This assignment consists of making graphical companions to some of the data aggregation tasks .Again, we are working with the Gapminder data (available [here](http://www.stat.ubc.ca/~jenny/notOcto/STAT545A/examples/gapminder/data/gapminderDataFiveYear.txt)).

First, we load the necessary libraries:

library(lattice)

library(plyr)

library(xtable)

Then we import the Gapminder data into the data.frame gDat:

gDat <- read.delim("gapminderDataFiveYear.txt")

A quick check that the data import went well can be performed by str(), summary(), and head() functions:

str(gDat)

## 'data.frame': 1704 obs. of 6 variables:

## $ country : Factor w/ 142 levels "Afghanistan",..: 1 1 1 1 1 1 1 1 1 1 ...

## $ year : int 1952 1957 1962 1967 1972 1977 1982 1987 1992 1997 ...

## $ pop : num 8425333 9240934 10267083 11537966 13079460 ...

## $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 3 3 3 3 3 3 3 3 3 ...

## $ lifeExp : num 28.8 30.3 32 34 36.1 ...

## $ gdpPercap: num 779 821 853 836 740 ...

summary(gDat)

## country year pop continent

## Afghanistan: 12 Min. :1952 Min. :6.00e+04 Africa :624

## Albania : 12 1st Qu.:1966 1st Qu.:2.79e+06 Americas:300

## Algeria : 12 Median :1980 Median :7.02e+06 Asia :396

## Angola : 12 Mean :1980 Mean :2.96e+07 Europe :360

## Argentina : 12 3rd Qu.:1993 3rd Qu.:1.96e+07 Oceania : 24

## Australia : 12 Max. :2007 Max. :1.32e+09

## (Other) :1632

## lifeExp gdpPercap

## Min. :23.6 Min. : 241

## 1st Qu.:48.2 1st Qu.: 1202

## Median :60.7 Median : 3532

## Mean :59.5 Mean : 7215

## 3rd Qu.:70.8 3rd Qu.: 9325

## Max. :82.6 Max. :113523

##

head(gDat, 10)

## country year pop continent lifeExp gdpPercap

## 1 Afghanistan 1952 8425333 Asia 28.80 779.4

## 2 Afghanistan 1957 9240934 Asia 30.33 820.9

## 3 Afghanistan 1962 10267083 Asia 32.00 853.1

## 4 Afghanistan 1967 11537966 Asia 34.02 836.2

## 5 Afghanistan 1972 13079460 Asia 36.09 740.0

## 6 Afghanistan 1977 14880372 Asia 38.44 786.1

## 7 Afghanistan 1982 12881816 Asia 39.85 978.0

## 8 Afghanistan 1987 13867957 Asia 40.82 852.4

## 9 Afghanistan 1992 16317921 Asia 41.67 649.3

## 10 Afghanistan 1997 22227415 Asia 41.76 635.3

From the results we see that we have to do with a data.frame with 1704 observations of 6 variables. The variables are: country, year, pop (population), continent, lifeExp (life expectancy), and gdpPercap (GDP per capita). We also get a summary of some statistical properties of each variable in the data. Finally, we see a little snippet of the data.frame itself. It appears that the data import went well.

Finally, we create a function that easens the task of converting and printing data to an HTML table.

htmlPrint <- function(x, ..., digits = 0, include.rownames = FALSE) {

print(xtable(x, digits = digits, ...), type = "html", include.rownames = include.rownames,

...)

}

**Change of life expectancy over time for different continents**

This part considers the temporal change of life expectancy for the different continents. Each continent contains a number of countries, each with its own life expectancy data. Obtaining a measure of the life expectancy for each continent can then be done by averaging the life expectancies for each of its constituent countries. Doing this for each year gives us the desired results. The last line in the following chunk can be uncommented to show the table of results. But be warned that it is a long table! :-)

avgLifeExpByContAndYear <- ddply(gDat, ~continent + year, summarize, avgLifeExp = mean(lifeExp))

# htmlPrint(avgLifeExpByContAndYear)

Let us now try to visualize this data. What we want is the average life expectancy as a function of time–and this we want for each of the continents. We could therefore attempt a multi-panel plot as follows. A line between the data points has also been added via the type = argument.

xyplot(avgLifeExp ~ year | continent, avgLifeExpByContAndYear, type = c("p",

"smooth"))

We can also show the graphs for the five continents in the same plot, using different colors (rather than panels as above) to distinguish them. Again, we add lines through the data points via the type = argument. Furthermore, a key, or legend, is added to the plot via auto.key = TRUE to see which graphs and continents correspond to each other. By default in this case, the order in the key is not the same as the plots. This can be remedied by reversing the rows as done in the code chunk below. (However, it is not perfectly remedied by this solution, since Asia and Americas are still switched around in the key compared to the plot order. But the colors still match.)

xyplot(avgLifeExp ~ year, avgLifeExpByContAndYear, groups = continent, auto.key = list(reverse.rows = TRUE),

type = c("p", "smooth"))

Both of the above figures give a very nice visual representation of how the temportal change in the average life expectancy for the different countries. Compared to the woefully long table (that can be seen by uncommenting a line of code in an earlier chunk) comprising the same data, these figures are just lovely to look at! For comparative purposes, the latter figure provides the better overview. First, we notice how Africa started off with a good, positive slope–indicating a larger increase in average life expectancy per year. However, the trend line indicates that this slope is less positive in the more recent years. We actually observe the opposite for Oceania; starting off with a very small but positive slope, the graph then shows a slightly larger increase. Europe seems to have kept a relatively constant increase in life expectancy per year, as evident from the lack of curvature in the graph. Asia and Americas appear to follow each other rather well in the sense that their slopes–i.e. the change in life expectancy per year–are almost the same. They are merely set apart by a little gap, indicating that the Americas has a larger life expectancy than Asia. Looking less roughly at the graphs, however, it appears that Asia might be slowly catching on; the difference between Asia and America is smaller in 2007 than in 1952.

**Number and proportion of countries with low/high life expectancy over time by continent**

Above, we looked at the temporal change in the life expectancy for the different continents. We can dig deeper into the state, or condition, of the different continents by examining how many of the constituent countries have a life expectancy higher or lower than a certain threshold. First, we define the threshold. This could be e.g. the mean or another quantile of the worldwide life expectancy, or simply a chosen number. Afterwards, we save a data.frame with the desired data which is obtained by summing the occurrences for which a country has a life expectancy lower than or equal to the threshold. This is done in the chunk below.

# lifeExpThreshold = mean(gDat$lifeExp)

lifeExpThreshold = 50

nCountriesWithLowLifeExpOverTime <- ddply(gDat, ~continent + year, function(x) c(nCntryLowLE = sum(x$lifeExp <=

lifeExpThreshold)))

# htmlPrint(avgLifeExpByContAndYear)

Uncommenting the last line of code shows a table of the data. But again, be warned that this is a long table. We are much more interested in getting a visual representation of the data it contains!

# xyplot(nCntryLowLE ~ year, nCountriesWithLowLifeExpOverTime, groups =

# continent, auto.key = TRUE, type = c('p', 'smooth')) # returns a

# warning/error message on top of the plot.

xyplot(nCntryLowLE ~ year, nCountriesWithLowLifeExpOverTime, groups = continent,

auto.key = TRUE)

Including the type = argument resulted in a kind of warning message overlayed on the plot in this case. Maybe that has to do with some of the data entries being zero. In any case, one can almost make out the trends oneself. Now, in the figure above, the vertical axis is the number of countries (in a particular continent) with a life expectancy lower than the threshold of 50. What springs to eye is the amount of African countries that have life expectancies lower than 50.

Fortunately, the trend is decreasing; there are less and less countries with such low life expectancies. The trends for Asia and the Americas are also decreasing, although from an initial value much smaller than in the case of Africa. Towards the end of the 1980s, both Asia and the Americas have almost no countries with a life expectancy lower than 50. As for Europe and Oceania, these two continents have stayed almost the entire time at a constant level of zero countries with a life expectancy lower than lifeExpThreshold.

Now, instead of looking at an absolute quantity such as the number of countries, we can also consider a relative quantity such as the proportion of countries with a life expectancy lower than a certain threshold–with the proportion being taking with respect to the total number of countries for the particular continent. The code chunk below computes both the absolute count (as above) and the relative proportion.

propCntryLowLifeOverTime <- ddply(gDat, ~continent + year, function(x) {

jCount = sum(x$lifeExp <= lifeExpThreshold)

c(count = jCount, prop = jCount/nrow(x))

})

# htmlPrint(propCntryLowLifeOverTime, digits = c(0, 0, 0, 0, 2)) # quite a

# long table.

Again, the resulting table of data is rather long, and therefore the line of code is commented out. We would much rather visualize the data! This is done below.

xyplot(prop ~ year, propCntryLowLifeOverTime, groups = continent, auto.key = TRUE)

The vertical axis is the proportion of countries with a life expectancy lower than 50 (relative to the total number of countries in the continent). The results–and the conclusions that can be drawn from them–are basically the same as for the previous plot. It is just nicer and more useful to show a relative quantity such as the proportion rather than an absolute one. One difference between the plots is also that Asia and the Americas are closer to Africa in the beginning.

**Max, min, and spread of GDP per capita for the different continents**

Steering away from the life expectancy, let us now look at the GDP per capita for the different continents. More specifically, we will consider how certain statistical properties–max, min, and spread–of the GDP per capita have evolved over time. This will be done for each continent. Code-wise, the most straightforward–and easy–thing to do would be to consider just a single year. However, regarding the year variable as a factor allows us to get a sense of the temporal evolution. This is done in the code chunk below by use of the bwplot command. This produces a so-called “box-and-whiskers” plot. We have excluded Oceania from the resulting figure, as there is really nothing interesting to see for that continent.

# bwplot(gdpPercap ~ as.factor(year) | continent, gDat)

bwplot(gdpPercap ~ as.factor(year) | continent, subset(gDat, continent != "Oceania"))

The “box-and-whiskers” plot provides information about the minimum, maximum, mean/median, and the interquartile range.

We notice that Africa is rather poor, with only a few outlying richer countries. The trend for the Americas seems quite flat. However, the maximum and the spread seem to be increasing through time. Also, the outlying richer countries have steadily become even richer. The trend for Europe is a steady increase. Moreover, the range has also grown with time; the maxima and interquartile ranges have increased. Thus, while the European countries have become richer, the difference between rich and less rich has also increased. What stands out for Europe compared to the other continents is also the fact that there aren't any real outliers. There are plenty of outliers for Asia, though–and very extreme ones, even. Evidently, there are, or have been, some really rich countries there. But this was mostly in the beginning and middle time periods considered; for later years, these outliers are not really present. However, the trend is still an expanding spread–i.e. larger differences between rich and poor–as well as increasing maxima.

**Population over time for each continent**

Let's take a quick look at the population over time for the different continents. Here, we compute the average population for each continent, with the average being taken over all of the constituent countries.

avgPopByContAndYear <- ddply(gDat, ~continent + year, summarize, avgPop = mean(pop))

xyplot(avgPop ~ year, avgPopByContAndYear, groups = continent, auto.key = TRUE,

type = c("p", "smooth"))

Now, this is okay in its own right. But we are only looking at the five different continents - there might be interesting things to discover on a more country-wide scale, and these wholly hidden from us right now.

**Interesting countries with respect to population**

Let us therefore spice things up a bit by making a linear model of population versus time.

yearMin <- min(gDat$year)

lmFun <- function(x) {

estCoefs <- coef(lm(pop ~ I(year - yearMin), x))

names(estCoefs) <- c("intercept", "slope")

return(estCoefs)

}

Passing this function as an input to the ddply call gives us the model coefficients for each country:

lmCoefsForCountries <- ddply(gDat, ~country + continent, lmFun)

str(lmCoefsForCountries)

## 'data.frame': 142 obs. of 4 variables:

## $ country : Factor w/ 142 levels "Afghanistan",..: 1 2 3 4 5 6 7 8 9 10 ...

## $ continent: Factor w/ 5 levels "Africa","Americas",..: 3 4 1 1 2 5 4 3 3 4 ...

## $ intercept: num 6009355 1328288 6869878 3340306 17109890 ...

## $ slope : num 356886 45526 472928 144330 417904 ...

This is a rather long data.frame consisting of 142 observations (corresponding to the 142 countries) of 4 variables. The variables are: country, continent, intercept and slope. The latter two are the model coefficients.

Now we have a variety of interesting options. For instance, we can see which country in each of the five continents had the largest population in 1952 (i.e. the largest intercept):

largestIntercepts <- ddply(lmCoefsForCountries, ~continent, function(x) {

theMax <- which.max(x$intercept)

x[theMax, c("continent", "country", "intercept", "slope")]

})

htmlPrint(largestIntercepts, digits = c(0, 0, 0, 0, 2))

|  |  |  |  |
| --- | --- | --- | --- |
| **continent** | **country** | **intercept** | **slope** |
| Africa | Nigeria | 22927507 | 1846564.01 |
| Americas | United States | 158463562 | 2536278.92 |
| Asia | China | 556263528 | 14614419.05 |
| Europe | Germany | 71315923 | 226586.20 |
| Oceania | Australia | 8678698 | 217113.25 |

We can dig further down into these five countries by plotting their population versus time. This is done by use of the subset argument in the plotting command:

xyplot(pop ~ year | country, gDat, subset = country %in% largestIntercepts$country,

type = c("p", "r"))

China must be said to stand out!

It might be more interesting to consider the slope. Let's see which countries have the had the largest population changes over time:

largestSlopes <- ddply(lmCoefsForCountries, ~continent, function(x) {

theMax <- which.max(x$slope)

x[theMax, c("continent", "country", "intercept", "slope")]

})

htmlPrint(largestSlopes, digits = c(0, 0, 0, 0, 2))

|  |  |  |  |
| --- | --- | --- | --- |
| **continent** | **country** | **intercept** | **slope** |
| Africa | Nigeria | 22927507 | 1846564.01 |
| Americas | United States | 158463562 | 2536278.92 |
| Asia | China | 556263528 | 14614419.05 |
| Europe | Turkey | 20512157 | 923521.86 |
| Oceania | Australia | 8678698 | 217113.25 |

And now, let's visualize the population through time for these “interesting” countries:

xyplot(pop ~ year | country, gDat, subset = country %in% largestSlopes$country,

type = c("p", "r"))

Again, China really stands out with not only a huge population but also a massive population increase over time. But one might wonder what is up with e.g. Australia. It seems like they have no population at all. But this is far from true, as evident by the plot below:

xyplot(pop ~ year | country, gDat, subset = country == "Australia", type = c("p",

"r"))

Australia has a fairly large population as well as a very constant slope. What made Australia in particular look very odd in the other figures was the fact that its population was completely dwarfed by that of China.

Now, instead of looking at the slopes and intercepts, we could consider the *residuals* in the linear model. More specifically, we will examine which countries have the largest residuals in the linear model of population versus time. The code chunk below defines the function to do this, saves the data in a data.frame, and presents it in a table.

lmFun2 <- function(x) {

lmFit <- lm(pop ~ I(year - yearMin), x)

lmCoef <- coef(lmFit)

names(lmCoef) <- NULL

return(c(intercept = lmCoef[1], slope = lmCoef[2], maxResid = max(abs(resid(lmFit)))/summary(lmFit)$sigma))

}

maxResids <- ddply(gDat, ~country + continent, lmFun2)

maxResiduals <- ddply(maxResids, ~continent, function(x) {

theMax <- which.max(x$maxResid)

x[theMax, ]

})

htmlPrint(maxResiduals, digits = c(0, 0, 0, 2, 2, 2))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **country** | **continent** | **intercept** | **slope** | **maxResid** |
| Equatorial Guinea | Africa | 167939.27 | 5804.06 | 2.25 |
| Dominican Republic | Americas | 2244248.40 | 126961.78 | 2.21 |
| Lebanon | Asia | 1575270.94 | 43875.47 | 2.26 |
| Portugal | Europe | 8568161.01 | 37022.29 | 2.31 |
| Australia | Oceania | 8678698.12 | 217113.25 | 1.88 |

Let's visualize it!

xyplot(pop ~ year | country, gDat, subset = country %in% maxResiduals$country,

type = c("p", "r"))

The countries are different from the ones that appeared when considering the largest slopes and intercepts. However, funnily enough, Australia is still there. :-) Admittedly, though, there is not much to see here–at least when one is looking for extreme outliers and such. There might be much more interesting things popping up if one were to consider e.g. the life expectancy or GDP per capita. But then it's great that the above code, formulas, functions, etc.–the entire framework–carries over nicely, and one would just have to replace the instance of the population, pop, with those of e.g. the GDP per capita, gdpPercap.