```
setwd("F:\\Work\\Jigsaw Academy\\Corporate Trainings\\Dat Manipulation")
oj <- read.csv("oj.csv")
oj
View(oj)

str(oj)

#dataframe[rows,columns]
oj[3,3]

oj[c(1,2,8,456),c(1,3,6)]


oj[c(1:5),"brand"]


#Logical Subseting

#Selecting only those rows where brand bought is tropicana
dat<-oj[oj$brand=='tropicana',]


#Using Or condition, brand bought is tropicana or dominicks
dat1<-oj[oj$brand=='tropicana'|oj$brand=='dominicks',]
head(dat1)



#Using And condition, brand bought is tropicana and no feature advertisement is run
dat2<-oj[oj$brand=='tropicana' & oj$feat==0,]
head(dat2,10)
```

```
#Subsetting using which() operator
ind<-which(oj$brand=="dominicks")
ind
class(ind)
head(ind)
dat3<-oj[ind,]


#Selecting Columns
dat4<-oj[,c("week","brand")]
head(dat4)


#Selecting+Subsetting
dat5<-oj[oj$brand=='tropicana' & oj$feat==0,
        c("week","store")]
head(dat5)



#Adding new columns
oj$logInc<-log(oj$INCOME)

dim(oj)
View(oj)
```

```
oj1 <- oj[,-18]
View(oj1)



#Revenue Column
head(oj$logmove)
head(exp(oj$logmove))
oj$revenue<-exp(oj$logmove)*oj$price

oj$revenue
View(oj)



#Sorting data
numbers<-c(10,100,5,8)
order(numbers)
order(-numbers)


dat6<-oj[order(oj$week),]
head(dat6)
min(oj$week)


dat7<-oj[order(-oj$week),]
head(dat7)
max(oj$week)
```

```
##Group by summaries

class(oj$brand)
unique(oj$brand)



#Summarize-Price
#Summarize by-Brand (factor)
#Summarize how-Mean

#Syntax aggregate(variable to be summarized,
by=list(variable by which grouping is to be done),function)



aggregate(oj$price,by=list(oj$brand),mean)
aggregate(oj$price,by=list(oj$brand,oj$feat),mean)



tapply(oj$price,oj$brand,sd)
class(tapply(oj$price,oj$brand,mean))


#Mean income of people by brand
#Summarize-Income
#Summarize by-Brand
#Summarize how-Mean
aggregate(oj$INCOME,by=list(oj$brand),mean)
class(aggregate(oj$INCOME,by=list(oj$brand),mean))
tapply(oj$INCOME[oj$INCOME<=10.5&oj$brand!='dominicks']
      ,oj$brand[oj$INCOME<=10.5&oj$brand!='dominicks'],mean)
```

```
class(tapply(oj$INCOME,oj$brand,mean))




#dplyr
install.packages("dplyr")

library(dplyr)
dat8<-filter(oj,brand=="tropicana")
dim(filter(oj,brand=="tropicana"))



dat9<-filter(oj,brand=="tropicana"|brand=="dominicks")
dim(filter(oj,brand=="tropicana"|brand=="dominicks"))




#Selecting Columns
dat10<-select(oj,brand,INCOME,feat)
dat10


dat11<-select(oj,-brand,-INCOME,-feat)



#Creating a new column
dat12<-mutate(oj,logIncome=log(INCOME),sqrtInc=sqrt(INCOME))
View(dat12)
```

```
#Arranging data
dat13<-arrange(oj,INCOME)
dat13

View(dat13)


dat14<-arrange(oj,desc(INCOME),)
View(dat14)


dat14<-arrange(oj,-INCOME)




#Group Wise summaries
gr_brand<-group_by(oj,brand)


summarize(gr_brand,mean(INCOME),sd(INCOME))


class(gr_brand)
group<-as.data.frame(gr_brand)
class(group)
print(group)




#Pipelines
#Base R code
mean(oj[oj$INCOME>=10.5,"price"])
```

```
#dplyr code
summarize(filter(oj,INCOME>=10.5),mean(price))


oj%>%filter(price>=2.5)%>%mutate(logIncome=log(INCOME))
%>%summarize(mean(logIncome),
          median(logIncome),sd(logIncome))

##Date
fd<-read.csv("Fd.csv")
str(fd)
dim(fd)
class(fd)

library(lubridate)
fd$FlightDate<-dmy(fd$FlightDate)



head(months(fd$FlightDate))
unique(months(fd$FlightDate))
head(weekdays(fd$FlightDate))
unique(weekdays(fd$FlightDate))

#Finding time interval
fd$FlightDate[60]-fd$FlightDate[900]
difftime(fd$FlightDate[3000],fd$FlightDate[90],units = "weeks")
difftime(fd$FlightDate[3000],fd$FlightDate[90],units = "days")
difftime(fd$FlightDate[3000],fd$FlightDate[90],units = "hours")
#Subsetting data based on time information
library(dplyr)
#Subset the data for day=Sunday
dim(fd)
fd_s<-fd%>%filter(weekdays(FlightDate)=="Sunday")
```

```
dim(fd_s)
#Find the number of flights on Sundays for destination Atlanta
fd%>%filter(weekdays(FlightDate)=="Sunday",
          DestCityName=="Atlanta, GA")%>%nrow()
#Find the number of flights on Sundays by cities
fd%>%filter(weekdays(FlightDate)=="Sunday")%>%group_by(DestCityName)%>
%summarize(n())
#Merging data
##Joins using Merge
df1 = data.frame(CustomerId=c(1:6),Product=c(rep("Toaster",3),
                                      rep("Radio",3)))
df1
df2 = data.frame(CustomerId=c(2,4,6),
              State=c(rep("Alabama",2),rep("Ohio",1)))
df2

merge(x = df1, y = df2, by = "CustomerId", all = TRUE)#Outer join:

merge(x = df1, y = df2, by = "CustomerId", all.x=TRUE)#Left join

merge(x = df1, y = df2, by = "CustomerId", all.y=TRUE)#Right join

merge(x=df1,y=df2,by="CustomerId")#Inner Join/Intersection of both tables

#Missing values
a<-c(1,2,3,4,5,6,NA,NA,NA,7,8,9)
is.na(a)
sum(is.na(a))
mean(a, na.rm=TRUE)

air<-airquality
head(air)

sum(is.na(air$Ozone))
sum(is.na(air$Solar.R))

summary(air)

#Imputing Missing values

air$Ozone[is.na(air$Ozone)]<-45

air$Solar.R[is.na(air$Solar.R)]<-mean(air$Solar.R,na.rm=TRUE)

summary(air)



#Keep in mind the compatibility of the classes returned and accepted by various functions



#String manipulation

a<-"Batman"

substr(a,start=2,stop=6)

nchar(a)

tolower(a)

toupper(a)


b<-"Bat-Man"
```

```
strsplit(b,split="-")

c<-"Bat/Man"

strsplit(c,split="/")

paste(b,split=c)


grep("-",b)

grepl("/",c)


sub("-","/",b)

d<-"Bat-Ma-n"


sub("-","/",d)




gsub("-","/",d)

dat5<-read.csv("F:\\Work\\Jigsaw Academy\\Corporate Trainings\\Dat Manipulation\\Strings.csv")
str(dat5)
head(dat5)#is there something wrong?
mean(dat5$Income_M)#Why will this happen

#Need to clean the data
```

```
dat5$Income_M<-gsub("Rs","",dat5$Income_M)
head(dat5)

dat5$Income_M<-gsub("/-","",dat5$Income_M)
head(dat5)
mean(dat5$Income_M)#Now why an error?

str(dat5)

dat5$Income_M<-as.numeric(dat5$Income_M)
mean(dat5$Income_M)

#Sometimes you might need to use Regexes to work with character data you can refer to this link http://www.zytrax.com/tech/web/regex.htm

x<-paste("$",seq(1,100,10))
x
#How to remove $?
x<-gsub("$","",x)
x
#Why?? Need to use regex
x<-gsub("[$]","",x)
x

#sqldf, This is optional
install.packages("sqldf")
library(sqldf)
#Using SELECT statement
oj_s<-sqldf("select brand, income, feat from oj ")
#Subseting using where statement
oj_s<-sqldf("select brand, income, feat from oj where price<3.8 and income<10")
#Order by statement
```

```
oj_s<-sqldf("select store,brand,week,logmove,feat,price, income from oj order
by income asc")
#distinct
sqldf("select  distinct brand from oj")
#Demo sql functions
sqldf("select avg(income) from oj")
sqldf("select min(price) from oj")

##dplyr corner cases
#Selecting odd column names

library(arules)
data("AdultUCI")

names(AdultUCI)

AdultUCI%>%select(capital-gain)%>%dim()#Why this error?


AdultUCI%>%select(`capital-gain`)%>%dim()#Notice the column name
specification








##Window functions in dplyr()
#group_by and summarise would usually produce a single aggregation per
group, group mean, sum, count etc

#Window family: ranking functions, finding top 10, top 5% in
each group
```

```
#Top two income  numbers per group of gender
dat1<-read.csv("F:\\Work\\Jigsaw Academy\\Corporate Trainings\\Dat
Manipulation\\audit.csv")
dat1%>%select(Age,Gender,Income)%>%group_by(Gender)
%>%filter(min_rank(desc(Income))<=3)
%>%arrange(desc(Income))#notice how arrange() works here

#Top 1% by income in each group
dat1%>%select(Gender,Income)%>%group_by(Gender)
%>%filter(cume_dist(desc(Income))<=0.01)%>%arrange(desc(Income))

#Dividing Income into 10 equal parts
dat1%>%mutate(Group=ntile(Income,10))->dat2
head(dat2)

dat2%>%group_by(Group)%>%summarise(Maximum=max(Income),Minimum=
min(Income))

#If we have to create groups in descending order??

dat1%>%mutate(Group=ntile(desc(Income),10))%>%group_by(Group)%>%sum
marise(Maximum=max(Income),Minimum=min(Income),Count=n())
```