day 2 data visualization

Charles

2023-02-28

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

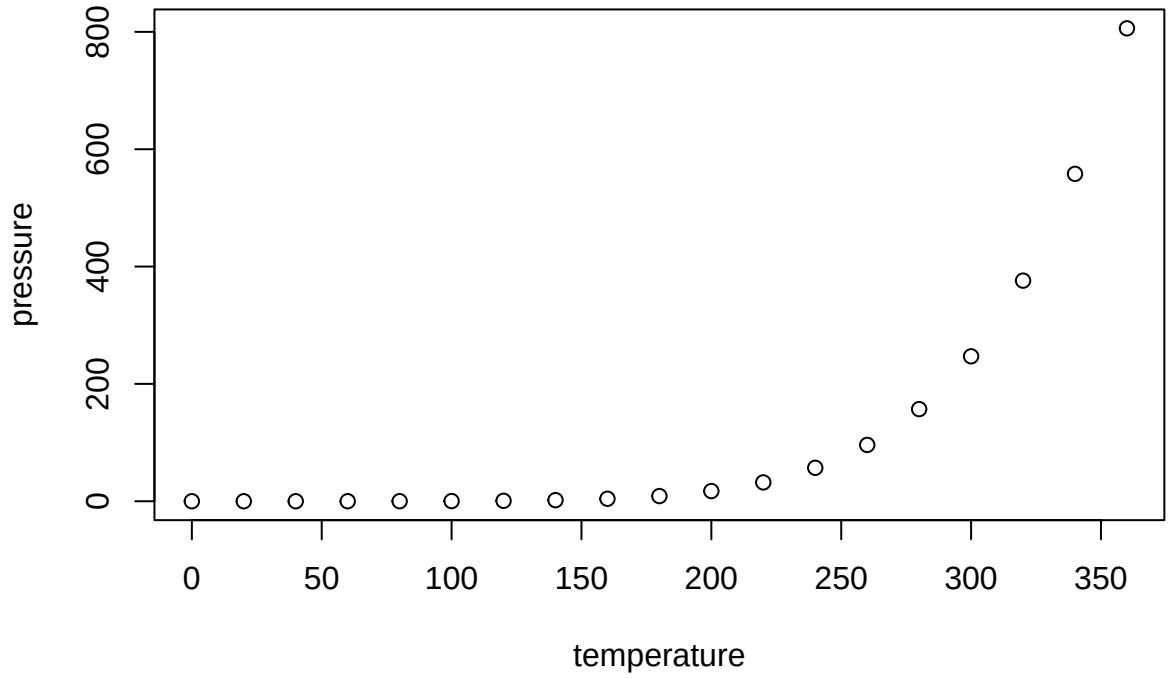
When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist       
##  Min.   : 4.0   Min.   :  2.00  
##  1st Qu.:12.0   1st Qu.: 26.00  
##  Median :15.0   Median : 36.00  
##  Mean   :15.4   Mean   : 42.98  
##  3rd Qu.:19.0   3rd Qu.: 56.00  
##  Max.   :25.0   Max.   :120.00
```

Including Plots

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
acacia <- read.csv(file = "../data-raw/ACACIA_DREPANOLOBIUM_SURVEY.txt",
                   sep = "\t",
                   na.strings = 'dead')
```

```
library(readr)
tree <- read_tsv("../data-raw/TREE_SURVEYS.txt",
                 col_types = list(HEIGHT = col_double(),
                                  AXIS_2 = col_double()))
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

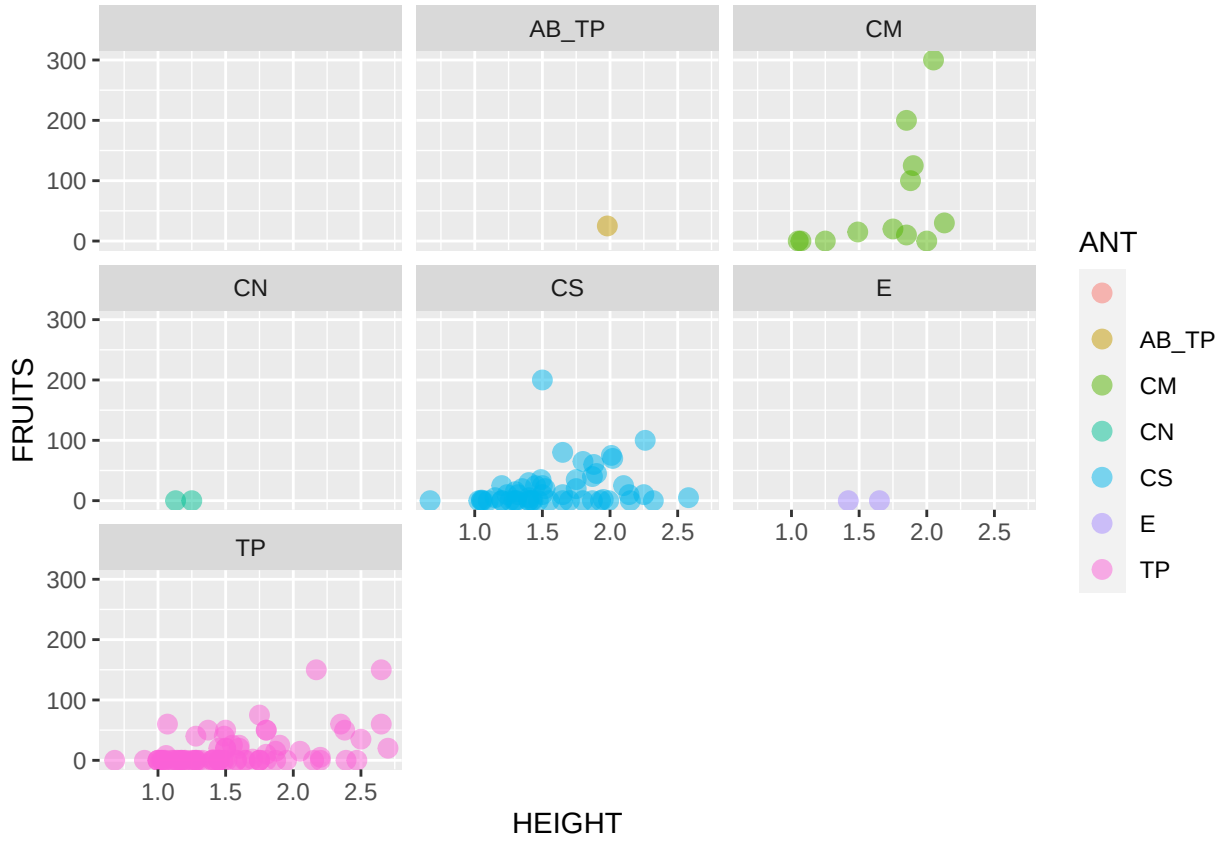
```
head(acacia)
```

```
##   SURVEY YEAR  SITE BLOCK TREATMENT    PLOT   ID HEIGHT AXIS1 AXIS2 CIRC
## 1      1 2012 SOUTH     1     TOTAL S1TOTAL  581   2.25  2.75  2.15   20
## 2      1 2012 SOUTH     1     TOTAL S1TOTAL  582   2.65  4.10  3.90   28
## 3      1 2012 SOUTH     1     TOTAL S1TOTAL 3111   1.50  1.70  0.85   17
## 4      1 2012 SOUTH     1     TOTAL S1TOTAL 3112   2.01  1.80  1.60   12
## 5      1 2012 SOUTH     1     TOTAL S1TOTAL 3113   1.75  1.84  1.42   13
```

```
## 6      1 2012 SOUTH     1     TOTAL S1TOTAL 3114   1.65  1.62  0.85   15
##   FLOWERS BUDS FRUITS ANT
## 1       0    0     10  CS
## 2       0    0    150  TP
## 3       2    1     50  TP
## 4       0    0     75  CS
## 5       0    0     20  CS
## 6       0    0      0   E
```

```
library(ggplot2)
ggplot(data = acacia, mapping = aes(x = HEIGHT, y = FRUITS, color = ANT)) +
  geom_point(size = 3, alpha = 0.5) +
  facet_wrap(~ANT)
```

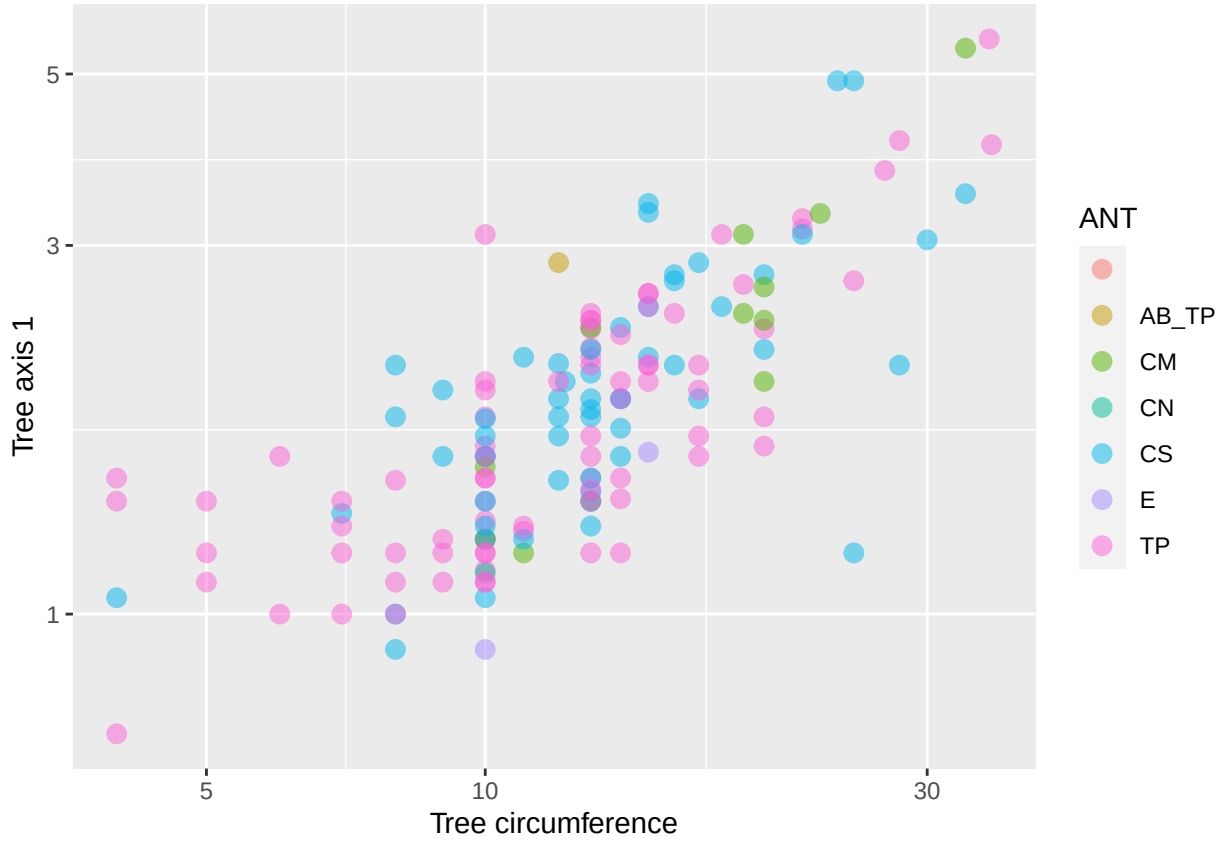
```
## Warning: Removed 4 rows containing missing values (‘geom_point()‘).
```



Exercise 1.

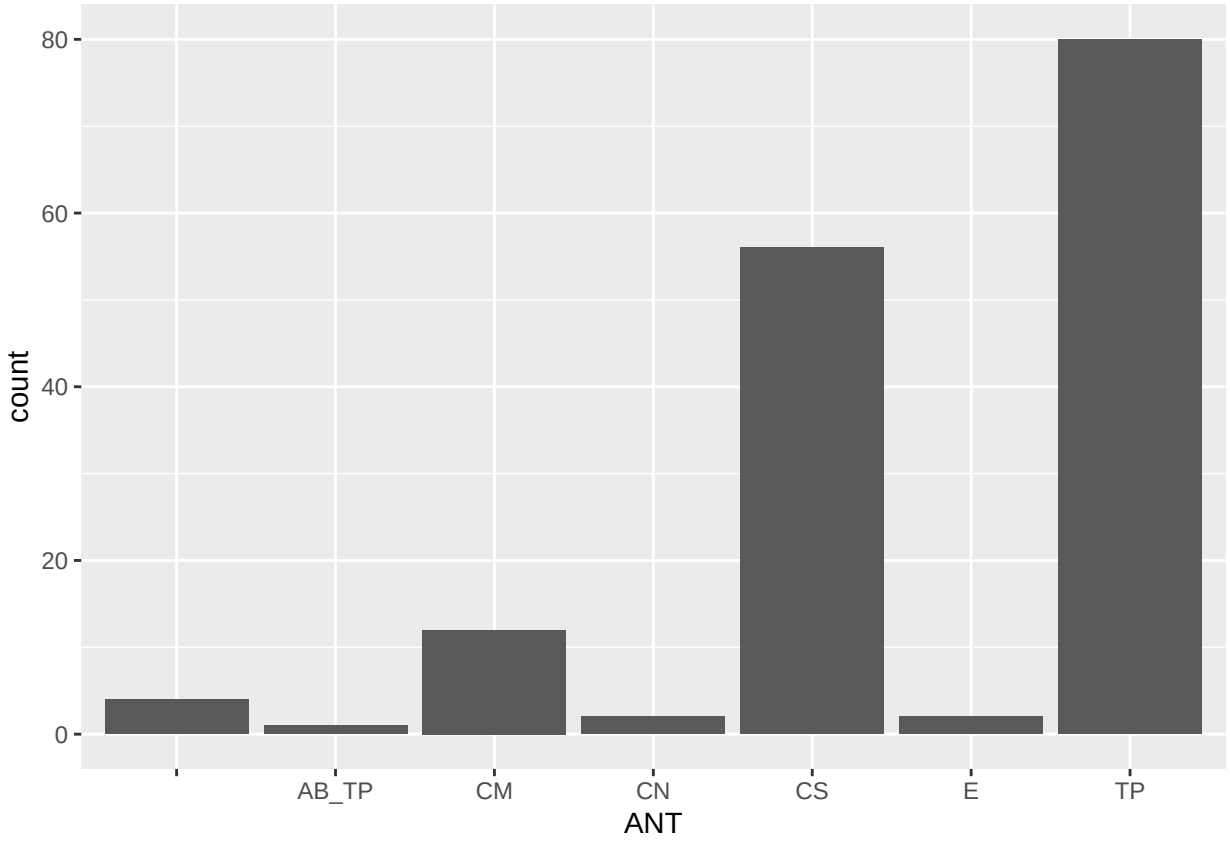
```
library(ggplot2)
ggplot(data = acacia, mapping = aes(x = CIRC, y = AXIS1, color = ANT)) +
  geom_point(size = 3, alpha = 0.5) +
  scale_y_log10() +
  scale_x_log10() +
  labs(x = "Tree circumference", y = "Tree axis 1")
```

```
## Warning: Removed 4 rows containing missing values (`geom_point()`).
```



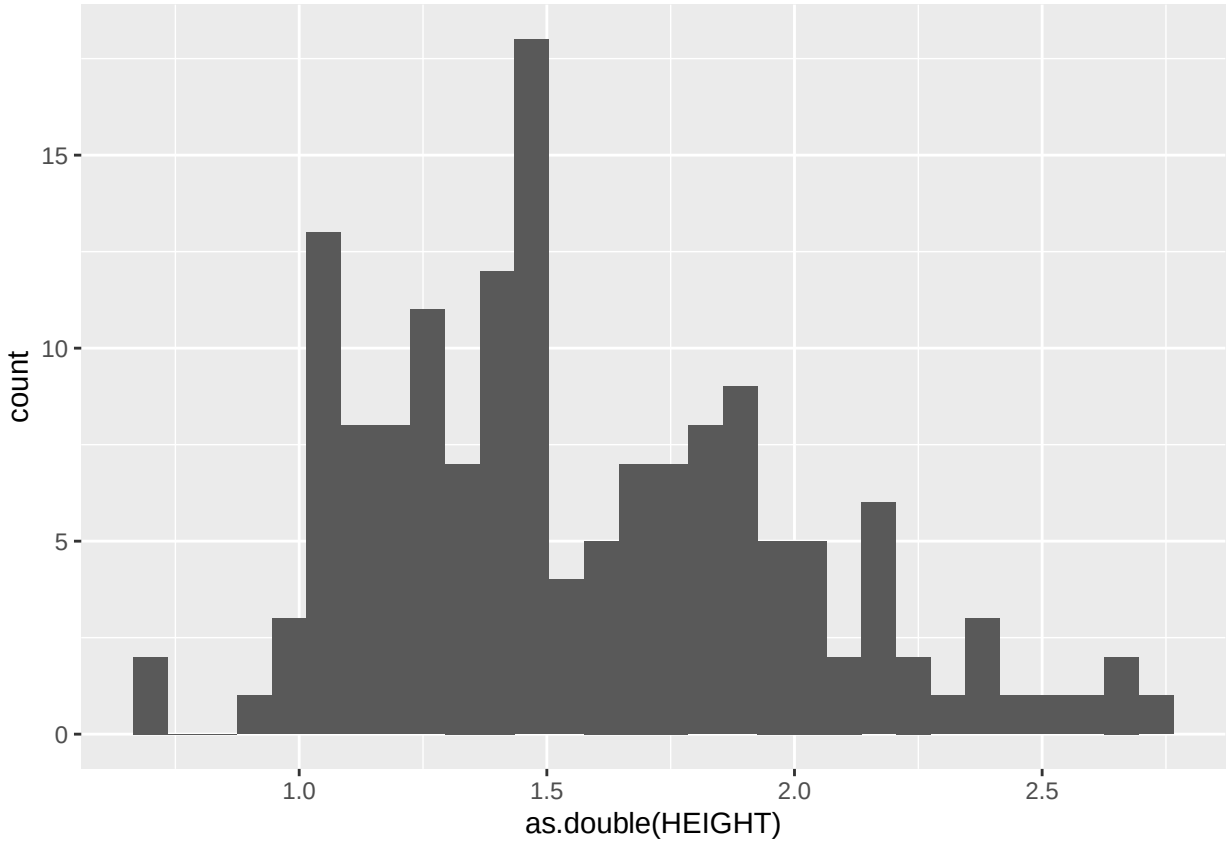
#Exercise 2

```
library(ggplot2)

ggplot(data = acacia, mapping = aes(x = ANT)) +
  geom_bar()
```



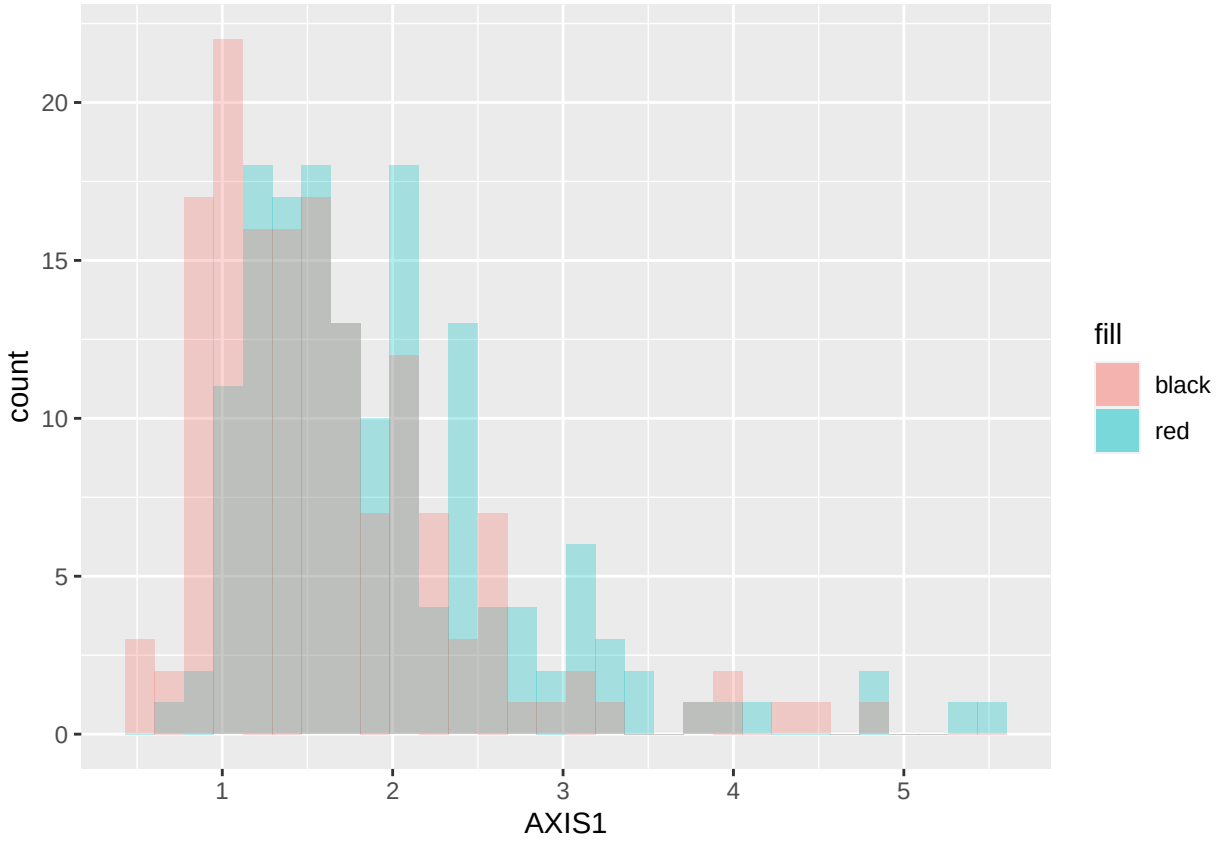

```
library(ggplot2)

ggplot(data = acacia, mapping = aes(x = as.double(HEIGHT))) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 4 rows containing non-finite values (`stat_bin()`).
```



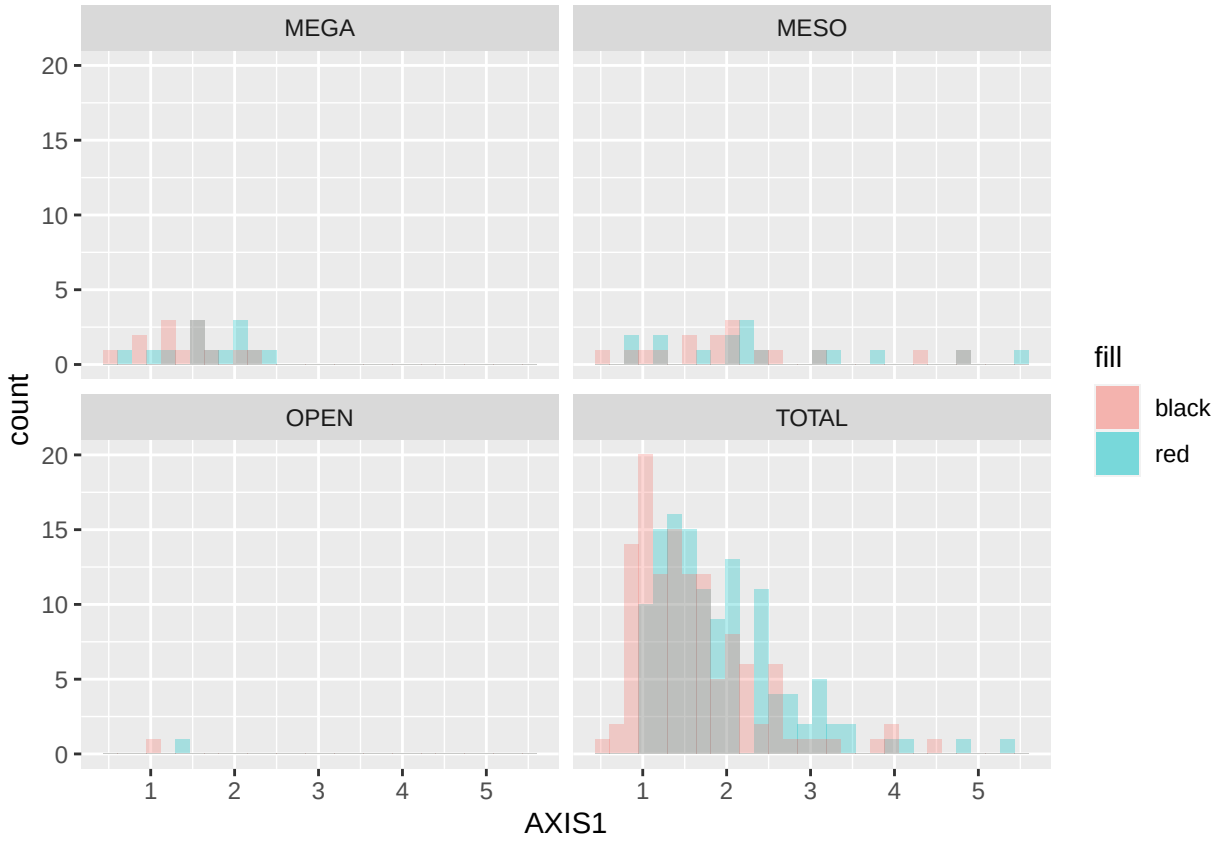

```
library(ggplot2)

ggplot() +
  geom_histogram(data = acacia,
                 mapping = aes(x = AXIS1, fill = "red"),
                 alpha = 0.3) +
  geom_histogram(data = acacia,
                 mapping = aes(x = AXIS2, fill = "black"),
                 alpha = 0.3)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 4 rows containing non-finite values (`stat_bin()`).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 4 rows containing non-finite values (`stat_bin()`).
```




```
library(ggplot2)

ggplot() +
  geom_histogram(data = acacia,
                 mapping = aes(x = AXIS1, fill = "red"),
                 alpha = 0.3) +
  geom_histogram(data = acacia,
                 mapping = aes(x = AXIS2, fill = "black"),
                 alpha = 0.3) +
  facet_wrap(~TREATMENT)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 4 rows containing non-finite values (`stat_bin()`).

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## Warning: Removed 4 rows containing non-finite values (`stat_bin()`).
```



Visual QA and control

```
str(acacia)
```

```
## 'data.frame':    157 obs. of  15 variables:
##  $ SURVEY   : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ YEAR     : int  2012 2012 2012 2012 2012 2012 2012 2012 2012 2012 ...
##  $ SITE     : chr  "SOUTH" "SOUTH" "SOUTH" "SOUTH" ...
##  $ BLOCK    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ TREATMENT: chr  "TOTAL" "TOTAL" "TOTAL" "TOTAL" ...
##  $ PLOT     : chr  "S1TOTAL" "S1TOTAL" "S1TOTAL" "S1TOTAL" ...
##  $ ID       : int  581 582 3111 3112 3113 3114 3115 3199 941 942 ...
##  $ HEIGHT   : num  2.25 2.65 1.5 2.01 1.75 1.65 1.2 1.45 1.87 2.38 ...
##  $ AXIS1    : num  2.75 4.1 1.7 1.8 1.84 1.62 1.95 2 2.15 5.55 ...
##  $ AXIS2    : num  2.15 3.9 0.85 1.6 1.42 0.85 0.9 1.75 1.82 4.82 ...
##  $ CIRC     : num  20 28 17 12 13 15 9 12.2 13 35 ...
##  $ FLOWERS  : int  0 0 2 0 0 0 0 0 0 0 ...
##  $ BUDS     : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ FRUITS   : int  10 150 50 75 20 0 0 25 0 50 ...
##  $ ANT      : chr  "CS" "TP" "TP" "CS" ...
```

```
is.numeric(acacia$CIRC)
```

```
## [1] TRUE
```

```
is.numeric(acacia$HEIGHT)
```

```
## [1] TRUE
```

```
str(tree)
```

```
## spc_tbl_ [7,508 x 16] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ SURVEY      : num [1:7508] 1 2 3 4 5 1 2 3 4 5 ...
##  $ YEAR        : num [1:7508] 2009 2010 2011 2012 2013 ...
##  $ SITE        : chr [1:7508] "SOUTH" "SOUTH" "SOUTH" "SOUTH" ...
##  $ TREATMENT   : chr [1:7508] "TOTAL" "TOTAL" "TOTAL" "TOTAL" ...
##  $ BLOCK       : num [1:7508] 2 2 2 2 2 2 2 2 2 2 ...
##  $ PLOT        : chr [1:7508] "S2TOTAL" "S2TOTAL" "S2TOTAL" "S2TOTAL" ...
##  $ SPECIES     : chr [1:7508] "Acacia_etbaica" "Acacia_etbaica" "Acacia_etbaica" "Acacia_etbaica" ..
##  $ ORIGINAL_TAG: num [1:7508] 1 1 1 1 1 2 2 2 2 2 ...
##  $ NEW_TAG     : num [1:7508] NA NA NA NA NA NA NA NA NA NA ...
##  $ DEAD        : chr [1:7508] "N" "N" "N" "N" ...
##  $ HEIGHT      : num [1:7508] 3.4 3.32 3.65 3.74 3.59 2.3 2.32 2.75 NA 2.86 ...
##  $ AXIS_1      : num [1:7508] 6.1 8.25 8.85 5.5 5 2.2 2.75 3.3 NA 3.7 ...
##  $ AXIS_2      : num [1:7508] 5 8.45 9 7.1 8.15 2.8 2.65 3.8 NA 2.6 ...
##  $ CIRC        : num [1:7508] 37.8 18.8 57 60 55 14.2 18.4 25 NA 31 ...
##  $ MEASUREMENT : chr [1:7508] "D" "D" "C" "C" ...
##  $ STEMS       : chr [1:7508] "1" "1" "1" "1" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   SURVEY = col_double(),
##   ..   YEAR = col_double(),
##   ..   SITE = col_character(),
##   ..   TREATMENT = col_character(),
##   ..   BLOCK = col_double(),
##   ..   PLOT = col_character(),
##   ..   SPECIES = col_character(),
##   ..   ORIGINAL_TAG = col_double(),
##   ..   NEW_TAG = col_double(),
##   ..   DEAD = col_character(),
##   ..   HEIGHT = col_double(),
##   ..   AXIS_1 = col_double(),
##   ..   AXIS_2 = col_double(),
##   ..   CIRC = col_double(),
##   ..   MEASUREMENT = col_character(),
##   ..   STEMS = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
is.numeric(tree$CIRC)
```

```
## [1] TRUE
```

```
is.numeric(tree$HEIGHT)
```

```
## [1] TRUE
```

```
ggplot() +
  geom_point(data = tree, mapping = aes(x = CIRC, y = HEIGHT, color = "gray",
  alpha = 0.5)) +
  geom_point(data = acacia, mapping = aes(x = CIRC, y = HEIGHT, color = "red")) +
  geom_smooth(data = tree, mapping = aes(x = CIRC, y = HEIGHT)) +
  geom_smooth(data = acacia, mapping = aes(x = CIRC, y = HEIGHT))
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```

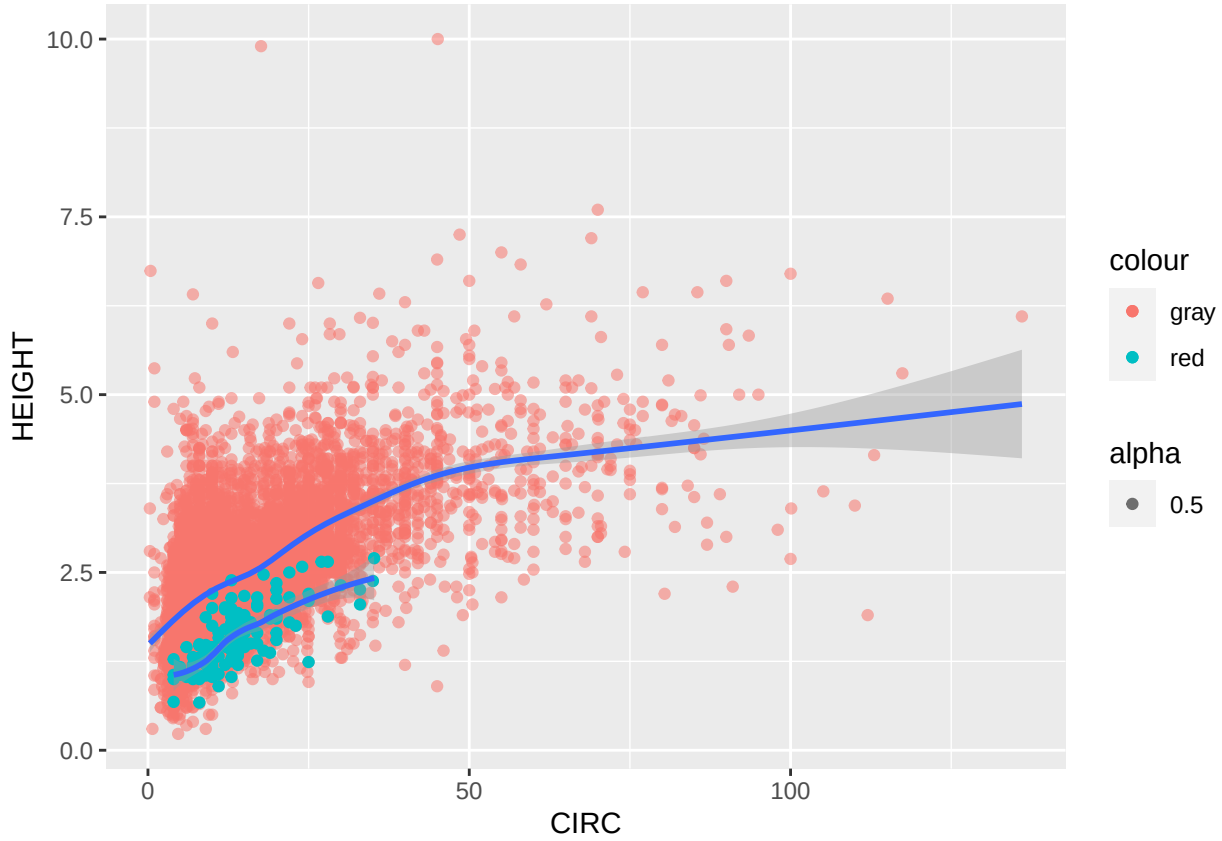
```
## Warning: Removed 414 rows containing non-finite values (`stat_smooth()`).
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```

```
## Warning: Removed 4 rows containing non-finite values (`stat_smooth()`).
```

```
## Warning: Removed 414 rows containing missing values (`geom_point()`).
```

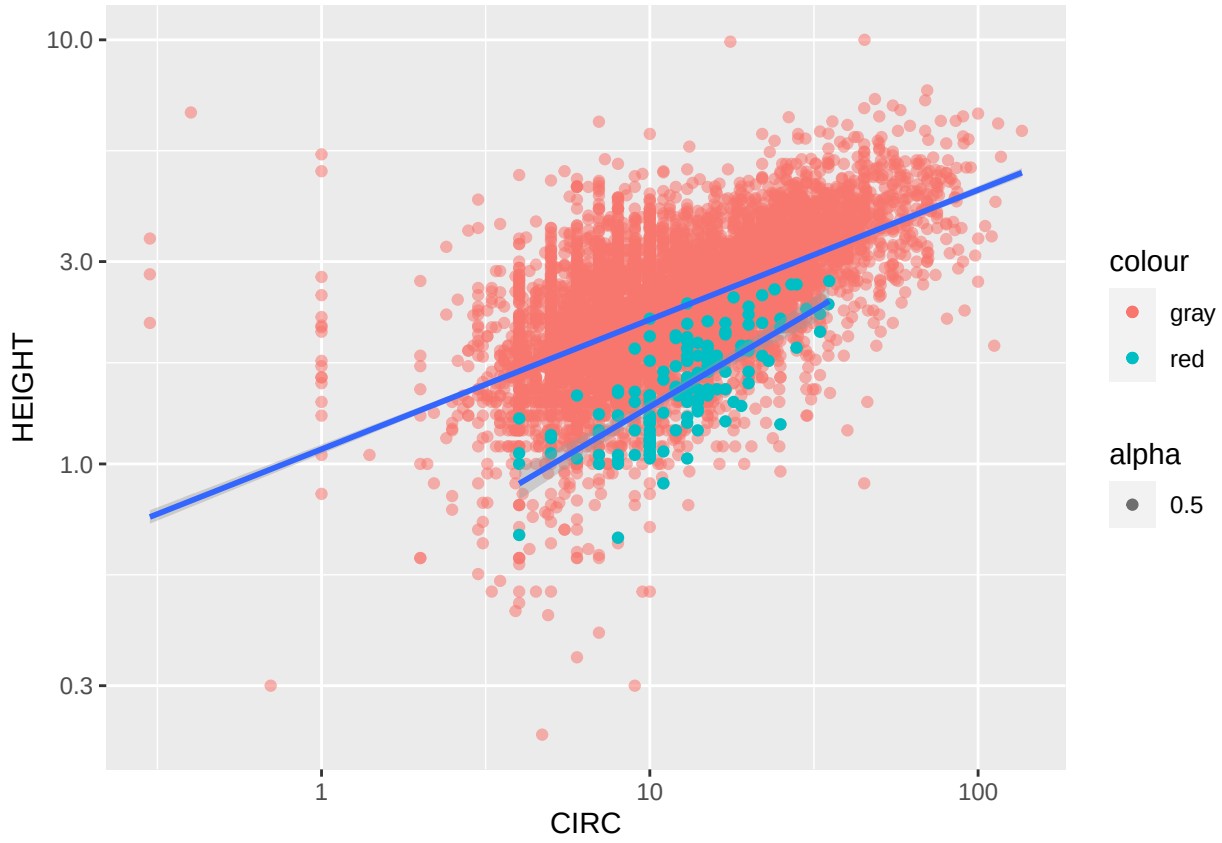
```
## Warning: Removed 4 rows containing missing values (`geom_point()`).
```



#LINEAR MODELS

```
ggplot() +
  geom_point(data = tree, mapping = aes(x = CIRC, y = HEIGHT, color = "gray",
  alpha = 0.5)) +
  geom_point(data = acacia, mapping = aes(x = CIRC, y = HEIGHT, color = "red")) +
  scale_x_log10() +
  scale_y_log10() +
  geom_smooth(data = tree, mapping = aes(x = CIRC, y = HEIGHT), method = "lm") +
  geom_smooth(data = acacia, mapping = aes(x = CIRC, y = HEIGHT), method = "lm")
```

```
## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 414 rows containing non-finite values (`stat_smooth()`).

## `geom_smooth()` using formula = 'y ~ x'

## Warning: Removed 4 rows containing non-finite values (`stat_smooth()`).

## Warning: Removed 414 rows containing missing values (`geom_point()`).

## Warning: Removed 4 rows containing missing values (`geom_point()`).
```



```
read.csv(file = "../data-raw/surveys.csv") %>%
  filter(species_id == "DS", !is.na(weight)) %>%
  arrange(year) %>%
  select(year, weight) ->
  ds_weight_by_year
str(ds_weight_by_year)
```

```
## 'data.frame':    2344 obs. of  2 variables:
##  $ year  : int  1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 ...
##  $ weight: int  117 121 115 120 118 126 132 113 122 107 ...
```

piping to an argument that is not the first one

Some functions do not take data as the first argument

```
surveys <- read.csv(file = "../data-raw/surveys.csv")
```

```
str(surveys)
```

```
## 'data.frame':    35549 obs. of  9 variables:
##  $ record_id      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ month          : int  7 7 7 7 7 7 7 7 7 7 ...
##  $ day            : int  16 16 16 16 16 16 16 16 16 16 ...
##  $ year           : int  1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 ...
##  $ plot_id        : int  2 3 2 7 3 1 2 1 1 6 ...
##  $ species_id     : chr  "NL" "NL" "DM" "DM" ...
##  $ sex            : chr  "M" "M" "F" "M" ...
##  $ hindfoot_length: int  32 33 37 36 35 14 NA 37 34 20 ...
##  $ weight         : int  NA NA NA NA NA NA NA NA NA NA ...
```

```
lm(weight ~ year, data = surveys)
```

```
##
## Call:
## lm(formula = weight ~ year, data = surveys)
##
## Coefficients:
## (Intercept)         year
##    2752.137       -1.361
```

```
surveys %>%
  lm(formula = weight ~ year, data = .)
```

```
##
## Call:
## lm(formula = weight ~ year, data = .)
##
## Coefficients:
## (Intercept)         year
##    2752.137       -1.361
```

In Class Exercise

```
surveys %>% filter(species_id == "DS", !is.na(weight)) %>%
  lm(weight ~ year, data = .) %>%
  summary()
```

```
##
## Call:
## lm(formula = weight ~ year, data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -109.787  -12.440    3.723   14.886   69.886
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -709.1968   263.2510  -2.694  0.00711 **
## year           0.4184     0.1328   3.150  0.00165 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.86 on 2342 degrees of freedom
## Multiple R-squared:  0.00422,    Adjusted R-squared:  0.003795
## F-statistic: 9.925 on 1 and 2342 DF,  p-value: 0.001651
```

AGGREGAT

```
surveys %>% group_by(year) -> grouped_surveys
str(grouped_surveys)
```

```
## gropd_df [35,549 x 9] (S3: grouped_df/tbl_df/tbl/data.frame)
##  $ record_id      : int [1:35549] 1 2 3 4 5 6 7 8 9 10 ...
##  $ month          : int [1:35549] 7 7 7 7 7 7 7 7 7 7 ...
##  $ day            : int [1:35549] 16 16 16 16 16 16 16 16 16 16 ...
##  $ year           : int [1:35549] 1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 ...
##  $ plot_id        : int [1:35549] 2 3 2 7 3 1 2 1 1 6 ...
##  $ species_id     : chr [1:35549] "NL" "NL" "DM" "DM" ...
##  $ sex            : chr [1:35549] "M" "M" "F" "M" ...
##  $ hindfoot_length: int [1:35549] 32 33 37 36 35 14 NA 37 34 20 ...
##  $ weight         : int [1:35549] NA NA NA NA NA NA NA NA NA NA ...
##  - attr(*, "groups")= tibble [26 x 2] (S3: tbl_df/tbl/data.frame)
##   ..$ year : int [1:26] 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 ...
##   ..$ .rows: list<int> [1:26]
##   .. ..$ : int [1:503] 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..$ : int [1:1048] 504 505 506 507 508 509 510 511 512 513 ...
##   .. ..$ : int [1:719] 1552 1553 1554 1555 1556 1557 1558 1559 1560 1561 ...
##   .. ..$ : int [1:1415] 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 ...
##   .. ..$ : int [1:1472] 3686 3687 3688 3689 3690 3691 3692 3693 3694 3695 ...
##   .. ..$ : int [1:1978] 5158 5159 5160 5161 5162 5163 5164 5165 5166 5167 ...
```

```
##   .. ..$ : int [1:1673] 7136 7137 7138 7139 7140 7141 7142 7143 7144 7145 ...
##   .. ..$ : int [1:981] 8809 8810 8811 8812 8813 8814 8815 8816 8817 8818 ...
##   .. ..$ : int [1:1438] 9790 9791 9792 9793 9794 9795 9796 9797 9798 9799 ...
##   .. ..$ : int [1:942] 11228 11229 11230 11231 11232 11233 11234 11235 11236 11237 ...
##   .. ..$ : int [1:1671] 12170 12171 12172 12173 12174 12175 12176 12177 12178 12179 ...
##   .. ..$ : int [1:1469] 13841 13842 13843 13844 13845 13846 13847 13848 13849 13850 ...
##   .. ..$ : int [1:1569] 15310 15311 15312 15313 15314 15315 15316 15317 15318 15319 ...
##   .. ..$ : int [1:1311] 16879 16880 16881 16882 16883 16884 16885 16886 16887 16888 ...
##   .. ..$ : int [1:1347] 18190 18191 18192 18193 18194 18195 18196 18197 18198 18199 ...
##   .. ..$ : int [1:1038] 19537 19538 19539 19540 19541 19542 19543 19544 19545 19546 ...
##   .. ..$ : int [1:750] 20575 20576 20577 20578 20579 20580 20581 20582 20583 20584 ...
##   .. ..$ : int [1:668] 21325 21326 21327 21328 21329 21330 21331 21332 21333 21334 ...
##   .. ..$ : int [1:1222] 21993 21994 21995 21996 21997 21998 21999 22000 22001 22002 ...
##   .. ..$ : int [1:1706] 23215 23216 23217 23218 23219 23220 23221 23222 23223 23224 ...
##   .. ..$ : int [1:2493] 24921 24922 24923 24924 24925 24926 24927 24928 24929 24930 ...
##   .. ..$ : int [1:1610] 27414 27415 27416 27417 27418 27419 27420 27421 27422 27423 ...
##   .. ..$ : int [1:1135] 29024 29025 29026 29027 29028 29029 29030 29031 29032 29033 ...
##   .. ..$ : int [1:1552] 30159 30160 30161 30162 30163 30164 30165 30166 30167 30168 ...
##   .. ..$ : int [1:1610] 31711 31712 31713 31714 31715 31716 31717 31718 31719 31720 ...
##   .. ..$ : int [1:2229] 33321 33322 33323 33324 33325 33326 33327 33328 33329 33330 ...
##   .. ..@ ptype: int(0)
##   ..- attr(*, ".drop")= logi TRUE
```

```
group_by(surveys, year, sex)
```

```
## # A tibble: 35,549 x 9
## # Groups:   year, sex [78]
##    record_id month   day  year plot_id species_id sex   hindfoot_length weight
##        <int> <int> <int> <int>   <int> <chr>      <chr>           <int>  <int>
##  1         1     7    16  1977       2 NL         M                  32     NA
##  2         2     7    16  1977       3 NL         M                  33     NA
##  3         3     7    16  1977       2 DM         F                  37     NA
##  4         4     7    16  1977       7 DM         M                  36     NA
##  5         5     7    16  1977       3 DM         M                  35     NA
##  6         6     7    16  1977       1 PF         M                  14     NA
##  7         7     7    16  1977       2 PE         F                  NA     NA
##  8         8     7    16  1977       1 DM         M                  37     NA
##  9         9     7    16  1977       1 DM         F                  34     NA
## 10        10     7    16  1977       6 PF         F                  20     NA
## # ... with 35,539 more rows
```

Set summary statistics of groups

```
group_by(surveys, year, sex) %>%
  summarize(count = n())
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 78 x 3
```

```
## # Groups:   year [26]
##     year sex   count
##    <int> <chr> <int>
##  1  1977 ""       85
##  2  1977 "F"     204
##  3  1977 "M"     214
##  4  1978 ""      112
##  5  1978 "F"     503
##  6  1978 "M"     433
##  7  1979 ""       68
##  8  1979 "F"     327
##  9  1979 "M"     324
## 10  1980 ""       83
## # ... with 68 more rows
```

```
group_by(surveys, year, sex) %>%
  summarize(mean_weight = mean(weight, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'year'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 78 x 3
## # Groups:   year [26]
##     year sex   mean_weight
##    <int> <chr>       <dbl>
##  1  1977 ""           28
##  2  1977 "F"          47.6
##  3  1977 "M"          46.1
##  4  1978 ""           82.4
##  5  1978 "F"          70.0
##  6  1978 "M"          65.3
##  7  1979 ""          110.
##  8  1979 "F"          65.6
##  9  1979 "M"          60.9
## 10  1980 ""          129.
## # ... with 68 more rows
```

```
group_by(surveys, species_id) %>%
  summarize(count = n())
```

```
## # A tibble: 49 x 2
##    species_id count
##    <chr>      <int>
##  1 ""           763
##  2 "AB"         303
##  3 "AH"         437
##  4 "AS"           2
##  5 "BA"          46
##  6 "CB"          50
##  7 "CM"          13
##  8 "CQ"          16
##  9 "CS"           1
## 10 "CT"           1
## # ... with 39 more rows
```

```
group_by(surveys, species_id, year) %>%
  summarize(count = n())
```

```
## `summarise()` has grouped output by 'species_id'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 535 x 3
## # Groups:   species_id [49]
##    species_id  year count
##    <chr>      <int> <int>
##  1 ""          1977    16
##  2 ""          1978    56
##  3 ""          1979    61
##  4 ""          1980    40
##  5 ""          1981    55
##  6 ""          1982    14
##  7 ""          1983    21
##  8 ""          1984    30
##  9 ""          1985    22
## 10 ""          1986    20
## # ... with 525 more rows
```

```
filter(surveys, species_id == "DO") %>%
  group_by(year) %>%
  summarize(mean_weight = mean(weight, na.rm = TRUE))
```

```
## # A tibble: 26 x 2
##     year mean_weight
##    <int>       <dbl>
##  1  1977        42.7
##  2  1978        45
##  3  1979        45.9
##  4  1980        48.1
##  5  1981        49.1
##  6  1982        47.9
##  7  1983        47.2
##  8  1984        48.4
##  9  1985        48.0
## 10  1986        49.4
## # ... with 16 more rows
```