Part 1: CMS Data Exercise

Here at [redacted] we provide staffing to long-term care facilities. The Centers for Medicaid and Medicare Services publish a quarterly report containing daily staffing data for all registered nursing homes in the U.S. This dataset is called Payroll Based Journal (PBJ) Daily Nurse Staffing and can be found here. You can review the data dictionary to better understand the data that's available, but a few notes pertaining to this exercise:

- Nursing homes are staffed by both employees and contractors. Employees work for the nursing home full time, whereas contractors work for the nursing home on a temporary basis. Most nursing homes are staffed using some combination of full time employees and contractors, but the proportions within each nursing home can vary substantially. All workers from [redacted] (and our competitors) are classified as contractors.
- [redacted] is a nationwide staffing platform, operating in all 50 states.
- There are a variety of other datasets that can be joined to this dataset, all found under the nursing home data section on CMS' website.
- As mentioned previously, the PBJ data is separated by quarter. For this exercise, please focus on the most recent quarter available (2024Q1).

Can you please use the PBJ data and any other CMS data that you see fit to make a few recommendations to the [redacted] sales leadership team? There are no right or wrong answers but a few tips:

- Be sure to include the supporting data (charts, tables, etc.) for any recommendation you make. The purpose of this exercise is largely to see how well you can quickly learn and analyze new datasets, as well as communicate the learnings and recommendations from those analyses.
- Feel free to use any analytical tools that you have at your disposal.
- We do a lot of writing at [redacted] (we believe that writing is thinking), so each recommendation should include some writing around how you analyzed the data, what you uncovered, and the specifics of what you'd recommend the sales team do based on your findings.
- There is no minimum number of recommendations we'd like you to make, but please include no more than 5.

Nursing Contract Staffing Study – 2024 Q1

Executive Summary

- 3 Contract Roles (out of 8) make up 97% of Total contract staffing hours with Certified Nurse Assistant (CNA) making up ~56%.
- Northeast Region offers the most contract hours, NY at the top with 4 Million contract hours in 2024-Q1.
- Focusing on biggest state NY and role Certified Nurse Assistant, 7 counties in NY all located in southeast NY offered over 100K CNA contract hours with Queens at top with around 325K CNA contract hours.

Assumptions and Methodology

- Payroll Based Journal (PBJ) Daily Nurse Staffing Data on CMS.gov contains data for 50 states plus District of Columbia and Puerto Rico -For this analysis only the 50 States have been considered.

- Provider Information Table was used as a secondary table for analysis to narrow down details for key providers.

- Python/ Jupyter Notebook was used to load, clean and explore the datasets.
- Pandas was the main library used as it allows us to join, filter and create summaries.

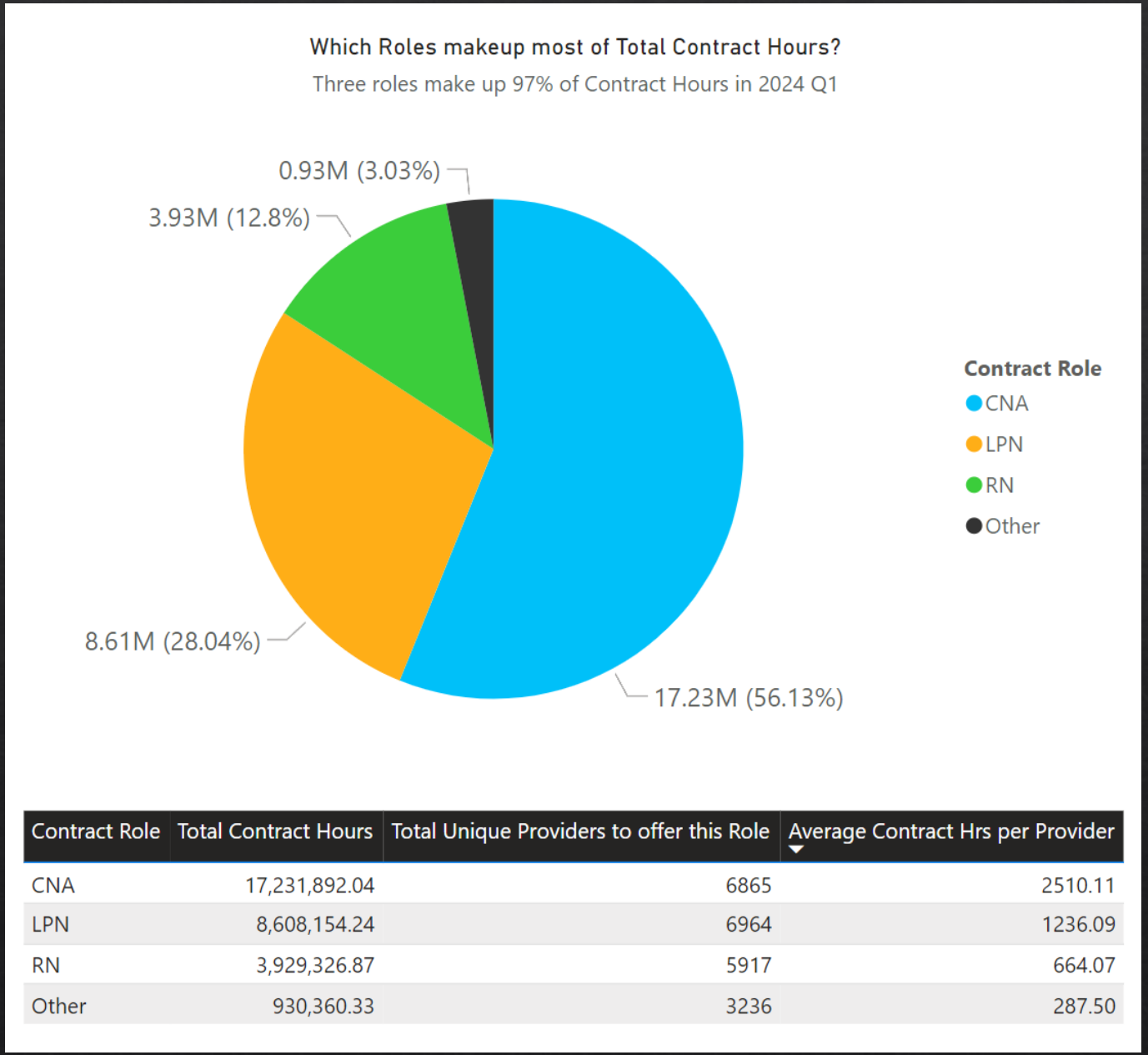
- PowerBI was used for most of the visualizations with Tableau being used for the maps.

Big Picture

- For the first two recommendations we focus on the big picture.
  - How were the contract staffing hours in nursing homes for 2024-Q1 broken down in terms of Job roles?
  - Which states and regions offer the biggest opportunities for Contract Staffing?

Recommendation #1

- Overall, the Contract Hours available were broadly broken into 8 roles.
- 3 of those 8 roles make up 97% of total contract hours in 2024 Q1.
  - Certified Nurse Assistant (CNA) ~56%
  - Licensed Practical Nurse (LPN) ~28%
  - Registered Nurse (RN) ~13%
- Recommendation is to treat these roles as primary markets for Contract Staffing purposes.



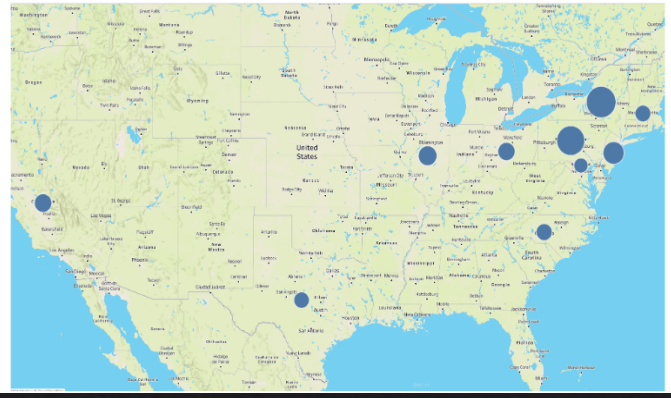
Contract Role	Total Contract Hours	Total Unique Providers to offer this Role	Average Contract Hrs per Provider
CNA	17,231,892.04	6865	2510.11
LPN	8,608,154.24	6964	1236.09
RN	3,929,326.87	5917	664.07
Other	930,360.33	3236	287.50

Recommendation #2

- 10 states make up around 60% of total contract staffing hours in nursing homes.
- NY alone makes up 13.2% of total contract staffing hours, closely followed by PA and NJ. The three combined offer nearly 1/3rd of contract hours!
- NY, PA, NJ, MA, MD are 5 Northeastern states that made it to Top 10 list.
- Recommendation is to treat Northeast as the key market region for building relations with providers or owners.
- The amount of opportunity offered by 3 neighboring states NY, PA, and NJ is hard to ignore!

Most of Top 10 States are located in NE Region

(Size of Bubble indicates total Contract Hours)



NY, PA, and NJ also offer high contract hours per Provider

State	Total Contract Hours	Total Providers	Total contract Hours per Provider
NY	4,039,449.99	601	6,721.21
PA	3,577,744.31	665	5,380.07
NJ	2,036,180.84	345	5,901.97
IL	1,557,327.83	675	2,307.15
CA	1,346,552.77	1140	1,181.19
OH	1,304,113.78	918	1,420.60
NC	1,126,111.32	410	2,746.61
MA	1,090,336.45	342	3,188.12
TX	1,037,825.88	1164	891.60
MD	870,030.27	220	3,954.68

Zoning In!

- For the third recommendation we zone in on the biggest state, NY and biggest role CNA.
- NY had 1.9M contract staffing hours for CNA role, nearly half of the state.
- That is more than total contract hours offered by 4th highest state! (IL offered 1.56M total contract hours)

  - How many NY providers offer CNA contract hours?

  - Which NY counties offers most opportunities for CNA Contract Staffing Hours?

  - Based on NY counties and Day of the week selected, which providers offered highest average contract hours for that day and location?

    What were the lowest and highest number of hours offered by that provider for same day and location?

Recommendation #3

- 374 Providers in NY offered CNA contract hours, across 51 NY counties.
- 7 NY counties offered 100K+ CNA contract hours all located in Southeast NY.
- Based on the county and day selected, (multiple selection allowed) we get Providers with most consistent offerings for CNA contract hours.

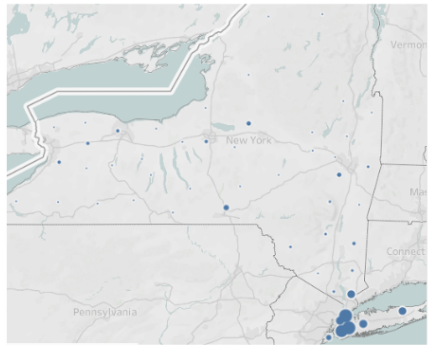
Summary of NY 2024-Q1 CNA Contract Hours with Key Counties and Providers

State	Total Providers	Total CNA Providers	Total County	Counties with CNA Provider	Total CNA Contract Hours
NY	601	374	60	51	1.92M

7 NY Counties offered more than 100K CNA Contract Hours

NY County	Total 2024 Q1 Contract Hours	Unique CNA Contract Hours Providers	CNA contract Hours per Provider
Queens	324,330.09	45	7,207.34
Bronx	310,866.55	40	7,771.66
Kings	273,339.90	38	7,193.16
Nassau	135,224.23	25	5,408.97
New York	126,746.99	15	8,449.80
Suffolk	111,150.09	27	4,116.67
Westchester	109,617.03	27	4,059.89
Richmond	56,137.70	8	7,017.21

Most NY Counties with High CNA Contract Hours are located in SE



NY County: Queens

Day of The Week: Sunday

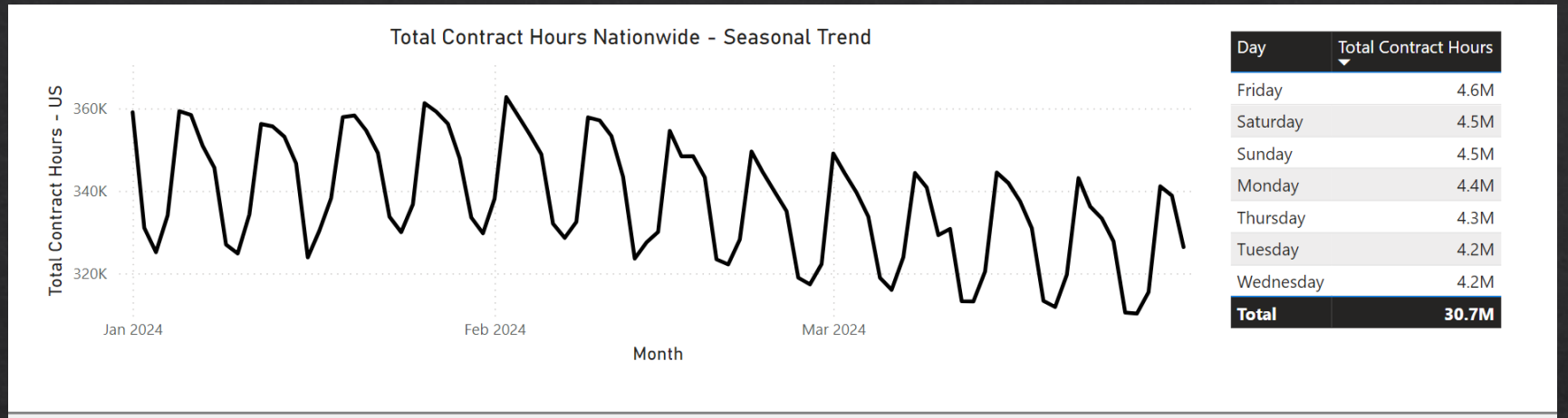
Key Providers for Selected County and Day of The Week

Provider #	AVG CNA Hours	Min CNA Hours	Max CNA Hours	NY County	Provider Name	Provider Address	Telephone Number	ZIP Code
335791	519.46	470.75	573.25	Queens	QUEENS BOULEVARD EXTENDED CARE FACILITY	61 11 QUEENS BOULEVARD	7182050288	11377
335266	223.46	173.75	265.00	Queens	WOODCREST REHAB & RESIDENTIAL H C CENTER, L L C	119 09 26TH AVENUE	7187626100	11354
335317	214.06	185.00	240.75	Queens	PARK TERRACE CARE CENTER	59 20 VAN DOREN STREET	7185929200	11368
335363	205.03	163.00	263.25	Queens	OZANAM HALL OF QUEENS NURSING HOME INC	42 41 201ST STREET	7184232000	11361
335448	179.19	101.00	252.75	Queens	QUEENS NASSAU REHABILITATION AND NURSING CENTER	520 BEACH 19TH STREET	7184717400	11691
335349	168.13	124.75	223.25	Queens	CLIFFSIDE REHAB & RESIDENTIAL HEALTH CARE CENTER	119 - 19 GRAHAM COURT	7188860700	11354
335310	148.54	133.25	166.75	Queens	FOREST VIEW CENTER FOR REHABILITATION & NURSING	71 20 110TH STREET	7187933200	11375
335093	145.38	98.25	185.50	Queens	PARK NURSING HOME	128 BEACH 115TH STREET	7184746400	11694
335426	142.85	106.00	174.25	Queens	FRANKLIN CENTER FOR REHABILITATION AND NURSING	142 27 FRANKLIN AVENUE	7186703400	11355
335838	133.10	91.00	189.00	Queens	NEW YORK CENTER FOR REHABILITATION & NURSING	26-13 21ST STREET	7186264800	11102
335379	117.90	95.00	163.25	Queens	REGO PARK NURSING HOME	111 26 CORONA AVENUE	7185926400	11368
335799	89.69	77.75	104.50	Queens	UNION PLAZA CARE CENTER	33 23 UNION STREET	7186700700	11354

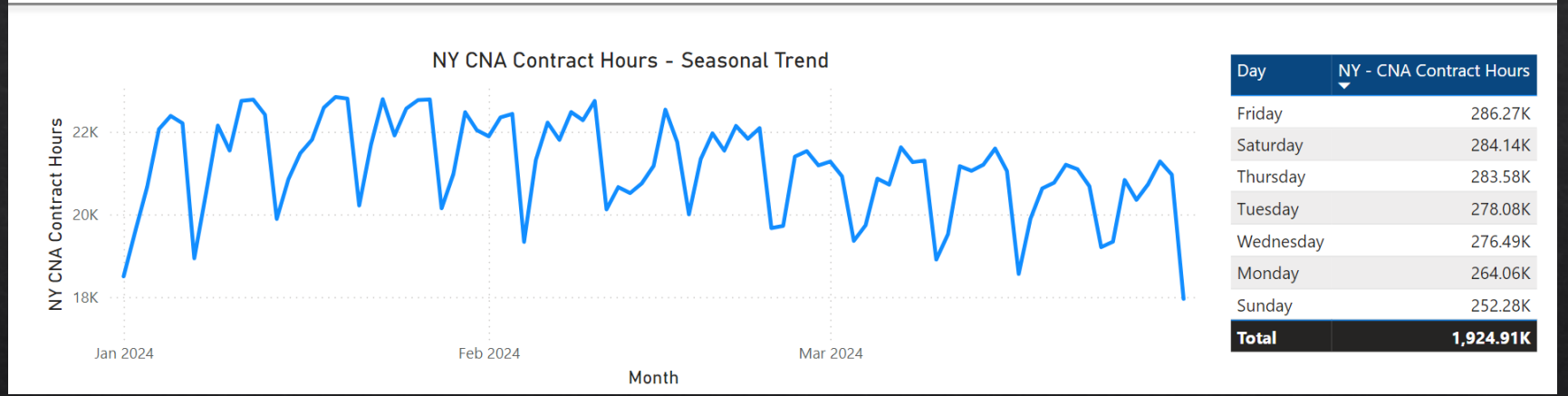
Recommendation is to focus on individual states and roles to locate key counties and providers that offer consistent high demand. Using this provides data driven sales target with Provider Name, Address and Phone Number ready to go!

Bonus Recommendation

- It is hard to wrap this up without noting the Time-Series nature of data, it's seasonality and trend.
- Across the US there is slight dip in total contract hours during midweek around Wednesday.
- However, zoning in on NY - CNA contract hours, surprisingly the dip occurs on Sunday.



Day	Total Contract Hours
Friday	4.6M
Saturday	4.5M
Sunday	4.5M
Monday	4.4M
Thursday	4.3M
Tuesday	4.2M
Wednesday	4.2M
Total	**30.7M**



Day	NY - CNA Contract Hours
Friday	286.27K
Saturday	284.14K
Thursday	283.58K
Tuesday	278.08K
Wednesday	276.49K
Monday	264.06K
Sunday	252.28K
Total	**1,924.91K**

Bonus Recommendation is that we have a dataset well suited for time-series forecasting. This is extremely valuable in staffing estimations as the amount of data is readily available and usually two columns, date and value are enough.

While for purpose of this submission I have not pursued a timeseries forecasting model. I have explored this topic in project.

Conclusion

- Across US with nearly 30.7M Total contract Hours available in Nursing homes, the market is strong for Contract Staffing.
- Over half of these hours were made up by CNA role (56%), followed by LPN(28%) and RN(13%).
- Northeast is an important market region for sales, with Providers in NY, PA, and NJ offering highest contract hours.
- Focusing on individual state, we can locate key counties and providers that consistently offer high demand for sales opportunities.

Future Scope

- The 8 Distinct Contract roles might provide opportunities for cross-training. This allows more flexibility and contract staff to experience opportunities across different roles. The overlap of roles can be explored further.

- During analysis Ownership table was explored. However, due to incomplete data in single file the information was not used.
- Cleaning and compiling ownership data to note Owners that offer high contract demand, can be used by sales to target potential partnerships.

- Thanks to Work date and Contract hours available to us on CMS website for recent quarters. It is feasible to create and test accuracy of a timeseries forecast model. This can be used to estimate contract need in coming weeks and proactively reach out to Providers with percent of need we are able to meet.

Thank You

Thank you for this opportunity and project!
It was a gift and a pleasure to work on this submission.

Looking forward to your feedback!

(The Jupyter notebook and PowerBI dashboards are attached for reference after Part 2)

Part 2: SQL Test

Instructions: Please write SQL queries for each of the following questions. You may assume that all tables follow typical database conventions unless otherwise specified.

Tables:

Assume you have the following tables in your database:

1. **Sales**
    - `sales_id` (INT)
    - `customer_id` (INT)
    - `product_id` (INT)
    - `sale_date` (DATE)
    - `quantity` (INT)
    - `total_amount` (DECIMAL)
2. **Customers**
    - `customer_id` (INT)
    - `customer_name` (VARCHAR)
    - `sales_region` (VARCHAR)
    - `sign_up_date` (DATE)
3. **Products**
    - `product_id` (INT)
    - `product_name` (VARCHAR)SELEC
    - `category` (VARCHAR)
    - `price` (DECIMAL)

Questions:

1. Write a query to return the `customer_name`, `product_name`, and `total_amount` for each sale in the last 30 days.

```
SELECT
c.customer_name
, p.product_name
, SUM(s.total_amount)
FROM Sales s
LEFT JOIN Customers c
ON s.customer_id = c.customer_id
LEFT JOIN Products p
ON s.product_id = p.prodcut_id

WHERE ((s.sale_date <= getdate()) AND (s.sale_date >= DATEADD(d,-
30,GETDATE())))

GROUP BY c.customer_name, p.product_name
```

2. Write a query to find the total revenue generated by each product category in the last year. The output should include the product category and the total revenue for that category.

```
SELECT
p.category
, SUM(s.total_amount)

FROM Products p
LEFT JOIN Sales s
ON p.product_id = s.product_id

WHERE ((s.sale_date <= getdate()) AND (s.sale_date >= DATEADD(m,-
12,GETDATE())))

GROUP BY p.category
```

3. Write a query to return all customers who made purchases in 2023 and are located in the "West" region.

```
SELECT DISTINCT
c.customer_id
, c.customer_name


FROM Sales s
LEFT JOIN Customers c
ON s.customer_id = c.customer_id

WHERE sales_region = 'West'
AND DATEPART(Year, s.sale_date) = 2023
```

4. Write a query to display the total number of sales, total quantity sold, and total revenue for each customer. The result should include the `customer_name`, total sales, total quantity, and total revenue.

```
SELECT
c.customer_name
, COUNT(s.sale_id) AS total_sales
, SUM(s.quantity) AS total_quantity
, SUM(s.total_amount) AS total_revenue

FROM Sales s
LEFT JOIN Customers c
ON s.customer_id = c.customer_id
LEFT JOIN Products p
ON s.product_id = p.prodcut_id

GROUP BY c.customer_name
```

5. Write a query to find the top 3 customers (by total revenue) in the year 2023.

```
SELECT TOP 3 -- Following MS SQL Server Syntax
c.customer_id
, c.customer_name
, SUM(s.total_amount) AS total_revenue

FROM Sales s
LEFT JOIN Customers c
ON s.customer_id = c.customer_id
LEFT JOIN Products p
ON s.product_id = p.prodcut_id

WHERE DATEPART(Year, s.sale_date) = 2023

GROUP BY c.customer_id, c.customer_name
ORDER BY 3 DESC
```

6. Write a query to rank products by their total sales quantity in 2023. The result should include the `product_name`, total quantity sold, and rank.

```
SELECT
p.product_name
, SUM(s.quantity) AS total_quantity_sold
, ROW_NUMBER() OVER(ORDER BY SUM(s.quantity) DESC)  AS rank_

FROM Products p
LEFT JOIN Sales s
ON p.product_id = s.product_id

WHERE DATEPART(Year, s.sale_date) = 2023

GROUP BY p.product_name

ORDER BY 2 desc
```

7. Write a query that categorizes customers into "New" (if they signed up in the last 6 months) or "Existing" based on their `sign_up_date`. Include the `customer_name`, `region`, and category in the result.

```
SELECT DISTINCT
c.customer_name
, c.sales_region
, CASE WHEN ((c.sign_up_date <= getdate()) AND (c.sign_up_date >= DATEADD(m,-
6,GETDATE()))) THEN 'New'
ELSE 'Existing'
END AS customer_category

FROM Customers c
```

8. Write a query to return the month and year along with the total sales for each month for the last 12 months.

```
SELECT
DATEPART(Month, s.sale_date) AS month_
, DATEPART(Year, s.sale_date) AS year_
, SUM(s.quantity) AS total_sales

FROM Sales s

WHERE ((s.sale_date <= getdate()) AND (s.sale_date >= DATEADD(m,-
12,GETDATE())))

GROUP BY DATEPART(Month, s.sale_date)
, DATEPART(Year, s.sale_date)
```

9. Write a query to return the product categories that generated more than $50,000 in revenue during the last 6 months.

```
SELECT
p.category
, SUM(s.total_amount) AS revenue

FROM Products p
LEFT JOIN Sales s
ON p.product_id = s.product_id

WHERE ((s.sale_date <= getdate()) AND (s.sale_date >= DATEADD(m,-
6,GETDATE())))

GROUP BY p.category
HAVING SUM(s.total_amount) >=50000
```

10. Write a query to check for any sales where the `total_amount` doesn't match the expected value (i.e., `quantity` * `price`).

```
with temp AS (
SELECT
s.sales_id -- pulling to be able to refer where the values don't match
, s.total_amount
, (s.quantity * p.price) AS expected_value

FROM Sales s
LEFT JOIN Products p
ON s.product_id = p.product_id
)

SELECT
*
FROM temp t
WHERE t.total_amount != t.expected_value
```

Reference Materials

Attached a pdf copy of Jupyter Notebook and Power BI dashboards for reference

Imports and Data Load

In [1]:
```
# imports
import matplotlib as mpl
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
import pandas as pd
import numpy as np

import os
```

In [2]:
```
pbj_raw = pd.read_csv('raw_data/PBJ_Daily_Nurse_Staffing_Q1_2024.csv' , encoding='ISO-8859-1', dtype={0: 'str'}, low_memory=False)
```

PROVNUM or Medicare Provider Number is also known as CMS Certification Number (CCN)

In [3]:
```
pbj = pbj_raw.copy()
# saving a copy of raw data without touching it further down
```

Data Cleaning and Manipulation

In [4]:
```
pbj.head()
```

Out[4]:

	ï»¿PROVNUM	PROVNAME	CITY	STATE	COUNTY_NAME	COUNTY_FIPS	CY_Qtr	WorkDate	MDScensus	Hrs_RNDON	...	Hrs_LPN_ctr	Hrs_CNA	Hrs_CNA_emp	Hrs_CNA_ctr	Hrs_NAtrn	Hr
0	015009	BURNS NURSING HOME, INC.	RUSSELLVILLE	AL	Franklin	59	2024Q1	20240101	50	8.0	...	0.0	156.34	156.34	0.0	0.0	
1	015009	BURNS NURSING HOME, INC.	RUSSELLVILLE	AL	Franklin	59	2024Q1	20240102	49	8.0	...	0.0	149.40	149.40	0.0	0.0	
2	015009	BURNS NURSING HOME, INC.	RUSSELLVILLE	AL	Franklin	59	2024Q1	20240103	49	8.0	...	0.0	147.15	147.15	0.0	0.0	
3	015009	BURNS NURSING HOME, INC.	RUSSELLVILLE	AL	Franklin	59	2024Q1	20240104	50	8.0	...	0.0	142.21	142.21	0.0	0.0	
4	015009	BURNS NURSING HOME, INC.	RUSSELLVILLE	AL	Franklin	59	2024Q1	20240105	51	8.0	...	0.0	149.40	149.40	0.0	0.0	

5 rows × 33 columns

In [5]:
```
pbj.rename(columns = {'ï»¿PROVNUM':'PROVNUM'}, inplace =True)
```

In [6]:
```
pbj.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1330966 entries, 0 to 1330965
Data columns (total 33 columns):
 #   Column            Non-Null Count    Dtype
---  ------            --------------    -----
 0   PROVNUM           1330966 non-null  object
 1   PROVNAME          1330966 non-null  object
 2   CITY              1330966 non-null  object
 3   STATE             1330966 non-null  object
 4   COUNTY_NAME       1330966 non-null  object
 5   COUNTY_FIPS       1330966 non-null  int64
 6   CY_Qtr            1330966 non-null  object
 7   WorkDate          1330966 non-null  int64
 8   MDScensus         1330966 non-null  int64
 9   Hrs_RNDON         1330966 non-null  float64
 10  Hrs_RNDON_emp     1330966 non-null  float64
 11  Hrs_RNDON_ctr     1330966 non-null  float64
 12  Hrs_RNadmin       1330966 non-null  float64
 13  Hrs_RNadmin_emp   1330966 non-null  float64
 14  Hrs_RNadmin_ctr   1330966 non-null  float64
 15  Hrs_RN            1330966 non-null  float64
 16  Hrs_RN_emp        1330966 non-null  float64
 17  Hrs_RN_ctr        1330966 non-null  float64
 18  Hrs_LPNadmin      1330966 non-null  float64
 19  Hrs_LPNadmin_emp  1330966 non-null  float64
 20  Hrs_LPNadmin_ctr  1330966 non-null  float64
 21  Hrs_LPN           1330966 non-null  float64
 22  Hrs_LPN_emp       1330966 non-null  float64
 23  Hrs_LPN_ctr       1330966 non-null  float64
 24  Hrs_CNA           1330966 non-null  float64
 25  Hrs_CNA_emp       1330966 non-null  float64
 26  Hrs_CNA_ctr       1330966 non-null  float64
 27  Hrs_NAtrn         1330966 non-null  float64
 28  Hrs_NAtrn_emp     1330966 non-null  float64
 29  Hrs_NAtrn_ctr     1330966 non-null  float64
 30  Hrs_MedAide       1330966 non-null  float64
 31  Hrs_MedAide_emp   1330966 non-null  float64
 32  Hrs_MedAide_ctr   1330966 non-null  float64
dtypes: float64(24), int64(3), object(6)
memory usage: 335.1+ MB
```

In [7]:
```
pbj.isnull().sum()
# we have no mussing values
```

Out[7]:
```
PROVNUM             0
PROVNAME            0
CITY                0
STATE               0
COUNTY_NAME         0
COUNTY_FIPS         0
CY_Qtr              0
WorkDate            0
MDScensus           0
Hrs_RNDON           0
Hrs_RNDON_emp       0
Hrs_RNDON_ctr       0
Hrs_RNadmin         0
Hrs_RNadmin_emp     0
Hrs_RNadmin_ctr     0
Hrs_RN              0
Hrs_RN_emp          0
Hrs_RN_ctr          0
Hrs_LPNadmin        0
Hrs_LPNadmin_emp    0
Hrs_LPNadmin_ctr    0
Hrs_LPN             0
Hrs_LPN_emp         0
Hrs_LPN_ctr         0
Hrs_CNA             0
Hrs_CNA_emp         0
Hrs_CNA_ctr         0
Hrs_NAtrn           0
Hrs_NAtrn_emp       0
Hrs_NAtrn_ctr       0
Hrs_MedAide         0
Hrs_MedAide_emp     0
Hrs_MedAide_ctr     0
dtype: int64
```

In [8]:
```
pbj[pbj.duplicated() == True]

# we have no duplicate enteries
```

Out[8]:

PROVNUM	PROVNAME	CITY	STATE	COUNTY_NAME	COUNTY_FIPS	CY_Qtr	WorkDate	MDScensus	Hrs_RNDON	...	Hrs_LPN_ctr	Hrs_CNA	Hrs_CNA_emp	Hrs_CNA_ctr	Hrs_NAtrn	Hrs_NAtrn_emp

0 rows × 33 columns

In [9]:
```
pbj.columns
```

Out[9]:
```
Index(['PROVNUM', 'PROVNAME', 'CITY', 'STATE', 'COUNTY_NAME', 'COUNTY_FIPS',
       'CY_Qtr', 'WorkDate', 'MDScensus', 'Hrs_RNDON', 'Hrs_RNDON_emp',
       'Hrs_RNDON_ctr', 'Hrs_RNadmin', 'Hrs_RNadmin_emp', 'Hrs_RNadmin_ctr',
       'Hrs_RN', 'Hrs_RN_emp', 'Hrs_RN_ctr', 'Hrs_LPNadmin',
       'Hrs_LPNadmin_emp', 'Hrs_LPNadmin_ctr', 'Hrs_LPN', 'Hrs_LPN_emp',
       'Hrs_LPN_ctr', 'Hrs_CNA', 'Hrs_CNA_emp', 'Hrs_CNA_ctr', 'Hrs_NAtrn',
       'Hrs_NAtrn_emp', 'Hrs_NAtrn_ctr', 'Hrs_MedAide', 'Hrs_MedAide_emp',
       'Hrs_MedAide_ctr'],
      dtype='object')
```

In [10]:
```
pbj['Total_Hours'] = pbj[['Hrs_RNDON', 'Hrs_RNadmin', 'Hrs_RN', 'Hrs_LPNadmin', 'Hrs_LPN', 'Hrs_CNA', 'Hrs_NAtrn', 'Hrs_MedAide']].sum(axis=1)
```

In [11]:
```
pbj[[ 'Hrs_RNDON', 'Hrs_RNadmin', 'Hrs_RN', 'Hrs_LPNadmin', 'Hrs_LPN', 'Hrs_CNA', 'Hrs_NAtrn', 'Hrs_MedAide']].head()
```

Out[11]:

	Hrs_RNDON	Hrs_RNadmin	Hrs_RN	Hrs_LPNadmin	Hrs_LPN	Hrs_CNA	Hrs_NAtrn	Hrs_MedAide
0	8.0	8.00	40.07	0.0	18.16	156.34	0.0	0.0
1	8.0	18.24	58.89	0.0	22.96	149.40	0.0	0.0
2	8.0	15.10	55.02	0.0	20.70	147.15	0.0	0.0
3	8.0	14.90	57.13	0.0	12.70	142.21	0.0	0.0
4	8.0	15.47	46.76	0.0	27.44	149.40	0.0	0.0

In [12]:
```
pbj[['Total_Hours']].head()
```

Out[12]:

	Total_Hours
0	230.57
1	257.49
2	245.97
3	234.94
4	247.07

In [13]:
```
pbj['Total_Hours_emp'] = pbj[['Hrs_RNDON_emp', 'Hrs_RNadmin_emp', 'Hrs_RN_emp', 'Hrs_LPNadmin_emp', 'Hrs_LPN_emp', 'Hrs_CNA_emp', 'Hrs_NAtrn_emp', 'Hrs_MedAide_emp']].sum(axis=1)
```

In [14]:
```
pbj['Total_Hours_ctr'] = pbj[['Hrs_RNDON_ctr', 'Hrs_RNadmin_ctr', 'Hrs_RN_ctr', 'Hrs_LPNadmin_ctr', 'Hrs_LPN_ctr', 'Hrs_CNA_ctr', 'Hrs_NAtrn_ctr', 'Hrs_MedAide_ctr']].sum(axis=1)
```

In [15]:
```
pbj.head()
```

Out[15]:

	PROVNUM	PROVNAME	CITY	STATE	COUNTY_NAME	COUNTY_FIPS	CY_Qtr	WorkDate	MDScensus	Hrs_RNDON	...	Hrs_CNA_ctr	Hrs_NAtrn	Hrs_NAtrn_emp	Hrs_NAtrn_ctr	Hrs_MedAid
0	015009	BURNS NURSING HOME, INC.	RUSSELLVILLE	AL	Franklin	59	2024Q1	20240101	50	8.0	...	0.0	0.0	0.0	0.0	0.
1	015009	BURNS NURSING HOME, INC.	RUSSELLVILLE	AL	Franklin	59	2024Q1	20240102	49	8.0	...	0.0	0.0	0.0	0.0	0.
2	015009	BURNS NURSING HOME, INC.	RUSSELLVILLE	AL	Franklin	59	2024Q1	20240103	49	8.0	...	0.0	0.0	0.0	0.0	0.
3	015009	BURNS NURSING HOME, INC.	RUSSELLVILLE	AL	Franklin	59	2024Q1	20240104	50	8.0	...	0.0	0.0	0.0	0.0	0.
4	015009	BURNS NURSING HOME, INC.	RUSSELLVILLE	AL	Franklin	59	2024Q1	20240105	51	8.0	...	0.0	0.0	0.0	0.0	0.

5 rows × 36 columns

In [16]:
```
pbj.STATE.nunique()
```

Out[16]: 52

In [17]:
```
pbj.STATE.unique()

# 50 states + Distict of Columbia and Puerto Rico
```

Out[17]:
```
array(['AL', 'AK', 'AZ', 'AR', 'CA', 'CO', 'CT', 'DE', 'DC', 'FL', 'GA',
       'HI', 'ID', 'IL', 'IN', 'IA', 'KS', 'KY', 'LA', 'ME', 'MD', 'MA',
       'MI', 'MN', 'MS', 'MO', 'MT', 'NE', 'NV', 'NH', 'NJ', 'NM', 'NY',
       'NC', 'ND', 'OH', 'OK', 'OR', 'PA', 'PR', 'RI', 'SC', 'SD', 'TN',
       'TX', 'UT', 'VT', 'VA', 'WA', 'WV', 'WI', 'WY'], dtype=object)
```

In [18]:
```
pbj = pbj[(pbj['STATE'] != 'DC') & (pbj['STATE'] != 'PR')]

# Dropping Distict of Columbia and Puerto Rico from Analysis for now - Goal is to focus on 50 states as per problem statement in Google doc
```

In [19]:
```
pbj.STATE.nunique()
```

Out[19]: 50

In [20]:
```
pbj.PROVNUM.nunique()
```

Out[20]: 14604

Exploring Secondary Table: Provider Information

In [21]:
```
# Loading Secondary Tables to gather more information on Providers

provider_sep = pd.read_csv('raw_data/NH_ProviderInfo_Sep2024.csv' , encoding='ISO-8859-1', dtype={0: 'str','Telephone Number':'str'}, low_memory=False)
provider_mar= pd.read_csv('raw_data/NH_ProviderInfo_Mar2024.csv' , encoding='ISO-8859-1', dtype={0: 'str','Telephone Number':'str'}, low_memory=False)

# Provider details for certain Providers from Q1-2024 is missing in latest September file, checking Q1 March file for those Providers covers the gap
```

In [22]:
```
provider_sep = provider_sep[['CMS Certification Number (CCN)', 'Provider Address', 'ZIP Code', 'Telephone Number','Provider Type', 'Provider Resides in Hospital','Provider Changed Owne
provider_mar = provider_mar[['CMS Certification Number (CCN)', 'Provider Address', 'ZIP Code', 'Telephone Number','Provider Type', 'Provider Resides in Hospital','Provider Changed Owne
```

In [23]:
```
provider_sep.rename(columns = {'CMS Certification Number (CCN)':'PROVNUM'}, inplace =True)
provider_mar.rename(columns = {'CMS Certification Number (CCN)':'PROVNUM'}, inplace =True)
```

In [24]:
```
# capturing all Provider Numbers present in latest file
latest_PROVNUM = provider_sep.PROVNUM.to_list()
```

In [25]:
```
# capturing provider numbers present in March file but not in latest September file
provider_mar = provider_mar[~provider_mar['PROVNUM'].isin(latest_PROVNUM)]
```

In [26]:
```
# combineing latest file with PROVNUMs only present in March
provider =pd.concat([provider_mar, provider_sep],ignore_index=True)
```

In [27]:
```
provider.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14900 entries, 0 to 14899
Data columns (total 8 columns):
 #   Column                                      Non-Null Count  Dtype
---  ------                                      --------------  -----
 0   PROVNUM                                     14900 non-null  object
 1   Provider Address                            14900 non-null  object
 2   ZIP Code                                    14900 non-null  int64
 3   Telephone Number                            14900 non-null  object
 4   Provider Type                               14900 non-null  object
 5   Provider Resides in Hospital                14900 non-null  object
 6   Provider Changed Ownership in Last 12 Months  14900 non-null  object
 7   Ownership Type                              14900 non-null  object
dtypes: int64(1), object(7)
memory usage: 931.4+ KB
```

In [28]:
```
pbj.shape
```

Out[28]: (1328964, 36)

In [29]:
```
# Joining provider info to PBJ Q1-2024 file
df = pd.merge(pbj, provider, how='left', left_on=['PROVNUM'], right_on=['PROVNUM'])
```

In [30]:
```
df.shape
```

Out[30]: (1328964, 43)

In [31]:
```
df.isnull().sum()
```

Out[31]:
```
PROVNUM                                  0
PROVNAME                                 0
CITY                                     0
STATE                                    0
COUNTY_NAME                              0
COUNTY_FIPS                              0
CY_Qtr                                   0
WorkDate                                 0
MDScensus                                0
Hrs_RNDON                                0
Hrs_RNDON_emp                            0
Hrs_RNDON_ctr                            0
Hrs_RNadmin                              0
Hrs_RNadmin_emp                          0
Hrs_RNadmin_ctr                          0
Hrs_RN                                   0
Hrs_RN_emp                               0
Hrs_RN_ctr                               0
Hrs_LPNadmin                             0
Hrs_LPNadmin_emp                         0
Hrs_LPNadmin_ctr                         0
Hrs_LPN                                  0
Hrs_LPN_emp                              0
Hrs_LPN_ctr                              0
Hrs_CNA                                  0
Hrs_CNA_emp                              0
Hrs_CNA_ctr                              0
Hrs_NAtrn                                0
Hrs_NAtrn_emp                            0
Hrs_NAtrn_ctr                            0
Hrs_MedAide                              0
Hrs_MedAide_emp                          0
Hrs_MedAide_ctr                          0
Total_Hours                              0
Total_Hours_emp                          0
Total_Hours_ctr                          0
Provider Address                         0
ZIP Code                                 0
Telephone Number                         0
Provider Type                            0
Provider Resides in Hospital             0
Provider Changed Ownership in Last 12 Months    0
Ownership Type                           0
dtype: int64
```

In [32]:
```
df['PROVNUM'][df['ZIP Code'].isnull()].unique()
```

Out[32]:
```
array([], dtype=object)
```

In [33]:
```
df.head()
```

Out[33]:

	PROVNUM	PROVNAME	CITY	STATE	COUNTY_NAME	COUNTY_FIPS	CY_Qtr	WorkDate	MDScensus	Hrs_RNDON	...	Total_Hours	Total_Hours_emp	Total_Hours_ctr	Provider Address	ZIP Code	Te
0	015009	BURNS NURSING HOME, INC.	RUSSELLVILLE	AL	Franklin	59	2024Q1	20240101	50	8.0	...	230.57	230.57	0.0	701 MONROE STREET NW	35653	256
1	015009	BURNS NURSING HOME, INC.	RUSSELLVILLE	AL	Franklin	59	2024Q1	20240102	49	8.0	...	257.49	257.49	0.0	701 MONROE STREET NW	35653	256
2	015009	BURNS NURSING HOME, INC.	RUSSELLVILLE	AL	Franklin	59	2024Q1	20240103	49	8.0	...	245.97	245.97	0.0	701 MONROE STREET NW	35653	256
3	015009	BURNS NURSING HOME, INC.	RUSSELLVILLE	AL	Franklin	59	2024Q1	20240104	50	8.0	...	234.94	234.94	0.0	701 MONROE STREET NW	35653	256
4	015009	BURNS NURSING HOME, INC.	RUSSELLVILLE	AL	Franklin	59	2024Q1	20240105	51	8.0	...	247.07	247.07	0.0	701 MONROE STREET NW	35653	256

5 rows × 43 columns

Exploring Secondary Table: Ownership

Note Ownership table not used at this time:

After exploration, we noticed information sky rockets as a Provider Number can be linked to multiple Owner Names and Types

For example, we see that Provider Number 745038 is assoiacted with two Owners since 2023-06-06.

Digging deeper into CMS.gov website we notice that there percentages are listed as 'NO PERCENTAGE PROVIDED' and 'NOT APPLICABLE'. There is another provider with Association Date 2022-12-01, with Owenrship Percentage as 'NO PERCENTAGE PROVIDED'

The Goal of exploring Ownership was to find common ownership across providers and note if any patterns jump out to highlight Key Owners for building partnerships.

Given the vaguness of information and to narrow scope of this submission, we are only adding Provider Information as secondary Table.

Ownership table is not used for recommendations below, but based on further study and data cleaning it might provide value in the future.

In [34]:
```
# Loading Ownership Table

owner =  pd.read_csv('raw_data/NH_Ownership_Sep2024.csv' , encoding='ISO-8859-1', dtype={0: 'str'}, low_memory=False)
```

In [35]:
```
owner.rename(columns = {'CMS Certification Number (CCN)':'PROVNUM'}, inplace =True)
```

In [36]:
```
owner = owner[['PROVNUM', 'Owner Type', 'Owner Name','Association Date']]
```

In [37]:
```
owner.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 145367 entries, 0 to 145366
Data columns (total 4 columns):
 #   Column            Non-Null Count   Dtype 
---  ------            --------------   ----- 
 0   PROVNUM           145367 non-null  object
 1   Owner Type        144827 non-null  object
 2   Owner Name        144827 non-null  object
 3   Association Date  144827 non-null  object
dtypes: object(4)
memory usage: 4.4+ MB
```

In [38]:
```
# convert Association Date to Date field and catch owner info mased on latest date
owner['Association Date']=owner['Association Date'].str.replace('since ','')
```

In [39]:
```
owner[['PROVNUM','Association Date']]
```

Out[39]:

	PROVNUM	Association Date
0	015009	01/25/2012
1	015009	09/01/1969
2	015009	09/07/1969
3	015009	12/26/2019
4	015009	12/26/2019
...	...	...
145362	745040	09/01/2023
145363	745040	06/01/2023
145364	745040	09/01/2023
145365	745040	04/08/2024
145366	745049	NaN

145367 rows × 2 columns

In [40]:
```
owner['Association Date'] = pd.to_datetime(owner['Association Date'], errors='coerce')
```

In [41]:
```
owner[['PROVNUM','Association Date']]
```

Out[41]:

	PROVNUM	Association Date
0	015009	2012-01-25
1	015009	1969-09-01
2	015009	1969-09-07
3	015009	2019-12-26
4	015009	2019-12-26
...	...	...
145362	745040	2023-09-01
145363	745040	2023-06-01
145364	745040	2023-09-01
145365	745040	2024-04-08
145366	745049	NaT

145367 rows × 2 columns

In [42]:
```
owner.head()
```

Out[42]:

	PROVNUM	Owner Type	Owner Name	Association Date
0	015009	Individual	DEARMAN, CAMERON	2012-01-25
1	015009	Individual	DEARMAN, MARTHA	1969-09-01
2	015009	Individual	DEARMAN, MARTHA	1969-09-07
3	015009	Individual	DEARMAN, CAMERON	2019-12-26
4	015009	Individual	DEARMAN, MARK	2019-12-26

In [43]:
```
owner.groupby(['PROVNUM'])['Association Date'].max().reset_index()
```

Out[43]:

	PROVNUM	Association Date
0	015009	2019-12-26
1	015010	2018-11-16
2	015012	2021-12-01
3	015014	2017-05-08
4	015015	2002-09-13
...	...	...
14812	745022	2022-12-30
14813	745038	2023-06-06
14814	745039	2023-02-01
14815	745040	2024-04-08
14816	745049	NaT

14817 rows × 2 columns

In [44]:
```
temp = owner.groupby(['PROVNUM'])['Association Date'].max().reset_index()
```

In [45]:
```
owner_latest = pd.merge(temp, owner, how='left', left_on = ['PROVNUM','Association Date'], right_on= ['PROVNUM','Association Date'])
```

In [46]:
```
owner_latest
```

Out[46]:

	PROVNUM	Association Date	Owner Type	Owner Name
0	015009	2019-12-26	Individual	DEARMAN, CAMERON
1	015009	2019-12-26	Individual	DEARMAN, MARK
2	015010	2018-11-16	Individual	CHAPMAN, ARCHIE
3	015010	2018-11-16	Organization	PRIME MANAGEMENT, LLC
4	015010	2018-11-16	Individual	CHAPMAN, ARCHIE
...	...	...	...	...
47596	745039	2023-02-01	Individual	HAWLEY, GRACEN
47597	745039	2023-02-01	Individual	HAWLEY, RUSSELL
47598	745039	2023-02-01	Individual	HAWLEY, GRACEN
47599	745040	2024-04-08	Individual	ROBINSON, MYRNA
47600	745049	NaT	NaN	NaN

47601 rows × 4 columns

In [47]:
```
pbj.shape
```

Out[47]:
```
(1328964, 36)
```

In [48]:
```
df_2 = pd.merge(pbj, owner_latest, how='left', left_on=['PROVNUM'], right_on=['PROVNUM'],)
```

In [49]:
```
df_2.shape
# the number of enteris sky rockets as a Provider Number can be linked to multiple Owner Names and Types
```

Out[49]:
```
(4280367, 39)
```

In [50]:
```
col_to_check = 'Owner Type'
new = df_2.groupby(['PROVNUM'])[col_to_check].nunique()
```

In [51]:
```
new.sort_values(ascending=False)
```

Out[51]:
```
PROVNUM
745038    2
145887    2
365554    2
455968    2
145866    2
         ..
455522    0
455510    0
185384    0
455493    0
056351    0
Name: Owner Type, Length: 14604, dtype: int64
```

In [52]:
```
df_2[['PROVNUM', 'Owner Type', 'Owner Name','Association Date']][df_2['PROVNUM'] == '745038'].head()
```

Out[52]:

	PROVNUM	Owner Type	Owner Name	Association Date
4280094	745038	Individual	DOUGLAS, LLOYD	2023-06-06
4280095	745038	Organization	CONQUEST HEALTHCARE MANAGEMENT LLC	2023-06-06
4280096	745038	Individual	DOUGLAS, LLOYD	2023-06-06
4280097	745038	Individual	DOUGLAS, LLOYD	2023-06-06
4280098	745038	Organization	CONQUEST HEALTHCARE MANAGEMENT LLC	2023-06-06

We see that Provider Number 745038 is assoiacted with two Owners since 2023-06-06.

Digging deeper into CMS.gov website we notice that there percentages are listed as 'NO PERCENTAGE PROVIDED' and 'NOT APPLICABLE'. There is another provider with Association Date 2022-12-01, with Owenrship Percentage as 'NO PERCENTAGE PROVIDED'

The Goal of exploring Ownership was to find common ownership across providers and note if there are any patterns jump out to highlight Key Owners.

Given the vaguness of information and to narrow scope of this submission, we are only adding Provider Information as secondary Table.

Ownership table is not used for further analysis below, but based on further study and data cleaning might provide potential value in future.

Exploratory Data Analysis

In [53]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1328964 entries, 0 to 1328963
Data columns (total 43 columns):
 #   Column                                       Non-Null Count    Dtype  
---  ------                                       --------------    -----  
 0   PROVNUM                                      1328964 non-null  object 
 1   PROVNAME                                     1328964 non-null  object 
 2   CITY                                         1328964 non-null  object 
 3   STATE                                        1328964 non-null  object 
 4   COUNTY_NAME                                  1328964 non-null  object 
 5   COUNTY_FIPS                                  1328964 non-null  int64  
 6   CY_Qtr                                       1328964 non-null  object 
 7   WorkDate                                     1328964 non-null  int64  
 8   MDScensus                                    1328964 non-null  int64  
 9   Hrs_RNDON                                    1328964 non-null  float64
 10  Hrs_RNDON_emp                                1328964 non-null  float64
 11  Hrs_RNDON_ctr                                1328964 non-null  float64
 12  Hrs_RNadmin                                  1328964 non-null  float64
 13  Hrs_RNadmin_emp                              1328964 non-null  float64
 14  Hrs_RNadmin_ctr                              1328964 non-null  float64
 15  Hrs_RN                                       1328964 non-null  float64
 16  Hrs_RN_emp                                   1328964 non-null  float64
 17  Hrs_RN_ctr                                   1328964 non-null  float64
 18  Hrs_LPNadmin                                 1328964 non-null  float64
 19  Hrs_LPNadmin_emp                             1328964 non-null  float64
 20  Hrs_LPNadmin_ctr                             1328964 non-null  float64
 21  Hrs_LPN                                      1328964 non-null  float64
 22  Hrs_LPN_emp                                  1328964 non-null  float64
 23  Hrs_LPN_ctr                                  1328964 non-null  float64
 24  Hrs_CNA                                      1328964 non-null  float64
 25  Hrs_CNA_emp                                  1328964 non-null  float64
 26  Hrs_CNA_ctr                                  1328964 non-null  float64
 27  Hrs_NAtrn                                    1328964 non-null  float64
 28  Hrs_NAtrn_emp                                1328964 non-null  float64
 29  Hrs_NAtrn_ctr                                1328964 non-null  float64
 30  Hrs_MedAide                                  1328964 non-null  float64
 31  Hrs_MedAide_emp                              1328964 non-null  float64
 32  Hrs_MedAide_ctr                              1328964 non-null  float64
 33  Total_Hours                                  1328964 non-null  float64
 34  Total_Hours_emp                              1328964 non-null  float64
 35  Total_Hours_ctr                              1328964 non-null  float64
 36  Provider Address                             1328964 non-null  object 
 37  ZIP Code                                     1328964 non-null  int64  
 38  Telephone Number                             1328964 non-null  object 
 39  Provider Type                                1328964 non-null  object 
 40  Provider Resides in Hospital                 1328964 non-null  object 
 41  Provider Changed Ownership in Last 12 Months  1328964 non-null  object 
 42  Ownership Type                               1328964 non-null  object 
dtypes: float64(27), int64(4), object(12)
memory usage: 436.0+ MB
```

In [54]:
```
df['WorkDate'] = pd.to_datetime(df['WorkDate'].astype(str), format='%Y%m%d')
df['COUNTY_FIPS'] = df['COUNTY_FIPS'].astype('str')
df['ZIP Code'] = df['ZIP Code'].astype('str')
```

In [55]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1328964 entries, 0 to 1328963
Data columns (total 43 columns):
 #   Column                                      Non-Null Count    Dtype
---  ------                                      --------------    -----
 0   PROVNUM                                     1328964 non-null  object
 1   PROVNAME                                    1328964 non-null  object
 2   CITY                                        1328964 non-null  object
 3   STATE                                       1328964 non-null  object
 4   COUNTY_NAME                                 1328964 non-null  object
 5   COUNTY_FIPS                                 1328964 non-null  object
 6   CY_Qtr                                      1328964 non-null  object
 7   WorkDate                                    1328964 non-null  datetime64[ns]
 8   MDScensus                                   1328964 non-null  int64
 9   Hrs_RNDON                                   1328964 non-null  float64
 10  Hrs_RNDON_emp                               1328964 non-null  float64
 11  Hrs_RNDON_ctr                               1328964 non-null  float64
 12  Hrs_RNadmin                                 1328964 non-null  float64
 13  Hrs_RNadmin_emp                             1328964 non-null  float64
 14  Hrs_RNadmin_ctr                             1328964 non-null  float64
 15  Hrs_RN                                      1328964 non-null  float64
 16  Hrs_RN_emp                                  1328964 non-null  float64
 17  Hrs_RN_ctr                                  1328964 non-null  float64
 18  Hrs_LPNadmin                                1328964 non-null  float64
 19  Hrs_LPNadmin_emp                            1328964 non-null  float64
 20  Hrs_LPNadmin_ctr                            1328964 non-null  float64
 21  Hrs_LPN                                     1328964 non-null  float64
 22  Hrs_LPN_emp                                 1328964 non-null  float64
 23  Hrs_LPN_ctr                                 1328964 non-null  float64
 24  Hrs_CNA                                     1328964 non-null  float64
 25  Hrs_CNA_emp                                 1328964 non-null  float64
 26  Hrs_CNA_ctr                                 1328964 non-null  float64
 27  Hrs_NAtrn                                   1328964 non-null  float64
 28  Hrs_NAtrn_emp                               1328964 non-null  float64
 29  Hrs_NAtrn_ctr                               1328964 non-null  float64
 30  Hrs_MedAide                                 1328964 non-null  float64
 31  Hrs_MedAide_emp                             1328964 non-null  float64
 32  Hrs_MedAide_ctr                             1328964 non-null  float64
 33  Total_Hours                                 1328964 non-null  float64
 34  Total_Hours_emp                             1328964 non-null  float64
 35  Total_Hours_ctr                             1328964 non-null  float64
 36  Provider Address                            1328964 non-null  object
 37  ZIP Code                                    1328964 non-null  object
 38  Telephone Number                            1328964 non-null  object
 39  Provider Type                               1328964 non-null  object
 40  Provider Resides in Hospital                1328964 non-null  object
 41  Provider Changed Ownership in Last 12 Months  1328964 non-null  object
 42  Ownership Type                              1328964 non-null  object
dtypes: datetime64[ns](1), float64(27), int64(1), object(14)
memory usage: 436.0+ MB
```

In [56]:
```
df['day_of_week']=df['WorkDate'].dt.day_name()
```

In [57]:
```
#conditions
conditions = [
    (df['day_of_week'] == 'Saturday'),
    (df['day_of_week'] == 'Sunday')
    ]

#values
values = [1.0,1.0]

# create a new indicator column
df['is_weekend'] = np.select(conditions, values,0)
```

In [58]:
```
df.head()
```

Out[58]:

	PROVNUM	PROVNAME	CITY	STATE	COUNTY_NAME	COUNTY_FIPS	CY_Qtr	WorkDate	MDScensus	Hrs_RNDON	...	Total_Hours_ctr	Provider Address	ZIP Code	Telephone Number	Provider Type	Provider Resides in Hospital	O i
0	015009	BURNS NURSING HOME, INC.	RUSSELLVILLE	AL	Franklin	59	2024Q1	2024-01-01	50	8.0	...	0.0	701 MONROE STREET NW	35653	2563324110	Medicare and Medicaid	N	
1	015009	BURNS NURSING HOME, INC.	RUSSELLVILLE	AL	Franklin	59	2024Q1	2024-01-02	49	8.0	...	0.0	701 MONROE STREET NW	35653	2563324110	Medicare and Medicaid	N	
2	015009	BURNS NURSING HOME, INC.	RUSSELLVILLE	AL	Franklin	59	2024Q1	2024-01-03	49	8.0	...	0.0	701 MONROE STREET NW	35653	2563324110	Medicare and Medicaid	N	
3	015009	BURNS NURSING HOME, INC.	RUSSELLVILLE	AL	Franklin	59	2024Q1	2024-01-04	50	8.0	...	0.0	701 MONROE STREET NW	35653	2563324110	Medicare and Medicaid	N	
4	015009	BURNS NURSING HOME, INC.	RUSSELLVILLE	AL	Franklin	59	2024Q1	2024-01-05	51	8.0	...	0.0	701 MONROE STREET NW	35653	2563324110	Medicare and Medicaid	N	

5 rows × 45 columns

At this Point we have clean and ready to plug data with day_of_week, weekend check and Provider Information such as zip code and address included

For building a dashboard I will take this data over the raw data and plug it into a data visualization tool such as Tableau or Pwer BI

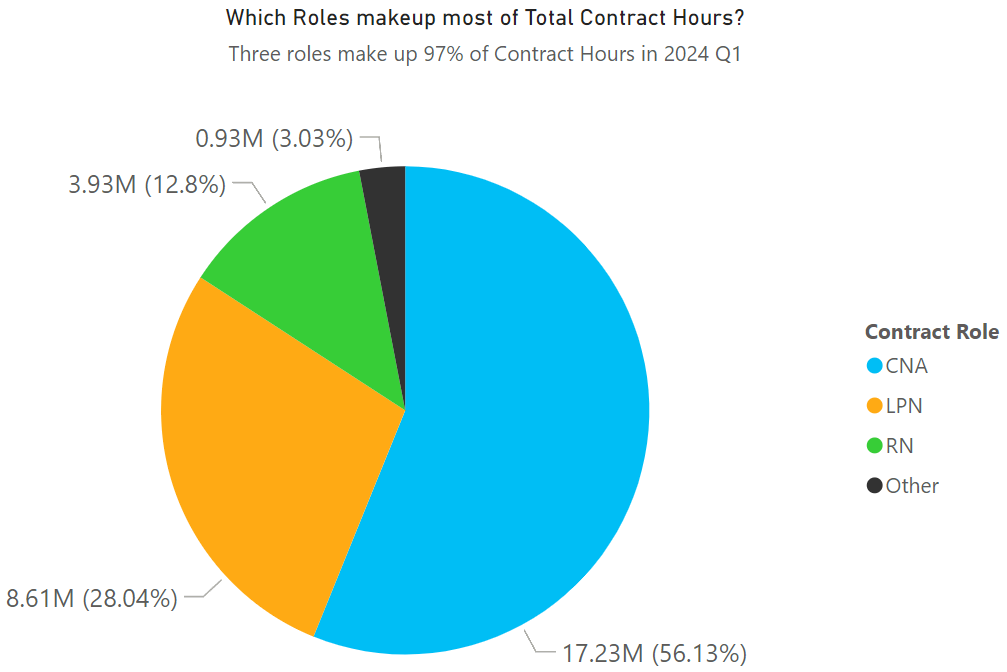
Here for the sake of continuity I am moving ahead in Jupyter Notebook. I will be presenting findings in pdf format and if time allows look to build a basic forecast model so that we can predict number of contract hours/ market for next week

In [59]:
```
# save the clean data as a csv file
datapath = './clean_Data'
if not os.path.exists(datapath):
```

```
    os.mkdir(datapath)
datapath_pbj_data = os.path.join(datapath, 'pbj_clean.csv')
if not os.path.exists(datapath_pbj_data):
    df.to_csv(datapath_pbj_data, index=False)
```

Recommendation 1

Focusing on Contract Roles, 3 roles RN, LPN, and CNA make up 97% of total contract hours and can be considered key line of businesses.



Contract Role	Total Contract Hours	Total Unique Providers to offer this Role	Average Contract Hrs per Provider
CNA	17,231,892.04	6865	2510.11
LPN	8,608,154.24	6964	1236.09
RN	3,929,326.87	5917	664.07
Other	930,360.33	3236	287.50

In [60]:
```
df.rename(columns={'Total_Hours_ctr':'Total_contract_hours','Hrs_RN_ctr':'RN_contract_hours', 'Hrs_LPN_ctr':'LPN_contract_hours', 'Hrs_CNA_ctr':'CNA_contract_hours'}, inplace=True)
```

In [61]:
```
other_contracts = ['Hrs_RNDON_ctr', 'Hrs_RNadmin_ctr', 'Hrs_LPNadmin_ctr', 'Hrs_NAtrn_ctr', 'Hrs_MedAide_ctr']

df['Other_contract_hours']=df[other_contracts].sum(axis=1)
```

In [62]:
```
# Example of cases with contract hours outside of 3 main roles
df[df['Other_contract_hours']>0].head().T
```

Out[62]:

	1117	1120	1121	1122	1123
PROVNUM	015032	015032	015032	015032	015032
PROVNAME	DIVERSICARE OF FOLEY	DIVERSICARE OF FOLEY	DIVERSICARE OF FOLEY	DIVERSICARE OF FOLEY	DIVERSICARE OF FOLEY
CITY	FOLEY	FOLEY	FOLEY	FOLEY	FOLEY
STATE	AL	AL	AL	AL	AL
COUNTY_NAME	Baldwin	Baldwin	Baldwin	Baldwin	Baldwin
COUNTY_FIPS	3	3	3	3	3
CY_Qtr	2024Q1	2024Q1	2024Q1	2024Q1	2024Q1
WorkDate	2024-01-26 00:00:00	2024-01-29 00:00:00	2024-01-30 00:00:00	2024-01-31 00:00:00	2024-02-01 00:00:00
MDScensus	129	130	132	132	128
Hrs_RNDON	10.25	0.0	0.0	0.0	0.0
Hrs_RNDON_emp	0.0	0.0	0.0	0.0	0.0
Hrs_RNDON_ctr	10.25	0.0	0.0	0.0	0.0
Hrs_RNadmin	23.82	39.09	40.95	39.75	36.85
Hrs_RNadmin_emp	23.82	31.09	32.95	31.75	28.85
Hrs_RNadmin_ctr	0.0	8.0	8.0	8.0	8.0
Hrs_RN	40.96	34.64	40.63	40.48	32.86
Hrs_RN_emp	40.96	34.64	40.63	40.48	32.86
RN_contract_hours	0.0	0.0	0.0	0.0	0.0
Hrs_LPNadmin	0.0	0.0	0.0	0.0	0.0
Hrs_LPNadmin_emp	0.0	0.0	0.0	0.0	0.0
Hrs_LPNadmin_ctr	0.0	0.0	0.0	0.0	0.0
Hrs_LPN	81.68	92.16	93.53	104.16	114.23
Hrs_LPN_emp	81.68	92.16	93.53	104.16	114.23
LPN_contract_hours	0.0	0.0	0.0	0.0	0.0
Hrs_CNA	266.96	250.92	261.12	249.02	273.81
Hrs_CNA_emp	266.96	250.92	261.12	249.02	273.81
CNA_contract_hours	0.0	0.0	0.0	0.0	0.0
Hrs_NAtrn	0.0	0.0	0.0	0.0	0.0
Hrs_NAtrn_emp	0.0	0.0	0.0	0.0	0.0
Hrs_NAtrn_ctr	0.0	0.0	0.0	0.0	0.0
Hrs_MedAide	29.2	23.32	18.88	21.77	3.07
Hrs_MedAide_emp	29.2	23.32	18.88	21.77	3.07
Hrs_MedAide_ctr	0.0	0.0	0.0	0.0	0.0
Total_Hours	452.87	440.13	455.11	455.18	460.82
Total_Hours_emp	442.62	432.13	447.11	447.18	452.82
Total_contract_hours	10.25	8.0	8.0	8.0	8.0
Provider Address	1701 NORTH ALSTON STREET	1701 NORTH ALSTON STREET	1701 NORTH ALSTON STREET	1701 NORTH ALSTON STREET	1701 NORTH ALSTON STREET
ZIP Code	36535	36535	36535	36535	36535
Telephone Number	2519432781	2519432781	2519432781	2519432781	2519432781
Provider Type	Medicare and Medicaid	Medicare and Medicaid	Medicare and Medicaid	Medicare and Medicaid	Medicare and Medicaid
Provider Resides in Hospital	N	N	N	N	N
Provider Changed Ownership in Last 12 Months	N	N	N	N	N
Ownership Type	For profit - Corporation	For profit - Corporation	For profit - Corporation	For profit - Corporation	For profit - Corporation
day_of_week	Friday	Monday	Tuesday	Wednesday	Thursday
is_weekend	0.0	0.0	0.0	0.0	0.0
Other_contract_hours	10.25	8.0	8.0	8.0	8.0

In [63]:
```
df_lob = df[['PROVNUM','RN_contract_hours','LPN_contract_hours', 'CNA_contract_hours','Other_contract_hours']]
```

In [64]:
```
# [df_lob.RN_contract_hours.sum(),df_lob.LPN_contract_hours.sum(),df_lob.CNA_contract_hours.sum(),df_lob.Other_contract_hours.sum()]
```

In [65]:
```
# [df_lob[df_lob['RN_contract_hours']>0].PROVNUM.nunique(),df_lob[df_lob['LPN_contract_hours']>0].PROVNUM.nunique(),df_lob[df_lob['CNA_contract_hours']>0].PROVNUM.nunique(),df_lob[df_l
```

In [66]:
```
df_lob_summary = pd.DataFrame({'Contract Role' : ['RN','LPN','CNA','Other']
                    ,'Total Hours':[df_lob.RN_contract_hours.sum(),df_lob.LPN_contract_hours.sum(),df_lob.CNA_contract_hours.sum(),df_lob.Other_contract_hours.sum()]
                    ,'Total Providers':[df_lob[df_lob['RN_contract_hours']>0].PROVNUM.nunique(),df_lob[df_lob['LPN_contract_hours']>0].PROVNUM.nunique(),df_lob[df_lob['CNA_contract_hou
                   )
df_lob_summary['Average Hrs per Provider'] = df_lob_summary['Total Hours']/df_lob_summary['Total Providers']
df_lob_summary = df_lob_summary.sort_values(by='Average Hrs per Provider', ascending=False)


df_lob_summary
```

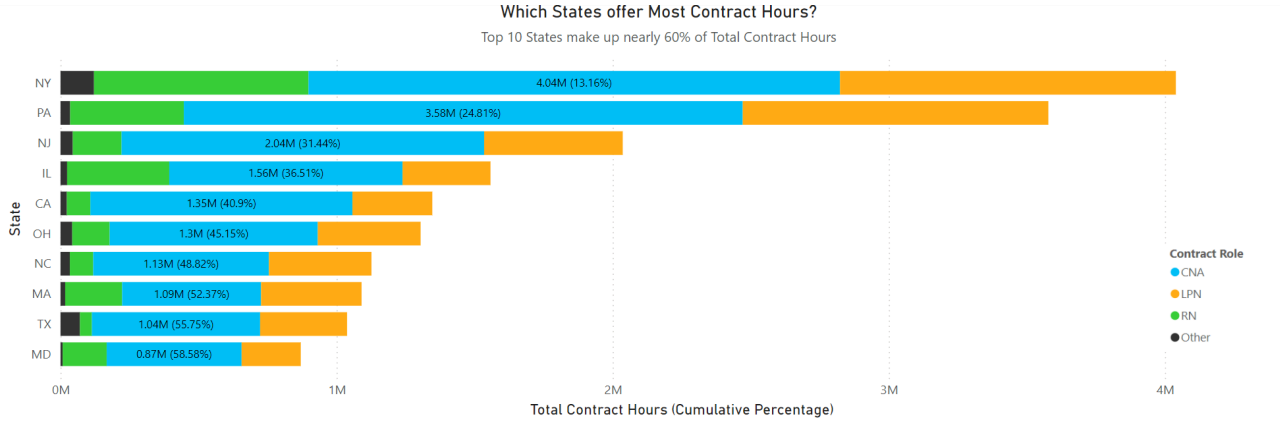
Out[66]:

	Contract Role	Total Hours	Total Providers	Average Hrs per Provider
2	CNA	17231892.04	6865	2510.108090
1	LPN	8608154.24	6964	1236.093372
0	RN	3929326.87	5917	664.074171
3	Other	930360.33	3236	287.503192

In [67]:
```
# df_lob_summary['label'] = ((df_lob_summary['Total Hours']/1000000).round(2).astype(str))+"M ("+(((df_lob_summary['Total Hours']/df_lob_summary['Total Hours'].sum())*100).round(2).ast
```

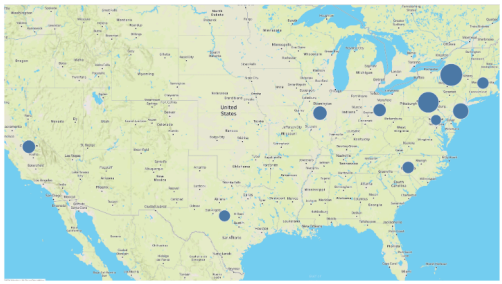
In [68]:
```
# save the clean data as a csv file
datapath = './clean_Data'
if not os.path.exists(datapath):
    os.mkdir(datapath)
datapath_pbj_data = os.path.join(datapath, 'df_lob_summary.csv')
if not os.path.exists(datapath_pbj_data):
    df_lob_summary.to_csv(datapath_pbj_data, index=False)
```

Recommendation 2



Most of Top 10 States are located in NE Region

(Size of Bubble indicates total Contract Hours)



NY, PA, and NJ also offer high contract hours per Provider

State	Total Contract Hours	Total Providers	Total contract Hours per Provider
NY	4,039,449.99	601	6,721.21
PA	3,577,744.31	665	5,380.07
NJ	2,036,180.84	345	5,901.97
IL	1,557,327.83	675	2,307.15
CA	1,346,552.77	1140	1,181.19
OH	1,304,113.78	918	1,420.60
NC	1,126,111.32	410	2,746.61
MA	1,090,336.45	342	3,188.12
TX	1,037,825.88	1164	891.60
MD	870,030.27	220	3,954.68

In [69]:
```
# df.columns.to_list()
```

In [70]:
```
df_state = df[['PROVNUM','STATE','COUNTY_NAME','RN_contract_hours','LPN_contract_hours', 'CNA_contract_hours','Other_contract_hours','Total_contract_hours','ZIP Code']]
```

In [71]:
```
df_state
```

Out[71]:

	PROVNUM	STATE	COUNTY_NAME	RN_contract_hours	LPN_contract_hours	CNA_contract_hours	Other_contract_hours	Total_contract_hours	ZIP Code
0	015009	AL	Franklin	0.0	0.0	0.0	0.0	0.0	35653
1	015009	AL	Franklin	0.0	0.0	0.0	0.0	0.0	35653
2	015009	AL	Franklin	0.0	0.0	0.0	0.0	0.0	35653
3	015009	AL	Franklin	0.0	0.0	0.0	0.0	0.0	35653
4	015009	AL	Franklin	0.0	0.0	0.0	0.0	0.0	35653
...	...	...	...	...	...	...	...	...	...
1328959	745038	TX	El Paso	0.0	0.0	0.0	0.0	0.0	79938
1328960	745038	TX	El Paso	0.0	0.0	0.0	0.0	0.0	79938
1328961	745038	TX	El Paso	0.0	0.0	0.0	0.0	0.0	79938
1328962	745038	TX	El Paso	0.0	0.0	0.0	0.0	0.0	79938
1328963	745038	TX	El Paso	0.0	0.0	0.0	0.0	0.0	79938

1328964 rows × 9 columns

In [72]:
```
df_state['STATE'].nunique()
```

Out[72]: 50

In [73]:
```
df_state['COUNTY_NAME'].nunique()
```

Out[73]: 1667

In [74]:
```
df_state['ZIP Code'].nunique()
```

Out[74]: 8871

In [75]:
```
df_state_summary = df_state.groupby('STATE')[['RN_contract_hours','LPN_contract_hours', 'CNA_contract_hours','Other_contract_hours','Total_contract_hours']].sum()
df_state_summary = df_state_summary.sort_values(by='Total_contract_hours', ascending=False).reset_index()
df_state_summary['Total Hours Percentage'] = ((df_state_summary['Total_contract_hours']/df_state_summary['Total_contract_hours'].sum())*100).round(2)



df_state_summary.head()
```

Out[75]:

	STATE	RN_contract_hours	LPN_contract_hours	CNA_contract_hours	Other_contract_hours	Total_contract_hours	Total Hours Percentage
0	NY	776877.65	1216464.72	1924905.03	121202.59	4039449.99	13.16
1	PA	411631.05	1107096.05	2024098.00	34919.21	3577744.31	11.65
2	NJ	176665.18	502039.89	1313177.45	44298.32	2036180.84	6.63
3	IL	369883.16	318700.45	844819.19	23925.03	1557327.83	5.07
4	CA	85274.23	289389.01	949042.36	22847.17	1346552.77	4.39

In [76]:
```
# df_state.groupby('STATE').PROVNUM.nunique()
```

In [77]:
```
#  pd.DataFrame(df_state.groupby('STATE').PROVNUM.nunique())
# df_state_summary['Total Unique Providers to offer RN role']=df_state[df_state['RN_contract_hours']>0].groupby('STATE').PROVNUM.nunique()
# df_state_summary['Total Unique Providers to offer LPN role']=df_state[df_state['LPN_contract_hours']>0].groupby('STATE').PROVNUM.nunique()
# df_state_summary['Total Unique Providers to offer CNA role']=df_state[df_state['CNA_contract_hours']>0].groupby('STATE').PROVNUM.nunique()
# df_state_summary['Total Unique Providers to offer Other role']=df_state[df_state['Other_contract_hours']>0].groupby('STATE').PROVNUM.nunique()
```

In [78]:
```
df_state_providers  = pd.DataFrame({'Total_Providers':df_state.groupby('STATE').PROVNUM.nunique()
            , 'Total Unique Providers to offer RN role': df_state[df_state['RN_contract_hours']>0].groupby('STATE').PROVNUM.nunique()
            , 'Total Unique Providers to offer LPN role': df_state[df_state['LPN_contract_hours']>0].groupby('STATE').PROVNUM.nunique()
             , 'Total Unique Providers to offer CNA role': df_state[df_state['CNA_contract_hours']>0].groupby('STATE').PROVNUM.nunique()
             , 'Total Unique Providers to offer Other role': df_state[df_state['Other_contract_hours']>0].groupby('STATE').PROVNUM.nunique()
            }).reset_index()

df_state_providers.head()
```

Out[78]:

	STATE	Total_Providers	Total Unique Providers to offer RN role	Total Unique Providers to offer LPN role	Total Unique Providers to offer CNA role	Total Unique Providers to offer Other role
0	AK	15	9	6	9	1
1	AL	223	29	41	31	9
2	AR	217	18	41	31	129
3	AZ	139	67	74	78	9
4	CA	1140	264	432	483	123

In [79]:
```
# df_state_summary = df_state_summary.DataFrame.insert(-1,'Total Unique Providers',[df_state.groupby('STATE').PROVNUM.nunique().to_list()])
```

In [80]:
```
df_state_summary = df_state_summary.merge(df_state_providers, left_on='STATE',right_on='STATE')
```

In [81]:
```
df_state_summary.head()
```

Out[81]:

	STATE	RN_contract_hours	LPN_contract_hours	CNA_contract_hours	Other_contract_hours	Total_contract_hours	Total Hours Percentage	Total_Providers	Total Unique Providers to offer RN role	Total Unique Providers to offer LPN role	Total Unique Providers to offer CNA role	Total Unique Providers to offer Other role
0	NY	776877.65	1216464.72	1924905.03	121202.59	4039449.99	13.16	601	343	411	374	162
1	PA	411631.05	1107096.05	2024098.00	34919.21	3577744.31	11.65	665	434	499	479	113
2	NJ	176665.18	502039.89	1313177.45	44298.32	2036180.84	6.63	345	208	248	250	105
3	IL	369883.16	318700.45	844819.19	23925.03	1557327.83	5.07	675	398	369	374	117
4	CA	85274.23	289389.01	949042.36	22847.17	1346552.77	4.39	1140	264	432	483	123

In [82]:
```
df_lob_summary['label'] = ((df_lob_summary['Total Hours']/1000000).round(2).astype(str))+"M ("+(((df_lob_summary['Total Hours']/df_lob_summary['Total Hours'].sum())*100).round(2).astyp
```

In [83]:
```
df_state_summary['label'] = ((df_state_summary['Total_contract_hours']/1000000).round(2).astype(str))+"M ("+(df_state_summary['Total Hours Percentage'].cumsum().round(2).astype(str))+'
```

In [84]:
```
# save the clean data as a csv file
datapath = './clean_Data'
if not os.path.exists(datapath):
    os.mkdir(datapath)
datapath_pbj_data = os.path.join(datapath, 'df_state_summary.csv')
if not os.path.exists(datapath_pbj_data):
    df_state_summary.to_csv(datapath_pbj_data, index=False)
```

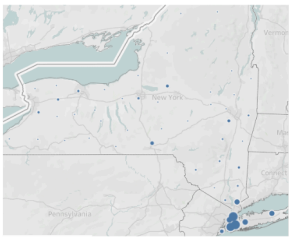
In [85]:
```
# df_state_summary['Total Hours Percentage'].cumsum().round(2).astype(str)
```

Recommendation 3

Summary of NY 2024-Q1 CNA Contract Hours with Key Counties and Providers

State	Total Providers	Total CNA Providers	Total County	Counties with CNA Provider	Total CNA Contract Hours
NY	601	374	60	51	1.92M

Most NY Counties with High CNA Contract Hours are located in SE



7 NY Counties offered more than 100K CNA Contract Hours

NY County	Total 2024 Q1 Contract Hours	Unique CNA Contract Hours Providers	CNA contract Hours per Provider
Queens	324,330.09	45	7,207.34
Bronx	310,866.55	40	7,771.66
Kings	273,339.90	38	7,193.16
Nassau	135,224.23	25	5,408.97
New York	126,746.99	15	8,449.80
Suffolk	111,150.09	27	4,116.67
Westchester	109,617.03	27	4,059.89
Richmond	56,137.70	8	7,017.21

NY County: Queens

Day of The Week: Sunday

Key Providers for Selected County and Day of The Week

Provider #	AVG CNA Hours	Min CNA Hours	Max CNA Hours	NY County	Provider Name	Provider Address	Telephone Number	ZIP Code
335791	519.46	470.75	573.25	Queens	QUEENS BOULEVARD EXTENDED CARE FACILITY	61 11 QUEENS BOULEVARD	7182050288	11377
335266	223.46	173.75	265.00	Queens	WOODCREST REHAB & RESIDENTIAL H C CENTER, L L C	119 09 26TH AVENUE	7187626100	11354
335317	214.06	185.00	240.75	Queens	PARK TERRACE CARE CENTER	59 20 VAN DOREN STREET	7185929200	11368
335363	205.03	163.00	263.25	Queens	OZANAM HALL OF QUEENS NURSING HOME INC	42 41 201ST STREET	7184232000	11361
335448	179.19	101.00	252.75	Queens	QUEENS NASSAU REHABILITATION AND NURSING CENTER	520 BEACH 19TH STREET	7184717400	11691
335349	168.13	124.75	223.25	Queens	CLIFFSIDE REHAB & RESIDENTIAL HEALTH CARE CENTER	119 - 19 GRAHAM COURT	7188860700	11354
335310	148.54	133.25	166.75	Queens	FOREST VIEW CENTER FOR REHABILITATION & NURSING	71 20 110TH STREET	7187933200	11375
335093	145.38	98.25	185.50	Queens	PARK NURSING HOME	128 BEACH 115TH STREET	7184746400	11694
335426	142.85	106.00	174.25	Queens	FRANKLIN CENTER FOR REHABILITATION AND NURSING	142 27 FRANKLIN AVENUE	7186703400	11355
335838	133.10	91.00	189.00	Queens	NEW YORK CENTER FOR REHABILITATION & NURSING	26-13 21ST STREET	7186264800	11102
335379	117.90	95.00	163.25	Queens	REGO PARK NURSING HOME	111 26 CORONA AVENUE	7185926400	11368
335799	89.69	77.75	104.50	Queens	UNION PLAZA CARE CENTER	33 23 UNION STREET	7186700700	11354

In [86]:
```
df_ny = df[df['STATE'] == 'NY']
```

In [87]:
```
df_ny.head()
```

Out[87]:

	PROVNUM	PROVNAME	CITY	STATE	COUNTY_NAME	COUNTY_FIPS	CY_Qtr	WorkDate	MDScensus	Hrs_RNDON	...	Provider Address	ZIP Code	Telephone Number	Provider Type	Provider Resides in Hospital	Provider Changed Ownership in Last 12 Months	
739375	335003	THE EMERALD PEEK REHABILITATION AND NURSING CE...	PEEKSKILL	NY	Westchester	119	2024Q1	2024-01-01	80	7.13	...	2000 EAST MAIN STREET	10566	9147378400	Medicare and Medicaid	N	N	(
739376	335003	THE EMERALD PEEK REHABILITATION AND NURSING CE...	PEEKSKILL	NY	Westchester	119	2024Q1	2024-01-02	79	8.00	...	2000 EAST MAIN STREET	10566	9147378400	Medicare and Medicaid	N	N	(
739377	335003	THE EMERALD PEEK REHABILITATION AND NURSING CE...	PEEKSKILL	NY	Westchester	119	2024Q1	2024-01-03	81	8.00	...	2000 EAST MAIN STREET	10566	9147378400	Medicare and Medicaid	N	N	(
739378	335003	THE EMERALD PEEK REHABILITATION AND NURSING CE...	PEEKSKILL	NY	Westchester	119	2024Q1	2024-01-04	80	8.00	...	2000 EAST MAIN STREET	10566	9147378400	Medicare and Medicaid	N	N	(
739379	335003	THE EMERALD PEEK REHABILITATION AND NURSING CE...	PEEKSKILL	NY	Westchester	119	2024Q1	2024-01-05	80	8.00	...	2000 EAST MAIN STREET	10566	9147378400	Medicare and Medicaid	N	N	(

5 rows × 46 columns

In [88]:
```
[df_ny['RN_contract_hours'].sum(), df_ny['CNA_contract_hours'].sum(), df_ny['LPN_contract_hours'].sum()]
```

Out[88]:
```
[776877.6499999999, 1924905.03, 1216464.7200000002]
```

In [89]:
```
df_ny_summary = df_state_summary[df_state_summary['STATE'] == 'NY']
df_ny_summary = df_ny_summary.assign(Total_Counties=df_state[df_state['STATE'] == 'NY']['COUNTY_NAME'].nunique())
df_ny_summary
```

Out[89]:

	STATE	RN_contract_hours	LPN_contract_hours	CNA_contract_hours	Other_contract_hours	Total_contract_hours	Total Hours Percentage	Total_Providers	Total Unique Providers to offer RN role	Total Unique Providers to offer LPN role	Total Unique Providers to offer CNA role	Total Unique Providers to offer Other role	label	Tota
0	NY	776877.65	1216464.72	1924905.03	121202.59	4039449.99	13.16	601	343	411	374	162	4.04M (13.16%)	

In [90]:
```
# save the clean data as a csv file
datapath = './clean_Data'
if not os.path.exists(datapath):
    os.mkdir(datapath)
```

```
datapath_pbj_data = os.path.join(datapath, 'df_ny_summary.csv')
if not os.path.exists(datapath_pbj_data):
    df_ny_summary.to_csv(datapath_pbj_data, index=False)
```

In [91]:
```
df_ny_cna_providers = df_ny[['PROVNUM','PROVNAME','Provider Address','Telephone Number','Provider Resides in Hospital','Provider Type','COUNTY_NAME','ZIP Code']]
```

In [92]:
```
df_ny_cna_providers = df_ny_cna_providers.drop_duplicates()
df_ny_cna_providers.head()
```

Out[92]:

	PROVNUM	PROVNAME	Provider Address	Telephone Number	Provider Resides in Hospital	Provider Type	COUNTY_NAME	ZIP Code
739375	335003	THE EMERALD PEEK REHABILITATION AND NURSING CE...	2000 EAST MAIN STREET	9147378400	N	Medicare and Medicaid	Westchester	10566
739466	335004	AUBURN REHABILITATION & NURSING CENTER	85 THORNTON AVENUE	3152537351	N	Medicare and Medicaid	Cayuga	13021
739557	335005	BRIARCLIFF MANOR CENTER FOR REHAB AND NURSING ...	620 SLEEPY HOLLOW ROAD	9149415100	N	Medicare and Medicaid	Westchester	10510
739648	335006	KATHERINE LUTHER RESIDENTIAL HLTH CARE & REHAB	110 UTICA ROAD	3158535515	N	Medicare and Medicaid	Oneida	13323
739739	335008	ST JOHNS HEALTH CARE CORPORATION	150 HIGHLAND AVENUE	5852715413	N	Medicare and Medicaid	Monroe	14620

In [93]:
```
df_ny_cna_prov_hours = df_ny.groupby(['PROVNUM','day_of_week'],as_index=False).agg({'CNA_contract_hours':['mean','min','max']}).reset_index()


df_ny_cna_prov_hours .columns = df_ny_cna_prov_hours .columns.to_flat_index()
df_ny_cna_prov_hours .columns = ['_'.join(col) for col in df_ny_cna_prov_hours .columns]

df_ny_cna_prov_hours.drop('index_',axis=1, inplace=True)
df_ny_cna_prov_hours.rename(columns={'PROVNUM_':'PROVNUM', 'day_of_week_':'day_of_week'}, inplace =True)

df_ny_cna_prov_hours = df_ny_cna_prov_hours.sort_values(by='CNA_contract_hours_mean', ascending=False)


df_ny_cna_prov_hours.head()
```

Out[93]:

	PROVNUM	day_of_week	CNA_contract_hours_mean	CNA_contract_hours_min	CNA_contract_hours_max
340	335100	Thursday	619.702308	547.97	694.30
3721	335791	Thursday	587.346154	486.75	666.25
342	335100	Wednesday	586.702308	490.38	689.18
336	335100	Friday	578.220000	492.50	642.99
3719	335791	Saturday	572.942308	486.00	638.75

In [94]:
```
# df_ny_zero_cna_hours = df_ny.groupby(['PROVNUM'])['CNA_contract_hours'].sum().reset_index()
# df_ny_zero_cna_hours = df_ny_zero_cna_hours[df_ny_zero_cna_hours.PROVNUM == 0]
# df_ny_zero_cna_hours
```

In [95]:
```
# save the clean data as a csv file
datapath = './clean_Data'
if not os.path.exists(datapath):
    os.mkdir(datapath)
datapath_pbj_data = os.path.join(datapath, 'df_ny_cna_providers.csv')
if not os.path.exists(datapath_pbj_data):
    df_ny_cna_providers.to_csv(datapath_pbj_data, index=False)
```

In [96]:
```
df_ny_cna_providers[df_ny_cna_providers['Telephone Number'] == '9142946300']
```

Out[96]:

	PROVNUM	PROVNAME	Provider Address	Telephone Number	Provider Resides in Hospital	Provider Type	COUNTY_NAME	ZIP Code
793975	33A246	ELIZABETH SETON CHILDREN'S CENTER	300 CORPORATE BLVD SOUTH	9142946300	N	Medicaid	Westchester	10701

In [97]:
```
# save the clean data as a csv file
datapath = './clean_Data'
if not os.path.exists(datapath):
    os.mkdir(datapath)
datapath_pbj_data = os.path.join(datapath, 'df_ny_cna_prov_hours.csv')
if not os.path.exists(datapath_pbj_data):
    df_ny_cna_prov_hours.to_csv(datapath_pbj_data, index=False)
```

In [98]:
```
# # save the clean data as a csv file
# datapath = './clean_Data'
# if not os.path.exists(datapath):
#     os.mkdir(datapath)
# datapath_pbj_data = os.path.join(datapath, 'df_ny_prov_hours.csv')
# if not os.path.exists(datapath_pbj_data):
#     df_ny_prov_hours.to_csv(datapath_pbj_data, index=False)
```

In [99]:
```
df_ny.COUNTY_NAME.nunique()
```

Out[99]: 60

In [100…
```
df_ny_county_prov = df_ny[df_ny['CNA_contract_hours']>0].groupby('COUNTY_NAME').PROVNUM.nunique().reset_index()
df_ny_county_prov = df_ny_county_prov.sort_values(by='PROVNUM', ascending=False)


df_ny_county_prov.head()
```

Out[100…

	COUNTY_NAME	PROVNUM
33	Queens	45
2	Bronx	40
16	Kings	38
48	Westchester	27
41	Suffolk	27

In [101…
```
df_ny_county_prov.nunique()
```

Out[101…
```
COUNTY_NAME    51
PROVNUM        16
dtype: int64
```

In [102…
```
df_ny_county_hours = df_ny[df_ny['CNA_contract_hours']>0].groupby('COUNTY_NAME',as_index=False).agg({'CNA_contract_hours':['sum','mean','min','max'],'PROVNUM':'nunique'}).reset_index()


df_ny_county_hours .columns = df_ny_county_hours .columns.to_flat_index()
df_ny_county_hours .columns = ['_'.join(col) for col in df_ny_county_hours .columns]

df_ny_county_hours.drop('index_',axis=1, inplace=True)
df_ny_county_hours.rename(columns={'COUNTY_NAME_':'COUNTY_NAME'}, inplace =True)

df_ny_county_hours = df_ny_county_hours.sort_values(by='CNA_contract_hours_sum', ascending=False)


df_ny_county_hours.head()
```

Out[102…

	COUNTY_NAME	CNA_contract_hours_sum	CNA_contract_hours_mean	CNA_contract_hours_min	CNA_contract_hours_max	PROVNUM_nunique
33	Queens	324330.09	91.334861	0.25	666.25	45
2	Bronx	310866.55	95.887276	0.95	474.75	40
16	Kings	273339.90	85.632801	0.75	483.70	38
22	Nassau	135224.23	73.015243	0.25	270.25	25
23	New York	126746.99	104.576724	0.35	694.30	15

In [103…
```
df_ny_summary = df_ny_summary.assign(Total_Counties=df_state[df_state['STATE'] == 'NY']['COUNTY_NAME'].nunique())
df_ny_county_hours = df_ny_county_hours.assign(CNA_contract_hours_per_provider=df_ny_county_hours['CNA_contract_hours_sum']/df_ny_county_hours['PROVNUM_nunique'])
```

In [104…
```
# df_ny_county_hours
```

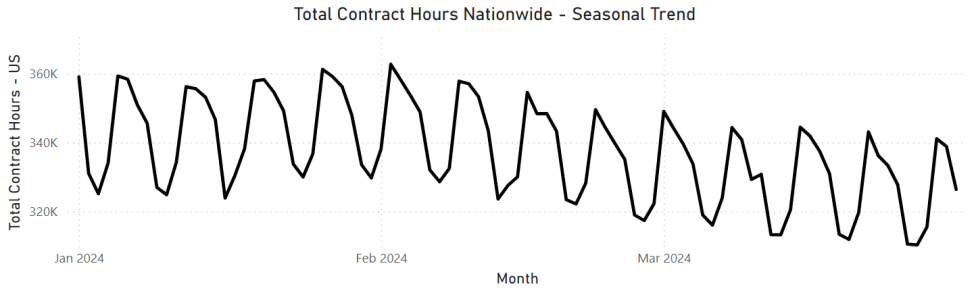
In [105…
```
# df_ny_county_hours.nunique()
```

In [106…
```
# save the clean data as a csv file
datapath = './clean_Data'
if not os.path.exists(datapath):
    os.mkdir(datapath)
datapath_pbj_data = os.path.join(datapath, 'df_ny_county_hours.csv')
if not os.path.exists(datapath_pbj_data):
    df_ny_county_hours.to_csv(datapath_pbj_data, index=False)
```

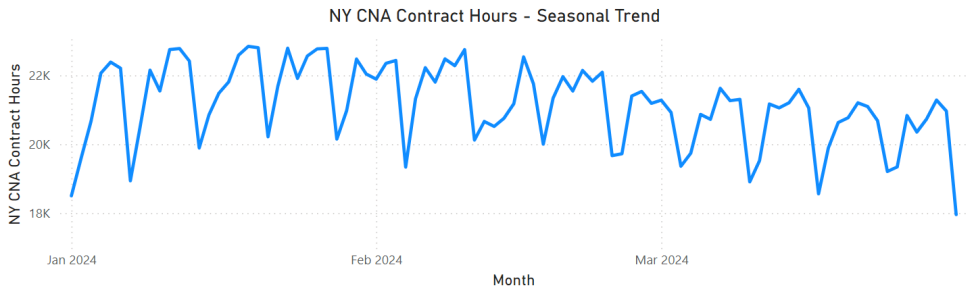
In [107…
```
df.columns
```

Out[107…
```
Index(['PROVNUM', 'PROVNAME', 'CITY', 'STATE', 'COUNTY_NAME', 'COUNTY_FIPS',
       'CY_Qtr', 'WorkDate', 'MDScensus', 'Hrs_RNDON', 'Hrs_RNDON_emp',
       'Hrs_RNDON_ctr', 'Hrs_RNadmin', 'Hrs_RNadmin_emp', 'Hrs_RNadmin_ctr',
       'Hrs_RN', 'Hrs_RN_emp', 'RN_contract_hours', 'Hrs_LPNadmin',
       'Hrs_LPNadmin_emp', 'Hrs_LPNadmin_ctr', 'Hrs_LPN', 'Hrs_LPN_emp',
       'LPN_contract_hours', 'Hrs_CNA', 'Hrs_CNA_emp', 'CNA_contract_hours',
       'Hrs_NAtrn', 'Hrs_NAtrn_emp', 'Hrs_NAtrn_ctr', 'Hrs_MedAide',
       'Hrs_MedAide_emp', 'Hrs_MedAide_ctr', 'Total_Hours', 'Total_Hours_emp',
       'Total_contract_hours', 'Provider Address', 'ZIP Code',
       'Telephone Number', 'Provider Type', 'Provider Resides in Hospital',
       'Provider Changed Ownership in Last 12 Months', 'Ownership Type',
       'day_of_week', 'is_weekend', 'Other_contract_hours'],
      dtype='object')
```

Bonus Recommendation



Day	Total Contract Hours
Friday	4.6M
Saturday	4.5M
Sunday	4.5M
Monday	4.4M
Thursday	4.3M
Tuesday	4.2M
Wednesday	4.2M
Total	**30.7M**



Day	NY - CNA Contract Hours
Friday	286.27K
Saturday	284.14K
Thursday	283.58K
Tuesday	278.08K
Wednesday	276.49K
Monday	264.06K
Sunday	252.28K
Total	**1,924.91K**

In [108…
```
time_series_overall = df[['WorkDate', 'day_of_week' ,'Total_contract_hours']]
```

In [109…

```
time_series_ny_cna= df[['WorkDate', 'day_of_week','CNA_contract_hours']][df['STATE']=='NY']
```

In [110…

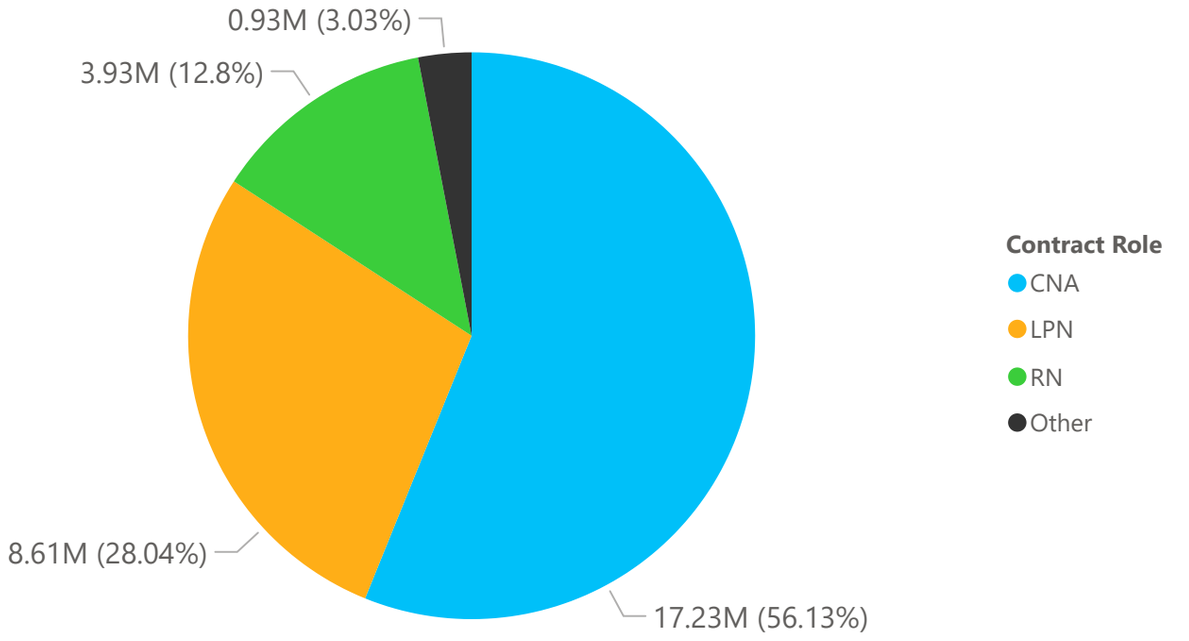
```
# save the clean data as a csv file
datapath = './clean_Data'
if not os.path.exists(datapath):
    os.mkdir(datapath)
datapath_pbj_data = os.path.join(datapath, 'time_series_overall .csv')
if not os.path.exists(datapath_pbj_data):
    time_series_overall .to_csv(datapath_pbj_data, index=False)
```

In [111…

```
# save the clean data as a csv file
datapath = './clean_Data'
if not os.path.exists(datapath):
    os.mkdir(datapath)
datapath_pbj_data = os.path.join(datapath, 'time_series_ny_cna.csv')
if not os.path.exists(datapath_pbj_data):
    time_series_ny_cna.to_csv(datapath_pbj_data, index=False)
```

Which Roles makeup most of Total Contract Hours?

Three roles make up 97% of Contract Hours in 2024 Q1

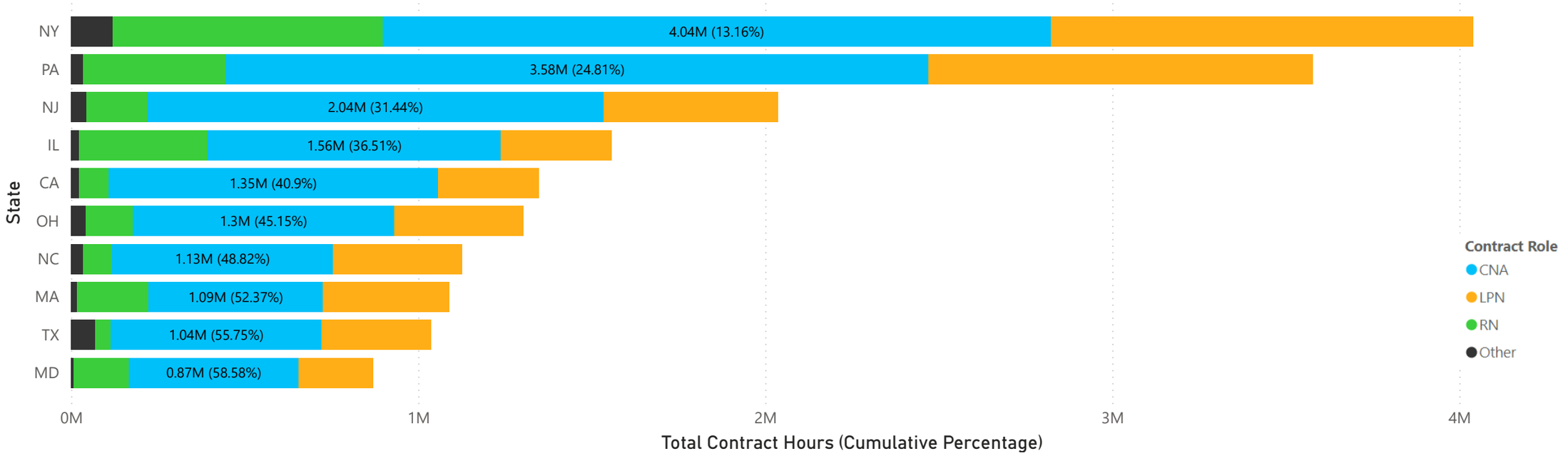


Contract Role	Total Contract Hours	Total Unique Providers to offer this Role	Average Contract Hrs per Provider
CNA	17,231,892.04	6865	2510.11
LPN	8,608,154.24	6964	1236.09
RN	3,929,326.87	5917	664.07
Other	930,360.33	3236	287.50

Quick measure

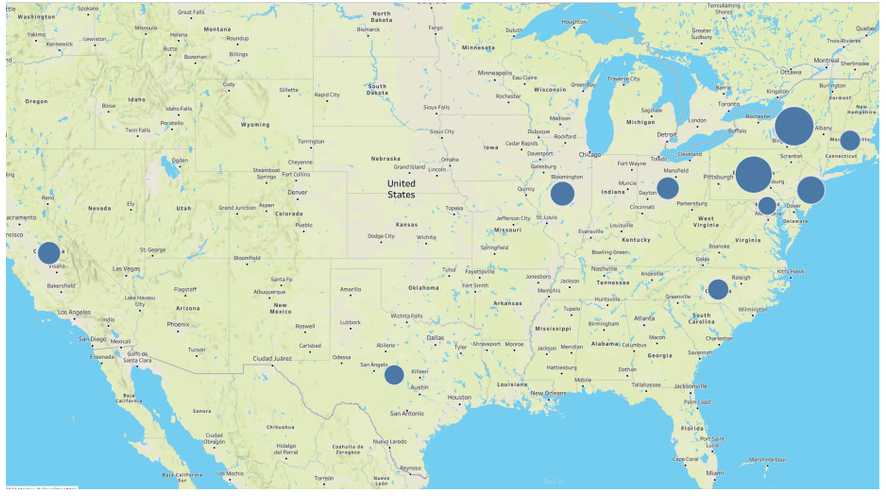
Which States offer Most Contract Hours?

Top 10 States make up nearly 60% of Total Contract Hours



Most of Top 10 States are located in NE Region

(Size of Bubble indicates total Contract Hours)



NY, PA, and NJ also offer high contract hours per Provider

State	Total Contract Hours	Total Providers	Total contract Hours per Provider
NY	4,039,449.99	601	6,721.21
PA	3,577,744.31	665	5,380.07
NJ	2,036,180.84	345	5,901.97
IL	1,557,327.83	675	2,307.15
CA	1,346,552.77	1140	1,181.19
OH	1,304,113.78	918	1,420.60
NC	1,126,111.32	410	2,746.61
MA	1,090,336.45	342	3,188.12
TX	1,037,825.88	1164	891.60
MD	870,030.27	220	3,954.68

Quick measure

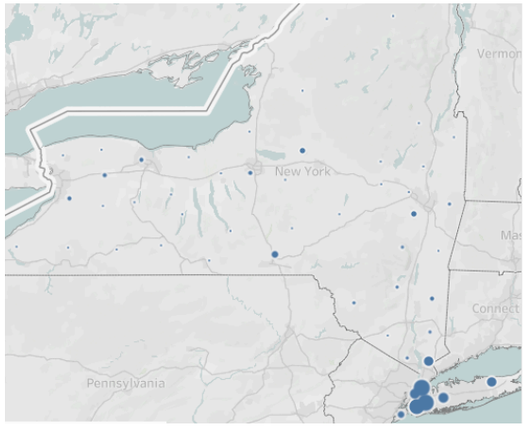
Summary of NY 2024-Q1 CNA Contract Hours with Key Counties and Providers

State	Total Providers	Total CNA Providers	Total County	Counties with CNA Provider	Total CNA Contract Hours
NY	601	374	60	51	1.92M

7 NY Counties offered more than 100K CNA Contract Hours

NY County	Total 2024 Q1 Contract Hours	Unique CNA Contract Hours Providers	CNA contract Hours per Provider
Queens	324,330.09	45	7,207.34
Bronx	310,866.55	40	7,771.66
Kings	273,339.90	38	7,193.16
Nassau	135,224.23	25	5,408.97
New York	126,746.99	15	8,449.80
Suffolk	111,150.09	27	4,116.67
Westchester	109,617.03	27	4,059.89
Richmond	56,137.70	8	7,017.21

Most NY Counties with High CNA Contract Hours are located in SE



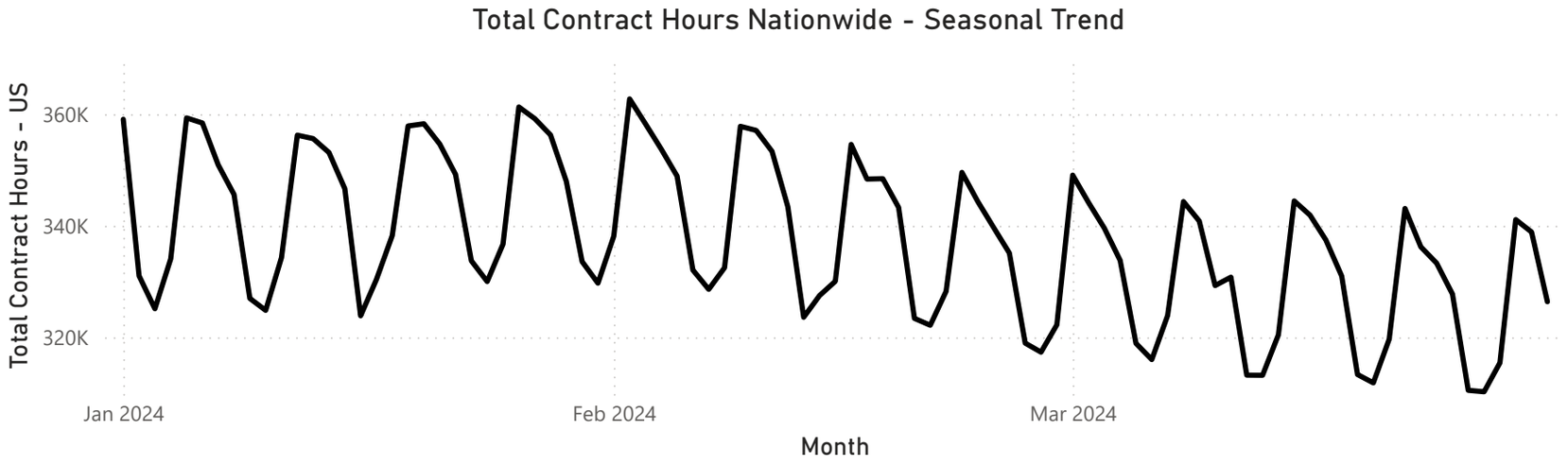
NY County: Queens

Day of The Week: Sunday

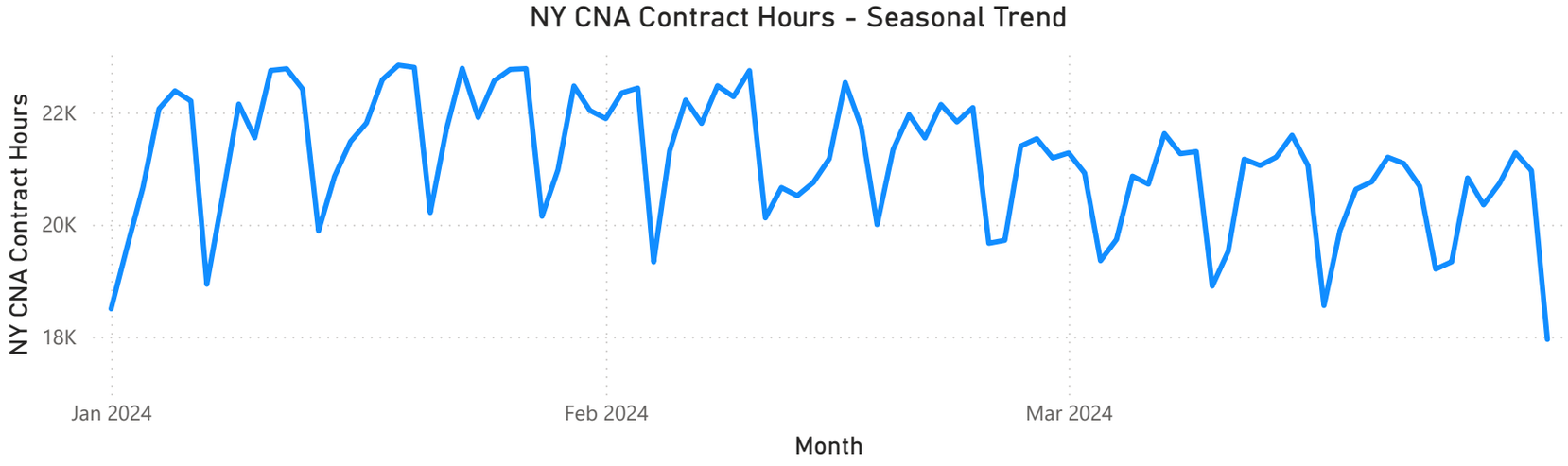
Key Providers for Selected County and Day of The Week

Provider #	AVG CNA Hours	Min CNA Hours	Max CNA Hours	NY County	Provider Name	Provider Address	Telephone Number	ZIP Code
335791	519.46	470.75	573.25	Queens	QUEENS BOULEVARD EXTENDED CARE FACILITY	61 11 QUEENS BOULEVARD	7182050288	11377
335266	223.46	173.75	265.00	Queens	WOODCREST REHAB & RESIDENTIAL H C CENTER, L L C	119 09 26TH AVENUE	7187626100	11354
335317	214.06	185.00	240.75	Queens	PARK TERRACE CARE CENTER	59 20 VAN DOREN STREET	7185929200	11368
335363	205.03	163.00	263.25	Queens	OZANAM HALL OF QUEENS NURSING HOME INC	42 41 201ST STREET	7184232000	11361
335448	179.19	101.00	252.75	Queens	QUEENS NASSAU REHABILITATION AND NURSING CENTER	520 BEACH 19TH STREET	7184717400	11691
335349	168.13	124.75	223.25	Queens	CLIFFSIDE REHAB & RESIDENTIAL HEALTH CARE CENTER	119 - 19 GRAHAM COURT	7188860700	11354
335310	148.54	133.25	166.75	Queens	FOREST VIEW CENTER FOR REHABILITATION & NURSING	71 20 110TH STREET	7187933200	11375
335093	145.38	98.25	185.50	Queens	PARK NURSING HOME	128 BEACH 115TH STREET	7184746400	11694
335426	142.85	106.00	174.25	Queens	FRANKLIN CENTER FOR REHABILITATION AND NURSING	142 27 FRANKLIN AVENUE	7186703400	11355
335838	133.10	91.00	189.00	Queens	NEW YORK CENTER FOR REHABILITATION & NURSING	26-13 21ST STREET	7186264800	11102
335379	117.90	95.00	163.25	Queens	REGO PARK NURSING HOME	111 26 CORONA AVENUE	7185926400	11368
335799	89.69	77.75	104.50	Queens	UNION PLAZA CARE CENTER	33 23 UNION STREET	7186700700	11354

Quick measure



Day	Total Contract Hours
Friday	4.6M
Saturday	4.5M
Sunday	4.5M
Monday	4.4M
Thursday	4.3M
Tuesday	4.2M
Wednesday	4.2M
Total	**30.7M**



Day	NY - CNA Contract Hours
Friday	286.27K
Saturday	284.14K
Thursday	283.58K
Tuesday	278.08K
Wednesday	276.49K
Monday	264.06K
Sunday	252.28K
Total	**1,924.91K**

Quick measure