R-Code from Mahler's Guides to **CAS Exam MAS-1**

prepared by
Howard C. Mahler, FCAS
Copyright ©2026 by Howard C. Mahler.

Howard Mahler
hmahler@mac.com
www.howardmahler.com/Teaching

For many examples in my MAS-1 Study Guides, I provide corresponding R-Code.
The order presented here is the same as that of my Study Guides.
(Not every Study Guide has any R-Code.)

You will not be asked to write R-Code on your exam.

Some of you will benefit from running R-Code, either mine or that in the textbooks, in order to get a better understanding of how the various algorithms operate.

I have tried to keep my R-Code simple.
I have not tried to produce the most elegant or efficient code.

Please let me know of any suspected errors.
hmahler@mac.com

My R-Code for the Example with Cars:

```
plot(cars$speed, cars$dist, cex=0.5, xlab="speed", ylab="distance")
plot(ksmooth(cars$speed, cars$dist,"normal", bandwidth=0.67449*4),
                type="l", xlab="speed", ylab="smoothed distance")  # sigma = 1
plot(ksmooth(cars$speed, cars$dist,"normal", bandwidth=0.67449*8),
                type="l", xlab="speed", ylab="smoothed distance")  # sigma = 2
```

My R-Code for My Automobile Collision Insurance Example:

```
install.packages("insuranceData")  # you only have to install a package once
library(insuranceData)

data(AutoClaims)
head(AutoClaims)

boxplot(log(PAID) ~ CLASS, data=AutoClaims, las=2)
boxplot(log(PAID) ~ STATE, data=AutoClaims, las=2,
                        names=c("1","2","3","4","6","7","10","11","12","13","14","15","17"))

table(AutoClaims$STATE)
table(AutoClaims$CLASS)

model<-lm(log(PAID) ~ relevel(as.factor(STATE),ref="STATE 02")
                              + relevel(as.factor(CLASS),ref="C11"), data=AutoClaims)
summary(model)
```

My R-Code for the BMI Example:

```
install.packages("MASS") # you only have to install a package once
library(MASS)
install.packages("car") # you only have to install a package once
library(car)
install.packages("MPV") # you only have to install a package once
library(MPV)
install.packages("lmtest")
library(lmtest)

# the data is on the website of Predictive Modeling Applications in Actuarial Science
# https://instruction.bus.wisc.edu/jfrees/jfreesbooks/PredictiveModelingVol1/index.htm
# you are interested in Chapter 2 of Volume 1
# you need to download the data from their website onto your computer
# then exactly how you get the data into R depends on you operating system
# how I read the data into R on my Mac after downloading the data from their website
dat <- read.csv(file.choose())
# you need to point to the location on your computer of the file MEPS.Diabetes.csv

dat$MALE <- I(dat$FEMALE==0)*1
train <- dat[dat$PANEL==13,]
test <- dat[dat$PANEL!=13,]

plot(BMI~AGE, data=train, cex=0.3)
m1 <- lm(BMI~AGE, train)
abline(m1, lwd=2, col="blue")

boxplot(BMI~MALE, train, las=1, varwidth=TRUE, names=c("Female","Male"), cex.axis=1.2)

m9 <- lm(BMI ~ AGE + MALE + RACE + LOGINCOME + I(AGE^2) + MALE:RACE, train)
summary(m9)
var(train$BMI) * (nrow(train)-1)  # the Total Sum of Squares
anova(m9)

logLik(m9)
AIC(m9)
vif(m9,type="predictor")
```

```
leverage9<-hatvalues(m9)
plot(leverage9, cex=0.3)
p <- length(coef(m9))
n <- nrow(train)
threshold <- 2 * (p / n) # threshold is taken equal to twice the mean leverage
abline(h=threshold, lwd=2, col="blue")
sum(leverage9 > threshold)
sum(leverage9 > 1.5*threshold)

plot(cooks.distance(m9), cex=0.3)

sr.9 <- rstandard(m9)
sum(sr.9>2)
sum(abs(sr.9)>2)
sum(sr.9>3)
sum(abs(sr.9)>3)

plot(sr.9, cex=0.3)
abline(h=c(-2,0,2), lwd=2, col="blue")
bptest(m9)

plot(train$AGE, sr.9,  cex=0.3, xlab="AGE", ylab="")
abline(h=c(-2,0,2), lwd=2, col="blue")
lines(lowess(train$AGE,sr.9), lwd=4, col="slategrey")

m10 <- lm(BMI ~ AGE + MALE + RACE + LOGINCOME + MALE:RACE, train)
sr.10 <- rstandard(m10)
plot(train$AGE, sr.10, cex=0.3, xlab="AGE", ylab="")
abline(h=c(-2,0,2), lwd=2, col="blue")
lines(lowess(train$AGE,sr.10), lwd=4, col="slategrey")

plot(train$INCOME, sr.9,  cex=0.3, xlab="Income", ylab="")
abline(h=c(-2,0,2), lwd=2, col="blue")
lines(lowess(train$INCOME,sr.9), lwd=4, col="slategrey")

ptrain9<-predict(m9, train)
plot(ptrain9, sr.9,  cex=0.3, xlab="Predicted", ylab="")
abline(h=c(-2,0,2), lwd=2, col="blue")
lines(lowess(ptrain9,sr.9), lwd=4, col="slategrey")

qqnorm(sr.9, cex=0.3)
qqline(sr.9, col="blue", lwd=2)
shapiro.test(sr.9)

ptest9<-predict(m9, test)
sum((test$BMI-ptest9)^2)  # the sum of squared prediction errors on the test data

PRESS(m9) # predicted residual error sum of squares
```

My R-Code for the Medical Expenditure Example:

```
install.packages("MASS") # you only have to install a package once
library(MASS)
install.packages("car") # you only have to install a package once
library(car)
install.packages("MPV") # you only have to install a package once
library(MPV)
install.packages("psych") # you only have to install a package once
library(psych)
install.packages("Hmisc") # you only have to install a package once
library(Hmisc)
install.packages("lmtest") # you only have to install a package once
library(lmtest)

# the data is on the website of Predictive Modeling Applications in Actuarial Science
# https://instruction.bus.wisc.edu/jfrees/jfreesbooks/PredictiveModelingVol1/index.htm
# you are interested in Chapter 2 of Volume 1
# you need to download the data from their website onto your computer
# then exactly how you get the data into R depends on you operating system
# how I read the data into R on my Mac after downloading the data from their website
dat <- read.csv(file.choose())
# you need to point to the location on your computer of the file MEPS.Diabetes.csv

dat$MENTHEALTH <- factor(dat$MENTHEALTH)
summary(dat$MENTHEALTH)

train <- dat[dat$PANEL==13,]
holdout <- dat[dat$PANEL==14,]
nrow(dat); nrow(train); nrow(holdout)

hist(log(dat$EXPEND))
shapiro.test(log(dat$EXPEND))

cont <- c("INCOME", "LOGINCOME", "BMI", "AGE", "COMORB.CNT", "LOGEXPEND")
pairs.panels(dat[,cont], pch='.', gap=0, ellipses=F, cex=0.3, lwd=2, hist.col='aliceblue')

temp <- as.matrix(dat[,cont])
rcorr(temp, type="pearson")

plot(dat$LOGEXPEND~dat$AGE, cex=0.3, xlab="AGE", ylab="LOG EXPENDITURE")
lines(lowess(dat$AGE,dat$LOGEXPEND), lwd=4, col="slategrey")
```

```
train$AGE.C <- train$AGE - 60
train$BMI.C <- train$BMI - 32
cs3b <- lm(LOGEXPEND ~ AGE.C + AGE.C:STROKE + BMI.C + RACE + HISPANIC
            + EMPLOYED + UNINSURED + MENTHEALTH + EMPHYSEMA + STROKE
            + CORONARY + CHOLEST + CANCER + ASTHMA + HIGHBP, data = train)

summary(cs3b)
anova(cs3b)

sr.3 <- rstandard(cs3b)

ptrain.3<-predict(cs3b,data=train)

plot(ptrain.3, sr.3,  cex=0.3, xlab="Predicted", ylab="")
abline(h=c(-2,0,2), lwd=2, col="blue")
lines(lowess(ptrain.3,sr.3), lwd=4, col="slategrey")

bptest(cs3b)

vars <- c("RACE","HISPANIC","EMPLOYED","UNINSURED", "MENTHEALTH",
      "EMPHYSEMA","STROKE","CORONARY","CHOLEST","CANCER","ASTHMA","HIGHBP")
vars <- vars[12:1]
par(mfrow=c(3,4), oma=c(6,4,1,1), mgp=c(2,.7,0), tck=-.01, mar=c(1,0,2,0))
for (i in 1:length(vars)){
 if (i != 12) boxplot(sr.3~train[,vars[i]], ylim=c(-3,3), yaxt="n", cex.axis=1.6)
 if (i == 12) boxplot(sr.3~train[,vars[i]], ylim=c(-3,3), yaxt="n", las=2, cex.axis=1.6)
 if (i %in% c(1,5,9)) axis(side=2, at=seq(-3,3,1), labels=T, cex.axis=1.6, las=2)
abline(h=median(sr.3))
title(vars[i], line=-1.2, cex.main=1.6)}
mtext("LOGEXPEND", side=2, outer=T, line=3, cex=1.2)

dev.off()

plot(train$AGE, sr.3,  cex=0.3, xlab="Age", ylab="")
abline(h=c(-2,0,2), lwd=2, col="blue")
lines(lowess(train$AGE,sr.3), lwd=4, col="slategrey")

plot(train$BMI , sr.3,  cex=0.3, xlab="BMI", ylab="")
abline(h=c(-2,0,2), lwd=2, col="blue")
lines(lowess(train$BMI,sr.3), lwd=4, col="slategrey")
```

```
vars <- c("RACE","HISPANIC","EMPLOYED","UNINSURED", "MENTHEALTH",
  "EMPHYSEMA","STROKE","CORONARY","CHOLEST","CANCER","ASTHMA","HIGHBP")
vars <- vars[12:1]
par(mfrow=c(3,4), oma=c(6,4,1,1), mgp=c(2,.7,0), tck=-.01, mar=c(1,0,2,0))
for (i in 1:length(vars)){
 if (i != 12) boxplot(sr.3~train[,vars[i]], ylim=c(-3,3), yaxt="n", cex.axis=1.6)
 if (i == 12) boxplot(sr.3~train[,vars[i]], ylim=c(-3,3), yaxt="n", las=2, cex.axis=1.6)
 if (i %in% c(1,5,9)) axis(side=2, at=seq(-3,3,1), labels=T, cex.axis=1.6, las=2)
abline(h=median(sr.3))
title(vars[i], line=-1.2, cex.main=1.6)}

dev.off()

qqnorm(sr.3, cex=0.3)
qqline(sr.3, col="blue", lwd=2)
shapiro.test(sr.3)

# model without centering AGE and BMI
cs3 <- lm(LOGEXPEND ~ AGE + AGE:STROKE + BMI + RACE + HISPANIC
            + EMPLOYED + UNINSURED + MENTHEALTH + EMPHYSEMA + STROKE
            + CORONARY + CHOLEST + CANCER + ASTHMA + HIGHBP, data = train)
yhat <- predict(update(cs3, data=holdout))
LOGEXPEND <- holdout$LOGEXPEND
plot(LOGEXPEND,yhat, cex=0.5, ylab="Predicted LOGEXPEND", xlab="Actual LOGEXPEND")
abline(lm(yhat~LOGEXPEND), lwd=2, col="blue")
cor(LOGEXPEND,yhat);cor.test(LOGEXPEND,yhat)

holdout_sorted <- holdout[order(yhat), ]
cum_losses <- cumsum(holdout_sorted$EXPEND) / sum(holdout_sorted$EXPEND)
cum_policies<-c(0,1:length(cum_losses))/length(cum_losses)
cum_losses <- c(0,cum_losses)
# Plotting the Lorenz Curve
plot(cum_policies, cum_losses, type = "l", col = "red", lwd = 2,
       xlab = "Cumulative % of Observations", ylab = "Cumulative % of Expenditures")
abline(0, 1, col = "blue", lwd = 2)
1 - 2*mean(cum_losses)  # Gini Index

# assuming a perfect model
holdout_sorted2 <- holdout[order(holdout$EXPEND), ]
cum_losses2 <- cumsum(holdout_sorted2$EXPEND) / sum(holdout_sorted2$EXPEND)
cum_policies2<-c(0,1:length(cum_losses2))/length(cum_losses2)
cum_losses2 <- c(0,cum_losses2)
# Plotting the Lorenz Curve
plot(cum_policies2, cum_losses2, type = "l", col = "red", lwd = 2,
       xlab = "Cumulative % of Observations", ylab = "Cumulative % of Expenditures")
abline(0, 1, col = "blue", lwd = 2)
1 - 2*mean(cum_losses2)  # Gini Index
```

My R-Code for My Private Passenger Automobile Collision Insurance Severity Example:

```
install.packages("insuranceData")  # you only have to install a package once
library(insuranceData)

data(AutoCollision)
head(AutoCollision)
# DriveShort ↔ Drive to Work < 10 miles
# DriveLong ↔ Drive to Work > 10 miles
# Age A ↔ Age Group 17-20
# Age B ↔ Age Group 21-24

model<-lm(log(Severity) ~ relevel(as.factor(Age), ref="F") +
                relevel(as.factor(Vehicle_Use),ref="DriveShort"), data=AutoCollision,
                weights=Claim_Count)
summary(model)

exp(predict(model))  # the fitted severities by cell
```

My R-Code for A Simple Automobile Insurance Example:

```
claims<-10*c(1,37 ,42, 14, 73, 101)
expos<-10*c(100, 1200, 500, 300, 500, 400)
Age<-c(1, 1, 1, 2, 2, 2)
Size<-c("L", "M", "S", "L", "M", "S")

model<-glm(claims~factor(Age) + relevel(factor(Size),ref="M"), family=poisson,
                    offset=log(expos))
# have the model treat Age and Size as categorical, with M being the base level for Size
summary(model)

exp(predict(model))  # the fitted numbers of claims
```

My R-Code for an Example with Automobile Insurance Data:

```
install.packages("insuranceData")  # you only need to install a package once
library(insuranceData)

data(dataCar) # we will use dataCar
head(dataCar)

table(dataCar$agecat)  # get the number of policies by age category
summary(dataCar$veh_value)
table(dataCar$veh_body)
sum(dataCar$numclaims)/sum(dataCar$exposure)  # the overall mean frequency per caryear

freq.model<-glm(numclaims~relevel(factor(agecat),ref="3") + veh_value
        + relevel(factor(veh_body),ref="SEDAN"), family=poisson(link="log"),
        offset=log(exposure), data=dataCar)
# relevel redefines the base level for a categorical variable

summary(freq.model)
```

My R-Code for My Example with Motorcycle Insurance Data:

```
install.packages("insuranceData")  # you only have to install a package once
library(insuranceData)

data(dataOhlsson)
head(dataOhlsson)

dataOhlsson<-dataOhlsson[-which(dataOhlsson$duration==0),]
# removing records with duration of zero
dataOhlsson<-dataOhlsson[-dataOhlsson$agarald<16,]
#removing records with owners age < 16

model <-glm(antskad ~ agarald + relevel(as.factor(zon),ref="4") +
        relevel(as.factor(mcklass),ref="3") + fordald + offset(log(duration)), data = dataOhlsson,
        family = poisson)
summary(model)

predfreq<-exp(predict(model))  # fitted frequencies
head(predfreq)
```

My R-Code for the Counterfeit Currency Example:

```
# the data is on the website of Predictive Modeling Applications in Actuarial Science
# https://instruction.bus.wisc.edu/jfrees/jfreesbooks/PredictiveModelingVol1/index.htm
# you are interested in Chapter 14 of Volume 1
# you need to download the data from their website onto your computer
# then exactly how you get the data into R depends on you operating system
# how I read the data into R on my Mac after downloading the data from their website
mxbills = read.csv(file.choose())
# you need to point to the location on your computer of the file MXbills.csv

attach(mxbills)
dim(mxbills)
mxbills

observed<-c(F50/C50, F100/C100, F200/C200, F500/C500)
time<-c(0:11, 0:11, 0:11, 0:11)

den=c(rep("50",12), rep("100",12), rep("200",12), rep("500",12))
den<-as.factor(den)
den<-relevel(den, ref="50")  # making 50 Pesos the base level

y<-1000* cbind(c(F50,F100,F200,F500), c(C50-F50,C100-F100,C200-F200,C500-F500))
# the first column is the number of fakes, while the second column is the number of non-fakes

counterfeit.model<-glm(y ~ den + time, family=binomial)
summary(counterfeit.model)

fitted<-1/(1+exp(-predict(counterfeit.model)))  # converting the linear predictor to probabilities

plot(observed, fitted, cex=0.5, xlab="observed rate", ylab="fitted rate")
abline(c(0,1), lwd=2, col="blue")
```

My R-Code for the Titanic Example:

```
install.packages("stats") # you only have to install a package once
library(stats) # so that one can run the function glm

titanic_df <- as.data.frame(Titanic)
titanic_df$Class<-as.factor(titanic_df$Class)
titanic_df$Sex<-as.factor(titanic_df$Sex)
titanic_df$Age<-as.factor(titanic_df$Age)
titanic_df$Survived<-as.factor(titanic_df$Survived)

model<-glm(Survived~Class+Sex+Age, data=titanic_df, family=binomial, weights=Freq)
summary(model)

predict(model, type="response") # includes rows with no observations
```

My R-Code for My Attorney Involvement Example:

```
install.packages("insuranceData")  # you only have to install a package once
library(insuranceData)

data(AutoBi)
head(AutoBi)

AutoBi$CLMINSUR=2-AutoBi$CLMINSUR  # converted this so that no = 0 and yes = 1
AutoBi$ATTORNEY=2-AutoBi$ATTORNEY  # converted this so that no = 0 and yes = 1

model.logit<-glm(ATTORNEY ~ CLMAGE + CLMSEX + CLMINSUR + log(LOSS),
                 family=binomial, data=AutoBi)
summary(model.logit)

model.probit<-glm(ATTORNEY ~ CLMAGE + CLMSEX + CLMINSUR + log(LOSS),
                 family=binomial(link="probit"), data=AutoBi)
summary(model.probit)

pred.logit<-predict(model.logit)
prob.logit<-exp(pred.logit) / (1+exp(pred.logit))
head(prob.logit)

pred.probit<-predict(model.probit)
prob.probit<-pnorm(pred.probit)
head(prob.probit)

plot(prob.logit, prob.probit, cex=0.2, xlab="probability with logit link",
                 ylab="probability with probit link")
abline(c(0,1), lwd=3, col="blue")
```

My R-Code for This Example of GLM with a Nominal Response Variable:

```
install.packages("mlogit") # you only have to install a package once
library(mlogit)

# the data is on the website of Predictive Modeling Applications in Actuarial Science
# https://instruction.bus.wisc.edu/jfrees/jfreesbooks/PredictiveModelingVol1/index.htm
# you are interested in Chapter 3 of Volume 1
# you need to download the data from their website onto your computer
# then exactly how you get the data into R depends on you operating system
# how I read the data into R on my Mac after downloading the data from their website
Vehown = read.csv(file.choose())
# you need to point to the location on your computer of the file VehOwned.csv

attach(Vehown)
Vehown$veh<-as.factor(Vehown$veh)
mldata<-mlogit.data(Vehown, varying=NULL, choice="veh", shape="wide")
head(mldata)

glogit.model<- mlogit(veh~1|men+age+urban, data = mldata, reflevel="C")
summary(glogit.model)
```

My R-Code for This Example of a Proportional Odds Model:

```
install.packages("MASS") # you only have to install a package once
library(MASS)

# the data is on the website of Predictive Modeling Applications in Actuarial Science
# https://instruction.bus.wisc.edu/jfrees/jfreesbooks/PredictiveModelingVol1/index.htm
# you are interested in Chapter 3 of Volume 1
# you need to download the data from their website onto your computer
# then exactly how you get the data into R depends on you operating system
# how I read the data into R on my Mac after downloading the data from their website
Fullcov =  read.csv(file.choose())
# you need to point to the location on your computer of the file TypeofCoverage.csv

# fitting an initial model
ologit<-polr(ordered(yord) ~ men + urban + private + marital_M + marital_O + age + seniority,
        data = Fullcov)
summary(ologit)
# fitting another model excluding non-significant variables
ologit2<-polr(ordered(yord) ~ urban + private + seniority, data = Fullcov)
summary(ologit2)
# the function polr subtracts the linear predictor while the textbook adds the linear predictor
# thus one has to reverse the signs of the coefficients of the predictor variables
# in order to match the presentation in the textbook

# the predicted values are in ologit2$fitted.values
# with the fitted probabilities for the three categories in three columns
# for example, the results of the above exercise are in ologit2$fitted.values[900,]
# for the observation Fullcov[900,]
```

My R-Code for my Example with Automobile Pure Premiums:

```
# the data is on the website of Predictive Modeling Applications in Actuarial Science
# https://instruction.bus.wisc.edu/jfrees/jfreesbooks/PredictiveModelingVol1/index.htm
# you are interested in Chapter 10 of Volume 1
# you need to download the data from their website onto your computer
# then exactly how you get the data into R depends on you operating system
# how I read the data into R on my Mac after downloading the data from their website
claim <- read.table(file.choose(), header=T)
# you need to point to the location on your computer of the file claim.txt

claim2 <- subset(claim,YEAR<1998)  # excluding the data for year 1998
attach(claim2)
PP <- AC
pci <- PCI/1e3  # putting the per capita income per 1000
logppsm <- log(PPSM)  # taking the log of the people per square mile
year <- YEAR - 1992  # year 1993 will correspond to 1

mass.model <- glm(PP ~ pci+ I(pci^2) + logppsm + year, family=Gamma(link="log"))
summary(mass.model)

plot(fitted(mass.model), residuals(mass.model),
        ylab="Deviance Residual", xlab="Fitted Value", cex=0.5)
# the default in R for a glm object is deviance residuals

qqnorm(residuals(mass.model), cex=0.3)
qqline(residuals(mass.model), col="blue", lwd=2)
shapiro.test(residuals(mass.model))

claim3 <- subset(claim,YEAR > 1997)
claim3$PP<- claim3$AC
claim3$pci <- claim3$PCI/1e3
claim3$logppsm <- log(claim3$PPSM)
claim3$year <- 6  # year 1998 corresponds to 6

fitted1998=exp(5.789776 - 0.166614 * claim3$pci + 0.003036*claim3$pci^2
        + 0.173165 * claim3$logppsm + 0.029371*6)

plot(claim3$PP, fitted1998, xlab="observed PP", ylab="fitted PP")
abline(c(0,1),col="blue", lwd=2)

mean(subset(claim,YEAR == 1993)$AC)  # mean pure premiums by year
mean(subset(claim,YEAR == 1994)$AC)
mean(subset(claim,YEAR == 1995)$AC)
mean(subset(claim,YEAR == 1996)$AC)
mean(subset(claim,YEAR == 1997)$AC)
mean(subset(claim,YEAR == 1998)$AC)
```

My R-Code for my Example with Medical Expenditures:

```
# the data is on the website of Predictive Modeling Applications in Actuarial Science
# https://instruction.bus.wisc.edu/jfrees/jfreesbooks/PredictiveModelingVol1/index.htm
# you are interested in Chapter 10 of Volume 1
# you need to download the data from their website onto your computer
# then exactly how you get the data into R depends on you operating system
# how I read the data into R on my Mac after downloading the data from their website
office <- read.table(file.choose(), header=TRUE)
# you need to point to the location on your computer of the file officeout.txt

attach(office)
nrow(office)
head(office)

gamma.model<-glm(OBVEXP08 ~ age + female + married+ hispanic + black + asian + edu +
                              familysize +msa + limitation + chronic + factor(health) +
                              hmo + employed, family=Gamma(link="log"), data=office)
summary(gamma.model)

# model excluding four variables that are not significant
gamma.model2<-glm(OBVEXP08 ~ age + female + married + hispanic + black + asian + edu +
                              limitation + chronic+ factor(health), family=Gamma(link="log"),
                              data=office)
summary(gamma.model2)

anova(gamma.model2, gamma.model, test="F")
# F-Test of nested models
# the p-value turns out to be 64.6%
# thus we do not reject the null hypothesis of the simpler model
```

My R-Code for the Simulation Experiments:

```
install.packages("MASS")  # you only need to install a package once
library(MASS)
set.seed(42)
n<-500
p<-5
CovMatrix<-outer(1:p, 1:p, function(x,y) {ifelse(x==y,1,0)})
x<-mvrnorm(n,rep(0,p),CovMatrix)
y<-10*x[,1]-8*x[,2]+6*x[,3]-4*x[,4]+2*x[,5]+rnorm(n)

install.packages("glmnet")  # you only need to install a package once
library(glmnet)
fit.Ridge<-glmnet(x, y, family="gaussian", alpha=0)
plot(fit.Ridge, xvar="lambda", label=TRUE)
abline(a=0, b=0)

fit.Lasso<-glmnet(x, y, family="gaussian", alpha=1)
plot(fit.Lasso, xvar="lambda", label=TRUE)
abline(a=0, b=0)

p<-10
x<-matrix(rnorm(n*p), nrow=n, ncol=p)
y<-10*x[,1]-8*x[,2]+6*x[,3]-4*x[,4]+2*x[,5]+rnorm(n)

fit.enet<-glmnet(x, y, family="gaussian", alpha=0.5)
plot(fit.enet, xvar="lambda", label=TRUE)
abline(a=0, b=0)

CovMatrix<-outer(1:p, 1:p, function(x,y) {ifelse(x==y,1,0)})
CovMatrix[1,2]<-0.9
CovMatrix[2,1]<-0.9
x<-mvrnorm(n, rep(0,p), CovMatrix)
y<-3*x[,1]+5*rnorm(n)

fit.enet2<-glmnet(x, y, family="gaussian", alpha=0.2)
plot(fit.enet2, xvar="lambda", label=TRUE)
abline(a=0, b=0)
```

My R-Code for Global Annual Temperature, Splines Example:[1]

```
years = 1752 + 1:268

temp=c(-0.803, -0.275, -0.597, -0.319, -0.164, -2.271, -1.243, -2.137, -0.584, -0.699, -1.708,
-0.641, -0.417, -0.444, -0.562, -1.941, -0.993, -0.988, -1.056, -0.506, -0.328, 0.086, 0.141,
-0.928, -0.941, -0.720, -0.301, 0.313, -0.461, -0.694, -0.894, -0.671, -0.916, -0.570, -0.484,
-0.294, -0.188, -0.475, -0.181, -0.407, -0.263, 0.090, 0.185, -0.212, -0.008, -0.052, -0.552,
-0.431, 0.251, 0.093, -0.101, -0.009, -0.076, -0.179, -0.483, -1.068, -1.502, -1.618, -1.645,
-1.499, -0.919, -1.026, -1.439, -1.776, -1.534, -0.506, -1.336, -1.049, -0.754, -0.663, -0.938,
-0.315, -0.485, -0.297, 0.071, -0.618, -0.864, -0.207, -1.050, -1.234, -0.677, -0.570, -1.312,
-1.049, -1.310, -1.198, -0.982, -0.934, -1.114, -0.712, -0.571, -1.082, -0.933, -0.164, -0.652,
-0.733, -0.742, -0.852, -0.574, -0.679, -0.677, -0.489, -0.555, -0.637, -0.883, -0.480, -0.298,
-0.699, -0.759, -1.126, -0.489, -0.648, -0.436, -0.335, -0.167, -0.449, -0.206, -0.501, -0.625,
-0.546, -0.367, -0.241, -0.746, -0.518, -0.102, 0.263, -0.441, -0.553, -0.321, -0.480, -0.569,
-0.777, -0.626, -0.634, -0.667, -0.452, -0.256, -0.626, -0.565, -0.530, -0.535, -0.400, -0.402,
-0.348, -0.265, -0.406, -0.155, -0.042, -0.037, -0.294, -0.372, -0.507, -0.351, -0.194, -0.609,
-0.398, -0.411, -0.370, -0.425, -0.453, -0.315, -0.005, -0.042, -0.398, -0.626, -0.505, -0.254,
-0.282, -0.066, -0.239, -0.221, -0.150, -0.132, 0.052, -0.173, -0.066, -0.473, -0.092, 0.005,
-0.008, -0.361, -0.085, -0.204, -0.166, -0.013, 0.164, 0.073, 0.065, 0.053, 0.022, 0.058, 0.149,
-0.114, -0.018, 0.106, 0.043, -0.116, -0.320, -0.060, -0.038, 0.194, -0.115, -0.051, -0.395, -0.005,
0.088, 0.077, -0.097, 0.121, 0.061, 0.184, -0.276, -0.144, -0.058, 0.043, -0.155, -0.066, 0.056,
-0.042, -0.149, 0.325, -0.168, 0.119, -0.268, 0.255, 0.087, 0.107, 0.373, 0.549, 0.045, 0.429,
0.080, 0.058, 0.246, 0.408, 0.608, 0.335, 0.643, 0.592, 0.229, 0.276, 0.446, 0.750, 0.438, 0.585,
0.924, 0.676, 0.595, 0.804, 0.971, 0.913, 0.723, 1.106, 0.936, 1.133, 0.837, 0.931, 1.104, 0.929,
0.888, 1.005, 0.979, 1.238, 1.445, 1.322, 1.162, 1.366, 1.525)

library(splines)

plot(years, temp, cex =.5)
fit.ns=lm(temp~ns(years , df =10))
pred.ns=predict(fit.ns ,newdata = list(years), se=T)
lines(years , pred.ns$fit , lwd =2, col="red")
lines(years , pred.ns$fit +2* pred.ns$se , lty ="dashed", col="red")
lines(years , pred.ns$fit -2* pred.ns$se , lty ="dashed", col="red")

fit.sscv=smooth.spline(years , temp , cv=TRUE)
# degrees of freedom chosen via cross-validation
plot(years, temp, cex =.5)
lines(fit.sscv , col ="red ", lwd =2)
fit.sscv$lambda

fit.ss15=smooth.spline(years , temp , df=15)  # 15 degrees of freedom
plot(years, temp, cex =.5)
lines(fit.ss15 , col ="blue", lwd =2)
```

---

[1] Similar to Section 7.8.2 in An Introduction to Statistical Learning.

My R-Code for my Example with Automobile Pure Premiums:

```
# the data is on the website of Predictive Modeling Applications in Actuarial Science
# https://instruction.bus.wisc.edu/jfrees/jfreesbooks/PredictiveModelingVol1/index.htm
# you are interested in Chapter 10 of Volume 1
# you need to download the data from their website onto your computer
# then exactly how you get the data into R depends on you operating system
# how I read the data into R on my Mac after downloading the data from their website
claim = read.table(file.choose(), header=T)
# you need to point to the location on your computer of the file claim.txt

head(claim)

claim2 <- subset(claim,YEAR<1998)  # excluding the data for year 1998
attach(claim2)
PP <- AC
pci <- PCI/1e3  # putting the per capita income per 1000
logppsm <- log(PPSM)  # taking the log of the people per square mile
year <- YEAR - 1992  # year 1993 will correspond to 1

install.packages("mgcv") # you only have to install a package once
library(mgcv)

mass.gam <- gam(PP ~ s(pci) + logppsm + year, family=Gamma(link="log"))
# one could instead specify the degrees of freedom of the smoothing spline
# with instead for example s(pci, 10)
summary(mass.gam)

plot.gam(mass.gam, se=TRUE, col="blue")
mass.gam$sp  # to get the smoothing parameter lambda

mass.glm <- glm(PP ~ pci + I(pci^2) + logppsm + year, family=Gamma(link="log"))
summary(mass.glm)

predict.glm<-predict(mass.glm)
predict.gam<-predict(mass.gam)

plot(predict.glm, predict.gam, cex=0.3, xlab="Predictions of GLM", ylab="Predictions of GAM")
abline(c(0,1), col="blue", lwd=2)
```