

Vectors plotting practice

DSC 105, Introduction to data science, Lyon College, Fall 2024

Marcus Birkenkrahe (pledged)

October 17, 2024

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").

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

```
[1] "uspop" "women"
```

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 17.1 23.2 31.4 39.8
```

3. Print the values of the `uspop` dataset.

```
uspop
```

```
Time Series:
```

```
Start = 1790
```

```
End = 1970
```

```
Frequency = 0.1
```

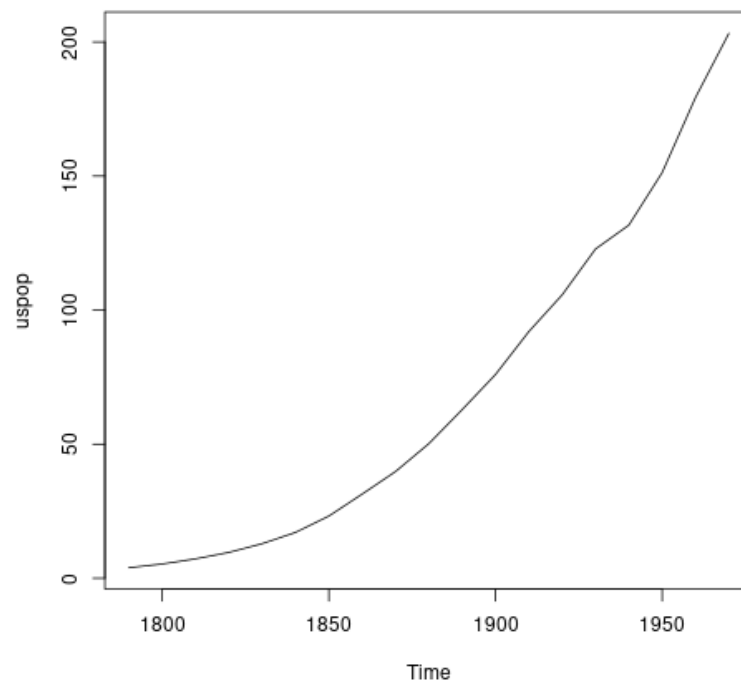
```
[1] 3.93 5.31 7.24 9.64 12.90 17.10 23.20 31.40 39.80 50.20
[11] 62.90 76.00 92.00 105.70 122.80 131.70 151.30 179.30 203.20
```

4. Look up the help for `uspop` - you need to switch to the R console to do this. Copy the description into this **example** block:

This data set gives the population of the United States (in millions) as recorded by the decennial census for the period 1790-1970.

5. `uspop` is a time series just like `Nile` from class. Create a line plot for `uspop`. Make sure you can open/close it with `<F6>/<F7>`.

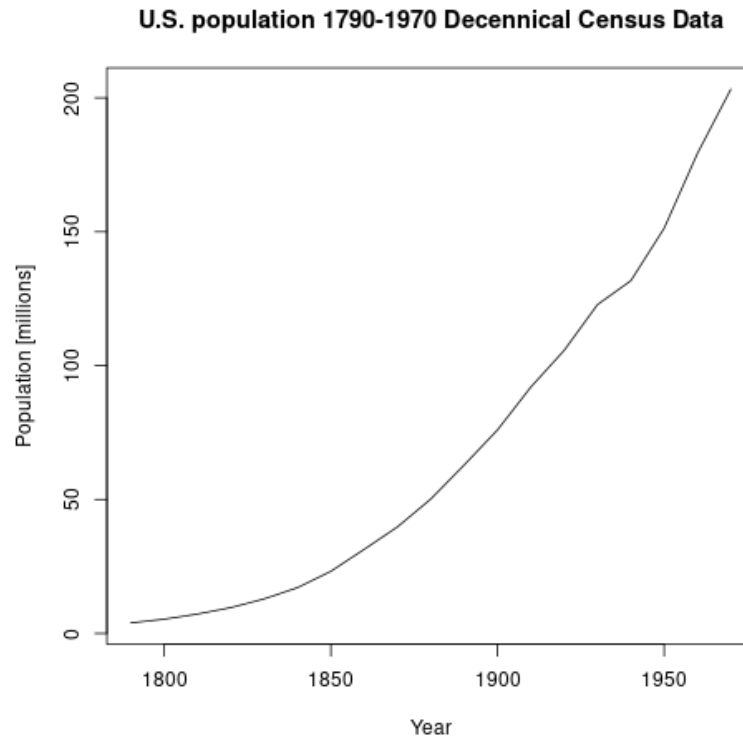
```
plot(uspop)
```



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 Decennial Census Data"

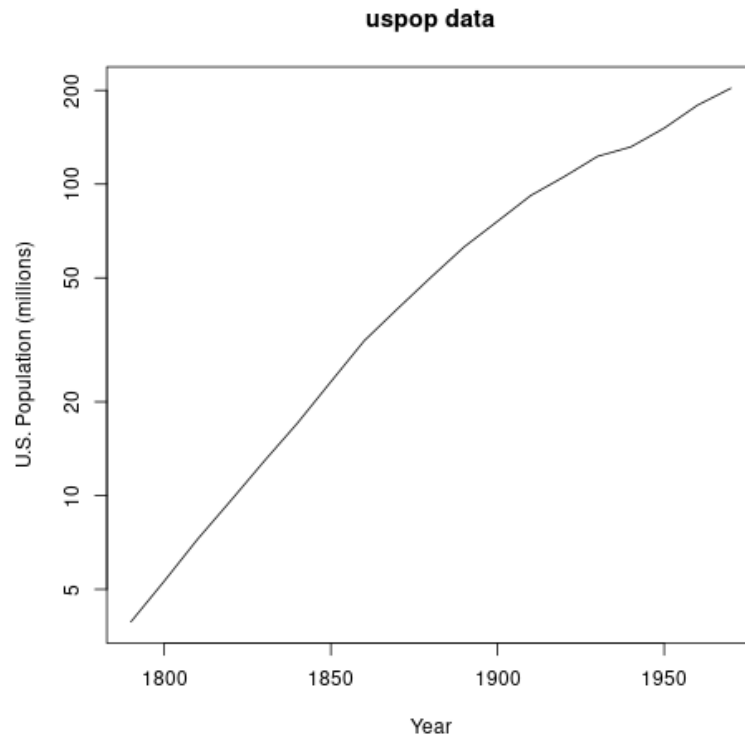
```
plot(uspop,
```

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



- Return to the help file. It should still be open in one of your Emacs buffers: `*help[R] (uspop)*`. Scroll to the end of the document (`M-SHIFT->`) and copy the `plot` command in the "Examples" section into the code block below, then run it.

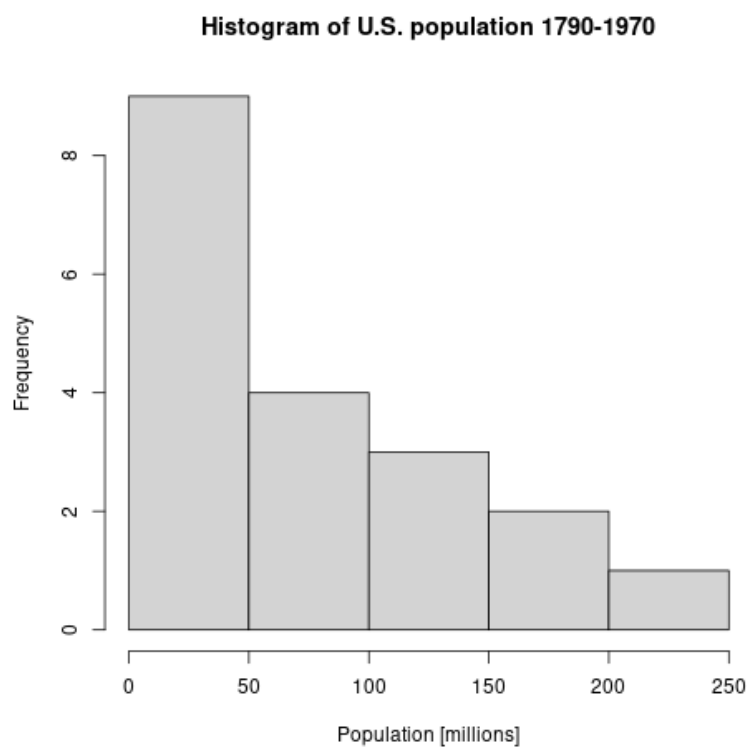
```
plot(uspop,
log = "y",
main = "uspop data",
xlab = "Year",
ylab = "U.S. Population (millions)")
```



8. Can you explain the difference between these two plots? Write your answer in the box below:
- In the first plot, the population is plotted on a linear scale where equal distances correspond to the same numerical change in population.
 - In the second plot, the y-axis is plotted on a logarithmic scale, which makes it easier to see relative growth rates: on a logarithmic scale, equal distances represent multiplicative rather than additive changes.
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",
xlab="Population [millions]")
```



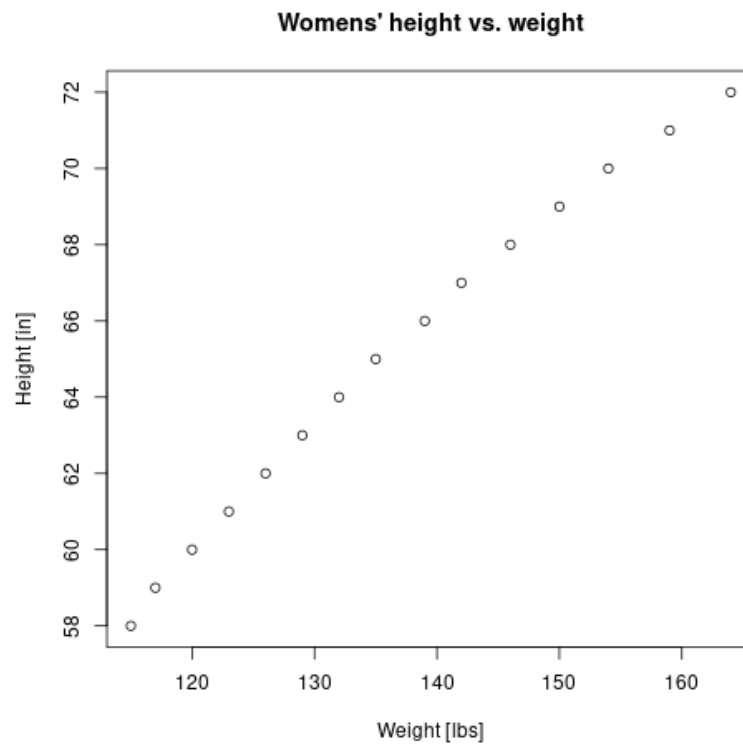
10. Load the `women` dataset and display its structure:

```
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.

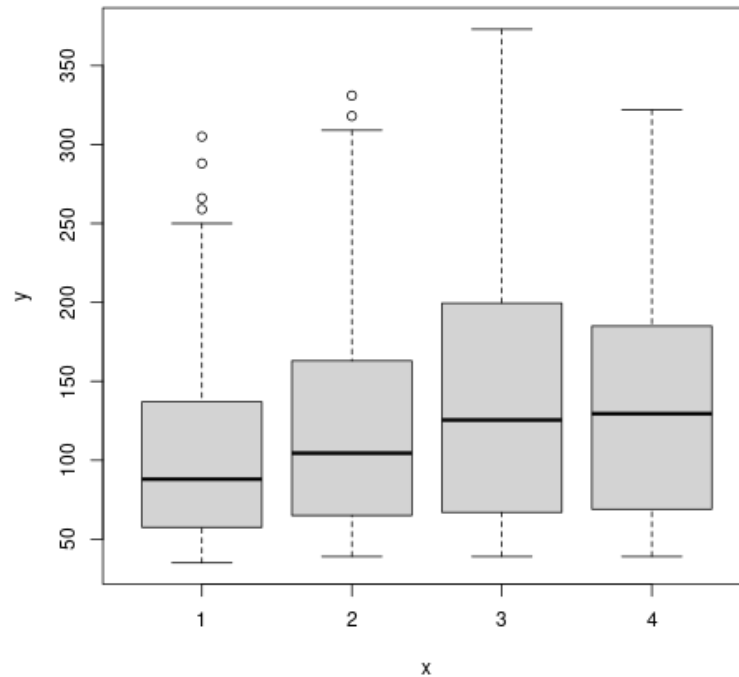
```
plot(women$height ~ women$weight,
     main="Womens' height vs. weight",
     xlab="Weight [lbs]",
     ylab="Height [in]")
```



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`:

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



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".

```
barplot(table(ChickWeight$Diet),
        main="Chickens by diet type",
        names.arg=c("Diet 1", "Diet 2", "Diet 3", "Diet 4"),
        ylab="Number of Chickens")
```

