

Nurse staffing strategies for enhanced patient care

Matthew Bain

2024-03-22

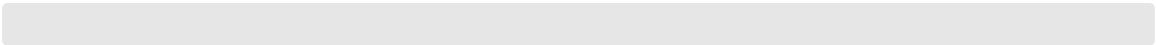
abstract I analyze a medical staffing dataset and identify avenues to improve work satisfaction among nurses and the quality of care provided at United States medical institutions.

Table of contents

- Imports 1
- The dataset 2
 - Load the data 2
 - Inspect the data 2
 - Group the features 5
 - Clean the data 7
- Explore the dataset 7
 - Visualize distributions 7
 - Visualize relationships 7
 - Compare groups 9
- Feature engineer 9
 - Join geographical data 9
 - Join seasonal data 9
- Analyze geography 9
- Analyze seasonality 9
- Model 10
- Extra visualizations 10
 - Sparklines 10
- Archive 10
- Bibliography 11

[...]

Imports



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from great_tables import GT
from pandas.plotting import scatter_matrix

from src.stylesheet import customize_plots
from src.inspection import make_df, display, display2
```

The dataset

Load the data

We begin by exploring the data to get to know the features and patterns on which we will base our analysis.

```
if 'data' not in locals():
    data = pd.read_csv(
        "../data/raw/PBJ_Daily_Nurse_Staffing_Q1_2024.zip",
        encoding='ISO-8859-1',
        low_memory=False
    )
else:
    print("data loaded.")
```

Inspect the data

```
data.sample(5)
```

[illegible]

```

<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)

```

```

data.describe().round(1)
# display(Markdown(data.describe().to_markdown()))

```

[illegible]

```
df = data.loc[:, [
    "STATE",
    "COUNTY_NAME", "COUNTY_FIPS",
    "CITY",
    "PROVNAME", "PROVNUM",
    # "MDScensus"
]]
df.value_counts()
df.to_frame()
# GT(df.reset_index().head(n=5))
```

| STATE | COUNTY_ NAME | COUNTY_ FIPS | CITY | PROVNAME | PROVNUM | count |
|-------|-----------------|-----------------|-------------------|--|---------|-------|
| | | | | | | |
| AK | Anchorage | 20 | ANCHOR- AGE | PRESTIGE CARE & RE- HAB CEN- TER OF AN- CHORAGE | 025025 | 91 |
| OH | Allen | 3 | LIMA | LIMA CON- VALESCENT HOME | 366297 | 91 |
| | | | | SHAWNEE MANOR | 365361 | 91 |
| | | | | SPRINGS OF LIMA THE | 366464 | 91 |
| | | | | SPRINGVIEW MANOR | 366221 | 91 |
| ... | ... | ... | ... | ... | ... | ... |
| IN | Tippecanoe | 157 | WEST LAFAYETTE | HERITAGE HEALTH- CARE | 155402 | 91 |
| | | | | INDIANA VETERANS HOME | 155787 | 91 |
| | | | | UNIVER- SITY PLACE HEALTH CENTER AND AS- SISTED LIV- ING | 155725 | 91 |
| | | | | WEST- MINSTER VILLAGE - WEST LAFAYETTE | 155177 | 91 |
| WY | Weston | 45 | NEWCAS- TLE | WESTON COUNTY HEALTH SERVICES | 535023 | 91 |

```
display2(
    "data['STATE'].value_counts()",
    "data['COUNTY_NAME'].value_counts()",
    "data['CITY'].value_counts()",
    "data['PROVNAME'].value_counts()",
    "data['MDScensus'].value_counts()",
    width="340px",
    globs=globals()
)
```

<IPython.core.display.HTML object>

```
data[["CY_Qtr", "WorkDate", "MDScensus"]]
```

| | CY_Qtr | WorkDate | MDScensus |
|---------|--------|----------|-----------|
| 0 | 2024Q1 | 20240101 | 50 |
| 1 | 2024Q1 | 20240102 | 49 |
| 2 | 2024Q1 | 20240103 | 49 |
| 3 | 2024Q1 | 20240104 | 50 |
| 4 | 2024Q1 | 20240105 | 51 |
| ... | ... | ... | ... |
| 1330961 | 2024Q1 | 20240327 | 81 |
| 1330962 | 2024Q1 | 20240328 | 83 |
| 1330963 | 2024Q1 | 20240329 | 85 |
| 1330964 | 2024Q1 | 20240330 | 82 |
| 1330965 | 2024Q1 | 20240331 | 82 |

Clean the data

Explore the dataset

Visualize distributions

Visualize relationships

```
attributes = ["Hrs_RN", "Hrs_LPN_ctr", "Hrs_CNA", "Hrs_NAtrn", "Hrs_MedAide"]
n = len(attributes)

fig, axs = plt.subplots(n, n, figsize=(8, 8))
scatter_matrix(
```

```
data[attributes].sample(200),
ax=axis, alpha=.7,
hist_kwds=dict(bins=15, linewidth=0)
)
fig.align_ylabels(axes[:, 0])
fig.align_xlabels(axes[-1, :])
for ax in axes.flatten():
    ax.tick_params(axis='both', which='both', length=3.5)

# save_fig("scatter_matrix_plot")

plt.show()
```

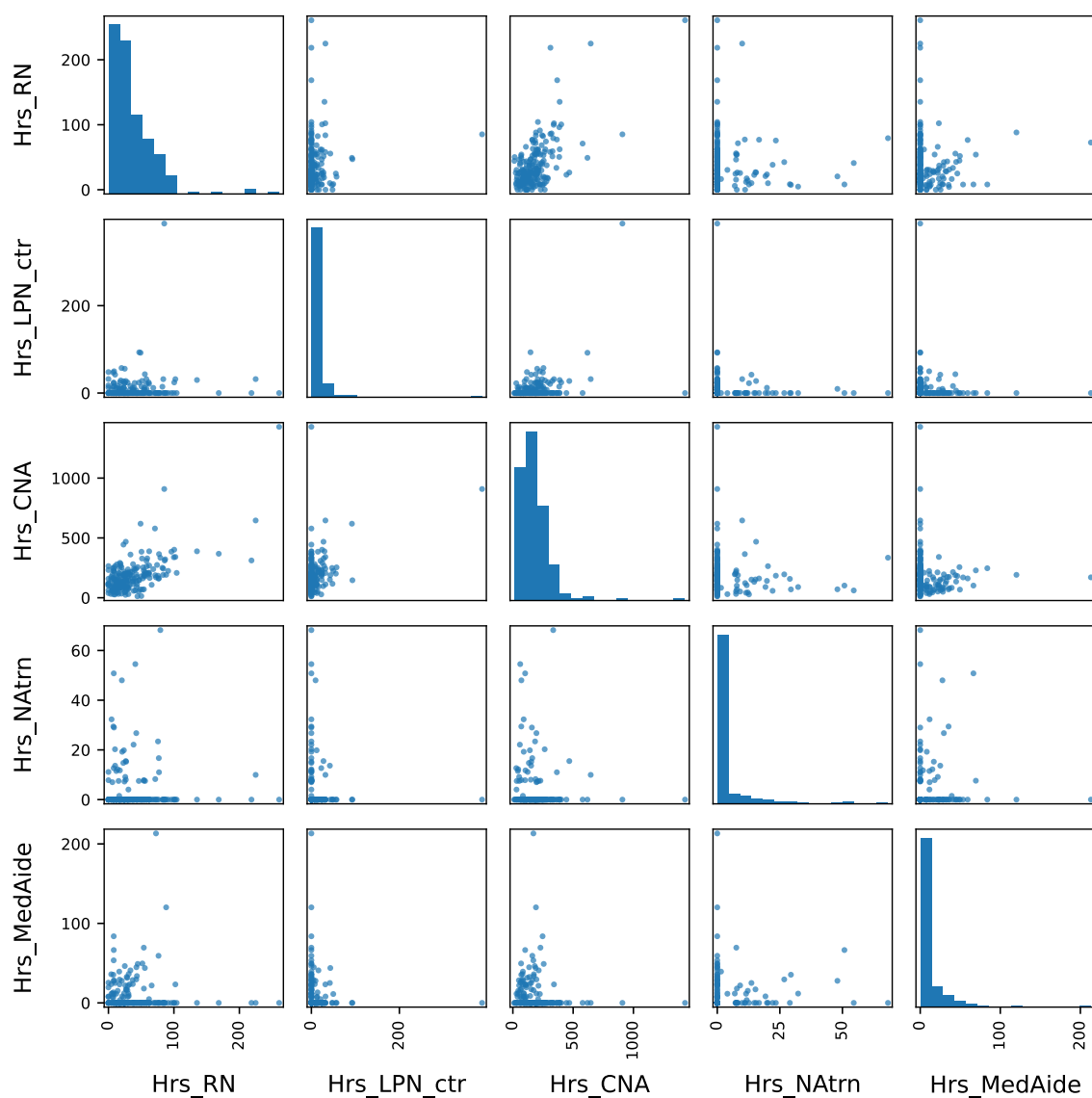



Figure 1: Scatter matrix of nursing worker working hours

Compare groups

Feature engineer

Join geographical data

Join seasonal data

Analyze geography

Analyze seasonality

Model

Extra visualizations

Sparklines

```
# TODO: pivot on day

data_pivoted = data.pivot_table(
    index="STATE",
    columns="WorkDate",
    values="Hrs_RN",
    aggfunc='mean'
)

# Resetting the index for easier column access
# data_pivoted.reset_index(inplace=True)
data_pivoted.head()
```

```
WorkDate
Date
STATE
AL 6.6210898634335120569317503000578090536393332.5253939736333704060404053708035736513034000
AK 16.903579206040816840379627156887430184089713908.314524008009490186292965003474256405397947085
AR 1.224450424073456534823594271433401244873546083.38288129981368255871924120706122615253998571
AZ 3.9386736870361853619823031652883935257818494941.056390897746627680386674046599432260202188417
CA 29.643293224833316235268428279272246332500463080734.4604748094738812418308685798089500562092820675
```

```
# (
#     GT(data_pivoted, rowname_col="STATE")
#     .fmt_nanoplot(
#         columns=data_pivoted.columns[1:],
#         reference_line="mean",
#         reference_area=["min", "q1"]
#     )
#     .fmt_nanoplot(
#         columns=data_pivoted.columns[1:],
#         plot_type="bar",
#         reference_line="max",
#         reference_area=["max", "median"]
#     )
# )
```

Archive

Bibliography