Monday Notes

Tim Riffe

July 1, 2019

# What is R?

1. It’s functional:

a <- 43  
b <- 21  
a+b

## [1] 64

# same thing but this actually looks like a function:  
sum(a,b)

## [1] 64

1. extensible: people make packages

install.packages("gapminder")  
# install.packages("adiv")  
install.packages("tidyverse")

1. a statistical language

x <- rnorm(10)  
y <- x \* 2 \* runif(10, min = .9, max = 1.1)  
lm(y~x)

##   
## Call:  
## lm(formula = y ~ x)  
##   
## Coefficients:  
## (Intercept) x   
## 0.03262 2.01628

# gapminder

Gapminder is a tidy dataset.

Right about here I showed examples of tidy and not tidy datasets.

library(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.

We also see code to look into a data object and see what’s there. Plus a clicky way.

And then Kristen asked about case sensitivity in R (yes it is!) and we invoked an error and then I showed how to search for errors. The searcher package seems pretty great.

See, try typing this in the console (both lines)

library(searcher)  
str(Gapminder) # wrong case!  
search\_google() # what happened?  
"a" + 1  
search\_stackoverflow()

It would appear that if you want to use this search shortcut, then it’s better to copy and paste the error straight in the search function as a character string, e.g.: search\_stackoverflow("Error in "a" + 1 : non-numeric argument to binary operator").

FYI: if you knit this document with install.packages() or anything trying to use the internet then it can hang. Best to include eval = FALSE in the chunk statement.

library(tidyverse)

## ── Attaching packages ───────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.1.0 ✔ purrr 0.3.2   
## ✔ tibble 2.1.1 ✔ dplyr 0.8.0.1  
## ✔ tidyr 0.8.3 ✔ stringr 1.4.0   
## ✔ readr 1.3.1 ✔ forcats 0.4.0

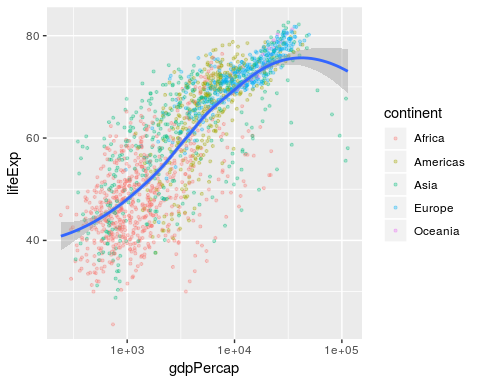
## ── Conflicts ──────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

p <- ggplot(data = gapminder,  
 mapping = aes(x = gdpPercap,  
 y = lifeExp,  
 color = continent))   
p <- p + geom\_point()   
p



Now we can *add* elements/metadata/instructions to the plot to make it the way we want:

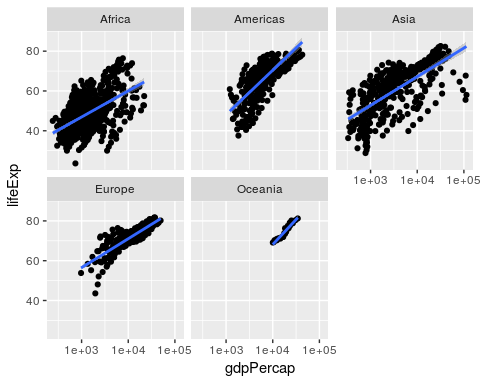
p <- ggplot(data = gapminder,  
 mapping = aes(x = gdpPercap,  
 y = lifeExp))  
p +   
 geom\_point(alpha = .3,   
 size = .8,  
 mapping=aes(color = continent)) +  
 scale\_x\_log10() +  
 geom\_smooth(method = "loess")



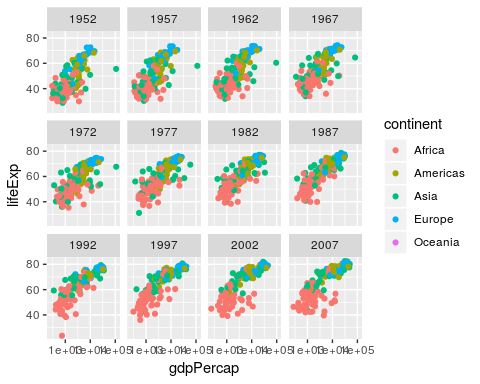
Keys to find:

# <- ^ \* + - {} () [] ` ~

p <- ggplot(gapminder,  
 mapping = aes(x = gdpPercap,  
 y = lifeExp))  
p +   
 geom\_point() +   
 facet\_wrap(~continent) +  
 scale\_x\_log10() +   
 geom\_smooth(method = "lm")



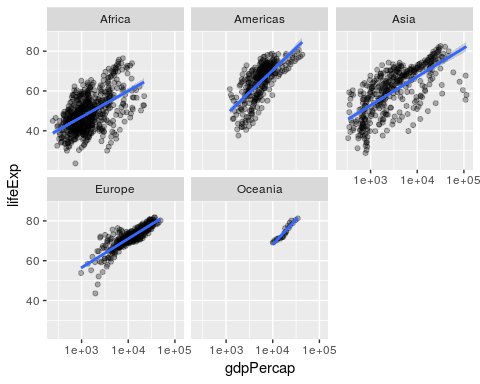
p <- ggplot(gapminder,  
 mapping = aes(x = gdpPercap,  
 y = lifeExp,  
 color = continent))  
p +   
 geom\_point() +   
 facet\_wrap(~year) +   
 scale\_x\_log10()



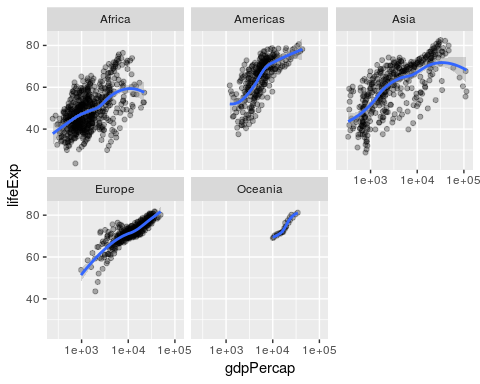
# Exercise 1

Smoothers: Do a Preston scatterplot panel by continent, showing different smoothing methods on top of the point clouds. For your own thinking: how come different methods gives such different uncertainties and what does that say about the data vs the method (way out of the realm of this course)?

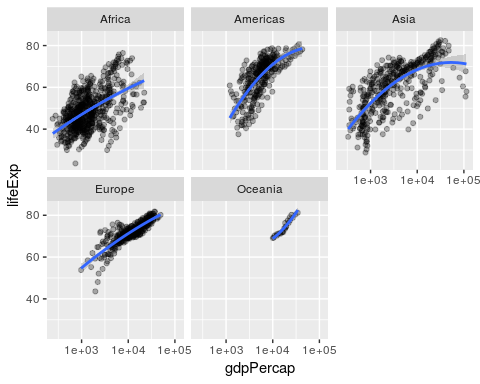
p <- ggplot(gapminder,  
 mapping = aes(x = gdpPercap,  
 y = lifeExp)) +  
 geom\_point(alpha = .3) +   
 facet\_wrap(~continent) +  
 scale\_x\_log10()  
# 1) linear fits  
p + geom\_smooth(method = 'lm')



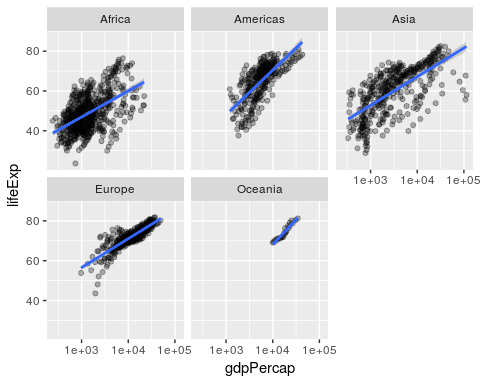
# 2) loess fits  
p + geom\_smooth(method = "loess")



# 3) GAM  
p + geom\_smooth(method = "gam",   
 formula = y~poly(x, 2))



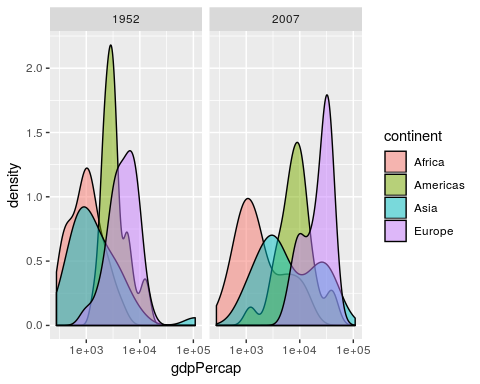
# 4) GLM  
p + geom\_smooth(method = "glm")



# Exercise 2

There is a geom called geom\_density() that draws a density function fit to your data. It only wants an x, not a y. You can have different overlapped densities by specifying alpha = .5, fill=x (where x is the thing you want to vary color on, continent in our case). You can subset data using filter(gapminder,continent != "Oceania"), to get rid of Oceania for example. Your challenge: Make a two panel plot, showing the gdpPercap density by continents in 1952 and 2007, where the density for each continent has a different semitransparent color. This will exercise like 5 things we learned, plus you’ll use subsetting. This one’s for Mr. Rosling. Do the same for lifeExp.

gap2 <- filter(gapminder, continent != "Oceania" &  
 ((year == 1952) | (year == 2007))  
 )  
gap2 <- filter(gapminder, continent != "Oceania" &  
 year %in% c(1952, 2007)  
 )  
ggplot(data = gap2,  
 mapping = aes(x = gdpPercap,  
 fill = continent)) +  
 geom\_density(alpha = .5) +   
 facet\_wrap(~year) +   
 scale\_x\_log10()



ggplot(data = gap2,  
 mapping = aes(x = lifeExp,  
 fill = continent)) +  
 geom\_density(alpha = .5) +   
 facet\_wrap(~year)

