

# Exploratory Data Analysis with dplyr

## Functions

- pull()
- select()
- unique()
- distinct()
- n\_distinct()
- sort()
- filter()
- count()
- group\_by()
- summarise() (or summarize())
- arrange()
- disc()
- summary()

```
library("dplyr")
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

dplyr package has a sample dataframe “storms”.

```
storms
```

```
## # A tibble: 10,010 x 13
##   name  year month  day hour  lat  long status category  wind pressure
##   <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>  <ord>    <int>    <int>
## 1 Amy   1975     6   27    0  27.5 -79  tropi~ -1        25     1013
## 2 Amy   1975     6   27    6  28.5 -79  tropi~ -1        25     1013
## 3 Amy   1975     6   27   12  29.5 -79  tropi~ -1        25     1013
## 4 Amy   1975     6   27   18  30.5 -79  tropi~ -1        25     1013
## 5 Amy   1975     6   28    0  31.5 -78.8 tropi~ -1        25     1012
## 6 Amy   1975     6   28    6  32.4 -78.7 tropi~ -1        25     1012
## 7 Amy   1975     6   28   12  33.3 -78   tropi~ -1        25     1011
```

```
## 8 Amy 1975 6 28 18 34 -77 tropi~ -1 30 1006
## 9 Amy 1975 6 29 0 34.4 -75.8 tropi~ 0 35 1004
## 10 Amy 1975 6 29 6 34 -74.8 tropi~ 0 40 1002
## # ... with 10,000 more rows, and 2 more variables: ts_diameter <dbl>,
## # hu_diameter <dbl>
```

We can check a documentation about “storms”.

```
?storms # help(storms) works too
```

## Exploratory Data Analysis

To get to know the data. First step of analysis.

1. Check data structure: first and last 5 rows, summarize statistics (mean, sd, etc), number of column and row...
2. Clean the data: deleting some rows, filtering, sorting, deleting or replacing missing values
3. Visualize the data: histogram, scatter plot

`pull()` extracts column as a vector.

```
pull(storms, year)
```

```
## [1] 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975
## [15] 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975
## [29] 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975
## [43] 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975
## [57] 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975
## [71] 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975 1975
## [85] 1975 1975 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976
## [99] 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976
## [113] 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976
## [127] 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976 1976 1977 1977
## [141] 1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 1977
## [155] 1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 1977
## [169] 1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 1977
## [183] 1977 1977 1977 1977 1977 1977 1977 1977 1977 1977 1978 1978 1978 1978 1978
## [197] 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978
## [211] 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978
## [225] 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978
## [239] 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978 1978
## [253] 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979
## [267] 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979
## [281] 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979
## [295] 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979
## [309] 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979
## [323] 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979
## [337] 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979
## [351] 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979
## [365] 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979
## [379] 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979
## [393] 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979
## [407] 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979
## [421] 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979 1979
```

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

`select()` extracts column as a table.

```
## # A tibble: 10,010 x 1
##   year
##   <dbl>
## 1  1975
## 2  1975
## 3  1975
## 4  1975
## 5  1975
## 6  1975
## 7  1975
## 8  1975
## 9  1975
```

```
## 10 1975
## # ... with 10,000 more rows
```

unique() returns only unique values.

```
unique(pull(storms, year))
```

```
## [1] 1975 1976 1977 1978 1979 1980 1981 1982 1983 1984 1985 1986 1987 1988 1989
## [16] 1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004
## [31] 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015
```

```
unique(select(storms, year))
```

```
## # A tibble: 41 x 1
##   year
##   <dbl>
## 1 1975
## 2 1976
## 3 1977
## 4 1978
## 5 1979
## 6 1980
## 7 1981
## 8 1982
## 9 1983
## 10 1984
## # ... with 31 more rows
```

storms has records during a 40-year period from 1975 to 2015.

```
max(pull(storms, year)) - min(pull(storms, year))
```

```
## [1] 40
```

Inspect the column month.

```
unique(pull(storms, month))
```

```
## [1] 6 7 8 9 10 11 12 5 4 1
```

sort() sorts the order.

```
sort(unique(pull(storms, month))) # increasing order (default)
```

```
## [1] 1 4 5 6 7 8 9 10 11 12
```

```
sort(unique(pull(storms, month)), decreasing = TRUE) # decreasing order
```

```
## [1] 12 11 10 9 8 7 6 5 4 1
```

sort() takes only a vector, not a table.



```
# This returns an error.
# sort(unique(select(storms, month)))
```

`filter()` returns rows that meet a logical condition.

```
filter(storms, year == 1975) # extracts rows that year is 1975
```

```
## # A tibble: 86 x 13
##   name   year month   day hour   lat   long status category   wind pressure
##   <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>   <ord>     <int>     <int>
## 1 Amy   1975     6    27     0  27.5 -79   tropi~ -1         25     1013
## 2 Amy   1975     6    27     6  28.5 -79   tropi~ -1         25     1013
## 3 Amy   1975     6    27    12  29.5 -79   tropi~ -1         25     1013
## 4 Amy   1975     6    27    18  30.5 -79   tropi~ -1         25     1013
## 5 Amy   1975     6    28     0  31.5 -78.8 tropi~ -1         25     1012
## 6 Amy   1975     6    28     6  32.4 -78.7 tropi~ -1         25     1012
## 7 Amy   1975     6    28    12  33.3 -78   tropi~ -1         25     1011
## 8 Amy   1975     6    28    18   34   -77   tropi~ -1         30     1006
## 9 Amy   1975     6    29     0  34.4 -75.8 tropi~  0         35     1004
## 10 Amy  1975     6    29     6   34   -74.8 tropi~  0         40     1002
## # ... with 76 more rows, and 2 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>
```

Create an object that year is 1975.

```
storms75 <- filter(storms, year == 1975)
```

How many kinds of storms in 1975?

```
unique(pull(storms75, name)) # 3 storms recorded in 1975
```

```
## [1] "Amy"      "Caroline" "Doris"
```

Another way to find how many kinds of storms in 1975 is `distinct()`. It returns a tibble (table) instead of a vector.

```
distinct(storms75, name)
```

```
## # A tibble: 3 x 1
##   name
##   <chr>
## 1 Amy
## 2 Caroline
## 3 Doris
```

`distinct()` takes only a table, not a vector.

```
# This returns an error.
# distinct(pull(storms75, name))
```

`n_distinct()` returns a number of unique values.

```
n_distinct(select(storms75, wind))
```

```
## [1] 15
```

```
unique(select(storms75, wind))
```

```
## # A tibble: 15 x 1
##   wind
##   <int>
## 1    25
## 2    30
## 3    35
## 4    40
## 5    45
## 6    50
## 7    55
## 8    60
## 9    65
## 10   70
## 11  100
## 12   90
## 13   20
## 14   75
## 15   95
```

count() returns frequencies of each value.

```
count(storms75, name)
```

```
## # A tibble: 3 x 2
##   name      n
##   <chr> <int>
## 1 Amy      30
## 2 Caroline 33
## 3 Doris    23
```

group\_by() groups values. It is often used with summarise() to compute a summary on the specified column(s).

```
summarise(group_by(storms75, name), avg_wind = mean(wind), avg_pressure = mean(pressure))
```

```
## # A tibble: 3 x 3
##   name      avg_wind avg_pressure
##   <chr>      <dbl>      <dbl>
## 1 Amy         46.5         995.
## 2 Caroline    38.9        1002.
## 3 Doris       73.7         983.
```

summarize() returns the same result , same usage (“summarise” and “summarize” are the synonyms).

```
summarize(group_by(storms75, name), avg_wind = mean(wind), avg_pressure = mean(pressure))
```

```
## # A tibble: 3 x 3
##   name      avg_wind avg_pressure
##   <chr>      <dbl>      <dbl>
## 1 Amy        46.5        995.
## 2 Caroline   38.9       1002.
## 3 Doris     73.7        983.
```

Store the output.

```
avg_wind_pressure <- summarise(group_by(storms75, name), avg_wind = mean(wind), avg_pressure = mean(pressure))
avg_wind_pressure # ordered by name alphabetically
```

```
## # A tibble: 3 x 3
##   name      avg_wind avg_pressure
##   <chr>      <dbl>      <dbl>
## 1 Amy        46.5        995.
## 2 Caroline   38.9       1002.
## 3 Doris     73.7        983.
```

arrange() orders rows and returns an ordered table.

```
arrange(avg_wind_pressure, avg_wind) # order by average wind in increasing order
```

```
## # A tibble: 3 x 3
##   name      avg_wind avg_pressure
##   <chr>      <dbl>      <dbl>
## 1 Caroline   38.9       1002.
## 2 Amy        46.5        995.
## 3 Doris     73.7        983.
```

desc() orders decreasingly.

```
arrange(avg_wind_pressure, desc(avg_wind))
```

```
## # A tibble: 3 x 3
##   name      avg_wind avg_pressure
##   <chr>      <dbl>      <dbl>
## 1 Doris     73.7        983.
## 2 Amy        46.5        995.
## 3 Caroline   38.9       1002.
```

The difference between `sort()` and `arrange()` is that `sort()` takes a vector and returns a vector, while `arrange()` takes a table and returns a table.

Focus on Amy in 1975.

```
amy75 <- filter(storms75, name == "Amy")
amy75
```

```
## # A tibble: 30 x 13
##   name   year month   day hour   lat   long status category  wind pressure
##   <chr> <dbl> <dbl> <int> <dbl> <dbl> <dbl> <chr>   <ord>    <int>    <int>
## 1 Amy    1975     6    27     0  27.5 -79   tropi~ -1        25    1013
## 2 Amy    1975     6    27     6  28.5 -79   tropi~ -1        25    1013
## 3 Amy    1975     6    27    12  29.5 -79   tropi~ -1        25    1013
## 4 Amy    1975     6    27    18  30.5 -79   tropi~ -1        25    1013
## 5 Amy    1975     6    28     0  31.5 -78.8 tropi~ -1        25    1012
## 6 Amy    1975     6    28     6  32.4 -78.7 tropi~ -1        25    1012
## 7 Amy    1975     6    28    12  33.3 -78   tropi~ -1        25    1011
## 8 Amy    1975     6    28    18   34   -77   tropi~ -1        30    1006
## 9 Amy    1975     6    29     0  34.4 -75.8 tropi~  0        35    1004
## 10 Amy   1975     6    29     6   34   -74.8 tropi~  0        40    1002
## # ... with 20 more rows, and 2 more variables: ts_diameter <dbl>,
## #   hu_diameter <dbl>
```

Inspect amy75.

```
distinct(amy75, status)
```

```
## # A tibble: 2 x 1
##   status
##   <chr>
## 1 tropical depression
## 2 tropical storm
```

```
distinct(amy75, month)
```

```
## # A tibble: 2 x 1
##   month
##   <dbl>
## 1     6
## 2     7
```

```
count(distinct(amy75, day)) # How many days was Amy active?
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1     8
```

summary() shows the statistical summary.

```
summary(amy75) # Full stats
```

```
##      name          year      month      day
## Length:30      Min.   :1975      Min.   :6.000      Min.   : 1.00
## Class :character 1st Qu.:1975      1st Qu.:6.000      1st Qu.: 2.25
## Mode  :character Median :1975      Median :6.000      Median :27.00
##                      Mean  :1975      Mean  :6.467      Mean  :16.27
##                      3rd Qu.:1975      3rd Qu.:7.000      3rd Qu.:28.75
##                      Max.   :1975      Max.   :7.000      Max.   :30.00
##
##      hour      lat      long      status      category
## Min.   : 0.0      Min.   :27.50      Min.   : -79.00      Length:30      -1: 8
## 1st Qu.: 1.5      1st Qu.:33.80      1st Qu.: -76.70      Class :character 0 :22
## Median : 6.0      Median :36.05      Median : -70.35      Mode  :character 1 : 0
## Mean   : 8.6      Mean   :35.42      Mean   : -69.78      2 : 0
## 3rd Qu.:12.0      3rd Qu.:37.30      3rd Qu.: -65.30      3 : 0
## Max.   :18.0      Max.   :44.50      Max.   : -51.60      4 : 0
##                                     5 : 0
##      wind      pressure      ts_diameter      hu_diameter
## Min.   :25.00      Min.   : 981.0      Min.   : NA      Min.   : NA
## 1st Qu.:31.25      1st Qu.: 986.0      1st Qu.: NA      1st Qu.: NA
## Median :50.00      Median : 987.0      Median : NA      Median : NA
## Mean   :46.50      Mean   : 995.1      Mean   :NaN      Mean   :NaN
## 3rd Qu.:60.00      3rd Qu.:1005.5      3rd Qu.: NA      3rd Qu.: NA
## Max.   :60.00      Max.   :1013.0      Max.   : NA      Max.   : NA
##                                     NA's   :30      NA's   :30
```

```
summary(select(amy75, wind)) # stats only for wind
```

```
##      wind
## Min.   :25.00
## 1st Qu.:31.25
## Median :50.00
## Mean   :46.50
## 3rd Qu.:60.00
## Max.   :60.00
```

## Exercises

- 1) Use “dplyr” functions/commands to create a table (e.g. tibble) `storm_names_1980s` containing the name and year of storms recorded during the 1980s (i.e. from 1980 to 1989).

```
storm_names_1980s <- distinct(filter(storms, year >= 1980 & year <= 1989), name, year)
storm_names_1980s
```

```
## # A tibble: 70 x 2
##   name      year
##   <chr>    <dbl>
## 1 Bonnie  1980
## 2 Charley 1980
## 3 Georges 1980
## 4 Danielle 1980
## 5 Hermine 1980
## 6 Ivan    1980
```

```
## 7 Jeanne      1980
## 8 Karl        1980
## 9 Emily       1981
## 10 Floyd      1981
## # ... with 60 more rows
```

- 2) Use “dplyr” functions/commands to create a table (e.g. tibble) **storms\_per\_year** containing the number of storms recorded in each year (i.e. counts or frequencies of storms in each year). This table should contain two columns: **year** values in the first column, and number of storms in the second column.

```
storms_per_year <- count(distinct(storms, name, year), year)
storms_per_year
```

```
## # A tibble: 41 x 2
##   year      n
##   <dbl> <int>
## 1  1975      3
## 2  1976      2
## 3  1977      3
## 4  1978      4
## 5  1979      7
## 6  1980      8
## 7  1981      5
## 8  1982      5
## 9  1983      4
## 10 1984     10
## # ... with 31 more rows
```

- 3) Use “dplyr” functions/commands to create a table (e.g. tibble) **storm\_records\_per\_year** containing three columns: 1) **name** of storm, 2) **year** of storm, and 3) **count** for number of records (of the corresponding storm).

```
storm_records_per_year <- arrange(rename(count(storms, name, year), count = n), year)
storm_records_per_year
```

```
## # A tibble: 426 x 3
##   name      year count
##   <chr>    <dbl> <int>
## 1 Amy      1975     30
## 2 Caroline 1975     33
## 3 Doris     1975     23
## 4 Belle     1976     18
## 5 Gloria    1976     34
## 6 Anita     1977     20
## 7 Clara     1977     24
## 8 Evelyn    1977      9
## 9 Amelia    1978      6
## 10 Bess      1978     13
## # ... with 416 more rows
```

- 4) Use “dplyr” functions/commands to display the different (unique) types of storm **status**.

```
# they are all the same (but the last one returns a vector)
distinct(storms, status)
```

```
## # A tibble: 3 x 1
##   status
##   <chr>
## 1 tropical depression
## 2 tropical storm
## 3 hurricane
```

```
unique(select(storms, status))
```

```
## # A tibble: 3 x 1
##   status
##   <chr>
## 1 tropical depression
## 2 tropical storm
## 3 hurricane
```

```
unique(pull(storms, status))
```

```
## [1] "tropical depression" "tropical storm"      "hurricane"
```

5) Use “dplyr” functions/commands to display the different types of storm categories.

```
distinct(storms, category)
```

```
## # A tibble: 7 x 1
##   category
##   <ord>
## 1 -1
## 2 0
## 3 1
## 4 3
## 5 2
## 6 5
## 7 4
```

6) Use “dplyr” functions/commands to create a table (e.g. tibble) `storms_categ5` containing the `name` and `year` of those storms of category 5.

```
storms_categ5 <- select(filter(storms, category == 5), name, year)
storms_categ5
```

```
## # A tibble: 68 x 2
##   name      year
##   <chr>   <dbl>
## 1 Anita    1977
## 2 Anita    1977
```

```
## 3 David 1979
## 4 David 1979
## 5 David 1979
## 6 David 1979
## 7 David 1979
## 8 David 1979
## 9 David 1979
## 10 Gilbert 1988
## # ... with 58 more rows
```

- 7) Use “dplyr” functions/commands to display a table showing the `status`, `avg_pressure` (average pressure), and `avg_wind` (average wind speed), for each type of storm category. This table should contain four columns: 1) `category`, 2) `status`, 3) `avg_pressure`, and 4) `avg_wind`.

```
summarise(group_by(storms, category, status), avg_pressure = mean(pressure), avg_wind = mean(wind))
```

```
## # A tibble: 8 x 4
## # Groups:   category [7]
##   category status      avg_pressure avg_wind
##   <ord>    <chr>          <dbl>    <dbl>
## 1 -1      tropical depression 1008.    27.3
## 2 0       tropical storm    999.    45.8
## 3 1       hurricane      982.    70.9
## 4 1       tropical storm    975     70
## 5 2       hurricane      967.   89.4
## 6 3       hurricane      954.  105.
## 7 4       hurricane      940.  122.
## 8 5       hurricane      916.  145.
```

- 8) Use “dplyr” functions/commands to create a table (e.g. tibble) `max_wind_per_storm` containing three columns: 1) year of storm, 2) name of storm, and 3) `max_wind` maximum wind speed record (for that storm).

```
max_wind_per_storm <- summarise(group_by(storms, year, name), max_wind = max(wind))
max_wind_per_storm
```

```
## # A tibble: 426 x 3
## # Groups:   year [41]
##   year name      max_wind
##   <dbl> <chr>    <int>
## 1 1975 Amy        60
## 2 1975 Caroline 100
## 3 1975 Doris    95
## 4 1976 Belle   105
## 5 1976 Gloria   90
## 6 1977 Anita   150
## 7 1977 Clara    65
## 8 1977 Evelyn   70
## 9 1978 Amelia   45
## 10 1978 Bess    45
## # ... with 416 more rows
```



- 9) Use “dplyr” functions/commands to create a table (e.g. tibble) `max_wind_per_year` containing three columns: 1) `year` of storm, 2) `name` of storm, and 3) wind maximum wind speed record (for that year). Arrange rows by wind speed in decreasing order.

```
max_wind_per_year <- arrange(summarise(group_by(storms, year),
                                          name = name[which.max(wind)],
                                          max_wind = max(wind)),
                             desc(max_wind))
max_wind_per_year
```

```
## # A tibble: 41 x 3
##   year name    max_wind
##   <dbl> <chr>    <int>
## 1  1988 Gilbert    160
## 2  2005 Wilma     160
## 3  1998 Mitch     155
## 4  1977 Anita     150
## 5  1979 David     150
## 6  1992 Andrew    150
## 7  2007 Dean      150
## 8  2003 Isabel    145
## 9  2004 Ivan      145
## 10 1989 Hugo       140
## # ... with 31 more rows
```