

Pandas Data Wrangling Project – Phase 2

Introduction

This is Phase 2 of the Pandas Data Wrangling Project. Because this phase builds on the work completed in Phase 1, it assumes that the dataset you are working with is in the state it would have been at the end of Phase 1. You have a couple of options for maintaining this continuity:

1. **Continue in the same notebook:** Build out Phase 2 in the same notebook you used for Phase 1, in which case you would simply continue to modify the core DataFrame.
 2. **Start a new notebook:** Export the final version of your DataFrame from Phase 1 to a .csv file, then import that file into a new notebook to begin Phase 2.
-

Task 1: Assess Missing Values

Start by assessing how many missing values are in each column of the DataFrame.

HINT, part 1: if you call the `isna` method on the DataFrame as a whole - rather than an individual Series - it will return a new DataFrame with the same dimensions as the original, but where every value is either the logical True if the original value in that position is missing, and False otherwise.

HINT, part 2: if you perform mathematical operations on Boolean True/False values, True values are treated as "1"s and False values are treated as "0s". So what happens if you apply the `sum` method to a Series (or DataFrame!) containing such Boolean values?

Task 2: Interpolate Missing Values

Bruno is perturbed by the missing values in the "total_charges" column, and wants us to *interpolate* those missing values (i.e., fill in the blanks via an estimate) based on the following expression: **monthly_charges * tenure**.

Complete the following steps:

1. Calculate estimated total charges
 2. Replace missing estimated total charges values with estimated values
-

Task 3: Confirm Remaining Missing Values

Next, confirm whether there are still missing values in the "total_charges" column. If so, try to identify why.

HINT: you can accomplish this in ALMOST the same way that you tested for missing values across the entire DataFrame; you're pretty much applying the same operation, but to a single Series this time.

Complete the following steps:

1. Check for missing values in "total_charges" column
 2. Filter for rows with missing values in the "total_charges" column
-

Task 4: Remove Unrecoverable Records

Unfortunately, it turns out there are *some* records in the dataset with missing values for both "total_charges" AND "monthly_charges". Since we can't even extrapolate a value for total charges in this scenario, Bruno wants us to remove these records from the dataset altogether.

Task 5: Remove True Duplicates

Bruno is also concerned there may be duplicate records in the dataset due to a system migration issue. Identify whether any "true" duplicates (i.e., duplicated across all fields) in the DataFrame, and remove them if so.

Complete the following steps:

1. Identify duplicate records
 2. Remove duplicate records
-

Task 6: Identify Functional Duplicates

It's also a good idea to look for potential "functional duplicates" - records that have identical values across all columns EXCEPT the unique identifier "customer_id". These might represent the same customer with different IDs.

HINT: instead of manually creating a list of all the DataFrame column names you want to target (not exactly practical if the DataFrame has a lot of columns), you can call the drop method on the DataFrame's columns property, passing in the name(s) of whatever column or columns you DON'T want to include. For example:

```
your_df.columns.drop('column_you_dont_care_about')
```

This will return a list-like object (technically an Index), containing just the columns you're interested in, that you can then use interchangeably with a list in many Pandas methods.

Task 7: Remove Functional Duplicates

Now remove any functional duplicates you find, making sure to reset the index of the DataFrame. Verify that overall record count of the DataFrame decreases by the expected amount.

Task 8: Standardize Inconsistent Values

Bruno has noticed what appear to be inconsistent values in the "gender" and "internet_service" columns of the dataset. Specifically:

- The values in the "gender" column are not capitalized consistently
- In the "internet_service" column, there appear to be alternate multiple instances of different values that mean the same thing; for example, "Fiber optic"/"Fiber-optic" and "DSL"/"Digital Subscriber Line".

Now *your* job is to clean up these inconsistencies using the Pandas methods you've learned!

Complete the following steps:

1. Standardize the capitalization of the values in the "gender" column to use proper case
 2. Replace the values "Fiber-optic" and "Digital Subscriber Line" in the "internet_service" column with their equivalents
-

Task 9: Extract Customer Sequence Numbers

Bruno has also noticed that the *numeric* portions of the customer IDs appear to be both unique *and* sequential, and as such may indicate the order in which customers signed up with Keiko corp. Meanwhile, the non-numeric characters in those same IDs don't appear to have any meaning.

Given this, Bruno wants us to create a derived "customer_seq" column, consisting of only the **numeric** portion "customer_id" field, converted to actual numeric *values* that can (for example) be meaningfully sorted. This may be a bit tricky however, since the numeric portions of the IDs can be found at both the beginning and end of those IDs. (*HINT: try replacing those non-numeric characters via a regex, which you should be able to research/find fairly easily.*)

Complete the following steps:

1. Remove non-numeric characters from "customer_id"
 2. Convert "customer_seq" Series to a numeric data type
-

Task 10: Sort by Customer Sequence

Next up, sort the dataset by those customer ID numbers you just extracted.

Task 11: Answer Bruno's Questions

Now that we have the dataset cleaned up nicely, Bruno has a couple of specific questions about the data:

1. What is the churn rate of customers with tenure less than 6 months?
 2. What percentage of customers pay automatically? (*HINT: try testing whether the values in the "payment_method" column contain the word "automatic."*)
-

Task 12: Filter the Dataset

Finally, after meeting with other executives at Keiko, Bruno has decided to narrow the scope of our analysis to customers who:

3. Have phone service (Keiko's executive team considers this a "core" service)
4. AND have a tenure of at least 1 month, since it's difficult to infer any meaningful information from customers who have been with the company for such a short period of time.

As such, Bruno would like us to filter the primary dataset we're analyzing to only include customers meeting those criteria.