Table of Contents

Question 1

Assignment 1.1

Command for creating the volume:

‘docker volume create volume_ordinary’

Assignment 1.2

Looking at the file “run_0_exam.R”, it requires library “microbenchmark”, which is not in the docker file, so we need to add it in there before creating the image.

Corrected dockerfile:

```
‘
# Start from the rocker/rstudio base image which has R and RStudio pre-installed
FROM rocker/rstudio

# The RUN command executes shell commands during the image building process.
RUN apt-get update && apt-get install -y curl

# Install R packages using R's built-in 'install.packages' function.
RUN R -e 'install.packages(c("RPostgres", "microbenchmark"))'
‘
```

Assignment 1.3

Building image from “docker_file_0_exam”:

‘docker image build --tag rstudio:1.1.1 -f docker_file_0_exam .’

Running the container from the image:

```
‘
docker run -d --network db_r_shiny -p 8787:8787 -e PASSWORD=hidden123 --name rstudio -v volume_ordinary:/home/rstudio rstudio:1.1.1
’
```

Assignment 1.4

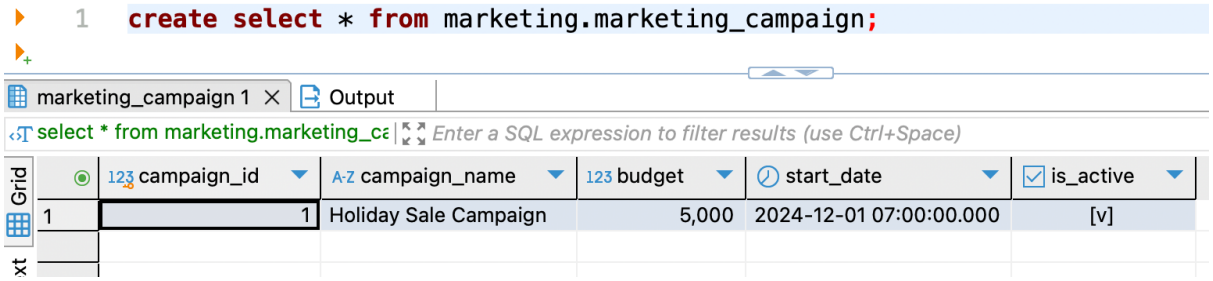
Output from the R console:

```
> library(microbenchmark)
> source("fun_0_exam.R")
[1] "16523266152.XX"
```

Question 2

Assignment 2.1

Screenshot of final table from DBeaver:



R script for this question can be found in the file: assignment_2.1.R

Assignment 2.2

Query:

```
'
select distinct -- Use distinct to ensure actors are listed once per
language
    a.actor_id,
    a.name_first,
    a.name_last,
    l.lang_name as language_name
from actor a
join film_actor fa on a.actor_id = fa.actor_id
join film f on fa.film_id = f.film_id
join language l on f.language_id = l.language_id
order by a.actor_id asc;
'
```

Query for this question can be found in the file: assignment_2.2.sql

Assignment 2.3

Explanation of the SCD type I used:

I implemented SCD Type 7, which is a combination of Type 1 (overwriting) and Type 2 (historical tracking).

product_current (Type 1):

- Contains only current values
- Immediately overwrites data when changes occur
- Has a product_durable_sk as a stable identifier that doesn't change
- Good for quick "as-is" queries and current state analysis

product_history (Type 2):

Keeps historical records

- Creates new rows when changes occur
- Tracks effective and ineffective dates
- Uses current_indicator to show active records
- Good for historical analysis and tracking changes over time

fact_sale:

- Links to both dimensions using product_sk and product_durable_sk
- Can access either current or historical product information
- Enables both current state and historical analysis

SQL code for this question can be found in the file: assignment_2.3.sql

Question 3

Assignment 3.1

1)
R code for this question can be found in the file: assignment_3.1.R

2)
Movie title and ID of the first movie in the response:

Title of first movie is "The Nightmare Before Christmas"

```
> print(result$search[[1]]$title)
[1] "The Nightmare Before Christmas"
```

ID of the first movie is “tt0107688”

```
> print(result$search[[1]]$id)
[1] "tt0107688"
```

Assignment 3.2

Output from the first assignment:

```
> file_path <- ("/home/rstudio/git_training/git_2024_1/fun_0_exam2.R")
> source(file_path)
> flow_id("82")
[1] "82.AA"
```

What I see in “example_conflict.R”:

I see a Git merge conflict.

The file contains a merge conflict between two different versions of the same code section, marked by Git's conflict markers:

<<<<<<< HEAD: marks the beginning of the changes in the current branch

=======: separates the conflicting changes

>>>>>>> 7e4f3b2c9a6d8f5b123456789abcdef12345678: marks the end of changes from the other branch (with the commit hash)

The conflict is between two different function calls:

- One version uses psql_append_df()
- The other version uses psql_manipulate()

I modified the file and chose to keep psql_append_df() because:

- The comment indicates this is for "inserting" datasets

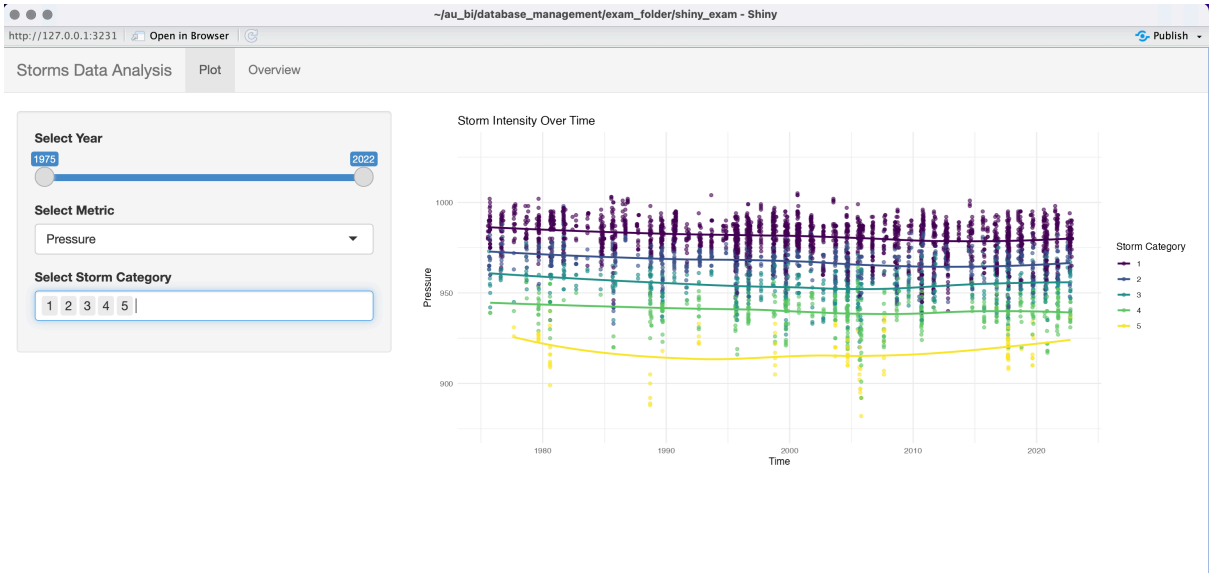
- append is more specific to the operation of adding data than the generic manipulate
- The append appears in the HEAD version, which typically represents the most current version of the code

The modified file can be found attached as the file: assignment_3.2.R

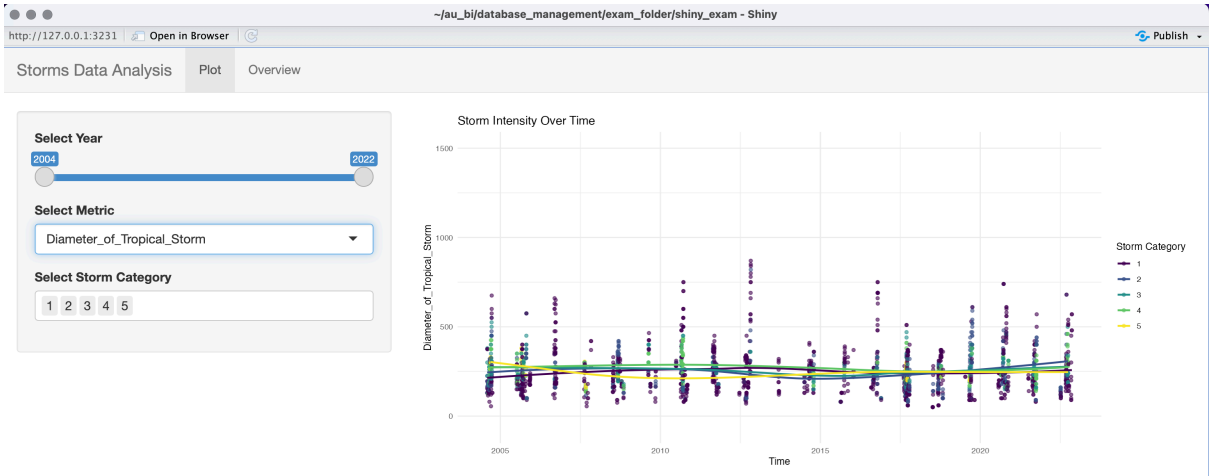
Question 4

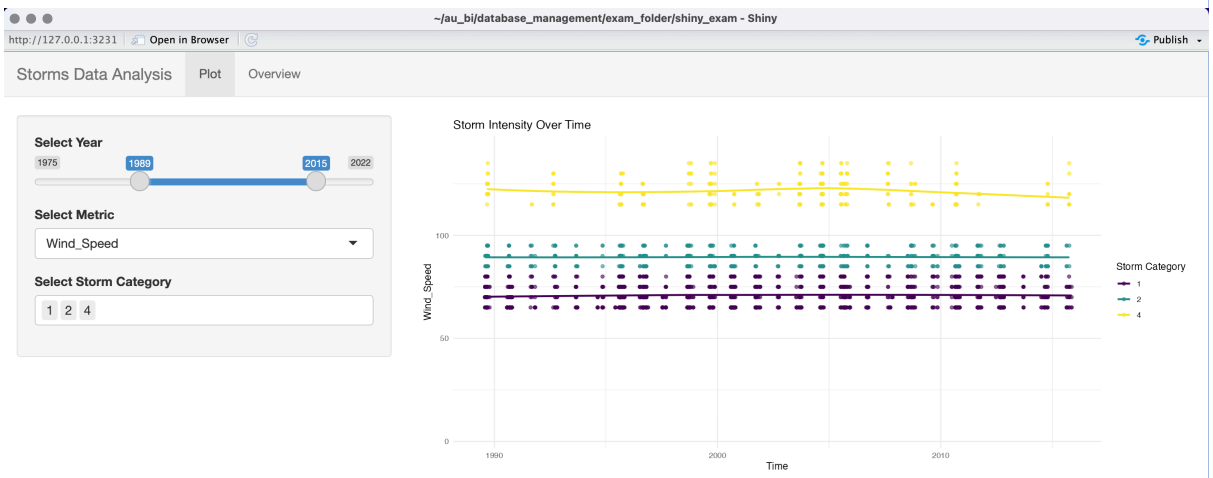
Assignment 4.1

Screenshots:



I implemented dynamic updates of the UI – I wanted to change the range of the Select Year based on the Metric selected.
I implemented the same logic for the X-axis of the plot to avoid weird looking plots – so the x-axis now dynamically adopts to the Metric selected.

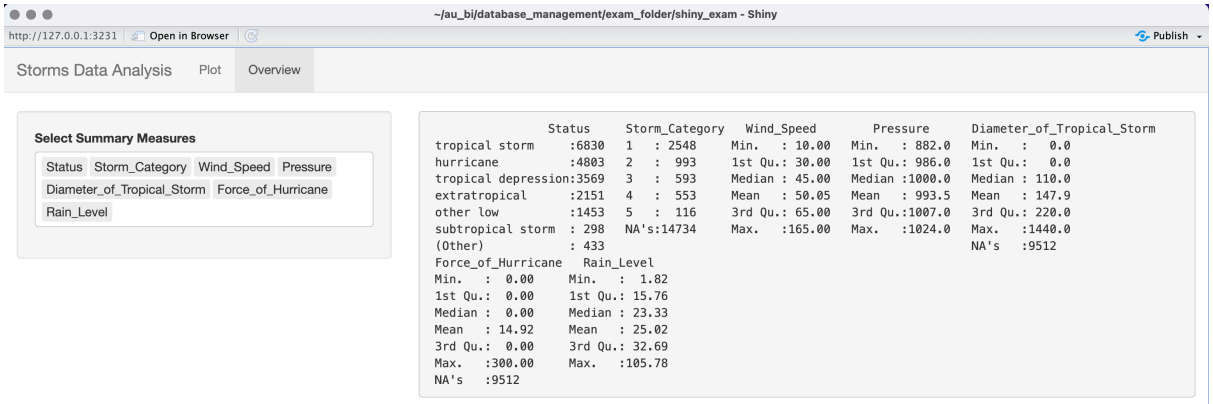


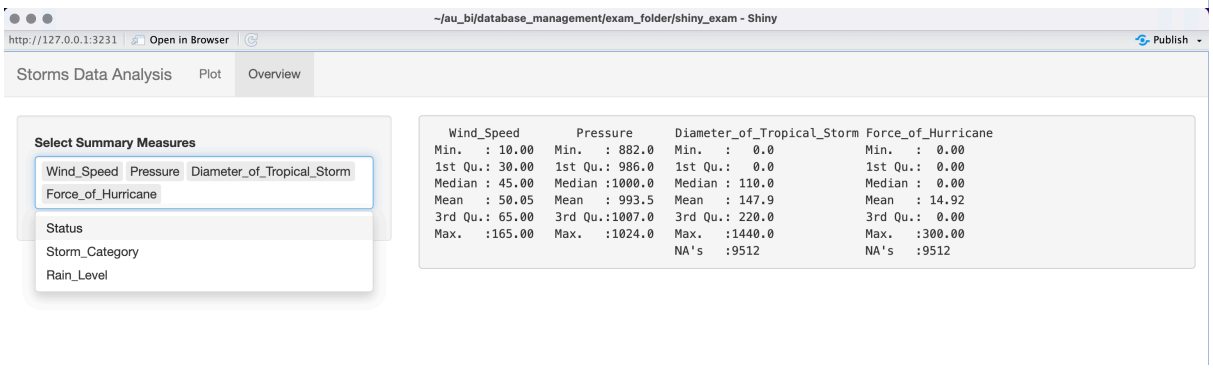


Assignment 4.2

I feel like the Summary Measure for Status does not make sense to display as character, so I will convert it to factor to see the different Status levels. Even though this differs from the Figure 4.2, I believe it will be more informative.

Screenshots:





R code for this question can be found in the file: assignment_4.R