Subseting elements:
selected_elements ← my-matrix[my-matrix ≥10]

- ccessing elements
  matrix[, 3] 3rd column | matrix[2,3] second row, 3rd colum
- transpose
  t(my.matrix)
- multiplication, addition
  my-matrix %*% another matix , matrix1 + matrix 2
- Combine Vertically
  combined_matrix ← rbind (matrix1, matrix2)
- Selecting element based on sequence
  matrix [, m[1,]>2] # select columns for which 1st row is >2.
- diag(A) , diag(n).
- Reshape matrix : dim(matrix) ← c(2,3)

Arrays

(Multi dimensional data structures)

- create array : my_array ← array (1:18, dim=c(3,3,2))
- accessing elements: element ← my-array [2,3,1]
  row col depth.
- subarray: sub_array ← my-array[1:2, 1:2,1]
- transpose: transposed_array ← aperm (my-array, c(3,2,1))
- dimension set: dim (array) ← c(2,3,3).
- * - names to array: dimnames(my-array) ← list(c("Row1","Row2","Row3"),
  c("Col1", "Col2", "Col3")
  c("Depth1", "Depth2"))
- reshaping: reshaped-array ← aperm (array, c(3,2,1))

Data frames

- data1 ← data[data$age > 25,] : Select column with condition.
- columns ← data[, c('age', 'name')] : Select multiple columns
- sorted_data ← data[order(data $ age, decreasing = TRUE),]
  τα ξινόμενα ....., decreasing = FALSE,)]
- Subset - subset_data ← subset (data, age >25)
- * - Προσθηκη στηλης: new_data ← transform (data, new_col = age * 2)

- Δύο περιορισμοί, σαν διάστημα:

```
selected_rows <- data[(data$ Income <100) &
                      (data$ Income >1000),]
```

Basics

- Rm (list = ls()) = καθαρισμός της R
- save image (file = "df. RData")

- Recycle R: εάν προσθέσω έναν vector με 10 στοιχεία σε ένα διάνυσμα με 5, τότε θα προστεθούν μόνο τα 5 πρώτα.
- data[-c(X,Y)] παίρνω τα μη NAN. στοιχεία
- στα αρνητικά seq βάζω -0,5.

Apply family

apply: operate functions in specific parts of an array.
Suppose we want to find median for every column
- apply (state.x77 , 2 , median)
- apply (data, 2, median , c(0,25, 0,75))

Άσκηση
- column_sums <- apply (data, 2, sum)
- new_data <- rbind (data, column_sums)

lapply: applies a function to every element in a list and returns a list.
- my_list <- list (a = 1:3, b = 4:6 , c = 7:9)
result <- lapply (my_list , mean)
result <- lapply (ataframe, sum) # Same to dataframe

sapply: returns a vector or matrix. 'S' → simplify
- data_means <- sapply(data , means).

Αλλάζει μόνο η εμφάνιση

★ apply (state.x77,2, quantile , c(0,25, 0.75))

Σε κάθε αλλαγή ή υπολογισμό μετά ή πριν την παρένθεση βάζουμε ";"
για να προσδιορίσουμε αλλαγή σε γραμμές ή στήλες.

* Διαγραφή low correlated variables

```
cors ← sapply (pred, cor, y = response) # calculate correlations.
mask ← (rank (- abs(cors)) <= 10)
```

* - Εύρεση μεγάλης συσχέτισης: cor-values ← sapply (data, function(col),
selected_cols ← names(cor-values[cor...]) cor(col, data$Y) >0.75)

tapply: Ομαδοποιώ και υπολογίζω συναρτήσεις σε data.frames βάσει ενός factor. Π.χ. tapply(X, f, fun) f=factor στήλη, X=vector

- chickwts : weight = numeric, feed = factor.
  tapply (chickwts$weight, chickwts$feed, mean, sd, lenght etc.)

Αφαίρεση στηλών

1ος τρόπος:

```
remove_col ← c("X1", "X2" ....)
new_df ← df[, !(names(df) %in% remove_col)]
```

2ος τρόπος

```
clear_data ← subset (data, select = - c("col1", "col2"))
```

Υπολογισμός statistics με apply

```
descriptives_df ← data.frame (
min = sapply (df, function(x), min(x, na.rm = TRUE)),
quantile = sapply (df, function(x), quantile(x, 0,25, na.rm = TRUE)),
mean = sapply (df ....., median (x, na.rm = TRUE)) ) ...
```

Sequences και κατανομές

- rnorm (n, $\overset{mean}{mu}$, $\overset{sd}{sigma}$) (generate)	rbinom($\overset{numbers}{n}$, $\overset{number\ of\ trials}{n}$, p) → p, prob
- pnorm (X, mu, sigma) (πίνακας Φ(x))	pbinom (X, n, p)
- dnorm (x, mu, sigma) (shape)	dbinom (x, n, p)
- qnorm (0.95, mu, sigma) (given 95%)	qbinom (0.95, n, p)

σαν να βρίσκουμε critical value

```
seq(from=value, to=value, by=value, length.out=value)
- seq(0, 20, 4)
- seq(20,1,-1)
- replicated_seq ← rep(1:3, times=3)

- paste("no", 1:5) -> πέντε no με αριθμό δίπλα
paste("no", 1:5, sep="") χωρίς κενό στα strings.
```

Διαστήματα Εμπιστοσύνης.

$\bar{X} \pm Z_{1-a/2}\, \sigma/\sqrt{n}$, $\bar{X} - \bar{Y} \pm Z_{1-a/2} \sqrt{\sigma_X^2/n + \sigma_Y^2/m}$.

Παράδειγμα1: Έστω ένα data sample.

```
conf_interval ← t.test(sample_data)$conf.int.   default 95%
conf_int90 ← t.test(sample_data, conf.level=0,90)$conf.int.
print(conf_int)
```

Παράδειγμα2: Βρίσκω και θέτω εγώ τα μέτρα.

```
n=175
mean ← mean(data) , sd_value ← (sqrt(sum(X-mean)^2)/(n-1))
t_value ← qt(Z1-a/2, df=n-1)
ci_lower ← mean - t_value * (sd_value/sqrt(n))
ci_upper ← mean + t_value * (sd_value/sqrt(n))
print(paste(....., ci_lower, ci_upper))
```

Άσκηση

Έστω το UKDriversdeaths dataset. Να βρω το 95% CI για τον μέσο των θανάτων ανά μήνα για κάθε χρονιά

Hint: χρήση της apply ονομάζω data το set.

```
Xbars ← apply(data, 2, mean) , sds ← apply(data, 2, sd)
n ← nrow(data) , Z ← 1.96
standar_error ← sds/ sqrt(n)
```

Για να υπολογίσω μία πιθανότητα
βάζω απλά qnorm(0.975) = 1,96

```
lower_CI ← xbars − Z · standard_error.
upper_CI ← xbars + Z · standard_error.
print(paste( ...., lower_CI, upper_CI))
```

αλλιώς Z ← qnorm(0.975) αντί να δώσουμε εμείς τον αριθμό.

Άσκηση

Έστω ένα dataset με 1 numeric και factor. Να βρεθεί το διάστημα 95% για τους μέσους των 2 ομάδων.

1ος:
```
xbars ← tapply(data$scores, data$teams, mean)
sds ← tapply(...)   n ← tapply(...)
Z ← qnorm(0.975)
standard_error ← sds / sqrt(n)
lower_CI ← xbars − (z * st...)   Upper_CI ← xbars + (z * st...)
print(paste(lower_CI, upper_CI))
```

2ος:
```
                    numeric  factor
result ← tapply( ....... , ....... , function(x) t.test(x)$conf.int)
print(result)
```

- Διάβασμα αρχείων: dec=',' εάν έχει κόμμα μέσα το κάνει τελεία

Έλεγχοι Υποθέσεων test

t.test:
```
t.test(sample_data, mu = expected_mean)   #1 sample
t.test(group1, group2)                    #2 sample
```

▶ t.test(x, y=null, alternative = c("two sided", "less", "greater"), mu = 0,
paired = FALSE, var.equal = FALSE, conf.level = 0.95)

- Από default δεν θεωρεί το t test ότι έχουμε ίσες διακυμάνσεις, αλλά:

var.test(group1, group2) ελέγχουμε με F-test.
t.test(group1, group2, var.equal = TRUE)

paired = FALSE τα δείγματα είναι ανεξάρτητα
paired = TRUE είναι συσχετισμένα ή paired

- Extract pvalue: p-value ← result$p.value

less"

```
Ho : μ= 75 , H1 : μ< 75.
 t_test_result ← t.test (sample, mu=75, alternative="less")
 Θέτουμε a = 0.05
 if ( a > p-value) {
   cat ("reject null hypothesis)
 } else {
   cat ("fail to reject null hypothesis)}.
```

- wilcox. test (group 1, group2)

Regression

Stepwise AIC

lm (Y~. , data = data)

- stepAIC (model, direction = "forward")   και χωρίς το AIC
- stepAIC (model, direction = "backward")
- stepAIC (model, direction = "both")

Influence.measures(model)
confint(fit)

Stepwise BIC

- step (model, K= log (length(x)), direction ="backward")
- step (model, K= log (length (x)), direction = "forward")
- . . . . . . direction ="both")

Predictions, Plots and style

Predictions,

confint (fit) = φτιάχνει διάστημα για τα coefficients.

- muhat ← predict (fit, interval = "confidence")

μέση τιμή των predictions.

- yhat ← predict (fit, inerval = "prediction")

y καπέλο με διάστημα

★ Προσομοίωση νέων δεδομένων υπολογίζοντας xbar=mu, sd =, και βάζοντας rnorm (1, xbar, sd) → βλέπε video

συνδυασμός με apply για να το κάνω για πολλές στήλες.

Φτιάχνω ένα νέο data frame. ή σειρά-row και το βάζω στα predict   apply (data, 2 mean, sd

explanatory-data ← data.frame(

ορίζω τους αριθμούς.   X1 = rnorm(1) xbars[1], sds[1], ...)

predict (model, explanatory-data) → το νέο $\hat{y}$.

Plots.

- Setting the layout.

par(..... c(2,2), bg = "...")   pch=16 για fill

Plot every explanatory vs fitted.   fitted ← fitted(model)

- plot (data$X1, model$residuals, main = "fitted vs predicted X1")
- abline (h=0, col= "red")

fitted vs residuals : plot (model$residuals, fitted-values)

Διάστημα 95% για τα regression coefficients.

CI ← confint (model, level = 0.95)

★ Διαγραφή έκτροπων τιμών   influence ← dataframe, order → διαγράφω τα μεγαλύτερα coocks.distance

Extract R^2:   Summary(model)$r.squared.

★ New predictions με CI 95%:

φτιάχνω τα νέα data: new-predictions ← predict (model, explanatory, interval = "confidence", level = 0.95)

★ New predictions με CI 95% για το responce:

observed_predictions ← predict (model, explanatory, interval = "prediction", level = 0,95)

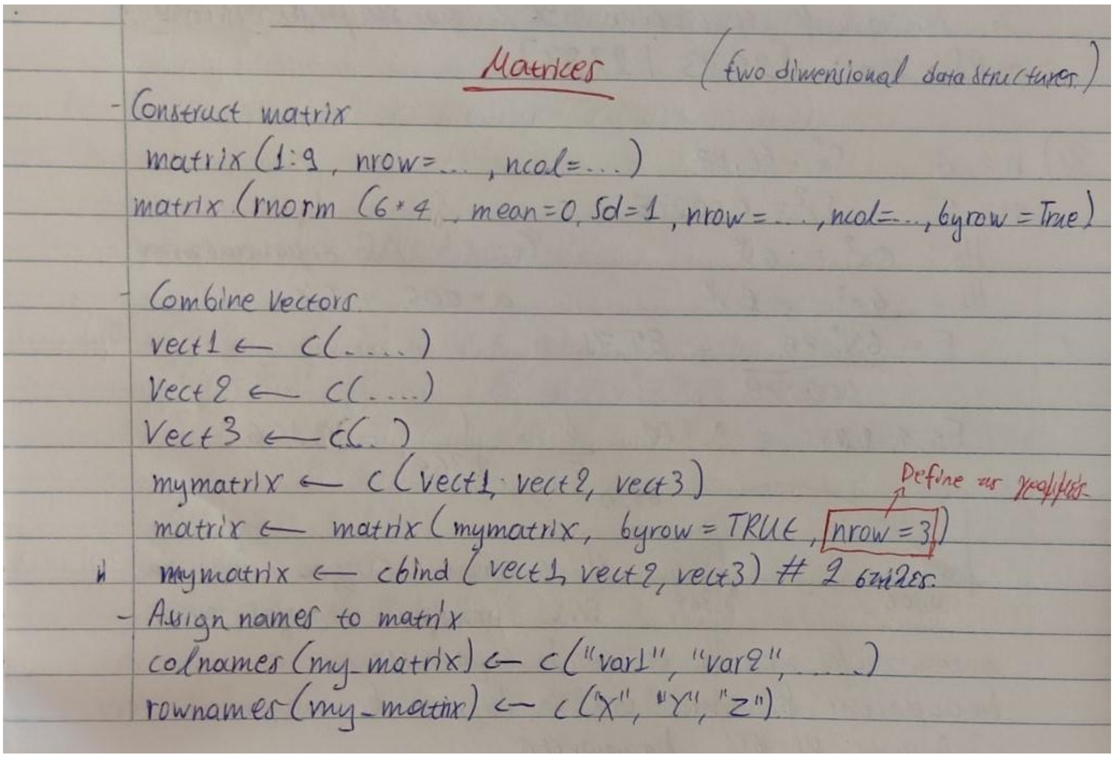
★ cook ← influence $ cooks.distance

threshold ← 4 / nrow (data) (το 4? το αλλάζω)

influencial_obs ← which (cook > threshold)

clean_data ← data [cook <= threshold,]

new-model ← lm (Y~ ., data = clean_data)



Matrices (two dimensional data structures)

- Construct matrix

matrix (1:9 , nrow= ... , ncol= ...)

matrix (rnorm (6*4 , mean=0, sd=1 , nrow= ... , ncol= ... , byrow = True)

- Combine vectors

vect1 ← c(....)

Vect2 ← c(....)

Vect3 ← c(.)

mymatrix ← c(vect1, vect2, vect3)

matrix ← matrix(mymatrix, byrow = TRUE, nrow = 3) Define ως γραμμές

ή mymatrix ← cbind (vect1, vect2, vect3) # 2 στήλες

- Assign names to matrix

colnames (my_matrix) ← c("var1", "var2", ...)

rownames (my_matrix) ← c("X", "Y", "Z")