

Project Instructions

- Task 2: Instructions
- Clean the dataset.
 - Create a list named `chars_to_remove` that contains the following characters: `+` and `$`.
 - Create a list named `cols_to_clean` that contains the following column names: `Installs` and `Price`.
 - For each column in `cols_to_clean` in the `apps` DataFrame, replace each character in `chars_to_remove` with the empty string `''`.
Note: Make sure to use an empty string `''` and not a space character .
 - Finally, print a summary of the `apps` dataframe using the `info()` function. Observe the output.
Note: Notice that `Installs` and `Price` are still of type `object` and not `int` or `float` as we would have expected after removal of the special characters. We will solve this issue in the next task.
- Helpful links:
 - [pandas apply\(\) documentation](#)
 - [Python replace\(\) documentation](#)
 - [pandas info\(\) documentation](#)

Take Hint

Previous Task Next Task

Project: The Android App Market on Google Play

Jupyter notebook (autosaved)

File Edit View Insert Cell Kernel Help

2. Data cleaning

Data cleaning is one of the most essential subtasks any data science project. Although it can be a very tedious process, it's worth should never be undermined. By looking at a random sample of the dataset rows (from the above task), we observe that some entries in the columns like `Installs` and `Price` have a few special characters (`+`, `$`, `/`, `code`) due to the way the numbers have been represented. This prevents the columns from being purely numeric, making it difficult to use them in subsequent future mathematical calculations. Ideally, as their names suggest, we would want these columns to contain only digits from `[0-9]`.

Hence, we now proceed to clean our data. Specifically, the special characters `<code>`, `</code>` and `<code>+</code>` present in `<code>Installs</code>` column and `<code>$</code>` present in `Price` column need to be removed.

It is also always a good practice to print a summary of your dataframe after completing data cleaning. We will use the `info()` method to achieve this.

```
In [ ]: # list of characters to remove
chars_to_remove = ['+', '$']
# list of column names to clean
cols_to_clean = ['Installs', 'Price']

# loop for each column in cols_to_clean
for col in cols_to_clean:
    # loop for each char in chars_to_remove
    for char in chars_to_remove:
        # Replace the character with an empty string
        apps[col] = apps[col].apply(lambda x: x.replace(char, ''))

# Print a summary of the apps dataframe
print(apps.info())
```

3. Correcting data types

From the previous task we noticed that `Installs` and `Price` were categorized as `object` data type (and not `int` or `float`) as we would like. This is because these two columns originally had mixed input types: digits and special characters. To know more about Pandas data types, read [this](#).

The four features that we will be working with most frequently henceforth are `Installs`, `Size`, `Rating` and `Price`. While `Size` and `Rating` are both `float` (i.e. purely numerical data types), we still need to work on `Installs` and `Price` to make them numeric.

Check Project

Google Play Store Analysis - Task 2

Task 2 Instructions

1. Clean the dataset:

- Create a list named `chars_to_remove` that contains the following characters: `+` and `$`.
- Create a list named `cols_to_clean` that contains the following column names: `Installs` and `Price`.
- For each column in `cols_to_clean` in the `apps` DataFrame, replace each character in `chars_to_remove` with the empty string `''`.
- Finally, print a summary of the `apps` DataFrame using the `info()` function.

Note: Make sure to use an empty string `''` and not a space character . Observe the output to ensure the columns are cleaned.

Correct Code Implementation

```
# Step 1: Create a list of characters to remove
chars_to_remove = ['+', '$']
```

```
# Step 2: Create a list of column names to clean
cols_to_clean = ['Installs', 'Price']
```

```
# Step 3: Loop over each column in 'cols_to_clean' and remove unwanted
characters
for col in cols_to_clean:
    for char in chars_to_remove:
        apps[col] = apps[col].apply(lambda x: x.replace(char, '') if isinstance(x,
str) else x)

# Step 4: Print a summary of the apps dataframe
print(apps.info())
```

Explanation of the Code

1. **chars_to_remove**: Specifies the special characters (`+` and `\$`) to be removed from the specified columns.
2. **cols_to_clean**: Indicates the columns (`Installs` and `Price`) in which to perform the cleaning.
3. **Nested loop**:
 - Outer loop iterates over the columns in `cols_to_clean`.
 - Inner loop iterates over the characters in `chars_to_remove`, replacing each occurrence in the column with an empty string `''`.
4. **apply()**: Applies the `replace()` function to each element of the column, ensuring strings are cleaned.
5. **apps.info()**: Prints a summary of the DataFrame to verify that the cleaning operation has been performed.