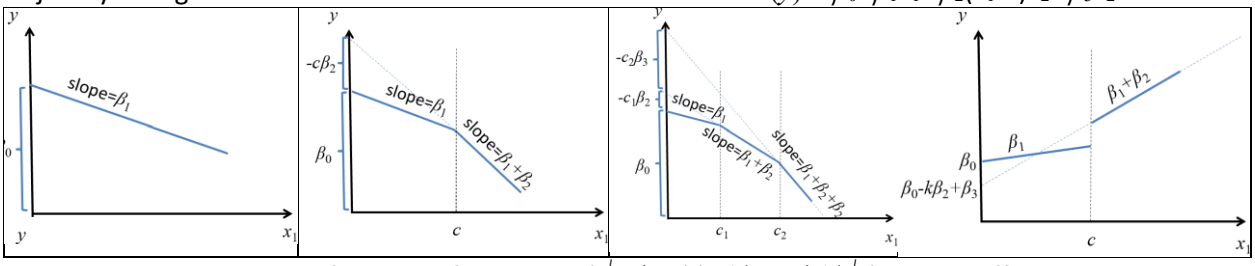
MATH 2780 Chapter 9B Worksheet

Summary

Sometimes a continuous **single line model** to data $E(y) = \beta_0+\beta_1x_1$ is not correct and a **continuous two-line model** $E(y) = \beta_0+\beta_1x_1+\beta_2(x_1-c)x_2$ where $x_2=1$ if $x_1>c$, $x_2=0$ if $x_1\le c$, and c is called a knot. **Continuous three line model** (or more) $E(y) = \beta_0+\beta_1x_1+\beta_2(x_1-c_1)x_2+\beta_3(x_1-c_2)x_2$ are also possible where $x_2=1$ if $x_1>c_1$, $x_2=0$ if $x_1\le c_1$, $x_3=1$ if $x_1>c_2$, $x_2=0$ if $x_1\le c_2$, and c_1, c_2 are called a knots. Occasionally the process may disjointly change and we have a discontinuous **two-line model** $E(y) = \beta_0+\beta_1x_1+\beta_2(x_1-c)x_2+\beta_3x_2$.



Weighted Least Squares: Often, transformations ($\sqrt{y}$, $\log(y)$, $1/y$ and $1/\sqrt{y}$) are not effective in stabilizing the variance. $WSSE = \sum_{i=1}^{n} w_i(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} w_i(y - \hat{\beta}_0 - \hat{\beta}_1x_1 - \hat{\beta}_2x_2 - \cdots - \hat{\beta}_kx_k)^2$, $r_i^* = \sqrt{w_i}(y_i - \hat{y}_i)$

1. Divide the data into several approximately equal groups according to the independent variable, x.
   a. If the data is replicated and balanced, then create one group for each value of x.
   b. If the data is not replicated, group the data according into ranges of x.
2. Determine the sample mean $\bar{x}$ and variance s^2 of the residuals in each group.
3. For each group, compare the residual variance s^2 to different functions of $\bar{x}$ by calculating $s^2/f(\bar{x})$.
4. Find the function of $\bar{x}$ for which the ratio is nearly constant across groups.
5. The appropriate weights for the groups are $1/f(\bar{x})$. Generally $w_i = 1/\sigma_i^2$, or $w_i = 1/\bar{x}_j$, or $w_i = 1/\bar{x}_i^2$.

Coefficient and Residual Variance Estimation: The ordinary least squares regression coefficients.

$$Y = X\beta + E$$
$$\hat{\beta} = (X'X)^{-1}X'y$$
$$s^2 = \frac{(y - X\hat{\beta})'(y - X\hat{\beta})}{n-k-1}$$
$$MSE = s^2,\ s = \sqrt{s^2}$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix}, X = \begin{bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{kn} \\ 1 & x_{12} & x_{22} & \cdots & x_{k2} \\ 1 & x_{13} & x_{23} & \cdots & x_{k3} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{1n} & x_{2n} & \cdots & x_{kn} \end{bmatrix}, \beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_k \end{bmatrix}, E = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Regression Residuals: Residuals are $\hat{\varepsilon}_i = y_i - \hat{\beta}_0 - \hat{\beta}_1x_1 - \ldots - \hat{\beta}_kx_k$, $s^2 = \sum(y_i - \hat{y}_i)^2/(n-k-1)$.

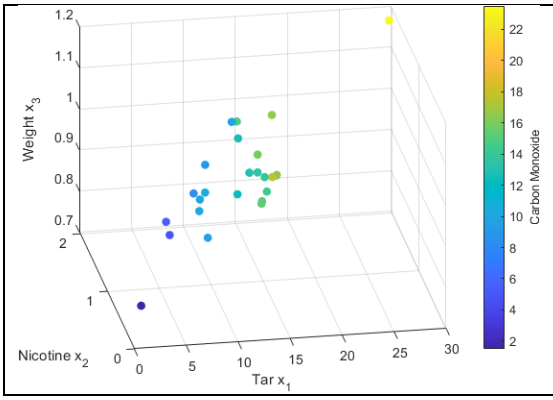
All of the same methods we learned to work with residuals continue to apply.

Ridge and LASSO Regression: When we examine the assumptions of our model, particularly multicollinearity, we can calculate the correlation between regressors and VIFs. Multicollinearity affects our solution to the system of equation and difficulty inverting the matrix $(X'X)$. Instead of removing a variable, we may want to keep all variables and use Ridge Regression. Ridge Regression in essence adds a small amount c to the diagonals of $(X'X)$ and inverts $(X'X+cI)$. And estimates coefficients as $\hat{\beta}_R = (X'X + cI_{k+1})^{-1}X'y$. Or in essence using the objective function $Q = \sum(y_i - \hat{y}_i)^2 + c\sum(\beta_j)^2$ which is OLS when c=0. LASSO is similar and has objective function $Q = \sum(y_i - \hat{y}_i)^2 + c\sum|\beta_j|$, but must be solved numerically. Both Ridge and LASSO regression result in coefficients biased toward 0. Both Ridge and LASSO regression have origins from Bayesian statistics.

Logistic Regression: This dependency of a probability p(x) on an independent variable x through the logistic mapping function $p(x) = 1/(1+e^{-\beta_0-\beta_1x})$ which can be linearized $ln(\hat{p}/(1-\hat{p})) = \hat{\beta}_0 + \hat{\beta}_1x_1 + \ldots + \hat{\beta}_px_p$, but don't use linear regression on this, need to use $LL = \sum_{i=1}^{n} y_i(\beta_0 + \beta_1x_i) - \sum_{i=1}^{n} ln(1+e^{\beta_0+\beta_1x_i})$. Can calculate odds $\hat{o}_i = \hat{p}_i/(1-\hat{p}_i) = e^{\hat{\beta}_0+\hat{\beta}_1x_i}$ and odds ratio $\hat{OR} = e^{\hat{\beta}_0+\hat{\beta}_1x_b} / e^{\hat{\beta}_0+\hat{\beta}_1x_a} = e^{\hat{\beta}_1\Delta}$, $\Delta = x_b - x_a$.

MATH 2780 Chapter 9B Worksheet

Example: $E(y)=\beta_0+\beta_1x_1+\beta_2x_2+\beta_3x_3$. $\hat{\beta}_R=(X'X+cI_{k+1})^{-1}X'y$ and $Q=\sum(y_i-\hat{y}_i)^2+c\sum(\beta_j)^2$.



y= carbon monoxide content, $x1$= tar content, $x2$= nicotine content, $x3$= weight

	TAR	NICOTINE	WEIGHT	CO
1	14.1	0.86	0.9853	13.6
2	16.0	1.06	1.0938	16.6
3	29.8	2.03	1.1650	23.5
4	8.0	0.67	0.9280	10.2
5	4.1	0.40	0.9462	5.4

Run R code and examine results.

```
# FTCCIGAR #
# read data
# parse out variables
# scatter plot with line
# scatter plot
# x1-x3 fit
# Ridge Regression
# make Identity matrix
# ordinary least squares
# manual ridge regression
# Ridge Regression with R function
# Setting the range of lambda values
#fit ridge regression model
#view summary of model
#perform k-fold cross-validation to find optimal lambda value
#find optimal lambda value that minimizes test MSE
#produce plot of test MSE by lambda value
#find coefficients of best model
```

TAR	NICOTINE	WEIGHT	CO
14.1	0.86	0.9853	13.6
16	1.06	1.0938	16.6
29.8	2.03	1.165	23.5
8	0.67	0.928	10.2
4.1	0.4	0.9462	5.4
15	1.04	0.8885	15
8.8	0.76	1.0267	9
12.4	0.95	0.9225	12.3
16.6	1.12	0.9372	16.3
14.9	1.02	0.8858	15.4
13.7	1.01	0.9643	13
15.1	0.9	0.9316	14.4
7.8	0.57	0.9705	10
11.4	0.78	1.124	10.2
9	0.74	0.8517	9.5
1	0.13	0.7851	1.5
17	1.26	0.9186	18.5
12.8	1.08	1.0395	12.6
15.8	0.96	0.9573	17.5
4.5	0.42	0.9106	4.9
14.5	1.01	1.007	15.9
7.3	0.61	0.9806	8.5
8.6	0.69	0.9693	10.6
15.2	1.02	0.9496	13.9
12	0.82	1.1184	14.9

```
# FTCCIGAR #
#install.packages("car")
library(car)
#install.packages("glmnet")
```

```
# x1-x3 fit
lmx1to3<- lm(y~x1+x2+x3,data=df)
temp<-anova(lmx1to3)
out <- temp
```

MATH 2780 Chapter 9B Worksheet

```
library(glmnet)
#install.packages("Matrix")
library(Matrix)

# read data
mydata <- read.delim("ftccigar.txt",header=TRUE,
sep="",dec=".")
head(mydata)

# parse out variables
n <- nrow(mydata)
k <- ncol(mydata)-1
x1 <- c(mydata[, 1]) #x1 tar content
x2 <- c(mydata[, 2]) #x2 nicotine content
x3 <- c(mydata[, 3]) #x3 weight
y  <- c(mydata[, 4]) #y  carbon monoxide

df       <- data.frame(cbind(x1,x2,x3))
names(df) <- c("x1","x2","x3")
head(df)

# scatter plot with line
plot(x1,y,xlab='Tar Content',     ylab='Carbon Monoxide',
    pch=19,col="blue")
abline(lm(y~x1),col='red',lty=2)
plot(x2,y,xlab='Nicotine Content',ylab='Carbon
Monoxide', pch=19,col="blue")
abline(lm(y~x2),col='red',lty=2)
plot(x3,y,xlab='Weight',         ylab='Carbon Monoxide',
    pch=19,col="blue")
abline(lm(y~x3),col='red',lty=2)

# scatter plot
library("plot3D")
scatter3D(x1,x2,y,pch=19,cex=1,colvar=NULL,
        col="red",theta=20, phi=10,bty="b",
        xlab="Tar",ylab="Nicotine",zlab="Carbon
Monoxide",  main = "Cigarettes")
scatter3D(x1,x3,y,pch=19,cex=1,colvar=NULL,
        col="red",theta=20,phi=10,bty="b",
        xlab="Tar",ylab="Weight",  zlab="Carbon
Monoxide",  main = "Cigarettes")
scatter3D(x2,x3,y,pch=19,cex=1,colvar=NULL,
        col="red",theta=20,phi=10,bty="b",xlab=
"Nicotine", ylab = "Weight", zlab = "Carbon Monoxide",
        main = "Cigarettes")
```

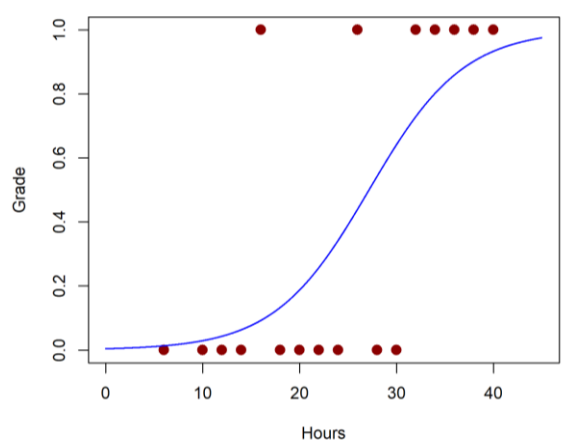
```
m <- nrow(temp)
out$Df <- with(temp,c(sum(Df[1:(m-
1)]),Df[m],rep(NA_real_,m-2)))
out$`Sum Sq` <- with(temp,c(sum(`Sum Sq`[1:(m-1)]),
                    `Sum Sq`[m],rep(NA_real_,m-2)))
out$`Mean Sq` <- with(out,out$`Sum Sq`/out$Df)
out$`F value` <- c(out$`Mean Sq`[1]/out$`Mean
Sq`[2],rep(NA_real_,m-1))
out$`Pr(>F)` <- c(pf(out$`F value`[1],out$Df[1],out$Df[2],
                lower.tail = FALSE),rep(NA_real_,m-1))
out <- out[1:2,]
rownames(out) <- c("Model","Residuals")
out

summary(lmx1to3)
sigma(lmx1to3)

# Ridge Regression
X<-cbind(rep(1,n),x1,x2,x3)
c<-0.1
# make Identity matrix
eye<-`diag<-`(matrix(0,k+1,k+1),1)
delt<-c(rep(0,k+1))
#delt<-c(1.5,1.15,-2.5,-1.2)

bLS   <-solve(t(X)%*%X)%*%t(X)%*%y
yhatXLS<-X%*%bLS
bRidge<-
solve(t(X)%*%X+c*eye)%*%(c*delt+t(X)%*%X%*%bLS)
yhatXRidge<-X%*%bRidge
cbind(bLS,bRidge)

eLS <- y-X%*%bLS
hist(eLS)
mean(eLS)          # 0
s2LS  <- t(eLS)%*%eLS/(n-k-1) #
sLS   <- sqrt(s2LS)    #
sLS

eRidge <- y-X%*%bRidge
hist(eRidge)
mean(eRidge)          # not 0
s2Ridge  <- t(eRidge)%*%eRidge/(n-k-1) #
sRidge   <- sqrt(s2Ridge)    #
sRidge
# compute the coefficients of determination
SSyy <- sum(y^2)-(sum(y))^2/n
R2LS<-1-s2LS/SSyy
R2Ridge<-1-s2Ridge/SSyy
```

MATH 2780 Chapter 9B Worksheet

```
# Ridge Regression with R function

# Setting the range of lambda values
lambda_seq <- seq(2,0,by=-.1)

#fit ridge regression model
model <- glmnet(X,y,alpha=0,lambda=lambda_seq)
#view summary of model
summary(model)

#perform k-fold cross-validation to find optimal lambda value
cv_model <- cv.glmnet(X,y,alpha=0,grouped=FALSE,lambda=lambda_seq)

#find optimal lambda value that minimizes test MSE
#best_lambda <- cv_model$lambda.min
nlams<-length(cv_model$lambda)
num<-c/.1# c a multiple of 0.1
best_lambda<-cv_model$lambda[nlams-num]
#best_lambda<-0
best_lambda

#produce plot of test MSE by lambda value
plot(cv_model)

#find coefficients of best model
best_model<-glmnet(X,y,alpha=0,grouped=FALSE,lambda=best_lambda)
coef(best_model)
a0<-best_model$a0
b1<-best_model$beta[2,1]
b2<-best_model$beta[3,1]
b3<-best_model$beta[4,1]
bR<-c(a0,b1,b2,b3)

yhatXR<-X%*%bR

eR <- y-X%*%bR
hist(eR)
mean(eR)          # not 0
s2R  <- t(eR)%*%eR/(n-k-1) #
sR   <- sqrt(s2R)
sR

c(sLS,sRidge,sR)
# compute the coefficients of determination
SSyy <- sum(y^2)-(sum(y))^2/n
R2R=1-s2R/SSyy

betas<-data.frame(cbind(bLS,bRidge,bR))
names(betas) <- c("bLS","bRidge","bR")
betas
```

MATH 2780 Chapter 9B Worksheet

Example: Grade logistic regression model $p(x) = 1/(1+e^{-\beta_0-\beta_1 x})$.



$$p(x) = \frac{1}{1+e^{5.66-0.21x}}$$

Hours (x)	A (y)
6	0
8	0
10	0
12	0
14	0
16	1
18	0
20	0
22	0
24	0
26	1
28	0
30	0
32	1
34	1
36	1
38	1
40	1

Run R code and examine results.

```
# grade data
xx <- c(6, 8,10,12,14,16,18,20,22,24,26,28,30,32,34,36,38,40)
yy <- c(0, 0, 0, 0, 0, 1, 0, 0, 0, 0 ,1, 0, 0, 1, 1, 1, 1, 1)

#scatter plot
plot(x = xx,y = yy,xlab = "Hours",ylab = "Grade",  xlim = c(0,45),ylim = c(0,1),col = "darkred",
    cex = 1.5, main = "Hours vs. Grade", pch = 16)

logistic_model <- glm(yy~xx, family=binomial(link="logit"))
summary(logistic_model)
b0 <- logistic_model$coefficients[1]
b1 <- logistic_model$coefficients[2]

phat <- round(1/(1+exp(-b0-b1*xx)), digits = 4)
O    <- round(phat/(1-phat)       , digits = 4)
df <- data.frame(xx,yy,phat,O)
df

xhat <- (1:4500)/100
yhat <- 1/(1+exp(-b0-b1*xhat))
#scatter plot with curve
plot(x = xx,y = yy,xlab = "Hours",ylab = "Grade",  xlim = c(0,45),ylim = c(0,1),col = "darkred",
    cex = 1.5, main = "Hours vs. Grade", pch = 16)
points(xhat,yhat,cex = .1,col = "blue")
```