Practice Questions for Prelim 2

AEM 2850 / AEM 5850

Answer Key

READ THESE NOTES FIRST:

- Prelim 2 will cover all content we covered in weeks 7 through 14
  - Prelim 2 will also rely on prerequisite knowledge of fundamental concepts we learned in the first part of the course, but the questions are not designed to test that knowledge directly
- These practice questions are intended as a study resource, not a comprehensive guide
- These practice questions are not exhaustive in terms of topics and question types
- These practice questions are not necessarily representative of the weight that different topics and question types will receive on Prelim 2

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.

Instructions

- You must complete Prelim 2 in person during class
- Prelim 2 is a closed-book paper prelim
- When done, hand in your completed prelim and have a great summer!

Additional notes

- There are X questions worth a total of 100 points. The total number of points per question is stated with each question
- 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

Multiple choice: circle one answer per question.

Q. [X points] What is one risk of truncating the y-axis in a bar chart?

a. Harder to match color to category
b. Chart may look too crowded
c. It may exaggerate small differences
d. It causes R to crash

Q. [X points] Which ggplot geom is best for visualizing data over time?

a. `geom_point()`
b. `geom_col()`
c. `geom_line()`
d. `geom_histogram()`

Q. [X points] What is the purpose of an `if` statement?

a. To rename variables
b. To automate downloads
c. To apply styles to ggplot2 charts
d. To control which code runs under certain conditions

Q. [X points] Which function listed below is used for web scraping?

a. `read_csv()`
b. `read_sf()`
c. `read_html()`
d. `read_lines()`

Short answer

Q. [X points] Name one advantage and one disadvantage of using word clouds for text analysis.

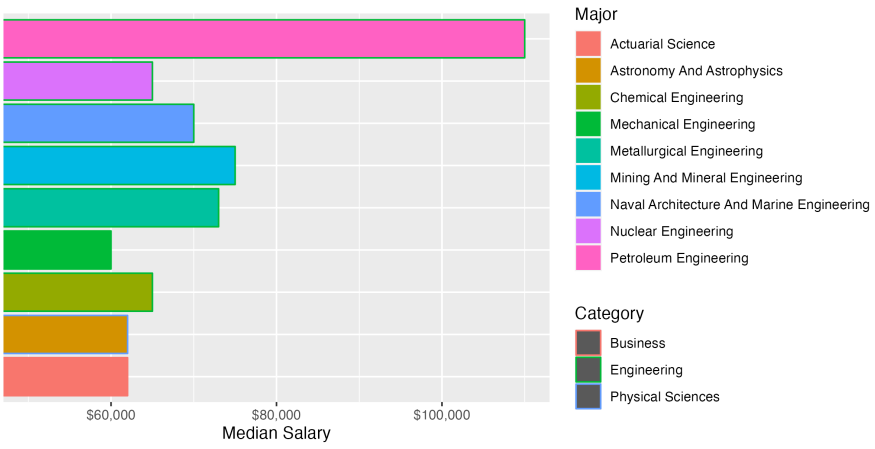
Advantage:

Disdvantage:

Q. [X points] True or False: "Pie charts are best for subtle differences between many categories." Explain.

Q. [X points] True or False: "Web scraping using the `rvest` function `read_html()` works well on client-side websites because the function does not rely on having html code that contains all the information we see when we visit a webpage." Explain.

Q. [X 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, and explain what issue each change is meant to address.

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.

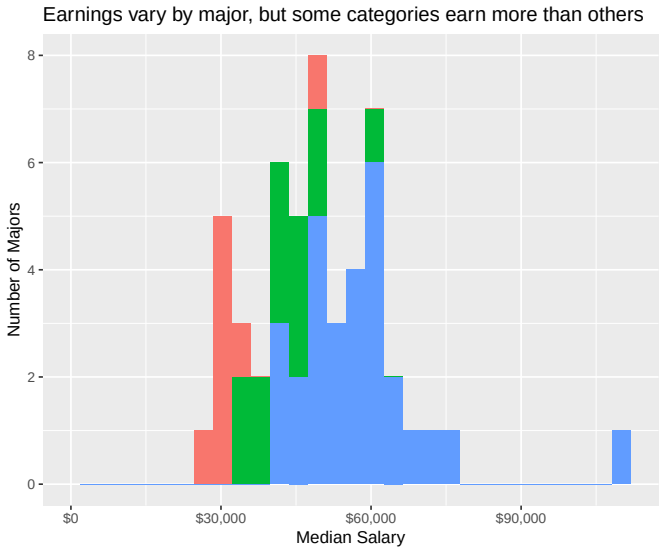
Note: we will accept other answers as long as they are reasonable and would improve the data visualization. For example, this plot would benefit from a title that explains what it is, and where the data came from.

Q. [X points] Now we want to compare *categories* of majors according to the median salaries of all the majors in each category, not just the top 10 majors overall. The code below creates the graph below, which presents the distribution of median salaries for three categories. Describe two changes you could make to more effectively compare the three different categories, explain what issue each change is meant to address, and then edit the code below to implement your proposed changes.

1.

2.

```
college_majors |>
  ggplot(aes(x = median, fill = major_category)) +
  geom_histogram() +
  scale_x_continuous(
    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"
  )
```



ANSWER:

1. Label the major categories.

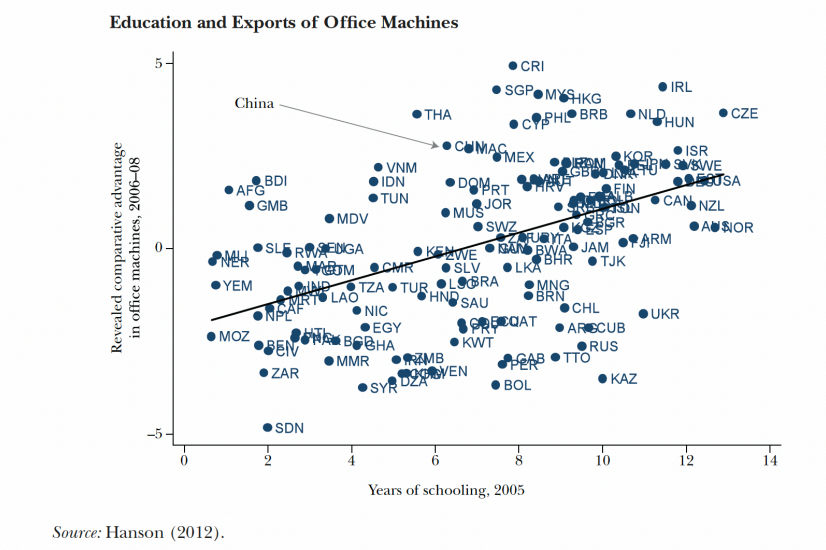
2. Separate the three histograms so the underlying data is visible.

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

```
college_majors |>
  ggplot(aes(x = median, fill = major_category)) +
  geom_histogram() +
  scale_x_continuous(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) # one way to answer the question
```

Q. [X 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."



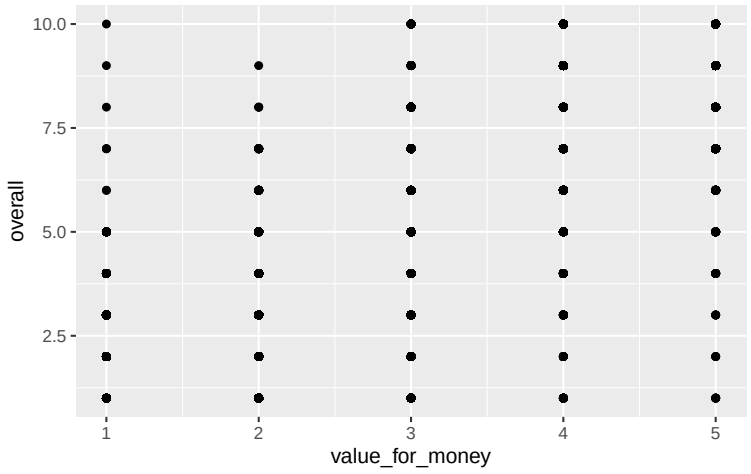
Source: Hanson (2012).

Describe three changes you could make to this visualization to better illustrate the ideas in the text above, and explain what issue each change is meant to address.

Notes: Do not worry about what "comparative advantage in office machines" means – we are looking for generic data visualization suggestions, not anything specific to the outcome plotted on the y axis. Also do not suggest changes to the data used to make this chart – think instead about how you could modify aesthetic mappings, layers, etc.

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.

Q. [X points] Below is a basic scatterplot we made using a data frame called `airline_reviews` that contains reviews of different airlines' flights. In the data frame, each observation (i.e., row) corresponds to an individual review, and the `overall` and `value_for_money` review scores take on integer values.



Write code to replicate the plot above using the data frame `airline_reviews`.

Note: We are asking you to replicate the plot above, even if you do not think it is an effective data visualization. Do not spend time trying to improve it.

```
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, write a brief code snippet below that would produce more informative output. It could be an adaptation of the visualization above, but it does not necessarily have to be a visualization.

No, it is not informative at all!

Correct ways to make it more informative include, but are not limited to:

- add a smoothed fit line using `geom_smooth()`
- use `geom_jitter()` to shift the points so they are more visible
- adjust the transparency of points using `alpha`

- use `geom_boxplot()` or similar to compare the distribution of `overall` for different values of `value_for_money`
- compute the correlation between the two variables using `summarize(cor())`
- estimate a regression model using `lm()`

Q. [X points] The code below computes Apple's annualized return using the data frame `aapl_prices`, which includes data that range in `date` from 2020-01-02 through 2024-12-31. The data frame also contains a variable `year` that contains the year corresponding to each `date`. Here are the first two rows of the data frame:

```
aapl_prices |> head(2)
```

```
# A tibble: 2 x 9
  symbol date        open  high   low close    volume adjusted  year
  <chr>  <date>     <dbl> <dbl> <dbl> <dbl>     <dbl>    <dbl> <dbl>
1 AAPL   2020-01-02  74.1  75.2  73.8  75.1 135480400     72.7  2020
2 AAPL   2020-01-03  74.3  75.1  74.1  74.4 146322800     72.0  2020
```

Consider the function `get_annual_return()` that returns Apple's annualized return since a `start_year` of the user's choice:

```
get_annual_return <- function(start_year){
  years <- 2025 - start_year
  aapl_prices |>
    filter(year >= start_year) |>
    filter(date==min(date) | date==max(date)) |>
    mutate(cum_return = (adjusted - lag(adjusted)) / lag(adjusted) ) |>
    filter(!is.na(cum_return)) |>
    mutate(ann_return = ((1 + cum_return)^(1/years) - 1) * 100) |>
    pull(ann_return)
}
```

Circle all the function calls below that will return valid annualized returns:

a. `get_annual_return(2015)` returns an invalid result because the data start after 2015

b. **`get_annual_return(2020)` returns a valid result**

c. `get_annual_return("January 1, 2022")` returns an invalid result, in the form of an error, since the first line of the function attempts to subtract the character "January 1, 2022" from the number 2025

d. `get_annual_return(2025)` returns an invalid result because the data end prior to 2025

Q. [X points] Your report on Volkswagen's Dieselgate circulated among executives in the auto industry and a few of them are interested in hiring you as a consultant. Ford, Toyota, and General Motors shared spreadsheets with you containing their sales in 2024 of various vehicle types (sedan, SUV, truck, etc) by state. The three spreadsheets are very different, but upon importing them into R you noticed that some of the variables have the same names across all three and are measured in comparable ways (`vehicle_type` and `n_sales`).

Describe two different approaches you could take to calculate the total number of vehicles sold in the US by brand and vehicle type. For each one, name specific functions you would use, and how they would work in this context.

Note: We will award credit based on the logic of your approaches first and foremost, followed by your understanding of key functions needed to implement the approach. You are not expected to write a complete code snippet that will run without errors (and will not receive any extra credit if you do).

1. In a separate step for each data frame, `select()` the relevant variables and use `mutate()` to create a new column `brand` containing the brand name. Use `bind_rows()` to combine the data frames. Then, group the data by `brand` and `vehicle_type` using `group_by()`, and `summarize()` the total number of sales using `sum(n_sales)` (potentially with the `na.rm = TRUE` argument).

2. Write a custom `function()` that takes a data frame as its input. For each input, have the function `group_by()` the `vehicle_type` and `summarize()` the total number of sales using `sum(n_sales)` (potentially with the `na.rm = TRUE` argument). Then call the function multiple times using `map()`, and use `bind_rows()` to combine them into a single data frame.

Note: Other answers are acceptable. For example, you could do this using a loop.

How do the two approaches you proposed compare? Would you say one is better than the other? Why?

In general, using a custom function with `map()` would be preferable to copying and pasting code to do the same operation many times on different inputs.