Avanth Pakanati
2/16/25

I ran into issues during the data cleaning aspect of this assignment and was able to produce a cleaned dataset, but it is missing many values and was not done correctly. For this reason, in this PDF I have provided the code and attempted answers for each of the questions. Because there were issues with my data cleaning and missing values, much of the code does not run. However, once I fix the data cleaning for the next submission, this code will run.

Question 1

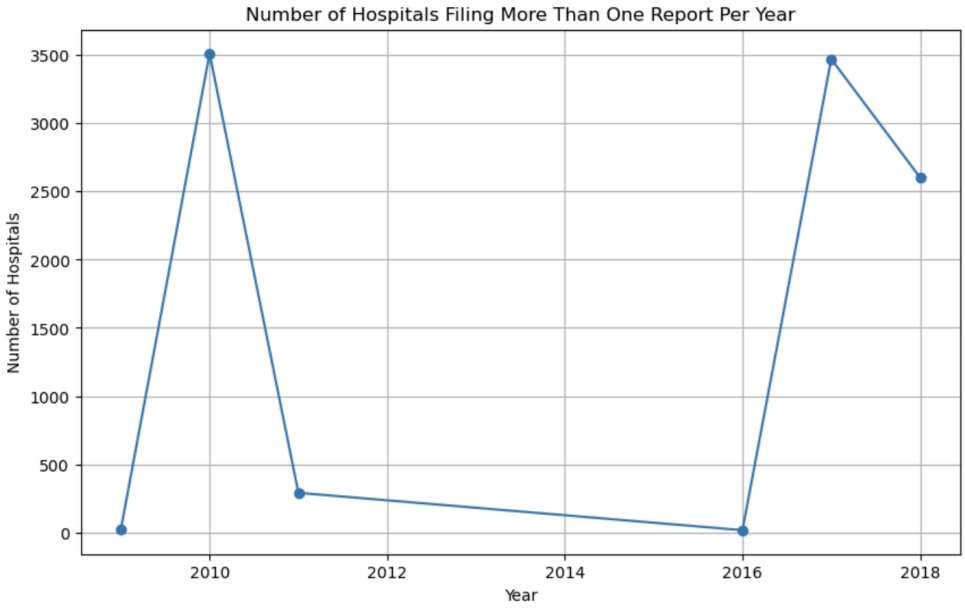
:

```python
#1
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


#reading data
hcris = pd.read_csv('../../data/Output/HCRIS.csv')
# Filter data to get hospitals with more than one report in the same year
duplicate_hospitals = hcris[hcris.duplicated(subset=['provider_number', 'fyear'], keep=False)]

# Count unique hospitals that filed more than one report per year
hospitals_over_time = duplicate_hospitals.groupby('fyear')['provider_number'].nunique()

# Plot the line graph
plt.figure(figsize=(10, 6))
plt.plot(hospitals_over_time.index, hospitals_over_time.values, marker='o')
plt.title("Number of Hospitals Filing More Than One Report Per Year")
plt.xlabel("Year")
plt.ylabel("Number of Hospitals")
plt.grid(True)
plt.show()
```



Question 2:

```
#Question 2

#Removing duplicate reports
unique_hospitals = hcris.drop_duplicates(subset=['provider_number', 'fyear'], keep='first')

#Count of number of unique hospital IDs
unique_hospital_count = unique_hospitals['provider_number'].nunique()

print(f"Number of Unique Hospital IDs: {unique_hospital_count}")
```

[4] ✓ 0.0s Python

```
Number of unique hospital IDs: 6383
```

Question 3:

```python
#Question 3
hcris['tot_charges'] = pd.to_numeric(hcris['tot_charges'], errors='coerce')

#Remove rows with missing charges or years
charges_by_year = hcris[['fyear', 'tot_charges']].dropna()

# Plot violin plot
plt.figure(figsize=(12, 6))
sns.violinplot(x='fyear', y='tot_charges', data=charges_by_year)
plt.title("Total Charges by Year")
plt.xlabel("Year")
plt.ylabel("Total Charges")
plt.xticks(rotation=45)
plt.grid(True, which='both', linestyle='--', linewidth=0.5)
plt.show()
```

[11] ⊗ 0.0s Python

```
[11] ⊗ 0.0s                                                    Python
...
    ValueError                                Traceback (most recent call last)
    Input In [11], in <cell line: 9>()
          7 # Plot violin plot
          8 plt.figure(figsize=(12, 6))
    ----> 9 sns.violinplot(x='fyear', y='tot_charges', data=charges_by_year)
         10 plt.title("Distribution of Total Charges by Year")
         11 plt.xlabel("Year")

    File ~/anaconda/lib/python3.9/site-packages/seaborn/_decorators.py:46, in _deprecate_positional_args.<
         36     warnings.warn(
         37         "Pass the following variable{} as {}keyword arg{}: {}. "
         38         "From version 0.12, the only valid positional argument "
       (...)
         43         FutureWarning
         44     )
         45 kwargs.update({k: arg for k, arg in zip(sig.parameters, args)})
    ---> 46 return f(**kwargs)

    File ~/anaconda/lib/python3.9/site-packages/seaborn/categorical.py:2400, in violinplot(x, y, hue, data
       2388 @_deprecate_positional_args
       2389 def violinplot(
       2390     *,
       (...)
       2397     ax=None, **kwargs,
    ...
    --> 319 lum = min(light_vals) * .6
        320 gray = mpl.colors.rgb2hex((lum, lum, lum))
        322 # Assign object attributes

    ValueError: min() arg is an empty sequence
    Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

...  <Figure size 1200x600 with 0 Axes>
```

Question 4:

```
#Question 4
numeric_columns = [
    'tot_discounts', 'tot_charges', 'ip_charges', 'icu_charges', 'ancillary_charges',
    'tot_mcare_payment', 'tot_discharges', 'mcare_discharges'
]
hcris[numeric_columns] = hcris[numeric_columns].apply(pd.to_numeric, errors='coerce')

#Calculating estimated price (using formula)
hcris['discount_factor'] = 1 - hcris['tot_discounts'] / hcris['tot_charges']
hcris['price_num'] = (
    (hcris['ip_charges'] + hcris['icu_charges'] + hcris['ancillary_charges'])
    * hcris['discount_factor']
    - hcris['tot_mcare_payment']
)
hcris['price_denom'] = hcris['tot_discharges'] - hcris['mcare_discharges']
hcris['price'] = hcris['price_num'] / hcris['price_denom']

#remove negatives and outliers
hcris_df = hcris[(hcris['price'] > 0)]

#Plot violin plot
plt.figure(figsize=(12, 6))
sns.violinplot(x='fyear', y='price', data=hcris)
plt.title("Estimated Prices by Year")
plt.xlabel("Year")
plt.ylabel("Estimated Price")
plt.xticks(rotation=45)
plt.grid(True, which='both', linestyle='--', linewidth=0.5)
plt.show()
```

Question 5:

```python
#Question 5
hcris_2012 = hcris_df[hcris_df['fyear'] == 2012]

#Calculating estimated price for 2012
hcris_2012['discount_factor'] = 1 - hcris_2012['tot_discounts'] / hcris_2012['tot_charges']
hcris_2012['price_num'] = (
    (hcris_2012['ip_charges'] + hcris_2012['icu_charges'] + hcris_2012['ancillary_charges'])
    * hcris_2012['discount_factor'] - hcris_2012['tot_mcare_payment']
)
hcris_2012['price_denom'] = hcris_2012['tot_discharges'] - hcris_2012['mcare_discharges']
hcris_2012['price'] = hcris_2012['price_num'] / hcris_2012['price_denom']

# NA payments
hcris_2012['hvbp_payment'] = hcris_2012['hvbp_payment'].fillna(0)
hcris_2012['hrrp_payment'] = hcris_2012['hrrp_payment'].fillna(0).abs()

#Defining penalty
hcris_2012['penalty'] = (hcris_2012['hvbp_payment'] - hcris_2012['hrrp_payment'] < 0).astype(int)

# Calculate average price for penalized vs non-penalized hospitals
mean_penalized = round(hcris_2012.loc[hcris_2012['penalty'] == 1, 'price'].mean(), 2)
mean_non_penalized = round(hcris_2012.loc[hcris_2012['penalty'] == 0, 'price'].mean(), 2)

print(f"Average price for penalized hospitals: {mean_penalized}")
print(f"Average price for non-penalized hospitals: {mean_non_penalized}")
```

0.0s Python

```
Average price for penalized hospitals: nan
Average price for non-penalized hospitals: nan
```

Question 6:

```python
#Question 6
hcris_2012['beds_quartile'] = pd.qcut(hcris_2012['beds'], 4, labels=[1, 2, 3, 4])

# Create indicator variables for each quartile
for i in range(1, 5):
    hcris_2012[f'quartile_{i}'] = (hcris_2012['beds_quartile'] == i).astype(int)


# Calculate average price for treated and control groups within each quartile
Avg_per_group = []
for i in range(1, 5):
    treated_mean = hcris_2012.loc[(hcris_2012[f'quartile_{i}'] == 1) & (hcris_2012['penalty'] == 1), 'price'].mean()
    control_mean = hcris_2012.loc[(hcris_2012[f'quartile_{i}'] == 1) & (hcris_2012['penalty'] == 0), 'price'].mean()
```

0.0s

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Input In [31], in <cell line: 2>()
      1 #Question 6
----> 2 hcris_2012['beds_quartile'] = pd.qcut(hcris_2012['beds'], 4, labels=[1, 2, 3, 4])
      4 # Create indicator variables for each quartile
      5 for i in range(1, 5):

File ~/anaconda/lib/python3.9/site-packages/pandas/core/reshape/tile.py:376, in qcut(x, q, labels, retbins, precision, duplicates)
    374 x_np = np.asarray(x)
    375 x_np = x_np[~np.isnan(x_np)]
--> 376 bins = np.quantile(x_np, quantiles)
    378 fac, bins = _bins_to_cuts(
    379     x,
    380     bins,
   (...)
    385     duplicates=duplicates,
    386 )
    388 return _postprocess_for_cut(fac, bins, retbins, dtype, original)

File <__array_function__ internals>:5, in quantile(*args, **kwargs)

File ~/anaconda/lib/python3.9/site-packages/numpy/lib/function_base.py:3979, in quantile(a, q, axis, out, overwrite_input, interpolation, ke
   3977 if not _quantile_is_valid(q):
   3978     raise ValueError("Quantiles must be in the range [0, 1]")
...
   4101     ap.partition(concatenate((
   4102         indices_below.ravel(), indices_above.ravel()
   4103     )), axis=0)

IndexError: index -1 is out of bounds for axis 0 with size 0
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Question 7 (Nearest Neighbor Match W /inverse variance distance):

```python
#Question 7 - Nearest Neighbor match
from sklearn.neighbors import NearestNeighbors

#Nearest neighbor match
results = []
for q in [1, 2, 3, 4]:
    subset = hcris_2012[hcris_2012['beds_quartile'] == q]
    treated = subset[subset['penalty'] == 1]
    control = subset[subset['penalty'] == 0]

    if treated.empty or control.empty:
        results.append({'Quartile': q, 'ATE': np.nan})
        continue
```

⊗ 0.7s

```
KeyError                                  Traceback (most recent call last)
File ~/anaconda/lib/python3.9/site-packages/pandas/core/indexes/base.py:3629, in Index.get_loc(self, k
   3628 try:
-> 3629     return self._engine.get_loc(casted_key)
   3630 except KeyError as err:

File ~/anaconda/lib/python3.9/site-packages/pandas/_libs/index.pyx:136, in pandas._libs.index.IndexEng

File ~/anaconda/lib/python3.9/site-packages/pandas/_libs/index.pyx:163, in pandas._libs.index.IndexEng

File pandas/_libs/hashtable_class_helper.pxi:5198, in pandas._libs.hashtable.PyObjectHashTable.get_ite

File pandas/_libs/hashtable_class_helper.pxi:5206, in pandas._libs.hashtable.PyObjectHashTable.get_ite

KeyError: 'beds_quartile'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Input In [33], in <cell line: 6>()
      5 results = []
      6 for q in [1, 2, 3, 4]:
----> 7     subset = hcris_2012[hcris_2012['beds_quartile'] == q]
      8     treated = subset[subset['penalty'] == 1]
...
   3634     #  InvalidIndexError. Otherwise we fall through and re-raise
   3635     #  the TypeError.
   3636     self._check_indexing_error(key)

KeyError: 'beds_quartile'
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Question 7 (Nearest Neighbor Match W/ Mahalanobis distance):

```python
# Question 7 - Mahalanobis distance Nearest Neighbor Match

results = []
for q in [1, 2, 3, 4]:
    subset = hcris_2012[hcris_2012['beds_quartile'] == q]
    treated = subset[subset['penalty'] == 1]
    control = subset[subset['penalty'] == 0]

    if treated.empty or control.empty:
        results.append({'Quartile': q, 'ATE': np.nan})
        continue

    covariates = ['beds', 'mcare_discharges', 'ip_charges', 'tot_mcare_payment']
    treated_cov = treated[covariates].values
    control_cov = control[covariates].values

    # Calculate the covariance matrix for Mahalanobis distance
    cov_matrix = np.cov(np.vstack([treated_cov, control_cov]).T)
    inv_cov_matrix = np.linalg.inv(cov_matrix)

    treated_prices = []
    control_prices = []

    for i, treated_row in enumerate(treated_cov):
        distances = [distance.mahalanobis(treated_row, control_row, inv_cov_matrix) for control_row in control_cov]
        nearest_idx = np.argmin(distances)

        treated_prices.append(treated.iloc[i]['price'])
        control_prices.append(control.iloc[nearest_idx]['price'])

    ate = np.mean(np.array(treated_prices) - np.array(control_prices))
    results.append({'Quartile': q, 'ATE': round(ate, 2)})
```

0.0s

```
KeyError                                  Traceback (most recent call last)
File ~/anaconda/lib/python3.9/site-packages/pandas/core/indexes/base.py:3629, in Index.get_loc(self, key, method, tolerance)
   3628 try:
-> 3629     return self._engine.get_loc(casted_key)
   3630 except KeyError as err:

File ~/anaconda/lib/python3.9/site-packages/pandas/_libs/index.pyx:136, in pandas._libs.index.IndexEngine.get_loc()

File ~/anaconda/lib/python3.9/site-packages/pandas/_libs/index.pyx:163, in pandas._libs.index.IndexEngine.get_loc()

File pandas/_libs/hashtable_class_helper.pxi:5198, in pandas._libs.hashtable.PyObjectHashTable.get_item()

File pandas/_libs/hashtable_class_helper.pxi:5206, in pandas._libs.hashtable.PyObjectHashTable.get_item()

KeyError: 'beds_quartile'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Input In [34], in <cell line: 4>()
      3 results = []
      4 for q in [1, 2, 3, 4]:
----> 5     subset = hcris_2012[hcris_2012['beds_quartile'] == q]
      6     treated = subset[subset['penalty'] == 1]
...
   3634     #  InvalidIndexError. Otherwise we fall through and re-raise
   3635     #  the TypeError.
   3636     self._check_indexing_error(key)

KeyError: 'beds_quartile'
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Question 7 (Inverse propensity weighting):

```python
# Question 7 - Inverse Propensity Weighting
results = []
for q in [1, 2, 3, 4]:
    subset = hcris_2012[hcris_2012['beds_quartile'] == q]

    # Estimate propensity score
    covariates = ['beds', 'mcare_discharges', 'ip_charges', 'tot_mcare_payment']
    X = sm.add_constant(subset[covariates])
    y = subset['penalty']
    logit_model = sm.Logit(y, X).fit(disp=0)
    subset['propensity_score'] = logit_model.predict(X)
```

0.0s

```
KeyError                                  Traceback (most recent call last)
File ~/anaconda/lib/python3.9/site-packages/pandas/core/indexes/base.py:3629, in Index.get_loc(self, k
   3628 try:
-> 3629     return self._engine.get_loc(casted_key)
   3630 except KeyError as err:

File ~/anaconda/lib/python3.9/site-packages/pandas/_libs/index.pyx:136, in pandas._libs.index.IndexEng

File ~/anaconda/lib/python3.9/site-packages/pandas/_libs/index.pyx:163, in pandas._libs.index.IndexEng

File pandas/_libs/hashtable_class_helper.pxi:5198, in pandas._libs.hashtable.PyObjectHashTable.get_ite

File pandas/_libs/hashtable_class_helper.pxi:5206, in pandas._libs.hashtable.PyObjectHashTable.get_ite

KeyError: 'beds_quartile'

The above exception was the direct cause of the following exception:

KeyError                                  Traceback (most recent call last)
Input In [35], in <cell line: 3>()
      2 results = []
      3 for q in [1, 2, 3, 4]:
----> 4     subset = hcris_2012[hcris_2012['beds_quartile'] == q]
      6     # Estimate propensity score
...
   3634     #  InvalidIndexError. Otherwise we fall through and re-raise
   3635     #  the TypeError.
   3636     self._check_indexing_error(key)

KeyError: 'beds_quartile'
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Question 8:
Unable to respond, because my data cleaning was done improperly.

Question 9:
Unable to respond, because my data cleaning was done improperly.

Question 10:
Overall, I had a difficult experience working with this data, but learned a lot of valuable lessons along the way. When I have worked with data in other classes in the past, I have always been given a nice, clean dataset and am able to begin analyzing the dataset easily. When working with this data, the bulk of the work is cleaning and merging the data, which I definitely found to be challenging. However, I definitely recognize how important these skills are. When working with data in the real world, it is rarely ever clean, so I am really happy to be getting experience with this.