

Vectors plotting practice

DSC 105, Introduction to data science, Lyon College, Fall 2025

Readme

- Put your name at the top next to the pledge
- Answer all questions that you can answer.
- Don't use AI.
- Upload the URL to your completed datalab notebook to Canvas at the end of the session, no matter how far you've got.

Vectors plotting practice

This is a short 30-45 minute exercise to review plotting vectors in various formats - as histogram, barplot, line plot, or scatter plot.

1. Load the dataset uspop into your current R session, and list all user-defined objects of this session (the output should be "uspop" - if not, you should know what to do using `rm`).

```
data(uspop)
ls()
```

```
[1] "comics"   "crime"    "tab_cnt"   "uspop"
```

2. Check the structure of the uspop dataset.

```
str(uspop)
```

```
Time-Series [1:19] from 1790 to 1970: 3.93 5.31 7.24 9.64 12.9 ...
```

3. Print the values of the uspop dataset.

```
uspop
```

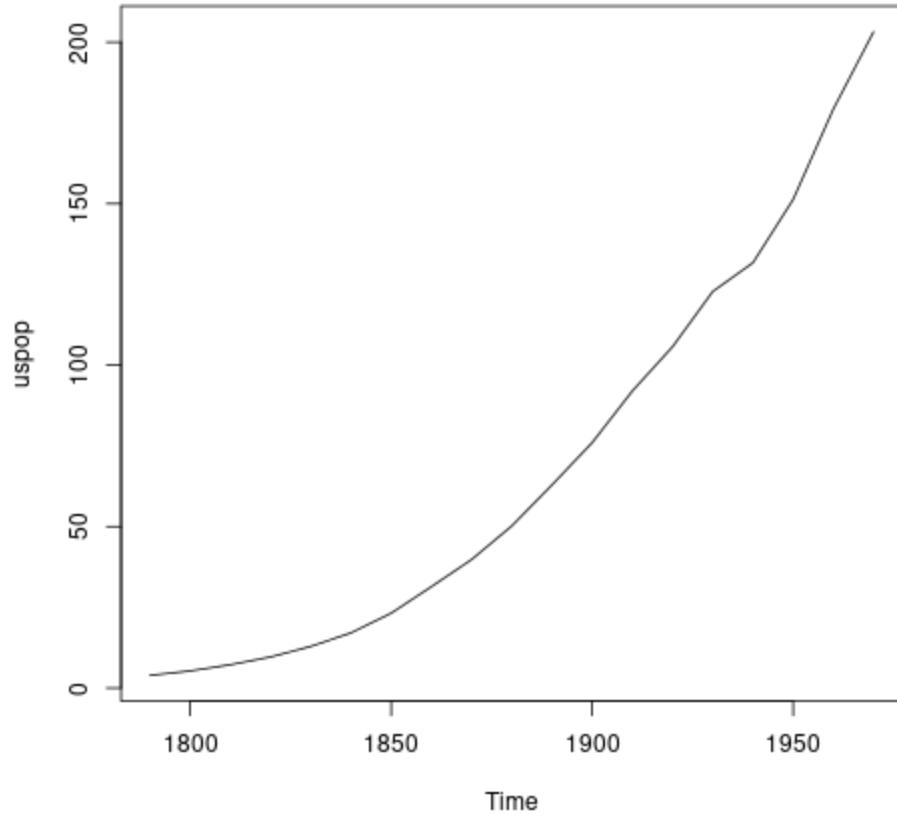
```
Time Series:
Start = 1790
End = 1970
Frequency = 0.1
[1] 3.93 5.31 7.24 9.64 12.90 17.10 23.20 31.40 39.80 50.20 62.90
[13] 92.00 105.70 122.80 131.70 151.30 179.30 203.20
```

4. Look up the help for uspop - save it to a file to look at it using `capture.output`.

```
help(uspop) |> capture.output(file="uspop.txt")
```

5. uspop is a time series just like Nile from class. Create a line plot for uspop.

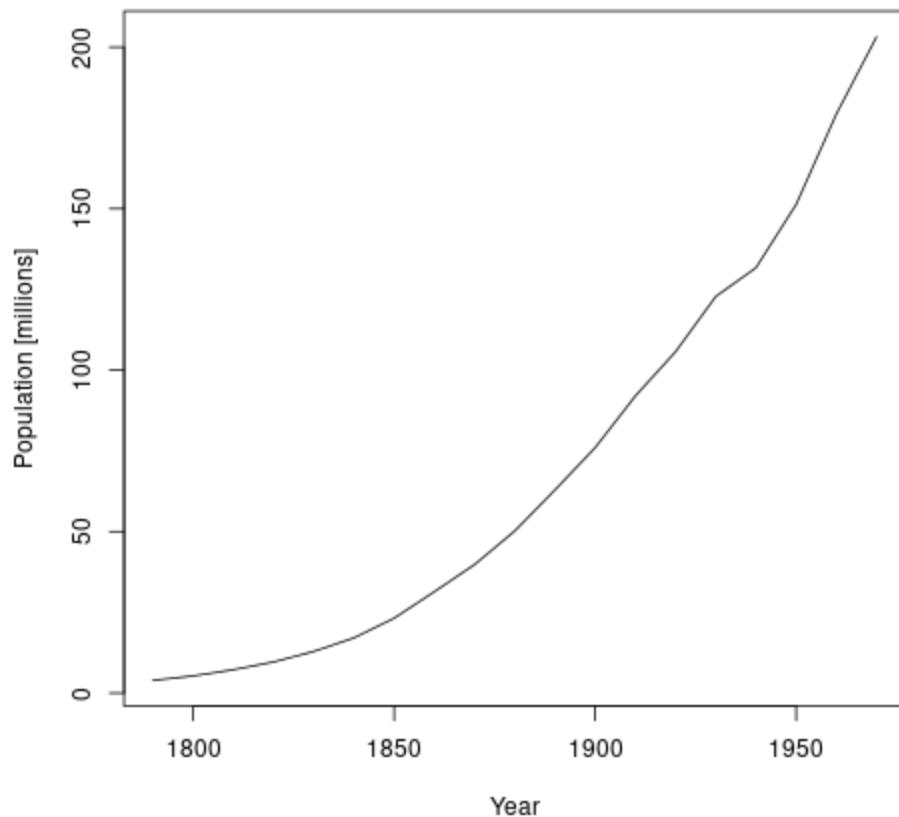
```
plot(uspop)
```



6. Replace the x- and y-axis labels and add a title: use "Year", and for the y-axis, use "Population [millions]". For the title, use "U.S. population 1790-1970 Decennical Census Data".

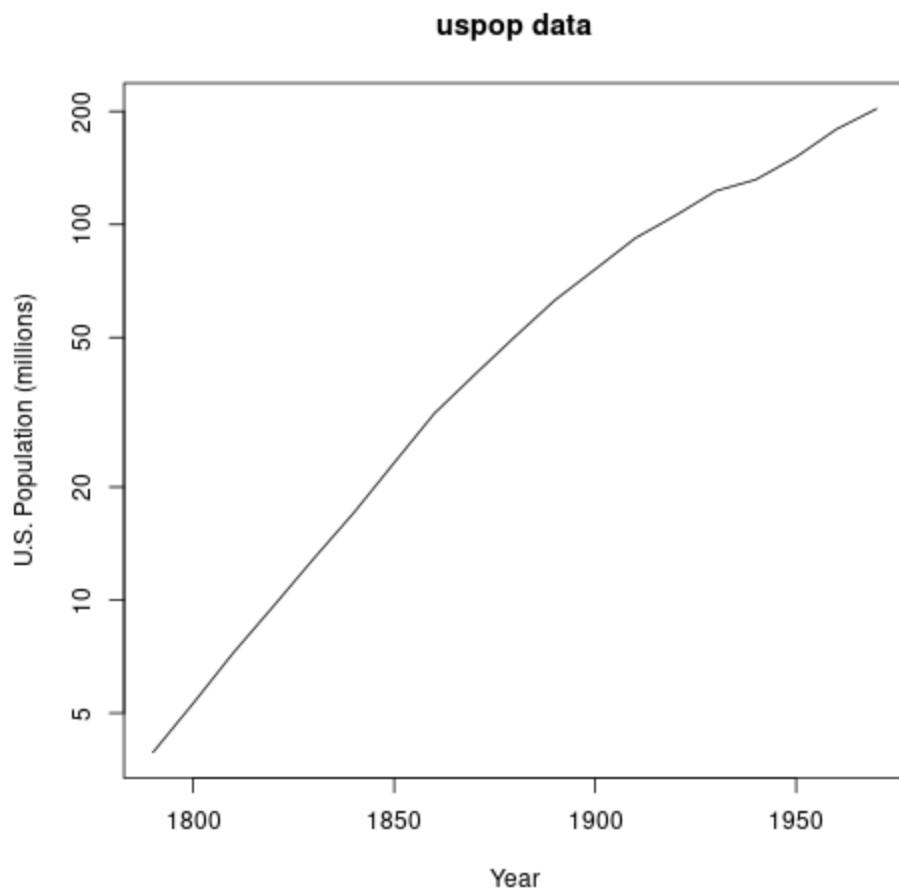
```
plot(uspop,
      xlab="Year",
      ylab="Population [millions]",
      main="U.S. population 1790-1970 Decennical Census Data")
```

U.S. population 1790-1970 Decennial Census Data



7. Run the `plot` command from the examples section of the help file for `uspop`.

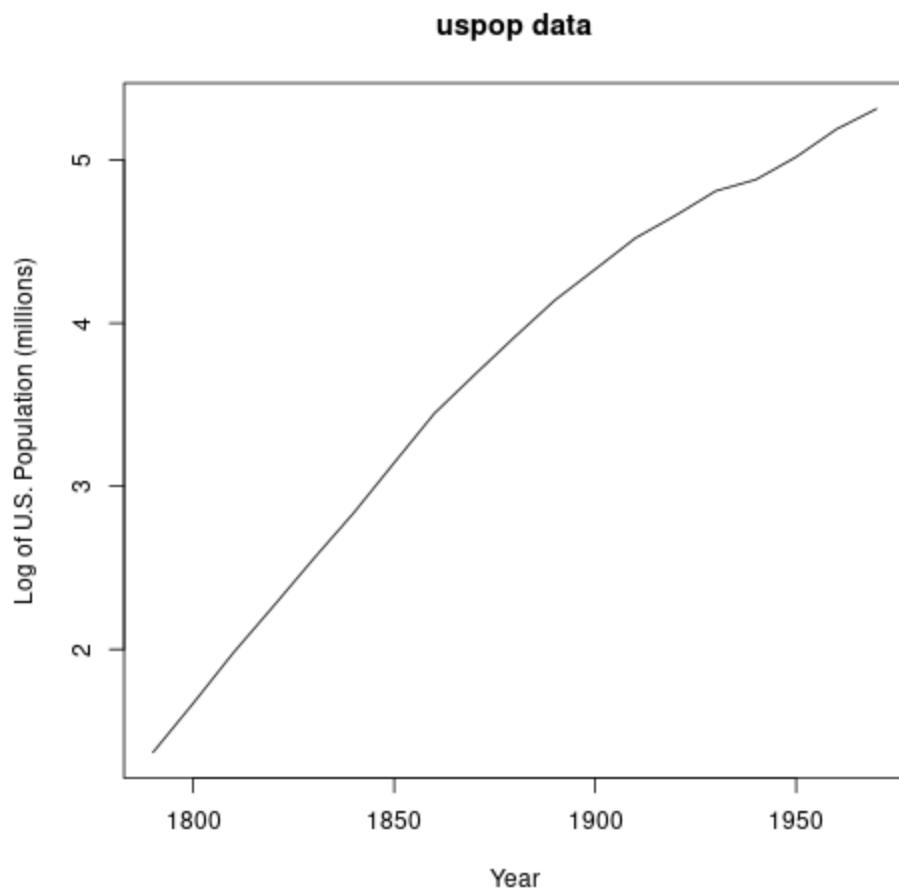
```
plot(uspop,
  log = "y",
  main = "uspop data",
  xlab = "Year",
  ylab = "U.S. Population (millions)")
```



8. Can you explain the difference between these two plots? Write your answer in the box below:

The `log` parameter is set to "y", which makes this a semi-logarithmic plot (in the y-axis). You can see that the y-axis ticks are not spaced evenly any more - looks linear but is in fact logarithmic.

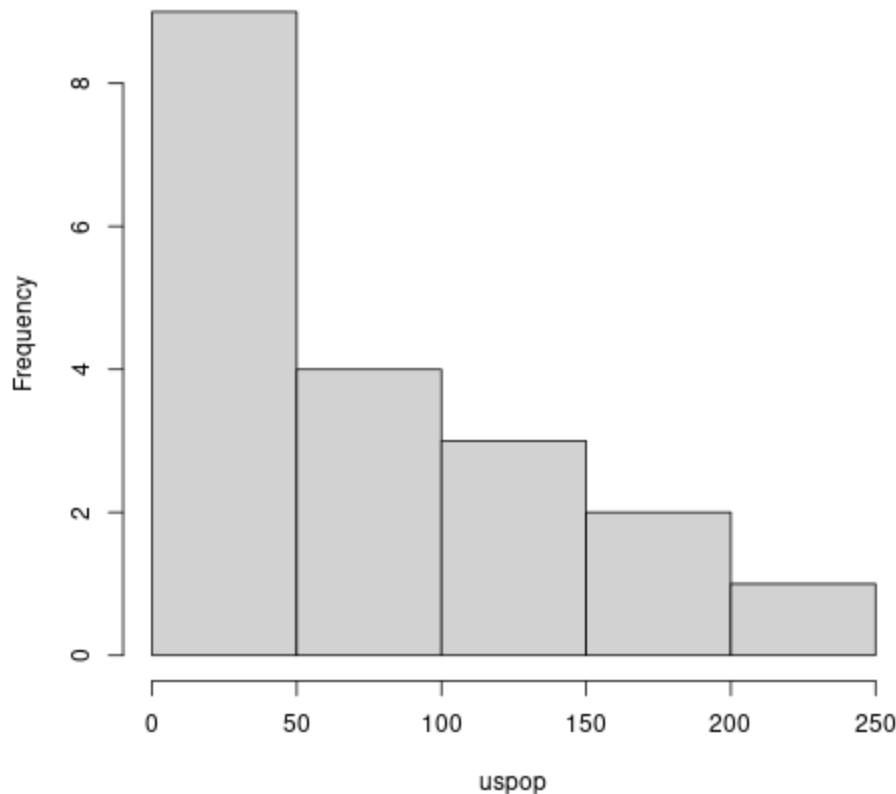
```
plot(log(uspop),  
     main = "uspop data",  
     xlab = "Year",  
     ylab = "Log of U.S. Population (millions)")
```



9. What would a histogram of this dataset look like? Create one (it's not very interesting because it doesn't show change over time). Change the title to "Histogram of U.S. population 1790-1970", and the x-axis label to "Population [millions]".

```
hist(uspop,  
      main="Histogram of U.S. population 1790-1970")
```

Histogram of U.S. population 1790-1970



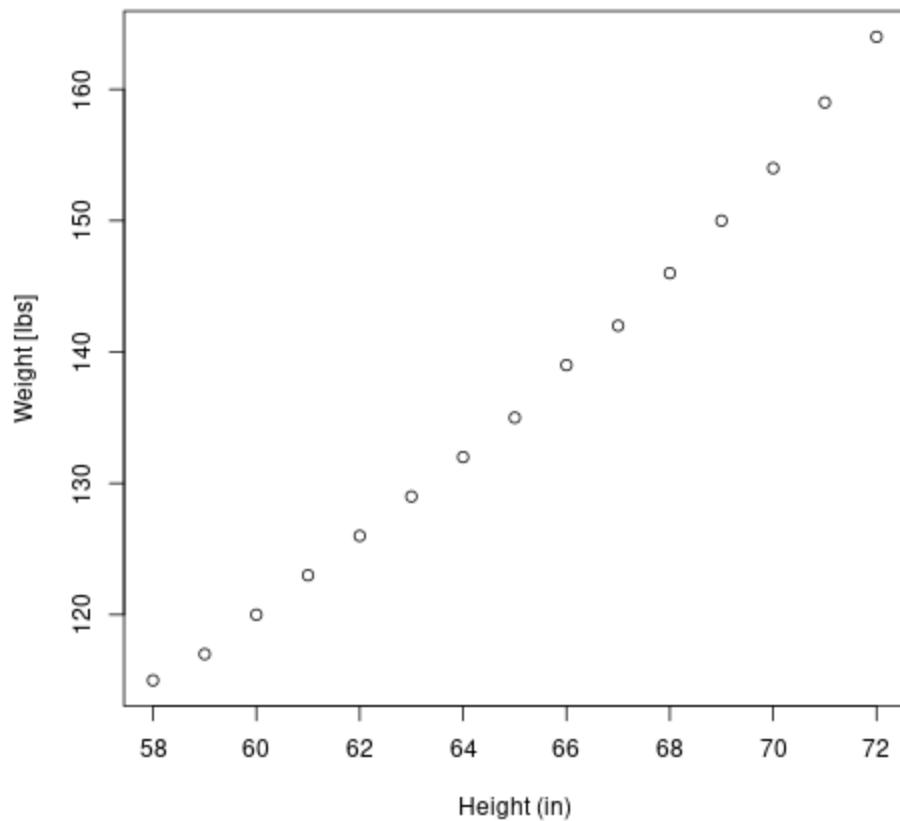
10. Load the built-in `women` dataset and display its structure:

```
data(women)
str(women)
```

```
'data.frame': 15 obs. of 2 variables:
$ height: num  58 59 60 61 62 63 64 65 66 67 ...
$ weight: num 115 117 120 123 126 129 132 135 139 142 ...
```

11. Create a scatterplot of the womens' weight as a function of height. Title the plot "Womens' height vs. weight", and rename the x- and y-axis labels to "Height (in)" and "Weight [lbs]", respectively.

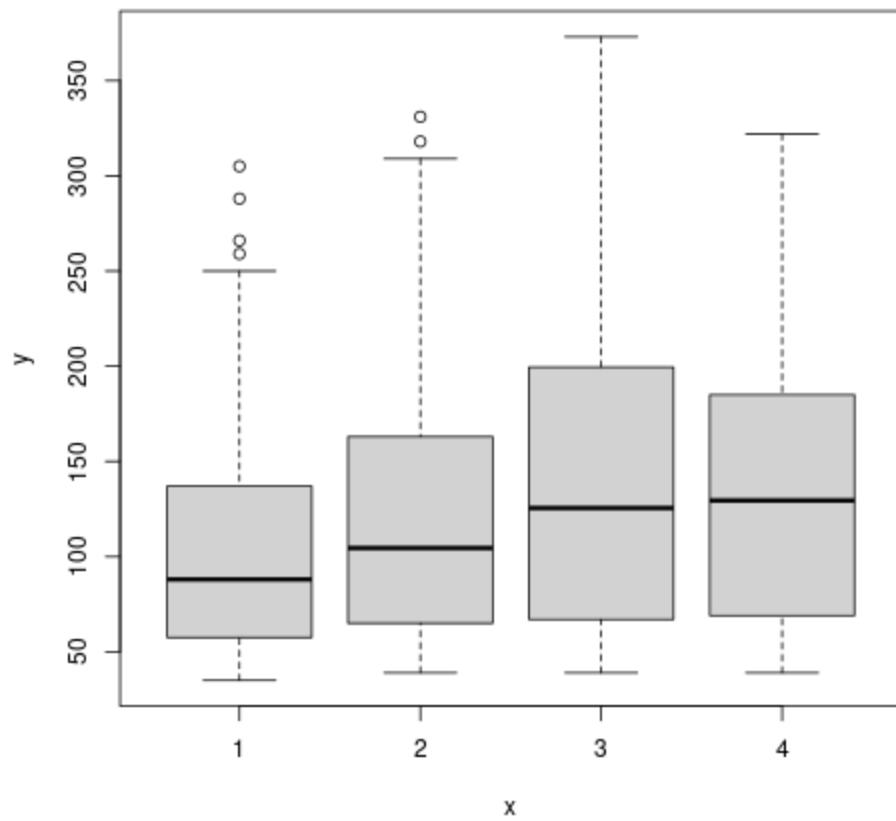
```
plot(x=women$height,
      y=women$weight,
      xlab="Height (in)",
      ylab="Weight [lbs]")
```



12. Load the `ChickWeight` dataset. It has a numeric vector `weight` and a factor named `Diet` for different diets of the chickens.

Plot the `weight` of the chickens as a function of the `Diet`:

```
data(ChickWeight)
plot(x=ChickWeight$Diet,
     y=ChickWeight$weight)
```

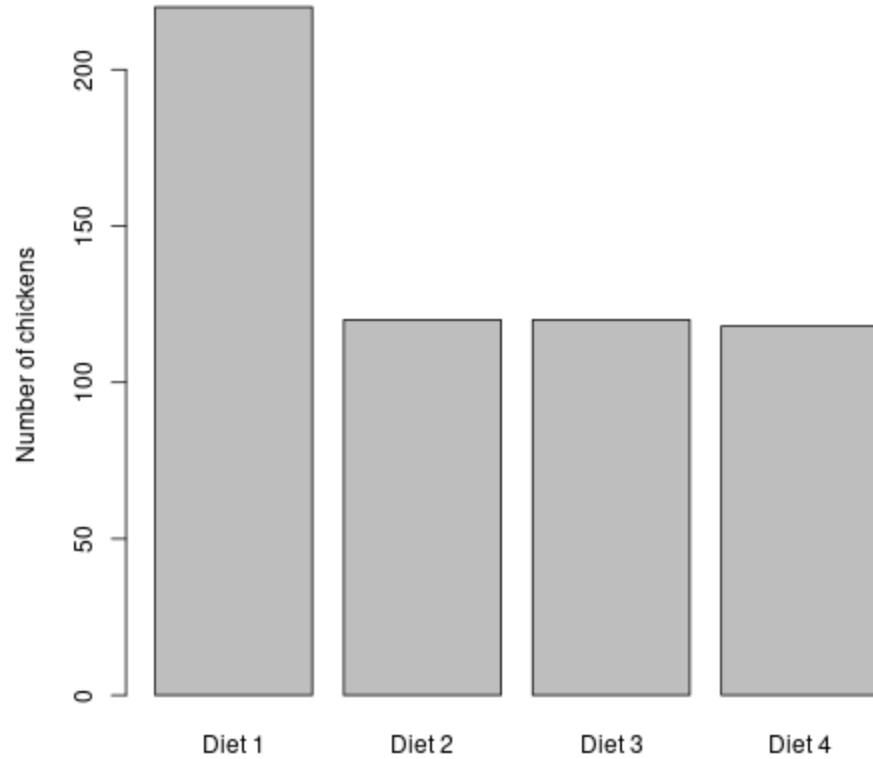


```
data(ChickWeight)
str(ChickWeight)
```

```
Classes 'nfnGroupedData', 'nfGroupedData', 'groupedData' and 'data.frame':      57
$ weight: num  42 51 59 64 76 93 106 125 149 171 ...
$ Time   : num  0 2 4 6 8 10 12 14 16 18 ...
$ Chick  : Ord.factor w/ 50 levels "18"<"16"<"15"<...: 15 15 15 15 15 15 15 15 15 15 ...
$ Diet    : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
- attr(*, "formula")=Class 'formula' language weight ~ Time | Chick
....- attr(*, ".Environment")=<environment: R_EmptyEnv>
- attr(*, "outer")=Class 'formula' language ~Diet
....- attr(*, ".Environment")=<environment: R_EmptyEnv>
- attr(*, "labels")=List of 2
..$ x: chr "Time"
..$ y: chr "Body weight"
- attr(*, "units")=List of 2
..$ x: chr "(days)"
..$ y: chr "(gm)"
```

13. Create a bar plot of the factor variable Diet of the ChickWeight dataset, title it "Chickens by diet type", change the factor level labels to "Diet 1", "Diet 2", "Diet 3" and "Diet 4", and add an y-axis label "Number of chickens".

```
levels(ChickWeight$Diet) <- c("Diet 1","Diet 2","Diet 3","Diet 4")
plot(ChickWeight$Diet,
     title="Chickens by diet type",
     ylab="Number of chickens")
```



Author: [your name] (pledged)

Created: 2025-11-13 Thu 20:47