data_ingestion

December 8, 2025

1 Data Ingestion

This notebook pulls the raw tables used throughout the project directly from the NASA Exoplanet Archive and stores the results under `data/raw/` so downstream notebooks (such as `eda.ipynb`) can rely on versioned CSVs instead of re-querying the API every run.

[10]:
```
from pathlib import Path
from io import StringIO

import pandas as pd
import requests
```

[11]:
```
PROJECT_ROOT = Path.cwd()
if PROJECT_ROOT.name == "notebooks":
    PROJECT_ROOT = PROJECT_ROOT.parent

DATA_RAW = PROJECT_ROOT / "data" / "raw"
DATA_RAW.mkdir(parents=True, exist_ok=True)
```

[12]:
```
TAP_URL = "https://exoplanetarchive.ipac.caltech.edu/TAP/sync"
```

[13]:
```
exoplanets_path = DATA_RAW / "exoplanets.csv"
pscomppars_path = DATA_RAW / "exoplanets_pscomppars.csv"
solar_system_path = DATA_RAW / "solar_system_planets.csv"
```

[14]:
```
SOLAR_SYSTEM_ROWS = [
    {"pl_name": "Mercury", "hostname": "Sun", "pl_orbper": 87.969, "pl_orbsmax":
 ↪ 0.387, "pl_rade": 0.3829,
     "pl_masse": 0.0553, "pl_eqt": 440.0, "pl_insol": 6.68, "pl_orbeccen": 0.
 ↪2056,
     "st_teff": 5778.0, "st_rad": 1.0, "st_mass": 1.0},
    {"pl_name": "Venus", "hostname": "Sun", "pl_orbper": 224.701, "pl_orbsmax":␣
 ↪0.723, "pl_rade": 0.9499,
     "pl_masse": 0.8150, "pl_eqt": 737.0, "pl_insol": 1.91, "pl_orbeccen": 0.
 ↪0067,
     "st_teff": 5778.0, "st_rad": 1.0, "st_mass": 1.0},
    {"pl_name": "Earth", "hostname": "Sun", "pl_orbper": 365.256, "pl_orbsmax":␣
 ↪1.000, "pl_rade": 1.0,
```

```
    "pl_masse": 1.0, "pl_eqt": 255.0, "pl_insol": 1.0, "pl_orbeccen": 0.0167,
    "st_teff": 5778.0, "st_rad": 1.0, "st_mass": 1.0},
   {"pl_name": "Mars", "hostname": "Sun", "pl_orbper": 686.980, "pl_orbsmax":␣
↪1.524, "pl_rade": 0.5320,
    "pl_masse": 0.1074, "pl_eqt": 210.0, "pl_insol": 0.43, "pl_orbeccen": 0.
↪0934,
    "st_teff": 5778.0, "st_rad": 1.0, "st_mass": 1.0},
   {"pl_name": "Jupiter", "hostname": "Sun", "pl_orbper": 4332.589,␣
↪"pl_orbsmax": 5.203, "pl_rade": 11.209,
    "pl_masse": 317.8, "pl_eqt": 110.0, "pl_insol": 0.04, "pl_orbeccen": 0.
↪0489,
    "st_teff": 5778.0, "st_rad": 1.0, "st_mass": 1.0},
   {"pl_name": "Saturn", "hostname": "Sun", "pl_orbper": 10759.22,␣
↪"pl_orbsmax": 9.537, "pl_rade": 9.449,
    "pl_masse": 95.16, "pl_eqt": 82.0, "pl_insol": 0.01, "pl_orbeccen": 0.0565,
    "st_teff": 5778.0, "st_rad": 1.0, "st_mass": 1.0},
   {"pl_name": "Uranus", "hostname": "Sun", "pl_orbper": 30685.4, "pl_orbsmax":
↪ 19.191, "pl_rade": 4.007,
    "pl_masse": 14.54, "pl_eqt": 76.0, "pl_insol": 0.0037, "pl_orbeccen": 0.
↪0463,
    "st_teff": 5778.0, "st_rad": 1.0, "st_mass": 1.0},
   {"pl_name": "Neptune", "hostname": "Sun", "pl_orbper": 60189.0,␣
↪"pl_orbsmax": 30.07, "pl_rade": 3.883,
    "pl_masse": 17.15, "pl_eqt": 72.0, "pl_insol": 0.0015, "pl_orbeccen": 0.
↪0097,
    "st_teff": 5778.0, "st_rad": 1.0, "st_mass": 1.0},
]
```

The Solar System canonical values used below are sourced from NASA's Planetary Fact Sheet (https://nssdc.gsfc.nasa.gov/planetary/factsheet/planetfact_table.html). NASA publishes these parameters under the U.S. Government's Public Domain policy (Title 17, Section 105), so they may be freely reused and redistributed with attribution.

[15]:
```python
def run_tap_query(adql_query: str) -> pd.DataFrame:
    """Execute an ADQL query against the NASA Exoplanet Archive TAP endpoint."""
    params = {
        "request": "doQuery",
        "lang": "ADQL",
        "format": "csv",
        "query": adql_query,
    }
    response = requests.post(TAP_URL, data=params, timeout=60)
    response.raise_for_status()
    return pd.read_csv(StringIO(response.text))
```

[16]:
```python
if pscomppars_path.exists():
    pscomppars_df = pd.read_csv(pscomppars_path)
```

```
    print(f"Loaded cached pscomppars table: {pscomppars_df.shape}")
else:
    pscomppars_df = run_tap_query("SELECT * FROM pscomppars")
    pscomppars_df.to_csv(pscomppars_path, index=False)
    print(f"Downloaded pscomppars table: {pscomppars_df.shape}")

pscomppars_df.head()
```

```
Loaded cached pscomppars table: (6060, 683)
```

[16]:

	objectid	pl_name	pl_letter	hostid	hostname	hd_name	hip_name	\
0	3.2390	Kepler-1167 b	b	2.136990	Kepler-1167	NaN	NaN	
1	3.1444	Kepler-1740 b	b	2.433343	Kepler-1740	NaN	NaN	
2	3.4135	Kepler-1581 b	b	2.442550	Kepler-1581	NaN	NaN	
3	3.6590	Kepler-644 b	b	2.512738	Kepler-644	NaN	NaN	
4	3.1575	Kepler-1752 b	b	2.507010	Kepler-1752	NaN	NaN	

	tic_id	disc_pubdate	disc_year	…	cb_flag	pl_angsep	pl_angseperr1	\
0	TIC 273875149	2016-05	2016.0	…	0.0	0.0213	NaN	
1	TIC 138479461	2022-02	2021.0	…	0.0	0.0734	NaN	
2	TIC 121215710	2016-05	2016.0	…	0.0	0.1390	NaN	
3	TIC 271669616	2016-05	2016.0	…	0.0	0.0352	NaN	
4	TIC 417655835	2022-02	2021.0	…	0.0	0.2800	NaN	

	pl_angseperr2	pl_angseplim	pl_angsepformat	pl_angsepstr	pl_angsepsymerr	\
0	NaN	0.0	NaN	0.0213	NaN	
1	NaN	0.0	NaN	0.0734	NaN	
2	NaN	0.0	NaN	0.1390	NaN	
3	NaN	0.0	NaN	0.0352	NaN	
4	NaN	0.0	NaN	0.2800	NaN	

	pl_angsep_reflink	pl_ndispec
0	<a refstr=CALCULATED_VALUE href=/docs/pscp_cal…	0.0
1	<a refstr=CALCULATED_VALUE href=/docs/pscp_cal…	0.0
2	<a refstr=CALCULATED_VALUE href=/docs/pscp_cal…	0.0
3	<a refstr=CALCULATED_VALUE href=/docs/pscp_cal…	0.0
4	<a refstr=CALCULATED_VALUE href=/docs/pscp_cal…	0.0

[5 rows x 683 columns]

[17]:
```
if solar_system_path.exists():
    solar_df = pd.read_csv(solar_system_path)
    print(f"Loaded cached solar system table: {solar_df.shape}")
else:
    solar_df = pd.DataFrame(SOLAR_SYSTEM_ROWS)
    solar_df.to_csv(solar_system_path, index=False)
    print(f"Created solar system table from NASA fact sheet values: {solar_df.
 ↪shape}")
```

```
solar_df.head()
```

```
Created solar system table from NASA fact sheet values: (8, 12)
```

[17]:

	pl_name	hostname	pl_orbper	pl_orbsmax	pl_rade	pl_masse	pl_eqt	\
0	Mercury	Sun	87.969	0.387	0.3829	0.0553	440.0	
1	Venus	Sun	224.701	0.723	0.9499	0.8150	737.0	
2	Earth	Sun	365.256	1.000	1.0000	1.0000	255.0	
3	Mars	Sun	686.980	1.524	0.5320	0.1074	210.0	
4	Jupiter	Sun	4332.589	5.203	11.2090	317.8000	110.0	

	pl_insol	pl_orbeccen	st_teff	st_rad	st_mass
0	6.68	0.2056	5778.0	1.0	1.0
1	1.91	0.0067	5778.0	1.0	1.0
2	1.00	0.0167	5778.0	1.0	1.0
3	0.43	0.0934	5778.0	1.0	1.0
4	0.04	0.0489	5778.0	1.0	1.0

[18]:
```
solar_aligned = solar_df.reindex(columns=pscomppars_df.columns, fill_value=pd.
 ↪NA)
pscomppars_df = (
    pd.concat([pscomppars_df, solar_aligned], ignore_index=True)
    .drop_duplicates(subset="pl_name", keep="first")
    .reset_index(drop=True)
)
pscomppars_df.to_csv(pscomppars_path, index=False)
print(
    "Combined table saved to",
    pscomppars_path,
    pscomppars_df.shape,
)
pscomppars_df.tail(len(solar_df))
```

```
/var/folders/v9/pj6tyjzj4ssfd95k4nfrdszr0000gn/T/ipykernel_81381/4137367739.py:3
: FutureWarning: The behavior of DataFrame concatenation with empty or all-NA
entries is deprecated. In a future version, this will no longer exclude empty or
all-NA columns when determining the result dtypes. To retain the old behavior,
exclude the relevant entries before the concat operation.
  pd.concat([pscomppars_df, solar_aligned], ignore_index=True)

Combined table saved to /Users/aaditya.chopra/Desktop/Aaditya/Udub/Courses/DATA
512 - Human Centric Data Science/A4/data/raw/exoplanets_pscomppars.csv (6060,
683)
```

[18]:

	objectid	pl_name	pl_letter	hostid	hostname	hd_name	hip_name	tic_id	\
6052	NaN	Mercury	NaN	NaN	Sun	NaN	NaN	NaN	
6053	NaN	Venus	NaN	NaN	Sun	NaN	NaN	NaN	
6054	NaN	Earth	NaN	NaN	Sun	NaN	NaN	NaN	

```
6055       NaN     Mars       NaN     NaN      Sun     NaN      NaN    NaN
6056       NaN  Jupiter       NaN     NaN      Sun     NaN      NaN    NaN
6057       NaN   Saturn       NaN     NaN      Sun     NaN      NaN    NaN
6058       NaN   Uranus       NaN     NaN      Sun     NaN      NaN    NaN
6059       NaN  Neptune       NaN     NaN      Sun     NaN      NaN    NaN

     disc_pubdate  disc_year  … cb_flag pl_angsep pl_angseperr1  \
6052          NaN        NaN  …     NaN       NaN           NaN
6053          NaN        NaN  …     NaN       NaN           NaN
6054          NaN        NaN  …     NaN       NaN           NaN
6055          NaN        NaN  …     NaN       NaN           NaN
6056          NaN        NaN  …     NaN       NaN           NaN
6057          NaN        NaN  …     NaN       NaN           NaN
6058          NaN        NaN  …     NaN       NaN           NaN
6059          NaN        NaN  …     NaN       NaN           NaN

     pl_angseperr2 pl_angseplim pl_angsepformat pl_angsepstr  pl_angsepsymerr  \
6052           NaN          NaN             NaN          NaN              NaN
6053           NaN          NaN             NaN          NaN              NaN
6054           NaN          NaN             NaN          NaN              NaN
6055           NaN          NaN             NaN          NaN              NaN
6056           NaN          NaN             NaN          NaN              NaN
6057           NaN          NaN             NaN          NaN              NaN
6058           NaN          NaN             NaN          NaN              NaN
6059           NaN          NaN             NaN          NaN              NaN

      pl_angsep_reflink  pl_ndispec
6052                NaN         NaN
6053                NaN         NaN
6054                NaN         NaN
6055                NaN         NaN
6056                NaN         NaN
6057                NaN         NaN
6058                NaN         NaN
6059                NaN         NaN

[8 rows x 683 columns]
```