Project 1 - Result

Shengda Hu

1 Project 1. A case study: `storms`

1.1 Team members:

Table 1.1: Members and contribution

Student ID	First Name	Last Name	@mylaurier.ca Email	Contributions

---

`Provide further details on contributions from the team members below:`

NOTE: If you have used AI(s) in producing some of your work, please list the respective AI(s) as a collaborator in the **Team members** table above. In this case, please also describe the contribution from the input of the AI(s) in the textbox above, as well as provide details on how you have used the AI(s) in the process.

1.2 Overview of the data

We will work with the `storms` data set coming with the `dplyr` package.

`storms`: from NOAA Atlantic hurricane database, measured every 6 hours during the lifetime of a storm.

Use `select` to see the numerical, character, and factor variables in the dataframe.

```
# A tibble: 19,066 x 11
    year month   day  hour   lat  long category  wind pressure
   <dbl> <dbl> <int> <dbl> <dbl> <dbl>    <dbl> <int>    <int>
 1  1975     6    27     0  27.5 -79         NA    25     1013
 2  1975     6    27     6  28.5 -79         NA    25     1013
```

```
 3  1975     6    27    12  29.5 -79         NA    25     1013
 4  1975     6    27    18  30.5 -79         NA    25     1013
 5  1975     6    28     0  31.5 -78.8       NA    25     1012
 6  1975     6    28     6  32.4 -78.7       NA    25     1012
 7  1975     6    28    12  33.3 -78         NA    25     1011
 8  1975     6    28    18  34   -77         NA    30     1006
 9  1975     6    29     0  34.4 -75.8       NA    35     1004
10  1975     6    29     6  34   -74.8       NA    40     1002
# i 19,056 more rows
# i 2 more variables: tropicalstorm_force_diameter <int>,
#   hurricane_force_diameter <int>

# A tibble: 19,066 x 1
   name
   <chr>
 1 Amy
 2 Amy
 3 Amy
 4 Amy
 5 Amy
 6 Amy
 7 Amy
 8 Amy
 9 Amy
10 Amy
# i 19,056 more rows

# A tibble: 19,066 x 1
   status
   <fct>
 1 tropical depression
 2 tropical depression
 3 tropical depression
 4 tropical depression
 5 tropical depression
 6 tropical depression
 7 tropical depression
 8 tropical depression
 9 tropical storm
10 tropical storm
# i 19,056 more rows
```

One thing to notice is that `category` has integer value, but is of `double` type. We can see this by getting the distinct values for the `category` variable.

```
# A tibble: 6 x 1
  category
     <dbl>
1       NA
2        1
3        3
4        2
5        4
6        5
```

We will later create a new `ordinal` type (a special `<fctr>` type) variable to capture the same information.

1.3 Rough idea

Get some rough idea on the whole data set, such as period and (rough) total number of storms recorded. Here we follow the reasonable idea that the storms are uniquely determined by the `year` and `name`.

```
# A tibble: 1 x 3
  start_year end_year count
       <dbl>    <dbl> <int>
1       1975     2021   639
```

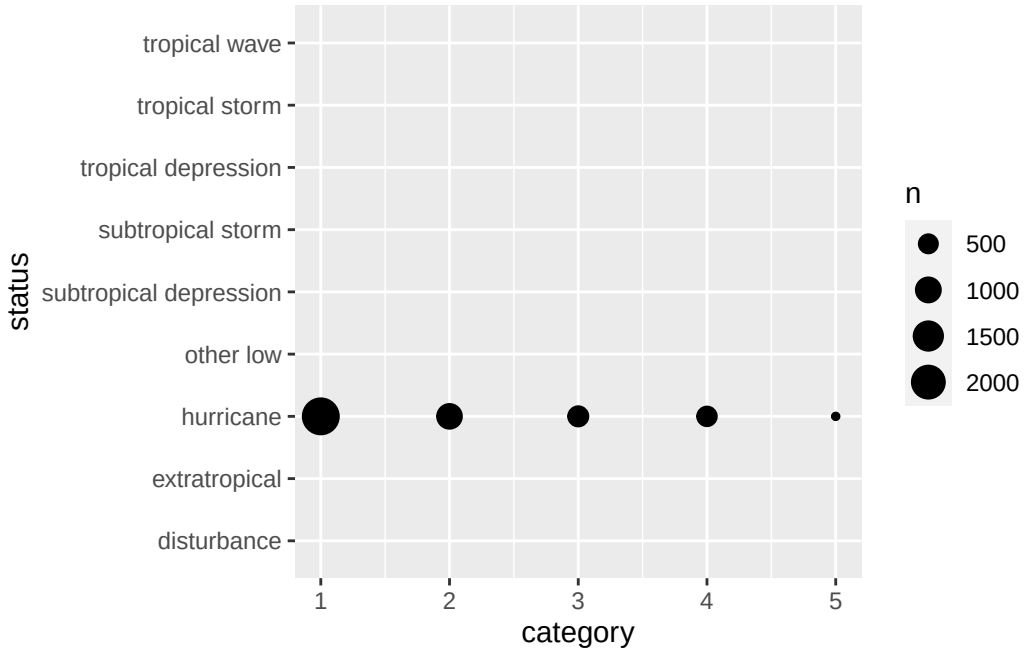
Can summarize along other variables to see their ranges, such as storm strength (in terms of wind speed) and the latitudes storms go to in each year.

```
# A tibble: 47 x 5
    year min_wind max_wind min_lat max_lat
   <dbl>    <int>    <int>   <dbl>   <dbl>
 1  1975       20      120    10.3    55
 2  1976       20      105    12.5    46.5
 3  1977       20      150    17.2    49.5
 4  1978       15      120    12      57
 5  1979       15      150    10      52.5
 6  1980       20      165    10.7    54
 7  1981       20      115    10.5    45.1
 8  1982       20      115    13.7    51.8
 9  1983       20      100    25.1    41.7
10  1984       15      115    10.5    46.2
# i 37 more rows
```

1.4 A bit more details

Learn about how the category corresponds to the status, which is basically covariance between two categorical variables, using geom_count().

```
Warning: Removed 14382 rows containing non-finite values (`stat_sum()`).
```



It does not seem to be interesting, since all the dots are on the same row. **BUT** read the Warning message shown on top of the plot

- Removed 14382 rows containing non-finite values

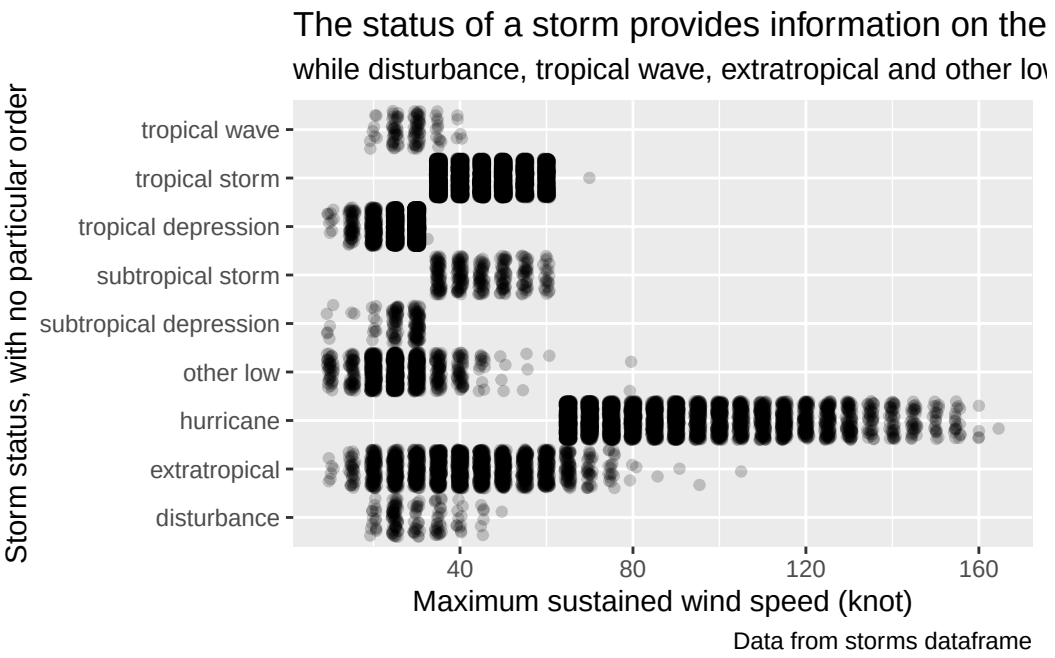
From previous outputs, we see that there must be NA values for category corresponding to other status values. So, it actually indicates

- category is only assigned for hurricanes

Also, from the sizes of the dots, it conforms to the expectation that higher categories contain fewer number of hurricanes.

1.5 Review the data and make adjustments

We are not experts in hurricanes, but with enough data and enough time to mess around, we can learn quite a bit. For instance, plotting `status` v.s. `wind` speed could inform us of how the storms are classified, besides the actual `status` names that already contain some information.



We see that the *depression*s, *storm*s and *hurricane*s are quite neatly separated; while the rest of them, *disturbance*s, *other low*s, *tropical wave*s and *extratropical*s are not exactly cleanly defined. Especially, *other low* and *extratropical* covers quite large variations of wind speed, which is reflected in their names sounding like catch-alls.

One issue: Probably **one** `tropical storm` with a `hurricane` strength `wind`.

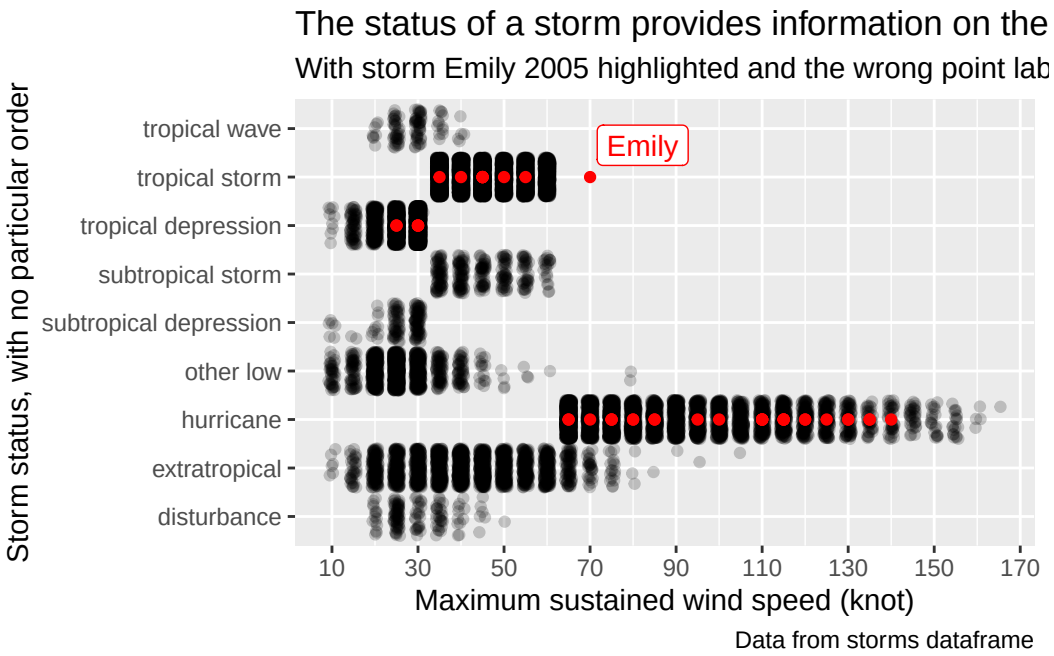
Check it out by filtering for a `tropical storm` that has wind speed larger than `65`.

```
# A tibble: 1 x 13
  name   year month   day  hour   lat  long status      category  wind pressure
  <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <fct>          <dbl> <int>    <int>
1 Emily  2005     7    20    18    25 -98.7 tropical st~       NA    70      975
# i 2 more variables: tropicalstorm_force_diameter <int>,
#   hurricane_force_diameter <int>
```

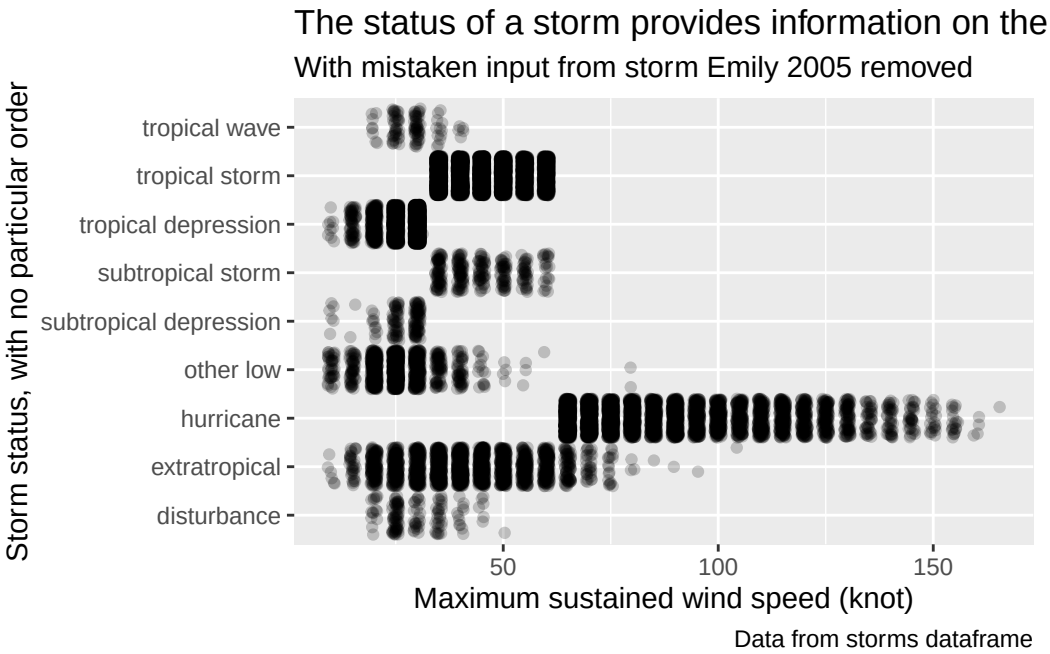
So it is storm `Emily` in `2005`. Let's find out more about the particular storm, by filtering for all the rows about the named storm in the given year.

```
# A tibble: 45 x 13
   name   year month   day  hour   lat  long status      category  wind pressure
   <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <fct>          <dbl> <int>    <int>
 1 Emily  2005     7    11     0  10.7 -42.4 tropical d~       NA    25     1010
 2 Emily  2005     7    11     6  10.8 -43.4 tropical d~       NA    30     1009
 3 Emily  2005     7    11    12  10.9 -44.4 tropical d~       NA    30     1009
 4 Emily  2005     7    11    18  11   -45.4 tropical d~       NA    30     1007
 5 Emily  2005     7    12     0  11   -46.8 tropical s~       NA    35     1006
 6 Emily  2005     7    12     6  11   -48.5 tropical s~       NA    40     1005
 7 Emily  2005     7    12    12  11   -50.2 tropical s~       NA    45     1004
 8 Emily  2005     7    12    18  11   -52   tropical s~       NA    45     1004
 9 Emily  2005     7    13     0  11   -53.7 tropical s~       NA    45     1003
10 Emily  2005     7    13     6  11.1 -55.4 tropical s~       NA    45     1003
# i 35 more rows
# i 2 more variables: tropicalstorm_force_diameter <int>,
#   hurricane_force_diameter <int>
```

Looks like the offending entry is the last one before `Emily` ceased to have hurricane wind strength. We'll chalk it up with data input error. We can replot, color `Emily` observations `red`, and label the offending record by `Emily`:



To make sure that we found the offending case, we can replot the thing filtering out that particular row.



We can also change the particular row, instead of throwing it out, using the `if_else` function.

```
(new_storms <- storms |> # not recommended to use the same variable name
  mutate(
    status = if_else(
      !((name == 'Emily') & (year == 2005) & (month == 7) & (day == 20) & (hour == 18)),
      status,
      'hurricane'
      ),
    category = if_else(
      !((name == 'Emily') & (year == 2005) & (month == 7) & (day == 20) & (hour == 18)),
      category,
      1
      )
    )
)
```

```
# A tibble: 19,066 x 13
   name   year month   day  hour   lat  long status     category  wind pressure
   <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>         <dbl> <int>    <int>
 1 Amy    1975     6    27     0  27.5 -79   tropical d~       NA    25     1013
 2 Amy    1975     6    27     6  28.5 -79   tropical d~       NA    25     1013
 3 Amy    1975     6    27    12  29.5 -79   tropical d~       NA    25     1013
 4 Amy    1975     6    27    18  30.5 -79   tropical d~       NA    25     1013
```

```
 5 Amy    1975     6    28     0  31.5 -78.8 tropical d~       NA    25     1012
 6 Amy    1975     6    28     6  32.4 -78.7 tropical d~       NA    25     1012
 7 Amy    1975     6    28    12  33.3 -78   tropical d~       NA    25     1011
 8 Amy    1975     6    28    18  34   -77   tropical d~       NA    30     1006
 9 Amy    1975     6    29     0  34.4 -75.8 tropical s~       NA    35     1004
10 Amy    1975     6    29     6  34   -74.8 tropical s~       NA    40     1002
# i 19,056 more rows
# i 2 more variables: tropicalstorm_force_diameter <int>,
#   hurricane_force_diameter <int>
```

Then make a new variable `factor_cat` with levels from `0` up to `5`, and assign level `0` when the value of `category` is *missing*.

```
(new2_storms <- new_storms |>
  mutate(
    factor_cat = if_else(is.na(category), 0, category) |>
      factor(
        ordered = TRUE,
        levels = c(0, 1, 2, 3, 4, 5)
        ) # create `factor_cat` to be category as <ord> type
    )
)
```

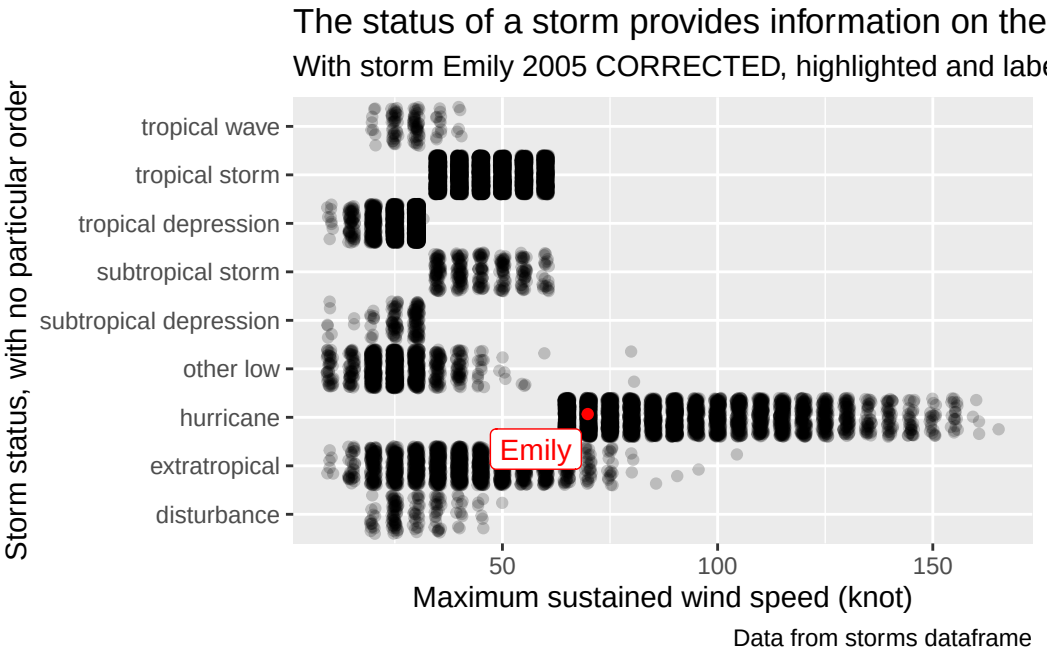
```
# A tibble: 19,066 x 14
   name   year month   day  hour   lat  long status     category  wind pressure
   <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>         <dbl> <int>    <int>
 1 Amy    1975     6    27     0  27.5 -79   tropical d~       NA    25     1013
 2 Amy    1975     6    27     6  28.5 -79   tropical d~       NA    25     1013
 3 Amy    1975     6    27    12  29.5 -79   tropical d~       NA    25     1013
 4 Amy    1975     6    27    18  30.5 -79   tropical d~       NA    25     1013
 5 Amy    1975     6    28     0  31.5 -78.8 tropical d~       NA    25     1012
 6 Amy    1975     6    28     6  32.4 -78.7 tropical d~       NA    25     1012
 7 Amy    1975     6    28    12  33.3 -78   tropical d~       NA    25     1011
 8 Amy    1975     6    28    18  34   -77   tropical d~       NA    30     1006
 9 Amy    1975     6    29     0  34.4 -75.8 tropical s~       NA    35     1004
10 Amy    1975     6    29     6  34   -74.8 tropical s~       NA    40     1002
# i 19,056 more rows
# i 3 more variables: tropicalstorm_force_diameter <int>,
#   hurricane_force_diameter <int>, factor_cat <ord>
```

Comment: `if_else(CONDITION, TRUE_output, FALSE_output)` is a function that computes the `CONDITION` first, and if the `CONDITION` evaluates to `TRUE` it returns the `TRUE_output`; while if the `CONDITION` evaluates to `FALSE`, it returns the `FALSE_output`.

After this,

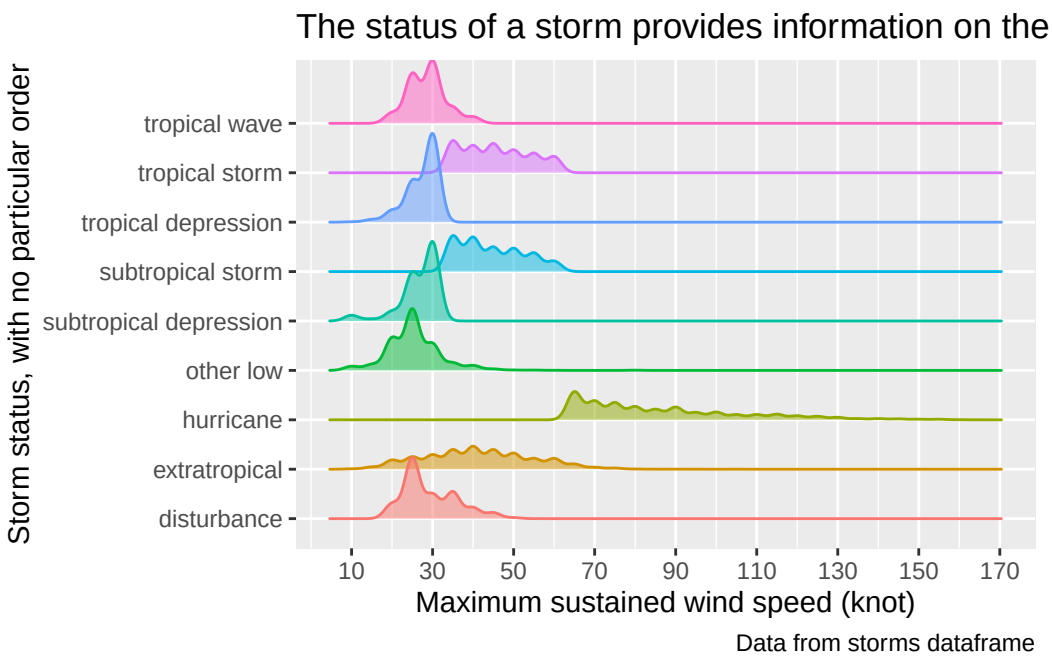
- `new2_storms` have *correct* information concerning `status` and `wind` speed, as far as we understand,
- the `factor_cat` variable in `new2_storms` is of `<ord>` type, with the same information as the `category` variable.

We can plot it and label where the corrected data point for `Emily` goes in red again.



Using `geom_density_ridges` shows distributions better than just the scatter plot.

```
Picking joint bandwidth of 1.8
```



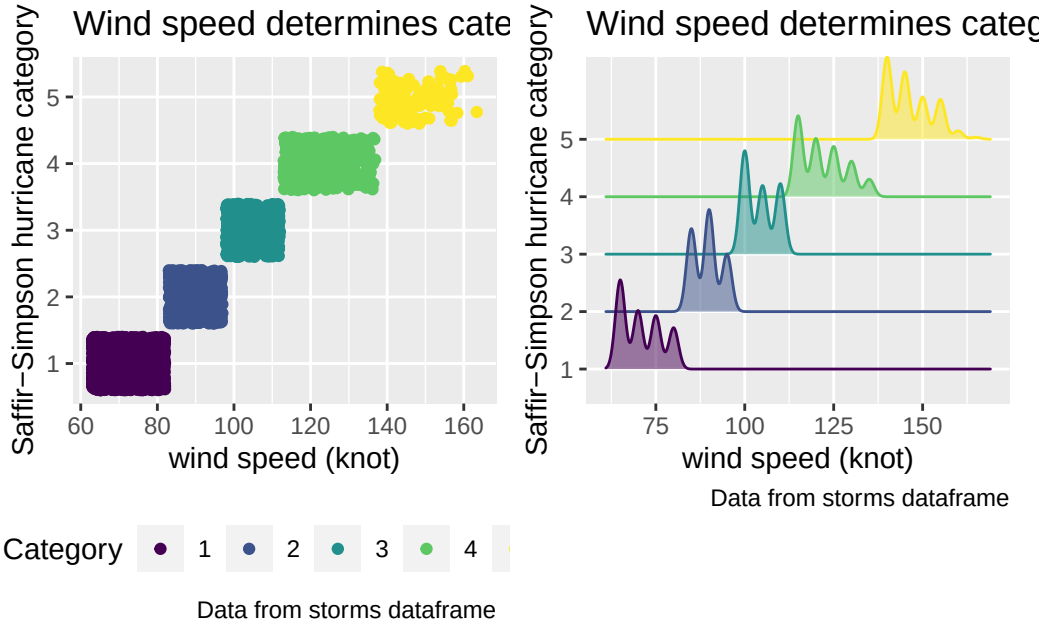
1.6 Simple learning from the data

Using similar method, we can see the definition of the `category` of a hurricane – even though a simple google or `?storms` would already give us the information. First issue is that the `category` is NOT a categorical variable and plotting using it causes various issues. Luckily, we already have a categorical variable `factor_cat` carrying the same information.

Comment: We are not using the full power of a `factor` type variable here. Will come back to `factor` later.

We compare the output of `geom_jitter` and `geom_density_ridges` plots of the category `factor_cat` against `wind` speed, for the `hurricanes`.

```
Picking joint bandwidth of 1.35
```

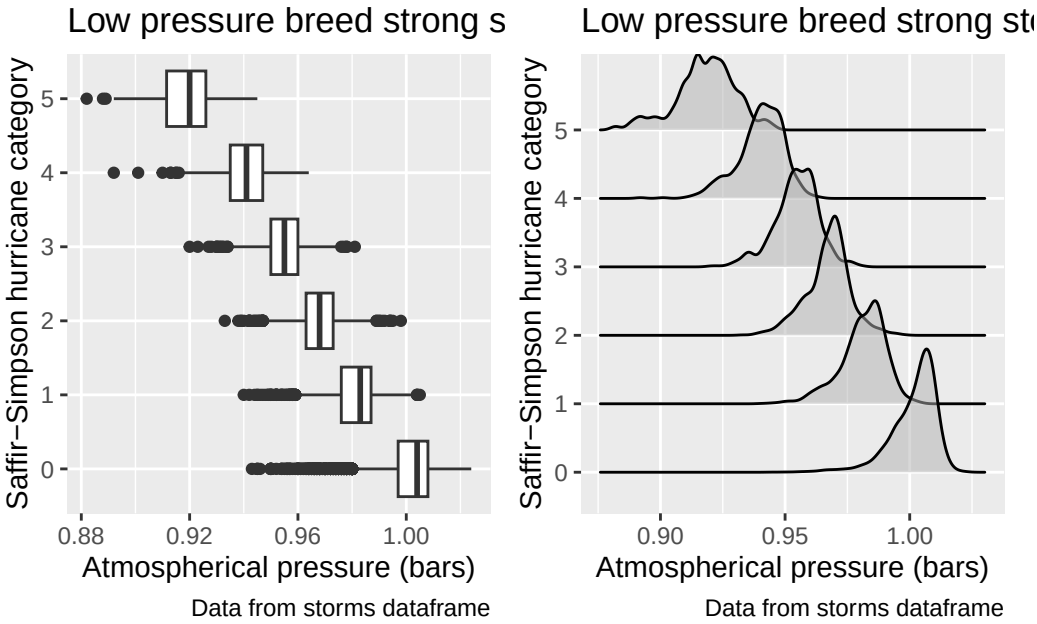


The pattern indicates that data are boxed in, most likely artificially, as there is no way storms box themselves in. Indeed, the notion of category is a concept designed to help human understanding. We can get more details concerning the criteria, even if we know nothing *a priori* about what the rules are, by finding the boundaries defining each category.

```
# A tibble: 6 x 3
  category min_wind max_wind
     <dbl>    <int>    <int>
1        1       65       80
2        2       85       95
3        3      100      110
4        4      115      135
5        5      140      165
6       NA       10      105
```

The strength of a storm is related to the air pressure, so they say. We can see how it is, using simple boxplot or ridge plots. It indeed justifies what we hear. Note that the *standard atmospheric pressure* is slightly above 1 bar, (at 1.01325 bar, see the Wikipedia).

```
Picking joint bandwidth of 2.05
```



Some averages don't make sense, at least not obviously. For example, the following computes the average strength for wind speed recorded in each year.

```
new_storms |>
  summarize(
    mean_wind = mean(wind, na.rm = TRUE),
    .by = year
    )
```

```
# A tibble: 47 x 2
    year mean_wind
   <dbl>     <dbl>
 1  1975      52.4
 2  1976      56.3
 3  1977      49.0
 4  1978      51.6
 5  1979      47.9
 6  1980      60.8
 7  1981      51.8
 8  1982      48.6
 9  1983      45
10  1984      48.4
# i 37 more rows
```

We cannot be exactly sure if the answer makes any sense at all. The average could be low for

many reasons, for instance, maybe more low speeds are kept in the record for some years than others. Moreover, it is the strongest wind speed that do most of the damages, which can not be reflected by the average alone.

1.7 Getting more details on individual storms

Look at individual storms

- Storms are named individually in each year, while the same name may be reused across different years.

We will take the difference of the `first_day` and the `last_day` as the length of the storm, even though there might be exceptions to this rule.

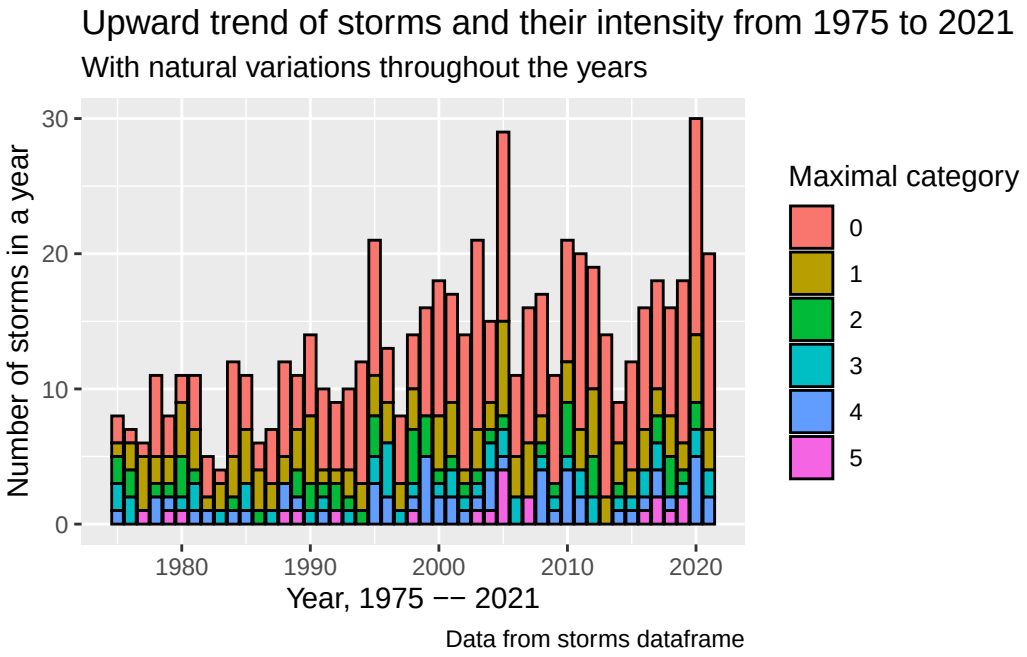
First, `summarize` by individual storms.

```
library(lubridate)
storms_by_storm <- new2_storms |>
  mutate(
    date = make_date(year, month, day)  # <- from lubridate package
    )|>
  summarize(
    first_day = min(date),
    last_day = max(date),
    days = n_distinct(month, day), # gives the actual number of days
    max_cat = max(category |> replace_na(0)), # this is numerical
    med_cat = median(category |> replace_na(0)),
    factor_max_cat = max(factor_cat),
    max_wind = max(wind),
    median_wind = median(wind, na.rm = TRUE),
    avg_hu_diam = mean(hurricane_force_diameter, na.rm = TRUE), # test using na.rm = TRUE
    max_hu_diam = max(hurricane_force_diameter),
    min_pressure = min(pressure, na.rm = TRUE),
    median_pressure = median(pressure, na.rm = TRUE),
    .by = c(year, name)
    )
glimpse(storms_by_storm)
```

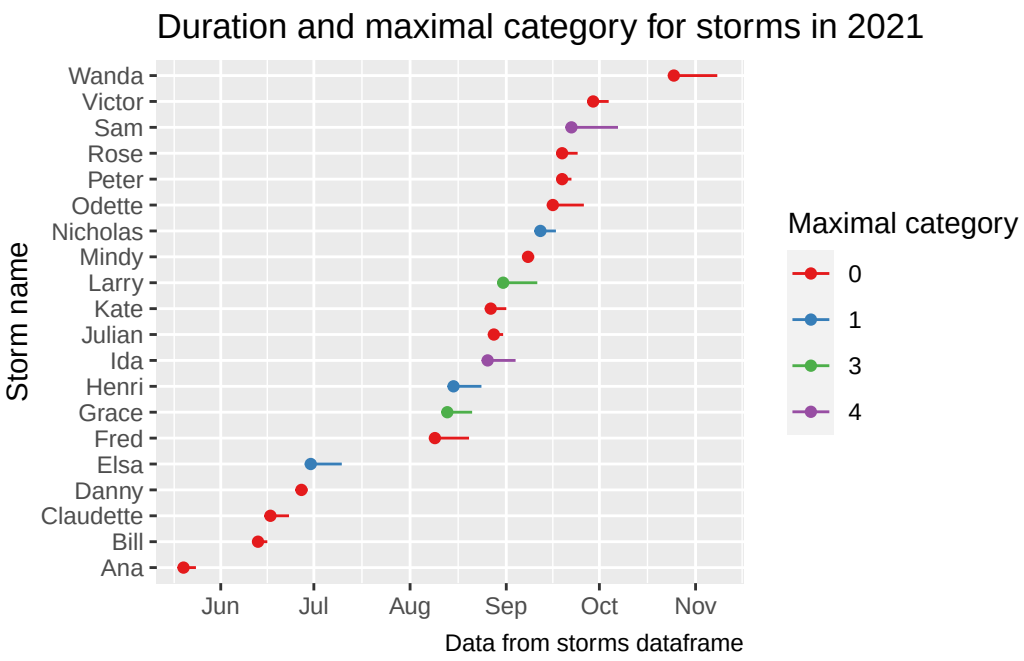
```
Rows: 639
Columns: 14
$ year           <dbl> 1975, 1975, 1975, 1975, 1975, 1975, 1975, 1975, 1976, ~
$ name           <chr> "Amy", "Blanche", "Caroline", "Doris", "Eloise", "Faye~
```

```
$ first_day      <date> 1975-06-27, 1975-07-24, 1975-08-24, 1975-08-28, 1975-~
$ last_day       <date> 1975-07-04, 1975-07-28, 1975-09-01, 1975-09-04, 1975-~
$ days            <int> 8, 5, 9, 8, 12, 6, 13, 5, 5, 4, 6, 15, 8, 10, 5, 6, 7,~
$ max_cat         <dbl> 0, 1, 3, 2, 3, 2, 4, 0, 3, 0, 1, 2, 3, 2, 1, 5, 1, 1, ~
$ med_cat         <dbl> 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, ~
$ factor_max_cat  <ord> 0, 1, 3, 2, 3, 2, 4, 0, 3, 0, 1, 2, 3, 2, 1, 5, 1, 1, ~
$ max_wind        <int> 60, 75, 100, 95, 110, 90, 120, 45, 105, 45, 80, 90, 10~
$ median_wind     <dbl> 50.0, 35.0, 25.0, 65.0, 37.5, 75.0, 65.0, 30.0, 75.0, ~
$ avg_hu_diam     <dbl> NaN, NaN, NaN, NaN, NaN, NaN, NaN, NaN, NaN, NaN, NaN,~
$ max_hu_diam     <int> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA~
$ min_pressure    <int> 981, 980, 963, 965, 955, 977, 939, 1002, 957, 996, 964~
$ median_pressure <dbl> 987.0, 1003.5, 1010.0, 990.0, 1000.0, 985.0, 990.0, 10~
```

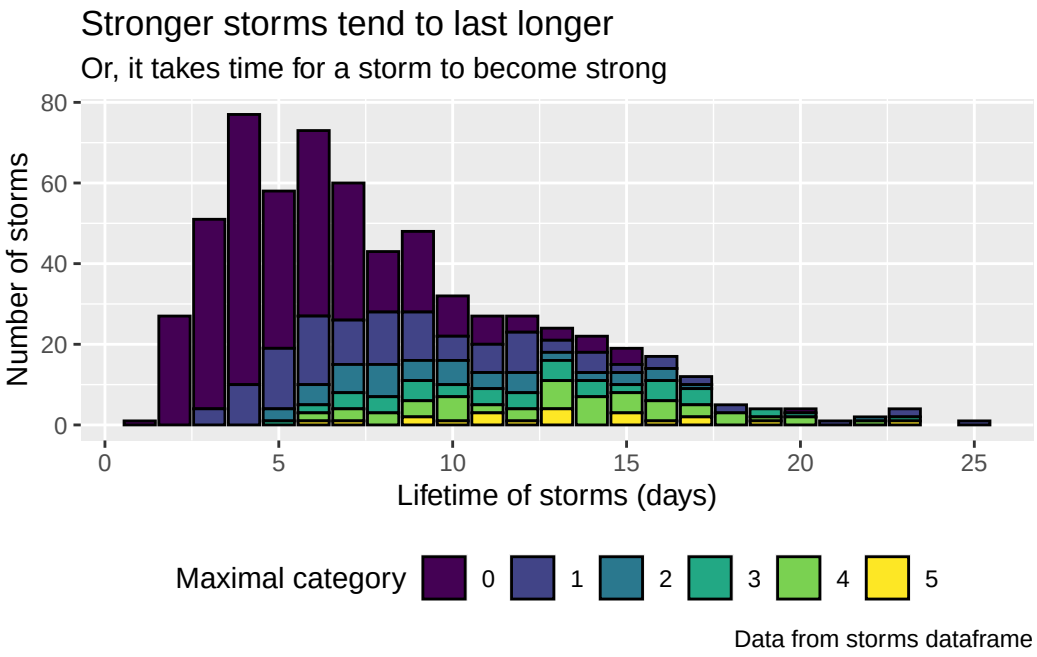
We can look at the number of storms in each year, colored to gain details on maximal strength as `factor_max_cat`



In an individual year (e.g. 2021), we can try to visualize the duration of storms, including their maximal strengths.

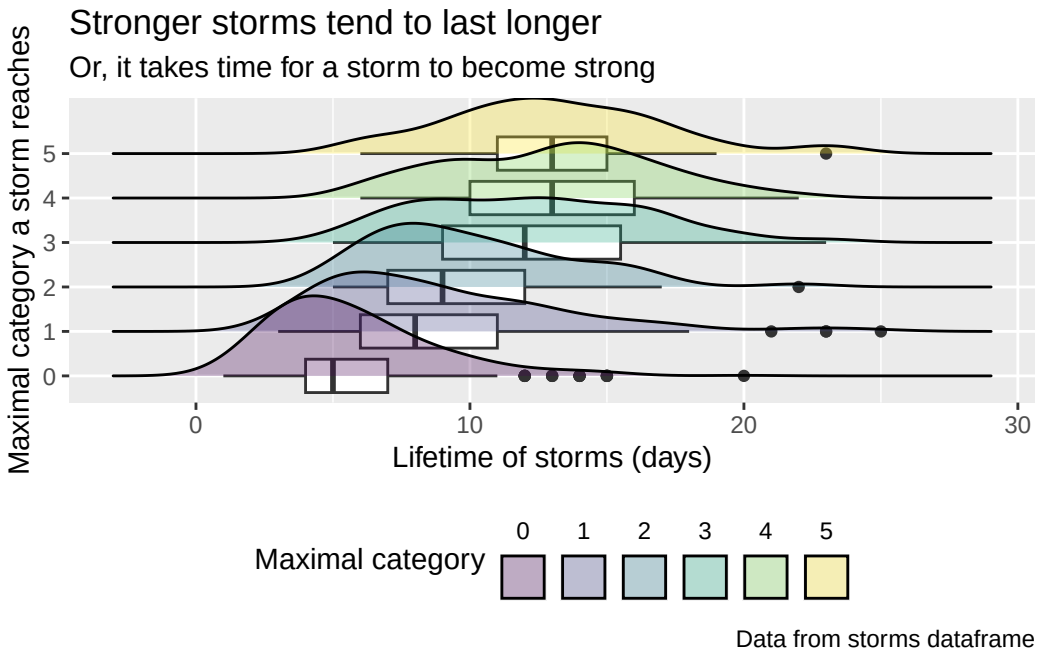


We can also look at the number of storms in terms of their lifetime duration, together with the information on the maximal category they reached by a bar chart.



Or with a combined boxplot and ridges plot

```
Picking joint bandwidth of 1.34
```



While the project will end here, the next step should be more exploratory data analysis, and build models.