Assignment #1 LLM Transcript - Fraz Aslam

The primary LLM I used was Claude. I did not need to use it as much this assignment as I followed the instructor example as the primary source for this assignment. I used the LLM to help me extract the data and information I was most interested in from the main data-set. I am new to R programming so Claude proved to be a useful tool in learning techniques that I wanted to perform.

[1] Creating a sub-set of data with certain information

I am currently able to display the data in my qmd and viewer. I have filtered to show the first 100 rows of the dataset. How would i go about displaying the data set in decreasing order of SNOW with a maximum number of rows of 100

Perfect! Here's how to sort by SNOW in decreasing order and show the top 100:

Solution:

```
df |>
  arrange(desc(SNOW)) |>  # Sort by SNOW column, highest first
  head(100) |>           # Take first 100 rows
  gt()
```

Breaking It Down:

- `arrange(desc(SNOW))` - Sorts the SNOW column in **descending** order (largest snowfall first)
- `head(100)` - Takes the first 100 rows (which are now the 100 snowiest days!)
- `gt()` - Displays as a nice table

Alternative: Show Top N Snowiest Days

If you want to be more explicit:

```
df |>
  arrange(desc(SNOW)) |>
  slice(1:100) |>  # Same as head(100)
  gt()
```

Bonus: Add a Title to Your Table

```
df |>
  arrange(desc(SNOW)) |>
  head(100) |>
  gt() |>
  tab_header(
    title = "Top 100 Snowiest Days in NYC Central Park",
    subtitle = "1869-2022"
  )
```

Show Only Relevant Columns

To make it cleaner, select just the important columns:

```
df |>
  arrange(desc(SNOW)) |>
  head(100) |>
  select(DATE, SNOW, SNWD, TMAX, TMIN) |>  # Pick relevant columns
  gt()
```

Try the first solution and you'll see the 100 snowiest days in NYC history! 🌨️

This will be great for your analysis comparing to this past weekend's storm!

[2] Further Iteration on previous data subset

what if i wanted to further refine this code block and only display the name with the SNOW data

Easy! Just select only the `DATE` and `SNOW` columns:

Solution:

```
df |>
  arrange(desc(SNOW)) |>
  head(100) |>
  select(DATE, SNOW) |>  # Only show DATE and SNOW columns
  gt()
```

Or if the date column has a different name:

Check what your date column is actually called. It might be `date`, `DATE`, or something else. You can check with:

```
# See all column names
names(df)
```

Then adjust accordingly:

```
df |>
  arrange(desc(SNOW)) |>
  head(100) |>
  select(date, SNOW) |>  # Use lowercase if that's the column name
  gt()
```

With a Nice Title:

```
df |>
  arrange(desc(SNOW)) |>
  head(100) |>
  select(DATE, SNOW) |>
  gt() |>
  tab_header(
    title = "Top 100 Snowiest Days",
    subtitle = "NYC Central Park (1869-2022)"
  )
```

That will give you a clean two-column table showing just the date and snowfall amount!

Citations

Anthropic. Claude Sonnet 4.5 [Large Language Model]. https://claude.ai. Accessed February 1st, 2026