Exam 01 (in class)

DSST 289: Introduction to Data Science

1 Honor

You may only use a pen/pencil and scratch paper on this exam.

> "I pledge that I will neither give nor receive unauthorized assistance during the completion of this work."

Name____

Signature____

Section start time____________________________________

2 Exam

Please write neatly.

If you cannot solve a problem, write what you *do* know about the question to maximize partial credit. For example, you could write something like, "I need the function that adds a new column to a table here, but I don't remember its name."

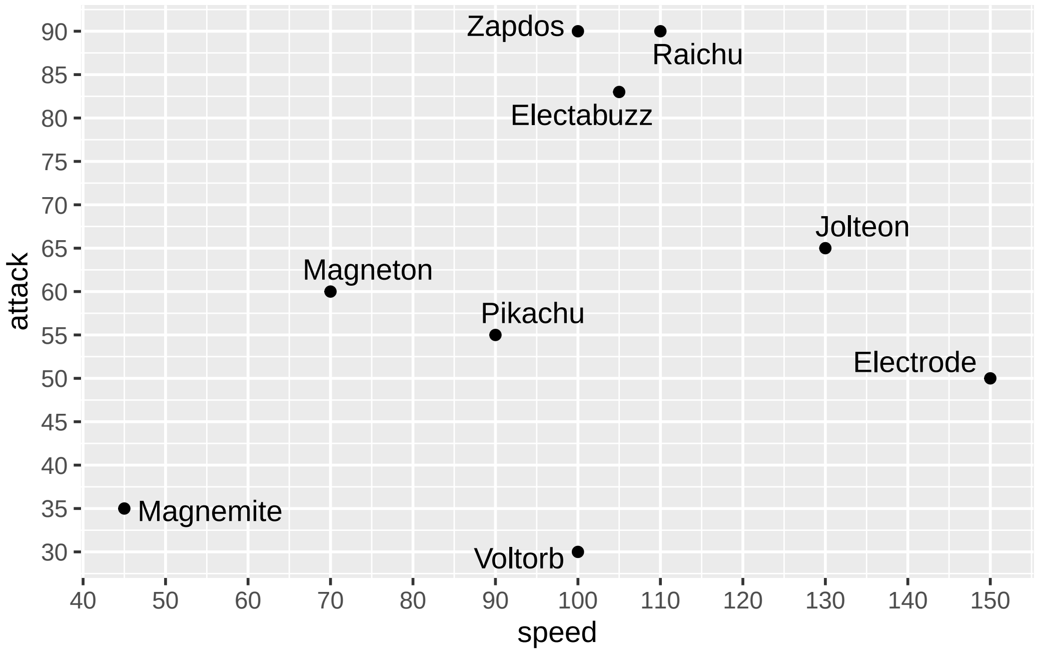
Your code will be graded on its *quality*, which includes both accuracy and formatting. In addition to the other formatting rules we have discussed, don't forget to add vertical spaces if a line would otherwise exceed approximately 80 characters in length.

For this exam, we'll be working with the data about Pokémon that we discussed in lecture.

2.1 Reconstruct tabular data

Based solely on the information available in the following plot, draw a table containing the data that produced this plot. (The labels are Pokémon names.)

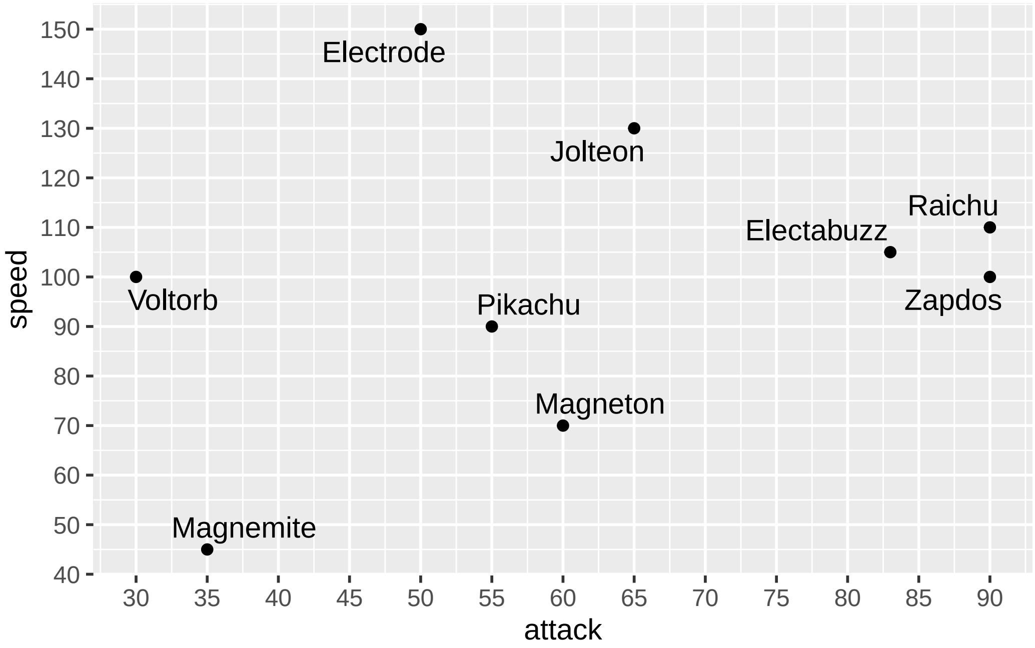
```
library(tidyverse)
library(ggrepel)
library(knitr)

pokemon <- read.csv("/Users/erik/code/dsst289-2024/data/pokemon.csv")

pokemon |>
  filter(generation == 1) |>
  filter(type_1 == "Electric") |>
  ggplot(aes(x = speed, y = attack)) +
  geom_point() +
  geom_text_repel(aes(label = name)) +
  scale_x_continuous(n.breaks = 15) +
  scale_y_continuous(n.breaks = 15)
```



2.2 Swap axes

Assume that you have the table you created in the previous question stored in a variable called `pokemon_electric`. Write code that would *swap* the `x` and `y` axes of the previous plot, as shown below. The plot should otherwise remain the same. (Hint: The number of breaks in each scale is 15.)

```r
pokemon |>
  filter(generation == 1) |>
  filter(type_1 == "Electric") |>
  ggplot(aes(x = attack, y = speed)) +
  geom_point() +
  geom_text_repel(aes(label = name)) +
  scale_x_continuous(n.breaks = 15) +
  scale_y_continuous(n.breaks = 15)
```



2.3 More aesthetics

```
pokemon_sample <- pokemon |>
  filter(generation == 1) |>
  filter(type_1 %in% c("Electric", "Rock", "Psychic"))

pokemon_sample |>
  ggplot(aes(x = attack, y = speed)) +
  geom_point(aes(color = type_1, size = stat_total)) +
  geom_text_repel(aes(label = name, color = type_1)) +
  scale_color_viridis_d() +
  scale_x_continuous(n.breaks = 15) +
  scale_y_continuous(n.breaks = 15)
```

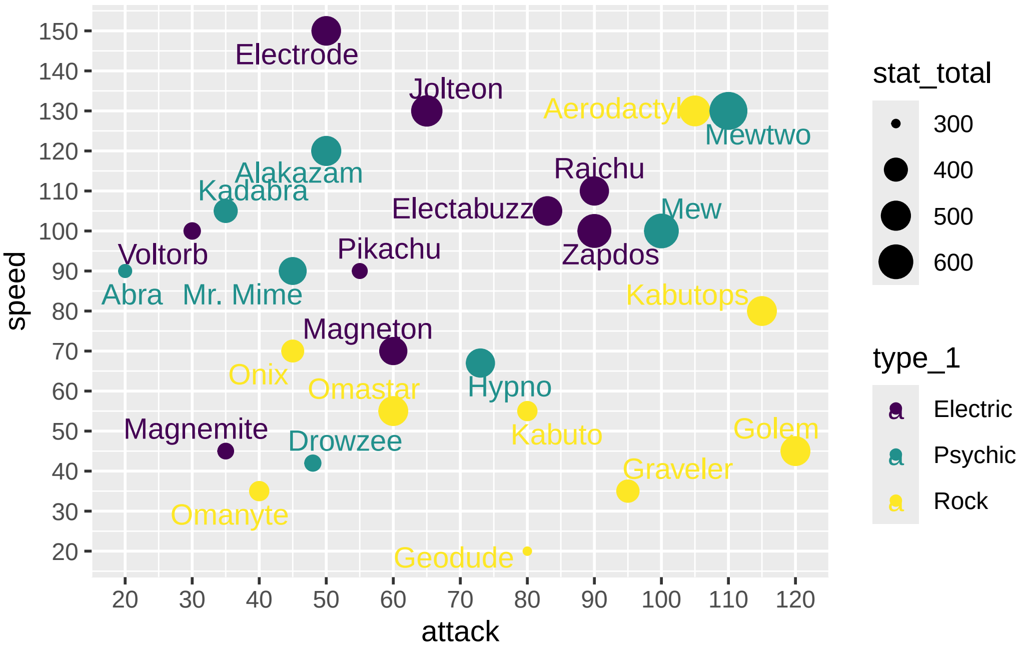


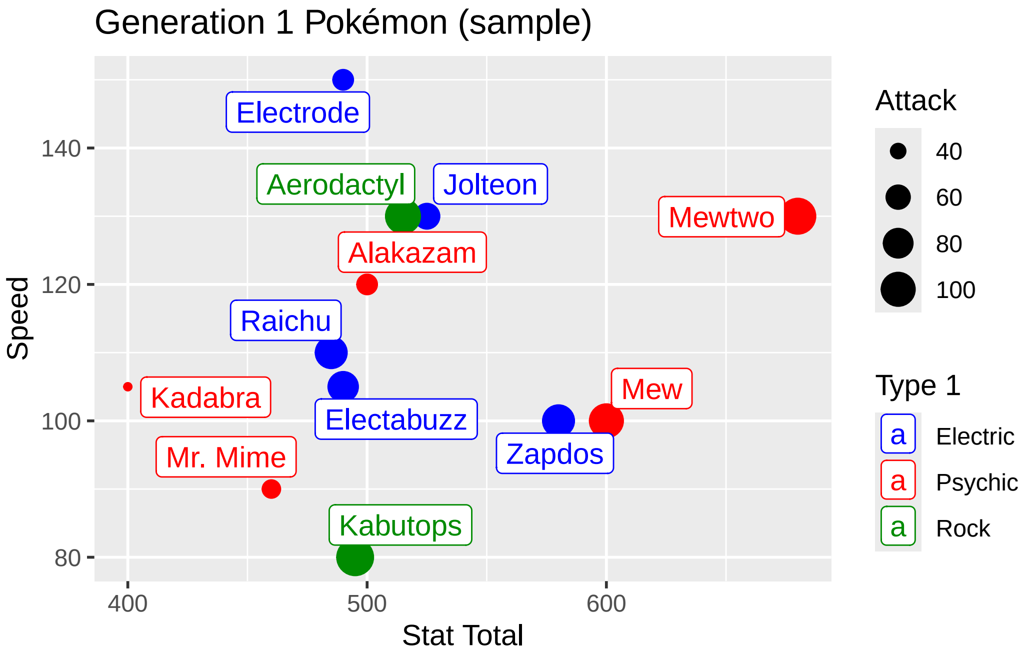
Figure 1: Speed and attack of Electric, Psychic, and Rock Pokémon of Generation 1.

Assuming that a variable called `pokemon_sample` contains the data needed for the plot above, fill in the blanks in the following code. Rewrite the code in the blank part of the page if you need more room.

```
pokemon_sample |>
  ggplot
  geom_
  geom_
  scale_
  scale_x_
  scale_y_
```

2.4 Plot variations

```r
pokemon_sample |>
  filter(speed >= 80 & stat_total >= 400) |>
  ggplot(aes(x = stat_total, y = speed)) +
  geom_point(aes(color = type_1, size = attack)) +
  geom_label_repel(aes(label = name, color = type_1)) +
  scale_color_manual(values = c("blue", "red", "green4")) +
  labs(title = "Generation 1 Pokémon (sample)",
       x = "Stat Total",
       y = "Speed",
       color = "Type 1",
       size = "Attack")
```



This plot also uses the data in `pokemon_sample`. List **five** differences between this plot and the previous plot.

When you observe the differences, indicate either *what* was changed or *where* the change was made (e.g., in `geom_...`, in `scale_...`, etc.)

Only list five differences. No extra credit for additional answers. You may answer in code, in sentences, or a mix. Do whatever is easiest.

2.5 Subsetting data

Below are ten random rows from the 1,194 rows in the `pokemon` data set containing the variables that we will use for the remainder of this exam:

```
pokemon |>
  slice_sample(n = 10) |>
  select(name, generation, type_1, type_2, stat_total, attack, speed) |>
  kable()
```

name	generation	type_1	type_2	stat_total	attack	speed
Pincurchin	8	Electric		435	101	15
Mewtwo	1	Psychic		680	110	130
Houndour	2	Dark	Fire	330	60	65
Loudred	3	Normal		360	71	48
Pinsir	1	Bug		500	125	85
Bagon	3	Dragon		300	75	50
Togedemaru	7	Electric	Steel	435	98	96
Ariados	2	Bug	Poison	400	90	40
Lillipup	5	Normal		275	60	55
Gogoat	6	Grass		531	100	68

- Write code that would filter the complete `pokemon` data set sampled above to include *only* the Pokémon in the plot under the header **"2.3 More Aesthetics."**
- Save the results of your filtering steps into a variable called `pokemon_sample`.
- *Nota bene*:
  - Only **one** of the numeric columns has been filtered. The filtered column is identified in plots **2.3** and **2.4**.
  - Do **not** use Pokémon names to select the correct Pokémon. There are far more efficient approaches.

```
pokemon_sample <- pokemon |>
  filter(generation == 1) |>
  filter(type_1 %in% c("Electric", "Rock", "Psychic")) |>
  # this is unnecessary as I don't show all columns:
  select(name, generation, type_1, type_2, stat_total, attack, speed)
```

2.6 Highest stat_total in ascending order

Sort pokemon_sample by stat_total, with the lowest values first. Output only the first five rows as shown below:

```
pokemon_sample |>
  arrange(stat_total) |>
  slice_head(n = 5) |>
  select(name, stat_total) |>
  kable()
```

name	stat_total
Geodude	300
Abra	310
Pikachu	320
Magnemite	325
Drowzee	328

2.7 Highest speed by type_1

Using pokemon_sample, get the Pokémon with the highest speed *within each* type_1. Write code to reproduce the following table:

```
pokemon_sample |>
  group_by(type_1) |>
  filter(speed == max(speed)) |>
  select(type_1, name, speed) |>
  kable()
```

type_1	name	speed
Electric	Electrode	150
Rock	Aerodactyl	130
Psychic	Mewtwo	130

2.8 Pokémon on average by type

Using pokemon_sample, calculate the average speed, attack, and stat_total for each type_1. Write code to reproduce the following table:

```r
pokemon_sample |>
  group_by(type_1) |>
  # students don't have to include round(); just for printing:
  summarize(
    avg_speed = round(mean(speed), 1),
    avg_attack = round(mean(attack), 1),
    avg_stat_total = round(mean(stat_total), 1)
  ) |>
  kable()
```

type_1	avg_speed	avg_attack	avg_stat_total
Electric	100.0	62.0	445.6
Psychic	93.0	60.1	470.1
Rock	58.3	82.2	420.6

2.9 Tallying Pokémon by types

Using pokemon_sample, tally Pokémon by type_1 *and* type_2. Write code to reproduce the following table:

```r
pokemon_sample |>
  count(type_1, type_2) |>
  kable()
```

type_1	type_2	n
Electric		6
Electric	Flying	1
Electric	Steel	2
Psychic		7
Psychic	Fairy	1
Rock	Flying	1
Rock	Ground	4
Rock	Water	4