Exam 3 Closed Notes

DSST 289: Introduction to Data Science

Erik Fredner

1 Name

Name_____

Section start time_____

2 Exam

- You may only use a pen or pencil and scratch paper on this exam.
- If you cannot solve a problem, write what you do know for partial credit.
- Code will be graded on accuracy and formatting.

3 Questions

3.1 Data types

```
traffic_violations
```

```
# A tibble: 3 x 5
  violation         occurred                        location  fine paid
  <chr>             <dttm>                          <POINT> <dbl> <lgl>
1 Speeding          2024-01-01 08:30:00 (37.57754 -77.53546) 150   TRUE
2 Illegal parking   2024-01-02 14:45:00 (37.57841 -77.53486)  75.5 FALSE
3 Running red light 2024-01-03 17:20:00 (37.57264 -77.54745) 200   FALSE
```

Identify the data type of each column in `traffic_violations` and describe its purpose. For example: `<int>` represents the integer data type, which stores whole numbers like `1`, as distinct from decimals like `1.5`.

3.2 The novice

Using the table above, a novice programmer is trying to calculate the total amount of unpaid fines. Here is their code so far:

```
  traffic | > filtar(Paid %is%   FALSE) + mutate[outstanding== sun(fine) ) |>
```

Ultimately, their code should have the following output:

```
traffic_violations |>
  filter(paid == FALSE) |>
  summarize(outstanding = sum(fine))
```

```
# A tibble: 1 x 1
  outstanding
        <dbl>
1        276.
```

Identify problems with their code as written. You may either rewrite it correctly *or* list the issues.

3.3 Average fine

```
officer_fines
```

```
# A tibble: 6 x 4
  officer_id date        fine paid
       <dbl> <date>     <dbl> <lgl>
1        101 2024-01-01   100 TRUE
2        102 2024-01-01   150 FALSE
3        101 2024-01-01   300 FALSE
4        101 2024-01-02   200 TRUE
5        102 2024-01-02   100 TRUE
6        102 2024-01-02   400 FALSE
```

```
officer_fines |>
  summarize(average_fine = mean(fine))
```

```
# A tibble: 1 x 1
  average_fine
         <dbl>
1         208.
```

```
officer_fines |>
  group_by(officer_id) |>
  summarize(total_fines = sum(fine)) |>
  ungroup() |>
  summarize(average_fine = mean(total_fines))
```

```
# A tibble: 1 x 1
  average_fine
         <dbl>
1          625
```

Will `average_fine` differ in the two code blocks above? If so, how?

3.4 Ticket fees

Unpaid tickets automatically incur a late fee. The late fee is one-third of the fine. Using `officer_fines`, write code that will produce the following table:

```
officer_fines |>
  mutate(
    fee = if_else(paid, 0, fine / 3),
    total = fine + fee
  )
```

```
# A tibble: 6 x 6
  officer_id date        fine paid    fee total
       <dbl> <date>     <dbl> <lgl> <dbl> <dbl>
1        101 2024-01-01   100 TRUE      0   100
2        102 2024-01-01   150 FALSE    50   200
3        101 2024-01-01   300 FALSE   100   400
4        101 2024-01-02   200 TRUE      0   200
5        102 2024-01-02   100 TRUE      0   100
6        102 2024-01-02   400 FALSE   133.  533.
```

3.5 Flowers of Virginia

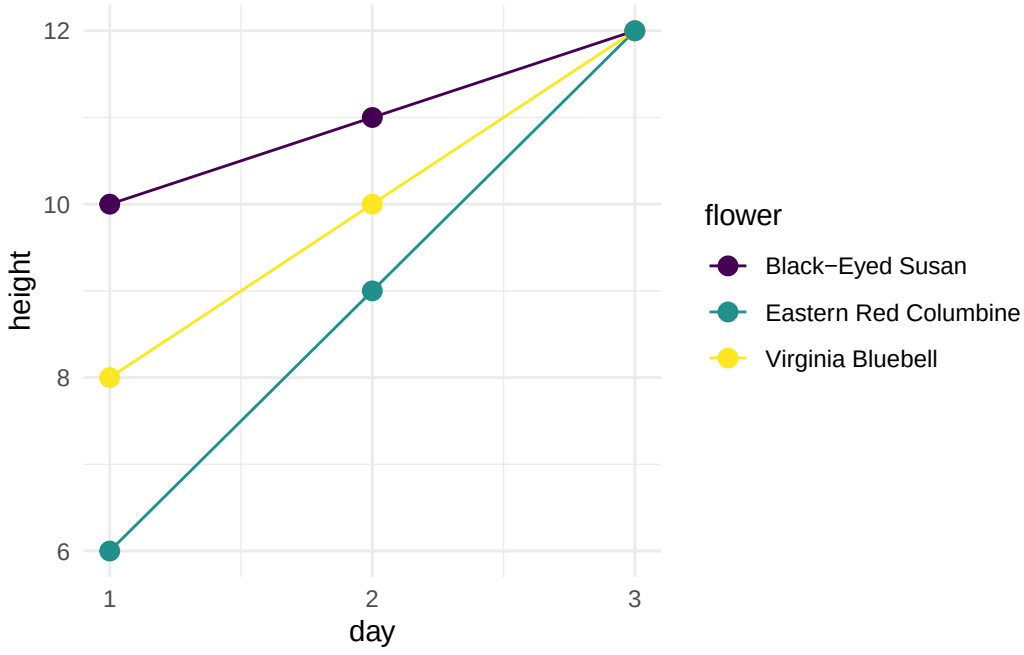
```
flowers
```

```
# A tibble: 3 x 4
  flower                height_day_1 height_day_2 height_day_3
  <chr>                        <dbl>        <dbl>        <dbl>
1 Black-Eyed Susan                10           11           12
2 Virginia Bluebell                8           10           12
3 Eastern Red Columbine            6            9           12
```

Fill in the blanks to reshape `flowers` such that it can be used to make the plot shown below. *Nota bene*: No custom labels have been applied to the plot.

```
_________________ <- flowers |>
  _________________(
    cols = _________________
    names_to = _________________
    names_prefix = "height_day_",
    names_transform = as.integer,
    values_to = _________________
  )
```

```
flowers |>
  pivot_longer(
    cols = -flower,
    names_to = "day",
    names_prefix = "height_day_",
    names_transform = as.integer,
    values_to = "height"
  ) |>
  ggplot(aes(x = day, y = height, color = flower)) +
  geom_line() +
  geom_point(size = 3) +
  scale_color_viridis_d() +
  scale_x_continuous(breaks = 1:3)
```



3.6 Flower plot

Beginning from your reshaped `flowers`, write code to reproduce the plot above.

3.7 Fast cars

These tables show when and where two artists performed the song "Fast Car:"

```
setlists
```

```
# A tibble: 6 x 4
  artist        date       song     set_order
  <chr>         <date>     <chr>        <dbl>
1 Tracy Chapman 1988-08-01 Fast Car         1
2 Tracy Chapman 1990-07-05 Fast Car        18
3 Tracy Chapman 1995-09-10 Fast Car        18
4 Luke Combs    2023-02-15 Fast Car         6
5 Luke Combs    2023-06-10 Fast Car         1
6 Luke Combs    2023-11-20 Fast Car        10
```

```
concert_venues
```

```
# A tibble: 6 x 4
  artist        date       venue             audience_size
  <chr>         <date>     <chr>                     <dbl>
1 Tracy Chapman 1988-08-01 The Greek                  6000
2 Tracy Chapman 1990-07-05 SummerStage               15000
3 Tracy Chapman 1995-09-10 Red Rocks                  9000
4 Luke Combs    2023-02-15 The Greek                  6000
5 Luke Combs    2023-06-10 Bridgestone Arena         19000
6 Luke Combs    2024-07-20 Ford Field                45000
```

Write code that will output the following table. After piping from the table you begin with, you may only use *one* function.

```
concert_venues |>
  inner_join(setlists, by = c("artist", "date"))
```

```
# A tibble: 5 x 6
  artist       date       venue           audience_size song     set_order
  <chr>        <date>     <chr>                   <dbl> <chr>        <dbl>
1 Tracy Chapman 1988-08-01 The Greek                6000 Fast Car        1
2 Tracy Chapman 1990-07-05 SummerStage             15000 Fast Car       18
3 Tracy Chapman 1995-09-10 Red Rocks                9000 Fast Car       18
4 Luke Combs    2023-02-15 The Greek                6000 Fast Car        6
5 Luke Combs    2023-06-10 Bridgestone Arena       19000 Fast Car        1
```

3.8 Coordinating changes

```
states <- read_sf("../data/state.geojson")

states |>
  filter(name == "Virginia") |>
  st_transform(crs = 4326) |>
  ggplot() +
  geom_sf()
```

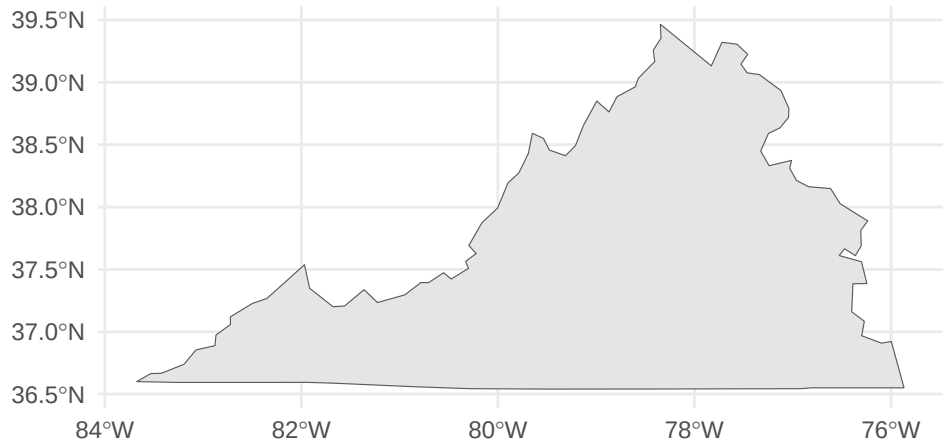


Figure 1: First Virginia map

```
states |>
  filter(name == "Virginia") |>
  st_transform(crs = 5069) |>
  ggplot() +
  geom_sf()
```

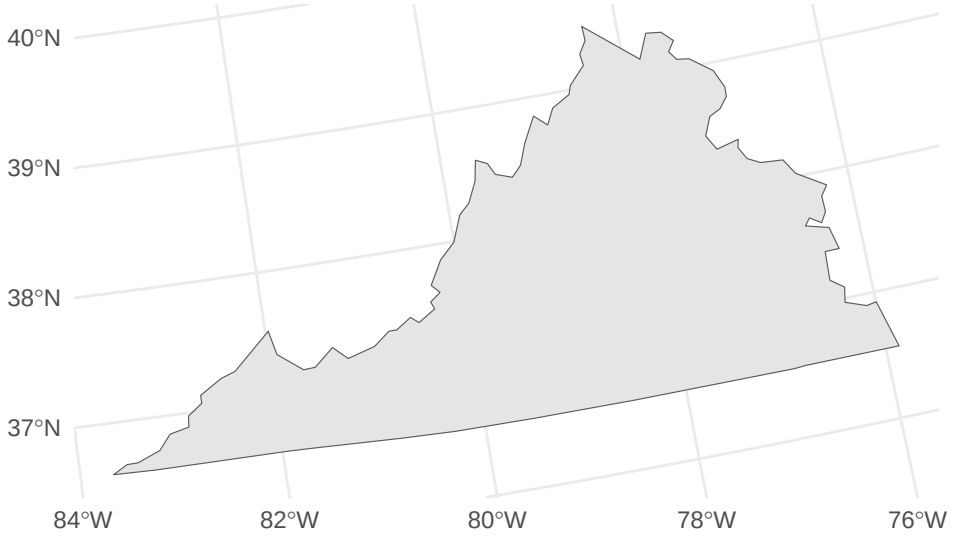
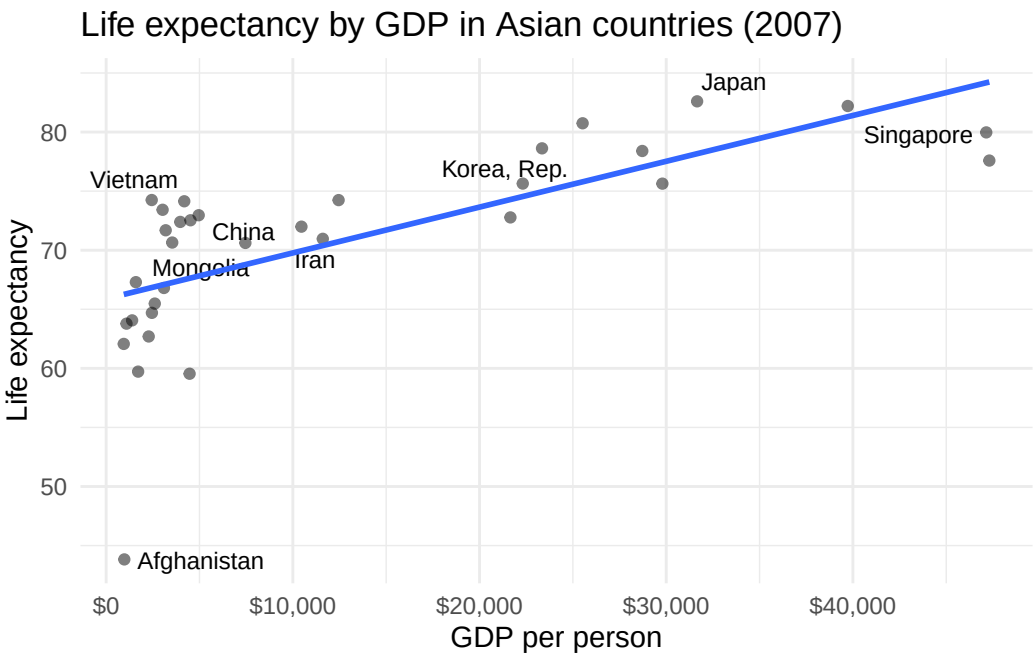


Figure 2: Second Virginia map

These plots use the same geospatial data. Only one line of the code has been changed between them. What was it? Why does that change matter for mapmaking? You do not need to write the *exact* line of code, but you do need to explain its function.

3.9 Interpret residuals



Among the labeled points, which residual has the largest absolute value? Interpret what the residual means for that point with respect to the linear model.

3.10 Extract values from `<dttm>`

```
destinys_child
```

```
# A tibble: 3 x 2
  name              birth_datetime
  <chr>             <dttm>
1 Beyoncé Knowles   1981-09-04 03:59:05
```

```
2 Kelly Rowland     1981-02-11 12:30:28
3 Michelle Williams 1979-07-23 23:01:44
```

Draw the output of the following code:

```
destinys_child |>
  mutate(
    birth_year = year(birth_datetime),
    birth_month = month(birth_datetime, label = TRUE, abbr = TRUE),
    birth_hour = hour(birth_datetime)
  ) |>
  select(name, birth_year, birth_month, birth_hour)
```

```
# A tibble: 3 x 4
  name              birth_year birth_month birth_hour
  <chr>                  <dbl> <ord>            <int>
1 Beyoncé Knowles         1981 Sep                  3
2 Kelly Rowland           1981 Feb                 12
3 Michelle Williams       1979 Jul                 23
```

3.11 Coffee shop

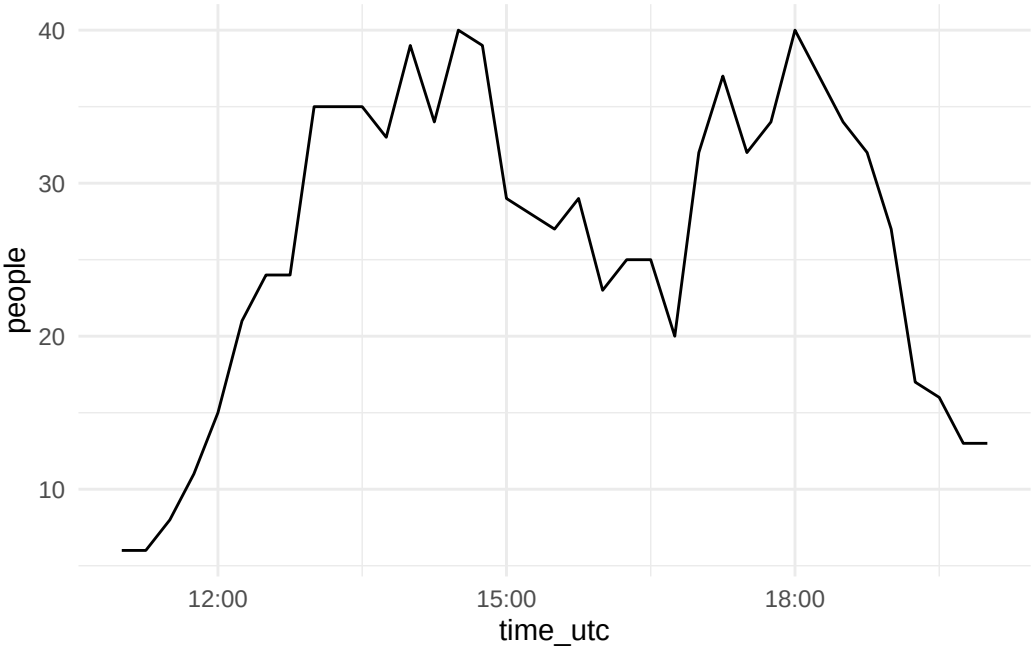
This dataset observes the number of people in a Richmond, VA coffee shop in 15 minute intervals over the business day, which runs from 6 AM to 3 PM.

```
coffee |>
  slice_head(n = 3)
```

```
# A tibble: 3 x 2
  time_utc            people
  <dttm>               <dbl>
```

```
1 2024-11-15 11:00:00      6
2 2024-11-15 11:15:00      6
3 2024-11-15 11:30:00      8
```

Plotting `coffee` gives the following:



What, if anything, is wrong with this plot? How would you fix it? You do not need to write the *exact* line of code, but you do need to explain its function.

4 Honor

> "I pledge that I neither gave nor received unauthorized assistance during the completion of this work."

Signature_____

5 Question values

Question Number	Points
1	5
2	4
3	4
4	4
5	5
6	5
7	5
8	5
9	5
10	4
11	4