Stat 302
Statistical Software and Its Applications
SAS: Distributions

Yen-Chi Chen

Department of Statistics, University of Washington

Autumn 2016

Distributions in R and SAS

Distribution	R	SAS
Beta	beta	BETA
binomial	binom	BINOMIAL
Cauchy	cauchy	CAUCHY
chi-square	chisq	CHISQUARE
exponential	exp	EXPONENTIAL
F	f	F
gamma	gamma	GAMMA
geometric	geom	GEOMETRIC
hypergeometric	hyper	HYPERGEOMETRIC
lognormal	lnorm	LOGNORMAL
negative binomial	nbinom	NEGBINOMIAL
normal	norm	NORMAL
Poisson	Pois	POISSON
Student's t	t	T
uniform	unif	UNIFORM
Weibull	weibull	WEIBULL

Generating Normal – 1

- Use function `rand` to generate random values.

```
data newdata;
do i = 1 to 100;
        x = rand('normal');
        output;
end;
run;
title "Random Normal";
proc print data=newdata noobs;
run;
```

- This generates 100 points from a standard normal distribution, saved in the variable `x`.

Generating Normal – 2

```
data newdata2;
do i = 1 to 100;
        x = rand('normal',10,2);
        output;
end;
run;
title "Random Normal";
proc print data=newdata2 noobs;
run;
```

- This generates 100 points from $N(10, 2^2)$, put in the variable `x`.

Generating from Exponential (10)

```
data expdata;
do i = 1 to 100;
        x = rand('exponential', 10);
        output;
end;
run;
title "Random Exponenetial";
proc print data=expdata noobs;
run;
```

- This generates 100 points from $Exp(10)$, put in the variable `x`.

Summary statistics from `proc univariate`

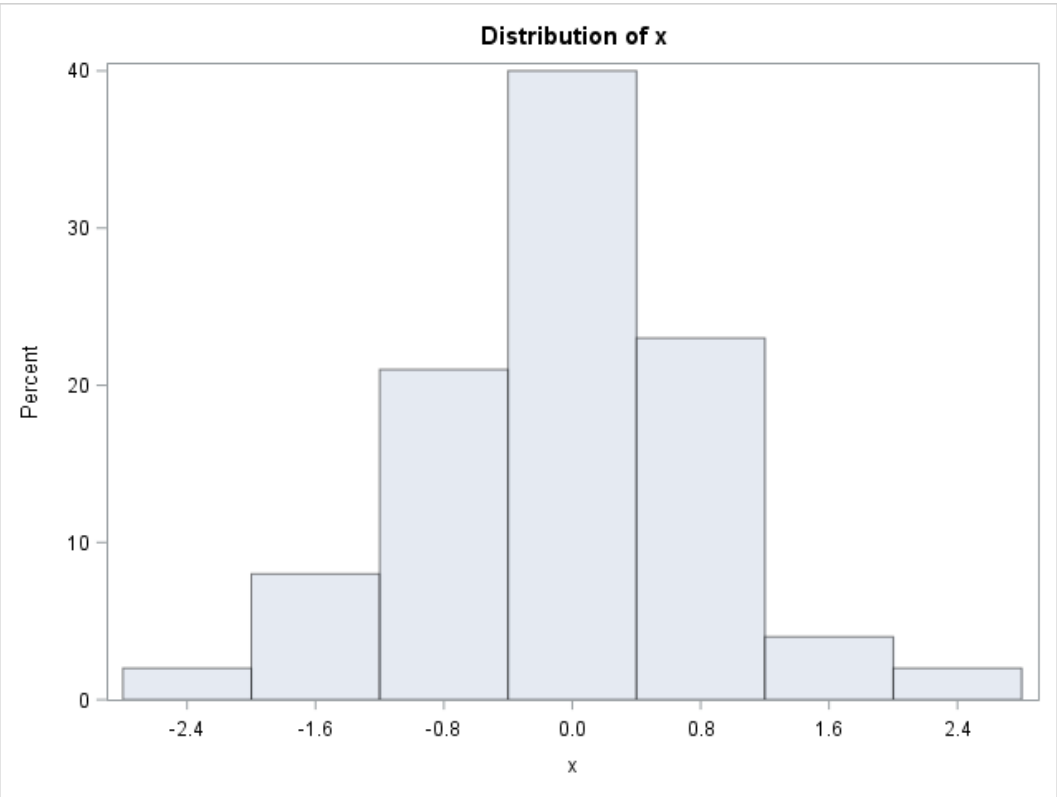
```
proc univariate data=newdata;
   var x;
run;
```

- This outputs a summary of the univariate variable x in the data object `newdata`.

proc univariate + histogram

```
proc univariate data=newdata;
   histogram x;
run;
```

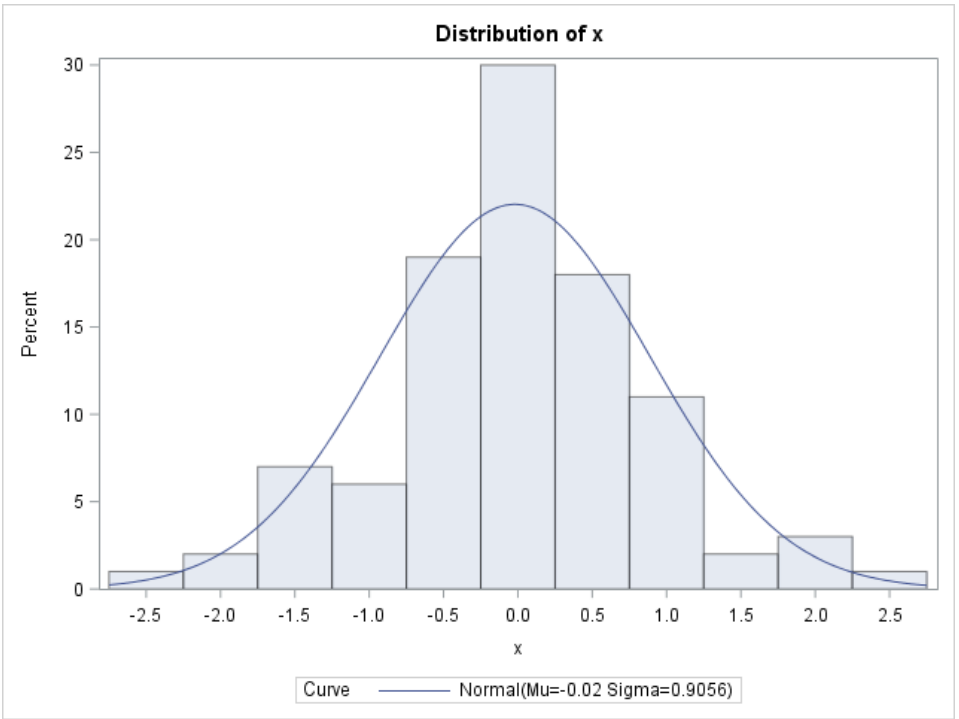
- This adds a histogram to the summary table.



proc univariate + histogram + normal fit – 1

```
proc univariate data=newdata;
   histogram x/normal
                midpoints = -1 -0.5 0 0.5 1;
run;
```

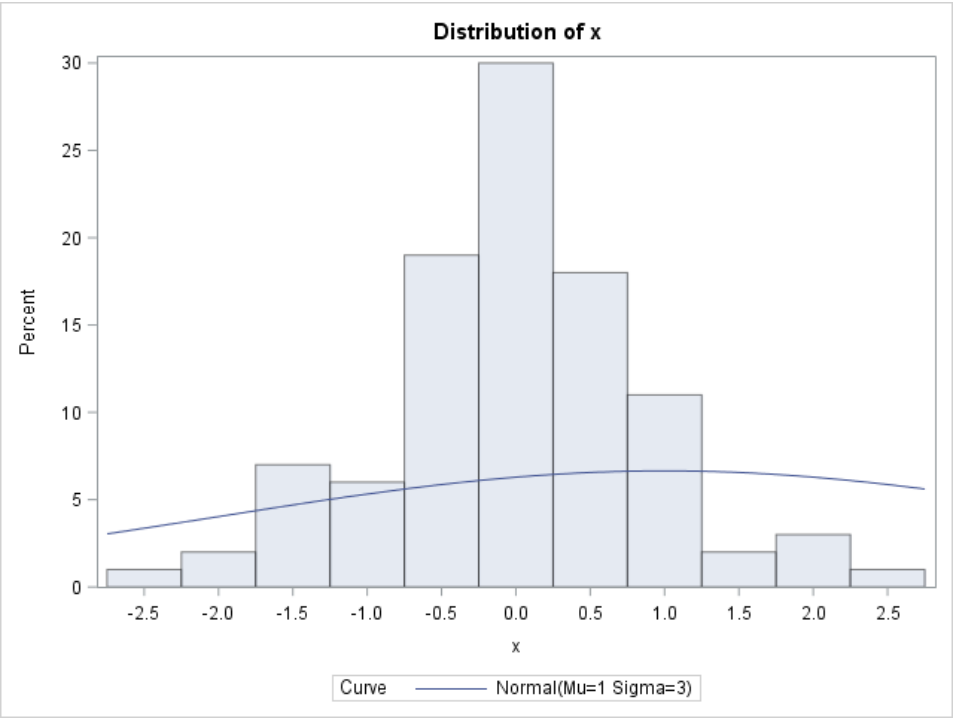
- This adds a normal fit to both summary table and the histogram.
- `midpoints:` this specify the bins of histograms.



proc univariate + histogram + normal fit – 2

```
proc univariate data=newdata;
   histogram x/normal(mu=1, sigma=3)
                midpoints = -1 -0.5 0 0.5 1;
run;
```

- This specify which normal distribution being fitted.



proc univariate: noprint

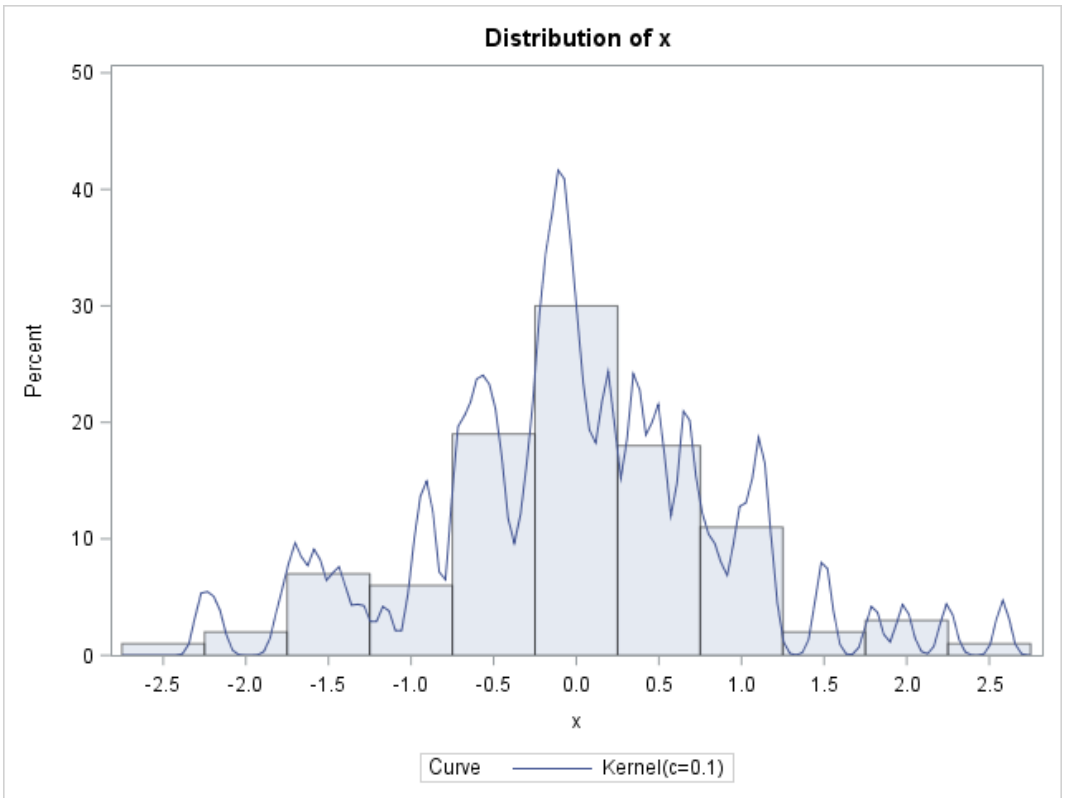
```
proc univariate data=newdata noprint;
   histogram x/normal(noprint)
                midpoints = -1 -0.5 0 0.5 1;
run;
```

- `noprint`: this will not print out the analysis table.

proc univariate+ histogram+KDE – 1

```
proc univariate data=newdata noprint;
   histogram x /
kernel(c=0.1)
midpoints = -1 -0.5 0 0.5 1;
run;
```

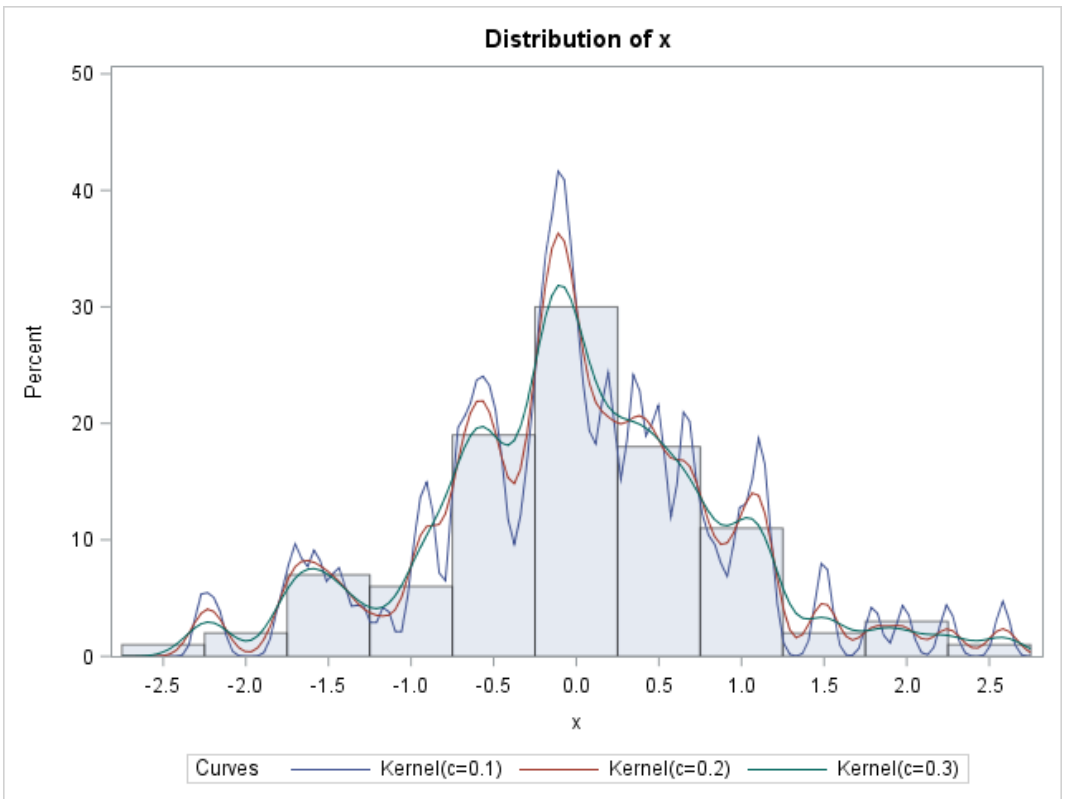
- c: this specifies the smoothing bandwidth.



proc univariate+ histogram+KDE – 2

```
proc univariate data=newdata noprint;
   histogram x /
kernel(c=0.1 0.2 0.3)
midpoints = -1 -0.5 0 0.5 1;
run;
```

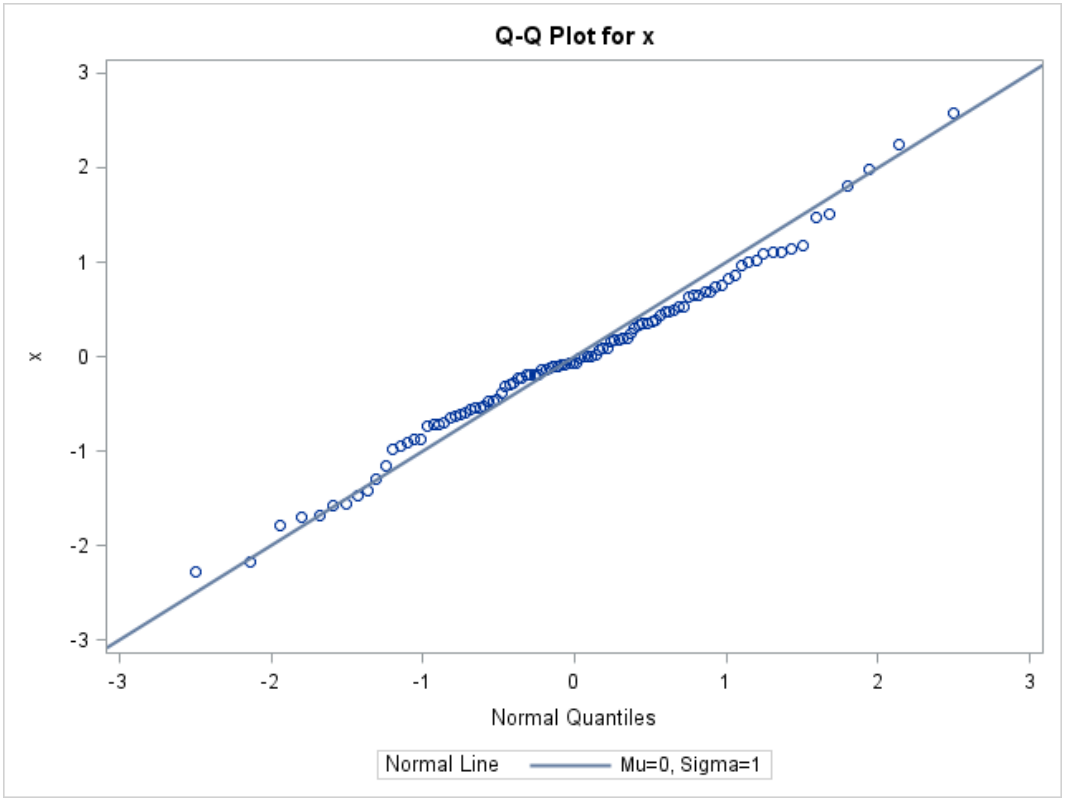
- If we specify multiple values in `c`, it will display every curve.



proc univariate+ qqplot

```
proc univariate data=newdata noprint;
    qqplot X/
          normal(mu=0 sigma=1);
run;
```

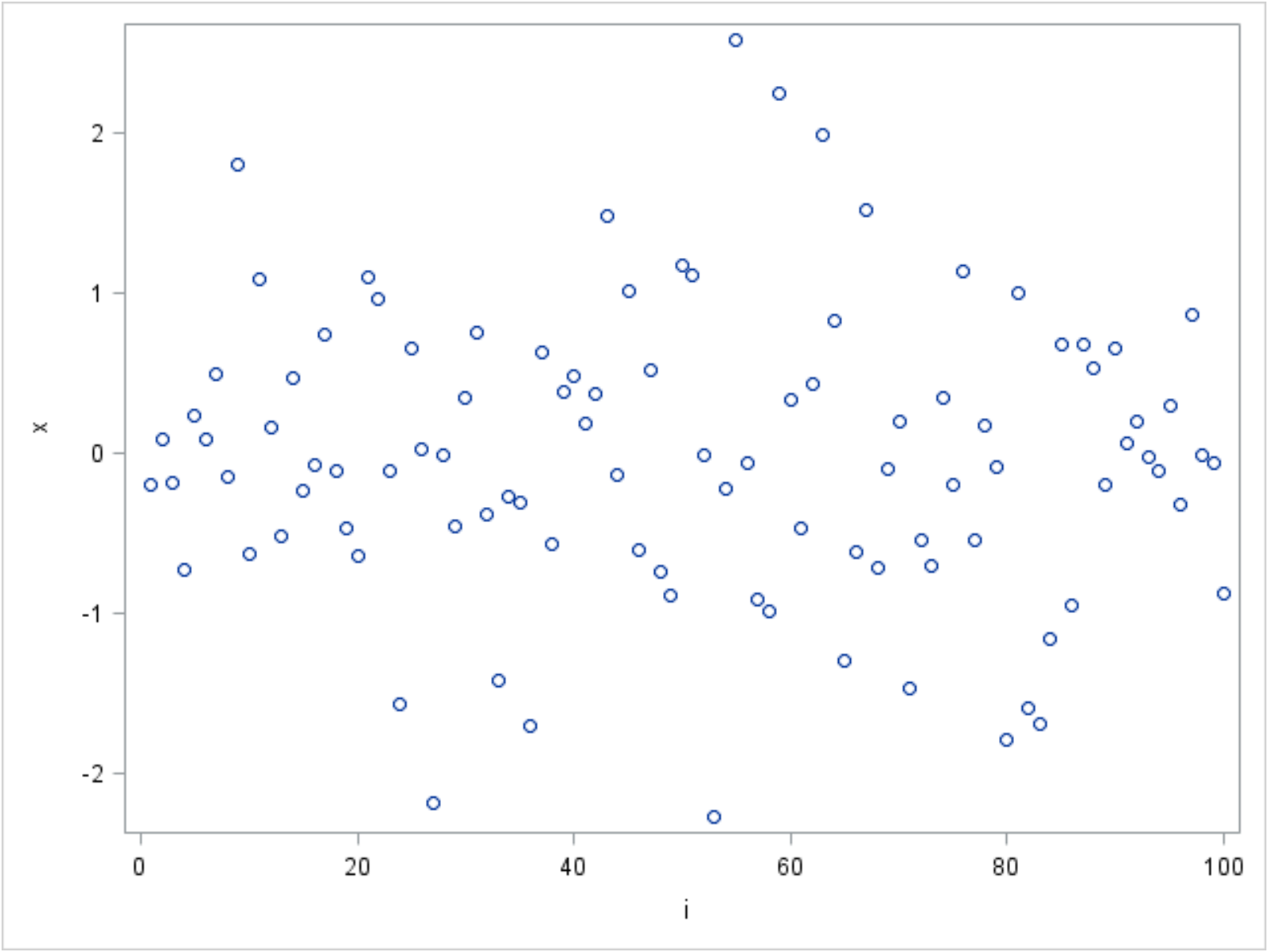
- You need to specify the line (`normal (mu=0 sigma=1)`) for reference.



sgrender+ scatterplot - 1

```
proc template;
  define statgraph minimumreq;
    begingraph;
      layout overlay;
        scatterplot x=i y=X;
      endlayout;
    endgraph;
  end;
run;
title "Scatter plot";
proc sgrender data=newdata template=minimumreq;
run;
```

sgrender+ scatterplot – 1 (plot)

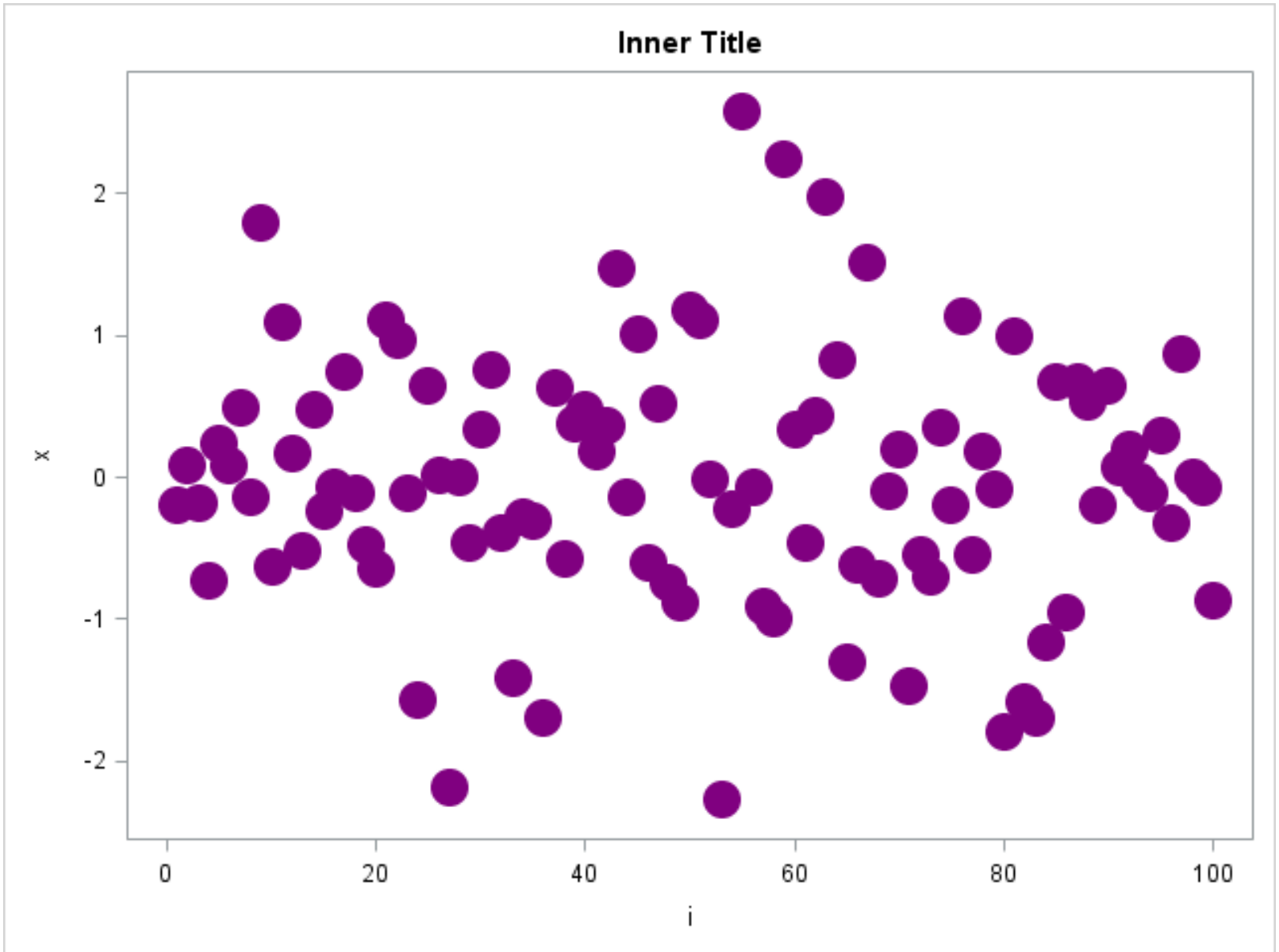


sgrender+ scatterplot - 2

```
proc template;
  define statgraph minimumreq;
    begingraph;
    entrytitle "Inner Title";
      layout overlay;
        scatterplot x=i y=X/
          markerattrs=(symbol=circlefilled size=20 color=purple);
      endlayout;
    endgraph;
  end;
run;
title "Scatter plot: change symbol";
proc sgrender data=newdata template=minimumreq;
run;
```

- `markerattrs`: controls the attributes of markers.
- `symbol`: DiamondFilled, SquareFilled, StarFilled.

sgrender+ scatterplot – 2 (plot)

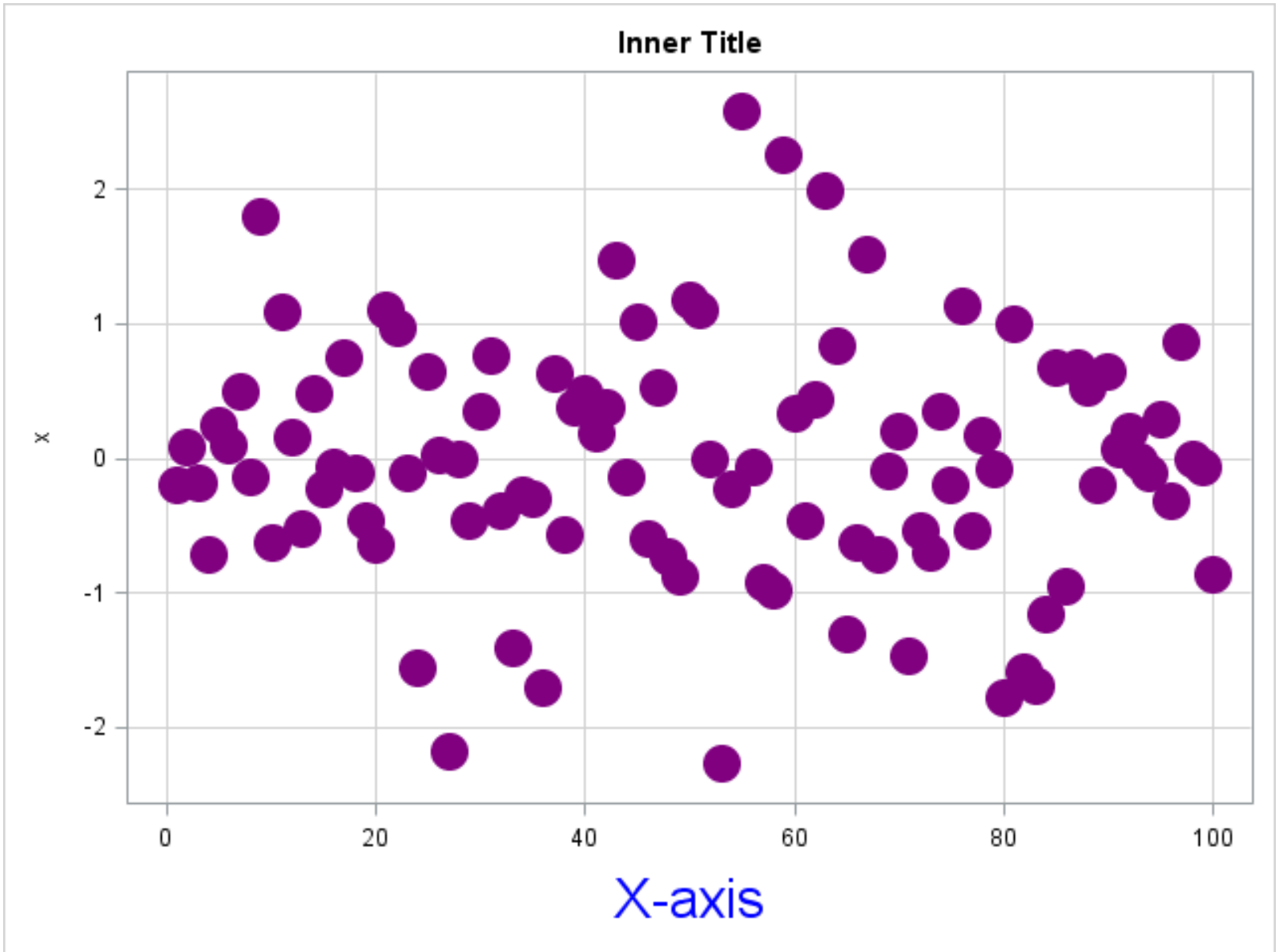


sgrender+ scatterplot – 3

```
proc template;
  define statgraph minimumreq;
    begingraph;
    entrytitle "Inner Title";
      layout overlay/
        xaxisopts=(griddisplay=on
          gridattrs=(color=lightgray)
          label="X-axis" labelattrs=(size=20 color=blue))
        yaxisopts=(griddisplay=on
          gridattrs=(color=lightgray));
        scatterplot x=i y=X/
          markerattrs=(symbol=circlefilled size=20 color=purple);
      endlayout;
    endgraph;
  end;
run;
title "Scatter plot: change axes layout";
proc sgrender data=newdata template=minimumreq;
run;
```

- `xaxisopts`: changes the features of `x` axis.

sgrender+ scatterplot – 3 (plot)

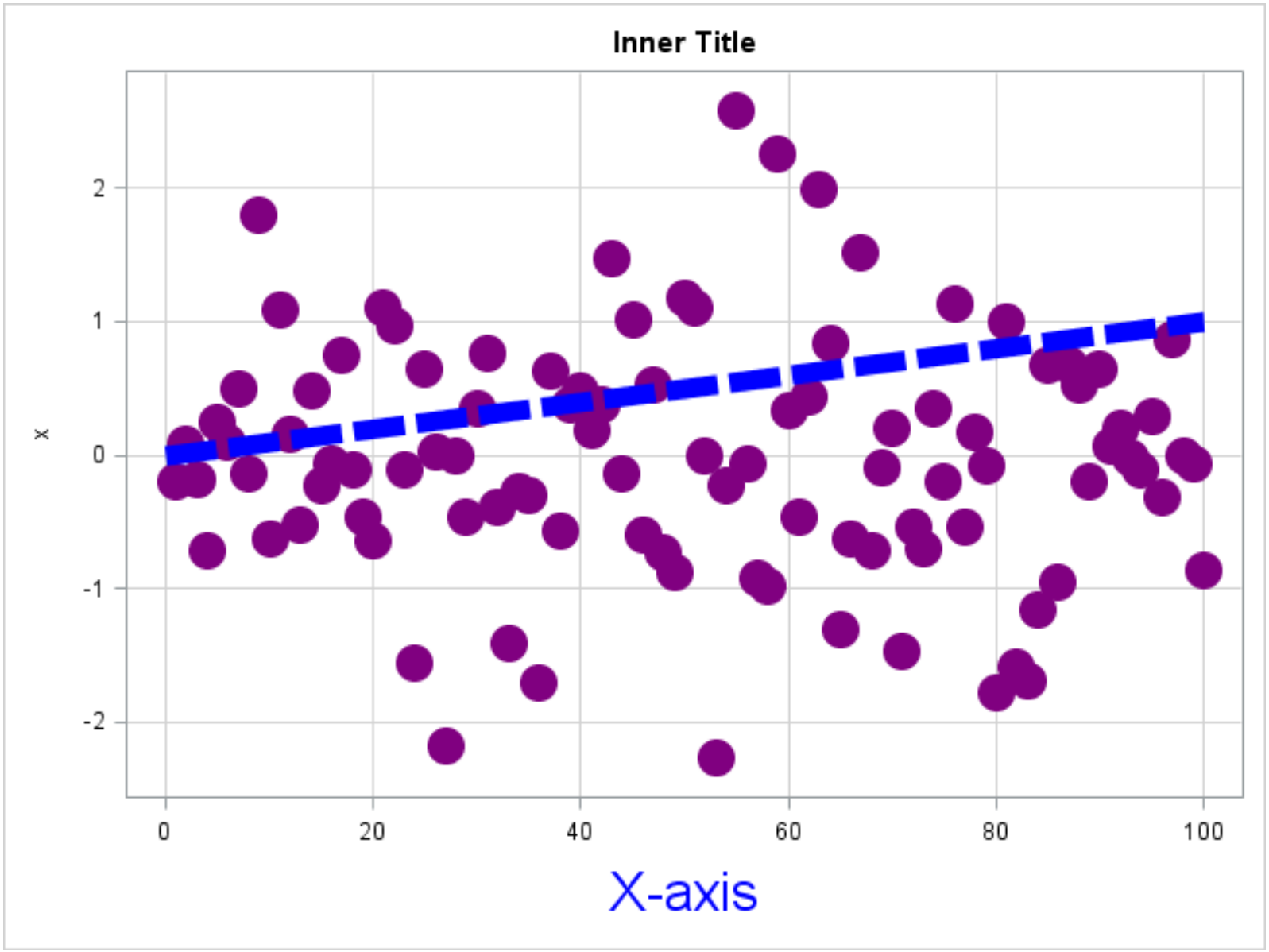


sgrender+ scatterplot – 4

```
proc template;
  define statgraph minimumreq;
    begingraph;
    entrytitle "Inner Title";
      layout overlay/
        xaxisopts=(griddisplay=on
          gridattrs=(color=lightgray)
          label="X-axis" labelattrs=(size=20 color=blue))
        yaxisopts=(griddisplay=on
          gridattrs=(color=lightgray));
        scatterplot x=i y=X/
          markerattrs=(symbol=circlefilled size=20 color=purple);
        lineparm x=0 y=0 slope=0.01/
          lineattrs=(color=blue pattern=5 thickness=10);
      endlayout;
    endgraph;
  end;
run;
title "Scatter plot: add a line";
proc sgrender data=newdata template=minimumreq;
run;
```

- `lineparm`: add a new line.

sgrender+ scatterplot – 4 (plot)

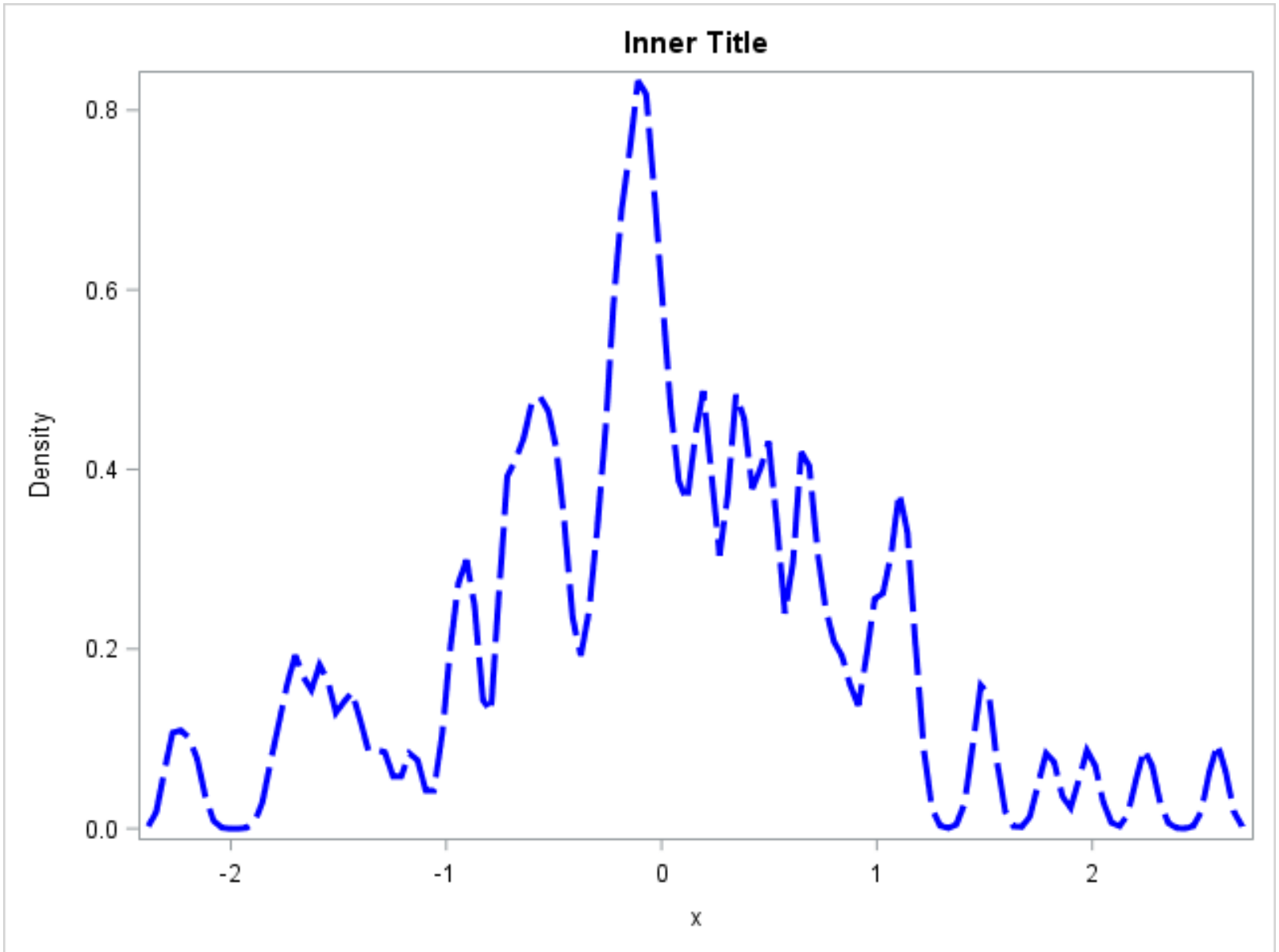


sgrender+ densityplot - 1

```
proc template;
  define statgraph densityplot1;
    begingraph;
    entrytitle "Inner Title";
      layout overlay;
        densityplot X/ kernel(c=0.1)
          lineattrs=(thickness=3 color=blue pattern=5);
      endlayout;
    endgraph;
  end;
run;
title "Density plot: KDE";
proc sgrender data=newdata template=densityplot1;
run;
```

- `lineattrs`: controls the attributes of the density curve.

sgrender+ densityplot – 1 (plot)

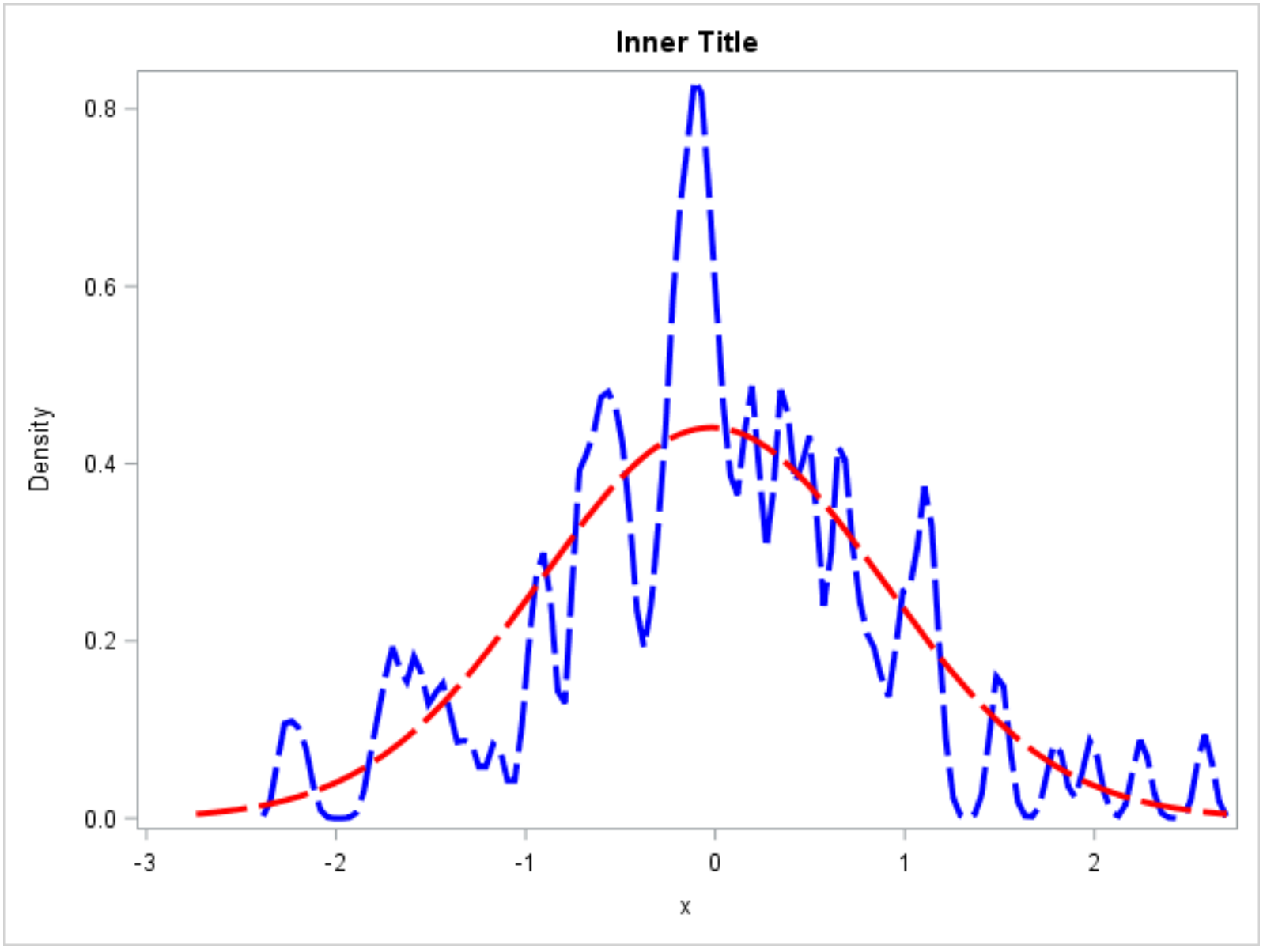


sgrender+ densityplot - 2

```
proc template;
  define statgraph densityplot1;
    begingraph;
    entrytitle "Inner Title";
      layout overlay;
        densityplot X/ kernel(c=0.1)
          lineattrs=(thickness=3 color=blue pattern=5);
          densityplot X/ normal()
          name="n" lineattrs=(thickness=3 color=red pattern=5);
      endlayout;
    endgraph;
  end;
run;
title "Density plot: KDE + Gaussian";
proc sgrender data=newdata template=densityplot1;
run;
```

- This overlays KDE and normal fit together.

sgrender+ densityplot – 2 (plot)

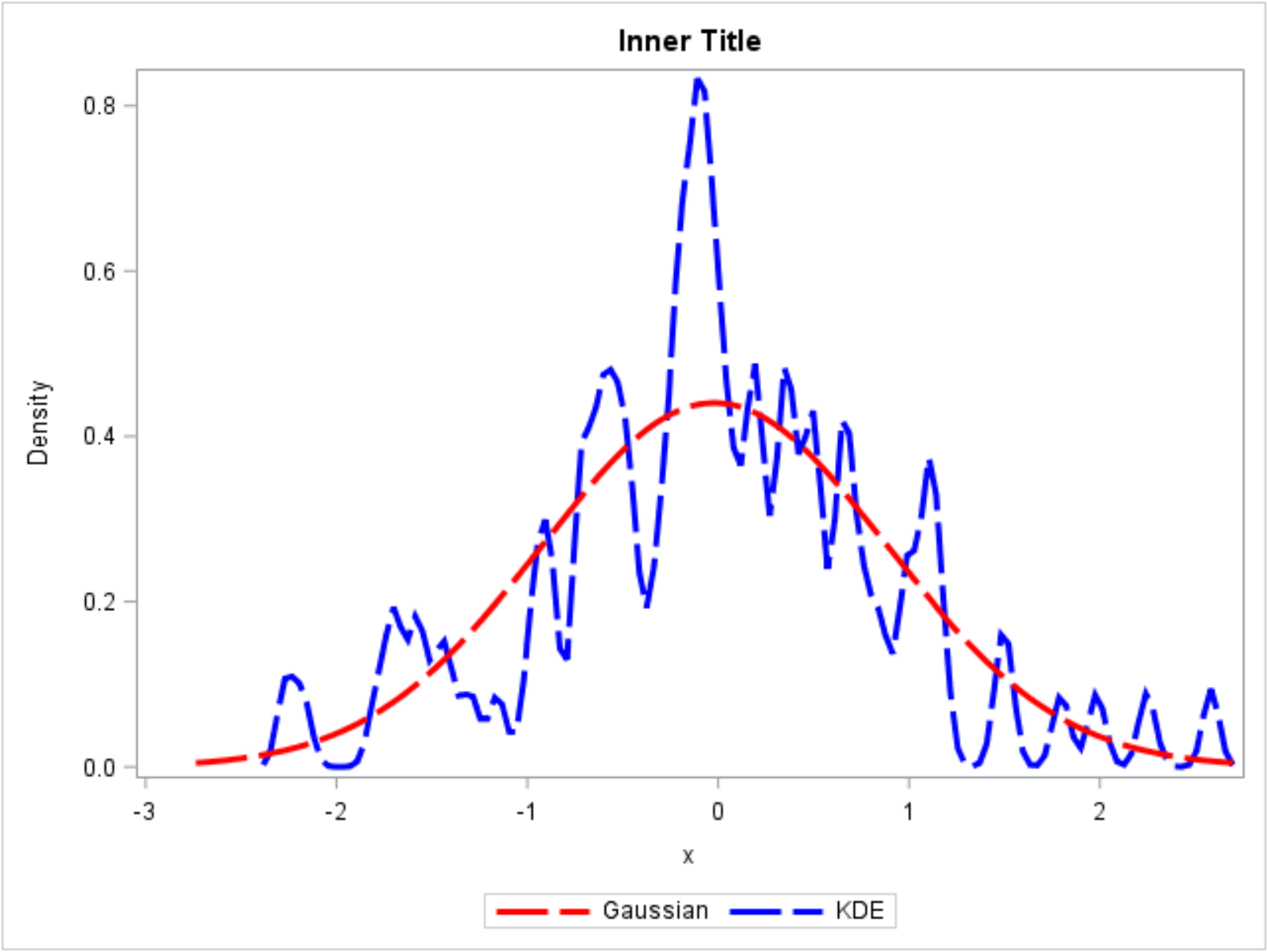


sgrender+ densityplot – 3

```
proc template;
  define statgraph densityplot1;
    begingraph;
    entrytitle "Inner Title";
      layout overlay;
        densityplot X/ kernel(c=0.1)
          lineattrs=(thickness=3 color=blue pattern=5)
          name="k" legendlabel="KDE";
          densityplot X/ normal()
          name="n" lineattrs=(thickness=3 color=red pattern=5)
          legendlabel="Gaussian";
          discretelegend "n" "k";
      endlayout;
    endgraph;
  end;
run;
title "Density plot: legends";
proc sgrender data=newdata template=densityplot1;
run;
```

- `discretelegend`, `legendlabel`, and `name`: to specify the legend.
- Highly recommend you to practice changing them to understand their functions.

sgrender+ densityplot – 3 (plot)



Generating data with labels

```
data newdata;
do i = 1 to 100;
        x = rand('normal');
        if i > 40 then G="Group 2";
        else G= "Group 1";
        output;
end;
run;
title "Random Normal";
proc print data=newdata;
run;
```

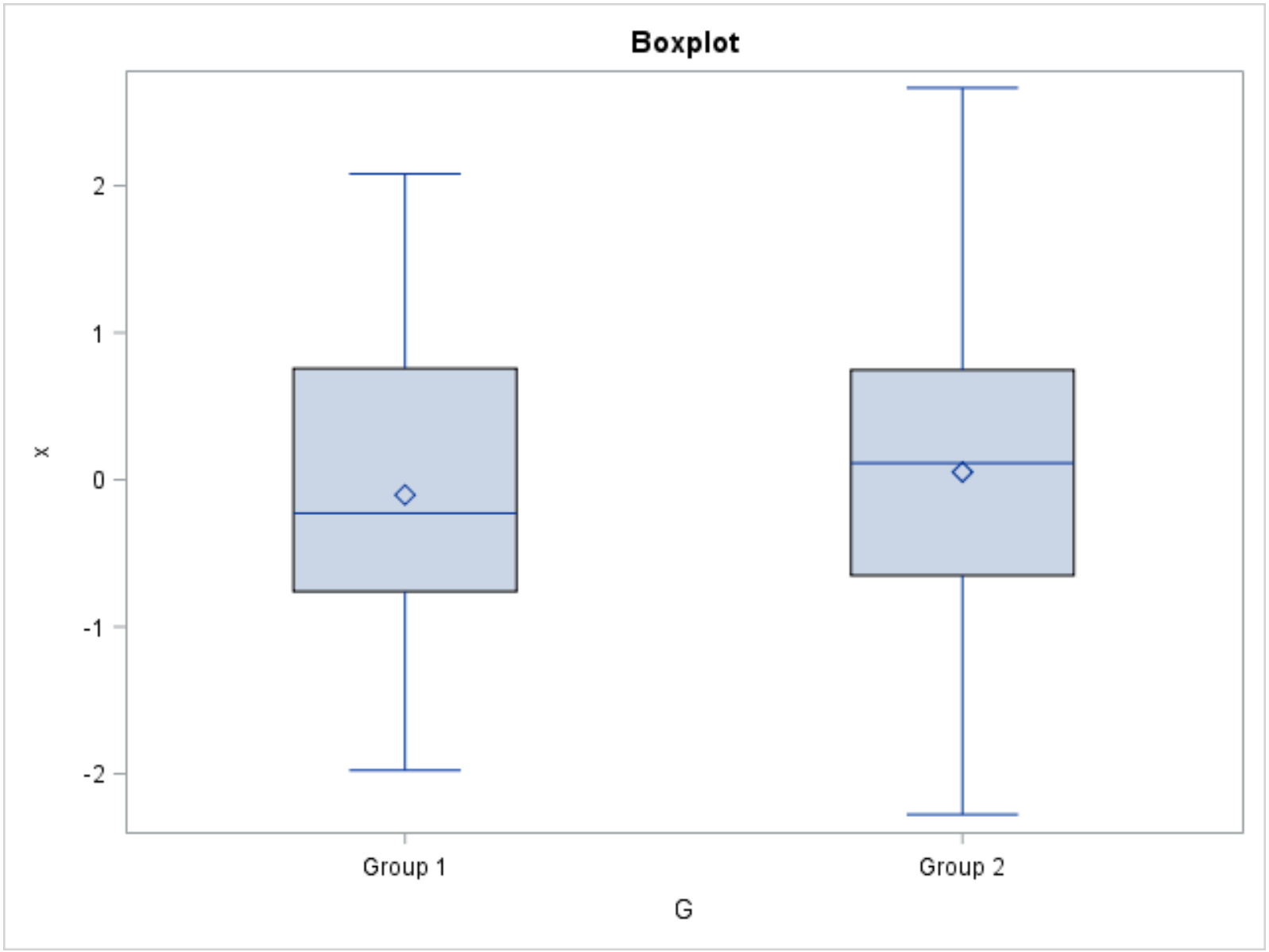
- This generates the standard normal with first 40 observations with variable `G` being 'Group 1' and other 60 observations have `G=` 'Group 2'.

sgrender + boxplot – 1

```
proc template;
  define statgraph boxplot;
    begingraph;
      entrytitle "Boxplot";
      layout overlay ;
        boxplot y=X x=G;
      endlayout;
    endgraph;
  end;
run;
title "Show boxplot";
proc sgrender data=newdata template=boxplot;
run;
```

- This creates a boxplot for variable `X` by variable `G`.

sgrender+ boxplot – 1 (plot)

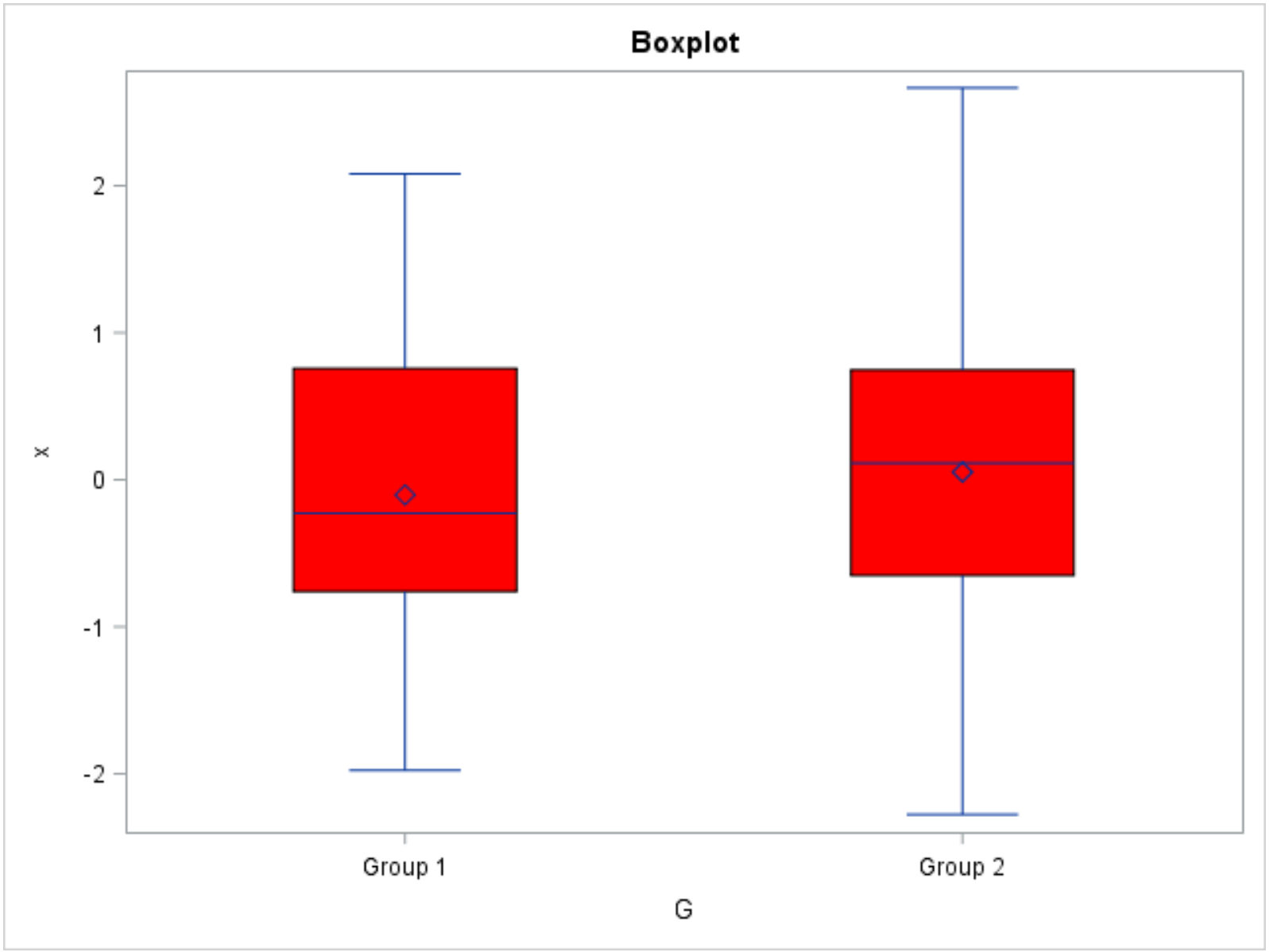


sgrender + boxplot – 2

```
proc template;
  define statgraph boxplot;
    begingraph;
      entrytitle "Boxplot";
      layout overlay ;
        boxplot y=X x=G/
        fillattrs = (color=red);
      endlayout;
    endgraph;
  end;
run;
title "Show boxplot with color";
proc sgrender data=newdata template=boxplot;
run;
```

- This changes the color into red.

sgrender+ boxplot – 2 (plot)



PMF – 1

```
data density;
do i = 0 to 10;
pmf = PMF('Binomial',i,0.45,10);
cdf = CDF('Binomial',i,0.45,10);
output;
end;
run;
title 'Binomial Distribution (p=0.45, n=10)';
proc print data = density noobs;
run;
```

- `PMF`: computing the probability mass function for a given distribution.
- `CDF`: computing the cumulative distribution function for a given distribution.

PMF – 1 (plot)

Binomial Distribution (p=0.45, n=10)

i	pmf	cdf
0	0.00253	0.00253
1	0.02072	0.02326
2	0.07630	0.09956
3	0.16648	0.26604
4	0.23837	0.50440
5	0.23403	0.73844
6	0.15957	0.89801
7	0.07460	0.97261
8	0.02289	0.99550
9	0.00416	0.99966
10	0.00034	1.00000

PMF – 2

```
data density;
do i = 0 to 10;
pmf = PMF('Poisson',i,5);
cdf = CDF('Poisson',i,5);
output;
end;
run;
title 'Possion Distribution (lambda=5)';
proc print data = density noobs;
run;
```

PMF – 2 (plot)

Possion Distribution (5)

i	pmf	cdf
0	0.00674	0.00674
1	0.03369	0.04043
2	0.08422	0.12465
3	0.14037	0.26503
4	0.17547	0.44049
5	0.17547	0.61596
6	0.14622	0.76218
7	0.10444	0.86663
8	0.06528	0.93191
9	0.03627	0.96817
10	0.01813	0.98630

PDF – 1

```
data density;
do i = 0 to 10 by 0.1;
pmf = PMF('normal',i,2);
cdf = CDF('normal',i,2);
output;
end;
run;
title 'Normal Distribution (mu=2)';
proc print data = density noobs;
run;
```

PDF – 2

```
data density;
do i = 0 to 10 by 0.1;
pmf = PMF('exponential',i,2);
cdf = CDF('exponential',i,2);
output;
end;
run;
title 'Expoenential Distribution (lambda=2)';
proc print data = density noobs;
run;
```

In-class Exercise

1. Generate 100 random points from $N(5, 2)$. Print out the data table.
2. Use `proc univariate` to get the summary table. What are the `mean`? `variance`? `skewness`? and `kurtosis`?
3. Show the histogram of the data and attach a fitted normal curve to it.
4. Show the histogram of the data and attach the density curve based on the KDE with smoothing bandwidth 0.5.
5. Use QQplot to compare the data points to the quantile of $N(5, 2)$.
6. Show the scatter plot of $x =$ the index of each observation and $y =$ its value.
7. Based on the previous scatter plot, add a line starting from $x = 0, y = 5$ with slope $= 0.02$ (this corresponds to the equation $y = 5 + 0.02x$).