Exercise 2

Tobias Raidl, 11717659

2023-11-02

Setup data

```
load("building.RData")
set.seed(11717659)
sample = sample(c(TRUE, FALSE), nrow(df), replace=TRUE, prob=c(2/3,1/3))
train = df[sample, ]
test = df[!sample, ]
```

1

(a)

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

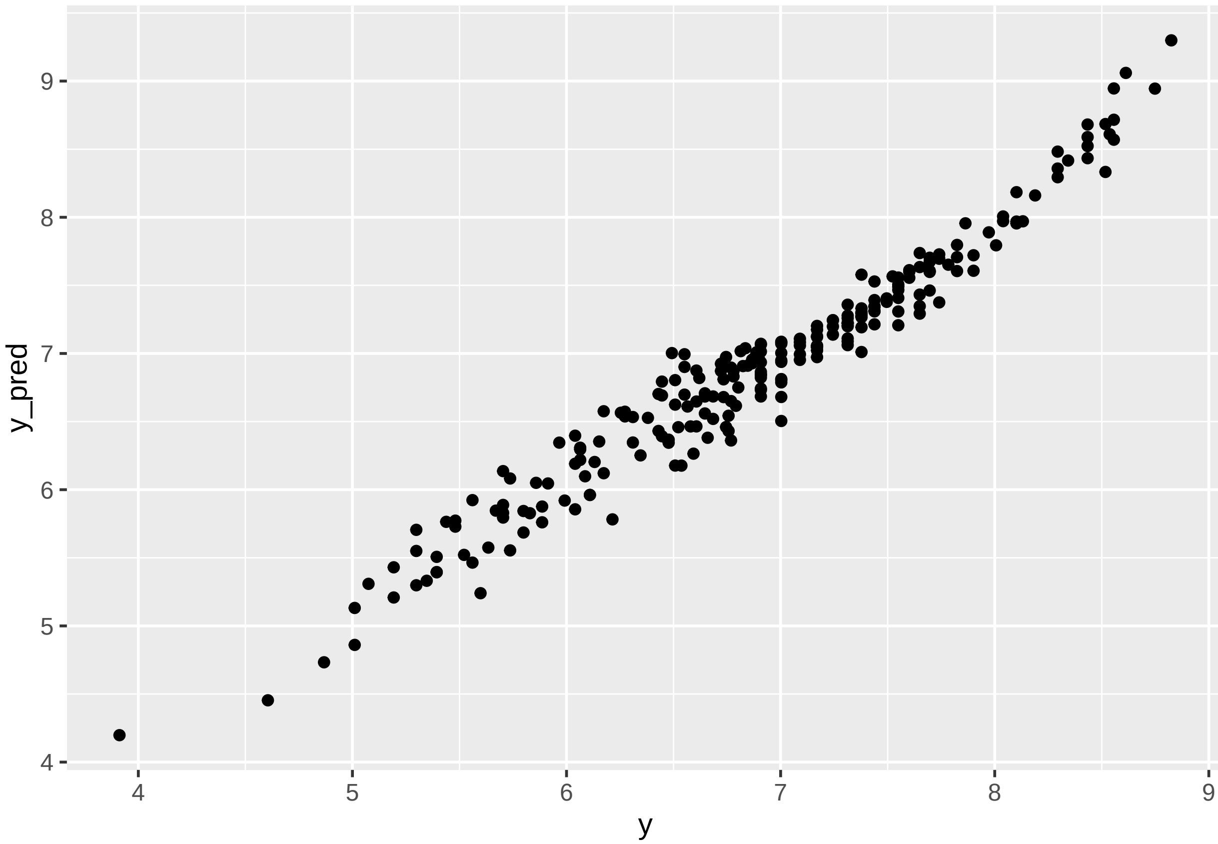
```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
model = lm(y~., train)
train.y_pred = predict(model, select(train, -y))
```

```
## Warning in predict.lm(model, select(train, -y)): prediction from a
## rank-deficient fit may be misleading
```

```
res.train = data.frame(y=train$y, y_pred=train.y_pred)
ggplot(res.train, aes(x=y, y=y_pred)) +
  geom_point()
```




```
get_rmse = function(y, y_pred) {
  residuals = (y-y_pred)^2
  return(sqrt(sum(residuals)/length(residuals)))
}

cat(paste("RMSE for train set:", get_rmse(res.train$y, res.train$y_pred)))
```

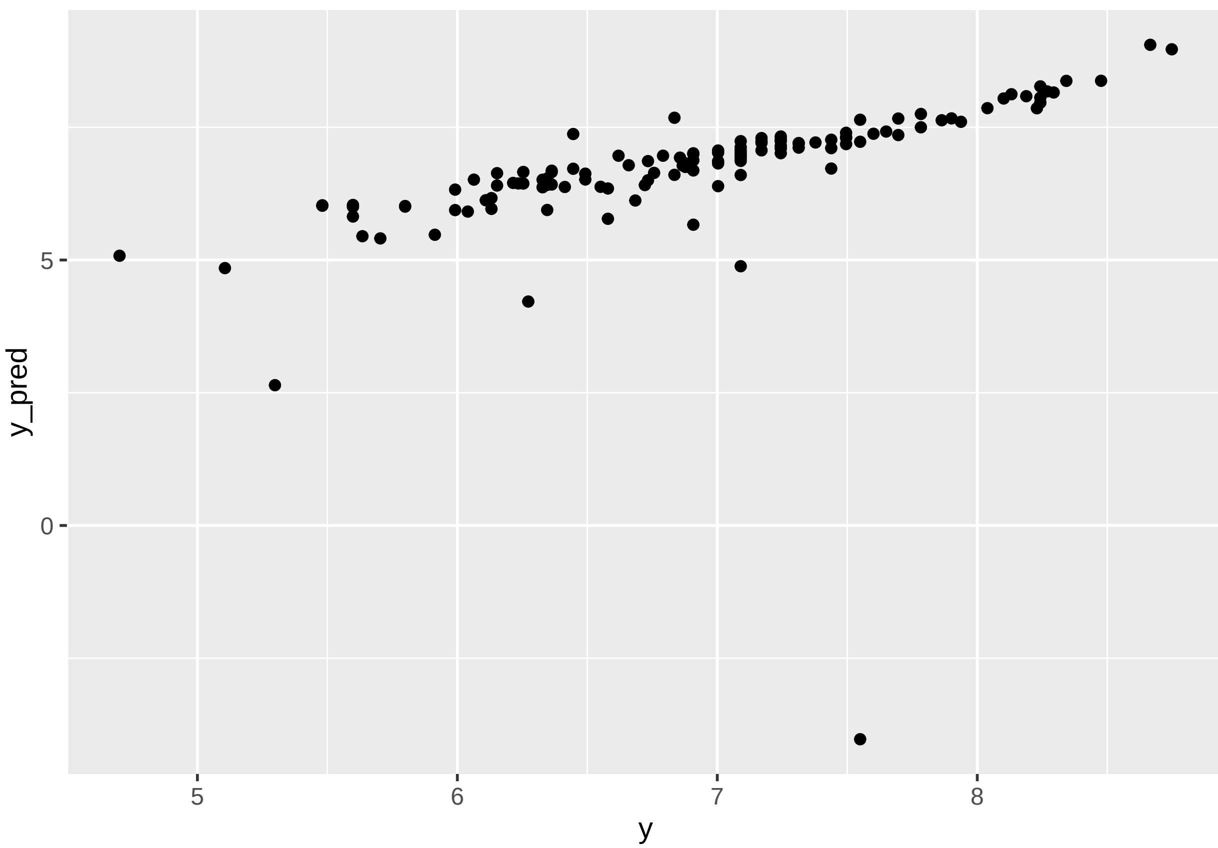
```
## RMSE for train set: 0.187445523226305
```

(b)

```
test.y_pred = predict(model, select(test, -y))
```

```
## Warning in predict.lm(model, select(test, -y)): prediction from a
## rank-deficient fit may be misleading
```

```
res.test = data.frame(y=test$y, y_pred=test.y_pred)
ggplot(res.test, aes(x=y, y=y_pred)) +
  geom_point()
```



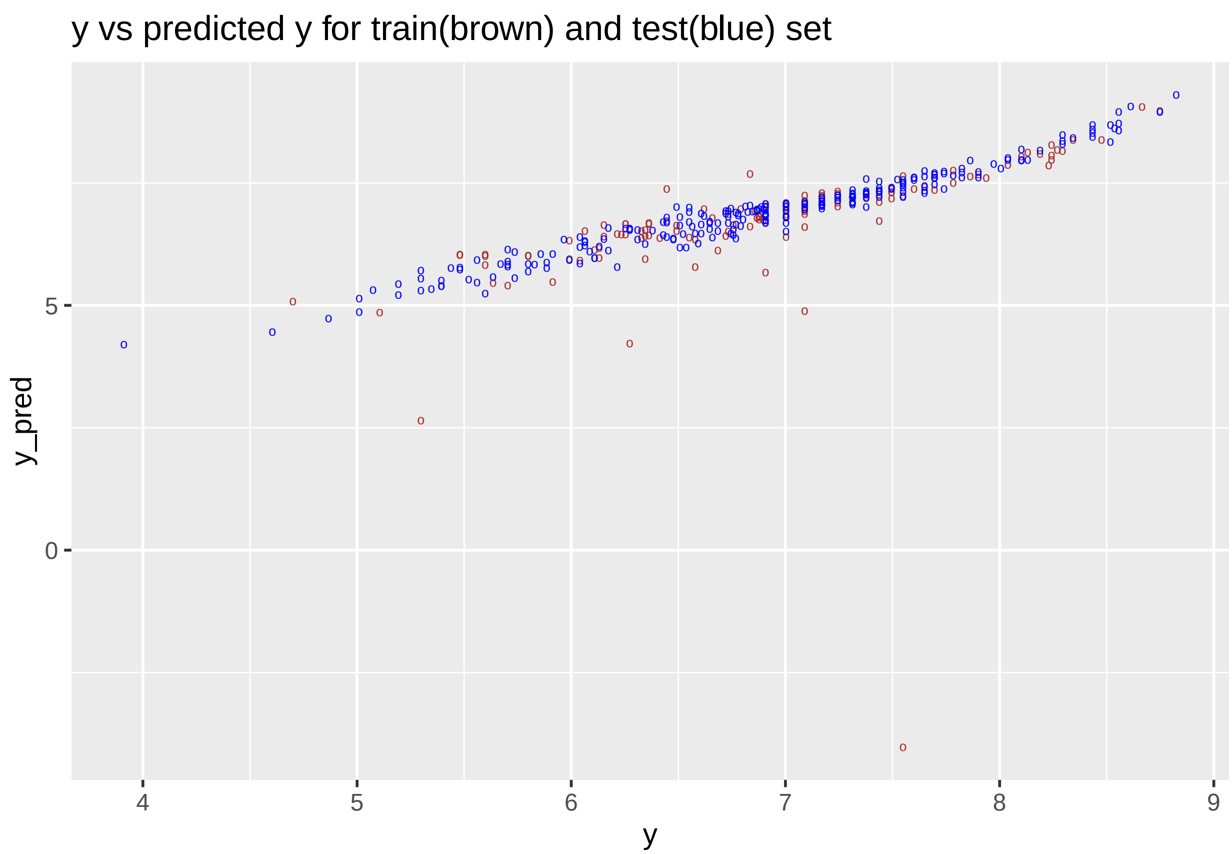

```r
get_rmse = function(y, y_pred) {
  residuals = (y-y_pred)^2
  return(sqrt(sum(residuals)/length(residuals)))
}

cat(paste("RMSE for test set:", get_rmse(res.test$y, res.test$y_pred)))
```

```
## RMSE for test set: 1.12629535886144
```

(c)

The RMSE is higher for the test set than the train set. This is due to the model being fitted on the train set. A high error difference between these to sets indicates an overfitted model.

```r
ggplot() +
  geom_point(data=res.test, mapping=aes(x=y, y=y_pred), color="brown", pch="o") +
  geom_point(data=res.train, mapping=aes(x=y, y=y_pred), color="blue", pch="o") +
  ggtitle("y vs predicted y for train(brown) and test(blue) set")
```



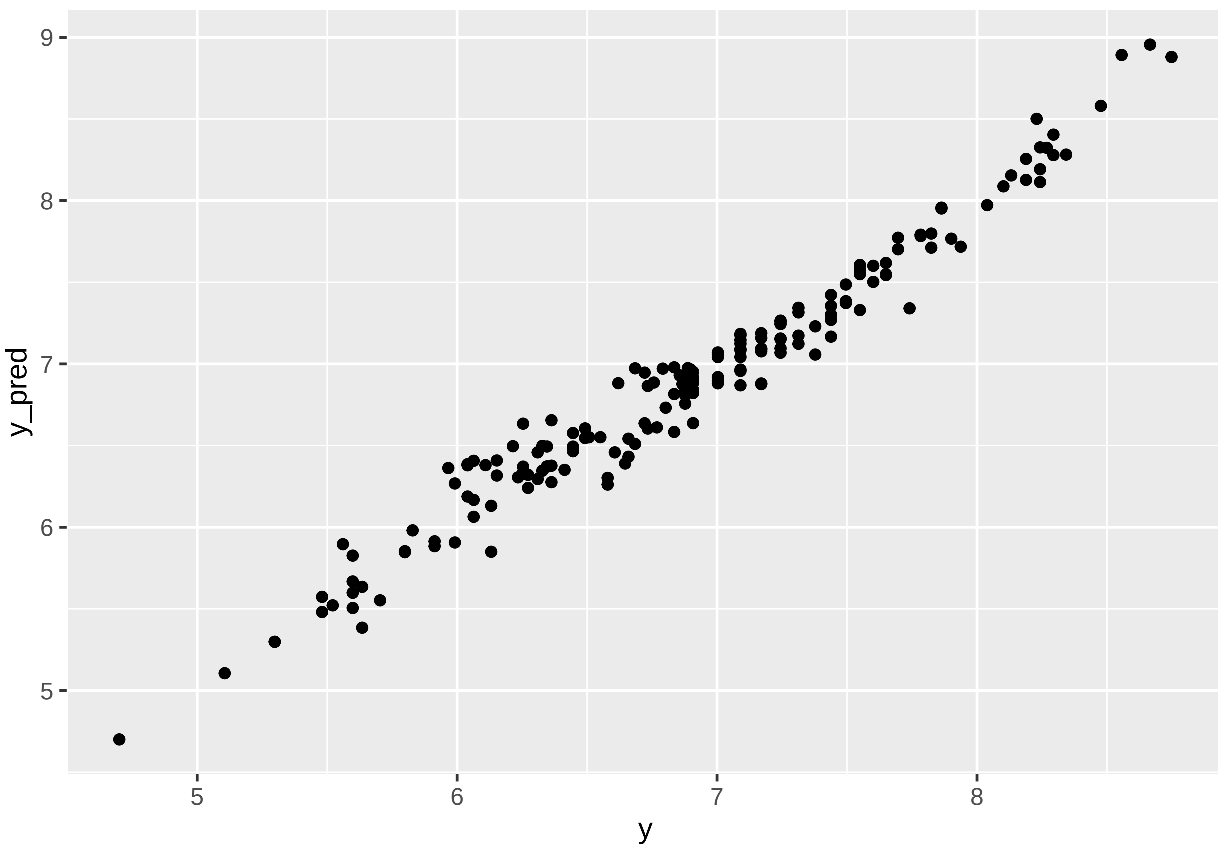
(d)

I chose a 50/50 split and the RMSE for the train set is even lower, but the rmse for the test split higher than with the previous split. More training data being available translates into a more overfitted model. But also more test data being available means that the evaluation is probably more accurate. than before.

```
set.seed(11717659)
sample2 = sample(c(TRUE, FALSE), nrow(df), replace=TRUE, prob=c(0.5, 0.5))
train2 = df[sample2, ]
test2 = df[!sample2, ]

model = lm(y~., train2)
train2.y_pred = predict(model, select(train2, -y))
```

```
## Warning in predict.lm(model, select(train2, -y)): prediction from a
## rank-deficient fit may be misleading
```

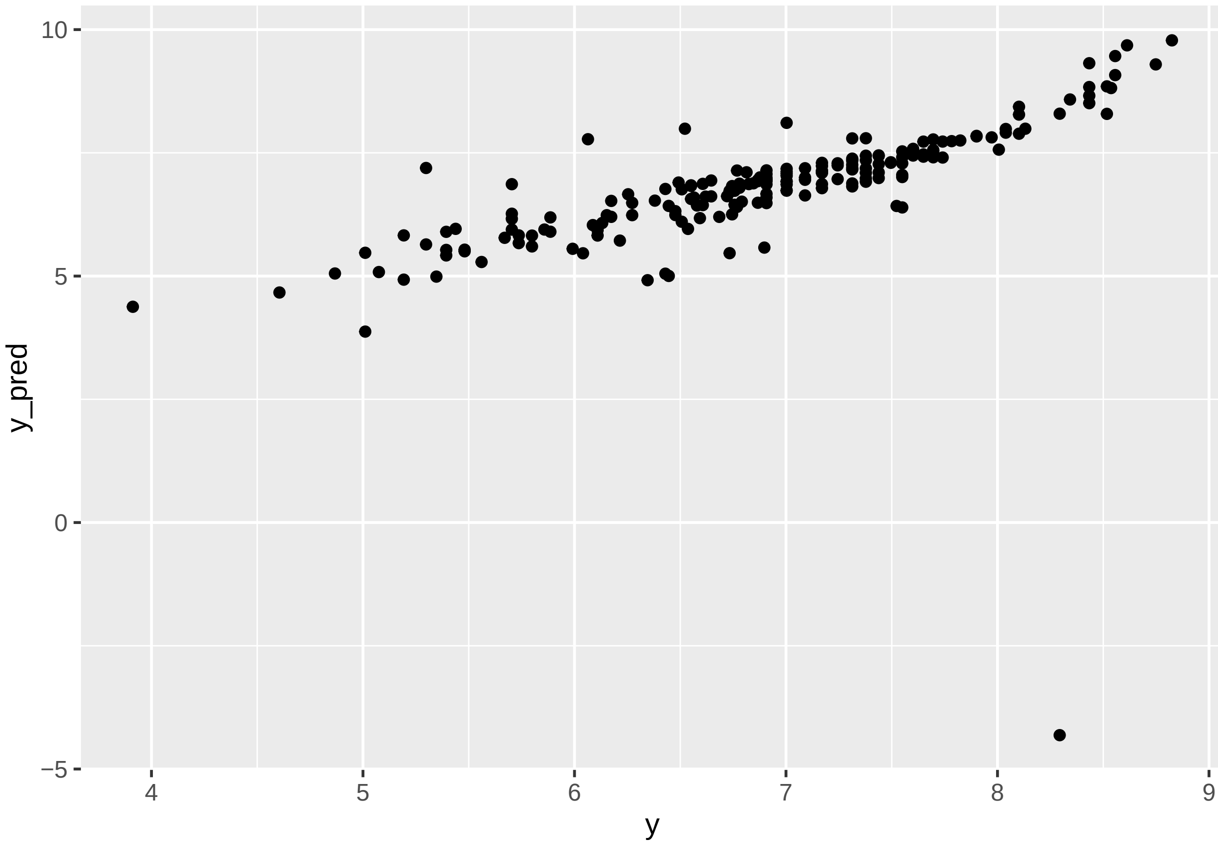
```
res.train2 = data.frame(y=train2$y, y_pred=train2.y_pred)
ggplot(res.train2, aes(x=y, y=y_pred)) +
  geom_point()
```




```r
get_rmse = function(y, y_pred) {
  residuals = (y-y_pred)^2
  return(sqrt(sum(residuals)/length(residuals)))
}

test2.y_pred = predict(model, select(test2, -y))
```

```
## Warning in predict.lm(model, select(test2, -y)): prediction from a
## rank-deficient fit may be misleading
```

```r
res.test2 = data.frame(y=test2$y, y_pred=test2.y_pred)
ggplot(res.test2, aes(x=y, y=y_pred)) +
  geom_point()
```




```
get_rmse = function(y, y_pred) {
  residuals = (y-y_pred)^2
  return(sqrt(sum(residuals)/length(residuals)))
}

cat(paste("RMSE for train set:", get_rmse(res.train2$y, res.train2$y_pred)))
```

```
## RMSE for train set: 0.151012262080028
```

```
cat("\n")
```

```
cat(paste("RMSE for test set:", get_rmse(res.test2$y, res.test2$y_pred)))
```

```
## RMSE for test set: 1.00575049742675
```

2

Nothin to do here

3

(a)

```
library(pls)
```

```
##
## Attaching package: 'pls'

## The following object is masked from 'package:stats':
##
##     loadings
```

```
set.seed(11717659)
# train_idxs = which(sample)
pcr_fit = pcr(y~., data=train, scale=TRUE, validation="CV", segments=10,
↪  segment.type="random")
summary(pcr_fit)
```

```
## Data:    X dimension: 245 107
##  Y dimension: 245 1
## Fit method: svdpc
## Number of components considered: 107
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV          0.8748   0.6086   0.5970   0.5617   0.5172   0.4939   0.4464
## adjCV       0.8748   0.6084   0.5966   0.5613   0.5163   0.4930   0.4442
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV       0.440   0.4418   0.3982    0.3739    0.3679    0.3731    0.3544
## adjCV    0.438   0.4407   0.3902    0.3702    0.3652    0.3714    0.3530
##        14 comps  15 comps  16 comps  17 comps  18 comps  19 comps  20 comps
## CV       0.3366    0.3281    0.2945    0.2953    0.2899    0.2914    0.2921
## adjCV    0.3342    0.3275    0.2919    0.2927    0.2887    0.2902    0.2913
##        21 comps  22 comps  23 comps  24 comps  25 comps  26 comps  27 comps
## CV       0.2921    0.2895    0.2916    0.2906    0.2864    0.2889    0.2904
## adjCV    0.2917    0.2875    0.2895    0.2888    0.2844    0.2869    0.2884
##        28 comps  29 comps  30 comps  31 comps  32 comps  33 comps  34 comps
## CV       0.2927    0.2870    0.2795    0.2671    0.2676    0.2671    0.2664
## adjCV    0.2913    0.2852    0.2765    0.2650    0.2657    0.2654    0.2639
##        35 comps  36 comps  37 comps  38 comps  39 comps  40 comps  41 comps
## CV       0.2697    0.2699    0.2629    0.2642    0.2642    0.2677    0.2704
## adjCV    0.2670    0.2679    0.2597    0.2611    0.2614    0.2648    0.2674
##        42 comps  43 comps  44 comps  45 comps  46 comps  47 comps  48 comps
## CV       0.2727    0.2709    0.2711    0.2703    0.2715    0.2744    0.2755
## adjCV    0.2698    0.2676    0.2679    0.2673    0.2684    0.2711    0.2722
##        49 comps  50 comps  51 comps  52 comps  53 comps  54 comps  55 comps
## CV       0.2769    0.2782    0.2796    0.2801    0.2834    0.2856    0.2825
## adjCV    0.2735    0.2746    0.2760    0.2767    0.2796    0.2819    0.2785
##        56 comps  57 comps  58 comps  59 comps  60 comps  61 comps  62 comps
## CV       0.2839    0.2880    0.2783    0.2771    0.2801    0.2830    0.2834
## adjCV    0.2801    0.2853    0.2741    0.2731    0.2759    0.2782    0.2795
##        63 comps  64 comps  65 comps  66 comps  67 comps  68 comps   69 comps
## CV       0.2877    0.3245    0.3212    0.3172    0.3301    0.3342  1.677e+11
## adjCV    0.2827    0.3178    0.3134    0.3092    0.3217    0.3258  1.589e+11
##         70 comps   71 comps   72 comps   73 comps   74 comps   75 comps
## CV     1.879e+11  9.711e+11  1.343e+12  1.549e+12  1.132e+12  1.425e+12
## adjCV  1.780e+11  9.205e+11  1.274e+12  1.469e+12  1.074e+12  1.353e+12
##         76 comps   77 comps   78 comps   79 comps   80 comps   81 comps
## CV     1.226e+12  1.280e+12  1.426e+12  1.196e+12  1.482e+12  2.334e+12
```

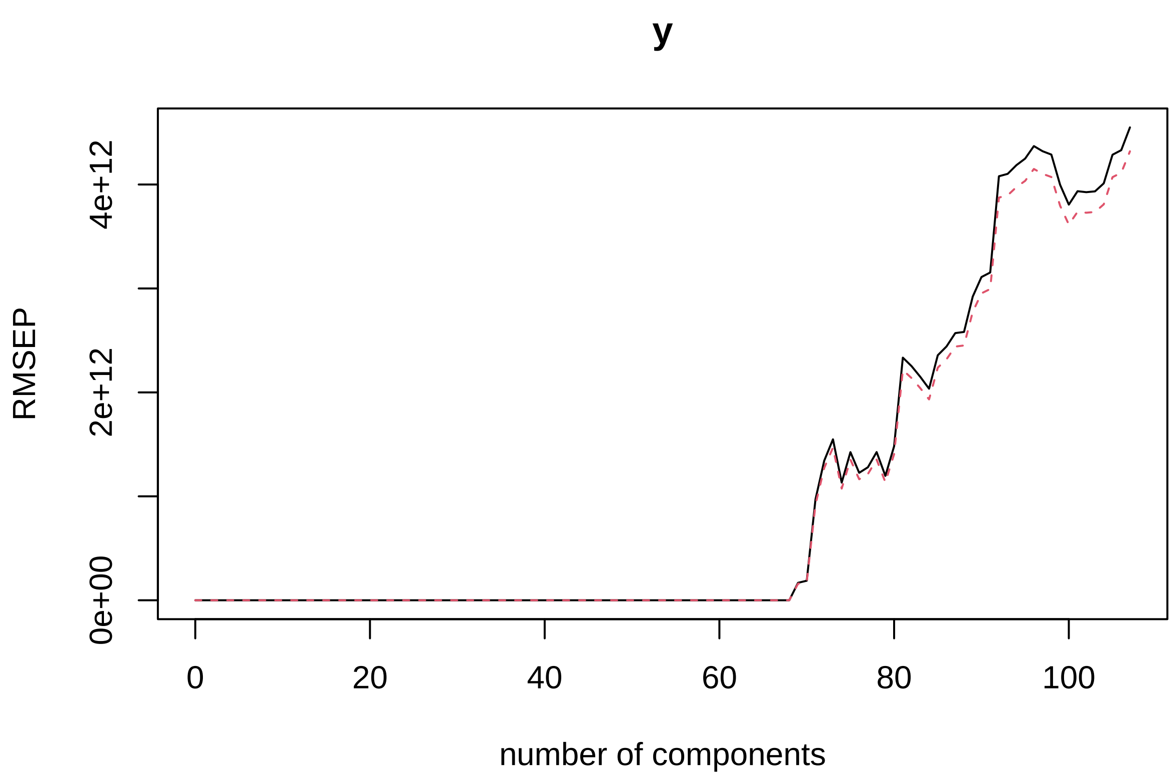
```
## adjCV  1.164e+12  1.215e+12  1.354e+12  1.136e+12  1.407e+12  2.215e+12
##         82 comps   83 comps   84 comps   85 comps   86 comps   87 comps
## CV     2.252e+12  2.149e+12  2.035e+12  2.357e+12  2.442e+12  2.570e+12
## adjCV  2.137e+12  2.040e+12  1.932e+12  2.238e+12  2.318e+12  2.441e+12
##         88 comps   89 comps   90 comps   91 comps   92 comps   93 comps
## CV     2.581e+12  2.921e+12  3.110e+12  3.153e+12  4.080e+12  4.103e+12
## adjCV  2.451e+12  2.774e+12  2.953e+12  2.995e+12  3.874e+12  3.896e+12
##         94 comps   95 comps  96 comps   97 comps   98 comps   99 comps
## CV     4.186e+12  4.249e+12  4.37e+12  4.320e+12  4.288e+12  3.997e+12
## adjCV  3.975e+12  4.035e+12  4.15e+12  4.103e+12  4.072e+12  3.796e+12
##        100 comps  101 comps  102 comps  103 comps  104 comps  105 comps
## CV     3.806e+12  3.936e+12  3.926e+12  3.934e+12  4.011e+12  4.286e+12
## adjCV  3.615e+12  3.737e+12  3.729e+12  3.736e+12  3.809e+12  4.071e+12
##        106 comps  107 comps
## CV     4.330e+12   4.55e+12
## adjCV  4.113e+12   4.32e+12
##
## TRAINING: % variance explained
##    1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X    65.59    72.62    77.58    81.83    85.01    87.62    89.55    91.09
## y    51.99    54.92    60.62    67.50    70.75    77.19    78.77    78.86
##    9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X    92.35     93.45     94.34     95.18     95.93     96.49     96.96
## y    84.98     85.01     85.28     85.35     86.61     88.37     88.85
##    16 comps  17 comps  18 comps  19 comps  20 comps  21 comps  22 comps
## X     97.40     97.70     97.97     98.22     98.42     98.61     98.79
## y     90.72     90.72     90.74     90.74     90.76     90.83     91.08
##    23 comps  24 comps  25 comps  26 comps  27 comps  28 comps  29 comps
## X     98.94     99.05     99.17     99.25     99.34     99.41     99.48
## y     91.42     91.61     91.89     91.90     91.90     91.91     92.17
##    30 comps  31 comps  32 comps  33 comps  34 comps  35 comps  36 comps
## X     99.53     99.58     99.63     99.66      99.7     99.73     99.76
## y     92.70     93.12     93.12     93.24      93.4     93.46     93.47
##    37 comps  38 comps  39 comps  40 comps  41 comps  42 comps  43 comps
## X     99.79     99.81     99.83     99.85     99.87     99.89     99.90
## y     93.80     93.83     93.83     93.84     93.86     93.86     93.95
##    44 comps  45 comps  46 comps  47 comps  48 comps  49 comps  50 comps
## X     99.92     99.93     99.94     99.95     99.95     99.96     99.97
## y     93.96     93.96     93.97     93.98     93.99     94.03     94.08
##    51 comps  52 comps  53 comps  54 comps  55 comps  56 comps  57 comps
## X     99.97     99.98     99.98     99.98     99.99     99.99     99.99
## y     94.08     94.08     94.17     94.23     94.33     94.34     94.34
##    58 comps  59 comps  60 comps  61 comps  62 comps  63 comps  64 comps
## X     99.99    100.00    100.00    100.00    100.00    100.00    100.00
## y     94.66     94.66     94.68     94.74     94.74     94.88     94.88
##    65 comps  66 comps  67 comps  68 comps  69 comps  70 comps  71 comps
## X    100.00    100.00    100.00    100.00    100.00    100.00    100.00
## y     95.09     95.26     95.27     95.29     95.37     95.37     95.47
##    72 comps  73 comps  74 comps  75 comps  76 comps  77 comps  78 comps
## X    100.00    100.00    100.00    100.00    100.00    100.00    100.00
## y     95.48     95.53     95.55     95.57     95.64     95.64     95.64
##    79 comps  80 comps  81 comps  82 comps  83 comps  84 comps  85 comps
## X    100.00    100.00    100.00    100.00    100.00    100.00    100.00
## y     95.64     95.65     95.65     95.67     95.69     95.94     95.94
```

```
##     86 comps  87 comps  88 comps  89 comps  90 comps  91 comps  92 comps
## X    100.00       100       100    100.00    100.00    100.00    100.00
## y     95.99        96        96     96.03     96.04     96.12     96.14
##     93 comps  94 comps  95 comps  96 comps  97 comps  98 comps  99 comps
## X    100.00    100.00    100.00    100.00    100.00    100.00    100.00
## y     96.14     96.23     96.24     96.27     96.27     96.29     96.35
##     100 comps  101 comps  102 comps  103 comps  104 comps  105 comps  106 comps
## X      100.00     100.00     100.00     100.00     100.00     100.00     100.00
## y       96.35      96.41      96.42      96.43      96.44      96.45      96.45
##     107 comps
## X      100.00
## y       96.46
```

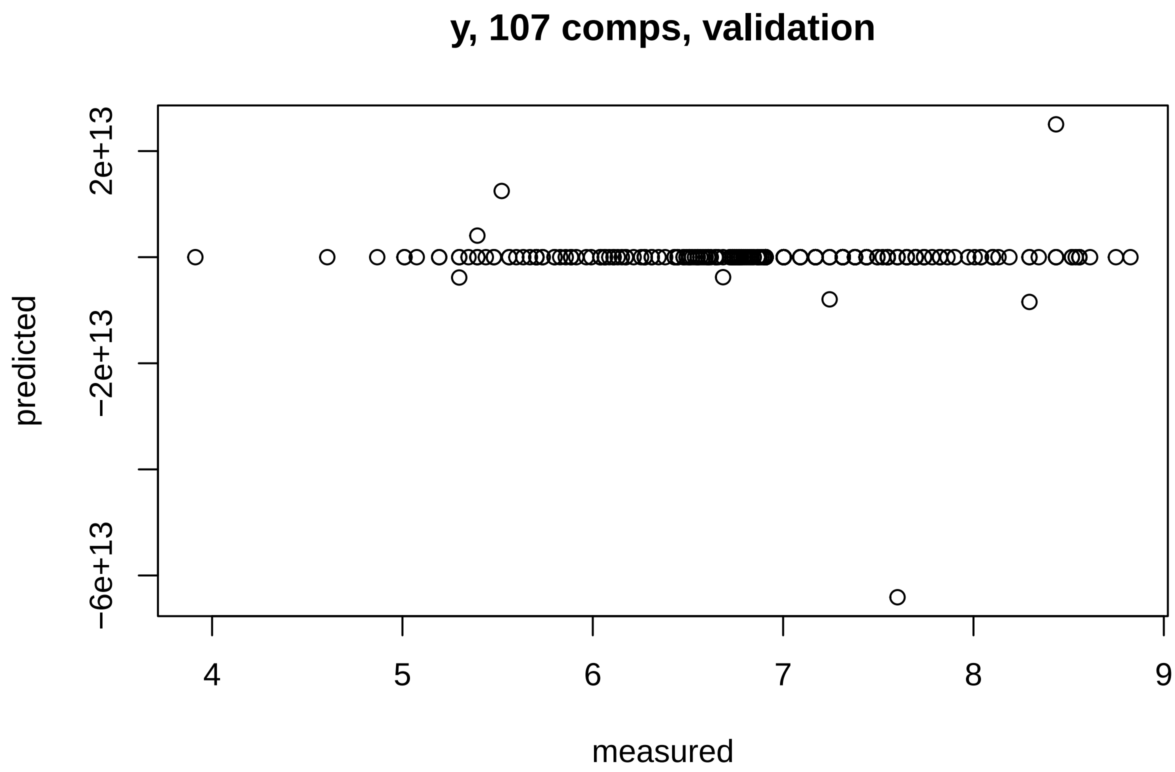
(b)

34 components seem to be optimal

```r
validationplot(pcr_fit)
```

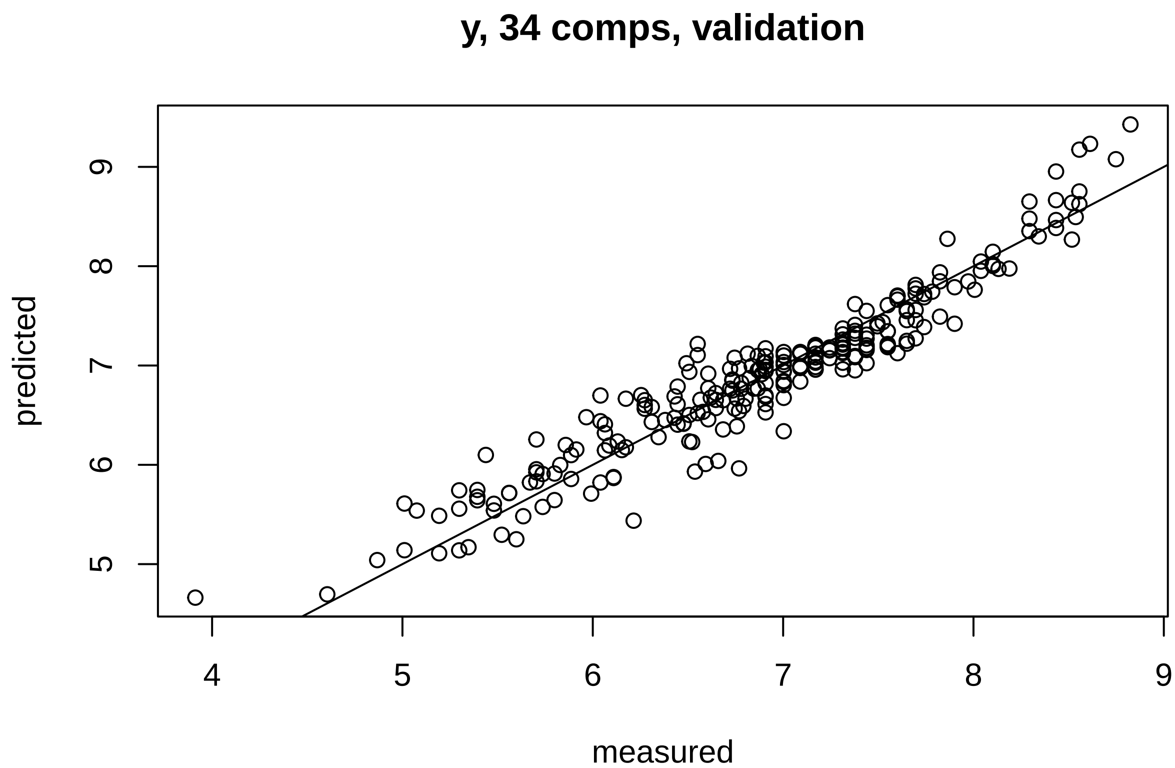



```r
predplot(pcr_fit)
```



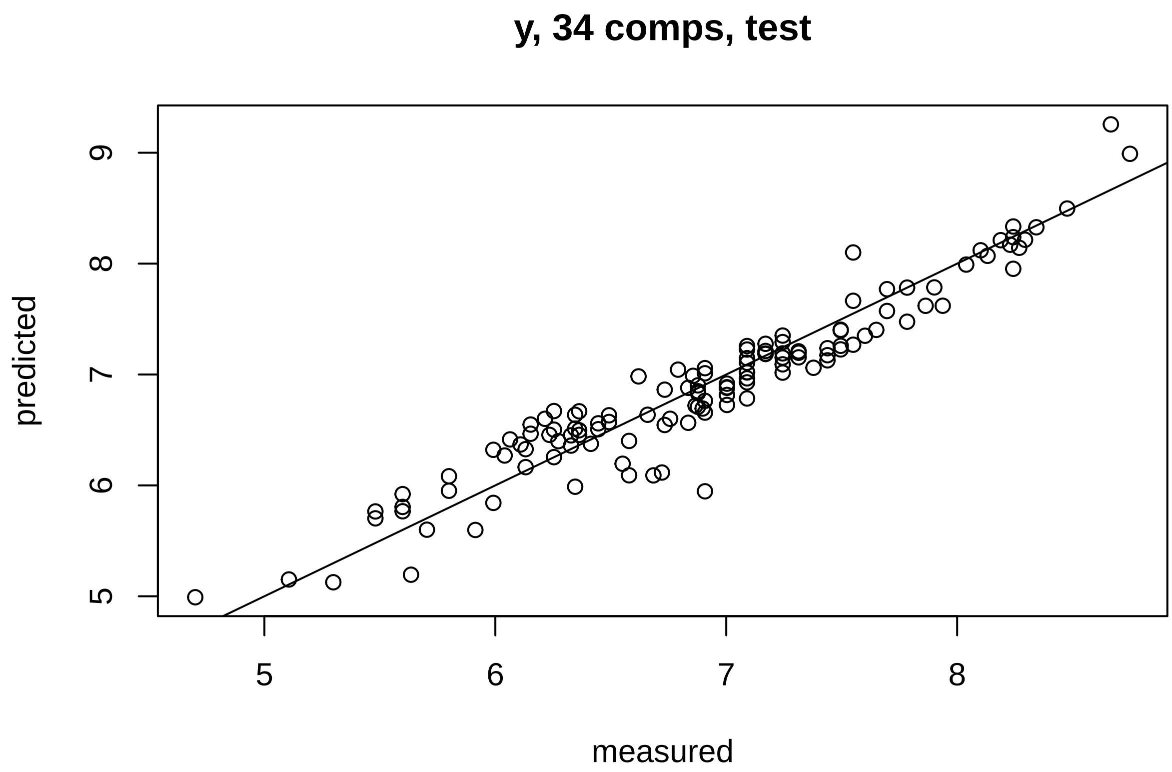
(c)

```
predplot(pcr_fit, ncomp=34, line=TRUE)
```



(d)

```
predplot(pcr_fit, newdata=test, ncomp=34, line=TRUE)
```



4

(a)

```r
plsr_fit = plsr(y~., data=train, scale=TRUE, validation="CV", segments=10,
↪  segment.type="random")
summary(plsr_fit)
```

```
## Data:    X dimension: 245 107
##  Y dimension: 245 1
## Fit method: kernelpls
## Number of components considered: 107
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV          0.8748   0.5880   0.4068   0.3569   0.3201   0.2934   0.2842
## adjCV       0.8748   0.5877   0.4053   0.3546   0.3176   0.2917   0.2827
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV      0.2808   0.2797   0.2819    0.2795    0.2758    0.2762    0.2725
## adjCV   0.2789   0.2777   0.2789    0.2762    0.2725    0.2727    0.2693
##        14 comps  15 comps  16 comps  17 comps  18 comps  19 comps  20 comps
## CV       0.2712    0.2724    0.2765    0.2781    0.2813    0.2850    0.2885
## adjCV    0.2682    0.2693    0.2730    0.2746    0.2774    0.2808    0.2841
##        21 comps  22 comps  23 comps  24 comps  25 comps  26 comps  27 comps
```

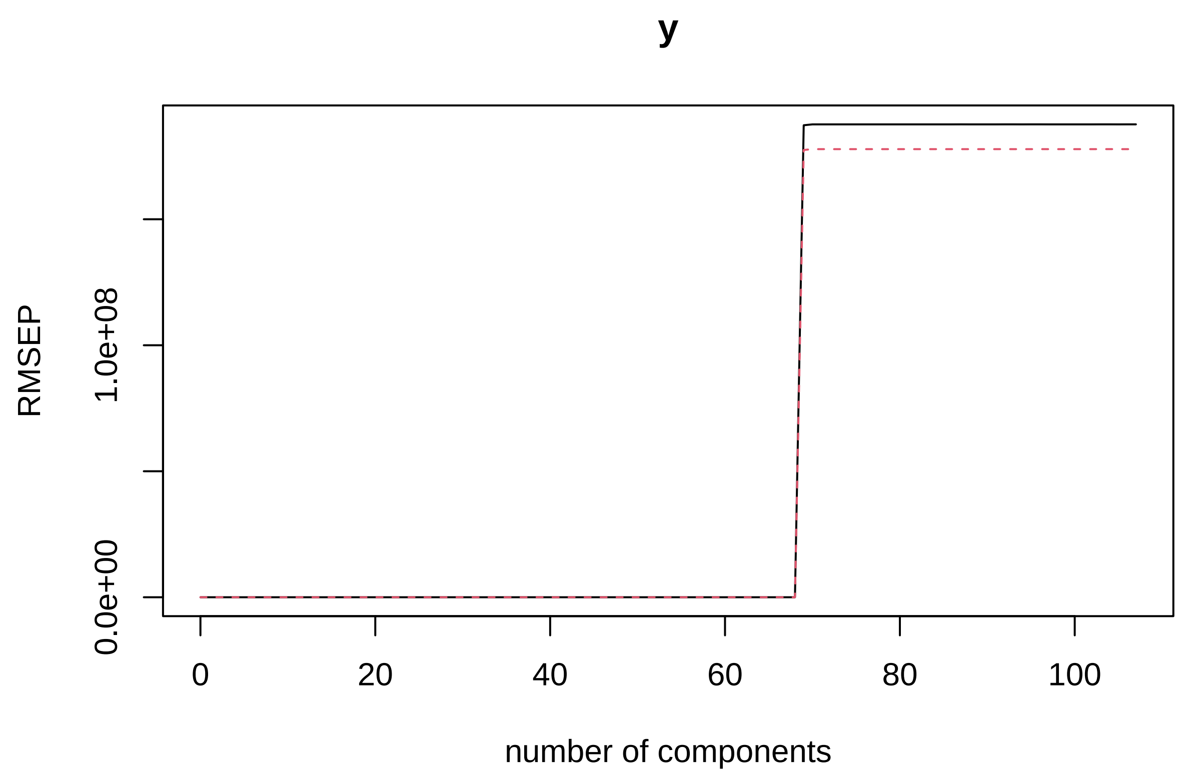
```
## CV       0.2908    0.2953    0.2985    0.2973    0.2976    0.2989    0.3010
## adjCV    0.2862    0.2898    0.2925    0.2914    0.2916    0.2928    0.2947
##        28 comps  29 comps  30 comps  31 comps  32 comps  33 comps  34 comps
## CV       0.3011    0.3016    0.3028    0.3105    0.3156    0.3162    0.3221
## adjCV    0.2949    0.2953    0.2965    0.3036    0.3081    0.3086    0.3139
##        35 comps  36 comps  37 comps  38 comps  39 comps  40 comps  41 comps
## CV       0.3320    0.3419    0.3542    0.3618    0.3694    0.3772    0.3822
## adjCV    0.3229    0.3321    0.3435    0.3504    0.3573    0.3646    0.3693
##        42 comps  43 comps  44 comps  45 comps  46 comps  47 comps  48 comps
## CV       0.3852    0.3870    0.3893    0.3894    0.3893    0.3904    0.3904
## adjCV    0.3722    0.3739    0.3760    0.3760    0.3759    0.3769    0.3770
##        49 comps  50 comps  51 comps  52 comps  53 comps  54 comps  55 comps
## CV       0.3909    0.3913    0.3914    0.3911    0.3902    0.3881    0.3873
## adjCV    0.3774    0.3778    0.3779    0.3776    0.3768    0.3748    0.3741
##        56 comps  57 comps  58 comps  59 comps  60 comps  61 comps  62 comps
## CV       0.3861    0.3859    0.3859    0.3855    0.3854    0.3849    0.3844
## adjCV    0.3729    0.3727    0.3727    0.3724    0.3723    0.3718    0.3714
##        63 comps  64 comps  65 comps  66 comps  67 comps  68 comps   69 comps
## CV       0.3841     0.384    0.3839    0.3839    0.3839    0.3839  187312648
## adjCV    0.3711     0.371    0.3709    0.3709    0.3709    0.3709  177498811
##         70 comps   71 comps   72 comps   73 comps   74 comps   75 comps
## CV     187660409  187661188  187662567  187662309  187661897  187662069
## adjCV  177828353  177829092  177830398  177830153  177829763  177829927
##         76 comps   77 comps   78 comps   79 comps   80 comps   81 comps
## CV     187662965  187661921  187660765  187662837  187663024  187661987
## adjCV  177830775  177829786  177828691  177830654  177830831  177829849
##         82 comps   83 comps   84 comps   85 comps   86 comps   87 comps
## CV     187662944  187661460  187661877  187662531  187662776  187663473
## adjCV  177830756  177829349  177829744  177830364  177830596  177831257
##         88 comps   89 comps   90 comps   91 comps   92 comps   93 comps
## CV     187661927  187661295  187662294  187662485  187664325  187663795
## adjCV  177829792  177829193  177830139  177830320  177832064  177831561
##         94 comps   95 comps   96 comps   97 comps   98 comps   99 comps
## CV     187661586  187664029  187664972  187662391  187661370  187660661
## adjCV  177829468  177831784  177832677  177830232  177829264  177828592
##        100 comps  101 comps  102 comps  103 comps  104 comps  105 comps
## CV     187662029  187661499  187663043  187664291  187664773  187662172
## adjCV  177829888  177829386  177830850  177832032  177832488  177830024
##        106 comps  107 comps
## CV     187663319  187663770
## adjCV  177831111  177831539
##
## TRAINING: % variance explained
##    1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps
## X    65.46    70.44    75.15    78.75    81.42    84.41    86.51    88.79
## y    55.58    81.08    86.67    89.50    90.97    91.62    92.18    92.48
##    9 comps  10 comps  11 comps  12 comps  13 comps  14 comps  15 comps
## X    90.13     91.49     92.16     93.02     93.90     94.49     95.19
## y    93.00     93.37     93.65     93.81     93.95     94.04     94.08
##    16 comps  17 comps  18 comps  19 comps  20 comps  21 comps  22 comps
## X     95.69     96.37     96.62     97.10     97.65     97.96     98.09
## y     94.16     94.21     94.30     94.36     94.40     94.46     94.61
##    23 comps  24 comps  25 comps  26 comps  27 comps  28 comps  29 comps
## X     98.22     98.38     98.50     98.68     98.83     99.02     99.08
```

```
## y     94.69     94.75     94.81     94.85     94.89     94.92     94.96
##    30 comps  31 comps  32 comps  33 comps  34 comps  35 comps  36 comps
## X     99.20     99.33     99.38     99.45     99.49     99.54     99.57
## y     94.99     95.02     95.08     95.10     95.15     95.19     95.22
##    37 comps  38 comps  39 comps  40 comps  41 comps  42 comps  43 comps
## X     99.60     99.63     99.65     99.68     99.72     99.77     99.80
## y     95.25     95.28     95.30     95.32     95.33     95.33     95.33
##    44 comps  45 comps  46 comps  47 comps  48 comps  49 comps  50 comps
## X     99.82     99.84     99.85     99.87     99.88     99.90     99.91
## y     95.34     95.35     95.35     95.36     95.36     95.36     95.36
##    51 comps  52 comps  53 comps  54 comps  55 comps  56 comps  57 comps
## X     99.92     99.93     99.94     99.95     99.96     99.97     99.97
## y     95.37     95.37     95.37     95.37     95.37     95.37     95.37
##    58 comps  59 comps  60 comps  61 comps  62 comps  63 comps  64 comps
## X     99.98     99.98     99.99     99.99     99.99     99.99    100.00
## y     95.37     95.37     95.37     95.37     95.37     95.37     95.37
##    65 comps  66 comps  67 comps  68 comps  69 comps  70 comps  71 comps
## X    100.00    100.00    100.00    100.00    100.00    100.00    100.01
## y     95.37     95.37     95.37     95.37     95.37     95.37     95.37
##    72 comps  73 comps  74 comps  75 comps  76 comps  77 comps  78 comps
## X    100.01    100.02    100.02    100.03    100.03    100.04    100.04
## y     95.37     95.37     95.37     95.37     95.37     95.37     95.37
##    79 comps  80 comps  81 comps  82 comps  83 comps  84 comps  85 comps
## X    100.05    100.05    100.06    100.06    100.07    100.07    100.08
## y     95.37     95.37     95.37     95.37     95.37     95.37     95.37
##    86 comps  87 comps  88 comps  89 comps  90 comps  91 comps  92 comps
## X    100.08    100.09    100.09    100.10    100.10    100.11    100.11
## y     95.37     95.37     95.37     95.37     95.37     95.37     95.37
##    93 comps  94 comps  95 comps  96 comps  97 comps  98 comps  99 comps
## X    100.12    100.12    100.13    100.13    100.14    100.14    100.15
## y     95.37     95.37     95.37     95.37     95.37     95.37     95.37
##    100 comps  101 comps  102 comps  103 comps  104 comps  105 comps  106 comps
## X     100.15     100.16     100.16     100.17     100.18     100.18     100.19
## y      95.37      95.37      95.37      95.37      95.37      95.37      95.37
##    107 comps
## X     100.19
## y      95.37
```

(b)

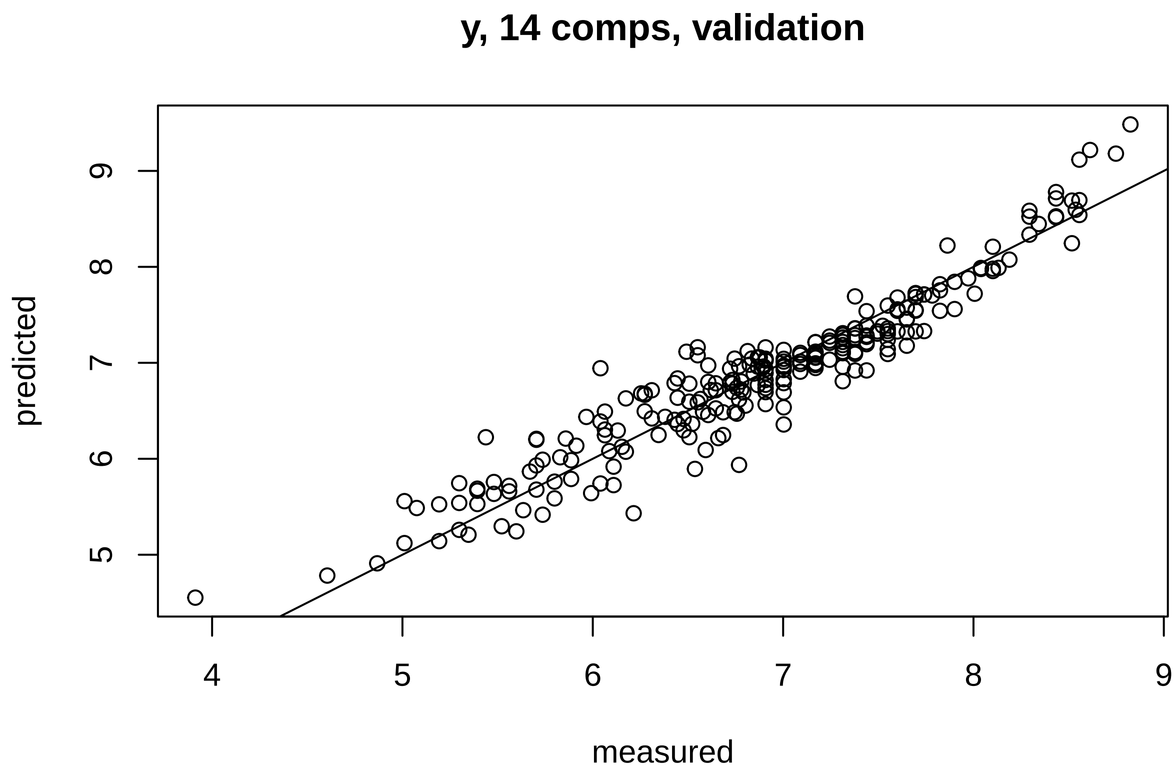
14 components seem to be optimal

```r
validationplot(plsr_fit)
```



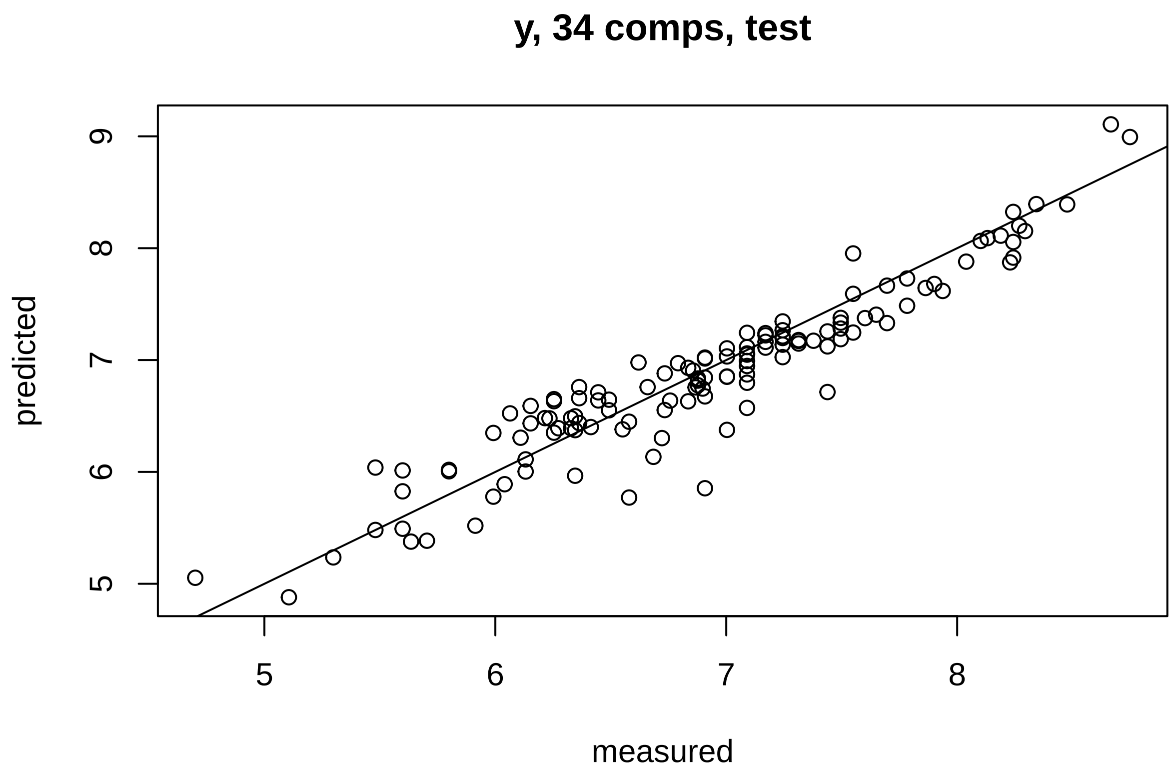
(c)

```
predplot(plsr_fit, ncomp=14, line=TRUE)
```



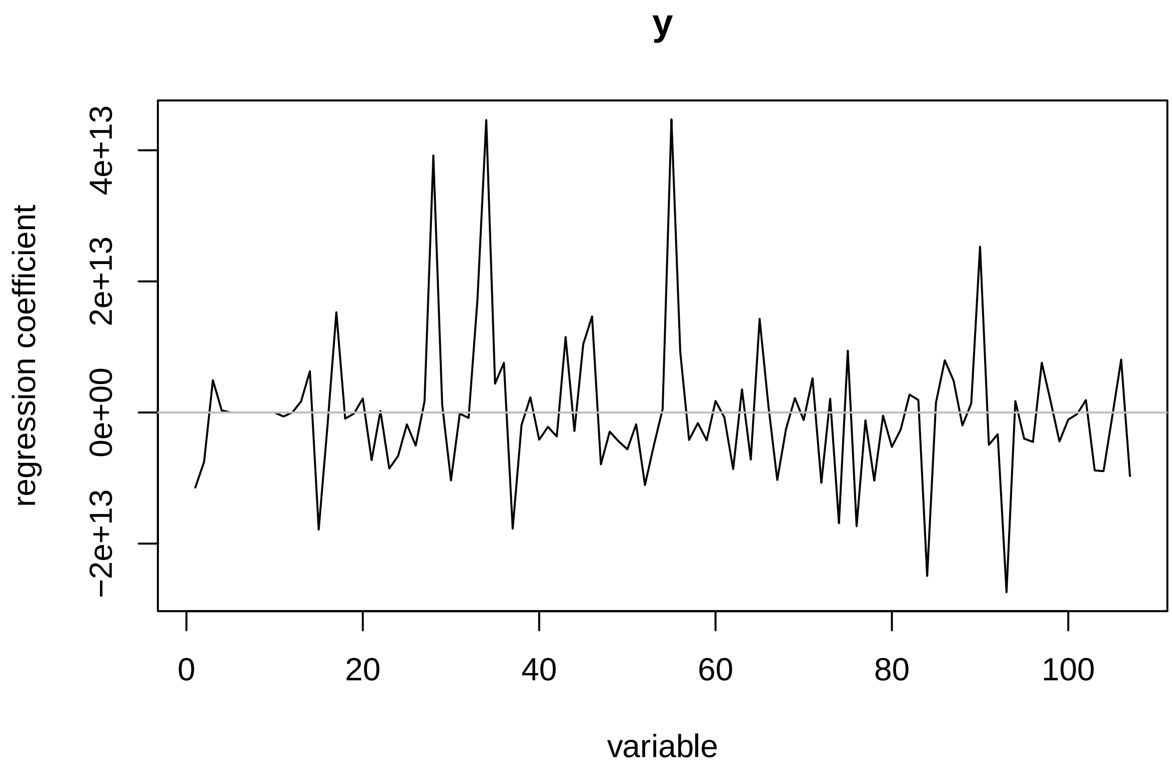
(d)

```
predplot(plsr_fit, newdata=test, ncomp=34, line=TRUE)
```

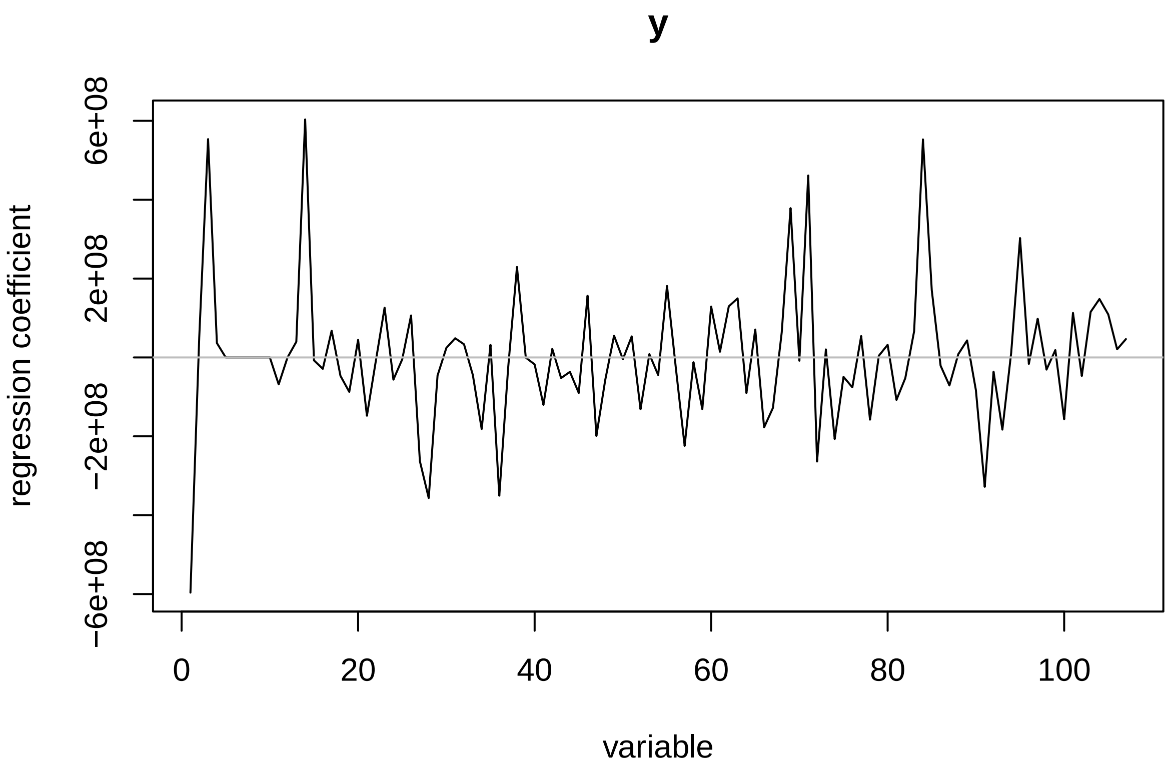


(e)

```r
coefplot(pcr_fit)
```




```
coefplot(plsr_fit)
```


y
regression coefficient
6e+08
2e+08
−2e+08
−6e+08
0
20
40
60
80
100
variable