Practice Prelim 2

AEM 2850 / AEM 5850

Answer Key

Preface

The goal of this prelim is to assess your understanding of data visualization concepts and facility with key visualization and programming tools covered in weeks 7 through 14 of the course.

Instructions

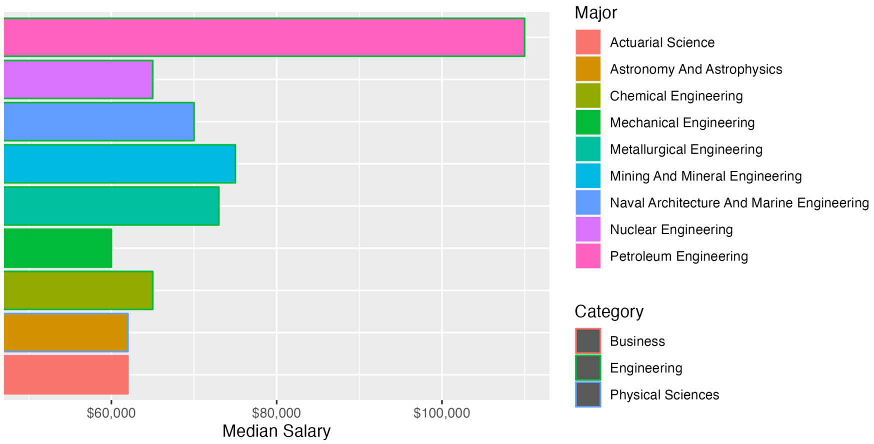
- You must complete Prelim 2 **in person** in Warren 150 during class
- Prelim 2 is open internet, but **do not communicate with classmates (or others!)**
- Do not use packages outside the `tidyverse` packages we have already loaded for you (penalties may apply)
- When done, **upload BOTH your .qmd and .pdf files** to canvas

Additional notes

- There are 8 questions worth a total of 100 points. The total number of points per question is stated with each question
- **Render early and often** to avoid wasting time sorting out what code needs debugging
- We will give partial credit if your answers are incomplete, especially if you provide comments or text that describes the logic of what you *would* do if you had more time
- If you have trouble rendering your document, do not delete your work in progress code. That will make it hard for us to give you partial credit. Instead, you can:
  - Comment out problematic code using `#` or keyboard shortcut Cntrl/Cmd-Shift-C
  - Replace `{r}` with `{r, eval = FALSE}` at the top of the relevant code chunk
  - Ask questions!
- FYI: we added page breaks between each question and spacing in some places using `\vfill`, please leave them in place and just ignore them

Improving Data Visualizations

1. [12 points] Earlier this semester we worked with data on college major salaries. We want to make a plot to compare the median salaries for the top 10 college majors relative to one another, both as individual majors and across categories of majors (e.g., Business, Engineering, etc.). Here is a first attempt:



Describe four changes you could make to improve this data visualization without losing any information.

Note: do not write code for this part. You may name the function(s) you would use, but that is not required for full credit – you just need to describe the changes you would make.

1. Make the x axis start at zero. Bar charts should always start at zero!
2. Label the majors directly on the y-axis. Using fill with the legend is problematic because the labels are far from the data and because they are in a different order.
3. Map the major category to fill rather than color. It is difficult to distinguish the category of each major because the color borders are so thin.
4. Order the bars by the value of median salary. Since the focus here is on the top 10 majors it makes more sense to rank than to alphabetize them.

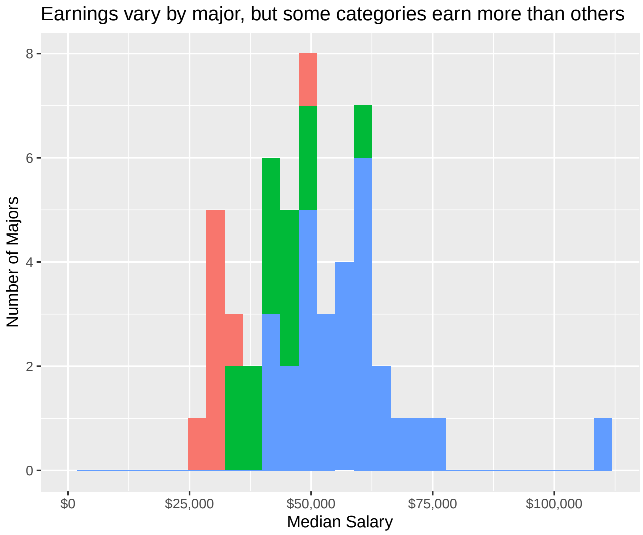
Note: we will accept other answers as long as they are reasonable and would improve the data visualization. For example, this plot would benefit from a title that explains what it is, and where the data came from.

2. [12 points] We want to compare *categories* of majors according to the median salary of all the majors in each category, not just the top 10. The graph below presents the distribution of median salaries for three categories. Describe two changes you could make to more effectively compare the three different categories, and then edit the code below to implement your proposed changes.

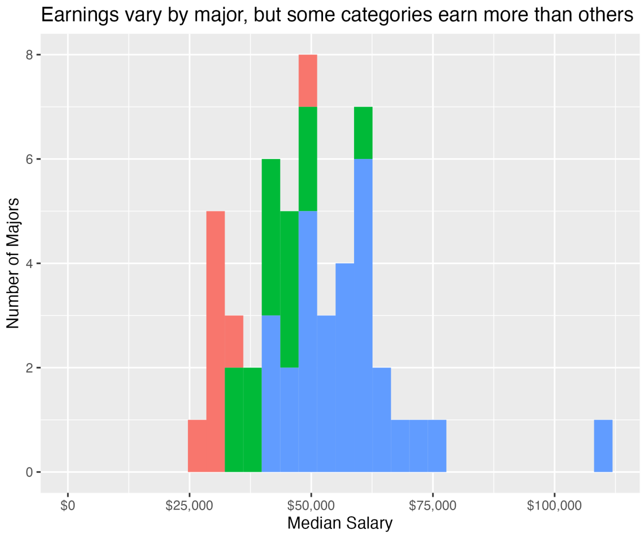
1. …

2. …

```
fivethirtyeight::college_recent_grads |>
  filter(major_category %in% c("Arts", "Engineering", "Business")) |>
  ggplot(aes(x = median, fill = major_category)) +
  geom_histogram() +
  scale_x_continuous(breaks = seq(0, 100000, 25000),
                     labels = scales::label_dollar(),
                     limits = c(0, NA)) +
  guides(fill = "none") +
  labs(x = "Median Salary",
       y = "Number of Majors",
       title = "Earnings vary by major, but some categories earn more than others")
```



Note: we included a static version of the image on this page so you can easily compare your revised graphic to the original:

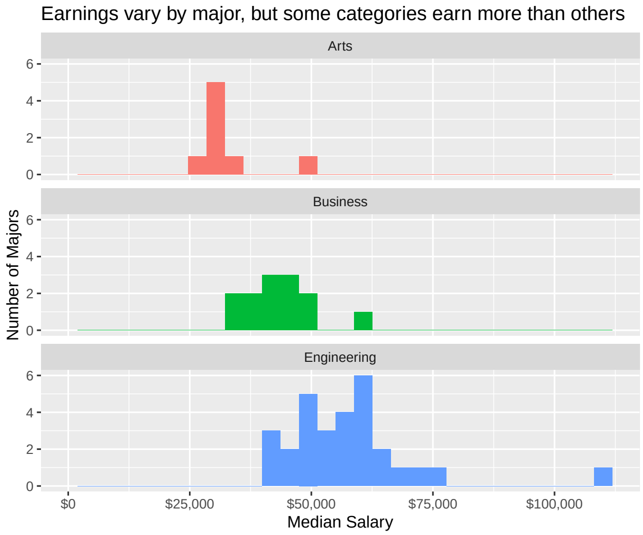


ANSWER:

1. Label the major categories.
2. Separate the three histograms so the underlying data is visible.

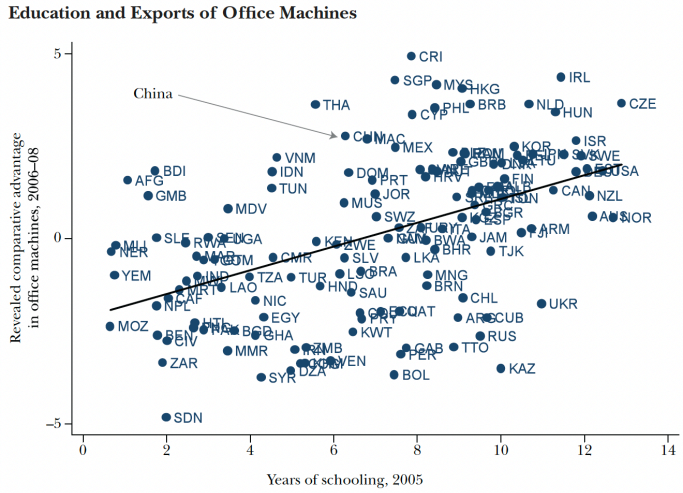
Both can be achieved by adding a single `facet_wrap()` layer to the plot:

```r
fivethirtyeight::college_recent_grads |>
  filter(major_category %in% c("Arts", "Engineering", "Business")) |>
  ggplot(aes(x = median, fill = major_category)) +
  geom_histogram() +
  scale_x_continuous(breaks = seq(0, 100000, 25000),
                     labels = scales::label_dollar(),
                     limits = c(0, NA)) +
  guides(fill = "none") +
  labs(x = "Median Salary",
       y = "Number of Majors",
       title = "Earnings vary by major, but some categories earn more than others") +
  facet_wrap(vars(major_category), ncol = 1) # this is one way to answer the question
```



3. [9 points] The plot below comes from an article about economic development and globalization. The text of the article explains it as follows:

> "[The figure] plots countries' revealed comparative advantage in office machines... against the average years of schooling of the adult population... China is above the regression line, indicating that its specialization in the sector is greater than one would expect given its level of education, but it is hardly an extreme outlier. Other middle-income countries—including Costa Rica, the Philippines, Malaysia, and Thailand—have larger positive residuals."



Source: Hanson (2012).

Describe three changes you could make to this visualization to better illustrate the ideas in the text above.

Note: do not write code for this part. You may name the function(s) you would use, but that is not required for full credit – you just need to describe the changes you would make.

1. Do not label every point. Instead, label only the five countries described in the text.
2. Do not label the points with three-letter country codes. Instead, spell the country names out, using arrows or jitter as needed to avoid overlap.
3. Do not give all points equal weight in visual terms. Instead, use another aesthetic such as color to distinguish the five countries discussed in the text from all the other countries.

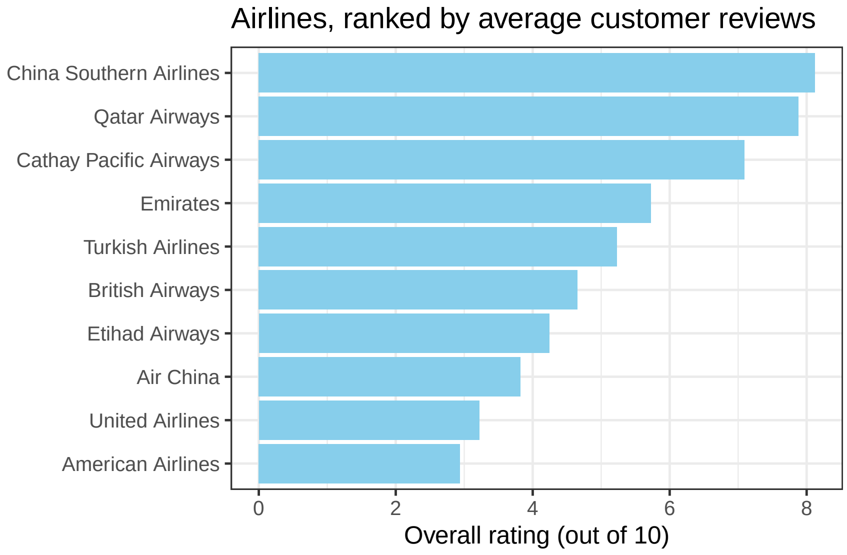
Note: we will accept other answers as long as they are reasonable and would improve the data visualization. For example, a more radical approach would be to plot the residuals for each country as amounts, rather than using the two-dimensional scatterplot above.

Analyzing Airline Reviews

4. [12 points] We imported some airline reviews data for you at the beginning of this file and assigned it to the name `airline_reviews`. Use `airline_reviews` to make a bar/column plot to compare the average of `overall` ratings for each `airline`. There should be one bar/column per `airline`, and the length/height should correspond to the average of `overall` ratings for that `airline`. Arrange the bars/columns in descending order by rating, and make the area of the bars/columns `skyblue`. Make sure that any text on the plot is clear and understandable, and does not overlap other text. Use `theme_bw()`.

Note: in the problem statement above, the terms bar and column are used as synonyms. It is your job to choose the appropriate geometry based on what information the prompt asks you to convey in your visualization.

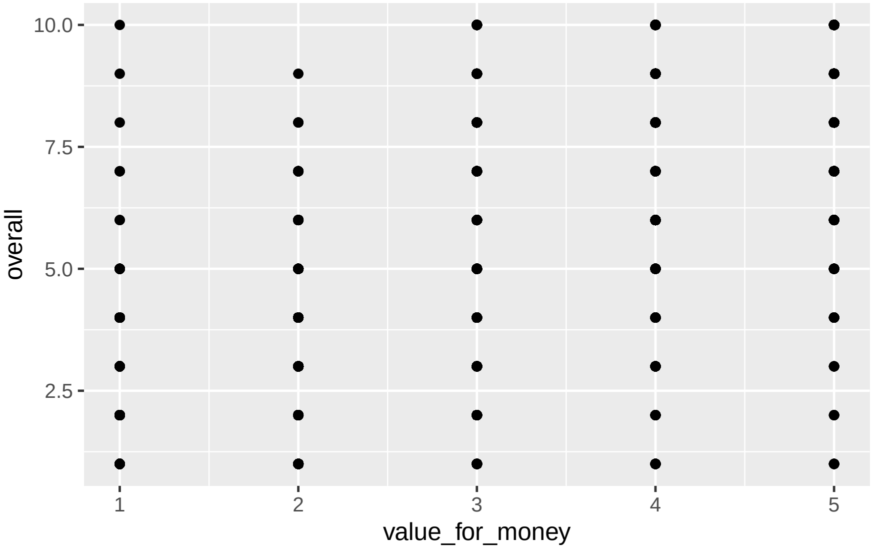
```
airline_reviews |>
  group_by(airline) |>
  summarize(overall = mean(overall)) |>
  ggplot(aes(y = fct_reorder(airline, overall), x = overall)) +
  geom_col(fill = "skyblue") +
  labs(x = "Overall rating (out of 10)",
       y = "",
       title = "Airlines, ranked by average customer reviews") +
  theme_bw()
```



5. [12 points] In the data set, customers not only give an overall rating, but also rate different aspects of flights. Make a basic scatterplot of `overall` vs `value_for_money` without customizing the geometry or adding other layers.

Note: For this question, you will not be graded on the aesthetic presentation of your graph, so don't waste time making nice labels, etc.

```
airline_reviews |>
  ggplot(aes(x = value_for_money, y = overall)) +
  geom_point()
```



Is your basic scatterplot very informative about the relationship between the two variables? If not, name one way you could make it more informative, and use words to describe how you would implement your suggestion.

Note: do not write code for this part. You may name the function(s) you would use, but that is not required for full credit – you just need to describe the approach you would take.

No, it is not informative at all!

Correct ways to make it more informative include: add a flexible fit using `geom_smooth()`, add a linear fit using `geom_smooth(method = "lm")`, use `geom_jitter()` to shift the points so they are more visible, adjust the transparency of points using `alpha`, etc.

6. [12 points] Visualizations are not always the best way to summarize relationships. Use linear regression to model `overall` as a function of `value_for_money`. Print the results using `summary()`. Describe the interpretation of the coefficient on `value_for_money` in words.

```r
lm(overall ~ value_for_money,
   data = airline_reviews) |>
  summary()
```

```

Call:
lm(formula = overall ~ value_for_money, data = airline_reviews)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.0429 -0.3035 -0.1081  0.8919  8.6965 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)     -0.631362   0.031395  -20.11   <2e-16 ***
value_for_money  1.934861   0.009131  211.90   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.485 on 10683 degrees of freedom
Multiple R-squared:  0.8078,    Adjusted R-squared:  0.8078 
F-statistic: 4.49e+04 on 1 and 10683 DF,  p-value: < 2.2e-16
```

On average, a one unit increase in `value_for_money` is *associated* with a 1.9 unit increase in `overall`.

How would your interpretation of the coefficient on `value_for_money` change if you added other factors such as `seat_comfort` and `entertainment` into the model? Do you think the coefficient would be larger or smaller than the coefficient you estimated above?

The interpretation would be modified to be an association "holding other factors in the model constant." The coefficient would proabbly be smaller than the one above because of other factors that are correlated with both `value_for_money` and `overall`.

Scraping the Course Roster

7. [15 points] Pre-enroll has come and gone, but the course roster lives on. Scrape the names of Fall 2023 AEM courses from the course roster. Use the url contained in the object `roster_url`, pre-assigned for you at the beginning of this file. Use the CSS selector ".title-coursedescr" to extract course titles. Convert the extracted html to a character vector (not a table/data frame) called `course_names`. Print the `head()` of `course_names`.

```r
course_names <- read_html(roster_url) |>
  html_elements(".title-coursedescr") |>
  html_text2()

head(course_names)
```

```
[1] "Design Your Dyson"
[2] "The Business of Modern Medicine"
[3] "Spreadsheet Modeling for Management and Economics"
[4] "Spreadsheet Modeling for Non-Dyson Majors"
[5] "Introduction to Agricultural Finance"
[6] "Introductory Statistics"
```

You could use a similar method to scrape course numbers using the CSS selector ".title-subjectcode". Suppose you did, and assigned it to another object, `course_numbers`. How could you combine the information you scraped to identify course numbers for all the courses with the word "Accounting" in the name?

Note: do not write code for this part. You may name the function(s) you would use, but that is not required for full credit – you just need to describe the approach you would take.

One option would be to combine `course_numbers` and `course_names` into a data frame and then filter for rows in which `course_names` contain "Accounting" using a pattern matching function such as `str_detect()`.

Another option would be to work with the two separate character vectors, and use pattern matching to identify the position of course names that contain "Accounting" and then extract the course numbers from `course_numbers` based on those positions.

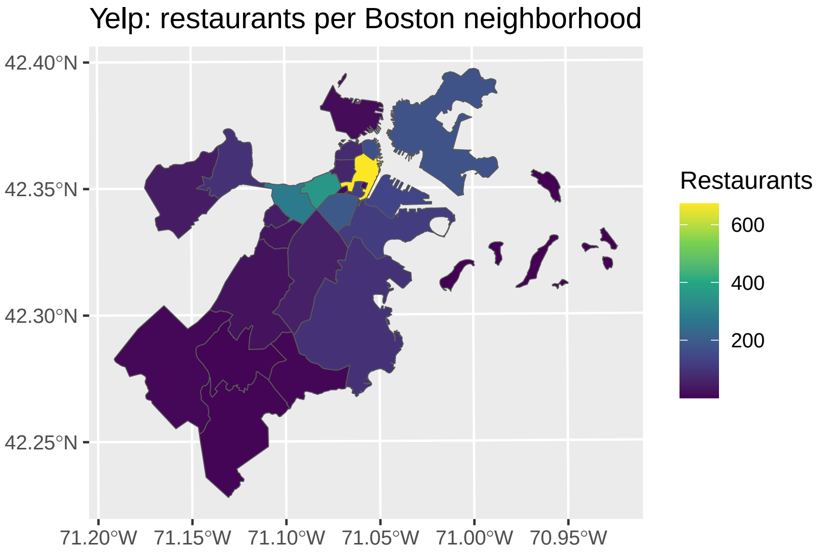
Mapping Yelp Reviews in Boston

8. [16 points] Read in the shapefile with spatial data about the neighborhoods of Boston contained in the folder `boston`. Use `left_join` to augment these data with the data in `yelp` (created above for you), which contains restaurant listings in each row and the neighborhood of each listing in the column `Name`. Be careful to start with the neighborhoods data to preserve the `sf` nature of the data frame. Then count the number of listings in each neighborhood and make a map of Boston neighborhoods, using `fill` to shade each neighborhood according to the number of listings in it. Customize the title, fill scale, and legend as time allows.

Note: if you have trouble, consider plotting the map of neighborhoods without the number of restaurant listings (using just the `boston` spatial data), or printing a table of the number of listings per neighborhood (using just the `yelp` data), or both, so we can award partial credit.

```
neighborhoods <- read_sf("boston/neighborhoods.shp")

left_join(neighborhoods, yelp, join_by(Name)) |>
  count(Name) |>
  ggplot() +
  geom_sf(aes(fill = n)) +
  labs(title = "Yelp: restaurants per Boston neighborhood",
       fill = "Restaurants") +
  scale_fill_viridis_c()
```


Yelp: restaurants per Boston neighborhood
42.40°N
42.35°N
42.30°N
42.25°N
71.20°W
71.15°W
71.10°W
71.05°W
71.00°W
70.95°W
Restaurants
600
400
200