Clipboard Health Assignment Solution - By Jaydeep Chavda (Email)

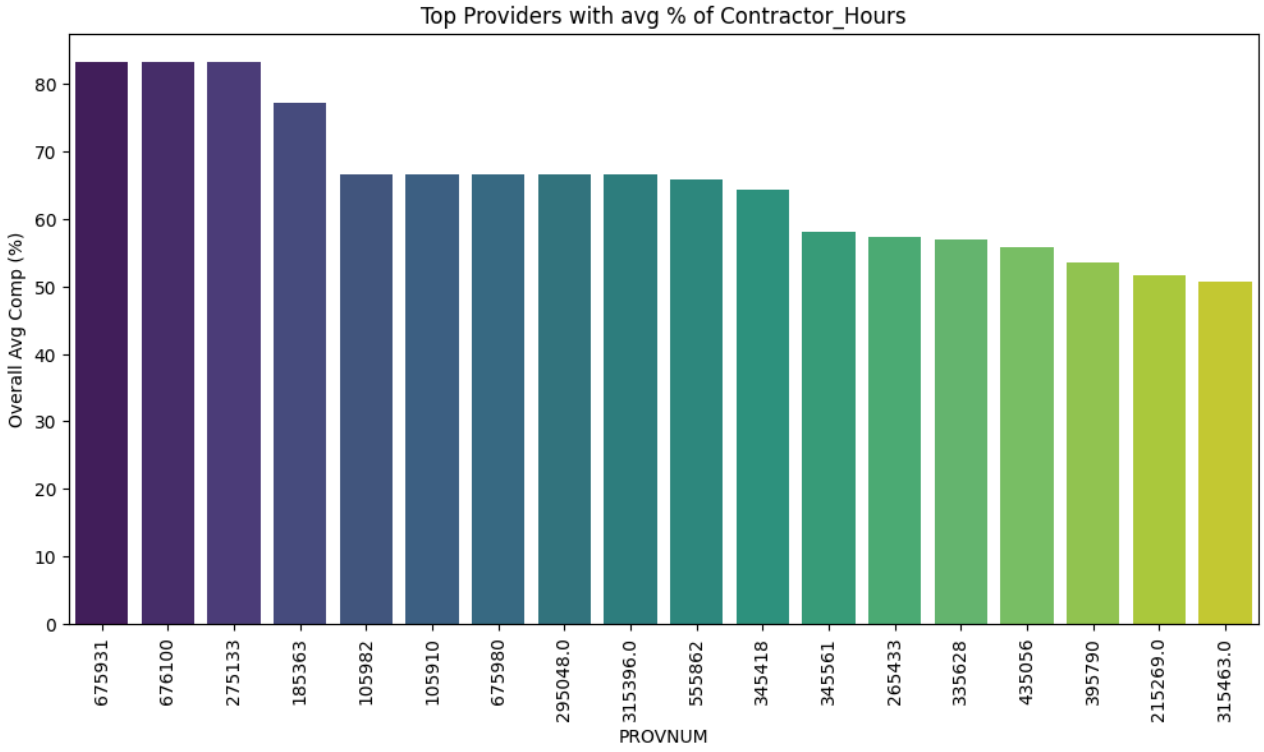
Data Cleaning :
→ Removed NULL value rows
→ CNA and LPN are major category so rows which has zero value for both of these removed
→ Removed all rows with contractor_hours is significantly less than Employee_Hours
→ RNDON and RNAdmin have neglected because of small contribution

Assumption :
→ on an average contractor nurse works for 7 hours/day
→ Data which is neglected has no/minimal significance

Methodologies
→ summed up all ctr hours(CNA_ctr+LPN_ctr+RN_ctr etc) and emp hours(CNA_emp +LPN_emp+RN_emp etc) for all rows
→ Found ctr% by (ctr_hours)*100/(total_hours) for each row
→ removed all ctr% < 50% (so data represents locations with ctr > emp)
→ for providers, I grouped data by providers and sum(ctr_hours), sum(emp_hours) to get providers wise analysis and finally got no.of.ctr or ctr_count = avg(ctr_hours)/7 hours
→ did same for location wise analysis
→ used power BI (used measures, charts) for graphs, and python for data cleaning

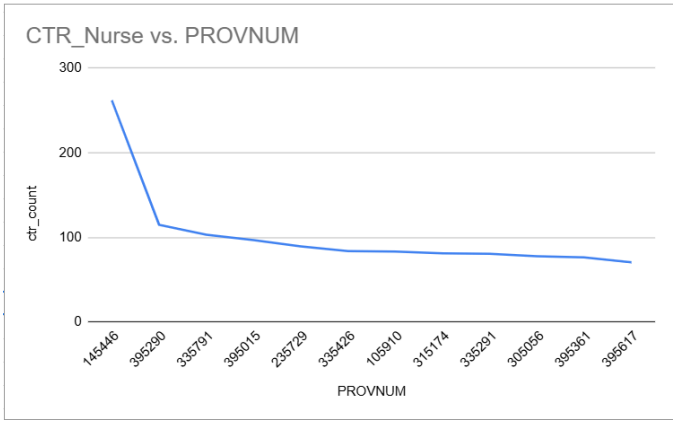
i) Analysis of top Providers

1) List of providers whose proportion of contractor is high (data shows % of contractor workers out of total in all units) in centers
→ sales team should start focusing upon their strategies to understand their market presence and why they have a high proportion of contractors in centers.



Providers with their number of contractor Nurses

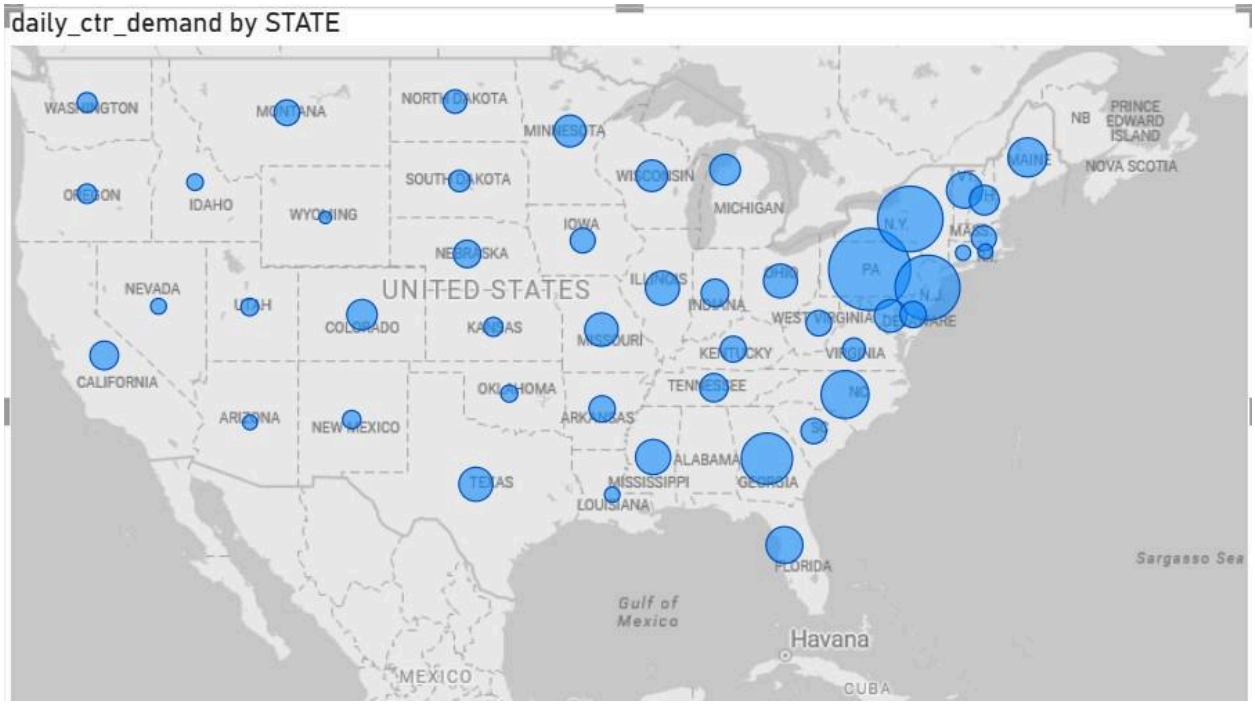
PROVNUM	PROVNAME	ctr_count
145446	MARIGOLD REHABILITATION HCC	262
395290	PLEASANT ACRES REHABILITATION AND NURS...	115
335791	QUEENS BOULEVARD EXTENDED CARE FACILITY	103
395015	BRIGHTON REHABILITATION AND WELLNESS C...	97
235729.0	MICHIGAN VETERAN HOMES AT GRAND RAPIDS	89
335426	FRANKLIN CENTER FOR REHABILITATION AND ...	84
105910	CORAL REEF SUBACUTE CARE CENTER LLC	83
315174	DEPTFORD CENTER FOR REHABILITATION AND...	81
335291	WILLOW POINT REHABILITATION AND NURSIN...	81
305056.0	MERRIMACK COUNTY NURSING HOME	78
395361	PLEASANT RIDGE MANOR EAST/WEST	77
395617	JOHN J KANE REGIONAL CENTER-SC	71



→ This data shows the count of contractor nurses across cities. Sales team should focus on their strategies to understand their strong market presence and network with their high number of contractors. Sales team should come up with an approach to bring those contractors to Clipboard.

ii) Analysis of Locations

This graph shows state wise contractor nurse demand, we can clearly see we should focus in western US



CITY	daily_ctr_demand	dailly_emp_demand	Average of ctr_hours %
GALESBURG	262	16	61.63
YORK	115	49	69.99
WOODSIDE	103	43	70.48
BEAVER	97	72	57.49
GRAND RAPIDS	89	32	73.90
MIAMI	83	0	100.00
DEPTFORD	81	28	74.31
VESTAL	81	53	60.46
BOSCAWEN	78	63	55.23
GIRARD	77	41	65.10
BRIDGETON	70	0	100.00

This table shows daily contractor nurses demand by cities, and the sales team should target these cities.
By strategizing and giving good service can help Clipboard Health to capture market

CITY	daily_ctr_demand	dailly_emp_demand	Average of ctr_hours %
BRIDGETON	70	0	100.00
CAMP WOOD	31	0	100.00
CEDARBURG	9	0	100.00
CUMMING	46	0	100.00
DRIPPING SPRINGS	16	0	100.00
ELK GROVE VILLAGE	14	0	100.00
HARRISBURG	9	0	100.00
MIAMI	83	0	100.00
PORT CHARLOTTE	59	0	100.00
RANDOLPH CENTER	7	0	100.00
SAN ANGELO	39	0	100.00
SUN VALLEY	30	0	99.66

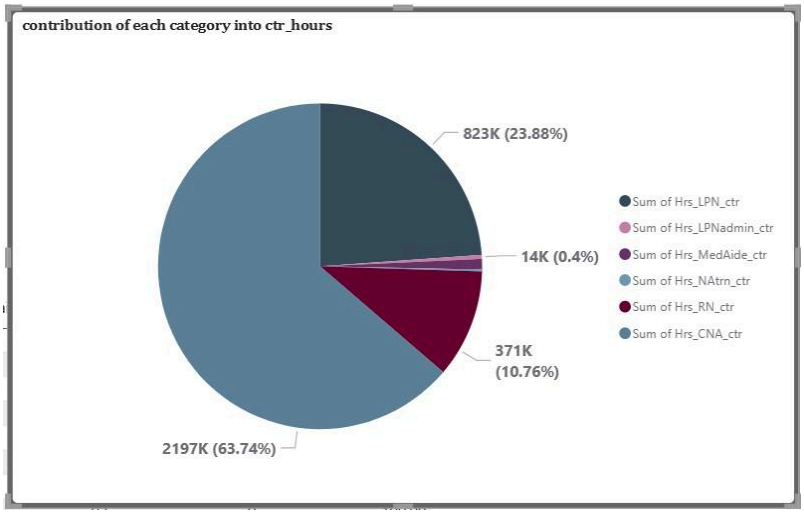
This is the data of cities where centers are relying totally on contractor nurses.

By targeting high demand centers, sales team can help in capturing market

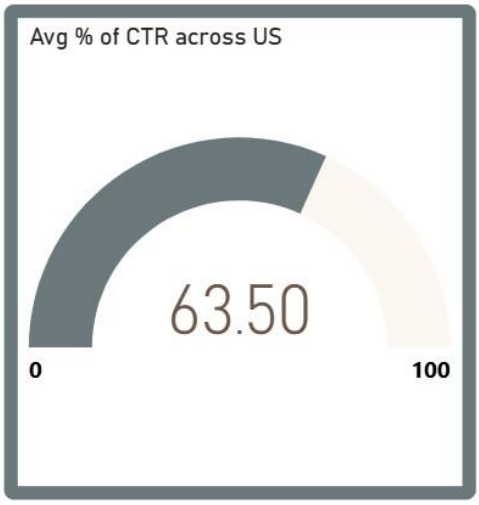
iii) Categories of nurses to focus on

Given chart shows contribution of nurses category wise, data shows that below three categories are important
1) CNA (63% demand)
2) LPN (24% demand)
3) RN (10% demand)

Pie chart of nurse categories



63.5% nurses are working in contract



SQL Solution :

Q1)

```sql
SELECT c.customer_name,  p.product_name, s.total_amount
 from Sales s
JOIN Customers c ON s.customer_id = c.customer_id
  join Products p ON s.product_id = p.product_id
where s.sale_date >= CURDATE() - INTERVAL 30 DAY;
```

Q2)

```sql
SELECT p.category,ROUND(SUM(s.total_amount), 2) as total_revenue
FROM  Sales s
JOIN Products p on s.product_id = p.product_id
WHERE s.sale_date >= DATE_SUB(CURRENT_DATE, INTERVAL 1 YEAR)
GROUP BY p.category
ORDER by total_revenue desc;
```

Q3)

```sql
SELECT c.customer_name FROM Sales s
JOIN Customers c ON c.customer_id = s.customer_id
WHERE c.sales_region = 'West' AND YEAR(s.sale_date) = 2023
```

Q4)

```sql
SELECT  c.customer_name,COUNT(s.sales_id) AS total_sales,
```

```
    SUM(s.quantity) AS total_quantity,
  ROUND(SUM(s.total_amount), 2) AS total_revenue
FROM  Customers c
LEFT JOIN   Sales s ON c.customer_id = s.customer_id
GROUP BY  c.customer_name
```

Q5)

```
SELECT  c.customer_name,SUM(s.total_amount) AS total_revenue
FROM Customers c
JOIN Sales s ON c.customer_id = s.customer_id
WHERE YEAR(s.sale_date) = 2023
GROUP BY c.customer_name
ORDER BY total_revenue desc limit 3;
```

Q6)

```
SELECT  p.product_name,SUM(s.quantity) AS total_quantity_sold
       DENSE_RANK() OVER (ORDER BY SUM(s.quantity) DESC) AS sales_rank
FROM  Products p
JOIN Sales s ON p.product_id = s.product_id
WHERE YEAR(s.sale_date) = 2023
GROUP BY p.product_id,
ORDER BY sales_rank, total_quantity_sold DESC;
```

Q7)

```
SELECT customer_name, sales_region,
    CASE
       WHEN sign_up_date >= DATE_SUB(CURRENT_DATE, INTERVAL 6 MONTH)
        THEN 'New' ELSE 'Existing'
  END AS customer_category
FROM  Customers
```

Q8)

```
SELECT
 DATE_FORMAT(s.sale_date, '%Y-%m') AS month_year,
 SUM(s.total_amount) AS total_sales
FROM Sales s
```

```sql
WHERE s.sale_date >= CURDATE() - INTERVAL 12 MONTH
GROUP BY month_year
ORDER BY month_year ASC;
```

Q9)

```sql
select  p.category, SUM(s.total_amount) AS total_revenue
from Sales s
JOIN Products p ON s.product_id = p.product_id
WHERE  s.sale_date >= DATE_SUB(CURRENT_DATE, INTERVAL 6 MONTH)
group by p.category
HAVING   total_revenue > 50000
ORDER BY  total_revenue DESC;
```

Q10)

```sql
SELECT
    s.sales_id, s.customer_id,s.product_id,
  p.product_name,s.quantity, p.price AS unit_price,
   s.total_amount AS actual_total_amount,
 s.quantity * p.price AS expected_total_amount,
   ABS(s.total_amount - (s.quantity * p.price)) AS difference
FROM Sales s
JOIN Products p ON s.product_id = p.product_id
WHERE s.total_amount != (s.quantity * p.price)
ORDER BY difference DESC;
```