Introduction to R

Karthik Sriram
Indian Institute of Management Ahmedabad
karthiks@iima.ac.in

Outline

The R package

- R is a language built for statistical analysis and graphics
- Actively used in academic research and also in industry
- Incorporates advanced methods and latest techniques
- It is freely downloadable !
- Lot of help available online for R usage.

Downloading R and R-studio

- Download R from the CRAN website
  `https://cran.r-project.org/`
- Install on your machine
- Download the user interface R-Studio
  `https://www.rstudio.com/products/RStudio/#Desktop`

  Note: Download the free version.
- Install R-Studio on your machine

Basic Operations

Let us start with basic operations by typing these expressions in the command prompt (shows as >) in the R-studio Console (see left hand bottom window within R -studio)

```
# Addition
2+3
#Subtraction
2-3
#Multiplication
2*3
#Division
2/3
#To the power
2^3
```

Assigning outcome to a variable

Type the following expressions in the R-studio console

```
x<-2+3
x

## [1] 5
```

Assigning outcome to a variable

Type the following expressions in the R-studio console

```
x<-2+3
x

## [1] 5
```

Creating vectors

Type the following expressions in the R-studio console

```
x<-c(10,20,30,40,50,60,70)
x
```

Creating vectors

Type the following expressions in the R-studio console

```
x<-c(10,20,30,40,50,60,70)
x

## [1] 10 20 30 40 50 60 70

## Accessing 2nd entry
x[2]

## [1] 20
```

Creating vectors

Type the following expressions in the R-studio console

```
x<-c(10,20,30,40,50,60,70)
x

## [1] 10 20 30 40 50 60 70

x<- c(1, 19, 20, 11, 111)

## Accessing 2nd entry
x[2]

## [1] 19
```

Creating Sequence of numbers

Type the following expressions in the R-studio console

```
#using the seq function
x<-seq(10,70, by=10)
x
x<- seq(10,70, length=7)
x
```

Creating vector of numbers

Type the following expressions in the R-studio console

```
#using the seq function
x<-seq(10,70, by=10)
x

## [1] 10 20 30 40 50 60 70

x<- seq(10,70, length=7)
x

## [1] 10 20 30 40 50 60 70

x<-c(10,20,30,40,50,60,70)
x

## [1] 10 20 30 40 50 60 70
```

Help in R

First determine what concept or procedure you need help on e.g. seq

Use help() function or ? in the R-studio console.

```
using the seq function
help(seq)
 or
?seq
```

Vector operations

Type the following in the console

```
s1<-seq(1,20, length=5)

s2<-seq(40,60, length=5)

# Vector Addition

s1+s2

## [1] 41.00 50.75 60.50 70.25 80.00

# element-wise multiplication
s1*s2

## [1]   40.00  258.75  525.00  838.75 1200.00
```

Vector summaries

Type the following in the console

```
# Sum

sum(s1)

## [1] 52.5

#Mean

mean(s1)

## [1] 10.5

# sumproduct

sum(s1*s2)

## [1] 2862.5
```

Vector Multiplication

Type the following in the console

```
# sumproduct

sum(s1*s2)

## [1] 2862.5

# By dot product
t(s1) %*% s2

##         [,1]
## [1,] 2862.5
```

Vector Multiplication

Type the following in the console

```
# sumproduct

sum(s1*s2)

## [1] 2862.5

# By dot product
t(s1) %*% s2

##         [,1]
## [1,] 2862.5
```

Logical Operation

Type the following in the console

```
# 1.

(2<1)

## [1] FALSE

# 2.
(1<2)

## [1] TRUE

# 3.

(2<1)*1

## [1] 0

# 4.

(1<2)*1

## [1] 1
```

Logical Operation with vectors

Type the following in the console

```
# 5.
x<-c(1,2,3,4,5,6,23,4,5,6,7,8,9,10)
(x<4)

##  [1]  TRUE  TRUE  TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## [12] FALSE FALSE FALSE

(x<4)*1

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

# How many elements of x are less than 4 ?

sum((x<4)*1)

## [1] 3
```

Logical Operation with vectors

Type the following in the console

```
# 5.
x<-c(1,2,3,4,5,6,23,4,5,6,7,8,9,10)

# How many elements of x are greater than 4 but less than 8 ?

sum((x>4) & (x<8))

## [1] 5

# How many elements of x are either equal to 4 or less greater than 6 ?

sum((x==4) | (x>6))

## [1] 7
```

Exercise 1

Interest rate on a bond with face value Rs 10,000 and a 10 year term is 10% per annum. What is the maturity value of the bond?

Exercise 2

Every day an insurance company processes and releases payments on damages for insured cars, whose policies have a deductible of Rs 1000 and a Limit of Rs 100,000. Only amount in excess of the deductible is paid and maximum paid is 100,000. On one day, the damage assessments for 10 cars (in Rs) is reported as follows:

10000, 10500, 20000, 40000, 50000, 120000, 130000, 200000, 250000, 250000

What is the total payment released by the company?

Functions

Repetitive formulaic computations based on some inputs can be coded a function and called later. For example, let us define the function $f(x) = x^2 + x^3$.

```
# Defining the function squarepluscube
f<- function(x){
  x^2+x^3
}

# Calling
f(3)

## [1] 36
```

Exercise 1 - continued

Write a function to compute the maturity value of a bond with face value V (Rs.), term T (years) and an annual interest rate r (in %).

Exercise 1 - continued

Write a function to compute the maturity value of a bond with face value V (Rupees), term T (years) and an annual interest rate r (in %).

```
Maturity<-function(V, T, r){
  M<- V* (1+r/100)^T
  M   # This needs to be typed so that value is returned by the function
}

Maturity(100,10, 10)

## [1] 259.3742
```

Exercise 3

An airline with 100 seats on an aircraft usually overbooks when possible. If the number of people who show up exceeds the number of seats, the airline pays a penalty of Rs. 5000 for each person who cannot board the aircraft. On 10 different days, the number of people who show up for the flight is as follows 110, 105, 99, 90, 111, 95, 100, 98, 101, 102. What is the total penalty paid over the 10 days ?.

Setting Working Directory

We map a folder on our computer as a working directory. This means that the codes, figures and other output, when saved will be automatically saved in this folder.

NOTE: replace backslash with forward slash / while specifying location

```
# To know what is the default working directory

getwd()

# Setting a Working Directory using setwd()


setwd('C:/Users/Admin/Google Drive/EPAPB/RSessions')

getwd()
```

Creating basic data frame manually

```
testdata<- data.frame()

#editing the data frame

fix(testdata)
```

A data frame should appear in the R-Studio screen. Enter data into it.

Adding columns

```
coffee<- c(3,0,2,2, 0)

testdata<-cbind(testdata,coffee)

View(testdata)                #Note: V is in capitals


#editing the data frame

fix(testdata)
```

An empty data frame will appear in the R-Studio screen. Enter data into it.

Basic Summaries

```
summary(testdata)
```

```
##     Name     Experience   gender     coffee
##  Name1:1   Min.   : 5.0   F:3    Min.   :0.0
##  Name2:1   1st Qu.: 5.0   M:2    1st Qu.:0.0
##  Name3:1   Median : 6.0          Median :2.0
##  Name4:1   Mean   : 8.2          Mean   :1.4
##  Name5:1   3rd Qu.:10.0          3rd Qu.:2.0
##            Max.   :15.0          Max.   :3.0
```

```
table(coffee)
```

```
## coffee
## 0 2 3
## 2 2 1
```

```
with(testdata,table(coffee, gender))
```

```
##       gender
## coffee F M
##      0 2 0
##      2 1 1
##      3 0 1
```

Frequency Distribution

```
# Univariate
table(coffee)

## coffee
## 0 2 3
## 2 2 1

# Bi-variate
with(testdata,table(coffee, gender))

##       gender
## coffee F M
##      0 2 0
##      2 1 1
##      3 0 1
```

Import Data

Most convenient approach is to store the data file as a CSV file in the working directory. Run the following command which imports a dataset

```
setwd('C:/Users/Admin/Google Drive/EPAPB/RSessions')

data1<- read.csv('Percentage_of_households_DLHS4.csv')
```

Inspect the data

```
head(data1)

names(data1)
```

Renaming of columns

```
colnames(data1)<- c("States", "Districts", "pct_electricity",
                    "pct_drinkingwater", "pct_toiletfacility", "pct_cookingfuel
```

Selecting rows and columns by number

```
# Select rows 1 & 4 ,  and columns 2 & 3

data1[c(1,4), c(2,3)]
```

Selectiong rows and columns by number

```
# Select rows 1 & 4 ,  and columns 2 & 3

data1[c(1,4), c(2,3)]
```

Identifying rows with some condition

```
# Select rows 1 & 4 ,  and columns 2 & 3

ll<- with(data1, which(States=="Andhra Pradesh"))

ll

##  [1]  5  6  7  8  9 10 11 12 13 14 15 16 17 18
```

Subset of data based on conditions

```
### select records for Andhra , Karnataka


ll<-with(data1, which(States %in% c("Andhra Pradesh", "Karnataka")))

subdata1<-data1[ll,]
```

Exercise 1

Which districts have percentage drinking water accessible to more than 98% of the people ?

```
ll<- with(data1, which(pct_drinkingwater>=98))

with(data1, Districts[ll])
```

Exercise 2

which states have at least 1 district with atleast 98% drinking water ?

```
ll<- with(data1, which(pct_drinkingwater>=98))

unique(with(data1, States[ll]))

##  [1] Andhra Pradesh   Chandigarh       Goa              Haryana
##  [5] Himachal Pradesh Karnataka        Kerala           Maharashtra
##  [9] Pudducherry      Punjab           Tamil Nadu       Telangana
## [13] West Bengal
## 21 Levels: Andaman and Nicobar Islands ... West Bengal
```

Exercise 3

Suppose we want to compute mean and standard deviation of pct_drinkingwater by state.

```
# prepare state level summaries for Mean
Inter_dataM<- aggregate(pct_drinkingwater~ States,data=data1, FUN=mean)

# prepare state level summaries for SD
Inter_dataSD<- aggregate(pct_drinkingwater~ States,data=data1, FUN=sd)


# compute mean, sd, min , max by state and merge

merge(Inter_dataM, Inter_dataSD, by.x="States", by.y="States")
```

Exercise 4

Suppose I wanted SD to be computed only for states with atleast 5 records.

Exercise 4: Solution

Suppose I wanted SD to be computed only for states with atleast 5 records.

```
# First prepare a summary to find out how many records are there by state

Inter_dataCount<- aggregate(pct_drinkingwater~ States,data=data1, FUN=length)

## Determine list of states which have more than 5 records

ll<-with(Inter_dataCount, which(pct_drinkingwater>=5))

liststst<-Inter_dataCount$States[ll]

## Obtain SD for only those states in the list obtained

ll<-with(Inter_dataSD, which(States %in% liststst))

length(ll)

Inter_dataSD1<-Inter_dataSD[ll,]

mergeddata<-merge(Inter_dataM, Inter_dataSD1, by.x="States",
                  by.y="States", all.x=TRUE)
```

Exporting Data

Suppose I wanted SD to be computed only for states with atleast 5 records.

```
## Use all.x=FALSE

mergeddata<-merge(Inter_dataM, Inter_dataSD1, by.x="States",
                  by.y="States", all.x=FALSE)


## Run this and change the column names
fix(mergeddata)

write.csv(mergeddata, "outputdata.csv")
```

Import data

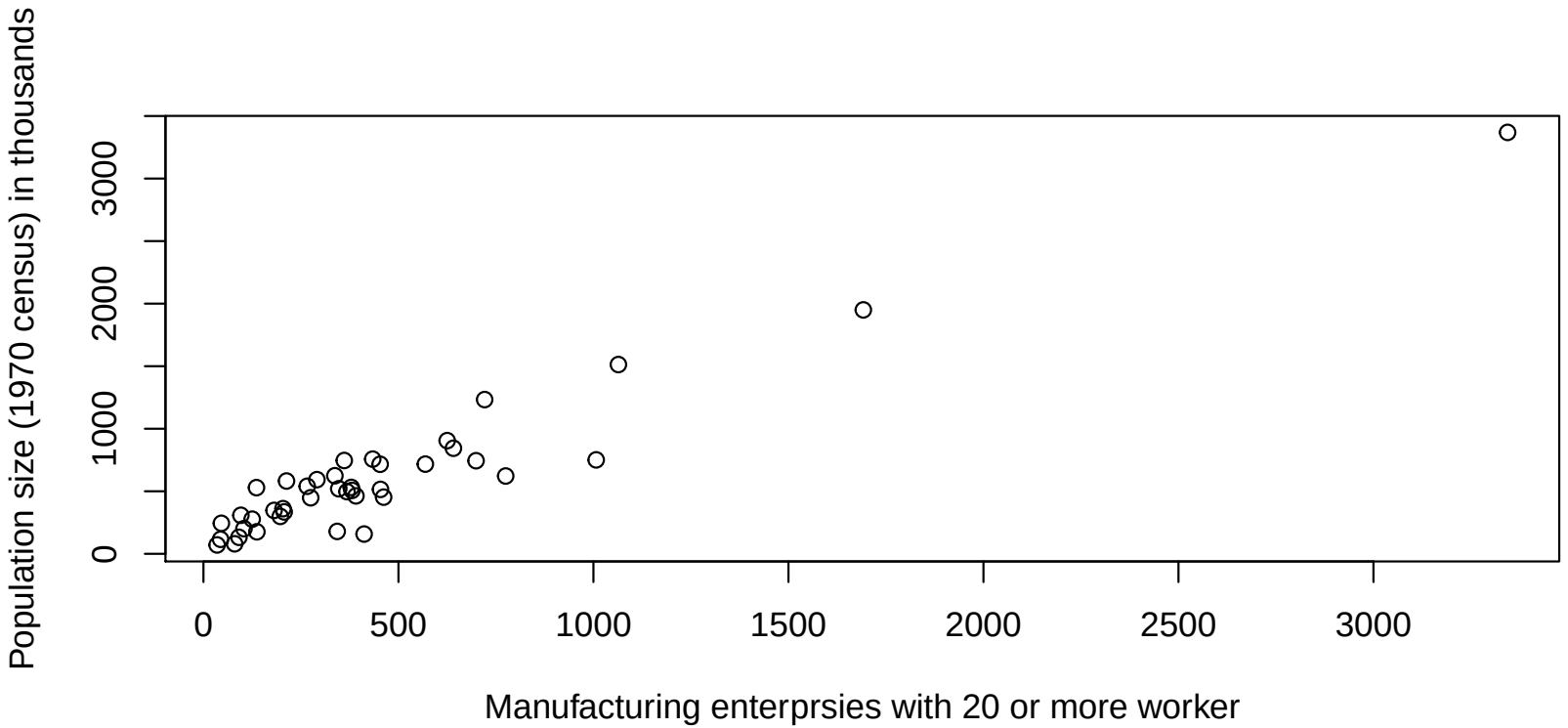
```
setwd('C:/Users/Admin/Google Drive/EPAPB/RSessions')

Data2<-read.csv("USairpollution.csv")
```

Inspect Data

```
dim(Data2)

head(Data2)
```

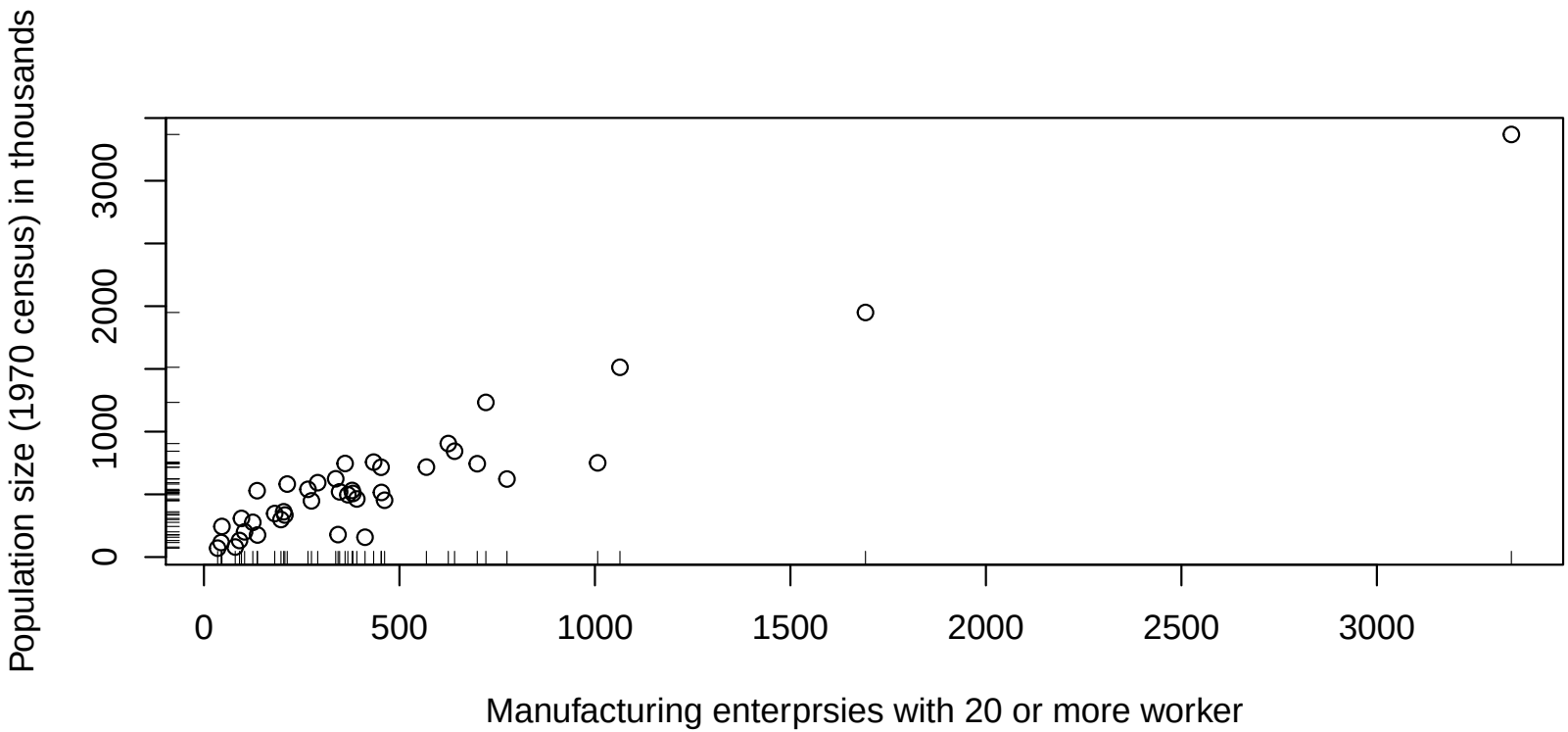
Scatter Plot

```
mlab<-"Manufacturing enterprsies with 20 or more worker"

plab<- "Population size (1970 census) in thousands"

plot(popul~ manu, data=Data2, xlab=mlab, ylab=plab)
```



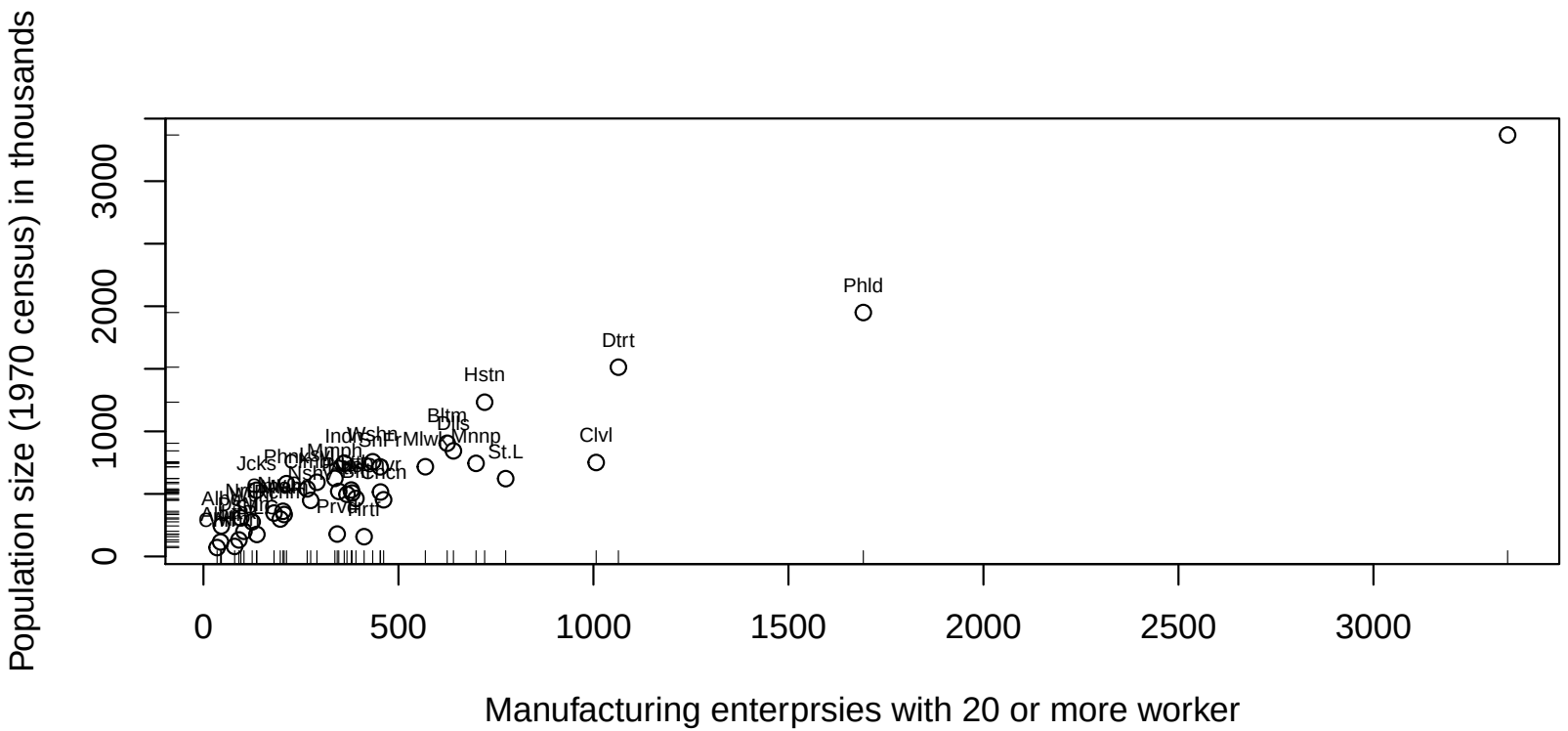
Scatter Plot + Rug plot

```
#show marginal dist
plot(popul~ manu, data=Data2, xlab=mlab, ylab=plab)
rug(Data2$manu, side=1)
rug(Data2$popul, side=2)
```



Scatter Plot + Rug plot + Labels

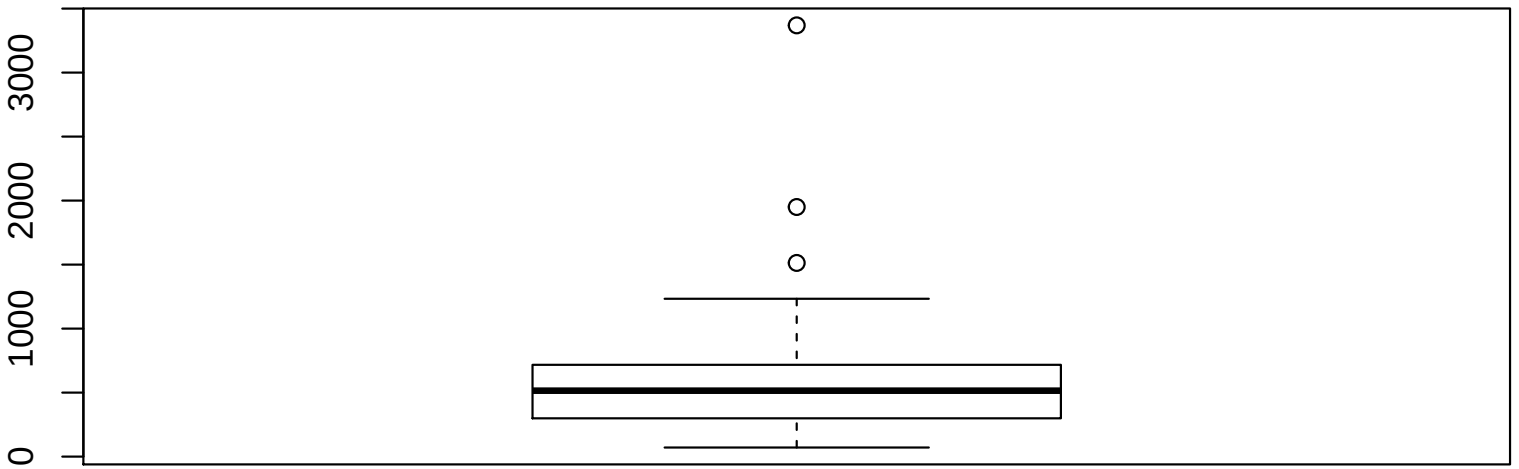
```
plot(popul~ manu, data=Data2, xlab=mlab, ylab=plab)
rug(Data2$manu, side=1)
rug(Data2$popul, side=2)
with(Data2, text(manu, popul, cex=.6,pos=3, labels=abbreviate(Data2[,1])))
```



Box plot

A plot that shows Mean and quartiles (we will learn later)

```
with(Data2,boxplot(popul)) # univariate box plot
```



More in 2D plots

Can we plot more than 2 dimensions on X-Y axis?

Run this code

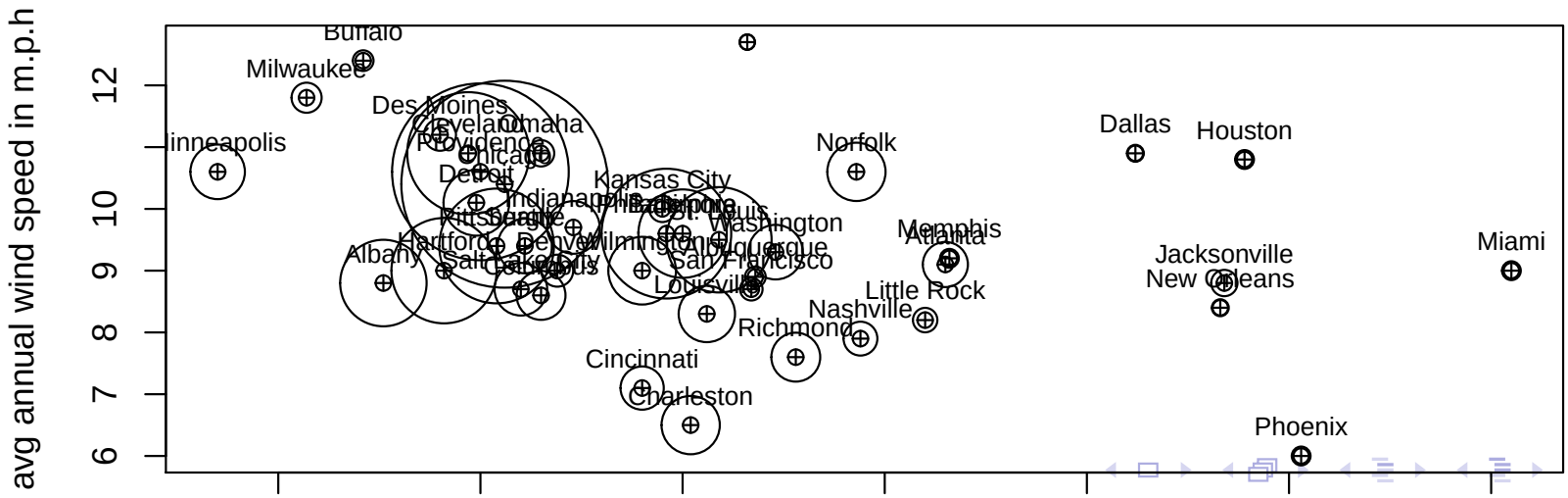
```
plot(wind~temp, data=Data2, xlab="Avg annual temperature in Fahrenheit",
     ylab="avg annual wind speed in m.p.h",
     pch=10, ylim=range(Data2$wind), xlim=range(Data2$temp))
# introduce bubble showing SO2 levels
with(Data2, symbols(temp, wind, circles=SO2, inches=.5, add=TRUE))
with(Data2, text(temp, wind, cex=.75,pos=3, labels=Data2[,1]))
```

More in 2D plots

Can we plot more than 2 dimensions on X-Y axis?

Run this code

```
plot(wind~temp, data=Data2, xlab="Avg annual temperature in Fahrenheit",
     ylab="avg annual wind speed in m.p.h",
     pch=10, ylim=range(Data2$wind), xlim=range(Data2$temp))
# introduce bubble showing SO2 levels
with(Data2, symbols(temp, wind, circles=SO2, inches=.5, add=TRUE))
with(Data2, text(temp, wind, cex=.75,pos=3, labels=Data2[,1]))
```



Saving a Figure

After the plot command is run, do the following to save tha active plot

```
dev.copy(pdf,"Figuretest.pdf")
dev.off()
```

The active plot will be stored in the working directory as a pdf file.
One may also use post script("xx.ps") or jpeg ("xx.jpg").

3D plot

After the plot command is run, do the following to save tha active plot

```
## No need to run install.packages,  if already done
# install.packages(dependencies=TRUE, "scatterplot3d")

require(scatterplot3d)

## Warning:  package 'scatterplot3d' was built under R version 3.2.5

with(Data2, scatterplot3d(temp, wind, SO2, type="h", angle=55))
```