FPCP2

STAT 244

Eonbi Choi, Yerim Oh

Data

```
load("data/lap_dat.Rdata")
head(lap_dat)
```

```
# A tibble: 6 x 32
   time driver driver_number lap_time lap_number stint pit_out_time pit_in_time
  <dbl> <chr>  <chr>            <dbl>      <dbl> <dbl>        <dbl>       <dbl>
1 3438. VER    1                 94.3          1     1          NaN         NaN
2 3531. VER    1                 93.1          2     1          NaN         NaN
3 3624. VER    1                 93.1          3     1          NaN         NaN
4 3717. VER    1                 93.5          4     1          NaN         NaN
5 3810. VER    1                 92.8          5     1          NaN         NaN
6 3903. VER    1                 92.9          6     1          NaN         NaN
# i 24 more variables: sector1time <dbl>, sector2time <dbl>, sector3time <dbl>,
#   sector1session_time <dbl>, sector2session_time <dbl>,
#   sector3session_time <dbl>, speed_i1 <dbl>, speed_i2 <dbl>, speed_fl <dbl>,
#   speed_st <dbl>, is_personal_best <list>, compound <chr>, tyre_life <dbl>,
#   fresh_tyre <lgl>, team <chr>, lap_start_time <dbl>, lap_start_date <dttm>,
#   track_status <chr>, position <dbl>, deleted <lgl>, deleted_reason <chr>,
#   fast_f1generated <lgl>, is_accurate <lgl>, session_type <chr>
```

Part 1: Data Context

1. Variables in the data set that are interesting

quantitative variable:

- `lap_time`: recorded time to complete a lap (seconds)

- `lap_number`: lap number from which the telemetry data was recorded (number of laps)
- `tyre_life`: number of laps completed on a set of tires (number of laps)

categorical variable:

- `compound`: type of tire used (SOFT, MEDIUM, HARD)
- `pit_in`: whether a driver made a pit stop during a lap (binary: 0 = no pit stop, 1 = pit stop occured)

2. One observational unit (row) represent in the data set

```
head(lap_dat,1)
```

```
# A tibble: 1 x 32
   time driver driver_number lap_time lap_number stint pit_out_time pit_in_time
  <dbl> <chr>  <chr>            <dbl>      <dbl> <dbl>        <dbl>       <dbl>
1 3438. VER    1                 94.3          1     1          NaN         NaN
# i 24 more variables: sector1time <dbl>, sector2time <dbl>, sector3time <dbl>,
#   sector1session_time <dbl>, sector2session_time <dbl>,
#   sector3session_time <dbl>, speed_i1 <dbl>, speed_i2 <dbl>, speed_fl <dbl>,
#   speed_st <dbl>, is_personal_best <list>, compound <chr>, tyre_life <dbl>,
#   fresh_tyre <lgl>, team <chr>, lap_start_time <dbl>, lap_start_date <dttm>,
#   track_status <chr>, position <dbl>, deleted <lgl>, deleted_reason <chr>,
#   fast_f1generated <lgl>, is_accurate <lgl>, session_type <chr>
```

Each observational unit represents all records from one lap

3. How the sample was obtained

- The Formula 1 data used in this study are obtained from the f1dataR R package that accesses Formula 1 data via the FastF1 Python library. The dataset includes lap-by-lap session data from the 2024 Miami Grand Prix and comprise 1111 laps and 32 variables.
- Description: https://cran.r-project.org/web/packages/f1dataR/f1dataR.pdf
- Data sources: Obtain Formula 1 data via the unofficial API and the 'fastf1' 'Python' library.
- Last accessed date/time: April 28, 2025 16:51 PM

4. Ulterior motive for collecting the data

Could the data collectors have an ulterior motive for collecting the data (e.g., solicitation of private information)? Could the data collectors have an ulterior motive for collecting a biased sample, or otherwise misrepresenting the population in any way (e.g., trying to reinforce a predetermined narrative)?

- No, the data is collected from the telemetry systems, that includes sensors connected to the machine (cars in Formula 1) and wireless transmission through multiple networks.

5. Reliable Source

Do you think the source of your data is reliable? Do you trust how the data was collected?

- Yes, the package uses data from Formula 1's live timing services. There could still be minor errors because of the connected sensors.

Part 2: Data Cleaning

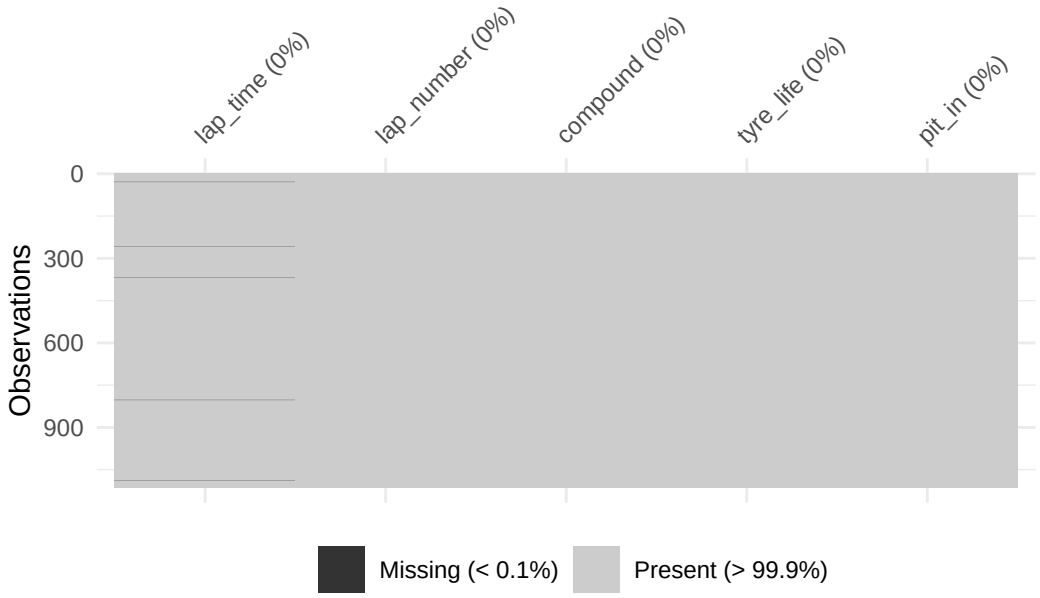
1. Clean/Rearrange Data

```
# clean data
miami2024 <- lap_dat %>%
  select(lap_time, lap_number, compound, tyre_life) %>%
  mutate(compound = as.factor(compound),
         pit_in = ifelse(is.nan(lap_dat$pit_in_time), 0, 1))
head(miami2024)
```

```
# A tibble: 6 x 5
  lap_time lap_number compound tyre_life pit_in
     <dbl>      <dbl> <fct>        <dbl>  <dbl>
1     94.3          1 MEDIUM           1      0
2     93.1          2 MEDIUM           2      0
3     93.1          3 MEDIUM           3      0
4     93.5          4 MEDIUM           4      0
5     92.8          5 MEDIUM           5      0
6     92.9          6 MEDIUM           6      0
```

2. Check missing values

```
vis_miss(miami2024)
```




```
#calculate extend of missingness
sum(is.na(miami2024$lap_time))
```

```
[1] 5
```

Data for `lap_time` are missing five values which are less than 0.1% of the entire observation.

```
# drop missing values
miami2024_complete <- na.omit(miami2024)

dim(miami2024_complete)
```

```
[1] 1106    5
```

number of observational units: **1106**

3. Why certain data points are missing

Out of 5 missing lap time records four records have a track status code of 41. However, no description of this code value is provided in the API. Thus, we assume that either the track was not fully cleared or conditions were not suitable for racing. The other missing record was due to a driver failing to complete a lap due to collision.

Part 3: Exploratory Data Analysis

1. Numerical summaries that are relevant

Quantitative variables

lap_time:

```
summary(miami2024_complete$lap_time)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  90.63   92.38   93.28   96.00   94.29  148.74
```

```
sd(miami2024_complete$lap_time)
```

```
[1] 8.884343
```

```
var(miami2024_complete$lap_time)
```

```
[1] 78.93155
```

lap_number:

```
summary(miami2024_complete$lap_number)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00   14.00   28.00   28.62   43.00   57.00
```

tyre_life:

```r
summary(miami2024_complete$tyre_life)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    7.00   13.50   14.78   22.00   45.00
```

```r
sd(miami2024_complete$tyre_life)
```

```
[1] 9.375079
```

```r
var(miami2024_complete$tyre_life)
```

```
[1] 87.89211
```

Categorical variables

compound:

```r
counts(miami2024_complete$compound)
```

```
  n_HARD n_MEDIUM   n_SOFT
     500      562       44
```

```r
props(miami2024_complete$compound)
```

```
  prop_HARD prop_MEDIUM   prop_SOFT
  0.4520796   0.5081374   0.0397830
```

pit_in:

```r
counts(miami2024_complete$pit_in)
```

```
 n_0  n_1
1078   28
```

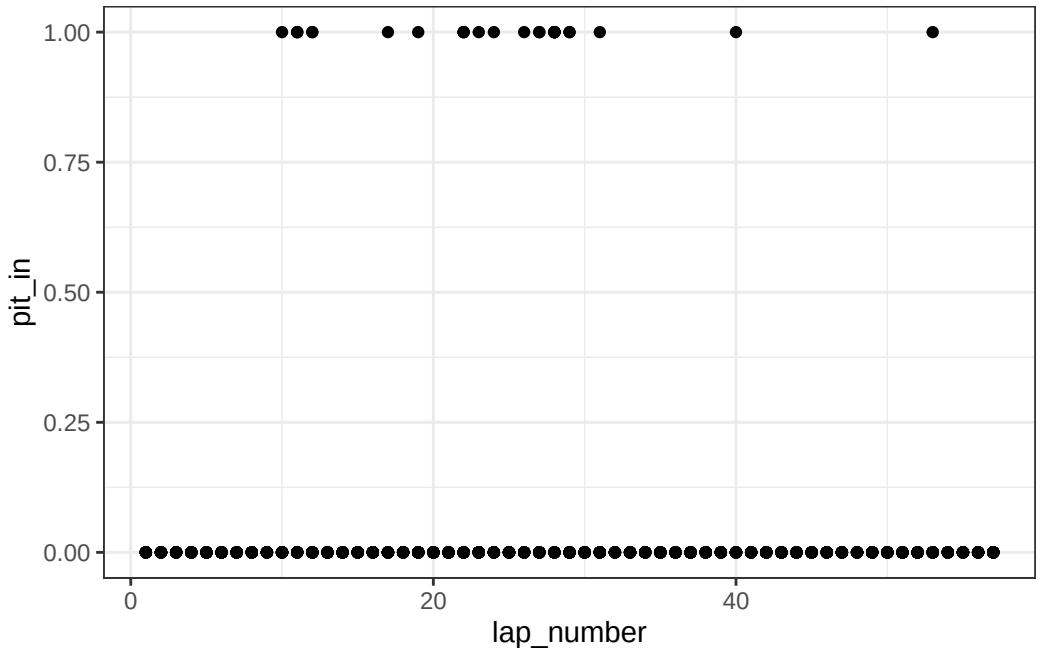
```
props(miami2024_complete$pit_in)
```

```
    prop_0     prop_1
0.97468354 0.02531646
```

2. Data visualizations

Our response variable: `pit_in`

```
ggplot(data = miami2024_complete,
       mapping = aes(x = lap_number, y = pit_in)) +
  geom_point() +
  theme_bw()
```

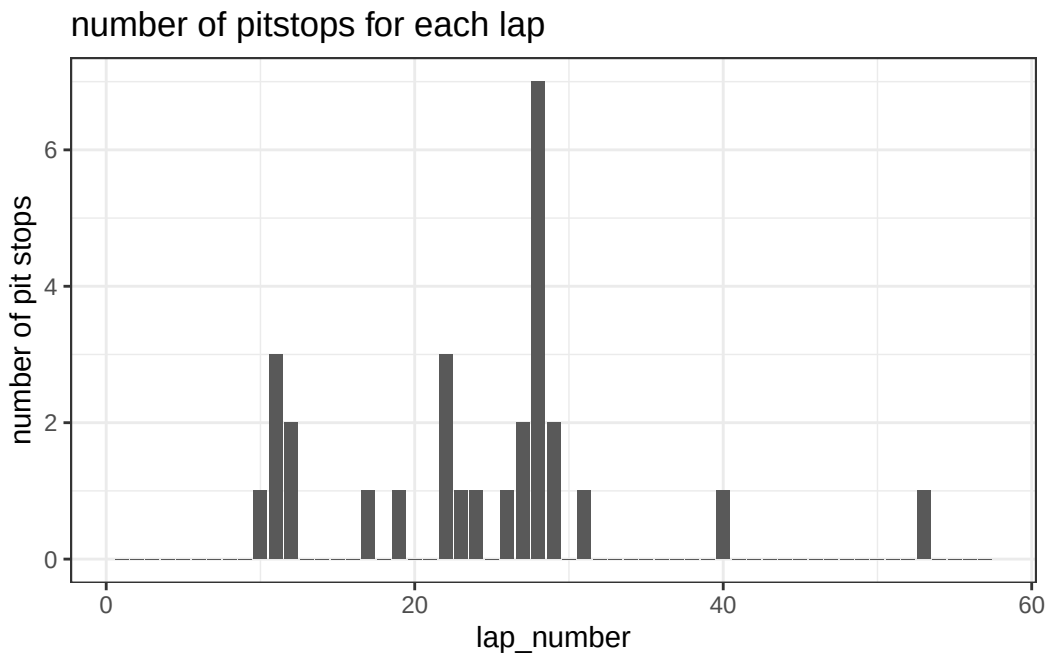


Relationship of `lap_number` and `pit_in`

```
lapnum_pit <- data.frame(lap_num = rep(NA, 57),
                         pit_num = rep(NA, 57))
for (i in 1:57){
  lapnum_pit$lap_num[i] <- i
  lapnum_pit$pit_num[i] <- count(miami2024_complete$pit_in == 1 &
                                 miami2024_complete$lap_number == i)
```

```
}

ggplot(lapnum_pit, aes(x = lap_num, y = pit_num)) +
  geom_bar(stat = "identity") +
  labs(title = "number of pitstops for each lap",
       x = "lap_number", y = "number of pit stops") +
  theme_bw()
```



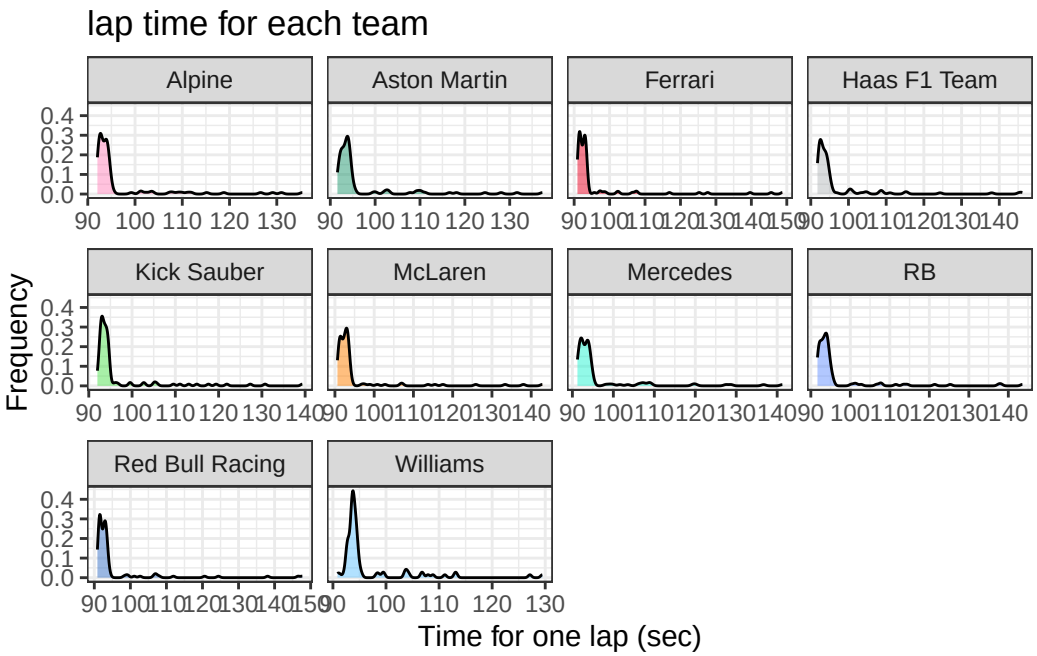
Density plot for `lap_time`

```
cols <- c("#FF87BC", "#229971", "#E80020", "#B6BABD", "#52E252",
          "#FF8000", "#27F4D2", "#6692FF", "#3671C6", "#64C4FF")

lap_dat %>%
  ggplot(aes(x=lap_time, fill=team)) +
  geom_density(colour="black", alpha=0.5, show.legend=FALSE) +
  facet_wrap(~team, scales="free_x") +
  scale_fill_manual(values = cols) +
  labs(x = "Time for one lap (sec)", y = "Frequency",
       title = "lap time for each team") +
  theme_bw()
```

```
Warning: Removed 5 rows containing non-finite outside the scale range
```

(`stat_density()`).



Box plot of `compound` vs `tyre_life`

```
# new data that stores the tyre life at each pit stop
compound_life <- data.frame(compound = character(),
                            tyre_life = double())
for (i in 1:nrow(miami2024_complete)){
  if (miami2024_complete$pit_in[i] == 1) {
    compound_life <- compound_life %>%
      add_row(compound = miami2024_complete$compound[i],
              tyre_life = miami2024_complete $tyre_life[i])
  }
}
head(compound_life)
```

```
  compound tyre_life
1   MEDIUM        23
2   MEDIUM        12
3   MEDIUM        17
4     HARD        11
5     HARD        23
6   MEDIUM        19
```

```r
ggplot(compound_life, aes(x = compound, y = tyre_life, fill=compound)) +
  geom_boxplot() +
  labs(x = "Tyre compound", y = "Tyre life",
       title = "Tyre life for each compound") +
  scale_fill_manual(values = c("#f0f0f0", "#edde09", "#ed0909")) +
  theme_bw()
```