

This will use the tidyverse package and the “gapminder” data set

```
library(tidyverse)

## -- Attaching packages ----- tidyverse 1.2.1 --
## v ggplot2 3.1.0      v purrr  0.2.5
## v tibble  2.0.1      v dplyr  0.7.8
## v tidyr   0.8.2      v stringr 1.4.0
## v readr   1.3.1      v forcats 0.3.0

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()

library(gapminder)
data("gapminder")
df = gapminder
head(gapminder)

## # A tibble: 6 x 6
##   country      continent  year lifeExp      pop gdpPercap
##   <fct>        <fct>    <int>  <dbl>    <int>    <dbl>
## 1 Afghanistan Asia      1952   28.8  8425333    779.
## 2 Afghanistan Asia      1957   30.3  9240934    821.
## 3 Afghanistan Asia      1962   32.0 10267083    853.
## 4 Afghanistan Asia      1967   34.0 11537966    836.
## 5 Afghanistan Asia      1972   36.1 13079460    740.
## 6 Afghanistan Asia      1977   38.4 14880372    786.
```

Exercise 1

```
levels(df$continent)

## [1] "Africa" "Americas" "Asia" "Europe" "Oceania"
```

Add “Antarctica” to the levels of possible continents

Exercise 2

Add 3 more continents to the factor levels: “North America”, “South America”, and “Central America”

Exercise 3

Consider the following groupings of countries

```
south <- c("Argentina", "Bolivia", "Brazil", "Chile", "Colombia", "Ecuador",
           "Paraguay", "Peru", "Uruguay", "Venezuela")
north <- c("United States", "Canada", "Mexico")
central <- c("Costa Rica", "El Salvador", "Guatemala", "Honduras", "Nicaragua", "Panama")
```

Currently, they are all in the “Americas” continent group. Write some code to move them into their new respective continents

Exercise 4

Clean up the “Continent” categories to remove “Americas” and “Antarctica”

```
levels(df$continent)

## [1] "Africa"    "Americas" "Asia"      "Europe"    "Oceania"
```

Exercise 5

Statistical models often use the FIRST level of a factor as a reference group Reorder the levels of the “Continent” factor so that “North America” will be the reference group

Exercise 6

Ordered Factors:

```
status <- c("Lo", "Hi", "Med", "Med", "Hi", "Lo", "Hi", "Med", "Med", "Hi")
status <- factor(status)
ordered.status <- factor(status, levels = c("Lo", "Med", "Hi"), ordered = TRUE)

table(status)
```

```
## status
##  Hi  Lo Med
##   4   2   4
```

```
table(ordered.status)
```

```
## ordered.status
##  Lo Med  Hi
##   2   4   4
```

Some functions in R treat ordered factors differently than unordered factors What is different about an ordered factor?

Exercise 7

```
as.numeric(status) + 1
```

```
## [1] 3 2 4 4 2 3 2 4 4 2
```

Figure out what is going on with the expression above.

Exercise 8

```
sub_status <- status[status %in% c("Lo", "Hi")] # extract only the "Lo" and "Hi" values
sub_status
```

```
## [1] Lo Hi Hi Lo Hi Hi
## Levels: Hi Lo Med
```

Re-factor “sub_status” to remove the “Med” level

Exercise 9

```
status2 <- factor(c("Med", "Hi", "Lo", "Hi", "Medium High"))  
c(status, status2)
```

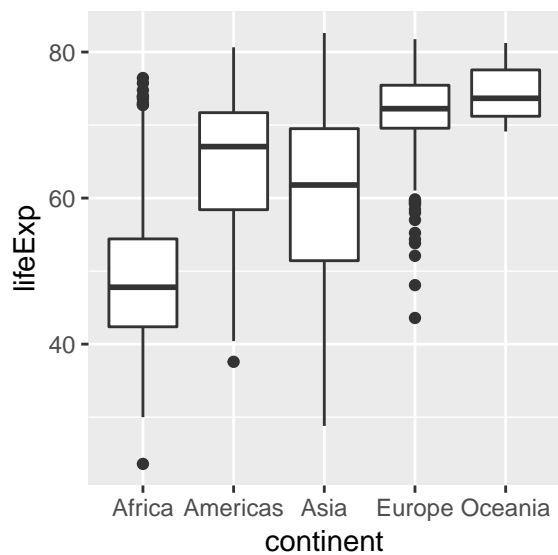
```
## [1] 2 1 3 3 1 2 1 3 3 1 3 1 2 1 4
```

Find a way to properly combine the two factored vectors “status” and “status2” Make sure the resulting vector is a factor as well

Exercise 10

Look at the following plot:

```
ggplot(df, aes(x=continent, y=lifeExp)) + geom_boxplot()
```

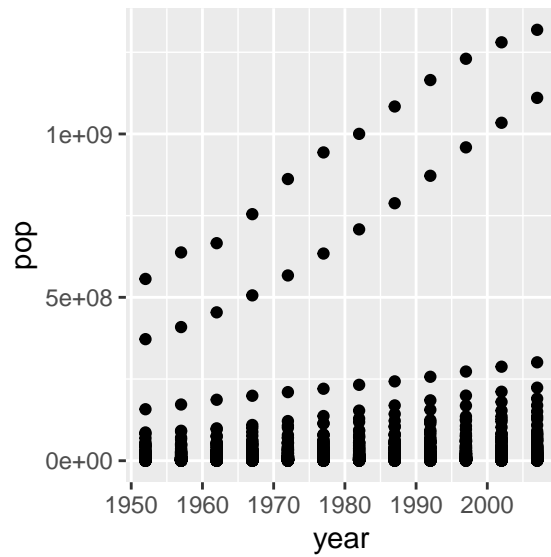


Re-make the plot so that the bars are in descending order

Exercise 11

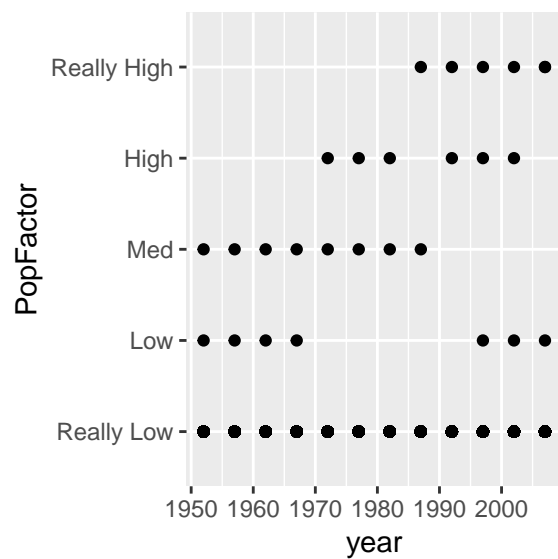
If:

```
ggplot(df, aes(x=year, y=pop)) + geom_point()
```



And ...

```
df$PopFactor <- cut(df$pop,5,labels = c("Really Low","Low","Med","High","Really High"))
ggplot(df, aes(x=year,y=PopFactor)) + geom_point()
```



Convert the `gdpPerCap` to an ordered factor with 6 levels and plot it against year