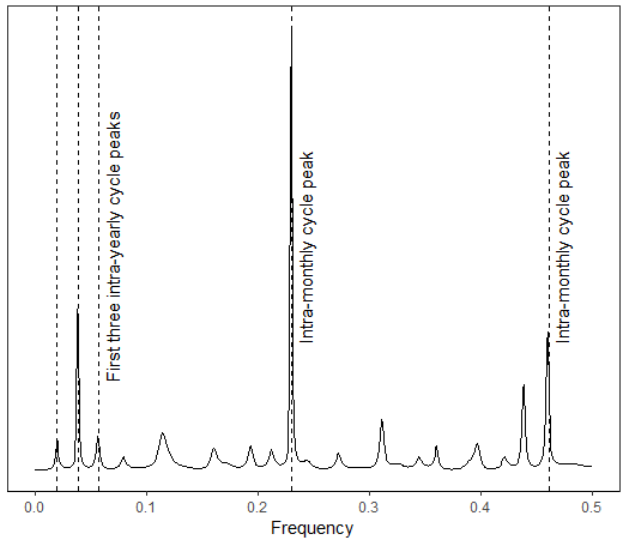


Weekly Number of Initial Registrations in the Israeli Employment Service

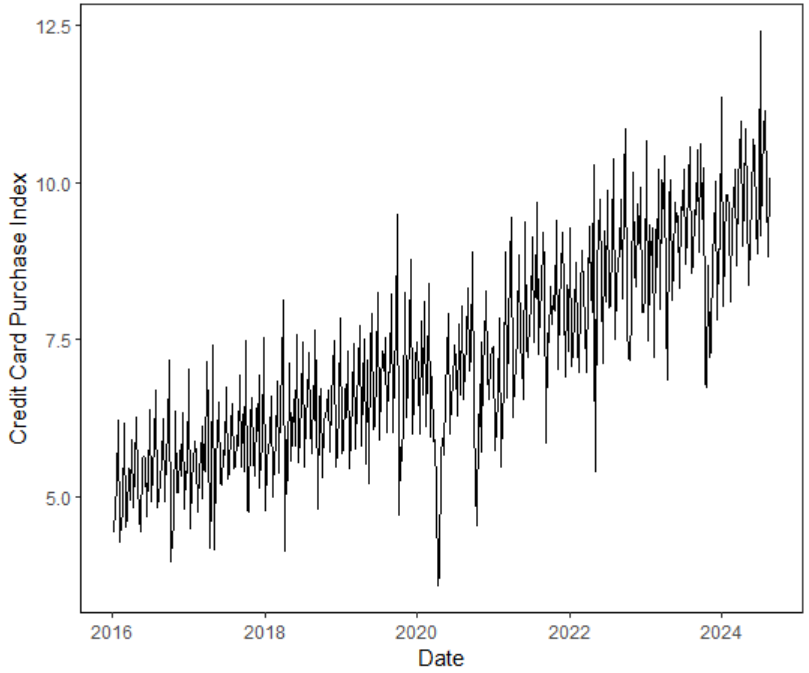


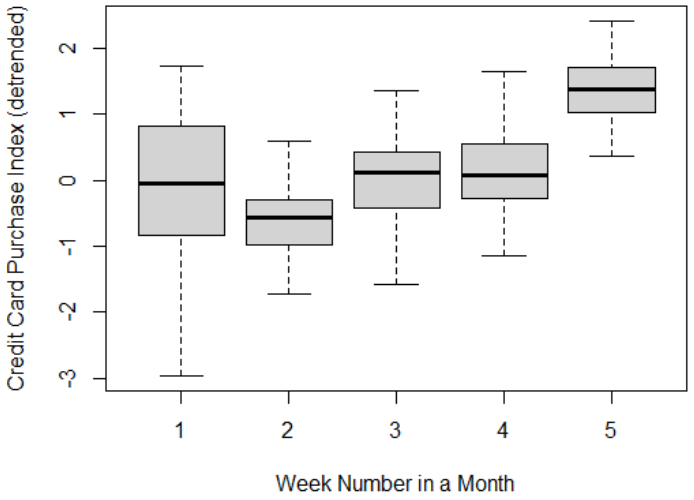


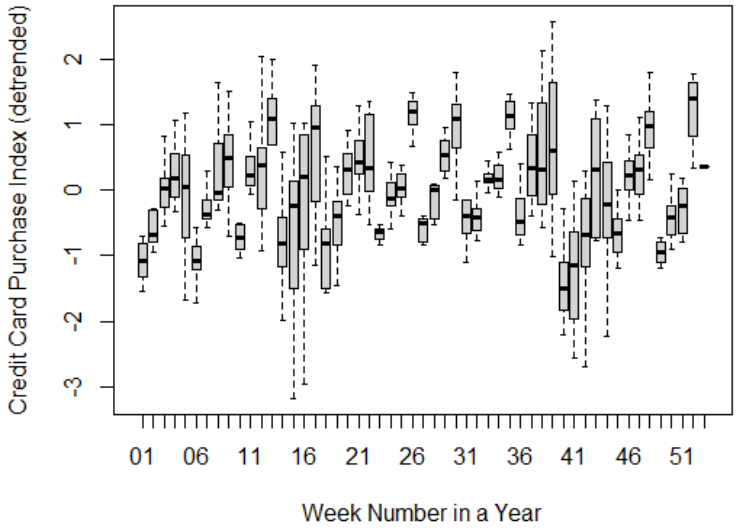


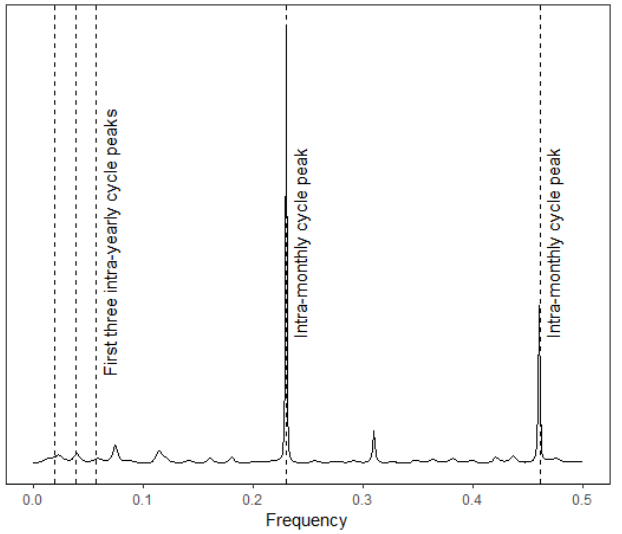


Weekly Credit Card Purchases in Israel

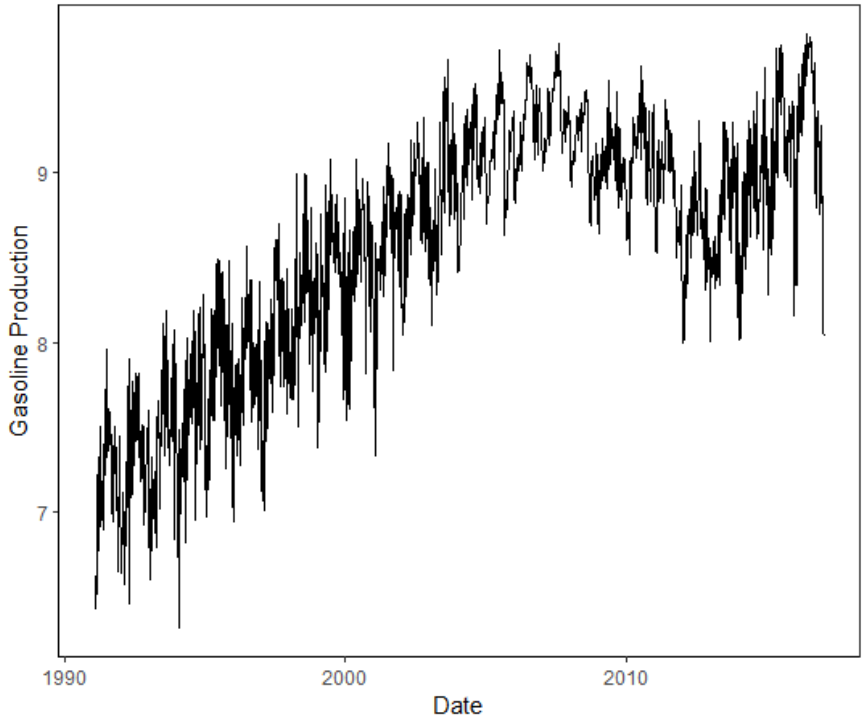


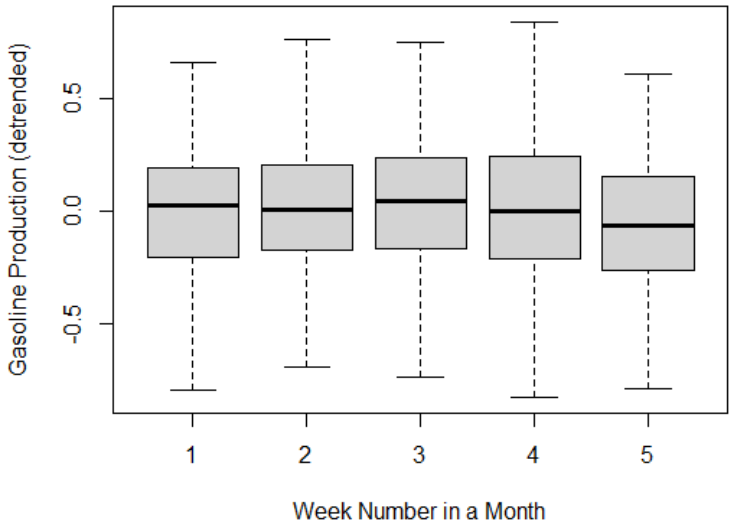


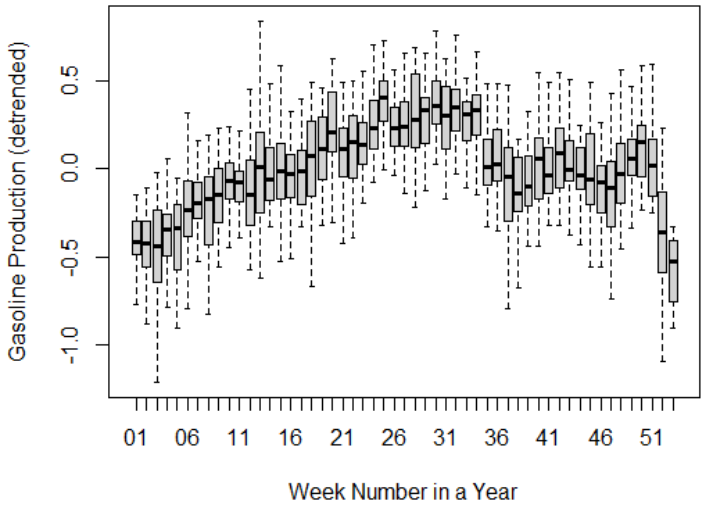


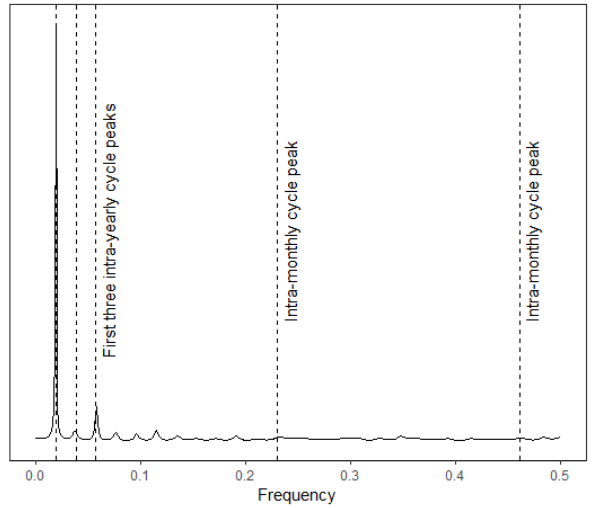


Gasoline Production in the US









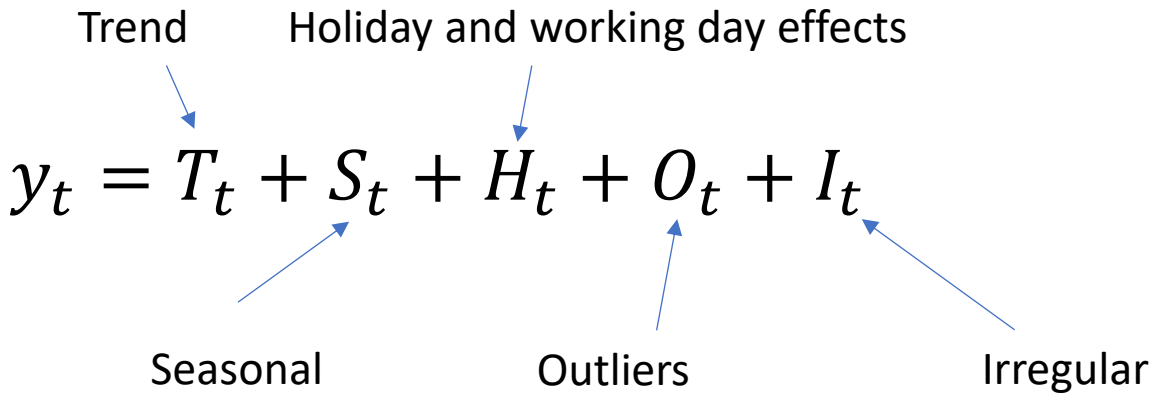
R programs, occasionally employed for seasonal adjustment of weekly data

- **M**ultiple **S**easonal-**T**rend decomposition using **L**oess (MSTL) (Bandara et al., 2021)
- Prophet (Taylor and Letham, 2018)
- TBATS (De Livera et al., 2011)
- stR (Dokumentov and Hyndman, 2022)

Software tools for the seasonal adjustment of weekly data

- MoveReg (SAS, Eviews) (Cleveland et al., 2014)
- FAM (Ecce Signum)  (McElroy and Livsey, 2022)
- boiwsa ☺ 

Methodology



t denotes the date of the last day within a given week.

Methodology

Similarly to Cleveland et al. (2014), the seasonal component is modeled as

$$S_t = \sum_{k=1}^{K} \left(\alpha_k^y \sin\left(\frac{2\pi k D_t^y}{n_t^y}\right) + \beta_k^y \cos\left(\frac{2\pi k D_t^y}{n_t^y}\right)\right) + \sum_{l=1}^{L} \left(\alpha_l^m \sin\left(\frac{2\pi l D_t^m}{n_t^m}\right) + \beta_l^m \cos\left(\frac{2\pi l D_t^m}{n_t^m}\right)\right),$$

where D_t^y and D_t^m are the day of the year and the day of the month, and n_t^y and n_t^m are the number of days in the given year or month.

Thus, the seasonal adjustment procedure takes into account the existence of two cycles, namely intra-yearly and intra-monthly.

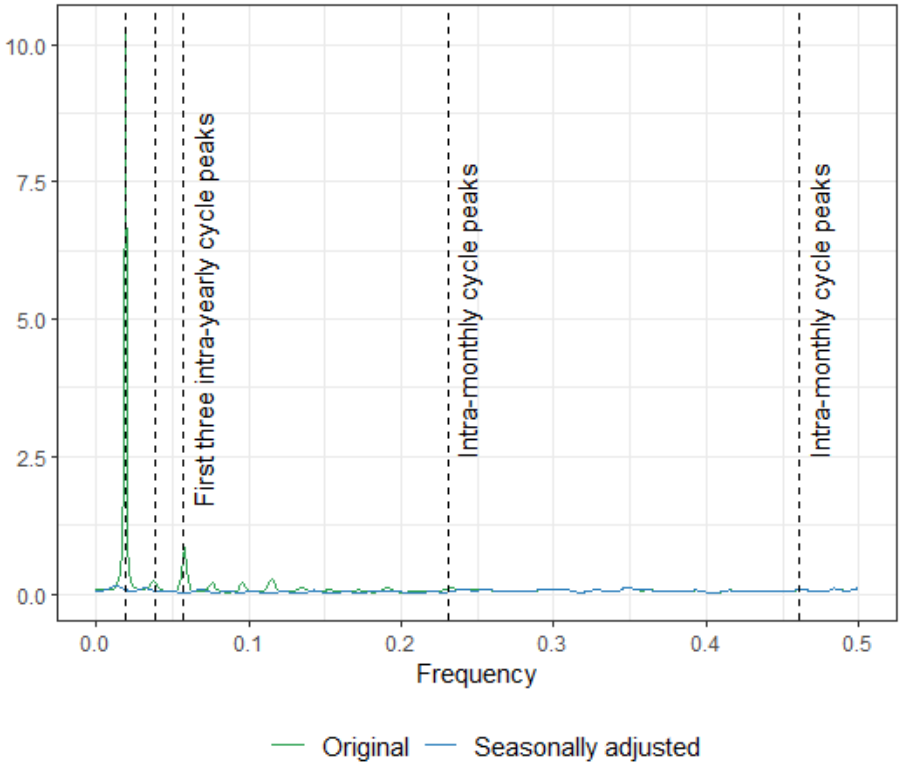
Methodology

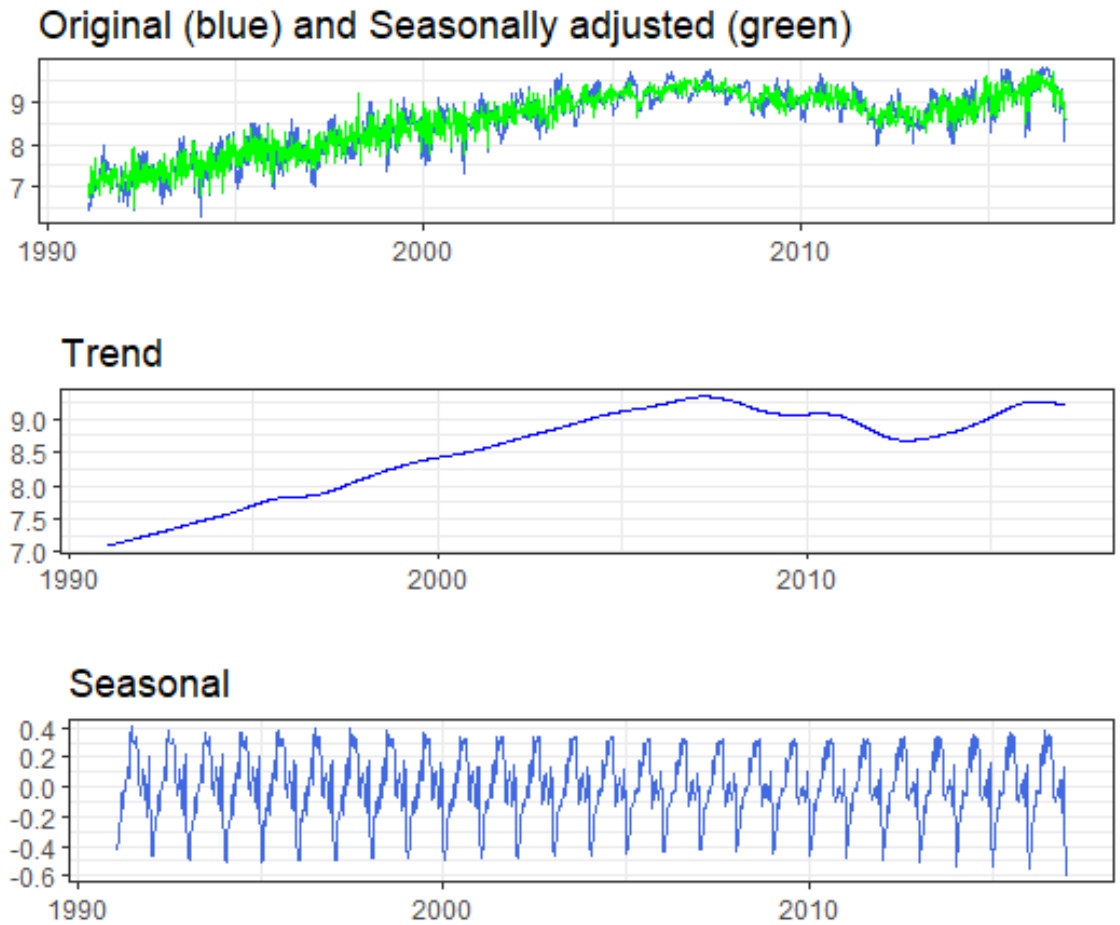
Similarly to the X-11 method (Ladiray and Quenneville, 2001), the boiwsa procedure employs an iterative approach to estimate the different components. The seasonal adjustment algorithm comprises eight steps, which are detailed below:

1. Estimation of trend using stats::supsmu().
2. Estimation of the Seasonal-Irregular component; Searching for additive outliers; Identifying the optimal number of trigonometric variables.
3. Calculation of seasonal factors.
4. Estimation of trend from seasonally and outlier adjusted series.
5. Estimation of the Seasonal-Irregular component.
6. Computing the final seasonal factors.
7. Estimation of the final seasonally adjusted series.
8. Computing the final trend.

Gasoline Consumption in the US

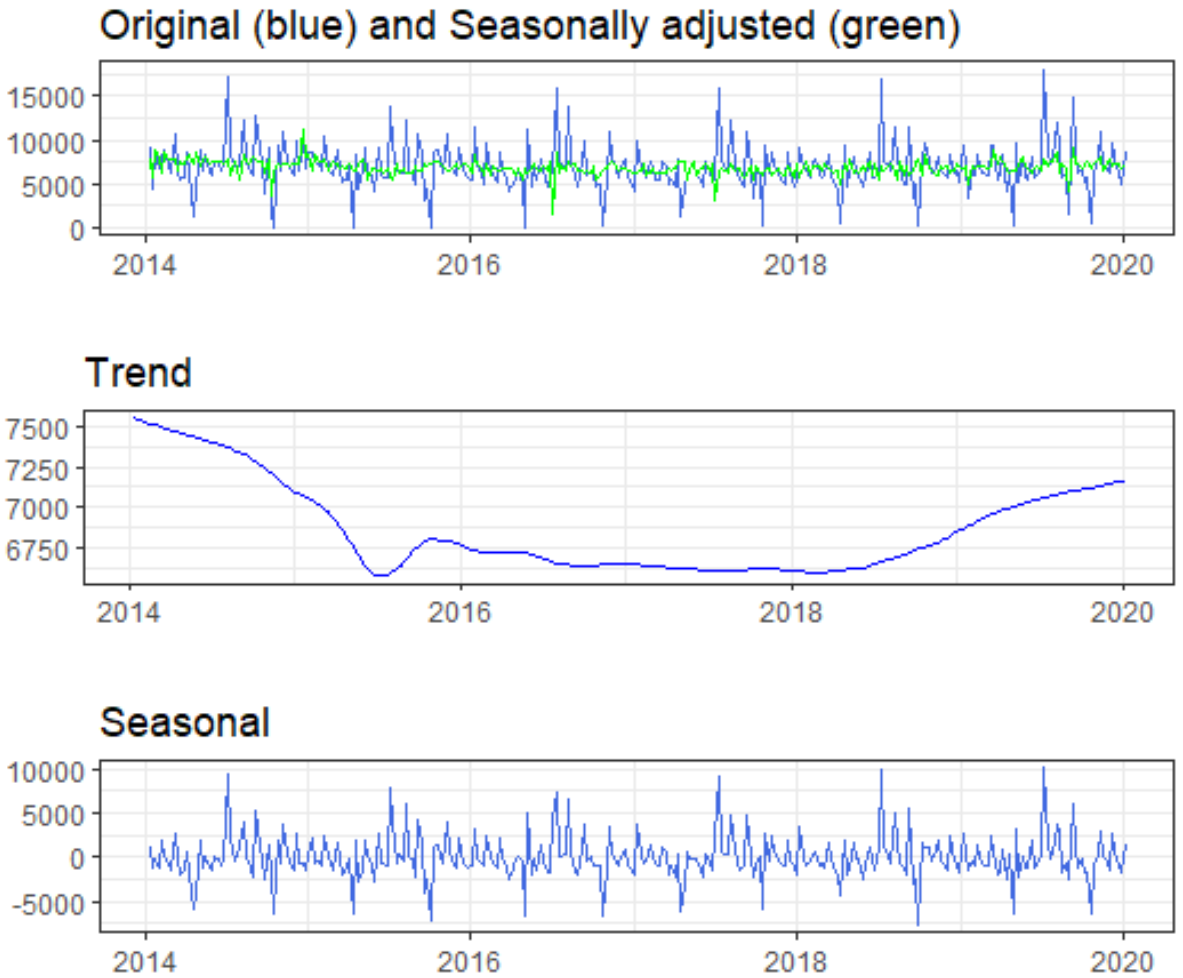
```
```{r}
res=boiwsa(x=gasoline.data$y,
           dates=gasoline.data$date)
```
```





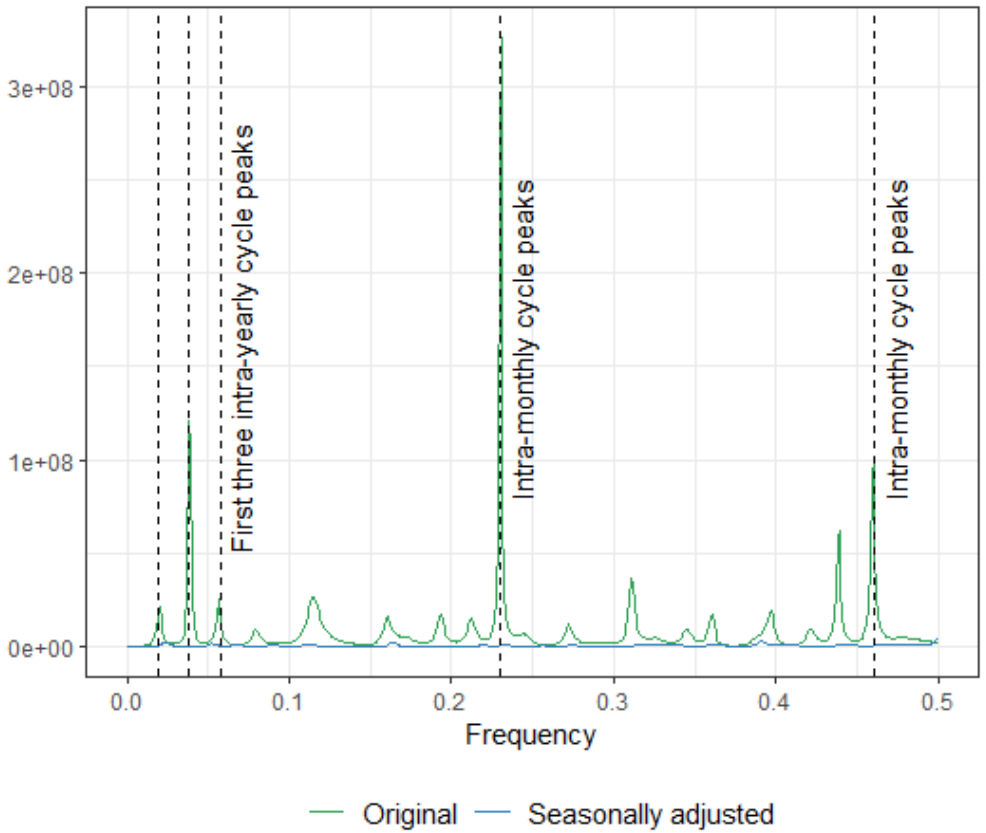
Weekly number of initial registrations in the Israeli Employment Service

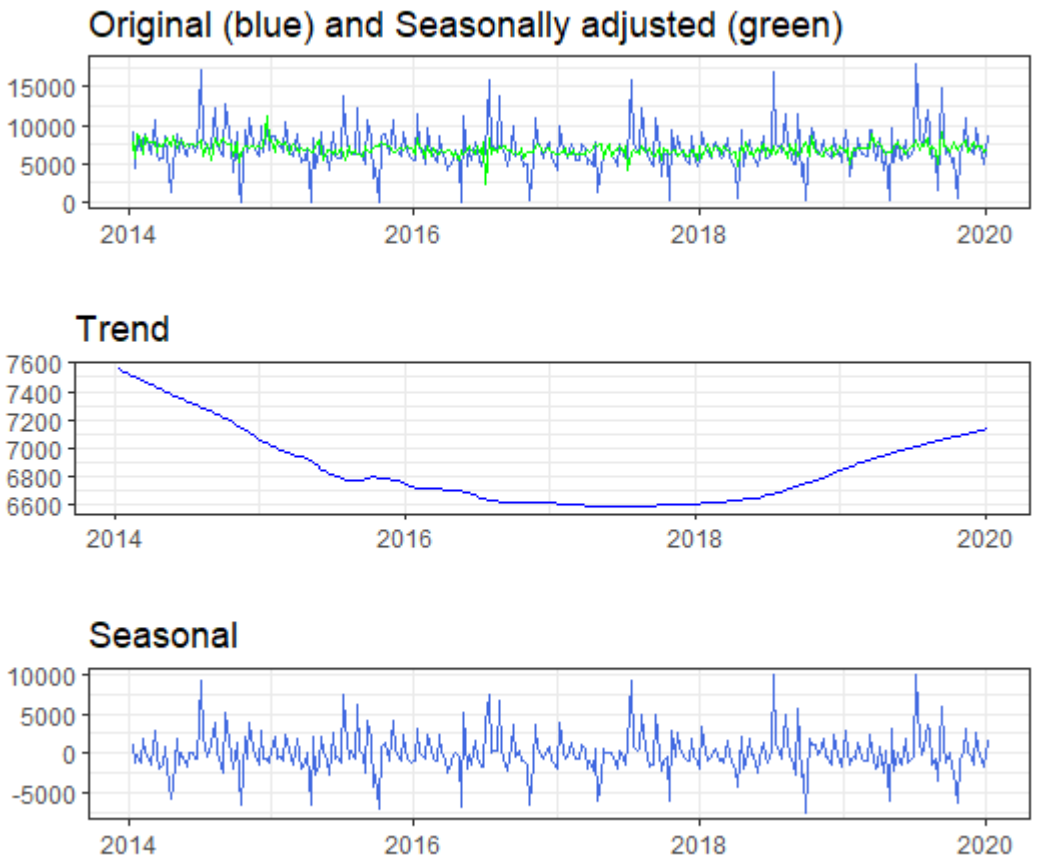
```r
# creating an input for simple_td
dates_il%>%
  dplyr::select(DATE_VALUE,ISR_WORKING_DAY_PART)%>%
  `colnames<-`(c("date","WORKING_DAY_PART"))%>%
  dplyr::mutate(date=as.Date(date))->df.td
# creating a matrix with a working day variable
td=simple_td(dates = lbm$date,df.td = df.td)

# generating Rosh Hashanah and Pesach moving holiday variables
rosh=my_rosh(dates=lbm$date,
             holiday.dates = holiday_dates_il$rosh)
# naming (make sure that all the variables in H have distinct names)
colnames(rosh)=paste0("rosh",colnames(rosh))

pesach=my_rosh(dates=lbm$date,
               holiday.dates = holiday_dates_il$pesah,
               start=3,end=-1)
colnames(pesach)=paste0("pesach",colnames(pesach))

# combining our working day and moving holiday variables
H=as.matrix(cbind(rosh[,-1],pesach[,-1],td[,-1]))
# running seasonal adjustment routine
res=boiwsa(x=lbm$IES_IN_W_ADJ,
           dates = lbm$date,
           H=H,
           out.threshold = 3.8)
```



Weekly number of initial registrations in the Israeli Employment Service

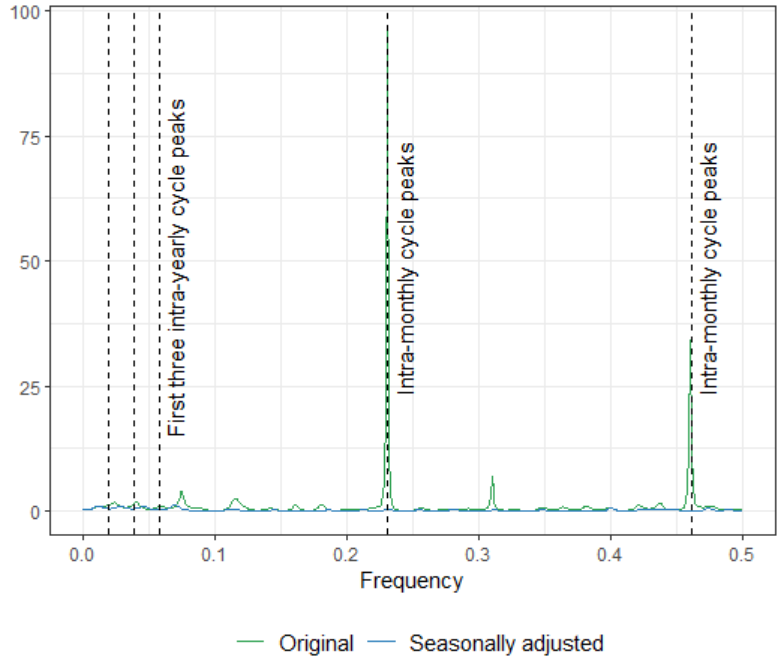
```r
res=boiwsa(x=lbm$IES_IN_W_ADJ,
           dates = lbm$date,
           H=H,
           out.threshold = 5)
```

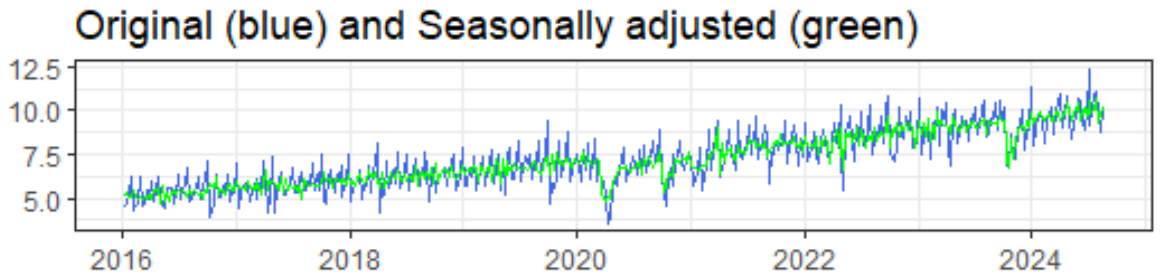


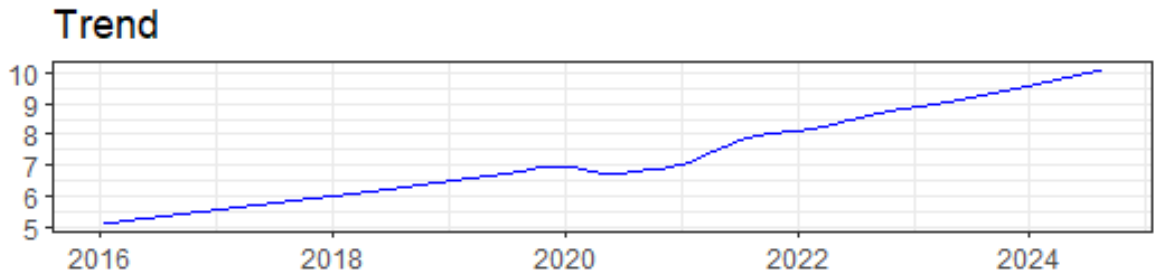


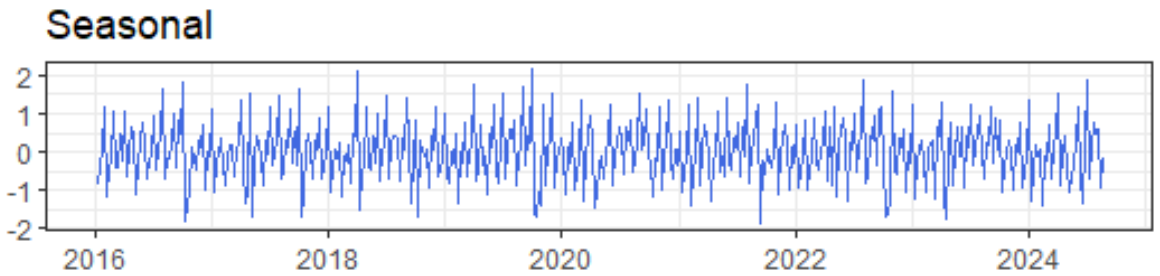
Weekly Credit Card Purchases

```r
pesach=genhol(dates = dates,holiday.dates = holiday_dates_il$pesah,start=0, end=15)
rosh=genhol(dates = dates,holiday.dates = holiday_dates_il$rosh,start=4, end=28)

dates_il%>%
  dplyr::select(DATE_VALUE,ISR_WORKING_DAY_PART)%>%
  `colnames<-`(c("date","WORKING_DAY_PART"))%>%
  dplyr::mutate(date=as.Date(date))->df.td

td=simple_td(dates = dates,df.td = df.td)

H=cbind(rosh[,-1],pesach[,-1],td[,-1])
colnames(H)=c("rosh","pesach","td")

res=boiwsa(x=y,
           dates=dates,
           H=as.matrix(H),
           ic = "bic",
           my.k_l = c(6,12))
```









Thank You!