Data Programming

Homework 3

Lucas Lorenzo Jakin

2025-04-01

Solutions

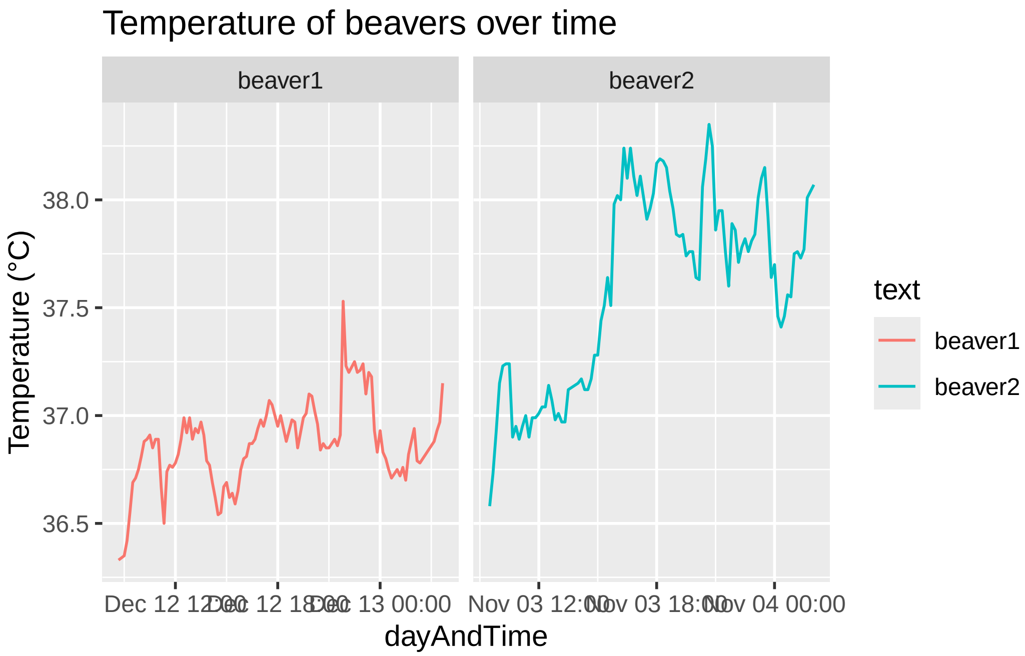
Submit a **.html** file created using Quarto via e-classroom. A sample is attached. Display all code (packages,input) and output.

1

Draw a line chart with ggplot of temperature over time side by side for the 2 build-in data frames **beaver1** and **beaver2**. Add a Title and units on the Y-axis for temperature.

```r
# Solve 1 here!
library(tidyverse)
library(ggplot2)

#BEAVER1
# day to date format
beaverMonth <- as.Date(beaver1$day, origin = "1989-12-31")
#time to format
hour <- beaver1$time %/% 100 # divide by 100 to get hour from time
minutes <- beaver1$time %% 100 # in order to get the minutes take the rest from before, so
combinedTime <- sprintf("%02d:%02d", hour, minutes)
beaver1$dayAndTime <- ymd_hm(paste(beaverMonth,combinedTime))

#BEAVER2
# day to date format
beaverMonth2 <- as.Date(beaver2$day, origin = "1989-12-31")
#time to format
hour <- beaver2$time %/% 100 # divide by 100 to get hour from time
```

```
minutes <- beaver2$time %% 100 # in order to get the minutes take the rest from before, so
combinedTime <- sprintf("%02d:%02d", hour, minutes)
beaver2$dayAndTime <- ymd_hm(paste(beaverMonth2,combinedTime))

beaver1$text <- "beaver1"
beaver2$text <- "beaver2"

groupedData <- bind_rows(beaver1,beaver2) #combine both data frames to plot them together

#plotting the data frames
ggplot(groupedData, aes(x=dayAndTime, y = temp, color = text)) +
  geom_line() +
  facet_wrap(~text, scales = "free_x") + labs(title = "Temperature of beavers over time",
  theme(legend.position = "right")
```



2

You are given the worth of 1 EUR in some currencies.

```
c(EUR=1, datasets::euro)
```

```
         EUR          ATS          BEF          DEM          ESP          FIM
    1.000000    13.760300    40.339900     1.955830   166.386000     5.945730
         FRF          IEP          ITL          LUF          NLG          PTE
    6.559570     0.787564  1936.270000    40.339900     2.203710   200.482000
```

Write a function of two parameters:

1. a named numeric vector, with values representing the amount of currency in the corresponding name owned,
2. (optional) a currency to convert to.

The function returns the total amount from 1. as the currency specified in 2. If currency in 2. is not specified then return the value in EUR. If a currency from 1. does not exists, then that part of the vector contributes 0 to the output. If the currency from 2. does not exist then return 0.

```
# Solve 2 here (Enter the function)

convert.all <- function(amount, currency = "EUR"){
  # Define conversion rates
  currencies <- c(EUR = 1, datasets::euro)
  # if currency doesn't exist return 0
  if (!(currency %in% names(currencies))){
    return(0)
  }
  converted_values <- amount / currencies[names(amount)]
  sum_amount <- sum(converted_values, na.rm=T) # na.rm for taking care of NA values (if cu
  return (sum_amount * currencies[currency])
}

convert.all(c(EUR=6.5))
```

```
EUR
6.5
```

```
convert.all(c(EUR=3,ATS=2))
```

```
     EUR
3.145346
```

```
convert.all(c(EUR=3,EUR=2))
```

```
EUR
  5
```

```
convert.all(c(EUR=3,DEM=2,SIT=10000))
```

```
     EUR
4.022584
```

```
convert.all(c(EUR=3,DEM=2,SIT=10000),"EUR")
```

```
     EUR
4.022584
```

```
convert.all(c(EUR=3,SIT=10000,DEM=2),"ESP")
```

```
     ESP
669.3016
```

```
convert.all(c(EUR=3,DEM=2,LIRA=100),"SIT")
```

```
[1] 0
```