assignment 1 #2

Anh Ha

11/18/2020

```
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/macbookpro/Downloads
```

```
library(gifski)
library(forcats)
library(tufte)
library(colorspace)
library(viridisLite)
library(formatR)
library(DiagrammeR)
library(xaringan)
library(ggridges)
library(GGally)
```

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

```
library(ggplot2movies)
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
library(ggpointdensity)
library(ggstatsplot)
```

```
## Registered S3 method overwritten by 'broom.mixed':
##   method      from
##   tidy.gamlss broom
```

```
## Registered S3 methods overwritten by 'lme4':
##   method                          from
##   cooks.distance.influence.merMod car
##   influence.merMod                car
##   dfbeta.influence.merMod         car
##   dfbetas.influence.merMod        car
```

```
## In case you would like cite this package, cite it as:
##      Patil, I. (2018). ggstatsplot: "ggplot2" Based Plots with Statistical Details. CRAN.
##      Retrieved from https://cran.r-project.org/web/packages/ggstatsplot/index.html
```

```
library(ggTimeSeries)
library(ggbeeswarm)
library(gghalves)
```

Exercise 1:

- There is also a dark theme option with black background

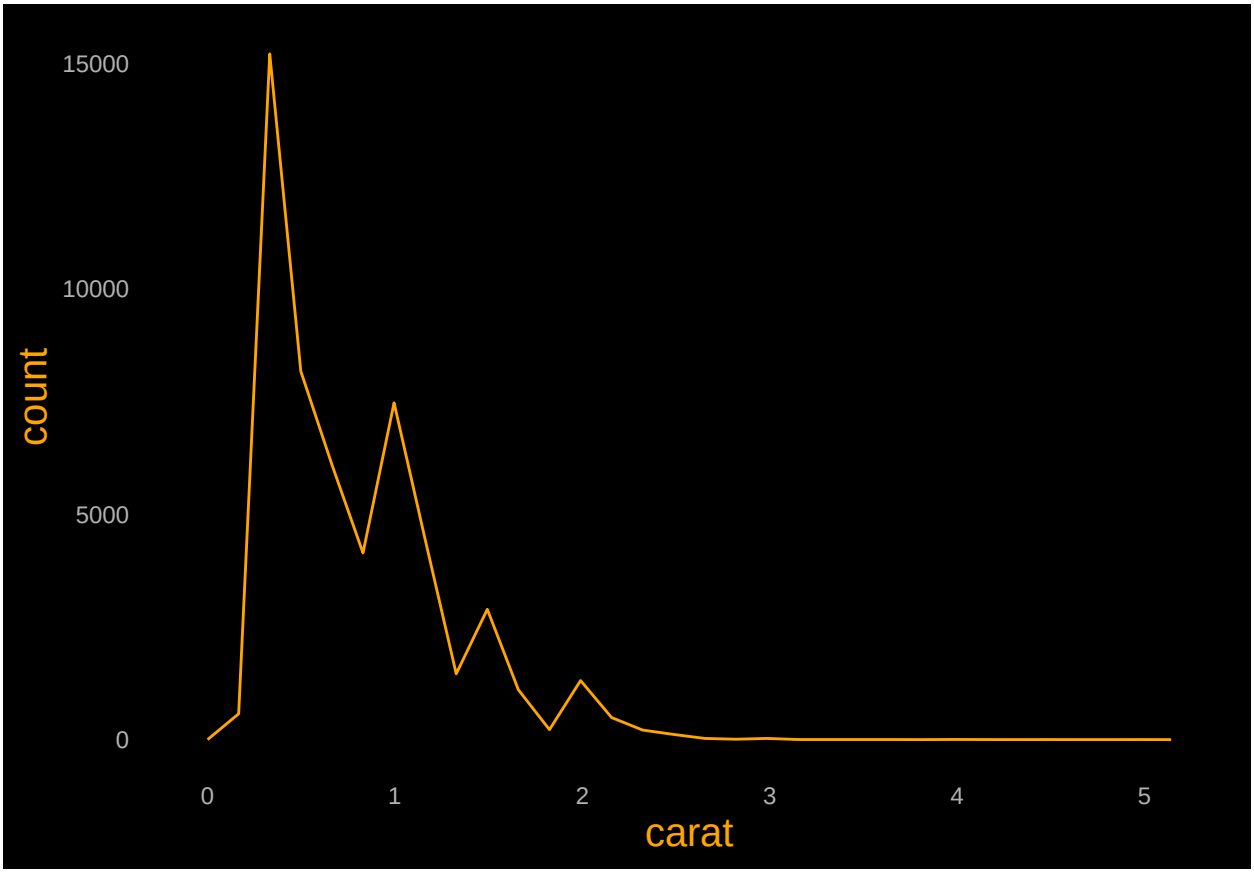
```
trend_color = 'orange'
theme_set(dark_theme_gray()+ theme(
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  plot.title = element_text(size = 18, hjust = 0, color = trend_color),
  axis.ticks = element_blank(),
  axis.title = element_text(size = 15, hjust = 0.5, color = trend_color),
  legend.title = element_blank(),
  panel.background =element_rect(fill = "black"),
  strip.background =element_rect(fill = "black"),
  plot.background = element_rect(fill = "black"),
  legend.background = element_rect(fill = "black")
))
```

```
## Inverted geom defaults of fill and color/colour.
## To change them back, use invert_geom_defaults().
```

- #Simple chart

```
trend_color = 'orange'
ggplot(diamonds, aes(carat)) +
  geom_freqpoly(colour = trend_color)
```

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



Excercise 2:

- #Explorative Analysis (gapminder dataset)
-

Creating a visual analytic 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)
```

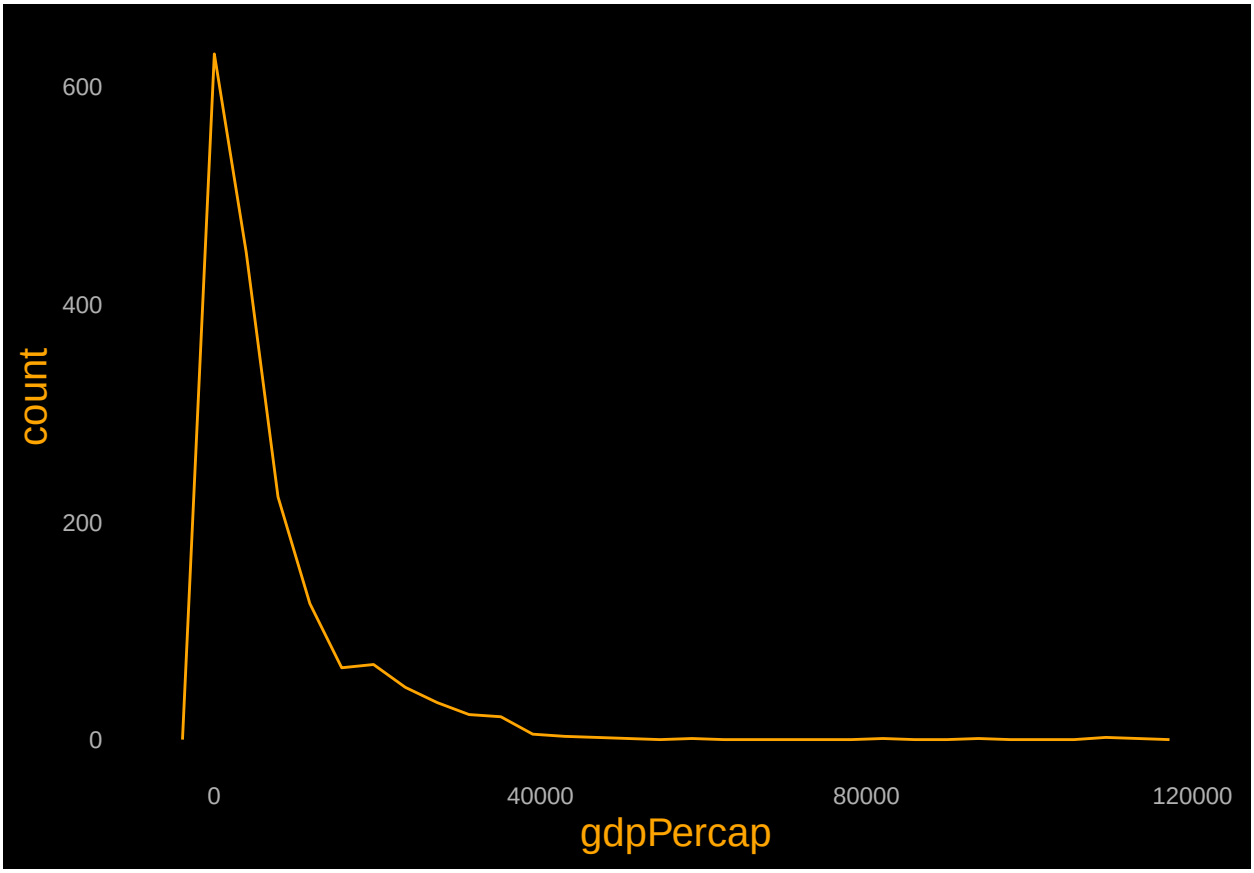
```
## tibble [1,704 x 6] (S3: tbl_df/tbl/data.frame)
##  $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ year     : int [1:1704] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...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

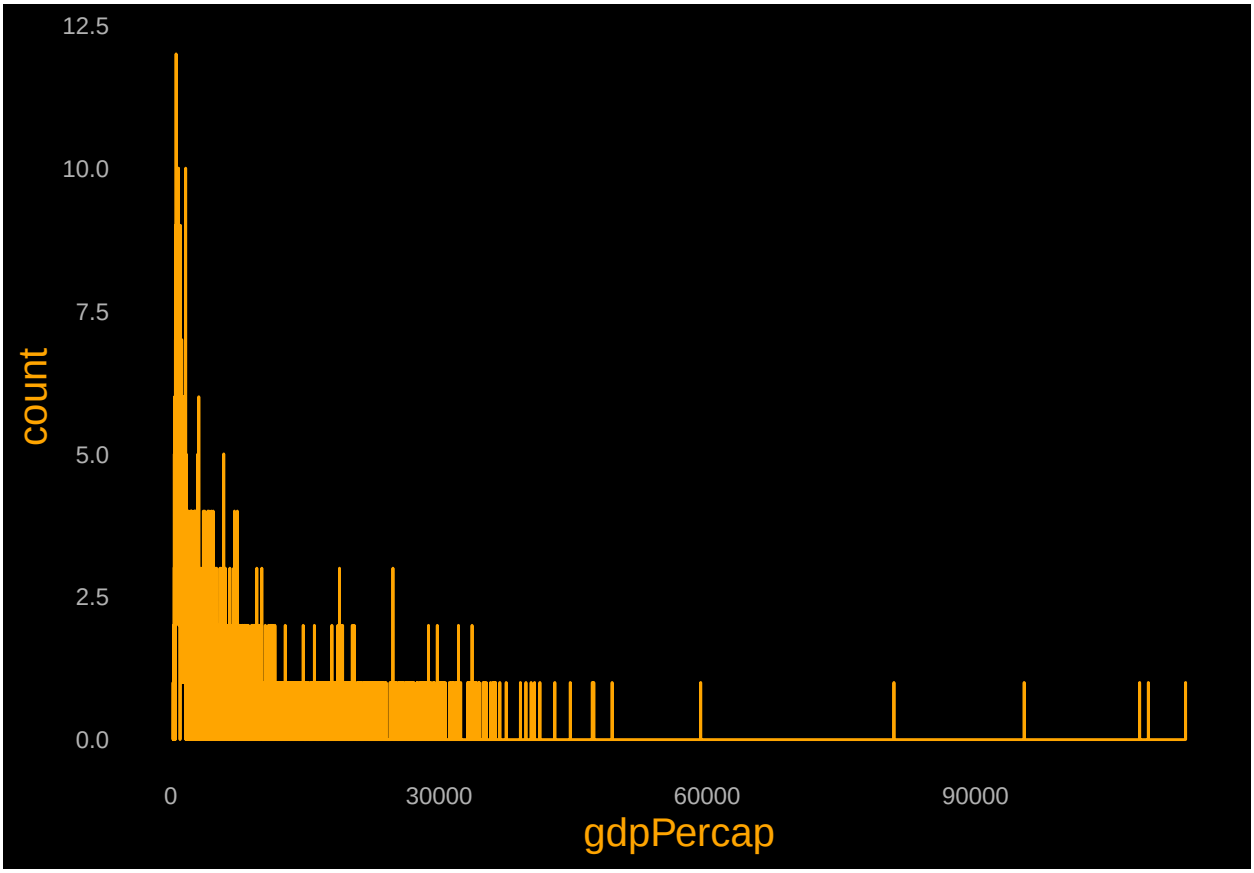
- #Simple chart

```
ggplot(gapminder, aes(gdpPercap)) +
  geom_freqpoly(colour = trend_color, bins=30)
```



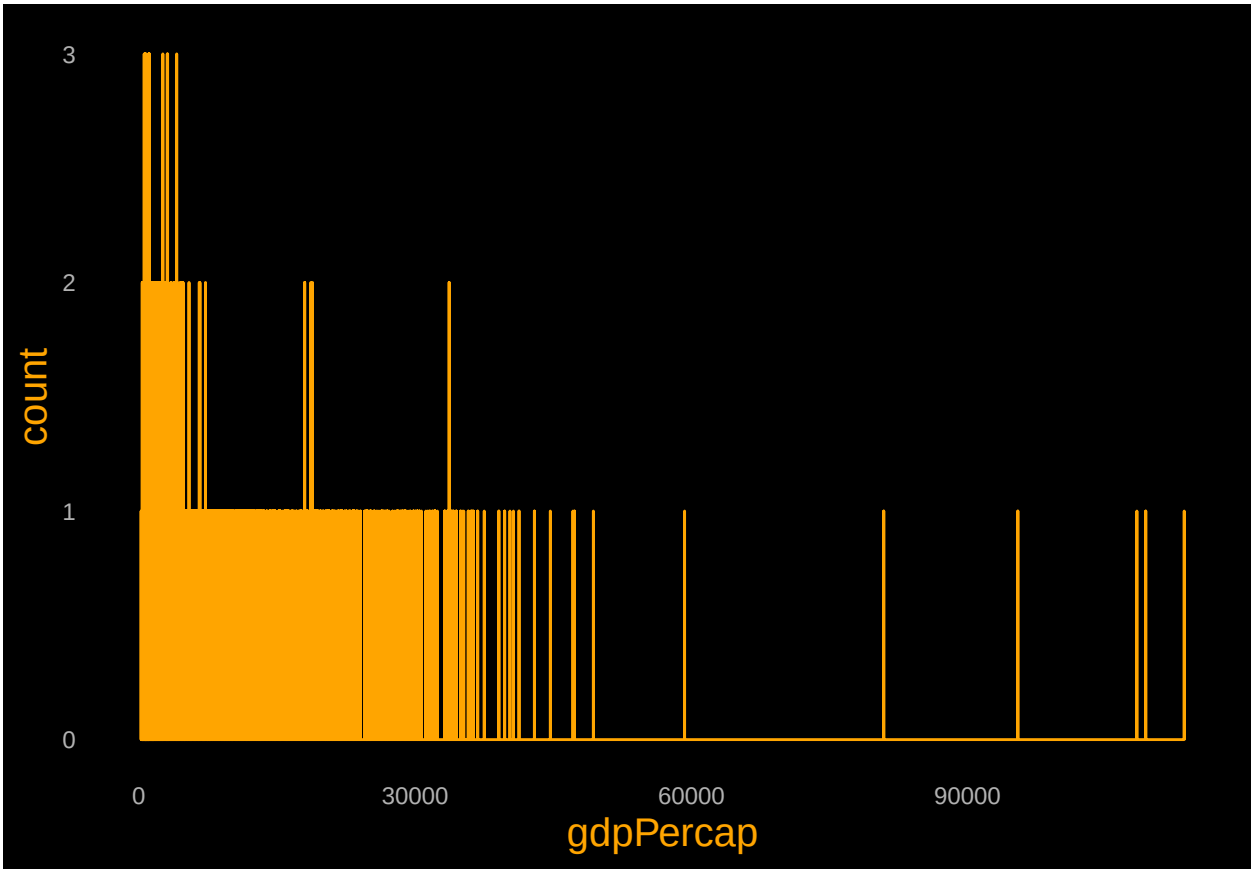
- #Changing the bin width (less details)

```
ggplot(gapminder, aes(gdpPercap)) +
  geom_freqpoly(colour = trend_color, binwidth = 10)
```



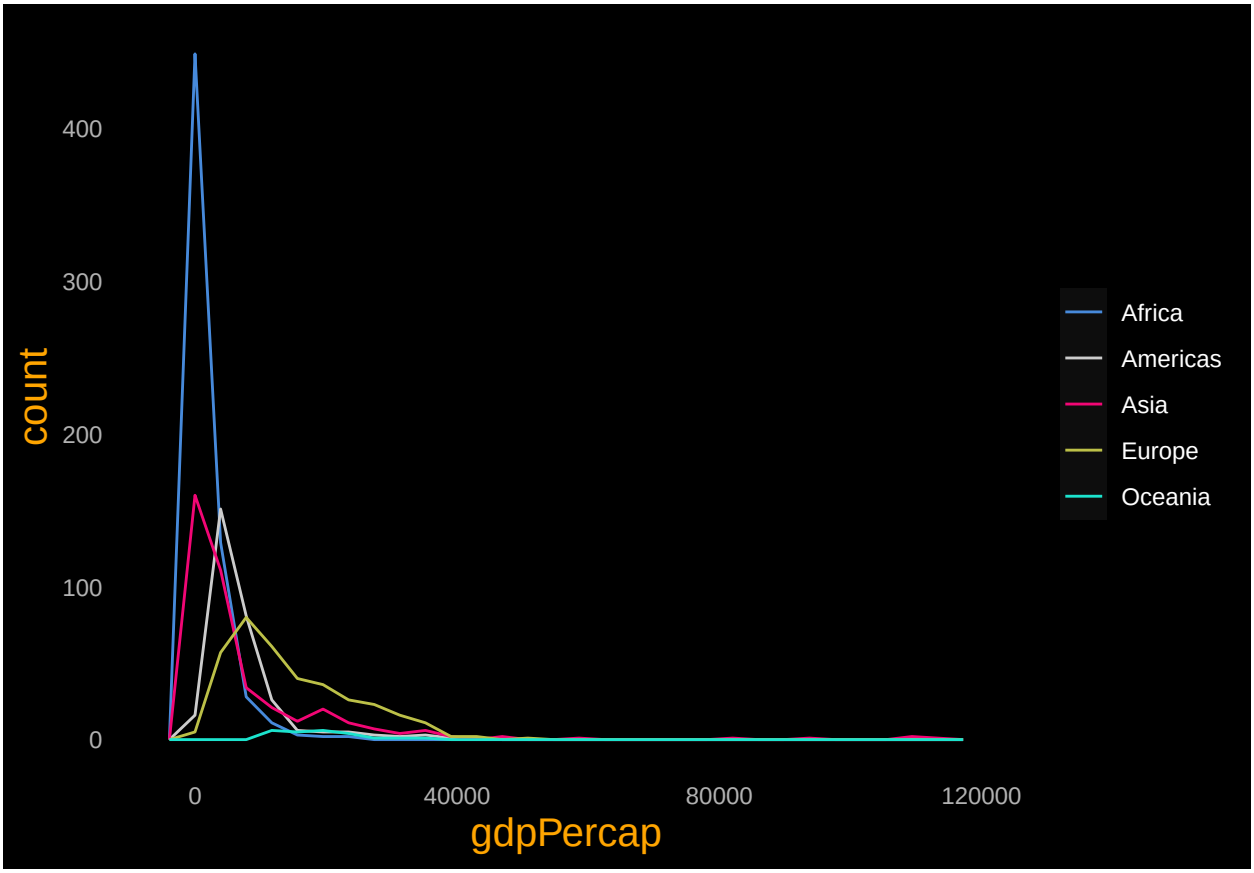
- #Changing the bin width (more details)

```
ggplot(gapminder, aes(gdpPercap)) +
  geom_freqpoly(colour = trend_color, binwidth = 0.8)
```



- #Adding color as a visual encoding

```
ggplot(gapminder, aes(gdpPercap, colour = continent)) +
  geom_freqpoly(bins=30) +
  scale_color_manual(values=c("#478adb", "#cccccc", "#f20675", "#bcc048", "#1ce3cd"))
```

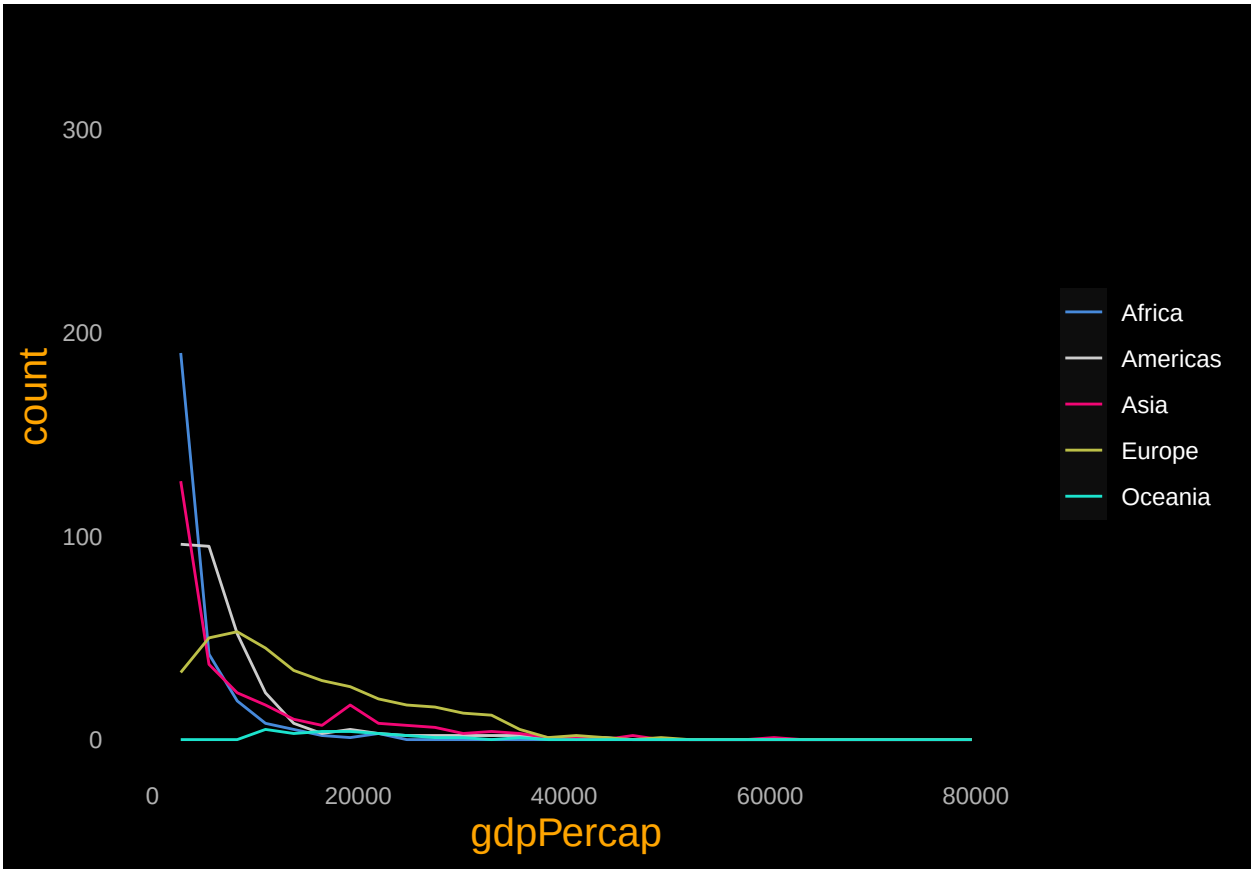


- #How to zoom by defining the limits for the x axis

```
ggplot(gapminder, aes(gdpPercap, colour = continent)) +
  geom_freqpoly(bins=30) +
  scale_color_manual(values=c("#478adb", "#cccccc", "#f20675", "#bcc048", "#1ce3cd")) +
  xlim(400, 80000)
```

```
## Warning: Removed 31 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 15 row(s) containing missing values (geom_path).
```



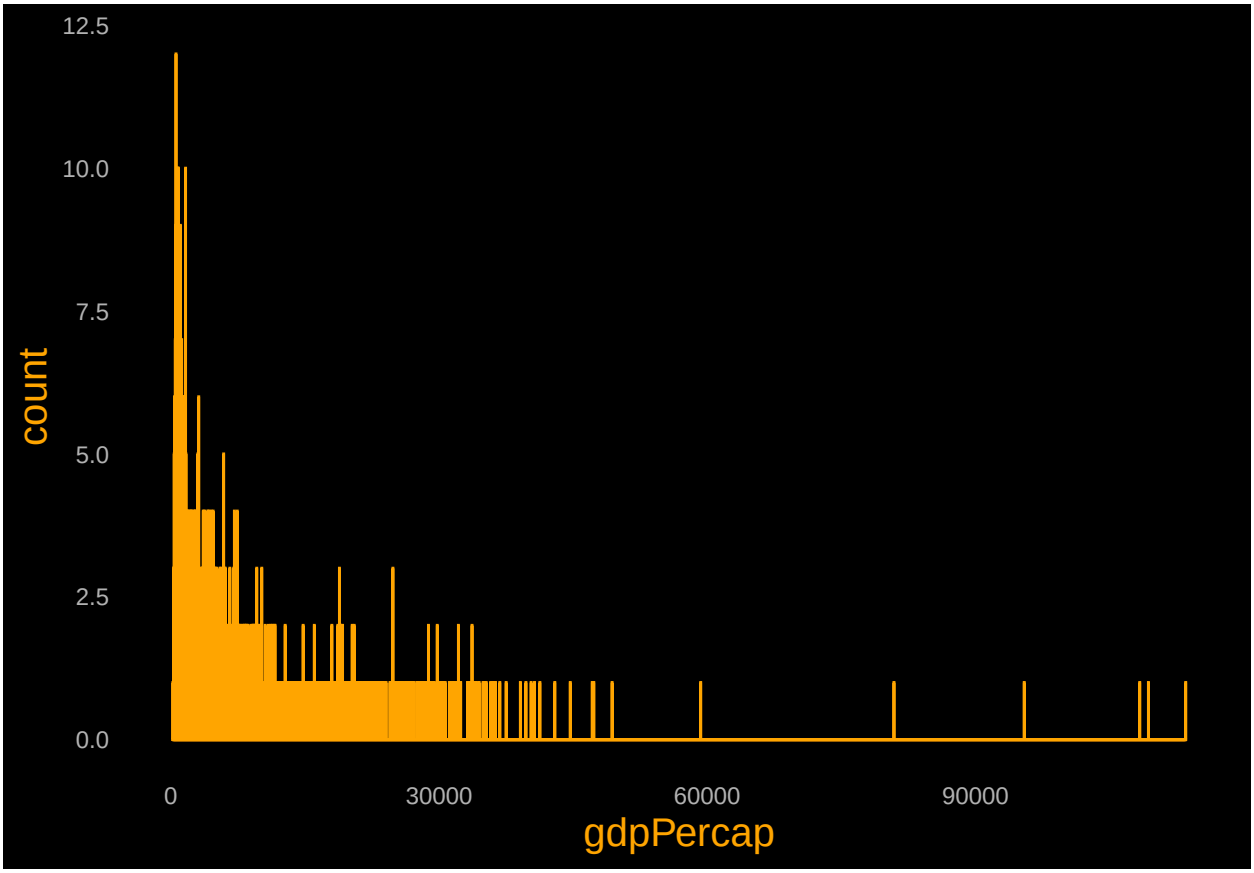
Excercise 3:

- #Checking the options

```
?geom_histogram
```

- #Simple chart, the same with a histogram

```
ggplot(gapminder, aes(gdpPercap)) +
  geom_histogram(colour = trend_color, fill = trend_color, binwidth = 10)
```



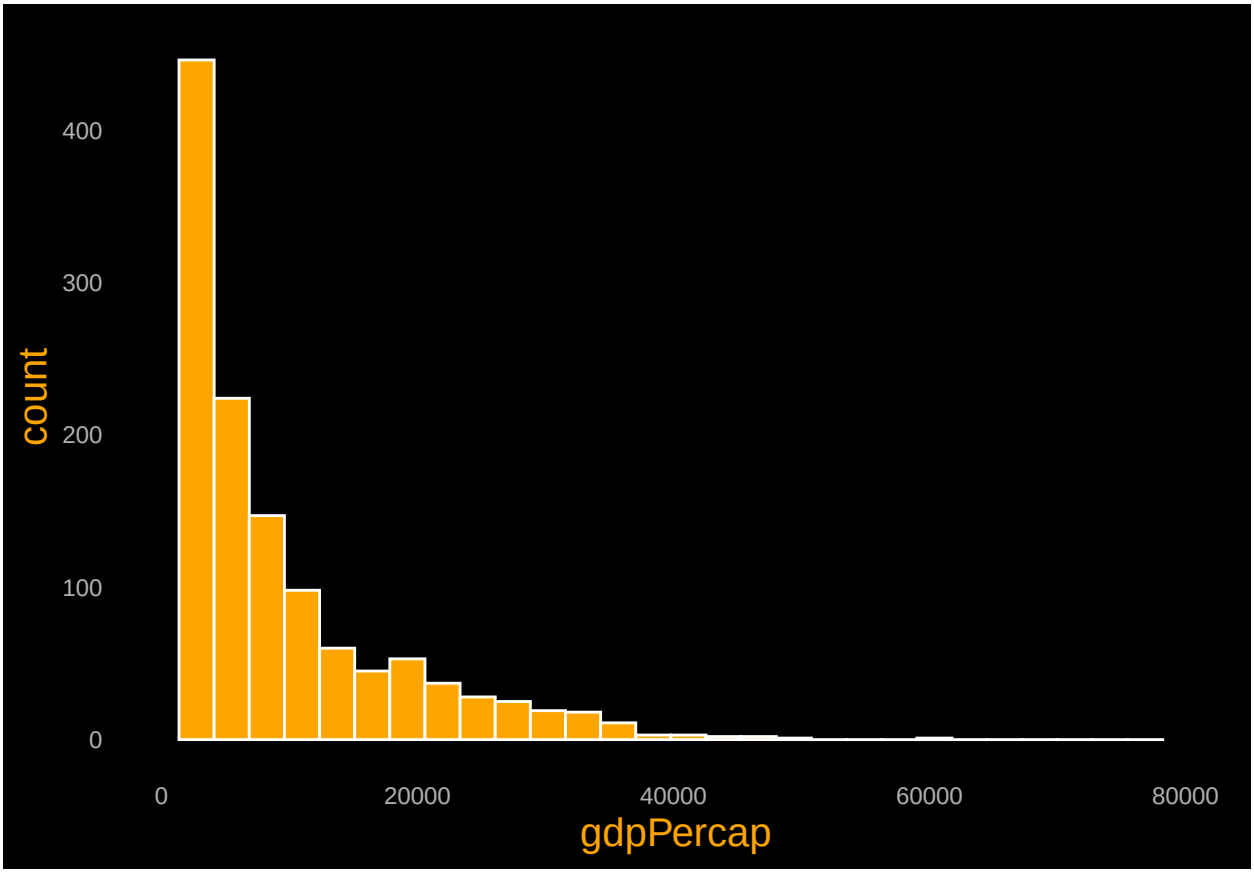
- #How to zoom by defining the limits for the x axis

```
ggplot(gapminder, aes(gdpPercap)) +
  geom_histogram(colour = "white", fill = trend_color) +
  xlim(400, 80000)
```

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

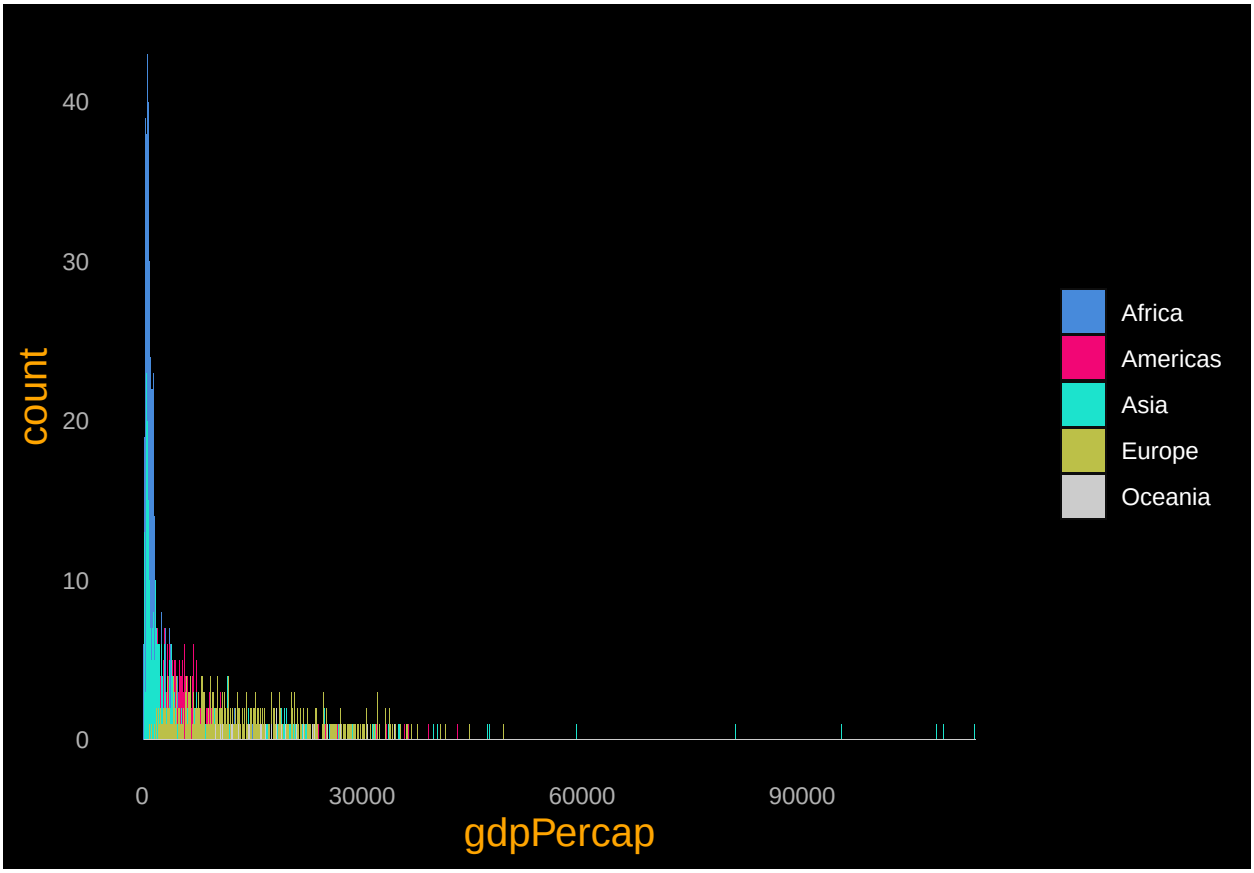
```
## Warning: Removed 31 rows containing non-finite values (stat_bin).
```

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



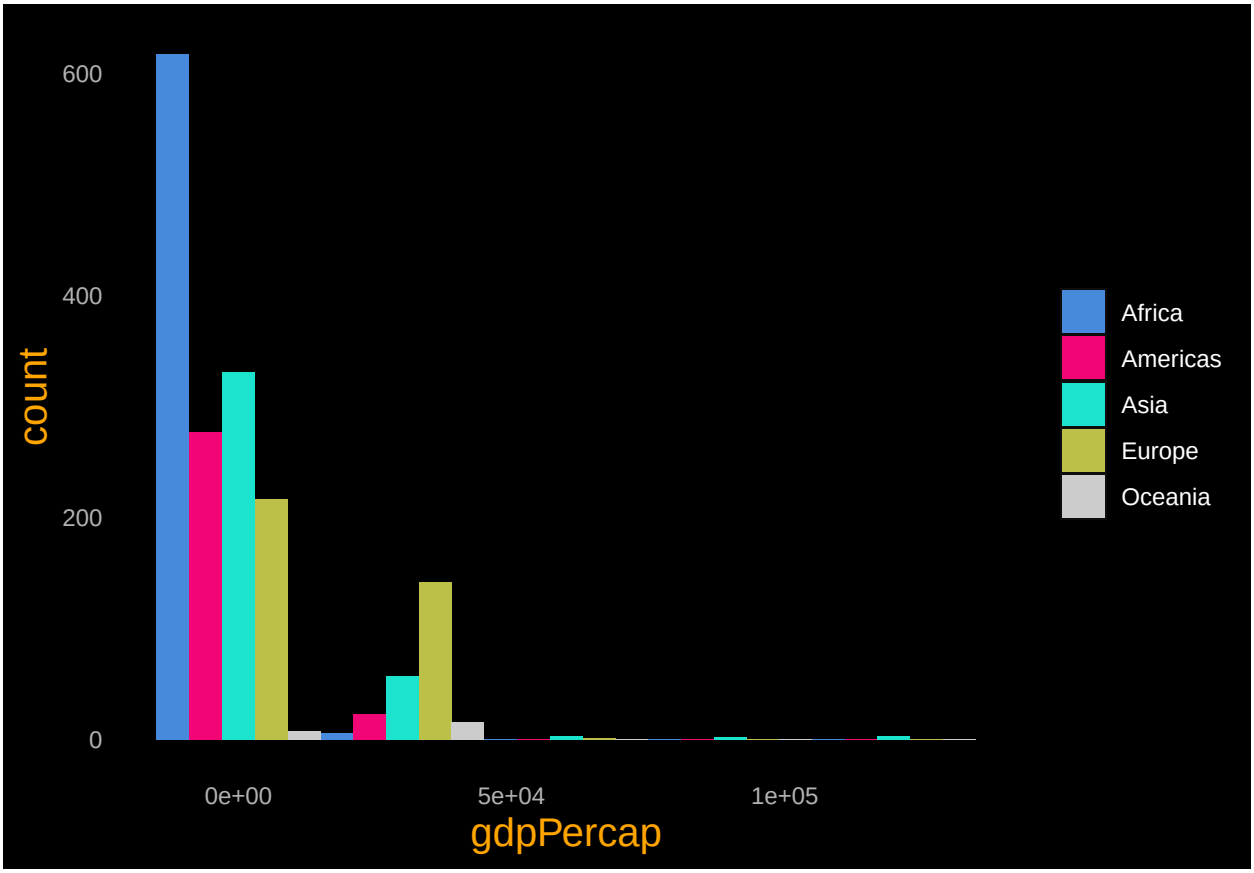
- #Histogram for different cut options

```
ggplot(gapminder, aes(gdpPercap, fill = continent)) +
  geom_histogram(position = "dodge", bins=1200) +
  scale_fill_manual(values=c("#478adb", "#f20675", "#1ce3cd", "#bcc048", "#cccccc"))
```



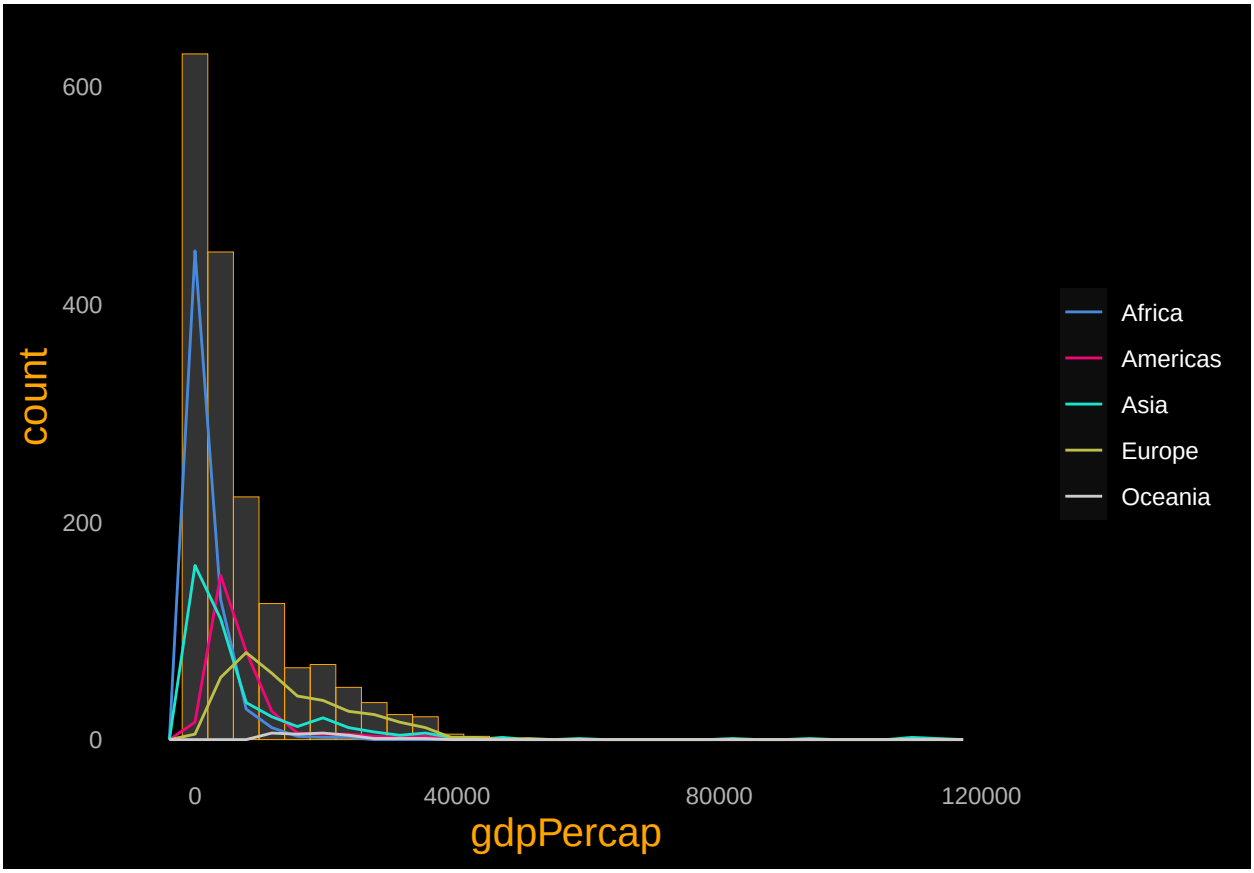
- #Changing the bin options

```
ggplot(gapminder, aes(gdpPercap, fill = continent)) +
  geom_histogram(position = "dodge", binwidth = 30000) +
  scale_fill_manual(values=c("#478adb", "#f20675", "#1ce3cd", "#bcc048", "#cccccc"))
```



- #The whole idea of the grammar of graphs

```
ggplot(gapminder, aes(gdpPercap, color = continent)) +
  geom_histogram(colour= trend_color, fill = "white", alpha = 0.2, size =0, bins=30) +
  geom_freqpoly(bins=30)+
  scale_colour_manual(values=c("#478adb", "#f20675", "#1ce3cd", "#bcc048", "#cccccc"))
```

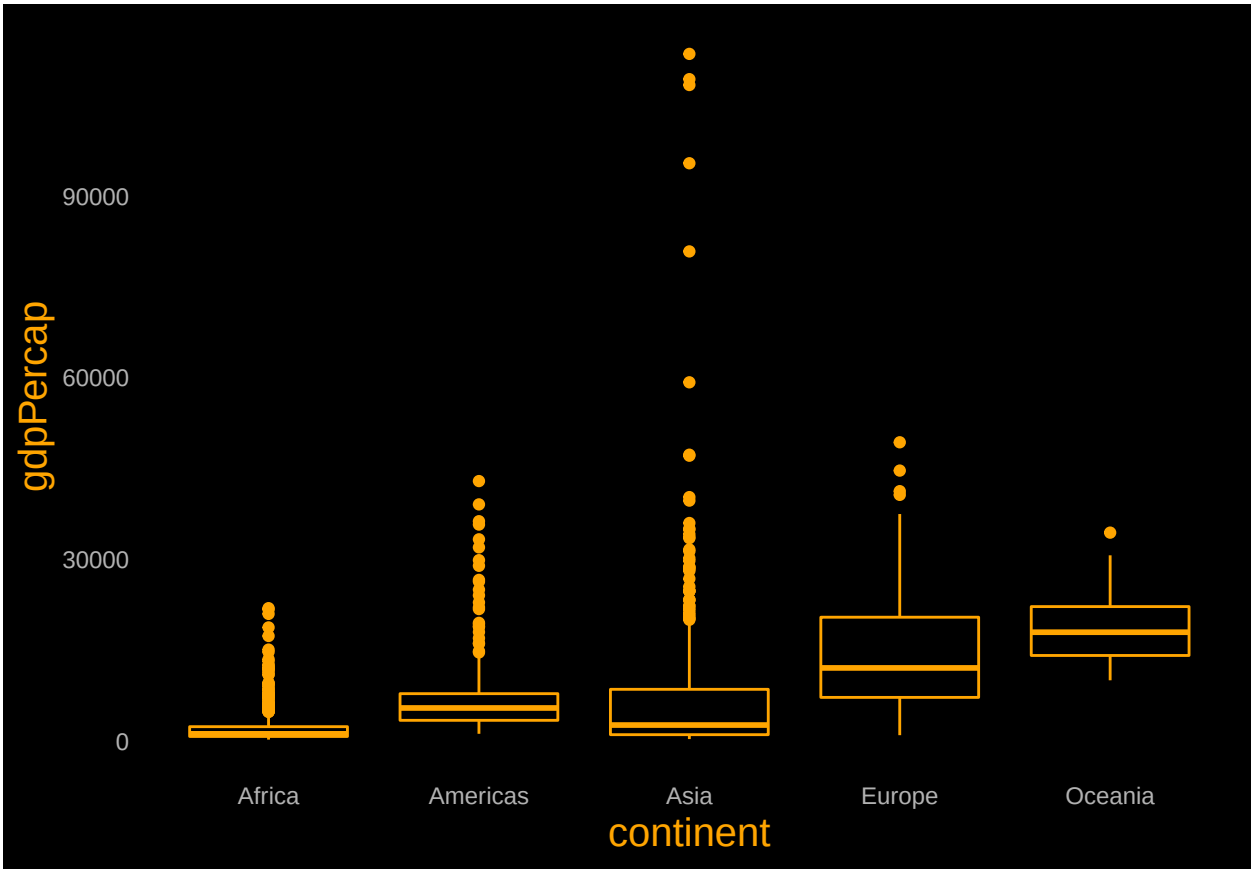


Excercise 4: - #Checking the options

```
?geom_boxplot
```

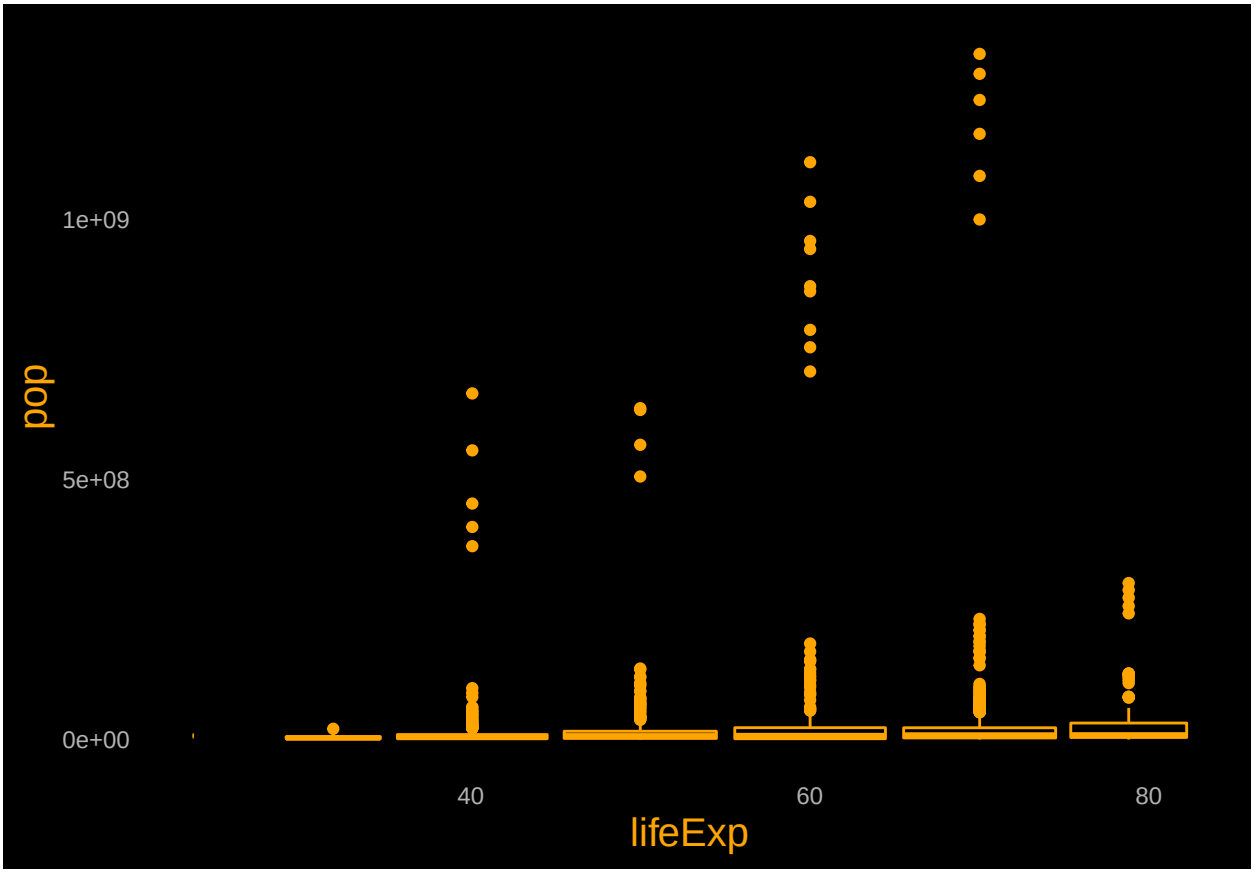
- #Simple boxplot by category

```
ggplot(gapminder, aes(continent, gdpPercap)) +
  geom_boxplot(colour=trend_color)
```



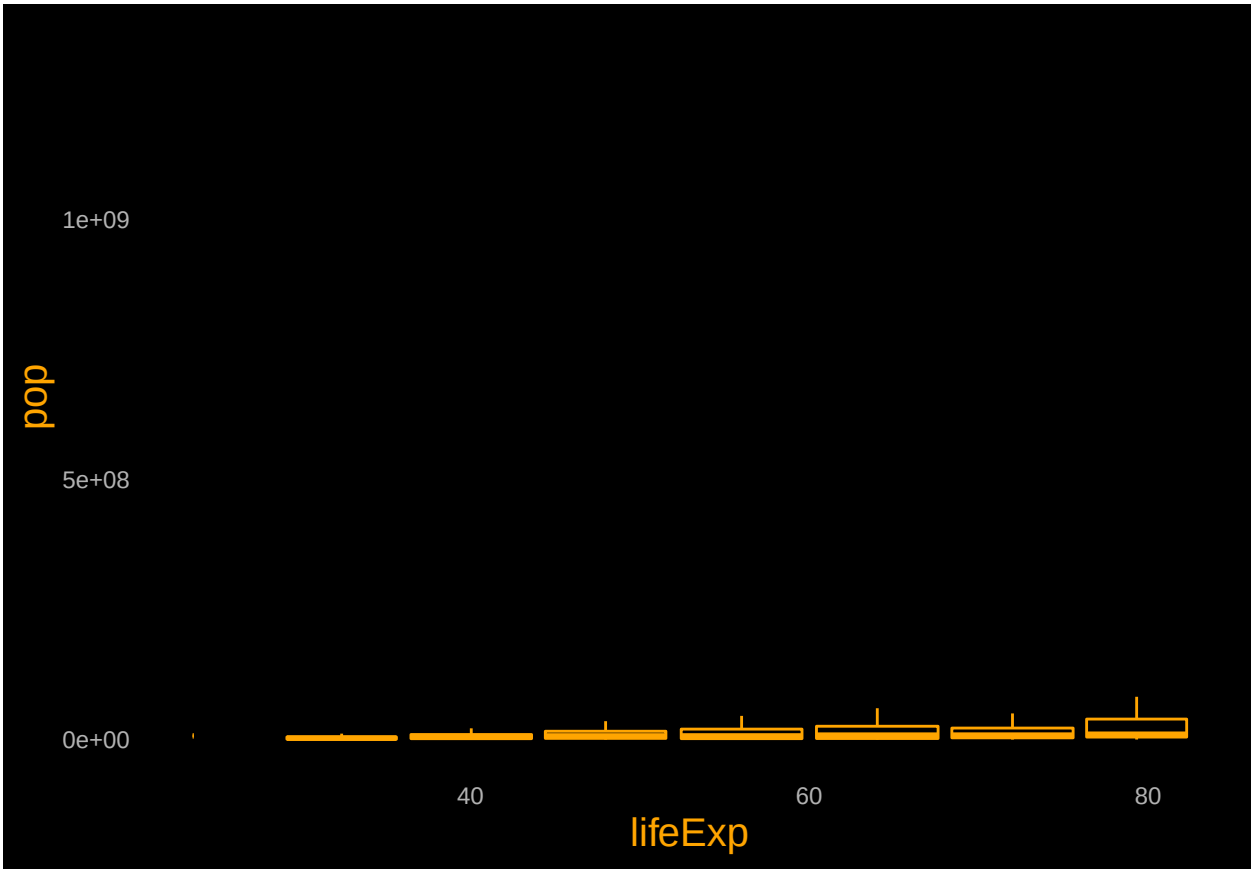
- #Boxplot using to numeric variable, we need to define a grouping rule

```
ggplot(gapminder, aes(lifeExp, pop)) +
  geom_boxplot(aes(group = cut_width(lifeExp, 10)), color=trend_color)
```



- #Without outliers

```
ggplot(gapminder, aes(lifeExp, pop)) +
  geom_boxplot(aes(group = cut_width(lifeExp, 8)), color=trend_color,
               outlier.alpha=0)
```

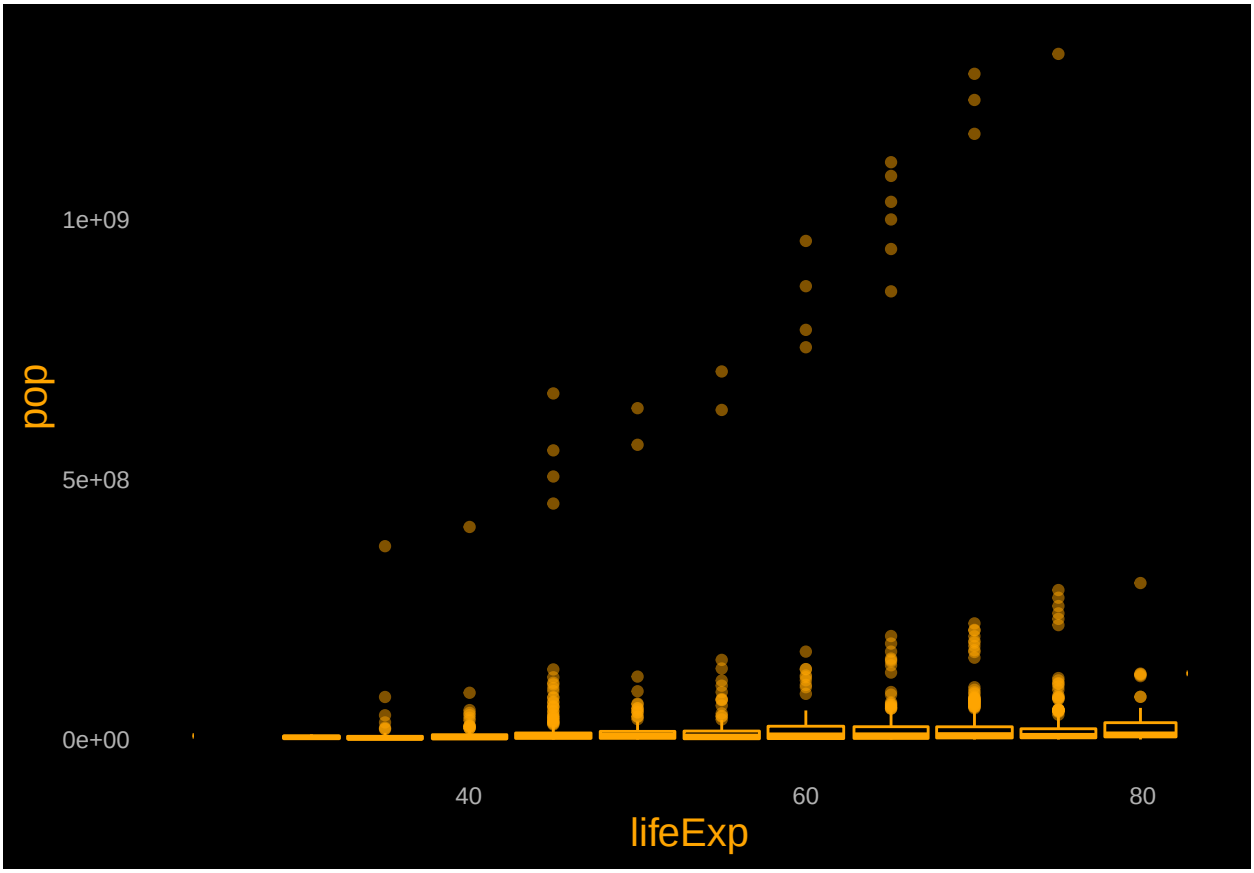


-

Use a specific encoding for the outliers

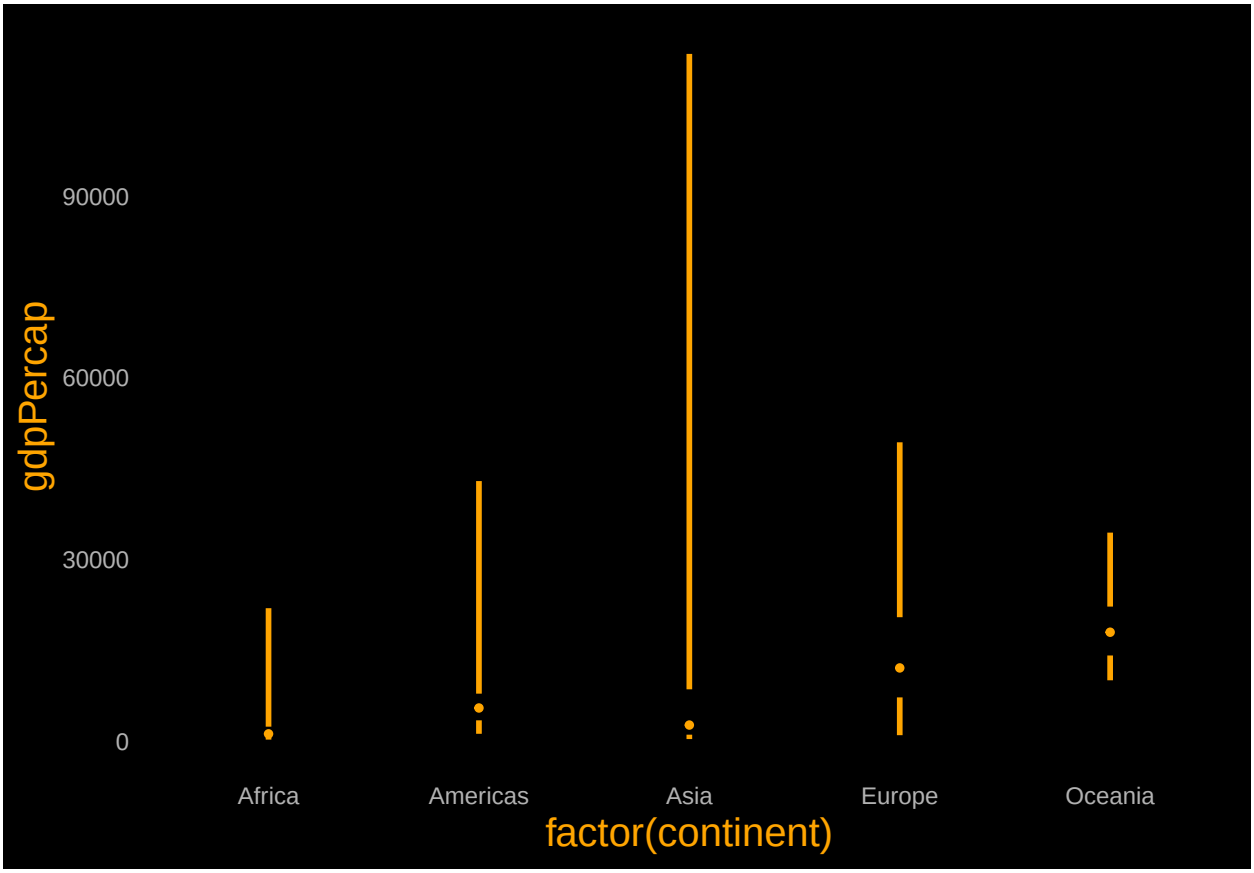
```
ggplot(gapminder, aes(lifeExp, pop)) +
  geom_boxplot(aes(group = cut_width(lifeExp, 5)), color=trend_color,
               color= "white",
               outlier.alpha = 0.5,
               outlier.shape = 19,
               outlier.color=trend_color)
```

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



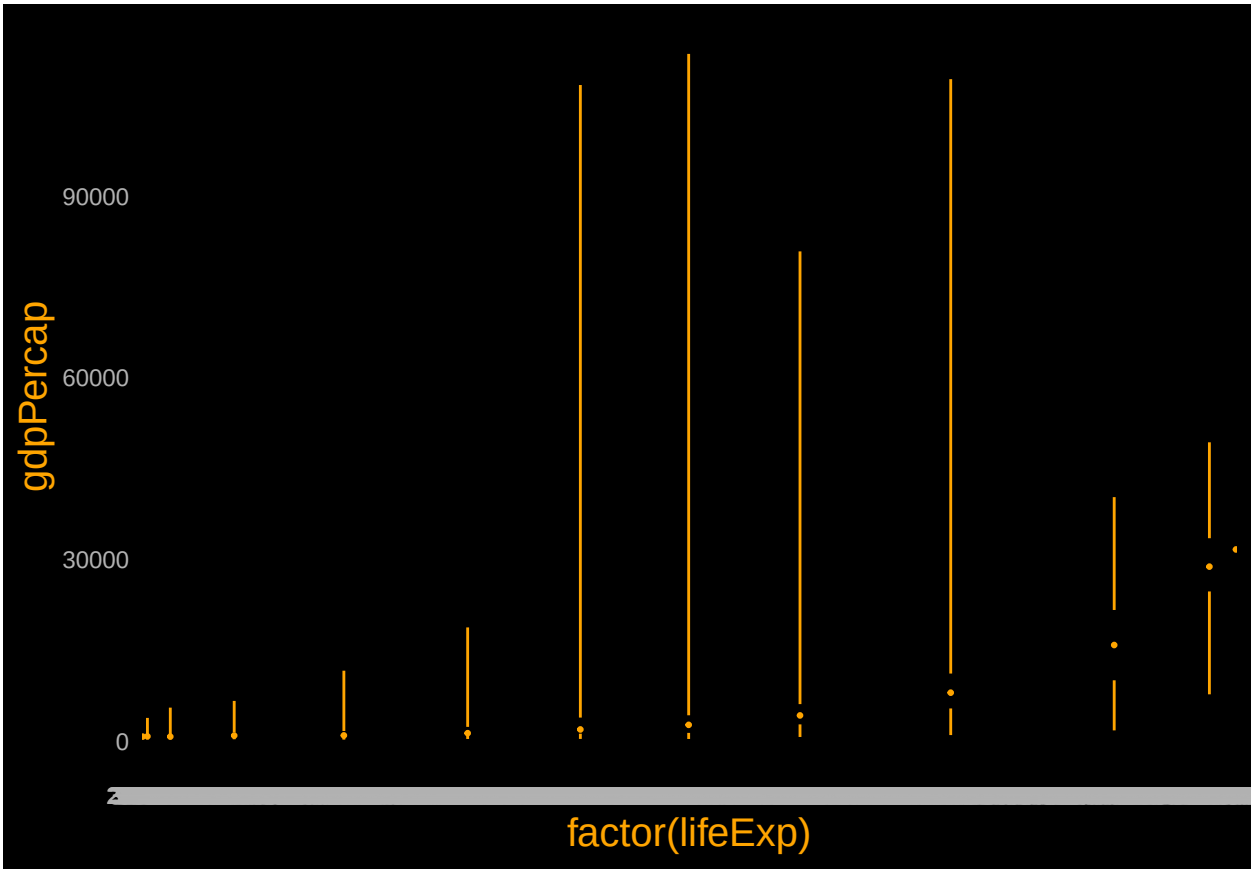
- #Tufe boxplot

```
ggplot(gapminder, aes(factor(continent),gdpPercap)) +
  geom_tufteboxplot(outlier.colour="transparent", size=1, color=trend_color)
```



- #Tufe boxplot

```
ggplot(gapminder, aes(factor(lifeExp),gdpPercap)) +
  geom_tufteboxplot(aes(group = cut_width(lifeExp, 5)), color=trend_color)
```

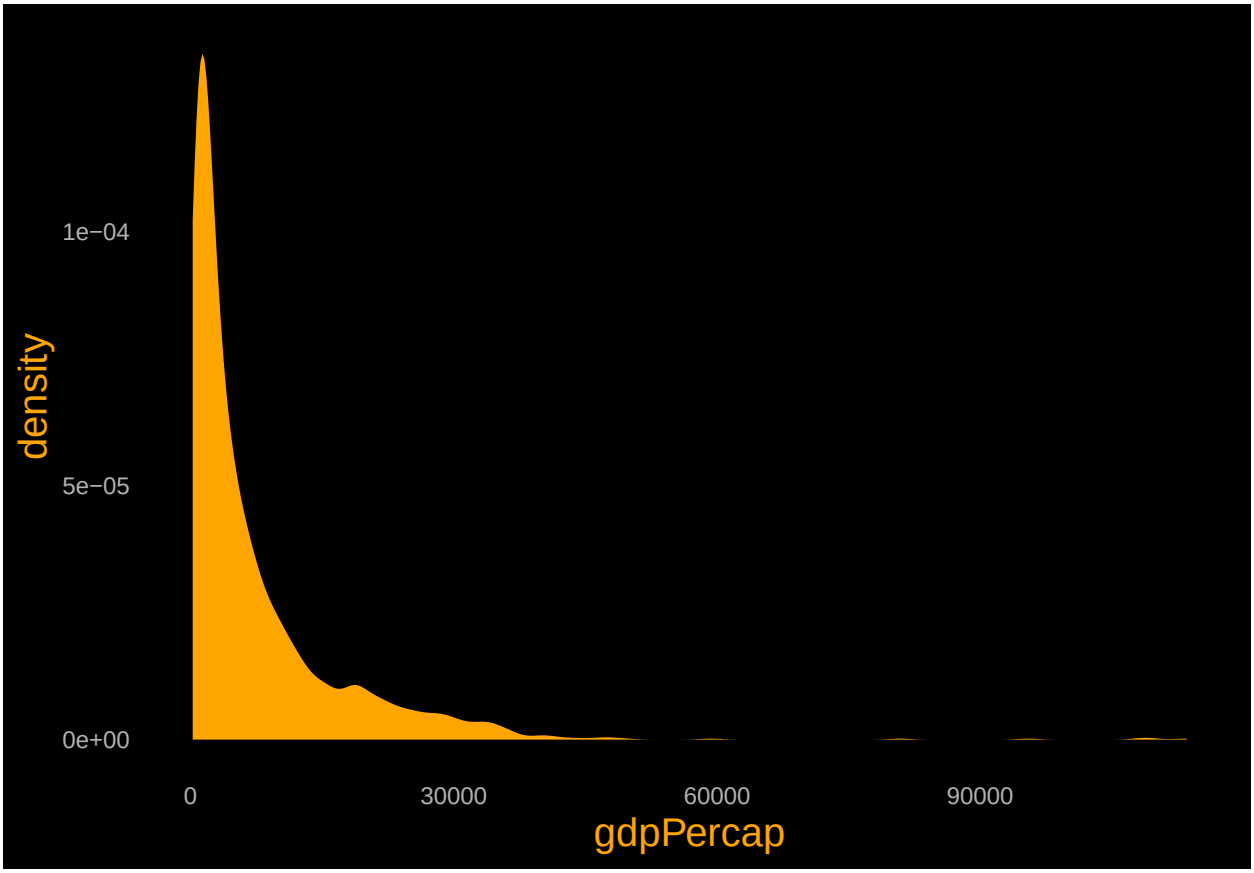


Excercise 5: - #Checking the options

```
?geom_density
```

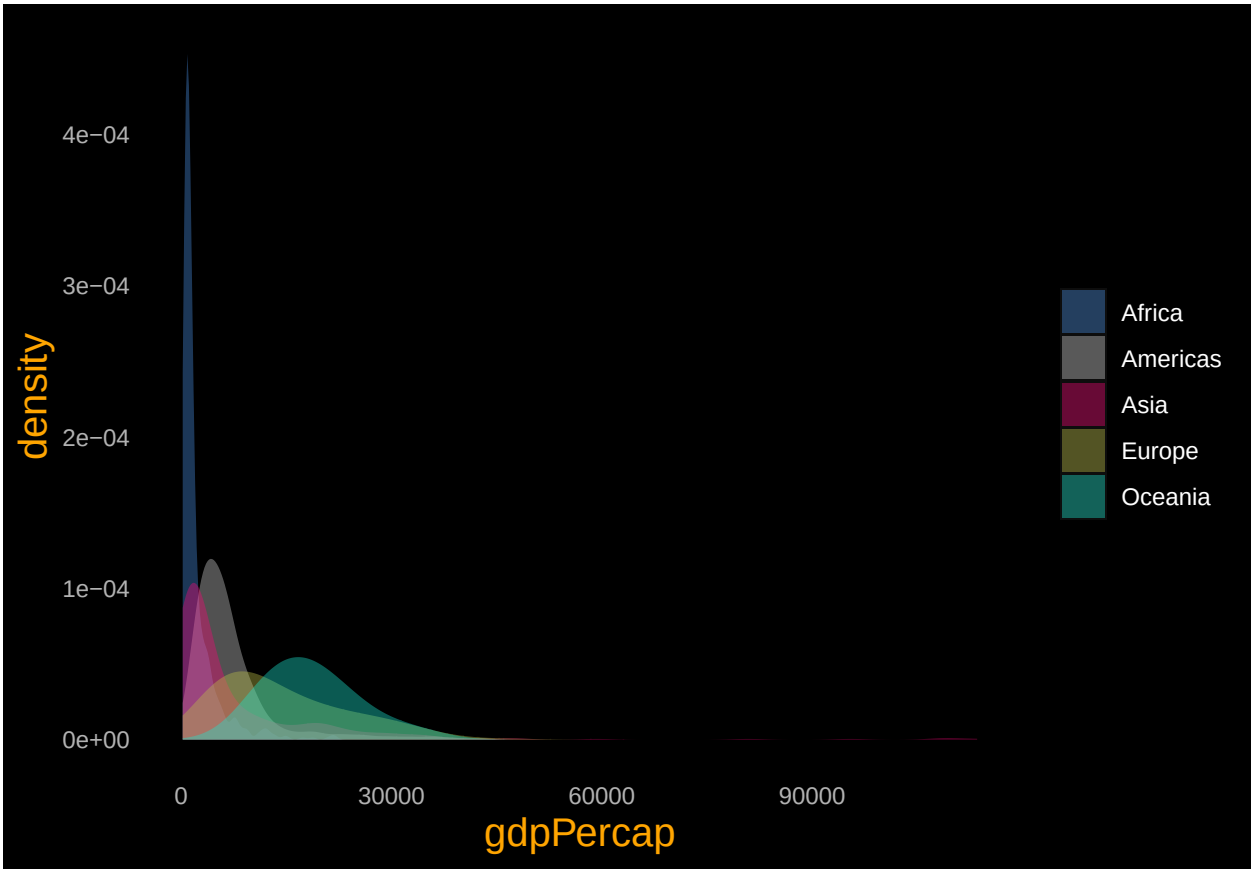
- #Simple chart - the same with a density chart

```
ggplot(gapminder, aes(gdpPercap)) +
  geom_density(fill = trend_color, color = NA)
```



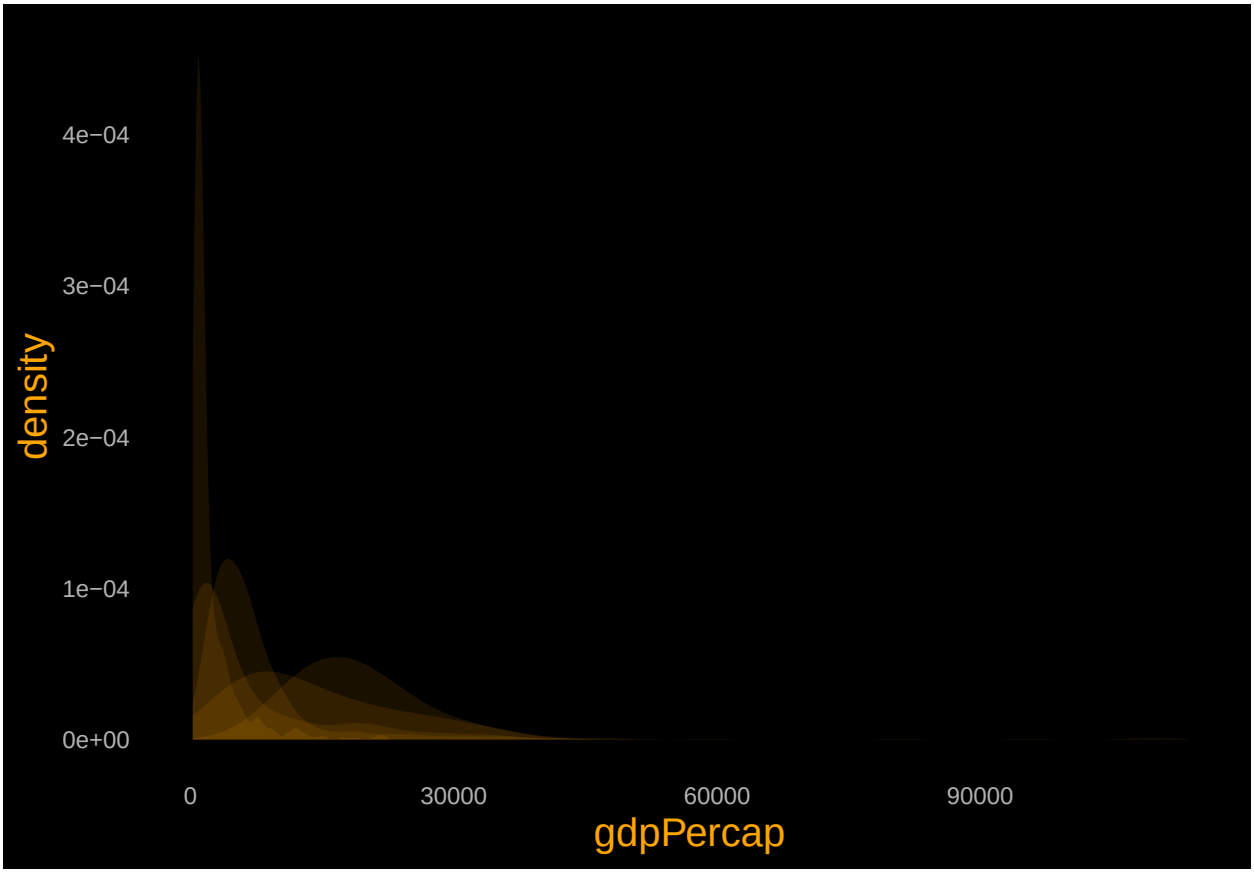
- #Multiple density chart

```
ggplot(gapminder, aes(gdpPercap, group = continent, fill = continent)) +
  geom_density(adjust = 1.5 , color = NA, alpha = 0.4) +
  scale_fill_manual(values=c("#478adb", "#cccccc", "#f20675", "#bcc048", "#1ce3cd"))
```



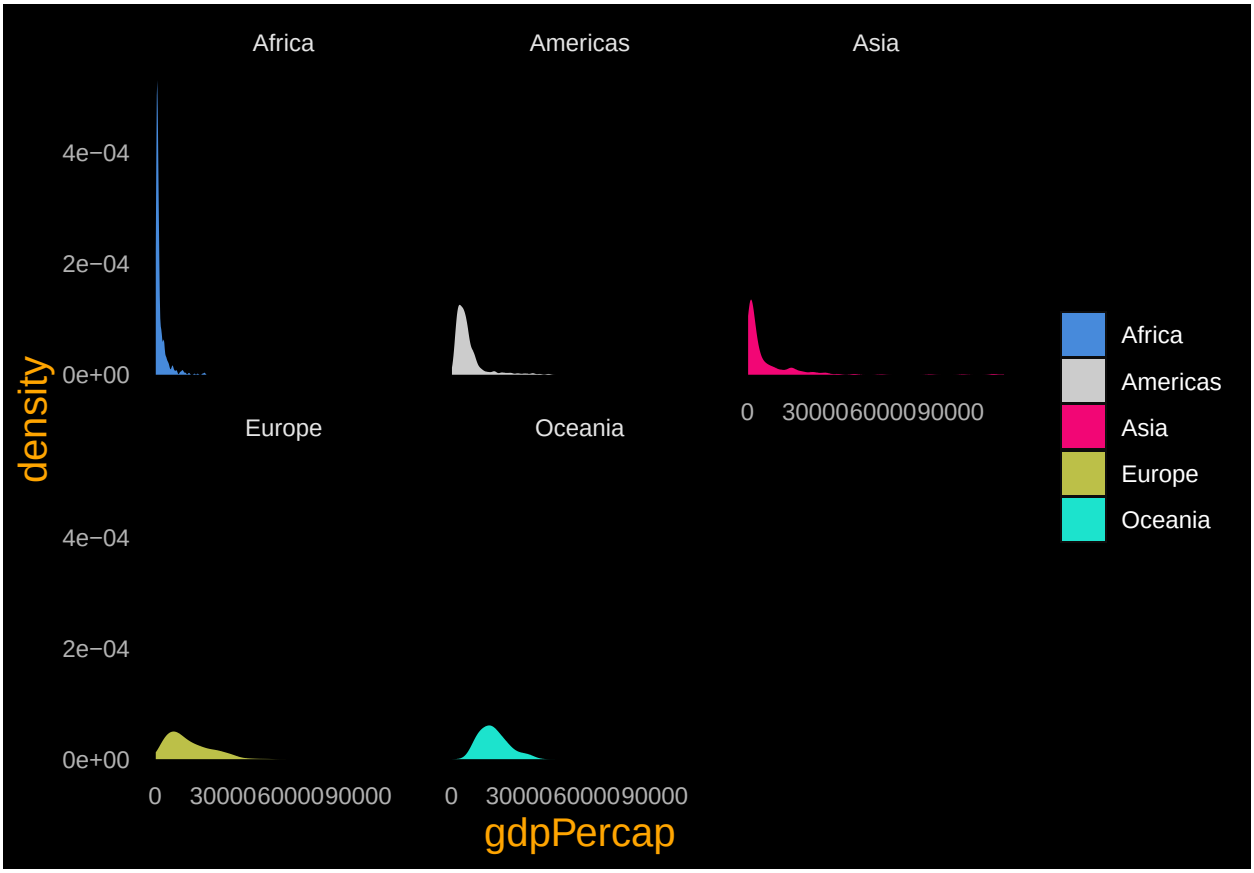
- #Multiple density chart, with using one color and transparency we can identify easly the overlap as a more dense part through all cuts

```
ggplot(gapminder, aes(gdpPercap, group=continent, fill=continent)) +
  geom_density(adjust=1.5 , color= NA, fill=trend_color, alpha =0.1)
```



- #Small multiple density for carat by the different cuts

```
ggplot(gapminder, aes(gdpPercap, stat(density), fill=continent)) +
  geom_density(color = NA) +
  scale_fill_manual(values=c("#478adb", "#cccccc", "#f20675", "#bcc048", "#1ce3cd")) +
  facet_wrap(. ~ continent)
```

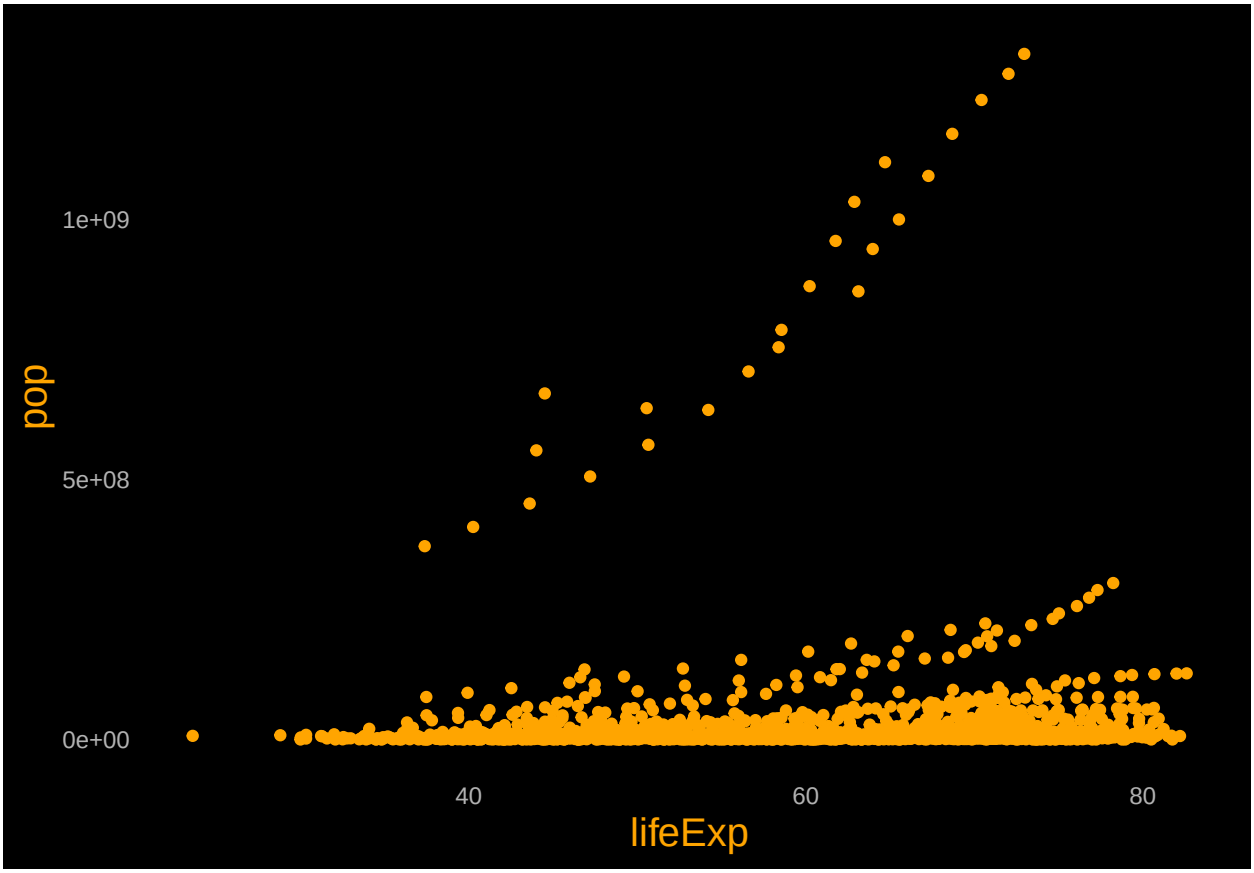


Excercise 6: - #Scatter plot - #Checking the options

```
?geom_point
```

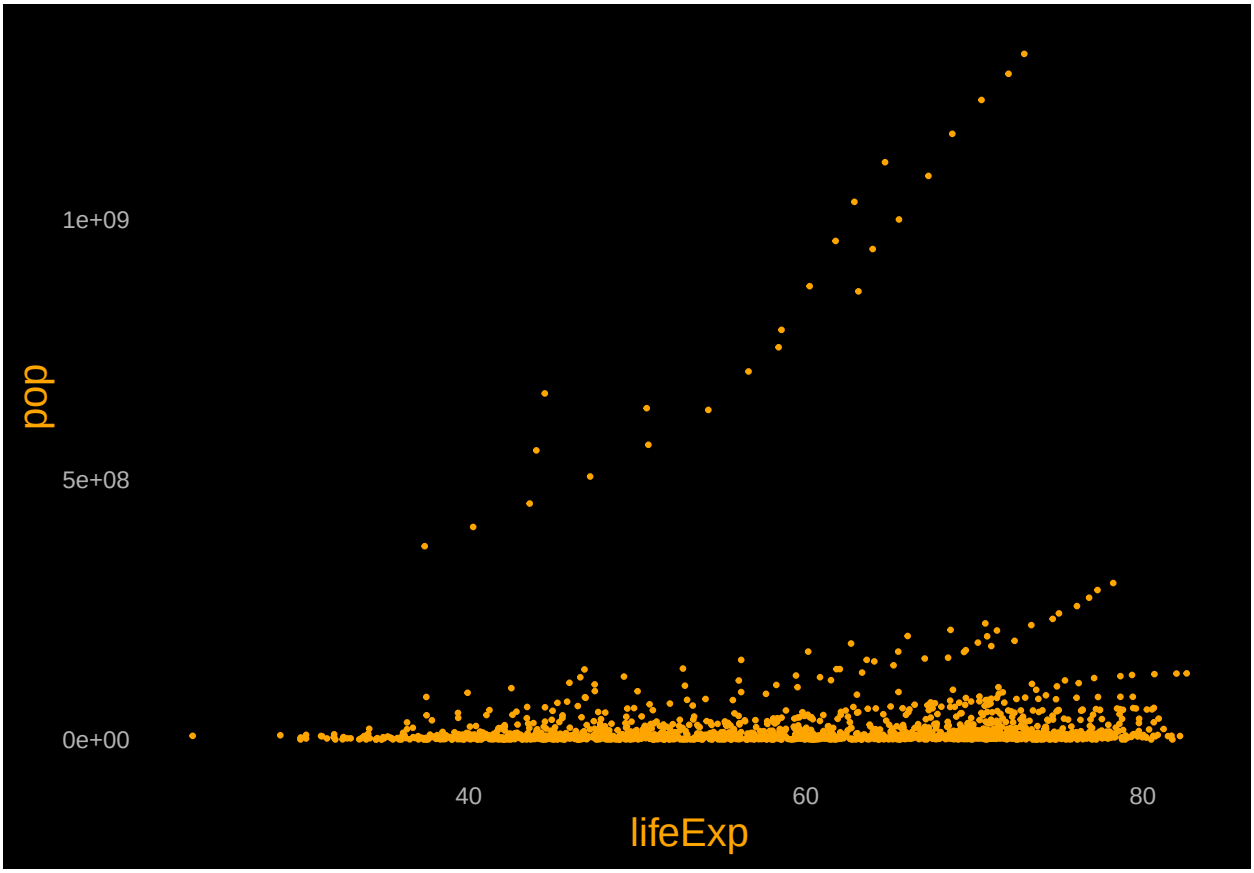
- #Basic scatter plot

```
ggplot(gapminder, aes(x=lifeExp, y=pop)) +
  geom_point(color=trend_color)
```



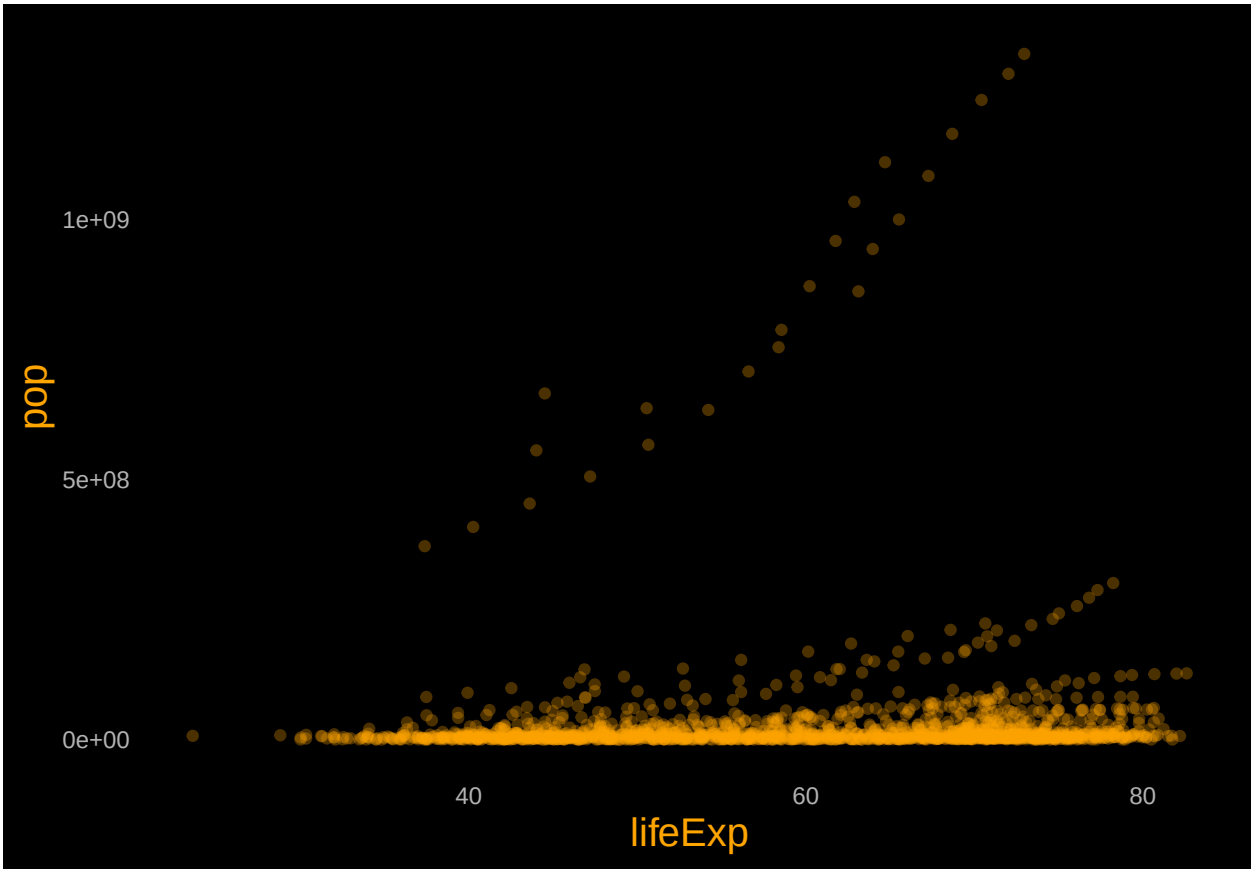
- #Basic scatter plot - adjusting the size

```
ggplot(gapminder, aes(x=lifeExp, y=pop)) +
  geom_point(size=0.5, color=trend_color)
```



- #Basic scatter plot - adjusting the opacity

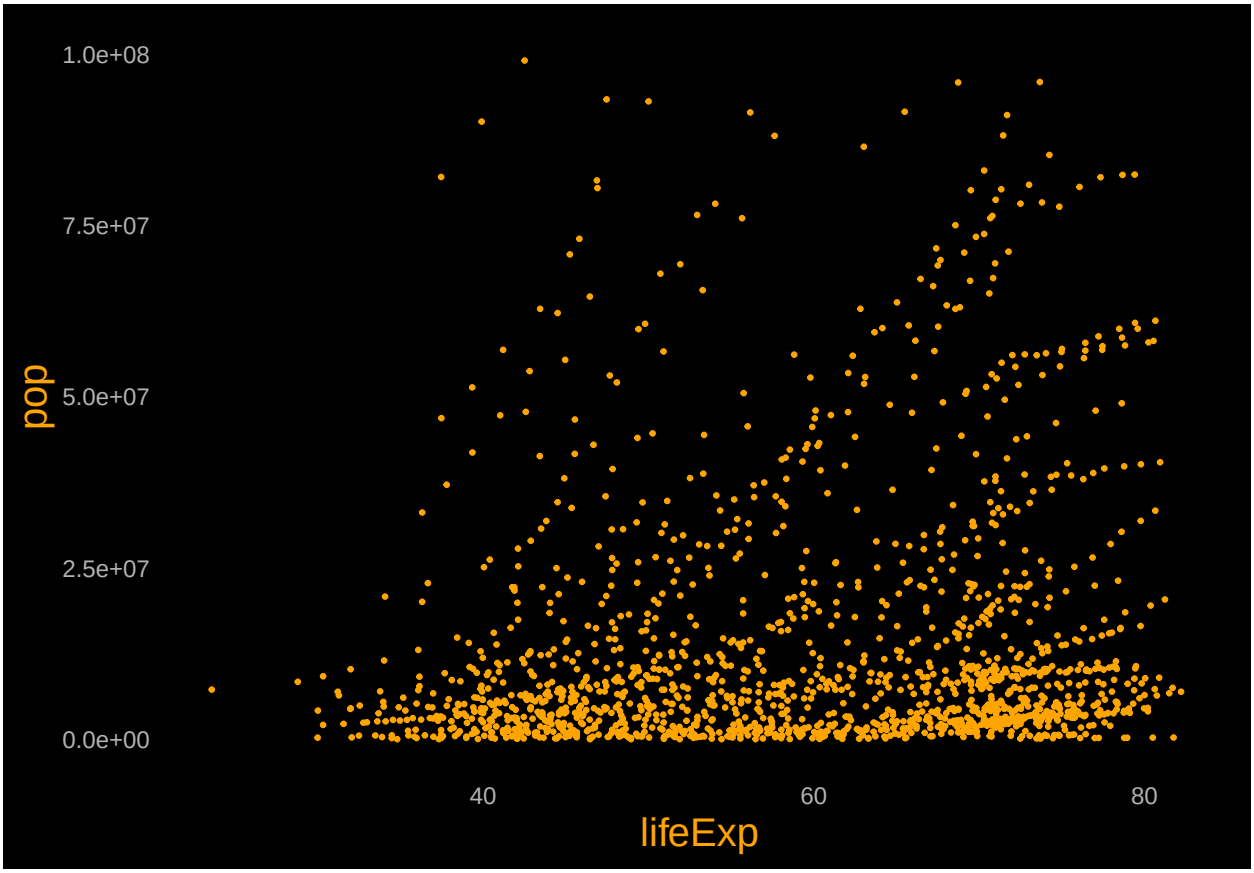
```
ggplot(gapminder, aes(x=lifeExp, y=pop)) +
  geom_point(alpha=0.3, color=trend_color)
```



- #Basic scatter plot changing the Y limits

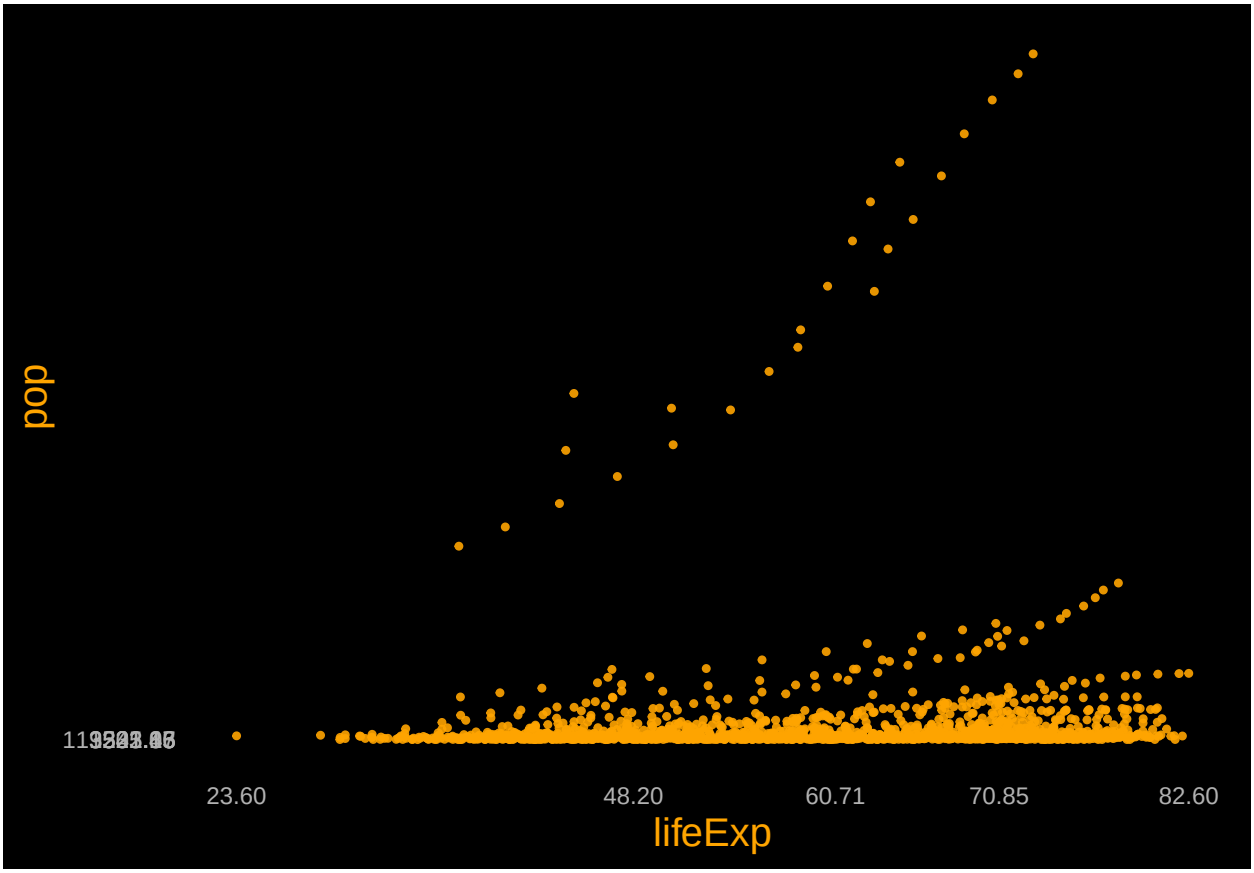
```
ggplot(gapminder, aes(x=lifeExp, y=pop)) +
  geom_point(size=0.5, color=trend_color) +
  ylim(0, 1e+08)
```

```
## Warning: Removed 77 rows containing missing values (geom_point).
```



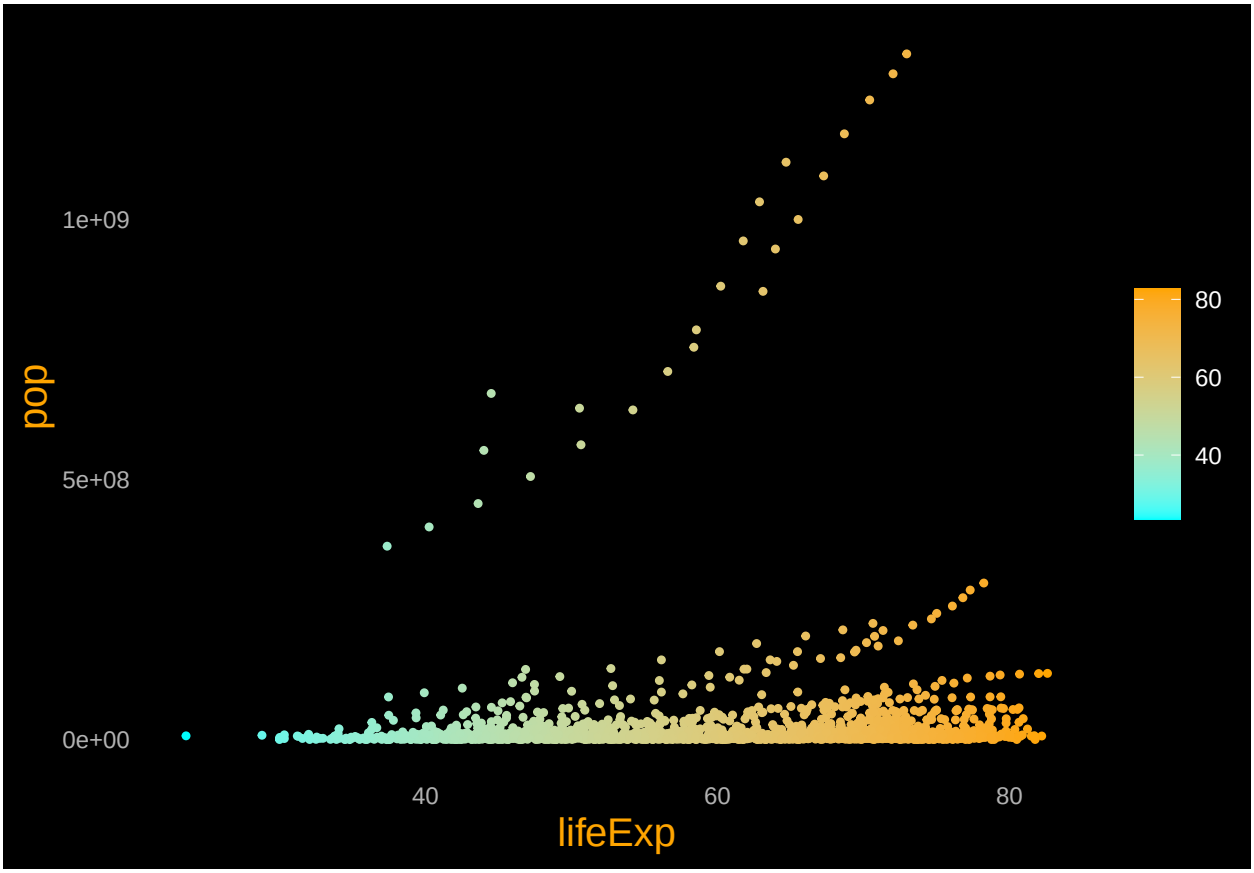
- #Axis labeling depending on the quantiles

```
ggplot(gapminder, aes(x=lifeExp, y=pop)) +
  geom_point(size=0.9, alpha=0.9, color=trend_color)  +
  scale_x_continuous(breaks = round(as.vector(quantile(gapminder$lifeExp)), digits = 2))+
  scale_y_continuous(breaks = round(as.vector(quantile(gapminder$gdpPercap)), digits = 2))
```



- #Adding price as another visual encoding using a colour code

```
ggplot(gapminder, aes(x=lifeExp, y=pop, colour = lifeExp)) +
  geom_point(size=0.9, alpha=5)+
  scale_colour_gradient(low = "cyan", high = trend_color)
```

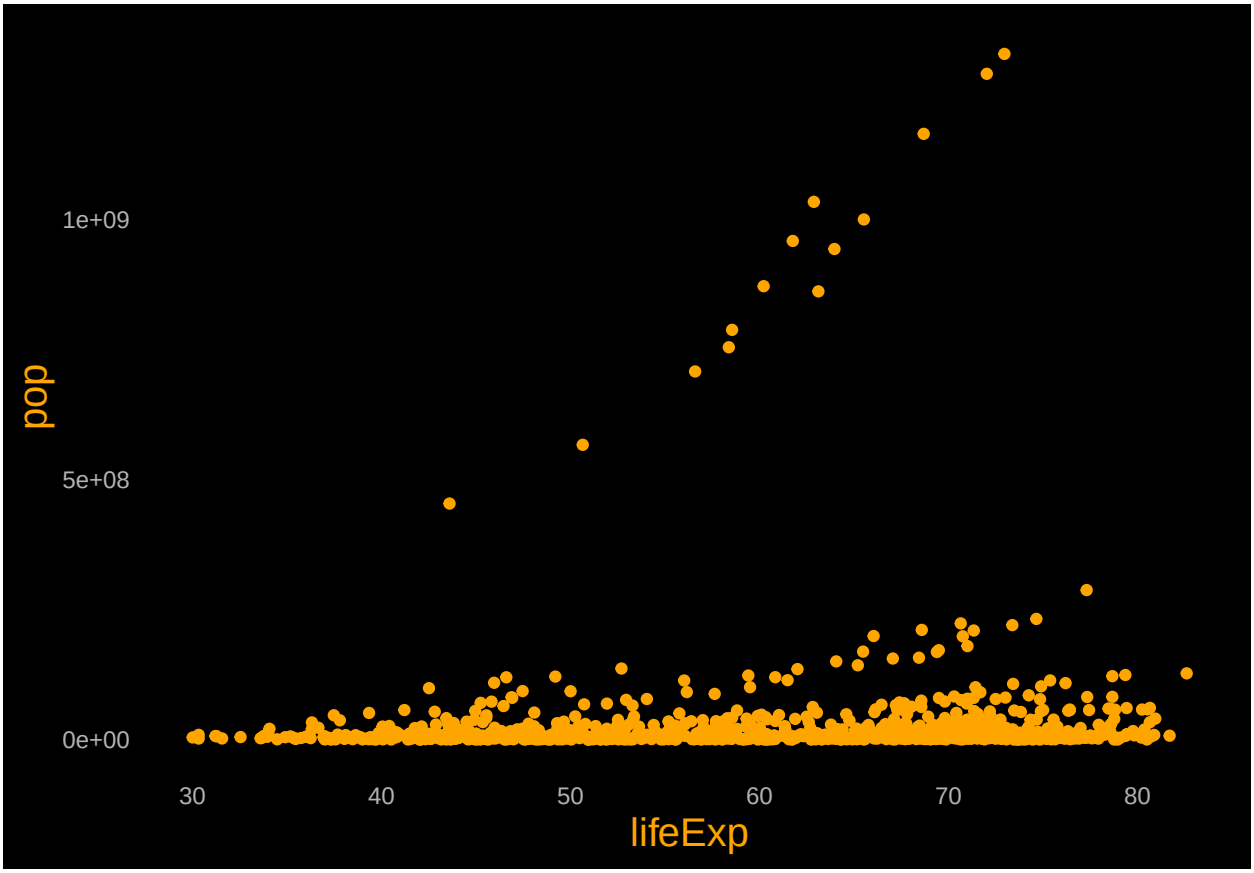


- #Another way to handle big datasets is to create a sample

```
gapminder_sample <- gapminder[sample(nrow(gapminder), 1000),]
```

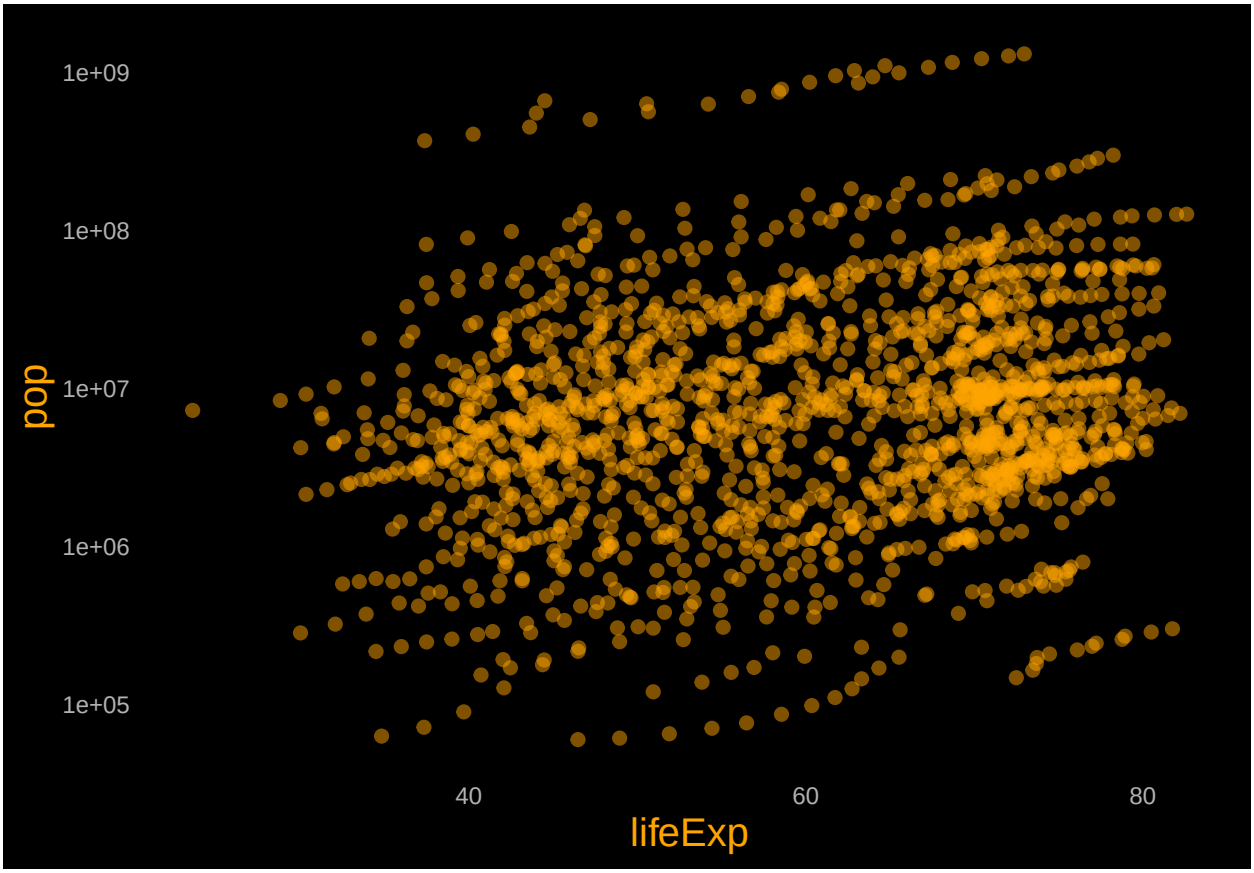
- #Basic scatter plot

```
ggplot(gapminder_sample, aes(x=lifeExp, y=pop)) +
  geom_point(color=trend_color)
```



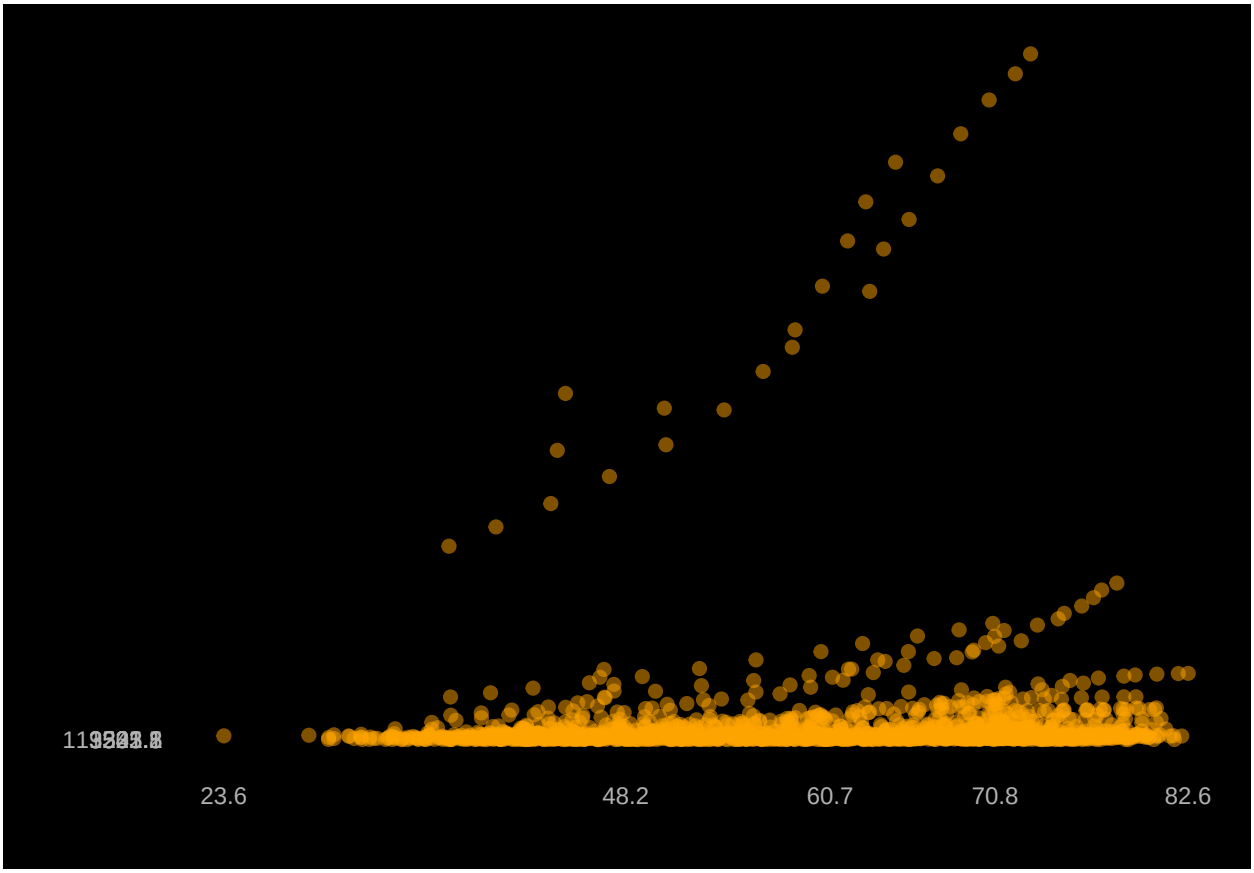
- #Change the position scale to logarithmic scaling

```
ggplot(gapminder, aes(x=lifeExp, y=pop)) +
  geom_point(size=2, alpha=0.5, color=trend_color) +
  scale_y_log10()
```



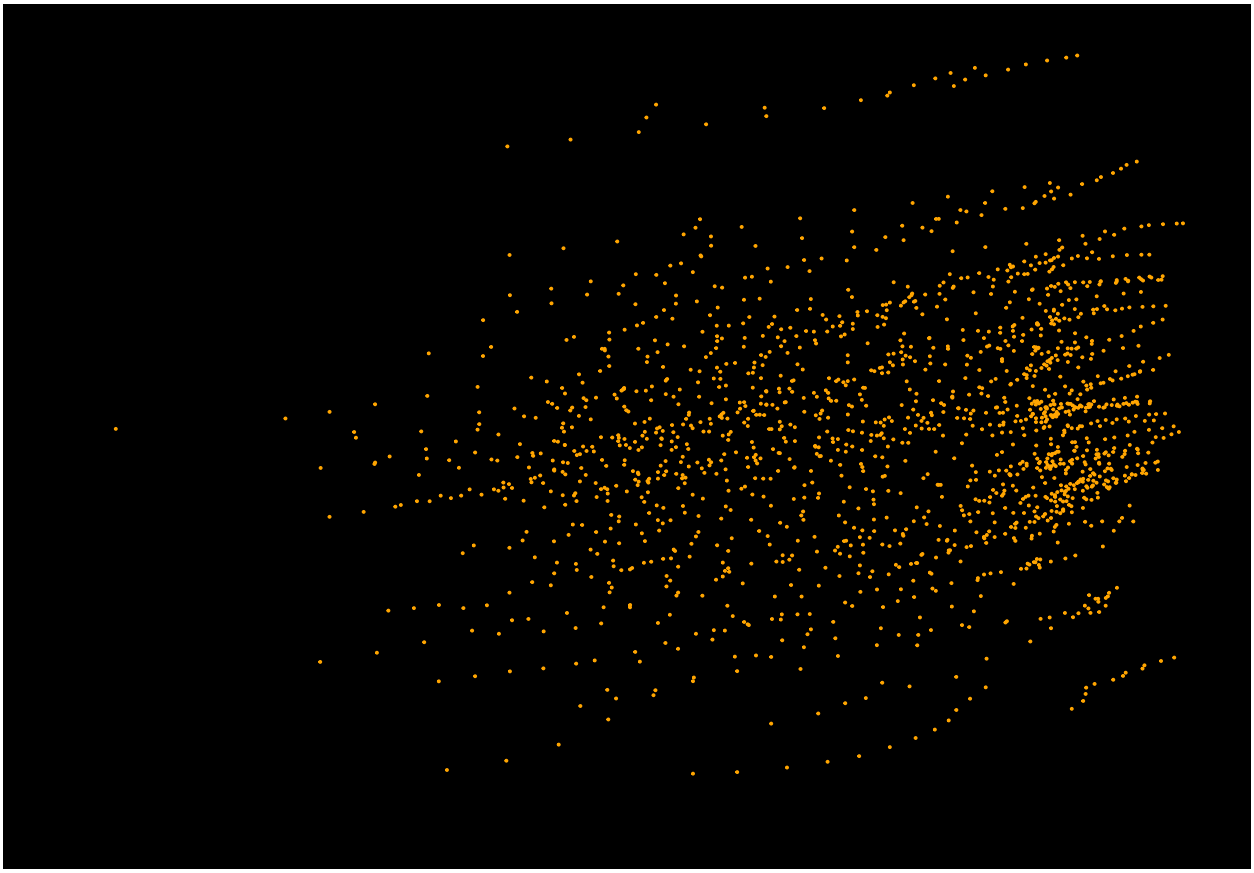
- #Axis labeling depending on the quantiles

```
ggplot(gapminder, aes(x=lifeExp, y=pop)) +
  geom_point(size=2, alpha=0.5, color=trend_color) +
  xlab("")+
  ylab("")+
  scale_x_continuous(breaks = round(as.vector(quantile(gapminder$lifeExp)), digits = 1))+
  scale_y_continuous(breaks = round(as.vector(quantile(gapminder$gdpPercap)), digits = 1))
```



- #Axis labeling depending on the quantiles for logaritmic scaling

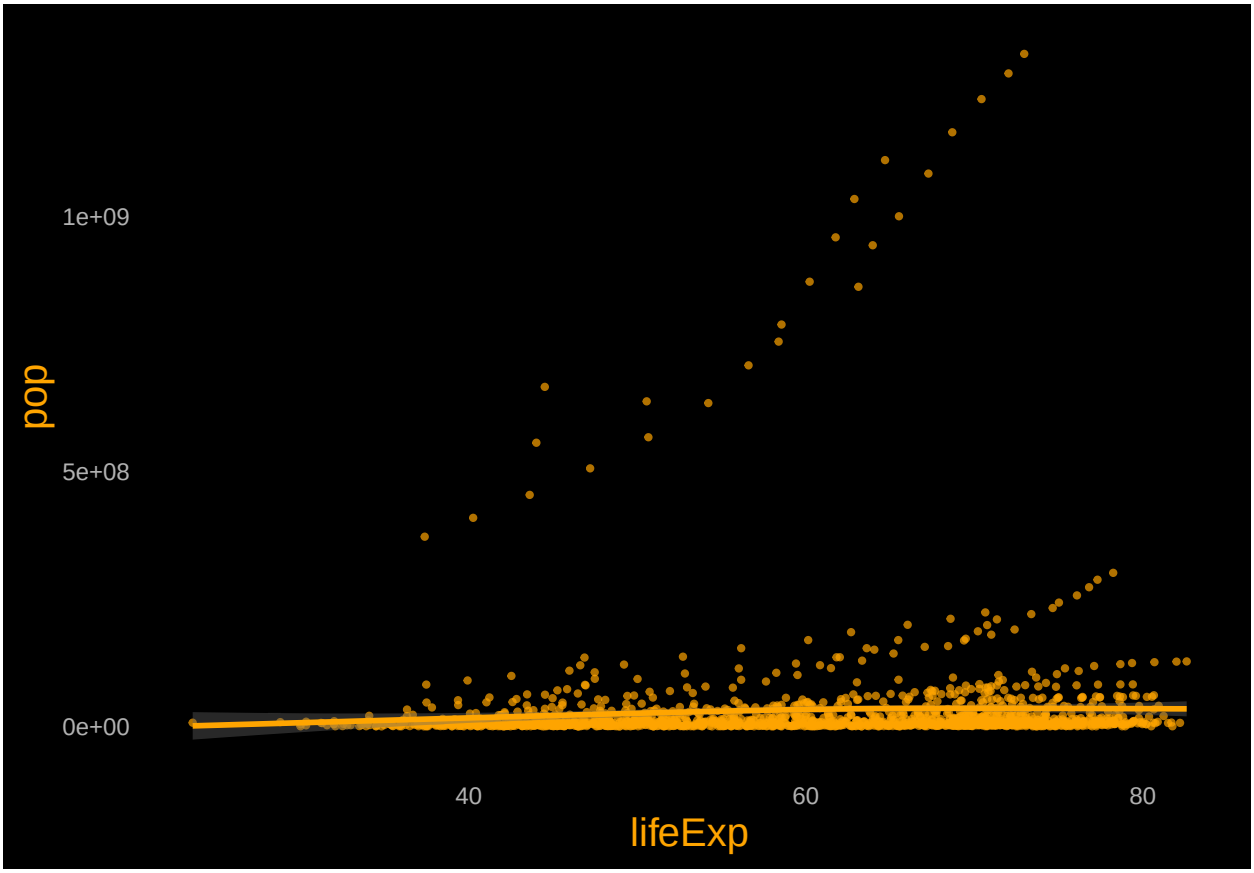
```
ggplot(gapminder, aes(x=lifeExp, y=pop)) +
  geom_point(size=0.02, alpha=1, color=trend_color)  +
  xlab("")+
  ylab("")+
  scale_x_log10(breaks = round(as.vector(quantile(diamonds$carat)), digits = 2))+
  scale_y_log10(breaks = round(as.vector(quantile(diamonds$price)), digits = 2))
```



- #Adding a trend line

```
ggplot(gapminder, aes(x=lifeExp, y=pop)) +
  geom_point(color=trend_color, size=0.8, alpha=0.7)+
  stat_smooth(color=trend_color)
```

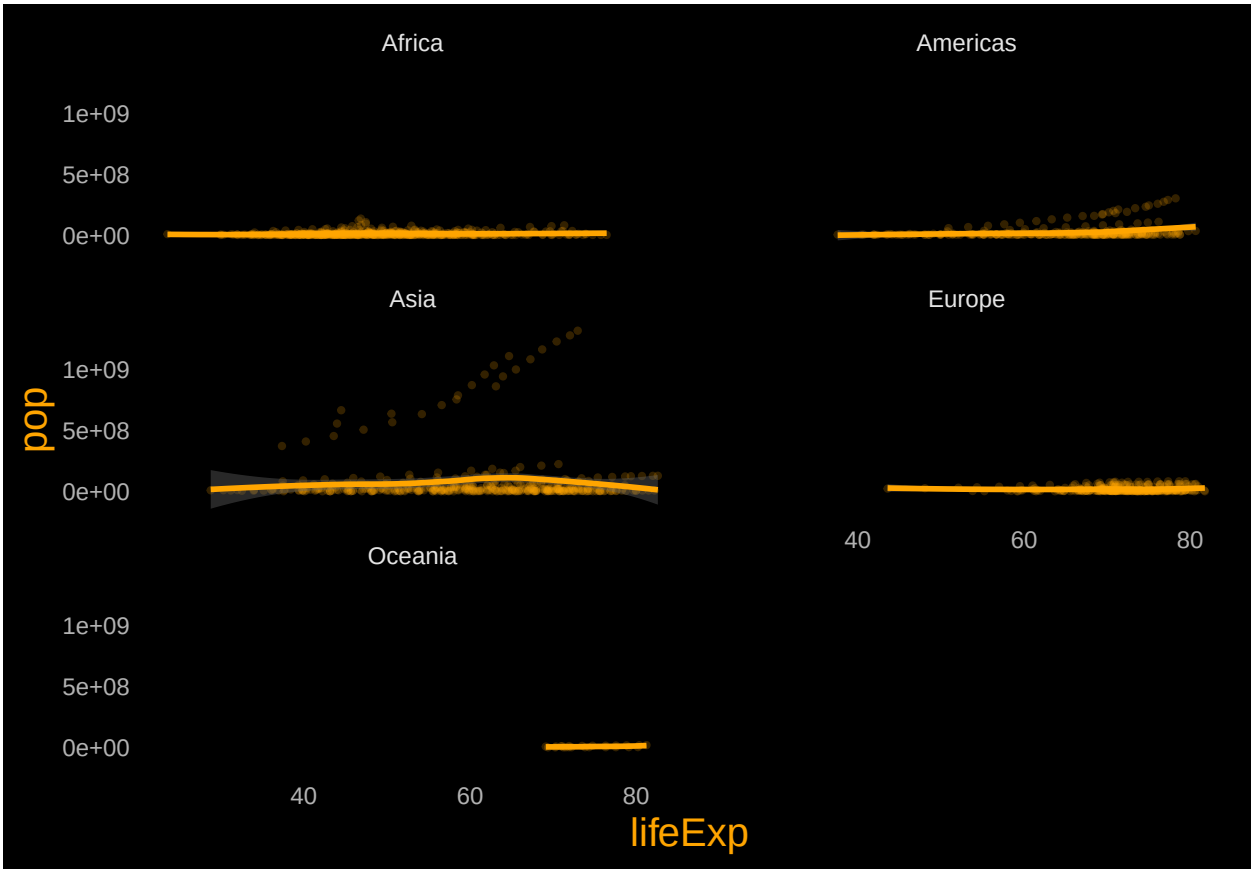
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



- #Small multiples - one variable

```
ggplot(gapminder, aes(x=lifeExp, y=pop)) +
  geom_point(color=trend_color, size=0.8, alpha=0.2)+
  facet_wrap( ~ continent, ncol=2) +
  stat_smooth(color=trend_color)
```

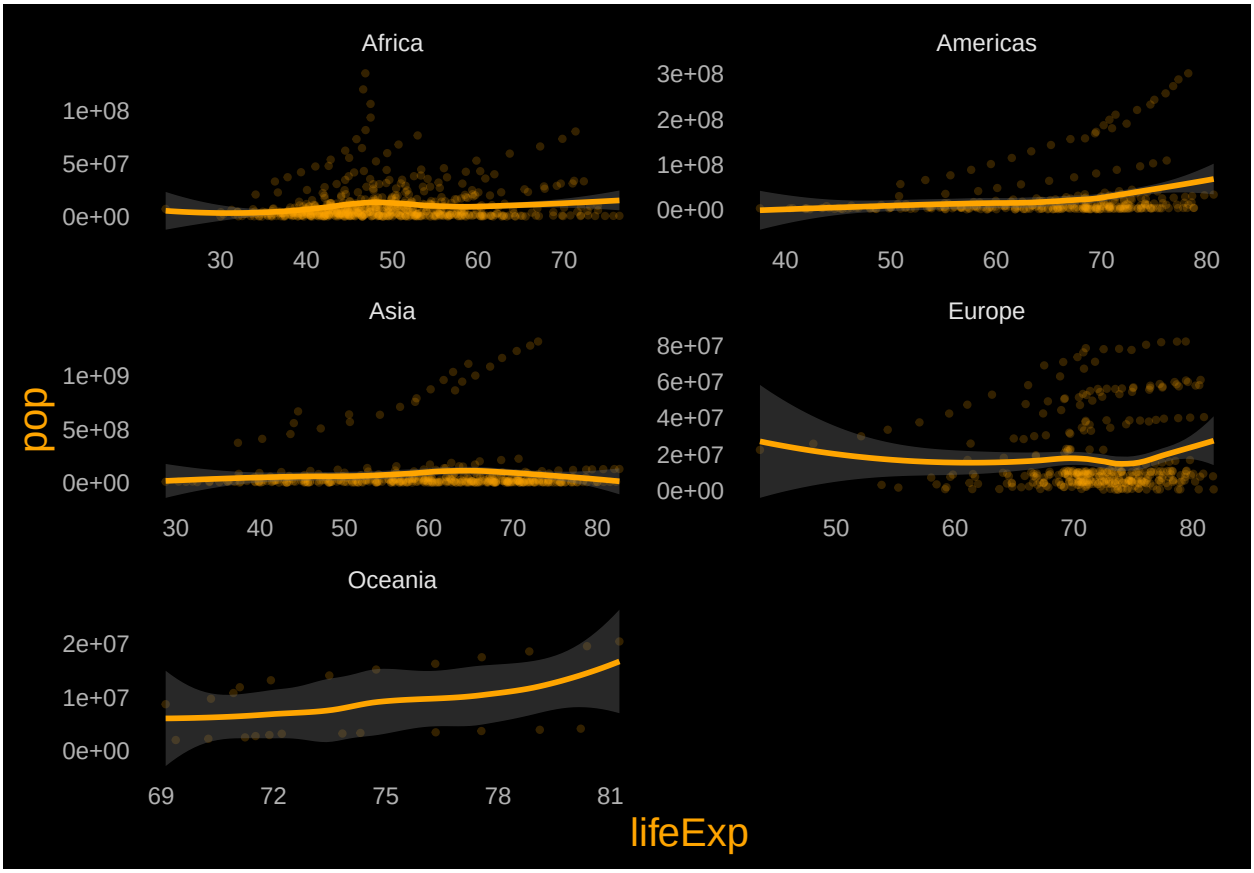
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



- #Small multiples - one variable with free scale

```
ggplot(gapminder, aes(x=lifeExp, y=pop)) +
  geom_point(color=trend_color, size=0.8, alpha=0.2)+
  facet_wrap( ~ continent, ncol=2, scales = "free") +
  stat_smooth(color=trend_color)
```

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

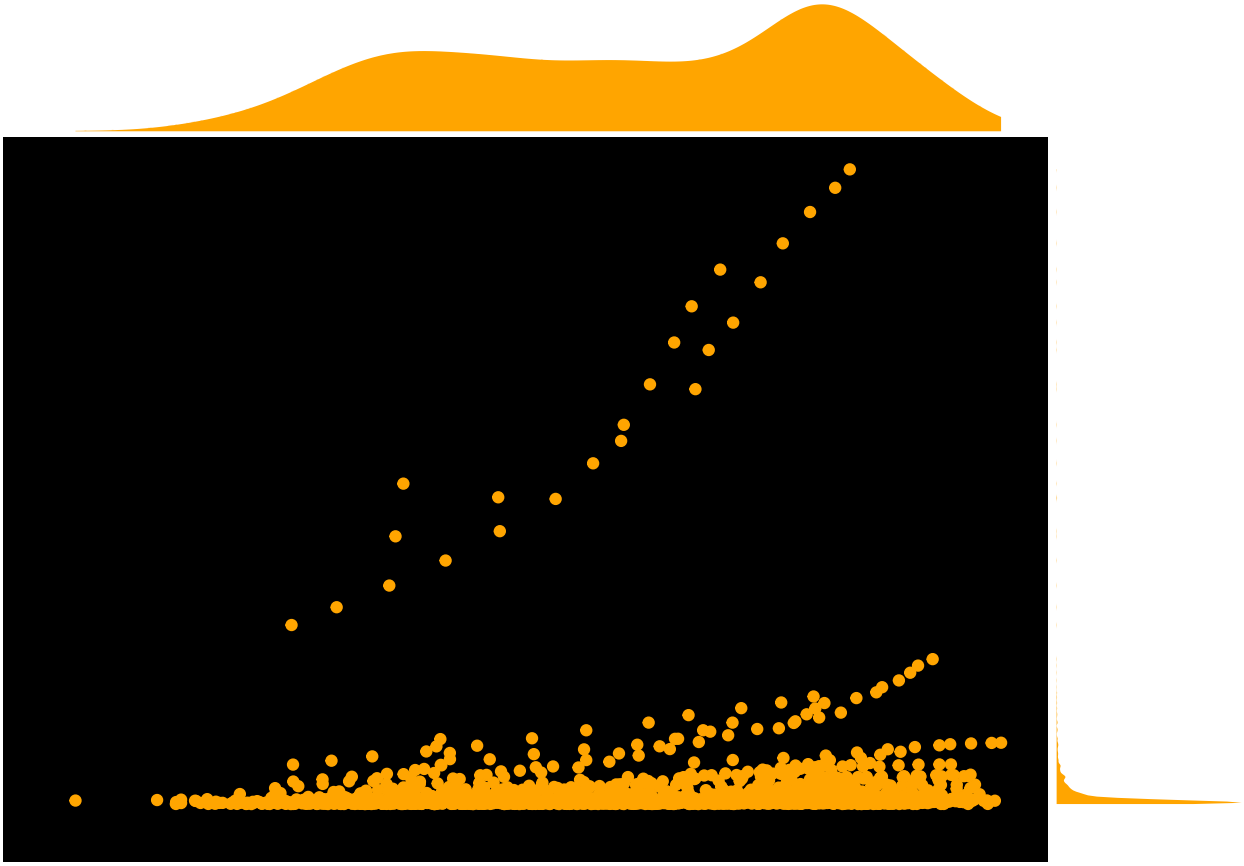


Exercise 7: - #Now we set the new defined theme to the default option

```
?ggMarginal
```

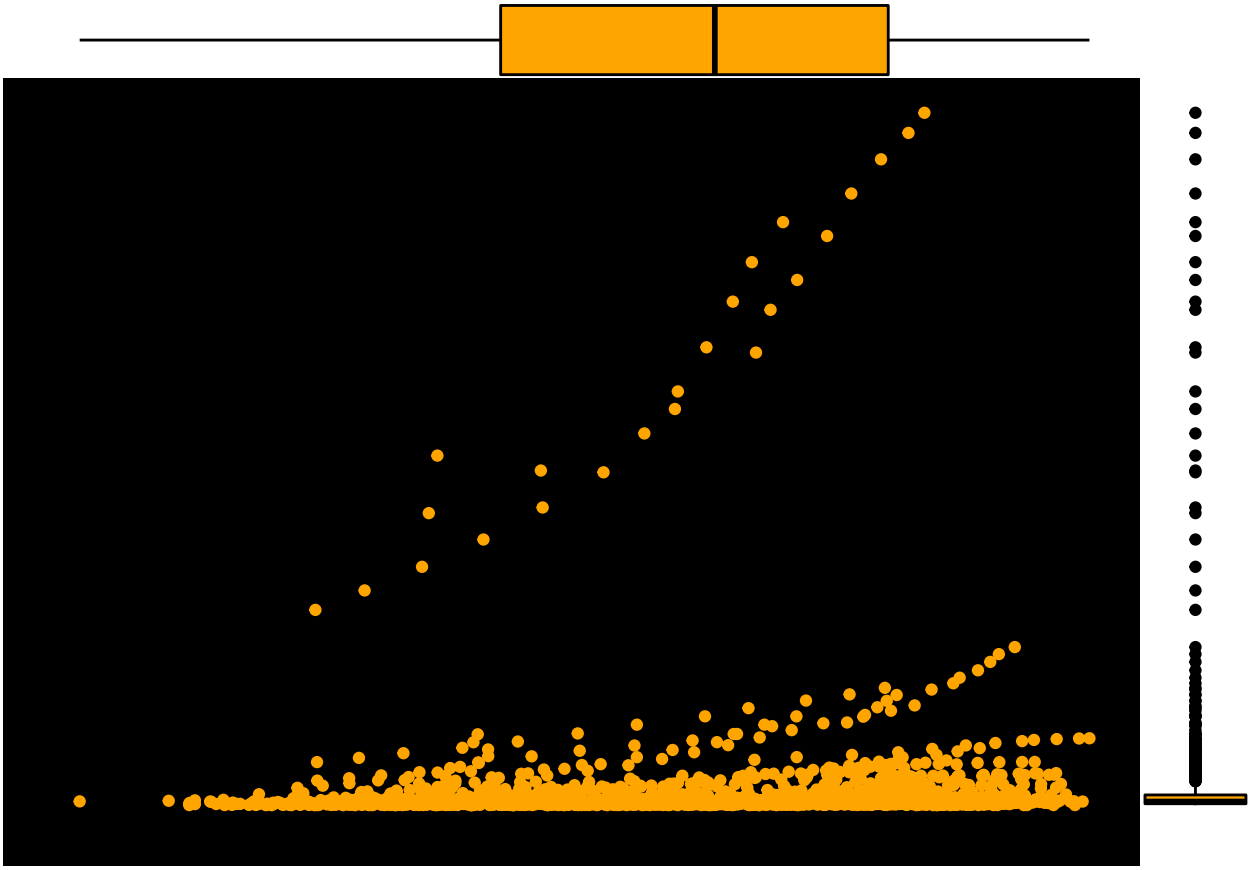
- #Density

```
pp <- ggplot(gapminder, aes(x=lifeExp, y=pop)) +
  geom_point(color=trend_color) +
  theme(axis.title=element_blank(), axis.text=element_blank())
ggMarginal(pp, type = "density", fill = trend_color, alpha=1, color='transparent')
```



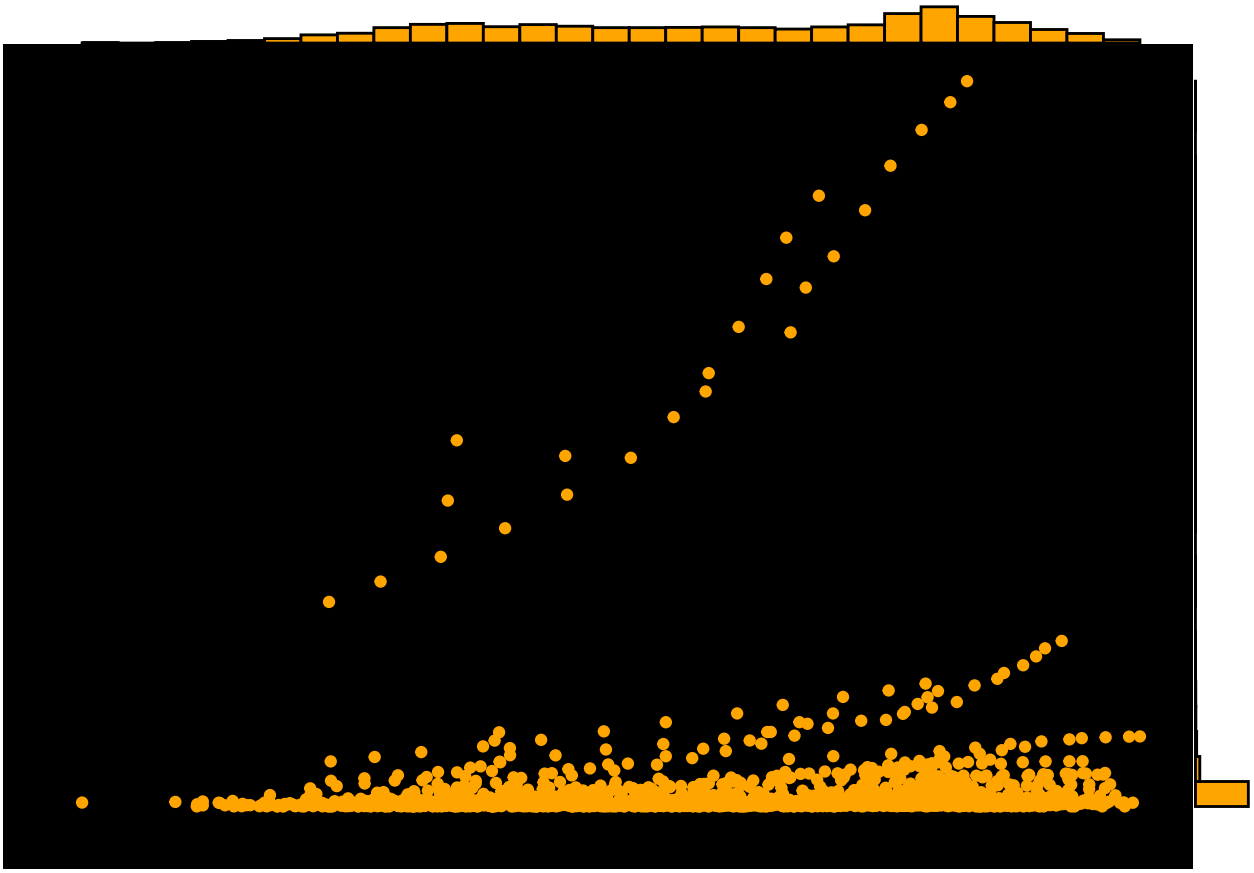
- #Box-plot

```
ggMarginal(pp, type = "boxplot", size=10, fill=trend_color)
```



- #Histogram

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



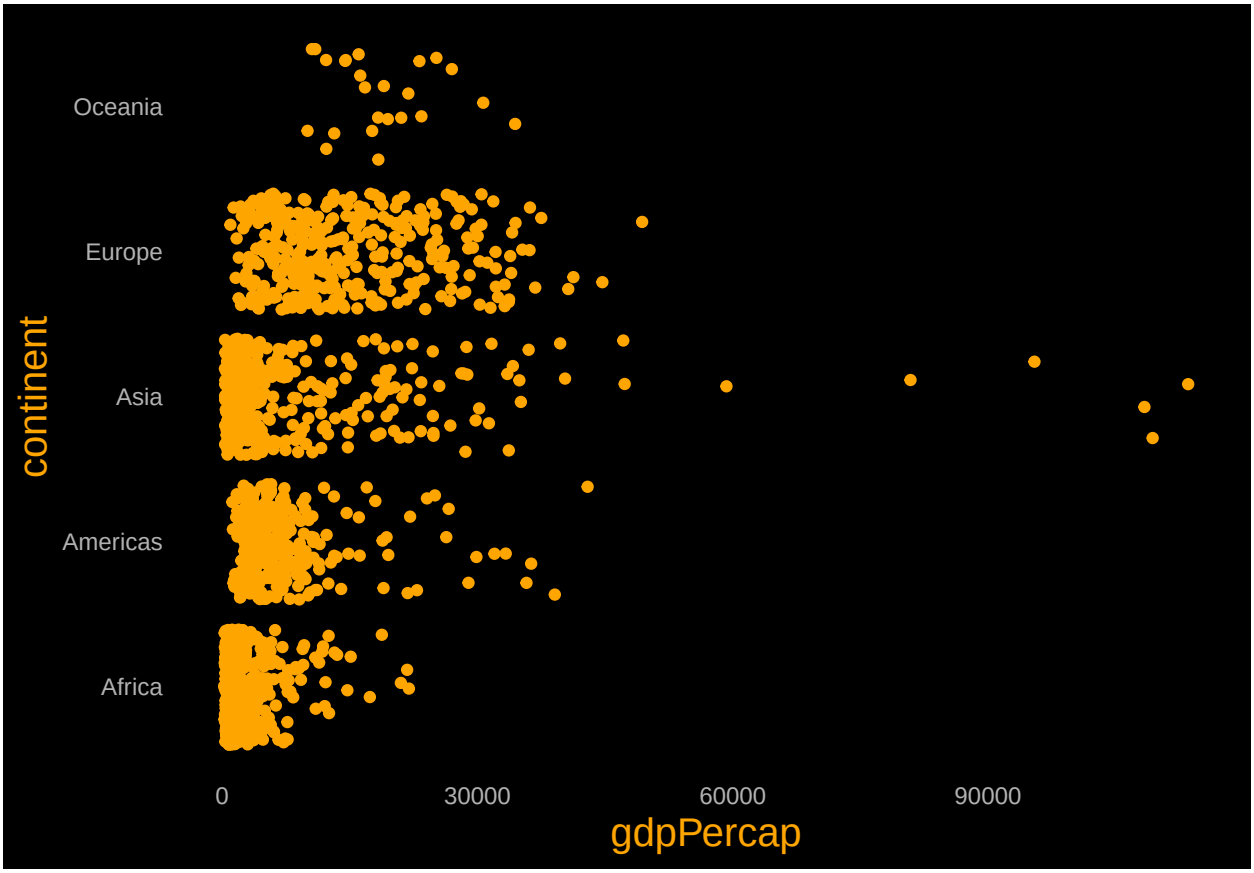
Exercise 8:

- #Beeswarm

```
?geom_jitter()
```

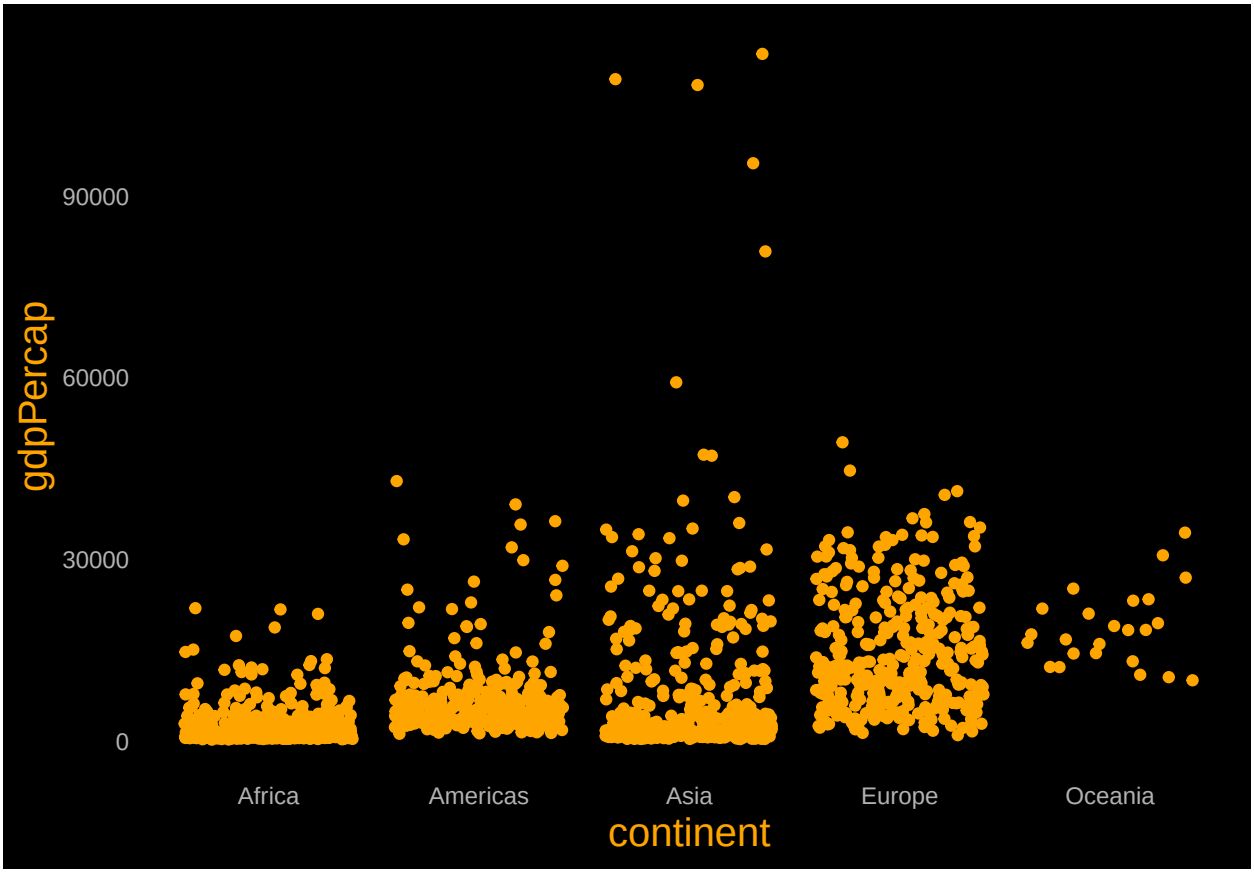
- #Simple jitter plot

```
ggplot(gapminder, aes(x=gdpPercap, y=continent)) +
  geom_jitter(color=trend_color)
```



- #Switching axis

```
ggplot(gapminder, aes(y=gdpPercap, x=continent)) +
  geom_jitter(color=trend_color)
```

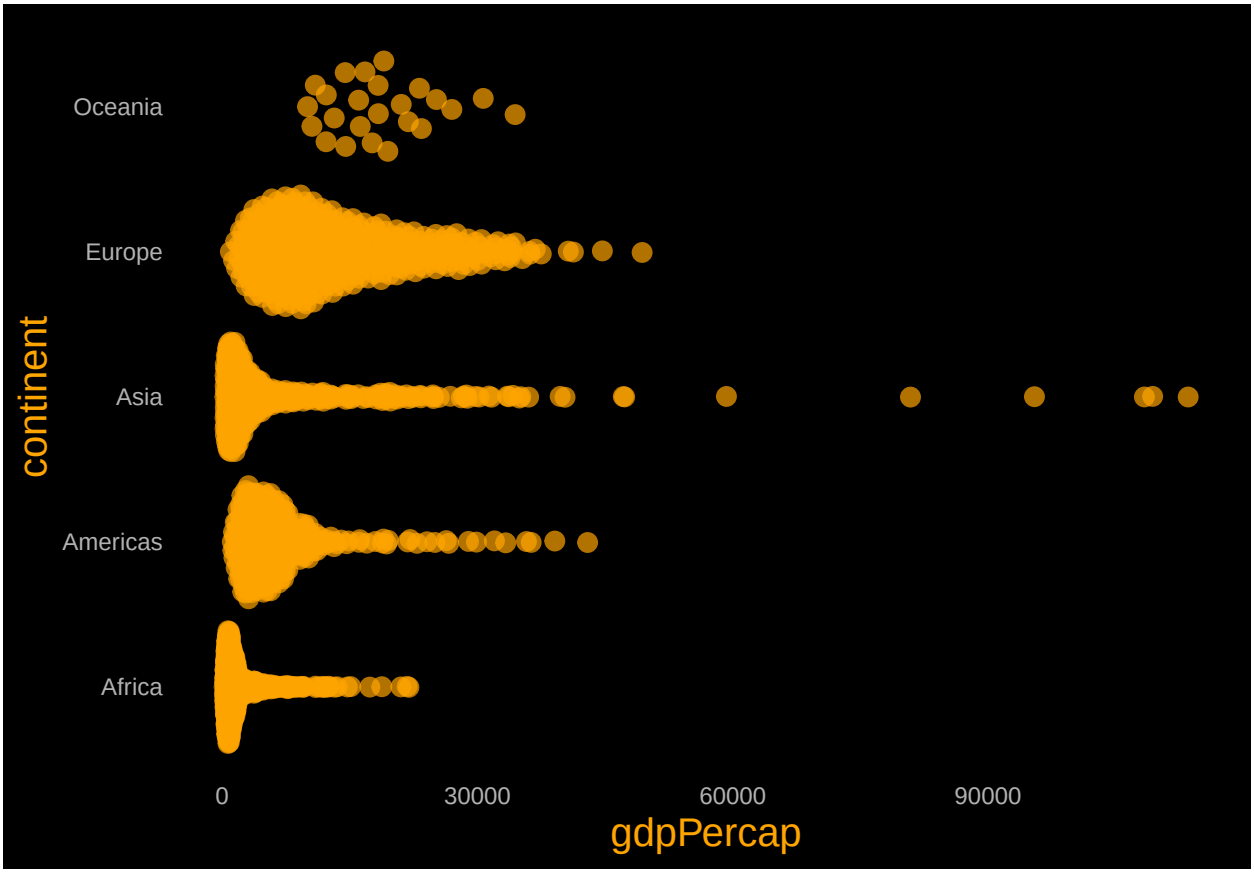


- #Check the options

```
?geom_quasirandom()
```

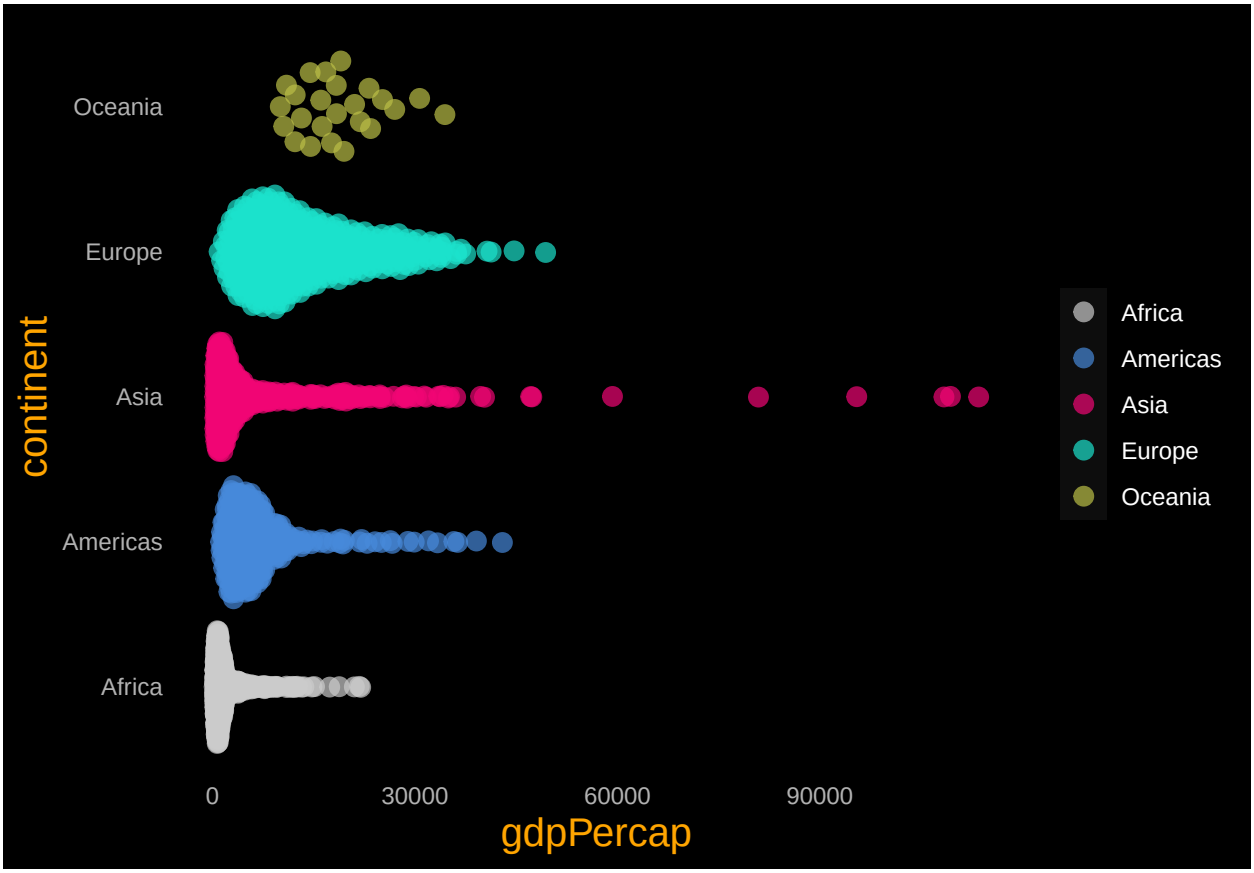
- #Simple beewswarm

```
ggplot(gapminder, aes(x=gdpPercap, y=continent)) +
  geom_quasirandom(size=3, alpha=0.7, color=trend_color, groupOnX=FALSE)
```



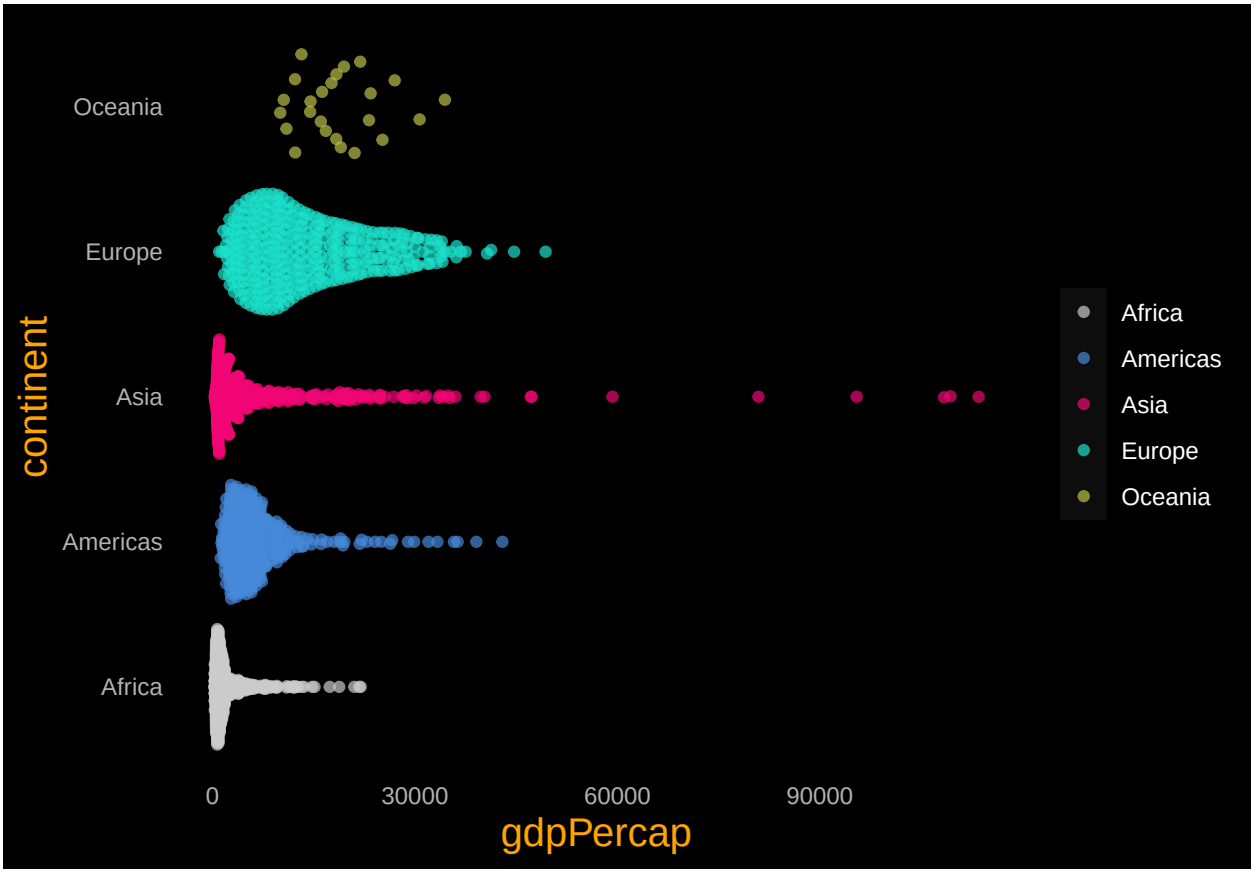
- #Simple beewswarm

```
ggplot(gapminder, aes(x=gdpPercap, y=continent, colour=continent)) +
  geom_quasirandom(size=3, alpha=0.7, groupOnX=FALSE) +
  scale_colour_manual(values=c("#cccccc", "#478adb", "#f20675", "#1ce3cd", "#bcc048"))
```



- #Simple beewswarm

```
ggplot(gapminder, aes(x=gdpPercap, y=continent, colour=continent)) +
  geom_quasirandom(alpha=0.7, groupOnX=FALSE, method = "smiley") +
  scale_colour_manual(values=c("#cccccc", "#478adb", "#f20675", "#1ce3cd", "#bcc048"))
```

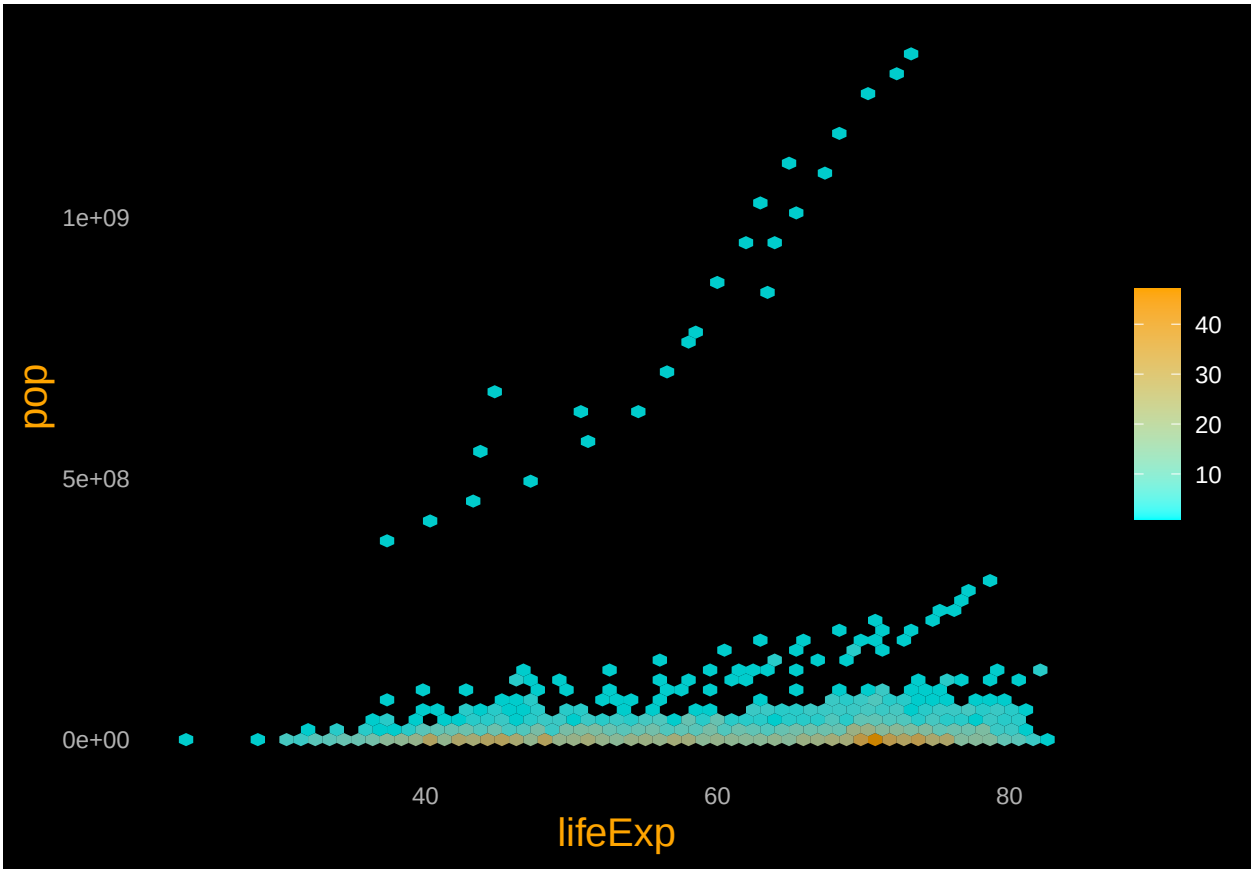


- #Exercise 9:
- #Hexagonal binning
- #Checking the options

```
?geom_hex
```

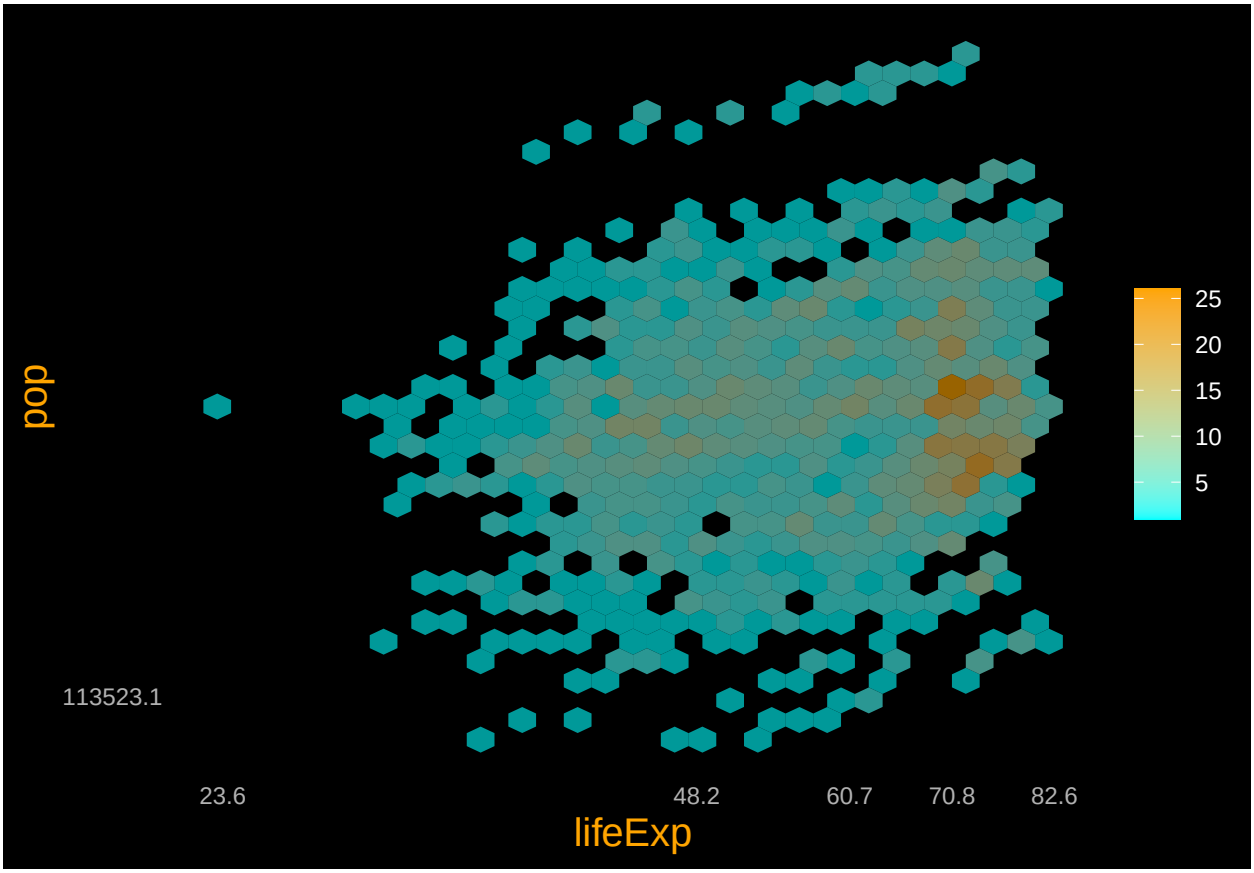
- #Aggregation through hexagonal binning - defining the number of bins

```
ggplot(gapminder, aes(x=lifeExp, y=pop)) +
  geom_hex(bins=60, alpha =0.8)+
  scale_fill_gradient(low="cyan", high=trend_color)
```



- #Aggregation through hexagonal binning - logarithmic scaling

```
ggplot(gapminder, aes(x=lifeExp, y=pop)) +
  geom_hex(alpha = 0.6) +
  scale_x_log10(breaks = round(as.vector(quantile(gapminder$lifeExp)), digits = 1))+
  scale_y_log10(breaks = round(as.vector(quantile(gapminder$gdpPercap)), digits = 1))+
  scale_fill_gradient(low="cyan", high=trend_color)
```

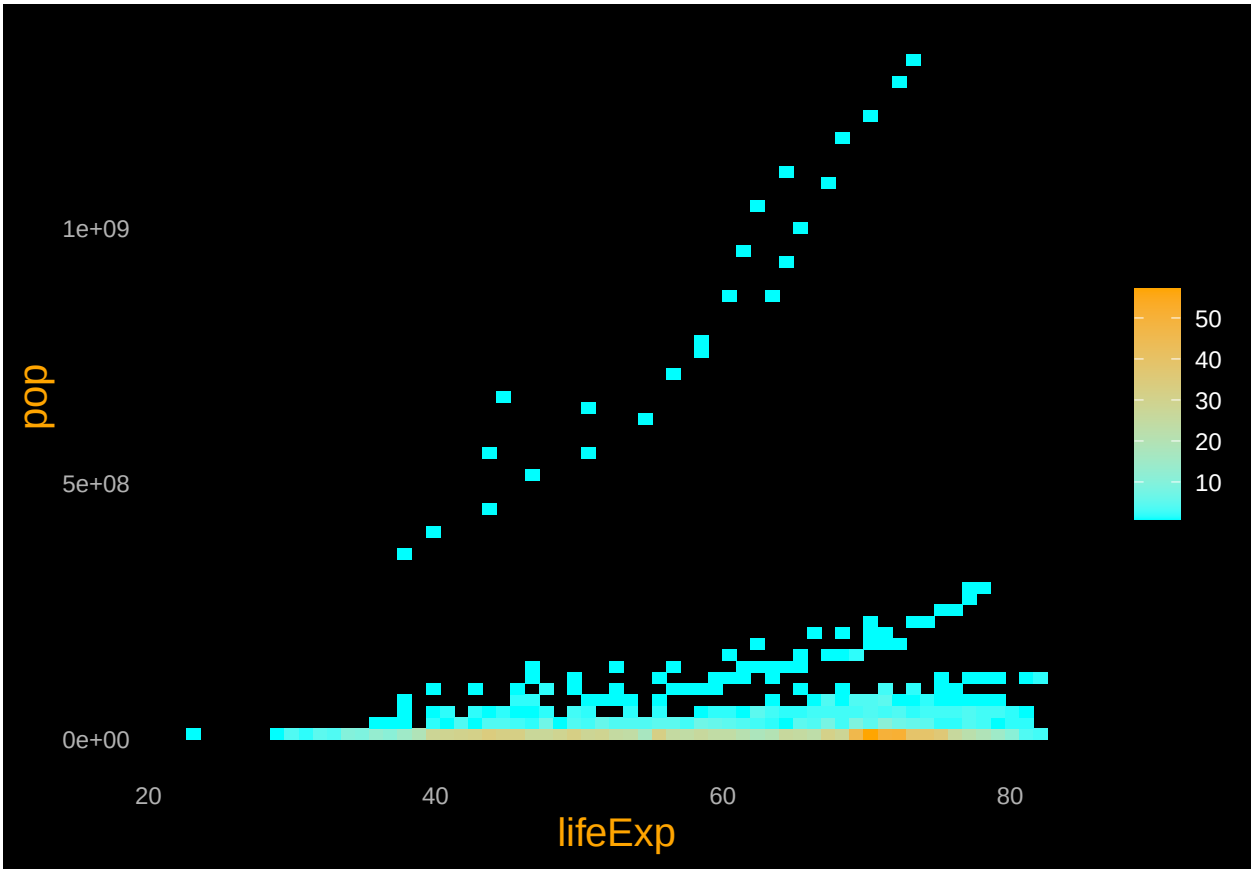


- #Checking the options

```
?geom_bin2d
```

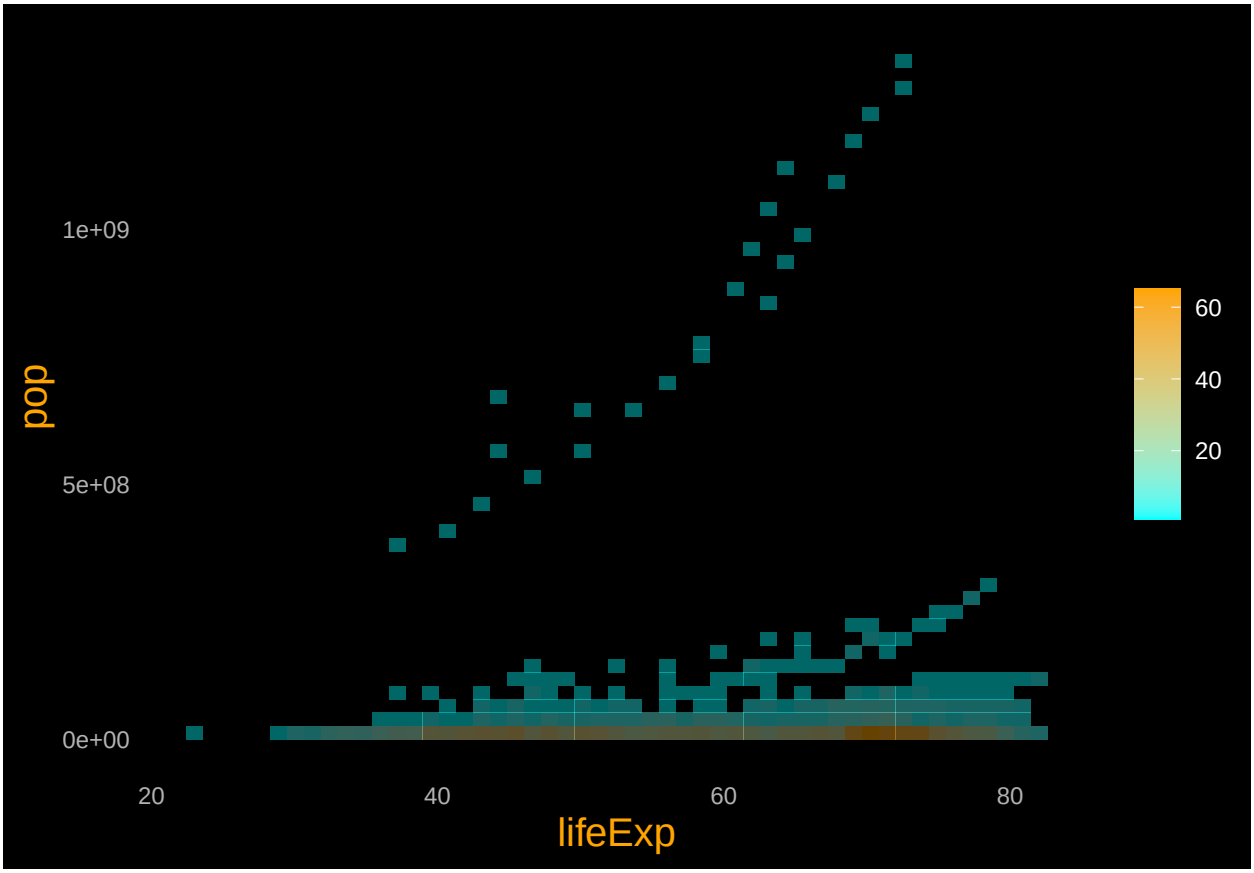
- #Heatmap based on rectangles

```
ggplot(gapminder, aes(x=lifeExp, y=pop)) +
  geom_bin2d(bins = 60) +
  scale_fill_gradient(low="cyan", high=trend_color)
```



- #Heatmap based on rectangles

```
ggplot(gapminder, aes(x=lifeExp, y=pop)) +
  geom_bin2d(bins = 50, alpha = 0.4)+
  scale_fill_gradient(low="cyan", high=trend_color)
```

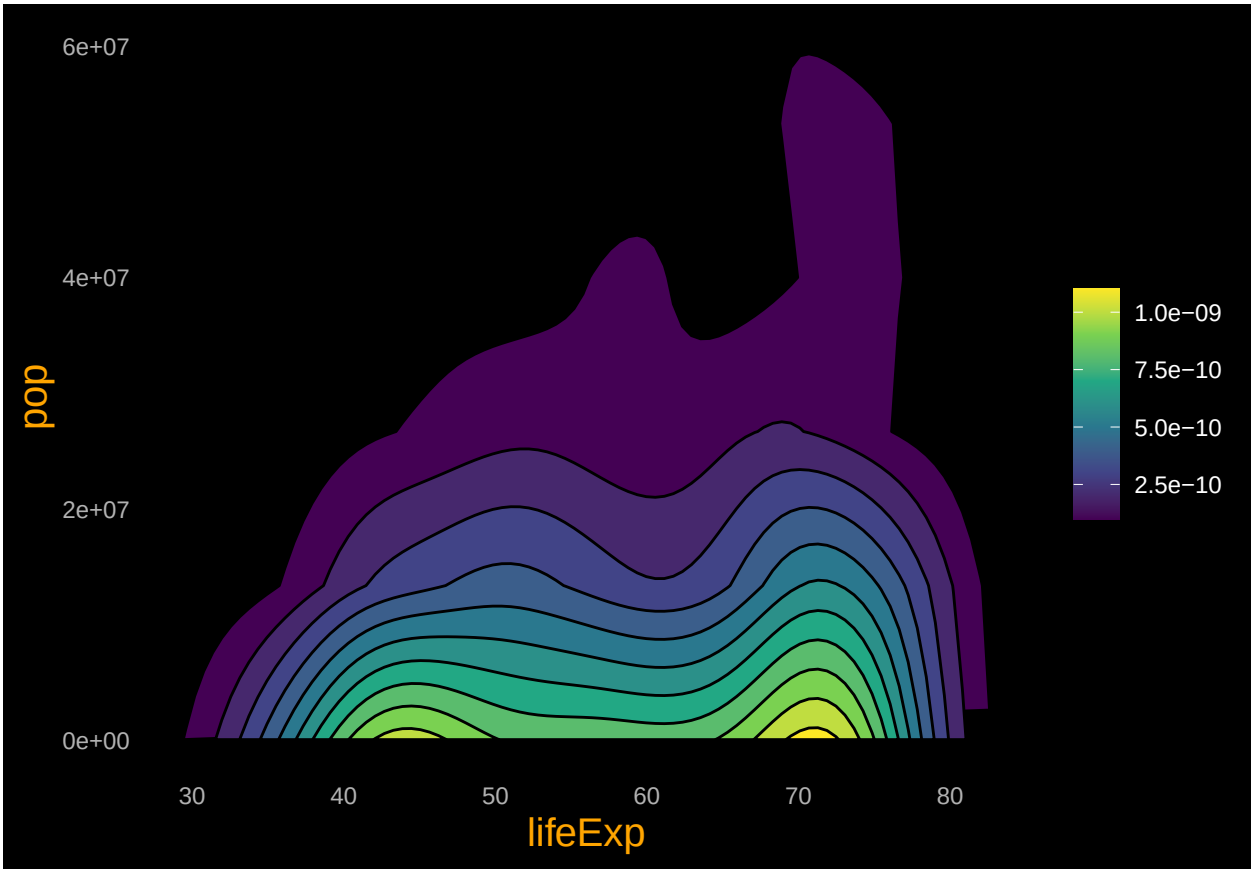


- #Checking the options

```
?stat_density_2d
```

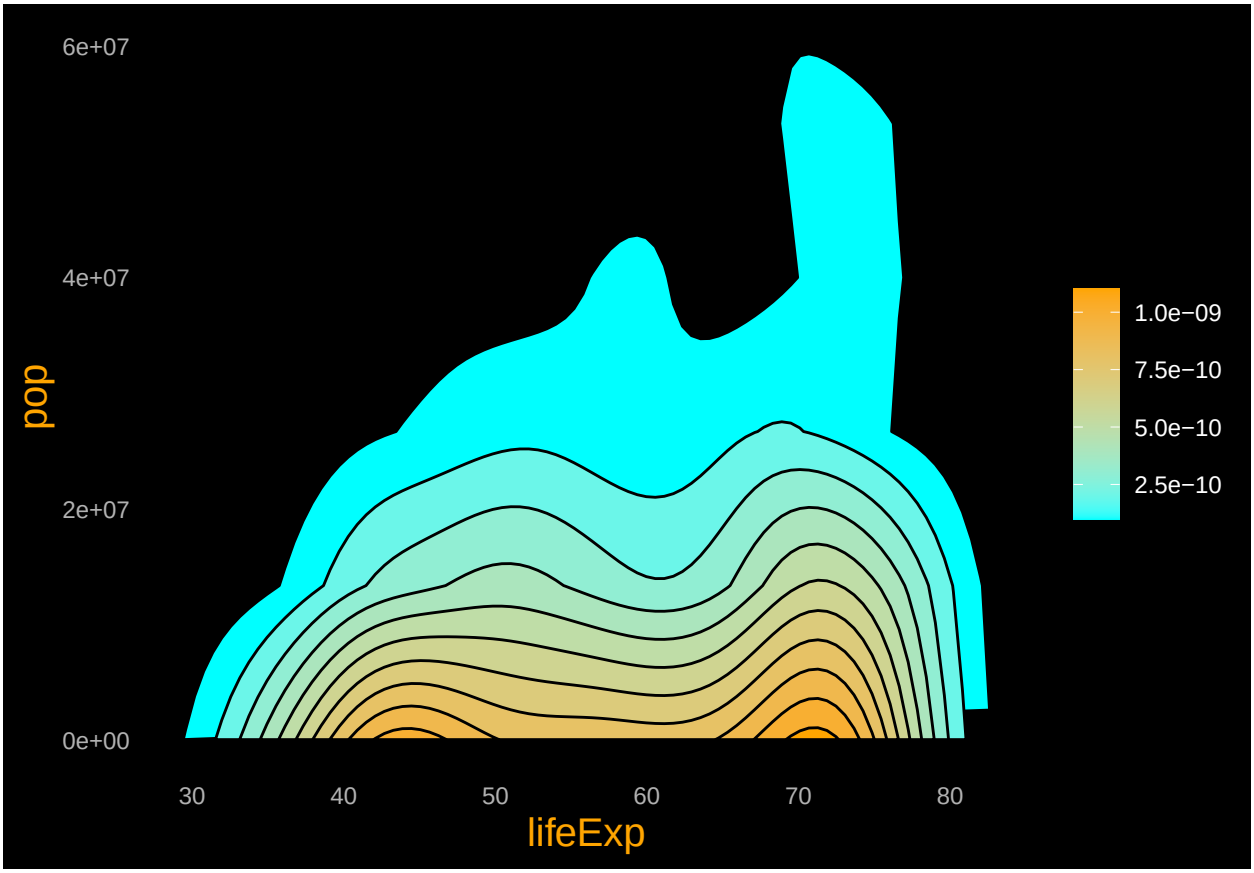
- #Density estimation with contours

```
ggplot(gapminder, aes(x=lifeExp, y=pop)) +
  stat_density_2d(aes(fill = ..level..), geom = "polygon")  +
  scale_fill_continuous(type = "viridis")
```



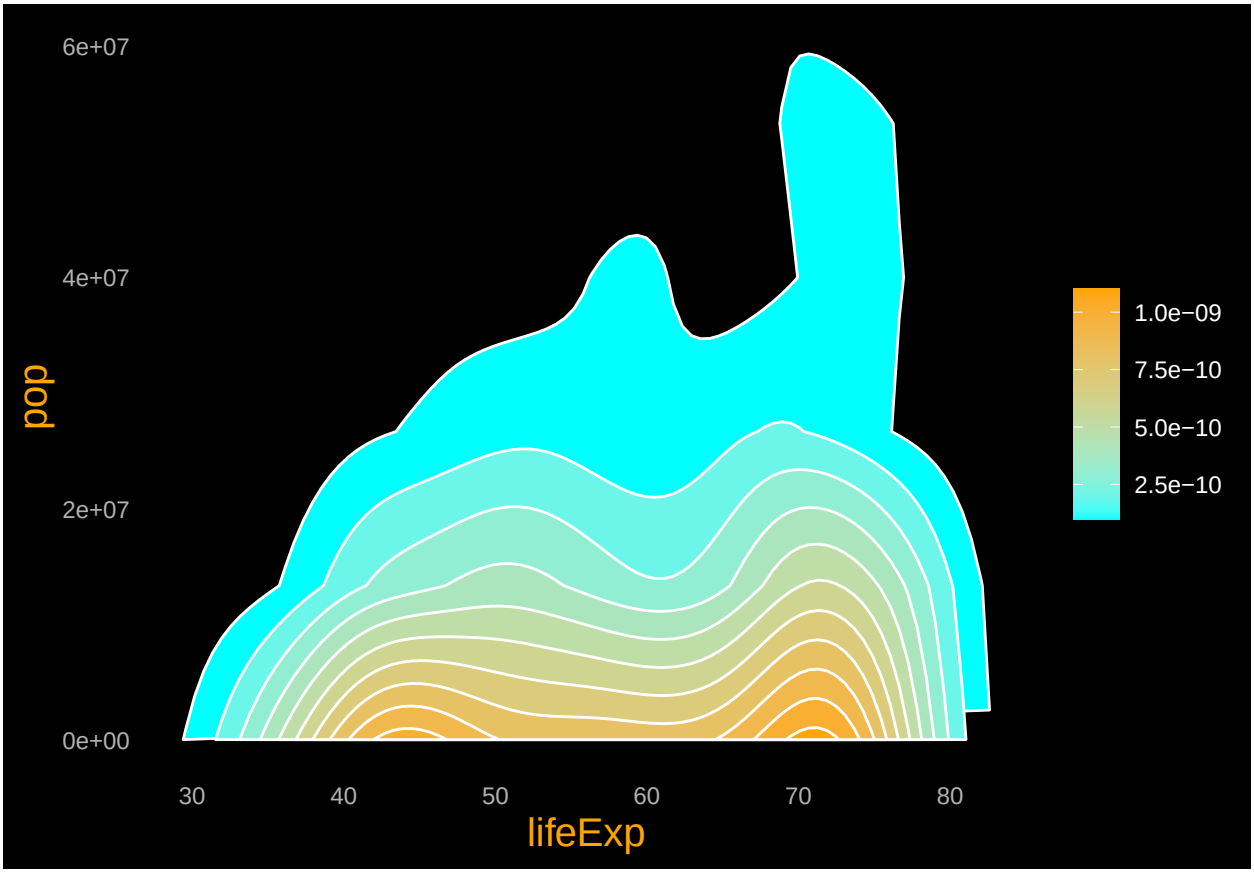
- #Density estimation with contours

```
ggplot(gapminder, aes(x=lifeExp, y=pop)) +
  stat_density_2d(aes(fill = ..level..), geom = "polygon") +
  scale_fill_gradient(low="cyan", high=trend_color)
```



- #Adding a stroke

```
ggplot(gapminder, aes(x=lifeExp, y=pop)) +
  stat_density_2d(aes(fill = ..level..), geom = "polygon", colour="white") +
  scale_fill_gradient(low="cyan", high=trend_color)
```



- #Exercise 10:
- #Scales
- #Check on 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.
```

```
str(gapminder)
```

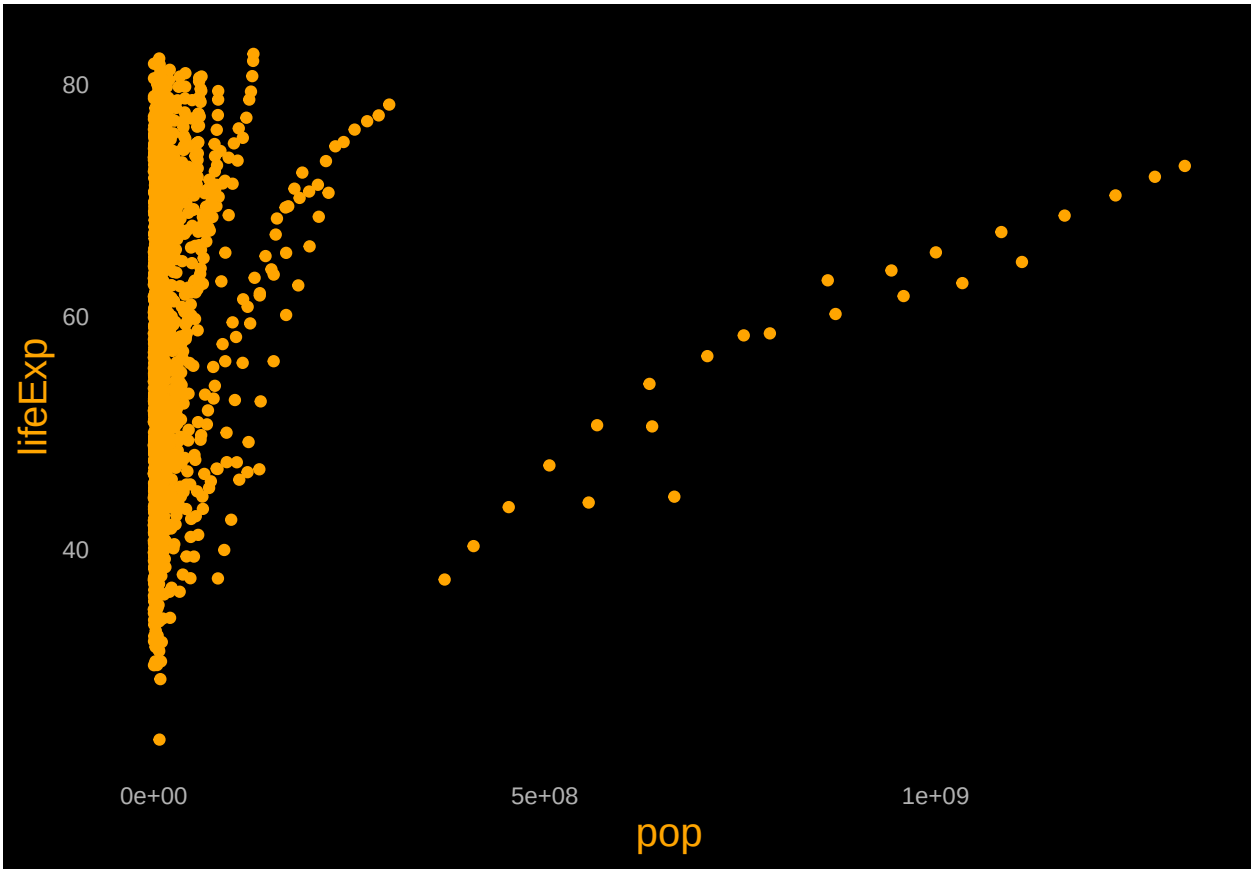
```
## tibble [1,704 x 6] (S3: tbl_df/tbl/data.frame)
##  $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ year     : int [1:1704] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...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

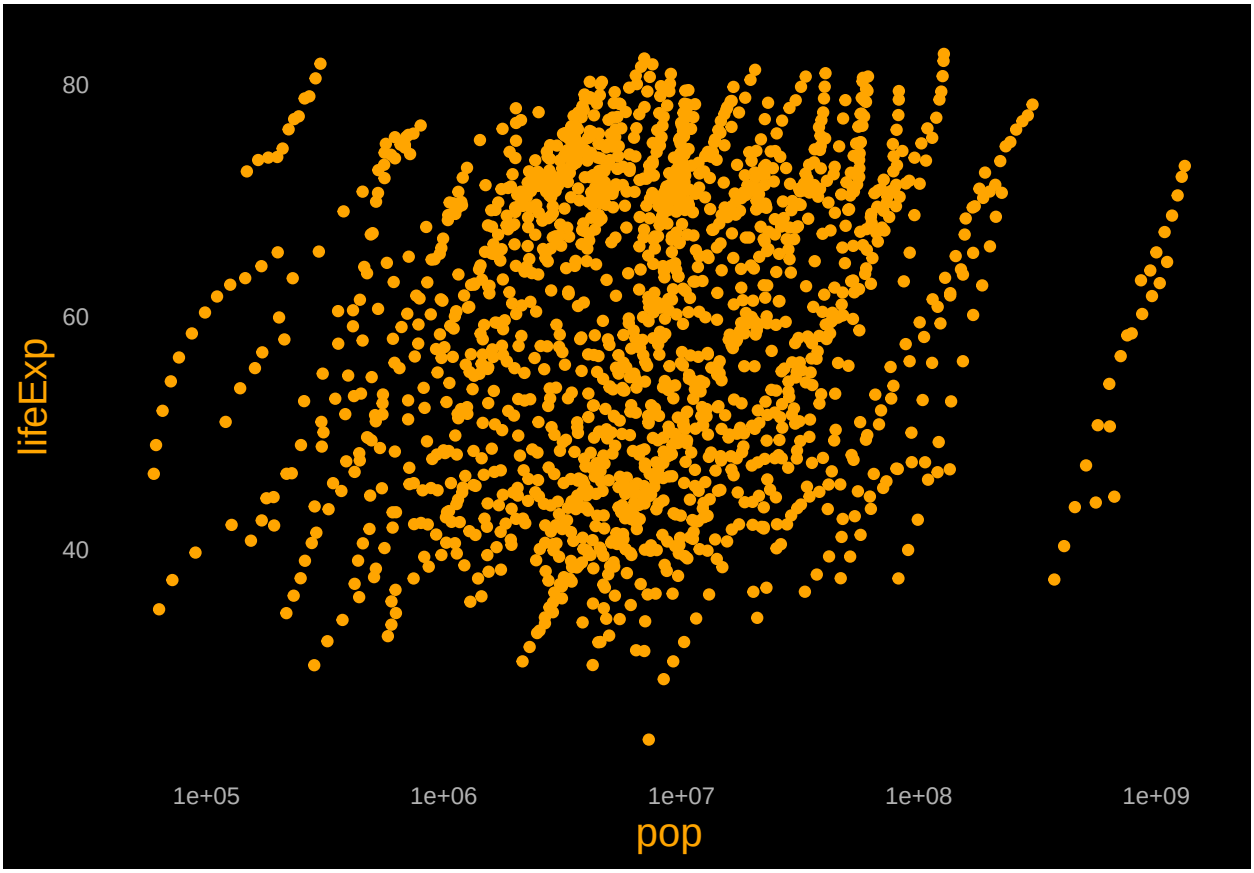
- #General scatter plot

```
ggplot(gapminder, aes(pop, lifeExp)) +
  geom_point(colour = trend_color)
```



- #Apply a log scale to the X axis postiion

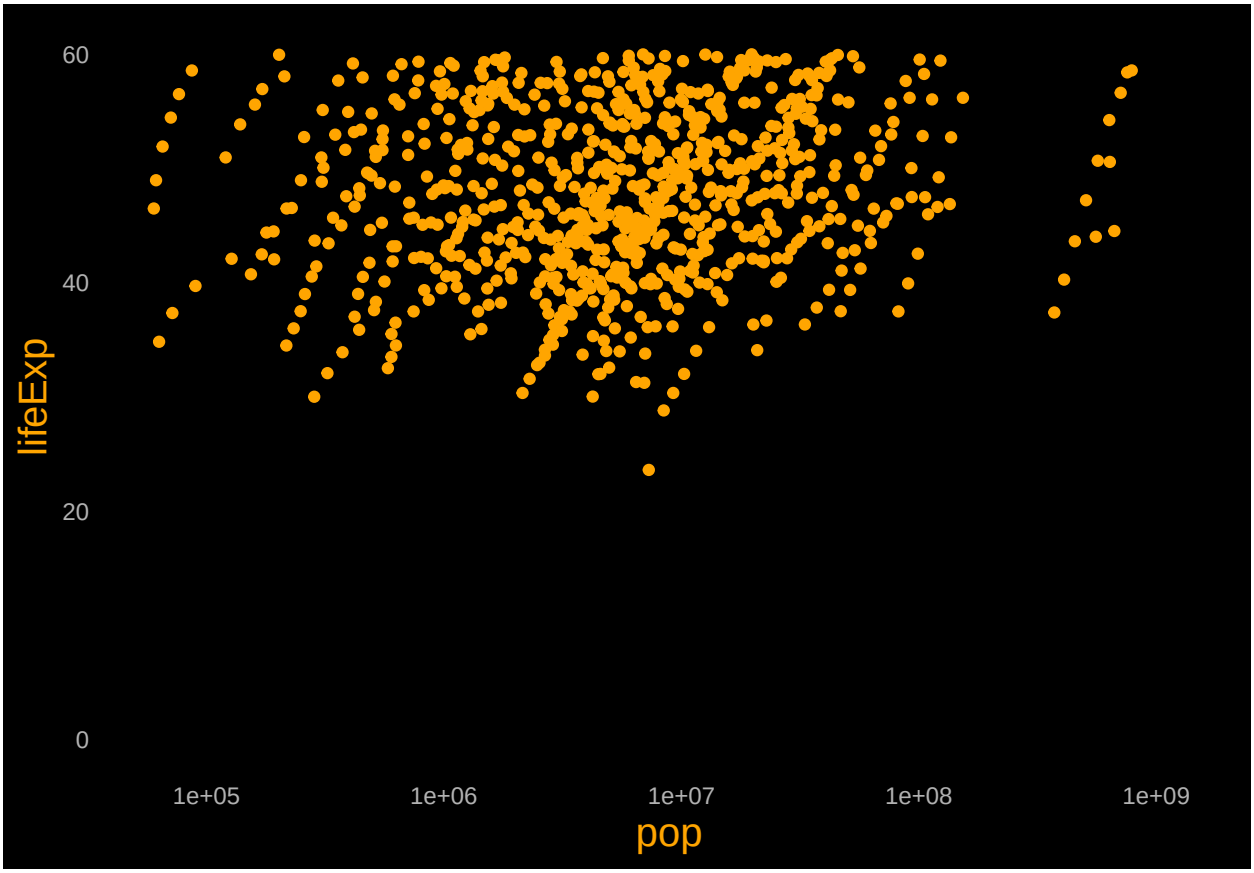
```
ggplot(gapminder, aes(pop, lifeExp)) +
  geom_point(colour = trend_color) +
  scale_x_log10()
```



- #Apply a a linear transformation to the Y axis position with limits

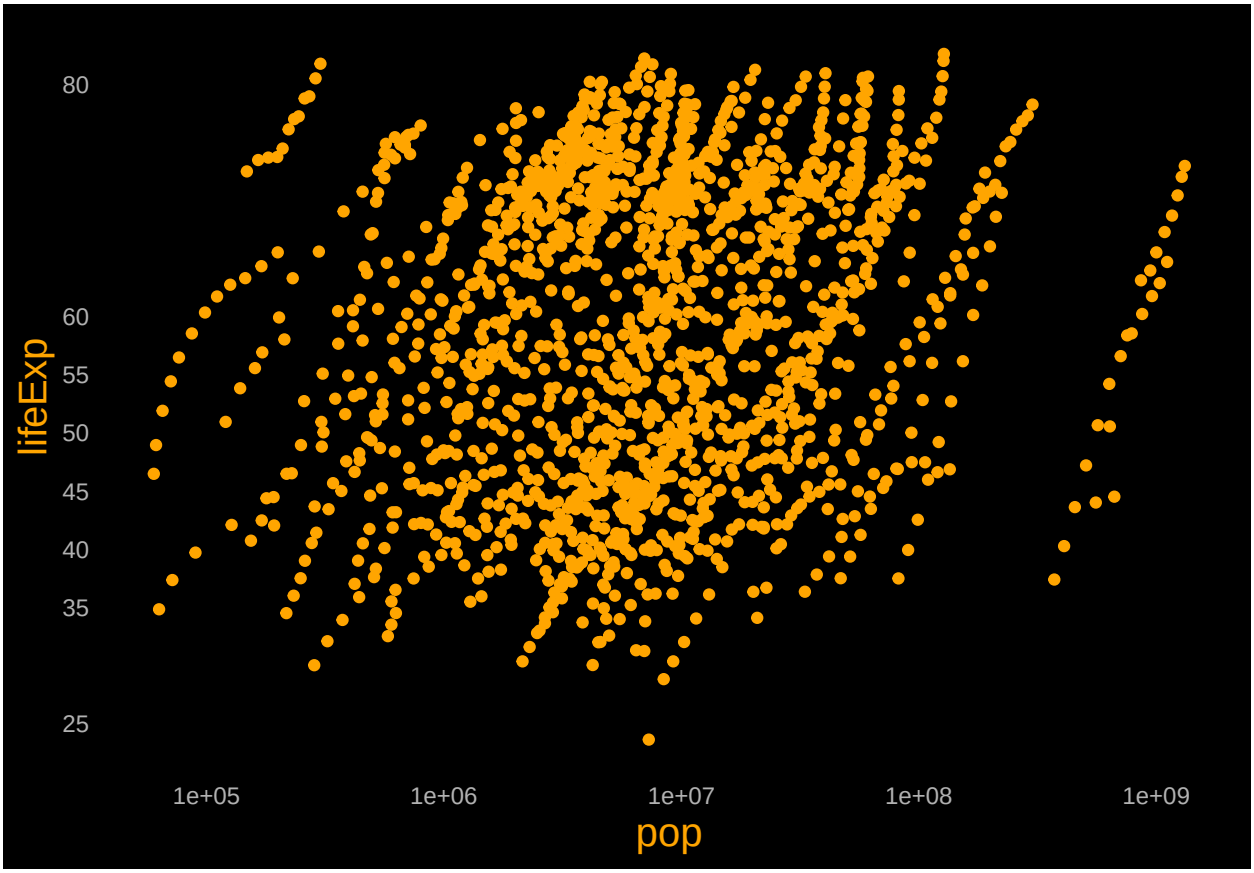
```
ggplot(gapminder, aes(pop, lifeExp)) +
  geom_point(colour = trend_color) +
  scale_x_log10() +
  scale_y_continuous(limits = c(0, 60))
```

```
## Warning: Removed 877 rows containing missing values (geom_point).
```



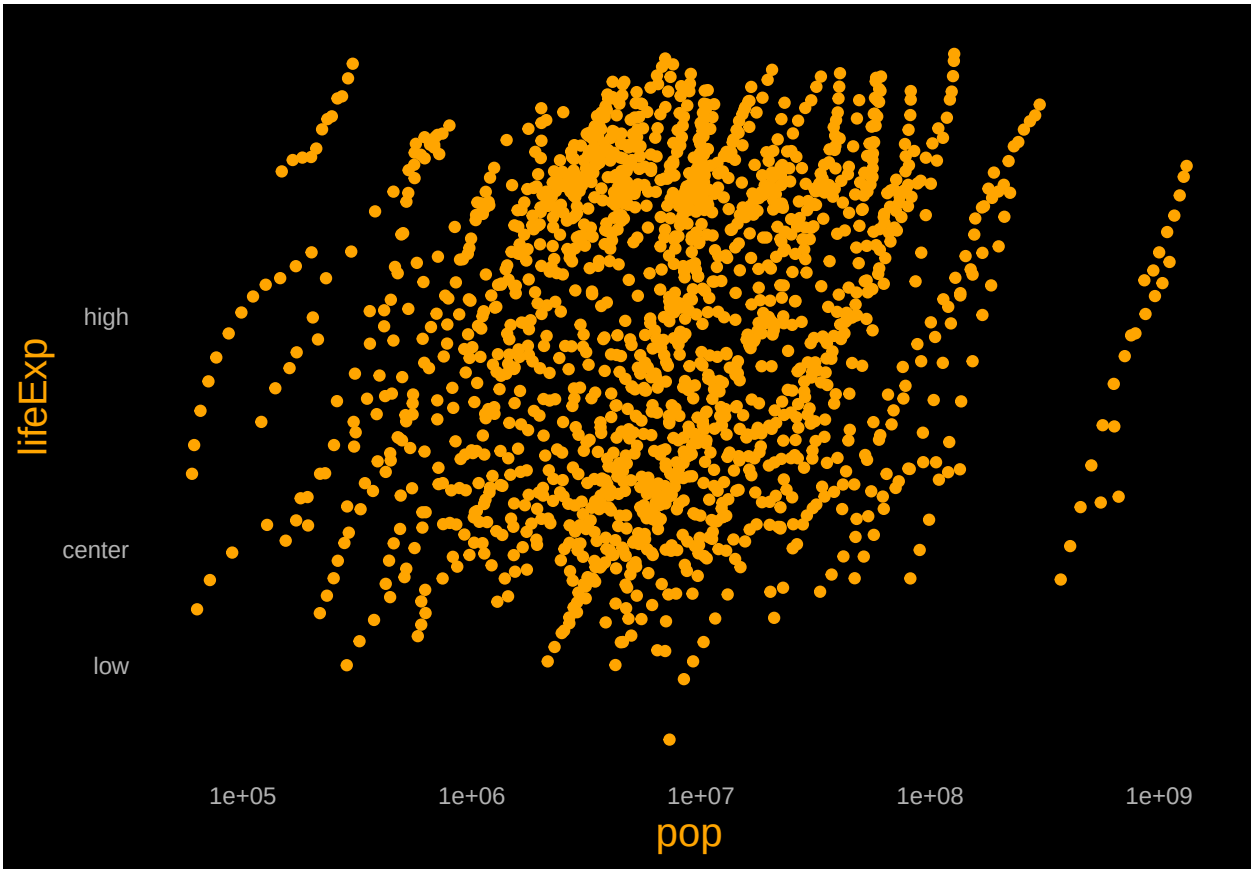
- #Apply a a linear transformation to the Y axis position with defining the breaks

```
ggplot(gapminder, aes(pop, lifeExp)) +
  geom_point(colour = trend_color) +
  scale_x_log10() +
  scale_y_continuous(breaks = c(0, 20, 25, 35, 40, 45, 50, 55, 60, 80))
```



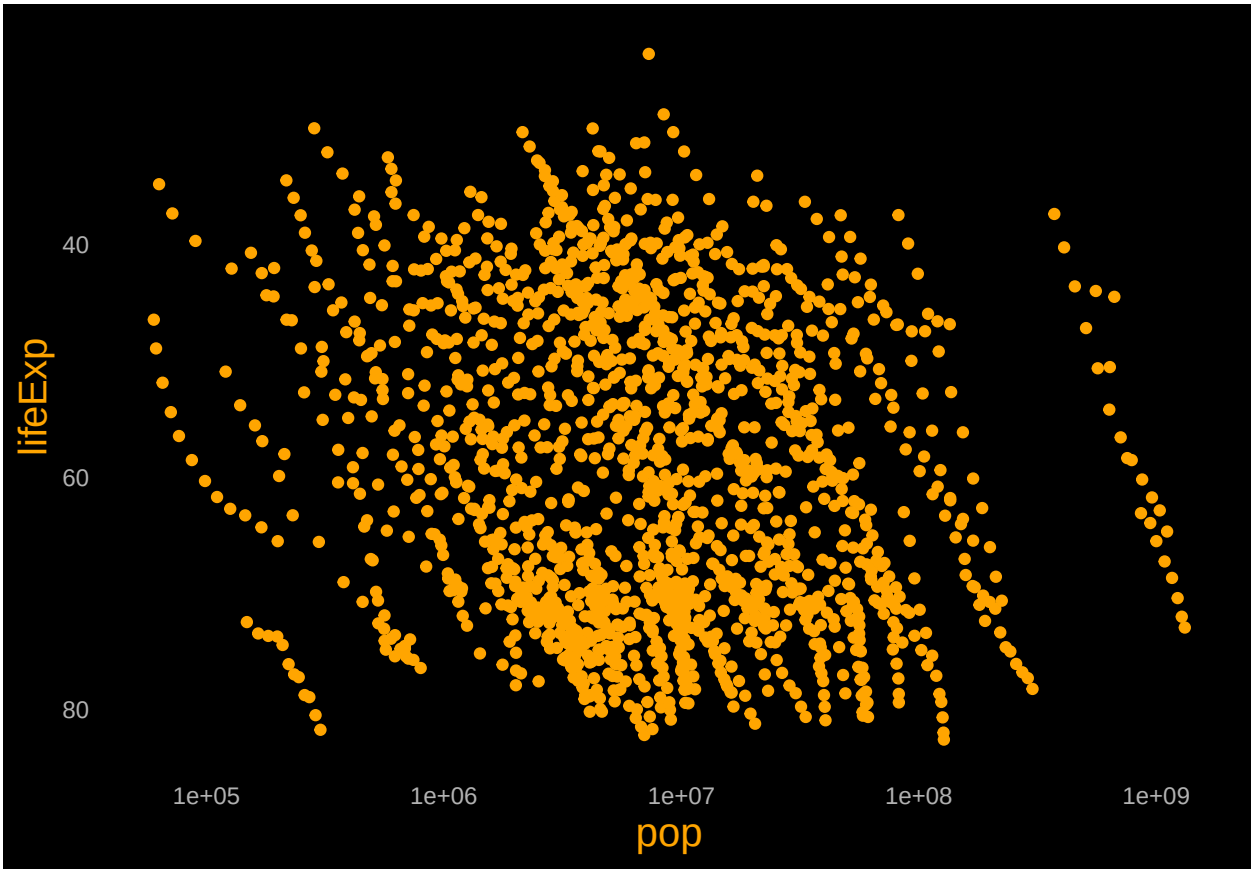
- #Add labels

```
ggplot(gapminder, aes(pop, lifeExp)) +
  geom_point(colour = trend_color) +
  scale_x_log10() +
  scale_y_continuous(breaks = c(30, 40, 60), label = c("low", "center", "high"))
```



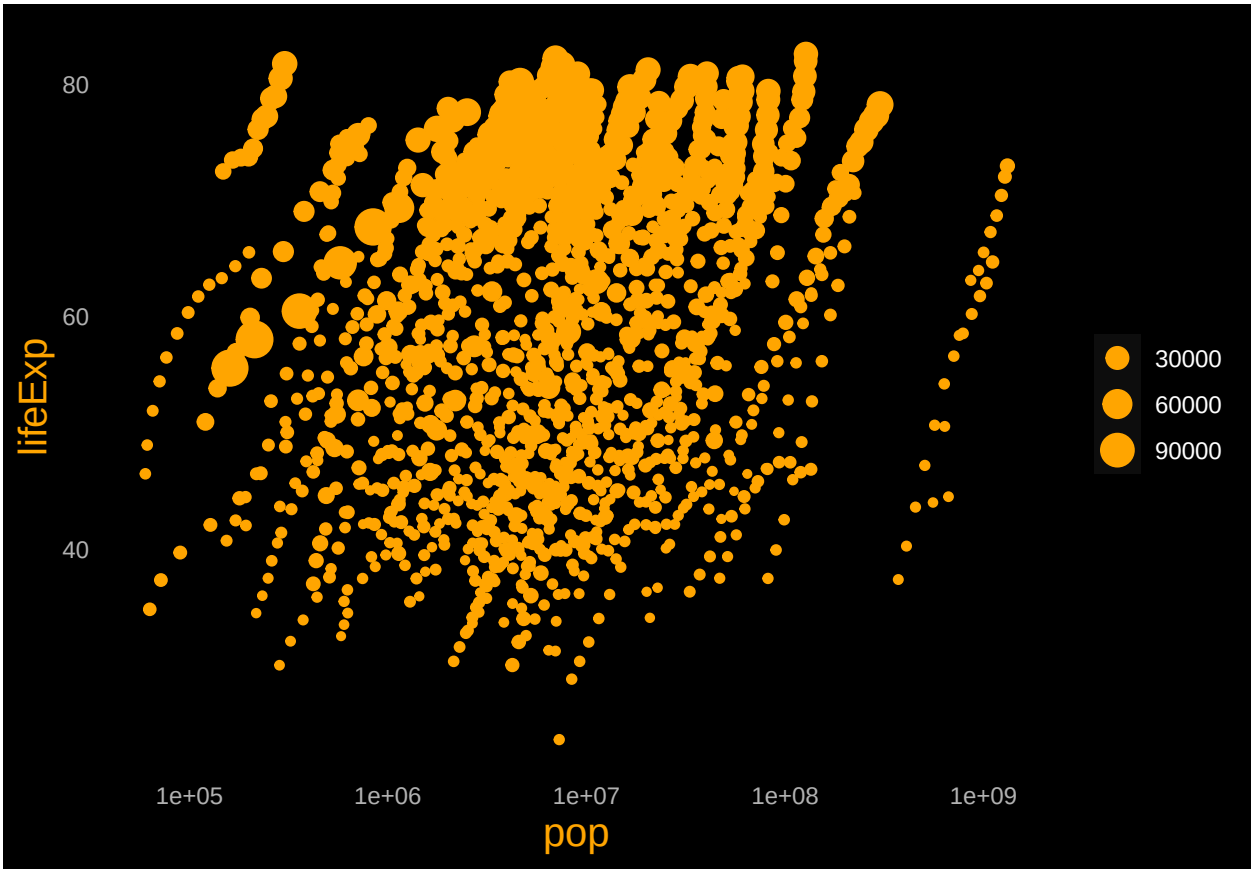
- #Change the Y scale in reverse

```
ggplot(gapminder, aes(pop, lifeExp)) +
  geom_point(colour = trend_color) +
  scale_x_log10() +
  scale_y_reverse()
```



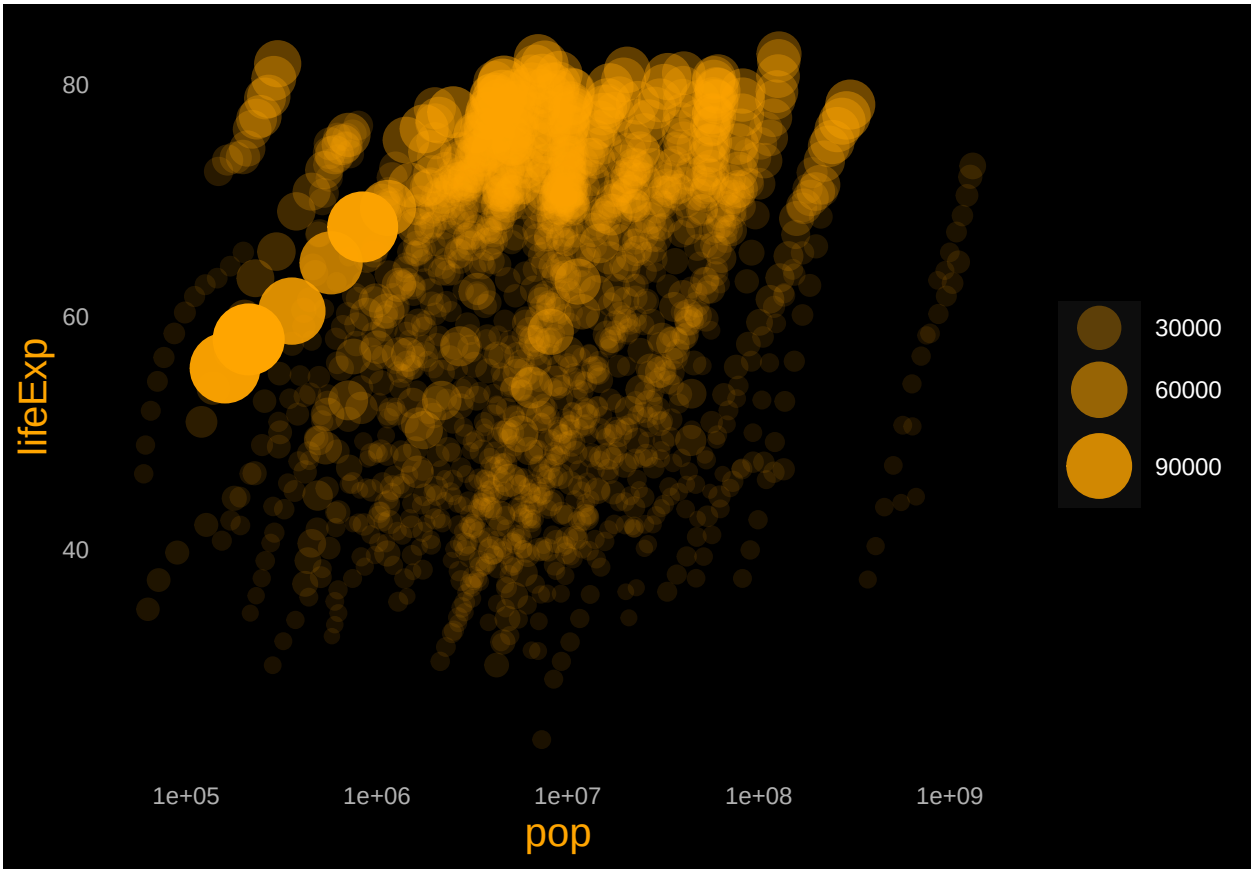
- #Add another visual encoding size

```
ggplot(gapminder, aes(pop, lifeExp, size = gdpPercap)) +
  geom_point(colour = trend_color) +
  scale_x_log10()  +
  scale_size()
```



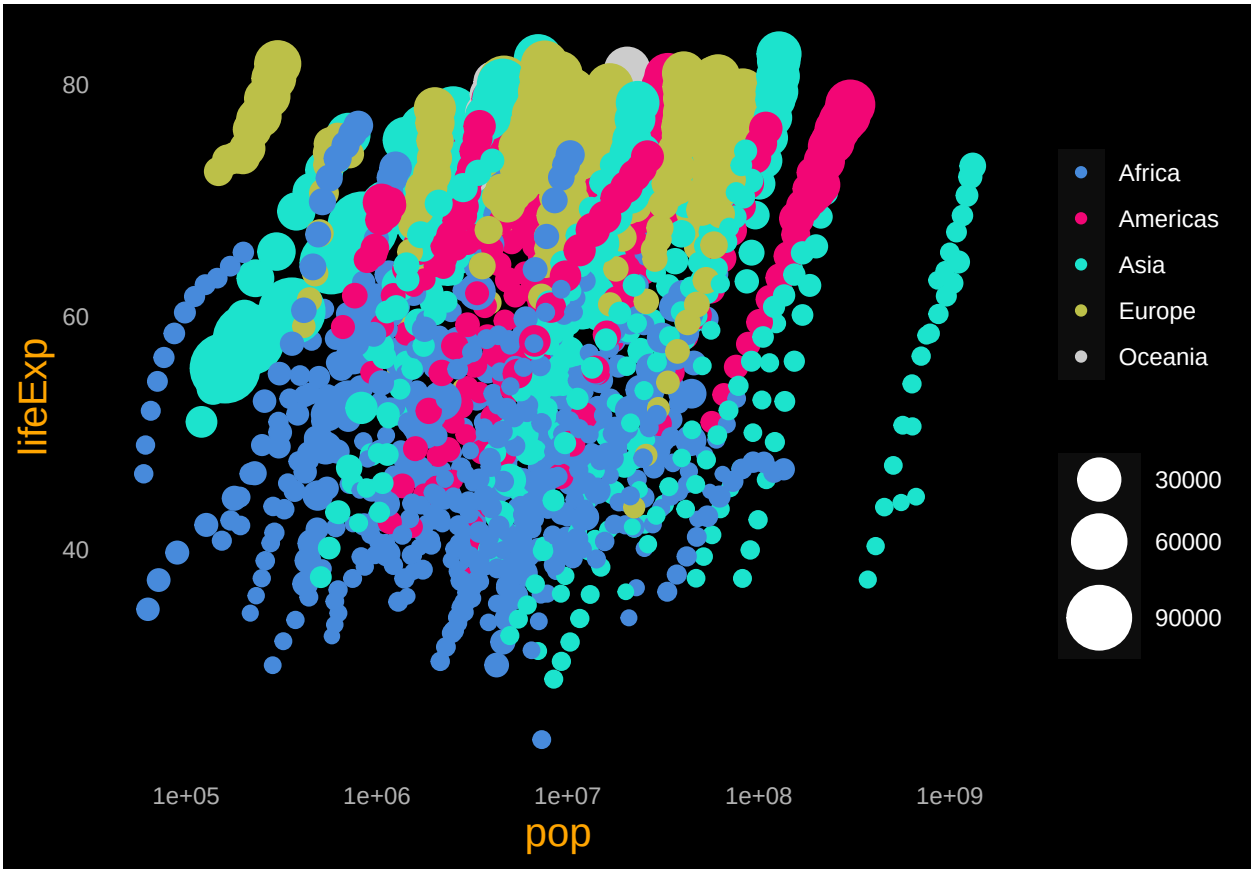
- #Apply a scale rage to the variable size

```
ggplot(gapminder, aes(pop, lifeExp, size = gdpPercap, alpha=gdpPercap)) +
  geom_point(colour = trend_color) +
  scale_x_log10()  +
  scale_size(range = c(2, 12))
```



- #Add another visual encoding color

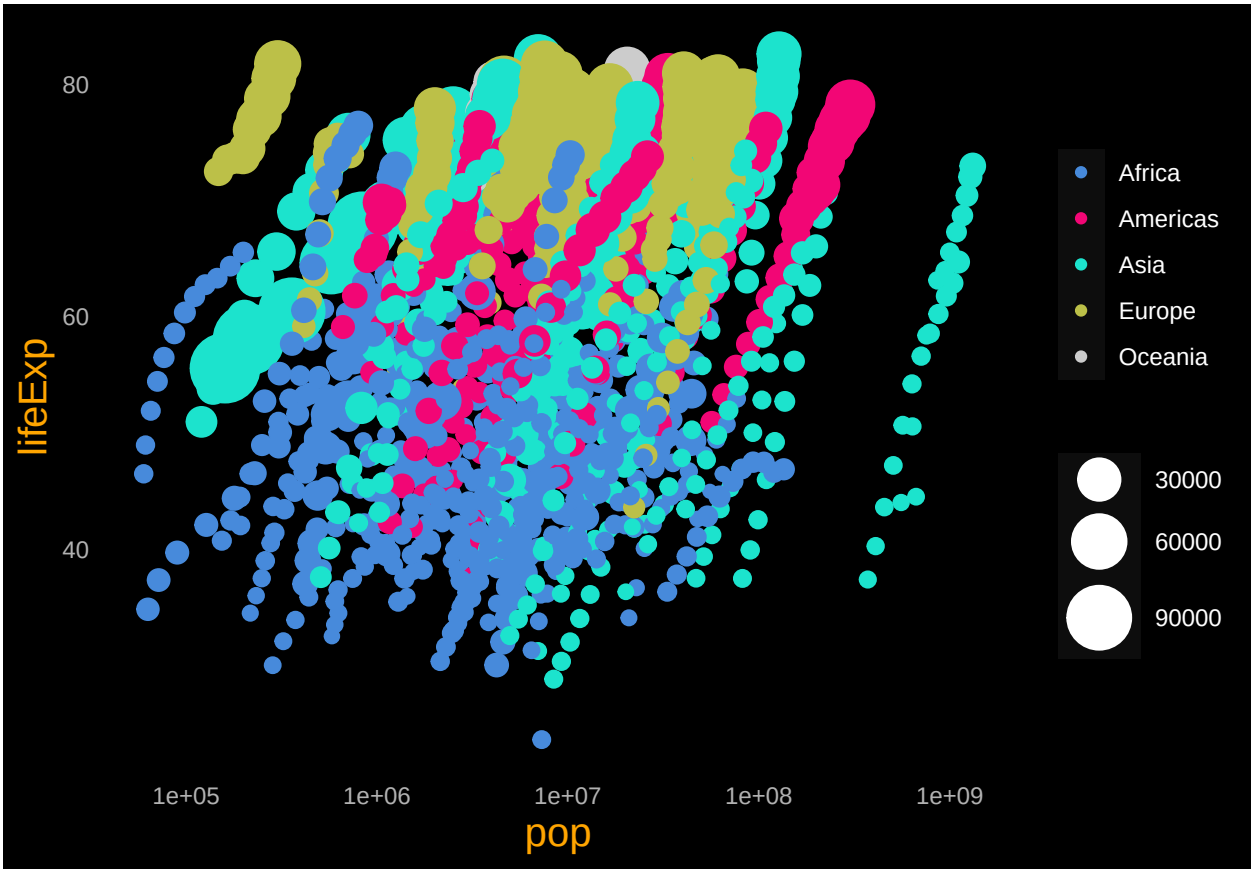
```
ggplot(gapminder, aes(pop, lifeExp, size = gdpPercap,  colour = continent)) +
  geom_point() +
  scale_x_log10()  +
  scale_size(range = c(2, 12)) +
  scale_colour_manual(values=c("#478adb", "#f20675", "#1ce3cd", "#bcc048", "#cccccc"))
```



- #Apply another scale to color

```
ggplot(gapminder, aes(pop, lifeExp, size = gdpPercap,  color = continent)) +
  geom_point() +
  scale_x_log10()  +
  scale_size(range = c(2, 12)) +
  scale_colour_manual(values = continent_colors) +
  scale_colour_manual(values=c("#478adb", "#f20675", "#1ce3cd", "#bcc048", "#cccccc"))
```

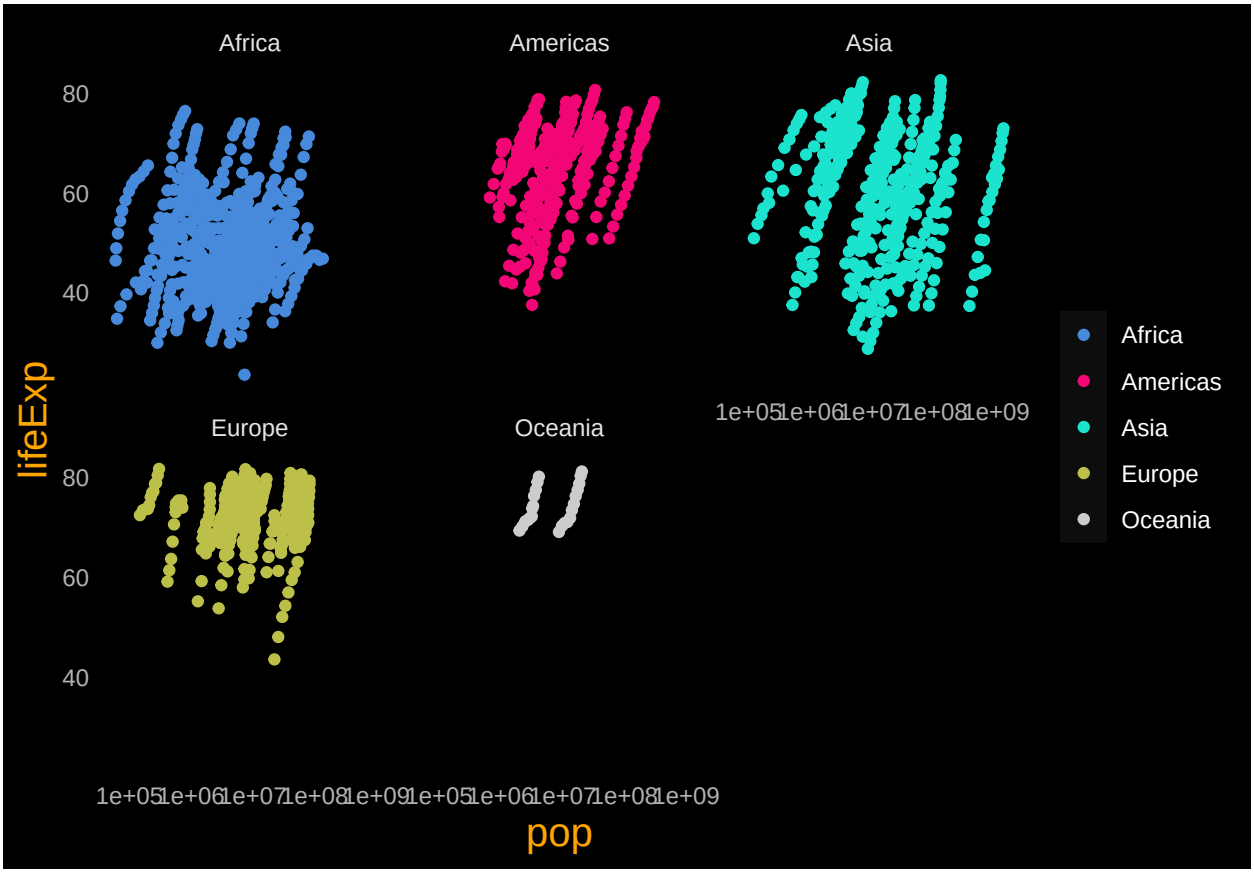
```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```



- #Change to facet

```
ggplot(gapminder, aes(pop, lifeExp,  colour = continent)) +
  geom_point() +
  scale_x_log10()  +
  scale_size(range = c(2, 12)) +
  scale_colour_manual(values = continent_colors) +
  facet_wrap(~continent) +
  scale_colour_manual(values=c("#478adb", "#f20675", "#1ce3cd", "#bcc048", "#cccccc"))
```

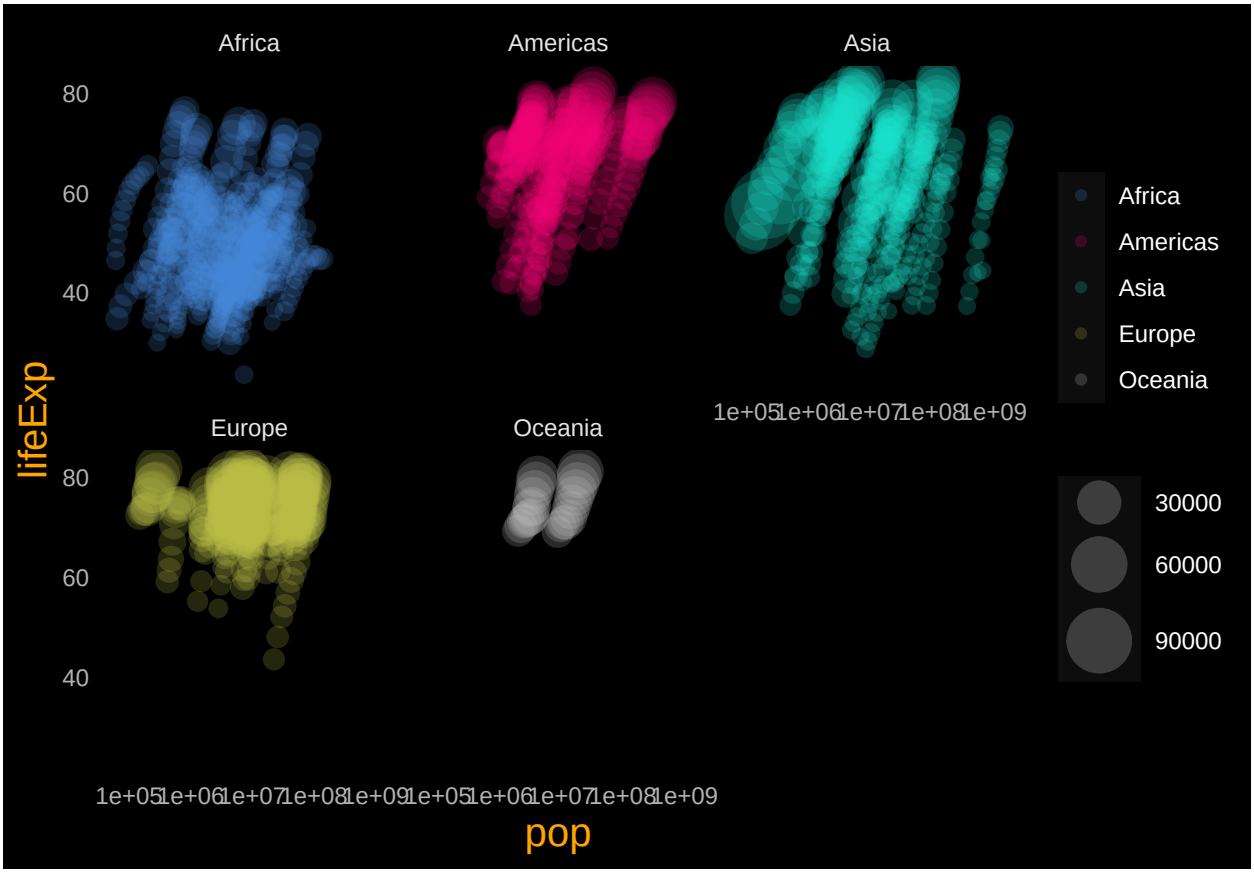
```
## Scale for 'colour' is already present. Adding another scale for 'colour',
## which will replace the existing scale.
```



- #Adding transparency

```
ggplot(gapminder, aes(pop, lifeExp, size = gdpPercap,  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_colour_manual(values=c("#478adb", "#f20675", "#1ce3cd", "#bcc048", "#cccccc"))
```

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



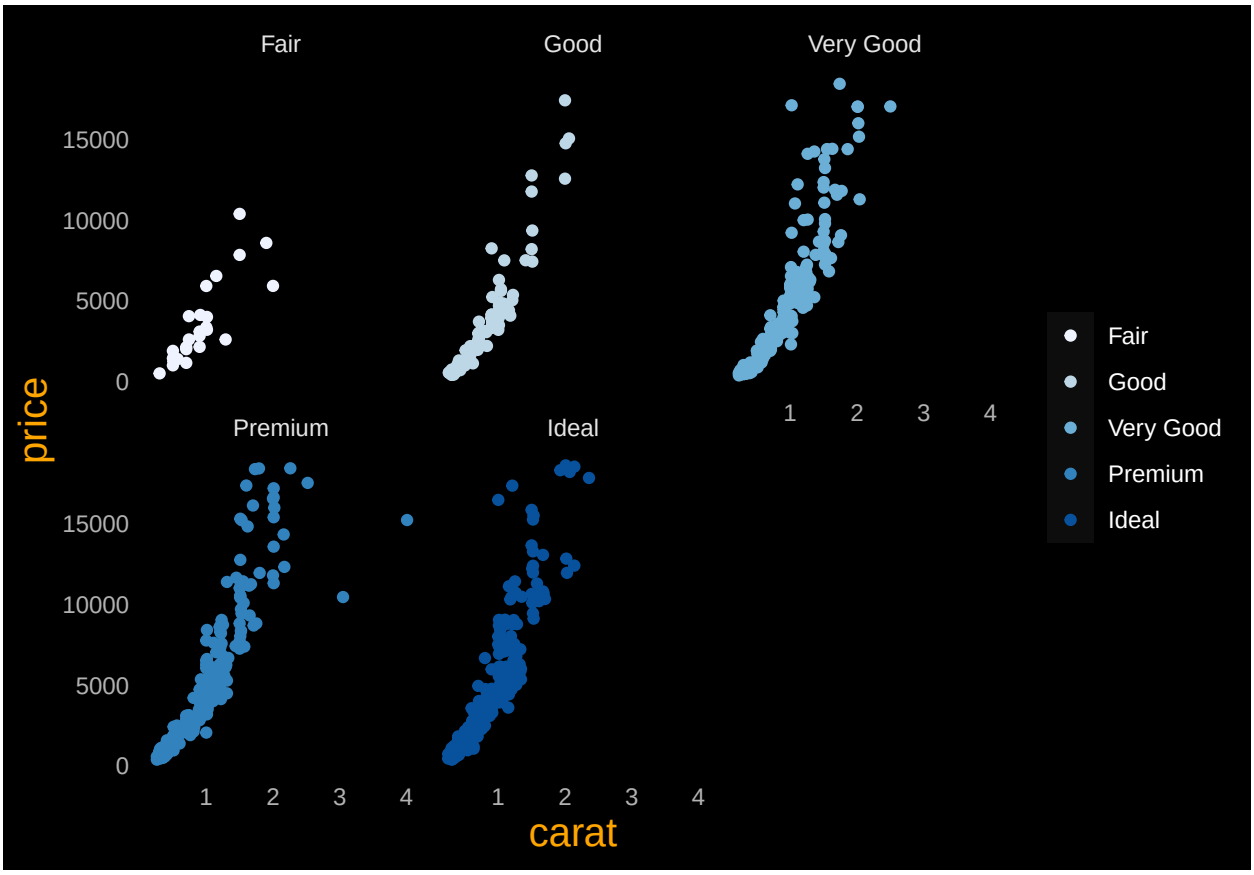
- #Creating a subsample and define the color scale

```
diamonds_sample <- diamonds[sample(nrow(diamonds), 1000),]

d <- ggplot(diamonds_sample, aes(carat, price)) +
  geom_point(aes(colour = cut))+
  facet_wrap(~cut)
```

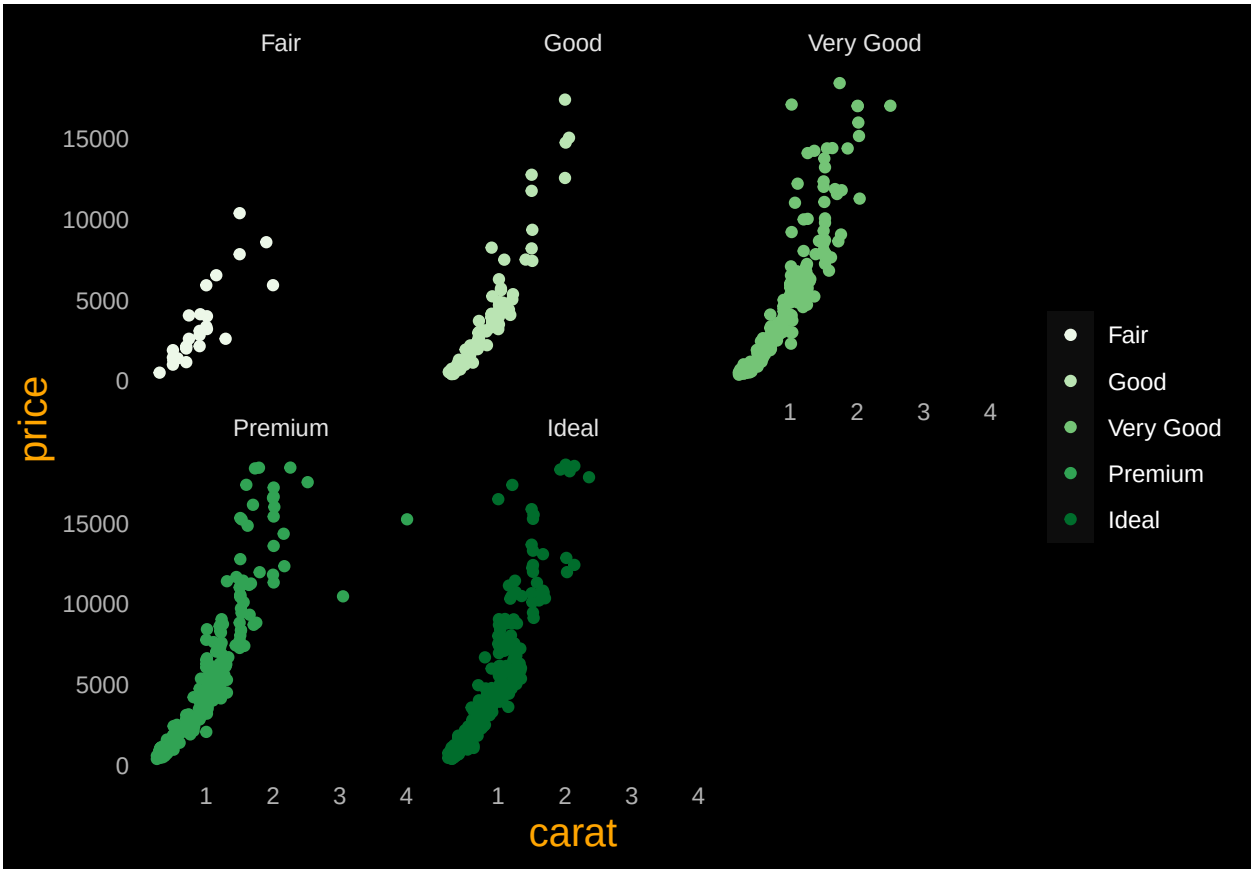
-

Change scale label

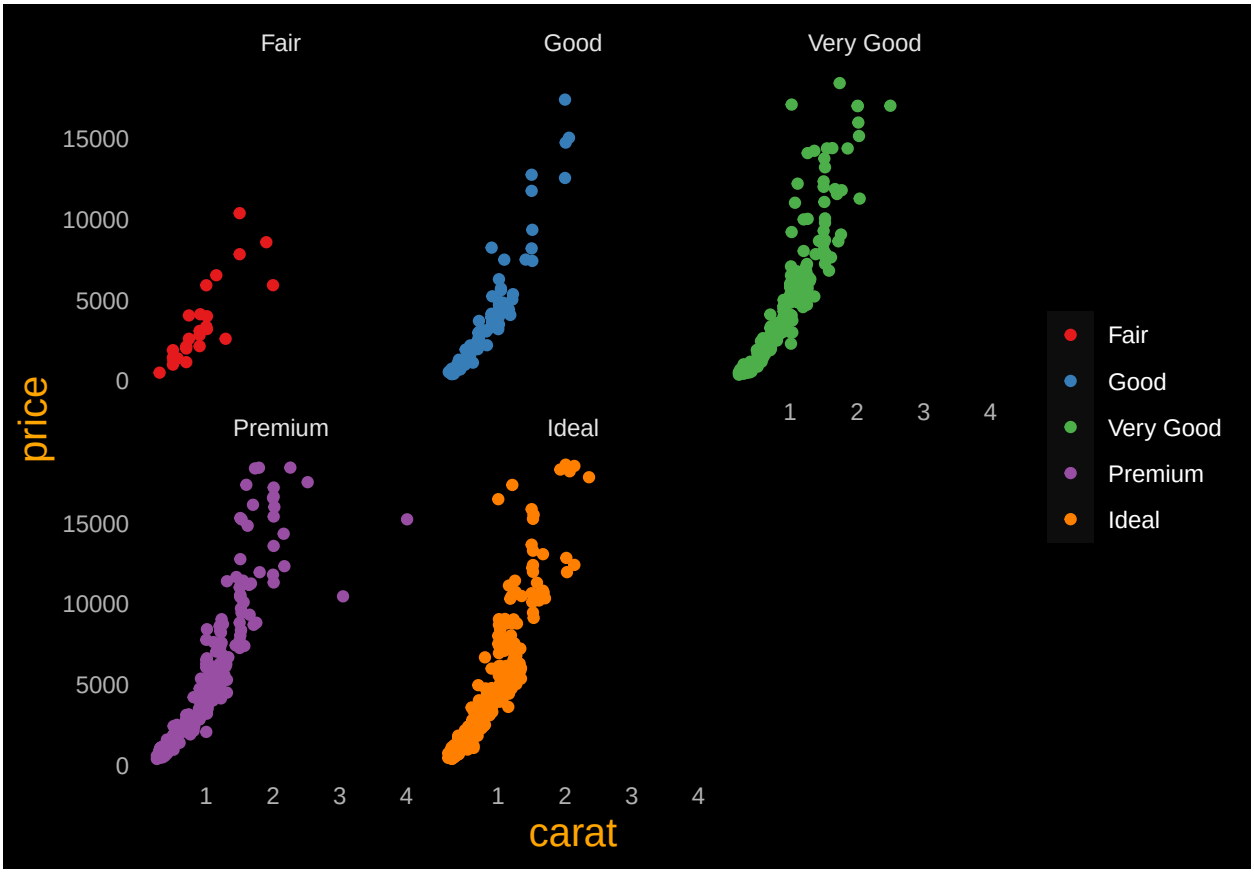
```
d + scale_colour_brewer("Diamond\nclarity")
```




```
d + scale_colour_brewer(palette = "Greens")
```

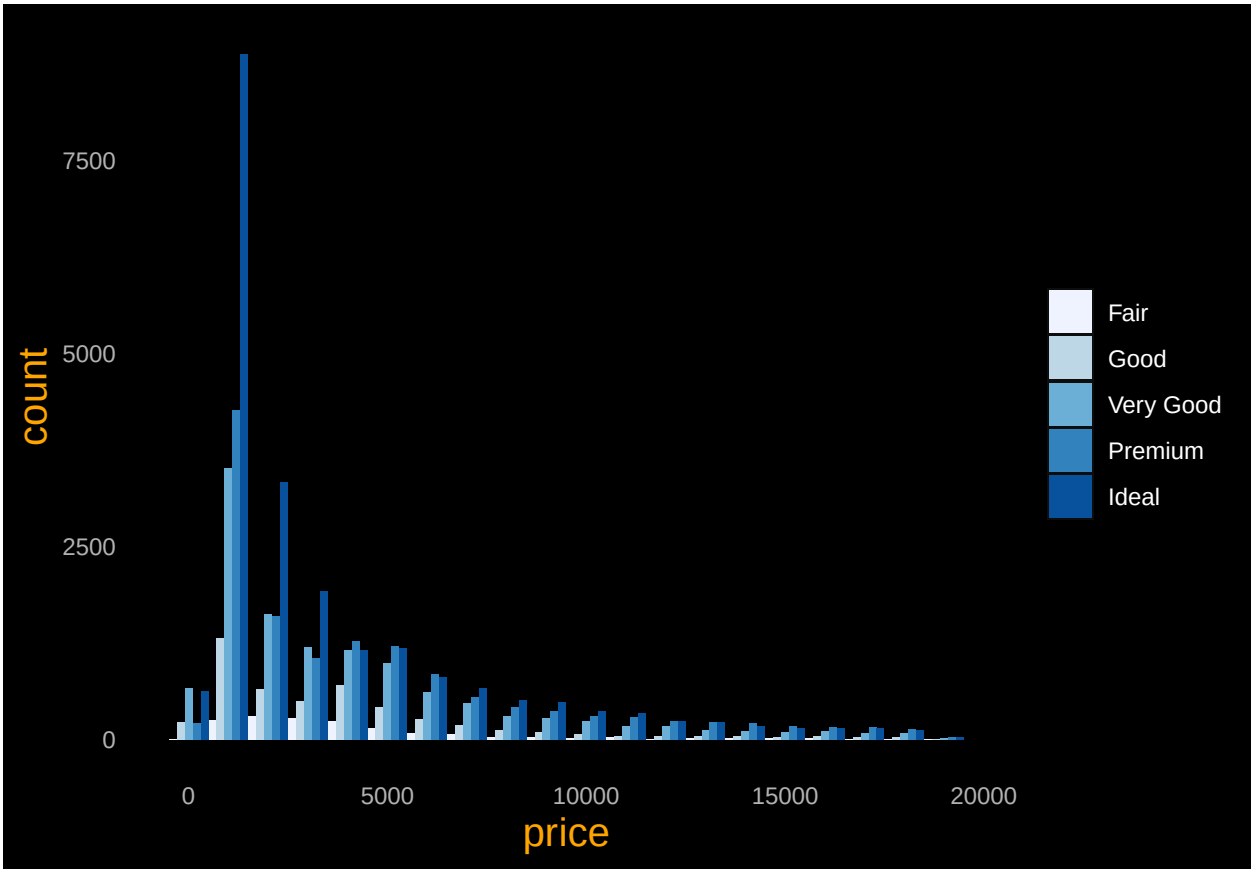



```
d + scale_colour_brewer(palette = "Set1")
```



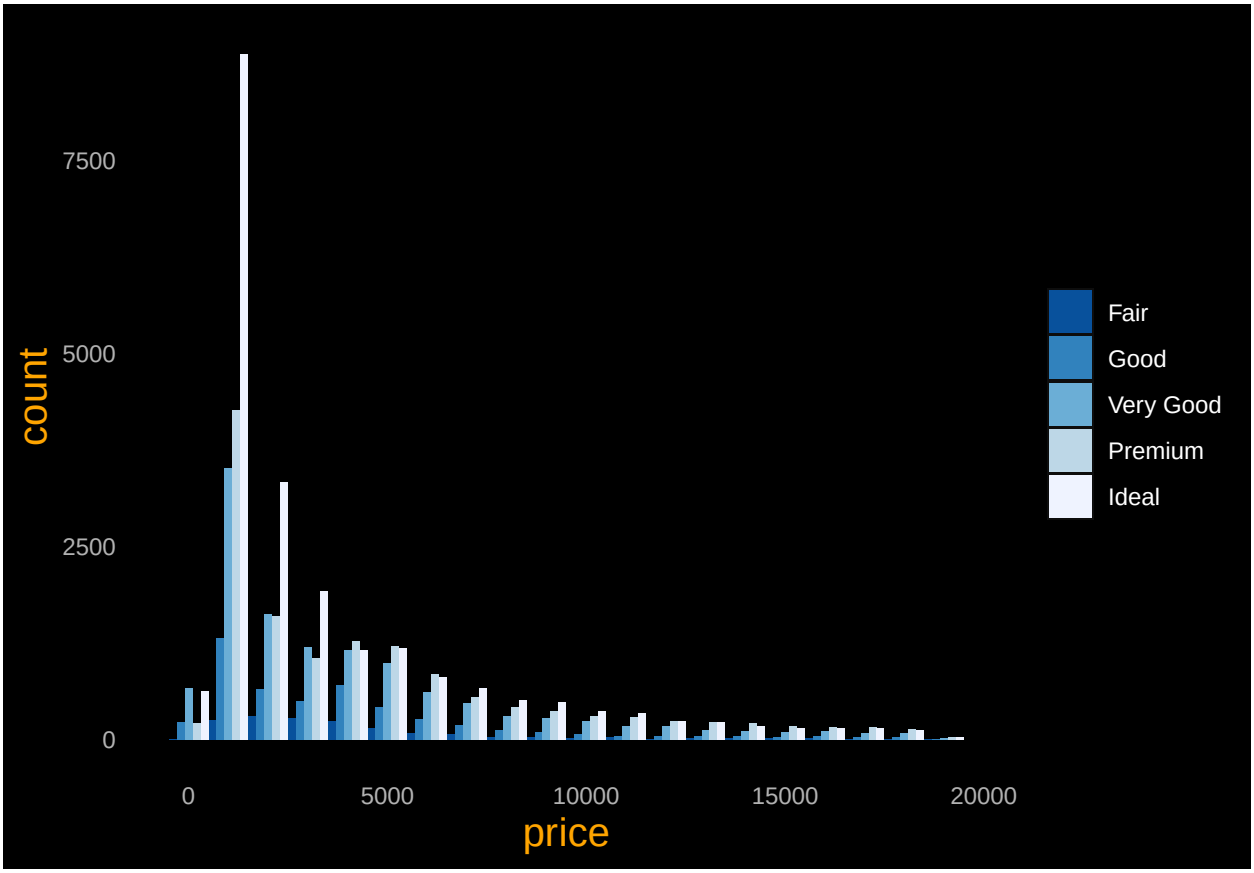
- scale_fill_brewer works just the same as
- scale_color_brewer but for fill colors

```
p <- ggplot(diamonds, aes(x = price, fill = cut)) +
  geom_histogram(position = "dodge", binwidth = 1000)

p + scale_fill_brewer()
```



- **the order of color can be reversed**

```
p + scale_fill_brewer(direction = -1)
```



- #Creating some random numbers

```
df <- data.frame(
  x = runif(100),
  y = runif(100),
  z1 = rnorm(100),
  z2 = abs(rnorm(100))
)
```

-#Check on the data

```
names(df)
```

```
## [1] "x"  "y"  "z1" "z2"
```

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

```
##             x          y         z1        z2
## 1  0.61499694 0.96905316  0.4479580 0.1259890
## 2  0.05497459 0.65730426 -1.9662132 0.3358868
## 3  0.05644488 0.47167873  0.8647621 0.4316280
## 4  0.71442110 0.85600347 -0.8258292 1.1799936
## 5  0.80239951 0.36596066 -0.3289556 0.4396213
## 6  0.90406166 0.28372453 -1.1728194 0.4064910
```

```
## 7  0.87067341 0.72693292 -0.3366023 0.8735598
## 8  0.85075538 0.81712643 -0.6903117 3.2364415
## 9  0.27524748 0.01559947  1.5637033 0.7876481
## 10 0.30763668 0.59154968 -1.7296209 0.2220633
```

```
str(df)
```

```
## 'data.frame':    100 obs. of  4 variables:
##  $ x : num  0.615 0.055 0.0564 0.7144 0.8024 ...
##  $ y : num  0.969 0.657 0.472 0.856 0.366 ...
##  $ z1: num  0.448 -1.966 0.865 -0.826 -0.329 ...
##  $ z2: num  0.126 0.336 0.432 1.18 0.44 ...
```

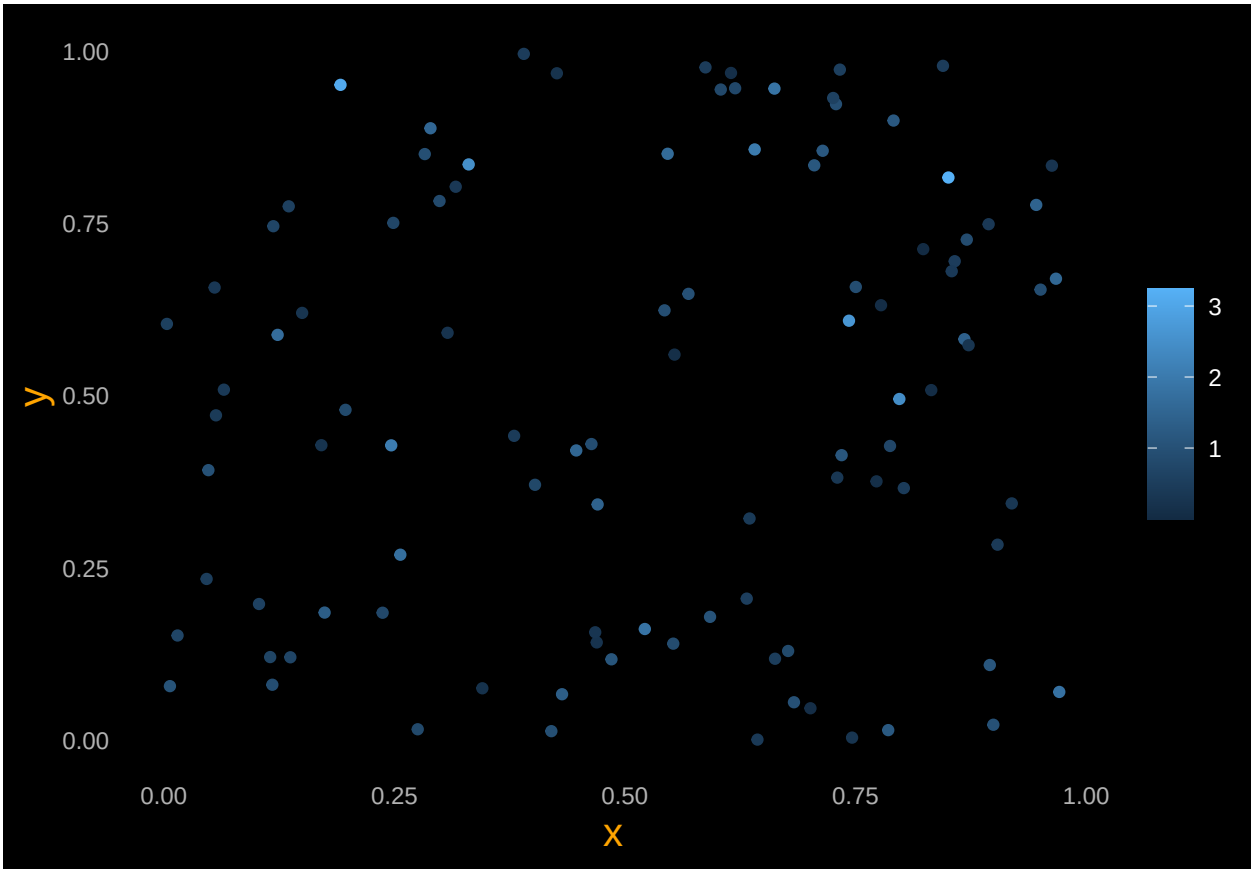
```
summary(df)
```

```
##        x                 y                 z1                z2
##  Min.   :0.003252   Min.   :0.0005733   Min.   :-1.96621   Min.   :0.002142
##  1st Qu.:0.280989   1st Qu.:0.1833791   1st Qu.:-0.74650   1st Qu.:0.396049
##  Median :0.578085   Median :0.5018492   Median :-0.09993   Median :0.764822
##  Mean   :0.527723   Mean   :0.4960624   Mean   :-0.04070   Mean   :0.878158
##  3rd Qu.:0.773997   3rd Qu.:0.7758002   3rd Qu.: 0.62036   3rd Qu.:1.142540
##  Max.   :0.970982   Max.   :0.9967273   Max.   : 2.05299   Max.   :3.236441
```

-

Default colour scale colours from light blue to dark blue

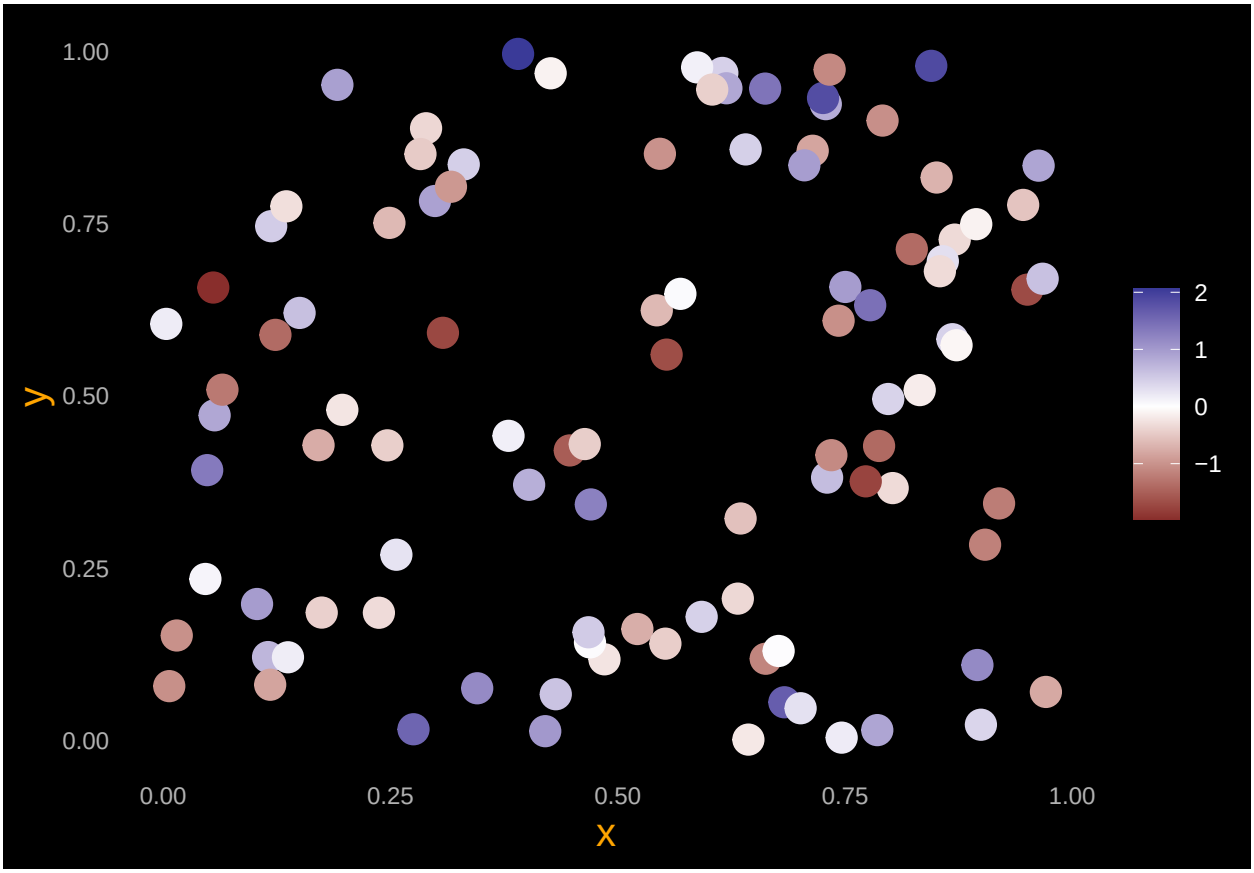
```
ggplot(df, aes(x, y)) +
  geom_point(aes(colour = z2))
```



-

For diverging colour scales use gradient2

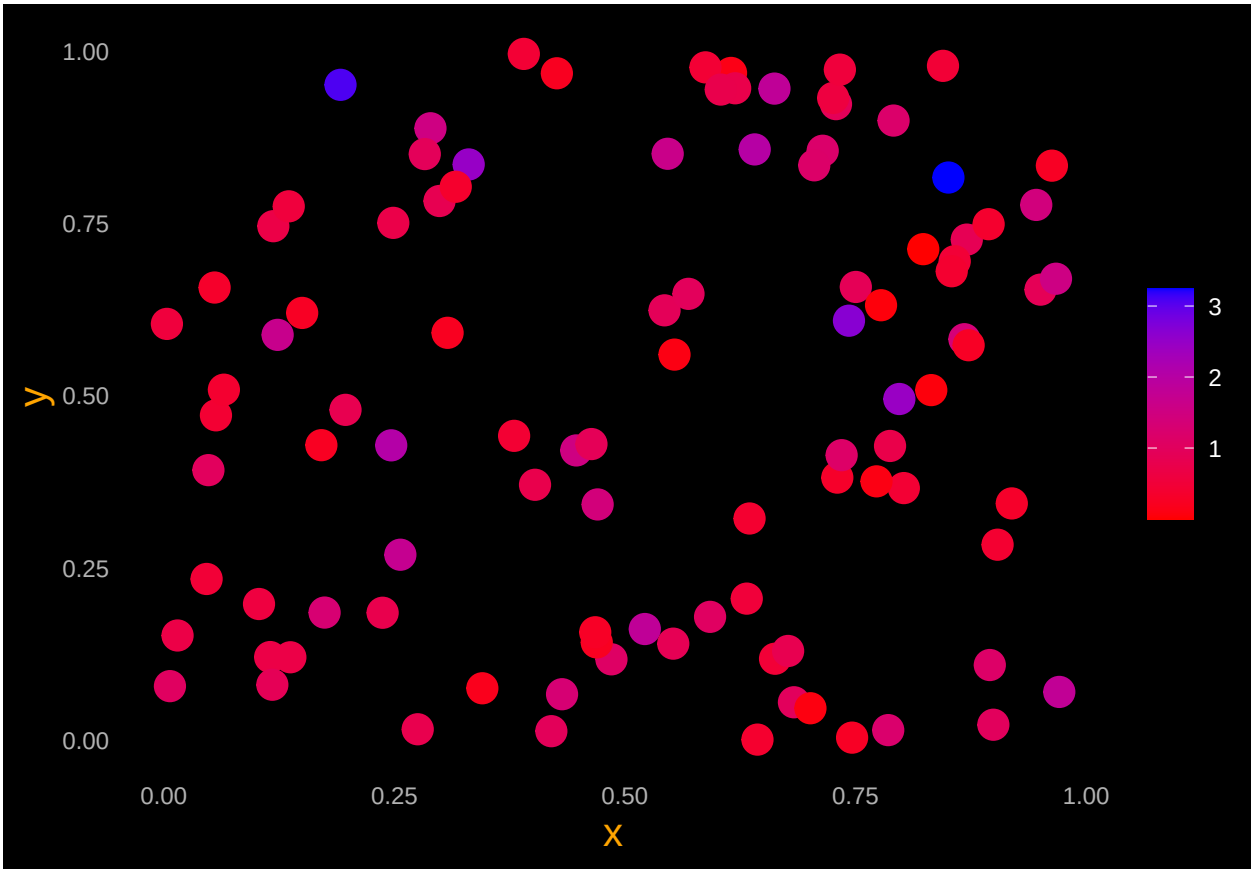
```
ggplot(df, aes(x, y)) +
  geom_point(aes(colour = z1), size=5) +
  scale_colour_gradient2()
```



-

Adjust colour choices with low and high

```
ggplot(df, aes(x, y)) +
  geom_point(aes(colour = z2), size=5) +
  scale_colour_gradient(low = "red", high = "blue")
```



- #Check on the data

```
names(mtcars)
```

```
##  [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
## [11] "carb"
```

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

```
##                    mpg cyl  disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6 160.0 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6 160.0 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4 108.0  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6 258.0 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8 360.0 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6 225.0 105 2.76 3.460 20.22  1  0    3    1
## Duster 360        14.3   8 360.0 245 3.21 3.570 15.84  0  0    3    4
## Merc 240D         24.4   4 146.7  62 3.69 3.190 20.00  1  0    4    2
## Merc 230          22.8   4 140.8  95 3.92 3.150 22.90  1  0    4    2
## Merc 280          19.2   6 167.6 123 3.92 3.440 18.30  1  0    4    4
```

```
str(mtcars)
```

```
## 'data.frame':    32 obs. of  11 variables:
```

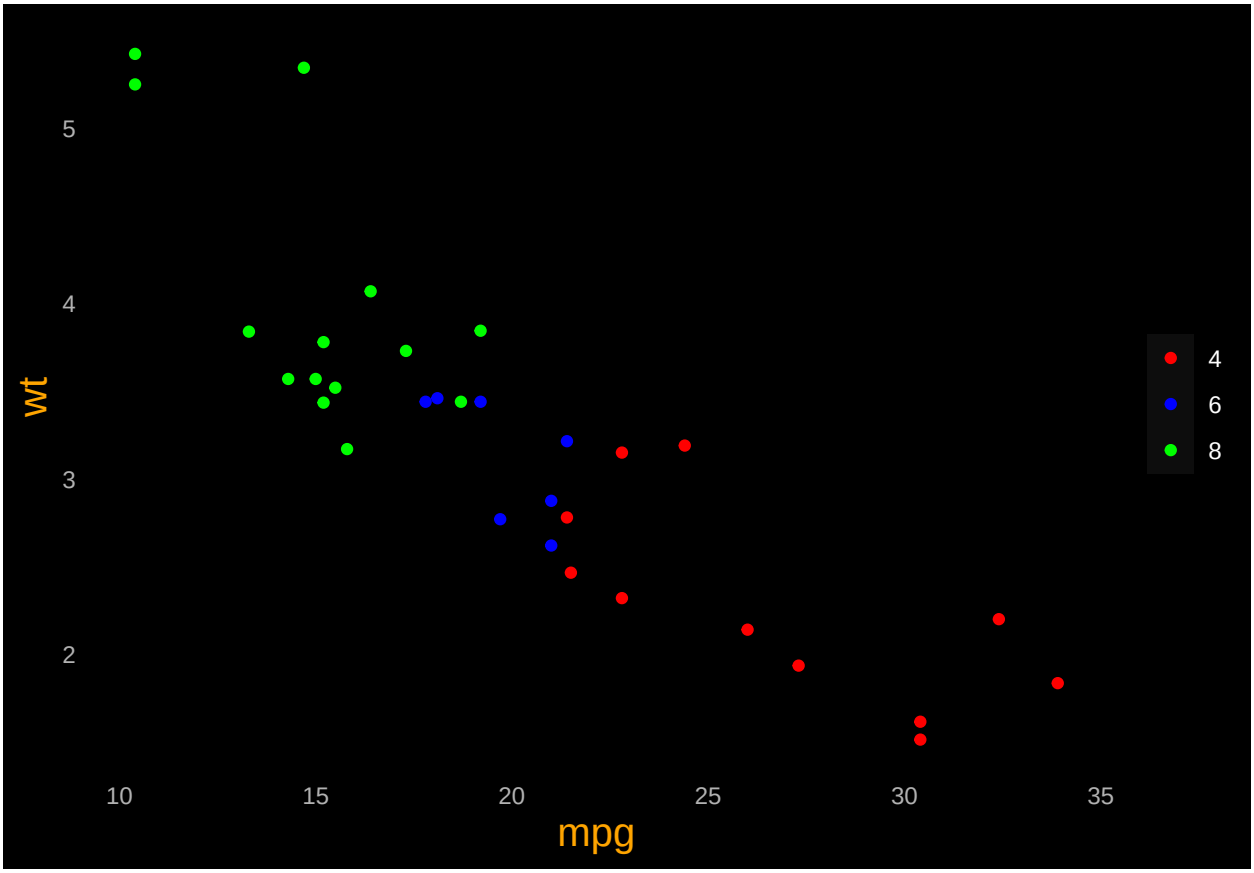
```
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

```
summary(mtcars)
```

```
##       mpg             cyl             disp             hp       
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0  
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5  
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0  
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7  
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0  
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0  
##       drat             wt             qsec             vs        
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000  
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000  
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000  
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375  
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000  
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000  
##        am              gear            carb      
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000  
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000  
##  Median :0.0000   Median :4.000   Median :2.000  
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812  
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000  
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

- #Creating color scale manual

```
p <- ggplot(mtcars, aes(mpg, wt)) +
  geom_point(aes(colour = factor(cyl)))

p + scale_colour_manual(values = c("red", "blue", "green"))
```



- #Check on the data

```
names(mpg)
```

```
##  [1] "manufacturer" "model"        "displ"        "year"         "cyl"
##  [6] "trans"        "drv"          "cty"          "hwy"          "fl"
## [11] "class"
```

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

```
## # A tibble: 10 x 11
##    manufacturer model    displ  year   cyl trans   drv     cty   hwy fl    class
##    <chr>        <chr>    <dbl> <int> <int> <chr>   <chr> <int> <int> <chr> <chr>
##  1 audi         a4         1.8  1999     4 auto(l~ f        18    29 p     comp~
##  2 audi         a4         1.8  1999     4 manual~ f        21    29 p     comp~
##  3 audi         a4         2    2008     4 manual~ f        20    31 p     comp~
##  4 audi         a4         2    2008     4 auto(a~ f        21    30 p     comp~
##  5 audi         a4         2.8  1999     6 auto(l~ f        16    26 p     comp~
##  6 audi         a4         2.8  1999     6 manual~ f        18    26 p     comp~
##  7 audi         a4         3.1  2008     6 auto(a~ f        18    27 p     comp~
##  8 audi         a4 quat~   1.8  1999     4 manual~ 4        18    26 p     comp~
##  9 audi         a4 quat~   1.8  1999     4 auto(l~ 4        16    25 p     comp~
## 10 audi         a4 quat~   2    2008     4 manual~ 4        20    28 p     comp~
```

```
str(mpg)
```

```
## tibble [234 x 11] (S3: tbl_df/tbl/data.frame)
##  $ manufacturer: chr [1:234] "audi" "audi" "audi" "audi" ...
##  $ model       : chr [1:234] "a4" "a4" "a4" "a4" ...
##  $ displ       : num [1:234] 1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int [1:234] 1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int [1:234] 4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr [1:234] "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr [1:234] "f" "f" "f" "f" ...
##  $ cty         : int [1:234] 18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int [1:234] 29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr [1:234] "p" "p" "p" "p" ...
##  $ class       : chr [1:234] "compact" "compact" "compact" "compact" ...
```

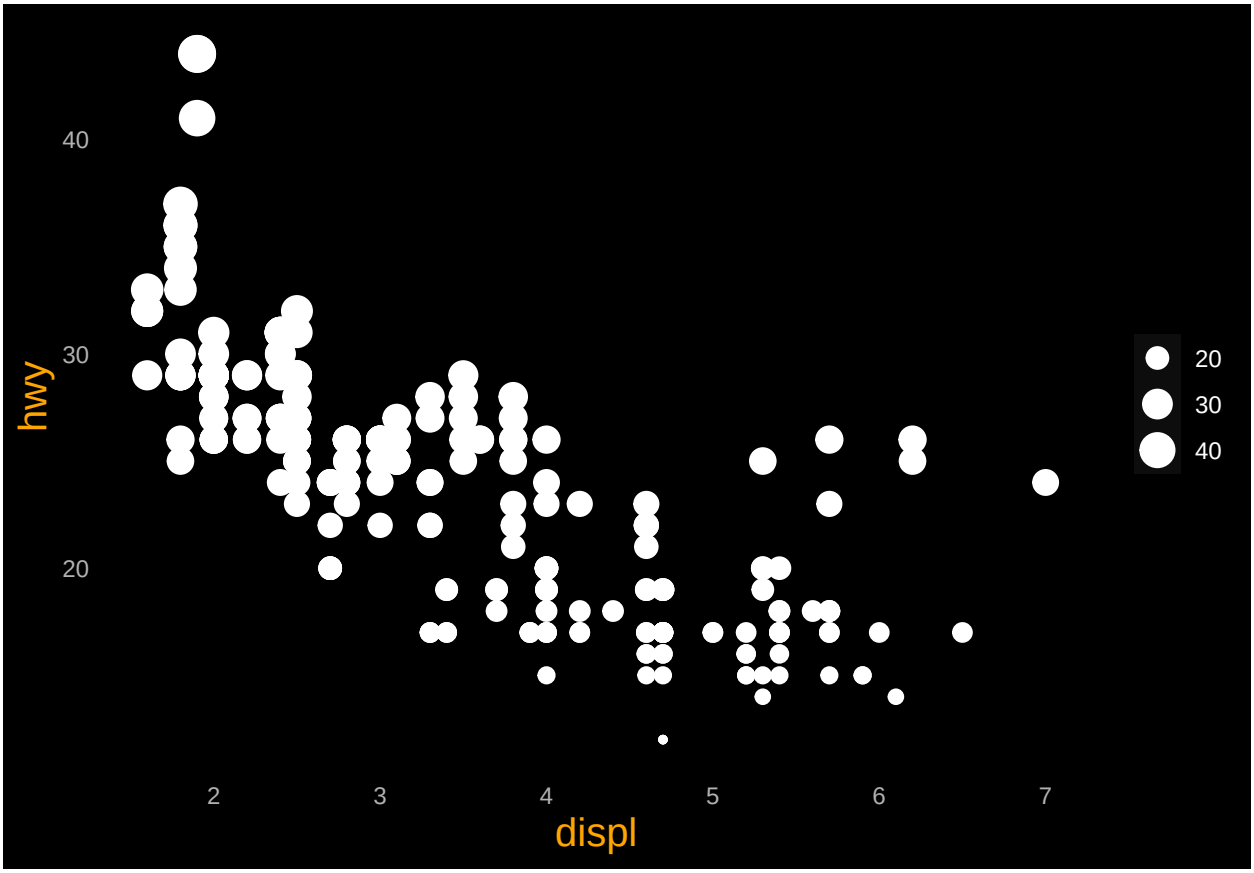
```
summary(mpg)
```

```
##  manufacturer          model               displ            year
##  Length:234         Length:234         Min.   :1.600   Min.   :1999
##  Class :character   Class :character   1st Qu.:2.400   1st Qu.:1999
##  Mode  :character   Mode  :character   Median :3.300   Median :2004
##                                        Mean   :3.472   Mean   :2004
##                                        3rd Qu.:4.600   3rd Qu.:2008
##                                        Max.   :7.000   Max.   :2008
##       cyl           trans               drv                 cty
##  Min.   :4.000   Length:234         Length:234         Min.   : 9.00
##  1st Qu.:4.000   Class :character   Class :character   1st Qu.:14.00
##  Median :6.000   Mode  :character   Mode  :character   Median :17.00
##  Mean   :5.889                                         Mean   :16.86
##  3rd Qu.:8.000                                         3rd Qu.:19.00
##  Max.   :8.000                                         Max.   :35.00
##       hwy             fl                class
##  Min.   :12.00   Length:234         Length:234
##  1st Qu.:18.00   Class :character   Class :character
##  Median :24.00   Mode  :character   Mode  :character
##  Mean   :23.44
##  3rd Qu.:27.00
##  Max.   :44.00
```

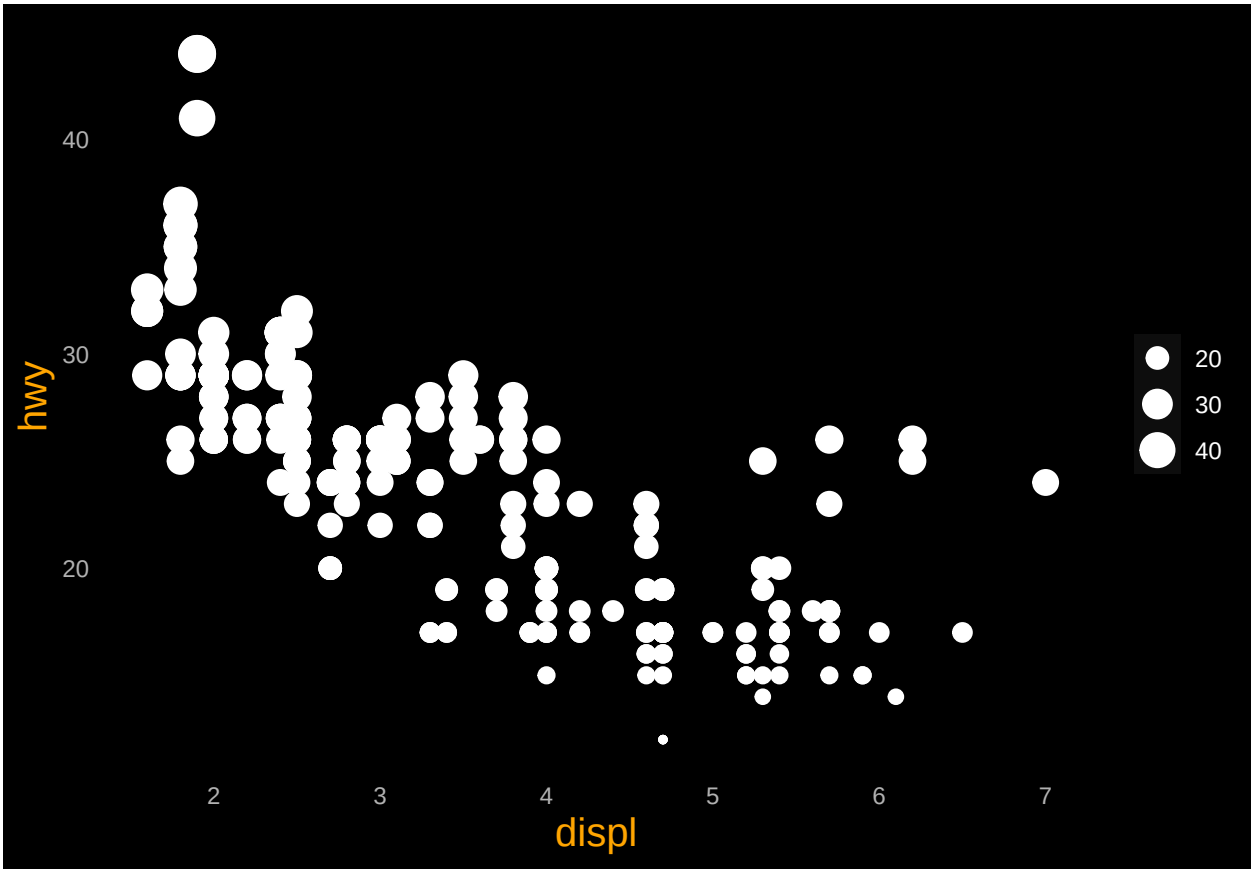
-

Scale size

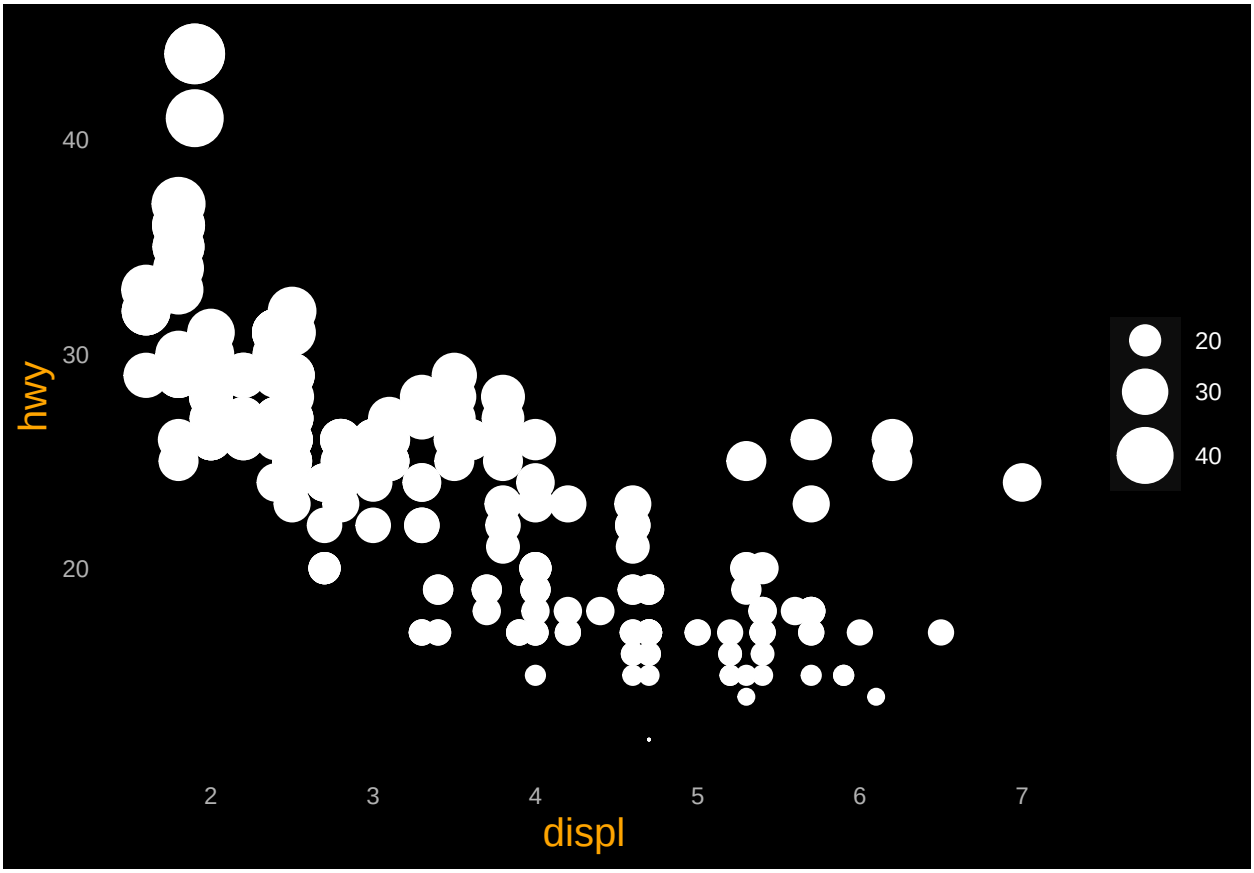
```
p <- ggplot(mpg, aes(displ, hwy, size = hwy)) +
  geom_point()
plot(p)
```




```
p + scale_size("Highway mpg")
```



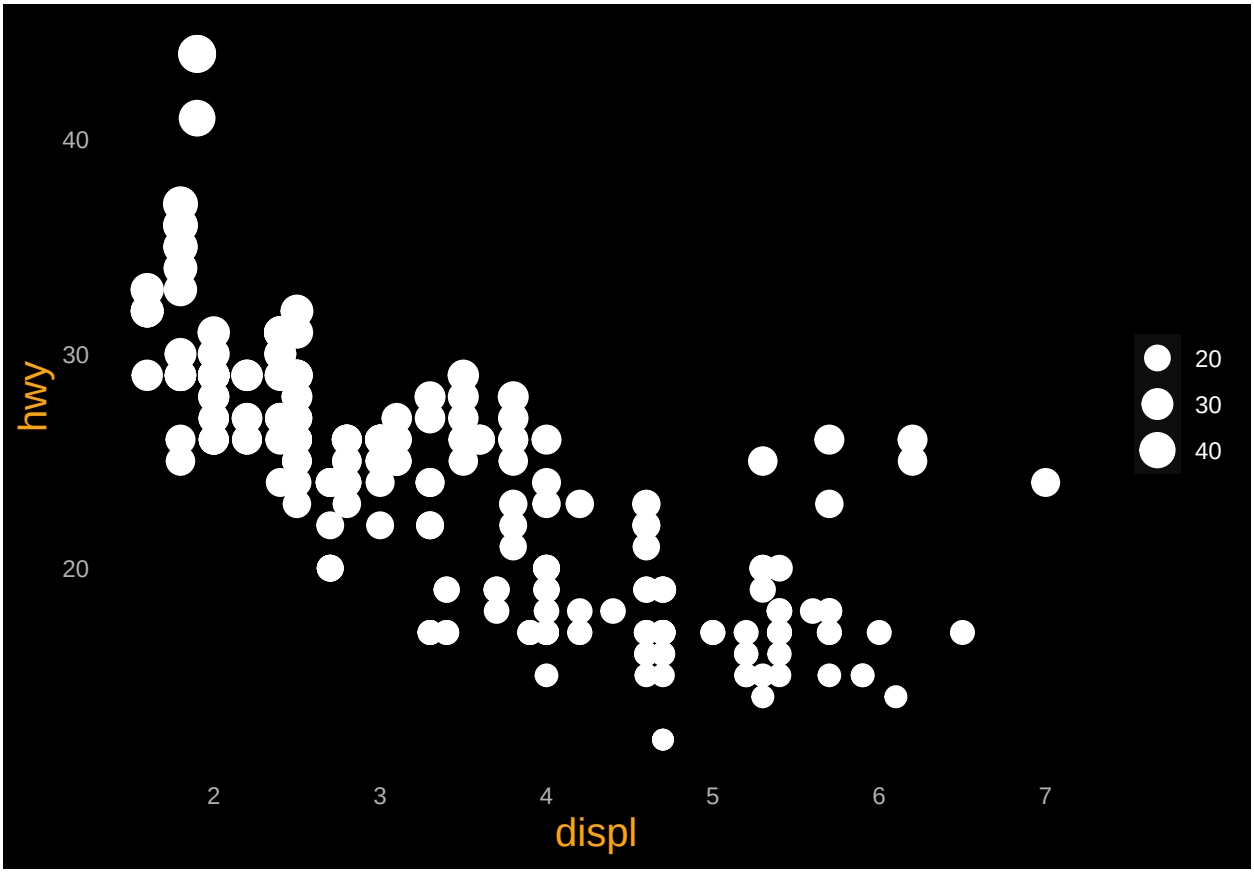

```
p + scale_size(range = c(0, 10))
```



-

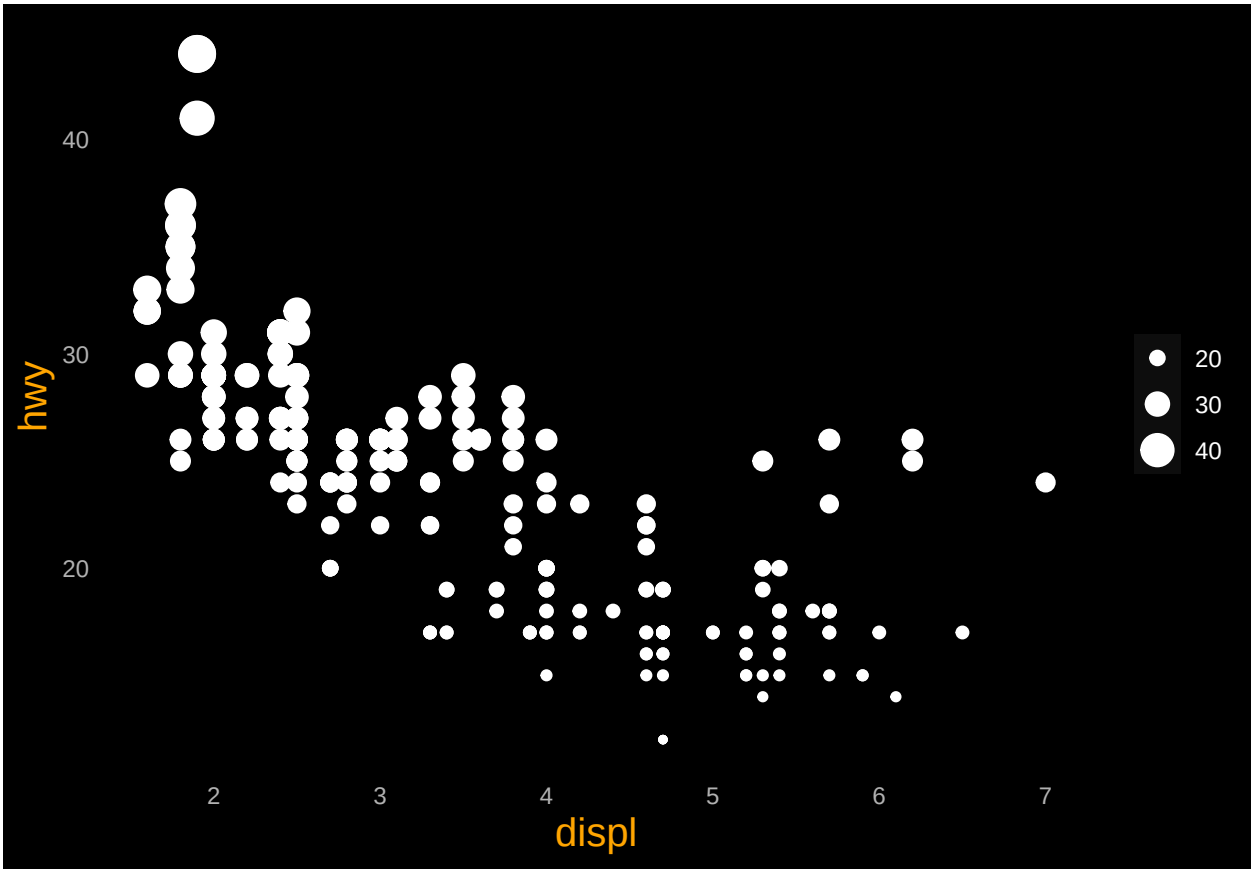
If you want zero value to have zero size, use scale_size_area:

```
p + scale_size_area()
```



- **If you want to map size to radius (usually bad idea), use scale_radius**

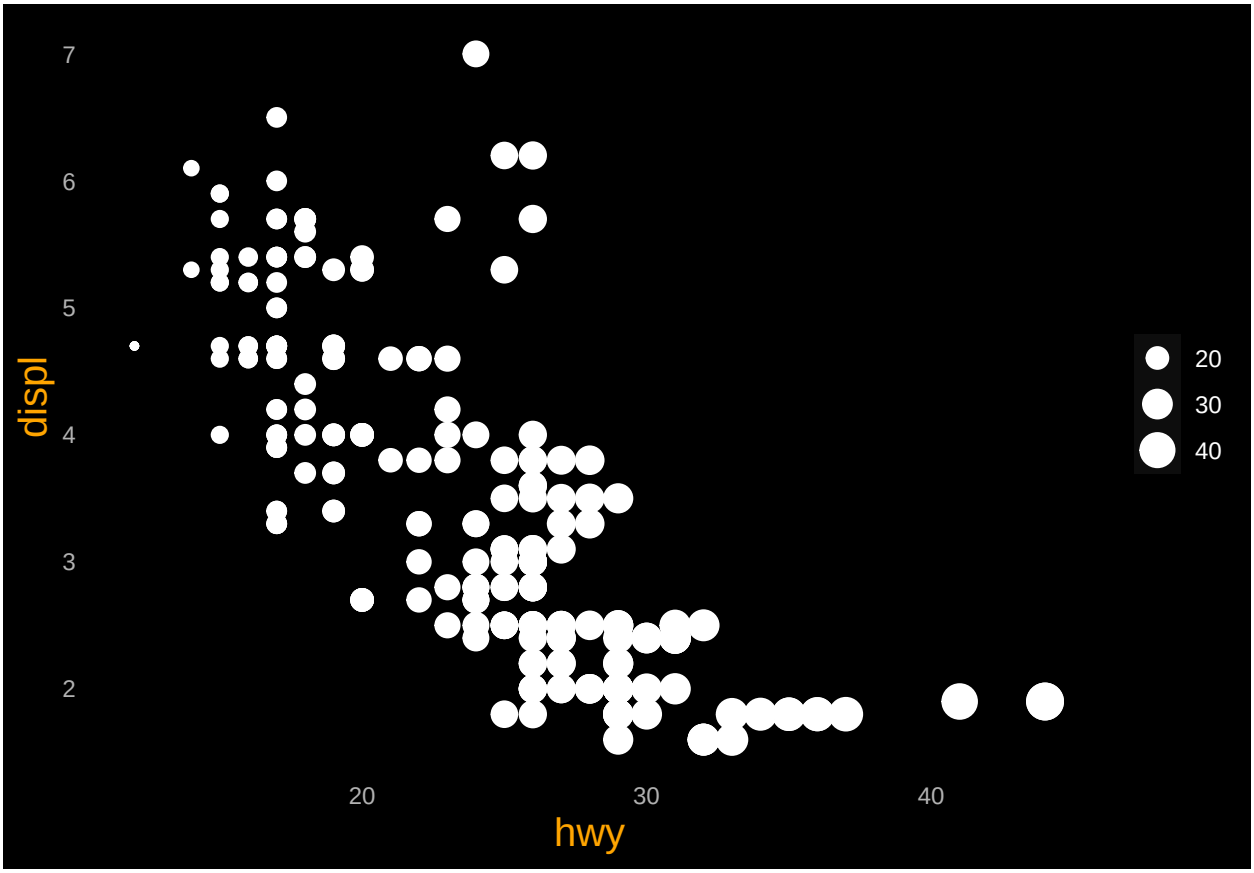
```
p + scale_radius()
```



- #Axis
-

Flip the X and Y axis

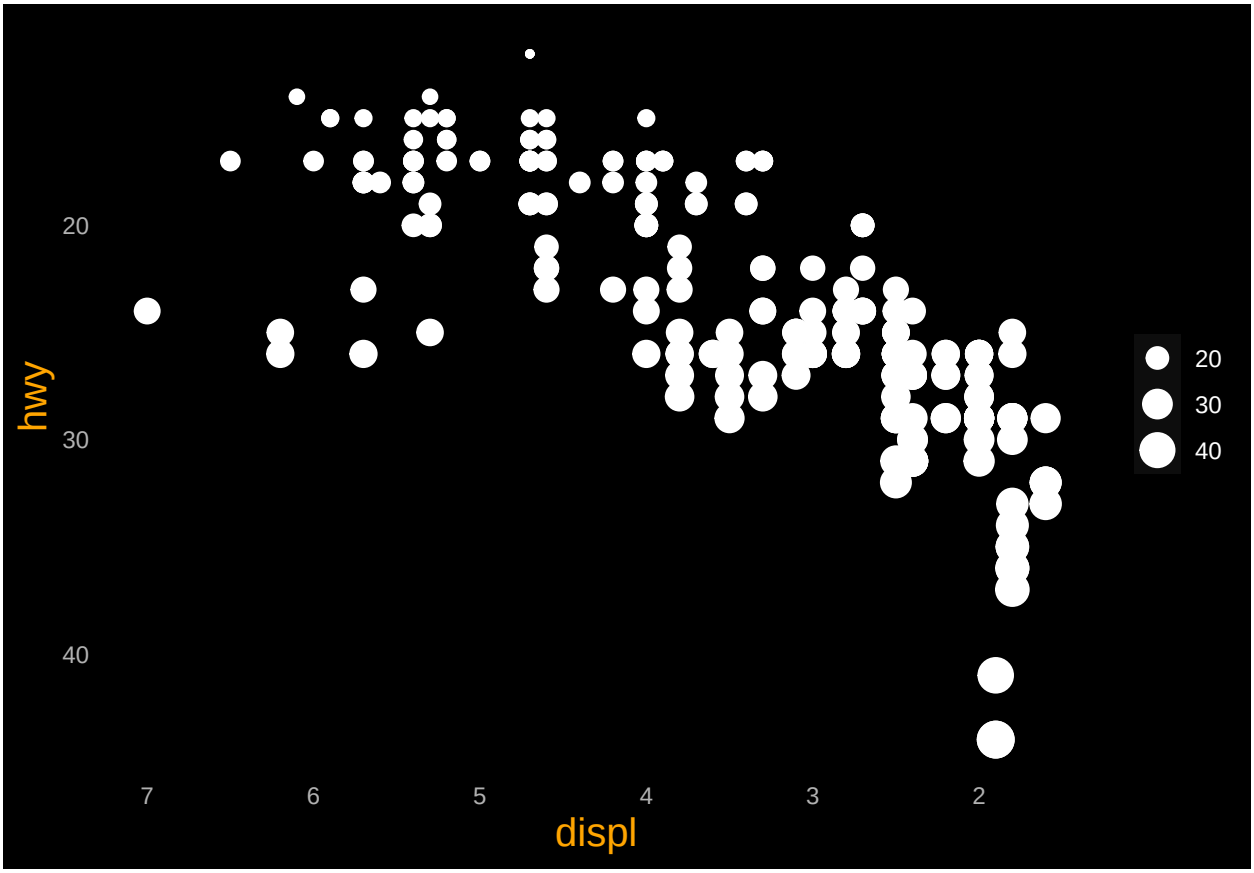
```
p + coord_flip()
```



-

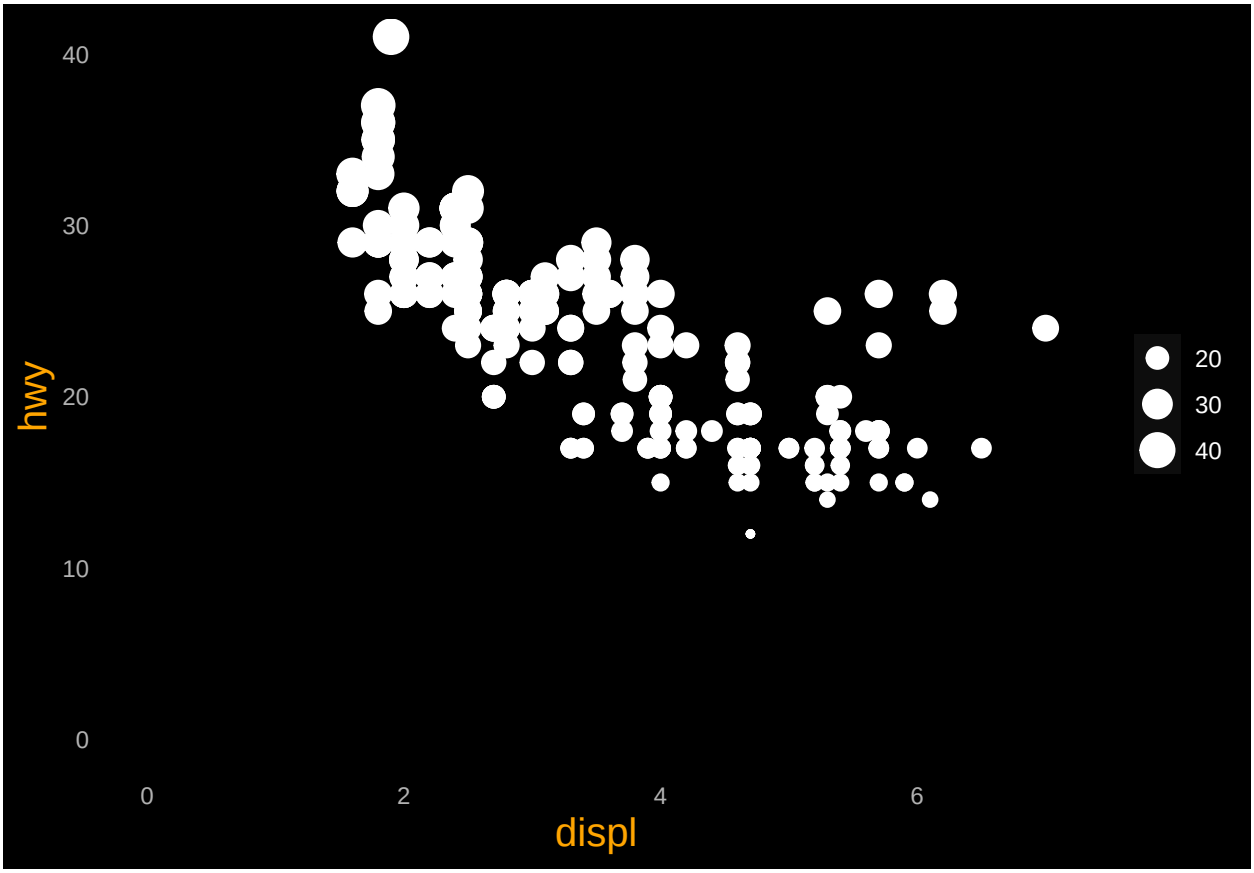
Reverse the X and Y Axis

```
p + scale_x_reverse() + scale_y_reverse()
```



- **Zoom in by defining the limits of the axis**

```
p + coord_cartesian(xlim=c(0,7), ylim=c(0, 40))
```



- #Exercise 11:
-

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)
```

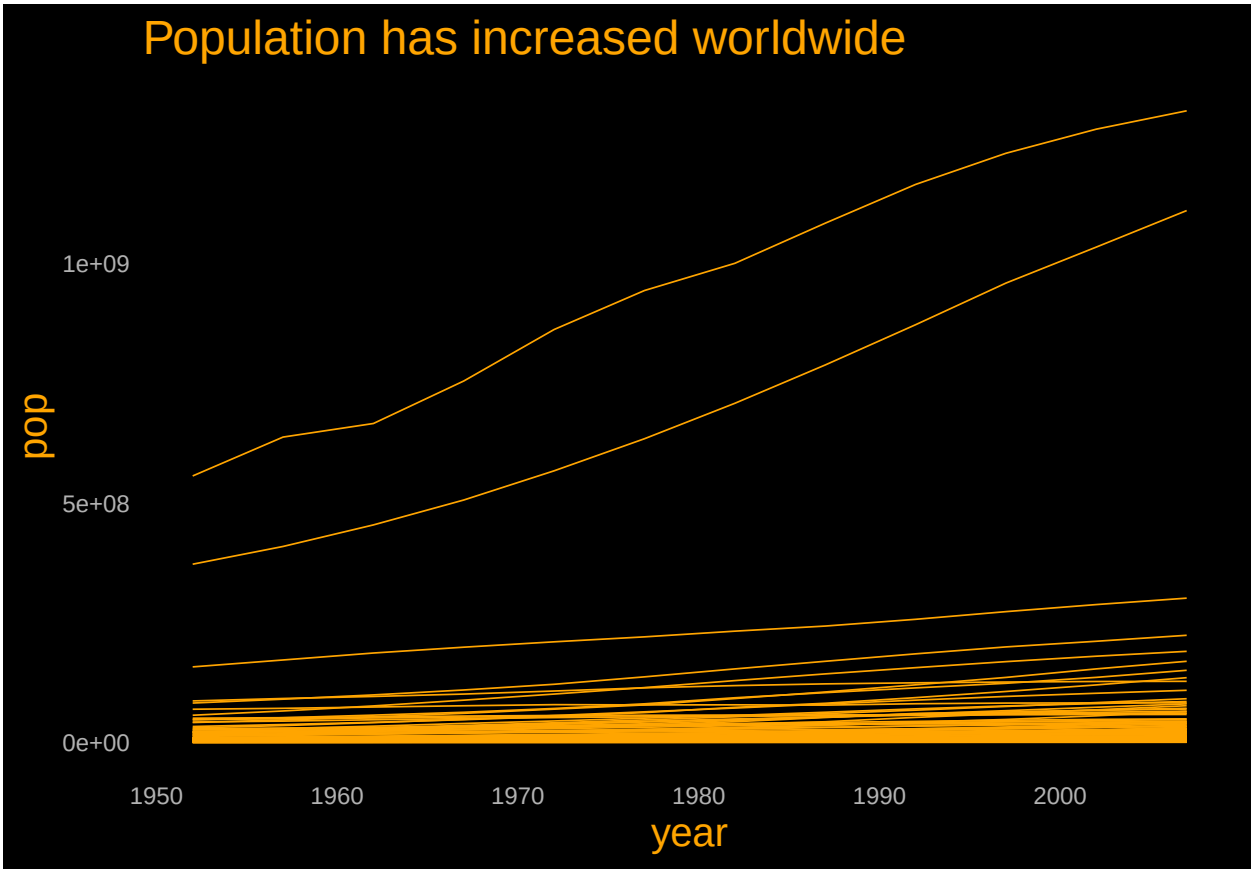
```
## tibble [1,704 x 6] (S3: tbl_df/tbl/data.frame)
##  $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ year     : int [1:1704] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...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

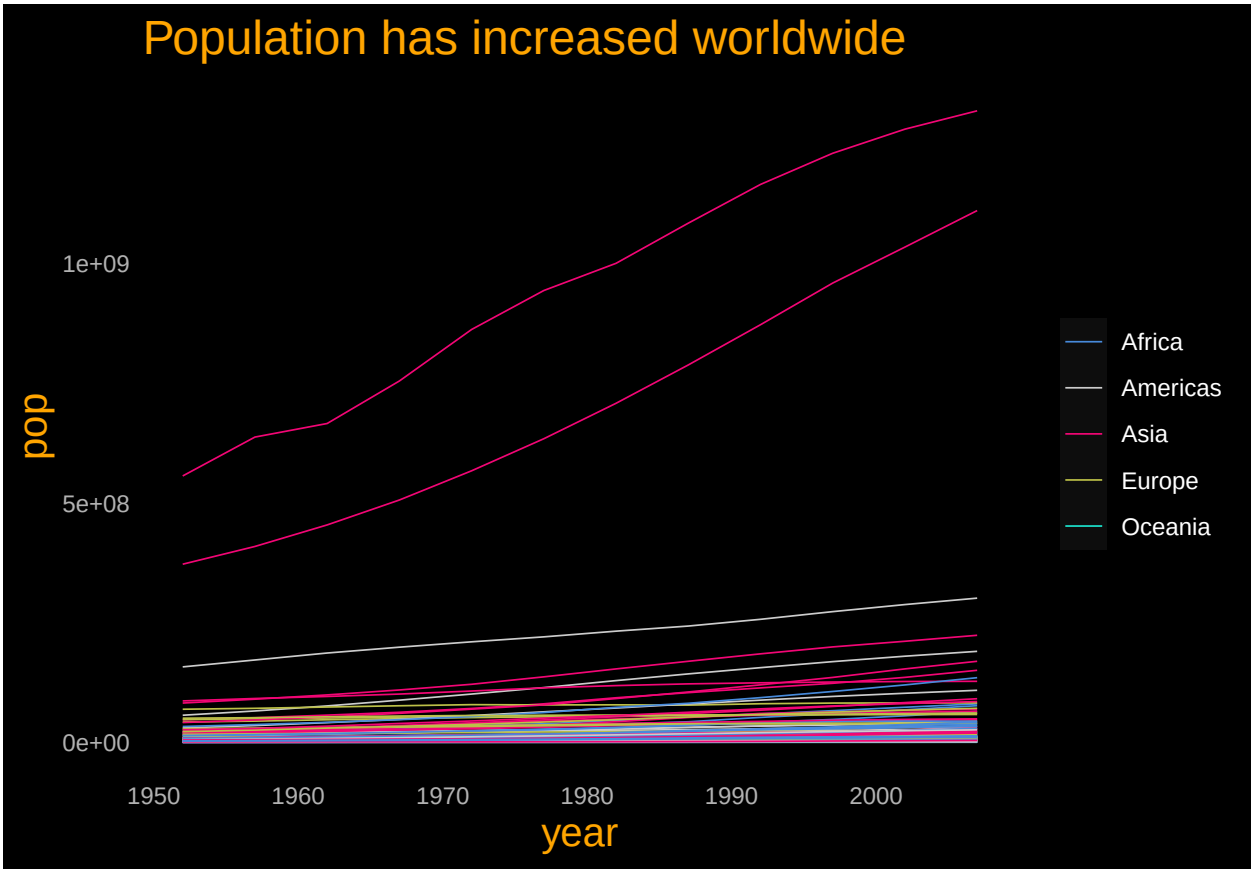
- #General trend in Population

```
ggplot(gapminder) +
  geom_line(aes (year, pop, group = country), lwd = 0.3, show.legend = FALSE, colour = trend_color) +
  labs(title = "Population has increased worldwide")
```



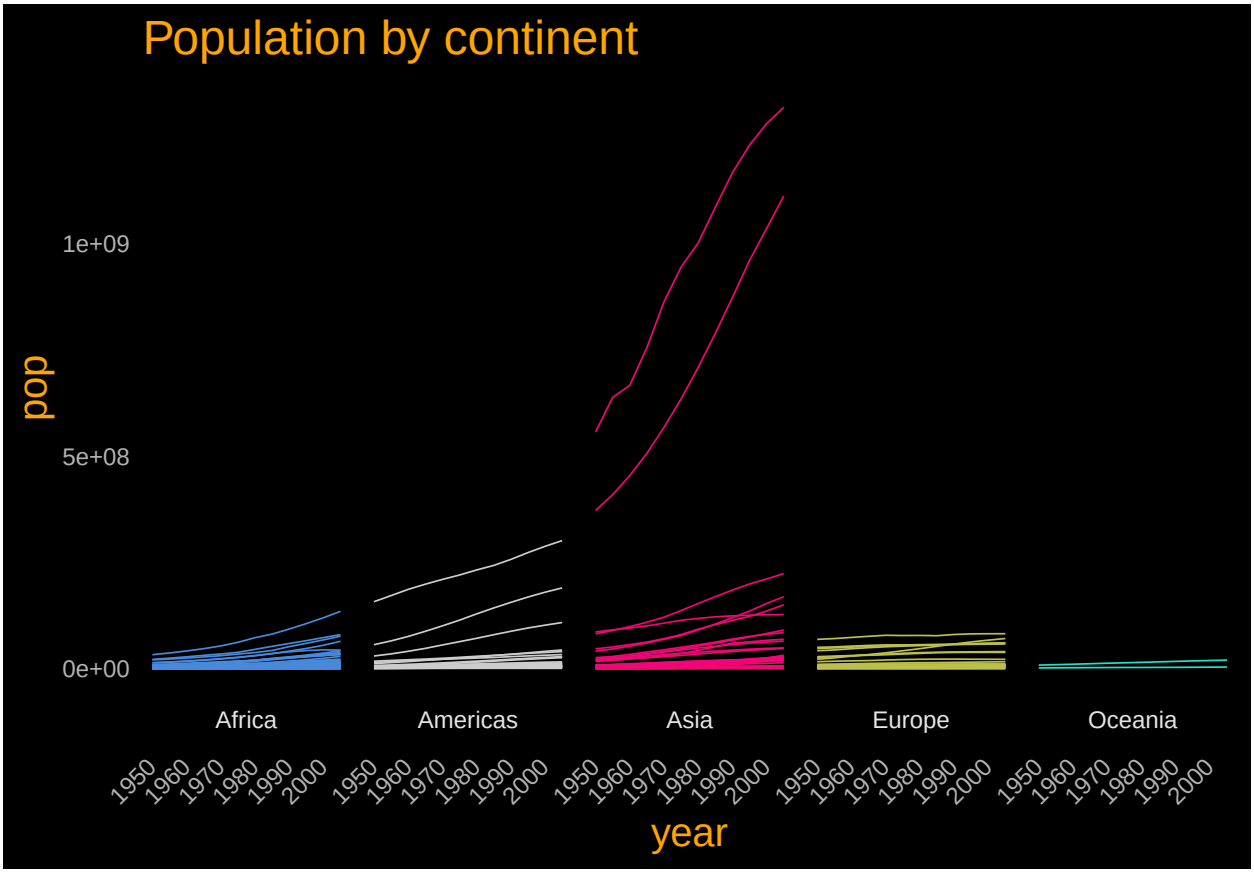
- #Checking on continents

```
ggplot(gapminder) +
  geom_line(aes (year, pop, group = country, color= continent), lwd = 0.3, show.legend = TRUE) +
  scale_color_manual(values=c("#478adb", "#cccccc", "#f20675", "#bcc048", "#1ce3cd")) +
  labs(title = "Population has increased worldwide")
```



- #Introducing a small multiple to better distinguish between continents

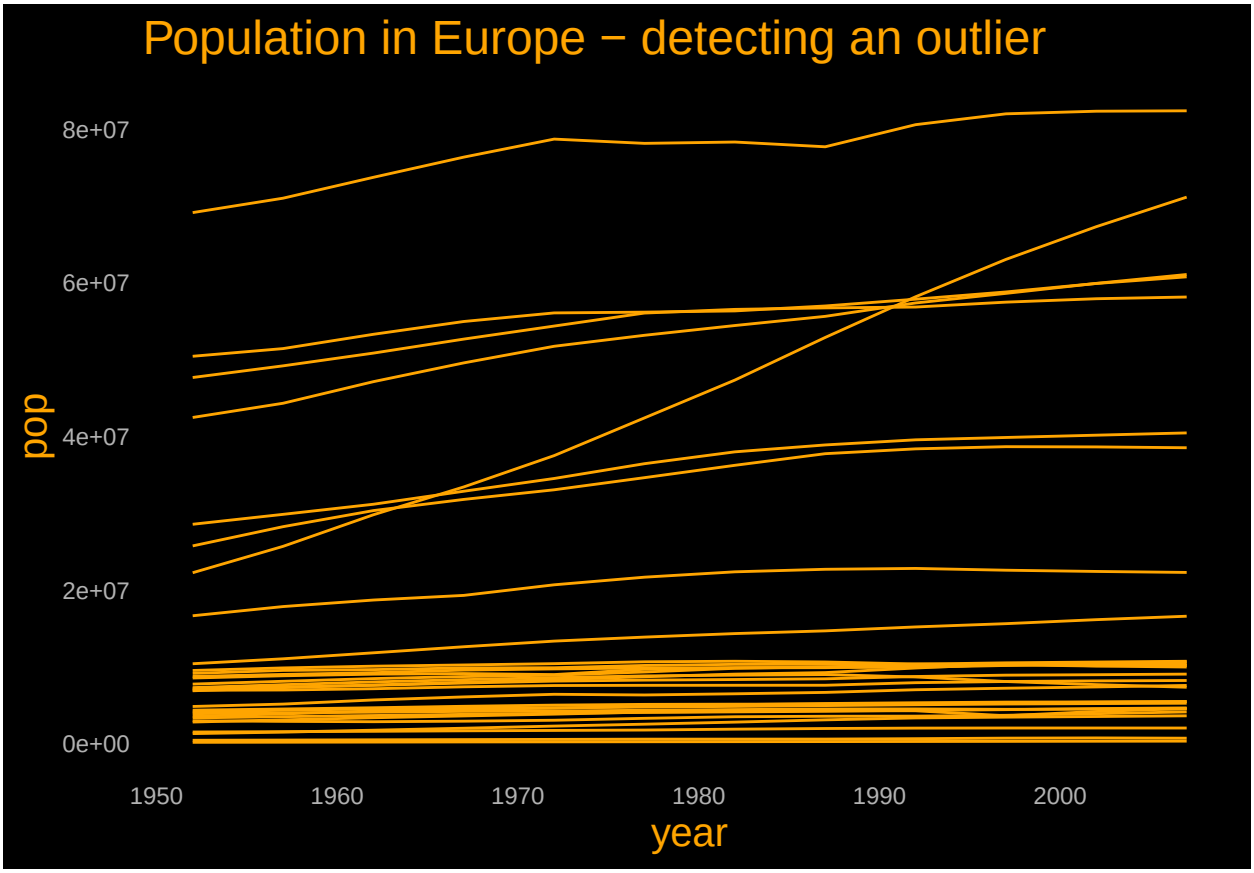
```
ggplot() +
  geom_line(data=gapminder, aes (year, pop, group = country, color = continent), lwd = 0.3, show.legend
  facet_wrap(~ continent, ncol=5, strip.position = "bottom") +
  scale_color_manual(values=c("#478adb", "#cccccc", "#f20675", "#bcc048", "#1ce3cd")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Population by continent")
```



-

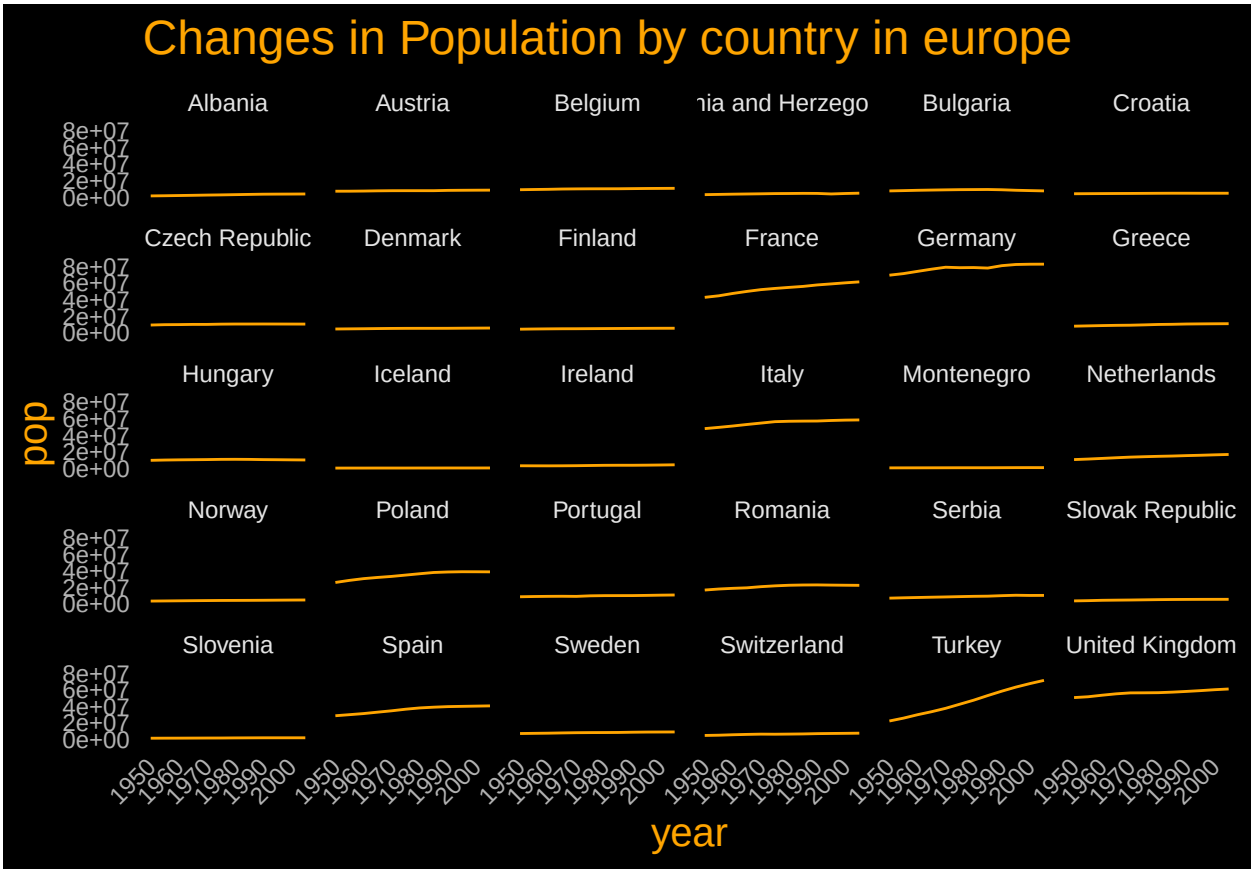
Zooming in to see only Europe

```
ggplot(subset(gapminder, continent ==  "Europe")) +
  geom_line(aes(year, pop, group = country), color= trend_color, show.legend = FALSE) +
  labs(title = "Population in Europe - detecting an outlier")
```



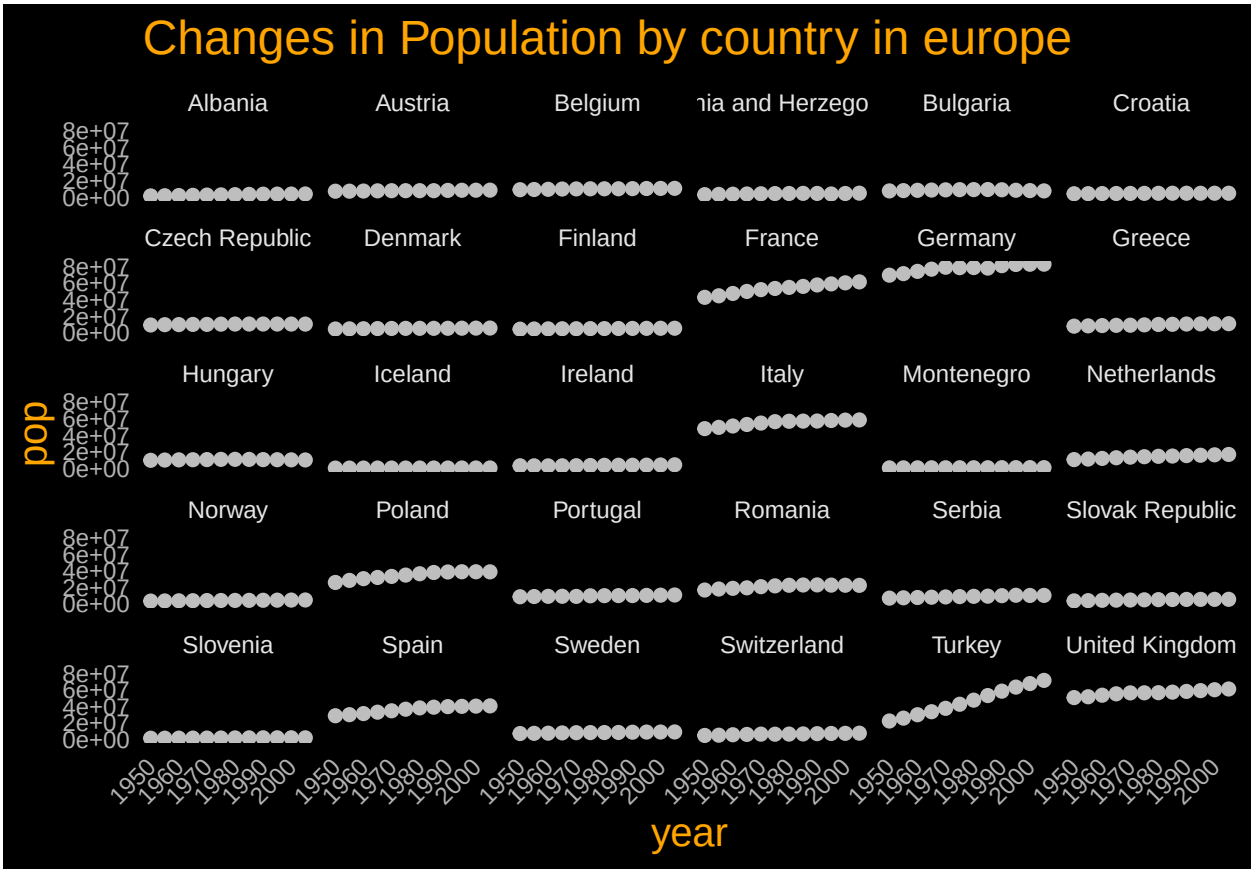
- **Select only Europe in order to understand which country is the outlier**

```
europe <- dplyr::filter(gapminder, continent == "Europe")

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



- #We can also show the trend as dots

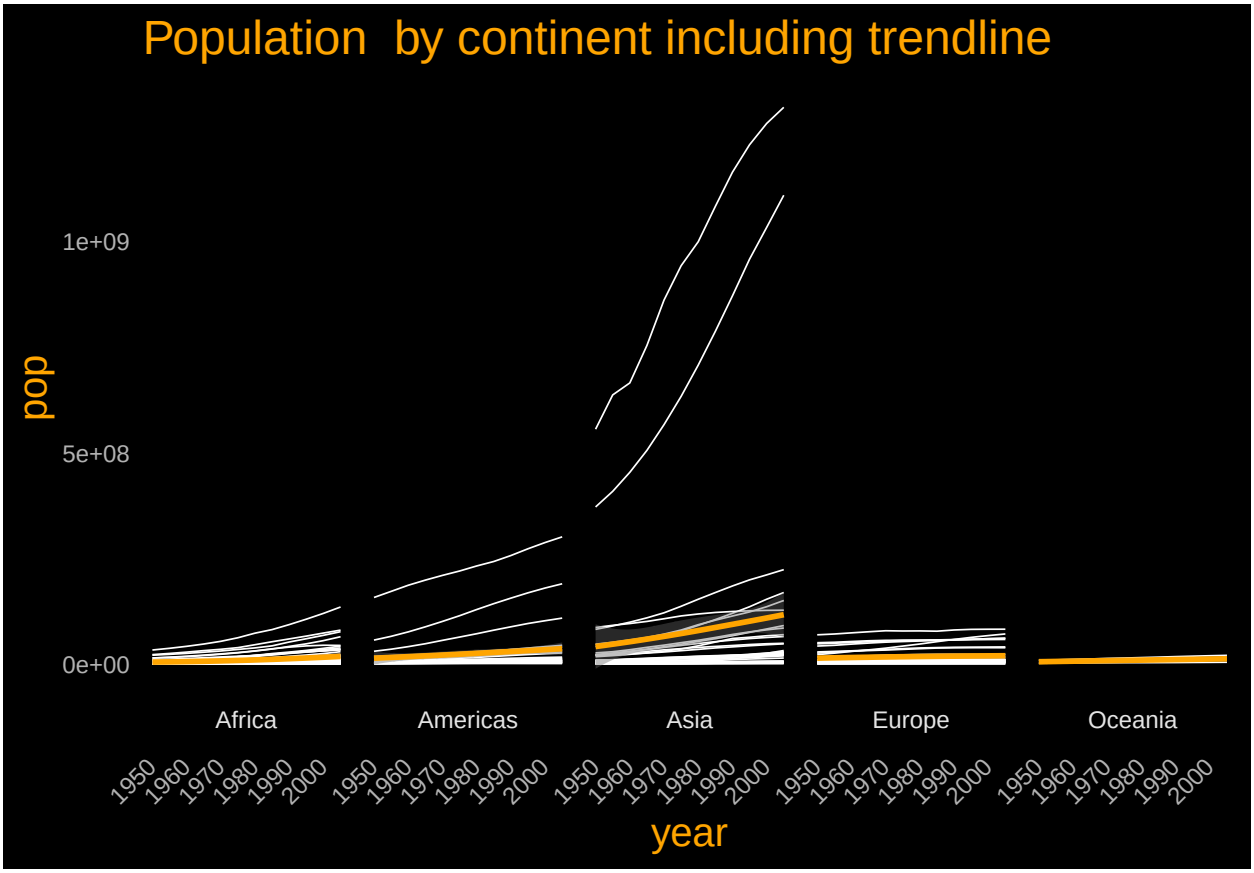
```
ggplot(europe, aes(year, pop)) +
  geom_point(color="grey", size=2) +
  facet_wrap(~country) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Changes in Population by country in europe")
```



- #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, pop, group = country), lwd = 0.3, show.legend = FALSE, color= "wh
  facet_wrap(~ continent, ncol=5, strip.position = "bottom") +
  geom_smooth(data=gapminder, aes(year, pop, group = 1), lwd = 1, method = 'loess', span = 2, se = TRUE
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Population  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, pop, group = country), alpha = 0.5
  geom_line(data=gapminder, aes (year, pop, group = country), lwd = 0.3, show.legend = FALSE, color= tr
  geom_smooth(data=gapminder, aes(year, pop, group = 1), lwd = 1, method = 'loess', span = 0.1, se = TRU
  facet_wrap(~ continent, ncol=5, strip.position = "bottom") +
  theme(strip.background = element_blank(), strip.placement = "outside") +
  theme(axis.text.x = element_blank()) +
  labs(title = "Population 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
```

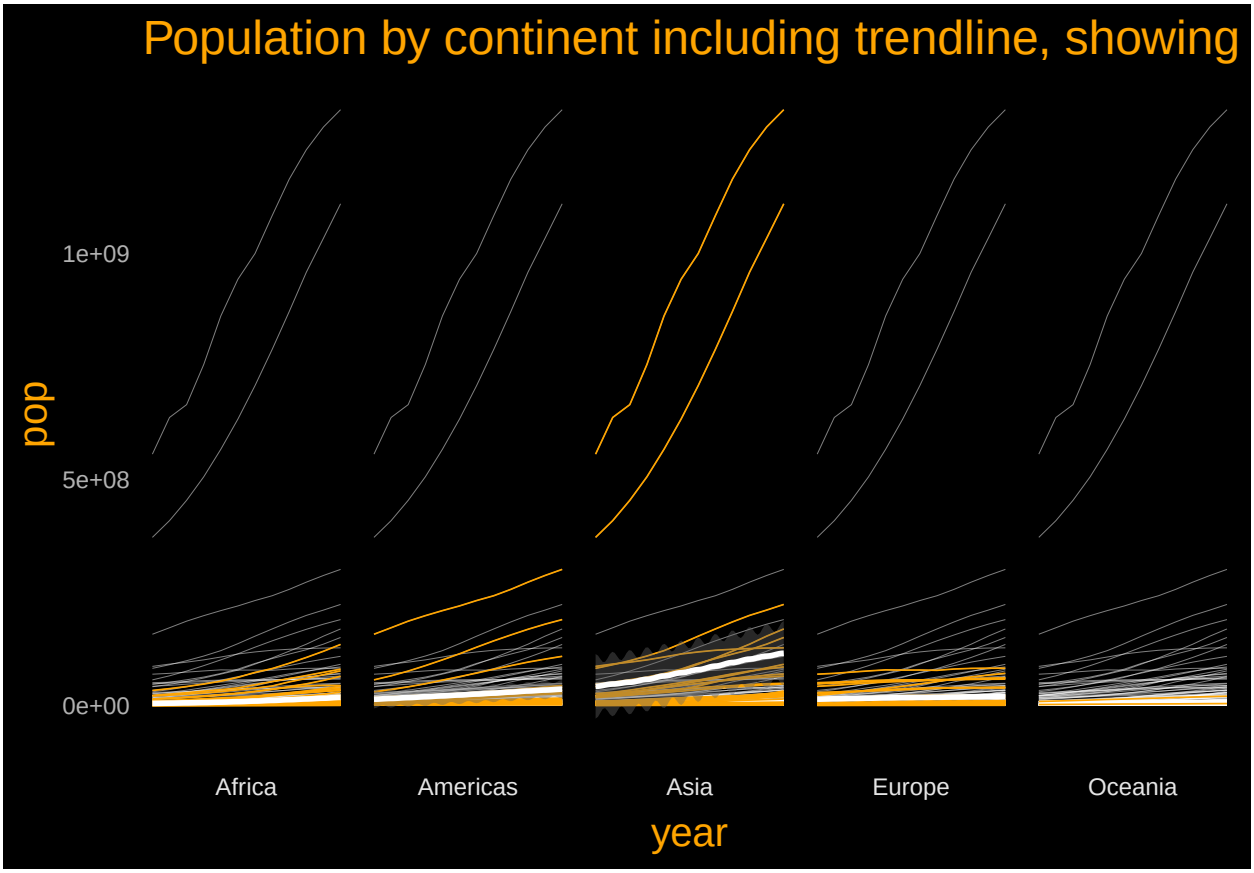
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : There are other near singularities as well. 27.826
```

```
## 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 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)
```




```
trend_color = 'orange'
```

- #Now we could filter again on Europe and have far more context

```
ggplot() +
  geom_line(data = transform(gapminder, continent = NULL), aes (year, pop, group = country), alpha = 0.5
  geom_line(data=europe, aes (year, pop, group = country), lwd = 0.3, show.legend = FALSE, color= "cyan
  geom_smooth(data=europe, aes(year, pop, group = 1), lwd = 1, method = 'loess', span = 0.1, se = TRUE,
  theme(strip.background = element_blank(), strip.placement = "outside") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Population by country in europe, including a trend line and showing all data in the back
```

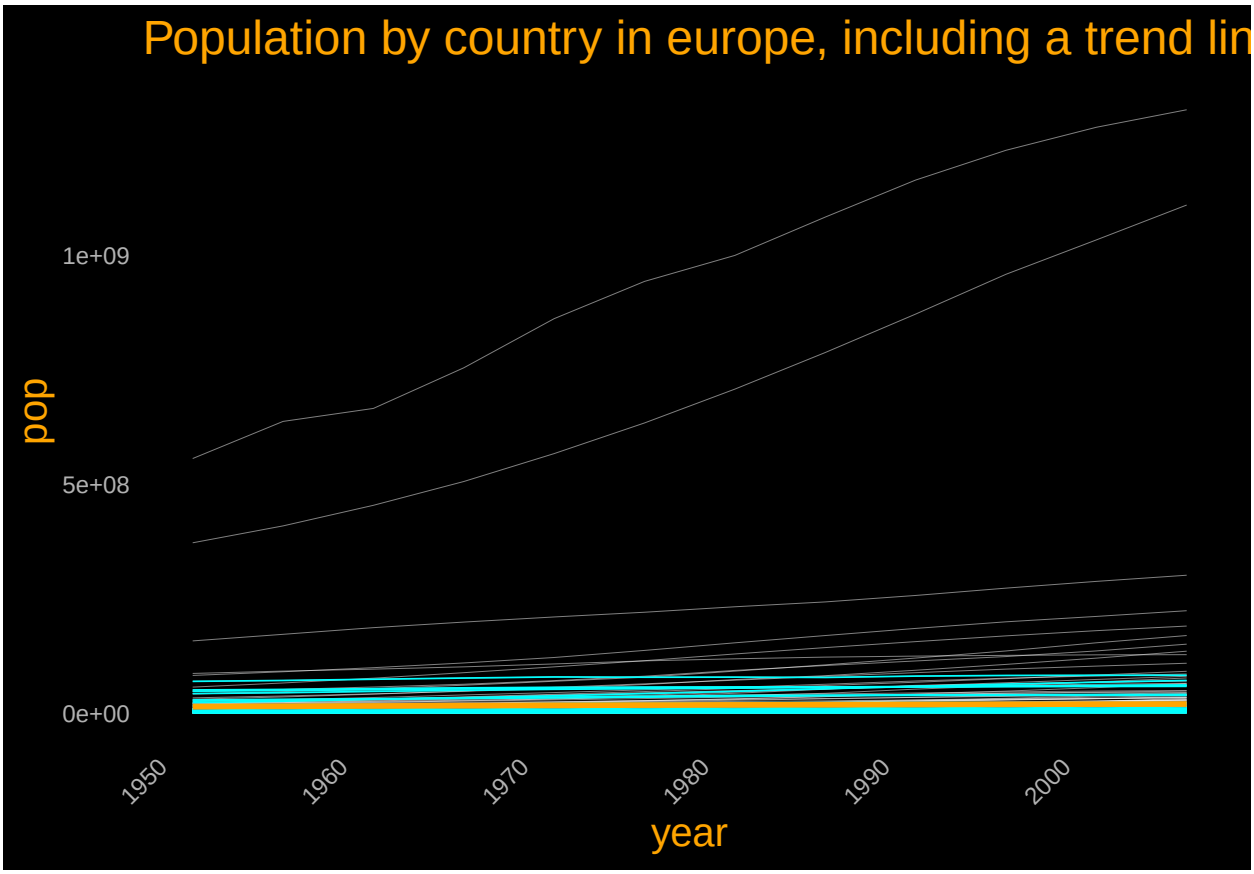
```
## `geom_smooth()` using formula 'y ~ x'

## 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
```



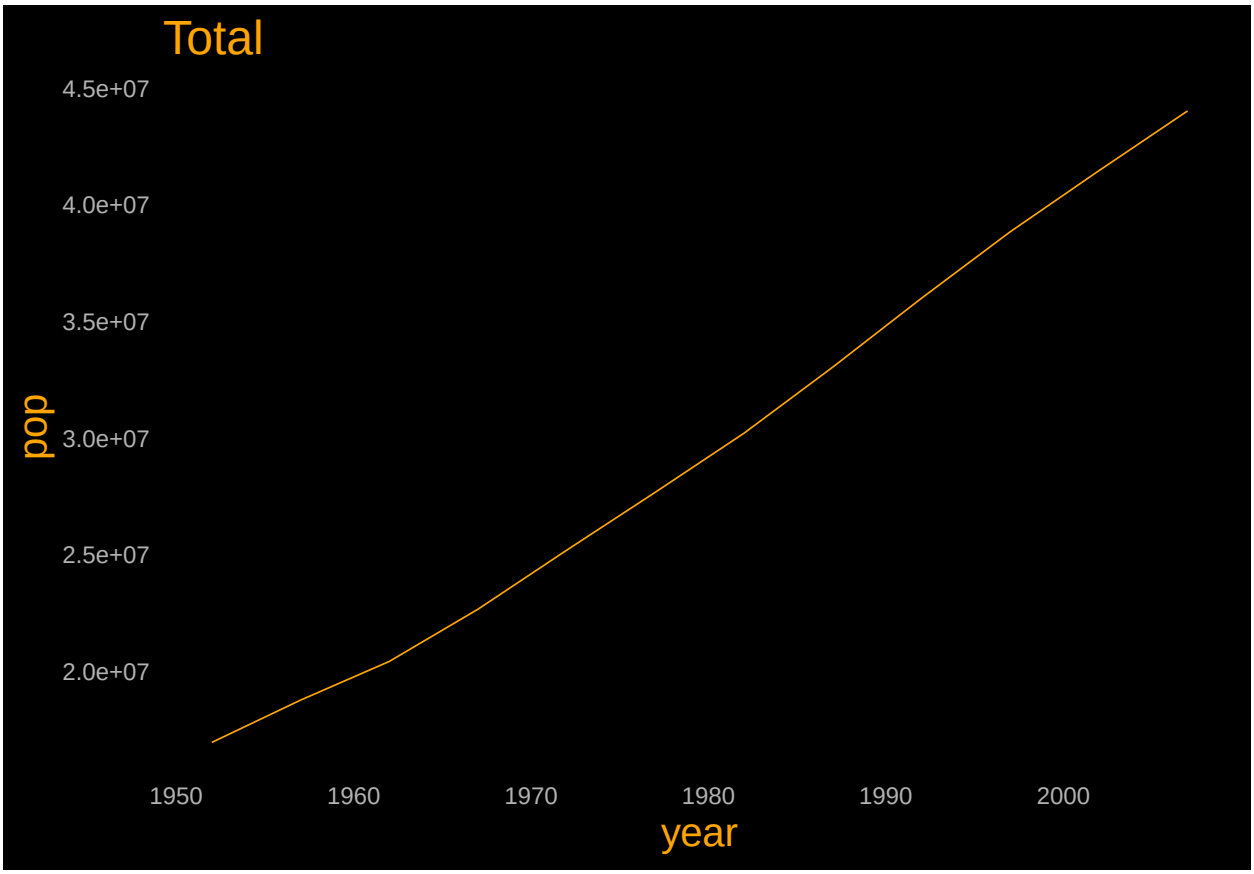
- #Showing how to add a line by aggregating the data
- #Aggregating the data

```
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
```

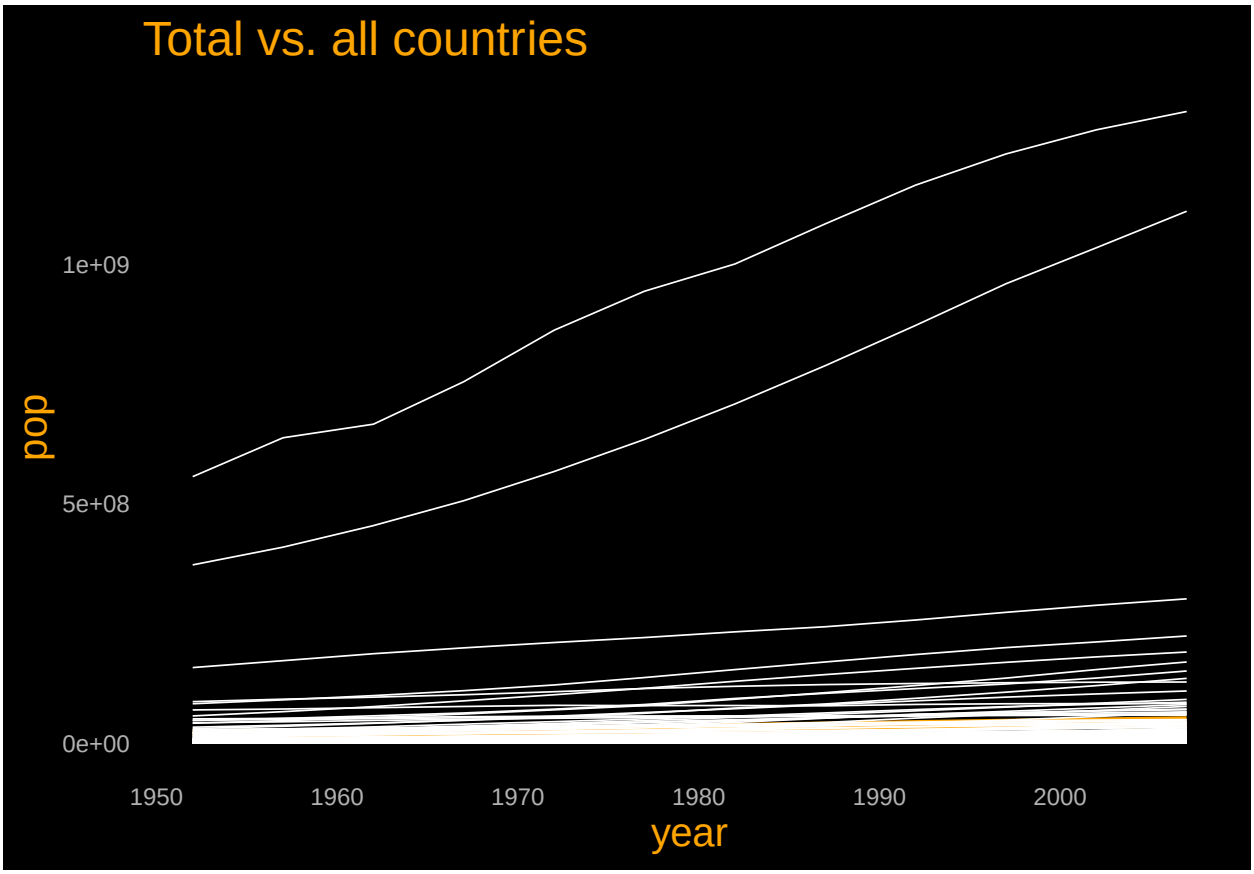
- #Make a plot with the aggregated data

```
ggplot(gapminderavg) +
  geom_line(aes (year, pop), lwd = 0.3, show.legend = FALSE, color = trend_color) +
  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, pop), lwd = 2, show.legend = FALSE, color = trend_color) +
  geom_line(data=gapminder, aes (year, pop, group = country), lwd = 0.3, show.legend = FALSE, color = "
  labs(title = "Total vs. all countries")
```



- #Exercise 12:
- #Advanced data visualization
- #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.
```

```
str(gapminder)
```

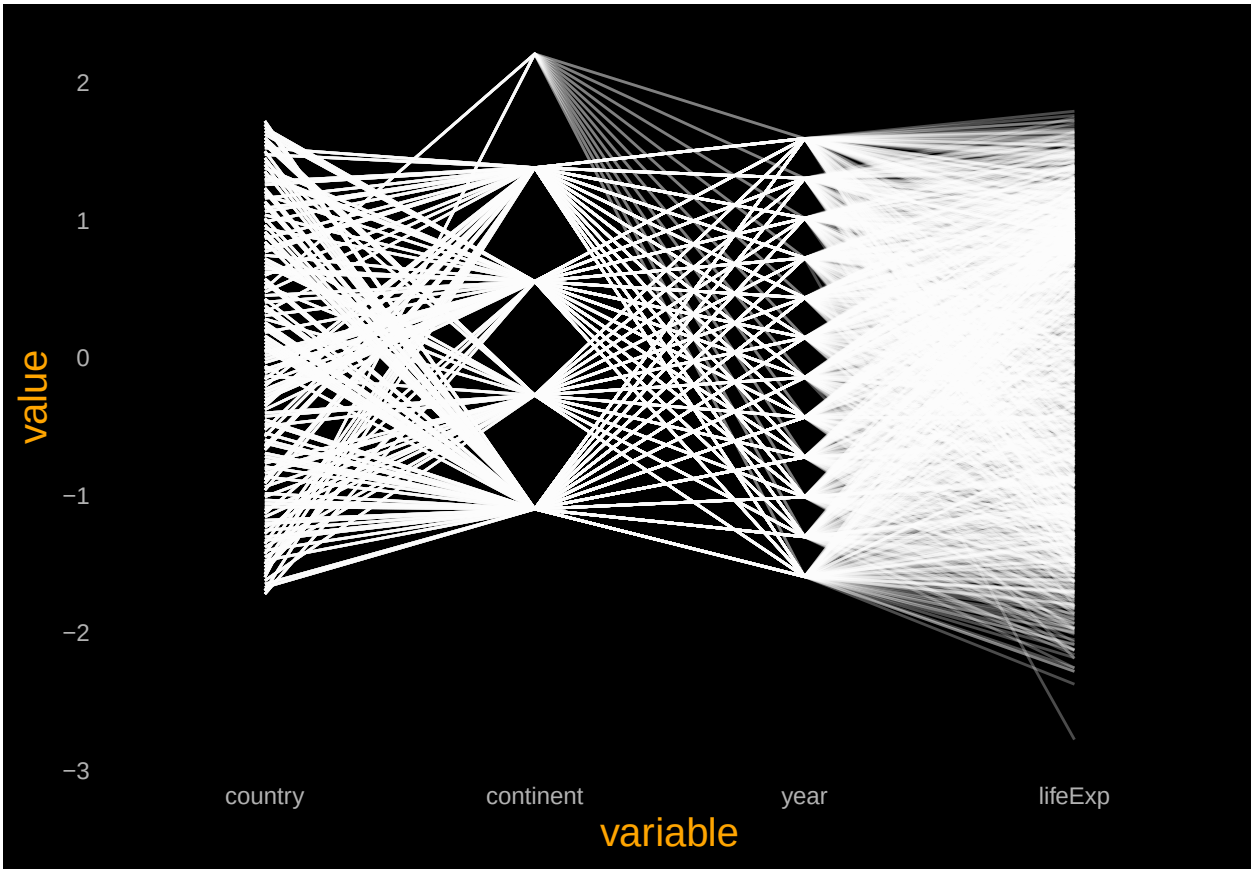
```
## tibble [1,704 x 6] (S3: tbl_df/tbl/data.frame)
##  $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ year     : int [1:1704] 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...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

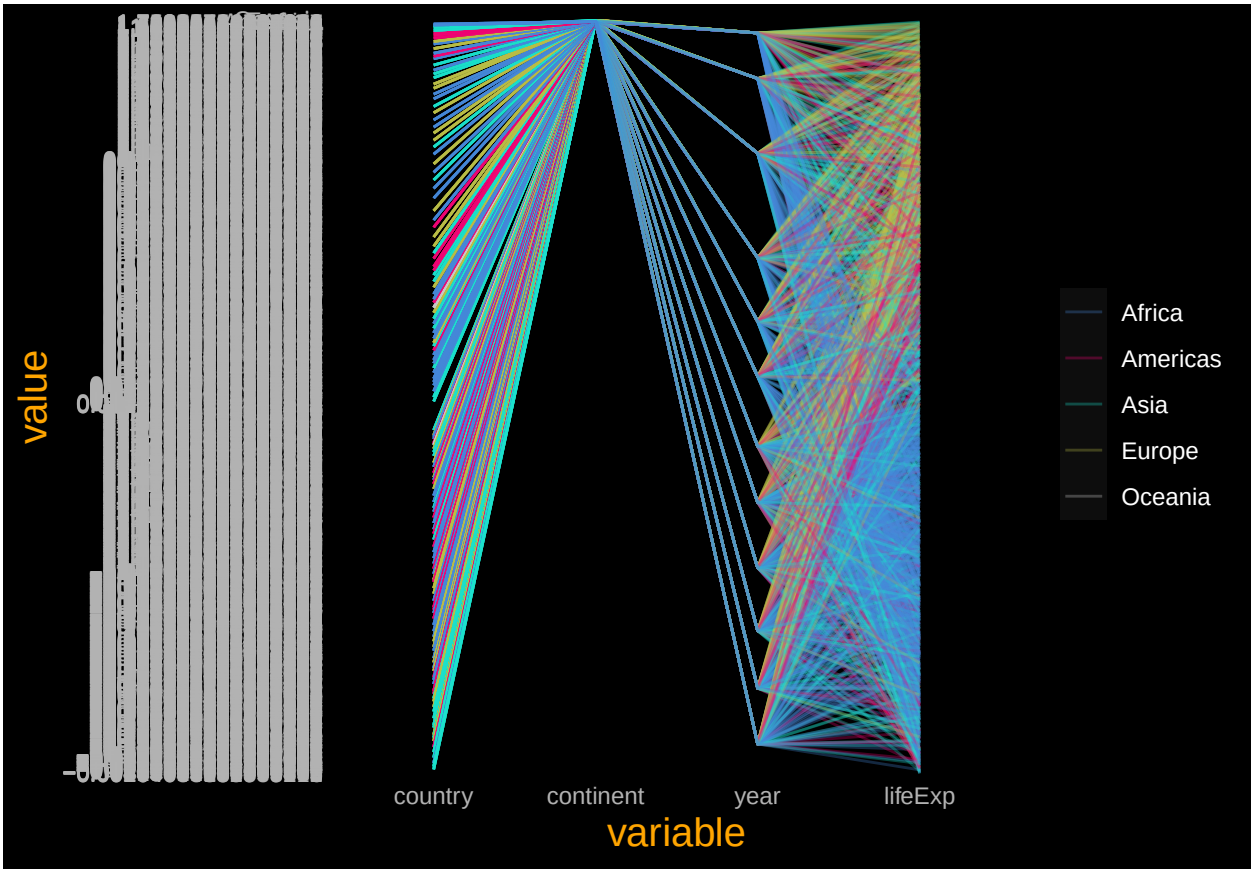
- #Simple chart

```
ggparcoord(gapminder, columns = 1:4, alphaLines = 0.3)
```



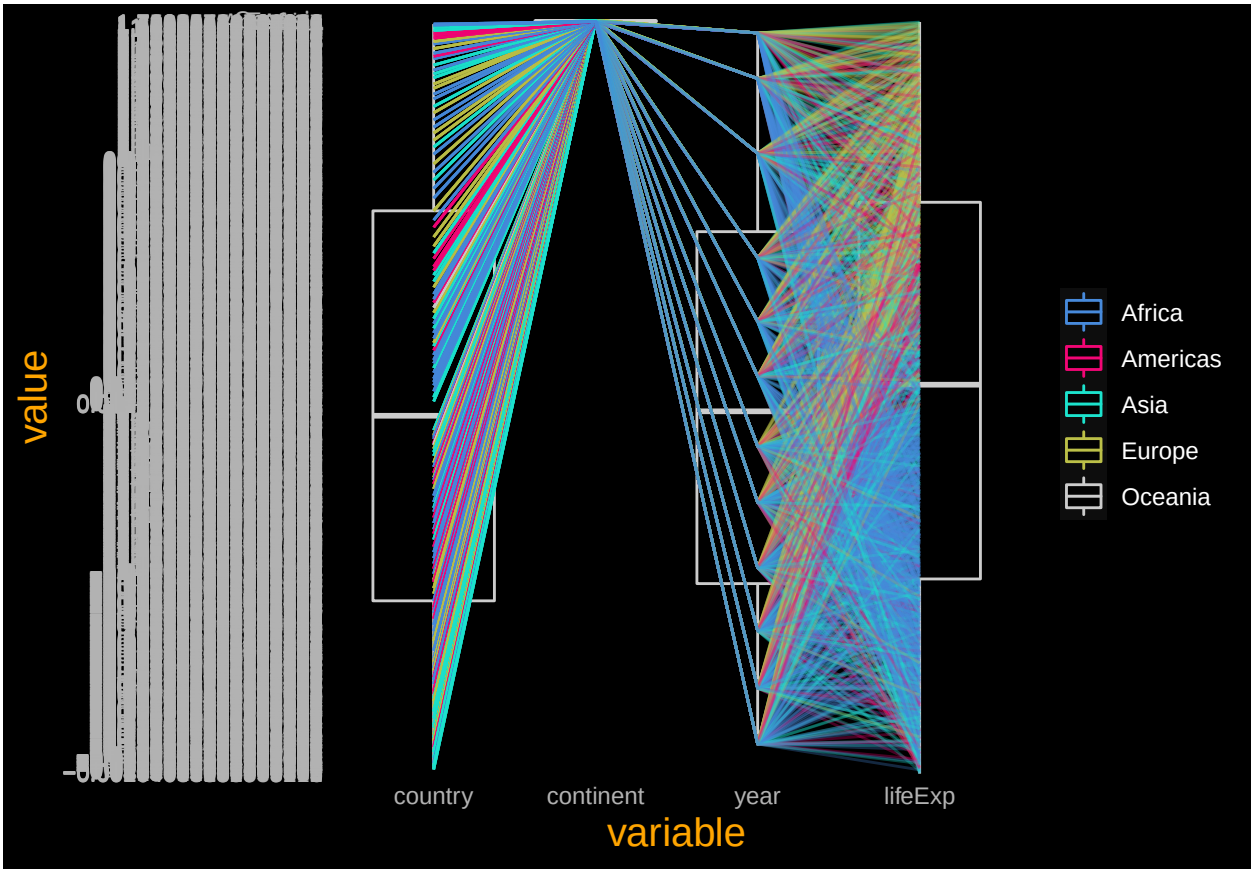
- #Simple chart, adding a color code

```
ggparcoord(gapminder, columns = 1:4,groupColumn = 2, alphaLines = 0.3) +
  scale_color_manual(values=c("#478adb", "#f20675", "#1ce3cd", "#bcc048", "#cccccc"))
```



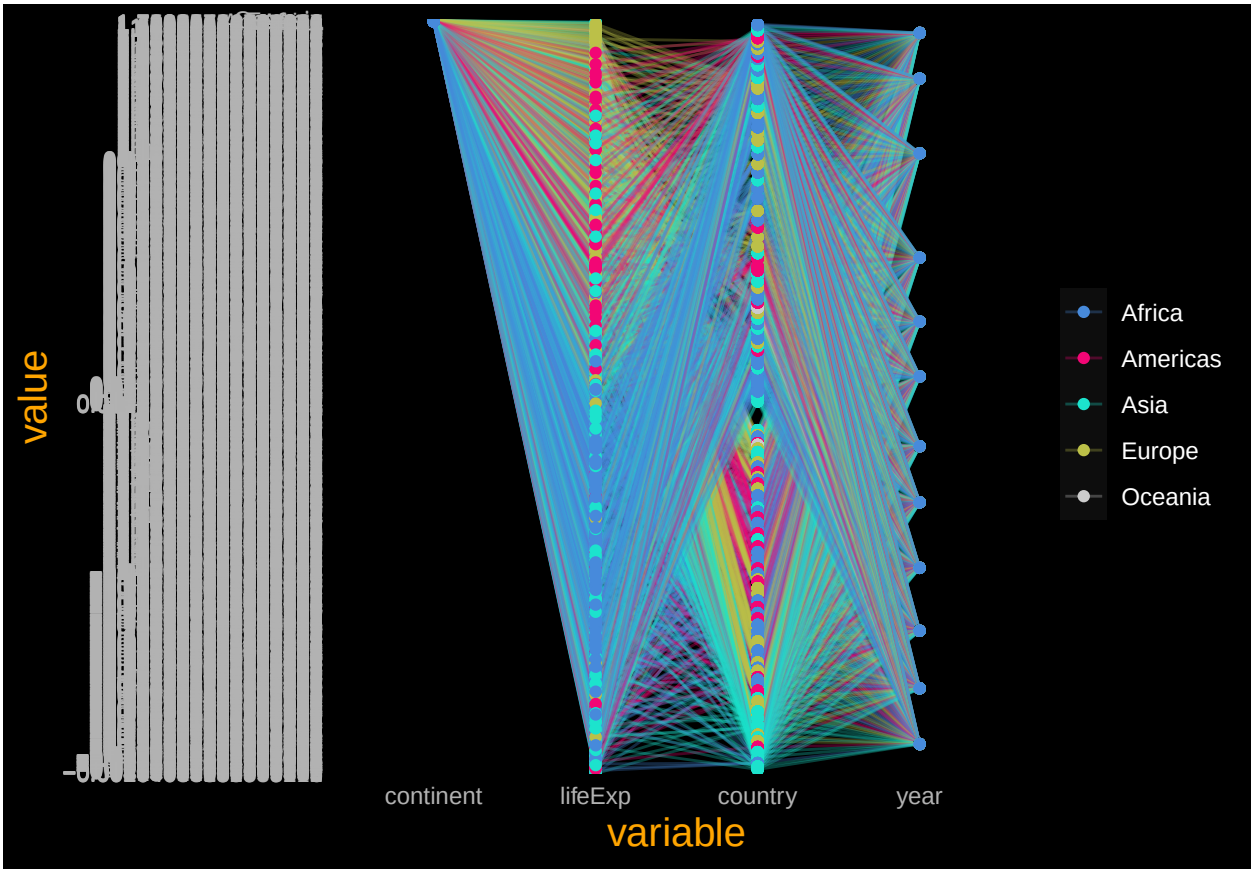
- #Simple chart, adding a color code

```
ggparcoord(gapminder, columns = 1:4, groupColumn = 2, alphaLines = 0.3, boxplot = TRUE) +
  scale_color_manual(values=c("#478adb", "#f20675", "#1ce3cd", "#bcc048", "#cccccc"))
```



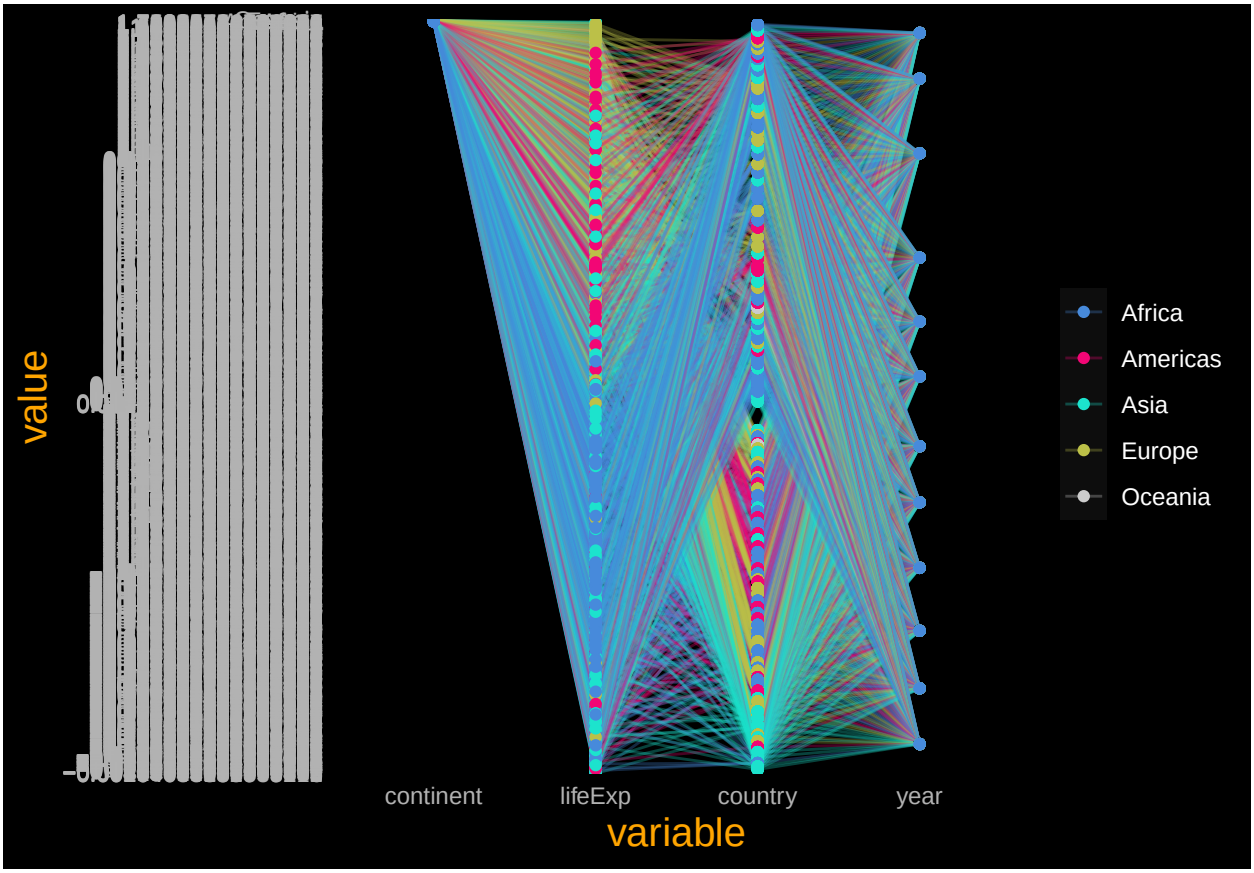
- #Showing points, changing transparency and color

```
ggparcoord(gapminder, columns = 1:4, groupColumn = 2, order = "anyClass",
           showPoints = TRUE, alphaLines = 0.3)+
  scale_color_manual(values=c("#478adb", "#f20675", "#1ce3cd", "#bcc048", "#cccccc"))
```



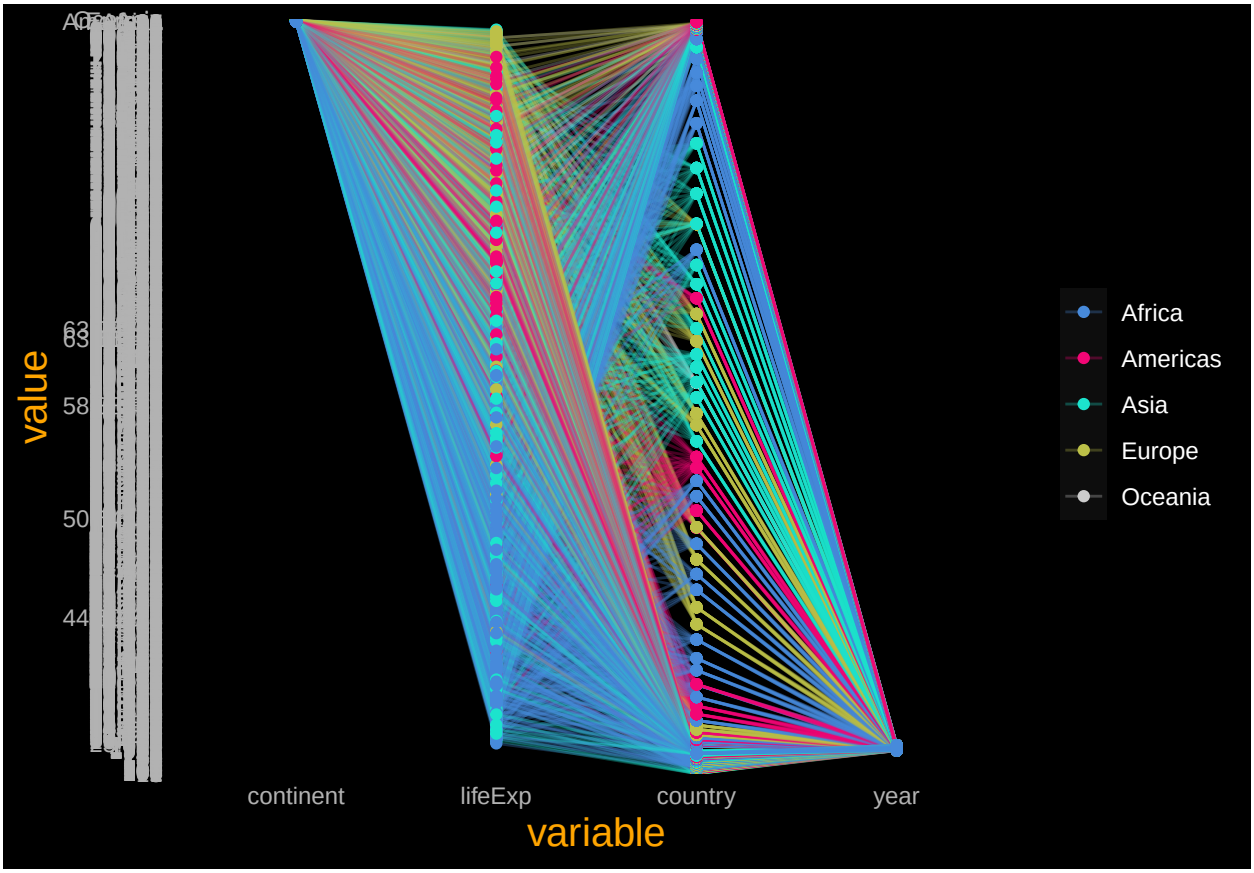
- #Showing points, changing transparency and color

```
ggparcoord(gapminder, columns = 1:4, groupColumn = 2, order = "anyClass",
           showPoints = TRUE, alphaLines = 0.3)+
  scale_color_manual(values=c("#478adb", "#f20675", "#1ce3cd", "#bcc048", "#cccccc"))
```



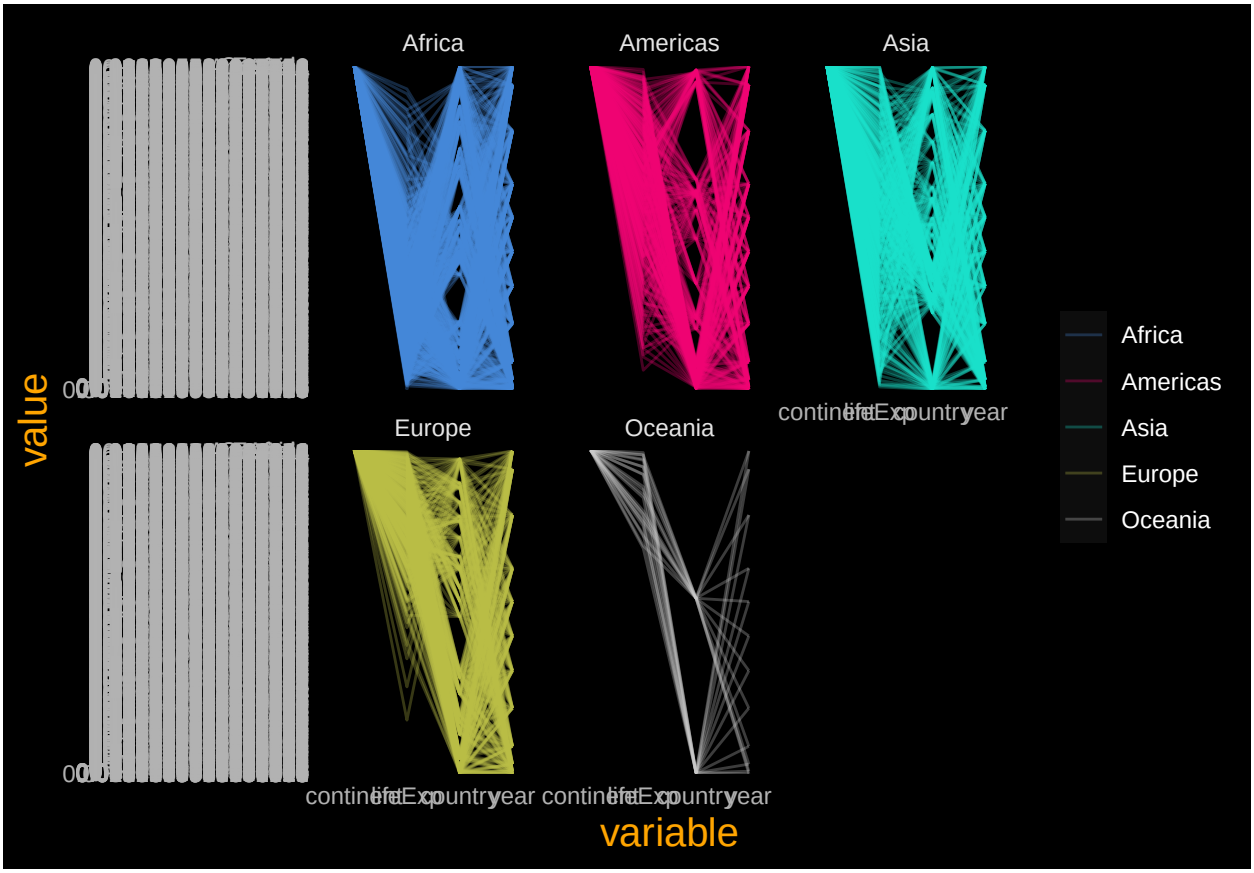
- #Different way of scaling: No scaling

```
ggparcoord(gapminder, columns = 1:4, groupColumn = 2, order = "anyClass",
           showPoints = TRUE, alphaLines = 0.3, scale="globalminmax") +
  scale_color_manual(values=c("#478adb", "#f20675", "#1ce3cd", "#bcc048", "#cccccc"))
```



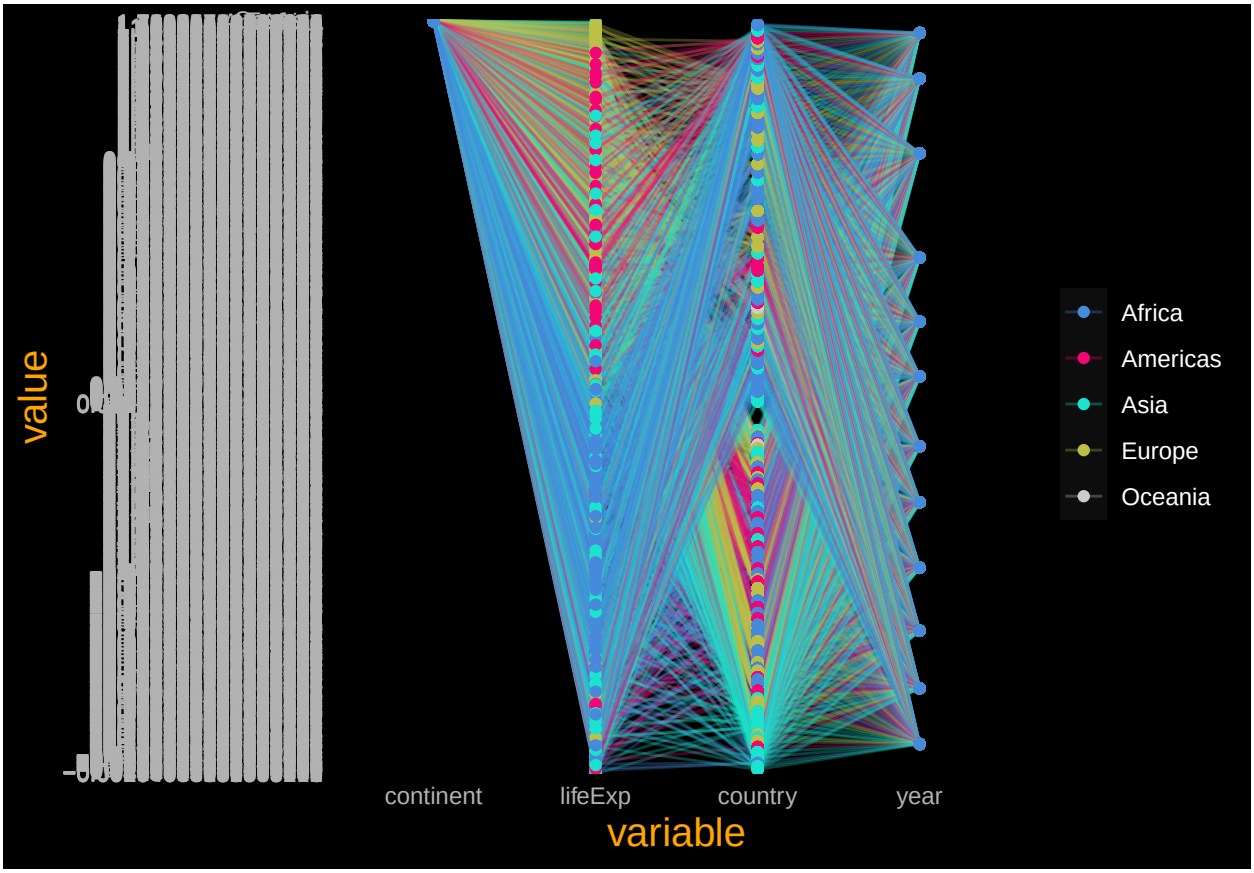
- #Different way of scaling: Standardize to Min = 0 and Max = 1

```
ggparcoord(gapminder, columns = 1:4, groupColumn = 2, order = "anyClass",
           alphaLines = 0.3, scale="uniminmax") +
  scale_color_manual(values=c("#478adb", "#f20675", "#1ce3cd", "#bcc048", "#cccccc"))+
  facet_wrap(. ~ continent)
```



- #Different way of scaling: Normalize univariately (substract mean & divide by sd)

```
ggparcoord(gapminder, columns = 1:4, groupColumn = 2, order = "anyClass",
           showPoints = TRUE, alphaLines = 0.3, scale="std") +
  scale_color_manual(values=c("#478adb", "#f20675", "#1ce3cd", "#bcc048", "#cccccc"))
```



- #Exercise 13:
- #Creating a subsample

```
years <- filter(gapminder, year %in% c(1952, 2007)) %>% select(country, continent, year, lifeExp)
```

- #Check the data

```
names(years)
```

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

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

```
## # A tibble: 10 x 4
##    country     continent  year lifeExp
##    <fct>       <fct>     <int>   <dbl>
##  1 Afghanistan Asia       1952    28.8
##  2 Afghanistan Asia       2007    43.8
##  3 Albania     Europe     1952    55.2
##  4 Albania     Europe     2007    76.4
##  5 Algeria     Africa     1952    43.1
##  6 Algeria     Africa     2007    72.3
```

```
##  7 Angola      Africa     1952    30.0
##  8 Angola      Africa     2007    42.7
##  9 Argentina   Americas   1952    62.5
## 10 Argentina   Americas   2007    75.3
```

```
str(years)
```

```
## tibble [284 x 4] (S3: tbl_df/tbl/data.frame)
##  $ country  : Factor w/ 142 levels "Afghanistan",..: 1 1 2 2 3 3 4 4 5 5 ...
##  $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 4 4 1 1 1 1 2 2 ...
##  $ year     : int [1:284] 1952 2007 1952 2007 1952 2007 1952 2007 1952 2007 ...
##  $ lifeExp  : num [1:284] 28.8 43.8 55.2 76.4 43.1 ...
```

```
summary(years)
```

```
##         country       continent        year         lifeExp
##  Afghanistan:  2   Africa  :104   Min.   :1952   Min.   :28.80
##  Albania    :  2   Americas: 50   1st Qu.:1952   1st Qu.:43.47
##  Algeria    :  2   Asia    : 66   Median :1980   Median :59.13
##  Angola     :  2   Europe  : 60   Mean   :1980   Mean   :58.03
##  Argentina  :  2   Oceania :  4   3rd Qu.:2007   3rd Qu.:72.25
##  Australia  :  2                  Max.   :2007   Max.   :82.60
##  (Other)    :272
```

- #Convert data to wide format

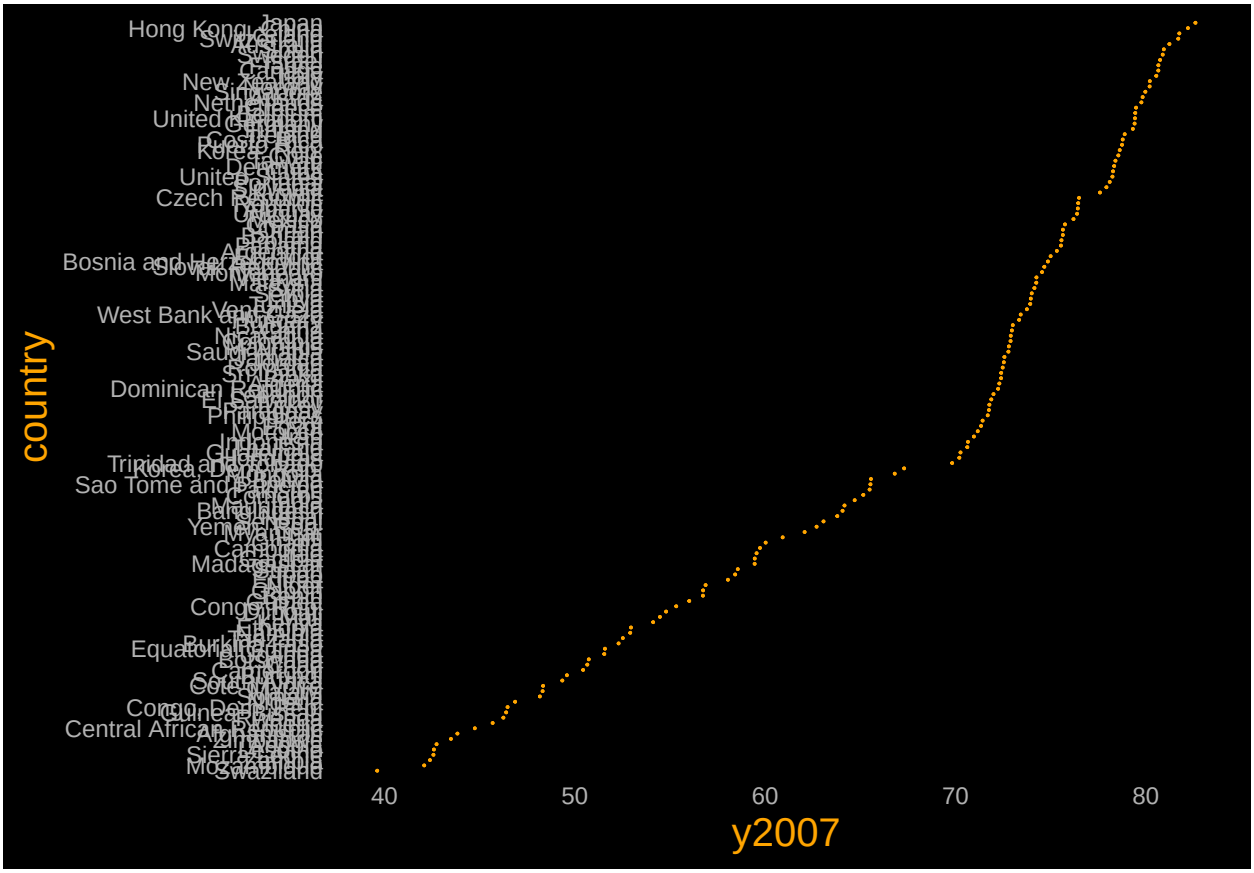
```
years2 <- spread(years, year, lifeExp)
names(years2) <- c("country", "continent", "y1952", "y2007")
```

- #Sorted by 2007

```
years3 <- arrange(years2, desc(y2007))
years3$country <- factor(years3$country, levels=rev(years3$country))
```

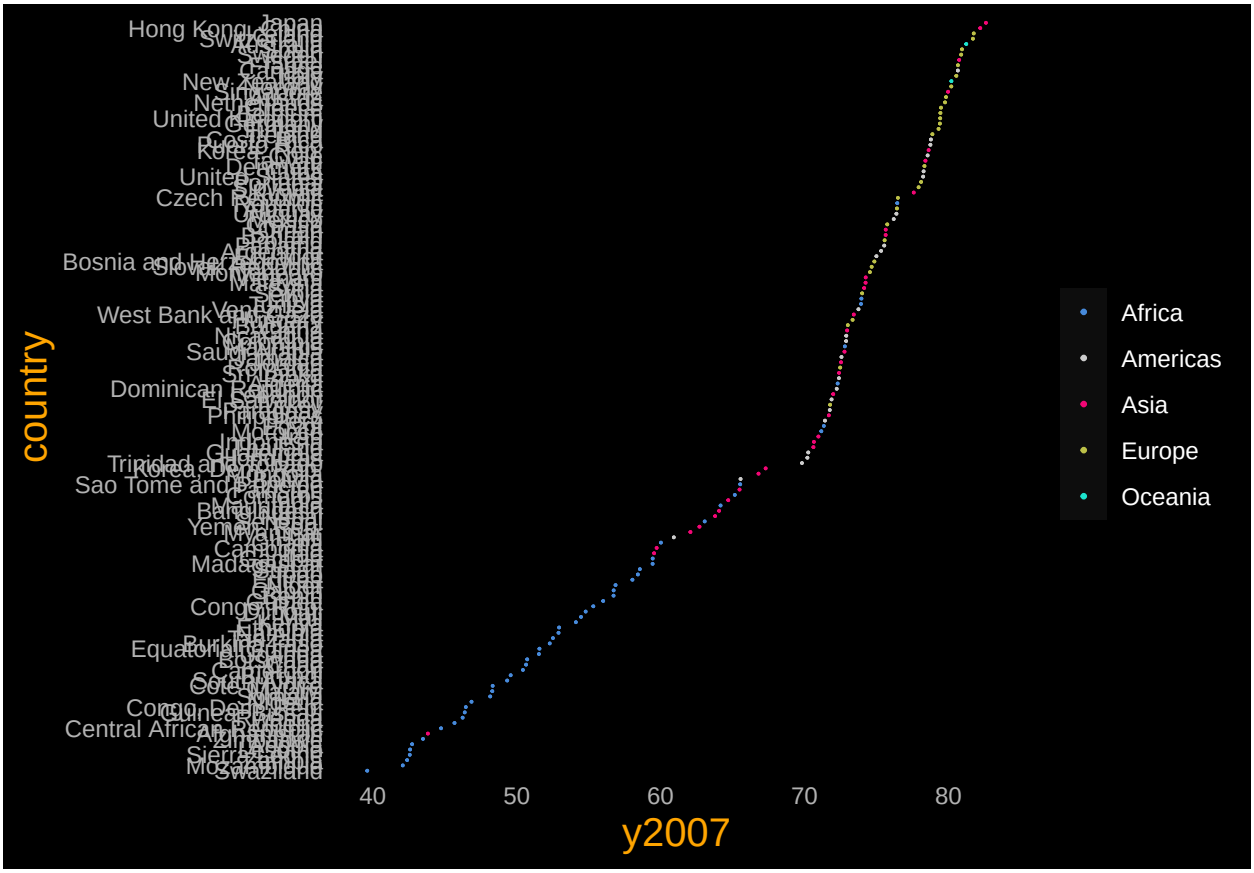
- #Create a simple dumbbell plot

```
ggplot(years3, aes(country, x=y2007, xend=y2007))+
  geom_dumbbell(colour=trend_color, size_x=0, size_xend=0)
```



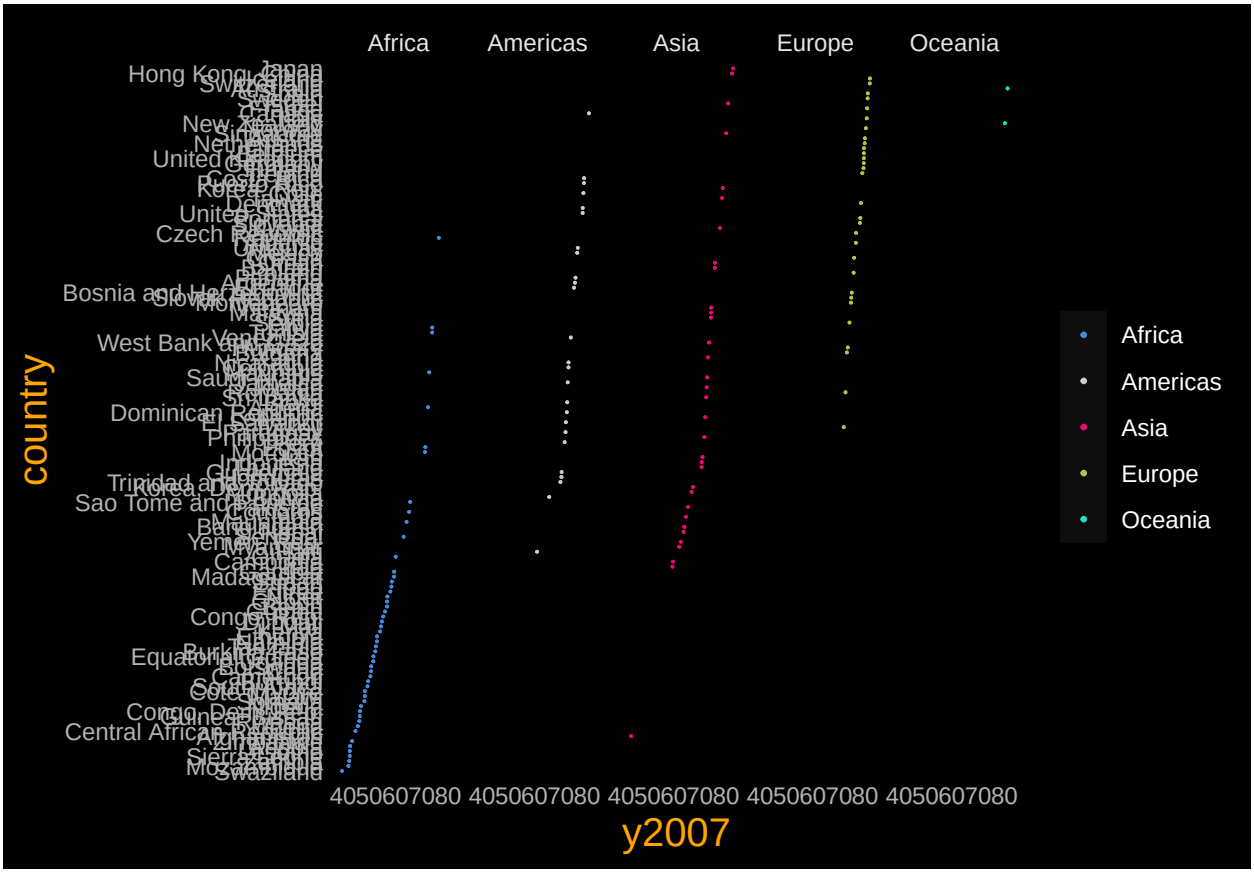
- #Create a simple dumbbell plot

```
ggplot(years3, aes(country, x=y2007, xend=y2007, color=continent))+
  scale_color_manual(values=c("#478adb", "#cccccc", "#f20675", "#bcc048", "#1ce3cd"))+
  geom_dumbbell(size_x=0, size_xend=0, dot_guide=FALSE, dot_guide_size=0.2, dot_guide_colour="white")
```



- #Create a simple dumbbell plot

```
ggplot(years3, aes(country, x=y2007, xend=y2007, color=continent))+
  scale_color_manual(values=c("#478adb", "#cccccc", "#f20675", "#bcc048", "#1ce3cd"))+
  geom_dumbbell(size_x=0, size_xend=0, dot_guide=FALSE, dot_guide_size=0.2, dot_guide_colour="white") +
  facet_wrap(. ~ continent, ncol=5)
```



- #Creating a subsample

```
asia2 <- filter(gapminder, continent == "Asia" & year %in% c(1952, 2007)) %>% select(country, year, life
```

- #Checking

```
head(asia2, n=10)
```

```
## # A tibble: 10 x 3
##    country      year lifeExp
##    <fct>       <int>   <dbl>
##  1 Afghanistan  1952    28.8
##  2 Afghanistan  2007    43.8
##  3 Bahrain      1952    50.9
##  4 Bahrain      2007    75.6
##  5 Bangladesh   1952    37.5
##  6 Bangladesh   2007    64.1
##  7 Cambodia     1952    39.4
##  8 Cambodia     2007    59.7
##  9 China        1952    44
## 10 China        2007    73.0
```

- #Convert data to wide format

```
asia3 <- spread(asia2, year, lifeExp)
names(asia3) <- c("country", "y1952", "y2007")
```

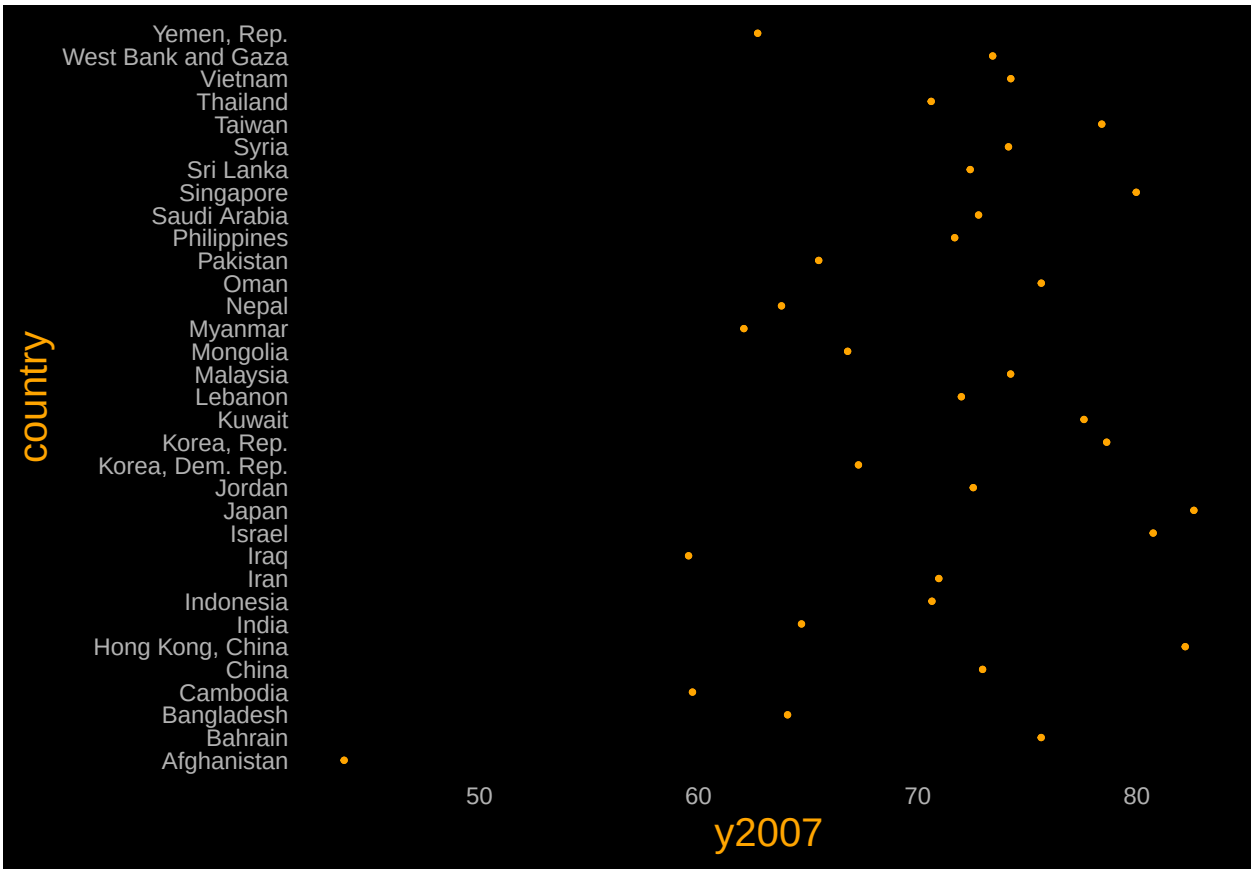
- #Checking

```
head(asia3, n=10)
```

```
## # A tibble: 10 x 3
##    country          y1952 y2007
##    <fct>            <dbl> <dbl>
##  1 Afghanistan       28.8  43.8
##  2 Bahrain           50.9  75.6
##  3 Bangladesh        37.5  64.1
##  4 Cambodia          39.4  59.7
##  5 China             44    73.0
##  6 Hong Kong, China  61.0  82.2
##  7 India             37.4  64.7
##  8 Indonesia         37.5  70.6
##  9 Iran              44.9  71.0
## 10 Iraq              45.3  59.5
```

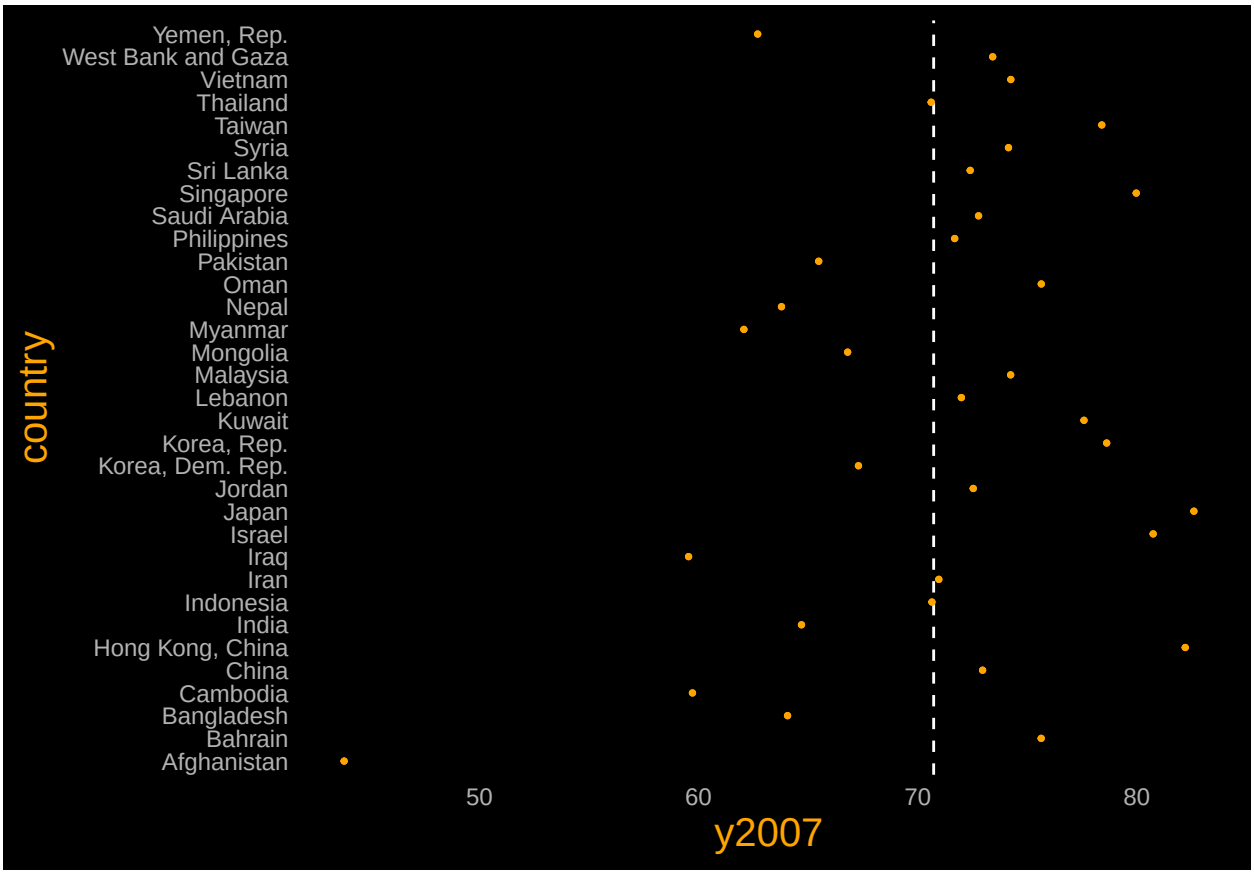
- #Create a simple dumbbell plot

```
ggplot(asia3, aes(country, x = y2007, xend = y2007))+
  geom_dumbbell(color=trend_color)
```



- #Create a simple dumbbell plot

```
ggplot(asia3, aes(country, x = y2007, xend = y2007)) +
  geom_vline(xintercept=mean(asia3$y2007), color= "white", linetype = "dashed") +
  geom_dumbbell(color=trend_color)
```



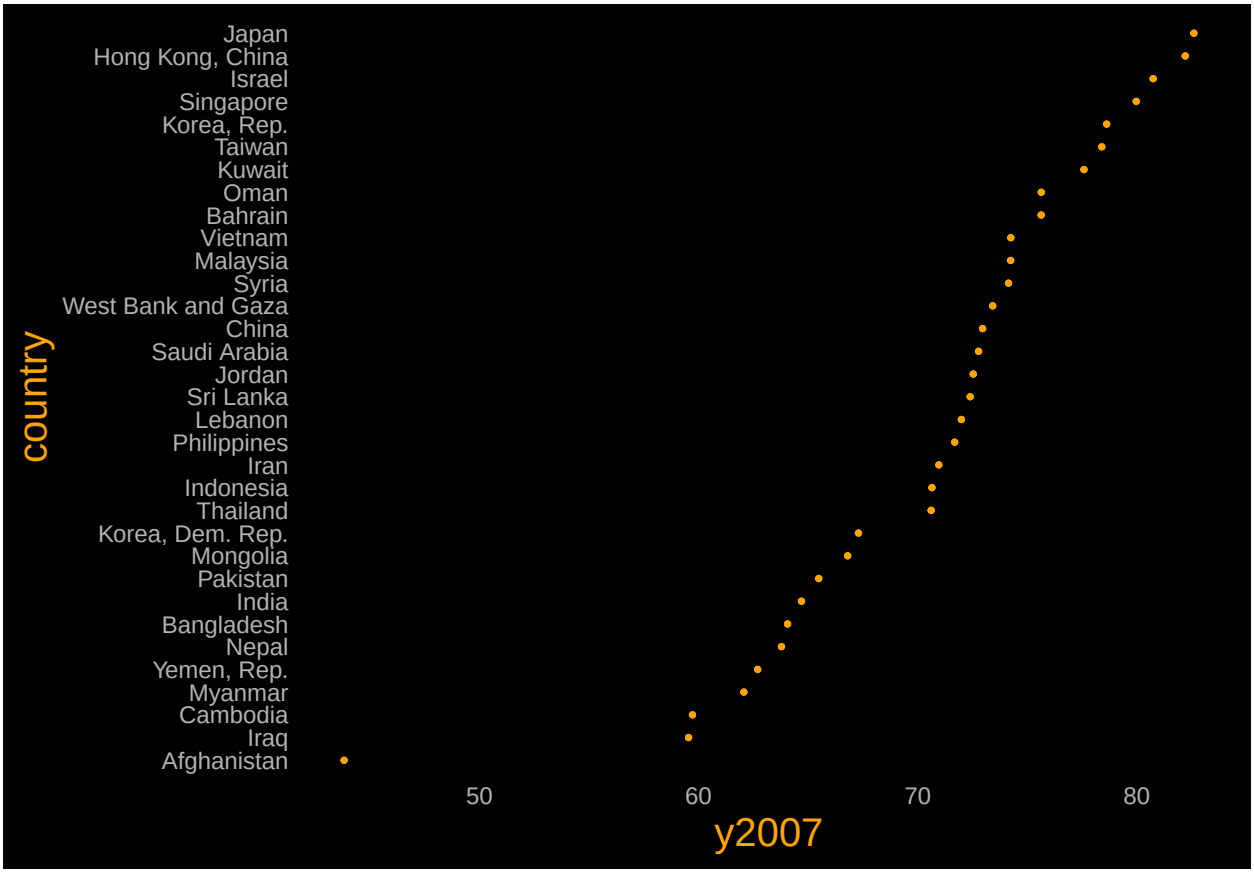
- #Normally what we want is a sorted dumbbell plot

-#Sorted by 2007

```
asia4 <- arrange(asia3, desc(y2007))
asia4$country <- factor(asia4$country, levels=rev(asia4$country))
```

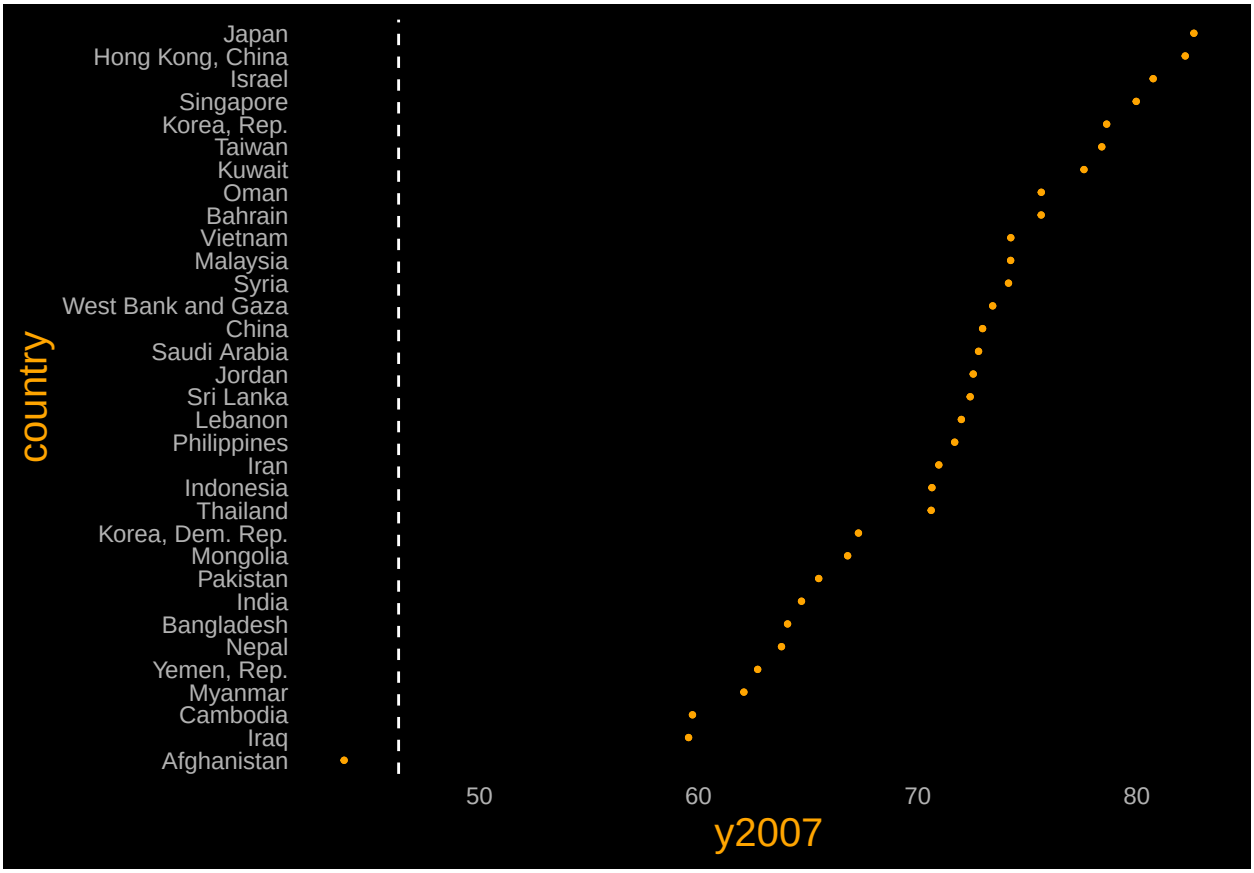
- #Create dumbbell plot now sorted

```
ggplot(asia4, aes(country, x = y2007, xend = y2007)) +
  geom_dumbbell(color=trend_color)
```



- #Create dumbbell plot now sorted

```
ggplot(asia4, aes(country, x = y2007, xend = y2007)) +
  geom_vline(xintercept=mean(asia4$y1952), color= "white", linetype = "dashed")+
  geom_dumbbell(color=trend_color)
```

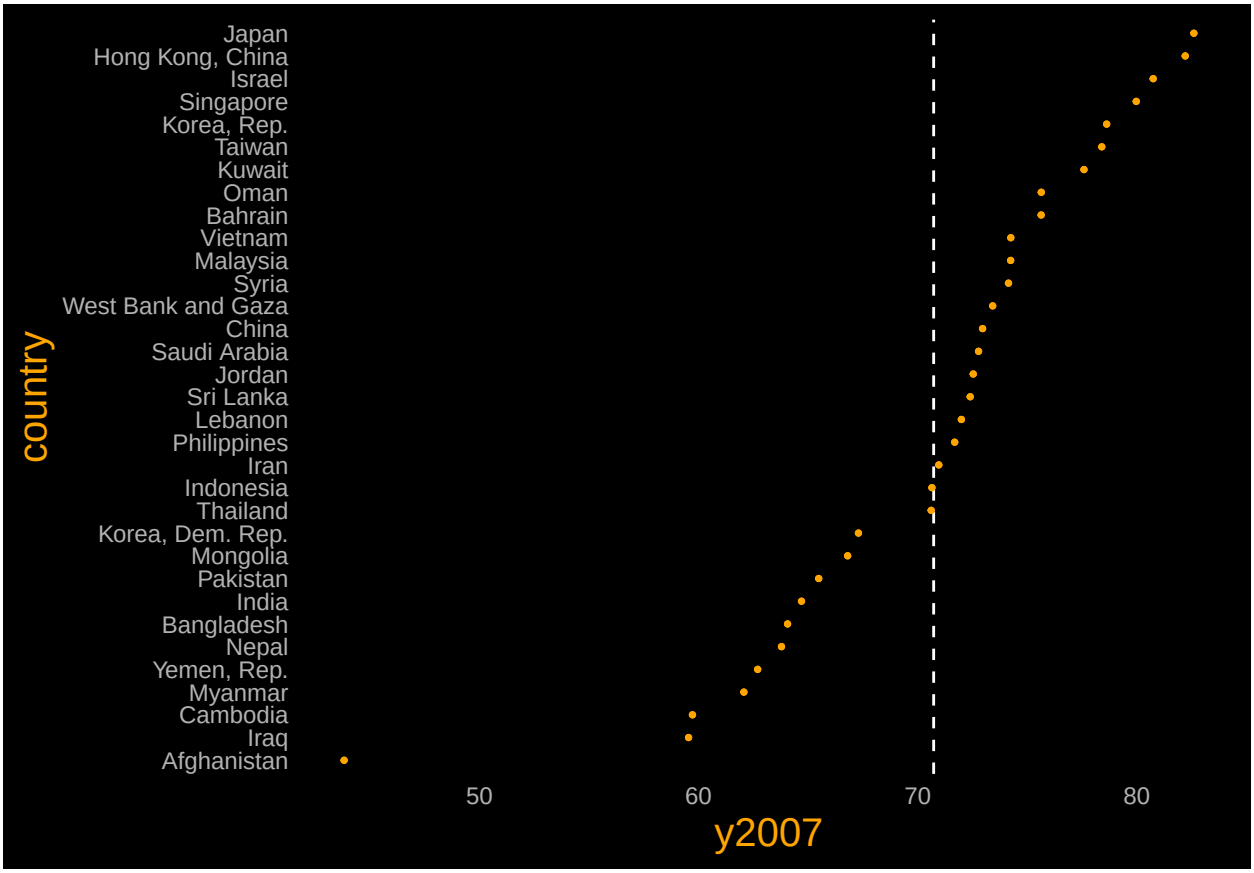


- #Sorted by 2007

```
asia5 <- arrange(asia3, desc(y2007))
asia5$country <- factor(asia5$country, levels=rev(asia5$country))
```

- #Create dumbbell plot now sorted

```
ggplot(asia5, aes(country, x = y2007, xend = y2007)) +
  geom_vline(xintercept=mean(asia5$y2007), color= "white", linetype = "dashed") +
  geom_dumbbell(color=trend_color)
```



- #Exercise 14:
- #Waffle chart

```
?waffle
```

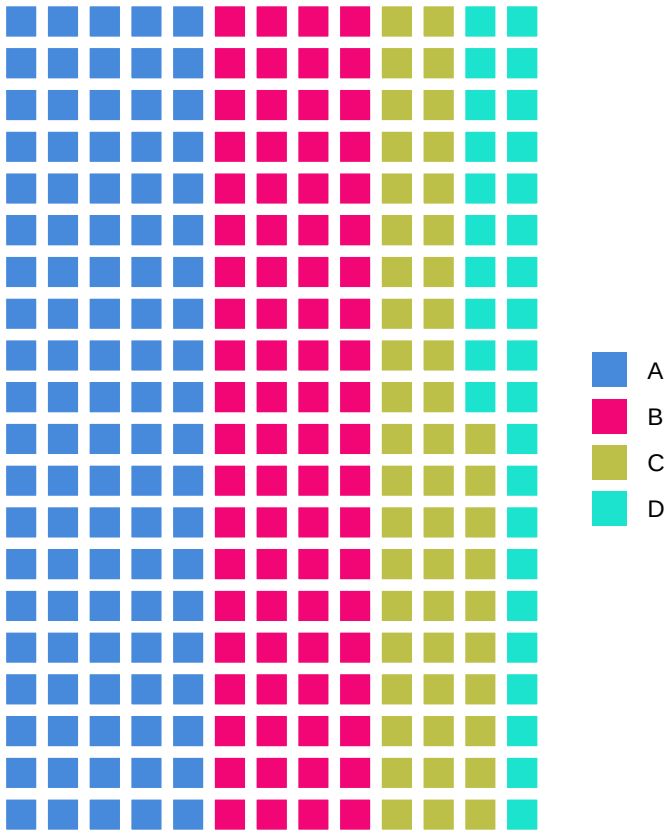
- #Create a random data set

```
d <- c(100, 80, 50, 30)
```

-

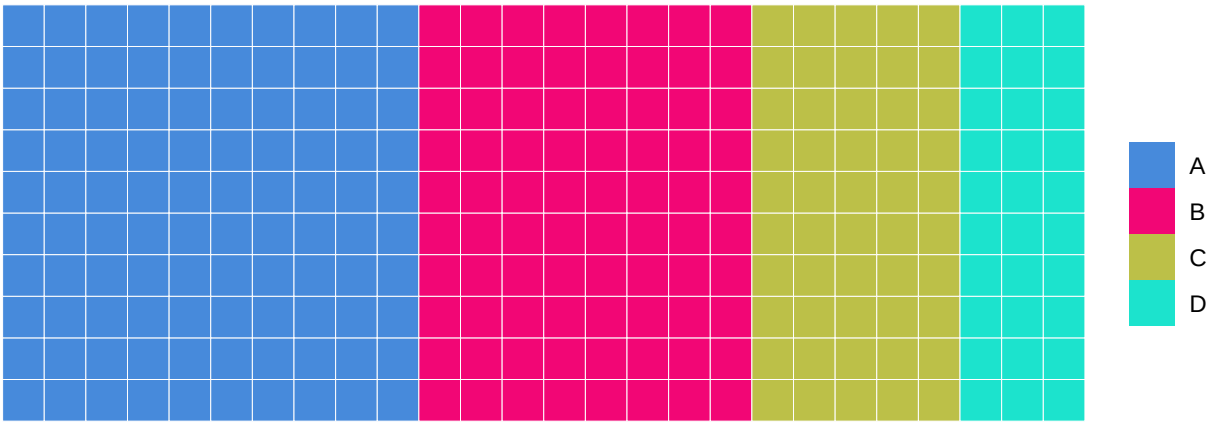
- #1. Basic waffle

```
waffle(d, rows = 20, colors = c("#478adb", "#f20675", "#bcc048", "#1ce3cd"))
```



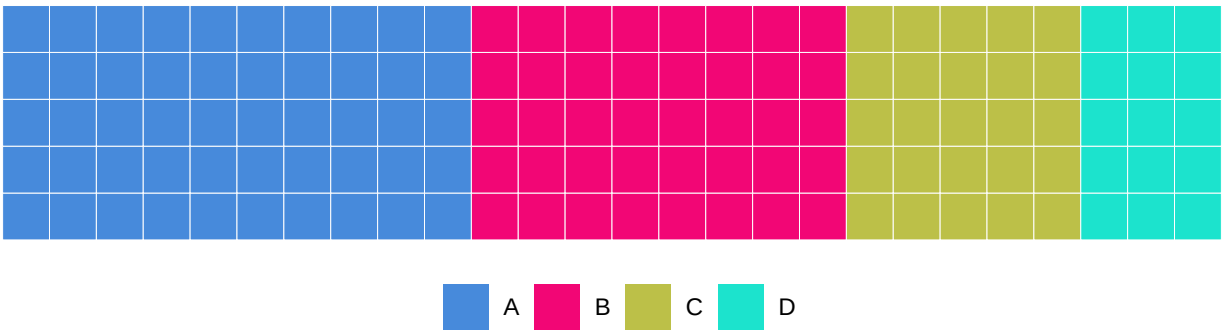
- #2. Change size

```
waffle(d, rows = 10, colors = c("#478adb", "#f20675", "#bcc048", "#1ce3cd"), size = 0.1)
```



- #4. Change the position of the legend

```
waffle(d/2 , rows = 5, colors = c("#478adb", "#f20675", "#bcc048", "#1ce3cd"), size = 0.1, legend_pos =
```

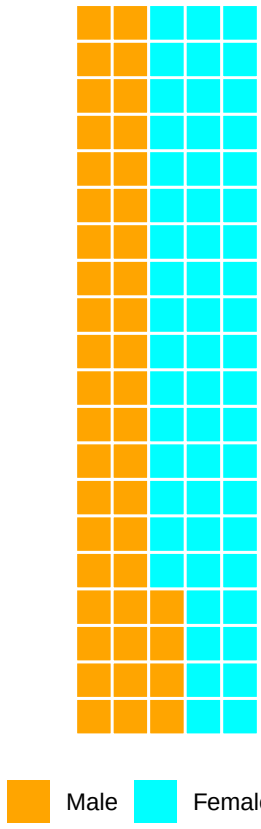


- #New simple dataset created

```
professional <- c(‘Male‘ = 44, ‘Female (56%)‘ = 56)
```

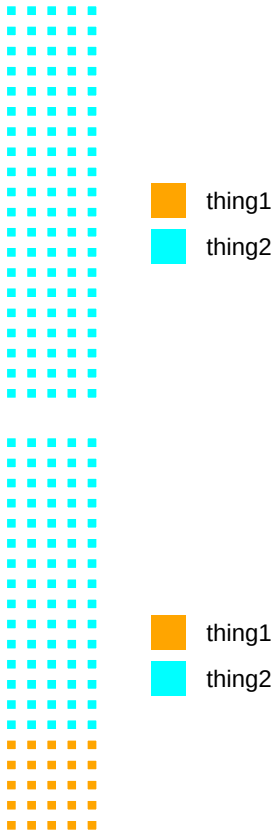
- #A simple waffle

```
waffle(
  professional, rows = 20, size = 0.5,
  colors = c(trend_color, "cyan"), legend_pos = "bottom"
)
```



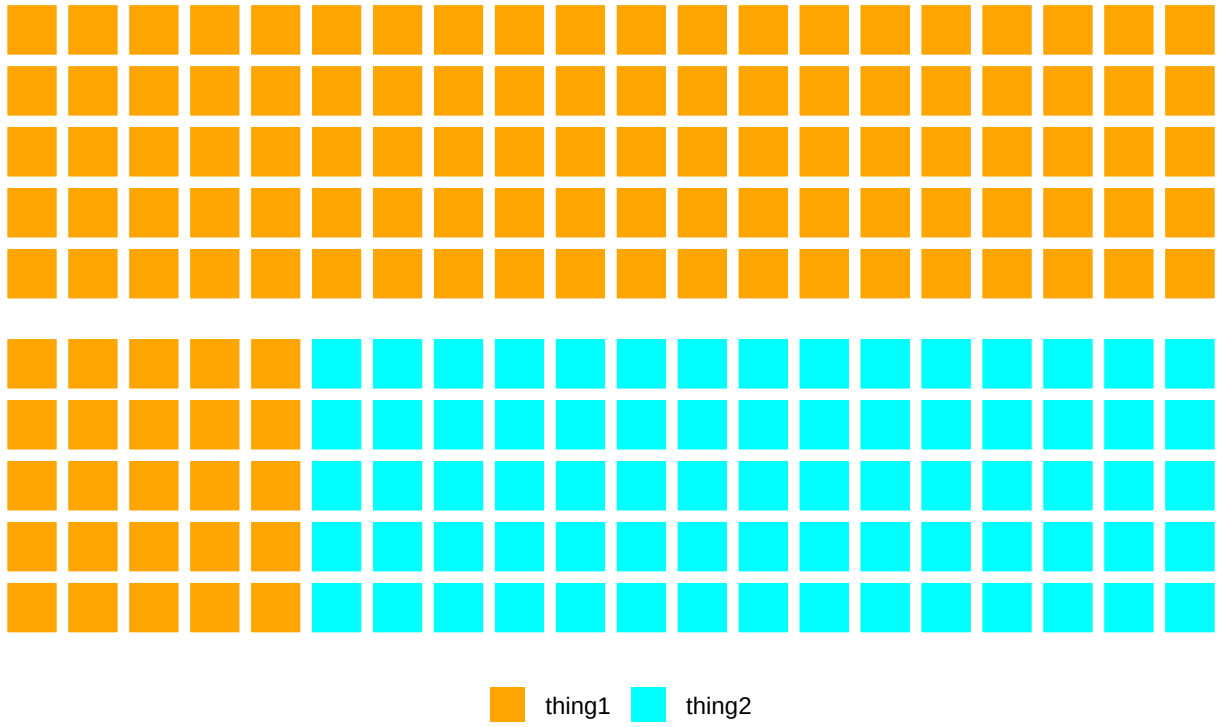
- #You can use the iron statment to create a small multiple of waffles

```
iron(
  waffle(c(thing1 = 0, thing2 = 100), colors = c(trend_color, "cyan"), rows = 5, flip=TRUE),
  waffle(c(thing1 = 25, thing2 = 75), colors = c(trend_color, "cyan"), rows = 5, flip=TRUE)
)
```



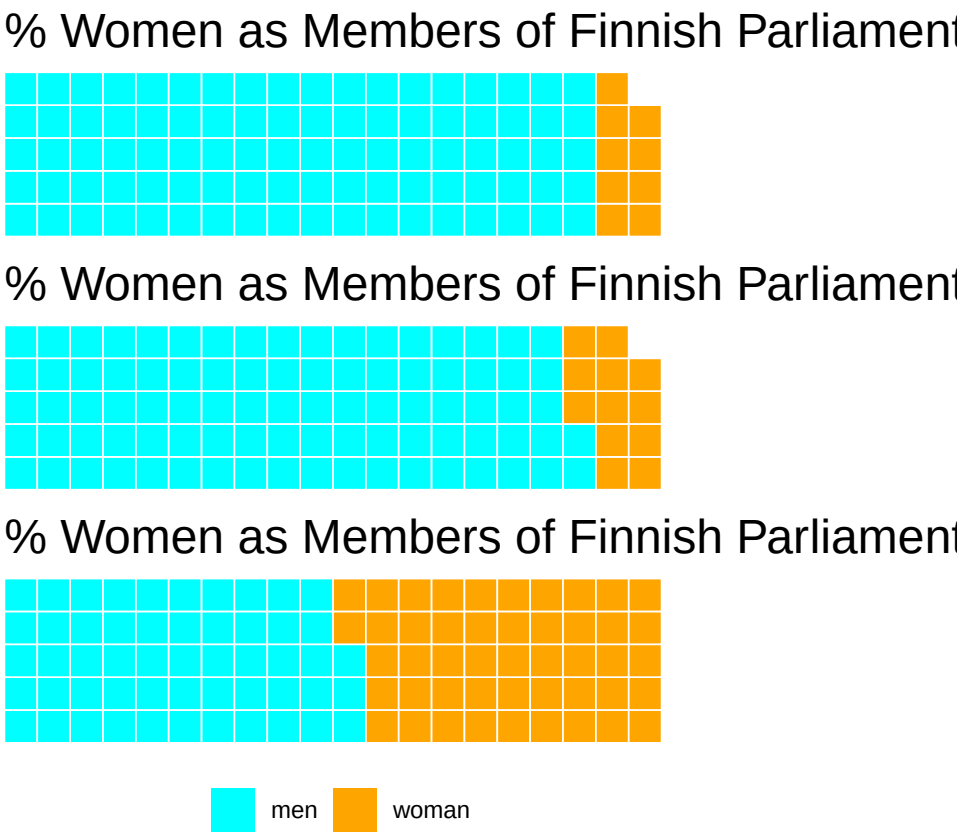
- #It's better to add the legend then separatly instead of showing it in every chart

```
iron(
  waffle(c(thing1 = 0, thing2 = 100), colors = c(trend_color, "cyan"), rows = 5, keep = FALSE, legend='
  waffle(c(thing1 = 25, thing2 = 75), colors = c(trend_color, "cyan"), rows = 5, keep = FALSE, legend_p
)
```



- #5. Adding the legend only to one

```
iron(

  waffle(
    c(men = 90.5, woman = 9.5), rows = 5, size = 0.3,
    colors = c("cyan", trend_color),
    keep = FALSE,
    title = "% Women as Members of Finnish Parliament 1907",
    legend='none'),

  waffle(
    c(men = 87.5, woman = 12.5), rows = 5, size = 0.3,
    colors = c("cyan", trend_color),
    keep = FALSE,
    title = "% Women as Members of Finnish Parliament 1916",
    legend='none'),

  waffle(
    c(men = 53, woman = 47), rows = 5, size = 0.3,
    colors = c("cyan", trend_color),
    keep = FALSE,
    title = "% Women as Members of Finnish Parliament 2019",
    legend_pos = "bottom")

)
```



- #PART 2:
- #Load data:

```
data <- read.csv("~/Downloads/Big Data analytics/Course 3 Visual Analytics/Assignment 1/DP_LIVE_1611202
```

- #Check the data

```
names(data)
```

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

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

```
##    LOCATION INDICATOR SUBJECT MEASURE FREQUENCY    TIME    Value Flag.Codes
## 1       AUS       HUR     TOT   PC_LF         M 2005-01 5.073780
## 2       AUS       HUR     TOT   PC_LF         M 2005-02 5.085003
## 3       AUS       HUR     TOT   PC_LF         M 2005-03 5.163290
## 4       AUS       HUR     TOT   PC_LF         M 2005-04 5.123025
## 5       AUS       HUR     TOT   PC_LF         M 2005-05 5.100072
## 6       AUS       HUR     TOT   PC_LF         M 2005-06 4.950172
## 7       AUS       HUR     TOT   PC_LF         M 2005-07 4.971967
## 8       AUS       HUR     TOT   PC_LF         M 2005-08 4.900029
## 9       AUS       HUR     TOT   PC_LF         M 2005-09 5.001081
## 10      AUS       HUR     TOT   PC_LF         M 2005-10 5.015691
```

```
str(data)
```

```
## 'data.frame':    7437 obs. of  8 variables:
##  $ LOCATION  : chr  "AUS" "AUS" "AUS" "AUS" ...
##  $ INDICATOR : chr  "HUR" "HUR" "HUR" "HUR" ...
##  $ SUBJECT   : chr  "TOT" "TOT" "TOT" "TOT" ...
##  $ MEASURE   : chr  "PC_LF" "PC_LF" "PC_LF" "PC_LF" ...
##  $ FREQUENCY : chr  "M" "M" "M" "M" ...
##  $ TIME      : chr  "2005-01" "2005-02" "2005-03" "2005-04" ...
##  $ Value     : num  5.07 5.09 5.16 5.12 5.1 ...
##  $ Flag.Codes: chr  "" "" "" "" ...
```

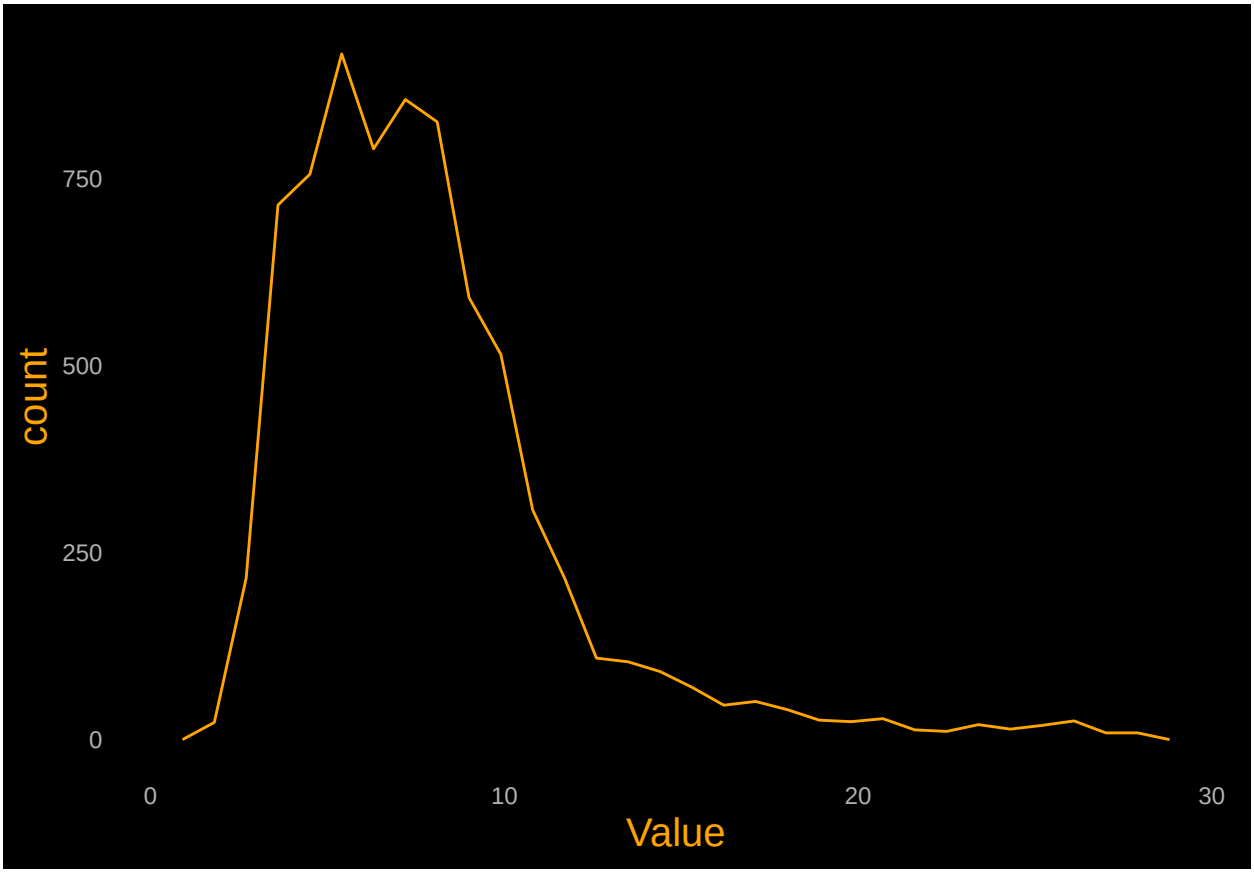
```
summary(data)
```

```
##    LOCATION          INDICATOR           SUBJECT            MEASURE
##  Length:7437        Length:7437        Length:7437        Length:7437
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##   FREQUENCY             TIME               Value         Flag.Codes
##  Length:7437        Length:7437        Min.   : 1.800   Length:7437
##  Class :character   Class :character   1st Qu.: 5.100   Class :character
##  Mode  :character   Mode  :character   Median : 7.100   Mode  :character
##                                        Mean   : 7.748
##                                        3rd Qu.: 9.300
##                                        Max.   :27.900
```

- #DISTRIBUTION
- #Chart of the world unemployment rate over years:

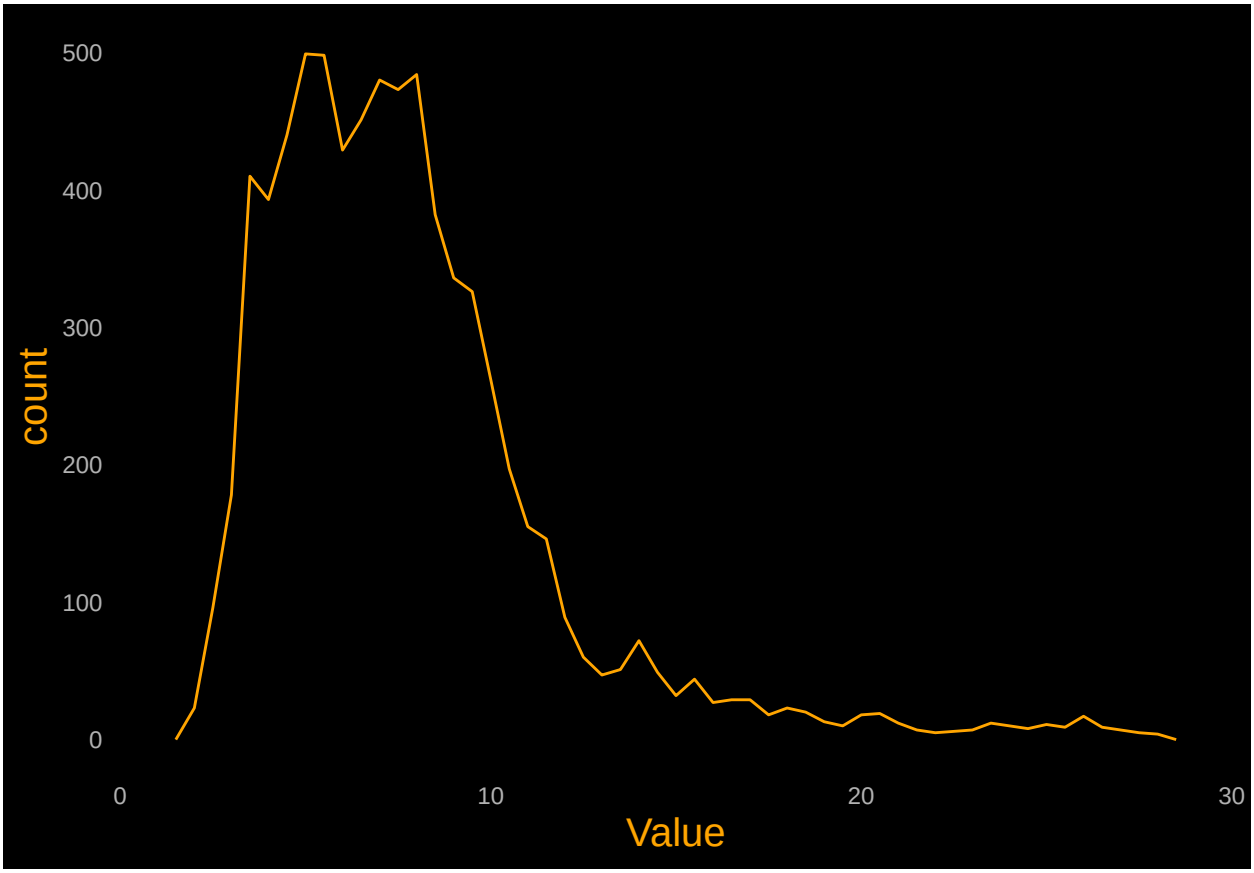
```
ggplot(data, aes(Value)) +
  geom_freqpoly(colour = trend_color)
```

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



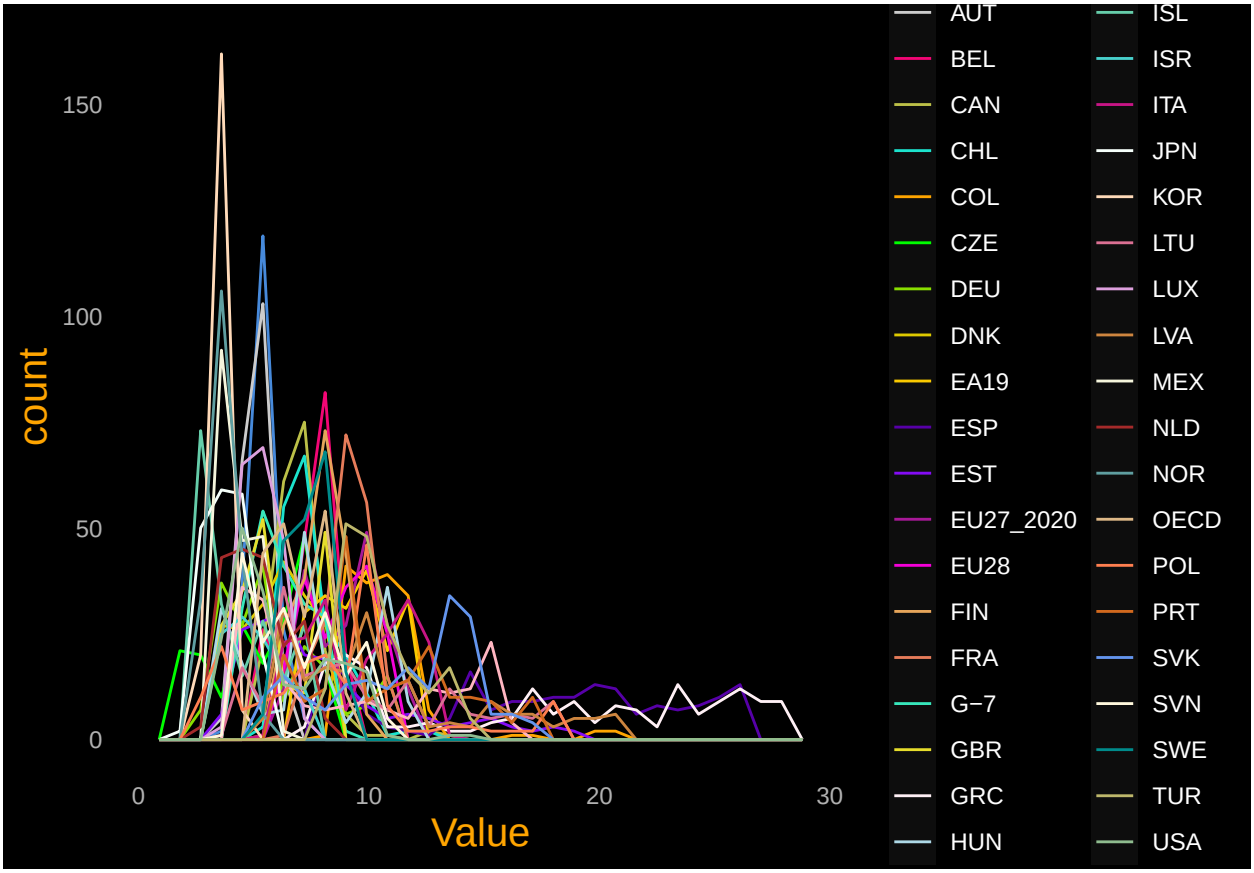
- #Changing the bin width (more details)

```
ggplot(data, aes(Value)) +
  geom_freqpoly(colour = trend_color, binwidth = 0.5)
```



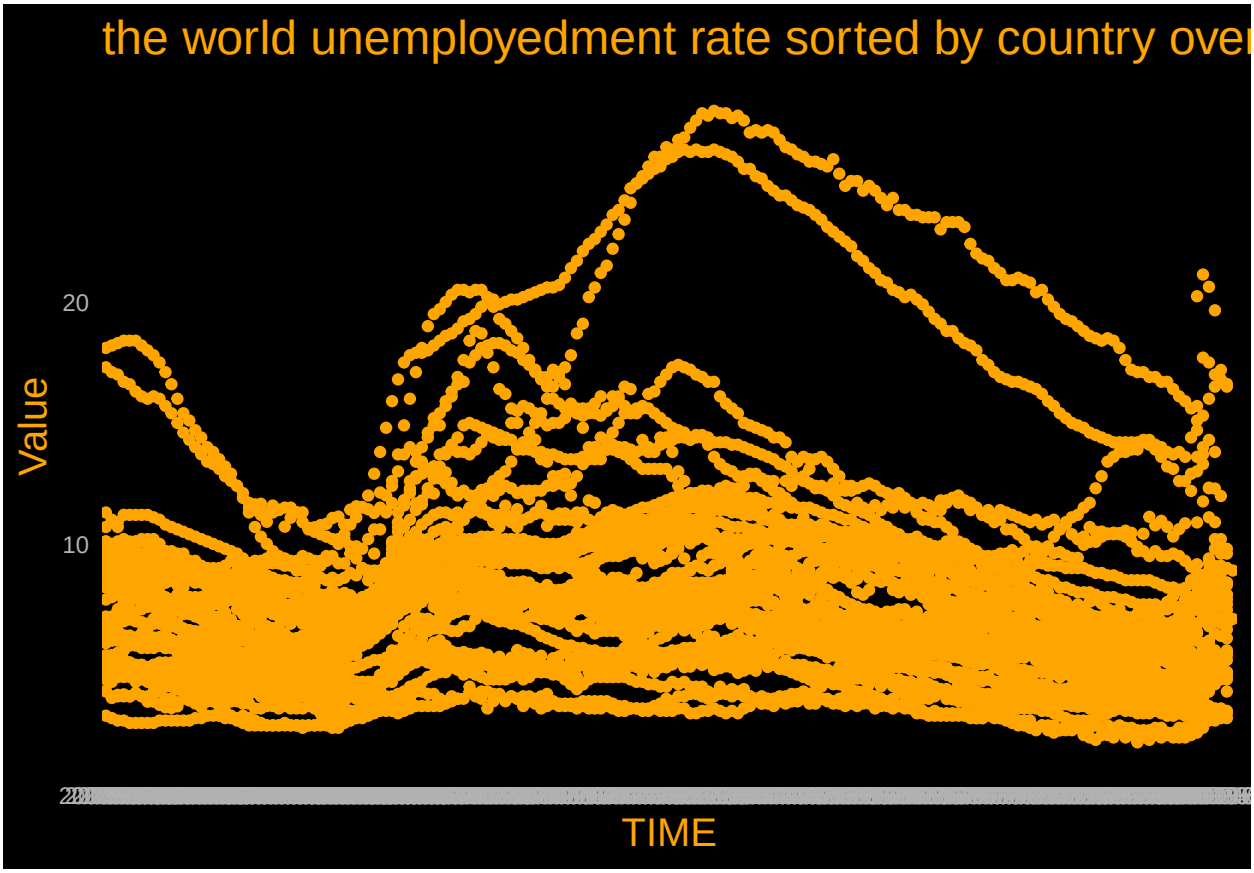
-#Adding color as a visual encoding

```
ggplot(data, aes(Value, colour = LOCATION)) +
  geom_freqpoly(bins=30) +
  scale_color_manual(values=c("#478adb", "#cccccc", "#f20675", "#bcc048", "#1ce3cd","orange","green","#
```



- **Simple chart of the world unemployedment rate sorted by country over year 2005-2020**

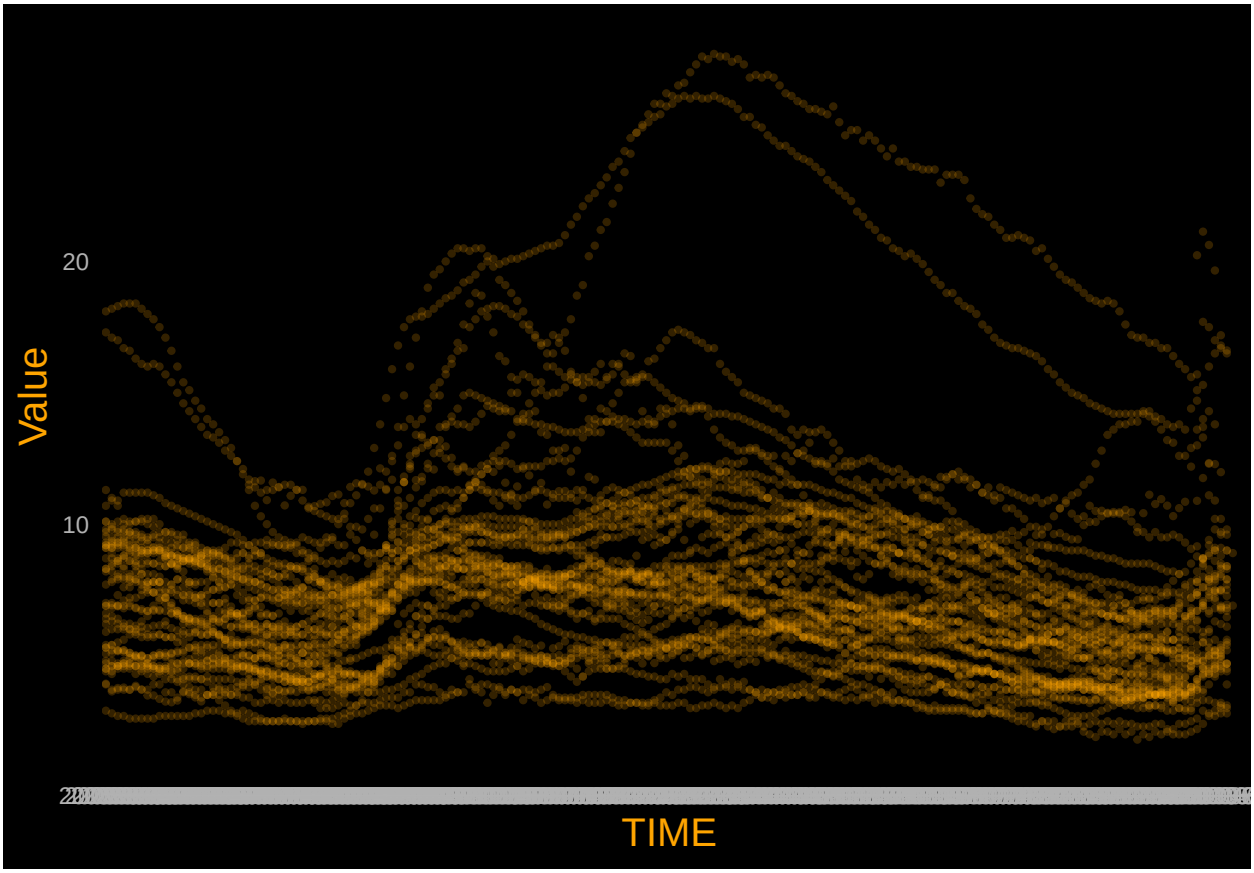
```
ggplot(data, aes(x=TIME, y=Value)) +
  geom_point(color=trend_color)+
  labs(title = "the world unemployedment rate sorted by country over year 2005-2020")
```



- #Adding a trend line

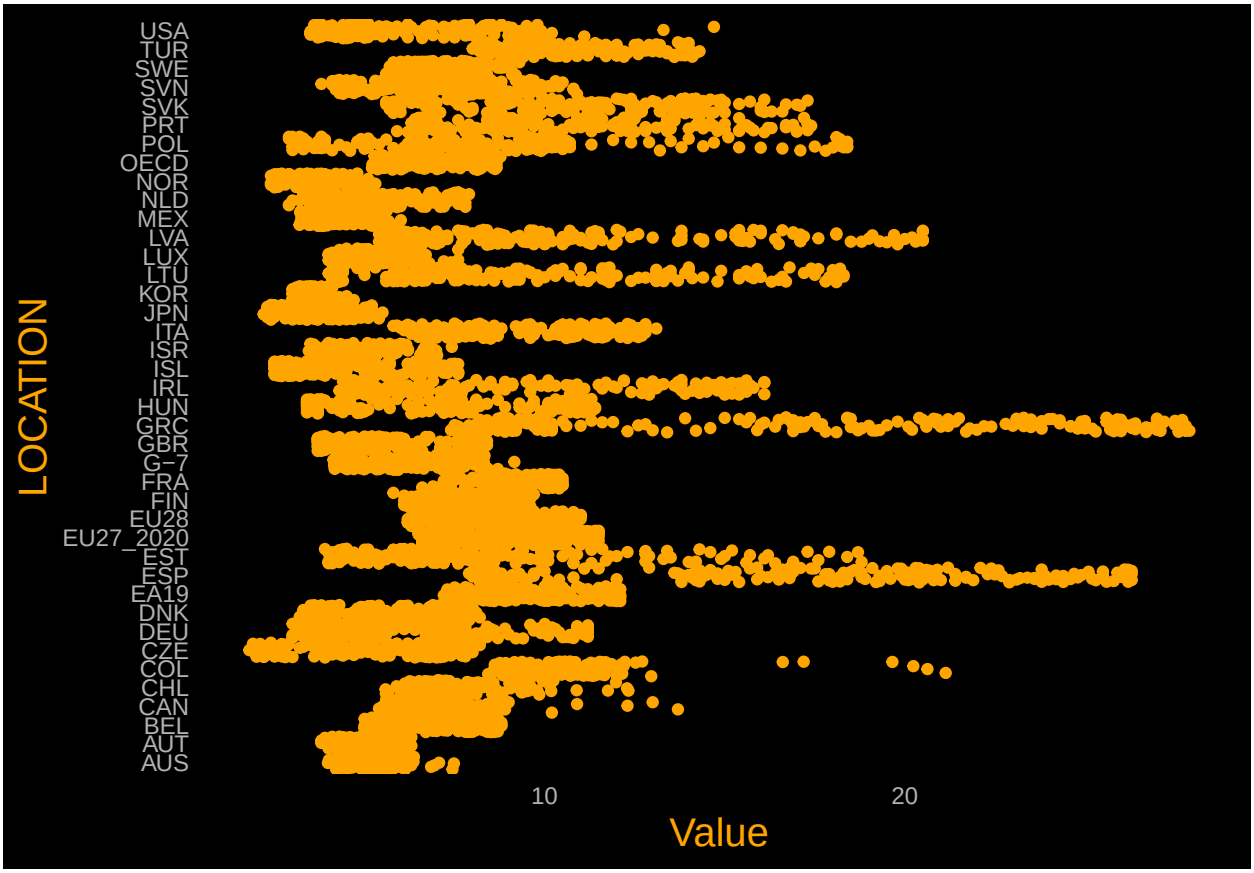
```
ggplot(data, aes(x=TIME, y=Value)) +
  geom_point(color=trend_color, size=0.8, alpha=0.2)+
  stat_smooth(color="white")
```

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

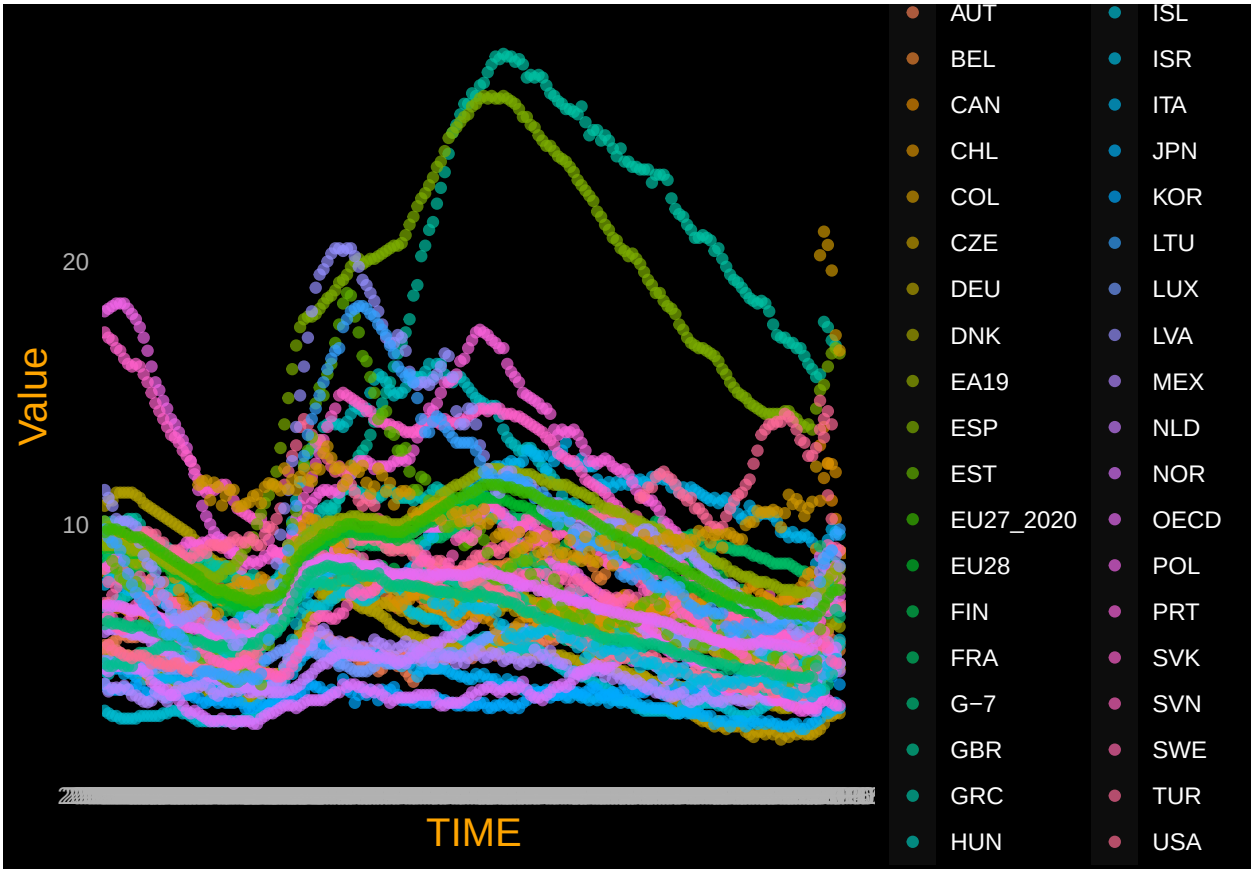


- #Simple jitter plot

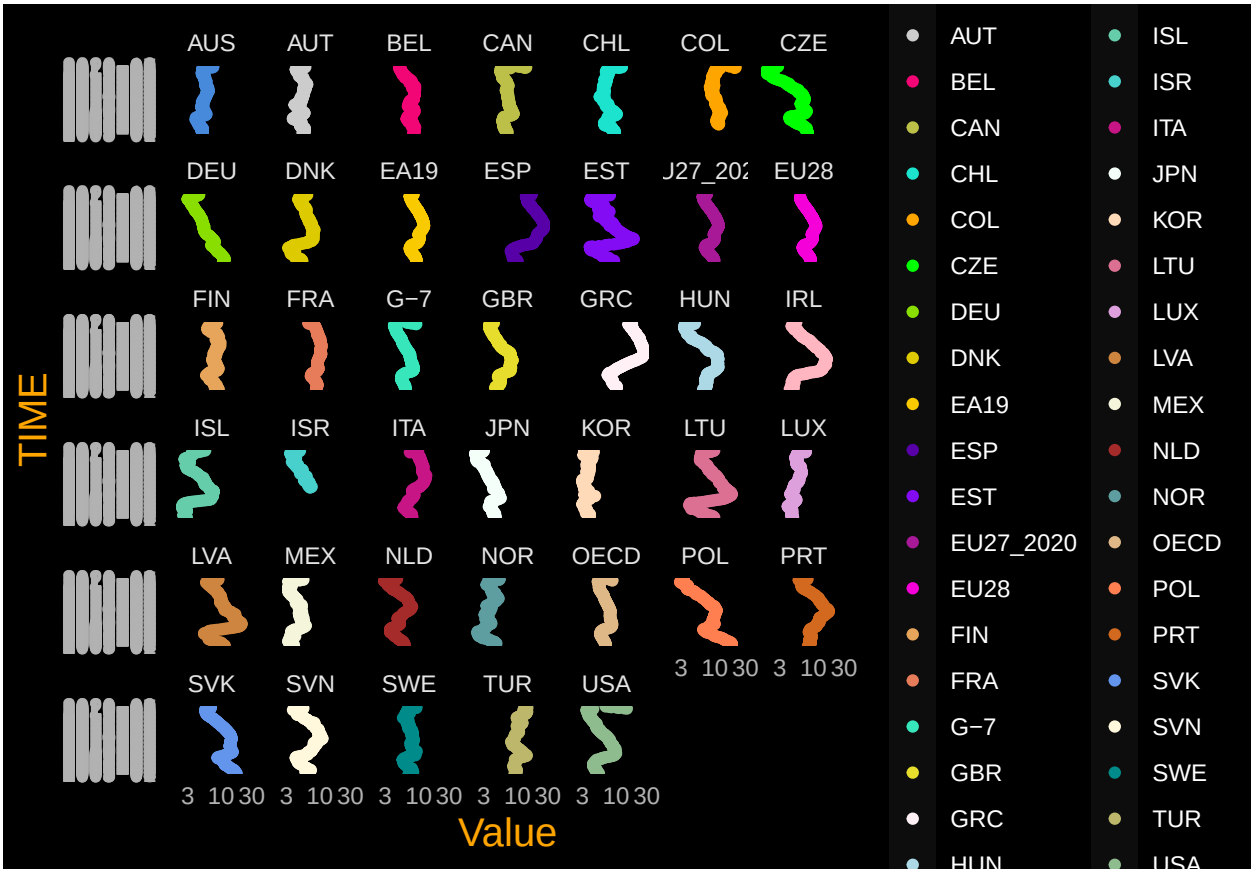
```
ggplot(data, aes(y= LOCATION, x=Value)) +
geom_jitter(color=trend_color)
```




```
ggplot(data, aes(x= TIME, y=Value, colour=LOCATION)) +
geom_quasirandom(alpha=0.7, groupOnX=FALSE, method = "smiley")
```

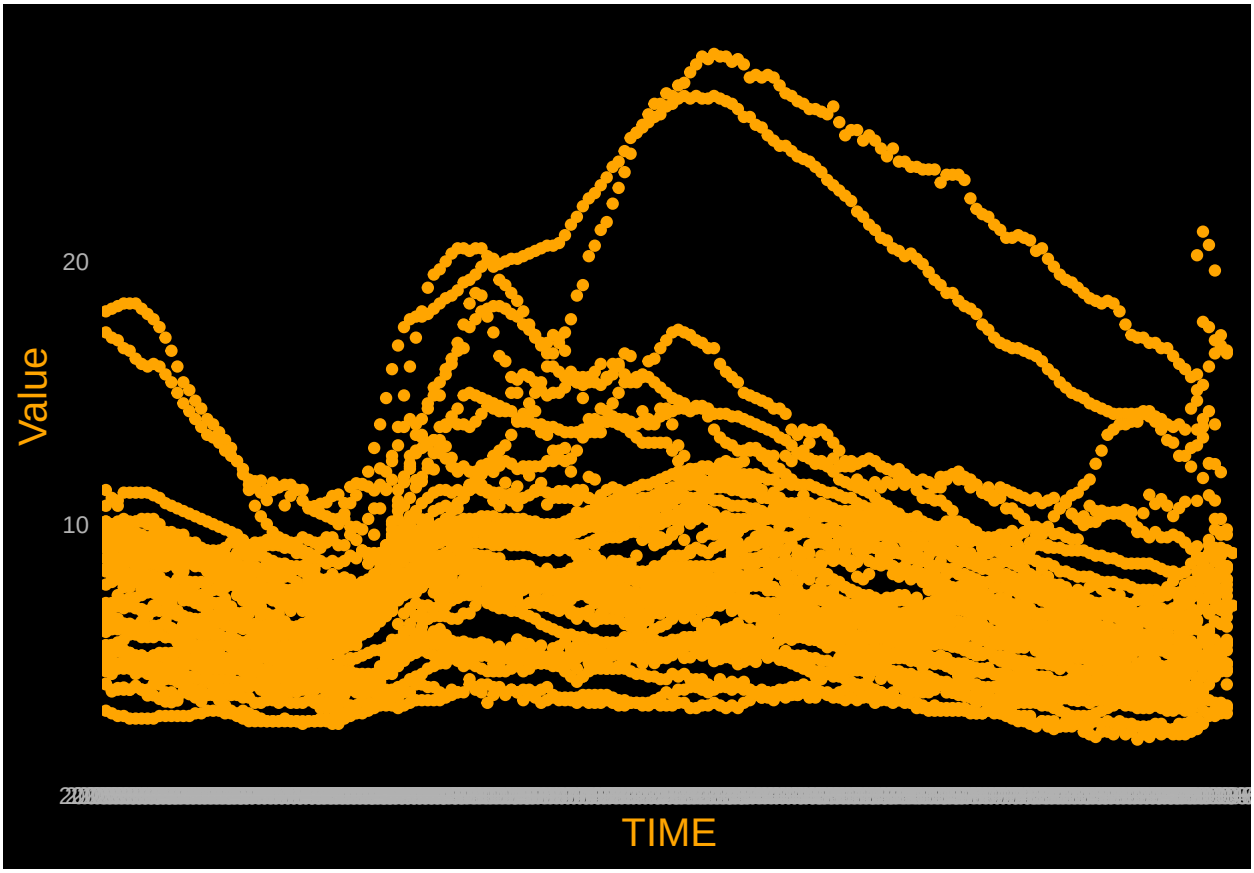



```
ggplot(data, aes(y=TIME, x=Value, colour=LOCATION)) +
geom_point() +
scale_x_log10() +
scale_size(range = c(2, 12)) +
facet_wrap(~LOCATION) +
scale_colour_manual(values=c("#478adb", "#cccccc", "#f20675", "#bcc048", "#1ce3cd","orange","green","#8
```



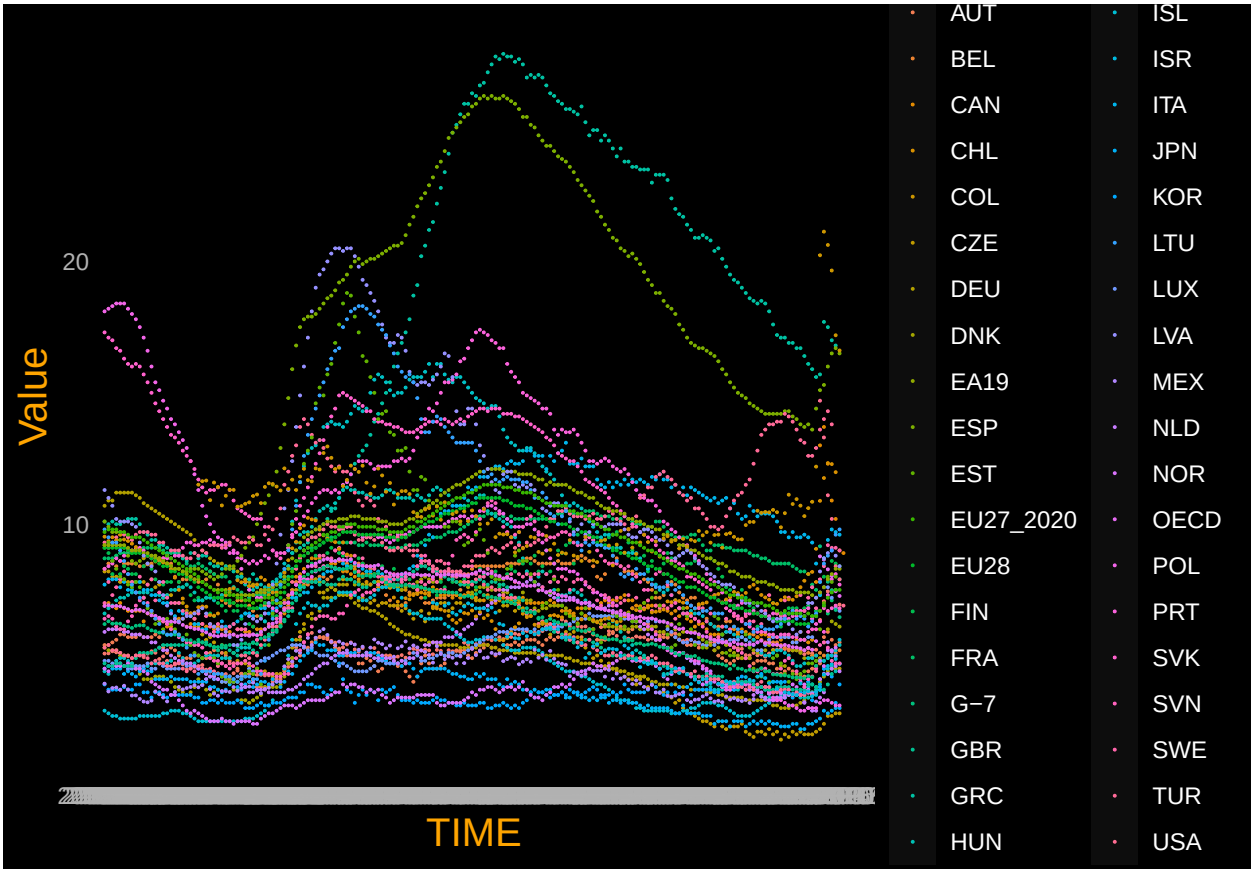
- #RELATIONSHIP Analysis
- #Basic scatter plot

```
ggplot(data, aes(x= TIME, y=Value)) +
  geom_point(color=trend_color)
```



- #Basic scatter plot - color as visual encoding redundant

```
ggplot(data, aes(x= TIME, y=Value, color=LOCATION)) +
  geom_point(size=0.02)
```

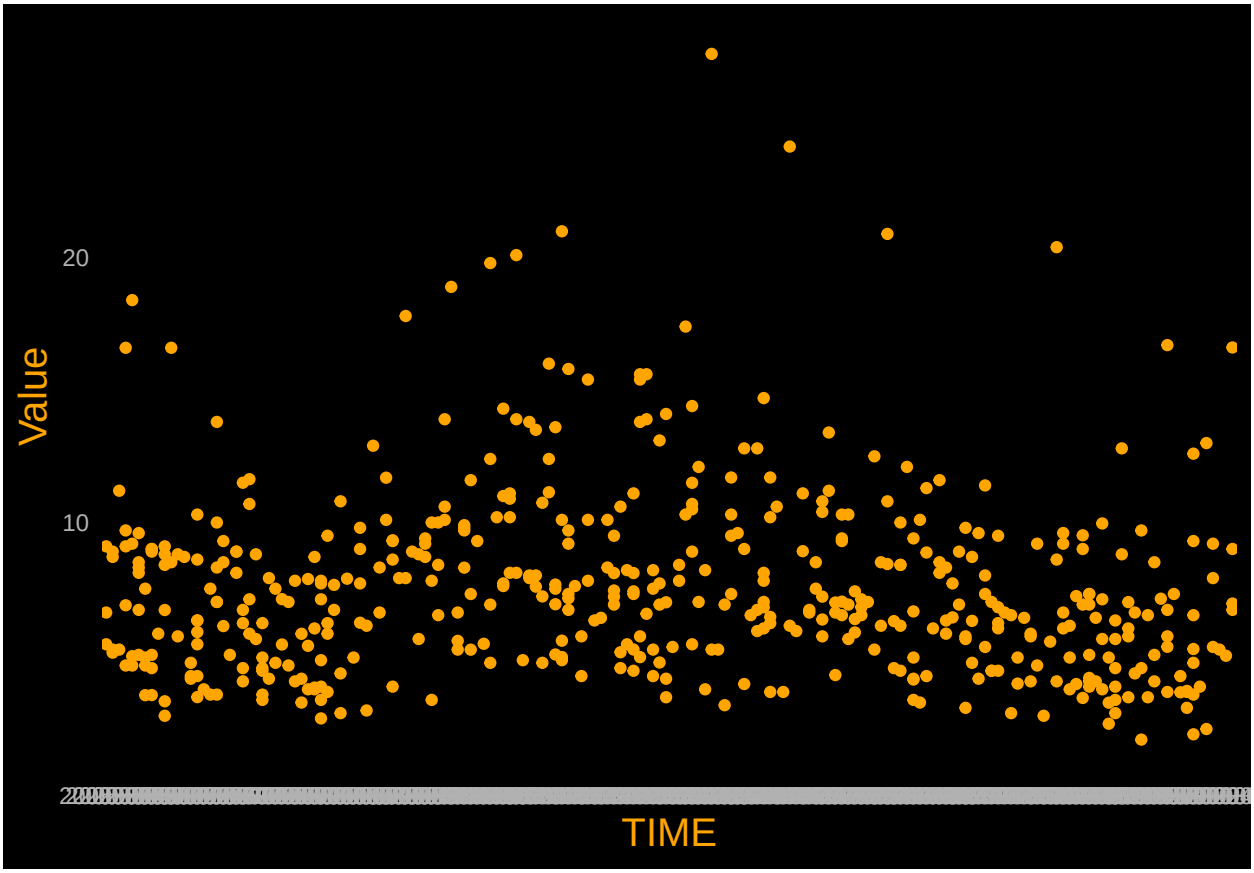


- #Another way to handle big data sets is to create a sample

```
data_sample <- data[sample(nrow(data), 500),]
```

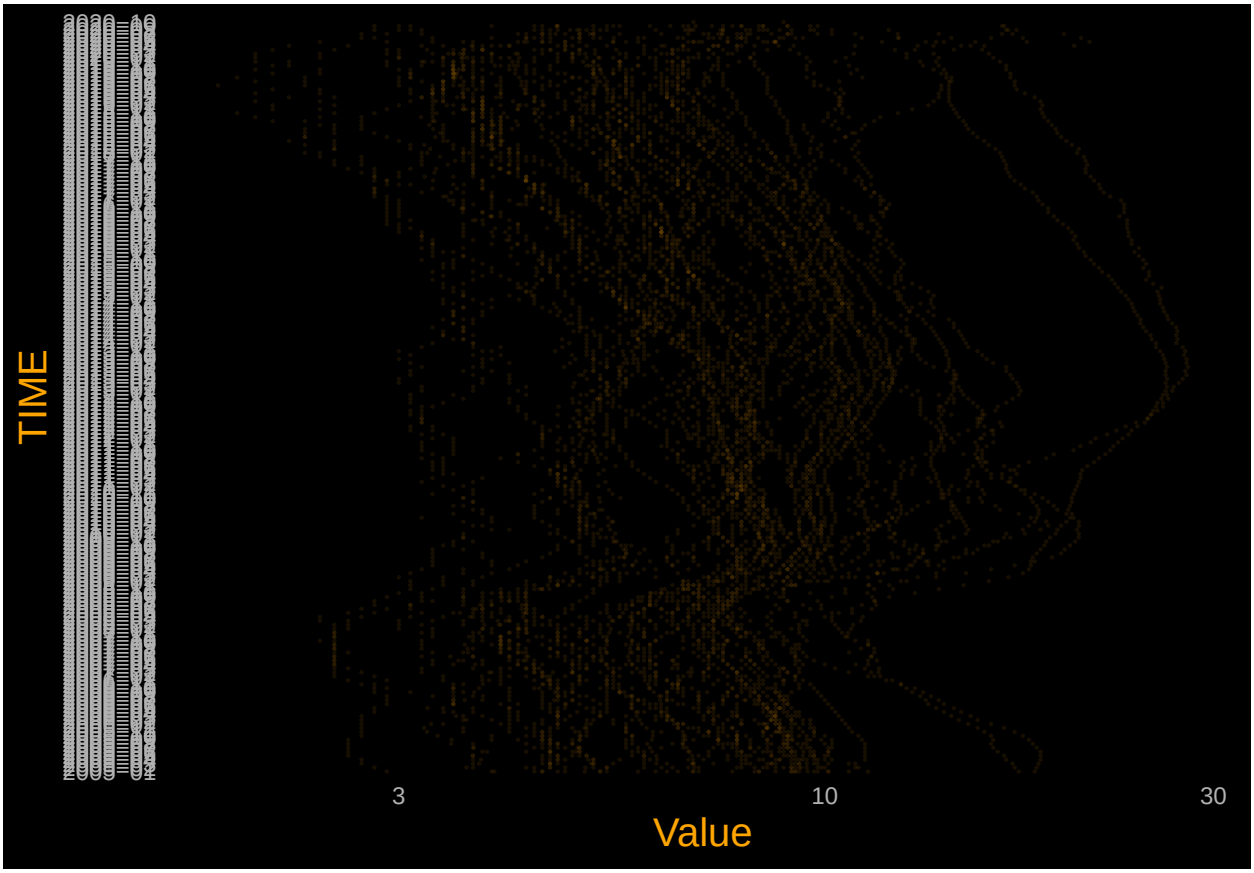
- #Basic scatter plot

```
ggplot(data_sample, aes(x=TIME, y=Value)) +
  geom_point(color=trend_color)
```



- #Change the position scale to logarithmic scaling

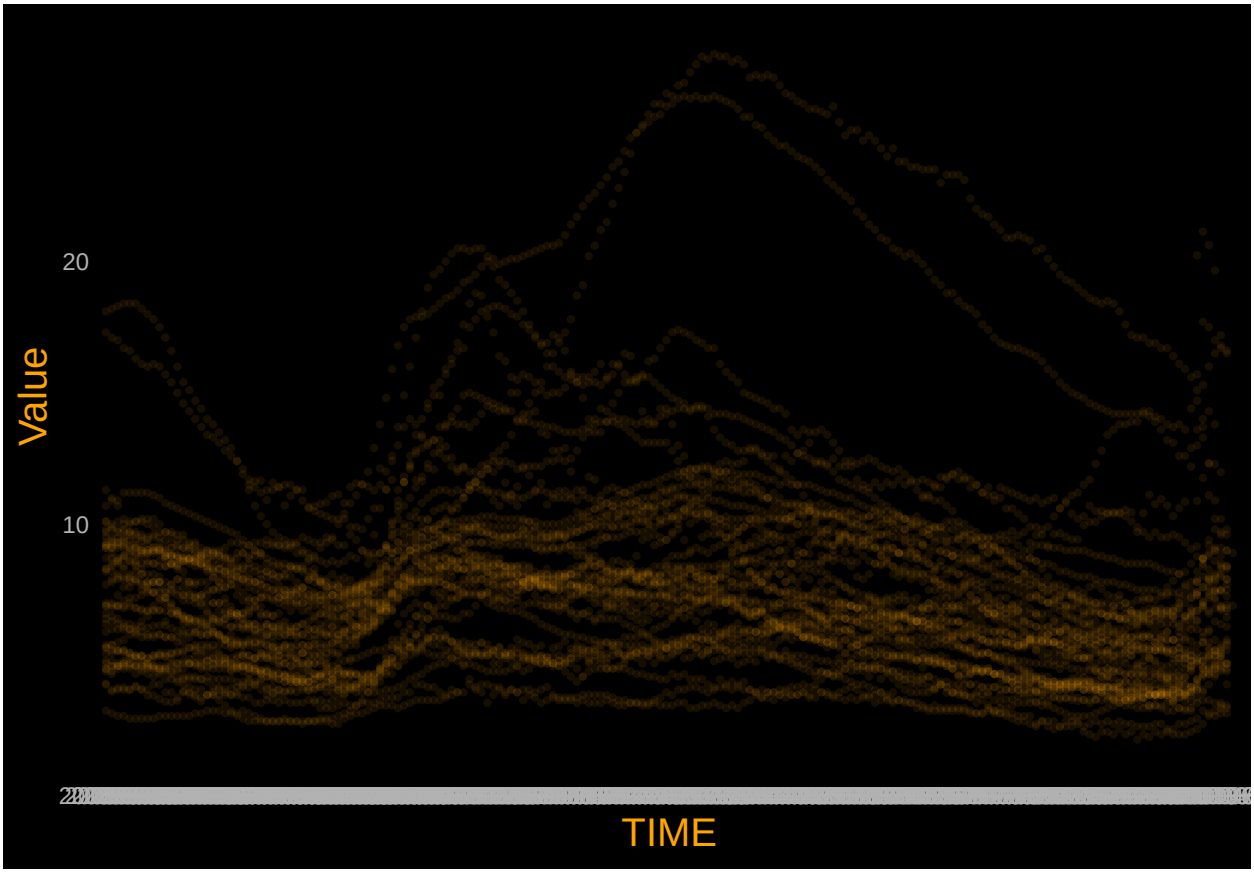
```
ggplot(data, aes(y=TIME, x=Value)) +
  geom_point(size=0.1, alpha=0.09, color=trend_color) +
  scale_x_log10()
```



- #Adding a trend line

```
ggplot(data, aes(x=TIME, y=Value)) +
  geom_point(color=trend_color, size=0.8, alpha=0.09)+
  stat_smooth(color="white")
```

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



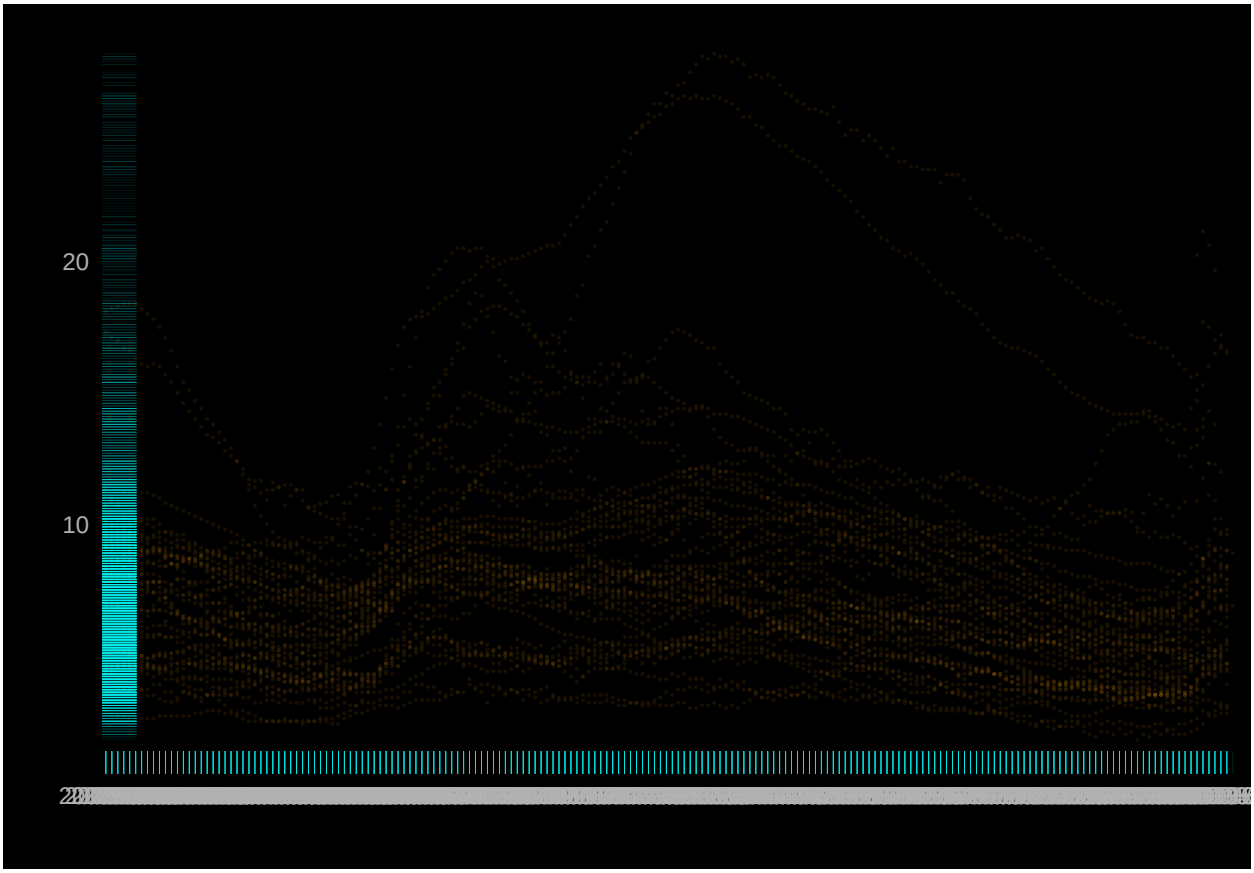
- #Small multiples- two variables

```
ggplot(data, aes(x=TIME, y=Value)) +
  geom_point(color=trend_color, size=0.8, alpha=0.09)+
  facet_wrap(Value ~ LOCATION) +
  stat_smooth(color="white")
```

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

-#Dot-dash scatter plot

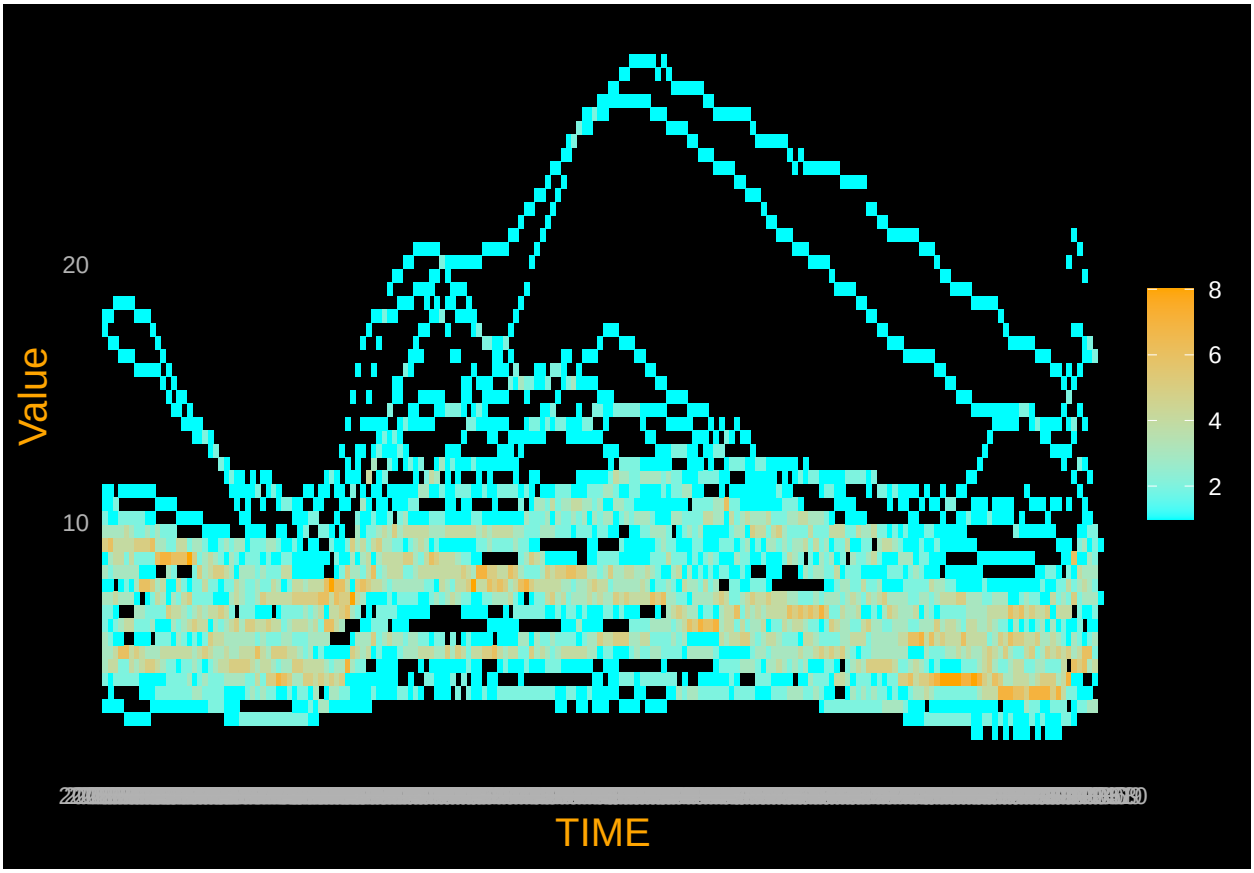
```
ggplot(data, aes(TIME, Value)) +
  geom_point(size=0.05, alpha=0.09, color=trend_color) +
  geom_rug(size=0.04, alpha=0.08, color="cyan") +
  xlab("") +
  ylab("")
```



-#Marginal plot to compare all simple distributions with the scatter plot relationship representation

- #Heatmap based on rectangles

```
ggplot(data, aes(x=TIME, y=Value)) +
  geom_bin2d(bins = 50) +
  scale_fill_gradient(low="cyan", high=trend_color)
```



- #Time series analysis:
- #Check the data

```
names(data)
```

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

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

```
##    LOCATION INDICATOR SUBJECT MEASURE FREQUENCY    TIME    Value Flag.Codes
## 1       AUS       HUR     TOT   PC_LF         M 2005-01 5.073780
## 2       AUS       HUR     TOT   PC_LF         M 2005-02 5.085003
## 3       AUS       HUR     TOT   PC_LF         M 2005-03 5.163290
## 4       AUS       HUR     TOT   PC_LF         M 2005-04 5.123025
## 5       AUS       HUR     TOT   PC_LF         M 2005-05 5.100072
## 6       AUS       HUR     TOT   PC_LF         M 2005-06 4.950172
## 7       AUS       HUR     TOT   PC_LF         M 2005-07 4.971967
## 8       AUS       HUR     TOT   PC_LF         M 2005-08 4.900029
## 9       AUS       HUR     TOT   PC_LF         M 2005-09 5.001081
## 10      AUS       HUR     TOT   PC_LF         M 2005-10 5.015691
```

```
str(data)
```

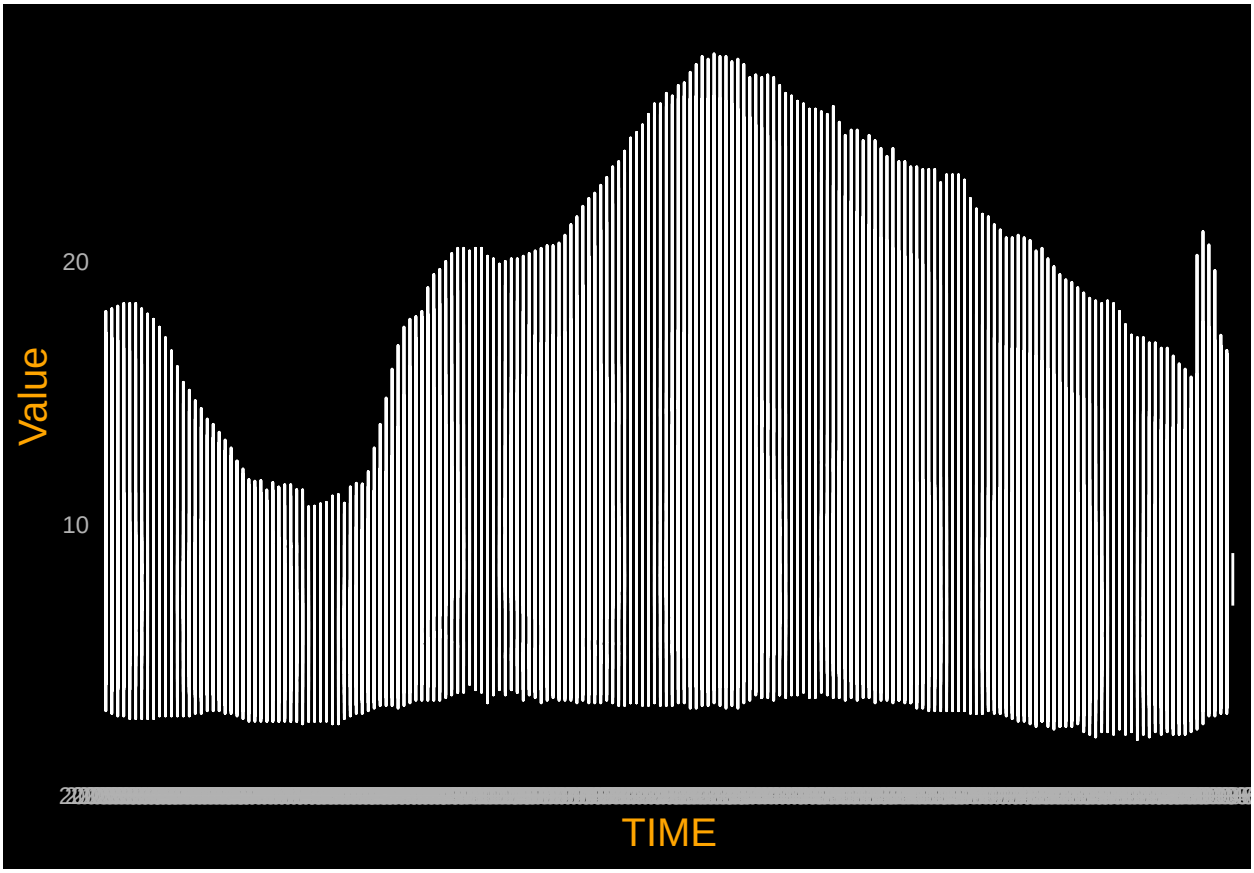
```
## 'data.frame':    7437 obs. of  8 variables:
##  $ LOCATION  : chr  "AUS" "AUS" "AUS" "AUS" ...
##  $ INDICATOR : chr  "HUR" "HUR" "HUR" "HUR" ...
##  $ SUBJECT   : chr  "TOT" "TOT" "TOT" "TOT" ...
##  $ MEASURE   : chr  "PC_LF" "PC_LF" "PC_LF" "PC_LF" ...
##  $ FREQUENCY : chr  "M" "M" "M" "M" ...
##  $ TIME      : chr  "2005-01" "2005-02" "2005-03" "2005-04" ...
##  $ Value     : num  5.07 5.09 5.16 5.12 5.1 ...
##  $ Flag.Codes: chr  "" "" "" "" ...
```

```
summary(data)
```

```
##    LOCATION          INDICATOR           SUBJECT            MEASURE
##  Length:7437        Length:7437        Length:7437        Length:7437
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##   FREQUENCY             TIME               Value         Flag.Codes
##  Length:7437        Length:7437        Min.   : 1.800   Length:7437
##  Class :character   Class :character   1st Qu.: 5.100   Class :character
##  Mode  :character   Mode  :character   Median : 7.100   Mode  :character
##                                        Mean   : 7.748
##                                        3rd Qu.: 9.300
##                                        Max.   :27.900
```

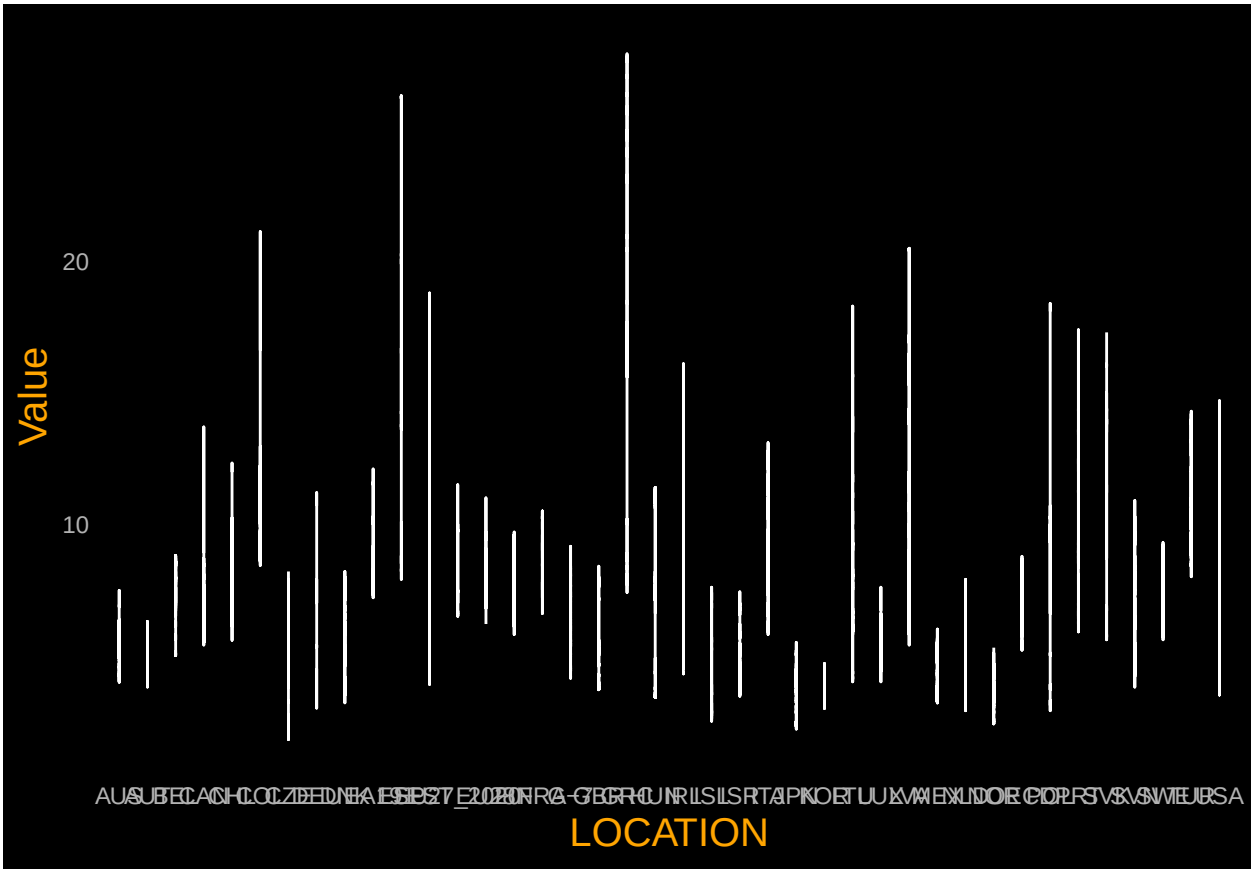
- #The normal line chart

```
ggplot(data, aes(TIME, Value)) + geom_line()
```



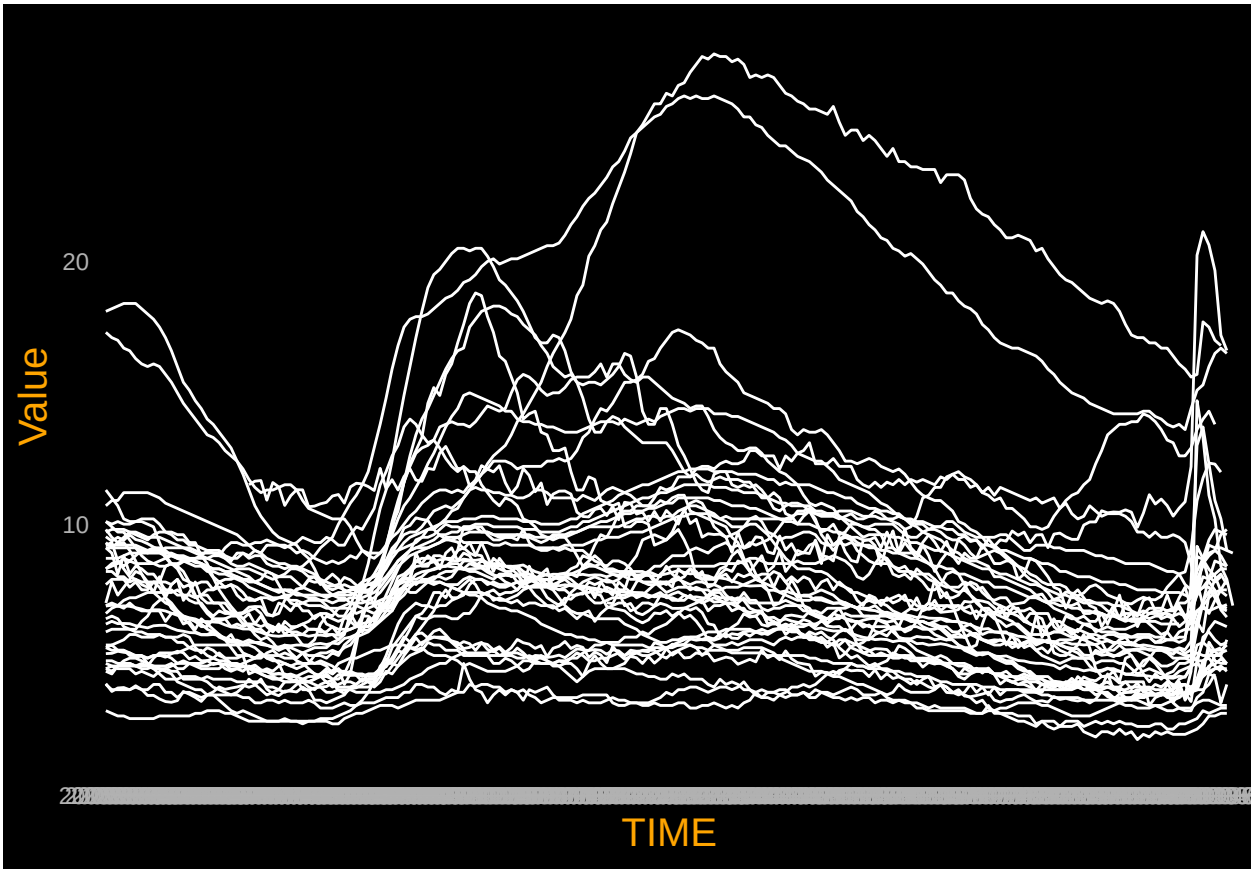
- **A single line tries to connect all the observations**

```
h<- ggplot(data, aes(LOCATION, Value))
h + geom_line()
```



- #Grouping the observation by the location

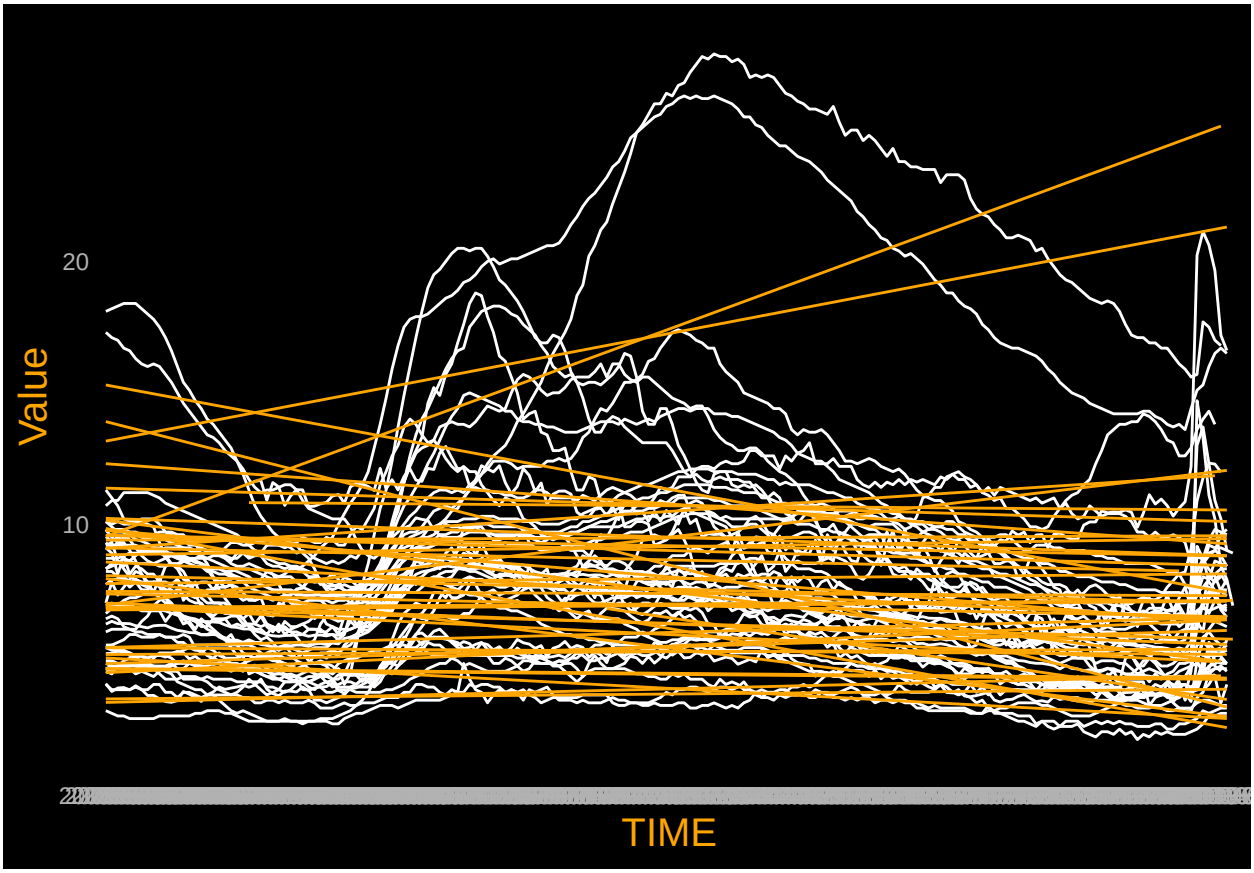
```
h1 <- ggplot(data, aes(TIME, Value, group=LOCATION))
h1 + geom_line()
```



- **groups the data the same way for both layers**

```
h1 + geom_line() +
     geom_smooth(aes(), colour = trend_color, size = 0.5, method = "lm", se = FALSE)
```

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

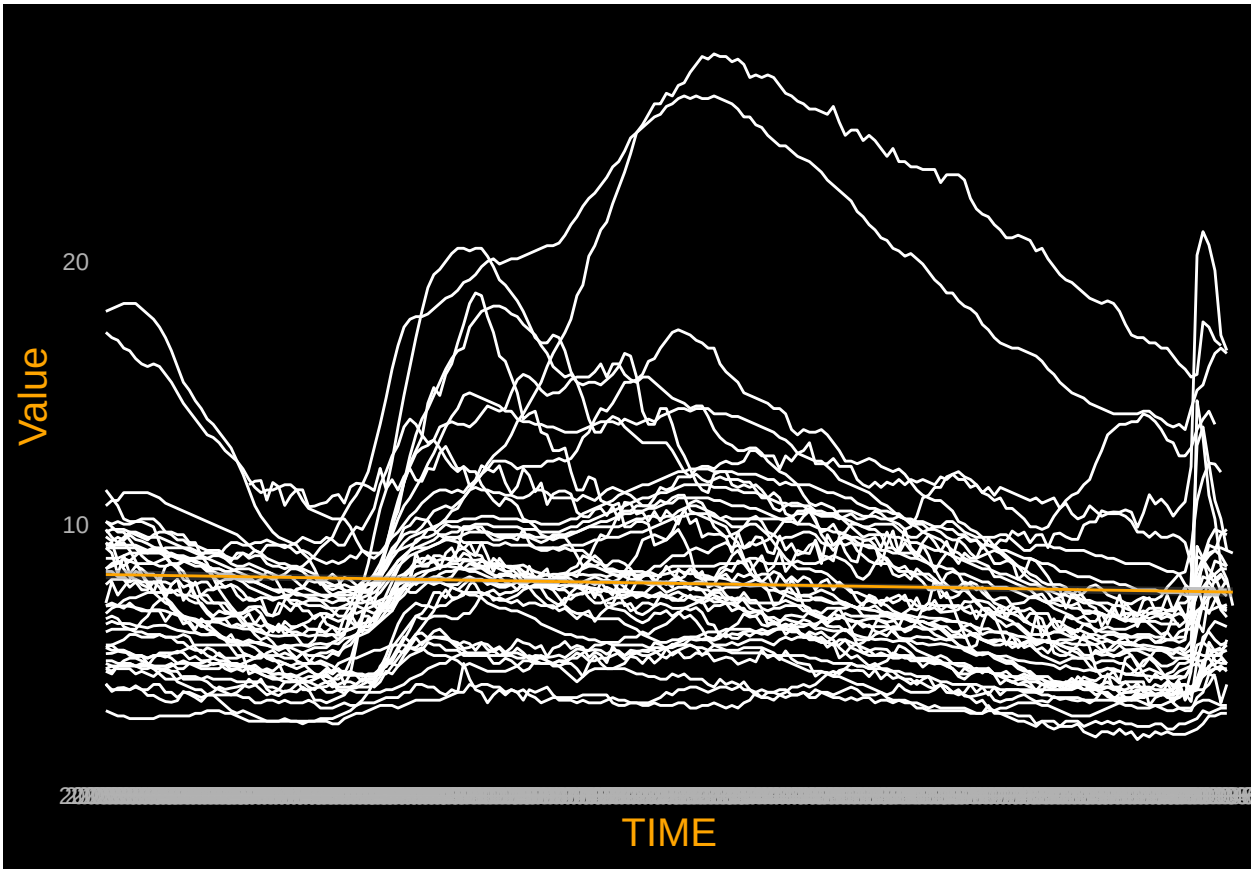


-

Adding a confidence intervall

```
h1 + geom_line() +
  geom_smooth(aes(group = 1), colour = trend_color, size = 0.5, method = "lm", se = TRUE)
```

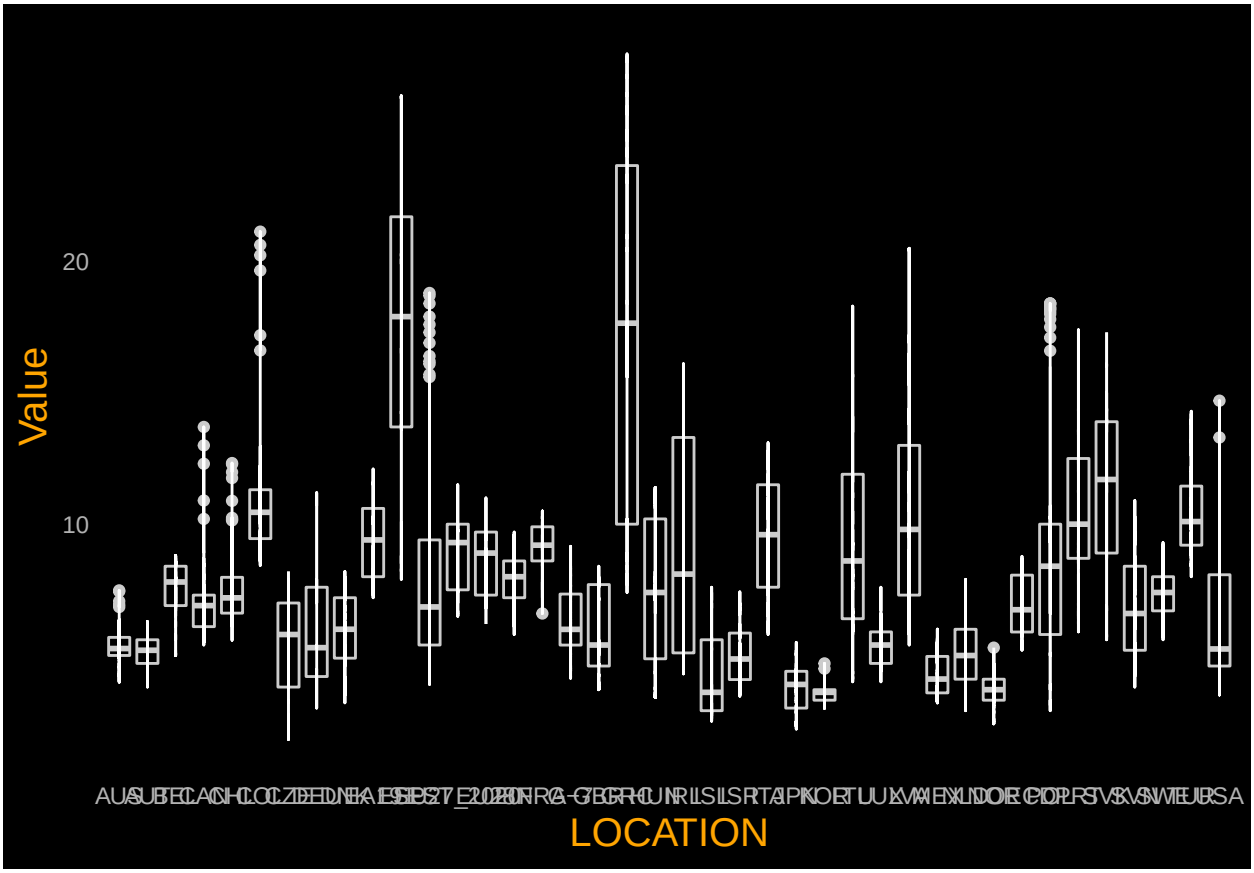
```
## `geom_smooth()` using formula 'y ~ x'
```



-

Now we combine a box-plot with the line chart

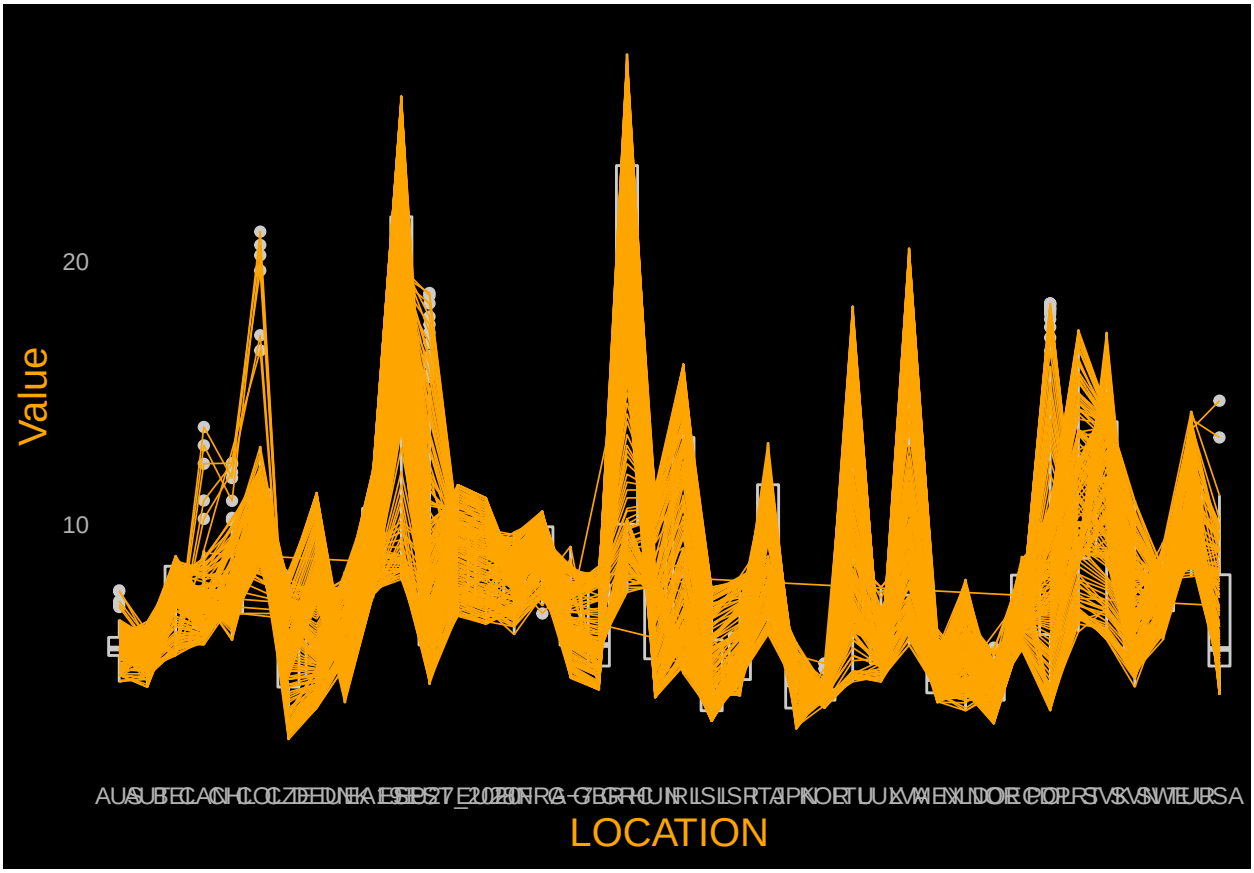
```
h2 <- ggplot(data, aes(LOCATION, Value))
h2 + geom_boxplot() + geom_line()
```



-

  We can add the a line chart again for time

```
h2 + geom_boxplot() + geom_line(aes(group = TIME), size=0.3, colour=trend_color)
```

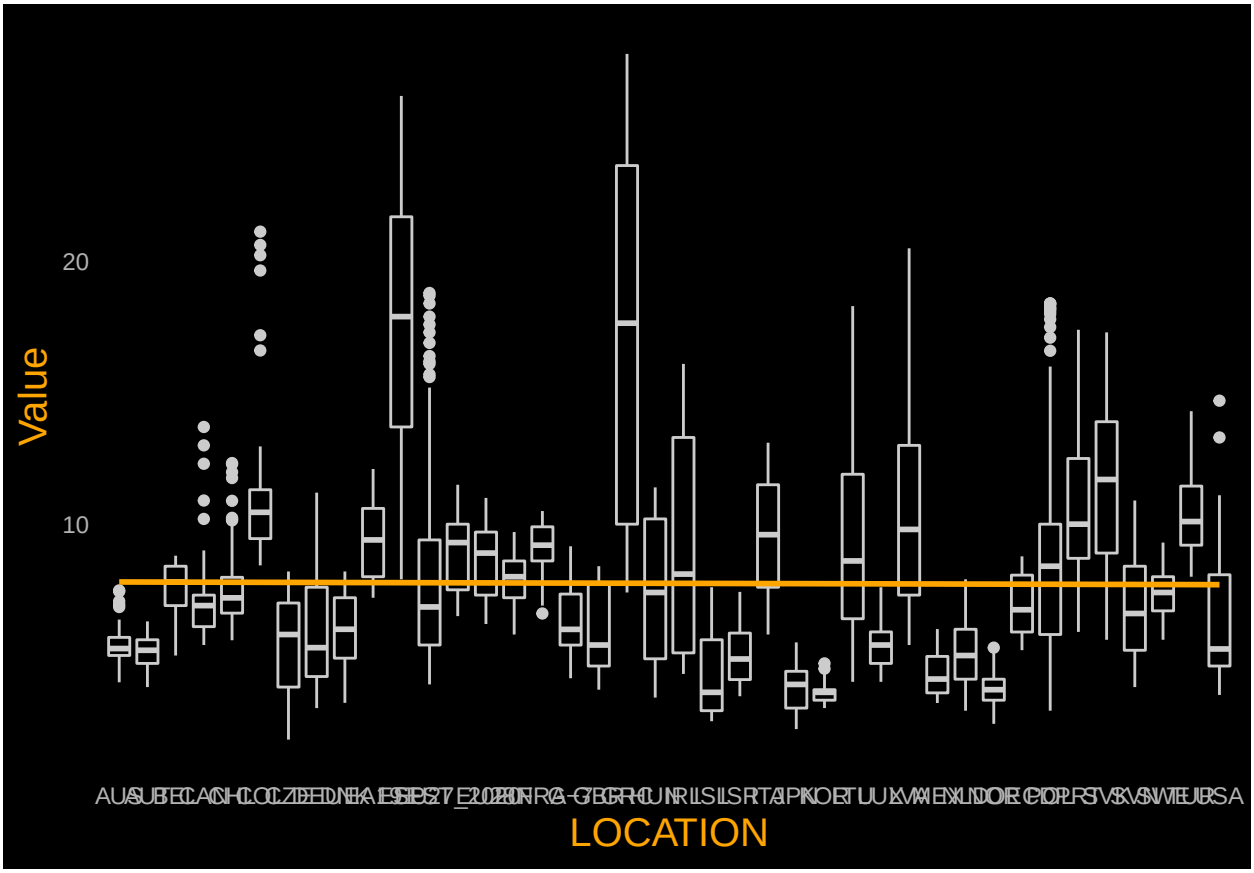


-

We can add the a line chart grouped

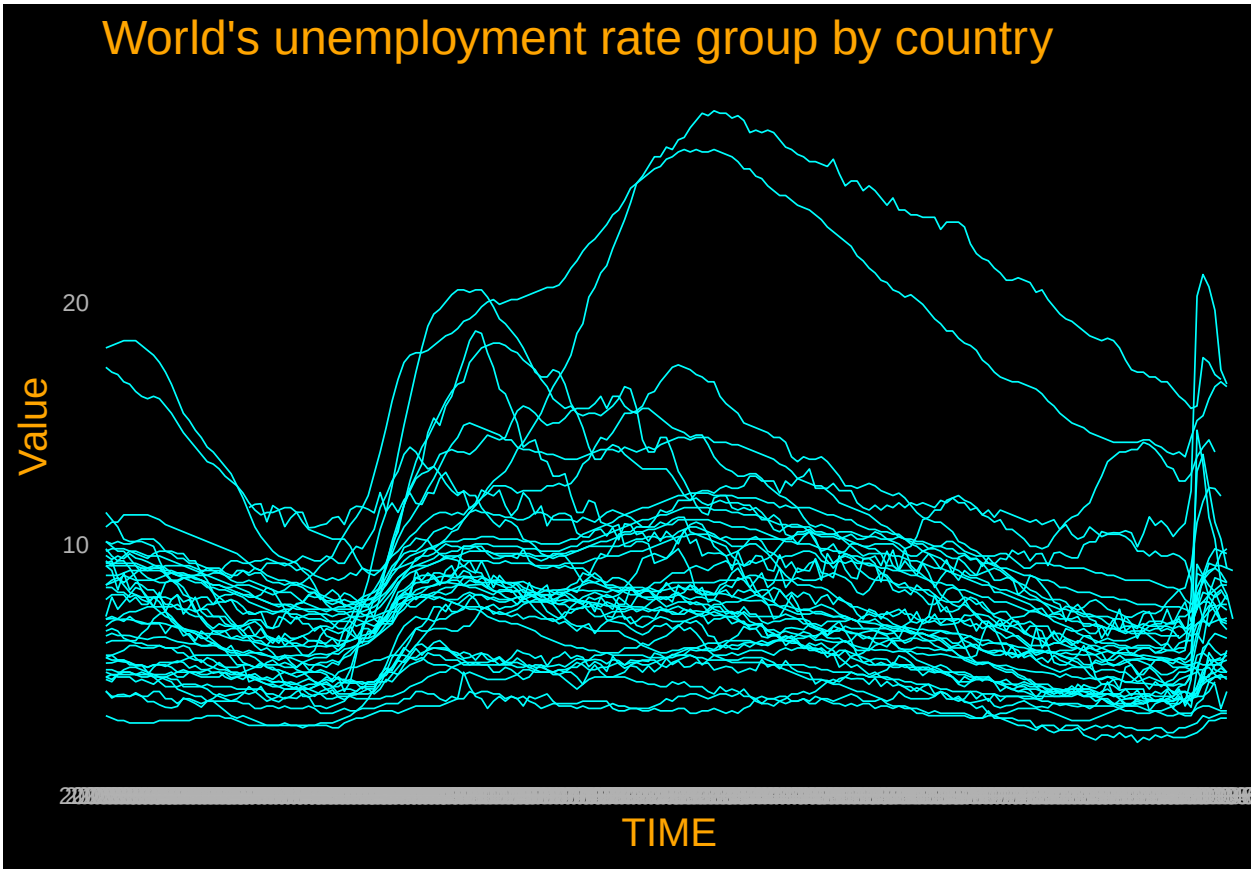
```
h2 + geom_boxplot() + geom_smooth(aes(group = 1), method = "lm", se = FALSE, colour=trend_color)
```

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



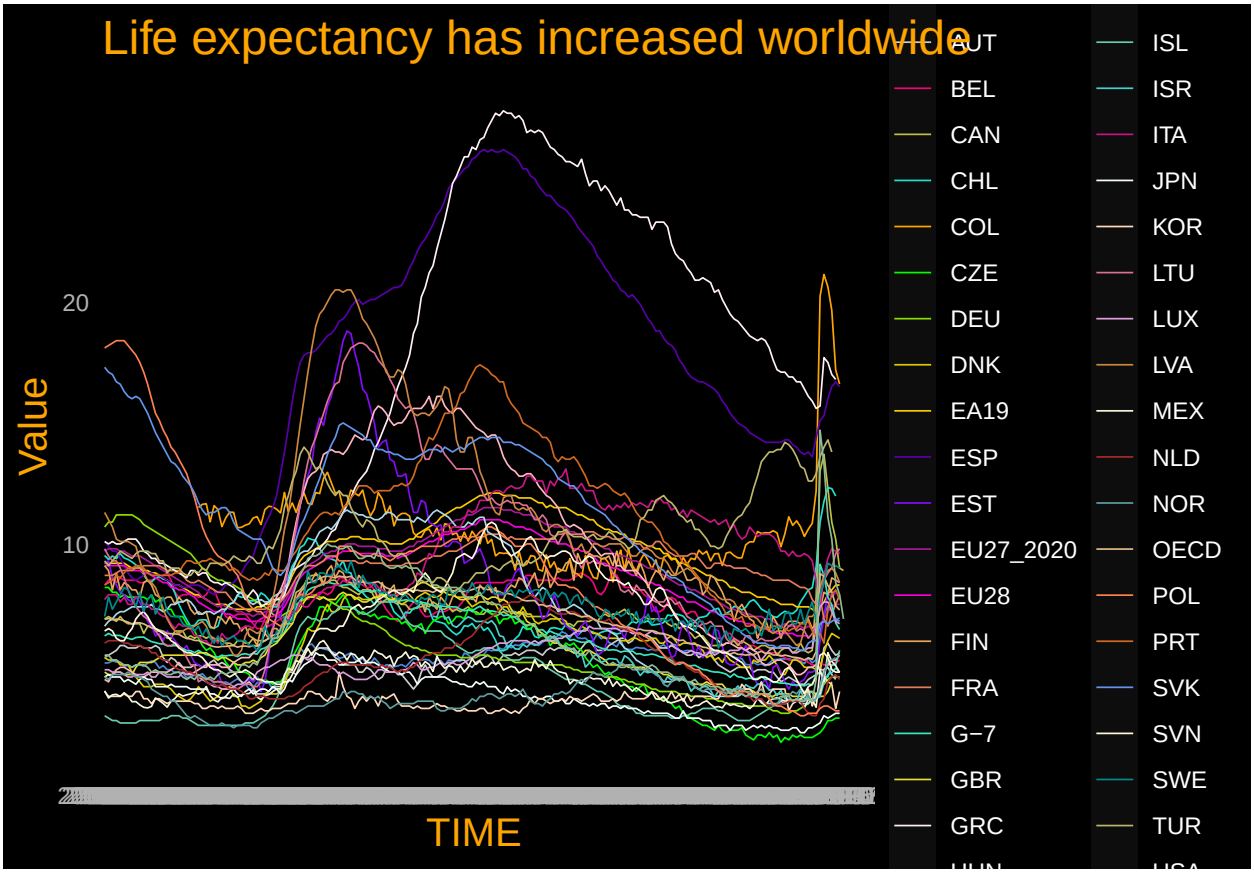
- #General trend in unemployment rate

```
ggplot(data) +
  geom_line(aes (TIME, Value, group = LOCATION), lwd = 0.3, show.legend = FALSE, colour = "cyan") +
  labs(title = "World's unemployment rate group by country ")
```



- #Checking on country

```
ggplot(data) +
  geom_line(aes (TIME, Value, group = LOCATION, color= LOCATION), lwd = 0.3, show.legend = TRUE) +
  scale_color_manual(values=c("#478adb", "#cccccc", "#f20675", "#bcc048", "#1ce3cd","orange","green","#8
  labs(title = "Life expectancy has increased worldwide")
```



-

Zooming in to see only Europe

```
ggplot(subset(data, LOCATION ==  "FIN")) +
  geom_line(aes(TIME, Value, group = LOCATION), color= "cyan", show.legend = FALSE) +
  labs(title = "unemployment rate in Finland - detecting an outlier")
```

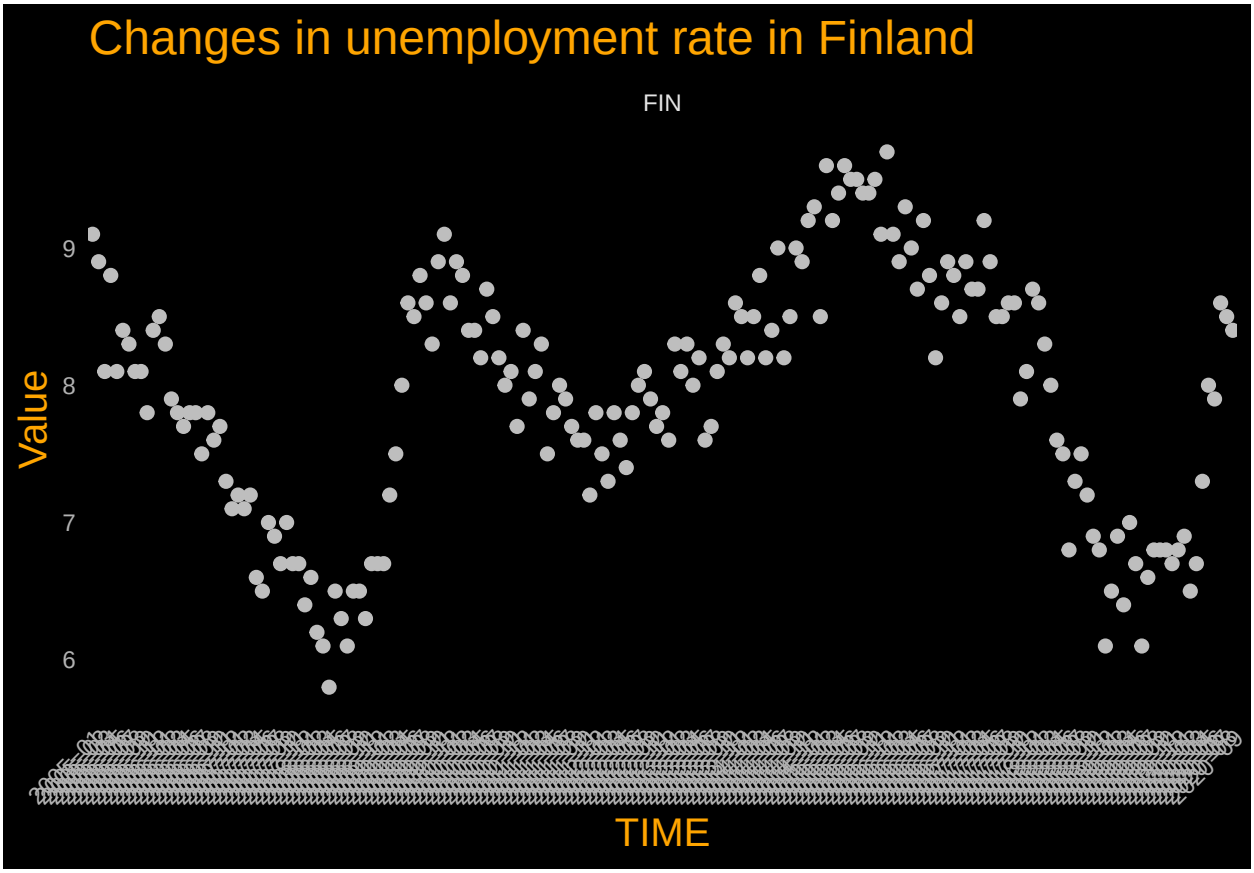
-

Select only Finland in order to understand the outlier

```
finland <- dplyr::filter(data, LOCATION == "FIN")
```

- #We can also show the trend as dots

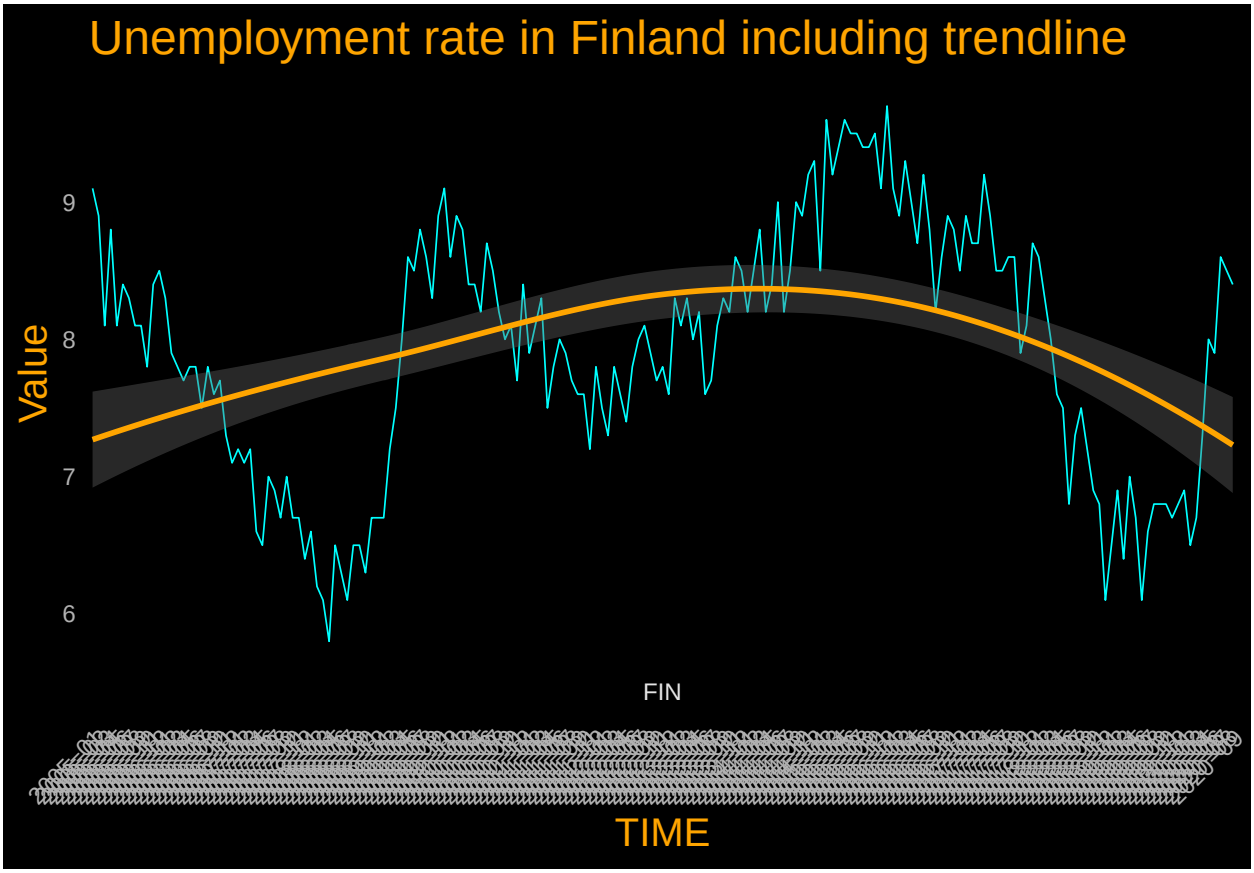
```
ggplot(finland, aes(TIME, Value)) +
  geom_point(color="grey", size=2) +
  facet_wrap(~LOCATION) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Changes in unemployment rate in Finland")
```



- #Adding a trend line - defining the method as loss

```
ggplot() +
  geom_line(data=finland, aes (TIME, Value, group = LOCATION), lwd = 0.3, show.legend = FALSE, color= "
  facet_wrap(~LOCATION, ncol=5, strip.position = "bottom") +
  geom_smooth(data=finland, aes(TIME, Value, group = 1), lwd = 1, method = 'loess', span = 2, se = TRUE
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Unemployment rate in Finland including trendline")
```

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



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.