

# 2008-351-351m-451-w03a-p-Gapminder-EDA-ggplot2-tutorial.R

Jenny Bryan, modified by Roger French

13 September, 2022

```
library(knitr)
opts_chunk$set(fig.path = 'figs/bryan/scatterplot-',
              fig.width = 5, # the width for plots created by code chunk
              fig.height = 4, # the height for plots created by code chunk
              fig.align = 'center', # how to align graphics. 'left', 'right', 'center'
              dpi = 150,
              echo = TRUE, # if FALSE knitr won't display code in chunk above it's results
              error = TRUE) # report errors
```

Note: this report is made by rendering an R script. So the narrative is very minimal.

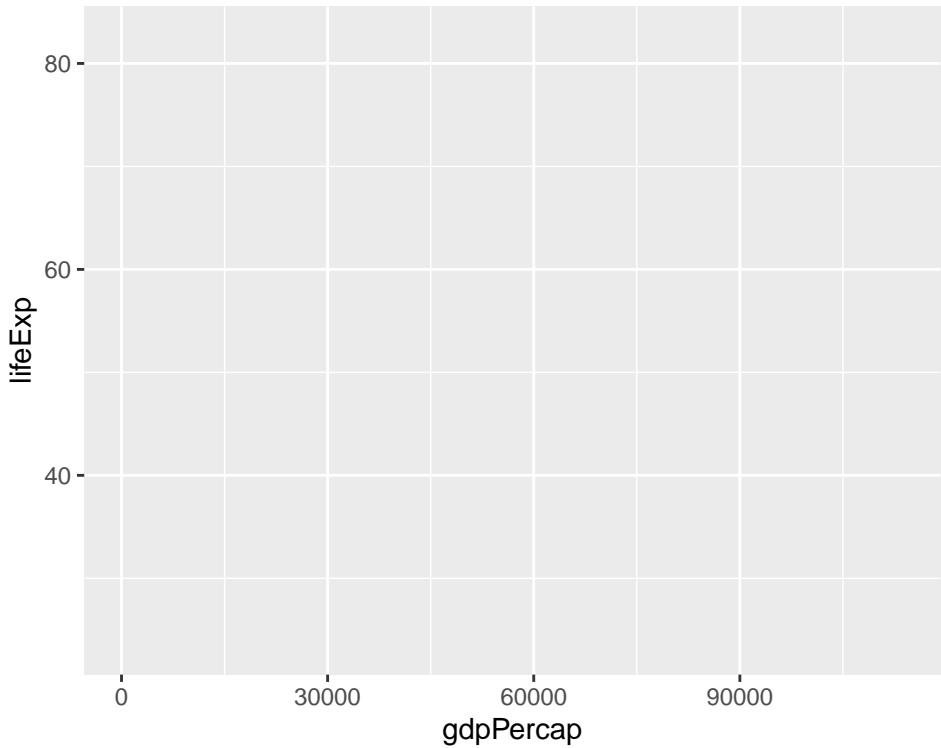
```
#####
# Scatterplot
#####

library(tibble)
library(ggplot2)
```

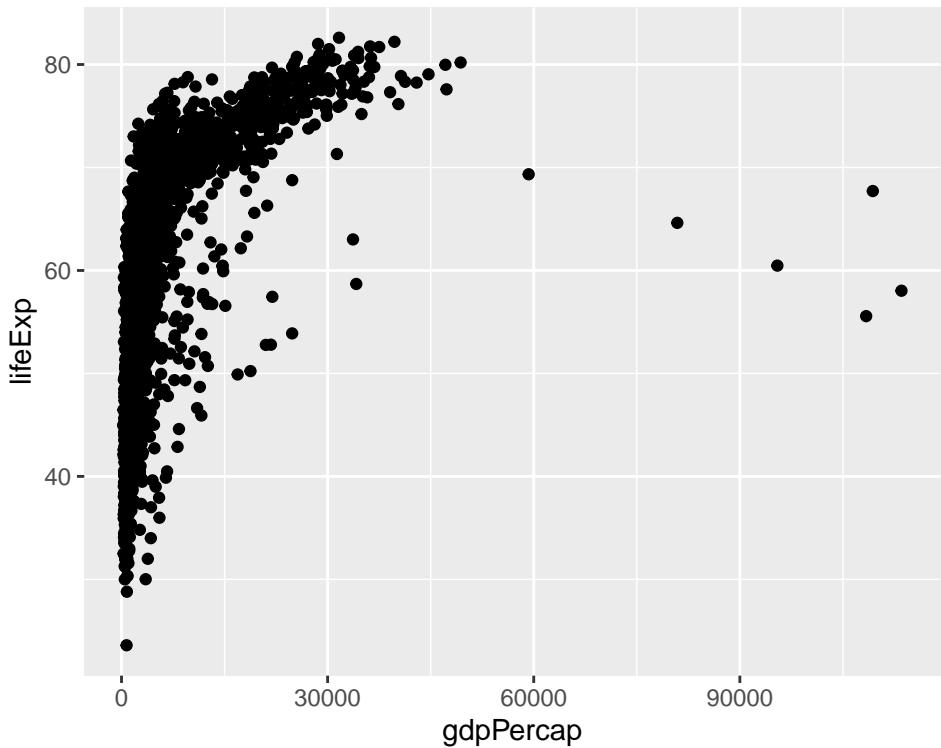
Load the gapminder data package.

```
# install.packages("gapminder")
library(gapminder)
gapminder

## # A tibble: 1,704 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.
## 7 Afghanistan Asia     1982    39.9  12881816     978.
## 8 Afghanistan Asia     1987    40.8  13867957     852.
## 9 Afghanistan Asia     1992    41.7  16317921     649.
## 10 Afghanistan Asia    1997    41.8  22227415     635.
## # ... with 1,694 more rows
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) # nothing to plot yet!
```



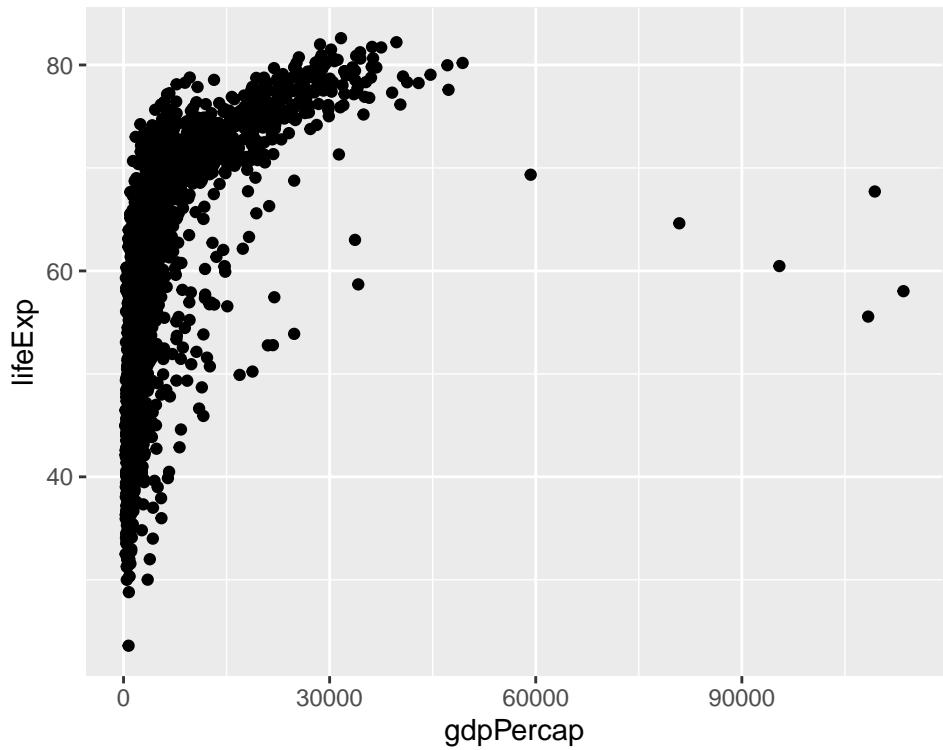
```
ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) +  
  geom_point()
```



```
p <-  
  ggplot(data = gapminder, aes(x = gdpPercap, y = lifeExp)) # just initializes
```

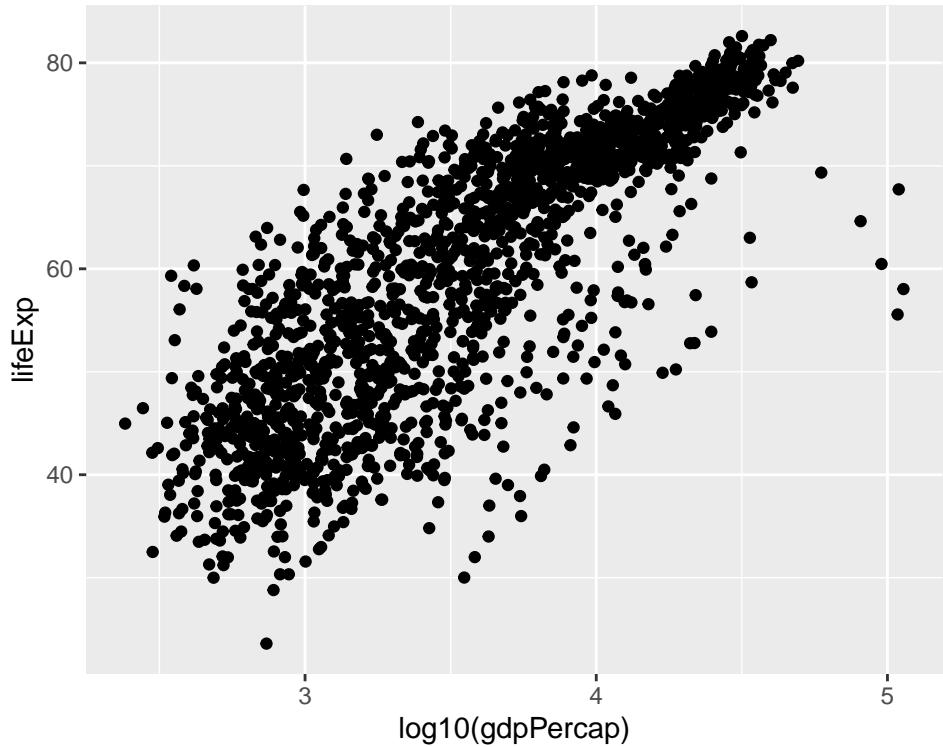
scatterplot

```
p + geom_point()
```



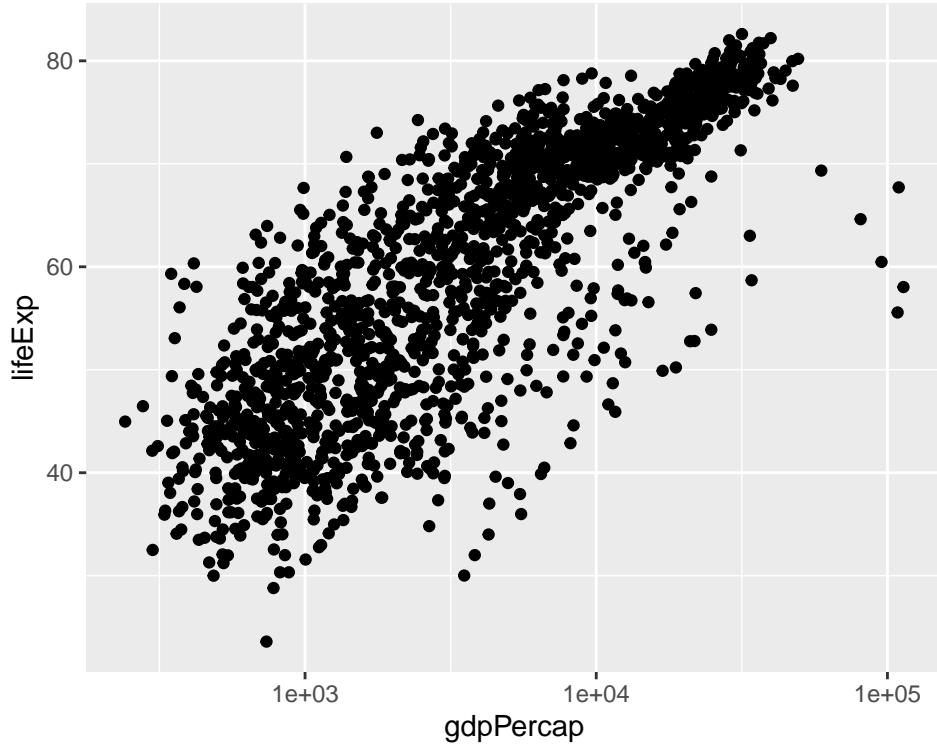
log transformation ... quick and dirty

```
ggplot(gapminder, aes(x = log10(gdpPerCap), y = lifeExp)) +  
  geom_point()
```



a better way to log transform

```
p + geom_point() + scale_x_log10()
```



let's make that stick

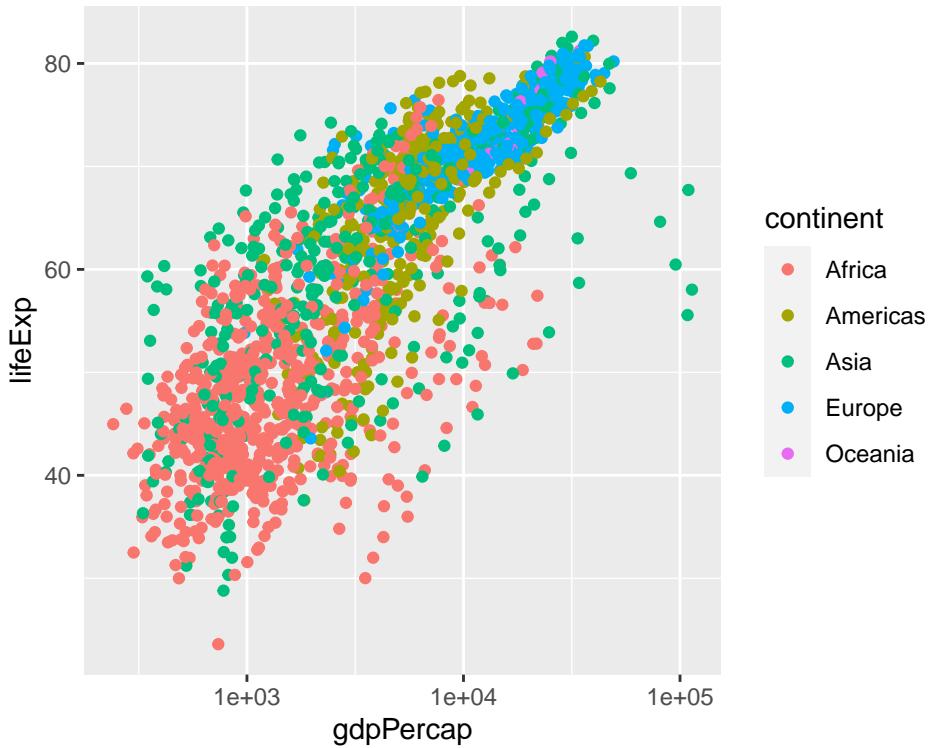
```
p <- p + scale_x_log10()
```

common workflow: gradually build up the plot you want

re-define the object 'p' as you develop "keeper" commands

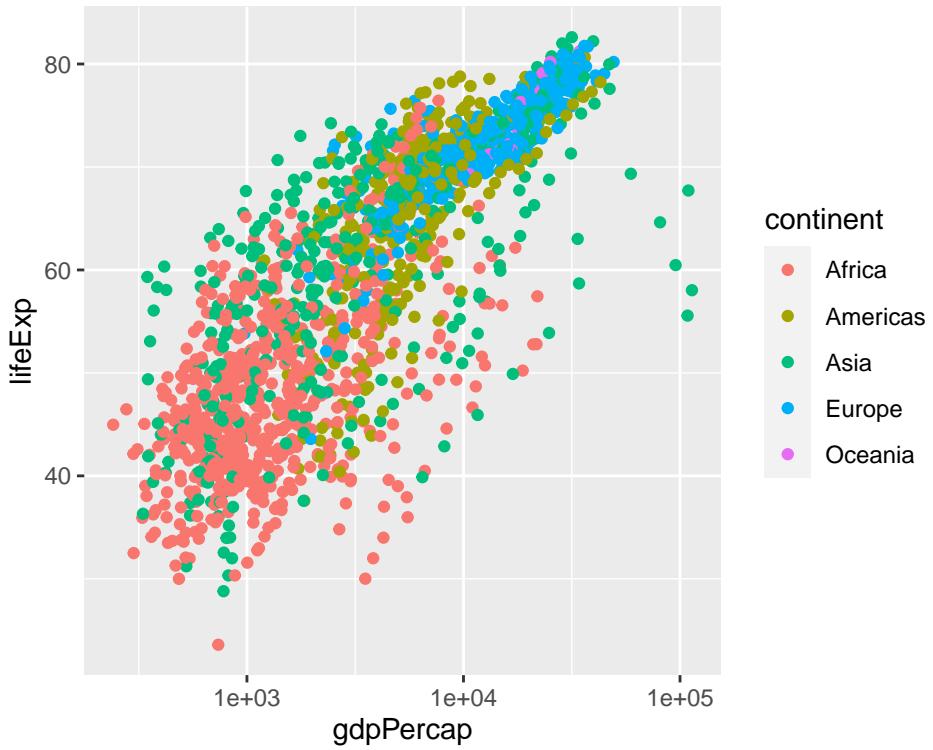
convey continent by color: MAP continent variable to aesthetic color

```
p + geom_point(aes(color = continent))
```



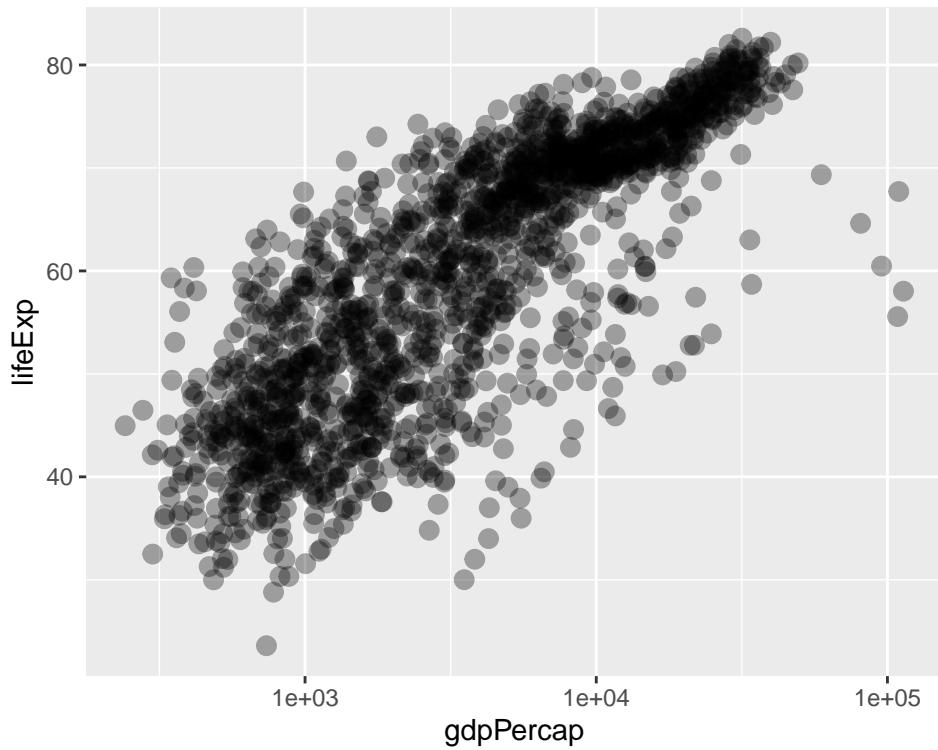
```
## add summary(p)!

ggplot(gapminder, aes(x = gdpPercap, y = lifeExp, color = continent)) +
  geom_point() + scale_x_log10() # in full detail, up to now
```



address overplotting: SET alpha transparency and size to a value

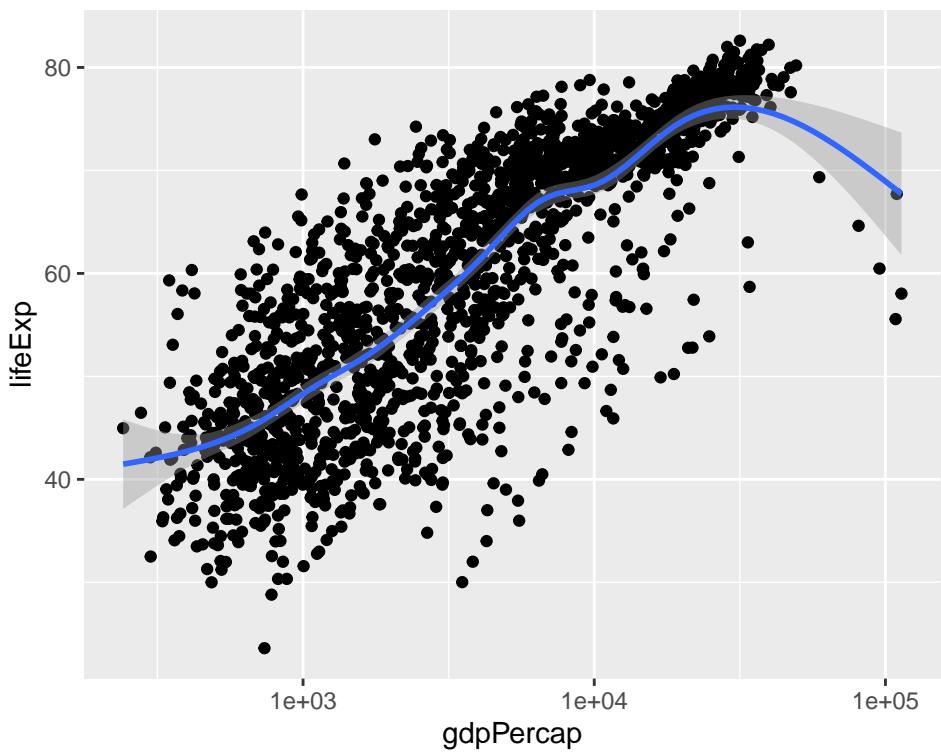
```
p + geom_point(alpha = (1/3), size = 3)
```



add a fitted curve or line. “lm” fit s a linear model line

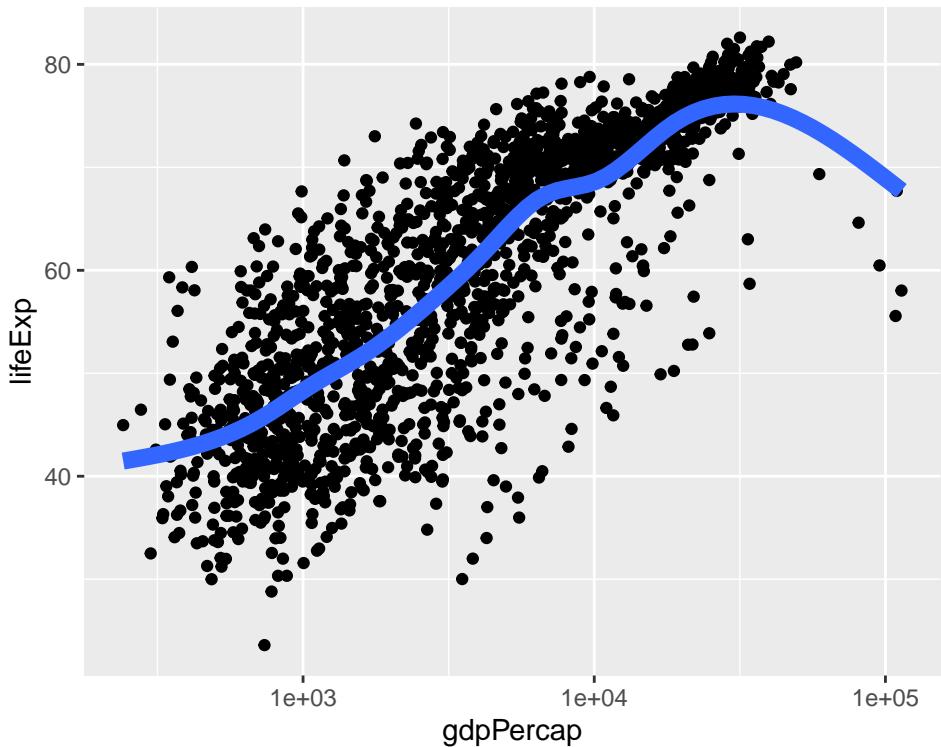
```
p + geom_point() + geom_smooth()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



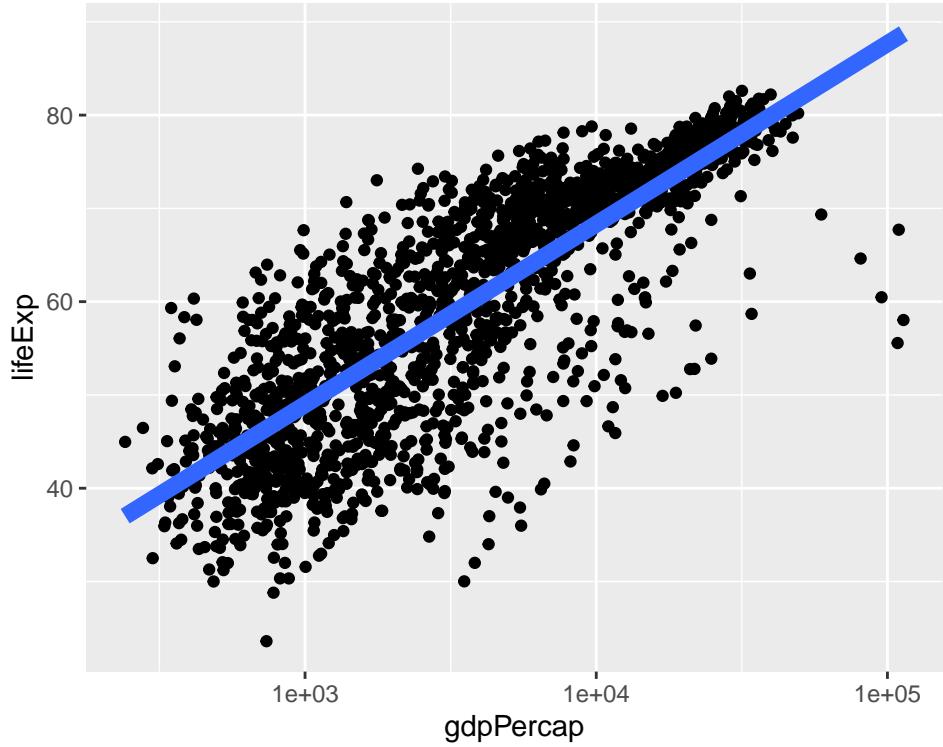
```
p + geom_point() + geom_smooth(lwd = 3, se = FALSE)
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



```
p + geom_point() + geom_smooth(lwd = 3, se = FALSE, method = "lm")
```

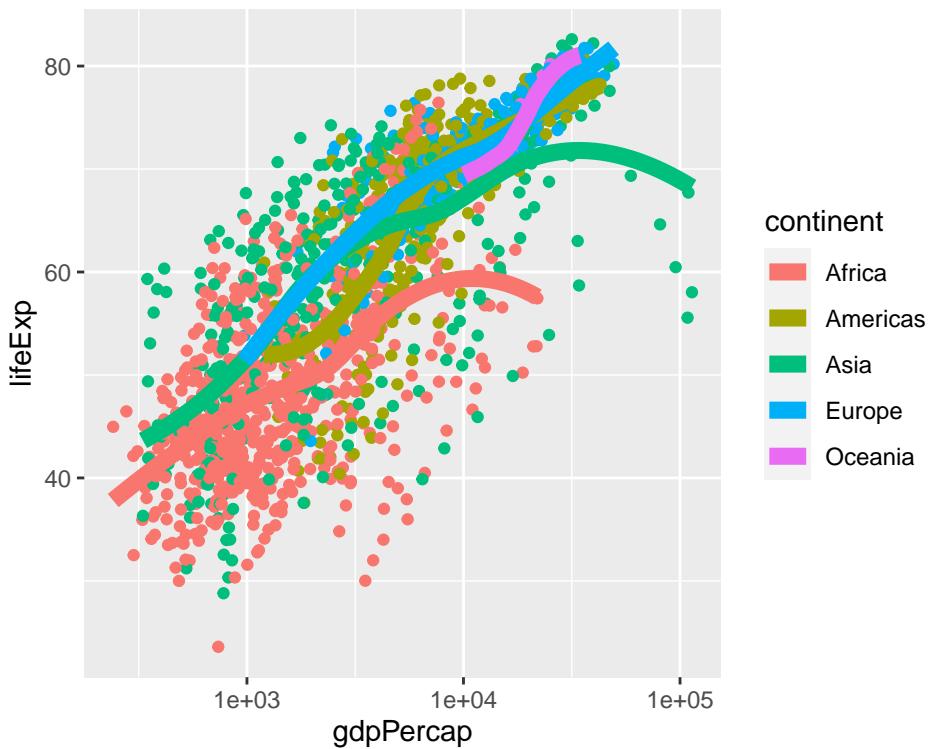
```
## `geom_smooth()` using formula 'y ~ x'
```



revive our interest in continents!

```
p + aes(color = continent) + geom_point() +
  geom_smooth(lwd = 3, se = FALSE)

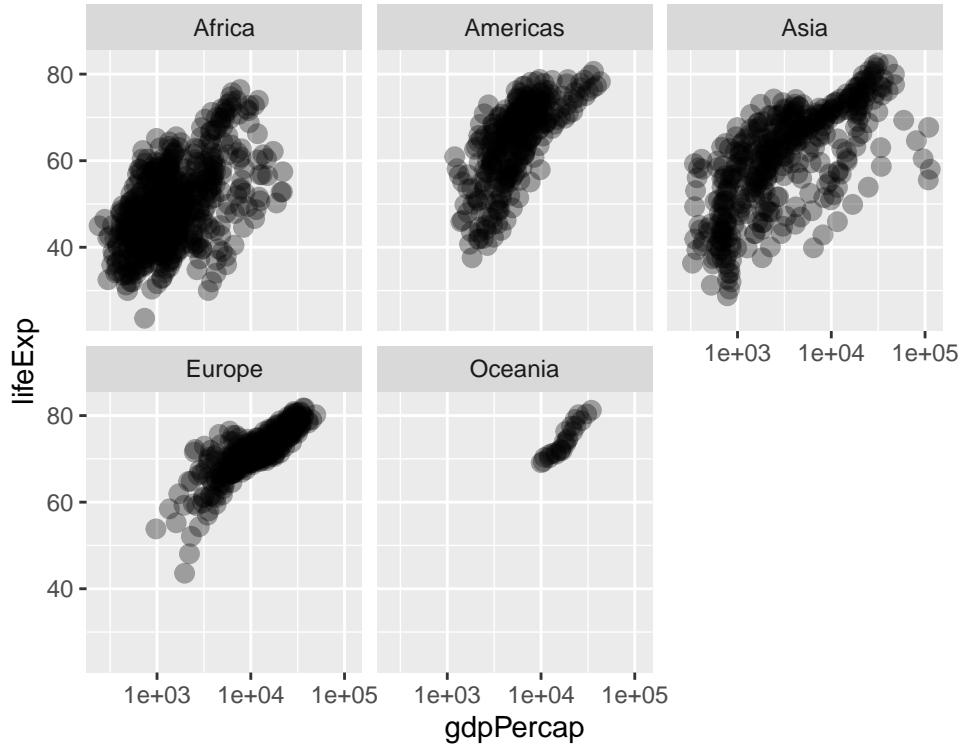
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



```
# GGPlot Facetting is a very useful higher order visualization
```

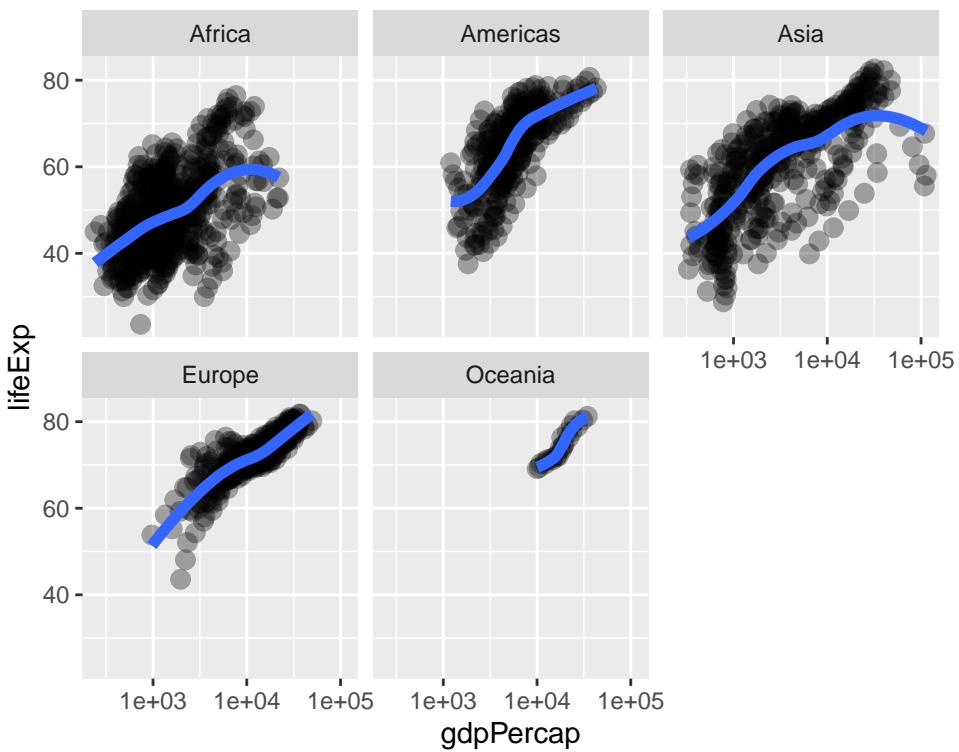
facetting: another way to exploit a factor

```
p + geom_point(alpha = (1 / 3), size = 3) +  
  facet_wrap(~ continent)
```



```
p + geom_point(alpha = (1 / 3), size = 3) +  
  facet_wrap(~ continent) +  
  geom_smooth(lwd = 2, se = FALSE)
```

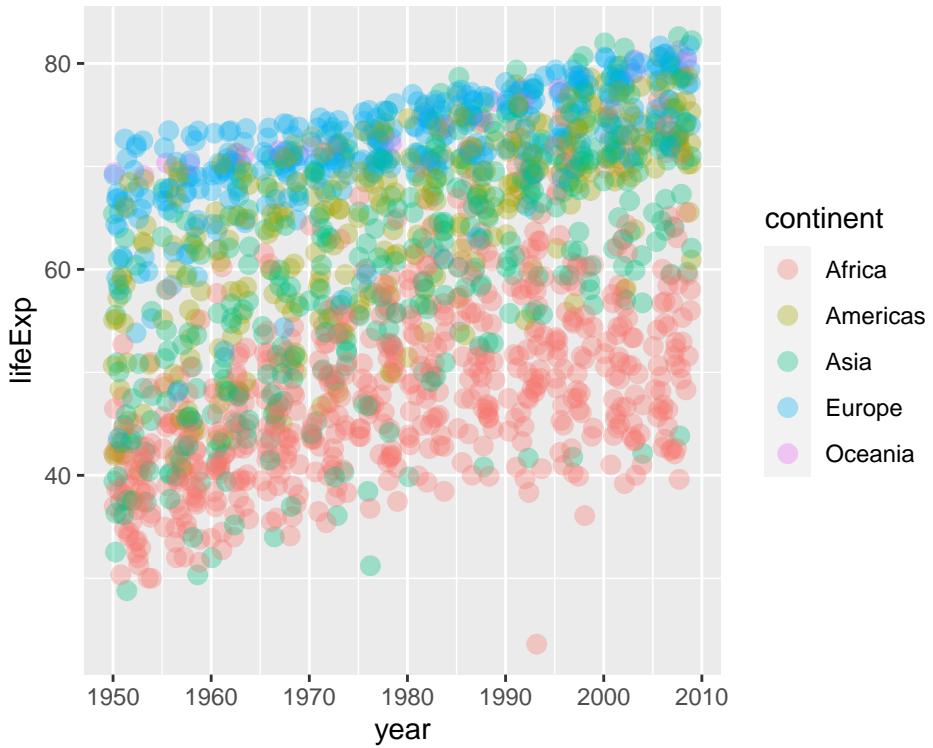
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



## Exercises:

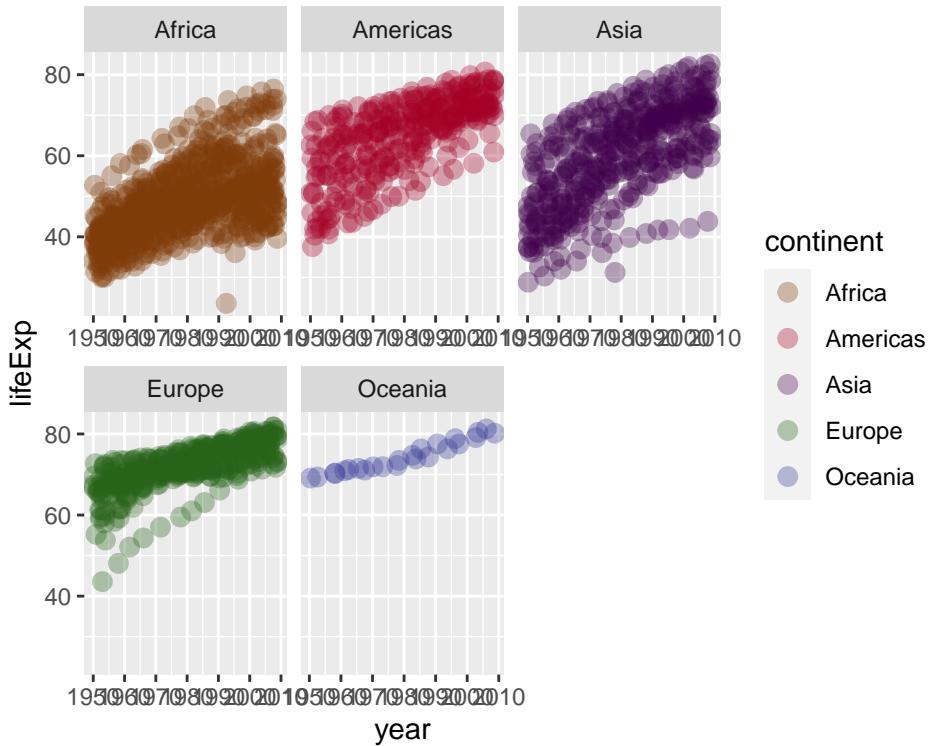
- plot lifeExp against year

```
ggplot(gapminder, aes(x = year, y = lifeExp,
                      color = continent)) +
  geom_jitter(alpha = 1 / 3, size = 3)
```



- make mini-plots, split out by continent HINT: use facet\_wrap()

```
ggplot(gapminder, aes(x = year, y = lifeExp,
                      color = continent)) +
  facet_wrap(~ continent, scales = "free_x") +
  geom_jitter(alpha = 1 / 3, size = 3) +
  scale_color_manual(values = continent_colors)
```

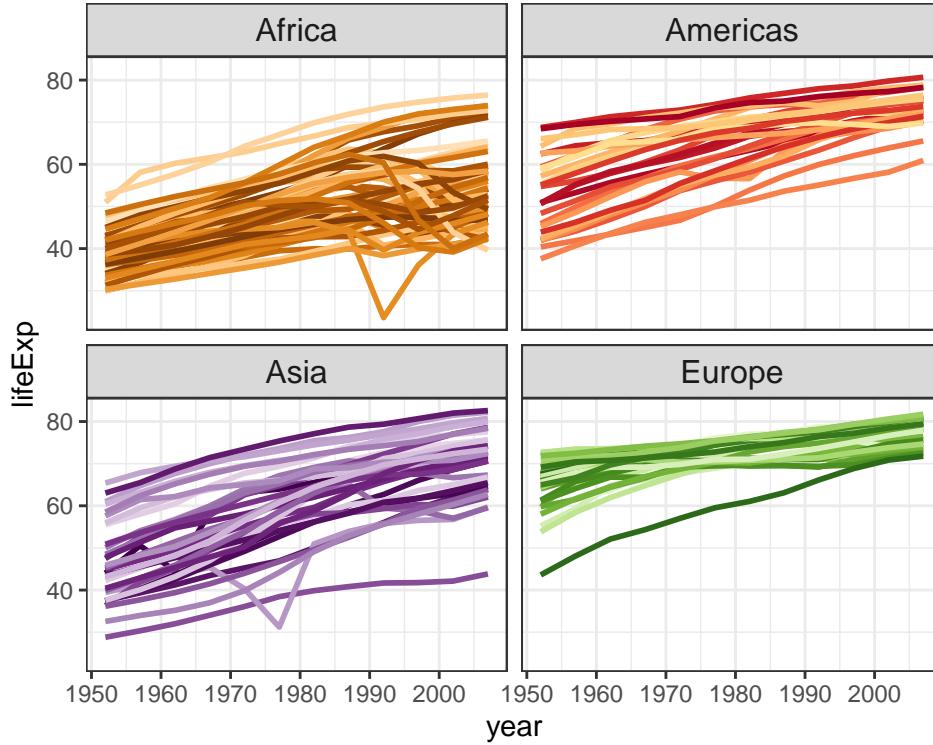


```

ggplot(
  subset(gapminder, continent != "Oceania"),
  aes(
    x = year,
    y = lifeExp,
    group = country,
    color = country
  )
) +
  geom_line(lwd = 1, show_guide = FALSE) + facet_wrap(~ continent) +
  scale_color_manual(values = country_colors) +
  #scale_color_brewer()+
  theme_bw() + theme(strip.text = element_text(size = rel(1.1)))

```

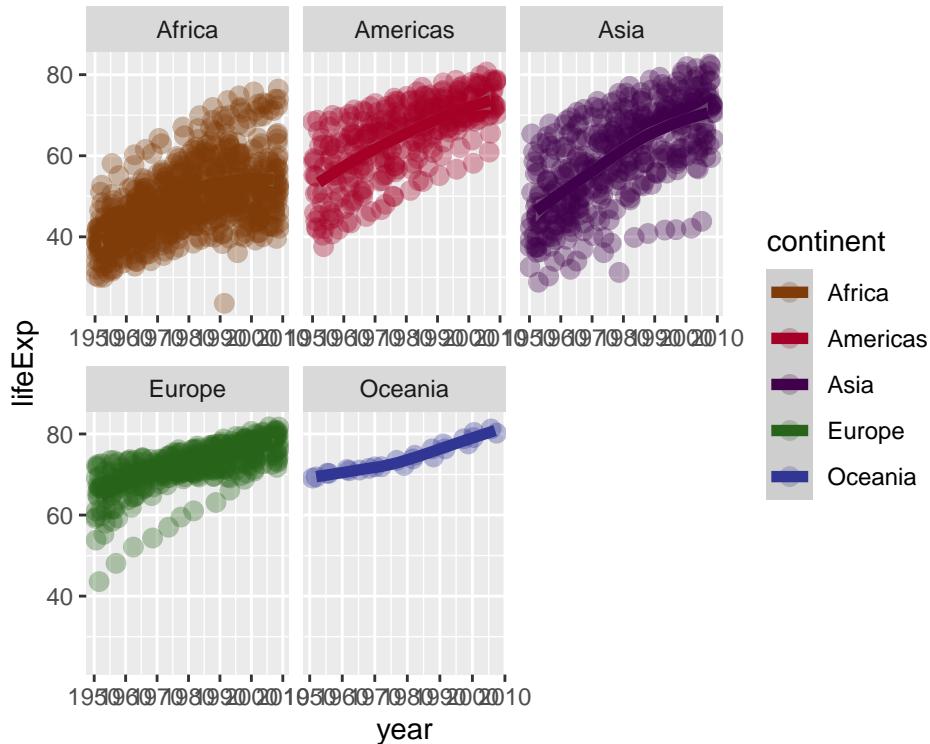
## Warning: `show\_guide` has been deprecated. Please use `show.legend` instead.



- add a fitted smooth and/or linear regression, w/ or w/o facetting

```
ggplot(gapminder, aes(x = year, y = lifeExp,
                      color = continent)) +
  facet_wrap(~ continent, scales = "free_x") +
  geom_jitter(alpha = 1 / 3, size = 3) +
  scale_color_manual(values = continent_colors) +
  geom_smooth(lwd = 2)

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



To make explicit what package a function comes from Use explicity naming by dplyr::filter()

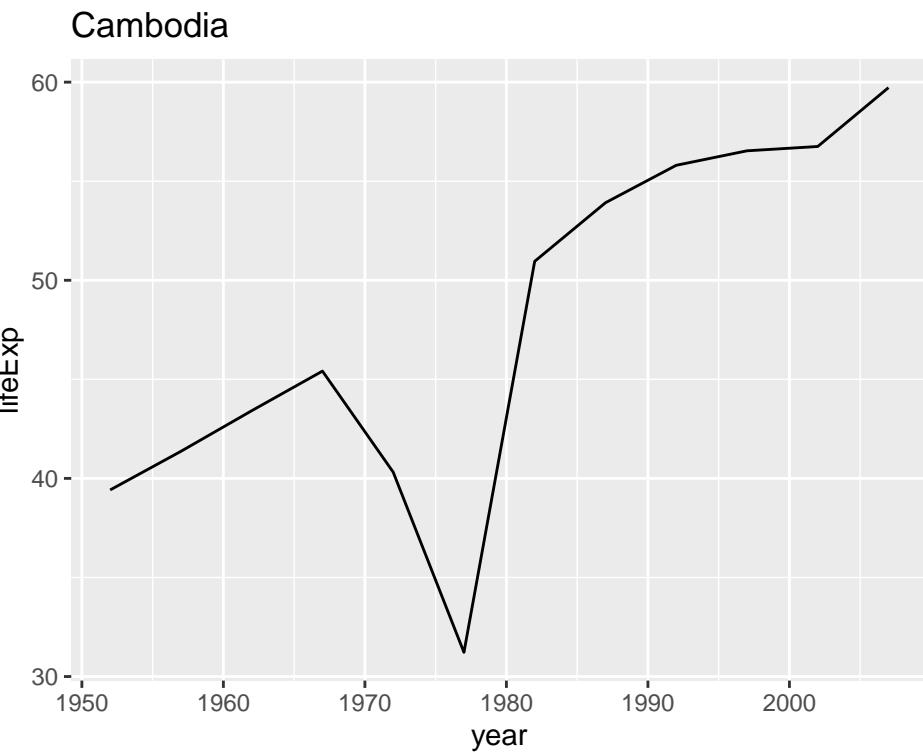
```
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'  
## The following objects are masked from 'package:stats':  
##  
##     filter, lag  
## The following objects are masked from 'package:base':  
##  
##     intersect, setdiff, setequal, union
```

- use dplyr::filter() to plot lifeExp against year for just one country or continent

```
jc <- "Cambodia"
```

```
gapminder %>%  
  filter(country == jc) %>%  
  ggplot(aes(x = year, y = lifeExp)) +  
  labs(title = jc) +  
  geom_line()
```



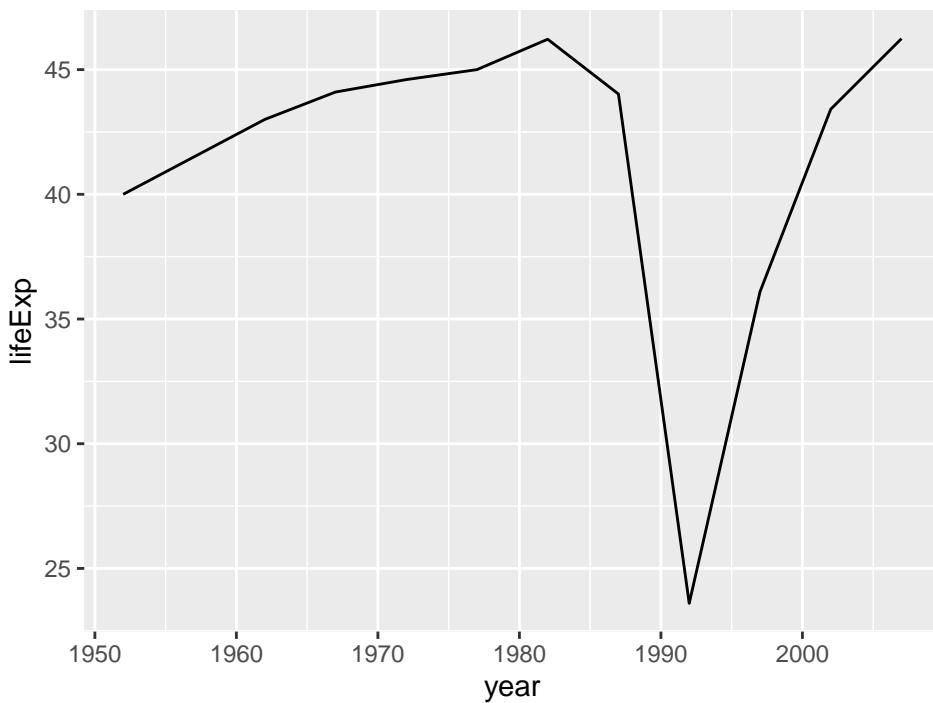
```
# [Khmer Rouge] (https://en.wikipedia.org/wiki/Khmer\_Rouge#Crimes\_against\_humanity)
```

```
rwanda <- gapminder %>%
  filter(country == "Rwanda")

p <- ggplot(rwanda, aes(x = year, y = lifeExp)) +
  labs(title = "Rwanda") +
  geom_line()

print(p)
```

## Rwanda



```
ggsave("rwanda.pdf")
```

```
## Saving 5 x 4 in image
```

```
ggsave("rwanda.pdf", plot = p)
```

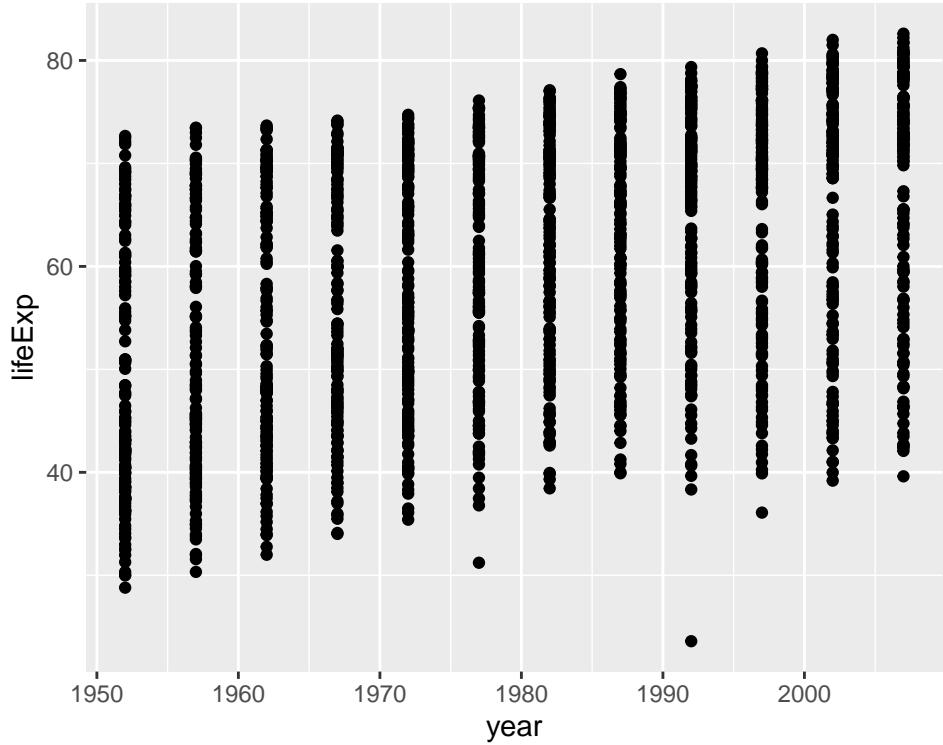
```
## Saving 5 x 4 in image
```

```
# https://en.wikipedia.org/wiki/Rwandan\_genocide
```

- other ideas?

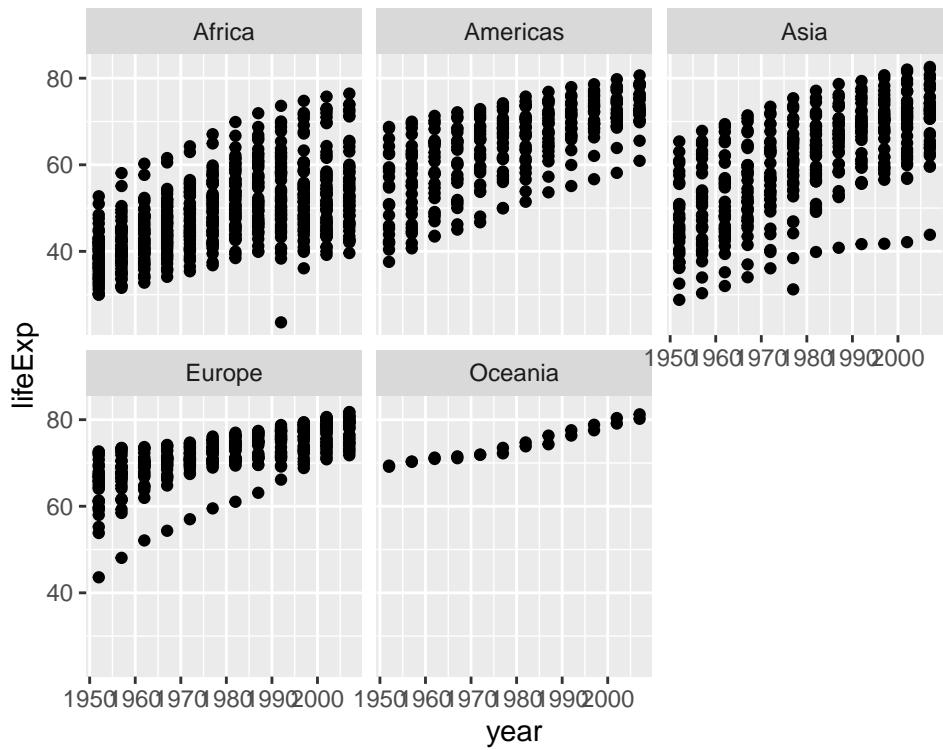
```
plot lifeExp against year
```

```
(y <- ggplot(gapminder, aes(x = year, y = lifeExp)) + geom_point())
```



make mini-plots, split out by continent

```
y + facet_wrap(~ continent)
```



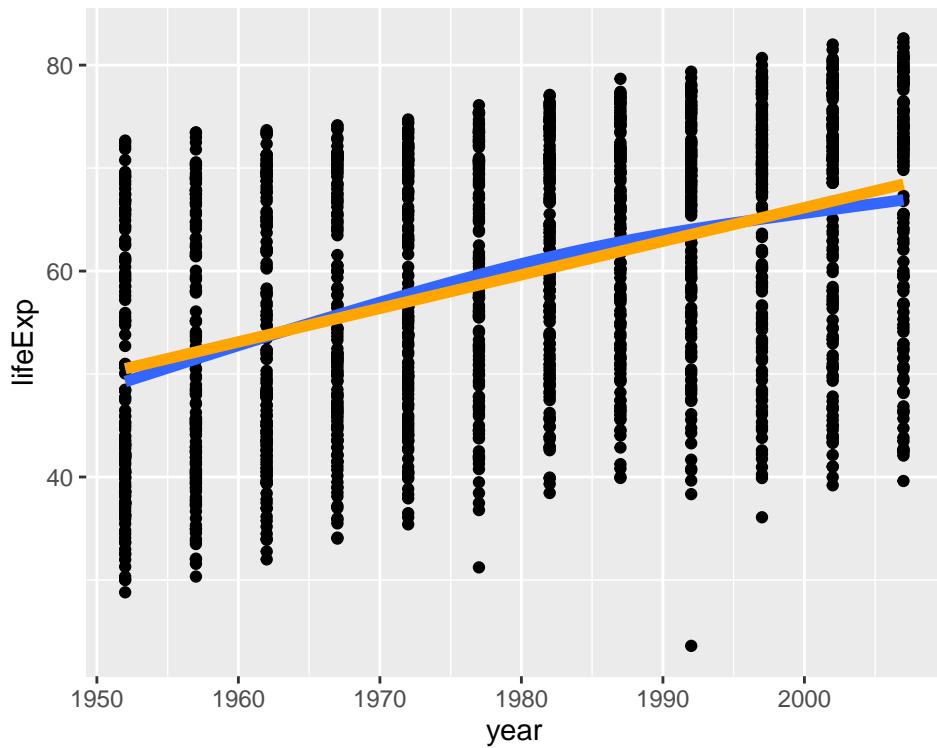
add a fitted smooth and/or linear regression, w/ or w/o facetting

```

y + geom_smooth(se = FALSE, lwd = 2) +
  geom_smooth(
    se = FALSE,
    method = "lm",
    color = "orange",
    lwd = 2
  )

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
## `geom_smooth()` using formula 'y ~ x'

```

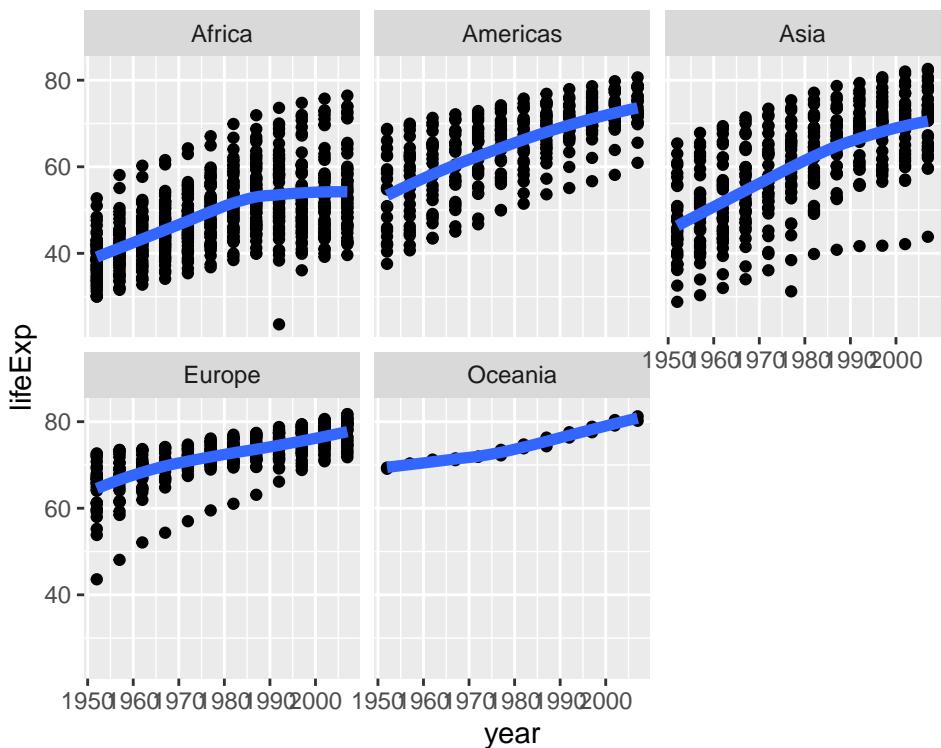


```

y + geom_smooth(se = FALSE, lwd = 2) +
  facet_wrap(~ continent)

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

```

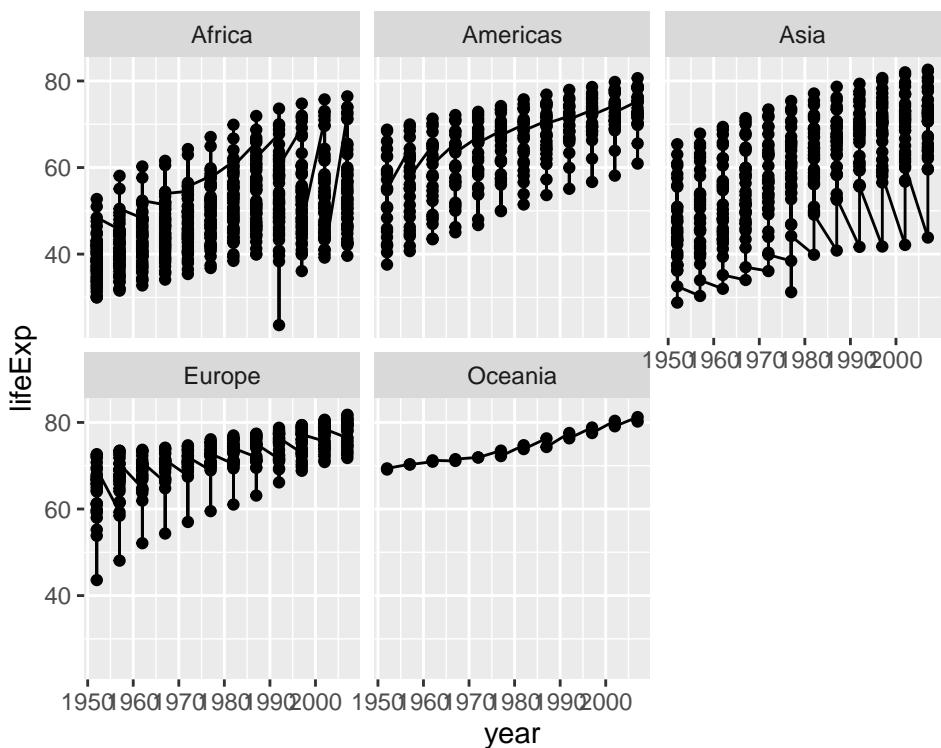


last bit on scatterplots

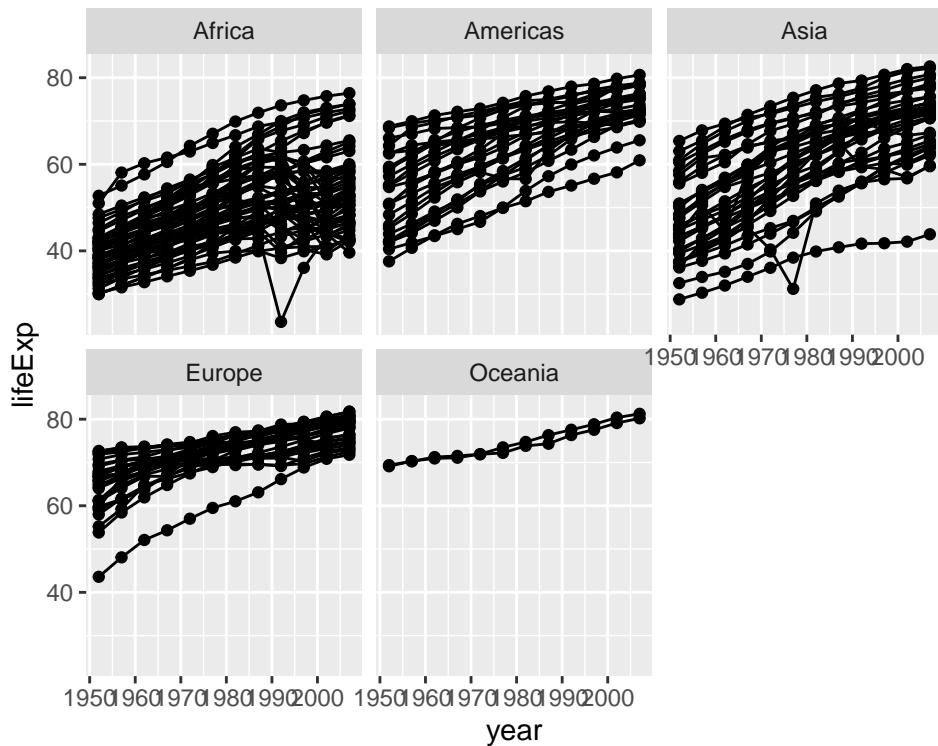
how can we “connect the dots” for one country?

i.e. make a spaghetti plot?

```
y + facet_wrap(~ continent) + geom_line() # uh, no
```

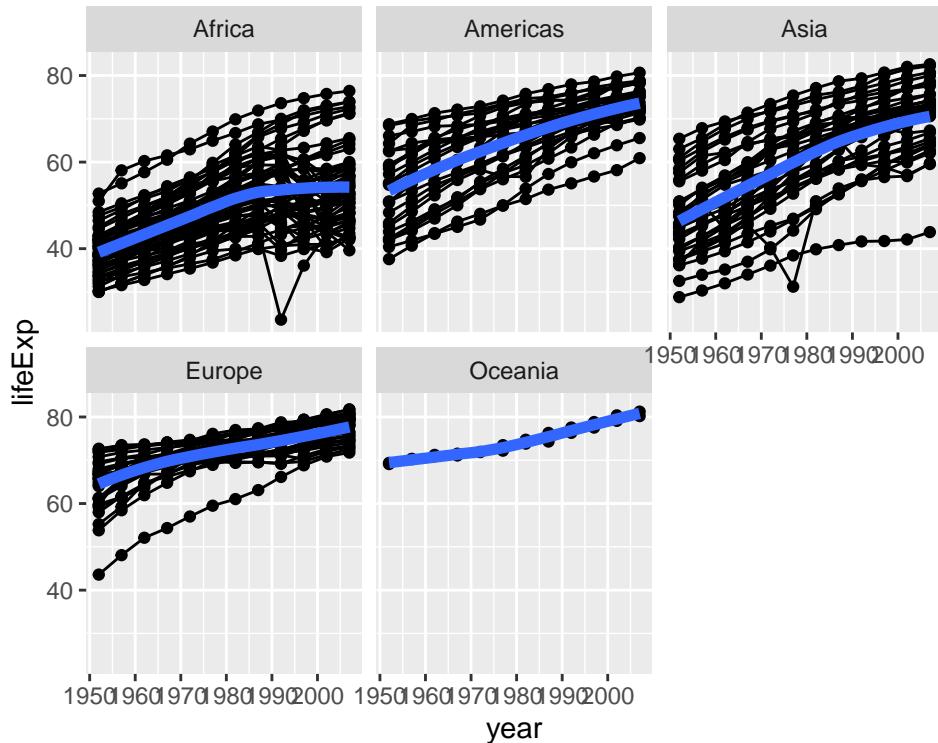


```
y + facet_wrap(~ continent) + geom_line(aes(group = country)) # yes!
```



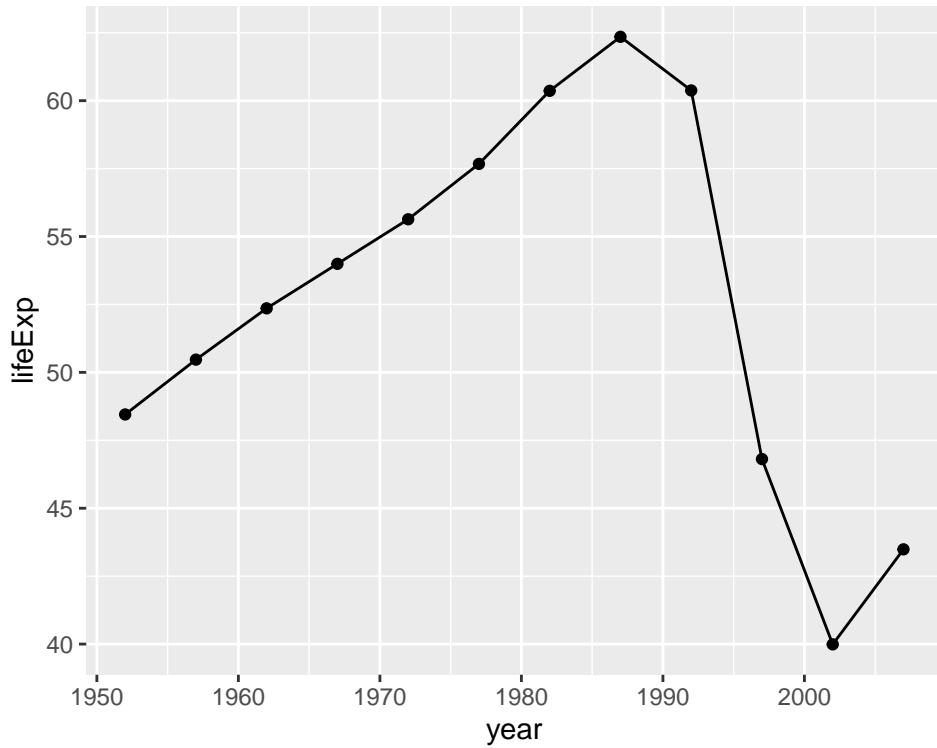
```
y + facet_wrap(~ continent) + geom_line(aes(group = country)) +  
geom_smooth(se = FALSE, lwd = 2)
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



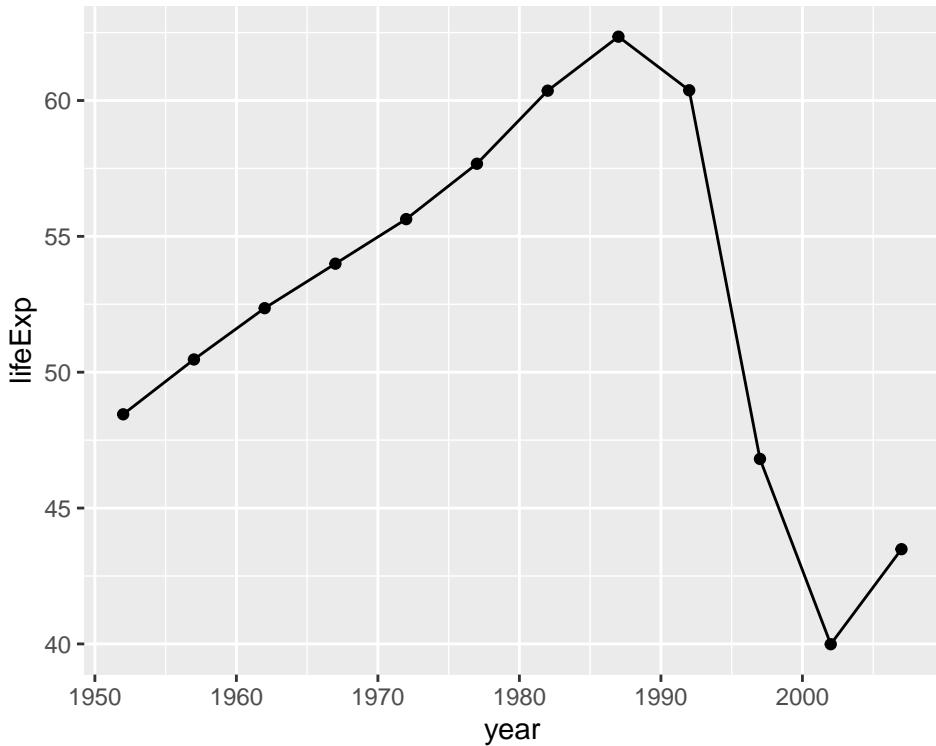
note about subsetting data sadly, `ggplot()` does not have a ‘subset =’ argument so do that ‘on the fly’ with `subset(..., subset = ...)`

```
ggplot(subset(gapminder, country == "Zimbabwe"),
       aes(x = year, y = lifeExp)) + geom_line() + geom_point()
```



or could do with `dplyr::filter`

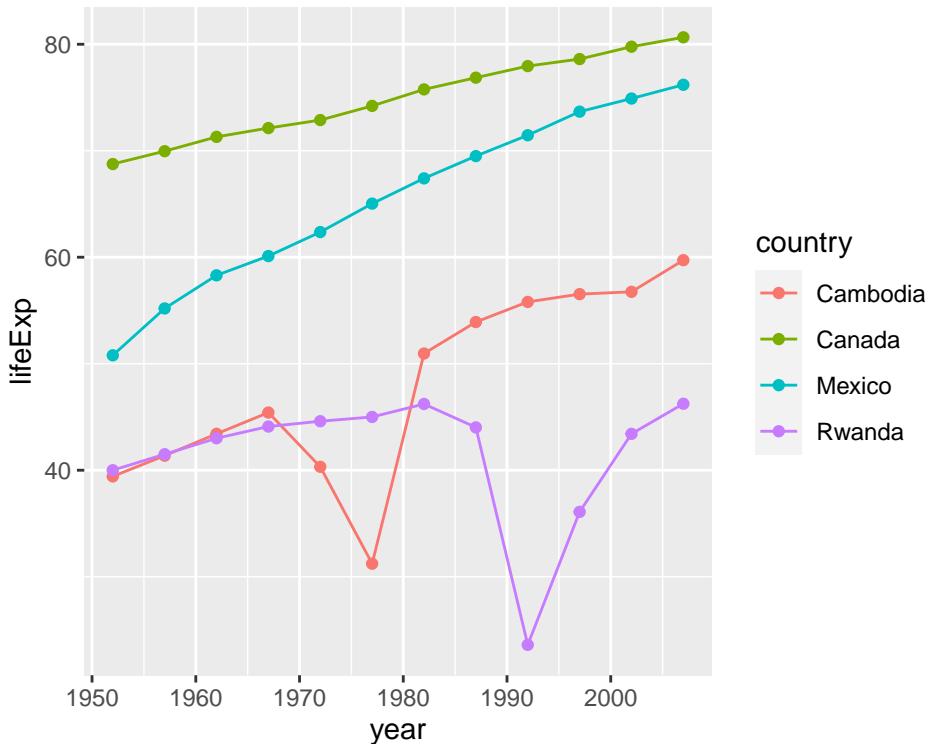
```
ggplot(gapminder %>% filter(country == "Zimbabwe"),
       aes(x = year, y = lifeExp)) + geom_line() + geom_point()
```



let just look at four countries

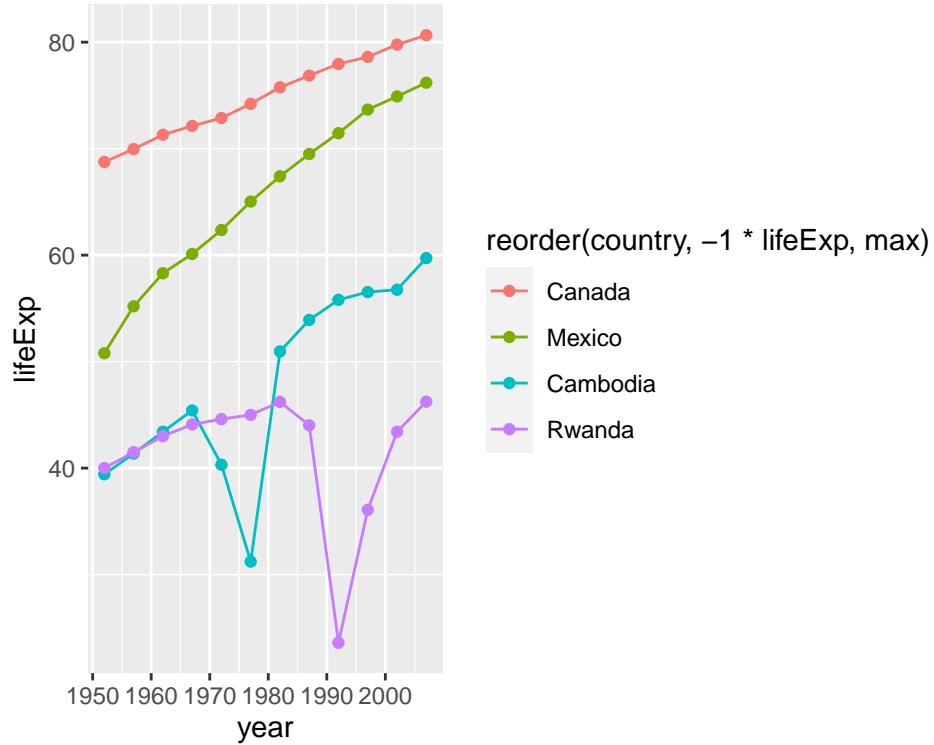
```
jCountries <- c("Canada", "Rwanda", "Cambodia", "Mexico")

ggplot(subset(gapminder, country %in% jCountries),
       aes(x = year, y = lifeExp, color = country)) + geom_line() + geom_point()
```



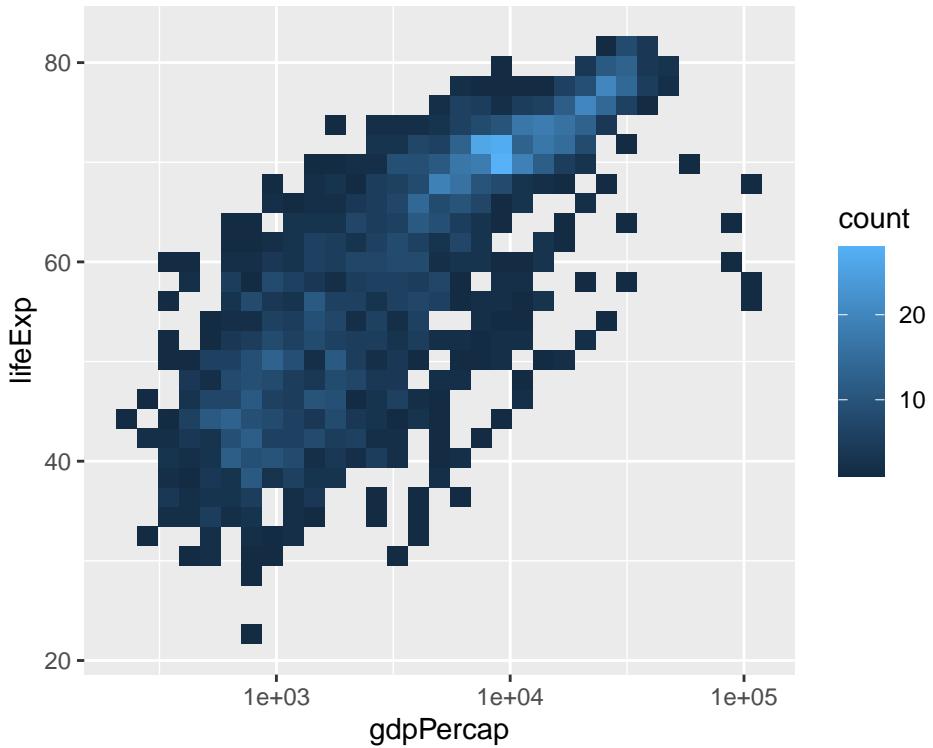
when you really care, make your legend easy to navigate  
 this means visual order = data order = factor level order

```
ggplot(subset(gapminder, country %in% jCountries),
       aes(
         x = year,
         y = lifeExp,
         color = reorder(country, -1 * lifeExp, max)
       )) +
  geom_line() + geom_point()
```



another approach to overplotting ggplot(gapminder, aes(x = gdpPercap, y = lifeExp)) +

```
ggplot(gapminder, aes(x = gdpPercap, y = lifeExp)) +
  scale_x_log10() + geom_bin2d()
```



```
# Check what's loaded in your session

sessionInfo()

## R version 4.2.1 (2022-06-23)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.5 LTS
##
## Matrix products: default
## BLAS:    /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK:  /usr/lib/x86_64-linux-gnu/openblas-pthread/liblapack.so.3
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8          LC_NUMERIC=C
## [3] LC_TIME=en_US.UTF-8          LC_COLLATE=en_US.UTF-8
## [5] LC_MONETARY=en_US.UTF-8       LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=en_US.UTF-8          LC_NAME=C
## [9] LC_ADDRESS=C                  LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8   LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats      graphics   grDevices utils      datasets   methods    base
##
## other attached packages:
## [1] dplyr_1.0.10    gapminder_0.3.0 ggplot2_3.3.6   tibble_3.1.7
## [5] knitr_1.40
##
## loaded via a namespace (and not attached):
## [1] pillar_1.8.1     compiler_4.2.1   highr_0.9      tools_4.2.1
## [5] digest_0.6.29    lattice_0.20-45 nlme_3.1-159   evaluate_0.16
```

```

## [ 9] lifecycle_1.0.2    gtable_0.3.1      mgcv_1.8-40     pkgconfig_2.0.3
## [13] rlang_1.0.5        Matrix_1.5-0     cli_3.4.0       DBI_1.1.3
## [17] rstudioapi_0.14    yaml_2.3.5       xfun_0.32       fastmap_1.1.0
## [21] withr_2.5.0        stringr_1.4.1   generics_0.1.3 vctrs_0.4.1
## [25] grid_4.2.1         tidyselect_1.1.2 glue_1.6.2       R6_2.5.1
## [29] fansi_1.0.3        rmarkdown_2.16   farver_2.1.1   purrr_0.3.4
## [33] magrittr_2.0.3     splines_4.2.1   scales_1.2.1   ellipsis_0.3.2
## [37] htmltools_0.5.3    assertthat_0.2.1 colorspace_2.0-3 labeling_0.4.2
## [41] utf8_1.2.2         stringi_1.7.8   munsell_0.5.0

#=====
# Stripplots
#=====

#' ---
#' author: "Jenny Bryan"
#' output:
#'   html_document:
#'     keep_md: TRUE
#'   ---

#+ setup, include = FALSE

library(knitr)

?knitr

opts_chunk$set(fig.path = 'figs/bryan/striplot-', error = TRUE)

```

Note: this HTML is made by applying `knitr::spin()` to an R script. So the narrative is very minimal.

```
library(ggplot2)
```

pick a way to load the data

```
#gdURL <- "http://tiny.cc/gapminder"  
#gapminder <- read.delim(file = gdURL)  
#gapminder <- read.delim("gapminderDataFiveYear.tsv")
```

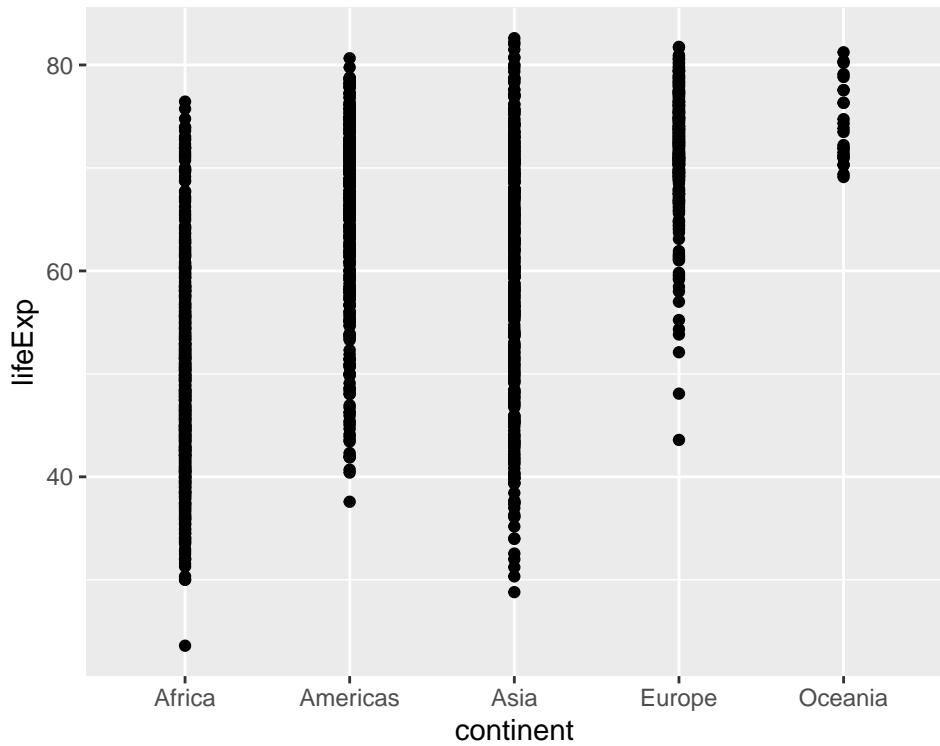
```
library(gapminder)
```

```
str(gapminder)

### tibble [1,704 x 6] (S3: tbl_df/tbl/data.frame)
### $ country : Factor w/ 142 levels "Afghanistan",..
### $ continent: Factor w/ 5 levels "Africa","America"
### $ year     : int [1:1704] 1952 1957 1962 1967 197...
### $ lifeExp  : num [1:1704] 28.8 30.3 32 34 36.1 ...
### $ pop      : int [1:1704] 8425333 9240934 1026708...
### $ gdpPercap: num [1:1704] 779 821 853 836 740 ...
```

stripplots: univariate scatterplots (but w/ ways to also convey 1+ factors)

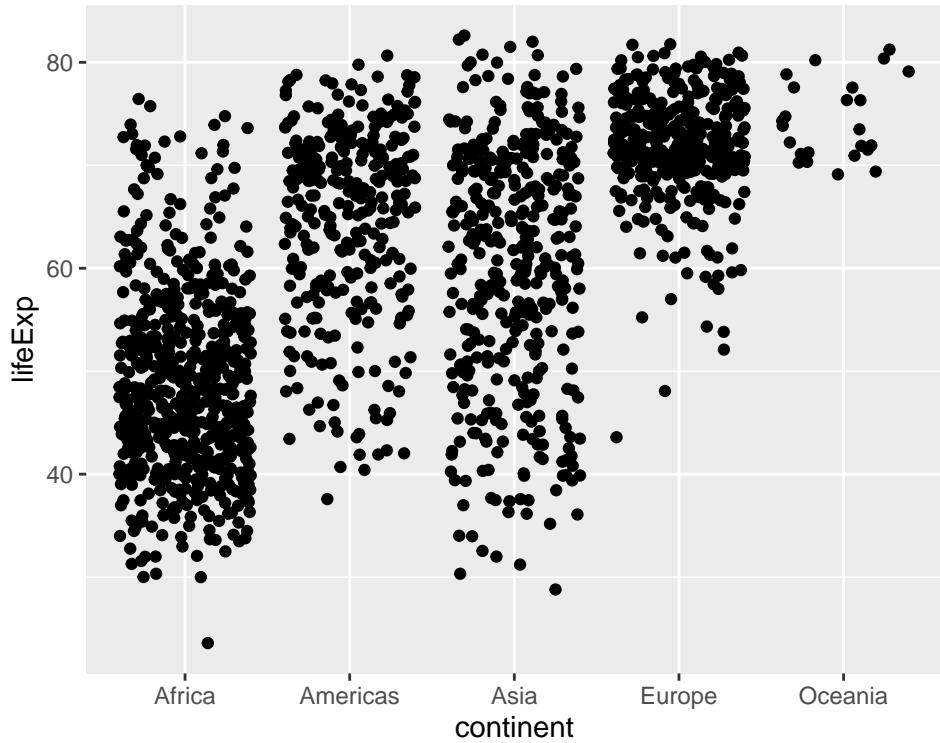
```
ggplot(gapminder, aes(x = continent, y = lifeExp)) + geom_point()
```



```
# Adding Jitter to a categorical variable plot can help visualization
```

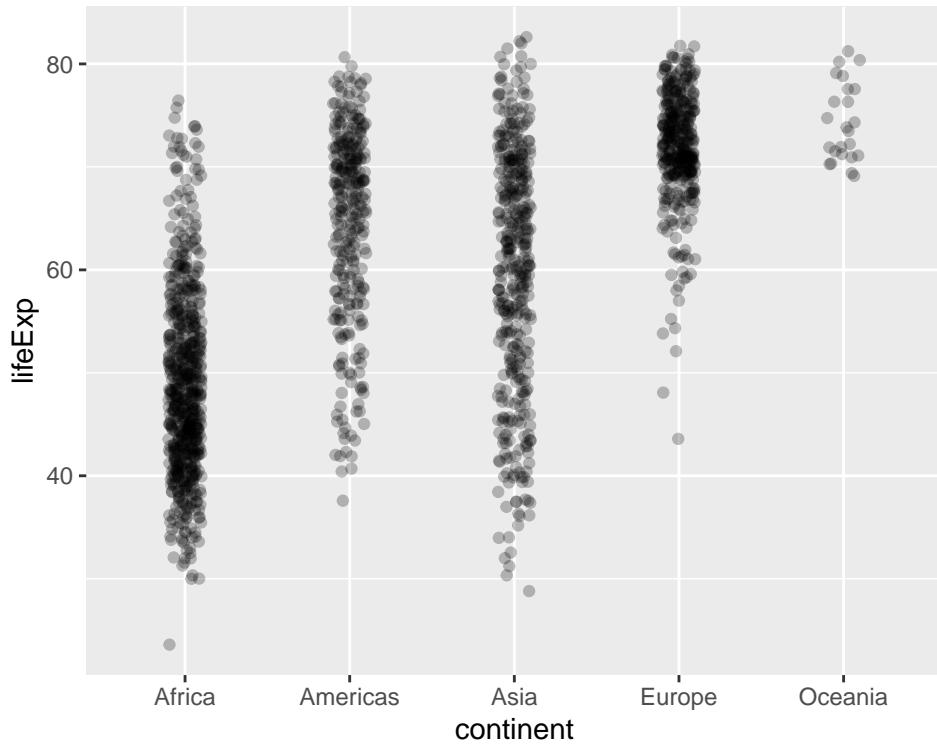
we have an overplotting problem; need to spread things out

```
ggplot(gapminder, aes(x = continent, y = lifeExp)) + geom_jitter()
```



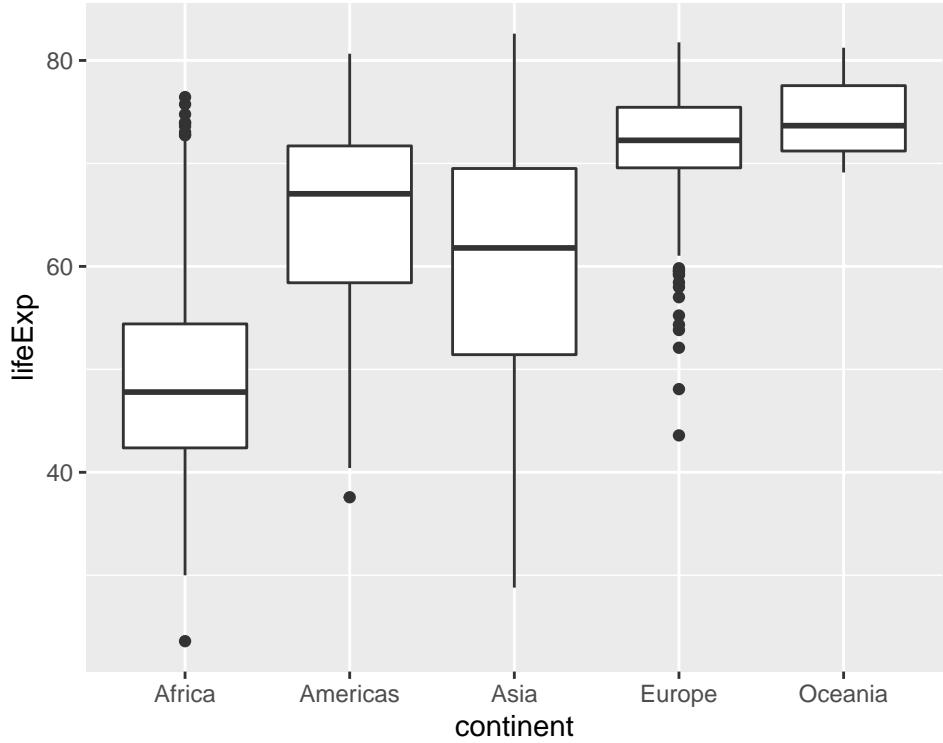
we can have less jitter in x, no jitter in y, more alpha transparency

```
ggplot(gapminder, aes(x = continent, y = lifeExp)) +  
  geom_jitter(position_jitter(width = 0.1, height = 0), alpha = 1 /  
  4)
```



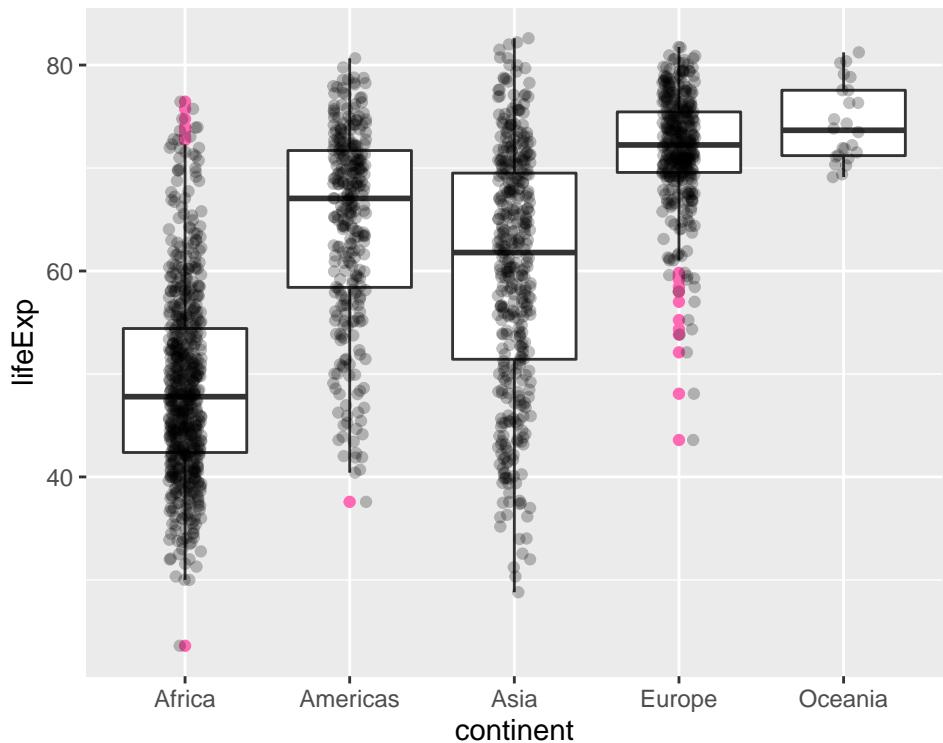
boxplots – covered properly elsewhere

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



raw data AND boxplots

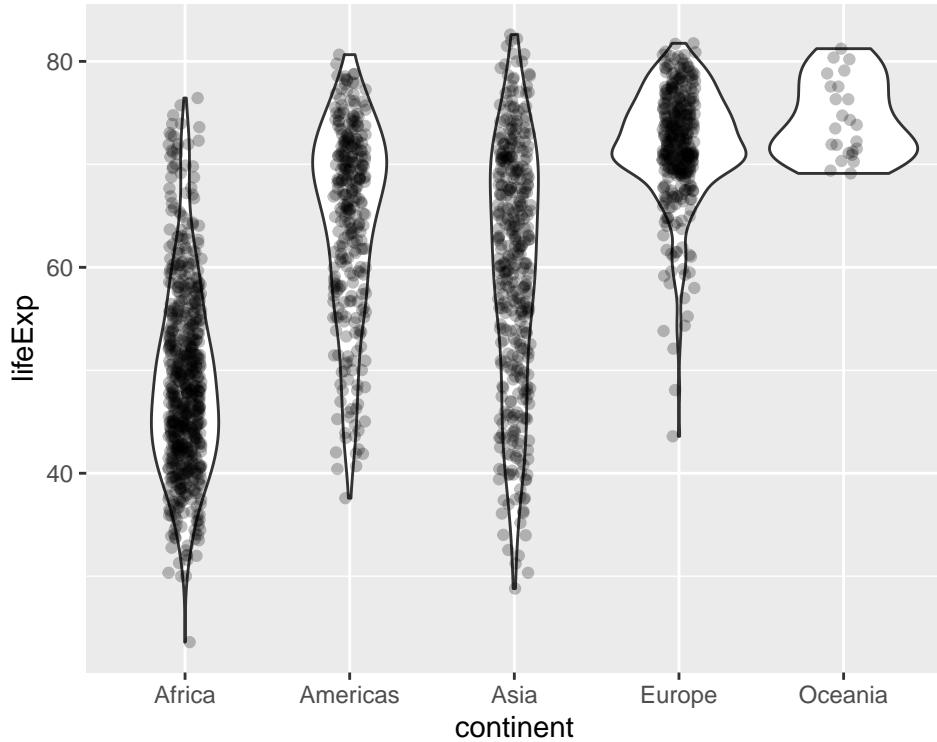
```
ggplot(gapminder, aes(x = continent, y = lifeExp)) +
  geom_boxplot(outlier.colour = "hotpink") +
  geom_jitter(position = position_jitter(width = 0.1, height = 0), alpha = 1 / 4)
```



```
#' raw data AND violinplots

ggplot(gapminder, aes(x = continent, y = lifeExp)) +
  geom_violin(outlier.colour = "hotpink") +
  geom_jitter(position = position_jitter(width = 0.1, height = 0), alpha = 1 / 4)

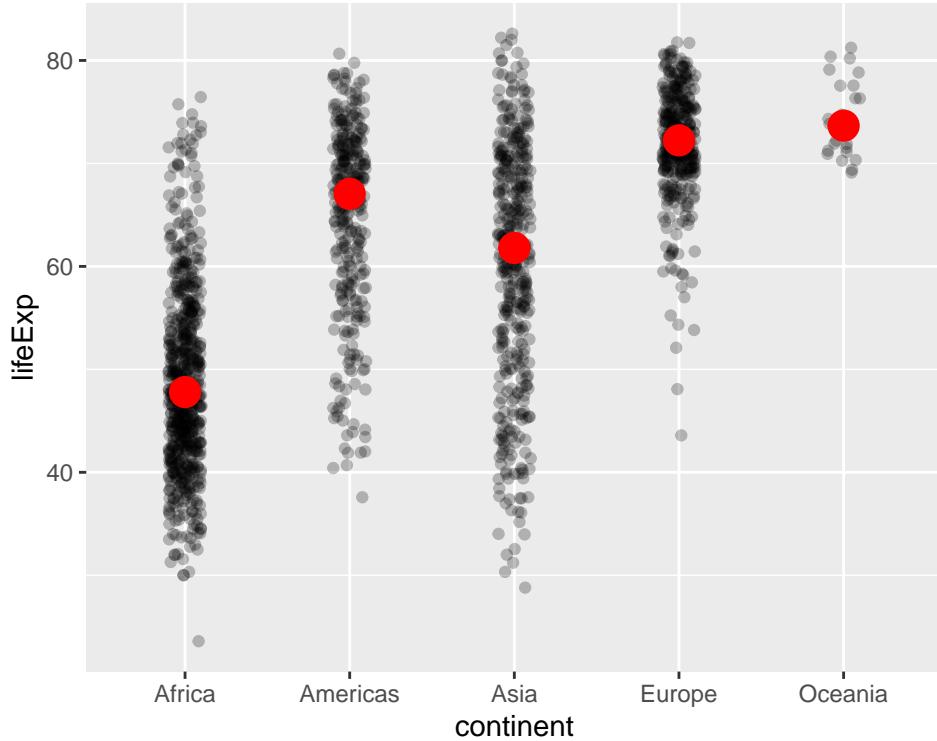
## Warning: Ignoring unknown parameters: outlier.colour
```



superpose a statistical summary

```
ggplot(gapminder, aes(x = continent, y = lifeExp)) +
  geom_jitter(position = position_jitter(width = 0.1), alpha = 1 / 4) +
  stat_summary(
    fun.y = median,
    colour = "red",
    geom = "point",
    size = 5
  )

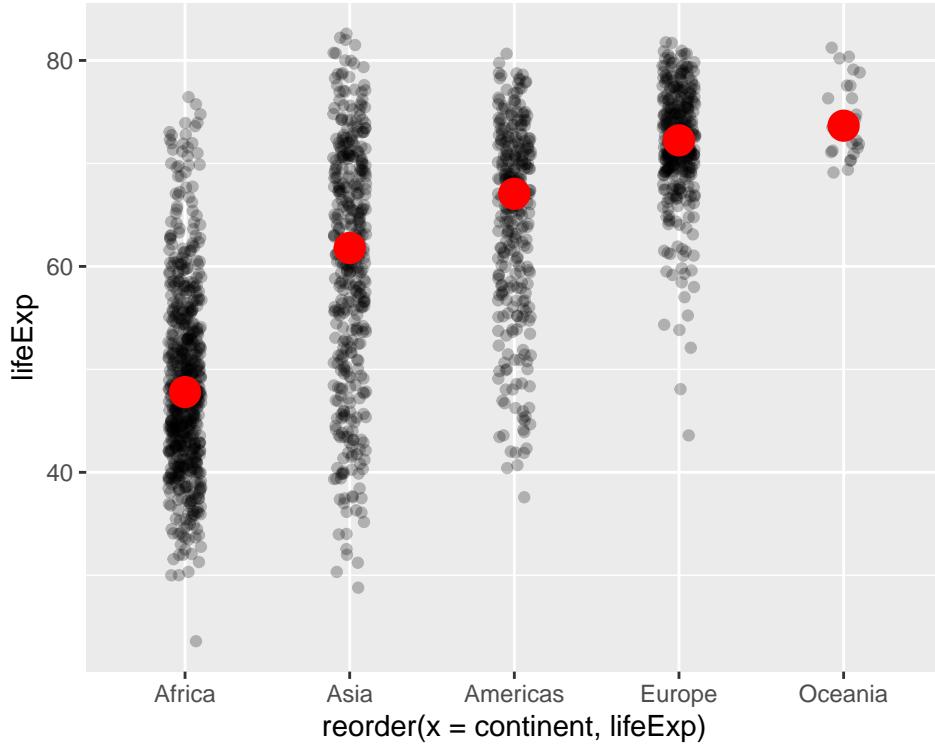
## Warning: `fun.y` is deprecated. Use `fun` instead.
```



let's reorder the continent factor based on lifeExp

```
ggplot(gapminder, aes(reorder(x = continent, lifeExp), y = lifeExp)) +
  geom_jitter(position = position_jitter(width = 0.1), alpha = 1 / 4) +
  stat_summary(
    fun.y = median,
    colour = "red",
    geom = "point",
    size = 5
  )
```

## Warning: `fun.y` is deprecated. Use `fun` instead.



```

sessionInfo()

## R version 4.2.1 (2022-06-23)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.5 LTS
##
## Matrix products: default
## BLAS:    /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK:  /usr/lib/x86_64-linux-gnu/openblas-pthread/liblapack.so.3
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8          LC_NUMERIC=C
## [3] LC_TIME=en_US.UTF-8          LC_COLLATE=en_US.UTF-8
## [5] LC_MONETARY=en_US.UTF-8       LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=en_US.UTF-8          LC_NAME=C
## [9] LC_ADDRESS=C                  LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8   LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats      graphics   grDevices  utils      datasets   methods    base
##
## other attached packages:
## [1] dplyr_1.0.10    gapminder_0.3.0 ggplot2_3.3.6    tibble_3.1.7
## [5] knitr_1.40
##
## loaded via a namespace (and not attached):
## [1] pillar_1.8.1     compiler_4.2.1   highr_0.9      tools_4.2.1
## [5] digest_0.6.29    lattice_0.20-45 nlme_3.1-159    evaluate_0.16
## [9] lifecycle_1.0.2   gtable_0.3.1    mgcv_1.8-40    pkgconfig_2.0.3
## [13] rlang_1.0.5      Matrix_1.5-0    cli_3.4.0      DBI_1.1.3

```

```
## [17] rstanioapi_0.14    yaml_2.3.5      xfun_0.32       fastmap_1.1.0
## [21] withr_2.5.0        stringr_1.4.1   generics_0.1.3  vctrs_0.4.1
## [25] grid_4.2.1         tidyselect_1.1.2 glue_1.6.2      R6_2.5.1
## [29] fansi_1.0.3        rmarkdown_2.16   farver_2.1.1   purrr_0.3.4
## [33] magrittr_2.0.3     splines_4.2.1   scales_1.2.1   ellipsis_0.3.2
## [37] htmltools_0.5.3   assertthat_0.2.1 colorspace_2.0-3 labeling_0.4.2
## [41] utf8_1.2.2         stringi_1.7.8   munsell_0.5.0

#=====
# Exploring distribution of a quantitative variable
#=====

#' ---
#' author: "Jenny Bryan"
#' output: github_document
#' ---

##+ setup, include = FALSE

library(knitr)
opts_chunk$set(fig.path = 'figs/bryan/uni-quant-', error = TRUE)
```

Note: this is rendered by applying `knitr::spin()` to an R script. So the narrative is very minimal. load the data and `ggplot2` (part of the tidyverse)

```
library(tidyverse)
## -- Attaching packages ----- tidyverse 1.3.1 --
## v tidyverse 1.3.1     v stringr 1.4.1
## v readr   2.1.2      vforcats 0.5.2
## v purrr   0.3.4

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

?tidyverse

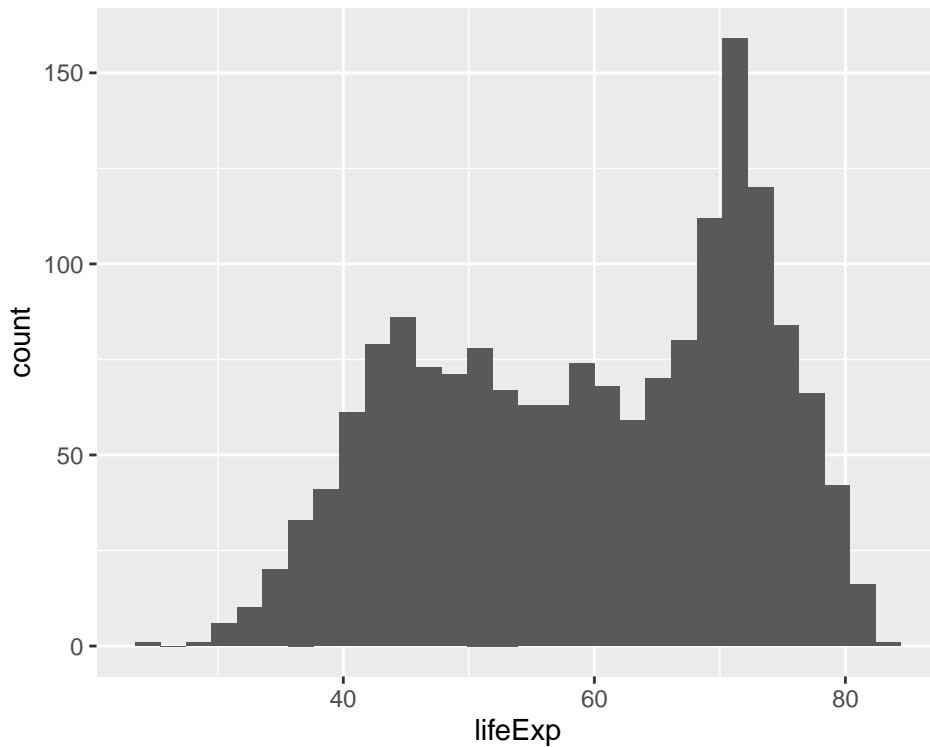
library(gapminder)

gapminder

## # A tibble: 1,704 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.
## 7 Afghanistan Asia     1982    39.9 12881816     978.
## 8 Afghanistan Asia     1987    40.8 13867957     852.
## 9 Afghanistan Asia     1992    41.7 16317921     649.
## 10 Afghanistan Asia    1997    41.8 22227415     635.
```

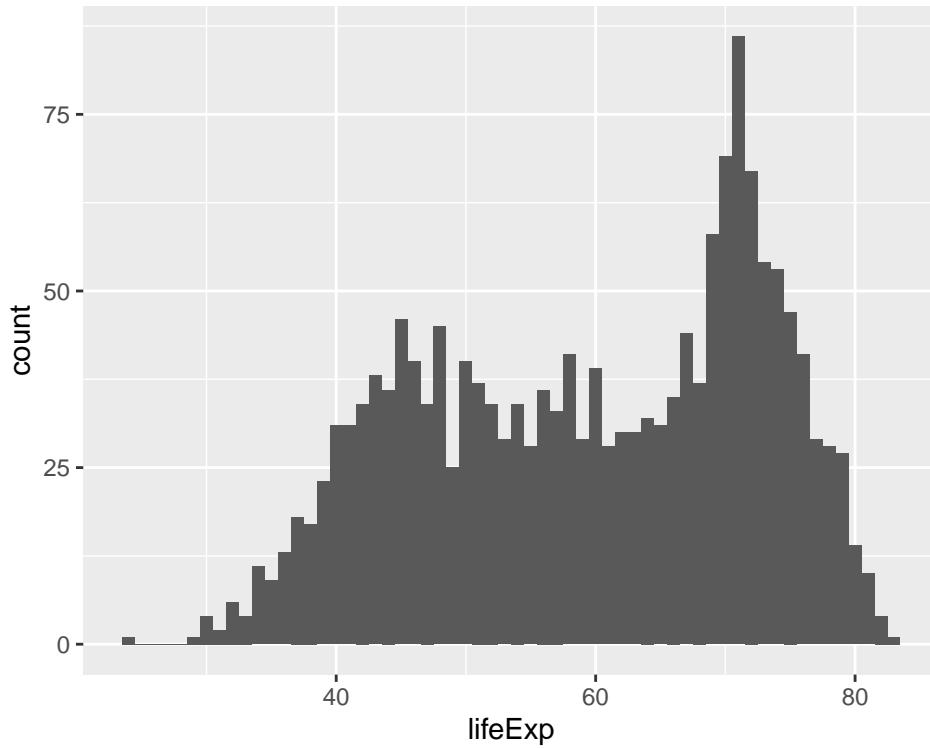
```
## # ... with 1,694 more rows
distribution of a quant var: histogram
ggplot(gapminder, aes(x = lifeExp)) +
  geom_histogram()

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



experiment with bin width; think in terms of the units of the x variable

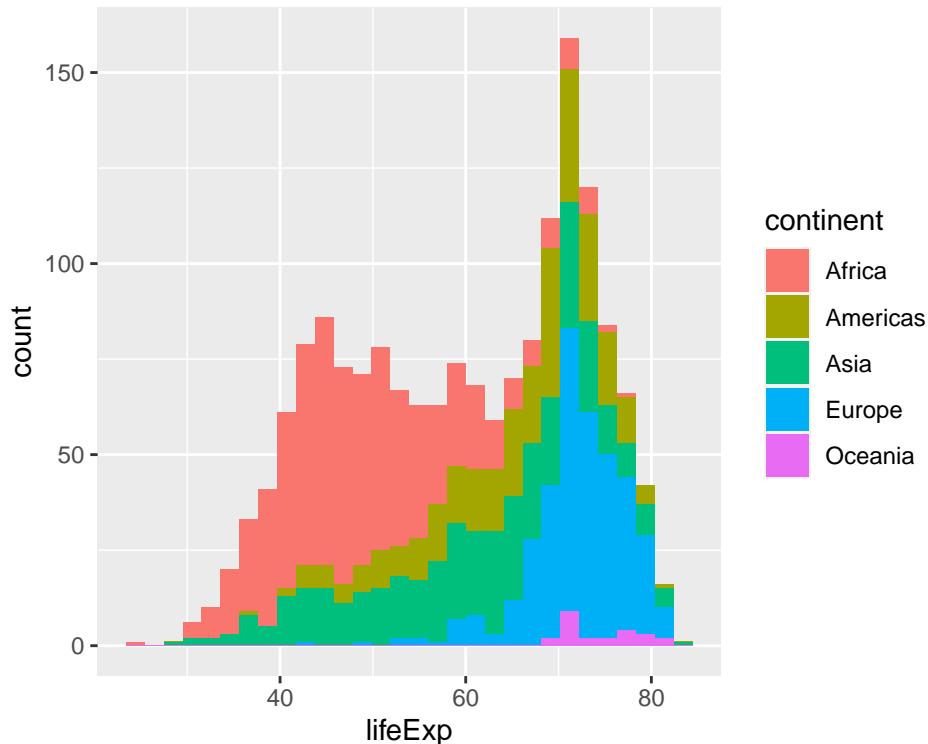
```
ggplot(gapminder, aes(x = lifeExp)) +
  geom_histogram(binwidth = 1)
```



show the different continents, but it's weird to stack up the histograms, which is what default of `position = "stack"` delivers

```
ggplot(gapminder, aes(x = lifeExp, fill = continent)) +
  geom_histogram()
```

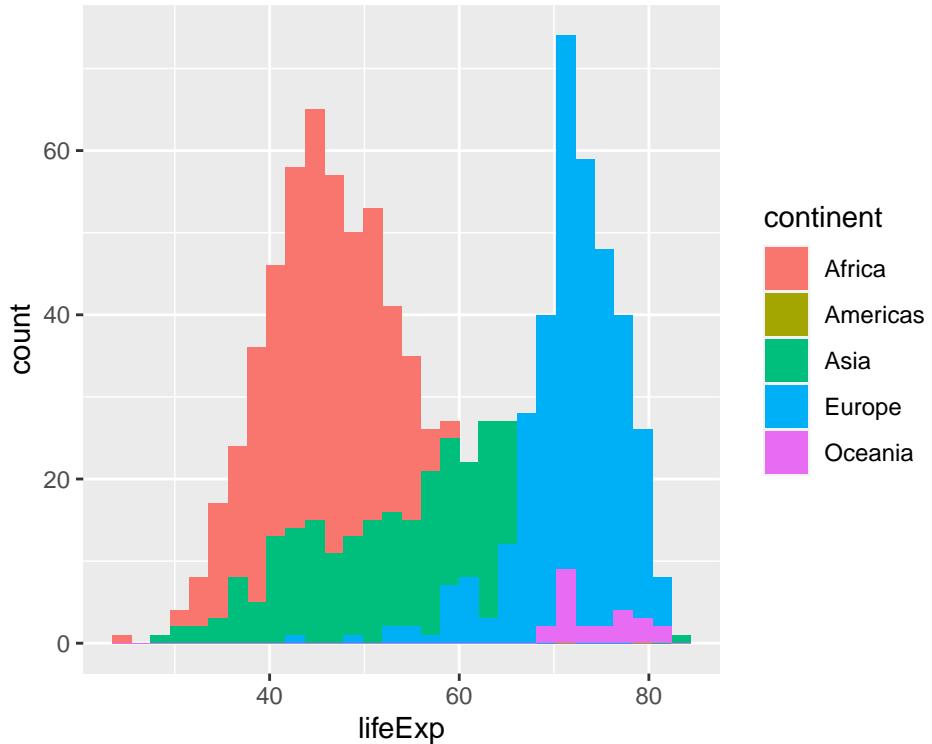
`## `stat_bin()` using `bins = 30` . Pick better value with `binwidth`.`



```
position = "identity" is good to know about it's still weird to layer them on top of each other like this
```

```
ggplot(gapminder, aes(x = lifeExp, fill = continent)) +  
  geom_histogram(position = "identity")
```

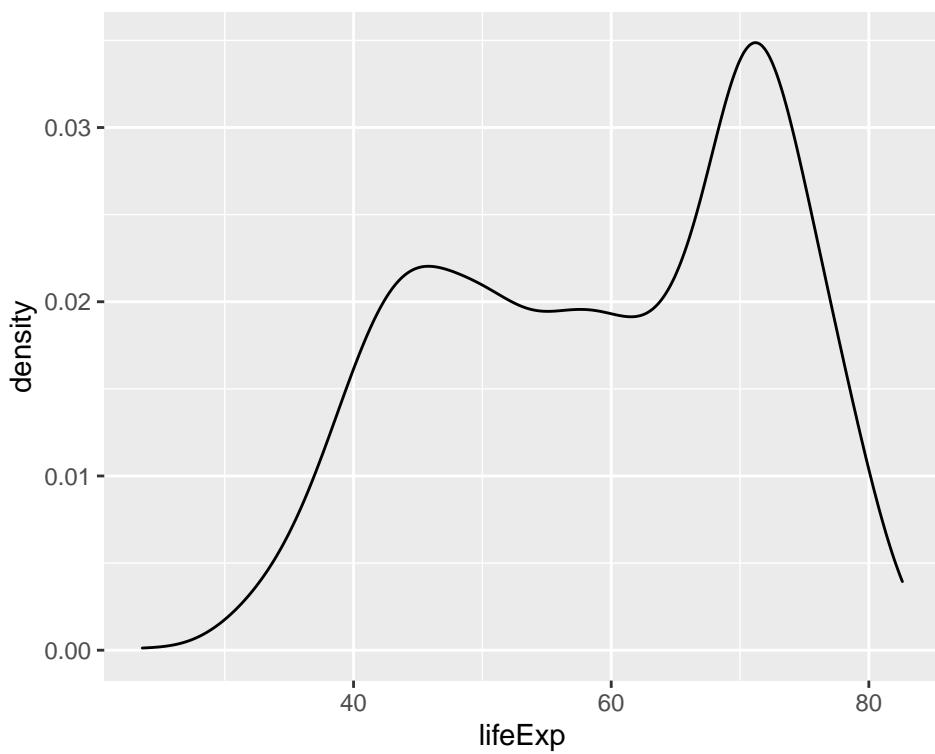
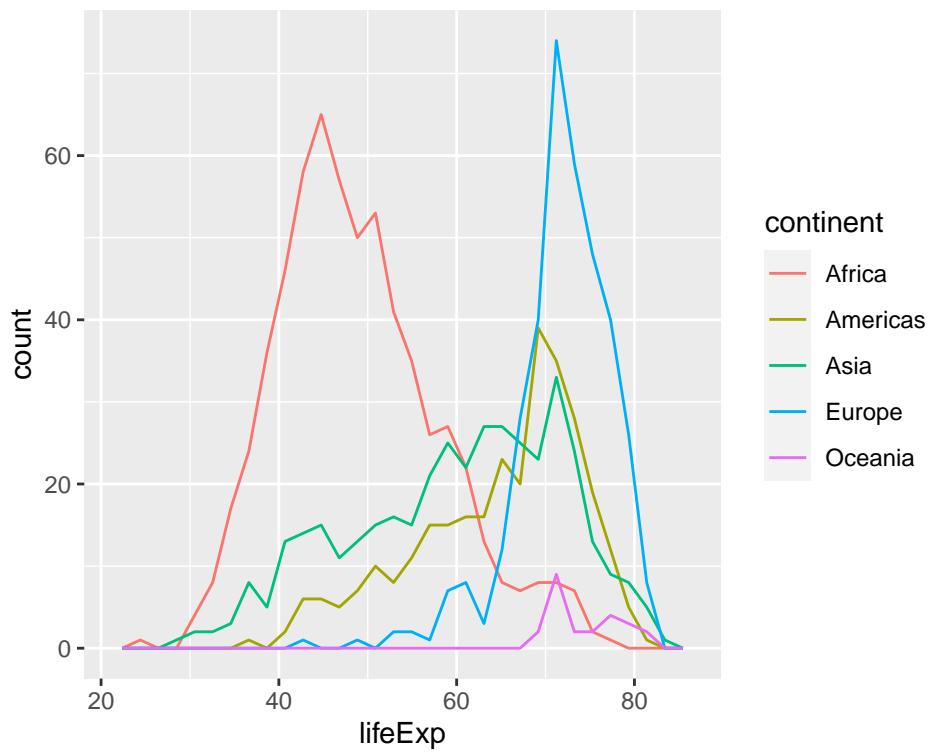
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
geom_freqpoly() is better in this case
```

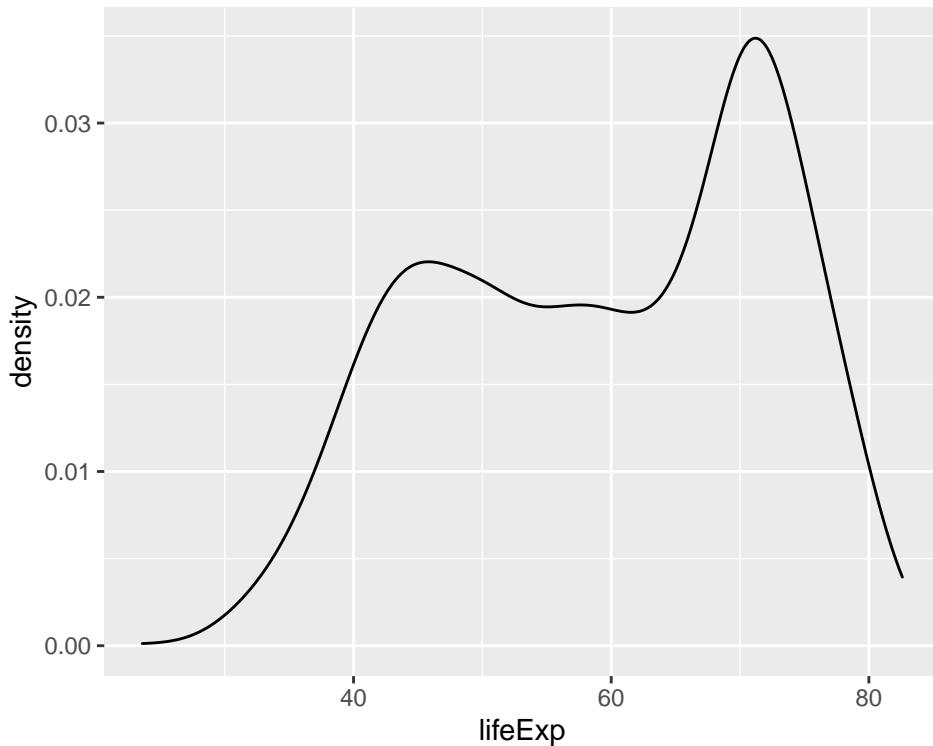
```
ggplot(gapminder, aes(x = lifeExp, color = continent)) +  
  geom_freqpoly()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

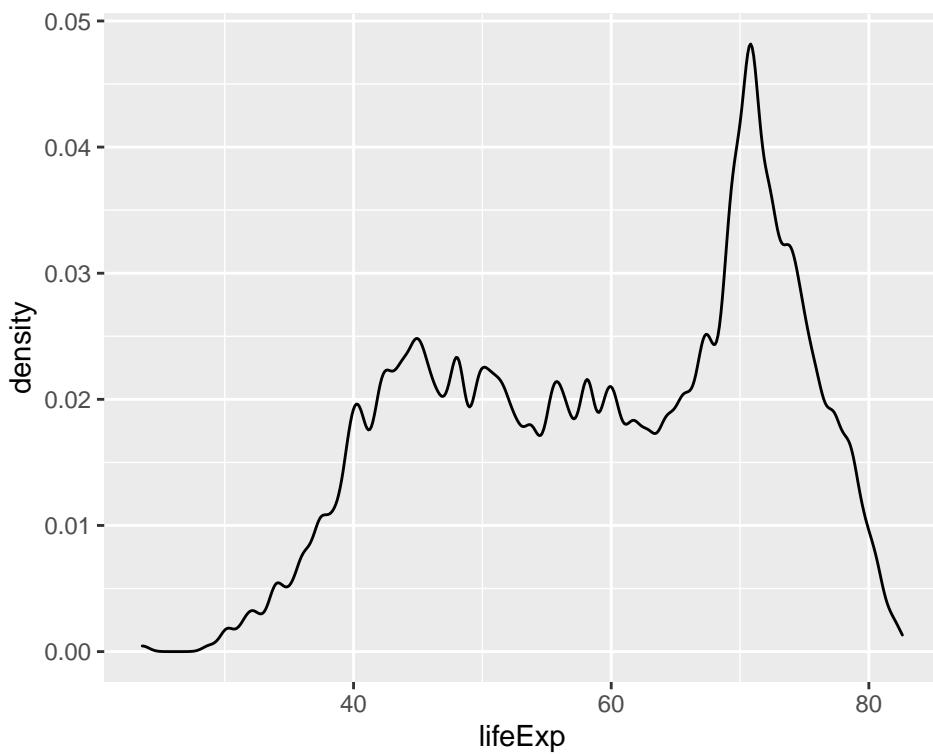


you should look at different levels of smoothing

```
ggplot(gapminder, aes(x = lifeExp)) + geom_density(adjust = 1)
```

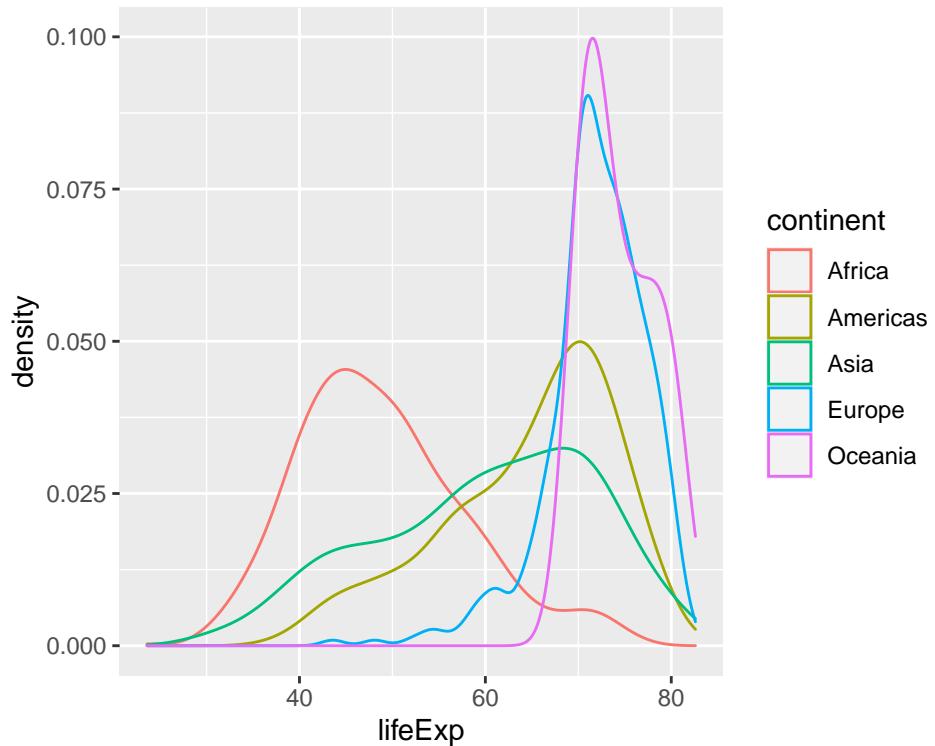


```
ggplot(gapminder, aes(x = lifeExp)) + geom_density(adjust = 0.2)
```



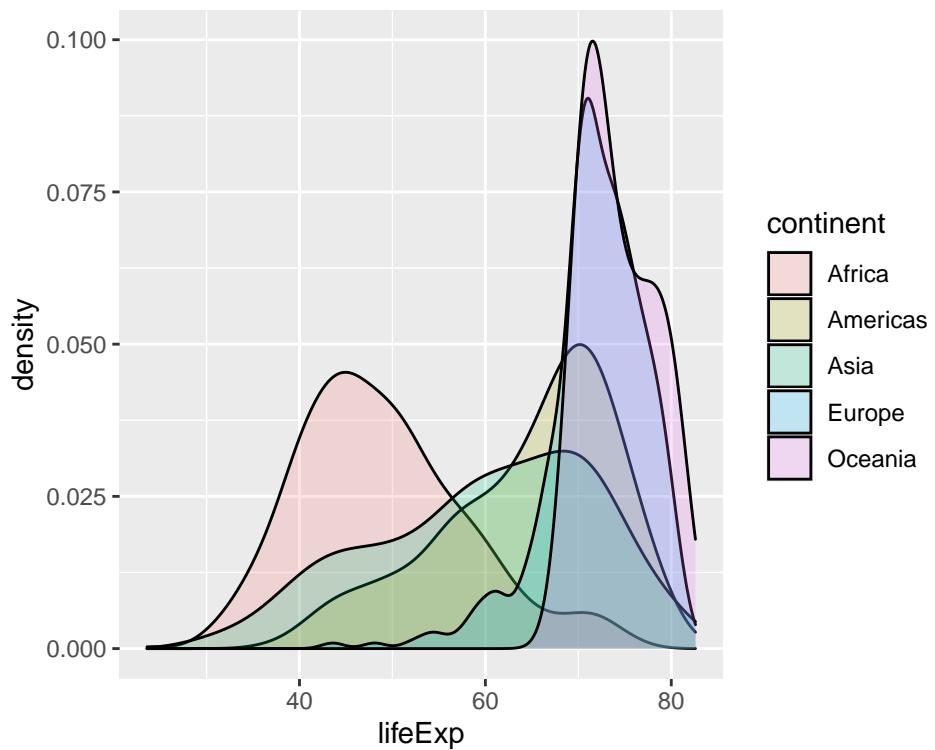
densityplots work better in terms of one continent not obscuring another

```
ggplot(gapminder, aes(x = lifeExp, color = continent)) + geom_density()
```



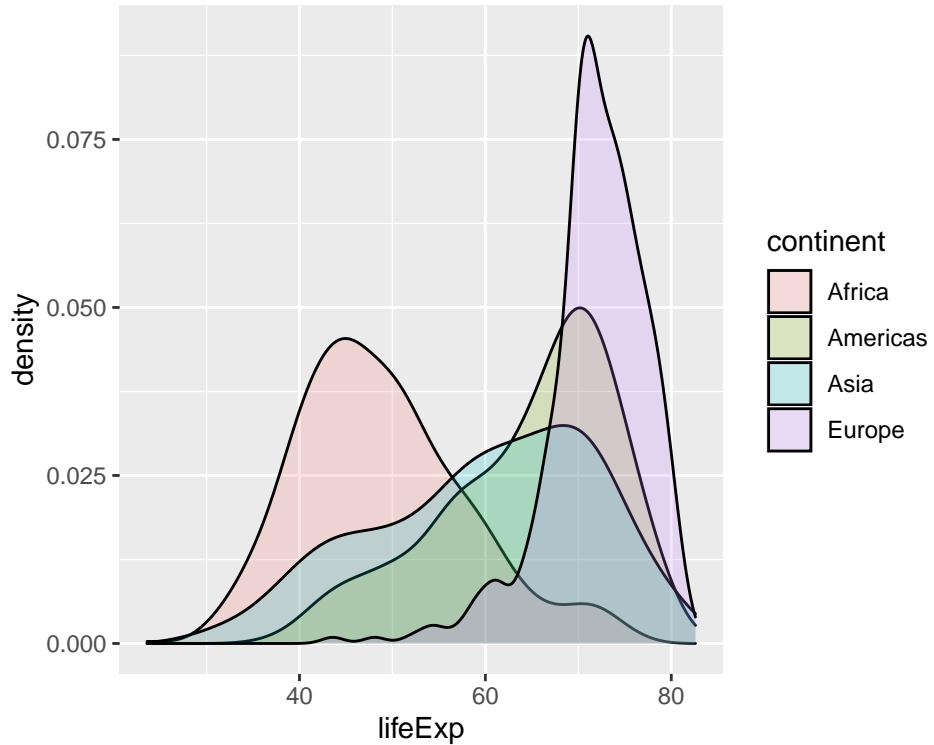
alpha transparency works here too

```
ggplot(gapminder, aes(x = lifeExp, fill = continent)) +  
  geom_density(alpha = 0.2)
```



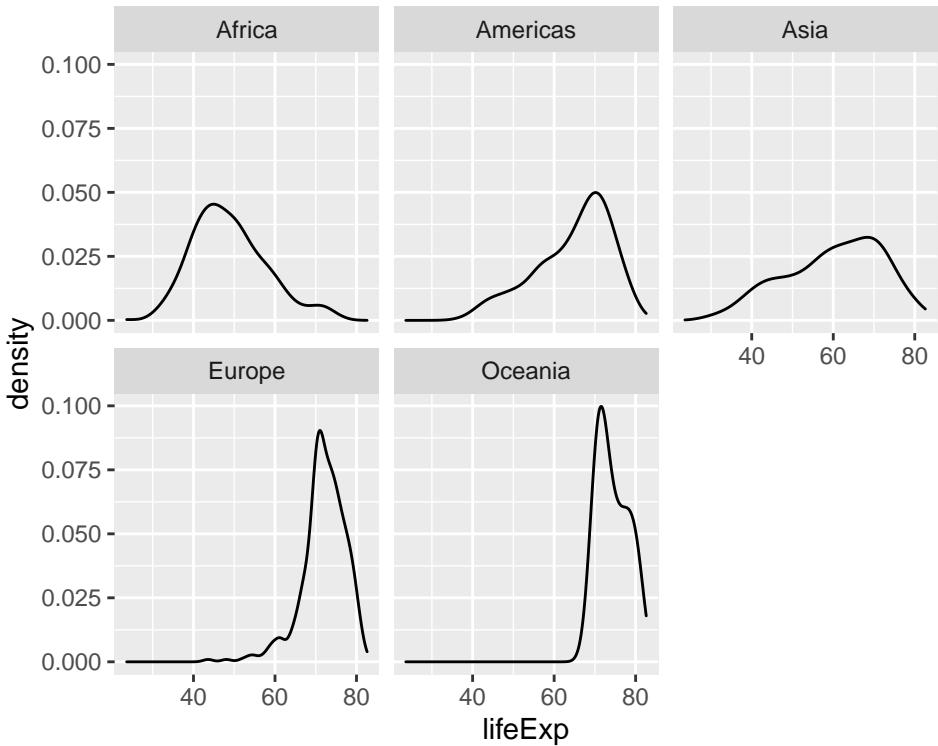
with only two countries, maybe we should ignore Oceania?

```
ggplot(subset(gapminder, continent != "Oceania"),  
       aes(x = lifeExp, fill = continent)) + geom_density(alpha = 0.2)
```



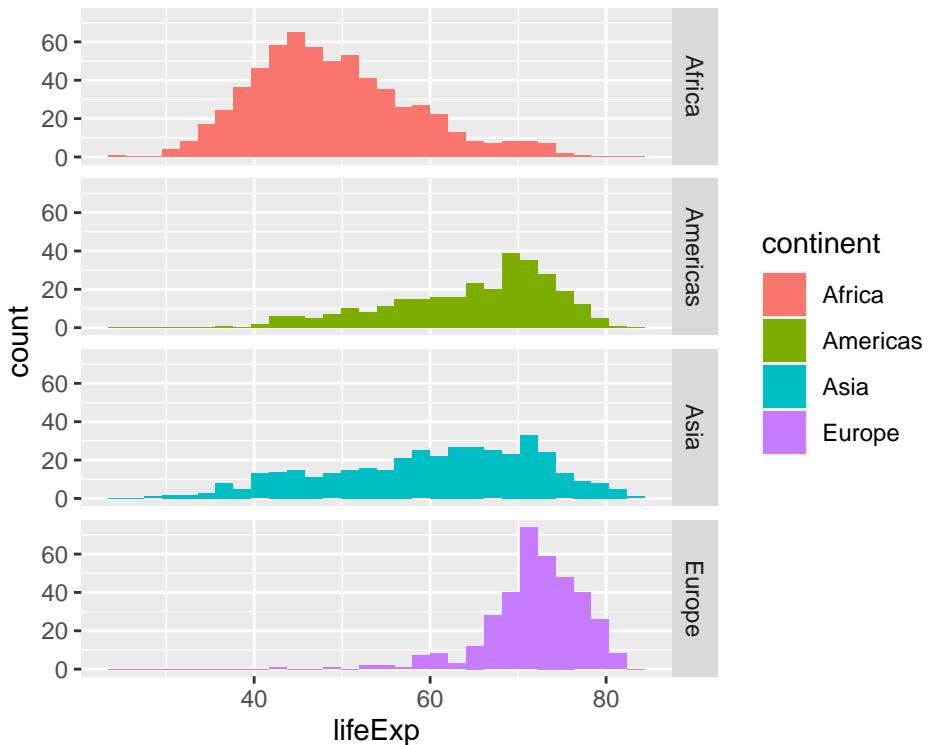
facets work here too

```
ggplot(gapminder, aes(x = lifeExp)) + geom_density() + facet_wrap(~ continent)
```



```
ggplot(subset(gapminder, continent != "Oceania"),
       aes(x = lifeExp, fill = continent)) + geom_histogram() +
       facet_grid(continent ~ .)
```

## `stat\_bin()` using `bins = 30` . Pick better value with `binwidth` .



boxplot for one quantitative variable against a discrete variable

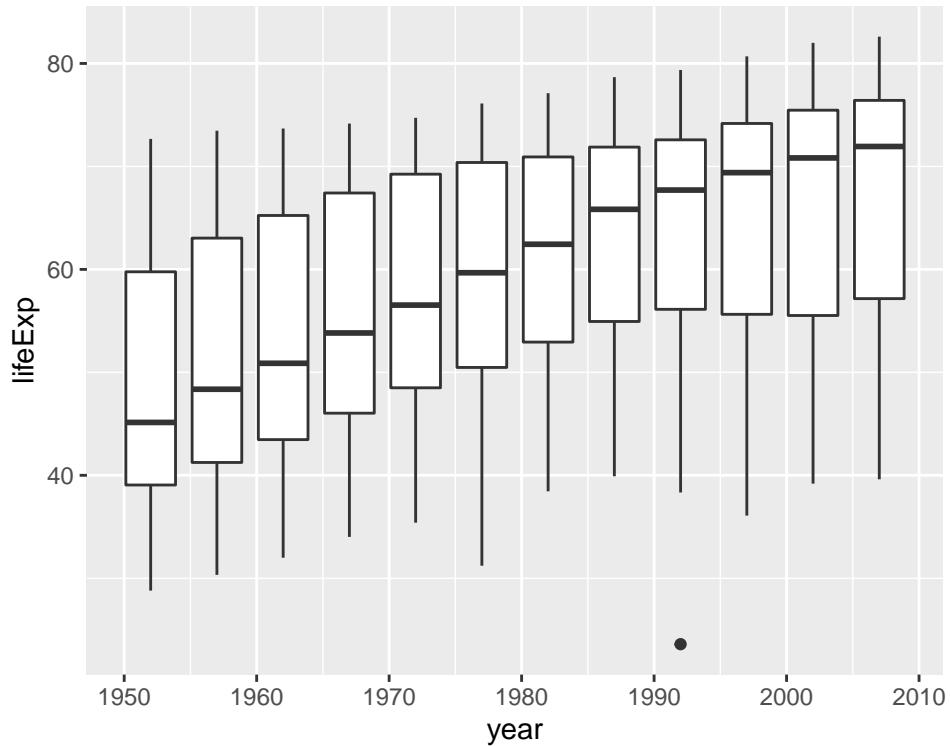
first attempt does not work since year is not formally a factor

```
ggplot(gapminder, aes(x = year, y = lifeExp)) + geom_boxplot()  
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



by explicitly specifying year as the grouping variable, we get what we want

```
ggplot(gapminder, aes(x = year, y = lifeExp)) + geom_boxplot(aes(group = year))
```

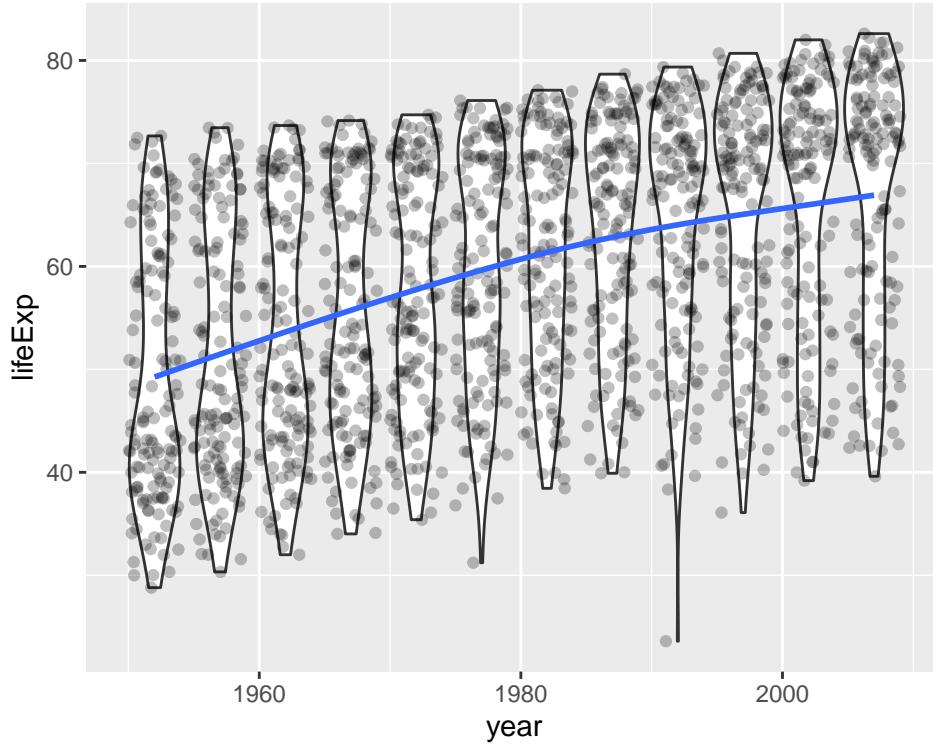


```
# Violin Plots, a new visualization
# http://ggplot2.tidyverse.org/reference/geom_violin.html
```

try geom\_violin() instead and just generally goofing off now

```
ggplot(gapminder, aes(x = year, y = lifeExp)) +
  geom_violin(aes(group = year)) +
  geom_jitter(alpha = 1 / 4) +
  geom_smooth(se = FALSE)

## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



```
#####
# Drawing Bars
#####

#' ---
#' author: "Jenny Bryan"
#' output: github_document
#' ---

#+ setup, include = FALSE

library(knitr)

opts_chunk$set(fig.path = 'figs/bryan/uni-factor-', error = TRUE)
```

Note: this is rendered by applying `knitr::spin()` to an R script. So the narrative is very minimal. load the data and `ggplot2` (part of the tidyverse)

```
library(tidyverse)

library(gapminder)

gapminder

## # A tibble: 1,704 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.
## 7 Afghanistan Asia     1982    39.9 12881816    978.
## 8 Afghanistan Asia     1987    40.8 13867957    852.
## 9 Afghanistan Asia     1992    41.7 16317921    649.
## 10 Afghanistan Asia    1997    41.8 22227415    635.
## # ... with 1,694 more rows

```

bar charts

consider: no. of observations for each continent

```
table(gapminder$continent)
```

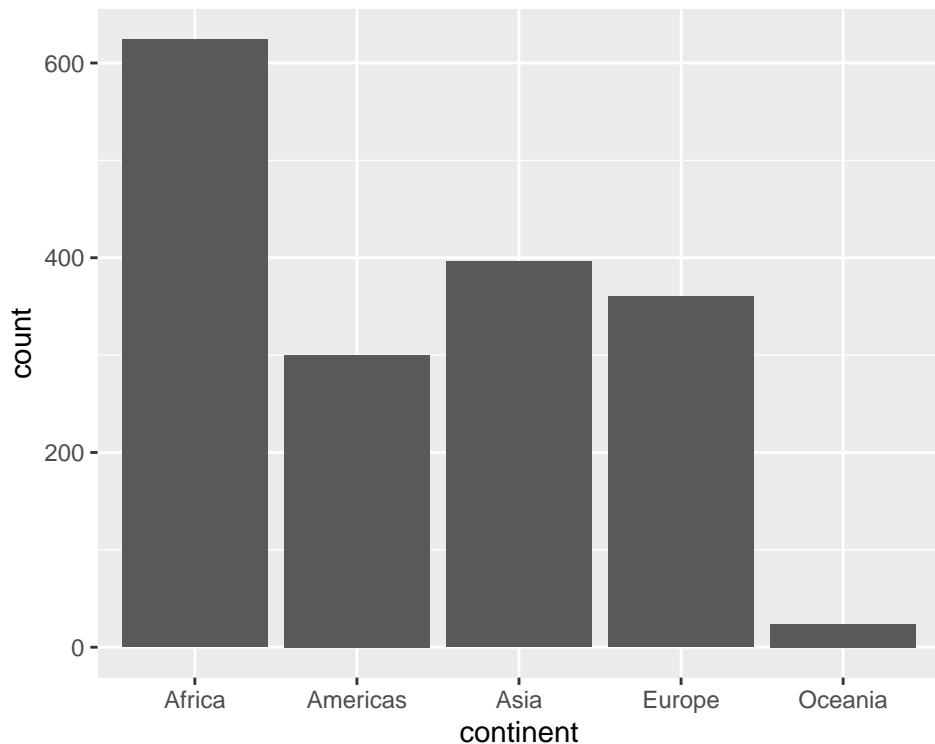
```

##
##   Africa Americas      Asia   Europe Oceania
##       624      300      396      360       24

```

this works because default stat for geom\_bar() is "bin"

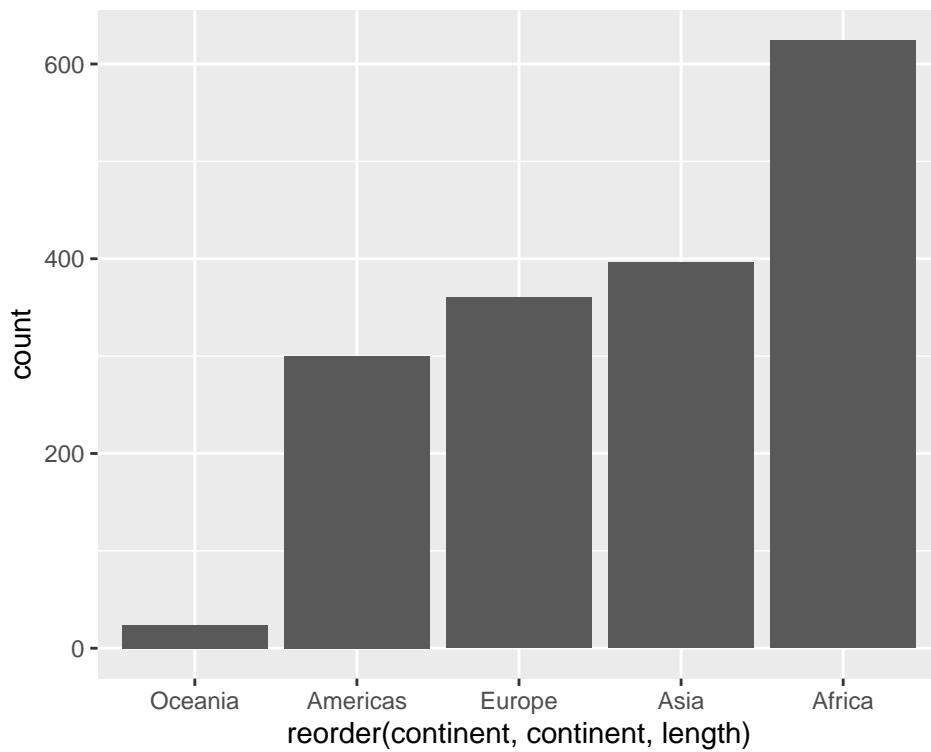
```
ggplot(gapminder, aes(x = continent)) + geom_bar()
```



let's reorder the continents based on frequency

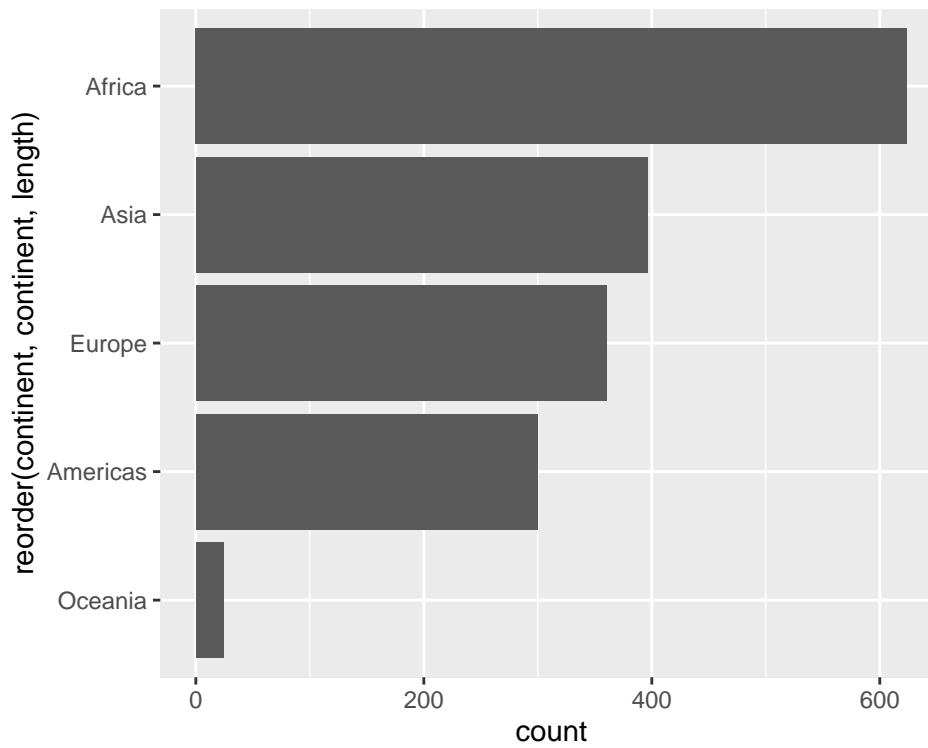
```
p <- ggplot(gapminder, aes(x = reorder(continent, continent, length)))
```

```
p + geom_bar()
```



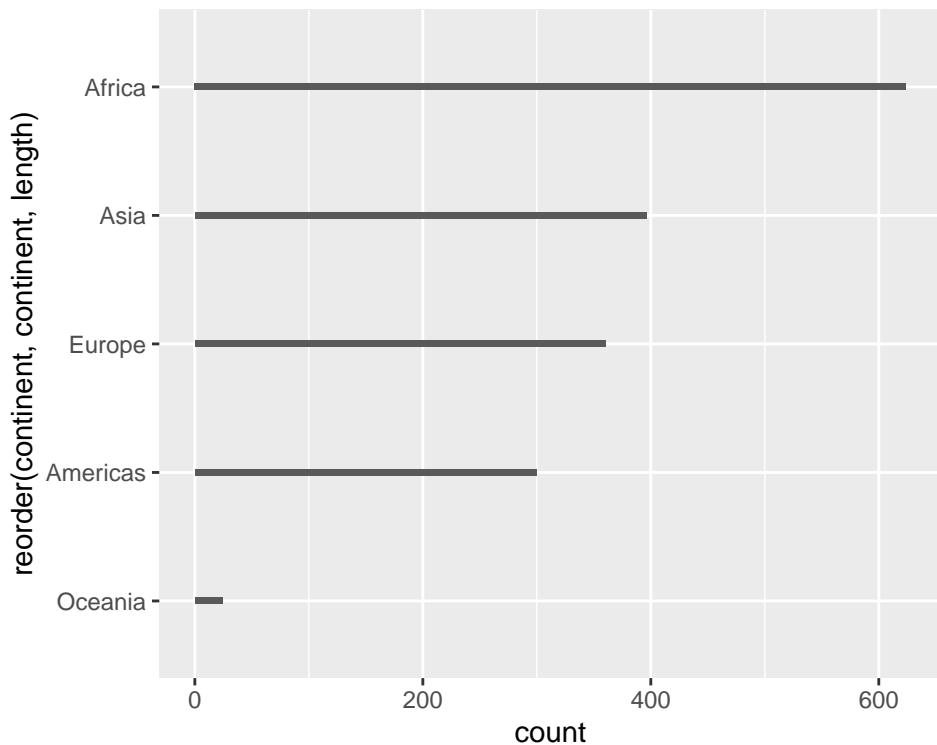
would you rather the bars run horizontally?

```
p + geom_bar() + coord_flip()
```



how about a better data:ink ratio?

```
p + geom_bar(width = 0.05) + coord_flip()
```



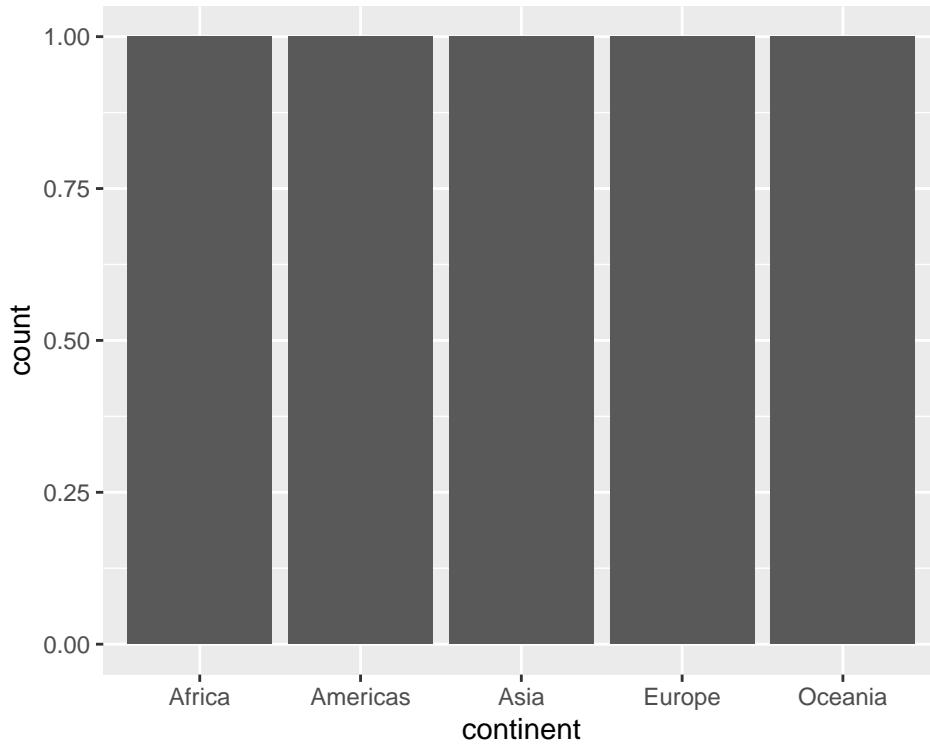
consider a scenario where you DON'T want the default "bin" stat, i.e. the bar length or height already exists as a variable

```
(continent_freq <- gapminder %>% count(continent))
```

```
## # A tibble: 5 x 2
##   continent     n
##   <fct>     <int>
## 1 Africa      624
## 2 Americas    300
## 3 Asia        396
## 4 Europe      360
## 5 Oceania     24
```

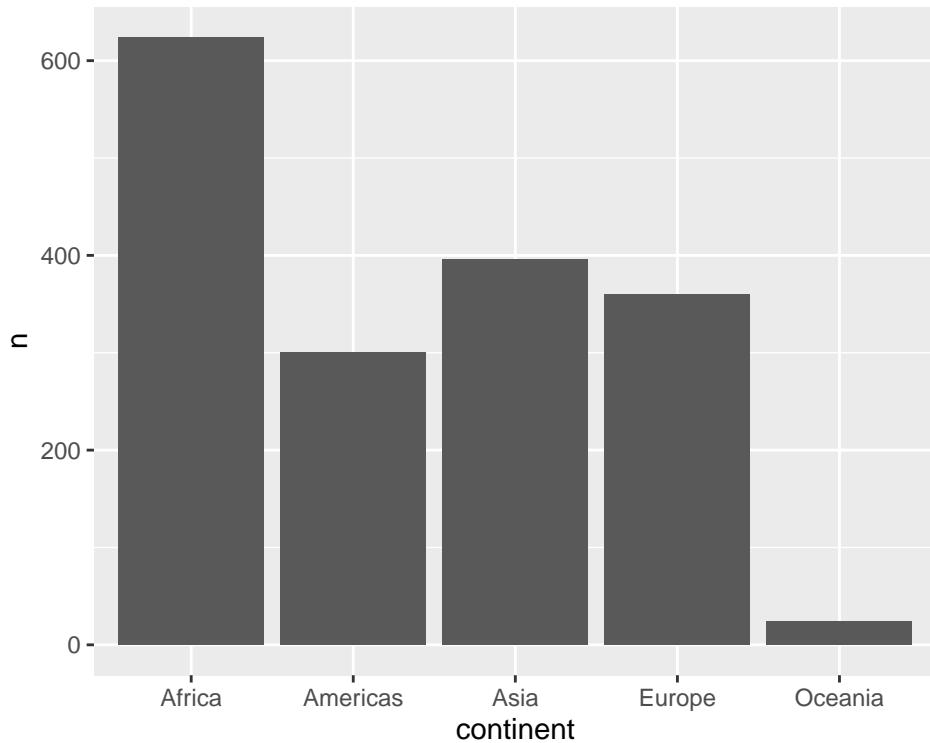
this simple call no longer works, because we have pre-tabulated

```
ggplot(continent_freq, aes(x = continent)) + geom_bar()
```



THIS works when bar length or height already exists

```
ggplot(continent_freq, aes(x = continent, y = n)) + geom_bar(stat = "identity")
```



```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23)
## Platform: x86_64-pc-linux-gnu (64-bit)
```

```

## Running under: Ubuntu 20.04.5 LTS
##
## Matrix products: default
## BLAS:    /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK:  /usr/lib/x86_64-linux-gnu/openblas-pthread/liblapack.so.3
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
## [3] LC_TIME=en_US.UTF-8       LC_COLLATE=en_US.UTF-8
## [5] LC_MONETARY=en_US.UTF-8   LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=en_US.UTF-8     LC_NAME=C
## [9] LC_ADDRESS=C              LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats      graphics   grDevices utils      datasets   methods    base
##
## other attached packages:
## [1] forcats_0.5.2   stringr_1.4.1   purrr_0.3.4    readr_2.1.2
## [5] tidyverse_1.3.1  dplyr_1.0.10   gapminder_0.3.0
## [9] ggplot2_3.3.6   tibble_3.1.7   knitr_1.40
##
## loaded via a namespace (and not attached):
## [1] tidyselect_1.1.2 xfun_0.32      splines_4.2.1   haven_2.5.1
## [5] lattice_0.20-45 colorspace_2.0-3 vctrs_0.4.1    generics_0.1.3
## [9] htmltools_0.5.3  yaml_2.3.5    mgcv_1.8-40    utf8_1.2.2
## [13] rlang_1.0.5     pillar_1.8.1   glue_1.6.2     withr_2.5.0
## [17] DBI_1.1.3      dbplyr_2.2.1  readxl_1.4.1   modelr_0.1.9
## [21] lifecycle_1.0.2  cellranger_1.1.0 munsell_0.5.0  gtable_0.3.1
## [25] rvest_1.0.3     evaluate_0.16 labeling_0.4.2  tzdb_0.3.0
## [29] fastmap_1.1.0   fansi_1.0.3   highr_0.9     broom_1.0.1
## [33] scales_1.2.1    backports_1.4.1 jsonlite_1.8.0 fs_1.5.2
## [37] farver_2.1.1    hms_1.1.2    digest_0.6.29  stringi_1.7.8
## [41] grid_4.2.1      cli_3.4.0    tools_4.2.1   magrittr_2.0.3
## [45] crayon_1.5.1    pkgconfig_2.0.3 ellipsis_0.3.2 Matrix_1.5-0
## [49] xml2_1.3.3     reprex_2.0.2  lubridate_1.8.0 assertthat_0.2.1
## [53] rmarkdown_2.16   httr_1.4.4   rstudioapi_0.14 R6_2.5.1
## [57] nlme_3.1-159    compiler_4.2.1

#=====
# Change overall look and feel via themes
#=====

#' ---
#' author: "Jenny Bryan"
#' output:
#'   html_document:
#'     keep_md: TRUE
#' ---

#+ setup, include = FALSE

library(knitr)

```

```
opts_chunk$set(fig.path = 'figs/bryan/themes-', error = TRUE)
```

Note: this HTML is made by applying `knitr::spin()` to an R script. So the narrative is very minimal.

```
library(ggplot2)

# install.packages("ggthemes")

library(ggthemes)

?ggthemes
```

pick a way to load the data

```
#gdURL <- "http://tiny.cc/gapminder"
#gapminder <- read.delim(file = gdURL)
#gapminder <- read.delim("gapminderDataFiveYear.tsv")

library(gapminder)

str(gapminder)
```

```
## tibble [1,704 x 6] (S3:tbl_df/tbl/data.frame)
## $ country : Factor w/ 142 levels "Afghanistan",...: 1 1 1 1 1 1 1 1 1 ...
## $ continent: Factor w/ 5 levels "Africa","Americas",...: 3 3 3 3 3 3 3 3 3 ...
## $ year     : int [1:1704] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
## $ lifeExp   : num [1:1704] 28.8 30.3 32 34 36.1 ...
## $ pop       : int [1:1704] 8425333 9240934 10267083 11537966 13079460 14880372 12881816 13867957 163
## $ gdpPercap: num [1:1704] 779 821 853 836 740 ...
```

revisit a plot from earlier

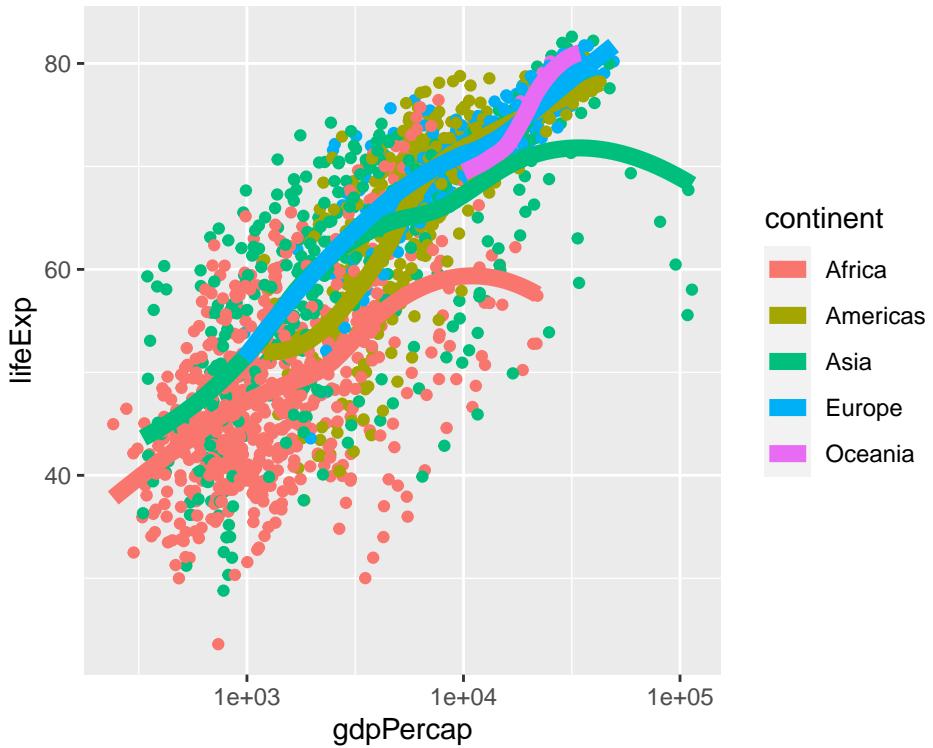
```
p <- ggplot(gapminder, aes(x = gdpPercap, y = lifeExp))

p <- p + scale_x_log10()

p <- p + aes(color = continent) + geom_point() + geom_smooth(lwd = 3, se = FALSE)

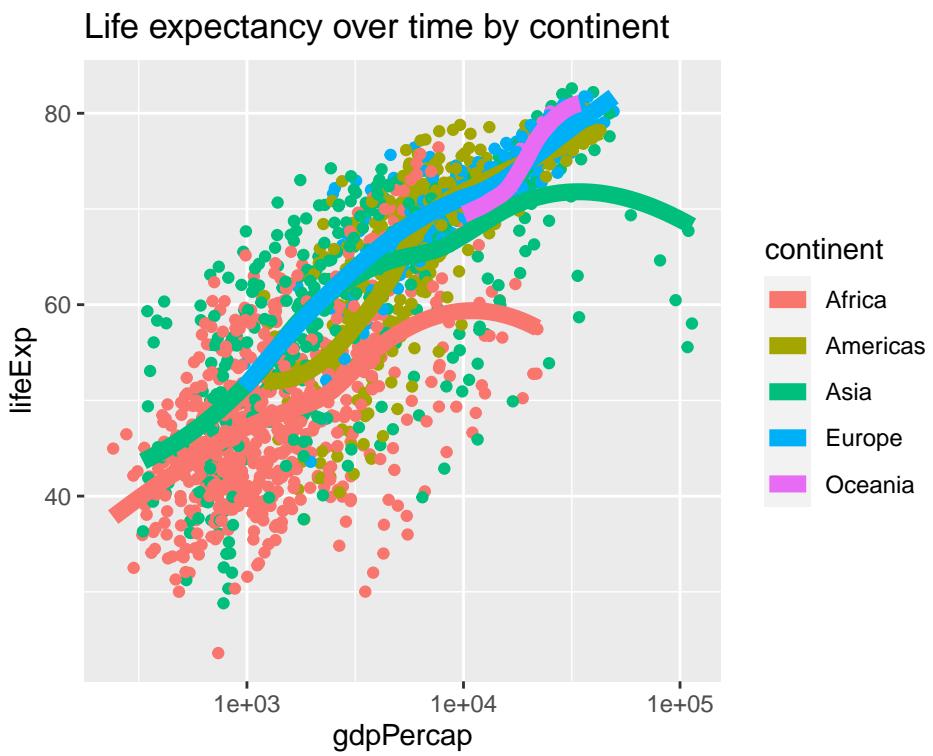
p

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



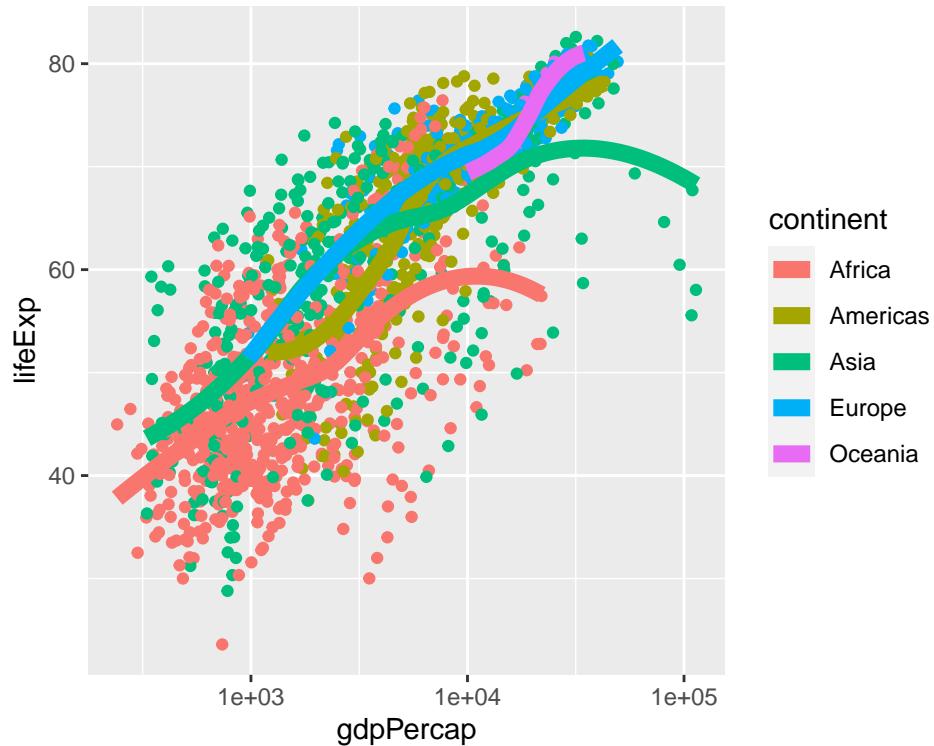
give it a title

```
p + ggtitle("Life expectancy over time by continent")
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



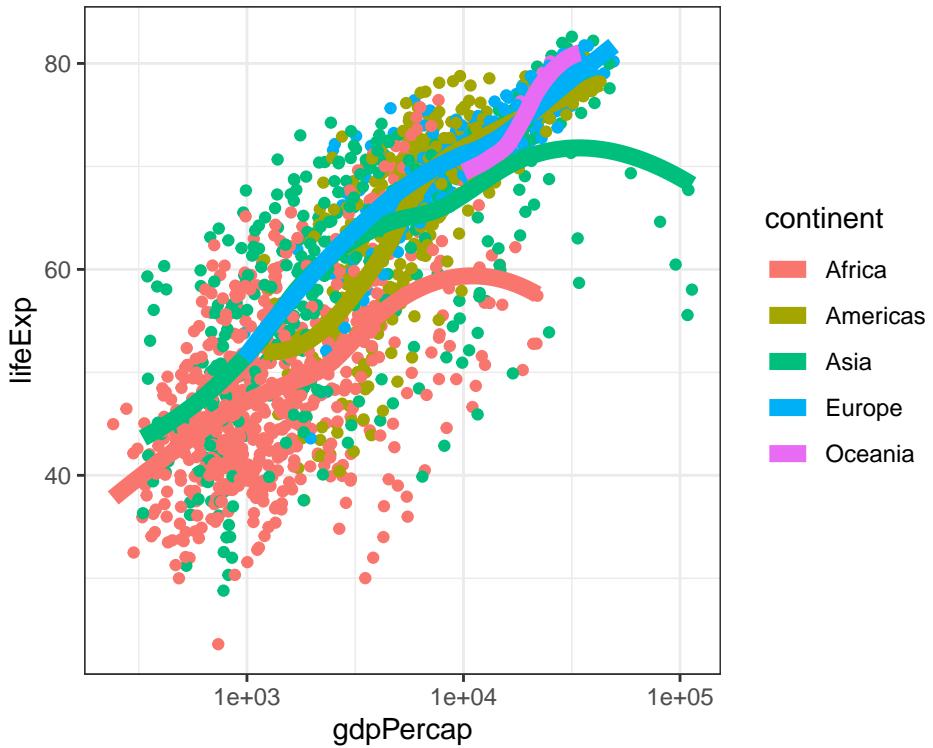
change overall look and feel with a premade theme

```
p + theme_grey() # the default  
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



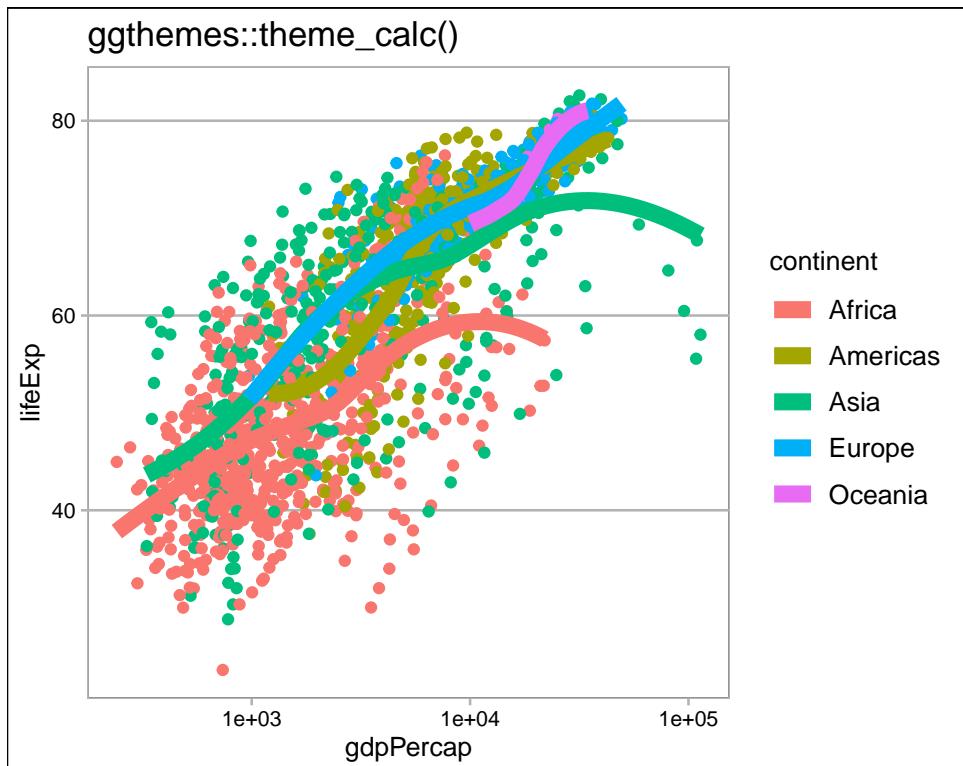
suppress the usual grey background

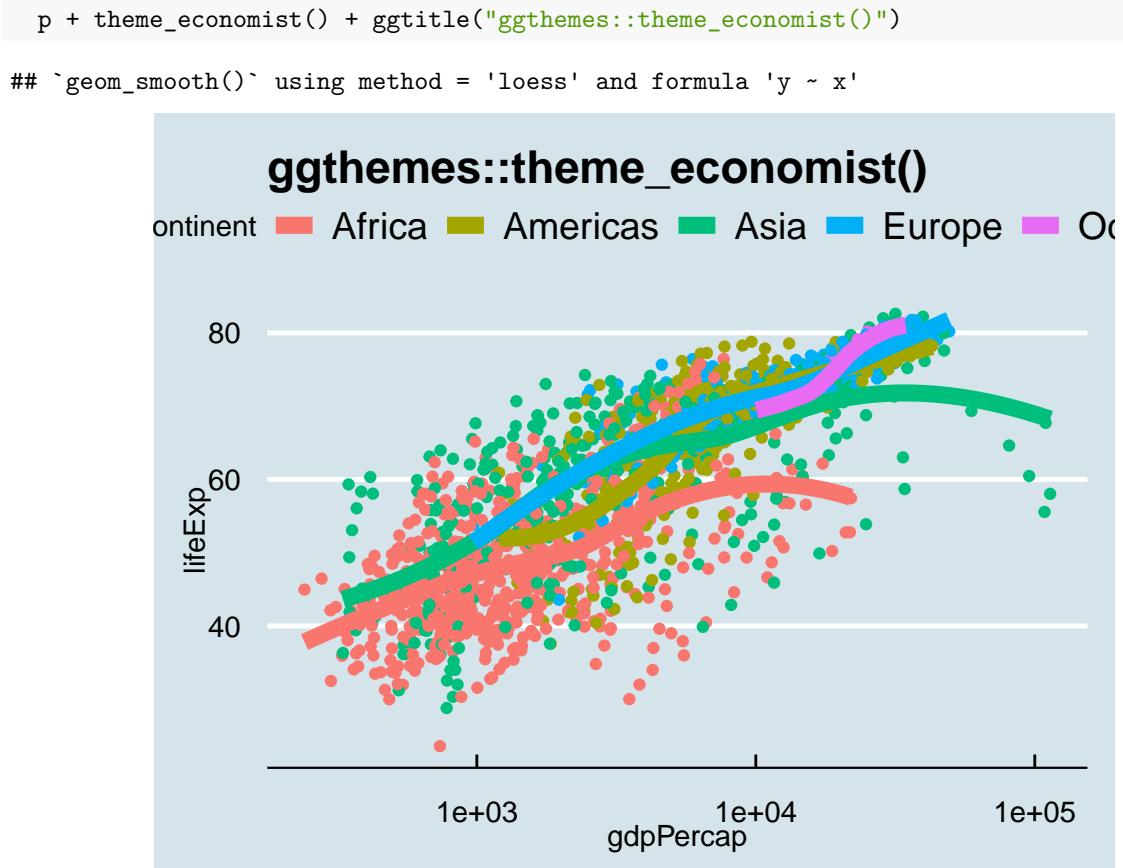
```
p + theme_bw()  
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



exploring some themes from the ggthemes package  
<https://github.com/jrnold/ggthemes>

```
p + theme_calc() + ggtitle("ggthemes::theme_calc()")  
  
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```





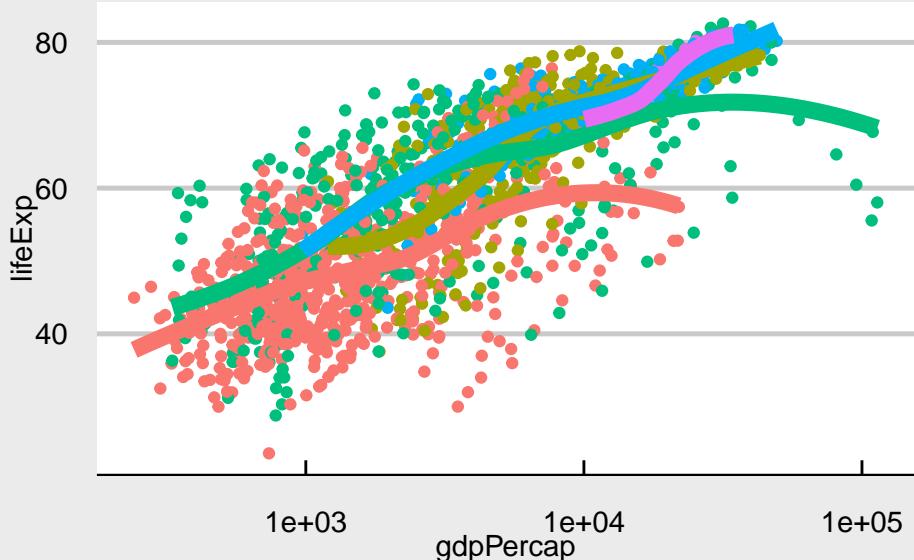
```

p + theme_economist_white() + ggtitle("ggthemes::theme_economist_white()")
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

```

**ggthemes::theme\_economist\_white()**

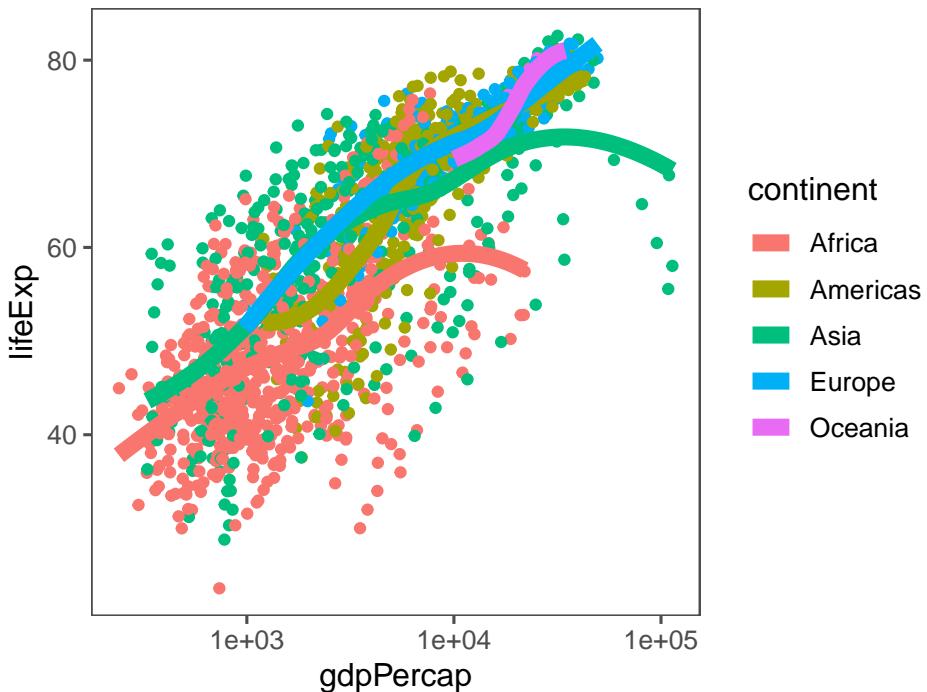
continent Africa Americas Asia Europe Oceania



```
p + theme_few() + ggtitle("ggthemes::theme_few()")
```

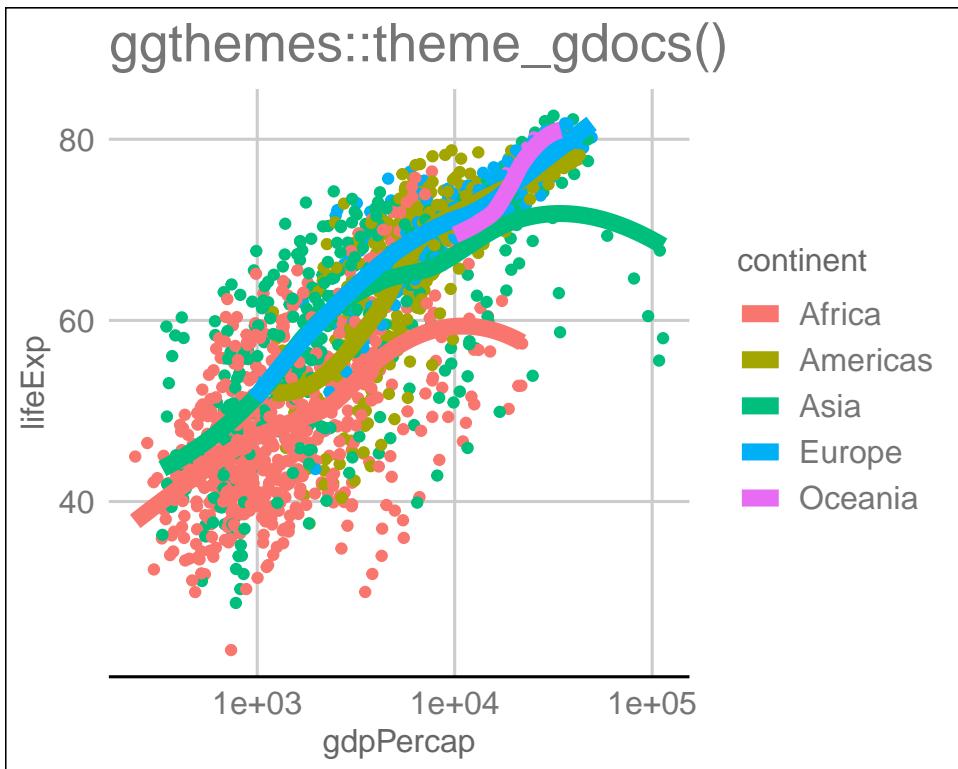
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```

**ggthemes::theme\_few()**



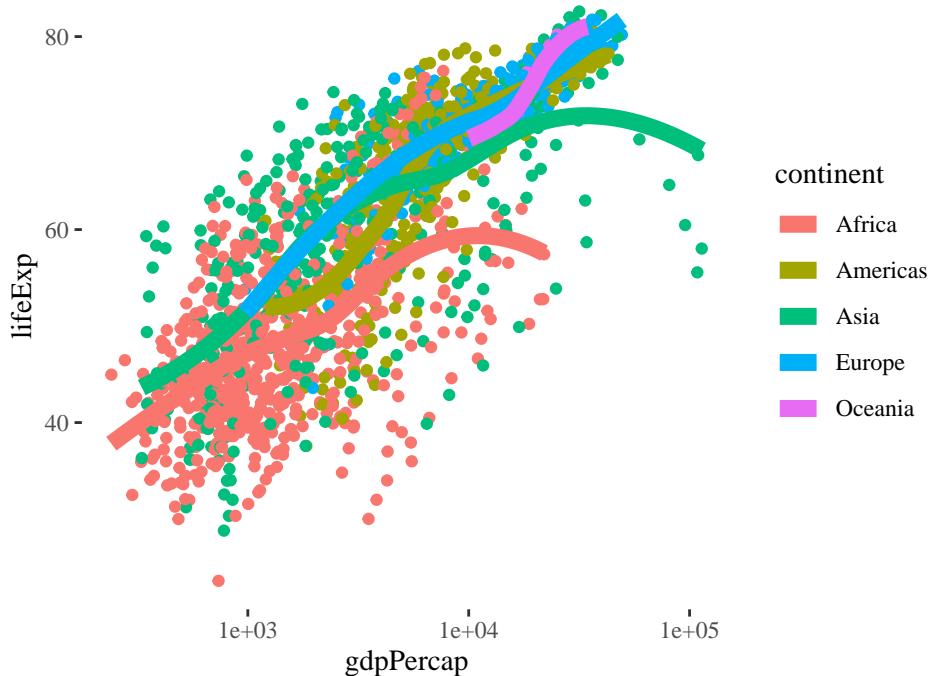
```
p + theme_gdocs() + ggtitle("ggthemes::theme_gdocs()")
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



```
p + theme_tufte() + ggtitle("ggthemes::theme_tufte()")
```

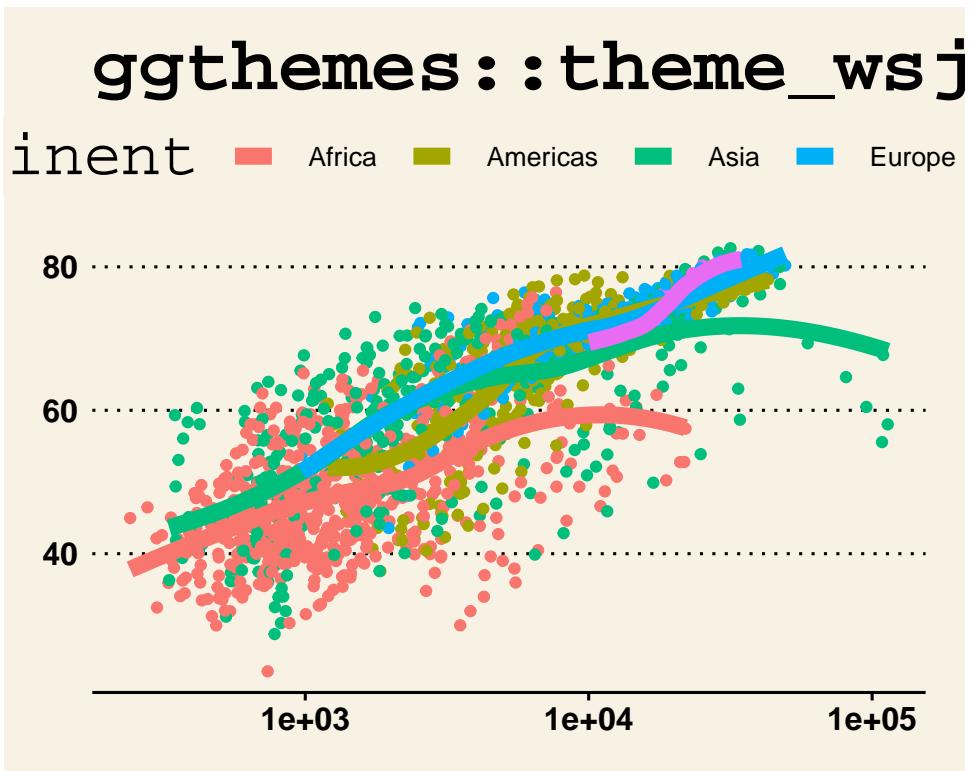
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'  
ggthemes::theme_tufte()
```



```

p + theme_wsj() + ggtitle("ggthemes::theme_wsj()")
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'

```



```

sessionInfo()

## R version 4.2.1 (2022-06-23)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.5 LTS
##
## Matrix products: default
## BLAS:    /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK:  /usr/lib/x86_64-linux-gnu/openblas-pthread/liblapack.so.3
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8        LC_NUMERIC=C
## [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
## [5] LC_MONETARY=en_US.UTF-8     LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
## [9] LC_ADDRESS=C                LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats      graphics   grDevices  utils      datasets   methods   base
##
## other attached packages:
## [1] ggthemes_4.2.4  forcats_0.5.2   stringr_1.4.1   purrr_0.3.4
## [5] readr_2.1.2     tidyr_1.2.1     tidyverse_1.3.1  dplyr_1.0.10
## [9] gapminder_0.3.0 ggplot2_3.3.6   tibble_3.1.7    knitr_1.40
## [11]
## [13]
## [15]
## [17]
## [19]
## [21]
## [23]
## [25]
## [27]
## [29]
## [31]
## [33]
## [35]
## [37]
## [39]
## [41]
## [43]
## [45]
## [47]
## [49]
## [51]
## [53]
## [55]
## [57]
## [59]
## [61]
## [63]
## [65]
## [67]
## [69]
## [71]
## [73]
## [75]
## [77]
## [79]
## [81]
## [83]
## [85]
## [87]
## [89]
## [91]
## [93]
## [95]
## [97]
## [99]
## [101]
## [103]
## [105]
## [107]
## [109]
## [111]
## [113]
## [115]
## [117]
## [119]
## [121]
## [123]
## [125]
## [127]
## [129]
## [131]
## [133]
## [135]
## [137]
## [139]
## [141]
## [143]
## [145]
## [147]
## [149]
## [151]
## [153]
## [155]
## [157]
## [159]
## [161]
## [163]
## [165]
## [167]
## [169]
## [171]
## [173]
## [175]
## [177]
## [179]
## [181]
## [183]
## [185]
## [187]
## [189]
## [191]
## [193]
## [195]
## [197]
## [199]
## [201]
## [203]
## [205]
## [207]
## [209]
## [211]
## [213]
## [215]
## [217]
## [219]
## [221]
## [223]
## [225]
## [227]
## [229]
## [231]
## [233]
## [235]
## [237]
## [239]
## [241]
## [243]
## [245]
## [247]
## [249]
## [251]
## [253]
## [255]
## [257]
## [259]
## [261]
## [263]
## [265]
## [267]
## [269]
## [271]
## [273]
## [275]
## [277]
## [279]
## [281]
## [283]
## [285]
## [287]
## [289]
## [291]
## [293]
## [295]
## [297]
## [299]
## [301]
## [303]
## [305]
## [307]
## [309]
## [311]
## [313]
## [315]
## [317]
## [319]
## [321]
## [323]
## [325]
## [327]
## [329]
## [331]
## [333]
## [335]
## [337]
## [339]
## [341]
## [343]
## [345]
## [347]
## [349]
## [351]
## [353]
## [355]
## [357]
## [359]
## [361]
## [363]
## [365]
## [367]
## [369]
## [371]
## [373]
## [375]
## [377]
## [379]
## [381]
## [383]
## [385]
## [387]
## [389]
## [391]
## [393]
## [395]
## [397]
## [399]
## [401]
## [403]
## [405]
## [407]
## [409]
## [411]
## [413]
## [415]
## [417]
## [419]
## [421]
## [423]
## [425]
## [427]
## [429]
## [431]
## [433]
## [435]
## [437]
## [439]
## [441]
## [443]
## [445]
## [447]
## [449]
## [451]
## [453]
## [455]
## [457]
## [459]
## [461]
## [463]
## [465]
## [467]
## [469]
## [471]
## [473]
## [475]
## [477]
## [479]
## [481]
## [483]
## [485]
## [487]
## [489]
## [491]
## [493]
## [495]
## [497]
## [499]
## [501]
## [503]
## [505]
## [507]
## [509]
## [511]
## [513]
## [515]
## [517]
## [519]
## [521]
## [523]
## [525]
## [527]
## [529]
## [531]
## [533]
## [535]
## [537]
## [539]
## [541]
## [543]
## [545]
## [547]
## [549]
## [551]
## [553]
## [555]
## [557]
## [559]
## [561]
## [563]
## [565]
## [567]
## [569]
## [571]
## [573]
## [575]
## [577]
## [579]
## [581]
## [583]
## [585]
## [587]
## [589]
## [591]
## [593]
## [595]
## [597]
## [599]
## [601]
## [603]
## [605]
## [607]
## [609]
## [611]
## [613]
## [615]
## [617]
## [619]
## [621]
## [623]
## [625]
## [627]
## [629]
## [631]
## [633]
## [635]
## [637]
## [639]
## [641]
## [643]
## [645]
## [647]
## [649]
## [651]
## [653]
## [655]
## [657]
## [659]
## [661]
## [663]
## [665]
## [667]
## [669]
## [671]
## [673]
## [675]
## [677]
## [679]
## [681]
## [683]
## [685]
## [687]
## [689]
## [691]
## [693]
## [695]
## [697]
## [699]
## [701]
## [703]
## [705]
## [707]
## [709]
## [711]
## [713]
## [715]
## [717]
## [719]
## [721]
## [723]
## [725]
## [727]
## [729]
## [731]
## [733]
## [735]
## [737]
## [739]
## [741]
## [743]
## [745]
## [747]
## [749]
## [751]
## [753]
## [755]
## [757]
## [759]
## [761]
## [763]
## [765]
## [767]
## [769]
## [771]
## [773]
## [775]
## [777]
## [779]
## [781]
## [783]
## [785]
## [787]
## [789]
## [791]
## [793]
## [795]
## [797]
## [799]
## [801]
## [803]
## [805]
## [807]
## [809]
## [811]
## [813]
## [815]
## [817]
## [819]
## [821]
## [823]
## [825]
## [827]
## [829]
## [831]
## [833]
## [835]
## [837]
## [839]
## [841]
## [843]
## [845]
## [847]
## [849]
## [851]
## [853]
## [855]
## [857]
## [859]
## [861]
## [863]
## [865]
## [867]
## [869]
## [871]
## [873]
## [875]
## [877]
## [879]
## [881]
## [883]
## [885]
## [887]
## [889]
## [891]
## [893]
## [895]
## [897]
## [899]
## [901]
## [903]
## [905]
## [907]
## [909]
## [911]
## [913]
## [915]
## [917]
## [919]
## [921]
## [923]
## [925]
## [927]
## [929]
## [931]
## [933]
## [935]
## [937]
## [939]
## [941]
## [943]
## [945]
## [947]
## [949]
## [951]
## [953]
## [955]
## [957]
## [959]
## [961]
## [963]
## [965]
## [967]
## [969]
## [971]
## [973]
## [975]
## [977]
## [979]
## [981]
## [983]
## [985]
## [987]
## [989]
## [991]
## [993]
## [995]
## [997]
## [999]
## [1001]
## [1003]
## [1005]
## [1007]
## [1009]
## [1011]
## [1013]
## [1015]
## [1017]
## [1019]
## [1021]
## [1023]
## [1025]
## [1027]
## [1029]
## [1031]
## [1033]
## [1035]
## [1037]
## [1039]
## [1041]
## [1043]
## [1045]
## [1047]
## [1049]
## [1051]
## [1053]
## [1055]
## [1057]
## [1059]
## [1061]
## [1063]
## [1065]
## [1067]
## [1069]
## [1071]
## [1073]
## [1075]
## [1077]
## [1079]
## [1081]
## [1083]
## [1085]
## [1087]
## [1089]
## [1091]
## [1093]
## [1095]
## [1097]
## [1099]
## [1101]
## [1103]
## [1105]
## [1107]
## [1109]
## [1111]
## [1113]
## [1115]
## [1117]
## [1119]
## [1121]
## [1123]
## [1125]
## [1127]
## [1129]
## [1131]
## [1133]
## [1135]
## [1137]
## [1139]
## [1141]
## [1143]
## [1145]
## [1147]
## [1149]
## [1151]
## [1153]
## [1155]
## [1157]
## [1159]
## [1161]
## [1163]
## [1165]
## [1167]
## [1169]
## [1171]
## [1173]
## [1175]
## [1177]
## [1179]
## [1181]
## [1183]
## [1185]
## [1187]
## [1189]
## [1191]
## [1193]
## [1195]
## [1197]
## [1199]
## [1201]
## [1203]
## [1205]
## [1207]
## [1209]
## [1211]
## [1213]
## [1215]
## [1217]
## [1219]
## [1221]
## [1223]
## [1225]
## [1227]
## [1229]
## [1231]
## [1233]
## [1235]
## [1237]
## [1239]
## [1241]
## [1243]
## [1245]
## [1247]
## [1249]
## [1251]
## [1253]
## [1255]
## [1257]
## [1259]
## [1261]
## [1263]
## [1265]
## [1267]
## [1269]
## [1271]
## [1273]
## [1275]
## [1277]
## [1279]
## [1281]
## [1283]
## [1285]
## [1287]
## [1289]
## [1291]
## [1293]
## [1295]
## [1297]
## [1299]
## [1301]
## [1303]
## [1305]
## [1307]
## [1309]
## [1311]
## [1313]
## [1315]
## [1317]
## [1319]
## [1321]
## [1323]
## [1325]
## [1327]
## [1329]
## [1331]
## [1333]
## [1335]
## [1337]
## [1339]
## [1341]
## [1343]
## [1345]
## [1347]
## [1349]
## [1351]
## [1353]
## [1355]
## [1357]
## [1359]
## [1361]
## [1363]
## [1365]
## [1367]
## [1369]
## [1371]
## [1373]
## [1375]
## [1377]
## [1379]
## [1381]
## [1383]
## [1385]
## [1387]
## [1389]
## [1391]
## [1393]
## [1395]
## [1397]
## [1399]
## [1401]
## [1403]
## [1405]
## [1407]
## [1409]
## [1411]
## [1413]
## [1415]
## [1417]
## [1419]
## [1421]
## [1423]
## [1425]
## [1427]
## [1429]
## [1431]
## [1433]
## [1435]
## [1437]
## [1439]
## [1441]
## [1443]
## [1445]
## [1447]
## [1449]
## [1451]
## [1453]
## [1455]
## [1457]
## [1459]
## [1461]
## [1463]
## [1465]
## [1467]
## [1469]
## [1471]
## [1473]
## [1475]
## [1477]
## [1479]
## [1481]
## [1483]
## [1485]
## [1487]
## [1489]
## [1491]
## [1493]
## [1495]
## [1497]
## [1499]
## [1501]
## [1503]
## [1505]
## [1507]
## [1509]
## [1511]
## [1513]
## [1515]
## [1517]
## [1519]
## [1521]
## [1523]
## [1525]
## [1527]
## [1529]
## [1531]
## [1533]
## [1535]
## [1537]
## [1539]
## [1541]
## [1543]
## [1545]
## [1547]
## [1549]
## [1551]
## [1553]
## [1555]
## [1557]
## [1559]
## [1561]
## [1563]
## [1565]
## [1567]
## [1569]
## [1571]
## [1573]
## [1575]
## [1577]
## [1579]
## [1581]
## [1583]
## [1585]
## [1587]
## [1589]
## [1591]
## [1593]
## [1595]
## [1597]
## [1599]
## [1601]
## [1603]
## [1605]
## [1607]
## [1609]
## [1611]
## [1613]
## [1615]
## [1617]
## [1619]
## [1621]
## [1623]
## [1625]
## [1627]
## [1629]
## [1631]
## [1633]
## [1635]
## [1637]
## [1639]
## [1641]
## [1643]
## [1645]
## [1647]
## [1649]
## [1651]
## [1653]
## [1655]
## [1657]
## [1659]
## [1661]
## [1663]
## [1665]
## [1667]
## [1669]
## [1671]
## [1673]
## [1675]
## [1677]
## [1679]
## [1681]
## [1683]
## [1685]
## [1687]
## [1689]
## [1691]
## [1693]
## [1695]
## [1697]
## [1699]
## [1701]
## [1703]
## [1705]
## [1707]
## [1709]
## [1711]
## [1713]
## [1715]
## [1717]
## [1719]
## [1721]
## [1723]
## [1725]
## [1727]
## [1729]
## [1731]
## [1733]
## [1735]
## [1737]
## [1739]
## [1741]
## [1743]
## [1745]
## [1747]
## [1749]
## [1751]
## [1753]
## [1755]
## [1757]
## [1759]
## [1761]
## [1763]
## [1765]
## [1767]
## [1769]
## [1771]
## [1773]
## [1775]
## [1777]
## [1779]
## [1781]
## [1783]
## [1785]
## [1787]
## [1789]
## [1791]
## [1793]
## [1795]
## [1797]
## [1799]
## [1801]
## [1803]
## [1805]
## [1807]
## [1809]
## [1811]
## [1813]
## [1815]
## [1817]
## [1819]
## [1821]
## [1823]
## [1825]
## [1827]
## [1829]
## [1831]
## [1833]
## [1835]
## [1837]
## [1839]
## [1841]
## [1843]
## [1845]
## [1847]
## [1849]
## [1851]
## [1853]
## [1855]
## [1857]
## [1859]
## [1861]
## [1863]
## [1865]
## [1867]
## [1869]
## [1871]
## [1873]
## [1875]
## [1877]
## [1879]
## [1881]
## [1883]
## [1885]
## [1887]
## [1889]
## [1891]
## [1893]
## [1895]
## [1897]
## [1899]
## [1901]
## [1903]
## [1905]
## [1907]
## [1909]
## [1911]
## [1913]
## [1915]
## [1917]
## [1919]
## [1921]
## [1923]
## [1925]
## [1927]
## [1929]
## [1931]
## [1933]
## [1935]
## [1937]
## [1939]
## [1941]
## [1943]
## [1945]
## [1947]
## [1949]
## [1951]
## [1953]
## [1955]
## [1957]
## [1959]
## [1961]
## [1963]
## [1965]
## [1967]
## [1969]
## [1971]
## [1973]
## [1975]
## [1977]
## [1979]
## [1981]
## [1983]
## [1985]
## [1987]
## [1989]
## [1991]
## [1993]
## [1995]
## [1997]
## [1999]
## [2001]
## [2003]
## [2005]
## [2007]
## [2009]
## [2011]
## [2013]
## [2015]
## [2017]
## [2019]
## [2021]
## [2023]
## [2025]
## [2027]
## [2029]
## [2031]
## [2033]
## [2035]
## [2037]
## [2039]
## [2041]
## [2043]
## [2045]
## [2047]
## [2049]
## [2051]
## [2053]
## [2055]
## [2057]
## [2059]
## [2061]
## [2063]
## [2065]
## [2067]
## [2069]
## [2071]
## [2073]
## [2075]
## [2077]
## [2079]
## [2081]
## [2083]
## [2085]
## [2087]
## [2089]
## [2091]
## [2093]
## [2095]
## [2097]
## [2099]
## [2101]
## [2103]
## [2105]
## [2107]
## [2109]
## [2111]
## [2113]
## [2115]
## [2117]
## [2119]
## [2121]
## [2123]
## [2125]
## [2127]
## [2129]
## [2131]
## [2133]
## [2135]
## [2137]
## [2139]
## [2141]
## [2143]
## [2145]
## [2147]
## [2149]
## [2151]
## [2153]
## [2155]
## [2157]
## [2159]
## [2161]
## [2163]
## [2165]
## [2167]
## [2169]
## [2171]
## [2173]
## [2175]
## [2177]
## [2179]
## [2181]
## [2183]
## [2185]
## [2187]
## [2189]
## [2191]
## [2193]
## [2195]
## [2197]
## [2199]
## [2201]
## [2203]
## [2205]
## [2207]
## [2209]
## [2211]
## [2213]
## [2215]
## [2217]
## [2219]
## [2221]
## [2223]
## [2225]
## [2227]
## [2229]
## [2231]
## [2233]
## [2235]
## [2237]
## [2239]
## [2241]
## [2243]
## [2245]
## [2247]
## [2249]
## [2251]
## [2253]
## [2255]
## [2257]
## [2259]
## [2261]
## [2263]
## [2265]
## [2267]
## [2269]
## [2271]
## [2273]
## [2275]
## [2277]
## [2279]
## [2281]
## [2283]
## [2285]
## [2287]
## [2289]
## [2291]
## [2293]
## [2295]
## [2297]
## [2299]
## [2301]
## [2303]
## [2305]
## [2307]
## [2309]
## [2311]
## [2313]
## [2315]
## [2317]
## [2319]
## [2321]
## [2323]
## [2325]
## [2327]
## [2329]
## [2331]
## [2333]
## [2335]
## [2337]
## [2339]
## [2341]
## [2343]
## [2345]
## [2347]
## [2349]
## [2351]
## [2353]
## [2355]
## [2357]
## [2359]
## [2361]
## [2363]
## [2365]
## [2367]
## [2369]
## [2371]
## [2373]
## [2375]
## [2377]
## [2379]
## [2381]
## [2383]
## [2385]
## [2387]
## [2389]
## [2391]
## [2393]
## [2395]
## [2397]
## [2399]
## [2401]
## [2403]
## [2405]
## [2407]
## [2409]
## [2411]
## [2413]
## [2415]
## [2417]
## [2419]
## [2421]
## [2423]
## [2425]
## [2427]
## [2429]
## [2431]
## [2433]
## [2435]
## [2437]
## [2439]
## [2441]
## [2443]
## [2445]
## [2447]
## [2449]
## [2451]
## [2453]
## [2455]
## [2457]
## [2459]
## [2461]
## [2463]
## [2465]
## [2467]
## [2469]
## [2471]
## [2473]
## [2475]
## [2477]
## [2479]
## [2481]
## [2483]
## [2485]
## [2487]
## [2489]
## [2491]
## [2493]
## [2495]
## [2497]
## [2499]
## [2501]
## [2503]
## [2505]
## [2507]
## [2509]
## [2511]
## [2513]
## [2515]
## [2517]
## [2519]
## [2521]
## [2523]
## [2525]
## [2527]
## [2529]
## [2531]
## [2533]
## [2535]
## [2537]
## [2539]
## [2541]
## [2543]
## [2545]
## [2547]
## [2549]
## [2551]
## [2553]
## [2555]
## [2557]
## [2559]
## [2561]
## [2563]
## [2565]
## [2567]
## [2569]
## [2571]
## [2573]
## [2575]
## [2577]
## [2579]
## [2581]
## [2583]
## [2585]
## [2587]
## [2589]
## [2591]
## [2593]
## [2595]
## [2597]
## [2599]
## [2601]
## [2603]
## [2605]
## [2607]
## [2609]
## [2611]
## [2613]
## [2615]
## [2617]
## [2619]
## [2621]
## [2623]
## [2625]
## [2627]
## [2629]
## [2631]
## [2633]
## [2635]
## [2637]
## [2639]
## [2641]
## [2643]
## [2645]
## [2647]
## [2649]
## [2651]
## [2653]
## [2655]
## [2657]
## [2659]
## [2661]
## [2663]
## [2665]
## [2667]
## [2669]
## [2671]
## [2673]
## [2675]
## [2677]
## [2679]
## [2681]
## [2683]
## [2685]
## [2687]
## [2689]
## [2691]
## [2693]
## [2695]
## [2697]
## [2699]
## [2701]
## [2703]
## [2705]
## [2707]
## [2709]
## [2711]
## [2713]
## [2715]
## [2717]
## [2719]
## [2721]
## [2723]
## [2725]
## [2727]
## [2729]
## [2731]
## [2733]
## [2735]
## [2737]
## [2739]
## [2741]
## [2743]
## [2745]
## [2747]
## [2749]
## [2751]
## [2753]
## [2755]
## [2757]
## [2759]
## [2761]
## [2763]
## [2765]
## [2767]
## [2769]
## [2771]
## [2773]
## [2775]
## [2777]
## [2779]
## [2781]
## [2783]
## [2785]
## [2787]
## [2789]
## [2791]
## [2793]
## [2795]
## [2797]
## [2799]
## [2801]
## [2803]
## [2805]
## [2807]
## [2809]
## [2811]
## [2813]
## [2815]
## [2817]
## [2819]
## [2821]
## [2823]
## [2825]
## [2827]
## [2829]
## [2831]
## [2833]
## [2835]
## [2837]
## [2839]
## [2841]
## [2843]
## [2845]
## [2847]
## [2849]
## [2851]
## [2853]
## [2855]
## [2857]
## [2859]
## [2861]
## [2863]
## [2865]
## [2867]
## [2869]
## [2871]
## [2873]
## [2875]
## [2877]
## [2879]
## [2881]
## [2883]
## [2885]
## [2887]
## [2889]
## [2891]
## [2893]
## [2895]
## [2897]
## [2899]
## [2901]
## [2903]
## [2905]
## [2907]
## [2909]
## [2911]
## [2913]
## [2915]
## [2917]
## [2919]
## [2921]
## [2923]
## [2925]
## [2927]
## [2929]
## [2931]
## [2933]
## [2935]
## [2937]
## [2939]
## [2941]
## [2943]
## [2945]
## [2947]
## [2949]
## [2951]
## [2953]
## [2955]
## [2957]
## [2959]
## [2961]
## [2963]
## [2965]
## [2967]
## [2969]
## [2971]
## [2973]
## [2975]
## [2977]
## [2979]
## [2981]
## [2983]
## [2985]
## [2987]
## [2989]
## [2991]
## [2993]
## [2995]
## [2997]
## [2999]
## [3001]
## [3003]
## [3005]
## [3007]
## [3009]
## [3011]
## [3013]
## [3015]
## [3017]
## [3019]
## [3021]
## [3023]
## [3025]
## [3027]
## [3029]
## [3031]
## [3033]
## [3035]
## [3037]
## [3039]
## [3041]
## [3043]
## [3045]
## [3047]
## [3049]
## [3051]
## [3053]
## [3055]
## [3057]
## [3059]
## [3061]
## [3063]
## [3065]
## [3067]
## [3069]
## [3071]
## [3073]
## [3075]
## [3077]
## [3079]
## [3081]
## [3083]
## [3085]
## [3087]
## [3089]
## [3091]
## [3093]
## [3095]
## [3097]
## [3099]
## [3101]
## [3103]
## [3105]
## [3107]
## [3109]
## [3111]
## [3113]
## [3115]
## [3117]
## [3119]
## [3121]
## [3123]
## [3125]
## [3127]
## [3129]
## [3131]
## [3133]
## [3135]
## [3137]
## [3139]
## [3141]
## [3143]
## [3145]
## [3147]
## [3149]
## [3151]
## [3153]
## [3155]
## [3157]
## [3159]
## [3161]
## [3163]
## [3165]
## [3167]
## [3169]
## [3171]
## [3173]
## [3175]
## [3177]
## [3179]
## [3181]
## [3183]
## [3185]
## [3187]
## [3189]
## [3191]
## [3193]
## [3195]
## [3197]
## [3199]
## [3201]
## [3203]

```

```

## loaded via a namespace (and not attached):
## [1] lubridate_1.8.0   lattice_0.20-45  assertthat_0.2.1 digest_0.6.29
## [5] utf8_1.2.2       R6_2.5.1      cellranger_1.1.0 backports_1.4.1
## [9] reprex_2.0.2     evaluate_0.16 httr_1.4.4      highr_0.9
## [13] pillar_1.8.1    rlang_1.0.5    readxl_1.4.1   rstudioapi_0.14
## [17] Matrix_1.5-0    rmarkdown_2.16 labeling_0.4.2  splines_4.2.1
## [21] munsell_0.5.0   broom_1.0.1    compiler_4.2.1 modelr_0.1.9
## [25] xfun_0.32       pkgconfig_2.0.3 mgcv_1.8-40   htmltools_0.5.3
## [29] tidyselect_1.1.2 fansi_1.0.3    crayon_1.5.1   tzdb_0.3.0
## [33] dbplyr_2.2.1    withr_2.5.0    grid_4.2.1    nlme_3.1-159
## [37] jsonlite_1.8.0   gtable_0.3.1  lifecycle_1.0.2 DBI_1.1.3
## [41] magrittr_2.0.3   scales_1.2.1   cli_3.4.0     stringi_1.7.8
## [45] farver_2.1.1    fs_1.5.2      xml2_1.3.3   ellipsis_0.3.2
## [49] generics_0.1.3   vctrs_0.4.1   tools_4.2.1   glue_1.6.2
## [53] hms_1.1.2       fastmap_1.1.0 yaml_2.3.5   colorspace_2.0-3
## [57] rvest_1.0.3     haven_2.5.1

#=====
# Take control of a qualitative color scheme
#=====

#' ---
#' author: "Jenny Bryan"
#' output:
#'   html_document:
#'     keep_md: TRUE
#'   ---

## setup, include = FALSE
library(knitr)

opts_chunk$set(fig.path = 'figs/bryan/colors-', error = TRUE)

```

Note: this HTML is made by applying `knitr::spin()` to an R script. So the narrative is very minimal.

```
library(ggplot2)  
# install.packages("RColorBrewer")  
  
library(RColorBrewer)  
  
?RColorBrewer
```

pick a way to load the data

```

## $ year      : int [1:1704] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
## $ lifeExp   : num [1:1704] 28.8 30.3 32 34 36.1 ...
## $ pop       : int [1:1704] 8425333 9240934 10267083 11537966 13079460 14880372 12881816 13867957 163...
## $ gdpPercap: num [1:1704] 779 821 853 836 740 ...

```

let just look at four countries

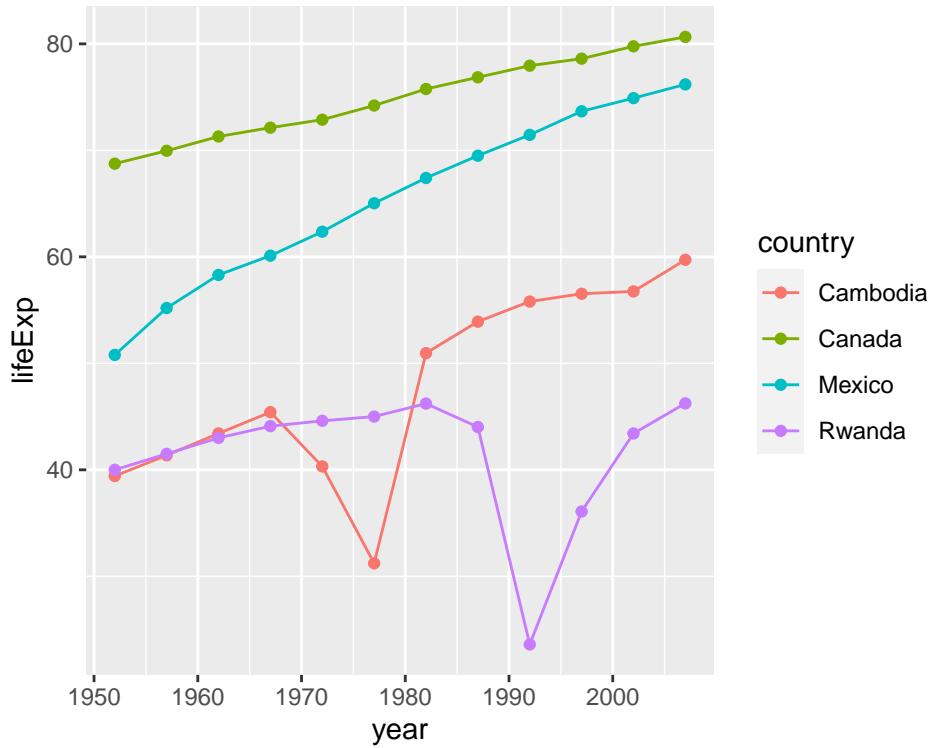
```

jCountries <- c("Canada", "Rwanda", "Cambodia", "Mexico")

x <- droplevels(subset(gapminder, country %in% jCountries))

ggplot(x, aes(x = year, y = lifeExp, color = country)) +
  geom_line() + geom_point()

```



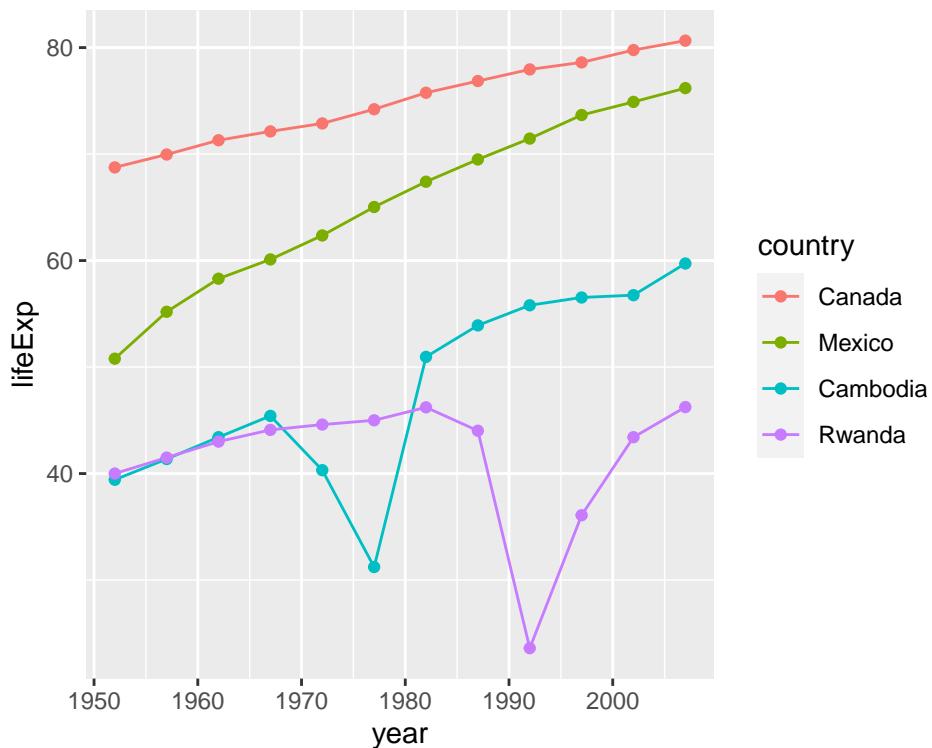
reorder the country factor to reflect lifeExp in 2007

```

x <- transform(x, country = reorder(country, -1 * lifeExp, max))

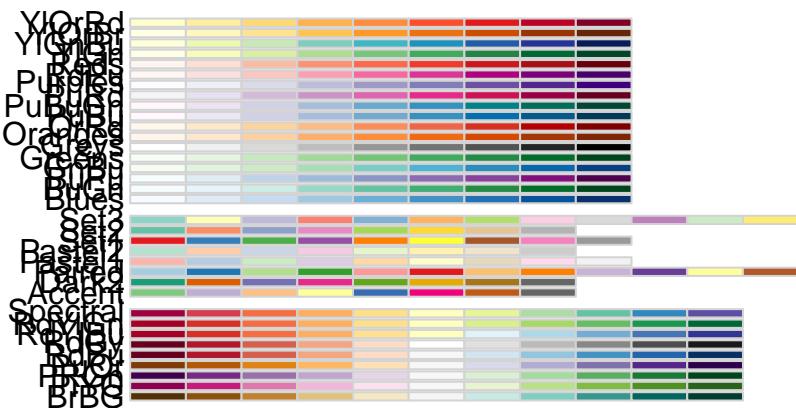
ggplot(x, aes(x = year, y = lifeExp, color = country)) +
  geom_line() + geom_point()

```



look at the RColorBrewer color palettes

```
display.brewer.all()
```



focus on the qualitative palettes

```
display.brewer.all(type = "qual")
```



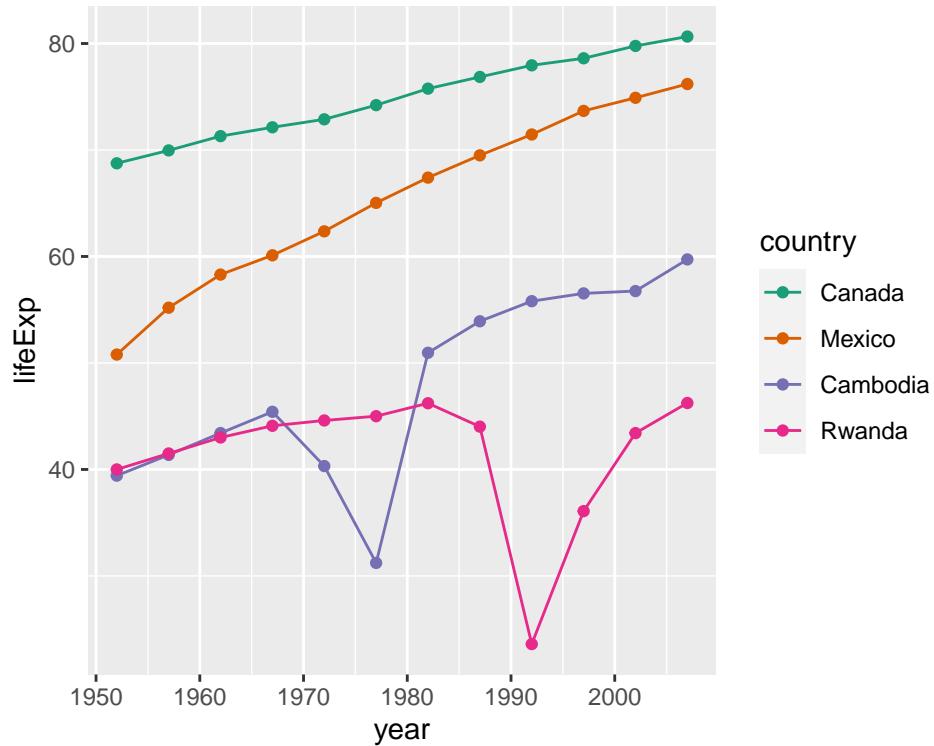
pick some colors

```
jColors = brewer.pal(n = 8, "Dark2")[seq_len(nlevels(x$country))]

names(jColors) <- levels(x$country)
```

remake the plot with our new colors

```
ggplot(x, aes(x = year, y = lifeExp, color = country)) +
  geom_line() + geom_point() +
  scale_color_manual(values = jColors)
```



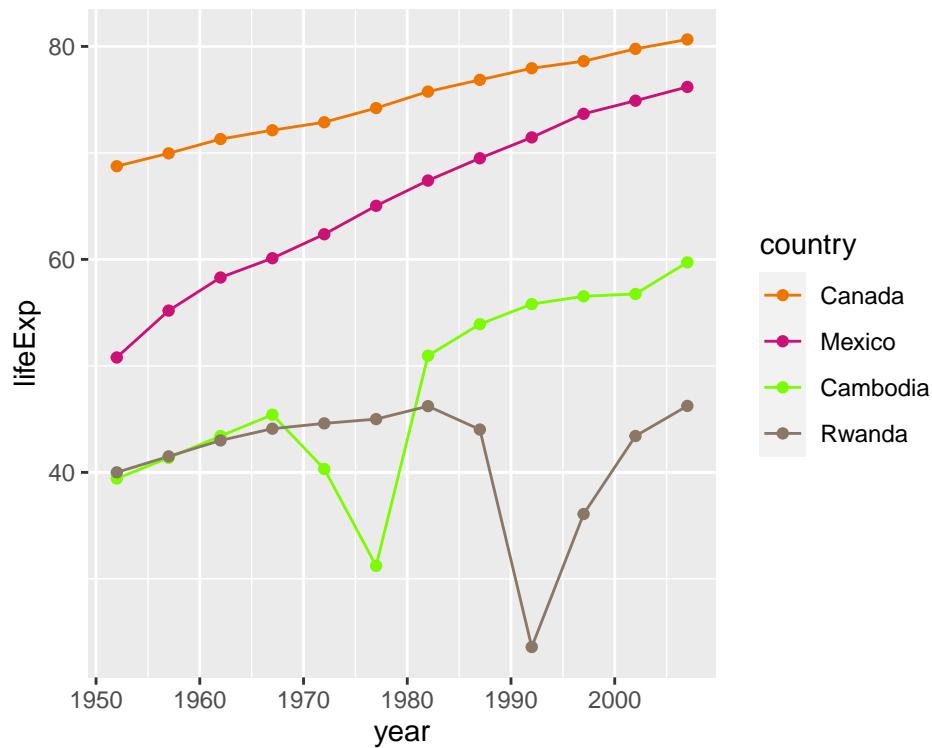
pick some super ugly colors for shock value

```
kColors = c("darkorange2", "deeppink3", "lawngreen", "peachpuff4")

names(kColors) <- levels(x$country)
```

remake the plot with our ugly colors

```
ggplot(x, aes(x = year, y = lifeExp, color = country)) +
  geom_line() + geom_point() +
  scale_color_manual(values = kColors)
```



```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.5 LTS
##
## Matrix products: default
## BLAS:    /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK:  /usr/lib/x86_64-linux-gnu/openblas-pthread/liblapack.so.3
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8          LC_NUMERIC=C
## [3] LC_TIME=en_US.UTF-8          LC_COLLATE=en_US.UTF-8
## [5] LC_MONETARY=en_US.UTF-8       LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=en_US.UTF-8          LC_NAME=C
## [9] LC_ADDRESS=C                  LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8   LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats      graphics   grDevices  utils      datasets   methods    base
##
## other attached packages:
## [1] RColorBrewer_1.1-3 ggthemes_4.2.4    forcats_0.5.2    stringr_1.4.1
## [5] purrrr_0.3.4      readr_2.1.2      tidyverse_1.3.1
## [9] dplyr_1.0.10     gapminder_0.3.0  ggplot2_3.3.6   tibble_3.1.7
## [13] knitr_1.40
```

```

## 
## loaded via a namespace (and not attached):
## [1] lubridate_1.8.0   lattice_0.20-45  assertthat_0.2.1  digest_0.6.29
## [5] utf8_1.2.2       R6_2.5.1      cellranger_1.1.0 backports_1.4.1
## [9] reprex_2.0.2     evaluate_0.16 httr_1.4.4       highr_0.9
## [13] pillar_1.8.1     rlang_1.0.5    readxl_1.4.1    rstudioapi_0.14
## [17] Matrix_1.5-0     rmarkdown_2.16 labeling_0.4.2   splines_4.2.1
## [21] munsell_0.5.0    broom_1.0.1    compiler_4.2.1  modelr_0.1.9
## [25] xfun_0.32        pkgconfig_2.0.3 mgcv_1.8-40    htmltools_0.5.3
## [29] tidyselect_1.1.2  fansi_1.0.3    crayon_1.5.1   tzdb_0.3.0
## [33] dbplyr_2.2.1     withr_2.5.0    grid_4.2.1     nlme_3.1-159
## [37] jsonlite_1.8.0    gtable_0.3.1   lifecycle_1.0.2 DBI_1.1.3
## [41] magrittr_2.0.3    scales_1.2.1   cli_3.4.0      stringi_1.7.8
## [45] farver_2.1.1     fs_1.5.2      xml2_1.3.3    ellipsis_0.3.2
## [49] generics_0.1.3    vctrs_0.4.1   tools_4.2.1    glue_1.6.2
## [53] hms_1.1.2        fastmap_1.1.0  yaml_2.3.5    colorspace_2.0-3
## [57] rvest_1.0.3       haven_2.5.1

#=====
# Bubble and line plots, lots of customization
#=====

#' ---
#' author: "Jenny Bryan"
#' output:
#'   html_document:
#'     keep_md: TRUE
#' ---

#+ setup, include = FALSE
library(knitr)

opts_chunk$set(fig.path = 'figs/bryan/shock-awe-', error = TRUE)

```

Note: this HTML is made by applying `knitr:::spin()` to an R script. So the narrative is very minimal.

```
library(ggplot2)
```

pick a way to load the data

```

#gdURL <- "http://tiny.cc/gapminder"
#gapminder <- read.delim(file = gdURL)
#gapminder <- read.delim("gapminderDataFiveYear.tsv")

library(gapminder)
str(gapminder)

## # tibble [1,704 x 6] (S3: tbl_df/tbl/data.frame)
## $ country : Factor w/ 142 levels "Afghanistan",...: 1 1 1 1 1 1 1 1 1 ...
## $ continent: Factor w/ 5 levels "Africa","Americas",...: 3 3 3 3 3 3 3 3 3 ...
## $ year     : int [1:1704] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
## $ lifeExp  : num [1:1704] 28.8 30.3 32 34 36.1 ...
## $ pop      : int [1:1704] 8425333 9240934 10267083 11537966 13079460 14880372 12881816 13867957 163
## $ gdpPercap: num [1:1704] 779 821 853 836 740 ...

```

drop Oceania

```
gapminder <- droplevels(subset(gapminder, continent != "Oceania"))
```

Note that the gapminder package ships with color schemes for countries and continents.

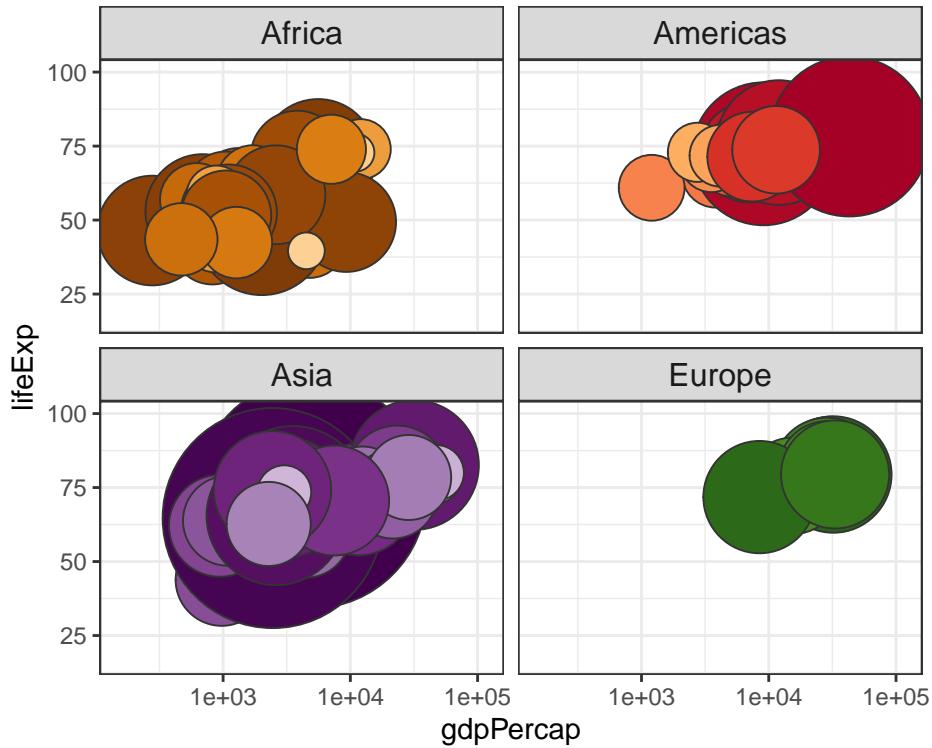
```
head(country_colors)
```

```
##          Nigeria           Egypt      Ethiopia Congo, Dem. Rep.
## "#7F3B08" "#833D07"      "#873F07"      "#8B4107"
##    South Africa           Sudan
## "#8F4407" "#934607"

jYear <- 2007 # this can obviously be changed
jPch <- 21
jDarkGray <- 'grey20'
jXlim <- c(150, 115000)
jYlim <- c(16, 100)

ggplot(subset(gapminder, year == jYear),
       aes(x = gdpPercap, y = lifeExp)) +
  scale_x_log10(limits = jXlim) + ylim(jYlim) +
  geom_point(
    aes(size = sqrt(pop / pi)),
    pch = jPch,
    color = jDarkGray,
    show_guide = FALSE
  ) +
  scale_size_continuous(range = c(1, 40)) +
  facet_wrap(~ continent) + coord_fixed(ratio = 1 / 43) +
  aes(fill = country) + scale_fill_manual(values = country_colors) +
  theme_bw() + theme(strip.text = element_text(size = rel(1.1)))

## Warning: `show_guide` has been deprecated. Please use `show.legend` instead.
```



```
ggplot(gapminder, aes(x = year, y = lifeExp, group = country)) +
  geom_line(lwd = 1, show_guide = FALSE) + facet_wrap(~ continent) +
  aes(color = country) + scale_color_manual(values = country_colors) +
  theme_bw() + theme(strip.text = element_text(size = rel(1.1)))
```

## Warning: `show\_guide` has been deprecated. Please use `show.legend` instead.

