Lecture 01 Data Basics

Damon G. Lamb

1/10/2024

Data!

Our basic data storage is an enhanced data frame called a tibble, although for most of our purposes, we can just think of it as a data frame.

These examples show methods you should rarely, if ever, expect to use during this class (though with some exceptions). We will use this to learn basic syntax and to better understand the basic organizational structure of our data storage in this framework, the tibble (a special tidyverse version of a data.frame).

```r
# make a list or vector using the base "combine" function, 'c'.
my_list <- c("first entry", "second", "3rd", 4)

# an extremely useful function: str
str(my_list)
```

```
##  chr [1:4] "first entry" "second" "3rd" "4"
```

```r
alternate_list <-c(1,2,3,4)
str(alternate_list)
```

```
##  num [1:4] 1 2 3 4
```

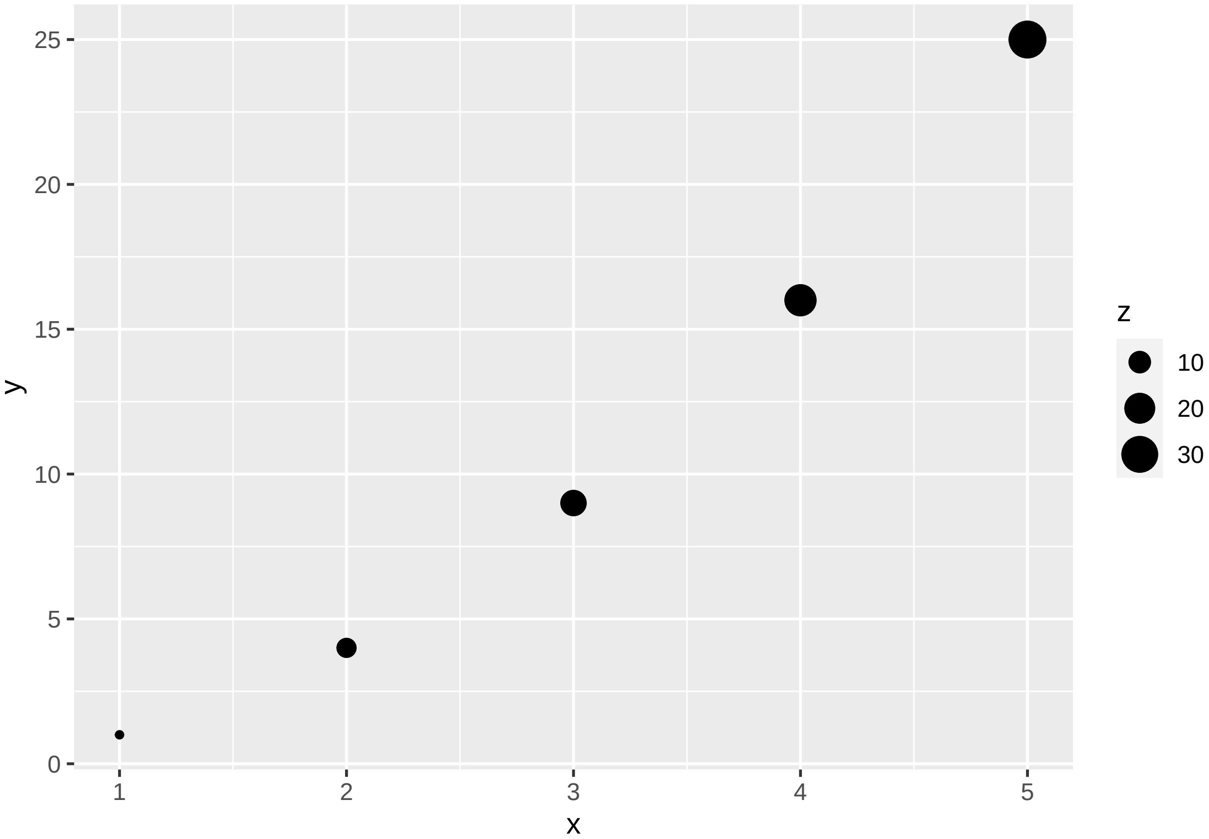
```r
sequence_list <-seq(0,10, by=2)
str(sequence_list)
```

```
##  num [1:6] 0 2 4 6 8 10
```

```r
the_best_tibble_ever <- tibble(x = 1:5,
                               y = x ^ 2,  # note the internal back-reference
                               z = 2 + x + y )
str(the_best_tibble_ever)
```

```
## tibble [5 x 3] (S3: tbl_df/tbl/data.frame)
##  $ x: int [1:5] 1 2 3 4 5
##  $ y: num [1:5] 1 4 9 16 25
##  $ z: num [1:5] 4 8 14 22 32
```

```r
# Let's make a quick, kinda bad figure:
ggplot(data = the_best_tibble_ever, mapping = aes(x=x, y=y, size=z))+
  geom_point()
```



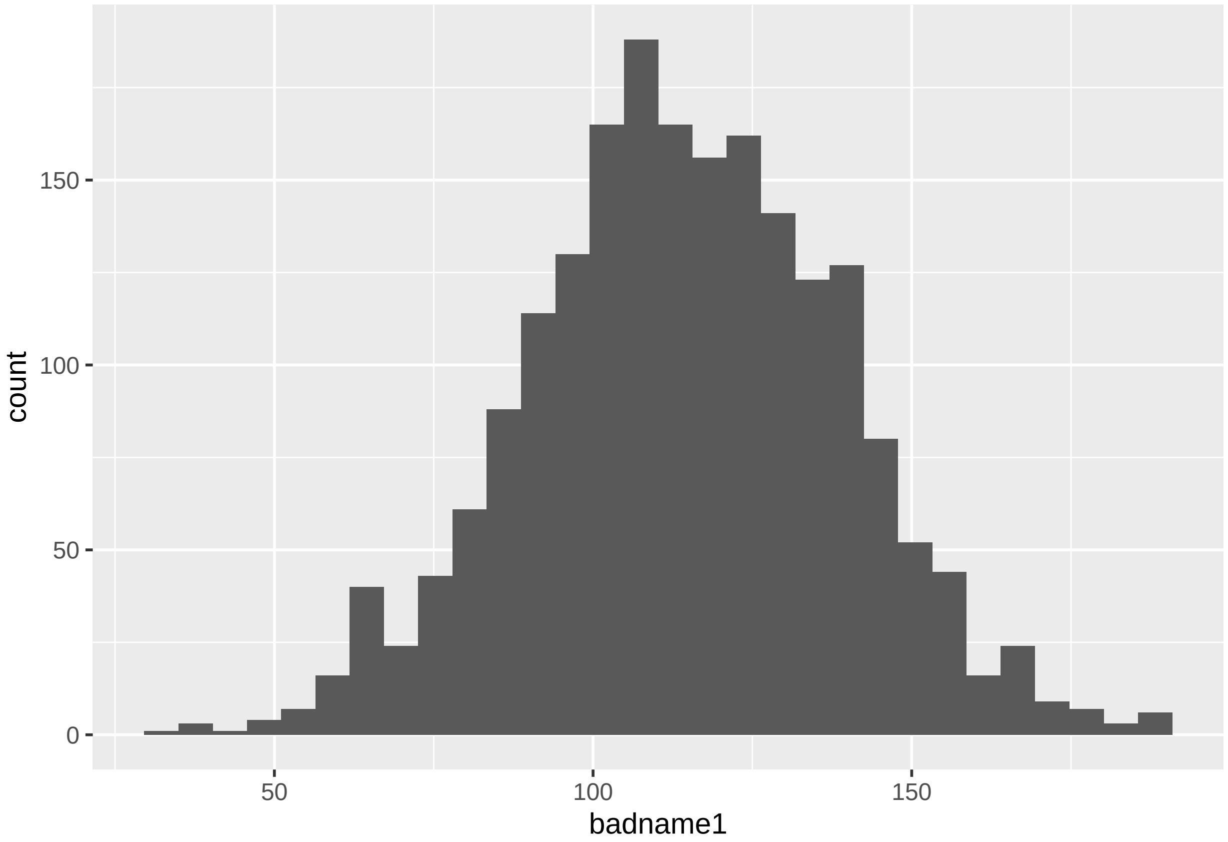
```
# What is it doing?
```

```
# using a tidyverse function read_excel in library readxl, load the data:
example_data <- readxl::read_excel("./data/L01_sample_tdata.xlsx")

# look at it's structure.
str(example_data)
```

```
## tibble [2,000 x 4] (S3: tbl_df/tbl/data.frame)
##  $ badname1: num [1:2000] 74.1 75.2 164.6 90.5 107.4 ...
##  $ badname2: num [1:2000] 129.6 97.3 143.7 85 51.2 ...
##  $ badname3: num [1:2000] 71.7 73.5 162.9 89.5 105.5 ...
##  $ delta1_3: num [1:2000] 2.356 1.683 1.683 0.989 1.951 ...
```

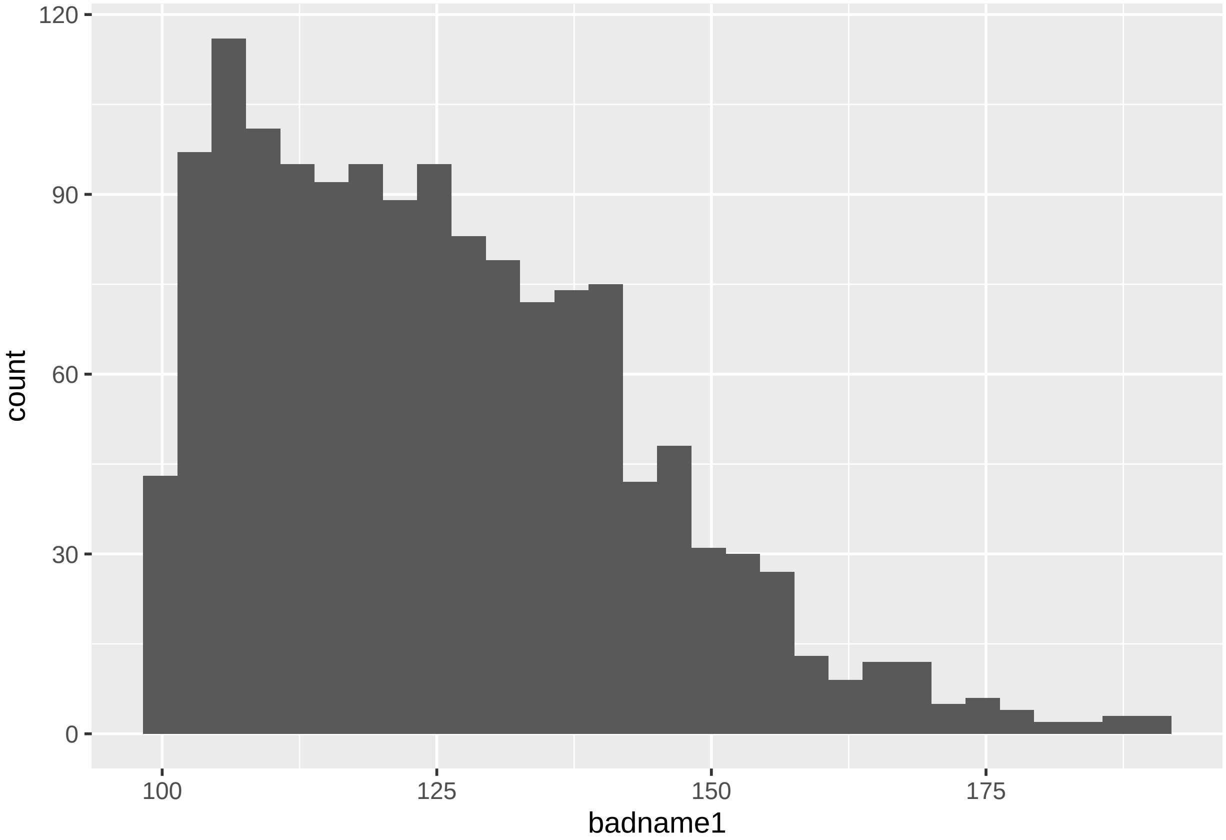
```
ggplot(example_data, aes(badname1))+
  geom_histogram()
```

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



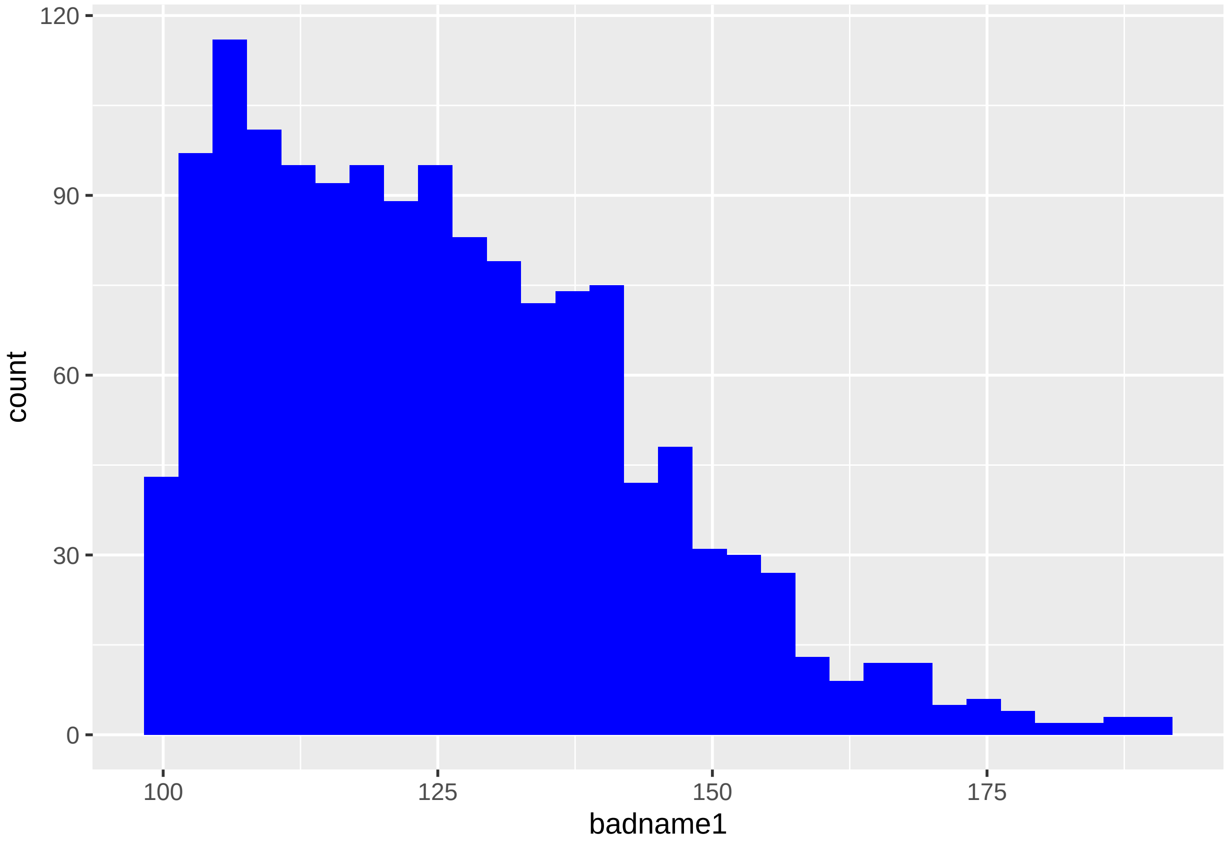
```
filtered_data <- filter(example_data, badname1 > 100)

ggplot(filtered_data, aes(badname1))+
  geom_histogram()
```

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



```
ggplot(filter(example_data, badname1 >100), aes(badname1))+
  geom_histogram(fill="blue")
```

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



```
selected_data <- select(example_data, badname1, badname2)
str(selected_data)
```

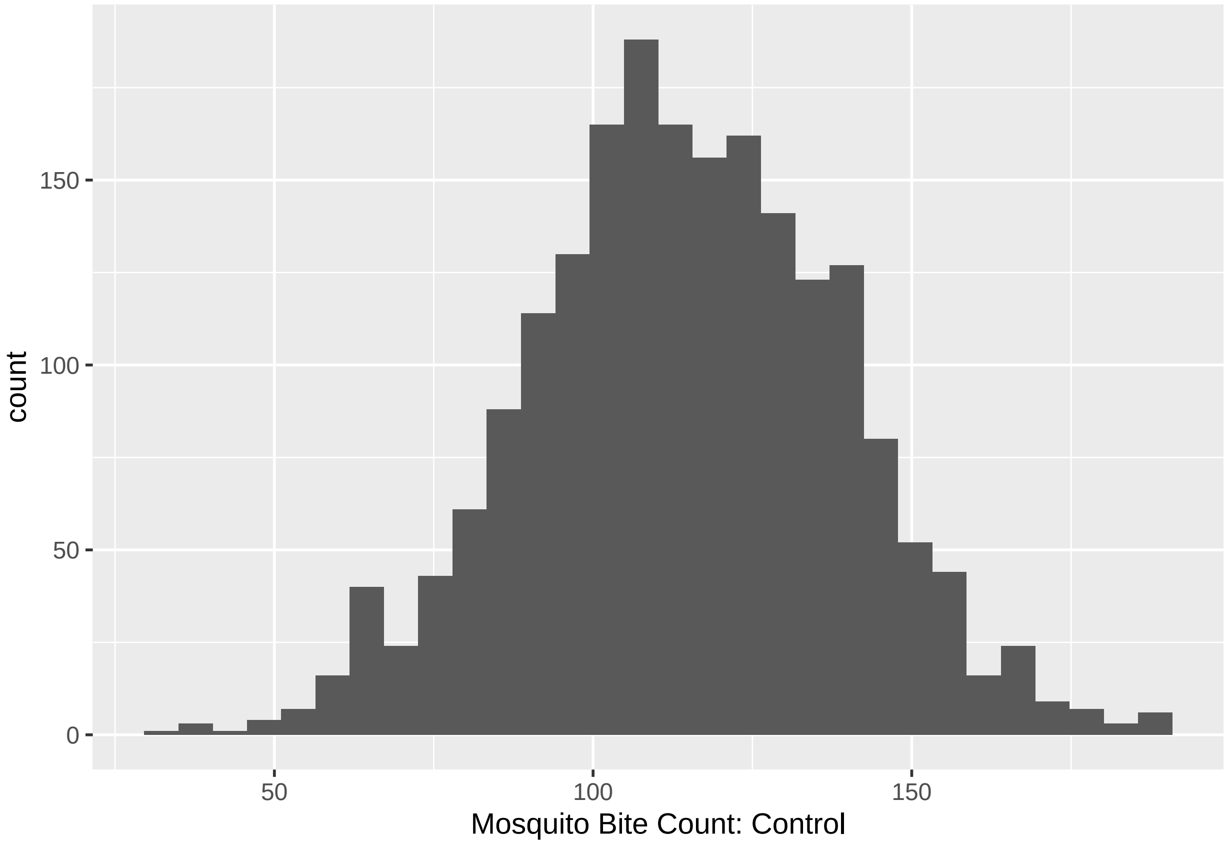
```
## tibble [2,000 x 2] (S3: tbl_df/tbl/data.frame)
##  $ badname1: num [1:2000] 74.1 75.2 164.6 90.5 107.4 ...
##  $ badname2: num [1:2000] 129.6 97.3 143.7 85 51.2 ...
```

```
selected_data_bn <- select(example_data, contains("badname"))
str(selected_data_bn)
```

```
## tibble [2,000 x 3] (S3: tbl_df/tbl/data.frame)
##  $ badname1: num [1:2000] 74.1 75.2 164.6 90.5 107.4 ...
##  $ badname2: num [1:2000] 129.6 97.3 143.7 85 51.2 ...
##  $ badname3: num [1:2000] 71.7 73.5 162.9 89.5 105.5 ...
```

```
lessbad <- rename(example_data,
                  "Mosquito Bite Count: Control" = badname1,
                  "Mosquito Bite Count: Permethrin" = badname2,
                  "Mosquito Bite Count: DEET" = badname3)


ggplot(lessbad, aes(`Mosquito Bite Count: Control`))+
  geom_histogram()
```

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



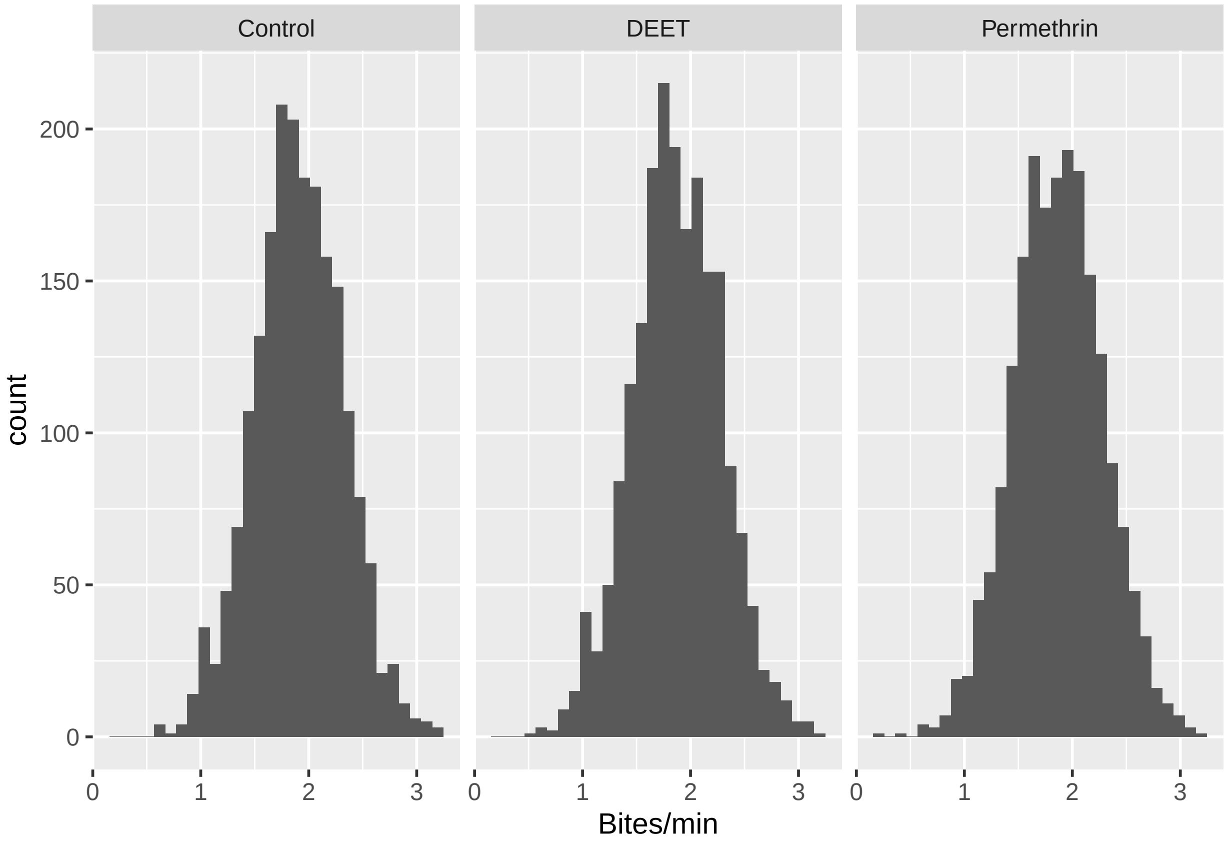
```
lessbad_bites_min <- mutate(lessbad,
                            `Mosquito Bite Count: Control` = `Mosquito Bite Count: Control` / 60,
                            `Mosquito Bite Count: Permethrin` =  `Mosquito Bite Count: Permethrin` / 60
                            `Mosquito Bite Count: DEET` =  `Mosquito Bite Count: DEET` / 60)

lessbad_bites_min1 <- select(lessbad_bites_min, contains("Mosquito"))

# drop delta1_3 by remove (negative selection)
lessbad_bites_minalt <- select(lessbad_bites_min, -delta1_3)

lessbad_bites_min <- lessbad_bites_min1
```

```
# Let's do some carpentry! A very common need in this context.
long_mosquito_data = pivot_longer(lessbad_bites_min, cols=contains("Mosquito"),
                                  names_prefix = "Mosquito Bite Count: ",
                                  names_to = "Repellent",
                                  values_to = "Bites/min")
str(long_mosquito_data)
```

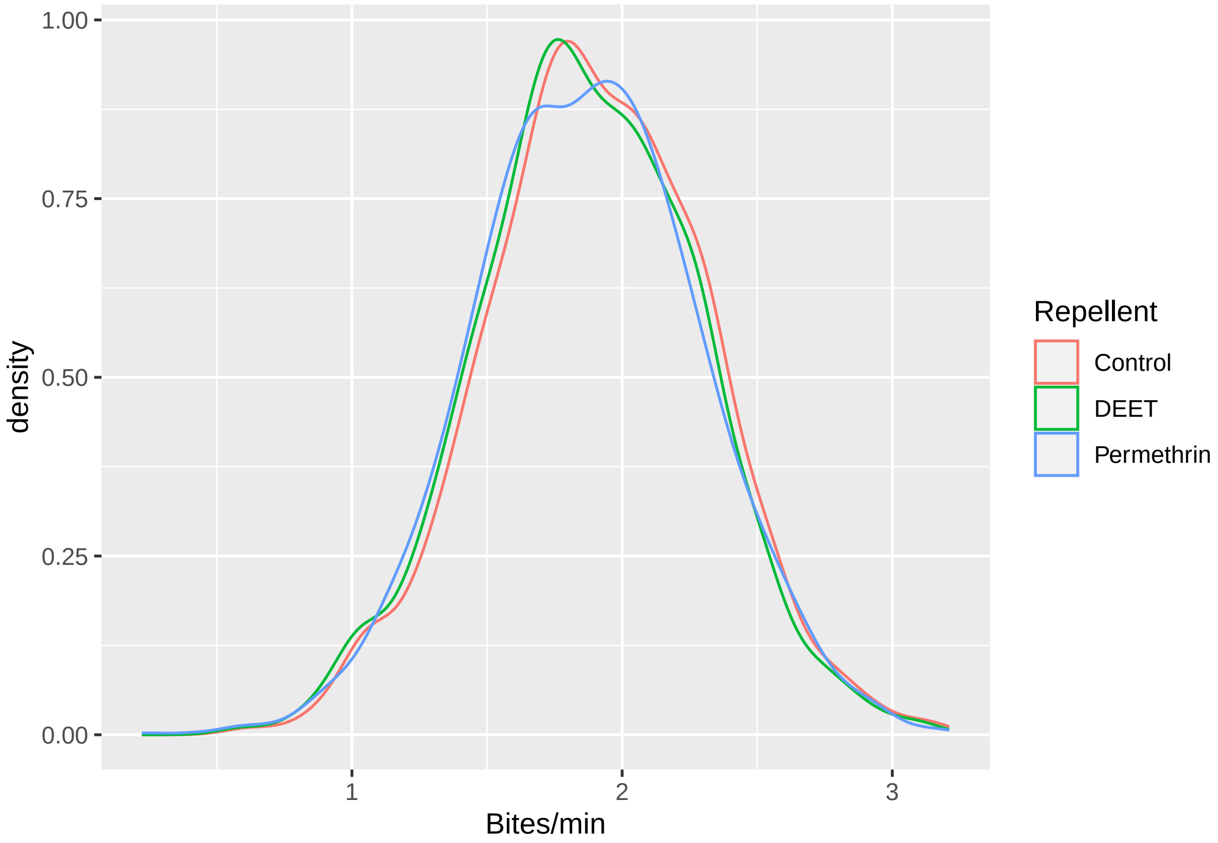
```
## tibble [6,000 x 2] (S3: tbl_df/tbl/data.frame)
##  $ Repellent: chr [1:6000] "Control" "Permethrin" "DEET" "Control" ...
##  $ Bites/min: num [1:6000] 1.23 2.16 1.2 1.25 1.62 ...
```

```
ggplot(long_mosquito_data, aes(`Bites/min`))+
  facet_grid(cols = vars(Repellent))+
  geom_histogram()
```

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

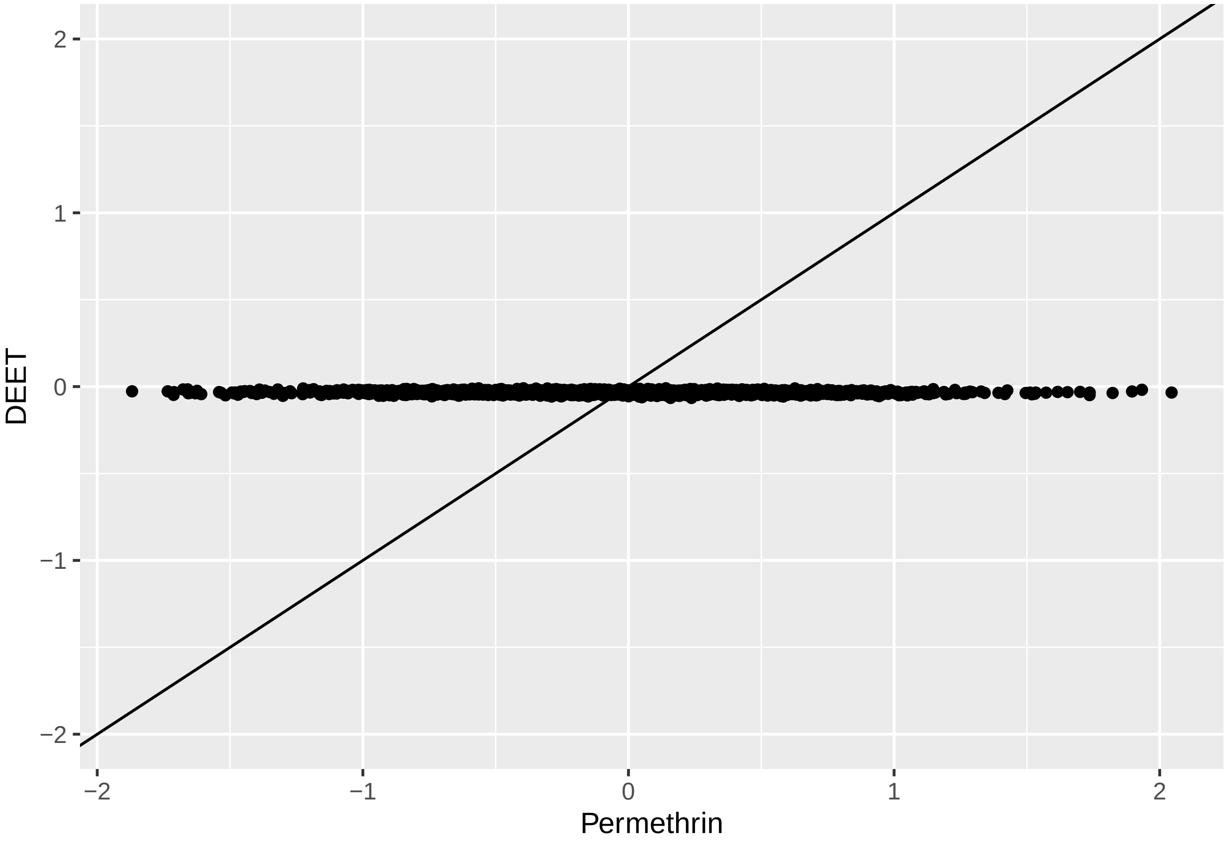


```
ggplot(long_mosquito_data, aes(`Bites/min`, color=Repellent))+
    geom_density()
```

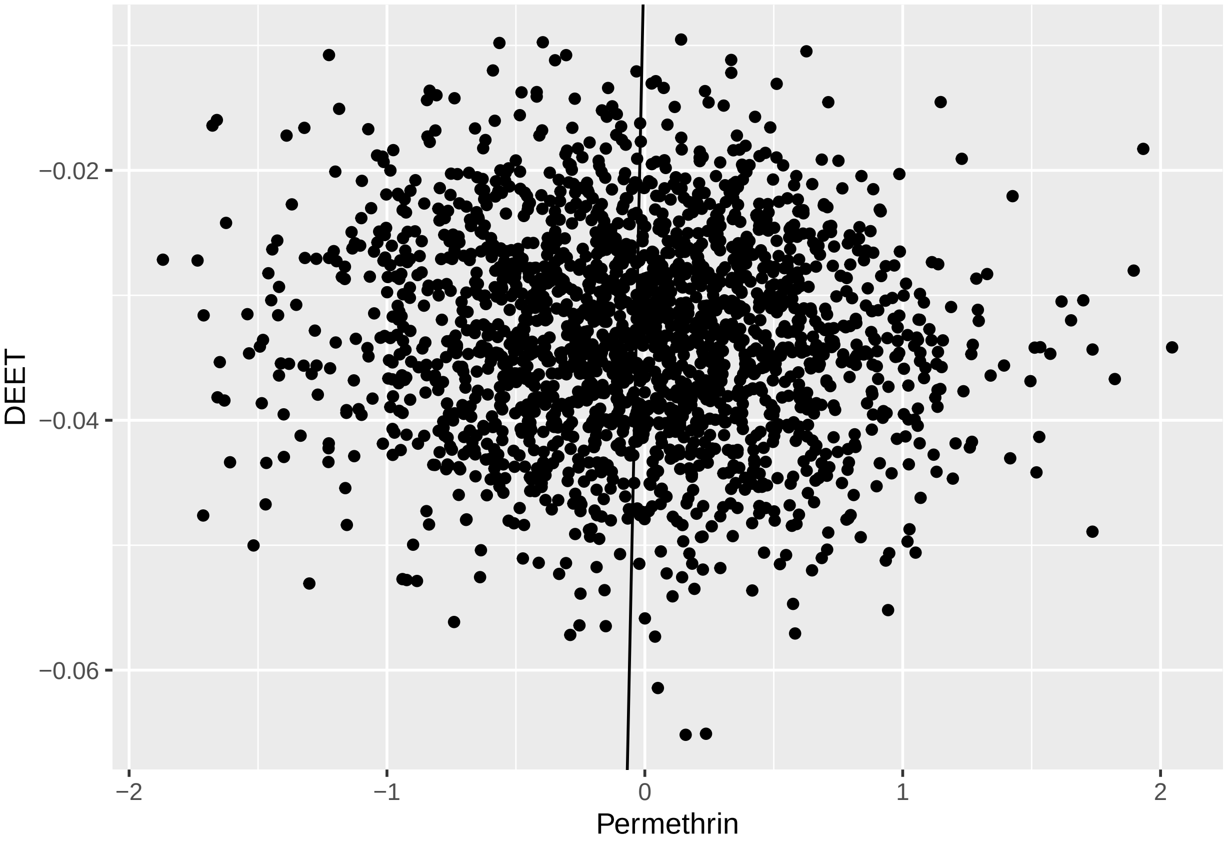


```r
# But what about if we really care about the differences and each row
# in the original data was 1 person?
# we would still be missing other important data, such as ordering, days, etc.
wide_Mozzie_data = transmute(lessbad_bites_min,
                   SubID = row_number(),
                   Permethrin = `Mosquito Bite Count: Permethrin`-`Mosquito Bite Count: Control`,
                   DEET = `Mosquito Bite Count: DEET`-`Mosquito Bite Count: Control`)

# We haven't seen transmute yet and it sounds like alchemy - let's look at the documentation!




# intentional gap.






# transmute is superseded, but not yet deprecated or gone.. so we should update!
# it is equivalent to:
wide_data2 = mutate(lessbad_bites_min,
                   SubID = row_number(),
                   Permethrin = `Mosquito Bite Count: Permethrin`-`Mosquito Bite Count: Control`,
                   DEET = `Mosquito Bite Count: DEET`-`Mosquito Bite Count: Control`,
                   .keep="none")
```

```
ggplot(wide_Mozzie_data, aes(Permethrin, DEET))+
  geom_point()+
  geom_abline(intercept=0, slope=1)+
  ylim(c(-2,2))
```

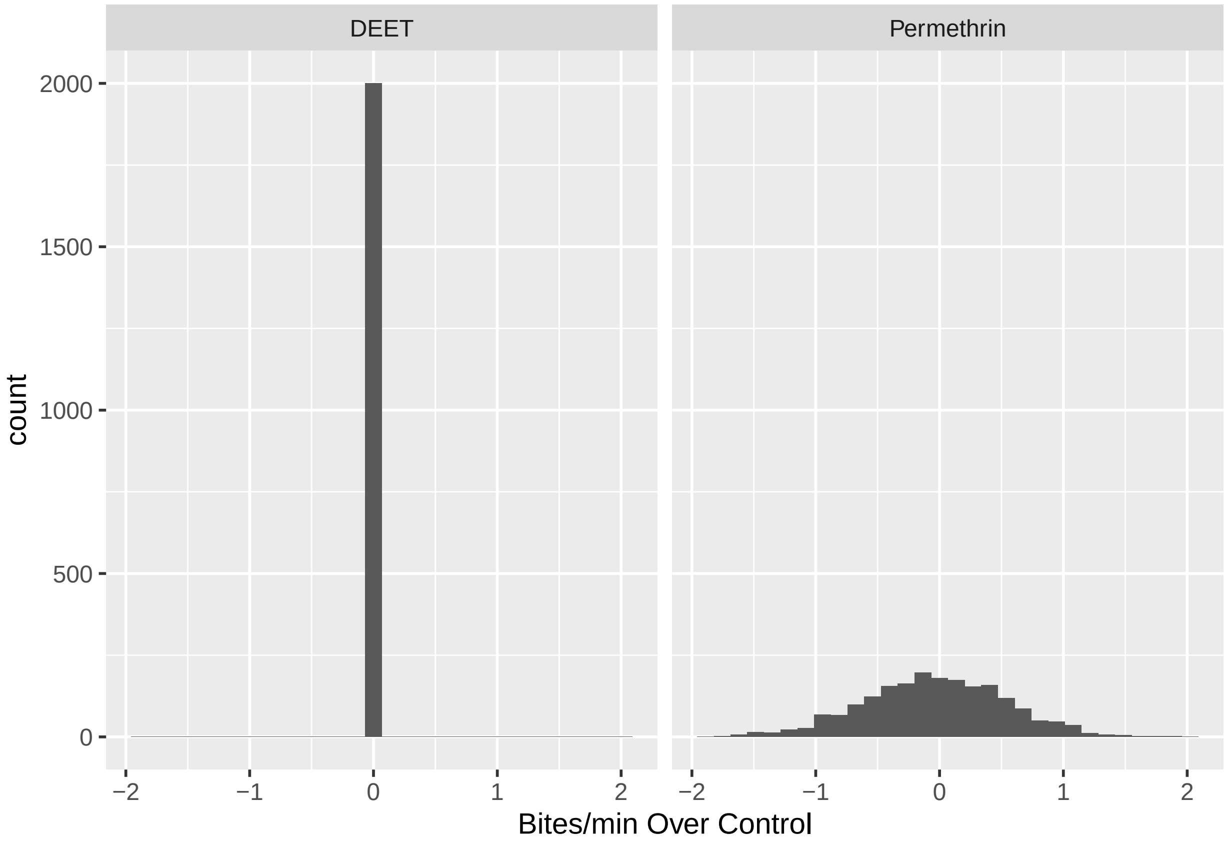


```
ggplot(wide_Mozzie_data, aes(Permethrin, DEET))+
  geom_point()+
  geom_abline(intercept=0, slope=1)
```

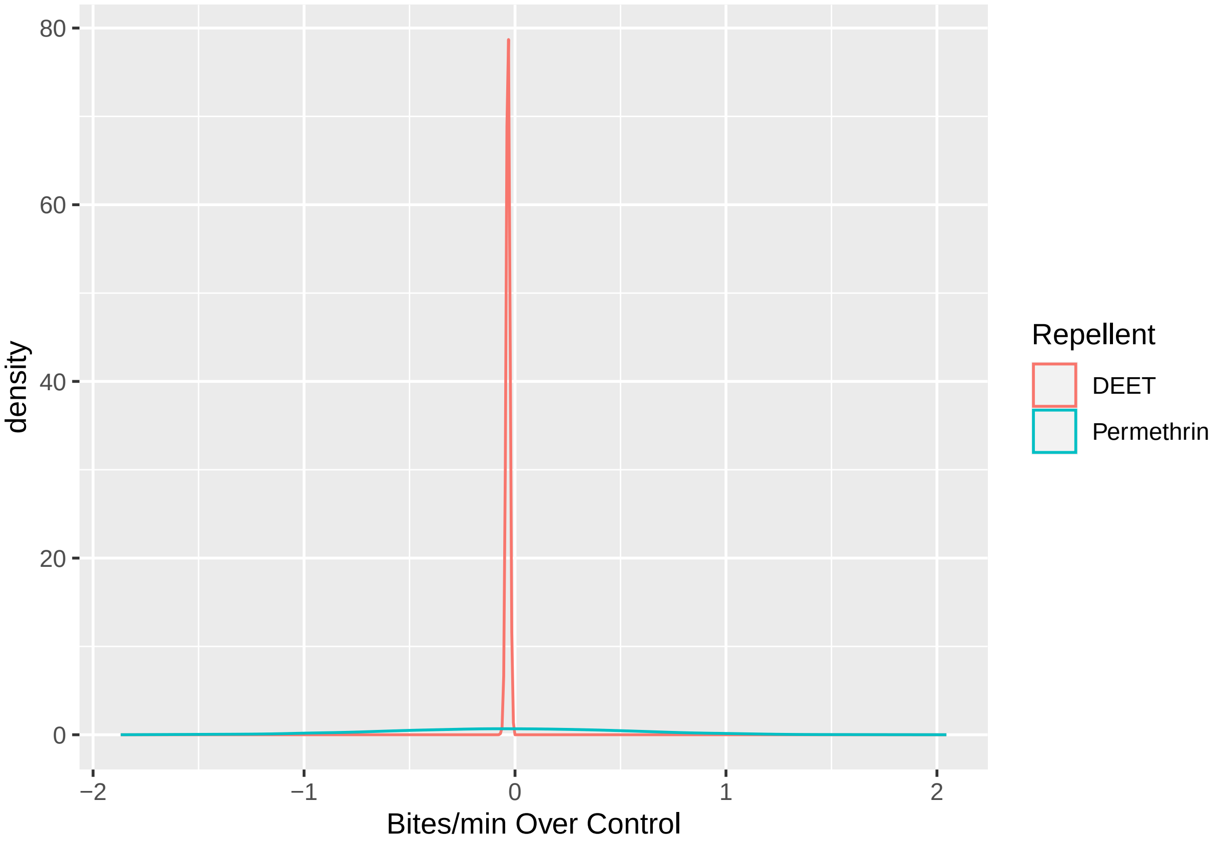


```
long_Mozzie_Delta = pivot_longer(wide_Mozzie_data, !SubID,
                                 names_to = "Repellent",
                                 values_to = "Bites/min Over Control")

ggplot(long_Mozzie_Delta,  aes(`Bites/min Over Control`))+
  facet_grid(cols = vars(Repellent))+
  geom_histogram()
```

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



```
ggplot(long_Mozzie_Delta, aes(`Bites/min Over Control`, color=Repellent))+
    geom_density()
```


80
60
40
20
0
density
−2
−1
0
1
2
Bites/min Over Control
Repellent
DEET
Permethrin