Module 4 R Exercise

In these exercises, we will use ggplot2 and plotly functions to plot some statistical plots about a data set.

Have your ggplot2 and plotly cheatsheets and documentation handy to find the right parameters for the functions.

Let's read the Gapminder data from the web resource.

In [1]:
```
library(ggplot2)
library(plotly)
library(RColorBrewer)

data <- read.csv("https://raw.githubusercontent.com/plotly/datasets/master/gapminderDataFiveYear.csv")
head(data)
```

```
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
```

A data.frame: 6 × 6

country	year	pop	continent	lifeExp	gdpPercap
<fct>	<int>	<dbl>	<fct>	<dbl>	<dbl>
Afghanistan	1952	8425333	Asia	28.801	779.4453
Afghanistan	1957	9240934	Asia	30.332	820.8530
Afghanistan	1962	10267083	Asia	31.997	853.1007
Afghanistan	1967	11537966	Asia	34.020	836.1971
Afghanistan	1972	13079460	Asia	36.088	739.9811
Afghanistan	1977	14880372	Asia	38.438	786.1134

Exercise 1: Use which() function to get a **subset** of the data between years 1951 and 1993.

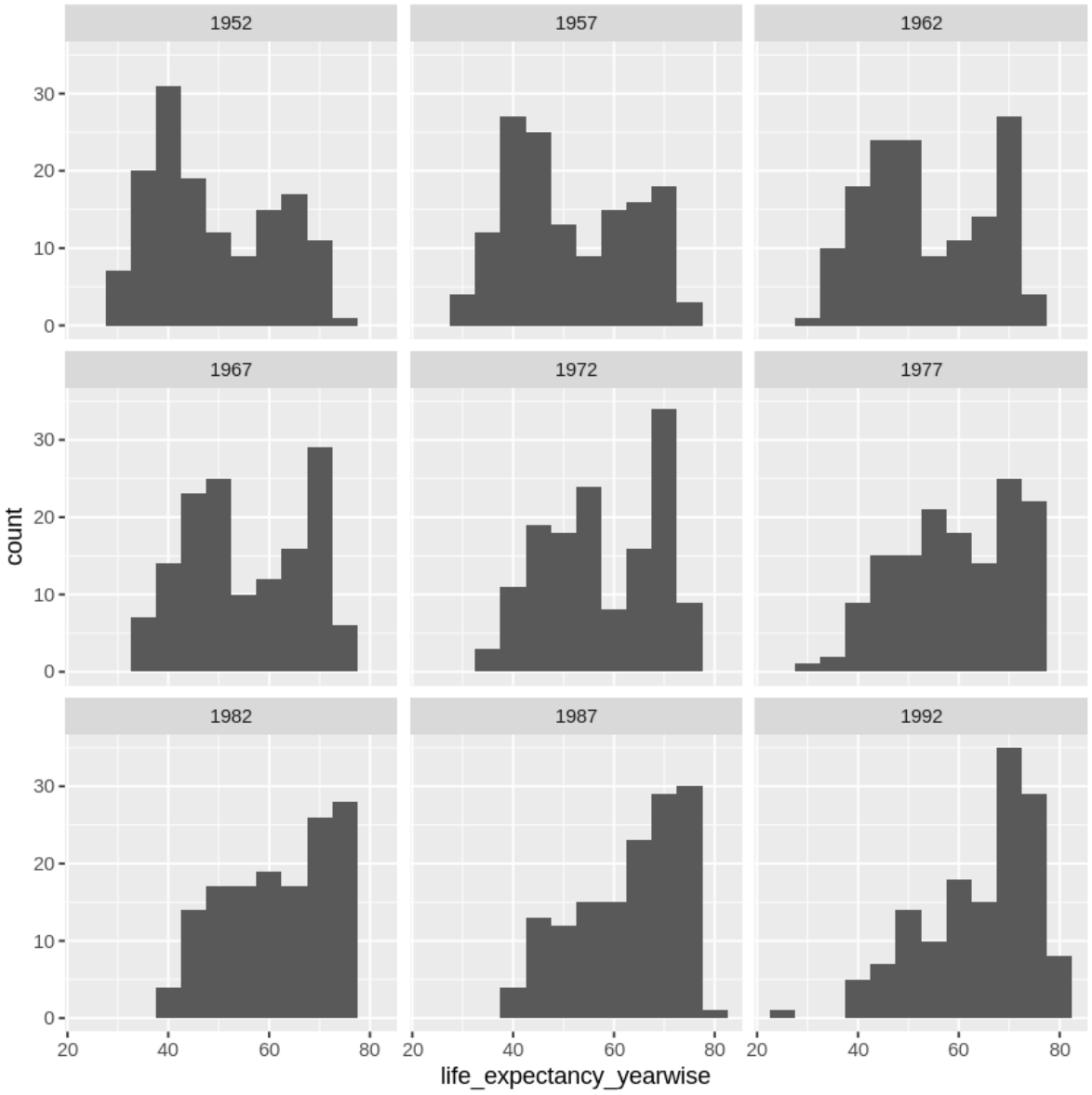
In [4]:
```
data_sub <- data[which(data$year>=1951 & data$year<=1993), ]
head(data_sub)
```

A data.frame: 6 × 6

country	year	pop	continent	lifeExp	gdpPercap
<fct>	<int>	<dbl>	<fct>	<dbl>	<dbl>
Afghanistan	1952	8425333	Asia	28.801	779.4453
Afghanistan	1957	9240934	Asia	30.332	820.8530
Afghanistan	1962	10267083	Asia	31.997	853.1007
Afghanistan	1967	11537966	Asia	34.020	836.1971
Afghanistan	1972	13079460	Asia	36.088	739.9811
Afghanistan	1977	14880372	Asia	38.438	786.1134

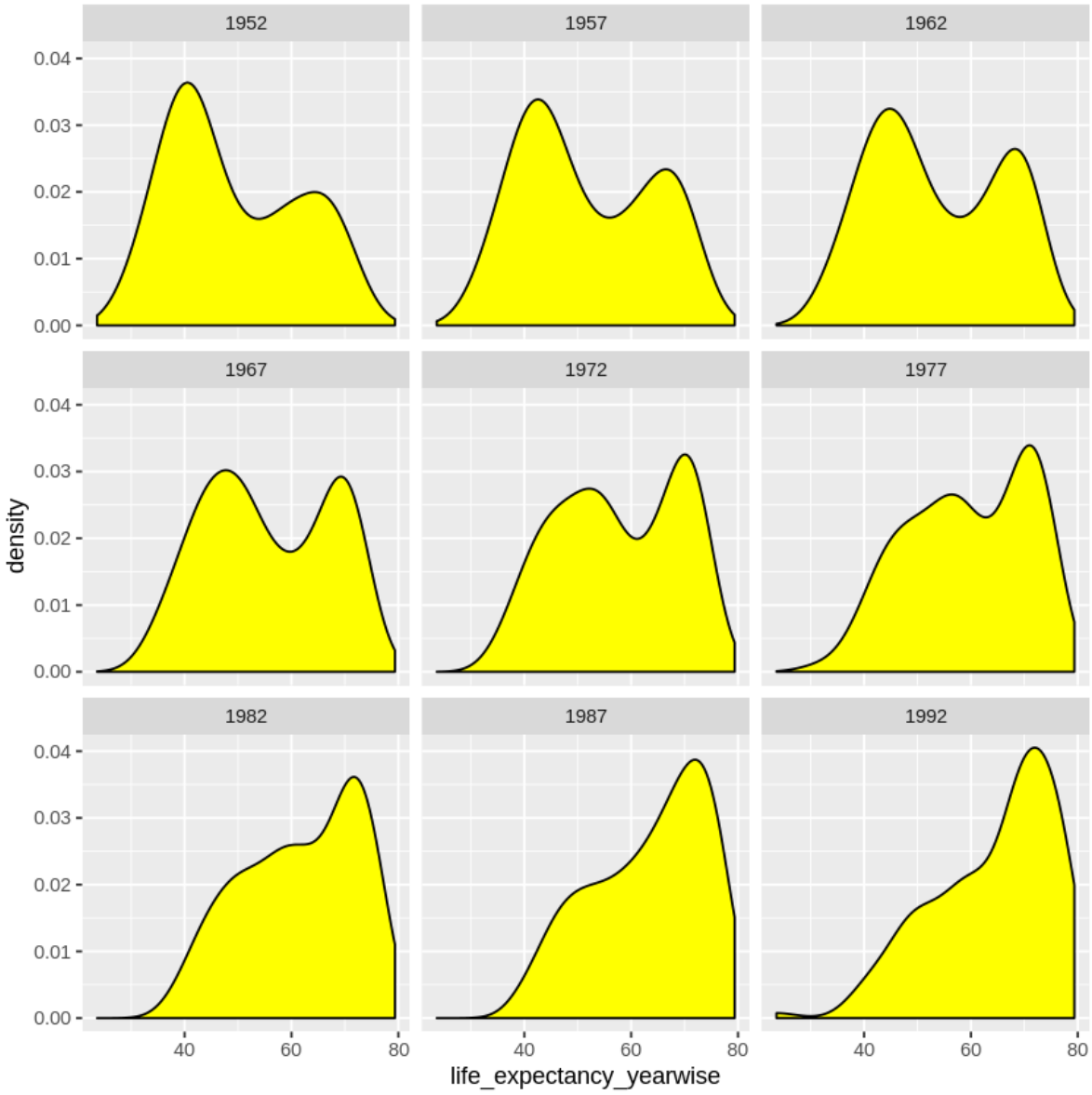
Exercise 2: Plot **small multiples of histograms** of **life expectancy for each year** for the subset. Use a binwidth of **5**, and use sensible axis labels.

In [14]:
```
ggplot(data_sub, aes(x= data_sub$lifeExp)) +
 geom_histogram(binwidth = 5,) +
  facet_wrap(~data_sub$year) + xlab("life_expectancy_yearwise")
```



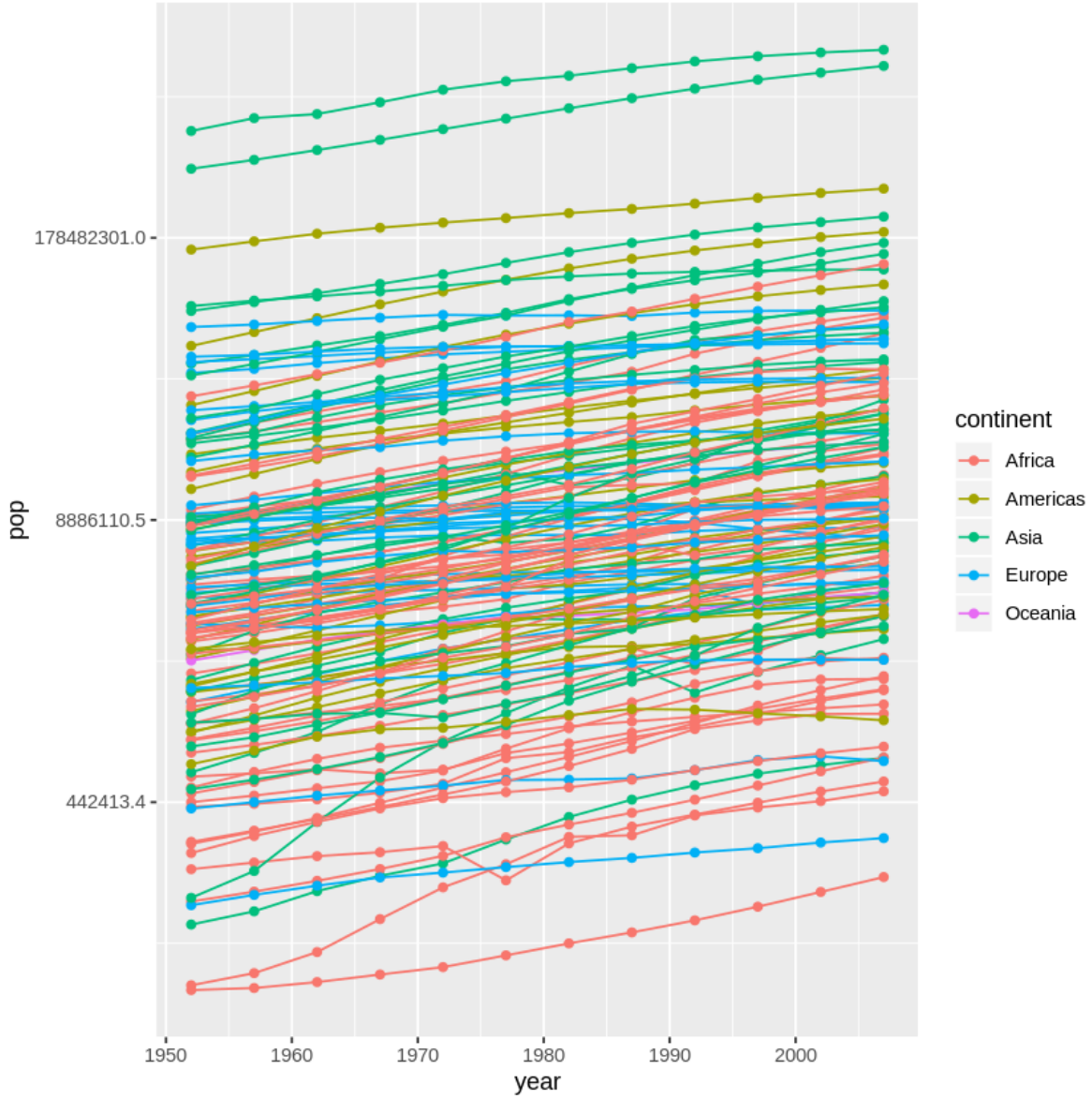
Exercise 3: Do the **same** as above, but with a density function this time. How do you interpret the change of density in years?

In [13]:
```
ggplot(data_sub, aes(x= data_sub$lifeExp)) +
 geom_density(alpha=1 ,fill= "yellow") + facet_wrap(~data_sub$year) +
xlab('life_expectancy_yearwise')
```



Exercise 4: Create a line plot (use **both** geom_line and geom_point) to plot year versus population for the **whole** data set. Use a logarithmic scale in **y axis** and **group by country**, **color by continent**. Can you see any pattern?

In [17]:
```
ggplot(data, aes(x= year,y=pop, group=country)) +
       geom_line(aes(color = continent)) + geom_point(aes(color= conti
nent)) + scale_y_continuous(trans = 'log')

#No, I cannot see the pattern.
```



Aggregate data: The above plot is too crowded to see anything. Let's aggregate data by continent and year so that we can have meaningful data to plot. The following code creates a new data frame by computing the sums of population for years and continents.

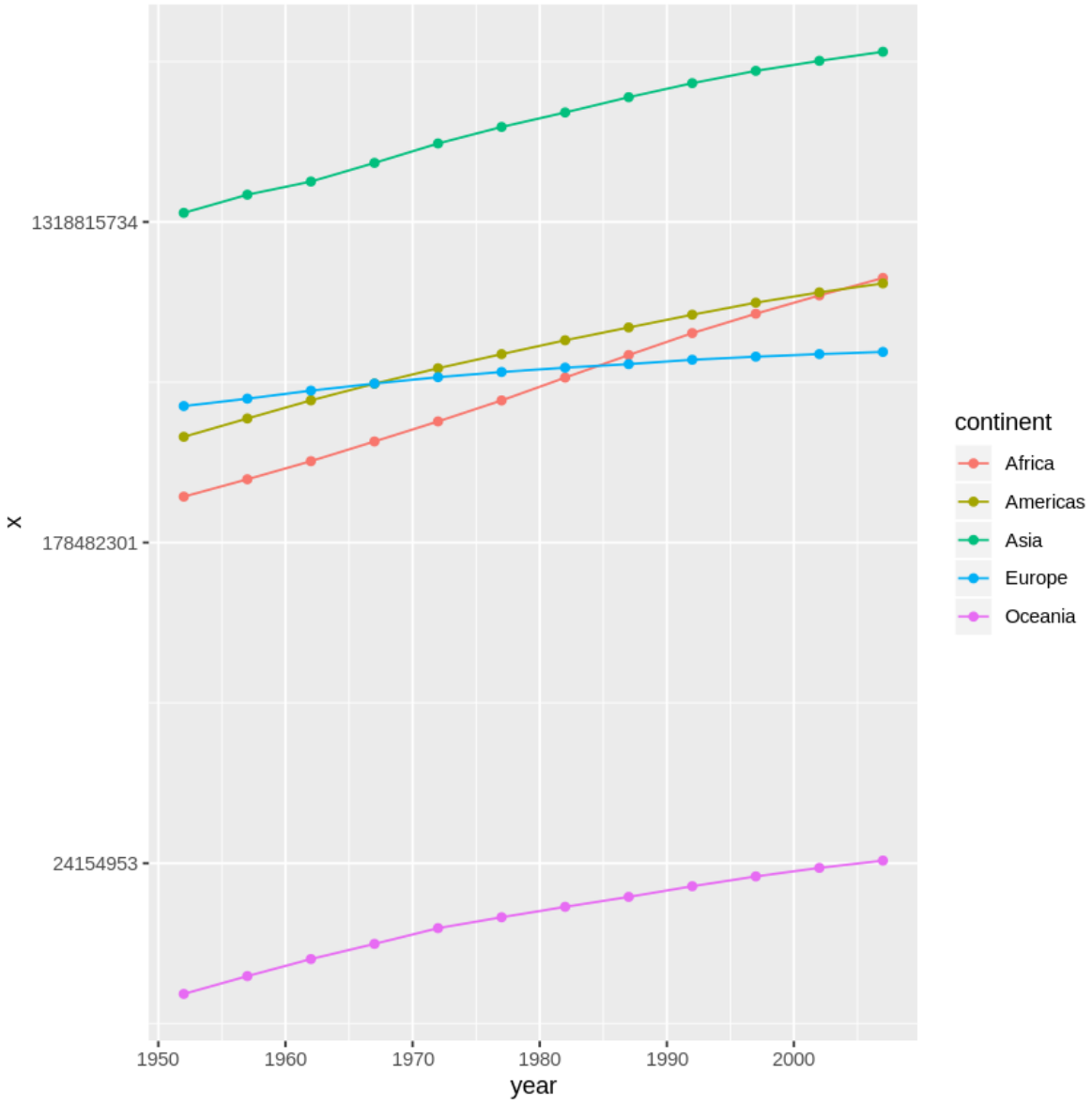
In [18]:
```
aggdata <- aggregate(data$pop, by=list(continent=data$continent, year=
data$year), FUN=sum, na.rm=TRUE)

head(aggdata)
```

A data.frame: 6 × 3

continent	year	x
<fct>	<int>	<dbl>
Africa	1952	237640501
Americas	1952	345152446
Asia	1952	1395357352
Europe	1952	418120846
Oceania	1952	10686006
Africa	1957	264837738

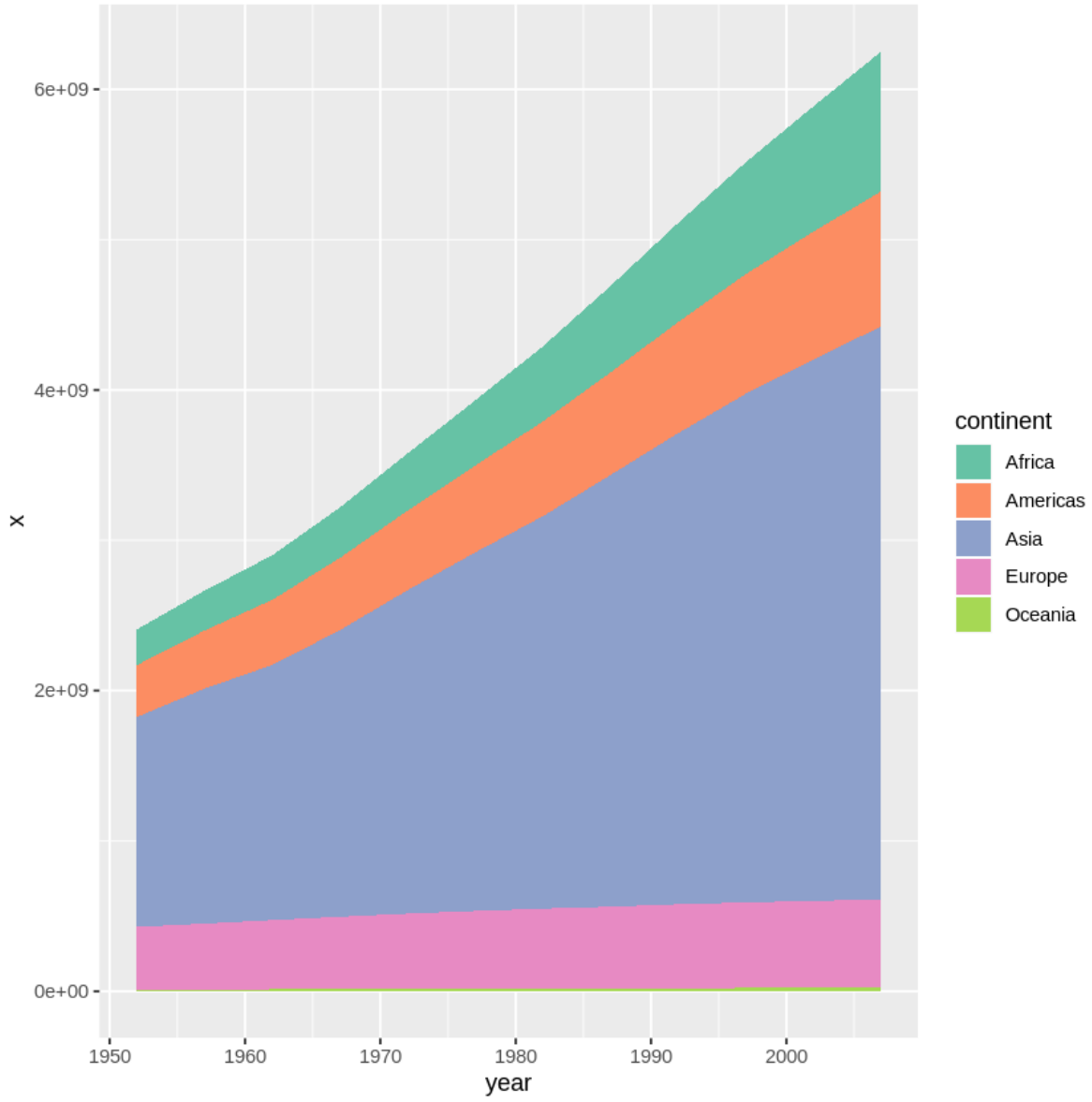
Exercise 5: Now **repeat exercise 4 with this aggregate data** and group and color by continent. Do you see a pattern??

In [25]:
```
ggplot(aggdata, aes(x= year,y=x, group=continent)) +
       geom_line(aes(color = continent)) + geom_point(aes(color= conti
nent)) + scale_y_continuous(trans = 'log')

#Yes, I can see the pattern now.
```



Exercise 6: Now, plot a **stacked area chart** to see the same as in exercise 5. Instead of group and color, use only **fill** parameter for **continent**, and use **geom_area**.

In [26]:
```
ggplot(aggdata, aes(x = year,y = x,fill=continent)) + geom_area() + scale_fill_brewer(palette="Set2")
```



Find percentages: The above plot shows actual population numbers and they grow in time. We want to see the percentage change of the continents' populations with respect to total world population. The code below computes that.

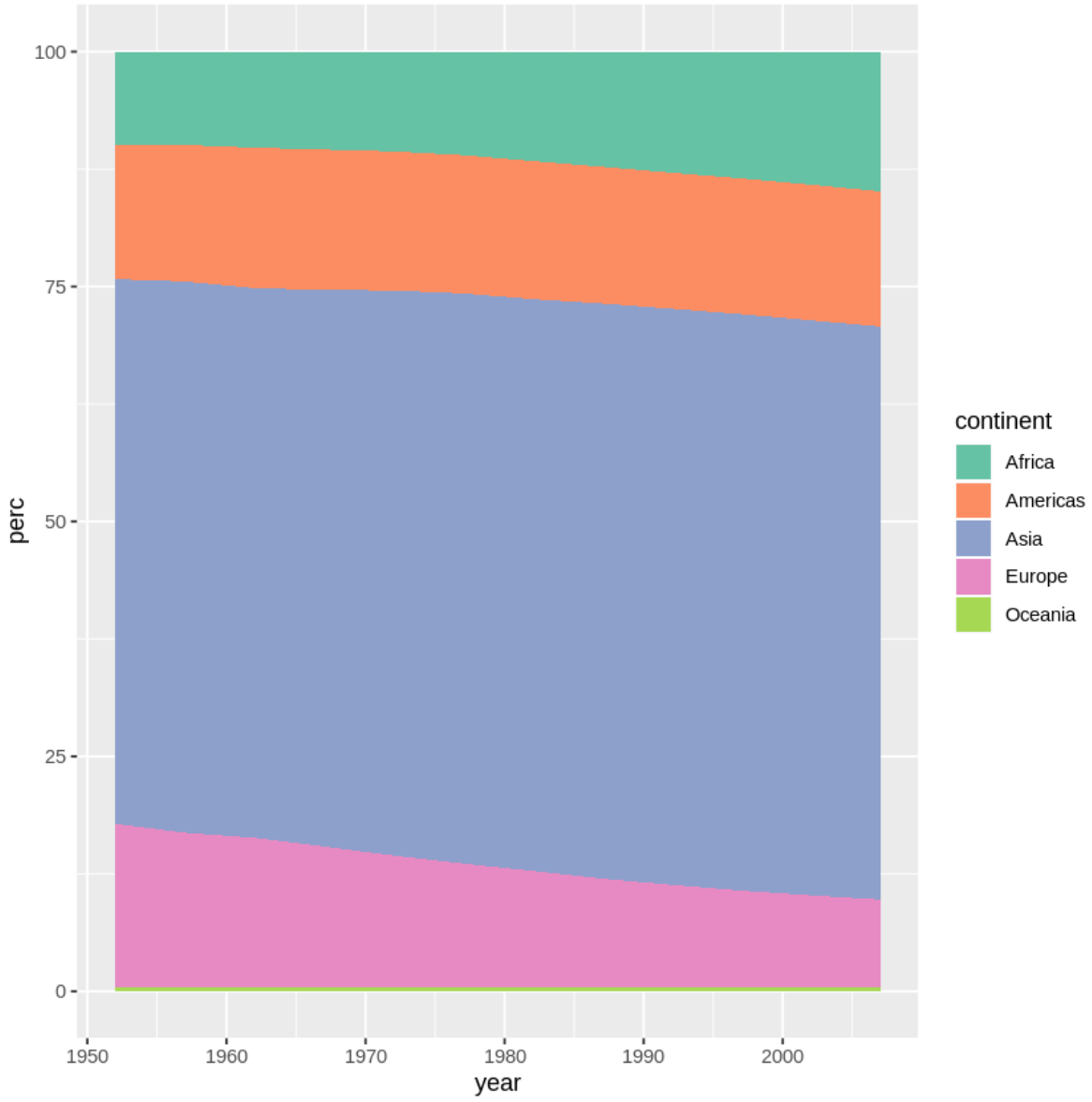
In [27]:
```
my_fun=function(vec){ as.numeric(vec[3]) / sum(aggdata$x[aggdata$year==vec[2]]) *100 }
aggdata$perc=apply(aggdata , 1 , my_fun)

head(aggdata)
```

A data.frame: 6 × 4

continent	year	x	perc
<fct>	<int>	<dbl>	<dbl>
Africa	1952	237640501	9.8730674
Americas	1952	345152446	14.3397836
Asia	1952	1395357352	57.9718401
Europe	1952	418120846	17.3713456
Oceania	1952	10686006	0.4439633
Africa	1957	264837738	9.9398470

Exercise 7: Now, **plot the same as exercise 6** but use **perc** as the y axis.

In [33]:
```
ggplot(aggdata, aes(x=year,y=perc, fill=continent)) +
    geom_area() + scale_fill_brewer(palette="Set2")
```



Exercise 8: We will aggregate once more; this time we will compute the **mean gdp per capita for continents and years**. **It's your turn this time.**

In [34]:
```
aggdata2 <- aggregate(data$gdpPercap, by=list(continent=data$continen
t, year=data$year), FUN= mean, na.rm=TRUE)
head(aggdata2)
```

A data.frame: 6 × 3

continent	year	x
<fct>	<int>	<dbl>
Africa	1952	1252.572
Americas	1952	4079.063
Asia	1952	5195.484
Europe	1952	5661.057
Oceania	1952	10298.086
Africa	1957	1385.236

Exercise 9: Plot a **heatmap** using **plot_ly function** for **years** (x-axis) vs. **continents** (y-axis) using the **mean gdp per capita as the z value**.

In [36]:
```
plot_ly(aggdata2, x = aggdata2$year , y = aggdata2$continent, z = ~agg
data2$x, type="heatmap", colors=colorRamp(c("white", "blue")))
```

Exercise 10: Plot a **boxplot** for **gdp per capita** using **plot_ly function** for **continents**.

Use the **whole** data set, **color by continent**, and make sure **y axis is in log scale**. Put continents in x-axis and gdp on y-axis.

When hovering over data, what do you notice about first and third quartiles for each continent (hint: think of income inequality) ?

In [37]:
```
plot_ly(data, x= data$continent, y=~data$gdpPercap ,color = ~data$cont
inent,type="box") %>% layout(yaxis = list(type = "log"))
```

In [ ]:
```

```