# Homework2

**Step 1:**

Simulate 200 observations from the following linear model: Y = alpha + beta1 \* X1 + beta2 \* X2 + noise where • alpha=1, beta1=2, beta2=-1.5 • X1 ~ N(1, 4), X2 ~ N(3,1), noise ~ N(0,1)

**Program:**

DATA LinearSimulation(keep=X1 X2 Y);

TITLE Simulation;

alpha=1;

beta1=2;

beta2=-1.5;

DO i = 1 TO 200; /\*200 observations \*/

UnifVals = rand("Uniform"); /\*U(0,1)\*/

X1 = 1 + (4-1)\*UnifVals; /\*Given X1 ~ (1,4) \*/

X2 = 3 + (1-3)\*UnifVals; /\*Given X2 ~ (3,1) \*/

noise=UnifVals; /\*Given noise ~ N(0,1)\*/

Y = alpha+beta1\*X1+beta2\*X2+noise;

OUTPUT;

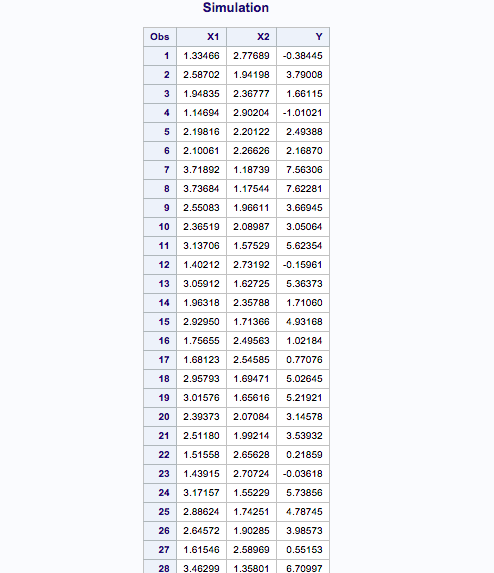
END;

RUN;

PROC PRINT DATA=LinearSimulation LABEL;

RUN;

**Output:**

****

**Step 2 & 3:**

Define a new binary variable Y\_bin such that Y\_bin=1 if Y>0 and Y\_bin=0 otherwise

Make the final data contain only 4 variables: X1, X2, Y and Y\_bin.

**Program:**

DATA LinearSimulation(keep=X1 X2 Y Y\_bin);

TITLE Simulation;

alpha=1;

beta1=2;

beta2=-1.5;

DO i = 1 TO 200; /\*200 observations \*/

UnifVals = rand("Uniform"); /\*U(0,1)\*/

X1 = 1 + (4-1)\*UnifVals; /\*Given X1 ~ (1,4) \*/

X2 = 3 + (1-3)\*UnifVals; /\*Given X2 ~ (3,1) \*/

noise=UnifVals; /\*Given noise ~ N(0,1)\*/

Y = alpha+beta1\*X1+beta2\*X2+noise;

if(Y>0)then Y\_bin=1;

ELSE Y\_bin=0;

OUTPUT;

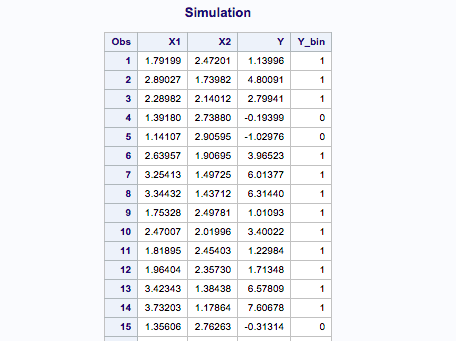
END;

RUN;

PROC PRINT DATA=LinearSimulation LABEL;

RUN;

**Output:**

****

**Step 4:**

Calculate the range of X1 and X2 (only range and no other statistics).

**Program:**

DATA LinearSimulation(keep=X1 X2 Y Y\_bin);

TITLE Simulation;

alpha=1;

beta1=2;

beta2=-1.5;

DO i = 1 TO 200; /\*200 observations \*/

UnifVals = rand("Uniform"); /\*U(0,1)\*/

X1 = 1 + (4-1)\*UnifVals; /\*Given X1 ~ (1,4) \*/

X2 = 3 + (1-3)\*UnifVals; /\*Given X2 ~ (3,1) \*/

noise=UnifVals; /\*Given noise ~ N(0,1)\*/

Y = alpha+beta1\*X1+beta2\*X2+noise;

if(Y>0)then Y\_bin=1;

ELSE Y\_bin=0;

OUTPUT;

END;

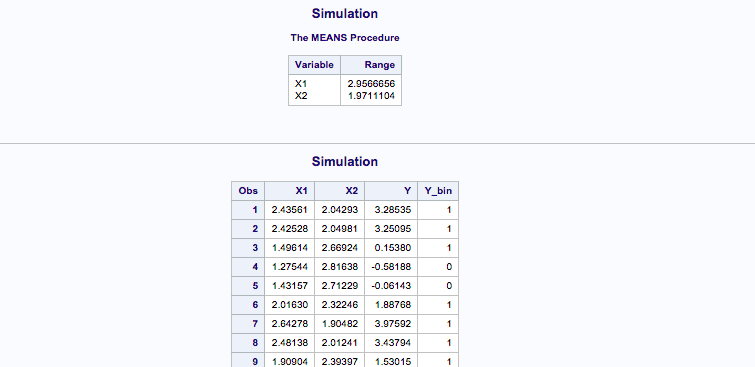
RUN;

PROC MEANS DATA=LinearSimulation RANGE;

VAR X1 X2;

PROC PRINT DATA=LinearSimulation LABEL;

RUN;

****

**Step 5:**

Check if Y follows a normal distribution. What graphics and statistics would you look into?

ANS:

After making histogram data, if the shape of the distribution resembles bell curve the data is likely normal.

Graphic: Plotting histogram and normal distribution curve

If the data meets the requirement of 68-95-99, that means 68% of the data should be in the range of one standard deviation, 95% data in the range of two standard deviations and 99% of the data in the range of 3 standard deviation

We look statistics of Mean, Mode and standard deviation. In normal distribution mean and mode are equal.

**Program:**

DATA LinearSimulation(keep=X1 X2 Y Y\_bin);

TITLE Simulation;

alpha=1;

beta1=2;

beta2=-1.5;

DO i = 1 TO 200; /\*200 observations \*/

UnifVals = rand("Uniform"); /\*U(0,1)\*/

X1 = 1 + (4-1)\*UnifVals; /\*Given X1 ~ (1,4) \*/

X2 = 3 + (1-3)\*UnifVals; /\*Given X2 ~ (3,1) \*/

noise=UnifVals; /\*Given noise ~ N(0,1)\*/

Y = alpha+beta1\*X1+beta2\*X2+noise;

if(Y>0)then Y\_bin=1;

ELSE Y\_bin=0;

OUTPUT;

END;

RUN;

PROC MEANS DATA=LinearSimulation RANGE;

VAR X1 X2;

proc univariate;

VAR Y;

HISTOGRAM Y / NORMAL;

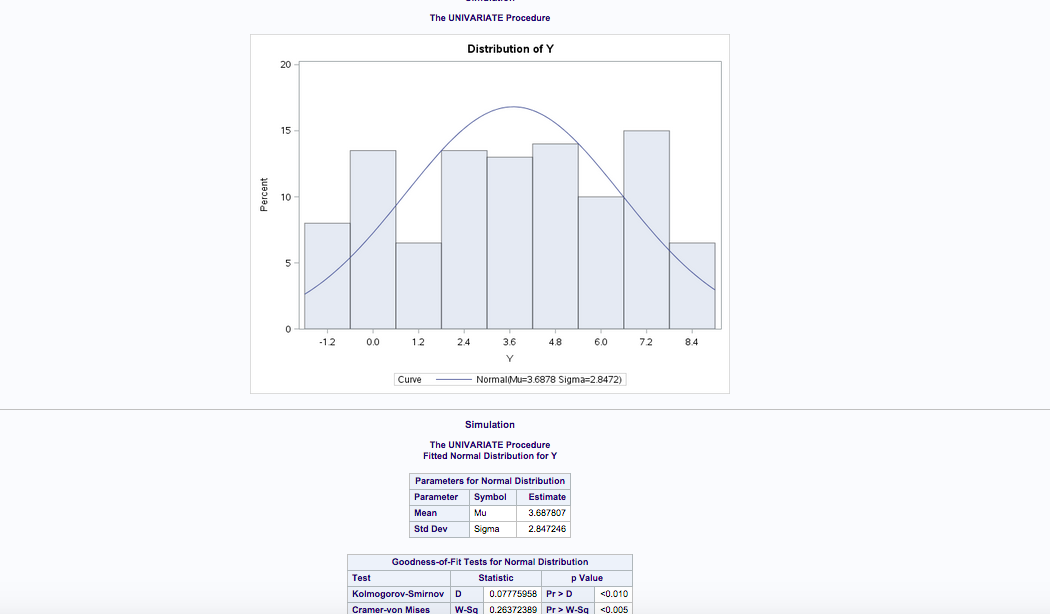
PROBPLOT Y / NORMAL;

run;

PROC PRINT DATA=LinearSimulation LABEL;

RUN;

**Output:**

****

**Step 6:**

Count how many ``1’’ s you observed in Y\_bin.

**Program:**

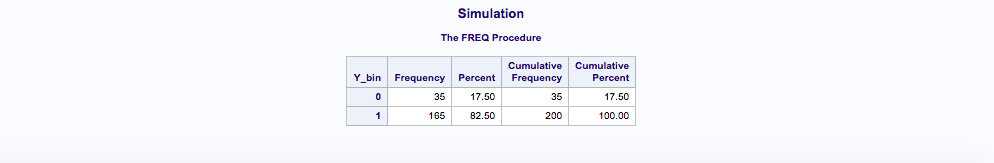
DATA YBinCount;

SET LinearSimulation;

PROC FREQ DATA=YBinCount;

TABLE Y\_bin;

RUN;

**Output:**

**Step 7:**

Create a new variable “sign” such that sign=“positive” when Y\_bin=1 and sign=“negative” otherwise.

**Program:**

DATA CreateSign;

SET LinearSimulation;

if(Y\_bin)then sign="Positive";

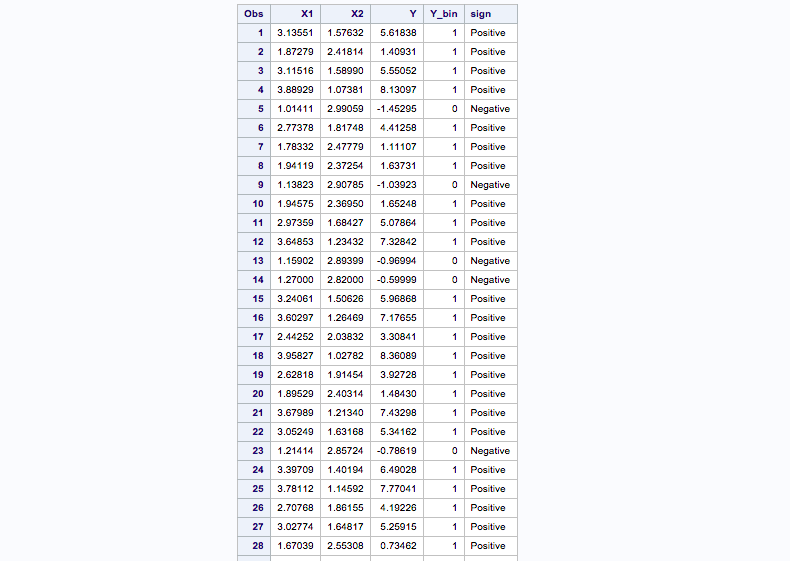
ELSE sign="Negative";

RUN;

PROC PRINT Data=CreateSign;

RUN;

**Output:**

****

**Step 8:**

Compare the distribution of X1 in the “positive” group and the “negative” group. What statistics would you look into?

Ans:

Mean, Median and standard deviation by running proc univariate command sorted by sign (either positive or negative)

**Program:**

PROC SORT DATA=CreateSign;

BY sign;

RUN; /\*in PROC UNIVARIATE\*/

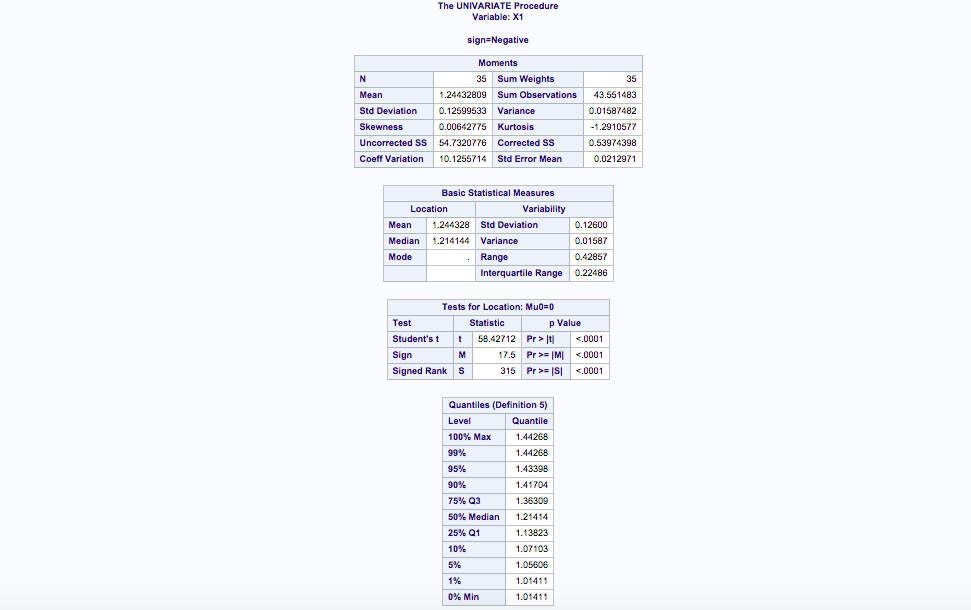
PROC UNIVARIATE DATA=CreateSign;

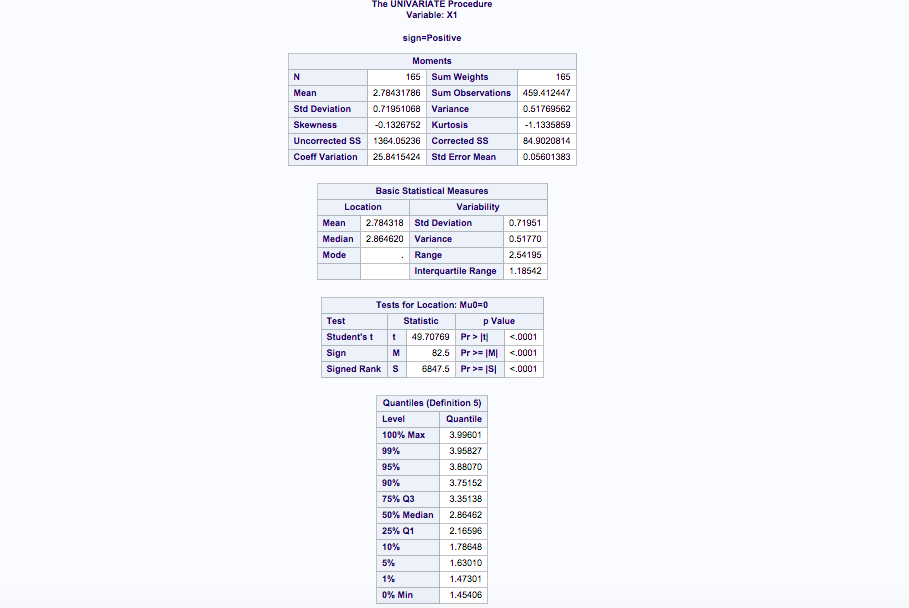
BY sign; /\* tells SAS to sort data by SIGN\*/

VAR X1; /\* tells SAS to produce statistics of X1\*/

RUN;

**Output:**

****

****