Day 1 R review

Kara Woo

April 6, 2017

Arithmetic

```r
1 + 100
```

```
## [1] 101
```

```r
1 == 2
```

```
## [1] FALSE
```

```r
log(1)
```

```
## [1] 0
```

Creating variables

```r
x <- 1/40
log(x)
```

```
## [1] -3.688879
```

```r
x + x
```

```
## [1] 0.05
```

```r
x <- 100
x
```

```
## [1] 100
```

Vectors

```r
# Create some vectors
y <- c(1, 4.5, 2.7, 9)
y
```

```
## [1] 1.0 4.5 2.7 9.0
```

```r
z <- 1:5
z
```

```
## [1] 1 2 3 4 5
```

```r
# Create a vector of animals
animals <- c("monkey", "rabbit", "dog")
animals
```

```
## [1] "monkey" "rabbit" "dog"
```

```r
# Add 4 to each element of y
y + 4
```

```
## [1]  5.0  8.5  6.7 13.0
```

```
y <- y + 4 # save y as the original y + 4
y
```

```
## [1]  5.0  8.5  6.7 13.0
```

```
paste("The animal is: ", animals)
```

```
## [1] "The animal is:  monkey" "The animal is:  rabbit"
## [3] "The animal is:  dog"
```

To inspect the type of an object:

```
typeof(y)
```

```
## [1] "double"
```

```
typeof(z)
```

```
## [1] "integer"
```

```
typeof(animals)
```

```
## [1] "character"
```

Predict what happens when I do this:

```
new_vec <- c(5, 7.5, "hat")
typeof(new_vec)
```

```
## [1] "character"
```

```
new_vec
```

```
## [1] "5"   "7.5" "hat"
```

What is the value of each variable after each statement in the following:

```
mass <- 47.5
age <- 122
mass <- mass * 2.3
age <- age - 20
```

Remove a variable:

```
y2 <- c(1, 5, 7)
rm(y2)
y2
```

```
## Error in eval(expr, envir, enclos): object 'y2' not found
```

Loading data

```
cats <- read.csv(file = "data/feline-data.csv")
cats
```

```
##     coat weight likes_string
## 1 calico    2.1            1
## 2  black    5.0            0
## 3  tabby    3.2            1
```

```r
# View the class of the object (a data frame)
class(cats)
```

```
## [1] "data.frame"
```

```r
# Extract a column
cats$weight
```

```
## [1] 2.1 5.0 3.2
```

```r
cats$weight + 2
```

```
## [1] 4.1 7.0 5.2
```

```r
# Show the structure of an object
str(cats)
```

```
## 'data.frame':    3 obs. of  3 variables:
##  $ coat        : Factor w/ 3 levels "black","calico",..: 2 1 3
##  $ weight      : num  2.1 5 3.2
##  $ likes_string: int  1 0 1
```

```r
# Read in data without treating strings as factors
cats2 <- read.csv(file = "data/feline-data.csv",
                  stringsAsFactors = FALSE)
str(cats2)
```

```
## 'data.frame':    3 obs. of  3 variables:
##  $ coat        : chr  "calico" "black" "tabby"
##  $ weight      : num  2.1 5 3.2
##  $ likes_string: int  1 0 1
```

```r
# Load gapminder dataset
gapminder <- read.csv(file = "data/gapminder.csv",
                      stringsAsFactors = FALSE)
```

View information about the data frame:

```r
# Structure of the dataset
str(gapminder)
```

```
## 'data.frame':    1704 obs. of  6 variables:
##  $ country  : chr  "Afghanistan" "Afghanistan" "Afghanistan" "Afghanistan" ...
##  $ continent: chr  "Asia" "Asia" "Asia" "Asia" ...
##  $ year     : int  1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ lifeExp  : num  28.8 30.3 32 34 36.1 ...
##  $ pop      : num  8425333 9240934 10267083 11537966 13079460 ...
##  $ gdpPercap: num  779 821 853 836 740 ...
```

```r
# Number of rows, columns, rows & columns
nrow(gapminder)
```

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

```r
ncol(gapminder)
```

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

```r
dim(gapminder)
```

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

```r
# Column names
colnames(gapminder)
```

```
## [1] "country"   "continent" "year"      "lifeExp"   "pop"       "gdpPercap"
```

```r
# View first few rows
head(gapminder, n = 5)
```

```
##       country continent year lifeExp      pop gdpPercap
## 1 Afghanistan      Asia 1952  28.801  8425333  779.4453
## 2 Afghanistan      Asia 1957  30.332  9240934  820.8530
## 3 Afghanistan      Asia 1962  31.997 10267083  853.1007
## 4 Afghanistan      Asia 1967  34.020 11537966  836.1971
## 5 Afghanistan      Asia 1972  36.088 13079460  739.9811
```

```r
tail(gapminder)
```

```
##       country continent year lifeExp      pop gdpPercap
## 1699 Zimbabwe    Africa 1982  60.363  7636524  788.8550
## 1700 Zimbabwe    Africa 1987  62.351  9216418  706.1573
## 1701 Zimbabwe    Africa 1992  60.377 10704340  693.4208
## 1702 Zimbabwe    Africa 1997  46.809 11404948  792.4500
## 1703 Zimbabwe    Africa 2002  39.989 11926563  672.0386
## 1704 Zimbabwe    Africa 2007  43.487 12311143  469.7093
```

```r
# Summarize data
summary(gapminder)
```

```
##    country           continent              year         lifeExp     
##  Length:1704        Length:1704        Min.   :1952   Min.   :23.60  
##  Class :character   Class :character   1st Qu.:1966   1st Qu.:48.20  
##  Mode  :character   Mode  :character   Median :1980   Median :60.71  
##                                        Mean   :1980   Mean   :59.47  
##                                        3rd Qu.:1993   3rd Qu.:70.85  
##                                        Max.   :2007   Max.   :82.60  
##       pop              gdpPercap       
##  Min.   :6.001e+04   Min.   :   241.2  
##  1st Qu.:2.794e+06   1st Qu.:  1202.1  
##  Median :7.024e+06   Median :  3531.8  
##  Mean   :2.960e+07   Mean   :  7215.3  
##  3rd Qu.:1.959e+07   3rd Qu.:  9325.5  
##  Max.   :1.319e+09   Max.   :113523.1
```

Subsetting: Vectors

First, we'll create a named vector:

```r
x <- c(5.4, 6.2, 7.1, 4.8, 7.5)
x
```

```
## [1] 5.4 6.2 7.1 4.8 7.5
```

```r
names(x) <- c("a", "b", "c", "d", "e")
x
```

```
##   a   b   c   d   e 
## 5.4 6.2 7.1 4.8 7.5
```

Indexing:

```
x[1]
```

```
##   a
## 5.4
```

```
x[4]
```

```
##   d
## 4.8
```

```
x[c(1, 3)]
```

```
##   a   c
## 5.4 7.1
```

```
x[1:4]
```

```
##   a   b   c   d
## 5.4 6.2 7.1 4.8
```

```
x[c(1, 1, 3)]
```

```
##   a   a   c
## 5.4 5.4 7.1
```

```
x[6]
```

```
## <NA>
##   NA
```

Excluding:

```
x[-2]
```

```
##   a   c   d   e
## 5.4 7.1 4.8 7.5
```

```
x[c(-1, -5)]
```

```
##   b   c   d
## 6.2 7.1 4.8
```

```
x[-c(1, 5)]
```

```
##   b   c   d
## 6.2 7.1 4.8
```

Subsetting by name:

```
x[c("a", "c")]
```

```
##   a   c
## 5.4 7.1
```

Several ways to get b, c, and d:

```
x[c(-1, -5)]
```

```
##   b   c   d
## 6.2 7.1 4.8
```

```
x[2:4]
```

```
##   b   c   d
## 6.2 7.1 4.8
```

```
x[c("b", "c", "d")]
```

```
##   b   c   d
## 6.2 7.1 4.8
```

```
x[c(2, 3, 4)]
```

```
##   b   c   d
## 6.2 7.1 4.8
```

Logical subsetting:

```
x > 6
```

```
##     a     b     c     d     e
## FALSE  TRUE  TRUE FALSE  TRUE
```

```
x
```

```
##   a   b   c   d   e
## 5.4 6.2 7.1 4.8 7.5
```

```
x[x > 6]
```

```
##   b   c   e
## 6.2 7.1 7.5
```

```
x[x < 7]
```

```
##   a   b   d
## 5.4 6.2 4.8
```

Counting how many elements of a variable meet some condition:

```
length(x)
```

```
## [1] 5
```

```
x_new <- x[x > 6]
length(x_new)
```

```
## [1] 3
```

Subsetting: Data Frames

Extract a column (I'll wrap this in `head` so we don't have to see all the output):

```
head(gapminder$year)
```

```
## [1] 1952 1957 1962 1967 1972 1977
```

Subset rows and columns by index

```
gapminder[1:3,]
```

```
##       country continent year lifeExp      pop gdpPercap
## 1 Afghanistan      Asia 1952  28.801  8425333  779.4453
## 2 Afghanistan      Asia 1957  30.332  9240934  820.8530
## 3 Afghanistan      Asia 1962  31.997 10267083  853.1007
```

```r
gapminder[3,]
```

```
##       country continent year lifeExp      pop gdpPercap
## 3 Afghanistan      Asia 1962  31.997 10267083  853.1007
```

```r
gapminder[1:3, 1:3]
```

```
##       country continent year
## 1 Afghanistan      Asia 1952
## 2 Afghanistan      Asia 1957
## 3 Afghanistan      Asia 1962
```

Subset by column names (I'll wrap this in `head` so we don't have to see all the output):

```r
head(gapminder[, c("lifeExp", "pop")])
```

```
##   lifeExp      pop
## 1  28.801  8425333
## 2  30.332  9240934
## 3  31.997 10267083
## 4  34.020 11537966
## 5  36.088 13079460
## 6  38.438 14880372
```

Logical subsetting:

```r
head(gapminder[gapminder$year < 1960, c("year", "pop")])
```

```
##    year      pop
## 1  1952  8425333
## 2  1957  9240934
## 13 1952  1282697
## 14 1957  1476505
## 25 1952  9279525
## 26 1957 10270856
```

Challenge exercise answers:

```r
# Extract observations collected for the year 1957 (using head() to show only the first few rows)
head(gapminder[gapminder$year == 1957,])
```

```
##        country continent year lifeExp      pop gdpPercap
## 2  Afghanistan      Asia 1957  30.332  9240934   820.853
## 14     Albania    Europe 1957  59.280  1476505  1942.284
## 26     Algeria    Africa 1957  45.685 10270856  3013.976
## 38      Angola    Africa 1957  31.999  4561361  3827.940
## 50   Argentina  Americas 1957  64.399 19610538  6856.856
## 62   Australia   Oceania 1957  70.330  9712569 10949.650
```

```r
# Extract all columns except 1 through to 4
head(gapminder[, c(-1:-4)])
```

```
##        pop gdpPercap
## 1  8425333  779.4453
## 2  9240934  820.8530
## 3 10267083  853.1007
## 4 11537966  836.1971
## 5 13079460  739.9811
## 6 14880372  786.1134
```

```
head(gapminder[, -1:-4])
```

```
##        pop gdpPercap
## 1  8425333  779.4453
## 2  9240934  820.8530
## 3 10267083  853.1007
## 4 11537966  836.1971
## 5 13079460  739.9811
## 6 14880372  786.1134
```

```
head(gapminder[, -c(1:4)])
```

```
##        pop gdpPercap
## 1  8425333  779.4453
## 2  9240934  820.8530
## 3 10267083  853.1007
## 4 11537966  836.1971
## 5 13079460  739.9811
## 6 14880372  786.1134
```

```
# Extract the rows where the life expectancy is longer the 80 years
head(gapminder[gapminder$lifeExp > 80, ])
```

```
##               country continent year lifeExp      pop gdpPercap
## 71          Australia   Oceania 2002  80.370 19546792  30687.75
## 72          Australia   Oceania 2007  81.235 20434176  34435.37
## 252            Canada  Americas 2007  80.653 33390141  36319.24
## 540            France    Europe 2007  80.657 61083916  30470.02
## 671 Hong Kong, China      Asia 2002  81.495  6762476  30209.02
## 672 Hong Kong, China      Asia 2007  82.208  6980412  39724.98
```

```
# Extract the first row, and the fourth and fifth columns (lifeExp and gdpPercap).
gapminder[1, 4:5]
```

```
##   lifeExp     pop
## 1  28.801 8425333
```

```
gapminder[1, c(4, 5)]
```

```
##   lifeExp     pop
## 1  28.801 8425333
```

```
# Advanced: extract rows that contain information for the years 2002 and 2007
head(gapminder[gapminder$year == 2002 | gapminder$year == 2007, ])
```

```
##        country continent year lifeExp      pop gdpPercap
## 11 Afghanistan      Asia 2002  42.129 25268405  726.7341
## 12 Afghanistan      Asia 2007  43.828 31889923  974.5803
## 23     Albania    Europe 2002  75.651  3508512 4604.2117
## 24     Albania    Europe 2007  76.423  3600523 5937.0295
## 35     Algeria    Africa 2002  70.994 31287142 5288.0404
## 36     Algeria    Africa 2007  72.301 33333216 6223.3675
```

```
head(gapminder[gapminder$year %in% c(2002, 2007), ])
```

```
##        country continent year lifeExp      pop gdpPercap
## 11 Afghanistan      Asia 2002  42.129 25268405  726.7341
## 12 Afghanistan      Asia 2007  43.828 31889923  974.5803
## 23     Albania    Europe 2002  75.651  3508512 4604.2117
## 24     Albania    Europe 2007  76.423  3600523 5937.0295
## 35     Algeria    Africa 2002  70.994 31287142 5288.0404
## 36     Algeria    Africa 2007  72.301 33333216 6223.3675
```

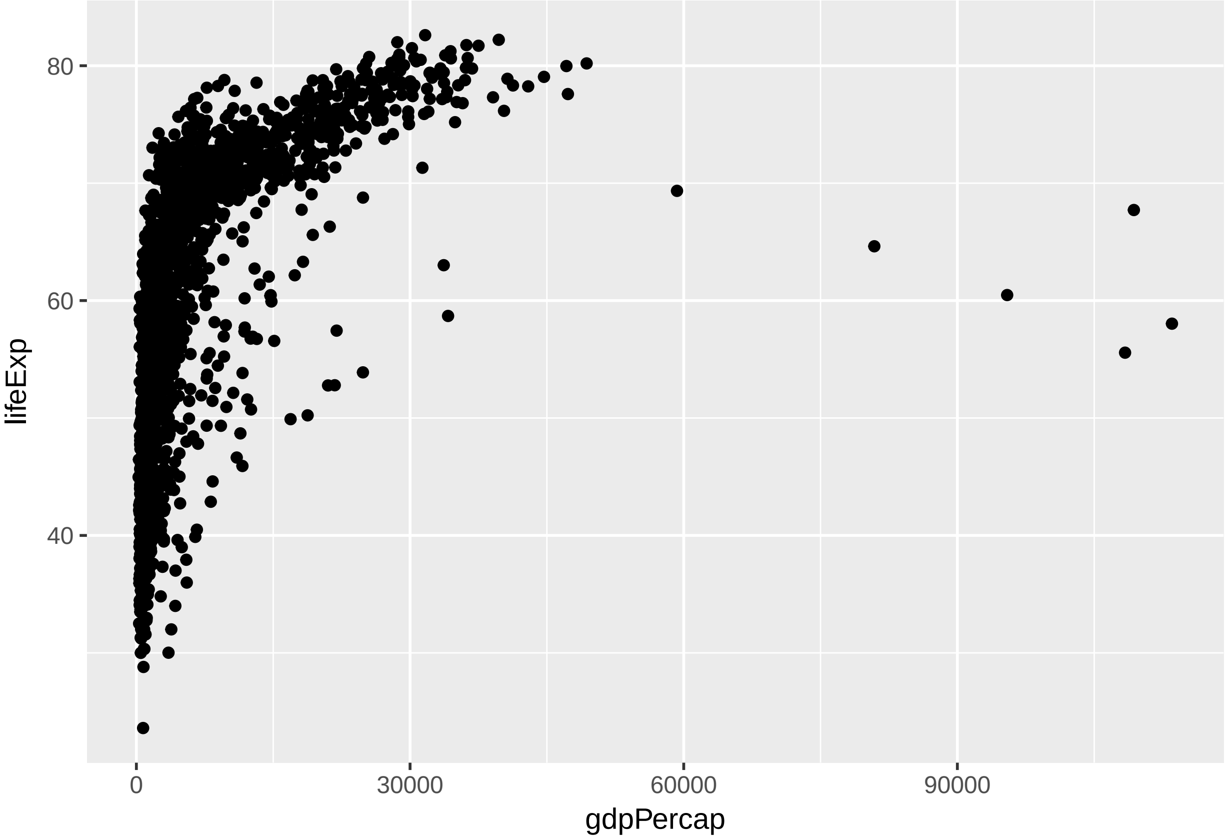
Plotting

```
# Installing the package (only needs to be done once)
install.packages("ggplot2")
```

```
# Load the package (needs to be done in each new R session when you want to use it)
library("ggplot2")
```

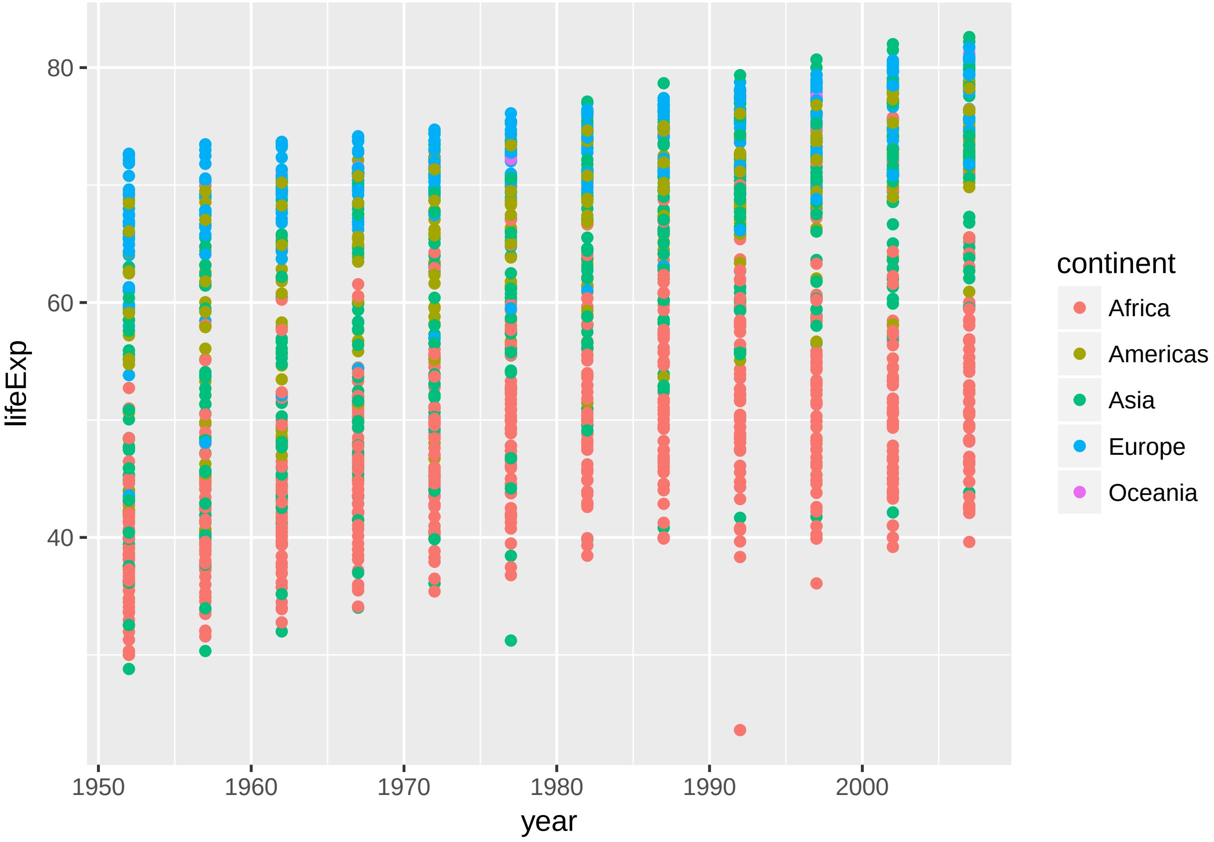
A basic scatter plot of GDP vs. life expectancy

```
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) +
  geom_point()
```



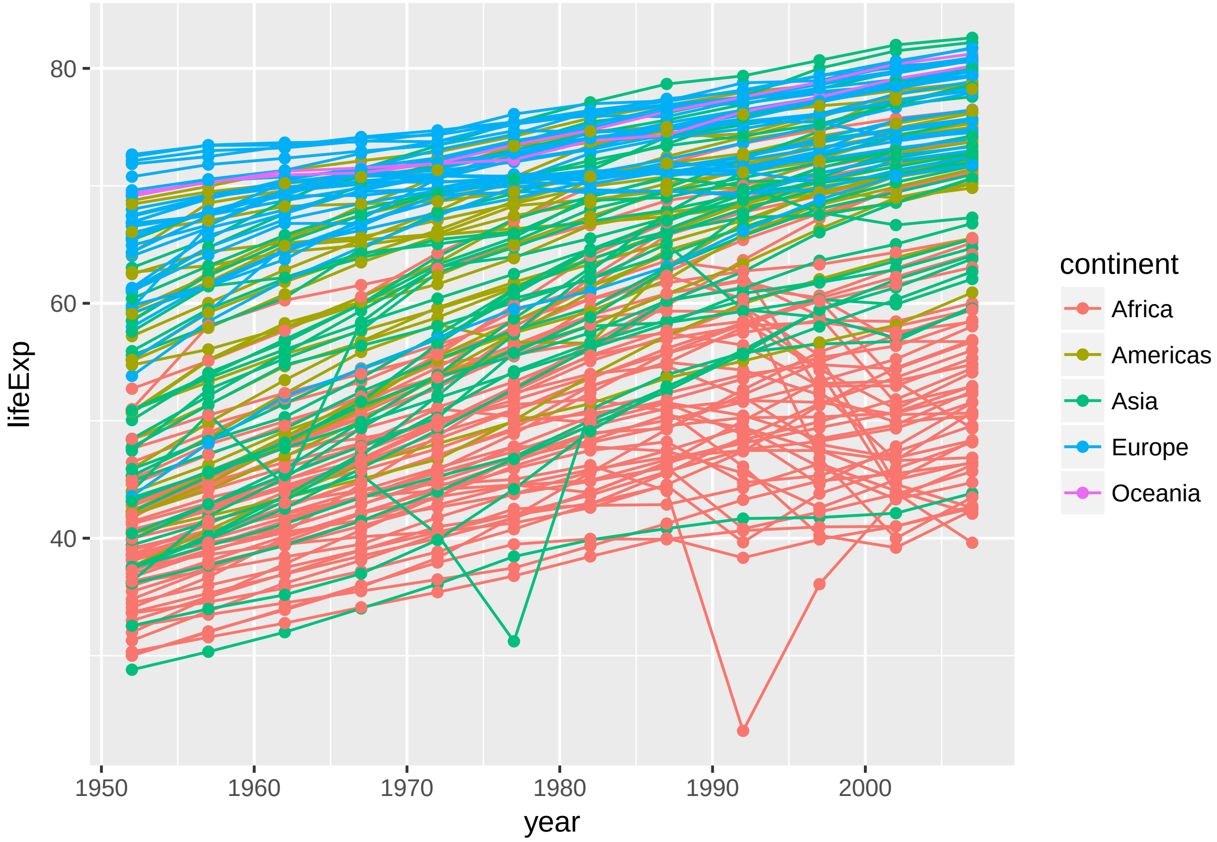
Life expectancy over time, points colored by continent:

```
ggplot(data = gapminder, aes(x = year, y = lifeExp, color = continent)) +
  geom_point()
```



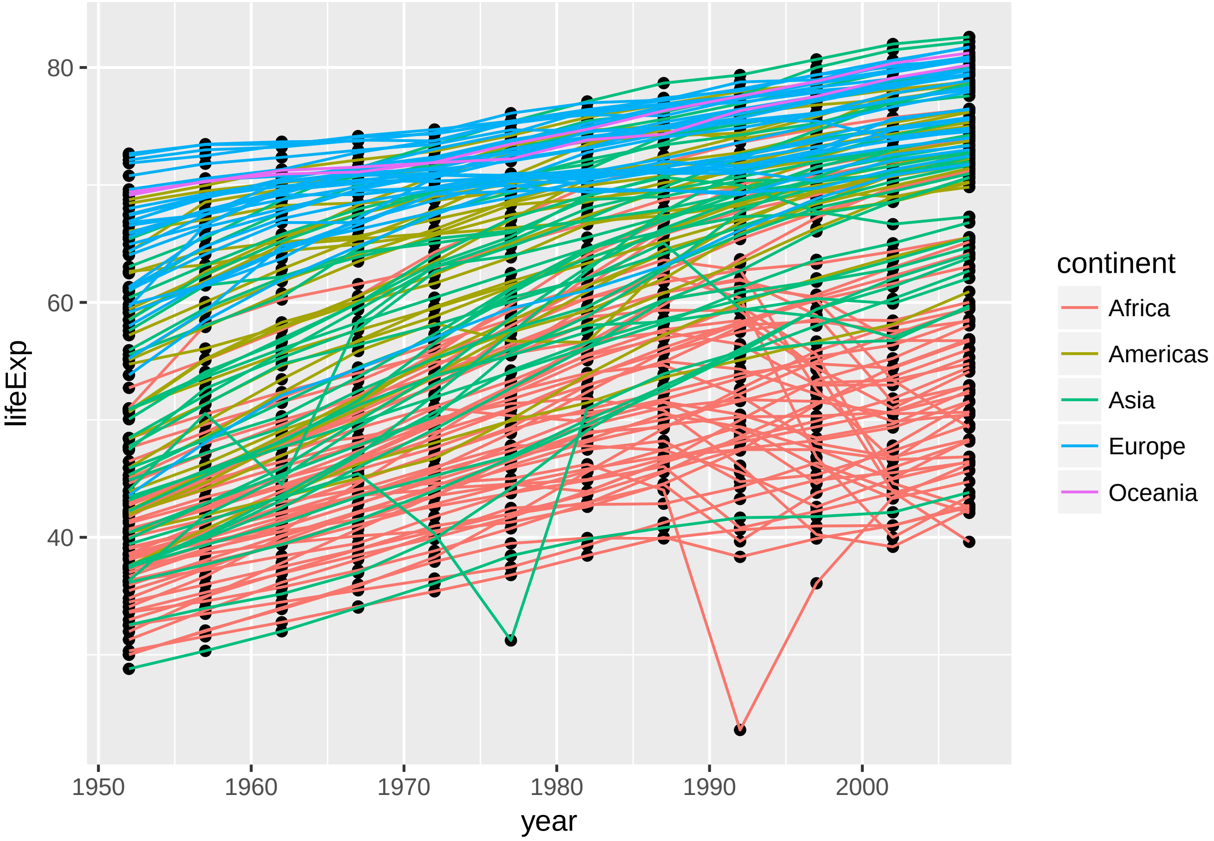
Life expectancy over time, lines and points

```
ggplot(data = gapminder, aes(x = year, y = lifeExp,
                             color = continent, by = country)) +
  geom_line() +
  geom_point()
```



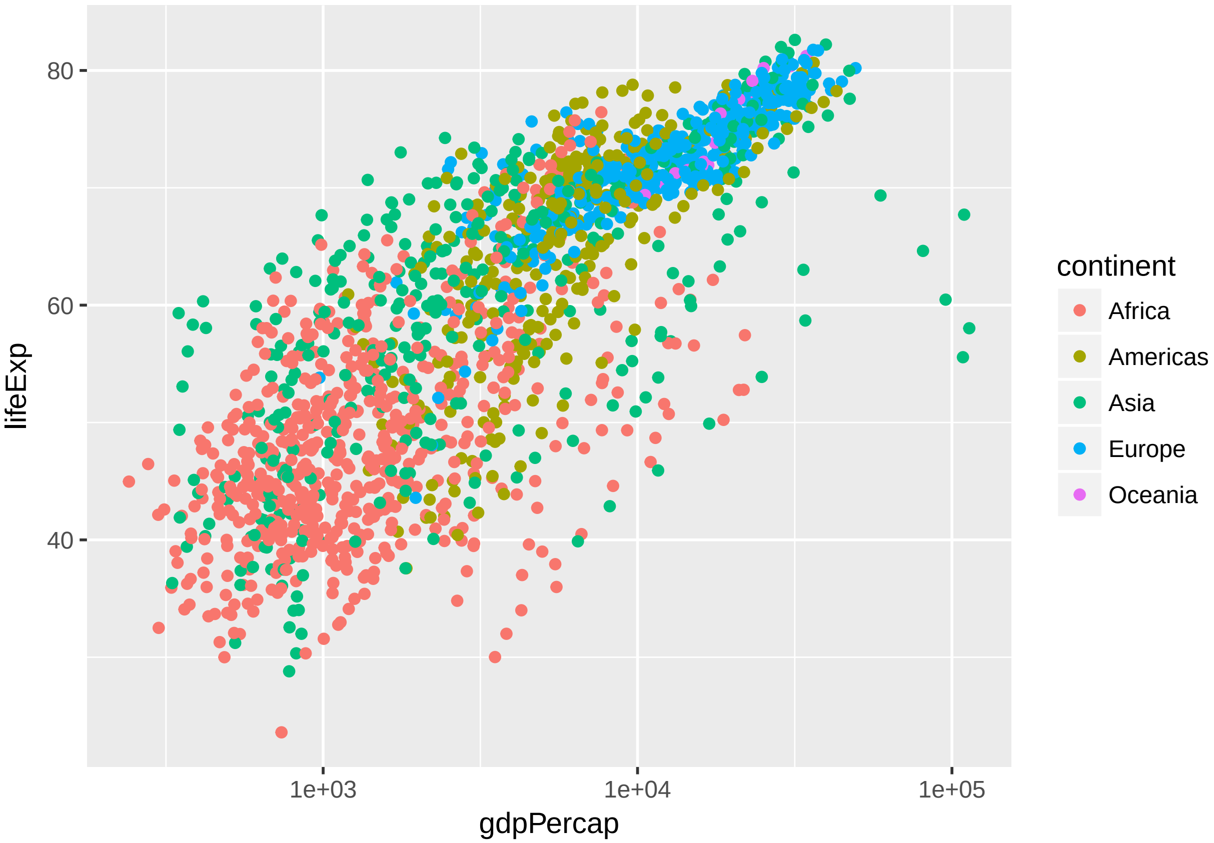
Only color the lines, not the points:

```
ggplot(data = gapminder, aes(x = year, y = lifeExp, by = country)) +
  geom_point() +
  geom_line(aes(color = continent))
```



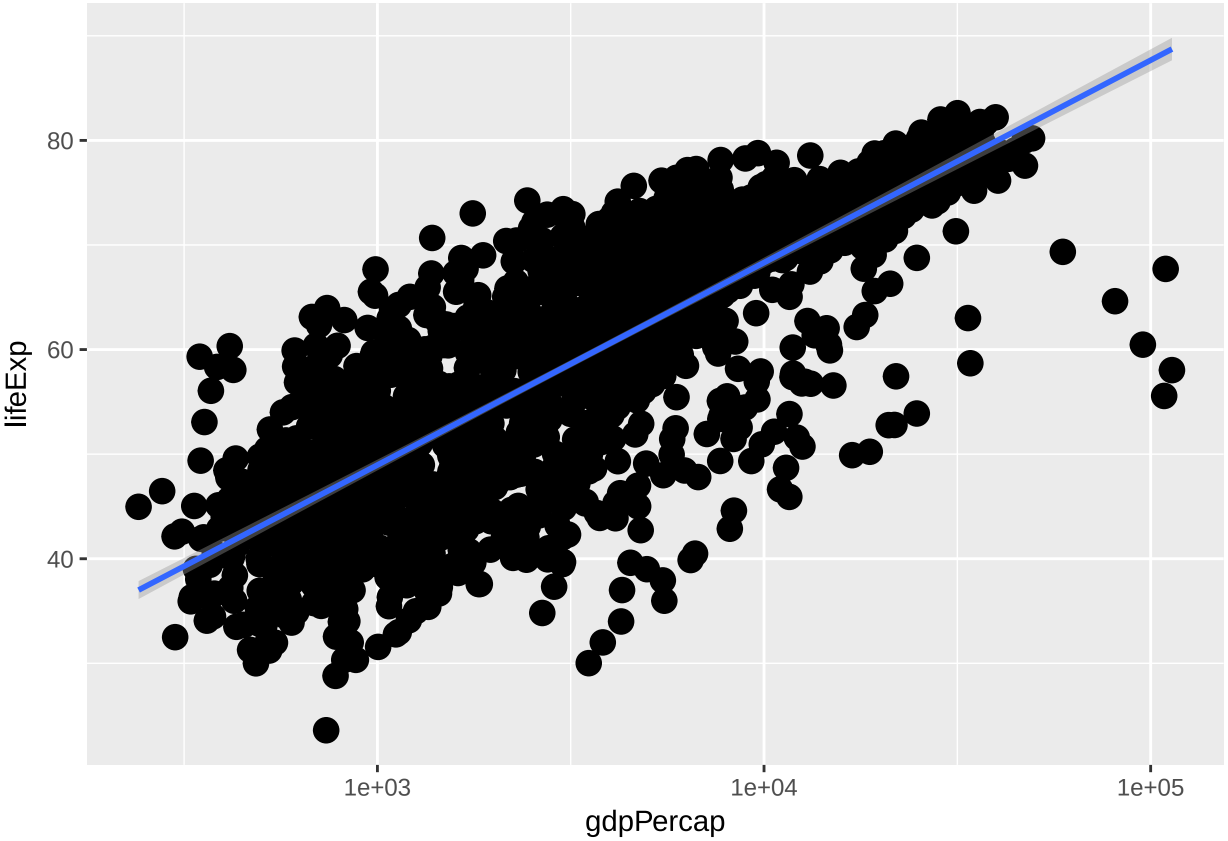
Log transformed x axis

```
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp, color = continent)) +
  geom_point() +
  scale_x_log10()
```



Fit a smoothing line and increase point size:

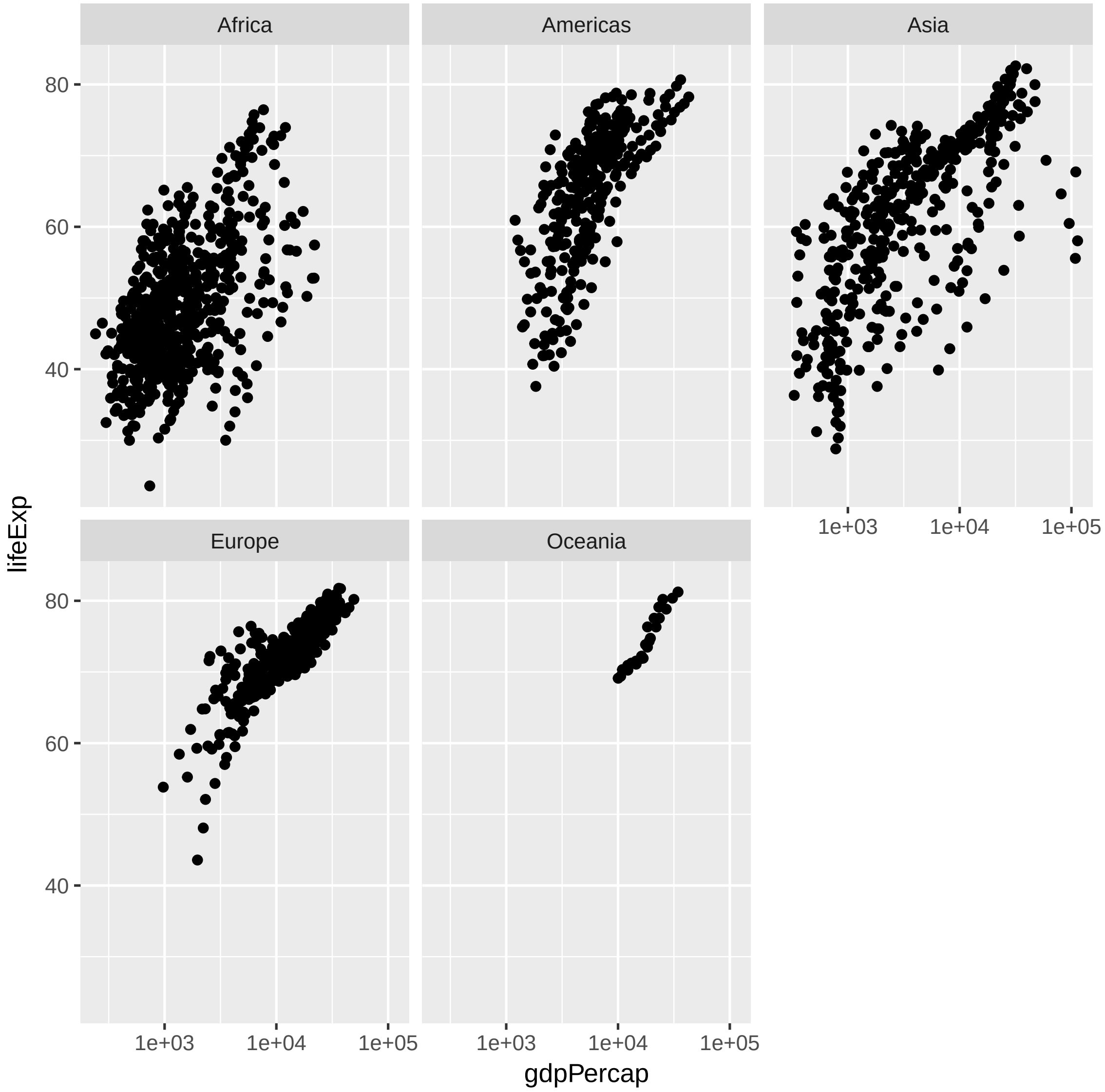
```
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) +
  geom_point(size = 4) +
  scale_x_log10() +
  geom_smooth(method = "lm")
```



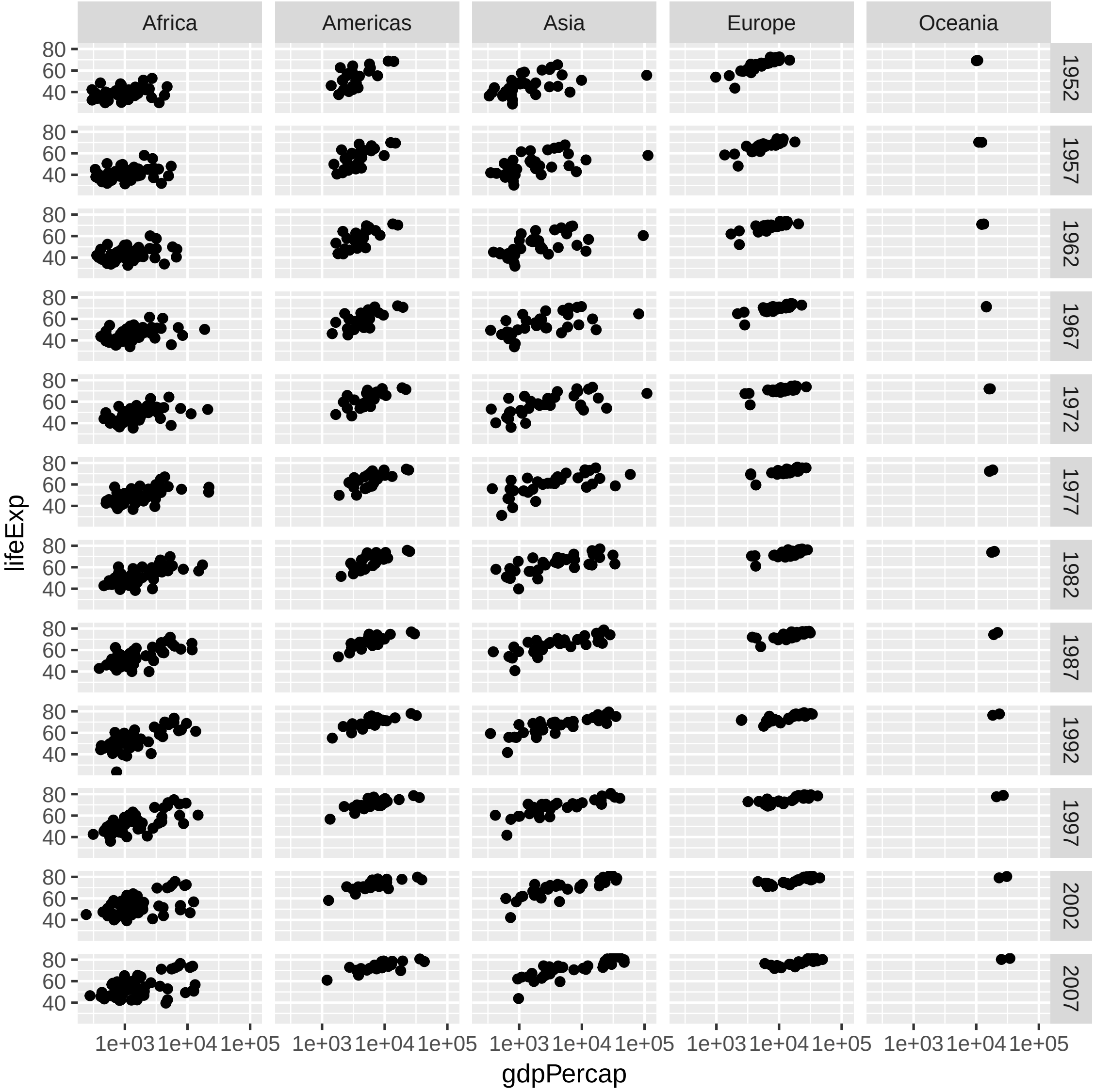
```
# Look up help for a function
?geom_smooth
```

Facetting:

```
# Facet grid
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) +
  geom_point() +
  scale_x_log10() +
  facet_wrap(~ continent)
```

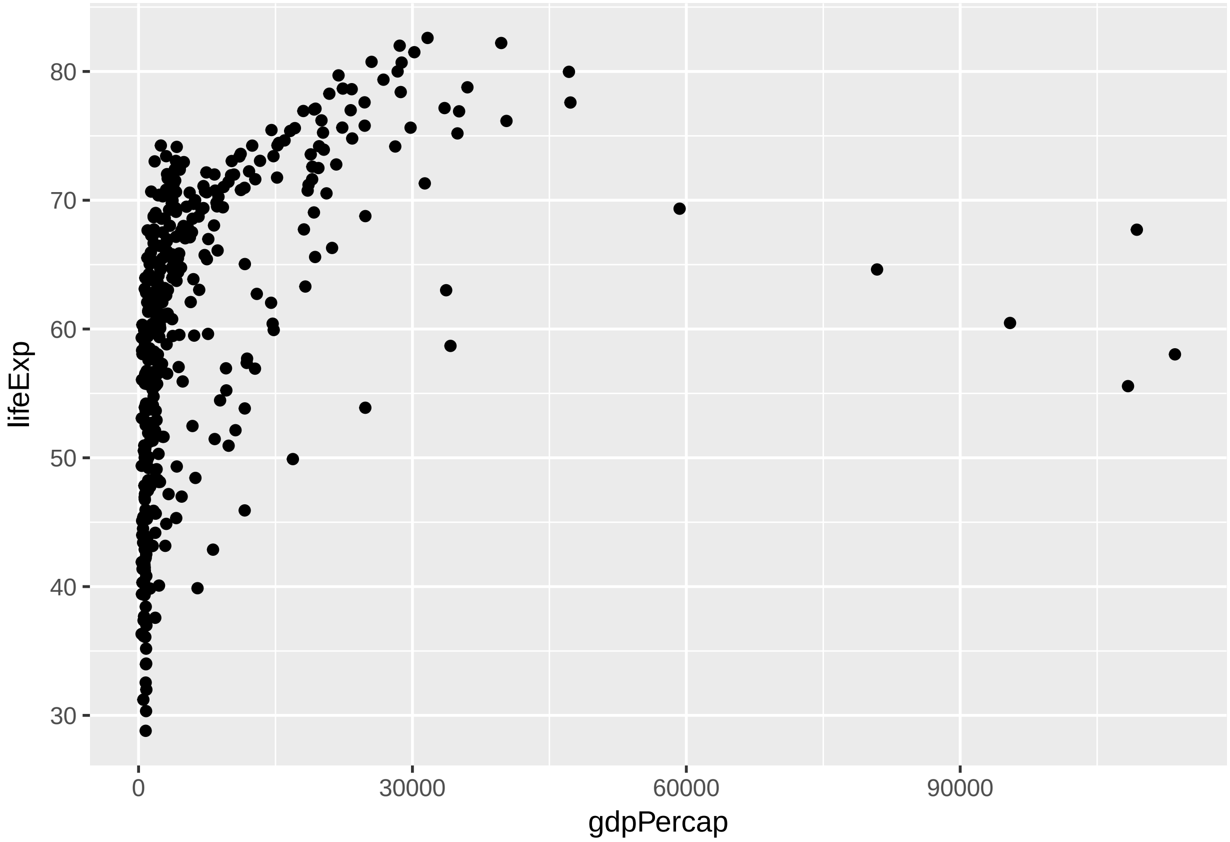


```
# Facet grid
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) +
  geom_point() +
  scale_x_log10() +
  facet_grid(year ~ continent)
```


Africa
Americas
Asia
Europe
Oceania
1952
1957
1962
1967
1972
1977
1982
1987
1992
1997
2002
2007
80
60
40
lifeExp
1e+03
1e+04
1e+05
gdpPercap

Subsetting data before plotting:

```
asia <- gapminder[gapminder$continent == "Asia", ]
ggplot(data = asia, aes(x = gdpPercap, y = lifeExp)) +
  geom_point()
```



Modifying labels and theme and saving a plot:

```r
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) +
  geom_point() +
  xlab("GDP per capita") +
  ylab("Life expectancy") +
  ggtitle("Figure 1") +
  theme_void() +
  ggsave("my_awesome_plot.png", width = 6, height = 4)
```