Here are the list of common mistakes made by a Data scientists:

1. **Do not share the data referenced in the code:**
   1. If we share only code and not the original data then anyone else won't be able to reproduce your results.
   2. To share the data, either upload it to S3/web/google drive or use d63pipe
   3. Currently d63pipe is giving Application error while registering.
2. **Hardcode inaccessible paths:**
   1. if you hardcode paths others don’t have access to, they can’t run your code
   2. Use relative paths, global path configs or d6tpipe to make your data easily accessible
3. **Mix data with code:**
   1. Along with data, we also create images, reports and other junks in the code
   2. If we save all these things together, it will be a mess
   3. Organize your directory into categories, like data, reports, code etc.
   4. See [Cookiecutter Data Science](https://drivendata.github.io/cookiecutter-data-science/#directory-structure)
4. **Git commit data with source code:**
   1. Most people now use git
   2. However, sometimes it may not be possible to upload the data to git as git is designed only for small sized files.
   3. To share the large data content, use d6tpipe, Git large file storage
5. **Write functions instead of DAGs: Directed Acyclic Graph**
   1. data science code is mostly organized as a series of functions that are run linearly. However this may cause several problems
   2. Instead of linearly chaining functions, data science code is better written as a set of tasks with dependencies between them. Use [d6tflow](https://github.com/d6t/d6tflow) or [airflow](https://airflow.apache.org/).
6. **Write for loops:**
   1. We use for loops many times. But they are slow and excessively wordy
   2. Numpy, scipy and pandas have vectorized functions for most things that you think might require for loops
   3. For example: df.mean, np.mean, df.apply, df.applymap, df.map etc.
7. **Don't write unit tests**
   1. Use assert statements to check data quality. Pandas has equality tests, d6tstack has checks for data ingestion and d6tjoin for data joins.
8. **Don’t document code:**
   1. We often forget to document the steps we did while writing the code
   2. So after a few days, we may not remember which step we did and why we did so.
   3. That is why, documenting the code is very important
9. **Save the data as csv or pickle:**
   1. We use csv and pickle to save our data. CSVs don't include schemas so everyone has to parse the numbers and dates. Pickle only works with python and is not compressed.
   2. Use parquet, d6tflow automatically saves the data output of tasks as parquet so you don't have to deal with it
10. **Use jupyter notebook:**
    1. Use of jupyter promotes some bad software engineering habits as below:
       1. You are tempted to dump all files in a single directory
       2. You write code that runs top-bottom instead of DAG
       3. Difficult to debug
       4. Code and output gets mixed in one file
       5. They don’t version control well
    2. So use pycharm or spyder