HW1: Markdown file

Kevin Liu 2023-09-18

Problem 1

1. “Load the ‘moderndive’ libary…”

```
library(moderndive)
```

- *For future reference, you can name blocks of code for identification when R throws errors when knitting. ie. it’s a good idea to name the above code as “r setup” because you are ‘setting up’ your markdown by importing appropriate libraries.*
- *You can also use “include=FALSE” within the brackets (ie. {r setup, include=FALSE}) to hide a block of code.*

2. “…and use the following code to load the ‘early_january_weather’ dataset.”

Write a short description of the dataset using inline R code; accessing the dataset help file can be informative. In your discussion, please include:

- the variables in this dataset, including names / values of important variables
- the size of the dataset (using nrow and ncol)
- the mean temperature”

```
#The eval, code output, is hidden for viewing clarity.

# Import Dataframe "early_january_weather")
data("early_january_weather")

#View Dataset
View(early_january_weather)

#skimr to help summarize data
skimr::skim(early_january_weather)
```

Response:

1. **Variables in Dataset:** Of the variables, 1 is “chr” type, 4 are “int” types, 9 are “num” types, and 1 is “POSIXct” type (some kind of time format). The 4 int type variables (year, month, day, hour) record the time and seem to be summarized by the “POSIXct” type “time_hour” by combining all 4 variables into one. The 9 “num” types are all some measurement of the climate (temp, dewp, humid, wind_dir, wind_speed, wind_gust, precip, pressure, visib) supposedly relating to flying conditions. The final “chr” type seems to be the names of airports (i think “EWR” for “Newark”).

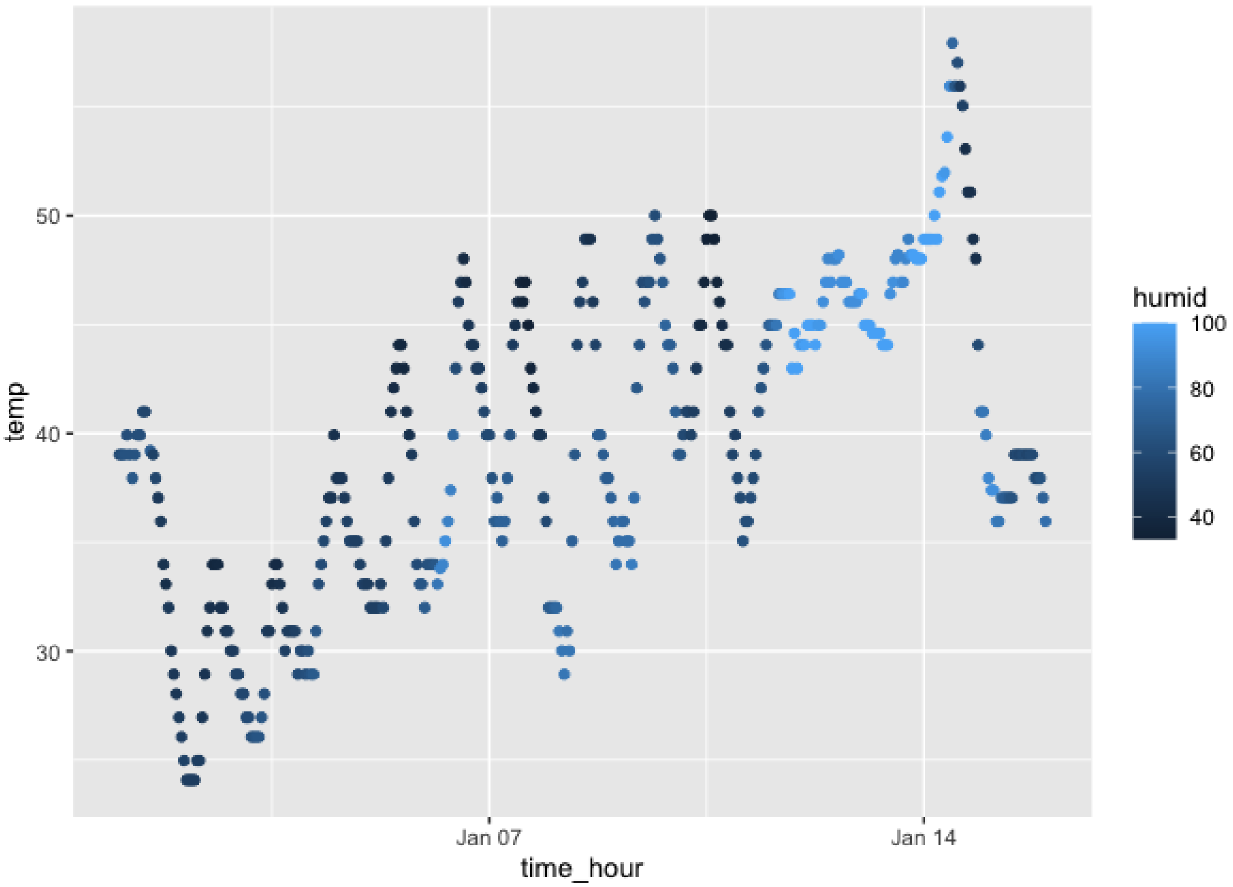
2. **Size of Dataset:** The dataframe appears to have 358 rows (nrow = 358) of information each with 15 variables (ncol = 15).

3. **Mean Temperature:** From the skim function in the skimr package, we can see that the mean temperature is 39.6.

3. “Make a scatterplot of…”

…temp (y) vs time_hour (x); color points using the humid variable (adding color = … inside of aes in your ggplot code should help). Describe patterns that are apparent in this plot.

```
ggplot(early_january_weather, aes(x = time_hour, y = temp, color = humid)) + geom_po
```



- **Description**: As time_hour increases, temp gradually increases up to around “Jan 14” where temperature exhibits a sharp drop. Humid, as seen in the color intensity, also exhibits an overall gradual increase up to around “Jan 14”. Amidst this overall trend, temp exhibits fluctuations as seen in alternating peaks and troughs. Humid, though difficult to tell, also exhibits some fluctuation. Visually it’s difficult to tell if there’s a relationship between humid and temp.

Export your scatterplot to your project directory using ggsave.

```
ggsave("scat_weather.pdf")
```

```
## Saving 7 x 5 in image
```

Problem 2

This problem is intended to emphasize variable types and introduce coercion; some awareness of how R treats numeric, character, and factor variables is necessary for working with these data types in practice.

1. Create a data frame comprised of:

- a random sample of size 10 from a standard Normal distribution
- a logical vector indicating whether elements of the sample are greater than 0
- a character vector of length 10
- a factor vector of length 10, with 3 different factor “levels”

```
problem2_df =
  tibble(
    rand_samp = rnorm (10),
    l_vec = rand_samp > 0,
    c_vec = character(10),
    f_vec = factor(c("Apple", "Apple", "Pear", "Orange", "Apple", "Apple", "Pear", '
  )
```

2. Create a data frame comprised of:

Try to take the mean of each variable in your dataframe. What works and what doesn’t?

```
#Take Mean (The variables were pulled, the code is hidden for clarity)
mean(rand_samp_p)
```

```
## [1] 0.03477211
```

```
mean(l_vec_p)
```

```
## [1] 0.5
```

```
mean(c_vec_p)
```

```
## Warning in mean.default(c_vec_p): argument is not numeric or logical: returning
## NA

## [1] NA
```

```
mean(f_vec_p)
```

```
## Warning in mean.default(f_vec_p): argument is not numeric or logical: returning
## NA

## [1] NA
```

Description: The mean of the random sample vector and the logic vector were taken. It seems that the mean was not able to be done on the factor vector and the character vector.

3. In some cases…

…you can explicitly convert variables from one type to another. Write a code chunk that applies the as.numeric function to the logical, character, and factor variables (please show this chunk but not the output). What happens, and why? Does this help explain what happens when you try to take the mean?

```
#To prevent output, add "eval = FALSE" into codeblock setup.
as.numeric(rand_samp_p)
as.numeric(l_vec_p)
as.numeric(c_vec_p)
as.numeric(f_vec_p)
```

The character vector was and the factor vector threw warning messages that indicated that they were not numerical or logical variable types and thus numerical functions are not able to be performed on these variable types.