# Diving into DataFrames

DATA MANIPULATION IN JULIA

**Katerina Zahradova**
Instructor

datacamp

# Course outline

- Working with columns
- Grouping of data
- Summary statistics
- Pivot tables
- Loading and saving of CSV files
- Visualizations
- Writing readable and organized code

# Datasets

[1] Pexels

# Strings and symbols

```
# Using strings
df[:, "col 1"]


df[:, "col2"]
```

```
# Using symbols
df[:, Symbol("col 1")]


df[:, :col2]
```

# What is missing

```
# Using first()
println(first(penguins))
```

```
Row    species    island    culmen_l_mm ...
       String15   String15  String7?    ...
-----------------------------------------------
1      Adelie     Torgersen 39.1
```

```
# Using describe
describe(penguins)
```

```
7x7 DataFrame
Row  variable      ...   nmissing  ...
     Symbol        ...   Int64     ...
-----------------------------------------
1    species       ...   0         ...
2    island        ...   0         ...
3    culmen_l_mm   ...   10        ...
4    culmen_d_mm   ...   10        ...
5    flipper_l_mm  ...   10        ...
...
```

# Describe it better

```
# Describe
describe(penguins)
```

```
Row  variable     mean    min      ...
     Symbol       Nothing Union    ...
-------------------------------------------
1    species              Adelie
2    island               Biscoe
3    culmen_l_mm  32.1    34.7
4    culmen_d_mm  13.1    16
5    flipper_l_mm 205.4   165
...
```

```
# Describe using only some columns
describe(penguins, :nmissing, :eltype)
```

```
Row  variable     nmissing    eltype
     Symbol       Int64       DataType
-----------------------------------------------
1    species      0           String15
2    island       0           String15
3    culmen_l_mm  10          Float64
4    culmen_d_mm  10          Float64
5    flipper_l_mm 10          Float64
...
```

# Describe it how we like it

```
# Using sum

describe(penguins, sum => :total)
```

```
7×2 DataFrame
Row  variable     total
     Symbol       Union

-------------------------------------
1    species
2    island
3    culmen_l_mm  15136.6
4    culmen_d_mm  5163.4
...
```

# DataFrames syntax

column_to_transform ⇒ transformation ⇒ transformed_column

# DataFrames syntax

```
column_to_transform => transformation => transformed_column

[columns_to_transform] .=> [transformations] .=> [transformed_columns]
```

# Let's practice!

DATA MANIPULATION IN JULIA

# Selecting columns

DATA MANIPULATION IN JULIA

**Katerina Zahradova**
Instructor

datacamp

# US minimum wages

```
# Print wages
wages
```

```
2703×10 DataFrame 2678 rows omitted 5 columns omitted
Row  year   state     region   state_min_wage    state_min_wage_2020_dollars ...
     Int64  String31  String3  Float64           Float64                     ...
─────────────────────────────────────────────────────────────────────────────
1    1968   Alabama   S        0.0               0.0                         ...
2    1968   Alaska    W        2.1               15.61                       ...
...
```

# How we slice

```
# Using position of the column
wages[:, 1]
```

```
# Using name of the column
wages[:, "year"]
wages[:, :year]
```

```
# Using name of the column
wages.year
```

```
# Selecting several columns
wages[:, ["year", "state"]]
wages[:, [:year, :state]]
wages[:, [1,2]]
```

# Selecting columns

```
# Selecting the state, year, and state minimum wage
select(wages, 2, 1, 4)
```

```
2703×3 DataFrame 2678 rows omitted
Row  state     year        state_min_wage
     String31  Int64       Float64

-----------------------------------------
1    Alabama   1968        0.0
2    Alaska    1968        2.1
...
```

# Selecting columns

```
# Selecting the state, year, and state minimum wage
select(wages, "state", :year, 4)
```

```
2703×3 DataFrame 2678 rows
Row  state     year        state_min_wage
     String31  Int64       Float64

-----------------------------------------
1    Alabama   1968        0.0
2    Alaska    1968        2.1
...
```

# Selecting using patterns

Selecting columns

- Starting/ending with a letter/word
- Containing a sub-string
- ...

... gets bothersome in large datasets

# Selecting using patterns

```
# All columns starting with state
select(wages, Cols(startswith("state")))
```

```
2703×3 DataFrame 2702 rows omitted
Row state    state_min_wage state_min_wage_2020_dollars
    String31 Float64        Float64

---------------------------------------
1   Alabama  0.0            0.0
...
```

# Selecting using patterns

```
# All columns ending with 2020_dollars
select(wages, Cols(endswith("2020_dollars")))
```

```
2703x3 DataFrame 2702 rows omitted
Row state_min_wage_2020_dollars  federal_min_wage_2020_dollars  effective_min_wage_2020_dollars
    Float64                      Float64                       Float64
──────────────────────────────────────────────────────────────────────────────────────────────
1   0.0                          8.55                          8.55
...
```

# Selecting using patterns

```
# All columns containing min
select(wages, Cols(contains.("min")))
```

```
2703×6 DataFrame 2702 rows omitted 3 columns omitted
Row  state_min_wage  state_min_wage_2020_dollars  federal_min_wage  ...
     Float64         Float64                      Float64           ...
─────────────────────────────────────────────────────────────────────
1    0.0             0.0                          1.15              ...
...
```

# Regex

regex = **reg**ular **ex**pressions

- String of text
- Helps to match and locate (complicated) patterns in text
- Learn more on DataCamp Cheat Sheet, regex101.com and other websites

# Using regex

```
# Selecting using regex
select(wages, r"min")
```

```
2703×6 DataFrame 2702 rows omitted 3 columns omitted
Row  state_min_wage  state_min_wage_2020_dollars  federal_min_wage  ...
     Float64         Float64                      Float64           ...
─────────────────────────────────────────────────────────────────────
1    0.0             0.0                          1.15              ...
...
```

# select!() vs. select()

```
# Mutates original DataFrame
select!(wages, :year, :state)

# Original DataFrame changed
println(first(wages))
```

```
DataFrameRow (2 columns)
Row  year    state
     Int64   String31

---------------------
1    1968    Alabama
```

```
# Returns new DataFrame
select(wages, :year, :state)

# Original DataFrame is intact
println(first(wages))
```

```
DataFrameRow (10 columns, 7 omitted)
Row  year    state     region    ...
     Int64   String31  String3   ...

-------------------------------------
1    1968    Alabama   S         ...
```

# Let's practice!

DATA MANIPULATION IN JULIA

# Exploring Data with Visualizations

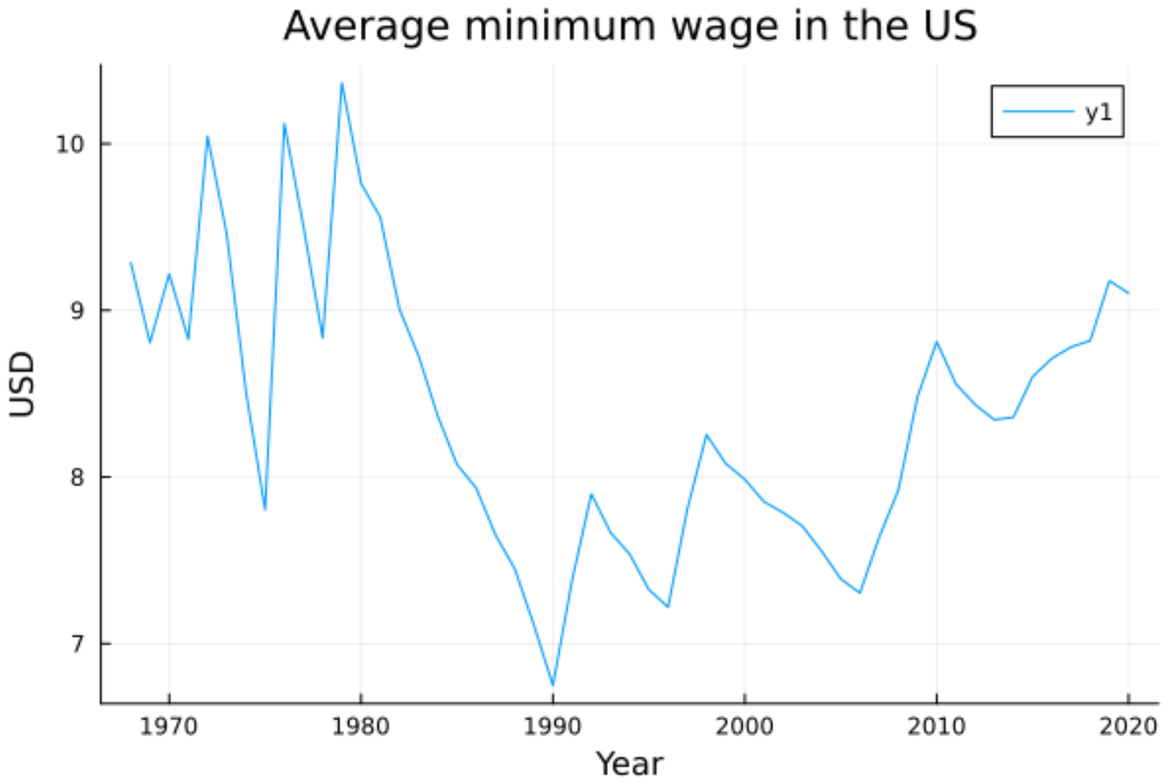
DATA MANIPULATION IN JULIA

**Katerina Zahradova**
Instructor

datacamp

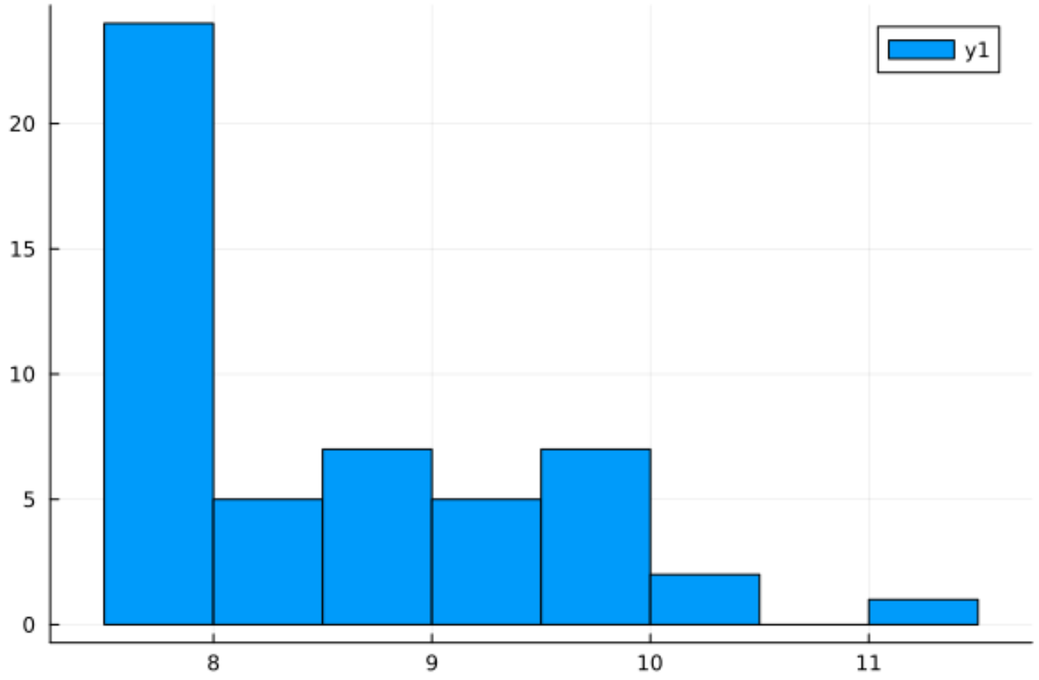
# Why we visualize?

```
Row  year   mean_min_wage_2020_dollars
     Int64  Float64

-------------------------------------
1    1968   9.28529
2    1969   8.80667
3    1970   9.21882
4    1971   8.82686
5    1972   10.0457
...
```

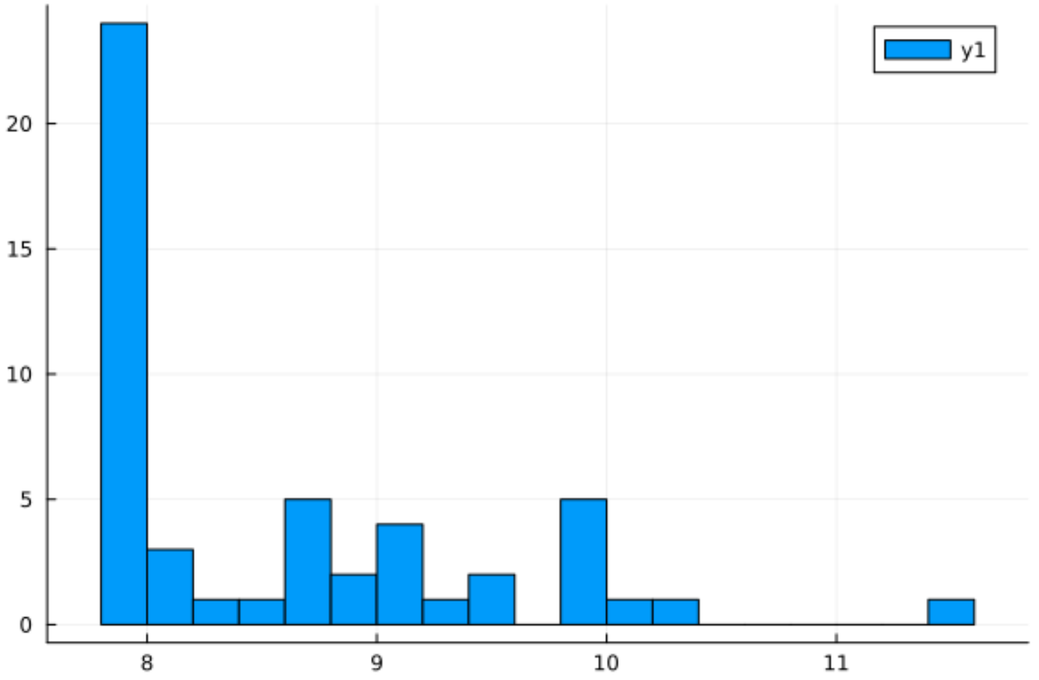


# Histogram

```
# Make a histogram with default bins
wages_2015 = filter(wages.year == 2015, wages)
histogram(wages_2015.eff_min_wage_2020_dollars)
```

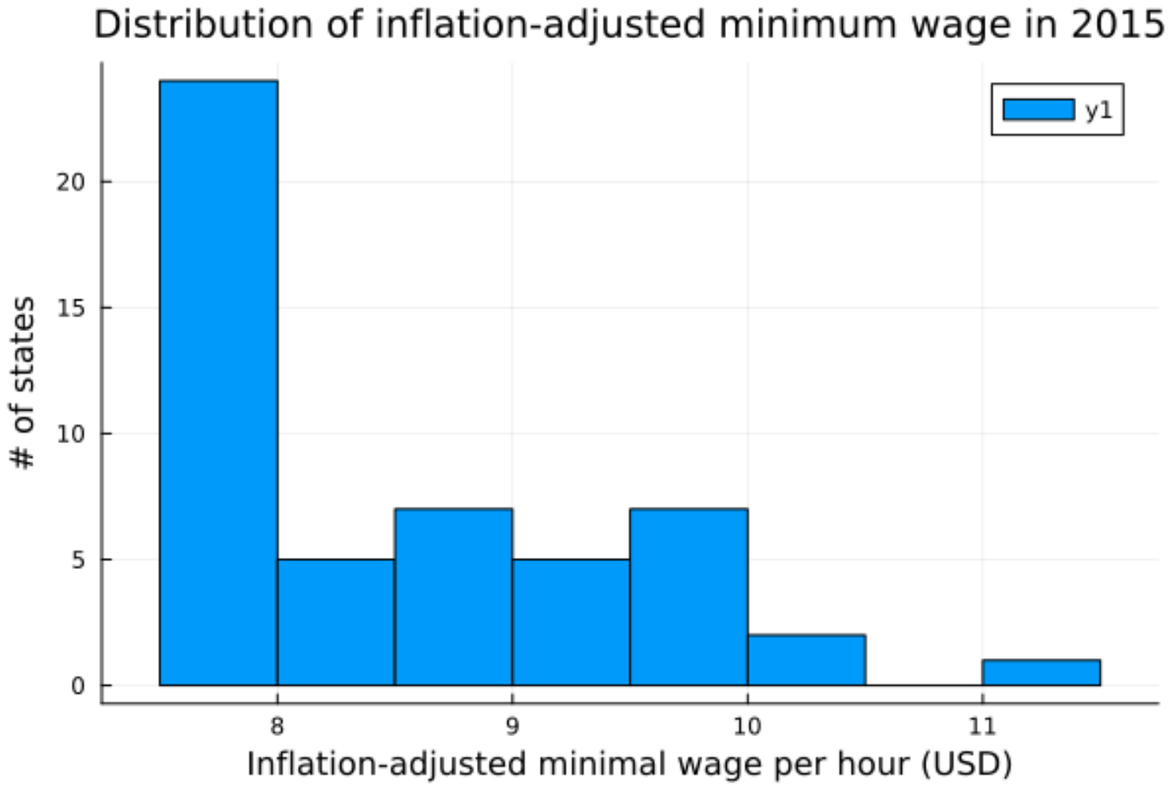



```
# Specifying the number of bins
wages_2015 = filter(wages.year == 2015, wages)
histogram(wages_2015.eff_min_wage_2020_dollars,
          bins = 25)
```



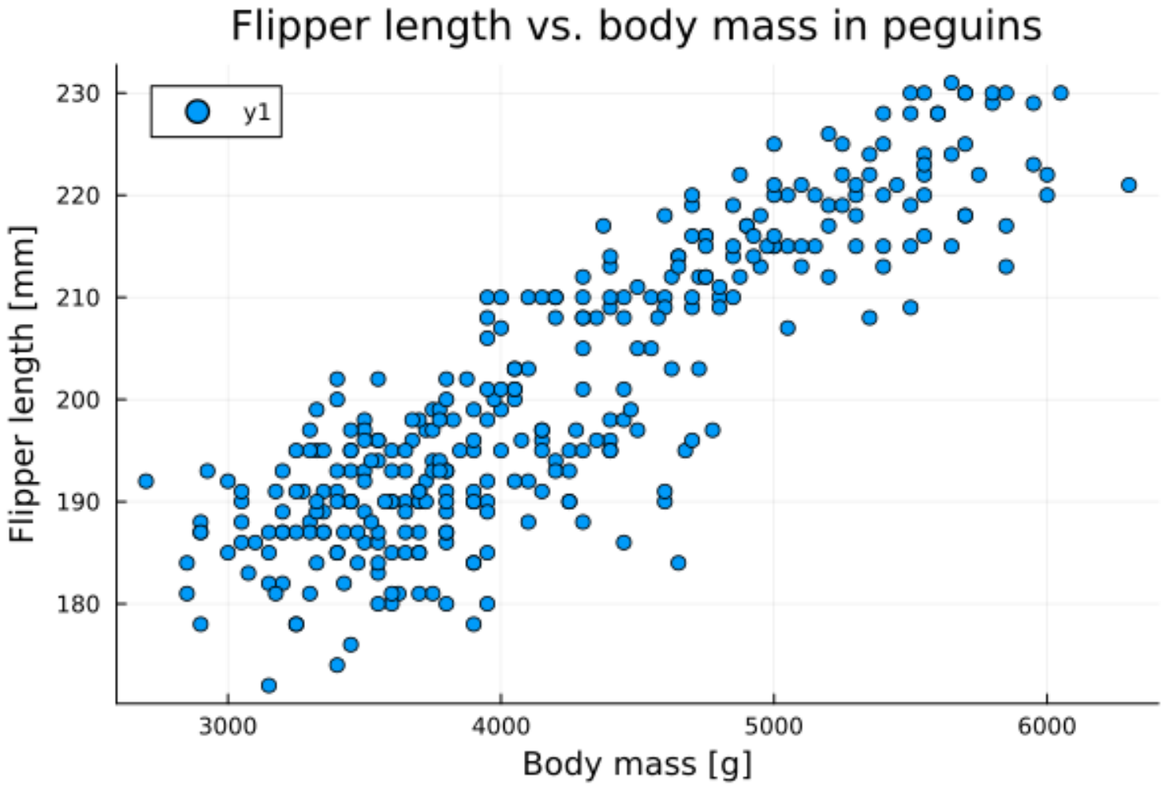
# Labeling our plot

```
# Make histogram
wages_2015 = filter(wages.year == 2015, wages)
histogram(wages_2015.eff_min_wage_2020_dollars)
# Include x label
xlabel!("Inflation-adjusted minimal
        wage per hour (USD)")
# Include y label
ylabel!("# of states")
# Make title
title!("Distribution of inflation-adjusted
        minimum wage in 2015")
```



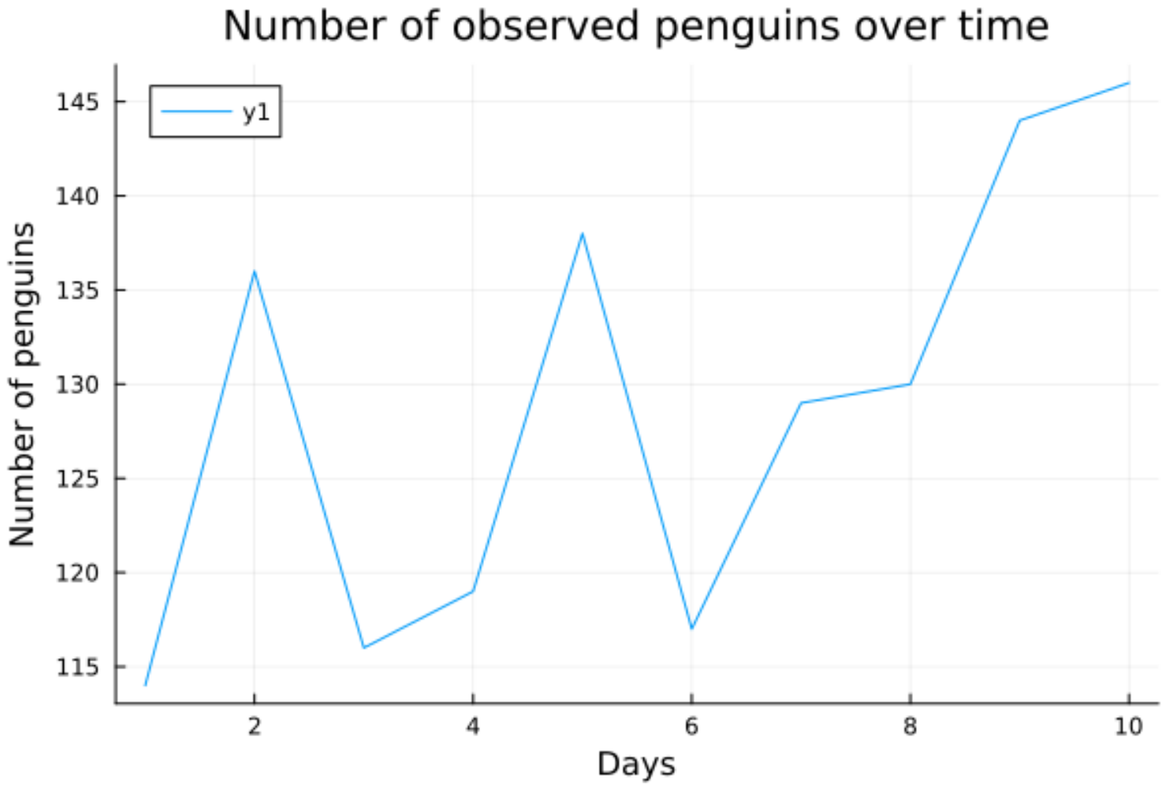
# Scatter plot

```
# Scatter plot
scatter(penguins.body_mass_g,
        penguins.flipper_length_mm)

# Labels
xlabel!("Body mass [g]")
ylabel!("Flipper length [mm]")
title!("Flipper length vs.
        body mass in peguins")
```



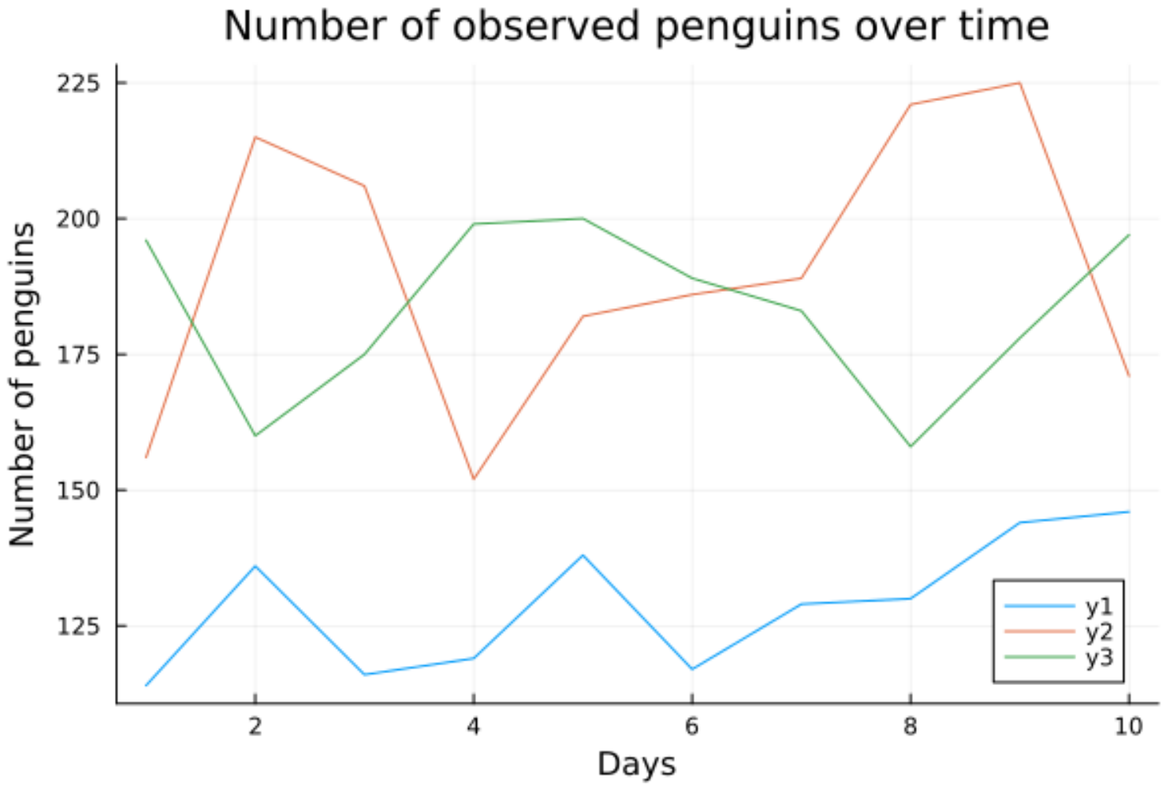
# Line plot

```
# Number of Adelie penguins over time
plot(observations.days,
    observations.adelie)

# Labels
xlabel!("Days")
ylabel!("Number of penguins")
title!("Number of observed
        penguins over time")
```



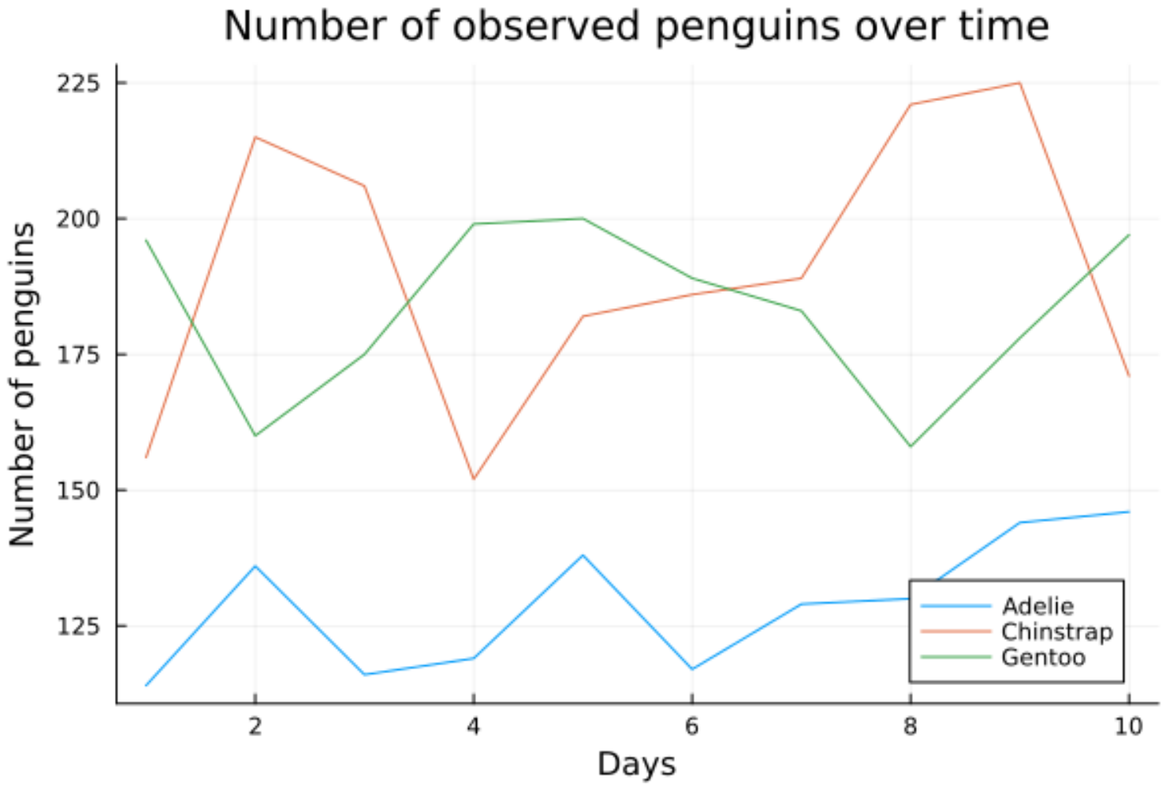
# Multiple lines

```
# Plot the first line
plot(observations.day, observations.adelie)

# Adding and modifying with new lines
plot!(observations.day, observations.chinstrap)
plot!(observations.day, observations.gentoo)

# Labels
xlabel!("Days")
ylabel!("Number of penguins")
title!("Number of observed penguins over time")
```



# Multiple lines with legend

```
# Make a plot
plot(observations.day, observatations.adelie,
    label = "Adelie" )
plot!(observations.day, observations.chinstrap,
    label = "Chinstrap")
plot!(observations.day, observations.gentoo,
    label = "Gentoo")

# Labels
xlabel!("Days")
ylabel!("Number of penguins")
title!("Number of observed penguins over time")
```



# Cheat sheet

Types of plots:

- **Histogram** - distribution of a numerical variable `histogram(df.n1, label = "n1")`
- **Scatter plot** - relationship of two numerical variables
  `scatter(df.x, df.y, label = "y")`
- **Line plot** - time evolution of a numerical variable `plot(df.x, df.y, label = "y")`

Adding another line to existing plot:

- `histogram!(df.n2, label = "n2")`
- `scatter!(df.x2, df.y2, label = "y2")`
- `plot!(df.x2, df.y2, label = "y2")`

Labels:

- `xlabel!("Text of your x label")`
- `ylabel!("Text of your y label")`
- `title!("Text of your title")`

# Let's practice!

DATA MANIPULATION IN JULIA