Source Code Đồ án cuối kỳ

Thông tin chung

- **Môn:** Thiết kế và Phân tích thực nghiệm
- **Lớp:** DS304.L21
- **Nhóm 9**
- **Thành viên nhóm:**

> - Thái Minh Triết - 19522397
> - Chu Hà Thảo Ngân - 19521882
> - Võ Tuấn Anh - 19521226

- **Đề tài báo cáo:** Phân tích ảnh hưởng của các chỉ số sức khỏe đến tiến triển bệnh đái tháo đường

Phân tích thăm dò và trực quan bộ dữ liệu (Python)

In [1]:

```
import pandas as pd
import numpy as np

import seaborn as sns
import matplotlib.pyplot as plt

df = pd.read_csv('/content/drive/MyDrive/#2020-2021 HK2/TK&PTTN/diabetes.tab.tsv',sep='\t')
df
```

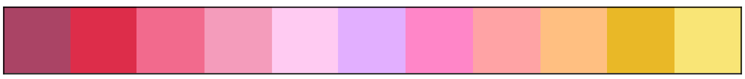
Out[1]:

	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6	Y
0	59	2	32.1	101.00	157	93.2	38.0	4.00	4.8598	87	151
1	48	1	21.6	87.00	183	103.2	70.0	3.00	3.8918	69	75
2	72	2	30.5	93.00	156	93.6	41.0	4.00	4.6728	85	141
3	24	1	25.3	84.00	198	131.4	40.0	5.00	4.8903	89	206
4	50	1	23.0	101.00	192	125.4	52.0	4.00	4.2905	80	135
...	...	...	...	...	...	...	...	...	...	...	...
437	60	2	28.2	112.00	185	113.8	42.0	4.00	4.9836	93	178
438	47	2	24.9	75.00	225	166.0	42.0	5.00	4.4427	102	104
439	60	2	24.9	99.67	162	106.6	43.0	3.77	4.1271	95	132
440	36	1	30.0	95.00	201	125.2	42.0	4.79	5.1299	85	220
441	36	1	19.6	71.00	250	133.2	97.0	3.00	4.5951	92	57

442 rows × 11 columns

In [2]:

```
import seaborn as sns
import matplotlib.pyplot as plt

#defining colour palette
def custom_palette(custom_colors):
    customPalette = sns.set_palette(sns.color_palette(custom_colors))
    sns.palplot(sns.color_palette(custom_colors),size=0.8)
    plt.tick_params(axis='both', labelsize=0, length = 0)

red = ["#4f000b","#720026","#ce4257","#ff7f51","#ff9b54"]
bo = ["#6930c3","#5e60ce","#0096c7","#48cae4","#ade8f4","#ff7f51","#ff9b54","#ffbf69"]
pink = ["#aa4465","#dd2d4a","#f26a8d","#f49cbb","#ffcbf2","#e2afff","#ff86c8","#ffa3a5","#ffbf81","#e9b827","#f9e576"]
custom_palette(pink)
```



In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 442 entries, 0 to 441
Data columns (total 11 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
```

```
 0   AGE     442 non-null    int64
 1   SEX     442 non-null    int64
 2   BMI     442 non-null    float64
 3   BP      442 non-null    float64
 4   S1      442 non-null    int64
 5   S2      442 non-null    float64
 6   S3      442 non-null    float64
 7   S4      442 non-null    float64
 8   S5      442 non-null    float64
 9   S6      442 non-null    int64
 10  Y       442 non-null    int64
dtypes: float64(6), int64(5)
memory usage: 38.1 KB
```

In [4]:

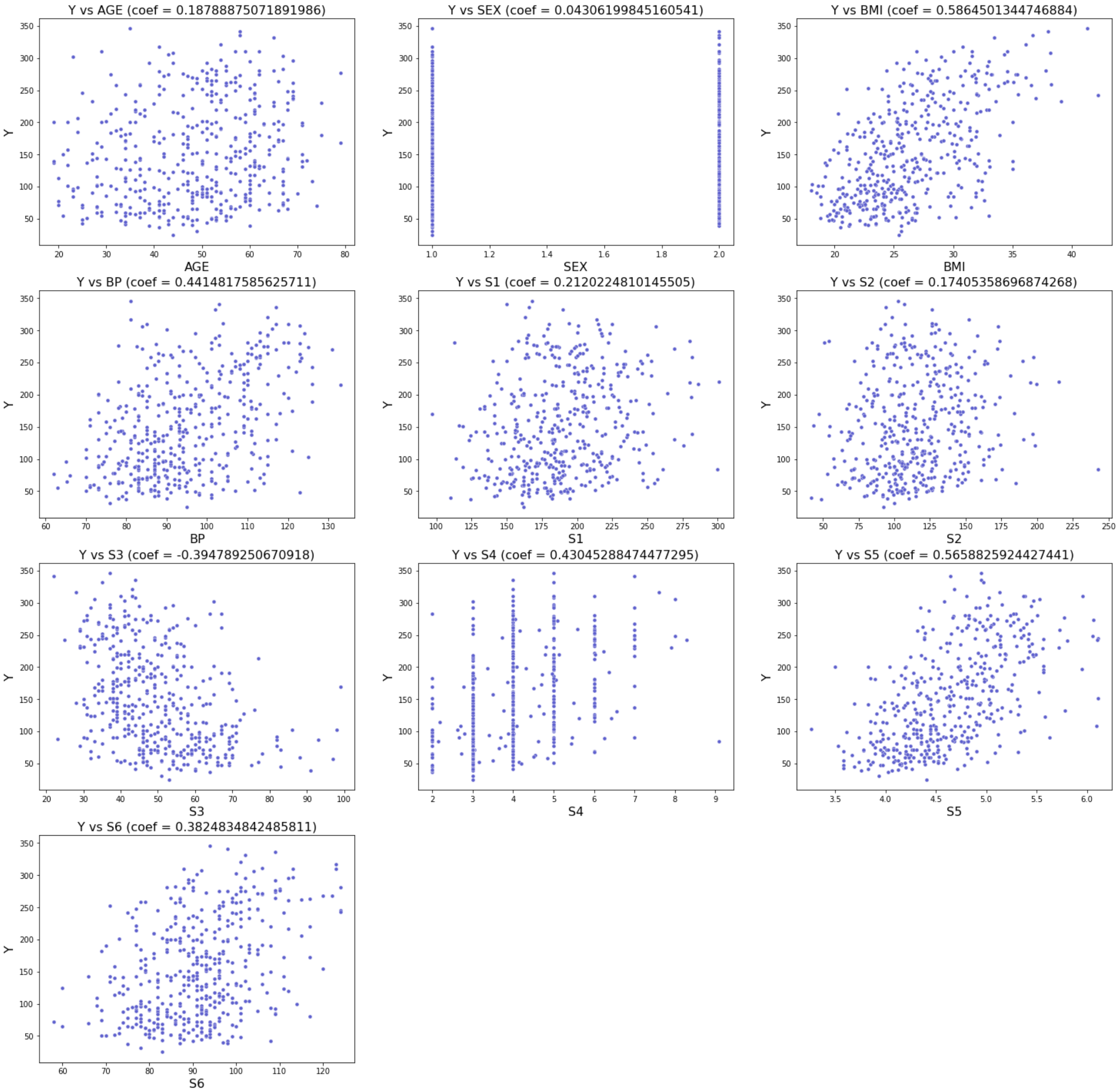
```
df.describe()
```

Out[4]:

	AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6	Y
count	442.000000	442.000000	442.000000	442.000000	442.000000	442.000000	442.000000	442.000000	442.000000	442.000000	442.000000
mean	48.518100	1.468326	26.375792	94.647014	189.140271	115.439140	49.788462	4.070249	4.641411	91.260181	152.133484
std	13.109028	0.499561	4.418122	13.831283	34.608052	30.413081	12.934202	1.290450	0.522391	11.496335	77.093005
min	19.000000	1.000000	18.000000	62.000000	97.000000	41.600000	22.000000	2.000000	3.258100	58.000000	25.000000
25%	38.250000	1.000000	23.200000	84.000000	164.250000	96.050000	40.250000	3.000000	4.276700	83.250000	87.000000
50%	50.000000	1.000000	25.700000	93.000000	186.000000	113.000000	48.000000	4.000000	4.620050	91.000000	140.500000
75%	59.000000	2.000000	29.275000	105.000000	209.750000	134.500000	57.750000	5.000000	4.997200	98.000000	211.500000
max	79.000000	2.000000	42.200000	133.000000	301.000000	242.400000	99.000000	9.090000	6.107000	124.000000	346.000000

In [5]:

```
corr = df.corr()
fig, axs = plt.subplots(4,3,figsize=(25,25))
for i, ax in zip(range(10), axs.flat):
  sns.scatterplot(ax=ax, data=df, x=df.columns[:-1][i] , y='Y',color=bo[1],s=25)
  c = corr['Y'][df.columns[:-1][i]]
  ax.set_title('Y vs {} (coef = {})'.format(df.columns[:-1][i],str(c)), fontsize=16)
  ax.set_xlabel(df.columns[:-1][i], fontsize = 16)
  ax.set_ylabel('Y', fontsize = 16)
fig.delaxes(axs[3][1])
fig.delaxes(axs[3][2])
plt.savefig('Scatter_Y.png',transparent=False,bbox_inches = 'tight',dpi=300)
plt.show()
```



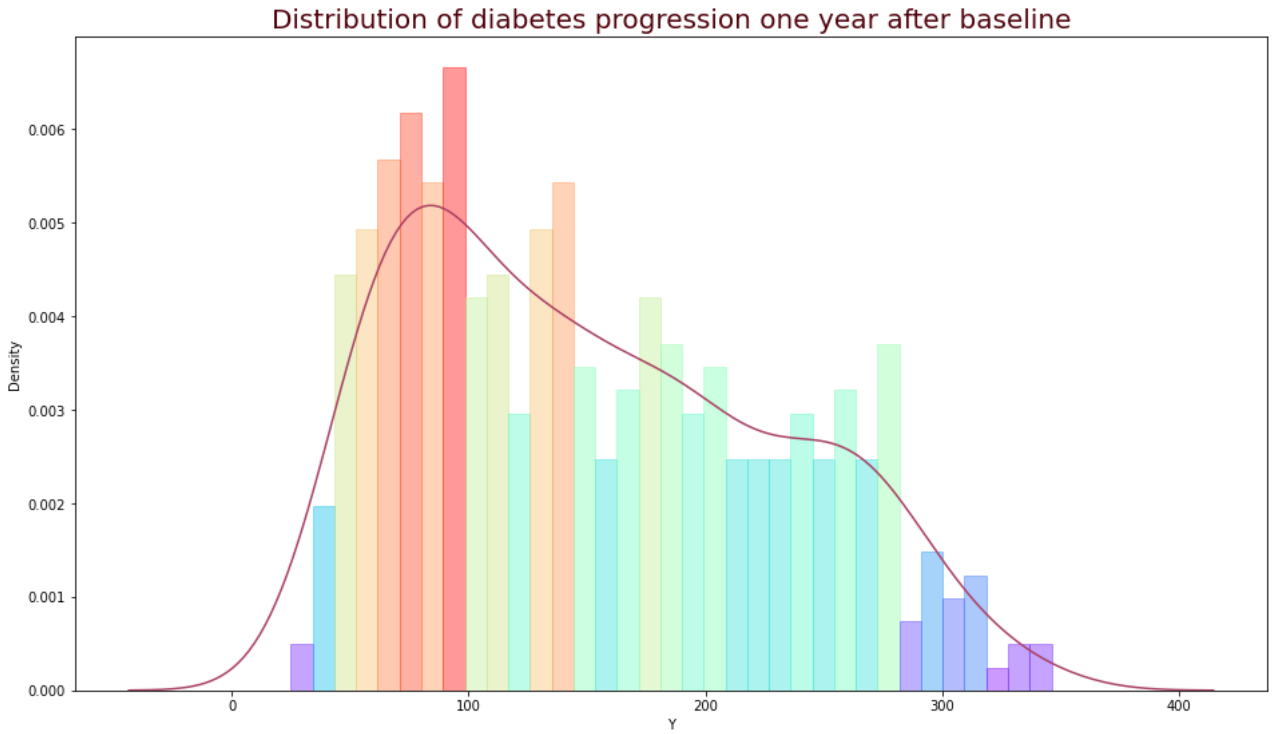
In [6]:

```
def hist(col,title,norm):
    fig, ax = plt.subplots(1,1, figsize=(16,9))
    #plt.figure(figsize = (16,9))
    #ax.set_xlim(0,400)
    ax = sns.distplot(col,hist=True,bins=35,fit=norm);

    values = np.array([patch.get_height() for patch in ax.patches])

    #normalizing the values to get a range of colours
    norm = plt.Normalize(values.min(), values.max())

    #range of colours from colourmap-rainbow
    colors = plt.cm.rainbow(norm(values))

    #set colour for each patch
    for patch, color in zip(ax.patches, colors):
        patch.set_color(color)

    plt.title(title, size = 20, color = red[0])
```

In [7]:

```
hist(df['Y'],'Distribution of diabetes progression one year after baseline',None)
plt.savefig('Dist_Y.png',transparent=False,bbox_inches = 'tight',dpi=300)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a deprecated functi
on and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function wi
th similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [8]:

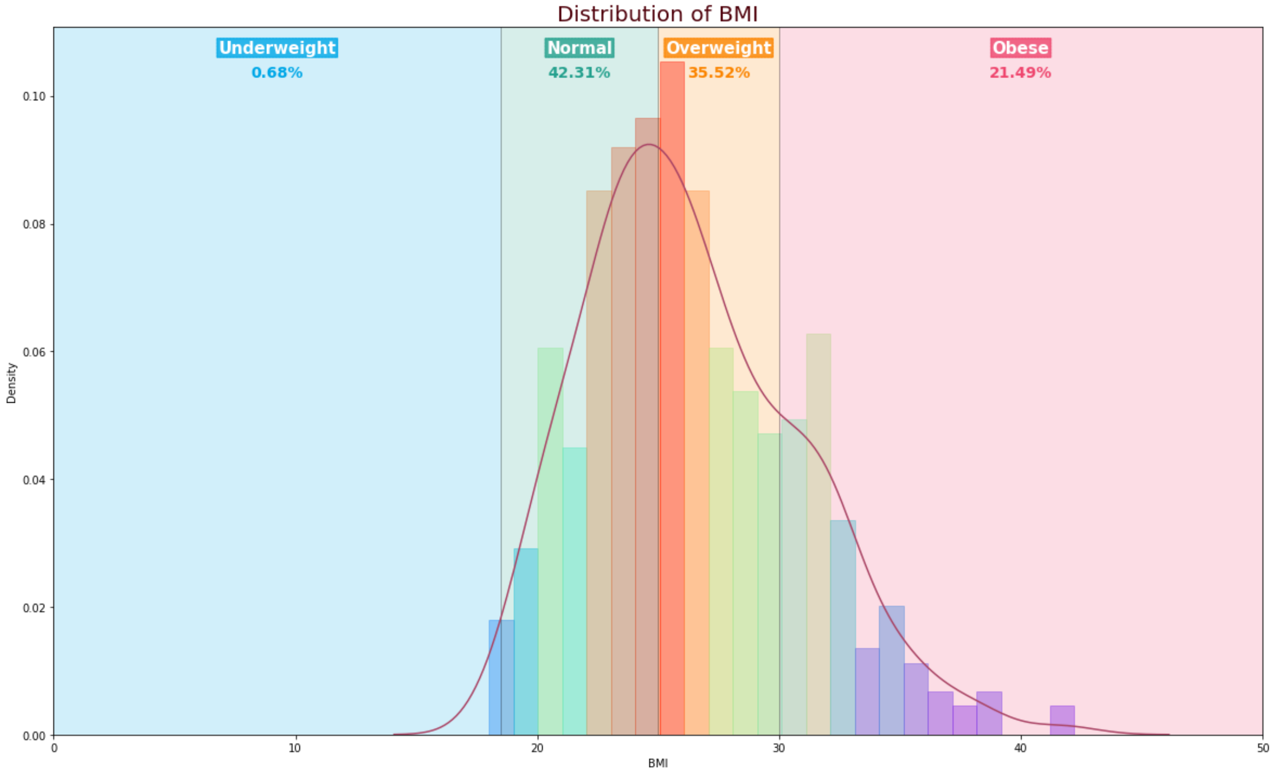
```
def get_percent(status):
  if(status=='Underweight'):
    return str(np.round((len(df[df['BMI']<=18.5])/442)*100,2))+'%'
  if(status=='Normal'):
    tmp = df[df['BMI']>18.5]
    return str(np.round((len(tmp[tmp['BMI']<=25])/442)*100,2))+'%'
  if(status=='Overweight'):
    tmp = df[df['BMI']>25]
    return str(np.round((len(tmp[tmp['BMI']<=30])/442)*100,2))+'%'
  tmp = df[df['BMI']>25]
  return str(np.round((len(tmp[tmp['BMI']>30])/442)*100,2))+'%'
```

In [9]:

```
fig, ax = plt.subplots(1,1, figsize=(20,12))
ax = sns.distplot(df['BMI'], kde=True,bins=24);

values = np.array([patch.get_height() for patch in ax.patches])

norm = plt.Normalize(values.min(), values.max())
colors = plt.cm.rainbow(norm(values))
for patch, color in zip(ax.patches, colors):
        patch.set_color(color)

#colours for different bmi categories
span_color =  ['#00a8e8','#25a18e','#fb8500','#ef476f']

#range of values for different bmi categories
span_range = [[0,18.5], [18.5,25], [25,30], [30,50]]
ax.set_xlim(0,50)
for idx, span_title in enumerate(['Underweight', 'Normal', 'Overweight', 'Obese']):
    ax.annotate(span_title,
                xy=(sum(span_range[idx])/2 ,0.029),
                xytext=(0,470), textcoords='offset points',
                va='top', ha="center",
                color="w", fontsize=5, fontweight='bold',
                size=15,
                bbox=dict(boxstyle='sawtooth', pad=0.1, color=span_color[idx], alpha=0.8))

    ax.annotate(get_percent(span_title),
            xy=(sum(span_range[idx])/2 ,0.025),
            xytext=(0,470), textcoords='offset points',
            va='top', ha="center",
            color=span_color[idx], fontsize=5, fontweight='bold',
            size=14,
            bbox=dict(boxstyle='sawtooth', pad=0.1, color=span_color[idx], alpha=0))

    ax.axvspan(span_range[idx][0],span_range[idx][1], color=span_color[idx], alpha=0.18,ec ='black')
plt.title("Distribution of BMI", size = 20, color = red[0])
plt.savefig('Dist_BMI.png',transparent=False,bbox_inches = 'tight',dpi=300)
plt.show()
```

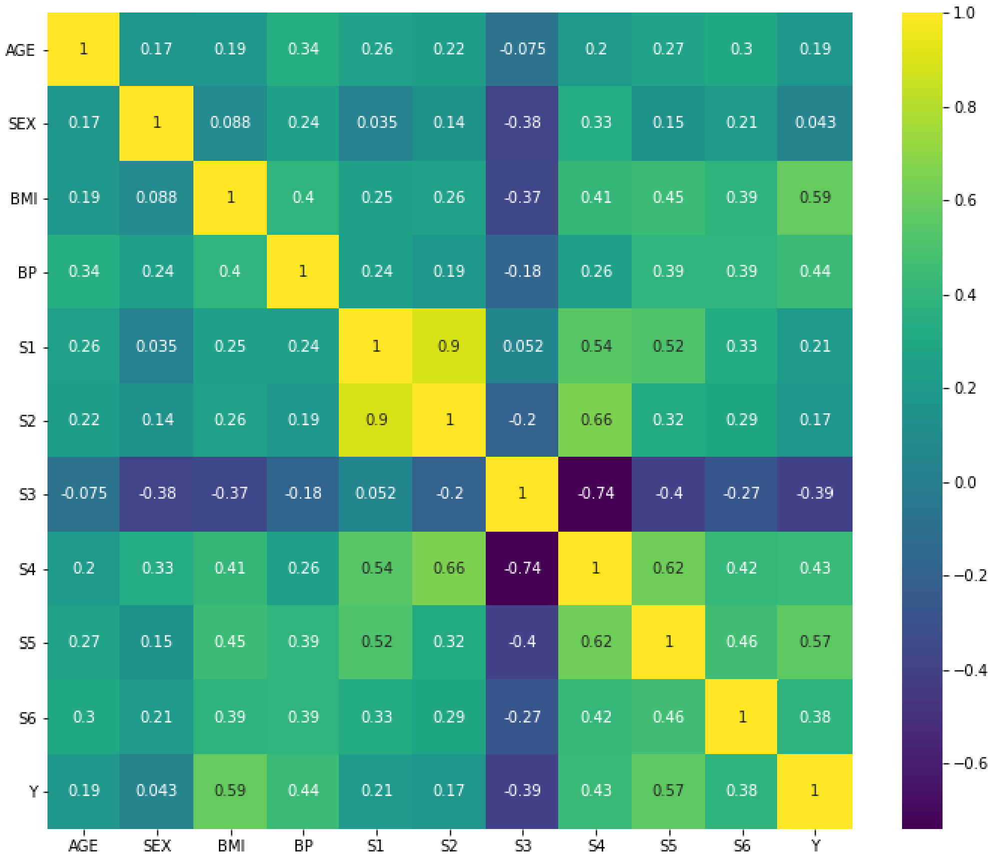
```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning: `distplot` is a deprecated functi
on and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function wi
```

```
th similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```



In [18]:

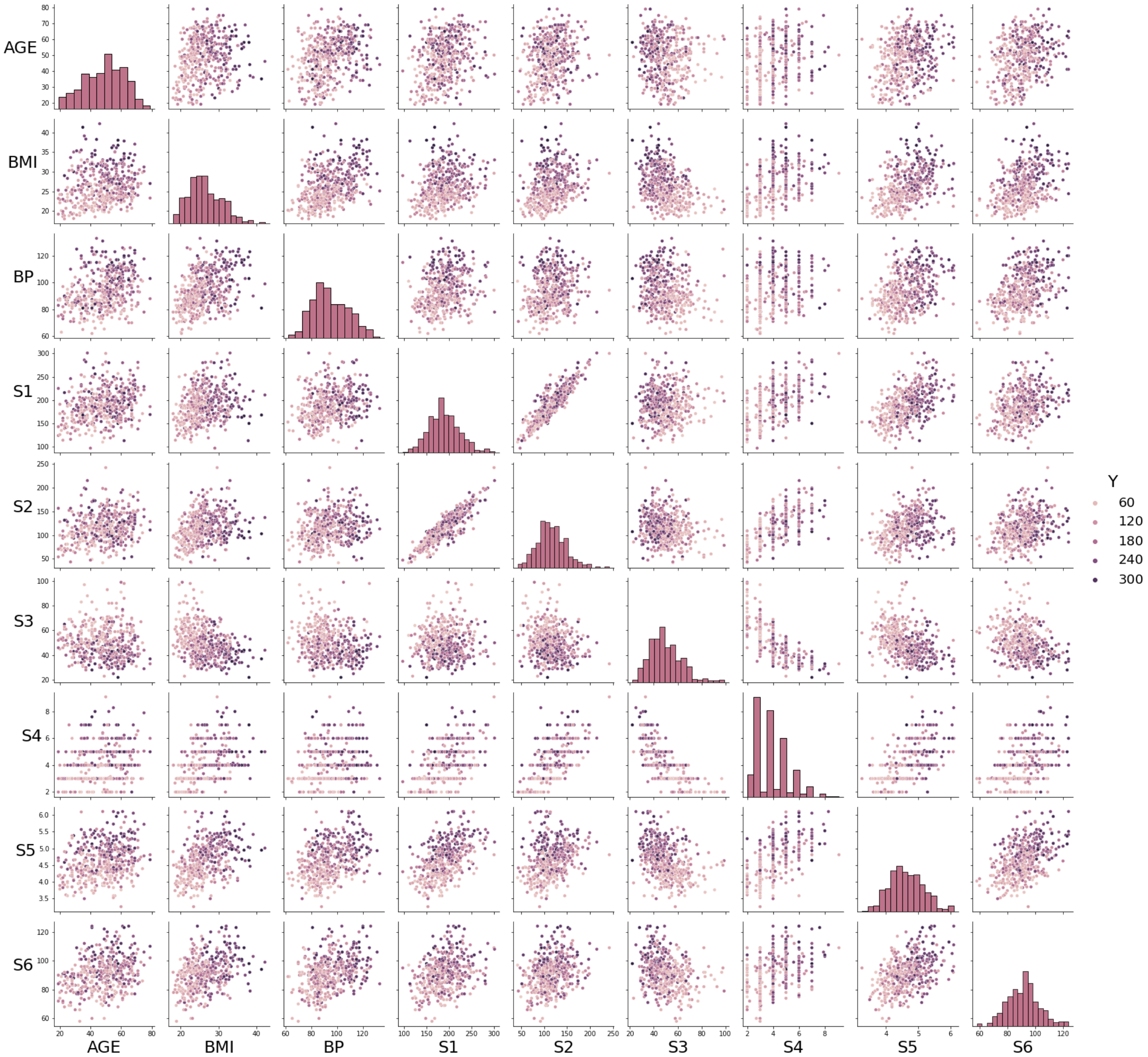
```
temp_df = df[df.columns]
corr = temp_df.corr()
plt.figure(figsize=(12,10))
sns.heatmap(corr,
            xticklabels=corr.columns.values,
            yticklabels=corr.columns.values,annot=True,cmap='viridis')
plt.yticks(rotation=360)
plt.savefig('Corr.png',transparent=False,bbox_inches = 'tight',dpi=300)
```



In [11]:

```
df2 = df.drop(['SEX'],axis=1)
g = sns.PairGrid(df2,hue='Y')
g.map_diag(sns.histplot,hue=None)
```

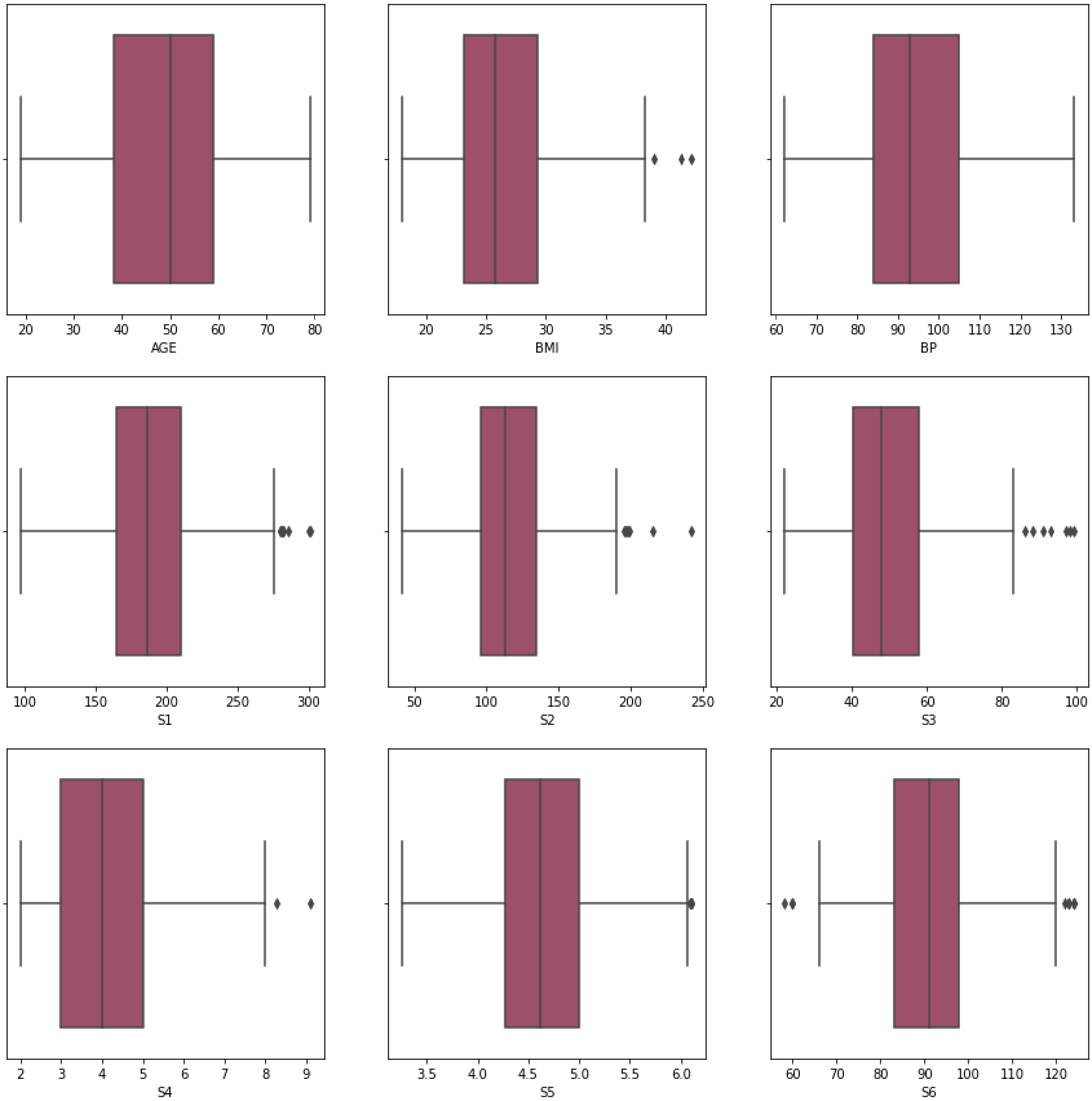
```
g.map_offdiag(sns.scatterplot,s=25).add_legend(title='Y', fontsize= 20)
for axes in g.axes.flat:
     axes.set_ylabel(axes.get_ylabel(), rotation=0, horizontalalignment='right')
     axes.xaxis.get_label().set_fontsize(25)
     axes.yaxis.get_label().set_fontsize(25)
plt.setp(g._legend.get_title(), fontsize=25)
plt.savefig('Scatter_factors.png',transparent=False,bbox_inches = 'tight',dpi=300)
plt.show()
```



Data Preprocessing

In [12]:

```
cols=['AGE','BMI','BP','S1','S2','S3','S4','S5','S6']
fig, axs = plt.subplots(3,3,figsize=(15,15))
for i, ax in zip(range(10), axs.flat):
  sns.boxplot(x=df[cols[i]],ax=ax)
plt.savefig('box.png',dpi=200)
```



In [13]:

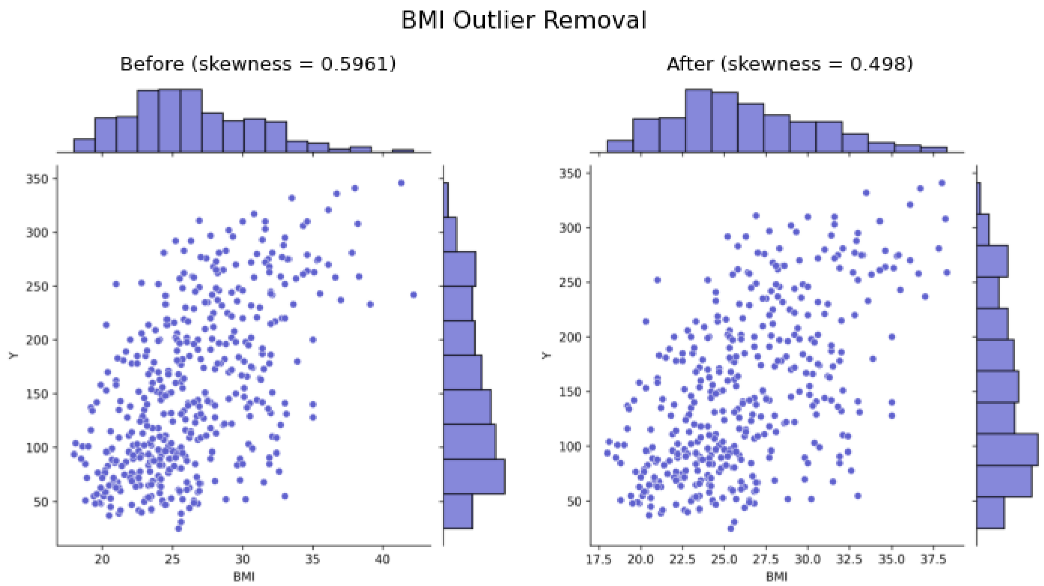
```
df_p =df.copy()
cols=['AGE','BMI','BP','S1','S2','S3','S4','S5','S6']
for attr in df_p.columns[:-1]:
  if(attr!='SEX' and attr!='Y'):
    Q1= np.percentile(df_p[attr], 25)
    Q3 = np.percentile(df_p[attr], 75)
    IQR = Q3 - Q1
    olr_up = Q3+1.5*IQR
    olr_low = Q1-1.5*IQR
    df_p = df_p.drop(df_p[df_p[attr]>olr_up].index)
    df_p = df_p.drop(df_p[df_p[attr]<olr_low].index)
```

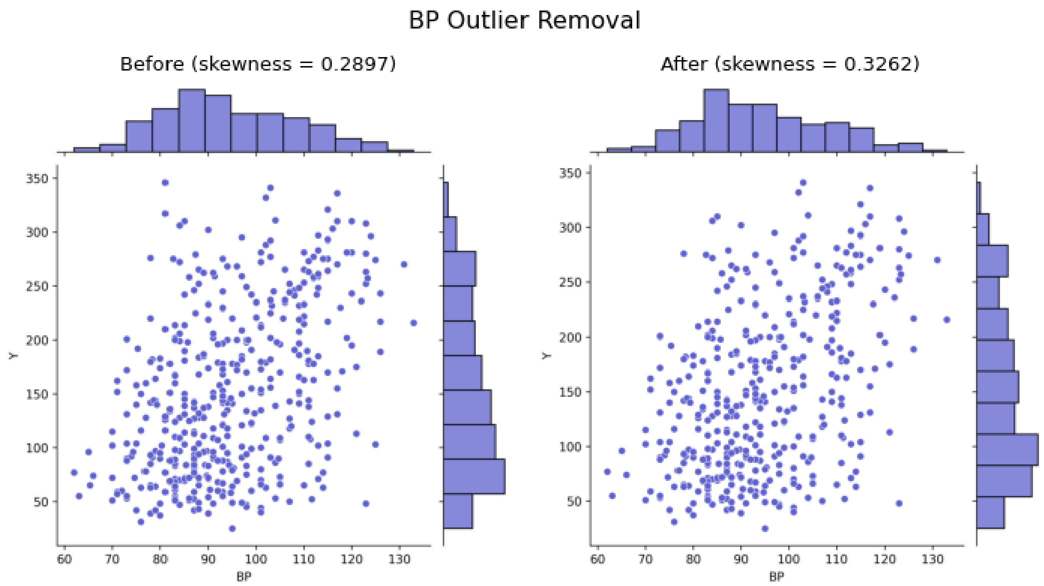
In [14]:

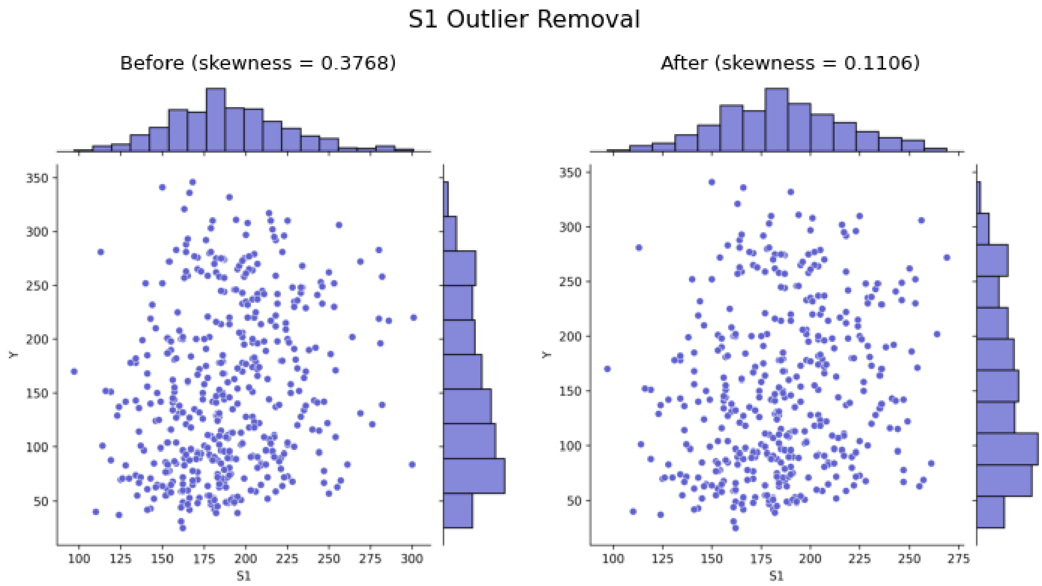
```
import matplotlib as mpl
from scipy.stats import skew
def compare(factor_name):
  i = factor_name

  tmp1 = sns.jointplot(data=df, x= i, y='Y',color=bo[1])
  tmp2 = sns.jointplot(data = df_p, x= i, y='Y',color=bo[1])
  tmp1.savefig('tmp1.png',dpi=300)
  plt.close(tmp1.fig)
  tmp2.savefig('tmp2.png',dpi=300)
  plt.close(tmp2.fig)
  fig, axs = plt.subplots(1,2,figsize=(10,10))
  axs[0].imshow(mpl.image.imread('tmp1.png'))
  axs[0].set_title('Before (skewness = {})'.format(np.round(skew(df[i]),4)),fontsize=13)

  axs[1].imshow(mpl.image.imread('tmp2.png'))
  axs[1].set_title('After (skewness = {})'.format(np.round(skew(df_p[i]),4)),fontsize=13)
  [ax.set_axis_off() for ax in axs.ravel()]
```

```
    plt.tight_layout()
    plt.suptitle(factor_name+' Outlier Removal',y=0.8,fontsize=16)
    fig.savefig('{}OR.png'.format(factor_name),dpi=100)
    plt.show()
```

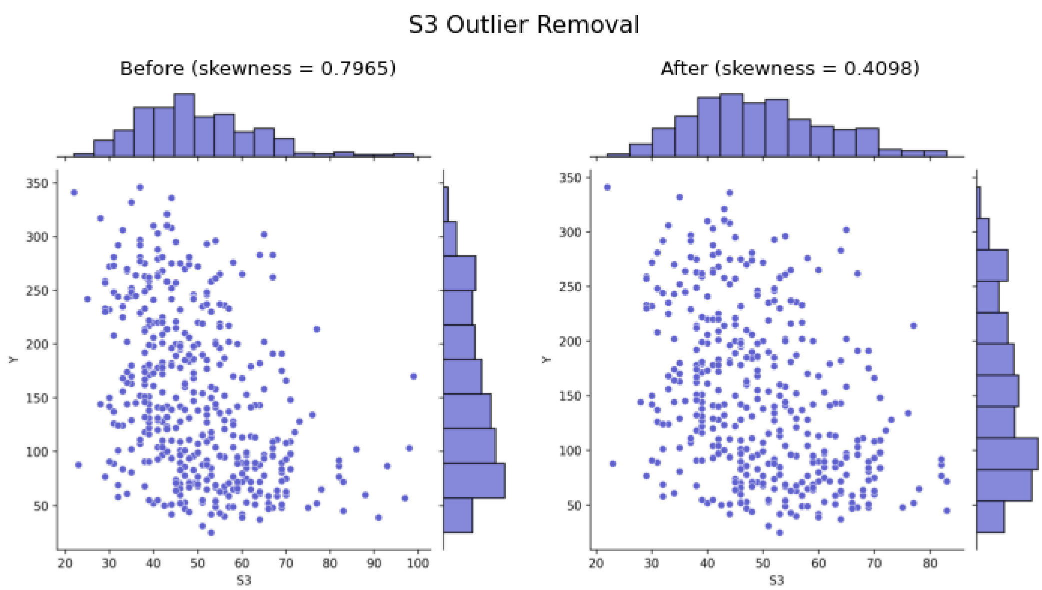
In [15]:
```
for i in cols:
    compare(i)
```



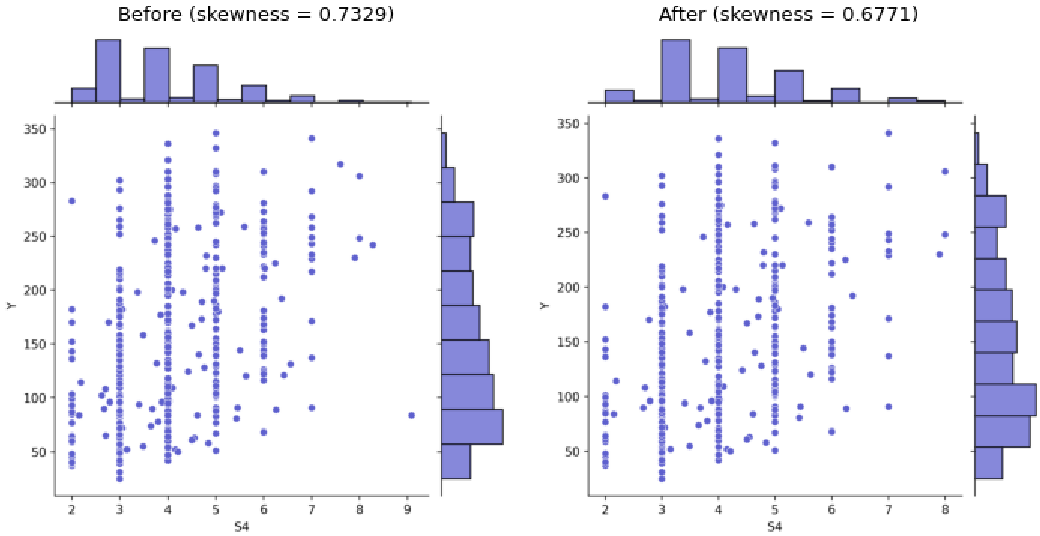



S1 Outlier Removal
Before (skewness = 0.3768)
After (skewness = 0.1106)
Y
S1

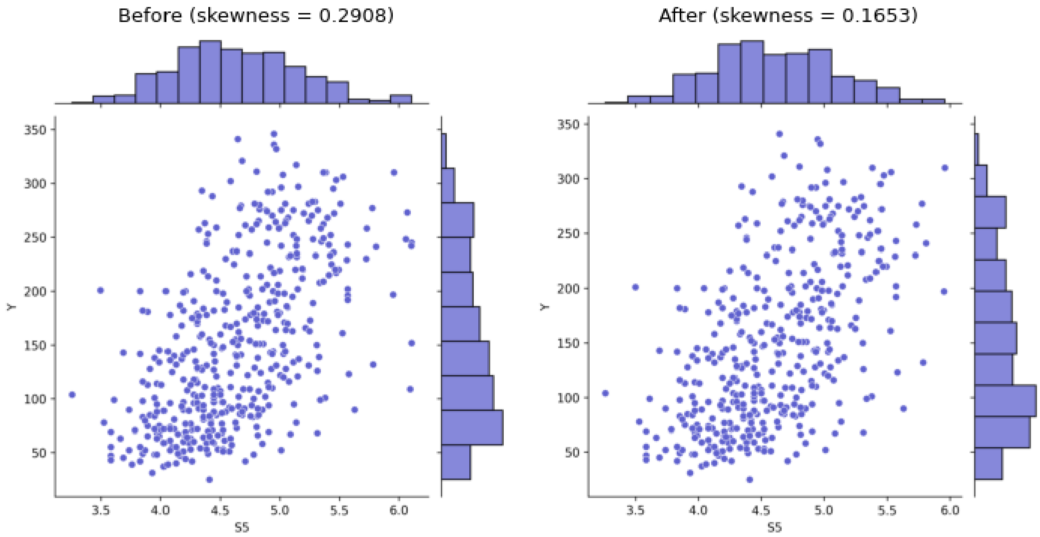

S2 Outlier Removal
Before (skewness = 0.4351)
After (skewness = 0.0339)
Y
S2


S3 Outlier Removal
Before (skewness = 0.7965)
After (skewness = 0.4098)
Y
S3

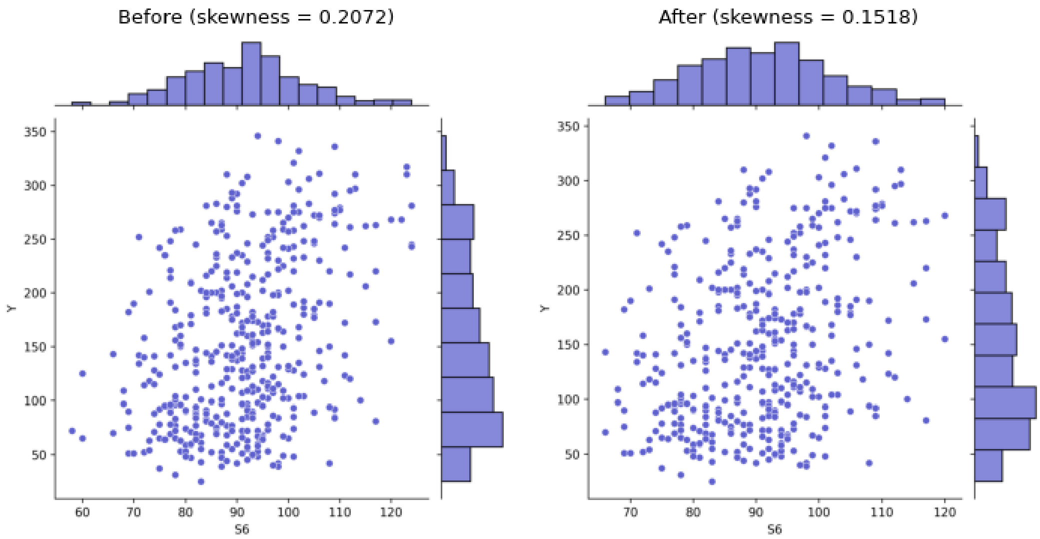
S4 Outlier Removal



S5 Outlier Removal



S6 Outlier Removal



Phân tích ảnh hưởng, tương tác của các yếu tố và xây dựng mô hình hồi quy (R)

1. Trên bộ dữ liệu trước khi xử lý

```
#library
library(kernlab)
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:kernlab':
##
##     alpha
```

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-2
```

```
library(moments)
library(knitr)

# Doc du lieu
df <- read.csv('diabetes.tab.tsv',header=TRUE,sep = '\t')

#train-test
  set.seed(123)
  train.index <- createDataPartition(df$Y, p = .8, list = FALSE)
  train <- df[ train.index,]
  test  <- df[-train.index,]

  x.train <- train[,1:10]
  x.test <- test[,1:10]
  y.train <- train[,11]
  y.test <- test[,11]

RMSE = function(m, o){
    sqrt(mean((m - o)^2))
```

```
}

# Effect of Factors no interactions
av <- aov(Y~.,data=train)
summary(av)
```

```
##              Df  Sum Sq Mean Sq F value   Pr(>F)    
## AGE           1   85855   85855  28.653 1.58e-07 ***
## SEX           1     252     252   0.084    0.772    
## BMI           1  591812  591812 197.514  < 2e-16 ***
## BP            1  104605  104605  34.911 8.30e-09 ***
## S1            1    5132    5132   1.713    0.191    
## S2            1    4662    4662   1.556    0.213    
## S3            1  173465  173465  57.893 2.68e-13 ***
## S4            1    1210    1210   0.404    0.526    
## S5            1   50637   50637  16.900 4.93e-05 ***
## S6            1    4458    4458   1.488    0.223    
## Residuals   344 1030730    2996                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Effect of Factor with interaction
av <- aov(Y~(AGE+BMI+BP+S1+S2+S3+S5+S6)*S4,data=train)
summary(av)
```

```
##              Df  Sum Sq Mean Sq F value   Pr(>F)    
## AGE           1   85855   85855  27.968 2.22e-07 ***
## BMI           1  591789  591789 192.779  < 2e-16 ***
## BP            1   98099   98099  31.956 3.36e-08 ***
## S1            1    5402    5402   1.760 0.185547    
## S2            1    7923    7923   2.581 0.109082    
## S3            1  147757  147757  48.133 2.05e-11 ***
## S5            1   47191   47191  15.373 0.000107 ***
## S6            1    3657    3657   1.191 0.275845    
## S4            1     599     599   0.195 0.659068    
## AGE:S4        1    5122    5122   1.669 0.197321    
## BMI:S4        1   10028   10028   3.267 0.071591 .  
## BP:S4         1     169     169   0.055 0.814705    
## S1:S4         1    1218    1218   0.397 0.529180    
## S2:S4         1     386     386   0.126 0.723111    
## S3:S4         1     182     182   0.059 0.807668    
## S5:S4         1    3702    3702   1.206 0.272916    
## S6:S4         1    9221    9221   3.004 0.083984 .  
## Residuals   337 1034516    3070                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
av <- aov(Y~(AGE+BMI+BP+S1+S2+S3+S5+S6)*SEX,data=train)
summary(av)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)    
## AGE           1  85855   85855  29.032 1.34e-07 ***
## BMI           1 591789  591789 200.115  < 2e-16 ***
```

```
## BP            1  98099   98099  33.172 1.90e-08 ***
## S1            1   5402    5402   1.827 0.177416
## S2            1   7923    7923   2.679 0.102591
## S3            1 147757  147757  49.965 9.06e-12 ***
## S5            1  47191   47191  15.958 7.96e-05 ***
## S6            1   3657    3657   1.237 0.266911
## SEX           1  33132   33132  11.204 0.000909 ***
## AGE:SEX       1  16983   16983   5.743 0.017101 *
## BMI:SEX       1   9617    9617   3.252 0.072228 .
## BP:SEX        1   1313    1313   0.444 0.505624
## S1:SEX        1    106     106   0.036 0.850020
## S2:SEX        1   6128    6128   2.072 0.150940
## S3:SEX        1     74      74   0.025 0.874516
## S5:SEX        1    776     776   0.262 0.608866
## S6:SEX        1    425     425   0.144 0.704953
## Residuals   337 996590    2957
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
av <- aov(Y~AGE*BMI*BP*S1*S2*S3*S5*S6+AGE*SEX,data=train)
summary(av)
```

```
##                         Df Sum Sq Mean Sq F value   Pr(>F)
## AGE                      1  85855   85855  28.663 5.59e-07 ***
## BMI                      1 591789  591789 197.571  < 2e-16 ***
## BP                       1  98099   98099  32.751 1.12e-07 ***
## S1                       1   5402    5402   1.804 0.182355
## S2                       1   7923    7923   2.645 0.107038
## S3                       1 147757  147757  49.329 2.77e-10 ***
## S5                       1  47191   47191  15.755 0.000137 ***
## S6                       1   3657    3657   1.221 0.271858
## SEX                      1  33132   33132  11.061 0.001238 **
## AGE:BMI                  1   9377    9377   3.131 0.079915 .
## AGE:BP                   1  11912   11912   3.977 0.048883 *
## BMI:BP                   1  21859   21859   7.298 0.008123 **
## AGE:S1                   1      2       2   0.001 0.979064
## BMI:S1                   1  10422   10422   3.479 0.065098 .
## BP:S1                    1    208     208   0.069 0.792619
## AGE:S2                   1   8377    8377   2.797 0.097623 .
## BMI:S2                   1    523     523   0.175 0.676877
## BP:S2                    1     85      85   0.028 0.866446
## S1:S2                    1    469     469   0.157 0.693186
## AGE:S3                   1   1185    1185   0.396 0.530743
## BMI:S3                   1     54      54   0.018 0.893198
## BP:S3                    1   2814    2814   0.939 0.334808
## S1:S3                    1   1916    1916   0.640 0.425734
## S2:S3                    1   3100    3100   1.035 0.311488
## AGE:S5                   1   7201    7201   2.404 0.124219
## BMI:S5                   1   2705    2705   0.903 0.344256
## BP:S5                    1   1357    1357   0.453 0.502542
## S1:S5                    1      1       1   0.000 0.984883
## S2:S5                    1   3635    3635   1.214 0.273279
## S3:S5                    1   1596    1596   0.533 0.467191
## AGE:S6                   1   3816    3816   1.274 0.261765
## BMI:S6                   1    890     890   0.297 0.586833
```

```
## BP:S6                  1   1259    1259   0.420 0.518276
## S1:S6                  1    895     895   0.299 0.585949
## S2:S6                  1    337     337   0.112 0.738034
## S3:S6                  1   5373    5373   1.794 0.183538
## S5:S6                  1    704     704   0.235 0.628842
## AGE:SEX                1  18271   18271   6.100 0.015230 *
## AGE:BMI:BP             1   5803    5803   1.937 0.167085
## AGE:BMI:S1             1    968     968   0.323 0.570925
## AGE:BP:S1              1    583     583   0.195 0.660098
## BMI:BP:S1              1   1299    1299   0.434 0.511745
## AGE:BMI:S2             1   1138    1138   0.380 0.538982
## AGE:BP:S2              1   7179    7179   2.397 0.124789
## BMI:BP:S2              1   1414    1414   0.472 0.493618
## AGE:S1:S2              1    813     813   0.271 0.603555
## BMI:S1:S2              1   4960    4960   1.656 0.201177
## BP:S1:S2               1    124     124   0.041 0.839153
## AGE:BMI:S3             1    375     375   0.125 0.724075
## AGE:BP:S3              1   1591    1591   0.531 0.467810
## BMI:BP:S3              1    112     112   0.037 0.847059
## AGE:S1:S3              1   4336    4336   1.448 0.231792
## BMI:S1:S3              1   3601    3601   1.202 0.275545
## BP:S1:S3               1   3811    3811   1.272 0.262034
## AGE:S2:S3              1    343     343   0.114 0.735880
## BMI:S2:S3              1   1290    1290   0.431 0.513204
## BP:S2:S3               1    122     122   0.041 0.840332
## S1:S2:S3               1   2427    2427   0.810 0.370271
## AGE:BMI:S5             1   3688    3688   1.231 0.269844
## AGE:BP:S5              1  11160   11160   3.726 0.056441 .
## BMI:BP:S5              1   1309    1309   0.437 0.510087
## AGE:S1:S5              1   7515    7515   2.509 0.116400
## BMI:S1:S5              1  23173   23173   7.737 0.006480 **
## BP:S1:S5               1   3932    3932   1.313 0.254694
## AGE:S2:S5              1    416     416   0.139 0.710188
## BMI:S2:S5              1  11092   11092   3.703 0.057179 .
## BP:S2:S5               1     92      92   0.031 0.861310
## S1:S2:S5               1      9       9   0.003 0.957614
## AGE:S3:S5              1     13      13   0.004 0.948480
## BMI:S3:S5              1   3067    3067   1.024 0.314091
## BP:S3:S5               1   7129    7129   2.380 0.126074
## S1:S3:S5               1   1311    1311   0.438 0.509794
## S2:S3:S5               1     91      91   0.030 0.861746
## AGE:BMI:S6             1     20      20   0.007 0.934821
## AGE:BP:S6              1    758     758   0.253 0.615936
## BMI:BP:S6              1      1       1   0.000 0.988486
## AGE:S1:S6              1   7772    7772   2.595 0.110397
## BMI:S1:S6              1    128     128   0.043 0.836704
## BP:S1:S6               1   2193    2193   0.732 0.394283
## AGE:S2:S6              1    217     217   0.073 0.788285
## BMI:S2:S6              1      0       0   0.000 0.998612
## BP:S2:S6               1    207     207   0.069 0.793402
## S1:S2:S6               1   1446    1446   0.483 0.488765
## AGE:S3:S6              1    246     246   0.082 0.775180
## BMI:S3:S6              1   6598    6598   2.203 0.140947
## BP:S3:S6               1   9932    9932   3.316 0.071631 .
## S1:S3:S6               1    547     547   0.183 0.670023
## S2:S3:S6               1   3286    3286   1.097 0.297486
## AGE:S5:S6              1   5080    5080   1.696 0.195853
```

```
## BMI:S5:S6              1    327     327   0.109 0.741668
## BP:S5:S6               1    224     224   0.075 0.785122
## S1:S5:S6               1   1429    1429   0.477 0.491414
## S2:S5:S6               1   5046    5046   1.685 0.197343
## S3:S5:S6               1     25      25   0.008 0.927733
## AGE:BMI:BP:S1          1    174     174   0.058 0.810231
## AGE:BMI:BP:S2          1   4509    4509   1.506 0.222736
## AGE:BMI:S1:S2          1     56      56   0.019 0.891595
## AGE:BP:S1:S2           1    452     452   0.151 0.698440
## BMI:BP:S1:S2           1    202     202   0.067 0.795564
## AGE:BMI:BP:S3          1    694     694   0.232 0.631409
## AGE:BMI:S1:S3          1   3934    3934   1.313 0.254560
## AGE:BP:S1:S3           1    459     459   0.153 0.696184
## BMI:BP:S1:S3           1   3389    3389   1.131 0.290048
## AGE:BMI:S2:S3          1    473     473   0.158 0.691930
## AGE:BP:S2:S3           1   8922    8922   2.979 0.087493 .
## BMI:BP:S2:S3           1    299     299   0.100 0.752688
## AGE:S1:S2:S3           1   4464    4464   1.490 0.225069
## BMI:S1:S2:S3           1   8022    8022   2.678 0.104910
## BP:S1:S2:S3            1   1029    1029   0.344 0.559078
## AGE:BMI:BP:S5          1    210     210   0.070 0.791508
## AGE:BMI:S1:S5          1   2977    2977   0.994 0.321222
## AGE:BP:S1:S5           1   2802    2802   0.936 0.335790
## BMI:BP:S1:S5           1   3643    3643   1.216 0.272745
## AGE:BMI:S2:S5          1     96      96   0.032 0.858016
## AGE:BP:S2:S5           1   4189    4189   1.399 0.239789
## BMI:BP:S2:S5           1   1560    1560   0.521 0.472130
## AGE:S1:S2:S5           1   3633    3633   1.213 0.273459
## BMI:S1:S2:S5           1   3148    3148   1.051 0.307754
## BP:S1:S2:S5            1    424     424   0.141 0.707683
## AGE:BMI:S3:S5          1    268     268   0.089 0.765510
## AGE:BP:S3:S5           1    645     645   0.215 0.643662
## BMI:BP:S3:S5           1   3993    3993   1.333 0.251055
## AGE:S1:S3:S5           1   3683    3683   1.230 0.270164
## BMI:S1:S3:S5           1     51      51   0.017 0.896038
## BP:S1:S3:S5            1    108     108   0.036 0.850044
## AGE:S2:S3:S5           1   4179    4179   1.395 0.240363
## BMI:S2:S3:S5           1   7586    7586   2.533 0.114705
## BP:S2:S3:S5            1   6304    6304   2.105 0.150023
## S1:S2:S3:S5            1  12891   12891   4.304 0.040624 *
## AGE:BMI:BP:S6          1   4623    4623   1.544 0.217032
## AGE:BMI:S1:S6          1   8599    8599   2.871 0.093337 .
## AGE:BP:S1:S6           1    410     410   0.137 0.712268
## BMI:BP:S1:S6           1   2995    2995   1.000 0.319810
## AGE:BMI:S2:S6          1    180     180   0.060 0.806952
## AGE:BP:S2:S6           1     34      34   0.011 0.915741
## BMI:BP:S2:S6           1   9743    9743   3.253 0.074342 .
## AGE:S1:S2:S6           1   3366    3366   1.124 0.291665
## BMI:S1:S2:S6           1   1383    1383   0.462 0.498411
## BP:S1:S2:S6            1     22      22   0.007 0.931540
## AGE:BMI:S3:S6          1    201     201   0.067 0.796198
## AGE:BP:S3:S6           1   4877    4877   1.628 0.204948
## BMI:BP:S3:S6           1   6383    6383   2.131 0.147528
## AGE:S1:S3:S6           1    148     148   0.050 0.824281
## BMI:S1:S3:S6           1   1352    1352   0.451 0.503319
## BP:S1:S3:S6            1  10654   10654   3.557 0.062231 .
## AGE:S2:S3:S6           1   8005    8005   2.672 0.105275
```

```
## BMI:S2:S3:S6           1   3519   3519   1.175 0.281033
## BP:S2:S3:S6            1   1496   1496   0.499 0.481444
## S1:S2:S3:S6            1   1997   1997   0.667 0.416119
## AGE:BMI:S5:S6          1    467    467   0.156 0.693922
## AGE:BP:S5:S6           1   2167   2167   0.724 0.397036
## BMI:BP:S5:S6           1     25     25   0.008 0.927013
## AGE:S1:S5:S6           1  12395  12395   4.138 0.044600 *
## BMI:S1:S5:S6           1   2869   2869   0.958 0.330102
## BP:S1:S5:S6            1  12000  12000   4.006 0.048071 *
## AGE:S2:S5:S6           1     13     13   0.004 0.948533
## BMI:S2:S5:S6           1   5392   5392   1.800 0.182764
## BP:S2:S5:S6            1   1925   1925   0.643 0.424635
## S1:S2:S5:S6            1   4975   4975   1.661 0.200494
## AGE:S3:S5:S6           1     52     52   0.017 0.895350
## BMI:S3:S5:S6           1    657    657   0.219 0.640482
## BP:S3:S5:S6            1   1263   1263   0.422 0.517577
## S1:S3:S5:S6            1   2787   2787   0.930 0.337095
## S2:S3:S5:S6            1   3948   3948   1.318 0.253713
## AGE:BMI:BP:S1:S2       1     12     12   0.004 0.948950
## AGE:BMI:BP:S1:S3       1   3183   3183   1.063 0.305121
## AGE:BMI:BP:S2:S3       1   4269   4269   1.425 0.235386
## AGE:BMI:S1:S2:S3       1     88     88   0.029 0.864395
## AGE:BP:S1:S2:S3        1     10     10   0.003 0.955105
## BMI:BP:S1:S2:S3        1    438    438   0.146 0.702982
## AGE:BMI:BP:S1:S5       1    915    915   0.305 0.581777
## AGE:BMI:BP:S2:S5       1   2448   2448   0.817 0.368182
## AGE:BMI:S1:S2:S5       1     32     32   0.011 0.917599
## AGE:BP:S1:S2:S5        1   1250   1250   0.417 0.519768
## BMI:BP:S1:S2:S5        1   3417   3417   1.141 0.288086
## AGE:BMI:BP:S3:S5       1   2949   2949   0.985 0.323483
## AGE:BMI:S1:S3:S5       1   1469   1469   0.490 0.485388
## AGE:BP:S1:S3:S5        1   2293   2293   0.766 0.383707
## BMI:BP:S1:S3:S5        1   1549   1549   0.517 0.473692
## AGE:BMI:S2:S3:S5       1   4066   4066   1.358 0.246754
## AGE:BP:S2:S3:S5        1      1      1   0.000 0.986737
## BMI:BP:S2:S3:S5        1   2910   2910   0.972 0.326703
## AGE:S1:S2:S3:S5        1   9545   9545   3.186 0.077312 .
## BMI:S1:S2:S3:S5        1  14151  14151   4.724 0.032122 *
## BP:S1:S2:S3:S5         1    351    351   0.117 0.732986
## AGE:BMI:BP:S1:S6       1   8841   8841   2.952 0.088920 .
## AGE:BMI:BP:S2:S6       1    364    364   0.122 0.728126
## AGE:BMI:S1:S2:S6       1   2743   2743   0.916 0.340930
## AGE:BP:S1:S2:S6        1   6501   6501   2.170 0.143859
## BMI:BP:S1:S2:S6        1     60     60   0.020 0.888023
## AGE:BMI:BP:S3:S6       1   9216   9216   3.077 0.082508 .
## AGE:BMI:S1:S3:S6       1    134    134   0.045 0.832823
## AGE:BP:S1:S3:S6        1      0      0   0.000 0.994720
## BMI:BP:S1:S3:S6        1   6837   6837   2.282 0.134035
## AGE:BMI:S2:S3:S6       1   6121   6121   2.044 0.156002
## AGE:BP:S2:S3:S6        1    367    367   0.123 0.727079
## BMI:BP:S2:S3:S6        1   2170   2170   0.725 0.396713
## AGE:S1:S2:S3:S6        1   4793   4793   1.600 0.208860
## BMI:S1:S2:S3:S6        1   2374   2374   0.793 0.375477
## BP:S1:S2:S3:S6         1   2223   2223   0.742 0.391102
## AGE:BMI:BP:S5:S6       1   1971   1971   0.658 0.419194
## AGE:BMI:S1:S5:S6       1   2908   2908   0.971 0.326882
## AGE:BP:S1:S5:S6        1   1285   1285   0.429 0.514077
```

```
## BMI:BP:S1:S5:S6          1   8467    8467   2.827 0.095862 .
## AGE:BMI:S2:S5:S6         1     90      90   0.030 0.862521
## AGE:BP:S2:S5:S6          1   3310    3310   1.105 0.295715
## BMI:BP:S2:S5:S6          1   3289    3289   1.098 0.297272
## AGE:S1:S2:S5:S6          1   2275    2275   0.759 0.385606
## BMI:S1:S2:S5:S6          1   1116    1116   0.373 0.543011
## BP:S1:S2:S5:S6           1   1984    1984   0.662 0.417714
## AGE:BMI:S3:S5:S6         1   5573    5573   1.860 0.175662
## AGE:BP:S3:S5:S6          1  14094   14094   4.705 0.032465 *
## BMI:BP:S3:S5:S6          1    471     471   0.157 0.692654
## AGE:S1:S3:S5:S6          1   1977    1977   0.660 0.418520
## BMI:S1:S3:S5:S6          1   3149    3149   1.051 0.307712
## BP:S1:S3:S5:S6           1    216     216   0.072 0.788850
## AGE:S2:S3:S5:S6          1   4627    4627   1.545 0.216834
## BMI:S2:S3:S5:S6          1   2493    2493   0.832 0.363782
## BP:S2:S3:S5:S6           1    125     125   0.042 0.838393
## S1:S2:S3:S5:S6           1     75      75   0.025 0.874342
## AGE:BMI:BP:S1:S2:S3      1   3923    3923   1.310 0.255236
## AGE:BMI:BP:S1:S2:S5      1    655     655   0.219 0.640987
## AGE:BMI:BP:S1:S3:S5      1    870     870   0.290 0.591185
## AGE:BMI:BP:S2:S3:S5      1   3895    3895   1.300 0.256897
## AGE:BMI:S1:S2:S3:S5      1   3525    3525   1.177 0.280623
## AGE:BP:S1:S2:S3:S5       1      6       6   0.002 0.965009
## BMI:BP:S1:S2:S3:S5       1   3021    3021   1.008 0.317730
## AGE:BMI:BP:S1:S2:S6      1    850     850   0.284 0.595373
## AGE:BMI:BP:S1:S3:S6      1   1430    1430   0.477 0.491196
## AGE:BMI:BP:S2:S3:S6      1    850     850   0.284 0.595404
## AGE:BMI:S1:S2:S3:S6      1    151     151   0.050 0.822883
## AGE:BP:S1:S2:S3:S6       1   2366    2366   0.790 0.376257
## BMI:BP:S1:S2:S3:S6       1   1426    1426   0.476 0.491756
## AGE:BMI:BP:S1:S5:S6      1   8223    8223   2.745 0.100708
## AGE:BMI:BP:S2:S5:S6      1    313     313   0.104 0.747254
## AGE:BMI:S1:S2:S5:S6      1   1794    1794   0.599 0.440781
## AGE:BP:S1:S2:S5:S6       1   1286    1286   0.429 0.513875
## BMI:BP:S1:S2:S5:S6       1   1020    1020   0.341 0.560784
## AGE:BMI:BP:S3:S5:S6      1      4       4   0.001 0.970736
## AGE:BMI:S1:S3:S5:S6      1   1828    1828   0.610 0.436524
## AGE:BP:S1:S3:S5:S6       1    125     125   0.042 0.838408
## BMI:BP:S1:S3:S5:S6       1   4108    4108   1.372 0.244343
## AGE:BMI:S2:S3:S5:S6      1    157     157   0.052 0.819297
## AGE:BP:S2:S3:S5:S6       1   1029    1029   0.343 0.559195
## BMI:BP:S2:S3:S5:S6       1    612     612   0.204 0.652249
## AGE:S1:S2:S3:S5:S6       1    844     844   0.282 0.596789
## BMI:S1:S2:S3:S5:S6       1   1024    1024   0.342 0.560139
## BP:S1:S2:S3:S5:S6        1   4609    4609   1.539 0.217761
## AGE:BMI:BP:S1:S2:S3:S5   1   1048    1048   0.350 0.555534
## AGE:BMI:BP:S1:S2:S3:S6   1   1086    1086   0.362 0.548547
## AGE:BMI:BP:S1:S2:S5:S6   1     31      31   0.010 0.918828
## AGE:BMI:BP:S1:S3:S5:S6   1    723     723   0.241 0.624329
## AGE:BMI:S1:S2:S3:S5:S6   1   1466    1466   0.490 0.485757
## AGE:BP:S1:S2:S3:S5:S6    1   7632    7632   2.548 0.113632
## BMI:BP:S1:S2:S3:S5:S6    1   3459    3459   1.155 0.285175
## Residuals               99 296537    2995
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
#Build Models
  #SLR
    #train
    fit <- lm(formula = Y~BMI,data=train)
    y.train.pred <- predict(fit,newdata=x.train)
    summary(fit)
```

```
##
## Call:
## lm(formula = Y ~ BMI, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -162.809  -43.569   -7.261   48.156  152.338
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -107.096     20.376  -5.256 2.55e-07 ***
## BMI            9.846      0.764  12.887  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 62.89 on 353 degrees of freedom
## Multiple R-squared:  0.3199, Adjusted R-squared:  0.318
## F-statistic: 166.1 on 1 and 353 DF,  p-value: < 2.2e-16
```

```r
    RMSE(y.train.pred,y.train)#62.7097
```

```
## [1] 62.7097
```

```r
    #test
    y.test.pred <- predict(fit,newdata=x.test)
    predict <- lm(y.test~y.test.pred)
    summary(predict)
```

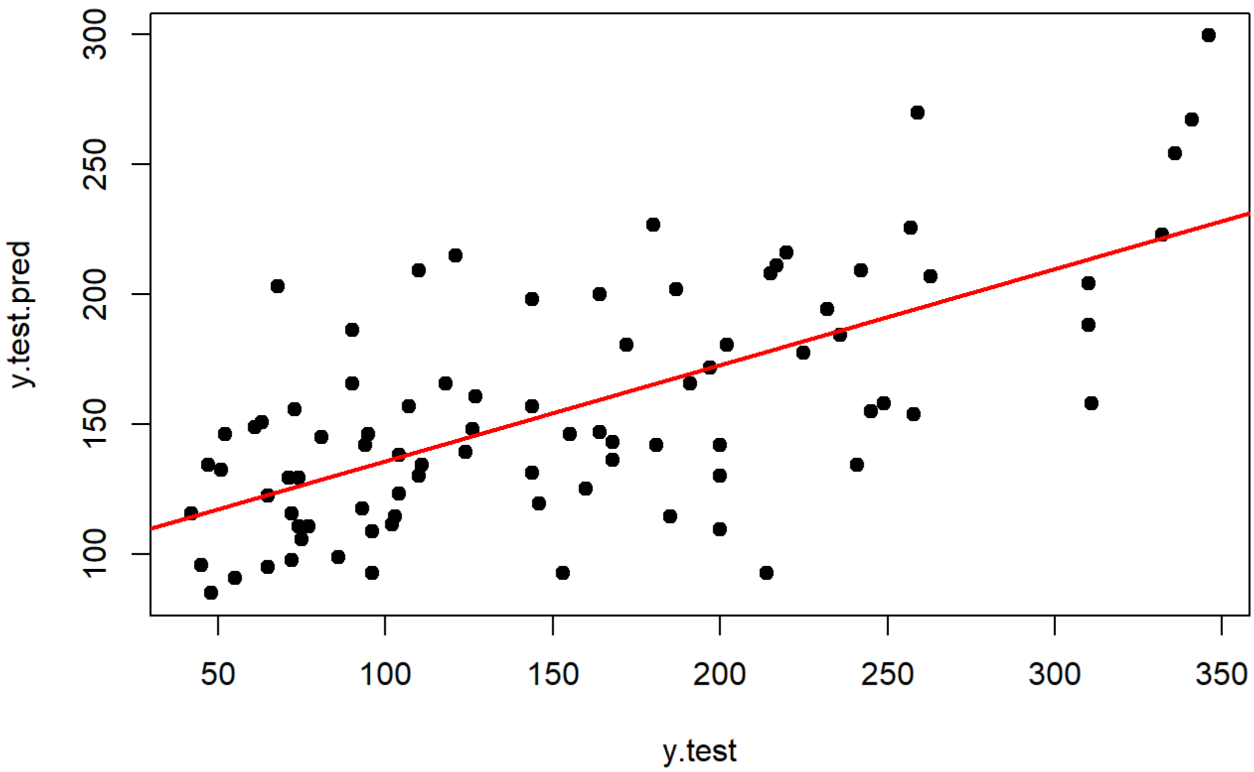
```
##
## Call:
## lm(formula = y.test ~ y.test.pred)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -141.74  -42.70  -10.73   36.79  155.10
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -31.6173    23.5179  -1.344    0.182
## y.test.pred   1.1887     0.1455   8.172 2.55e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 61.18 on 85 degrees of freedom
```

```
## Multiple R-squared:   0.44,  Adjusted R-squared:  0.4334
## F-statistic: 66.78 on 1 and 85 DF,  p-value: 2.551e-12
```

```
RMSE(y.test.pred,y.test)#61.11294
```

```
## [1] 61.11294
```

```
plot(y.test,y.test.pred, pch = 19, cex = 1, col = "black")
abline(lm(y.test.pred~y.test),col='red',lwd=2)
```




```
#multiple linear regression
  #train
  fit <- lm(formula = Y~SEX+BMI+BP+S1+S2+S5+S6,data=train)
  y.train.pred <- predict(fit,newdata=x.train)
  summary(fit)
```

```
##
## Call:
## lm(formula = Y ~ SEX + BMI + BP + S1 + S2 + S5 + S6, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -152.851  -38.319   -0.824   36.942  149.420
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -320.7058    29.6318 -10.823  < 2e-16 ***
## SEX         -21.9590     6.5182  -3.369  0.00084 ***
## BMI           4.9575     0.8188   6.055 3.65e-09 ***
## BP            1.0507     0.2497   4.208 3.29e-05 ***
## S1           -1.0232     0.2523  -4.055 6.19e-05 ***
## S2            0.8703     0.2622   3.319  0.00100 **
## S5           71.9213     8.7834   8.188 5.12e-15 ***
## S6            0.3715     0.2944   1.262  0.20784
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54.54 on 347 degrees of freedom
## Multiple R-squared:  0.4972, Adjusted R-squared:  0.4871
## F-statistic: 49.03 on 7 and 347 DF,  p-value: < 2.2e-16
```

```
RMSE(y.train.pred,y.train)#53.91863
```

```
## [1] 53.91863
```

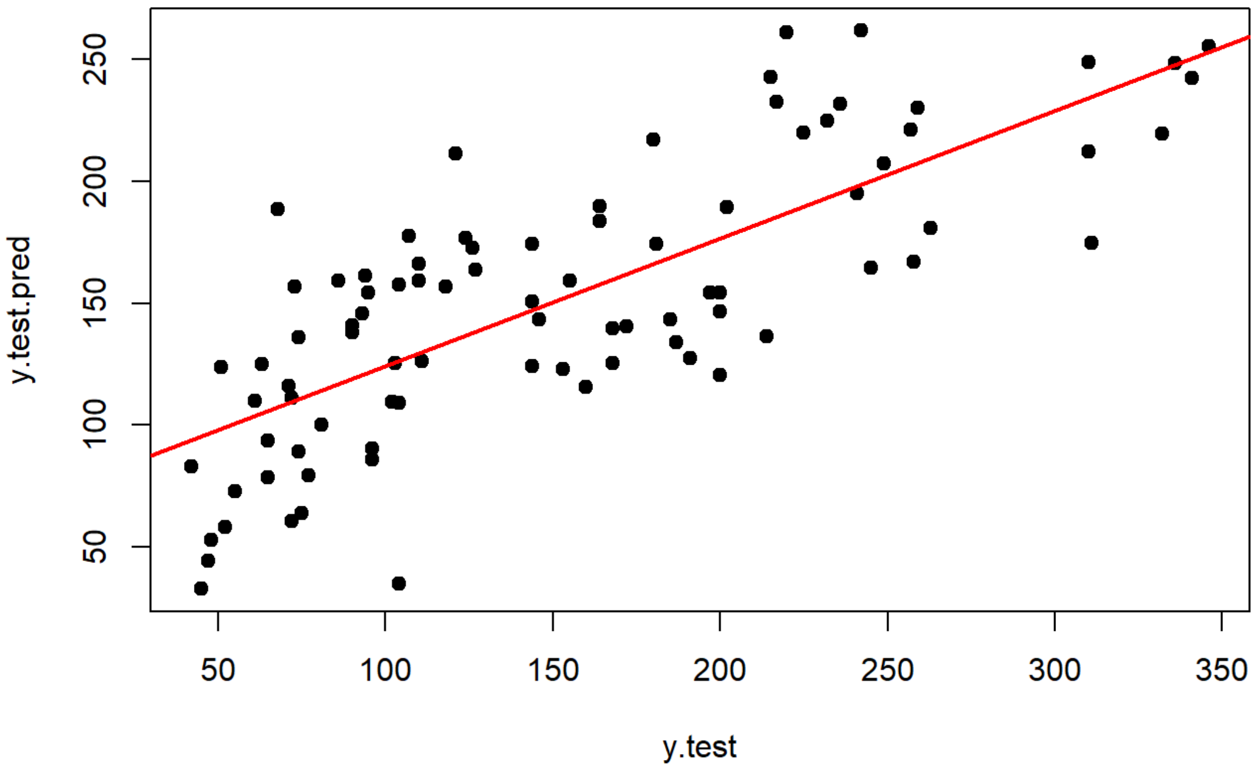
```
#test
y.test.pred <- predict(fit,newdata=x.test)
predict <- lm(y.test~y.test.pred)
summary(predict)
```

```
##
## Call:
## lm(formula = y.test ~ y.test.pred)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -125.766  -41.293   -3.789   36.917  132.700
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -16.311     16.485  -0.989    0.325
## y.test.pred    1.115      0.102  10.929   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52.71 on 85 degrees of freedom
## Multiple R-squared:  0.5843, Adjusted R-squared:  0.5794
## F-statistic: 119.5 on 1 and 85 DF,  p-value: < 2.2e-16
```

```
RMSE(y.test.pred,y.test)#52.50263
```

```
## [1] 52.50263
```

```
plot(y.test,y.test.pred, pch = 19, cex = 1, col = "black")
abline(lm(y.test.pred~y.test),col='red',lwd=2)
```




```
#polynomial regression
  #train
  fit <- lm(formula = Y~AGE+BP+BMI+S3+SEX +
                      I(AGE^2)+I(S5^2)+I(S3^2)+
                      I(AGE*SEX)+
                      I(BMI*BP) + I(S1*S2*S3*S5)+
                      I(BMI*S1*S5) + I(BMI*S2*S5)+
                      I(AGE*S2*S3*S6)+
                      I(BMI*S1*S2*S3*S5) + I(AGE*BP*S3*S5*S6),
            data=train)
  y.train.pred <- predict(fit,newdata=x.train)
  summary(fit)
```

```
##
## Call:
## lm(formula = Y ~ AGE + BP + BMI + S3 + SEX + I(AGE^2) + I(S5^2) +
##     I(S3^2) + I(AGE * SEX) + I(BMI * BP) + I(S1 * S2 * S3 * S5) +
##     I(BMI * S1 * S5) + I(BMI * S2 * S5) + I(AGE * S2 * S3 * S6) +
##     I(BMI * S1 * S2 * S3 * S5) + I(AGE * BP * S3 * S5 * S6),
##     data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -141.774  -37.239   -4.278   34.056  142.793
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   5.822e+02  1.603e+02   3.631 0.000326 ***
```

```
## AGE                          -4.194e+00  1.497e+00  -2.802 0.005378 **
## BP                           -3.800e+00  1.333e+00  -2.851 0.004630 **
## BMI                          -9.096e+00  4.660e+00  -1.952 0.051797 .
## S3                           -4.029e+00  1.788e+00  -2.254 0.024850 *
## SEX                          -7.898e+01  2.348e+01  -3.363 0.000858 ***
## I(AGE^2)                      1.928e-02  1.490e-02   1.294 0.196462
## I(S5^2)                       6.541e+00  2.762e+00   2.368 0.018431 *
## I(S3^2)                       2.170e-02  1.296e-02   1.674 0.095126 .
## I(AGE * SEX)                  1.103e+00  4.586e-01   2.405 0.016708 *
## I(BMI * BP)                   1.473e-01  4.599e-02   3.203 0.001491 **
## I(S1 * S2 * S3 * S5)         -1.270e-05  9.678e-06  -1.312 0.190352
## I(BMI * S1 * S5)             -8.057e-03  3.735e-03  -2.157 0.031720 *
## I(BMI * S2 * S5)              7.455e-03  4.012e-03   1.858 0.064044 .
## I(AGE * S2 * S3 * S6)        -1.541e-06  1.210e-06  -1.273 0.203986
## I(BMI * S1 * S2 * S3 * S5)  6.639e-07  3.585e-07   1.852 0.064908 .
## I(AGE * BP * S3 * S5 * S6)  8.688e-07  3.069e-07   2.831 0.004916 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52.69 on 338 degrees of freedom
## Multiple R-squared:  0.543,  Adjusted R-squared:  0.5213
## F-statistic:  25.1 on 16 and 338 DF,  p-value: < 2.2e-16
```

```
RMSE(y.train.pred,y.train)#51.4093
```

```
## [1] 51.4093
```

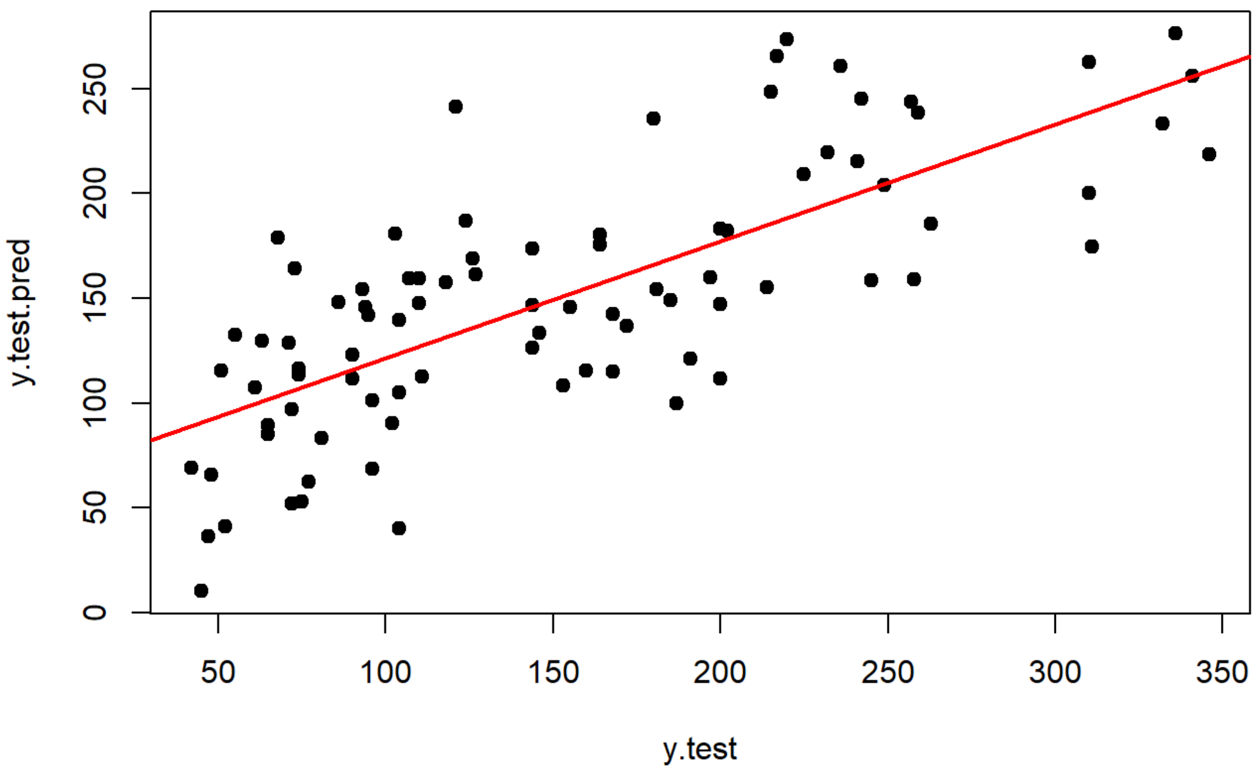
```
#test
y.test.pred <- predict(fit,newdata=x.test)
predict <- lm(y.test~y.test.pred)
summary(predict)
```

```
##
## Call:
## lm(formula = y.test ~ y.test.pred)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -122.067  -41.691   -3.283   33.429  134.537
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.83478   15.70374   0.181    0.857
## y.test.pred  0.99529    0.09664  10.299   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54.53 on 85 degrees of freedom
## Multiple R-squared:  0.5551, Adjusted R-squared:  0.5499
## F-statistic: 106.1 on 1 and 85 DF,  p-value: < 2.2e-16
```

```
RMSE(y.test.pred,y.test)#53.94088
```

```
## [1] 53.94088
```

```
plot(y.test,y.test.pred, pch = 19, cex = 1, col = "black")
abline(lm(y.test.pred~y.test),col='red',lwd=2)
```




```
#ridge (alpha = 0)
set.seed(123)
ridge.fit <- cv.glmnet(as.matrix(x.train),y.train,type.measure ='mse',alpha=0,fa
mily='gaussian')
ridge.fit$lambda.1se
```

```
## [1] 63.87225
```

```
coef(ridge.fit)
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept) -148.67220469
## AGE            0.12349837
## SEX           -8.51972205
## BMI            3.27742361
## BP             0.72694871
## S1             0.02000633
## S2            -0.03983784
## S3            -0.54474454
```

```
## S4              4.07225690
## S5             25.39873274
## S6              0.49832477
```

```
    #train
    ridge.predict.train <- predict(ridge.fit,s=ridge.fit$lambda.1se,newx = as.matr
ix(x.train))
    fit <- lm(y.train~ridge.predict.train)
    summary(fit)
```

```
##
## Call:
## lm(formula = y.train ~ ridge.predict.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -130.410  -41.102   -0.567   40.395  160.876
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -50.35408   11.66405  -4.317 2.06e-05 ***
## ridge.predict.train   1.33142    0.07431  17.917  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55.19 on 353 degrees of freedom
## Multiple R-squared:  0.4763, Adjusted R-squared:  0.4748
## F-statistic:   321 on 1 and 353 DF,  p-value: < 2.2e-16
```

```
    RMSE(ridge.predict.train,y.train)#56.56072
```

```
## [1] 56.56072
```

```
    #test
    ridge.predict.test <- predict(ridge.fit,s=ridge.fit$lambda.1se,newx = as.matri
x(x.test))
    predict <- lm(y.test~ridge.predict.test)
    summary(predict)
```

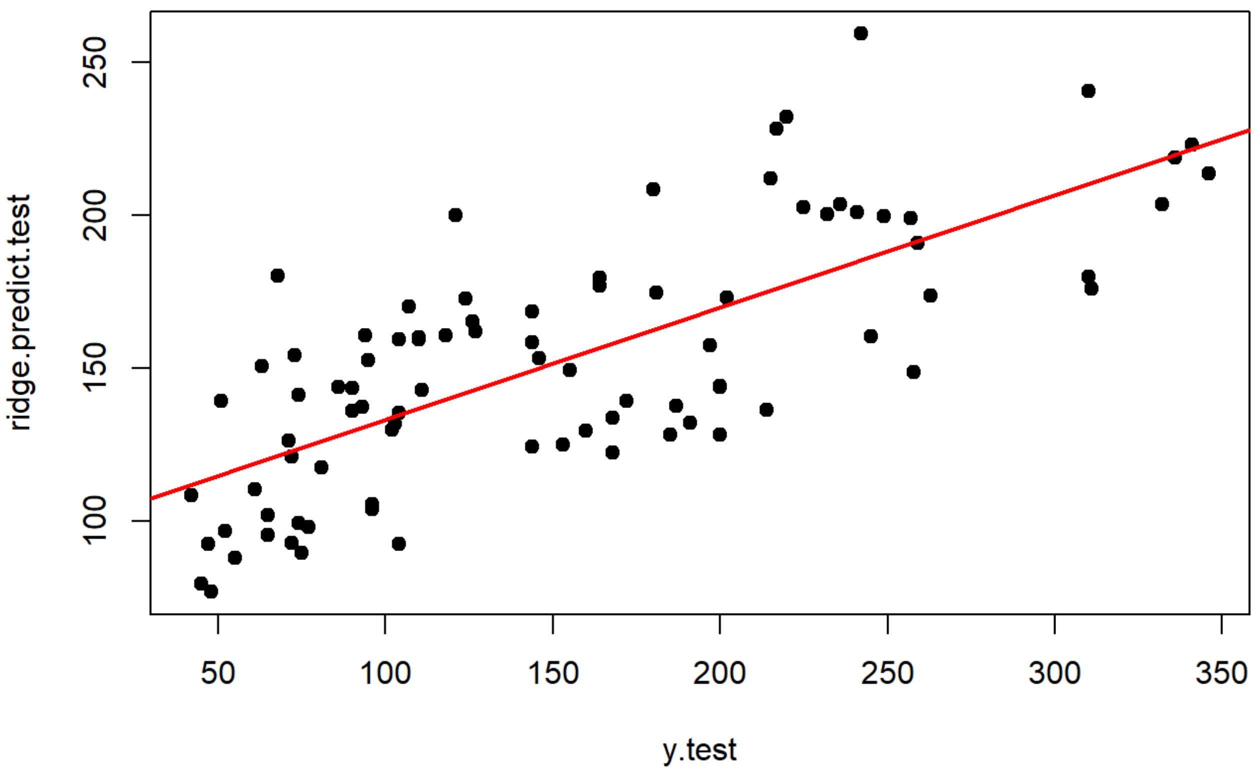
```
##
## Call:
## lm(formula = y.test ~ ridge.predict.test)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -125.346  -44.308   -3.681   39.165  123.582
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -71.9704    23.2521  -3.095  0.00266 **
## ridge.predict.test   1.4742     0.1473  10.005 5.03e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55.4 on 85 degrees of freedom
## Multiple R-squared:  0.5408, Adjusted R-squared:  0.5354
## F-statistic: 100.1 on 1 and 85 DF,  p-value: 5.028e-16
```

```
RMSE(ridge.predict.test,y.test)#58.00265
```

```
## [1] 58.00265
```

```
#plot
plot(y.test,ridge.predict.test, pch = 19, cex = 1, col = "black")
abline(lm(ridge.predict.test~y.test),col='red',lwd=2)
```




```
#lasso (alpha = 1)
set.seed(123)
lasso.fit <- cv.glmnet(as.matrix(x.train),y.train,type.measure ='mse',alpha=1,fa
mily='gaussian')
lasso.fit$lambda.1se
```

```
## [1] 8.845581
```

```
coef(lasso.fit)
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept) -198.2166275
## AGE            .
## SEX            .
## BMI            4.7320556
## BP             0.5720361
## S1             .
## S2             .
## S3            -0.2404157
## S4             .
## S5            39.4848485
## S6             .
```

```
    #summary train
    lasso.predict.train <- predict(lasso.fit,s=lasso.fit$lambda.1se,newx = as.matr
ix(x.train))
    fit <- lm(y.train~lasso.predict.train)
    summary(fit)
```

```
##
## Call:
## lm(formula = y.train ~ lasso.predict.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -139.201  -39.521   -0.881   40.409  144.993
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -42.0800    11.3590  -3.705 0.000246 ***
## lasso.predict.train   1.2770     0.0722  17.686  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55.53 on 353 degrees of freedom
## Multiple R-squared:  0.4698, Adjusted R-squared:  0.4683
## F-statistic: 312.8 on 1 and 353 DF,  p-value: < 2.2e-16
```

```
    RMSE(lasso.predict.train,y.train)#56.51291
```

```
## [1] 56.51291
```

```
    #summary test
    lasso.predict.test <- predict(lasso.fit,s=lasso.fit$lambda.1se,newx = as.matri
x(x.test))
    predict <- lm(y.test~lasso.predict.test)
    summary(predict)
```

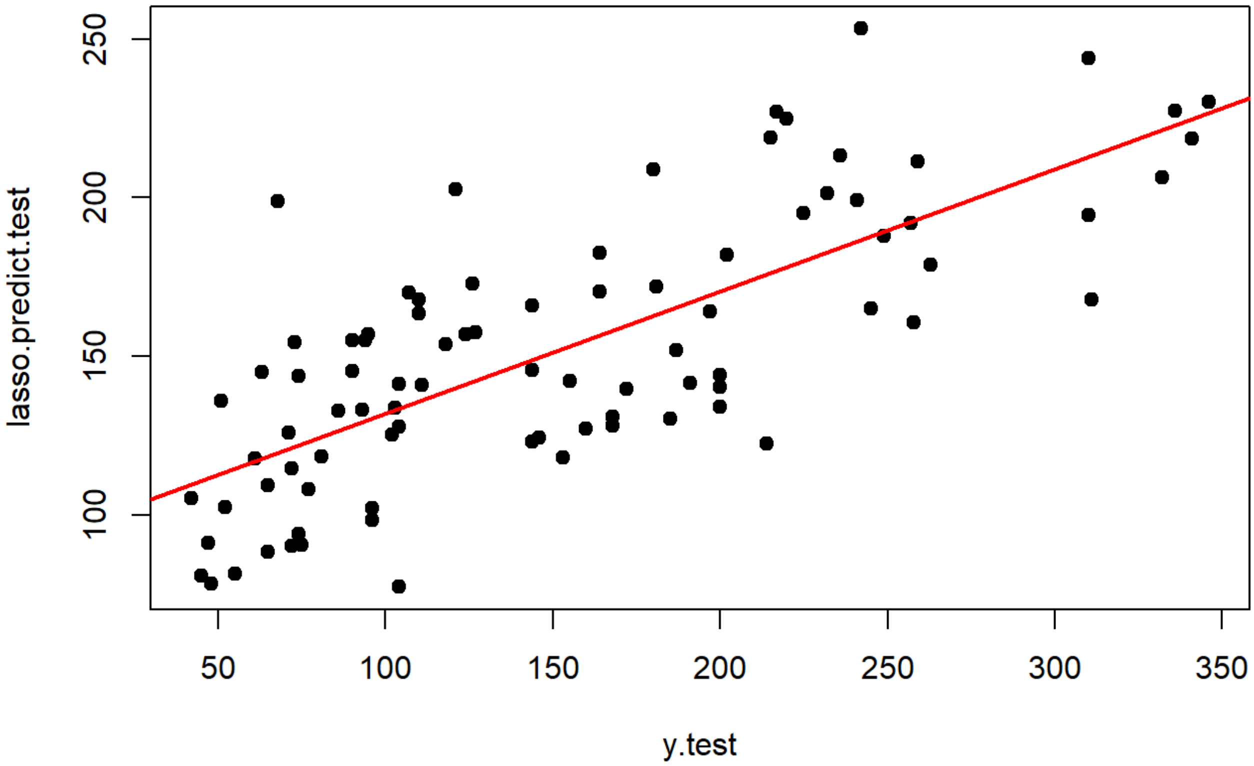
```
##
## Call:
## lm(formula = y.test ~ lasso.predict.test)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -151.854  -36.884   -0.261   40.122  135.796
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -66.1576    22.1202  -2.991  0.00364 **
## lasso.predict.test   1.4374     0.1399  10.271  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54.61 on 85 degrees of freedom
## Multiple R-squared:  0.5538, Adjusted R-squared:  0.5486
## F-statistic: 105.5 on 1 and 85 DF,  p-value: < 2.2e-16
```

```
RMSE(lasso.predict.test,y.test)#56.99875
```

```
## [1] 56.99875
```

```
#plot
plot(y.test,lasso.predict.test, pch = 19, cex = 1, col = "black")
abline(lm(lasso.predict.test~y.test),col='red',lwd=2)
```




```
#elastic net
results.train <-data.frame()
for (i in 0:20)
```

```
  {
    set.seed(123)
    fit <- cv.glmnet(as.matrix(x.train), y.train, type.measure="mse", alpha=i/20,
                     family="gaussian")
    y.pred <- predict(fit, s=fit$lambda.1se, newx=as.matrix(x.train))
    predict <- lm(y.train~y.pred)

    temp <- data.frame(alpha=i/20,R2= summary(predict)$r.squared,Adj_R2=summary(pr
edict)$adj.r.squared,rmse=RMSE(y.pred,y.train),lambda=fit$lambda.1se)
    results.train <- rbind(results.train, temp)
  }
  #alpha = 0.05 (best adj R2)
  set.seed(123)
  elastic.fit <- cv.glmnet(as.matrix(x.train), y.train, type.measure="mse", alpha=
0.05,
                   family="gaussian")
  elastic.fit$lambda.1se
```

```
## [1] 48.09499
```

```
  coef(fit)
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept) -198.2166275
## AGE            .
## SEX            .
## BMI            4.7320556
## BP             0.5720361
## S1             .
## S2             .
## S3            -0.2404157
## S4             .
## S5            39.4848485
## S6             .
```

```
    #train
    elastic.predict.train <- predict(elastic.fit, s=elastic.fit$lambda.1se, newx=a
s.matrix(x.train))
    fit <- lm(y.train~ elastic.predict.train)
    summary(fit)
```

```
##
## Call:
## lm(formula = y.train ~ elastic.predict.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -131.996  -41.093   -0.335   41.231  155.550
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            -50.98749   11.67363  -4.368 1.65e-05 ***
```

```
## elastic.predict.train   1.33559    0.07438  17.956  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55.13 on 353 degrees of freedom
## Multiple R-squared:  0.4774, Adjusted R-squared:  0.4759
## F-statistic: 322.4 on 1 and 353 DF,  p-value: < 2.2e-16
```

```
RMSE(elastic.predict.train,y.train)#56.53718
```

```
## [1] 56.53718
```

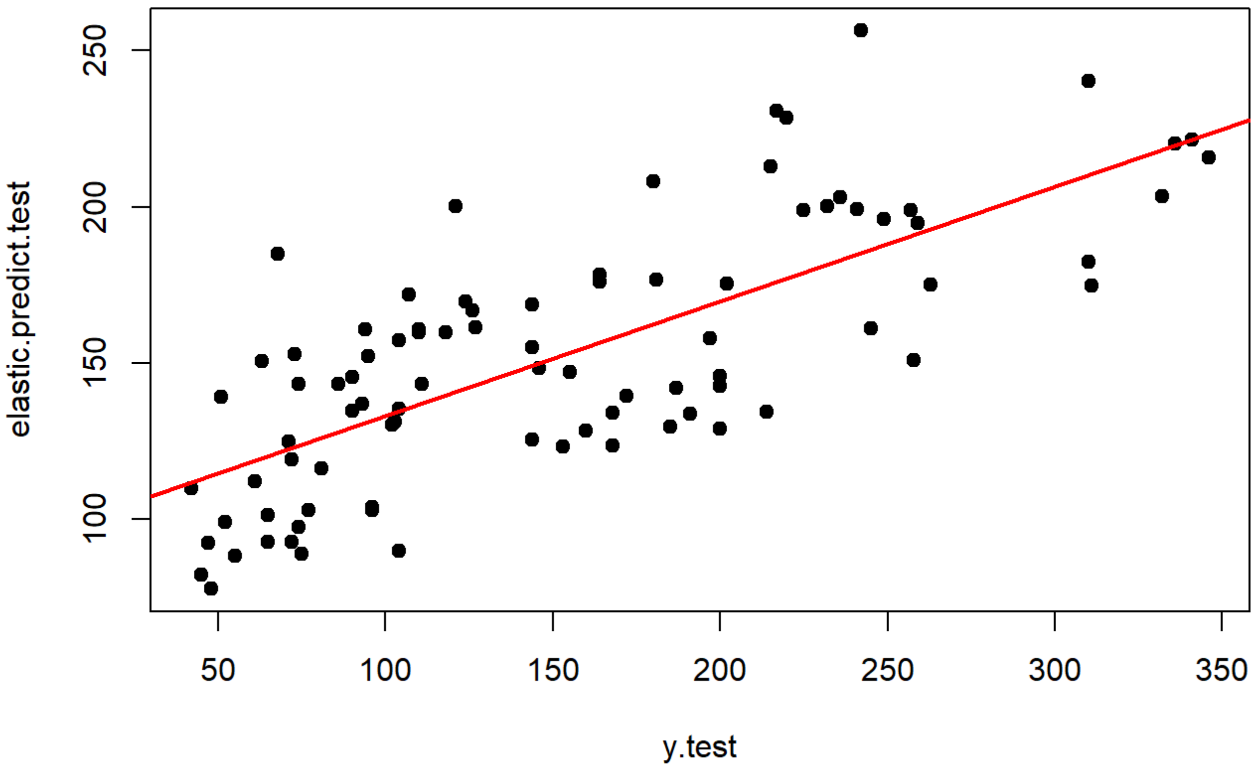
```
#test
elastic.predict.test <- predict(elastic.fit, s=elastic.fit$lambda.1se, newx=a
s.matrix(x.test))
predict <- lm(y.test~elastic.predict.test)
summary(predict)
```

```
##
## Call:
## lm(formula = y.test ~ elastic.predict.test)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -132.744  -43.152   -2.361   40.418  125.093
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -73.1442    23.1805  -3.155  0.00222 **
## elastic.predict.test   1.4820     0.1469  10.087 3.43e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55.16 on 85 degrees of freedom
## Multiple R-squared:  0.5448, Adjusted R-squared:  0.5395
## F-statistic: 101.7 on 1 and 85 DF,  p-value: 3.432e-16
```

```
RMSE(elastic.predict.test,y.test)#57.86804
```

```
## [1] 57.86804
```

```
plot(y.test, elastic.predict.test, pch = 19, cex = 1, col = "black")
abline(lm(elastic.predict.test~y.test),col='red',lwd=2)
```


elastic.predict.test
y.test
100
150
200
250
50
100
150
200
250
300
350

Phân tích ảnh hưởng, tương tác của các yếu tố và xây dựng mô hình hồi quy (R)

2. Trên bộ dữ liệu sau khi xử lý

```
#library
library(kernlab)
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:kernlab':
##
##     alpha
```

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-2
```

```
# Doc du lieu
df <- read.csv('diabetes.tab.tsv',header=TRUE,sep = '\t')

#preprocess

df_p = df
for( i in names(df_p))
{
  if(i!='SEX' & i!='Y')
  {
    Q3 = quantile(df_p[,i], 0.75)
    Q1 = quantile(df_p[,i], 0.25)
    IQR = Q3 - Q1
    olr_up = Q3+1.5*IQR
    olr_low = Q1-1.5*IQR

    df_p<-df_p[!(df_p[,i] > olr_up), ]
    df_p<-df_p[!(df_p[,i] < olr_low), ]
  }
```

```
}

#chia train-test
  set.seed(123)
  train.index <- createDataPartition(df_p$Y, p = .8, list = FALSE)

  train_p <- df_p[ train.index,]
  test_p  <- df_p[-train.index,]

  x.train_p <- train_p[,1:10]
  x.test_p <- test_p[,1:10]
  y.train_p <- train_p[,11]
  y.test_p <- test_p[,11]

  RMSE = function(m, o){
    sqrt(mean((m - o)^2))
  }

  # Effect of Factors
  av <- aov(Y~.,data=train_p)
  summary(av)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)    
## AGE           1  70622   70622  24.021 1.53e-06 ***
## SEX           1      2       2   0.001  0.98169    
## BMI           1 464953  464953 158.147  < 2e-16 ***
## BP            1  91331   91331  31.065 5.35e-08 ***
## S1            1   1124    1124   0.382  0.53685    
## S2            1    746     746   0.254  0.61481    
## S3            1 223592  223592  76.052  < 2e-16 ***
## S4            1    358     358   0.122  0.72744    
## S5            1  20759   20759   7.061  0.00828 ** 
## S6            1    661     661   0.225  0.63580    
## Residuals   316 929043    2940                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
  av <- aov(Y~(AGE+BMI+BP+S1+S2+S3+S5+S6)*S4,data=train_p)
  summary(av)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)    
## AGE           1  70622   70622  23.338 2.14e-06 ***
## BMI           1 464743  464743 153.583  < 2e-16 ***
## BP            1  86284   86284  28.514 1.81e-07 ***
## S1            1   1359    1359   0.449  0.50329    
## S2            1   2053    2053   0.678  0.41079    
## S3            1 192796  192796  63.713 2.85e-14 ***
## S5            1  22091   22091   7.301  0.00727 ** 
## S6            1    369     369   0.122  0.72731    
## S4            1    345     345   0.114  0.73595    
## AGE:S4        1   6128    6128   2.025  0.15574    
## BMI:S4        1   5028    5028   1.662  0.19836    
## BP:S4         1    268     268   0.088  0.76637    
## S1:S4         1     14      14   0.005  0.94559    
```

```
## S2:S4          1     10      10   0.003  0.95391
## S3:S4          1   1404    1404   0.464  0.49630
## S5:S4          1   4644    4644   1.535  0.21635
## S6:S4          1   9999    9999   3.304  0.07007 .
## Residuals    309 935035    3026
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
av <- aov(Y~(AGE+BMI+BP+S1+S2+S3+S5+S6)*SEX,data=train_p)
summary(av)
```

```
##               Df Sum Sq Mean Sq F value   Pr(>F)
## AGE            1  70622   70622  24.333 1.33e-06 ***
## BMI            1 464743  464743 160.132  < 2e-16 ***
## BP             1  86284   86284  29.730 1.02e-07 ***
## S1             1   1359    1359   0.468 0.494335
## S2             1   2053    2053   0.707 0.401000
## S3             1 192796  192796  66.430 9.13e-15 ***
## S5             1  22091   22091   7.612 0.006145 **
## S6             1    369     369   0.127 0.721790
## SEX            1  32870   32870  11.326 0.000861 ***
## AGE:SEX        1  18617   18617   6.415 0.011813 *
## BMI:SEX        1   6800    6800   2.343 0.126872
## BP:SEX         1    519     519   0.179 0.672724
## S1:SEX         1    614     614   0.212 0.645764
## S2:SEX         1   1687    1687   0.581 0.446379
## S3:SEX         1   2718    2718   0.936 0.333951
## S5:SEX         1    157     157   0.054 0.816196
## S6:SEX         1   2096    2096   0.722 0.396099
## Residuals    309 896796    2902
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Effect of interaction
av <- aov(Y~AGE*BMI*BP*S1*S2*S3*S5*S6+AGE*SEX,data=train_p)
summary(av)
```

```
##                         Df Sum Sq Mean Sq F value   Pr(>F)
## AGE                      1  70622   70622  23.261 7.81e-06 ***
## BMI                      1 464743  464743 153.076  < 2e-16 ***
## BP                       1  86284   86284  28.420 1.10e-06 ***
## S1                       1   1359    1359   0.448  0.50567
## S2                       1   2053    2053   0.676  0.41368
## S3                       1 192796  192796  63.503 1.91e-11 ***
## S5                       1  22091   22091   7.276  0.00872 **
## S6                       1    369     369   0.121  0.72853
## SEX                      1  32870   32870  10.827  0.00156 **
## AGE:BMI                  1   4782    4782   1.575  0.21357
## AGE:BP                   1    147     147   0.048  0.82652
## BMI:BP                   1  21990   21990   7.243  0.00887 **
## AGE:S1                   1    293     293   0.096  0.75700
## BMI:S1                   1   1316    1316   0.433  0.51244
## BP:S1                    1    129     129   0.043  0.83726
## AGE:S2                   1    875     875   0.288  0.59298
```

```
## BMI:S2               1   5232    5232   1.723  0.19351
## BP:S2                1    347     347   0.114  0.73628
## S1:S2                1    159     159   0.052  0.81972
## AGE:S3               1     92      92   0.030  0.86254
## BMI:S3               1    728     728   0.240  0.62597
## BP:S3                1   3345    3345   1.102  0.29743
## S1:S3                1     25      25   0.008  0.92727
## S2:S3                1    226     226   0.075  0.78568
## AGE:S5               1  10836   10836   3.569  0.06294 .
## BMI:S5               1   4246    4246   1.399  0.24091
## BP:S5                1   2439    2439   0.803  0.37317
## S1:S5                1      2       2   0.001  0.97820
## S2:S5                1   1431    1431   0.471  0.49461
## S3:S5                1   3165    3165   1.043  0.31068
## AGE:S6               1  10106   10106   3.329  0.07229 .
## BMI:S6               1    245     245   0.081  0.77732
## BP:S6                1   2123    2123   0.699  0.40582
## S1:S6                1   3629    3629   1.195  0.27798
## S2:S6                1    540     540   0.178  0.67438
## S3:S6                1    980     980   0.323  0.57175
## S5:S6                1      6       6   0.002  0.96502
## AGE:SEX              1  27985   27985   9.218  0.00335 **
## AGE:BMI:BP           1   3784    3784   1.246  0.26801
## AGE:BMI:S1           1   1301    1301   0.428  0.51485
## AGE:BP:S1            1    889     889   0.293  0.59019
## BMI:BP:S1            1     88      88   0.029  0.86528
## AGE:BMI:S2           1   2205    2205   0.726  0.39694
## AGE:BP:S2            1   4533    4533   1.493  0.22580
## BMI:BP:S2            1     40      40   0.013  0.90874
## AGE:S1:S2            1    989     989   0.326  0.57005
## BMI:S1:S2            1    283     283   0.093  0.76114
## BP:S1:S2             1    167     167   0.055  0.81545
## AGE:BMI:S3           1      1       1   0.000  0.98235
## AGE:BP:S3            1      5       5   0.002  0.96802
## BMI:BP:S3            1     34      34   0.011  0.91585
## AGE:S1:S3            1   1150    1150   0.379  0.54023
## BMI:S1:S3            1    262     262   0.086  0.76981
## BP:S1:S3             1   5144    5144   1.694  0.19725
## AGE:S2:S3            1      8       8   0.003  0.96014
## BMI:S2:S3            1    116     116   0.038  0.84538
## BP:S2:S3             1   1472    1472   0.485  0.48858
## S1:S2:S3             1   5631    5631   1.855  0.17754
## AGE:BMI:S5           1   2467    2467   0.813  0.37037
## AGE:BP:S5            1   2191    2191   0.722  0.39849
## BMI:BP:S5            1   1838    1838   0.605  0.43913
## AGE:S1:S5            1    277     277   0.091  0.76332
## BMI:S1:S5            1   7448    7448   2.453  0.12174
## BP:S1:S5             1    394     394   0.130  0.71958
## AGE:S2:S5            1   1360    1360   0.448  0.50545
## BMI:S2:S5            1   8731    8731   2.876  0.09430 .
## BP:S2:S5             1   1375    1375   0.453  0.50308
## S1:S2:S5             1   3371    3371   1.110  0.29555
## AGE:S3:S5            1    534     534   0.176  0.67631
## BMI:S3:S5            1   6886    6886   2.268  0.13651
## BP:S3:S5             1   5528    5528   1.821  0.18150
## S1:S3:S5             1   2012    2012   0.663  0.41838
## S2:S3:S5             1   7123    7123   2.346  0.13004
```

```
## AGE:BMI:S6              1    328    328   0.108  0.74341
## AGE:BP:S6               1    131    131   0.043  0.83624
## BMI:BP:S6               1    582    582   0.192  0.66290
## AGE:S1:S6               1   3894   3894   1.282  0.26125
## BMI:S1:S6               1    680    680   0.224  0.63748
## BP:S1:S6                1   1473   1473   0.485  0.48842
## AGE:S2:S6               1    324    324   0.107  0.74501
## BMI:S2:S6               1     33     33   0.011  0.91735
## BP:S2:S6                1    286    286   0.094  0.75986
## S1:S2:S6                1    526    526   0.173  0.67844
## AGE:S3:S6               1     17     17   0.006  0.94089
## BMI:S3:S6               1  15662  15662   5.159  0.02616 *
## BP:S3:S6                1    476    476   0.157  0.69340
## S1:S3:S6                1     70     70   0.023  0.88011
## S2:S3:S6                1    551    551   0.181  0.67146
## AGE:S5:S6               1     74     74   0.024  0.87636
## BMI:S5:S6               1    452    452   0.149  0.70087
## BP:S5:S6                1    645    645   0.213  0.64622
## S1:S5:S6                1   2583   2583   0.851  0.35943
## S2:S5:S6                1     90     90   0.029  0.86413
## S3:S5:S6                1   1490   1490   0.491  0.48585
## AGE:BMI:BP:S1           1   1733   1733   0.571  0.45250
## AGE:BMI:BP:S2           1   2922   2922   0.962  0.32993
## AGE:BMI:S1:S2           1    351    351   0.115  0.73498
## AGE:BP:S1:S2            1   1650   1650   0.544  0.46340
## BMI:BP:S1:S2            1    364    364   0.120  0.73034
## AGE:BMI:BP:S3           1     49     49   0.016  0.89894
## AGE:BMI:S1:S3           1  14307  14307   4.712  0.03329 *
## AGE:BP:S1:S3            1   1643   1643   0.541  0.46437
## BMI:BP:S1:S3            1   4256   4256   1.402  0.24035
## AGE:BMI:S2:S3           1   2836   2836   0.934  0.33705
## AGE:BP:S2:S3            1    238    238   0.078  0.78024
## BMI:BP:S2:S3            1   2183   2183   0.719  0.39935
## AGE:S1:S2:S3            1    483    483   0.159  0.69111
## BMI:S1:S2:S3            1   2180   2180   0.718  0.39968
## BP:S1:S2:S3             1    173    173   0.057  0.81225
## AGE:BMI:BP:S5           1    900    900   0.296  0.58790
## AGE:BMI:S1:S5           1    450    450   0.148  0.70140
## AGE:BP:S1:S5            1   2425   2425   0.799  0.37450
## BMI:BP:S1:S5            1   2912   2912   0.959  0.33074
## AGE:BMI:S2:S5           1   2675   2675   0.881  0.35112
## AGE:BP:S2:S5            1   1168   1168   0.385  0.53700
## BMI:BP:S2:S5            1    222    222   0.073  0.78756
## AGE:S1:S2:S5            1    116    116   0.038  0.84576
## BMI:S1:S2:S5            1      5      5   0.001  0.96923
## BP:S1:S2:S5             1   7879   7879   2.595  0.11162
## AGE:BMI:S3:S5           1   5180   5180   1.706  0.19571
## AGE:BP:S3:S5            1   2504   2504   0.825  0.36689
## BMI:BP:S3:S5            1   3453   3453   1.137  0.28982
## AGE:S1:S3:S5            1   1548   1548   0.510  0.47756
## BMI:S1:S3:S5            1   1447   1447   0.477  0.49219
## BP:S1:S3:S5             1   1484   1484   0.489  0.48668
## AGE:S2:S3:S5            1    331    331   0.109  0.74236
## BMI:S2:S3:S5            1     69     69   0.023  0.88058
## BP:S2:S3:S5             1   1962   1962   0.646  0.42417
## S1:S2:S3:S5             1   4808   4808   1.584  0.21238
## AGE:BMI:BP:S6           1     16     16   0.005  0.94238
```

```
## AGE:BMI:S1:S6           1   4171    4171   1.374  0.24507
## AGE:BP:S1:S6            1     30      30   0.010  0.92057
## BMI:BP:S1:S6            1    234     234   0.077  0.78216
## AGE:BMI:S2:S6           1   7740    7740   2.550  0.11477
## AGE:BP:S2:S6            1      2       2   0.001  0.97775
## BMI:BP:S2:S6            1   7763    7763   2.557  0.11426
## AGE:S1:S2:S6            1    367     367   0.121  0.72901
## BMI:S1:S2:S6            1    726     726   0.239  0.62642
## BP:S1:S2:S6             1    193     193   0.064  0.80151
## AGE:BMI:S3:S6           1    790     790   0.260  0.61163
## AGE:BP:S3:S6            1   2957    2957   0.974  0.32704
## BMI:BP:S3:S6            1   1357    1357   0.447  0.50594
## AGE:S1:S3:S6            1    942     942   0.310  0.57934
## BMI:S1:S3:S6            1  10686   10686   3.520  0.06475 .
## BP:S1:S3:S6             1  14426   14426   4.752  0.03259 *
## AGE:S2:S3:S6            1   9321    9321   3.070  0.08406 .
## BMI:S2:S3:S6            1    666     666   0.219  0.64090
## BP:S2:S3:S6             1     67      67   0.022  0.88274
## S1:S2:S3:S6             1   9781    9781   3.222  0.07692 .
## AGE:BMI:S5:S6           1   7751    7751   2.553  0.11452
## AGE:BP:S5:S6            1    810     810   0.267  0.60719
## BMI:BP:S5:S6            1    165     165   0.054  0.81654
## AGE:S1:S5:S6            1   7214    7214   2.376  0.12764
## BMI:S1:S5:S6            1      0       0   0.000  0.99403
## BP:S1:S5:S6             1   6580    6580   2.167  0.14539
## AGE:S2:S5:S6            1    758     758   0.250  0.61882
## BMI:S2:S5:S6            1  15618   15618   5.144  0.02637 *
## BP:S2:S5:S6             1   4558    4558   1.501  0.22454
## S1:S2:S5:S6             1    512     512   0.169  0.68260
## AGE:S3:S5:S6            1   1262    1262   0.416  0.52116
## BMI:S3:S5:S6            1   1295    1295   0.427  0.51574
## BP:S3:S5:S6             1     29      29   0.009  0.92289
## S1:S3:S5:S6             1  15143   15143   4.988  0.02868 *
## S2:S3:S5:S6             1   9144    9144   3.012  0.08699 .
## AGE:BMI:BP:S1:S2        1   1032    1032   0.340  0.56166
## AGE:BMI:BP:S1:S3        1   2229    2229   0.734  0.39436
## AGE:BMI:BP:S2:S3        1    410     410   0.135  0.71424
## AGE:BMI:S1:S2:S3        1   3810    3810   1.255  0.26640
## AGE:BP:S1:S2:S3         1     26      26   0.009  0.92669
## BMI:BP:S1:S2:S3         1   1290    1290   0.425  0.51656
## AGE:BMI:BP:S1:S5        1   6598    6598   2.173  0.14485
## AGE:BMI:BP:S2:S5        1   1304    1304   0.429  0.51443
## AGE:BMI:S1:S2:S5        1   1759    1759   0.579  0.44911
## AGE:BP:S1:S2:S5         1    416     416   0.137  0.71249
## BMI:BP:S1:S2:S5         1    174     174   0.057  0.81173
## AGE:BMI:BP:S3:S5        1   2282    2282   0.752  0.38889
## AGE:BMI:S1:S3:S5        1    503     503   0.166  0.68508
## AGE:BP:S1:S3:S5         1     14      14   0.005  0.94631
## BMI:BP:S1:S3:S5         1   4485    4485   1.477  0.22824
## AGE:BMI:S2:S3:S5        1   2631    2631   0.866  0.35509
## AGE:BP:S2:S3:S5         1    721     721   0.237  0.62762
## BMI:BP:S2:S3:S5         1   3519    3519   1.159  0.28530
## AGE:S1:S2:S3:S5         1    664     664   0.219  0.64140
## BMI:S1:S2:S3:S5         1   6208    6208   2.045  0.15712
## BP:S1:S2:S3:S5          1   1953    1953   0.643  0.42516
## AGE:BMI:BP:S1:S6        1   8588    8588   2.829  0.09699 .
## AGE:BMI:BP:S2:S6        1    116     116   0.038  0.84578
```

```
## AGE:BMI:S1:S2:S6        1   2042    2042   0.673  0.41489
## AGE:BP:S1:S2:S6         1    580     580   0.191  0.66351
## BMI:BP:S1:S2:S6         1    959     959   0.316  0.57587
## AGE:BMI:BP:S3:S6        1     52      52   0.017  0.89653
## AGE:BMI:S1:S3:S6        1    553     553   0.182  0.67070
## AGE:BP:S1:S3:S6         1   7192    7192   2.369  0.12821
## BMI:BP:S1:S3:S6         1   5564    5564   1.833  0.18009
## AGE:BMI:S2:S3:S6        1    933     933   0.307  0.58101
## AGE:BP:S2:S3:S6         1   1249    1249   0.411  0.52329
## BMI:BP:S2:S3:S6         1    363     363   0.120  0.73035
## AGE:S1:S2:S3:S6         1    616     616   0.203  0.65390
## BMI:S1:S2:S3:S6         1   5100    5100   1.680  0.19915
## BP:S1:S2:S3:S6          1  23514   23514   7.745  0.00690 **
## AGE:BMI:BP:S5:S6        1   9093    9093   2.995  0.08787 .
## AGE:BMI:S1:S5:S6        1   8536    8536   2.811  0.09799 .
## AGE:BP:S1:S5:S6         1   5864    5864   1.931  0.16894
## BMI:BP:S1:S5:S6         1   2526    2526   0.832  0.36482
## AGE:BMI:S2:S5:S6        1   2691    2691   0.886  0.34968
## AGE:BP:S2:S5:S6         1   6561    6561   2.161  0.14598
## BMI:BP:S2:S5:S6         1     73      73   0.024  0.87729
## AGE:S1:S2:S5:S6         1   8866    8866   2.920  0.09184 .
## BMI:S1:S2:S5:S6         1   3263    3263   1.075  0.30342
## BP:S1:S2:S5:S6          1   1918    1918   0.632  0.42934
## AGE:BMI:S3:S5:S6        1   3296    3296   1.086  0.30097
## AGE:BP:S3:S5:S6         1   2571    2571   0.847  0.36056
## BMI:BP:S3:S5:S6         1   2200    2200   0.725  0.39746
## AGE:S1:S3:S5:S6         1   6816    6816   2.245  0.13846
## BMI:S1:S3:S5:S6         1   4357    4357   1.435  0.23492
## BP:S1:S3:S5:S6          1   4313    4313   1.421  0.23725
## AGE:S2:S3:S5:S6         1   1179    1179   0.388  0.53512
## BMI:S2:S3:S5:S6         1   4899    4899   1.614  0.20812
## BP:S2:S3:S5:S6          1     58      58   0.019  0.89018
## S1:S2:S3:S5:S6          1   3042    3042   1.002  0.32023
## AGE:BMI:BP:S1:S2:S3     1    130     130   0.043  0.83650
## AGE:BMI:BP:S1:S2:S5     1   1380    1380   0.455  0.50234
## AGE:BMI:BP:S1:S3:S5     1   2118    2118   0.698  0.40641
## AGE:BMI:BP:S2:S3:S5     1    808     808   0.266  0.60762
## AGE:BMI:S1:S2:S3:S5     1   2674    2674   0.881  0.35121
## AGE:BP:S1:S2:S3:S5      1     32      32   0.011  0.91863
## BMI:BP:S1:S2:S3:S5      1    384     384   0.126  0.72326
## AGE:BMI:BP:S1:S2:S6     1   1962    1962   0.646  0.42419
## AGE:BMI:BP:S1:S3:S6     1   2763    2763   0.910  0.34332
## AGE:BMI:BP:S2:S3:S6     1    146     146   0.048  0.82709
## AGE:BMI:S1:S2:S3:S6     1   2933    2933   0.966  0.32897
## AGE:BP:S1:S2:S3:S6      1   1331    1331   0.438  0.51007
## BMI:BP:S1:S2:S3:S6      1   5452    5452   1.796  0.18448
## AGE:BMI:BP:S1:S5:S6     1   4169    4169   1.373  0.24518
## AGE:BMI:BP:S2:S5:S6     1   5742    5742   1.891  0.17337
## AGE:BMI:S1:S2:S5:S6     1  21394   21394   7.047  0.00980 **
## AGE:BP:S1:S2:S5:S6      1      9       9   0.003  0.95580
## BMI:BP:S1:S2:S5:S6      1   7851    7851   2.586  0.11226
## AGE:BMI:BP:S3:S5:S6     1    435     435   0.143  0.70631
## AGE:BMI:S1:S3:S5:S6     1    577     577   0.190  0.66429
## AGE:BP:S1:S3:S5:S6      1    664     664   0.219  0.64153
## BMI:BP:S1:S3:S5:S6      1   1477    1477   0.486  0.48782
## AGE:BMI:S2:S3:S5:S6     1    477     477   0.157  0.69310
## AGE:BP:S2:S3:S5:S6      1    276     276   0.091  0.76384
```

```
## BMI:BP:S2:S3:S5:S6      1    930     930   0.306  0.58177
## AGE:S1:S2:S3:S5:S6      1   4483    4483   1.476  0.22835
## BMI:S1:S2:S3:S5:S6      1     42      42   0.014  0.90643
## BP:S1:S2:S3:S5:S6       1   5983    5983   1.971  0.16474
## AGE:BMI:BP:S1:S2:S3:S5  1   5757    5757   1.896  0.17282
## AGE:BMI:BP:S1:S2:S3:S6  1   1170    1170   0.385  0.53681
## AGE:BMI:BP:S1:S2:S5:S6  1    747     747   0.246  0.62151
## AGE:BMI:BP:S2:S3:S5:S6  1   3722    3722   1.226  0.27191
## AGE:BMI:S1:S2:S3:S5:S6  1   2288    2288   0.754  0.38822
## AGE:BP:S1:S2:S3:S5:S6   1   5154    5154   1.698  0.19682
## BMI:BP:S1:S2:S3:S5:S6   1     21      21   0.007  0.93424
## Residuals              71 215558    3036
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
  #Build Models
  #SLR
    #train
    fit <- lm(formula = Y~BMI,data=train_p)
    y.train.pred <- predict(fit,newdata=x.train_p)
    summary(fit)
```

```
##
## Call:
## lm(formula = Y ~ BMI, data = train_p)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -156.418  -45.526   -7.454   47.302  157.173
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -100.1405    21.8255  -4.588 6.39e-06 ***
## BMI            9.4412     0.8184  11.536  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 62.74 on 325 degrees of freedom
## Multiple R-squared:  0.2905, Adjusted R-squared:  0.2883
## F-statistic: 133.1 on 1 and 325 DF,  p-value: < 2.2e-16
```

```
    RMSE(y.train.pred,y.train_p)
```

```
## [1] 62.54935
```

```
    #test
    y.test.pred <- predict(fit,newdata=x.test_p)
    predict <- lm(y.test_p~y.test.pred)
    summary(predict)
```

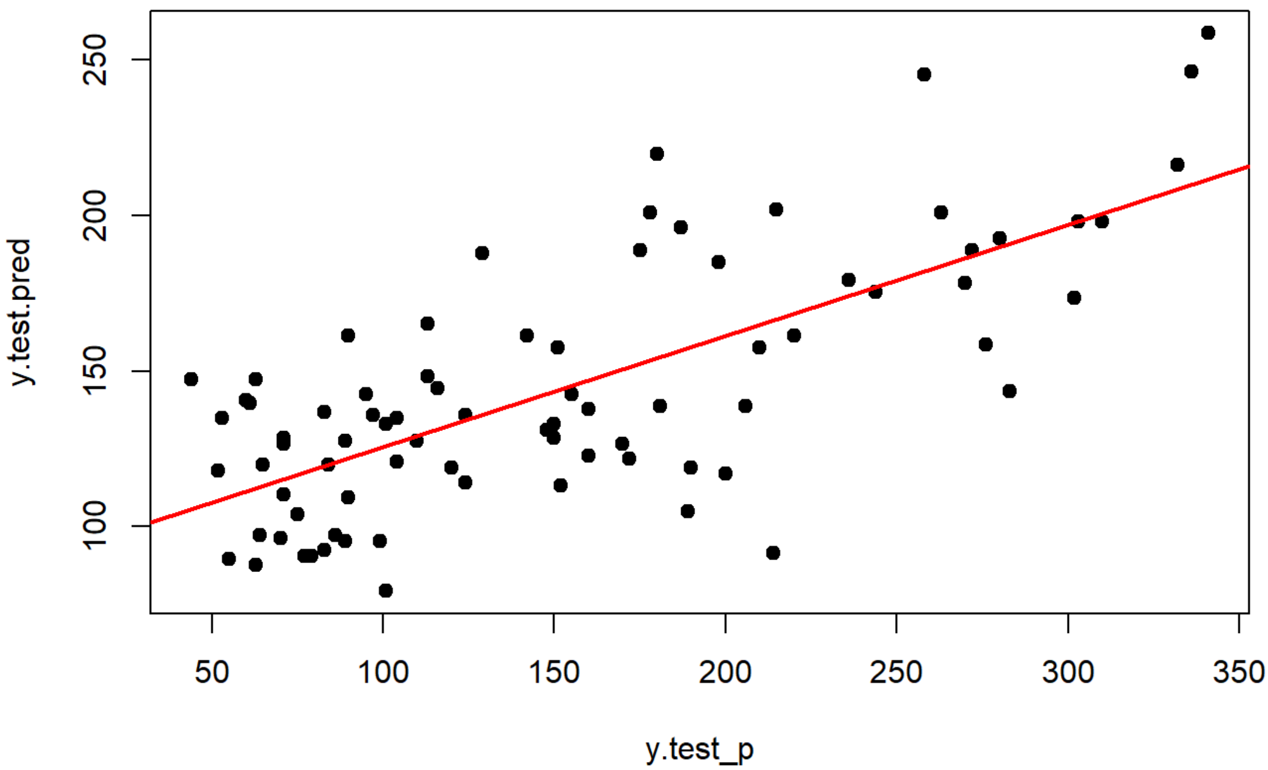
```
##
## Call:
```

```
## lm(formula = y.test_p ~ y.test.pred)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -112.682  -40.283   -3.243   39.606  137.618
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -55.5430    23.6648  -2.347   0.0215 *
## y.test.pred   1.4416     0.1581   9.120 6.31e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 56.46 on 78 degrees of freedom
## Multiple R-squared:  0.5161, Adjusted R-squared:  0.5099
## F-statistic: 83.18 on 1 and 78 DF,  p-value: 6.311e-14
```

```
RMSE(y.test.pred,y.test_p)
```

```
## [1] 59.04361
```

```
plot(y.test_p,y.test.pred, pch = 19, cex = 1, col = "black")
abline(lm(y.test.pred~y.test_p),col='red',lwd=2)
```




```
#multiple linear regression
  #train
  fit <- lm(formula = Y~SEX+BMI+BP+S1+S3+S5,data=train_p)
```

```
y.train.pred <- predict(fit,newdata=x.train_p)
summary(fit)
```

```
##
## Call:
## lm(formula = Y ~ SEX + BMI + BP + S1 + S3 + S5, data = train_p)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -149.543  -37.501   -0.557   35.273  141.916
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -215.9629    44.5227  -4.851 1.93e-06 ***
## SEX          -22.8206     6.8001  -3.356 0.000886 ***
## BMI            4.0890     0.8783   4.655 4.74e-06 ***
## BP             1.3396     0.2709   4.945 1.23e-06 ***
## S1            -0.1542     0.1181  -1.306 0.192529
## S3            -1.1371     0.3262  -3.486 0.000560 ***
## S5            54.1603     8.6748   6.243 1.36e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 53.93 on 320 degrees of freedom
## Multiple R-squared:  0.4838, Adjusted R-squared:  0.4741
## F-statistic: 49.98 on 6 and 320 DF,  p-value: < 2.2e-16
```

```
RMSE(y.train.pred,y.train_p)
```

```
## [1] 53.3535
```

```
#test
y.test.pred <- predict(fit,newdata=x.test_p)
predict <- lm(y.test_p~y.test.pred)
summary(predict)
```

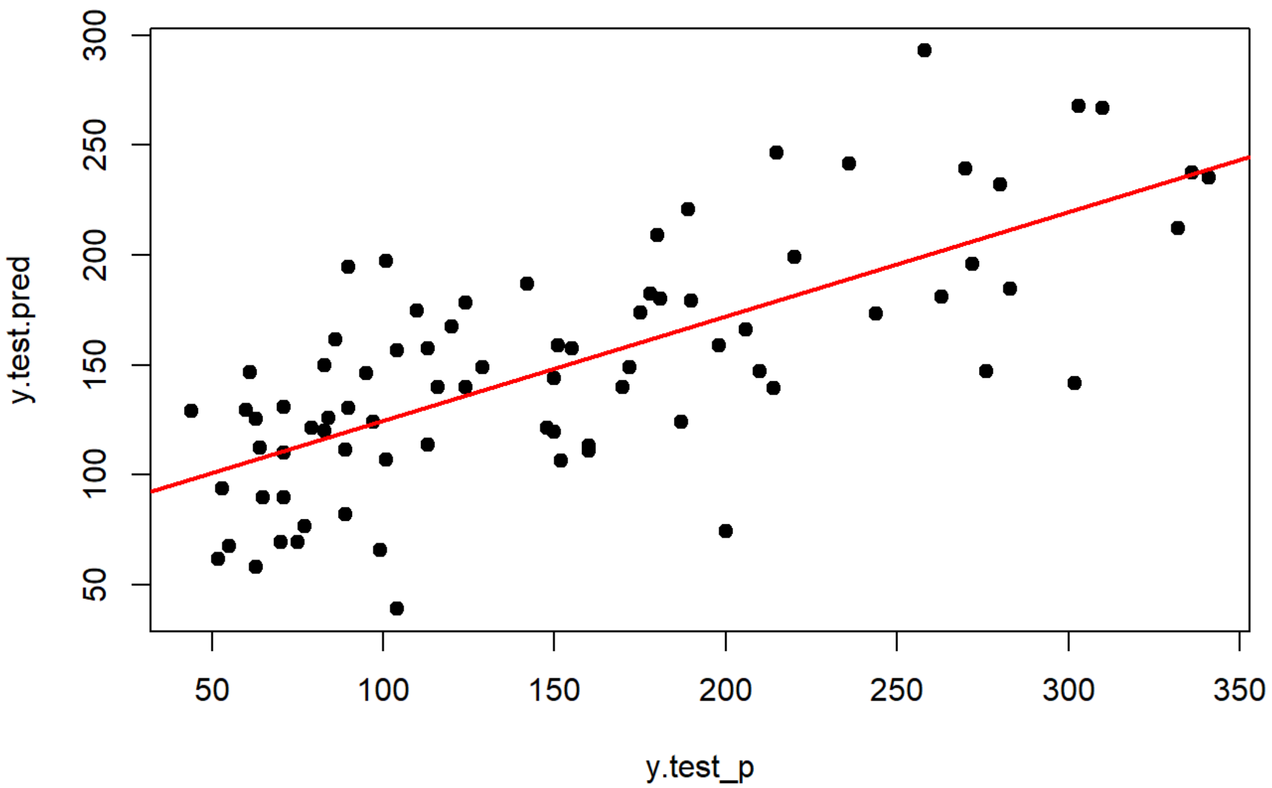
```
##
## Call:
## lm(formula = y.test_p ~ y.test.pred)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -110.539  -42.363   -5.933   35.771  158.181
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -6.9275    18.9331  -0.366    0.715
## y.test.pred   1.0656     0.1192   8.941  1.4e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 57.04 on 78 degrees of freedom
```

```
## Multiple R-squared:  0.5061, Adjusted R-squared:  0.4998
## F-statistic: 79.94 on 1 and 78 DF,  p-value: 1.404e-13
```

```
    RMSE(y.test.pred,y.test_p)
```

```
## [1] 56.50382
```

```
    plot(y.test_p,y.test.pred, pch = 19, cex = 1, col = "black")
    abline(lm(y.test.pred~y.test_p),col='red',lwd=2)
```




```
  #polynomial regression
    #train
    fit <- lm(formula = Y~AGE+BMI+S3+SEX+S5+I(BP^2)+I(AGE*SEX)+I(BMI*BP)+I(BP*S1*S
3*S6)+I(AGE*BMI*S1*S3)+
              I(BMI*S2*S5*S6)+I(BP*S1*S2*S3*S6),
              data=train_p)
    y.train.pred <- predict(fit,newdata=x.train_p)
    summary(fit)
```

```
##
## Call:
## lm(formula = Y ~ AGE + BMI + S3 + SEX + S5 + I(BP^2) + I(AGE *
##     SEX) + I(BMI * BP) + I(BP * S1 * S3 * S6) + I(AGE * BMI *
##     S1 * S3) + I(BMI * S2 * S5 * S6) + I(BP * S1 * S2 * S3 *
##     S6), data = train_p)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -142.038  -37.410   -0.199   34.083  133.681 
## 
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                 2.947e+02  1.329e+02   2.218 0.027300 *  
## AGE                        -2.731e+00  1.053e+00  -2.594 0.009927 ** 
## BMI                        -1.548e+01  5.445e+00  -2.842 0.004776 ** 
## S3                         -2.081e+00  1.026e+00  -2.028 0.043409 *  
## SEX                        -9.272e+01  2.422e+01  -3.829 0.000155 ***
## S5                          4.012e+01  1.002e+01   4.005 7.76e-05 ***
## I(BP^2)                    -1.931e-02  8.467e-03  -2.281 0.023212 *  
## I(AGE * SEX)                1.431e+00  4.810e-01   2.976 0.003147 ** 
## I(BMI * BP)                 1.841e-01  5.534e-02   3.326 0.000985 ***
## I(BP * S1 * S3 * S6)        6.006e-07  5.372e-07   1.118 0.264405    
## I(AGE * BMI * S1 * S3)      2.929e-06  2.529e-06   1.158 0.247655    
## I(BMI * S2 * S5 * S6)       2.103e-05  1.936e-05   1.086 0.278230    
## I(BP * S1 * S2 * S3 * S6) -5.426e-09  3.016e-09  -1.799 0.072951 .  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 52.78 on 314 degrees of freedom
## Multiple R-squared:  0.5149, Adjusted R-squared:  0.4964 
## F-statistic: 27.78 on 12 and 314 DF,  p-value: < 2.2e-16
```

```
RMSE(y.train.pred,y.train_p)
```

```
## [1] 51.71859
```

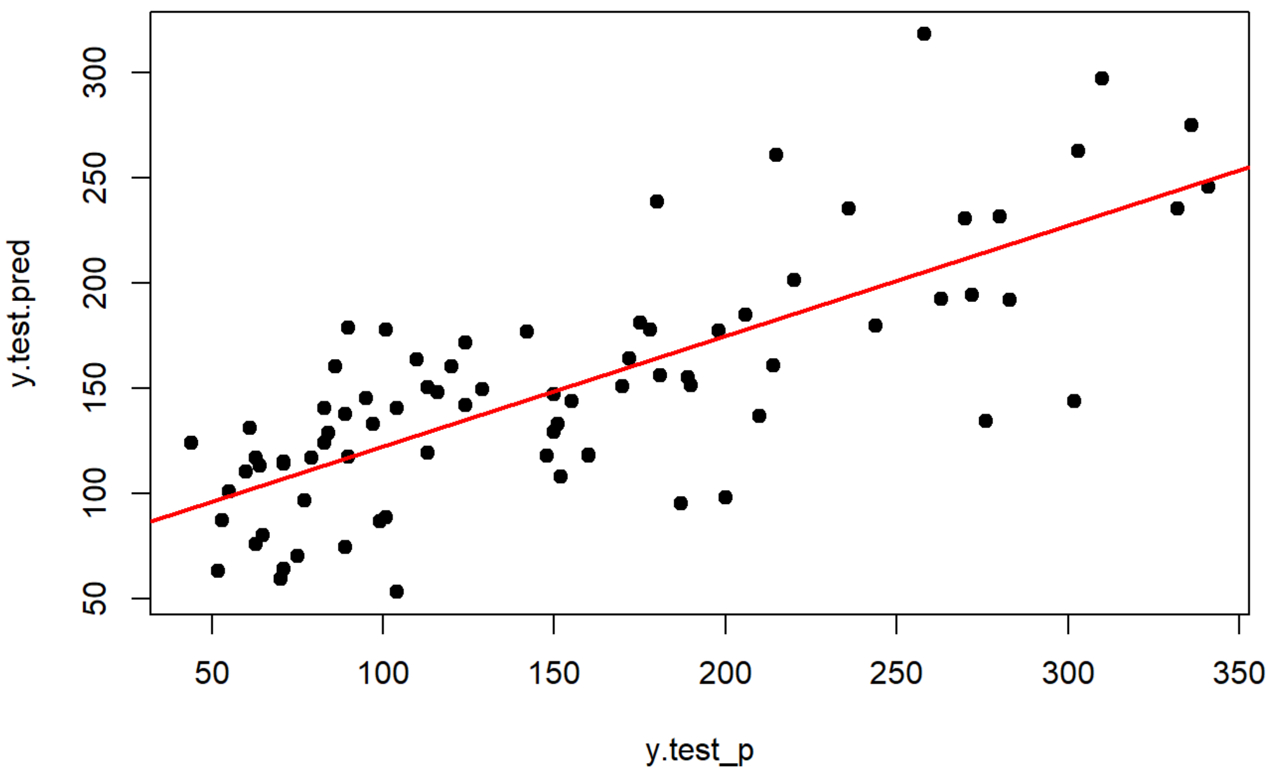
```
#test
y.test.pred <- predict(fit,newdata=x.test_p)
predict <- lm(y.test_p~y.test.pred)
summary(predict)
```

```
## 
## Call:
## lm(formula = y.test_p ~ y.test.pred)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -93.94 -43.09  -4.08  30.47 155.92 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -10.6151    17.0283  -0.623    0.535    
## y.test.pred   1.0891     0.1066  10.217 4.83e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 53.08 on 78 degrees of freedom
## Multiple R-squared:  0.5724, Adjusted R-squared:  0.5669 
## F-statistic: 104.4 on 1 and 78 DF,  p-value: 4.83e-16
```

```
RMSE(y.test.pred,y.test_p)
```

```
## [1] 52.71497
```

```
plot(y.test_p,y.test.pred, pch = 19, cex = 1, col = "black")
abline(lm(y.test.pred~y.test_p),col='red',lwd=2)
```




```
#ridge (alpha = 0)
set.seed(123)
ridge.fit <- cv.glmnet(as.matrix(x.train_p),y.train_p,type.measure ='mse',alpha=
0,family='gaussian')
ridge.fit$lambda.1se
```

```
## [1] 56.88734
```

```
coef(ridge.fit)
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                         s1
## (Intercept) -1.447451e+02
## AGE          1.147955e-01
## SEX         -9.351097e+00
## BMI          3.051928e+00
## BP           8.154471e-01
## S1           2.035922e-03
```

```
## S2          -6.766365e-02
## S3          -6.811350e-01
## S4           4.990239e+00
## S5           2.890386e+01
## S6           3.586102e-01
```

```
    #summary train
    ridge.predict.train <- predict(ridge.fit,s=ridge.fit$lambda.1se,newx = as.matr
ix(x.train_p))
    fit <- lm(y.train_p~ridge.predict.train)
    summary(fit)
```

```
## 
## Call:
## lm(formula = y.train_p ~ ridge.predict.train)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -133.567  -39.101   -1.942   40.989  128.080 
## 
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)    
## (Intercept)        -47.28045   11.94939  -3.957 9.33e-05 ***
## ridge.predict.train   1.31854    0.07792  16.921  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 54.31 on 325 degrees of freedom
## Multiple R-squared:  0.4684, Adjusted R-squared:  0.4667 
## F-statistic: 286.3 on 1 and 325 DF,  p-value: < 2.2e-16
```

```
    RMSE(ridge.predict.train,y.train_p)
```

```
## [1] 55.51858
```

```
    #summary test
    ridge.predict.test <- predict(ridge.fit,s=ridge.fit$lambda.1se,newx = as.matri
x(x.test_p))
    predict <- lm(y.test_p~ridge.predict.test)
    summary(predict)
```

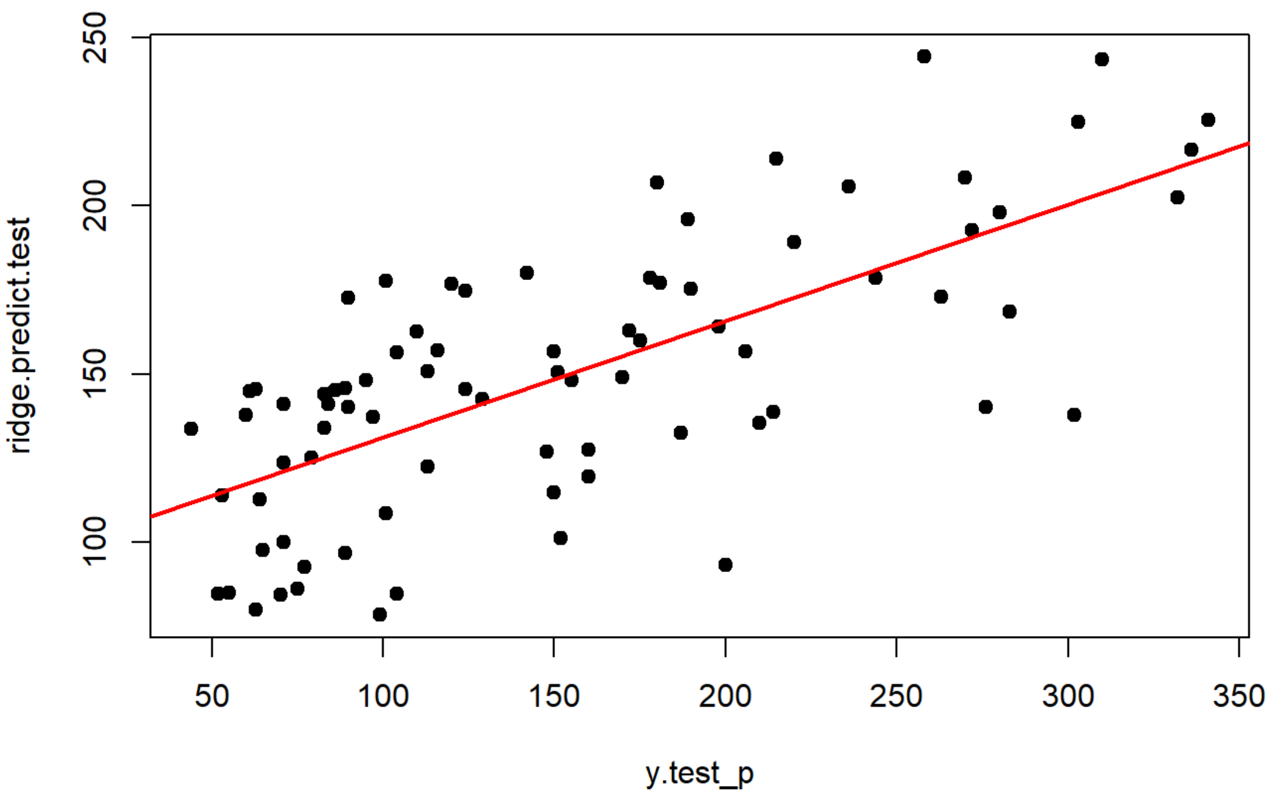
```
## 
## Call:
## lm(formula = y.test_p ~ ridge.predict.test)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -95.51 -48.11  -2.48  42.97 165.79 
## 
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)  
## (Intercept)        -58.6149    25.2386  -2.322   0.0228 *
```

```
## ridge.predict.test   1.4134     0.1633   8.654 5.08e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 57.97 on 78 degrees of freedom
## Multiple R-squared:  0.4898, Adjusted R-squared:  0.4833
## F-statistic: 74.88 on 1 and 78 DF,  p-value: 5.078e-13
```

```
RMSE(ridge.predict.test,y.test_p)
```

```
## [1] 59.6312
```

```
#plot
plot(y.test_p,ridge.predict.test, pch = 19, cex = 1, col = "black")
abline(lm(ridge.predict.test~y.test_p),col='red',lwd=2)
```




```
#lasso (alpha = 1)
set.seed(123)
lasso.fit <- cv.glmnet(as.matrix(x.train_p),y.train_p,type.measure ='mse',alpha=
1,family='gaussian')
lasso.fit$lambda.1se
```

```
## [1] 7.87825
```

```
coef(lasso.fit)
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept) -192.5409138
## AGE             .
## SEX             .
## BMI             3.9613444
## BP              0.6776335
## S1              .
## S2              .
## S3             -0.5086763
## S4              .
## S5             43.0942120
## S6              .
```

```
    #summary train
    lasso.predict.train <- predict(lasso.fit,s=lasso.fit$lambda.1se,newx = as.matr
ix(x.train_p))
    fit <- lm(y.train_p~lasso.predict.train)
    summary(fit)
```

```
##
## Call:
## lm(formula = y.train_p ~ lasso.predict.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -142.971  -38.190   -1.197   41.732  135.871
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -38.37113   11.64643  -3.295  0.00109 **
## lasso.predict.train   1.25852    0.07576  16.611  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54.78 on 325 degrees of freedom
## Multiple R-squared:  0.4592, Adjusted R-squared:  0.4575
## F-statistic: 275.9 on 1 and 325 DF,  p-value: < 2.2e-16
```

```
    RMSE(lasso.predict.train,y.train_p)
```

```
## [1] 55.58046
```

```
    #summary test
    lasso.predict.test <- predict(lasso.fit,s=lasso.fit$lambda.1se,newx = as.matri
x(x.test_p))
    predict <- lm(y.test_p~lasso.predict.test)
    summary(predict)
```

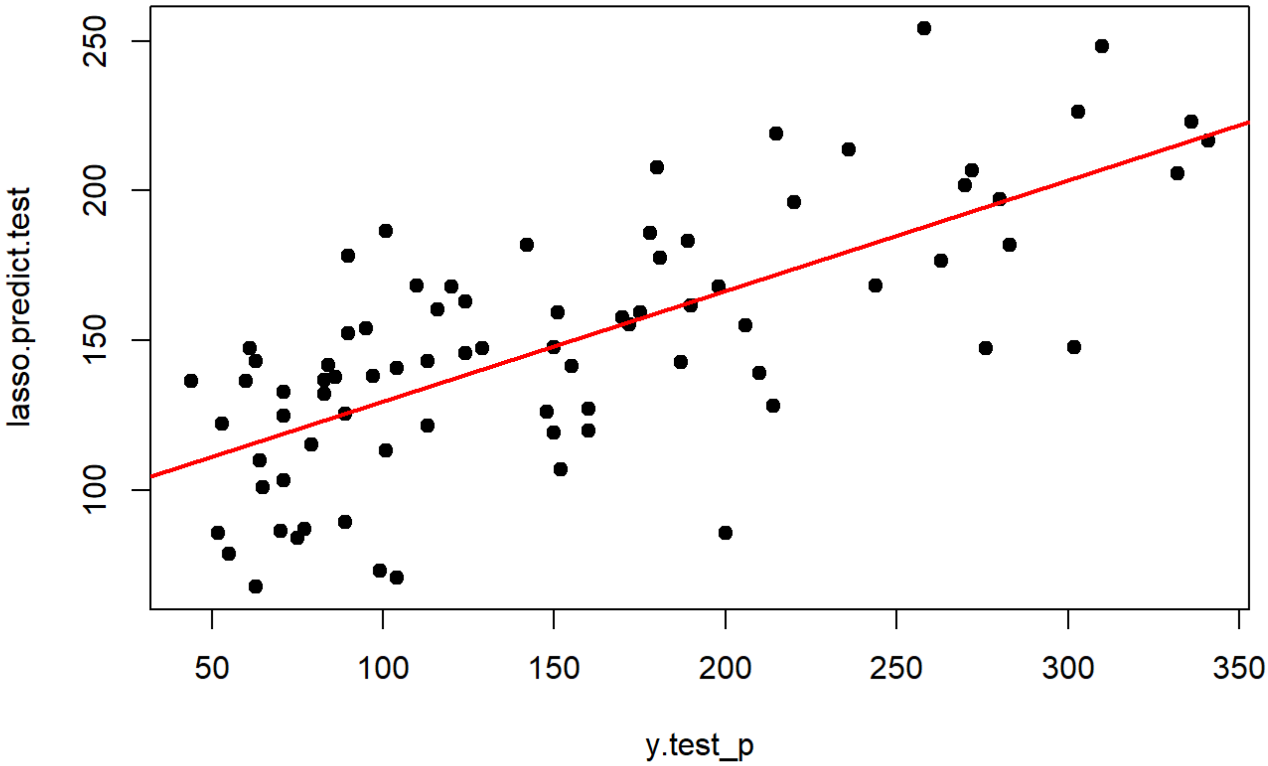
```
##
## Call:
## lm(formula = y.test_p ~ lasso.predict.test)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -102.830  -46.847   -2.457   41.180  151.067
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -50.8834    23.7395  -2.143   0.0352 *
## lasso.predict.test   1.3650     0.1535   8.894 1.73e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 57.19 on 78 degrees of freedom
## Multiple R-squared:  0.5035, Adjusted R-squared:  0.4972
## F-statistic: 79.11 on 1 and 78 DF,  p-value: 1.73e-13
```

```
RMSE(lasso.predict.test,y.test_p)
```

```
## [1] 58.58606
```

```
#plot
plot(y.test_p,lasso.predict.test, pch = 19, cex = 1, col = "black")
abline(lm(lasso.predict.test~y.test_p),col='red',lwd=2)
```




```
#elastic net
results.train <-data.frame()
for (i in 0:20)
```

```
  {
    set.seed(123)
    fit <- cv.glmnet(as.matrix(x.train_p), y.train_p, type.measure="mse", alpha=i/
20,
                     family="gaussian")
    y.pred <- predict(fit, s=fit$lambda.1se, newx=as.matrix(x.train_p))
    predict <- lm(y.train_p~y.pred)

    temp <- data.frame(alpha=i/20,R2= summary(predict)$r.squared,Adj_R2=summary(pr
edict)$adj.r.squared,rmse=RMSE(y.pred,y.train_p),lambda=fit$lambda.1se)
    results.train <- rbind(results.train, temp)
  }
  #best alpha = 0.05 (best adj R2)

  set.seed(123)
  elastic.fit <- cv.glmnet(as.matrix(x.train_p), y.train_p, type.measure="mse", al
pha=0.05,
                   family="gaussian")
  elastic.fit$lambda.1se
```

```
## [1] 39.03005
```

```
  coef(elastic.fit)
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept) -156.68332510
## AGE            0.00625415
## SEX           -8.01131052
## BMI            3.28575464
## BP             0.84842402
## S1             .
## S2            -0.01193017
## S3            -0.69496170
## S4             3.61105763
## S5            32.11917544
## S6             0.26372880
```

```
    #train
    elastic.predict.train <- predict(elastic.fit, s=elastic.fit$lambda.1se, newx=a
s.matrix(x.train_p))
    fit <- lm(y.train_p~elastic.predict.train)
    summary(fit)
```

```
##
## Call:
## lm(formula = y.train_p ~ elastic.predict.train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -135.223  -38.024   -1.198   40.713  130.613
##
## Coefficients:
```

```
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -43.03977   11.69242  -3.681 0.000272 ***
## elastic.predict.train   1.28997    0.07614  16.943  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 54.28 on 325 degrees of freedom
## Multiple R-squared:  0.469,  Adjusted R-squared:  0.4674
## F-statistic: 287.1 on 1 and 325 DF,  p-value: < 2.2e-16
```

```
    RMSE(elastic.predict.train,y.train_p)
```

```
## [1] 55.30564
```

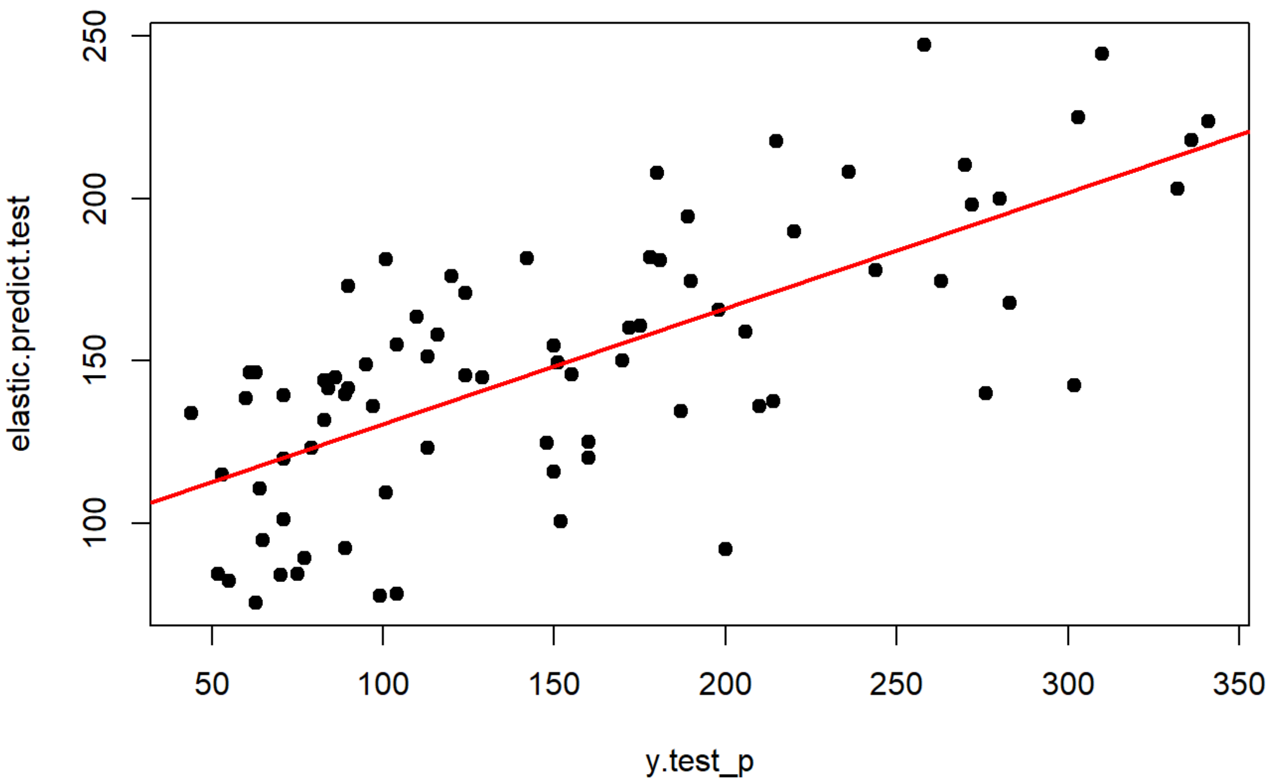
```
    #test
    elastic.predict.test <- predict(elastic.fit, s=elastic.fit$lambda.1se, newx=a
s.matrix(x.test_p))
    predict <- lm(y.test_p~elastic.predict.test)
    summary(predict)
```

```
##
## Call:
## lm(formula = y.test_p ~ elastic.predict.test)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -95.690 -49.031  -2.624  41.652 158.960
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          -53.7196    24.5495  -2.188   0.0316 *
## elastic.predict.test   1.3812     0.1587   8.706 4.02e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 57.8 on 78 degrees of freedom
## Multiple R-squared:  0.4928, Adjusted R-squared:  0.4863
## F-statistic: 75.79 on 1 and 78 DF,  p-value: 4.021e-13
```

```
    RMSE(elastic.predict.test,y.test_p)
```

```
## [1] 59.23687
```

```
    #plot
    plot(y.test_p, elastic.predict.test, pch = 19, cex = 1, col = "black")
    abline(lm(elastic.predict.test~y.test_p),col='red',lwd=2)
```



250
200
150
100
elastic.predict.test
50
100
150
200
250
300
350
y.test_p