Check-in 8: Answers

Insert answers within the code chunks. Unless specified otherwise, do not assign your output to an object. If directed to assign output to an object, wrap the pipe in parentheses to print the output.

For the following problems, run this code to create `dog_data`:

```
dog_data <- read_csv("https://decisionslab.unl.edu/data/stevens_etal_2020_obed_data1.csv") |>
  select(class, dog_sex, dias_overall_score, latency_sit_mean, starts_with("cort"))
```

Problem 1: (4 pts) Let's look at the distribution of the time that it took the dogs to sit after hearing the command (`latency_sit_means`). First, find the maximum latency and add 1 to get the total number of bins that we want in our histogram (there are latencies of 0, so we need an extra bin for them). Then create a histogram of the latency counts, with the area of all bars colored *steelblue1* and the borders of the bars colored *steelblue*.

Problem 2: (3 pts) Adjust the `class` column to remove the *S18* data and put the other levels in the following order: *U18*, *F18*, *S19*, *U19*, *F19*. Reassign the output to `dog_data`.

Problem 3: (3 pts) Create a boxplot of `dias_overall_score` as a function of `class`, making the lines of the boxplot *grey60*. Then overlay the mean and standard error over the boxplot, using a ggplot function to calculate the mean and standard error. Class should be ordered as specified in Problem 2.

Problem 4: (7 pts) Now make a violin plot of `dias_overall_score` as a function of `class`. Make the areas of the violins colored based on class and transparent at level 0.25 and make the borders *grey60*. Add the raw data points **under** the violins, make them *grey60*, but don't let the points show up in the legend. Use a coordinate function to switch the axes.

Problem 5: (4 pts) Remove from the data observations when `dog_sex` is *NA*. Then create a bar plot where ggplot counts the number of observations for each class (using a geom that calculates the counts). Plot this with `dog_sex` as groups with different colored areas and plot *Male* and *Female* side-by-side. *Note that U19 and F19 have only one sex in each, and they look a little funny. Don't worry about that.*

Problem 6: (5 pts) Create a scatterplot of `cort3` and `cort4` where points are colored based on `dog_sex`. Assign *Female* to color #E69F00 and *Male* to #56B4E9 (NA can use the default color). Include a reference line with slope 1 and maintain a 1:1 aspect ratio.

Problem 7: (4 pts) First run the following code to create a new data frame with dog id, dog sex, and two cort measurements.

```
dog_cort <- dog_data |>
  mutate(id = 1:nrow(dog_data)) |>
  select(id, dog_sex, cort3, cort4) |>
  filter(!is.na(dog_sex) & !is.na(cort3) & !is.na(cort4))
```

Now reshape the data fKrame where `cort3` and `cort4` are rotated to long format with the name column called `time` and the values column called `cort`. Then create a slopegraph with `time` on the x axis, `cort` on the y axis, and lines connecting the times based on `id`.

Problem 8: (7 pts) Based on the reshaped data from Problem 7, plot an interaction plot with mean $\pm$ 95% confidence intervals of `cort` with `time` (cort3 and cort4) on the x-axis and separately colored lines for `dog_sex`. Include lines connecting times within each dog sex level. Make sure to avoid overlap in dots and error bars.