

SaS for Data Science

▼ Class	SAS
▼ Type	Self Learning
⚙ Status	Done

SAS-Variables - represent the column names of the data tables

▼ types

- numeric (INPUT VAR1 VAR2 VAR3; #Define numeric variables in the data set.)
- character (INPUT VAR1 \$ VAR2 \$ VAR3 \$; #Define character variables in the data set.)
- date (INPUT VAR1 DATE11. VAR2 MMDDYY10. ; #Define date variables in the data set.)

use of variables in sas program

```

1  /* sas variables - numeric, character, date */
2
3  DATA TEMP;
4  INPUT ID NAME $ SALARY DEPT $ DOJ DATE9. ;
5  FORMAT DOJ DATE9. ;
6  DATALINES;
7  1 Rick 623.3 IT 02APR2001
8  2 Dan 515.2 OPS 11JUL2012
9  3 Michelle 611 IT 21OCT2000
10 4 Ryan 729 HR 30JUL2012
11 5 Gary 843.25 FIN 06AUG2000
12 6 Tusar 578 IT 01MAR2009
13 7 Pranab 632.8 OPS 16AUG1998
14 8 Rasmi 722.5 FIN 13SEP2014
15 ;
16 PROC PRINT DATA = TEMP;
17 RUN;

```

OUTPUT

Obs	ID	NAME	SALARY	DEPT	DOJ
1	1	Rick	623.30	IT	02APR2001
2	2	Dan	515.20	OPS	11JUL2012
3	3	Michelle	611.00	IT	21OCT2000
4	4	Ryan	729.00	HR	30JUL2012
5	5	Gary	843.25	FIN	06AUG2000
6	6	Tusar	578.00	IT	01MAR2009
7	7	Pranab	632.80	OPS	16AUG1998
8	8	Rasmi	722.50	FIN	13SEP2014

SAS - String - values which are enclosed with in a pair of single quotes or are declared by adding a space and \$ sign at the end of the variable

- declaring string Variables

code:

```
data string_examples;
  LENGTH string1 $ 6 String2 $ 5;
  /*String variables of length 6 and 5 */
  String1 = 'Hello';
  String2 = 'World';
  Joined_strings = String1 ||String2 ;
run;
proc print data = string_examples noobs;
run;
```

output:

string1	String2	Joined_strings
Hello	World	Hello World

▼ string function

▼ SUBSTRN - This function extracts a substring using the start and end positions. In case of no end position is mentioned it extracts all the characters till end of the string.

input:

```
data string_examples;
  LENGTH string1 $ 6 ;
  String1 = 'Hello';
  sub_string1 = substrn(String1,2,4) ;
  /*Extract from position 2 to 4 */
  sub_string2 = substrn(String1,3) ;
  /*Extract from position 3 onwards */
run;
proc print data = string_examples noobs;
run;
```

output:

string1	sub_string1	sub_string2
Hello	ello	llo

▼ TRIMN - This function removes the trailing space form a string.

input:

```
data string_examples;  
  LENGTH string1 $ 7 ;  
  String1='Hello ';  
  length_string1 = lengthc(String1);  
  length_trimmed_string = lengthc(TRIMN(String1));  
run;  
proc print data = string_examples noobs;  
run;
```

output:

string1	length_string1	length_trimmed_string
Hello	7	5

SAS - Arrays - Arrays in SAS are used to store and retrieve a series of values using an index value. The index represents the location in a reserved memory area.

▼ array declaration - syntax(ARRAY ARRAY-NAME(SUBSCRIPT) (\$) VARIABLE-LIST ARRAY-VALUES)

input:

```

/*Declare an array of length 5 named AGE with values.*/
data sample1;
ARRAY AGE[5] (12 18 5 62 44);
proc print data=sample1;
run;

```

output:

Obs	AGE1	AGE2	AGE3	AGE4	AGE5
1	12	18	5	62	44

Declare an array of length 5 named AGE with values.ARRAY AGE[5] (12 18 5 62 44);

Declare an array of length 5 named COUNTRIES with values starting at index 0.ARRAY COUNTRIES(0:8) A B C D E F G H I;

Declare an array of length 5 named QUESTS which contain character values.ARRAY QUESTS(1:5) \$ Q1-Q5;

Declare an array of required length as per the number of values supplied.ARRAY ANSWER(*) A1-A100;

▼ accessing array values

input:

```

/* ACCESS DATA IN ARRAY */
DATA array_example;
INPUT a1 $ a2 $ a3 $ a4 $ a5 $ a6 $;
ARRAY name(6) $ a1-a6;
DATALINES;
Aman Riddhi Nirmal Anu Sameer bobby
;
RUN;
PROC PRINT DATA = array_example;
RUN;

```

output:

Obs	a1	a2	a3	a4	a5	a6
1	Aman	Riddhi	Nirmal	Anu	Sameer	bobby

▼ using the OF operator

input:

```

DATA array_example_OF;
INPUT A1 A2 A3 A4;
ARRAY A(4) A1-A4;
A_SUM = SUM(OF A(*));
A_MEAN = MEAN(OF A(*));
A_MIN = MIN(OF A(*));
DATALINES;
21 4 52 11
96 25 42 6
;
RUN;
PROC PRINT DATA = array_example_OF;
RUN;

```

output:

Obs	A1	A2	A3	A4	A_SUM	A_MEAN	A_MIN
1	21	4	52	11	88	22.00	4
2	96	25	42	6	169	42.25	6

▼ using the IN operator

input:

```
DATA array_in_example;
  INPUT A1 $ A2 $ A3 $ A4 $;
  ARRAY COLOURS(4) A1-A4;
  IF 'yellow' IN COLOURS THEN available = 'Yes'; ELSE available = 'No';
  DATALINES;
  Orange pink violet yellow
;
RUN;
PROC PRINT DATA = array_in_example;
RUN;
```

output:

Obs	A1	A2	A3	A4	available
1	Orange	pink	violet	yellow	Yes

SAS - Numeric formats

- reading numeric formats
- displaying numeric formats

SAS - Operators

- ▼ arithmetic operators

- addition (+)
- subtraction (-)
- multiplication (*)
- division (/)
- exponentiation (**)

code:

input:

```
data arithmeticdata;
input @1col1 @7col2;
add_res = col1+col2;
sub_res = col1-col2;
mul_res = col1*col2;
divi_res = col1/col2;
expo_res = col1**col2;
datalines;
11.21 5.3
3.11 11
;
proc print data=arithmeticdata;
run;
```

output:

Obs	col1	col2	add_res	sub_res	mul_res	divi_res	expo_res
1	11.21	5.3	16.51	5.91	59.413	2.11509	365518.82
2	3.11	11.0	14.11	-7.89	34.210	0.28273	263247.51

▼ logical operators

- AND (&)
- OR (|)
- NOT (~)

code:

input:

```
Data mydata;  
input @1 col1 5.2 @7 col2 4.1;  
and_=(col1 > 10 & col2 > 5);  
or_=(col1 > 12 | col2 > 15);  
not_=(col1 > 7);  
datalines;  
11.21 15.3  
33.11 11.4  
;  
run;  
proc print data=mydata;  
run;
```

output:

Obs	col1	col2	and_	or_	not_
1	11.21	15.3	1	1	0
2	33.11	11.4	1	1	0

▼ comparison operators

- EQUAL (=)
- NOT EQUAL (^=)
- LESS THAN (<)
- LESS THAN OR EQUAL TO (≤)
- GREATER THAN (>)
- GREATER THAN OR EQUAL TO (≥)
- IN

code:

input:

```

data mydata1;
input @1col1 5.2    @7col2 4.1;
EQ_=(col1 = 11.21);
Neq_=(col1 ^= 11.21);
gt_=(col2 =>8);
lt_=(col2 <= 12);
in_=col2 in(6.2,5.3,12 );
max_=max(col1,col2);
min_=min(col1,col2);
datalines;
11.21 15.3
3.11  11.4
;
proc print data=mydata1;
run;

```

output:

Obs	col1	col2	EQ_	Neq_	gt_	lt_	in_	max_	min_
1	11.21	15.3	1	0	1	0	0	15.3	11.21
2	3.11	11.4	0	1	1	1	0	11.4	3.11

▼ minimum/maximum operators

- MIN
- MAX

code: above explained

▼ concatenation operator

- CONCATINATION (||)

code:

input:

```

data mydata123;
input col1$ col2$ col3$;
concat_=(col1 || col2 || col3);
datalines;
Hello Uni World
Aman Nirmal Sameer
;
proc print data=mydata123;
run;

```

output:

Obs	col1	col2	col3	concat_
1	Hello	Uni	World	Hello Uni World
2	Aman	Nirmal	Sameer	Aman Nirmal Sameer

SAS - LOOPS

▼ DO

▼ DO - Index loop

input:

```

data mydata129;
sum = 0;
DO var = 1 to 5;
    sum = sum + var;
END;
proc print data=mydata129;
run;

```

output:

Obs	sum	var
1	15	6

▼ DO - While loop

input:

```
/* DO WHILE (variable condition)
END; */
data data1;
sum = 0;
var = 1;
DO while(var < 10);
  sum = sum + var;
  var+1;
END;
proc print;
run;
```

output:

Obs	sum	var
1	45	10

▼ DO - Until loop

input:

```

/* DO UNTIL(variable condition)
END;
*/

data data2;
sum = 0;
var = 1;
DO Until(Var > 5);
sum = sum + var;
var+1;
end;
proc print;
run;

```

output:

Obs	sum	var
1	45	10

SAS - Conditional statements

data:

Obs	ID	NAME	SALARY	DEPT	DOJ
1	1	Rick	623.30	IT	02APR2001
2	2	Dan	515.20	OPS	11JUL2012
3	3	Michelle	611.00	IT	21OCT2000
4	4	Ryan	729.00	HR	30JUL2012
5	5	Gary	843.25	FIN	06AUG2000
6	6	Tusar	578.00	IT	01MAR2009
7	7	Pranab	632.80	OPS	16AUG1998
8	8	Rasmi	722.50	FIN	13SEP2014

▼ IF statement

code:

input:

```
/* IF */  
data emp1;  
set empdata;  
if salary>600;  
proc print data=emp1;  
run;
```

output:

Obs	ID	NAME	SALARY	DEPT	DOJ
1	1	Rick	623.30	IT	02APR2001
2	3	Michelle	611.00	IT	21OCT2000
3	4	Ryan	729.00	HR	30JUL2012
4	5	Gary	843.25	FIN	06AUG2000
5	7	Pranab	632.80	OPS	16AUG1998
6	8	Rasmi	722.50	FIN	13SEP2014

▼ IF-THEN-ELSE statement

code:

input:

```
/* if-then-else */  
data emp2;  
set empdata;  
if salary>600 then salrange = "high";  
else salrange = "low";  
proc print data=emp2;  
run;
```

output:

Obs	ID	NAME	SALARY	DEPT	DOJ	salrange
1	1	Rick	623.30	IT	02APR2001	high
2	2	Dan	515.20	OPS	11JUL2012	low
3	3	Michelle	611.00	IT	21OCT2000	high
4	4	Ryan	729.00	HR	30JUL2012	high
5	5	Gary	843.25	FIN	06AUG2000	high
6	6	Tusar	578.00	IT	01MAR2009	low
7	7	Pranab	632.80	OPS	16AUG1998	high
8	8	Rasmi	722.50	FIN	13SEP2014	high

▼ IF-THEN-ELSE-IF statement

code:

input:

```
/* if-then-else-if */
data emp3;
set empdata;
if salary < 600 then salrange = "low";
else if 600 <= salary <= 700 then salrange = "medium";
else if 700 < salary then salrange = "high";
proc print data=emp3;
run;
```

output:

Obs	ID	NAME	SALARY	DEPT	DOJ	salrange
1	1	Rick	623.30	IT	02APR2001	med
2	2	Dan	515.20	OPS	11JUL2012	low
3	3	Michelle	611.00	IT	21OCT2000	med
4	4	Ryan	729.00	HR	30JUL2012	hig
5	5	Gary	843.25	FIN	06AUG2000	hig
6	6	Tusar	578.00	IT	01MAR2009	low
7	7	Pranab	632.80	OPS	16AUG1998	med
8	8	Rasmi	722.50	FIN	13SEP2014	hig

▼ IF-THEN-DELETE statement

code:

input:

```
/* if-else-delete */
data emp4;
set empdata;
if salary > 600 then delete;
proc print data = emp4;
run;
```

output:

Obs	ID	NAME	SALARY	DEPT	DOJ
1	2	Dan	515.2	OPS	11JUL2012
2	6	Tusar	578.0	IT	01MAR2009

SAS - Functions

▼ Mathematical

- MAX
- MIN
- MEDIAN
- RANUNI
- SQRT

input:

```
/* MATHEMATICAL FUNCTIONS */
data Math_functions;
a1=21;a2=34;a3=53;a4=57;a5=32;a6=98;a7=2;

max_val = MAX(a1,a2,a3,a4,a5,a6,a7);
min_val = MIN(a1,a2,a3,a4,a5,a6,a7);
med_val = MEDIAN(a1,a2,a3,a4,a5,a6,a7);
ran_val = RANUNI(0);
sq_val = SQRT(sum(a1,a2,a3,a4,a5,a6,a7));

proc print data = Math_functions noobs;
run;
```

output:

a1	a2	a3	a4	a5	a6	a7	max_val	min_val	med_val	ran_val	sq_val
21	34	53	57	32	98	2	98	2	34	0.56462	17.2337

▼ Date and time

- INTCK()
- WEEKDAY()
- TODAY()
- TIME()

code:

input:

```
/* date function */
data data_func;
input @1 date1 date9. @11 date2 date9.;
format date1 date9. date2 date9.;
years = INTCK("Year",date1,date2);
months = INTCK("Months",date1,date2);
weekday = WEEKDAY(date1);
todaydate = TODAY();
time = TIME();
datalines;
27OCT2020 16APR1990
01MAR2009 11JUL2012
;
PROC PRINT DATA = data_func noobs;
run;
```

output:

date1	date2	years	months	date2	weekday	todaydate	time
27OCT2020	16APR1990	-30	.	.	3	23027	13424.97
01MAR2009	11JUL2012	3	.	.	1	23027	13424.97

▼ Character

- LOWCASE
- UPCASE
- REVERSE
- SCAN

code:

input:

```

/* character functions */
data char_func;
  lowercase = LOWCASE("HELLO");
  upcase = UPCASE("hello");
  reverse = REVERSE("Hello");
  returnNthWord = SCAN("sas with riddhi",2);
run;
proc print data = char_func noobs;
run;

```

output:

lowercase	upcase	reverse	returnNthWord
hello	HELLO	olleH	with

▼ Truncation

- CEIL
- FLOOR
- INT
- ROUND

code:

input:

```

/* truncation function */
data trunc_func;
  nearGreatInt = ceil(11.85);
  nearSmallInt = floor(11.85);
  intportion = Int(32.41);
  roundOff = ROUND(54218.98);
run;
proc print data = trunc_func noobs;
run;

```

output:

nearGreatInt	nearSmallInt	intportion	roundOff
12	11	32	54219

▼ Miscellaneous

- ZIPSTATE()
- MORT()

input:

```

data misc_func;
  state = zipstate('85001');
proc print data = misc_func noobs;
run;

```

output:

state
AZ

SAS - Input methods

▼ List Input Method

```
/* list input methods */  
data temp;  
input empid ename$ dept$;  
datalines;  
1 Aman IT  
2 Riddhi FIN  
3 Nirmal OPS  
4 Anu IT  
;  
proc print data = temp;  
run;
```

output:

Obs	empid	ename	dept
1	1	Aman	IT
2	2	Riddhi	FIN
3	3	Nirmal	OPS
4	4	Anu	IT

▼ Named Input Method

```
/* name input method*/  
data temp1;  
input  
EMPID= ENAME=$ DEPT=$;  
datalines;  
EMPID=1 ENAME=ANU DEPT=IT  
EMPID=2 ENAME=AMAN DEPT=FIN  
EMPID=3 ENAME=PRIYANSHU DEPT=OPS  
;  
proc print data = temp1;  
run;
```

output:

Obs	EMPID	ENAME	DEPT
1	1	ANU	IT
2	2	AMAN	FIN
3	3	PRIYANSH	OPS

▼ Column Input Method

```

/* column input method */
data temp3;
input EMPID 1-3 ENAME $ 4-12 DEPT $ 13-16;
DATALINES;
14 ANU IT
241 RIDDHI OPS
70 AMAN FIN
410 SAMEER IT
420 PRIYANSHU OPS
530 NIRMAL FIN
;
PROC PRINT DATA = temp3;
run;

```

output:

▼ Formatted Input Method

```

/* formatted input method */
data mydata;
input @1 EmpID @4 EName $ @13 Dept $;
datalines;
15 Anu      IT
214 Ram     OPS
30 sameer   IT
410 Sanvi   OPS
52 Payal    FIN
;
proc print data=mydata;
run;

```

output:

Obs	EmpID	EName	Dept
1	15	Anu	IT
2	214	Ram	OPS
3	30	sameer	IT
4	410	Sanvi	OPS
5	52	Payal	FIN

SAS - BAR CHART

create data according to need

```

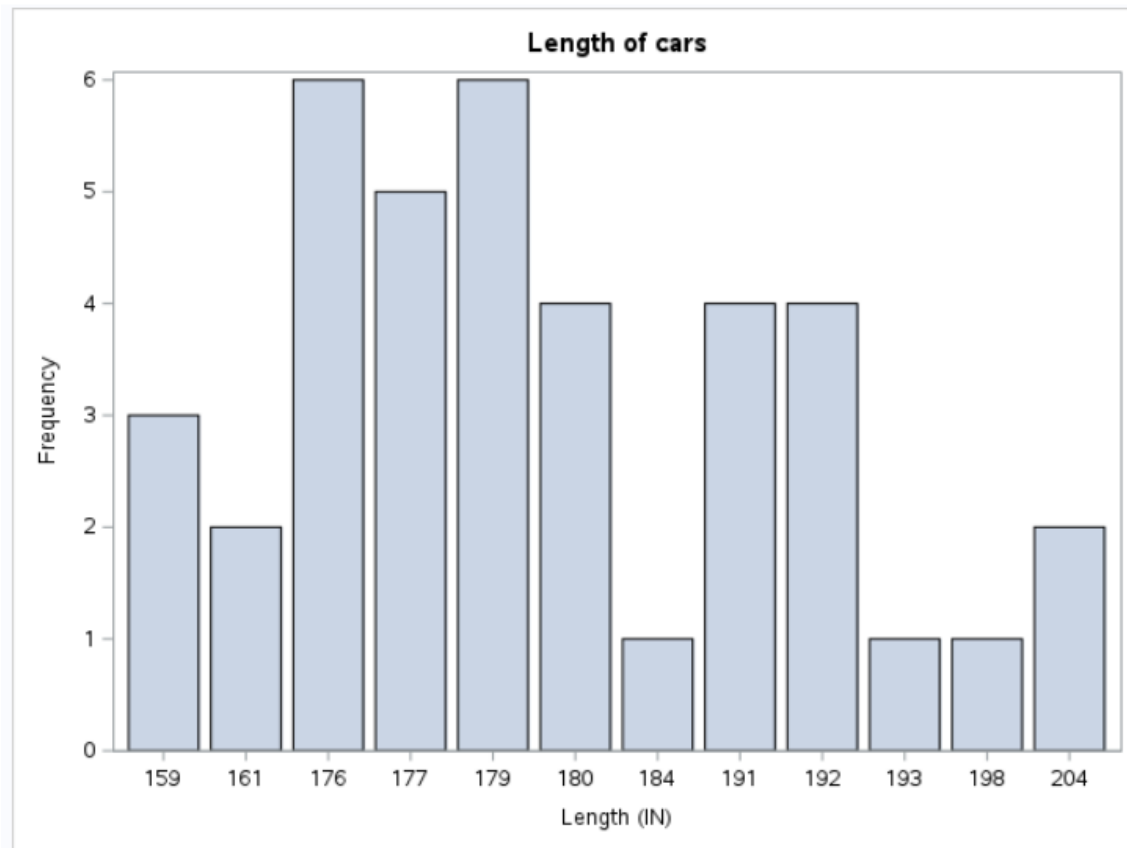
1 /* bar chart */
2 proc sql;
3 create table cars1 as
4 select * from sashelp.cars
5 where make in ("Audi","BMW")
6 ;
7 run;

```

normal bar chart code

```
proc SGPLOT data = work.cars1;  
vbar length;  
title "Length of cars";  
run;  
quit;
```

output :



▼ stacked bar chart

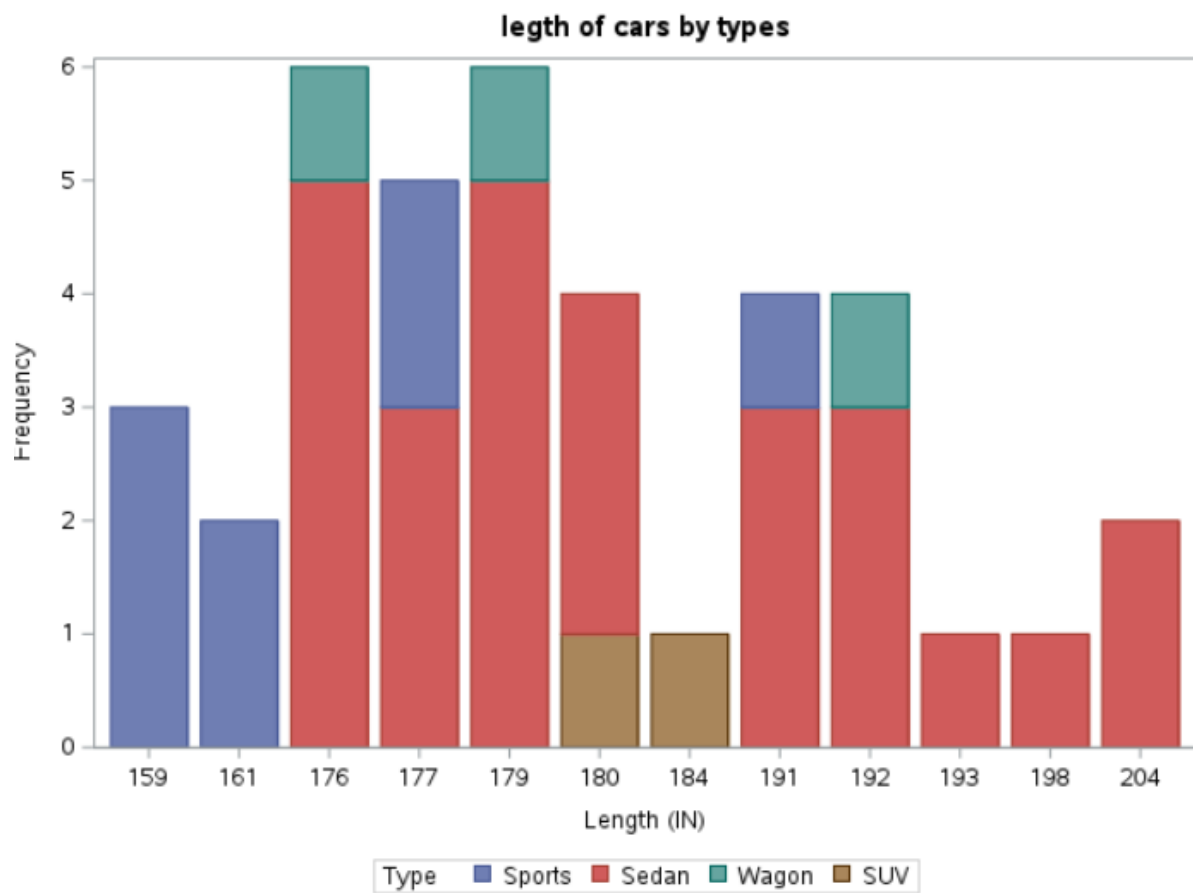
code:


```

/* stack bar chart */
proc SGPLOT data = work.cars1;
vbar length /group = type groupdisplay=STACK;
title "legth of cars by types";
run;
quit;

```

output:



▼ cluster bar chart

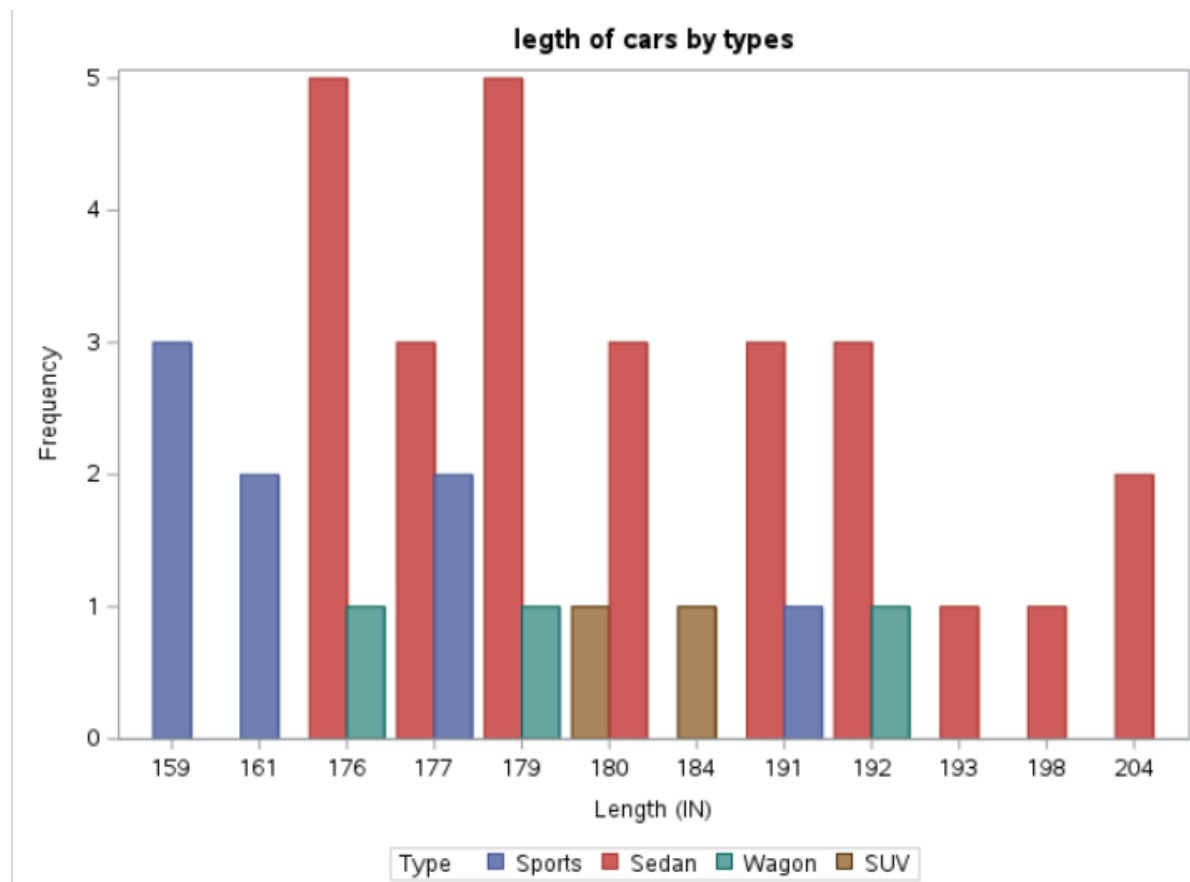
code :

```

/* cluster bar chart */
proc SGPLOT data = work.cars1;
vbar length /group = type groupdisplay=CLUSTER;
title "legth of cars by types";
run;
quit;

```

output:



SAS - GCHART

▼ code:

create dataset

```

data mpoint;
input team $ 1-9 points month conf $;
cards;
Chicago 17 1 W
Chicago 18 2 W
Chicago 22 3 W
Chicago 21 4 W
Chicago 9 5 W
Chicago 14 6 W
Chicago 11 7 W
Pittsburg 12 1 E
Pittsburg 19 2 E
Pittsburg 11 3 E
Pittsburg 11 4 E
Pittsburg 15 5 E
Pittsburg 22 6 E
Pittsburg 9 7 E
Detroit 21 1 W
Detroit 15 2 W
Detroit 25 3 W
Detroit 19 4 W
Detroit 10 5 W
Detroit 21 6 W
Detroit 4 7 W
Anaheim 21 1 W
Anaheim 23 2 W
Anaheim 18 3 W
Anaheim 10 4 W
Anaheim 12 5 W
Anaheim 22 6 W
Anaheim 4 7 W
;
run;

```

create gchart for the above dataset;

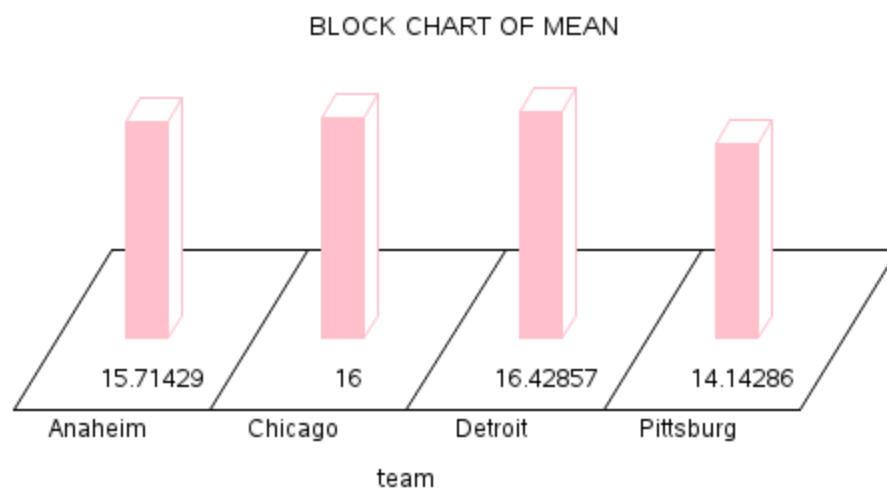
```

proc gchart data=mpoint ;
block team / type = mean sumvar=points;
HBAR team / discrete sumvar=points group=conf type=percent percent subgroup=month;
pie team / sumvar=points;
star team / sumvar=points;
pattern1 color=pink value=solid;
pattern2 color=blue value=solid;
pattern3 color=orange value=solid;
pattern4 color=lime value=solid;
pattern5 color=cyan value=solid;
pattern6 color=yellow value=solid;
pattern7 color=grey value=solid;
title1 color=blue box=2 color=red
'Teams points per month dependency ';
run;
quit;

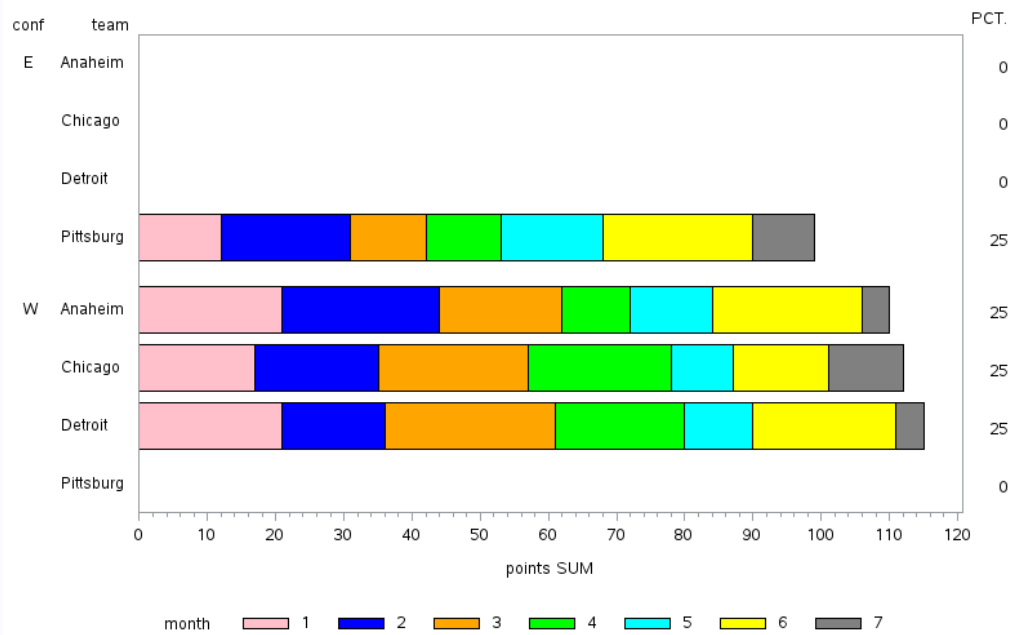
```

output:

Teams points per month dependency

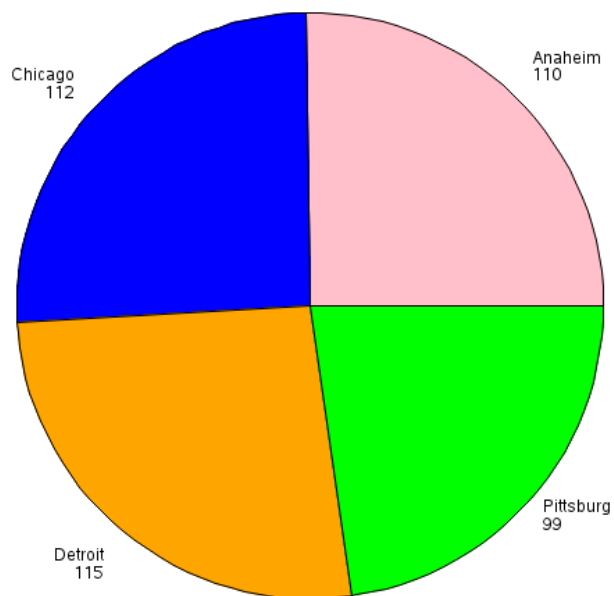


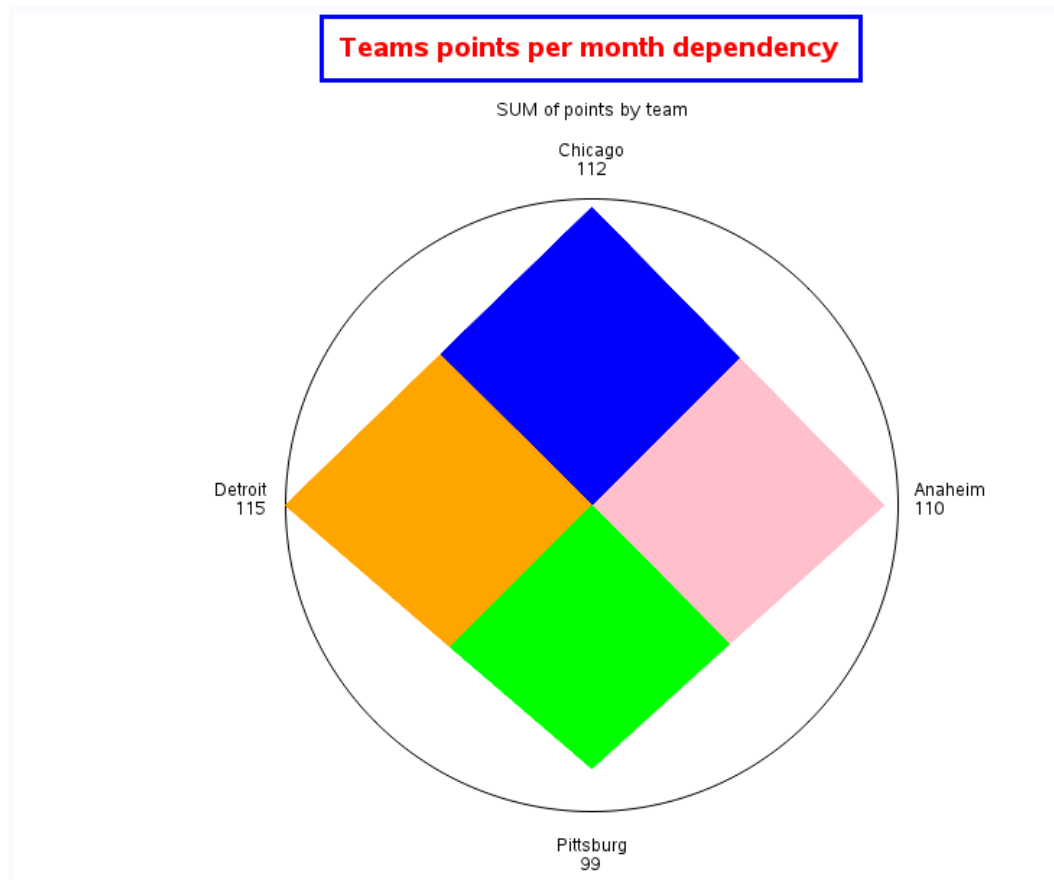
Teams points per month dependency



Teams points per month dependency

SUM of points by team





SAS - GGRAPH

▼ code:

create the same data as Gchart:

```

data nhl;
input team $ 1-9 points month;
cards;
Chicago    17 1
Chicago    35 2
Chicago    57 3
Chicago    78 4
Chicago    87 5
Chicago   101 6
Chicago   112 7
Pittsburg  12 1
Pittsburg  31 2
Pittsburg  42 3
Pittsburg  53 4
Pittsburg  68 5
Pittsburg  90 6
Pittsburg  99 7
Detroit      21 1
Detroit      36 2
Detroit      61 3
Detroit      80 4
Detroit      90 5
Detroit     111 6
Detroit     115 7
Anaheim     21 1
Anaheim     44 2
Anaheim     62 3
Anaheim     72 4
Anaheim     84 5
Anaheim    106 6
Anaheim    110 7
;
run;

```

▼ plot - 1

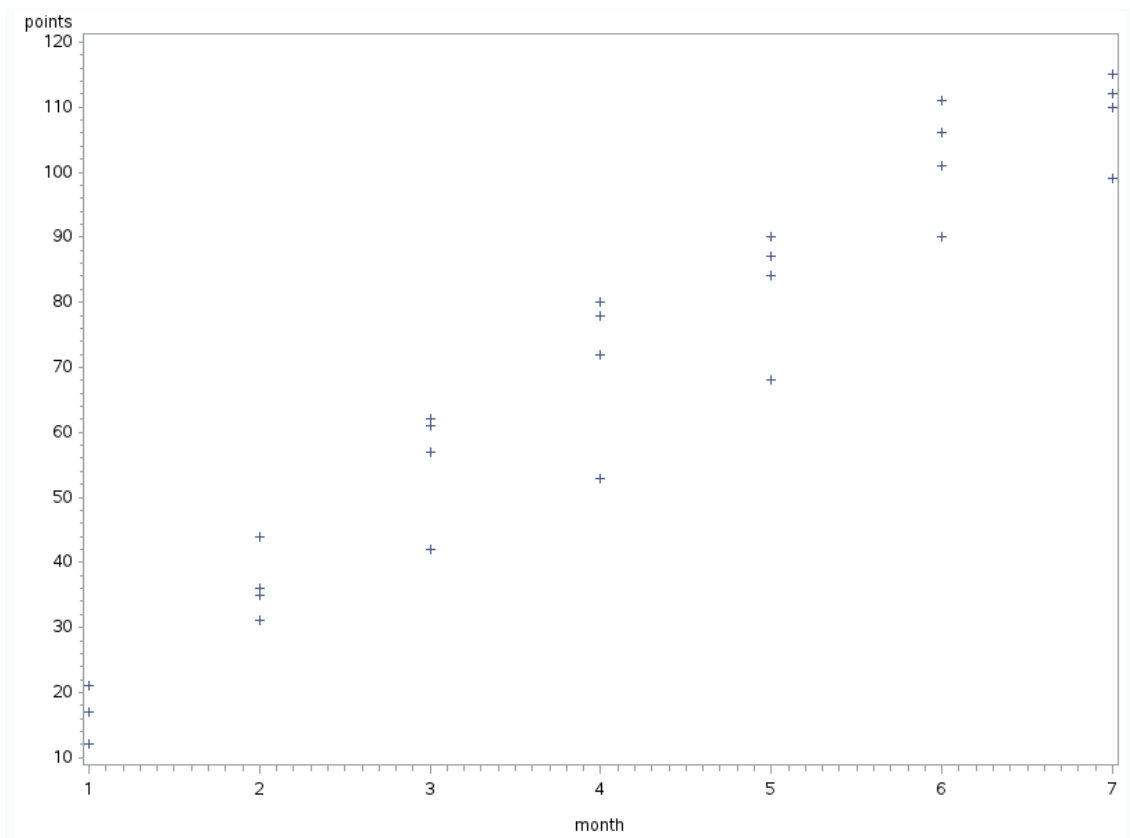
input :

```

.....
proc gplot data=nhl;
plot points*month;
run;
quit;

```

output:

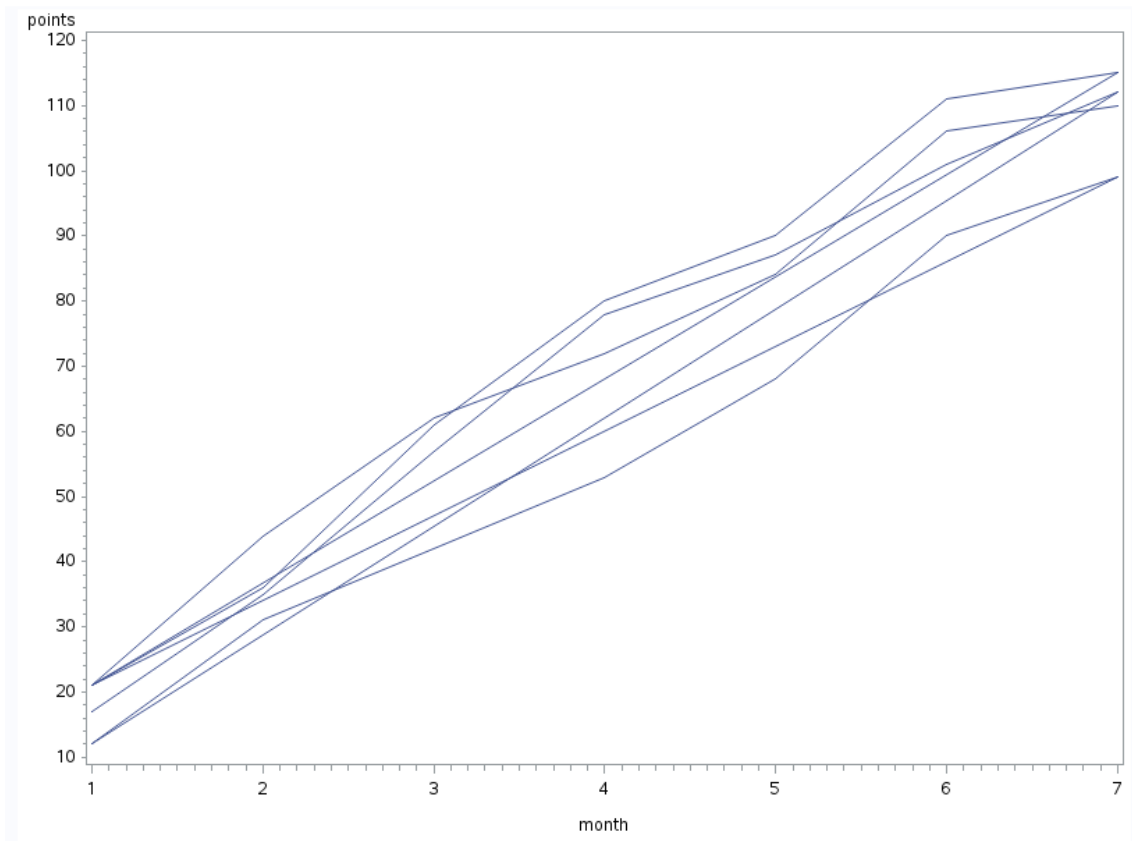


▼ plot - 2

input :

```
proc gplot data=nhl;  
plot points*month;  
symbol i=join;  
run;  
quit;
```

output:

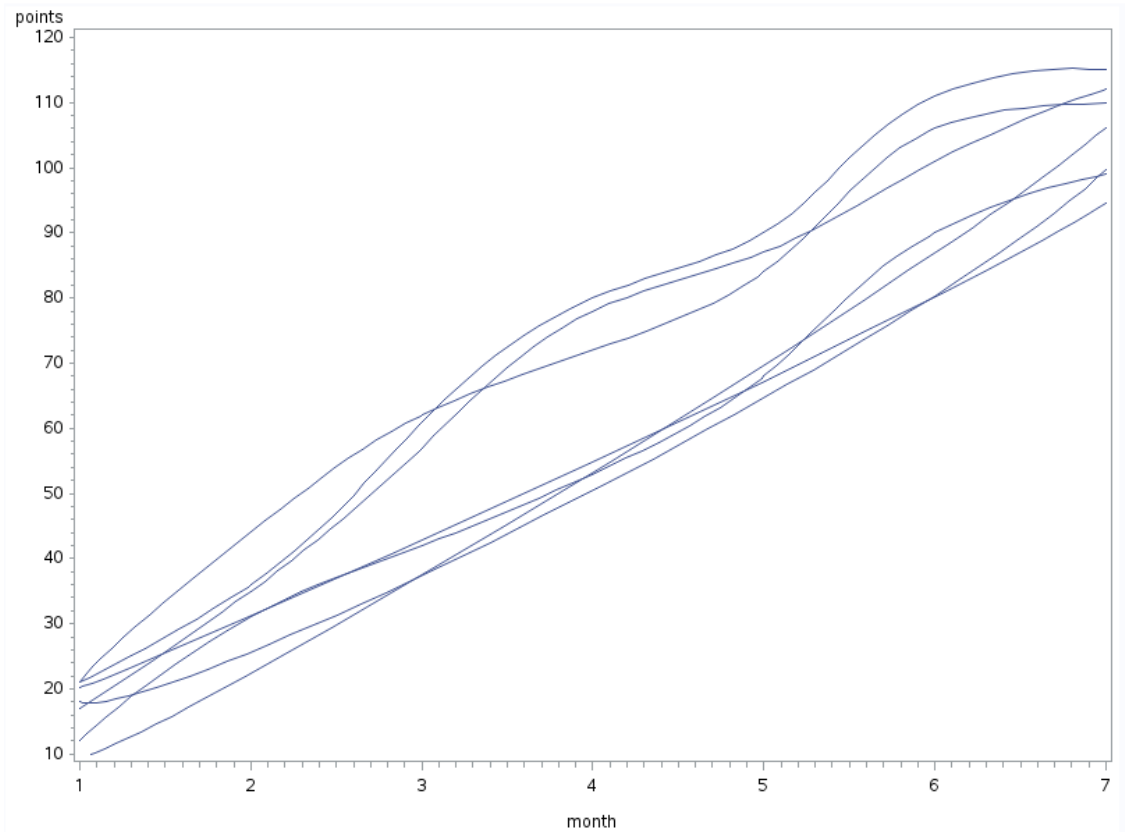


▼ plot - 3

input:

```
proc gplot data=nhl;  
plot points*month;  
symbol i=spline;  
run;  
quit;
```

output:

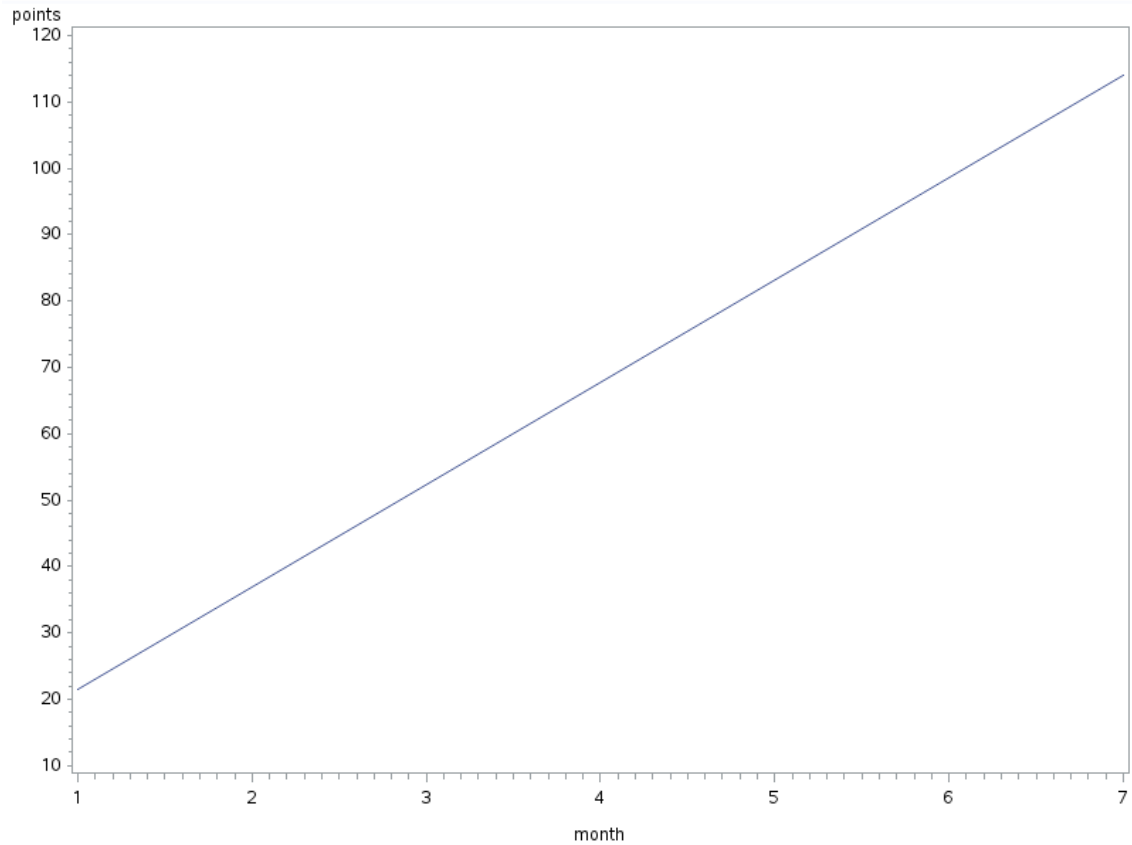


▼ plot - 4

input:

```
proc gplot data=nhl;  
plot points*month;  
symbol i=r;  
run;  
quit;
```

output:

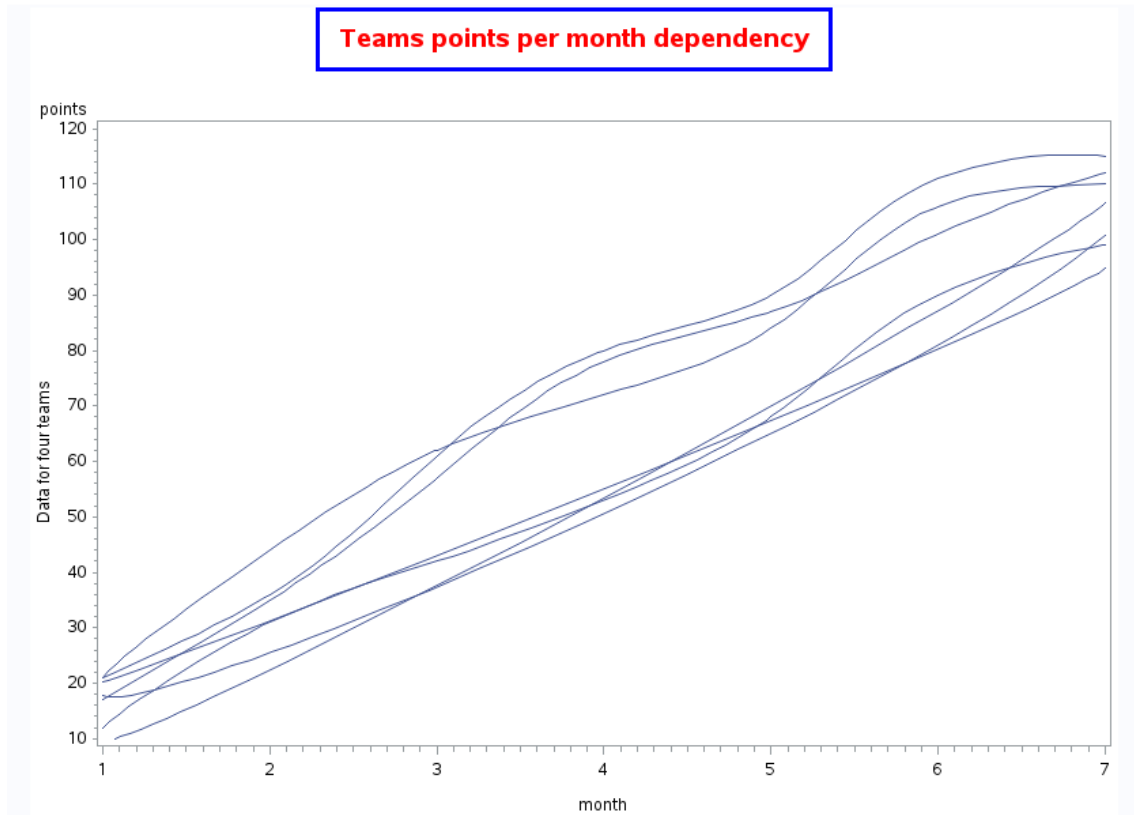


▼ plot -5

input:

```
proc gplot data=nhl;  
plot points*month;  
symbol i=spline;  
title color=blue box=2 color=red  
  ' Teams points per month dependency ';  
note angle=90 justify=center 'Data for four teams';  
run;  
quit;
```

output:

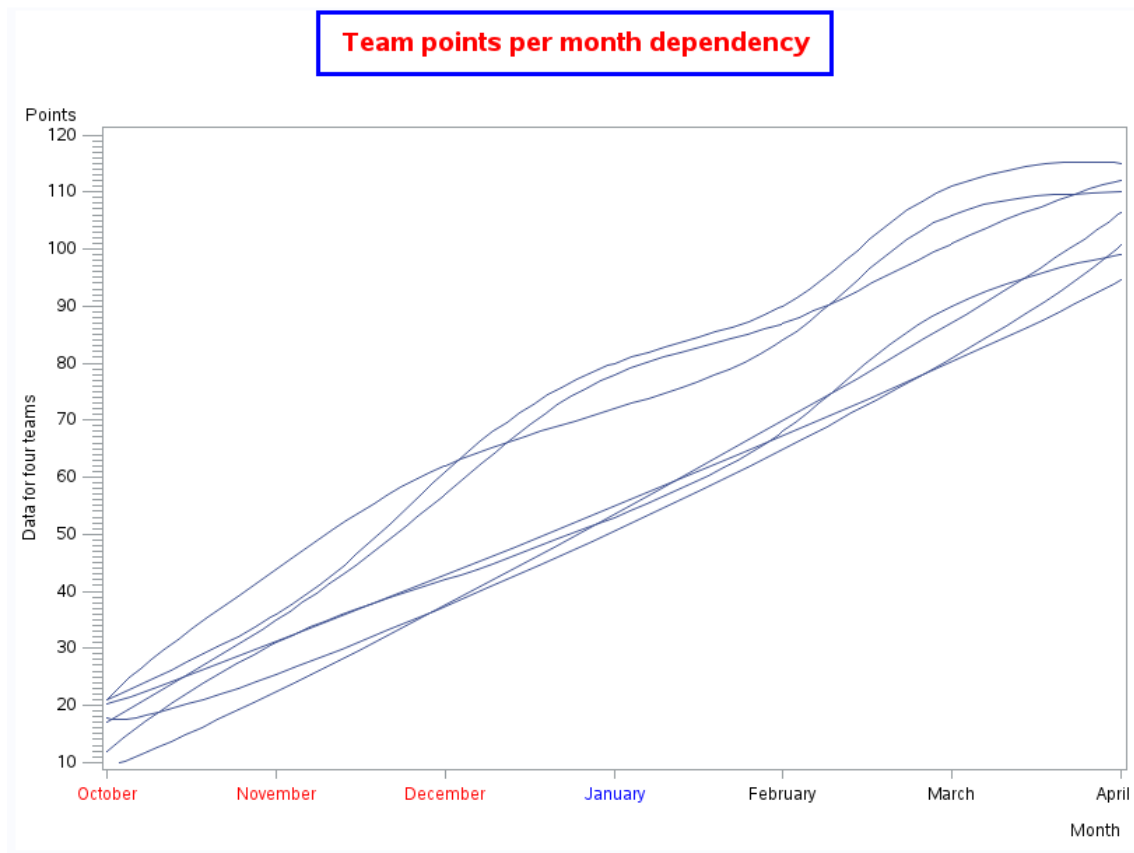


▼ plot - 6

input:

```
proc gplot data=nhl;
plot points*month / vaxis=axis1 haxis=axis2;
symbol i=spline;
title color=blue box=2 color=red
' Team points per month dependency ';
note angle=90 justify=center 'Data for four teams';
axis1 minor=(n=9 h=1) major=(h=2) label=('Points');
axis2 minor=none major=(h=0.5) label=(justify=right 'Month')
value=(color=red 'October' 'November' 'December' color=blue 'January'
color=black 'February' 'March' 'April');
run;
quit;
```

output:

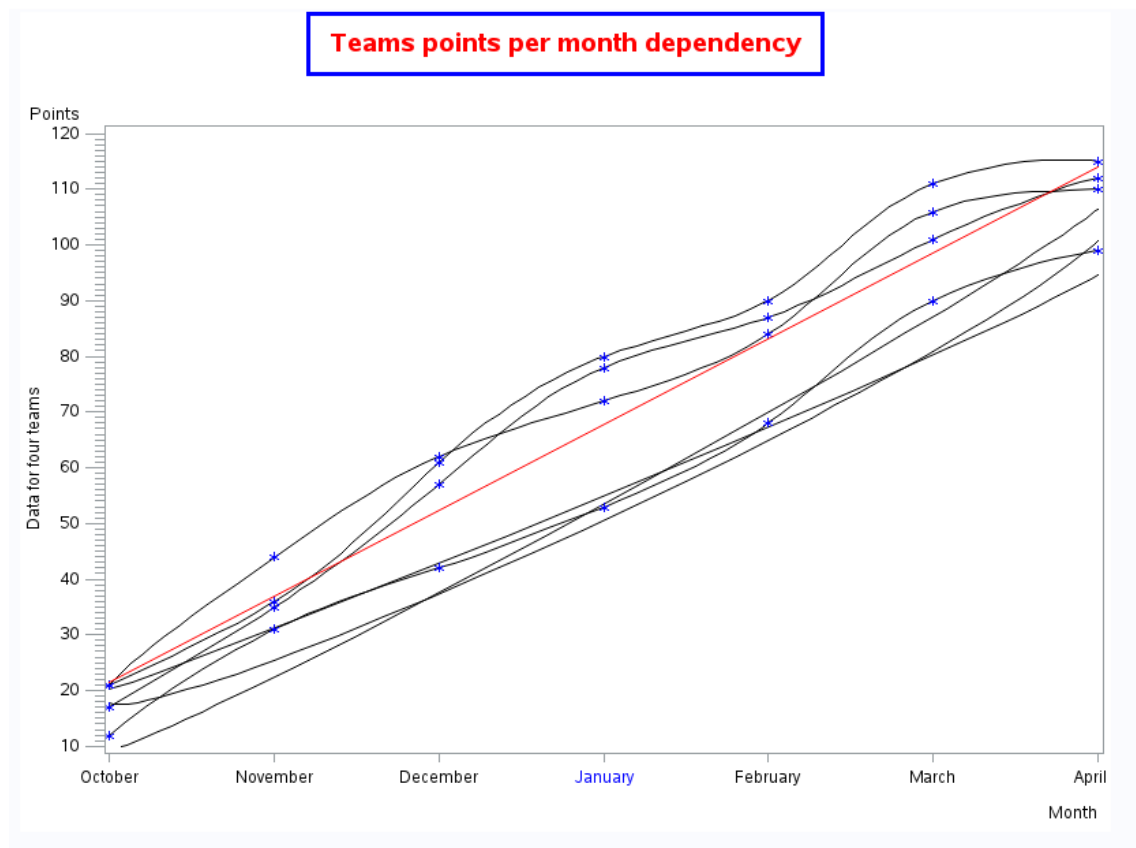


▼ plot - 7

input:

```
/* with points on plotted line */
proc gplot data=nhl;
plot points*month=1 points*month=2 points*month=3 / overlay
vaxis=axis1 haxis=axis2;
symbol1 i=spline color=black;
symbol2 i=none color=blue value=star;
symbol3 i=r color=red;
title1 color=blue box=2 color=red
' Teams points per month dependency ';
note angle=90 justify=center 'Data for four teams';
axis1 minor=(n=9 h=1) major=(h=2) label=('Points');
axis2 minor=none major=(h=0.5) label=(justify=right 'Month')
value=('October' 'November' 'December' color=blue 'January'
color=black 'February' 'March' 'April');
run;
quit;
```

output:



▼ plot - 8

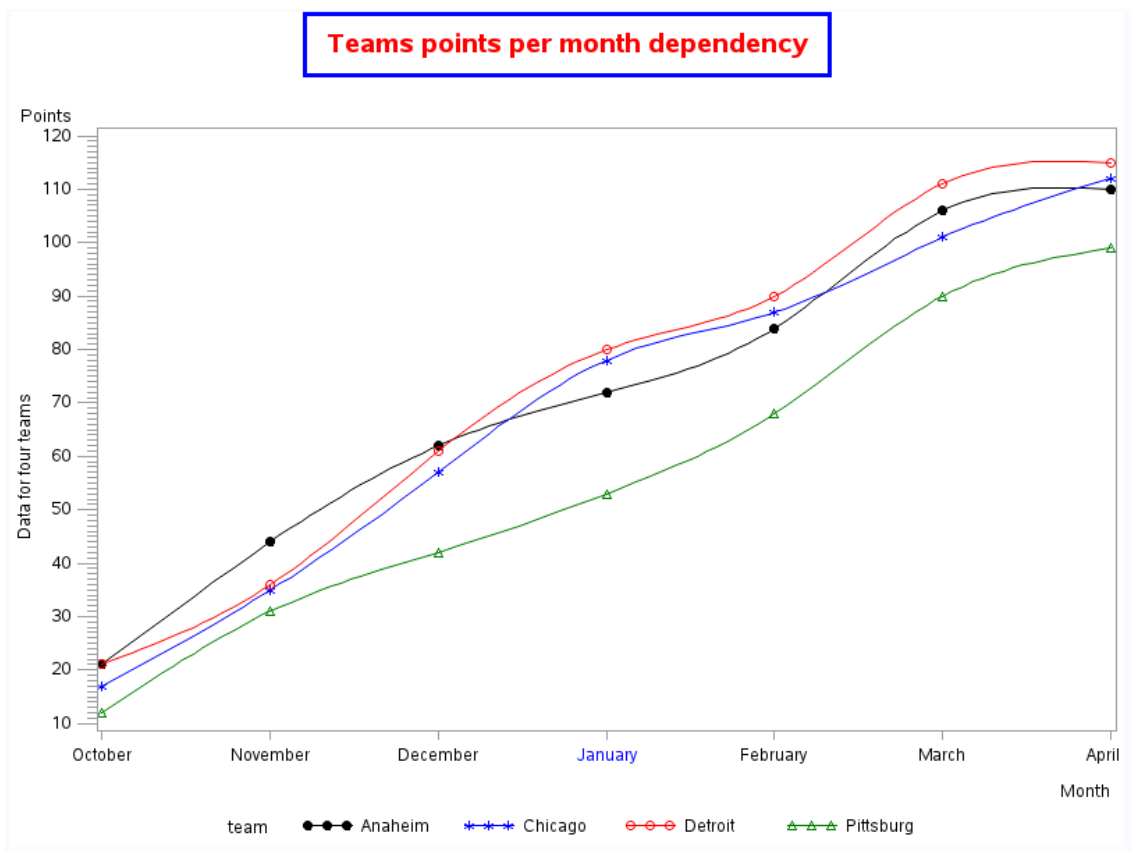
input :

```

/* plot with team */
proc gplot data=nhl;
plot points*month=team / vaxis=axis1 haxis=axis2;
symbol1 i=spline color=black value=dot;
symbol2 i=spline color=blue value=star;
symbol3 i=spline color=red value=circle;
symbol4 i=spline color=green value=triangle;
title1 color=blue box=2 color=red
' Teams points per month dependency ';
note angle=90 justify=center 'Data for four teams';
axis1 minor=(n=9 h=1) major=(h=2) label=('Points');
axis2 minor=none major=(h=0.5) label=(justify=right 'Month')
value=('October' 'November' 'December' color=blue 'January'
color=black 'February' 'March' 'April');
run;
quit;

```

output :



SAS - HISTOGRAM

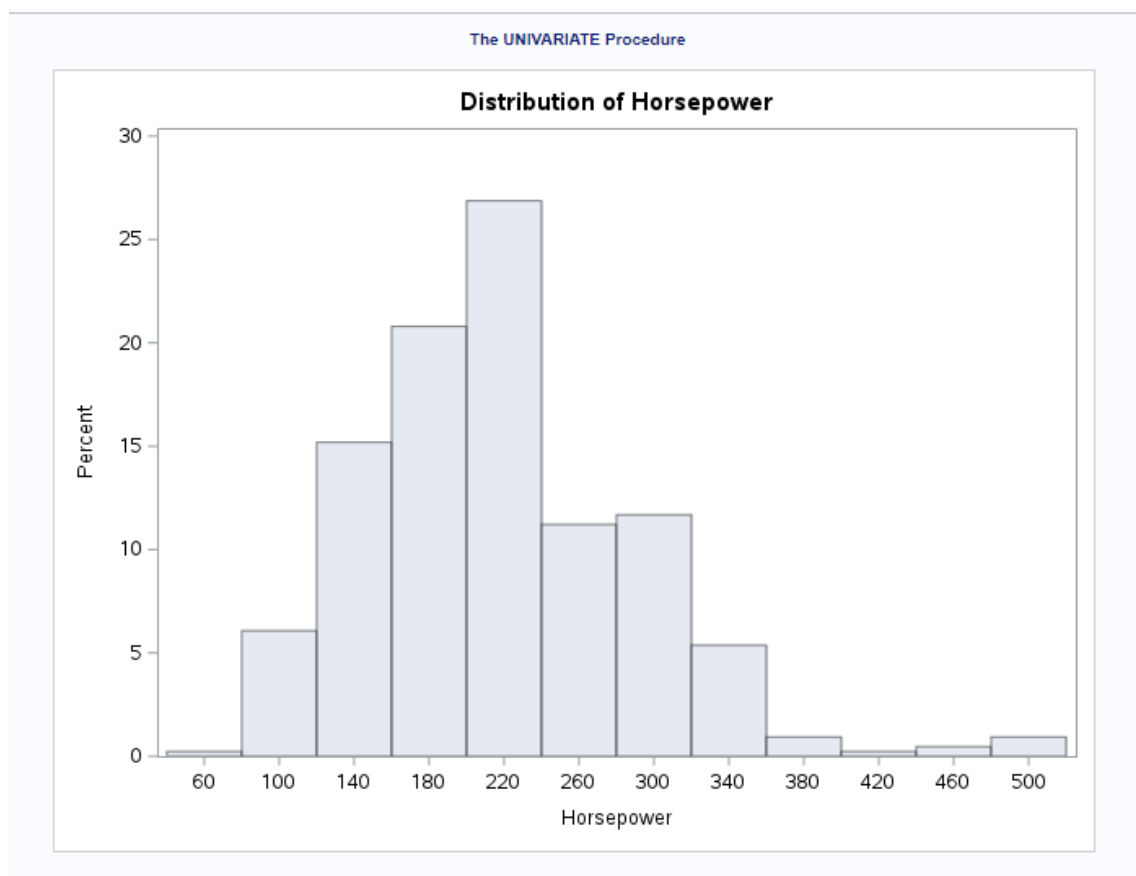
▼ code:

▼ plot - 1

input :

```
/* plotting a histogram */  
  
PROC UNIVARIATE data = sashelp.cars;  
  histogram horsepower;  
  /* midpoints = 176 to 350 by 50 */  
run;
```

output :

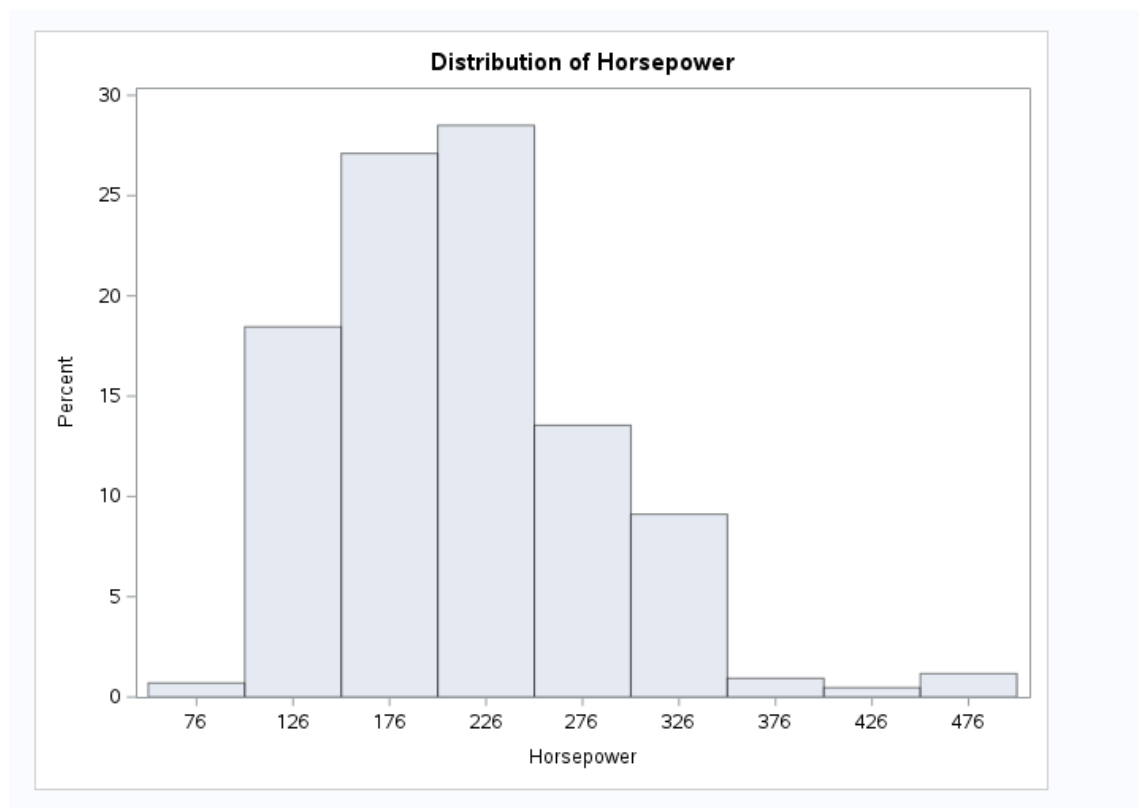


▼ plot - 2

input:

```
/* define the midpoints for it */  
PROC UNIVARIATE data = sashelp.cars;  
  histogram horsepower  
    /midpoints = 176 to 350 by 50;  
run;  
  
PROC UNIVARIATE data = sashelp.cars;  
  histogram cylinders;  
run;
```

output:

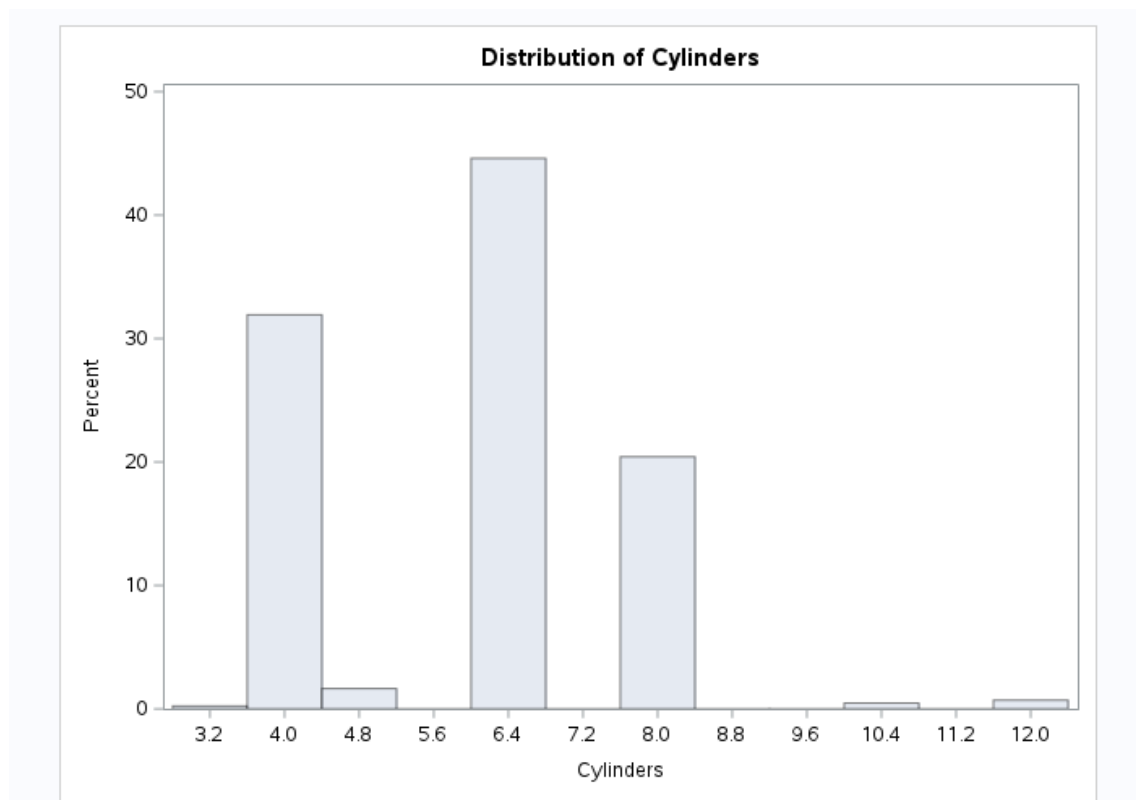


▼ plot - 3

input:

```
PROC UNIVARIATE data = sashelp.cars;  
  histogram cylinders;  
run;
```

output:



▼ plot - 4

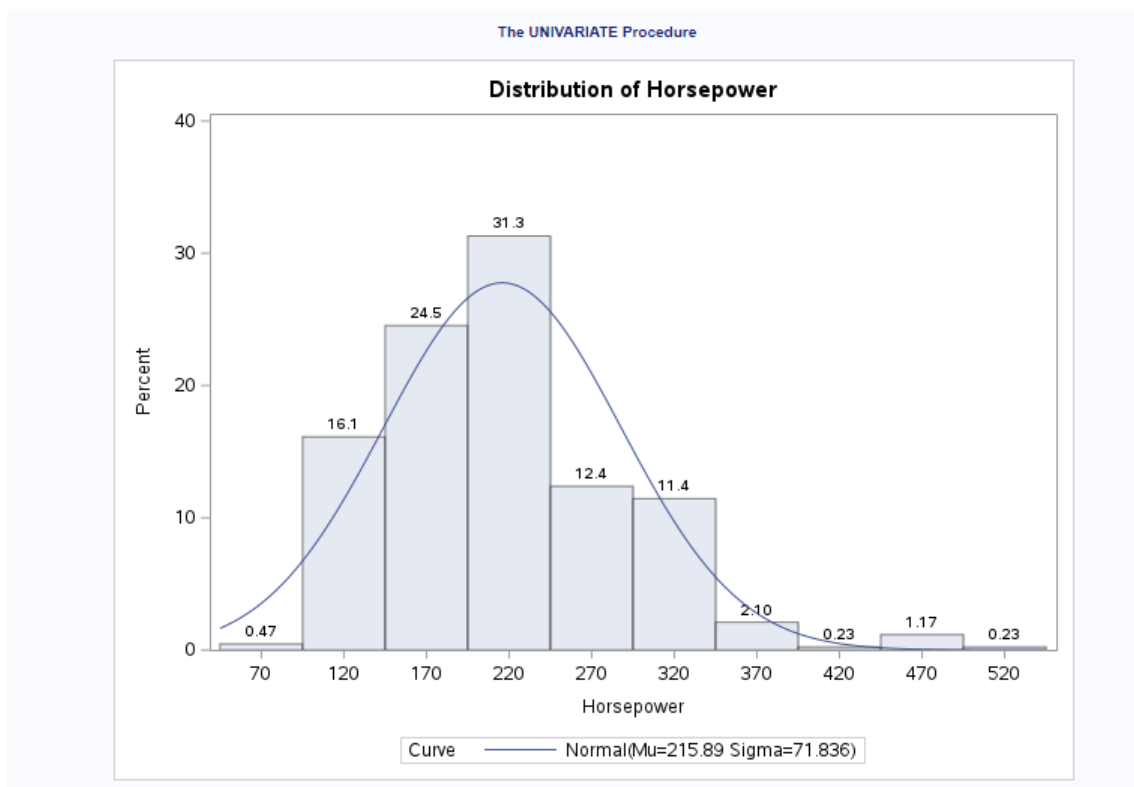
input:

```

/* random mu sigma values */
proc univariate data=sashelp.cars;
  histogram horsepower
  /
  normal(
    mu = est
    sigma = est
    color = blue
    w = 2.5
  )
  barlable = percent
  midpoint = 70 to 550 by 50;
run;

```

output:



SAS - PIE CHART

▼ create template and data

input:

```

/* pie charts */
proc template;
define statgraph pie;
begingraph;
layout region;
piechart category = type /
datalabellocation = outside
categorydirection = clockwise
start=180 name = 'pie';
discretelegend 'pie' /
title = 'cars type';
endlayout;
endgraph;
end;

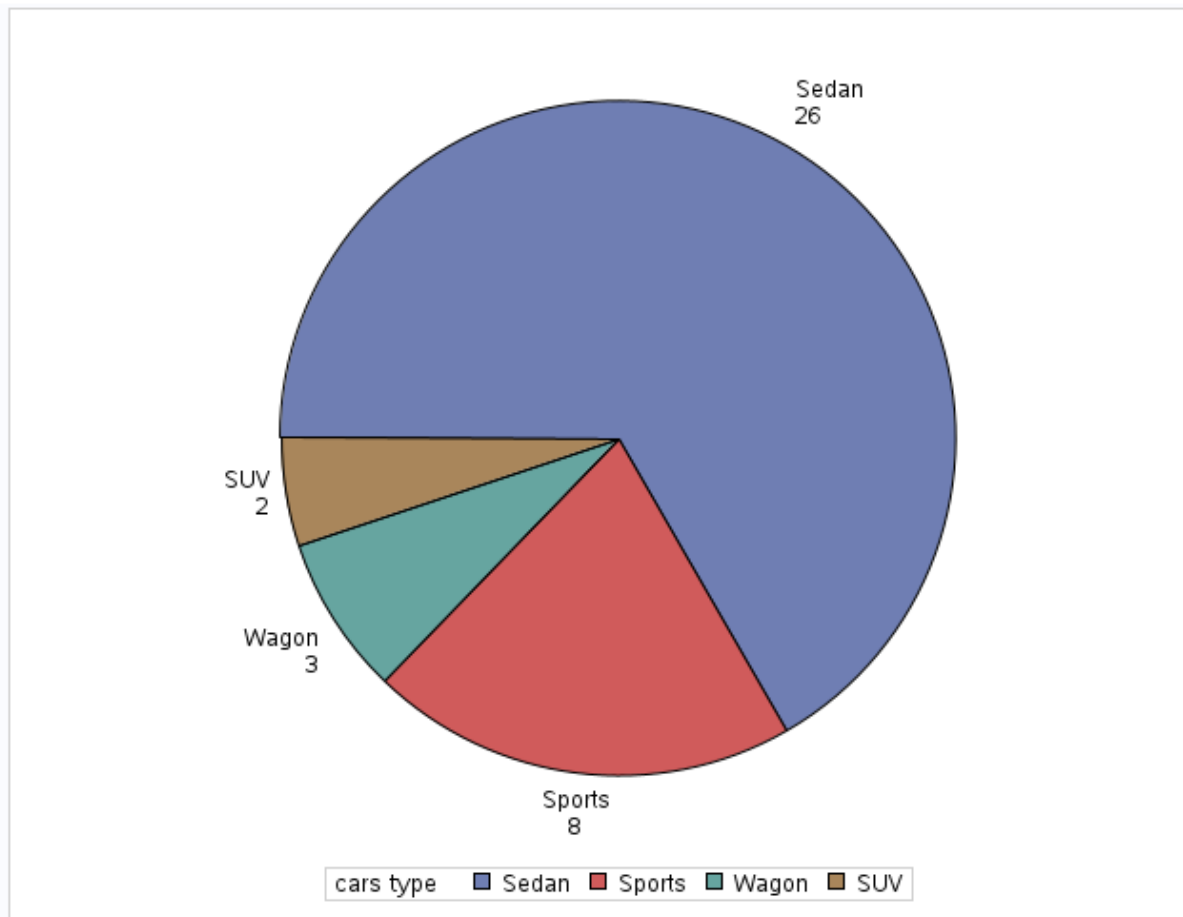
proc sql;
create table cars1 as
select * from sashelp.cars
where make in ("Audi","BMW")
;
run;

```

▼ pie chart

input:

output:



SAS - SQL

▼ normal data

input:

data:

```
data temp;
input ID $ Name $ Salary Dept $;
datalines;
1 Aman 673.5 FIN
2 Sameer 856.3 IT
3 Riddhi 756.6 OPS
4 Anu 653.4 FIN
5 Arpit 453.6 IT
6 Priyanshu 345.2 FIN
;
run;
```

output:

Obs	ID	Name	Salary	Dept
1	1	Aman	673.5	FIN
2	2	Sameer	856.3	IT
3	3	Riddhi	756.6	OPS
4	4	Anu	653.4	FIN
5	5	Arpit	453.6	IT
6	6	Priyansh	345.2	FIN

query 1 :

```
/* all values */  
proc sql;  
create table Emp as  
select * from temp;  
quit;  
proc print data = Emp;  
run;
```

output:

Obs	ID	Name	Salary	Dept
1	1	Aman	673.5	FIN
2	2	Sameer	856.3	IT
3	3	Riddhi	756.6	OPS
4	4	Anu	653.4	FIN
5	5	Arpit	453.6	IT
6	6	Priyansh	345.2	FIN

query 2 :

```

/* sql connect */
proc sql;
create table Emp as
select Name, Salary from temp;
quit;
proc print data = Emp;
run;

```

output:

Obs	Name	Salary
1	Aman	673.5
2	Sameer	856.3
3	Riddhi	756.6
4	Anu	653.4
5	Arpit	453.6
6	Priyansh	345.2

▼ sas data

read data:

```

/* read operation */
proc sql;
select * from sashelp.cars;
quit;

```

output data:

Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	Engine Size (L)	Cylinders	Horsepower	MPG (City)	MPG (Highway)	Weight (LBS)	Wheelbase (IN)	Length (IN)
Acura	MDX	SUV	Asia	All	\$36,945	\$33,337	3.5	6	265	17	23	4451	106	189
Acura	RSX Type S 2dr	Sedan	Asia	Front	\$23,820	\$21,761	2	4	200	24	31	2778	101	172
Acura	TSX 4dr	Sedan	Asia	Front	\$26,690	\$24,647	2.4	4	200	22	29	3230	105	183
Acura	TL 4dr	Sedan	Asia	Front	\$33,195	\$30,299	3.2	6	270	20	28	3575	108	186
Acura	3.5 RL 4dr	Sedan	Asia	Front	\$43,755	\$39,014	3.5	6	225	18	24	3880	115	197
Acura	3.5 RL w/Navigation 4dr	Sedan	Asia	Front	\$46,100	\$41,100	3.5	6	225	18	24	3893	115	197
Acura	NSX coupe 2dr manual S	Sports	Asia	Rear	\$69,785	\$79,978	3.2	6	290	17	24	3153	100	174
Audi	A4 1.8T 4dr	Sedan	Europe	Front	\$25,940	\$23,508	1.8	4	170	22	31	3252	104	179
Audi	A4 1.8T convertible 2dr	Sedan	Europe	Front	\$35,940	\$32,508	1.8	4	170	23	30	3638	105	180
Audi	A4 3.0 4dr	Sedan	Europe	Front	\$31,840	\$28,846	3	6	220	20	28	3462	104	179
Audi	A4 3.0 Quattro 4dr manual	Sedan	Europe	All	\$33,430	\$30,366	3	6	220	17	26	3583	104	179
Audi	A4 3.0 Quattro 4dr auto	Sedan	Europe	All	\$34,480	\$31,388	3	6	220	18	25	3627	104	179
Audi	A6 3.0 4dr	Sedan	Europe	Front	\$36,640	\$33,129	3	6	220	20	27	3561	109	192
Audi	A6 3.0 Quattro 4dr	Sedan	Europe	All	\$39,640	\$35,992	3	6	220	18	25	3880	109	192
Audi	A4 3.0 convertible 2dr	Sedan	Europe	Front	\$42,490	\$38,325	3	6	220	20	27	3814	105	180
Audi	A4 3.0 Quattro convertible 2dr	Sedan	Europe	All	\$44,240	\$40,075	3	6	220	18	25	4013	105	180
Audi	A6 2.7 Turbo Quattro 4dr	Sedan	Europe	All	\$42,840	\$38,840	2.7	6	250	18	25	3836	109	192
Audi	A6 4.2 Quattro 4dr	Sedan	Europe	All	\$49,690	\$44,936	4.2	8	300	17	24	4024	109	193
Audi	A8 L Quattro 4dr	Sedan	Europe	All	\$69,190	\$64,740	4.2	8	330	17	24	4399	121	204
Audi	S4 Quattro 4dr	Sedan	Europe	All	\$46,040	\$43,556	4.2	8	340	14	20	3825	104	179
Audi	RS 6 4dr	Sports	Europe	Front	\$84,600	\$76,417	4.2	8	450	15	22	4024	109	191
Audi	TT 1.8 convertible 2dr (coupe)	Sports	Europe	Front	\$35,940	\$32,512	1.8	4	180	20	28	3131	95	159
Audi	TT 1.8 Quattro 2dr (convertible)	Sports	Europe	All	\$37,390	\$33,891	1.8	4	225	20	28	2921	96	159
Audi	TT 3.2 coupe 2dr (convertible)	Sports	Europe	All	\$40,590	\$36,739	3.2	6	250	21	29	3351	96	159
Audi	A6 3.0 Avant Quattro	Wagon	Europe	All	\$40,840	\$37,060	3	6	220	18	25	4035	109	192
Audi	S4 Avant Quattro	Wagon	Europe	All	\$49,090	\$44,446	4.2	8	340	15	21	3936	104	179
BMW	X3 3.0i	SUV	Europe	All	\$37,000	\$33,873	3	6	225	16	23	4023	110	180
BMW	X5 4.4i	SUV	Europe	All	\$52,195	\$47,720	4.4	8	325	16	22	4824	111	184
BMW	325i 4dr	Sedan	Europe	Rear	\$28,495	\$26,155	2.5	6	184	20	29	3219	107	176
BMW	325Ci 2dr	Sedan	Europe	Rear	\$30,795	\$28,245	2.5	6	184	20	29	3197	107	177
BMW	325Ci convertible 2dr	Sedan	Europe	Rear	\$37,995	\$34,800	2.5	6	184	19	27	3560	107	177
BMW	325xi 4dr	Sedan	Europe	All	\$30,245	\$27,745	2.5	6	184	19	27	3461	107	176
BMW	330i 4dr	Sedan	Europe	Rear	\$35,495	\$32,525	3	6	225	20	30	3285	107	176

query:

```
/* where in table */
proc sql;
select * from sashelp.cars
where type = "Sedan"
and origin = "Asia";
quit;
```

output:

Make	Model	Type	Origin	DriveTrain	MSRP	Invoice	Engine Size (L)	Cylinders	Horsepower	MPG (City)	MPG (Highway)	Weight (LBS)	Wheelbase (IN)	Length (IN)
Acura	RSX Type S 2dr	Sedan	Asia	Front	\$23,620	\$21,761	2	4	200	24	31	2778	101	172
Acura	TSX 4dr	Sedan	Asia	Front	\$26,990	\$24,647	2.4	4	200	22	29	3230	105	183
Acura	TL 4dr	Sedan	Asia	Front	\$33,195	\$30,299	3.2	6	270	20	28	3575	108	188
Acura	3.5 RL 4dr	Sedan	Asia	Front	\$43,755	\$39,014	3.5	6	225	18	24	3880	115	197
Acura	3.5 RL w/Navigation 4dr	Sedan	Asia	Front	\$46,100	\$41,100	3.5	6	225	18	24	3893	115	197
Honda	Civic DX 2dr	Sedan	Asia	Front	\$13,270	\$12,175	1.7	4	115	32	38	2432	103	175
Honda	Civic HX 2dr	Sedan	Asia	Front	\$14,170	\$12,996	1.7	4	117	36	44	2500	103	175
Honda	Civic LX 4dr	Sedan	Asia	Front	\$15,850	\$14,531	1.7	4	115	32	38	2513	103	175
Honda	Accord LX 2dr	Sedan	Asia	Front	\$19,880	\$17,924	2.4	4	160	26	34	2994	105	188
Honda	Accord EX 2dr	Sedan	Asia	Front	\$22,280	\$20,080	2.4	4	160	26	34	3047	105	188
Honda	Civic EX 4dr	Sedan	Asia	Front	\$17,750	\$16,265	1.7	4	127	32	37	2801	103	175
Honda	Civic Si 2dr hatch	Sedan	Asia	Front	\$19,490	\$17,849	2	4	160	26	30	2782	101	166
Honda	Accord LX V6 4dr	Sedan	Asia	Front	\$23,760	\$21,428	3	6	240	21	30	3349	108	190
Honda	Accord EX V6 2dr	Sedan	Asia	Front	\$26,960	\$24,304	3	6	240	21	30	3264	105	188
Honda	Odyssey LX	Sedan	Asia	Front	\$24,950	\$22,498	3.5	6	240	18	25	4310	118	201
Honda	Odyssey EX	Sedan	Asia	Front	\$27,450	\$24,744	3.5	6	240	18	25	4365	118	201
Hyundai	Accent 2dr hatch	Sedan	Asia	Front	\$10,539	\$10,107	1.6	4	103	29	33	2255	96	167
Hyundai	Accent GL 4dr	Sedan	Asia	Front	\$11,639	\$11,116	1.6	4	103	29	33	2260	96	167
Hyundai	Accent GT 2dr hatch	Sedan	Asia	Front	\$11,939	\$11,209	1.6	4	103	29	33	2339	96	167
Hyundai	Elantra GLS 4dr	Sedan	Asia	Front	\$13,639	\$12,781	2	4	138	26	34	2635	103	178
Hyundai	Elantra GT 4dr	Sedan	Asia	Front	\$15,389	\$14,207	2	4	138	26	34	2635	103	178
Hyundai	Elantra GT 4dr hatch	Sedan	Asia	Front	\$15,389	\$14,207	2	4	138	26	34	2698	103	178
Hyundai	Sonata GLS 4dr	Sedan	Asia	Front	\$19,339	\$17,574	2.7	6	170	19	27	3217	106	187
Hyundai	Sonata LX 4dr	Sedan	Asia	Front	\$20,339	\$18,380	2.7	6	170	19	27	3217	106	187
Hyundai	XG350 4dr	Sedan	Asia	Front	\$24,599	\$22,055	3.5	6	194	17	26	3651	108	192
Hyundai	XG350 L 4dr	Sedan	Asia	Front	\$26,189	\$23,486	3.5	6	194	17	26	3651	108	192
Infiniti	G35 4dr	Sedan	Asia	Rear	\$28,495	\$26,157	3.5	6	260	18	26	3336	112	187
Infiniti	G35 Sport Coupe 2dr	Sedan	Asia	Rear	\$29,795	\$27,536	3.5	6	260	18	26	3416	112	182
Infiniti	G35 4dr	Sedan	Asia	All	\$32,445	\$29,783	3.5	6	260	18	26	3677	112	187
Infiniti	I35 4dr	Sedan	Asia	Front	\$31,145	\$28,320	3.5	6	255	19	26	3306	108	194
Infiniti	M45 4dr	Sedan	Asia	Rear	\$42,845	\$38,792	4.5	8	340	17	23	3851	110	197
Infiniti	Q45 Luxury 4dr	Sedan	Asia	Rear	\$52,545	\$47,575	4.5	8	340	17	23	3977	113	200
Kia	Optima LX 4dr	Sedan	Asia	Front	\$16,040	\$14,910	2.4	4	138	23	30	3281	106	188
Kia	Rio 4dr manual	Sedan	Asia	Front	\$10,280	\$9,875	1.6	4	104	26	33	2403	95	167