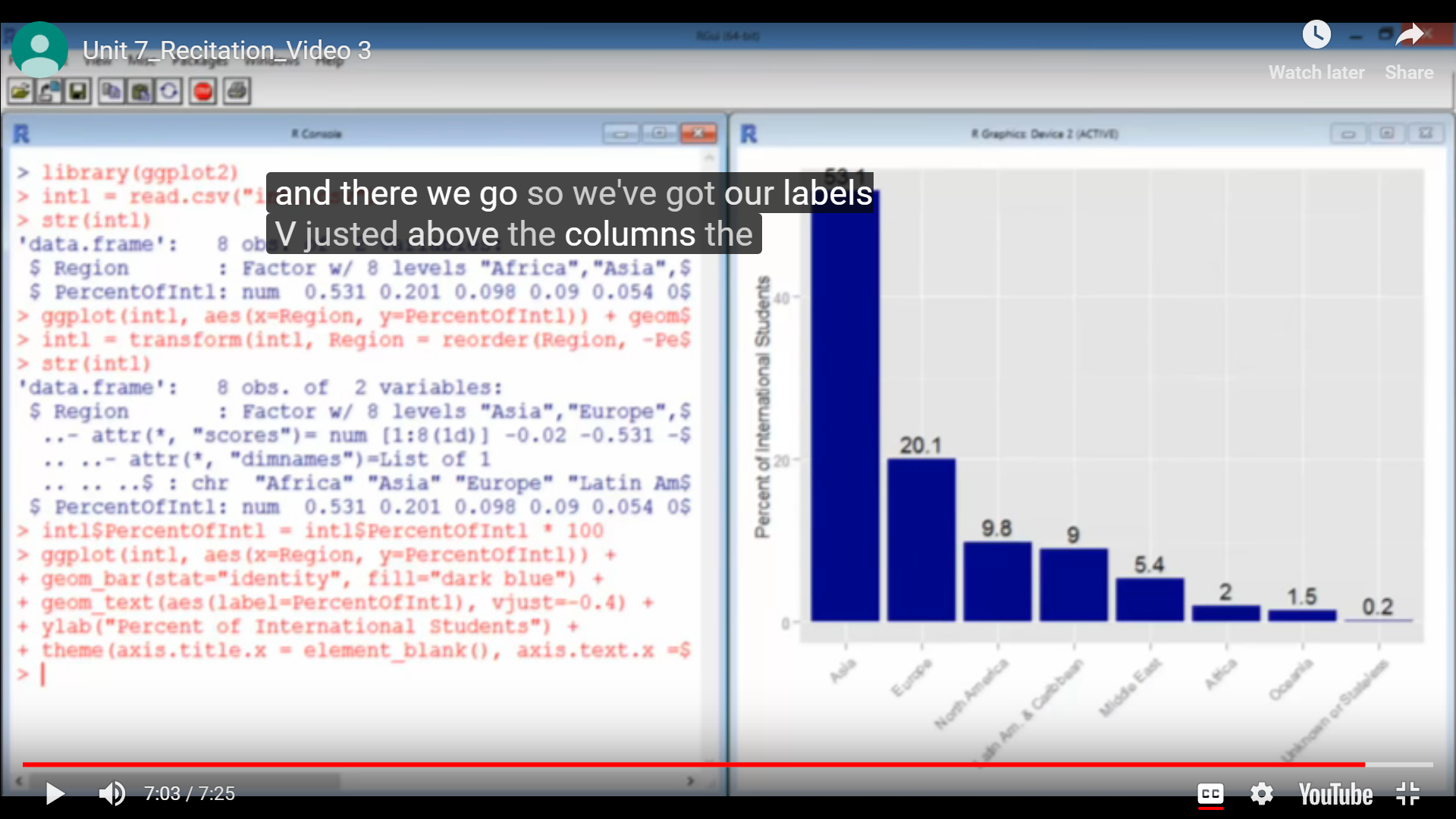
* **VISUALISATION:**
  + Mapping of numerical or categorical data properties to visual properties like co-ordinates, colors, sizes.
* We use **ggplot2**, a plotting system in R, based on grammer of graphics that provides a powerful model of graphics that makes it easy to produce complex multi-layered graphics.
* In basic R plots, graphics contains simple points ans lines an it is difficult to add elements to existing plots.
* But in ggplot, mappings are done by adding layers to the plot which makes easier to add new elements to existing plots.
* Ggplot consist of atleast 3 elements:
  + **DATA**, in a data frame
  + **Aesthetic mapping** describing how variables in dataframe are mapped to graphical attributes.
    - Ex: color, shape, scale
  + **Geometric objects** determine how values are rendered graphically.
    - EX: points, boxplots, bars, polygons
* install.packages(“ggplot2” )
* library(ggplot2)
* Scatterplot = ggplot(WHO, aes(x=GNT, y= FertilityRate))
  + WHO ⇒ dataframe name
  + Aes ⇒ aethetic mapping
* Now we can plot differnet plots using same ggplot object.
* Scatterplot + geom\_point() ⇒ gives the graphs with points.
* Scatterplot + geom\_line() ⇒ gives the graphs with line
* Scatterplot + geom\_point(color = “ blue” , size = 3, shape = 17) + ggtitle(“ FertilityRateVS GNT”)
  + Size ⇒ the size of the points
  + Shape = 17 ⇒ the number correspoinding to traingle. so, it gives the graphs with traiangle points istead of black dots.
* To get the colors according to the values in the region variable:
  + ggplot(WHO, aes(x = GNT, y =FertilityRate, color=Region)) + geom\_point()
* Model = lm(Under15 ~ log(FertitlityRate), data= WHO)
* To add regression model line in the plot
  + ggplot(WHO, aes(x = log(FertilityRate), y = Under15)) + geom\_point() + stat\_smooth(method=”lm”)
* We can add the confidence level to the regression line by:
  + ggplot(WHO, aes(x = log(FertilityRate), y = Under15)) + geom\_point() + stat\_smooth(method=”lm” + level = 0.99)
  + And to remove confidence level:
  + ggplot(WHO, aes(x = log(FertilityRate), y = Under15)) + geom\_point() + stat\_smooth(method=”lm” + se = FALSE)
* **HEATMAPS**  are a way of visualising dat using three attributes
  + X-axis
  + Y-axis
  + Shades of colors
  + We can use siffernet color schemes based on type of data to covey different messages.
  + X-axis ang y-axis no need to be continuous. It can be categorical
* We can convert the date into the format that R can understand and can extraxt day and month seperately using:
  + mvt$Date = strptime(mvt$Date, format = “%m/%d/%y %H:%M”)
  + mvt$Weekday = weekdays(mvt$Date)
  + mvt$Hour = mvt$Date$hour
* To get the values in the vector in order:
  + weekdayCounts$Var1 = factor(WeekdayCOunts$Var1, ordered = TRUE, levels = c(“Monday”, “Tuesday”, “Wednesday”, “Thursday”, “Friday” , “Saturday” , “Sunday”))
* ggplot(WeekdayCounts, aes(x=Var1, y = Freq)) + geom\_line(aes(group=1)) + Xlab= “days” + Ylab=”frequency”
  + geom\_line(aes(group=1)) ⇒ tells that to give a single line for all groups instead of multiple lines.
* To set new color shade to the heat map plot:
  + ggplot(DayHourCounts, aes(x=Hour, y=Var1)) + geom\_tile(aes(fill = Freq)) + scale\_fill\_gradient(name = “Total Mv Tefts”, low = “white , high = “red”) + theme(axis.title.y = element\_blank())
    - Geom\_tile ⇒ method for heat map
    - scale\_fill\_gradient(name = “Total Mv Tefts”, low = “white , high = “red”) ⇒ to set the title and color shade for gradient table
* **MAPS:**
  + install.packages(“ maps” )
  + library(maps)
  + install.packages(“ ggmap” )
  + library(ggmap)
  + Chicago = get\_map(location=” chicago” , zoom = 11)
  + ggmap(chicago) ⇒ to display the geographial map of chicago city.
  + ggmap(chicago) + geom\_point(dta=mvt[1:100,], aes(x=Longitude, y = Latitude)) ⇒ map of chicago with black points of the locations in the dataset observations.
* statesMap = map\_data(“ state” )==> to get US map
* murder$region = tolower(murder$State)
* murderMap = merge(stateMap, murders, by=” region” )
* ggplot(murderMap, aes(x=long, y = lat, group = group, fill = murders)) + geom\_polygon(color = “ black” ) + scale\_fill\_gradient(low=” black, high=” red”, guide = legend)
* **CONS in PIE:**
  + Not all points can be labeled, so data is lost
  + Colors are meaningless, are close enough to be confused
  + 3D add nothing than larger visible volume.
* **BAR GRAPHS:**
  + ggplot(dataframe, aes(x=Region, y = PercentOfInt1)) + geom\_bar(stat=” identity” ) + geom\_text(aes(label=PercentOfInt1))
  + Geom\_text ⇒ to give the label name for each bar in the graph
  + stat=” identity” ⇒ it says that use the value of Y variable as is.
* We can thranform the unordered varible into ordered using:
  + Dataframe = transform(dataframe, Region = reorder(Region, -PercentOfInt1))
    - Heere negative sign is to indicate decreasing order



* To get the world map:
  + World\_map = map\_data(“ world” )
* To merge the world map with the dataset using country as a key
  + World\_map = merge(world\_map, int1all, by.x=” region” , by.y=” citizenship” )
    - By.x = region ⇒ countries are present in region variale in world map
    - By.y = citizenship ⇒ countries are present in citizenship in int1all dataset
* World\_map = world\_map[order(world\_map$group, world\_map$order), ]
* ggplot(world\_map, aes(x=Long, y = Lat, group=group)) + geom\_polygon(fill=” white” , color =” black”) + coord\_map(“ mercator” )

