R Notebook

The following is your first chunk to start with. Remember, you can add chunks using the menu above (Insert -> R) or using the keyboard shortcut Ctrl+Alt+I. A good practice is to use different code chunks to answer different questions. You can delete this comment if you like.

Other useful keyboard shortcuts include Alt- for the assignment operator, and Ctrl+Shift+M for the pipe operator. You can delete these reminders if you don't want them in your report.

```
#setwd("") #Don't forget to set your working directory before you start!

library("tidyverse")

## -- Attaching packages --------------------------------------- tidyverse 1.
3.0 --

## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0

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

library("tidymodels")

## Registered S3 method overwritten by 'xts':
##   method     from
##   as.zoo.xts zoo

## -- Attaching packages -------------------------------------- tidymodels 0.
0.3 --

## v broom     0.5.3     v recipes   0.1.9
## v dials     0.0.4     v rsample   0.0.5
## v infer     0.5.1     v yardstick 0.0.4
## v parsnip   0.0.5

## -- Conflicts ----------------------------------------- tidymodels_conflict
s() --
## x scales::discard()   masks purrr::discard()
## x dplyr::filter()     masks stats::filter()
## x recipes::fixed()    masks stringr::fixed()
## x dplyr::lag()        masks stats::lag()
## x dials::margin()     masks ggplot2::margin()
```

```
## x yardstick::spec()   masks readr::spec()
## x recipes::step()     masks stats::step()
## x recipes::yj_trans() masks scales::yj_trans()
```

```r
library("plotly")
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```r
library("skimr")
```

```r
library(gapminder)
dfGap <- gapminder
```

Explore the data Use the skim function on the dfGap dataframe to get summary statistics in a nice format. I suggest you use the widest screen possible for the best reading.

```r
#3a
#dfGap
skim(dfGap)
```

Data summary

Name	dfGap
Number of rows	1704
Number of columns	6
_______________________	
Column type frequency:	
factor	2
numeric	4
________________________	
Group variables	None

Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts

country	0	1	FALSE	142	Afg: 12, Alb: 12, Alg: 12, Ang: 12
continent	0	1	FALSE	5	Afr: 624, Asi: 396, Eur: 360, Ame: 300

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
year	0	1	1979.50	17.27	1952.00	1965.75	1979.50	1993.25	2007.0	▇▅▅▅▅▇
lifeExp	0	1	59.47	12.92	23.60	48.20	60.71	70.85	82.6	▁▆▇
pop	0	1	29601212.32	106157896.74	60011.00	2793664.00	7023595.50	19585221.75	1318683096.0	▇▁▁▁▁
gdpPercap	0	1	7215.33	9857.45	241.17	1202.06	3531.85	9325.46	113523.1	▇▁▁▁▁

3)b)Filter the dfGap	for the yea	r 2007 and sort	it in descendi	ng order of lif	e expectanc	y. Don't forg	et to use pip	es!	
What are the nam	es of the co	untries with a l	ife expectancy	over 81?					

```
#3b

##Filter dfGap for the year 2007 and sort it in descending order of life expe
ctancy. Don't forget to use pipes!
##What are the names of the countries with a life expectancy over 81?



dfGap3b <- dfGap %>%
  filter(year==2007)%>%
  arrange(desc(lifeExp)) %>%
  filter(lifeExp>81)
dfGap3b

## # A tibble: 5 x 6
##   country          continent  year lifeExp       pop gdpPercap
##   <fct>            <fct>     <int>   <dbl>     <int>     <dbl>
```

```
## 1 Japan            Asia       2007    82.6 127467972    31656.
## 2 Hong Kong, China Asia       2007    82.2   6980412    39725.
## 3 Iceland          Europe     2007    81.8    301931    36181.
## 4 Switzerland      Europe     2007    81.7   7554661    37506.
## 5 Australia        Oceania    2007    81.2  20434176    34435.
```

```
#3b)i)
#What are the names of the countries with a life expectancy over 81?

dfGap10 <- dfGap3b %>%
  distinct(country)
dfGap10
```

```
## # A tibble: 5 x 1
##   country
##   <fct>
## 1 Japan
## 2 Hong Kong, China
## 3 Iceland
## 4 Switzerland
## 5 Australia
```

c)Add a calculated column totalGDP to dfGap showing the total GDP per country, filter the dataframe for 2007, and sort in descending order for totalGDP. If you like, save the new dataframe as a new one for repeated use. i)What are some names of the countries with the top levels of total GDP?

ii)Which ones of these countries overlap with the countries from 3-b? iii)What if you selected only the two columns country and gdpPercap and sorted the dataframe in descending order for gdpPercap? Do you observe more of an overlap now? What do you infer from this difference?

```
#3)c)

dfGap3c <- dfGap %>%
  #group_by(country)%>%
  filter(year==2007)%>%
  mutate(totalGdp=pop*gdpPercap)%>%
  arrange(desc(totalGdp))

dfGap3c
```

```
## # A tibble: 142 x 7
##    country        continent  year lifeExp        pop gdpPercap totalGdp
##    <fct>          <fct>     <int>   <dbl>      <int>     <dbl>    <dbl>
##  1 United States  Americas   2007    78.2  301139947    42952.  1.29e13
##  2 China          Asia       2007    73.0 1318683096     4959.  6.54e12
##  3 Japan          Asia       2007    82.6  127467972    31656.  4.04e12
##  4 India          Asia       2007    64.7 1110396331     2452.  2.72e12
##  5 Germany        Europe     2007    79.4   82400996    32170.  2.65e12
```

```
## 6 United Kingdom Europe      2007    79.4   60776238    33203.  2.02e12
## 7 France         Europe      2007    80.7   61083916    30470.  1.86e12
## 8 Brazil         Americas    2007    72.4  190010647     9066.  1.72e12
## 9 Italy          Europe      2007    80.5   58147733    28570.  1.66e12
## 10 Mexico         Americas    2007    76.2  108700891    11978.  1.30e12
## # ... with 132 more rows
```

i)What are some names of the countries with the top levels of total GDP?

```
#3c)i)
dfGap100 <- dfGap3c %>%
distinct(country)
dfGap100
```

```
## # A tibble: 142 x 1
##    country
##    <fct>
##  1 United States
##  2 China
##  3 Japan
##  4 India
##  5 Germany
##  6 United Kingdom
##  7 France
##  8 Brazil
##  9 Italy
## 10 Mexico
## # ... with 132 more rows
```

iii)What if you selected only the two columns country and gdpPercap and sorted the dataframe in descending order for gdpPercap? Do you observe more of an overlap now? What do you infer from this difference?

```
#3)c)iii)

 #Countries from descending order of gdpPercap

dfGap4 <- dfGap %>%
  #group_by(country)%>%
  filter(year==2007)%>%
  select(country,gdpPercap)%>%
  arrange(desc(gdpPercap))
dfGap4
```

```
## # A tibble: 142 x 2
##    country          gdpPercap
##    <fct>                <dbl>
##  1 Norway              49357.
##  2 Kuwait              47307.
```

```
##   3 Singapore            47143.
##   4 United States        42952.
##   5 Ireland              40676.
##   6 Hong Kong, China     39725.
##   7 Switzerland          37506.
##   8 Netherlands          36798.
##   9 Canada               36319.
## 10 Iceland              36181.
## # ... with 132 more rows
```

3)d)Filter dfGap for 2007, group it by continent, and then calculate the median life expectancy and median total GDP (so you need to have totalGDP already). Remember, you will pipe the filtered and grouped dataframe into summarize() to get the medians. Then, sort it in descending order for the median life expectancy. Before you sort it, don't forget to use ungroup() to ungroup. i)What continent has the highest median of life expectancy? ii)Does it seem to be correlated with the median total GDP?

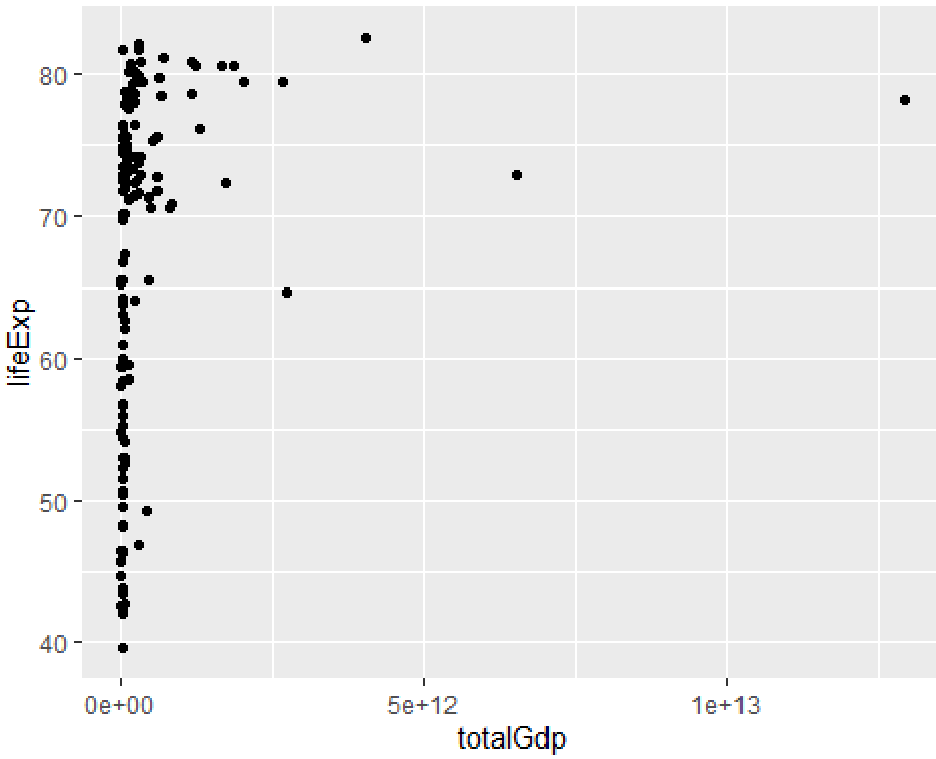
```
#3)d)

dfGap3d <- dfGap %>%
  group_by(continent)%>%
  filter(year==2007)%>%
  mutate(totalGdp=pop*gdpPercap)%>%
  summarize(medianlife=median(lifeExp, na.rm=TRUE),mediantotalGdp=median(tota
lGdp, na.rm=TRUE))%>%
  ungroup()# %>%



#dfGap3d

  dfGap3d%>%

      arrange(desc(medianlife))
```

```
## # A tibble: 5 x 3
##   continent medianlife mediantotalGdp
##   <fct>          <dbl>          <dbl>
## 1 Oceania         80.7   403657044512.
## 2 Europe          78.6   230988745548.
## 3 Americas        72.9    65203833292.
## 4 Asia            72.4   164029908950.
## 5 Africa          52.9    13755919229.
```

4) Visualize the data a)Now that you have explored the relationship between life expectancy and totalGDP in a table format, let's also visualize it to see a bigger picture. i)Create a scatter plot to understand the relationship between life expectancy (y-axis) and totalGDP (x-axis) in 2007. Does this plot help?

```
#dfGap
#ggplot(data=dfGap3d)+
```

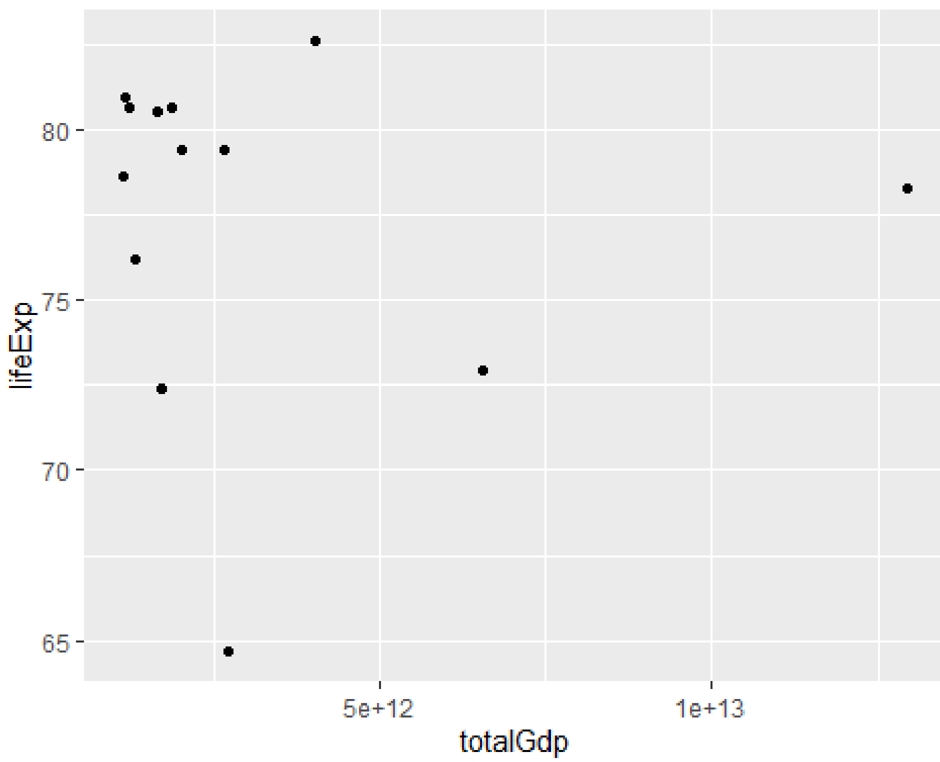
```r
#  geom_point(mapping=aes(x=lifeExp,y=totalGDP))

#4)a)i)

dfGap4<-dfGap%>%
  filter(year==2007)%>%
  mutate(totalGdp=pop*gdpPercap)

dfGap4 %>%
  ggplot(aes(y = lifeExp, x = totalGdp)) + geom_point()
```



ii)Copy the same code, but this time also filter for countries with a totalGDP of over a billion (use the scientific notation 1e+12). What about now?

```r
#4a)ii)
  dfGap5<-dfGap%>%
  filter(year==2007)%>%
  mutate(totalGdp=pop*gdpPercap)%>%
  filter(totalGdp>1e+12)
dfGap5

## # A tibble: 13 x 7
##    country        continent  year lifeExp        pop gdpPercap totalGdp
##    <fct>          <fct>     <int>   <dbl>      <int>     <dbl>    <dbl>
##  1 Brazil         Americas   2007    72.4  190010647     9066.  1.72e12
##  2 Canada         Americas   2007    80.7   33390141    36319.  1.21e12
##  3 China          Asia       2007    73.0 1318683096     4959.  6.54e12
```

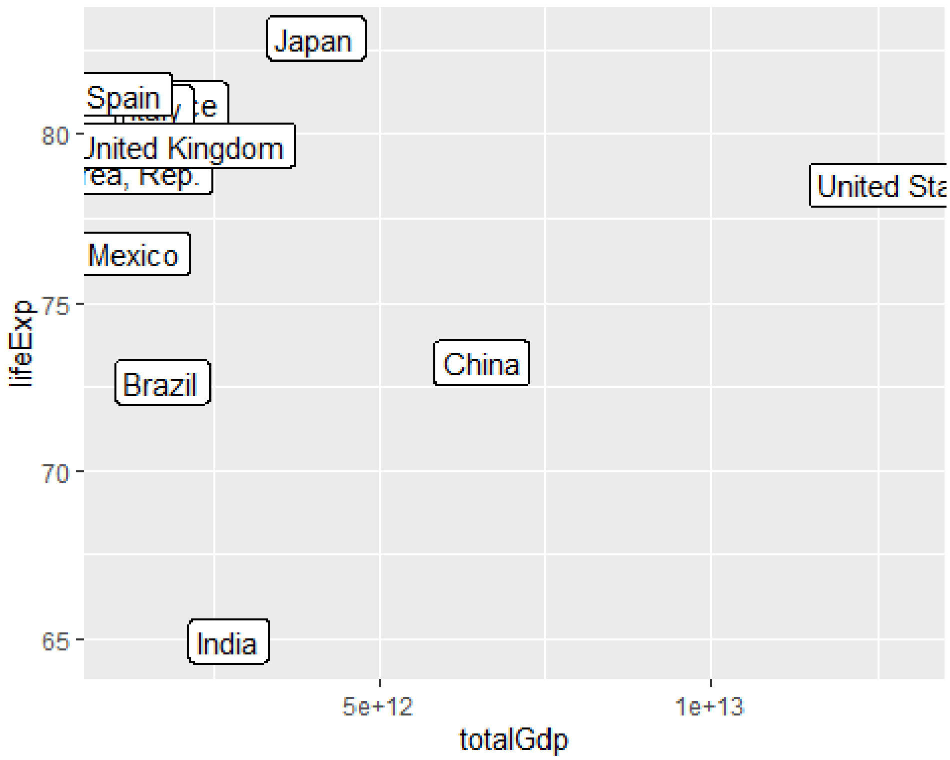
```
## 4 France          Europe     2007    80.7   61083916    30470.  1.86e12
## 5 Germany         Europe     2007    79.4   82400996    32170.  2.65e12
## 6 India           Asia       2007    64.7 1110396331    2452.   2.72e12
## 7 Italy           Europe     2007    80.5   58147733    28570.  1.66e12
## 8 Japan           Asia       2007    82.6  127467972    31656.  4.04e12
## 9 Korea, Rep.     Asia       2007    78.6   49044790    23348.  1.15e12
## 10 Mexico         Americas   2007    76.2  108700891    11978.  1.30e12
## 11 Spain          Europe     2007    80.9   40448191    28821.  1.17e12
## 12 United Kingdom Europe     2007    79.4   60776238    33203.  2.02e12
## 13 United States  Americas   2007    78.2  301139947    42952.  1.29e13

dfGap6 <- dfGap5 %>%
  ggplot(aes(y = lifeExp, x = totalGdp))+ geom_point()
#Plot for countries with over 1 billion Total GDP
dfGap6
```



iii)Copy the same code, and add labels this time. Do you see a cluster now? What are the names of the countries that are outside of the cluster?

```
#4a)iii)



library(ggplot2)
```

```r
# 1/ add text with geom_text, use nudge to nudge the text
ggplot(dfGap5, aes(x=totalGdp, y=lifeExp,label=country)) +
  geom_point() + # Show dots
  geom_label(
    aes(label=country),
    #label=rownames(data),
    nudge_x = 0.25, nudge_y = 0.25,

  )
```

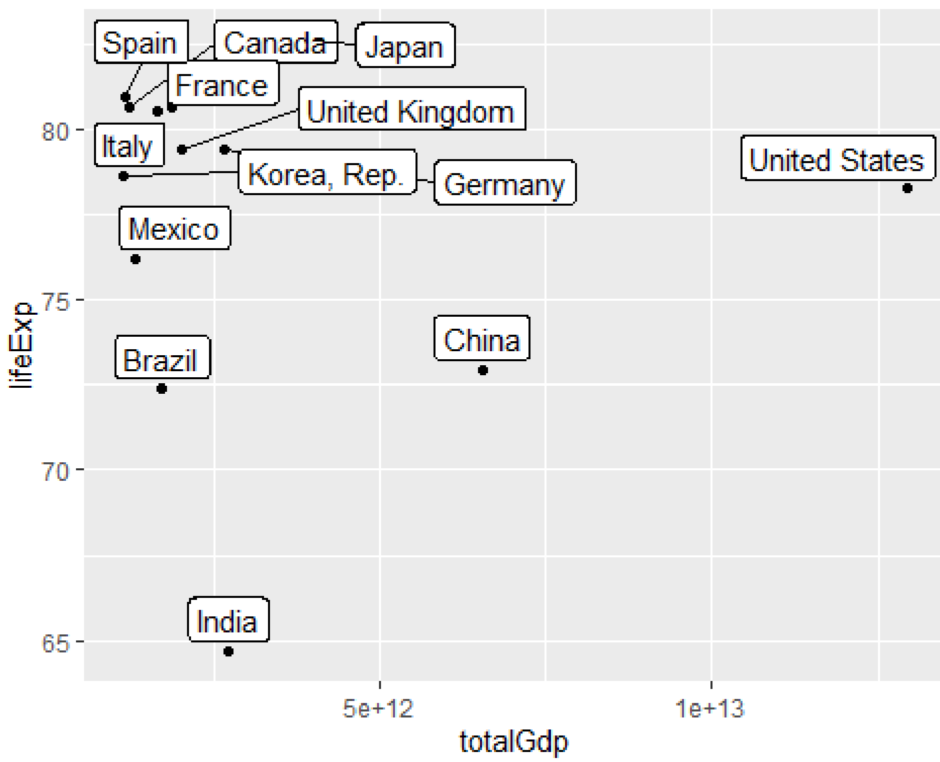



```r
  #check_overlap = T)
```

iv)Here is a pro tip. The labels you used in (iii) overlap and hide the points. This causes poor visibility. Install and load the ggrepel library. After that, copy the same code and use geom_label_repel() function instead of geom_label(). Does it look better now? Describe what has changed.

```r
#4a)iv)

library(ggrepel)

ggplot(dfGap5, aes(x=totalGdp, y=lifeExp)) +
  geom_point() +
  geom_label_repel(
```

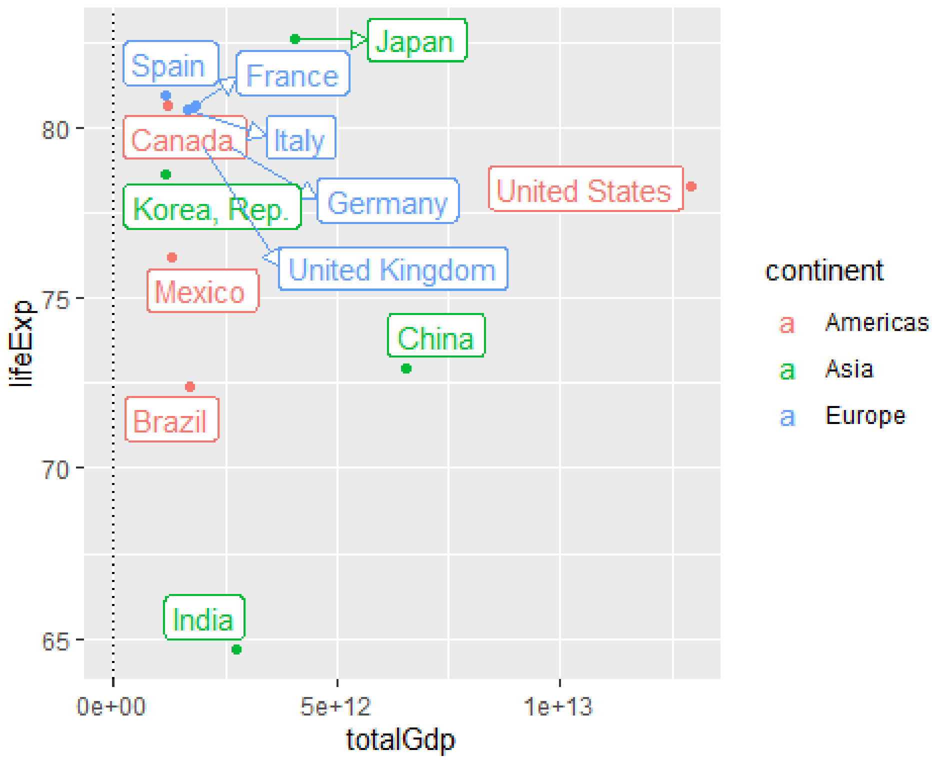
```
    nudge_x = 0.25, nudge_y = 0.25,

  aes(label=country)
  )
```



v)Copy the same code. This time, add a color for the continent. What are the continents that are missing from your visual? Why do you think so?

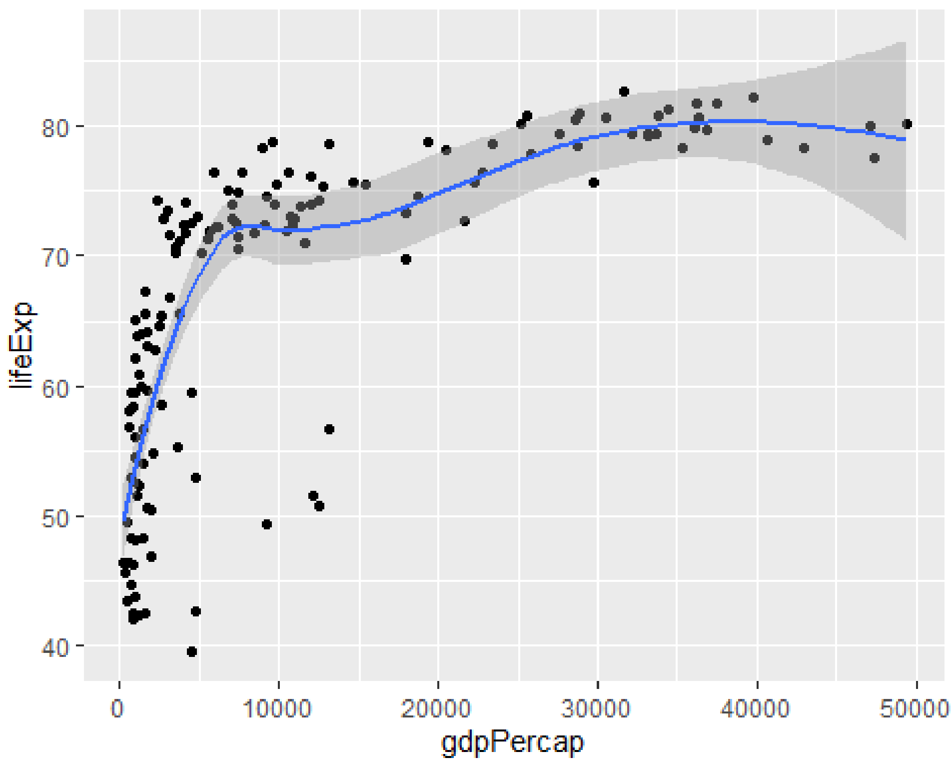
```
#4a)v)

set.seed(42)

# All labels should be to the right of 3.
x_limits <- c(3, NA)

#ggplot(dat, aes(wt, mpg, label = car, color = factor(cyl)))

ggplot(dfGap5, aes(x=totalGdp, y=lifeExp,label=country,color=factor(continent
)))+
  geom_vline(xintercept = x_limits, linetype = 3) +
  geom_point() +
  geom_label_repel(
    arrow = arrow(length = unit(0.03, "npc"), type = "closed", ends = "first"
),
    force = 10,
    xlim  = x_limits
```

```
  ) +
  scale_color_discrete(name = "continent")

## Warning: Removed 1 rows containing missing values (geom_vline).
```



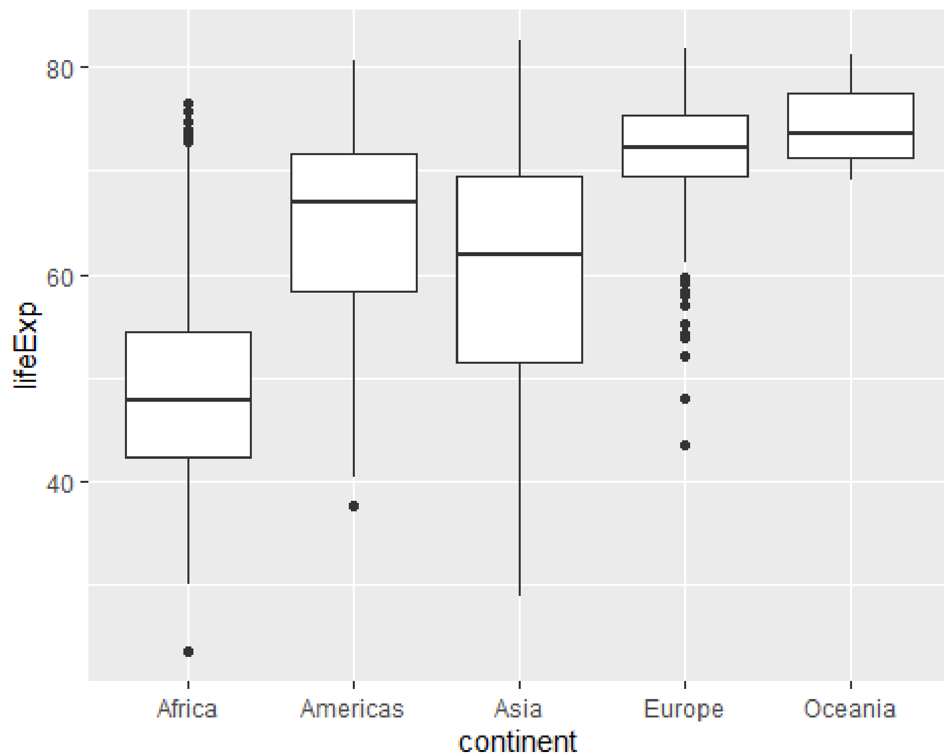
Q4)b)You have an idea about the relationship between life expectancy and totalGDP even though you have not tested it statistically. Now, let's examine a more realistic relationship between life expectancy and gdpPercap (GDP per capita). Plot life expectancy (y-axis) against gdpPercap (x-axis) for 2007, add a smoothed line (no need to define any parameters, use the defaults). What do you observe about the overall relationship? Don't use any labels, just focus on the aggregate.

```
#4)b)

dfGap101<-dfGap%>%
  filter(year==2007)

dfGap102 <- dfGap101 %>%
  ggplot(aes(y = lifeExp, x = gdpPercap))+ geom_point()

 dfGap102 + geom_smooth(method = "loess")
```



Q4)c)Now let's find out the variations in life expectancy across different continents. Create box plots for each continent (in the same plot) and add a title this time.

```
#4)c)

boxPlotsForAll <- ggplot(dfGap, aes(x=continent, y=lifeExp)) + geom_boxplot()
boxPlotsForAll
```

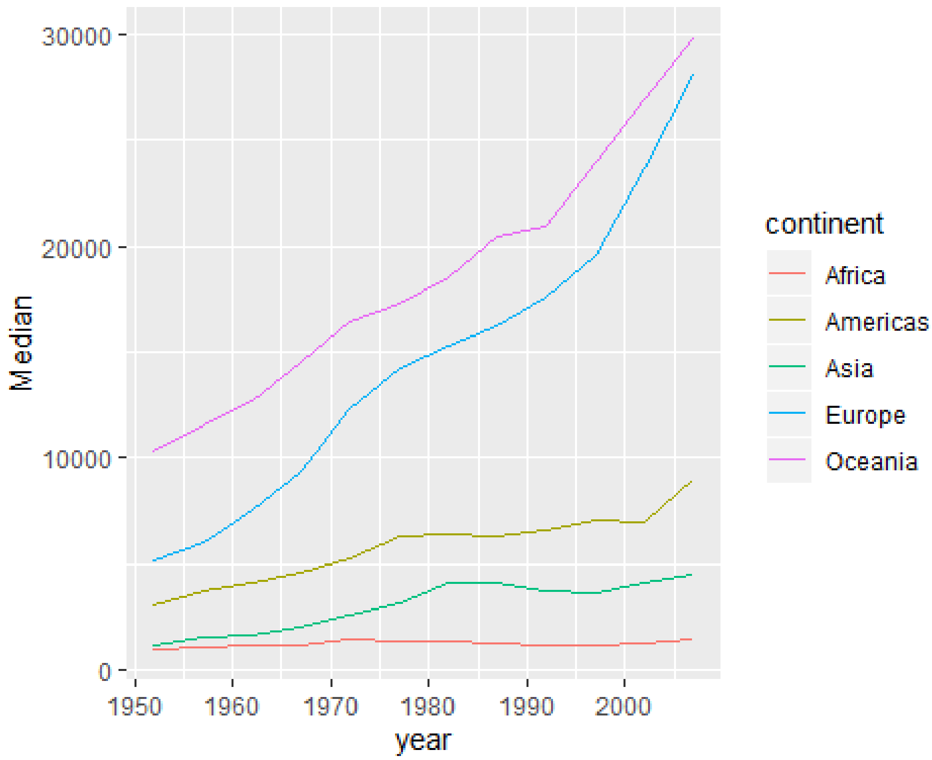



```
boxPlotsForAll <- ggplotly(boxPlotsForAll)

boxPlotsForAll
```

Q4)d)Finally, it is time to create a more advanced (and likely more helpful) plot. Create a line plot to show how median GDP per capita by continent changes over time. [Hint: For the continents, use the color parameter]. Describe what you observe. What continents have a clearer trend than others? Why do you think so?

```
#4)d)i)
df1 <- dfGap%>%
group_by(continent,year)%>%
mutate(Median=median(gdpPercap))%>%
distinct(continent,.keep_all=TRUE)
df1

## # A tibble: 60 x 7
## # Groups:   continent, year [60]
##    country     continent  year lifeExp      pop gdpPercap Median
##    <fct>       <fct>     <int>   <dbl>    <int>     <dbl>  <dbl>
##  1 Afghanistan Asia       1952    28.8  8425333      779.  1207.
##  2 Afghanistan Asia       1957    30.3  9240934      821.  1548.
##  3 Afghanistan Asia       1962    32.0 10267083      853.  1650.
##  4 Afghanistan Asia       1967    34.0 11537966      836.  2029.
##  5 Afghanistan Asia       1972    36.1 13079460      740.  2571.
##  6 Afghanistan Asia       1977    38.4 14880372      786.  3195.
##  7 Afghanistan Asia       1982    39.9 12881816      978.  4107.
```

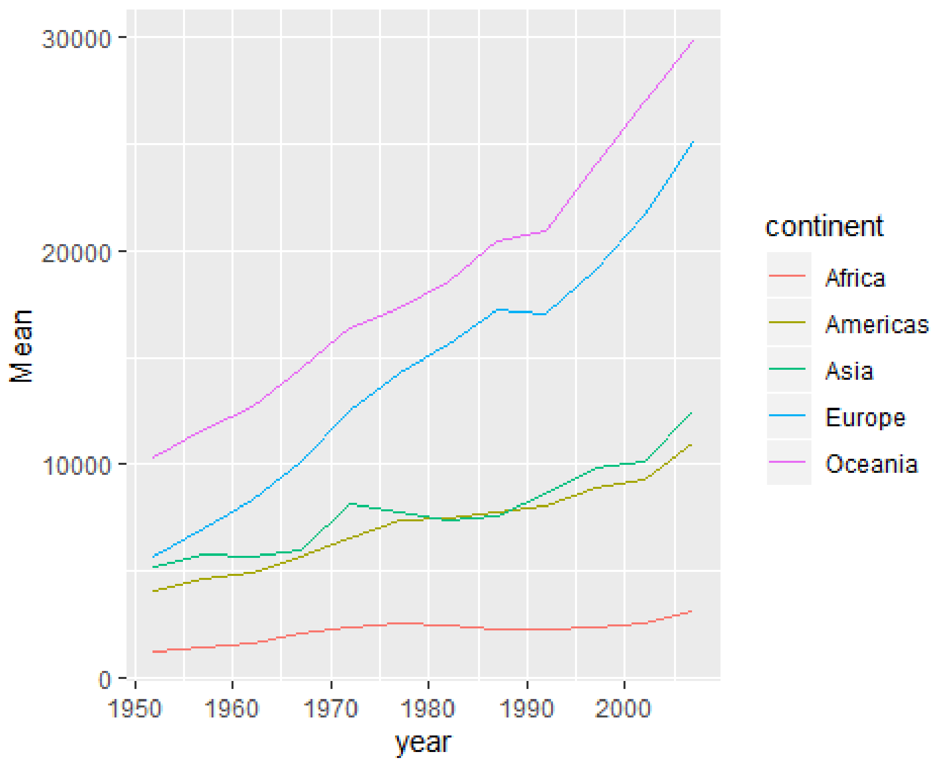
```
## 8 Afghanistan Asia      1987    40.8 13867957      852.  4106.
## 9 Afghanistan Asia      1992    41.7 16317921      649.  3726.
## 10 Afghanistan Asia     1997    41.8 22227415      635.  3645.
## # ... with 50 more rows

df2<-ggplot(df1, aes(x=year,y=Median, color=continent)) + geom_line()
df2
```




```
ggplotly(df2)
```

Q4)d)ii)Change the summary metric from median to mean. What has changed? Why do you think so?

```
#4)d)ii)
df3 <- dfGap%>%
group_by(continent,year)%>%
mutate(Mean=mean(gdpPercap))%>%
distinct(continent,.keep_all=TRUE)


df4<-ggplot(df3, aes(x=year,y=Mean, color=continent)) + geom_line()

df4
```




```
ggplotly(df4)
```

Q4)iii)Finally, don't you think these plots would be much more useful in plotly? Pick one and save it as gdpOverTime and call ggplotly() on it. You can now read the actual GDP values per year. What are some of the breakthrough years (steep changes) for GDP in different continents?

```
#4)d)iii)

df1 <- dfGap%>%
group_by(continent,year)%>%
mutate(Median=median(gdpPercap))%>%
distinct(continent,.keep_all=TRUE)

df1

## # A tibble: 60 x 7
## # Groups:   continent, year [60]
##    country     continent  year lifeExp      pop gdpPercap Median
##    <fct>       <fct>     <int>   <dbl>    <int>     <dbl>  <dbl>
##  1 Afghanistan Asia       1952    28.8  8425333      779.  1207.
##  2 Afghanistan Asia       1957    30.3  9240934      821.  1548.
##  3 Afghanistan Asia       1962    32.0 10267083      853.  1650.
##  4 Afghanistan Asia       1967    34.0 11537966      836.  2029.
##  5 Afghanistan Asia       1972    36.1 13079460      740.  2571.
##  6 Afghanistan Asia       1977    38.4 14880372      786.  3195.
##  7 Afghanistan Asia       1982    39.9 12881816      978.  4107.
```

```
## 8 Afghanistan Asia 1987 40.8 13867957 852. 4106.
## 9 Afghanistan Asia 1992 41.7 16317921 649. 3726.
## 10 Afghanistan Asia 1997 41.8 22227415 635. 3645.
## # ... with 50 more rows
```

```
gdpOverTime<-ggplot(df1, aes(x=year,y=Median, color=continent)) + geom_line()

ggplotly(gdpOverTime)
```