

# Passage 24 A #2

## The World Population

We have seen that the world population has reached 7.3 billion in 2015.

```
CountryData["World", "Population"]
```


$7.13001 \times 10^9$  people

```
CountryData["World", {"Population", 1970}]
```

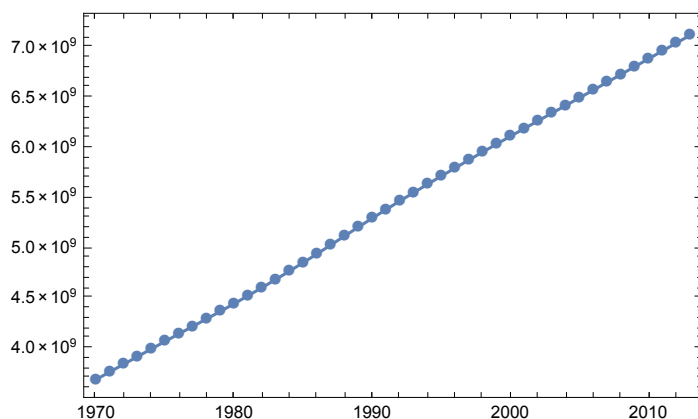
$3.69619 \times 10^9$  people

We connect to a data base to retrieve various data.

```
data = CountryData["World", {"Population", All}]
```

```
TimeSeries [  Time: 01 Jan 1970 to 01 Jan 2013  
Data points 44 ]
```

```
DateListPlot [data, PlotMarkers -> Automatic]
```




## Country by country population

Today, we are going to see population country by country. Let's start with Japan.

```
CountryData["Japan", "Population"]
```

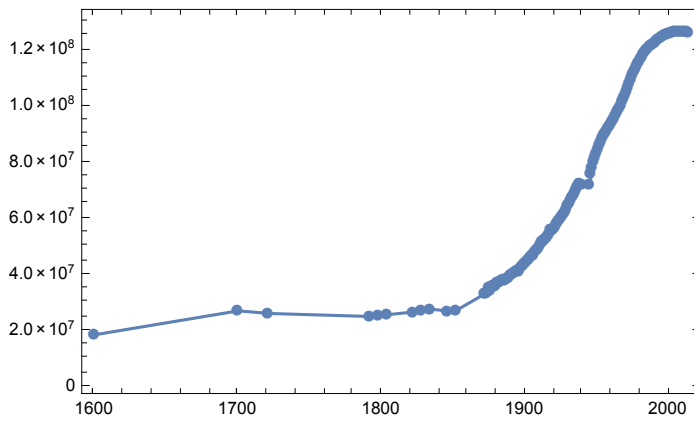
126 225 259 people

```
dataJP = CountryData["Japan", {"Population", All}]
```

```
TimeSeries [  Time: 01 Jan 1600 to 01 Jan 2014  
Data points 149 ]
```

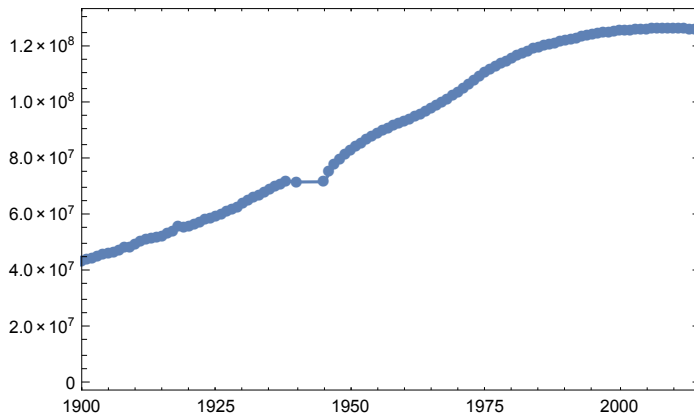
Data is available since 1600!

```
DateListPlot[dataJP, PlotMarkers -> Automatic]
```

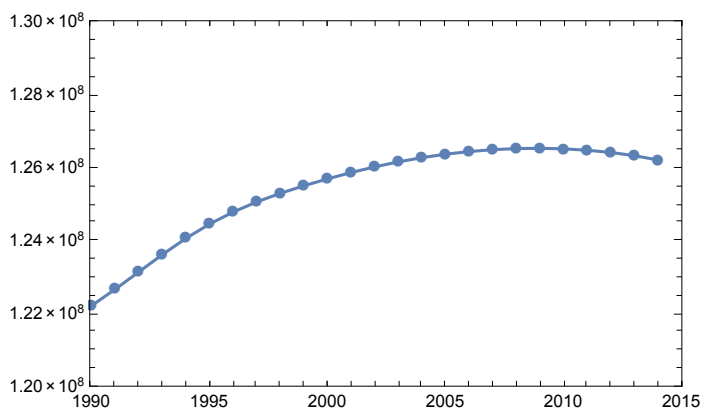


Some data are missing during the war. Japanese population had been increasing every year after the war. But it has just started to decline.

```
DateListPlot[dataJP, PlotMarkers -> Automatic,
  PlotRange -> {{DateList["1900"], DateList["2015"]}, All}]
```



```
DateListPlot[dataJP, PlotMarkers -> Automatic,
  PlotRange -> {{DateList["1990"], DateList["2015"]}, {1.2 x 10^8, 1.3 x 10^8}}]
```

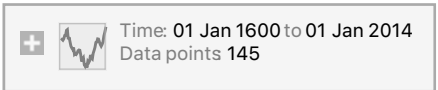


Let's see the population of USA.

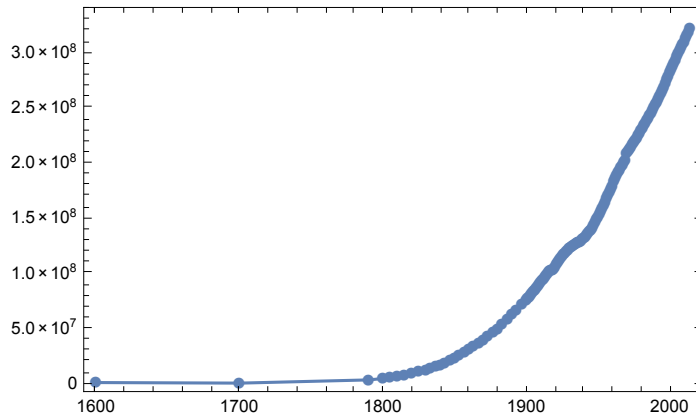
```
CountryData["USA", "Population"]
```

322 422 965 people

```
dataUSA = CountryData["USA", {"Population", All}]
```

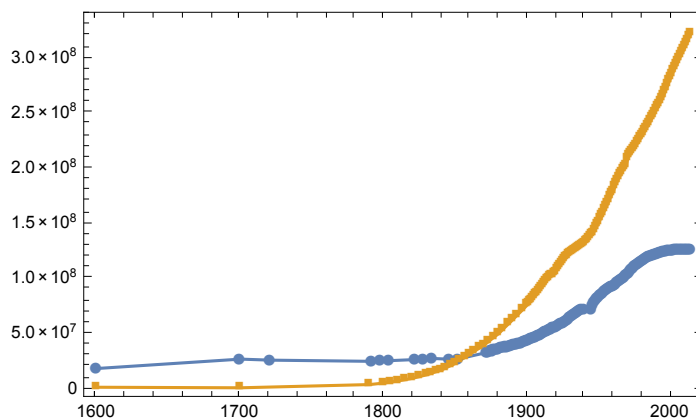
TimeSeries [  ]

```
DateListPlot[dataUSA, PlotMarkers -> Automatic]
```



It's obvious that USA is much more populous than Japan and it is still increasing in the population.

```
DateListPlot[{dataJP, dataUSA}, PlotMarkers -> Automatic]
```

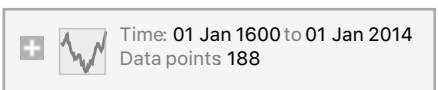


How about China, the most populous country in the world?

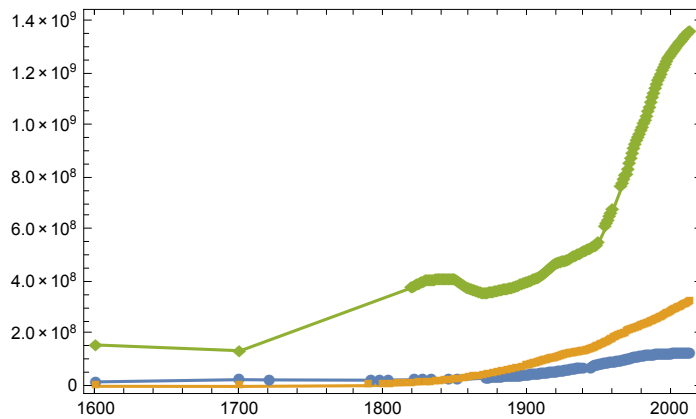
```
CountryData["China", "Population"]
```

1 364 773 138 people

```
dataChina = CountryData["China", {"Population", All}]
```

TimeSeries [  ]

```
DateListPlot[{dataJP, dataUSA, dataChina}, PlotMarkers -> Automatic]
```



The database has various information about many countries.

```
CountryData["Countries"]
```

```
{Entity[Country, Afghanistan], Entity[Country, Albania],
 Entity[Country, Algeria], Entity[Country, AmericanSamoa],
 Entity[Country, Andorra], Entity[Country, Angola], Entity[Country, Anguilla],
 Entity[Country, AntiguaBarbuda], Entity[Country, Argentina],
 Entity[Country, Armenia], Entity[Country, Aruba], Entity[Country, Australia],
 Entity[Country, Austria], Entity[Country, Azerbaijan],
 Entity[Country, Bahamas], Entity[Country, Bahrain],
 Entity[Country, Bangladesh], Entity[Country, Barbados],
 Entity[Country, Belarus], Entity[Country, Belgium], Entity[Country, Belize],
 Entity[Country, Benin], Entity[Country, Bermuda], Entity[Country, Bhutan],
 Entity[Country, Bolivia], Entity[Country, BosniaHerzegovina],
 Entity[Country, Botswana], Entity[Country, Brazil],
 Entity[Country, BritishVirginIslands], Entity[Country, Brunei],
 Entity[Country, Bulgaria], Entity[Country, BurkinaFaso],
 Entity[Country, Burundi], Entity[Country, Cambodia],
 Entity[Country, Cameroon], Entity[Country, Canada], Entity[Country, CapeVerde],
 Entity[Country, CaymanIslands], Entity[Country, CentralAfricanRepublic],
 Entity[Country, Chad], Entity[Country, Chile], Entity[Country, China],
 Entity[Country, ChristmasIsland], Entity[Country, CocosKeelingIslands],
 Entity[Country, Colombia], Entity[Country, Comoros],
 Entity[Country, CookIslands], Entity[Country, CostaRica],
 Entity[Country, Croatia], Entity[Country, Cuba], Entity[Country, Curacao],
 Entity[Country, Cyprus], Entity[Country, CzechRepublic],
 Entity[Country, DemocraticRepublicCongo], Entity[Country, Denmark],
 Entity[Country, Djibouti], Entity[Country, Dominica],
 Entity[Country, DominicanRepublic], Entity[Country, EastTimor],
 Entity[Country, Ecuador], Entity[Country, Egypt], Entity[Country, ElSalvador],
 Entity[Country, EquatorialGuinea], Entity[Country, Eritrea],
 Entity[Country, Estonia], Entity[Country, Ethiopia],
 Entity[Country, FalklandIslands], Entity[Country, FaroeIslands],
 Entity[Country, Fiji], Entity[Country, Finland], Entity[Country, France],
 Entity[Country, FrenchGuiana], Entity[Country, FrenchPolynesia],
 Entity[Country, Gabon], Entity[Country, Gambia], Entity[Country, GazaStrip],
 Entity[Country, Georgia], Entity[Country, Germany], Entity[Country, Ghana],
 Entity[Country, Gibraltar], Entity[Country, Greece], Entity[Country, Greenland],
 Entity[Country, Grenada], Entity[Country, Guadeloupe], Entity[Country, Guam],
```


Entity[Country, Guatemala], Entity[Country, Guernsey], Entity[Country, Guinea],  
 Entity[Country, GuineaBissau], Entity[Country, Guyana],  
 Entity[Country, Haiti], Entity[Country, Honduras], Entity[Country, HongKong],  
 Entity[Country, Hungary], Entity[Country, Iceland], Entity[Country, India],  
 Entity[Country, Indonesia], Entity[Country, Iran], Entity[Country, Iraq],  
 Entity[Country, Ireland], Entity[Country, IsleOfMan], Entity[Country, Israel],  
 Entity[Country, Italy], Entity[Country, IvoryCoast], Entity[Country, Jamaica],  
 Entity[Country, Japan], Entity[Country, Jersey], Entity[Country, Jordan],  
 Entity[Country, Kazakhstan], Entity[Country, Kenya], Entity[Country, Kiribati],  
 Entity[Country, Kosovo], Entity[Country, Kuwait], Entity[Country, Kyrgyzstan],  
 Entity[Country, Laos], Entity[Country, Latvia], Entity[Country, Lebanon],  
 Entity[Country, Lesotho], Entity[Country, Liberia], Entity[Country, Libya],  
 Entity[Country, Liechtenstein], Entity[Country, Lithuania],  
 Entity[Country, Luxembourg], Entity[Country, Macau], Entity[Country, Macedonia],  
 Entity[Country, Madagascar], Entity[Country, Malawi], Entity[Country, Malaysia],  
 Entity[Country, Maldives], Entity[Country, Mali], Entity[Country, Malta],  
 Entity[Country, MarshallIslands], Entity[Country, Martinique],  
 Entity[Country, Mauritania], Entity[Country, Mauritius],  
 Entity[Country, Mayotte], Entity[Country, Mexico], Entity[Country, Micronesia],  
 Entity[Country, Moldova], Entity[Country, Monaco], Entity[Country, Mongolia],  
 Entity[Country, Montenegro], Entity[Country, Montserrat],  
 Entity[Country, Morocco], Entity[Country, Mozambique], Entity[Country, Myanmar],  
 Entity[Country, Namibia], Entity[Country, Nauru], Entity[Country, Nepal],  
 Entity[Country, Netherlands], Entity[Country, NewCaledonia],  
 Entity[Country, NewZealand], Entity[Country, Nicaragua],  
 Entity[Country, Niger], Entity[Country, Nigeria], Entity[Country, Niue],  
 Entity[Country, NorfolkIsland], Entity[Country, NorthernMarianaIslands],  
 Entity[Country, NorthKorea], Entity[Country, Norway], Entity[Country, Oman],  
 Entity[Country, Pakistan], Entity[Country, Palau], Entity[Country, Panama],  
 Entity[Country, PapuaNewGuinea], Entity[Country, Paraguay],  
 Entity[Country, Peru], Entity[Country, Philippines],  
 Entity[Country, PitcairnIslands], Entity[Country, Poland],  
 Entity[Country, Portugal], Entity[Country, PuertoRico], Entity[Country, Qatar],  
 Entity[Country, RepublicCongo], Entity[Country, Reunion],  
 Entity[Country, Romania], Entity[Country, Russia], Entity[Country, Rwanda],  
 Entity[Country, SaintHelena], Entity[Country, SaintKittsNevis],  
 Entity[Country, SaintLucia], Entity[Country, SaintPierreMiquelon],  
 Entity[Country, SaintVincentGrenadines], Entity[Country, Samoa],  
 Entity[Country, SanMarino], Entity[Country, SaoTomePrincipe],  
 Entity[Country, SaudiArabia], Entity[Country, Senegal], Entity[Country, Serbia],  
 Entity[Country, Seychelles], Entity[Country, SierraLeone],  
 Entity[Country, Singapore], Entity[Country, SintMaarten],  
 Entity[Country, Slovakia], Entity[Country, Slovenia],  
 Entity[Country, SolomonIslands], Entity[Country, Somalia],  
 Entity[Country, SouthAfrica], Entity[Country, SouthKorea],  
 Entity[Country, SouthSudan], Entity[Country, Spain],  
 Entity[Country, SriLanka], Entity[Country, Sudan], Entity[Country, Suriname],  
 Entity[Country, Svalbard], Entity[Country, Swaziland], Entity[Country, Sweden],  
 Entity[Country, Switzerland], Entity[Country, Syria], Entity[Country, Taiwan],  
 Entity[Country, Tajikistan], Entity[Country, Tanzania],  
 Entity[Country, Thailand], Entity[Country, Togo], Entity[Country, Tokelau],  
 Entity[Country, Tonga], Entity[Country, TrinidadTobago],

```
Entity[Country, Tunisia], Entity[Country, Turkey],
Entity[Country, Turkmenistan], Entity[Country, TurksCaicosIslands],
Entity[Country, Tuvalu], Entity[Country, Uganda], Entity[Country, Ukraine],
Entity[Country, UnitedArabEmirates], Entity[Country, UnitedKingdom],
Entity[Country, UnitedStates], Entity[Country, UnitedStatesVirginIslands],
Entity[Country, Uruguay], Entity[Country, Uzbekistan], Entity[Country, Vanuatu],
Entity[Country, VaticanCity], Entity[Country, Venezuela],
Entity[Country, Vietnam], Entity[Country, WallisFutuna],
Entity[Country, WestBank], Entity[Country, WesternSahara],
Entity[Country, Yemen], Entity[Country, Zambia], Entity[Country, Zimbabwe]}
```

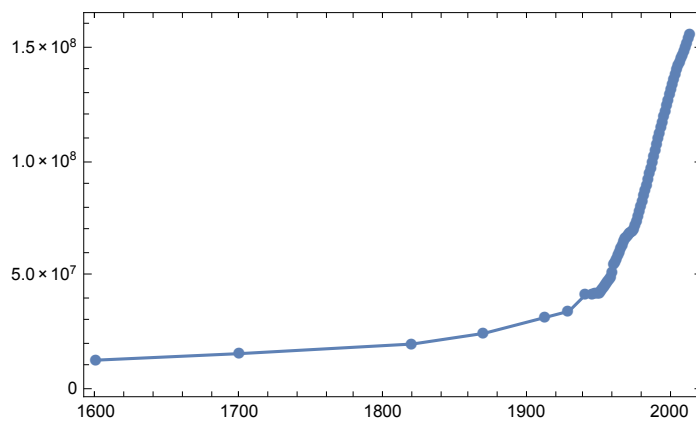
```
CountryData["Bangladesh", "Population"]
```

156 380 192 people

```
data = CountryData["Bangladesh", {"Population", All}]
```

```
TimeSeries[ Time: 01 Jan 1600 to 01 Jan 2014  
Data points 74]
```

```
DateListPlot[data, PlotMarkers -> Automatic]
```



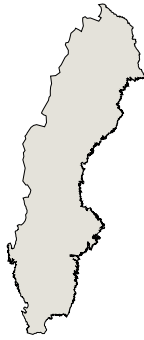
```
CountryData["Bangladesh", "Shape"]
```



```
CountryData["Sweden", "Population"]
```

9 595 619 people

`CountryData["Sweden", "Shape"]`



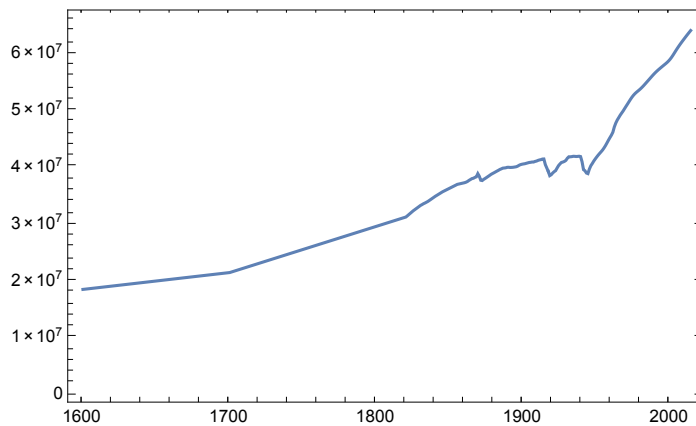
`CountryData["Sweden", "LifeExpectancy"]`

81.818 yr

`CountryData["France", "Shape"]`



`DateListPlot[CountryData["France", {"Population", All}]]`



`CountryData["Properties"]`

{AdultPopulation, AgriculturalProducts, AgriculturalValueAdded, Airports, AlternateNames, AlternateStandardNames, AMRadioStations, AnnualBirths, AnnualDeaths, AnnualHIVAIDSDeaths, ArableLandArea, ArableLandFraction, Area, BirthRateFraction, BorderingCountries, BordersLengths, BoundaryLength, CallingCode, CapitalCity, CapitalLocation, CapitalLocationLink, CellularPhones, CenterCoordinates, CenterLocationLink, ChildPopulation, Classes, ClimateTypes, CoastlineLength, ConstructionValueAdded, Continent, Coordinates, Countries, CountryCode, CropsLandArea, CropsLandFraction, CurrencyCode, CurrencyName, CurrencyShortName, CurrencyUnit, CurrentAccountBalance, DeathRateFraction, Dependencies, DependencyParent, EconomicAid, ElderlyPopulation, ElectricalGridFrequency,

ElectricalGridPlugImages, ElectricalGridPlugs, ElectricalGridSocketImages, ElectricalGridSockets, ElectricalGridVoltages, ElectricityConsumption, ElectricityExports, ElectricityImports, ElectricityProduction, EnvironmentalAgreements, EnvironmentalIssues, EthnicGroups, EthnicGroupsFractions, ExchangeRate, ExpenditureFractions, ExportCommodities, ExportPartners, ExportPartnersFractions, ExportValue, ExternalDebt, FemaleAdultPopulation, FemaleChildPopulation, FemaleElderlyPopulation, FemaleInfantMortalityFraction, FemaleLifeExpectancy, FemaleLiteracyFraction, FemaleMedianAge, FemalePopulation, FiscalYearDate, FixedInvestment, Flag, FlagDescription, FMRadioStations, ForeignExchangeReserves, ForeignOwnedShips, ForeignRegisteredShips, FullCoordinates, FullName, FullNativeName, FullPolygon, GDP, GDPAtParity, GDPPerCapita, GDPRealGrowth, GDPSectorFractions, GiniIndex, GovernmentConsumption, GovernmentDebt, GovernmentExpenditures, GovernmentReceipts, GovernmentSurplus, GrossInvestment, Groups, HighestElevation, HighestPoint, HIVAIDSDeathRateFraction, HIVAIDSFraction, HIVAIDSPopulation, HouseholdConsumption, ImportCommodities, ImportPartners, ImportPartnersFractions, ImportValue, IndependenceDate, IndependenceYear, IndustrialProductionGrowth, IndustrialValueAdded, InfantMortalityFraction, InfectiousDiseases, InflationRate, InternationalOrganizations, InternationalOrganizationsObserver, InternetCode, InternetHosts, InternetUsers, InventoryChange, IrrigatedLandArea, IrrigatedLandFraction, ISOName, LaborForce, LandArea, Languages, LanguagesDialects, LanguagesFractions, LargestCities, LifeExpectancy, LiteracyFraction, LowestElevation, LowestPoint, MajorIndustries, MajorPorts, MaleAdultPopulation, MaleChildPopulation, MaleElderlyPopulation, MaleInfantMortalityFraction, MaleLifeExpectancy, MaleLiteracyFraction, MaleMedianAge, MalePopulation, ManufacturingValueAdded, MaritimeClaims, MedianAge, Memberships, MerchantShips, MerchantShipsDeadWeight, MerchantShipsGross, MerchantShipTypes, MigrationRateFraction, MilitaryAgeFemales, MilitaryAgeMales, MilitaryAgePopulation, MilitaryAgeRate, MilitaryExpenditureFraction, MilitaryExpenditures, MilitaryFitFemales, MilitaryFitMales, MilitaryFitPopulation, MiscellaneousValueAdded, Name, NationalIncome, NationalityName, NativeName, NaturalGasConsumption, NaturalGasExports, NaturalGasImports, NaturalGasProduction, NaturalGasReserves, NaturalHazards, NaturalResources, OilConsumption, OilExports, OilImports, OilProduction, OilReserves, PavedAirportLengths, PavedAirports, PavedRoadLength, PhoneLines, Pipelines, Polygon, Population, PopulationGrowth, PovertyFraction, PriceIndex, RadioStations, RailwayGaugeLengths, RailwayGaugeRules, RailwayLength, RegionNames, Regions, Religions, ReligionsFractions, RoadLength, SchematicCoordinates, SchematicPolygon, SectorLaborFractions, Shape, ShortWaveRadioStations, SignedEnvironmentalAgreements, StandardName, SuffrageType, TelevisionStations, TerrainTypes, TimeZones, TotalConsumption, TotalFertilityRate, TradeValueAdded, TransportationValueAdded, UNCode, UnemploymentFraction, UNNumber, UnpavedAirportLengths, UnpavedAirports, UnpavedRoadLength, ValueAdded, WaterArea, WaterwayLength}

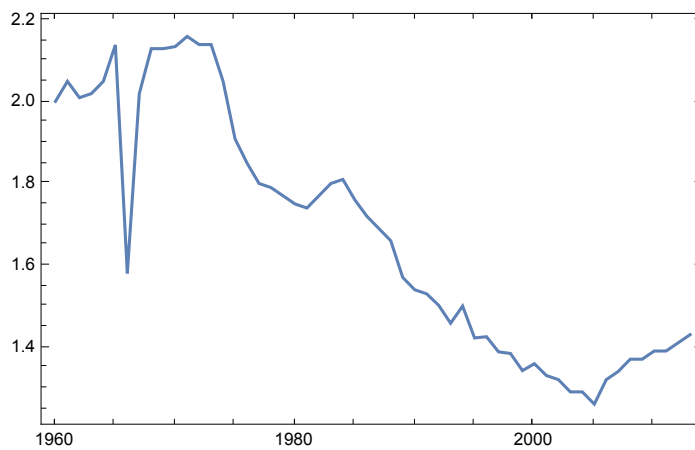
"Total fertility rate" TFR is the averaged number of children a woman has in her life. TFR has to be larger than 2 for a population to sustain itself. We will later learn about this fact in this course.

**CountryData["Japan", "TotalFertilityRate"]**

1.43 people/person



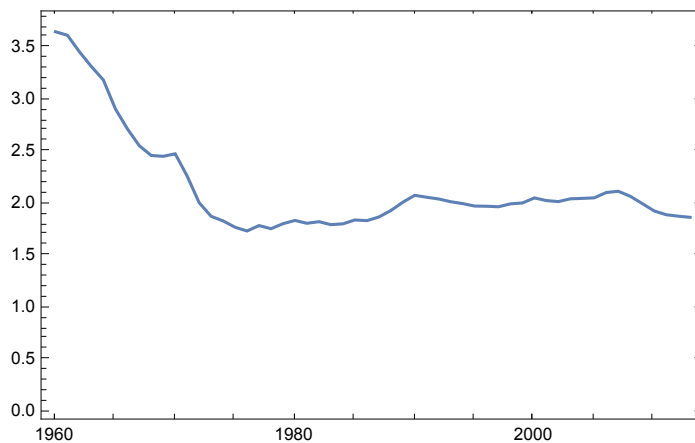
```
DateListPlot[CountryData["Japan", {"TotalFertilityRate", All}]]
```



```
CountryData["USA", "TotalFertilityRate"]
```

1.8695 people/person

```
DateListPlot[CountryData["USA", {"TotalFertilityRate", All}]]
```



```
CountryData["Japan", "LifeExpectancy"]
```

83.58 yr

```
CountryData["USA", "LifeExpectancy"]
```

78.941 yr

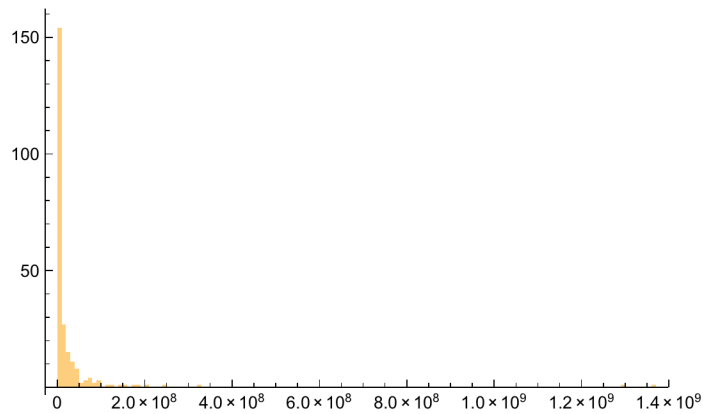
## Distribution of the populations by country

Let's see which country has how many people.

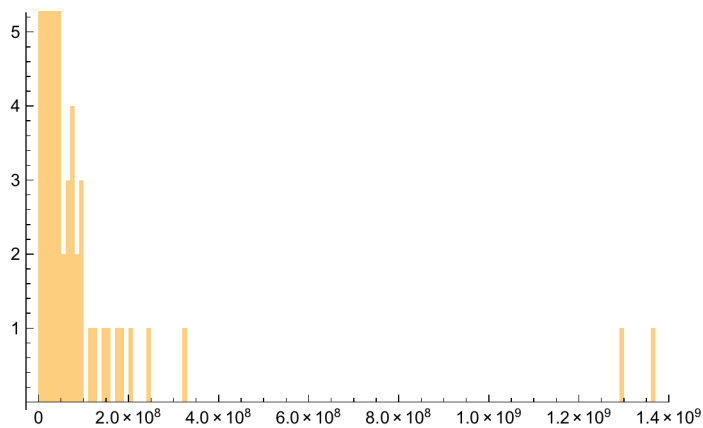
```
dataPopSizes = CountryData["Countries", "Population"];
```

Let's draw "histogram" of the populations of each country.

```
Histogram[dataPopSizes, PlotRange -> All]
```



```
Histogram[dataPopSizes, PlotRange -> {All, {0, 5}}]
```



China is top ranked and India is the next.

```
CountryData["China", "Population"]
```

1 364 773 138 people

```
CountryData["India", "Population"]
```

1 291 780 156 people

```
CountryData["USA", "Population"]
```

322 422 965 people

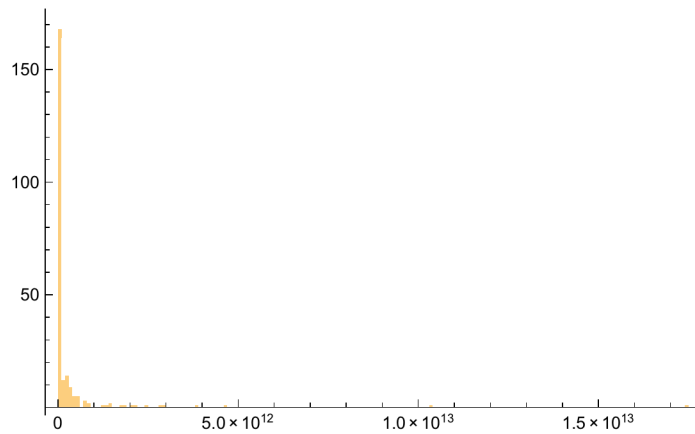
List the top ten countries in terms of the population size.

```
dataName = CountryData["Countries", "Name"];
Take[SortBy[Transpose[{dataName, dataPopSizes}], Last] // Reverse, 10]
{{China, 1 364 773 138 people }, {India, 1 291 780 156 people },
 {United States, 322 422 965 people }, {Indonesia, 249 563 467 people },
 {Brazil, 201 700 544 people }, {Pakistan, 186 428 686 people },
 {Nigeria, 175 288 238 people }, {Bangladesh, 156 380 192 people },
 {Russia, 142 400 066 people }, {Japan, 126 225 259 people }}
```

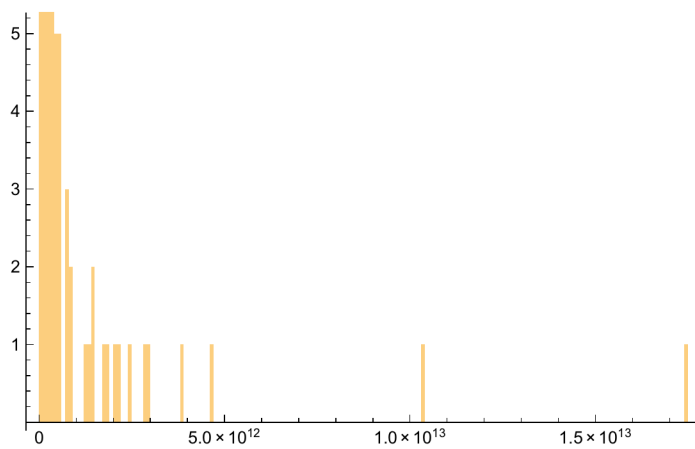
How about GDP?

```
dataGDP = CountryData["Countries", "GDP"];
```

**Histogram[dataGDP, PlotRange -> All]**



**Histogram[dataGDP, PlotRange -> {All, {0, 5}}]**



**dataName = CountryData["Countries", "Name"];**

**data = {dataName, dataGDP} // Transpose;**

**Take[SortBy[data, Last] // Reverse, 10]**

```
{ {United States, $1.7419 × 1013 per year }, {China, $1.03548 × 1013 per year },
  {Japan, $4.60146 × 1012 per year }, {Germany, $3.86829 × 1012 per year },
  {United Kingdom, $2.98889 × 1012 per year }, {France, $2.82919 × 1012 per year },
  {Brazil, $2.41664 × 1012 per year }, {Italy, $2.14116 × 1012 per year },
  {India, $2.04852 × 1012 per year }, {Russia, $1.8606 × 1012 per year } }
```

## Examples just for fun!

```
CountryData["World", "Shape"]
```



## Which country is member of G7, Groupe of Seven?

```
g7 = CountryData["G7"]
```

```
{Entity[Country, Canada], Entity[Country, France],  
 Entity[Country, Germany], Entity[Country, Italy], Entity[Country, Japan],  
 Entity[Country, UnitedKingdom], Entity[Country, UnitedStates]}
```

```
CountryData[#, "Population"] & /@g7
```

```
{35 309 555 people, 64 101 308 people, 81 625 599 people,  
 61 175 248 people, 126 225 259 people, 63 556 184 people, 322 422 965 people}
```

```
CountryData[#, "GDP"] & /@g7
```

```
{$1.78539 × 1012 per year, $2.82919 × 1012 per year,  
 $3.86829 × 1012 per year, $2.14116 × 1012 per year,  
 $4.60146 × 1012 per year, $2.98889 × 1012 per year, $1.7419 × 1013 per year}
```

```
CountryData[#, "PopulationGrowth"] & /@g7
```

```
{0.00920091 people/(person yr),  
 0.00512445 people/(person yr), -0.00227597 people/(person yr),  
 0.00202168 people/(person yr), -0.000707227 people/(person yr),  
 0.0060401 people/(person yr), 0.00969061 people/(person yr)}
```

```
CountryData[#, "Area"] & /@g7
```

```
{9.98467 × 106 km2, 551 500. km2, 357 022. km2,  
 301 340. km2, 377 835. km2, 243 610. km2, 9.63142 × 106 km2}
```

## Draw country flags in proportion to the population size

```
cdata = CountryData["Countries", "Population"];  
fdata = CountryData["Countries", "Flag"];
```

```
list = Thread[cdata → fdata];  
th = Total[list[[All, 1]]] / 500;  
list2 = Select[list, #[[1]] > th &];  
ImageCollage[list2, ImagePadding → 2, Background → GrayLevel[.8], ImageSize → 500]
```

