Essential Skills in R

with Marieke Jones

Homework

Getting Started

Create a new R project directory for the homework assignment. (In RStudio > File > New Project)

We're going to work with a different dataset for the homework than we did in the workshop. This one is a cleaned-up excerpt from the famous Gapminder dataset. Download **gapminder.csv** from The HSL Workshop Materials page (data.hsl.virginia.edu/workshop-materials). Save it into your project directory so you can access it easily from R.

Load the **tidyverse** OR the **readr**, **dplyr** and **ggplot2** packages, and read the gapminder data into R using the `read_csv()` function. Assign the data to an object called `gm`. Run `gm` to display it.

NUMBER 1

A. What are the dimensions of this dataset?

```
## [1] 1704    6
```

B. Calculate the mean life expectancy (`lifeExp`) overall (for all the data).

```
## [1] 59.47444
```

C. How many countries are included in this dataset? Nest the functions `length()` and `unique()` together to find out.

```
## [1] 142
```

dplyr

Many of the below problems deal with the dplyr package. If you want to learn more about dplyr, Here is the package introduction (https://dplyr.tidyverse.org/) and Here is a nice tutorial (https://rpubs.com/justmarkham/dplyr-tutorial)

NUMBER 2

A. What is the lowest (`min()`) life expectancy?

```
## [1] 23.599
```

B. Which observation (country & year) had the lowest life expectancy? One suggestion for a solution is to use the `%>%` from the {tidyverse} or {dplyr} to take the dataset then `arrange()` to sort the data by life expectancy then `head(1)` to get the first row of the sorted dataframe. There are other ways to solve this too.

```
## # A tibble: 1 x 6
##   country continent  year lifeExp     pop gdpPercap
##   <chr>   <chr>     <int>   <dbl>   <int>     <dbl>
## 1 Rwanda  Africa     1992    23.6 7290203      737.
```

C. Find the 10 observations with the lowest life expectancy. Use the code from B as a start.

```
## # A tibble: 10 x 6
##    country      continent  year lifeExp     pop gdpPercap
##    <chr>        <chr>     <int>   <dbl>   <int>     <dbl>
##  1 Rwanda       Africa     1992    23.6 7290203      737.
##  2 Afghanistan  Asia       1952    28.8 8425333      779.
##  3 Gambia       Africa     1952    30.0  284320      485.
##  4 Angola       Africa     1952    30.0 4232095     3521.
##  5 Sierra Leone Africa     1952    30.3 2143249      880.
##  6 Afghanistan  Asia       1957    30.3 9240934      821.
##  7 Cambodia     Asia       1977    31.2 6978607      525.
##  8 Mozambique   Africa     1952    31.3 6446316      469.
##  9 Sierra Leone Africa     1957    31.6 2295678     1004.
## 10 Burkina Faso Africa     1952    32.0 4469979      543.
```

D. What is the average gdpPercap for these observations? Use the code from C and then add a call to `summarize()`. Compare that number to the average gdpPercap for the whole dataset.

```
## # A tibble: 1 x 1
##   `mean(gdpPercap)`
##               <dbl>
## 1              976.
```

```
## [1] 7215.327
```

E. Use `filter()` then `group_by()` and then `summarize()` to find the mean life expectancy for each continent in the year 1997.

```
## # A tibble: 5 x 2
##   continent `mean(lifeExp)`
##   <chr>               <dbl>
## 1 Africa               53.6
## 2 Americas             71.2
## 3 Asia                 68.0
## 4 Europe               75.5
## 5 Oceania              78.2
```

F. How many unique countries are represented per continent? (Try `group_by` then `summarize(n_distinct())`)

```
## # A tibble: 5 x 2
##   continent `n_distinct(country)`
##   <chr>                     <int>
## 1 Africa                       52
## 2 Americas                     25
## 3 Asia                         33
## 4 Europe                       30
## 5 Oceania                       2
```

ggplot2

These next problems will deal with ggplot2. The ggplot2 package allows you to build a plot layer-by-layer by specifying:

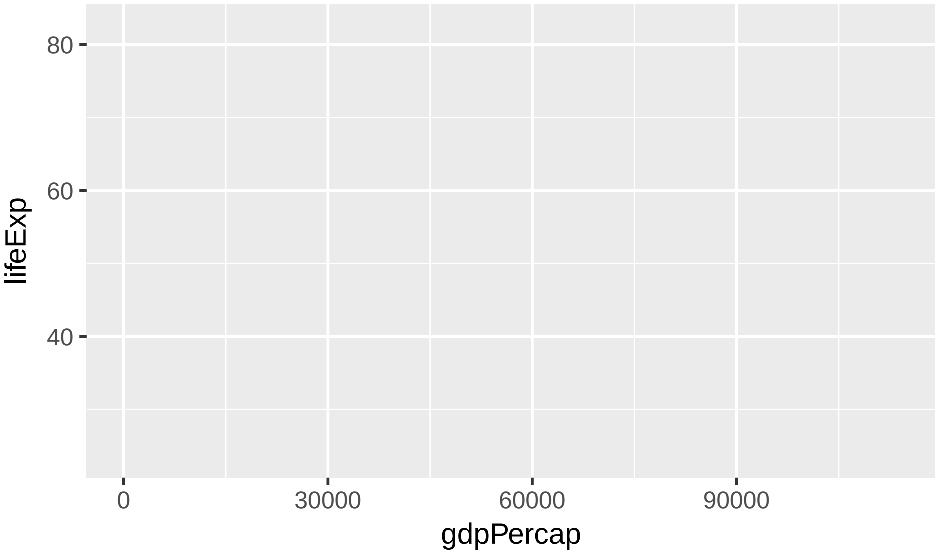
- **aesthetics** that map variables in the data to axes on the plot or to plotting size, shape, color, etc.,
- a **geom**, which specifies how the data are represented on the plot (points, lines, bars, etc.),
- a **stat**, a statistical transformation or summary of the data applied prior to plotting,

- **facets**, which we've already seen above, that allow the data to be divided into chunks on the basis of other categorical or continuous variables and the same plot drawn for each chunk.

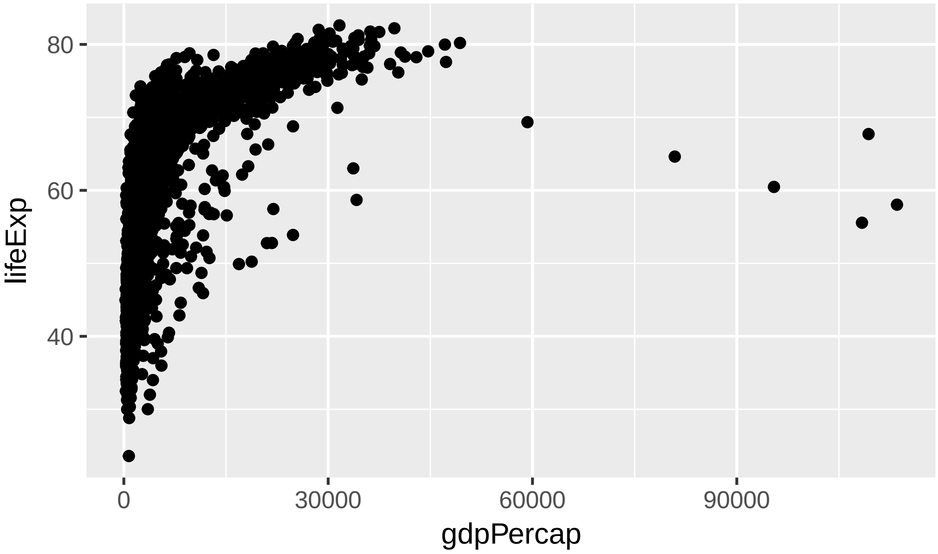
A great resource for help is the R Graphics Cookbook (http://www.cookbook-r.com/Graphs/)

NUMBER 3

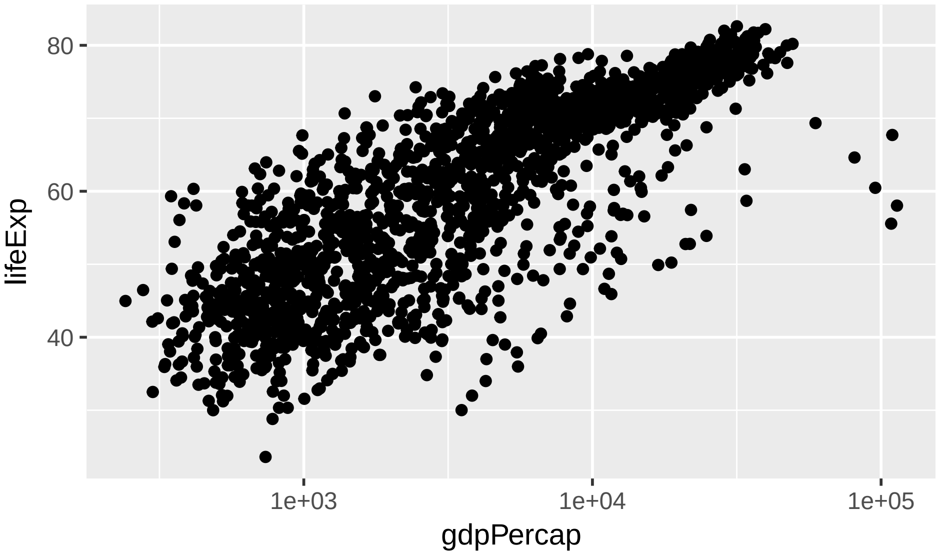
A. Create a blank canvas of a plot showing `gdpPercap` on the X-axis and `lifeExp` on the Y-axis



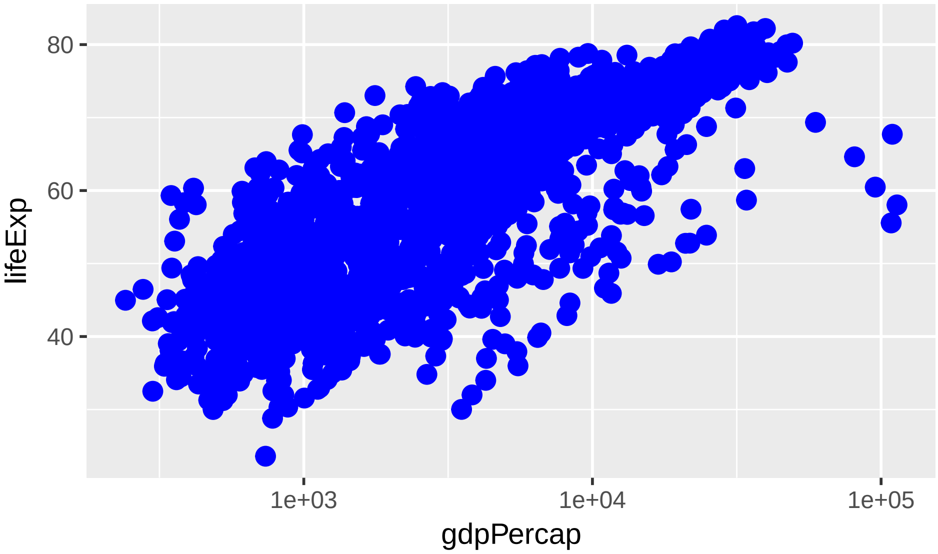
B. Add `geom_point()` to the above canvas.



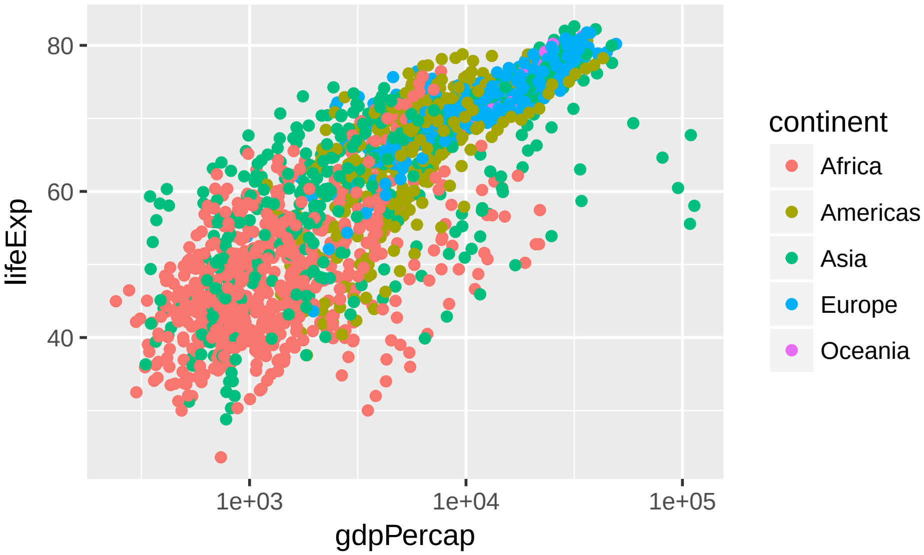
C. Based on the above plot, let's take the log10 of the x-axis. Add `scale_x_log10()` to the canvas and plot the points



D. Keep the log10 x-axis for the rest of the plots. Now change the color of the points (`color == "blue"`) and make them larger (`size = 3`)

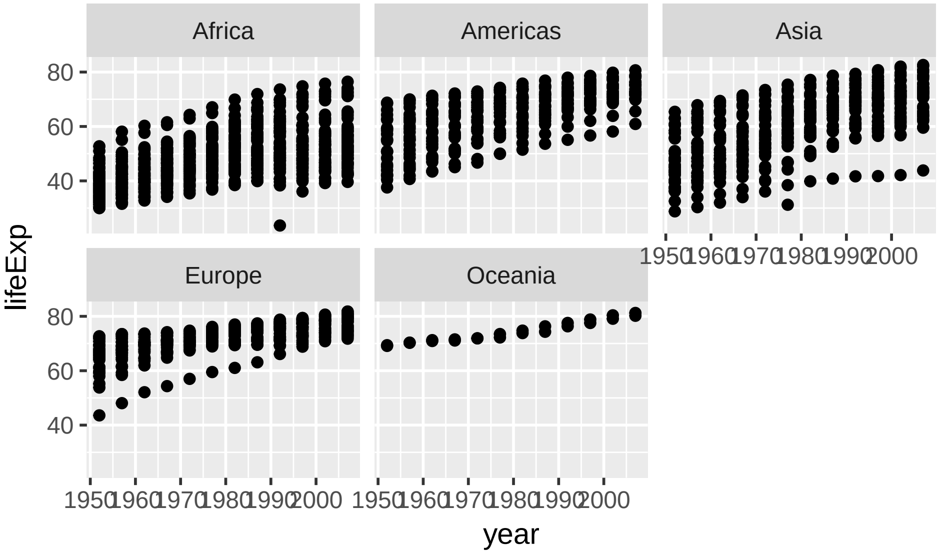


E. Instead of changing the color and point size in the call to geom_point, let's color by continent (as a call to `aes`)

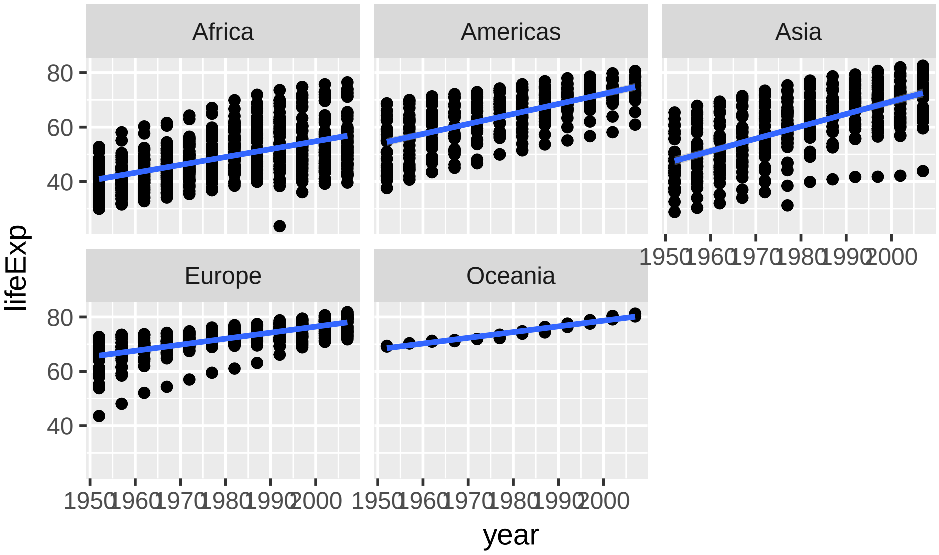


NUMBER 4

A. Make a scatter plot of `year` on the x against `lifeExp` on the y-axis, faceted by continent (`facet_wrap(~ continent)`)

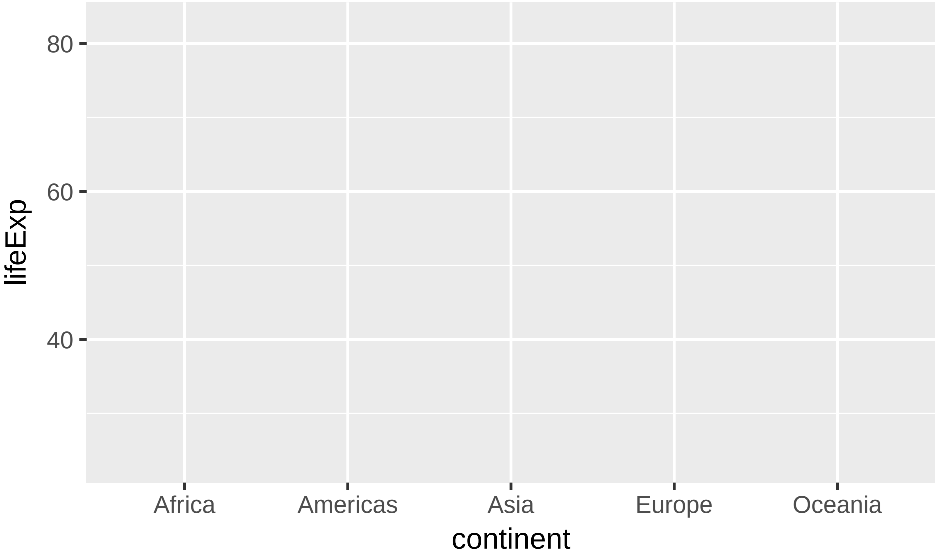


B. Add `geom_smooth()` with method = “lm” or “loess” to each facet. *Hint*: put the geom_smooth before the facet_wrap(). *I’ve shown the method = “lm” here*

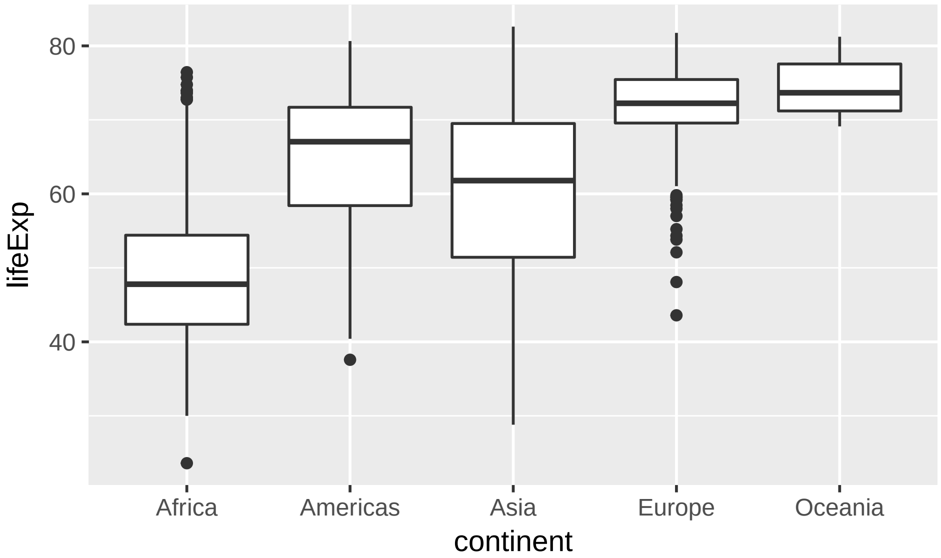


NUMBER 5

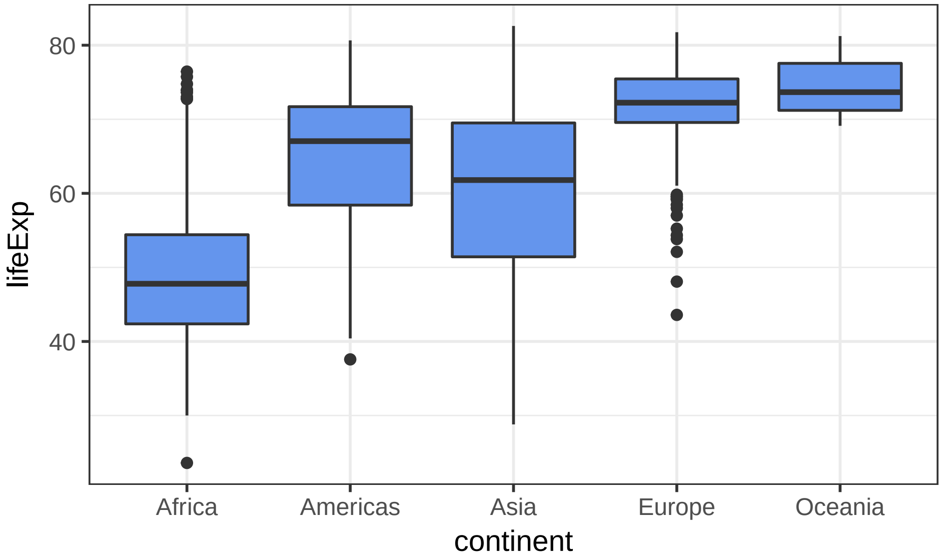
A. Make a blank canvas with continent on the x and lifeExp on the y



B. Add geom_boxplot() to the above canvas. For categorical variables, boxplots are a nice way to visualize data.



C. Use fill = "cornflower blue" to color the boxplots and add theme_bw() to the canvas to plot without the gray background. Check out other themes too!



Solutions

- Want to see the worked answers? Check them out at on the Workshop Materials page (data.hsl.virginia.edu/workshop-materials)
- Have questions about code? Email Marieke Jones at marieke@virginia.edu