# R base

# R basics

- The command line is shown as “>”
- A “+” substitutes “>” if a command in incomplete
- Assignments are defined using “<-” or ”=”
- # is used for commenting
- Missing values are shown as NA
- Impossible numbers as NAN
- Infinity is displayed as Inf

# R data structures

- vectors
- arrays
- matrices
- dataframes
- lists

# R types

- logical (TRUE, FALSE)
- numeric
- integer
- character
- factor (categorical variable)
- date

# R objects

- data structures
- commands
- functions
- models
- data sets
- library objects

# R working space

- Data structures and functions are stored in the user's working space as objects

dataframe table where

columns - variables

rows - observations

# R data structures

dataframe — table where

- columns - variables
- rows - observations

list — collection of R objects

# R variables in a dataframe

| | Manufacturer | Model | Type | Min.Price | Price | Max.Price | MPG.city | MPG.highway | AirBags | DriveTrain |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Acura | Integra | Small | 12.9 | 15.9 | 18.8 | 25 | 31 | None | Front |
| 2 | Acura | Legend | Midsize | 29.2 | 33.9 | 38.7 | 18 | 25 | Driver & Passenger | Front |
| 3 | Audi | 90 | Compact | 25.9 | 29.1 | 32.3 | 20 | 26 | Driver only | Front |
| 4 | Audi | 100 | Midsize | 30.8 | 37.7 | 44.6 | 19 | 26 | Driver & Passenger | Front |
| 5 | BMW | 535i | Midsize | 23.7 | 30 | 36.2 | 22 | 30 | Driver only | Rear |
| 6 | Buick | Century | Midsize | 14.2 | 15.7 | 17.3 | 22 | 31 | Driver only | Front |
| 7 | Buick | LeSabre | Large | 19.9 | 20.8 | 21.7 | 19 | 28 | Driver only | Front |
| 8 | Buick | Roadmaster | Large | 22.6 | 23.7 | 24.9 | 16 | 25 | Driver only | Rear |
| 9 | Buick | Riviera | Midsize | 26.3 | 26.3 | 26.3 | 19 | 27 | Driver only | Front |
| 10 | Cadillac | DeVille | Large | 33 | 34.7 | 36.3 | 16 | 25 | Driver only | Front |
| 11 | Cadillac | Seville | Midsize | 37.5 | 40.1 | 42.7 | 16 | 25 | Driver & Passenger | Front |
| 12 | Chevrolet | Cavalier | Compact | 8.5 | 13.4 | 18.3 | 25 | 36 | None | Front |
| 13 | Chevrolet | Corsica | Compact | 11.4 | 11.4 | 11.4 | 25 | 34 | Driver only | Front |
| 14 | Chevrolet | Camaro | Sporty | 13.4 | 15.1 | 16.8 | 19 | 28 | Driver & Passenger | Rear |
| 15 | Chevrolet | Lumina | Midsize | 13.4 | 15.9 | 18.4 | 21 | 29 | None | Front |
| 16 | Chevrolet | Lumina_APV | Van | 14.7 | 16.3 | 18 | 18 | 23 | None | Front |
| 17 | Chevrolet | Astro | Van | 14.7 | 16.6 | 18.6 | 15 | 20 | None | 4WD |
| 18 | Chevrolet | Caprice | Large | 18 | 18.8 | 19.6 | 17 | 26 | Driver only | Rear |
| 19 | Chevrolet | Corvette | Sporty | 34.6 | 38 | 41.5 | 17 | 25 | Driver only | Rear |
| 20 | Chrylser | Concorde | Large | 18.4 | 18.4 | 18.4 | 20 | 28 | Driver & Passenger | Front |
| 21 | Chrysler | LeBaron | Compact | 14.5 | 15.8 | 17.1 | 23 | 28 | Driver & Passenger | Front |

categorical vars

numerical vars

categorical vars

# The combine function c( )

- to create a vector

  c(1,3,4.5)  gives  [1]  1  3  4.5

- all elements in a vector must be of same type

  c(1,3,"a")  gives  [1]  "1" "3"  "a"

- store vector in memory

  x = c(1,3,4.5)

- Vectors can be used for subsetting dataframes

# Selecting dataframe columns with $

## One column

- select column grades d1$grades

## Two columns

- select columns sex, grades

c(d1$sex,d1$grades)

# Subsetting (querying) dataframes

- columns

```
d1[,c(2,4,9)]
```

- rows

```
d1[c(1:9),]
```

- both

```
d1[c(1:9),c(2,4,9)]
```

# Summarizing by categories

- Counting rows (by factor levels)

  table( )

  prop.table( )

  ftable( )

# Summarizing by categories

## One categorical variable

- table(d1$sex) count males and females
- prop.table(table(d1$sex)) proportion of m,f

## Two categorical vars

- ftable(d1$sex,d1$status) count rows in four levels

  male, female, international students

  male, female, non-international students

# Applying a function - numeric columns

- function apply( ) all cols in dataframe d1

  apply(d1,2,mean)

  apply(d1,2,function(x) exp(mean(log(x))) )
- substitute 2 with 1 to apply the function by rows
- won't work apply(d1,2,mean – 3.5 )
- will do apply(d1,2,function(x) mean(x) – 3.5 )

# Applying a function - numeric and categorical columns

- function tapply( )

  tapply(numeric column, factor column, function)

- find average GPA per sex

  tapply(d1$GPA, d1$sex, mean)

- many columns

# Applying a function - numeric and categorical columns

- many columns

```
tapply(list of numeric columns,
        list of factor columns, function)
```

- find average SAT and GPA, per sex and status

```
listnumcols = list(d1$SAT,d1$GPA)
listcatcols  = list(d1$sex,d1$status)
tapply(listnumcols,listcatcols, mean)
```