Project

2022-11-28

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.2.2
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.0      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(ggcorrplot)
```

```
## Warning: package 'ggcorrplot' was built under R version 4.2.2
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.2.2
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.2.2
```

```
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 4.2.2
```

```
## Loading required package: Matrix
##
## Attaching package: 'Matrix'
##
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
##
## Loaded glmnet 4.1-4
```

```
library(modelr)
```

Description of data

```
bd_df <- readxl::read_excel("data/body_density_data.xlsx")
dim(bd_df)
```

```
## [1] 252  17
```

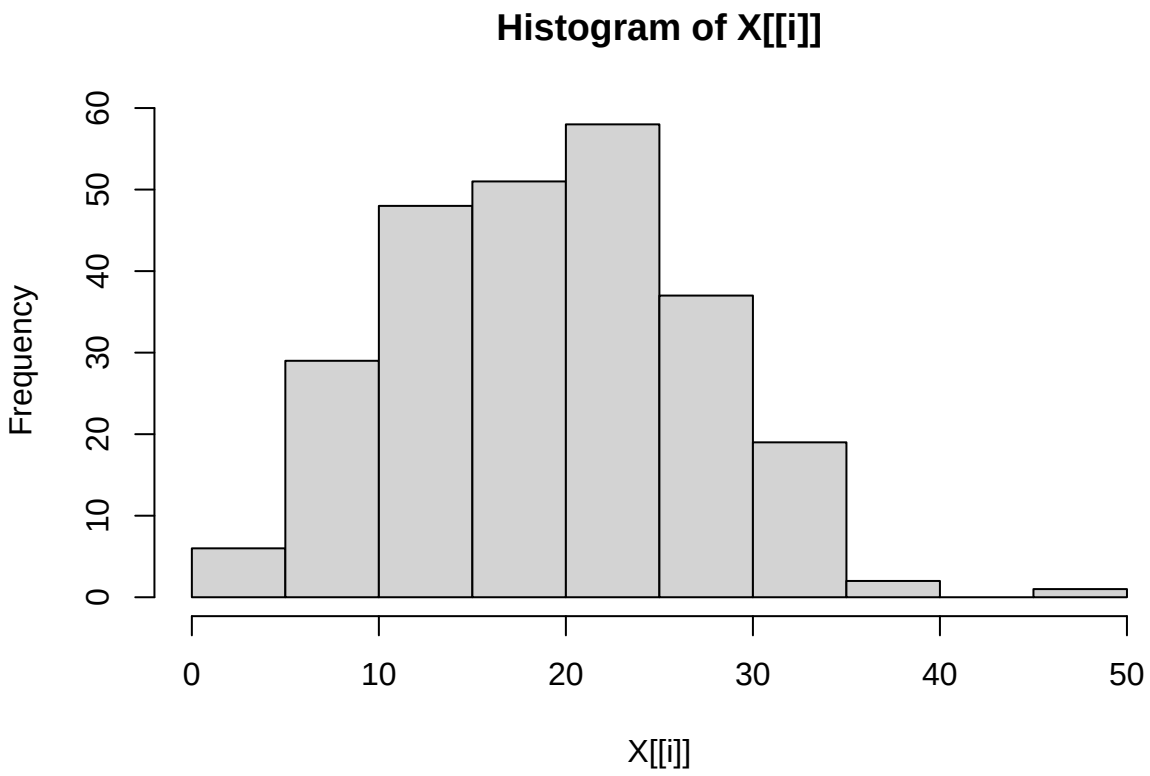
```
bd_df <- bd_df%>%
  select(bodyfat_brozek, age:wrist)%>%
  filter(bodyfat_brozek != 0)
```

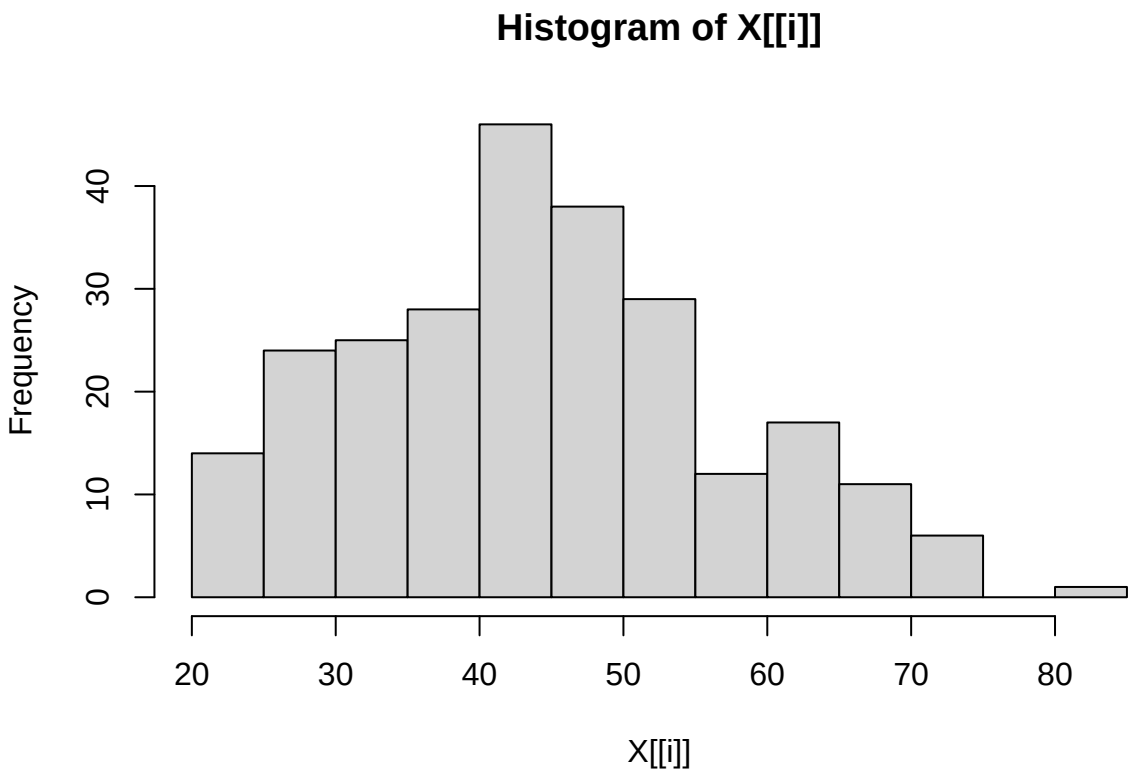
We first looked into the content of the dataset. It has 252 observations across 17 columns. The outcome we selected is bodyfat_brozek (body fat calculated by brozek), the other key variables are age, height, weight and circumference of body part like neck, chest, abdomen, hip, thigh, knee, ankle, bicep, forearm and wrist.

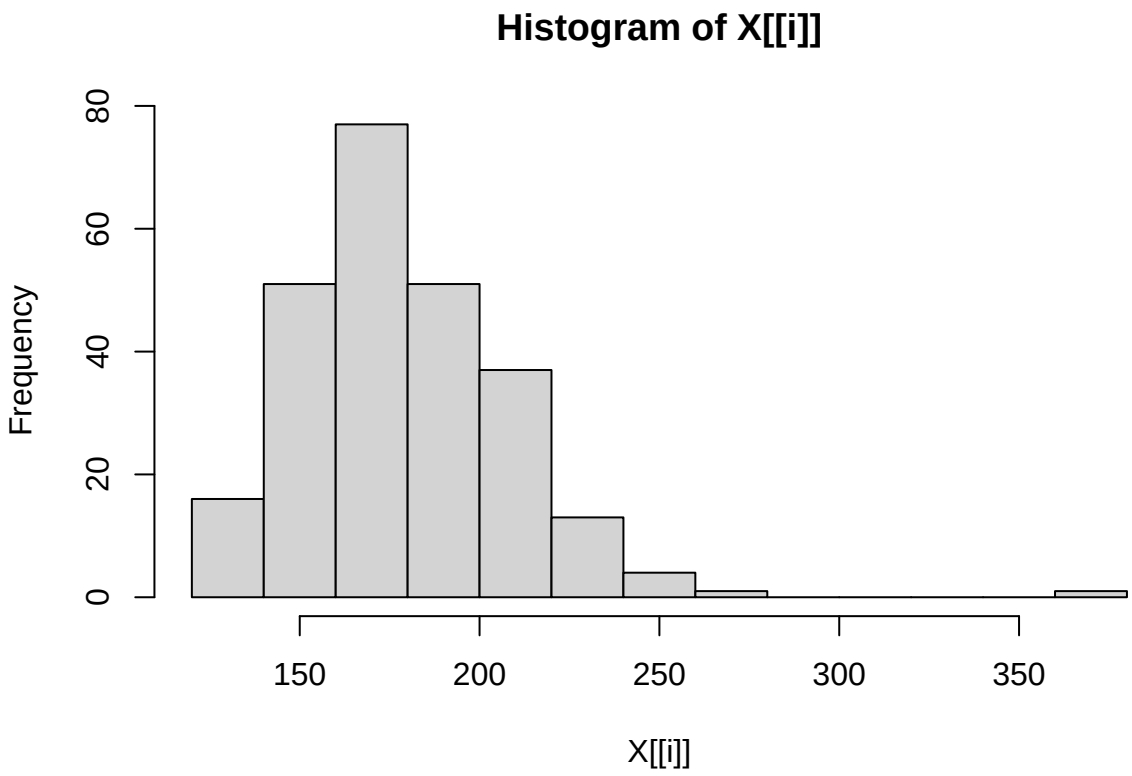
plots

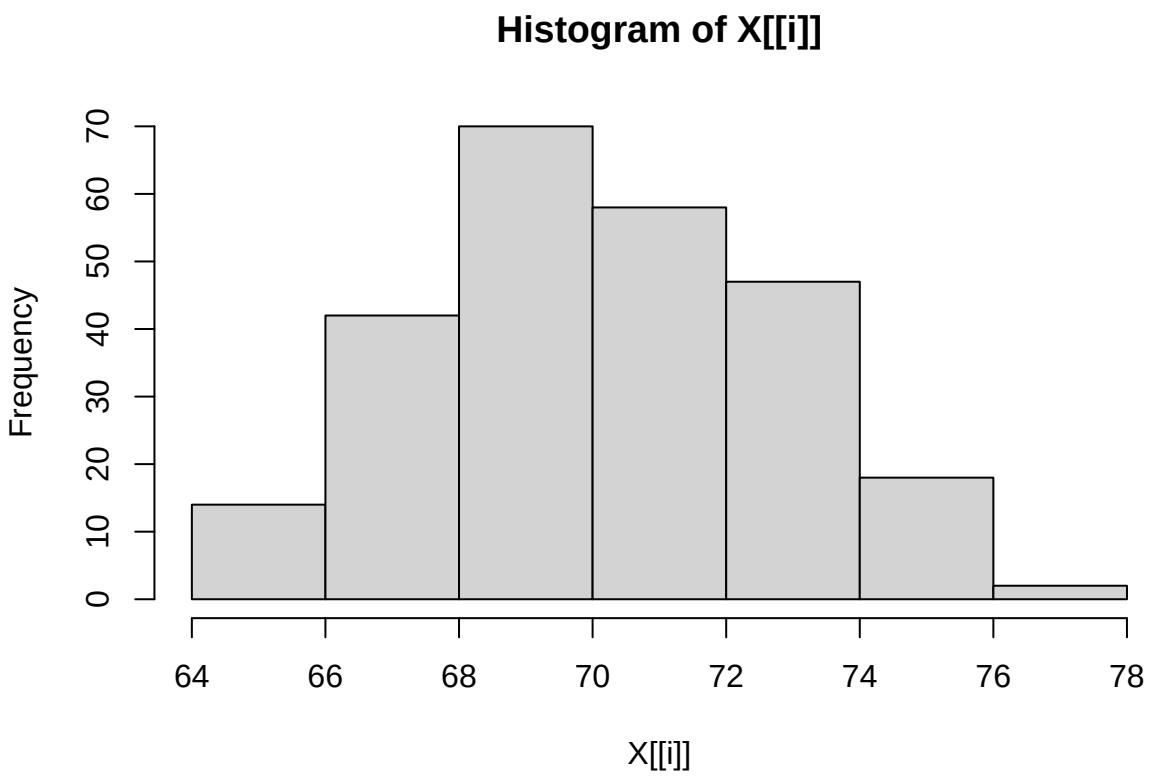
And we use plots to check the distribution of our variables and relationship between our outcome and each of the variables.

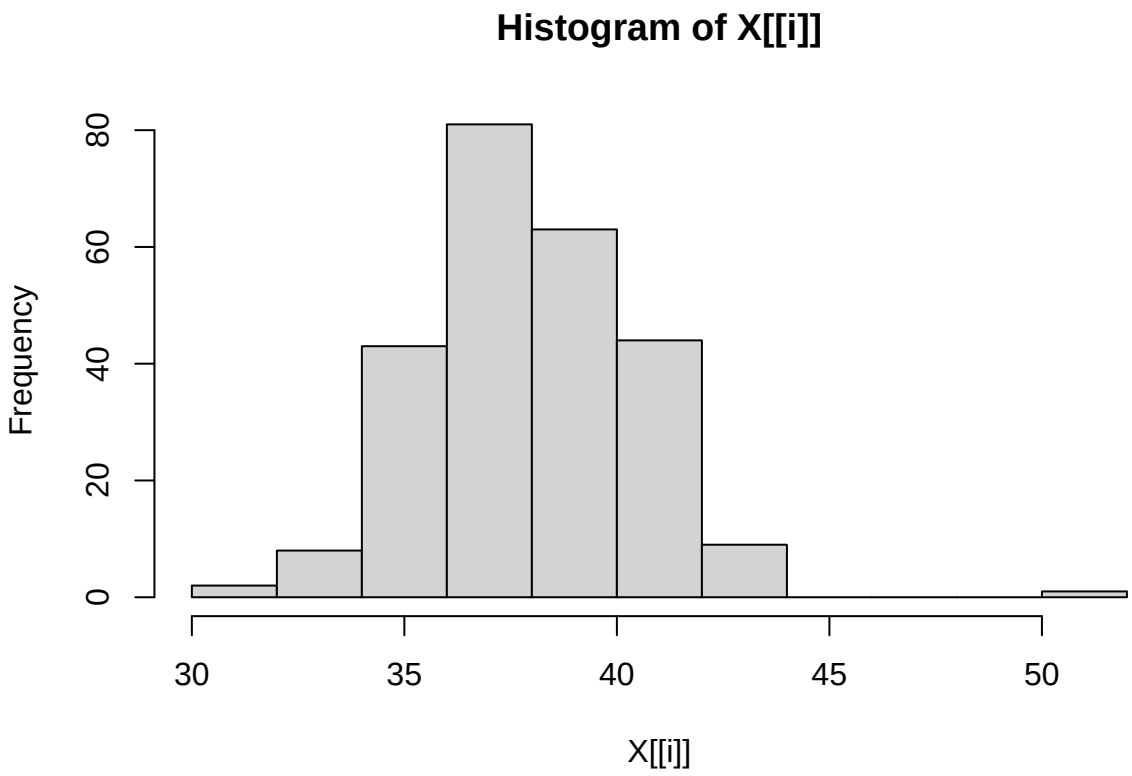
```
lapply(bd_df, hist)
```

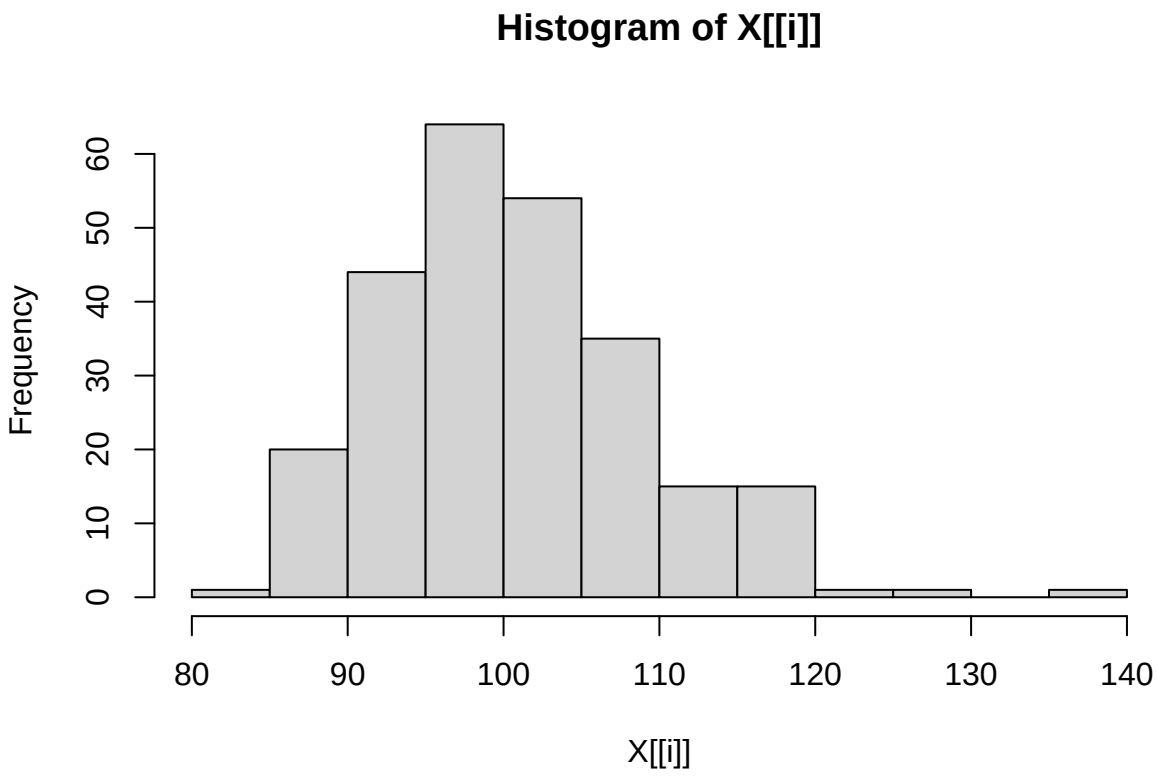

Histogram of X[[i]]
Frequency
0
10
20
30
40
50
60
0
10
20
30
40
50
X[[i]]

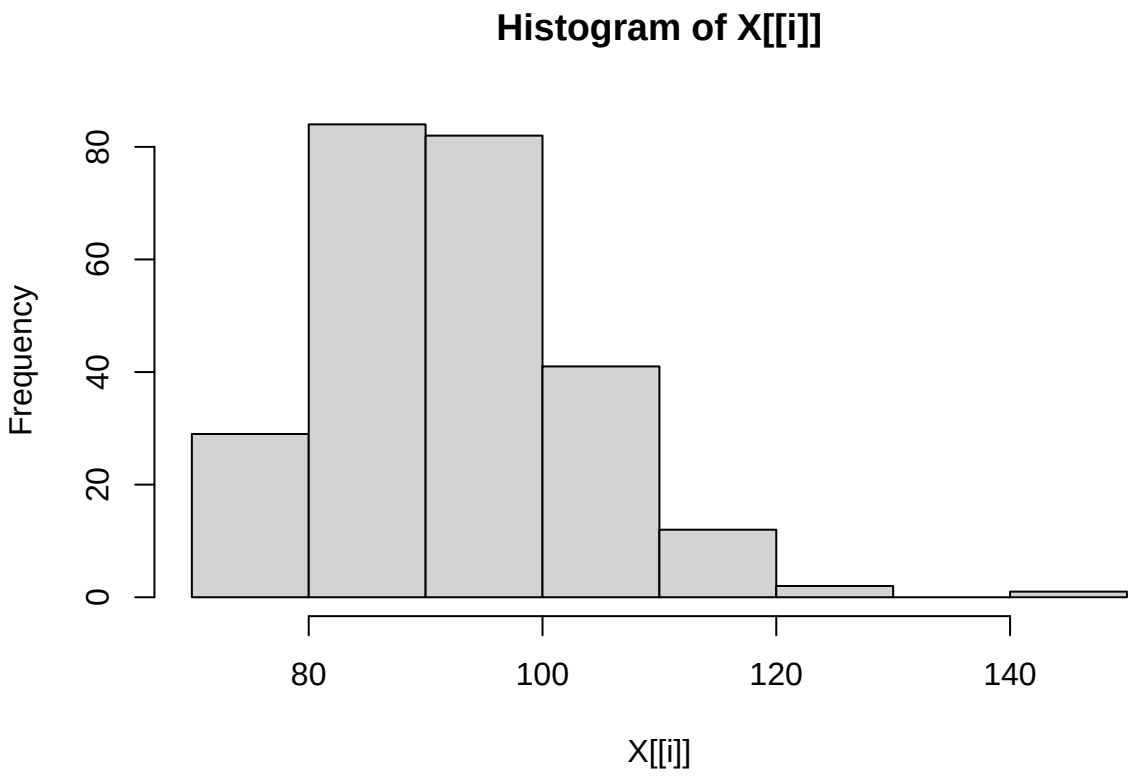

Histogram of X[[i]]
Frequency
0
10
20
30
40
20
30
40
50
60
70
80
X[[i]]

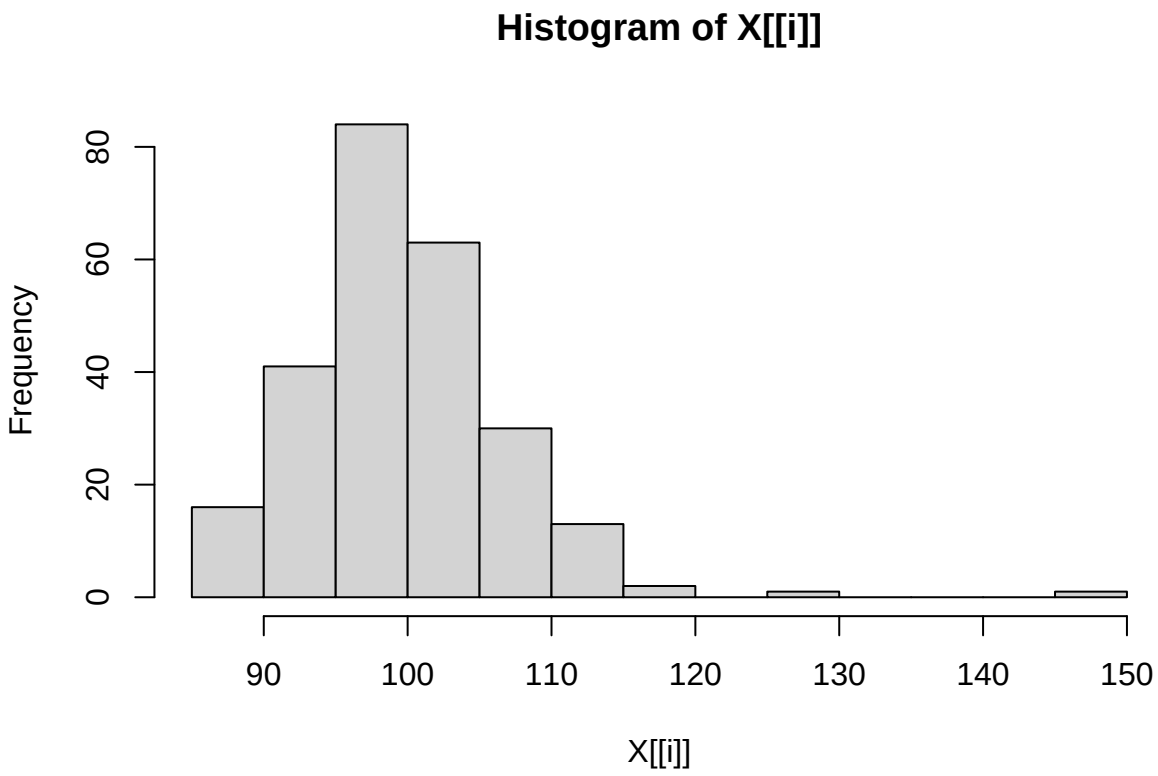

Histogram of X[[i]]
Frequency
0
20
40
60
80
150
200
250
300
350
X[[i]]

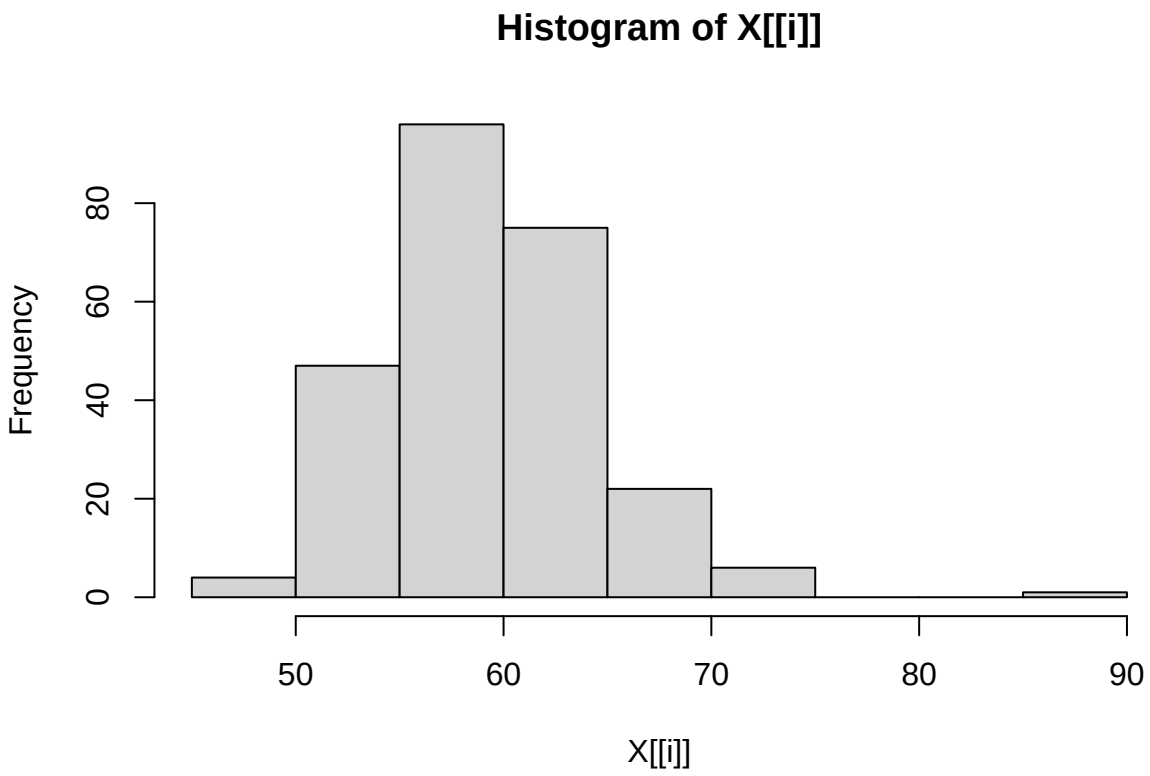

Histogram of X[[i]]
Frequency
0
10
20
30
40
50
60
70
64
66
68
70
72
74
76
78
X[[i]]

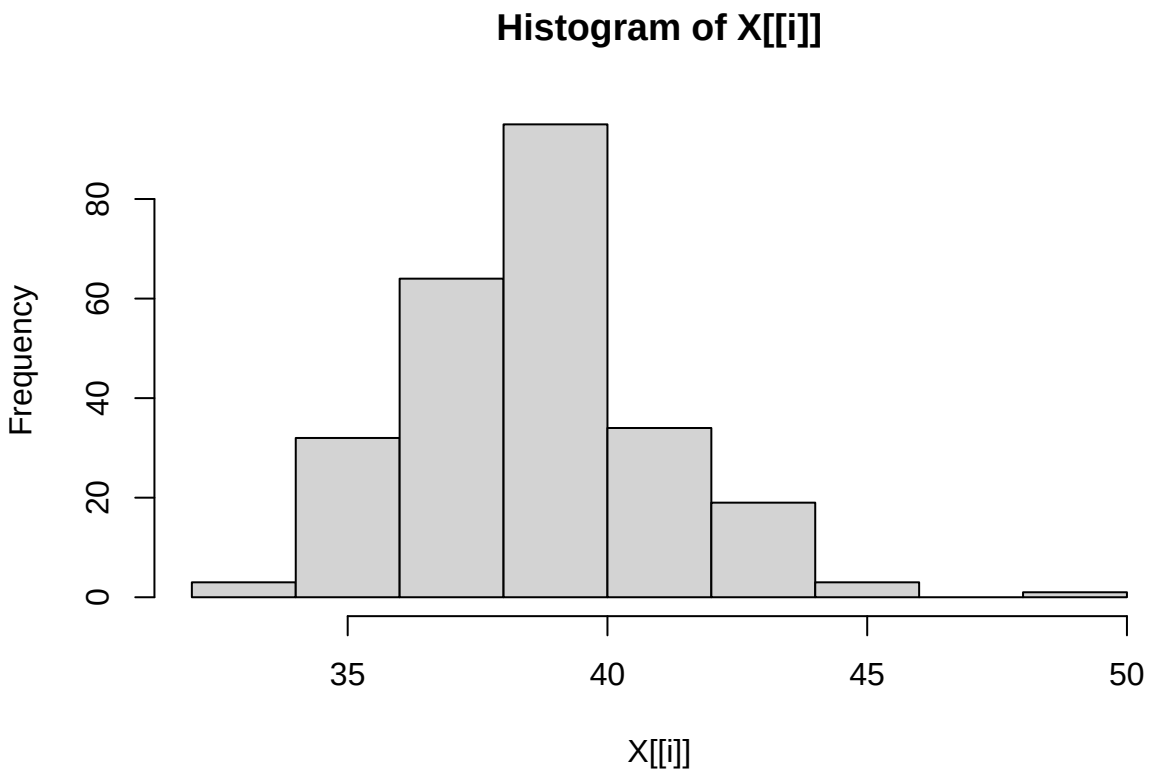

Histogram of X[[i]]
Frequency
0
20
40
60
80
30
35
40
45
50
X[[i]]

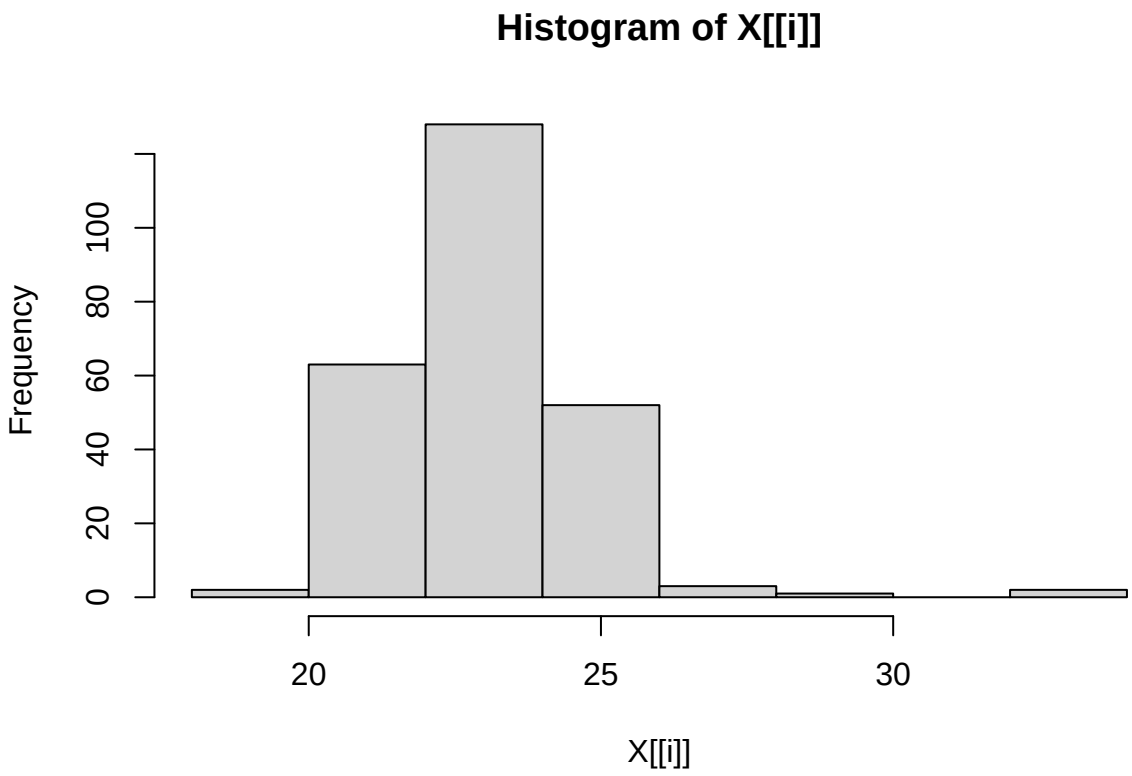

Histogram of X[[i]]
Frequency
0
10
20
30
40
50
60
80
90
100
110
120
130
140
X[[i]]

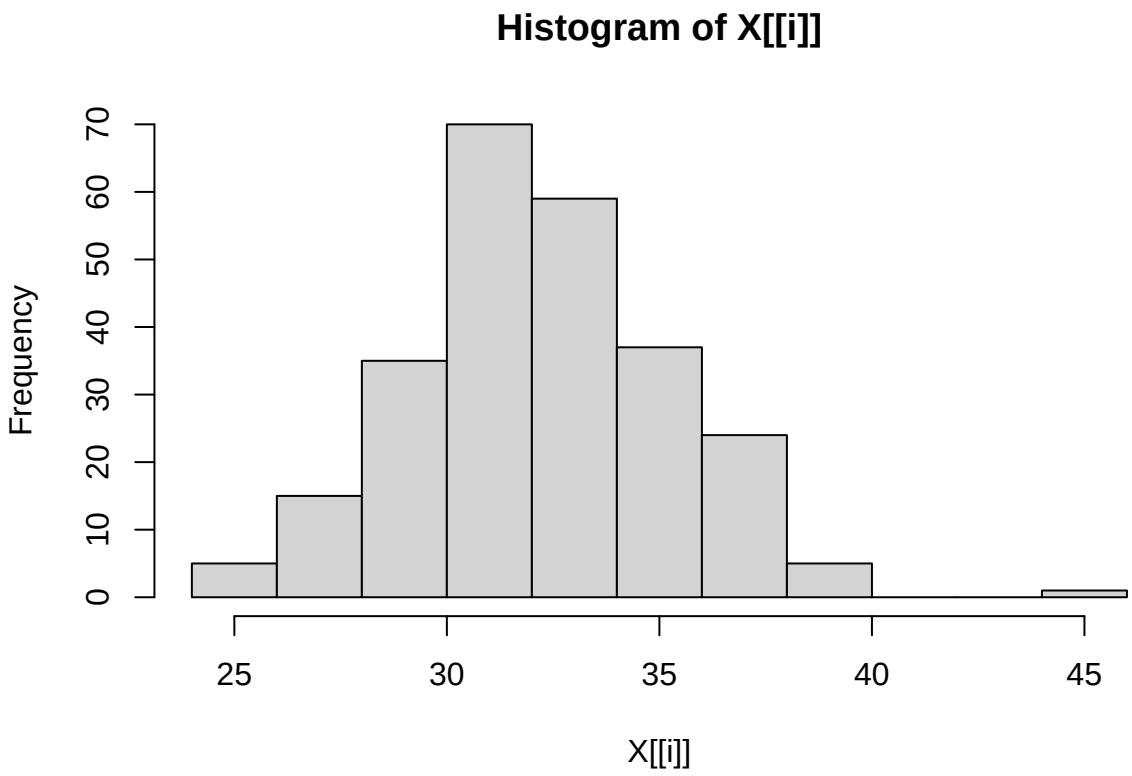

Histogram of X[[i]]
Frequency
0
20
40
60
80
80
100
120
140
X[[i]]

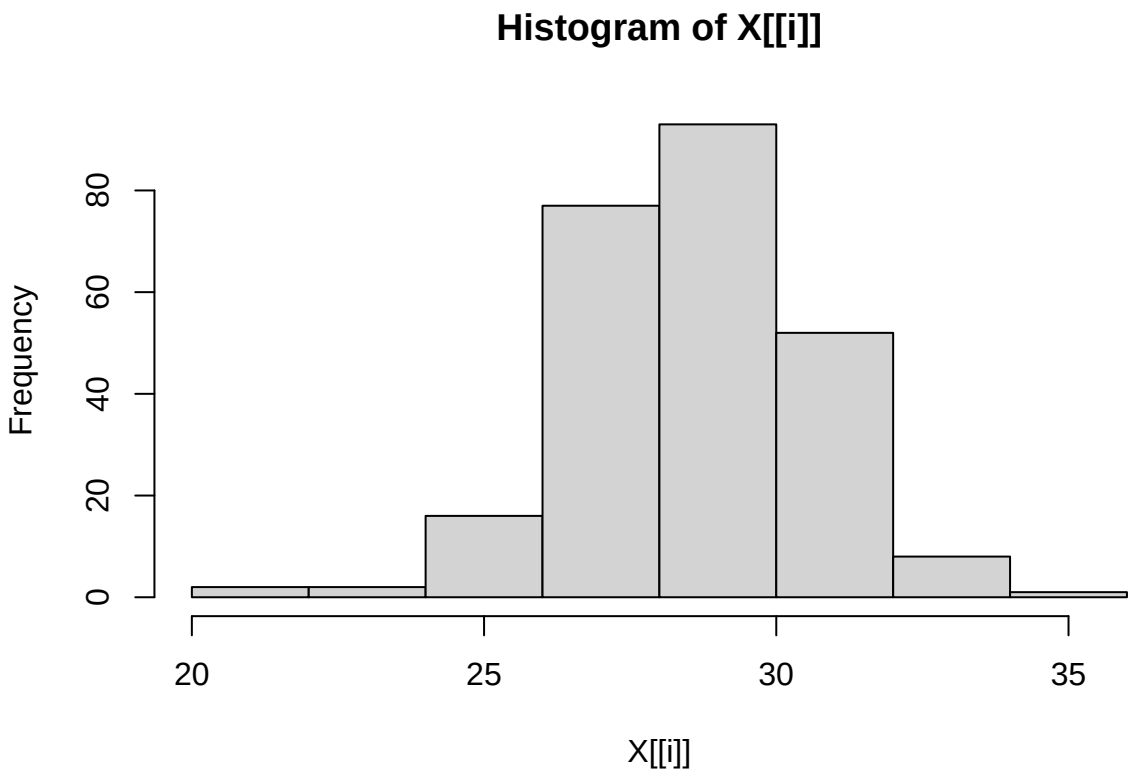

Histogram of X[[i]]
Frequency
0
20
40
60
80
90
100
110
120
130
140
150
X[[i]]

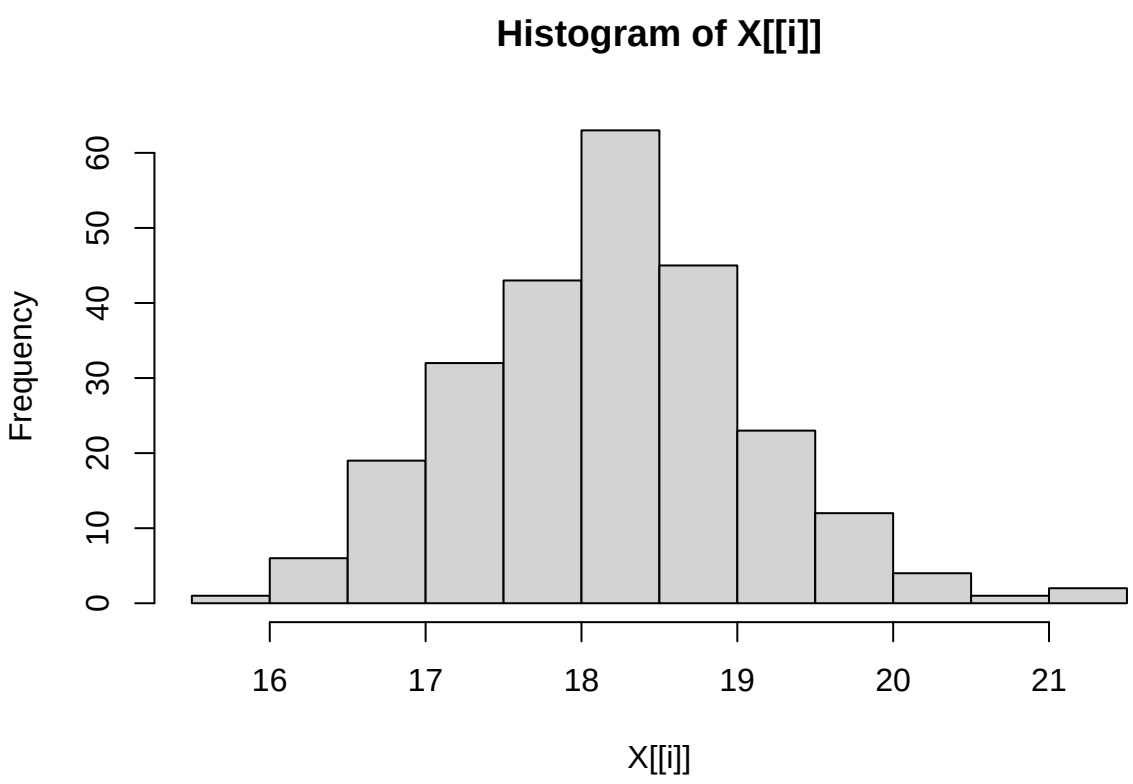

Histogram of X[[i]]
Frequency
0
20
40
60
80
50
60
70
80
90
X[[i]]


Histogram of X[[i]]
Frequency
0
20
40
60
80
35
40
45
50
X[[i]]


Histogram of X[[i]]
Frequency
0
20
40
60
80
100
20
25
30
X[[i]]


Histogram of X[[i]]
Frequency
0
10
20
30
40
50
60
70
25
30
35
40
45
X[[i]]


Histogram of X[[i]]
Frequency
0
20
40
60
80
20
25
30
35
X[[i]]




```
## $bodyfat_brozek
## $breaks
##  [1]  0  5 10 15 20 25 30 35 40 45 50
##
## $counts
##  [1]  6 29 48 51 58 37 19  2  0  1
##
## $density
##  [1] 0.0047808765 0.0231075697 0.0382470120 0.0406374502 0.0462151394
##  [6] 0.0294820717 0.0151394422 0.0015936255 0.0000000000 0.0007968127
##
## $mids
##  [1]  2.5  7.5 12.5 17.5 22.5 27.5 32.5 37.5 42.5 47.5
##
## $xname
## [1] "X[[i]]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
##
## $age
## $breaks
##  [1] 20 25 30 35 40 45 50 55 60 65 70 75 80 85
```

```
##
## $counts
##  [1] 14 24 25 28 46 38 29 12 17 11  6  0  1
##
## $density
##  [1] 0.0111553785 0.0191235060 0.0199203187 0.0223107570 0.0366533865
##  [6] 0.0302788845 0.0231075697 0.0095617530 0.0135458167 0.0087649402
## [11] 0.0047808765 0.0000000000 0.0007968127
##
## $mids
##  [1] 22.5 27.5 32.5 37.5 42.5 47.5 52.5 57.5 62.5 67.5 72.5 77.5 82.5
##
## $xname
## [1] "X[[i]]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
##
## $weight
## $breaks
##  [1] 120 140 160 180 200 220 240 260 280 300 320 340 360 380
##
## $counts
##  [1] 16 51 77 51 37 13  4  1  0  0  0  0  1
##
## $density
##  [1] 0.0031872510 0.0101593625 0.0153386454 0.0101593625 0.0073705179
##  [6] 0.0025896414 0.0007968127 0.0001992032 0.0000000000 0.0000000000
## [11] 0.0000000000 0.0000000000 0.0001992032
##
## $mids
##  [1] 130 150 170 190 210 230 250 270 290 310 330 350 370
##
## $xname
## [1] "X[[i]]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
##
## $height
## $breaks
## [1] 64 66 68 70 72 74 76 78
##
## $counts
## [1] 14 42 70 58 47 18  2
##
## $density
## [1] 0.027888446 0.083665339 0.139442231 0.115537849 0.093625498 0.035856574
```

```
## [7] 0.003984064
##
## $mids
## [1] 65 67 69 71 73 75 77
##
## $xname
## [1] "X[[i]]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
##
## $neck
## $breaks
##  [1] 30 32 34 36 38 40 42 44 46 48 50 52
##
## $counts
##  [1]  2  8 43 81 63 44  9  0  0  0  1
##
## $density
##  [1] 0.003984064 0.015936255 0.085657371 0.161354582 0.125498008 0.087649402
##  [7] 0.017928287 0.000000000 0.000000000 0.000000000 0.001992032
##
## $mids
##  [1] 31 33 35 37 39 41 43 45 47 49 51
##
## $xname
## [1] "X[[i]]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
##
## $chest
## $breaks
##  [1]  80  85  90  95 100 105 110 115 120 125 130 135 140
##
## $counts
##  [1]  1 20 44 64 54 35 15 15  1  1  0  1
##
## $density
##  [1] 0.0007968127 0.0159362550 0.0350597610 0.0509960159 0.0430278884
##  [6] 0.0278884462 0.0119521912 0.0119521912 0.0007968127 0.0007968127
## [11] 0.0000000000 0.0007968127
##
## $mids
##  [1]  82.5  87.5  92.5  97.5 102.5 107.5 112.5 117.5 122.5 127.5 132.5 137.5
##
## $xname
## [1] "X[[i]]"
```

```
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
##
## $abdomen
## $breaks
## [1]  70  80  90 100 110 120 130 140 150
##
## $counts
## [1] 29 84 82 41 12  2  0  1
##
## $density
## [1] 0.0115537849 0.0334661355 0.0326693227 0.0163346614 0.0047808765
## [6] 0.0007968127 0.0000000000 0.0003984064
##
## $mids
## [1]  75  85  95 105 115 125 135 145
##
## $xname
## [1] "X[[i]]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
##
## $hip
## $breaks
##  [1]  85  90  95 100 105 110 115 120 125 130 135 140 145 150
##
## $counts
##  [1] 16 41 84 63 30 13  2  0  1  0  0  0  1
##
## $density
##  [1] 0.0127490040 0.0326693227 0.0669322709 0.0501992032 0.0239043825
##  [6] 0.0103585657 0.0015936255 0.0000000000 0.0007968127 0.0000000000
## [11] 0.0000000000 0.0000000000 0.0007968127
##
## $mids
##  [1]  87.5  92.5  97.5 102.5 107.5 112.5 117.5 122.5 127.5 132.5 137.5 142.5
## [13] 147.5
##
## $xname
## [1] "X[[i]]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

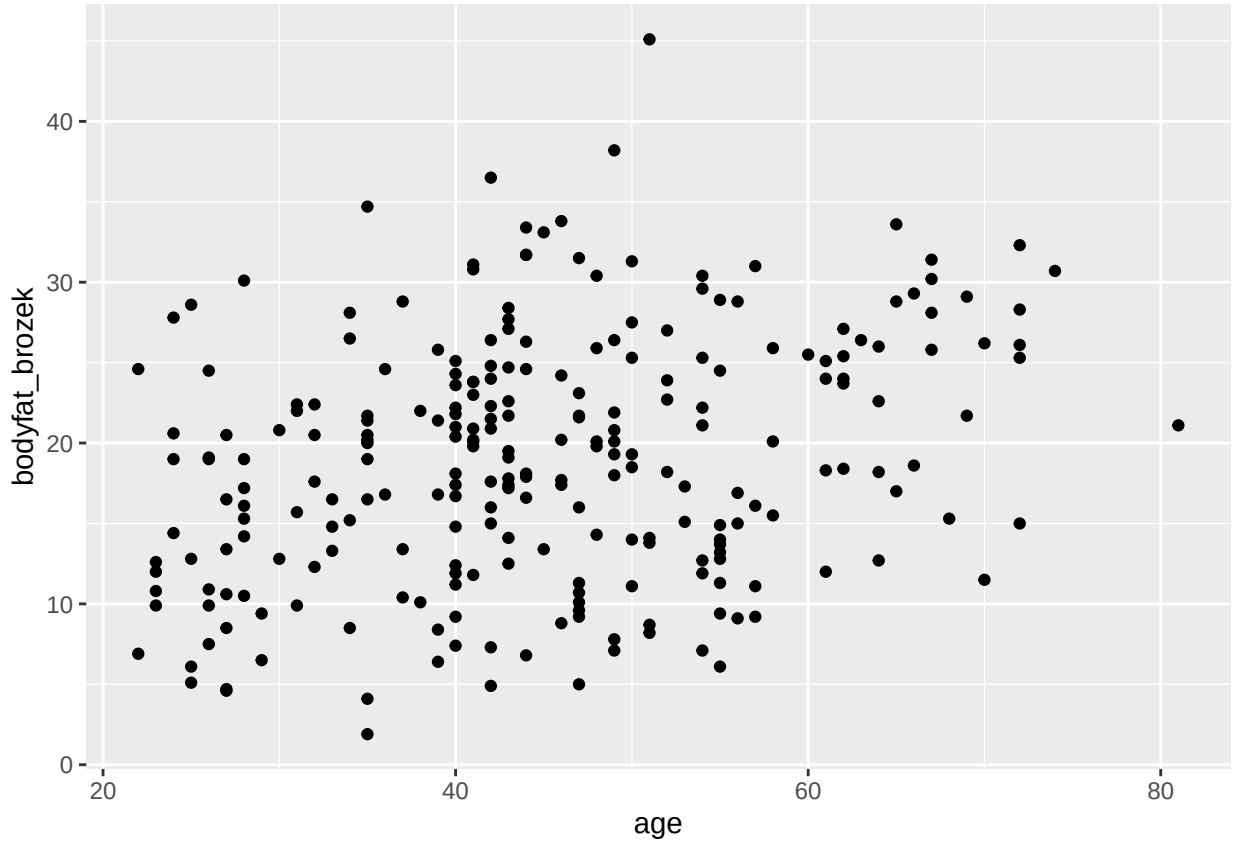
```
##
## $thigh
## $breaks
##  [1] 45 50 55 60 65 70 75 80 85 90
##
## $counts
## [1]  4 47 96 75 22  6  0  0  1
##
## $density
## [1] 0.0031872510 0.0374501992 0.0764940239 0.0597609562 0.0175298805
## [6] 0.0047808765 0.0000000000 0.0000000000 0.0007968127
##
## $mids
## [1] 47.5 52.5 57.5 62.5 67.5 72.5 77.5 82.5 87.5
##
## $xname
## [1] "X[[i]]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
##
## $knee
## $breaks
##  [1] 32 34 36 38 40 42 44 46 48 50
##
## $counts
## [1]  3 32 64 95 34 19  3  0  1
##
## $density
## [1] 0.005976096 0.063745020 0.127490040 0.189243028 0.067729084 0.037848606
## [7] 0.005976096 0.000000000 0.001992032
##
## $mids
## [1] 33 35 37 39 41 43 45 47 49
##
## $xname
## [1] "X[[i]]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
##
## $ankle
## $breaks
## [1] 18 20 22 24 26 28 30 32 34
##
## $counts
## [1]   2  63 128  52   3   1   0   2
##
```

```
## $density
## [1] 0.003984064 0.125498008 0.254980080 0.103585657 0.005976096 0.001992032
## [7] 0.000000000 0.003984064
##
## $mids
## [1] 19 21 23 25 27 29 31 33
##
## $xname
## [1] "X[[i]]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
##
## $bicep
## $breaks
##  [1] 24 26 28 30 32 34 36 38 40 42 44 46
##
## $counts
##  [1]  5 15 35 70 59 37 24  5  0  0  1
##
## $density
##  [1] 0.009960159 0.029880478 0.069721116 0.139442231 0.117529880 0.073705179
##  [7] 0.047808765 0.009960159 0.000000000 0.000000000 0.001992032
##
## $mids
##  [1] 25 27 29 31 33 35 37 39 41 43 45
##
## $xname
## [1] "X[[i]]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
##
## $forearm
## $breaks
## [1] 20 22 24 26 28 30 32 34 36
##
## $counts
## [1]  2  2 16 77 93 52  8  1
##
## $density
## [1] 0.003984064 0.003984064 0.031872510 0.153386454 0.185258964 0.103585657
## [7] 0.015936255 0.001992032
##
## $mids
## [1] 21 23 25 27 29 31 33 35
##
## $xname
```

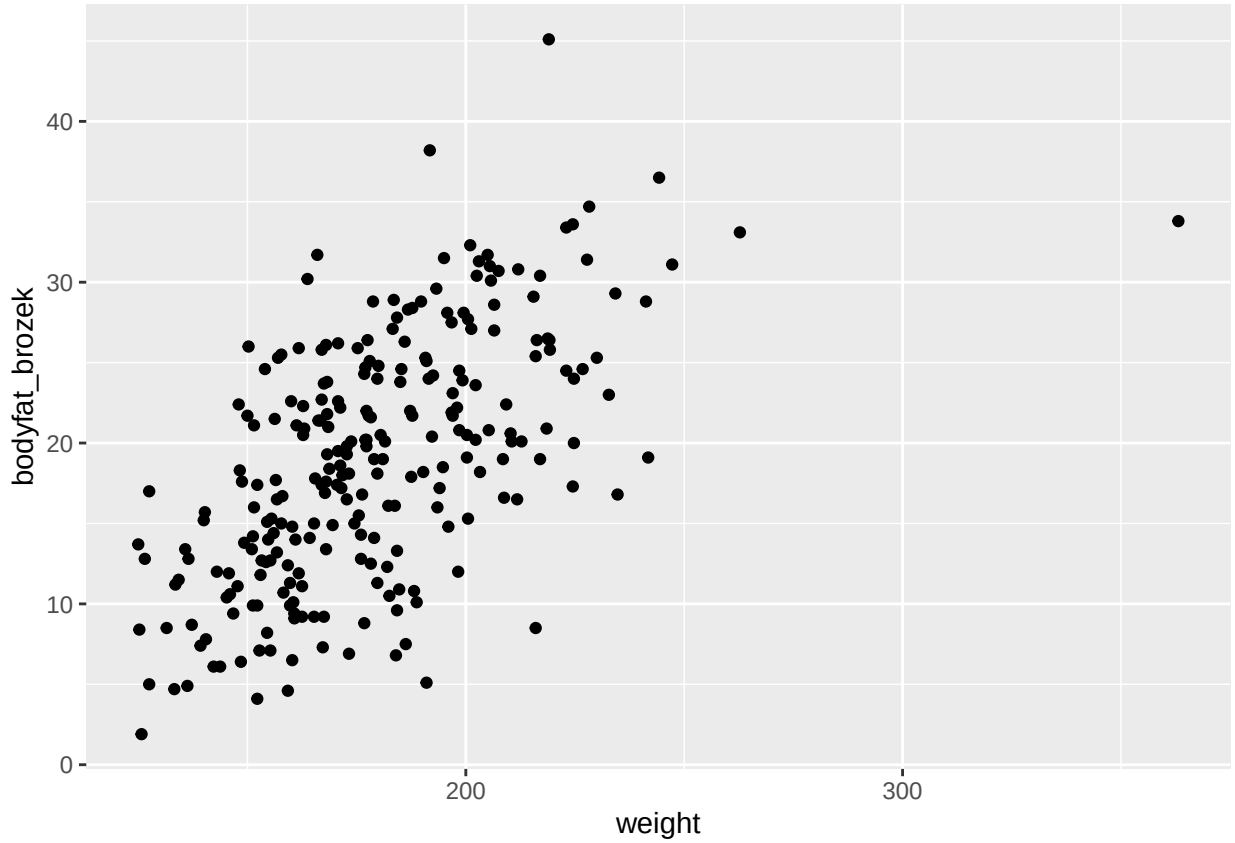
```
## [1] "X[[i]]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
##
## $wrist
## $breaks
##  [1] 15.5 16.0 16.5 17.0 17.5 18.0 18.5 19.0 19.5 20.0 20.5 21.0 21.5
##
## $counts
##  [1]  1  6 19 32 43 63 45 23 12  4  1  2
##
## $density
##  [1] 0.007968127 0.047808765 0.151394422 0.254980080 0.342629482 0.501992032
##  [7] 0.358565737 0.183266932 0.095617530 0.031872510 0.007968127 0.015936255
##
## $mids
##  [1] 15.75 16.25 16.75 17.25 17.75 18.25 18.75 19.25 19.75 20.25 20.75 21.25
##
## $xname
## [1] "X[[i]]"
##
## $equidist
## [1] TRUE
##
## attr(,"class")
## [1] "histogram"
```

```
par(mfrow = c(2,2))
lapply(colnames(bd_df)[2:length(colnames(bd_df))], function(nm){
  ggplot(bd_df) +
    geom_point(aes_string(y =colnames(bd_df)[1],
                          nm))
})
```

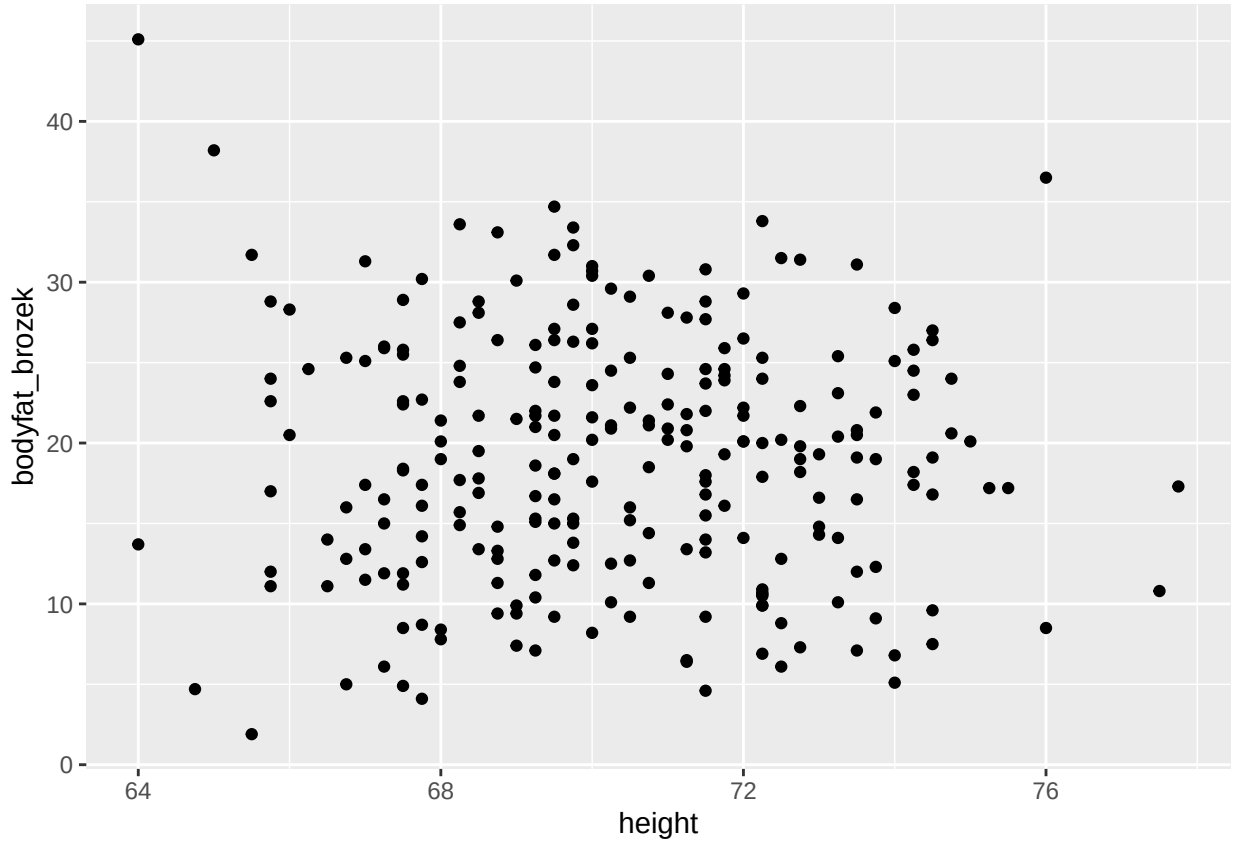
```
## [[1]]
```



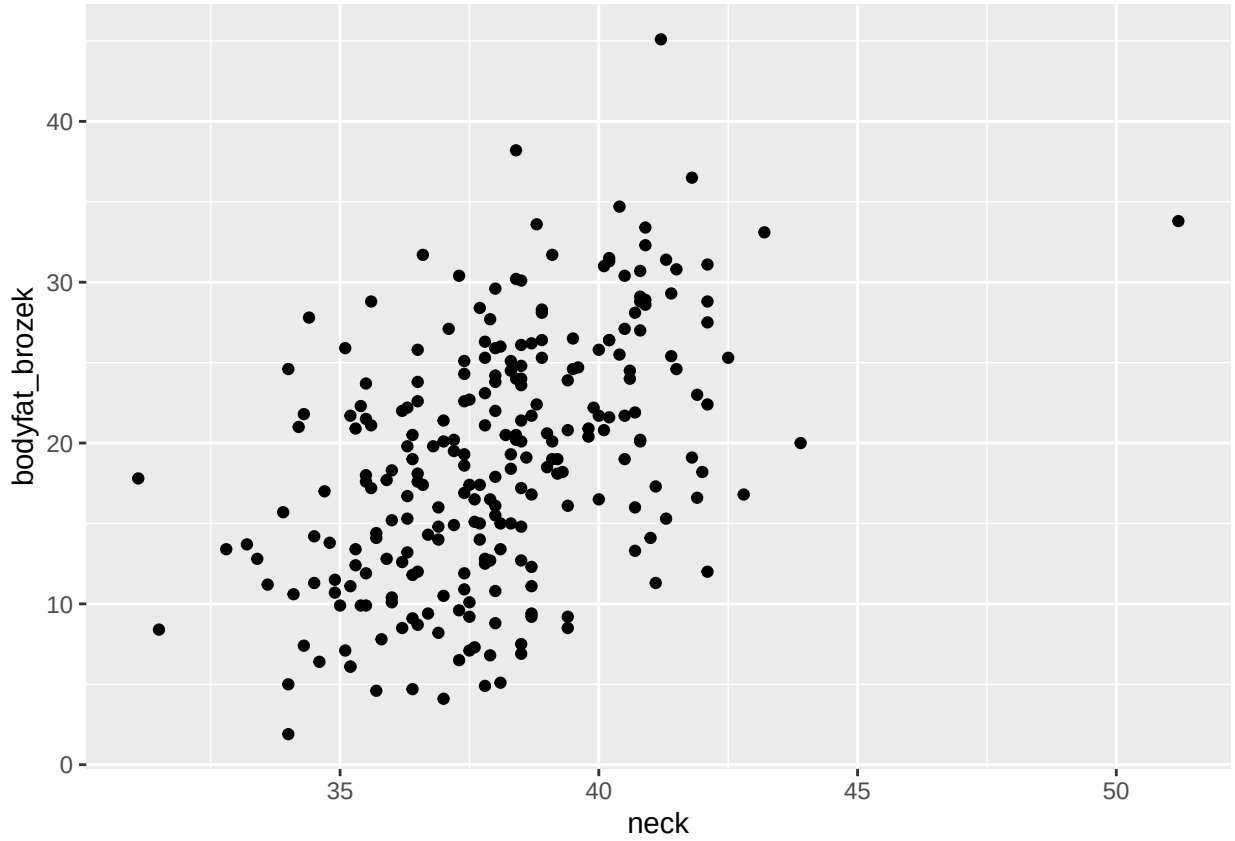

```
##
## [[2]]
```



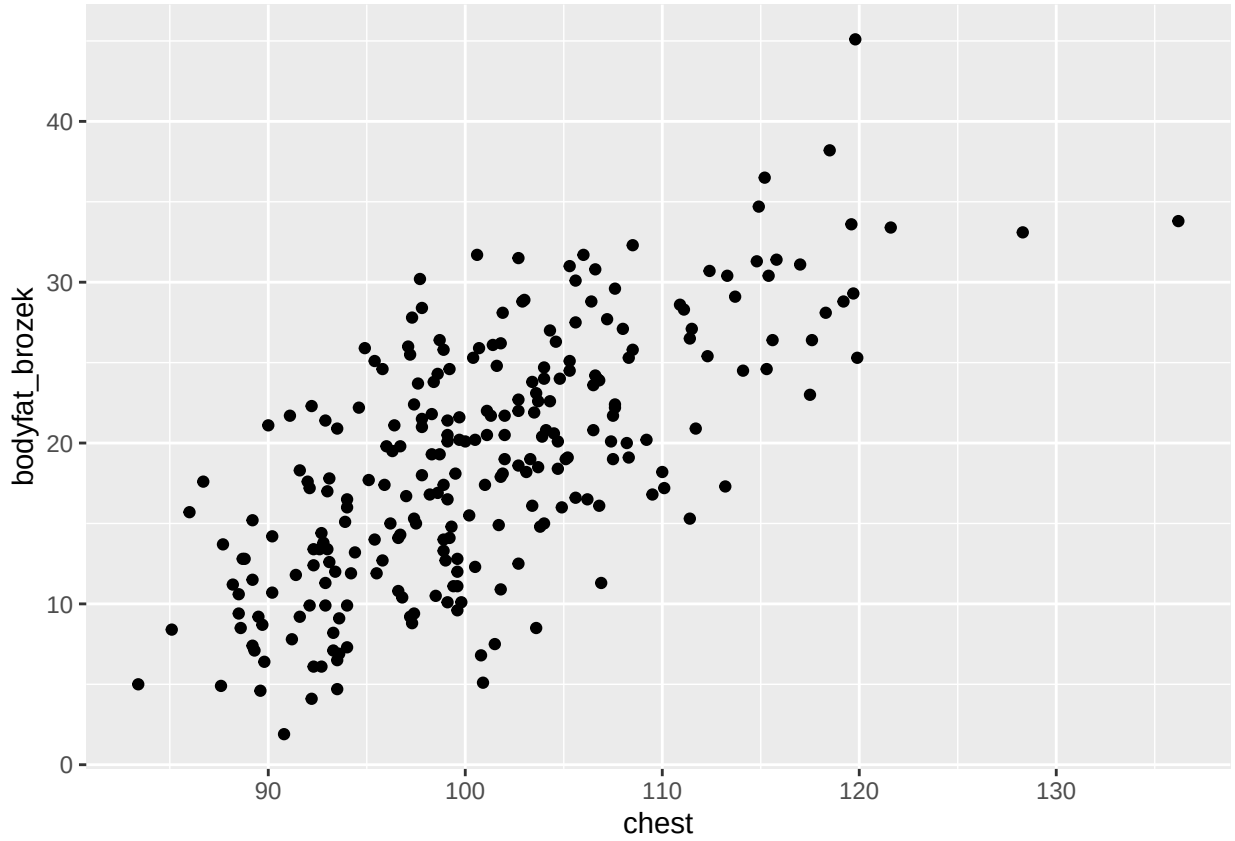

```
##
## [[3]]
```



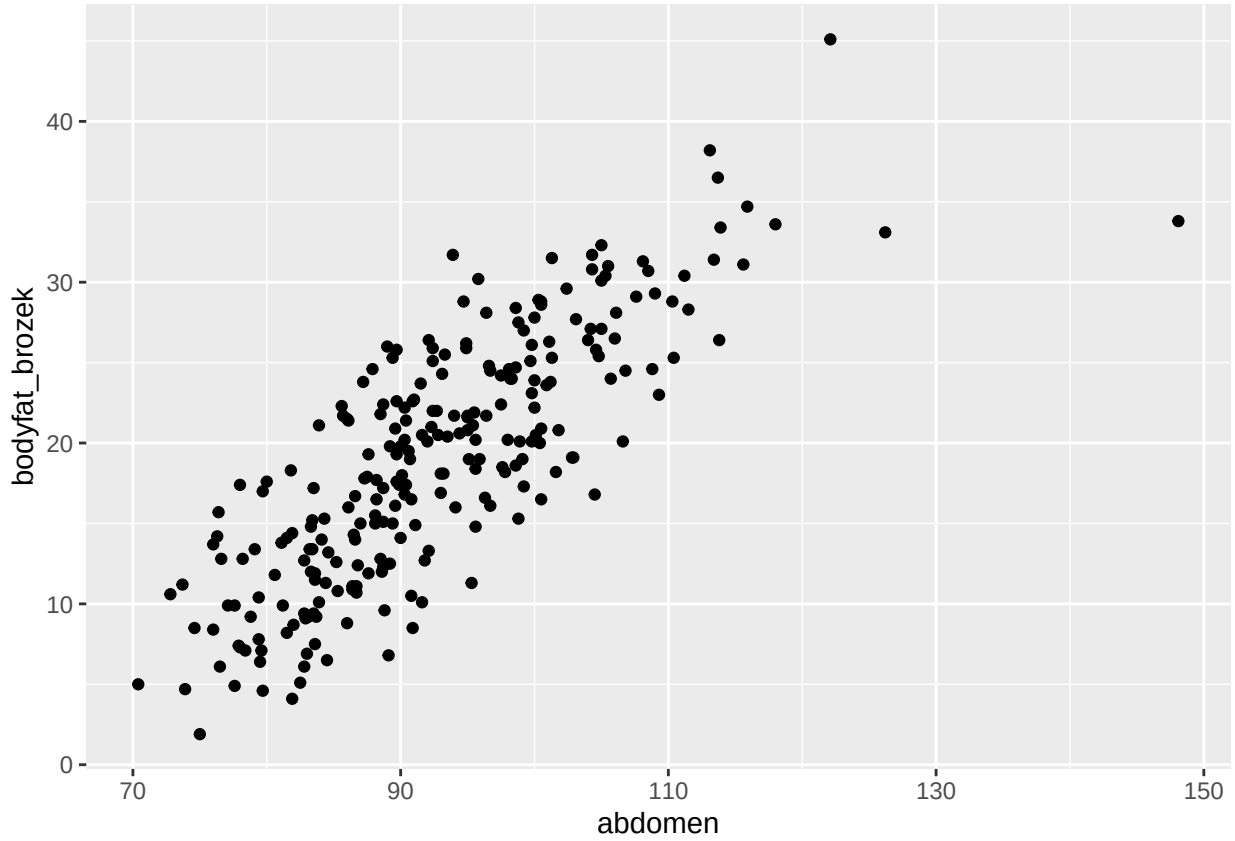

```
##
## [[4]]
```



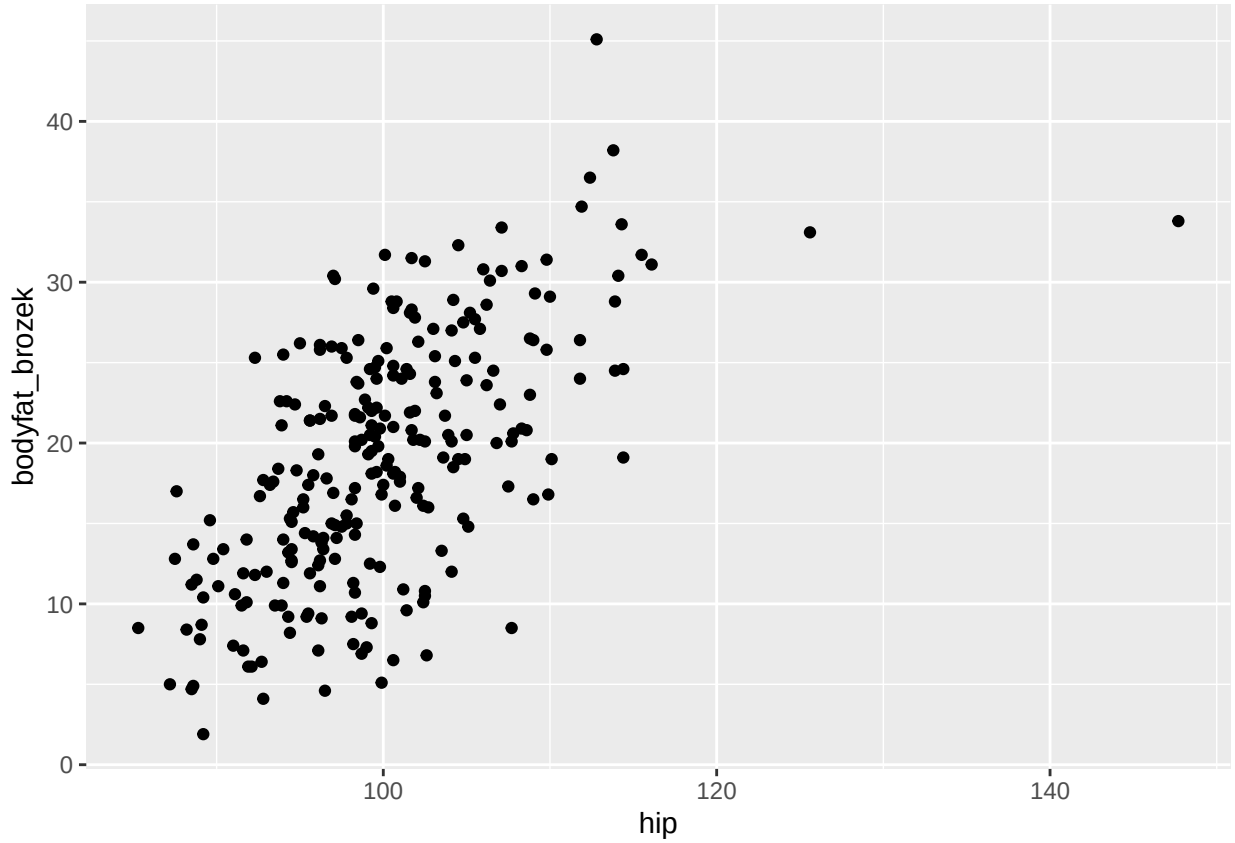

```
##
## [[5]]
```



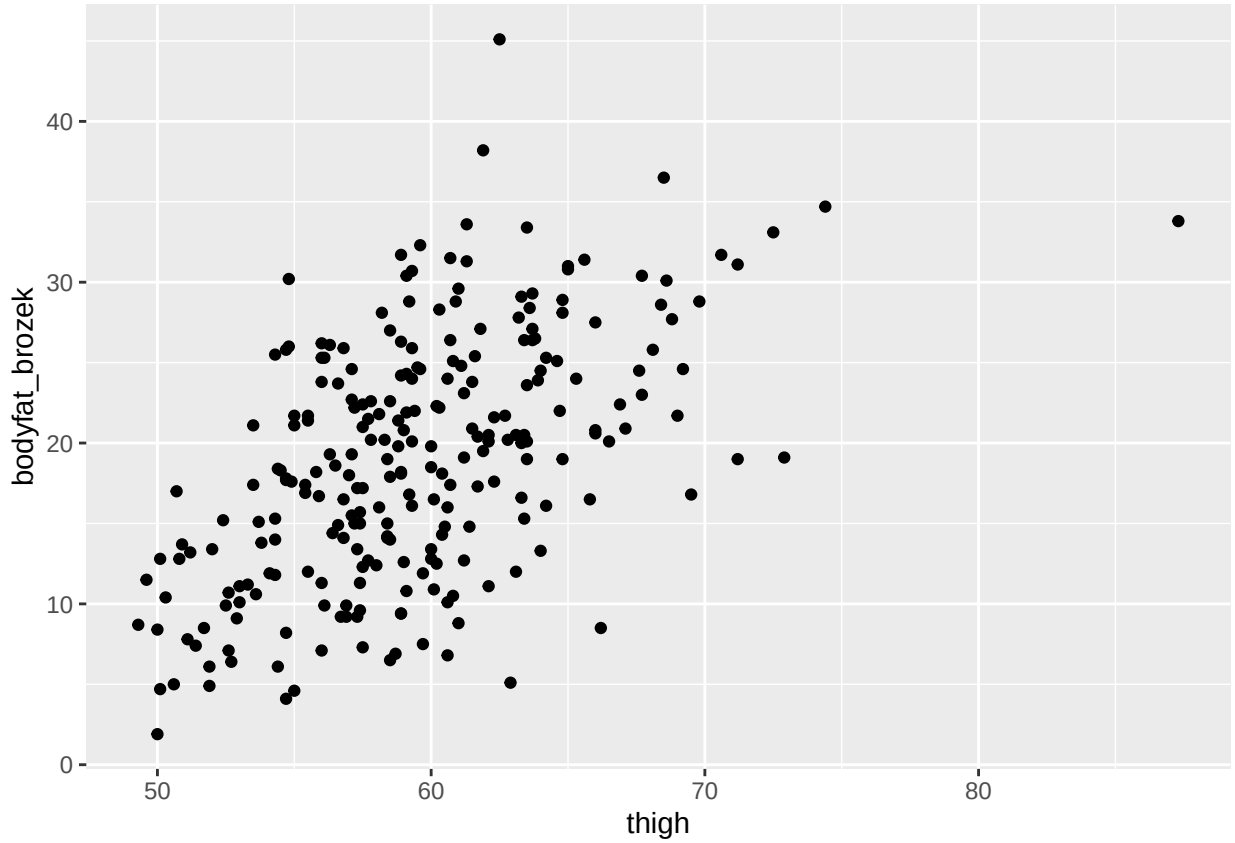

```
##
## [[6]]
```



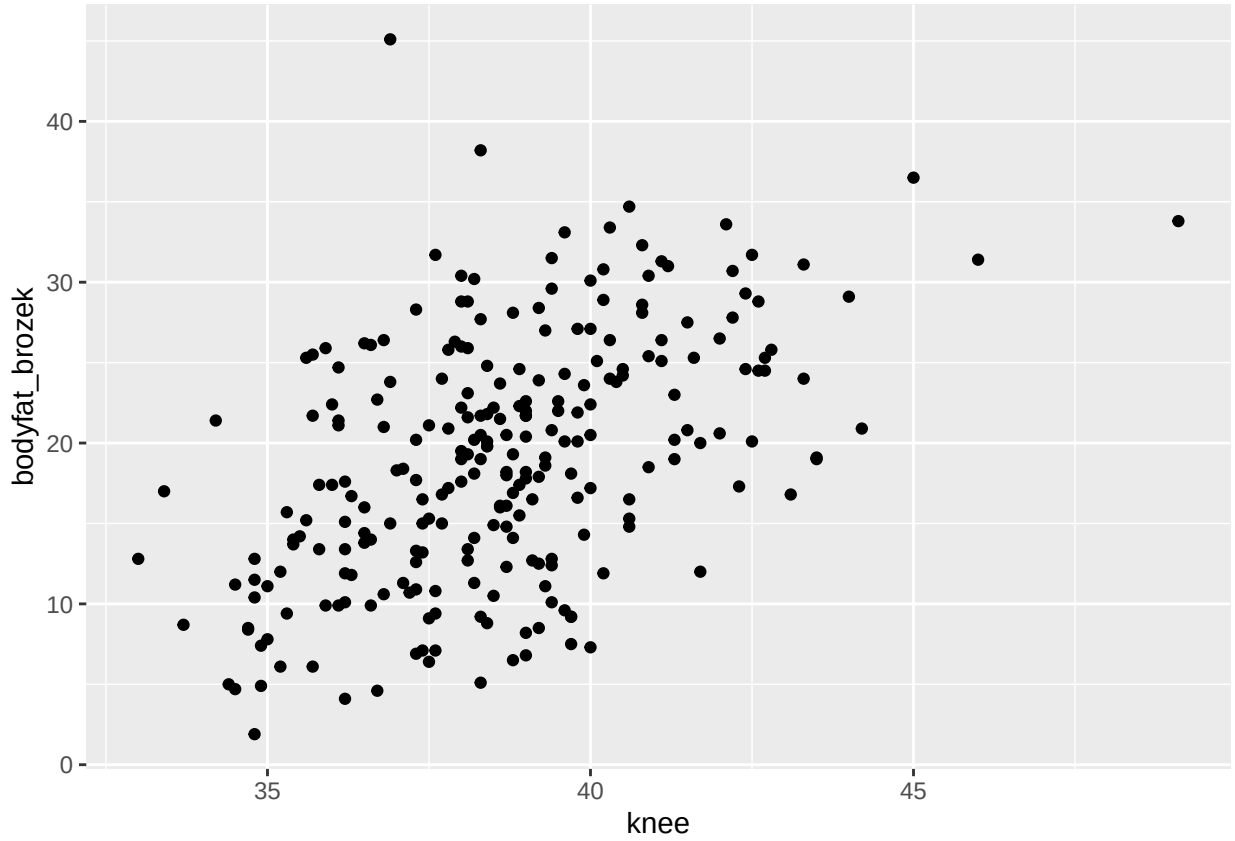

```
##
## [[7]]
```



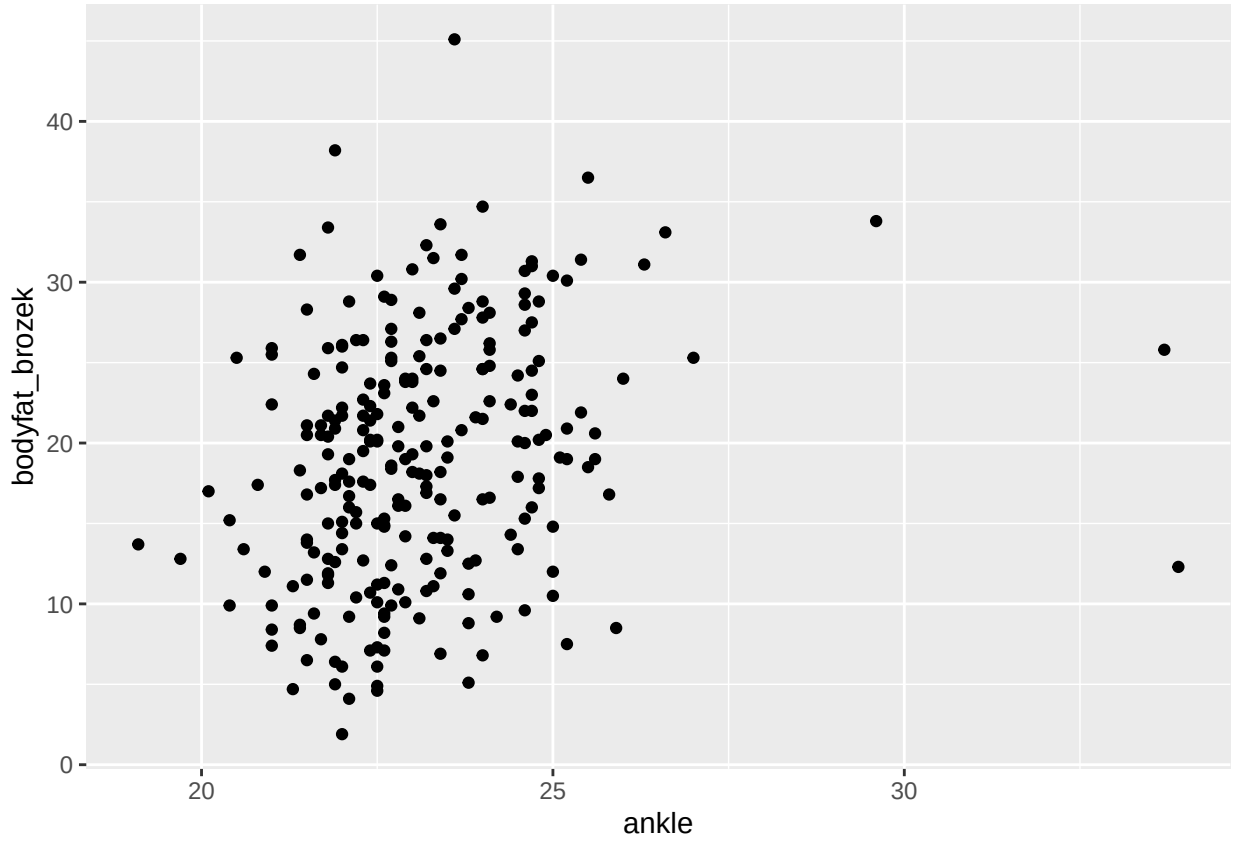

```
##
## [[8]]
```



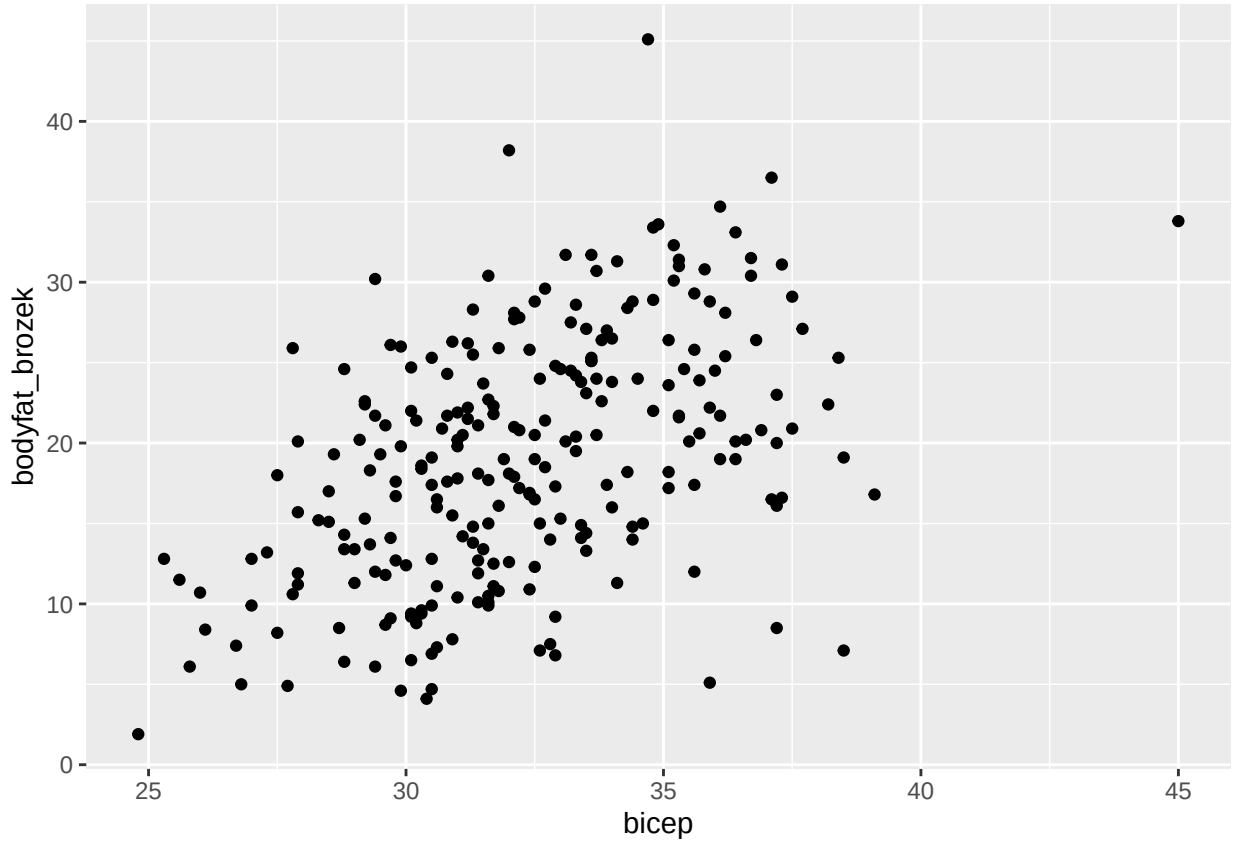

```
##
## [[9]]
```



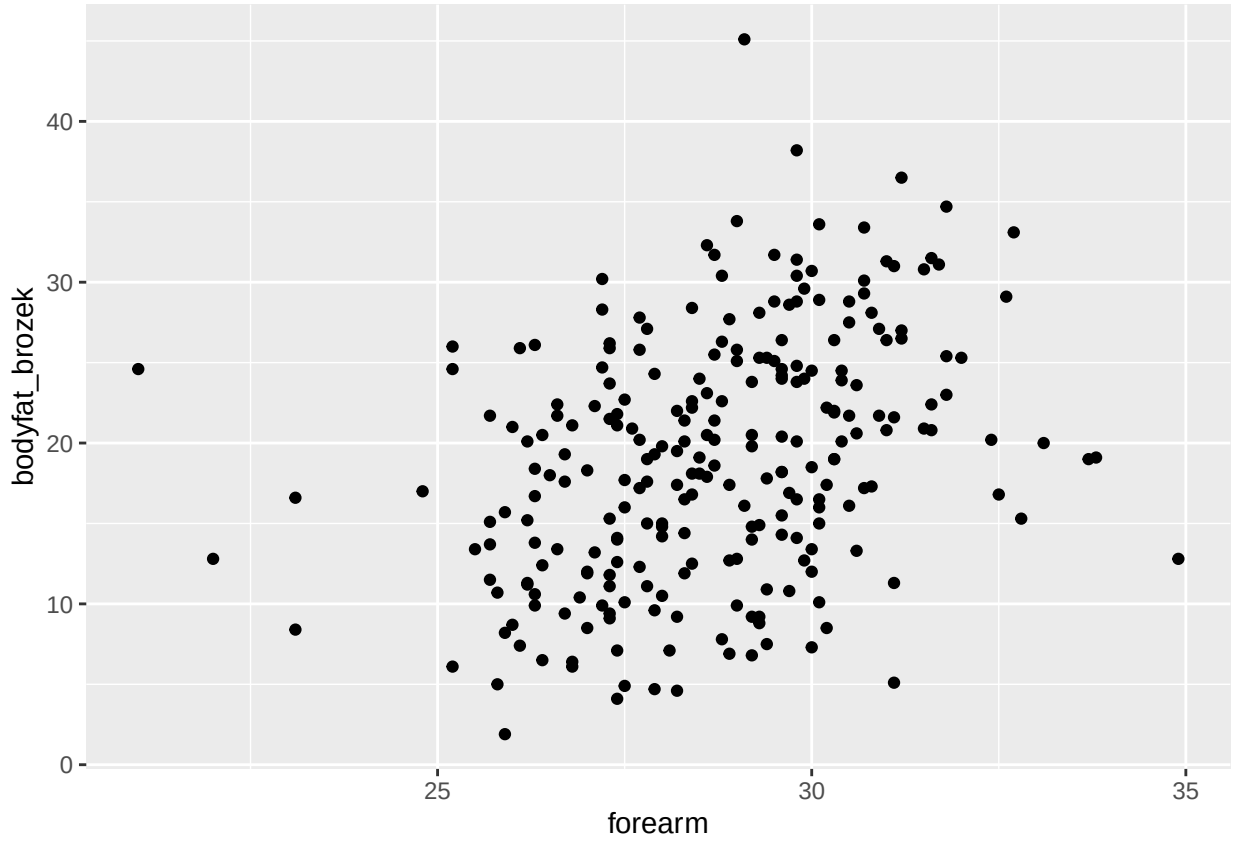

```
##
## [[10]]
```



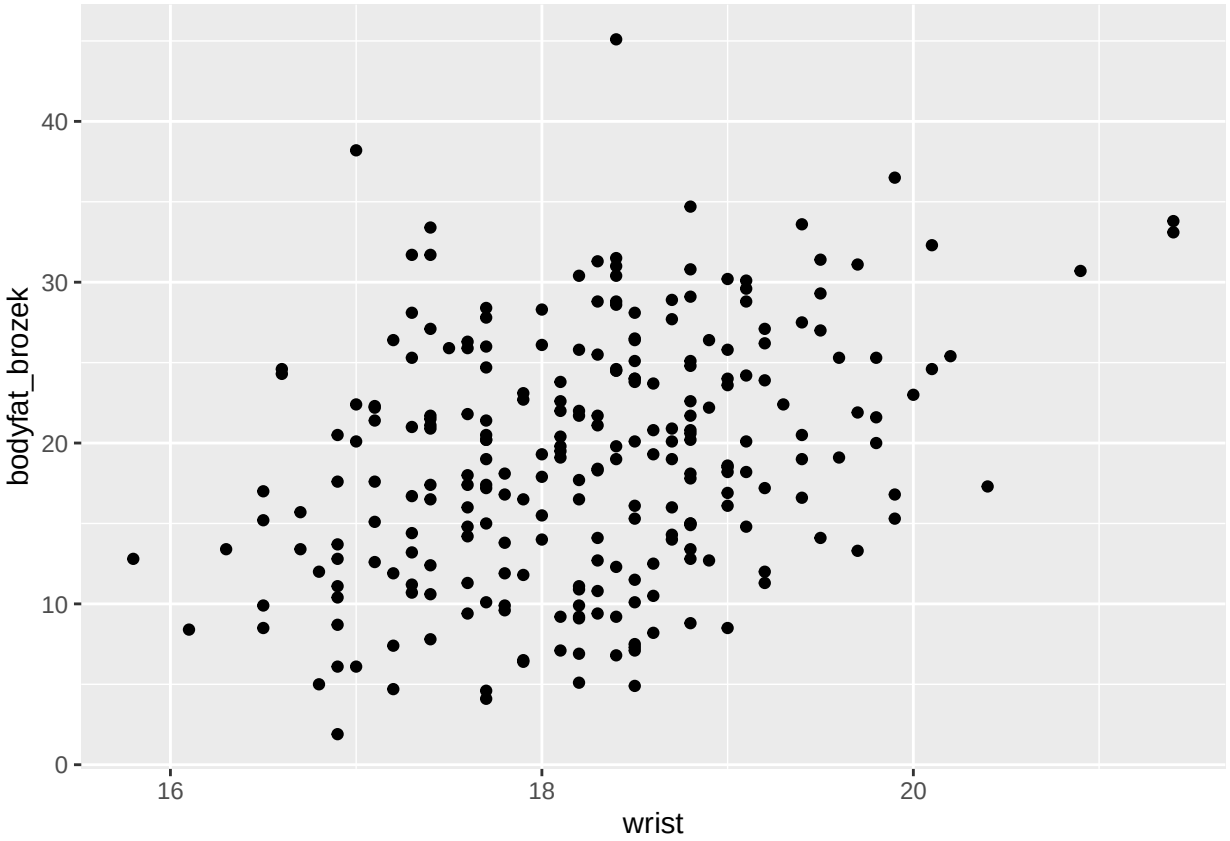

```
##
## [[11]]
```




```
##
## [[12]]
```




```
##
## [[13]]
```



From the plots, it seems that all data are symmetrically distributed. We also saw that there is no obvious relation ship between height and body fat.

```
set.seed(1)
sub<-sample(nrow(bd_df),round(nrow(bd_df)*0.8))
data_train<-bd_df[sub,]
data_test<-bd_df[-sub,]
X.test = as.matrix(data_test[,-1])
Y.test = as.matrix(data_test[,1])
X.train = as.matrix(data_train[,-1])
Y.train = as.matrix(data_train[,1])
```

Use stepwise methods

```
full_fit = lm(bodyfat_brozek ~ age + height +weight + neck + chest + abdomen + hip + thigh + knee + ank
Step_model_1 = step(full_fit, direction = "backward", trace = 0)
summary(Step_model_1)
```

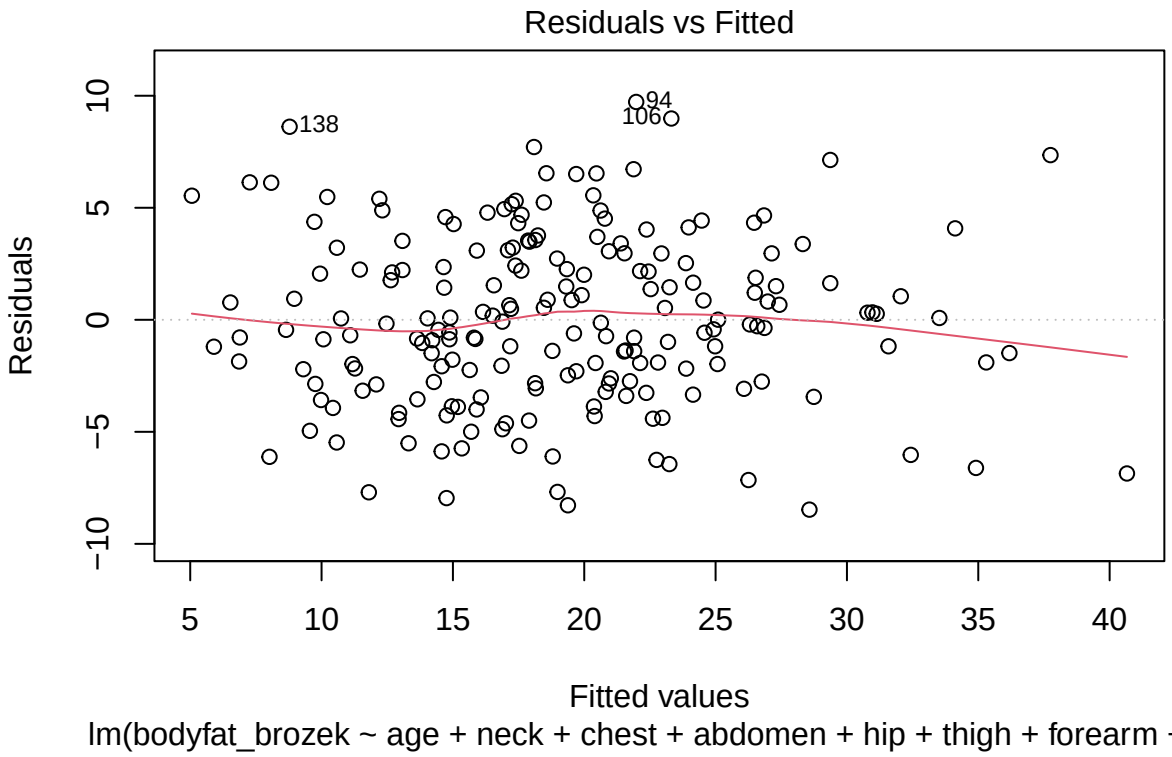
```
##
## Call:
## lm(formula = bodyfat_brozek ~ age + neck + chest + abdomen +
##     hip + thigh + forearm + wrist, data = data_train)
##
## Residuals:
```

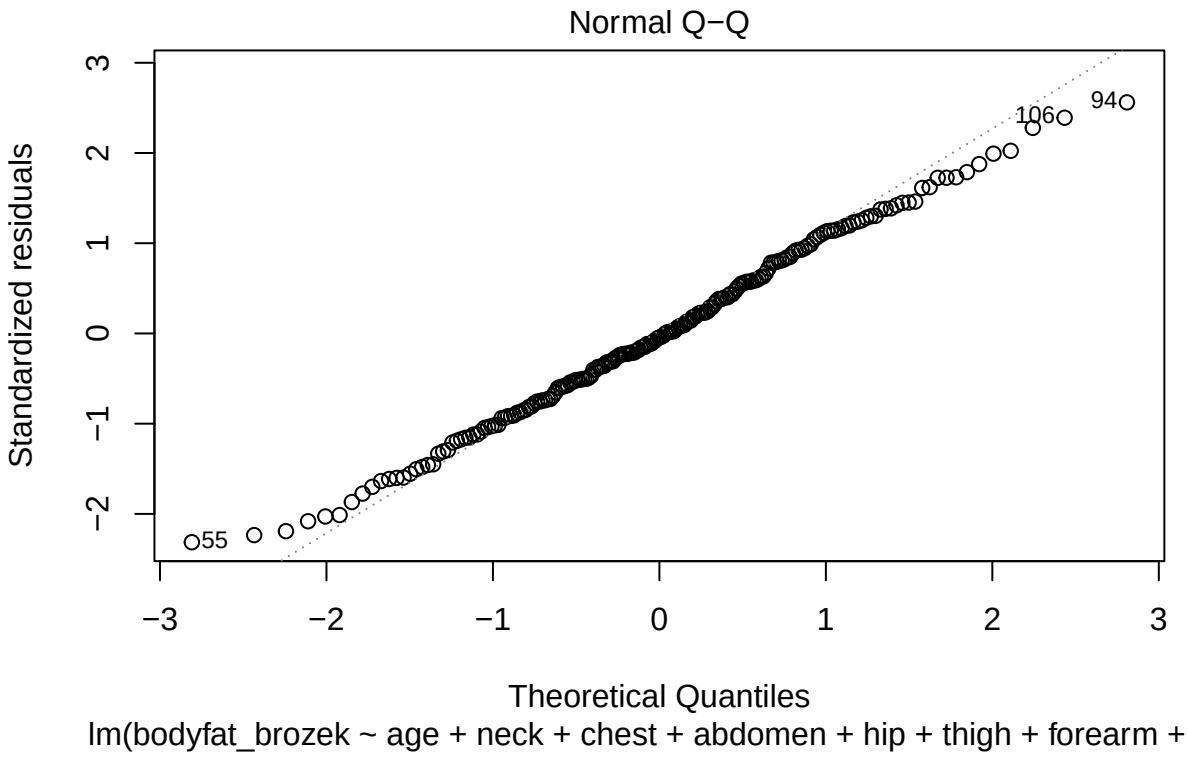
```
##     Min      1Q  Median      3Q     Max
## -8.4727 -2.7587 -0.1712  2.9613  9.7228
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.37001    6.90530   1.212  0.22696
## age          0.06877    0.03061   2.247  0.02580 *
## neck        -0.43484    0.21659  -2.008  0.04608 *
## chest       -0.16575    0.09418  -1.760  0.08001 .
## abdomen      0.91764    0.08543  10.741  < 2e-16 ***
## hip         -0.35018    0.11963  -2.927  0.00383 **
## thigh        0.23851    0.12836   1.858  0.06467 .
## forearm      0.69146    0.22744   3.040  0.00269 **
## wrist       -2.38737    0.49371  -4.836 2.72e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.865 on 192 degrees of freedom
## Multiple R-squared:  0.7565, Adjusted R-squared:  0.7463
## F-statistic: 74.55 on 8 and 192 DF,  p-value: < 2.2e-16
```

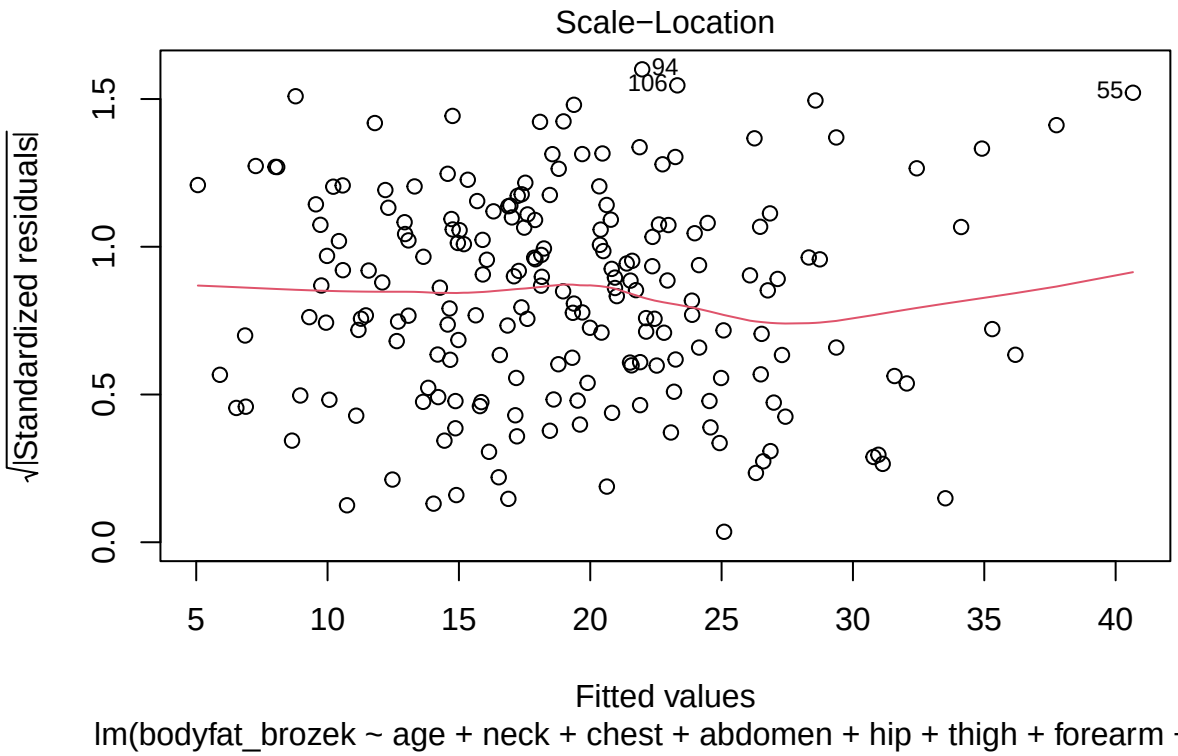
```
vif(Step_model_1)
```

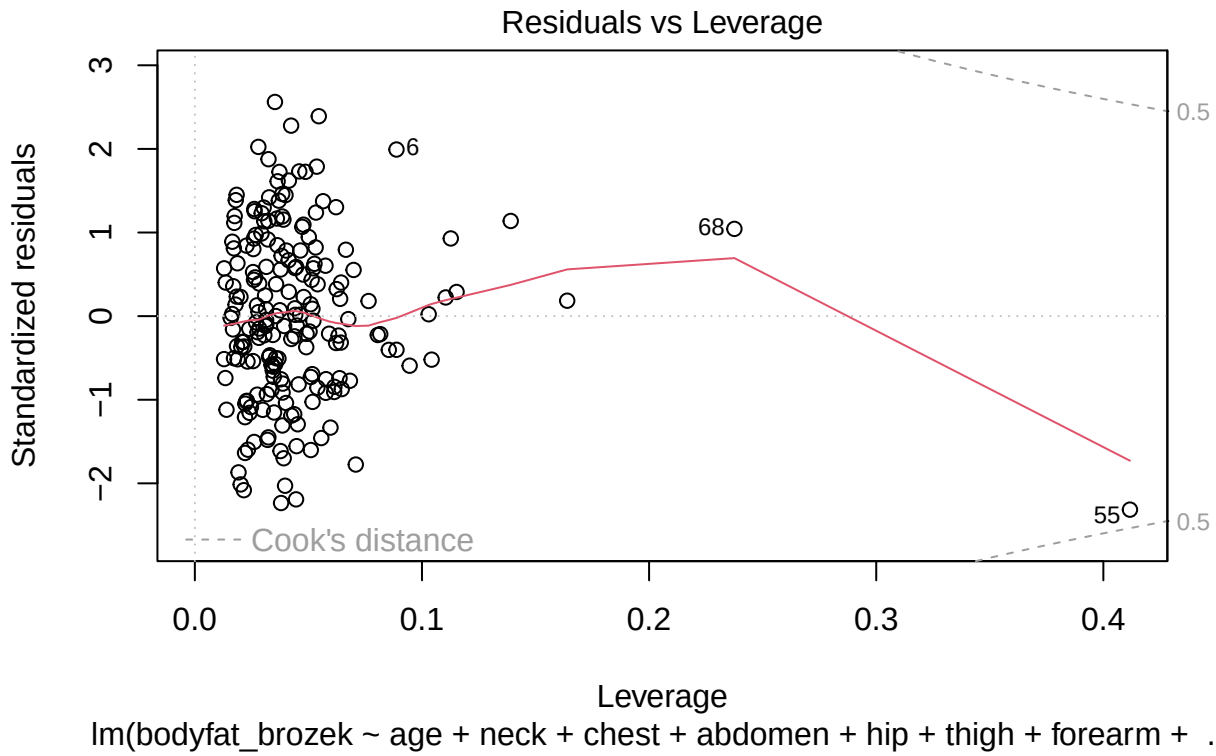
```
##       age      neck     chest   abdomen       hip     thigh   forearm     wrist
##  1.900663  3.725082  8.143448 11.570329 10.199697  6.401295  2.569941  2.663161
```

```
plot(Step_model_1)
```


Residuals vs Fitted
Residuals
Fitted values
lm(bodyfat_brozek ~ age + neck + chest + abdomen + hip + thigh + forearm + ...
138
94
106
10
5
0
−5
−10
5
10
15
20
25
30
35
40


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(bodyfat_brozek ~ age + neck + chest + abdomen + hip + thigh + forearm + ...
55
106
94
−3
−2
−1
0
1
2
3


Scale−Location
94
106
55
√|Standardized residuals|
0.0
0.5
1.0
1.5
5
10
15
20
25
30
35
40
Fitted values
lm(bodyfat_brozek ~ age + neck + chest + abdomen + hip + thigh + forearm + ...




```
RMSE_Step <- rmse(Step_model_1,data = data_test)
RMSE_Step
```

```
## [1] 4.687117
```

Although circumference of hip is included in the stepwise model, it is not significant. We then calculated VIF in our stepwise model and found that the VIF of hip is >10. So we doubted that whether it can be included in a model.

```
model2 <- lm(bodyfat_brozek ~ age  +weight + neck + abdomen  + thigh  +forearm + wrist, data = data_tra
anova(Step_model_1,model2)
```

```
## Analysis of Variance Table
##
## Model 1: bodyfat_brozek ~ age + neck + chest + abdomen + hip + thigh +
##     forearm + wrist
## Model 2: bodyfat_brozek ~ age + weight + neck + abdomen + thigh + forearm +
##     wrist
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1    192 2867.5
## 2    193 2915.2 -1   -47.756 3.1977 0.07532 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(model2)
```

```
##
## Call:
## lm(formula = bodyfat_brozek ~ age + weight + neck + abdomen +
##     thigh + forearm + wrist, data = data_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.6730 -2.6112 -0.1597  2.4925  8.6731
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -24.29952    9.16755  -2.651 0.008702 **
## age           0.06250    0.03145   1.987 0.048287 *
## weight       -0.10653    0.03503  -3.041 0.002685 **
## neck         -0.30666    0.22091  -1.388 0.166683
## abdomen       0.83136    0.06926  12.003  < 2e-16 ***
## thigh         0.20293    0.11774   1.724 0.086392 .
## forearm       0.66901    0.21950   3.048 0.002628 **
## wrist        -2.04676    0.52904  -3.869 0.000149 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.887 on 193 degrees of freedom
## Multiple R-squared:  0.7524, Adjusted R-squared:  0.7434
## F-statistic: 83.79 on 7 and 193 DF,  p-value: < 2.2e-16
```

```
RMSE_2 <- rmse(model2,data = data_test)
RMSE_2
```

```
## [1] 4.429224
```

We performed ANOVA on these two models, p-value is greater than 0.05. The ANOVA test indicates that the model not include hip circumference may be better. This model also has a lower BIC. However, it's AIC is larger than the stepwise model and adj R-squared is also a little bit smaller than the stepwise model. Therefore, we can't decided our final model yet.

```
mat = as.matrix(data_train)
leaps::leaps(x = mat[,-1], y = mat[,1], nbest = 1, method = "Cp")
```

```
## $which
##       1     2     3     4     5    6     7     8     9     A     B     C     D
## 1 FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2 FALSE  TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 3 FALSE  TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
## 4 FALSE  TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE
## 5 FALSE  TRUE FALSE FALSE  TRUE TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE
## 6  TRUE  TRUE FALSE FALSE FALSE TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE
## 7  TRUE FALSE FALSE  TRUE FALSE TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE
## 8  TRUE FALSE FALSE  TRUE  TRUE TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE
```

```
## 9   TRUE FALSE FALSE  TRUE  TRUE TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE
## 10  TRUE  TRUE FALSE  TRUE  TRUE TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE
## 11  TRUE  TRUE FALSE  TRUE  TRUE TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## 12  TRUE  TRUE FALSE  TRUE  TRUE TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## 13  TRUE  TRUE  TRUE  TRUE  TRUE TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##
## $label
##  [1] "(Intercept)" "1"           "2"           "3"           "4"
##  [6] "5"           "6"           "7"           "8"           "9"
## [11] "A"           "B"           "C"           "D"
##
## $size
##  [1]  2  3  4  5  6  7  8  9 10 11 12 13 14
##
## $Cp
##  [1] 67.593447 26.429086 15.256941  8.953374  8.759300  8.133340  8.105090
##  [8]  7.039263  7.283463  8.735947 10.389678 12.094124 14.000000
```

```
leaps::leaps(x = mat[,-1], y = mat[,1], nbest = 1, method = "adjr2")
```

```
## $which
##       1     2     3     4     5    6     7     8     9     A     B     C     D
## 1  FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 2  FALSE  TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
## 3  FALSE  TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
## 4  FALSE  TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE
## 5  FALSE  TRUE FALSE FALSE  TRUE TRUE FALSE FALSE FALSE FALSE FALSE  TRUE  TRUE
## 6   TRUE  TRUE FALSE FALSE FALSE TRUE FALSE  TRUE FALSE FALSE FALSE  TRUE  TRUE
## 7   TRUE FALSE FALSE  TRUE FALSE TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE
## 8   TRUE FALSE FALSE  TRUE  TRUE TRUE  TRUE  TRUE FALSE FALSE FALSE  TRUE  TRUE
## 9   TRUE FALSE FALSE  TRUE  TRUE TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE
## 10  TRUE  TRUE FALSE  TRUE  TRUE TRUE  TRUE  TRUE  TRUE FALSE FALSE  TRUE  TRUE
## 11  TRUE  TRUE FALSE  TRUE  TRUE TRUE  TRUE  TRUE  TRUE FALSE  TRUE  TRUE  TRUE
## 12  TRUE  TRUE FALSE  TRUE  TRUE TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## 13  TRUE  TRUE  TRUE  TRUE  TRUE TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##
## $label
##  [1] "(Intercept)" "1"           "2"           "3"           "4"
##  [6] "5"           "6"           "7"           "8"           "9"
## [11] "A"           "B"           "C"           "D"
##
## $size
##  [1]  2  3  4  5  6  7  8  9 10 11 12 13 14
##
## $adjr2
##  [1] 0.6592385 0.7133883 0.7290696 0.7385449 0.7400877 0.7422170 0.7435747
##  [8] 0.7463315 0.7473593 0.7467682 0.7458979 0.7449492 0.7437142
```

If we choose predictors based on Cp, The Cp is the smallest when the predictors are age, neck,chest, abdomen,hip, thigh, forearm and wrist. If we choose predictors based on adjust R-squared, the predictors are the same.

fitted model based on above Criterion

```
model3 <- lm(bodyfat_brozek ~ age +  neck + chest + abdomen + hip + thigh  +forearm + wrist, data = data
summary(model3)
```

```
##
## Call:
## lm(formula = bodyfat_brozek ~ age + neck + chest + abdomen +
##     hip + thigh + forearm + wrist, data = data_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.4727 -2.7587 -0.1712  2.9613  9.7228
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.37001    6.90530   1.212  0.22696
## age          0.06877    0.03061   2.247  0.02580 *
## neck        -0.43484    0.21659  -2.008  0.04608 *
## chest       -0.16575    0.09418  -1.760  0.08001 .
## abdomen      0.91764    0.08543  10.741  < 2e-16 ***
## hip         -0.35018    0.11963  -2.927  0.00383 **
## thigh        0.23851    0.12836   1.858  0.06467 .
## forearm      0.69146    0.22744   3.040  0.00269 **
## wrist       -2.38737    0.49371  -4.836 2.72e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.865 on 192 degrees of freedom
## Multiple R-squared:  0.7565, Adjusted R-squared:  0.7463
## F-statistic: 74.55 on 8 and 192 DF,  p-value: < 2.2e-16
```

```
RMSE_3 <- rmse(model3,data = data_test)
RMSE_3
```

```
## [1] 4.687117
```

LASSO

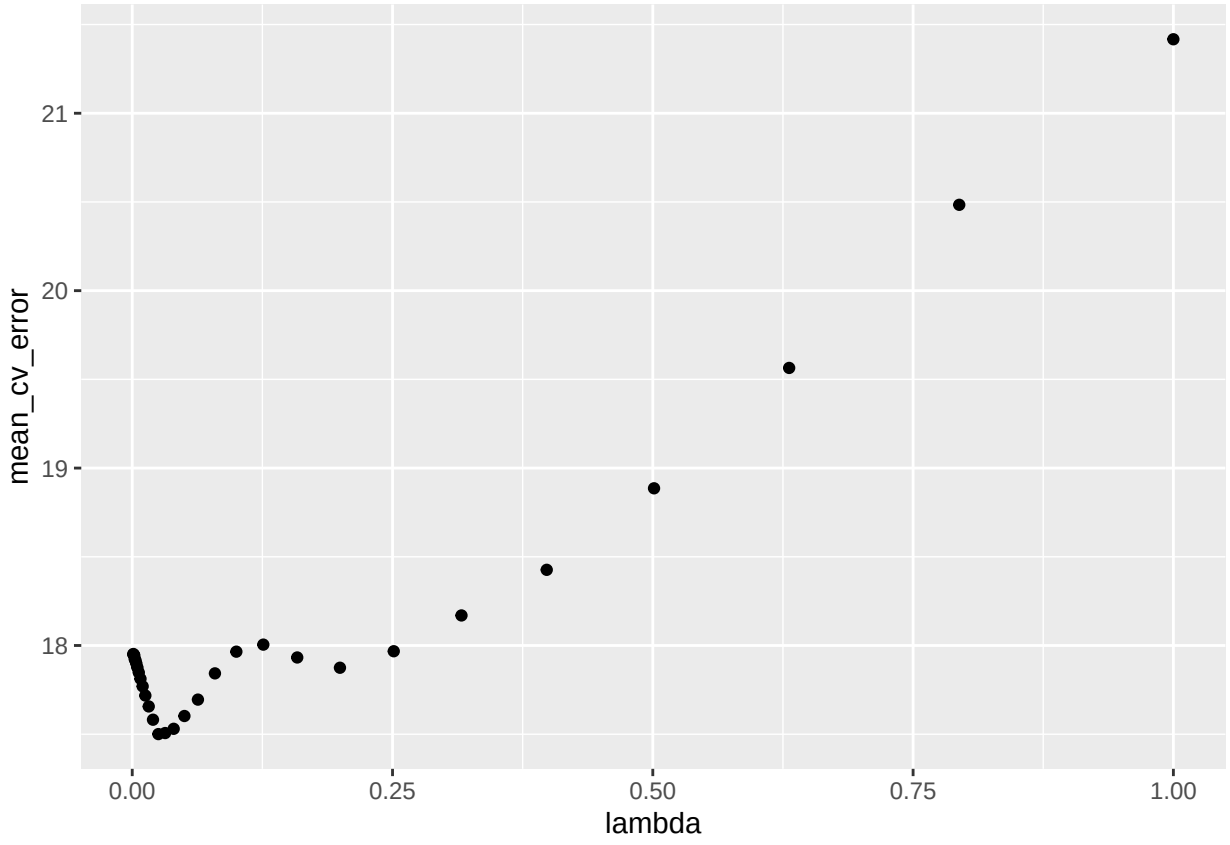
We also want to use LASSO to generate a reasonable model. First, we want to choose the best lambda. So we conducted a cross validation. The cross validation's result shows that the best lambda is 0.04,which is small. This indicates that we may include many variables in our model if we using LASSO. So we fit a model with LASSO Regression.

```
set.seed(123)
lambda_seq <- 10^seq(-3, 0, by = .1)
cv_object <- cv.glmnet(as.matrix(data_train[,-1]), data_train$bodyfat_brozek, lambda = lambda_seq,
nfolds = 5)
cv_object
```

```
##
## Call:  cv.glmnet(x = as.matrix(data_train[, -1]), y = data_train$bodyfat_brozek,      lambda = lambda
##
```

```
## Measure: Mean-Squared Error
##
##     Lambda Index Measure    SE Nonzero
## min 0.0251    17   17.50 1.887      12
## 1se 0.5012     4   18.89 1.377       4
```

```
tibble(lambda = cv_object$lambda,
mean_cv_error = cv_object$cvm) %>%
ggplot(aes(x = lambda, y =
mean_cv_error)) +
geom_point()
```




```
cv_object$lambda.min
```

```
## [1] 0.02511886
```

```
fit_bestcv <- glmnet(as.matrix(data_train[,-1]), data_train$bodyfat_brozek, lambda = cv_object$lambda.mi
best_lambda = cv_object$lambda.min
coef(fit_bestcv)
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept) -2.66318546
## age          0.06305549
```

```
## weight      -0.03981678
## height       .
## neck        -0.35748584
## chest       -0.11450449
## abdomen      0.88564322
## hip         -0.19777927
## thigh        0.22445962
## knee        -0.18799617
## ankle        0.04664407
## bicep        0.07792838
## forearm      0.61660889
## wrist       -2.13035975
```

```
lasso_predict_train = predict(fit_bestcv, s = best_lambda, newx = X.train)
lasso_predict_test = predict(fit_bestcv, s = best_lambda, newx = X.test)
RMSE_LASSO = sqrt(mean((Y.test-lasso_predict_test)^2))
SSE_LASSO = sum((Y.train - lasso_predict_train )^2)
SSTO_LASSO = sum((Y.train - mean(Y.train))^2)
n = nrow(data_train)
p = 12
adj_Rsquared_LASSO_trained = 1 - (SSE_LASSO/SSTO_LASSO)*((n-1)/(n-p-1))
adj_Rsquared_LASSO_trained
```

```
## [1] 0.7439698
```

```
RMSE_LASSO
```

```
## [1] 4.578159
```