PSTAT100

Background

```
# Initialize Otter
import otter
grader = otter.Notebook("hw2-seda.ipynb")
```

```
import numpy as np
import pandas as pd
import altair as alt
# disable row limit for plotting
alt.data_transformers.disable_max_rows()
# uncomment to ensure graphics display with pdf export
# alt.renderers.enable('mimetype')
```

Gender achievement gaps in education have been well-documented over the years – studies consistently find boys outperforming girls on math tests and girls outperforming boys on reading and language tests. A particularly controversial article was published in Science in 1980 arguing that this pattern was due to an 'innate' difference in ability (focusing on mathematics rather than on reading and language). Such views persisted in part because studying systematic patterns in achievement nationwide was a challenge due to differential testing standards across school districts and the general lack of availability of large-scale data.

It is only recently that data-driven research has begun to reveal socioeconomic drivers of achievement gaps. The Standford Educational Data Archive (SEDA), a publicly available database on academic achievement and educational opportunity in U.S. schools, has supported this effort. The database is part of a broader initiave aiming to improve educational opportunity by enabling researchers and policymakers to identify systemic drivers of disparity.

> SEDA includes a range of detailed data on educational conditions, contexts, and outcomes in school districts and counties across the United States. It includes measures of academic achievement and achievement gaps for school districts and counties, as well as district-level measures of racial and socioeconomic composition, racial and socioeconomic segregation patterns, and other features of the schooling system.

The database standardizes average test scores for schools 10,000 U.S. school districts relative to national standards to allow comparability between school districts and across grade levels and years. The test score data come from the U.S. Department of Education. In addition, multiple data sources (American Community Survey and Common Core of Data) are integrated to provide district-level socioeconomic and demographic information.

A study of the SEDA data published in 2018 identified the following persistent patterns across grade levels 3 - 8 and school ears from 2008 through 2015: * a consistent reading and language achievement gap favoring girls; * *no* national math achievement gap on average; and * local math achievement gaps that depend on the socioeconomic conditions of school districts. You can read about the main findings of the study in this brief NY Times article.

Below, we'll work with selected portions of the database. The full datasets can be downloaded here.

Assignment objectives

In this assignment, you'll explore achievement gaps in California school districts in 2018, reproducing the findings described in the article above on a more local scale and with the most recent SEDA data. You'll practice the following:

- review of data documentation
- assessment of sampling design and scope of inference
- data tidying operations
    - slicing and filtering
    - merging multiple data frames
    - pivoting tables
    - renaming and reordering variables
- constructing exploratory graphics and visualizing trends
- data aggregations
- narrative summary of exploratory analysis

Import and assessment of datasets

You'll work with test data and socioeconomic covariates aggregated to the school district level. These data are stored in two separate tables. Here you'll examine them and review data documentation.

Test score data

The first few rows of the test data are shown below. The columns are:

Column name	Meaning
`sedalea`	District ID
`grade`	Grade level
`stateabb`	State abbreviation
`sedaleaname`	District name
`subject`	Test subject

Column name	Meaning
`cs_mn_...`	Estimated mean test score
`cs_mnse_...`	Standard error for estimated mean test score
`totgyb_...`	Number of individual tests used to estimate the mean score

```
# import seda data
ca_main = pd.read_csv('data/ca-main.csv')
ca_cov = pd.read_csv('data/ca-cov.csv')

# preview test score data
ca_main.head(3)
```

The test score means for each district are named `cs_mn_...` with an abbreviation indicating subgroup (such as mean score for all `cs_mean_all`, for boys `cs_mean_mal`, for white students `cs_mn_wht`, and so on). Notice that these are generally small-ish: decimal numbers between -0.5 and 0.5.

These means are *estimated* from a number of individual student tests and *standardized* relative to national averages. They represent the number of standard deviations by which a district mean differs from the national average. So, for instance, the value `cs_mn_all = 0.1` indicates that the district average is estimated to be 0.1 standard deviations greater than the national average on the corresponding test and at the corresponding grade level.

Question 1: Interpreting test score values

Interpret the average math test score for all 4th grade students in Acton-Agua Dulce Unified School District (the first row of the dataset shown above).

Type your answer here, replacing this text.

Covariate data

The first few rows of the covariate data are shown below. The column information is as follows:

Column name	Meaning
`sedalea`	District ID
`grade`	Grade level
`sedaleanm`	District name
`urban`	Indicator: is the district in an urban locale?

Column name	Meaning
`suburb`	Indicator: is the district in a suburban locale?
`town`	Indicator: is the district in a town locale?
`rural`	Indicator: is the district in a rural locale?
`locale`	Description of district locale
Remaining variables	Demographic and socioeconomic measures

```
ca_cov.head(3)
```

You will only be working with a handful of the demographic and socioeconomic measures, so you can put off getting acquainted with those until selecting a subset of variables.

Question 2: Data semantics

In the non-public data, observational units are students – test scores are measured for each student. However, in the SEDA data you've imported, scores are *aggregated* to the district level by grade. Let's regard estimated test score means for each grade as distinct variables, so that an observation consists in a set of estimated means for different grade levels and groups. In this view, what are the observational units in the test score dataset? Are they the same or different for the covariate dataset?

Type your answer here, replacing this text.

Question 3: Sample sizes

How many observational units are in each dataset? Count the number of units in the test dataset and the number of units in the covariate dataset separately. Store the values as `ca_cov_units` and `ca_main_units`, respectively.

(*Hint*: use `.nunique()`.)

```
ca_cov_units = ...
ca_main_units = ...

print('units in covariate data: ', ca_cov_units)
print('units in test score data: ', ca_main_units)
```

```
grader.check("q3")
```

Question 4: Sample characteristics and scope of inference

Answer the questions below about the sampling design in a short paragraph. You do not need to dig through any data documentation in order to resolve these questions.

- i. What is the relevant population for the datasets you've imported?
- ii. About what proportion (to within 0.1) of the population is captured in the sample? (*Hint*: have a look at this website.)
- iii. Considering that the sampling frame is not identified clearly, what kind of dataset do you suspect this is (*e.g.*, administrative, data from a 'typical sample', census, etc.)?
- iv. In light of your description of the sample characteristics, what is the scope of inference for this dataset?

Type your answer here, replacing this text.

Data tidying

Since you've already had some guided practice doing this in previous assignments, you'll be left to fill in a little bit more of the details on your own in this assignment. You'll work with the following variables from each dataset:

- **Test score data**
  - District ID
  - District name
  - Grade
  - Test subject
  - Estimated male-female gap
- **Covariate data**
  - District ID
  - Locale
  - Grade
  - Socioeconomic status (all demographic groups)
  - Log median income (all demographic groups)
  - Poverty rate (all demographic groups)
  - Unemployment rate (all demographic groups)
  - SNAP benefit receipt rate (all demographic groups)

Question 5: Variable names of interest

Download the codebooks by opening the 'data' directory from your Jupyter Lab file navigator and downloading the codebook files. Identify the variables listed above, and store the column names in lists named `main_vars` and `cov_vars`.

```
# store variable names of interest
main_vars = ...
cov_vars = ...
```

```
grader.check("q5")
```

Question 6: Slice columns

Use your result from above to slice the columns of interest from the covariate and test score data. Store the resulting data frames as `main_sub` and `cov_sub` (for 'subset').

```
# slice columns to select variables of interest
main_sub = ...
cov_sub = ...
```

```
grader.check("q6")
```

In the next step you'll merge the covariate data with the test score data. In order to do this, you can use the `pd.merge(A, B, how = ..., on = SHARED_COLS)` function, which will match the rows of `A` and `B` based on the shared columns `SHARED_COLS`. If `how = 'left'`, then only rows in `A` will be retained in the output (so `B` will be merged *to* `A`); conversely, if `how = 'right'`, then only rows in `B` will be retained in the output (so `A` will be merged *to* `B`).

A simple example of the use of `pd.merge` is illustrated below:

```
# toy data frames
A = pd.DataFrame(
    {'shared_col': ['a', 'b', 'c'],
    'x1': [1, 2, 3],
    'x2': [4, 5, 6]}
)

B = pd.DataFrame(
    {'shared_col': ['a', 'b'],
    'y1': [7, 8]}
)
```

```
A
```

```
B
```

Below, if `A` and `B` are merged retaining the rows in `A`, notice that a missing value is input because `B` has no row where the shared column (on which the merging is done) has value `c`. In other words, the third row of `A` has no match in `B`.

```
# left join
pd.merge(A, B, how = 'left', on = 'shared_col')
```

If the direction of merging is reversed, and the row structure of `B` is dominant, then the third row of `A` is dropped altogether because it has no match in `B`.

```
# right join
pd.merge(A, B, how = 'right', on = 'shared_col')
```

Question 7: Merge

Merge the covariate and test score data on both the ***district ID*** and ***grade level*** columns, and retain only the columns from the test score data (meaning, merge the covariate data *to* the test score data). Store the resulting data frame as `rawdata` and print the first four rows.

```
# merge covariates with gap data
rawdata = ...

# print first four rows
...
```

```
grader.check("q7")
```

Question 8: Rename and reorder columns

Now rename and rearrange the columns of `rawdata` so that they appear in the following order and with the following names:

- District ID, District, Locale, log(Median income), Poverty rate, Unemployment rate, SNAP rate, Socioeconomic index, Grade, Subject, Gender gap

Store the resulting data frame as `rawdata_mod1` and print the first four rows.

(*Hint*: first define a dictionary to map the old names to the new ones; then create a list of the new names specified in the desired order; then use `.rename()` and `.loc[]`. You can follow the renaming steps in HW1 as an example if needed.)

```
# define dictionary mapping for renaming columns
...

# specify order of columns
...

# rename and reorder
...

# print first four rows
...
```

```
grader.check("q8")
```

Question 9: Pivot

Notice that the Gender gap column contains the values of two variables: the gap in estimated mean test scores for math tests, and the gap in estimated mean test scores for reading and language tests. To put the data in tidy

format, use `.pivot` and `.rename()` to pivot the table so that the gender gap column is spread into two columns named `Math gap` and `Reading gap`. Store the result as `seda_data` and print the first four rows.

Hint: to avoid unweildy column indexing, make sure you specify a `values = ...` argument when using `.pivot()`. Doing so will result in the column index being named `Subject`; remove this name in your solution.

Aside: an alternative solution is to manipulate the indices and use `.unstack()`, but this method will produce a dataframe with hierarchical column indexing (you'll see) in which `Subject` is retained as a lower-level index; this will need to be collapsed in order to rename the columns as instructed using `MultiIndex.droplevel()` or similar.

```
# pivot to unstack gender gap (fixing tidy issue: multiple variables in one column)
...

# print first four rows
...
```

```
grader.check("q9")
```

Your final dataset should match the dataframe below. You can use this to check your answer and revise any portions above that lead to different results.

```
# intended result
data_reference = pd.read_csv('data/tidy-seda-check.csv')
data_reference
```

Question 10: Sanity check

Ensure that your tidying did not inadvertently drop any observations: count the number of units in `seda_data`. Does this match the number of units represented in the original test score data `ca_main`? Store these values as `data_units` and `ca_main_units`, respectively.

(*Hint*: use `.nunique()`.)

```
# number of districts in tidied data compared with raw
data_units = ...
ca_main_units = ...
```

```
grader.check("q10")
```

Question 11: Missing values

Gap estimates were not calculated for certain grades in certain districts due to small sample sizes (not enough individual tests recorded). Answer the following:

- i. What proportion of rows are missing for each of the reading and math gap variables? Store these values as `math_missing` and `reading_missing`, respectively.

- ii. What proportion of *districts* (not rows!) have missing gap estimates for one or both test subjects for at least one grade level?Store the value as `district_missing`.

```
# proportion of missing values
math_missing, reading_missing = ...
# proportion of districts with missing values

...
```

```
grader.check("q11")
```

Question 12: Missing mechanism

Do you expect that this missingness is more likely for some districts than for others? If so, explain; why is this, and is bias a concern if missing values are dropped?

Type your answer here, replacing this text.

Exploratory graphics

For the purpose of visualizing the relationship between estimated gender gaps and socioeconomic variables, you'll find it more helpful to store a non-tidy version of the data. The cell below rearranges the dataset so that one column contains an estimated gap, one column contains the value of a socioeconomic variable, and the remaining columns record the gap type and variable identity.

Ensure that your results above match the reference dataset before running this cell.

```
# format data for plotting
plot_df = seda_data.melt(
    id_vars = name_order[0:9],
    value_vars = ['Math gap', 'Reading gap'],
    var_name = 'Gap type',
    value_name = 'Gap'
).melt(
    id_vars = ['District ID', 'District', 'Locale', 'Gap type', 'Gap', 'Grade'],
    value_vars = name_order[3:8],
    var_name = 'Socioeconomic variable',
    value_name = 'Measure'
)

# preview
plot_df.head()
```

Gender gaps and socioeconomic factors

The cell below generates a panel of scatterplots showing the relationship between estimated gender gap and socioeconomic factors for all grade levels by test subject. The plot suggests that the reading gap favors girls consistently across the socioeconomic spectrum – in a typical district girls seem to outperform boys by 0.25 standard deviations of the national average. By contrast, the math gap appears to depend on socioeconomic factors – boys only seem to outperform girls under *better* socioeconomic conditions.

```
# plot gap against socioeconomic variables by subject for all grades
fig1 = alt.Chart(plot_df).mark_circle(opacity = 0.1).encode(
    y = 'Gap',
    x = alt.X('Measure', scale = alt.Scale(zero = False), title = ''),
    color = 'Gap type'
).properties(
    width = 100,
    height = 100
).facet(
    column = alt.Column('Socioeconomic variable')
).resolve_scale(x = 'independent')

fig1
```

Question 13: Relationships by grade level

Does the pattern shown in the plot above persist within each grade level? Modify the plot above to show these relationships by grade level: generate a panel of scatterplots of gap against socioeconomic measures by subject, where each column of the panel corresponds to one socioeconomic variable and each row corresponds to one grade level; the result should by a 5x5 panel. Resize the width and height of each facet so that the panel is of reasonable size. Keep a fixed axis scale for the variable of interest, but allow the axis scales for socioeconomic variables to vary independently. Store the plot as `fig2`; display the figure and provide an answer to the question of interest in the text cell.

(*Hint*: you may find it useful to have a look at the altair documentation on compound charts, and lab 3, for examples to follow.)

Type your answer here, replacing this text.

```
# plotting codes here
...

# display
...
```

Question 14: Association with grade level

Do gaps shift across grade levels? It's not so easy to tell from the last figure. Construct a 2x5 panel of scatterplots showing estimated achievement gap against each of the 5 socioeconomic variables, with one row per test subject. Display grade level using a color gradient. Store the plot as `fig3`; display the figure and answer the question of interest in a short sentence or two in the text cell provided.

Type your answer here, replacing this text.

```
# plotting codes here
...

# display
...
```

While the magnitude of the achievement gaps seems to depend very slightly on grade level (figure 3), the form of relationship between achievement gap and socioeconomic factors does not differ from grade to grade (figure 2).

Given that the relationships between achievement gaps and socioeconomic factors don't change drastically across grade levels, it is reasonable to look at the average relationship between estimated achievement gap and median income after aggregating across grade.

Question 15: Aggregation across grade levels

Compute the mean estimated achievement gap in each subject across grade levels by district using `District ID` and retain the district-level socioeconomic variables. Store the resulting data frame as `seda_data_agg`.

Note: best practice here would be to aggregate just the test scores by district and then re-merge the result with the district-level socioeconomic variables. However, since the district-level socioeconomic variables do not differ by grade within a district, averaging them across grade levels by district together with the test scores will simply return their unique values; so the aggregation can be applied across *all* columns for a fast-and-loose way to obtain the desired result.

```
# aggregate across grades
...

# print first few rows
...
```

```
grader.check("q15")
```

Question 16: Melt aggregated data for plotting

Similar to working with the disaggregated data, it will be helpful for plotting to melt the two gap variables into a single column. Follow the example above at the beginning of this section to melt *only the test score gap columns* (not the district-level variables – we will not create scatterplot panels as before). Name the new columns `Subject` and `Average estimated gap`; store the resulting data frame as `agg_plot_df` and print the first four rows.

```
# format for plotting
...

# print four rows
...
```

```
grader.check("q16")
```

Question 17: District average gaps

Construct a scatterplot of the average estimated gap against log(Median income) by subject for each district and add trend lines (see lab 4). Store the plot as `fig4`. Describe and interpret the plot in a few sentences.

Type your answer here, replacing this text.

```
# scatterplot
...

# trend line
trend = ...

# combine layers
fig4 = ...

# display
...
```

Now let's try to capture this pattern in *tabular* form. The cell below adds an `Income bracket` variable by cutting the median income into 8 contiguous intervals using `pd.cut()`, and tabulates the average socioeconomic measures and estimated gaps across districts by income bracket. Notice that with respect to the gaps, this displays the pattern that is shown visually in the figures above.

```
seda_data_agg['Income bracket'] = pd.cut(np.e**seda_data_agg['log(Median income)'], 8)
seda_data_agg.groupby('Income bracket').mean().drop(columns = ['District ID', 'log(Median income)
```

Question 18: Proportion of districts with a math gap

What proportion of districts in each income bracket have an average estimated math achievement gap favoring boys? Answer this question by performing the following steps:

- Append an indicator variable `Math gap favoring boys` to `seda_data_agg` that records whether the average estimated math gap favors boys by more than 0.1 standard deviations relative to the national average.
- Compute the proportion of districts in each income bracket for which the indicator is true: group by bracket and take the mean. Store this as `income_bracket_boys_favored`

```
# define indicator
seda_data_agg['Math gap favoring boys'] = ...

# proportion of districts with gap favoring boys, by income bracket
...
```

```
# print result
...
```

```
grader.check("q18")
```

Question 19: Statewide averages

To wrap up the exploration, calculate a few statewide averages to get a sense of how some of the patterns above compare with the state as a whole.

- i. Compute the statewide average estimated achievement gaps. Store the result as `state_avg`.
- ii. Compute the proportion of districts in the state with a math gap favoring boys. Store this result as `math_boys_proportion`
- iii. Compute the proportion of districts in the state with a math gap favoring girls. You will need to define a new indicator within `seda_data_agg` to perform this calculation.

```
# statewide average
state_avg = ...

# proportion of districts in the state with a math gap favoring boys
math_boys_proportion = ...

# proportion of districts in the state with a math gap favoring girls
seda_data_agg['Math gap favoring girls'] = ...
math_girls_proportion = ...
```

```
grader.check("q19")
```

Communicating results

Take a moment to review and reflect on your findings and consider what you have learned from the analysis.

Question 20: Summary

Write a brief summary of your exploratory analysis. What have you discovered about educational achievement gaps in California school districts? Aim to answer in 3-5 sentences or less.

Type your answer here, replacing this text.

Submission

1. Save the notebook.
2. Restart the kernel and run all cells. (**CAUTION**: if your notebook is not saved, you will lose your work.)
3. Carefully look through your notebook and verify that all computations execute correctly and all graphics are displayed clearly. You should see **no errors**; if there are any errors, make sure to correct them before you

submit the notebook.

4. Download the notebook as an `.ipynb` file. This is your backup copy.
5. Export the notebook as PDF and upload to Gradescope.

To double-check your work, the cell below will rerun all of the autograder tests.

```
grader.check_all()
```