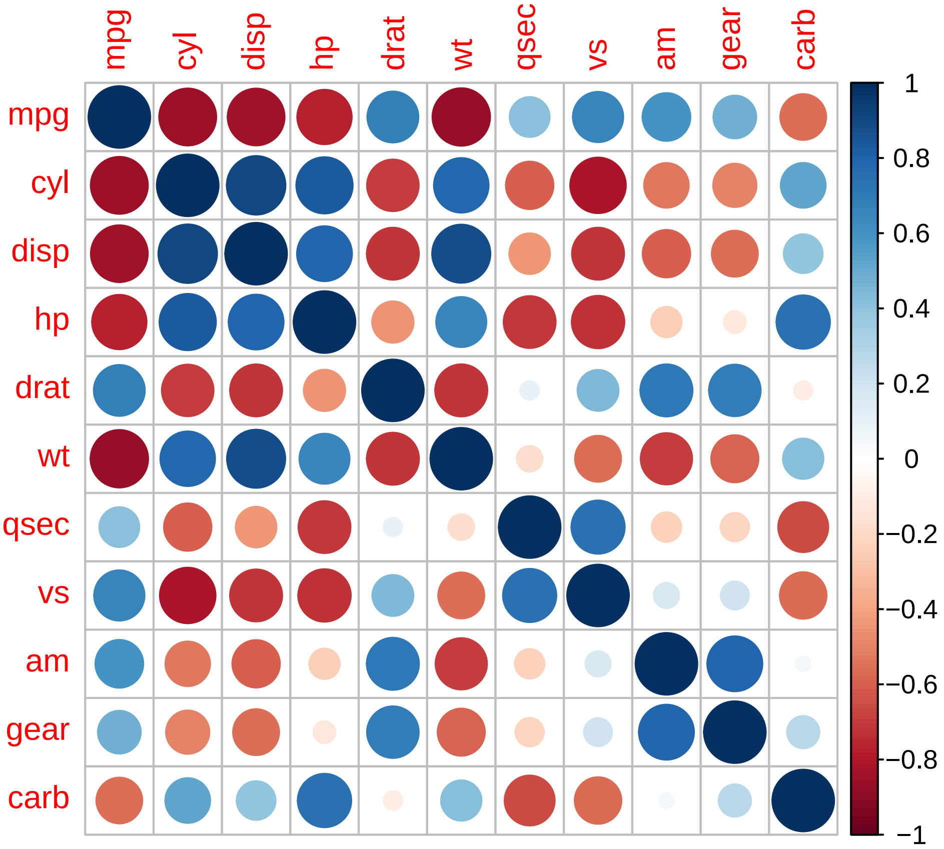
13_Miscellaneous_Graphs

Gino Tesei

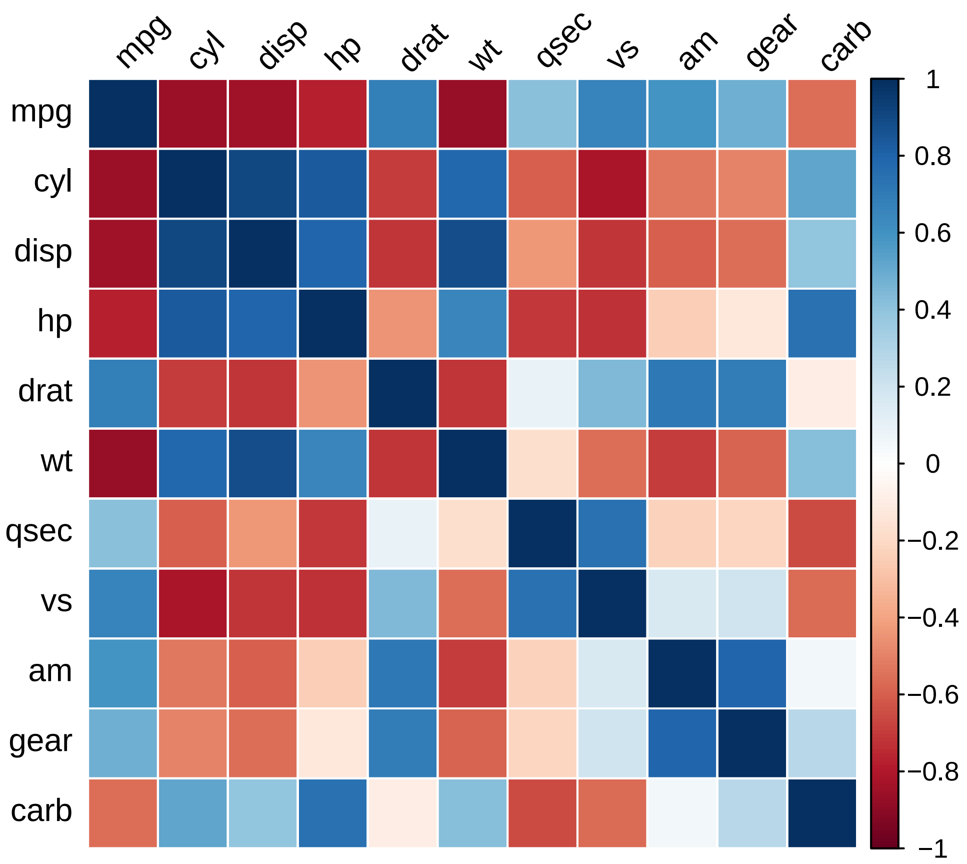
December 14, 2015

1. Making a Correlation Matrix

```
library(ggplot2)
library(gcookbook) # For the data set

library(plyr)

library(corrplot)
mcor <- cor(mtcars)

## corrplot
corrplot( mcor)
```



```
corrplot( mcor, method ="shade", shade.col = NA, tl.col ="black", tl.srt = 45)
```

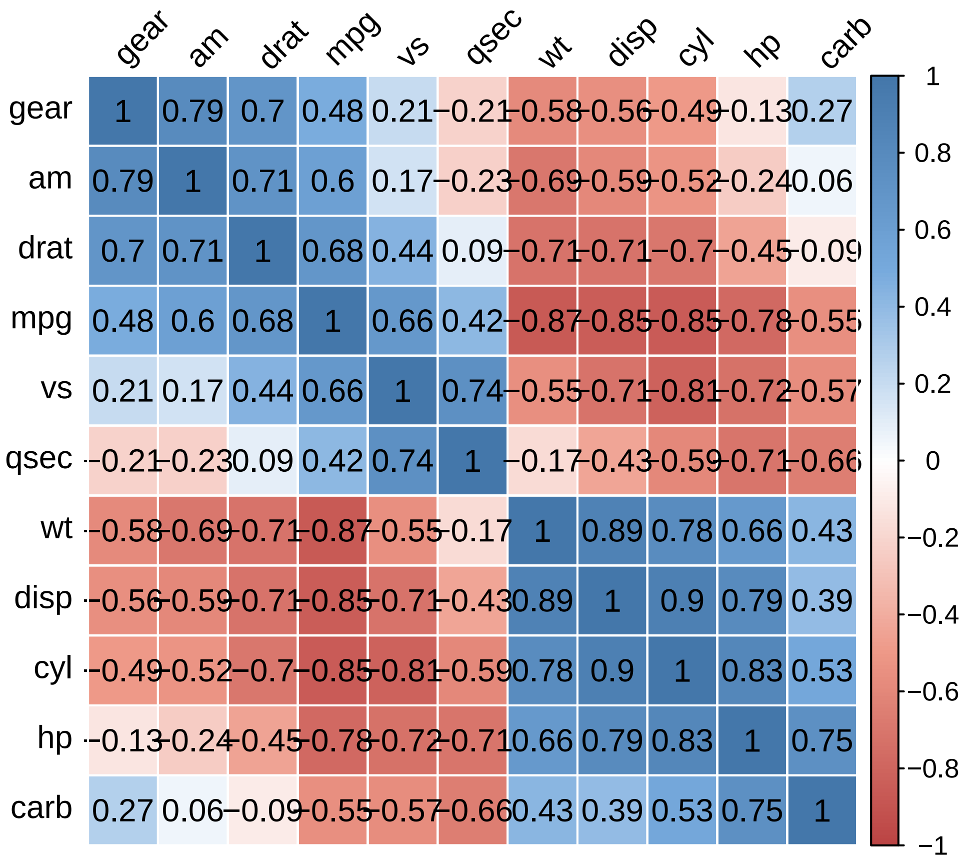


```
# Generate a lighter palette
col <- colorRampPalette( c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))

corrplot( mcor, method ="shade", shade.col = NA, tl.col ="black", tl.srt = 45,
          col = col(200),
          addCoef.col ="black",
          addcolorlabel ="no", order ="AOE")
```

```
## Warning in text.default(pos.xlabel[, 1], pos.xlabel[, 2], newcolnames, srt
## = tl.srt, : "addcolorlabel" is not a graphical parameter
```

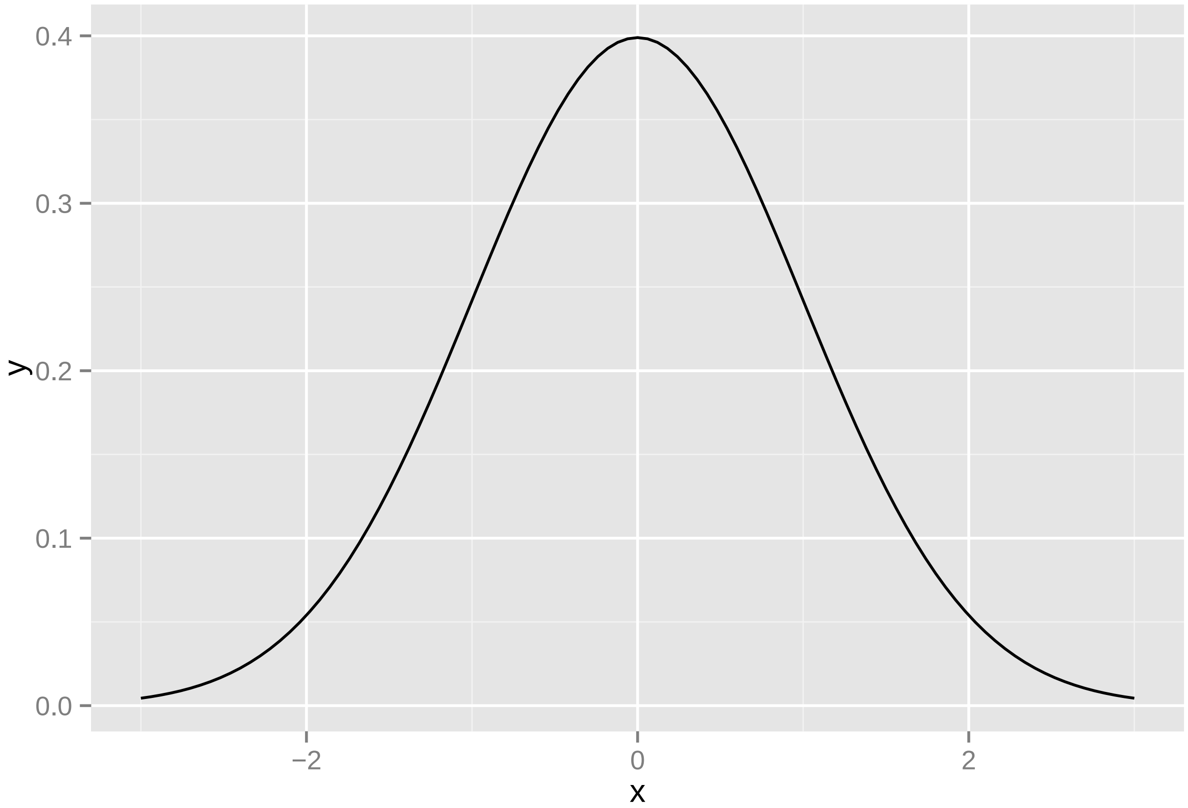
```
## Warning in text.default(pos.ylabel[, 1], pos.ylabel[, 2], newrownames, col
## = tl.col, : "addcolorlabel" is not a graphical parameter
```

```
## Warning in title(title, ...): "addcolorlabel" is not a graphical parameter
```

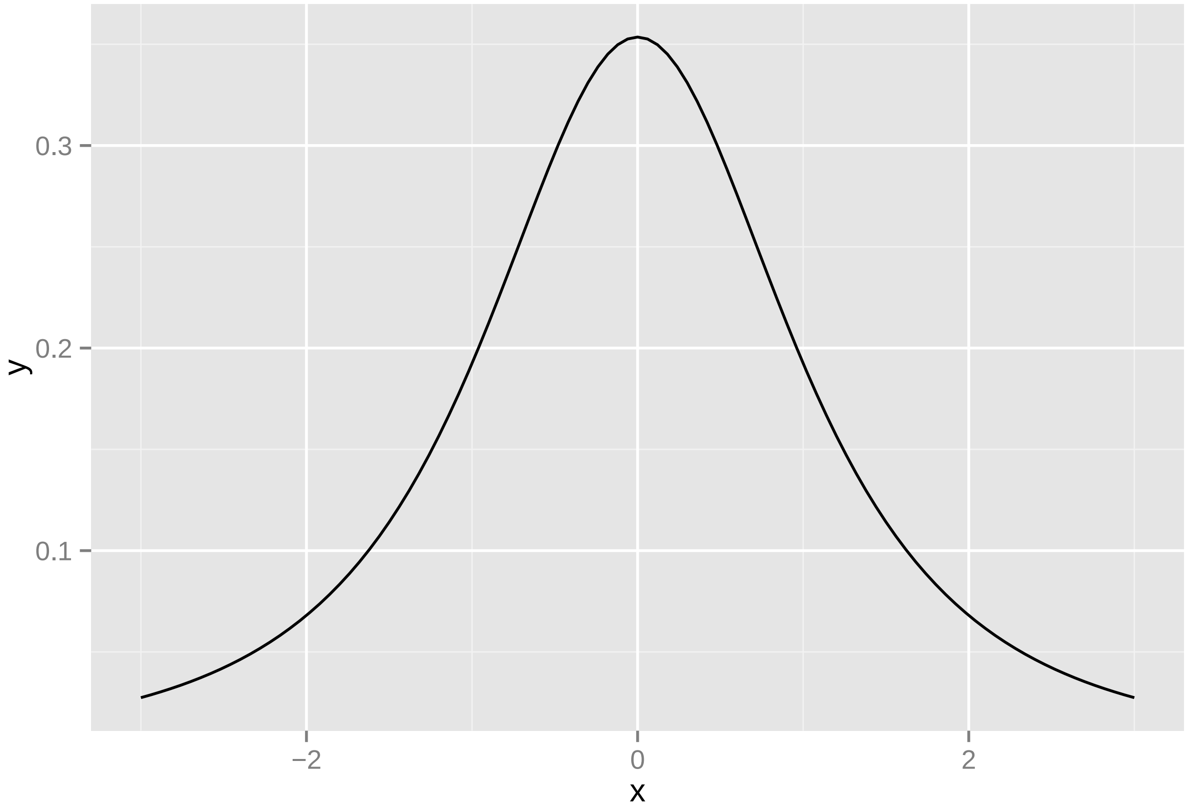


2. Plotting a Function

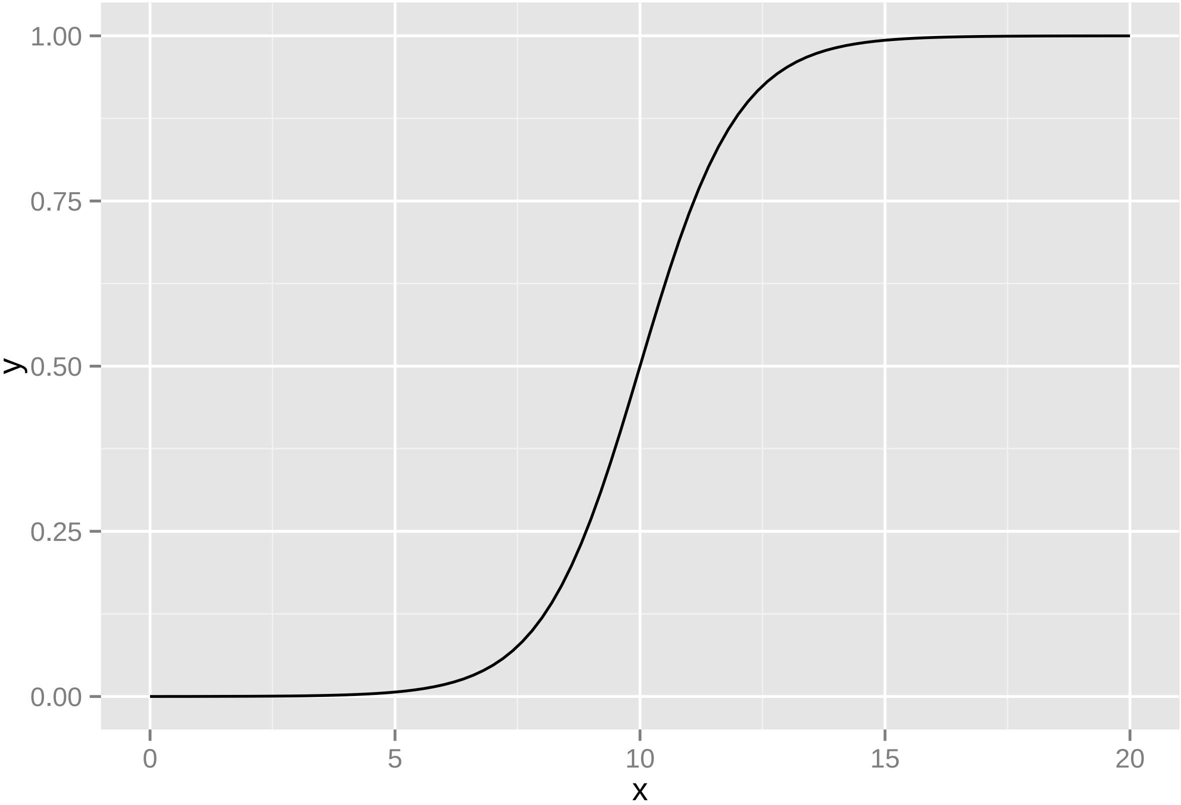
```
# The data frame is only used for setting the range
p <- ggplot( data.frame( x = c(-3,3)), aes( x = x))

p +
  stat_function( fun = dnorm)
```



```
p +
  stat_function( fun = dt, args = list( df = 2))
```

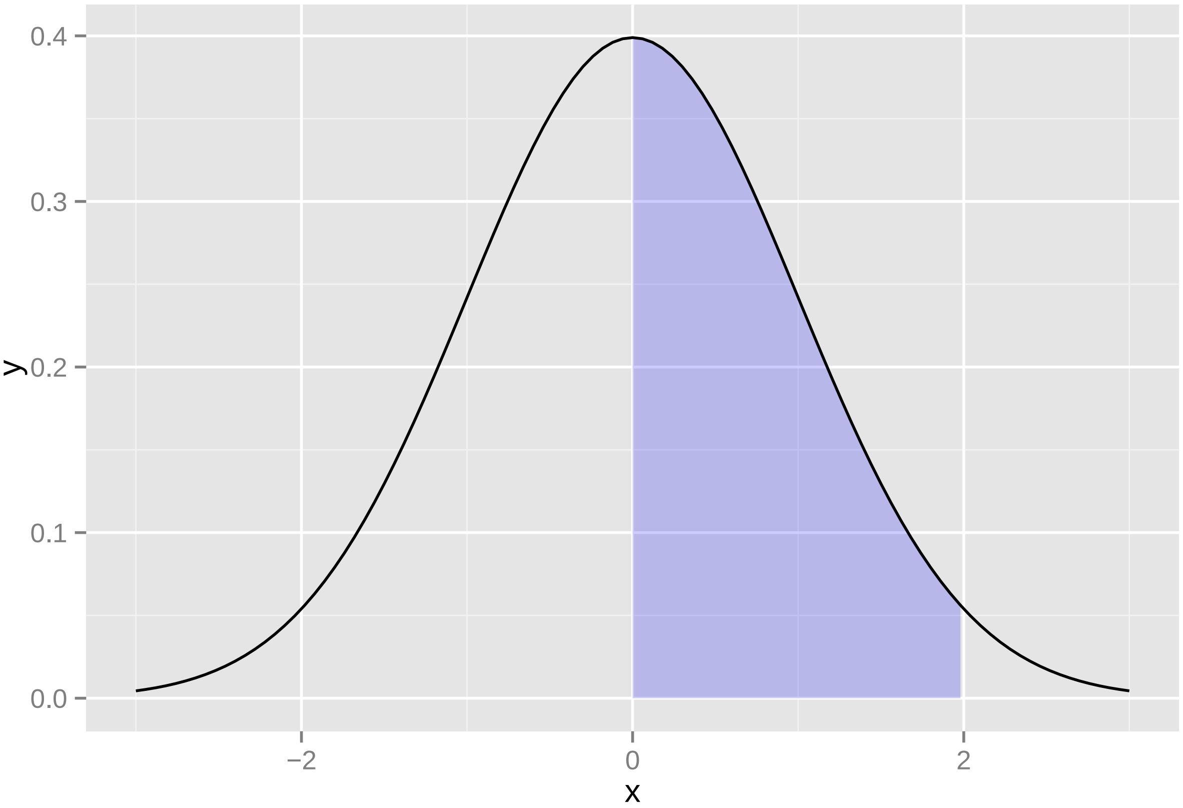


```r
## my funcs
myfun <- function( xvar) { 1/( 1 + exp(-xvar + 10)) }

ggplot( data.frame( x = c( 0, 20)), aes( x = x)) +
  stat_function( fun = myfun)
```



3.Shading a Subregion Under a Function Curve

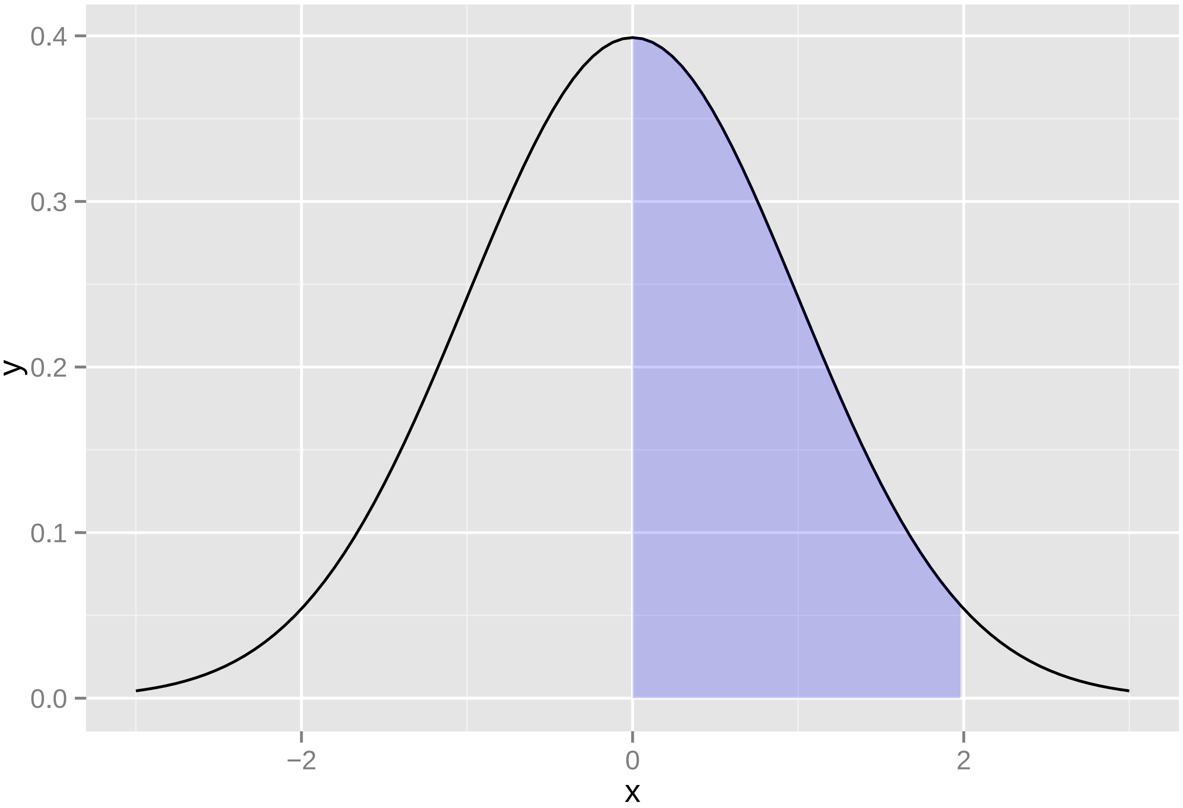
```r
# Return dnorm( x) for 0 < x < 2, and NA for all other x
dnorm_limit <- function( x) {
  y <- dnorm( x)
  y[ x < 0 | x > 2] <- NA
  return(y) }
# ggplot() with dummy data
p <- ggplot( data.frame( x = c(-3, 3)), aes( x = x))

p +
  stat_function( fun = dnorm_limit, geom ="area", fill ="blue", alpha = 0.2) +
  stat_function( fun = dnorm)
```



```r
# limit range
limitRange <- function( fun, min, max) {
  function( x) {
    y <- fun( x)
    y[ x < min | x > max] <- NA
    return(y)
    }
  }

# This returns a function
dlimit <- limitRange( dnorm, 0, 2)
# Now we'll try out the new function -- it only returns values for inputs # between 0 and 2
dlimit(-2:4)
```

```
## [1]         NA         NA 0.39894228 0.24197072 0.05399097         NA
## [7]         NA
```

```r
p +
  stat_function( fun = dnorm) +
  stat_function( fun = limitRange( dnorm, 0, 2), geom ="area", fill ="blue", alpha = 0.2)
```

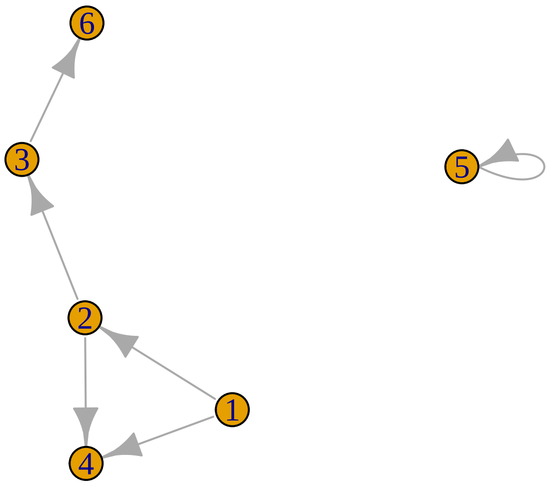


4. Creating a Network Graph

```
library( igraph)
```

```
##
## Attaching package: 'igraph'
##
## The following objects are masked from 'package:stats':
##
##     decompose, spectrum
##
## The following object is masked from 'package:base':
##
##     union
```

```
# Specify edges for a directed graph
gd <- graph( c( 1,2, 2,3, 2,4, 1,4, 5,5, 3,6))
plot(gd)
```



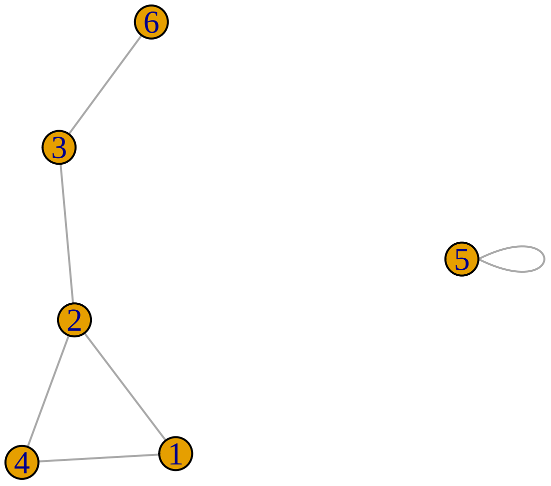
```
# For an undirected graph
gu <- graph( c( 1,2, 2,3, 2,4, 1,4, 5,5, 3,6), directed = FALSE)

#
str(gd)
```

```
## IGRAPH D--- 6 6 --
## + edges:
## [1] 1->2 2->3 2->4 1->4 5->5 3->6
```

```
str(gu)
```

```
## IGRAPH U--- 6 6 --
## + edges:
## [1] 1--2 2--3 2--4 1--4 5--5 3--6
```

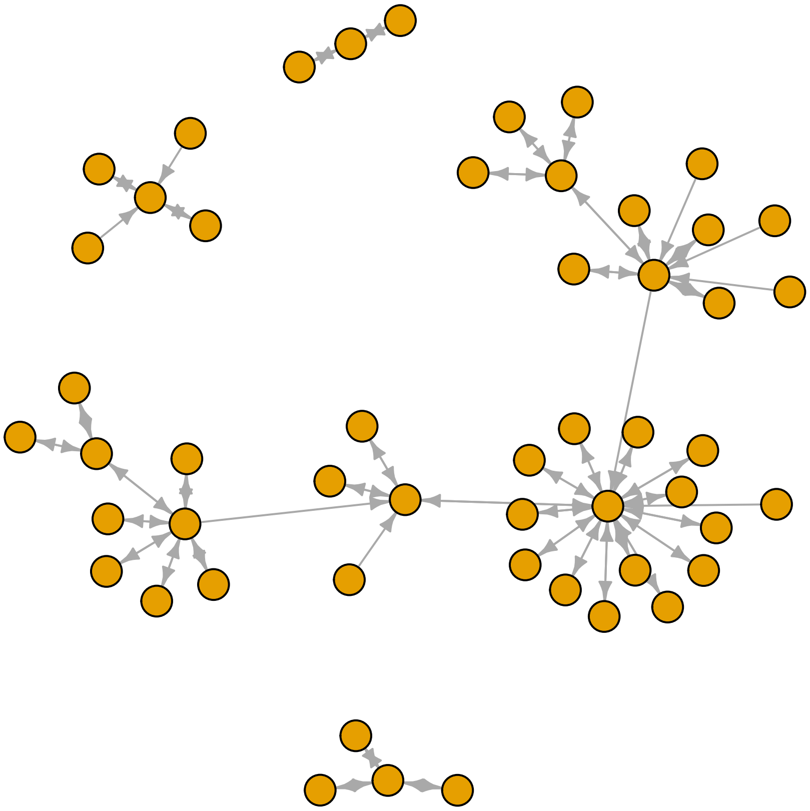
```
set.seed(229)
plot(gu)
```



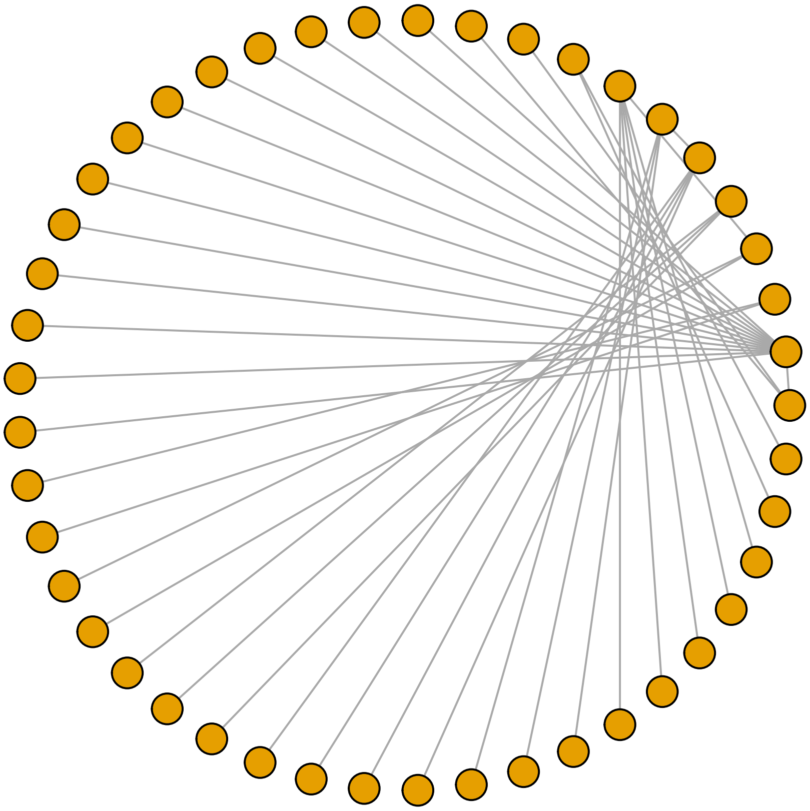
```r
# Create a graph object from the data set
str(madmen2)
```

```
## 'data.frame':    87 obs. of  2 variables:
##  $ Name1: chr  "Abe Drexler" "Allison" "Arthur Case" "Bellhop in Baltimore" ...
##  $ Name2: chr  "Peggy Olson" "Don Draper" "Betty Draper" "Sal Romano" ...
```

```r
g <- graph.data.frame( madmen2, directed = TRUE)
# Remove unnecessary margins
par( mar = c( 0,0,0,0))
plot( g, layout = layout.fruchterman.reingold, vertex.size = 8, edge.arrow.size = 0.5, vertex.label = NA
```



```r
## undirected graph.
g <- graph.data.frame( madmen, directed = FALSE)
par( mar = c( 0,0,0,0))
# Remove unnecessary margins
plot( g, layout = layout.circle, vertex.size = 8, vertex.label = NA)
```

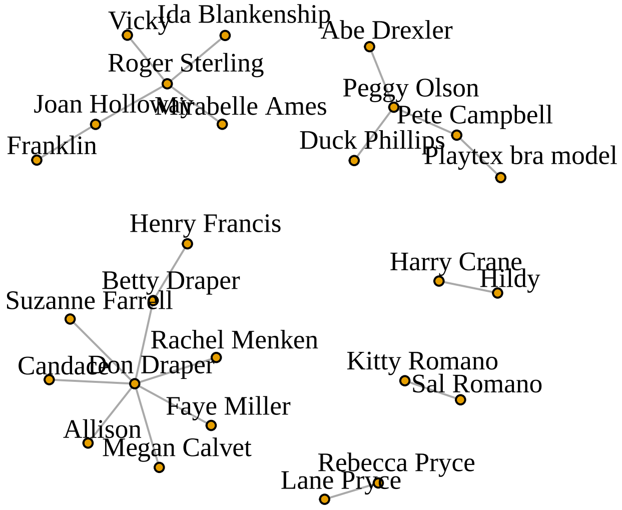


5. Using Text Labels in a Network Graph

```
# Copy madmen and drop every other row
m <- madmen[ 1: nrow( madmen) %% 2 == 1, ]
g <- graph.data.frame( m, directed = FALSE)

V(g)$name
```

```
##  [1] "Betty Draper"       "Don Draper"         "Harry Crane"
##  [4] "Joan Holloway"      "Lane Pryce"         "Peggy Olson"
##  [7] "Pete Campbell"      "Roger Sterling"     "Sal Romano"
## [10] "Henry Francis"      "Allison"            "Candace"
## [13] "Faye Miller"        "Megan Calvet"       "Rachel Menken"
## [16] "Suzanne Farrell"    "Hildy"              "Franklin"
## [19] "Rebecca Pryce"      "Abe Drexler"        "Duck Phillips"
## [22] "Playtex bra model" "Ida Blankenship"    "Mirabelle Ames"
## [25] "Vicky"              "Kitty Romano"
```

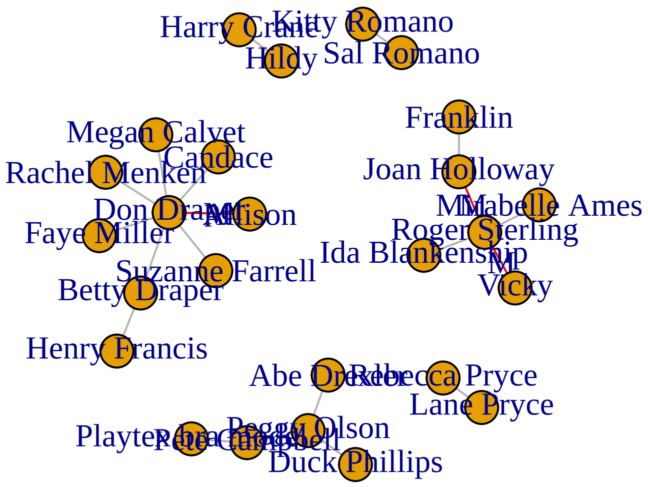
```
plot( g, layout = layout.fruchterman.reingold, vertex.size = 4, # Smaller nodes
      vertex.label = V(g)$name, # Set the labels
      vertex.label.cex = 0.8, # Slightly smaller font
      vertex.label.dist = 0.4, # Offset the labels
      vertex.label.color = "black")
```



```
# View the edges
E(g) # Set some of the labels to "M"
```

```
## + 20/20 edges (vertex names):
##  [1] Betty Draper  --Henry Francis     Don Draper    --Allison
##  [3] Betty Draper  --Don Draper        Don Draper    --Candace
##  [5] Don Draper    --Faye Miller       Don Draper    --Megan Calvet
##  [7] Don Draper    --Rachel Menken     Don Draper    --Suzanne Farrell
##  [9] Harry Crane   --Hildy             Joan Holloway --Franklin
## [11] Joan Holloway --Roger Sterling    Lane Pryce    --Rebecca Pryce
## [13] Peggy Olson   --Abe Drexler       Peggy Olson   --Duck Phillips
## [15] Peggy Olson   --Pete Campbell     Pete Campbell --Playtex bra model
## [17] Roger Sterling--Ida Blankenship   Roger Sterling--Mirabelle Ames
## [19] Roger Sterling--Vicky             Sal Romano    --Kitty Romano
```

```
E(g)[ c( 2,11,19)]$label <- "M" # Set color of all to grey, and then color a few red
E(g)$color <- "grey70"
E(g)[ c( 2,11,19)]$color <- "red"
plot( g)
```



6. Creating a Heat Map

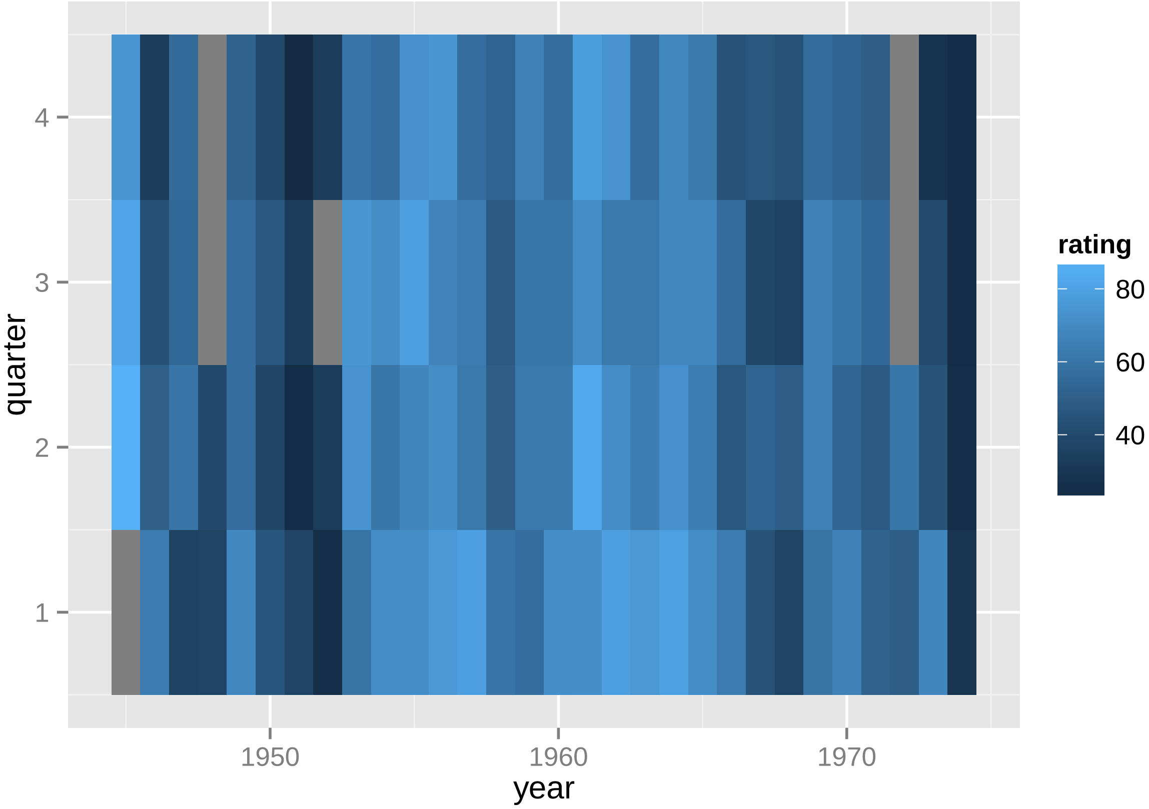
```
str( presidents)
```

```
##  Time-Series [1:120] from 1945 to 1975: NA 87 82 75 63 50 43 32 35 60 ...
```

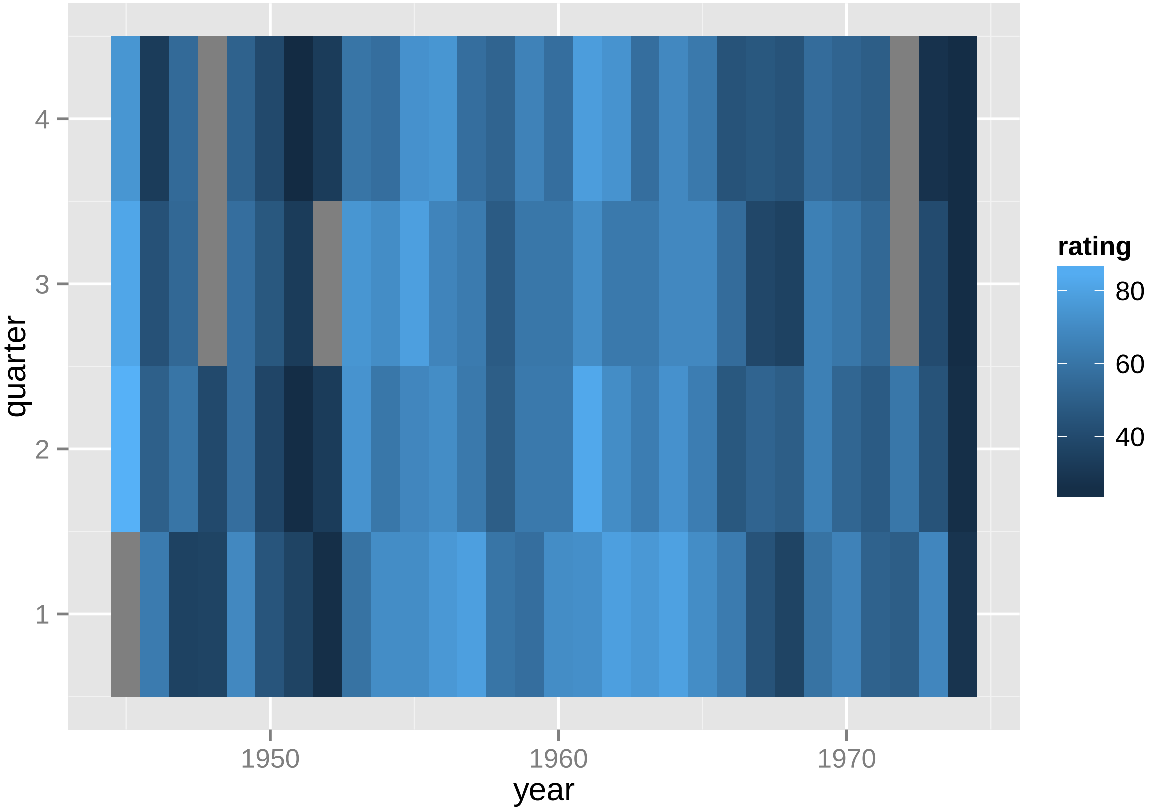
```
pres_rating <- data.frame( rating = as.numeric( presidents),
                           year = as.numeric( floor( time( presidents))),
                           quarter = as.numeric( cycle( presidents)) )
str(pres_rating)
```

```
## 'data.frame':    120 obs. of  3 variables:
##  $ rating : num  NA 87 82 75 63 50 43 32 35 60 ...
##  $ year   : num  1945 1945 1945 1945 1946 ...
##  $ quarter: num  1 2 3 4 1 2 3 4 1 2 ...
```

```
# Base plot
p <- ggplot( pres_rating, aes( x = year, y = quarter, fill = rating))

# Using geom_tile()
p +
  geom_tile() # Using geom_raster() - looks the same, but a little more efficient
```



```
p +
  geom_raster()
```



7. Creating a Dendrogram

```
# Get data from year 2009
c2 <- subset( countries, Year == 2009)

# Drop rows that have any NA values
c2 <- c2[ complete.cases( c2), ]
# Pick out a random 25 countries # (Set random seed to make this repeatable)
set.seed( 201)
c2 <- c2[ sample( 1: nrow( c2), 25), ]
str(c2)
```

```
## 'data.frame':    25 obs. of  7 variables:
##  $ Name        : Factor w/ 216 levels "Afghanistan",..: 132 34 79 12 215 29 13 75 21 58 ...
##  $ Code        : Factor w/ 216 levels "ABW","AFG","AGO",..: 134 33 80 12 215 19 13 77 16 58 ...
##  $ Year        : int  2009 2009 2009 2009 2009 2009 2009 2009 2009 2009 ...
##  $ GDP         : num  1690 39599 2685 45555 1006 ...
##  $ laborrate   : num  72.9 67.8 66.9 60.4 69.2 54.5 63 53.7 72.7 48.8 ...
##  $ healthexp   : num  74.2 4379.8 186.1 5037.3 47.1 ...
##  $ infmortality: num  27.8 5.2 25.9 3.6 71.5 11.1 41.1 3.5 74.7 20 ...
```

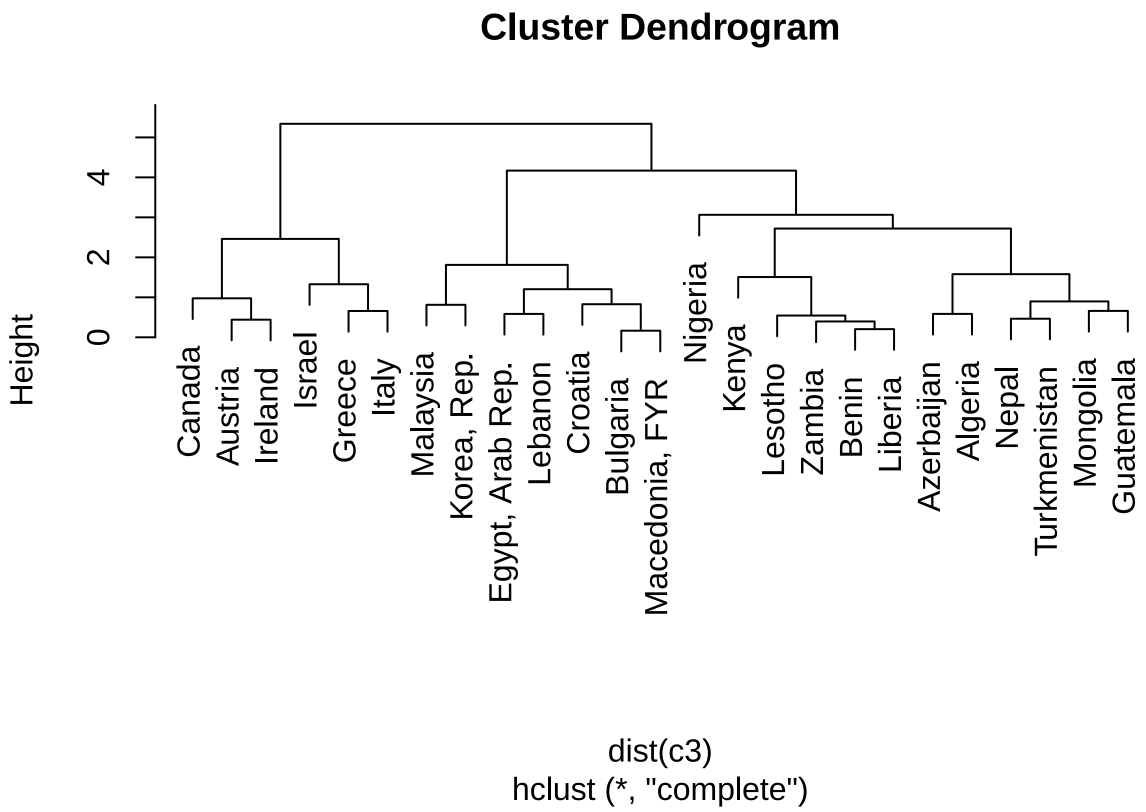
```
rownames( c2) <- c2$Name
c2 <- c2[, 4:7]
str(c2)
```

```
## 'data.frame':    25 obs. of  4 variables:
##  $ GDP         : num  1690 39599 2685 45555 1006 ...
##  $ laborrate   : num  72.9 67.8 66.9 60.4 69.2 54.5 63 53.7 72.7 48.8 ...
##  $ healthexp   : num  74.2 4379.8 186.1 5037.3 47.1 ...
##  $ infmortality: num  27.8 5.2 25.9 3.6 71.5 11.1 41.1 3.5 74.7 20 ...
```

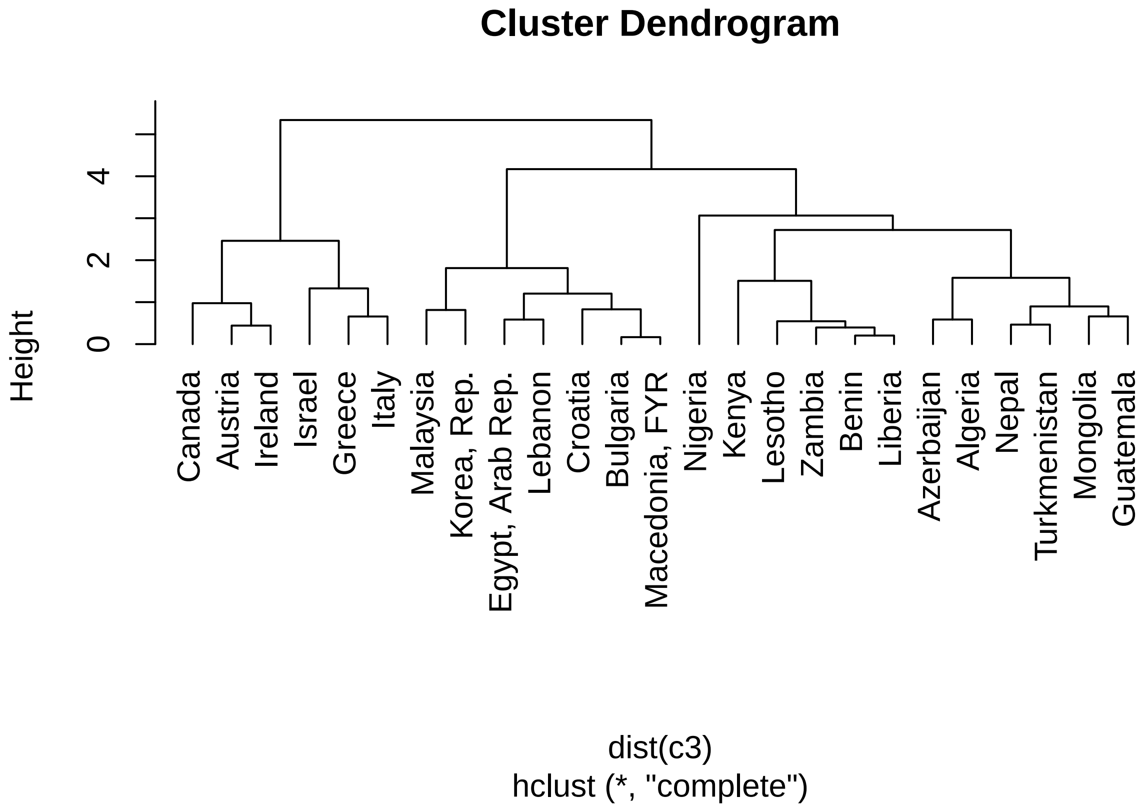
```
c3 <- scale( c2)
str(c3)
```

```
##  num [1:25, 1:4] -0.678 1.75 -0.615 2.132 -0.722 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:25] "Mongolia" "Canada" "Guatemala" "Austria" ...
##   ..$ : chr [1:4] "GDP" "laborrate" "healthexp" "infmortality"
##  - attr(*, "scaled:center")= Named num [1:4] 12278 62.3 1121.2 30.2
##   ..- attr(*, "names")= chr [1:4] "GDP" "laborrate" "healthexp" "infmortality"
##  - attr(*, "scaled:scale")= Named num [1:4] 15607.85 9.23 1651.06 28.36
##   ..- attr(*, "names")= chr [1:4] "GDP" "laborrate" "healthexp" "infmortality"
```

```
hc <- hclust( dist( c3))
# Make the dendrogram
plot( hc)
```



```
# With text aligned
plot( hc, hang = -1)
```



8. Creating a Mosaic Plot

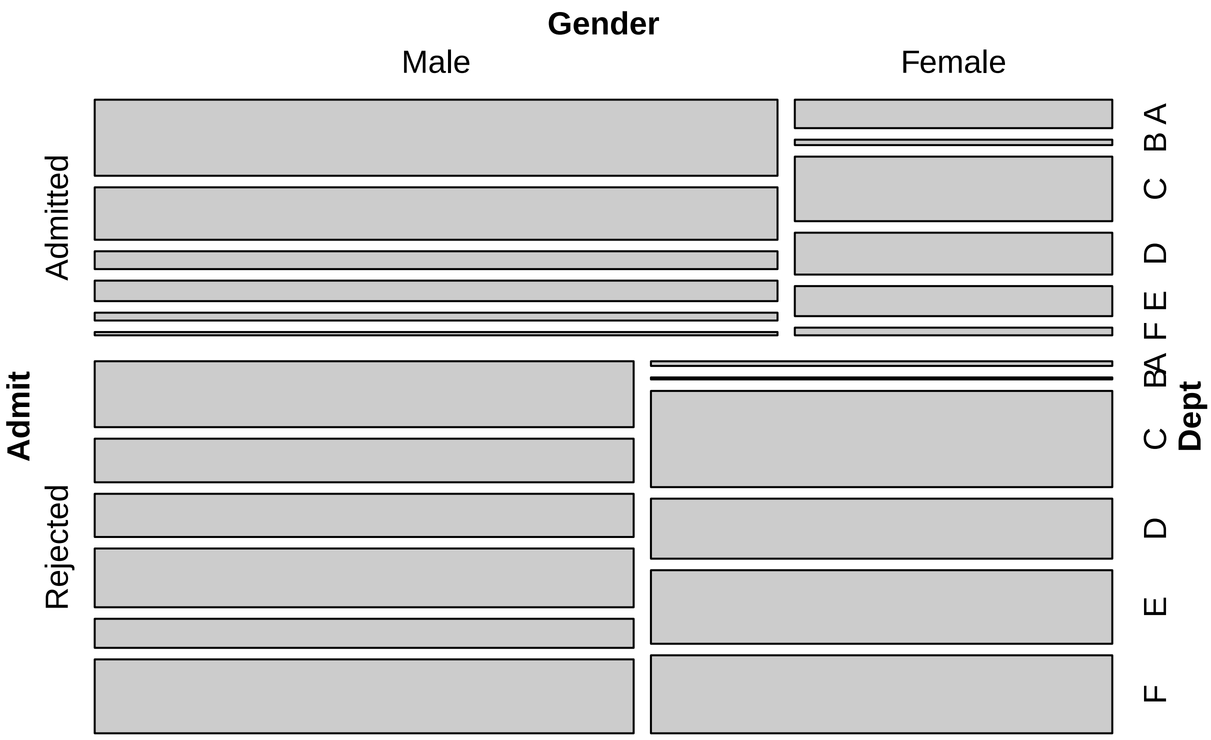
```
# Print a "flat" contingency table
ftable( UCBAdmissions)
```

```
##                    Dept   A   B   C   D   E   F
## Admit    Gender                                
## Admitted Male         512 353 120 138  53  22
##          Female        89  17 202 131  94  24
## Rejected Male         313 207 205 279 138 351
##          Female        19   8 391 244 299 317
```

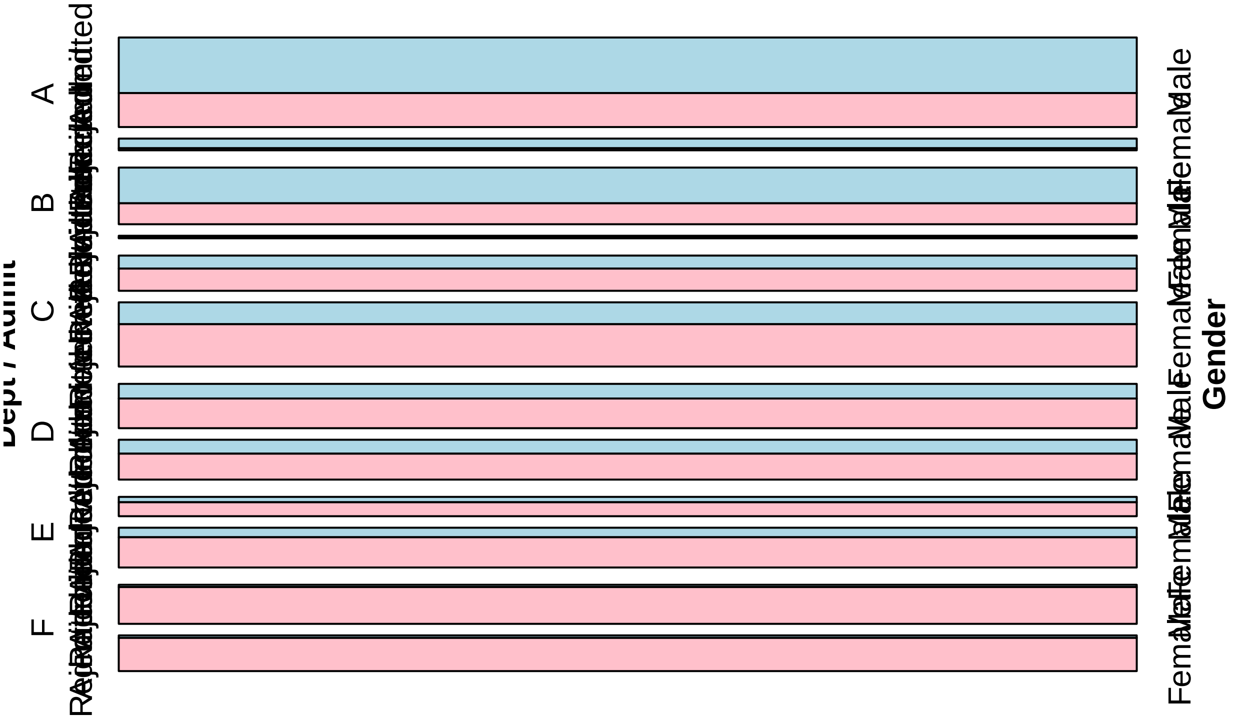
```
library( vcd)
```

```
## Loading required package: grid
```

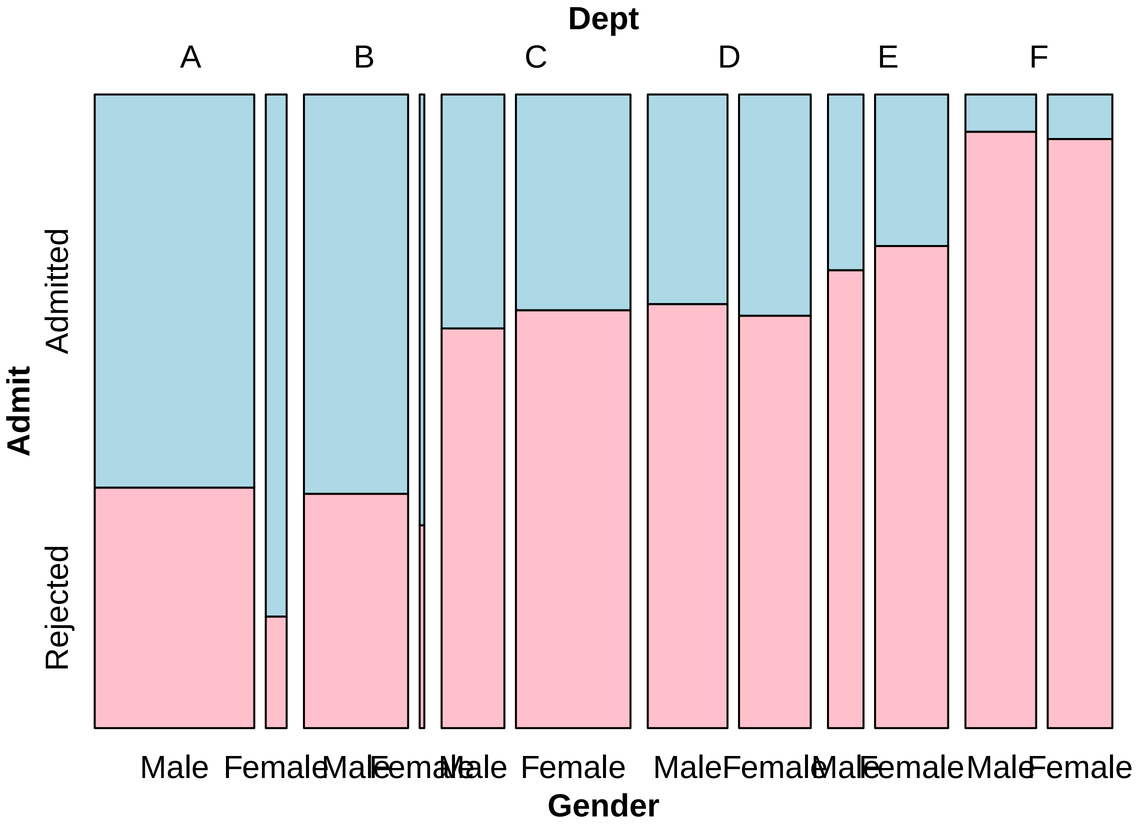
```
# Split by Admit, then Gender, then Dept
mosaic( ~ Admit + Gender + Dept, data = UCBAdmissions)
```



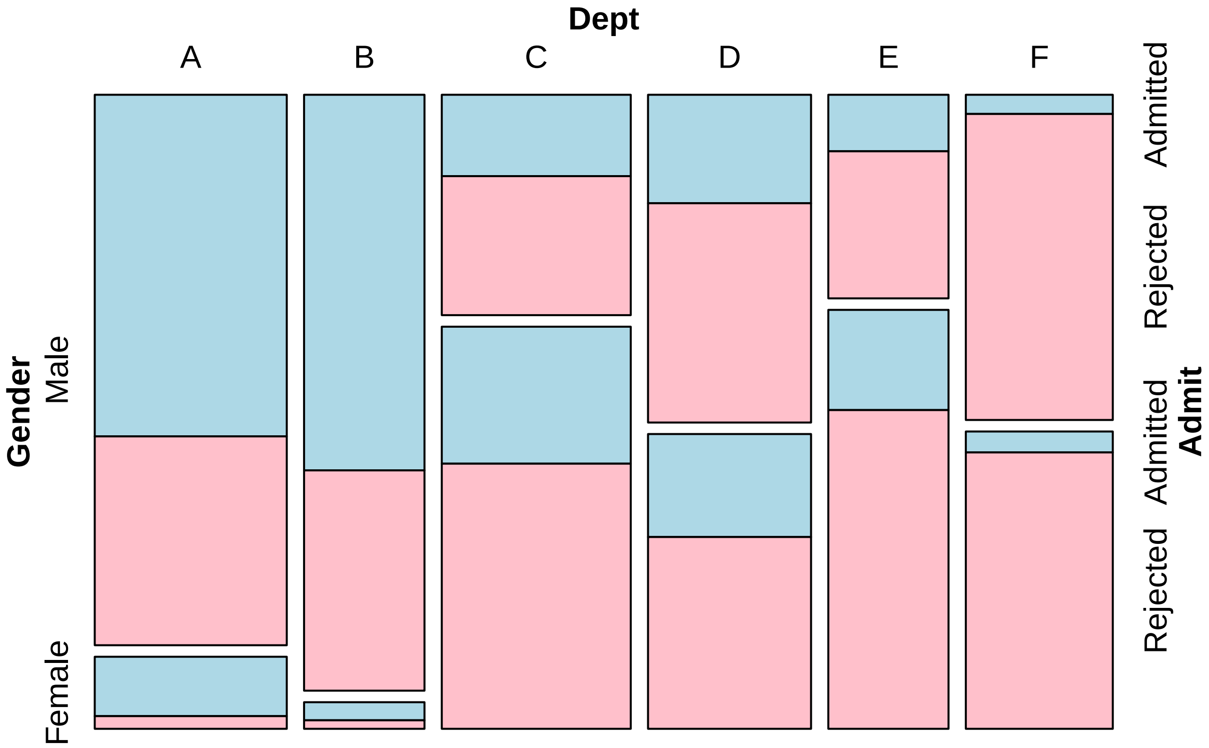
```
mosaic( ~ Dept + Gender + Admit, data = UCBAdmissions,
        highlighting ="Admit",
        highlighting_fill = c("lightblue", "pink"), direction = c(" v"," h"," v"))
```



```
# Another possible set of splitting directions
mosaic( ~ Dept + Gender + Admit, data = UCBAdmissions,
        highlighting ="Admit",
        highlighting_fill = c("lightblue", "pink"),
        direction = c("v", "v", "h"))
```



```
# This order makes it difficult to compare male and female
mosaic( ~ Dept + Gender + Admit, data = UCBAdmissions,
        highlighting ="Admit",
        highlighting_fill = c("lightblue", "pink"),
        direction = c("v", "h", "h"))
```

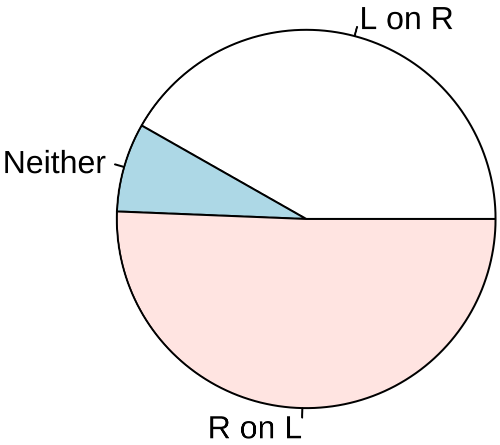


9. Creating a Pie Chart

```
library( MASS) # For the data set

# Get a table of how many cases are in each level of fold
fold <- table( survey$Fold)
fold
```

```
##
##  L on R Neither  R on L
##      99      18     120
```

```
# Make the pie chart
pie( fold)

pie( c( 99, 18, 120),
     labels = c("L on R", "Neither", "R on L"))
```

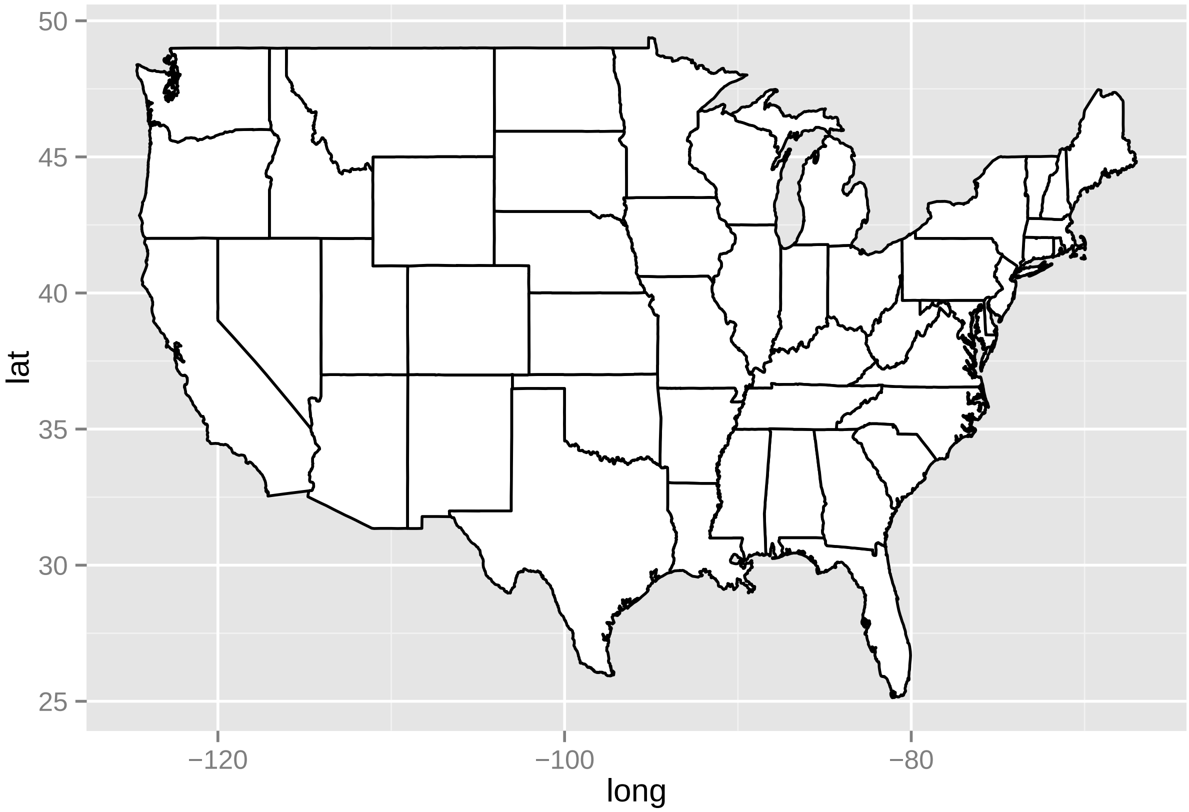


10. Creating a geographical map

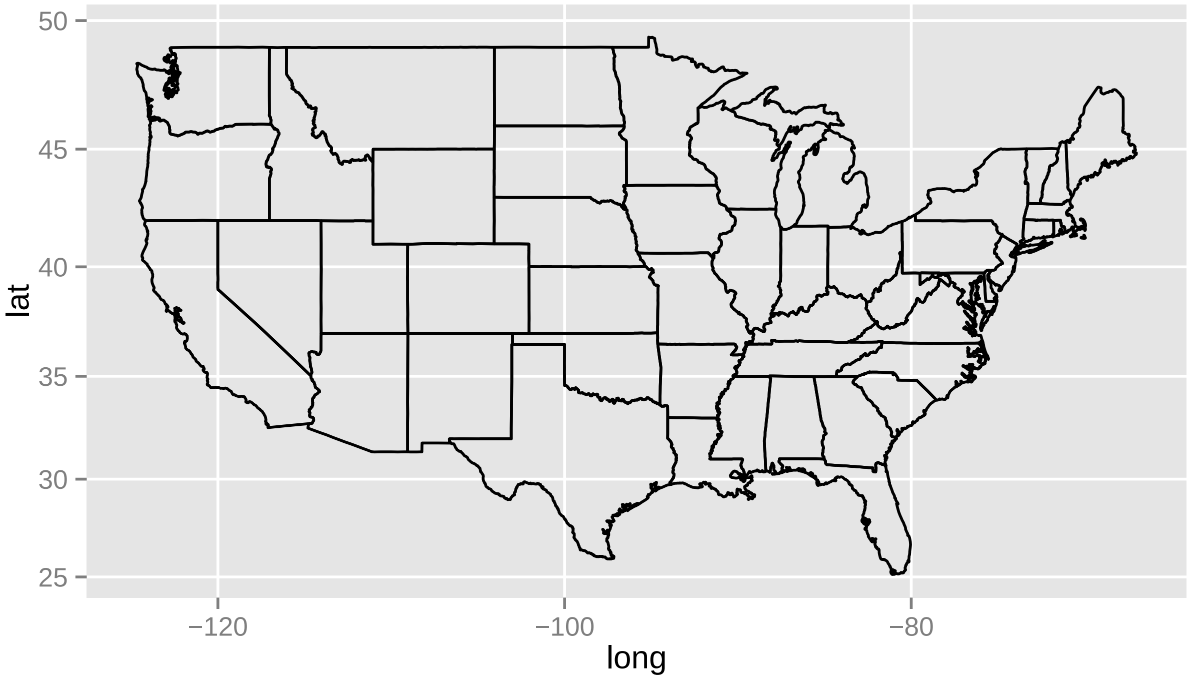
```
library( maps) # For map data
```

```
##
##  # ATTENTION: maps v3.0 has an updated 'world' map.        #
##  # Many country borders and names have changed since 1990. #
##  # Type '?world' or 'news(package="maps")'. See README_v3. #
##
##
##
## Attaching package: 'maps'
##
## The following object is masked from 'package:plyr':
##
##     ozone
```

```
# Get map data for USA
states_map <- map_data("state")

# ggplot2 must be loaded to use map_data()
ggplot( states_map, aes( x = long, y = lat, group = group)) +
  geom_polygon( fill ="white", colour ="black")
```



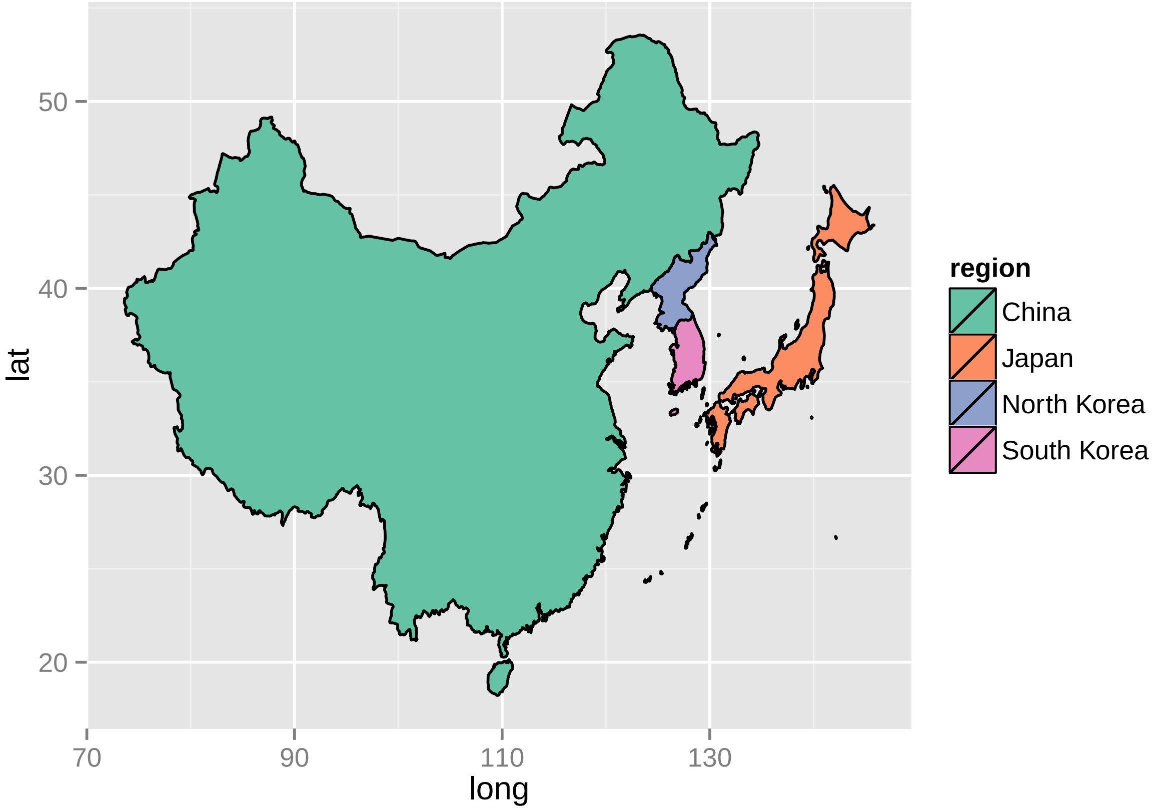
```
# geom_path (no fill) and Mercator projection
ggplot( states_map, aes( x = long, y = lat, group = group)) +
  geom_path() + coord_map("mercator")
```



```
# Get map data for world
world_map <- map_data("world")
str(world_map)
```

```
## 'data.frame':    101913 obs. of  6 variables:
##  $ long     : num  -69.9 -69.9 -69.9 -70 -70.1 ...
##  $ lat      : num  12.5 12.4 12.4 12.5 12.5 ...
##  $ group    : num  1 1 1 1 1 1 1 1 1 1 ...
##  $ order    : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ region   : chr  "Aruba" "Aruba" "Aruba" "Aruba" ...
##  $ subregion: chr  NA NA NA NA ...
```

```
east_asia <- map_data("world", region = c("Japan", "China", "North Korea", "South Korea"))

# Map region to fill color
ggplot( east_asia, aes( x = long, y = lat, group = group, fill = region)) +
  geom_polygon( colour ="black") +
  scale_fill_brewer( palette ="Set2")
```



11. Creating a Choropleth Map

```
# Transform the USArrests data set to the correct format
crimes <- data.frame( state = tolower( rownames( USArrests)), USArrests)
str(crimes)
```

```
## 'data.frame':    50 obs. of  5 variables:
##  $ state   : Factor w/ 50 levels "alabama","alaska",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Murder  : num  13.2 10 8.1 8.8 9 7.9 3.3 5.9 15.4 17.4 ...
##  $ Assault : int  236 263 294 190 276 204 110 238 335 211 ...
##  $ UrbanPop: int  58 48 80 50 91 78 77 72 80 60 ...
##  $ Rape    : num  21.2 44.5 31 19.5 40.6 38.7 11.1 15.8 31.9 25.8 ...
```

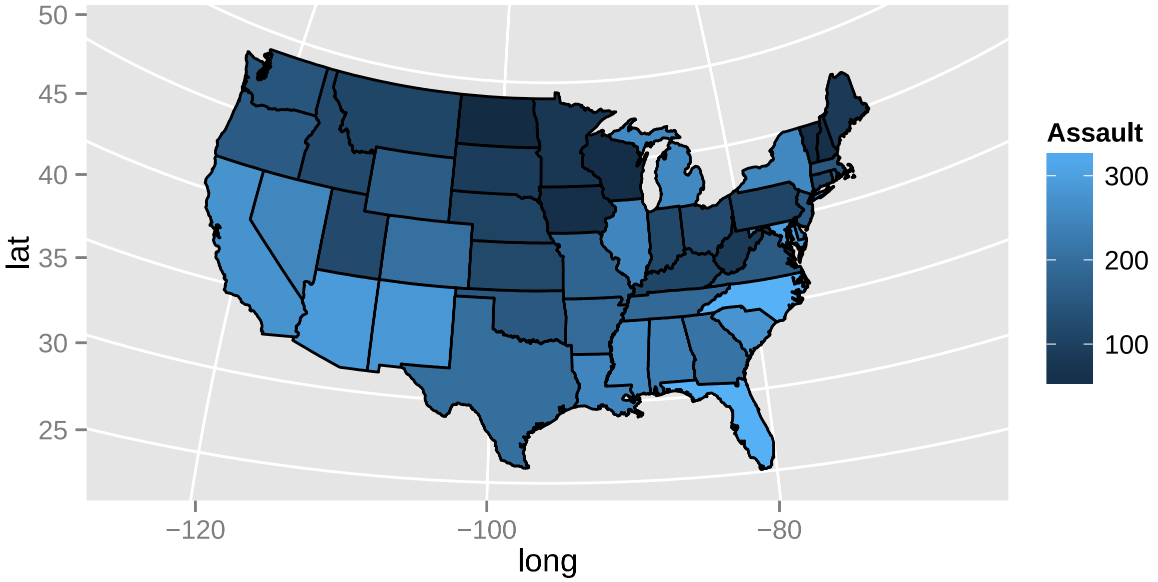
```
# Merge the data sets together
crime_map <- merge( states_map, crimes, by.x ="region", by.y ="state")

# After merging, the order has changed, which would lead to polygons drawn in # the incorrect order. So
head( crime_map)
```

```
##    region      long      lat group order subregion Murder Assault UrbanPop
## 1 alabama -87.46201 30.38968     1     1      <NA>   13.2     236       58
## 2 alabama -87.48493 30.37249     1     2      <NA>   13.2     236       58
## 3 alabama -87.95475 30.24644     1    13      <NA>   13.2     236       58
## 4 alabama -88.00632 30.24071     1    14      <NA>   13.2     236       58
## 5 alabama -88.01778 30.25217     1    15      <NA>   13.2     236       58
## 6 alabama -87.52503 30.37249     1     3      <NA>   13.2     236       58
##   Rape
## 1 21.2
## 2 21.2
## 3 21.2
## 4 21.2
## 5 21.2
## 6 21.2
```

```
# Sort by group, then order
crime_map <- arrange( crime_map, group, order)
head( crime_map)
```

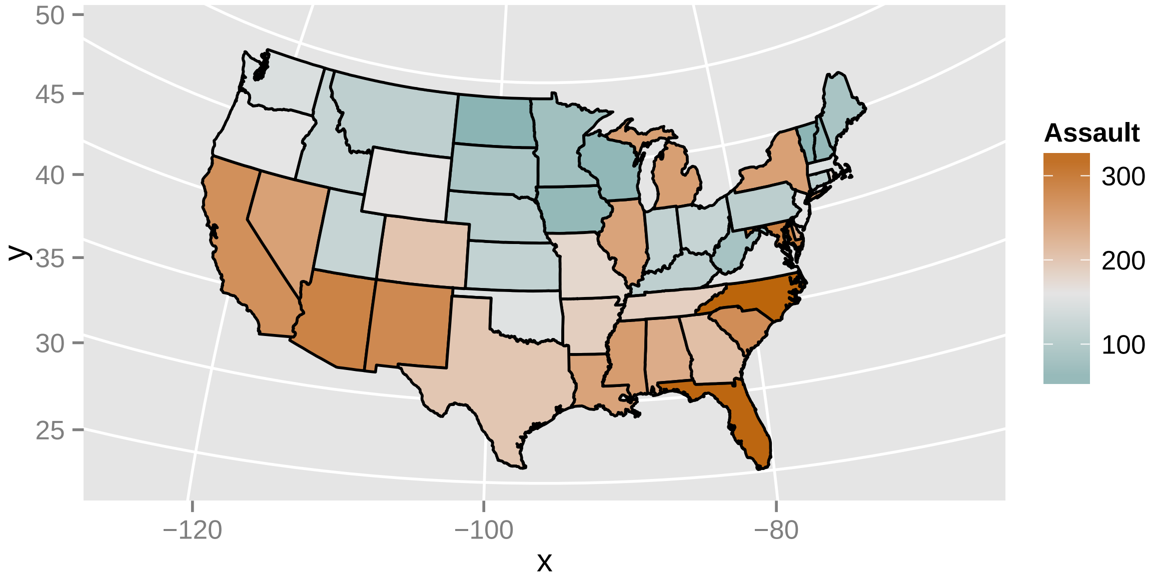
```
##    region      long      lat group order subregion Murder Assault UrbanPop
## 1 alabama -87.46201 30.38968     1     1      <NA>   13.2     236       58
## 2 alabama -87.48493 30.37249     1     2      <NA>   13.2     236       58
## 3 alabama -87.52503 30.37249     1     3      <NA>   13.2     236       58
## 4 alabama -87.53076 30.33239     1     4      <NA>   13.2     236       58
## 5 alabama -87.57087 30.32665     1     5      <NA>   13.2     236       58
## 6 alabama -87.58806 30.32665     1     6      <NA>   13.2     236       58
##   Rape
## 1 21.2
## 2 21.2
## 3 21.2
## 4 21.2
## 5 21.2
## 6 21.2
```

```
ggplot( crime_map, aes( x = long, y = lat, group = group, fill = Assault)) +
  geom_polygon( colour ="black") +
  coord_map("polyconic")
```



```
ggplot( crimes, aes( map_id = state, fill = Assault)) +
  geom_map( map = states_map, colour ="black") +
  scale_fill_gradient2( low ="#559999", mid ="grey90", high ="#BB650B", midpoint = median( crimes$Assaul
  expand_limits( x = states_map$long, y = states_map$lat) + coord_map("polyconic")
```

```
## Warning: Non Lab interpolation is deprecated
```



```
# discretized.
qa <- quantile( crimes$Assault, c( 0, 0.2, 0.4, 0.6, 0.8, 1.0))
qa
```

```
##    0%   20%   40%   60%   80%  100%
##  45.0  98.8 135.0 188.8 254.2 337.0
```

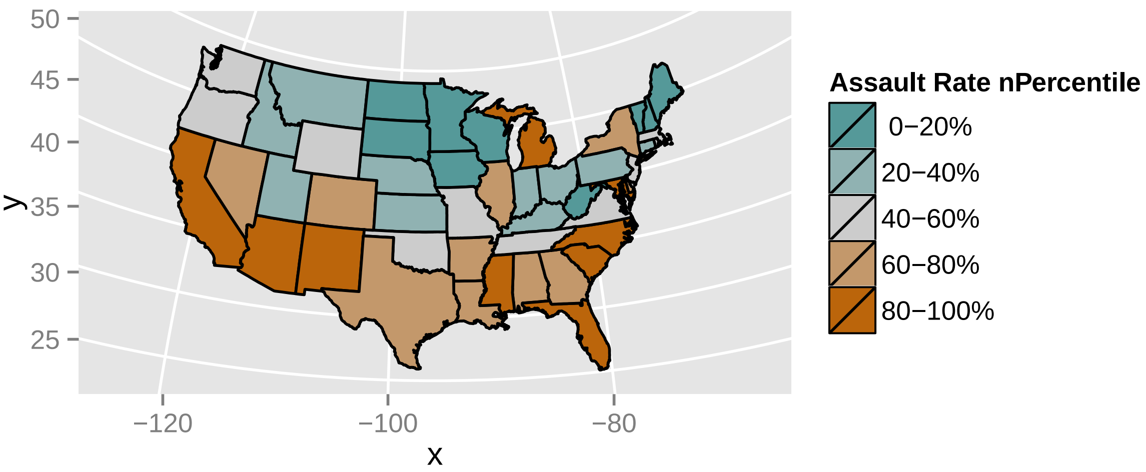
```
# Add a column of the quantile category
crimes$Assault_q <- cut( crimes$Assault, qa, labels = c(" 0-20%", "20-40%", "40-60%", "60-80%", "80-100
str(crimes)
```

```
## 'data.frame':    50 obs. of  6 variables:
##  $ state    : Factor w/ 50 levels "alabama","alaska",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Murder   : num  13.2 10 8.1 8.8 9 7.9 3.3 5.9 15.4 17.4 ...
##  $ Assault  : int  236 263 294 190 276 204 110 238 335 211 ...
##  $ UrbanPop : int  58 48 80 50 91 78 77 72 80 60 ...
##  $ Rape     : num  21.2 44.5 31 19.5 40.6 38.7 11.1 15.8 31.9 25.8 ...
##  $ Assault_q: Factor w/ 5 levels " 0-20%","20-40%",..: 4 5 5 4 5 4 2 4 5 4 ...
```

```
# Generate a discrete color palette with 5 values
pal <- colorRampPalette( c("#559999", "grey80", "#BB650B"))( 5)
pal
```

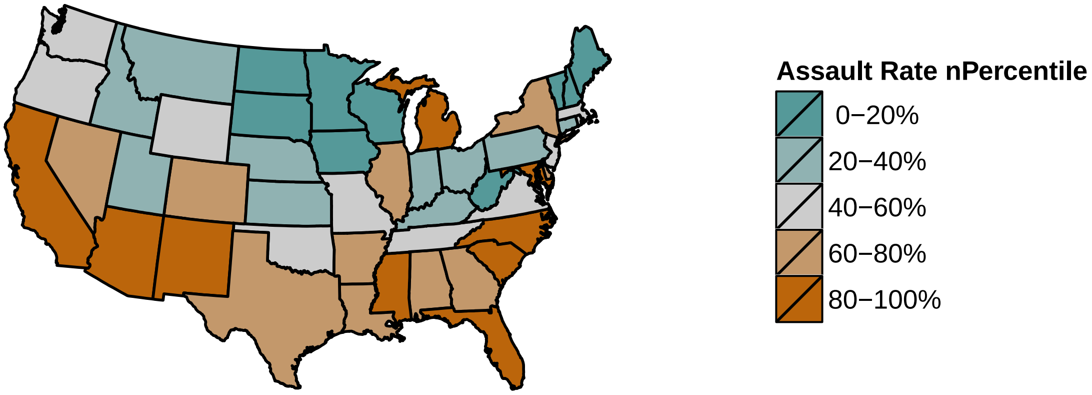
```
## [1] "#559999" "#90B2B2" "#CCCCCC" "#C3986B" "#BB650B"
```

```
ggplot( crimes, aes( map_id = state, fill = Assault_q)) +
  geom_map( map = states_map, colour ="black") +
  scale_fill_manual( values = pal) +
  expand_limits( x = states_map$long, y = states_map$lat) + coord_map("polyconic") +
  labs( fill ="Assault Rate\ nPercentile")
```



12. Making a Map with a Clean Background

```
# Create a theme with many of the background elements removed
theme_clean <- function( base_size = 12) {
  require( grid)
  # Needed for unit()
  theme_grey( base_size) %+replace%
    theme( axis.title = element_blank(),
           axis.text = element_blank(),
           panel.background = element_blank(),
           panel.grid = element_blank(),
           axis.ticks.length = unit( 0, "cm"),
           axis.ticks.margin = unit( 0, "cm"),
           panel.margin = unit( 0, "lines"),
           plot.margin = unit( c( 0, 0, 0, 0), "lines"),
           complete = TRUE
           )
  }

ggplot( crimes, aes( map_id = state, fill = Assault_q)) +
  geom_map( map = states_map, colour ="black") +
  scale_fill_manual( values = pal) +
  expand_limits( x = states_map$long, y = states_map$lat) +
  coord_map("polyconic") +
  labs( fill ="Assault Rate\ nPercentile") +
  theme_clean()
```


Assault Rate nPercentile
0−20%
20−40%
40−60%
60−80%
80−100%