

## Introduction to Graphics in R

Cori Mar 7/10/18

```
### Reading in the data
dat <-read.table(file="thedata.txt",header=T)
head(dat)
```

```
##   region age gender income
## 1    AA  28     F   6621
## 2    AA  41     F   6030
## 3    AA  55     N   7178
## 4    AA  56     F   5152
## 5    AA  57     N   4859
## 6    AA  56     M   7720
```

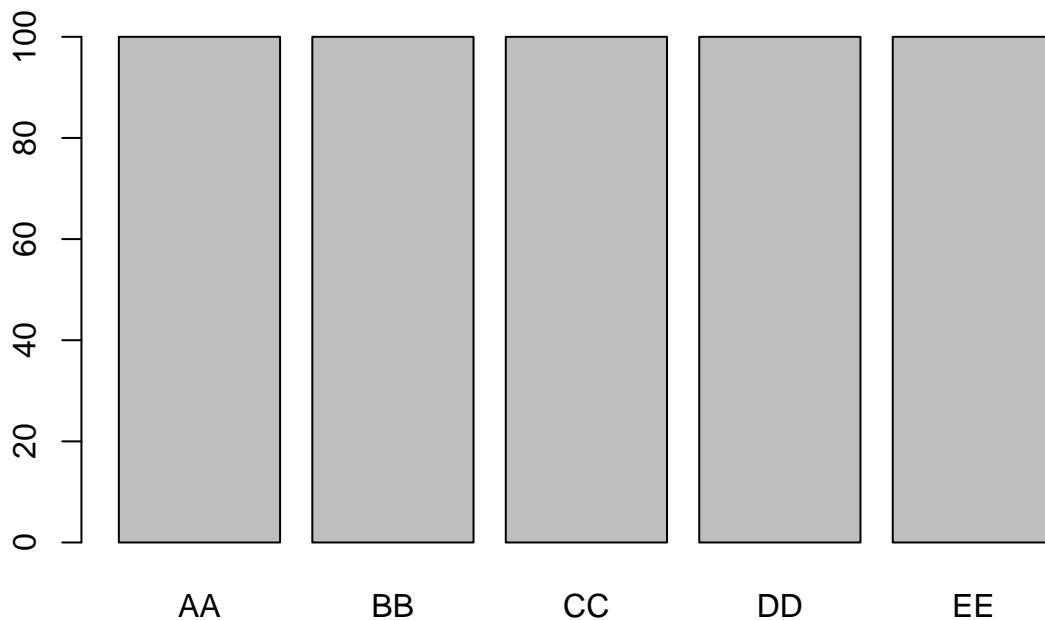
```
tail(dat)
```

```
##   region age gender income
## 495    EE  29     N   6025
## 496    EE  55     M   6906
## 497    EE  39     M   8562
## 498    EE  44     M   5894
## 499    EE  35     F   9028
## 500    EE  51     F   8623
```

```
# Exploring the data
table(dat$region)
```

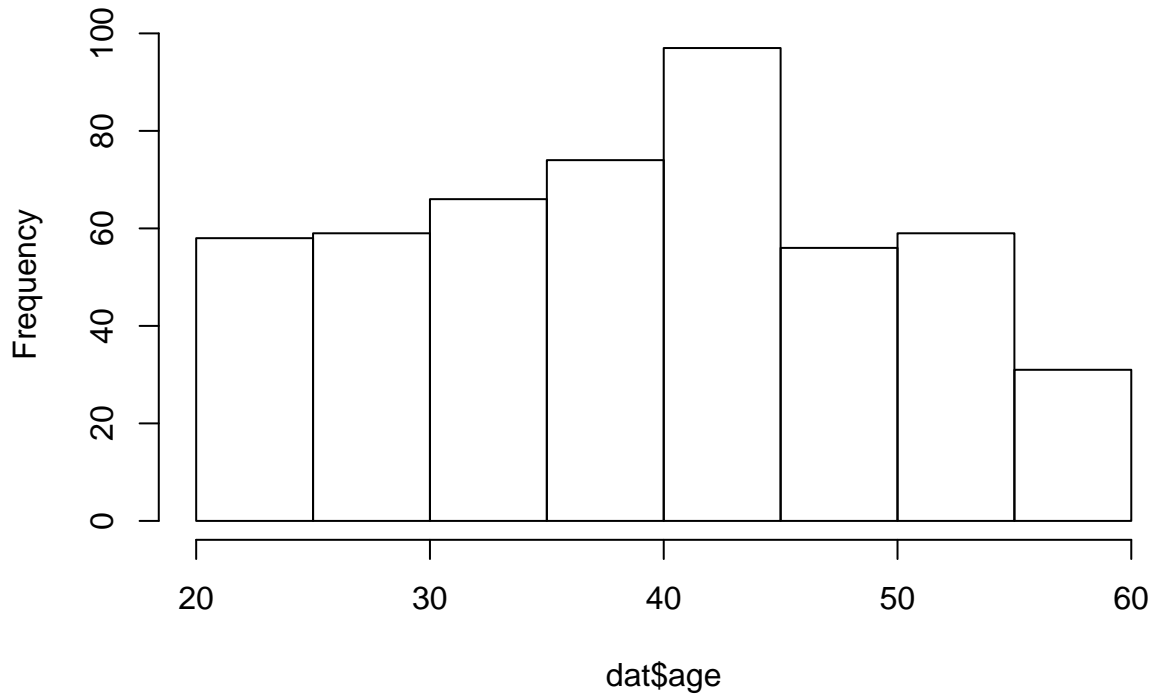
```
##
##  AA  BB  CC  DD  EE
## 100 100 100 100 100
```

```
barplot(table(dat$region))
```



```
hist(dat$age)
```

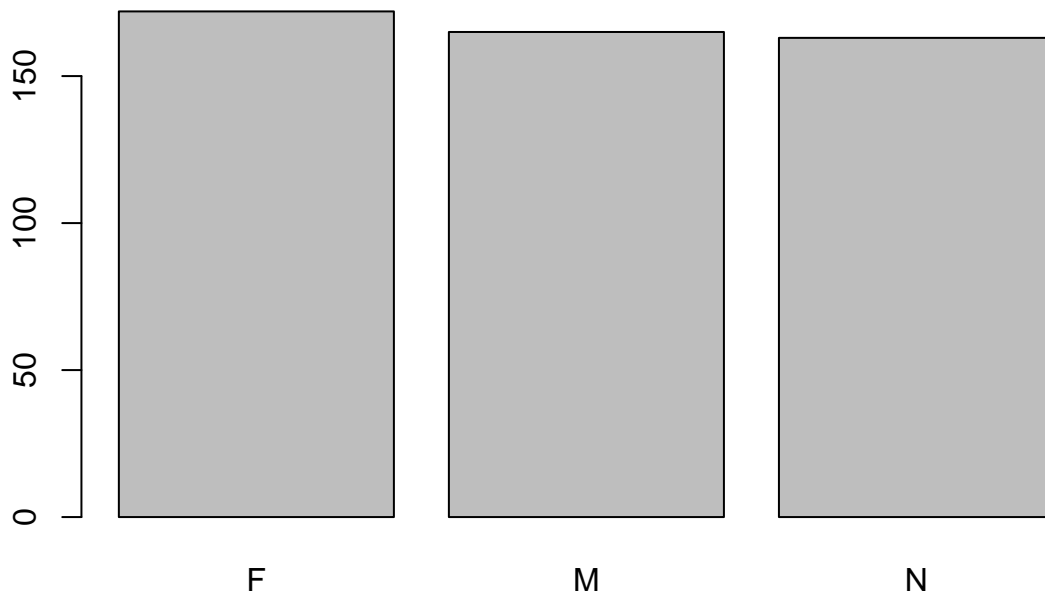
Histogram of dat\$age



```
table(dat$gender)
```

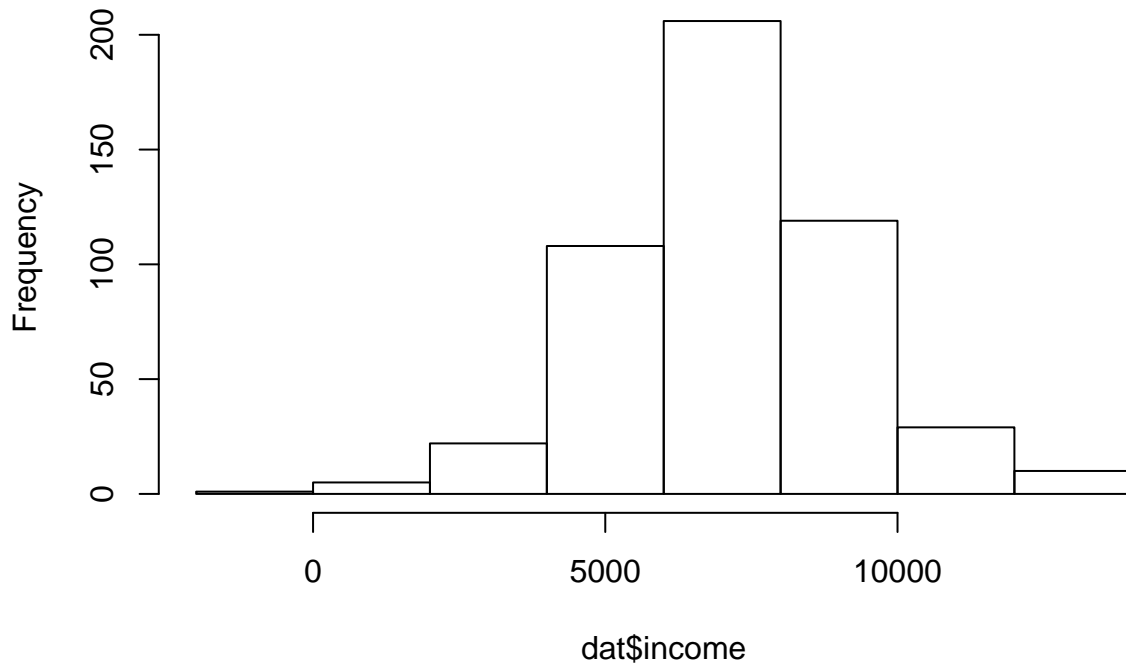
```
##  
##  F  M  N  
## 172 165 163
```

```
barplot(table(dat$gender))
```



```
hist(dat$income)
```

## Histogram of dat\$income



```
sum(dat$income < 0)
```

```
## [1] 1
```

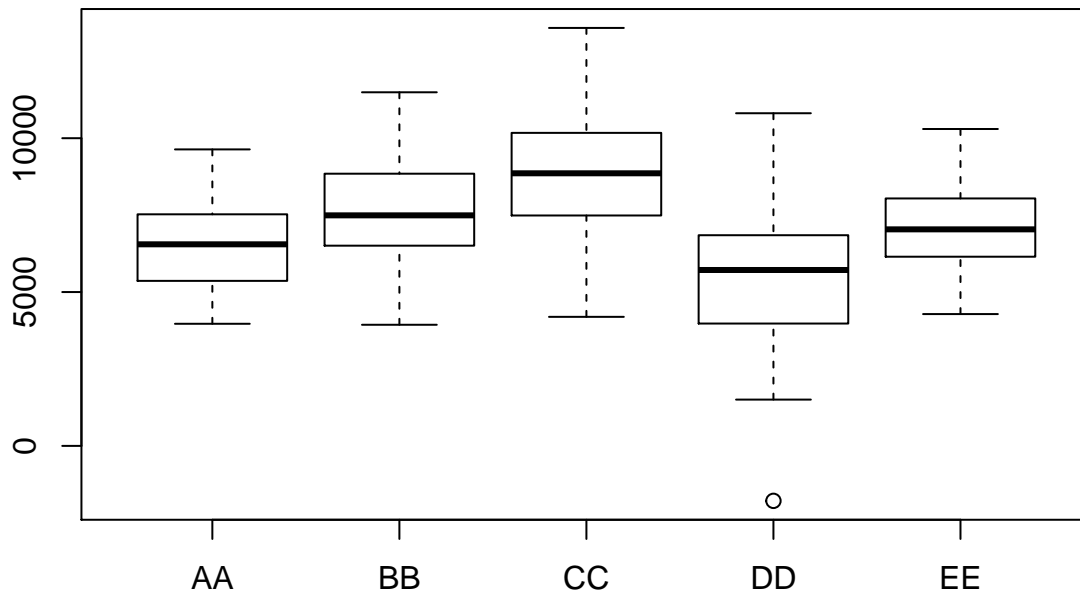
```
sum(dat$income == 0)
```

```
## [1] 0
```

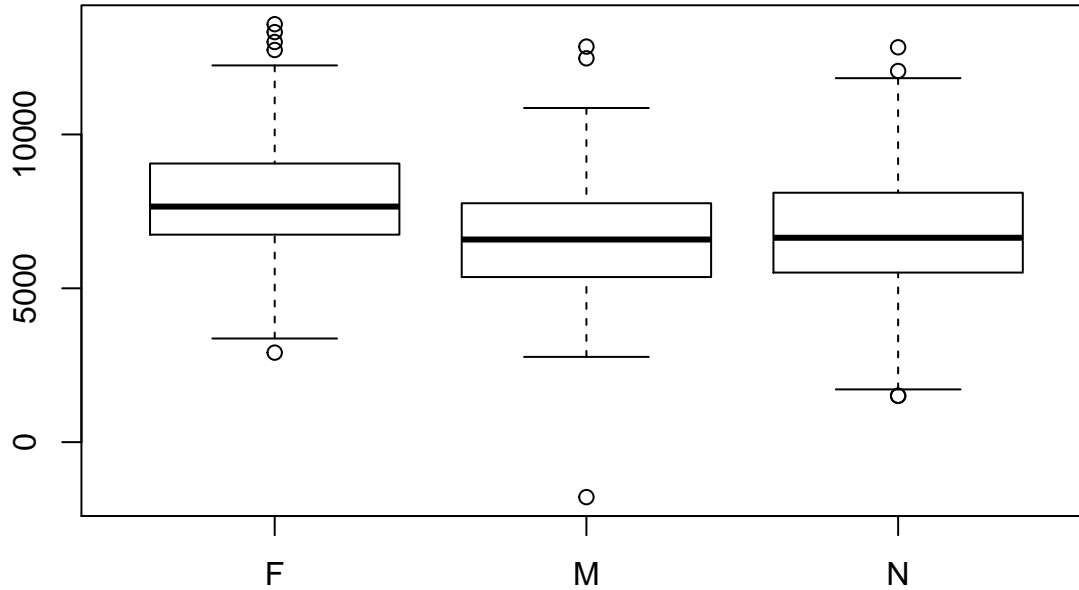
```
min(dat$income)
```

```
## [1] -1786
```

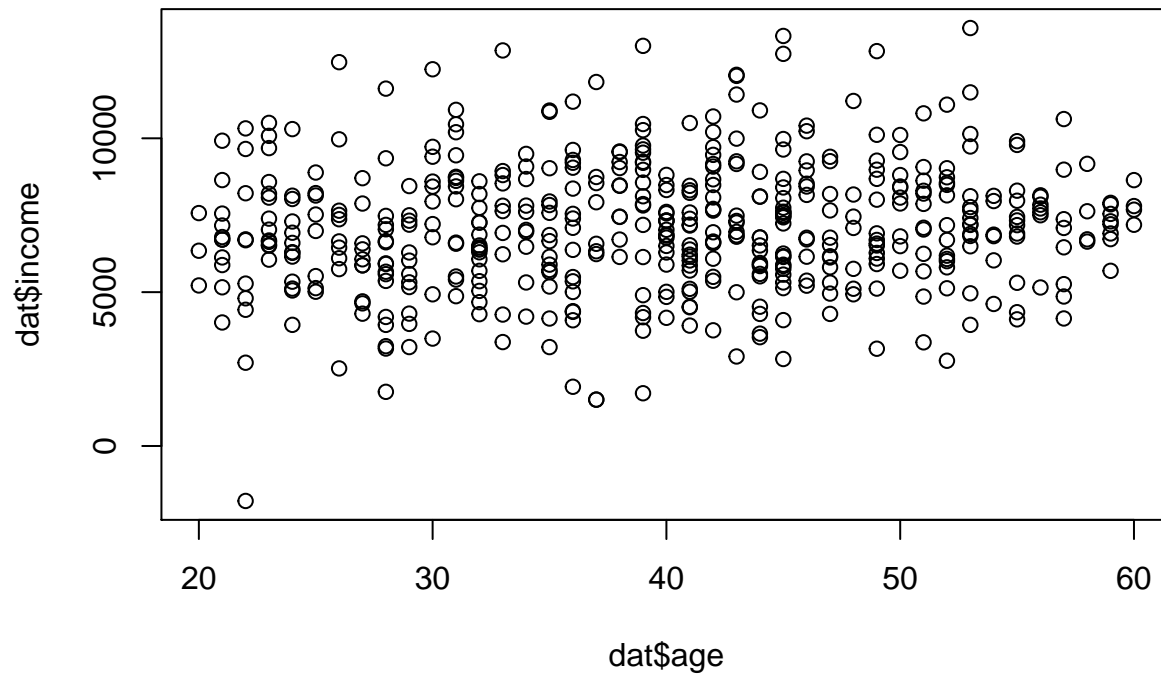
```
boxplot(income ~ region, data=dat)
```



```
boxplot(income ~ gender, data=dat)
```



```
plot(dat$age, dat$income)
```



```
fit.age <- lm(income~age, data=dat)  
summary(fit.age)
```

```
##  
## Call:  
## lm(formula = income ~ age, data = dat)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max   
## -8519.5 -1281.0   -76.8  1284.1  6183.6
```

```
##
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) 6260.262   361.873  17.300 <2e-16 ***
## age          21.512     8.858   2.429  0.0155 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2062 on 498 degrees of freedom
## Multiple R-squared:  0.0117, Adjusted R-squared:  0.00972
## F-statistic: 5.898 on 1 and 498 DF,  p-value: 0.01551
```

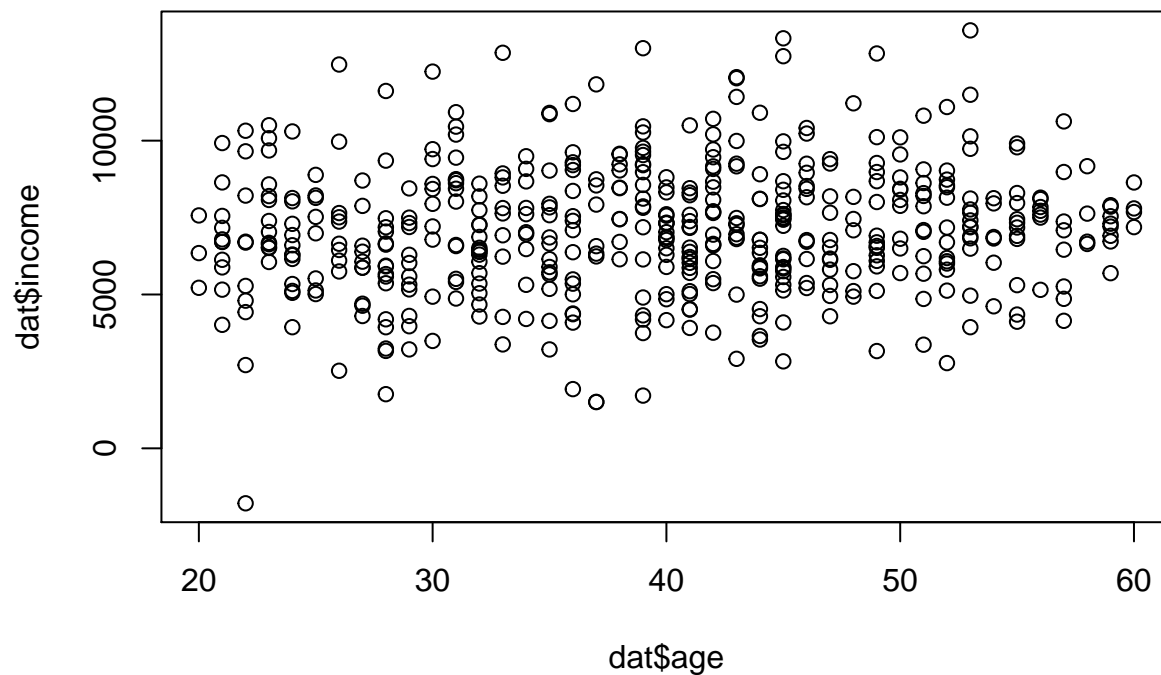
### Customizing a scatter plot

```
* help(plot.default)
* help(par)
```

### Two useful graphics functions

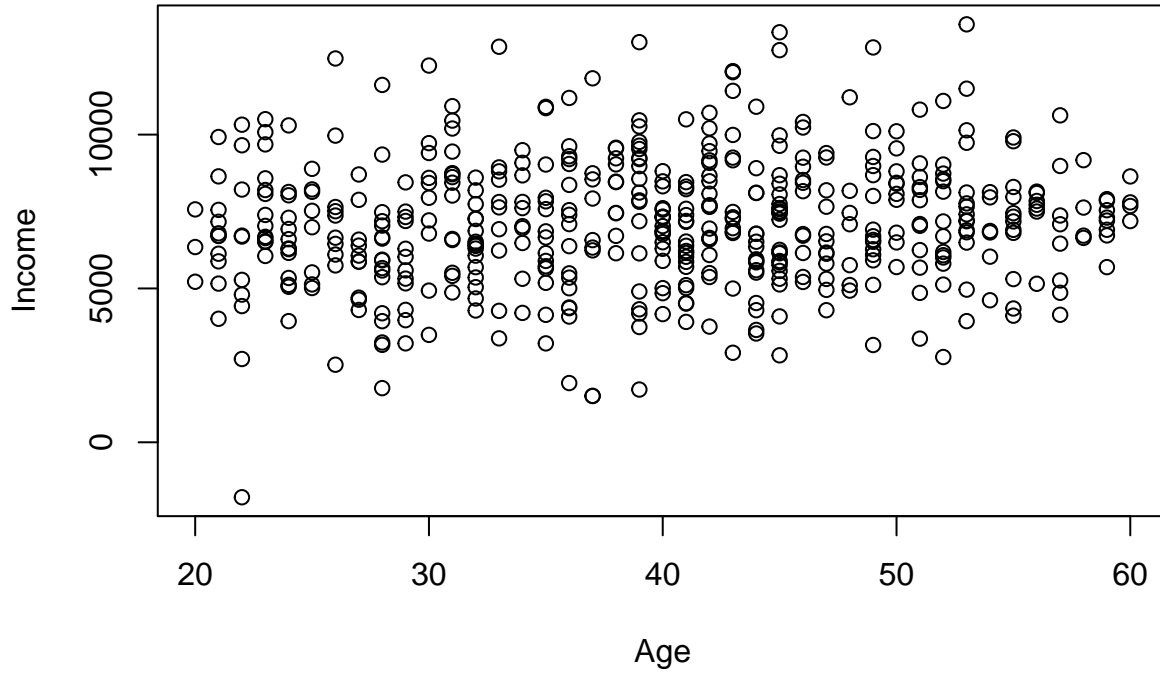
```
* x11() ### opens another graphics window
* graphics.off() ### closes all graphics devices
```

```
# Default plot
plot(dat$age,dat$income)
```



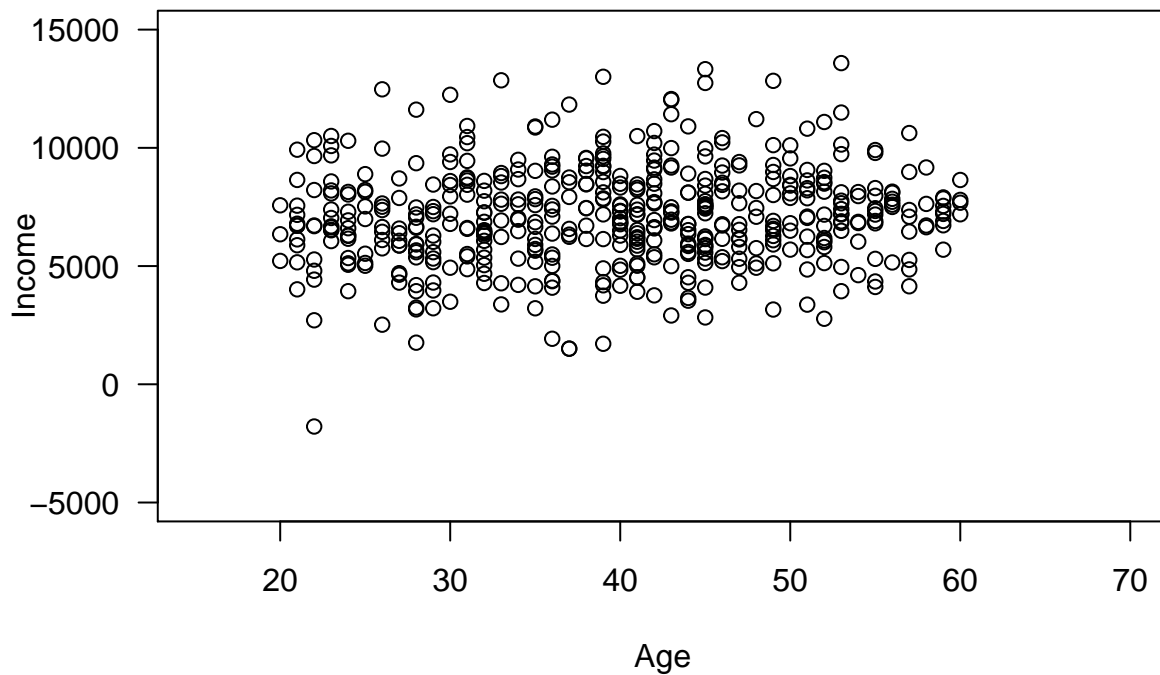
```
# Adding axis labels and title using xlab, ylab, main tags
plot(dat$age,dat$income,xlab="Age",ylab="Income",
     main="Relationship between Age and Income")
```

## Relationship between Age and Income



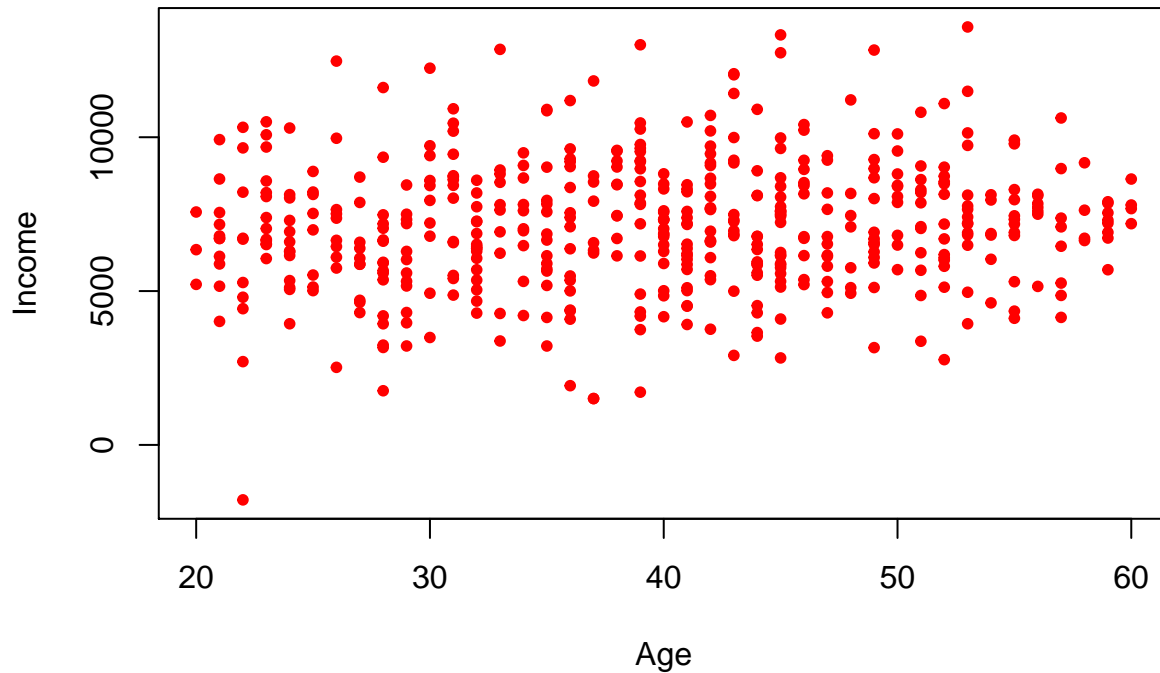
```
# Adding axis ranges and tic mark label orientation using xlim, ylim, las tags  
plot(dat$age,dat$income,xlab="Age",ylab="Income",  
      main="Relationship between Age and Income",  
      ylim=c(-5000,15000), xlim=c(15,70),las=1)
```

## Relationship between Age and Income



```
# Changing the plotting symbol and its color using pch and col tags
plot(dat$age,dat$income,xlab="Age",ylab="Income",
     main="Relationship between Age and Income",
     pch=20,col="red")
```

## Relationship between Age and Income



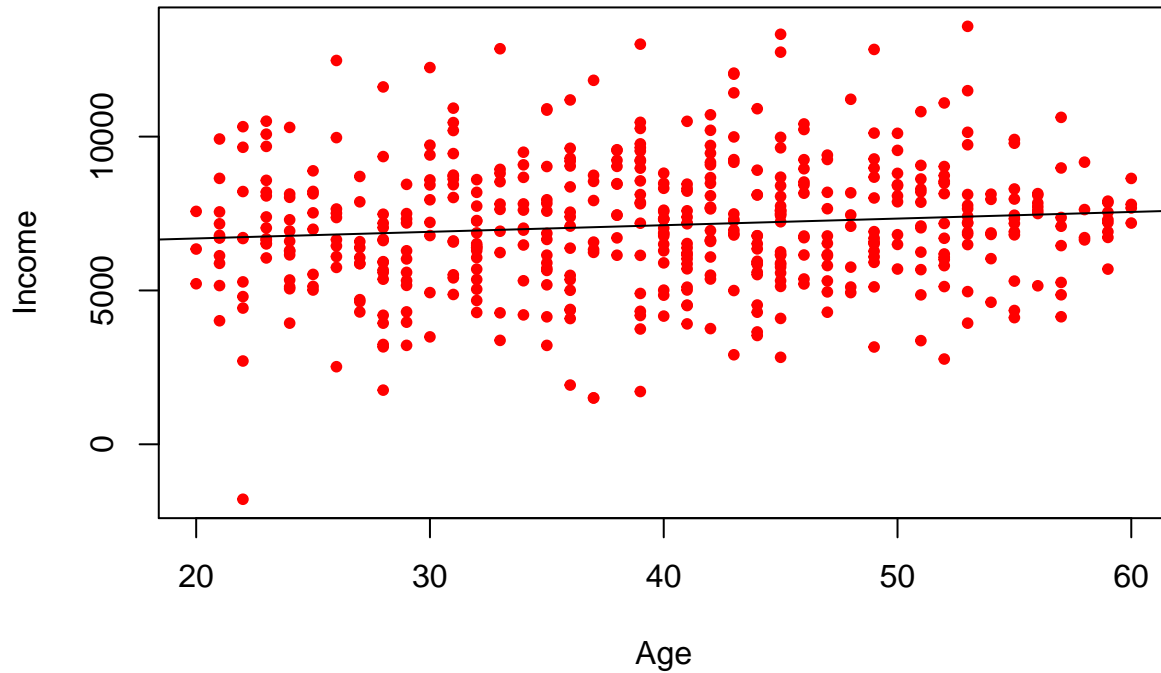
See document [pch.pdf](#) for the different plotting symbols

\* `colors()` `###` will list the colors

```
# Adding the regression line using the abline function
plot(dat$age,dat$income,xlab="Age",ylab="Income",
     main="Relationship between Age and Income",
     pch=20,col="red")

abline(fit.age)
```

## Relationship between Age and Income



Adding an arbitrary straight line using the segments function

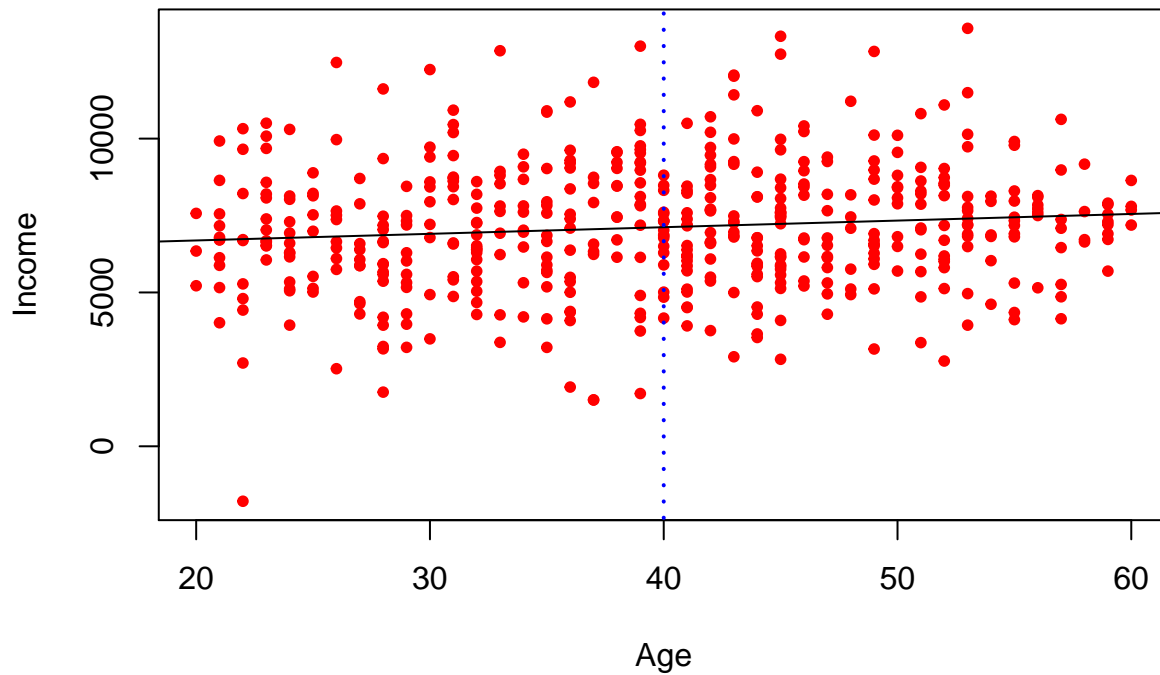
\* `help(segments)`

```
# Tag lty is the line type and lwd is the line width see help(par) for more information  
# This example adds a vertical wider dotted blue line at the median of age
```

```
plot(dat$age,dat$income,xlab="Age",ylab="Income",  
     main="Relationship between Age and Income",  
     pch=20,col="red")  
abline(fit.age)  
  
segments(x0=median(dat$age),y0=-5500,x1=median(dat$age),y1=15500,  
        lty="dotted",lwd=2,col="blue")
```



## Relationship between Age and Income



Adding points to the plot using the points function

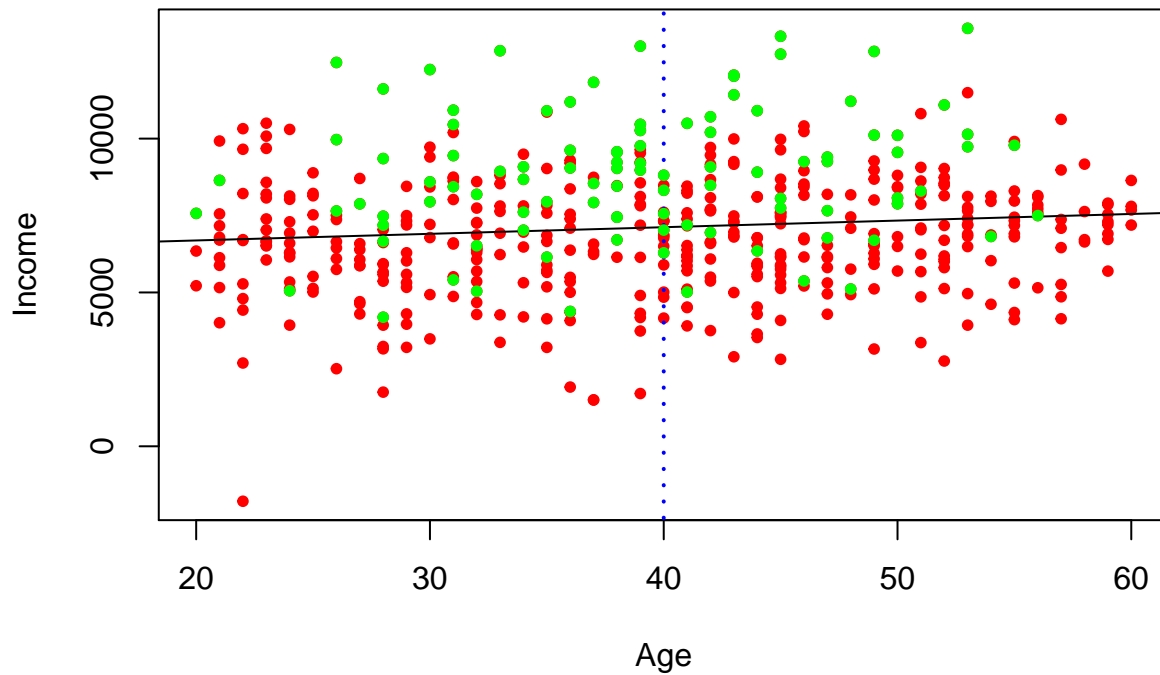
\* help(points)

*# In this example, we're actually writing over existing points, coloring the points from region CC green*

```
plot(dat$age,dat$income,xlab="Age",ylab="Income",
     main="Relationship between Age and Income",
     pch=20,col="red")
abline(fit.age)
segments(x0=median(dat$age),y0=-5500,x1=median(dat$age),y1=15500,
        lty="dotted",lwd=2,col="blue")

points(dat$age[dat$region=="CC"],dat$income[dat$region=="CC"],
       col="green",pch=20)
```

## Relationship between Age and Income



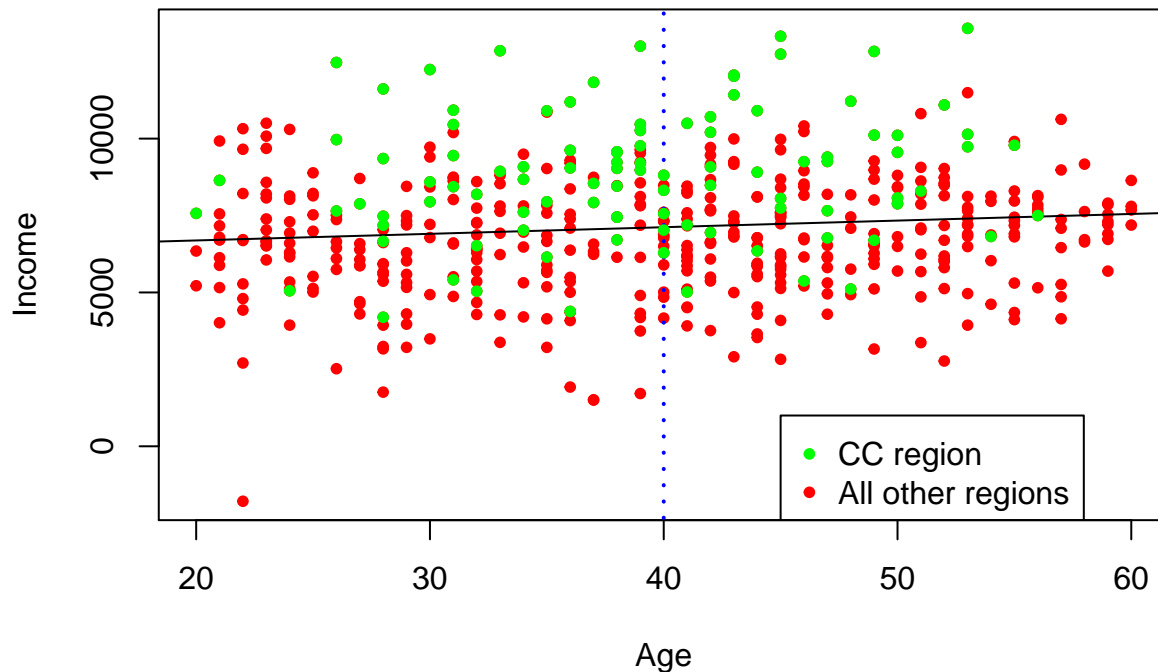
Adding a legend to the plot using the legend function

- `help(legend)`

```
plot(dat$age,dat$income,xlab="Age",ylab="Income",
     main="Relationship between Age and Income",
     pch=20,col="red")
abline(fit.age)
segments(x0=median(dat$age),y0=-5500,x1=median(dat$age),y1=15500,
        lty="dotted",lwd=2,col="blue")
points(dat$age[dat$region=="CC"],dat$income[dat$region=="CC"],
       col="green",pch=20)

legend(45,1000,col=c("green","red"),pch=20,
      legend=c("CC region","All other regions"))
```

## Relationship between Age and Income



Saving this plot into a pdf document using the pdf function and the dev.off function

- help(pdf)
- help(Devices)

```
### This creates the file "scatter.pdf" in your local directory
```

```
pdf(file="scatter.pdf",height=8,width=10)

plot(dat$age,dat$income,xlab="Age",ylab="Income",
     main="Relationship between Age and Income",
     pch=20,col="red")
abline(fit.age)
segments(x0=median(dat$age),y0=-5500,x1=median(dat$age),y1=15500,
        lty="dotted",lwd=2,col="blue")
points(dat$age[dat$region=="CC"],dat$income[dat$region=="CC"],
       col="green",pch=20)
legend(25,500,col=c("green","red"),pch=20,
      legend=c("CC region","All other regions"))

dev.off()
```

```
## pdf
## 2
```

Adding text to a plot using the *text* function

```
### First create the vectors with the information I wish to add to plot
```

```
region <- unique(dat$region)
mean.age <- NULL
```

```

for (i in 1:length(region))
{
  mean.age <- c(mean.age, mean(dat$age[dat$region==region[i]]))
}

region

## [1] AA BB CC DD EE
## Levels: AA BB CC DD EE
mean.age

## [1] 45.40 33.13 39.03 38.85 41.11

mean.income <- NULL
for (i in 1:length(region))
{
  mean.income <- c(mean.income, mean(dat$income[dat$region==region[i]]))
}

mean.income

## [1] 6536.10 7650.51 8857.31 5461.30 7045.14

### Now the plot

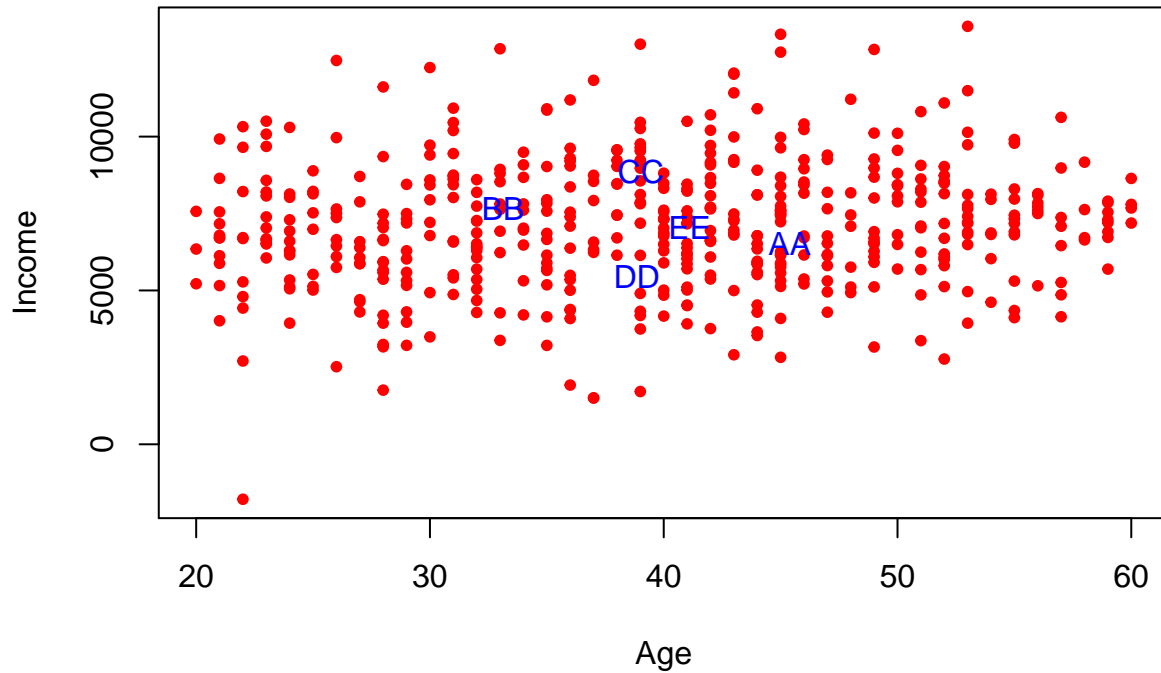
# help(text)

plot(dat$age,dat$income,xlab="Age",ylab="Income",
      main="Relationship between Age and Income",
      pch=20,col="red")

text(x=mean.age, y=mean.income, labels=region, col="blue")

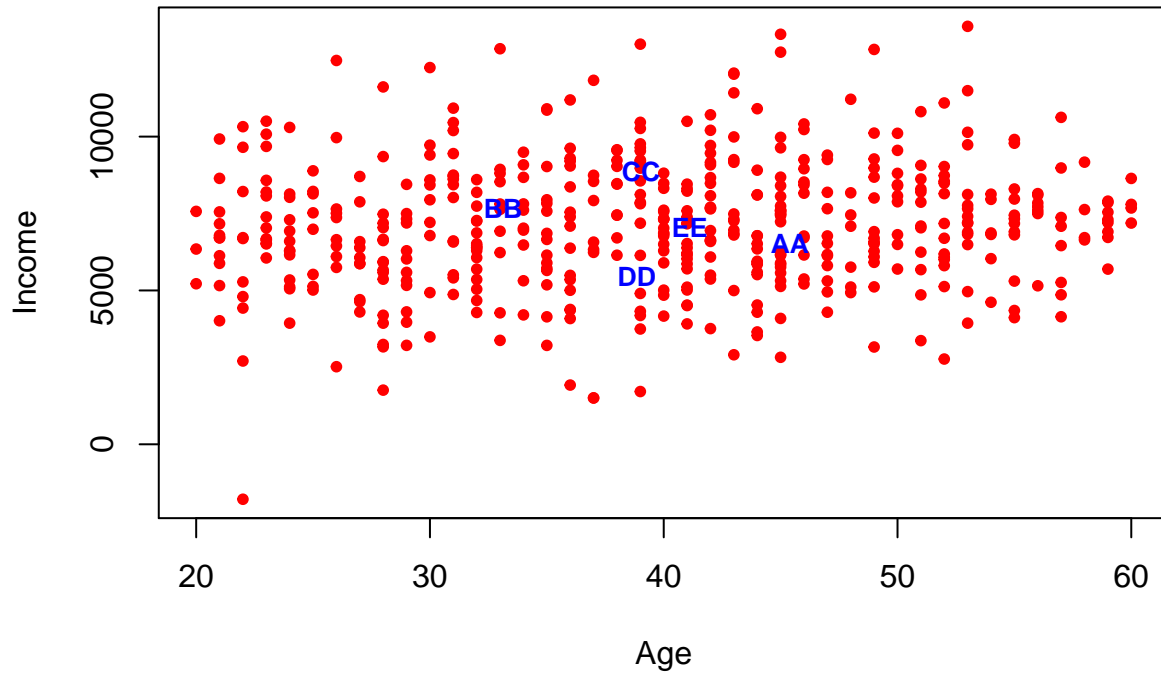
```

## Relationship between Age and Income



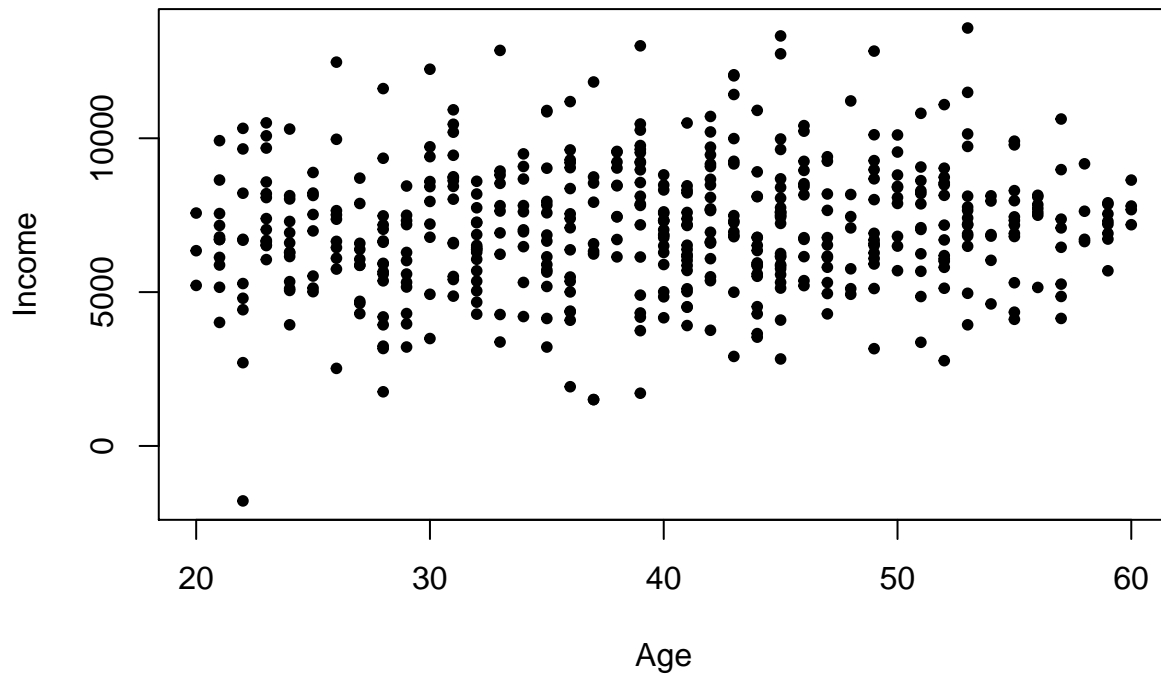
```
### Making the text smaller and bold using the cex and font tags (see help(par))  
  
plot(dat$age,dat$income,xlab="Age",ylab="Income",  
      main="Relationship between Age and Income",  
      pch=20,col="red")  
  
text(x=mean.age, y=mean.income, labels=region, col="blue", cex=0.8, font=2)
```

## Relationship between Age and Income



Customizing the y axis using the *yaxt* tag and the *axis* function

```
### Plot before customizing y axis  
plot(dat$age,dat$income,xlab="Age",ylab="Income",pch=20)
```



```
### Finding the y axis range  
min(dat$income)
```

```
## [1] -1786
```

```
max(dat$income)
```

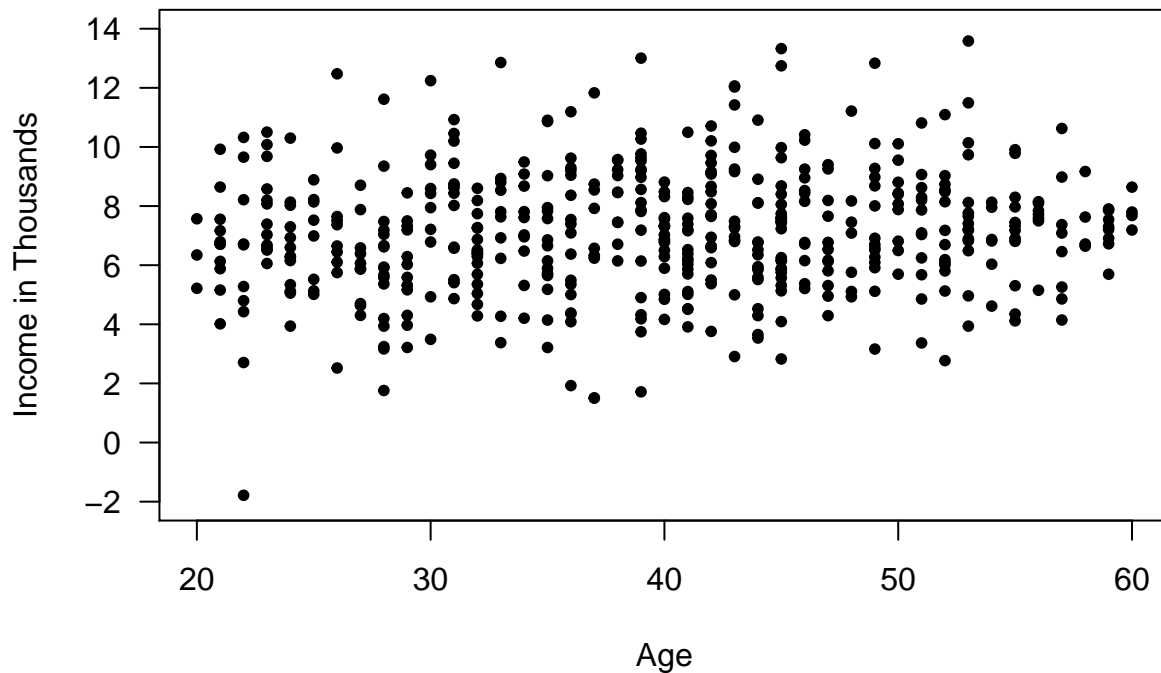
```
## [1] 13584
```

```
#help(axis)
```

```
### Plot after customizing y axis
```

```
plot(dat$age,dat$income,xlab="Age", xlim=c(20,60),pch=20,  
      ylim=c(-2000,14000), ylab="Income in Thousands",yaxt="n")
```

```
axis(side=2,at=seq(from=-2000,to=14000,by=2000),  
      labels=seq(from=-2,to=14,by=2),las=1)
```



### Spacing and page layout

\* Set placement on page via margin sizes using mai tag in par function

\* Order is bottom,left,top,right

\* mai: A numerical vector of the form c(bottom, left, top, right) which gives the margin size specified in inches" see help(par)

\* par()

\* par()\$mai

```
# First look at default
```

```
pdf(file="mean.plot.default.pdf",height=8, width=10)
```

```
plot(dat$age,dat$income,xlab="Age", xlim=c(20,60),pch=20,  
      ylim=c(-2000,14000), ylab="Income in Thousands",yaxt="n")
```

```
axis(side=2,at=seq(from=-2000,to=14000,by=2000),  
      labels=seq(from=-2,to=14,by=2),las=1)
```

```
text(x=mean.age, y=mean.income, labels=region, col="blue", cex=0.8, font=2)
```

```
dev.off()
```

```

## pdf
## 2
# Now change margins

pdf(file="mean.plot.margins.pdf",height=8, width=10)

### bottom is 2, left is 2, top is 1.5, right is 1
par(mai=c(2,2,1.5,1))

plot(dat$age,dat$income,xlab="Age", xlim=c(20,60),pch=20,
      ylim=c(-2000,14000), ylab="Income in Thousands",yaxt="n")

axis(side=2,at=seq(from=-2000,to=14000,by=2000),
      labels=seq(from=-2,to=14,by=2),las=1)

text(x=mean.age, y=mean.income, labels=region, col="blue", cex=0.8, font=2)

dev.off()

```

```

## pdf
## 2

```

### Multiple plots on the same page using *par(mfrow)*

*help(par)*

- `mfc`, `mfrow`
- A vector of the form `c(nr, nc)`. Subsequent figures will be drawn in an `nr`-by-`nc` array on the device by columns (`mfc`), or rows (`mfrow`), respectively.
- In a layout with exactly two rows and columns the base value of “`cex`” is reduced by a factor of 0.83: if there are three or more of either rows or columns, the reduction factor is 0.66.
- If either of these is queried it will give the current layout, so querying cannot tell you the order the array will be filled.
- Consider the alternatives, `layout` and `split.screen`.

```

pdf(file="multiple1.pdf",height=5,width=10)

### one row by two columns
par(mfrow=c(1,2))

### left plot
plot(dat$age,dat$income,xlab="Age",ylab="Income",
      main="Relationship between Age and Income",
      pch=20,col="red")
text(x=mean.age, y=mean.income, labels=region, col="blue", cex=0.8, font=2)

### right plot
barplot(table(dat$gender,dat$region),beside=T,las=1,xlab="Region",
         col=c("red","green","blue"), ylim=c(0,70))
legend(1,60, fill=c("red","green","blue"),
       legend= c("Female","Male","Nux"), cex=0.7 )
text(1,65,"Distribution of Gender by Region",pos=4)

dev.off()

```

```

## pdf

```



```

## 2
### using a loop
pdf(file="multiple2.pdf",height=8,width=10)

### two rows by three columns
par(mfrow=c(2,3))

### get the five regions "AA" thru "FF"
region <- unique(dat$region)

### use loop to do the same thing for each region
for (i in 1:length(region))
{
  ### get data for just one region and store in "temp"
  temp <- dat[dat$region==region[i],]

  ### get the regression line fit
  fit <- lm(income~age,data=temp)

  ### plot data points
  plot(temp$age,temp$income,xlab="Age", xlim=c(20,60),pch=20,
        ylim=c(-2000,14000), ylab="Income in Thousands",yaxt="n",
        main=paste("Region ",region[i],sep=""))

  ### custom y-axis
  axis(side=2,at=seq(from=-2000,to=14000,by=2000),
        labels=seq(from=-2,to=14,by=2),las=1)

  ### add regression line
  abline(fit$coef,col="red")
}

### close pdf file
dev.off()

```

```

## pdf
## 2

```

### Multiple page pdf using same loop

```

pdf(file="multiple3.pdf",height=8,width=10)

### get the five regions "AA" thru "FF"
region <- unique(dat$region)

### use loop to do the same thing for each region
for (i in 1:length(region))
{
  ### get data for just one region and store in "temp"
  temp <- dat[dat$region==region[i],]

  ### get the regression line fit
  fit <- lm(income~age,data=temp)

  ### plot data points

```

```
plot(temp$age,temp$income,xlab="Age", xlim=c(20,60),pch=20,
      ylim=c(-2000,14000), ylab="Income in Thousands",yaxt="n",
      main=paste("Region ",region[i],sep=""))

### custom y-axis
axis(side=2,at=seq(from=-2000,to=14000,by=2000),
      labels=seq(from=-2,to=14,by=2),las=1)

### add regression line
abline(fit$coef,col="red")
}

### close pdf file
dev.off()

## pdf
## 2
```