result

January 4, 2021

```
In [1]: import pandas as pd
        from datetime import timedelta, datetime
        import os
        import math
        from IPython.core.interactiveshell import InteractiveShell
        InteractiveShell.ast_node_interactivity = "all"

In [2]: # Set this variable True if running locally, False if running on Losant
        LOCAL = False

        # load data sources
        if LOCAL:
            ALL_DATA_PATH1 = './Downloads/all_data1.csv' # device data
            ALL_DATA_PATH2 = './Downloads/all_data2.csv' # device data
            ALL_DATA_PATH3 = './Downloads/all_data3.csv' # device data
            DEVICE_DATES_PATH = "./Downloads/Device_IO.csv"
            SHIPPED_PATH = "./Downloads/0_rolls_shipped.csv"
            USED_PATH = "./Downloads/0_rolls_used.csv"
        else:
            ALL_DATA_PATH1 = os.path.join(os.environ['INPUT_DIR'],'all_data1.csv') # device data
            ALL_DATA_PATH2 = os.path.join(os.environ['INPUT_DIR'],'all_data2.csv') # device data
            ALL_DATA_PATH3 = os.path.join(os.environ['INPUT_DIR'],'all_data3.csv')
            DEVICE_DATES_PATH = os.path.join(os.environ['INPUT_DIR'],"Device_IO.csv")
            SHIPPED_PATH = os.path.join(os.environ['INPUT_DIR'], "0_rolls_shipped.csv")
            USED_PATH = os.path.join(os.environ['INPUT_DIR'], "0_rolls_used.csv")

In [3]: all_data1 = pd.read_csv(ALL_DATA_PATH1)
        all_data2 = pd.read_csv(ALL_DATA_PATH2)
        all_data3 = pd.read_csv(ALL_DATA_PATH3)
        device_dates = pd.read_csv(DEVICE_DATES_PATH)

In [4]: all_data = pd.concat([all_data1, all_data2, all_data3])
        all_data = all_data[all_data["rollId"] != 0]

In [5]: long_to_short = pd.Series(device_dates["Device_ID"].values,
        index=device_dates["Device_ID_long"].values).to_dict()

In [6]: all_data["short_name"] = all_data["ID"].map(long_to_short)

In [7]: try:
            device_dates["Date_in"] = device_dates["Date_in"].apply(
                lambda x: datetime.strptime(str(x), '%Y-%m-%d'))
            device_dates["Date_out"] = device_dates["Date_out"].apply(
                lambda x: datetime.strptime(str(x), '%Y-%m-%d'))
        except:
            try:
                device_dates["Date_in"] = device_dates["Date_in"].apply(
                    lambda x: datetime.strptime(str(x)[:10], '%Y-%m-%d'))
                device_dates["Date_out"] = device_dates["Date_out"].apply(
                    lambda x: datetime.strptime(str(x)[:10], '%Y-%m-%d'))
            except:
                device_dates["Date_in"] = device_dates["Date_in"].apply(lambda x:
        datetime.strptime(str(x), '%m/%d/%y'))
```

```
            device_dates["Date_out"] = device_dates["Date_out"].apply(lambda x:
        datetime.strptime(str(x), '%m/%d/%y'))
        all_data["Timestamp"] = all_data["Timestamp"].astype(
            "datetime64[ms]") - timedelta(hours=5)
```

```
In [8]: for i in range(len(device_dates)):
            row = device_dates.iloc[i]
            all_data.loc[all_data["Timestamp"].between(row["Date_in"], row["Date_out"]) &
        (all_data["short_name"] == row["Device_ID"]), "telemetry_id"] = row["telem_ID"]
```

```
In [9]: rolls_used = all_data.groupby(by="telemetry_id")["rollId"].nunique()
```

```
In [10]: df_shipped = pd.read_csv(SHIPPED_PATH)[
             ["telemetry_id", "devices_placed", "rolls_shipped"]]
         df_manual_used = pd.read_csv(USED_PATH)[
             ["telemetry_id", "Manual_Rolls_Used"]]
```

```
In [11]: df_shipped.set_index("telemetry_id", inplace=True)
         df_manual_used.set_index("telemetry_id", inplace=True)
```

```
In [12]: test = pd.concat([df_shipped, df_manual_used, rolls_used], axis=1)
```

```
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:1: FutureWarning: Sorting
because non-concatenation axis is not aligned. A future version
of pandas will change to not sort by default.

To accept the future behavior, pass 'sort=False'.

To retain the current behavior and silence the warning, pass 'sort=True'.

  """Entry point for launching an IPython kernel.
```

```
In [13]: test["rolls_used_adjusted"] = test["rollId"] + test["Manual_Rolls_Used"]
```

```
In [14]: test["rolls_remaining"] = test["rolls_shipped"] - test["rolls_used_adjusted"]
```

```
In [15]: test.reset_index(inplace=True)
```

```
In [16]: test = test[["devices_placed", "rollId", "Manual_Rolls_Used",
                      "rolls_used_adjusted", "rolls_shipped", "rolls_remaining", "index"]]
```

```
In [17]: test.columns = ["devices_placed", "rolls_used", "Manual_Rolls_Used",
                      "rolls_used_adjusted", "rolls_shipped", "rolls_remaining", "telemetry_id"]
```

```
In [18]: test.to_csv(os.path.join(os.environ['OUTPUT_DIR'], '0_rolls_used.csv'), index=False)
```

```
In [ ]:
```