Introduction to R, Day 3 Handout

In this handout, we cover the following topics and R commands:

Topics

- Correlation
- Linear Regression
- Visualizing Bivariate Relationships
- More on Visualization: Visualization with ggplot2

R Commands

- Calculating correlation using `cor()`
- Fitting linear regression models using `lm()`
- Summarizing linear regression models using `summary()`
- Extracting coefficients, standard errors, t-values, and p-values from lm
- Obtaining model coefficients using `coef()`
- Creating a scatterplot using `plot()`
- Adding best fit lines to scatter plots using `abline()`
- Obtaining residuals using `resid()`
- Obtaining fitted values using `fitted()`
- Creating residual plots using `plot()`
- Placing multiple plots in one figure using `par`
- Creating graphs with ‘`ggplot2()`

Correlation

- Correlation is one of the most frequently used statistics to summarize the relationship between two variables, which measures the degree to which two variables are associated to each other. [1]
- More specifically, the `correlation coefficient` tells us whether the change in the average values of one variable are related to the change in the average values of the other variable.
- Helps us answer two important Qs 1.Is there a positive/negative relationship between X and Y? 2.Is there a weak/strong relationship between X and Y?
- Correlations ranges from -1 to 1. A score of -1 means a perfect negative association and a score of 1 a perfect positive asssociation between the two variables, i.e. 0.81 would count as a strong positive correlation. A score of zero means that there is no association.

⊙ We will specifically focus on `Pearson's r`, which is the correlation coefficient for a `linear` relationship. Correlation is often not suitable for representing a nonlinear relationship.

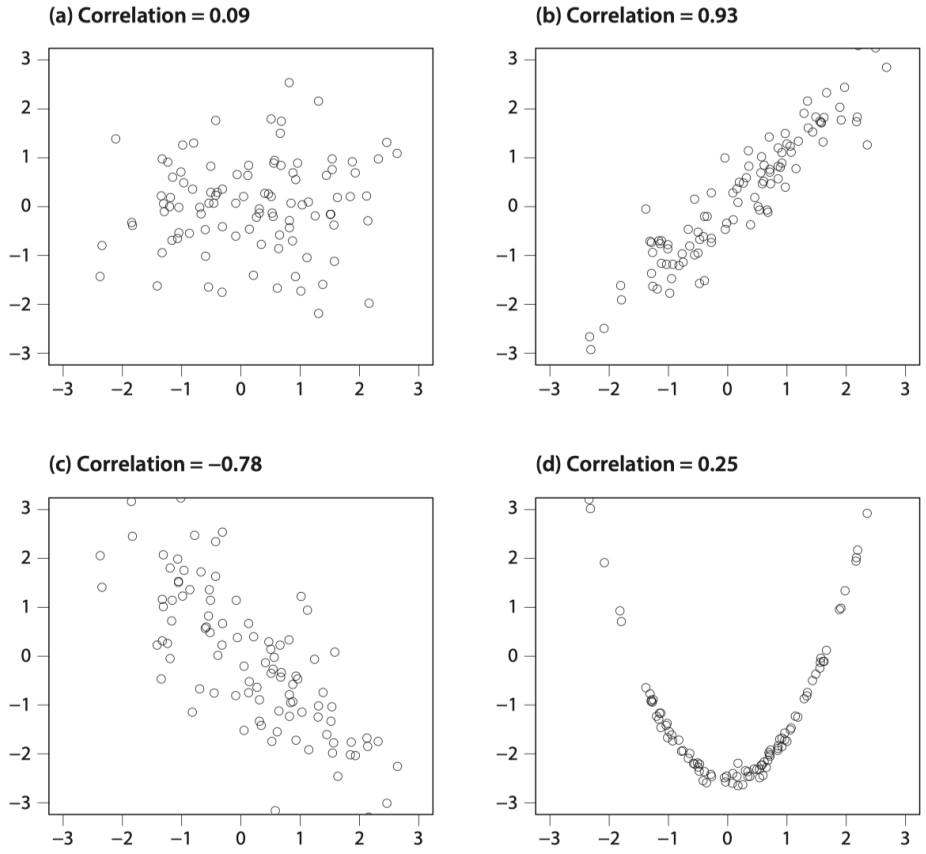


Figure 4.3. Correlation Coefficients and Patterns of the Data Cloud in Scatter Plots.

Figure A1: Source: Imai, QSS, 142

[1] Some people refer to as correlation coefficient.

- We calculate correlations in R with the `cor()` function. It takes two arguments x and y, as follows:

```
cor(x,y)
```

- To makes things more concrete, we will illustrate how to calculate a correlation coefficient in R with some running examples.

Trade Union Membership and Government Structure

- In their 1991 exchange article, Wallterstein and Stephens discusses the relationship between union membership and government structure (the control of government by left-wing and socialist parties). (((Wallerstein argues that the size of the labor force provides the most important determinant of variation in union density)) [2].
- The table below shows the variables and their description:

- `country`. Country
- `union`. Percentage of workers who belong to a union
- `left`. Extent to which parties of the left have controlled government
- `size`. Size of the labor force
- `concent` Measure of economic concentration in top four industries

- Load the data

```
tradedat <- read.csv("day3data/tradeunion.csv")
```

- Calculate the correlation using the `cor()` function

```
cor(tradedat$union, tradedat$left)
```

```
## [1] 0.6779826
```

```
## we can round to two decimal places
round(cor(tradedat$union, tradedat$left), digits = 2)
```

```
## [1] 0.68
```

```
round(cor(tradedat$union, tradedat$left), 2)
```

```
## [1] 0.68
```

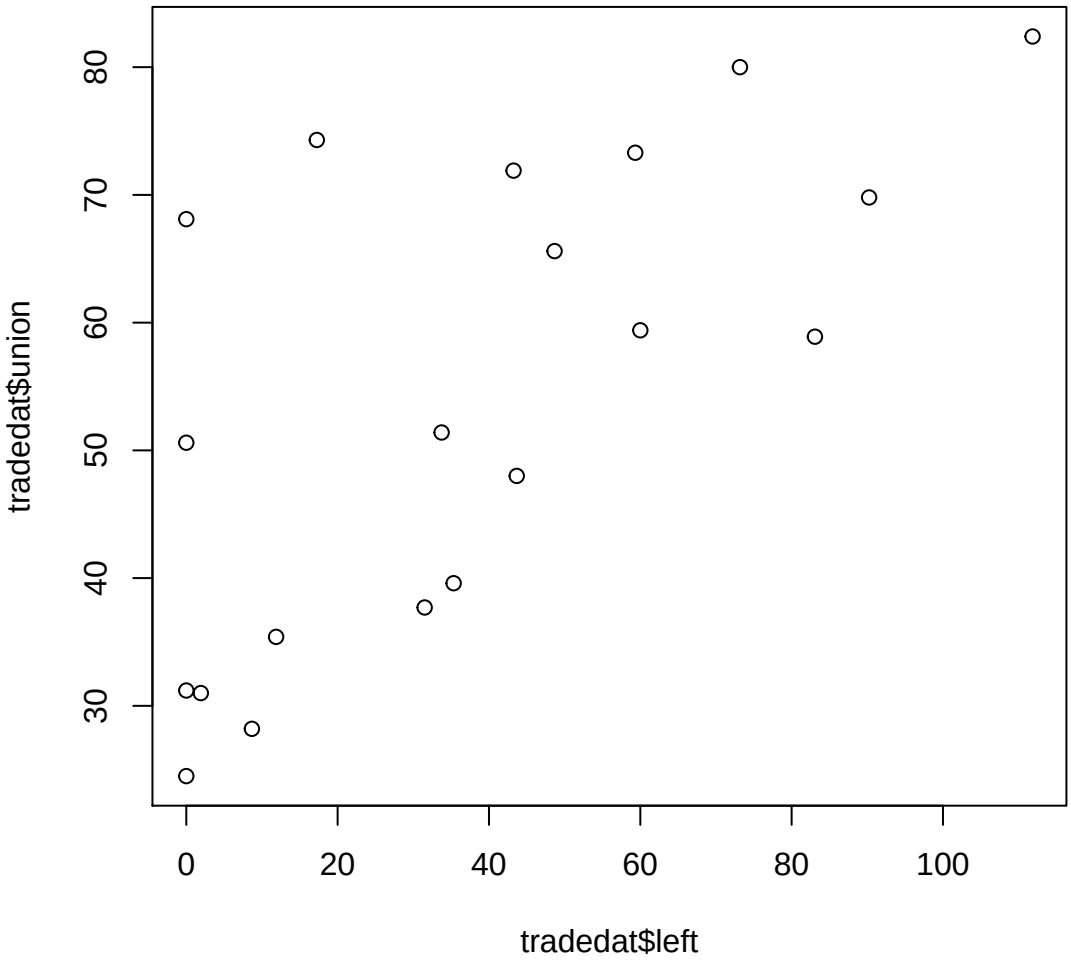
- This correlation coefficient suggests that there is a strong positive relationship between union membership and control of government by left-wing parties.

[2] Example adapted from Kosuke Imai's online teaching materials

Correlation and Scatterplots

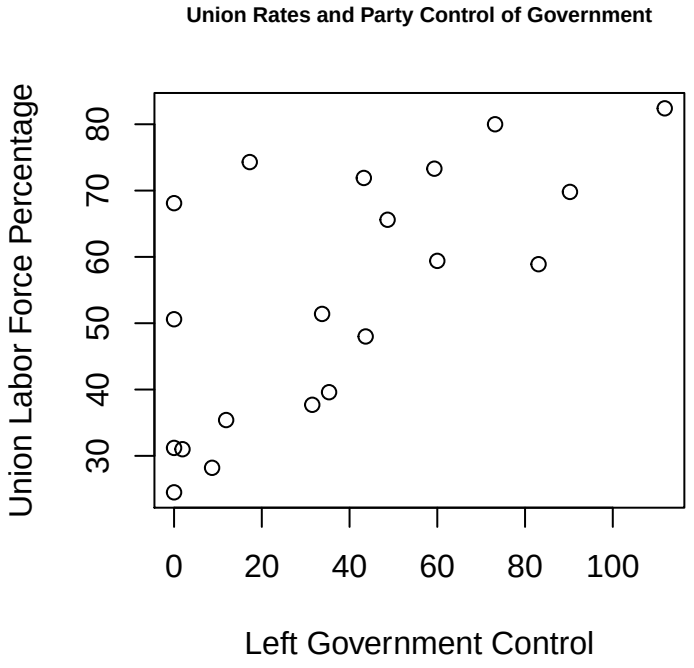
- Scatterplots allows us to visually compare two variables measured on the same set of units by plotting the value of one variable against that of the other for each unit.
- We can visually inspect the correlation between union membership and control of government by left-wing parties.
- To create a scatterplot in R, we use the `plot()` function.
- The syntax for this function is plot(x, y), where x and y are vectors of horizontal and vertical coordinates, respectively.
- Similar to yesterday, let's take a two step procedure to plotting: Step 1: Creating a very simple plot, Step2: Iteratively making our graph beautiful

Step 1: A Very Simple Plot (default)

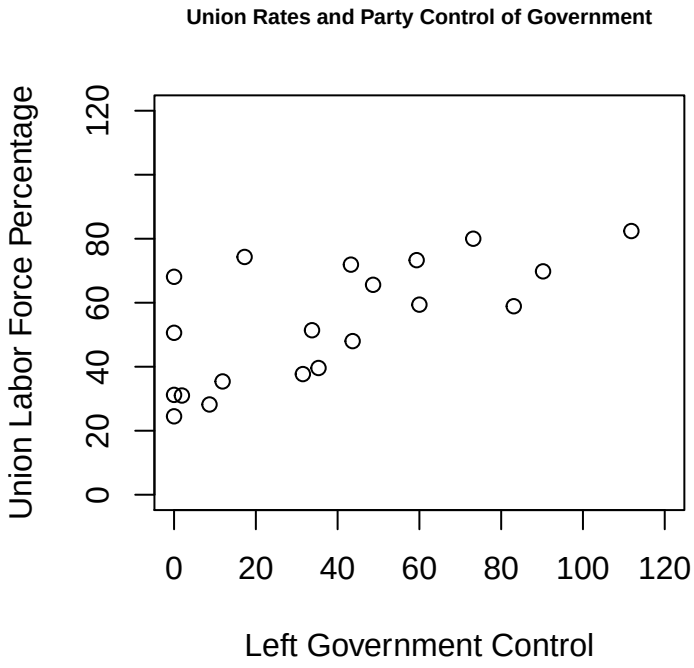


Step 2: Iteratively Making it Better

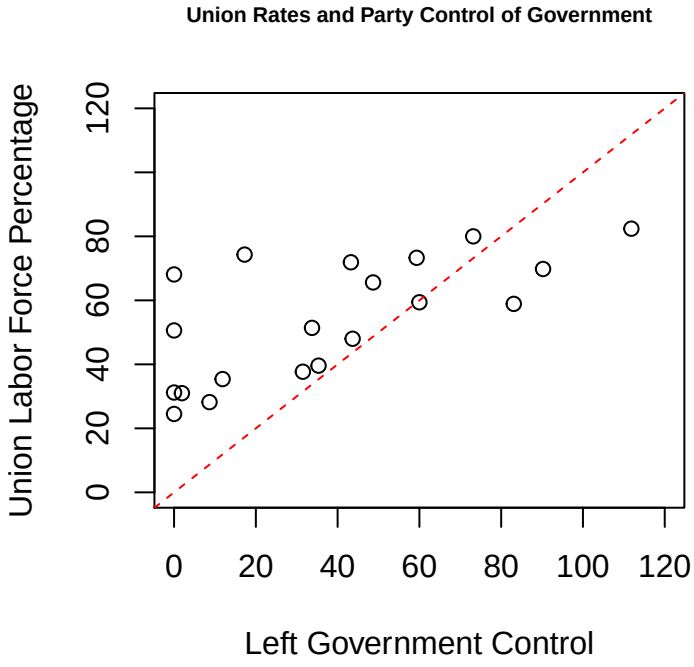
- Label the main title and the x-axis, decrease the font size of the title



- Fix the x-axis and y-axis limits, set the size of points,



- Add a 45 degree line



Linear Regression

- Correlation describes a linear relationship between two variables. However, such a relationship is best characterized by using a *linear model*.

The linear regression model estimated with ordinary least squares (OLS) is a workhorse model in Political Science. Even when a scholar uses a more advanced method that may make more accurate assumptions about his or her data—such as probit regression, a count model, or even a uniquely crafted Bayesian model— the researcher often draws from the basic form of a model that is linear in the parameters. By a similar token, many of the R commands for these more advanced techniques use functional syntax that resembles the code for estimating a linear regression. Therefore, an understanding of how to use R to estimate, interpret, and diagnose the properties of a linear model lends itself to sophisticated use of models with a similar structure (Mogomani)

- To fit a linear regression model in R, we use the `lm()` (`l`linear `m`model) function. This function takes a formula of the form Y ~ X as the main argument that connects one variable y to one (or more) variables x, taken from a specified data frame.

```
lm(y ~ x, data)
```

- y is the outcome (dependent) variable and x is a single independent variable. In R, the intercept will be automatically added to the regression model.
- The argument fed into the `lm` function is of the form `y ~ x`, where `~` shows that it is a formula. In other words, we are assuming a linear model that has the following form:

$$y_i = \alpha + \beta x_i + \epsilon_i$$

Note:The 'lm' function allows us to estimate the coefficients $\widehat{\alpha}$ and $\widehat{\beta}$ such that the line (best fit line) is as close to the data as possible.

Trade Union Membership and Leftist Government Control Revisited

- To explore linear regression and its diagnostics, we will continue from the study that focuses on the relationship between trade union membership and government structure.
- Let's `regress` trade union membership on the control of government by left-wing parties.

```
lm1 <- lm(union ~ left, data = tradedat)
lm1
```

```
##
## Call:
## lm(formula = union ~ left, data = tradedat)
##
## Coefficients:
## (Intercept)         left
##     39.8841       0.3764
```

- `lm1` is the object that contains all the information we need to interpret the linear regression.

- The output shows that the estimated intercept is 39.8841 whereas the estimated slope is 0.6604. The slope represents the average change in Y given a one unit change in X. In other words, according to the model, the estimated effect of a one percentage point increase in the control of government by left parties on the trade union membership is captured by the estimated regression coefficient `0.3764`.
- To access `the coefficients` from the regression, we can call them directly within `lm1` object.

```
lm1$coef  # Both coefficients
```

```
## (Intercept)        left
##  39.8840609   0.3763868
```

```
lm1$coef[1]  # The intercept
```

```
## (Intercept)
##    39.88406
```

```
lm1$coef[2]  # The slope
```

```
##      left
## 0.3763868
```

- When needed, we can also store information about our coefficients by creating an object:

```
coefs <- lm1$coef
coefs[1]
```

```
## (Intercept)
##    39.88406
```

```
coefs[2]
```

```
##      left
## 0.3763868
```

- Is the effect we observe significant? Now, we can explore the components of the regression model in more detail. To obtain a detailed summary of the regression output, we can use the `summary()` function.

```
summary(lm1)
```

```
##
## Call:
## lm(formula = union ~ left, data = tradedat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.384 -10.269  -3.558  10.808  28.216
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.88406    4.81269   8.287 1.48e-07 ***
## left         0.37639    0.09619   3.913  0.00102 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.16 on 18 degrees of freedom
## Multiple R-squared:  0.4597, Adjusted R-squared:  0.4296
## F-statistic: 15.31 on 1 and 18 DF,  p-value: 0.001019
```

- The `p-value`, in the last column, is approximately 0.001, which is below the standard 0.05 threshold. Therefore, we can reject the null hypothesis that there is no relationship between trade union membership and control of government by left-wing parties. However, we didn't consider many other factors that might have an effect on the trade union membership.
- We can also use `coef` here to access the coefficients, standard errors, t-statistics, and p-values of the regression:

```
summary(lm1)$coef
```

```
##               Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 39.8840609 4.81268695 8.287275 1.477466e-07
## left         0.3763868 0.09618625 3.913104 1.019215e-03
```

```
summary(lm1)$coef
```

```
##               Estimate Std. Error  t value     Pr(>|t|)
## (Intercept) 39.8840609 4.81268695 8.287275 1.477466e-07
## left         0.3763868 0.09618625 3.913104 1.019215e-03
```

- Suppose that we want to access the p-value for the effect of leftist government control

```
summary(lm1)$coef[2, 4]
```

```
## [1] 0.001019215
```

- Lastly, let's calculate the confidence intervals for th effect of leftist government control

```
mean.gov <- summary(lm1)$coef[2, 1]  # mean of leftist gov
se.gov <- summary(lm1)$coef[2, 2]  # standard error of leftist gov
ci.gov <- c(mean.gov - 1.96 * se.gov, mean.gov + 1.96 *
    se.gov)
ci.gov
```

```
## [1] 0.1878618 0.5649119
```

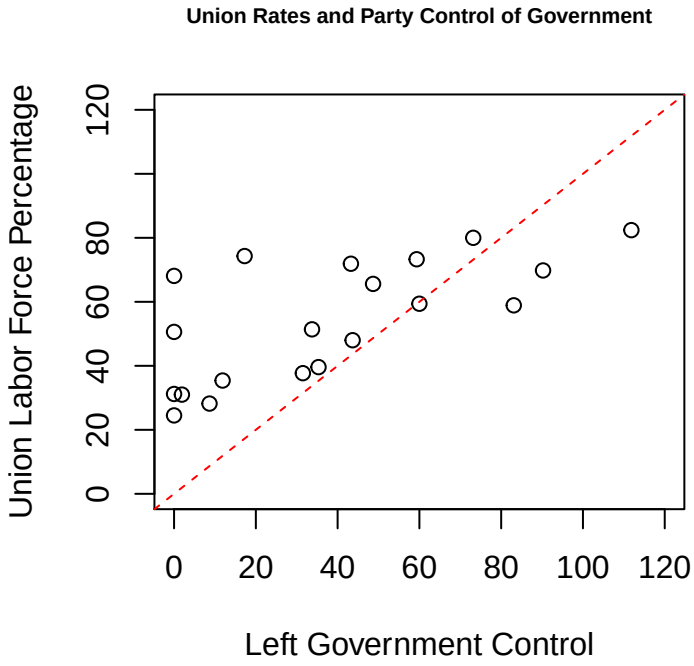
- Alternatively, we can calculate the confidence interval in a more straightforward way:

```r
confint(lm1, level = 0.95)
```

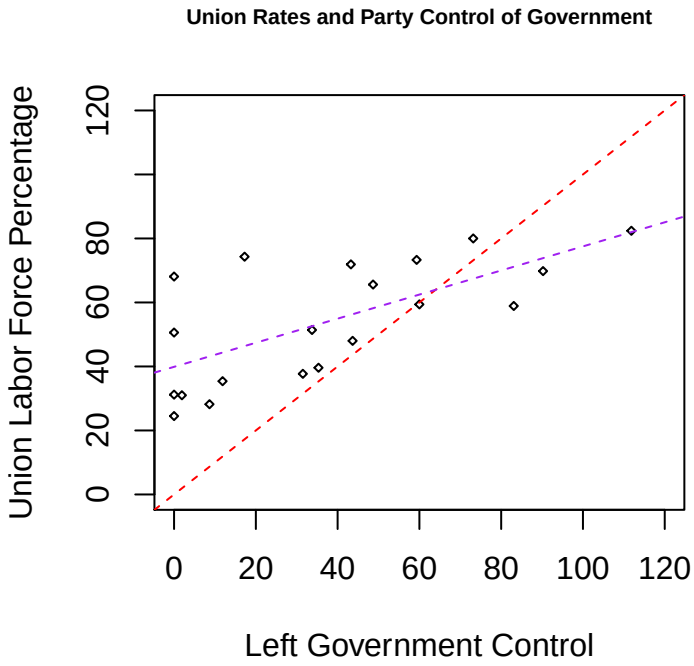
```
##                 2.5 %     97.5 %
## (Intercept) 29.772981 49.9951410
## left         0.174307  0.5784667
```

Regression and Scatterplots

- Previously, we computed the correlation between trade union membership and leftist government control. Using a scatterplot, we had created the following scatterplot:



- Based on our regression model, we can update this scatterplot by adding a `regression line` to the graph. To do so , we will use the `abline()` function:



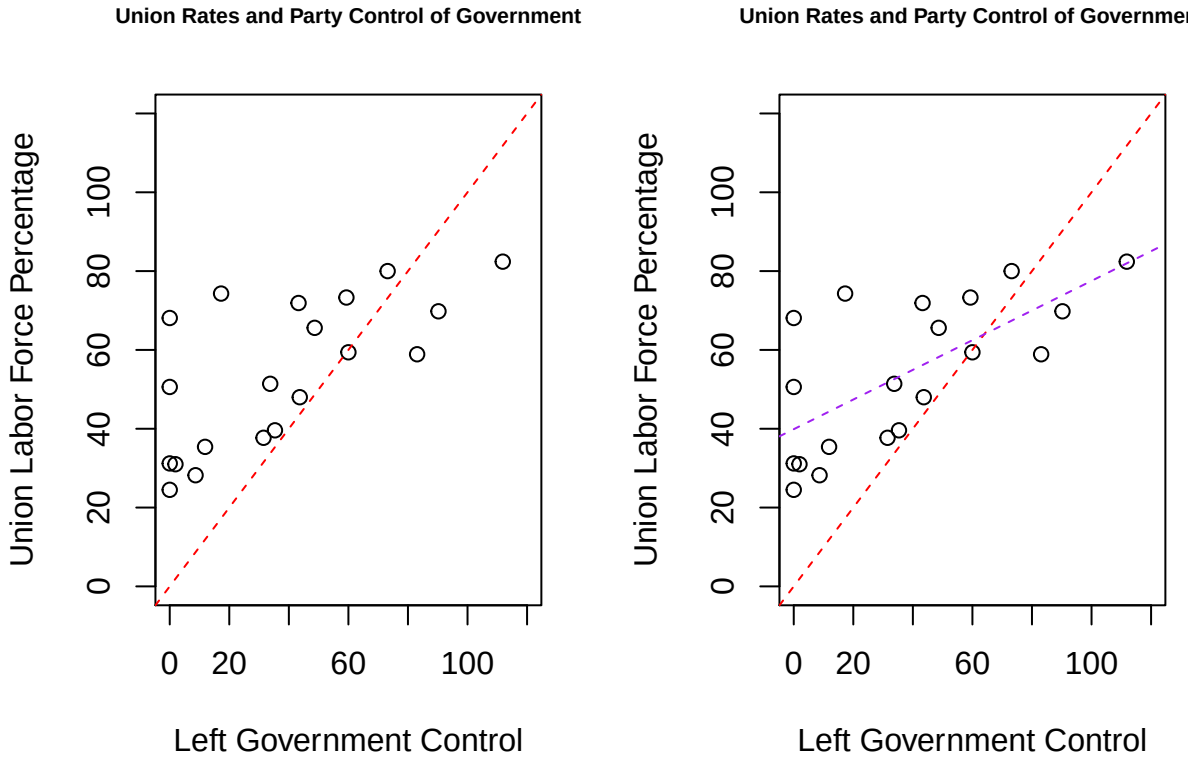
- Let's put these two scatterplots side by side in one figure. To do so, we will use a function called `par`.

  par(mfrow = c(number of rows, number of columns))

where the par function set the value of mfrow which determines how our plots will be arranged in the plots window

- In our case, we only have two scatterplots. It would make sense to put them side by side by setting our number of rows to `1` and the number of columns to `2`:

```r
par(mfrow = c(1, 2))
plot(tradedat$left, tradedat$union, cex = 1, xlim = c(0,
    120), ylim = c(0, 120), xlab = "Left Government Control",
    ylab = "Union Labor Force Percentage", main = "Union Rates and Party Control of Government",
    cex.main = 0.7)
abline(0, 1, lty = 2, col = "red")
plot(tradedat$left, tradedat$union, cex = 1, xlim = c(0,
    120), ylim = c(0, 120), xlab = "Left Government Control",
    ylab = "Union Labor Force Percentage", main = "Union Rates and Party Control of Government",
    cex.main = 0.7)
abline(0, 1, lty = 2, col = "red")
abline(lm1, lty = 2, col = "purple")
```



More on Visualization: Visualization with ggplot2

Base R plots vs. ggplot2

What is ggplot2?

- The `tidyverse` is a set of related packages for R. The `ggplot2` package is one of its com- ponents. The other pieces make it easier to get data into R and manipulate it once it is there. `ggplot2` is a powerful, versatile, and widely used visualization package for R (Wickham 2016).
- Within the `tidyverse` family, there are some very packages such as `dplyr` that are commonly used with 'ggplot2. `Dplyr` is a very powerful package for manipulating your data similar to the things we have done before but by using different functions.
- To cut a long story short, we will install the `tidyverse` package which includes but not limited to `ggplot2`.
- Note: We need to install a package only once, but we will need to load it at the beginning of each R session with `library()` if you want to use the tools it contains.
- To install a package: > install.packages("packagename")
- To load the package > library("packagename")

```
## -- Attaching packages --------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.1     v dplyr   0.8.5
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

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

- To check your loaded packages, you can use `search()`

```
## Load packages
search()
```

```
##  [1] ".GlobalEnv"        "package:forcats"   "package:stringr"
##  [4] "package:dplyr"     "package:purrr"     "package:readr"
##  [7] "package:tidyr"     "package:tibble"    "package:ggplot2"
## [10] "package:tidyverse" "package:formatR"   "package:knitr"
## [13] "package:foreign"   "package:stats"     "package:graphics"
## [16] "package:grDevices" "package:utils"     "package:datasets"
## [19] "package:methods"   "Autoloads"         "package:base"
```

The Intuition Behind ggplot2

- The most important thing to get used to with ggplot is the way you use it to think about `the logical structure` of your plot. Once you get used to the structure, you will realize that visualization process will more or less involve very similar sequence of steps
- Let's see step by step how `ggplot2` works with an example.
- We will use a dataset called `gapminder` for our running example. This dataset includes variables about a large number of countries, each observed over several years.
- Our aim is to plot `life expectancy` against `per capita GDP` for all country-years in the data.
- This dataset that belongs to an R package called `gapminder. That's why we need to install and load a package called`gapminder'.

```
library(gapminder)
```

Ggplot with Steps: Creating Scatterplots

- The first two steps will help us to create the core component of our plot. Once we complete these two main steps, the process will be additive: we will add `layers` to this core plot.

Note: *In R, usually functions cannot simply be added to objects. Rather, they take objects as inputs and produce objects as outputs. But the objects created by ggplot() are special. This makes it easier to assemble plots one piece at a time, and to inspect how they look at every step* (Healy,DViz, 60)

- Let's check the first few rows of the dataset quicky.

```
head(gapminder)
```

```
## # A tibble: 6 x 6
##   country     continent  year lifeExp      pop gdpPercap
##   <fct>       <fct>     <int>   <dbl>    <int>     <dbl>
## 1 Afghanistan Asia       1952    28.8  8425333      779.
## 2 Afghanistan Asia       1957    30.3  9240934      821.
## 3 Afghanistan Asia       1962    32.0 10267083      853.
## 4 Afghanistan Asia       1967    34.0 11537966      836.
## 5 Afghanistan Asia       1972    36.1 13079460      740.
## 6 Afghanistan Asia       1977    38.4 14880372      786.
```

1. Tell the `ggplot()` function what dataset we are using. We would also want to store this plot, so let's create an object as well.

```
library(gapminder)
```

1. Tell `ggplot()` what data we are using. The first argument we give is the `data` argument.

```
myp <- ggplot(data = gapminder)
```

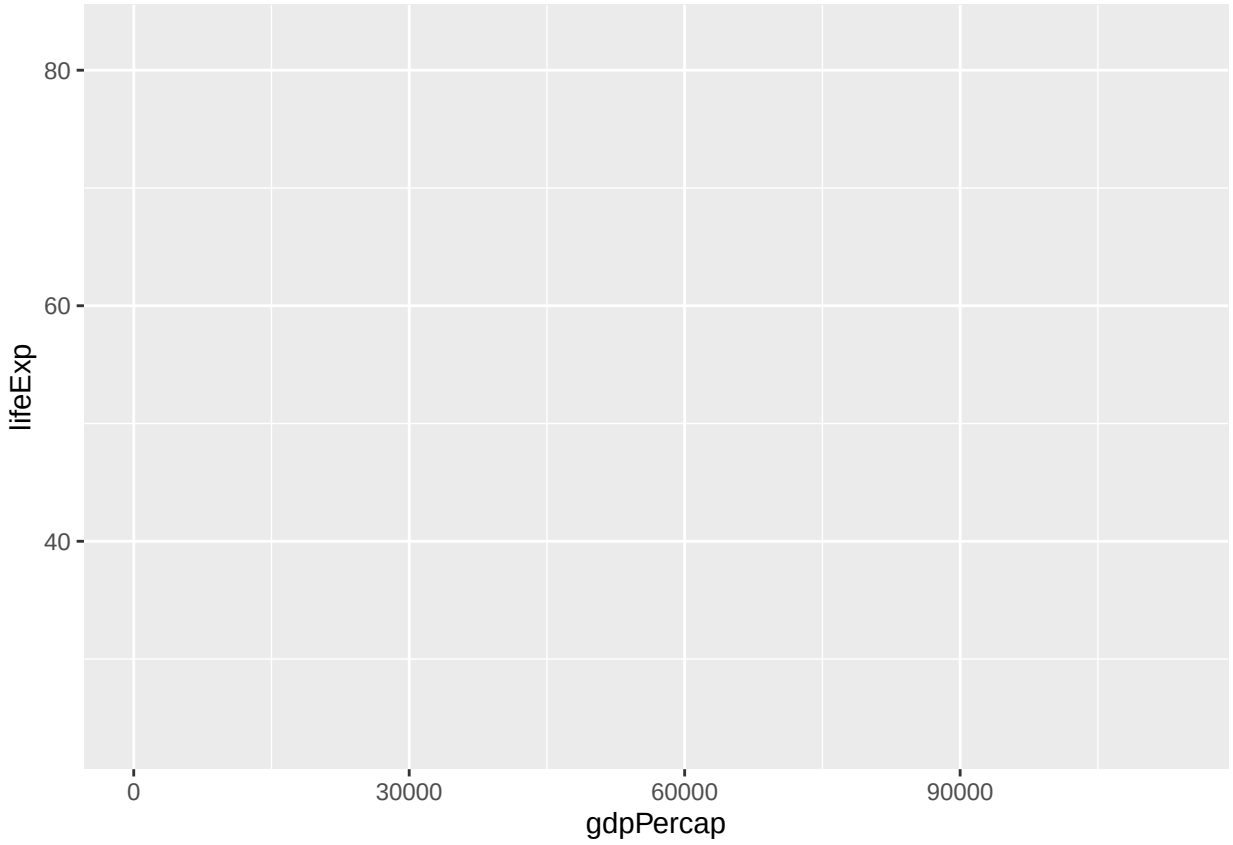
2. We need tell which specific variables we would like to `map` to each other. The second argument we give is the `mapping` argument.

Note: The `mapping = aes(...)` argument links variables to things you will see on the plot. The x and y values are the most obvious ones. Other aesthetic mappings can include, for example, color, shape, size, and line type

```
myp <- ggplot(data = gapminder, mapping = aes(x = gdpPercap,
    y = lifeExp))
```

- Let's try to display the plot by printing our object

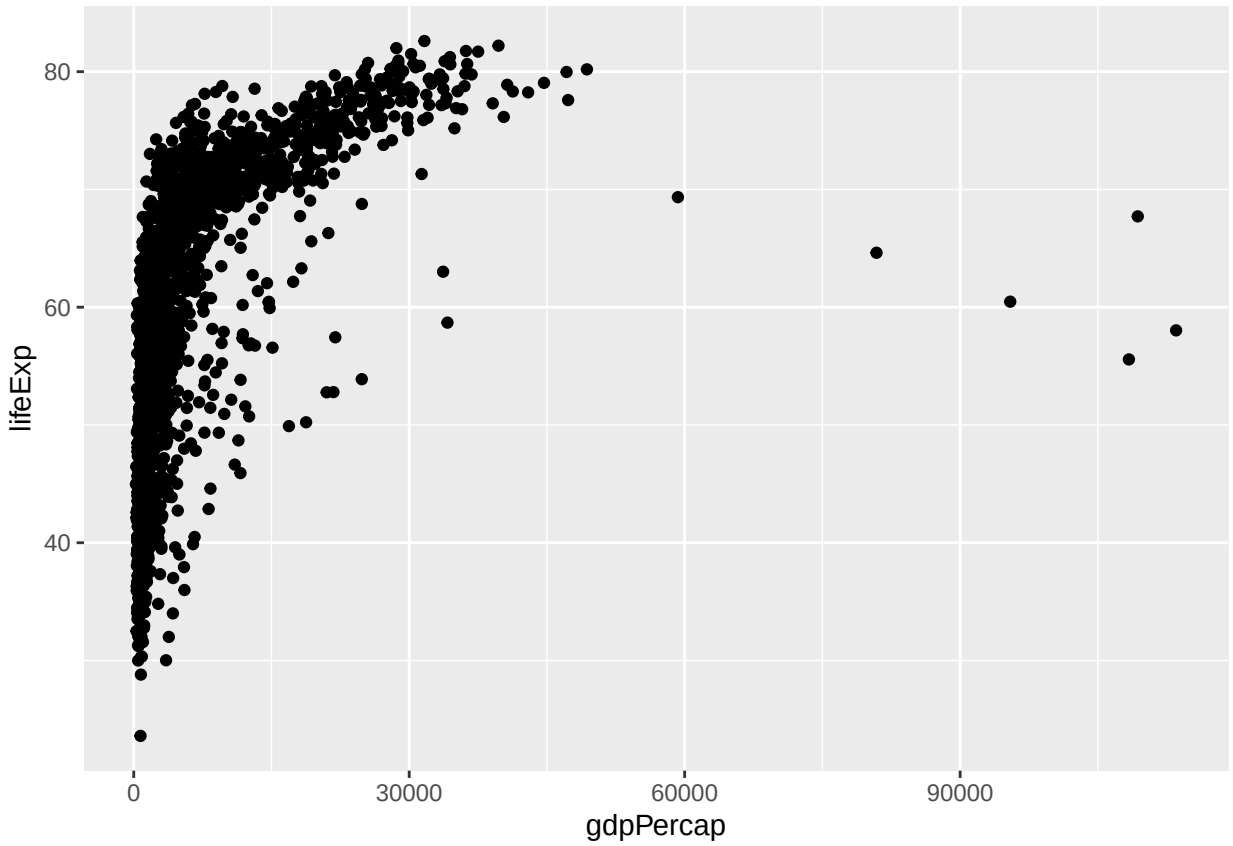
```
myp
```



- What do you see?
- Thus far, we didn't passed any argument about what `sort` of plot we want. Do we want a scatterplot, histogram etc.? We need to tell this.

3. To tell the sort of plot we want, we need to add a new layer to the basic plot. Simply, adding a new layer means adding a new argument to the plot. Because we want to create a scatterplot, we will use the function `geom_point()` to add a layer.

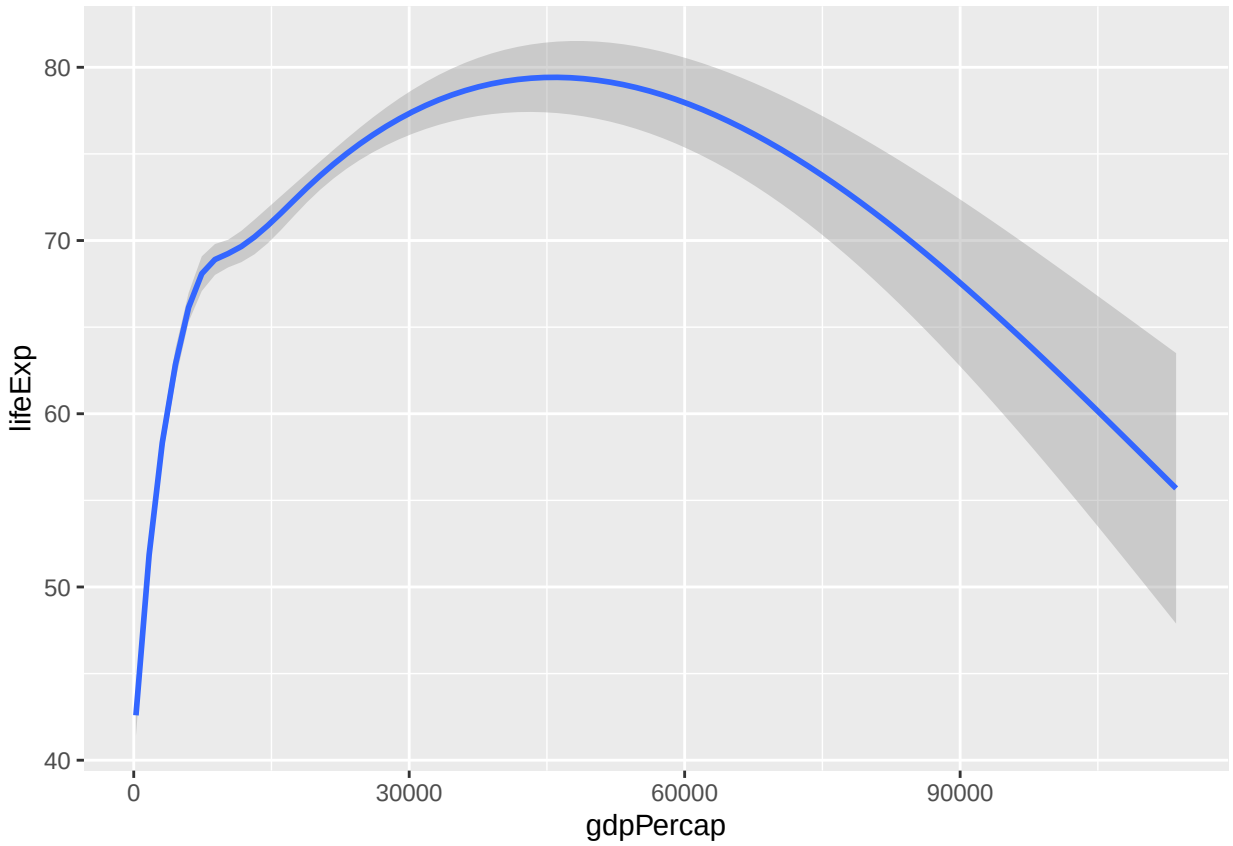
```
myp <- ggplot(data = gapminder, mapping = aes(x = gdpPercap,
    y = lifeExp)) + geom_point()
myp
```



3.1 We can change the sort of the plot, for example, we can plot a smoothed line shaded in a ribbon showing the standard error for the line. We will use `geom_smooth()` as our layer.

```
myp <- ggplot(data = gapminder, mapping = aes(x = gdpPercap,
    y = lifeExp)) + geom_smooth()
myp
```

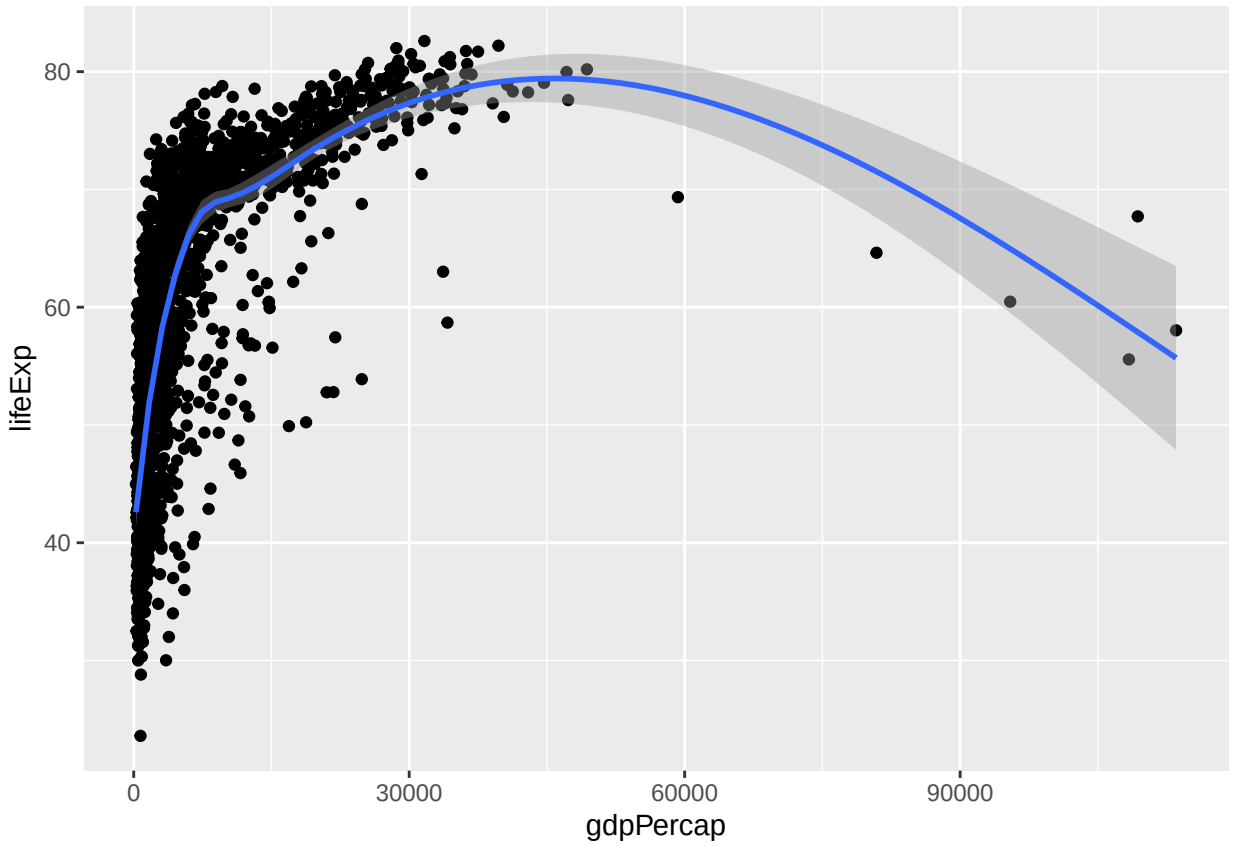
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



3.2 We can also see the points and lines together by adding both layers.

```
myp <- ggplot(data = gapminder, mapping = aes(x = gdpPercap,
    y = lifeExp)) + geom_point() + geom_smooth()
myp
```

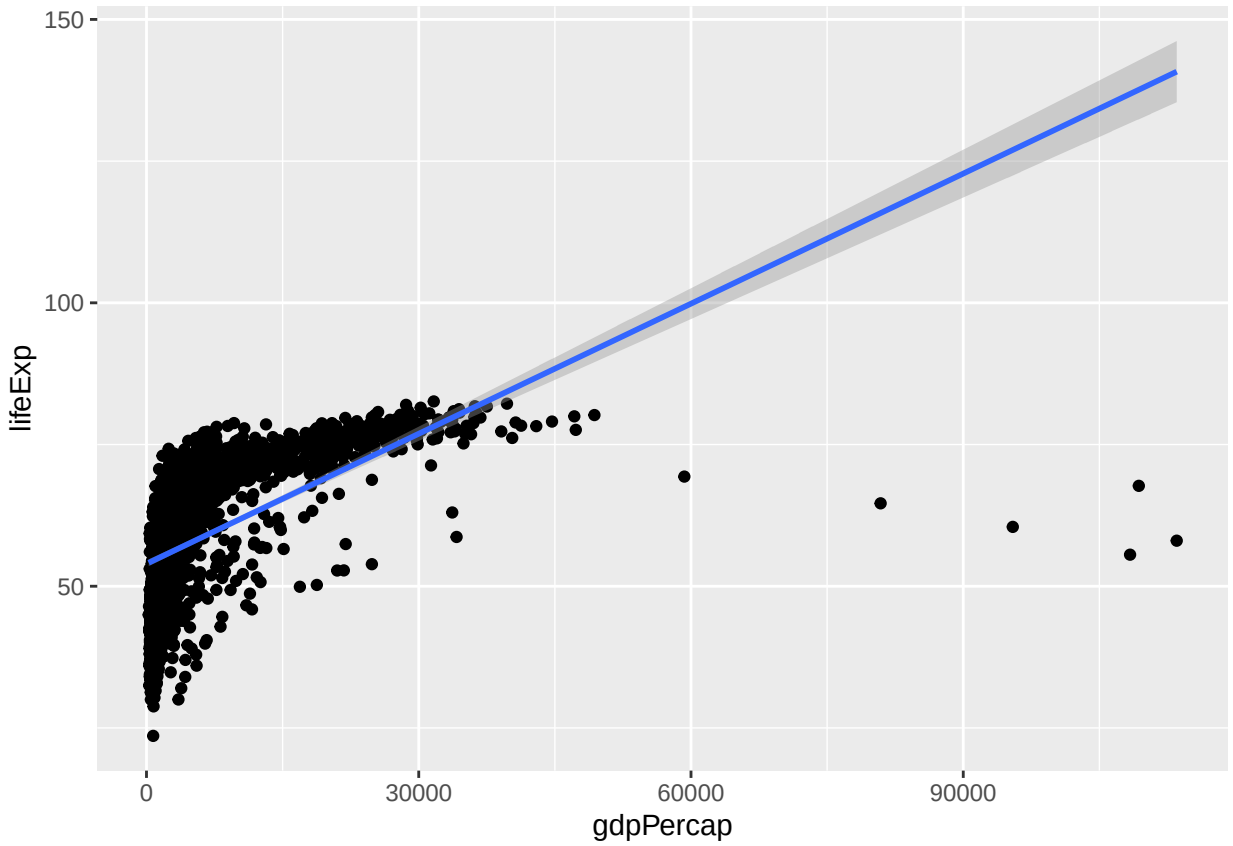
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



3.3. In your R console, you will see a message that tells \textit{the geom_smooth() function is using a method called gam}, which means it has fit a generalized additive model. Let's change this by setting the argument of the `geom_smooth()` function.

```
myp <- ggplot(data = gapminder, mapping = aes(x = gdpPercap,
    y = lifeExp)) + geom_point() + geom_smooth(method = "lm")
myp
```

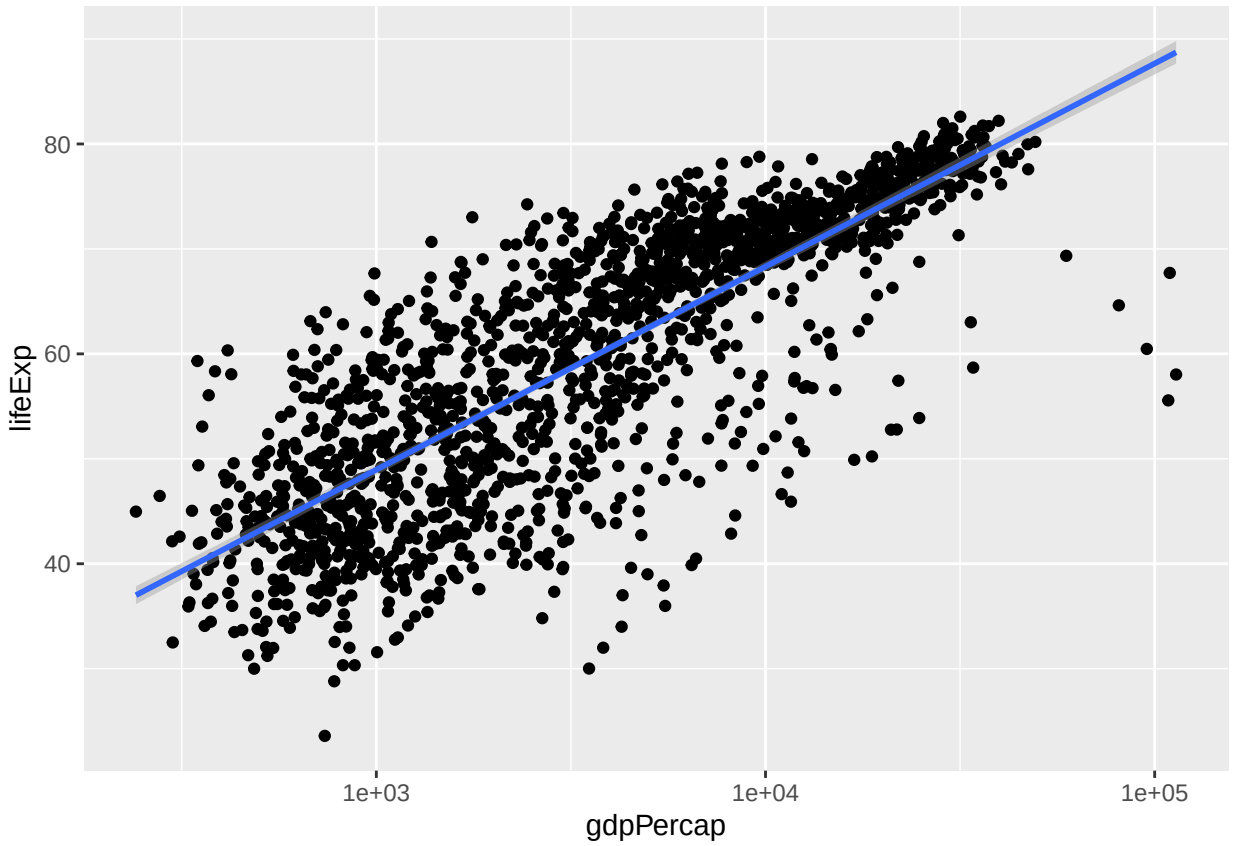
```
## `geom_smooth()` using formula 'y ~ x'
```



4.GDP per capita is not normally distributed across our country years. The graph will look better if we scale the x-axis to a log. We will add another layer called `scale_x_log10()`

```
myp <- ggplot(data = gapminder, mapping = aes(x = gdpPercap,
    y = lifeExp)) + geom_point() + geom_smooth(method = "lm") +
    scale_x_log10()
myp
```

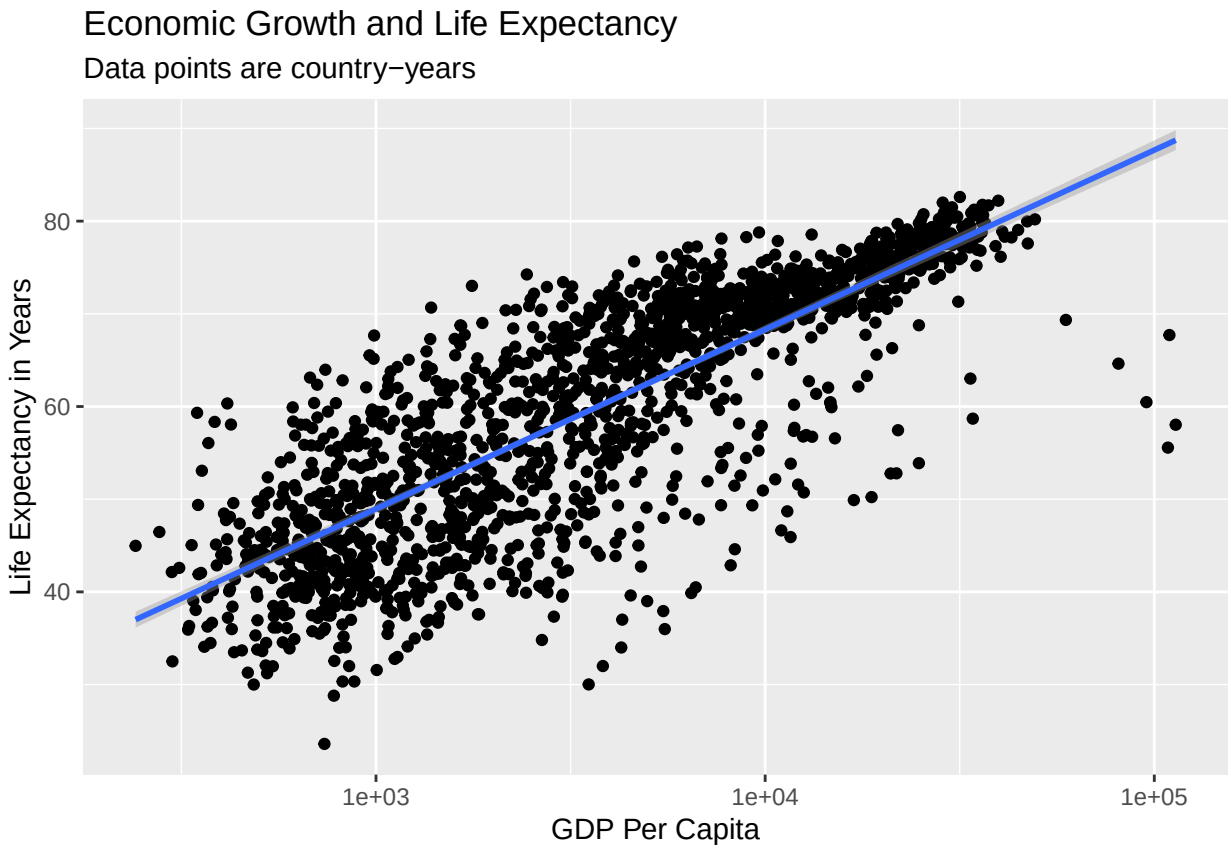
```
## `geom_smooth()` using formula 'y ~ x'
```



5. It seems like we have a simple plot that looks okay. Let's work on the details: add a main title and a subtitle, fix the names of x-axis and y-axis lables,

```
myp <- ggplot(data = gapminder, mapping = aes(x = gdpPercap,
    y = lifeExp)) + geom_point() + geom_smooth(method = "lm") +
    scale_x_log10() + labs(x = "GDP Per Capita", y = "Life Expectancy in Years",
    title = "Economic Growth and Life Expectancy", subtitle = "Data points are country-years")
myp
```

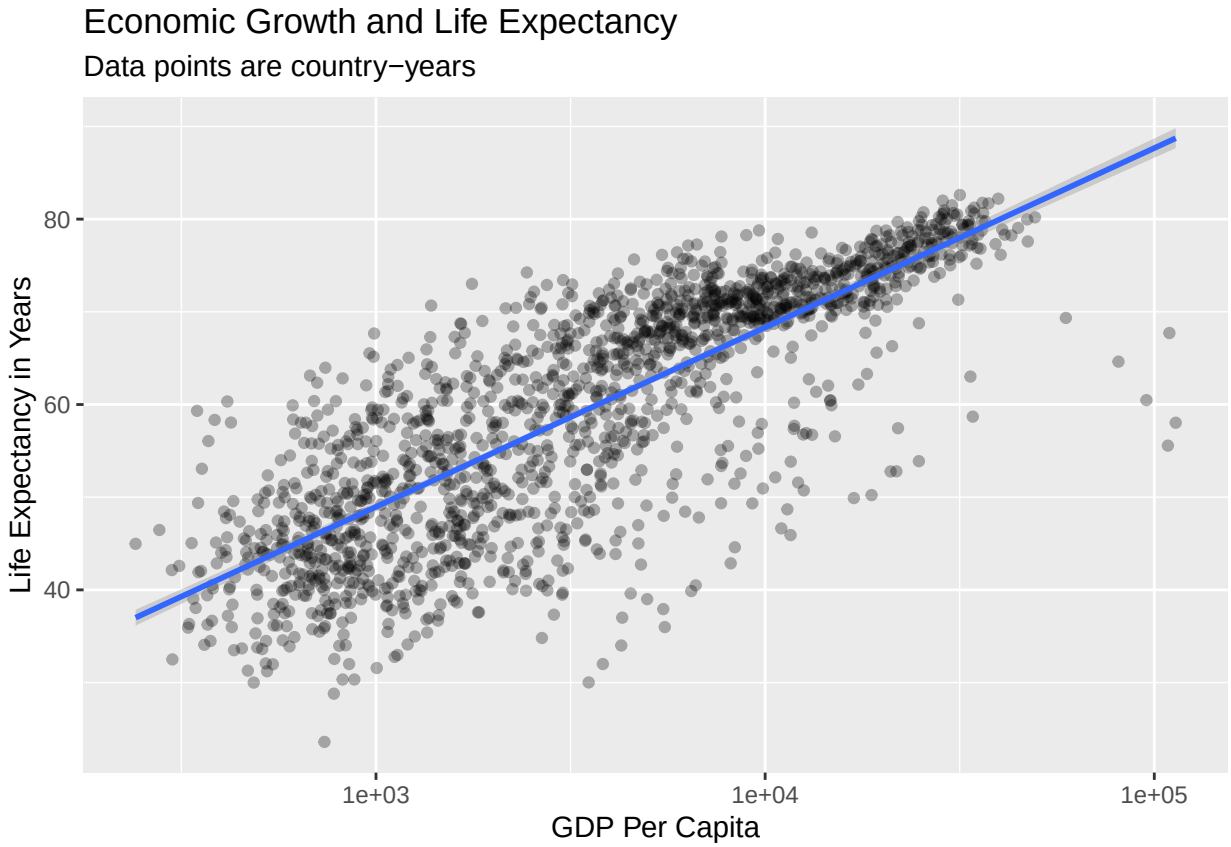
```
## `geom_smooth()` using formula 'y ~ x'
```



6. Let's make the points transparent. To do so, we will feed an argument into the `geom_point` that takes care of the points. "alpha" is an aesthetic property that points (and some other plot elements) have, and to which variables can be mapped. It controls how transparent the object will appear when drawn.

```
myp <- ggplot(data = gapminder, mapping = aes(x = gdpPercap,
    y = lifeExp)) + geom_point(alpha = 0.3) + geom_smooth(method = "lm") +
    scale_x_log10() + labs(x = "GDP Per Capita", y = "Life Expectancy in Years",
    title = "Economic Growth and Life Expectancy", subtitle = "Data points are country-years")
myp
```

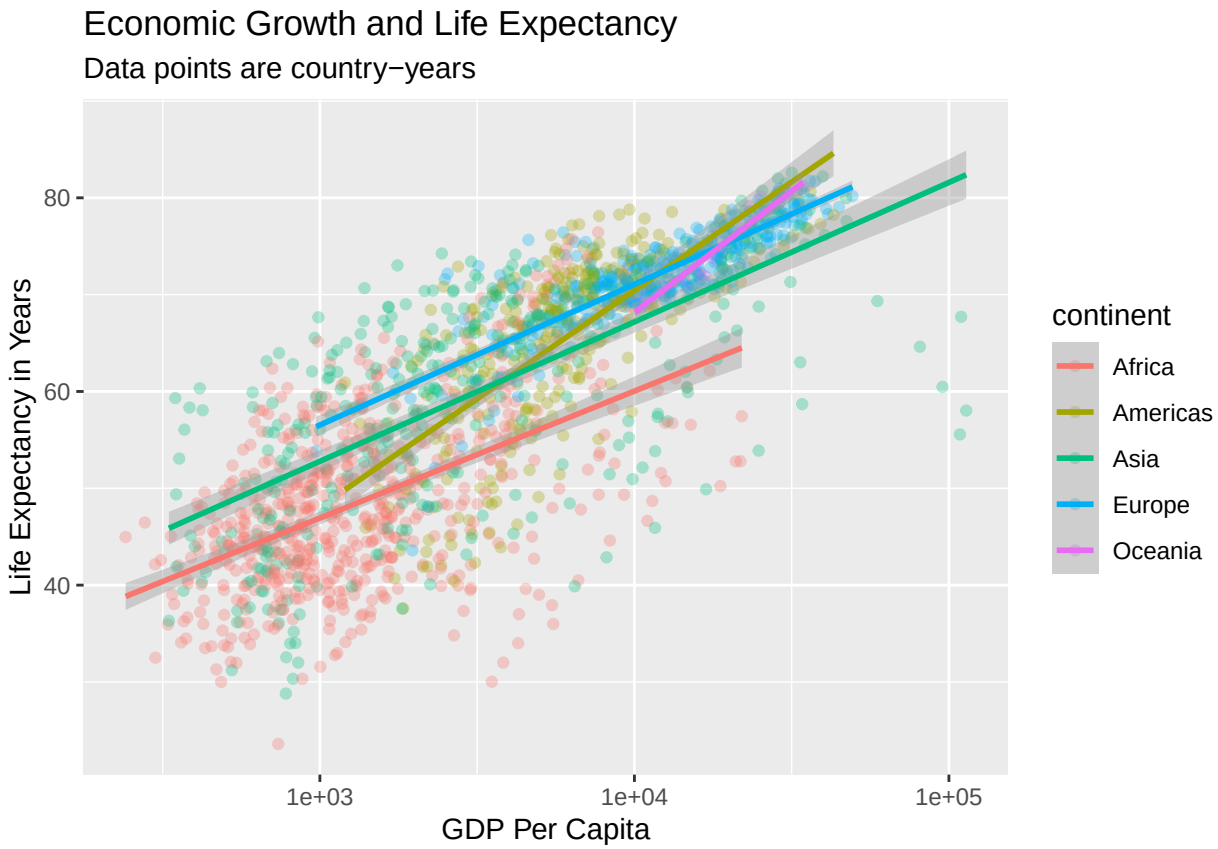
```
## `geom_smooth()` using formula 'y ~ x'
```



7. We have a variable called continet. Let's try whether we can color individual data points by continent. Because continent is a variable, we will pass this argument inside the `mapping()` argument

```
myp <- ggplot(data = gapminder, mapping = aes(x = gdpPercap,
    y = lifeExp, color = continent)) + geom_point(alpha = 0.3) +
    geom_smooth(method = "lm") + scale_x_log10() + labs(x = "GDP Per Capita",
    y = "Life Expectancy in Years", title = "Economic Growth and Life Expectancy",
    subtitle = "Data points are country-years")
myp
```

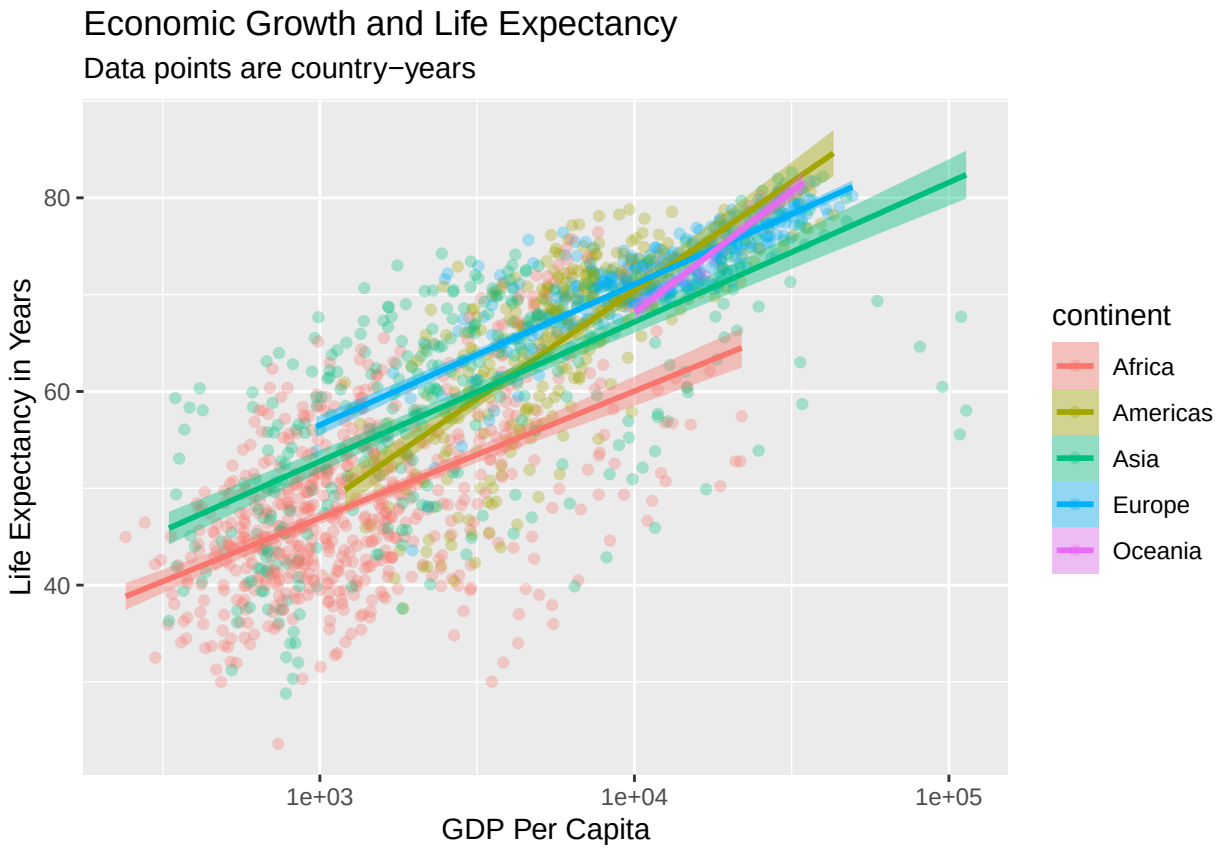
```
## `geom_smooth()` using formula 'y ~ x'
```



8. We can also shade the standard error ribbon of each line to match its dominant color for each continent.

```
myp <- ggplot(data = gapminder, mapping = aes(x = gdpPercap,
    y = lifeExp, color = continent, fill = continent)) +
    geom_point(alpha = 0.3) + geom_smooth(method = "lm") +
    scale_x_log10() + labs(x = "GDP Per Capita", y = "Life Expectancy in Years",
    title = "Economic Growth and Life Expectancy", subtitle = "Data points are country-years")
myp
```

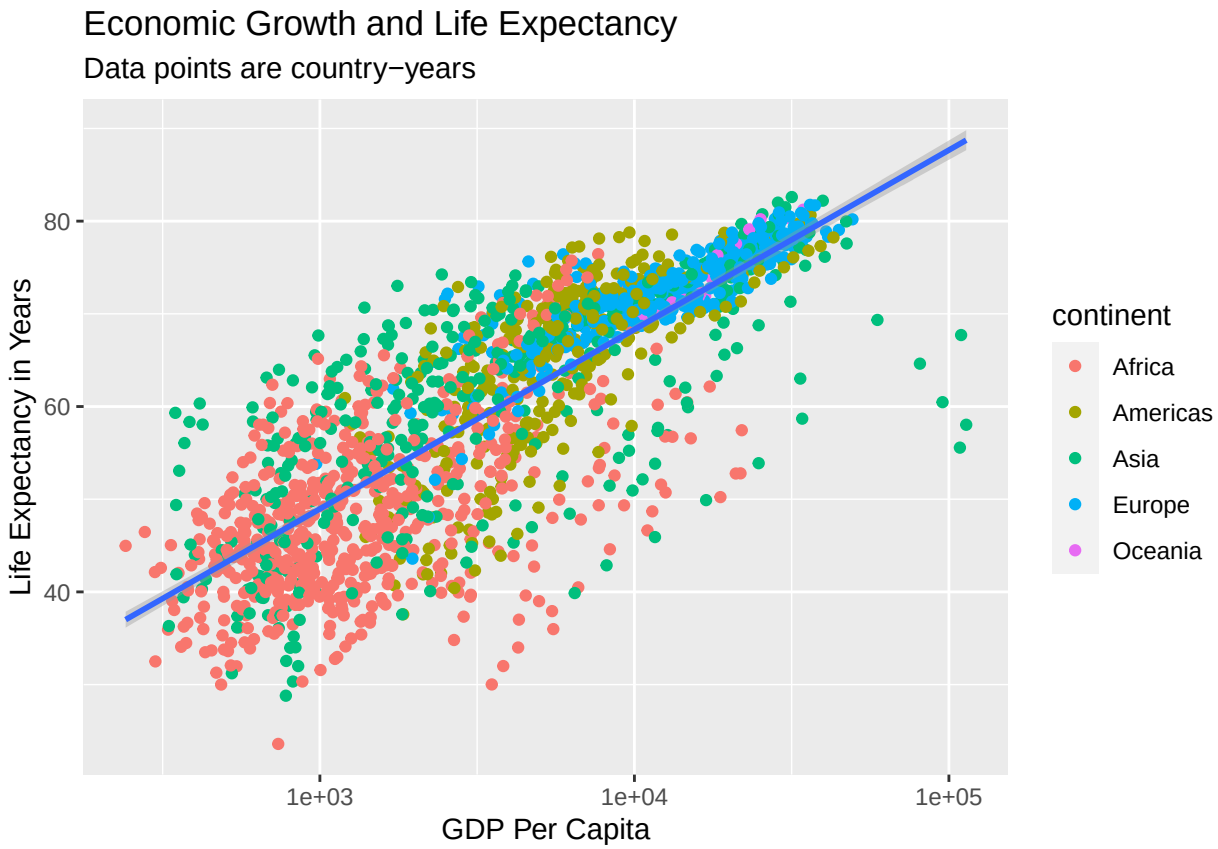
```
## `geom_smooth()` using formula 'y ~ x'
```



8.1 Instead of having five seperate smooth lines, we might consider having one smooth line while keeping the colors of the points. To do so, we need to tell our wish to color the continents to the `geom_points()` instead of the `mapping()` function.

```
myp <- ggplot(data = gapminder, mapping = aes(x = gdpPercap,
    y = lifeExp)) + geom_point(mapping = aes(colour = continent)) +
    geom_smooth(method = "lm") + scale_x_log10() + labs(x = "GDP Per Capita",
    y = "Life Expectancy in Years", title = "Economic Growth and Life Expectancy",
    subtitle = "Data points are country-years")
myp
```

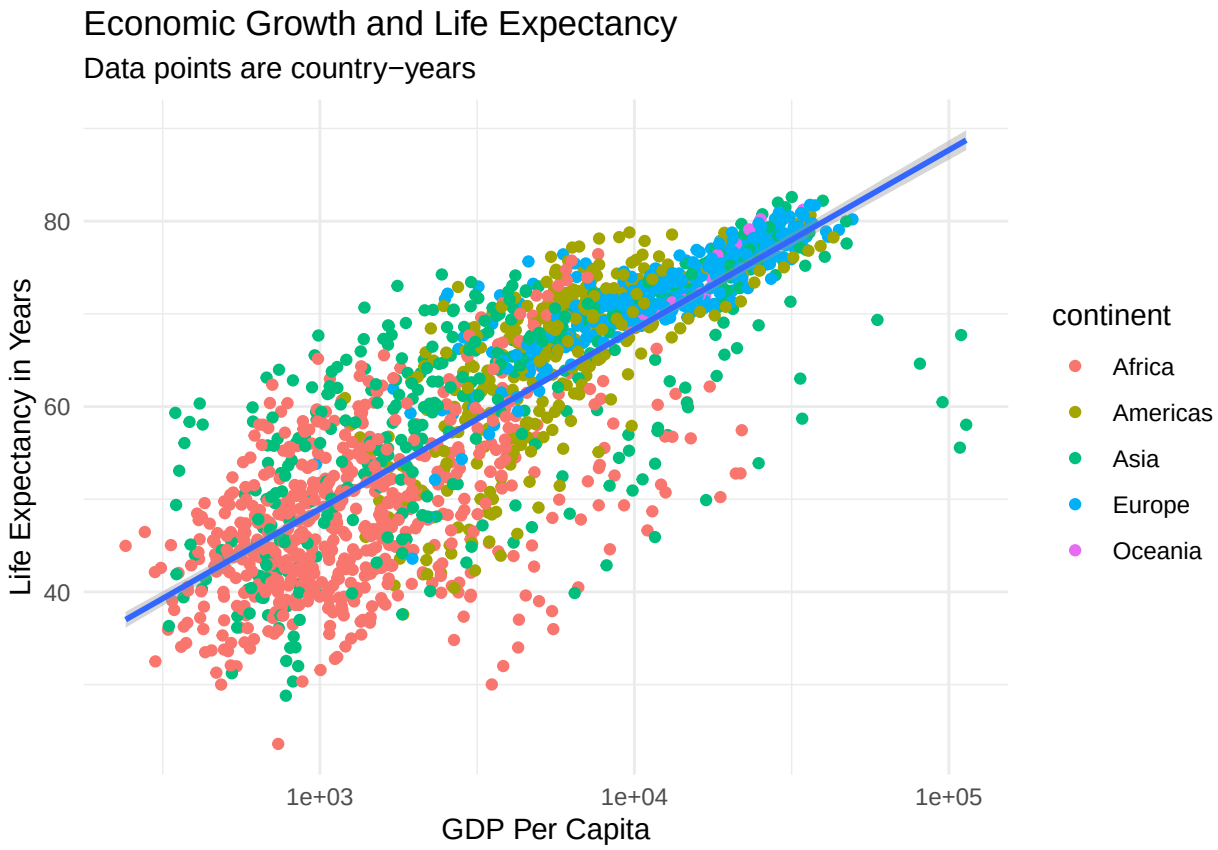
```
## `geom_smooth()` using formula 'y ~ x'
```



9 Lastly, let's change the grey background color by adding another layer to our plot

```
myp <- ggplot(data = gapminder, mapping = aes(x = gdpPercap,
    y = lifeExp)) + geom_point(mapping = aes(colour = continent)) +
    geom_smooth(method = "lm") + scale_x_log10() + labs(x = "GDP Per Capita",
    y = "Life Expectancy in Years", title = "Economic Growth and Life Expectancy",
    subtitle = "Data points are country-years") + theme_minimal()
myp
```

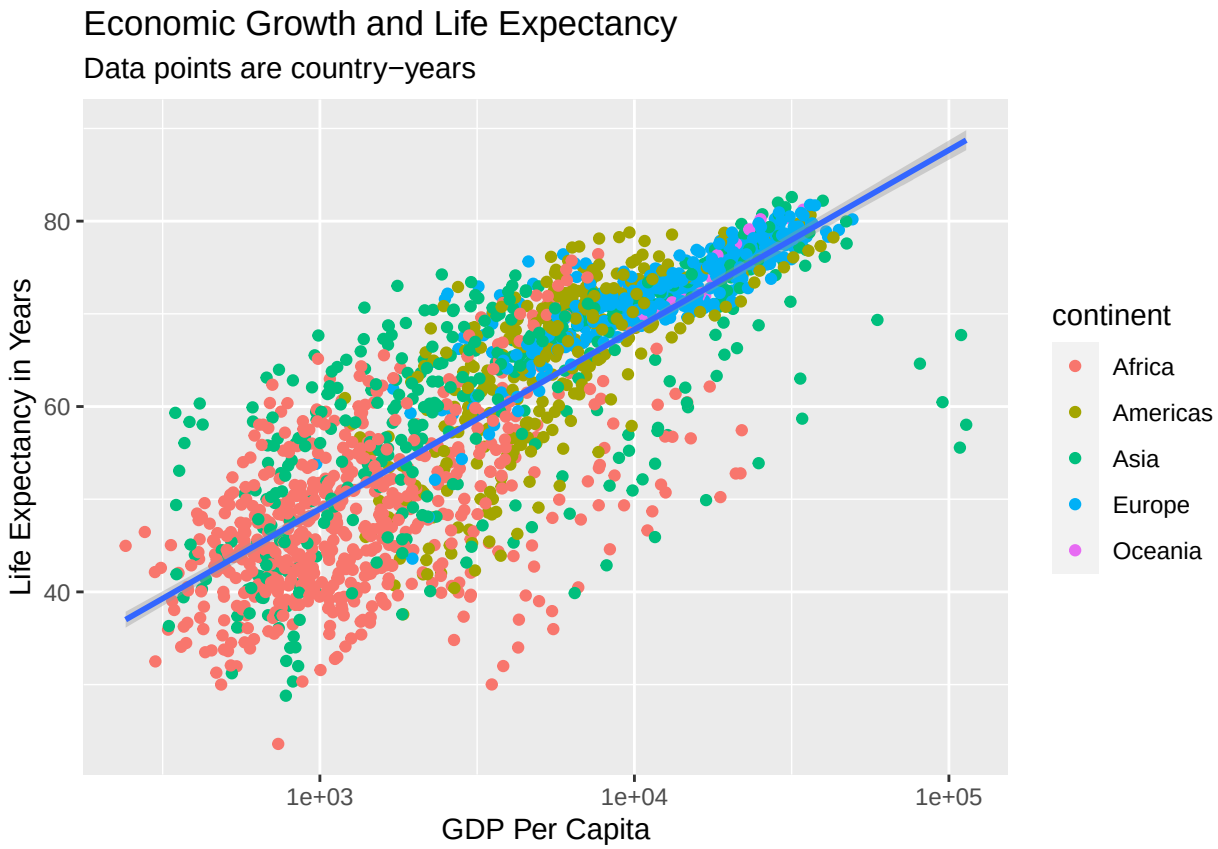
```
## `geom_smooth()` using formula 'y ~ x'
```



10. Lastly, let's save our first plot with ggplot! We will use the '`ggsave` function. R will save the plot to our working directory folder

```
myp <- ggplot(data = gapminder, mapping = aes(x = gdpPercap,
    y = lifeExp)) + geom_point(mapping = aes(colour = continent)) +
    geom_smooth(method = "lm") + scale_x_log10() + labs(x = "GDP Per Capita",
    y = "Life Expectancy in Years", title = "Economic Growth and Life Expectancy",
    subtitle = "Data points are country-years")
myp
```

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




```
ggsave("myfirstgg1.pdf")  # save as a pdf
```

```
## Saving 6.5 x 4.5 in image
## `geom_smooth()` using formula 'y ~ x'
```

A Short Practice

1. In the morning session of the workshop, we created a scatterplot. In general, to create graphs, we have been using `base R` - functions that are part of R by default. Let's try to do the same scatterplot by using `ggplot`.

℧ Practice Exercises

In this exercise, we will focus on a critical question raised by Berkman and Plutzer (2010): Who should decide what children are taught in schools? Focusing on the curricular policy on evolution, they argue that this policy outcome is affected by state-level factors (such as curriculum standards) and teacher attributes (such as training). Their data come from the National Survey of High School Biology Teachers and consist of 854 observations of high school biology teachers who were surveyed in the spring of 2007.

- The outcome of interest is the number of hours a teacher devotes to human and general evolution in his or her high school biology class (hrs_allev), and the twelve variables included in the dataset are as follows:

- `phase1`. An index of the rigor of ninth & tenth grade evolution standards in 2007 for the state the teacher works in. This variable is coded on a standardized scale with mean 0 and standard deviation 1.
- `senior_c`. An ordinal variable for the seniority of the teacher. Coded 3 for 1–2 years experience, 2 for 3–5 years, 1 for 6–10 years, 0 for 11–20 years, and 1 for 21+ years.
- `ph_senior`. An interaction between standards and seniority.
- `notest_p`. An indicator variable coded 1 if the teacher reports that the state does not have an assessment test for high school biology, 0 if the state does have such a test.

`ph_notest_p`. An interaction between standards and no state test.

- `female`.An indicator variable coded 1 if the teacher is female, 0 if male. Missing values are coded 9.
- `biocred3`: An ordinal variable for how many biology credit hours the teacher has (both graduate and undergraduate). Coded 0 for 24 hours or less, 1 for 25– 40 hours, and 2 for 40+ hours.
- `degr3`: The number of science degrees the teacher holds, from 0 to 2.
- `evol_course`: An indicator variable coded 1 if the instructor took a specific college- level course on evolution, 0 otherwise.
- `certified`: An indicator coded 1 if the teacher has normal state certification, 0 otherwise.
- `idsci_trans`: A composite measure, ranging from 0 to 1, of the degree to which the teacher thinks of him or herself as a scientist.
- `confident`: Self-rated expertise on evolutionary theory. Coded 1 for "less" than many other teachers, 0 for "typical" of most teachers, 1 for "very good" compared to most high school biology teachers, and 2 for "exceptional" and on par with college-level instructors.

- First, let's read in the data, and make sure we loaded it in properly.
- Checking the first few observations with `head()`

```
##   st_fip female hrs_allev evol_course     phase1 senior_c  ph_senior notest_p
## 1      2      0         8           0  0.2010949        1  0.2010949        1
## 2      2      0         3           0  0.2010949        0  0.0000000        1
## 3      2      0        26           1  0.2010949        0  0.0000000        1
## 4      2      0        44           1  0.2010949        1  0.2010949        1
## 5      1      1         8           0 -1.3663943        1 -1.3663943        0
## 6      1      1         4           0 -1.3663943        0  0.0000000        0
##   ph_notest_p idsci_trans biocred3 certified degr3 confident
## 1   0.2010949   0.9639460        2         0     2         0
```

```
## 2   0.2010949   0.5422196        2         1     2         0
## 3   0.2010949   0.9639460        2         0     0         2
## 4   0.2010949   0.1710664        1         1     0         2
## 5   0.0000000   0.6852229        2         0     0         1
## 6   0.0000000   0.3391082        2         1     2         0
```

- Checking the dimension of the data frame

```
## [1] 854  14
```

- A quick view of descriptive statistics

```
##      st_fip       female         hrs_allev      evol_course
##  Min.   : 1   Min.   :0.0000   Min.   : 0.00   Min.   :0.0000
##  1st Qu.:18   1st Qu.:0.0000   1st Qu.: 8.00   1st Qu.:0.0000
##  Median :34   Median :1.0000   Median :12.00   Median :0.0000
##  Mean   :31   Mean   :0.6475   Mean   :13.94   Mean   :0.4368
##  3rd Qu.:42   3rd Qu.:1.0000   3rd Qu.:19.50   3rd Qu.:1.0000
##  Max.   :55   Max.   :9.0000   Max.   :44.00   Max.   :1.0000
##      phase1            senior_c          ph_senior            notest_p
##  Min.   :-2.48770   Min.   :-3.00000   Min.   :-4.385987   Min.   :0.0000
##  1st Qu.:-0.43064   1st Qu.:-1.00000   1st Qu.:-0.252031   1st Qu.:0.0000
##  Median : 0.12602   Median : 0.00000   Median : 0.000000   Median :0.0000
##  Mean   :-0.05133   Mean   :-0.02693   Mean   : 0.008798   Mean   :0.2342
##  3rd Qu.: 0.79572   3rd Qu.: 1.00000   3rd Qu.: 0.126016   3rd Qu.:0.0000
##  Max.   : 1.46200   Max.   : 1.00000   Max.   : 7.463101   Max.   :1.0000
##   ph_notest_p        idsci_trans        biocred3       certified
##  Min.   :-2.48770   Min.   :0.0000   Min.   :0.000   Min.   :0.0000
##  1st Qu.: 0.00000   1st Qu.:0.4344   1st Qu.:1.000   1st Qu.:1.0000
##  Median : 0.00000   Median :0.5641   Median :2.000   Median :1.0000
##  Mean   :-0.08567   Mean   :0.6083   Mean   :1.582   Mean   :0.7927
##  3rd Qu.: 0.00000   3rd Qu.:0.8192   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   : 0.96358   Max.   :1.0000   Max.   :2.000   Max.   :1.0000
##      degr3          confident
##  Min.   :0.0000   Min.   :-1.0000
##  1st Qu.:0.0000   1st Qu.: 0.0000
##  Median :1.0000   Median : 1.0000
##  Mean   :0.8735   Mean   : 0.7283
##  3rd Qu.:1.0000   3rd Qu.: 1.0000
##  Max.   :2.0000   Max.   : 2.0000
```

- The value of 9 in the female variable represents missing values, we need to fix this by recoding missing observations of sex as `NA`
- Let's look at the resived view of summary

```
##      st_fip       female         hrs_allev      evol_course
##  Min.   : 1   Min.   :0.0000   Min.   : 0.00   Min.   :0.0000
##  1st Qu.:18   1st Qu.:0.0000   1st Qu.: 8.00   1st Qu.:0.0000
##  Median :34   Median :1.0000   Median :12.00   Median :0.0000
##  Mean   :31   Mean   :0.5184   Mean   :13.94   Mean   :0.4368
##  3rd Qu.:42   3rd Qu.:1.0000   3rd Qu.:19.50   3rd Qu.:1.0000
##  Max.   :55   Max.   :1.0000   Max.   :44.00   Max.   :1.0000
```

```
##               NA's   :13
##      phase1            senior_c          ph_senior            notest_p
##  Min.   :-2.48770   Min.   :-3.00000   Min.   :-4.385987   Min.   :0.0000
##  1st Qu.:-0.43064   1st Qu.:-1.00000   1st Qu.:-0.252031   1st Qu.:0.0000
##  Median : 0.12602   Median : 0.00000   Median : 0.000000   Median :0.0000
##  Mean   :-0.05133   Mean   :-0.02693   Mean   : 0.008798   Mean   :0.2342
##  3rd Qu.: 0.79572   3rd Qu.: 1.00000   3rd Qu.: 0.126016   3rd Qu.:0.0000
##  Max.   : 1.46200   Max.   : 1.00000   Max.   : 7.463101   Max.   :1.0000
##
##   ph_notest_p        idsci_trans        biocred3       certified
##  Min.   :-2.48770   Min.   :0.0000   Min.   :0.000   Min.   :0.0000
##  1st Qu.: 0.00000   1st Qu.:0.4344   1st Qu.:1.000   1st Qu.:1.0000
##  Median : 0.00000   Median :0.5641   Median :2.000   Median :1.0000
##  Mean   :-0.08567   Mean   :0.6083   Mean   :1.582   Mean   :0.7927
##  3rd Qu.: 0.00000   3rd Qu.:0.8192   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   : 0.96358   Max.   :1.0000   Max.   :2.000   Max.   :1.0000
##
##      degr3          confident
##  Min.   :0.0000   Min.   :-1.0000
##  1st Qu.:0.0000   1st Qu.: 0.0000
##  Median :1.0000   Median : 1.0000
##  Mean   :0.8735   Mean   : 0.7283
##  3rd Qu.:1.0000   3rd Qu.: 1.0000
##  Max.   :2.0000   Max.   : 2.0000
##
```

- Another way to look at NAs

```
##
## FALSE  TRUE
## 11943    13
```

- Let's remove the NAs. Using the `subset()` function we will remove the missing values from our data frame. `!is.na` is telling R to only keep the observations that are not missing.
- In their study, the researcheres estimate a linear regression model of hours spent teaching evolution as follows:
- There are two special terms in this regression model: `phase1 * senior_c` and `phase1 * notest_p`. These are `interaction terms` that allow to test conditional effects of a variable, e.g. the effect of conditional on...
- The `data` option of `lm()` function is very handy because it allows us to call variables from the same dataset without having to refer to the dataset's name with each variable.

```
##
## Call:
## lm(formula = hrs_allev ~ phase1 * senior_c + phase1 * notest_p +
##     female + biocred3 + degr3 + evol_course + certified + idsci_trans +
##     confident, data = evoldat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.378  -6.148  -1.314   4.744  32.148
```

```
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      10.2313     1.1905   8.594  < 2e-16 ***
## phase1            0.6285     0.3331   1.886   0.0596 .
## senior_c         -0.5813     0.3130  -1.857   0.0636 .
## notest_p          0.4852     0.7222   0.672   0.5019
## female           -1.3546     0.6016  -2.252   0.0246 *
## biocred3          0.5559     0.5072   1.096   0.2734
## degr3            -0.4003     0.3922  -1.021   0.3077
## evol_course       2.5108     0.6300   3.985 7.33e-05 ***
## certified        -0.4446     0.7212  -0.617   0.5377
## idsci_trans       1.8549     1.1255   1.648   0.0997 .
## confident         2.6262     0.4501   5.835 7.71e-09 ***
## phase1:senior_c  -0.5112     0.2717  -1.881   0.0603 .
## phase1:notest_p  -0.5362     0.6233  -0.860   0.3899
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.397 on 828 degrees of freedom
## Multiple R-squared:  0.1226, Adjusted R-squared:  0.1099
## F-statistic: 9.641 on 12 and 828 DF,  p-value: < 2.2e-16
```