anscombe <- read.delim("C:/Teaching@cofc/Math 550/Chapter 3/anscombe.txt", header = TRUE)

attach(anscombe)

plot(x1, y1, main="Anscombe",xlab="x1 ", ylab="y1", pch=19)

plot(x2, y2, main="Anscombe",xlab="x2 ", ylab="y2", pch=19)

plot(x3, y3, main="Anscombe",xlab="x3 ", ylab="y3", pch=19)

plot(x4, y4, main="Anscombe",xlab="x4 ", ylab="y4", pch=19)

data1.lm<-lm(y1~x1,data=anscombe)

data2.lm<-lm(y2~x2,data=anscombe)

data3.lm<-lm(y3~x3,data=anscombe)

data4.lm<-lm(y4~x4,data=anscombe)

abline(data1.lm)

abline(data2.lm)

abline(data3.lm)

abline(data4.lm)

data1.res = resid(data1.lm)

data2.res = resid(data2.lm)

data3.res = resid(data3.lm)

data4.res = resid(data4.lm)

plot(anscombe$x1, data1.res, ylab="Residuals", xlab="x1 ",  main="residual plot for data1")   
abline(0, 0)

plot(anscombe$x2, data2.res, ylab="Residuals", xlab="x2 ",  main="residual plot for data2")   
abline(0, 0)

plot(anscombe$x3, data3.res, ylab="Residuals", xlab="x3 ",  main="residual plot for data3")   
abline(0, 0)

plot(anscombe$x4, data4.res, ylab="Residuals", xlab="x4 ",  main="residual plot for data4")   
abline(0, 0)

Huber <- read.delim("C:/Teaching@cofc/Math 550/Chapter 3/huber.txt", header = TRUE)

hub1<-lm(YBad~x, data=Huber)

attach(Huber)

plot(x, YBad, main="Huber bad Scatterplot",xlab="x ", ylab="Ybad", pch=19)

abline(hub1, col="red")

hub2<-lm(YGood~x, data=Huber)

plot(x, YGood, main="Huber good Scatterplot",xlab="x ", ylab="YGood")

abline(hub2, col="blue")

#leverage values

#review formula

Sxx<-sum((x-mean(x))^2)

Hii<-1/nrow(Huber)+(x-mean(x))^2/Sxx

#advanced function

H1<-hatvalues(hub1)

# what would be leverage for hub2?

# further investigation of leverage points

I(hatvalues(hub1)>2\*mean(H1))

# residuals and standard residuals

# review formula

Out<-summary(hub1)

Res<-YBad-(Out$coefficients[1,1]+ Out$coefficients[2,1]\*x)

Resa<-YBad-fitted(hub1)

RSS<-sum (Res^2)

S<-sqrt(RSS/(nrow(Huber)-2))

sigma(hub1)

Rst<-Res/(S\*sqrt(1-hatvalues(hub1)))

#advanced function in R

rstandard (hub1) # standard residuals

bonds <- read.delim("C:/Teaching@cofc/Math 550/Chapter 3/bonds.txt")

attach(bonds)

plot(CouponRate, BidPrice, main="US treasury Bonds Scatterplot",xlab="Coupon rate ", ylab="Bid Price")

Bondslm<-lm(BidPrice~CouponRate, data=bonds)

outhat<-hatvalues(Bondslm)

bonds.sres <- rstandard(Bondslm)

Leverage<-I(outhat>4/nrow(bonds))

Outlier<-I(abs(bonds.sres)>2)

BadL<- I(outhat>4/nrow(bonds) & abs(bonds.sres)>2)

#CD calculation in math

WCD1<-bonds[-c(1),]

WCD1lm<-lm(BidPrice~CouponRate, data=WCD1)

newbonds.CouponRate <- data.frame(CouponRate = bonds$CouponRate)

predictWCD1<-predict(WCD1lm, newdata = newbonds.CouponRate)

predictbonds<-predict(Bondslm, newdata = newbonds.CouponRate)

numD1<-sum((predictWCD1- predictbonds)^2)

outs<-summary(Bondslm)

S<-outs$sigma

D1<- numD1/(2\* S^2)

#equivalent formula for Cook’s distance

Res<-rstandard(Bondslm)

H<-hatvalues(Bondslm)

D2<-Res^2/2\*H/(1-H)

Forinv<-I(D2>4/(35-2))

Forsinv<-I(D2>1)

# advanced function

CD<-cooks.distance(Bondslm)

plot(bonds$CouponRate, CD, ylab="Cook’s distance", xlab="CouponRate ",  main="Cook’s distance for Bonds")   
abline(4/(nrow(bonds)-2), 0)

abline(1, 0)

plot(bonds$CouponRate, bonds.sres, ylab="Standard Residuals", xlab="Coupon Rate ",  main="residual plot for data1")

abline(0,0)

abline(2,0)

abline(-2,0)

plot(bonds$CouponRate, outhat, ylab="Leverage", xlab="Coupon Rate ",  main="residual plot for data1")

abline(4/nrow(bonds),0)

#advanced diagnostic plots in R

plot(Bondslm)

abline(v=4/nrow(bonds))

abline(-2,0)

abline(2,0)

# turns out some observations don’t belong

newdata <- bonds[-c(4, 13, 35), ]

attach(newdata)

plot(newdata$CouponRate, BidPrice, ylab="Bid Price", xlab="Coupon Rate ", main="Scatterplot for the new data")

newdatalm<-lm(BidPrice~CouponRate, data=newdata)

#for residuals vs fitted value; normal Q-Q plot, sqrt(standardized residuals) vs fitted values, standardized residuals vs leverage.

cleaning <- read.delim("C:/Teaching@cofc/Math 550/Chapter 3/cleaning.txt")

plot(Rooms~Crews, data=cleaning)

linearMod<-lm(Rooms~Crews, data=cleaning)

abline(linearMod, col="red")

new.Crews<- data.frame(Crews= c(4, 16))

predict(linearMod,newdata = new.Crews, interval = "prediction")

cleaning.sres<-rstandard(linearMod)

# motivate next topic: transformation

plot(cleaning$Crews, cleaning.sres, ylab="Standard Residuals", xlab="crewa ", main="residual plot for cleaning data")

abline(0,0)

abline(2,0)

abline(-2,0)

plot(linearMod)

sd<-aggregate(cleaning$Rooms, by=list(Crews = cleaning$Crews), FUN=sd)

sqrty<-sqrt(cleaning$Rooms)

sqrtx<-sqrt(cleaning$Crews)

linearmodtr<-lm(sqrty~sqrtx)

summary(linearmodtr)

plot(linearmodtr)

new.x<- data.frame(sqrtx= c(2, 4))

predictedn<-predict(linearmodtr,newdata = new.x, interval = "prediction")

predictedn^2

# details about transformation methods

responsetransformation <- read.delim("C:/Teaching@cofc/Math 550/Chapter 3/responsetransformation.txt")

attach(responsetransformation)

plot(y~x)

lnm<-lm(y~x, data= responsetransformation)

dy<-density(y)

plot(dy)

qqnorm(y, pch = 1, frame = FALSE)

qqline(y, col = "steelblue", lwd = 2)

boxplot(y,data=responsetransformation, main="generated data",ylab="y")

plot(density(x,bw=5))

qqnorm(x, pch = 1, frame = FALSE)

qqline(x, col = "steelblue", lwd = 2)

boxplot(x,data=responsetransformation, main="generated data",ylab="x")

predicty <- predict(lnm,list(x=x))

yno<-y^(-1)

ynh<-y^(-0.5)

ynt <-y^(-1/3)

yz<-log(y)

yt <-y^(1/3)

yh<-y^(0.5)

yo<-y

lamno<-lm(predicty ~yno, data=responsetransformation)

rssno<- sum(resid(lamno)^2)

lamnh<-lm(predicty ~ynh, data=responsetransformation)

rssnh<- sum(resid(lamnh)^2)

lamnt<-lm(predicty ~ynt, data=responsetransformation)

rssnt<- sum(resid(lamnt)^2)

lamz<-lm(predicty~yz, data=responsetransformation)

rssz<- sum(resid(lamz)^2)

lamo<-lm(predicty ~yo, data=responsetransformation)

rsso<- sum(resid(lamo)^2)

lamh<-lm(predicty ~yh, data=responsetransformation)

rssh<- sum(resid(lamh)^2)

lamt<-lm(predicty~yt, data=responsetransformation)

rsst<- sum(resid(lamt)^2)

min(rssno, rssnh,rssnt,rssz,rsso,rssh,rsst)

Advanced function for inverse response plot

library(alr4)

lam<-invResPlot(lnm, lambda=c(-1,-1/2,-1/3,0,1/3,1/2,1))

lam$lambda

lam$RSS

plot(lam$RSS~lam$lambda)

lam<-invResPlot(lnm, lambda=c(0,0.33,1))

#Box-Cox method review

#install.packages("psych")

library("psych")

lambda<-rep(0,100)

RSSnl<-rep(0,100)

for(i in 1:100)

{ lambda[i]<- -3+(i-1)\*6/99

# geometric.mean

ynl<-( geometric.mean(y))^(1-lambda[i])\*(y^lambda[i]-1)/lambda[i]

lmnl<-lm(ynl~x, data=responsetransformation)

RSSnl[i]<-sum(resid(lmnl)^2)

}

best.lambda<-lambda [which(RSSnl==min(RSSnl))]

min(RSSnl)

yn<- geometric.mean(y)\*log(y)

lmn<-lm(yn~x, data=responsetransformation)

RSSn<-sum(resid(lmn)^2)

RSSn

# advanced Box-Cox

library(faraway)

library(MASS)

#note by in seq (to-from)/(n-1)

lnm<-lm(y~x, data= responsetransformation)

bc<-boxcox(lnm, lambda=seq(-3,3, length.**out** = 100

))

Blambda<-bc$x[which(bc$y==max(bc$y))]

ny<-y^(1/3)

dny<-density(ny)

plot(dny)

lmt<-lm(ny~x, data= responsetransformation)

summary(lmt)

plot(lmt)

qqnorm(ny, pch = 1, frame = FALSE)

qqline(ny, col = "steelblue", lwd = 2)

boxplot(ny,data=responsetransformation, main="generated data",ylab="transformed")

plot(ny~x)

# for illustration of power transforming the predictor

responsetransformation <- read.delim("C:/Teaching@cofc/Math 550/Chapter 3/responsetransformation.txt")

attach(responsetransformation)

dy<-density(y)

plot(dy)

dx<-density(x)

plot(dx)

lnm<-lm(x~y, data= responsetransformation)

plot(lnm)

library("psych")

lambda<-rep(0,100)

RSSnl<-rep(0,100)

RSSnls<-rep(0,100)

RSSnlm<-rep(0,100)

# power transformation

for(i in 1:100)

{ lambda[i]<- -3+(i-1)\*6/99

ynls<- (y^lambda[i]-1)/lambda[i]

lmnls<-lm(x~ynls, data=responsetransformation)

RSSnls[i]<-sum(resid(lmnls)^2)

ynl<- y^lambda[i]

lmnl<-lm(x~ynl, data=responsetransformation)

RSSnl[i]<-sum(resid(lmnl)^2)

ynlm<- geometric.mean(y)^(1-lambda[i])\* (y^lambda[i]-1)/lambda[i]

lmnlm<-lm(x~ynlm, data=responsetransformation)

RSSnlm[i]<-sum(resid(lmnlm)^2)

}

sum((RSSnl-RSSnls)^2)

sum((RSSnl-RSSnlm)^2)

best.lambda<-lambda [which(RSSnl==min(RSSnl))]

best.lambdas<-lambda [which(RSSnls==min(RSSnls))]

best.lambdam<-lambda [which(RSSnlm==min(RSSnlm))]

best.lambda

best.lambdas

best.lambdam

#when transforming the predictor, three power transformations are equivalent

#when transforming the response, three power transformations are quite different. Only modified power transformation works because it is fair comparisons between RSS(λ) for

different λ values

library("psych")

lambda<-rep(0,100)

RSSnl<-rep(0,100)

RSSnls<-rep(0,100)

RSSnlm<-rep(0,100)

for(i in 1:100)

{ lambda[i]<- -3+(i-1)\*6/99

ynls<- (y^lambda[i]-1)/lambda[i]

lmnls<-lm(ynls~x, data=responsetransformation)

RSSnls[i]<-sum(resid(lmnls)^2)

ynl<- y^lambda[i]

lmnl<-lm(ynl~x, data=responsetransformation)

RSSnl[i]<-sum(resid(lmnl)^2)

ynlm<- geometric.mean(y)^(1-lambda[i])\* (y^lambda[i]-1)/lambda[i]

lmnlm<-lm(ynlm~x, data=responsetransformation)

RSSnlm[i]<-sum(resid(lmnlm)^2)

}

sum((RSSnl-RSSnls)^2)

sum((RSSnl-RSSnlm)^2)

best.lambda<-lambda [which(RSSnl==min(RSSnl))]

best.lambdas<-lambda [which(RSSnls==min(RSSnls))]

best.lambdam<-lambda [which(RSSnlm==min(RSSnlm))]

best.lambda

best.lambdas

best.lambdam

# advanced power transformation of the predictor

library(car)

pc<-powerTransform(y~1)

summary(pc)

salarygov <- read.csv("C:/Teaching@cofc/Math 550/Chapter 3/salarygov.txt", sep="")

attach(salarygov)

plot(salarygov$MaxSalary~salarygov$Score)

MS<-density(salarygov$MaxSalary)

plot(MS)

qqnorm(salarygov$MaxSalary, pch = 1, frame = FALSE)

qqline(salarygov$MaxSalary, col = "steelblue", lwd = 2)

boxplot(salarygov$MaxSalary,data=salarygov, main="Salary data",ylab="Max salary")

S<-density(salarygov$Score)

plot(S)

qqnorm(salarygov$Score, pch = 1, frame = FALSE)

qqline(salarygov$Score, col = "steelblue", lwd = 2)

boxplot(salarygov$Score,data=salarygov, main="Salary data",ylab="Score")

# apply Box-Cox method on both response and predictor

library(car)

pc<-powerTransform(cbind(salarygov$MaxSalary,salarygov$Score)~1)

summary(pc)

tx<-sqrt(salarygov$Score)

ty<-log(salarygov$MaxSalary,base=exp(1))

mty<-density(ty)

plot(mty)

qqnorm(ty, pch = 1, frame = FALSE)

qqline(ty, col = "steelblue", lwd = 2)

boxplot(ty,data=salarygov, main=" Salary data",ylab="transformed Max salary")

tS<-density(tx)

plot(tS)

qqnorm(tx, pch = 1, frame = FALSE)

qqline(tx, col = "steelblue", lwd = 2)

boxplot(tx,data=salarygov, main="Salary data",ylab="transformed Score")

plot(ty~tx)

abline(tlm)

tlm<-lm(ty~tx)

summary(tlm)

plot(tlm)

attach(salarygov)

#Box-Cox method on predictor then Inverse Response method on response

pscore<-powerTransform(Score~1)

summary(pscore)

transy<-lm(MaxSalary~tx)

lam<-invResPlot(transy, lambda=c(-1,-1/2,-1/3,-1/4,1/4,1/3,1/2,1))

lam$lambda

plot(lam$RSS~lam$lambda)

pty<- MaxSalary^(-0.19)

ptyx<-lm(pty~tx)

plot(pty~tx)

abline(ptyx)

plot(ptyx)

#round to -0.25

npty<- MaxSalary^(-0.25)

nptyx<-lm(npty~tx)

plot(npty~tx)

abline(nptyx)

plot(nptyx)

# based on the diagnostic plots, box-cox methods on both response and predictor are preferred

airfares <- read.delim("C:/Teaching@cofc/Math 550/Chapter 3/airfares.txt")

|  |
| --- |
|  |
| |  | | --- | |  | |