

Cleaning the File

```
import pandas as pd

# Read the original CSV file
input_file = 'DoL_Contracts_PrimeTransactions_2024-11-04_H22M57S28_1.csv'
output_file = 'DoL_output_file.csv' # Changed output file name

# List of columns to keep
columns_to_keep = [
    'contract_transaction_unique_key', 'parent_award_agency_name',
    'current_total_value_of_award',
    'period_of_performance_start_date',
    'period_of_performance_potential_end_date',
    'awarding_agency_name', 'funding_sub_agency_name',
    'object_classes_funding_this_award',
    'recipient_name', 'recipient_state_name',
    'primary_place_of_performance_state_name',
    'award_type', 'transaction_description',
    'prime_award_base_transaction_description',
    'product_or_service_code_description', 'naics_description',
    'recovered_materials_sustainability',
    'information_technology_commercial_item_category',
    'extent_competed', 'solicitation_procedures',
    'evaluated_preference', 'fair_opportunity_limited_sources',
    'other_than_full_and_open_competition',
    'number_of_offers_received', 'clinger_cohen_act_planning_code',
    'materials_supplies_articles_equipment',
    'labor_standards', 'performance_based_service_acquisition',
    'contingency_humanitarian_or_peacekeeping_operation',
    'minority_owned_business',
    'black_american_owned_business',
    'hispanic_american_owned_business', 'native_american_owned_business',
    'woman_owned_business', 'organizational_type'
]

# Read the CSV file, selecting only the specified columns
df = pd.read_csv(input_file, usecols=columns_to_keep)
df = df[df['awarding_agency_name'] == 'Department of Labor']

# Write the filtered data to a new CSV file
df.to_csv(output_file, index=False)

print(f"Filtered CSV file has been created: {output_file}")

Filtered CSV file has been created: DoL_output_file.csv
```

What is the total number of AI/ML contracts?

```
import pandas as pd
import numpy as np

# File name
file_name = 'DoL_output_file.csv'

# Read the CSV file
df = pd.read_csv(file_name)

# Filter for Department of Homeland Security contracts
dol_df = df[df['awarding_agency_name'] == 'Department of Labor']

# Count the number of DoL contracts
dol_contract_count = len(dol_df)

# Calculate total value of DoL contracts
dol_total_value = dol_df['current_total_value_of_award'].sum()

# Get unique vendors for DoL contracts
dol_unique_vendors = dol_df['recipient_name'].nunique()

# Print results
print(f"Total number of contracts: {len(df)}")
print(f"Number of Department of Defence contracts: {dol_contract_count}")
print(f"Percentage of DoL contracts: {(dol_contract_count / len(df)) * 100:.2f}%")
print(f"Total value of DoL contracts: ${dol_total_value:,.2f}")
print(f"Number of unique vendors for DoL contracts: {dol_unique_vendors}")

# Top 5 vendors for DoL by number of contracts
top_dol_vendors = dol_df['recipient_name'].value_counts().head(5)
print("\nTop 5 vendors for DoL by number of contracts:")
for vendor, count in top_dol_vendors.items():
    print(f"{vendor}: {count} contracts")

# Top 5 NAICS descriptions for DoL contracts
top_dol_naics = dol_df['naics_description'].value_counts().head(5)
print("\nTop 5 NAICS descriptions for DoL contracts:")
for desc, count in top_dol_naics.items():
    print(f"{desc}: {count} contracts")

# Optional: Distribution of contract values for DoL
print("\nDistribution of DoL contract values:")
print(dol_df['current_total_value_of_award'].describe())

# Check for any contracts with other awarding agencies
other_agencies = df[df['awarding_agency_name'] != 'Department of']
```

```

Homeland Security'] ['awarding_agency_name'].unique()
if len(other_agencies) > 0:
    print("\nOther awarding agencies found in the dataset:")
    for agency in other_agencies:
        count = df[df['awarding_agency_name'] == agency].shape[0]
        print(f"{agency}: {count} contracts")
else:
    print("\nAll contracts in the dataset are from the Department of
Homeland Security.")

```

```

Total number of contracts: 1
Number of Department of Defence contracts: 1
Percentage of DoL contracts: 100.00%
Total value of DoL contracts: $99,099.00
Number of unique vendors for DoL contracts: 1

```

```

Top 5 vendors for DoL by number of contracts:
DELOITTE CONSULTING LLP: 1 contracts

```

```

Top 5 NAICS descriptions for DoL contracts:
COMPUTER SYSTEMS DESIGN SERVICES: 1 contracts

```

```

Distribution of DoL contract values:
count      1.0
mean      99099.0
std         NaN
min       99099.0
25%       99099.0
50%       99099.0
75%       99099.0
max       99099.0
Name: current_total_value_of_award, dtype: float64

```

```

Other awarding agencies found in the dataset:
Department of Labor: 1 contracts

```

What is the total spending on AI/ML contracts?

```

import pandas as pd
import numpy as np

# File name
file_name = 'DoL_output_file.csv'

# Read the CSV file
df = pd.read_csv(file_name)

# Ensure current_total_value_of_award is numeric
df['current_total_value_of_award'] =
pd.to_numeric(df['current_total_value_of_award'], errors='coerce')

```

```

# Calculate the sum
total_award_value = df['current_total_value_of_award'].sum()

# Print the result
print(f"Sum of current_total_value_of_award: $
{total_award_value:,.2f}")

# Optional: Print some additional statistics
print("\nAdditional statistics:")
print(f"Mean award value: $
{df['current_total_value_of_award'].mean():,.2f}")
print(f"Median award value: $
{df['current_total_value_of_award'].median():,.2f}")
print(f"Maximum award value: $
{df['current_total_value_of_award'].max():,.2f}")
print(f"Minimum award value: $
{df['current_total_value_of_award'].min():,.2f}")

# Check for any null or zero values
null_count = df['current_total_value_of_award'].isnull().sum()
zero_count = (df['current_total_value_of_award'] == 0).sum()
print(f"\nNumber of null values: {null_count}")
print(f"Number of zero values: {zero_count}")

Sum of current_total_value_of_award: $99,099.00

Additional statistics:
Mean award value: $99,099.00
Median award value: $99,099.00
Maximum award value: $99,099.00
Minimum award value: $99,099.00

Number of null values: 0
Number of zero values: 0

```

What proportion of AI procurement contracts awarded to minority-owned?

```

import pandas as pd

# Read the CSV file
df = pd.read_csv('DoL_output_file.csv')

# List of columns to check
columns_to_check = [
    'minority_owned_business',
    'black_american_owned_business',
    'hispanic_american_owned_business',
    'native_american_owned_business',

```

```

        'woman_owned_business'
    ]

    # Count 't' occurrences for each column
    for column in columns_to_check:
        count = df[column].eq('t').sum()
        print(f"Number of 't' in {column}: {count}")

    # Calculate percentage for minority_owned_business
    total_entries = len(df)
    minority_owned_count = df['minority_owned_business'].eq('t').sum()
    percentage = (minority_owned_count / total_entries) * 100

    print(f"\nTotal entries: {total_entries}")
    print(f"Number of 't' in minority_owned_business: {minority_owned_count}")
    print(f"Percentage of minority-owned businesses: {percentage:.2f}%")

    Number of 't' in minority_owned_business: 0
    Number of 't' in black_american_owned_business: 0
    Number of 't' in hispanic_american_owned_business: 0
    Number of 't' in native_american_owned_business: 0
    Number of 't' in woman_owned_business: 0

    Total entries: 1
    Number of 't' in minority_owned_business: 0
    Percentage of minority-owned businesses: 0.00%

```

How clear and detailed are the transaction descriptions and product/service descriptions in AI procurement contracts? (What is the average number of words used in AI contract descriptions?)

```

import pandas as pd

# File name
file_name = 'DoL_output_file.csv'

# Read the CSV file
df = pd.read_csv(file_name)

# Function to count words in a string
def word_count(string):
    return len(str(string).split())

# Apply word count function to transaction_description column
df['word_count'] = df['transaction_description'].apply(word_count)

# Calculate average word count
average_word_count = df['word_count'].mean()

```

```

print(f"Average number of words in transaction descriptions:
{average_word_count:.2f}")

# Find row with highest word count
max_word_count_row = df.loc[df['word_count'].idxmax()]
print("\nRow with highest word count:")
print(f"Word count: {max_word_count_row['word_count']}")
print(f>Description: {max_word_count_row['transaction_description']}")

# Find row with lowest word count (excluding empty descriptions)
min_word_count_row = df[df['word_count'] >
0].loc[df['word_count'].idxmin()]
print("\nRow with lowest word count (excluding empty descriptions):")
print(f"Word count: {min_word_count_row['word_count']}")
print(f>Description: {min_word_count_row['transaction_description']}")

# Optional: Display some statistics
print("\nWord count statistics:")
print(df['word_count'].describe())

# Count how many descriptions mention 'AI' or 'artificial
intelligence'
ai_mentions = df['transaction_description'].str.contains('AI|
artificial intelligence', case=False, na=False).sum()
print(f"\nNumber of descriptions mentioning AI: {ai_mentions}")

# Calculate average word count for AI-related descriptions
ai_descriptions = df[df['transaction_description'].str.contains('AI|
artificial intelligence', case=False, na=False)]
ai_average_word_count = ai_descriptions['word_count'].mean() if not
ai_descriptions.empty else 0

print(f"Average number of words in AI-related contract descriptions:
{ai_average_word_count:.2f}")

```

Average number of words in transaction descriptions: 19.00

Row with highest word count:

Word count: 19

Description: DOL CYBERSECURITY AND INFORMATION ASSURANCE REVIEW OF
POTENTIAL ARTIFICIAL INTELLIGENCE USAGE IN THE OCCUPATIONAL SAFETY AND
HEALTH ADMINISTRATION (OSHA)

Row with lowest word count (excluding empty descriptions):

Word count: 19

Description: DOL CYBERSECURITY AND INFORMATION ASSURANCE REVIEW OF
POTENTIAL ARTIFICIAL INTELLIGENCE USAGE IN THE OCCUPATIONAL SAFETY AND
HEALTH ADMINISTRATION (OSHA)

Word count statistics:

```
count      1.0
mean       19.0
std        NaN
min        19.0
25%        19.0
50%        19.0
75%        19.0
max        19.0
Name: word_count, dtype: float64
```

Number of descriptions mentioning AI: 1

Average number of words in AI-related contract descriptions: 19.00

What is the ratio of offers received to contracts awarded in AI procurements, indicating the level of competitiveness?

```
import pandas as pd
import numpy as np

# File name
file_name = 'DoL_output_file.csv'

# Read the CSV file
df = pd.read_csv(file_name)

# Ensure 'number_of_offers_received' is numeric
df['number_of_offers_received'] =
pd.to_numeric(df['number_of_offers_received'], errors='coerce')

# Calculate overall statistics
total_contracts = len(df)
total_offers = df['number_of_offers_received'].sum()
avg_offers_per_contract = total_offers / total_contracts

print(f"Total contracts: {total_contracts}")
print(f"Total offers received: {total_offers}")
print(f"Average offers per contract: {avg_offers_per_contract:.2f}")
print(f"Ratio of offers to contracts: {avg_offers_per_contract:.2f} : 1")

# Distribution of offers
print("\nDistribution of offers received:")
print(df['number_of_offers_received'].describe())

# Categorize competitiveness
df['competitiveness'] = pd.cut(df['number_of_offers_received'],
                               bins=[-np.inf, 1, 3, 5, np.inf],
                               labels=['Single offer', 'Low
competition', 'Moderate competition', 'High competition'])
```

```

print("\nCompetitiveness breakdown:")
print(df['competitiveness'].value_counts(normalize=True).sort_index().
mul(100).round(2))

# Contracts with highest number of offers
top_competitive = df.nlargest(5, 'number_of_offers_received')
print("\nTop 5 most competitive contracts:")
print(top_competitive[['contract_transaction_unique_key',
'number_of_offers_received']])

# Percentage of contracts with only one offer
single_offer_percentage = (df['number_of_offers_received'] ==
1).mean() * 100
print(f"\nPercentage of contracts with only one offer:
{single_offer_percentage:.2f}%")

Total contracts: 1
Total offers received: 0.0
Average offers per contract: 0.00
Ratio of offers to contracts: 0.00 : 1

Distribution of offers received:
count      0.0
mean       NaN
std        NaN
min        NaN
25%        NaN
50%        NaN
75%        NaN
max        NaN
Name: number_of_offers_received, dtype: float64

Competitiveness breakdown:
Single offer      NaN
Low competition   NaN
Moderate competition NaN
High competition  NaN
Name: competitiveness, dtype: float64

Top 5 most competitive contracts:
              contract_transaction_unique_key
number_of_offers_received
0  1605_1605_1605TA24F00053_0_1605DC19A0016_0
NaN

Percentage of contracts with only one offer: 0.00%

```


What percentage of AI procurement contracts meet established labor standards?

```
import pandas as pd

# File name
file_name = 'DoL_output_file.csv'

# Read the CSV file
df = pd.read_csv(file_name)

# Ensure the column name is correct and data is cleaned
df['labor_standards'] = df['labor_standards'].str.upper().str.strip()

# Count the occurrences of each category
total_contracts = len(df)
yes_count = (df['labor_standards'] == 'YES').sum()
no_count = (df['labor_standards'] == 'NO').sum()
na_count = (df['labor_standards'] == 'NOT APPLICABLE').sum()

# Calculate percentages
yes_percentage = (yes_count / total_contracts) * 100
no_percentage = (no_count / total_contracts) * 100
na_percentage = (na_count / total_contracts) * 100

# Print results
print(f"Total number of contracts: {total_contracts}")
print(f"\nContracts meeting labor standards (YES):")
print(f"Count: {yes_count}")
print(f"Percentage: {yes_percentage:.2f}%")

print(f"\nContracts not meeting labor standards (NO):")
print(f"Count: {no_count}")
print(f"Percentage: {no_percentage:.2f}%")

print(f"\nContracts where labor standards are not applicable:")
print(f"Count: {na_count}")
print(f"Percentage: {na_percentage:.2f}%")

# Check for any other values
other_count = total_contracts - (yes_count + no_count + na_count)
if other_count > 0:
    print(f"\nContracts with other values:")
    print(f"Count: {other_count}")
    print(f"Percentage: {(other_count / total_contracts) * 100:.2f}%")
    print("\nUnique values in labor_standards column:")
    print(df['labor_standards'].value_counts())
```

Total number of contracts: 1

Contracts meeting labor standards (YES):

Count: 0
Percentage: 0.00%

Contracts not meeting labor standards (N0):
Count: 1
Percentage: 100.00%

Contracts where labor standards are not applicable:
Count: 0
Percentage: 0.00%

Are performance-based criteria present in AI procurement contracts to ensure service delivery accountability?

```
import pandas as pd

# File name
file_name = 'DoL_output_file.csv'

# Read the CSV file
df = pd.read_csv(file_name)

# Ensure the column name is correct and data is cleaned
df['performance_based_service_acquisition'] =
df['performance_based_service_acquisition'].str.upper().str.strip()

# Get value counts and percentages
value_counts =
df['performance_based_service_acquisition'].value_counts()
value_percentages =
df['performance_based_service_acquisition'].value_counts(normalize=True)
* 100

# Total number of contracts
total_contracts = len(df)

# Print results
print(f"Total number of contracts: {total_contracts}")
print("\nUnique values in 'performance_based_service_acquisition'
column:")
print("\nValue          Count      Percentage")
print("-" * 40)

for value, count in value_counts.items():
    percentage = value_percentages[value]
    print(f"{value:<16} {count:<9} {percentage:.2f}%")

# Check for null values
null_count =
df['performance_based_service_acquisition'].isnull().sum()
```

```

if null_count > 0:
    null_percentage = (null_count / total_contracts) * 100
    print(f"\nNull values:      {null_count:<9} {null_percentage:.2f}%")

# Number of unique values
num_unique = len(value_counts)
print(f"\nNumber of unique values: {num_unique}")

Total number of contracts: 1

Unique values in 'performance_based_service_acquisition' column:

Value                Count      Percentage
-----
NO - SERVICE WHERE PBA IS NOT USED. 1      100.00%

Number of unique values: 1

```

How was the contract awarded—through a competitive process or a sole-source arrangement?

```

import pandas as pd

# File name
file_name = 'DoL_output_file.csv'

# Read the CSV file
df = pd.read_csv(file_name)

# Clean the column
df['solicitation_procedures'] =
df['solicitation_procedures'].str.strip().str.upper()

# Get value counts and percentages
value_counts = df['solicitation_procedures'].value_counts()
value_percentages =
df['solicitation_procedures'].value_counts(normalize=True) * 100

# Total number of contracts
total_contracts = len(df)

# Print results
print(f"Total number of contracts: {total_contracts}")
print("\nSolicitation Procedures Breakdown:")
print("\nProcedure                                Count
Percentage")
print("-" * 70)

for value, count in value_counts.items():

```

```

percentage = value_percentages[value]
print(f"{value:<40} {count:<9} {percentage:.2f}%")

# Check for null or empty values
null_count = df['solicitation_procedures'].isnull().sum()
empty_count = (df['solicitation_procedures'] == '').sum()
if null_count > 0 or empty_count > 0:
    print(f"\nNull values: {null_count}")
    print(f"Empty values: {empty_count}")

# Number of unique values
num_unique = len(value_counts)
print(f"\nNumber of unique solicitation procedures: {num_unique}")

# Analyze the types of procedures
competitive_procedures = ['FULL AND OPEN COMPETITION', 'COMPETITIVE
DELIVERY ORDER', 'FULL AND OPEN COMPETITION AFTER EXCLUSION OF
SOURCES']
competitive_count =
df['solicitation_procedures'].isin(competitive_procedures).sum()
competitive_percentage = (competitive_count / total_contracts) * 100

print(f"\nContracts with clearly competitive procedures:
{competitive_count} ({competitive_percentage:.2f}%)")

# Check for specific AI-related keywords in other columns
ai_keywords = ['AI', 'ARTIFICIAL INTELLIGENCE', 'MACHINE LEARNING',
'DEEP LEARNING']
ai_related_count =
df['transaction_description'].str.contains('|'.join(ai_keywords),
case=False, na=False).sum()
ai_related_percentage = (ai_related_count / total_contracts) * 100

print(f"\nContracts potentially related to AI: {ai_related_count}
({ai_related_percentage:.2f}%)")

```

Total number of contracts: 1

Solicitation Procedures Breakdown:

Procedure	Count	Percentage
SUBJECT TO MULTIPLE AWARD FAIR OPPORTUNITY	1	100.00%

Number of unique solicitation procedures: 1

Contracts with clearly competitive procedures: 0 (0.00%)

Contracts potentially related to AI: 1 (100.00%)

How well do AI procurements align with IT standards, such as those specified by the Clinger-Cohen Act?

```
import pandas as pd
import numpy as np

# File name
file_name = 'DoL_output_file.csv'

# Read the CSV file
df = pd.read_csv(file_name)

# Clean the column
df['clinger_cohen_act_planning_code'] =
df['clinger_cohen_act_planning_code'].str.strip().str.upper()

# Get value counts and percentages
value_counts = df['clinger_cohen_act_planning_code'].value_counts()
value_percentages =
df['clinger_cohen_act_planning_code'].value_counts(normalize=True) *
100

# Total number of contracts
total_contracts = len(df)

# Print results
print(f"Total number of contracts: {total_contracts}")
print("\nClinger-Cohen Act Planning Code Breakdown:")
print("\nCode          Count      Percentage")
print("-" * 50)

for value, count in value_counts.items():
    percentage = value_percentages[value]
    print(f"{value:<25} {count:<9} {percentage:.2f}%")

# Check for null or empty values
null_count = df['clinger_cohen_act_planning_code'].isnull().sum()
empty_count = (df['clinger_cohen_act_planning_code'] == '').sum()
if null_count > 0 or empty_count > 0:
    print(f"\nNull values: {null_count}")
    print(f"Empty values: {empty_count}")

# Number of unique values
num_unique = len(value_counts)
print(f"\nNumber of unique Clinger-Cohen Act Planning Codes:
{num_unique}")

# Analyze compliance
compliant_codes = ['Y', 'YES']
compliant_count =
```

```

df['clinger_cohen_act_planning_code'].isin(compliant_codes).sum()
compliant_percentage = (compliant_count / total_contracts) * 100

print(f"\nContracts compliant with Clinger-Cohen Act:
{compliant_count} ({compliant_percentage:.2f}%)")

# Check for AI-related keywords in transaction description
ai_keywords = ['AI', 'ARTIFICIAL INTELLIGENCE', 'MACHINE LEARNING',
'DEEP LEARNING']
df['is_ai_related'] =
df['transaction_description'].str.contains('|'.join(ai_keywords),
case=False, na=False)

ai_related_count = df['is_ai_related'].sum()
ai_related_percentage = (ai_related_count / total_contracts) * 100

print(f"\nPotentially AI-related contracts: {ai_related_count}
({ai_related_percentage:.2f}%)")

# Analyze Clinger-Cohen Act compliance for AI-related contracts
ai_compliant_count = df[df['is_ai_related'] &
df['clinger_cohen_act_planning_code'].isin(compliant_codes)].shape[0]
ai_compliant_percentage = (ai_compliant_count / ai_related_count *
100) if ai_related_count > 0 else 0

print(f"\nAI-related contracts compliant with Clinger-Cohen Act:
{ai_compliant_count} ({ai_compliant_percentage:.2f}%)")

```

Total number of contracts: 1

Clinger-Cohen Act Planning Code Breakdown:

Code	Count	Percentage
Y	1	100.00%

Number of unique Clinger-Cohen Act Planning Codes: 1

Contracts compliant with Clinger-Cohen Act: 1 (100.00%)

Potentially AI-related contracts: 1 (100.00%)

AI-related contracts compliant with Clinger-Cohen Act: 1 (100.00%)

Who are the top three main vendors? (value and number of contracts)

```

import pandas as pd
import numpy as np

# File name
file_name = 'DoL_output_file.csv'

```

```

# Read the CSV file
df = pd.read_csv(file_name)

# Clean the recipient_name column
df['recipient_name'] = df['recipient_name'].str.strip().str.upper()

# Ensure current_total_value_of_award is numeric
df['current_total_value_of_award'] =
pd.to_numeric(df['current_total_value_of_award'], errors='coerce')

# Count unique vendors
unique_vendors = df['recipient_name'].nunique()

# Get top vendors by number of contracts
top_vendors_by_contracts = df['recipient_name'].value_counts().head(3)

# Get top vendors by total award value
vendor_stats = df.groupby('recipient_name').agg({
    'current_total_value_of_award': 'sum',
    'recipient_name': 'count'
}).rename(columns={'recipient_name': 'contract_count'})

top_vendors_by_value = vendor_stats.nlargest(3,
'current_total_value_of_award')

# Get unique vendor names
unique_vendor_names = df['recipient_name'].unique()

# Total number of contracts and total award value
total_contracts = len(df)
total_award_value = df['current_total_value_of_award'].sum()

# Print results
print(f"Total number of contracts: {total_contracts}")
print(f"Total award value: ${total_award_value:,.2f}")
print(f"Number of unique vendors: {unique_vendors}")

print("\nTop 3 vendors by number of contracts:")
for vendor, count in top_vendors_by_contracts.items():
    percentage = (count / total_contracts) * 100
    print(f"{vendor}: {count} contracts ({percentage:,.2f}%)")

print("\nTop 3 vendors by total award value:")
for vendor, row in top_vendors_by_value.iterrows():
    value = row['current_total_value_of_award']
    count = row['contract_count']
    value_percentage = (value / total_award_value) * 100
    count_percentage = (count / total_contracts) * 100
    print(f"{vendor}:")

```

```

    print(f" Total value: ${value:,.2f} ({value_percentage:.2f}% of
total value)")
    print(f" Number of contracts: {count} ({count_percentage:.2f}% of
total contracts)")

# Print first 20 unique vendor names
print("\nFirst 20 unique vendor names:")
for i, name in enumerate(unique_vendor_names[:20], 1):
    print(f"{i}. {name}")

if len(unique_vendor_names) > 20:
    print(f"... and {len(unique_vendor_names) - 20} more.")

# Optional: Display distribution of contracts among vendors
print("\nDistribution of contracts among vendors:")
vendor_contract_counts = df['recipient_name'].value_counts()
print(vendor_contract_counts.describe())

# Optional: Check for any unnamed or generic vendors
unnamed_count = df['recipient_name'].isin(['', 'UNNAMED', 'UNKNOWN',
'N/A']).sum()
if unnamed_count > 0:
    print(f"\nContracts with unnamed or generic vendors:
{unnamed_count}")

Total number of contracts: 1
Total award value: $99,099.00
Number of unique vendors: 1

Top 3 vendors by number of contracts:
DELOITTE CONSULTING LLP: 1 contracts (100.00%)

Top 3 vendors by total award value:
DELOITTE CONSULTING LLP:
    Total value: $99,099.00 (100.00% of total value)
    Number of contracts: 1.0 (100.00% of total contracts)

First 20 unique vendor names:
1. DELOITTE CONSULTING LLP

Distribution of contracts among vendors:
count      1.0
mean       1.0
std        NaN
min        1.0
25%        1.0
50%        1.0
75%        1.0
max        1.0
Name: recipient_name, dtype: float64

```


Vendor details and types of services they provide

```
import pandas as pd
import numpy as np

# File name
file_name = 'DoL_output_file.csv'

# Read the CSV file
df = pd.read_csv(file_name)

# Clean the recipient_name column
df['recipient_name'] = df['recipient_name'].str.strip().str.upper()

# Ensure current_total_value_of_award is numeric
df['current_total_value_of_award'] =
pd.to_numeric(df['current_total_value_of_award'], errors='coerce')

# Get top 3 vendors by number of contracts
top_vendors_by_contracts = df['recipient_name'].value_counts().head(3)

# Get top 3 vendors by total award value
vendor_stats = df.groupby('recipient_name').agg({
    'current_total_value_of_award': 'sum',
    'recipient_name': 'count'
}).rename(columns={'recipient_name': 'contract_count'})

top_vendors_by_value = vendor_stats.nlargest(3,
'current_total_value_of_award')

# Function to get NAICS descriptions for a vendor
def get_naics_descriptions(vendor_name):
    vendor_contracts = df[df['recipient_name'] == vendor_name]
    naics_desc = vendor_contracts['naics_description'].value_counts()
    return naics_desc.head(5) # Return top 5 NAICS descriptions

# Print results
print("Top 3 vendors by number of contracts:")
for vendor, count in top_vendors_by_contracts.items():
    print(f"\n{vendor}: {count} contracts")
print("Top 5 NAICS descriptions:")
naics_desc = get_naics_descriptions(vendor)
for desc, freq in naics_desc.items():
    print(f"  - {desc}: {freq} contracts")

print("\nTop 3 vendors by total award value:")
for vendor, row in top_vendors_by_value.iterrows():
    value = row['current_total_value_of_award']
    count = row['contract_count']
    print(f"\n{vendor}:")
    print(f"  Total value: ${value:,.2f}")
```

```

print(f"  Number of contracts: {count}")
print("Top 5 NAICS descriptions:")
naics_desc = get_naics_descriptions(vendor)
for desc, freq in naics_desc.items():
    print(f"  - {desc}: {freq} contracts")

# Calculate and print total award value
total_award_value = df['current_total_value_of_award'].sum()
print(f"\nTotal award value across all contracts: $
{total_award_value:,.2f}")

```

Top 3 vendors by number of contracts:

DELOITTE CONSULTING LLP: 1 contracts

Top 5 NAICS descriptions:

- COMPUTER SYSTEMS DESIGN SERVICES: 1 contracts

Top 3 vendors by total award value:

DELOITTE CONSULTING LLP:

Total value: \$99,099.00

Number of contracts: 1.0

Top 5 NAICS descriptions:

- COMPUTER SYSTEMS DESIGN SERVICES: 1 contracts

Total award value across all contracts: \$99,099.00