OLLSCOIL NA hÉIREANN, CORCAIGH
THE NATIONAL UNIVERSITY OF IRELAND, CORK

COLÁISTE NA hOLLSCOILE, CORCAIGH
UNIVERSITY COLLEGE, CORK

Examination Session and Year	Winter 2018
Module Code	ST4060, ST6015
Module Name	Computer Intensive Statistical Analytics I Computer Analytical Techniques for Actuarial Applications
Paper Number	Paper Number: 1
External Examiner	Dr. Ji Yao
Head of Department	Prof. Finbarr O'Sullivan
Internal Examiner(s)	Dr. Michael Cronin
Instructions to Candidates	Please answer all questions.
Duration of Paper	3 hours.
Special Requirements	15 minutes Reading Time. The use of a non-programmable calculator is permitted.

PLEASE DO NOT TURN THIS PAGE UNTIL INSTRUCTED TO DO SO

THEN ENSURE THAT YOU HAVE THE CORRECT EXAM PAPER

List of (possibly) useful R functions:

```
abline()
approx()
as.numeric()
boxplot()
coef()
cut()
dnorm()
fitted()
glmnet()
lines()
lm()
loess()
lowess()
nrow()
plot()
points()
predict()
quantile()
rchisq()
read.csv()
sample()
set.seed()
seq()
smooth.spline()
summary()
t.test()
which()
```

Question 1 [15 marks]

No code is required for this question.

(a) Name the methods defined by the following three criteria $J_1(\theta), J_2(\theta)$ and $J_3(\theta)$, for $\theta = (\theta_1, \ldots, \theta_p)$, $\lambda, \lambda_1, \lambda_2 \in \mathbb{R}$ and given a sample of n data points $(\mathbf{X}, Y)$, where $\mathbf{X}$ is a multivariate data frame containing p covariates $(X_1, \ldots, X_p)$:

$$J_1(\theta) = \sum_{i=1}^{n}(Y_i - \theta_0 - \theta_1 X_{1,i} - \cdots - \theta_p X_{p,i})^2 + \lambda \sum_{j=1}^{p} \theta_j^2$$

$$J_2(\theta) = \sum_{i=1}^{n}(Y_i - \theta_0 - \theta_1 X_{1,i} - \cdots - \theta_p X_{p,i})^2 + \lambda \sum_{j=1}^{p} |\theta_j|$$

$$J_3(\theta) = \sum_{i=1}^{n}(Y_i - \theta_0 - \theta_1 X_{1,i} - \cdots - \theta_p X_{p,i})^2 + \lambda_1 \sum_{j=1}^{p} \theta_j^2 + \lambda_2 \sum_{j=1}^{p} |\theta_j|$$

[9]

(b) In the following example, we are considering the problem of predicting Ozone level in the atmosphere, Y, from a set of three environmental features $\mathbf{X} = (X_1, X_2, X_3)$ (respectively measures of solar radiation, average wind speed and maximum daily temperature). Propose two possible explanations for the differences observed between the following output estimates for the effects of these features $\mathbf{X}$ on Y, knowing that all three outputs are obtained by minimizing some penalized sum of squared residuals for a linear model:

	Intercept	Solar Radiation	Wind Speed	Max. Temperature
Output 1	41.8042	0.0002	-0.0108	0.0046
Output 2	-4.1108	0	0	0.5940
Output 3	15.6775	0	-0.1966	0.3648

[6]

Solution:
See R code.

(a) Names:

- $J_1(\theta)$ is ridge regression;
- $J_2(\theta)$ is the LASSO (or least absolute shrinkage and selection operator);
- $J_3(\theta)$ is the elastic net.

(b) Explanations:

- Different methods were used; all three outputs may come from ridge regression, but outputs 2 and 3 may also come from the LASSO or elastic net.
- All three outputs were obtained from, say, the LASSO, but with different values of regularization parameter λ.

Code used to generate output:

```
library(glmnet)
dat = na.omit(airquality)
x = as.matrix(dat[,2:4])
y = dat[,1]
g.ridge = glmnet(x, y, alpha=0)
g.lasso = glmnet(x, y)
g.elnet = glmnet(x, y, alpha=0.5)
coef(g.ridge)[,4]
coef(g.lasso)[,4]
coef(g.elnet)[,4]
```

Question 2 [30 marks]

To implement P-splines, you may use `smooth.spline()` in `R`. For question items (a), (b), (c) and (e), use training sample "FranceRates2004.csv", which contains mortality rates at all ages between 0 and 110 for the Male French population in 2004, except for data between ages 80 and 89:

```
dat = read.csv(file='examdata_2018-19/FranceRates2004.csv')
```

For question item (d), use test sample "FranceRates2004_test.csv" containing the true mortality rate values for ages between 80 and 89:

```
dat.test = read.csv(file='examdata_2018-19/FranceRates2004_test.csv')
```

(a) Compute a first P-spline for the training sample, using 15 degrees of freedom, and leaving the smoothing control parameter unspecified.

- Provide the R command(s) you used.
- Provide a plot of the dataset (black dots) along with the P-spline (red solid curve). **[5]**

(b) Compute a second P-spline for the training sample, setting the smoothing control parameter to .05 and leaving the number of degrees of freedom unspecified.

- Provide the R command(s) you used.
- Add a blue solid curve that shows this second spline on the existing plot. **[5]**

(c) Quote and compare the root mean squared errors (RMSE) for the P-splines obtained in (a) and (b).

- Provide the R command(s) you used.
- Quote the values you obtained.
- Indicate, with reason, which approach seems better on the basis of RMSE performance. **[5]**

(d) Generate predicted mortality rates for ages 80 to 89, for each of the two smoothing splines of (a) and (b). Using the test sample `dat.test`, compute the corresponding prediction RMSE's and comment on the difference observed.

- Provide the R command(s) you used.
- Quote the values you obtained.
- Provide an explanation for this result.
- *Hint:* the help page for `predict.smooth.spline()` may be helpful. **[5]**

(e) Implement a grid search to determine the optimal value for argument `spar` in function `smooth.spline()`. Use a grid of 50 values between .01 and .40 (inclusive) for this parameter. Perform the grid search on the basis of the cross-validation criterion value returned by `smooth.spline()`.

- Provide all relevant R code.
- Quote the optimal values of the parameter and criterion found by grid search. **[10]**

Solution:

(a) See plot and R code below.

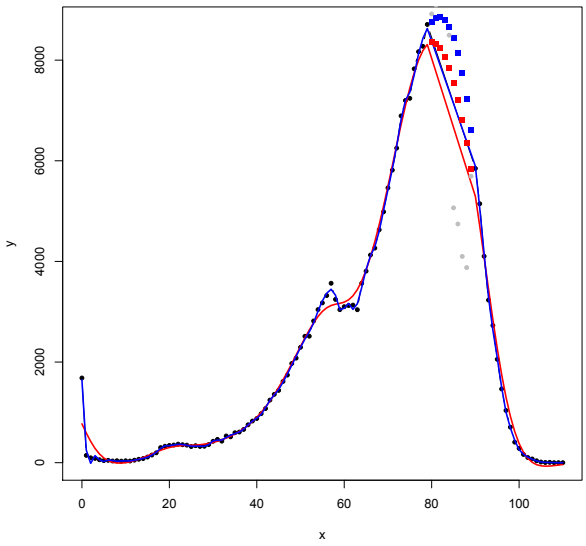
(b) See plot and R code below.

(c) RMSE for spline (a): **173.56**. RMSE for spline (b): **45.75**. RMSE (b) is clearly lower than RMSE (a), suggesting P-spline (b) is more appropriate.

(d)
- See code.
- Prediction RMSE for spline (a): **1778.495**. Prediction RMSE for spline (b): **2232.568**.
- Data points at ages 80-89 were actually not aligned to the data trend around this age bracket, with very large discrepancies. Interpolation at those ages using a better-fitting spline therefore results in worse estimates. (True data points can be added to the plot to illustrate/confirm this.)

(e) See code. Optimal value of `spar` from grid search: **0.1771429**. Corresponding value of criterion: **12337.69**.

Plot (with true data points for ages 80-89 in grey):



R code:

```
dat = read.csv(file='examdata_2018-19/FranceRates2004.csv')
dat.test = read.csv(file='examdata_2018-19/FranceRates2004_test.csv')
plot(dat$age, dat$D, pch=20, t='b')
x = dat$age
y = dat$D
points(dat.test$age, dat.test$D, pch=20, col=8)
# (a)
sp1 = smooth.spline(x, y, df=15)
plot(x, y, pch=20, t='b', xlim=c(0,110))
```

```
lines(sp1, lwd=2, col=2)
# (b)
sp2 = smooth.spline(x, y, spar=.05)
lines(sp2, lwd=2, col=4)
# (c)
names(sp2)
sqrt(mean((sp1$y-y)^2))
sqrt(mean((sp2$y-y)^2))
# (d)
xt = dat.test$age
yt = dat.test$D
p1 = predict(sp1, x=xt)
p2 = predict(sp2, x=xt)
points(xt,p1$y,pch=15,col=2)
points(xt,p2$y,pch=15,col=4)
# points(xt,yt,pch=20,col=8)
sqrt(mean((yt-p1$y)^2))
sqrt(mean((yt-p2$y)^2))
# (e)
L = 50
scrit = sval = seq(.01,.40,length=L)
for(i in 1:L)
spi = smooth.spline(x, y, spar=sval[i])
names(spi)
scrit[i] = spi$cv.crit

plot(sval,scrit)
abline(v=sval[which.min(scrit)])
```

Question 3 [30 marks]

Please run the R instruction `set.seed(4060)` *before* you run the rest of your R script, and again *each time* you re-run the script.

(a) Implement a Monte Carlo simulation of $M = 1,000$ random samples of observations, each following the same model

$$Y_i = \theta^* X_i + Z_i, \qquad i = 1, \ldots, n$$

with $n = 50$, $\theta^* = 4$ and a sequence of i.i.d. realizations $Z_i \overset{iid}{\sim} \chi^2_d$ with $d = 3$ degrees of freedom. To generate the noise you can use `R` instruction `z = rchisq(n, df=3)`. Note that all M Monte Carlo samples must be generated using the same sample $X \sim \mathcal{U}(1, 10)$, i.e. use `R` instruction `x = runif(n, 1, 10)` only once.
For each Monte Carlo sample, store the corresponding ordinary least squares estimator of θ^* (using `lm()`). Provide all R code relevant to the implementation of this simulation. **[10]**

(b) Quote the mean and standard error values for the ordinary least squares estimator of θ^*. **[5]**

(c) Explain what could have a negative impact on the estimation of θ^* in this analysis. **[5]**

(d) What is the theoretic expected value of $E(Y)$? *Hint:* you may recall that $E(Z) = d$, the number of degrees of freedom for the χ^2 distribution. **[5]**

(e) Quote the Monte Carlo sample mean estimates of $E(Y)$ and $E(Z)$ obtained from your implementation. **[5]**

Solution:

(a) See R code

(b) $\bar{\hat{\theta}} = \mathbf{4.435802}$; $\overline{SE(\hat{\theta})} = \mathbf{0.05503931}$.

(c) Small sample size could be mentioned, but it really is the fact that the noise is heavy tailed and skewed positively. The fact that it makes the observations larger than the true data is not necessarily an issue when estimating the slope; however the possibility of outliers due in the sample, to this skewness, may impact this estimation.

(d) $E(Y) = E(\theta^* X) + E(Z) = \theta^* E(X) + E(Z) = 4 \times 5.5 + 3 = \mathbf{25}$

(e) `mean(my)` = **25.68029** and `mean(mz)` = **2.990953**.

`R` code:

```
set.seed(4060)
N = 50
dfx = 3
thbar = 4
x = runif(N,1,10)
M = 1000
```

```
lmos = my = mz = numeric(M)
for(i in 1:M)
z = rchisq(n=N, df=dfx)
y = thbar*x + z
my[i] = mean(y)
mz[i] = mean(z)
lmo = lm(y~x+0)
lmos[i] = as.numeric(coef(lmo))

hist(z, col=8)
plot(x, y, pch=20)
points(x, thbar*x, pch=15, col=2)
#
mean(lmos)
sd(lmos)
#
mean(x)
mean(my)
mean(mz)
```

Question 4 [25 marks]

In this question we analyse the linear regression of stopping distance `dist` with respect to car speed `speed` based on `R`'s dataset `cars`. In particular we focus on the variability associated with the estimation procedure `lm(dist~speed, data=cars)`.

(a) Implement 10-fold cross-validation of the standard error of the slope estimate in this linear regression. Please run the R instruction `set.seed(1)` *before* you run the loop.

- Provide all relevant R code.
- Quote the mean (cross-validation) standard error estimate for the regression slope parameter. **[10]**

(b) Implement 10 bootstrap estimates of this same standard error, *in a separate loop*. Please run the R instruction `set.seed(1)` *before* you run the loop.

- Provide all relevant R code.
- Quote the mean (bootstrap) standard error estimate for the regression slope parameter. **[10]**

(c) Compare the sampling distributions of the cross-validation and bootstrap estimates using an appropriate boxplot and a t-test.

- Provide all relevant R code.
- Provide the boxplot and t-test output.
- Explain your findings.

[5]

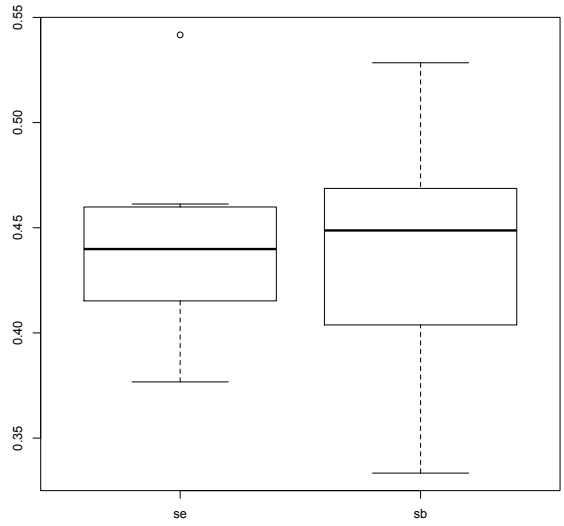
Solution:

(a) Average CV estimate of the slope parameter: **0.4423249**.

(b) Average bootstrap estimate of the slope parameter: **0.4357713**.

(c) See graph - no significant difference observed between the two mean estimates. We cannot infer a significant difference in SE estimates based on the t-test either (p=0.7723).

Boxplot:



R code:

```
plot(cars, pch=20)
abline(lm(dist~speed, cars), lwd=2, col=2)
x = cars$speed
y = cars$dist
N = nrow(cars)
K = 10
folds = cut(1:N,K,labels=FALSE)
p1 = se = sb = numeric(K)
set.seed(1)
for(i in 1:K)
# CV
itrain = which(folds!=i)
lmo = lm(y[itrain]~x[itrain])
se[i] = summary(lmo)$coef[2,2]

set.seed(1)
for(i in 1:K)
# bootstrapping
ib = sample(1:N,N,replace=TRUE)
lmb = lm(y[ib]~x[ib])
sb[i] = summary(lmb)$coef[2,2]

mean(se)
mean(sb)
boxplot(cbind(se,sb))
t.test(se,sb)
```

PLEASE DO NOT TURN THIS PAGE UNTIL INSTRUCTED TO DO SO

THEN ENSURE THAT YOU HAVE THE CORRECT EXAM PAPER

OLLSCOIL NA hÉIREANN, CORCAIGH
THE NATIONAL UNIVERSITY OF IRELAND, CORK

COLÁISTE NA hOLLSCOILE, CORCAIGH
UNIVERSITY COLLEGE, CORK

Examination Session and Year	Winter 2019 – Semester 1
Module Code	ST4060, ST6015, ST6040
Module Name	Statistical Methods for Machine Learning I Computer Analytical Techniques for Actuarial Applications Machine Learning and Statistical Analytics I
Paper Number	Paper Number: 1
External Examiner	Dr. Ji Yao
Head of Department	Dr. Michael Cronin
Internal Examiner(s)	Dr. Eric Wolsztynski
Instructions to Candidates	• Please answer all questions. • Provide all your answers in the Word document. • Paste your R code into the Word document at the end of each question.
Duration of Paper	3 hours.
Special Requirements	15 minutes Reading Time.

PLEASE DO NOT TURN THIS PAGE UNTIL INSTRUCTED TO DO SO

THEN ENSURE THAT YOU HAVE THE CORRECT EXAM PAPER

List of (possibly) useful R functions:

```
abline()
apply()
boxplot()
cbind()
coef()
dgamma()
fitted()
glmnet()
lines()
lm()
matrix()
nls()
nrow()
par()
plot()
points()
predict()
quantile()
read.csv()
rgamma()
round()
runif()
sample()
set.seed()
seq()
sum()
summary()
t.test()
which()
```

Question 1 [10 marks]

No code is required for this question.

Consider the following spline regression model used to fit a sample of observations Y with $\alpha_k \in \mathbb{R}$, $\beta_k \in \mathbb{R}$, $\forall k$, at given design points $\mathbf{X}$ and for a set of values $\{\xi\}$:

$$S(X) = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \sum_{j=1}^{10} \alpha_j (X - \xi_j)_+^3$$

using notation $(u)_+ = \max(u, 0)$.

(a) Indicate the order of this spline. **[2]**

(b) Indicate the number of internal knots used to fit this spline. **[4]**

(c) Once the values $\{\xi\}$ are determined, which estimation procedure can be used to fit this model to the data? **[4]**

Solution:

(a) This is a cubic spline.

(b) There are 10 internal knots.

(c) This is a linear regression problem, and Least Squares are usually used.

Question 2 [40 marks]

Load dataset `x1x2y.csv` into R using the following instruction:

```
dat = read.csv(file="examdata_2019-20/x1x2y.csv")
x1 = dat$x1
x2 = dat$x2
y  = dat$y
```

(a) Create a figure containing the following two plots:

(i) a set of boxplots showing the distributions of `x1`, `x2` and `y` respectively; **[3]**

(ii) a scatterplot of `x1` and `x2`, using full black dots to represent data points, and using the values in `y` as dot size. **[3]**

(b) Inspect the relationship between `y` and each of `x1` and `x2` as follows:

(i) Provide simple graphical representations of these relationships (using a maximum of 2 graphs). **[3]**

(ii) Comment on these graphs. **[3]**

(c) Fit a linear regression model to the observations `y`, using both `x1` and `x2` as unscaled predictors in the multivariate model, and using the whole dataset (i.e. without splitting the dataset). Quote the summary for this model fit. **[3]**

(d) Fit a nonlinear model of the form

$$Y = a + bX_1 + exp(-cX_2)$$

to the data, using `nls()` for optimisation and with initial parameter values `list(a=0, b=1, c=0.5)`.

(i) Quote the summary for this model fit. **[3]**

(ii) Comment on this output, especially with respect to the values obtained for the model coefficients, and on how they differ from those obtained for the linear regression model. **[3]**

(e) (i) Quote the residual sums of squares for the linear and nonlinear regression models obtained from (c) and (d). **[3]**

(ii) Comment on the percentage difference between these two values, and indicate which model you would rather use, and why. **[3]**

(f) Fit the LASSO model (using `library(glmnet)`) with regularization parameter 0.1 to the observations `y`, using both predictors `x1` and `x2`. Quote the coefficient estimates. **[3]**

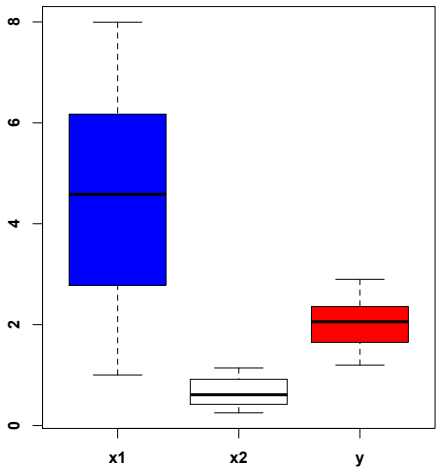
(g) Comment on the output of (f), and explain this output with respect to the plots obtained in (a) and (b).
If you did not manage to answer this previous question item, indicate what you expect to find in the LASSO output. **[5]**

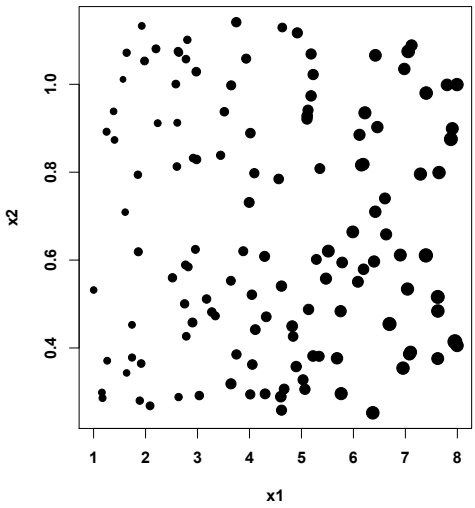
(h) Perform ridge regression with regularization parameter lambda=0.1 to the observations `y`, using both `x1` and `x2` as predictors. Quote the coefficient estimates. **[3]**

(i) Quote the percentage difference between the coefficient estimates obtained in (h) and those obtained for the linear regression model obtained in (c).
If you did not manage to answer this previous question item, provide the R instruction you would have used to obtain this result. **[2]**

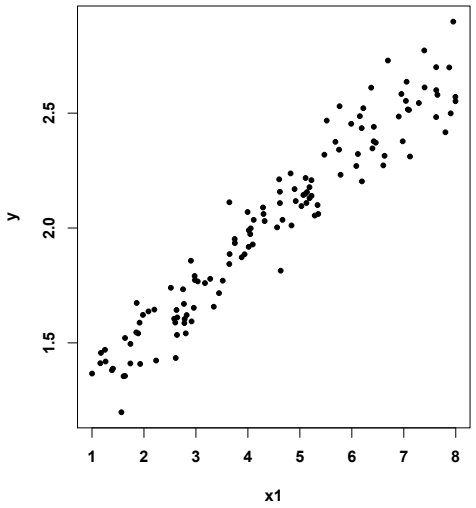
Solution:

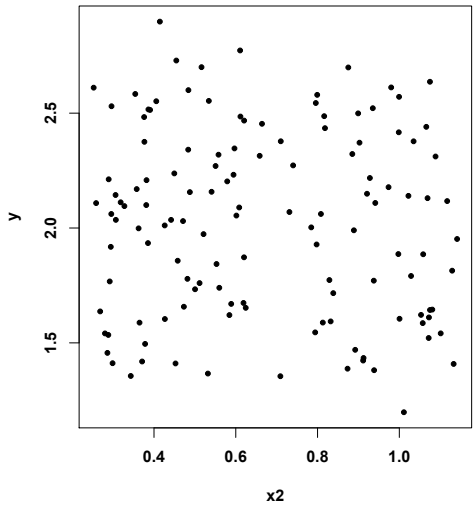
(a) Boxplots:





(b) Scatterplots: linear in x1, not clear whether y and x2 are related at all.





(c) Linear regression summary:

```
(Intercept)  1.229156   0.031943  38.480  < 2e-16 ***
x1           0.196014   0.004655  42.112  < 2e-16 ***
x2          -0.126934   0.033682  -3.769 0.000252 ***
```

(d) (i) Nonlinear regression summary:

```
  Estimate Std. Error t value Pr(>|t|)
a 0.232871   0.033343   6.984  1.5e-10 ***
b 0.196058   0.004654  42.127  < 2e-16 ***
c 0.140145   0.040997   3.418 0.000851 ***
```

(ii) Comments:

- intercept is different to adjust to this different model shape;
- linear coefficient estimate is very close to that of the linear regression model, hence capturing the same parameter effect (the nonlinear component not being dominant here);
- a positive coefficient for the effect of X_2 is found here, because the model shape translates this effect into a negative contribution by construction.

(e)
- RSS linear model: **1.3553**, RSS nonlinear model: **1.3549**
- %-difference: **0.0003**. There is no difference in RSS. Both models seem to represent the data similarly; we should therefore opt for the simpler linear regression model for easier interpretation.

(f) LASSO output:

```
(Intercept) 1.3712932
x1          0.1453618
x2          .
```

(g) Variable X_2 has been muted. Its contribution is deemed much less important in explaining Y than that from the linear term. This is consistent with the plots obtained in (a) and (b), which both show that most of the variability in Y is captured in the X_1 direction.

(h) Ridge regression coefficient estimates:

```
(Intercept)  1.3864093
x1           0.1569827
x2          -0.1006288
```

(i) %-difference of about 20% between the 2 sets of estimates:

```
(Intercept)  0.1279359
x1          -0.1991266
x2          -0.2072344
```

`R` code and comments:

```
# (a) Create a figure containing the following two plots:
par(mfrow=c(1,2), font=2, font.axis=2, font.lab=2)
# (i)
boxplot(x1x2y, col=c('blue','white','red'))
# (ii)
plot(x1, x2, pch=20, cex=y)

# (b)
par(mfrow=c(1,2), font=2, font.axis=2, font.lab=2)
plot(x1, y, pch=20)
plot(x2, y, pch=20)
# Linear in x1, not clear whether y and x2 are related at all.

# (c)
lmo = lm(y~x1+x2)
summary(lmo)

# (d)
nlmo = nls(y~a+b*x1+exp(-c*x2), start=list(a=0,b=1,c=.1))
summary(nlmo)

# (e)
# (i)
rss.lm = sum(lmo$residuals^2)
rss.nlm = sum(residuals(nlmo)^2)
# (ii)
round(c(rss.lm, rss.nlm, (rss.lm-rss.nlm)/rss.lm), 4)

# (f)
library(glmnet)
lasso = glmnet(cbind(x1,x2), y, alpha=1, lambda=.1)
coef(lasso)

# (g)

# (h)
ridge = glmnet(cbind(x1,x2), y, alpha=0, lambda=.1)

# (i)
(coef(ridge)-coef(lmo))/coef(lmo)
```

Question 3 [25 marks]

(a) Create a uniform grid of 1,000 values ranging between 0 and 10. Generate and plot the curve of the Gamma distribution $\mathcal{G}(a, b)$ with shape $a = 3$ and rate $b = 2$ evaluated at these grid points. **[2]**

(b) Implement a Monte Carlo simulation of $M = 1,000$ random samples of N observations, for N successively in $\{50, 100, 500\}$, of realizations of the Gamma distribution $\mathcal{G}(a, b)$ with shape $a = 3$ and rate $b = 2$. Use `set.seed(1)` before running the whole simulation.

(i) For each sample size, calculate the M Monte Carlo sample means, and quote the Monte Carlo sample mean estimate of the distribution mean. **[4]**

(ii) Provide a plot showing the three boxplots of the Monte Carlo distribution of sample means for each sample size, with careful labelling of the axes. Comment on this figure. **[4]**

(c) Implement a Monte Carlo simulation of $M = 1,000$ random samples of observations, each sample following the same model

$$Y_i = \theta^* X_i + Z_i, \qquad i = 1, \ldots, N$$

using:

- `set.seed(1)` before running the whole simulation,
- $N = 100$ and $\theta^* = 4$,
- for each Monte Carlo repetition, a random sample $X_i \sim \mathcal{U}(-5, 5)$,
- for each Monte Carlo repetition, a sequence of i.i.d. realizations $Z_i \sim \mathcal{G}(a, b)$ with shape $a = 3$ and rate $b = 2$.

(i) For each Monte Carlo sample, perform two estimations, one fitting the model

$$Y_i = \theta^* X_i + Z_i, \qquad i = 1, \ldots, N$$

to the data (X, Y) and another one fitting the model

$$Y_i = \theta_0^* + \theta_1^* X_i + Z_i, \qquad i = 1, \ldots, N$$

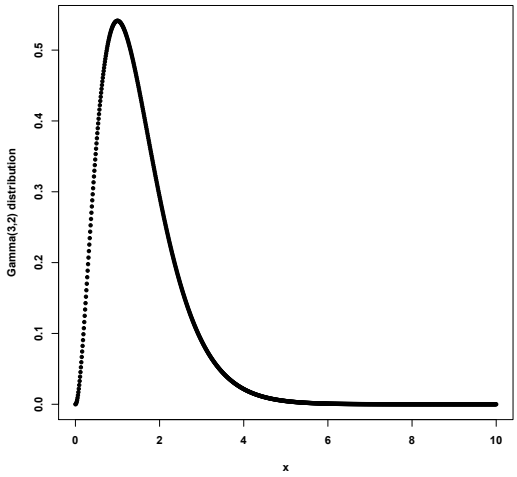
to the data (X, Y). Quote the nonparametric 95% confidence intervals for the ordinary least squares estimators of θ^*, θ_0^* and θ_1^*. **[6]**

(ii) What is the theoretic expected value of Y? Justify your answer. **[4]**

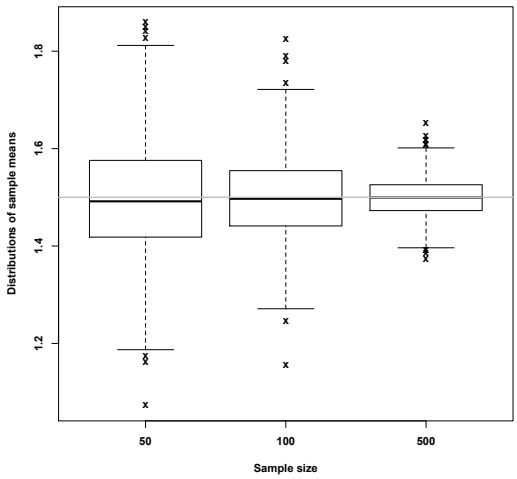
(iii) Plot the boxplots of Monte Carlo distributions of estimates of θ^*, θ_1^* and θ_2^* in one graph, and add a horizontal line at value 4 on the Y-axis. Explain the difference, if any, between median estimates for θ^* and θ_2^*.
Hint: recall that for the 2-parameter linear regression model, we have $\hat{\theta}_0^* = \bar{Y} - \hat{\theta}_1^* \bar{X}$ *and, using* $\tilde{X} = Y - \bar{X}$ *and* $\tilde{Y} = Y - \bar{Y}$, $\hat{\theta}_1^* = (\tilde{X}^T \tilde{X})^{-1} \tilde{X}^T \tilde{Y}$. **[5]**

Solution:

(a) Theoretic Gamma distribution:



(b) Monte Carlo estimate of sample means wrt sample size (comments: CLT in action; converge to the theoretic average $a/b = 1.5$):
(1.504, 1.496, 1.500)



(c) (i) Monte Carlo 95% CI's:

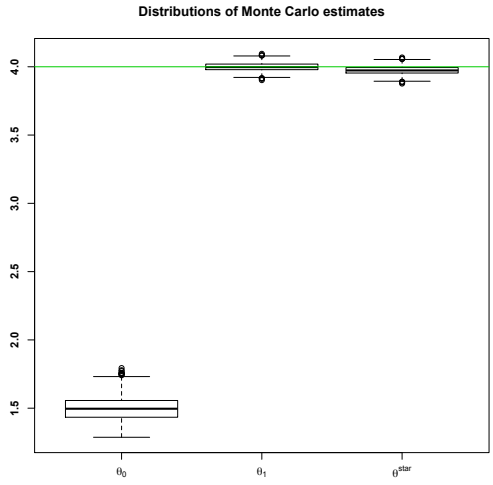
$$\theta^* = (\mathbf{3.881},\ \mathbf{4.112})$$

and

$$\theta_0^* = (\mathbf{1.330},\ \mathbf{1.670}), \quad \theta_1^* = (\mathbf{3.941},\ \mathbf{4.061})$$

(ii) $E(Y) = E(\theta^* X) + E(Z) = \theta^* E(X) + E(Z) = 4 \times 0 + a/b = \mathbf{1.5}$

(iii) Monte Carlo distributions:



For the 2-parameter linear regression model, we have

$$\hat{\theta}_0^* = \bar{Y} - \hat{\theta}_1^* \bar{X} = \bar{Y}$$

and, using $\tilde{X} = Y - \bar{X}$ and $\tilde{Y} = Y - \bar{Y}$,

$$\hat{\theta}_1^* = (\tilde{X}^T \tilde{X})^{-1} \tilde{X}^T \tilde{Y}$$

Here, adding θ_0^* to the model helps capture the mean of Y, which is necessary because the additive noise term has a non-zero mean (and $E(Y) = E(Z)$). This helps reducing finite bias when estimating the slope coefficient.

R code:

```
### (a)
a = 3 # shape
b = 2 # rate
x = seq(0,10,length=1000)
par(font=2, font.axis=2, font.lab=2)
plot(x, dgamma(x, shape=a, rate=b), pch=20, ylab="Gamma(3,2) distribution")

### (b)
set.seed(4060)
M = 1000
N = c(50,100,500)
means = NULL
for(n in N){
    z = matrix(rgamma(n*M, shape=a, rate=b), nrow=M, ncol=n)
    means = cbind(means, apply(z, 1, mean))
}
summary(means)
par(font=2, font.lab=2, font.axis=2)
boxplot(means, names=N, pch='x', xlab="Sample size",
    ylab="Distributions of sample means")
abline(h=a/b, lwd=2, col=8)
```

```
### (c)
M = 1000
N = 100
coefs0 = matrix(NA, nrow=M, ncol=1)
coefs = matrix(NA, nrow=M, ncol=2)
set.seed(4060)
x = runif(N, -5, 5)
for(m in 1:M){
    y = 4*x + rgamma(N, shape=a, rate=b)
    coefs0[m] = coef(lm(y~x+0))
    coefs[m,] = coef(lm(y~x))
}
c(apply(coefs0,2,quantile,.025), apply(coefs0,2,quantile,.975))
cbind(apply(coefs,2,quantile,.025), apply(coefs,2,quantile,.975))

par(font=2, font.axis=2, font.lab=2)
boxplot(cbind(coefs, coefs0),
    main="Distributions of Monte Carlo estimates",
    names=c(expression(theta[0]), expression(theta[1]),
    expression(theta^star)))
abline(h=4, col=3)
```

Question 4 [25 marks]

In this question we analyse the linear regression of tree height (`Height`, in feet) with respect to tree diameter (`Girth`, in inches) based on `R`'s dataset `trees` of 31 felled black cherry trees. In particular we focus on the variability associated with the estimation procedure `lm(Height~Girth, data=trees)`.

(a) Implement 10-fold cross-validation of the standard error of the slope estimate in this linear regression. Please run the R instruction `set.seed(4060)` *before* you perform any other action for this cross-validation analysis.

- Provide all relevant R code.
- Quote the cross-validated estimate for the regression slope parameter. **[10]**

(b) Implement 100 bootstrap estimates of this same standard error, *in a separate loop*. Please run the R instruction `set.seed(4060)` *before* you perform any other action for this bootstrapping analysis.

- Provide all relevant R code.
- Quote the bootstrap estimate for the regression slope parameter. **[10]**

(c) Compare the sampling distributions of the cross-validation and bootstrap estimates of standard errors using an appropriate boxplot. Perform a two-sided, two-sample t-test to compare these sampling distributions. Provide the boxplot and t-test output, and explain your findings. **[5]**

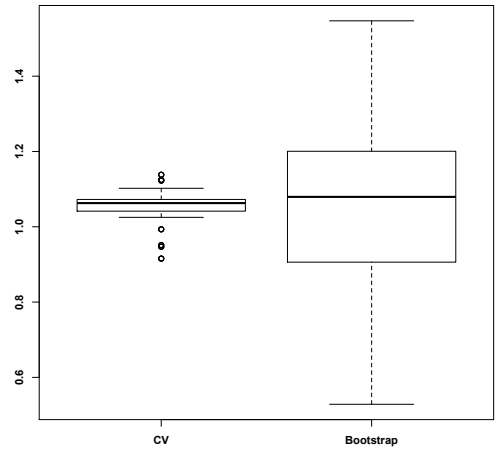
Solution:

(a) Average CV estimate of the slope parameter: **1.053393**.

(b) Average bootstrap estimate of the slope parameter: **1.064388**.

(c) See graph - despite an important difference in estimation variances, there is not a clearly significant difference observed between the two estimates on average. We cannot infer a significant difference in estimates based on the t-test (p=0.9802).

Boxplot:



R code:

```
x = trees$Girth
y = trees$Height
summary(lm(y~x))

N = nrow(trees)
cc = numeric(N)
set.seed(4060)
for(i in 1:N){
# CV
lmo = lm(y[-i]~x[-i])
cc[i] = summary(lmo)$coef[2,1]
}
mean(cc)

set.seed(4060)
K = N
cb = numeric(K)
for(i in 1:K){
# bootstrapping
ib = sample(1:N,N,replace=TRUE)
lmb = lm(y[ib]~x[ib])
cb[i] = summary(lmb)$coef[2,1]
}
mean(cb)

boxplot(cbind(cc,cb))
t.test(cc,cb)
```

PLEASE DO NOT TURN THIS PAGE UNTIL INSTRUCTED TO DO SO

THEN ENSURE THAT YOU HAVE THE CORRECT EXAM PAPER

OLLSCOIL NA hÉIREANN, CORCAIGH
THE NATIONAL UNIVERSITY OF IRELAND, CORK

COLÁISTE NA hOLLSCOILE, CORCAIGH
UNIVERSITY COLLEGE, CORK

Examination Session and Year	Winter 2020 – Semester 1
Module Code	ST4060, ST6015, ST6040
Module Name	Statistical Methods for Machine Learning I Computer Analytical Techniques for Actuarial Applications Machine Learning and Statistical Analytics I
Paper Number	Paper Number: 1
External Examiner	Mr Andrew Maclaren
Head of School	Dr. Kevin Hayes
Internal Examiner(s)	Dr. Eric Wolsztynski
Instructions to Candidates	• Answer all four questions. • Provide all your answers in the Word document. • Paste your R code into the Word document where indicated. • Upload a pdf version of the Word document as your final submission.
Duration of Paper	3 hours.
Special Requirements	15 minutes Reading Time.

List of (possibly) useful R functions:

```
abline()
apply()
boxplot()
cbind()
coef()
dgamma()
fitted()
glmnet()
lines()
lm()
matrix()
nls()
nrow()
par()
plot()
points()
predict()
quantile()
read.csv()
rgamma()
round()
runif()
sample()
set.seed()
seq()
sum()
summary()
t.test()
which()
```

Question 1 [20 marks]

No code is required for this question.

(a) Consider an i.i.d. sample $\{X_1, \ldots, X_N\}$ and a non-parametric estimate $\hat{f}$ of its probability density function f defined for any $u \in \mathbb{R}$ and some real constant $h > 0$ by

$$\hat{f}(u) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{X_i - u}{h}\right)$$

(i) Can $K(u) = \exp(-\frac{u^2}{2})$ be used to compute this estimate? Why? **[4]**

(ii) What is the standard deviation of function $K(u)$? **[4]**

(iii) What is the standard deviation of function $K_h(u) = K(u/h)/h$? **[4]**

(iv) In order to ensure a finite-sample estimate $\hat{f}$ of f with as small a bias as possible, and using the unbiased sample variance estimate $\hat{\sigma}^2$ of $Var(X)$, indicate which of the following values of h should be used and why:

$$h_1 = 1.06\ \hat{\sigma}\ N^{-\frac{1}{5}}$$
$$h_2 = 2.34\ \hat{\sigma}\ N^{-\frac{1}{5}}$$

Note: no marks awarded if no explanation is provided. **[4]**

(b) Consider a training dataset $\{(X_i, Y_i)\}_{i=1}^{N}$ and the following two estimators of the functional relationship $Y = \tilde{g}(X) + \varepsilon$ between predictors X and observations Y, for some unknown function $\tilde{g}$, i.i.d. additive noise ε and real constant $h > 0$:

$$\hat{g}_1 = \arg\min_{g} \sum_{i=1}^{N} (Y_i - g(X_i))^2$$
$$\hat{g}_2 = \arg\min_{g} \sum_{i=1}^{N} (Y_i - g(X_i))^2 + h \int_{u_{min}}^{u_{max}} \left(g''(u)\right)^2 du$$

(where $g''(u)$ denotes the second-order derivative of $g(u)$ with respect to u). Indicate which of these two estimators provides the better fit on the *training data* and why.
Note: no marks awarded if no explanation is provided. **[4]**

Question 2 [20 marks]

Start by running the following commands in your R session:

```
library(ISLR)
dat = na.omit(Hitters)
x = dat[,c("Years","Hits","Walks")]
y = log(dat$Salary)
n = nrow(x)
```

(a) Implement 5-fold cross-validation of the multivariate linear regression of log-salaries Y on the set of predictors X. Quote the prediction MSE estimate obtained from 5-fold CV. Run the R instruction `set.seed(4060)` *before* you perform any other action for this cross-validation analysis. **[5]**

(b) Implement leave-one-out cross-validation of the multivariate linear regression of log-salaries Y on the set of predictors X. Quote the prediction MSE estimate obtained from LOO-CV. Run the R instruction `set.seed(4060)` *before* you perform any other action for this cross-validation analysis. **[5]**

(c) Provide a single figure showing the boxplots corresponding to the test-set MSEs obtained from each of the above cross-validation frameworks. **[5]**

(d) Comment on the boxplots, indicate whether they are conform to your expectations, and why. **[5]**

Question 3 [20 marks]

Start by running the following commands in your R session:

```
dat = read.csv(file="nonlinear_dataset.csv")
x = dat$x
y = dat$y
```

This dataset contains a simulated sample of $N = 125$ data points (X, Y) with a nonlinear relationship between X and Y.

(a) Fit the following two nonlinear models to the data using `nls()`, both times with starting values $a = 0.05, b = 0.4, c = 2$:

$$Y_i = aX_i^2 + \sin(b + cX_i) + \varepsilon_i \quad (1)$$

$$Y_i = aX_i^2 + bX_i + c + \varepsilon_i \quad (2)$$

Quote the coefficient estimates and Root Mean Square Errors obtained for both models. **[4]**

(b) Provide a scatterplot of the data (X, Y) (as black dots), also showing both model fits obtained from (a), respectively in blue and red for models (1) and (2), over the data. **[4]**

(c) Bootstrap the model fitting procedure for nonlinear regression model (2), using $B = 100$ bootstrap resamples. Run the R instruction `set.seed(4060)` *before* you perform bootstrapping. Quote the final bootstrap estimates of model parameters a, b and c. **[4]**

(d) Using (c), quote the bootstrap estimate of the standard error associated with the estimator used within `nls()` to fit regression model (2). For this question, *assume that an evaluation of standard error was not available theoretically nor from* `nls()`. **[4]**

(e) Using (a) and (c), quote the 95% bootstrap confidence interval for the model parameters of regression model (2). **[4]**

Question 4 [40 marks]

Import the simulated dataset of female mortality rates from a hypothetical cohort of life insurance policyholders into R as follows:

```
dat = read.csv("insdata.csv")
age = dat$Age
mF = dat$mF
```

(a) Compute a first P-spline where the smoothing control parameter is set to 0.5. Provide a plot of the dataset (black dots) along with the P-spline (red solid curve). **[1]**

(b) Compute a second P-spline for the same dataset, where the smoothing control parameter is half that of the P-spline obtained in (a). Add this second spline, as a blue solid curve, to the plot of (a). **[1]**

(c) Show that the two P-spline outputs are evaluated over the same points on the x-axis. **[2]**

(d) Compute and compare the MSEs for the P-splines obtained in (a) and (b). Comment on their difference, and propose a reason as to why they differ. **[8]**

(e) Compute a B-spline basis using the first, second and third quartiles of the age data as knots. Provide a plot of this B-spline basis. **[4]**

(f) Quote the coordinates of a policyholder aged 60 on the B-spline basis computed in (e), up to four decimal places. Indicate these coordinates with a line on the plot obtained in (e). **[3]**

(g) Compute the corresponding B-spline for the (age, mF) data. Provide the output coefficients for the B-spline expression. **[3]**

(h) Compare and comment on the MSE obtained for that B-spline with the MSEs obtained from the two P-splines obtained in (d). **[8]**

(i) Compute interpolations for all ages within the range of age data, using respectively P-spline smoothing and local polynomial regression. Plot the interpolated points over the observations, using red for P-spline values and blue for local polynomial regression values. **[6]**

(j) Quote the standard deviations of each of the interpolated samples. **[4]**

OLLSCOIL NA hÉIREANN, CORCAIGH
THE NATIONAL UNIVERSITY OF IRELAND, CORK

COLÁISTE NA hOLLSCOILE, CORCAIGH
UNIVERSITY COLLEGE, CORK

Examination Session and Year	Winter 2021
Module Code	ST4060 ST6015 ST6040
Module Name	Statistical Methods for Machine Learning I Computer Analytical Techniques for Actuarial Applications Machine Learning and Statistical Analytics I
Paper Number	Paper Number: 1
External Examiner	Mr. Andrew Maclaren
Head of Department	Dr. Kevin Hayes
Internal Examiner(s)	Dr. Eric Wolsztynski
Instructions to Candidates	• Please answer all questions. • Provide all your answers in the Word document provided. • Paste your R code into the Word document at the end of each question. • Submit a pdf version of your final Word document for Canvas submission.
Duration of Paper	3 hours.

List of required R libraries:

```
ISLR
```

List of (possibly) useful R functions:

```
apply()
approx()
as.numeric()
boxplot()
cbind()
coef()
colnames()
fitted()
lines()
lm()
matrix()
mean()
median()
na.omit()
nls()
nrow()
numeric()
order()
par()
plot()
points()
predict()
quantile()
sample()
sd()
set.seed()
smooth.spline()
sqrt()
sum()
summary()
which()
```

Question 1 [15 marks]

No code is required for this question.

Let $S_M(X)$ denote a cubic spline evaluated at some d-dimensional design point $\mathbf{X} \in \Omega \subset \mathbb{R}^d$, and for a set of values $\{\xi_i\}_{i=1}^M \in \Omega$, with $\alpha_k \in \mathbb{R}$, $\beta_k \in \mathbb{R}$, $\forall k$:

$$S_M(X) = \sum_{k=0}^{3} \beta_k X^k + \sum_{j=1}^{M} \alpha_j (X - \xi_j)_+^3$$

using notation $(u)_+ = \max(u, 0)$. Let us now define the following criterion $C_\lambda(g)$ as a function of some continuous function $g(X)$, with the aim to fit a sample of N observations $\{Y_i\}_{i=1}^N$, using some parameter $\lambda \in \mathbb{R}^+$:

$$C_\lambda(g) = \sum_{i=1}^{N} (Y_i - g(X_i))^2 + \lambda \int_\Omega \left(g''(u)\right)^2 du$$

(a) Name the function estimate $\hat{g}$ that minimizes the above criterion $C_\lambda(g)$.

(b) Using the information given above, provide a name for the function $\hat{g}$ that minimizes $C_\lambda(g)$, and indicate how this function should be evaluated given the data (X, Y) for this problem.

(c) Describe, in a few sentences, the effect of parameter λ on the estimate $\hat{g}$.

Solution:

(a) This is a penalized spline.

(b) The solution to this optimisation problem is a cubic spline, i.e. $\hat{g}(X) = S_M(X)$, evaluated at the design points $\xi_i = X_i$.

(c) A larger value of λ will place more emphasis on the penalty term, which will gradually become more important over the standard Least Squares terms. Since this penalty evaluates the magnitude of the second-order derivative of the spline $\hat{g}$, a larger value of λ will yield a smoother spline, i.e. a smoother estimate of the function that links the observations Y to the design points X.

Question 2 [30 marks]

No code is required for this question.

A data analyst is implementing a Monte Carlo simulation of $M = 1,000$ random samples of realisations of the model

$$Y_i = \theta^* X_i + Z_i, \qquad i = 1, \ldots, n \tag{1}$$

with $n = 100$, $\theta^* = 8$ and a sequence of i.i.d. realizations $Z_i \overset{iid}{\sim} t_d$ with $d = 3$ degrees of freedom, using a single sample $\{X_i\}_{i=1}^n$ from $X \sim \mathcal{U}(1, 2)$ to generate all M Monte Carlo samples, and computes and stores the Monte Carlo least squares estimates of θ^* for analysis. Note the analyst is making sure to not include an intercept in the regression model when fitting it to the simulated data.

(a) Quote the theoretic expected value of Y, i.e. the true value of $E(Y)$, showing your calculation.

(b) Quote the theoretic expected value of $\hat{\theta}$, i.e. the true value of $E(\hat{\theta})$, justifying your answer with a brief statement.

(c) Briefly describe in which way(s) the distribution of Monte Carlo estimates of $\theta^\star$ would differ from the one the analyst is generating, if the additive noise Z was such that $Z \sim \mathcal{N}(0, 1)$, and why (all other settings of the Monte Carlo experiment remaining the same).

(d) Briefly describe in which way(s) the distribution of Monte Carlo estimates of $\theta^\star$ would differ from the one the analyst is generating, if the additive noise Z remained $Z \sim t_d$ with $d = 3$, but using $M = 5,000$ Monte Carlo samples instead of $M = 1,000$, and why (all other settings of the Monte Carlo experiment remaining the same).

(e) The analyst included the instruction

```
shapiro.test(estimates)
```

at the end of the R code, where `estimates` is the vector she used to store the Monte Carlo estimates of $\theta^\star$. Explain what output you would expect from this test, assuming the settings described in (d) above were used for this analysis (i.e. with $M = 5,000$), and why. Provide any information about the test that is relevant to your answer. (No code required.)

(f) Suppose now that the analyst allowed for an intercept in the regression model, when fitting the latter to the Monte Carlo samples simulated correctly from equation (1) above. Briefly describe how this would impact the distribution of estimates of $\theta^\star$ obtained from the original simulation settings described at the beginning of this question, and why.

Solution:

(a) $E(Y) = E(\theta^* X) + E(Z) = \theta^* E(X) + E(Z) = 8 \times 1.5 + 0 = \mathbf{12}$.

(b) The OLS being consistent under this model, $E(\hat{\theta}) = \theta^* = \mathbf{8}$. The MC simulation demonstrates this consistency.

(c)
- The MC mean should remain comparable, since the OLS remains unbiased in this new setting with, again, 0-mean noise.

- The MC estimation variance should decrease, since there would no longer be any outliers in the MC resamples.

(d)
- There would be a larger proportion of outliers in (ie a heavier tail for) the sample of MC estimates because of the heavy-tailed distribution used for Z. However these would be more symmetrically distributed due to the LLN, i.e. the distribution of MC estimates would become more symmetric as M increases.
- Finite-sample variance could increase with M due to more outliers occurring at finite sample horizon, although this increase would be only slight if any, since M is already large enough for the MC distribution to be close to its limit.
- Bias should decrease since $E(\hat{\theta}_M) \to \theta^\star$ as $M \to \infty$.

(e) The p-value should be large enough under $H_0 : \hat{\theta}_M \sim \mathcal{N}(\theta^\star, Var(\hat{\theta}_M))$, since we expect the distribution of `estimates` to be approximately Normally distributed.

(f) Including an intercept in the model would likely increase finite-sample bias slightly, and increase estimation variance noticeably. Even though we'd expect the MC estimates of the intercept to be roughly 0 on average, they would have a non-zero value, and this would "mechanically" affect the accuracy of the estimation of $\theta^\star$.

Question 3 [20 marks]

Note: if you do not manage to answer a question item, provide the R code you would have used, or a comment on the answer you would expect for that question, as relevant.

Load the following library and dataset into your `R` session:

```
library(ISLR)
dat = na.omit(Hitters)
x = dat$Years
y = dat$Salary
```

(a) Fit a nonlinear model of the form

$$y = a + bx + cx^2 + \epsilon$$

to the above data. Quote:

- the coefficient estimates for this model;
- the root mean squared error RMSE corresponding to the model fit.

(b) Fit a cubic smoothing spline to the same data (`x`,`y`), using the default value for the number of degrees of freedom. Quote the RMSE corresponding to this spline model fit.

(c) Fit a second cubic smoothing spline to the same data, using a number of degrees of freedom 4 times higher than the default value used in (b). Quote the RMSE corresponding to this second spline model fit.

(d) Compare the three model RMSEs obtained above, end explain any difference you may observe.

Solution:
R code:

```
# (a.i)
nlo2 = nls(y~a+b*x+c*I(x^2), start=list(a=10,b=1,c=1))
summary(nlo2)
# (a.ii)
(rmse.nlo = sqrt( mean((y.hat-y)^2) ))
# (b)
so = smooth.spline(x,y)
y.so = approx(so$x,so$y,xout=x)$y
(rmse.so = sqrt( mean((y.so-y)^2) ))
# (c)
so2 = smooth.spline(x,y,df=so$df*4)
y.so2 = approx(so2$x,so2$y,xout=x)$y
(rmse.so2 = sqrt( mean((y.so2-y)^2) ))
# (d)
(rmse.so-rmse.nlo)/rmse.nlo
(rmse.so2-rmse.nlo)/rmse.nlo
(rmse.so2-rmse.so)/rmse.so
```

```
# Bonus
plot(x,y)
is = order(x)
lines(x[is],fitted(nlo2)[is],lwd=2,col=2)
lines(so,col=4,lwd=2)
lines(so2,col='navy',lwd=2)
```

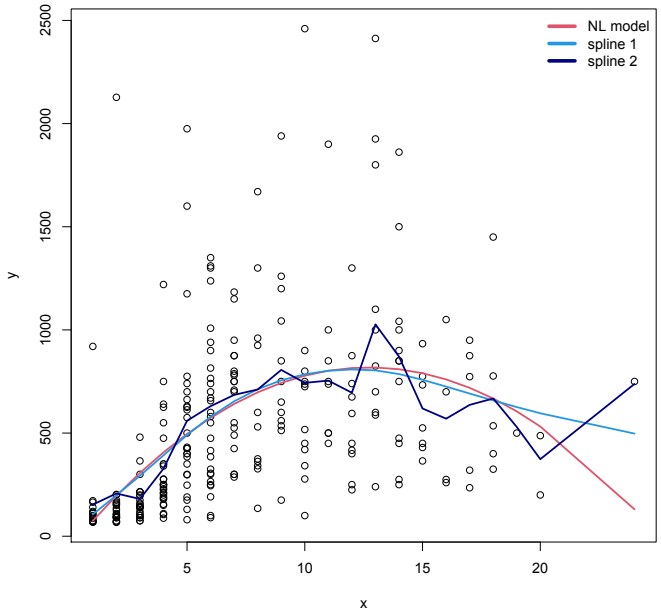
(a) Nonlinear model: a = **-57.442521**, b=**137.626725**, c=**-5.408112**

(b) RMSE of the quadratic model fit: **388.2181**

(c) RMSE of the first smoothing spline: **385.5049**

(d) RMSE of the second smoothing spline: **372.2388**

(e) There is a 0.7% difference between the RMSEs from the parametric model and the first spline, but a 3.4% difference between the RMSEs of the two splines (with a lower RMSE for the second spline). The second spline overfits the data because of its higher number of degrees of freedom, and this yields a reduction in RMSE. One could plot the graph too:



Question 4 [35 marks]

Note: if you do not manage to answer a question item, provide the `R` instruction you would have used, or a comment on the answer you would expect for that question, as relevant.

Load the following library and dataset into your `R` session:

```
library(ISLR)
dat = na.omit(Hitters)
itrain = c(1:200)
dat.train = dat[itrain,]
dat.test = dat[-itrain,]
Salary.test = dat.test$Salary
dat.test$Salary = NULL
```

Set the random seed to 1 (using `set.seed(1)`) before running your analysis. Bootstrap the effect of `HmRun` (the number of home runs in a season) on player salary `Salary`, when measured by univariate linear regression analysis of the training set `dat.train`, use 100 bootstrap resamples of the training set `dat.train`. Record also the corresponding bootstrapped p-values. It is up to you to decide whether an intercept parameter should be included or not.

(a) Run your bootstrap implementation, and:

- Quote the bootstrap mean estimate of the effect of variable `HmRun` on `Salary`.
- Name the quantity estimated in (a).

(b) Quote the 99^{th} percentile of bootstrapped p-values for variable `HmRun`.

(c) Comment on your results for (a) and (b).

(d) Predict the salaries of players in the test set `dat.test` using the means of bootstrap estimates of the univariate regression model parameters. Calculate and quote the root mean squared prediction error.

(e) Comment on your result for (d).

(f) Generate similar predictions for `Salary.test` from the linear regression model obtained by fitting the original dataset `dat.train` (i.e. without resampling). Quote the corresponding root mean squared error.

(g) Comment on your result for (f), in particular what it highlights about the use of bootstrapping to estimate the quantity named in (a).

Solution:
Code:

```
B = 100
set.seed(1)
int = pval = eff = numeric(B)
for(b in 1:B)
ib = sample(1:nrow(dat.train), nrow(dat.train), replace=TRUE)
xb = dat.train[ib,]
```

```
lmb = lm(Salary~HmRun, data=xb)
int[b] = summary(lmb)$coef[1,1]
eff[b] = summary(lmb)$coef[2,1]
pval[b] = summary(lmb)$coef[2,4]

# (a)
mean(eff)
# (c)
quantile(pval,.99)
# (e)
preds = mean(int) + mean(eff)*dat.test$HmRun
sqrt( mean((preds-Salary.test)^2) )
# (f)
mean(Salary.test)
sd(Salary.test)
# (g)
lmo = lm(Salary~HmRun, data=dat.train)
preds.lmo = predict(lmo,dat.test)
sqrt( mean((preds.lmo-Salary.test)^2) )
```

(a)
- Bootstrap mean effect: **18.79508**
- This is an estimate of the expected value of the OLS estimate of the linear parameter in the regression model; i.e. it is an estimate of $E\left[\hat{\theta}_1\right]$ in the regression model $Y = \theta_0 + \theta_1 X + \varepsilon$.

(b) 99-percentile of the p-value of this effect: **0.004532542**

(c) A unit increase in HmRun yields an increase in Salary by a factor of 18.795. This effect is statistically significant.

(d) Test set prediction RMSE from bootstrap model: **364.3579**

(e) The RMSE is very large compared to the mean Salary value (484.7646), and almost equal to its SD (365.7454). Therefore the prediction is not accurate. This variable alone is not sufficient in predicting Salary accurately.

(f) Test set prediction RMSE from original model: **364.9748**

(g) Not surprisingly, the original fit is comparable to the bootstrap estimates. Prediction RMSE is therefore also comparable. This is because bootstrapping provides an unbiased estimate of $E\left[\hat{\theta}_1\right]$.

OLLSCOIL NA hÉIREANN, CORCAIGH
THE NATIONAL UNIVERSITY OF IRELAND, CORK

COLÁISTE NA hOLLSCOILE, CORCAIGH
UNIVERSITY COLLEGE, CORK

Examination Session and Year	Winter 2022
Module Code	ST4060 ST6040
Module Name	Statistical Methods for Machine Learning I Machine Learning and Statistical Analytics I
Paper Number	Paper Number: 1
External Examiner	Mr. Andrew Maclaren
Head of School	Dr. Kevin Hayes
Internal Examiner(s)	Dr. Eric Wolsztynski
Instructions to Candidates	• Please answer all questions. • Provide all your answers in the Word document provided. • Paste your R code into the Word document at the end of each question. • Submit a pdf version of your final Word document for Canvas submission. Note: if you do not manage to answer a question item, provide the R code you would have used, or a comment on the answer you would expect for that question, as relevant.
Duration of Paper	3 hours

List of required R libraries:

```
glmnet
splines
```

List of (possibly) useful R functions:

```
apply()
approx()
as.numeric()
boxplot()
bs()
cbind()
coef()
colnames()
cor.test()
density()
fitted()
lines()
lm()
matrix()
mean()
median()
na.omit()
nls()
nrow()
numeric()
order()
par()
plot()
points()
predict()
quantile()
sample()
sd()
seq()
set.seed()
smooth.spline()
sqrt()
sum()
summary()
which()
```

Question 1 [15 marks]

Lucien is an analyst interested in the theoretic distribution of the sum of five Bernoulli random variables

$$X_i \overset{iid}{\sim} Bernoulli(p_i), \quad p_i = p \text{ (constant)}, \quad i = 1, \ldots, 5$$

To help him, implement a Monte Carlo simulation to evaluate the expected value and variance of this compound random variable

$$S = \sum_{i=1}^{5} X_i$$

Using $M = 1,000$ Monte Carlo resamples, each of size $N = 100$, compute Monte Carlo estimates of the mean and variance of S for all values of $p \in \{0.1, 0.2, \ldots, 0.9\}$ successively. Set the random seed to 4060 (`set.seed(4060)`) before running your analysis.

Note: one can generate a random sample of N realisations of a Bernoulli random variable with probability p in `R` as follows: `x = rbinom(N, size=1, prob=p)`.

(a) Quote your Monte Carlo estimate of the mean of S, for each value of p.

(b) Quote your Monte Carlo estimate of the variance of S, for each value of p.

(c) In a two-frame figure, with the two frames side by side, plot the boxplots of the Monte Carlo distributions of estimates of the means and variances of S, respectively, with respect to p.

(d) Based on the above results, briefly discuss whether S could be thought to follow a *Poisson binomial* distribution with $E(S) = \sum_{i=1}^{5} p_i = 5p$ and $Var(S) = \sum_{i=1}^{5}(1-p_i)p_i = 5p(1-p)$.

Solution:
R code:

```
# Monte Carlo illustration of Le Cam's Theorem
N = 100
p = .1
set.seed(4060)
M = 1000
means = vars = NULL
Ps = seq(.1,.9,by=.1)
for(p in Ps)
m = v = numeric(M)
for(i in 1:M)
X = matrix(rbinom(5*N,1,p),nrow=N,ncol=5)
S = apply(X,1,sum)
m[i] = mean(S)
v[i] = var(S)

means = cbind(means, m)
vars = cbind(vars, v)

boxplot(means, names=Ps, col='cyan')
abline(h=5*Ps, col=8)
```

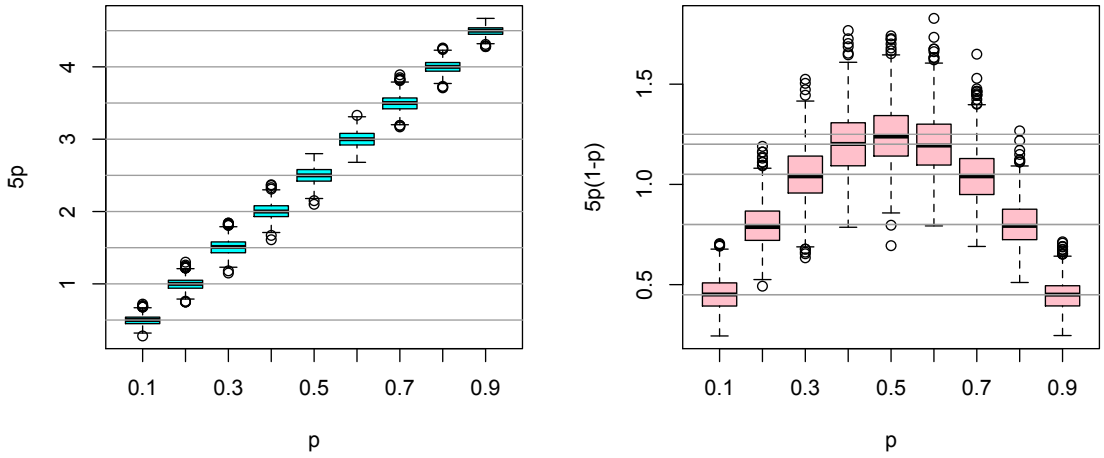
(a) Mean MC estimates of the mean should be something like the following:

p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$E(S)$	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5
$\hat{\mu}$	0.5010	0.9986	1.5047	2.0051	2.4989	3.0011	3.4952	4.0002	4.4966

(b) Mean MC estimates of the variance should be something like the following:

p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$Var(S)$	0.45	0.80	1.05	1.20	1.25	1.20	1.05	0.80	0.45
$\hat{\sigma}^2$	0.4509	0.7946	1.0500	1.2047	1.2485	1.2027	1.0480	0.8020	0.4507

(c) Boxplots of MC distributions:



(d) There is clear alignment between estimates $\{\hat{\mu}\}_{i=1}^{M}$ and $E(S)$ on one hand, and between estimates $\{\hat{\sigma}^2\}_{i=1}^{M}$ and $Var(S)$ on the other hand, in other words the analysis suggests that for M large,

$$\frac{1}{M}\sum_{i=1}^{M}\hat{\mu} \approx E(\mu) = E(S)$$

and

$$\frac{1}{M}\sum_{i=1}^{M}\hat{\sigma}^2 \approx E(\sigma^2) = Var(S)$$

Question 2 [30 marks]

Load the `commute.csv` dataset into your `R` session as follows:

```
dat = read.csv('commute.csv', stringsAsFactors=TRUE)
```

This dataset contains 500 observations of workers in a major US city on their age (in years), gender (M or F), and distance (in miles) and time (in minutes) of their commute to work each day. Each of the 500 respondents worked somewhere other than home.

Use $B = 1,000$ bootstrap resamples and set the random seed to 4060 (`set.seed(4060)`) before running each of your bootstrap analyses.

(a) Quote the bootstrap estimates of the mean commute time and of its associated standard error.

(b) Bootstrap the p-value of the correlation test (`cor.test()`) between the commuters' age and their commute time.

(c) What proportion of the bootstrap tests in (b) were significant at the 5% level?

(d) Compute and quote the bootstrap bias of the p-value from (b).

(e) Compute and quote a 95% confidence interval for the p-value estimation in (b), using the bootstrap mean and standard error and the normal approximation to formulate the confidence interval.

(f) Compute and quote an *ordinary* nonparametric bootstrap 95% confidence interval for the p-value estimation in (b).

(g) Briefly discuss whether the normal approximation is a reasonable approach for calculation of the confidence interval for the p-value of the correlation test.

Solution:
R code:

```
dat = read.csv('data/commute.csv',stringsAsFactors=TRUE)

set.seed(4060)
B = 1000
n = nrow(dat)
samples = matrix(sample(dat$Time, size=B*n, replace=TRUE), B, n)
bmeans = apply(samples, 1, mean)
time.se = sd(bmeans)
# BS expected value
mean(bmeans)
# BS std error
time.se

set.seed(4060)
pb = numeric(B)
for(b in 1:B)
```

```
ib = sample(1:n,n,replace=TRUE)
datb = dat[ib,]
pb[b] = cor.test(datb$Age,datb$Time)$p.value

mean(pb)
mean(pb<.05)
# statistic of original sample:
p0 = cor.test(dat$Age,dat$Time)$p.value
# BS bias:
mean(pb)-p0
# 95% Normal CI:
mean(pb)+c(-1,1)*1.96*sd(pb)
# BS-adjusted CI:
2*p0 - quantile(pb,c(.975,.025))
# Normal CI not appropriate given the distribution:
hist(pb)
```

(a) Bootstrap mean commute time: **29.11 minutes**.
Bootstrap standard error of the sample mean: **0.95 minutes**.

(b) Bootstrap p-value: **0.41**.

(c) Rate of bootstrap p-values belowe 0.05: **13%**.

(d) Bootstrap bias for the p-value: $\bar{p^*} - p^0 = \mathbf{-0.139}$

(e) Approximate 95% normal CI:

$$\bar{p^*} \pm 1.96 SE(p^*) = (\mathbf{-0.176}, \mathbf{0.999})$$

(alternative using actual bootstrap correction also accepted.)

(f) Nonparametric bootstrap 95% CI (q^*_α denoting quantile bootstrap p-value):

$$(2p^0 - q^*_{0.975}, 2p^0 - q^*_{0.025}) = (\mathbf{0.132}, \mathbf{1.098})$$

(g) Looking at the histogram of bootstrap p-values will show an almost uniform-looking distribution, in any case far from the normal pattern, which is sufficient to rule out the normal approximation baed on bootstrap statistics used in (e).

Question 3 [30 marks]

Load the `blood_pressure.csv` dataset into your `R` session as follows:

```
dat = read.csv(file='blood_pressure.csv')
x = dat$BMI
y = dat$Systolic
```

In this question we only use body mass index `x` (BMI, in kg/m^2) and systolic blood pressure `y` (in mmHg) from this dataset of 75 clinical observations of patients.

Set the random seed to 4060 (`set.seed(4060)`) before running your analysis.

(a) Fit a univariate GLM model to the whole dataset, to describe systolic blood pressure as a function of BMI. Compute and quote the root mean square error (RMSE) from this fit.

(b) Fit a kernel density estimator (KDE) to the *scaled* residuals from this fit. Quote the bandwidth that was automatically set for this KDE.

(c) Compute and quote the RMSE between the KDE obtained in (b) and a standard normal probability density function (pdf) evaluated at the same points.

(d) Compute and quote the RMSE between the KDE obtained in (b) and a Student pdf with two degrees of freedom evaluated at the same points.

(e) Based on (c) and (d), briefly discuss which model better describes the residuals, and what this tells you about your GLM fit obtained in (a).

(f) Using `library(splines)`, fit a B-spline to the data (`x`,`y`), using quantiles (0.15, 0.60, 0.70, 0.80) of `x` as knots. Compute and quote the RMSE from this fit.

(g) Briefly discuss whether you would use the GLM fit from (a) or the B-spline from (f) to analyse the data (`x`,`y`), using relevant output to support your opinion.

Solution:
R code:

```
library(glmnet)
dat = read.csv(file='data/blood_pressure.csv')
x = dat$BMI
y = dat$Systolic

glmo = glm(Systolic~BMI, data=dat)
sqrt(mean(residuals(glmo)^2))

kde = density(scale(glmo$residuals))
kde$bw

f.ref = dnorm(kde$x,0,1)
sqrt(mean((kde$y-f.ref)^2))

f.t = dt(kde$x,2)
```

```
sqrt(mean((kde$y-f.t)^2))

library(splines) # contains function bs()
KN = quantile(dat$BMI, c(0.15, 0.60, 0.70, 0.80))
BM = bs(dat$BMI, knots=KN)
B.spline = lm(y~BM)
sqrt(mean(residuals(B.spline)^2))

plot(dat$BMI,dat$Systolic,pch=20)
lines(dat$BMI,fitted(glmo))
os = order(dat$BMI)
lines(dat$BMI[os],fitted(B.spline)[os],col=3,lwd=2)
```

(a) GLM fit RMSE: **13.24**

(b) KDE bandwidth $h =$ **0.3280512**

(c) Distance from KDE to normal: **0.0180**

(d) Distance from KDE to Student: **0.0380**

(e) The KDE is clearly closer to the standard normal, indicating the model residuals are reasonably in line with the model assumption of normally distributed noise, which suggests a reasonable fit and model.

(f) B-spline fit RMSE: **12.62**

(g) The B-spline seems to yield a lower error (with a 4.7% decrease in RMSE over GLM), however plotting the fit will show an unreasonably complex structure; the B-spline overfits the data and the GLM is likely preferable here.

Question 4 [25 marks]

Load the `blood_pressure.csv` dataset into your `R` session as follows:

```
dat = read.csv(file='blood_pressure.csv')
x = model.matrix(Systolic~.+0, data=dat)
y = dat$Systolic
```

This dataset contains 75 observations of patients on five variables including their age (in years), waist circumference (in cm), systolic blood pressure (in mmHg), cholesterol level (in mg/dL) and body mass index (BMI, in kg/m^2).

Set the random seed to 4060 (`set.seed(4060)`) before running each of the below analyses.

(a) Implement 10-fold cross-validation of a lasso model of systolic blood pressure `y` as a function of the other four covariates `x`, using a fixed value $\lambda = 10$ for the regularisation parameter. Compute and store the training and test root means square errors (RMSEs) from the cross-validated fits. Quote the mean estimates of training and test prediction errors.

(b) Carry out the same analysis as in (a) but using a cross-validated value for the regularisation parameter. Compute and store the training and test RMSEs from the corresponding cross-validated fits. Quote the means of these training and test RMSEs.

(c) Carry out the same analysis as in (a) but using a multivariate GLM instead of the lasso. Compute and store the training and test RMSEs from the corresponding cross-validated fits. Quote the means of these training and test RMSEs.

(d) Plot the boxplots of all four sets of errors from (a) and (b) in a single plot frame. Briefly comment on these results, explaining any difference you may find.

(e) Based on the above results, briefly comment on each model performance and indicate whether one model is preferable over the other, and why.

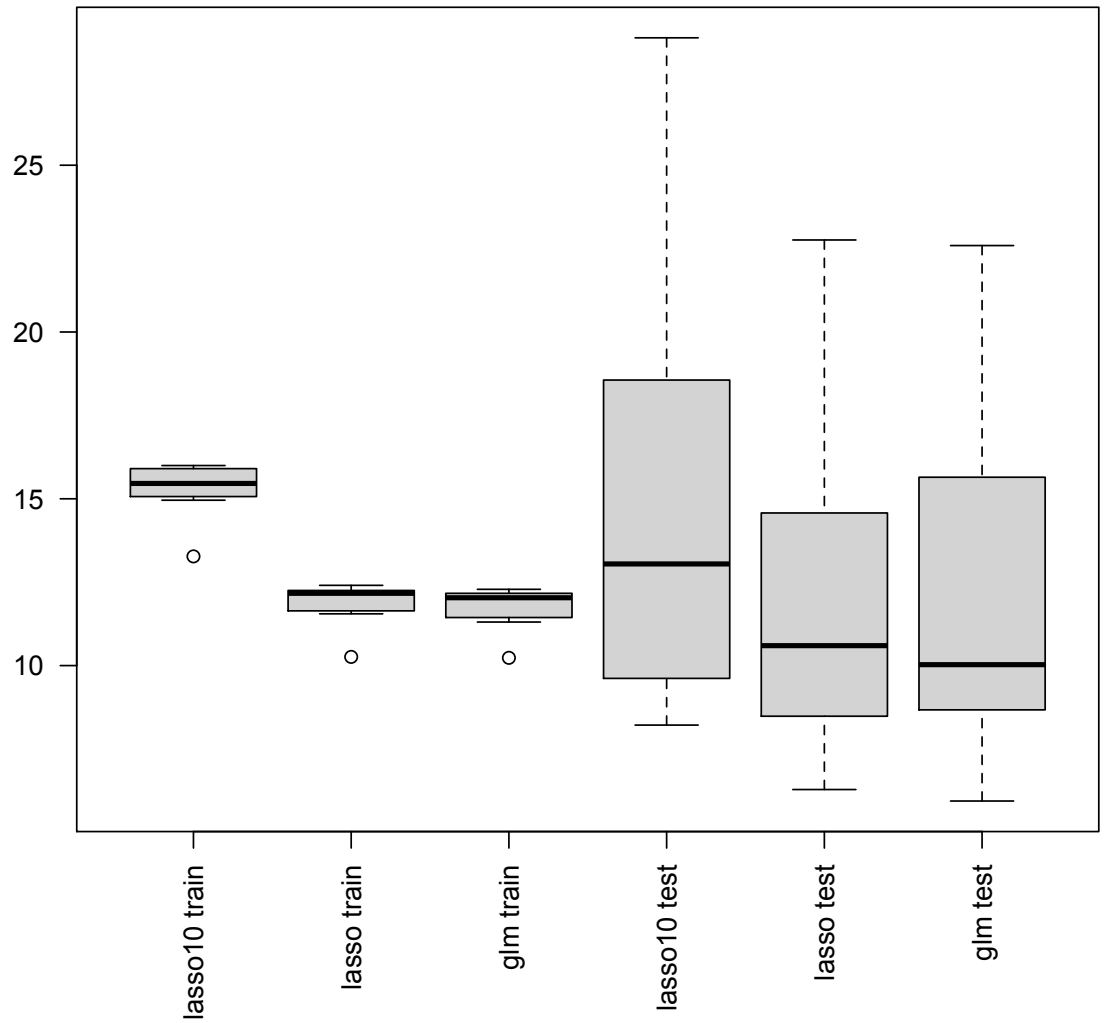
Solution:

(a) Naive lasso train RMSE: **15.3**, test RMSE: **14.6**.

(b) Tuned lasso train RMSE: **11.9**, test RMSE: **11.9**.

(c) GLM train RMSE: **11.8**, test RMSE: **11.9**.

(d) Boxplots:



(e) The naive lasso clearly underfits and is therefore not reliable for prediction. The tuned lasso and the traditional GLM perform comparably. There seems to be either little need for regularisation, or need to use an alternative strategy such as ridge regression to tackle colinearity in the data, but the GLM so far seems satisfactory. It would also make for a simpler model to use in a routine setting with non-experts.

R code:

```
library(glmnet)
dat = read.csv(file='data/blood_pressure.csv')
x = model.matrix(Systolic~.+0,data=dat)
y = dat$Systolic
n = nrow(x)
K = 10
folds = cut(1:n,K,labels=FALSE)
fit.rmse.lasso = p.rmse.lasso = numeric(K)
```

```
fit.rmse.lasso.cv = p.rmse.lasso.cv = numeric(K)
fit.rmse.glm = p.rmse.glm = numeric(K)
lam.cv = numeric(K)
set.seed(4060)
for(k in 1:K)
i.train = which(folds!=k)
i.test = which(folds==k)
x.train = x[i.train,]
y.train = y[i.train]
x.test = x[i.test,]
y.test = y[i.test]
#
lam = 10
lasso = glmnet(x.train,y.train,lambda=lam)
y.hat = predict(lasso,x.train)[,1]
fit.rmse.lasso[k] = sqrt(mean((y.hat-y.train)^2))
y.p = predict(lasso,x.test)[,1]
p.rmse.lasso[k] = sqrt(mean((y.p-y.test)^2))
#
lam.cv[k] = cv.glmnet(x.train,y.train)$lambda.min
lasso.cv = glmnet(x.train,y.train,lambda=lam.cv[k])
y.hat.cv = predict(lasso.cv,x.train)[,1]
fit.rmse.lasso.cv[k] = sqrt(mean((y.hat.cv-y.train)^2))
y.p.cv = predict(lasso.cv,x.test)[,1]
p.rmse.lasso.cv[k] = sqrt(mean((y.p.cv-y.test)^2))
#
glmk = glm(Systolic~.,dat[i.train,],family='gaussian')
glm.hat = predict(glmk,dat[i.train,])
fit.rmse.glm[k] = sqrt(mean((glm.hat-y.train)^2))
y.p.glm = predict(glmk,dat[i.test,])
p.rmse.glm[k] = sqrt(mean((y.p.glm-y.test)^2))

nms = c("lasso10","lasso","glm")
boxplot(fit.rmse.lasso,fit.rmse.lasso.cv,fit.rmse.glm,
p.rmse.lasso,p.rmse.lasso.cv,p.rmse.glm,
names=c(paste(nms,"train"),paste(nms,"test")),
las=2)
```

OLLSCOIL NA hÉIREANN, CORCAIGH
THE NATIONAL UNIVERSITY OF IRELAND, CORK

COLÁISTE NA hOLLSCOILE, CORCAIGH
UNIVERSITY COLLEGE, CORK

Examination Session and Year	Winter 2023
Module Code	ST4060 ST6040
Module Name	Statistical Methods for Machine Learning I Machine Learning and Statistical Analytics I
Paper Number	Paper Number: 1
External Examiner	Mr. Andrew Maclaren
Head of School	Dr. Kevin Hayes
Internal Examiner(s)	Dr. Eric Wolsztynski
Instructions to Candidates	• Please answer all questions. • Provide all your answers in the Word document provided. • Paste your R code into the Word document at the end of each question. • Submit a pdf version of your final Word document for Canvas submission. Note: if you do not manage to answer a question item, provide the R code you would have used, or a comment on the answer you would expect for that question, as relevant.
Duration of Paper	3 hours

List of required R libraries:

```
MASS
splines
```

List of (possibly) useful R functions:

```
abline()
apply()
approx()
as.numeric()
boxplot()
bs()
cbind()
coef()
colnames()
cut()
density()
fitted()
kmeans()
lines()
lm()
matrix()
mean()
median()
na.omit()
nrow()
numeric()
order()
par()
plot()
points()
prcomp()
predict()
quantile()
rnorm()
sample()
sd()
seq()
set.seed()
smooth.spline()
sqrt()
sum()
summary()
table()
which()
```

Question 1 [25 marks]

Consider the sample mean $\bar{X}$ of a sample of N independent and identically distributed realizations of a random variable $X \sim \mathcal{N}(\theta^\star, \sigma^2)$,defined by

$$\bar{X} = \frac{X_1 + \cdots + X_N}{N}$$

(a) Recall that the 95% confidence interval for $\bar{X}$ is obtained using the sample standard deviation s by

$$\mathcal{C} = \left[\bar{X} - 1.96\frac{s}{\sqrt{N}}, \bar{X} + 1.96\frac{s}{\sqrt{N}}\right]$$

Using $\theta^\star = 3$, $\sigma = 1.5$, and $M = 1,000$ Monte Carlo resamples, each of size $N = 30$, compute a Monte Carlo estimate of the proportion

$$p = \mathbf{I}(\theta^\star \in \mathcal{C})$$

i.e. the number of times that the confidence interval includes the true value $\theta^\star$. Set the random seed to 4060 (`set.seed(4060)`) before running your computation. Quote your Monte Carlo estimate of p. **[10]**

(b) Replicate the computation done in (a), but this time using

$$\mathcal{C} = \left[\bar{X} - 1.645\frac{s}{\sqrt{N}}, \bar{X} + 1.645\frac{s}{\sqrt{N}}\right]$$

Set the random seed to 4060 (`set.seed(4060)`) before running your computation. Quote your Monte Carlo estimate of p for this new confidence interval. **[5]**

(c) Comment on the values you obtained in (a) and (b), and indicate what these estimates correspond to. **[5]**

(d) Describe, in one or two sentences, what you would expect to happen if the sample size of each Monte Carlo sample was increased to $N = 100$ in the experiment described in (a), and why. **[5]**

Question 2 [25 marks]

Load the `Animals` dataset from library `MASS` into your `R` session as follows:

```
library(MASS)
x = Animals
```

This dataset contains average brain and body weights recorded for 28 species of land animals.

Implement a bootstrap analysis of the dataset using $B = 1,000$ bootstrap resamples, and setting the random seed to 4060 (`set.seed(4060)`) before running the analysis.

(a) Quote the bootstrap estimate of the mean brain weight of all land animals (your answer should be one value, calculated from the sample of all body weights for all species). Also quote the bootstrap estimate of the mean body weight of all land animals. **[5]**

(b) Quote the bootstrap estimate of the mean ratio of brain-to-body weight of all land animals. For any given animal species, this ratio is calculated as (brain weight)/(body weight). **[5]**

(c) Compute and quote the bootstrap estimate of the bias of the sample mean body weight estimate of these 28 species of land animals. **[5]**

(d) Compute and quote a bootstrap 95% confidence interval for the mean body weight of these 28 species of land animals. (You may use the *naive* (a.k.a. *quantile*) bootstrap confidence interval for this question.) **[5]**

(e) Except for the sample size, what explains the large width of the bootstrap confidence interval you obtained in (d) for mean body weight? Give your answer in two sentences maximum. Provide appropriate R output to support your answer. **[5]**

Question 3 [25 marks]

Load `R` libraries `splines` and `MASS` along with the `Boston` dataset as follows:

```
library(splines) # contains function bs()
library(MASS)
x = Boston$nox
y = Boston$medv
```

Set the random seed to 4060 (`set.seed(4060)`) before running your analysis.

(a) Fit a B-spline to the data, using knots placed at quantiles `c(0.15,0.40,0.60,0.70,0.85)` of `x`. Quote the B-spline coefficient estimates. **[5]**

(b) Generate predictions for new `x` values `newx = c(0.4,0.5,0.6)` from the B-spline obtained in (a). Quote the predicted values for `y`. **[5]**

(c) Fit a P-spline (i.e. smoothing spline) to the data, setting ordinary leave-one-out in the function arguments for computation of the smoothing parameter. Use `set.seed(4060)` before you run your code for this question.

(i) Quote the P-spline penalized criterion (RSS).

(ii) Provide a plot showing the fitted B-spline in (a) (in red) and the fitted P-spline (in blue) obtained in (c), over the (`x`,`y`) scatterplot (in black).

[5]

(d) Generate predictions for new `x` values `newx = c(0.4,0.5,0.6)` from the P-spline obtained in (c). Quote the predicted values for `y`. Compare those to the values obtained in (b) and explain any difference you may find between these two sets of predictions. **[5]**

(e) Implement 5-fold cross-validation of the P-spline. Quote the estimated prediction RMSE obtained from this analysis. Use `set.seed(4060)` before you run your code for this question.
Note: if you were not able to fit a P-spline, implement 5-fold cross-validation of a linear regression model (with intercept) instead, and quote the corresponding prediction error estimate. **[5]**

Question 4 [25 marks]

Load the `Pima.tr` dataset from the `MASS` package into your `R` session as follows:

```
library(MASS)
x = Pima.tr
x$type = NULL
y = Pima.tr$type
```

(a) Consider an analysis of this data that aims to predict `y` based on the measurements in `x`. Is this a regression or a classification problem? Justify your answer. **[5]**

(b) Perform k-means clustering on `x`, so as to cluster the data into k=2 clusters. Provide:

(i) The confusion matrix between the cluster labels and `y`.

(ii) A scatterplot of `x[,1:2]`, using `pch=20` to draw points as filled circles in the plot, and colour-coding (i.e. painting) the points with respect to their cluster (using black and red points).

[5]

(c) Briefly comment on the spatial distribution of the points, in terms of their cluster membership, in the scatterplot obtained in (b). In particular, explain any particular pattern you may see in this spatial distribution. **[5]**

(d) Perform *scaled* PCA on the feature matrix `x`. Indicate the number of principal components that together capture 90% of the information in `x`. Justify your answer. **[5]**

(e) Perform *unscaled* PCA on the feature matrix `x`. Indicate which features mainly influence the first 2 principal components. Justify your answer. **[5]**

OLLSCOIL NA hÉIREANN, CORCAIGH
THE NATIONAL UNIVERSITY OF IRELAND, CORK

COLÁISTE NA hOLLSCOILE, CORCAIGH
UNIVERSITY COLLEGE, CORK

Examination Session and Year	Winter 2024
Module Code	ST4060 ST6040
Module Name	Statistical Methods for Machine Learning I Machine Learning and Statistical Analytics I
Paper Number	Paper Number: 1
External Examiner	Mr. Andrew Maclaren
Head of School	Dr. Kevin Hayes
Internal Examiner(s)	Dr. Eric Wolsztynski
Instructions to Candidates	• There are 4 equally weighted questions, worth a total of 100 marks (25 marks each). • Please answer all questions. • Provide all your answers in the Word document provided. • Paste your R code into the Word document at the end of each question. • Submit a pdf version of your final Word document for Canvas submission. Note: if you do not manage to answer a question item, provide the R code you would have used, or a comment on the answer you would expect for that question, as relevant.
Duration of Paper	3 hours

List of required R libraries:

```
glmnet
splines
```

List of (possibly) useful R functions:

```
apply()
approx()
as.numeric()
boxplot()
bs()
cbind()
coef()
colnames()
cor.test()
density()
fitted()
lines()
lm()
matrix()
mean()
median()
na.omit()
nls()
nrow()
numeric()
order()
par()
plot()
points()
predict()
quantile()
sample()
sd()
seq()
set.seed()
smooth.spline()
sqrt()
sum()
summary()
which()
```

Question 1 [25 marks]

Your answers to any of the following question items should not exceed three sentences.

The scatterplot of Figure 1 (top) shows three models fit to a dataset $\{(X, Y)\}_{i=1}^{n}$ of n i.i.d. observations. You may refer to this plot to answer question items (a), (b), (c) and (d), which do not involve any resampling.

(a) Indicate whether each of sensitivity, RMSE, and AUROC may be used to assess model fit for this data, and briefly explain why. **[5]**

(b) Based on the graph, indicate which of the three models yields the best fit in your opinion, and briefly explain why. **[5]**

(c) The sums of squared residuals from the three models are as follows:

- Model 1: 26.0
- Model 2: 19.4
- Model 3: 23.6

Based on these values, and on your findings in (b), indicate which of the three models yields the best fit in your opinion, and briefly explain why. **[5]**

(d) Briefly discuss whether model 3 may have been obtained from a model whose cost function is defined for $\beta_0 \in \mathbb{R}$, $\beta_1 \in \mathbb{R}$ and $\lambda \in \mathbb{R}^+$ by

$$\sum_{i=1}^{n}(Y_i - \beta_0 - \beta_1 X_i)^2 + \lambda(\beta_0^2 + \beta_1^2)$$

[5]

(e) The boxplots labelled B1, B2, B3 and B4 in Figure 1 (bottom) correspond to four distributions of RMSEs, all obtained by bootstrapping fits and predictions from model 2 and model 3. For each of these four boxplots, indicate whether it corresponds to training or test set RMSEs, and from which model. Briefly justify your answer. **[5]**

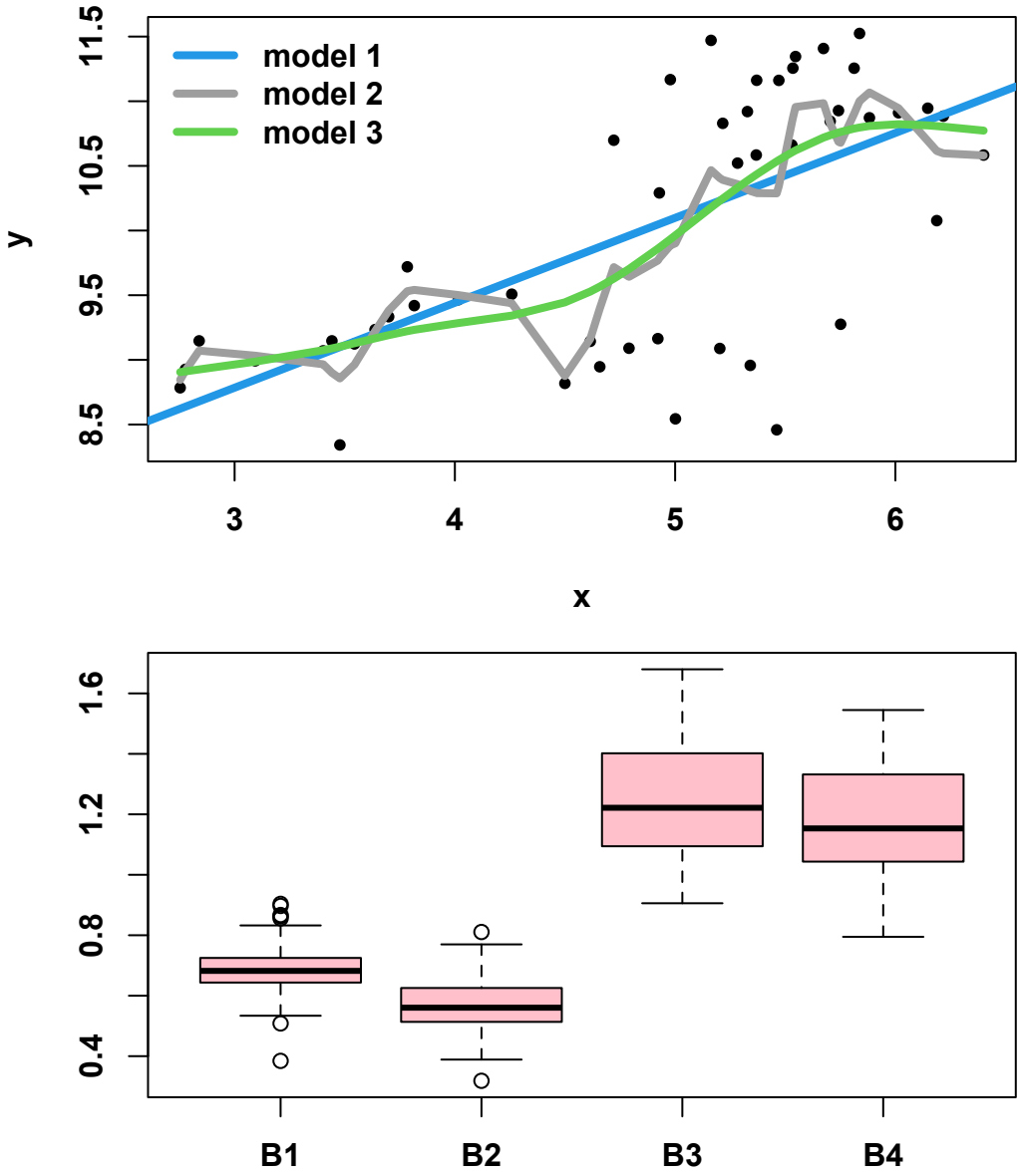


Figure 1: Top: scatterplot of Question 1. Bottom: boxplots of Question 1.

Question 2 [25 marks]

Your answers to any of the following question items should not exceed three sentences.

Load the `rock` dataset into your `R` session as follows:

```
x = rock$peri
y = rock$perm
```

This dataset contains measurements on 48 rock samples from a petroleum reservoir. In this question, we focus on the univariate relationship between variables `peri` (perimeter, in pixels) and `perm` (permeability, in milli-Darcies).

(a) Fit two smoothing splines (i.e., P-splines) to the entire dataset (X, Y). Set their smoothing parameter `spar` to .8 and 0.95 respectively. Compute and quote the RMSE for both fits. **[5]**

(b) Implement simple (i.e., not repeated, and not stratified) 10-fold cross-validation (CV) of the second smoothing spline (i.e. using `spar`=0.95). Set the random seed to 4060 (`set.seed(4060)`) before running the analysis. Compute and quote the estimated training and test RMSEs for this model. **[5]**

(c) Implement simple (i.e., not repeated, and not stratified) 10-fold cross-validation (CV) of a linear regression model with intercept. Set the random seed to 4060 (`set.seed(4060)`) before running the analysis. Compute and quote the estimated training and test RMSEs for this model. **[5]**

(d) Quote the standard deviation of training and test RMSEs obtained from CV of the linear regression model in (c). Briefly explains what causes the difference between these two values. **[5]**

(e) Briefly describe what would be achieved by changing from simple 10-fold to repeated 10-fold CV. (No code is required for this question item.) **[5]**

Question 3 [25 marks]

Your answers to any of the following question items should not exceed three sentences.

This question involves multivariate analyses of the `rock` dataset, with `perm` as the dependent variable Y. See `?rock` for more information on the data. As a pre-requisite for this question, fit a simple linear regression model to the data as follows:

```
lm.fit = lm(perm~., data=rock)
summary(lm.fit)
```

(a) Implement a bootstrap analysis of the dataset, fitting a linear regression similar to the one above, using all variables in the data and 100 bootstrap resamples. Do not scale any of the variables. Do not use function `boot` for this question. Set the random seed to 4060 (`set.seed(4060)`) before running the analysis. Quote the bootstrap estimates of the effect of `area` and of `peri` on `perm`. **[5]**

(b) Compute and quote the bootstrap estimate of the bias of the estimator of the regression coefficient associated with variable `shape` in this model fit. Briefly indicate how you calculated this value (you can paste R code to do this). **[5]**

(c) Compute and quote the naive 95% bootstrap confidence interval around the standard error of the estimator of the regression coefficient associated with variable `shape` in this model fit. Briefly indicate how you calculated this value (you can paste R code to do this). **[5]**

(d) Modify your bootstrap implementation to assess whether this model is prone to overfitting. Is there evidence of overfitting? Briefly comment on your findings, using numerical output of your choice. **[5]**

(e) Briefly explain what difference you would expect to see between the variances of the distributions of training errors obtained from this bootstrap analysis and from simple 5-fold cross-validation, and why. **[5]**

Question 4 [25 marks]

Your answers to any of the following question items should not exceed three sentences.

This question involves multivariate analyses of the `mtcars` dataset, with `mpg` as the dependent variable Y. See `?mtcars` for more information on the data. Load the dataset into your R session as follows:

```
x = mtcars
x$mpg = NULL
y = mtcars$mpg
```

(a) Apply scaled Principal Component Analysis (PCA) to the feature set `x` (make sure that `z` is of class `matrix`). How many principal components (PC) contain more than 5% of the total variance in the data? Quote the proportion of variance explained by each of these PCs. **[5]**

(b) Create `z` as the rotated (i.e. image) data of `x` obtained after PCA in (a). Quote the values obtained by evaluating `apply(z,2,sd)[1:3]`. **[5]**

(c) Perform two linear regression analyses as follows:

```
fit1 = lm(y~., data=x)
fit2 = lm(y~., data=z)
```

Quote and compare the sums of squared residuals and the adjusted R^2 for `fit1` and `fit2`, and briefly explain what you observe. (If you did not successfully answer question 4(b), indicate what you would expect to see.) **[5]**

(d) In this question, you considered using PCA to analyse the `mtcars` dataset. Considering the type of data this dataset contains, could this approach be criticised? Briefly justify your answer. **[5]**

(e) Perform k-means clustering of the unscaled data `x` using three clusters. Inspect the distribution of cluster labels from this output with respect to the variables in `x`. What drove the clustering algorithm? **[5]**

OLLSCOIL NA hEIREANN, CORCAIGH
THE NATIONAL UNIVERSITY OF IRELAND, CORK

COLAISTE NA hOLLSCOILE, CORCAIGH
UNIVERSITY COLLEGE, CORK

ST4060 - ST6015 - ST6040
Statistical Learning and Machine Learning I

Tutorials and exercises - 2021-22
Focused set version 1.1

Eric Wolsztynski
eric.w@ucc.ie

Contents

Note: some of the questions in this document are treated in *Statistical Computing with R* by Maria L. Rizzo (2008), Chapman & Hall.

0 Practicing with R

Question 0.1

Our objective here is to manipulate a dataset and run summary statistics on the data.

(a) Set the R working directory as a desired location folder (named for instance `ST4060_practicals`). Then open an R script in your editor and save it in the working directory (named e.g. `ST4060_practical1-1.R`).

(b) Start by looking into the built-in dataset `airquality`.

- Load the dataset `airquality` in the R environment
- Have a quick peak at it using function `head(airquality)`
- Study the dataset further with `?airquality`, `dim()`, `names()` and `summary()`

(c) A detailed summary of the `Month` and `Day` variables is not necessary. Run a more detailed analysis of the dataset as follows:

- Apply `summary()` to the first column of data
- Display contents of the first 4 columns at once
- Display contents of columns 1, 2 and 4 only, at once
- Attach the dataset to the environment (for ease of use) using `attach()`
- View summary statistics for variable `Ozone`
- Compute its median, and identify a potential problem in the data
- Compare its mean and median (use argument `na.rm=TRUE`)
- Count the number of `NA`'s contained in vector `Ozone`
- Create a new vector called `Oz` that does not contain these `NA`'s
- Compute the following summary statistics for this new variable: `mean()`, `median()`, `sd()`, `var()`, 25th quantile

(d) Run analyses on a specific subset...

- Display a subset of the whole dataset that includes entries with `Wind`>10 and `Temp`<70 only
- Store this subset in a new variable (named e.g. `cool_n_windy`)
- Compute the correlation coefficient between `Wind` and `Temperature` information for this subset
- Now remove `NA`'s from the two vectors and recomputed the correlation

(e) It's time to visualize the data.

- Simply instruct to plot the whole dataset with `plot(airquality)`
- Compare this plot with `pairs(airquality[c(1:4)])`
- Plot a histogram of the `Ozone` variable using `hist()`

- Plot a scatterplot of `Ozone` vs `Temperature` using formula `Ozone~Temp`
- Plot a boxplot of `Ozone` as a function of `Month` (`Ozone~Month`)

(f) Now save a plot you'd like to present in a talk/report as a `pdf`. We plan on having two panels side-by-side containing the scatterplot and the boxplot obtained earlier. You may first like to create a directory called e.g. `output` in your ST4060 practicals directory, for tidiness.

- Open a pdf file connection using `pdf(...)` – this will create the file on disk and make it available for R to write into
- Prepare your plotting window using
  `par(mfrow=c(1,2), font=2, font.lab=2, font.axis=2, pty='s')`
- Plot the scatterplot of `Ozone` against `Temperature`, with points marked by the symbol `'*'` and of `size 2`, add a main title `"Ozone vs Temperature"`, an x-label `"Temperature"`, and a y-label `"Ozone"`
- Plot the boxplot, with line width 2, filled with gray, and titled `"Nice boxplot"`
- Close the pdf file connection using `dev.off()`. Without this step the file will not be readable, i.e. it could not be used. Leaving the R session with an open connection will likely result in an I/O problem or error of some sort.

(g) Develop your own R function that computes the Inter-Quartile Range (IQR) for a given vector of values `x`, by filling up the blanks within the following structure:

```
iqr <- function(x){
      ...
}
```

Then test it! (Hint: `iqr(Oz)`)

(h) Clean up! This means detaching the dataset from the R environment using `detach()`; and/or using `rm()`.

Question 0.2

Our objective here is to create a short "video clip" from a few image frames, by implementing dedicated functions and using an array data object. Use the following six images: `cinq.pgm`, `quatre.pgm`, `trois.pgm`, `deux.pgm`, `un.pgm`, `zero.pgm`. Note that a `.pgm` file is editable; in these, the third row indicates image dimensions (e.g. in the form 662 310 for a 662 x 310 picture).

(a) Set the R working directory as a desired location folder. Ideally this folder will include another folder called `data`, which itself contains the image files. Otherwise adapt the file paths in your R instructions accordingly. Then open an R script in your editor and save it in the working directory (named e.g. `ST4060_practical1-2.R`).

(b) Start by looking into the data a bit:

- Scan the dimensions of `cinq.pgm` using `scan()` and save values in variable `dim5`
- Load image `cinq.pgm` into a variable named `mat5`, using functions `scan()` and `matrix()`

- View the matrix as an image using `image()`
- Analyse what's wrong with this plot (at least 3 things are!)

(c) A quick fix using `t()` inside the call to `image()` is not sufficient to fix the orientation. We need to define a process that we will apply to all such images in order to view them the way we want. Since we will repeat the same process, we implement it as an R function.

- Write a function called `flipv()` that flips a matrix (or image) vertically
- Write another function, called `mimage()`, that runs a "personalised" call to `image()`, using grayscale and removing axes
- Try these out on mat5 using instruction `mimage(flipv(mat5))`

(d) Repeat the process successively on `quatre.pgm`, `trois.pgm`, `deux.pgm`, `un.pgm` and `zero.pgm`. We may as well create a wrapping function wrap() that runs all required instructions in a nice short call, i.e. for example `wrap("cinq.pgm")`. Then run all calls at once:

```
wrap("cinq.pgm")
wrap("quatre.pgm")
wrap("trois.pgm")
wrap("deux.pgm")
wrap("un.pgm")
wrap("zero.pgm")
```

(e) Store all images into an array, using `flipv()`. This will allow to carry all frames in the one object. Then display each frame of the array successively using a `for()` loop. (E.g. from 5 down to 0.)

(f)
- Install package animation
- Load the package using `library(animation)`
- Create a function, say `gif.gen()`, that will be used in the final call below
- Generate a gif file using `saveGIF(gif.gen(),"mygif.gif",outdir="...")`
- Try out `saveHTML()` also!

(g) Want more? Look up package `rimage`...

1 Stochastic modelling and KDE's

1.1 Distribution models and simulation

Question 1.1 (Simulation)

Using a sample size of $N = 1000$, generate a sample of realizations of an $Exp(1/2)$ random variable, using an appropriate R command. Plot its histogram and overlay the curve of the theoretic probability distribution function, as well as the curve of a nonparametric density estimate for the sampling distribution.

Question 1.2 (Simulation)

(a) Implement pseudo-random generation of Huber's contamination model

$$f_\varepsilon(u) = \varepsilon\phi(u) + (1-\varepsilon)h(u)$$

where $\varepsilon \in (0,1)$, $\phi(u)$ denotes the Standard Normal distribution, and using the t-distribution with 3 degrees of freedom for $h()$. Generate 3 different samples of size $N = 100$ from Huber's contamination model, setting ε to be successively 0.95, 0.40 and 0.20. Provide the sample means and standard deviations, rounding off to 3 decimal places, for each of the three samples.

Note: you may write a function, e.g. `rhuber <- function(N,epsilon=0,dof=3){...}` out of convenience, but this is not a requirement.

(b) Create a 3-frame plot showing the histograms of each generated sample; the frames should be organised in 3 rows and 1 column and the ranges of the x-axes should be set equally for all 3 plots to allow for direct comparison.

(c) Create a dataframe that contains the 3 samples organised in columns, and specify names for each column so as to keep track of the value of epsilon used to generate same (e.g. `e095`, `e040` and `e020` could be used as names). Write this dataframe to a `.csv` file so that the file, once open (e.g. in Microsoft Excel), only shows 3 columns (i.e. it should not contain a first column with row numbers).

1.2 Nonparametric density estimation

Question 1.3 (kernel density estimation)

Generate samples of $n = 1000$ variates from the following distributions, and for each sample,

- compute its kernel density estimator (KDE),
- plot the KDE over the rug of sample points,
- overlay a line indicating the true underlying pdf used to generate the sample:

(a) a Normal $\mathcal{N}(2,1)$;

(b) a Student t-distribution with $t = 3$;

(c) an $Exp(2)$.

Any comments?

Question 1.4 (Multivariate density estimation)

(a) Compute and plot a 2D KDE for the Old Faithful dataset in R, using the appropriate function(s) from R's `KernSmooth` package.

(b) Compute and plot a nonparametric regression function estimate for the Old Faithful data using local polynomials from the same package.

Question 1.5 (Multivariate density estimation)

(a) Generate a random sample of size $n = 1000$ from the $\mathcal{N}_2(\mu, \Sigma)$ distribution.

(b) Fit a 2D (bivariate Normal) kernel density estimator to the sample (use R's `MASS::kde2d` function).

(c) Fit a 2D kernel density estimator to the sample using a product of univariate kernels, i.e. implement the KDE defined by

$$\hat{f}(u) = \frac{1}{n} \sum_{i=1}^{n} \prod_{j=1}^{d} \frac{1}{h_j} K\left(\frac{u^{(j)} - X_i^{(j)}}{h_j} \right)$$

with $d = 2$, $h_1 = h_2 = 0.5$, and u ranging over the (x, y) grid computed by `kde2d` in the previous question.

(d) For each KDE,

- plot the cloud of points and add a contour of the associated KDE;
- generate a perspective plot with angle $\theta = 30^o$;
- generate a perspective plot with angle $\theta = -60^o$.

2 Resampling methods

2.1 Monte Carlo

Question 2.1 (Monte-Carlo integration)

Compute a Monte Carlo estimate of

$$\theta = \int_2^4 e^{-x} dx$$

and compare your result with the exact value of the integral.

Question 2.2 (Monte-Carlo integration)

Use the Monte Carlo approach to evaluate the standard Normal cdf (assume $x \geq 0$ for simplicity):

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt$$

Evaluate also the variance and 95% confidence interval associated with your estimate.

Question 2.3 (Monte-Carlo estimation)

(a) Implement a function that given an integer $k < n$ and a sample X, computes the trimmed mean of X, which is defined for an ordered sample $X_{(1)}, \ldots, X_{(n)}$ by

$$\bar{X}_{[-k]} = \frac{1}{n-2k} \sum_{i=k+1}^{n-k} X_{(i)}$$

(b) Implement a Monte Carlo simulation using the standard Normal as sampling distribution, with $n = 20$ and $M = 1000$, to evaluate the distribution and MSE of this estimator under this model.

(c) Run a similar simulation using a Student t-distribution with 1 d.o.f. as samplig distribution, and compare output estimator performances.

Question 2.4 (Step-by-step question - do it yourself :)) (Monte-Carlo estimation)

With Monte Carlo repetitions, the objective is to set up M repetitions of a statistical experiment, where for each experiment:

- we generate a new sample of observations randomly,
- we perform a statistical analysis (including model fitting) on this sample,
- we store the results.

Once the M experiments are finished, we analyse the sample distribution of the parameter estimates, generate some plots and store some information. This approach is used in particular to approximate the asymptotic characteristics of some statistical procedure and benchmark several techniques in terms of their distribution. One example could be comparing two robust estimators for linear regression with heavy-tailed noise (e.g. log-Normal, Laplace)... For example we could use least squares (LS) and robust M-estimation (RM).

(a) Initialize the simulation parameters and storing variables.

- Let $N = 50$ (sample size) and $M = 100$ (number of Monte Carlo repetitions)
- Set $a = 7$ and $b = 3$ (resp. intercept and slope parameters in a linear model)
- Create the vector of regressors (design) `x = rep(c(1:5), N/5)`
- Set noise parameters to be $m = 0.5$ and $s = 1.2$ (mean and standard deviation)
- Set a random seed (e.g. `rseed=0`) for pseudo-random generation
- Allocate storage vectors `LSvec = RMvec = matrix(0,2,M)`
- Finally, run `set.seed(rseed)` and import libraries `MASS` and `VGAM`

(b) Implement the Monte Carlo repetitions. Create a for loop from 1 to M, within which you will:

- Generate a new sample of realizations of noise e from a $log\mathcal{N}(m, s)$
- Generate a new sample of observations `y = a + b*x + e`
- Estimate (a, b) via Least Squares for this new sample (use `lm` or `mylm`)
- Estimate (a, b) via robust M-estimation using `rlm`
- Store these estimates in the adequate vectors (use e.g. `rbind`)
- Note: you can also decide to store the samples of noise and observations for each loop – in case this may be useful later on, for instance

(c) Analyse the two sets of estimates.

- Create a plotting window with 2×3 panels with `par(mfrow=c(2,3))`
- Plot histograms for each set of estimates
- Plot nonparametric density estimates for each vector with `plot(density())`
- Compare the biases, variances and MSE's for all estimators
- Check that the tradeoff between bias and variance is found in the MSE
- Which approach seems more appropriate?
- Note: A similar analysis should be carried out with Normal noise to assess the potential loss incurred by the use of an M-estimator in place of the optimal Least Squares.

(d) Write outputs to file.

- Create a dataframe containing all outputs of interest, adding names for each column
- Write this dataframe to disk as an output `.csv` file, using `write.csv()`
- Test this file: view it in Excel

- Test again: load it up using `read.csv()` and recompute the biases as a check

Question 2.5 (Monte-Carlo estimation)

We aim to demonstrate the statistical properties of two estimators of the standard deviation via Monte Carlo simulations. Given a sample of observations $\{X_1, \ldots, X_N\}$, and using the sample mean

$$\bar{X} = \frac{1}{N} \sum_{i=1}^{N} X_i$$

the two estimators considered are denoted s and $\hat{\sigma}$ and are defined by

$$s^2 = \frac{1}{N-1} \sum_{i=1}^{N} (X_i - \bar{X})^2$$

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{i=1}^{N} (X_i - \bar{X})^2$$

(a) Implement a Monte Carlo experiment to generate M samples of size N from a Normal $\mathcal{N}(0, 2^2)$ distribution, and compute sample standard deviations estimates for s and $\hat{\sigma}$, for each of these samples using the formulas above. Include also a computation of the sample standard deviation using R's function `sd` for each Monte Carlo sample generated. Generate $M = 1000$ Monte Carlo estimates for four different sample sizes: $N \in (10, 20, 50, 100)$. Finally, prepare also a 4-panel plot window (using `par(mfrow=c(2,2))`) for the plots requested below.

(b) Check whether R's function `sd` corresponds to one of the estimators s or $\hat{\sigma}$.

(c) Plot the histograms for both Monte Carlo samples for s and $\hat{\sigma}$ for the case $N = 10$, in separate plot panels. Also provide values, rounded to two decimal places, for the Monte Carlo biases and variances for both estimators and for $N = 10$.

(d) Based on your analysis in this question, give your conclusions on the comparison of bias and variance of these two estimators, s and $\hat{\sigma}$.

(e) Plot boxplots corresponding to all four sample sizes, in order to show the progression of the distribution of each estimator with respect to sample size N. Display the plot corresponding to s in one plot panel, and that corresponding to $\hat{\sigma}$ in the other of the last two remaining plot panel.

Question 2.6 (Monte-Carlo estimation)

In this question, we aim to assess whether the rate of convergence of the sample mean of a χ^2-distributed sample may depend upon the number of degrees of freedom associated with the distribution.

(a) Implement `M=100` Monte Carlo repetitions of an experiment such that:

- All values in $\{2, 4, 10\}$ are successively used as number of degrees of freedom `ndf`;

- A sample size of `n=100` is used;
- For each value of `ndf`, `M=100` samples `x` of a χ^2-distribution with `ndf` degrees of freedom are generated and their mean stored in an array `ms` of dimensions `M`×`3`.

(b) After running the implementation in part (a), generate a figure showing the boxplots for the distributions of means corresponding to each number of degrees of freedom. Comment (briefly) on this figure. Can you observe a particular feature of the χ^2-distribution?

Question 2.7

Note: parts of this quesiton are also covered in Section 2.

(a) Implement pseudo-random generation of Huber's contamination model

$$f_\varepsilon(u) = \varepsilon\phi(u) + (1-\varepsilon)h(u)$$

where $\varepsilon \in (0,1)$, $\phi(u)$ denotes the Standard Normal distribution, and using the t-distribution with 3 degrees of freedom for $h()$. Generate 3 different samples of size $N = 100$ from Huber's contamination model, setting ε to be successively 0.95, 0.40 and 0.20. Provide the sample means and standard deviations, rounding off to 3 decimal places, for each of the three samples.

Note: you may write a function, e.g. `rhuber <- function(N,epsilon=0,dof=3){...}` out of convenience, but this is not a requirement.

(b) Create a 3-frame plot showing the histograms of each generated sample; the frames should be organised in 3 rows and 1 column and the ranges of the x-axes should be set equally for all 3 plots to allow for direct comparison.

(c) Create a dataframe that contains the 3 samples organised in columns, and specify names for each column so as to keep track of the value of epsilon used to generate same (e.g. `e095`, `e040` and `e020` could be used as names). Write this dataframe to a `.csv` file so that the file, once open (e.g. in Microsoft Excel), only shows 3 columns (i.e. it should not contain a first column with row numbers).

(d) Implement a Monte Carlo (MC) simulation in which you generate $M = 500$ samples of size $N = 100$ from the Huber $f_{0.40}(u)$ distribution (i.e. using $\varepsilon = 0.40$), and another M samples from the Normal $N(0,1)$ distribution. For each Monte Carlo repetition, compute and store the sample means and standard deviations of both the Normal and the Huber samples. Provide the averages of the MC samples of means and standard deviations for both distributions, rounding off all averaged values to 3 decimal places.

Note: If your implementation of Huber's model $f_\varepsilon(u)$ did not work out in (a), you may generate samples from Student's t-distribution with 3 degrees of freedom instead.

2.2 Bootstrapping

Question 2.8 (Bootstrap estimation of standard error)

Load the law school dataset in the `bootstrap` package, estimate the correlation between the two variables in this dataset, and evaluate the bootstrap estimate of the standard error associated with this estimation.

Question 2.9 (Bootstrap linear regression estimates)

Consider R's `cars` dataset.

(a) Obtain relevant regression parameter estimates for this dataset.

(b) Generate $M = 10000$ bootstrap estimates for these coefficients.

(c) Inspect the one-dimensional (i.e. marginal) distributions for all relevant bootstrap parameter estimates, and state your conclusions.

(d) Inspect the joint distribution for these sample parameter estimates, and state your conclusions.

Question 2.10 (Nonlinear estimation and bootstrapping)

Load the following data from R's `mtcars` dataset:

```
x = mtcars$disp
y = mtcars$mpg
```

(a) Task: fit an exponential model

$$Y_i = exp(\theta_1 + \theta_2 X_i) + \varepsilon_i, \qquad i = 1, \ldots, n, \qquad \varepsilon_i \text{ i.i.d.}$$

to the sample `y` using R's function `nls()`. Use initial values $\theta^{(0)} = (3, -0.01)$.
Required:

(i) quote the coefficient estimates for this model fit;

(ii) explain, based on numerical assessment but without performing any further computations, whether this model appropriately describes the relationship between `x` and `y`;

(iii) provide a plot showing the model fit *as a line* going through the data points. Plot the data points in black and the model fit in red.

(b) Name an alternative regression technique that would provide a nonlinear representation of the relationship between `x` and `y`, without assuming a particular model.

(c) Task: set the pseudo-random seed to 1 (R instruction `set.seed(1)`) and compute $B = 100$ bootstrap estimates for the model fit of (a).
Required:

(i) quote the bootstrap means and standard deviations for estimators $\hat{\theta}_1$ and $\hat{\theta}_2$;

(ii) quote the bootstrap estimate of the standard error for estimator $\hat{\theta}_1$;

(iii) quote a nonparametric 95% confidence intervals for model parameter θ_1;

(iv) comment on the confidence interval found in (c.iii).

Round off your numerical answers to 4 decimal places where applicable.

2.3 Cross-validation

Question 2.11 (Cross-validation frameworks)

Load the following data from dataset `Boston` in library `MASS`:

```
x = Boston[, c("crim", "indus", "rm", "tax")]
y = Boston$medv
```

You will probably need to coerce `x` into a matrix before passing into `lm`:

```
x = as.matrix(x)
lmo = lm(y~x)
summary(lmo)
```

This linear model seems alright, but what about its predictive performance?

(a) Perform a 50%-50% train-test split of the dataset. Fit the linear model on the training data, generate predictions from this model for the test data, and calculate the corresponding prediction Root Mean Square Error (RMSE).

(b) Implement Leave-One-Out CV on the data (`x`, `y`) and calculate the LOO-CV test set prediction RMSE estimate.

(c) Implement K-fold CV on the data (`x`, `y`) and calculate the K-fold CV test set prediction RMSE estimate, using K=5.

(d) Implement K-fold CV on the data (`x`, `y`) and calculate the K-fold CV test set prediction RMSE estimate, using K=10.

(e) Compare the prediction error estimates obtained from (a), (b), (c), and (d).

3 Estimation theory

3.1 Linear regression

Question 3.1 (Step-by-step question - do it yourself :))

The objective here is to implement linear regression analyses and explore basic aspects of stochastic modelling.

(a) Set the R working directory as a desired location folder. Then open an R script in your editor, and save it in the working directory, using a sensible name.

(b) We consider the in-built dataset faithful, which contains... well, look at `?faithful`. We want to fit the data as follows:

$$\text{eruptions} = \alpha + \beta \times \text{waiting} + \text{noise}$$

Let's look at the data first. Note: when plotting, think about presentation too.

- Plot eruptions against waiting times
- Fit a linear model to `eruptions~waiting` using `lm()`, and store its output in a variable called e.g. `lm.out`
- Display the coefficients α and β obtained from the fit
- Add a line $\alpha + \beta x$ to the plot with `abline()`
- Display a summary of the output of `lm()`
- Compare the components of `lm.out` and of `summary(lm.out)` with `names()`
- Store the value of the adjusted coefficient of determination (i.e. the adjusted R^2) associated with the linear fit

(c) Now we look at forecasting based on this linear fit.

- Develop a 95% confidence interval of the mean eruption duration for the waiting time of 80 minutes using

  ```
  predict( lm(...), newdata=data.frame(...), interval="confidence" )
  ```

- Generate a 5-step-ahead prediction, including corresponding confidence interval, for waiting times `max(waiting)+c(1:5)`
- In a plot window with extended x- and y-axes so as to include predicted values (use arguments `xlim=c(...,...)` and `ylim=c(...,...)` inside the call to `plot()`):
  - plot the actual data (eruptions against times)
  - add a line that indicates the linear fit
  - finally, add the 5 forecast points in red using `points()` and note whether they fit on the line or not

(d) We now carry out some significance tests, to assess whether there is a significant relationship between waiting times and eruptions. This significance is tested under the null hypothesis $H_0 : \beta = 0$.

- Display the coefficients of the output `lm.out`
- Identify what `summary(lm.out)$coefficients[,4]` corresponds to
- Learn more about it from the help page `?summary.lm`

We further run some diagnostic checks on the residuals:

- Prepare a 1×2 plotting window with `par(mfrow=c(1,2), ...)`
- Plot the residuals obtained from the linear fit `lm.out`
- Plot a QQ-plot of these residuals with `qqnorm()`, storing its output in `qqlm`
- Add the line corresponding to the linear fit `qqlm$y~qqlm$x` (e.g. in red)

(e) Write your own `lm()` function! This is good practice to learn to manipulate vector products. Create the structure for a function called `mylm()`, which takes two arguments (`x` and `y`), as follows:

```
mylm <- function(x, y){
# Returns my own linear fit ?
     ...
}
```

Inside the body of this function:

- compute `xm` and `ym`, centered versions of `x` and `y`
- Compute `b`, the linear estimate of β, defined by $(x^T x)^{-1}(x^T y)$, using `%*%`
- Compute the linear estimate of α, defined as the sample mean of `y-bx`
- Compute the residuals `y - a - bx`
- Create a list that contains 3 components, namely estimates for the intercept and slope, and the residuals
- Return this list as the last instruction in the body of your function `mylm()`

Now try it! Compare

```
lm(eruptions~waiting, data=faithful)
```

with

```
mylm(faithful$waiting , faithful$eruptions)
```

(f) Go further by trying out the following:

```
# Improve display...
summary.mylm <- function(out){
     print("")
     print("Coefficients:")
     print(paste("(Intercept)          x",sep=""))
     print(c(out$myintercept,out$mycoef))
```

```
}

# Test it!
summary.mylm( mylm(faithful$waiting,faithful$eruptions) )

# Improve (i.e. rewrite) this function to allow for plotting!
mylm <- function(x, y, DOPLOT=FALSE, ...){
     ...
     out = list(myintercept=a, mycoef=b, residuals=res)
     if(DOPLOT){
          plot(x, y, cex=1.2, ...)
          abline(a=a, b=b, col='red', lwd=1.5)
     }
     return(out)
}
# Test again!
mylm(faithful$waiting,faithful$eruptions)
mylm(faithful$waiting,faithful$eruptions,DOPLOT=T)
```

NB: the definition of function `mylm` that R will use is whichever one of the two definitions above that gets "run" last...

Question 3.2

Simulate a model with observations given for $i = 1, \ldots, N$ by

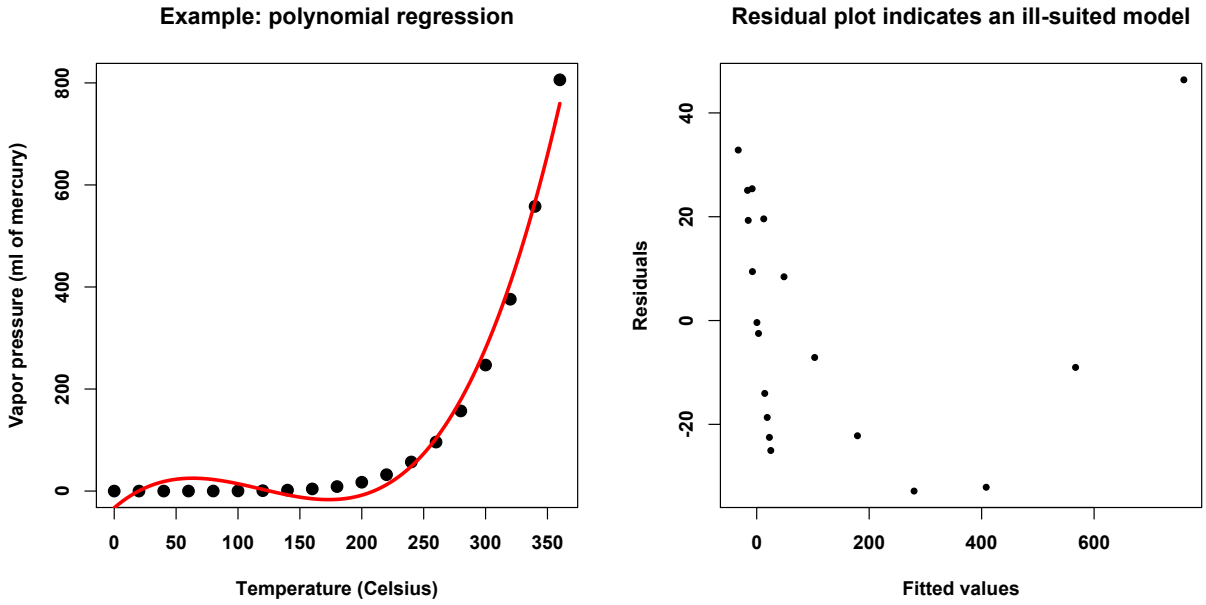
$$Y_i = \theta_0 + \theta_1 X_i + \varepsilon_i$$

where $X = \{1, 2, 5, 5.5, 9\}$, $\theta_0 = 3$, $\theta_1 = 1.5$, and $\varepsilon \sim N(0, 1.2)$, and using $N = 100$. Apply ordinary least squares estimation to your set of simulated observations. Then apply weighted least squares with weights $w = (.1, .1, .35, ..35, .1)$ and compare outputs. Comment on your results and how the comparison was carried out.

3.2 Polynomial and nonlinear regression

Question 3.3

`attach(pressure); x=temperature; y=pressure`
Implement the linear regression to obtain the following plot:



Question 3.4

Why does the following not "work"? Hint: complete the code to plot the objective function.

```
par(mfrow=c(1,2),font.lab= 2, font.axis=2)
model <- function(x, theta){
     return(theta*x)
}
crit <- function(theta,x,y){
# must have theta as 1st parameter and return a single value...
     return( sum( y-model(x,theta) ) )
}

thbar = 1.8
x = rep(c(1,3,7,8), len=100)
y = model(x,thbar) + rnorm(x)

plot(x, y, pch=20, cex=1.5, main="optim example 1: data")
points(x,model(x,thbar),col='green',pch=20,cex=2)

(optim.out <- optim(par=c(1), fn=crit, x=x, y=y,
     method="L-BFGS-B", bluelower=c(0.01), upper=c(3.05)))
```

Question 3.5

Implement a simulation of the nonlinear model

$$Y = \exp(-\theta X) + \varepsilon$$

and minimize the sum of least squares for this model using `optim`.

Question 3.6

Implement the same simulation and minimize the sum of least squares for this model using `optimx`, this time.

3.3 Regularisation

Question 3.7

The `Blood Pressure.txt` dataset contains measurements of systolic blood pressure, age, waist circumference, cholesterol and BMI index, for 75 subjects (variable `PatientID` is dropped from the dataset). Four linear regression models have been fit to the dataset in order to describe the variable of interest systolic blood pressure in terms of the other 4 variables. Regularisation parameters have been calibrated so that their corresponding models yield optimal regularisation. Table 3.7 below presents output from these 4 model fits (where "e-net" stands for "elastic net"). Analyse and comment on these results, for example in terms of:

- The effect of each of the regularisation schemes;
- Increase in model fit error;
- Potential "issues" or "challenges" within the dataset;
- Overall impact of, or necessity for regularisation for this data;
- Limitations of the output presented.

	GLM	ridge	e.net	LASSO
(Intercept)	56.710	64.703	65.854	64.707
Age	0.200	0.220	0.146	0.118
Waist	0.557	0.356	0.482	0.520
Cholesterol	0.003	0.004	0.000	0.000
BMI	0.030	0.378	0.048	0.000
Errors	139.76	141.96	143.26	143.26

Table 1: Table for Question 1.

4 Smoothing

4.1 Splines

Question 4.1 (NB: For this one we need to re-generate the dataset)

Read in the dataset `rough_Makeham_rates.csv`, which contains a sample of simulated Makeham mortality rates with additive noise. For information, the procedure used to generate this sample is as follows:

```
LT = read.table("irl_lifetable_2005.txt", sep=",", header=TRUE)
# keep only the first 106 rows from LT:
SLT = LT[c(1:106),]
mx = SLT[,8]
x = SLT$Age # age grid
# roughly fit a Makeham model to this data:
onls = nls(mx~A+B*c^x,start=list(A=.0003, B=.00002, c=1.108))
ABc = summary(onls)$coef[,1]
# now add noise to the fitted f.o.m. curve:
set.seed(1)
x = seq(0, 110, by=1)
mx = ABc[1]+ABc[2]*ABc[3]^x
mxn = mx
s1 = which(x<86)
s2 = which(x>85)
mxn[s1] = pmax(0.005,mx[s1]+rnorm(length(s1),0,.03))
mxn[s2] = mx[s2]+rnorm(length(s2),0,.06)
dat = cbind(x,mx,mxn)
```

This means that the data comes from a Makeham model of mortality rates μ_x (as a function of age x)

$$\mu_x = A + Bc^x, \quad x = 0, 1, \ldots$$

with true parameters $A = .0003$, $B = .00002$, $c = 1.108$, which was perturbed by some mostly-symmetric additive noise.

(a) Fit a Makeham model to the sample, using `nls()` with initial values `(A=.0003, B=.00002, c=1.108)`.

(b) Repeat the previous step, but on the smoothed curve obtained from a generic P-spline. Compare the mean squared errors of the model parameter estimates obtained from both nonlinear fits.

Question 4.2

Import the simulated dataset of female mortality rates from a hypothetical cohort of life insurance policyholders into R from `insdata.csv`:

```
dat = read.csv("insdata.csv")
age = dat$Age
mF = dat$mF
```

(a) Compute a first P-spline where the smoothing control parameter is set by cross-validation. Create a plot of the dataset (black dots) along with the P-spline (red solid curve).

(b) Compute a second P-spline for the same dataset, where the smoothing control parameter is half that of the P-spline obtained in (a).

(c) Show that the two P-spline outputs are evaluated over the same points on the x-axis.

(d) Compute and compare the MSE's for the P-splines obtained in (a) and (b). Comment on their difference, and propose a reason as to why they differ.

(e) Compute a B-spline basis using the first, second and third quartiles of the age data as knots. Create a plot of this B-spline basis.

(f) Quote the coordinates of a policyholder aged 60 on the B-spline basis computed in (e), up to four decimal places. Indicate these coordinates with a line on the plot obtained in (e).

(g) Compute the corresponding B-spline for the (age, mF) data. Find the value of the output coefficients for the B-spline expression.

(h) Compare and comment on the MSE obtained for that B-spline with the MSE's obtained from the two P-splines obtained in (d).

(i) Compute interpolations for all ages within the range of age data, using respectively P-spline smoothing and local polynomial regression. Plot the interpolated points over the observations, using red for P-spline values and blue for local polynomial regression values. Note the standard deviations of each of the interpolated samples.

Question 4.3

Fit a B-spline and a P-spline to the `Prestige` dataset (`library(cars)`). You'll need to run

```
library(splines)
library(car)
```

Plot the corresponding curves and compare the MSEs.