

Codice allegato

- Corso di Fondamenti delle Misurazione -

- Ingegneria Informatica e dell'Automazione -

A cura di:
Laura Loperfido
Andrea Lops
Teresa Pantone
Paolo Rotolo

Prof. Ing. Filippo Attivissimo

```
function main
clear
clearvars
clc

disp('*****')
disp('***_Welcome_***')
disp('*****')
disp('_')
disp('_')
disp('_')

choice = 0;
disp('What_kind_of_data_do_you_want_to_analyze?')

while(choice~=7)
    choice = 0;
    while(choice < 1 || choice > 6)
        disp('1_Resistance_of_superalloys')
        disp('2_Systolic_blood_pressure_(even_number_of_samples)')
        disp('3_Systolic_blood_pressure_(odd_number_of_samples)')
        disp('4_Splitting_Tensile_Strength_of_Concrete_Cylinders')
        disp('5_Random_noise_with_Gaussian_distribution_(5000_samples)')
        disp('6_Random_noise_with_Uniform_distribution_(5000_samples)')
        choice = input('Insert_choice:');
    end

    if(choice==1)
        v = parseFile('Dati_1.txt');
        eseguiEsercizio(v);
        break
    elseif(choice==2)
        v = parseFile('Dati_2.txt');
        eseguiEsercizio(v);
        break
    elseif(choice==3)
```

```

        v = parseFile('Dati_3.txt');
        eseguiEsercizio(v);
        break
    elseif(choice==4)
        v = parseFile('Dati_4.txt');
        eseguiEsercizio(v);
        break
    elseif(choice==5)
        v = distribuzioneGaussiana();
        eseguiEsercizio(v);
        break
    elseif(choice==6)
        v = distribuzioneUniforme();
        eseguiEsercizio(v);
        break
    end
end

disp(' ')
disp(' ')

function eseguiEsercizio(v)
    %% a) rappresenta l istogramma
    istogramma(v);

    %% b) calcola mediana e dispersione
    disp(strcat(['The median is ' num2str(mediana(v))]));
    disp(strcat(['The dispersion is ' num2str(dispersione(v))]));

    %% c) calcola il primo, secondo e terzo quartile
    disp(strcat(['The first quartile is ' num2str(firstQuartile(v))]));
    disp(strcat(['The second quartile is ' num2str(secondQuartile(v))]));
    disp(strcat(['The third quartile is ' num2str(thirdQuartile(v))]));

    %% d) misura lo scarto (o interquartile) e gli outliers
    disp(strcat(['The interquartile is ' num2str(interquartile(v))]));
    OUTLIERS = outliers(v);
    if (isempty(OUTLIERS))
        disp('There are no outliers');
    else
        disp(strcat(['The outliers are ' num2str(OUTLIERS)]));
    end

    %% e) rappresenta i dati con il boxplot
    dispBoxplot(v);

    %% f) calcola media e varianza
    disp(strcat(['The average is ' num2str(media(v))]));
    disp(strcat(['The variance is ' num2str(varianza(v))]));

    %% h) disegna il diagramma quantile-quantile (qqplot)
    dispQQPlot(v)

    %% i) verifica il tipo di distribuzione con il Chi Quadro test
    CHISQUARETEST = chiSquareTest(v);
    if (CHISQUARETEST < 0.5)
        disp('The curve could be a Gaussian');
    else
        disp('The curve is not a Gaussian');
    end
end

```

```

end

function v = parseFile(filename)
    file1 = importdata(filename, '\t');
    v = sort(file1);
end

function output = istogramma(v)
    % calcola il passo qui
    output = histogram(v);
end

function output = mediana(v)
    output = median(v);
end

function output = dispersione(v)
    output = range(v);
end

function output = firstQuartile(v)
    output = median(v(v<median(v)));
end

function output = secondQuartile(v)
    output = median(v);
end

function output = thirdQuartile(v)
    output = median(v(find(v>median(v))));
end

function output = interquartile(v)
    output = (thirdQuartile(v) - firstQuartile(v));
end

function nOutliers = outliers(v)
    nOutliers = thirdQuartile(v) + (1.5*interquartile(v));
end

function dispBoxplot(v)
    figure;
    boxplot(v);
end

function dispQQPlot(v)
    figure;
    qqplot(v);
end

function output = chiSquareTest(v)
    % 0 pu essere gaussiana, 1 non lo
    output = chi2gof(v);
end

function output = varianza(v)
    output = var(v);
end

function output = media(v)

```

```
        output = mean(v);  
end  
  
function output = distribuzioneUniforme()  
    output = rand(1, 5000);  
end  
  
function output = distribuzioneGaussiana()  
    output = randn(1, 5000);  
end  
  
end
```