Introduction to Machine Learning - Exercise 1

Mikko Ahro

Problem 1

Task a

Read p1.csv into dataframe and drop columns “id”, “SMILES”, “InChIKey”. Columns are dropped with following R commands:

```
p1data <- read.csv("data/p1.csv", header=TRUE, sep=",")
p1data <- subset(p1data, select=-c(id, SMILES, InChIKey))
```

Task b

Summary statistics for variables “pSat_PA”, “NumOfConf” and “ChemPot_kJmol”:

	pSat_Pa	NumOfConf	ChemPot_kJmol
	Min. : 0.0000	Min. : 2.00	Min. :-3.160
	1st Qu.: 0.0000	1st Qu.: 73.25	1st Qu.: 9.723
	Median : 0.0001	Median : 172.50	Median :12.781
	Mean : 2.9620	Mean : 223.50	Mean :12.434
	3rd Qu.: 0.0023	3rd Qu.: 324.25	3rd Qu.:15.659
	Max. :562.8970	Max. :1058.00	Max. :28.096

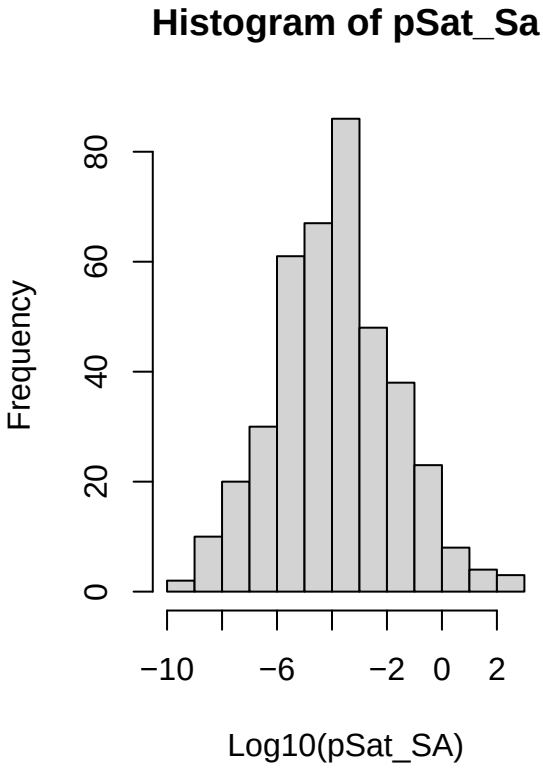
Task c

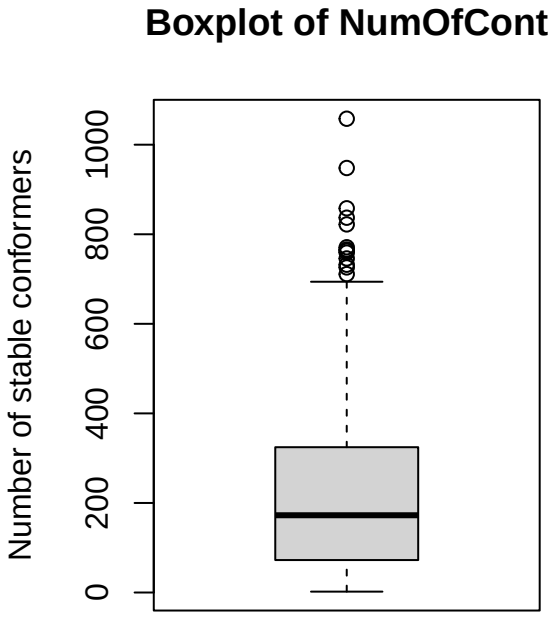
Mean and standard deviation of column ‘ChemPot_kJmol’ are:

[1] “Mean: 12.4344270896”

[1] “Standard deviation: 4.77887217784492”

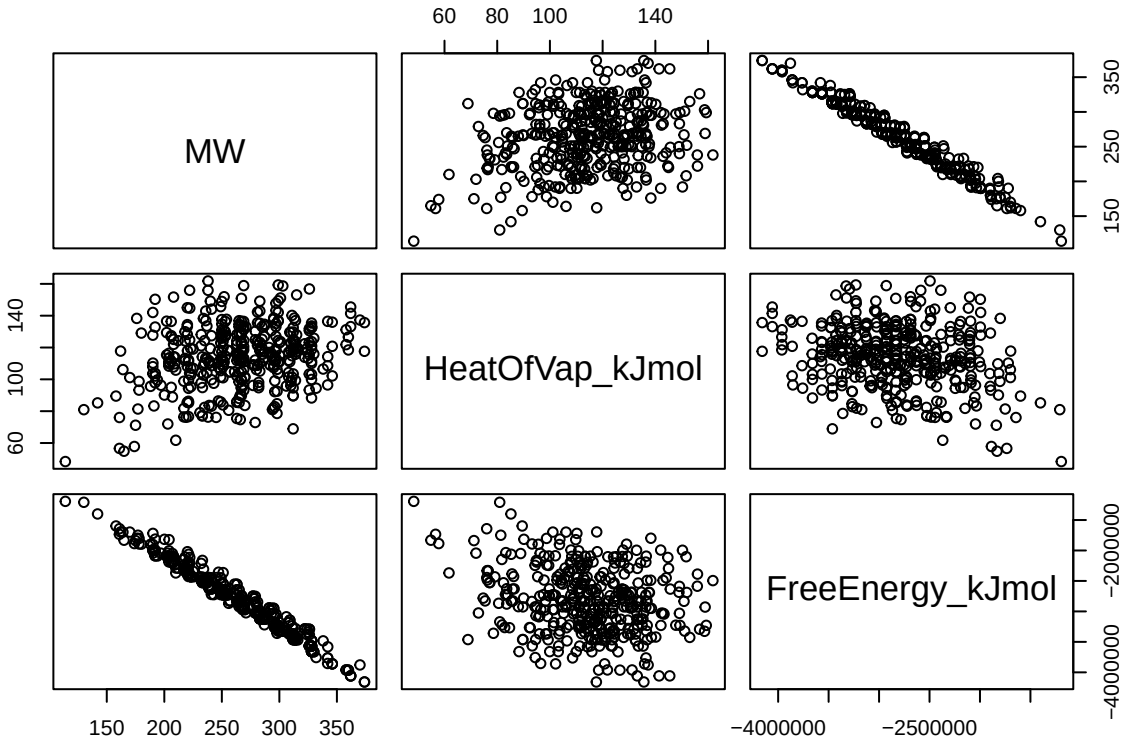
Task d





Task e

Scatterplot of variables "MW" (The molecular weight of the molecule [g/mol]), "HeatOFCap_kJmol" (the heat of vaporisation [kJ/mol]), and "FreeEnergy_kJmol" (the free energy of a molecule in mixture [kJ/mol])



Problem 2

Task a

Polynomial curve: $\hat{y} = \sum_{k=0}^{p} w_k x^k$ fitted to synthetic data from files "train_syn.csv" as training data and "valid_syn.csv" as validation data. Results are shown in table below. Explation of table columns and rows:

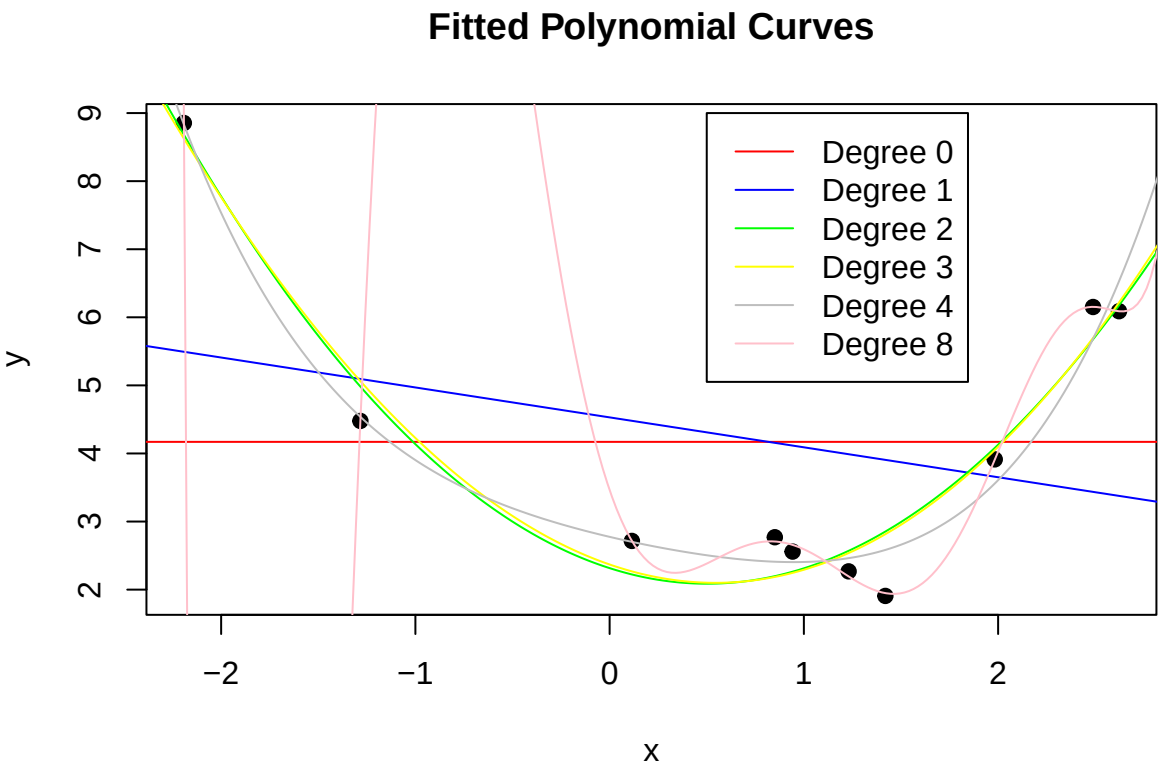
- Train is the training loss (train model on training set, report error on training set)
- Validation is the validation loss (train model on training set, report error on validation set)
- Test is the testing loss (train model on training set, report error on test set)
- TestTRVA is another testing loss (train model on the combined training and validation data, report error on test set)
- CV is the MSE from 5-fold cross-validation on the combined training and validation data

Degree	Train	Validation	Test	TestTRVA	CV
0	4.5122613	4.512261	4.512261e+00	4.587239	5.9599500
1	4.0885351	3.494124	5.206372e+00	4.786172	10.2371432
2	0.2185859	7.021118	1.424954e+01	14.791603	0.3598506
3	0.2168190	7.154893	1.383458e+01	14.096042	0.3699106
4	0.1187955	8.776121	1.968113e+01	15.009734	0.5187331
5	0.0965322	7.221166	2.975686e+01	20.134323	0.4561640
6	0.0075741	6.050151	1.564333e+02	12.060633	21.0914594
7	0.0049994	11.394430	1.104038e+03	15.628661	698.0709594
8	0.0020825	407.157118	1.561695e+05	10.979288	0.8388804

In this case I would choose model with polynomial degree 2 because 5-fold cross-validation has minimum loss here.

Task b

Training set and fitted polynomial curves for degrees 0,1,2,3,4 and 8 shown in below figure:



Task c

Datasets used are weather data "train_real.csv" for training and "test_real.csv" for testing.

Next day's maximum temperature (variable Next_Tmax) is predicted with following models:

- dummy model (see the discussion below)
- OLS linear regression (simple baseline)
- random forest (RF)
- support vector regression (SVR)
- ridge regression (RR)

	Train	Test	CV
Dummy	9.0180282	10.564483	9.085679
OLS	1.9750655	2.391045	2.163908
RF	0.3256189	2.348263	1.955547
SVR	2.0296744	2.374756	2.268225
RR	2.0140241	2.480093	2.176158

In the table the columns are:

- "Train": training loss
- "Test": testing loss
- "CV": 10-fold cross-validation loss

Answers to the questions:

1. Which regressor is the best? Why?

- Based on this data, Random forest seems to be the best model, because both test and CV losses are smaller than for other models.
- "Best" is to be taken with caution here: no special attempts to optimize models were made (for example no feature selection or hyperparameter optimization), but models were run with all features included, and single set of hyperparameters. Models should be optimized, and "best" should be selected based on optimization. See point 3.

2. How does Train compare to Test? How does CV compare to Test?

- Training error is smaller for all models than testing. This is natural, as model is trained with training set.
- For all models, CV is lower than test loss, but higher than training loss. This is also as it should be: CV is using only training data, but dividing it into random train and test sets, and repeating process k-times. The purpose of CV is to provide realistic estimate of test error, and for this train/data set for all models works this way.
- It is notable that for all other models train, test and CV losses are relatively close to each other, but for RF train error is significantly smaller than for any other model. The reason for this is likely to be built-in feature selection of RF, while all other models were used without feature selection.

3. How can you improve the performance of these regressors (on this training set)?

- Feature selection was not done, but all features were used for all models. Feature selection would likely improve the performance or at least validate that all features is proper selection
- Random Forest and SVR have hyperparameters that should be optimized. In this case, models were just run with default parameters (SVM with linear kernel). Hyperparameter optimization could improve performance.

Problem 3

Task a

Typical behaviour of the following terms, as we go from less flexible to more flexible statistical learning methods:

- training error and testing error: training error is generally decreasing when flexibility is increasing. This is because more flexibility the model has, easier it is to fit the model to the trainign data. Smamll training error, however, does not necessarily mean good model, as overfit may happen. Test error decreases first when flexibility increases, but at some point the model starts overfitting the training data, and from this point test error inceasees.

As a simple example, if we consider linear data $y = ax + \epsilon$ and we fit linear curve with linear regression, and we fit polynomial curve (degree > 1) with linear regression: polynomial curve has almost automatically smaller trainign error than linear, because polynomial model explains at least some of the noise with polynomial constants for terms higher than one. On the other hand, this is clear overfitting, and test error is likely to be higher with polynomial model than with linear model.

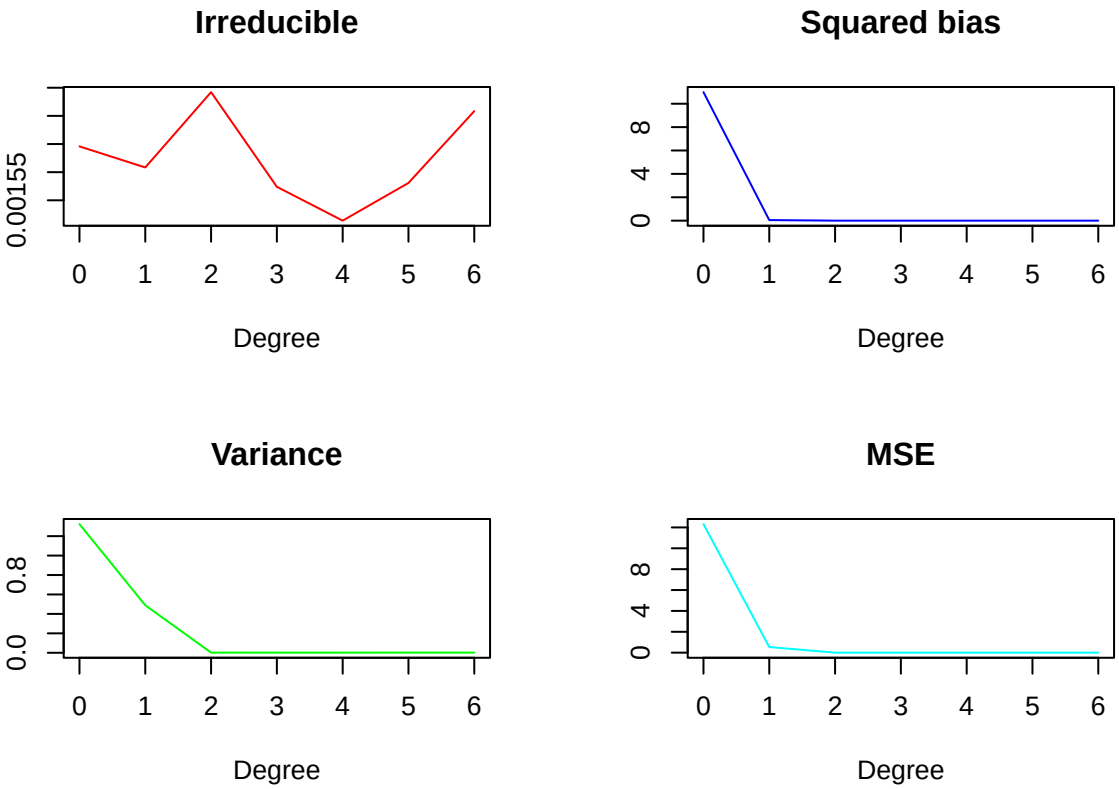
- (squared) bias: bias describes the difference between actual phenomenom modeled and model. Bias in general decreases when flexibility increases. However, at some point bias starts increasing, because too flexible model starts overfitting the model to the noise.
- variance: variance can be thought to describe the difference between estimates received as result from using different training sets. Generally more flexible methods have higher variance. But on the other hand, if method is too inflexible, it cannot catch the complexity of data, in which case variance is also high. As summary variance decreases first as flexibility increases, but starts increasing when flexibility grows too much.
- irreducible (or Bayes) error: irreducible error means the inherent error in data due to noise. Because noise is random and inherent property of training and testing data, this is the part of error that cannot be reduced by improving improving the model. In other words, irreducible error always remains, no matter what is done for the model. Hence, irreducible error doesn't vary according to model flexibility.

Task b

Data point $y = f(x) + \epsilon$ where $f(x) = 2 - x + x^2$ and $\epsilon \sim Normal(0, 0.04^2)$ and $x \sim Uniform(-3, 3)$. Polynomial regression function $\hat{f}$ is of degree p is trained using a data set D of n data points.

- 1000 training sets
- 10 data points in each
- each training set is fitted to polynomial function of degree 0-8
- Irreducible, BiasSq, Variance, Total and MSE calculated for each polynomial

Degree	Irreducible	BiasSq	Variance	Total	MSE
0	0.0016458	10.9915161	1.3234498	12.3166117	12.3149659
1	0.0016083	0.0479555	0.4887802	0.5383441	0.5367357
2	0.0017422	0.0000003	0.0015585	0.0033009	0.0015588
3	0.0015740	0.0000012	0.0014923	0.0030676	0.0014936
4	0.0015139	0.0000003	0.0014686	0.0029828	0.0014690
5	0.0015808	0.0000000	0.0015987	0.0031796	0.0015987
6	0.0017084	0.0000008	0.0015454	0.0032547	0.0015462



The error terms roughly behave as described in Task a:

- Irreducible is random without clear pattern, and order of magnitude as expected from ϵ
- Squared bias decreases until degree 2, after which it remains relatively constant
- Variance clearly decreases until degree 2, after which remains relatively constant.
- MSE decreases until degree 2, after which is remains relatively constant.

For degrees 0-1 Total is almost exactly MSE. For higher degrees, Total is higher than MSE, but the abosolute value and absolute differences are very small. We don't see bias and variance growing when degree is higher than 2, but they remain constant. These behaviors is slightly different from discussion in Task a, but there is quite clear explanation: the training data is 2nd degree polynomial function with very small level of noise added. In the range [-3, 3] the function takes values from 1.75 to 14, while noise is only $\epsilon \sim Normal(0, 0.04^2)$. This means that in practice the fitting is likely to give coefficients for polynomials of x for higher degrees than 2 very close to zero. In other words, the value of function is dominant over noise term, and therefore we don't see increase of bias and variance.

Problem 4

Not done

Problem 5

Task a

Datasets "d1.csv", "d2.csv", "d3.csv", and "d4.csv" each 11 observations of (x,y) pairs are fitted with 1 variable linear regression $\hat{y} = w_0 + w_1 x$. The range of data (depending on dataset) is x [3, 19] and y [3.1, 12.74]. For each following are presented in table below:

- Intercept term estimate, standard error and p-value
- Slope term estimate, standard error and p-value
- R-squared value for the model

Data	Intercept	Int_SE	Int_p_value	Slope	Slope_SE	Slope_p_value	R_Squared
d1	3.000091	1.124747	0.0257341	0.5000909	0.1179055	0.0021696	0.6665425
d2	3.000909	1.125302	0.0257589	0.5000000	0.1179637	0.0021788	0.6662420
d3	3.002454	1.124481	0.0256191	0.4997273	0.1178777	0.0021763	0.6663240
d4	3.001727	1.123921	0.0255904	0.4999091	0.1178189	0.0021646	0.6667073

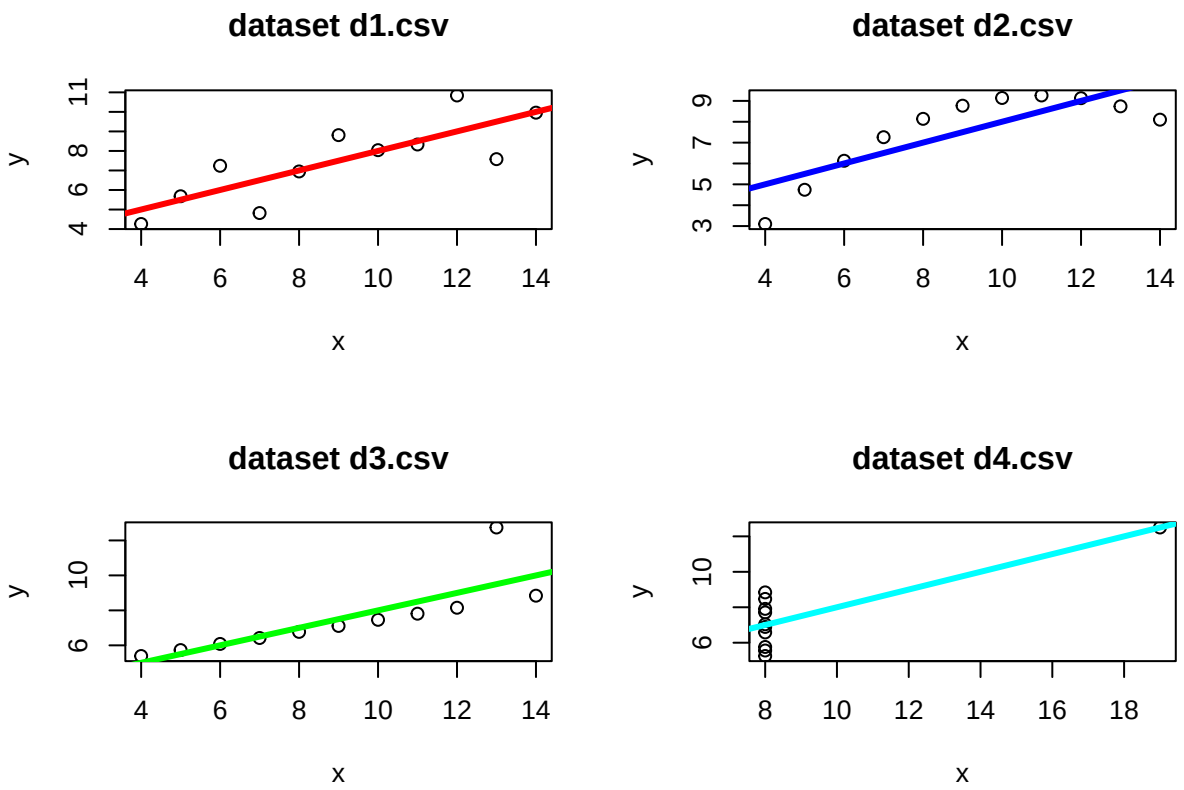
For all datasets reported values are almost equal, with positive slope and positive intercept term.

- p-value is approx 0.002 for each dataset, which can be considered small in indicating statistical significance
- standard errors are relatively high (approx 35% for intercept and 25% for slope), but still within range indicating that slope and intercept have correct sign and magnitude
- R-squared values are approx 67%. Interpretation of R-squared valua is highly application dependent, but one could say that 67% indicates that fit works somehow at least.

Still this is not sufficient to conclude safely that when x increases (or decreases) y increases (or decreases). Standard errors. Linear model can involve multiple problems that are not well described by standard error, p-value and R-squared alone.

Task b

Each data set and fitted regression line is plot below:

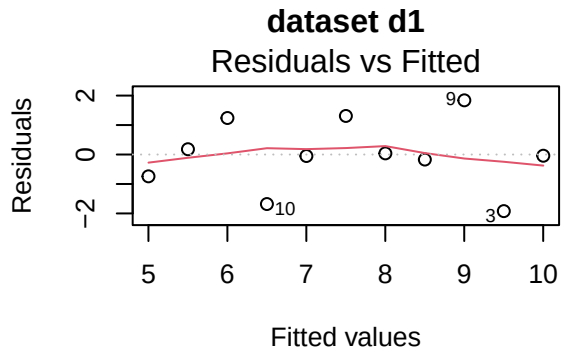


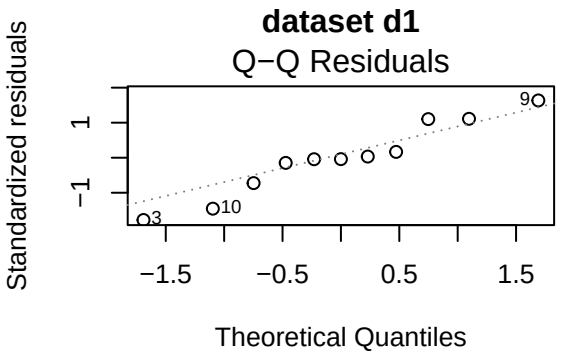
For each dataset the x-range is similar (approx 4-13 for others, except 8-19 for set d), also y-range is quite similar starting from 3-5 and and ending to 9-13. Fitted model is almost the same for each data set. The sets seem to have roughly raising trend (i.e. y increasing when x is increasing). All datasets vs models, however, have significant deviations from fitted model.

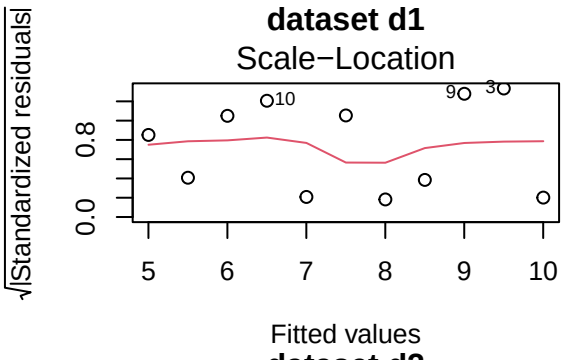
Task c

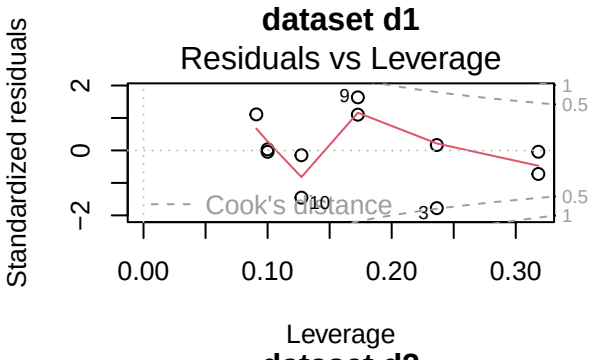
Potential problems with regression models: 1. Non-linearity of the response predictor relationship 2. Correlation of error terms 3. Non-constant variance of error terms 4. Outliers 5. High-leverage points 6. Collinearity

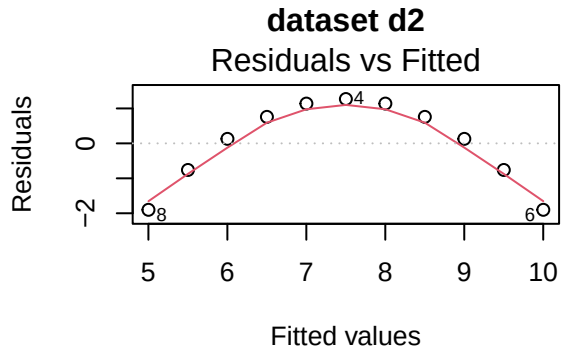
These potential problems can be investigated by analyzing various diagnostic plots of model. These plots are shown below:

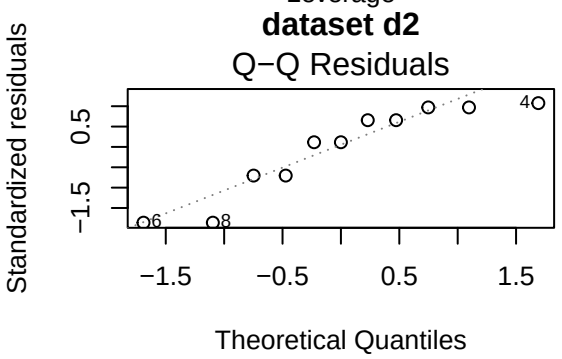

dataset d1
Residuals vs Fitted
Residuals
Fitted values

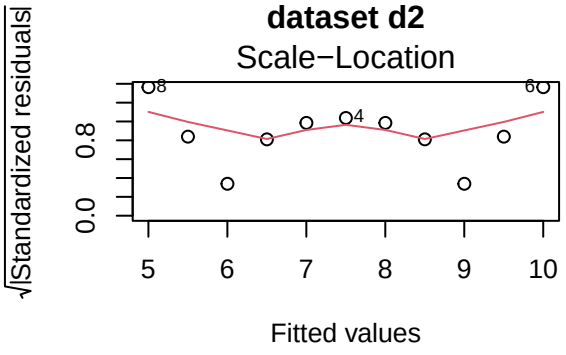

dataset d1
Q−Q Residuals
Standardized residuals
Theoretical Quantiles

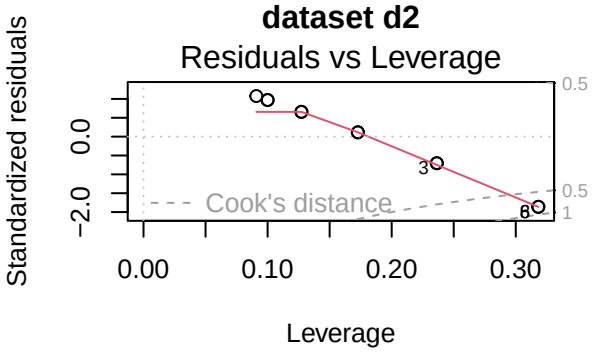

dataset d1
Scale−Location
√|Standardized residuals|
Fitted values

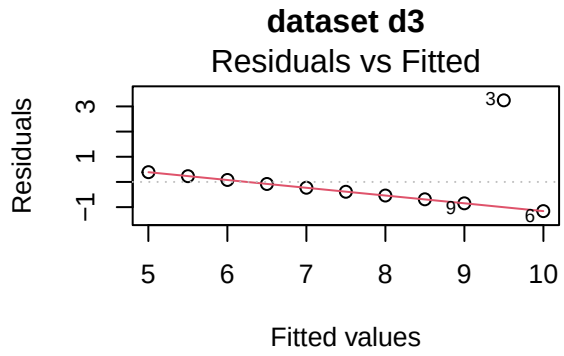

dataset d1
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage

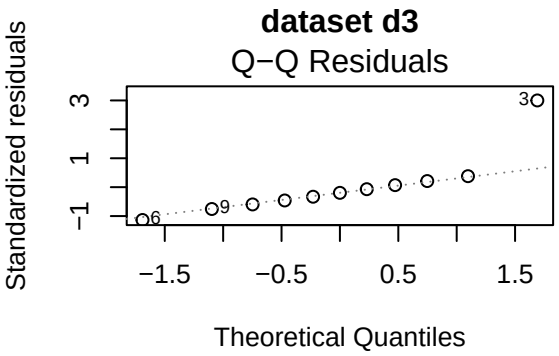

dataset d2
Residuals vs Fitted
Residuals
Fitted values

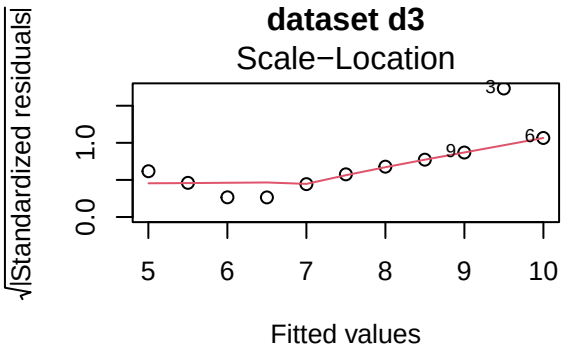

dataset d2
Q−Q Residuals
Standardized residuals
Theoretical Quantiles

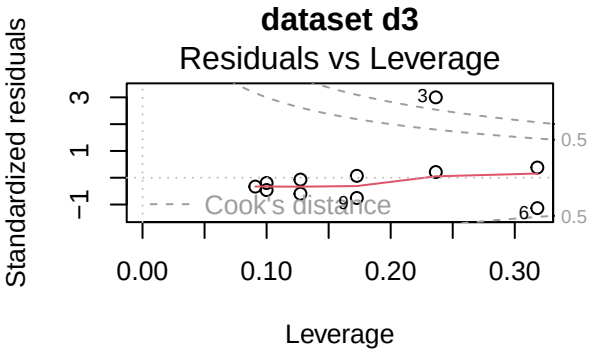

dataset d2
Scale−Location
√|Standardized residuals|
Fitted values


dataset d2
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage

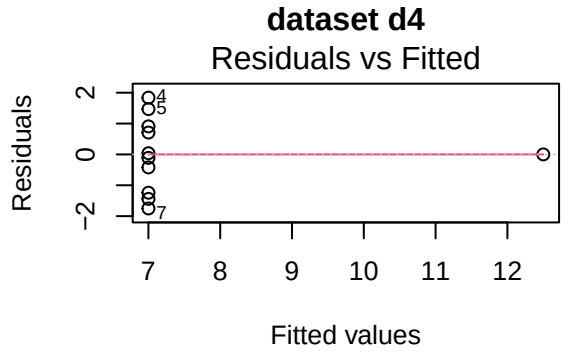


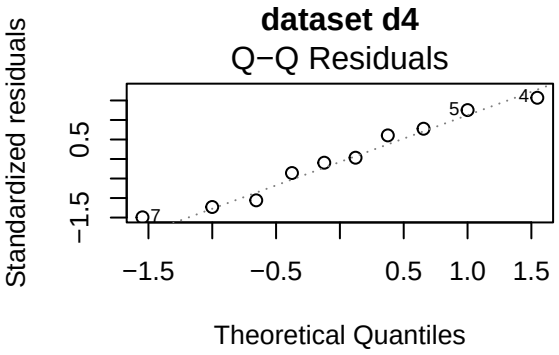


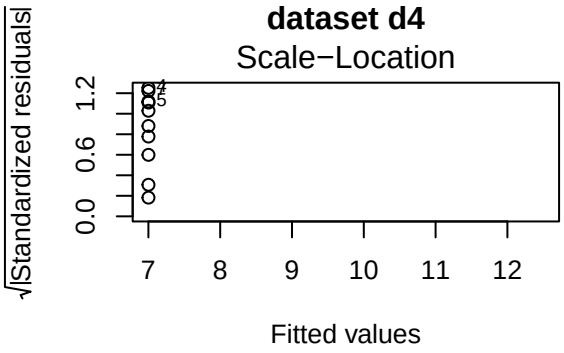


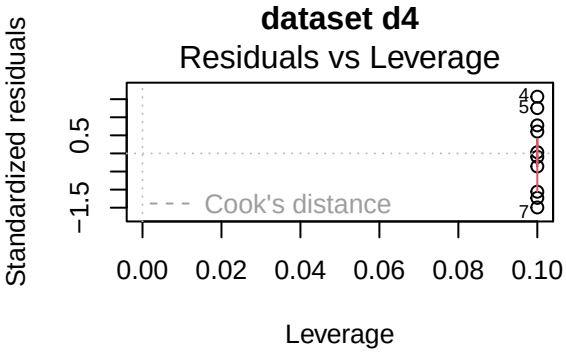



```
## Warning: not plotting observations with leverage one:
##   8
```









Potential problems analyzed from above diagnostic plots:

1. Non-linearity of the response predictor relationship can be best analyzed from Residuals vs fitted plot. If residuals vs fitted shows clear non-linear pattern, it indicates that there is non-linear relationship the model doesn't catch.

- dataset d1 does not seem to suffer from non-linear relationship
- residuals vs fitted plot for dataset d2 shows very clear parabolic pattern, indicating clear non-linearity problem
- residuals vs fitted plot for dataset d3 is harder to interpret: the residuals are clearly not evenly spread, but there is no non-linear pattern, but linear trend with residuals. The plot indicates that there is something wrong with d3, but the problem is not likely to be non-linear relationship
- dataset d4 does not seem to suffer from non-linear relationship

2. Correlation of error terms: correlation of error terms is not straightforward to analyze from standard diagnostic plots, but residuals vs fitted gives some indication:

- dataset d1: no pattern in residuals
- dataset d2: clear parabolic pattern indicating correlation of error terms
- dataset d3: no clear pattern in residuals
- dataset d4: no clear pattern in residuals

3. Non-constant variance of error terms can be analyzed from Q-Q Residuals plot and from Scale-Location plot.

- dataset d1 shows linear Q-Q residuals and approximately horizontal Scale-location line both with quite randomly spread error terms, indicating that there is no major problem
- dataset d2 shows also fairly linear Q-Q residuals and fairly horizontal scale-location. The error terms are not fully randomly spread, but still there is no clear indication of non-constant variance problem
- dataset d3: Q-Q is linear, but residuals are not randomly split. Scale-location polot shows raising tail and non-randomly split error terms. This indicates at least some level of non-constant variance problem.
- dataset d4: Q-Q plot is linear and residuals are fairly random. Scale-location plot has no trend-line at all, which indicates problem with the model, but not necessarily with non-constant variance of error tersm.

4. Outliers are best seen from Residuals vs Leverage plot.

- for dataset d1 there are no indications of clear outliers
- for dataset d2 there are clear indications of outliers
- for dataset d3 one observation is not even plotted, but instead warning is given. This is indication of serious problem. It can clearly be seen from plot of original data set points vs fitted model that point (13, 12.74) is clear outlier.
- for dataset d4 the plot is straight vertical line, which is not normal. Dataset 4 is easiest to analyze from plot of original dataset points vs fitted model, where one can clearly see that 10 x values are 8, and 1 x value is 19. Point (19, 12.50) is clear outlier

5. High-leverage points are also best seen from Residuals vs Leverage plot.

- for dataset d1 there are no indications of clear high leverage points
- for dataset d2 there are no clear indications of high leverage points
- for dataset d3 high-leverage point diagnostic plot is indicating leverange 1 for one point, and from plot of original data set points vs fitted model it can be seen that point (13, 12.74) is clearly high-leverage point.
- with similar analysis as for outliers, point (19, 12.50) is clear high leverage point.

6. Collinearity: in this case, each dataset contains only (x,y) pairs and hence only 1 explanatory variable. Collinearity means collinearity between explanatory variables, and requires minimum 2 explanatory variables to happen. So none of the datasets / models suffer from collinearity.

As a summary problems suffered by linear models for datasets d1-d4

Dataset	d1	d2	d3	d4
Non-linearity		Yes		
Correlation of error		Yes		
Non-constant variance of error			Yes	
Outliers			Yes	Yes
High-leverage			Yes	Yes
Collinearity				

Problem 6

Task a

For data set d2.csv intercept and slope and their standard errors are calculated with bootstrap method with 1000 bootstrap estimates. These are reported and compared to values obtained by single linear regression in table below:

	Intercept	SE intercept	Slope	SE slope
Linear model	3.000909	1.125302	0.5	0.1179637
Bootstrap	3.000909	1.456787	0.5	0.1566746

Bootstrap standard errors are more reliable. This is because the dataset is very small, only 11 observations, and bootstrap is resampling the dataset with replacement and repeating the process multiple times. Hence the standard errors are calculated from much larger set of fit data than in single linear regression.

Task b

Bootstrap algorightm takes following steps for calculating standard errors for intercept and slope parameters:

1. Algorithm samples 11 datapoints with replacement from original dataset (no subsetting was used), this is bootstrap sample
2. Bootstrap sample is taken as trainig data for linear regression (1st degree)
3. Intercept and slope coefficients are saved into table
4. Steps 1-3 are repeated 1000 times (depending on parameter given to bootstrap function) until the table has 1000 rows
5. Finally standard error is calculated as standard deviation for intercept column and slope column, respectively

Task c

In bootstrap, you sample n data points from a population of n points with replacement. Argue that the probability that the jth observation is not in the bootstrap sample is about 0.368 when n is very large.

For each draw probability of sampling jth observation from sample size of n is:

$$P(j) = 1/n$$

Probability of not sampling jth observation is complement of this:

$$P(\bar{j}) = 1 - P(j) = 1 - \frac{1}{n}$$

Because we are sampling with replacement probability of not sampling jth observation is the same at each draw. Also, each draw is independent from other draws, because we are sampling with replacement. Therefore combined probability of n draws is:

$$P(\text{observation j is not included in n draws}) = \left(1 - \frac{1}{n}\right)^n$$

with large n:

$$\lim_{n\to\infty} \left(1 - \frac{1}{n}\right)^n \approx 0.368$$

Problem 7

Task a

I had some practical experience with some of the machine learning concepts covered in lectures 1-4 such as linear simple and multiple linear regression. I also had rough conceptual understanding of training, validation and testing data. During this block I understood these concepts much deeper, and got some theoretical framework around them, which I didn't have much before.

Probably the biggest learning outcome was model evaluation and various error types. Training and testing losses as well as k-fold cross-validation were something that I knew superficially, but this block helped to understand what they actually mean. Division of error to irreducible, bias and variance was totally new concept to me, and helped to understand how regression models work. Bootstrap also was new technique for me, and I guess I understand now the principle how it works.

Ridge regression and Lasso were also new concepts. By now I can understand the basic concept, and how to apply them, but I have difficulties following the formal mathematics behind these models.

The contents seem quite interesting and probably relevant for other studies as well, but of course it is difficult to say what is relevant in future courses before taking them.

Feedback to the course: interesting and practical contents, but the workload is really quite high compared to most (or any) other course I've taken so far. I wouldn't say I'm especially slow, normally I need bit less than 27 hours per credit, but in this course the estimated 135 hours is not going be nearly enough.

Task b

Estimated hours: 32.