Using R for data analysis (SSA) : solutions

Boerhaave Nascholing LUMC

January 23rd, 2024

Introduction

You will analyse the `storms` table which comes with the `tidyverse` package.
Make sure you put `library( tidyverse )` in the R chunk at the top of your R Markdown file as shown here below:

```
library( tidyverse )
```

After the library has been loaded you will have access to the table in the `storms` variable. Each row of `storms` table is an observation of a storm recorded at a certain moment (date and time) at a geographical location (`lat`, `long`). Some additional storm features (`wind` speed, `pressure`, ...), classifications (`status`, `category`) and a `name` are also included.

For more details you may consult the help on storms tibble with `?storms` but the following column description is sufficient for the SSA:

- `name`: Name of the storm.
- `year`, `month`, `day`, `hour`: Date and time of the observation.
- `lat`, `long`: Geographical location of the storm centre (numbers).
- `wind`: Wind speed (number, in knots).
- `pressure`: Pressure at the storm's centre (number, in millibars).
- `tropicalstorm_force_diameter` (or `ts_diameter` in older versions of `tidyverse` library): Storm diameter (number, in nautical miles).
- `status`: Storm classification (a factor, many levels).
- `category`: Storm category (a number, range: -1..5; many values are missing).

Note, that a single storm is usually observed multiple times (so one storm may be described in multiple rows).

Here is a random part of the table (some columns are omitted):

```
# A tibble: 6 x 9
  name   year month   lat  long status         category  wind pressure
  <chr> <dbl> <dbl> <dbl> <dbl> <fct>             <dbl> <int>    <int>
1 Emily  1981     9  42.7 -41   extratropical        NA    30     1008
2 Klaus  1984    11  22.7 -58.7 hurricane             1    80      980
3 Hugo   1989     9  16.6 -62.5 hurricane             4   125      949
4 Ana    1991     7  37.9 -61.1 tropical storm       NA    45     1000
5 Ivan   2004     9  11.6 -59.4 hurricane             3   100      963
6 Kirk   2018     9  11   -46.8 tropical wave        NA    35     1007
```

Questions

Question 1: [4p] Percentage of storms with category at least 4.

Out of all storm measurements with non-missing `category` value, calculate the *percentage* of the storm observations that have `category` at least `4`. Find how to use `round` to round the result to 2 decimal places.

Assign the result to the `largeCategoryPercentage` variable.

```
# largeCategoryPercentage <- ...
```

```
# 1p the condition >= (at least 4) is correct
# 1p the number of observations with category known correct
# 1p the percentage correct
# 1p the rounding correct

largeCategoryPercentage <-
  ( storms %>% filter( !is.na( category ), category >= 4 ) %>% nrow() ) /
  ( storms %>% filter( !is.na( category ) ) %>% nrow() ) *
  100
largeCategoryPercentage <- round( largeCategoryPercentage, 2 )
largeCategoryPercentage
```

```
[1] 13.96
```

Question 2: [3p] Changing factor levels, counting occurrences.

Take the data from the `status` column and change the order of levels such that the first three levels are `("tropical storm", "tropical depression", "hurricane")` (in exactly this order).
Then, produce a table of counts of the number of observations for each storm `status` level.
Store the result in `statusCounts` variable.
Note: Do not modify the original `storms` table (a changed table may not work in other questions).

```
# statusCounts <- ...
```

```
# 1p some fct levels reordering is done
# 1p the order of levels is correct
# 1p the table of counts is correct

statusCounts <- storms$status %>%
  fct_relevel( "tropical storm", "tropical depression", "hurricane" ) %>%
  fct_count()
statusCounts
```

```
# A tibble: 9 x 2
  f                         n
  <fct>                 <int>
1 tropical storm         6684
2 tropical depression    3525
3 hurricane              4684
4 disturbance             146
5 extratropical          2068
6 other low              1405
7 subtropical depression  151
8 subtropical storm       292
9 tropical wave           111
```

Question 3: [7p] Table summary in a list.

Create a list with some summaries of the `storms` table and assign this list to the variable `stormsSummary`.
The list should have the following three elements:

- `obsNum` – the *number* of observations in the `storms` table,
- `avgWind` – the mean of observed `wind` speeds (force removal of missing values),

- `uniqueNames` – a *character vector* of names from the `name` column with duplicates removed, sorted in alphabetical order.

```
# stormsSummary <- ...
```

```
# 1p there is a list
# 1p elements in the list have names
# 1p obsNum is correct (nrow)
# 1p mean is calculated
# 1p NAs are skipped in mean calculation
# 1p names are uniqued
# 1p names are sorted

stormsSummary <- list(
  obsNum = nrow( storms ),
  avgWind = mean( storms$wind, na.rm = TRUE ),
  uniqueNames = storms$name %>% unique() %>% sort()
)
stormsSummary
```

```
$obsNum
[1] 19066

$avgWind
[1] 50.01741

$uniqueNames
  [1] "AL011993"  "AL012000"  "AL021992"  "AL021994"  "AL021999"  "AL022000"
  [7] "AL022001"  "AL022003"  "AL022006"  "AL031987"  "AL031992"  "AL041991"
 [13] "AL042000"  "AL051994"  "AL061988"  "AL061995"  "AL061997"  "AL062003"
 [19] "AL071999"  "AL072002"  "AL072003"  "AL081992"  "AL081994"  "AL091994"
 [25] "AL092000"  "AL092001"  "AL092003"  "AL101991"  "AL101993"  "AL101994"
 [31] "AL102004"  "AL111999"  "AL121991"  "AL121999"  "AL141995"  "AL142002"
 [37] "AL142003"  "Al202011"  "Alberto"   "Alex"      "Alicia"    "Allen"
 [43] "Allison"   "Alpha"     "Amelia"    "Amy"       "Ana"       "Andrea"
 [49] "Andrew"    "Anita"     "Arlene"    "Arthur"    "Babe"      "Barry"
 [55] "Belle"     "Bertha"    "Beryl"     "Bess"      "Beta"      "Bill"
 [61] "Blanche"   "Bob"       "Bonnie"    "Bret"      "Candice"   "Caroline"
 [67] "Cesar"     "Chantal"   "Charley"   "Chris"     "Cindy"     "Clara"
 [73] "Claudette" "Colin"     "Cora"      "Cristobal" "Danielle"  "Danny"
 [79] "David"     "Dean"      "Debby"     "Debra"     "Delta"     "Dennis"
 [85] "Diana"     "Dolly"     "Don"       "Dorian"    "Doris"     "Dorothy"
 [91] "Dottie"    "Earl"      "Edouard"   "Eight"     "Elena"     "Eleven"
 [97] "Ella"      "Eloise"    "Elsa"      "Emily"     "Emmy"      "Epsilon"
[103] "Erika"     "Erin"      "Ernesto"   "Eta"       "Evelyn"    "Fabian"
[109] "Fay"       "Faye"      "Felix"     "Fernand"   "Fifteen"   "Fiona"
[115] "Five"      "Florence"  "Flossie"   "Floyd"     "Four"      "Fran"
[121] "Frances"   "Franklin"  "Fred"      "Frederic"  "Frieda"    "Gabrielle"
[127] "Gamma"     "Gaston"    "Georges"   "Gert"      "Gilbert"   "Gladys"
[133] "Gloria"    "Gonzalo"   "Gordon"    "Grace"     "Greta"     "Gustav"
[139] "Hallie"    "Hanna"     "Harvey"    "Helene"    "Henri"     "Hermine"
[145] "Holly"     "Hope"      "Hortense"  "Hugo"      "Humberto"  "Ian"
[151] "Ida"       "Igor"      "Ike"       "Imelda"    "Ingrid"    "Iota"
[157] "Irene"     "Iris"      "Irma"      "Isaac"     "Isabel"    "Isaias"
[163] "Isidore"   "Ivan"      "Jeanne"    "Jerry"     "Joan"      "Joaquin"
```

```
[169] "Jose"      "Josephine" "Joyce"     "Juan"      "Julia"     "Julian"
[175] "Juliet"    "Karen"     "Karl"      "Kate"      "Katia"     "Katrina"
[181] "Keith"     "Kendra"    "Kirk"      "Klaus"     "Kyle"      "Larry"
[187] "Laura"     "Lee"       "Lenny"     "Leslie"    "Lili"      "Lisa"
[193] "Lorenzo"   "Luis"      "Marco"     "Maria"     "Marilyn"   "Matthew"
[199] "Melissa"   "Michael"   "Michelle"  "Mindy"     "Mitch"     "Nadine"
[205] "Nana"      "Nate"      "Nestor"    "Nicholas"  "Nicole"    "Nine"
[211] "Nineteen"  "Noel"      "Odette"    "Olga"      "Omar"      "One"
[217] "Opal"      "Ophelia"   "Oscar"     "Otto"      "Pablo"     "Paloma"
[223] "Patty"     "Paula"     "Paulette"  "Peter"     "Philippe"  "Rafael"
[229] "Rene"      "Richard"   "Rina"      "Rita"      "Rose"      "Roxanne"
[235] "Sally"     "Sam"       "Sandy"     "Sean"      "Sebastien" "Shary"
[241] "Sixteen"   "Stan"      "Tammy"     "Tanya"     "Teddy"     "Ten"
[247] "Theta"     "Three"     "Tomas"     "Tony"      "Two"       "Vicky"
[253] "Victor"    "Vince"     "Wanda"     "Wilfred"   "Wilma"     "Zeta"
```

Question 4: [6p] Dropping summer storms

Create a new tibble `stormsNoSummer` that contains all observations from `storms` except those that were made in a summer. Consider 21st of June to be the first day of summer and 22nd of September to be the last day of summer.

```
# stormsNoSummer <- ...
```

```
# 1p any filtering is done
# 1p the filtering is correct for months < 6
# 1p the filtering is correct for months == 6
# 1p the filtering is correct for months == 7,8
# 1p the filtering is correct for months == 9
# 1p the filtering is correct for months > 9

stormsNoSummer <- storms %>%
  filter(
    month < 6 |
    ( month == 6 & day < 21 ) |
    ( month == 9 & day > 22 ) |
    month > 9
  )
stormsNoSummer
```

```
# A tibble: 7,023 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 Eloise  1975     9    23     0  27.3 -88.5 hurricane         2    95      968
 2 Eloise  1975     9    23     6  28.4 -87.3 hurricane         3   105      958
 3 Eloise  1975     9    23    12  30.2 -86.3 hurricane         3   110      955
 4 Eloise  1975     9    23    18  33   -85.7 tropical ~       NA    55      982
 5 Eloise  1975     9    24     0  35.5 -84.3 tropical ~       NA    30      999
 6 Eloise  1975     9    24     6  36.5 -83.5 extratrop~       NA    20     1004
 7 Eloise  1975     9    24    12  37   -82.5 extratrop~       NA    20     1004
 8 Eloise  1975     9    24    18  37.5 -81.5 extratrop~       NA    20     1004
 9 Faye    1975     9    24    18  23   -56.9 tropical ~       NA    25     1005
10 Faye    1975     9    25     0  23.8 -57.2 tropical ~       NA    30     1005
# i 7,013 more rows
# i 2 more variables: tropicalstorm_force_diameter <int>,
#   hurricane_force_diameter <int>
```

Question 5: [6p] Summarizing storms by month.

Build a *tibble* reporting the fastest wind and the lowest pressure observed over all years in each `month`. Report also the total number of observations for each `month`. During the min/max calculations force omitting possible missing values in the respective columns.
The final table should have four columns: `month`, `fastestWind`, `lowestPressure`, `obsNum` and it should be sorted in descending order of the number of observations (the most frequent at the top row). Store the result in the variable `stormsByMonth`.

```
# stormsByMonth <- ...
```

```
# 1p grouping is good
# 1p obsNum is correct
# 1p lowestPressure is correct (NAs removed)
# 1p fastestWind is correct (NAs removed)
# 1p the table is sorted
# 1p the table is sorted in descending order

stormsByMonth <- storms %>%
  group_by( month ) %>%
  summarise(
    fastestWind = max( wind, na.rm = TRUE ),
    lowestPressure = min( pressure, na.rm = TRUE ),
    obsNum = n()
  ) %>%
  arrange( desc( obsNum ) )
stormsByMonth
```

```
# A tibble: 10 x 4
   month fastestWind lowestPressure obsNum
   <dbl>       <int>          <int>  <int>
 1     9         160            888   7509
 2     8         165            899   4440
 3    10         160            882   3077
 4     7         140            929   1603
 5    11         135            917   1109
 6     6          80            958    779
 7    12          75            979    212
 8     5          60            989    201
 9     1          75            978     70
10     4          55            986     66
```

Question 6. [4p] Cross-tabulation

Create a *tibble* `stormsByStatusAndMonth` that contains a cross-tabulation of `status` and `month`. The result should be a table with `status` represented by rows, `month` in columns, and table values representing the number of observations for each combination of `month` and `status` values. Some entries in the crosstable will be `NA`: check the manual and fill them with zeros.

```
# stormsByStatusAndMonth <- ...
```

```
# 1p counting is correct
# 1p spreading is used
# 1p spreading is correct
# 1p NAs are replaced by zeroes
```

```r
stormsByStatusAndMonth <- storms %>%
  count( status, month ) %>%
  pivot_wider( names_from = month, values_from = n, values_fill = 0L )
  #spread( month, n, fill = 0L )
stormsByStatusAndMonth
```

```
# A tibble: 9 x 11
  status               `6`   `7`   `8`   `9`  `10`  `11`   `1`   `4`   `5`  `12`
  <fct>              <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
1 disturbance           13    45    25    41    14     8     0     0     0     0
2 extratropical        130   135   275   732   520   175    29    40    18    14
3 hurricane             18   202  1038  2380   799   209     5     0     0    33
4 other low             82   175   317   446   219    81     5     0    49    31
5 subtropical depre~    35    11    36    34    22     4     0     4     5     0
6 subtropical storm     12     6    23    72    66    42     6     3    20    42
7 tropical depressi~   213   397   975  1315   413   139     2     1    49    21
8 tropical storm       276   625  1696  2448  1024   443    23    18    60    71
9 tropical wave          0     7    55    41     0     8     0     0     0     0
```

Question 7. [9p] Adding wind speed in km/h and its category.

Wind speed in the `wind` column is given in knots. Create a new column `windKPH` that expresses wind speed in km/h (1 knot = 1.852 km/h). Then, create a new column `windCategory` that contains a factor with levels `"low"`, `"medium"`, `"high"` (exactly in that order). The levels should be determined by the `windKPH` column values: `"low"` for `windKPH` < 75, `"medium"` for `windKPH` < 150, and `"high"` otherwise. The final table should only have columns: `name`, `windCategory` and `windKPH` (exactly in this order). Store the result in the variable `stormsWithWindCategory`.

```r
# stormsWithWindCategory <- ...
```

```r
# 1p windKPH column added
# 1p windKPH is correct
# 1p windCategory column added
# 1p windCategory at least one category has correct condition
# 1p windCategory all categories have correct conditions
# 1p windCategory is a factor
# 1p windCategory has correct levels
# 1p the table has correct columns
# 1p the table has correct column order

stormsWithWindCategory <- storms %>%
  mutate( windKPH = wind * 1.852 ) %>%
  mutate( windCategory = case_when(
    windKPH < 75 ~ "low",
    windKPH < 150 ~ "medium",
    TRUE ~ "high"
  ) %>% factor( levels = c( "low", "medium", "high" ) ) ) %>%
  select( name, windCategory, windKPH )
stormsWithWindCategory
```

```
# A tibble: 19,066 x 3
   name  windCategory windKPH
   <chr> <fct>          <dbl>
 1 Amy   low             46.3
 2 Amy   low             46.3
```

```
 3 Amy   low             46.3
 4 Amy   low             46.3
 5 Amy   low             46.3
 6 Amy   low             46.3
 7 Amy   low             46.3
 8 Amy   low             55.6
 9 Amy   low             64.8
10 Amy   low             74.1
# i 19,056 more rows
```

Question 8: [7p] A box plot.

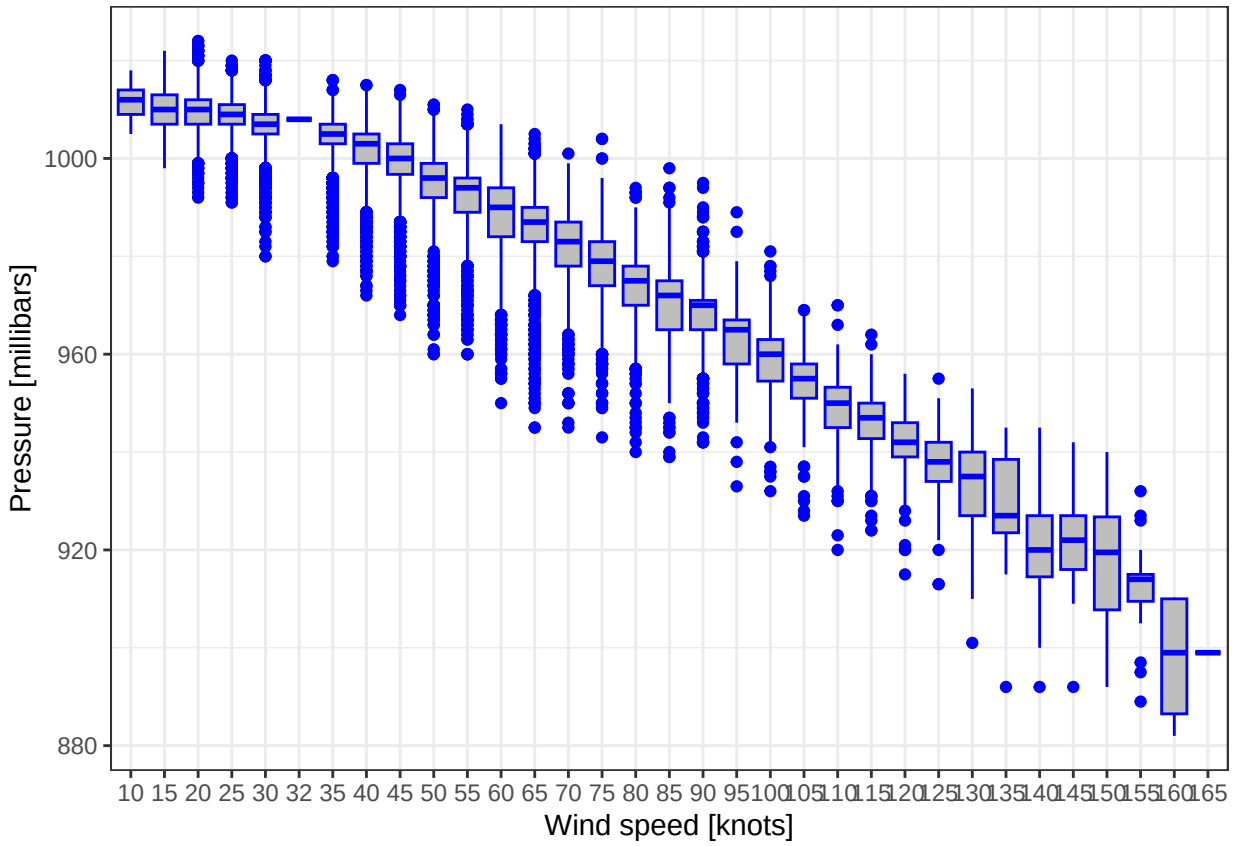
Based on the `storms` tibble create a box plot:

- The vertical axis should represent `pressure`.
- The horizontal axis: in `aes(...)` instead of `wind` use `factor(wind)` (to make `wind` a categorical variable).
- Use `gray` box fill and `blue` colour.
- Adjust the vertical title to `"Pressure [millibars]"` and horizontal to `"Wind speed [knots]"`.
- Use the black/white theme.

```
# ggplot( ... ) + ...
```

```
# 1p geom_boxplot is created
# 1p the vertical axis is correct
# 1p the horizontal axis is correct
# 1p both titles are correct
# 1p the theme is correct
# 1p the colour is set
# 1p the fill is set

ggplot( storms ) +
  aes( x = factor( wind ), y = pressure ) +
  geom_boxplot( fill = "gray", color = "blue" ) +
  labs( y = "Pressure [millibars]", x = "Wind speed [knots]" ) +
  theme_bw()
```



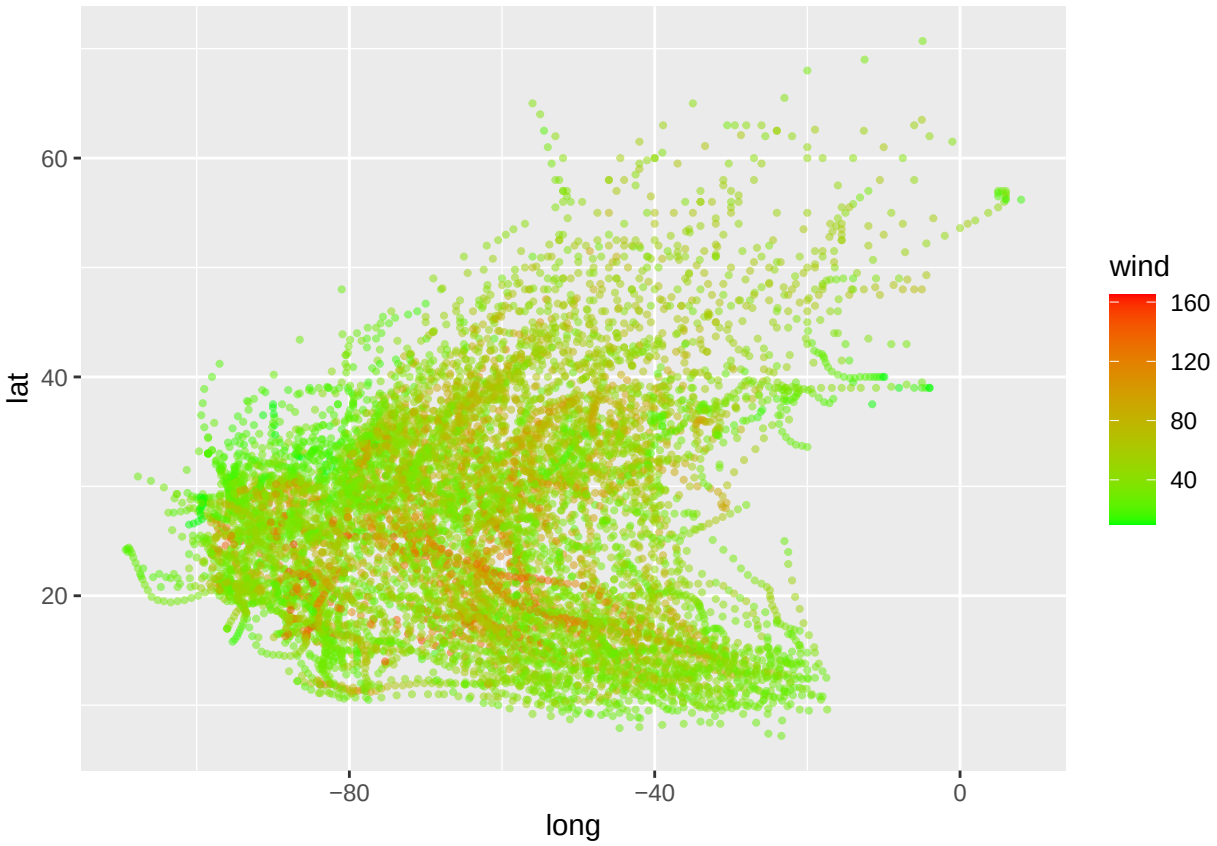
Question 9: [8p] Scatter plot

For this scatter plot take from `storms` only the rows with a missing `tropicalstorm_force_diameter` (or `ts_diameter`) value. Use `long` for the horizontal axis and `lat` for the vertical. Use transparency level of 0.5 and point size of 0.75. Colour points according to `wind`. Finally, use the colour scale with `green` for low and `red` for high `wind` values.

```
# ggplot( ... ) + ...
```

```
# 1p geom_point is used
# 1p rows with missing ts_diameter are selected
# 1p the horizontal axis is correct
# 1p the vertical axis is correct
# 1p the transparency is set
# 1p the point size is set
# 1p wind is used for colour in aes
# 1p the colour scale is set

filteredStorms <- storms %>% filter( is.na( tropicalstorm_force_diameter ) )
ggplot( filteredStorms ) +
  aes( x = long, y = lat, color = wind ) +
  geom_point( alpha = 0.5, size = 0.75 ) +
  scale_color_gradient( low = "green", high = "red" )
```


lat
60
40
20
−80
−40
0
long
wind
160
120
80
40