Predicting correct weight lifting

Hermann Hess

November 21, 2015

How the model was built

The model's main objective is to predict, according to data from accelerometers on the belt, forearm, arm, and dumbbell of 6 participants, whether the participants doing the weight-lifting exercises are doing it correctly (coded as classe A) or incorrectly (coded as classe B,C,D or E). [*]

The data were taken from http://groupware.les.inf.puc-rio.br/har (http://groupware.les.inf.puc-rio.br/har) and correspond to 19622 observations on 160 variables for the training set and 20 observations on the same number of variables for the testing set.

It was first necessary to compare the two data sets to make sure they both had the same columns, especially the NA columns. This was true except for the last column (response variable "classe") in the training set and the corresponding last (case id) column in the testing set. Due to the extensive size of the data set (19622 observations on each variable) and lack of domain-specific knowledge, corrections for outliers was not done for either training nor testing data.

Downloading the data sets and exploratory analysis were carried out running the following code chunk:

```
# Read training set
#
fileURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-training.csv"
download.file(fileURL,destfile = "./train.csv")
downloaded <- date()
training <- read.csv("./train.csv")
head(training)
str(training) # 'data.frame':   19622 obs. of  160 variables
# Read testing
fileURL <- "https://d396qusza40orc.cloudfront.net/predmachlearn/pml-testing.csv"
download.file(fileURL,destfile = "./test.csv")
downloaded <- date()
testing <- read.csv("./test.csv")
head(testing)
str(testing)  # 'data.frame':   20 obs. of  160 variables
#
names(testing) == names(training)  # FALSE for last (response) column
# To compare if NA columns the same in testing and training
length_testing_NAs <- vector(length=length(testing))
for (i in 1:length(testing)){
    length_testing_NAs[i] = sum(complete.cases(testing[,i]))
}
#
length_training_NAs <- vector(length=length(training))
for (i in 1:length(training)){
    length_training_NAs[i] = sum(complete.cases(training[,i]))
}
cbind(length_testing_NAs,length_training_NAs) # All complete testing -> training
```

The next step was to get rid of incomplete cases (columns with all or almost all NAs) and the first 7 columns of both data sets, which include unnecessary features such as *name* and *timestamp*. This follows the key idea 'to predict X use data related to X' and to focus only on important features.

In addition, the *classe* response variable was converted to numeric, and it should also be noted that Zero covariates (`nearZeroVar(training3,saveMetrics = TRUE)`) are no problem in this data set.

```
completes <- which(length_testing_NAs == 20, arr.ind = TRUE, useNames = TRUE)
testing1  <-  testing[,completes]    # Now 60 variables
training1 <- training[,completes]    # Also 60 variables
# Eliminate variables that don't seem to be good features (1-7)
training2 <- training1[,-(1:7)]      # 53 variables
testing2  <-  testing1[,-(1:7)]      # 53 variables
training2$classe <- as.numeric(as.character(training1$classe))
```

```
## Warning: NAs introduced by coercion
```

```
training2$classe <- as.numeric(training1$classe)
```

It is also important at this point to check for highly correlated regressors in the dataset, which was explored

estimating the correlation matrix and focusing on correlations larger than 0.80; in conjunction with the VIF stepwise procedure. This allowed for the elimination of 21 more variables from the model's dataset, for a final total of 31 covariates.

```
co <- abs(cor(training2[,-53]))
diag(co) <- 0
wh <- which(co > 0.80, arr.ind=TRUE)
elim <- c(1,2,3,4,9,10,11,19,21,22,25,26,33,34,35,36,37,39,46,47,48)
training3 <- training2[,-elim]
testing3  <-  testing2[,-elim]
```

Finally, the model was set up using the *caret* package and setting the seed to an arbitrary integer for reproducibility. A generalized linear regression was chosen as estimation technique, using all available predictors.

```
library("caret", lib.loc="~/R/win-library/3.1")
```

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

```
set.seed(201115)
mod3 <- train(classe ~., data = training3, method = "glm")
summary(mod3) # Almost all coeff significant
```

The results of this model appear quite good in that almost all coefficients (except *gyros_belt_z* , *gyros_arm_z* and *gyros_forearm_y*) are significant.

```
library(pander)
panderOptions('table.split.table',108)  # , Inf
tabl <- summary(mod3)[[11]]
pander(tabl, caption="Summary of mod3 coefficients", style = "rmarkdown")
```

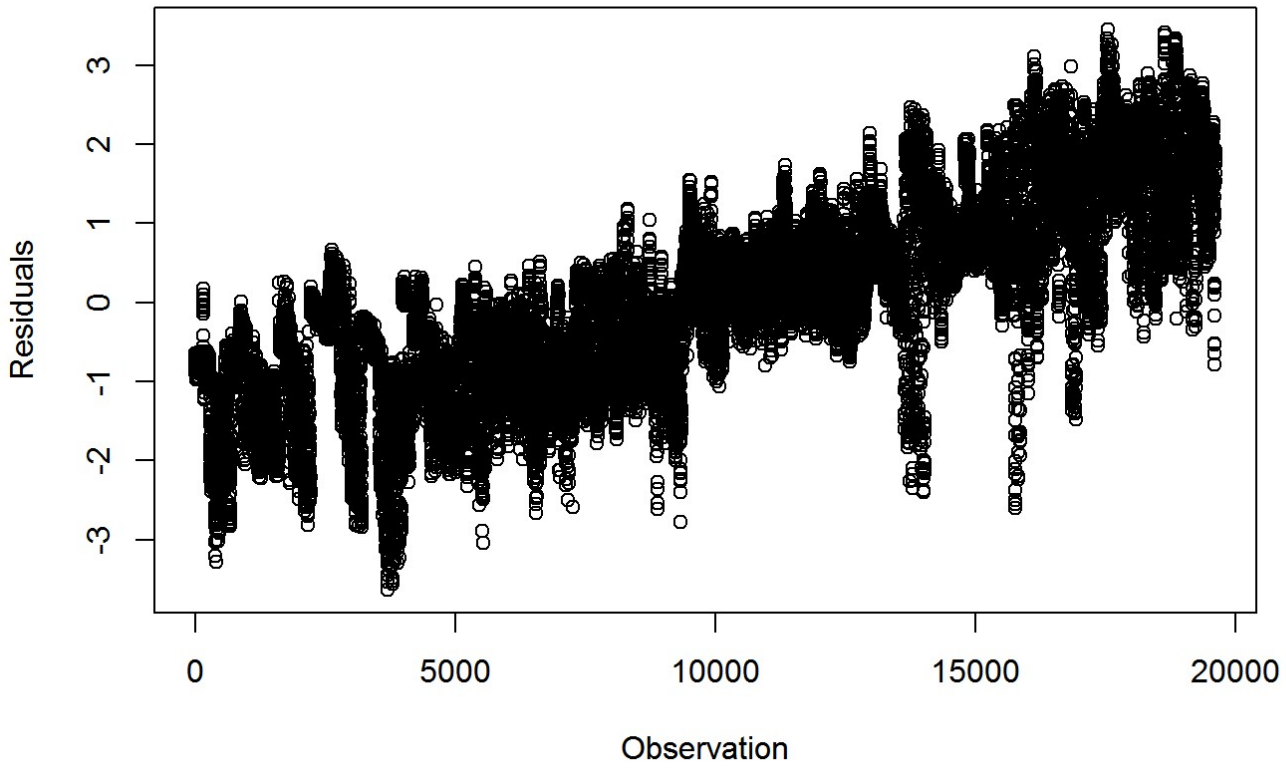
Summary of mod3 coefficients

	Estimate	Std. Error	t value	Pr(>
(Intercept)	12.98	0.3248	39.95	0
gyros_belt_x	0.4976	0.07707	6.457	1.095e-10
gyros_belt_y	0.3044	0.1773	1.717	0.08602
gyros_belt_z	-0.02426	0.05016	-0.4837	0.6286
accel_belt_x	-0.004982	0.0005627	-8.855	9.089e-19
magnet_belt_y	-0.019	0.0004342	-43.76	0
magnet_belt_z	0.002432	0.0002443	9.956	2.684e-23
roll_arm	0.0006779	0.0001586	4.274	1.926e-05

	Estimate	Std. Error	t value	Pr(>
pitch_arm	-0.006315	0.0003747	-16.86	2.668e-63
yaw_arm	0.0003099	0.0001381	2.244	0.02482
total_accel_arm	0.007774	0.001222	6.362	2.035e-10
gyros_arm_x	0.03466	0.004812	7.204	6.056e-13
gyros_arm_z	-0.03551	0.0221	-1.607	0.108
accel_arm_z	0.001511	0.0001052	14.36	1.657e-46
magnet_arm_x	0.0005575	2.725e-05	20.46	4.801e-92
roll_dumbbell	0.001221	0.0001753	6.965	3.396e-12
pitch_dumbbell	-0.002469	0.0003476	-7.103	1.261e-12
yaw_dumbbell	-0.002376	0.0001819	-13.06	8.013e-39
total_accel_dumbbell	0.01693	0.001278	13.25	6.458e-40
gyros_dumbbell_x	0.04665	0.01021	4.568	4.961e-06
gyros_dumbbell_y	0.08735	0.02067	4.225	2.397e-05
magnet_dumbbell_y	0.0002663	5.235e-05	5.086	3.692e-07
roll_forearm	0.001579	9.98e-05	15.82	4.75e-56
pitch_forearm	0.01799	0.0004771	37.7	2.269e-300
yaw_forearm	-0.0002189	0.000109	-2.007	0.04475
total_accel_forearm	0.03404	0.001057	32.2	9.63e-222
gyros_forearm_x	-0.1071	0.02234	-4.796	1.631e-06
gyros_forearm_y	-0.006908	0.004979	-1.387	0.1653
accel_forearm_z	-0.0008467	9.77e-05	-8.666	4.808e-18
magnet_forearm_x	0.0001594	3.575e-05	4.46	8.257e-06
magnet_forearm_y	-7.508e-05	2.304e-05	-3.259	0.001119
magnet_forearm_z	-0.0002158	3.918e-05	-5.507	3.687e-08

Analysis of residuals is in general not altogether unfavorable to the model (see the Appendix), but a downside is that the plot of residuals vs observations shows a clear upward trend. This result leads to possible future improvements of the model.

Figure 1 - Residuals vs fitted from mod3



Cross validation

Cross-validation is basically a way of measuring the predictive performance of a statistical model. Cross-validation is a general name for all techniques that use a test set different than the train set. By allowing cases in the testing set different from those in the training set, CV inherently offers protection against overfitting. In this case of linear regression, cross-validation consists basically of predictive performance, which will be discussed below.

Prediction was implemented according to the following code sequence, where numerical output was converted back to the original factor levels by rounding the numerical output to the nearest factor level (“A” is 1, “B” is 2, etc.):

```
mod3_pred <- predict(mod3,newdata = testing3)
nums <- round(mod3_pred)
letters <- unique(training1$classe)
results <- vector(length=20)
for (i in 1:length(nums)){
    results[i] <- as.character(letters[nums[i]])
}
```

Expected out of sample error

The predictive accuracy of a model can be measured by the mean squared error (or RMSE) on the test set,

although as will be briefly commented below, even if the response was converted to numeric the categorical nature of *classe* also deserves to look at out of sample error from another perspective. This latter measure will generally be larger than the MSE on the training set because the test data were not used for estimation. The results for this model are the following:

Generalized Linear Model

19622 samples 49 predictor

No pre-processing *Resampling: Bootstrapped (25 reps)* Summary of sample sizes: 19622, 19622, 19622, 19622, 19622, 19622, …

```
library(pander)
panderOptions('table.split.table',Inf)  # , Inf
pander(mod3[[4]][2:5], caption="Summary of mod3", style = "rmarkdown")
```

Summary of mod3

RMSE	Rsquared	RMSESD	RsquaredSD
1.202	0.3352	0.007416	0.007743

Justifying choices

The risk of correlation between variables was initially deemed to be small in principle since these are measurements on different and independent movements, but that assumption changed with the analysis and the regressor set was downsized considerably.

As to the choice of model, in the original paper the authors justify using Random Forests with Bagging. Here a linear regression model was chosen mainly because of **interpretability** - if the objective is to correct wrong ways of doing weights it is important to be able to interpret the contributing factors; and linear regression makes that comparatively easy. The author of this report does not have substantive knowledge of the field so as to interpret the magnitude and signs of the estimated model; but for the end users those results ought to be a high priority of a model's output.

Prediction of 20 different test cases

Predicted classes were the following:

"C" "B" "B" "B" "B" "C" "D" "D" "A" "B" "C" "C" "C" "A" "D" "B" "B" "C" "C" "C"

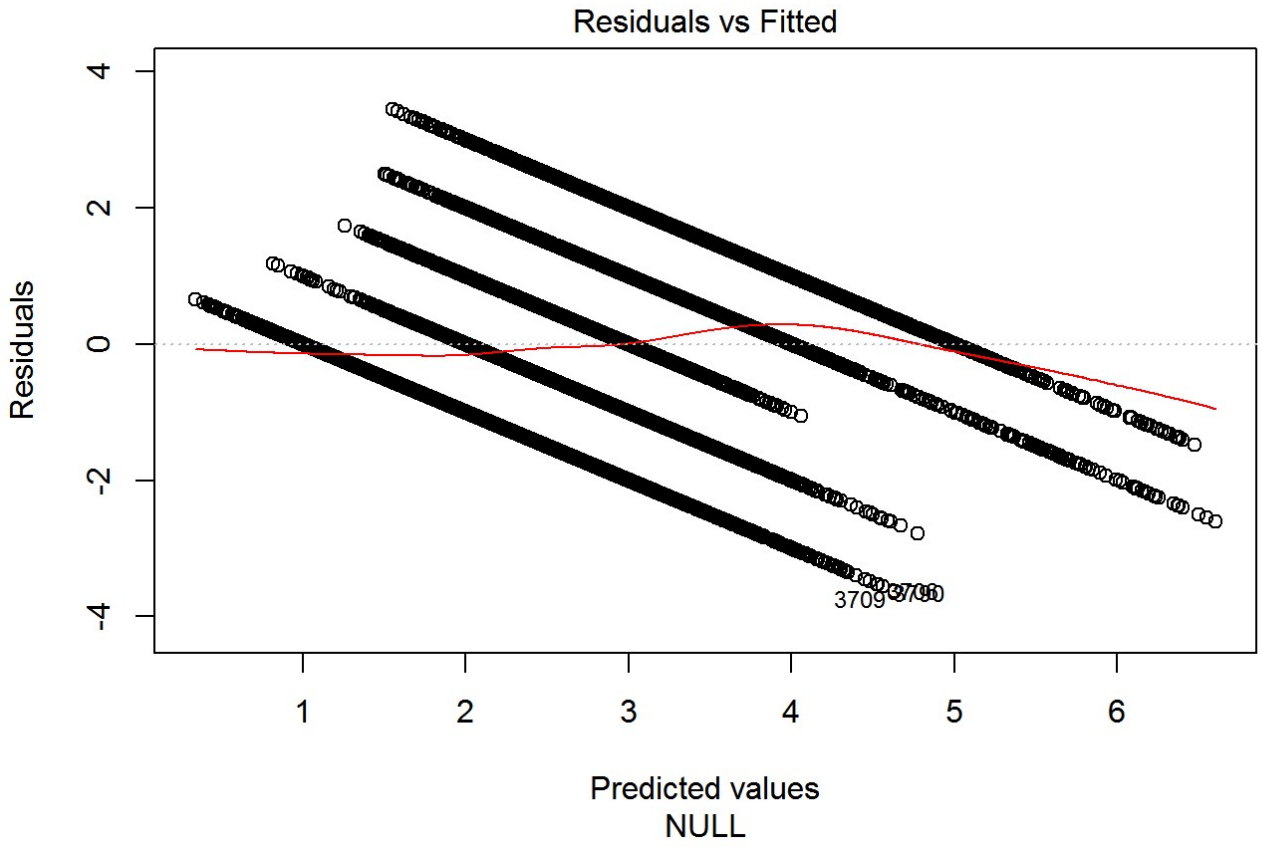
Going back to a previous comment on the categorical nature of the outcome, it is important to point out that the *accuracy* of this model (fraction of correct predictions) is not very good: only 8 out of 20 (40 percent). This result is probably induced by nonlinearity and/or overfitting (still too many variables) in the training set, and again points to possible future improvements in the model; or to simply changing the approach back to Random Forests with Bagging (as suggested by the authors) and using tools such as PCA to reduce the number of potential predictors in the model.

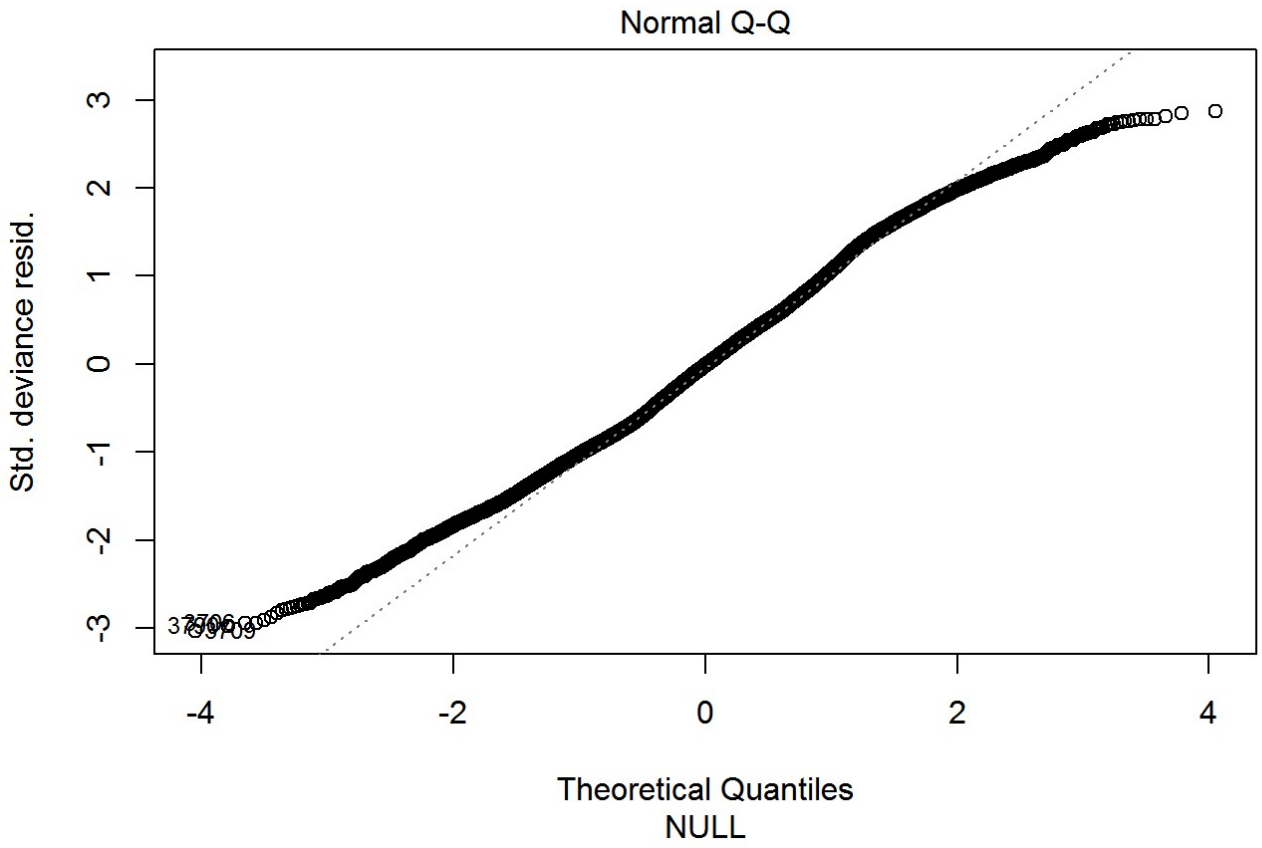
Reference

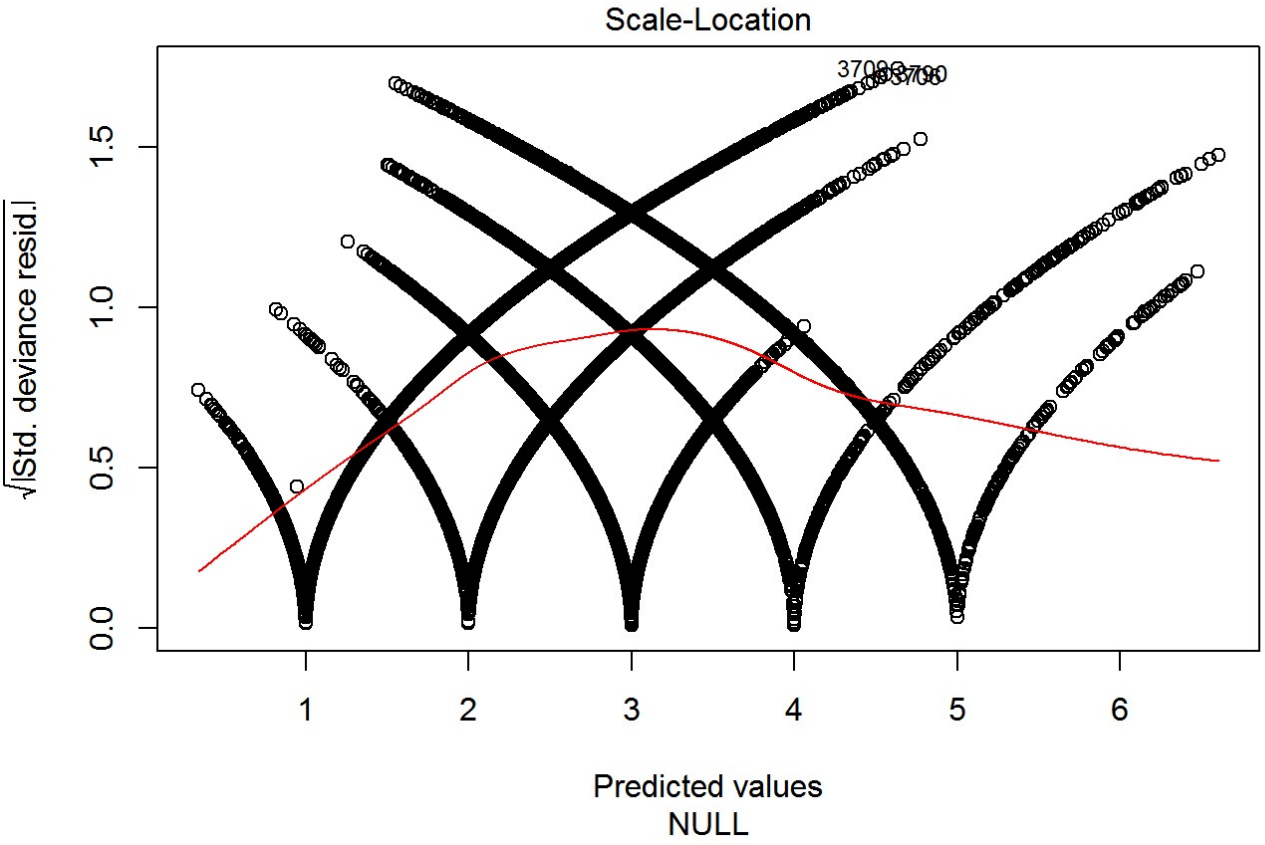
[*] Velloso, E.; Bulling, A.; Gellersen, H.; Ugulino, W.; Fuks, H. *Qualitative Activity Recognition of Weight Lifting*

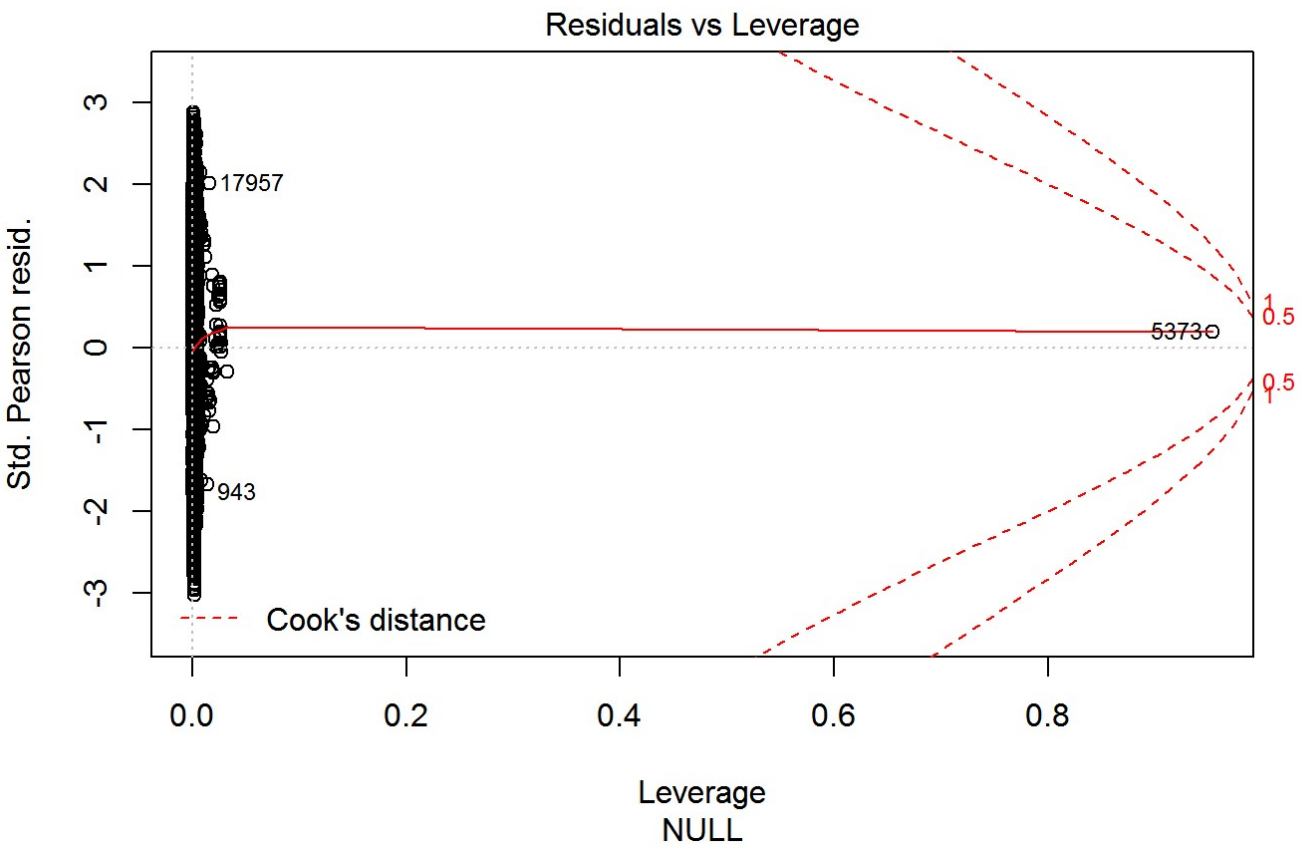
Exercises. Proceedings of 4th International Conference in Cooperation with SIGCHI (Augmented Human ’13). Stuttgart, Germany: ACM SIGCHI, 2013.

Appendix - analysis of residuals




Normal Q-Q
Std. deviance resid.
3
2
1
0
-1
-2
-3
-4
-2
0
2
4
Theoretical Quantiles
NULL


Scale-Location
√|Std. deviance resid.|
1.5
1.0
0.5
0.0
1
2
3
4
5
6
Predicted values
NULL


Residuals vs Leverage
Std. Pearson resid.
3
2
1
0
-1
-2
-3
17957
943
5373
1
0.5
0.5
1
Cook's distance
0.0
0.2
0.4
0.6
0.8
Leverage
NULL