Practice Session 6
Visualization of data and univariate stationary time series

Oxana Malakhovskaya

September, 24, 2019

Create a new script and save it. All the lines of codes must be written in this script (even attaching packages, importing datasets etc.). For accomplishing the task you may need to install or/and attach packages. The names of the packages are not always given to you explicitly in the tasks. You should decide yourself which packages you need if you need any.

Task 1

1. Load dataset `Cigar` from the `plm` package. Read carefully the help information about this dataset to understand what each variable means.
2. Select only observations just for one year (1980) and remove outliers with sales greater than 200 or smaller than 100. Assign the subsample to the variable named `cigar80`. Draw a scatterplot of `cigar80` using per capita income on x-axis and `sales` on y-axis.
3. Select from `cigar80` set variables starting with "pop" and sales. Create a new variable named `youth_ratio` equal to the ratio of kids and teenagers under 16 in the entire population of a state. Draw a scatterplot using `youth_ratio` on x-axis and `sales` on y-axis.
4. From original `Cigar` dataset Select variables `price`, `cpi`, and `sales`. Create a new variable named `real_pr` equal to the real price of a pack (use `cpi` as a deflator). Draw a scatterplot using `real_pr` on x-axis and `sales` on y-axis.
5. Sort `Cigar` according `sales` in descending order. Remove from the original `Cigar` dataset all the observations from three states with the largest cigarette sales per capita. Select from this new dataset variables `price`, `cpi`, and `sales` and create a `real_pr` variable as before. Draw a scatterplot using `real_pr` on x-axis and `sales` on y-axis.
6. Let us check if a negative correlation between price and sales that is clear on the previous graph (if we mix up data for all years), persists for any particular year. From original `Cigar` dataset select variables `price`, `cpi`, `sales`, and `year`. Select only observations from years 1960, 1965,..., 1990 using `%%` operator. Add real price variable as before. Draw scatterplots using real price on x-axis, `sales` on y-axis and splitting the sample into subsamples according the year, so that observations from different years are drawn on separate scatterplots.
7. From the original `Cigar` dataset select the observations from any three states you like. Draw a plot with three subplots (one subplot for each state) using `year` on x-axis, `sales` on y-axis and `geom_line` as a geometric function.
8. Group the observations in the original `Cigar` dataset according the year, find average cigaettes sales for each year and draw a plot showing the dynamics of average sales in time.
9. Group the observations in the original `Cigar` dataset according the year, find the average, lowest and highest values of cigarettes sales for each year. Draw a plot showing the dynamics of average, lowest and highest sales in time. The line showing the highest sales must be red, the line showing average sales must be blue and the line showing the lowest sales must be green.

Task 2

1. Import the dataset in `fed_funds.csv`. Check up the class of the `fed_funds` variable and look at its structure.
2. This part of the task must be done with the pipe operator. Sort the observations so that the series starts from the oldest one, then remove the Date variable, then convert the series into a ts object (determine the frequency and the starting date inspecting the series). Name the new variable ffr (federal funds rate).
3. Check up the class of the `ffr` variabe and print out first 36 observations. Read help about the `window` command and limit the series by December, 2008. Assign the new time series to a variable named `ffr2`.
4. Make a plot of `ffr2`. Axes must not have names, the plot should be named as "Federal Funds rate"
5. Find the first difference of `ffr2` and assign the value to the `dffr` variable.
6. Plot autocorrelation and partial autocorrelation functions of `dffr`.
7. Test if there is any autocorrelation (up to 10 lags) in the series `dffr` with Ljung-Box test.
8. Create a zero matrix A with 5 rows and 5 columns. Make loops for i and j that go from 0 to 4 (the loop for j must be inside of the loop for i). Inside of the two loops estimate an ARMA(i,j) model with a constant and assign it to the `model` variable. During each iteration assign AIC from the current model to (i+1, j+1) element of the matrix A (use `model$aic` to extract the AIC value from your ARMA model).
9. Print the minimum element of the matrix A. Name rows of the matrix A as `p=0 ,..., p=4`. Name columns of the matrix A as `q=0,...,q=4`. Print A matrix. Print AIC for the case when `dffr` series is fitted to the white noise process. Print AIC values for all models with 3 autoregressive lags.