File Edit View Run Kernel Tabs Settings Help

composing_plots.ipynb

Code Python 3

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

Prep Data

[2]:
```
# Read in hospital claims data
hospital_data = pd.read_csv(Path("../Resources/hospital_claims.csv"), header=0)
hospital_data.head()
```

[2]:

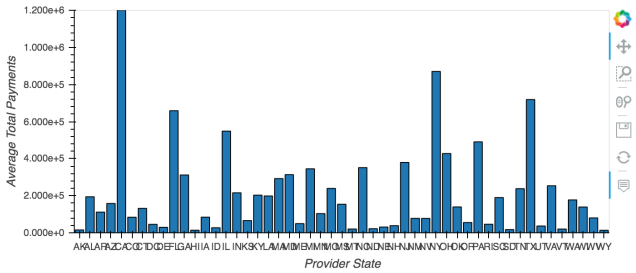
	DRG Definition	Provider Id	Provider Name	Provider Street Address	Provider City	Provider State	Provider Zip Code	Hospital Referral Region Description	Total Discharges	Average Covered Charges	Average Total Payments	Average Medicare Payments
0	039 - EXTRACRANIAL PROCEDURES W/O CC/MCC	10001	SOUTHEAST ALABAMA MEDICAL CENTER	1108 ROSS CLARK CIRCLE	DOTHAN	AL	36301	AL - Dothan	91	32963.07	5777.24	4763.73
1	039 - EXTRACRANIAL PROCEDURES W/O CC/MCC	10005	MARSHALL MEDICAL CENTER SOUTH	2505 U S HIGHWAY 431 NORTH	BOAZ	AL	35957	AL - Birmingham	14	15131.85	5787.57	4976.71
2	039 - EXTRACRANIAL PROCEDURES W/O CC/MCC	10006	ELIZA COFFEE MEMORIAL HOSPITAL	205 MARENGO STREET	FLORENCE	AL	35631	AL - Birmingham	24	37560.37	5434.95	4453.79
3	039 - EXTRACRANIAL PROCEDURES W/O CC/MCC	10011	ST VINCENT'S EAST	50 MEDICAL PARK EAST DRIVE	BIRMINGHAM	AL	35235	AL - Birmingham	25	13998.28	5417.56	4129.16
4	039 - EXTRACRANIAL PROCEDURES W/O CC/MCC	10016	SHELBY BAPTIST MEDICAL CENTER	1000 FIRST STREET NORTH	ALABASTER	AL	35007	AL - Birmingham	18	31633.27	5658.33	4851.44

[3]:
```
# Slice the data
procedure_699_charges = hospital_data[
    hospital_data["DRG Definition"] == "638 - DIABETES W CC"
]

# Select data just for NJ to get a baseline for Diabetes costs
procedure_NJ_charges = procedure_699_charges.loc[
    hospital_data["Provider State"] == "NJ"
]

# Select Average Medicare Payments and Provider ID Series
procedure_medicare_charges = procedure_NJ_charges[
    ["Average Medicare Payments", "Provider Id"]
]

# Group data by state and average total payments. and then sum
payment_by_state = procedure_699_charges[["Average Total Payments", "Provider State"]]
total_payment_by_state = payment_by_state.groupby("Provider State").sum()
```

Create plot objects to compose

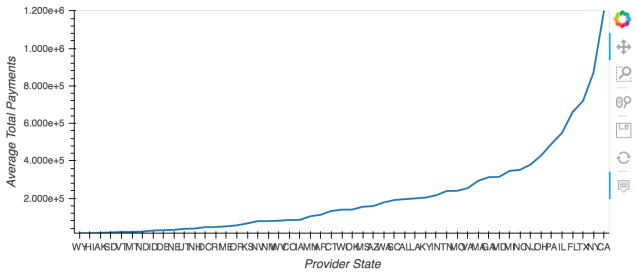
[4]:
```
# Plot data using hvplot.bar
total_payment_by_state.hvplot.bar()
```

[4]:



[5]:
```
# Sort data values
sorted_data = total_payment_by_state.sort_values("Average Total Payments")
sorted_data.hvplot()
```

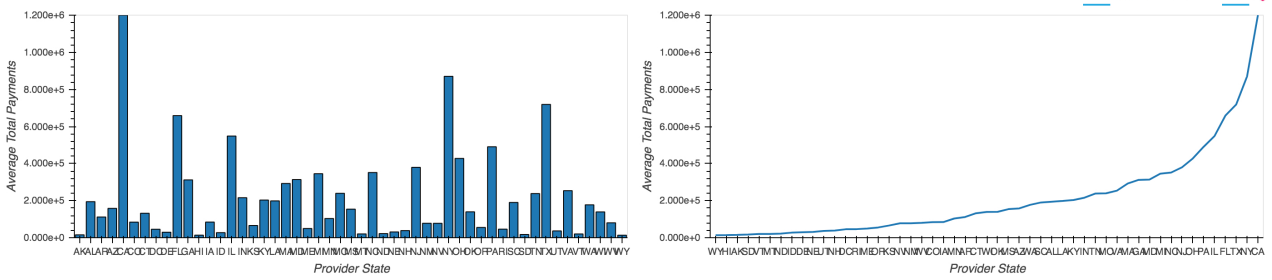
[5]:



Use compose operators

[6]:
```
# Compose plots using + operator
total_payment_by_state.hvplot.bar() + sorted_data.hvplot.line()
```

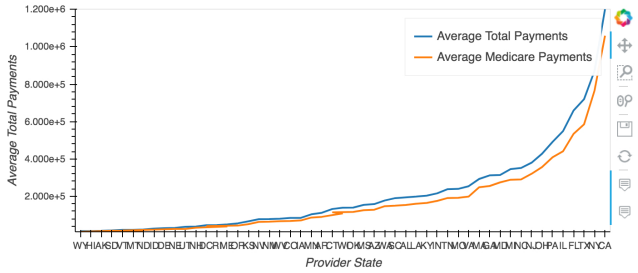
[6]:



[7]:
```
# Compose plots using * operator
payment_by_state_med = procedure_699_charges[
```

```
    ["Average Medicare Payments", "Provider State"]
]
total_payment_by_state_med = payment_by_state_med.groupby("Provider State").sum()
sorted_data_med = total_payment_by_state_med.sort_values("Average Medicare Payments")
sorted_data.hvplot(label="Average Total Payments") * sorted_data_med.hvplot(
    label="Average Medicare Payments"
)
```

[7]:



[8]:

```
# Overlay plots of different type using * operator
sorted_data.hvplot.line(label="Average Total Payments") * sorted_data_med.hvplot.bar(
    label="Average Medicare Payments"
)
```

[8]: