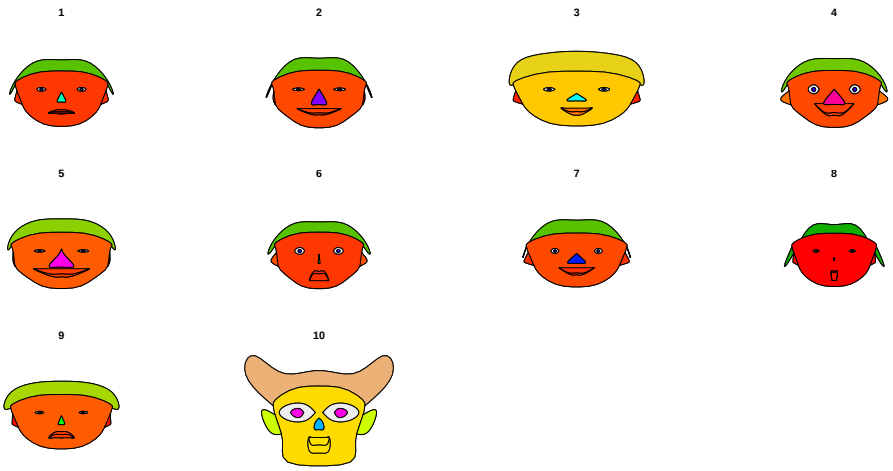
Chernoff Faces

- A method of comparing many (up to 17!) quantitative variables at once.
- The goal of Chernoff faces is to show a bunch of variables at once via facial features like lips, eyes, and nose size.
- Most of the time there can be a reduction in number of variables being visualized, but the faces can be interesting to work with.
- Most of the time there are better solutions, but sometimes this might be worth trying.
- The idea behind using faces is that humans easily recognize faces and notice small changes without difficulty. Chernoff faces handle each variable differently. Because the features of the faces vary in perceived importance, the way in which variables are mapped to the features should be carefully chosen (e.g. eye size and eyebrow-slant have been found to carry significant weight)
- There are 15 facial features we can use to visualize variables, additionally we can position them on the xy plane to visualize two more quantitative variables
- The `aplpack` package will be used to make these faces. From my experience, the package will install easily (like any other package) on a PC. If you are using a Mac, it is expected you will run into errors since the package has not been updated in a few years. I have been able to debug this issue on some macs, but it is likely not worth your trouble. Use a lab machine for this.

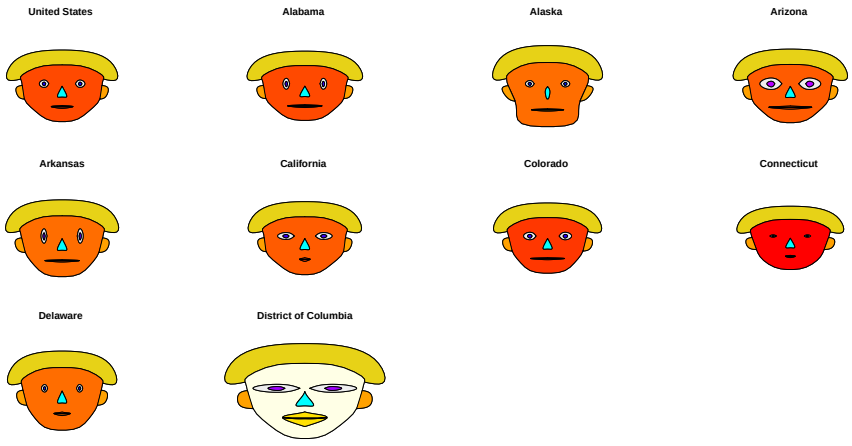
```
require(aplpack)
crime <- read.csv("http://datasets.flowingdata.com/crimeRatesByState-formatted.csv")
faces(crime[1:10, 2:8], face.type = 1)
```



```
## effect of variables:
##  modified item       Var
##  "height of face   " "murder"
##  "width of face    " "forcible_rape"
##  "structure of face" "robbery"
##  "height of mouth  " "aggravated_assault"
##  "width of mouth   " "burglary"
##  "smiling          " "larceny_theft"
##  "height of eyes   " "motor_vehicle_theft"
##  "width of eyes    " "murder"
##  "height of hair   " "forcible_rape"
##  "width of hair    "  "robbery"
##  "style of hair    "  "aggravated_assault"
##  "height of nose   "  "burglary"
##  "width of nose    "  "larceny_theft"
##  "width of ear     "  "motor_vehicle_theft"
##  "height of ear    "  "murder"
```

- Manipulate the data so that we do not have creepy smiling faces to represent larceny theft. In order to have a facial feature be unchanged, we need a variable that is constant among observational units. I am calling that variable 'neutral' and setting it equal to 0 for everyone.

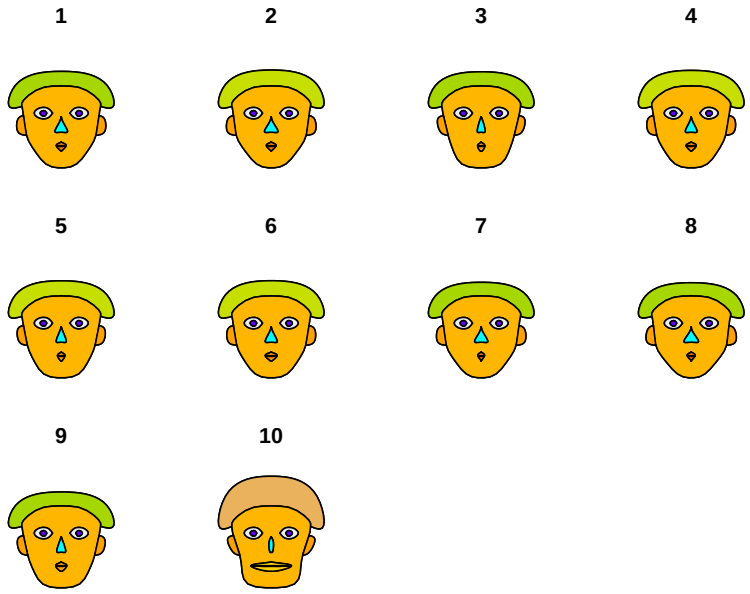
```
require(mosaic)
crime2<-mutate(crime, neutral=0)

##Re-order variables so that each variable I want to
## graph is ordered according to the facial feature
## I want it to correspond to.
## Further, let's say I only want to look at the first 10
## observations.

faces(crime2[1:10,c("aggravated_assault", "murder",
               "forcible_rape", "robbery", "larceny_theft", "neutral",
               "burglary", "motor_vehicle_theft", "neutral",
               "neutral", "neutral", "neutral", "neutral", "neutral")],
      labels=crime2$state[1:10])
```




```
## effect of variables:
##  modified item       Var
##  "height of face   " "aggravated_assault"
##  "width of face    " "murder"
##  "structure of face" "forcible_rape"
##  "height of mouth  " "robbery"
##  "width of mouth   " "larceny_theft"
##  "smiling          " "neutral"
##  "height of eyes   " "burglary"
##  "width of eyes    " "motor_vehicle_theft"
##  "height of hair   " "neutral.1"
##  "width of hair    "  "neutral.2"
##  "style of hair    "  "neutral.3"
##  "height of nose   "  "neutral.4"
##  "width of nose    "  "neutral.5"
##  "width of ear     "  "neutral.6"
##  "height of ear    "  "neutral.7"
```

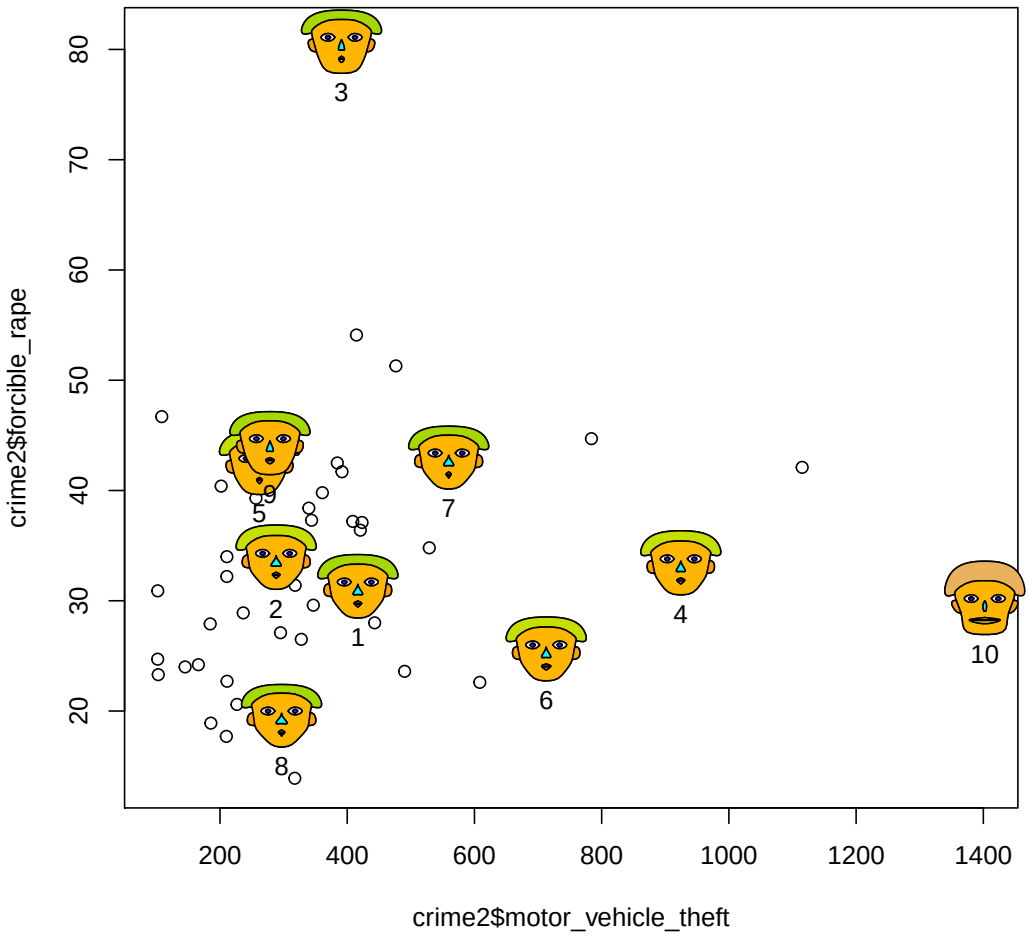
- We can incorporate in 2 more quantitative variables by making use of the position of the face. Suppose I want the structure of face to be aggravated assault, width of mouth to be robbery, height of hair to be murder. I then want to use motor vehicle theft on the x-axis and forcible rape on the y-axis.

```
a<-faces(crime2[1:10,c("neutral", "neutral",
                "aggravated_assault", "neutral", "robbery", "neutral",
                "neutral", "neutral", "murder",
                "neutral", "neutral", "neutral", "neutral", "neutral", "neutral")])
```




```
## effect of variables:
##  modified item       Var
##  "height of face   " "neutral"
##  "width of face    " "neutral.1"
##  "structure of face" "aggravated_assault"
##  "height of mouth  " "neutral.2"
##  "width of mouth   " "robbery"
##  "smiling          " "neutral.3"
##  "height of eyes   " "neutral.4"
##  "width of eyes    " "neutral.5"
##  "height of hair   " "murder"
##  "width of hair    "  "neutral.6"
##  "style of hair    "  "neutral.7"
##  "height of nose   "  "neutral.8"
##  "width of nose    "  "neutral.9"
##  "width of ear     "  "neutral.10"
##  "height of ear    "  "neutral.11"

plot(x=crime2$motor_vehicle_theft, crime2$forcible_rape)
```

```
plot.faces(a, x.pos=crime2$motor_vehicle_theft,
           y.pos=crime2$forcible_rape,
           width=200, height=10)
```



Wits and Wagers Activity

- Go to this google doc to record your answers.
- Make some Chernoff faces!
- Sample solution (I have supressed some of the output – please run code line by line if you are unsure what a particular line of code is trying to achieve)
- Recall, that the mean (when a sample is large) can generally be pretty informative! After I edited the data in which there were student-admitted typos the class average was generally pretty close to the true answer to those questions. This tells us that the most average of faces had the most consistently good guesses (compared to the rest of the class). So, how did you compare? Who stands out?