9.1 Choropleth Maps

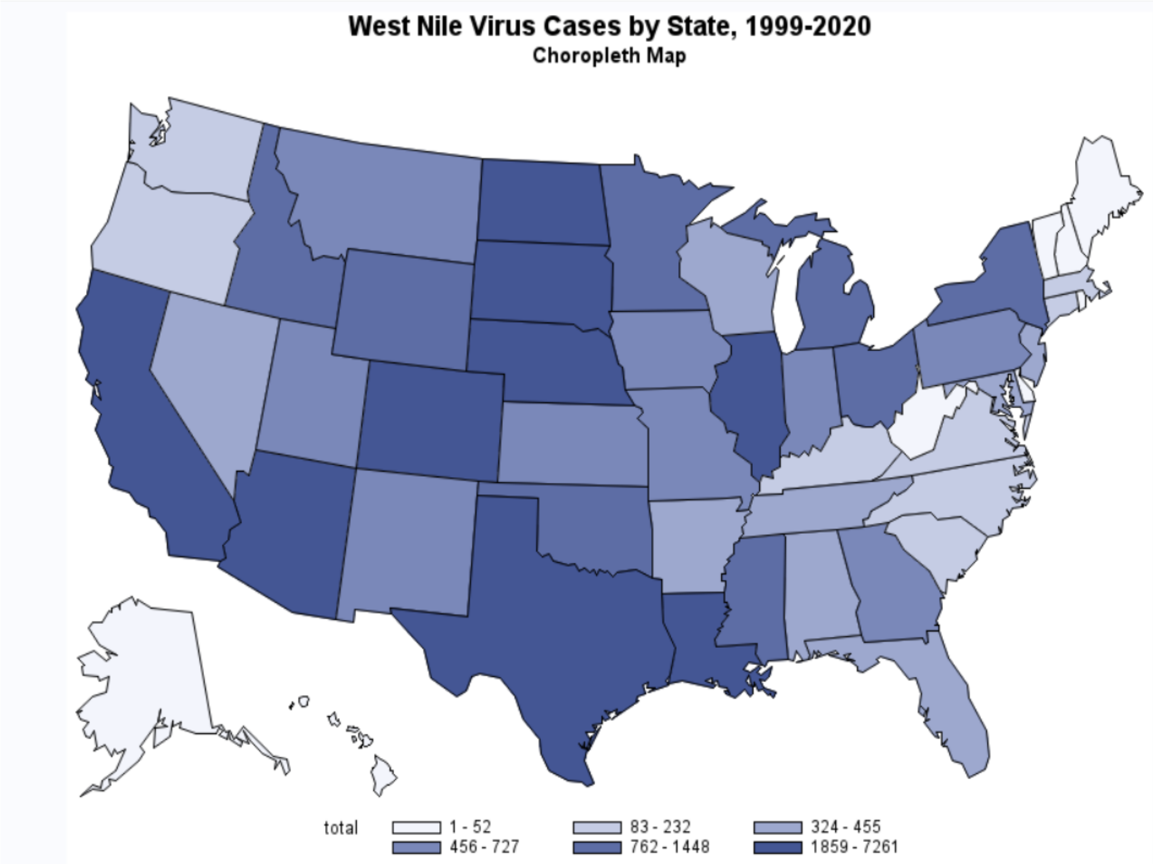
Definition. A **choropleth map** is a map that uses different colors (or shades of the same color) to indicate the level of magnitude of values of a certain variable in predefined areas. The word “choropleth” comes from Greek “khōra” place + “plēthos” multitude.

Example. The files “WestNileVirusCasesSAS.csv” and “WestNileVirusCasesR.csv”contain data on the number of West Nile virus diagnoses in the US in 2005, 2010, 2015, 2020, and a total for 1999-2020, by state. Below we create choropleth maps for these data.

First, we depict the total number of diagnoses.

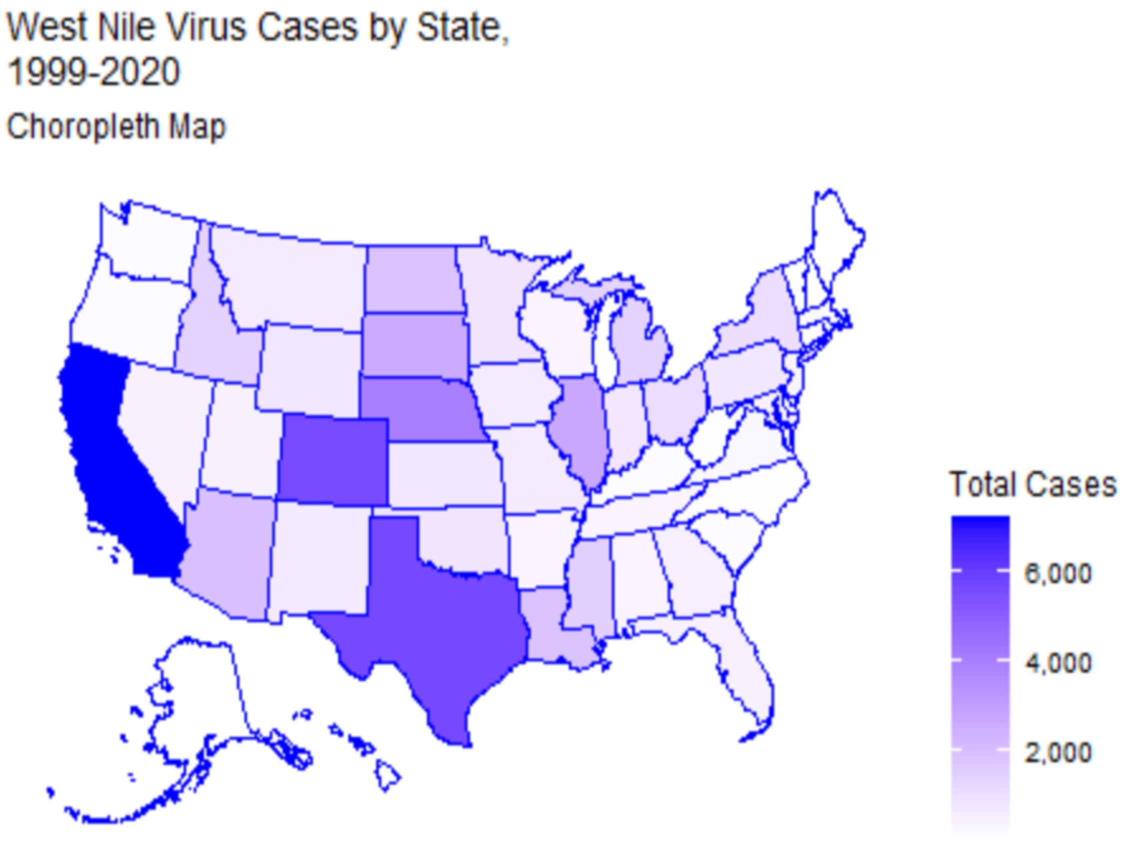
In SAS: The states on the map of US are identified by a numerical variable named “state” which values are the Federal Information Processing Standards (FIPS) codes.

```
proc import out=cases
datafile="./WestNileVirusCasesSAS.csv" dbms=csv replace;
run;

title "West Nile Virus Cases by State, 1999-2020";
title2 "Choropleth Map";
proc gmap data=cases map=maps.us;
id state;
choro total;
run;
quit;
```



In R: The state identification can be in the form of state name. The variable name should be “state”.

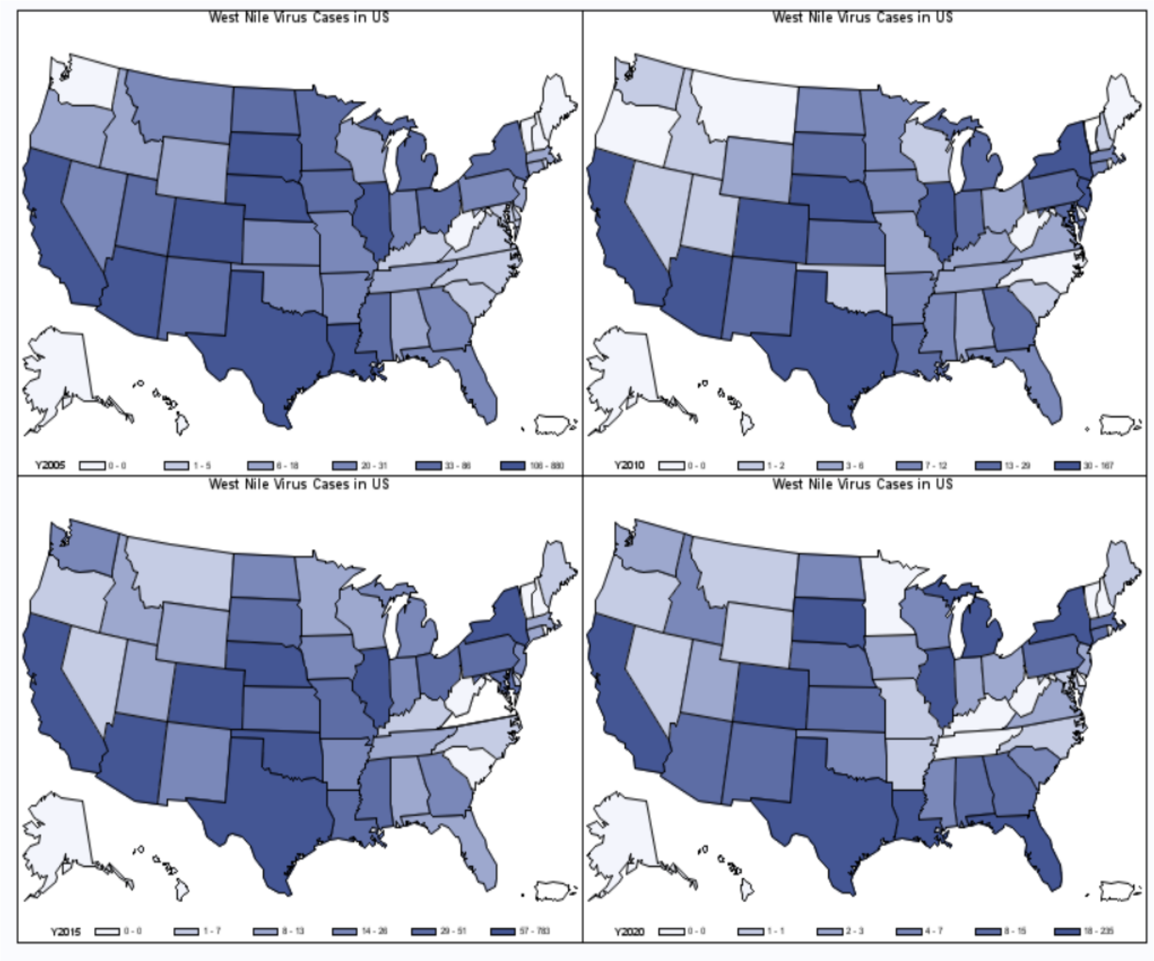
```
cases.data<- read.csv(file="./WestNileVirusCasesR.csv", header=TRUE, sep=",")

library(usmap)
library(ggplot2)

plot_usmap(data = cases.data, values = "total", color = "blue") +
scale_fill_continuous(low = "white", high = "blue", name = "Total Cases",
label = scales::comma) + labs(title = "West Nile Virus Cases by State,
1999-2020", subtitle = "Choropleth Map") + theme(legend.position = "right")
```



Next, we create four maps, for the number of cases in 2005, 2010, 2015, and 2020, respectively.

In SAS:

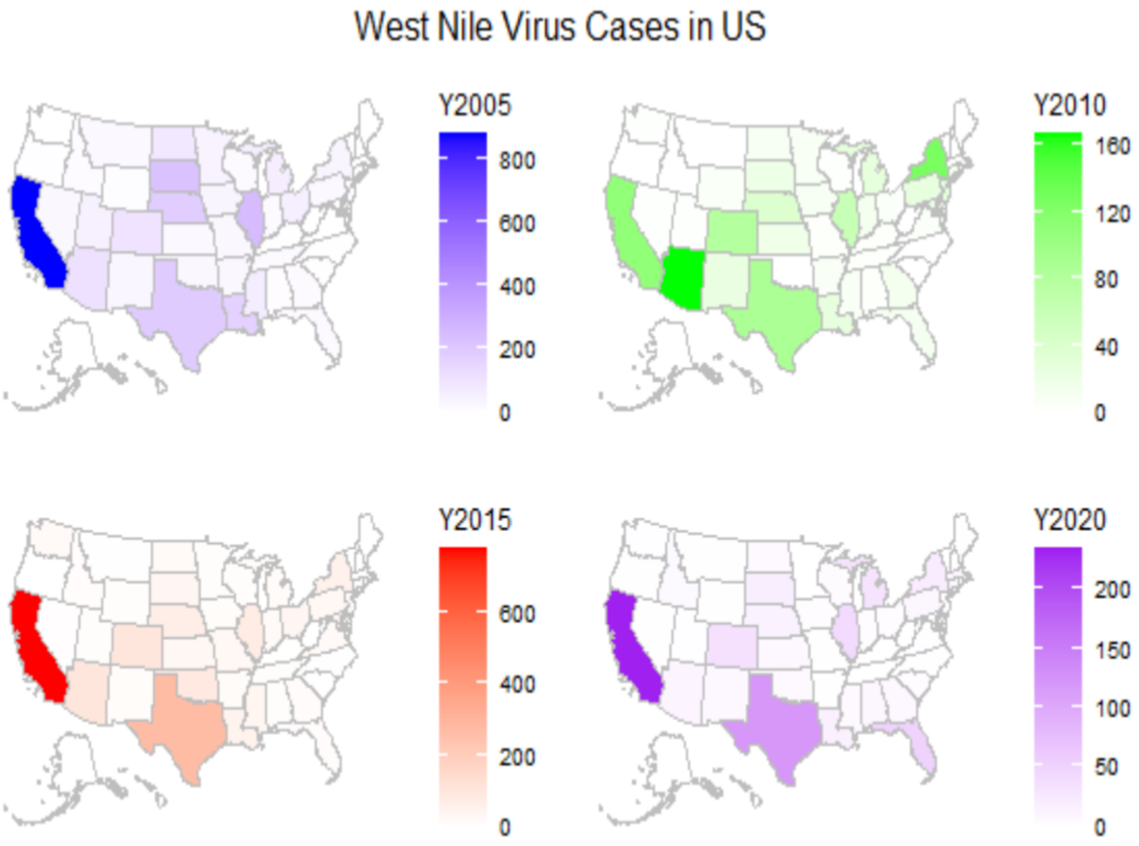
```
proc gmap data=cases map=maps.us all;
id state;
choro Y2005 Y2010 Y2015 Y2020/name="graph";
run;
quit;

title "West Nile Virus Cases in US";
proc greplay igout=work.gseg tc=sashelp.templt template=l2r2 nofs;
treplay 1:graph 2:graph2 3:graph1 4:graph3;
run;
```


West Nile Virus Cases in US
West Nile Virus Cases in US
Y2005 0 - 0 1 - 5 6 - 18 20 - 31 33 - 86 106 - 880
Y2010 0 - 0 1 - 2 3 - 6 7 - 12 13 - 29 30 - 167
West Nile Virus Cases in US
West Nile Virus Cases in US
Y2015 0 - 0 1 - 7 8 - 13 14 - 26 29 - 51 57 - 783
Y2020 0 - 0 1 - 1 2 - 3 4 - 7 8 - 15 18 - 235

In R:

```
library("gridExtra")

plot2005<- plot_usmap(data = cases.data, values = "Y2005", color = "gray")+
scale_fill_continuous(low = "white", high = "blue")+theme(legend.position
= "right")

plot2010<- plot_usmap(data = cases.data, values = "Y2010", color = "gray")+
scale_fill_continuous(low = "white", high = "green")+theme(legend.position
= "right")

plot2015<- plot_usmap(data = cases.data, values = "Y2015", color = "gray")+
scale_fill_continuous(low = "white", high = "red")+theme(legend.position
= "right")
```

```
plot2020<- plot_usmap(data = cases.data, values = "Y2020", color = "gray")+
scale_fill_continuous(low = "white", high = "purple")+theme(legend.position
= "right")

grid.arrange(plot2005, plot2010, plot2015, plot2020, ncol = 2, nrow=2, top
= "West Nile Virus Cases in US")
```

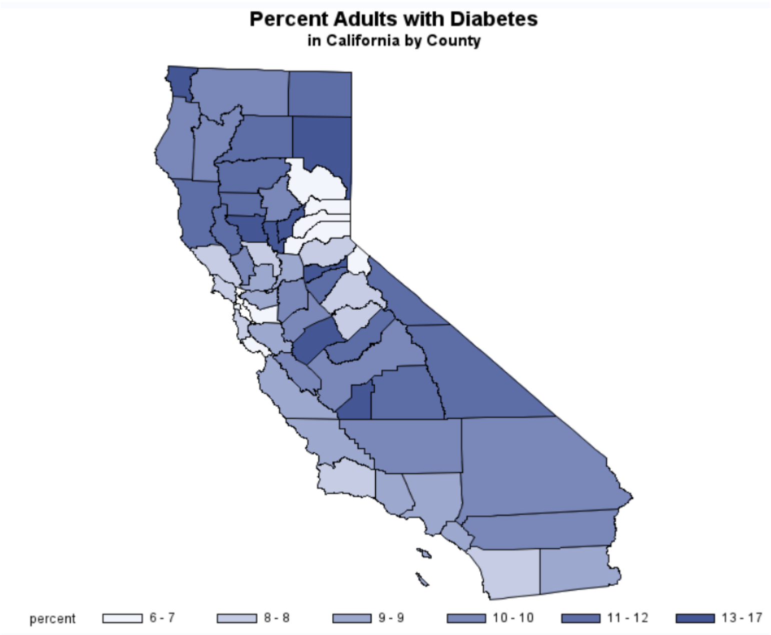


□

Example. The data sets “CA_Diabetes_by_CountySAS.csv” and “CA_Diabetes_by_CountyR.csv” give percentage of adults aged 20 and above with diagnosed diabetes in 2017 by county in California. We create choropleth maps for the data.

In SAS: Counties with the state should be identified by 2 or 3-digit FIPS codes. The variable should be called “county”.

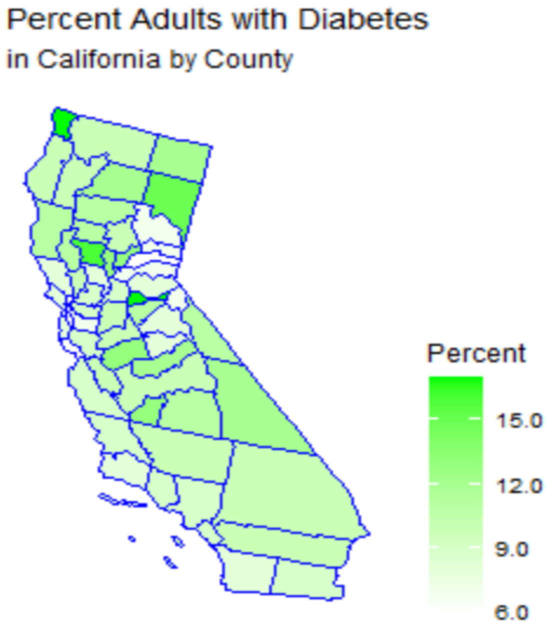
```
proc import out=diabetes
datafile="./CA_Diabetes_by_CountySAS.csv" dbms=csv replace;

proc gproject out=CA data=maps.counties;
id county;
where state eq 6;
run;

title "Percent Adults with Diabetes";
title2 "in California by County";
proc gmap data=diabetes map=CA;
id county;
choro percent;
run;
quit;
```



In R: Counties within a state should be identified by a 4-digit FIPS code. The variable should be called “fips”.

```
diabetes.data<- read.csv(file="./CA_Diabetes_by_CountyR.csv", header=TRUE,
sep=",")
```

```
library(usmap)
library(ggplot2)

plot_usmap(data = diabetes.data, values = "percent", regions="counties",
include="CA", color = "blue") + scale_fill_continuous(low = "white",
high = "green", name = "Percent", label = scales::comma) + labs(title =
"Percent Adults with Diabetes", subtitle = "in California by County") +
theme(legend.position = "right")
```

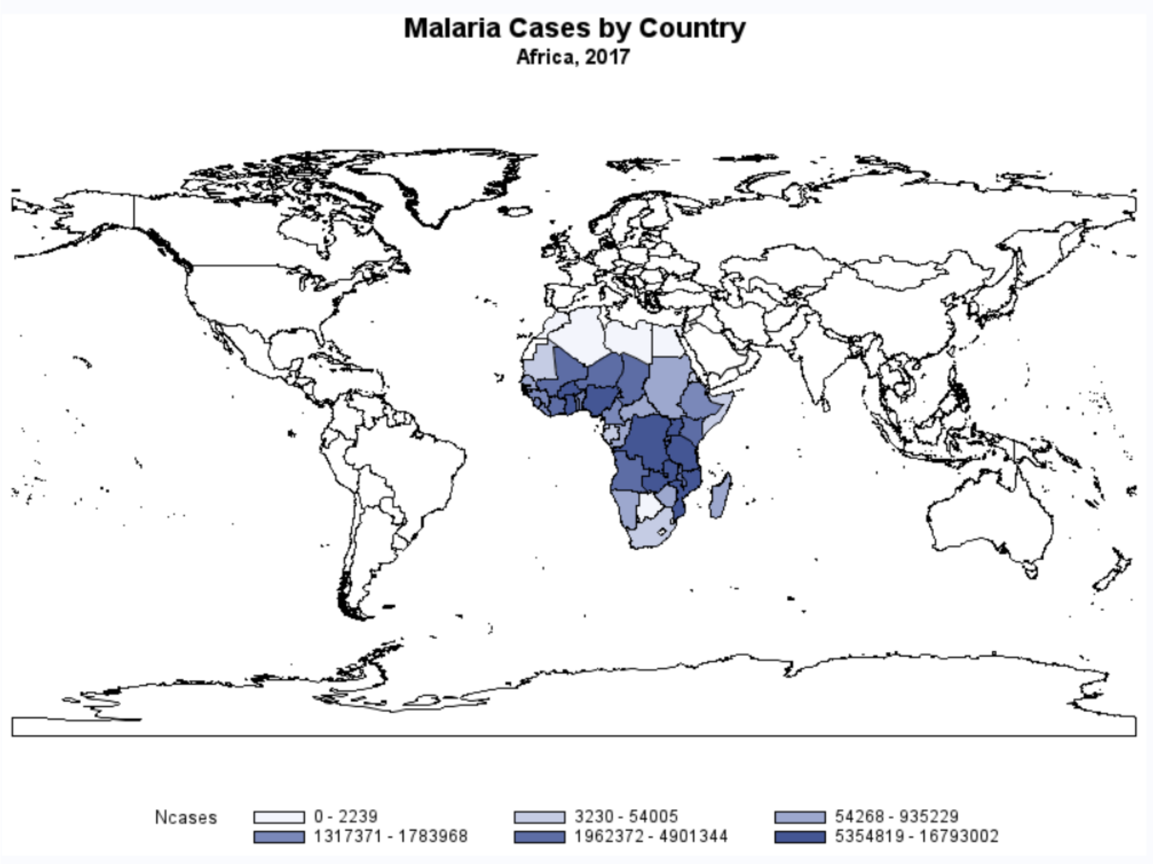


□

Example. Malaria transmission is endemic in most African countries south of Sahara desert. The data file “AfricaMalariaCases.csv” lists the number of malaria cases in 2017 by country. We plot the given information on a choropleth map. First we plot the map of the entire World, and then separately of Africa. In SAS, countries are identified by a numeric variable “ID”, which contains 3-digit Geographic Location Code (GLC) of countries. In R, countries are identified by the variable “Country” which gives a 3-letter International Standard Organization (ISO) abbreviation for countries. The codes can be retrieved from SAS by typing PROC PRINT DATA=MAPS.NAMES; RUN; The column “ID” gives the ids, the column “ISOALPHA3” gives ISO.

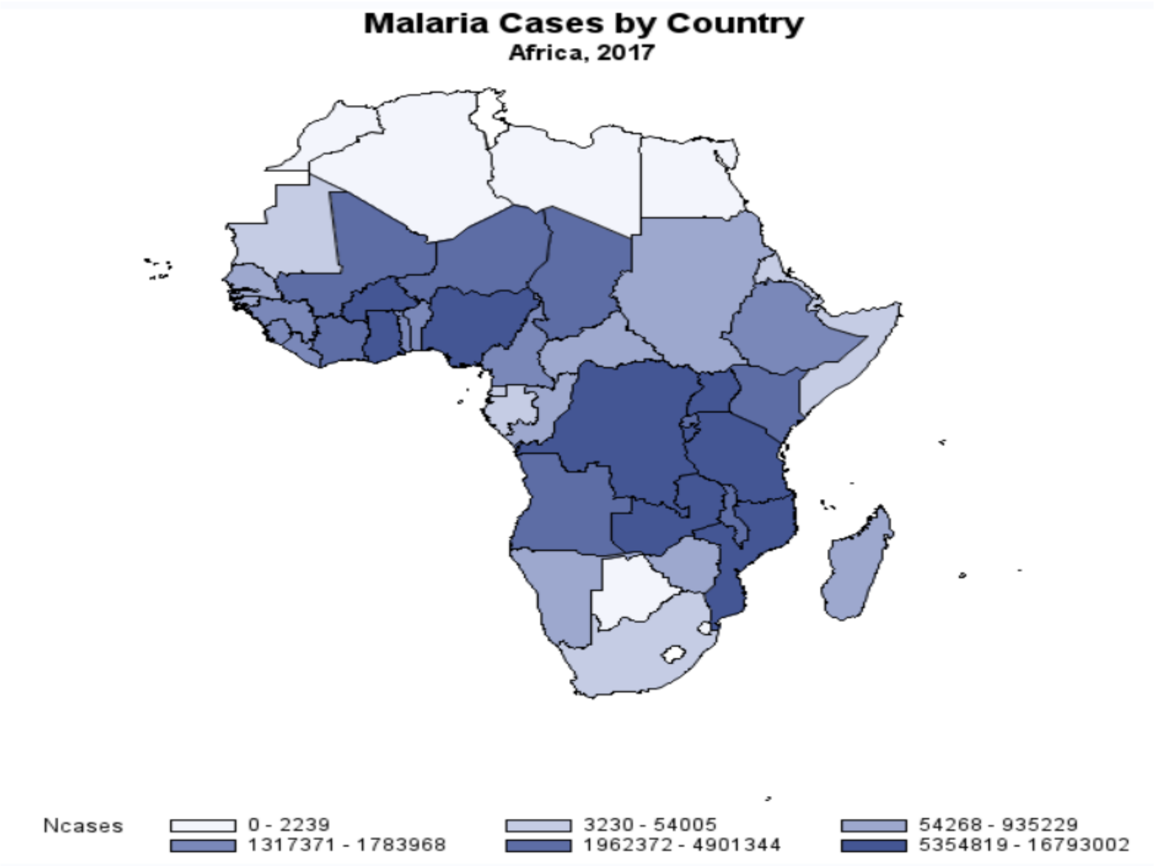
In SAS:

```
proc import out=malaria
datafile="./AfricaMalariaCases.csv" dbms=csv replace;

title "Malaria Cases by Country";
title2 "Africa, 2017" ;
proc gmap data=malaria map=maps.world all;
id id;
choro Ncases;
run;
quit;
```



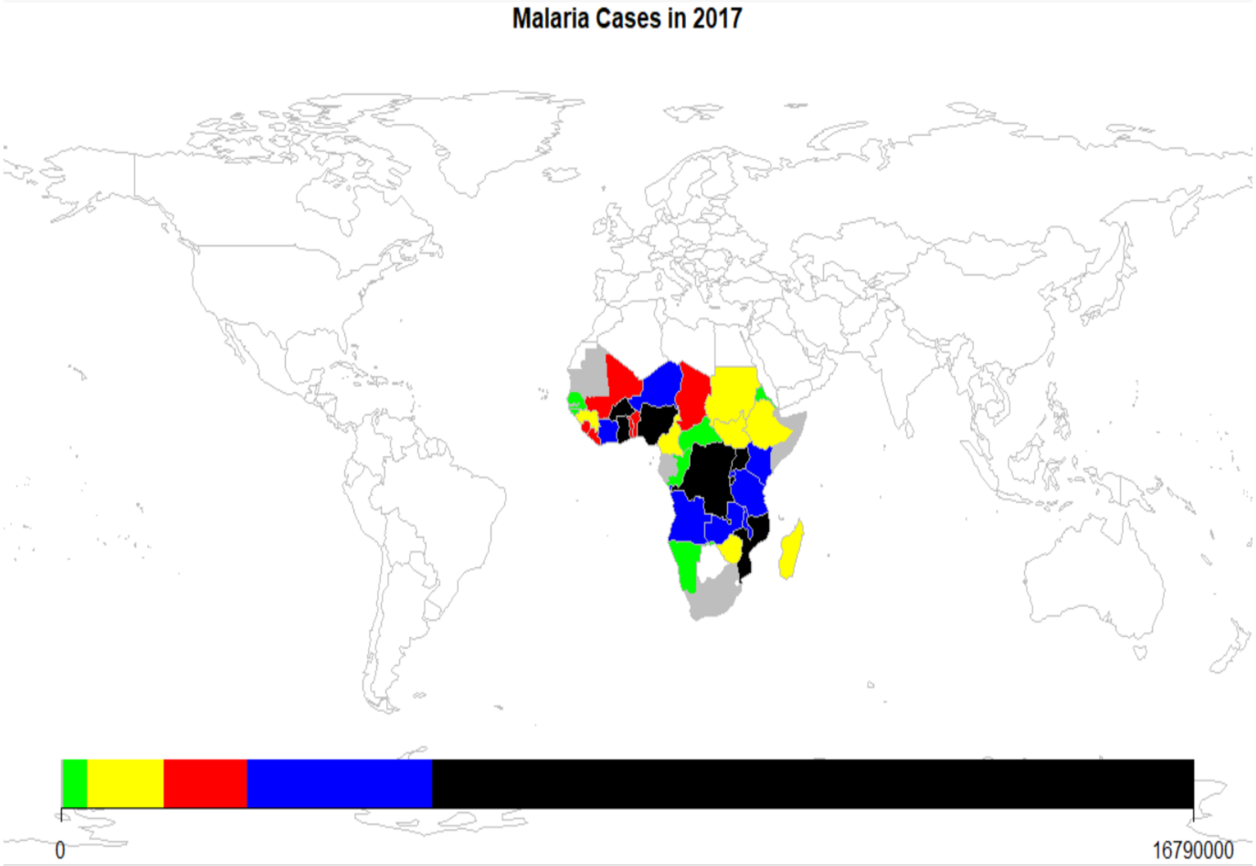
```
proc gmap data=malaria map=maps.world;
id id;
choro Ncases;
run;
```

```
quit;
```

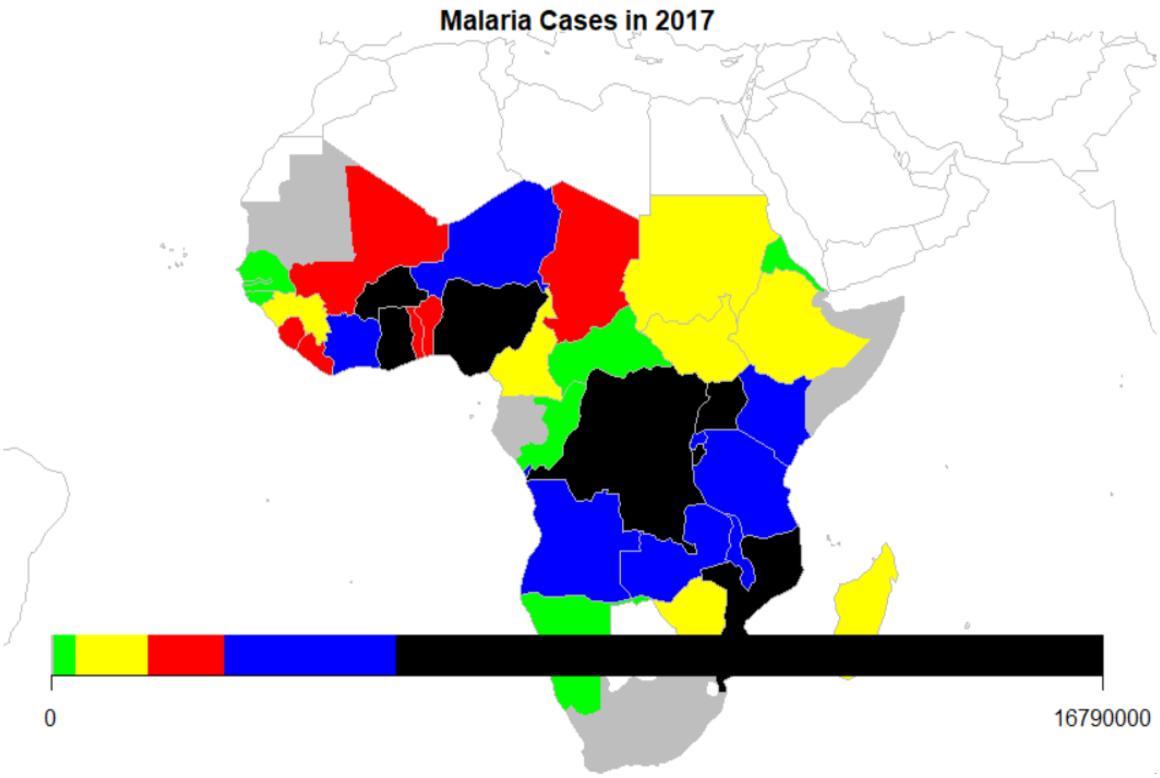


In R:

```
library(rworldmap)
malaria.data<- read.csv(file="./AfricaMalariaCases.csv", header=TRUE, sep=",")

map <- joinCountryData2Map(malaria.data, nameJoinColumn="Country",
joinCode="NAME" )

mapDevice("x11") #creating a separate window

#plotting map of the World
mapCountryData(Map, nameColumnToPlot="Ncases", mapTitle="Malaria
Cases in 2017",colourPalette=c("white","gray","green","yellow", "red","blue","black"))
```



```
#plotting map of Africa
mapCountryData(Map, nameColumnToPlot="Ncases", mapTitle="Malaria
Cases in 2017", mapRegion="Africa",colourPalette=c("white","gray","green",
"yellow","red","blue","black"))
```


Malaria Cases in 2017
0
16790000

□