Exercise 3

for Advanced Methods for Regression and Classification

Dzhamilia Kulikieva

06.11.2024

Let's use the data set "building.RData" from the last exercise, and make the same training/test split

```
load("/Users/djem/Downloads/building.RData")
head(df)
```

```
##          y START.YEAR START.QUARTER COMPLETION.YEAR COMPLETION.QUARTER PhysFin1
## 1 7.696213         81             1              85                  1        1
## 2 8.517193         84             1              89                  4        1
## 3 7.090077         78             1              81                  4        1
## 4 5.105945         72             2              73                  2        1
## 5 8.612503         87             1              90                  2        1
## 6 8.556414         87             1              90                  1        1
##   PhysFin2 PhysFin3 PhysFin4 PhysFin5 PhysFin6 PhysFin7 PhysFin8   Econ1 Econ2
## 1     3150      920    598.5      190  1010.84       16     1200 6713.00  56.2
## 2     7600     1140   3040.0      400   963.81       23     2900 3152.00 106.0
## 3     4800      840    480.0      100   689.84       15      630 1627.00  41.0
## 4      685      202     13.7       20   459.54        4      140 2580.93  12.1
## 5     3000      800   1230.0      410   631.91       13     5000 6790.00 203.8
## 6     2500      640   1050.0      420   647.32       12     4800 6790.00 203.8
##    Econ3 Econ4      Econ5   Econ6 Econ7  Econ8   Econ9 Econ10 Econ11 Econ12
## 1  61.52  6.11  320957.30  3485.8  64.5 239.50 12456.6     15  797.3  809.8
## 2 103.03  3.15  685697.50  3526.1 105.5 208.80 17584.3     15 1408.4 1473.5
## 3  41.25  1.74  160401.50  1217.5  34.4 285.80  6489.1     15  614.0  608.2
## 4  10.03  1.24   38193.64   287.2  13.6  17.03   154.4     12  183.6  211.1
## 5 162.84  6.46 1640293.00 10855.3 229.3 393.30 69444.8     11 2738.8 3148.0
## 6 162.84  6.46 1640293.00 10855.3 229.3 393.30 69444.8     11 2738.8 3148.0
##    Econ13 Econ14 Econ15 Econ16   Econ17   Econ18    Econ19 Econ1.lag1
## 1 1755.00   8003  67.81  63.25  3758.77 42587.00  628132.9       4986
## 2 8842.18   8864 105.52 105.32 12113.01 45966.00 1188995.8       2700
## 3 1755.00   7773  45.91  38.34  1537.96 39066.00  524764.8       1580
## 4 1612.95   1649  11.62  10.06   392.96  8435.75  141542.6       2952
## 5 9248.40   9380 158.63 169.50 10082.00 49572.00 2318397.0       6370
## 6 9248.40   9380 158.63 169.50 10082.00 49572.00 2318397.0       6370
##   Econ2.lag1 Econ3.lag1 Econ4.lag1 Econ5.lag1 Econ6.lag1 Econ7.lag1 Econ8.lag1
## 1       55.5      60.78       3.94   297210.1     3663.5      61.50     179.63
## 2      103.0     101.84       2.65   625829.2     4386.9     100.40     156.60
## 3       40.3      40.84       1.15   150266.8     1149.5      34.10     214.35
## 4       11.6       8.50       1.99    35859.4      322.5      12.67      56.60
## 5      190.3     154.36       5.33  1523166.6    12930.0     210.70     294.98
## 6      190.3     154.36       5.33  1523166.6    12930.0     210.70     294.98
##   Econ9.lag1 Econ10.lag1 Econ11.lag1 Econ12.lag1 Econ13.lag1 Econ14.lag1
## 1    9342.45          15    757.8000      861.80     1755.00        8018
```

```
## 2   13188.23          15   1424.1000     1584.30     8776.71        8799
## 3    4866.83          15    573.7265      680.29     1755.00        6714
## 4     610.40          12    165.1000      208.60     1504.36        1582
## 5   52083.60          11   2595.2000     3000.00     9329.64        9396
## 6   52083.60          11   2595.2000     3000.00     9329.64        9396
##   Econ15.lag1 Econ16.lag1 Econ17.lag1 Econ18.lag1 Econ19.lag1 Econ1.lag2
## 1       65.00       60.53     3538.71    31940.25    610502.7        6788
## 2      101.00      101.89    13571.80    34474.50   1067772.0        3561
## 3       43.40       36.45     1535.16    29299.50    466212.2        2628
## 4       10.86        9.79      435.10    32776.00    129102.4        2649
## 5      148.76      159.00     9700.00    37179.00   1908975.7        5909
## 6      148.76      159.00     9700.00    37179.00   1908975.7        5909
##   Econ2.lag2 Econ3.lag2 Econ4.lag2 Econ5.lag2 Econ6.lag2 Econ7.lag2 Econ8.lag2
## 1       54.2      59.40       5.41   280451.7     3755.8      58.10     119.75
## 2       98.2      98.64       2.76   602224.7     3819.0      97.20     104.40
## 3       39.3      40.21       1.52   143737.7     1284.5      33.50     142.90
## 4       11.4       6.97       2.25    32793.7      388.9      11.73      42.45
## 5      177.6     147.44       6.88  1451175.9     8146.1     188.90     196.65
## 6      177.6     147.44       6.88  1451175.9     8146.1     188.90     196.65
##   Econ9.lag2 Econ10.lag2 Econ11.lag2 Econ12.lag2 Econ13.lag2 Econ14.lag2
## 1    6228.30          15     795.000      818.50     1755.00        8001
## 2    8792.15          15    1298.800     1389.60     8699.73        8735
## 3    3244.55          15     554.082      663.97     1755.00        5827
## 4     457.80          12     167.900      209.60     1450.47        1507
## 5   34722.40          11    2284.400     2627.50     9297.06        9347
## 6   34722.40          11    2284.400     2627.50     9297.06        9347
##   Econ15.lag2 Econ16.lag2 Econ17.lag2 Econ18.lag2 Econ19.lag2 Econ1.lag3
## 1       63.69       58.55     3347.72     21293.5    589389.6        5728
## 2       98.12       98.45    13596.37     22983.0    973523.7        3157
## 3       41.79       34.76     1527.55     19533.0    409677.9        2374
## 4       10.17        9.35      508.64     24582.0    123618.0        2312
## 5      140.90      146.20    10149.00     24786.0   1681849.3        7045
## 6      140.90      146.20    10149.00     24786.0   1681849.3        7045
##   Econ2.lag3 Econ3.lag3 Econ4.lag3 Econ5.lag3 Econ6.lag3 Econ7.lag3 Econ8.lag3
## 1       52.4      57.65       5.40  262789.00     2931.4      54.20       59.88
## 2       92.8      96.49       3.05  552124.40     3896.7      96.90       52.20
## 3       38.0      39.43       0.92  134548.40     1191.1      33.70       71.45
## 4       10.6       5.44       2.58   30012.46      345.3      10.79       28.30
## 5      160.0     141.34       4.72 1341072.80     8245.0     173.80       98.33
## 6      160.0     141.34       4.72 1341072.80     8245.0     173.80       98.33
##   Econ9.lag3 Econ10.lag3 Econ11.lag3 Econ12.lag3 Econ13.lag3 Econ14.lag3
## 1    3114.15          15    746.8000      815.50     1755.00        8013
## 2    4396.08          15   1294.2000     1288.00     8555.54        8585
## 3    1622.28          15    574.6000      680.50     1755.00        5565
## 4     305.20          12    180.3715      158.45     1439.00        1450
## 5   17361.20          11   2451.2000     2526.40     9254.28        9306
## 6   17361.20          11   2451.2000     2526.40     9254.28        9306
##   Econ15.lag3 Econ16.lag3 Econ17.lag3 Econ18.lag3 Econ19.lag3 Econ1.lag4
## 1       62.78       56.45     3387.72    10646.75    606524.2        7196
## 2       95.35       94.34    12063.50    11491.50    954628.6        3678
## 3       41.03       33.37     1601.79     9766.50    403875.0        2693
## 4        9.91        8.85      590.64    16388.00    121857.2        1381
## 5      136.56      138.80     9291.00    12393.00   1732937.5        5606
## 6      136.56      138.80     9291.00    12393.00   1732937.5        5606
```

```
##   Econ2.lag4 Econ3.lag4 Econ4.lag4 Econ5.lag4 Econ6.lag4 Econ7.lag4 Econ8.lag4
## 1       51.3      56.13       5.97  249110.70     2562.3      52.80     217.00
## 2       86.2      83.21       3.25  526596.40     2790.6      94.10     334.80
## 3       36.2      37.64       1.55  134312.50     1529.0      31.43     175.70
## 4       10.0       3.91       3.00   27231.21      316.5       9.85      14.15
## 5      149.1     134.80       4.09 1284199.40     6622.5     147.60     432.40
## 6      149.1     134.80       4.09 1284199.40     6622.5     147.60     432.40
##   Econ9.lag4 Econ10.lag4 Econ11.lag4 Econ12.lag4 Econ13.lag4 Econ14.lag4
## 1    10445.6          15    733.8000      815.50     1755.00        8002
## 2    14488.6          15   1143.8000     1316.30     8364.78        8393
## 3     3994.7          15    589.5000      765.80     1755.00        4930
## 4      152.6          12    197.6796      152.25     1442.31        1456
## 5    73143.5          14   2220.6000     2244.10     9231.76        9286
## 6    73143.5          14   2220.6000     2244.10     9231.76        9286
##   Econ15.lag4 Econ16.lag4 Econ17.lag4 Econ18.lag4 Econ19.lag4
## 1       60.74       54.26     2978.26       41407    601988.1
## 2       90.95       89.79    11379.37       44835    929027.1
## 3       38.70       32.04     1653.06       37933    377828.6
## 4        9.73        8.34      686.16        8194    122031.7
## 5      136.60      140.20     9821.00       48260   1734973.5
## 6      136.60      140.20     9821.00       48260   1734973.5
```

```r
set.seed(2024)
sample_index <- sample(1:nrow(df), size = floor(2/3 * nrow(df)))
train_data <- df[sample_index, ]
test_data <- df[-sample_index, ]
```

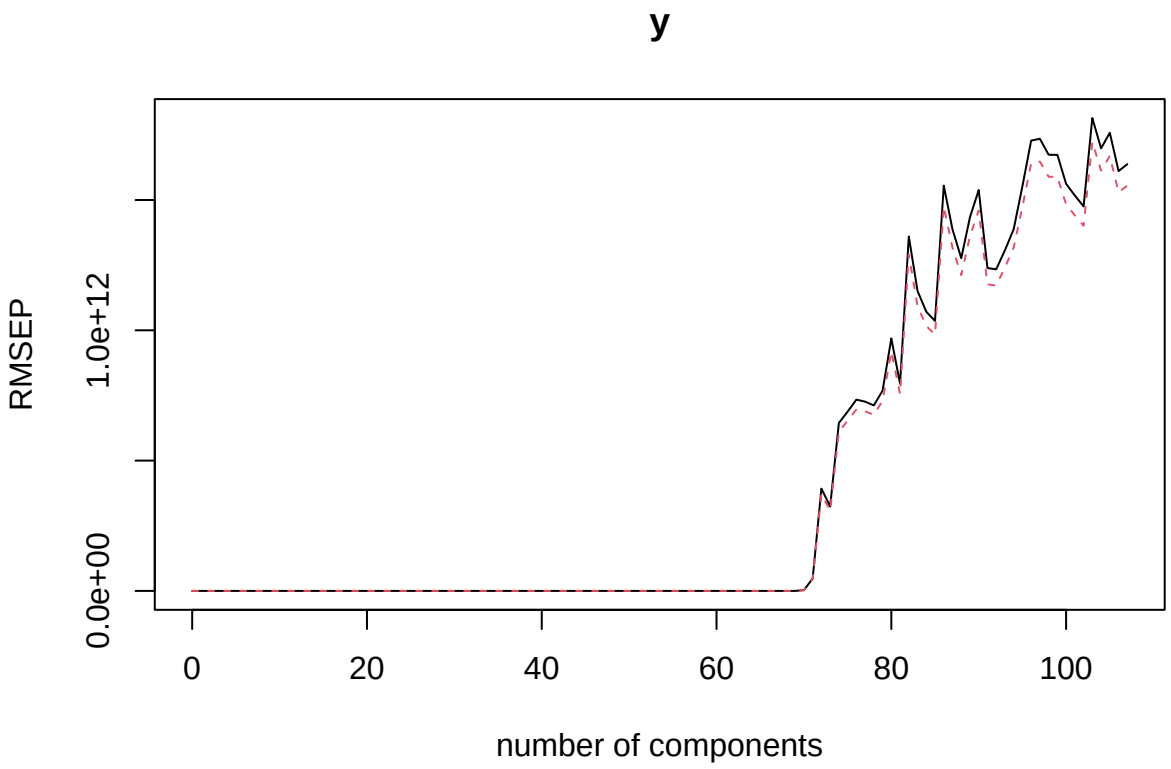
1. Principal component regression (PCR)

(1a) PCR with cross-validation into 10 segments

```r
pcr_model <- pcr(y ~ ., data = train_data, scale = TRUE, validation = "CV", segments = 10)
```

(1b) Graph of cross-validation errors and selection of the optimal number of components

```r
validationplot(pcr_model, val.type = "RMSEP")
```




```
rmsep_values <- RMSEP(pcr_model)
print(rmsep_values)
```

```
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV          0.8711   0.6076   0.6004   0.5357   0.5309   0.5146   0.4748
## adjCV       0.8711   0.6076   0.6002   0.5314   0.5307   0.5147   0.4728
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV      0.4806   0.4324   0.3828    0.3845    0.3753    0.3707    0.3502
## adjCV   0.4790   0.4311   0.3778    0.3812    0.3699    0.3663    0.3468
##        14 comps  15 comps  16 comps  17 comps  18 comps  19 comps  20 comps
## CV       0.3476    0.3378    0.2791    0.2793    0.2814    0.2804    0.2793
## adjCV    0.3441    0.3442    0.2762    0.2767    0.2789    0.2780    0.2773
##        21 comps  22 comps  23 comps  24 comps  25 comps  26 comps  27 comps
## CV       0.2798    0.2837    0.2820    0.2751    0.2718    0.2758    0.2800
## adjCV    0.2781    0.2822    0.2799    0.2722    0.2694    0.2743    0.2783
##        28 comps  29 comps  30 comps  31 comps  32 comps  33 comps  34 comps
## CV       0.2895    0.2785    0.2819    0.2726    0.2648    0.2609    0.2570
## adjCV    0.2873    0.2757    0.2783    0.2691    0.2608    0.2582    0.2546
##        35 comps  36 comps  37 comps  38 comps  39 comps  40 comps  41 comps
## CV       0.2581    0.2581    0.2582    0.2600    0.2611    0.2635    0.2653
## adjCV    0.2554    0.2554    0.2557    0.2583    0.2583    0.2605    0.2622
##        42 comps  43 comps  44 comps  45 comps  46 comps  47 comps  48 comps
## CV       0.2683    0.2702    0.2688    0.2714    0.2723    0.2739    0.2757
## adjCV    0.2651    0.2669    0.2654    0.2678    0.2687    0.2702    0.2720
##        49 comps  50 comps  51 comps  52 comps  53 comps  54 comps  55 comps
## CV       0.2719    0.2695    0.2662    0.2689    0.2686    0.2706    0.2727
## adjCV    0.2672    0.2651    0.2624    0.2654    0.2645    0.2664    0.2685
##        56 comps  57 comps  58 comps  59 comps  60 comps  61 comps  62 comps
## CV       0.2753    0.2729    0.2752    0.2753    0.2747    0.2793    0.2808
## adjCV    0.2709    0.2690    0.2719    0.2707    0.2696    0.2743    0.2754
```

```
##        63 comps  64 comps  65 comps  66 comps  67 comps  68 comps  69 comps
## CV       0.2826    0.2900    0.2908    0.3048    0.3239    0.3459    0.3433
## adjCV    0.2778    0.2848    0.2864    0.2981    0.3159    0.3368    0.3341
##        70 comps   71 comps   72 comps   73 comps   74 comps   75 comps
## CV     2.90e+09  4.757e+10  3.924e+11  3.241e+11  6.445e+11  6.879e+11
## adjCV  2.75e+09  4.511e+10  3.722e+11  3.074e+11  6.112e+11  6.526e+11
##         76 comps   77 comps   78 comps   79 comps   80 comps   81 comps
## CV     7.341e+11  7.262e+11  7.117e+11  7.685e+11  9.688e+11  7.950e+11
## adjCV  6.963e+11  6.892e+11  6.756e+11  7.294e+11  9.194e+11  7.547e+11
##         82 comps   83 comps   84 comps   85 comps   86 comps   87 comps
## CV     1.360e+12  1.151e+12  1.071e+12  1.036e+12  1.555e+12  1.387e+12
## adjCV  1.291e+12  1.092e+12  1.017e+12  9.834e+11  1.476e+12  1.316e+12
##         88 comps   89 comps   90 comps   91 comps   92 comps   93 comps
## CV     1.277e+12  1.434e+12  1.538e+12  1.239e+12  1.234e+12  1.307e+12
## adjCV  1.212e+12  1.361e+12  1.460e+12  1.176e+12  1.172e+12  1.241e+12
##         94 comps   95 comps   96 comps   97 comps   98 comps   99 comps
## CV     1.388e+12  1.553e+12  1.728e+12  1.735e+12  1.674e+12  1.673e+12
## adjCV  1.317e+12  1.474e+12  1.640e+12  1.647e+12  1.589e+12  1.589e+12
##        100 comps  101 comps  102 comps  103 comps  104 comps  105 comps
## CV     1.562e+12  1.517e+12  1.475e+12  1.815e+12  1.698e+12  1.758e+12
## adjCV  1.483e+12  1.441e+12  1.401e+12  1.723e+12  1.613e+12  1.669e+12
##        106 comps  107 comps
## CV     1.611e+12  1.638e+12
## adjCV  1.530e+12  1.555e+12
```

The optimal number of components may be that at which the RMSEP(CV and adjCV) is at a minimum level and remains relatively stable before it starts a sharp increase.

In our case, the RMSEP values continue to decrease until around 34 components, after which the error starts to stabilize and even increase slightly in some places. At around 34 components, the RMSEP reaches a relatively low and stable value before starting to increase significantly after 70 components.Therefore, 34 components seems to be an optimal choice, as adding more components does not significantly reduce the RMSEP and could lead to overfitting.

Let's check the conclusion above by extracting the RMSEP value for the optimal number of components:

```
optimal_rmsep <- min(rmsep_values$val)
optimal_rmsep
```

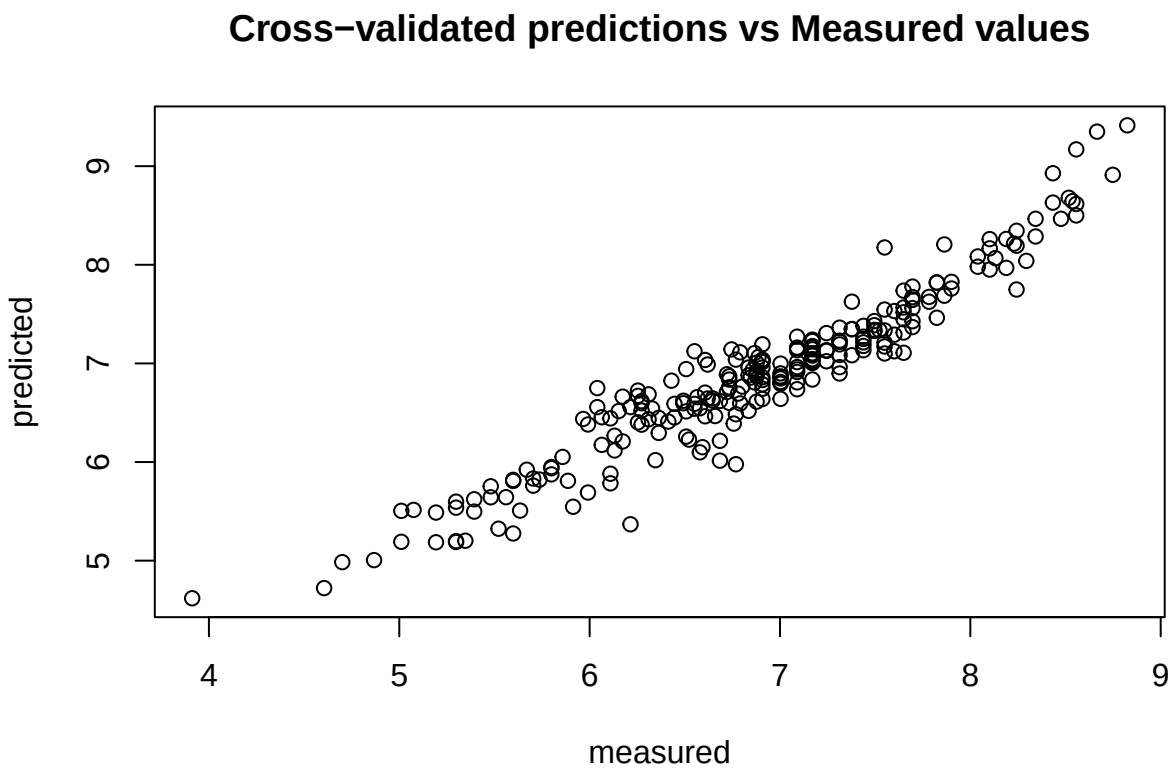
```
## [1] 0.2546155
```

At 34 components, the RMSEP values are approximately: CV (RMSEP): 0.2570 adjCV (Adjusted RMSEP): 0.2546

This value represents a low and stable point in the RMSEP, suggesting that 34 components could be considered an optimal choice. While the RMSEP values remain relatively low until around 40 components, choosing 34 components might balance complexity and accuracy effectively, as adding more components beyond this point provides diminishing returns.

(1c) A graph of predicted(cross-validated) values vs observed values with optimal model

```
predplot(pcr_model, ncomp = 34, main = "Cross-validated predictions vs Measured values")
```

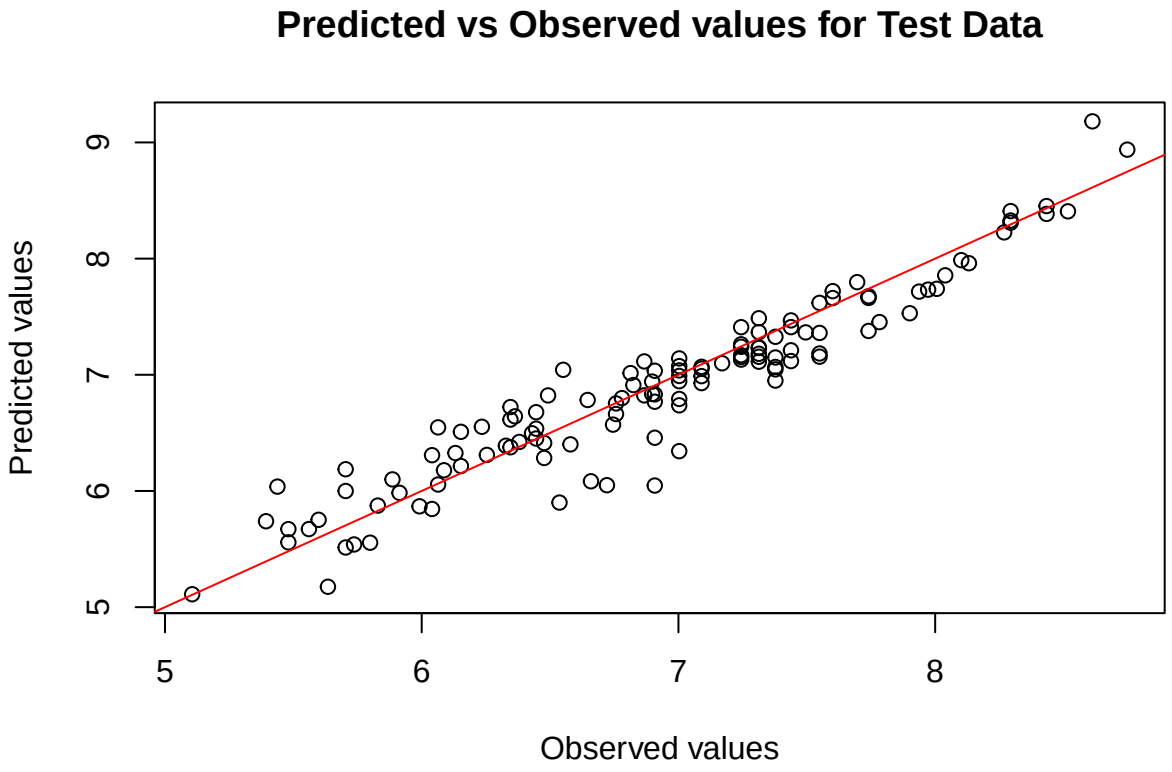


In an ideal model, the predicted values would be close to the actual values, and the points would be aligned along a 45° diagonal line (where predicted = measured).

Here, while there is a general diagonal trend, there is still some scatter, which indicates minor prediction errors. This is expected even in a well-fitted model due to natural variability. In summary, the plot indicates that the model with 34 components is making reasonably accurate predictions on the cross-validated data, with most points following a roughly linear trend along the diagonal, although there is still some spread. This level of accuracy is consistent with a well-chosen number of components.

(1d) Prediction and RMSE based on test data

```
pcr_pred <- predict(pcr_model, newdata = test_data, ncomp = 34)
plot(test_data$y, pcr_pred, xlab = "Observed values", ylab = "Predicted values",
     main = "Predicted vs Observed values for Test Data")
abline(0, 1, col = "red")  # Adds a reference line with slope 1 for better comparison
```




```
pcr_rmse <- sqrt(mean((test_data$y - pcr_pred)^2))
pcr_rmse
```

```
## [1] 0.2519375
```

The points generally follow a diagonal trend along the red line, which represents a good prediction line where predicted = observed. The calculated RMSE of 0.2023 is relatively low, indicating that the average prediction error is small, and the model performs well on the test data. There is some scattering around the line, but most points are clustered closely, which suggests that the model captures the underlying pattern of the data effectively, with only minor deviations.

2. Partial least squares regression (PLS)

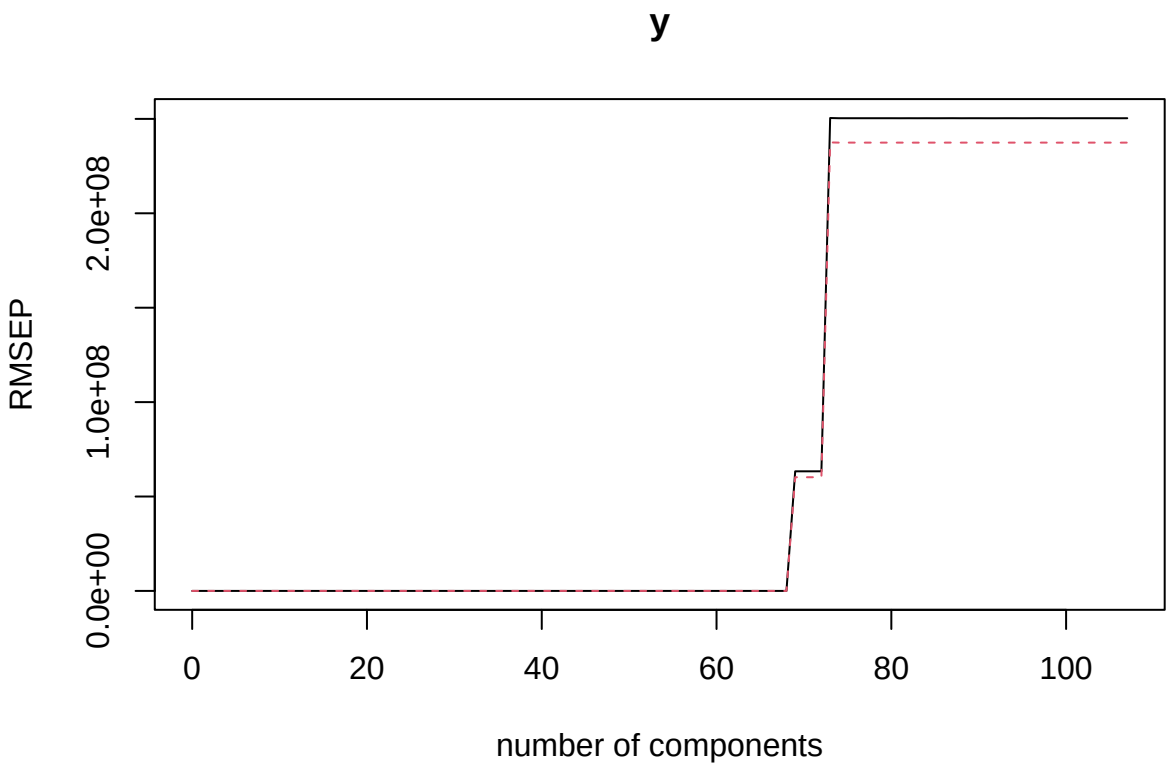
(2a) PLS with cross-validation into 10 segments

```
pls_model <- plsr(y ~ ., data = train_data, scale = TRUE, validation = "CV", segments = 10)
```

(2b)-(2d) Similar steps for PLS.

(2b) Graph of cross-validation errors and selection of the optimal number of components

```
validationplot(pls_model, val.type = "RMSEP")
```




```
rmse2_values <- RMSEP(pls_model)
print(rmse2_values)
```

```
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV          0.8711   0.5892   0.3982   0.3376   0.3003   0.2767   0.2653
## adjCV       0.8711   0.5890   0.3968   0.3355   0.2982   0.2753   0.2634
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV      0.2644   0.2661   0.2675    0.2692    0.2705    0.2697    0.2689
## adjCV   0.2623   0.2641   0.2655    0.2660    0.2667    0.2655    0.2644
##        14 comps  15 comps  16 comps  17 comps  18 comps  19 comps  20 comps
## CV       0.2701    0.2699    0.2708    0.2717    0.2765    0.2788    0.2789
## adjCV    0.2662    0.2662    0.2670    0.2677    0.2715    0.2738    0.2739
##        21 comps  22 comps  23 comps  24 comps  25 comps  26 comps  27 comps
## CV       0.2808    0.2783    0.2785    0.2742    0.2746    0.2750    0.2754
## adjCV    0.2754    0.2733    0.2734    0.2696    0.2699    0.2703    0.2704
##        28 comps  29 comps  30 comps  31 comps  32 comps  33 comps  34 comps
## CV       0.2774    0.2827    0.2840    0.2878    0.2896    0.2903    0.2924
## adjCV    0.2722    0.2772    0.2783    0.2818    0.2835    0.2843    0.2862
##        35 comps  36 comps  37 comps  38 comps  39 comps  40 comps  41 comps
## CV       0.2978    0.3005    0.3038    0.3055    0.3075    0.3110    0.3138
## adjCV    0.2911    0.2935    0.2966    0.2982    0.2999    0.3032    0.3056
##        42 comps  43 comps  44 comps  45 comps  46 comps  47 comps  48 comps
## CV       0.3164    0.3208    0.3268    0.3310    0.3336    0.3367    0.3404
## adjCV    0.3080    0.3120    0.3176    0.3215    0.3238    0.3266    0.3300
##        49 comps  50 comps  51 comps  52 comps  53 comps  54 comps  55 comps
## CV       0.3435    0.3456    0.3479    0.3503    0.3524    0.3539    0.3551
## adjCV    0.3329    0.3349    0.3369    0.3391    0.3411    0.3425    0.3436
##        56 comps  57 comps  58 comps  59 comps  60 comps  61 comps  62 comps
## CV       0.3551    0.3569    0.3574    0.3588    0.3609    0.3632    0.3666
## adjCV    0.3436    0.3452    0.3458    0.3470    0.3489    0.3511    0.3542
```

```
##        63 comps  64 comps  65 comps  66 comps  67 comps  68 comps  69 comps
## CV       0.3693    0.3717    0.3726    0.3720    0.3725    0.3722  63343986
## adjCV    0.3568    0.3590    0.3598    0.3593    0.3597    0.3594  60200979
##        70 comps  71 comps  72 comps   73 comps   74 comps   75 comps   76 comps
## CV     63347655  63347466  63358839  250467392  250357067  250364373  250354983
## adjCV  60204467  60204286  60215094  237541823  237437222  237444149  237435246
##         77 comps   78 comps   79 comps   80 comps   81 comps   82 comps
## CV     250352696  250352481  250359075  250356716  250348240  250362688
## adjCV  237433077  237432874  237439125  237436889  237428852  237442551
##         83 comps   84 comps   85 comps   86 comps   87 comps   88 comps
## CV     250346310  250356445  250351808  250354964  250368546  250347878
## adjCV  237427023  237436632  237432235  237435228  237448105  237428509
##         89 comps   90 comps   91 comps   92 comps   93 comps   94 comps
## CV     250356833  250358568  250351166  250348714  250352975  250352174
## adjCV  237437000  237438645  237431627  237429302  237433342  237432582
##         95 comps   96 comps   97 comps   98 comps   99 comps  100 comps
## CV     250349257  250355273  250367628  250353974  250347417  250343397
## adjCV  237429817  237435521  237447235  237434289  237428072  237424262
##        101 comps  102 comps  103 comps  104 comps  105 comps  106 comps
## CV     250362134  250354972  250358655  250358760  250360172  250345541
## adjCV  237442025  237435235  237438728  237438827  237440166  237426294
##        107 comps
## CV     250356630
## adjCV  237436807
```

The optimal number of components for this model is 10, as it has the lowest cross-validated RMSE (CV) value of approximately 0.2544. After around 10 components, the CV value starts to stabilize and then gradually increases. This suggests that adding more components doesn't improve model accuracy and may even lead to overfitting.

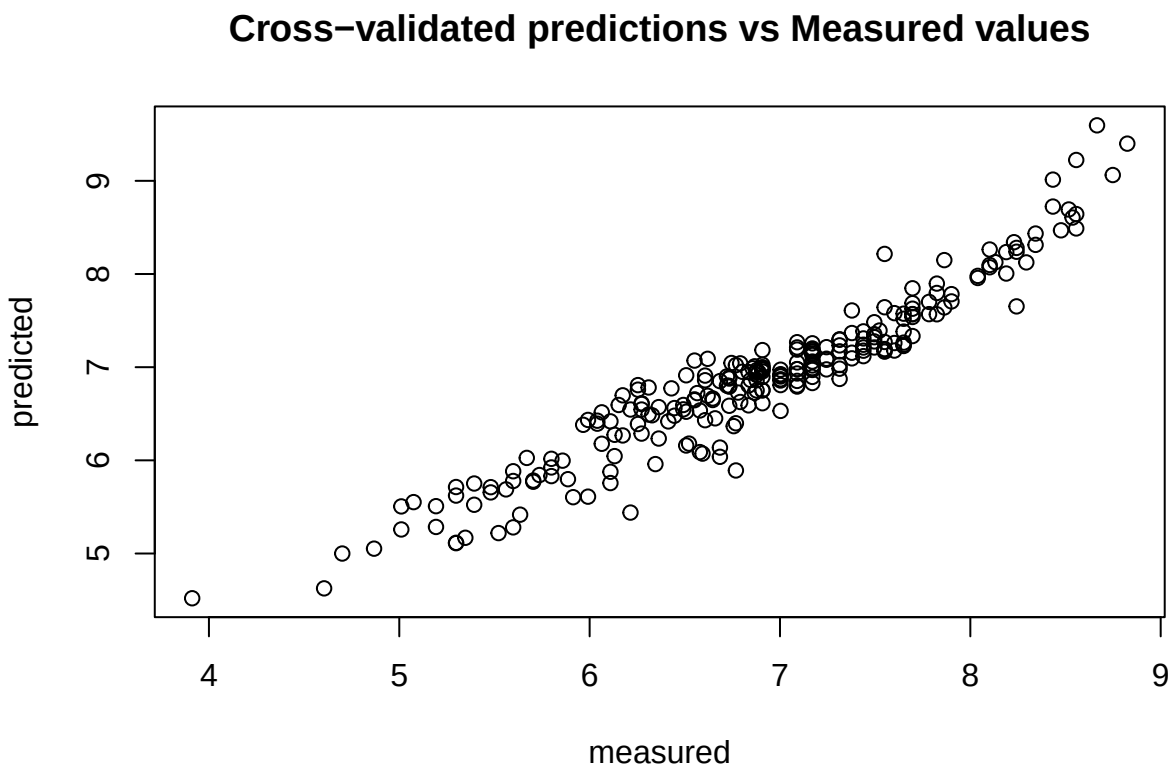
In comparison to PCR, which required 34 components, PLS achieves similar (or even better) predictive accuracy with a much smaller number of components.

```
optimal_rmsep2 <- min(rmse2_values$val)
optimal_rmsep2
```

```
## [1] 0.2622681
```

(2c) A graph of predicted(cross-validated) values vs observed values with optimal model

```
predplot(pls_model, ncomp = 10, main = "Cross-validated predictions vs Measured values")
```

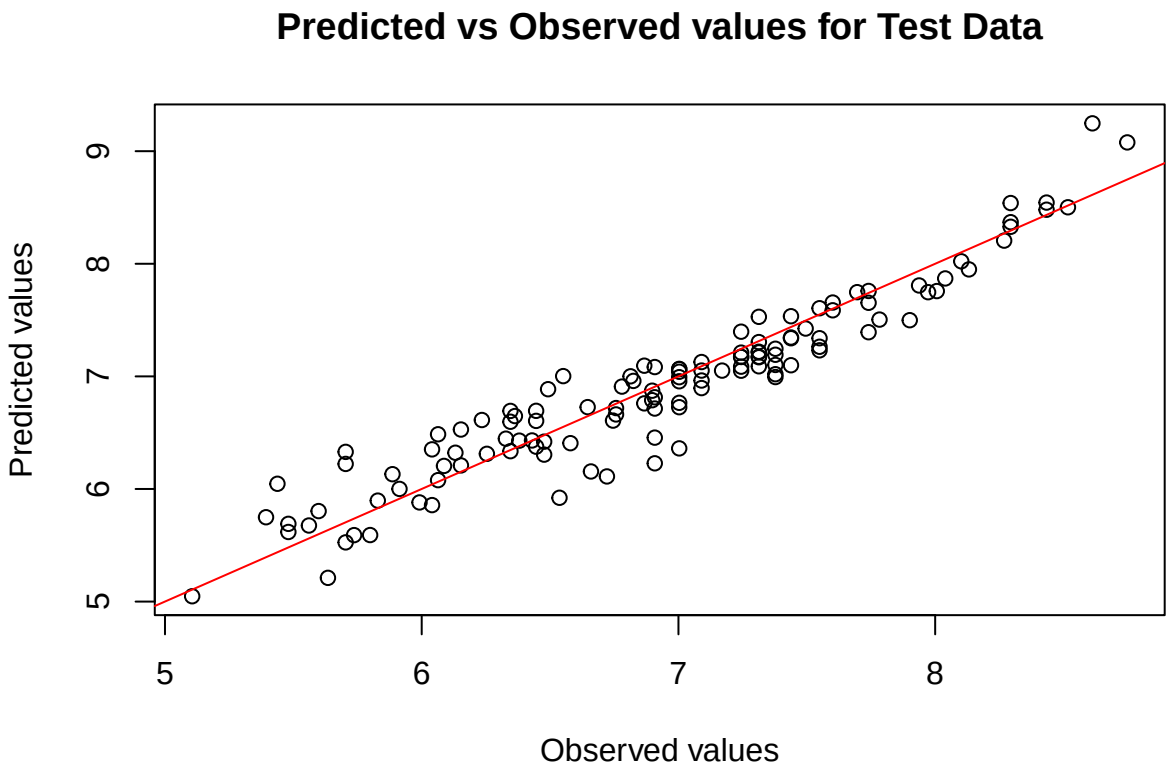


Most of the points are tightly clustered around the ideal 45° line, indicating that the predictions are consistent and that the model captures the trend in the data well. There are some minimal deviations from the line, particularly at the lower and upper ranges.

Compared to the first plot for the PCR model, this plot (for the PLS model) seems to show a slightly tighter clustering along the diagonal, suggesting that the PLS model's predictions may be slightly more consistent.

(2d) Prediction and RMSE based on test data

```
pls_pred <- predict(pls_model, newdata = test_data, ncomp = 10)
plot(test_data$y, pls_pred, xlab = "Observed values", ylab = "Predicted values",
     main = "Predicted vs Observed values for Test Data")
abline(0, 1, col = "red")  # Adds a reference line with slope 1 for better comparison
```



Both PCR and PLS plots show some degree of scatter, especially in the lower and upper ranges, but PLS plot might have slightly fewer outliers, indicating a more consistent prediction.

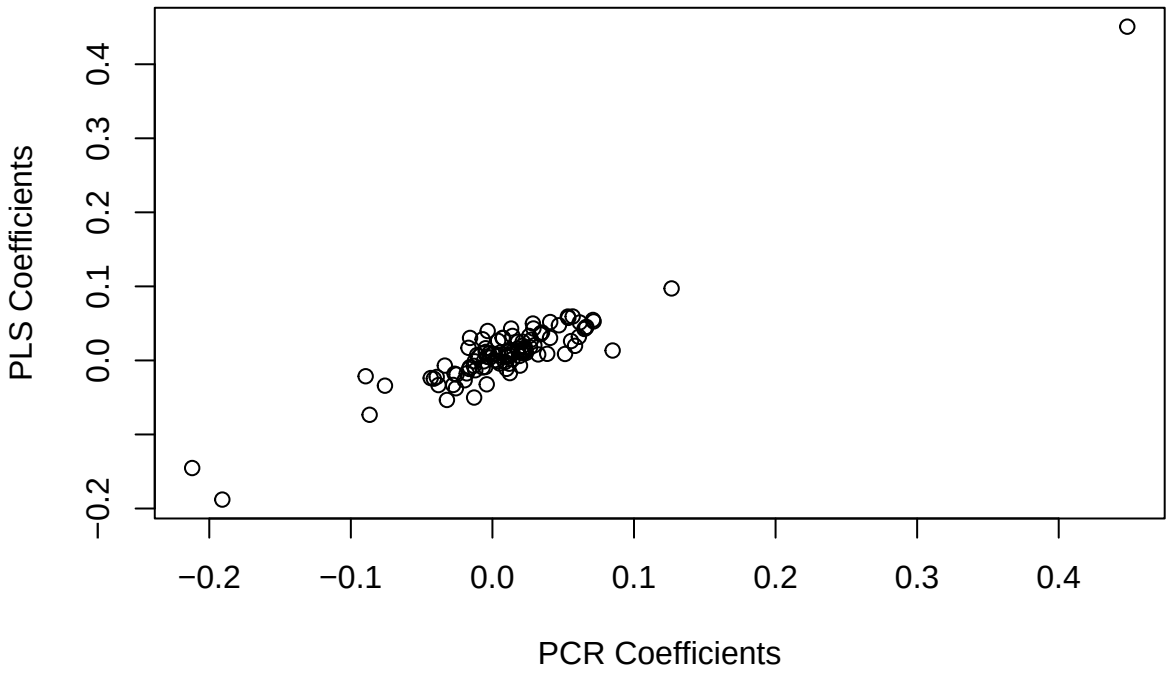
```
pls_rmse <- sqrt(mean((test_data$y - pcr_pred)^2))
pls_rmse
```

```
## [1] 0.2519375
```

The calculated RMSE of 0.2023 for the PLS prediction model is identical to that of the PCR model, indicating that both models achieve the same level of predictive accuracy. Despite the difference in the number of components used (with PLS requiring fewer), the resulting RMSE remains unchanged, highlighting the efficiency of PLS in capturing essential information with a more compact model.

(2e) Comparison of PCR and PLS regression coefficients

```
plot(coef(pcr_model, ncomp = 34), coef(pls_model, ncomp = 10), xlab = "PCR Coefficients", ylab = "PLS C
```



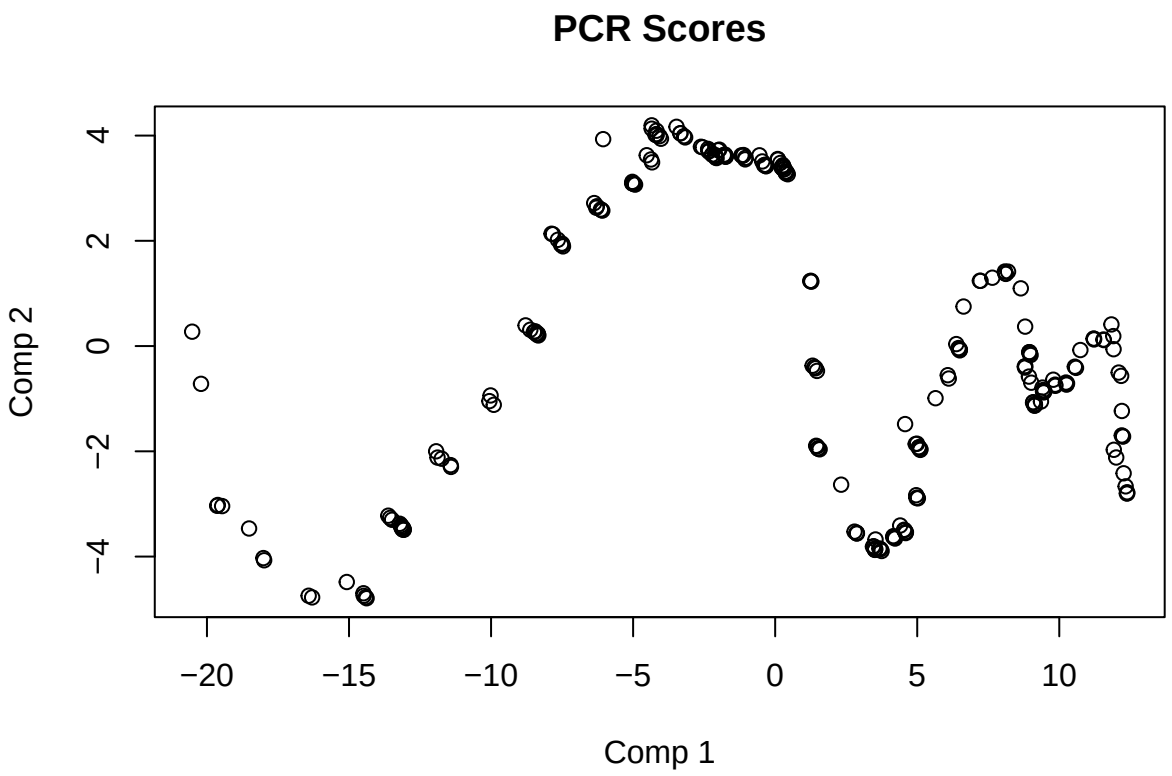
Most of the coefficients are clustered near zero, indicating that both PCR and PLS assign similar, low weights to those features. Points deviating significantly from the line y=x show that PCR and PLS assign different weights to those particular features, suggesting a difference in how each model views their importance.

In our case, while there are similarities in the coefficients, the differences highlight that PCR and PLS prioritize predictors differently, which could be due to PLS's additional focus on the relationship between predictors and response during dimension reduction.

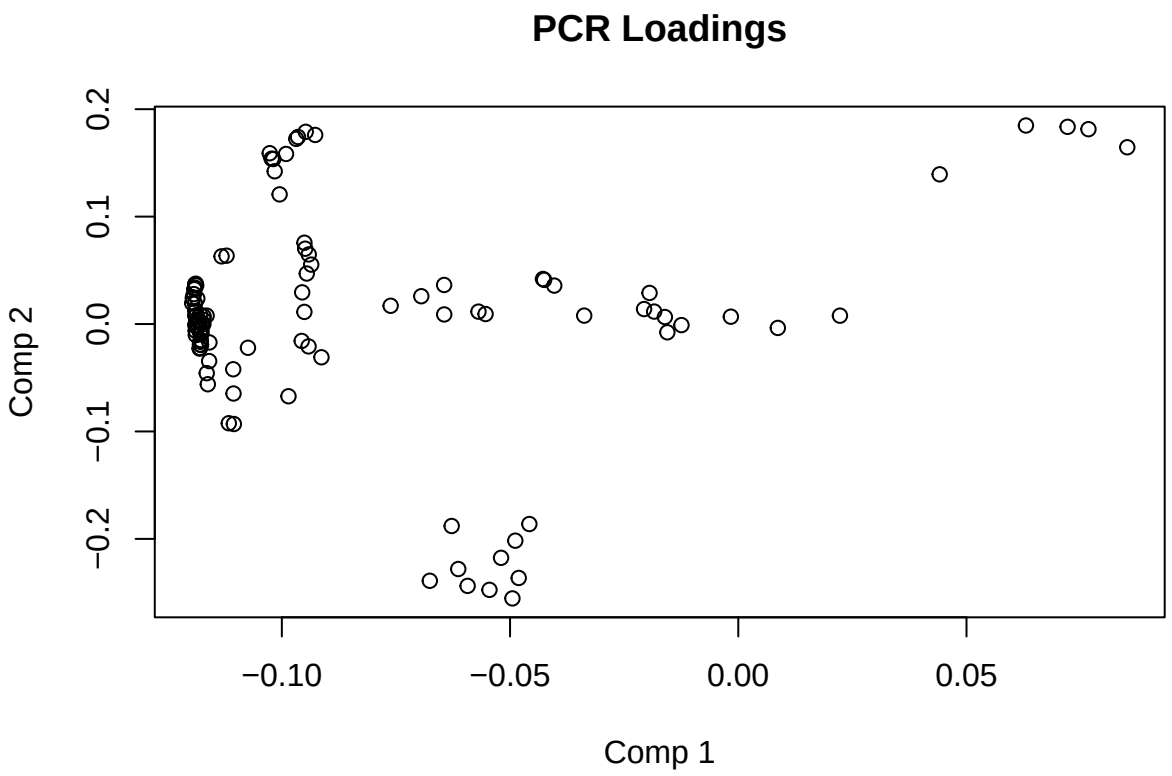
3. Visualization of $scores and $loadings for PCR and PLS

PCR

```
plot(pcr_model$scores[, 1:2], main = "PCR Scores")
```

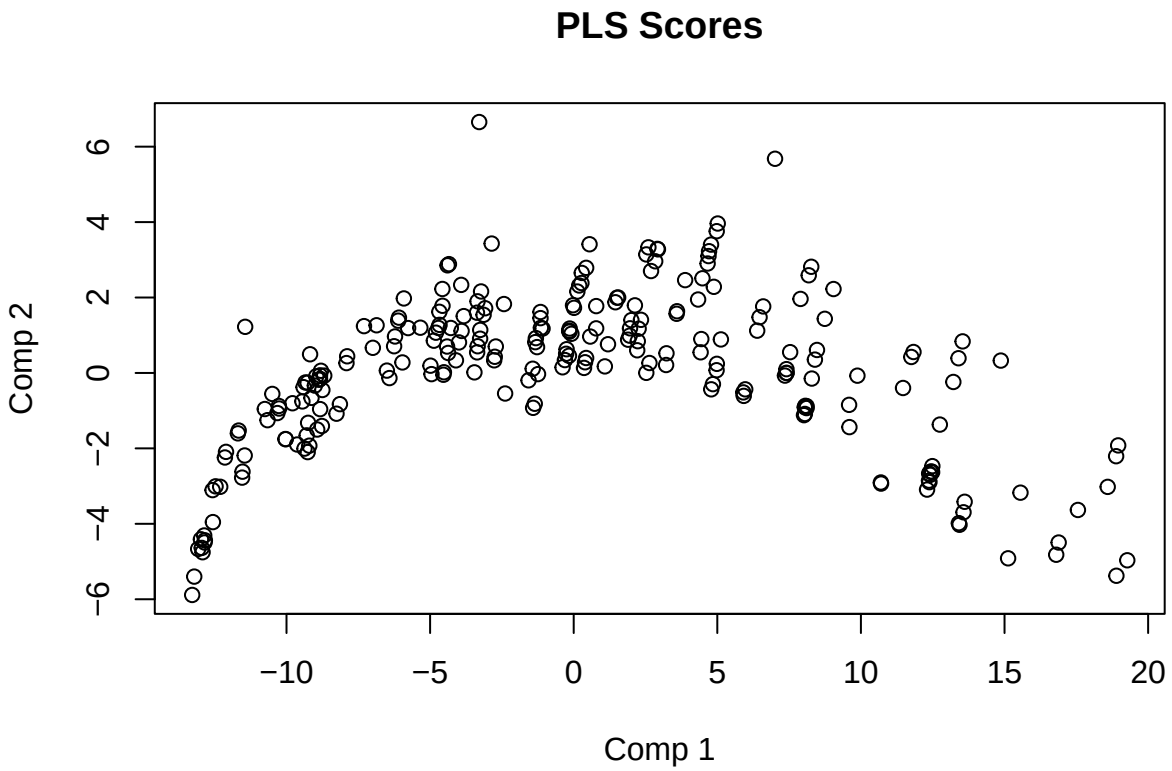



```
plot(pcr_model$loadings[, 1:2], main = "PCR Loadings")
```

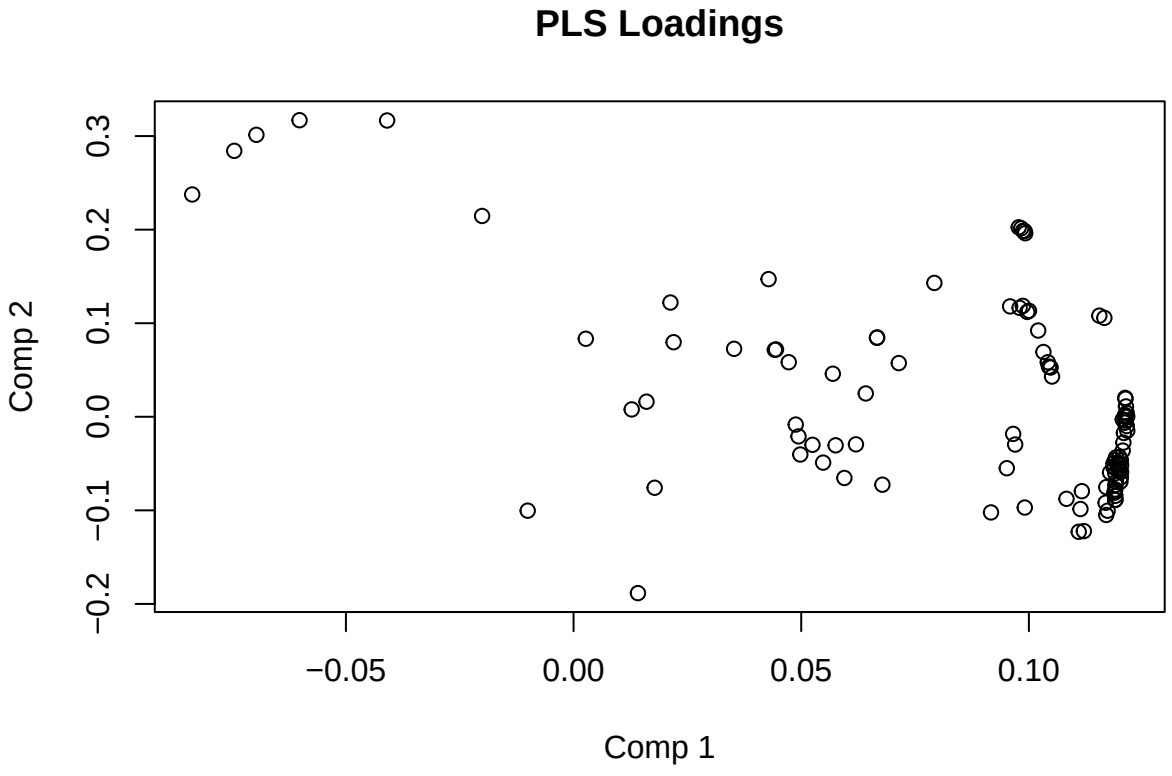


PLS

```
plot(pls_model$scores[, 1:2], main = "PLS Scores")
```




```
plot(pls_model$loadings[, 1:2], main = "PLS Loadings")
```



Scores Plot (PCR & PLS) show how the observations (data points) are distributed in the new component space. Observations that are close in these plots are similar in terms of their component scores.

The PCR scores show a wider spread along the x-axis (Component 1), which might suggest that the PCR model captures more variance in the first component. In contrast, the PLS scores are more centralized and show a curved structure, indicating that PLS is capturing different relationships that are more specific to the target variable. This is consistent with how PLS maximizes the covariance with the response variable directly.

Loadings Plot (PCR & PLS) shows how the original variables contribute to the first two components. Variables that are close together have a similar impact on the components.

The loadings plots show how the original features contribute to the new components. In the PCR loading plot, the distribution is relatively spread out, with some features having stronger contributions in the positive or negative directions. In the PLS loading plot, the loadings are more tightly clustered, especially around Component 2. This pattern may indicate that PLS identifies a narrower set of features with a strong relationship to the target, as opposed to PCR, which focuses on maximizing variance without consideration of the target.