

Check-in 7

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.

First, import `obed_data.csv` from your `checkins` directory (or [download](#) if needed) and save to `dog_data`.

Problem 1: (4 pts) Do the following:

1. Plot a scatterplot of log-transformed `cort1` and `cort2` from `dog_data`. Do the transformation inside the `ggplot`.
2. Overlay a linear regression line on top of the points.
3. Color the points `aquamarine4`, the line `dodgerblue4`, and the confidence band `dodgerblue3`.
4. Make the background of the plot white.

Problem 2: (6 pts)

1. Log transform `cort1` and `cort2` in the data, creating `logcort1` and `logcort2`.
2. Take the mean of `latency_sit_mean` and `latency_down_mean` to create a `latency` variable.
3. Remove observations with NA for `latency`.
4. Plot a scatterplot of `logcort1` and `logcort2`.
5. Color the points based on `latency` using a viridis color scale.
6. Move the legend above the plot.

Problem 3: (4 pts)

1. Plot `mdors_score` by `dias_overall_score`.
2. Color the points using the *Dark2* color palette and set the shape based on `class`.
3. Reorder the levels of `class` to chronological order: S18, U18, F18, S19, U19, F19
4. Remove the minor grid lines.

Problem 4: (5 pts) Do the following:

1. Plot `dias_overall_score` by `time_train_dog_weekly_num` as black points.
2. Overlay separate lines for `dog_sex` that are different colors.
3. Remove the confidence bands.
4. Move the legend inside the plot.
5. Set the aspect ratio to 1.

Problem 5: (5 pts) Do the following:

1. Plot the mean and confidence intervals of `cort1` for each level of `class` (calculate the mean and confidence interval inside the ggplot).
2. Color the dots and lines based on `class`.
3. Create separate subplots based on `dog_sex`.
4. Remove the legend.
5. Use the bw theme.

Problem 6: (3 pts) In your own creative way, I would like you to **extend** beyond what the previous questions have asked you to do. Create and customize your own plot of this data set or another data set that:

1. Adjusts a non-color property of the data (either inside or outside of `aes()`)
2. Adjusts a theme element
3. Applies a non-default color scale to the data