Example of a computational notebook for data analysis

In this example, we explore life history data on 2270 lemur individuals living in the Duke Lemur Center.

Data

```
lemurs_rawdf <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/da
                          col_types = cols(
                            .default = col_double(),
                            taxon = col_character(),
                            dlc_id = col_character(),
                            hybrid = col_character(),
                            sex = col_character(),
                            name = col_character(),
                            current_resident = col_character(),
                            stud_book = col_character(),
                            dob = col_date(format = ""),
                            estimated_dob = col_character(),
                            birth_type = col_character(),
                            birth_institution = col_character(),
                            estimated_concep = col_date(format = ""),
                            dam_id = col_character(),
                            dam_name = col_character(),
                            dam_taxon = col_character(),
                            dam_dob = col_date(format = ""),
                            sire_id = col_character(),
                            sire_name = col_character(),
                            sire_taxon = col_character(),
                            sire_dob = col_date(format = ""),
                            dod = col_date(format = ""),
                            age_of_living_y = col_double(), ## this column is typed wrong (as character
                            dob_estimated = col_character(),
                            weight_date = col_date(format = ""),
                            age_category = col_character(),
                            preg_status = col_character(),
                            concep_date_if_preg = col_date(format = ""),
                            infant_dob_if_preg = col_date(format = "")
                            )
)

## key of species name and abbreviation in the taxon columns
lemurs_sppnames_df <- readr::read_csv(file.path(rawdata_dir, "lemurs_sppnames.csv"),
                                      col_names = TRUE)

lemurs_rawdf %>%
  head() %>%
```

Table 1: Header of the original data set on the health, reproduction, and social dynamics of lemurs housed at the Duke Lemur Center, in North Carolina, USA.

taxon	dlc_id	hybrid	sex	name	current_resident	stud_book	dob	birth_month	estimated
OGG	0005	N	M	KANGA	N	NA	1961-08-25	8	NA
OGG	0005	N	M	KANGA	N	NA	1961-08-25	8	NA
OGG	0006	N	F	ROO	N	NA	1961-03-17	3	NA
OGG	0006	N	F	ROO	N	NA	1961-03-17	3	NA
OGG	0009	N	M	POOH BEAR	N	NA	1963-09-30	9	NA
OGG	0009	N	M	POOH BEAR	N	NA	1963-09-30	9	NA

```
  kableExtra::kbl(caption = "Header of the original data set on the health, reproduction, and social dy
  kableExtra::kable_styling(c("striped", "hover")) %>%
  kableExtra::scroll_box(width = "100%", height = "300px")
```

Pre-processing

The data can be organized temporally, thanks to the `weight_date` and `month_of_weight` variables, which report the full date and the month when the weight was measured, respectively. Moreover, we do not need all the 52 variables that were measured, so let's create smaller time-series with the variables of interest.

```
lemurs_smallts <- lemurs_rawdf %>%
  dplyr::mutate_at(vars(name, dam_name, sire_name), stringr::str_to_title) %>%
  dplyr::mutate(year = lubridate::year(weight_date)) %>%
  dplyr::select(c(year, month_of_weight, ## time variables
                  taxon, dlc_id, ## id variables
                  hybrid, sex, name, birth_month, litter_size, concep_month, ## birth variables
                  dam_id, dam_name, dam_name, sire_id, sire_name, sire_taxon, ## parental history varia
                  age_at_death_y, age_of_living_y, age_last_verified_y,
                  age_max_live_or_dead_y, age_at_wt_y, age_category, ## age variables
                  weight_g, avg_daily_wt_change_g, ## weight variables
                  preg_status,
                  n_known_offspring, infant_lit_sz_if_preg)) %>%
  dplyr::rename(month = month_of_weight,
                weight = weight_g,
                avg_d_wt_chg = avg_daily_wt_change_g,
                n_offspring = n_known_offspring) %>%
  dplyr::right_join(lemurs_sppnames_df,., by = "taxon") ## id species
```

Main text figures and tables

Table 1: Fertility rates per taxon

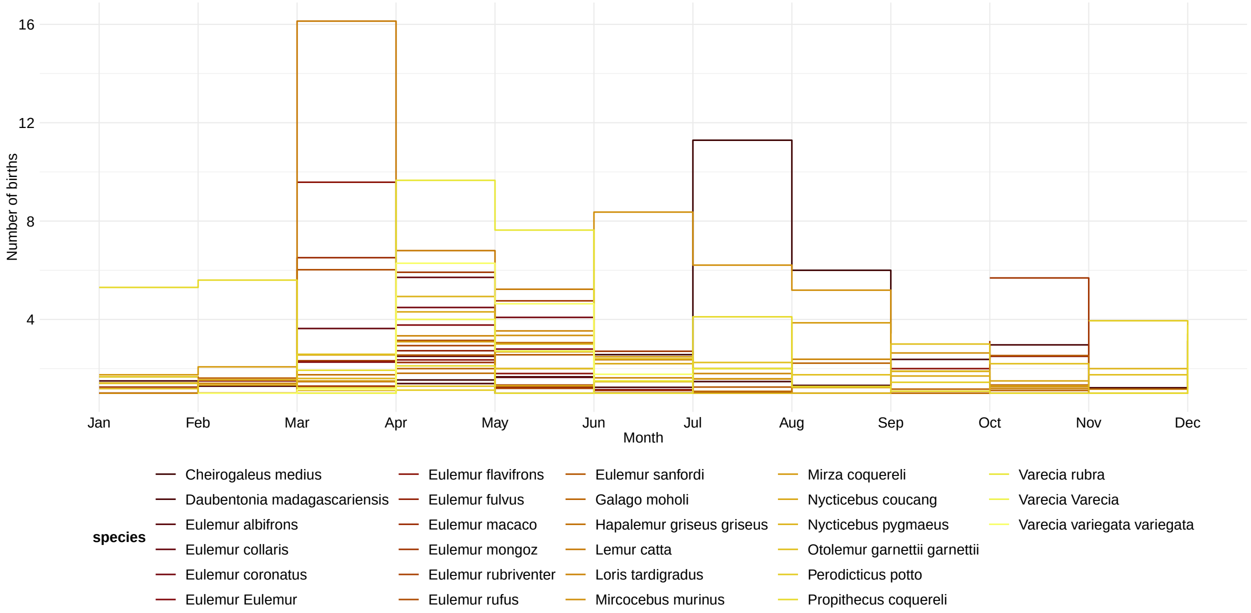
```
lemurs_smallts %>%
  dplyr::filter(!is.na(infant_lit_sz_if_preg)) %>% ## filter the animals for which this information was
  dplyr::group_by(dlc_id, species) %>%
  dplyr::summarize(inflt_mean_ind = mean(infant_lit_sz_if_preg)) %>%
  ungroup() %>%
  dplyr::group_by(species) %>%
  dplyr::summarize(inflt_mean = mean(inflt_mean_ind),
                   inflt_sd = sd(inflt_mean_ind),
                   n = n()) %>%
  dplyr::rename(Species = species,
                "Infant litter size (mean)" = inflt_mean,
                "Infant litter size (sd)" = inflt_sd) %>%
  readr::write_csv(file.path(figures_dir, "fertility_rates.csv"))
```

Seasonality of species

```
births_df <- lemurs_smallts %>%
  dplyr::select(dlc_id, taxon, birth_month, year) %>%
  dplyr::filter(!is.na(birth_month)) %>%
  dplyr::mutate_at(vars(birth_month),
                   lubridate::month, label = TRUE,
                   locale = Sys.getlocale(category = "LC_CTYPE")) %>% ## id months
  dplyr::right_join(lemurs_sppnames_df,., by = "taxon") %>% ## id species
  dplyr::arrange(taxon, birth_month)

months_fct <- lubridate::month(1:12, label = TRUE, locale = Sys.getlocale(category = "LC_CTYPE"))

births_countdf <- births_df %>%
  unique() %>%
  dplyr::group_by(species, common_name, taxon, birth_month, year) %>%
  dplyr::summarize(n_births = n()) %>%
  ungroup() %>%
  tidyr::pivot_wider(id_cols = c(species, common_name, taxon, year),
                     names_from = birth_month, values_from = n_births) %>%
  tidyr::pivot_longer(all_of(months_fct),
                      names_to = "birth_month", values_to = "n_births")%>%
  ## because strings can't be converted back to months, the following turns out quite cumbersome
  dplyr::mutate_at(vars(birth_month),
                   ~ ordered(.,
                             levels = all_of(months_fct)) %>%
                     as.numeric) %>%
  dplyr::mutate_at(vars(birth_month),
                   lubridate::month, label = TRUE,
                                       locale = Sys.getlocale(category = "LC_CTYPE")) %>%
  dplyr::arrange(year, species, birth_month)

births_summdf <- births_countdf  %>%
  dplyr::group_by(species, birth_month) %>%
  dplyr::summarize(n_births_mean = mean(n_births, na.rm = TRUE),
                 n_births_sd = sd(n_births, na.rm = TRUE))
```

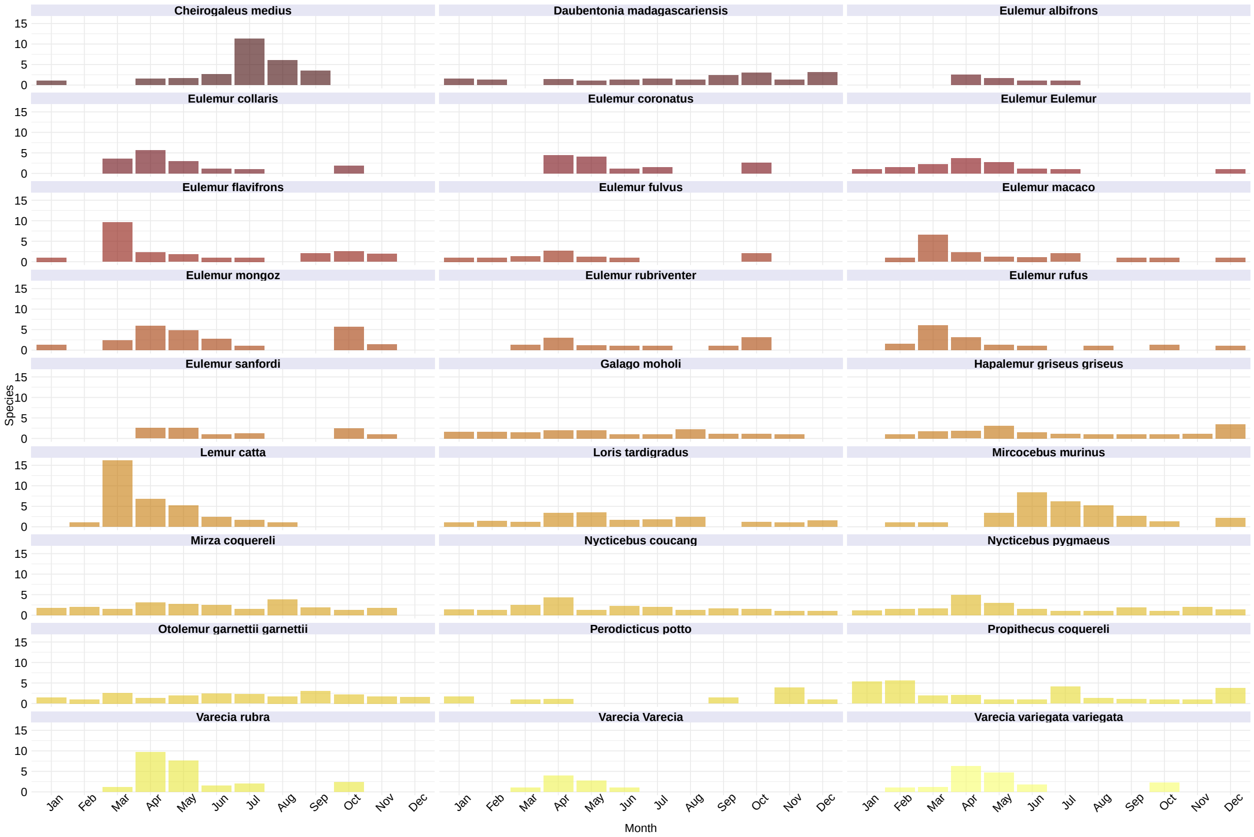
```
births_summdf %>%
  ggplot(aes(x = birth_month, y = n_births_mean, colour = species, group = species)) +
  geom_step() +
  theme_lemurs() +
  labs(x = "Month", y = "Number of births") +
  theme(legend.position = "bottom")
```

```
## Warning: Removed 58 row(s) containing missing values (geom_path).
```



```
births_summdf  %>%
  ggplot(aes(x = birth_month, y = n_births_mean, fill = species)) +
  geom_bar(alpha=0.6, stat = "identity") +
  theme_lemurs() +
  facet_wrap(~species, ncol = 3) +
  labs(x = "Month", y = "Species") +
  theme(legend.position = "none",
        axis.text.x = element_text(angle = 45))
```

```
## Warning: Removed 99 rows containing missing values (position_stack).
```



## Offspring production

```
offspring_df <- lemurs_smallts %>%
  dplyr::select(year, month, species, taxon, dlc_id, sex,
                litter_size, ## size of litter it was born into
                age_at_wt_y, weight,
                preg_status,
                n_offspring, ## total number of offspring produced until that day
                infant_lit_sz_if_preg)
```

Individual weight and litter size

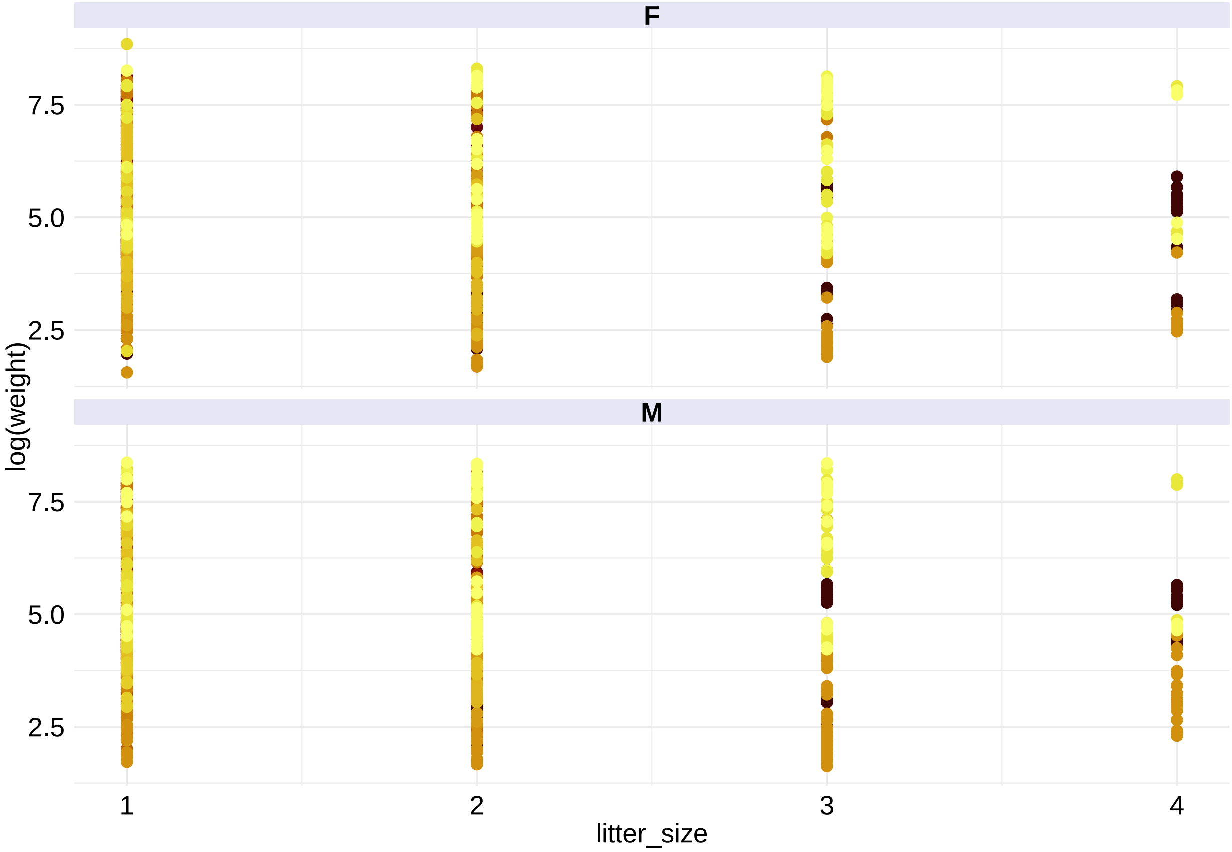
Get individual's weight at its younger age and plot it against against the litter it came from (separate males and females differently)

```
litterweight_df <- offspring_df %>%
  dplyr::group_by(dlc_id) %>%
  dplyr::filter(age_at_wt_y == min(age_at_wt_y)) %>%
  ungroup()

litterweight_df %>%
  dplyr::group_by(species) %>%
  dplyr::summarize(weight_mean = mean(weight),
                   weight_sd = sd(weight))
```

```
## # A tibble: 27 x 3
##    species                        weight_mean weight_sd
##    <chr>                                <dbl>     <dbl>
##   1 Cheirogaleus medius                  141.      112.
##   2 Daubentonia madagascariensis         410.      780.
##   3 Eulemur albifrons                   1757.      969.
##   4 Eulemur collaris                    1424.     1006.
##   5 Eulemur coronatus                    897.      694.
##   6 Eulemur Eulemur                     1015.      970.
##   7 Eulemur flavifrons                   552.      815.
##   8 Eulemur fulvus                      1617.     1008.
##   9 Eulemur macaco                      1239.     1091.
## 10 Eulemur mongoz                        757.      696.
## # ... with 17 more rows
```

```
litterweight_df  %>%
  dplyr::filter(sex != "ND") %>%
  ggplot(aes(x = litter_size, y = log(weight), colour = species, group = species))+
  geom_point() +
  facet_wrap(~sex, ncol = 1) +
  theme_lemurs() +
  theme(legend.position = "none")
```

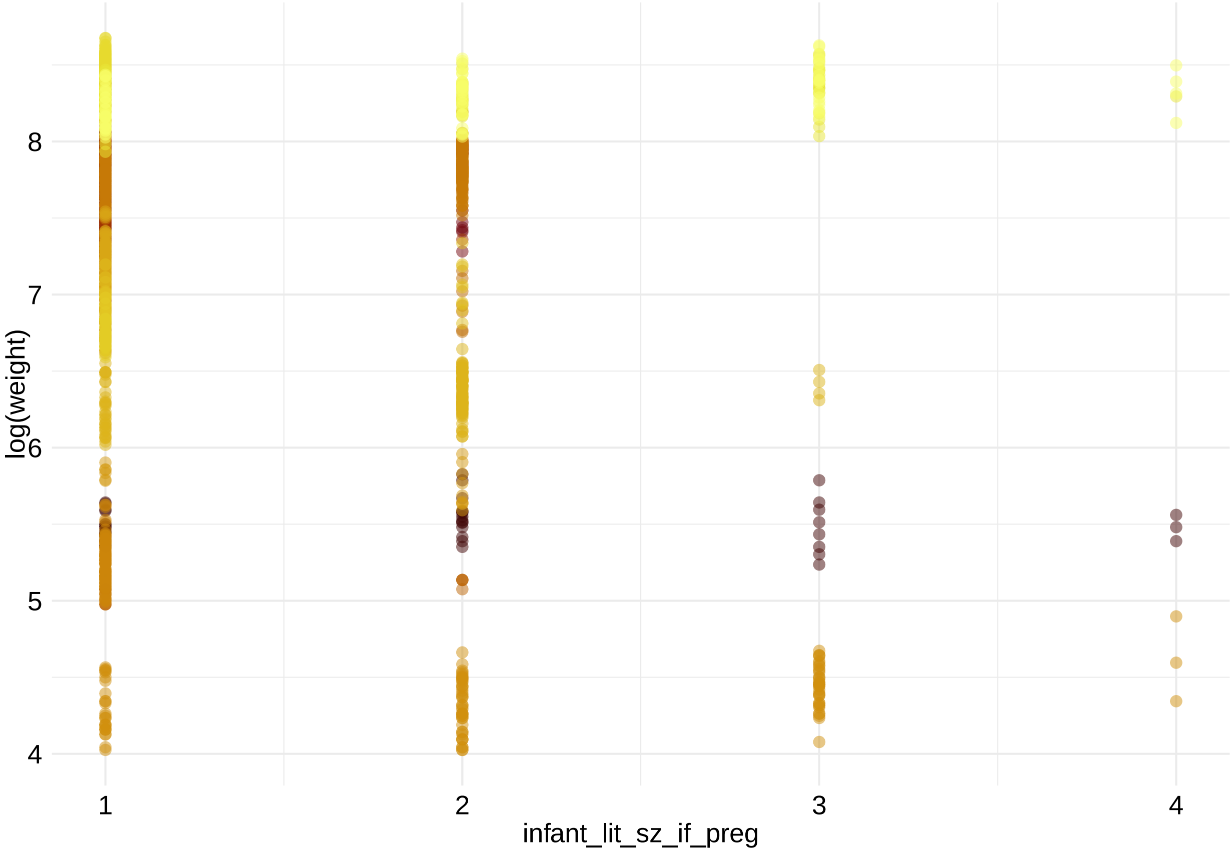
```
## Warning: Removed 374 rows containing missing values (geom_point).
```



Individual female weight and offspring production

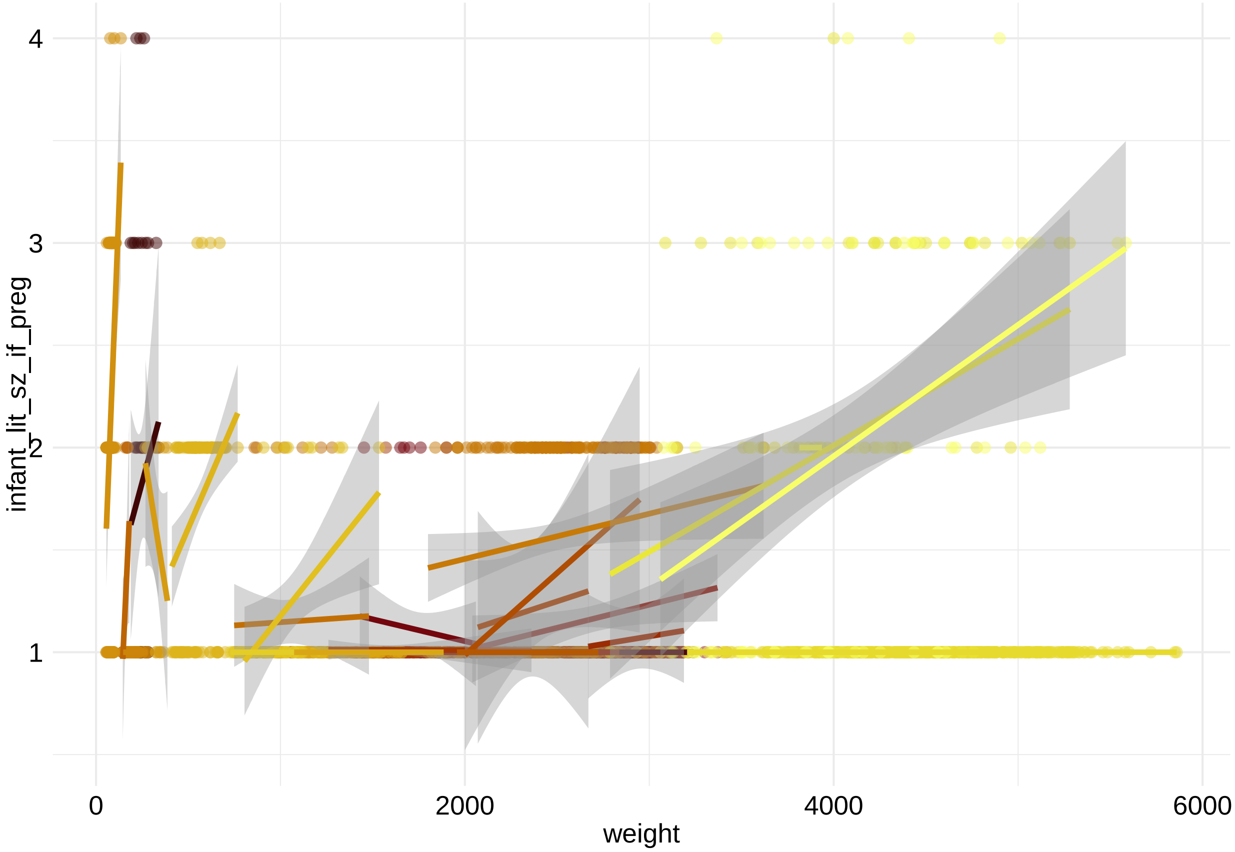
```
offspring_df  %>%
  dplyr::filter(preg_status == "P") %>%
  dplyr::group_by(dlc_id, species) %>%
  ggplot(aes(x = infant_lit_sz_if_preg, y = log(weight), colour = species))+
  geom_point(alpha = 0.5) +
  theme_lemurs() +
  theme(legend.position = "none")
```

```
## Warning: Removed 12 rows containing missing values (geom_point).
```



```
offspring_df  %>%
  dplyr::filter(preg_status == "P") %>%
  ggplot(aes(y = infant_lit_sz_if_preg, x = weight, colour = species))+
  geom_point(alpha = 0.5) +
  geom_smooth(method = lm) +
  theme_lemurs() +
  theme(legend.position = "none")
```

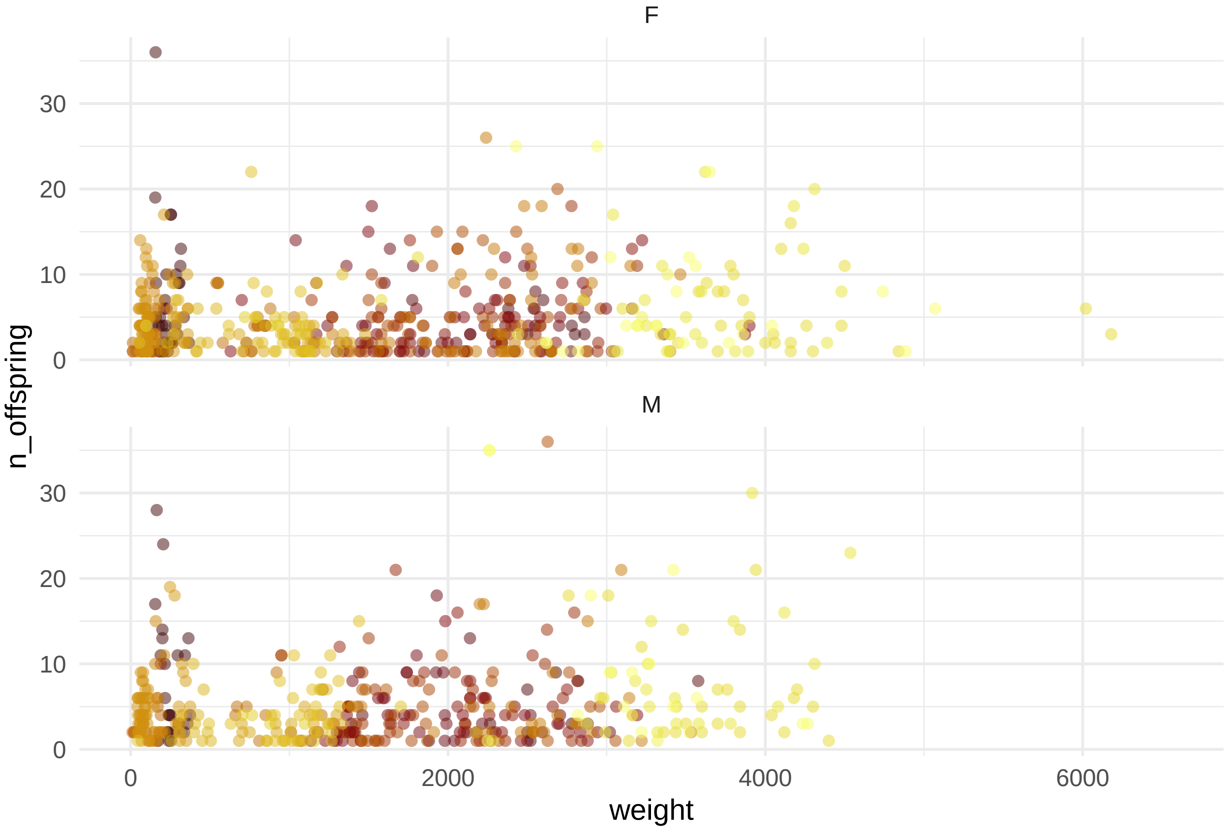
```
## Warning: Removed 12 rows containing non-finite values (stat_smooth).

## Warning in qt((1 - level)/2, df): NaNs produced

## Warning in qt((1 - level)/2, df): NaNs produced

## Warning in qt((1 - level)/2, df): NaNs produced

## Warning: Removed 12 rows containing missing values (geom_point).

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf
## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf
```



TODO: get weight of each individual at its oldest, and plot it against the n_offspring it produced. Facet for males and females

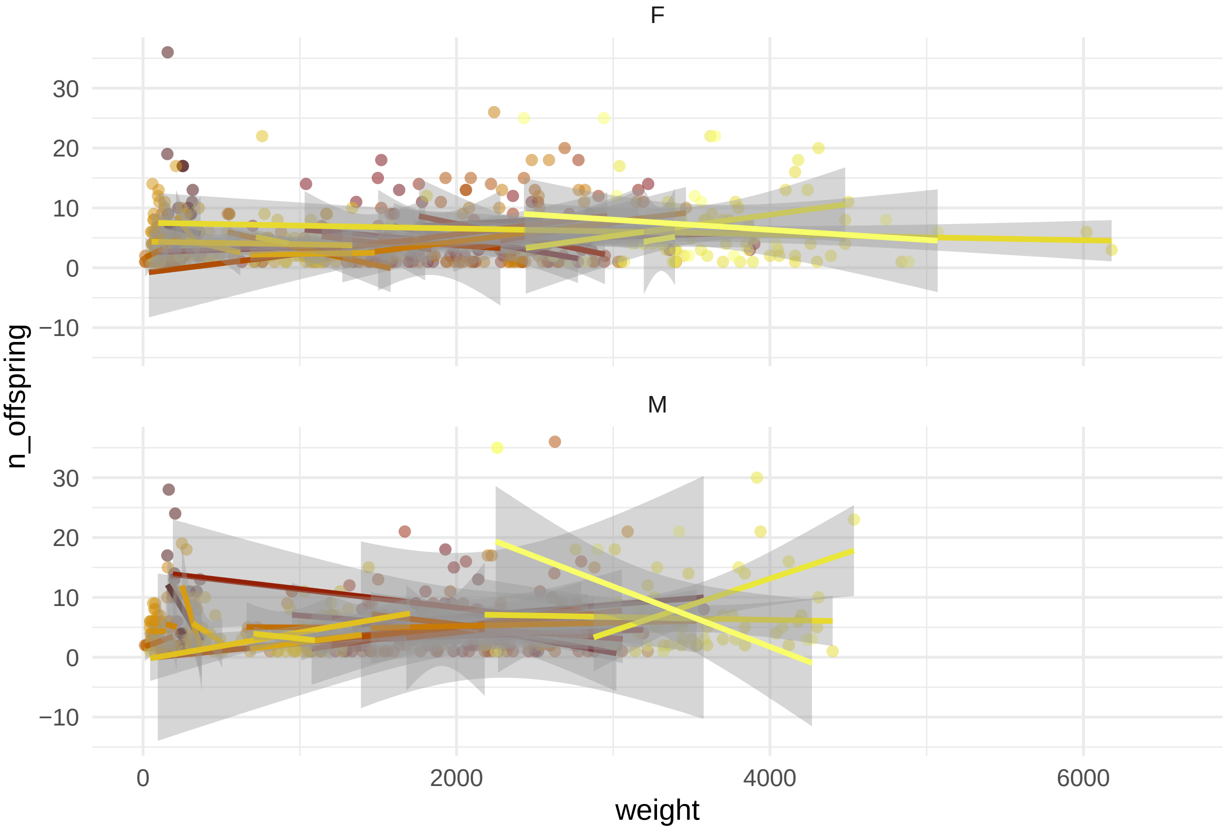
```
offspring_df  %>%
  filter(sex != "ND") %>%
  dplyr::group_by(dlc_id) %>%
  dplyr::filter(age_at_wt_y == max(age_at_wt_y)) %>%
  ggplot(aes(y = n_offspring, x = weight, colour = species))+
  geom_point(alpha = 0.5) +
  facet_wrap(~sex, ncol = 1) +
  theme_minimal() +
  theme(legend.position = "none")
```

```
## Warning: Removed 1375 rows containing missing values (geom_point).
```



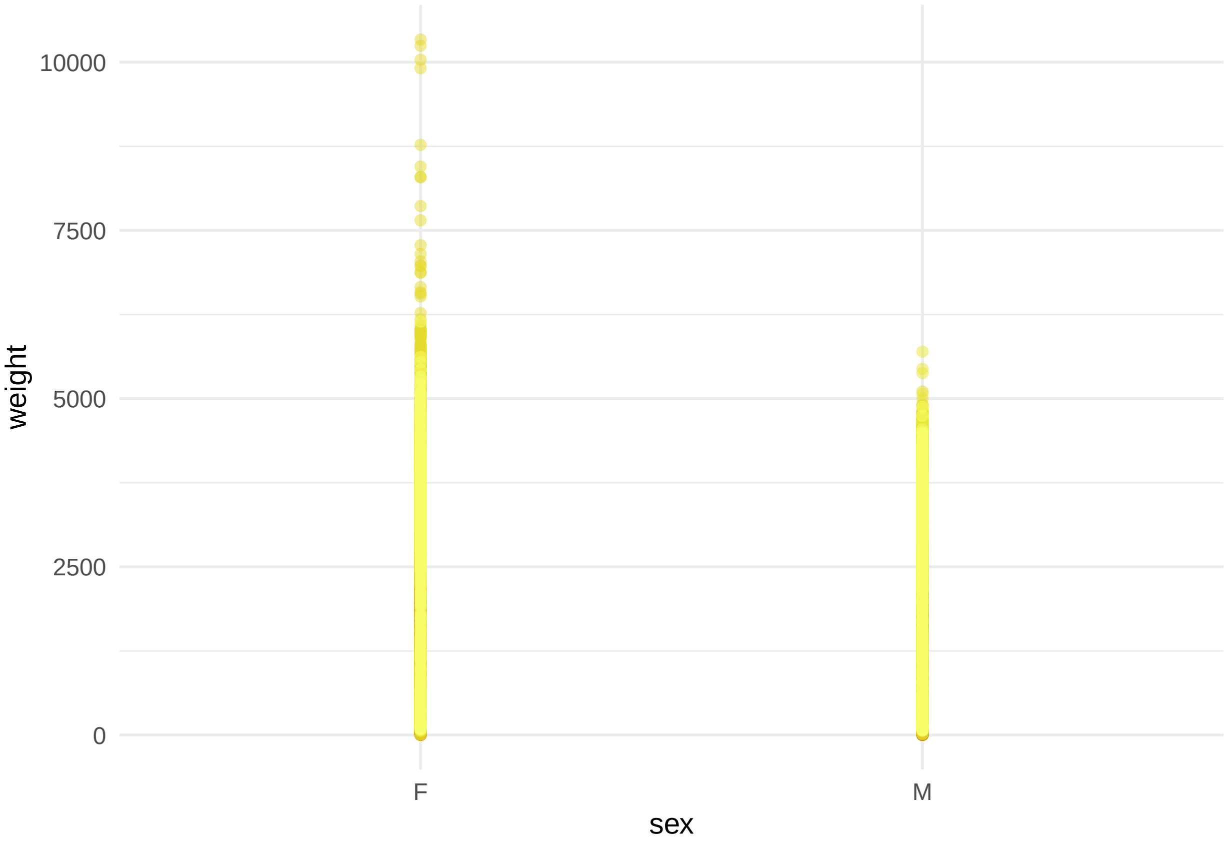
```
offspring_df  %>%
  filter(sex != "ND") %>%
  dplyr::group_by(dlc_id) %>%
  dplyr::filter(age_at_wt_y == max(age_at_wt_y)) %>%
  ggplot(aes(y = n_offspring, x = weight, colour = species))+
  geom_point(alpha = 0.5) +
  geom_smooth(method = lm) +
  facet_wrap(~sex, ncol = 1) +
  theme_minimal() +
  theme(legend.position = "none")
```

```
## Warning: Removed 1375 rows containing non-finite values (stat_smooth).

## Warning in qt((1 - level)/2, df): NaNs produced

## Warning: Removed 1375 rows containing missing values (geom_point).

## Warning in max(ids, na.rm = TRUE): no non-missing arguments to max; returning -
## Inf
```



Size difference between sexes

```
offspring_df  %>%
  filter(sex != "ND", preg_status == "NP") %>%
  ggplot(aes(x = sex, y = weight, colour = species))+
  geom_point(alpha = 0.5) +
  theme_minimal() +
  theme(legend.position = "none")
```



females

Supplementary material

TODO: daft punk graph of number of births per year, per species

R version, the OS and attached or loaded packages:

```
sessionInfo()
```

```
## R version 4.0.3 (2020-10-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 18.04.6 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/libopenblasp-r0.2.20.so
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=de_DE.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=de_DE.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=de_DE.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=de_DE.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
```

```
##
## other attached packages:
##  [1] kableExtra_1.3.1 ggridges_0.5.3   forcats_0.5.0    stringr_1.4.0
##  [5] dplyr_1.0.2      purrr_0.3.4      readr_1.4.0      tidyr_1.1.2
##  [9] tibble_3.1.1     ggplot2_3.3.3    tidyverse_1.3.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.6        lattice_0.20-41   lubridate_1.7.9   assertthat_0.2.1
##  [5] digest_0.6.27     utf8_1.2.1        R6_2.5.0          cellranger_1.1.0
##  [9] plyr_1.8.6        backports_1.1.10  reprex_0.3.0      evaluate_0.14
## [13] highr_0.9         httr_1.4.2        pillar_1.6.0      rlang_0.4.11
## [17] curl_4.3          readxl_1.3.1      rstudioapi_0.13   blob_1.2.1
## [21] Matrix_1.2-18     rmarkdown_2.11    splines_4.0.3     labeling_0.4.2
## [25] webshot_0.5.2     munsell_0.5.0     broom_0.7.2       compiler_4.0.3
## [29] modelr_0.1.8      xfun_0.22         pkgconfig_2.0.3   mgcv_1.8-33
## [33] htmltools_0.5.1.1 tidyselect_1.1.0  fansi_0.4.2       viridisLite_0.4.0
## [37] crayon_1.4.1      dbplyr_1.4.4      withr_2.4.2       grid_4.0.3
## [41] nlme_3.1-149      jsonlite_1.7.2    gtable_0.3.0      lifecycle_1.0.0
## [45] DBI_1.1.1         magrittr_2.0.1    scales_1.1.1      pals_1.7
## [49] cli_2.5.0         stringi_1.5.3     farver_2.1.0      mapproj_1.2.7
## [53] fs_1.5.0          xml2_1.3.2        ellipsis_0.3.2    generics_0.0.2
## [57] vctrs_0.3.8       tools_4.0.3       dichromat_2.0-0   glue_1.4.2
## [61] maps_3.4.0        hms_0.5.3         yaml_2.2.1        colorspace_2.0-2
## [65] rvest_0.3.6       knitr_1.33        haven_2.3.1
```