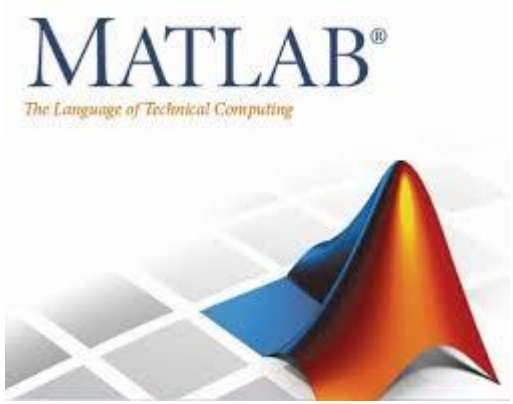
Električna mjerenja
(pomoćni materijal za predavanja)

Univerzitet Crne Gore

Elektrotehnički fakultet

Statistička obrada mjerenja korišćenjem MATLAB-a



Statistička obrada merenja korišćenjem Matlaba

MathWorks® Products Solutions Academia Support Community Events Get MATLAB

Documentation All Examples Functions Apps Search R2019a Documentation Documentation

CONTENTS Close

« Documentation Home

« Statistics and Machine Learning Toolbox

« Probability Distributions

Working with Probability Distributions

ON THIS PAGE

Probability Distribution Objects

Probability Distribution Functions

Probability Distribution Apps and User Interfaces

See Also

Related Topics

Trial Software Product Updates

Working with Probability Distributions

R2019a

Probability distributions are theoretical distributions based on assumptions about a source population. The distributions assign probability to the event that a random variable has a specific, discrete value, or falls within a specified range of continuous values.

Statistics and Machine Learning Toolbox™ offers several ways to work with probability distributions.

- Use Probability Distribution Objects to fit a probability distribution object to sample data, or to create a probability distribution object with specified parameter values.
- Use Probability Distribution Functions to work with data input from matrices.
- Use Probability Distribution Apps and User Interfaces to interactively fit, explore, and generate random numbers from probability distributions. Available apps and user interfaces include:
  - The **Distribution Fitter** app
  - The **Probability Distribution Function** user interface
  - The Random Number Generation user interface (`randtool`)

For a list of distributions supported by Statistics and Machine Learning Toolbox, see Supported Distributions.

You can define a probability object for a custom distribution and then use the Distribution Fitter app or probability object functions, such as `pdf`, `cdf`, `icdf`, and `random`, to evaluate the distribution, generate random numbers, and so on. For details, see Define Custom Distributions Using the Distribution Fitter App. You can also define a custom distribution using a function handle and use the `mle` function to find maximum likelihood estimates. For an example, see Fit Custom Distribution to Censored Data.

Generišimo u Matlabu skup podataka o temperaturi pravljenjem niza koji u osnovi ima sinusoidu, a zatim je „randomiziran“ korišćenjem Gaussovih slučajnih brojeva.

Command Window

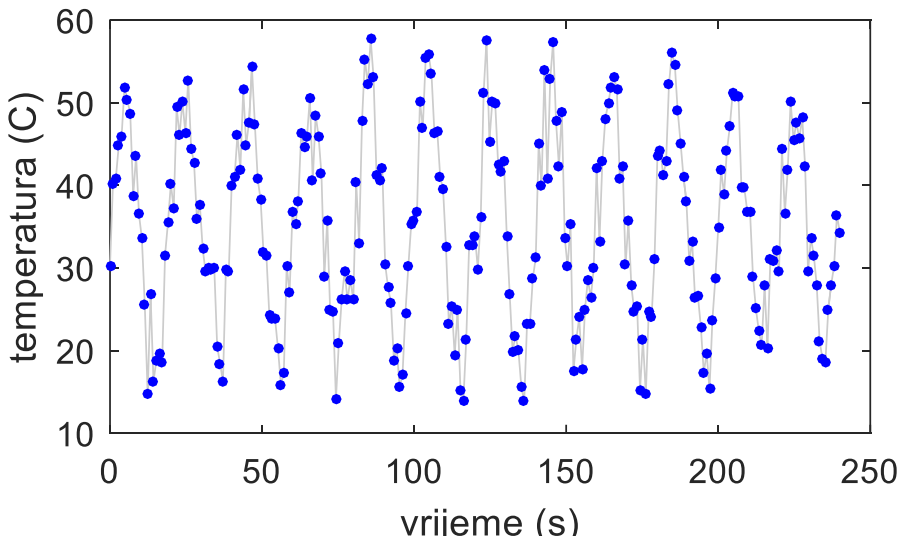
New to MATLAB? See resources for Getting Started.

```
>> f = 50e-3; % frekvencija sinusnog signala
a = 15; % amlituda oscilovanja
jo = 35; % jednosmjerni opseg oscilovanja
fo = 1; % frekvencija odabiranja
nT = 12; % broj perioda sinusnog signala
sigma = 4; % standardna devijacija
np = fo/f+1; % broj odbiraka po periodi
n = nT*np+1; % ukupan broj odbiraka
t = linspace(0,nT/f,n); % vremenski niz
sig = jo + a*sin(2*pi*f*t); % sinusni signal
rng(43); % generator slučajnih brojeva
shum = sigma*randn(size(t)); % Gausov šum
signal = sig + shum; % sinusni sinal+šum
```

Command Window

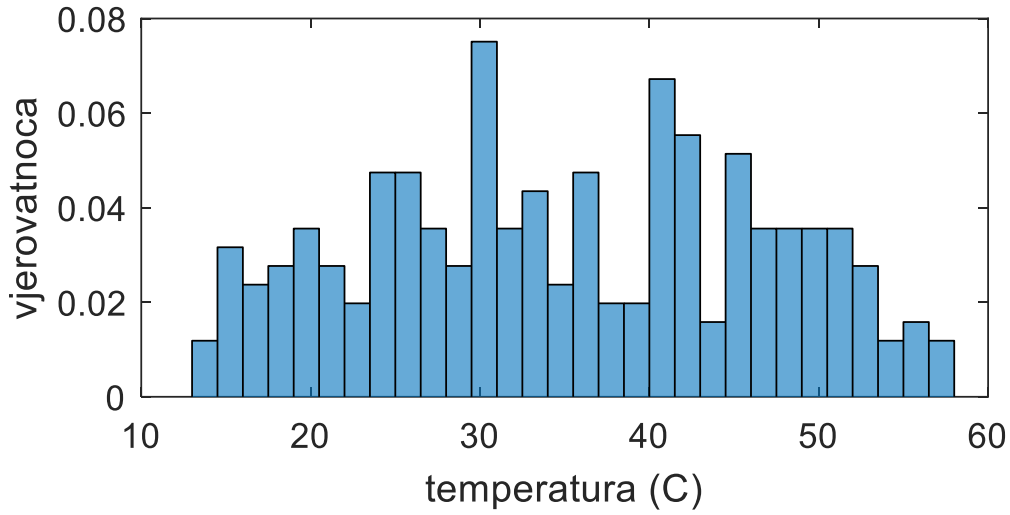
New to MATLAB? See resources for Getting Started.

```
>> h = figure;
p = plot(t,signal,'o-',...
'Color',[.8,.8,.8],...
'MarkerFaceColor','b',...
'MarkerEdgeColor','none',...
'MarkerSize',3);
xlabel('vrijeme (s)');
ylabel('temperatura (C)');
```



- Nacrtajmo histogram za dobijena mjerenja temperature:

```
Command Window
>> h = figure;
histogram(signal,...
30, ... % broj intervala mjerenja
'normalization','probability'...
);
xlabel('temperatura (C)')
ylabel('vjerovatnoca')
```



Nacrtali smo frekvencijsku distribuciju vrijednosti, da bi stekli utisak o karakteristikama slučajnih varijacija, ali ovaj podatak uključuje također i sinusoidu koja nam otežava posao.

- Da bismo izračunali srednju vrijednost uzorka i standardnu devijaciju za svaki uzorak unutar periode signala, moramo izabrati nT tačaka koje odgovaraju jedna drugoj.
- Trenutno, svi odbirci signala se nalaze u jednom dugom nizu signal duzine n.
- Potrebno je preoblikovati podatke u vidu matrice nT×np nizu, u kojoj svaki red odgovara novom periodu.
- To će omogućiti da su odgovarajuće tačke poređane po koloni.

```
>> signal_reorg = reshape(signal(1:end-1)',[np,nT])'; % preuređivanje niza po kolonama
size(signal_reorg) % provjera dimenzija signala
signal_reorg(1:3,1:4) % stampanje prva tri reda i prve cetiri kolone matrice

ans =

    12    21


ans =

   30.2718   40.0946   40.8341   44.7662
   40.1836   37.2245   49.4076   46.1137
   40.0571   40.9718   46.1627   41.9145
```

Napravimo kod za računanje srednje vrijednosti i standardne devijacije

Command Window

```
>> mi_f = @(niz) sum(niz)/length(niz);     % funkcija za srednju vrijednost
var_f = @(niz) sum((niz-mi_f(niz)).^2)/(length(niz)-1); %funkcija za varijansu
sigma_f = @(niz) sqrt(var_f(niz));  %funkcija za standardnu devijaciju
niz_mi = NaN*ones(1,np); % inicijalizacija niza srednjih vrijednosti
niz_var = NaN*ones(1,np); % inicijalizacija niza varijansi
niz_sigma = NaN*ones(1,np); % inicijalizacija niza standardnih devijacija
for i = 1:np % za svaku kolonu
niz_mi(i) = mi_f(signal_reorg(:,i));
niz_var(i) = var_f(signal_reorg(:,i));
niz_sigma(i) = sqrt(niz_var(i));
end
```

Kolone predstavljaju odbirke. Sljedeće što je potrebno uraditi je oduzeti srednju vrijednost od svake kolone.
Koristimo repmat za reprodukciju niz_mi kroz nT redova kako bi se moglo jednostavno oduzeti.

niz_mi =

35.3599   39.1421   41.1250   46.3362   50.0799   50.0117   52.1980   48.5146   43.6910   41.9932   37.2561   32.549

```
>> repmat(niz_mi,[nT,1])
```

```
35.3599   39.1421   41.1250   46.3362   50.0799   50.0117   52.1980   48.5146   43.6910   41.9932   37.2561   32.5490
35.3599   39.1421   41.1250   46.3362   50.0799   50.0117   52.1980   48.5146   43.6910   41.9932   37.2561   32.5490
35.3599   39.1421   41.1250   46.3362   50.0799   50.0117   52.1980   48.5146   43.6910   41.9932   37.2561   32.5490
35.3599   39.1421   41.1250   46.3362   50.0799   50.0117   52.1980   48.5146   43.6910   41.9932   37.2561   32.5490
35.3599   39.1421   41.1250   46.3362   50.0799   50.0117   52.1980   48.5146   43.6910   41.9932   37.2561   32.5490
35.3599   39.1421   41.1250   46.3362   50.0799   50.0117   52.1980   48.5146   43.6910   41.9932   37.2561   32.5490
35.3599   39.1421   41.1250   46.3362   50.0799   50.0117   52.1980   48.5146   43.6910   41.9932   37.2561   32.5490
35.3599   39.1421   41.1250   46.3362   50.0799   50.0117   52.1980   48.5146   43.6910   41.9932   37.2561   32.5490
35.3599   39.1421   41.1250   46.3362   50.0799   50.0117   52.1980   48.5146   43.6910   41.9932   37.2561   32.5490
35.3599   39.1421   41.1250   46.3362   50.0799   50.0117   52.1980   48.5146   43.6910   41.9932   37.2561   32.5490
35.3599   39.1421   41.1250   46.3362   50.0799   50.0117   52.1980   48.5146   43.6910   41.9932   37.2561   32.5490
35.3599   39.1421   41.1250   46.3362   50.0799   50.0117   52.1980   48.5146   43.6910   41.9932   37.2561   32.5490
```

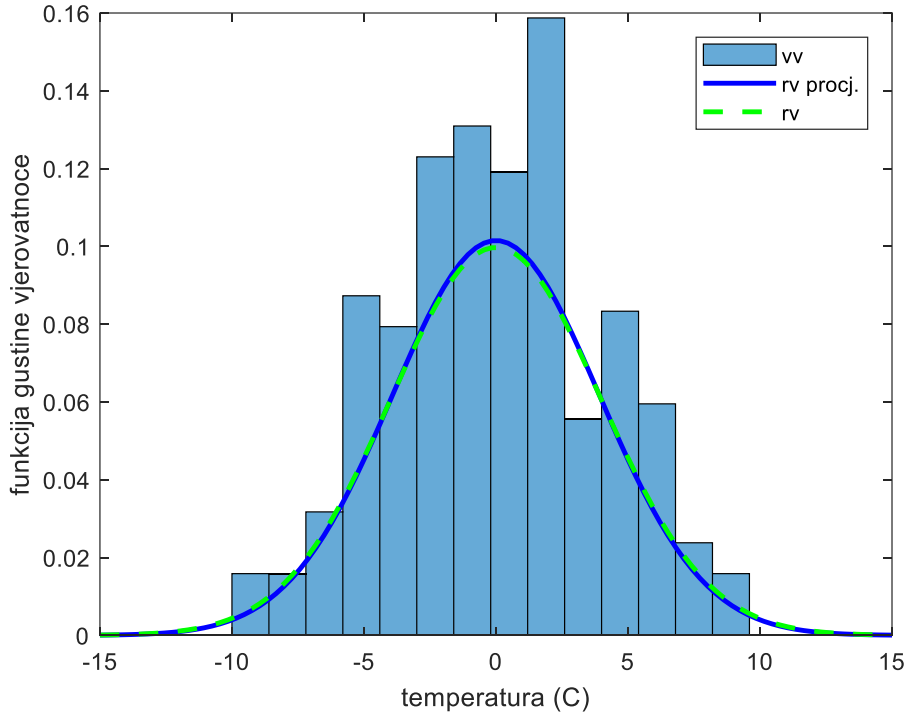
Command Window

New to MATLAB? See resources for Getting Started.

```
>> signal_s = signal_reorg - repmat(niz_mi,[nT,1]);
% smanjenje
signal_s_reorg = reshape(signal_s,[1,nT*np]);
size(signal_s_reorg) % provjera dimenzija
% Podesavanje raspodjela verovatnoce podacima.
rv_model = fitdist(signal_s_reorg','normal');
x= linspace(-15,15,100);
rv = pdf(rv_model,x);
trv = normpdf(x,0,sigma); %teorijska raspodjela vjerovatnoce

% vv - vjerovatnoce svih vrijednosti slucajne promenljive
%  zbir svih vrijednosti te funkcije mora da bude 1

h = figure;
histogram(signal_s_reorg,...
round(sigma*sqrt(nT)), ...
'normalization','probability');
hold on
plot(x,rv,'b-','linewidth',2); hold on
plot(x,trv,'g--','linewidth',2);
legend('vv','rv procj.','rv')
xlabel('temperatura (C)')
ylabel('funkcija gustine vjerovatnoce')
```



- Srednja vrijednost vektora srednjih vrijednosti se dobija kao:

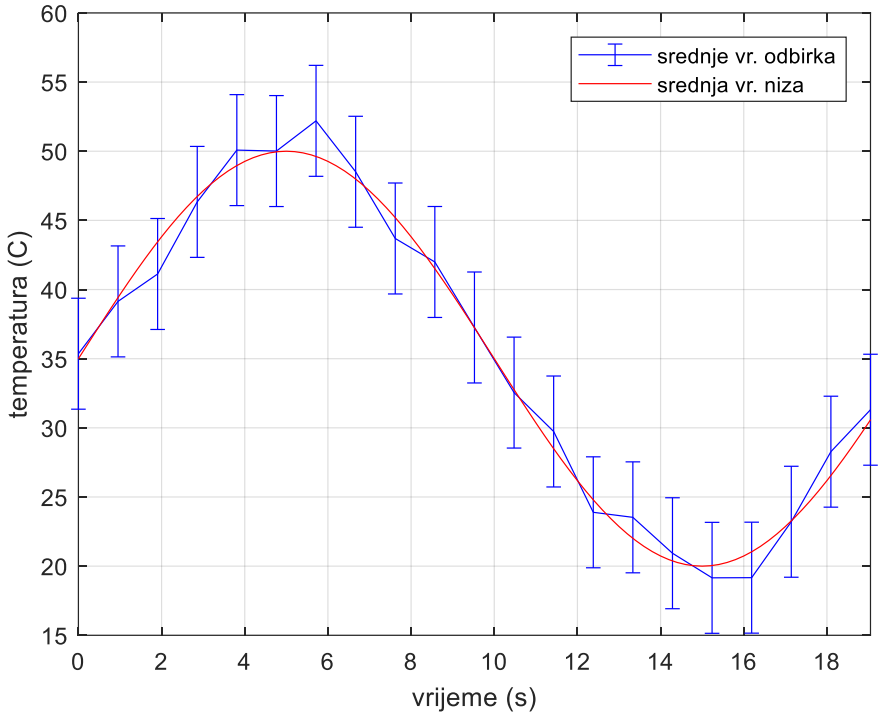
mean(niz_mi) ili mi_f(niz_mi)

- Srednja vrijednost od standardnih devijacija:

ssd = mi_f(niz_sigma)

- Standardna devijacija od standardnih devijacija:

sdsd = sigma_f(niz_sigma)



Command Window

```
>> ssd = mi_f(niz_sigma)
sdsd = sigma_f(niz_sigma)
h = figure;
gr1 = errorbar(t(1:np),niz_mi,ssd*ones(1,np),'b'); hold on
t1 = linspace(0,1/f,101);
gr2 = plot(t1,jo + a*sin(2*pi*f*t1),'r-');
xlim([t(1),t(np)])
grid on
xlabel('vrijeme (s)')
ylabel('temperatura (C)')
legend([gr1 gr2],'srednje vr. odbirka','srednja vr. niza','Location','NorthEast')
hgsave(h,'temp');
```