Journal (reproducible report)

Johannes Julius Halbe

2020-11-27

Contents

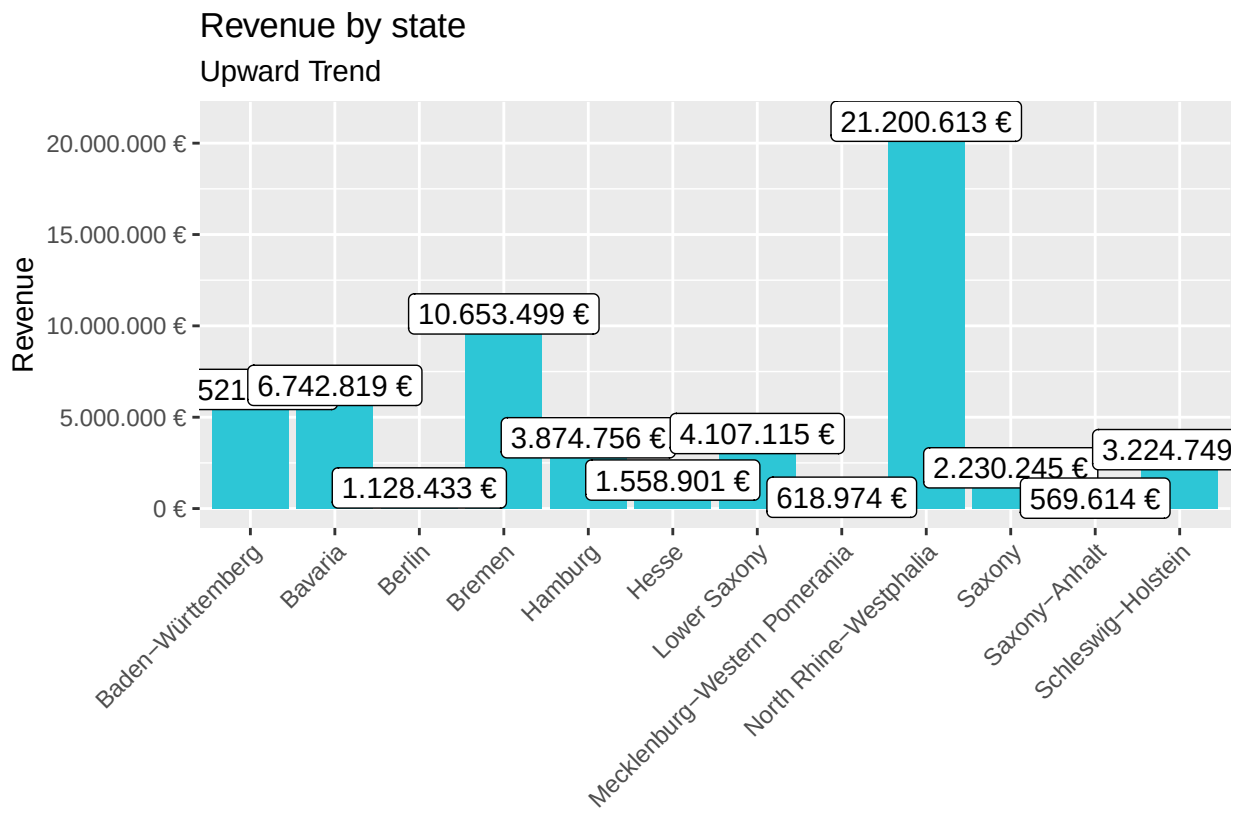
This is an `.Rmd` file. It is plain text with special features. Any time you write just like this, it will be compiled to normal text in the website. If you put a # in front of your text, it will create a top level-header.

Challenge 1

Last compiled: 2020-12-04

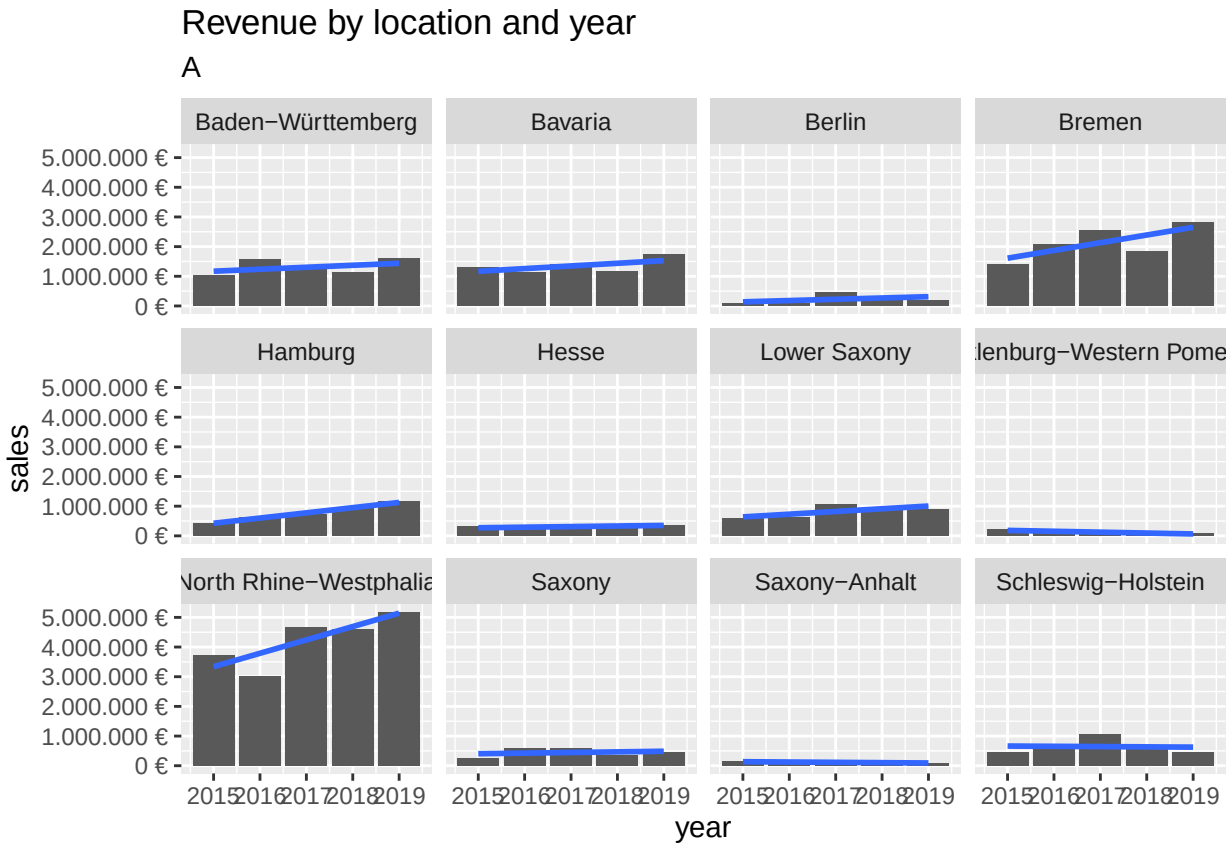
```
# Challenge 01 ----

# 1.0 Load libraries ----
library(tidyverse)

#Excel Files
library(readxl)

# 2.0 Importing Files ----
bikes_tbl      <- read_xlsx("docs/00_data/01_bike_sales/01_raw_data/bikes.xlsx")
orderlines_tbl <- read_xlsx("docs/00_data/01_bike_sales/01_raw_data/orderlines.xlsx")
bikeshops_tbl  <- read_xlsx("docs/00_data/01_bike_sales/01_raw_data/bikeshops.xlsx")

# 3.0 Examining Data ----

#oderlines_tbl

#glimpse(orderlines_tbl)

#view(orderlines_tbl)

# 4.0 Joining Data ----

bike_orderlines_joined_tbl <- orderlines_tbl %>%
```

```r
  left_join(bikes_tbl, by = c("product.id" = "bike.id")) %>%
  left_join(bikeshops_tbl, by = c("customer.id" = "bikeshop.id"))

# 5.0 Wrangling Data ----
bike_orderlines_wrangled_tbl <- bike_orderlines_joined_tbl %>%
  select(-...1) %>%
  rename(bikeshop = name) %>%
  set_names(names(.) %>% str_replace_all("\\.", "_")) %>%
  separate(col    = location,
           into   = c("city", "state"),
           sep    = ", ") %>%
  mutate(total_price = price * quantity)


# 6.0 Business Insights ----
# 6.1 Sales by location ----

# Step 1 - Manipulate

sales_by_location_tbl <- bike_orderlines_wrangled_tbl %>%
  select(state, total_price) %>%
  group_by(state) %>%
  summarize(sales = sum(total_price)) %>%
  mutate(sales_text = scales::dollar(sales, big.mark = ".",
                                     decimal.mark = ",",
                                     prefix = "",
                                     suffix = " €"))
# Step 2 - Visualize

sales_by_location_tbl %>%
  ggplot(aes(x = state, y = sales)) +
  geom_col(fill = "#2DC6D6") +
  geom_label(aes(label = sales_text)) +
  geom_smooth(method = "lm", se = FALSE) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  scale_y_continuous(labels = scales::dollar_format(big.mark = ".",
                                                    decimal.mark = ",",
                                                    prefix = "",
                                                    suffix = " €")) +
  labs(
    title    = "Revenue by state",
    subtitle = "Upward Trend",
    x = "",
    y = "Revenue"
  )
```




```
# 6.2 Sales by location & year ----

# Step 1 - Manipulate
library(lubridate)

sales_by_location_year_tbl <- bike_orderlines_wrangled_tbl %>%


  select(state, total_price, order_date) %>%
  mutate(year = year(order_date)) %>%
  group_by(state, year) %>%
  summarise(sales = sum(total_price)) %>%
  ungroup() %>%

  mutate(sales_text = scales::dollar(sales, big.mark = ".",
                                     decimal.mark = ",",
                                     prefix = "",
                                     suffix = " €"))
# Step 2 - Visualize
sales_by_location_year_tbl %>%

  # Set up x, y, fill
  ggplot(aes(x = year, y = sales)) +

  # Geometries
  geom_col() + # Run up to here to get a stacked bar plot
```

```
  geom_smooth(method = "lm", se = FALSE) +

  # Facet
  facet_wrap(~ state) +

  # Formatting
  scale_y_continuous(labels = scales::dollar_format(big.mark = ".",
                                                    decimal.mark = ",",
                                                    prefix = "",
                                                    suffix = " €")) +

  labs(
    title = "Revenue by location and year",
    subtitle = "A"
  )
```



My second post (note the order)

Last compiled: 2020-12-04

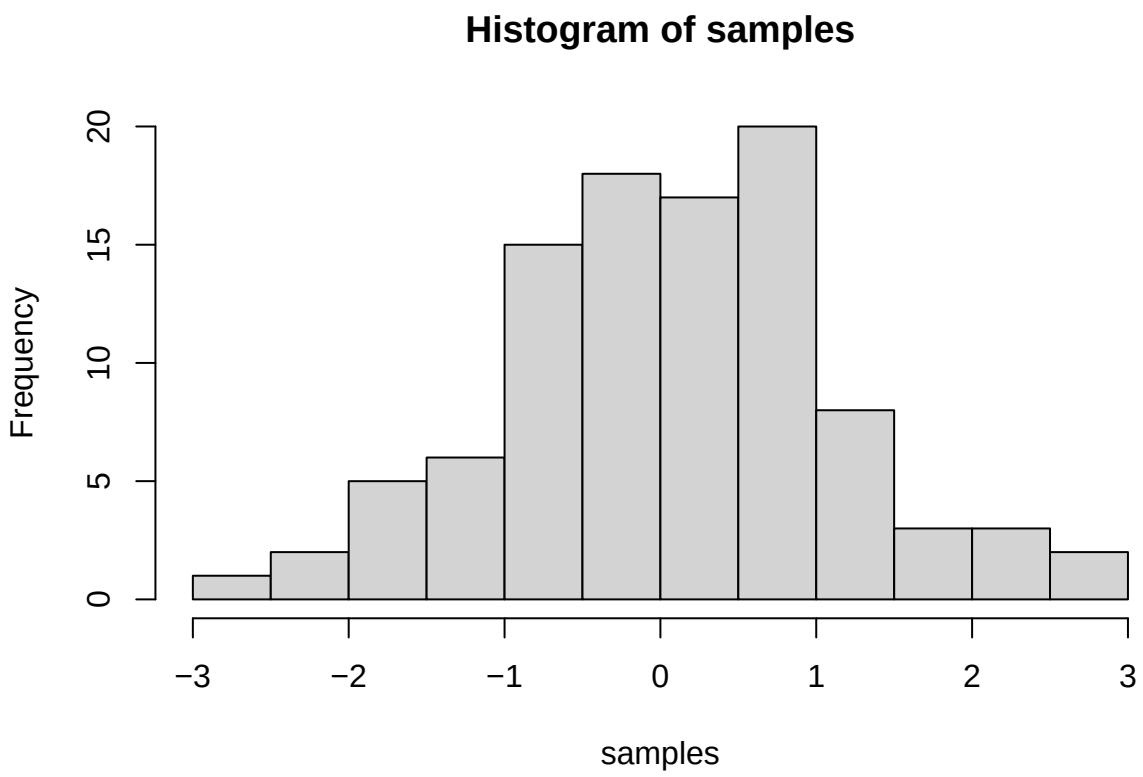
I'm writing this tutorial going from the top down. And, this is how it will be printed. So, notice the second post is second in the list. If you want your most recent post to be at the top, then make a new post starting at the top. If you want the oldest first, do, then keep adding to the bottom

Adding R stuff

So far this is just a blog where you can write in plain text and serve your writing to a webpage. One of the main purposes of this lab journal is to record your progress learning R. The reason I am asking you to use this process is because you can both make a website, and a lab journal, and learn R all in R-studio. This makes everything really convenient and in the same place.

So, let's say you are learning how to make a histogram in R. For example, maybe you want to sample 100 numbers from a normal distribution with mean = 0, and standard deviation = 1, and then you want to plot a histogram. You can do this right here by using an r code block, like this:

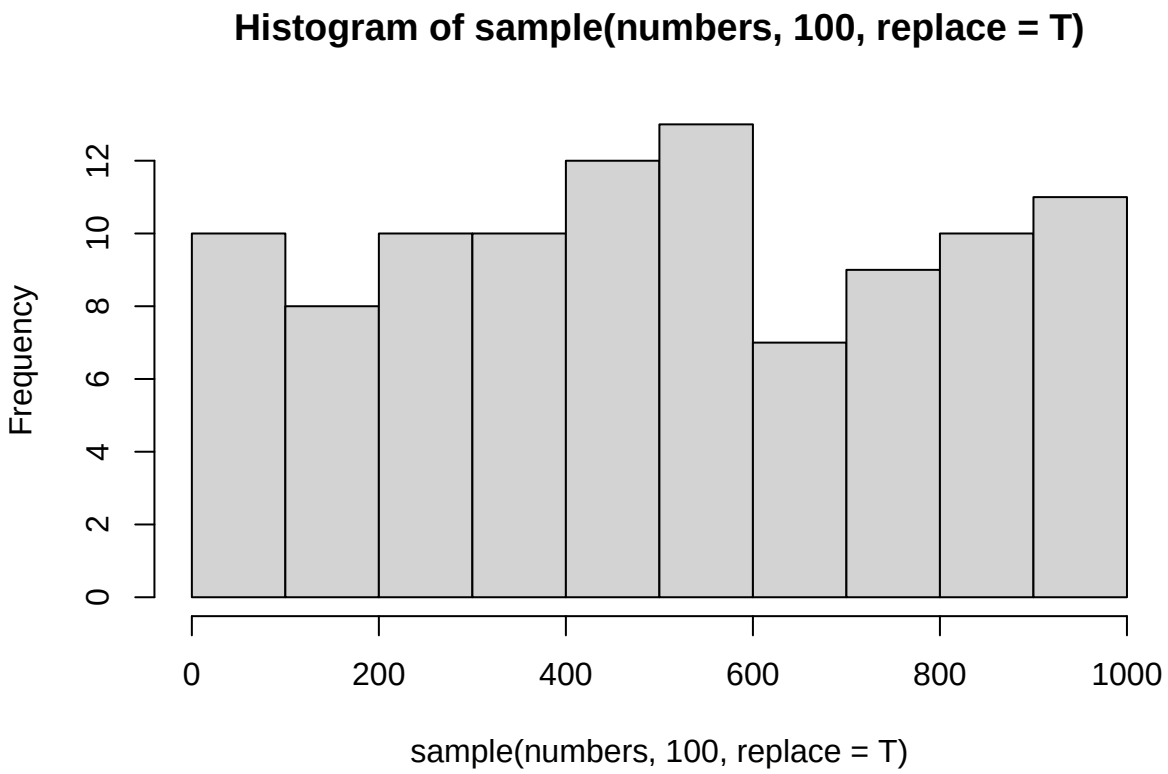
```
samples <- rnorm(100, mean=0, sd=1)
hist(samples)
```




```
numbers <- 1:1000

# This will print the first 10 elements of the vector numbers
numbers[1:10]
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
# This will plot a histogram of 100 random elements of the vector numbers
hist(sample(numbers, 100, replace = T))
```



When you knit this R Markdown document, you will see that the histogram is printed to the page, along with the R code. This document can be set up to hide the R code in the webpage, just delete the comment (hashtag) from the cold folding option in the yaml header up top. For purposes of letting yourself see the code, and me see the code, best to keep it the way that it is. You'll learn that all of these things and more can be customized in each R code block.