FBI WARNING

People under 18 ages are not admitted.

PUBG Finish Placement Prediction

Let's let the data do the talking!

CONTENTS


Data Description

Data Description

kaggle Search Competitions Datasets Notebooks Discussion Courses •••

Overview Data Notebooks Discussion Leaderboard Rules Team My Submissions Late Submission

Overview

Description

Evaluation

Kernels-FAQ

Prizes

So, where we droppin' boys and girls?

Battle Royale-style video games have taken the world by storm. 100 players are dropped onto an island empty-handed and must explore, scavenge, and eliminate other players until only one is left standing, all while the play zone continues to shrink.

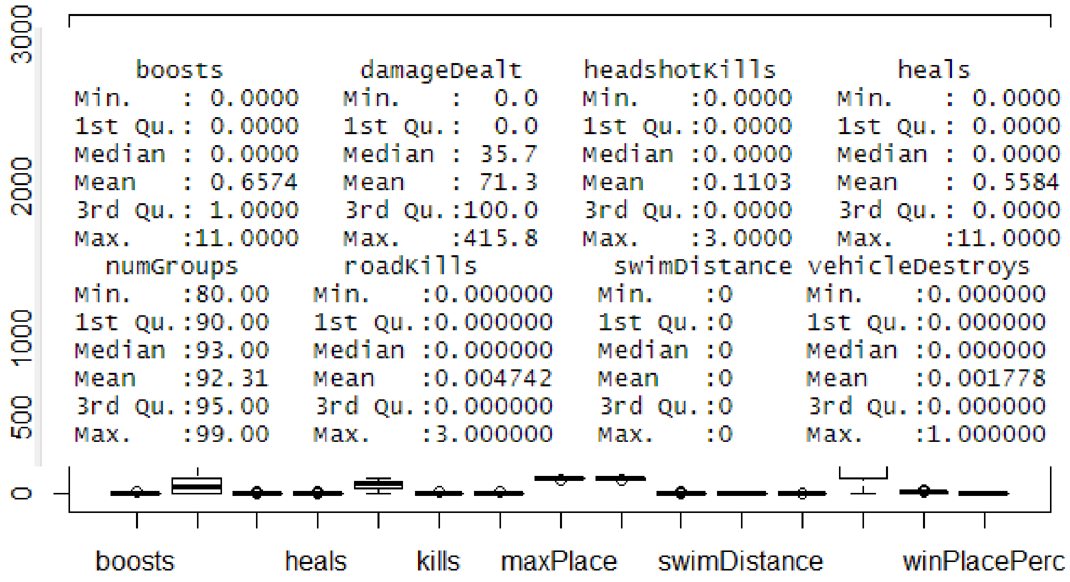
PlayerUnknown's BattleGrounds (PUBG) has

Data Processing and Analysis

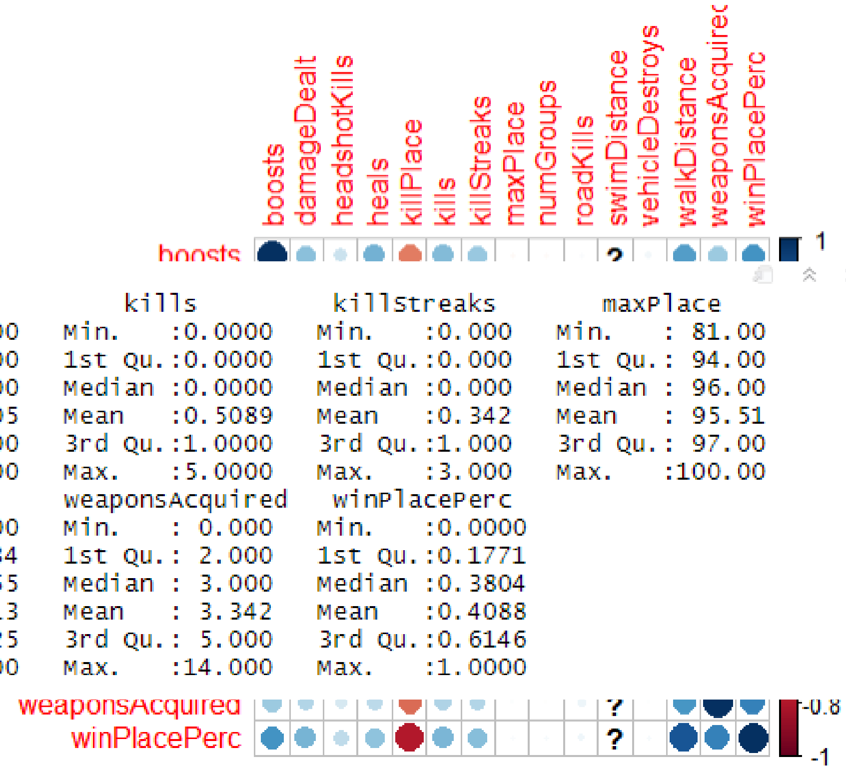
winPlacePerc

boosts	damageDealt	heals	killPlace	kills	killStreaks	headshotKills
maxPlace	numGroups	roadKills	vehicleDestr oys	walkDistance	weaponsAcq uired	swinDistance

Data Processing and Analysis




```
    boosts          damageDealt      headshotKills        heals            killPlace           kills           killStreaks       maxPlace
 Min.    : 0.0000  Min.   :  0.0   Min.    :0.0000  Min.    : 0.0000  Min.    :  2.00  Min.    :0.0000  Min.    :0.000  Min.   : 81.00
 1st Qu.: 0.0000   1st Qu.:  0.0   1st Qu.:0.0000   1st Qu.: 0.0000   1st Qu.: 35.00   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.: 94.00
 Median : 0.0000   Median : 35.7   Median :0.0000   Median : 0.0000   Median : 59.00   Median :0.0000   Median :0.000   Median : 96.00
 Mean   : 0.6574   Mean   : 71.3   Mean   :0.1103   Mean   : 0.5584   Mean   : 56.05   Mean   :0.5089   Mean   :0.342   Mean   : 95.51
 3rd Qu.: 1.0000   3rd Qu.:100.0   3rd Qu.:0.0000   3rd Qu.: 0.0000   3rd Qu.: 78.00   3rd Qu.:1.0000   3rd Qu.:1.000   3rd Qu.: 97.00
 Max.   :11.0000   Max.   :415.8   Max.   :3.0000   Max.   :11.0000   Max.   :100.00   Max.   :5.0000   Max.   :3.000   Max.   :100.00
  numGroups        roadKills          swimDistance   vehicleDestroys      walkDistance      weaponsAcquired    winPlacePerc
 Min.   :80.00   Min.   :0.000000   Min.   :0     Min.   :0.000000   Min.   :   0.00   Min.   : 0.000   Min.   :0.0000
 1st Qu.:90.00   1st Qu.:0.000000   1st Qu.:0     1st Qu.:0.000000   1st Qu.:  94.34   1st Qu.: 2.000   1st Qu.:0.1771
 Median :93.00   Median :0.000000   Median :0     Median :0.000000   Median : 392.55   Median : 3.000   Median :0.3804
 Mean   :92.31   Mean   :0.004742   Mean   :0     Mean   :0.001778   Mean   : 730.13   Mean   : 3.342   Mean   :0.4088
 3rd Qu.:95.00   3rd Qu.:0.000000   3rd Qu.:0     3rd Qu.:0.000000   3rd Qu.:1190.25   3rd Qu.: 5.000   3rd Qu.:0.6146
 Max.   :99.00   Max.   :3.000000   Max.   :0     Max.   :1.000000   Max.   :3014.00   Max.   :14.000   Max.   :1.0000
```

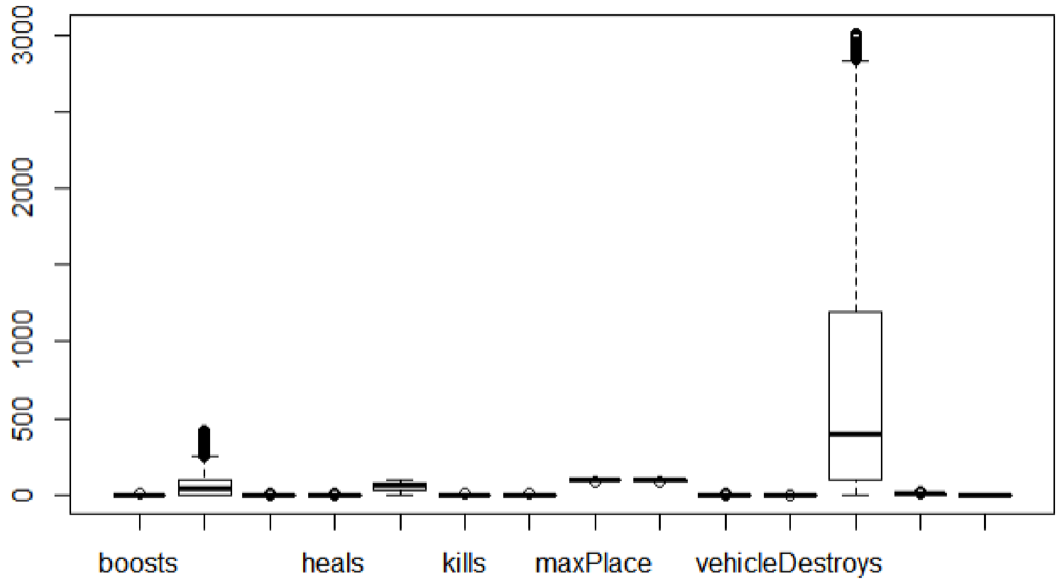


summary()

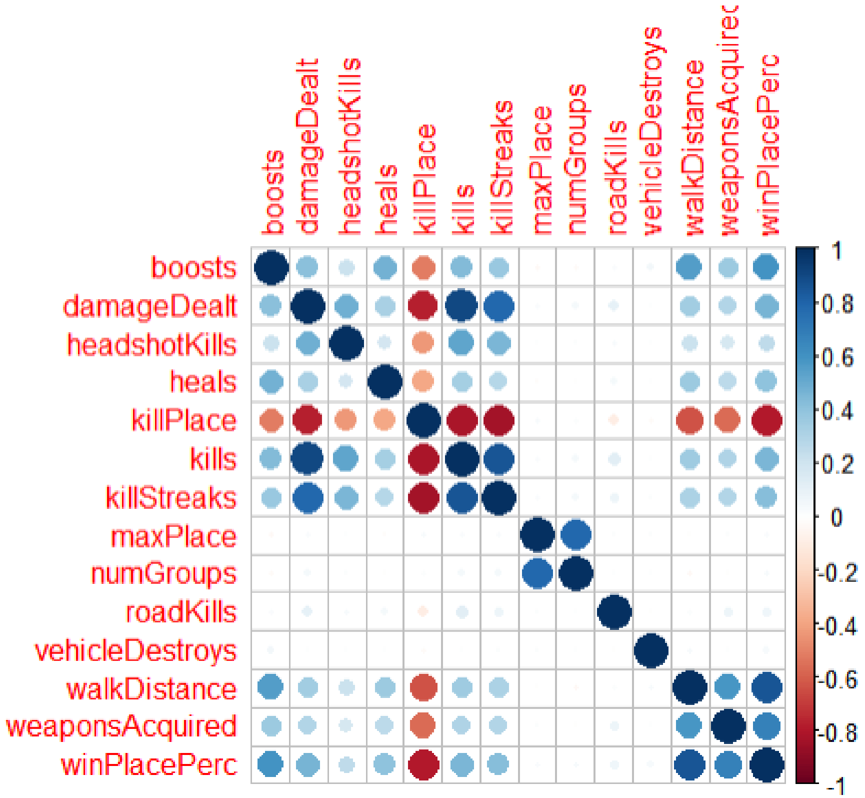
boxplot()

cor(), corrplot()

Data Processing and Analysis



boxplot()



cor(), corrplot(

Model Building and Selection

- LS
- Lasso, Ridge, PCA
- SVM, Neural Network
- KNN, Regression Learner (MATLAB)

Model Building and Selection

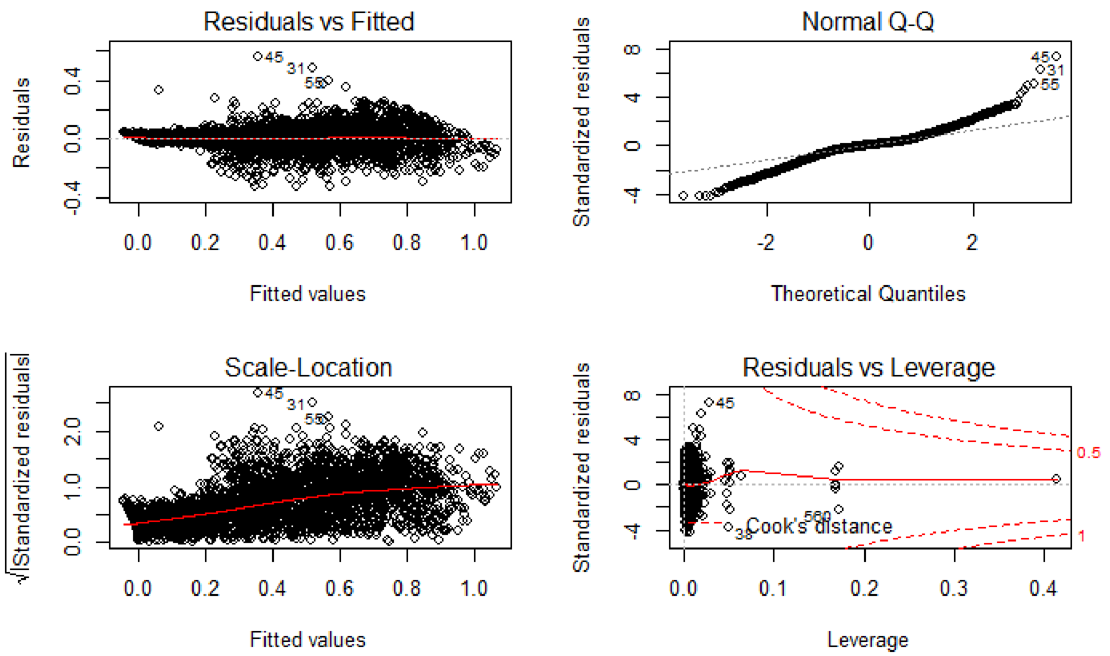
L S

LM: Fit All Variables——lm.fit1

lm.fit1 = lm(winPlacePerc ~ .)

```
lm(formula = winPlacePerc ~ ., data = data)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.32648 -0.03009  0.00360  0.03344  0.55674 

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)    
(Intercept)      3.131e-01  4.995e-02   6.268 4.13e-10 ***
boosts           2.333e-02  1.397e-03  16.697  < 2e-16 ***
damageDealt      6.331e-05  3.468e-05   1.826   0.0680 .  
headshotKills    1.293e-03  4.557e-03   0.284   0.7767    
heals            2.016e-03  1.114e-03   1.809   0.0705 .  
killPlace       -1.088e-02  1.598e-04 -68.050  < 2e-16 ***
kills           -7.588e-02  4.614e-03 -16.445  < 2e-16 ***
killStreaks     -2.196e-01  6.523e-03 -33.673  < 2e-16 ***
maxPlace         5.111e-04  8.374e-04   0.610   0.5416    
numGroups        7.069e-03  6.272e-04  11.271  < 2e-16 ***
roadKills        3.024e-02  1.684e-02   1.795   0.0727 .  
vehicleDestroys  2.857e-02  3.187e-02   0.896   0.3702    
walkDistance     1.085e-04  2.801e-06  38.731  < 2e-16 ***
weaponsAcquired  5.124e-03  7.770e-04   6.595 4.93e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.07783 on 3360 degrees of freedom
Multiple R-squared:  0.9181,	Adjusted R-squared:  0.9177 
F-statistic:  2895 on 13 and 3360 DF,  p-value: < 2.2e-16
```

summary()



plot()

LM: Fit All Variables——lm.fit1

lm.fit1 = lm(winPlacePerc ~ .)

R-squared: 0.9181   Adjusted R-squared: 0.9177

```
   boosts      damageDealt   headshotKills          heals       killPlace           kills     killStreaks
 1.759913         5.526565        1.381786       1.345420        9.303072        8.587941        5.769179
 maxPlace        numGroups       roadKills vehicleDestroys   walkDistance weaponsAcquired
 2.636388         2.633396        1.026993       1.004514        2.687854        1.881032
```

vif()

LM: Delete Muticollinearity——lm.fit2

lm.fit2 = lm(winPlacePerc~. – killPlace - kills)

R-squared: 0.8049 Adjusted R-squared: 0.8043

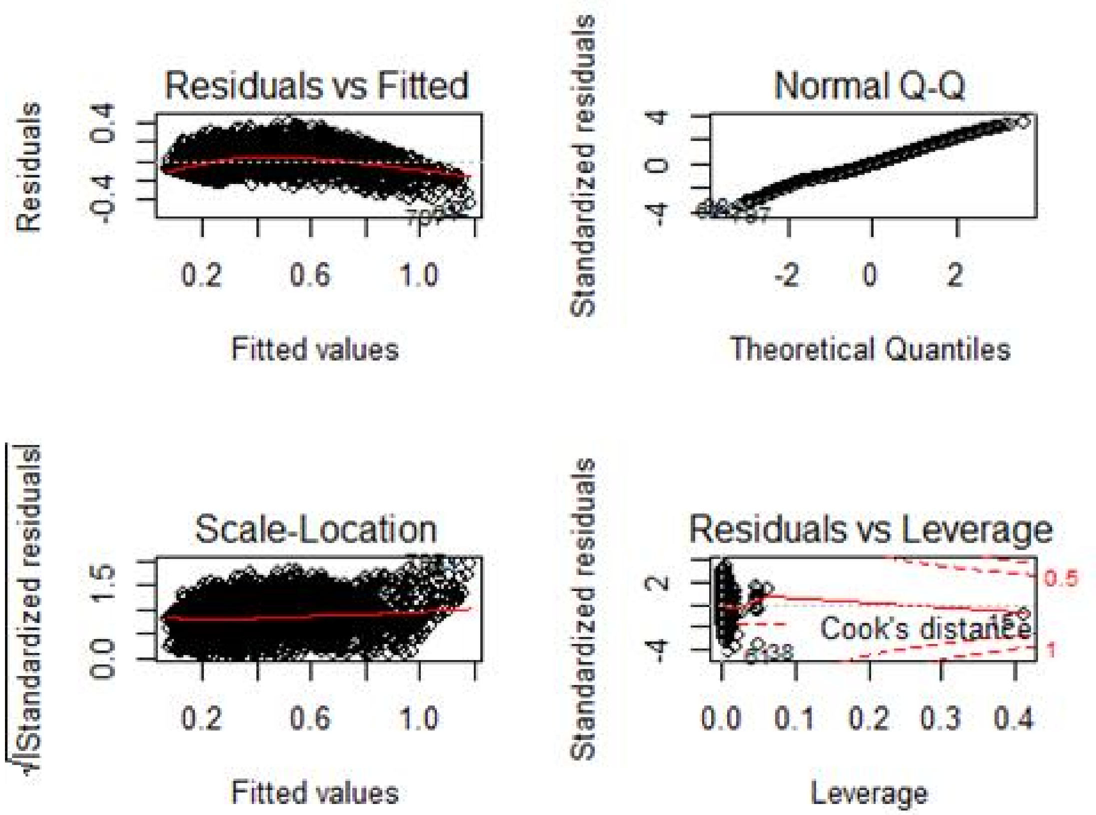
```
   boosts      damageDealt    headshotKills            heals      killStreaks         maxPlace        numGroups 
 1.738499         2.993433         1.333095         1.342214         2.739349         2.620701         2.629676 
roadKills vehicleDestroys     walkDistance weaponsAcquired 
 1.017941         1.004137         1.964768         1.572167 
```

vif()

LM: stepAIC(lm.fit2)——lm.fit3

lm.fit3 = lm(winPlacePerc ~ boosts + damageDealt + heals + killStreaks + maxPlace + numGroups + roadKills + walkDistance + weaponsAcquired)

R-squared: 0.8048   Adjusted R-squared: 0.8043



Compare lm.fit2 and lm.fit3

```
Analysis of Variance Table

Model 1: winPlacePerc ~ boosts + damageDealt + heals + killStreaks + maxPlace +
    numGroups + roadKills + walkDistance + weaponsAcquired
Model 2: winPlacePerc ~ (boosts + damageDealt + headshotKills + heals +
    killPlace + kills + killStreaks + maxPlace + numGroups +
    roadKills + vehicleDestroys + walkDistance + weaponsAcquired) -
    killPlace - kills
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1   3364 48.480
2   3362 48.452  2    0.0273 0.9471  0.388
```

anova(lm.fit3, lm.fit2)

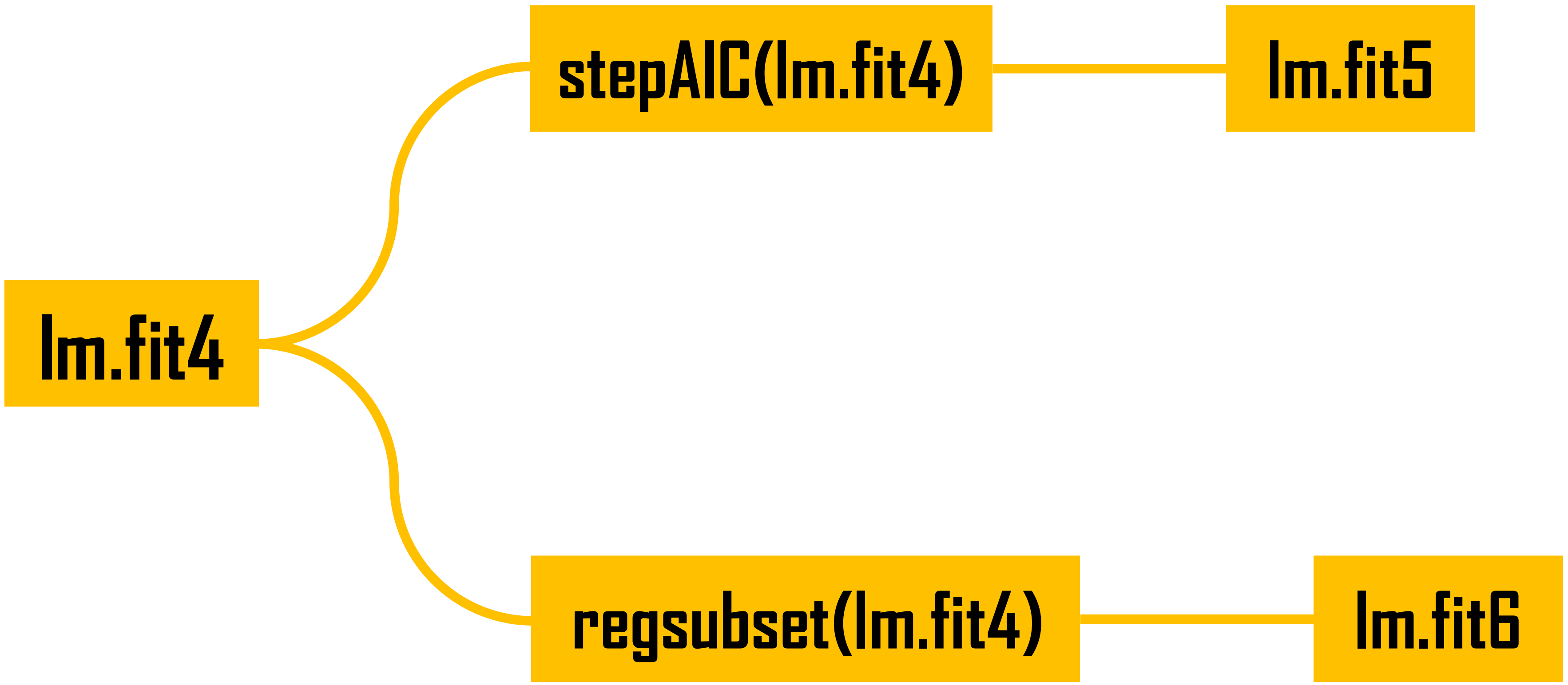
	df <dbl>	AIC <dbl>
lm.fit3	11	-4717.9
lm.fit2	13	-4715.8

AIC(lm.fit3, lm.fit2)

LM: Add Quadratic Forms ——lm.fit4

lm.fit4 = lm(winPlacePer Groups +
roadKills + wa + I(heals^2) +
I(killStreaks^2 stance^2) +
I(weaponsAcq

```
Call:
lm(formula = winPlacePerc ~ boosts + damageDealt + heals + killStreaks +
    maxPlace + numGroups + roadKills + walkDistance + weaponsAcquired +
    I(boosts^2) + I(damageDealt^2) + I(heals^2) + I(killStreaks^2) +
    I(maxPlace^2) + I(numGroups^2) + I(roadKills^2) + I(walkDistance^2) +
    I(weaponsAcquired^2), data = data)

Residuals:
     Min       1Q   Median       3Q      Max
-0.31703 -0.06627 -0.00474  0.06660  0.27875

Coefficients:
                       Estimate Std. Error t value Pr(>|t|)
(Intercept)          -5.973e+00  1.295e+00  -4.613 4.11e-06 ***
boosts                3.598e-02  3.647e-03   9.865  < 2e-16 ***
damageDealt           3.505e-04  7.308e-05   4.796 1.69e-06 ***
heals                 2.338e-03  3.486e-03   0.671   0.5025
killStreaks           8.611e-03  1.269e-02   0.679   0.4974
maxPlace              2.451e-01  3.370e-02   7.275 4.29e-13 ***
numGroups            -1.229e-01  2.222e-02  -5.528 3.48e-08 ***
roadKills             4.260e-02  4.568e-02   0.933   0.3511
walkDistance          4.294e-04  9.235e-06  46.497  < 2e-16 ***
weaponsAcquired       4.704e-02  2.558e-03  18.389  < 2e-16 ***
I(boosts^2)          -1.540e-03  6.009e-04  -2.563   0.0104 *
I(damageDealt^2)     -4.326e-07  1.954e-07  -2.214   0.0269 *
I(heals^2)            1.853e-04  4.723e-04   0.392   0.6949
I(killStreaks^2)     -3.830e-04  8.561e-03  -0.045   0.9643
I(maxPlace^2)        -1.325e-03  1.793e-04  -7.391 1.83e-13 ***
I(numGroups^2)        7.094e-04  1.223e-04   5.803 7.12e-09 ***
I(roadKills^2)        6.769e-04  2.205e-02   0.031   0.9755
I(walkDistance^2)    -9.080e-08  3.347e-09 -27.131  < 2e-16 ***
I(weaponsAcquired^2) -3.377e-03  2.447e-04 -13.801  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1012 on 3355 degrees of freedom
Multiple R-squared:  0.8617,    Adjusted R-squared:  0.8609
F-statistic:  1161 on 18 and 3355 DF,  p-value: < 2.2e-16
```


lm.fit4
stepAIC(lm.fit4)
lm.fit5
regsubset(lm.fit4)
lm.fit6

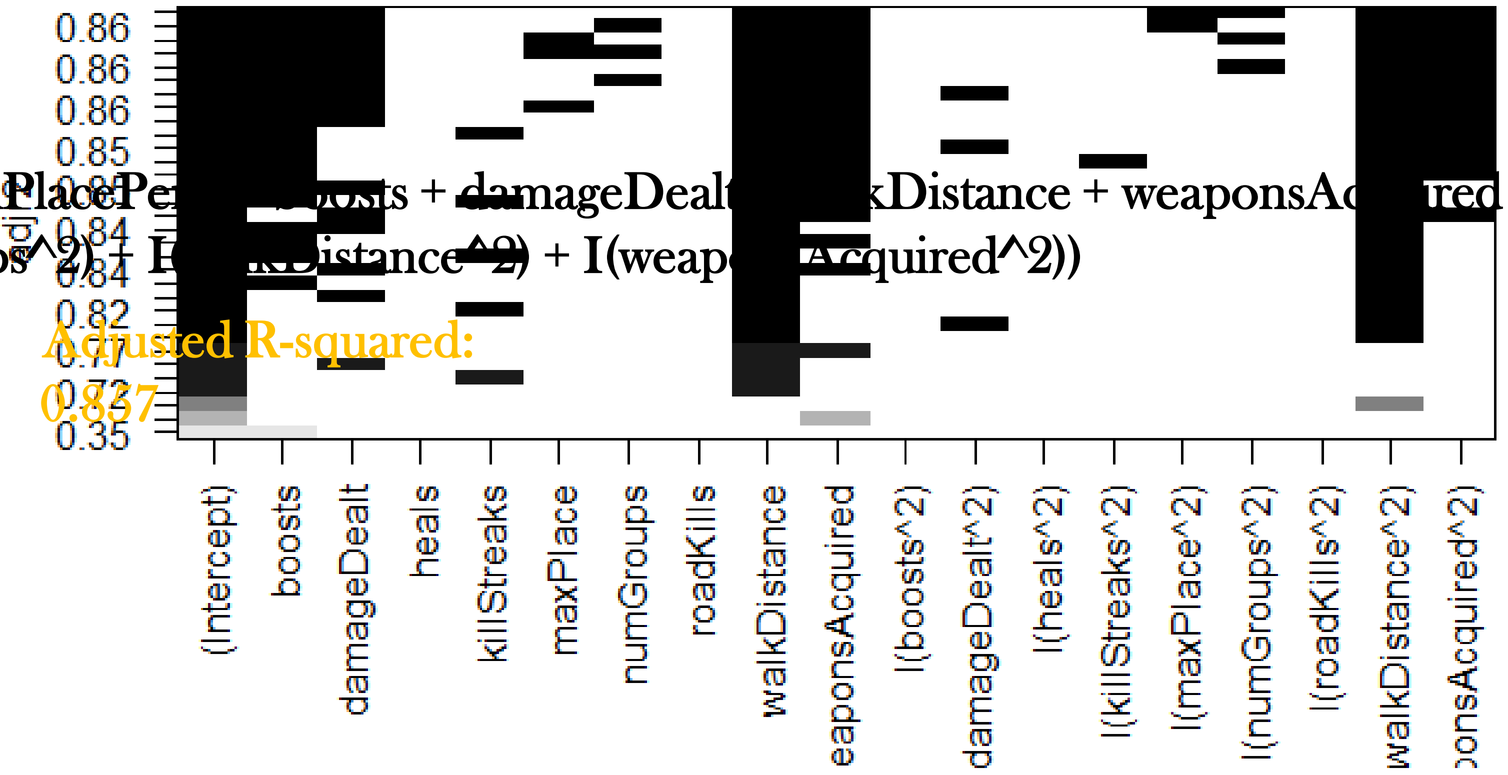
LM: stepAIC(lm.fit4)——lm.fit5

```
lm.fit5 = lm(winPlacePerc~boosts + damageDealt + heals + maxPlace + numGroups + roadKills +
             walkDistance + weaponsAcquired + I(boosts^2) + I(damageDealt^2) + I(maxPlace^2) +
             I(numGroups^2) + I(walkDistance^2) + I(weaponsAcquired^2))
```

R-squared:	Adjusted R-squared:
0.862	0.861

LM: All Subset Method——lm.fit6

lm.fit6=lm(winPlacePerc ~ boosts + damageDealt + walkDistance + weaponsAcquired + I(maxPlace^2) + I(numGroups^2) + I(walkDistance^2) + I(weaponsAcquired^2))



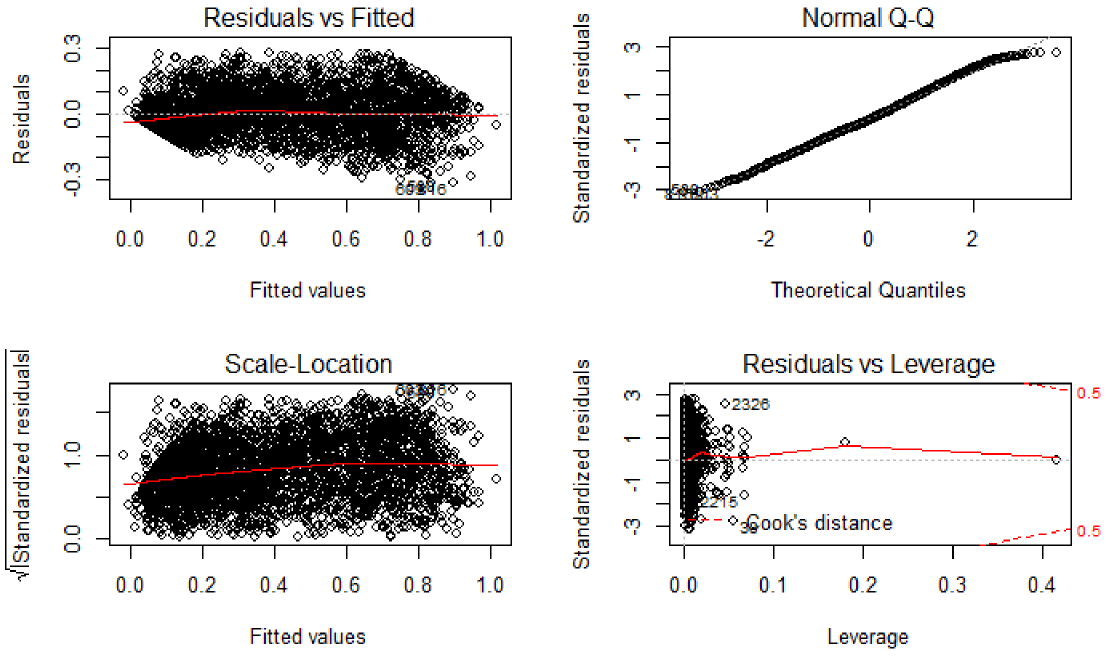
R-squared: 0.858

Adjusted R-squared: 0.857

regsubsets()

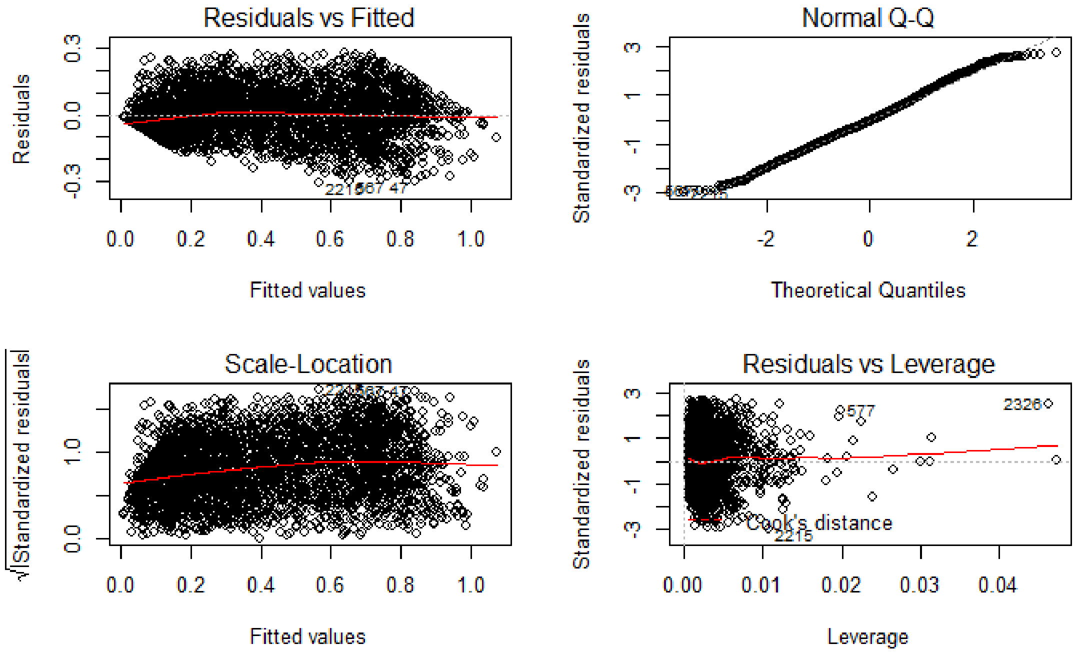
Compare lm.fit5 and lm.fit6

Adjusted R-squared: 0.861



plot(lm.fit5)

Adjusted R-squared: 0.8578

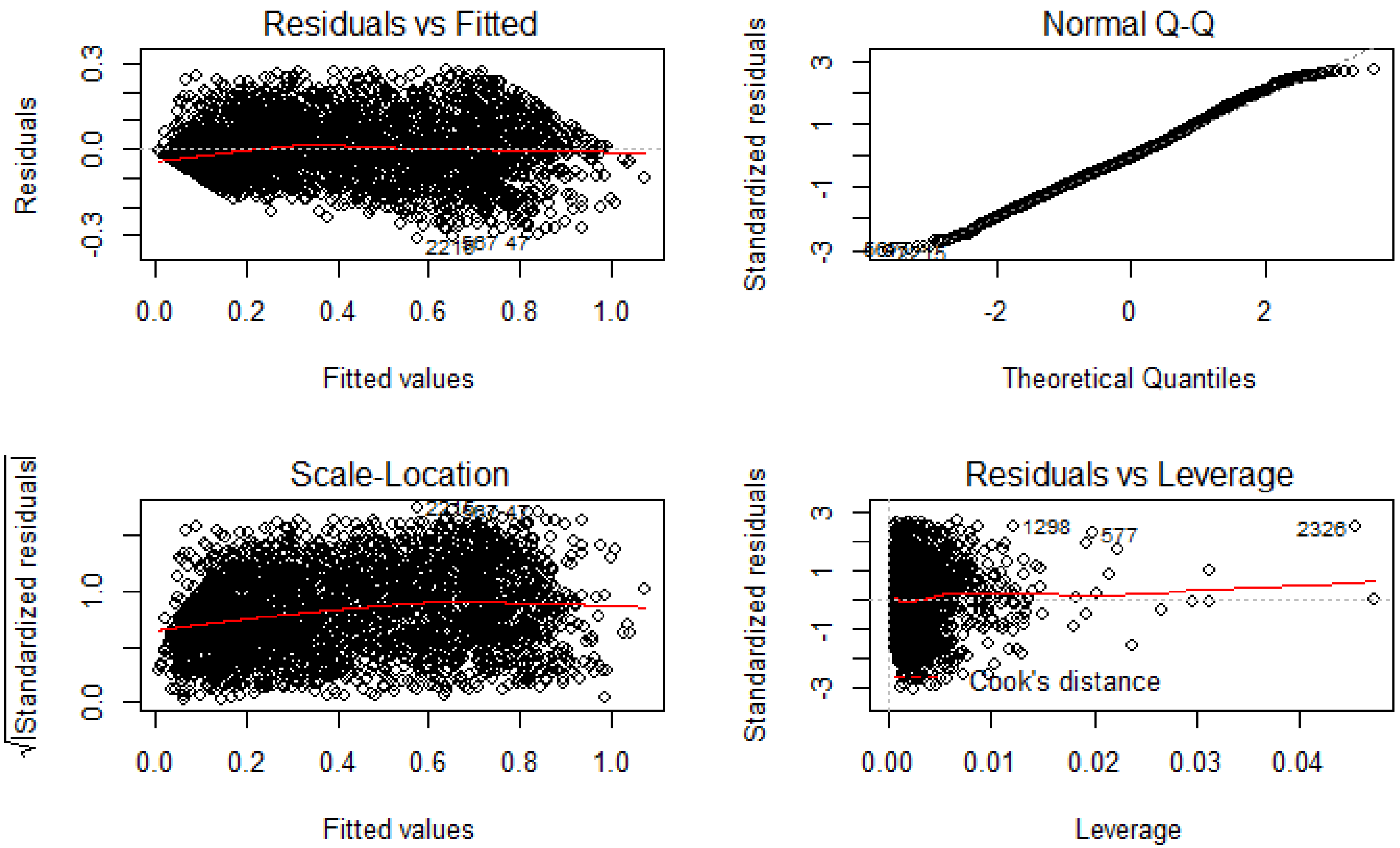


plot(lm.fit6)

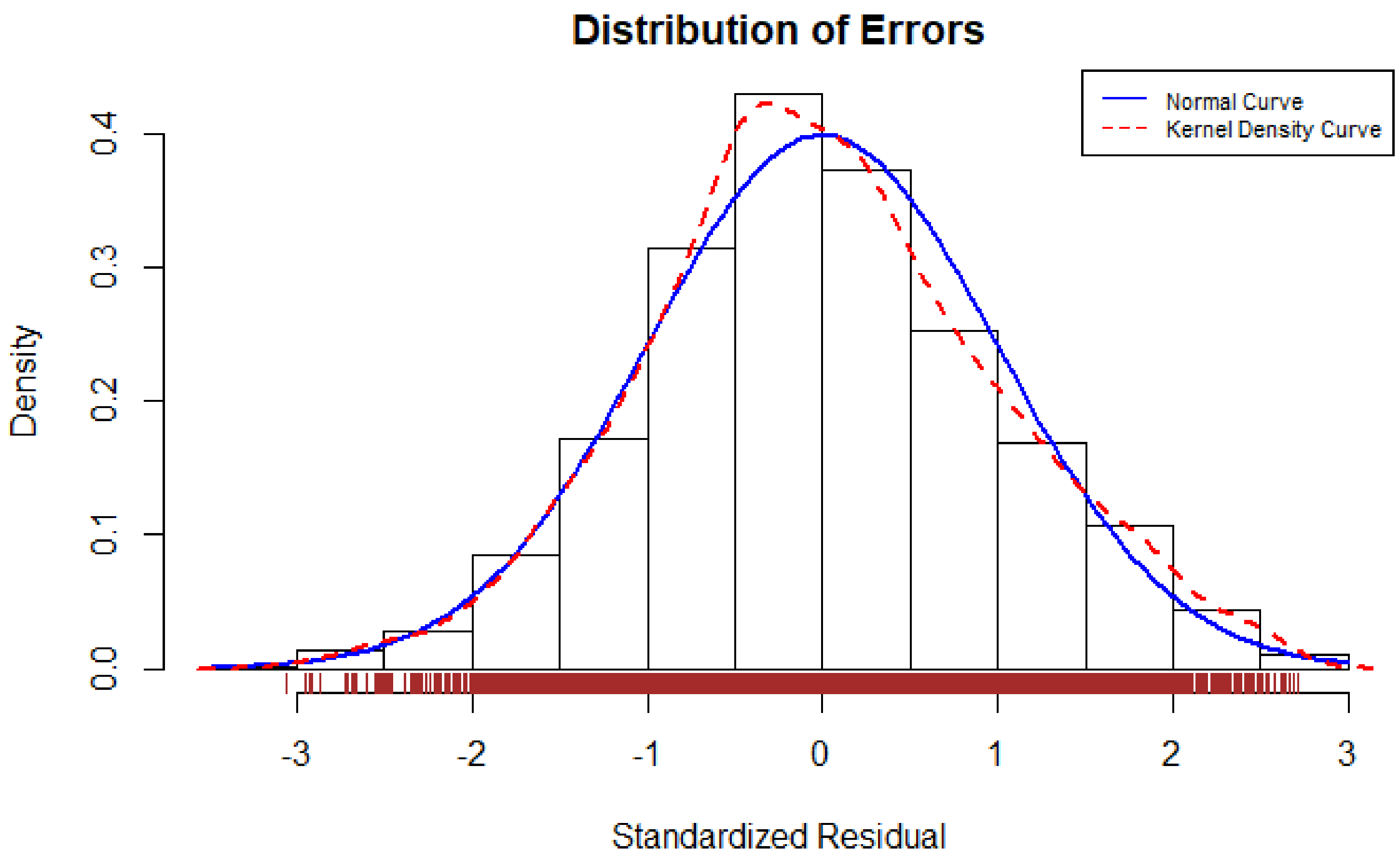
LM: Delete Intercept——lm.fit7

```
lm.fit7=lm(winPlacePerc ~ boosts + damageDealt + walkDistance + weaponsAcquired +
        I(maxPlace^2) + I(numGroups^2) + I(walkDistance^2) + I(weaponsAcquired^2) - 1)
```

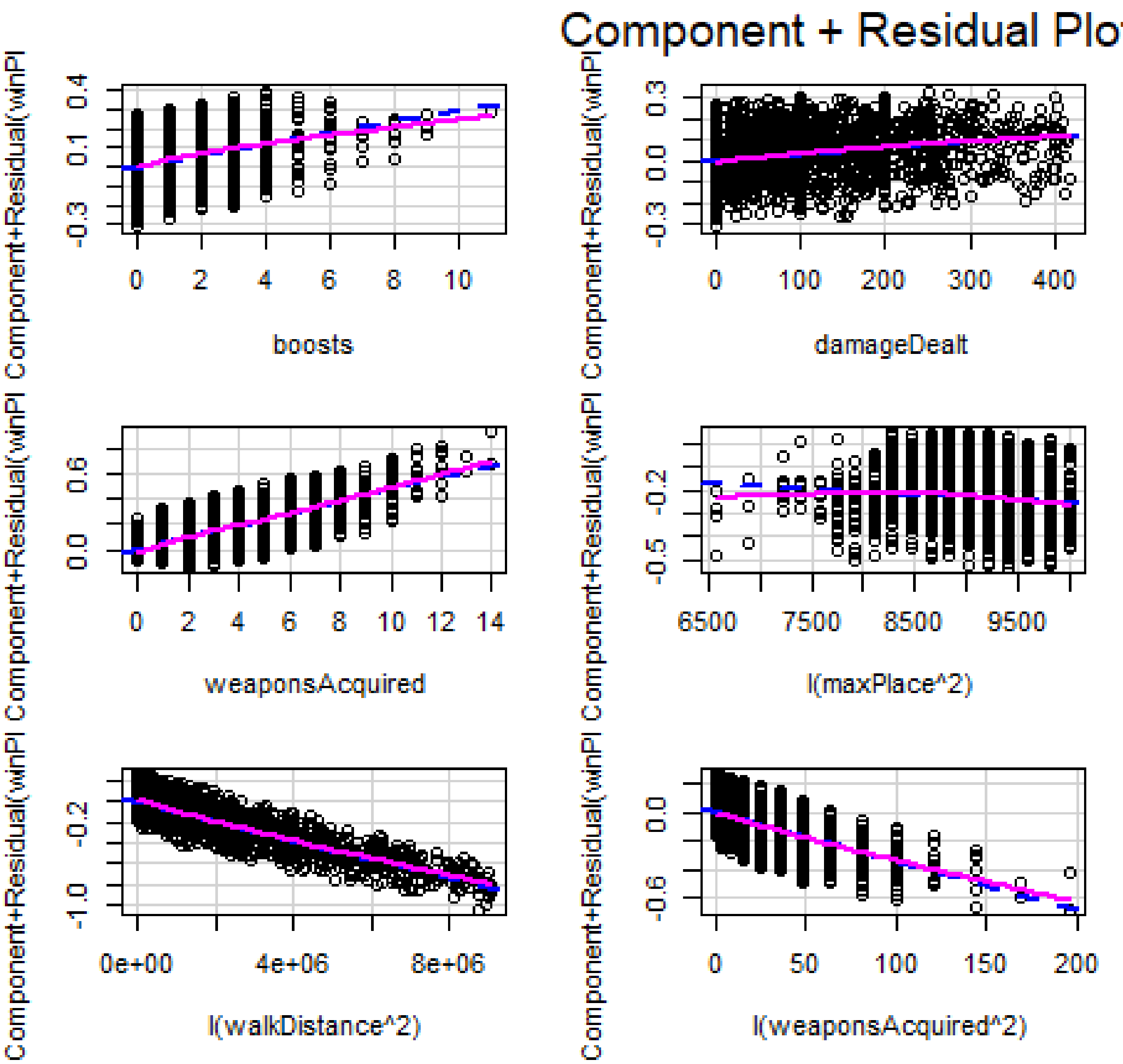
R-squared:	Adjusted R-squared:
0.956	0.956


Residuals vs Fitted
Residuals
Fitted values
0.0 0.2 0.4 0.6 0.8 1.0
-0.3 0.0 0.3
Normal Q-Q
Standardized residuals
Theoretical Quantiles
-2 0 2
-3 -1 1 3
Scale-Location
√|Standardized residuals|
Fitted values
0.0 1.0
Residuals vs Leverage
Standardized residuals
Leverage
0.00 0.01 0.02 0.03 0.04
Cook's distance
1298
577
2326

plot(lm.fit7)



residplot(lm.fit7)



crplot(lm.fit7)

Compare lm.fit6 and lm.fit7

```
Analysis of Variance Table

Model 1: winPlacePerc ~ boosts + damageDealt + walkDistance + weaponsAcquired +
    I(maxPlace^2) + I(numGroups^2) + I(walkDistance^2) + I(weaponsAcquired^2) -
    1
Model 2: winPlacePerc ~ boosts + damageDealt + walkDistance + weaponsAcquired +
    I(maxPlace^2) + I(numGroups^2) + I(walkDistance^2) + I(weaponsAcquired^2)
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1   3366 35.243
2   3365 35.238  1 0.0057646 0.5505 0.4582
```

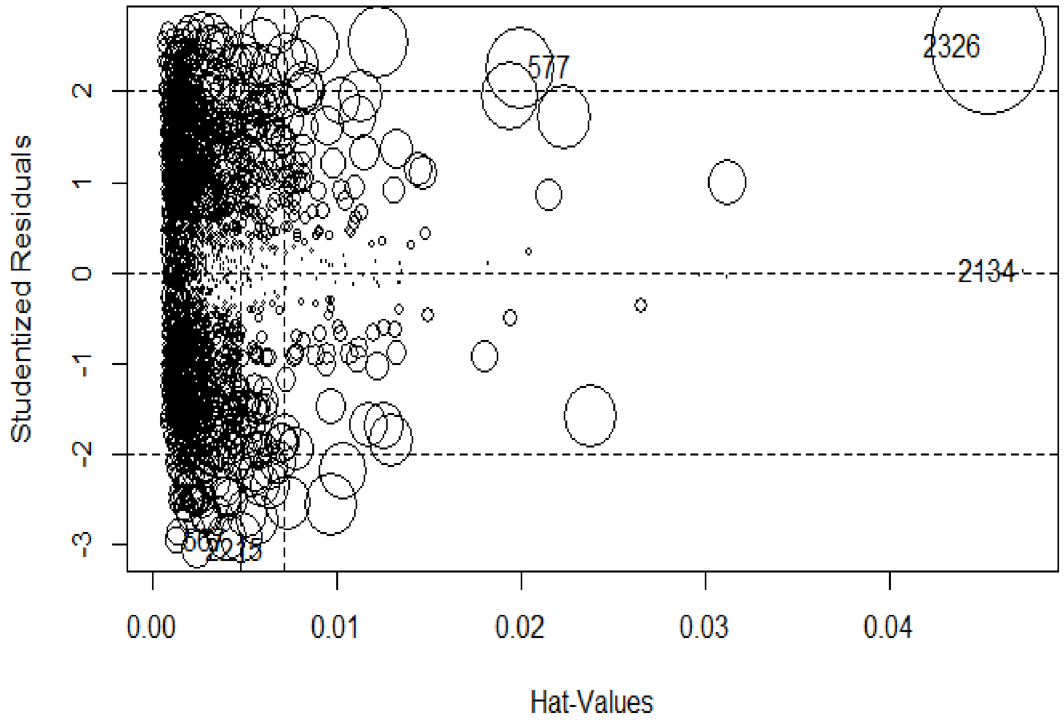
anova(lm.fit7,lm.fit6)

	df <dbl>	AIC <dbl>
lm.fit7	9	-5797.754
lm.fit6	10	-5796.306

AIC(lm.fit7,lm.fit6)

outlierTest(lm.fit7)

```
## No Studentized residuals with Bonferroni p < 0.05
## Largest |rstudent|:
## rstudent unadjusted p-value Bonferroni p
## 2215 -3.063882        0.0022022        NA
```



influencePlot(lm.fit7)

Predict

Validation Data Size: 1000

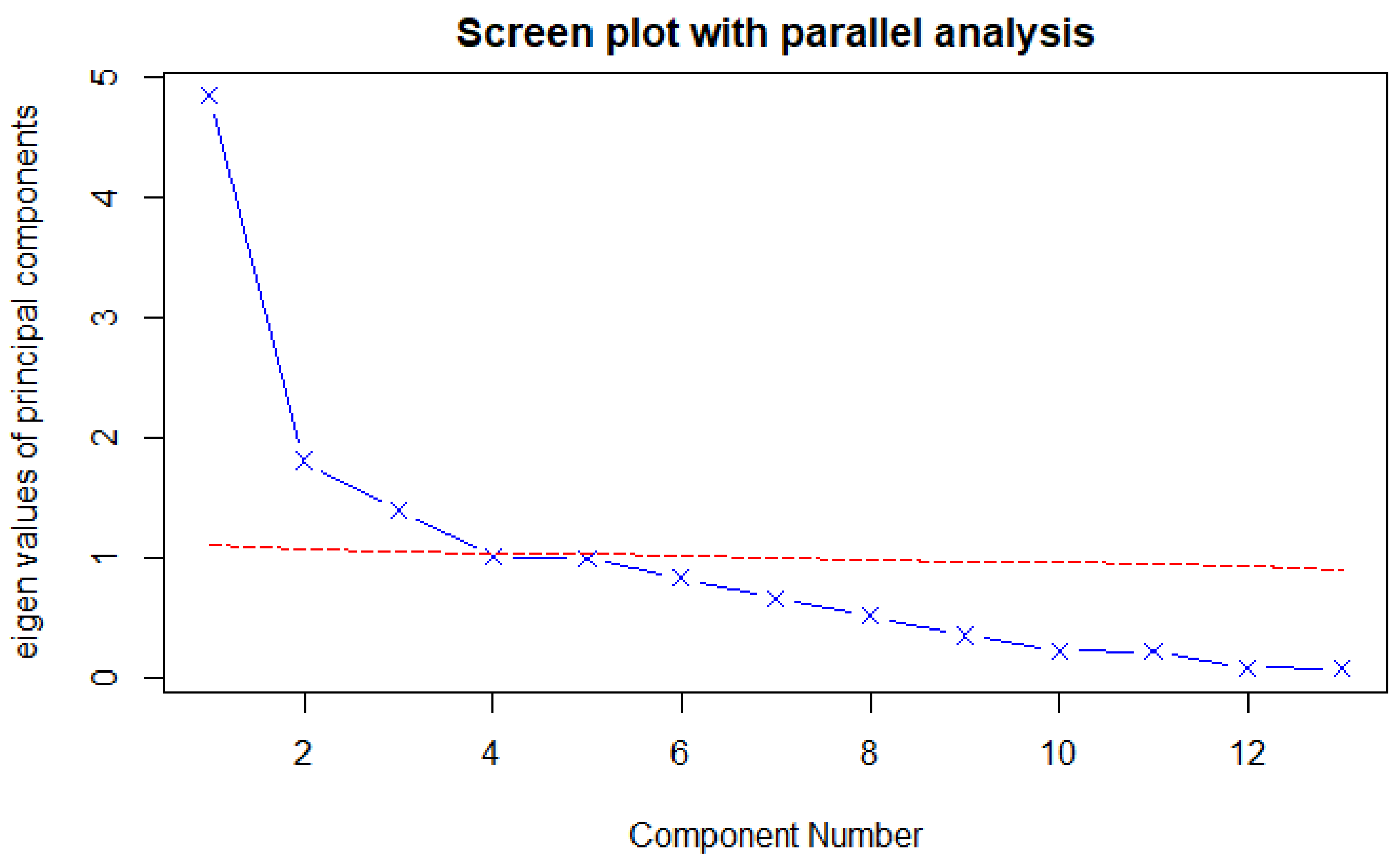
Testing R^2: 0.922

```{r}
# setwd("D:/Collection_NUT/SUSTech_31/R/HW/Project/data")  # delete
test_data <- read_excel("test_data.xlsx")
test_data <- as.data.frame(test_data)
test_pre <- predict(lm.fit7, newdata=test_data[-14])  # predict percentage
mean_test=mean(test_data$winPlacePerc)
SSR_test=sum((test_pre-mean_test)^2)
SST_test=sum((test_data$winPlacePerc-mean_test)^2)
paste("The R^2 of the linear model in the test set is", SSR_test/SST_test )
```

```
[1] "The R^2 of the linear model in the test set is 0.922478755668393"
```

Model Building and Selection

PCA, Lasso, Ridge


Screen plot with parallel analysis
eigen values of principal components
Component Number
0
1
2
3
4
5
2
4
6
8
10
12

```
## Loadings:
##                 Comp.1 Comp.2 Comp.3 Comp.4
## boosts           0.291         0.342       
## damageDealt      0.390        -0.276       
## headshotKills    0.253        -0.297       
## heals            0.233         0.293       
## killPlace       -0.426                     
## kills            0.404        -0.290       
## killStreaks      0.383        -0.291       
## maxPlace              -0.690  0.157       
## numGroups             -0.694  0.127       
## roadKills                           -0.539
## vehicleDestroys                      0.819
## walkDistance     0.291         0.477       
## weaponsAcquired  0.255         0.425 -0.145
```

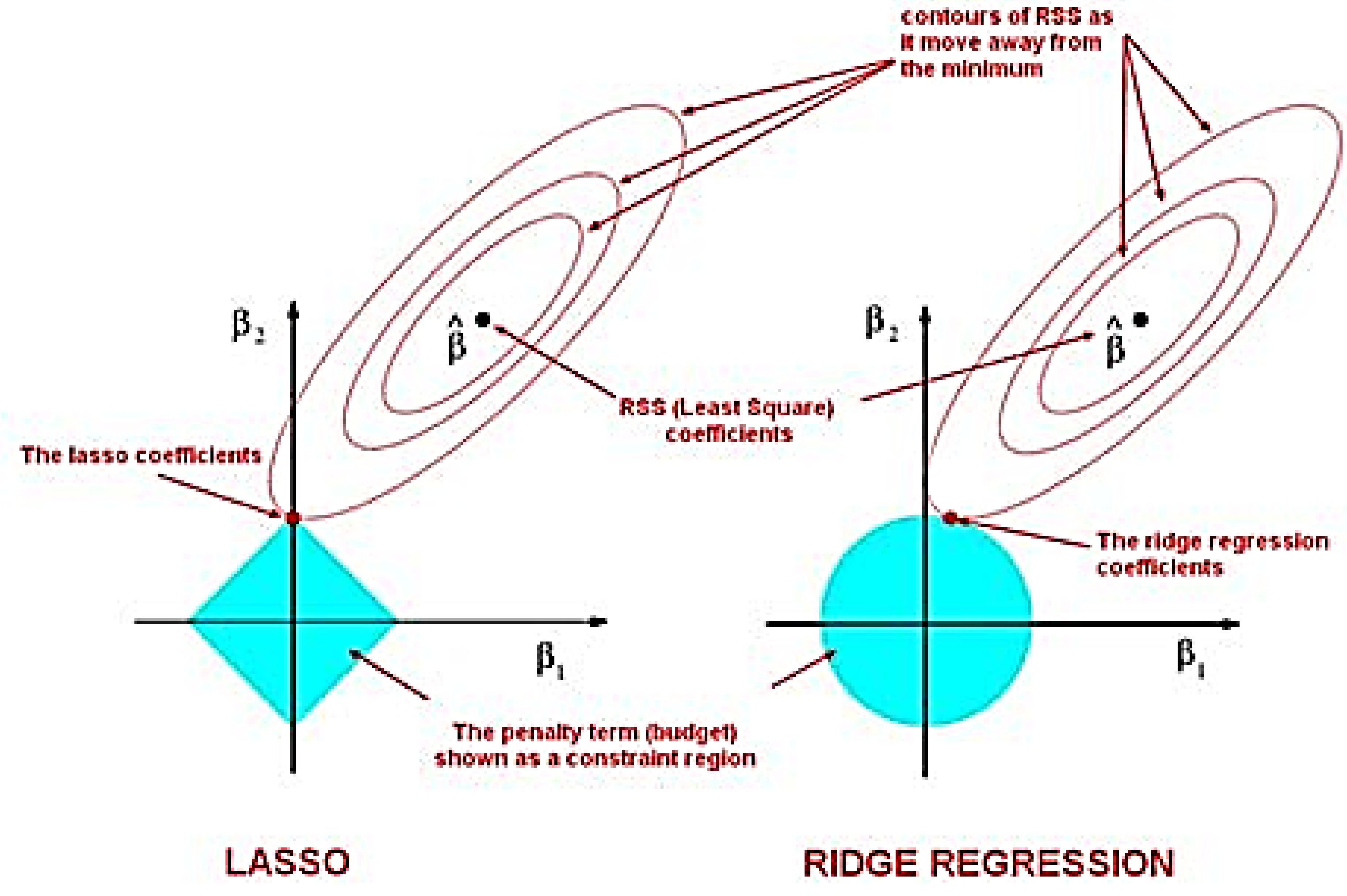
Training set:

```
##
## Call:
## lm(formula = winPlacePerc ~ a1 + a2 + a3 + a4, data = PUBG)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.77400 -0.09247 -0.01185  0.08797  0.45163
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.408760   0.002244 182.129  < 2e-16 ***
## a1           0.090960   0.001020  89.213  < 2e-16 ***
## a2           0.013121   0.001670   7.855 5.32e-15 ***
## a3           0.107304   0.001900  56.482  < 2e-16 ***
## a4          -0.012961   0.002239  -5.787 7.81e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1304 on 3369 degrees of freedom
## Multiple R-squared:  0.7695, Adjusted R-squared:  0.7692
## F-statistic:  2811 on 4 and 3369 DF,  p-value: < 2.2e-16
```

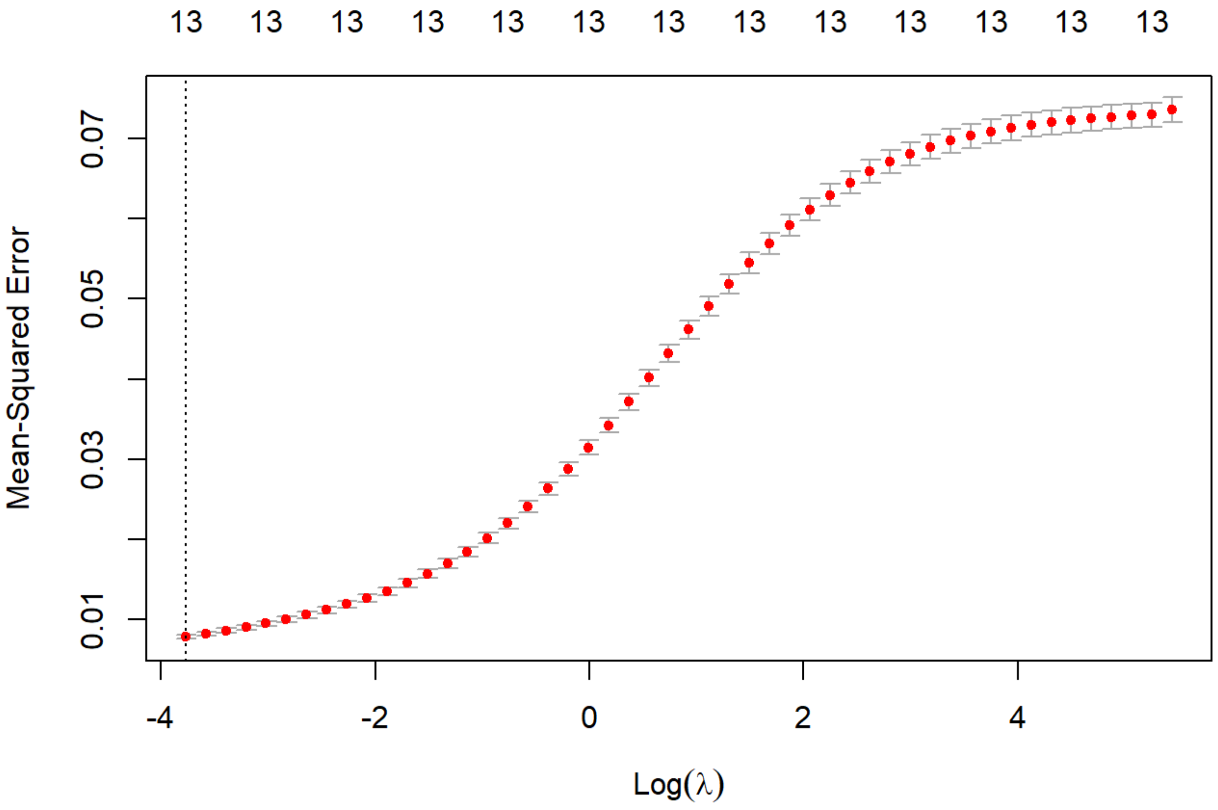
Validation set:

```
##
## Call:
## lm(formula = winPlacePerc ~ a1 + a2 + a3 + a4, data = PUBG_test)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.76573 -0.10878 -0.01456  0.09434  0.57249
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.495385   0.004829 102.588  < 2e-16 ***
## a1           0.102191   0.002142  47.705  < 2e-16 ***
## a2           0.017606   0.003525   4.994 6.98e-07 ***
## a3           0.094864   0.004210  22.531  < 2e-16 ***
## a4          -0.017781   0.004790  -3.713 0.000217 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1527 on 995 degrees of freedom
## Multiple R-squared:  0.7393, Adjusted R-squared:  0.7383
## F-statistic: 705.5 on 4 and 995 DF,  p-value: < 2.2e-16
```

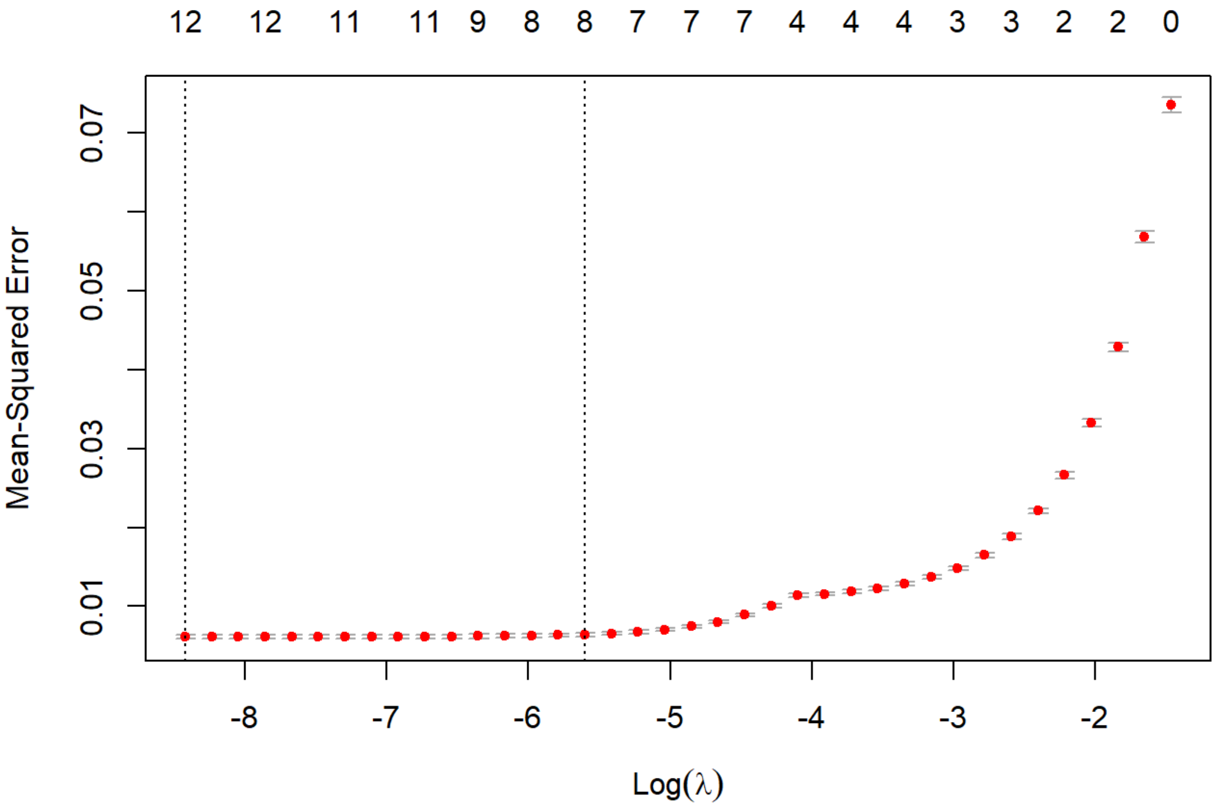
Lasso & Ridge Regression



Ridge Regression:



Lasso Regression:



Model	Training set R^2	Test set R^2
Ridge	0.895	0.774
Lasso	0.918	0.797

Model Building and Selection

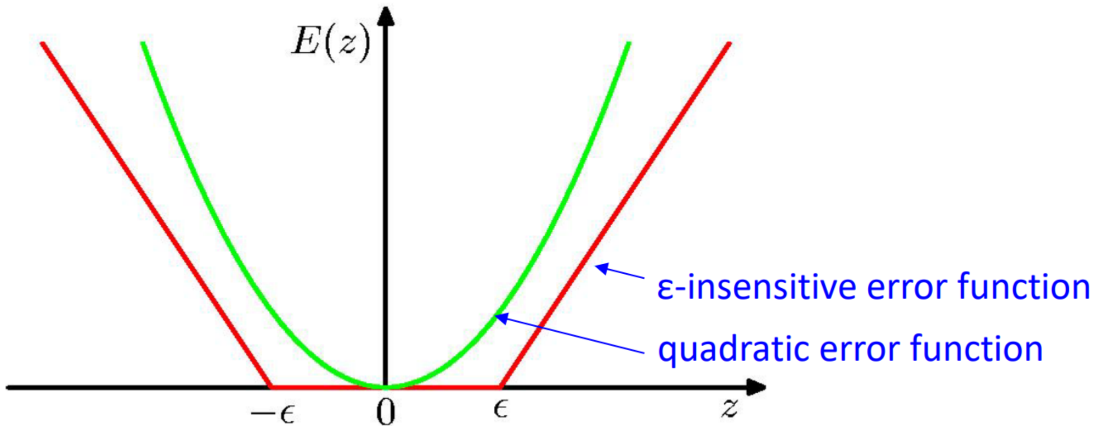
SVM, Neural Network

SVM: Formulation 1

Simple linear regression: minimize $\frac{1}{2}\sum_{n=1}^{N}\boxed{\{y_n - t_n\}^2} + \frac{\lambda}{2}\|\mathbf{w}\|^2$

ε-insensitive error function

$$E_\varepsilon(y(\mathbf{x})-t) = \begin{cases} 0, & \text{if } |y(\mathbf{x})-t| < \varepsilon \\ |y(\mathbf{x})-t| - \varepsilon, & \text{otherwise} \end{cases}$$



SVM: Formulation 2

Minimize

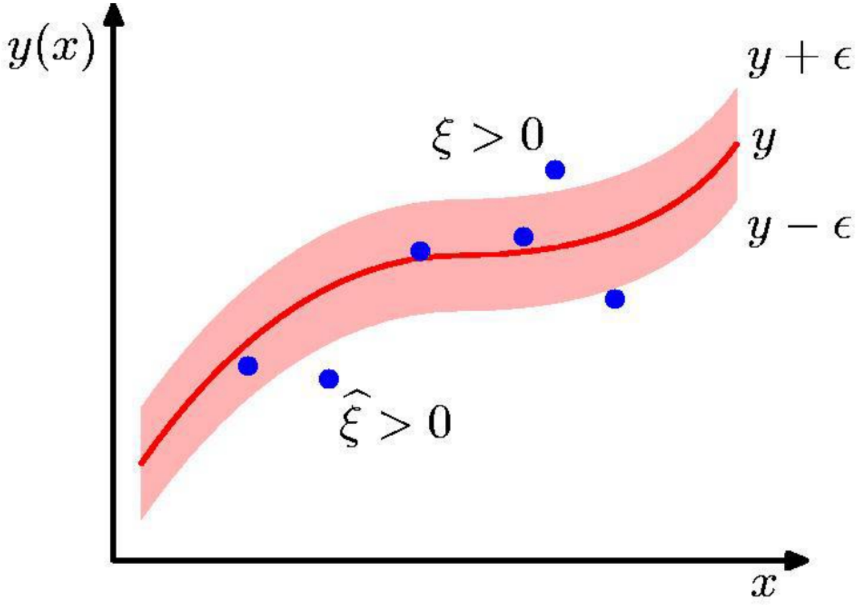
$$C\sum_{n=1}^{N} E_{\varepsilon}\left(y(\mathbf{x}_n)-t_n\right)+\frac{1}{2}\|\mathbf{w}\|^2$$

$$C\sum_{n=1}^{N}\left(\xi_n+\hat{\xi}_n\right)+\frac{1}{2}\|\mathbf{w}\|^2$$

where $t_n \leq y(\mathbf{x}_n)+\varepsilon+\xi_n$

$$t_n \geq y(\mathbf{x}_n)-\varepsilon-\hat{\xi}_n$$

$$\xi_n \geq 0,\ \hat{\xi}_n \geq 0$$



Use package ‘e1071’

`svm`	*Support Vector Machines*

Description

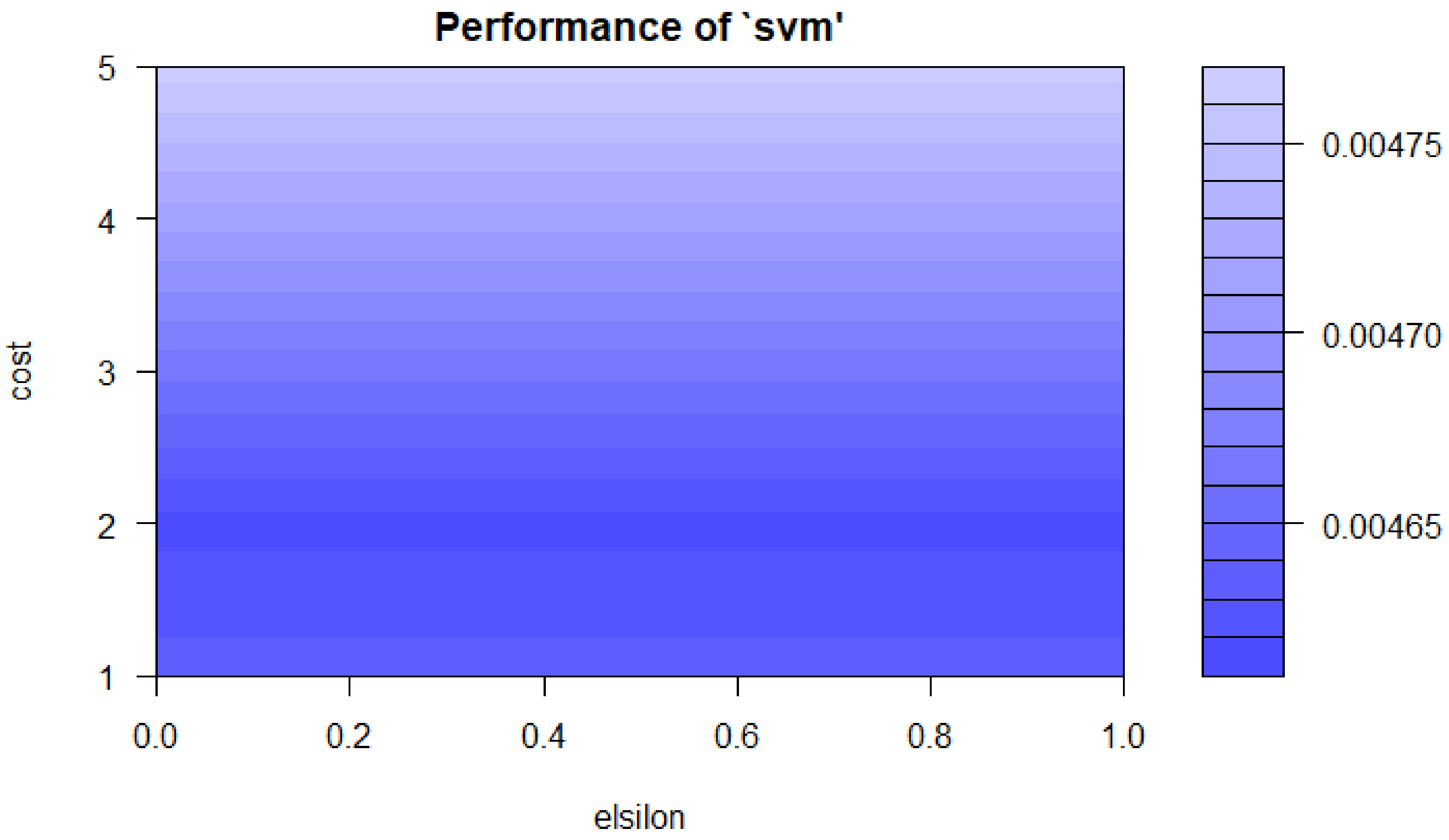
`svm` is used to train a support vector machine. It can be used to carry out general regression and classification (of nu and epsilon-type), as well as density-estimation. A formula interface is provided.

SVM: Implement 2

```
Call:
svm(formula = winPlacePerc ~ ., data = data)


Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.07692308
    epsilon:  0.1


Number of Support Vectors:  1694
```

Parameters of SVM-models usually *must* be tuned to yield sensible results!

SVM: Tuning SVM model by CV



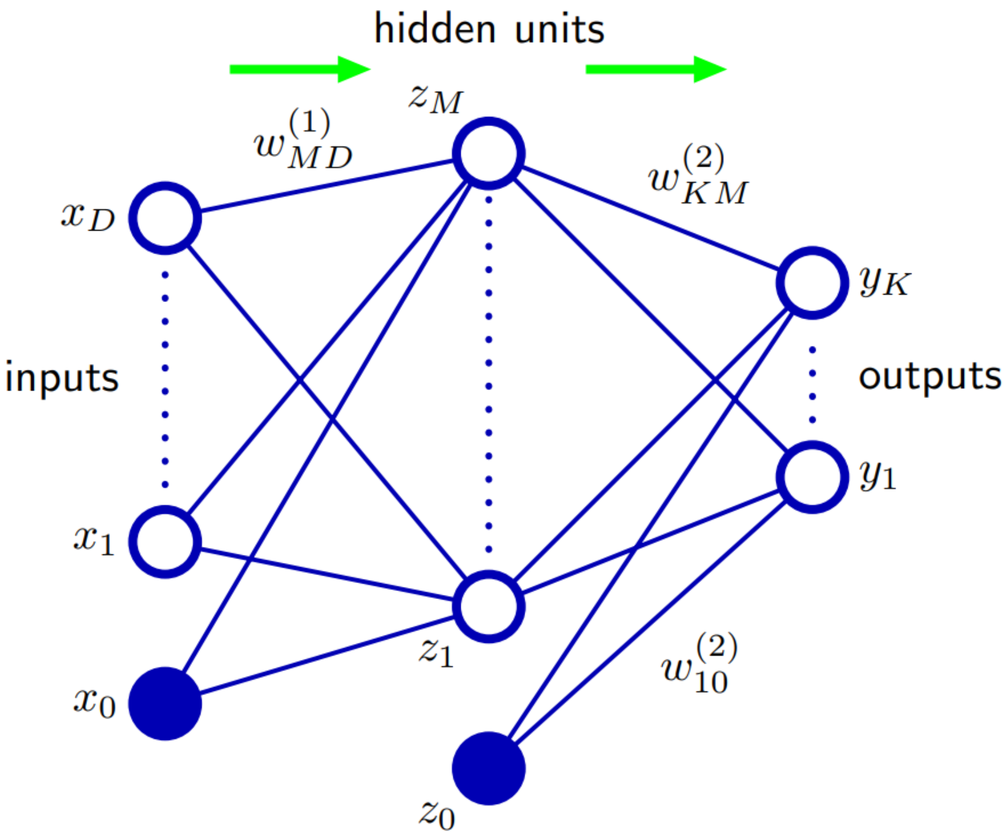
SVM: Finalized Model

```
Call:
svm(formula = winPlacePerc ~ ., data = data, cost = 2, epsilon = 0.1)


Parameters:
   SVM-Type:  eps-regression
 SVM-Kernel:  radial
       cost:  2
      gamma:  0.07692308
    epsilon:  0.1


Number of Support Vectors:  1663
```

Training Data Size: 3374
Training MSE: 0.002791205
Training R^2: 0.9620806

Testing Data Size: 997
Testing MSE: 0.009322744
Testing R^2: 0.8945221

NN: Diagram



$$E(X,\theta) = \frac{1}{2N}\sum_{i=1}^{N}\left(\hat{y}_i - y_i\right)^2,$$

$$\frac{\partial E(X,\theta)}{\partial w_{ij}^k} = \frac{1}{N}\sum_{d=1}^{N}\frac{\partial}{\partial w_{ij}^k}\left(\frac{1}{2}\left(\hat{y}_d - y_d\right)^2\right) = \frac{1}{N}\sum_{d=1}^{N}\frac{\partial E_d}{\partial w_{ij}^k}.$$

We use “neuralnet” package

Training Of Neural Networks

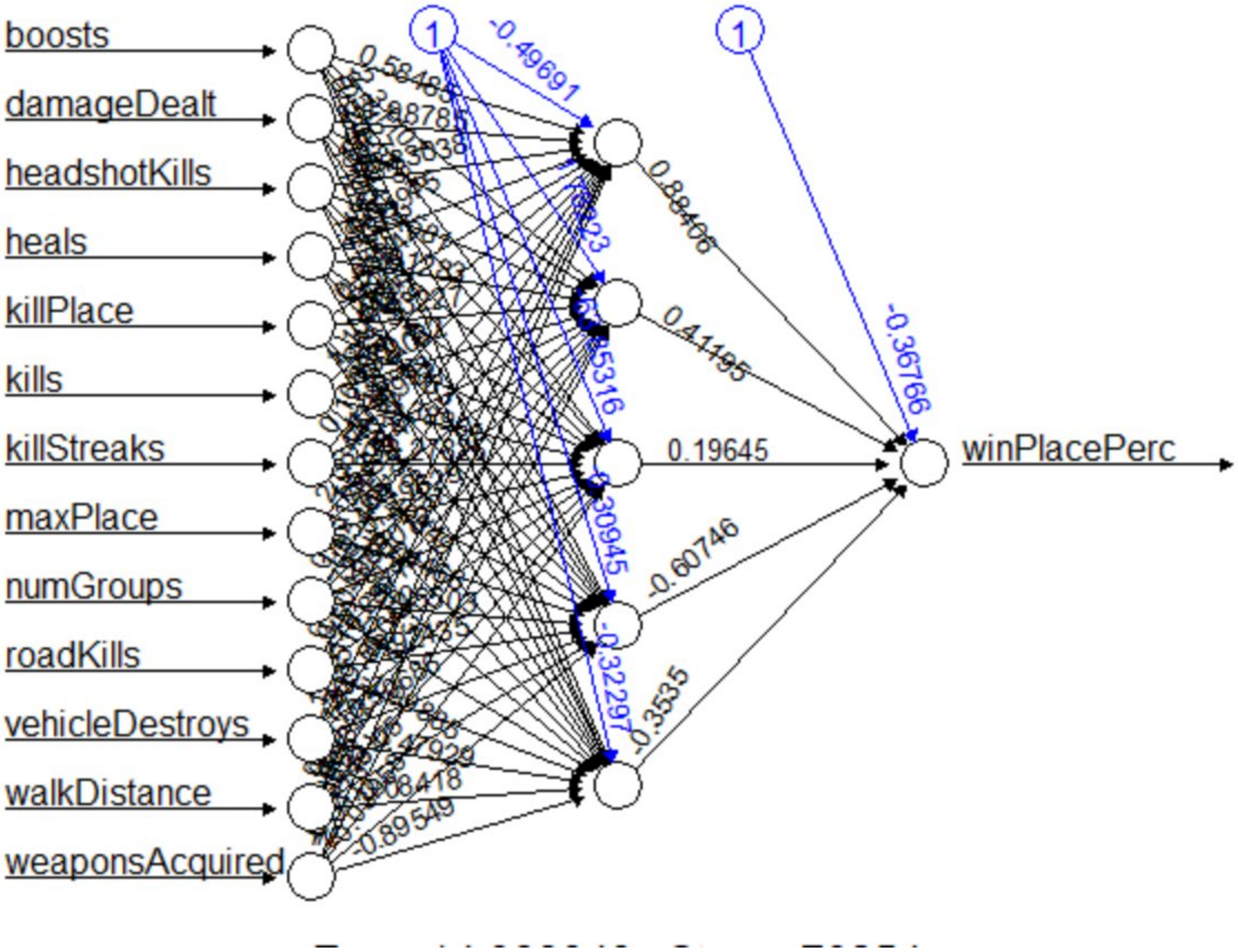
Train neural networks using backpropagation, resilient backpropagation (RPROP) with (Riedmiller, 1994) or without weight backtracking (Riedmiller and Braun, 1993) or the modified globally convergent version (GRPROP) by Anastasiadis et al. (2005). The function allows flexible settings through custom-choice of error and activation function. Furthermore, the calculation of generalized weights (Intrator O. and Intrator N., 1993) is implemented.

Keywords neural

Usage

```
neuralnet(formula, data, hidden = 1, threshold = 0.01,
  stepmax = 1e+05, rep = 1, startweights = NULL,
  learningrate.limit = NULL, learningrate.factor = list(minus = 0.5,
  plus = 1.2), learningrate = NULL, lifesign = "none",
  lifesign.step = 1000, algorithm = "rprop+", err.fct = "sse",
  act.fct = "logistic", linear.output = TRUE, exclude = NULL,
  constant.weights = NULL, likelihood = FALSE)
```

NN: Finalized Model



Training Data Size: 3374
Training R^2: 0.8867439

Testing Data Size: 997
Testing R^2: 0.861229

Model Building and Selection

KNN, Regression Learner

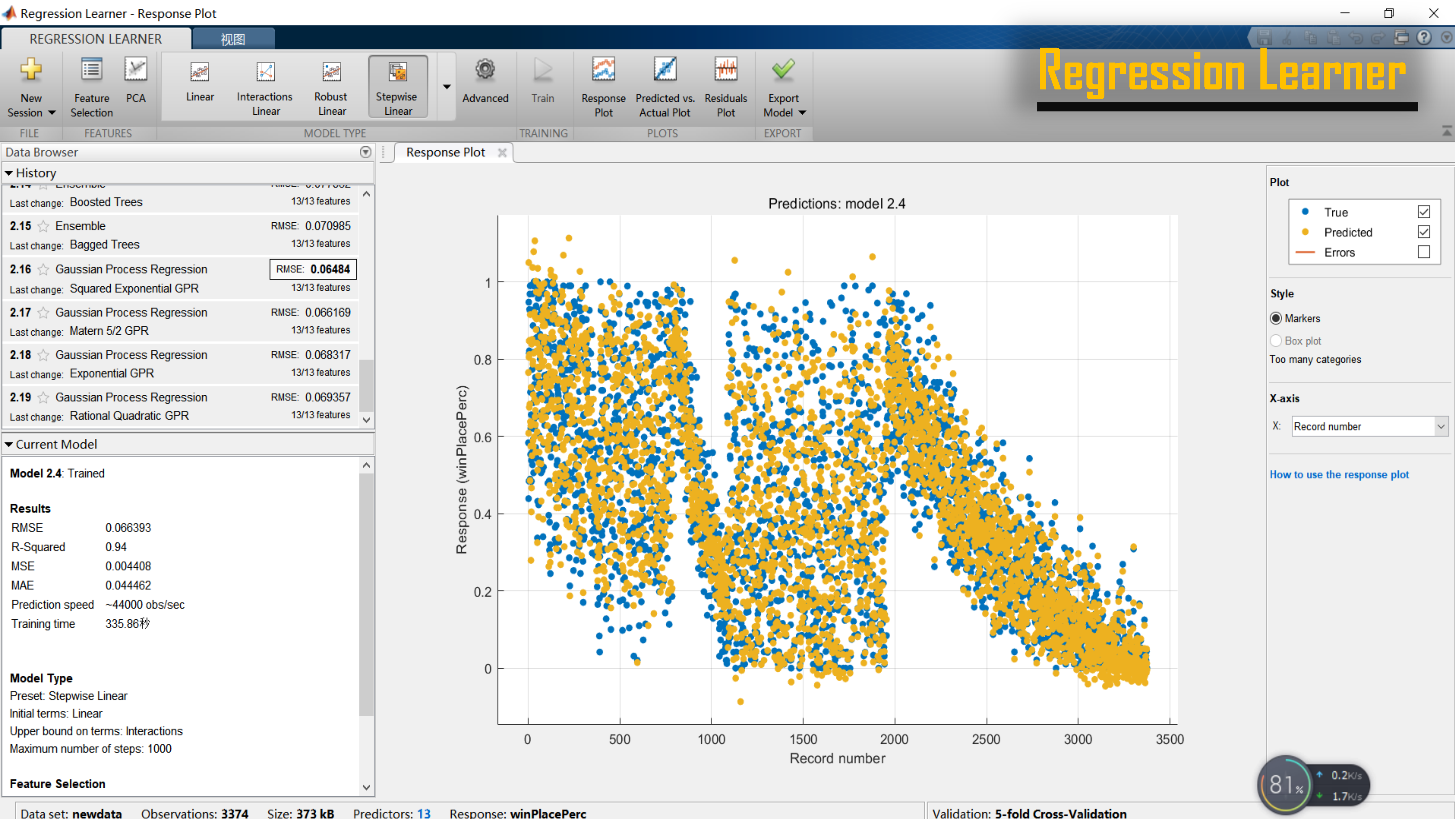
KNN

Scale → Euclidean Distance → Order → Prediction

N	10	20	30	40	50
R^2	0.821	0.792	0.772	0.756	0.743
N	60	70	80	90	100
R^2	0.731	0.721	0.711	0.702	0.694

```
SSR_test_KNN=sum((predict_test-mean_test)^2)
SST_test_KNN=sum((data_test[,14]-mean_test)^2)
paste("The R^2 of the KNN model in the test set is", SSR_test_KNN/SST_test_KNN )
```

```
[1] "The R^2 of the KNN model in the test set is 0.847805845557967"
```


Regression Learner - Response Plot
REGRESSION LEARNER
视图
New Session
Feature Selection
PCA
Linear
Interactions Linear
Robust Linear
Stepwise Linear
Advanced
Train
Response Plot
Predicted vs. Actual Plot
Residuals Plot
Export Model
Regression Learner
FILE
FEATURES
MODEL TYPE
TRAINING
PLOTS
EXPORT
Data Browser
History
Last change: Boosted Trees 13/13 features
2.15 Ensemble RMSE: 0.070985
Last change: Bagged Trees 13/13 features
2.16 Gaussian Process Regression RMSE: 0.06484
Last change: Squared Exponential GPR 13/13 features
2.17 Gaussian Process Regression RMSE: 0.066169
Last change: Matern 5/2 GPR 13/13 features
2.18 Gaussian Process Regression RMSE: 0.068317
Last change: Exponential GPR 13/13 features
2.19 Gaussian Process Regression RMSE: 0.069357
Last change: Rational Quadratic GPR 13/13 features
Current Model
Model 2.4: Trained
Results
RMSE 0.066393
R-Squared 0.94
MSE 0.004408
MAE 0.044462
Prediction speed ~44000 obs/sec
Training time 335.86秒
Model Type
Preset: Stepwise Linear
Initial terms: Linear
Upper bound on terms: Interactions
Maximum number of steps: 1000
Feature Selection
Response Plot
Predictions: model 2.4
Response (winPlacePerc)
Record number
0
0.2
0.4
0.6
0.8
1
0
500
1000
1500
2000
2500
3000
3500
Plot
True
Predicted
Errors
Style
Markers
Box plot
Too many categories
X-axis
X: Record number
How to use the response plot
Data set: newdata Observations: 3374 Size: 373 kB Predictors: 13 Response: winPlacePerc
Validation: 5-fold Cross-Validation

Model Comparison

Model Comparison

Model	LS	PCA	Lasso	Ridge	KNN	SVM	Neural Network
R^2 (training)	0.956	0.769	0.918	0.895	0.821	0.962	0.887
R^2 (testing)	0.922	0.739	0.797	0.774	0.848	0.894	0.861

What's the best strategy to win in PUBG?

lm.fit7=lm(winPlacePerc ~ boosts + damageDealt + walkDistance + weaponsAcquired + I(maxPlace^2) + I(numGroups^2) + I(walkDistance^2) + I(weaponsAcquired^2) - 1)

Run More!

Keep Calm !

Obtain Weapons !


蟹蟹