UK Government Departmental Spending Analysis

Project Overview

This project analyzes real public spending data from the **Department for Science, Innovation and Technology (DSIT)** — a UK government department responsible for science, research, and digital technology funding.

The dataset covers **monthly transactions over £25,000** from **2024 to 2025**, giving us a transparent view of how taxpayer money is being spent.

What has been done

started by collecting **18 monthly CSV files** spread across two years and combined them into one clean, unified dataset. The raw data was messy it had inconsistent date formats, currency symbols in numbers, missing values, and duplicate records. We cleaned all of that systematically before doing any analysis.

Once the data was clean, I asked real business questions:

- Where is the money going each month?
- Who are the biggest suppliers receiving government payments?
- Which expense categories consume the most budget?
- Are there unusually large transactions that need attention?
- How did spending change between 2024 and 2025?

What has has been Found

The analysis revealed clear patterns in government spending — which suppliers dominate the budget, which expense categories are growing, and where the biggest individual payments are being made. These insights help stakeholders, auditors, and the public understand how public money is being managed.

In [1]:
```
import pandas as pd
from pathlib import Path

# Define the path
# I'm in 'processed data', so go up one level (..) to reach 'data',
# then into 'raw data'
```

```
raw_data_path = Path('..') / 'data' / 'raw data'

# Get CSV files from both folders
csv_files_2024 = list((raw_data_path / '2024').glob('*.csv'))
csv_files_2025 = list((raw_data_path / '2025').glob('*.csv'))

# Combine the lists
all_files = csv_files_2024 + csv_files_2025

# CHECK HOW MANY FILES WERE FOUND
print(f"Files in 2024: {len(csv_files_2024)}")
print(f"Files in 2025: {len(csv_files_2025)}")
print(f"Total files: {len(all_files)}")

if len(all_files) == 0:
    print("No files found! Current working directory:")
    print(Path.cwd())
    print(f"Path exists: {raw_data_path.exists()}")
else:
    # Combine into DataFrame - simple one-liner with encoding fix
    df = pd.concat(
        [pd.read_csv(f, encoding='windows-1252') for f in all_files],
        ignore_index=True
    )
```

```
Files in 2024: 12
Files in 2025: 6
Total files: 18
```

Setting dates as an index and sorting them.

In [2]:
```
# df['Date of Payment'] = pd.to_datetime(df['Date of Payment'], format='%d-%m-%Y')

newdf = df.sort_values('Date of Payment').reset_index(drop= True).copy()
newdf
```

Out[2]:

	Date of Payment	Expense Type	Expense Area	Supplier	Transaction Number	Amount	Descri
0	01/02/2024	Other It Consultancy	Dsit - Science, Innovation And Growth - Dsit -...	Atkinsrealis Uk Ltd	567929	134394.66	Na Undergr Reg
1	01/02/2024	Grant-in-aid To Arms Length Bodies	Dsit - Science, Innovation And Growth - Dsit -...	Ukri - Engineering And Physical Sciences Resea...	566724	90000000.0	Dsit - Finar grant-i To Arm
2	01/02/2024	Grant-in-aid To Arms Length Bodies	Dsit - Science, Innovation And Growth - Dsit -...	Ukri - Medical Research Council	566725	36000000.0	Dsit Finar grant-i To L
3	01/02/2024	Grant-in-aid To Arms Length Bodies	Dsit - Science, Innovation And Growth - Dsit -...	Ukri - Biotechnology And Biological Science Re...	566726	10000000.0	Res Co Pe Sc (B
4	01/02/2024	Grant-in-aid To Arms Length Bodies	Dsit - Science, Innovation And Growth - Dsit -...	Ukri - Biotechnology And Biological Science Re...	566730	22000000.0	Dsit - Finar grant-i To Arm
...	...	...	...	...	...	...	
3491	31/12/2024	Current Grants To Private Sector - Npish	Dsit - Digital And Technology Group - Dsit - C...	The Uk Cyber Security Council	647161	75940.76	Dsit - (Se Co cu Grar
3492	31/12/2024	Capital Grants To Private Sector - Companies	Dsit - Digital And Technology Group - Dsit - D...	University Of Surrey	647122	1419423.64	Dsit- fonrc-f Ne Researc

	Date of Payment	Expense Type	Expense Area	Supplier	Transaction Number	Amount	Descri
3493	31/12/2024	R&D Current Grants To Public Corporations	Dsit - Science, Innovation And Growth - Dsit -...	Npl Management Ltd	647057	261483.81	Dsit Na T Centre Curr
3494	31/12/2024	Faststream - Full Cost	Dsit - Corporate Services - Dsit - Human Resou...	Cabinet Office	647049	33568.0	Dev Ac faststr Ful
3495	31/12/2024	Faststream - Full Cost	Dsit - Corporate Services - Dsit - Human Resou...	Cabinet Office	647065	92831.0	Dev Ac faststr Ful

3496 rows × 9 columns

Cleaning and checking quality data through 4 steps

- Remove / check duplicates data
- Handel null values
- Standradize Data
- Remove unnecessary coulmns or rows

- checking missing data

In [3]:
```
missing_data = newdf.isna().sum()
missing_data[missing_data > 0 ].sort_values(ascending=False)
```

Out[3]:
```
Supplier Post Code    16
Description            3
dtype: int64
```

- Show all duplicate rows, with full details

In [4]:
```
# Show all duplicate rows, with full details
duplicates = newdf[newdf.duplicated(subset=['Transaction Number'], keep= False)]
duplicates.sort_values('Transaction Number')
```

Out[4]:

	Date of Payment	Expense Type	Expense Area	Supplier	Transaction Number	Amount	Descrip
1197	08/01/2024	R & D Current Grants To Private Sector - Npish	Dsit - Science, Innovation And Growth - Dsit -...	The British Academy	561995	1925839.0	Dsit - B Acade & D Cu Grants
1195	08/01/2024	R & D Current Grants To Private Sector - Npish	Dsit - Science, Innovation And Growth - Dsit -...	The British Academy	561995	127881.0	Dsit - B Acad Transit Meas
1201	08/01/2024	R & D Current Grants To Private Sector - Npish	Dsit - Science, Innovation And Growth - Dsit -...	Royal Academy Of Engineering Rae	561997	3190723.32	Dsit - F Academ Enginee r & D C
1198	08/01/2024	R & D Current Grants To Private Sector - Npish	Dsit - Science, Innovation And Growth - Dsit -...	Royal Academy Of Engineering Rae	561997	68221.22	Dsit - F Academ Engine Transit
1193	08/01/2024	R & D Current Grants To Private Sector - Npish	Dsit - Science, Innovation And Growth - Dsit -...	Academy Of Medical Sciences	561998	540498.0	D Academ Me Science D Cu
...	...	...	...	...	...	...	
2331	20/06/2025	R&D Current Grants To Public Corporations	Dsit - Science, Innovation And Growth - Dsit -...	Met Office	696974	115960.86	Dsit - Of Stra Prio Fu
2832	25/06/2025	R&D Current Grants To Public Corporations	Dsit - Science, Innovation And Growth - Dsit -...	Met Office	698016	120012.55	Dsit - Of Clear Analy S
2833	25/06/2025	R&D Current Grants To	Dsit - Science, Innovation	Met Office	698016	39432.25	Dsit - Of Clear

	Date of Payment	Expense Type	Expense Area	Supplier	Transaction Number	Amount	Descrip
		Public Corporations	And Growth - Dsit -...				Analy S
3358	30/06/2025	Cl - Cash Cfers Paid Over To Hmt	Dsit - Digital And Technology Group - Dsit - D...	Consolidated Fund Account 6622	699919	48732174.44	Dsit- O Recei Centra Ca
3360	30/06/2025	Cl - Cash Cfers Paid Over To Hmt	Dsit - Digital And Technology Group - Dsit - D...	Consolidated Fund Account 6622	699919	620966.24	Dsit- Recei Ce (O S

417 rows × 9 columns

- Get Transaction Numbers that appear more than once
- How many each Transaction occures

In [5]:
```
duplicate_counts = newdf['Transaction Number'].value_counts()
duplicate_counts = duplicate_counts[duplicate_counts > 1]
duplicate_counts
```

Out[5]:
```
Transaction Number
618334    7
593836    6
611418    6
593837    6
672067    6
         ..
632135    2
575233    2
575235    2
575223    2
661677    2
Name: count, Length: 173, dtype: int64
```

- Number of duplicates

In [6]:
```
duplicates = newdf[newdf.duplicated(subset=['Transaction Number'], keep=False)]
len(duplicates)
```

Out[6]:
```
417
```

Validate date ranges and spot any anomalous formats.

In [7]:
```
# Convert to datetime and catch format errors
newdf['Date of Payment'] = pd.to_datetime(newdf['Date of Payment'], errors='coerce'

# Find invalid formats (became NaT after conversion)
print(f"Invalid date formats: {newdf['Date of Payment'].isna().sum()}")

# Check date range
print(f"Date range: {newdf['Date of Payment'].min()} to {newdf['Date of Payment'].m

# Find dates outside expected range
anomalies = newdf[
    (newdf['Date of Payment'] < '2024-01-01') |
    (newdf['Date of Payment'] > '2025-12-31')
]
print(f"Dates outside 2024-2025: {len(anomalies)}")

# Show any anomalies found
if len(anomalies) > 0:
    print(anomalies[['Date of Payment', 'Supplier', 'Amount']])
```

```
Invalid date formats: 1856
Date range: 2024-01-02 00:00:00 to 2025-12-06 00:00:00
Dates outside 2024-2025: 0
```

Removing null values in (dates of payment) column

In [8]:
```
newdf  = newdf[newdf['Date of Payment'].notna()].copy()
```

- Display invalid dates

In [9]:
```
invalid_dates = newdf[newdf['Date of Payment'].isna()]
print(f"Invalid dates found: {len(invalid_dates)}")
```

```
Invalid dates found: 0
```

In [10]:
```
# After removing
print(f"Rows after: {len(newdf)}")
print(f"Missing dates now: {newdf['Date of Payment'].isna().sum()}")
```

```
Rows after: 1640
Missing dates now: 0
```

- Duplicated Transaction Numbers after removing duplicates

In [11]:
```
duplicates = newdf[newdf.duplicated(subset=['Transaction Number'], keep=False)]
print(f'Duplicated Transaction Numbers after : {len(duplicates)}')
```

```
Duplicated Transaction Numbers after : 160
```

In [12]:
```
duplicate_counts = newdf['Transaction Number'].value_counts()
duplicate_counts = duplicate_counts[duplicate_counts > 1]
duplicate_counts
```

Out[12]:
```
Transaction Number
593836    6
672067    6
694164    5
575225    4
672550    3
          ..
575233    2
575235    2
575655    2
586609    2
587725    2
Name: count, Length: 72, dtype: int64
```

This code takes messy money values like "£134,394.66" and cleans them up into pure numbers like 134394.66.

In [13]:
```
newdf['Amount'] = newdf['Amount'].astype(str).str.replace('£', '').str.replace(',',
# newdf['Amount'] = pd.to_numeric(newdf['Amount'], errors='coerce')
newdf['Amount'].head()
```

Out[13]:
```
0      134394.66
1    90000000.00
2    36000000.00
3    10000000.00
4    22000000.00
Name: Amount, dtype: float64
```

- Getting a copy of the cleaned datasest

In [14]:
```
newdf.to_csv('master_spend_cleand_data.csv', index=False, encoding='utf-8')
```