0.0.1 Question 1c

Linear models are sensitive to multicollinearity among the columns of the design matrix. So, let's determine the extent of multicollinearity in the college rankings dataset.

In the following cell, we provide you with `arranged_cleaned_college_data`, which is a copy of `cleaned_college_data` containing just a subset of the columns arranged in a particular order.

Create a visualization that shows the pairwise correlation between each combination of columns in `arranged_cleaned_college_data`.

- For 2-D visualizations, consider the `sns.heatmap()` documentation.
- For full credit, title your plot, and set `annot=True`. This makes the plot easier to interpret.
- You may find your plot easier to read with a different color scale. For example, try including `cmap="coolwarm"` inside of `sns.heatmap()`.

Hint: Your plot should show 10×10 values corresponding to the pairwise correlations of the selected columns in `arranged_cleaned_college_data`:

```
['Overall Score (0-100)',
'Peer Assessment Score (1-5)',
'Predicted 6yr graduation rate',
'Actual 6yr graduation rate',
'Graduation and retention rank',
'Student Excellence rank',
'Acceptance rate',
'Financial resources rank',
'Is Public',
'Is Private']
```

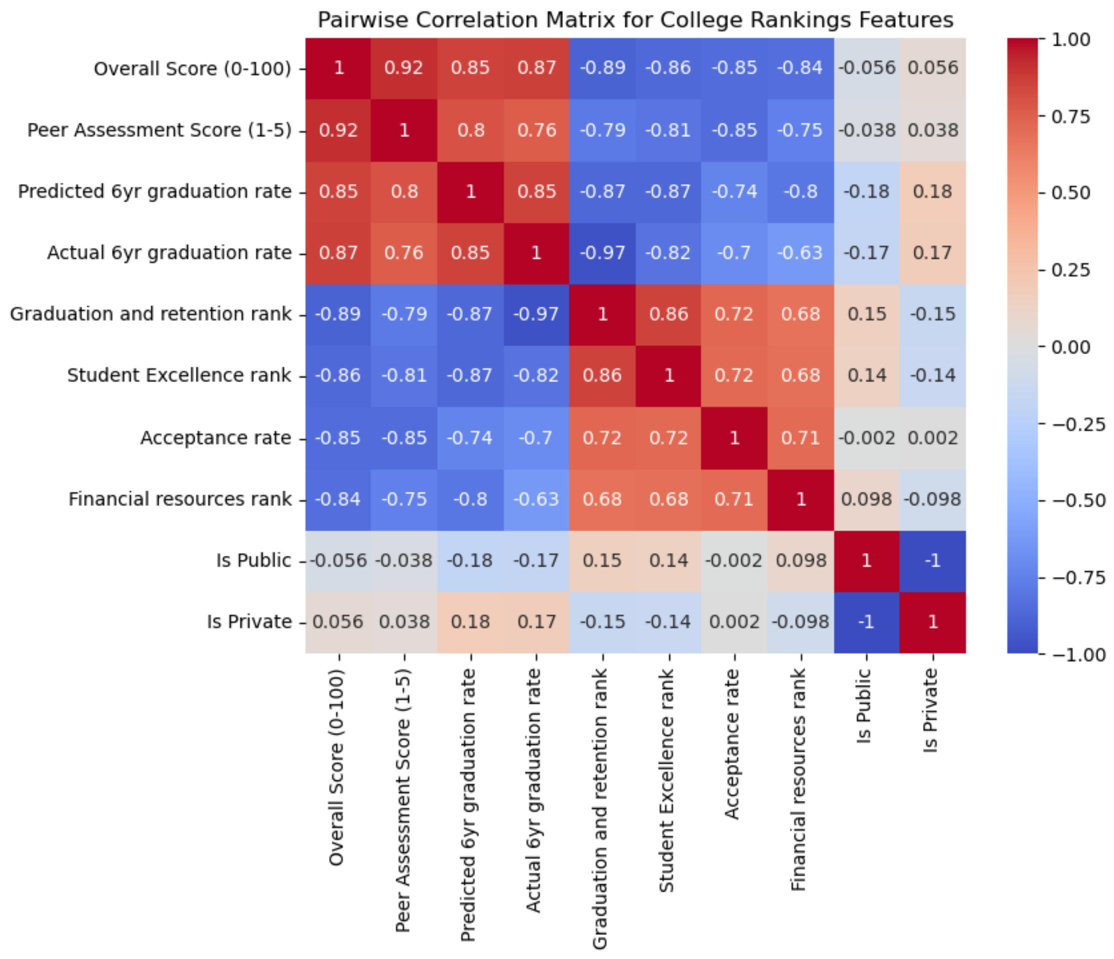
```
In [24]: # Running this cell helps you get a subset of cleaned_college_data with the above columns
         arranged_cleaned_college_data = cleaned_college_data[
             ['Overall Score (0-100)',
              'Peer Assessment Score (1-5)',
              'Predicted 6yr graduation rate',
              'Actual 6yr graduation rate',
              'Graduation and retention rank',
              'Student Excellence rank',
              'Acceptance rate',
              'Financial resources rank',
              'Is Public',
        'Is Private']
    ]
```

Your output figure should look similar to the following example:

```
In [25]: corr_matrix = arranged_cleaned_college_data.corr()
         fig, ax = plt.subplots(figsize=(8, 6))
         sns.heatmap(corr_matrix, ax=ax, annot=True, cmap="coolwarm")

         ax.set_title("Pairwise Correlation Matrix for College Rankings Features")

         plt.show()
```



	Overall Score (0-100)	Peer Assessment Score (1-5)	Predicted 6yr graduation rate	Actual 6yr graduation rate	Graduation and retention rank	Student Excellence rank	Acceptance rate	Financial resources rank	Is Public	Is Private
Overall Score (0-100)	1	0.92	0.85	0.87	-0.89	-0.86	-0.85	-0.84	-0.056	0.056
Peer Assessment Score (1-5)	0.92	1	0.8	0.76	-0.79	-0.81	-0.85	-0.75	-0.038	0.038
Predicted 6yr graduation rate	0.85	0.8	1	0.85	-0.87	-0.87	-0.74	-0.8	-0.18	0.18
Actual 6yr graduation rate	0.87	0.76	0.85	1	-0.97	-0.82	-0.7	-0.63	-0.17	0.17
Graduation and retention rank	-0.89	-0.79	-0.87	-0.97	1	0.86	0.72	0.68	0.15	-0.15
Student Excellence rank	-0.86	-0.81	-0.87	-0.82	0.86	1	0.72	0.68	0.14	-0.14
Acceptance rate	-0.85	-0.85	-0.74	-0.7	0.72	0.72	1	0.71	-0.002	0.002
Financial resources rank	-0.84	-0.75	-0.8	-0.63	0.68	0.68	0.71	1	0.098	-0.098
Is Public	-0.056	-0.038	-0.18	-0.17	0.15	0.14	-0.002	0.098	1	-1
Is Private	0.056	0.038	0.18	0.17	-0.15	-0.14	0.002	-0.098	-1	1

0.0.2 Question 1d

Do you notice any patterns in the plot from part (c)? What might explain these patterns? Comment and hypothesize on at least two patterns you notice.

Here are some example questions to ponder:

1. Why do some feature-pairs have correlations of ± 1? Is this a problem?
2. What does the correlation between pairs of features (i.e., graduation-related statistics) look like? Is the magnitude of any of the correlations problemtatically close to 1?
3. Are any features particularly strong predictors of the outcome? If so, why do you think this is the case?
4. Do any features seem potentially redundant? In other words, do you suspect that any features provide similar information about the outcome as other features?

When I looked at the heatmap, one thing that stood out right away was how strongly some of the graduation-related stats were correlated — like predicted graduation rate, actual graduation rate, and peer assessment score. They all had really high positive correlations with the overall score, which makes sense. If a school is good at graduating students and is well-reviewed by peers, it's probably going to have a higher overall ranking. That said, those features are also really similar to each other, so there might be some redundancy going on.

Another thing I noticed was that the rank-related features, like "graduation and retention rank" and "student excellence rank," had strong negative correlations with the overall score. At first, that might seem odd, but then I remembered that in rankings, lower is better — so a school ranked #1 in retention would likely have a high overall score. It just reflects the fact that better-ranked schools perform better overall.

One last thing: "Is Public" and "Is Private" were perfectly negatively correlated, which makes sense since they're basically just two sides of the same coin. You'd probably want to drop one of them when building a model to avoid issues from perfect redundancy.

0.0.3 Question 1e

If we tried to fit a linear regression model with an intercept term using all features in `cleaned_college_data`, we might run into some problems when fitting our model. The Data 100 staff suggests that we perform the following operation to the `DataFrame` before fitting a model:

```
In [26]: cleaned_college_data
```

```
Out[26]:      Overall Score (0-100)  Peer Assessment Score (1-5)  \
         0                      100                          4.7
         1                       98                          4.6
         2                       94                          4.6
         3                       93                          4.5
         4                       92                          4.5
         ..                       …                            …
         162                     42                          2.4
         163                     41                          2.4
         164                     40                          2.9
         165                     40                          2.4
         166                     40                          2.6

              Graduation and retention rank  Predicted 6yr graduation rate  \
         0                                1                           95.0
         1                                2                           92.0
         2                                2                           92.0
         3                                6                           92.0
         4                               14                           92.0
         ..                               …                              …
         162                            153                           63.0
         163                            163                           69.0
         164                            174                           61.0
         165                            191                           71.0
         166                            130                           69.0

              Actual 6yr graduation rate  Pell gradrate  Social Mobility Rank  \
         0                          96.0          96.00                  98.0
         1                          95.0          93.00                  38.0
         2                          97.0          98.00                 135.0
         3                          94.0          91.00                  80.0
         4                          94.0          95.00                  98.0
         ..                           …              …                     …
         162                        60.0          44.00                 122.0
         163                        56.0          71.87                 106.0
         164                        50.0          42.00                  80.0
         165                        32.0          33.00                  67.0
         166                        68.0          61.00                 150.0
```

```
     Percent of classes under 20  Percent of classes of 50 or more students  \
0                           84.0                                        1.0
1                           80.0                                        2.0
2                           76.0                                        1.0
3                           78.0                                        0.2
4                           71.0                                        0.2
..                           …                                        …
162                         78.0                                        0.0
163                         72.0                                        1.0
164                         73.0                                        0.0
165                         87.0                                        0.0
166                         49.0                                        1.0

     Student Excellence rank  …  Financial resources rank  \
0                          1  …                       2.0
1                          5  …                       5.0
2                          3  …                       6.0
3                          3  …                       8.0
4                          8  …                       8.0
..                       …  …                       …
162                      139  …                     161.0
163                      101  …                      85.0
164                      105  …                     194.0
165                      161  …                      19.0
166                      161  …                     173.0

     Average alumni giving rate  Pell gradrate_missing  \
0                        47.000                  False
1                        42.000                  False
2                        31.000                  False
3                        23.000                  False
4                        40.000                  False
..                          …                  …
162                      20.273                  False
163                       9.000                   True
164                       6.000                  False
165                      33.000                  False
166                      19.000                  False

     Social Mobility Rank_missing  Percent of classes under 20_missing  \
0                           False                                False
1                           False                                False
2                           False                                False
3                           False                                False
4                           False                                False
..                            …                                …
162                         False                                False
163                         False                                False
164                         False                                False
165                         False                                False
166                         False                                False

     Percent of classes of 50 or more students_missing  \
0                                                 False
1                                                 False
2                                                 False
3                                                 False
4                                                 False
..                                                  …
162                                               False
163                                               False
164                                               False
165                                               False
166                                               False

     First year students in top 10% of high school class_missing  \
0                                                 False
1                                                 False
2                                                 False
3                                                 False
4                                                 False
..                                                  …
162                                               False
163                                               False
164                                               False
165                                                True
166                                               False

     Average alumni giving rate_missing  Is Public  Is Private
0                                 False          0           1
1                                 False          0           1
2                                 False          0           1
3                                 False          0           1
4                                 False          0           1
..                                  …          …         …
162                                True          0           1
163                               False          0           1
164                               False          0           1
165                               False          0           1
166                               False          0           1

[167 rows x 22 columns]
```

```
In [28]: # You must run this cell to achieve pass the public tests for later questions.
         cleaned_college_data = cleaned_college_data.drop('Is Private', axis=1)
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[28], line 2
      1 # You must run this cell to achieve pass the public tests for later questions.
----> 2 cleaned_college_data = cleaned_college_data.drop('Is Private', axis=1)

File /srv/conda/envs/notebook/lib/python3.11/site-packages/pandas/core/frame.py:5258, i . DataFrame.drc
   5110 def drop(
   5111     self,
   5112     labels: IndexLabel = None,
   (…)
   5119     errors: IgnoreRaise = "raise",
   5120 ) -> DataFrame | None:
   5121     """
   5122     Drop specified labels from rows or columns.
   5123 
   (…)
   5256             weight  1.0     0.8
   5257     """
-> 5258     return super().drop(
   5259         labels=labels,
   5260         axis=axis,
   5261         index=index,
   5262         columns=columns,
   5263         level=level,
   5264         inplace=inplace,
   5265         errors=errors,
   5266     )

File /srv/conda/envs/notebook/lib/python3.11/site-packages/pandas/core/generic.py:4549, in NDFrame.dro
   4547 for axis, labels in axes.items():
   4548     if labels is not None:
-> 4549         obj = obj._drop_axis(labels, axis, level=level, errors=errors)
   4551 if inplace:
   4552     self._update_inplace(obj)

File /srv/conda/envs/notebook/lib/python3.11/site-packages/pandas/core/generic.py:4591, in NDFrame._dr
   4589         new_axis = axis.drop(labels, level=level, errors=errors)
   4590     else:
-> 4591         new_axis = axis.drop(labels, errors=errors)
   4592     indexer = axis.get_indexer(new_axis)
   4594 # Case for non-unique axis
   4595 else:

File /srv/conda/envs/notebook/lib/python3.11/site-packages/pandas/core/indexes/base.py:6696, in Index.
   6694 if mask.any():
   6695     if errors != "ignore":
-> 6696         raise KeyError(f"{list(labels[mask])} not found in axis")
   6697     indexer = indexer[~mask]
   6698 return self.delete(indexer)

KeyError: "['Is Private'] not found in axis"
```

Describe the reasoning behind this operation. What problem(s) do we avoid by removing the `Is Private` column from the model fitting process?

We remove the Is Private column because it's perfectly negatively correlated with the Is Public column. Since every college is either public or private, including both in a linear regression model would create what's called perfect multicollinearity — basically, one column is completely predictable from the other.

This causes problems during model fitting because linear regression tries to assign a unique coefficient to each feature. But if two features are perfectly dependent on each other, the math behind the model can't figure out how to separate their individual effects — leading to unstable or meaningless coefficient values.

By removing one of the two (in this case, Is Private), we keep the useful information but avoid that multicollinearity issue. The model can still understand whether a school is public or private, based on the remaining column.

0.0.4 Question 2b

Let's visualize the model performance from part 2(a). Plot the following: 1. The observed values vs. the predicted values on the test set. 2. The residuals plot. Recall that for multiple linear regression, we plot the residuals against the predicted values.

In both plots, the predicted values should be on the x-axis.

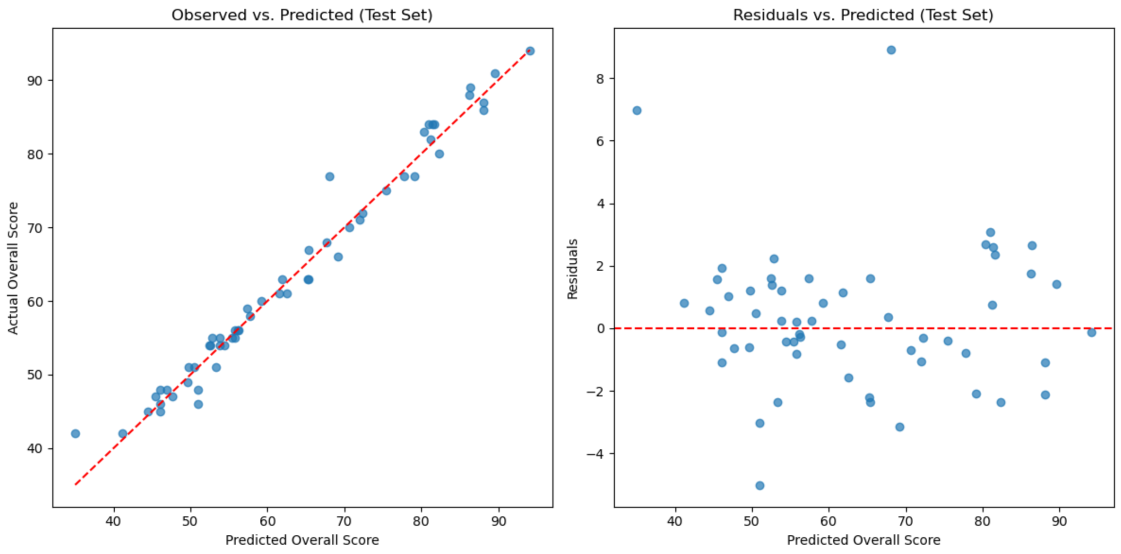
Note: * For a full-credit solution, you should use `plt.subplot()` (documentation) so that you can view both visualizations side-by-side. * The method `plt.subplot({# of rows}{# of cols}{index of plot})` sets the plottable area to the `index` of a `# rows` by `# cols` grid. * For example, `plt.subplot(121)` sets the plottable area to the first index of a 1x2 plot grid. Calling `Matplotlib` and `Seaborn` functions will plot on the first index. When you're ready to start plotting on the second index, run `plt.subplot(122)`. * **Remember to add a guiding line to both plots where $\hat{Y} = Y$, i.e., where the residual is 0.** * `plt.plot()`(documentation) and `plt.axhline()`(documentation) might be helpful here! * Make sure to add descriptive titles and axis labels. * To avoid distorted aspect ratios, ensure the limits of the x-axes for both plots are the same.

```
In [16]: plt.figure(figsize=(12,6))      # do not change this line

         plt.subplot(121)                # do not change this line
         # 1. plot observations vs. predictions
         plt.scatter(Y_test_pred, Y_test, alpha=0.7)
         plt.plot([Y_test_pred.min(), Y_test_pred.max()], [Y_test_pred.min(), Y_test_pred.max()], 'r--'
         plt.xlabel("Predicted Overall Score")
         plt.ylabel("Actual Overall Score")
         plt.title("Observed vs. Predicted (Test Set)")

         plt.subplot(122)               # do not change this line
         # 2. plot residual plot
         residuals = Y_test - Y_test_pred
         plt.scatter(Y_test_pred, residuals, alpha=0.7)
         plt.axhline(0, color='r', linestyle='--')
         plt.xlabel("Predicted Overall Score")
         plt.ylabel("Residuals")
         plt.title("Residuals vs. Predicted (Test Set)")

         plt.tight_layout()             # do not change this line
```


Observed vs. Predicted (Test Set)
Actual Overall Score
Predicted Overall Score
Residuals vs. Predicted (Test Set)
Residuals
Predicted Overall Score

0.0.5 Question 2c

Describe what the plots in part (b) indicate about this linear model. In particular, are the predictions good, and do the residuals appear uncorrelated with the predictions?

The observed vs. predicted plot shows that most predicted values are close to the actual overall scores, falling near the diagonal line Y=Y^. This indicates that the model performs reasonably well in capturing the overall trend, though there is some spread around the line, suggesting prediction errors for certain colleges.

In the residuals vs. predicted plot, the residuals appear fairly evenly scattered around zero, with no clear pattern or systematic curvature. This suggests that the residuals are approximately uncorrelated with the predictions, which is a good sign and supports the assumption of linearity and homoscedasticity in linear regression.

Overall, the model's predictions seem reasonably accurate, and the residual plot suggests that the assumptions of linear regression are not obviously violated.

Question 3d(i) Let us first interpret **Model B**, the linear regression model that use a subset of features from our dataset.

```
In [26]: display(Markdown('#### Model B: Subset of Features'))
         print_confidence_intervals(partial_feature_models, partial_feature_cis)
```

Model B: Subset of Features

Confidence Intervals:

parameter	feature name	lower	upper
θ_0	Intercept	1.769	24.2
θ_1	Peer Assessment Score (1-5)	15.399	20.148
θ_2	Acceptance rate	-0.207	-0.09

Are θ_1 and θ_2 significantly different than 0? How do you know?

Does your answer imply that the relationship between `Overall Score (0-100)`, `Peer Assessment Score (1-5)`, and `Acceptance rate` are causal? Do you think the relationships are causal? Explain.

Yes, both θ_1 (Peer Assessment Score) and θ_2 (Acceptance Rate) are significantly different from 0. We know this because: The confidence interval for θ_1 is [15.399, 20.148], which does not include 0. This implies a statistically significant positive relationship between peer assessment score and overall score. The confidence interval for θ_2 is [-0.207, -0.09], which also does not include 0. This indicates a statistically significant negative relationship between acceptance rate and overall score. No, statistical significance does not imply causality. The relationships between overall score and features like peer assessment score or acceptance rate may be driven by underlying confounding variables (e.g., prestige, historical reputation, funding levels, etc.).

In observational data like this, we can't control for all possible confounders or establish temporal order, so we should be cautious in interpreting these results as causal.

Question 3d(ii) In what situation(s) would you prefer a more compact model with just key features, like Model B? On the other hand, in what situation(s) would you want to consider many features, like in Model A? Explain your answer to both of these questions.

We'd want to use a simpler model like Model B when we care about keeping things understandable and efficient. For example, if we're presenting results to a non-technical audience, it's much easier to explain how just two features affect the outcome. It's also useful when collecting data is costly or time-consuming — fewer features means less effort and fewer resources needed.

On the other side, a more complex model like Model A makes sense when your main goal is accuracy and you have access to a lot of data. By including more features, you can often capture more of the real-world complexity and improve predictions. This is especially helpful if you're building a tool that prioritizes performance over simplicity.

0.0.6 Question 4b

Using the `simulate` function from above, we can compute the model risk, model variance, and variance-to-risk ratio of **Model B**:

```
In [29]: x_last = X.iloc[X.shape[0] - 100]
         y_last = Y.iloc[X.shape[0] - 100]

         (
             partial_feature_model_risk,
             partial_feature_model_var,
             partial_feature_model_ratio
         ) = simulate(x_last[model_b_features], y_last, partial_feature_models)

         print('Model B risk:')
         print(partial_feature_model_risk)

         print('Model B variance:')
         print(partial_feature_model_var)

         print('Model B ratio:')
         print(partial_feature_model_ratio)
```

```
Model B risk:
2.493347485161267
Model B variance:
0.27449971448852684
Model B ratio:
0.11009284350543401
```

Comment on the variance-to-risk ratio for Model B (`partial_feature_ratio`).

- Does the model variance appear to be the dominant term in the bias-variance decomposition? If not, what term(s) dominate the bias-variance decomposition?

Then, given your conclusion above, describe what operation(s) you might perform to reduce the model risk.

The variance-to-risk ratio for Model B is 0.11, which is quite low. This tells us that only a small portion of the model's total risk comes from variance. In other words, variance isn't the problem here — the bigger issue is bias.

Since bias seems to be the dominant term, that means the model is likely underfitting — it's either too simple or missing important information. To reduce the overall model risk, I'd consider adding more relevant features, using a more flexible model, or adjusting regularization if it's too strong. The goal is to help the model better capture the patterns in the data without overfitting.