

Socio-Informatics 348

Practical 4

Submission Instructions

- Submit your completed practical as `studentnumber.qmd` on SocSciLearn.
- Submissions are checked for completeness, not correctness.
- At least 80% of exercises must be attempted to receive 1% towards AF assessment.
- Attendance of at least one practical session per week is required to earn the 1% for that week's practical.

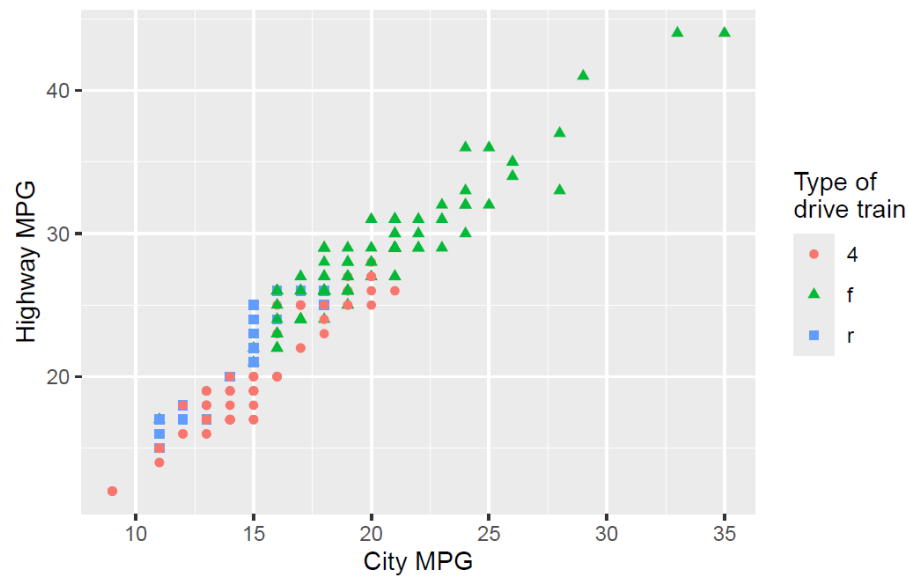
Deadline

Friday 5 September, 17:00 (submit on SocSciLearn)

Exercises

Section 1: mpg dataset from ggplot2 package

1. Using the `mpg` dataset that comes with `ggplot2`, create one plot on the fuel economy data with customised title, subtitle, caption, x, y, and colour labels.
2. Recreate the following plot using the fuel economy data (the `mpg` dataset). Note that both the colours and shapes of points vary by type of drive train (`drv`).



3. Combine the three plots provided into the patchwork layout shown below. To add figure labels (e.g., “Fig. A”, “Fig. B”), check the package documentation for functions such as `plot_annotation()` using `?plot_annotation` in R or by searching online.

```
p1 <- ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point() +
  labs(title = "Plot 1")
p2 <- ggplot(mpg, aes(x = drv, y = hwy)) +
  geom_boxplot() +
  labs(title = "Plot 2")
p3 <- ggplot(mpg, aes(x = cty, y = hwy)) +
  geom_point() +
  labs(title = "Plot 3")
```

Fig. A:

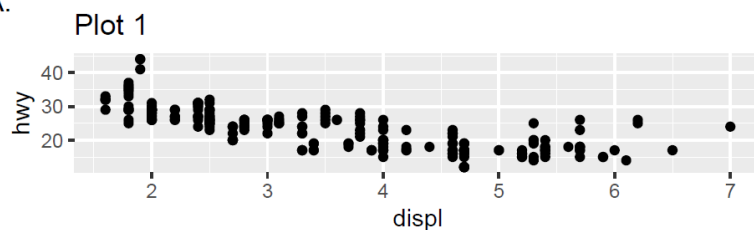


Fig. B:

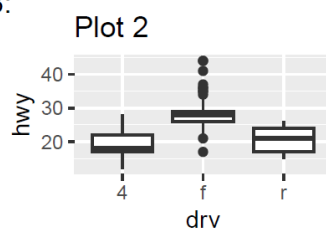
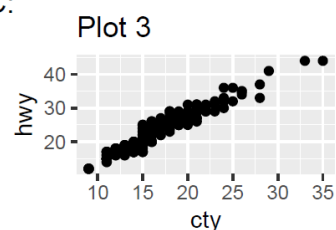


Fig. C:



Section 2: flights dataset from nycflights13 package

4. How many flights have a missing `dep_time`? What other variables are missing? What might these rows represent?
5. What time of day should you fly if you want to avoid delays as much as possible?
6. Create a new variable called `dep_period` that categorises each flight's `dep_time` into:
"morning" for departures from 05:00, but before 12:00 (noon),
"afternoon" for departures between 12:00 and 18:00,
"evening" for departures between 18:00 and 22:00,
"late night" for all other departures.

Use `case_when()` to build this variable. Then, calculate the average departure delay for each `dep_period`.

7. Repeat the task above, but this time create the same `dep_period` variable using nested `ifelse()` statements instead of `case_when()`. Compare the readability of your solution with the previous one.
8. Use `cut()` on `dep_time` to categorise flights into the same `dep_period` variable used previously. Calculate the average departure delay for each `dep_period`.