Stat 462: Lab 2 Solutions

```
* Question 1.2;
data two;
input score @@; /*input one numeric variable "score".
                        @@ holds the input line for processing
                        additional variables. */
if score > 50 then do; /*if the score variable is larger
                        than 50 when read in, do: */
result = 'P';  /* create a new variable "result" with value 'P' */
addon = (score - 50)*10; /* create a new variable "addon" with computed value
*/
end;
else if score < 50 then do; /*if the score variable is smaller
                        than 50 when read in, do: */
result = 'F'; /* create a new variable "result" with value 'F' */
end;
datalines;
47 49 50 52 55 .
;
run;
proc print data=two; run;
/* Notes:
1. When score < 50, "addon" is empty
2. When the value of score is exactly 50, neither do
   block is evaluated. "result" is blank (missing)
   and "addon" is empty.
3. When score is missing, it evaluates as < 50.
   This is because a missing numeric value is
   smaller than any other numeric value in SAS.
*/
```

```
Obs  score  result  addon

  1     47  F           .
  2     49  F           .
  3     50              .
  4     52  P          20
  5     55  P          50
  6      .  F           .
```

```
* Question 1.6;
data distance;
input miles 5.2;
datalines;
3
34
345
3451
34512
3.
3.4
34.
345.1
;
proc print;
run;
```

```
Obs   miles

  1    0.03
  2    0.34
  3    3.45
  4   34.51
  5  345.12
  6    3.00
  7    3.40
  8   34.00
  9  345.10
```

```
* Question 1.9;
data result_a;
input type c1 c2 @;
index = (2*c1) + c2;
datalines;
5 0 2
7 3 1
. 0 0
;
proc print;
run;
```

```
Obs  type  c1  c2  index

  1     5   0   2      2
  2     7   3   1      7
  3     .   0   0      0
```

Note that the @ line-hold specifier holds the input buffer.

```
data result_b;
input type c1 c2 @;
if type <= 6 then do;
  index = (2*c1) + c2;
  output;
end;
datalines;
5 0 2
7 3 1
. 0 0
;
proc print;
run;
```

```
Obs  type  c1  c2  index

  1     5   0   2      2
  2     .   0   0      0
```

Again, the @ holds the input buffer. After the first line is read, the do block is executed and the the line (including "index") is written to the dataset. After the second line is read the do block is not executed, and no line is output. NOTE: including the explicit output statement means that a line in the dataset is ONLY written when the output statement is encountered; the usual implicit output statement (performed at the end of a DATA step iteration) is ignored. After the third line is read the do block is executed because missing values are always < in logical statements.

```
data result_c;
input type c1 c2 @;
if type <= 6 then do;
  index = (2*c1) + c2;
end;
else delete;
datalines;
5 0 2
7 3 1
. 0 0
;
proc print;
run;
```

```
Obs  type  c1  c2  index

  1     5   0   2      2
  2     .   0   0      0
```

This one is a bit more straightforward; if the do block is not executed, then the line is deleted.

```
 data result_d;
input type c1 c2 @;
if type > 6 then delete;
datalines;
5 0 2
7 3 1
. 0 0
;
proc print;
run;
```

```
Obs  type  c1  c2
  1     5   0   2
  2     .   0   0
```

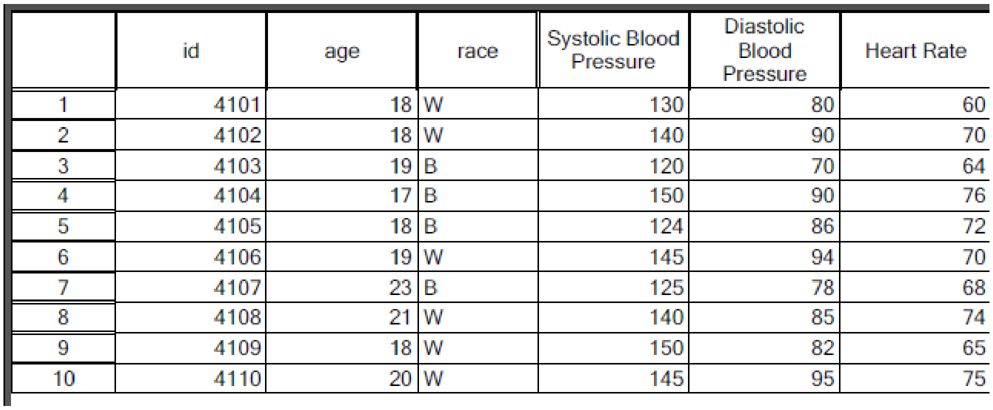
In this case, the second line is deleted because type = 7. Note that missing values are always <.

```
data result_e;
input type c1 c2 @;
 if type > 6 then delete;
index = (2*c1) + c2;
datalines;
5 0 2
7 3 1
. 0 0
;
proc print;
run;
```

```
Obs  type  c1  c2  index
  1     5   0   2      2
  2     .   0   0      0
```

In this case, the second line is deleted because type = 7. Note that missing values are always <.

```
* Question 1.12;
* parts a) b) c);
data athlete;
input id age race $ spb dbp hr;
label spb=Systolic Blood Pressure dbp=Diastolic Blood Pressure
      hr=Heart Rate;
datalines;
4101 18 W 130 80 60
4102 18 W 140 90 70
4103 19 B 120 70 64
4104 17 B 150 90 76
4105 18 B 124 86 72
4106 19 W 145 94 70
4107 23 B 125 78 68
4108 21 W 140 85 74
4109 18 W 150 82 65
4110 20 W 145 95 75
;
run; /* for b) look at work.athlete. */
proc print;
run;
```



	id	age	race	Systolic Blood Pressure	Diastolic Blood Pressure	Heart Rate
1	4101	18	W	130	80	60
2	4102	18	W	140	90	70
3	4103	19	B	120	70	64
4	4104	17	B	150	90	76
5	4105	18	B	124	86	72
6	4106	19	W	145	94	70
7	4107	23	B	125	78	68
8	4108	21	W	140	85	74
9	4109	18	W	150	82	65
10	4110	20	W	145	95	75

```
Obs    id  age  race  spb  dbp  hr

  1  4101   18  W     130   80  60
  2  4102   18  W     140   90  70
  3  4103   19  B     120   70  64
  4  4104   17  B     150   90  76
  5  4105   18  B     124   86  72
  6  4106   19  W     145   94  70
  7  4107   23  B     125   78  68
  8  4108   21  W     140   85  74
  9  4109   18  W     150   82  65
 10  4110   20  W     145   95  75
```

```
* part d);
data athlete;
input id age race $ spb dbp hr @;
abp = spb/3 + dbp*2/3;
label spb=Systolic Blood Pressure
      dbp=Diastolic Blood Pressure
      hr=Heart Rate;
datalines;
4101 18 W 130 80 60
4102 18 W 140 90 70
4103 19 B 120 70 64
4104 17 B 150 90 76
4105 18 B 124 86 72
4106 19 W 145 94 70
4107 23 B 125 78 68
4108 21 W 140 85 74
4109 18 W 150 82 65
4110 20 W 145 95 75
;
proc print;
run;
```

```
Obs    id  age  race  spb  dbp  hr      abp

  1  4101   18  W     130   80  60   96.667
  2  4102   18  W     140   90  70  106.667
  3  4103   19  B     120   70  64   86.667
  4  4104   17  B     150   90  76  110.000
  5  4105   18  B     124   86  72   98.667
  6  4106   19  W     145   94  70  111.000
  7  4107   23  B     125   78  68   93.667
  8  4108   21  W     140   85  74  103.333
  9  4109   18  W     150   82  65  104.667
 10  4110   20  W     145   95  75  111.667
```

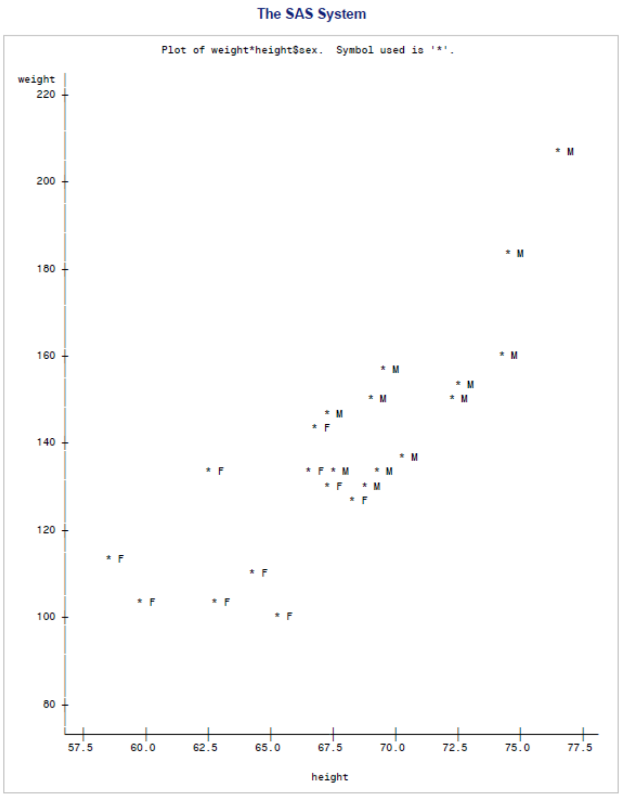
```
* part e);
data athlete_2;
set athlete;
if abp >=100 & hr > 70;
proc print noobs; * suppress row number printing;
run;
```

```
  id  age  race  spb  dbp  hr      abp
4104   17  B     150   90  76  110.000
4108   21  W     140   85  74  103.333
4110   20  W     145   95  75  111.667
```

```
* part f);
proc print data=athlete noobs;
where abp >=100 & hr > 70;
run;
```

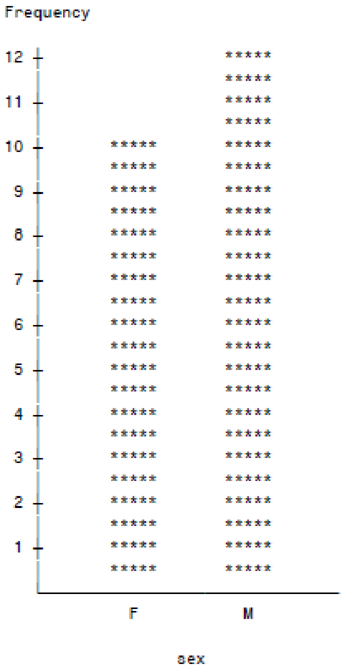
```
  id  age  race  spb  dbp  hr      abp
4104   17  B     150   90  76  110.000
4108   21  W     140   85  74  103.333
4110   20  W     145   95  75  111.667
```

```
* Question 2.1;
data biology;
infile 'Lab2-data1.txt';
input id sex $ age year height weight;
run;

*a);
proc plot;
plot weight*height='*' $ sex;
run;
```


The SAS System

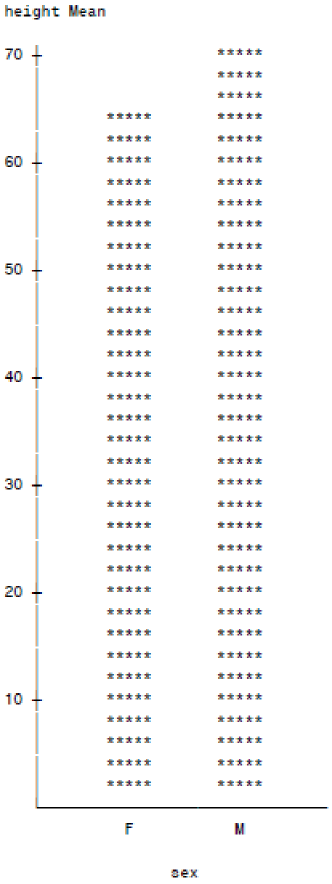
Plot of weight*height$sex. Symbol used is '*'.

```
*b);
proc chart;
vbar sex;
run;
```



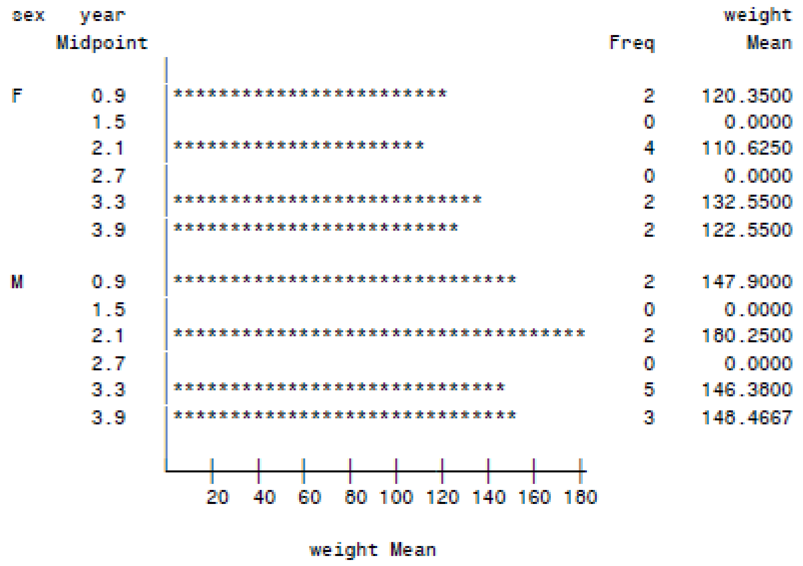
```
Frequency

12 +                 *****
   |                 *****
11 +                 *****
   |                 *****
10 +       *****     *****
   |       *****     *****
 9 +       *****     *****
   |       *****     *****
 8 +       *****     *****
   |       *****     *****
 7 +       *****     *****
   |       *****     *****
 6 +       *****     *****
   |       *****     *****
 5 +       *****     *****
   |       *****     *****
 4 +       *****     *****
   |       *****     *****
 3 +       *****     *****
   |       *****     *****
 2 +       *****     *****
   |       *****     *****
 1 +       *****     *****
   |       *****     *****
   --------------------------------
             F         M

                  sex
```

```
*c);
proc chart;
vbar sex/type=mean sumvar=height;
run;
```



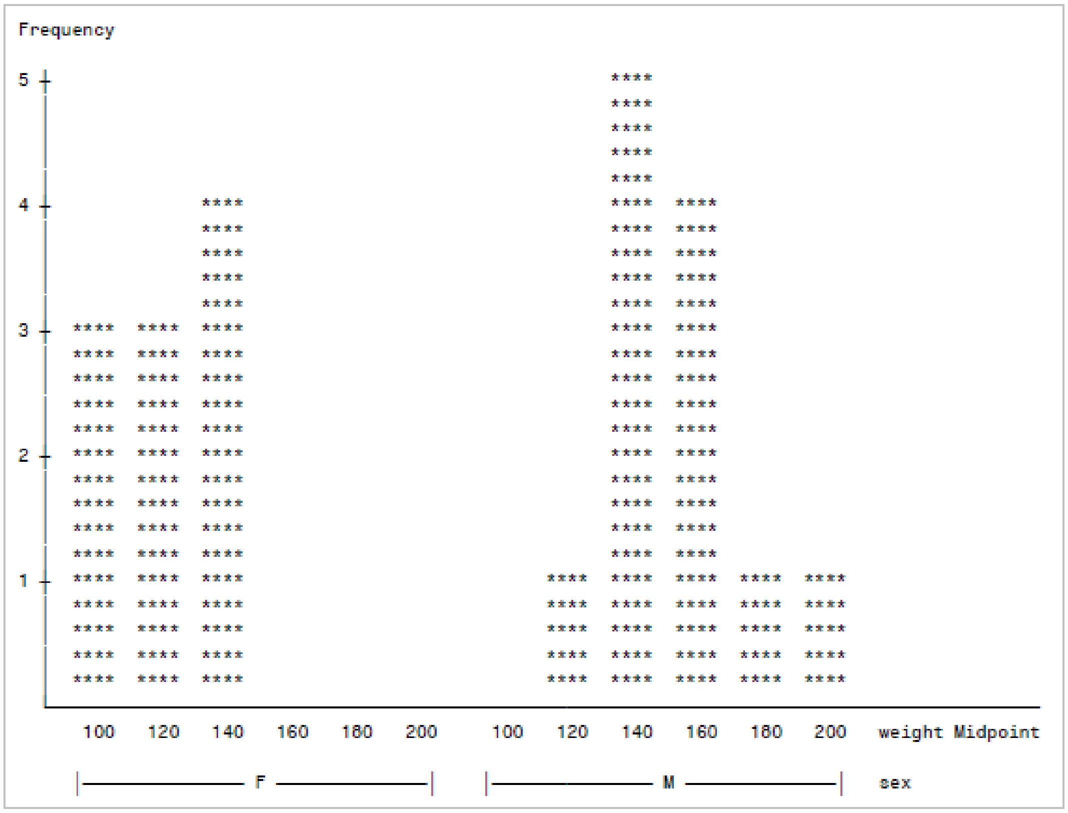
```
height Mean

70 +                  *****
   |                  *****
   |                  *****
   |       *****      *****
   |       *****      *****
60 +       *****      *****
   |       *****      *****
   |       *****      *****
   |       *****      *****
   |       *****      *****
50 +       *****      *****
   |       *****      *****
   |       *****      *****
   |       *****      *****
   |       *****      *****
40 +       *****      *****
   |       *****      *****
   |       *****      *****
   |       *****      *****
   |       *****      *****
30 +       *****      *****
   |       *****      *****
   |       *****      *****
   |       *****      *****
   |       *****      *****
20 +       *****      *****
   |       *****      *****
   |       *****      *****
   |       *****      *****
   |       *****      *****
10 +       *****      *****
   |       *****      *****
   |       *****      *****
   |       *****      *****
   |       *****      *****
   ---------------------------------
             F          M

                  sex
```

```
*d);
proc chart;
hbar year/type=mean sumvar=weight group=sex;
run;
```



```
sex   year                                                    weight
    Midpoint                                        Freq        Mean
             |
F      0.9   |************************                 2    120.3500
       1.5   |                                         0      0.0000
       2.1   |**********************                   4    110.6250
       2.7   |                                         0      0.0000
       3.3   |***************************              2    132.5500
       3.9   |*************************                2    122.5500
             |
M      0.9   |******************************           2    147.9000
       1.5   |                                         0      0.0000
       2.1   |************************************     2    180.2500
       2.7   |                                         0      0.0000
       3.3   |*****************************            5    146.3800
       3.9   |******************************           3    148.4667
             |
             ----+---+---+---+---+---+---+---+---+
                 20  40  60  80 100 120 140 160 180

                          weight Mean
```

```
*e);
proc chart;
vbar weight/midpoints=100 to 200 by 20 type=freq group=sex;
run;
```

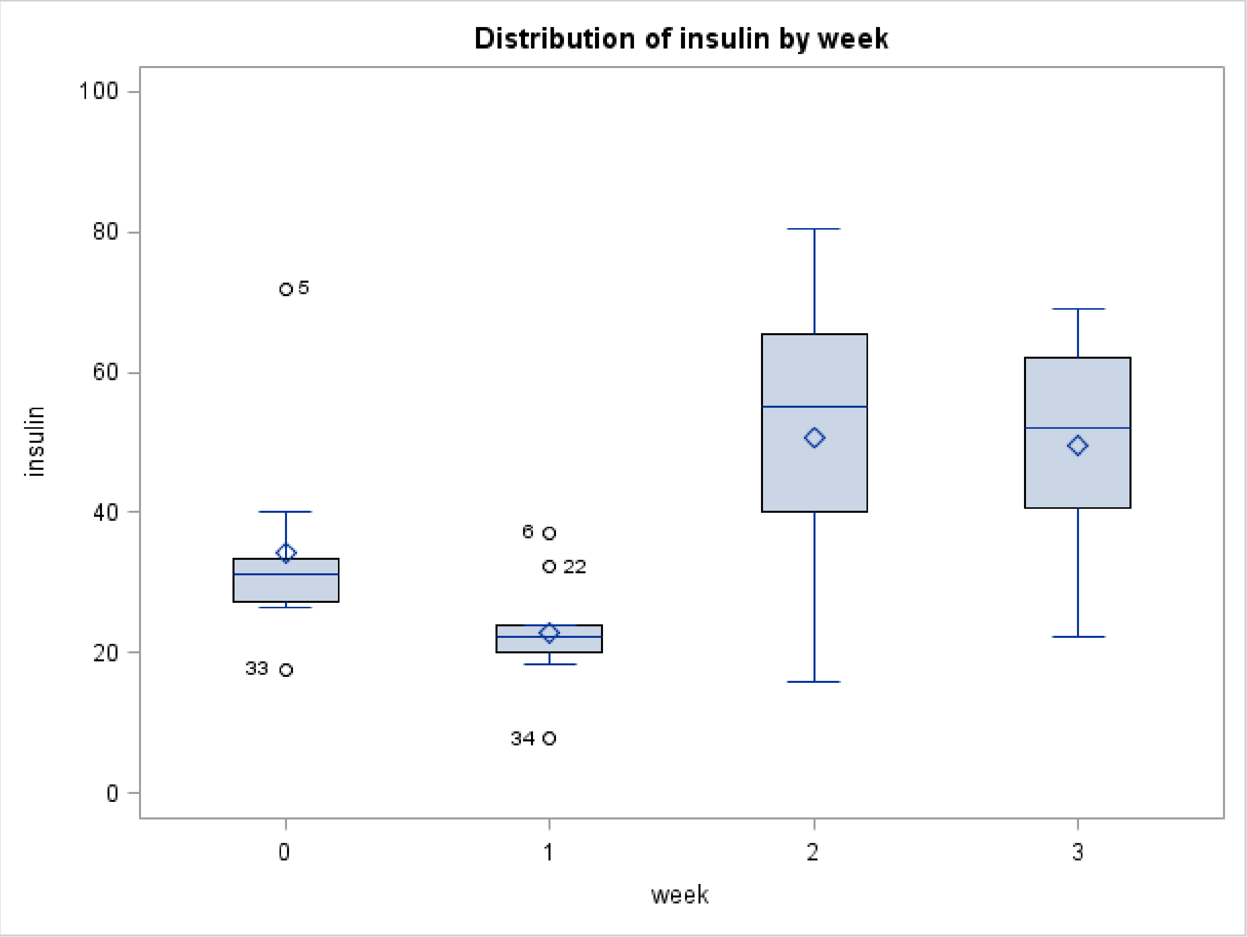


```
Frequency

5 +                                                     ****
  |                                                     ****
  |                                                     ****
  |                                                     ****
  |                                                     ****
4 +               ****                                  ****  ****
  |               ****                                  ****  ****
  |               ****                                  ****  ****
  |               ****                                  ****  ****
  |               ****                                  ****  ****
3 +   ****  ****  ****                                  ****  ****
  |   ****  ****  ****                                  ****  ****
  |   ****  ****  ****                                  ****  ****
  |   ****  ****  ****                                  ****  ****
  |   ****  ****  ****                                  ****  ****
2 +   ****  ****  ****                                  ****  ****
  |   ****  ****  ****                                  ****  ****
  |   ****  ****  ****                                  ****  ****
  |   ****  ****  ****                                  ****  ****
  |   ****  ****  ****                                  ****  ****
1 +   ****  ****  ****                            ****  ****  ****  ****  ****
  |   ****  ****  ****                            ****  ****  ****  ****  ****
  |   ****  ****  ****                            ****  ****  ****  ****  ****
  |   ****  ****  ****                            ****  ****  ****  ****  ****
  |   ****  ****  ****                            ****  ****  ****  ****  ****
  --------------------------------------------------------------------------------------------

       100   120   140   160   180   200    100   120   140   160   180   200    weight Midpoint

      |-------------- F --------------|    |---------------- M --------------|    sex
```

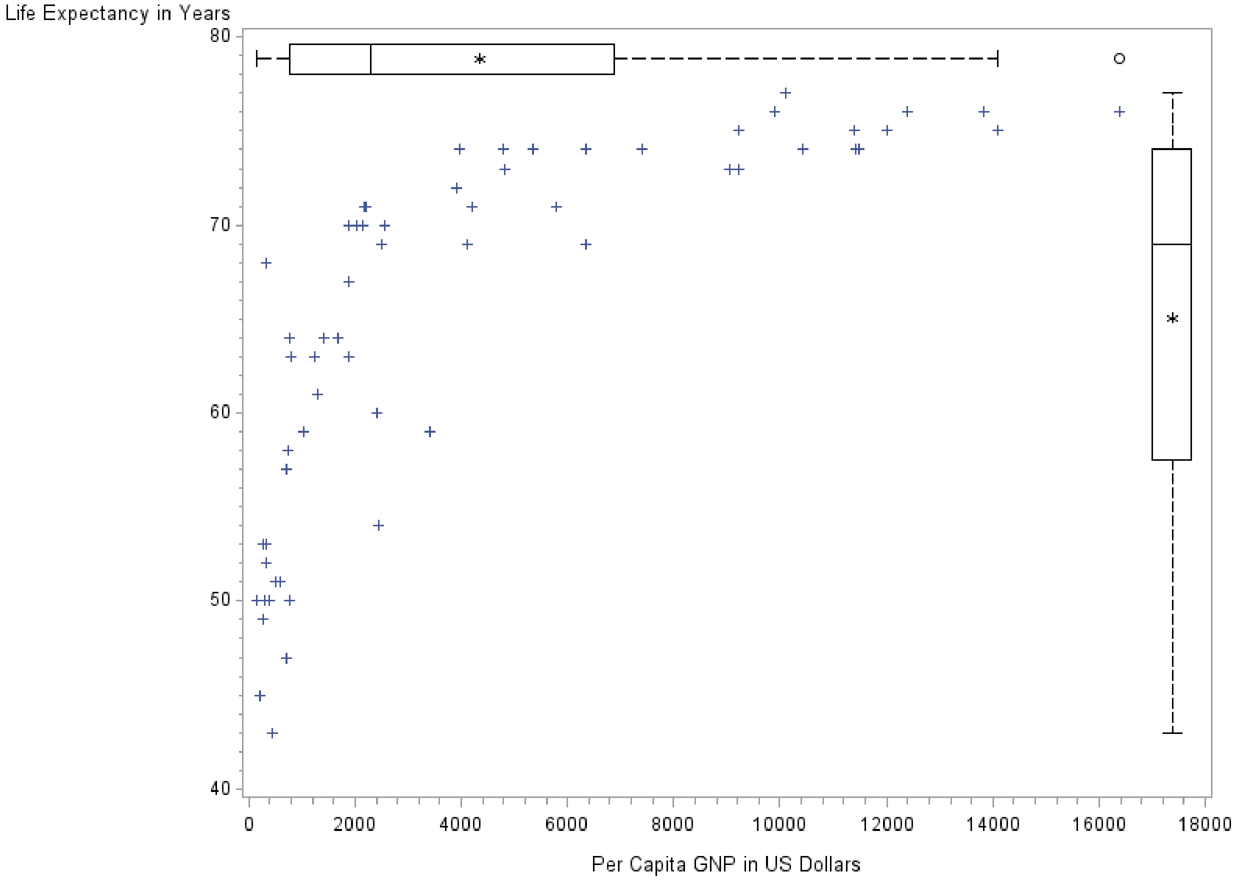
```
* Question 2.7;
libname lab2 'lab2';
data lab2.world;
infile 'Lab2-data2.txt';
input country & $15. birthrat deathrat infmort lifeexp
      popurban percgnp : comma. levtech civillib @;
infgrp = 2;
if infmort < 24 then infgrp = 1;
if infmort >= 74 then infgrp = 3;
if levtech < 24 then techgrp = 1;
else techgrp = 2;
civilgrp = 2;
if civillib < 3 then civilgrp = 1;
if civillib > 5 then civilgrp = 3;

label country = Country Name
      birthrat = Crude Birth Rate
      deathrat = Crude Death Rate
      infmort = Infant Mortality Rate
      lifeexp = Life Expectancy in Years
      popurban = Percent Population in Urban Areas
      percgnp = Per Capita GNP in US Dollars
      levtech = Level of Technology
      civillib = Degree of Civil Liberties
      infgrp = Infant Mortality Group
      techgrp = Level of Technology Group
      civilgrp = Degree of Civil Liberties Group;
run;

* Question 2.8;
proc univariate data = lab2.world noprint;
  var birthrat deathrat popurban;
  output out=lab2.stats pctlpts=33.3 66.7
         pctlpre=birthrat deathrat popurban;
run;
proc print data=lab2.stats;
run;
```

```
Obs  birthrat33_3  deathrat33_3  popurban33_3  birthrat66_7  deathrat66_7  popurban66_7

  1            15             9            45            35            12            71
```

```
* Question 3.9;
data insulin;
input insulin @@;
week=mod(_n_-1,4);
row=_n_;
datalines;
31.2 18.4 55.2 69.2
72.0 37.2 70.4 52.0
31.2 24.0 40.0 42.8
28.2 20.0 42.8 40.6
26.4 20.6 26.8 31.6
40.2 32.2 80.4 66.8
27.2 23.0 60.4 62.0
33.4 22.2 65.6 59.2
17.6 7.8 15.8 22.4
;
run;
proc sort data = insulin;
by week;
run;
proc boxplot data = insulin;
  plot insulin*week / boxstyle=schematicid;
  id row;
run;
```



```
* Question 3.12;
%include "lab2/boxanno.sas";
%boxanno( data=lab2.world, xvar=percgnp, yvar=lifeexp, out=boxplots);
proc gplot data=lab2.world;
  plot lifeexp*percgnp/ annotate = boxplots;
run;
```


Life Expectancy in Years
Per Capita GNP in US Dollars