The Not-So-Secret Diary of a TV Show Data Wrangler

Let me tell you about the time I wrestled with a CSV file full of TV shows and lived to tell the tale. Picture this: me, a cup of coffee (okay, several cups), and enough TV show data to make Netflix jealous.

First battle: parsing CSV files. Oh, how innocent I was, thinking a simple split(",") would do the trick. Then I met my nemesis - nested quotes. Here's what my sanity-saving revelation looked like:

```
if (c == '"') {
   if (inQuotes && i + 1 < line.length() && line.charAt(i + 1) == '"') {
      currentField.append('"');
      i++; // Skip next quote - my savior!
   } else {
      inQuotes = !inQuotes;
   }
}
```

Thank goodness for those helpful Indian developers who explained that double quotes within double quotes are like Russian nesting dolls - you need to handle each layer carefully or everything falls apart.

The first three tasks were deceptively simple. Sort by rating? Just a casual:

```
tvShowsList.sort(Comparator.comparingDouble(TVshows::getVoteAverage).reversed());
```

Look at how innocent that looks! It's like the "Hello World" of sorting - suspiciously straightforward.

Oh boy, Task 4... Finding shows between 2010-2019 reminded me of my life back in 2004 (I was born then and spent years just watching TV – I STILL DO). Here's the logic that made me question everything:

```
String startDate = "2010-01-01";
String endDate = "2019-12-31";
for (TVshows i : tvShowsList) {
   String firstAirDate = i.getFirstAirDate();
   String lastAirDate = i.getLastAirDate();
```

```
    if ((firstAirDate.compareTo(startDate) >= 0) && (lastAirDate.compareTo(endDate) <= 0)) {
        showsBetweenDates.add(i);
    }
}
```

Look at that innocent-looking `compareTo`! It sits there pretending it didn't cause me to have an existential crisis at 2 AM. "Is 2010 greater than 2019? Wait, am I greater than 2019? What even is time?" These are the questions that haunt a developer's dreams.

By this point, I was speaking fluent HashMap. Finding shows in production was like playing "Where's Waldo?" but Waldo is wearing a bright neon sign saying "IN_PRODUCTION":

```
List<TVshows> inProductionShows = new ArrayList<>();
for (TVshows i : tvShowsList) {
    if (i.getInProduction()) {
        inProductionShows.add(i);
    }
}
```

Simple? Yes. Did I still manage to write `getInProdcution()` first and spend 15 minutes debugging? Also yes.

Then came the single-word title challenge. Who knew this regex would become my new best friend:

```
if (i.getName().split("\\s+").length == 1) {
    singleWordTitleShows.add(i);
}
```

It's like having a bouncer at a club who only lets in shows with one-word names. "Sorry 'Game of Thrones', you're too wordy for this party. 'Friends'? Right this way!"

The final boss: grouping by genre. My HashMap game reached its final form:

```
Map<String, List<TVshows>> showsByGenre = new HashMap<>();
for (TVshows i : tvShowsList) {
    String[] genres = i.getGenres().split(",");
    for (String j : genres) {
        j = j.trim();
        showsByGenre.putIfAbsent(j, new ArrayList<>());
```

```
        showsByGenre.get(j).add(i);
    }
}
```

This code is like a TV show sorting hat - "Action! You go there! Drama! Over there! Reality TV... well, we have to put you somewhere."

Throughout all tasks, my trusty *BufferedWriter* was there, catching all my comma-separated tears:

```
try (BufferedWriter buff = new BufferedWriter(new FileWriter(filename))) {
    buff.write("Header,That,I,Triple,Checked\n");
    // Insert countless lines of careful string concatenation
} catch (IOException e) {
    e.printStackTrace(); // My way of saying "It's not me, it's you" to the file system
}
```

In the end, everything worked. The code runs, the files are generated, and I only slightly questioned my career choices seven times during the process. My coffee consumption reached new heights, and I now see CSV files in my dreams.

Special shoutout to:

- The countless System.out.println() statements that helped debug this beast
- The IDE's auto-complete feature that saved my sanity
- Stack Overflow, my midnight companion especially you, Rajjish.
- The energy drinks my sister got me, the real MVP

P.S. To all future maintainers of this code: The comments made sense at 3 AM, I swear. And yes, that variable name 'i' could be more descriptive, but by that point, I was running on fumes and spite.