# Yurii Mykhalchuk's third homework

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-13
```

```
library(readxl)
library(np)
```

```
## Nonparametric Kernel Methods for Mixed Datatypes (version 0.60-6)
## [vignette("np_faq",package="np") provides answers to frequently asked questions]
## [vignette("np",package="np") an overview]
## [vignette("entropy_np",package="np") an overview of entropy-based methods]
```

```
library(stats)
library(MASS)
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following object is masked from 'package:glmnet':
##
##     auc
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
set.seed(1)
data <- read_excel("ceo.xls")
data$X__1 <- NULL
```

– – 1 – –

*a*

**Q:** *Explain in your own words the idea of the lasso regression. Sketch a situation when a simple linear regression fails, but the lasso regression still can be estimated.*
**A:** The idea of the lasso regression is to drop hightly correlated variables and to leave only the most significant ones. The lasso regression is more suitable than linear when we have a lot of variables which are highly correlated or when a number of variables is greater than a number of observations.

*b*

**Q:** *For the usual regression model the variables are rarely normalized/standardized. However, in the case of the lasso regression the scaling becomes crucial. Why?*
**A:** Because if variables will have different magnitude of values they will have different impact on models coefficients.

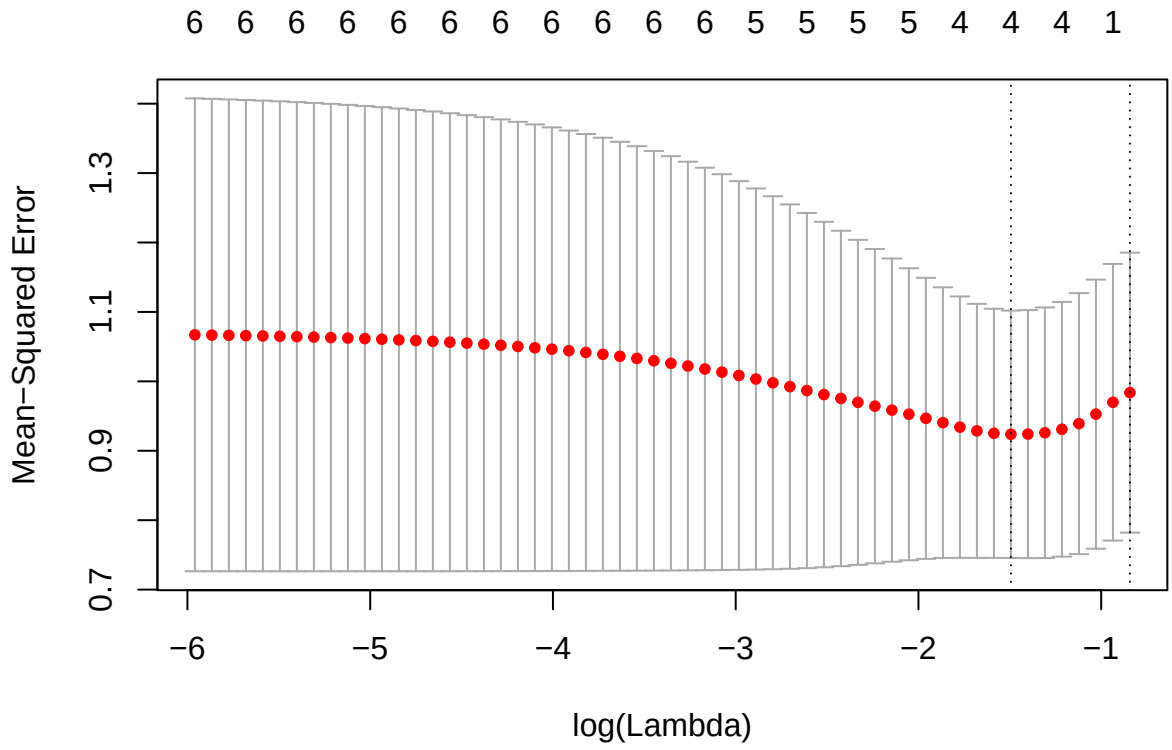
*c*

**Q:** *Run a lasso regression for scaled ((x i − x¯ )/ˆ x ) data with (0, 1). Plot the estimated parameters as functions of . Which value of would you recommend?*

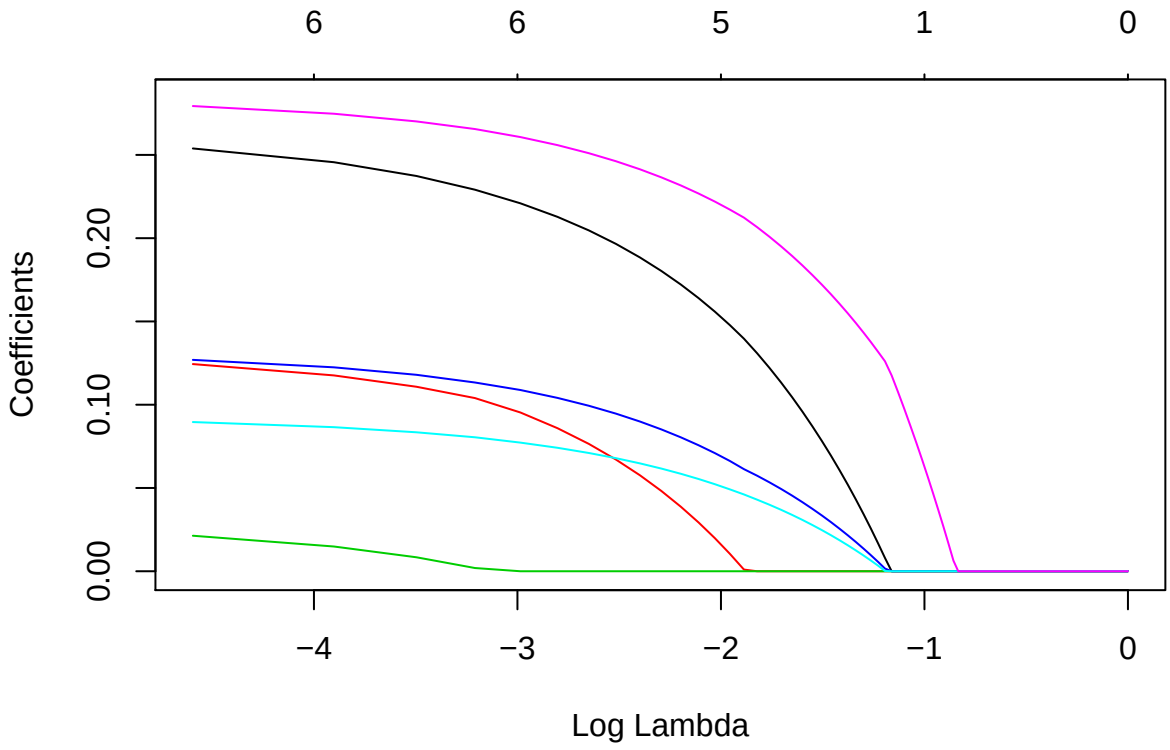
**A:**

```
variables <- scale(data[, -1])
salary <- scale(data[, 1])
x <- as.matrix(variables)
y <- as.matrix(salary)

grid <- seq(0, 1, length=100)
lasso <- glmnet(x, y, alpha=1, standardize=TRUE, lambda=grid)
cv.lasso <- cv.glmnet(x, y, alpha=1)
plot(cv.lasso)
```




```
plot.glmnet(lasso, xvar="lambda", label=TRUE)
```




```
cv.lasso$lambda.min
```

```
## [1] 0.224491
```

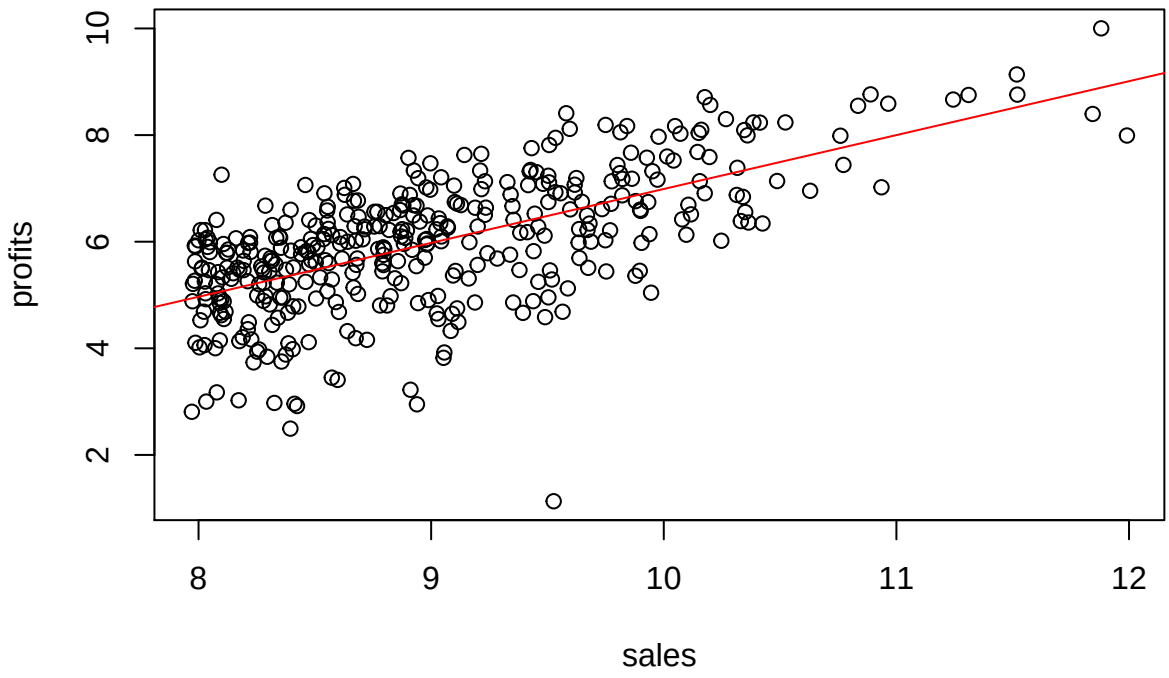
I would recommend to use lambda = 0.224491.

– – 2 – –

*a*

**Q:** *Make a bivariate scatter plot and estimate an appropriate linear model. Add the regression curve to the plot.*
**A:**

```
# remove negative profits
data <- data[data$profits > 0, ]
profits <- log(data$profits)
sales <- log(data$sales)
linear_model <- lm(profits ~ sales)
plot(sales, profits)
abline(linear_model, col='red')
```

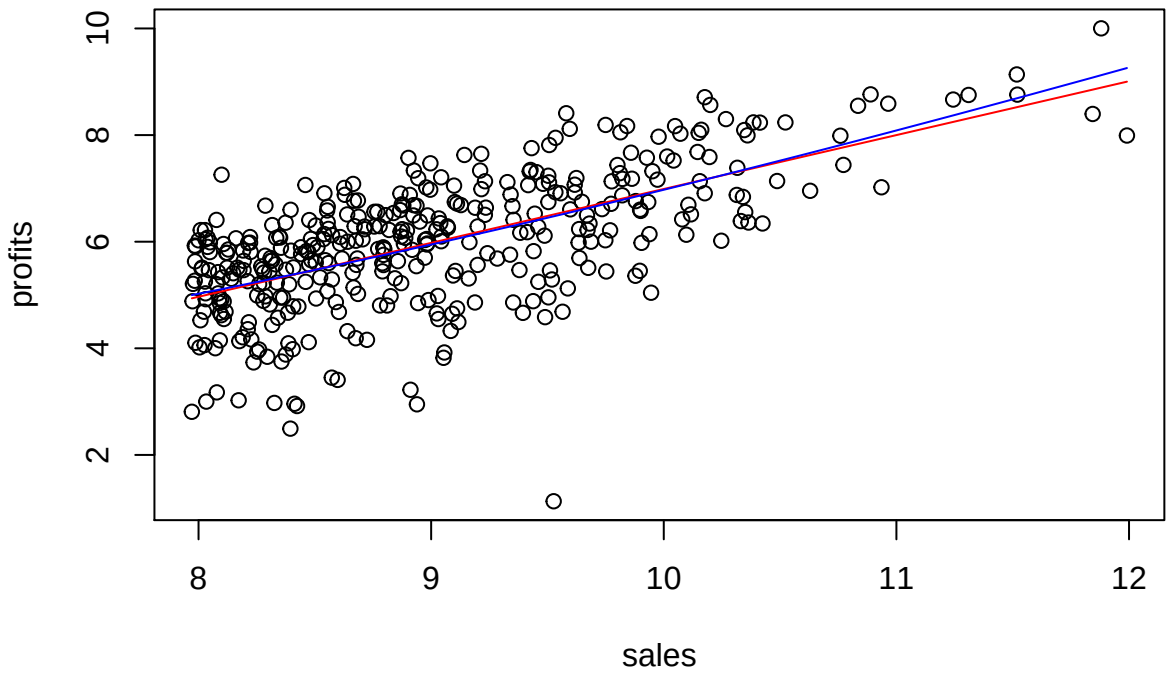



*b*

**Q:** *Estimate now an appropriate nonlinear regression which might fit the data better. Add the regression curve to the plot and compare the fit with the fit of the linear model.* **A:**

```
non_linear_model <- nls(profits ~ a * log(sales) + b * sales + c, start = list(a = -1, b = 1, c = 1000))
profits_hat_linear <- predict(linear_model)
profits_hat_non_linear <- predict(non_linear_model)

plot(sales, profits)
lines(sales, profits_hat_linear, col = "red")
lines(sales, profits_hat_non_linear, col = "blue")
```




```
loss.functions = function(x, x.hat)
{
  res = c(mean((x - x.hat) ^ 2),
          mean(abs(x - x.hat)),
          mean(abs((x - x.hat) / x )))
  names(res) = c("MSE","MAE", "MAPE")
  return(res);
}
loss.functions(profits, profits_hat_linear)
```

```
##       MSE       MAE      MAPE
## 0.9127472 0.7507421 0.1512142
```

```
loss.functions(profits, profits_hat_non_linear)
```

```
##       MSE       MAE      MAPE
## 0.9111812 0.7504160 0.1511381
```

```
cat("Correlation of profits with profits estimated by a linear model: ", cor(profits, profits_hat_linea
```

```
## Correlation of profits with profits estimated by a linear model:  0.6429063
```

```
cat("Correlation of profits with profits estimated by a non-linear model: ", cor(profits, profits_hat_n
```

```
## Correlation of profits with profits estimated by a non-linear model:  0.6436887
```

Model became better, but not much. I am not very good at guessing :(

*c*

**Q:** *Explain in your own words, why all the classical tests and inferences are not directly applicable to the NLS estimators.*
**A:** For a linear regression: SS Regression + SS Error = SS Total. But for non-linear models this equation is not true. Also I googled and found that:
1) R-squared tends to be uniformly high for both very bad and very good models.
2) R-squared and adjusted R-squared do not always increase for better nonlinear models.
3) Using R-squared and adjusted R-squared to choose the final model led to the correct model only 28-43% of the time.

– – 3 – –

*a*

**Q:** *An important calibration parameter of a nonparametric regression is the band-width. Explain what happens with the regression/the weights in the Nadaraya-Watson regression if the bandwidth is too high or too small.*
**A:** When a bandwidth is too high, our model is not fitting data good, it's more like a straight line. It's called underfitting. On the other hand, when bandwidth is too small, our model is overfitted. Tt's trying to pass through each point in out dataset, which is not good for future predictions.

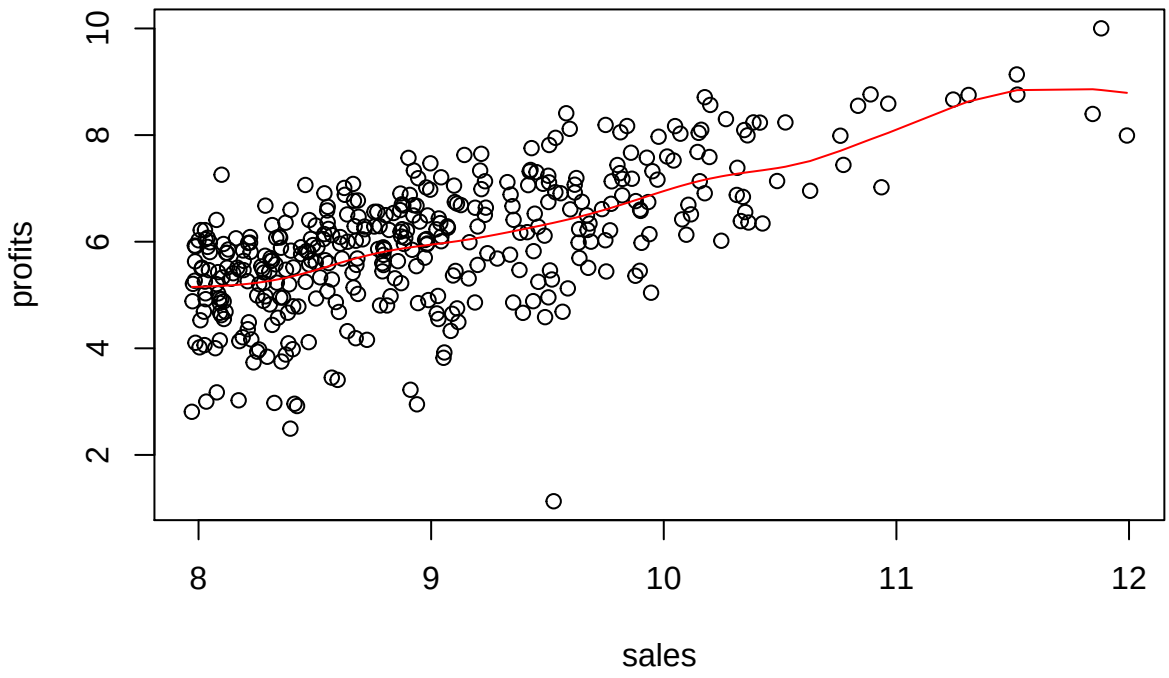
*b*

**Q:** *Fit a Nadaraya-Watson regression with Gaussian kernel and "optimal" bandwidth to the profits/sales data. Check and explain how the "optimal bandwidth" is determined in your software. Plot the data and the regression curve.*
**A:**

```
bandwidth <- npregbw(profits ~ sales, lt = "lc")
```

```
##
Multistart 1 of 1 |
Multistart 1 of 1 |
Multistart 1 of 1 |
Multistart 1 of 1 /
Multistart 1 of 1 |
Multistart 1 of 1 |
```

```
nw_model <- npreg(bws = bandwidth)
plot(sales, profits)
lines(sales, fitted(nw_model), col = "red")
```




The optimal bandwith is determined using Kullback-Leibler topology.

– – *c* – –

**Q:** *Compare the fit of the nonparametric regression and the nonlinear regression in the previous subproblem.*
**A:**

```
profits_hat_non_parametric <- predict(nw_model)
cor(profits, profits_hat_non_linear)
```

```
## [1] 0.6436887
```

```
cor(profits, profits_hat_non_parametric)
```

```
## [1] 0.6495991
```

```
loss.functions(profits, profits_hat_non_linear)
```

```
##       MSE       MAE      MAPE
## 0.9111812 0.7504160 0.1511381
```

```
loss.functions(profits, profits_hat_non_parametric)
```

```
##       MSE       MAE      MAPE
## 0.9015954 0.7444075 0.1498249
```

A non-parametric model has better fit than non-linear one.

– – 4 – –

*a*

**Q:** *Fit a logistic regression to explain Y by the remaining explanatory variables. Run a stepwise model selection using AIC as criterion. Further consider only the optimal model chosen here.* **A:**

```
new_data <- data
new_data$high_salary <- ifelse(new_data$salary > 2000, 1, 0)
new_data$salary <- NULL
new_data$totcomp <- NULL
logit_model <- glm(high_salary ~ ., family=binomial(link='logit'), data=new_data)
summary(logit_model)
```

```
##
## Call:
## glm(formula = high_salary ~ ., family = binomial(link = "logit"),
##     data = new_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3374  -0.7463  -0.6117   0.7783   1.9742
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.202e+00  1.137e+00  -2.817 0.004844 **
## tenure       2.918e-02  1.494e-02   1.953 0.050814 .
## age          2.061e-02  2.022e-02   1.019 0.308005
## sales        2.653e-05  1.752e-05   1.514 0.130018
## profits      8.656e-04  2.241e-04   3.863 0.000112 ***
## assets       6.563e-06  3.677e-06   1.785 0.074285 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 496.21  on 387  degrees of freedom
## Residual deviance: 402.63  on 382  degrees of freedom
## AIC: 414.63
##
## Number of Fisher Scoring iterations: 6
```

```
fitted_model <- step(logit_model, direction = "both")
```

```
## Start:  AIC=414.63
## high_salary ~ tenure + age + sales + profits + assets
##
##           Df Deviance    AIC
## - age      1   403.68 413.68
## <none>         402.63 414.63
## - sales    1   405.03 415.03
## - assets   1   406.30 416.30
## - tenure   1   406.41 416.41
## - profits  1   422.17 432.17
##
## Step:  AIC=413.68
## high_salary ~ tenure + sales + profits + assets
##
##           Df Deviance    AIC
```

```
## <none>          403.68 413.68
## - sales    1    406.06 414.06
## + age      1    402.63 414.63
## - assets   1    407.45 415.45
## - tenure   1    410.42 418.42
## - profits  1    423.12 431.12
```

$--b--$

**Q:** *Consider the explanatory variable sales. Obviously its parameter cannot be intepreted in the same way as for a linear regression. Provide the correct intepretation using odds.* **A:**

```
exp(coef(fitted_model))
```

```
## (Intercept)      tenure       sales     profits      assets
##   0.1249778   1.0361617   1.0000263   1.0008624   1.0000067
```

From this output we can see that a unit increase in sales increases the probability of having a high salary by 0.0000263%.

$--c--$

**Q:** *Randomly pick up five CEOs. Determine their probabilities of having the salary of more or less than 2000. Provide for the first of them the formula which may be used to compute this probability with inserted values of parameters and variables. If you want to predict the membership in one of the two groups for a particular CEO, what is the simplest way to proceed using these probabilities?*
**A:**

```
random_ceos <- new_data[10:15,]
random_ceos$high_salary_hat <- predict(fitted_model, newdata = random_ceos, type = "response")
random_ceos$high_salary_hat
```

```
## [1] 0.9951008 0.9997730 0.7514872 0.8975862 0.5755227 0.9139642
```

The simplest way to predict a membership in this case is to set threshold to 1/2. If high_salary_hat is greater than this threshold then CEO has high salary, otherwise - low salary.