

```

"""
    Day 14 Activity: Full Cleaning Pipeline
    :Tasks
    Build clean_data() that orchestrates type, missing, outliers, (1
    strings/dates
    Add basic validation checks (2
    Run end-to-end and inspect (3
    """

import pandas as pd
import numpy as np
(df=pd.read_csv("day14_users_raw.csv"
    (print(df.columns
    (TODO: Implement clean_types(df #
    : (def clean(m:pd.DataFrame
    ("m["age"]=pd.to_numeric(m["age"],errors="coerce
    ("m["income"]=pd.to_numeric(m["income"],errors="coerce
    return m
    (TODO: Implement clean_missing(df #
    : (def clean_missing(df
    ()out = df.copy
    ((out["age"] =out["age"].fillna(out["age"].median
    return out

    (TODO: Implement handle_outliers(df #

def winsorize_DataFrame(df: pd.DataFrame, lower_q=0.01,
    :upper_q=0.99) -> pd.DataFrame
    ((df["age"] = df["age"].clip(upper=df["age"].quantile(0.99
    return df

df["income_cap_1_99"] = winsorize_DataFrame(df["income"], 0.01,
    (0.99
    (TODO: Implement clean_strings_and_dates(df #
    : (def clean_strings_and_dates(df12:pd.DataFrame
df12["signup_time"]=pd.to_datetime(df12["signup_time"],errors="coe
    ("rce
    df12["signup_dt_local"] =
    ,df12["signup_time"].dt.tz_localize("US/Eastern", ambiguous="NaT
    ("nonexistent="NaT
    df12["signup_dt_utc"] =
    ("df12["signup_dt_local"].dt.tz_convert("UTC
    return df12
    (TODO: Implement validate_cleaned(df #

```

```

        : (def validate_cleaned(df:pd.DataFrame
            "assert df["age"].min() >= 0, "Negative ages found
            "assert df["income"].notna().all(), "Income still has NaN
            TODO: Implement clean_data(df) that calls the above in order #
            : (def clean_data(df: pd.DataFrame
                ()df = df.copy
                (df = clean(df
                (df = clean_missing(df
                (df = winsorize_DataFrame(df
            (df = clean_strings_and_dates(df
                (validate_cleaned(df
                    return df

(clean_data(df

```