**Topics of exercises**

* Data pre-processing with R

**Textbooks and readings**

Robert I. Kabacoff: R in Action. Data analysis and graphics with R. Manning Publications Co., 2011, **sections 5.4-5.6**

**Exercise 2.1**

1. Download and unpack the [florist.txt.zip](http://home.agh.edu.pl/~mszpyrka/lib/exe/fetch.php?media=lectures:dm:florist.txt.zip) file.
2. Load the file as **florist** data frame. Display a few record using the **head** function.
3. Select roses sales records and save them as **roses** data frame (**hint:** You can use the **subset** R function).
4. Use the **aggregate** function to calculate how many roses bouquets were sold each day (see **Examples**). Generate a scatter plot for the **roses\_by\_day** data. Can find some outliers using the plot? Find the outliers and look at dates. Do you think that these records are correct?

**Examples:**

florist <- read.table("florist.txt", header = T, sep = ",")

roses\_by\_day <- aggregate(roses$price, by = list(roses$date), FUN = length)

names(roses\_by\_day) <- c("date", "number\_of\_bouquets")

**Exercise 2.2**

1. Use the **rose** data frame and the **aggregate** function to calculate total sales value of roses in particular weeks. Save the result as **roses\_by\_week** data frame. Set the columns names to **year**, **week**, and **total**. Order the rows by **year** and **week**.
2. Plot the **roses\_by\_week$total** column using the **line** (**l**) style.
3. Implement the **binning** function that computes the result of smoothing by bin. The function takes as its arguments
   * the input data,
   * the bin size (the last bin may be incomplete),
   * the function used to compute the new value for the *i*th element.
4. Use the **binning** function to compute the result of smoothing **roses\_by\_week$total** by bin mean and median (**roses\_bin\_1**, **roses\_bin\_2**). Use the **bin\_size** parameter equal to 5.
5. Plot **roses\_by\_week$total**, **roses\_bin\_1**, and **roses\_bin\_2** on one graph and analyse the results.

**Piece of code:**

binning <- function(x, bin\_size, FUN = mean) {

len <- length(x)

y <- vector()

for(i in 1:len) {

...

}

return(y)

}

**Exercise 2.3**

1. Implement the **bin\_boundaries** function that computes the result of smoothing by bin boundaries. The function takes as its arguments
   * the input data,
   * the bin size (the last bin may be incomplete).
2. Use the **bin\_boundaries** function to compute the result of smoothing **roses\_by\_week$total** by bin boundaries. Use the **bin\_size** parameter equal to 5. Save the result as **roses\_bin\_3**.
3. Plot **roses\_by\_week$total**, **roses\_bin\_1**, **roses\_bin\_2**, and **roses\_bin\_3** on one graph and analyse the results.

**Exercise 2.4**

1. Use the **florist** data frame and the **aggregate** function to calculate total sales value of all types of bouquets in 2014 in particular sender province. Save the result as **bouquets\_by\_province** data frame. Set the columns names to **province** and **total**.
2. Use the **binning** function to compute the result of smoothing **bouquets\_by\_province$total** by bin mean using bin size: 3, 4, and 5.
3. Plot all results on one graph and analyse it.

**Exercise 2.5**

1. Normalize **bouquets\_by\_province$total** using min-max normalization (min = 0, max = 1). Save the result as **province\_norm\_1**.
2. Normalize **bouquets\_by\_province$total** using z-score normalization. Save the result as **province\_norm\_2**.
3. Normalize **bouquets\_by\_province$total** using decimal scaling normalization. Save the result as **province\_norm\_3**.
4. Plot **province\_norm\_1**, **province\_norm\_2**, and **province\_norm\_3** on one graph and analyse the results.

**Exercise 2.6**

1. Use the **florist** data frame and the **aggregate** function to calculate total sales value of all types of bouquets in particular months. Save the result as **florist\_by\_month** data frame. Set the columns names to **year**, **month**, and **total**. Order the rows by **year** and **month**.
2. Normalize **roses\_by\_month$total** using min-max normalization (min = 0, max = 1). Save the result as **roses\_norm\_1**.
3. Normalize **roses\_by\_month$total** using z-score normalization. Save the result as **roses\_norm\_2**.
4. Normalize **roses\_by\_month$total** using decimal scaling normalization. Save the result as **roses\_norm\_3**.
5. Plot **roses\_norm\_1**, **roses\_norm\_2**, and **roses\_norm\_3** on one graph and analyse the results.

**Exercise 2.7**

1. Use the **florist** data frame and the **aggregate** function to calculate total sales value of all types of bouquets in particular weeks. Save the result as **florist\_by\_week** data frame. Set the columns names to **year**, **week**, and **total**. Order the rows by **year** and **week**.
2. Select arbitrary cut points and use the **cut** function to assign the following level to the weeks: **low\_sale**, **normal\_sale**, **high\_sale**, **extreme\_sale**. Use **table** function to analyse the data.
3. Plot a histogram for the obtained data.

**Exercise 2.8**

1. Use the **florist** data frame and the **aggregate** function to calculate the total number of all types of bouquets sold in particular weeks. Save the result as **bouquets\_by\_week** data frame. Set the columns names to **year**, **week**, and **total**. Order the rows by **year** and **week**.
2. Select arbitrary cut points and use the **cut** function to assign the following level to the weeks: **low\_sale**, **normal\_sale**, **high\_sale**, **extreme\_sale**. Use **table** function to analyse the data.
3. Plot a histogram for the obtained data.
4. Compare the histogram with the histogram from Exercise 2.4. Are they similar?
5. Compute the Pearson's linear correlation coefficient for **florist\_by\_week$total** and **bouquets\_by\_week$total**. How do we interpret the result?