Visual Analysis

Shardendu Jha

11/14/2020

Part-1

Excercise - 1

```
#Loading the libaries

library(ggplot2)
library(ggthemes)
library(nlme)
library(gganimate)
library(gapminder)
library(ggExtra)
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(reshape2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:nlme':
##
##     collapse
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(nycflights13)
library(ggcorrplot)
library(waffle)
library(tidyr)
```

```
##
```

```
## Attaching package: 'tidyr'

## The following object is masked from 'package:reshape2':
##
##     smiths
```

```
library(scales)
```

```
##
## Attaching package: 'scales'

## The following objects are masked from 'package:psych':
##
##     alpha, rescale
```

```
library(ggalt)
```

```
## Registered S3 methods overwritten by 'ggalt':
##   method                  from
##   grid.draw.absoluteGrob  ggplot2
##   grobHeight.absoluteGrob ggplot2
##   grobWidth.absoluteGrob  ggplot2
##   grobX.absoluteGrob      ggplot2
##   grobY.absoluteGrob      ggplot2
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following objects are masked from 'package:reshape2':
##
##     dcast, melt
```

```
library(extrafont)
```

```
## Registering fonts with R
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:data.table':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(DT)
library(grid)
library(gridExtra)
```

```
##
```

```
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(prettydoc)
library(devtools)
```

```
## Loading required package: usethis
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v tibble  3.0.4     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x psych::%+%()             masks ggplot2::%+%()
## x scales::alpha()          masks psych::alpha(), ggplot2::alpha()
## x lubridate::as.difftime() masks base::as.difftime()
## x data.table::between()    masks dplyr::between()
## x readr::col_factor()      masks scales::col_factor()
## x dplyr::collapse()        masks nlme::collapse()
## x gridExtra::combine()     masks dplyr::combine()
## x lubridate::date()        masks base::date()
## x purrr::discard()         masks scales::discard()
## x dplyr::filter()          masks stats::filter()
## x data.table::first()      masks dplyr::first()
## x lubridate::hour()        masks data.table::hour()
## x lubridate::intersect()   masks base::intersect()
## x lubridate::isoweek()     masks data.table::isoweek()
## x dplyr::lag()             masks stats::lag()
## x data.table::last()       masks dplyr::last()
## x lubridate::mday()        masks data.table::mday()
## x lubridate::minute()      masks data.table::minute()
## x lubridate::month()       masks data.table::month()
## x lubridate::quarter()     masks data.table::quarter()
## x lubridate::second()      masks data.table::second()
## x lubridate::setdiff()     masks base::setdiff()
## x purrr::transpose()       masks data.table::transpose()
## x lubridate::union()       masks base::union()
## x lubridate::wday()        masks data.table::wday()
## x lubridate::week()        masks data.table::week()
## x lubridate::yday()        masks data.table::yday()
## x lubridate::year()        masks data.table::year()
```

```
library(ggdark)
library(here)
```

```
## here() starts at /Users/shardendujha/Desktop/pop
```

```
library(png)
library(gifski)
library(forcats)
library(tufte)
```

```r
library(colorspace)
library(viridisLite)
library(Zelig)
```

```
## Loading required package: boot
##
## Attaching package: 'boot'
## The following object is masked from 'package:psych':
##
##     logit
## Loading required package: MASS
##
## Attaching package: 'MASS'
## The following object is masked from 'package:dplyr':
##
##     select
## Loading required package: sandwich
## ZELIG (Versions 4.2-1, built: 2013-09-12)
##
## +-----------------------------------------------------------------+
## |  Please refer to http://gking.harvard.edu/zelig for full        |
## |  documentation or help.zelig() for help with commands and       |
## |  models support by Zelig.                                       |
## |                                                                 |
## |  Zelig project citations:                                       |
## |    Kosuke Imai, Gary King, and Olivia Lau.  (2009).             |
## |    ``Zelig: Everyone's Statistical Software,''                  |
## |    http://gking.harvard.edu/zelig                               |
## |   and                                                           |
## |    Kosuke Imai, Gary King, and Olivia Lau. (2008).              |
## |    ``Toward A Common Framework for Statistical Analysis         |
## |    and Development,'' Journal of Computational and              |
## |    Graphical Statistics, Vol. 17, No. 4 (December)              |
## |    pp. 892-913.                                                 |
## |                                                                 |
## |   To cite individual Zelig models, please use the citation      |
## |   format printed with each model run and in the documentation.  |
## +-----------------------------------------------------------------+
##
## Attaching package: 'Zelig'
## The following object is masked from 'package:gridExtra':
##
##     combine
## The following object is masked from 'package:scales':
##
##     alpha
## The following objects are masked from 'package:dplyr':
##
```

```
##     combine, summarize

## The following objects are masked from 'package:psych':
##
##     alpha, describe, sim

## The following object is masked from 'package:ggplot2':
##
##     alpha

## The following object is masked from 'package:utils':
##
##     cite
```

```
library(formatR)
library(DiagrammeR)
library(xaringan)
library(ggridges)
library(GGally)
```

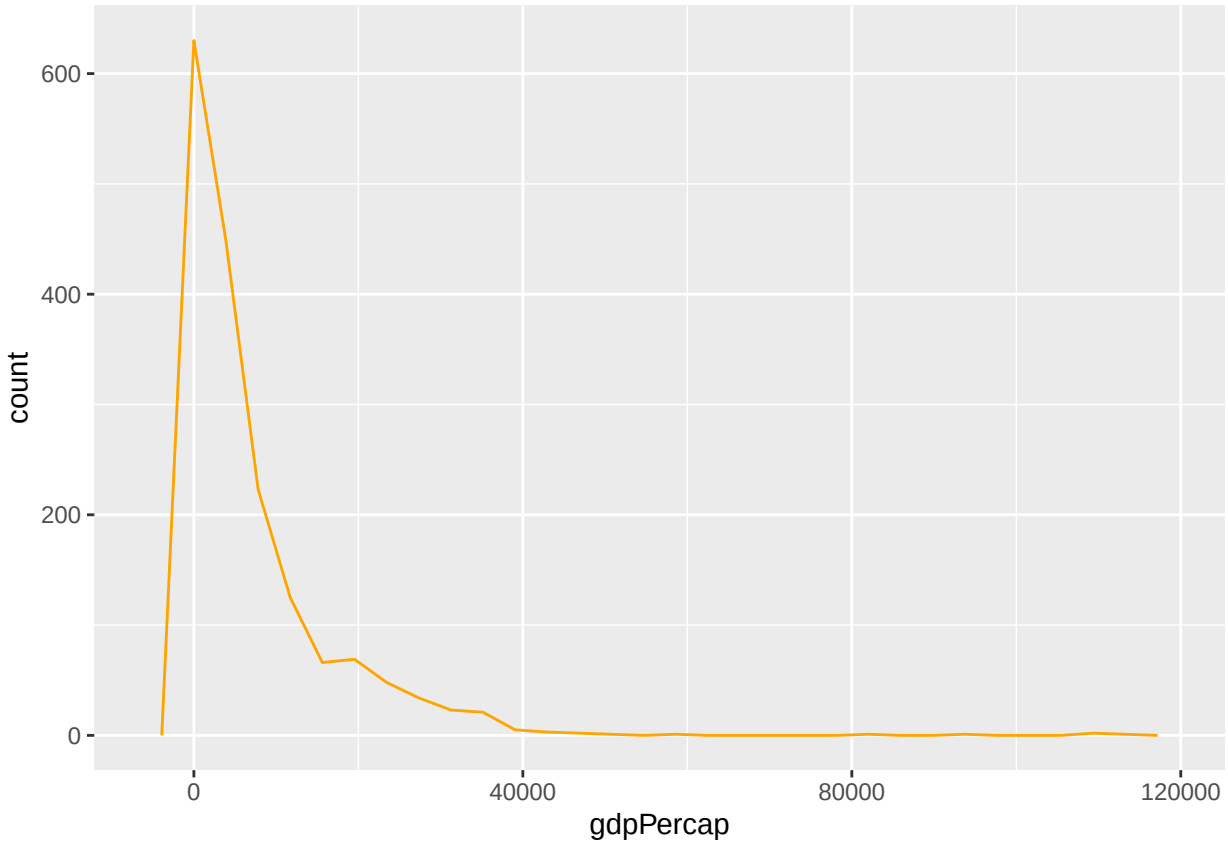
```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
#Defining the general colors to avoid hard coding
fill_color = 'darkred'
decoration_color = 'cyan4'
main_color = 'darkolivegreen'
sub_color = 'orange'

# customize theme
my_first_theme <- theme_gray() + theme(

  plot.title=element_text(size=20,face="bold", family="American Typewriter",color=decoration_color,
                          hjust=0.5,
                          lineheight=1.2),
  plot.subtitle=element_text(size= 15, family="American Typewriter",color=decoration_color,
                             face="bold",
                             hjust=0.5),
  plot.caption=element_text(size=15),
  plot.background=element_blank(),
  legend.key=element_blank(),
  legend.title=element_blank(),
  panel.grid.major=element_blank(),
  panel.grid.minor=element_blank(),
  panel.border=element_blank(),
  legend.position=c(0.815, 0.27),
  panel.background=element_blank(),
  strip.background=element_blank(),
  strip.text =element_text(size=12, color=decoration_color),
  axis.title.x=element_text(vjust=10,size=15),
  axis.ticks =element_blank(),
  axis.line =element_line(colour=decoration_color,size=0.3, linetype="dashed"),
  axis.text = element_text(size = 10,angle = 30,vjust = .5),

)
```

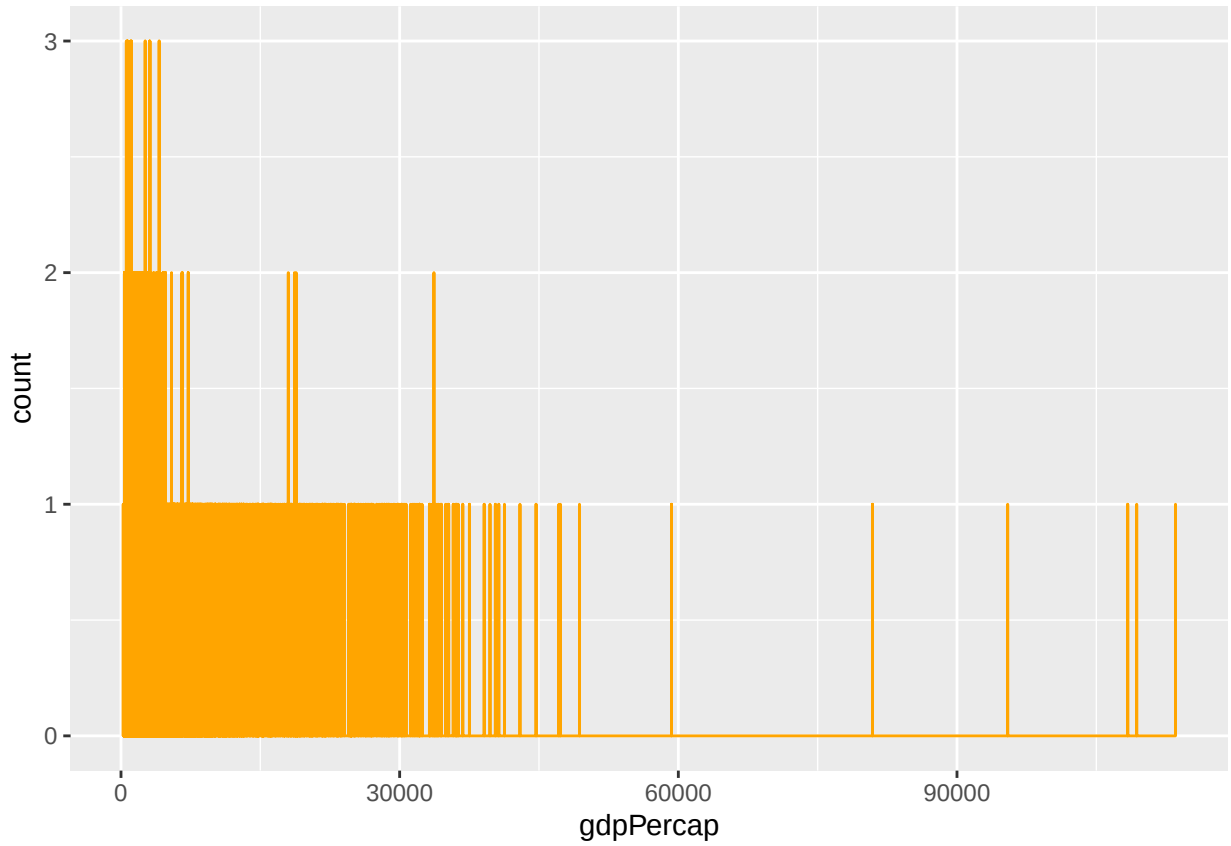
Excercise - 2

```
# simple chart
ggplot(gapminder, aes(gdpPercap)) +
  geom_freqpoly(color = sub_color)
```

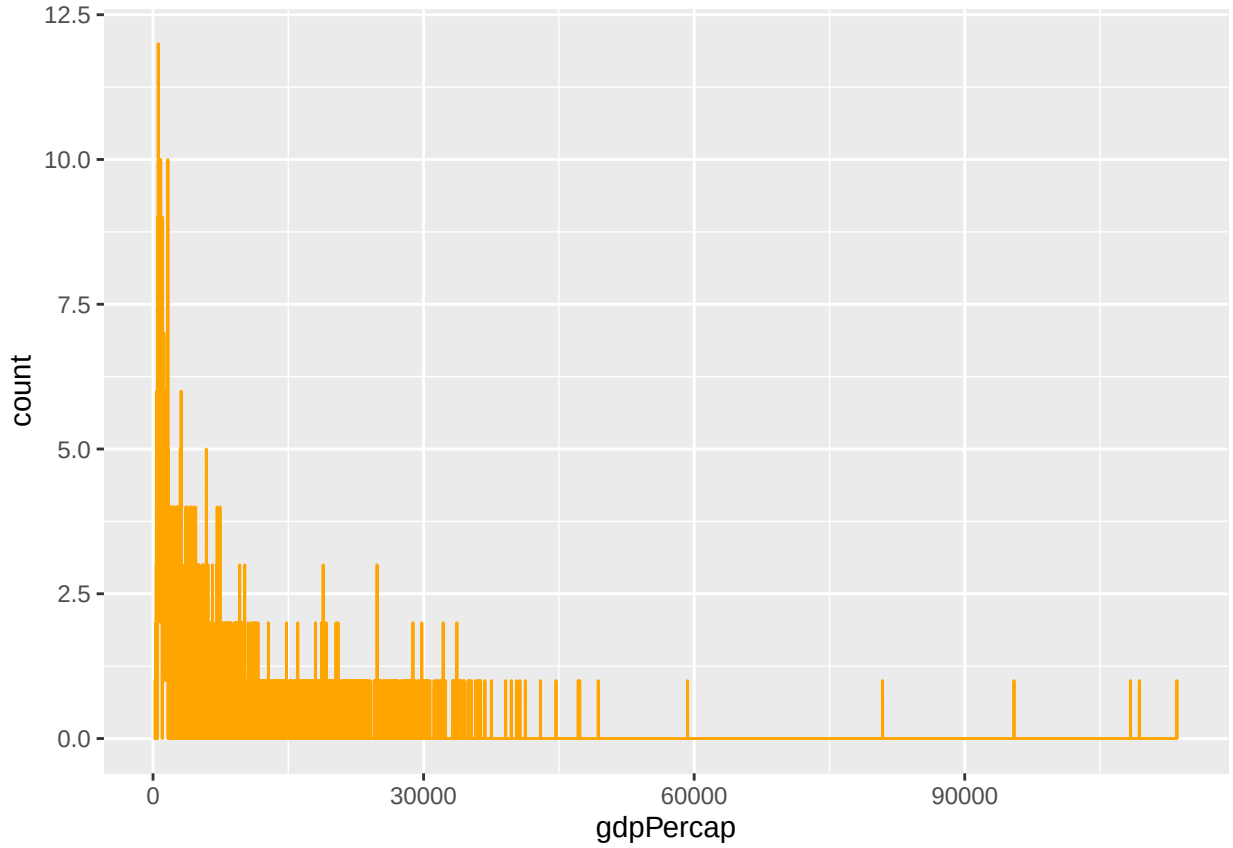
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
#display with bin deatils
ggplot(gapminder, aes(gdpPercap)) +
  geom_freqpoly(colour = sub_color, binwidth = 0.8)
```

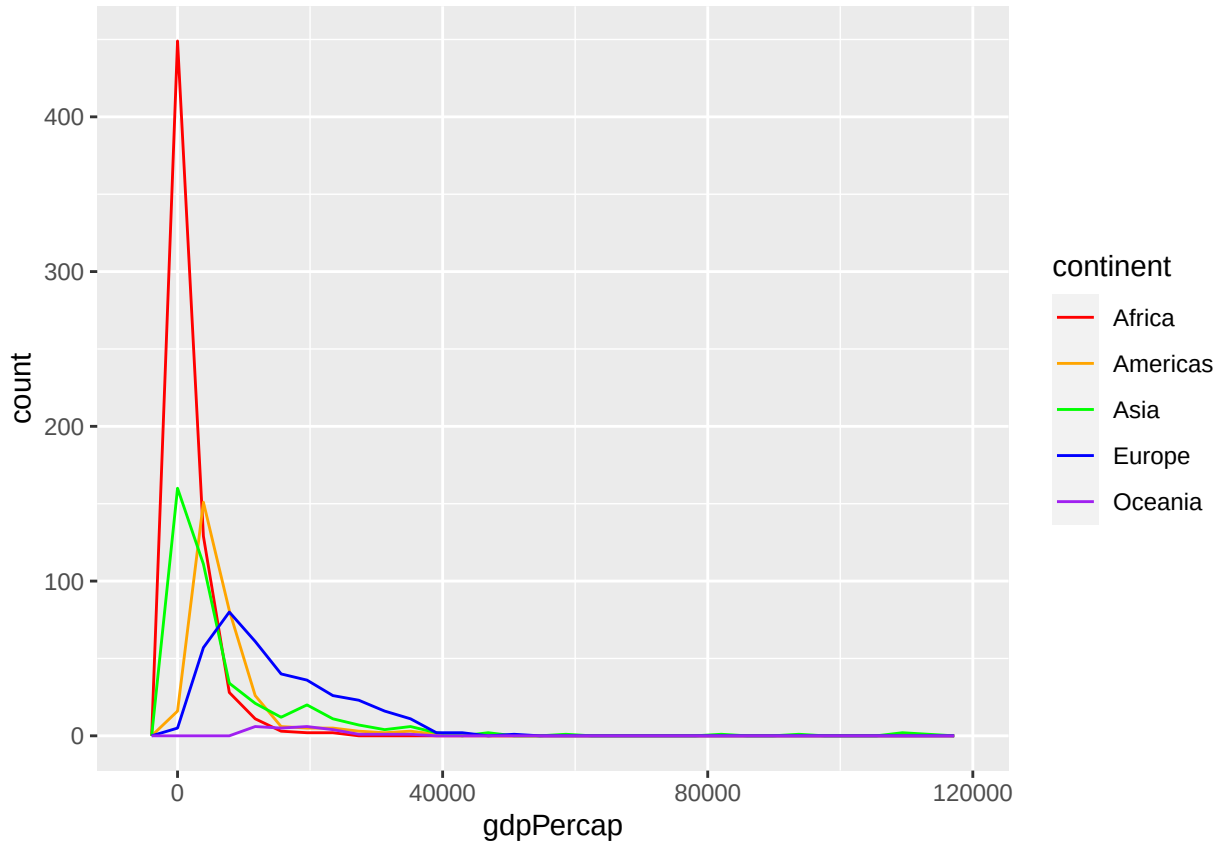



```
#Changing the bin width (less details)
ggplot(gapminder, aes(gdpPercap)) +
  geom_freqpoly(colour = sub_color, binwidth = 10)
```



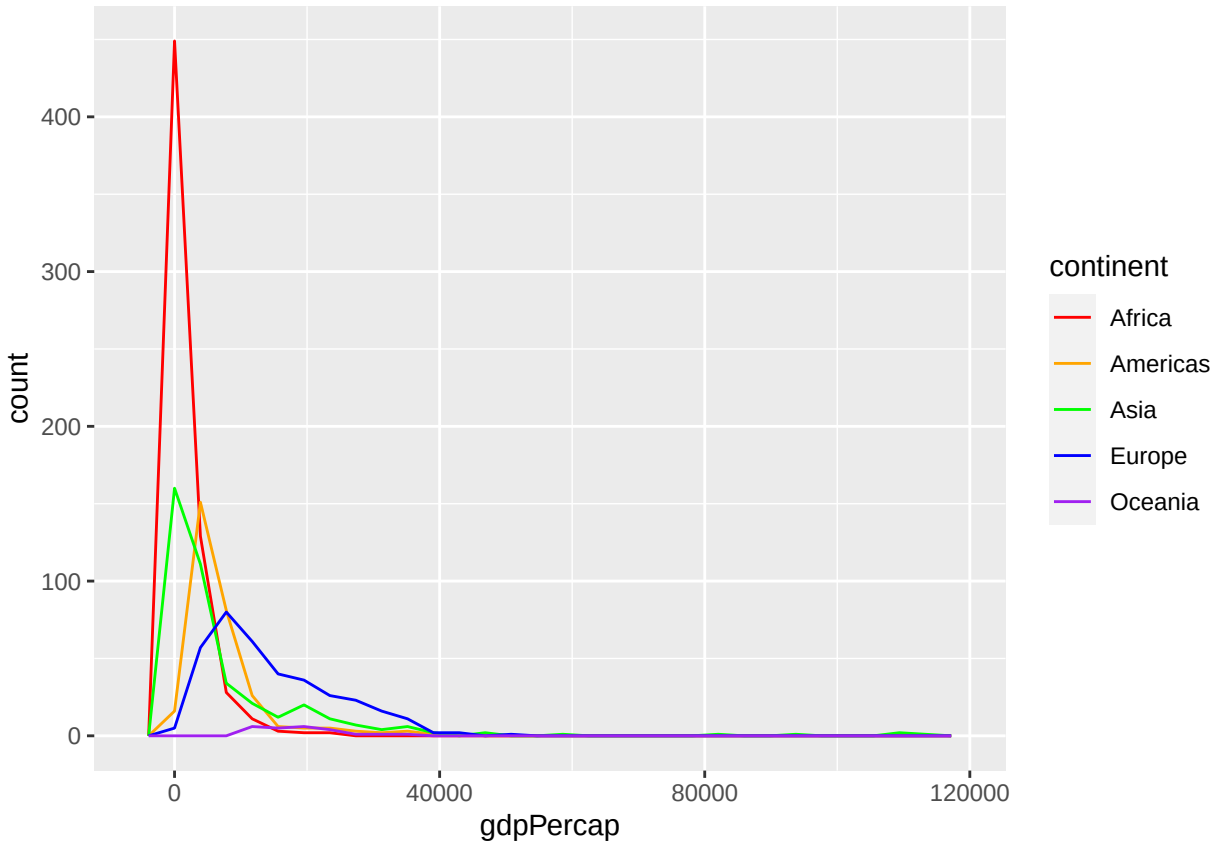

```
# display visual encoding as a color
ggplot(gapminder, aes(gdpPercap, colour = continent)) +
  geom_freqpoly() +
  scale_color_manual(values=c('red','orange','green','blue','purple'))
```

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




```
# display with range value
ggplot(gapminder, aes(gdpPercap, colour = continent)) +
  geom_freqpoly() +
  scale_color_manual(values=c('red','orange','green','blue','purple'))
```

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

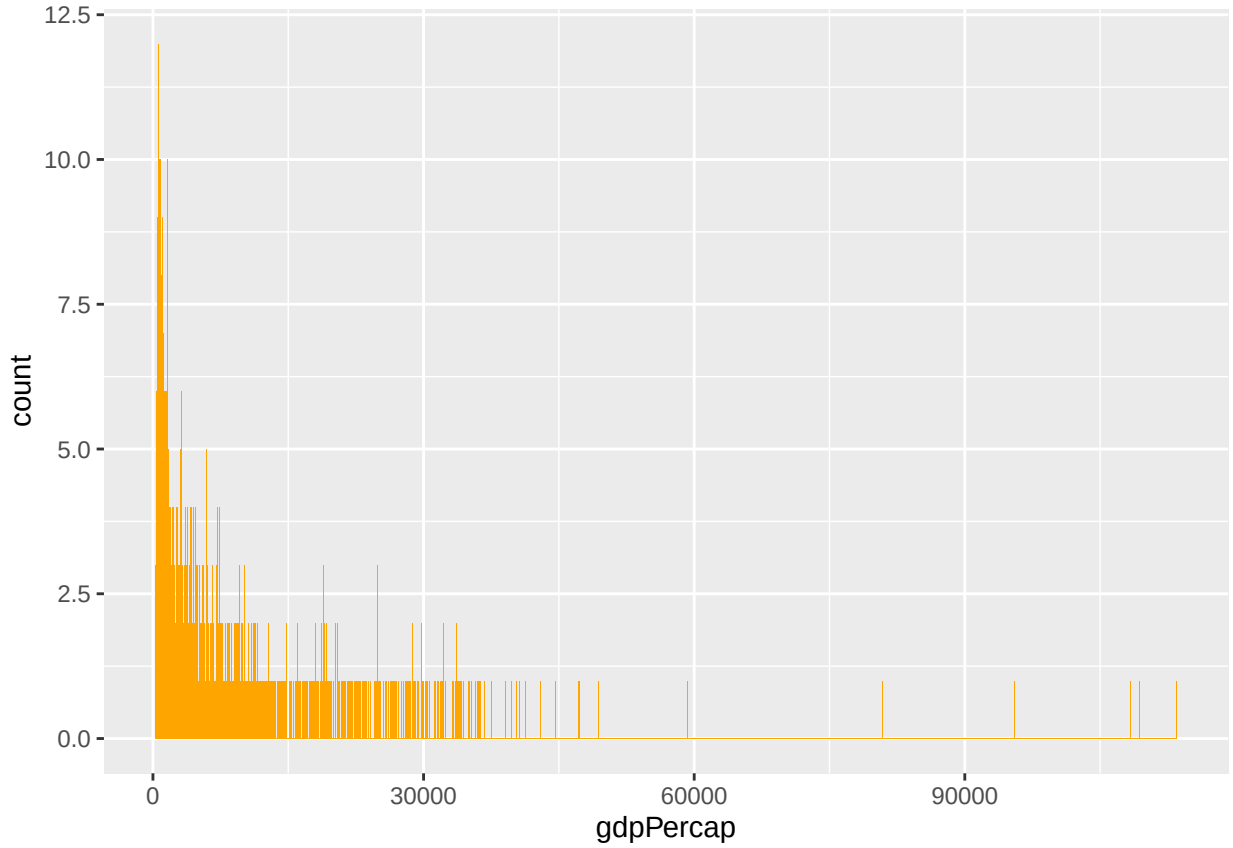



```
  xlim(20000, 40000)
```

```
## <ScaleContinuousPosition>
##  Range:
##  Limits: 2e+04 -- 4e+04
```

Excercise - 3

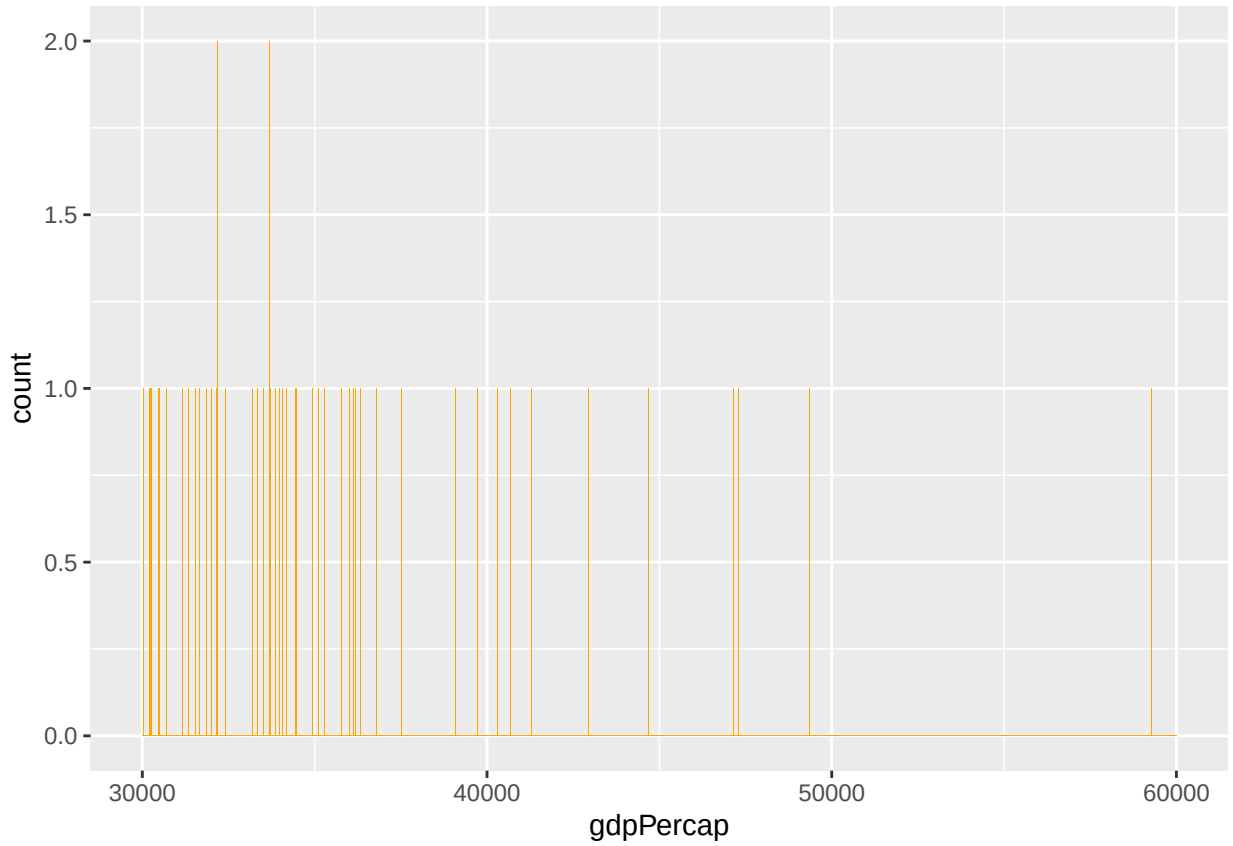
```
# check with histogram chart
ggplot(gapminder, aes(gdpPercap)) +
  geom_histogram(fill= sub_color, binwidth =10)
```




```
# check range with xrange with zoom
ggplot(gapminder, aes(gdpPercap)) +
  geom_histogram(fill=sub_color, binwidth =10) +
  xlim(30000, 60000)
```

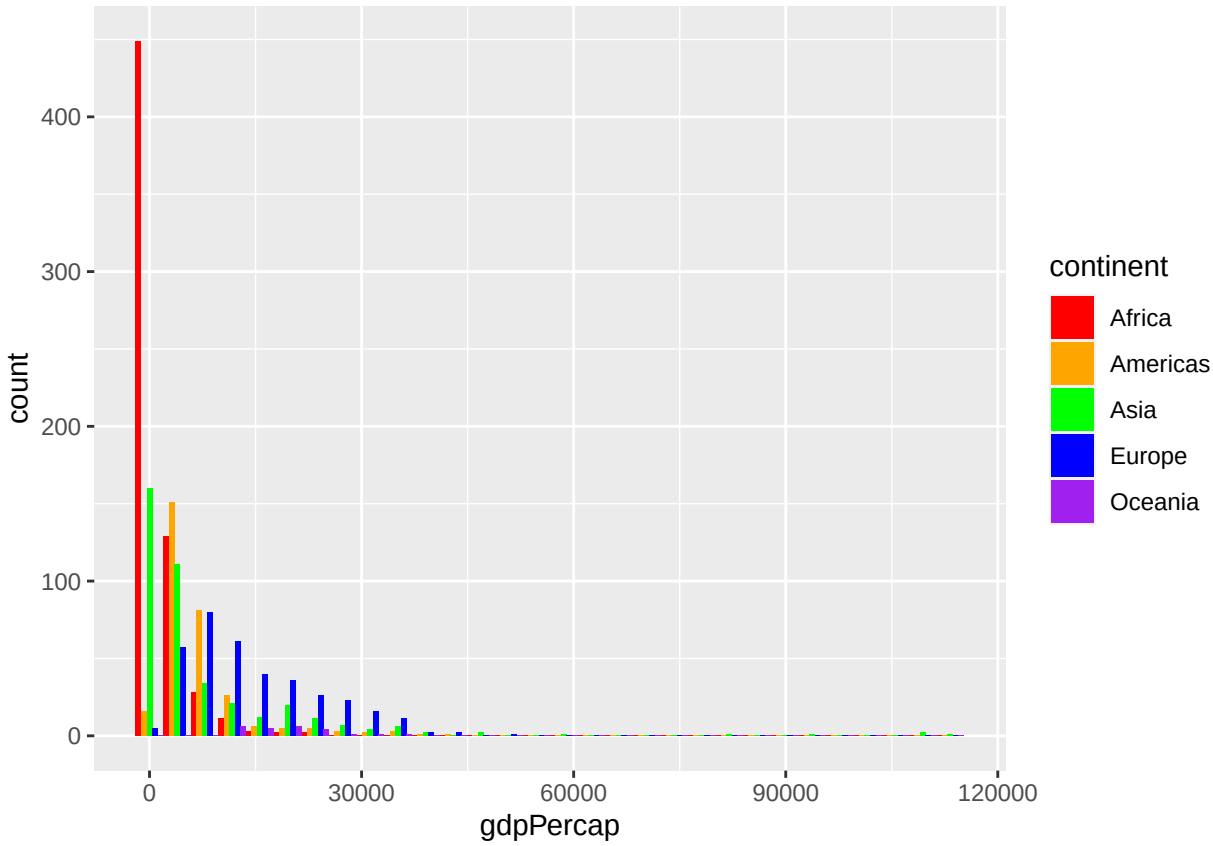
```
## Warning: Removed 1653 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 2 rows containing missing values (geom_bar).
```

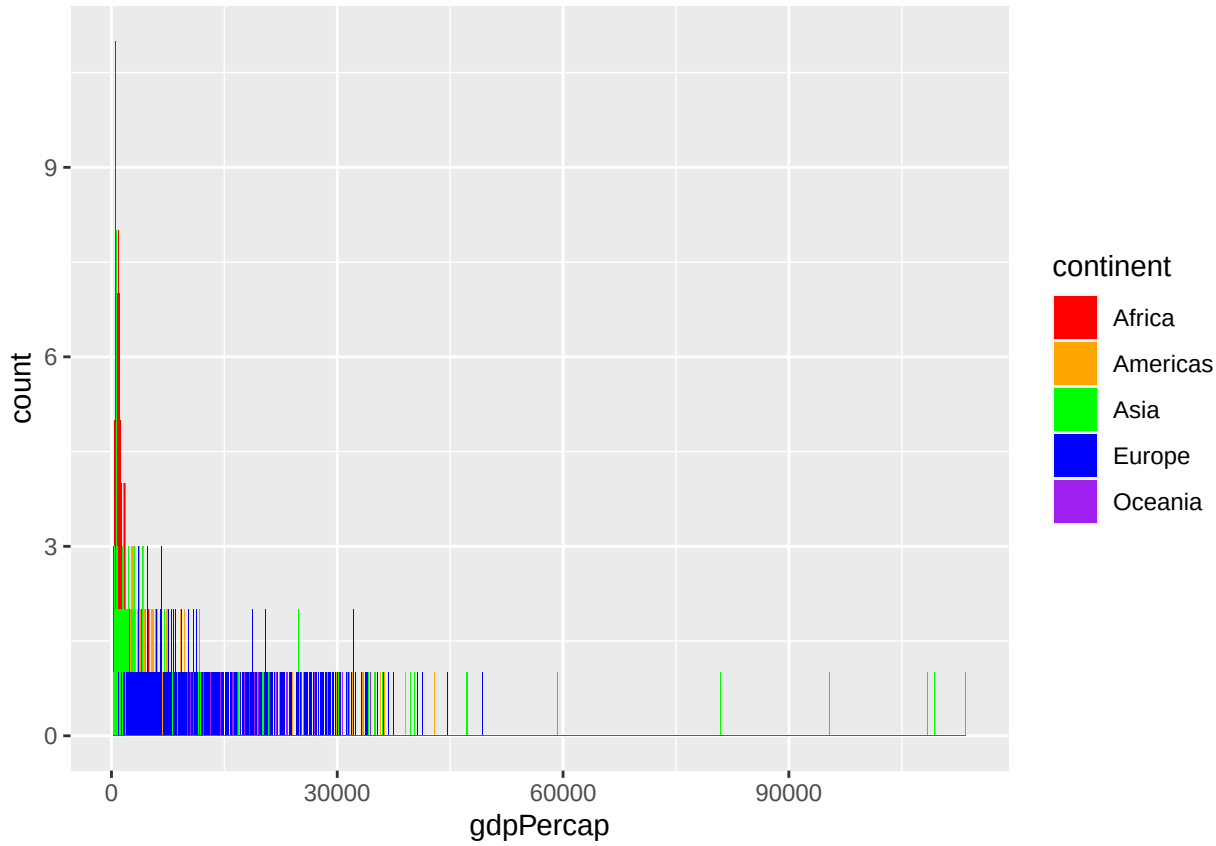



```
#Histogram for different cut options
ggplot(gapminder, aes(gdpPercap, fill = continent)) +
  geom_histogram(position = "dodge") +
  scale_fill_manual(values=c('red','orange','green','blue','purple'))
```

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

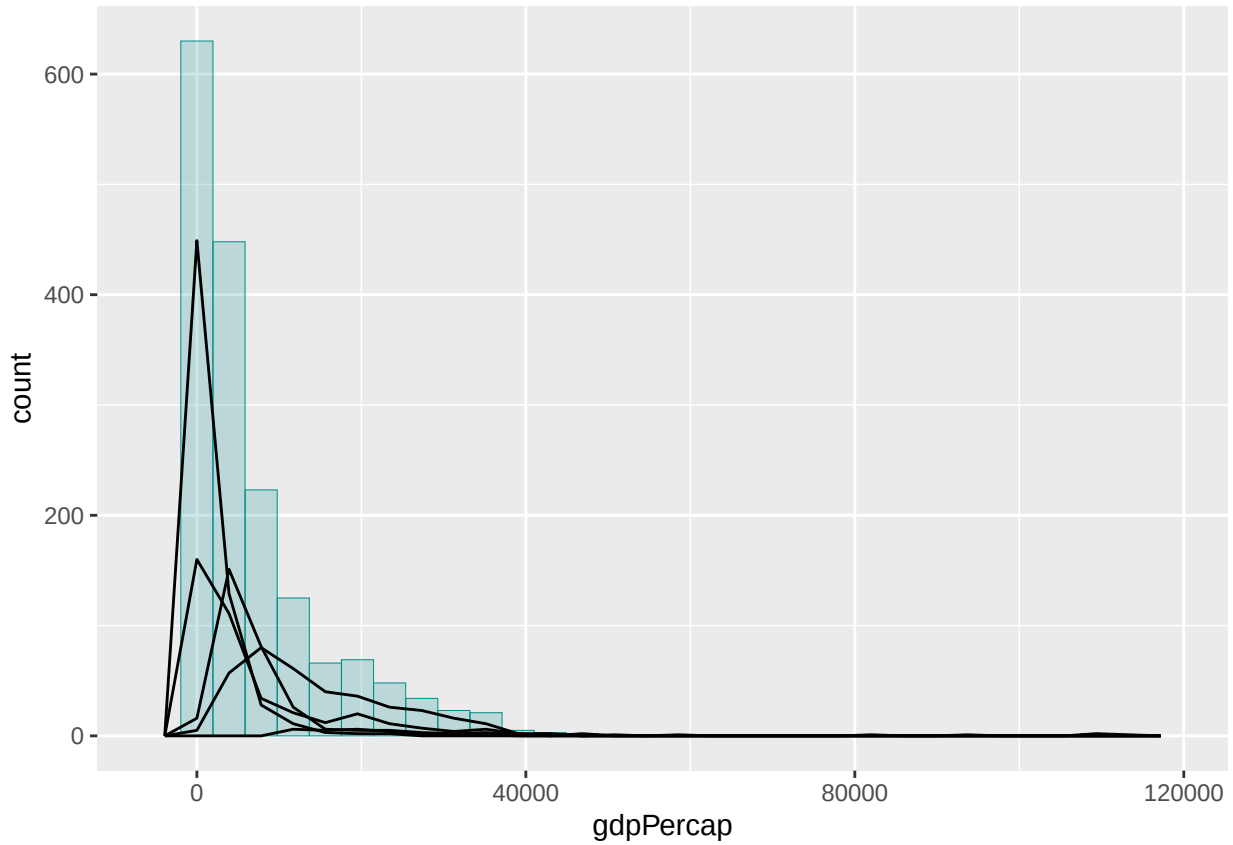



```
#Changing the bin options
ggplot(gapminder, aes(gdpPercap, fill = continent)) +
  geom_histogram(position = "dodge",binwidth = 10) +
  scale_fill_manual(values=c('red','orange','green','blue','purple'))
```



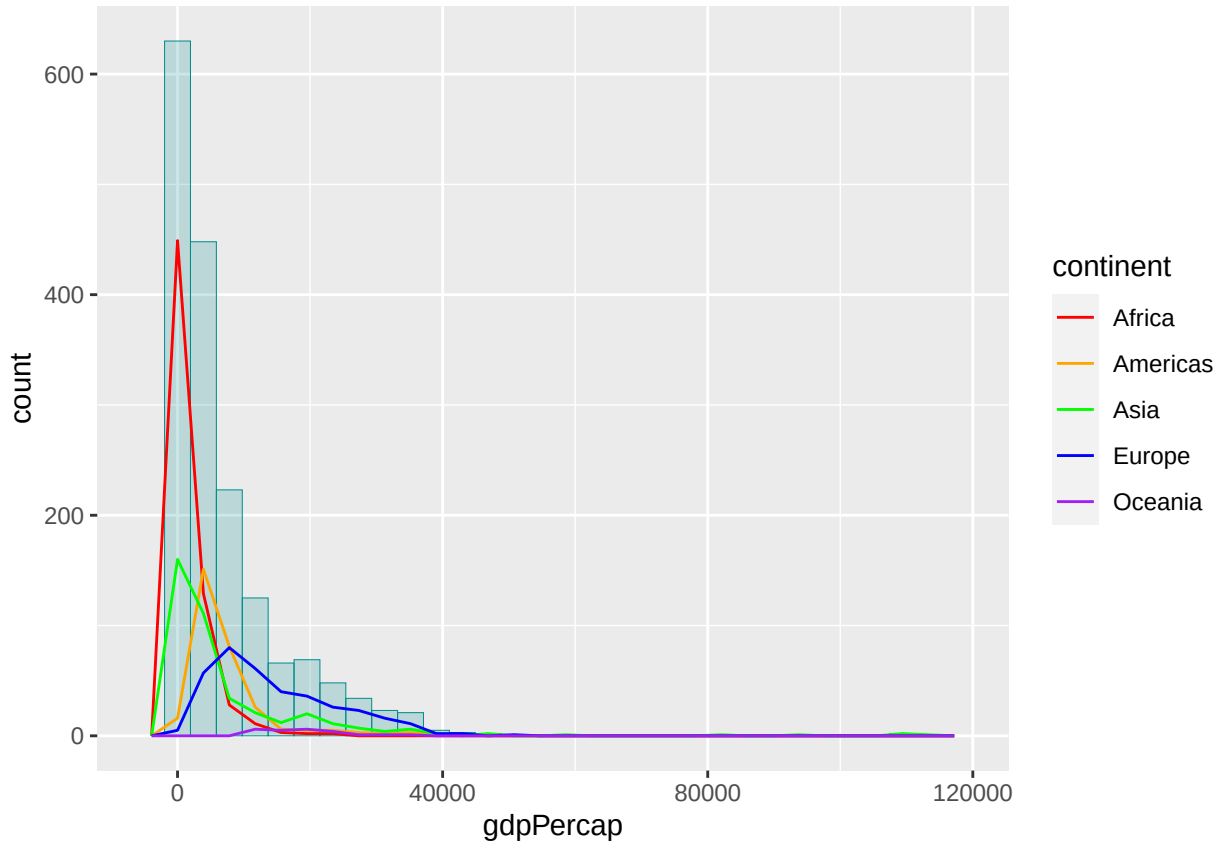

```
# two diffenrent chart comes together

ggplot(gapminder, aes(gdpPercap, fill = continent)) +
  geom_histogram(colour=decoration_color, fill = decoration_color, alpha = 0.2, size =0) +
  geom_freqpoly()+
  scale_colour_manual(values=c('red','orange','green','blue','purple'))
```

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




```
# two different chart comes together with range
ggplot(gapminder, aes(gdpPercap, colour = continent)) +
  geom_histogram(colour=decoration_color, fill = decoration_color, alpha = 0.2, size =0) +
  geom_freqpoly()+
  scale_colour_manual(values=c('red','orange','green','blue','purple'))
```

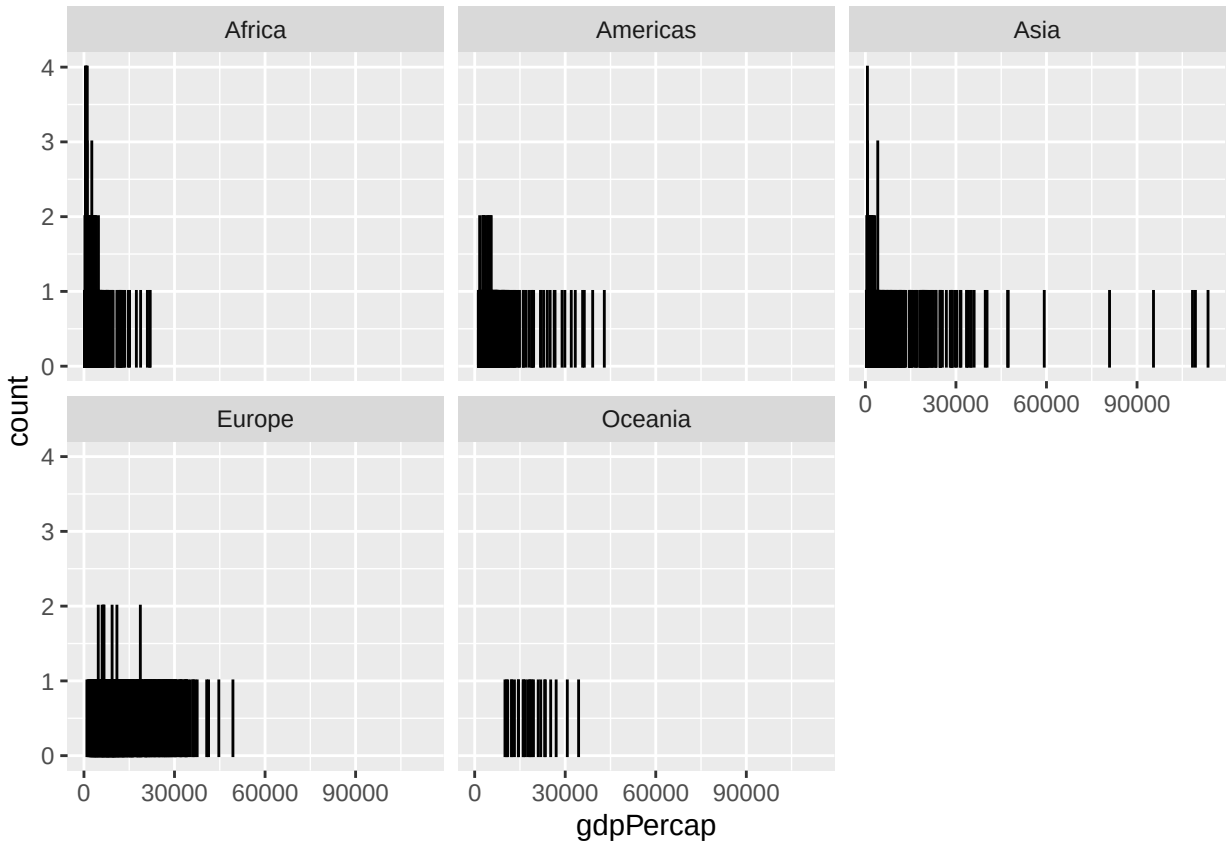
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



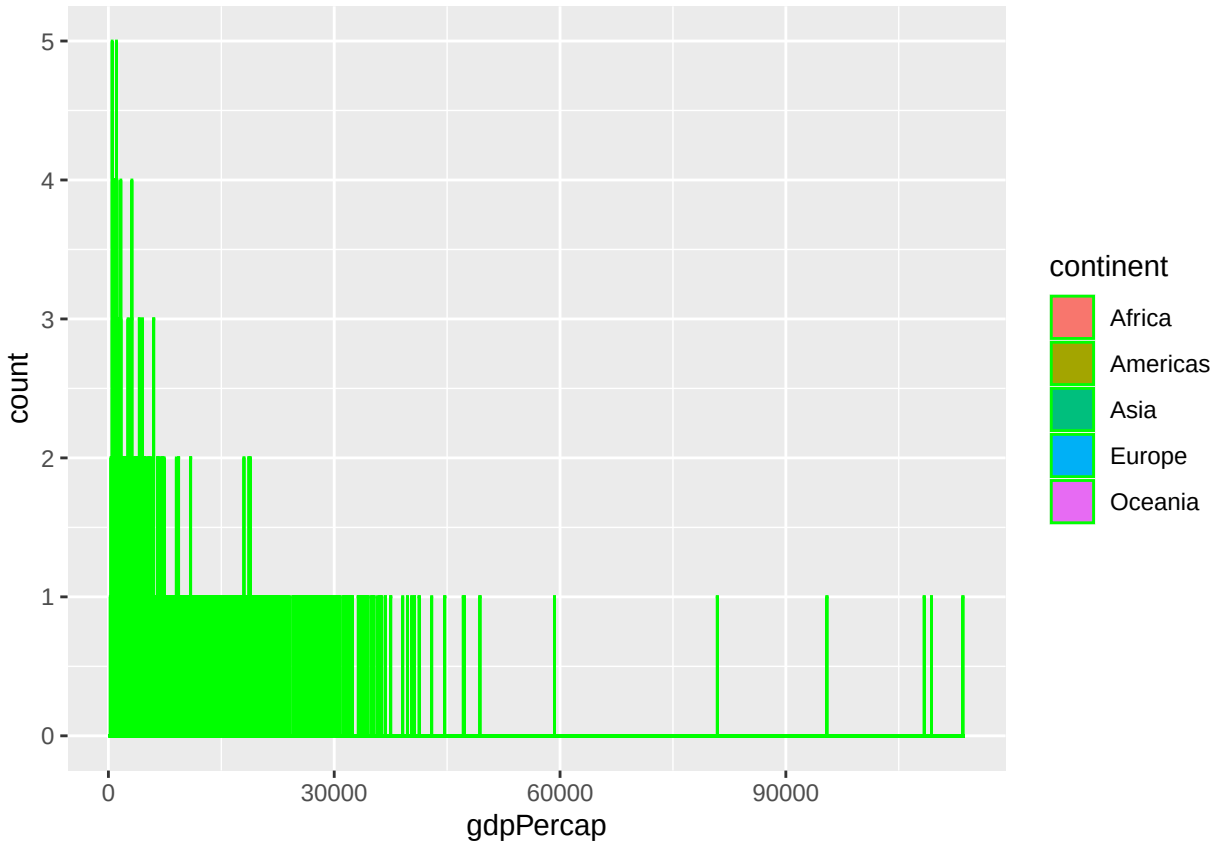

```
  xlim(25000,75000)
```

```
## <ScaleContinuousPosition>
##  Range:
##  Limits: 2.5e+04 -- 7.5e+04
```

```
ggplot(gapminder, aes(gdpPercap)) +
  geom_histogram(binwidth = 2,fill='blue', colour='black') +
  facet_wrap(. ~ continent)
```




```
# change color of the chart
ggplot(gapminder, aes(gdpPercap, fill = continent)) +
  geom_histogram(binwidth = 2,colour = 'green') +
  scale_colour_manual(values=c('red','orange','green','blue','purple'))
```



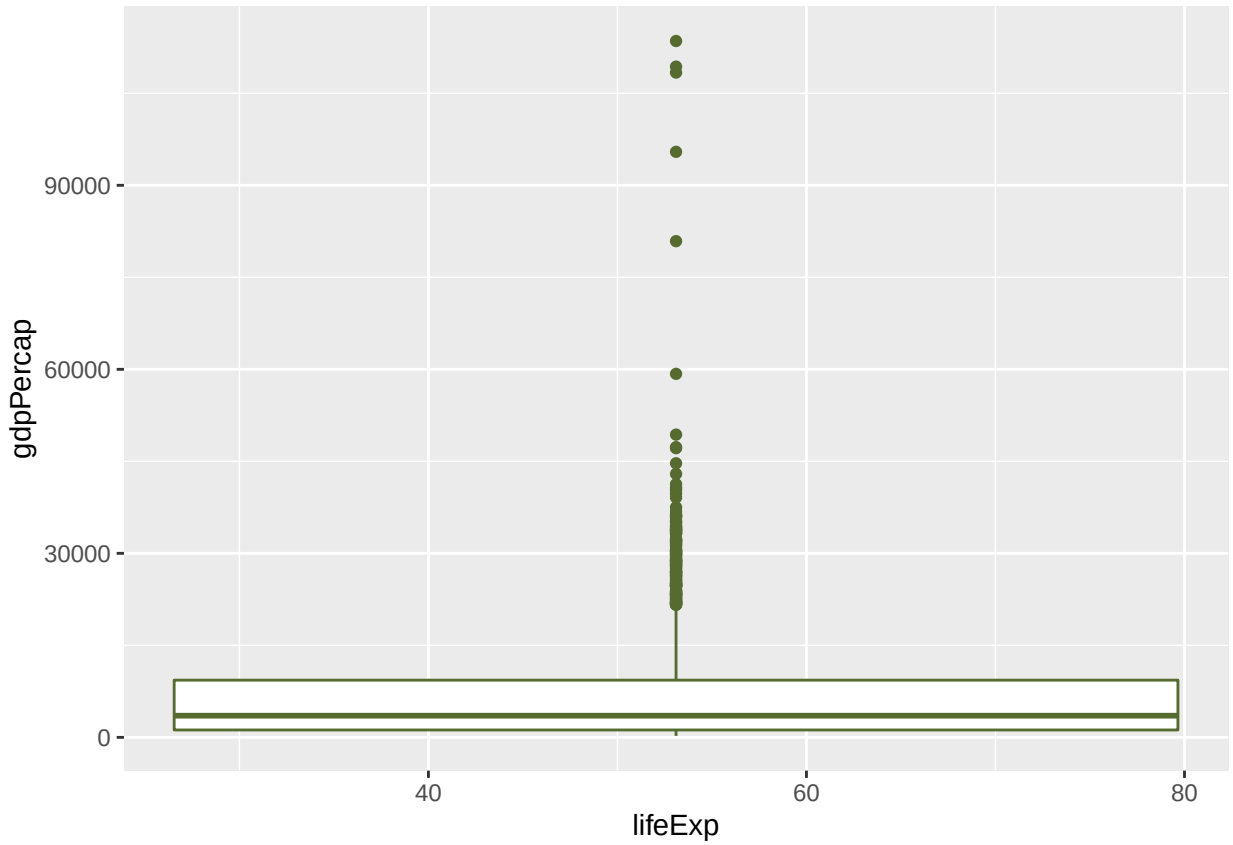

```
  facet_wrap(. ~ continent)
```

```
## <ggproto object: Class FacetWrap, Facet, gg>
##     compute_layout: function
##     draw_back: function
##     draw_front: function
##     draw_labels: function
##     draw_panels: function
##     finish_data: function
##     init_scales: function
##     map_data: function
##     params: list
##     setup_data: function
##     setup_params: function
##     shrink: TRUE
##     train_scales: function
##     vars: function
##     super:  <ggproto object: Class FacetWrap, Facet, gg>
```

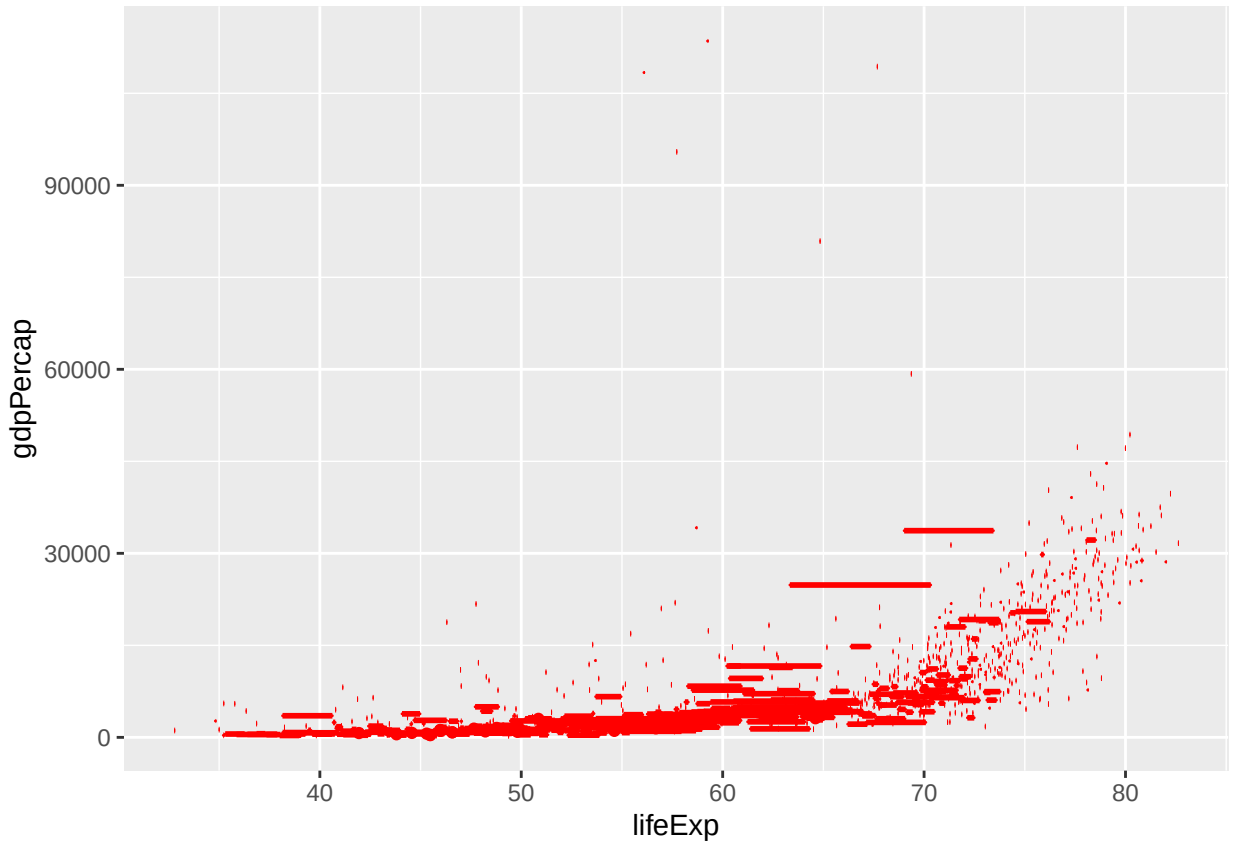
Excercise - 4

```
# Simple boxplot
ggplot(gapminder, aes(lifeExp,gdpPercap)) +
  geom_boxplot(colour=main_color)
```

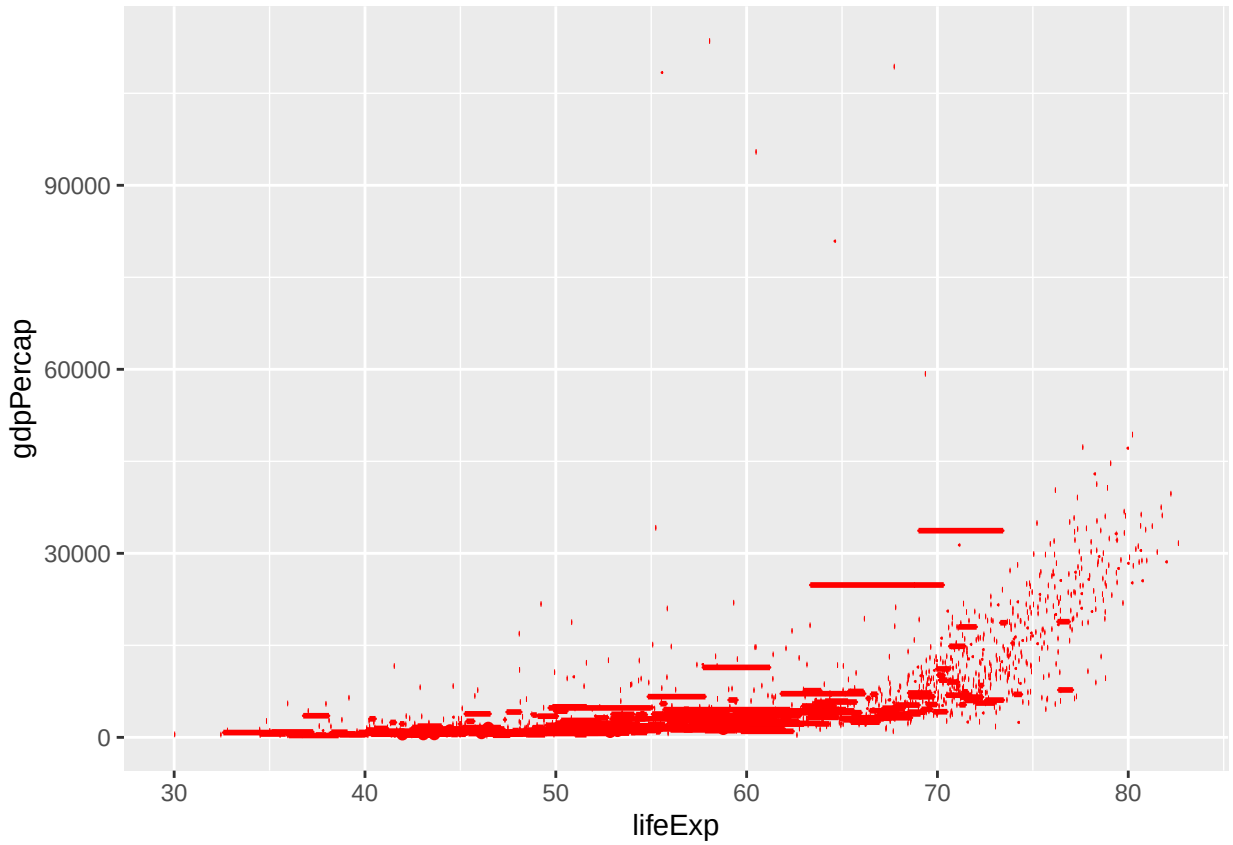
```
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



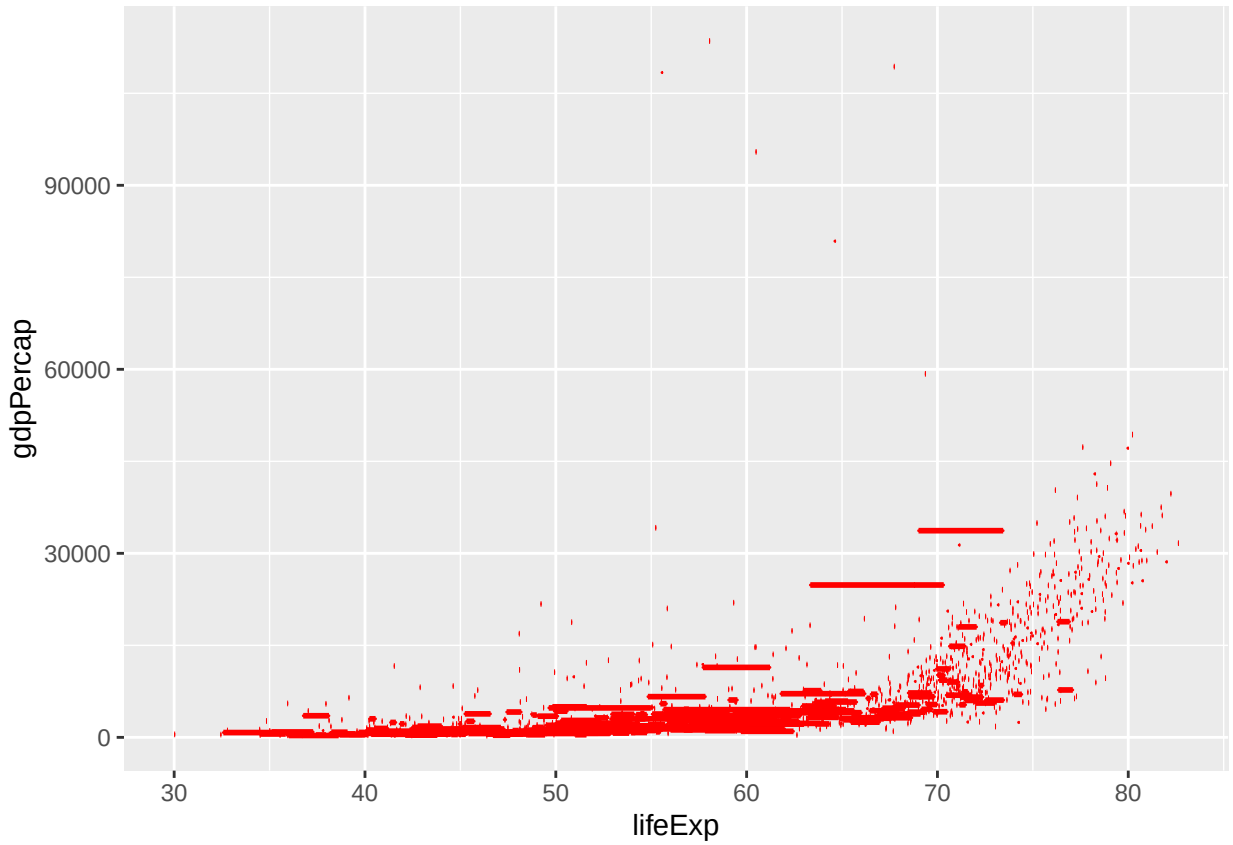

```
# Boxplot using to numeric variable
ggplot(gapminder, aes(lifeExp,gdpPercap)) +
  geom_boxplot(aes(group = cut_width(gdpPercap, 10)), color='red')
```




```
# Smaller groups
ggplot(gapminder, aes(lifeExp,gdpPercap)) +
  geom_boxplot(aes(group = cut_width(gdpPercap, 5)), color='red')
```

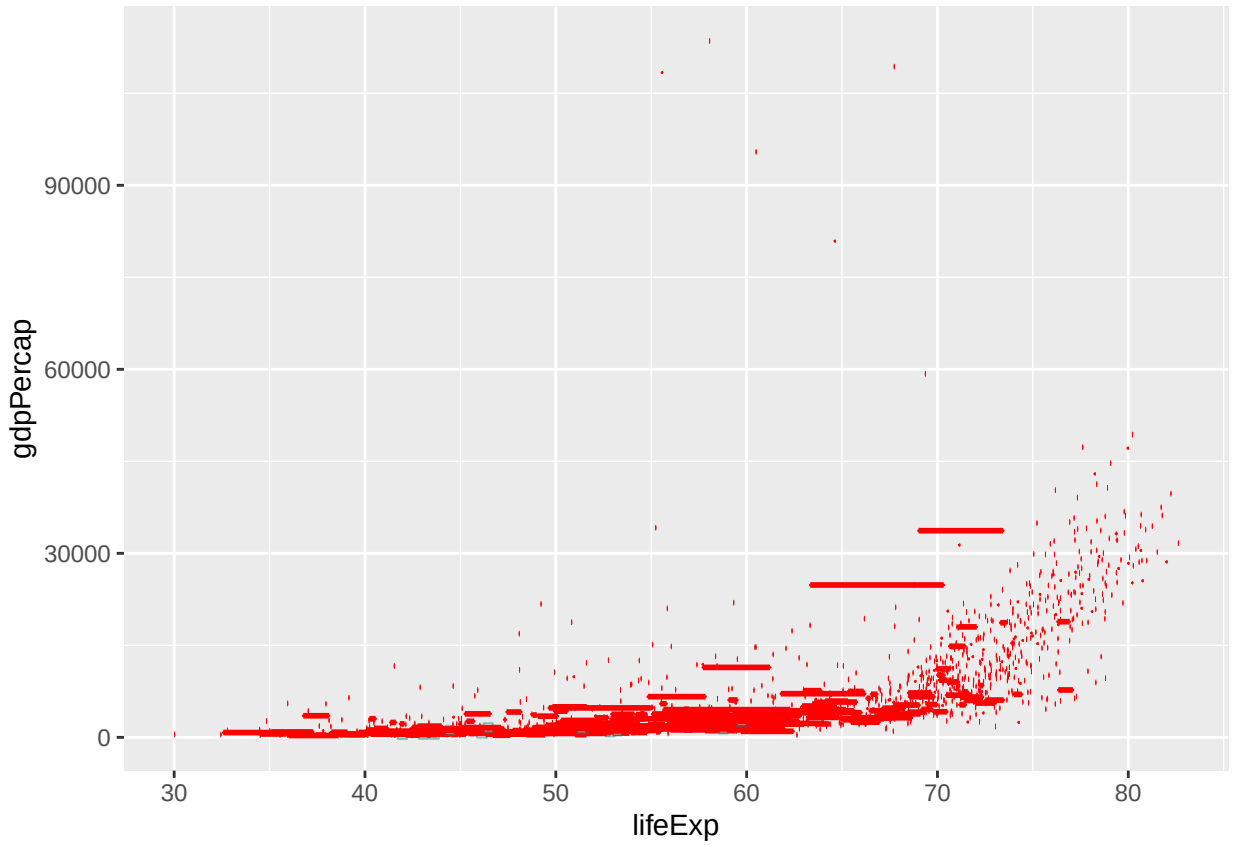



```
# Without outliers
ggplot(gapminder, aes(lifeExp,gdpPercap)) +
  geom_boxplot(aes(group = cut_width(gdpPercap, 5)), color='red',outlier.alpha=0)
```



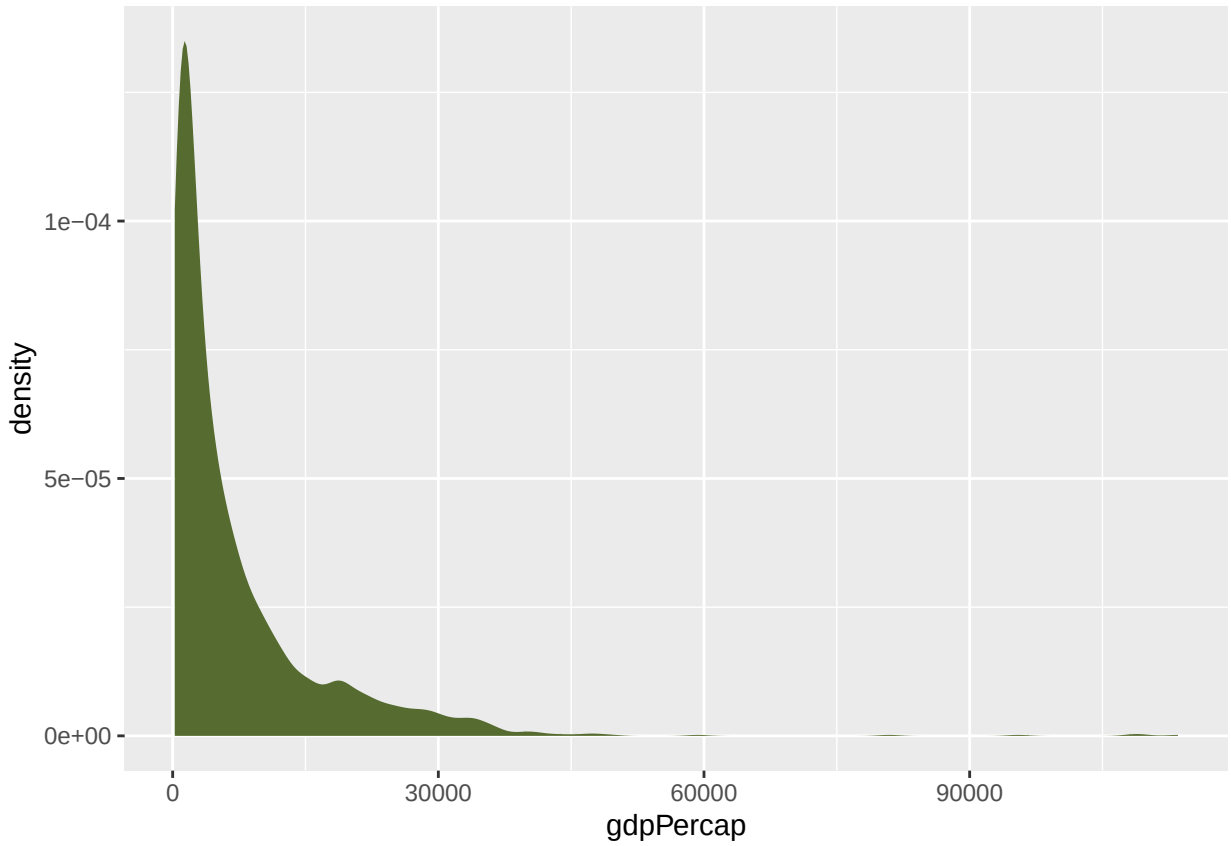

```
# encoding for the outliers
ggplot(gapminder, aes(lifeExp,gdpPercap)) +
  geom_boxplot(aes(group = cut_width(gdpPercap, 5)), color='red',
               color='blue',
               outlier.alpha = 0.5,
               outlier.shape = 22,
               outlier.color=decoration_color)
```

```
## Warning: Duplicated aesthetics after name standardisation: colour
```

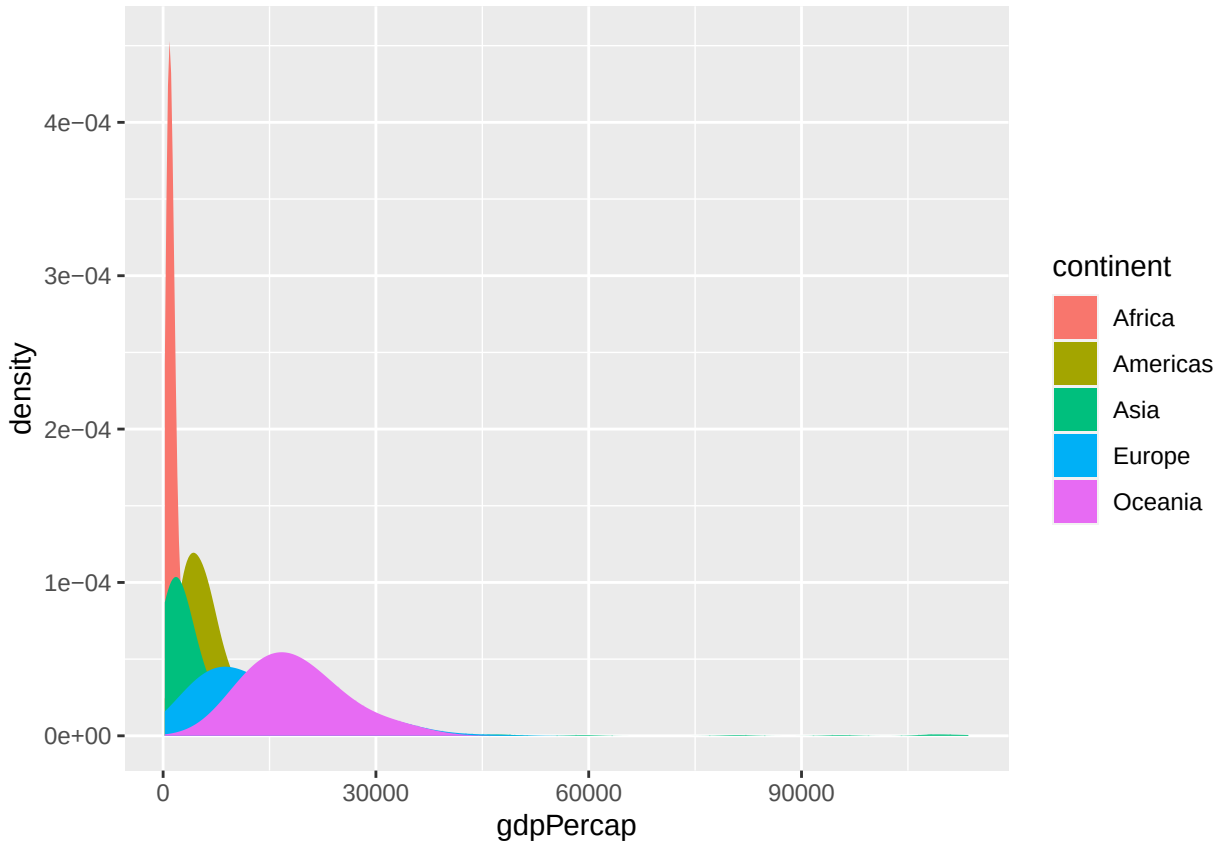


Excercise - 5

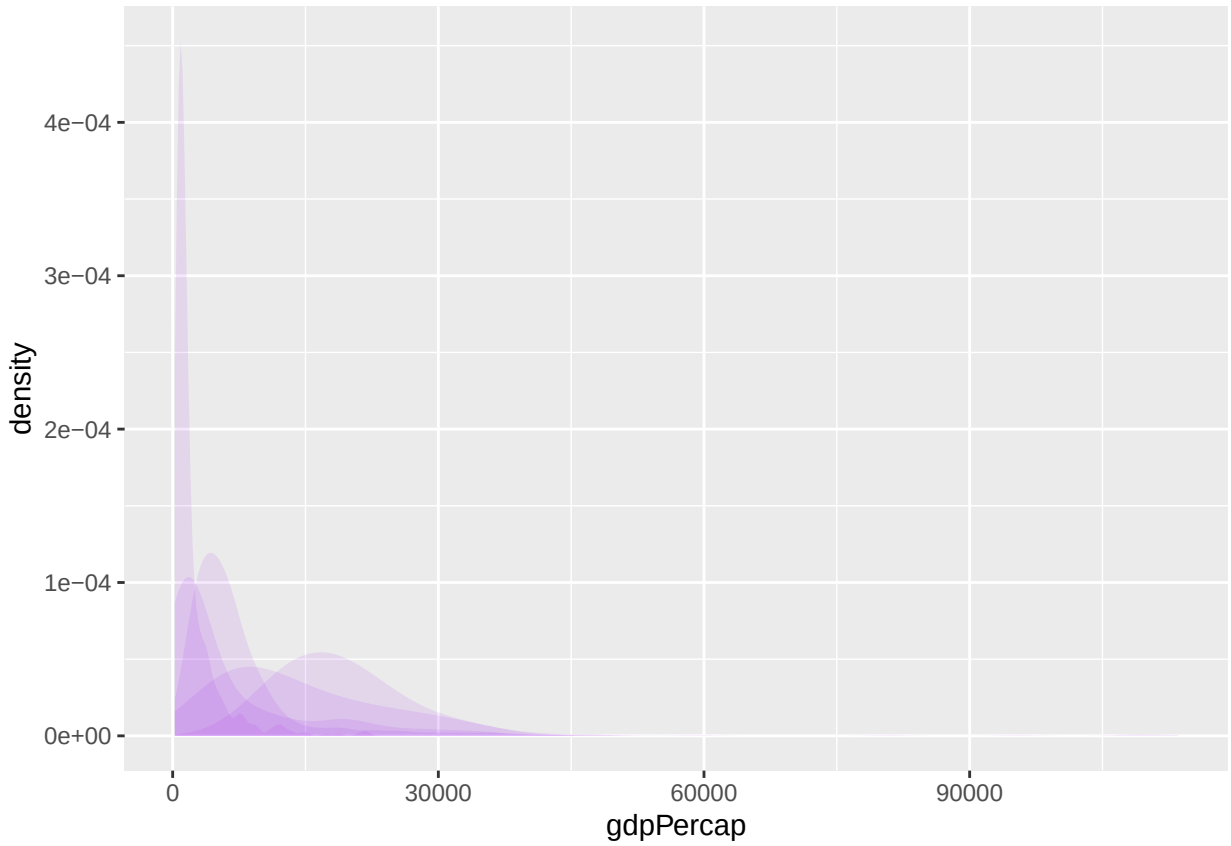
```
#Simple density chart
ggplot(gapminder, aes(gdpPercap)) +
  geom_density(fill = main_color, color = NA)
```



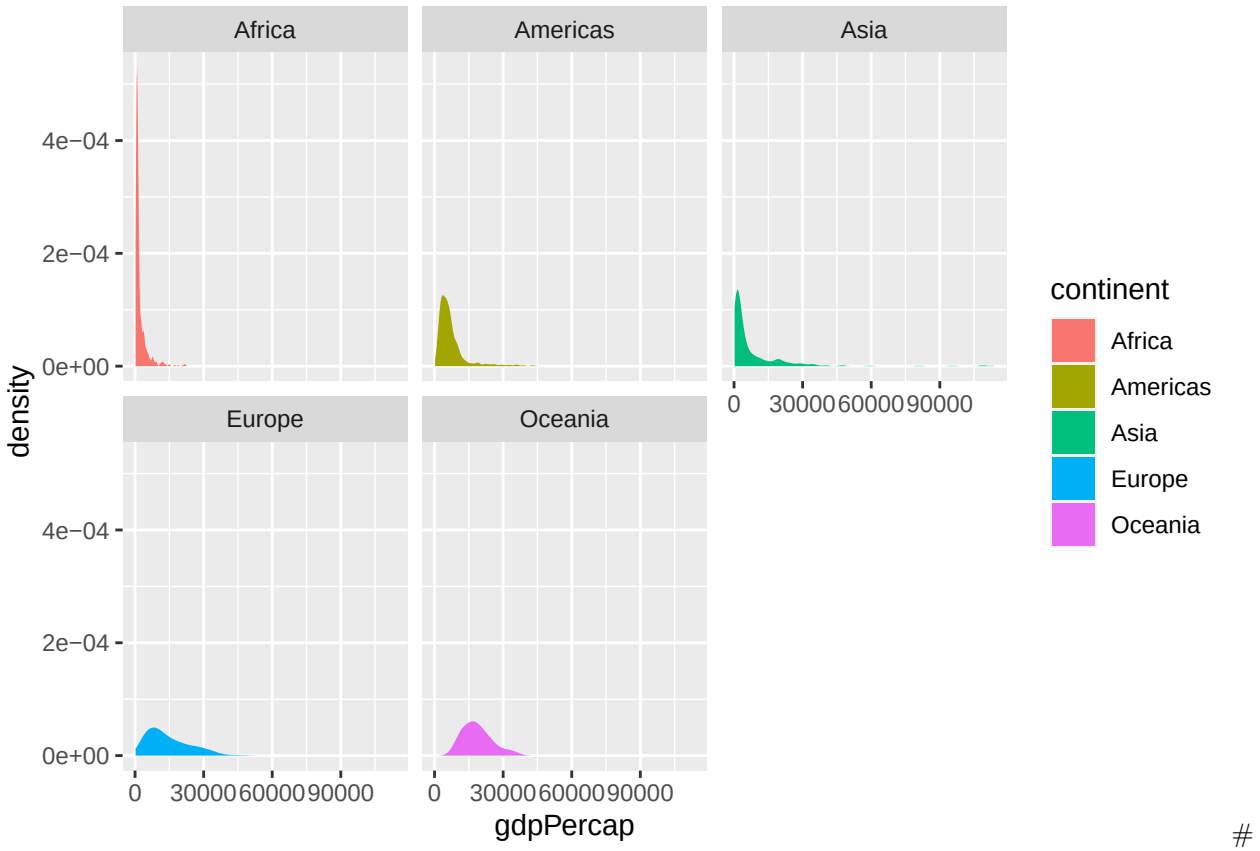

```
#Multiple density chart
ggplot(gapminder, aes(gdpPercap, group = continent, fill = continent)) +
  geom_density(adjust = 1.5 , color = NA) +
  scale_colour_manual(values=c('red','orange','green','blue','purple'))
```




```
#Multiple density chart, with using one color and transparency
ggplot(gapminder, aes(gdpPercap, group = continent, fill = continent)) +
  geom_density(adjust = 1.5 , color = NA, fill='purple', alpha =0.1)
```



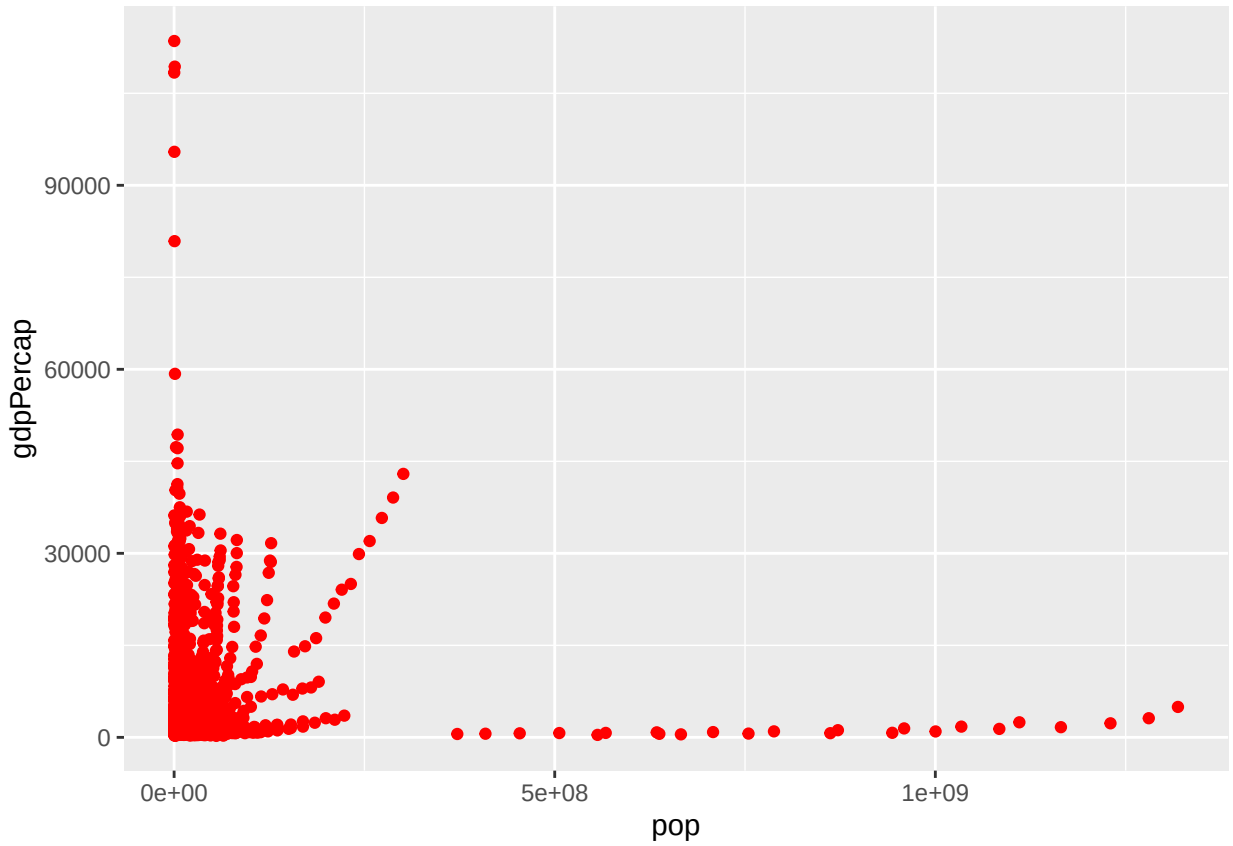

```
#Small multiple density for carat by the different cuts
ggplot(gapminder, aes(gdpPercap, group = continent, fill = continent)) +
  geom_density(color = NA) +
  scale_colour_manual(values=c('red','orange','green','blue','purple')) +
  facet_wrap(. ~ continent)
```



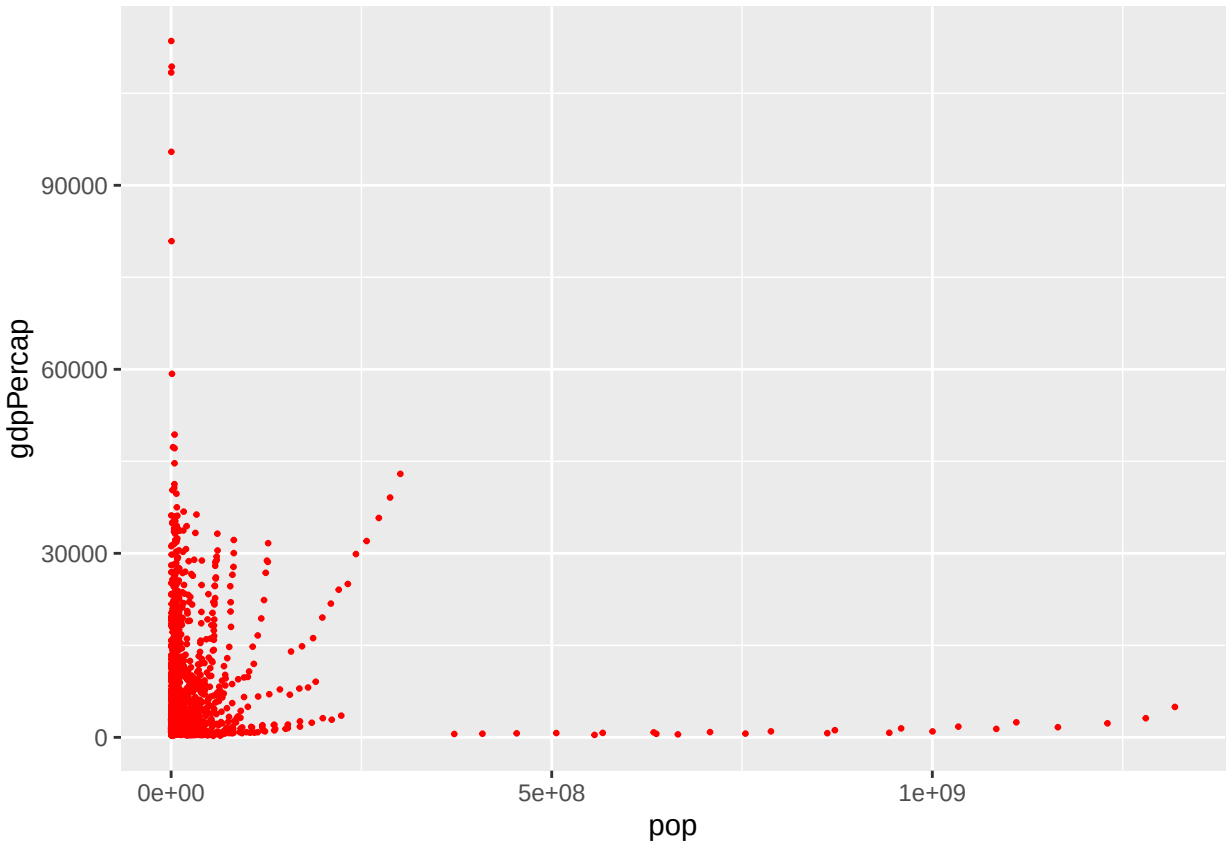
#

Excercise - 6

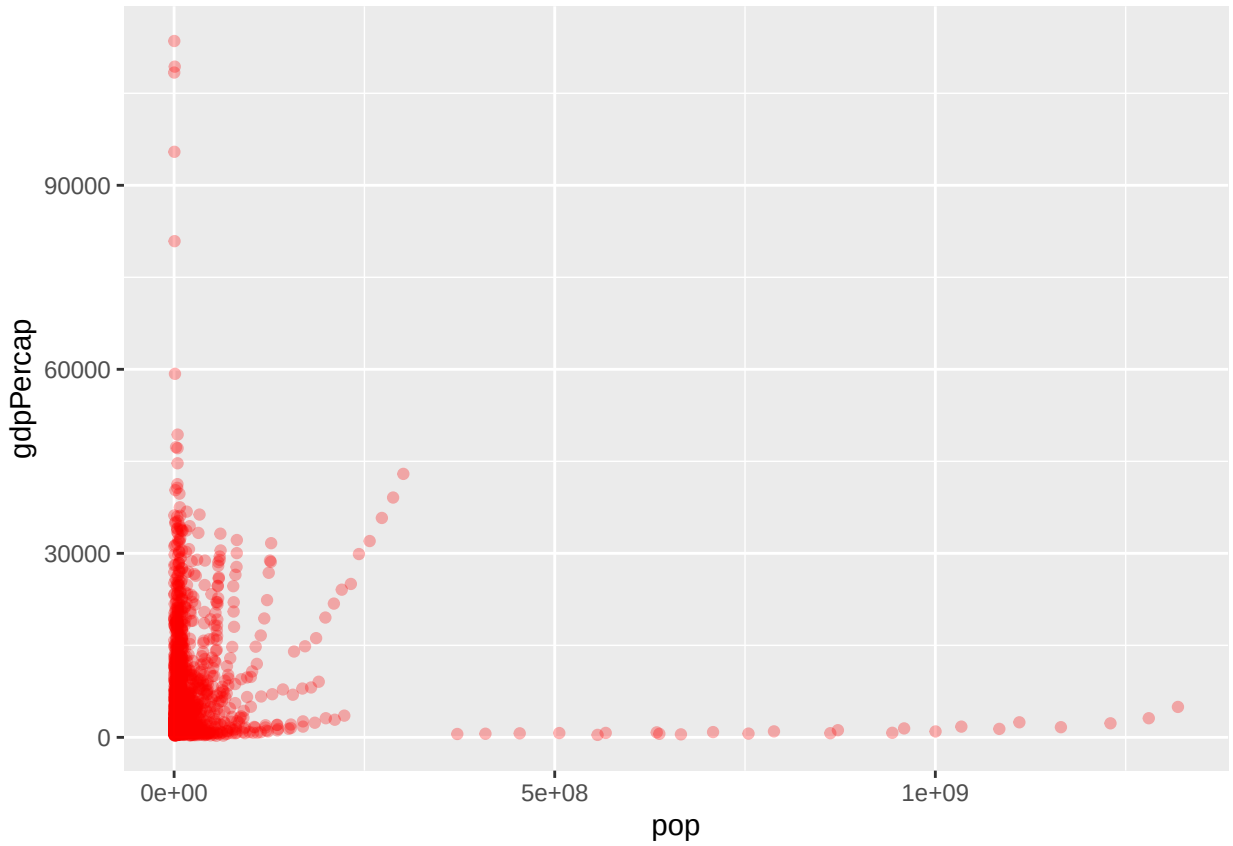
```
#Basic scatter plot
ggplot(gapminder, aes(x=pop, y=gdpPercap)) +
  geom_point(color='red')
```



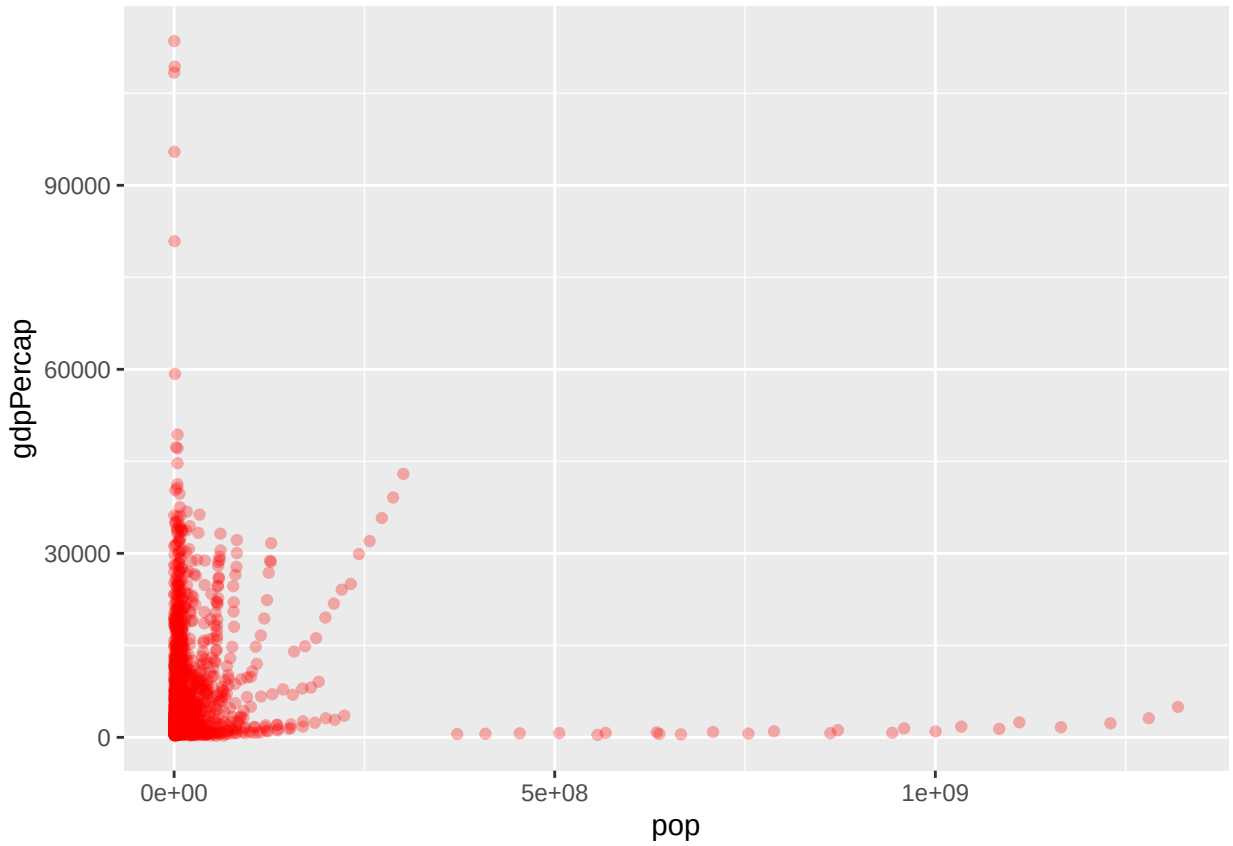

```
#Basic scatter plot - adjusting the size
ggplot(gapminder, aes(x=pop, y=gdpPercap)) +
  geom_point(size=0.5, color='red')
```




```
#Basic scatter plot - adjusting the opacity
ggplot(gapminder, aes(x=pop, y=gdpPercap)) +
  geom_point(alpha=0.3, color='red')
```



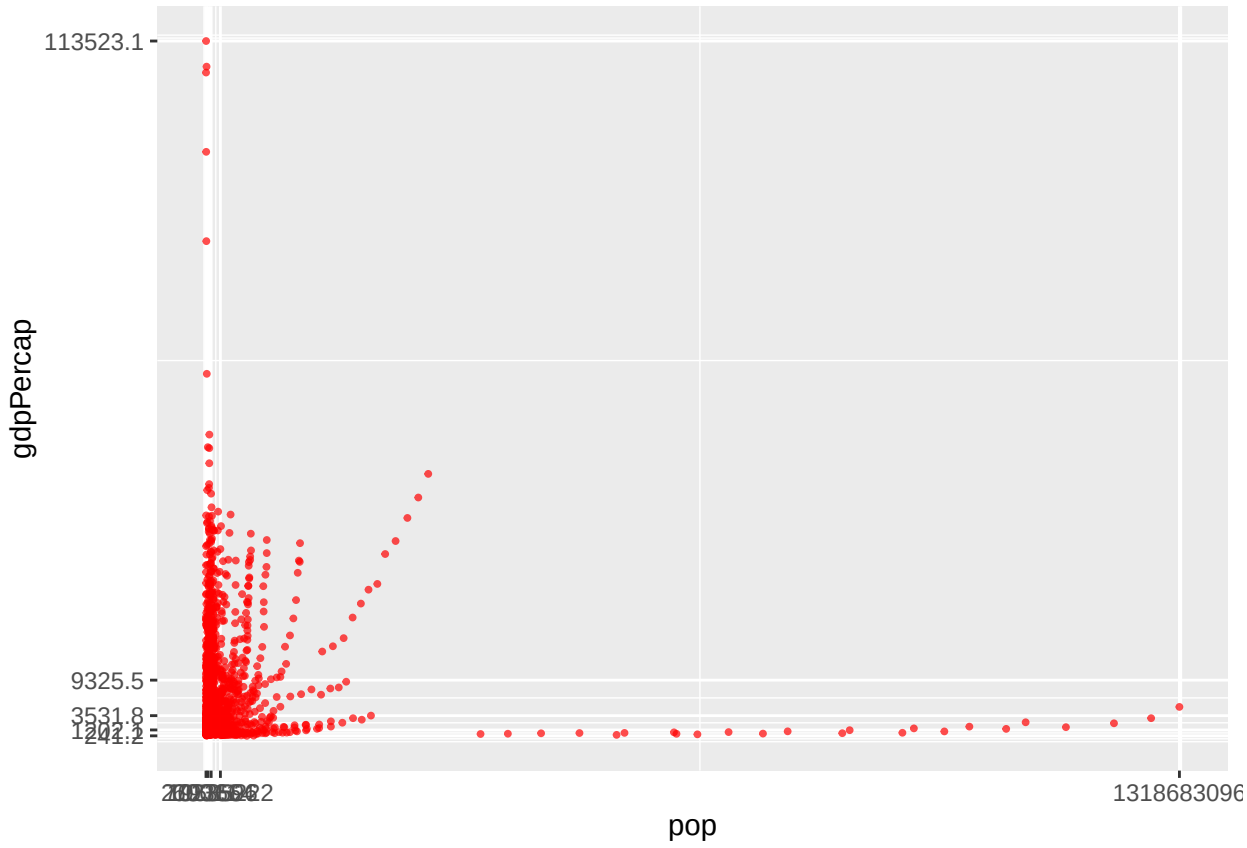

```
#Basic scatter plot changing the Y limits
ggplot(gapminder, aes(x=pop, y=gdpPercap)) +
  geom_point(alpha=0.3, color='red')
```



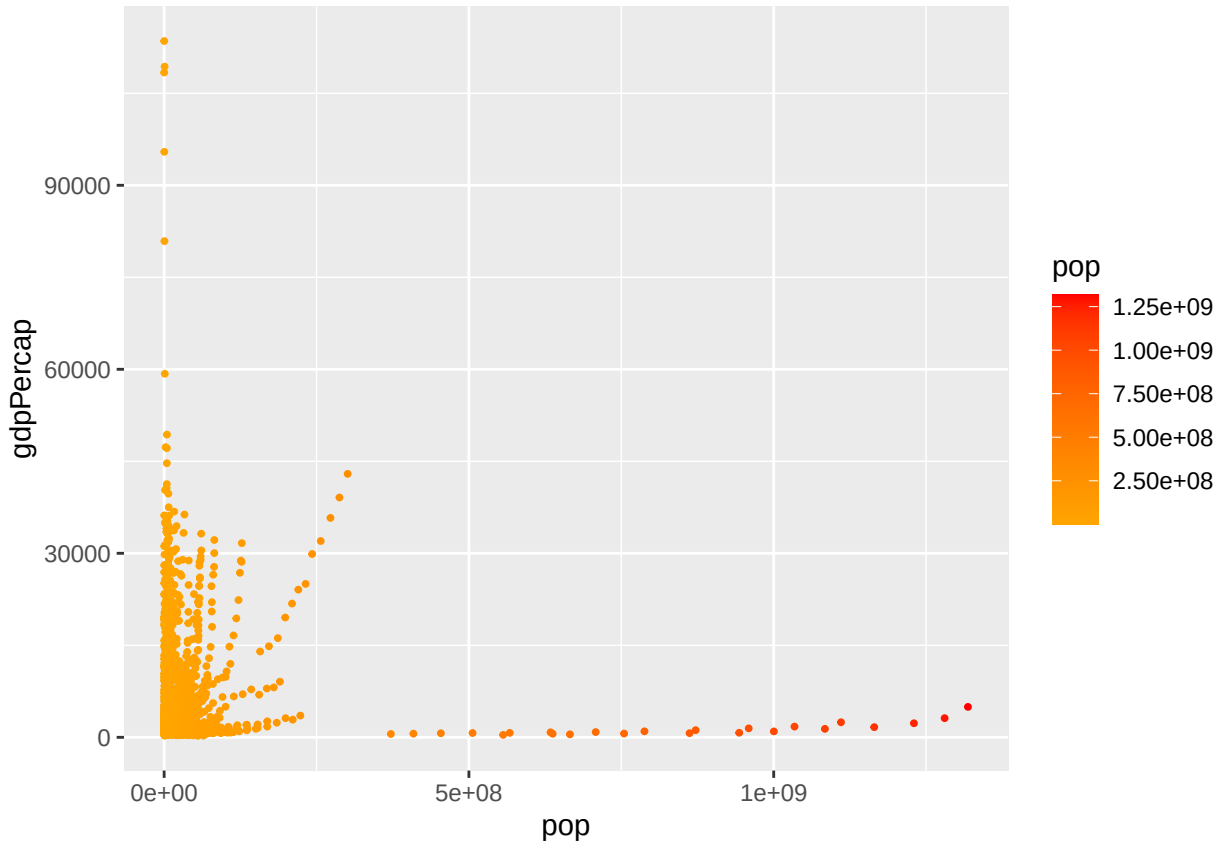

```
  ylim(836., 978.)
```

```
## <ScaleContinuousPosition>
##  Range:
##  Limits:  836 --  978
```

```
#Axis labeling depending on the quantiles
ggplot(gapminder, aes(x=pop, y=gdpPercap)) +
  geom_point(size=0.7, alpha=0.7, color='red')  +
  scale_x_continuous(breaks = round(as.vector(quantile(gapminder$pop)), digits = 1))+
  scale_y_continuous(breaks = round(as.vector(quantile(gapminder$gdpPercap)), digits = 1))
```

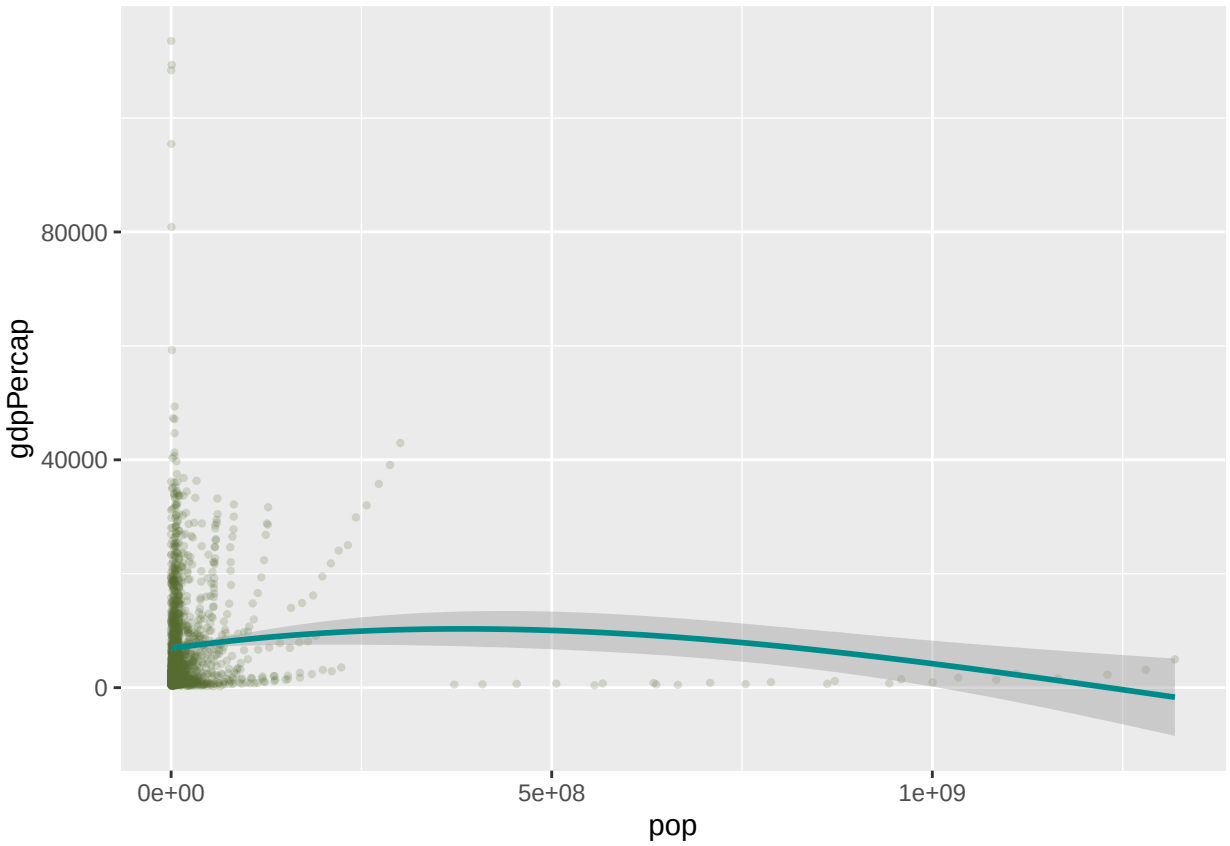



```
#Adding price as another visual encoding using a colour code
ggplot(gapminder, aes(x=pop, y=gdpPercap, colour = pop)) +
  geom_point(size=0.7, alpha=1)+
  scale_colour_gradient(low = sub_color, high = 'red')
```



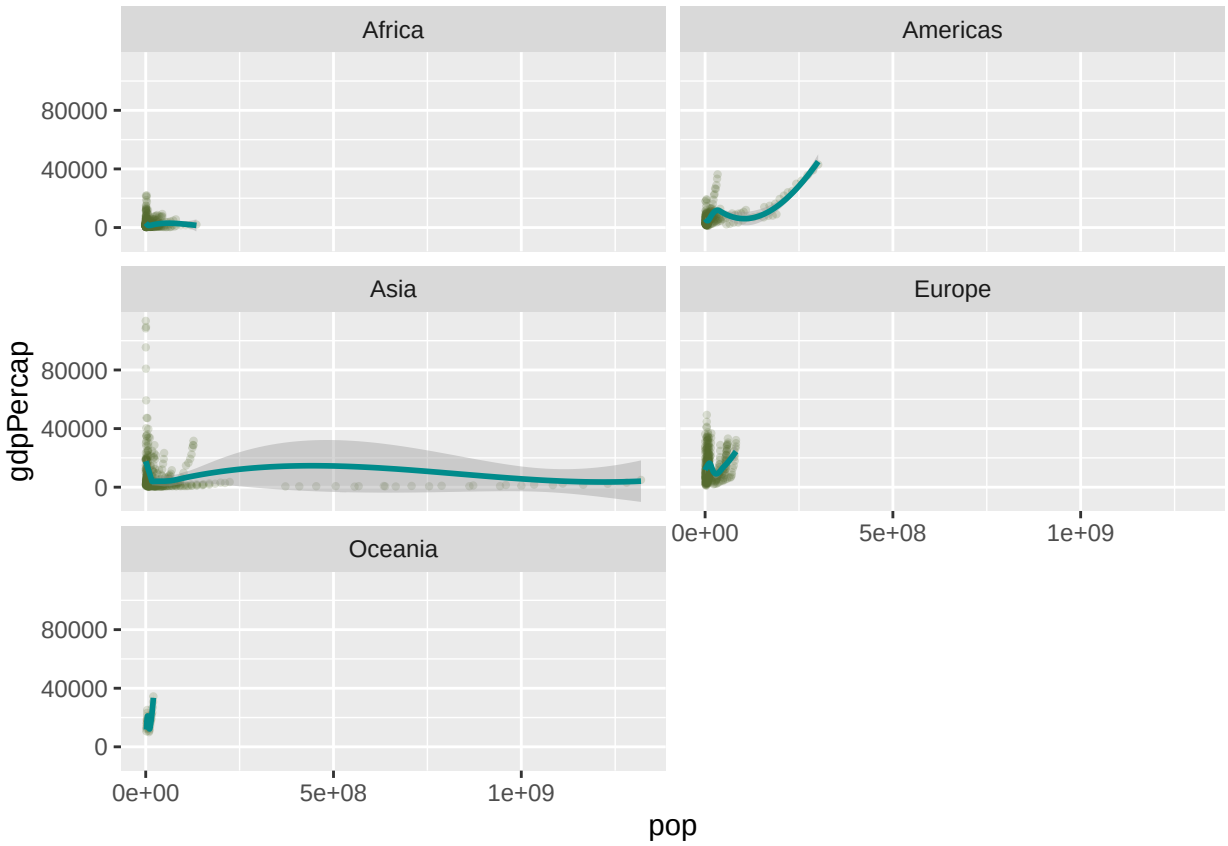

```
#Adding a trend line
ggplot(gapminder, aes(x=pop, y=gdpPercap)) +
  geom_point(color=main_color, size=0.8, alpha=0.2)+
  stat_smooth(color=decoration_color)
```

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



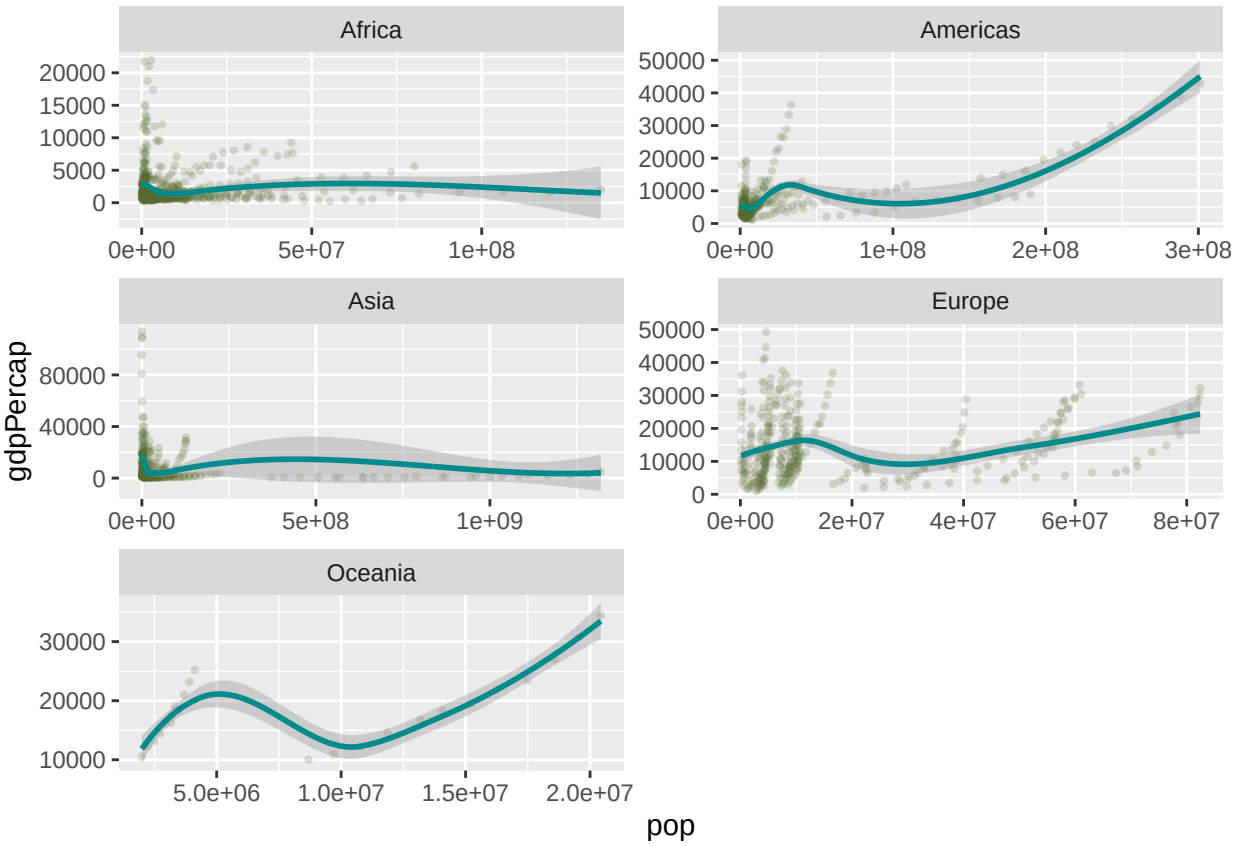

```
#Small multiples - one variable
ggplot(gapminder, aes(x=pop, y=gdpPercap)) +
  geom_point(color=main_color, size=0.8, alpha=0.2) +
  facet_wrap( ~ continent, ncol=2) +
  stat_smooth(color=decoration_color)
```

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



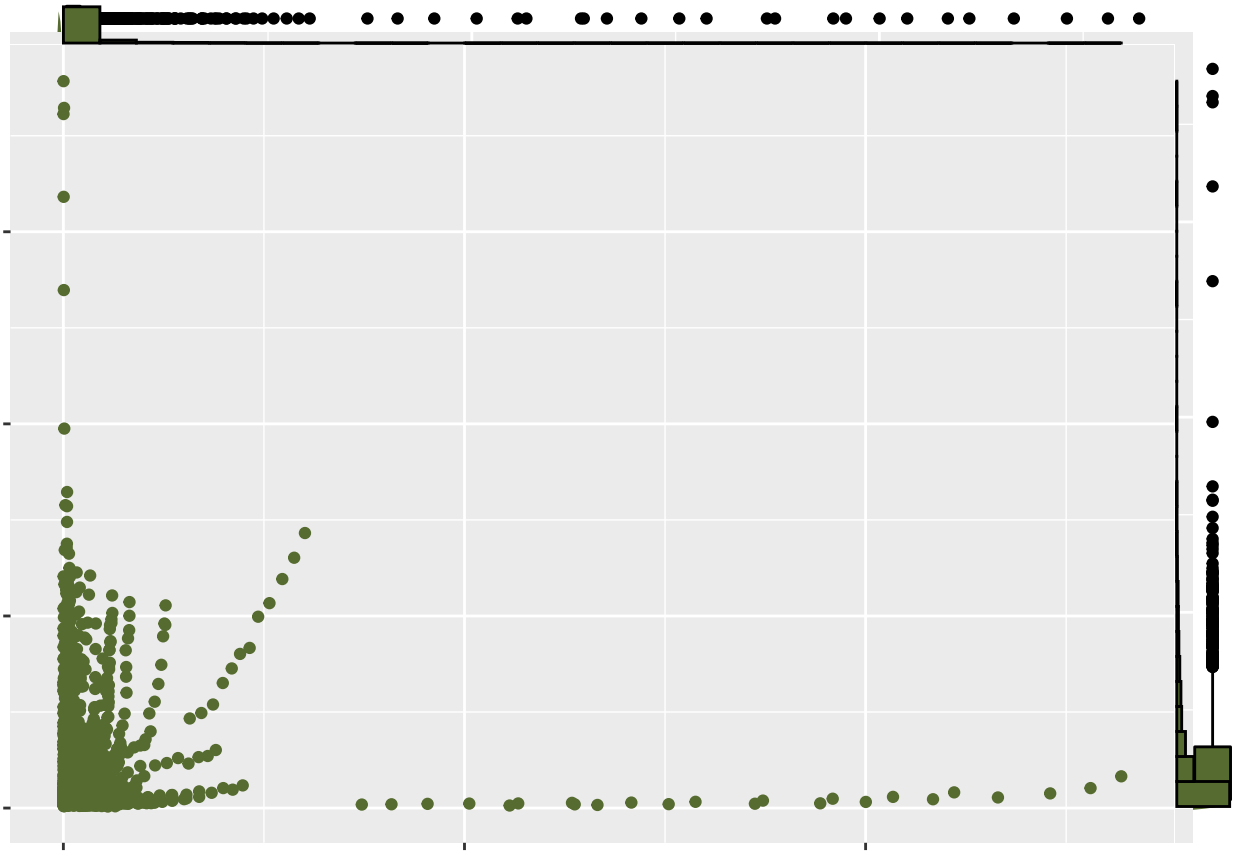

```
#Small multiples - one variable with free scale
ggplot(gapminder, aes(x=pop, y=gdpPercap)) +
  geom_point(color=main_color, size=0.8, alpha=0.2) +
  facet_wrap( ~ continent, ncol=2, scales = "free") +
  stat_smooth(color=decoration_color)
```

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



Excercise - 7

```
#Density
pp <- ggplot(gapminder, aes(x=pop, y=gdpPercap)) +
  geom_point(color=main_color) +
  theme(axis.title=element_blank(), axis.text=element_blank())

ggMarginal(pp, type = "density", fill = main_color, alpha=1, color='transparent')

#Box-plot

ggMarginal(pp, type = "boxplot", size=30, fill=main_color)

#Histogram

ggMarginal(pp, type = "histogram", size=20, fill=main_color)
```



Excercise - 8

#Simple beewswarm ggplot(gapminder, aes(x=gdpPercap, y=continent)) + geom_quasirandom(size=3, alpha=0.7, color=main_color, groupOnX=FALSE)

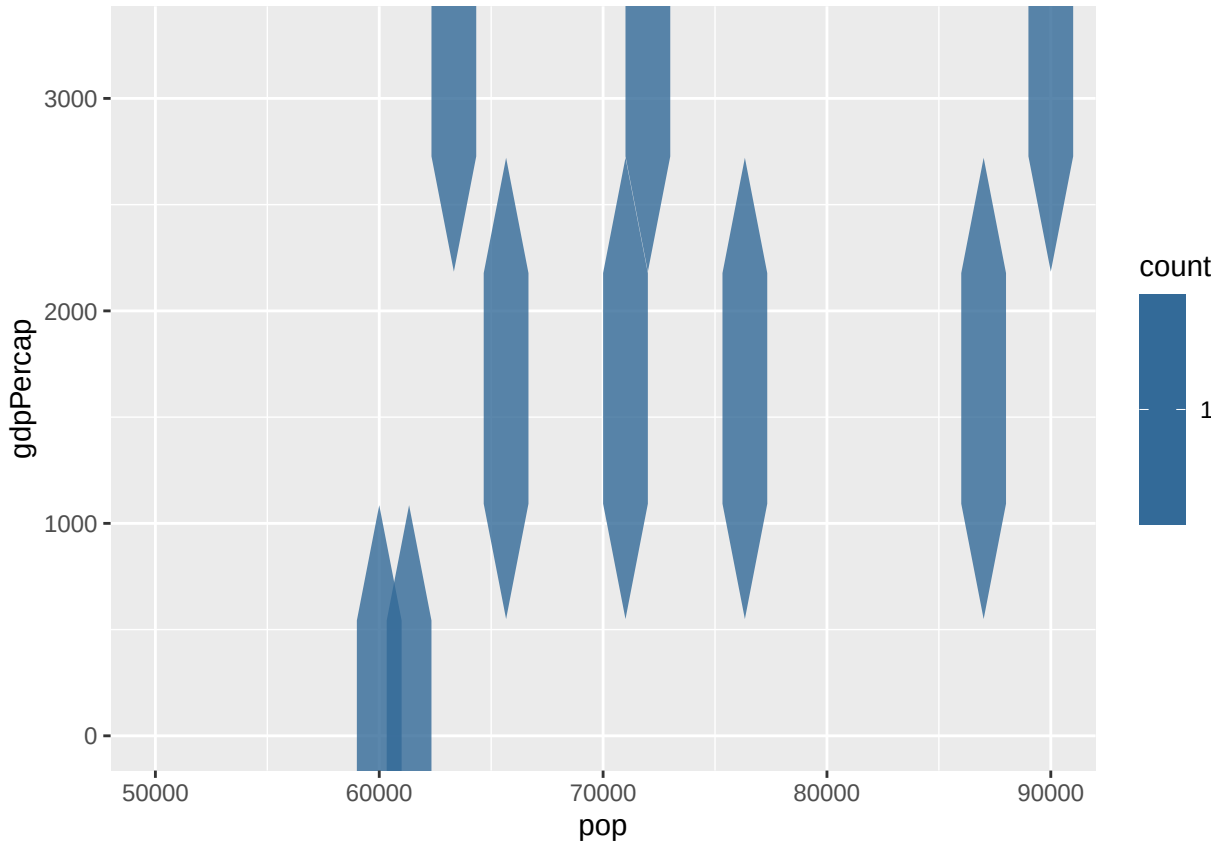
#Adding color ggplot(gapminder, aes(x=gdpPercap, y=continent, colour=continent)) + geom_quasirandom(size=3, alpha=0.7, groupOnX=FALSE) + scale_color_manual(values=c('red','orange','green','blue','purple'))

#Adding color ggplot(gapminder, aes(x=gdpPercap, y=continent, colour=continent)) + geom_quasirandom(alpha=0.7, groupOnX=FALSE, method = "smiley") + scale_color_manual(values=c('red','orange','green','blue','purple'))

Excercise - 9

```
#Aggregation through hexagonal binning - defining the number of bins
ggplot(gapminder, aes(x=pop, y=gdpPercap)) +
  geom_hex(bins=60, alpha =0.8)+
  xlim(50000, 90000)
```

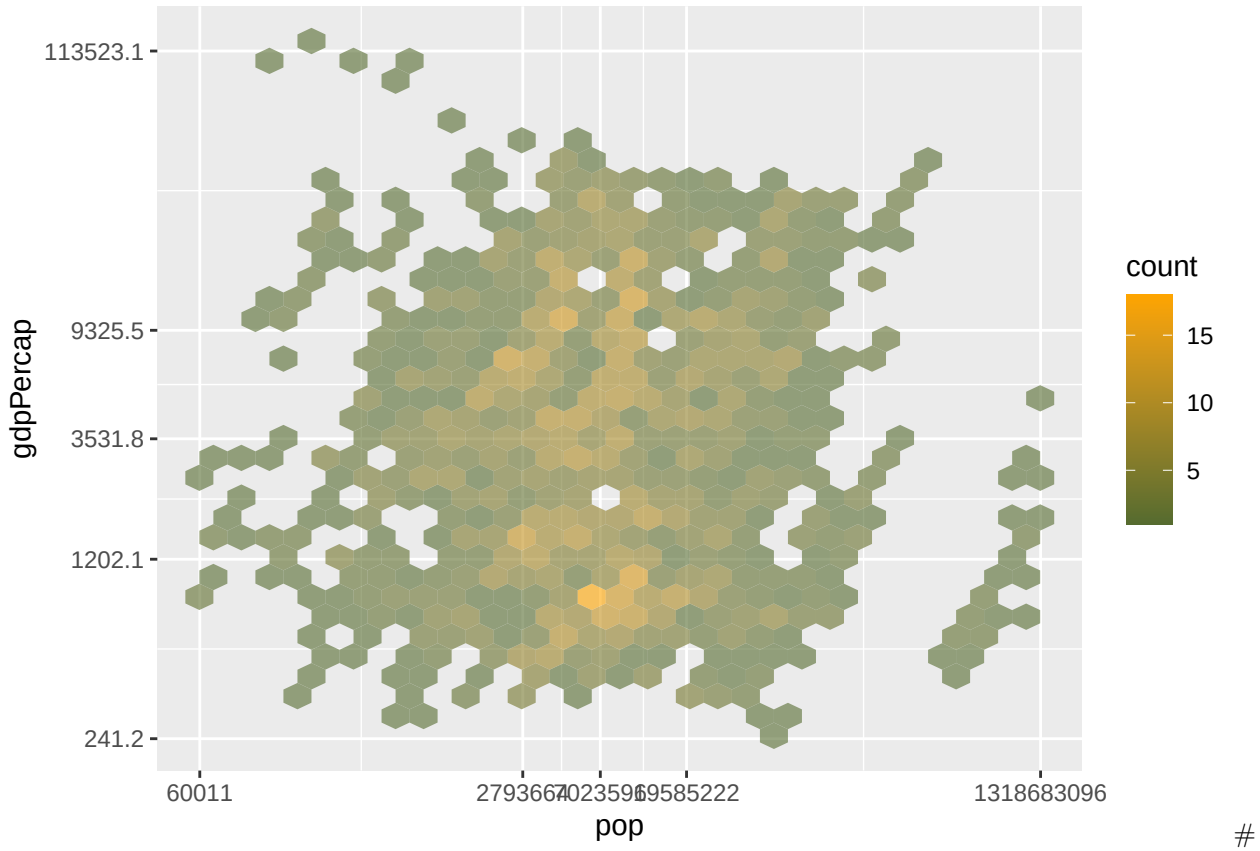
```
## Warning: Removed 1695 rows containing non-finite values (stat_binhex).
```




```
  scale_fill_gradient(low=main_color, high=sub_color)
```

```
## <ScaleContinuous>
##  Range:
##  Limits:    0 --    1
```

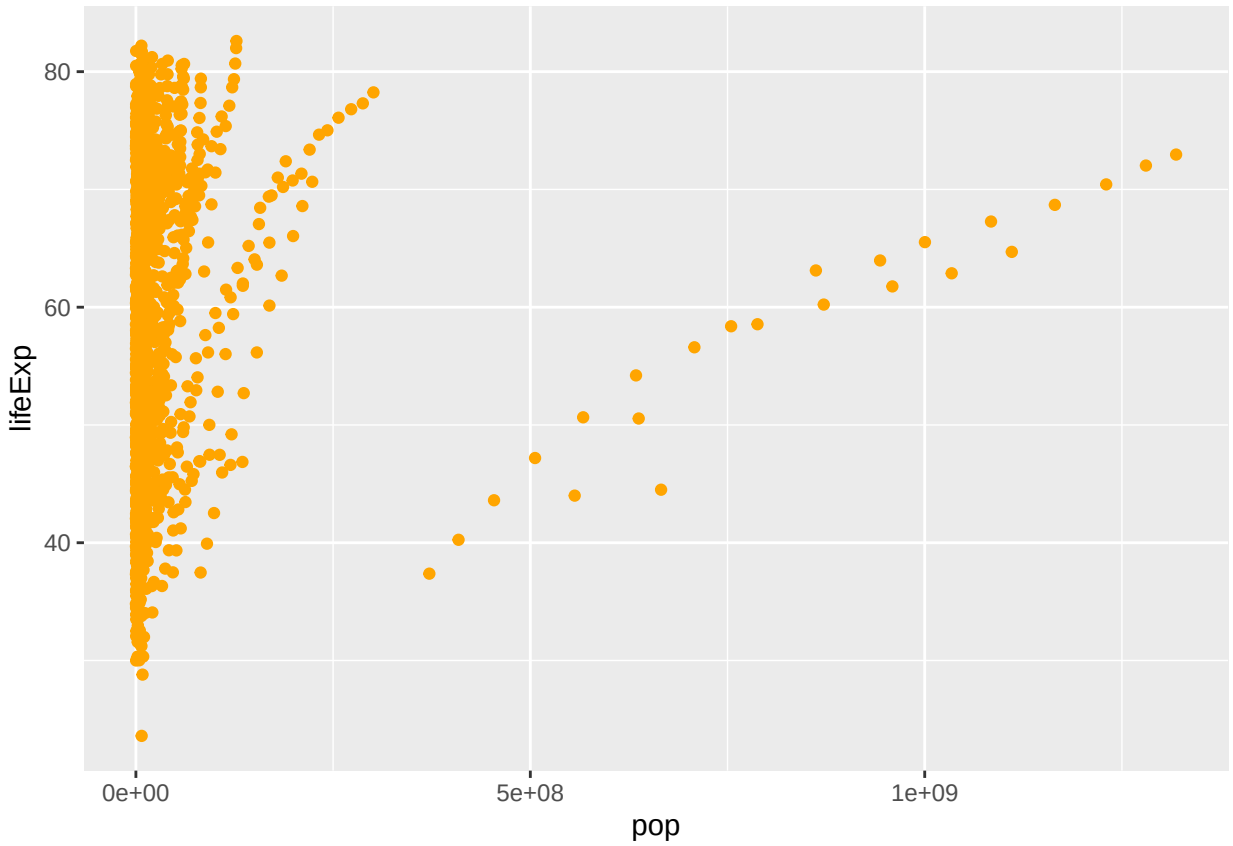
```
#Aggregation through hexagonal binning - logarithmic scaling
ggplot(gapminder, aes(x=pop, y=gdpPercap)) +
  geom_hex(alpha = 0.6) +
  scale_x_log10(breaks = round(as.vector(quantile(gapminder$pop)), digits = 1))+
  scale_y_log10(breaks = round(as.vector(quantile(gapminder$gdpPercap)), digits = 1))+
  scale_fill_gradient(low=main_color, high=sub_color)
```



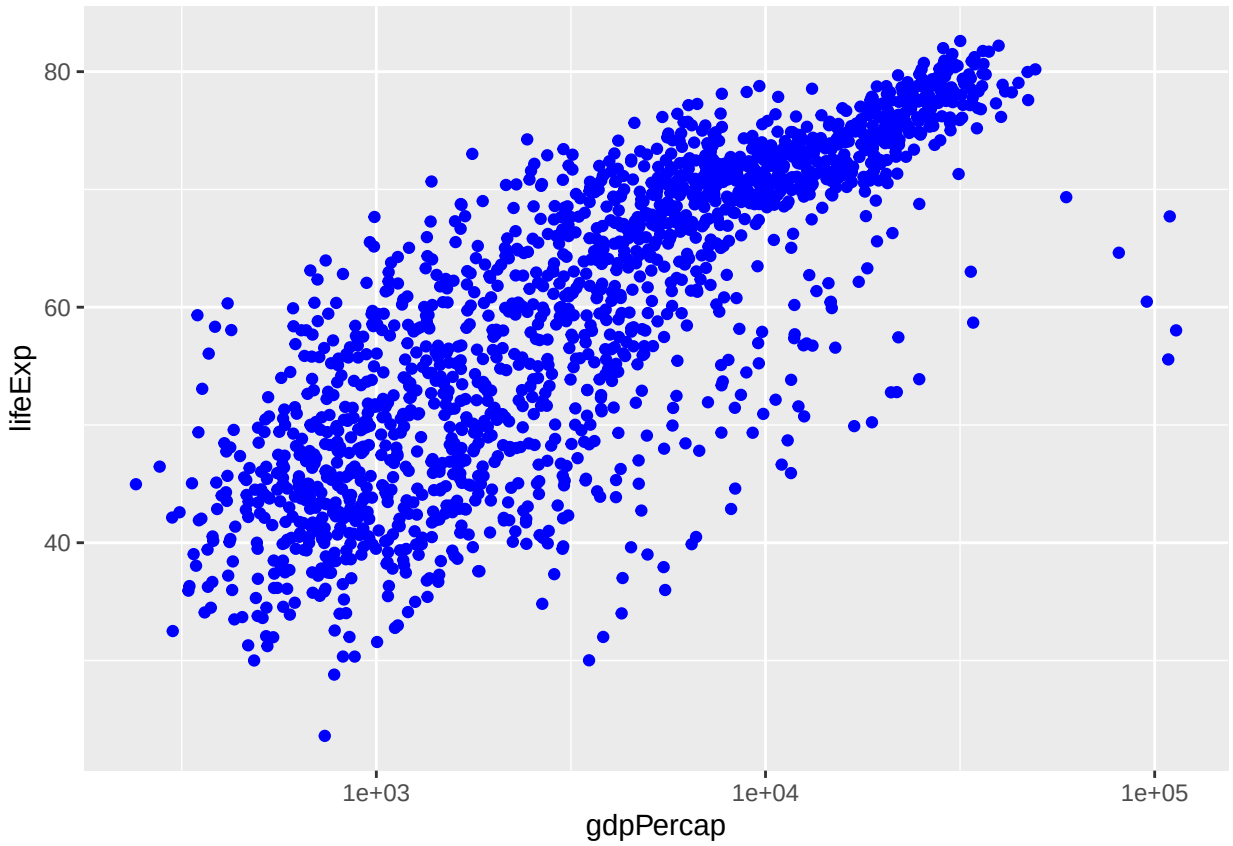
#

Excercise - 10

```
#General scatter plot
ggplot(gapminder, aes(pop, lifeExp)) +
  geom_point(colour = 'orange')
```

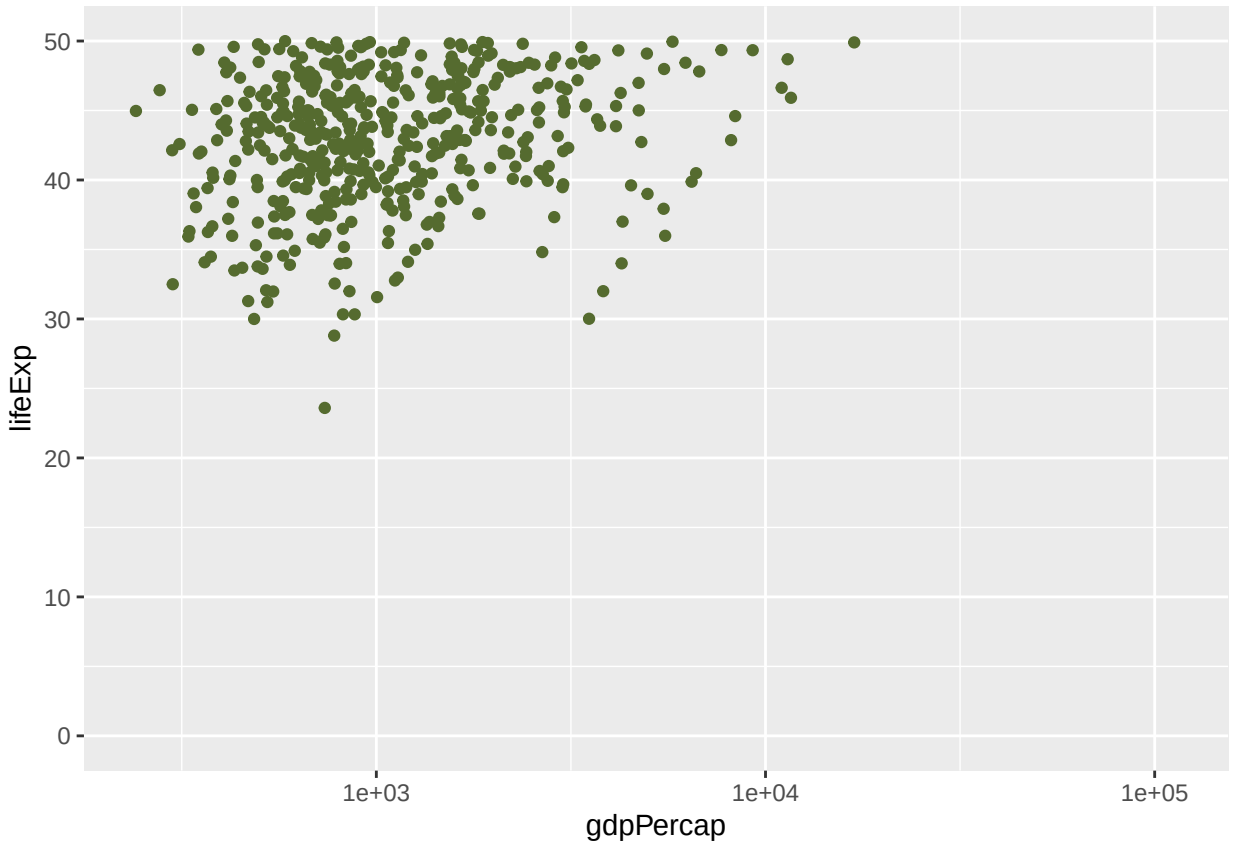



```
#Apply a log scale to the X axis postiion
ggplot(gapminder, aes(gdpPercap, lifeExp)) +
  geom_point(colour = 'blue') +
  scale_x_log10()
```

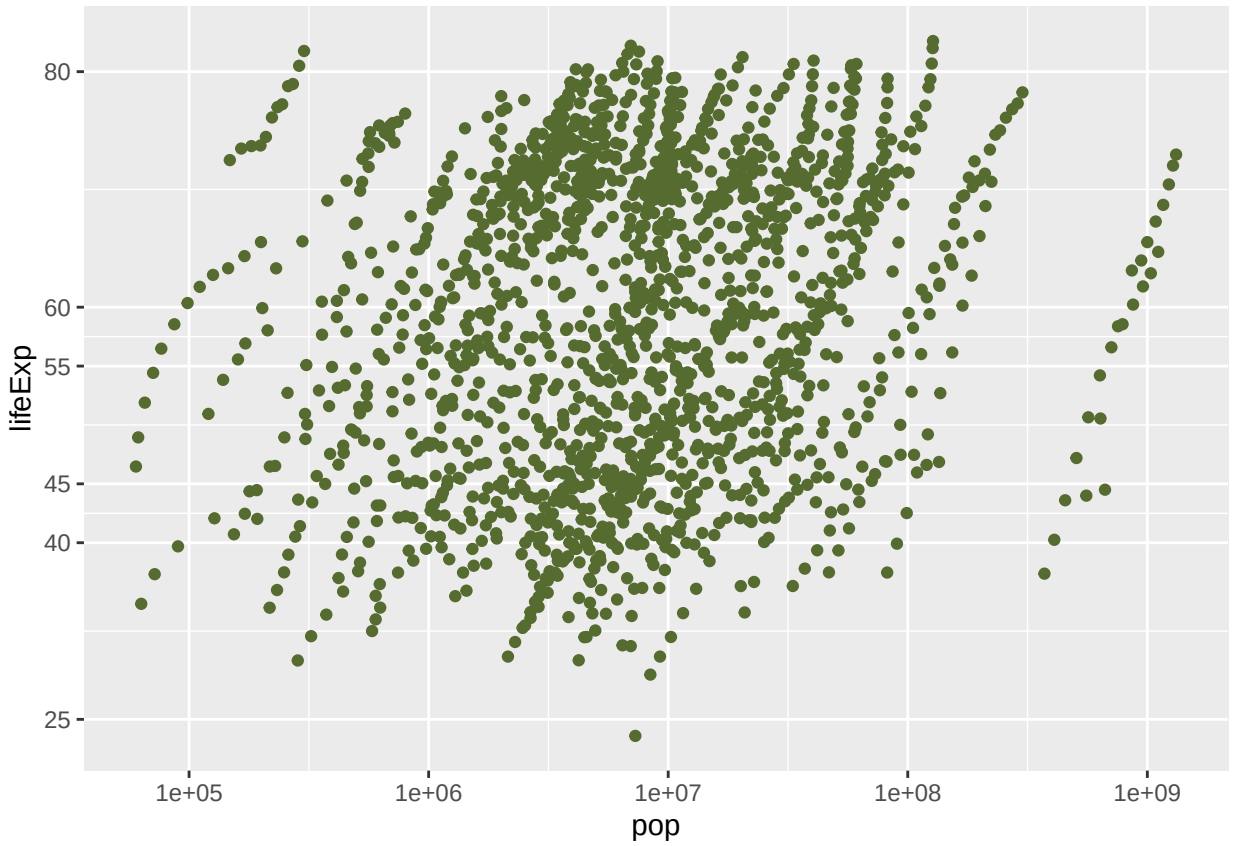



```
#Apply a a linear transformation to the Y axis position with limits
ggplot(gapminder, aes(gdpPercap, lifeExp)) +
  geom_point(colour = main_color) +
  scale_x_log10() +
  scale_y_continuous(limits = c(0, 50))
```

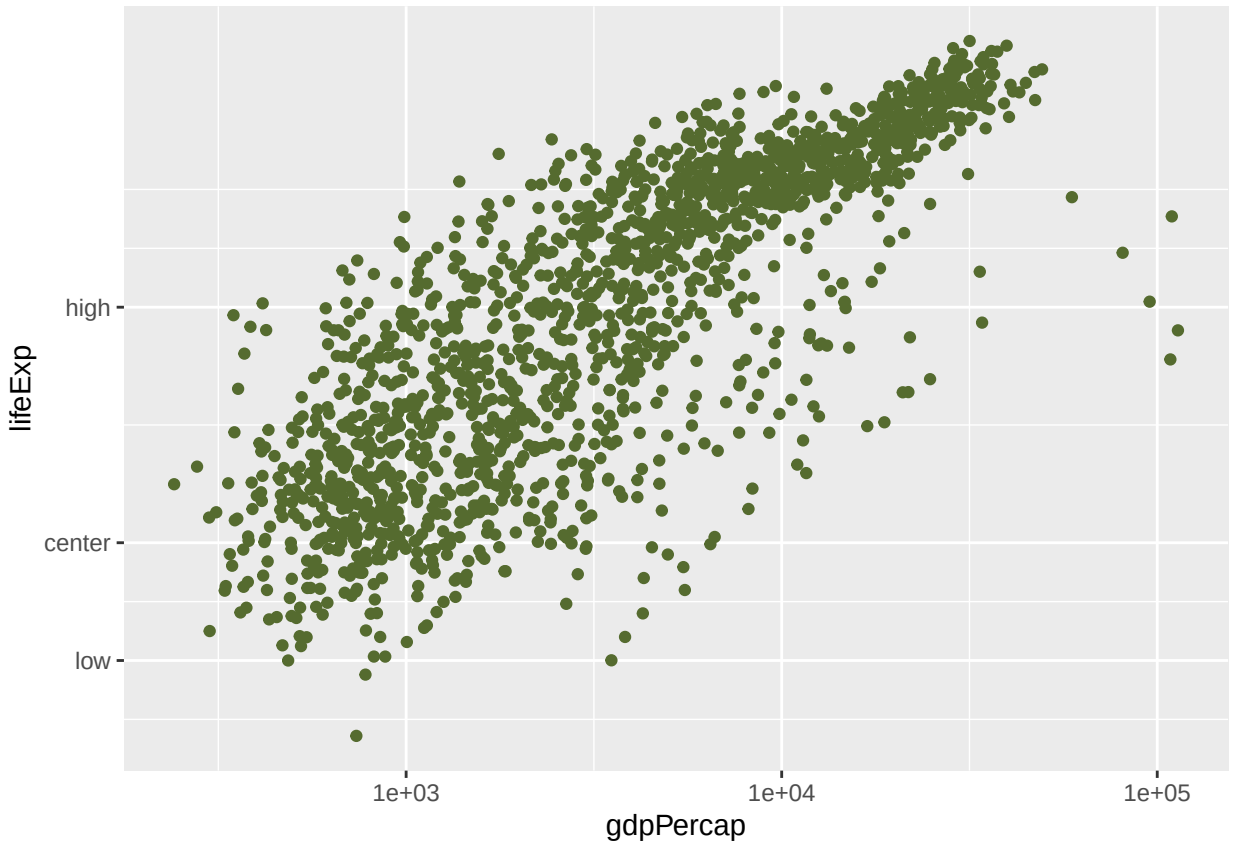
```
## Warning: Removed 1213 rows containing missing values (geom_point).
```



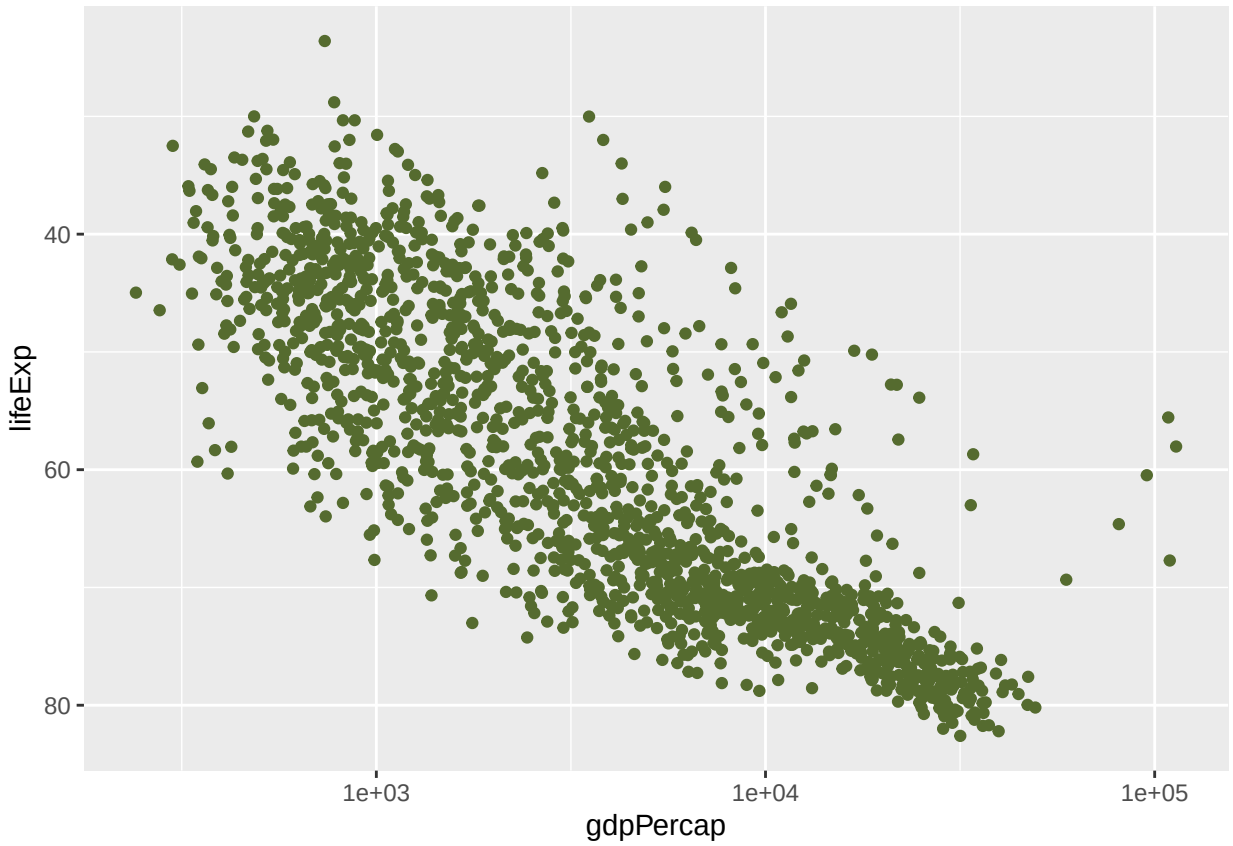

```
#Apply a a linear transformation to the Y axis position with defining the breaks
ggplot(gapminder, aes(pop, lifeExp)) +
  geom_point(colour = main_color) +
  scale_x_log10() +
  scale_y_continuous(breaks = c(0, 20, 25,40, 45,55, 60, 80))
```



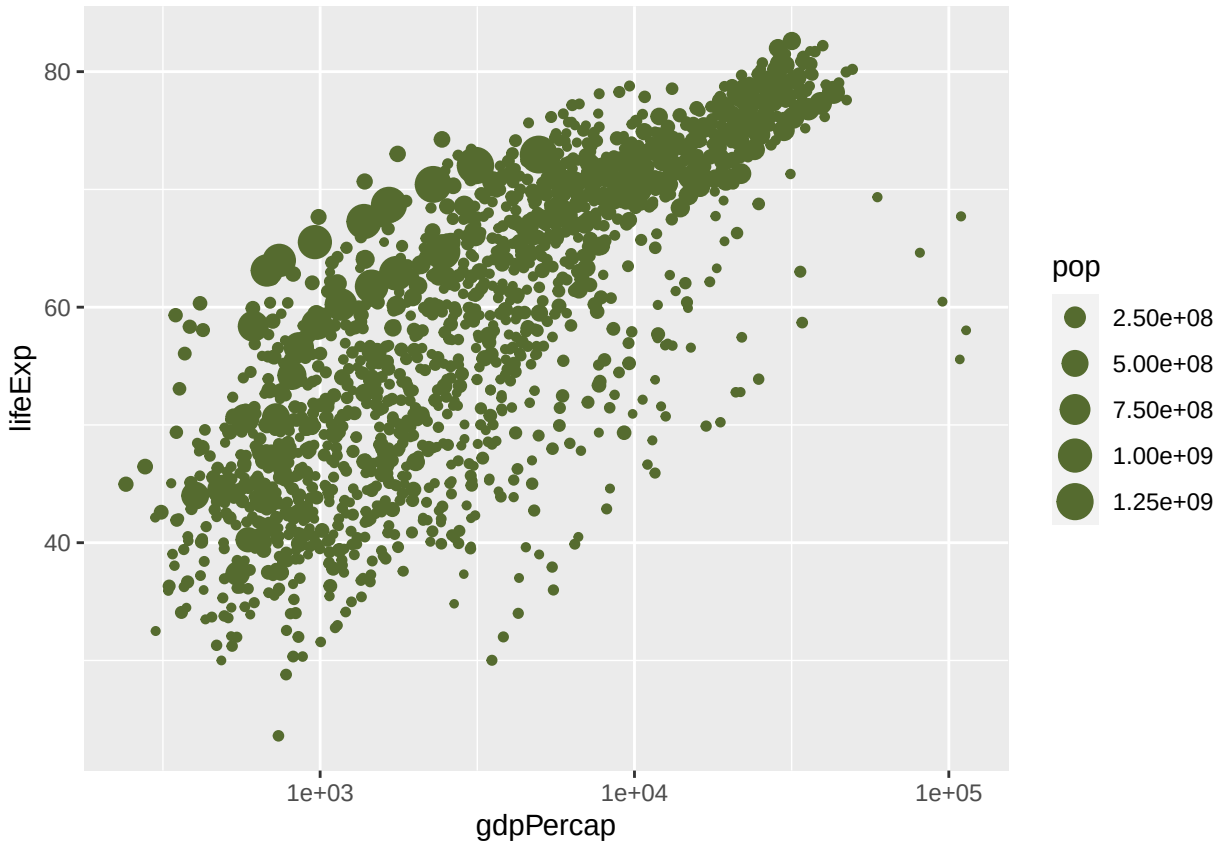

```
#Add labels
ggplot(gapminder, aes(gdpPercap, lifeExp)) +
  geom_point(colour = main_color) +
  scale_x_log10() +
  scale_y_continuous(breaks = c(30, 40, 60), label = c("low", "center", "high"))
```



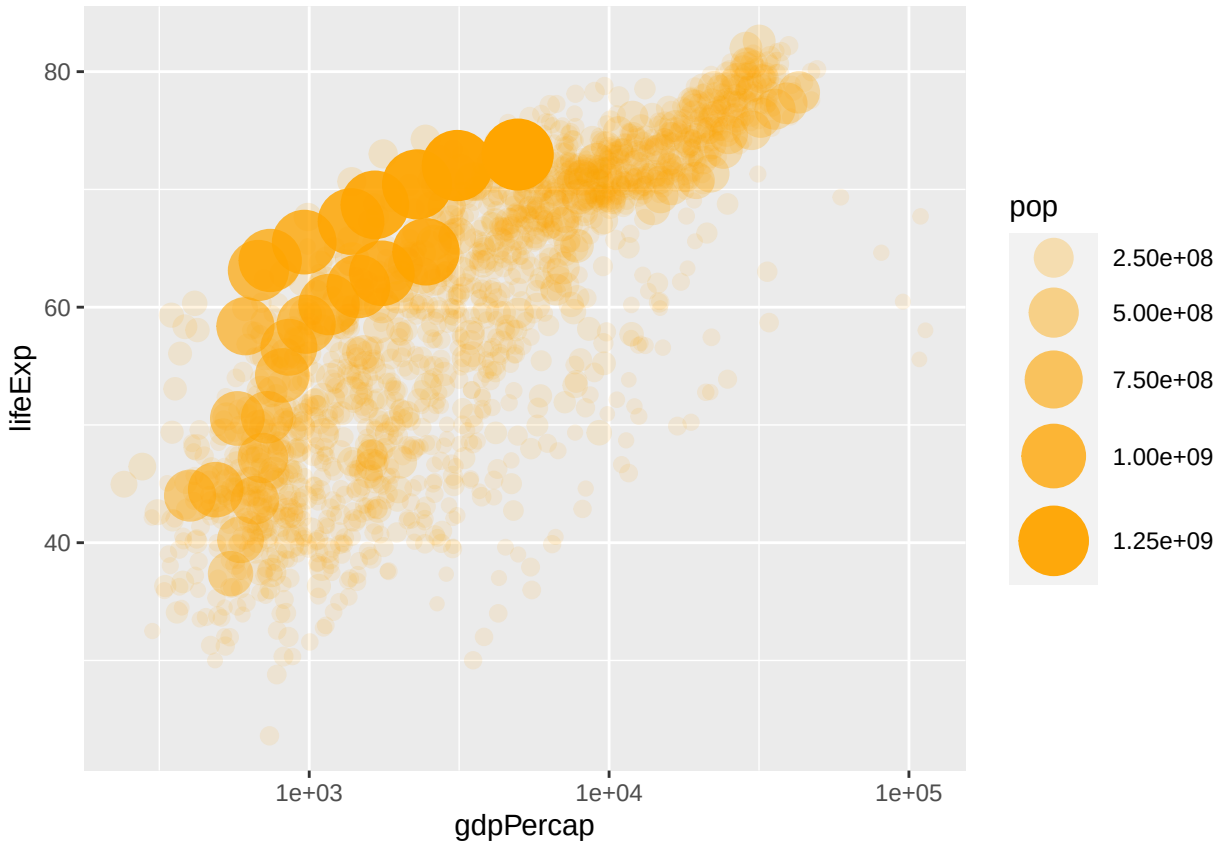

```
#Change the Y scale in reverse
ggplot(gapminder, aes(gdpPercap, lifeExp)) +
  geom_point(colour = main_color) +
  scale_x_log10() +
  scale_y_reverse()
```



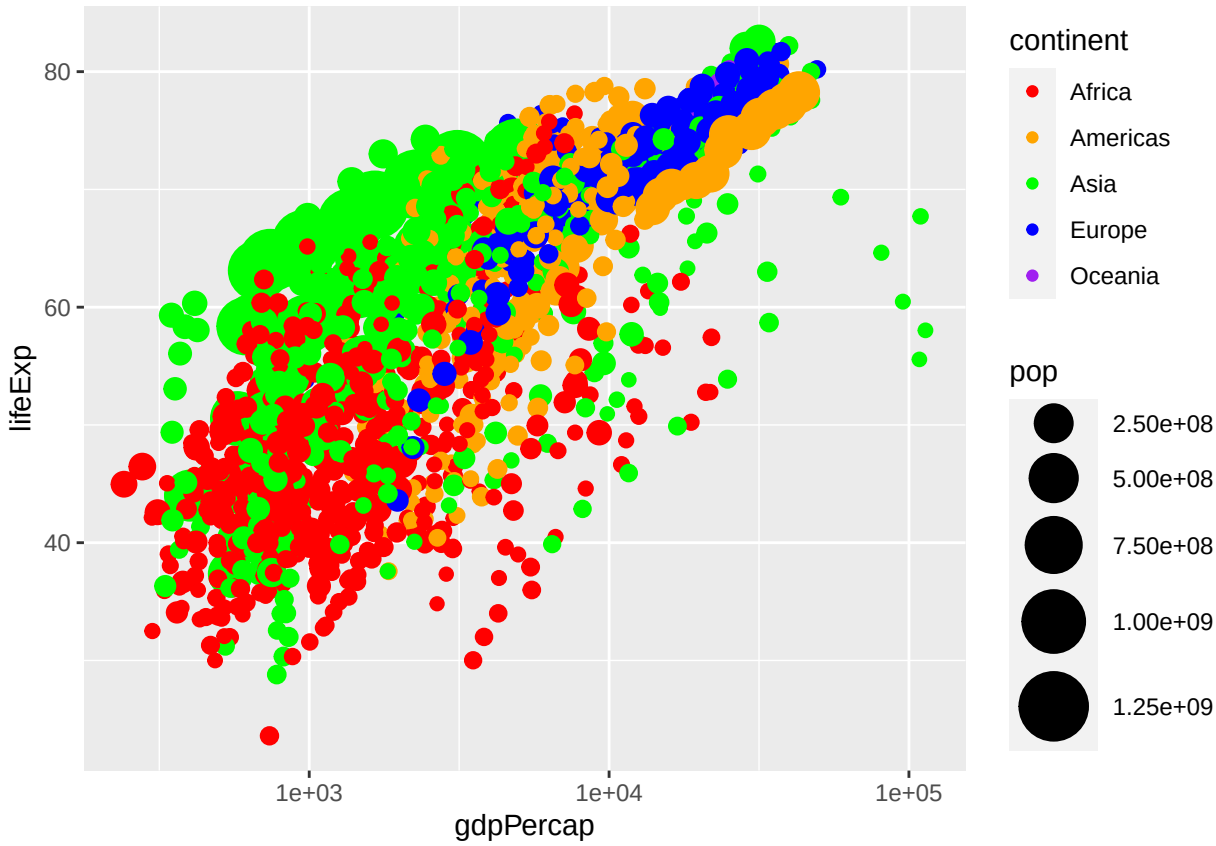

```
#Add another visual encoding size
ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop)) +
  geom_point(colour = main_color) +
  scale_x_log10()  +
  scale_size()
```




```
#Apply a scale rage to the variable size
ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop, alpha=pop)) +
  geom_point(colour = sub_color) +
  scale_x_log10()  +
  scale_size(range = c(2, 12))
```

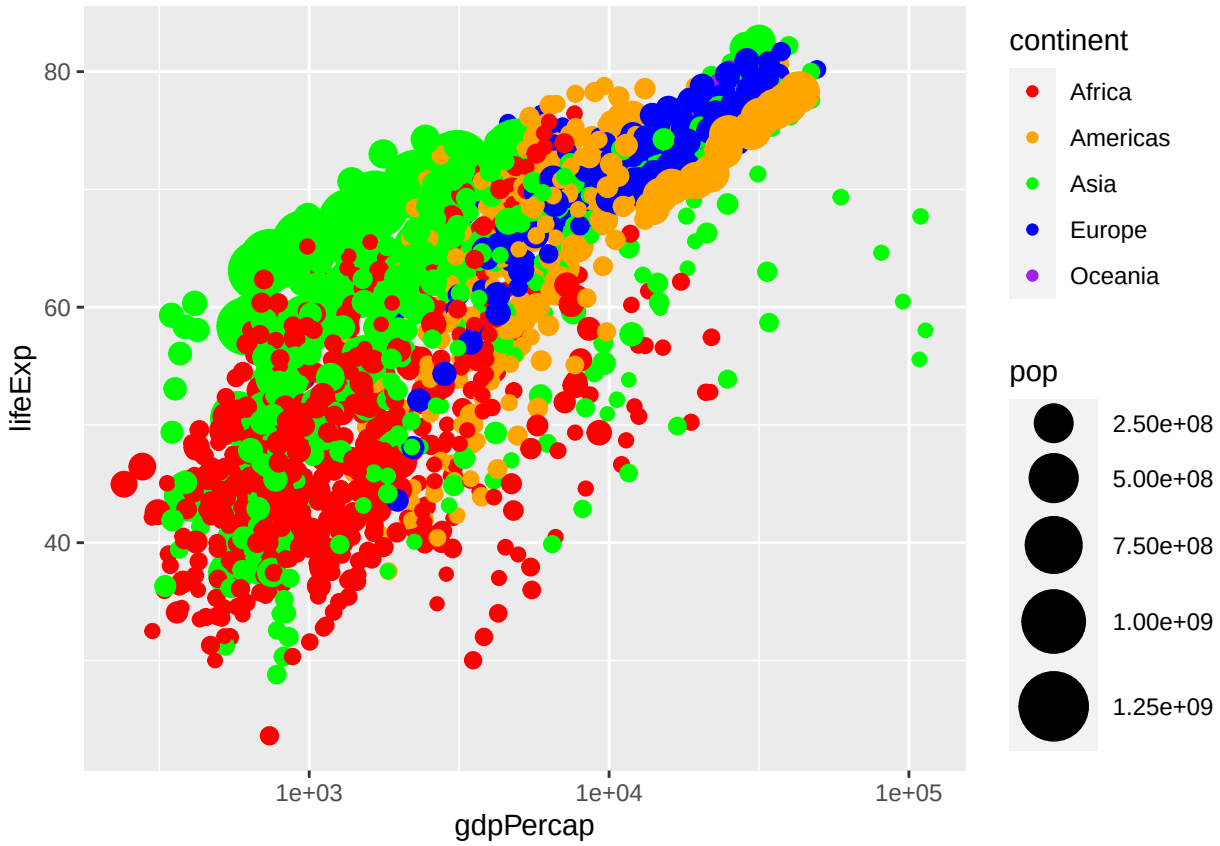



```
#Add another visual encoding color
ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop,  colour = continent)) +
  geom_point() +
  scale_x_log10()  +
  scale_size(range = c(2, 12)) +
  scale_color_manual(values=c('red','orange','green','blue','purple'))
```



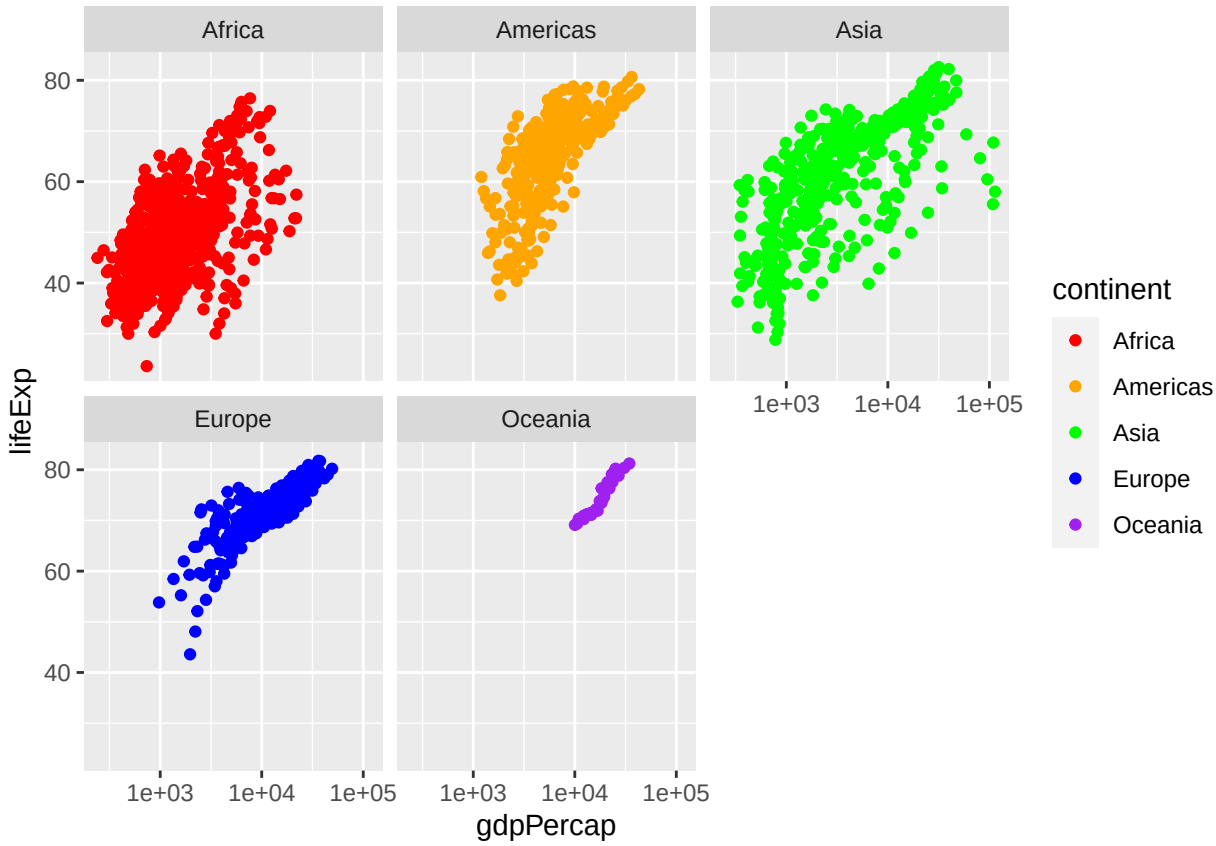

```
#Apply another scale to color
ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop,  color = continent)) +
  geom_point() +
  scale_x_log10()  +
  scale_size(range = c(2, 12)) +
  scale_colour_manual(values = continent_colors) +
  scale_color_manual(values=c('red','orange','green','blue','purple'))
```

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```



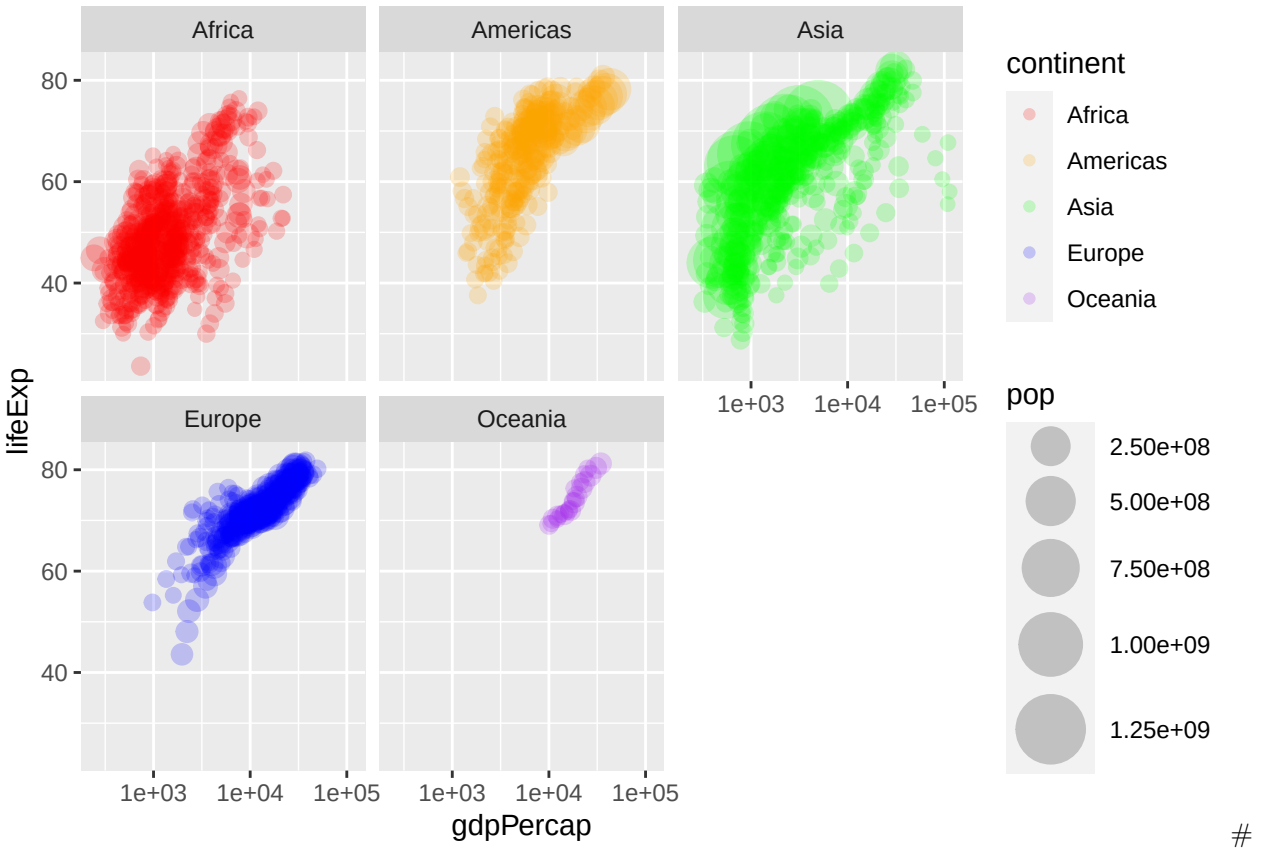

```
#Change to facet
ggplot(gapminder, aes(gdpPercap, lifeExp,  colour = continent)) +
  geom_point() +
  scale_x_log10()  +
  scale_size(range = c(2, 12)) +
  scale_colour_manual(values = continent_colors) +
  facet_wrap(~continent) +
  scale_color_manual(values=c('red','orange','green','blue','purple'))
```

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```




```
#Adding transparency
ggplot(gapminder, aes(gdpPercap, lifeExp, size = pop,  colour = continent)) +
  geom_point(alpha=0.2) +
  scale_x_log10()  +
  scale_size(range = c(2, 12)) +
  scale_colour_manual(values = continent_colors) +
  facet_wrap(~continent) +
  scale_color_manual(values=c('red','orange','green','blue','purple'))
```

```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```



#

Excercise - 11

```
#Check the data
names(economics)
```

```
## [1] "date"     "pce"      "pop"      "psavert"  "uempmed"  "unemploy"
```

```
head(economics, n=10)
```

```
## # A tibble: 10 x 6
##    date         pce    pop psavert uempmed unemploy
##    <date>     <dbl>  <dbl>   <dbl>   <dbl>    <dbl>
##  1 1967-07-01  507. 198712    12.6     4.5     2944
##  2 1967-08-01  510. 198911    12.6     4.7     2945
##  3 1967-09-01  516. 199113    11.9     4.6     2958
##  4 1967-10-01  512. 199311    12.9     4.9     3143
##  5 1967-11-01  517. 199498    12.8     4.7     3066
##  6 1967-12-01  525. 199657    11.8     4.8     3018
##  7 1968-01-01  531. 199808    11.7     5.1     2878
##  8 1968-02-01  534. 199920    12.3     4.5     3001
##  9 1968-03-01  544. 200056    11.7     4.1     2877
## 10 1968-04-01  544  200208    12.3     4.6     2709
```

```
str(economics)
```

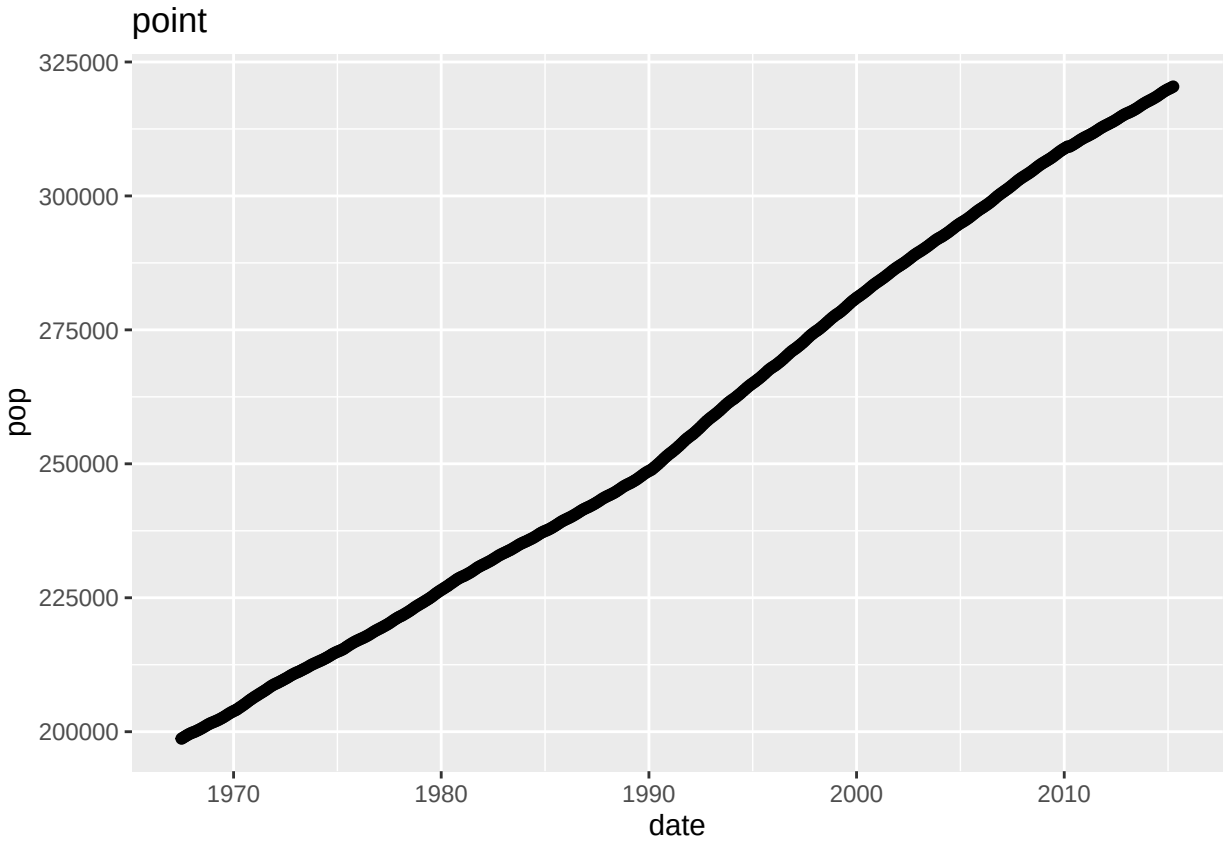
```
## tibble [574 x 6] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ date    : Date[1:574], format: "1967-07-01" "1967-08-01" ...
##  $ pce     : num [1:574] 507 510 516 512 517 ...
##  $ pop     : num [1:574] 198712 198911 199113 199311 199498 ...
##  $ psavert : num [1:574] 12.6 12.6 11.9 12.9 12.8 11.8 11.7 12.3 11.7 12.3 ...
```

```
##  $ uempmed : num [1:574] 4.5 4.7 4.6 4.9 4.7 4.8 5.1 4.5 4.1 4.6 ...
##  $ unemploy: num [1:574] 2944 2945 2958 3143 3066 ...
```

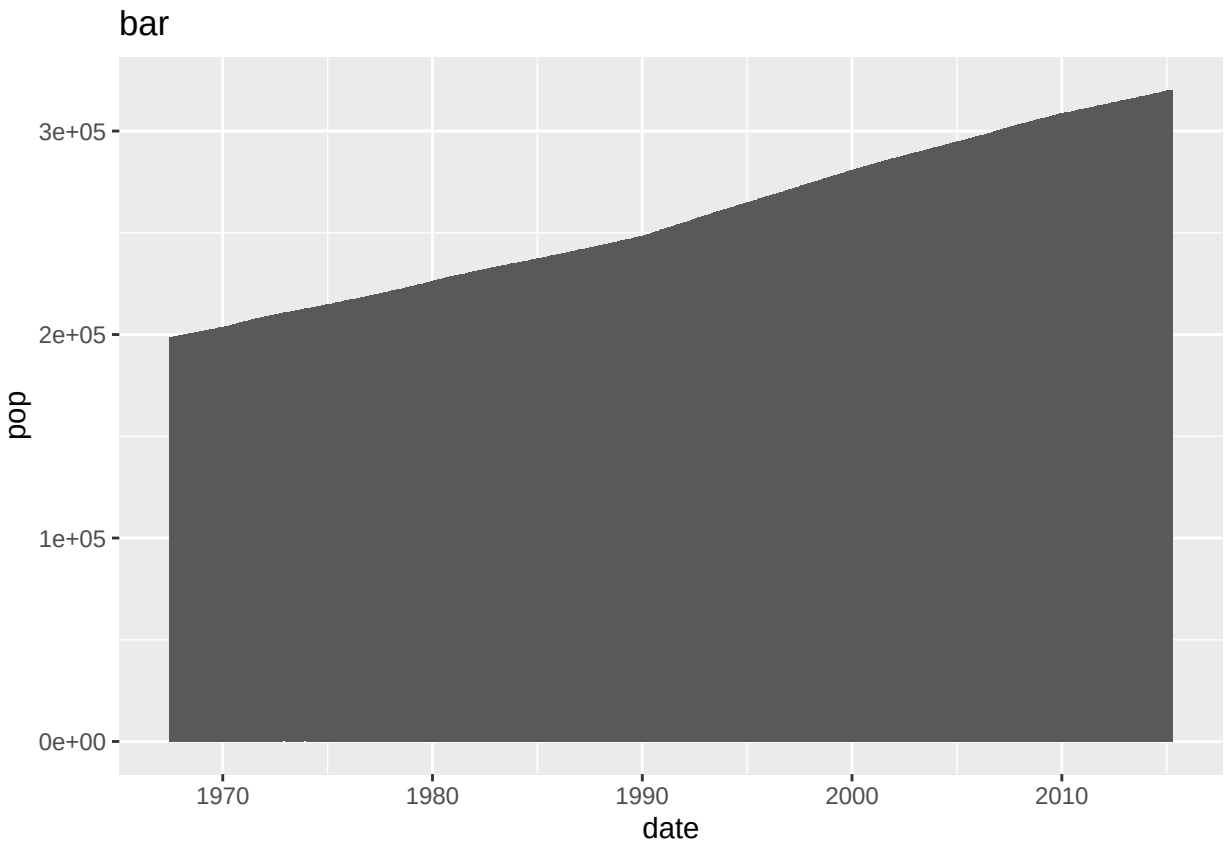
```
summary(economics)
```

```
##       date                 pce               pop            psavert
##  Min.   :1967-07-01   Min.   :  506.7   Min.   :198712   Min.   : 2.200
##  1st Qu.:1979-06-08   1st Qu.: 1578.3   1st Qu.:224896   1st Qu.: 6.400
##  Median :1991-05-16   Median : 3936.8   Median :253060   Median : 8.400
##  Mean   :1991-05-17   Mean   : 4820.1   Mean   :257160   Mean   : 8.567
##  3rd Qu.:2003-04-23   3rd Qu.: 7626.3   3rd Qu.:290291   3rd Qu.:11.100
##  Max.   :2015-04-01   Max.   :12193.8   Max.   :320402   Max.   :17.300
##     uempmed          unemploy
##  Min.   : 4.000   Min.   : 2685
##  1st Qu.: 6.000   1st Qu.: 6284
##  Median : 7.500   Median : 7494
##  Mean   : 8.609   Mean   : 7771
##  3rd Qu.: 9.100   3rd Qu.: 8686
##  Max.   :25.200   Max.   :15352
```

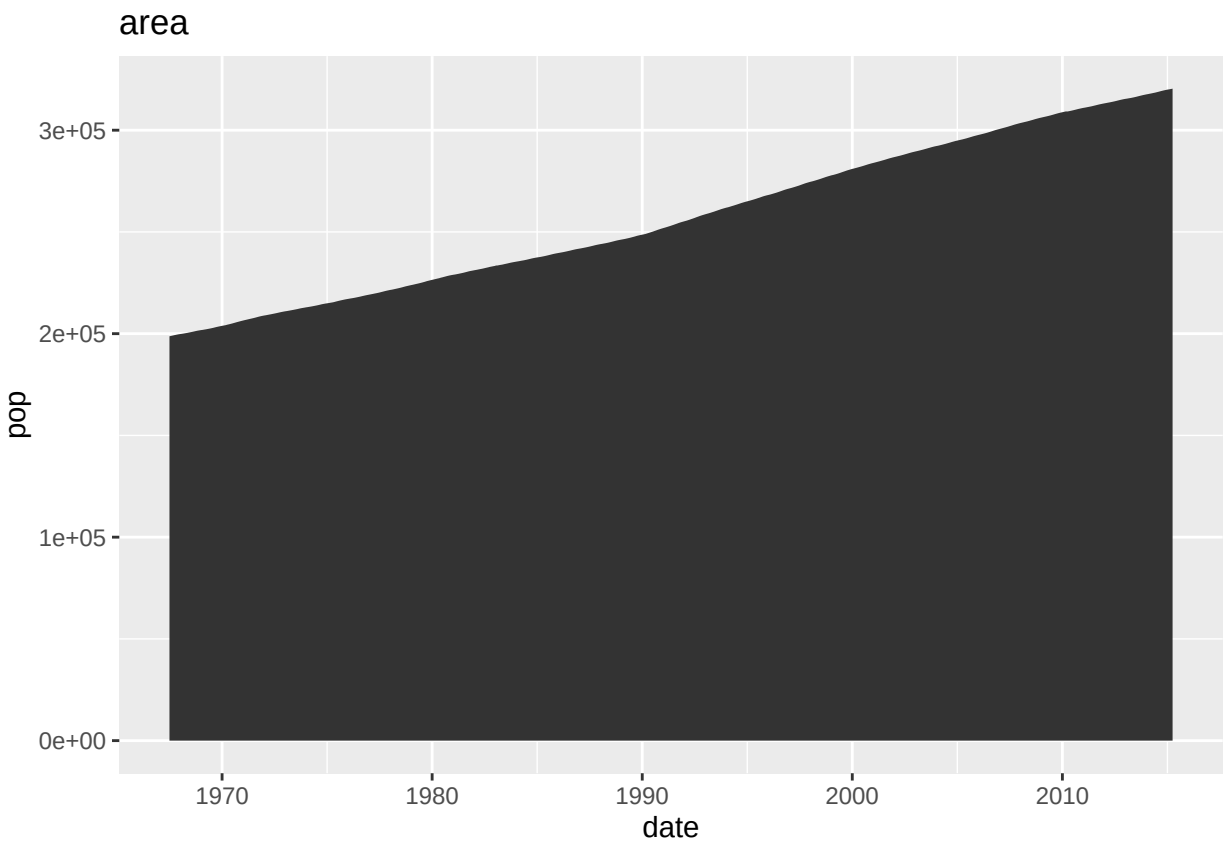
```
#1 General plot definition and theme saved in a variable
e <-ggplot(economics, aes(date, pop))

#Showing the different geoms for time data
e + geom_point() + ggtitle("point")
```



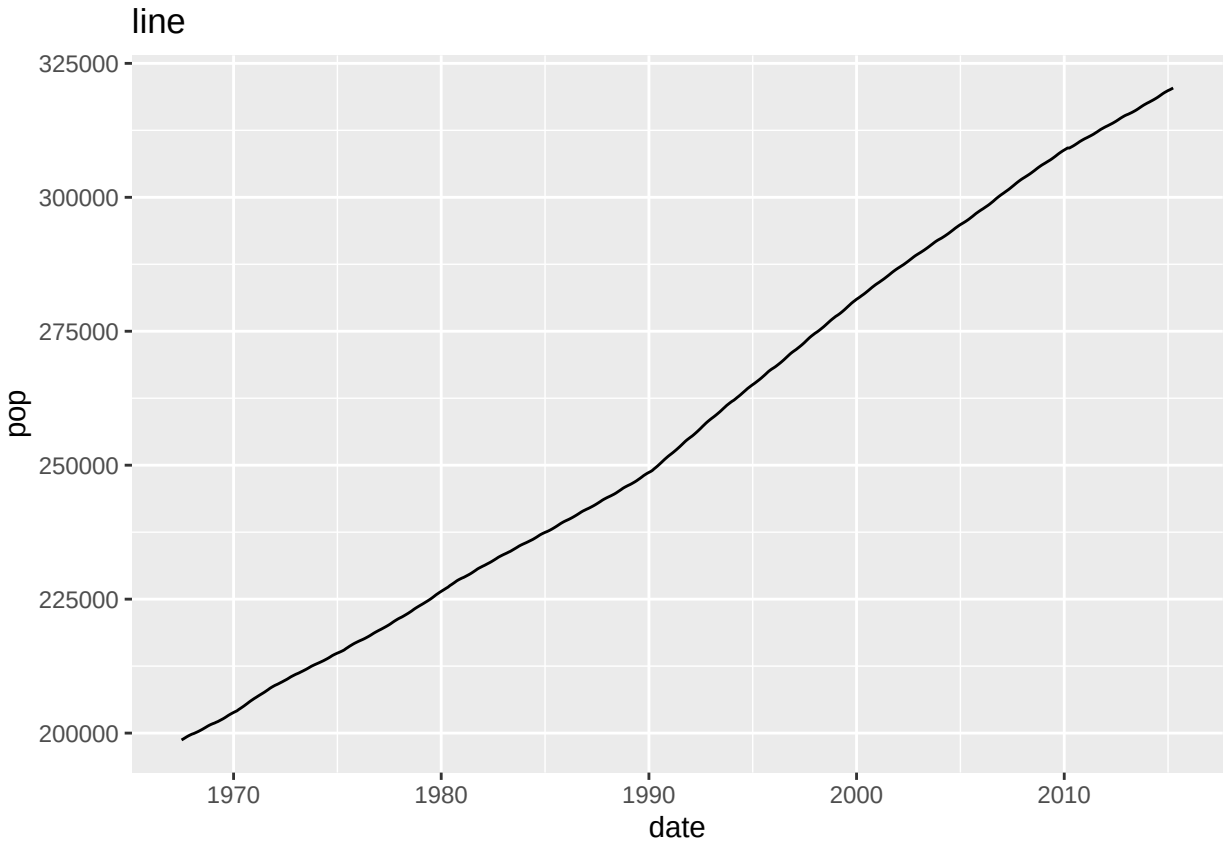
```
e + geom_bar(stat = "identity") + ggtitle("bar")
```




```
e + geom_area() + ggtitle("area")
```




```
e + geom_line() + ggtitle("line")
```




```
#2 Showing trends for different categories

#Check the data
names(economics_long)
```

```
## [1] "date"     "variable" "value"    "value01"
```

```
head(economics_long, n=10)
```

```
## # A tibble: 10 x 4
##    date       variable value  value01
##    <date>     <chr>    <dbl>    <dbl>
##  1 1967-07-01 pce       507. 0
##  2 1967-08-01 pce       510. 0.000265
##  3 1967-09-01 pce       516. 0.000762
##  4 1967-10-01 pce       512. 0.000471
##  5 1967-11-01 pce       517. 0.000916
##  6 1967-12-01 pce       525. 0.00157
##  7 1968-01-01 pce       531. 0.00207
##  8 1968-02-01 pce       534. 0.00230
##  9 1968-03-01 pce       544. 0.00322
## 10 1968-04-01 pce       544  0.00319
```

```
str(economics_long)
```

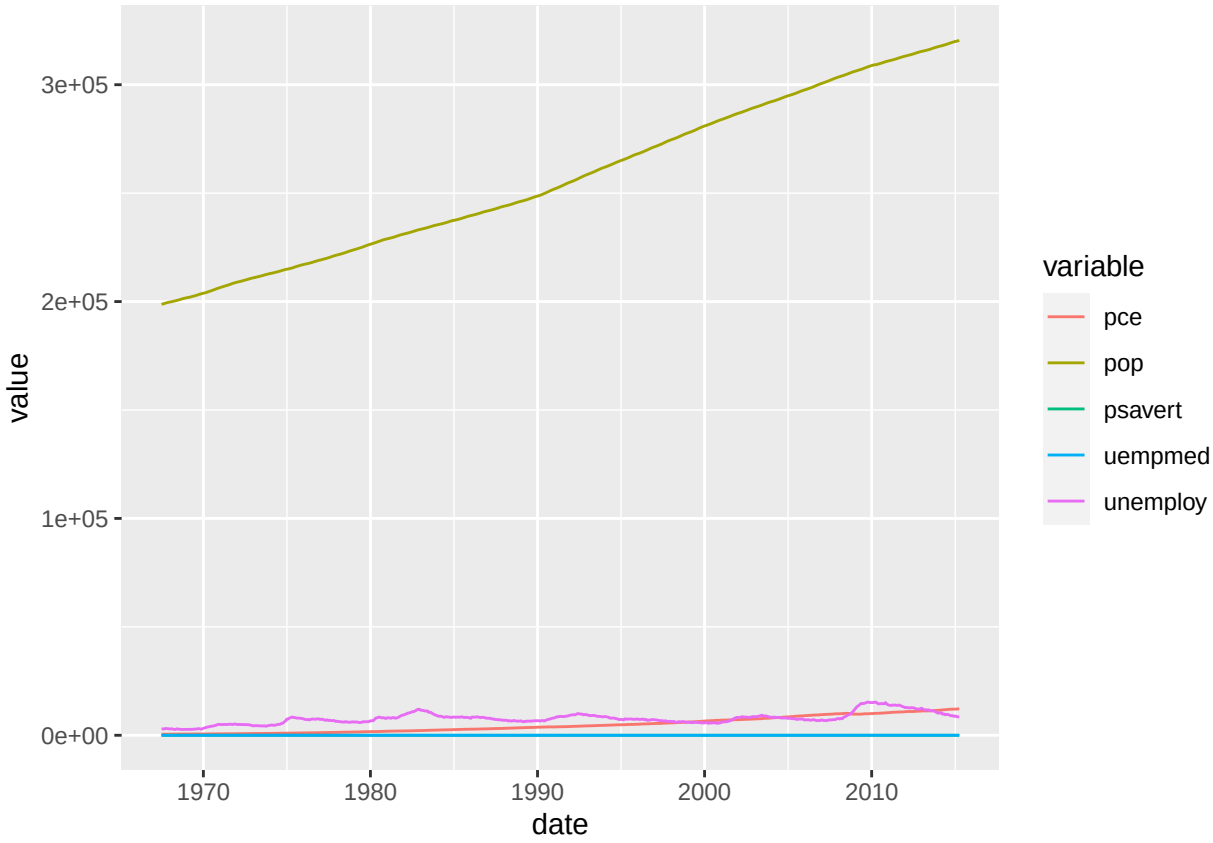
```
## tibble [2,870 x 4] (S3: tbl_df/tbl/data.frame)
##  $ date    : Date[1:2870], format: "1967-07-01" "1967-08-01" ...
##  $ variable: chr [1:2870] "pce" "pce" "pce" "pce" ...
##  $ value   : num [1:2870] 507 510 516 512 517 ...
```

```
##  $ value01 : num [1:2870] 0 0.000265 0.000762 0.000471 0.000916 ...
```

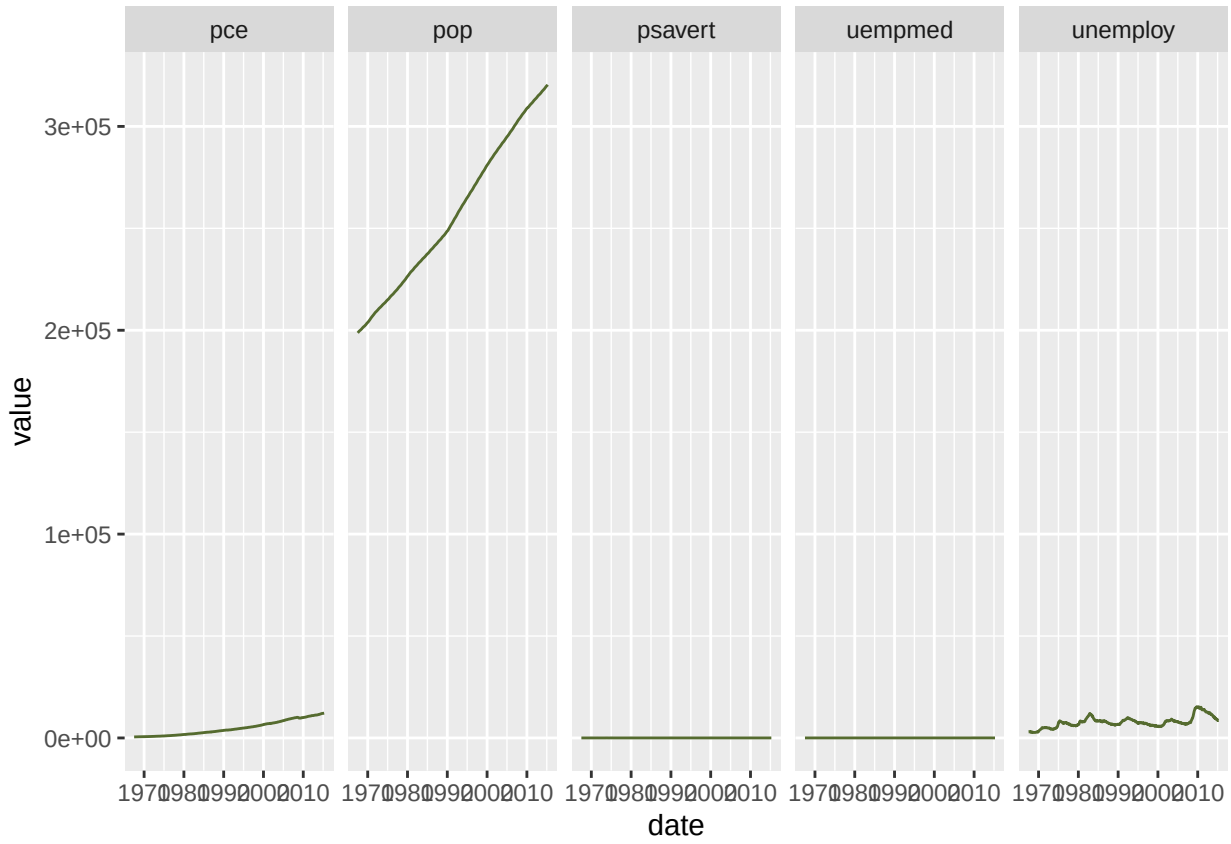
```
summary(economics_long)
```

```
##       date               variable              value             value01       
##  Min.   :1967-07-01   Length:2870        Min.   :     2.2   Min.   :0.0000  
##  1st Qu.:1979-06-01   Class :character   1st Qu.:     8.7   1st Qu.:0.1651  
##  Median :1991-05-16   Mode  :character   Median :  3421.9   Median :0.3274  
##  Mean   :1991-05-17                      Mean   : 53953.6   Mean   :0.3780  
##  3rd Qu.:2003-05-01                      3rd Qu.: 10111.6   3rd Qu.:0.5591  
##  Max.   :2015-04-01                      Max.   :320402.3   Max.   :1.0000
```

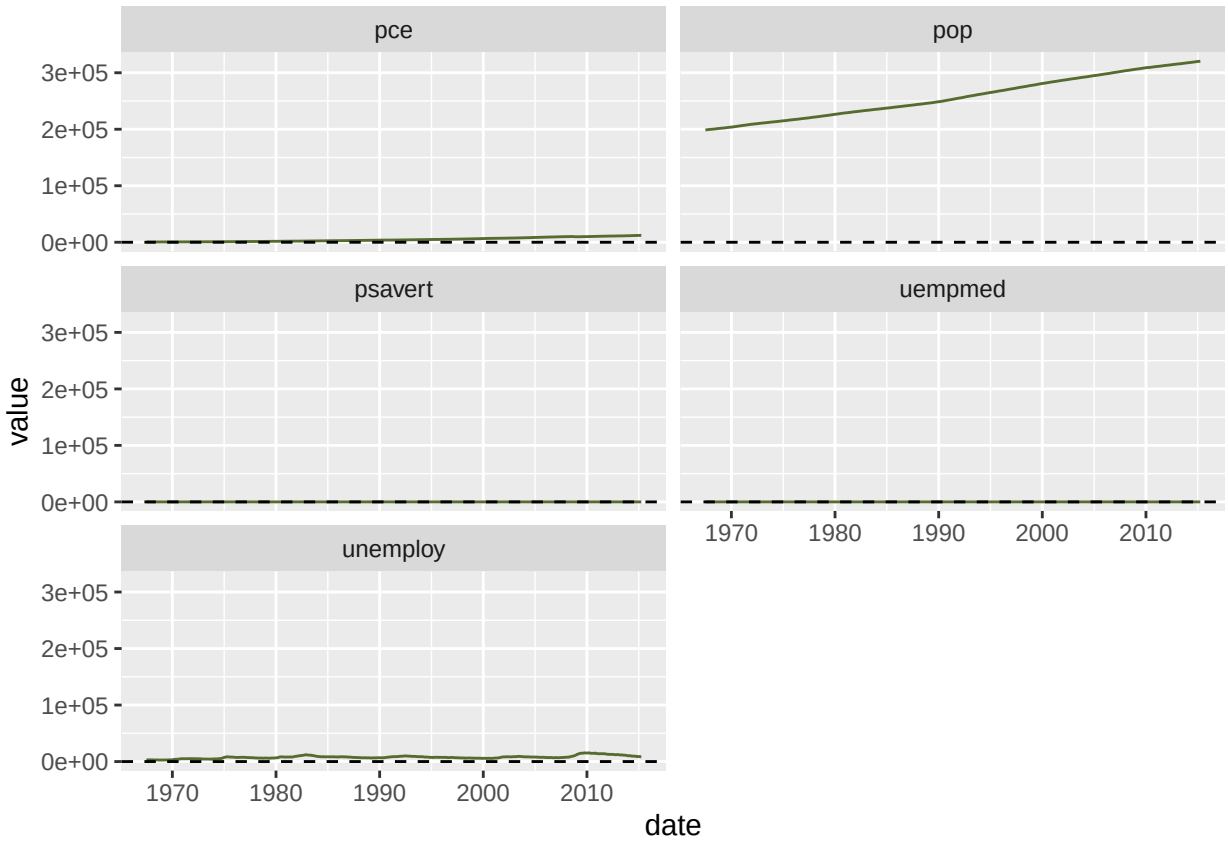
```
#Check the options
?geom_line()

#All in one chart, what is the problem?
ggplot(economics_long, aes(date, value, colour = variable)) +
  geom_line()
```




```
#Changing to a small multiple
ggplot(economics_long, aes(date, value)) +
  geom_line(colour=main_color) +
  facet_wrap(~variable, ncol=5)
```

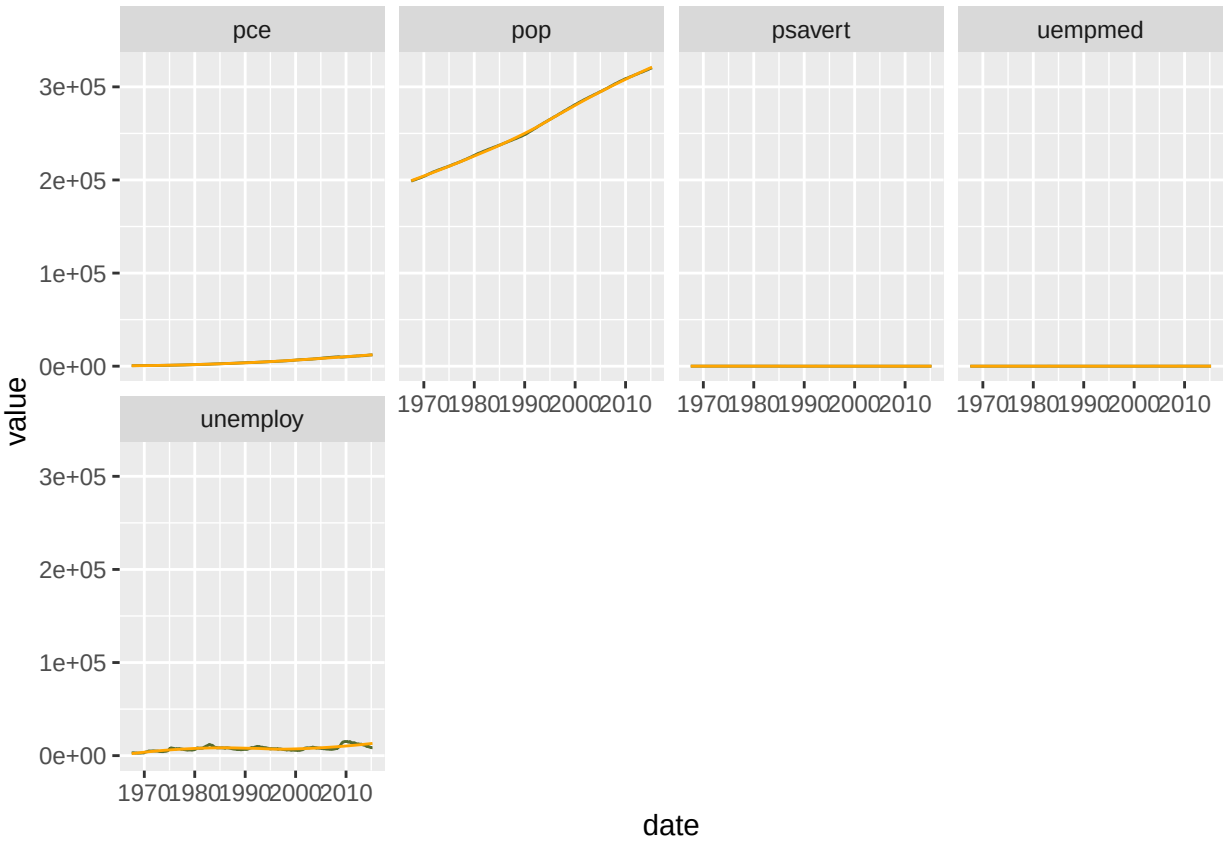



```
#Changing to a small multiple - adding a reference line to make them comparable
ggplot(economics_long, aes(date, value)) +
  geom_line(colour=main_color) +
  facet_wrap(~variable, ncol=2) +
  geom_hline(yintercept = 0.5, linetype = 2)
```




```
#Changing to a small multiple - adding a reference line to make them comparable
ggplot(economics_long, aes(date, value)) +
  geom_line(colour=main_color) +
  facet_wrap(~variable, ncol=4) +
  geom_smooth(se=FALSE, colour= sub_color, size=0.5)
```

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




```
#Group element in line and other charts

#Problem with line charts

#Check the data
names(Oxboys)
```

```
## [1] "Subject"  "age"      "height"   "Occasion"
```

```
head(Oxboys, n=10)
```

```
## Grouped Data: height ~ age | Subject
##    Subject     age height Occasion
## 1        1 -1.0000  140.5        1
## 2        1 -0.7479  143.4        2
## 3        1 -0.4630  144.8        3
## 4        1 -0.1643  147.1        4
## 5        1 -0.0027  147.7        5
## 6        1  0.2466  150.2        6
## 7        1  0.5562  151.7        7
## 8        1  0.7781  153.3        8
## 9        1  0.9945  155.8        9
## 10       2 -1.0000  136.9        1
```

```
str(Oxboys)
```

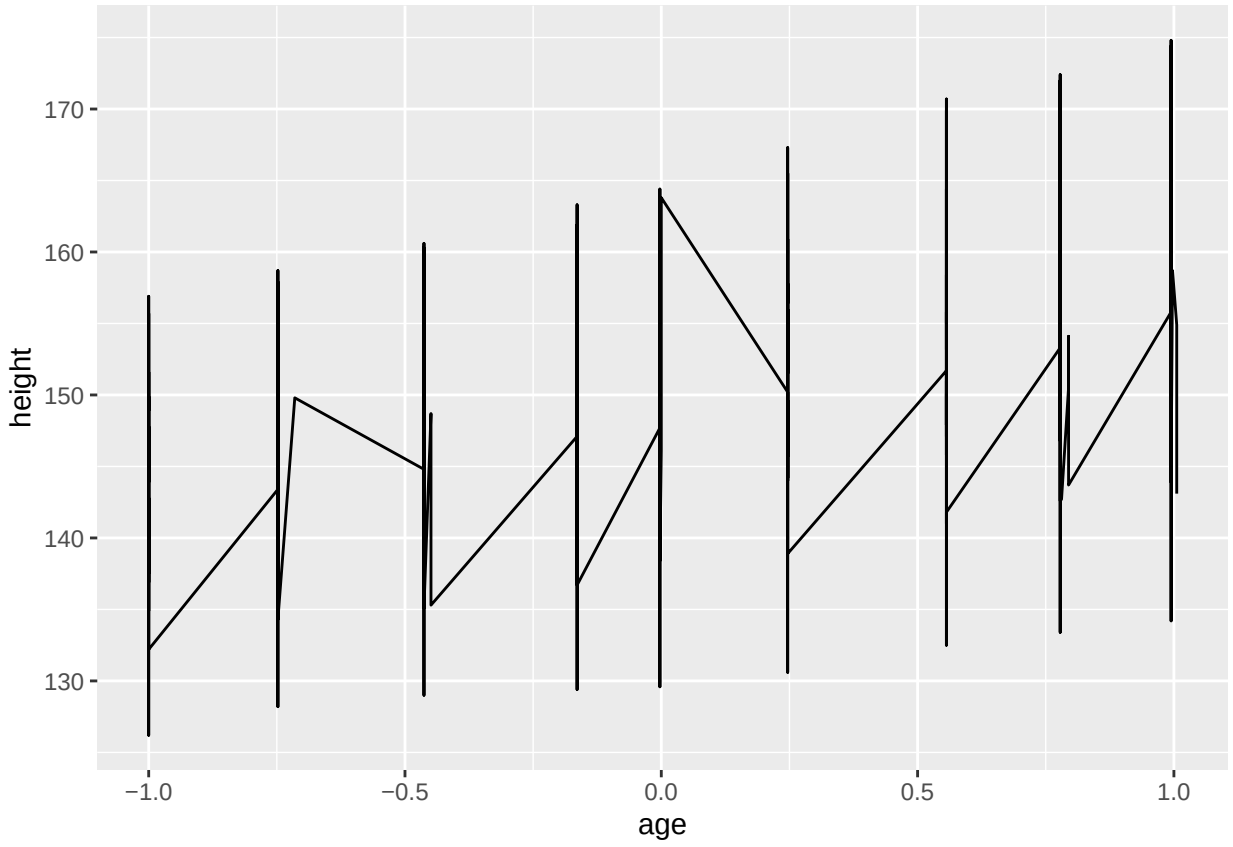
```
## Classes 'nfnGroupedData', 'nfGroupedData', 'groupedData' and 'data.frame':   234 obs. of  4 variables
##  $ Subject : Ord.factor w/ 26 levels "10"<"26"<"25"<..: 13 13 13 13 13 13 13 13 13 5 ...
##  $ age     : num  -1 -0.7479 -0.463 -0.1643 -0.0027 ...
```

```
##  $ height  : num  140 143 145 147 148 ...
##  $ Occasion: Ord.factor w/ 9 levels "1"<"2"<"3"<"4"<..: 1 2 3 4 5 6 7 8 9 1 ...
##  - attr(*, "formula")=Class 'formula'  language height ~ age | Subject
##   .. ..- attr(*, ".Environment")=<environment: R_GlobalEnv>
##  - attr(*, "labels")=List of 2
##   ..$ y: chr "Height"
##   ..$ x: chr "Centered age"
##  - attr(*, "units")=List of 1
##   ..$ y: chr "(cm)"
##  - attr(*, "FUN")=function (x)
##   ..- attr(*, "source")= chr "function (x) max(x, na.rm = TRUE)"
##  - attr(*, "order.groups")= logi TRUE
```

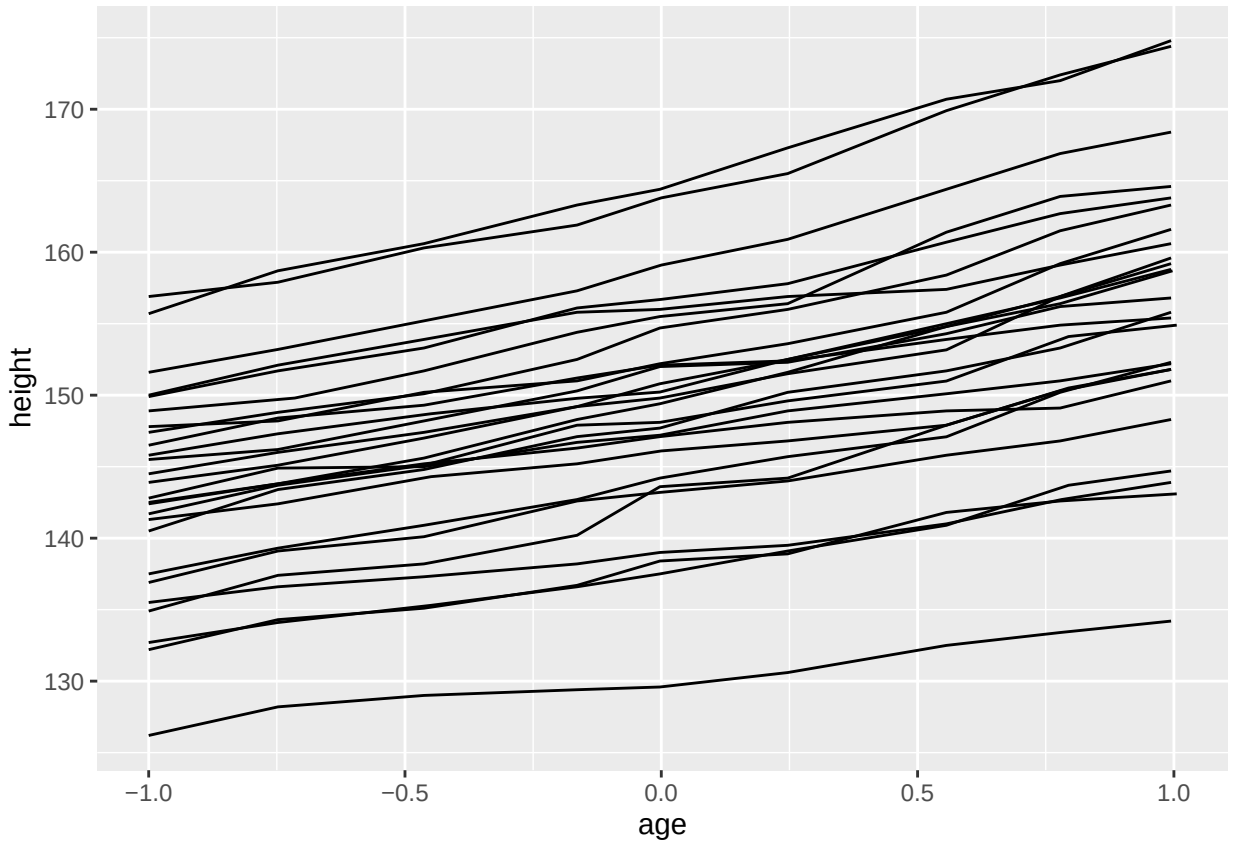
```
summary(Oxboys)
```

```
##     Subject         age               height         Occasion
##  10     :  9   Min.   :-1.00000   Min.   :126.2   1      :26
##  26     :  9   1st Qu.:-0.46300   1st Qu.:143.8   2      :26
##  25     :  9   Median :-0.00270   Median :149.5   3      :26
##  9      :  9   Mean   : 0.02263   Mean   :149.5   4      :26
##  2      :  9   3rd Qu.: 0.55620   3rd Qu.:155.5   5      :26
##  6      :  9   Max.   : 1.00550   Max.   :174.8   6      :26
##  (Other):180                                      (Other):78
```

```
# A single line tries to connect all the observations
h<- ggplot(Oxboys, aes(age, height))
h + geom_line()
```

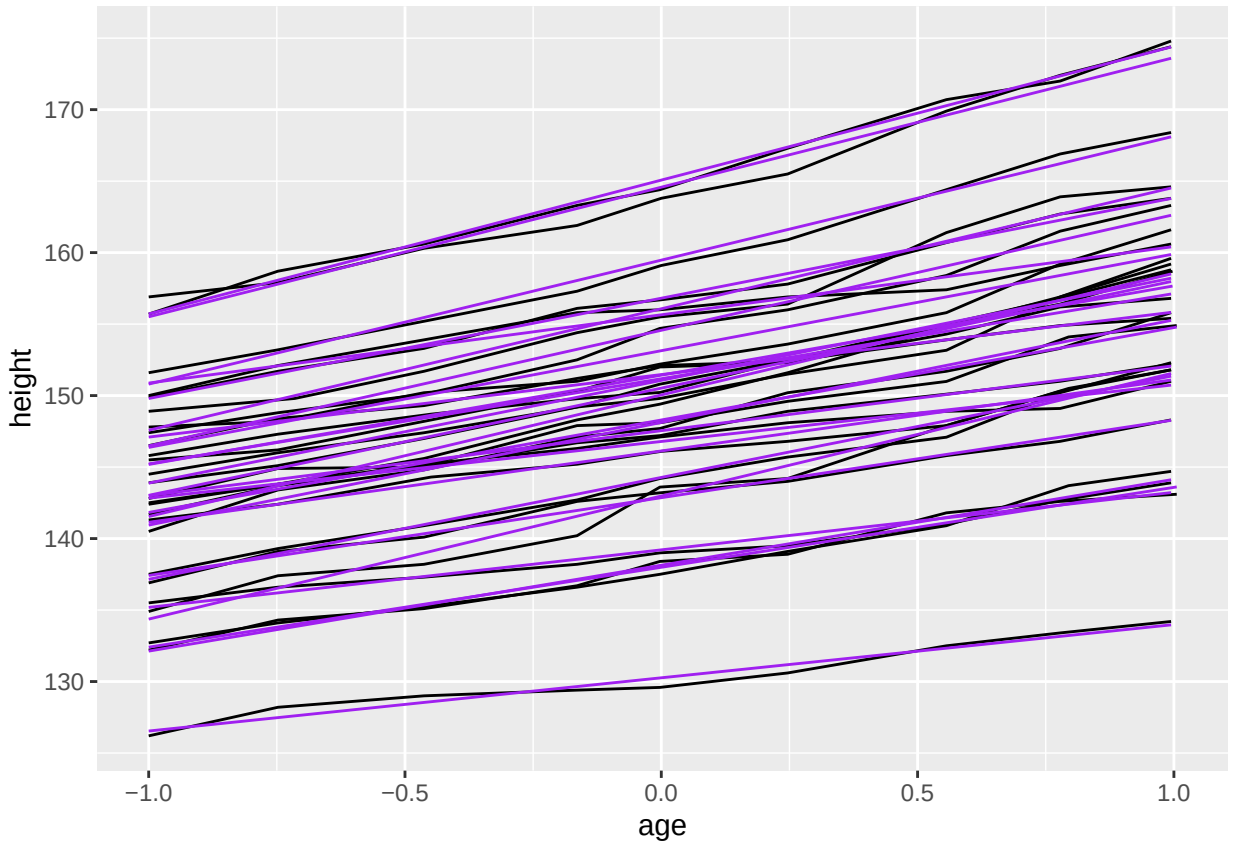


```
#Grouping the observation by the variables subject
h1 <- ggplot(Oxboys, aes(age, height, group=Subject))
h1 + geom_line()
```



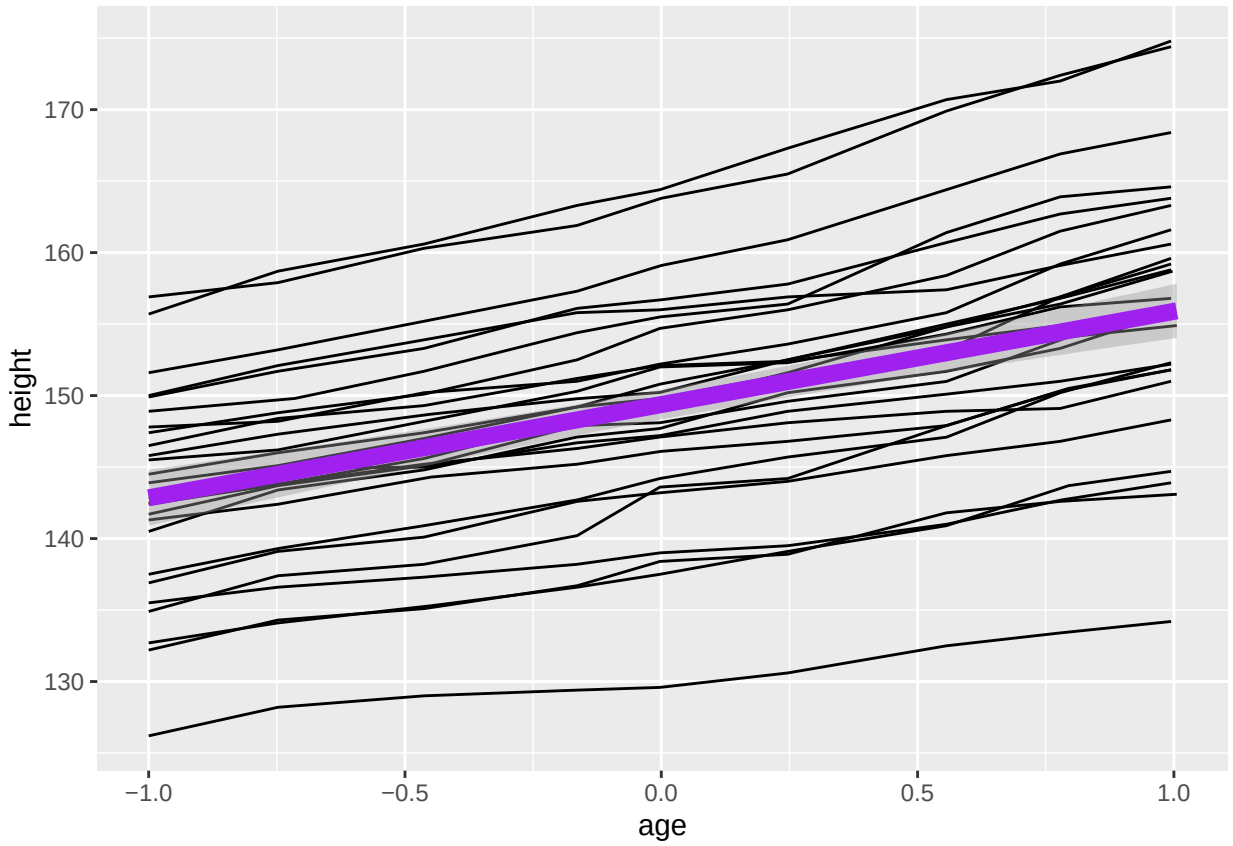

```
#Introducing the smooth element

# Using the group aesthetic with both geom_line() and geom_smooth()
# groups the data the same way for both layers
h1 + geom_line() +
  geom_smooth(aes(), colour = 'purple', size = 0.5, method = "lm", se = FALSE)
```

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



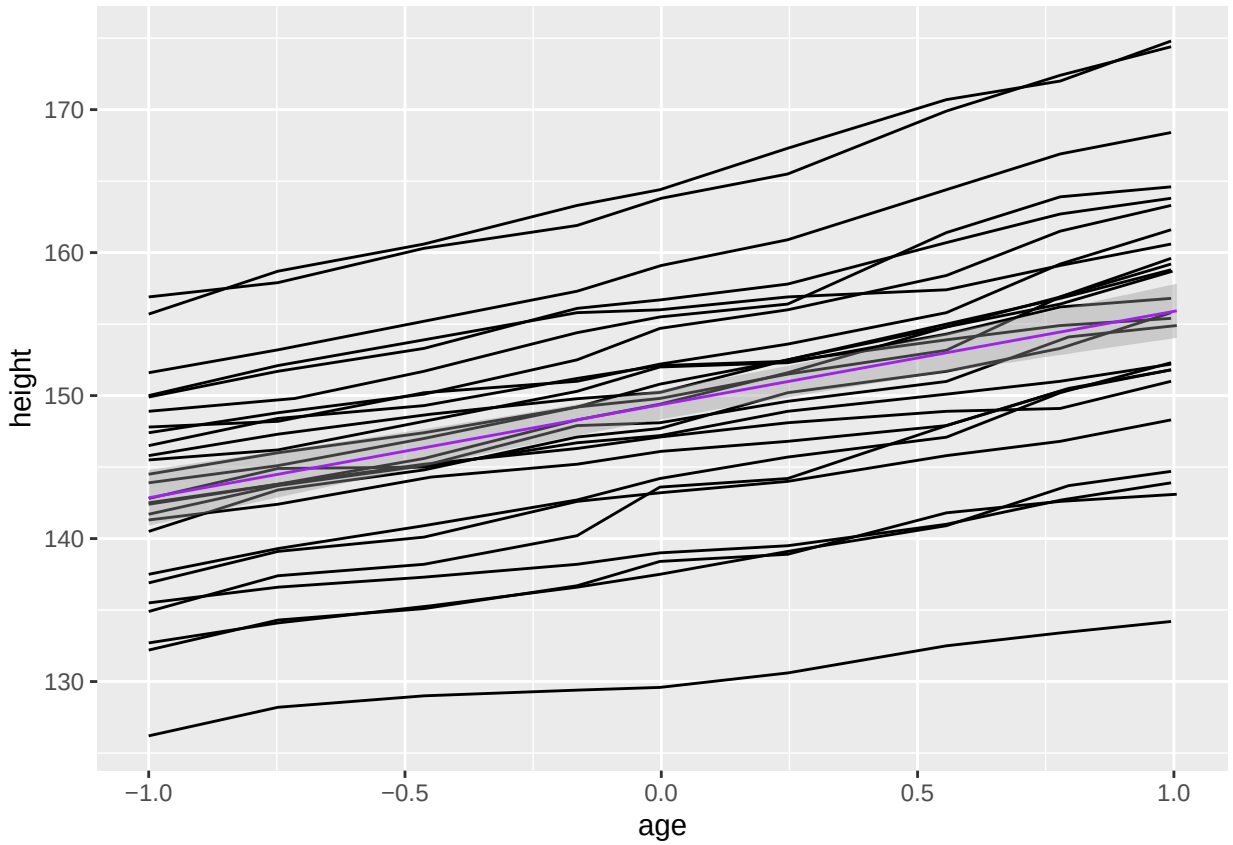

```
# Changing the group aesthetic for the smoother layer
# fits a single line of best fit across all boys
h1 + geom_line() +
  geom_smooth(aes(group = 1), colour = 'purple', size = 3, method = "lm", se = TRUE)
```

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

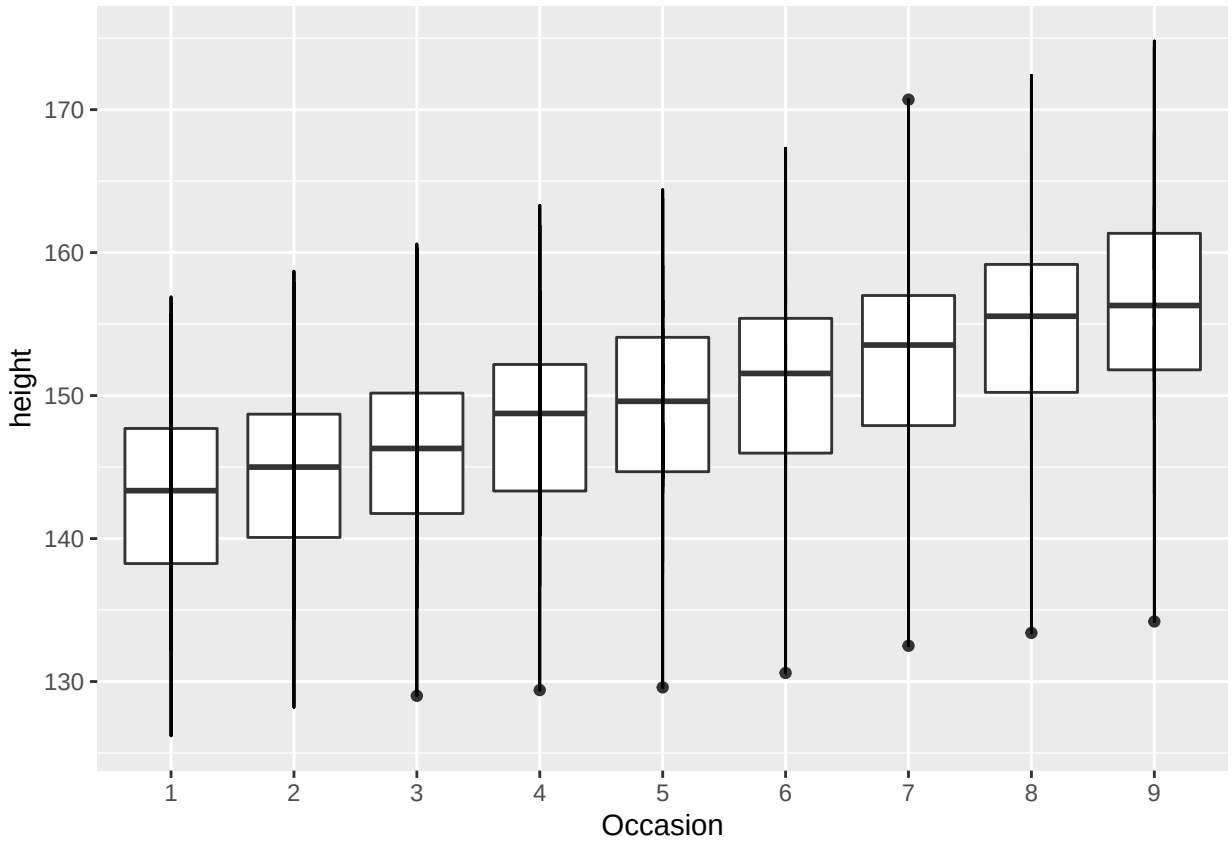



```
# Adding a confidence intervall
h1 + geom_line() +
  geom_smooth(aes(group = 1), colour = 'purple', size = 0.5, method = "lm", se = TRUE)
```

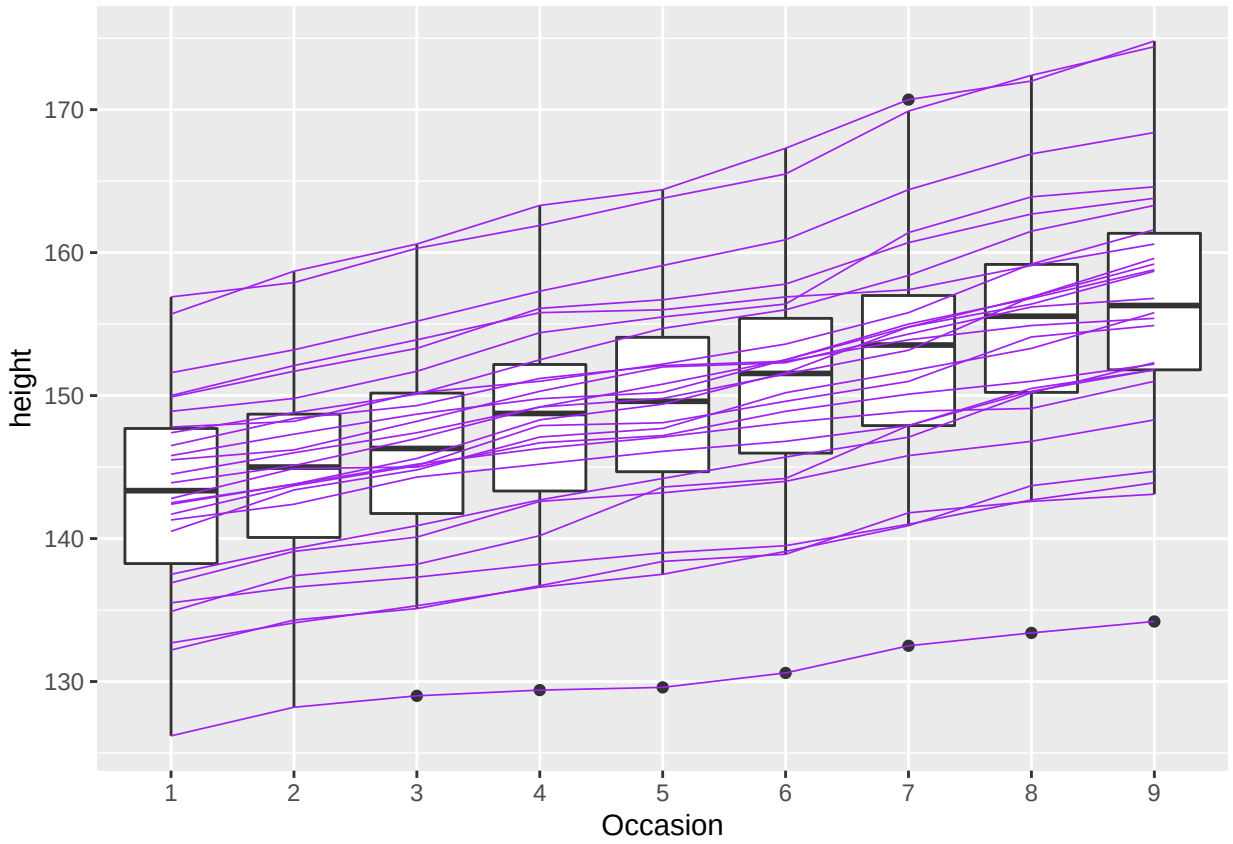
```
## `geom_smooth()` using formula 'y ~ x'
```




```
# Now we combine a box-plot with the line chart
h2 <- ggplot(Oxboys, aes(Occasion, height))
h2 + geom_boxplot() + geom_line()
```

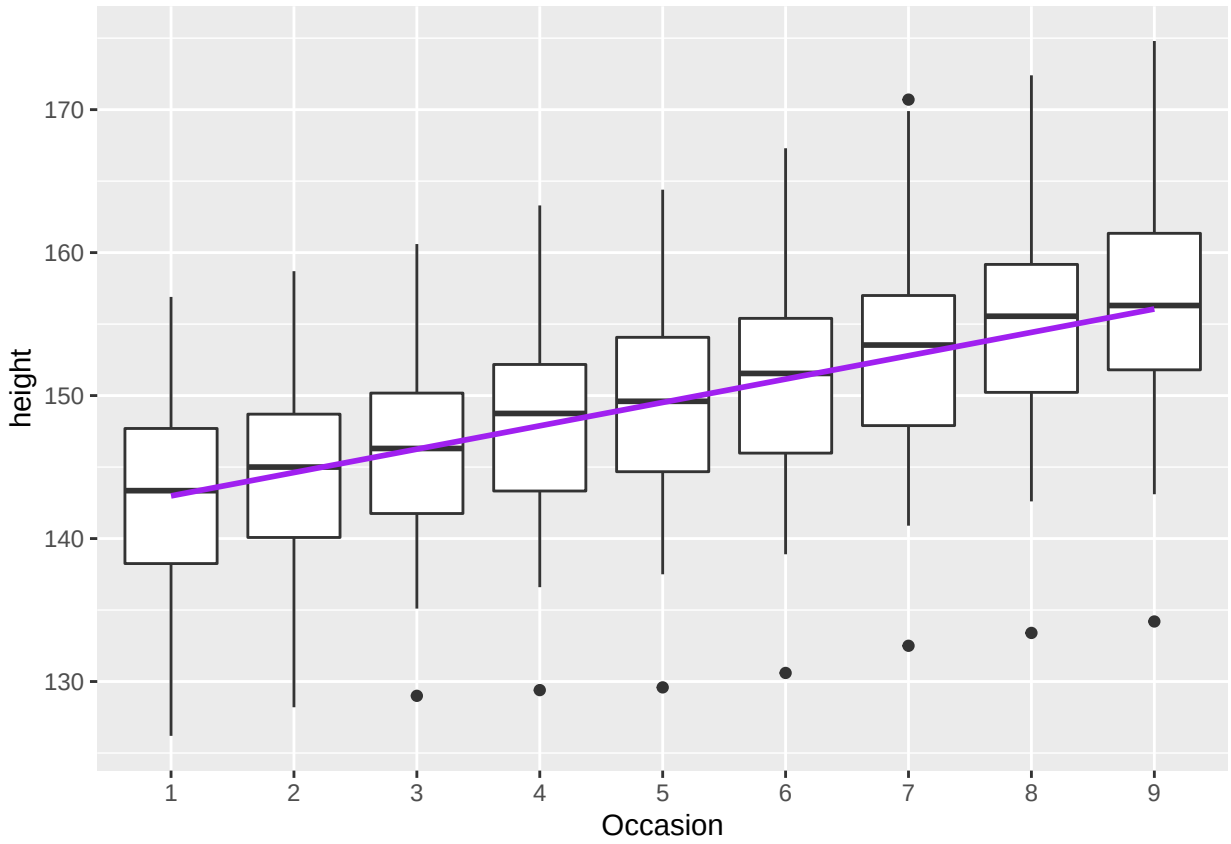



```
# We can add the a line chart again for all the subjects
h2 + geom_boxplot() + geom_line(aes(group = Subject), size=0.3, colour='purple')
```




```
# We can add the a line chart grouped
h2 + geom_boxplot() + geom_smooth(aes(group = 1), method = "lm", se = FALSE, colour='purple')
```

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




```
# Creating a visual analytical story
names(gapminder)
```

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

```
head(gapminder, n=10)
```

```
## # A tibble: 10 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.
```

```
str(gapminder)
```

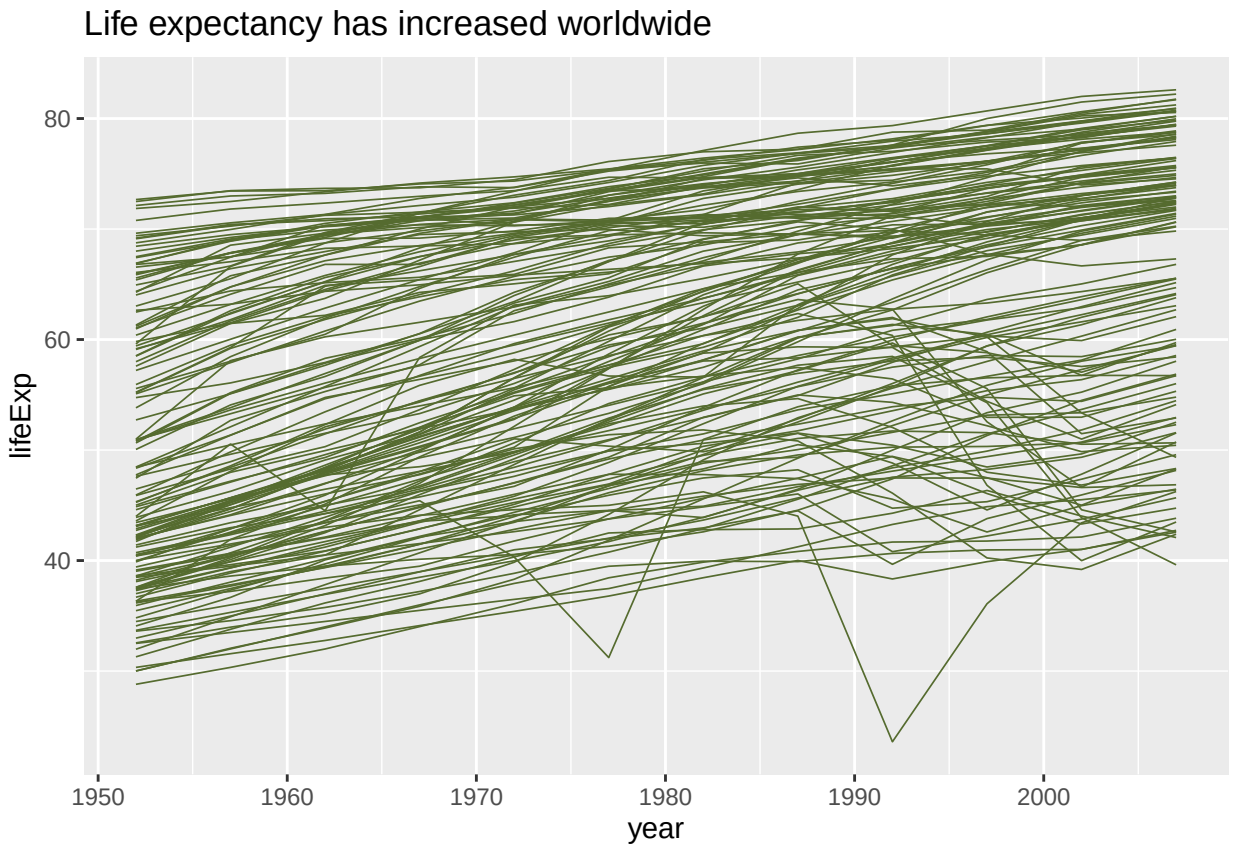
```
## tibble [1,704 x 6] (S3: tbl_df/tbl/data.frame)
##  $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ year     : int [1:1704] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...
##  $ 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 ...
```

```
summary(gapminder)
```

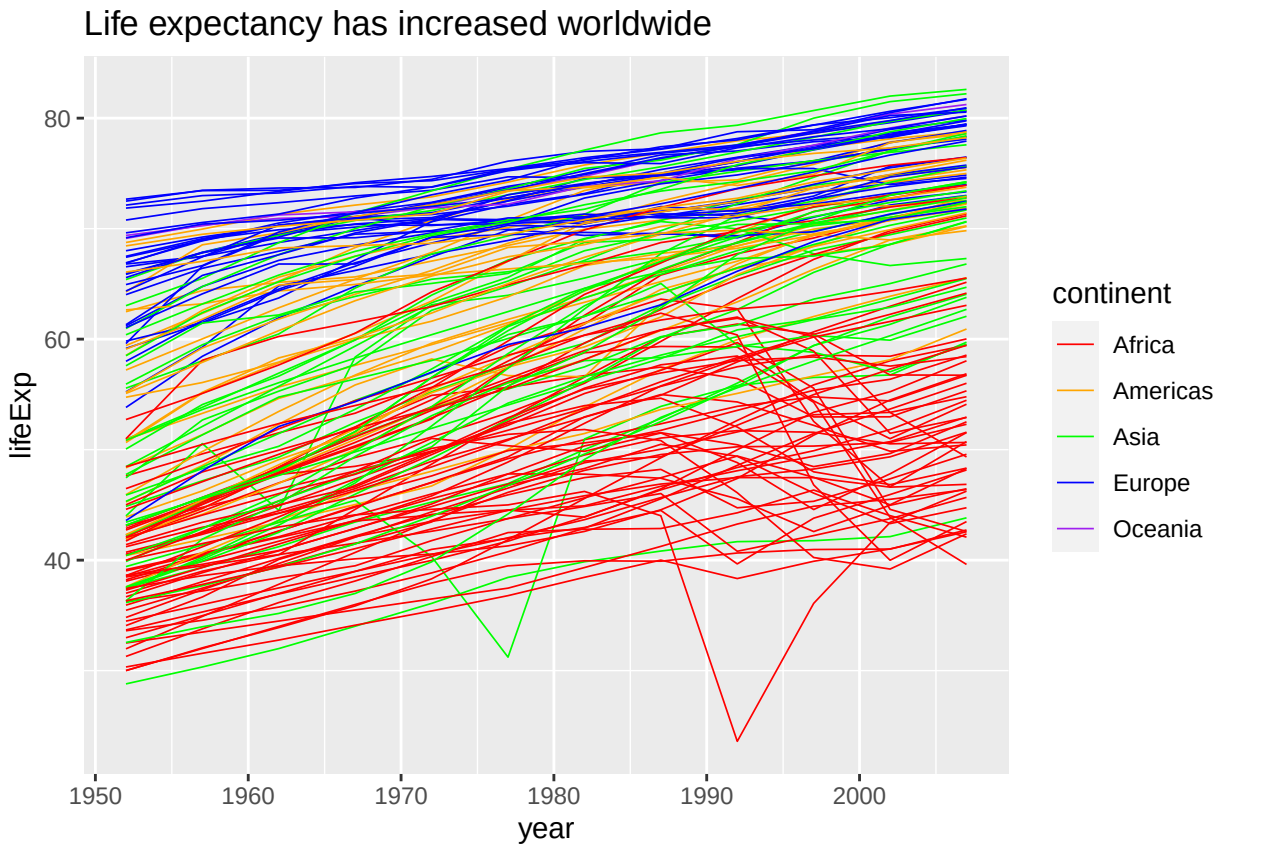
```
##         country        continent        year         lifeExp
##  Afghanistan:  12   Africa  :624   Min.   :1952   Min.   :23.60
##  Albania    :  12   Americas:300   1st Qu.:1966   1st Qu.:48.20
##  Algeria    :  12   Asia    :396   Median :1980   Median :60.71
##  Angola     :  12   Europe  :360   Mean   :1980   Mean   :59.47
##  Argentina  :  12   Oceania : 24   3rd Qu.:1993   3rd Qu.:70.85
##  Australia  :  12                  Max.   :2007   Max.   :82.60
##  (Other)    :1632
##       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
##
```

```
#General trend in life expectancy
ggplot(gapminder) +
  geom_line(aes (year, lifeExp, group = country), lwd = 0.3, show.legend = FALSE, colour = main_color) +
  labs(title = "Life expectancy has increased worldwide")
```

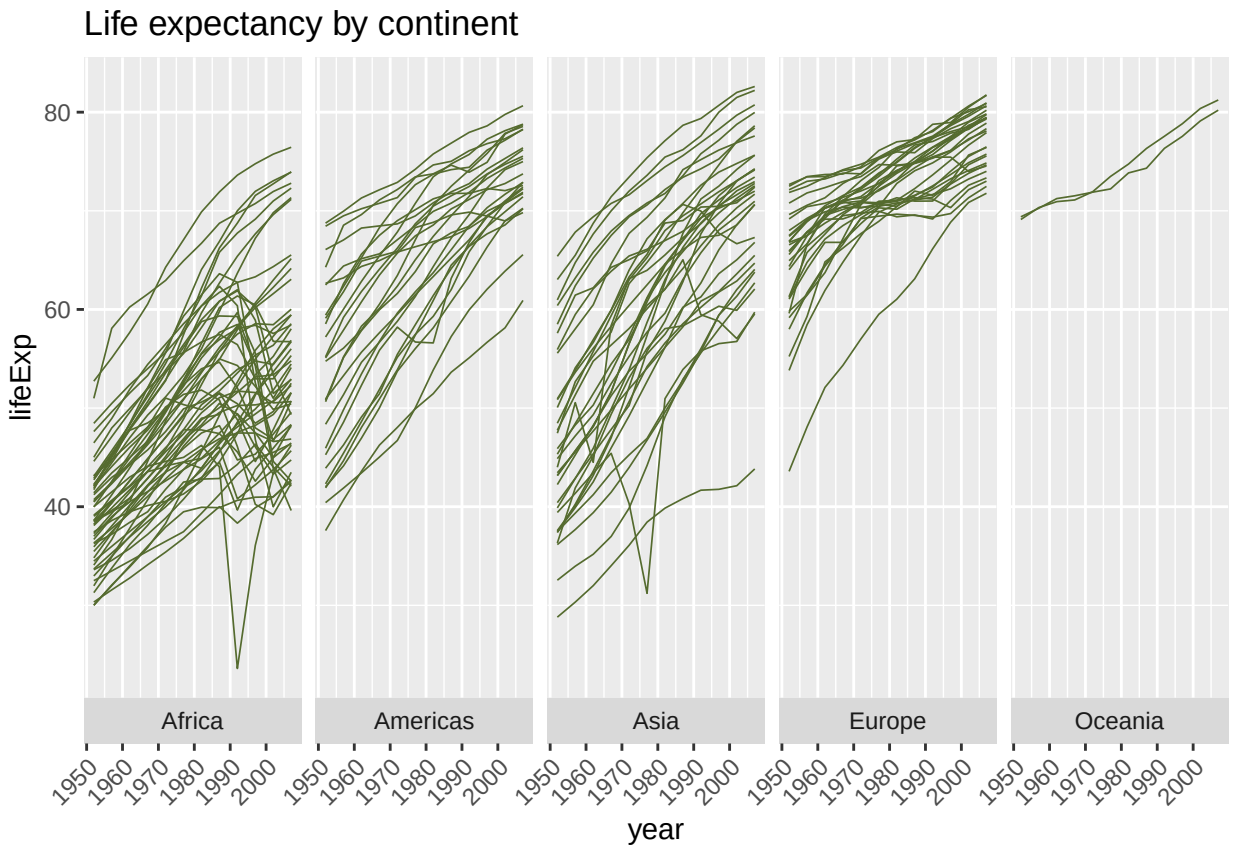



```
#Checking on continents
ggplot(gapminder) +
  geom_line(aes (year, lifeExp, group = country, color= continent), lwd = 0.3, show.legend = TRUE) +
```

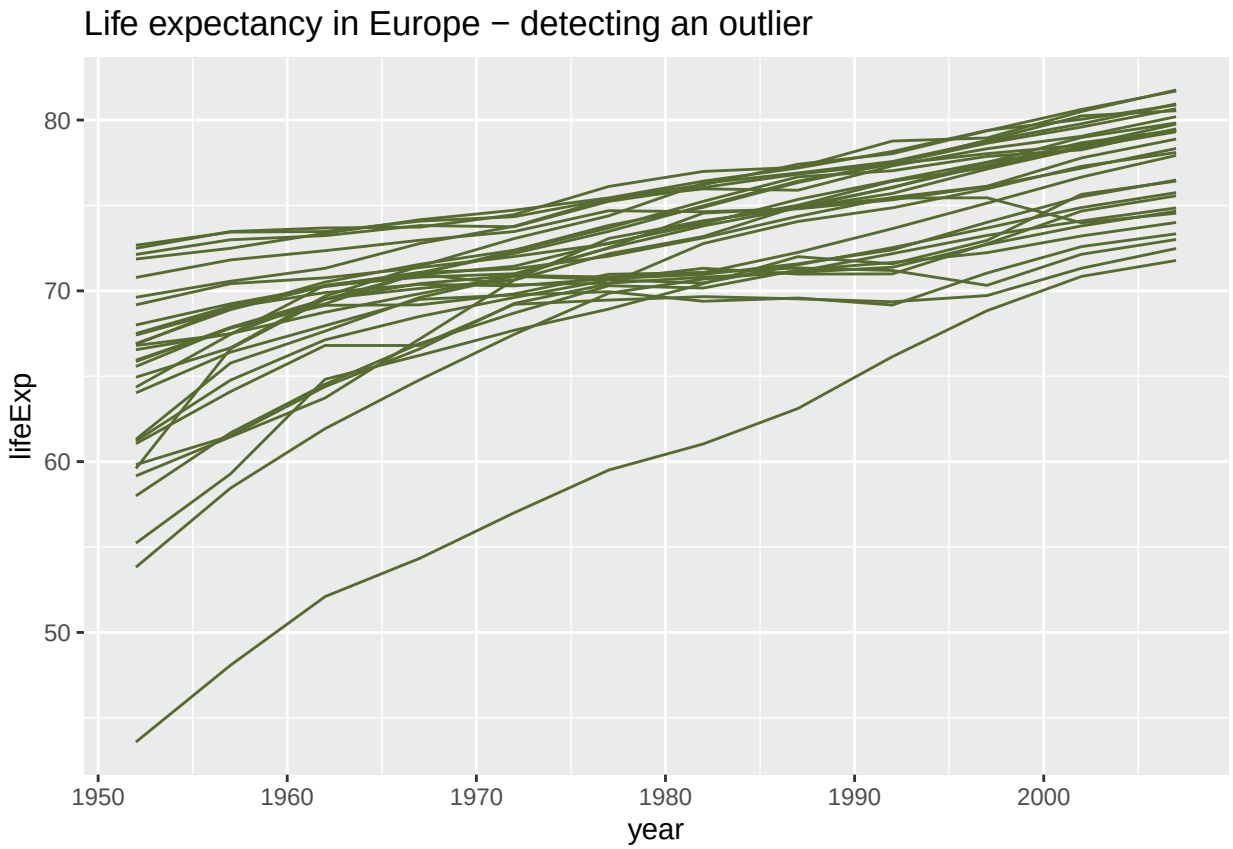
```
  scale_color_manual(values=c('red','orange','green','blue','purple')) +
  labs(title = "Life expectancy has increased worldwide")
```



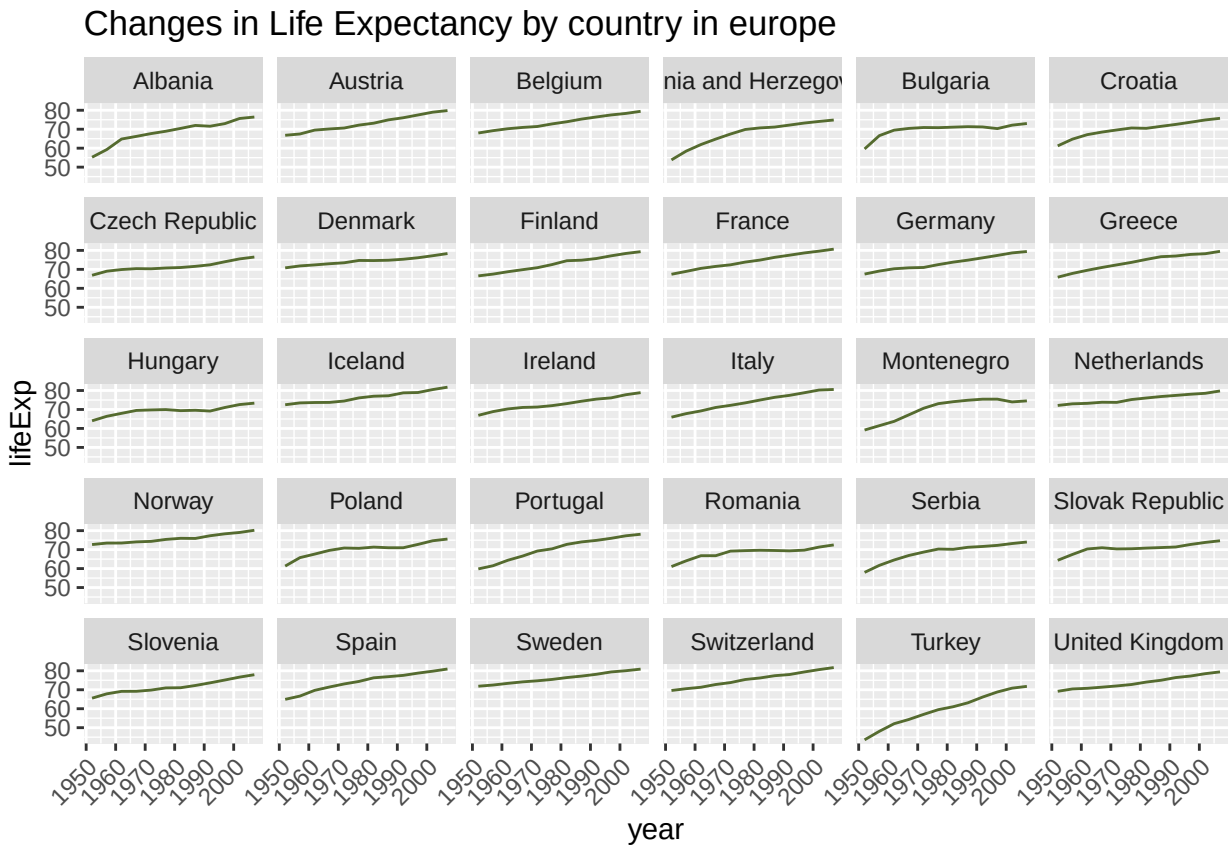

```
#Introducing a small multiple to better distinguish between continents
ggplot() +
  geom_line(data=gapminder, aes (year, lifeExp, group = country), lwd = 0.3, show.legend = FALSE, color=
  facet_wrap(~ continent, ncol=5, strip.position = "bottom") +
  scale_color_manual(values=c('red','orange','green','blue','purple')) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Life expectancy by continent")
```



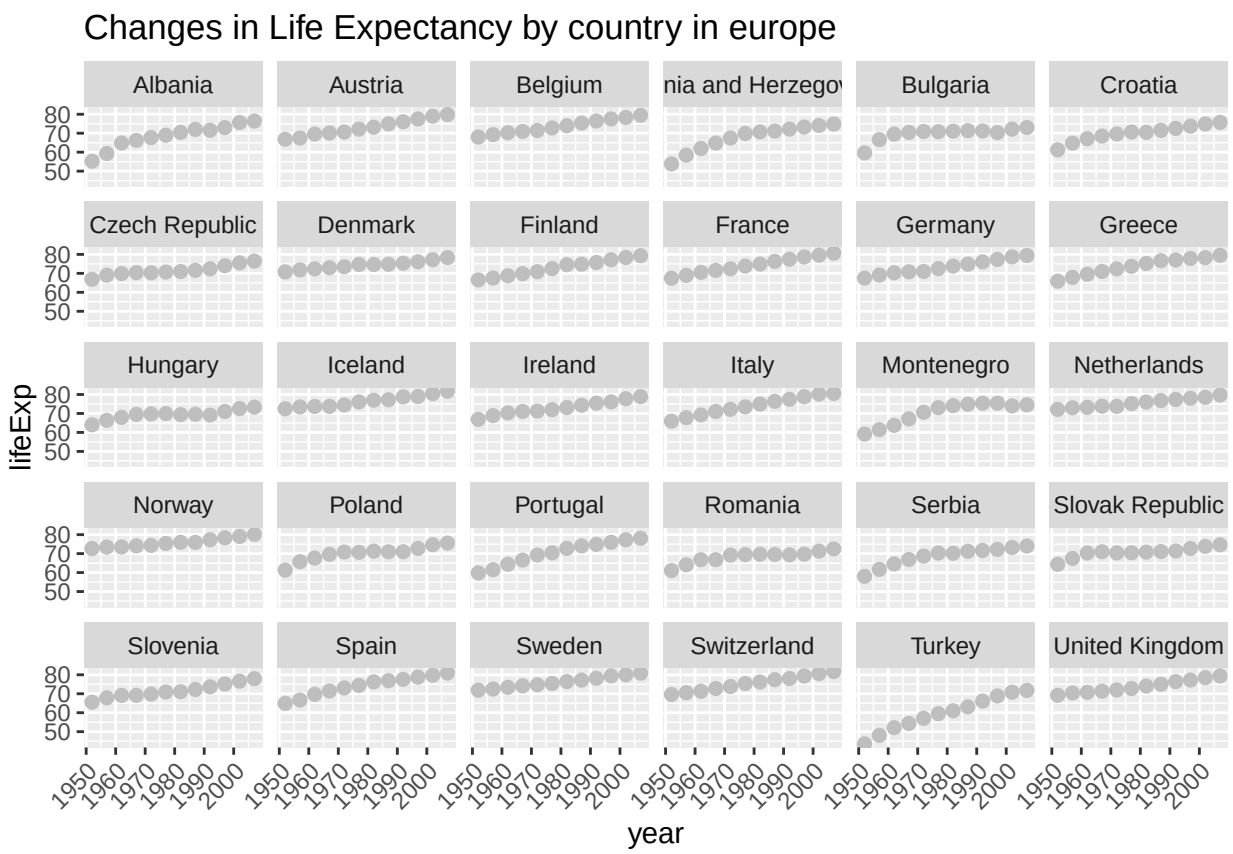

```
# Zooming in to see only Europe
ggplot(subset(gapminder, continent ==  "Europe")) +
  geom_line(aes(year, lifeExp, group = country), color= main_color, show.legend = FALSE) +
  labs(title = "Life expectancy in Europe - detecting an outlier")
```




```
# Select only Europe in order to understand which country is the outlier
europe <- dplyr::filter(gapminder, continent == "Europe")

ggplot(europe, aes(year, lifeExp)) +
  geom_line(color=main_color) +
  facet_wrap(~country) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Changes in Life Expectancy by country in europe")
```

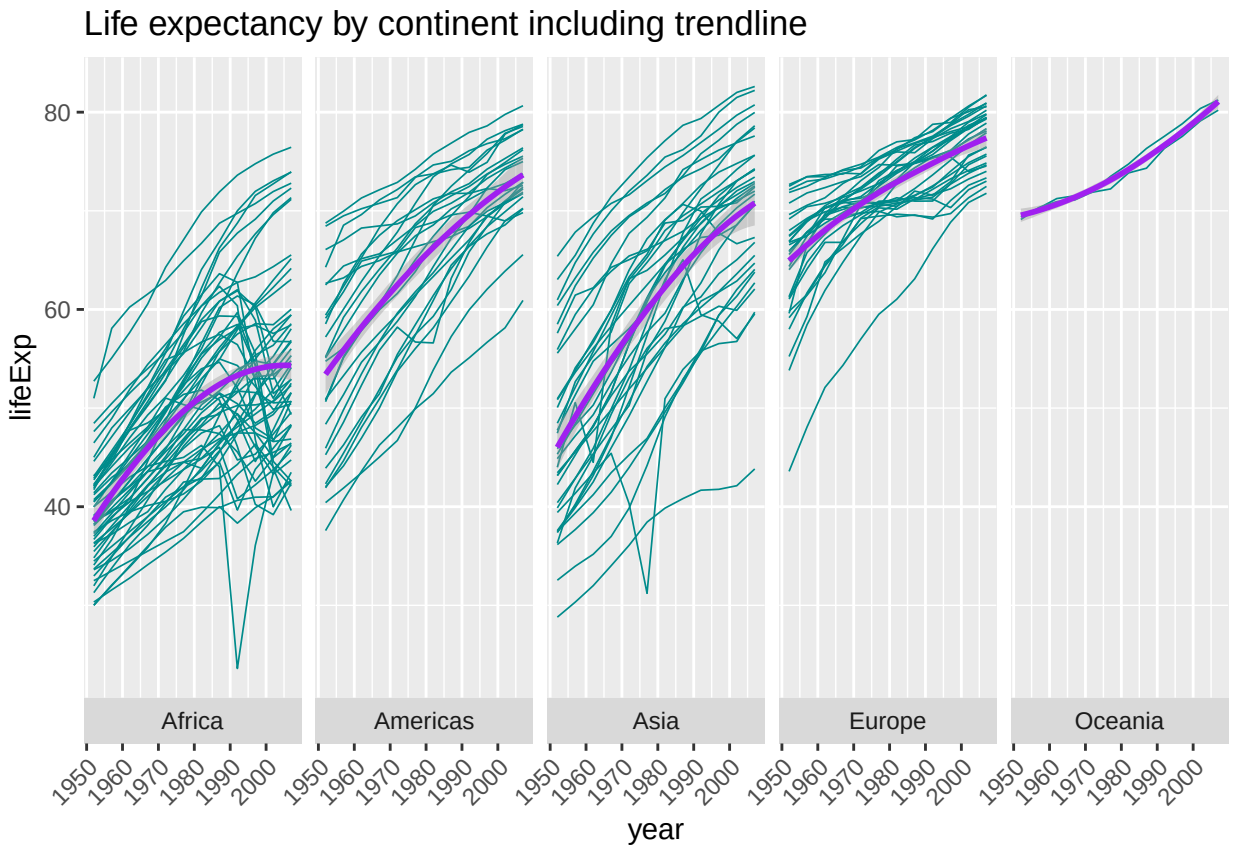



```
#We can also show the trend as dots
ggplot(europe, aes(year, lifeExp)) +
  geom_point(color="grey", size=2) +
  facet_wrap(~country) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Changes in Life Expectancy by country in europe")
```




```
#Coming back to the general checking on patterns globally

#What will be the output of this code?

#Adding a trend line - defining the method as loss
ggplot() +
  geom_line(data=gapminder, aes (year, lifeExp, group = country), lwd = 0.3, show.legend = FALSE, color=
  facet_wrap(~ continent, ncol=5, strip.position = "bottom") +
  geom_smooth(data=gapminder, aes(year, lifeExp, group = 1), lwd = 1, method = 'loess', span = 2, se = 
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Life expectancy by continent including trendline")
```

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




```
#We can even add all data in the background by setting the variable we do the facet with to zero
ggplot() +
  geom_line(data = transform(gapminder, continent = NULL), aes (year, lifeExp, group = country), alpha =
  geom_line(data=gapminder, aes (year, lifeExp, group = country), lwd = 0.3, show.legend = FALSE, color=
  geom_smooth(data=gapminder, aes(year, lifeExp, group = 1), lwd = 1, method = 'loess', span = 0.1, se =
  facet_wrap(~ continent, ncol=5, strip.position = "bottom") +
  theme(strip.background = element_blank(), strip.placement = "outside") +
  theme(axis.text.x = element_blank()) +
  labs(title = "Life expectancy by continent including trendline, showing all data in the back")
```

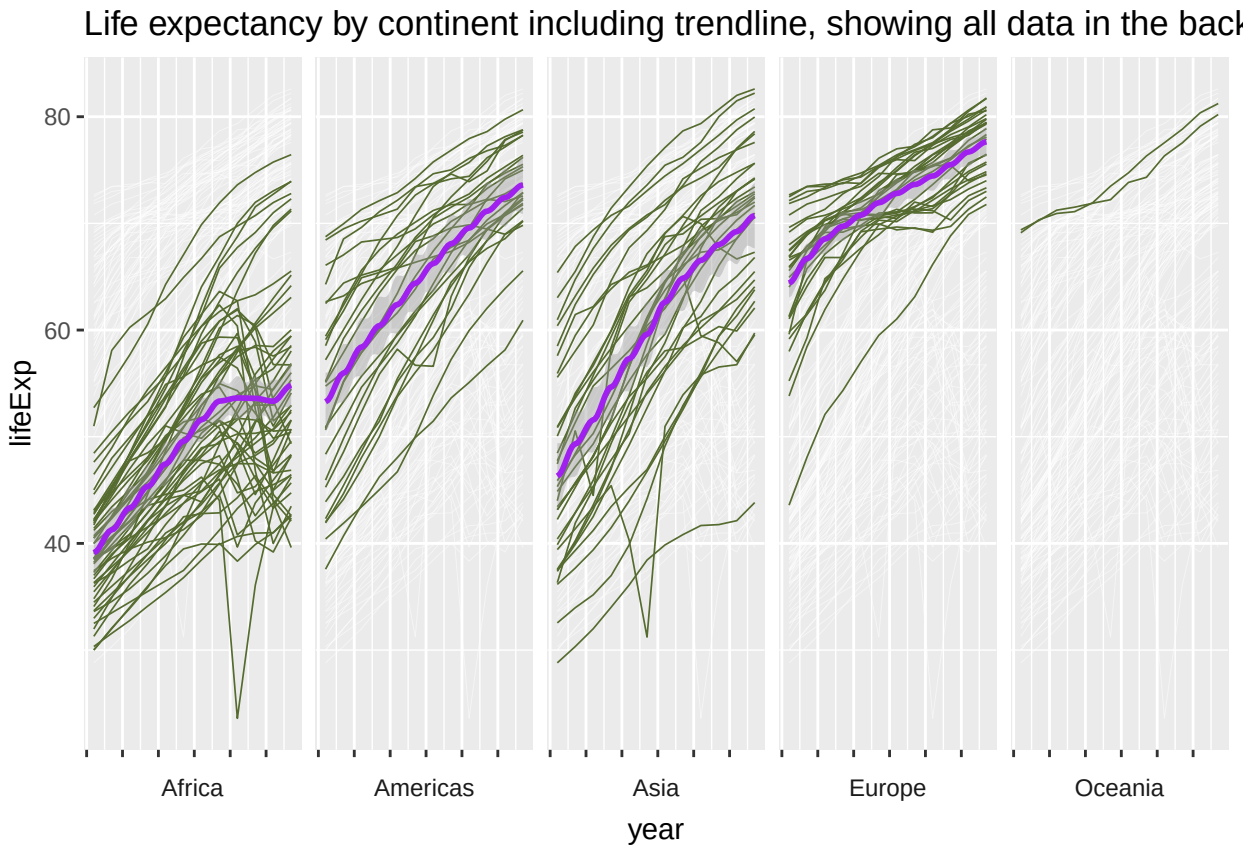
```
## `geom_smooth()` using formula 'y ~ x'

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 1951.7

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 5.275

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 27.826

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used at
## 1951.7

## Warning in predLoess(object$y, object$x, newx = if
```

```
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : neighborhood radius
## 5.275

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : reciprocal condition
## number 0

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : There are other near
## singularities as well. 27.826

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 1951.7

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 5.275

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 27.826

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used at
## 1951.7

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : neighborhood radius
## 5.275

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : reciprocal condition
## number 0

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : There are other near
## singularities as well. 27.826

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 1951.7

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 5.275

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 27.826

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used at
```

```
## 1951.7

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : neighborhood radius
## 5.275

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : reciprocal condition
## number 0

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : There are other near
## singularities as well. 27.826

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 1951.7

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 5.275

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 27.826

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used at
## 1951.7

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : neighborhood radius
## 5.275

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : reciprocal condition
## number 0

## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : There are other near
## singularities as well. 27.826

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : span too small. fewer data values than degrees of freedom.

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : at 1951.7

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : radius 0.075625

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : all data on boundary of neighborhood. make span bigger
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 1951.7

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 0.275

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 1

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : at 2007.3

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : radius 0.075625

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : all data on boundary of neighborhood. make span bigger

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 0.075625

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : zero-width neighborhood. make span bigger

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : zero-width neighborhood. make span bigger

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : zero-width neighborhood. make span bigger

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : zero-width neighborhood. make span bigger

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : zero-width neighborhood. make span bigger

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : zero-width neighborhood. make span bigger

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : zero-width neighborhood. make span bigger

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : zero-width neighborhood. make span bigger

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : zero-width neighborhood. make span bigger

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : zero-width neighborhood. make span bigger

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : zero-width neighborhood. make span bigger

## Warning: Computation failed in `stat_smooth()`:
```

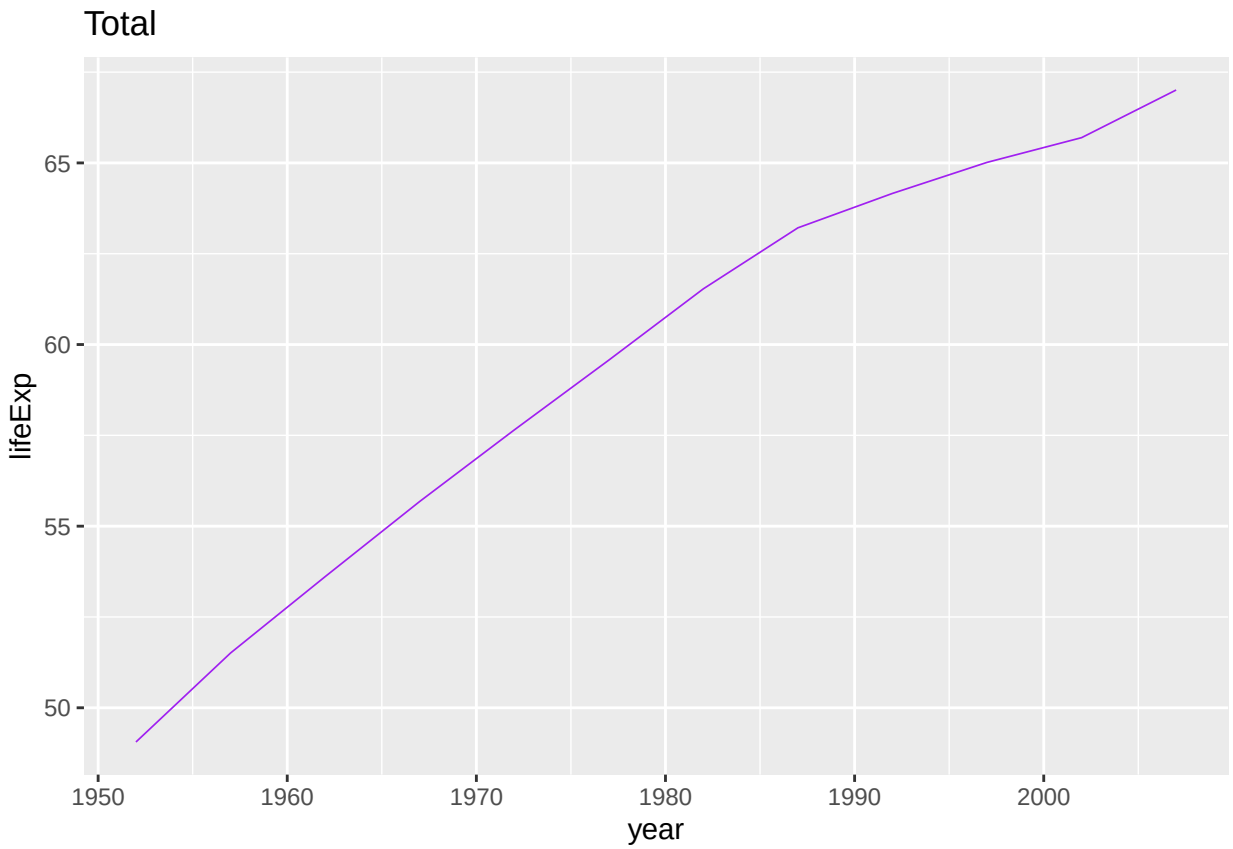
```
## NA/NaN/Inf in foreign function call (arg 5)
```



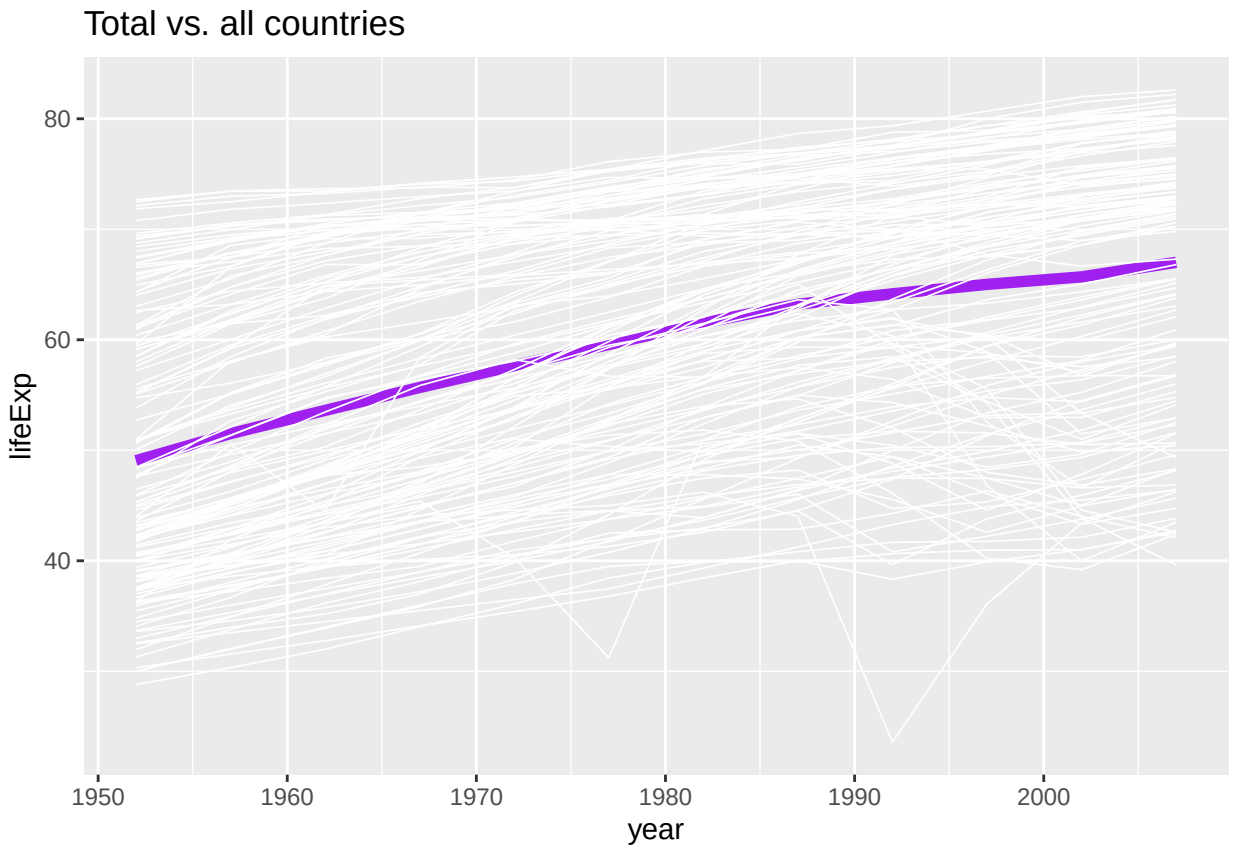

```
#Showing how to add a line by aggregating the data

#Aggregating the data
gapminderavg<-aggregate(. ~year, data=gapminder, mean, na.rm=TRUE)
head(gapminderavg, n=10)
```

```
##    year country continent  lifeExp      pop gdpPercap
## 1  1952    71.5  2.330986 49.05762 16950402  3725.276
## 2  1957    71.5  2.330986 51.50740 18763413  4299.408
## 3  1962    71.5  2.330986 53.60925 20421007  4725.812
## 4  1967    71.5  2.330986 55.67829 22658298  5483.653
## 5  1972    71.5  2.330986 57.64739 25189980  6770.083
## 6  1977    71.5  2.330986 59.57016 27676379  7313.166
## 7  1982    71.5  2.330986 61.53320 30207302  7518.902
## 8  1987    71.5  2.330986 63.21261 33038573  7900.920
## 9  1992    71.5  2.330986 64.16034 35990917  8158.609
## 10 1997    71.5  2.330986 65.01468 38839468  9090.175
```

```
#Make a plot with the aggregated data
ggplot(gapminderavg) +
  geom_line(aes (year, lifeExp), lwd = 0.3, show.legend = FALSE, color = 'purple') +
  labs(title = "Total")
```




```
#Adding this line to the general plot by using twice the geom_line with different data sets
ggplot() +
  geom_line(data=gapminderavg, aes (year, lifeExp), lwd = 2, show.legend = FALSE, color = 'purple') +
  geom_line(data=gapminder, aes (year, lifeExp, group = country), lwd = 0.3, show.legend = FALSE, color
  labs(title = "Total vs. all countries")
```



Excercise - 12

```
# parallel coordinate with gapminder data

#Parallel coordinates
?ggparcoord

#Check the data
names(gapminder)
```

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

```
head(gapminder, n=10)
```

```
## # A tibble: 10 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.
```

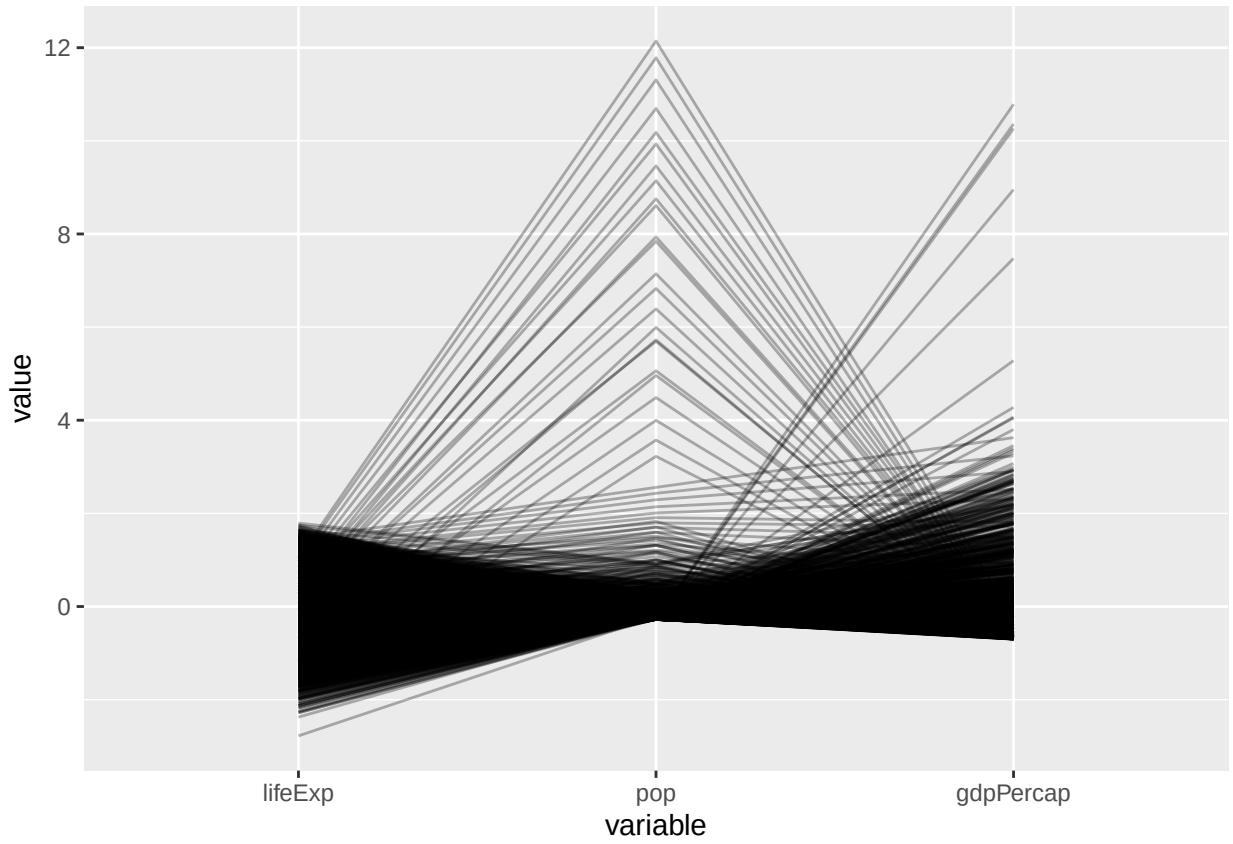
```r
str(gapminder)
```

```
## 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 1 ...
##  $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 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 ...
```

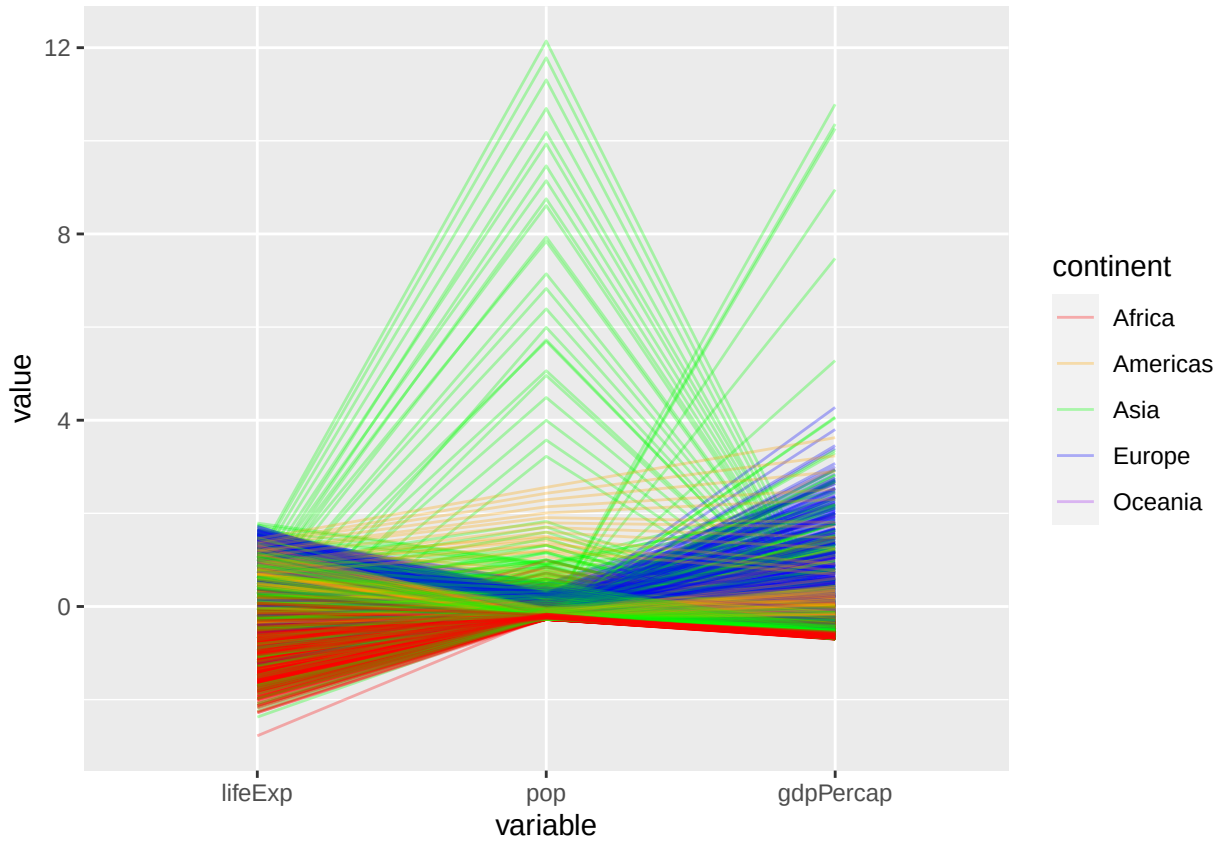
```r
summary(gapminder)
```

```
##         country        continent        year         lifeExp
##  Afghanistan:  12   Africa  :624   Min.   :1952   Min.   :23.60
##  Albania    :  12   Americas:300   1st Qu.:1966   1st Qu.:48.20
##  Algeria    :  12   Asia    :396   Median :1980   Median :60.71
##  Angola     :  12   Europe  :360   Mean   :1980   Mean   :59.47
##  Argentina  :  12   Oceania : 24   3rd Qu.:1993   3rd Qu.:70.85
##  Australia  :  12                  Max.   :2007   Max.   :82.60
##  (Other)    :1632
##       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
##
```

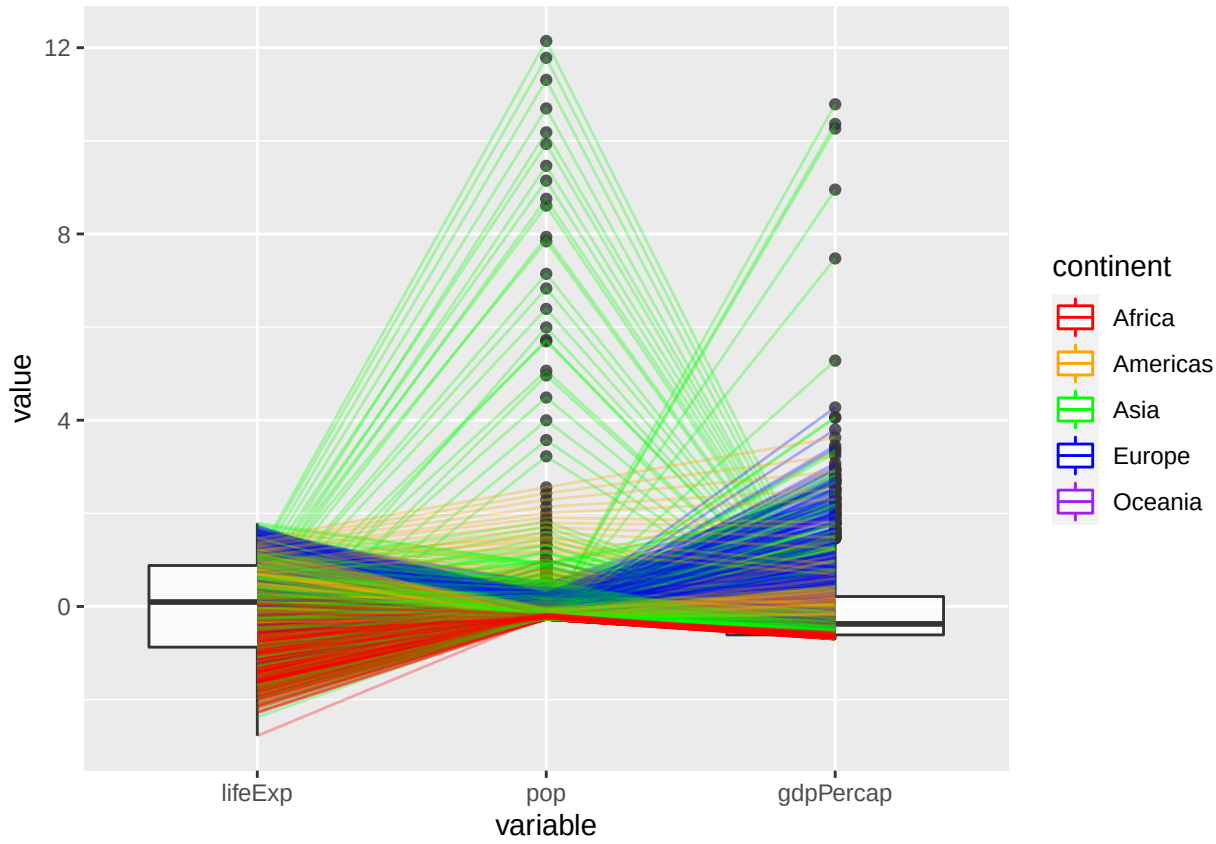
```r
#Simple chart
ggparcoord(gapminder, columns = 4:6, alphaLines = 0.3)
```



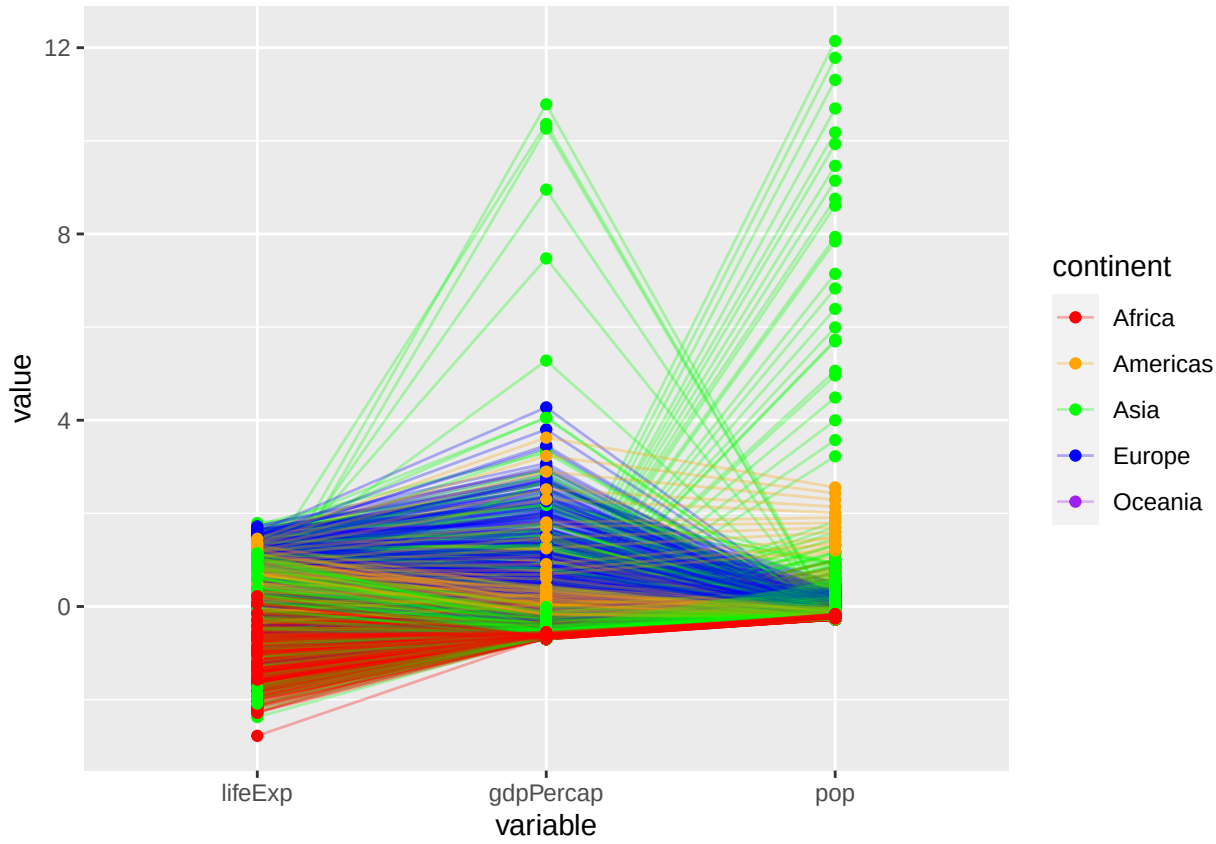

```
#Simple chart, adding a color code
ggparcoord(gapminder, columns = 4:6, groupColumn = 'continent', alphaLines = 0.3) +
  scale_color_manual(values=c('red','orange','green','blue','purple'))
```



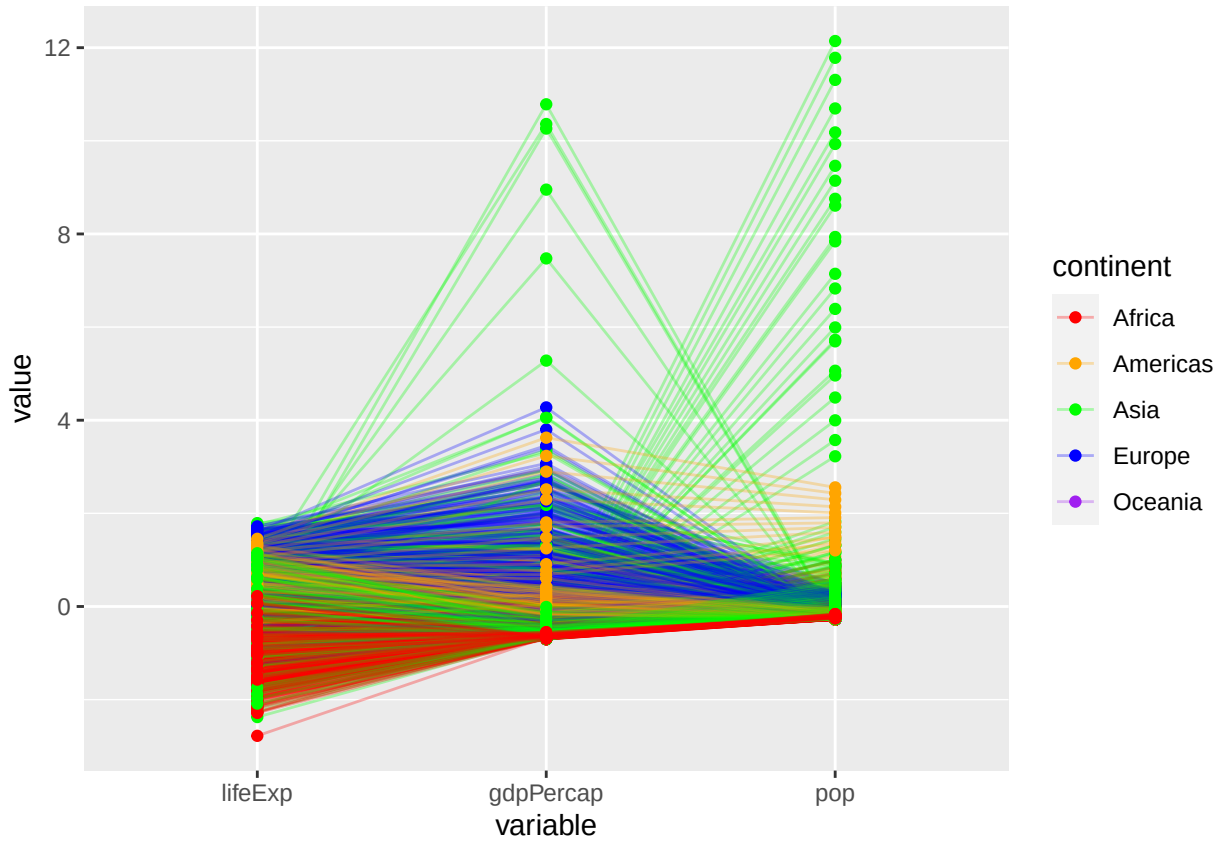

```
#Simple chart, adding a color code
ggparcoord(gapminder, columns = 4:6, groupColumn = 'continent', alphaLines = 0.3, boxplot = TRUE) +
  scale_color_manual(values=c('red','orange','green','blue','purple'))
```



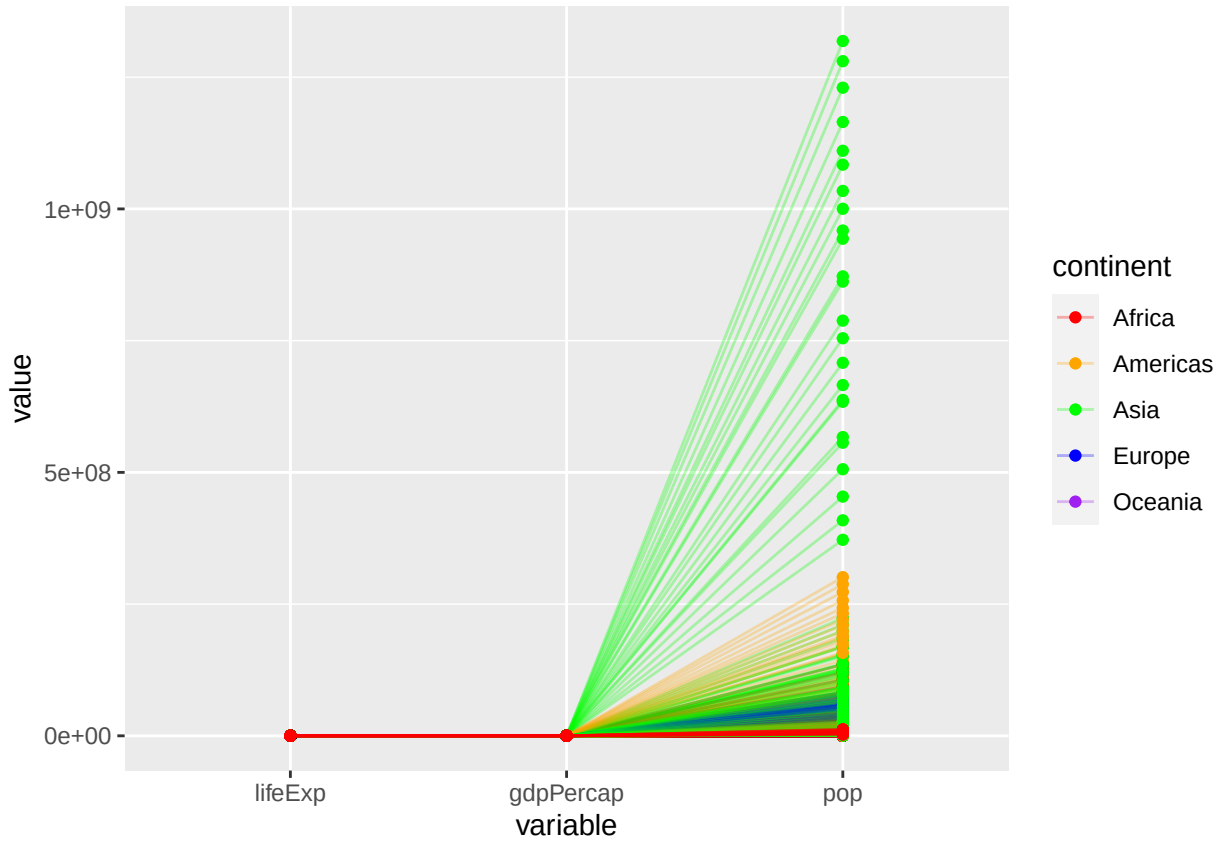

```
#Showing points, changing transparency and color
ggparcoord(gapminder, columns = 4:6, groupColumn = 'continent', order = "anyClass",
           showPoints = TRUE, alphaLines = 0.3)+
  scale_color_manual(values=c('red','orange','green','blue','purple'))
```



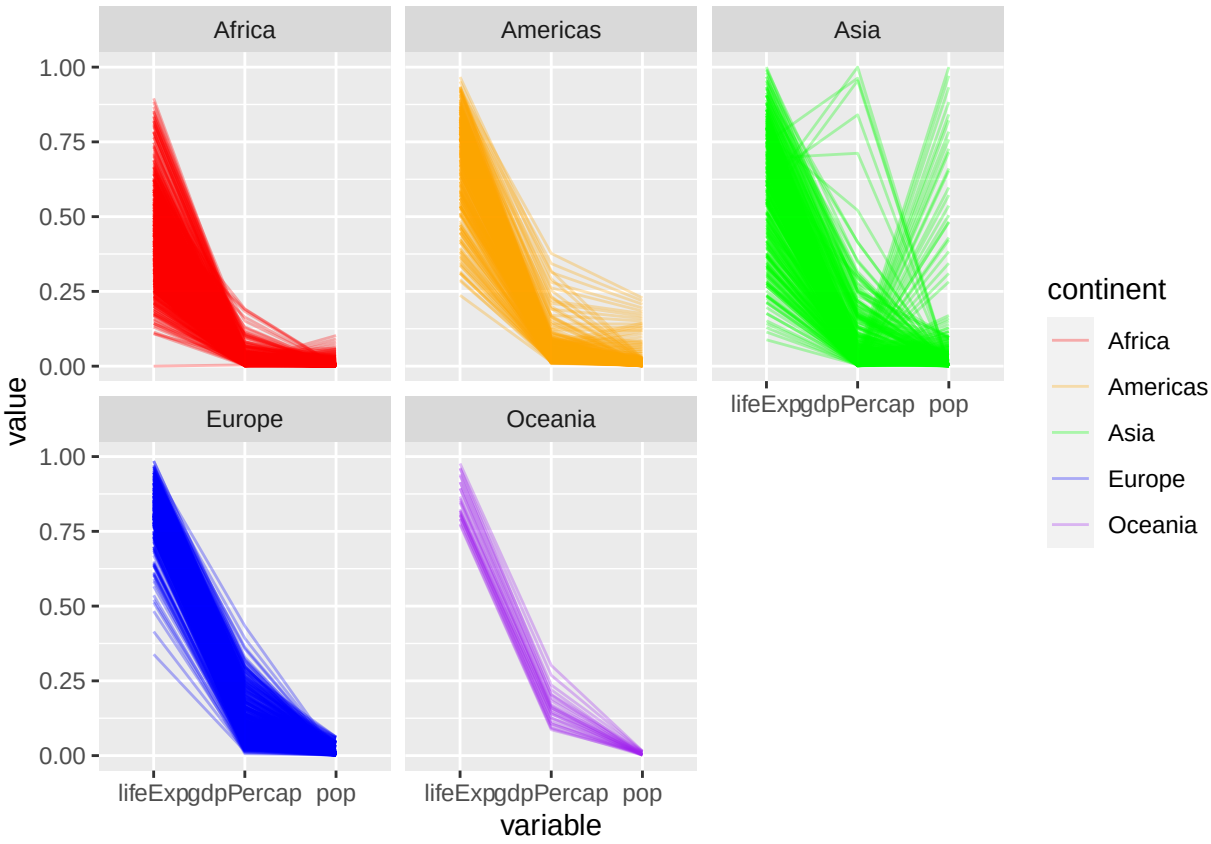

```
#Showing points, changing transparency and color
ggparcoord(gapminder, columns = 4:6, groupColumn = 'continent', order = "anyClass",
           showPoints = TRUE, alphaLines = 0.3)+
  scale_color_manual(values=c('red','orange','green','blue','purple'))
```



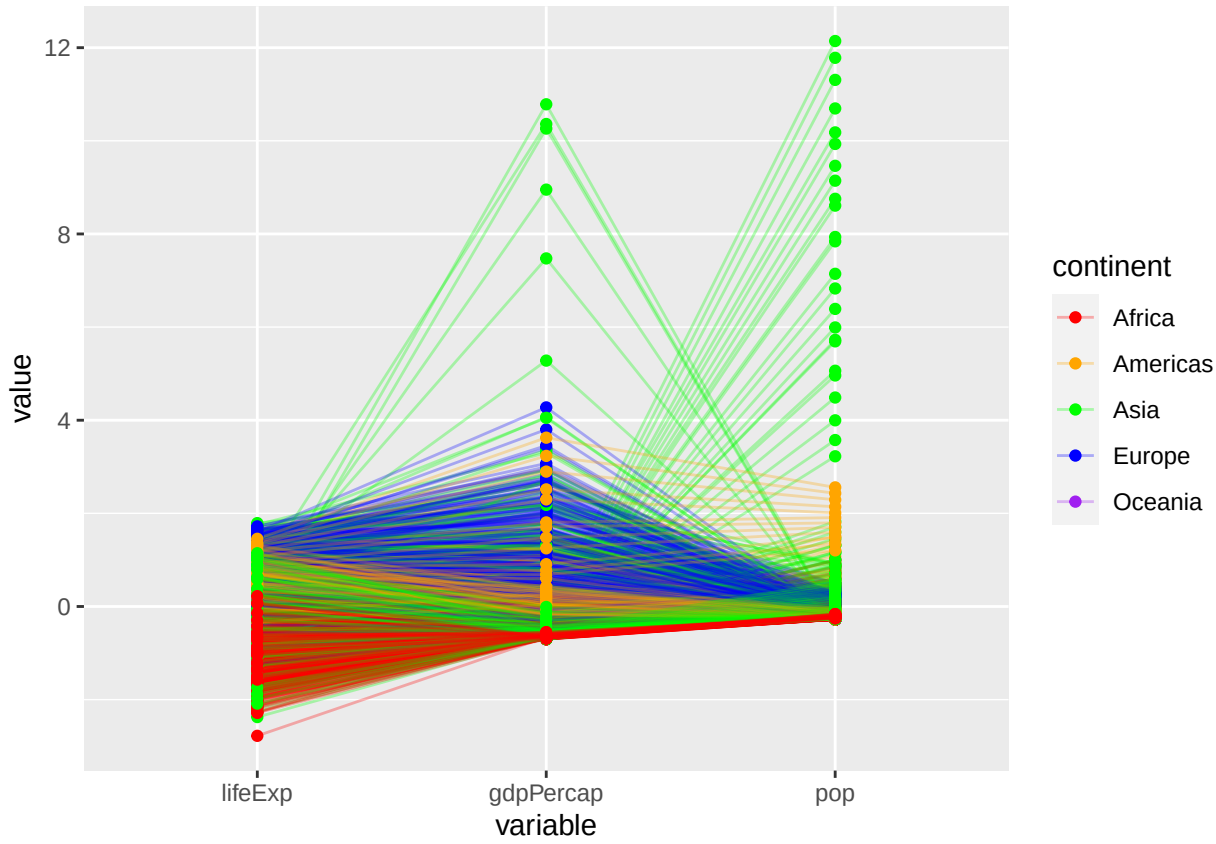

```
#Different way of scaling: No scaling
ggparcoord(gapminder, columns = 4:6, groupColumn = 'continent', order = "anyClass",
           showPoints = TRUE, alphaLines = 0.3, scale="globalminmax") +
  scale_color_manual(values=c('red','orange','green','blue','purple'))
```



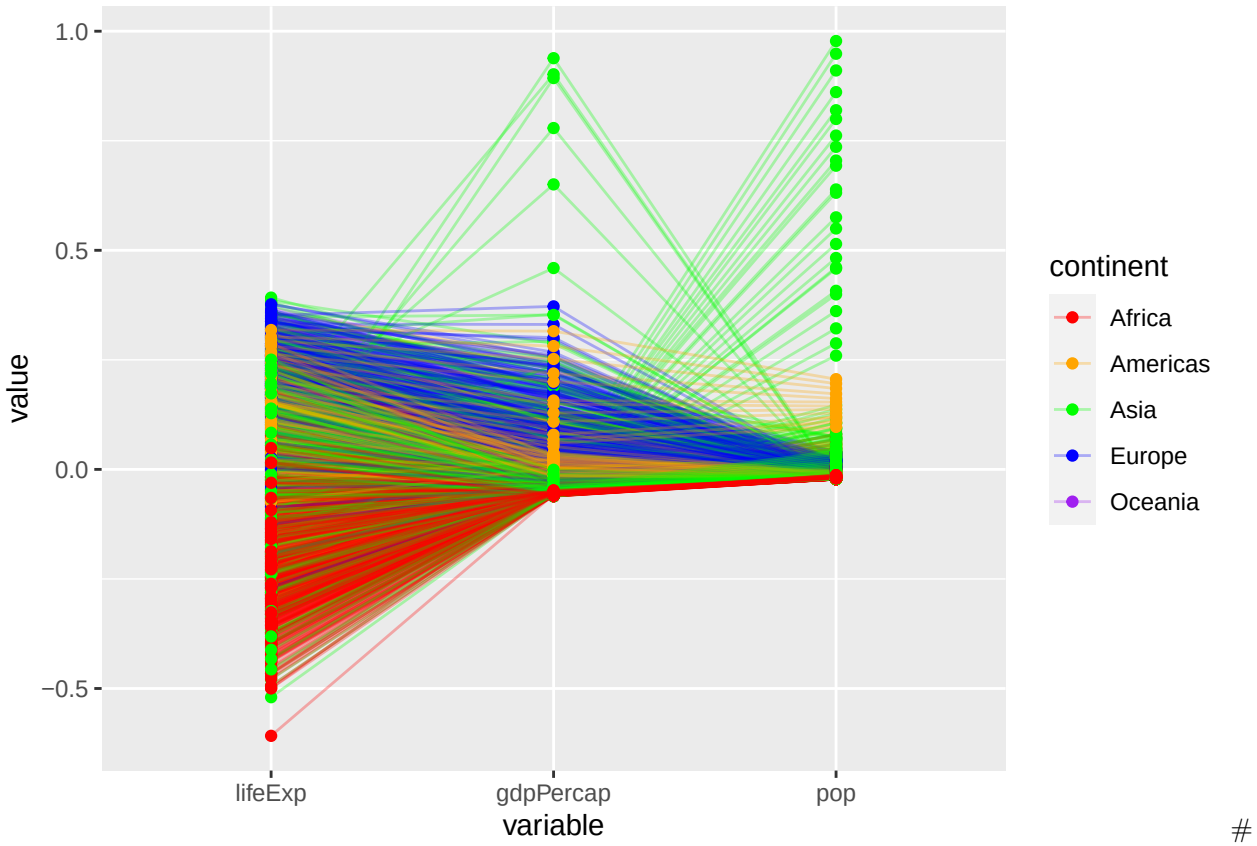

```
#Different way of scaling: Standardize to Min = 0 and Max = 1
ggparcoord(gapminder, columns = 4:6, groupColumn = 'continent', order = "anyClass",
           alphaLines = 0.3, scale="uniminmax") +
  scale_color_manual(values=c('red','orange','green','blue','purple'))+
  facet_wrap(. ~ continent)
```




```
#Different way of scaling: Normalize univariately (substract mean & divide by sd)
ggparcoord(gapminder, columns = 4:6, groupColumn = 'continent', order = "anyClass",
           showPoints = TRUE, alphaLines = 0.3, scale="std") +
  scale_color_manual(values=c('red','orange','green','blue','purple'))
```




```
#Different way of scaling: Standardize and center variables
ggparcoord(gapminder, columns = 4:6, groupColumn = 'continent', order = "anyClass",
           showPoints = TRUE, alphaLines = 0.3, scale="center")+
  scale_color_manual(values=c('red','orange','green','blue','purple'))
```



#

Excercise - 13

#Creating a subsample years<- filter(gapminder, year %in% c(1952, 2007)) %>% select(country, continent, year, lifeExp) #Check the data names(years) head(years, n=10) str(years) summary(years)

#Convert data to wide format years2 <- spread(years, year, lifeExp) names(years2) <- c("country", "continent", "y1952", "y2007")

#Create a simple dumbbell plot ggplot(years2, aes(country, x=y1952, xend=y2007))+ geom_dumbbell(colour=main_color, size_x=0, size_xend=0)

sorted by 2007

years3 <- arrange(years2, desc(y2007)) years3$country <- factor(years3country, level=rev(years3$country))

#Create a simple dumbbell plot ggplot(years3, aes(country, x=y1952, xend=y2007))+ geom_dumbbell(colour=main_color, size_x=0, size_xend=0)

#adding color for the continent ggplot(years3, aes(country, x=y1952, xend=y2007, color=continent))+ scale_color_manual(values=c('red','orange','green','blue','purple'))+ geom_dumbbell(size_x=0, size_xend=0, dot_guide=FALSE, dot_guide_size=0.2, dot_guide_colour=decoration_color)

to declutter small multiple

ggplot(years3, aes(country, x=y1952, xend=y2007, color=continent))+ scale_color_manual(values=c('red','orange','green','blue geom_dumbbell(size_x=0, size_xend=0, dot_guide=FALSE, dot_guide_size=0.2, dot_guide_colour=decoration_color) + facet_wrap(. ~ continent, ncol=5)

#Creating a subsample europe2 <- filter(gapminder, continent == "Asia" & year %in% c(1952, 2007)) %>% select(country, year, lifeExp)

#Checking head(europe2, n=10)

#Convert data to wide format europe3 <- spread(europe2, year, lifeExp) names(europe3) <- c("country", "y1952", "y2007")

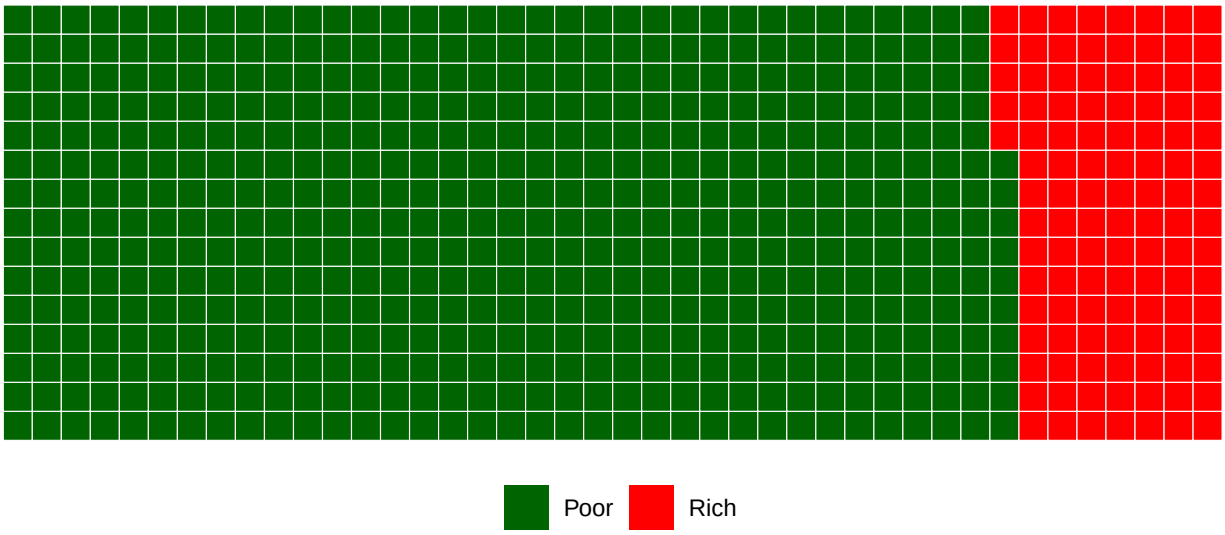
#Checking head(europe3, n=10)

#Create a simple dumbbell plot ggplot(europe3, aes(country, x = y1952, xend = y2007))+ geom_dumbbell(color=main_color)

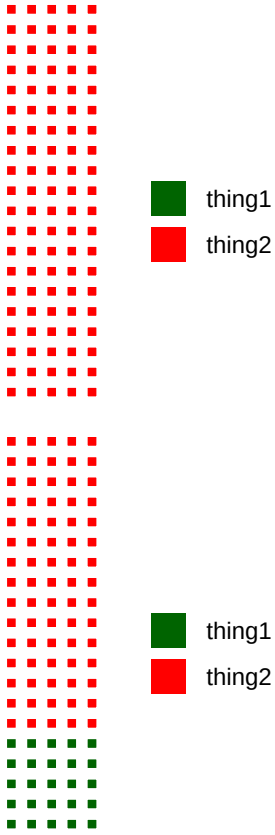
#Create a simple dumbbell plot ggplot(europe3, aes(country, x = y1952, xend = y2007)) + geom_vline(xintercept=mean(europe3$y2007), color= decoration_color, linetype = "dashed") + geom_dumbbell(color=main_color)

Excercise - 14

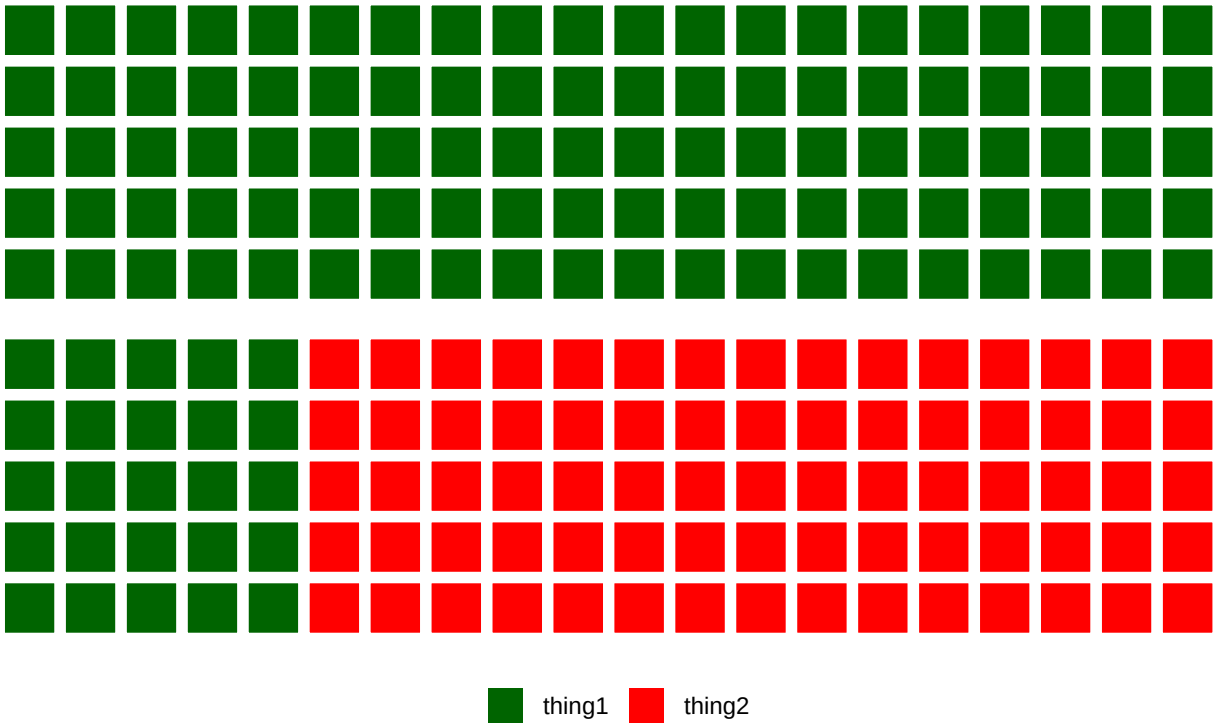
```
#Waffle chart

?waffle

#New simple dataset created
professional <- c(`Poor` = 520, `Rich` = 110)

#A simple waffle
waffle(
  professional, rows = 15, size = 0.2,
  colors = c('darkgreen', 'red'), legend_pos = "bottom"
)
```



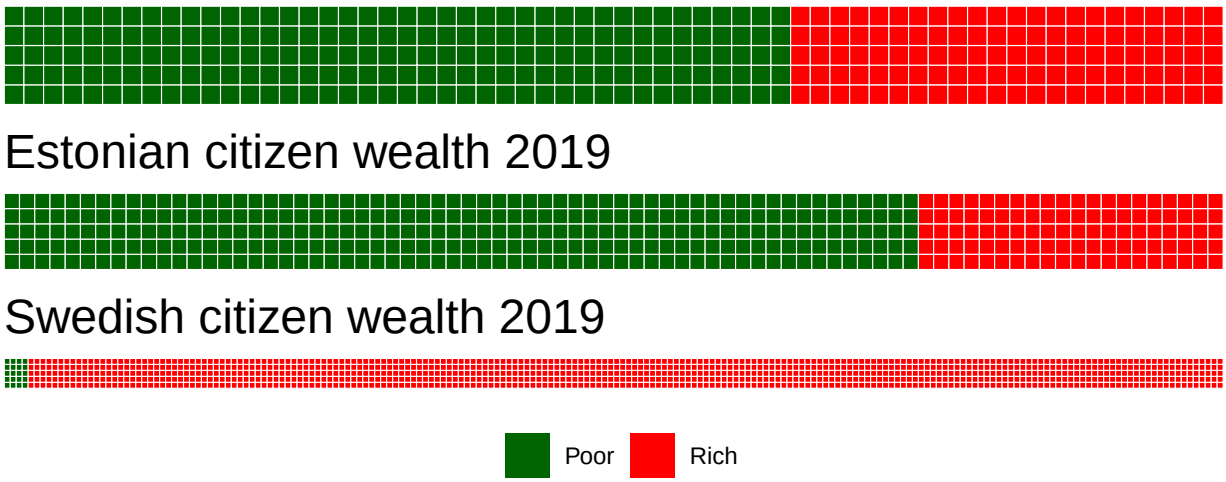

```
#You can use the iron statment to create a small multiple of waffles
iron(
  waffle(c(thing1 = 0, thing2 = 100), colors = c('darkgreen', 'red'), rows = 5, flip=TRUE),
  waffle(c(thing1 = 25, thing2 = 75), colors = c('darkgreen', 'red'), rows = 5, flip=TRUE)
)
```




```
#It's better to add the legend then separatly instead of showing it in every chart
iron(
  waffle(c(thing1 = 0, thing2 = 100), colors = c('darkgreen', 'red'), rows = 5, keep = FALSE, legend='n
  waffle(c(thing1 = 25, thing2 = 75), colors = c('darkgreen', 'red'), rows = 5, keep = FALSE, legend_pos
)
```




```
#5. Adding the legend only to one

iron(

  waffle(
    c(`Poor` = 200, `Rich` = 110), rows = 5, size = 0.2,
    colors = c('darkgreen', 'red'),
    keep = FALSE,
    title = "Finnish citizen wealth 2019",
    legend='none'),

  waffle(
    c(`Poor` = 300, `Rich` = 100), rows = 5, size = 0.2,
    colors = c('darkgreen', 'red'),
    keep = FALSE,
    title = "Estonian citizen wealth 2019",
    legend ='none'),

  waffle(
    c(`Poor` = 20, `Rich` = 1000), rows = 5, size = 0.2,
    colors = c('darkgreen', 'red'),
    keep = FALSE,
    title = "Swedish citizen wealth 2019",
    legend_pos = "bottom")

)
```



Part - 2

```
# setting the path for the folder
setwd("/Users/shardendujha/Desktop/pop")

# loading the file to R
population <- read.csv("DP_LIVE_13112020140848034.csv")

# checking the data

names(population)
```

```
## [1] "LOCATION"   "INDICATOR"  "SUBJECT"    "MEASURE"    "FREQUENCY"
## [6] "TIME"       "Value"      "Flag.Codes"
```

```
head(population, n=10)
```

```
##    LOCATION    INDICATOR   SUBJECT MEASURE FREQUENCY    TIME      Value
## 1       AUS GFCFFORECAST DOUBLEHIT  AGRWTH         Q 2020-Q4 -14.300598
## 2       AUS GFCFFORECAST SINGLEHIT  AGRWTH         Q 2020-Q4   6.752265
## 3       AUT GFCFFORECAST DOUBLEHIT  AGRWTH         Q 2020-Q4 -10.517716
## 4       AUT GFCFFORECAST SINGLEHIT  AGRWTH         Q 2020-Q4   5.718701
## 5       BEL GFCFFORECAST DOUBLEHIT  AGRWTH         Q 2020-Q4 -45.441256
## 6       BEL GFCFFORECAST SINGLEHIT  AGRWTH         Q 2020-Q4  36.440214
## 7       CAN GFCFFORECAST DOUBLEHIT  AGRWTH         Q 2020-Q4 -14.332621
## 8       CAN GFCFFORECAST SINGLEHIT  AGRWTH         Q 2020-Q4  17.728045
## 9       CZE GFCFFORECAST DOUBLEHIT  AGRWTH         Q 2020-Q4 -67.500000
## 10      CZE GFCFFORECAST SINGLEHIT  AGRWTH         Q 2020-Q4  64.785745
##    Flag.Codes
## 1          NA
## 2          NA
## 3          NA
## 4          NA
## 5          NA
## 6          NA
## 7          NA
```

```
## 8          NA
## 9          NA
## 10         NA
```

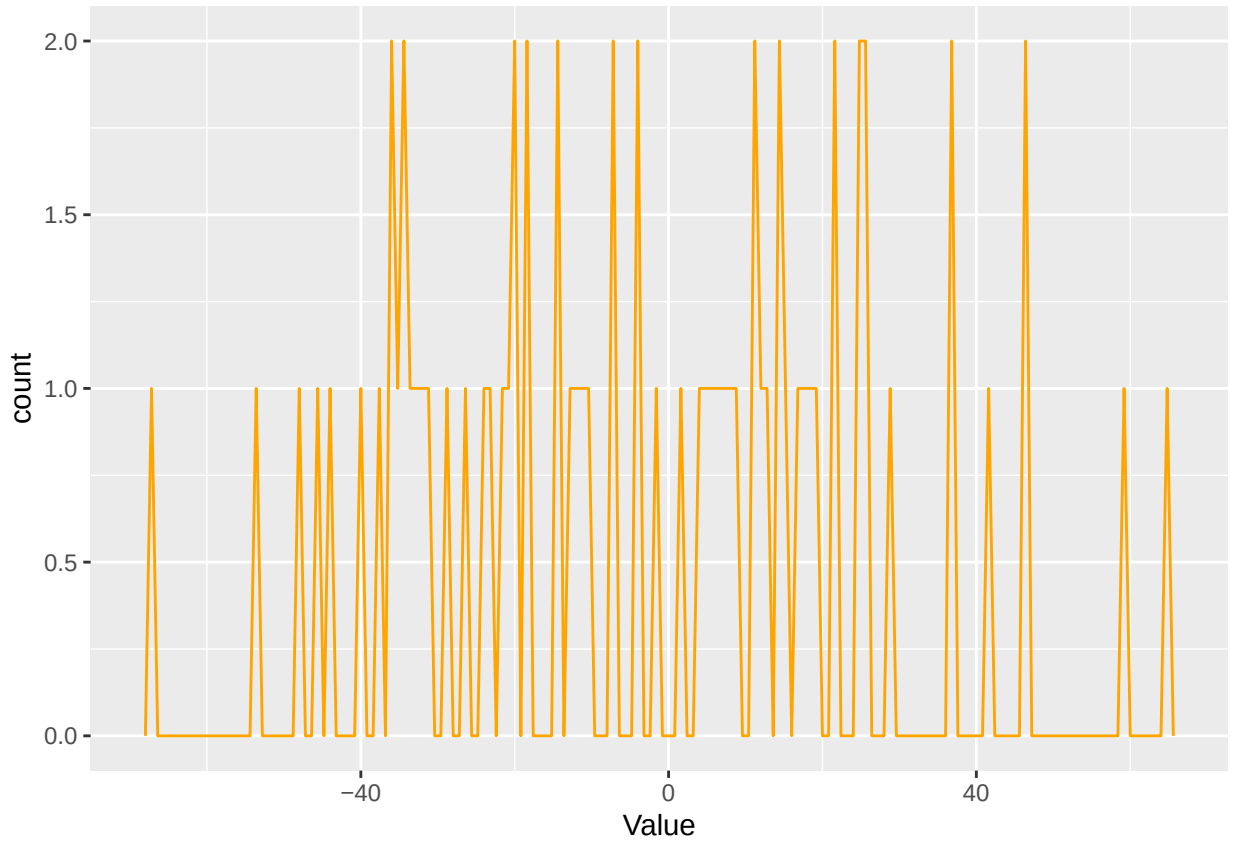
```
str(population)
```

```
## 'data.frame':    70 obs. of  8 variables:
##  $ LOCATION  : chr  "AUS" "AUS" "AUT" "AUT" ...
##  $ INDICATOR : chr  "GFCFFORECAST" "GFCFFORECAST" "GFCFFORECAST" "GFCFFORECAST" ...
##  $ SUBJECT   : chr  "DOUBLEHIT" "SINGLEHIT" "DOUBLEHIT" "SINGLEHIT" ...
##  $ MEASURE   : chr  "AGRWTH" "AGRWTH" "AGRWTH" "AGRWTH" ...
##  $ FREQUENCY : chr  "Q" "Q" "Q" "Q" ...
##  $ TIME      : chr  "2020-Q4" "2020-Q4" "2020-Q4" "2020-Q4" ...
##  $ Value     : num  -14.3 6.75 -10.52 5.72 -45.44 ...
##  $ Flag.Codes: logi  NA NA NA NA NA NA ...
```

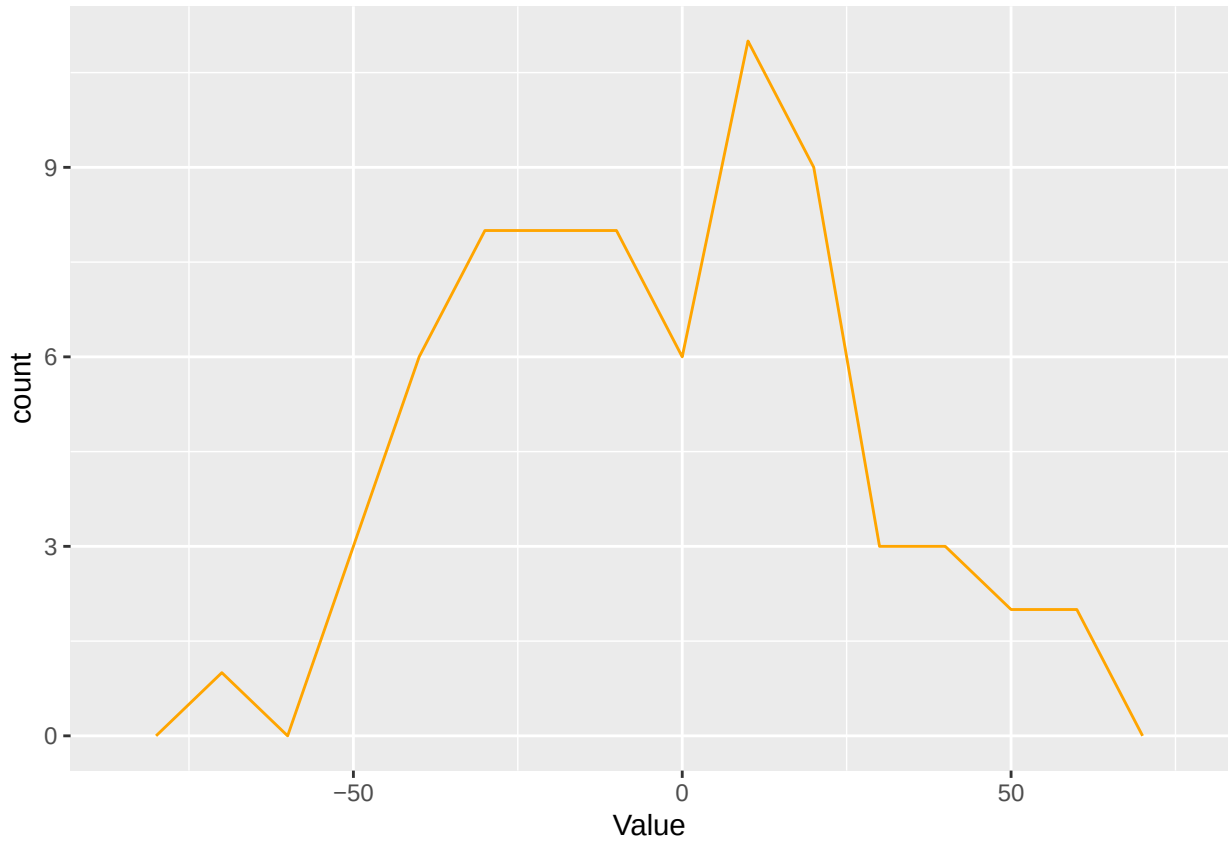
```
summary(population)
```

```
##    LOCATION          INDICATOR           SUBJECT            MEASURE
##  Length:70          Length:70          Length:70          Length:70
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##   FREQUENCY             TIME               Value          Flag.Codes
##  Length:70          Length:70          Min.   :-67.500   Mode:logical
##  Class :character   Class :character   1st Qu.:-25.617   NA's:70
##  Mode  :character   Mode  :character   Median : -3.894
##                                        Mean   : -3.535
##                                        3rd Qu.: 16.587
##                                        Max.   : 64.786
```

```
ggplot(population, aes(Value)) +
  geom_freqpoly(colour = sub_color, binwidth = 0.8)
```

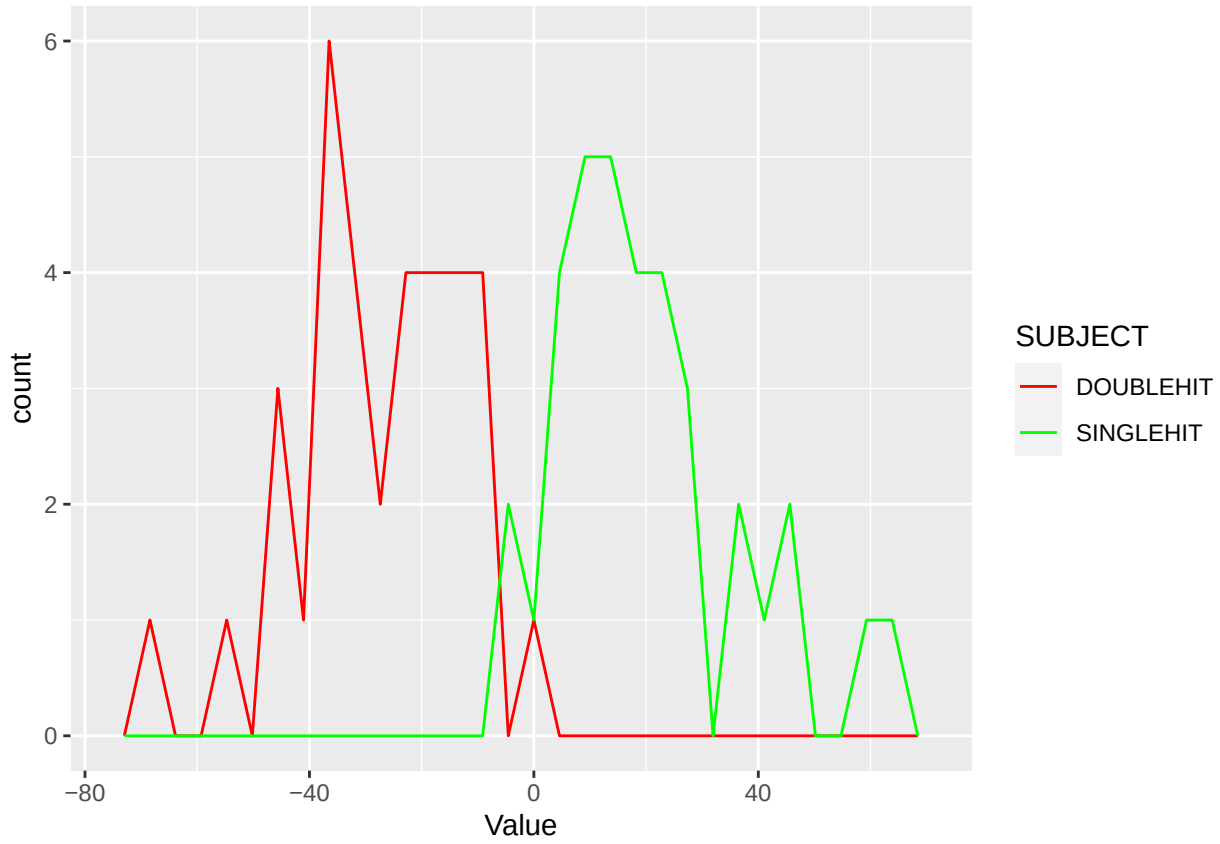



```
#Changing the bin width (less details)
ggplot(population, aes(Value)) +
  geom_freqpoly(colour = sub_color, binwidth = 10)
```



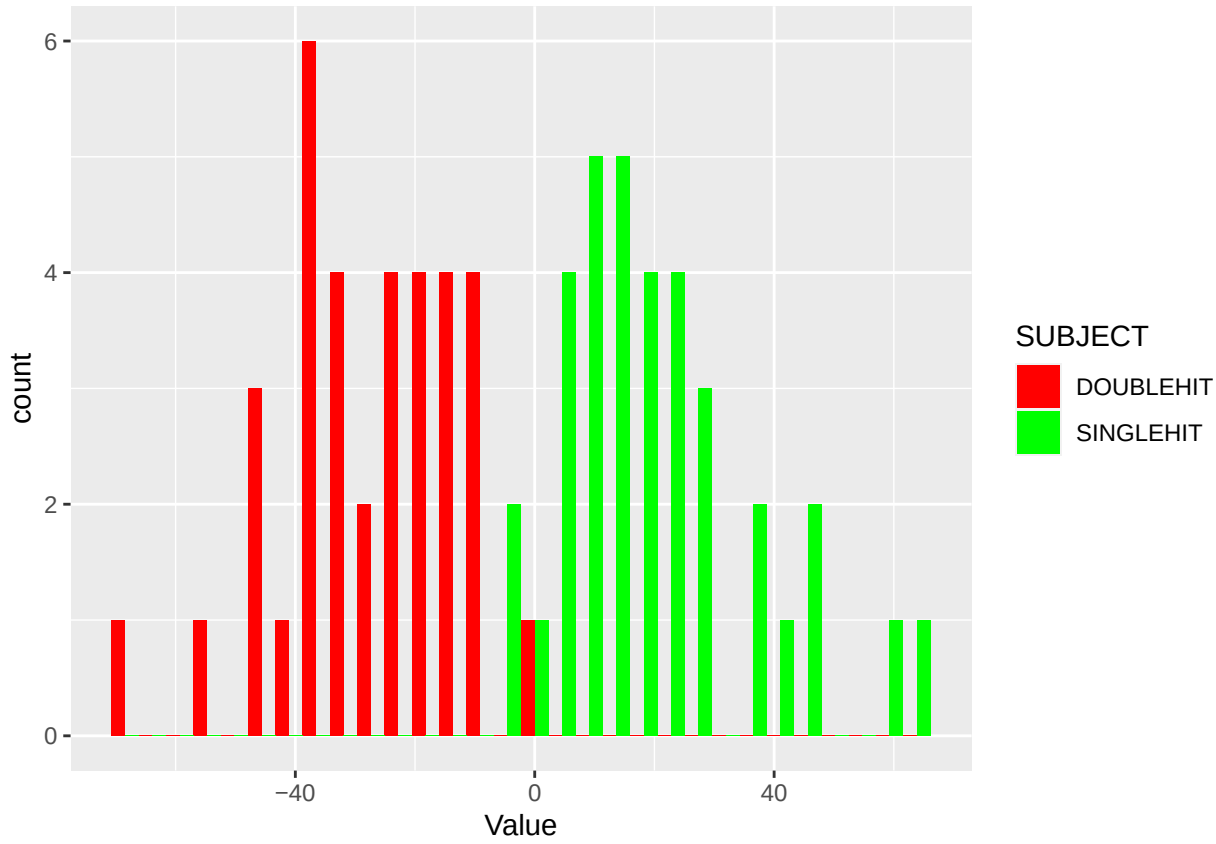

```
# display visual encoding as a color
ggplot(population, aes(Value, colour = SUBJECT)) +
  geom_freqpoly() +
  scale_color_manual(values=c('red','green'))
```

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

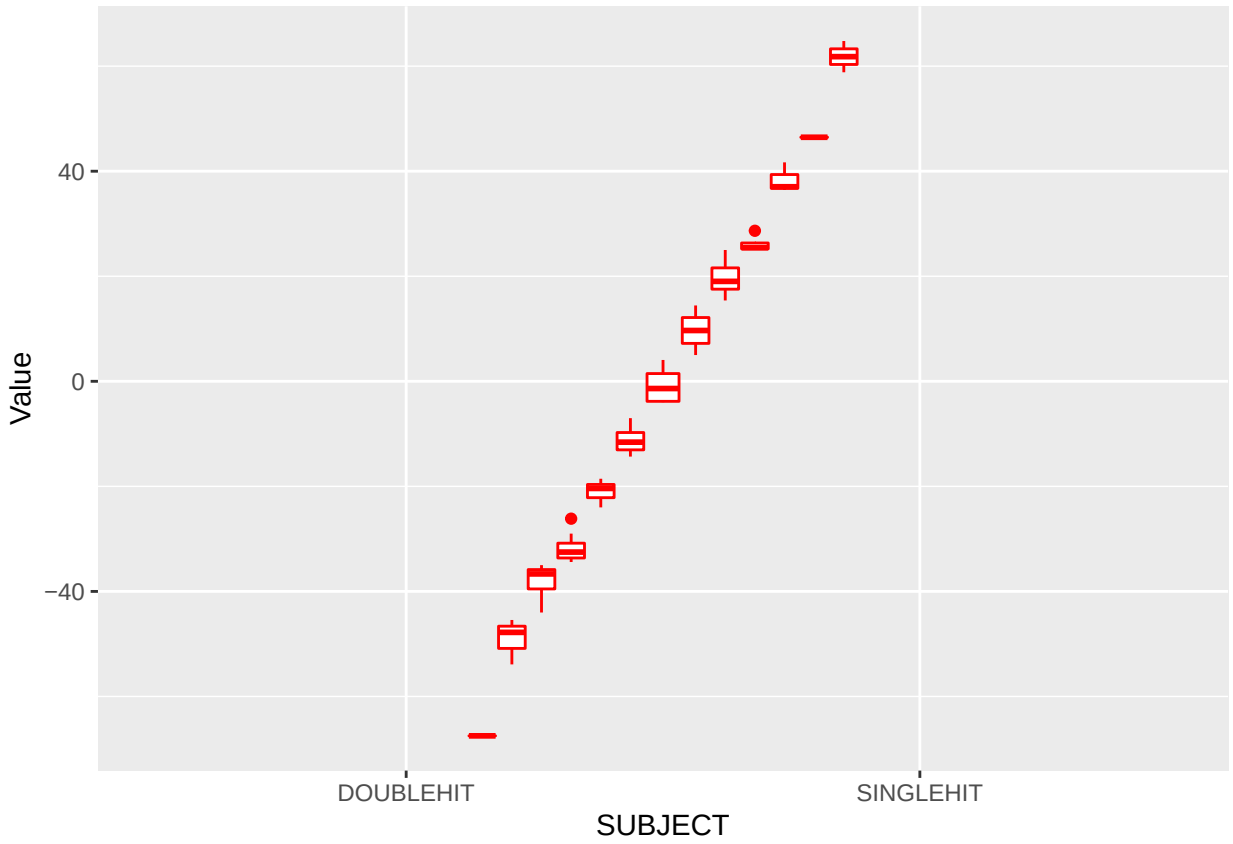



```
#Histogram for different cut options
ggplot(population, aes(Value, fill = SUBJECT)) +
  geom_histogram(position = "dodge") +
  scale_fill_manual(values=c('red','green'))
```

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




```
# Boxplot using to numeric variable
ggplot(population, aes(SUBJECT,Value)) +
  geom_boxplot(aes(group = cut_width(Value, 10)), color='red')
```


40
0
−40
Value
DOUBLEHIT
SINGLEHIT
SUBJECT