Stat 428 HW 4 vvgupta2

Question 1

```
library(bootstrap)
n <- nrow(law)
results <- replicate(n, expr= {
  theta_hat <- cor(law$LSAT,law$GPA)
  theta_hat_i <- sapply(1:n, function(i) cor(law$LSAT[-i], law$GPA[-i]))
  bias_jack <- (n-1)*(mean(theta_hat_i) - theta_hat)
  se_jack <- sqrt((n-1)*mean((theta_hat_i - mean(theta_hat_i))^2))
  return(c(theta_hat, bias_jack, se_jack))
})
print(se_jack <- mean(results[3,]))
```

```
## [1] 0.1425186
```

The Jackknife standard error estimate for the Pearson correlation between LSAT and GPA is 0.1425186

Question 2

```
library(bootstrap)
n = nrow (scor)
e = eigen(cov(scor), only = TRUE)$val
R = e[1]/sum(e)
B = 10000
R_1 = numeric(B)
for (b in 1:B) {
  thta_scor = sample (scor, size = n, replace = TRUE)
  eigens = eigen ( cov(thta_scor), only = TRUE)$val
  R_1[b] = eigens[1]/sum(eigens)
}
mean(R_1-R)
```

```
## [1] 0.00611486
```

```
sd(R_1)
```

```
## [1] 0.007760738
```

```
sort(R_1)[c(B*0.025, B*0.975)]
```

```
## [1] 0.6152146 0.6455206
```

Question 3

```
n = nrow(scor)
theta_hat = eigen(cov(scor), only = TRUE)$val[1]/sum(eigen(cov(scor), only = TRUE)$val)
theta_hat_i = sapply (1:n, function (i) (eigen(cov(scor[-i,]), only = TRUE)$val[1] / sum(eigen(cov(scor
bias_jack = (n-1)*(mean(theta_hat_i)-theta_hat)
mean(bias_jack)
```

```
## [1] 0.001069139
```

```
se_jack = sqrt((n-1)*mean((theta_hat_i-mean(theta_hat))^2))
mean(se_jack)
```

```
## [1] 0.04955244
```

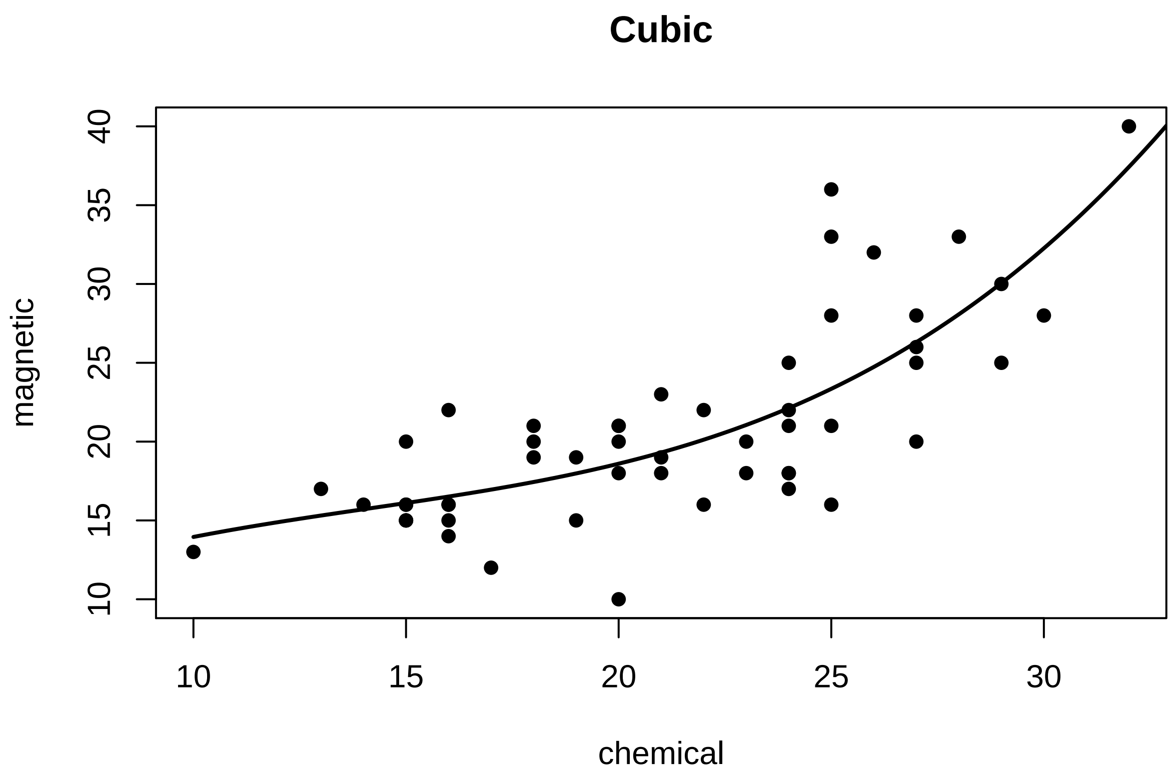
Jacknife Estimate bias: 0.001069139 Jacknife Estimate stderr: 0.049552

Question 4

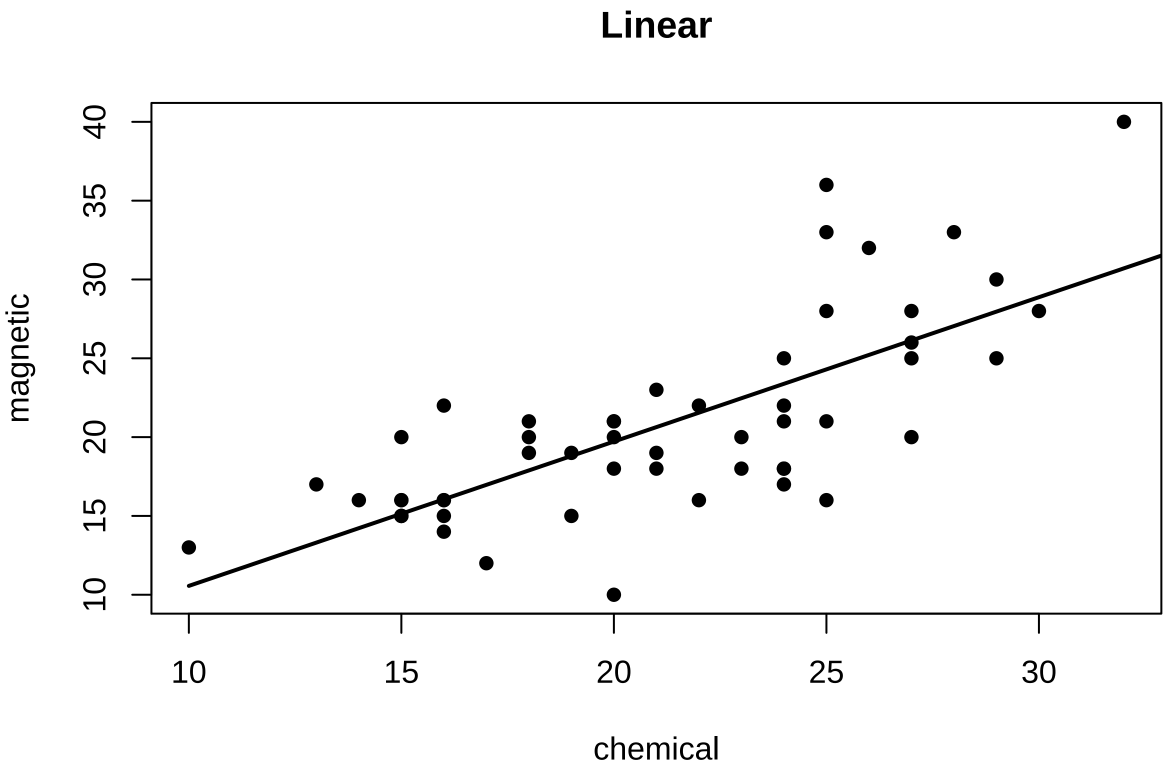
```
#install.packages('DAAG')
library(DAAG)
```

```
## Loading required package: lattice
```

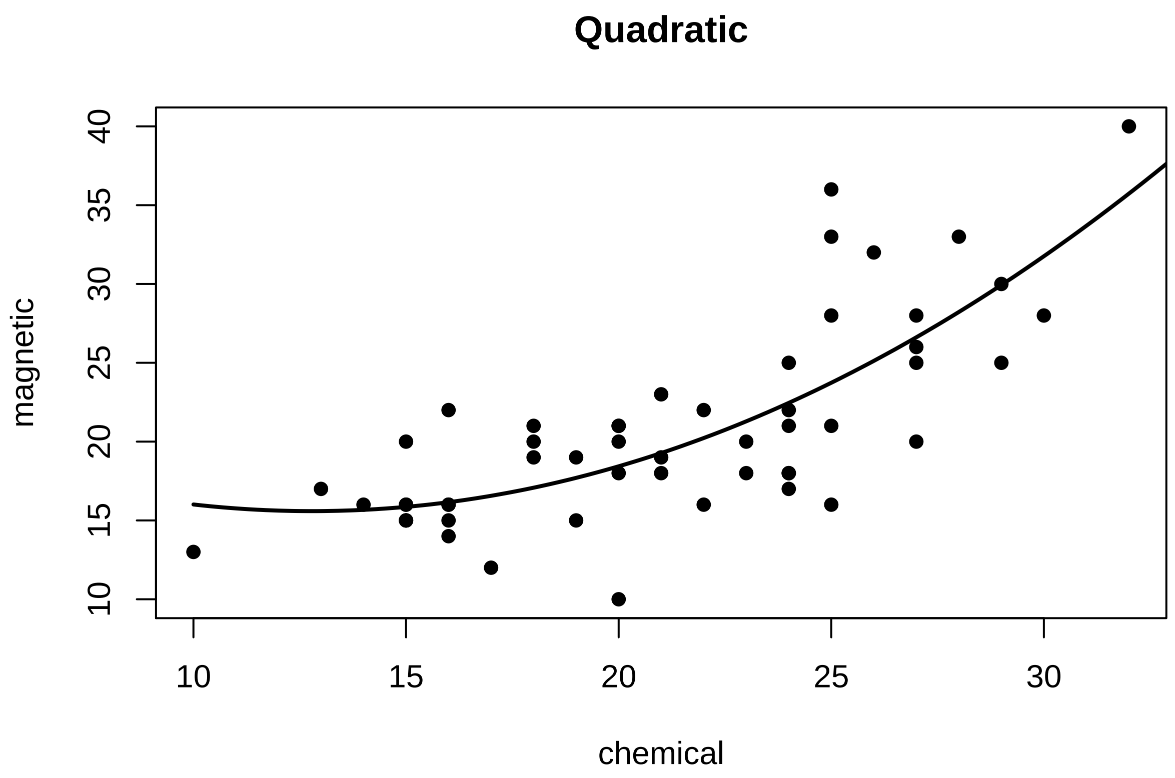
```
attach(ironslag)
L5 <- lm(magnetic ~ chemical + I(chemical^2)+ I(chemical^3))
a <- seq(10, 40, .1)
plot(chemical, magnetic, main="Cubic", pch=16)
hat5 <- L5$coef[1] + L5$coef[2] * a + L5$coef[3] * a^2 + L5$coef[4] * a^3
lines(a, hat5, lwd=2)
```



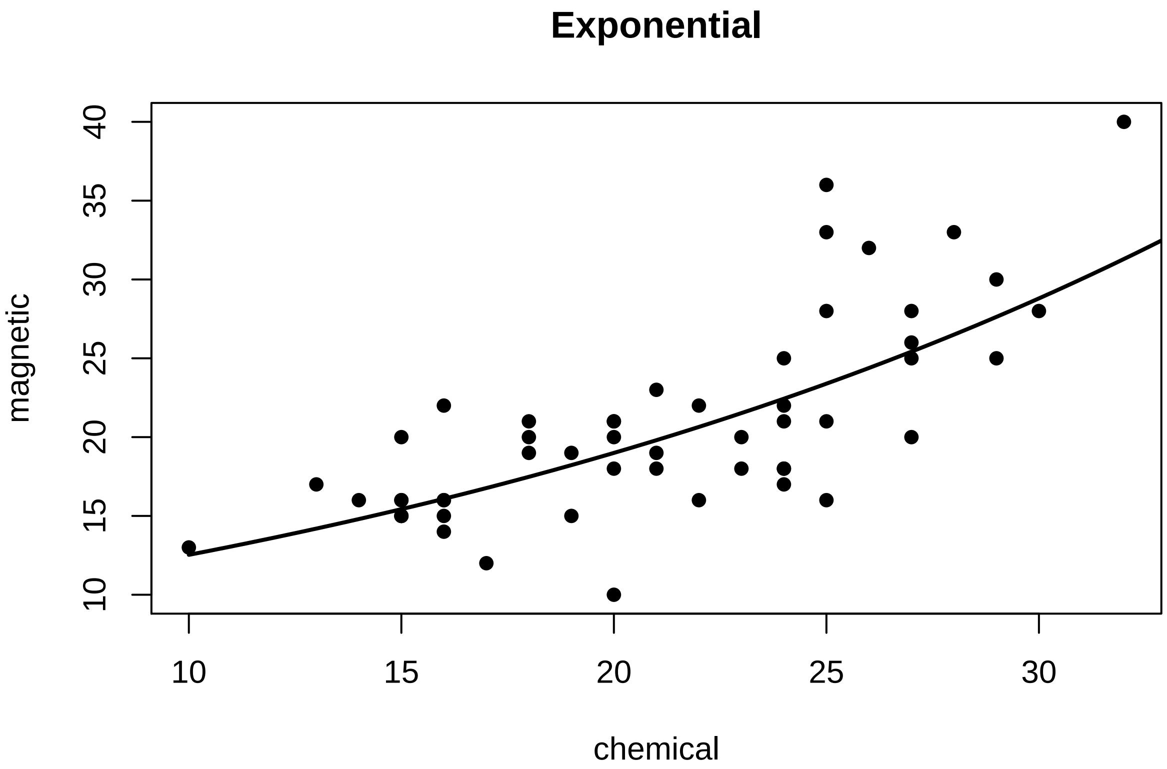
```
L1 <- lm(magnetic ~ chemical)
plot(chemical, magnetic, main = "Linear", pch = 16)
yhat1 <- L1$coef[1] + L1$coef[2] * a
lines(a, yhat1, lwd=2)
```



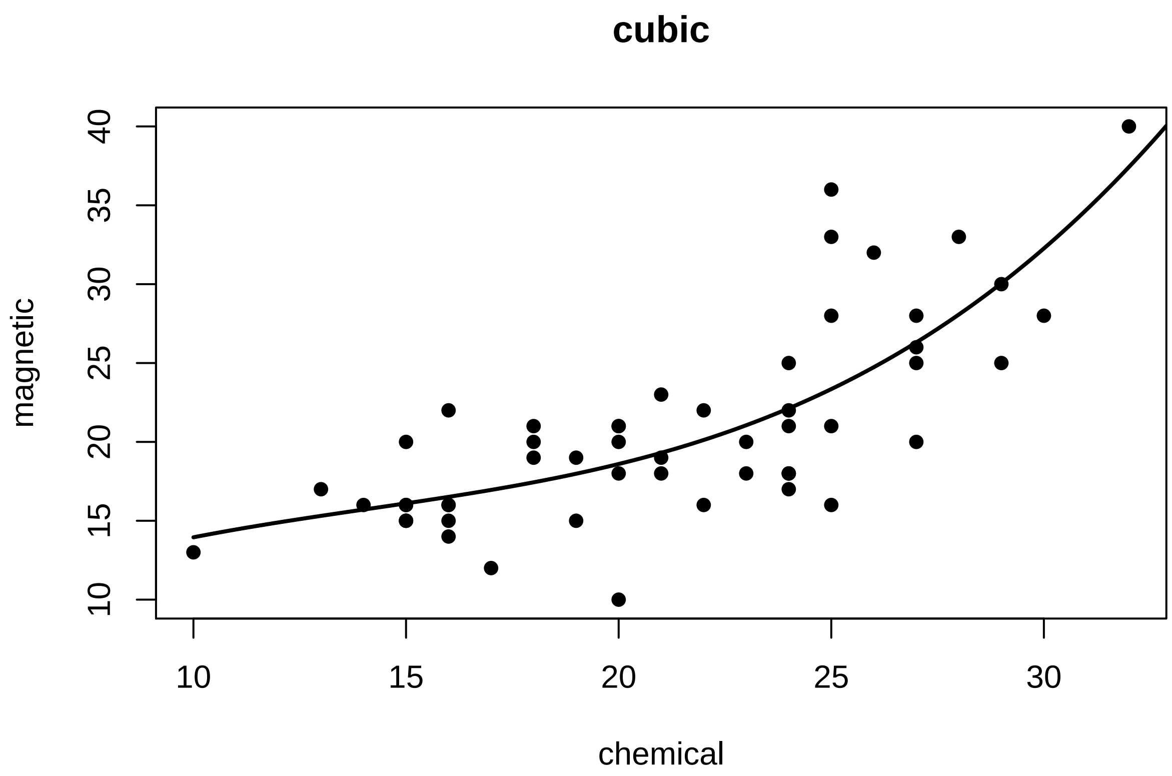
```
L2 <- lm(magnetic ~ chemical + I(chemical^2))
plot(chemical, magnetic, main = "Quadratic", pch = 16)
yhat2 <- L2$coef[1] + L2$coef[2] * a + L2$coef[3] * a^2
lines(a, yhat2, lwd=2)
```



```
L3 <- lm(log(magnetic) ~ chemical)
plot(chemical, magnetic, main = "Exponential", pch = 16)
logyhat3 <- L3$coef[1] + L3$coef[2] * a
yhat3 <- exp(logyhat3)
lines(a, yhat3, lwd=2)
```



```
L4 <- lm(magnetic ~ chemical + I(chemical^2) + I(chemical^3))
plot(chemical, magnetic, main = "cubic", pch = 16)
yhat4 <- L4$coef[1] + L4$coef[2] * a + L4$coef[3] * a^2 + L4$coef[4] * a^3
lines(a, yhat4, lwd=2)
```



```
n <- length(magnetic) #in DAAG ironslag
e1 <- e2 <- e3 <- e4 <- numeric(n)
# for n-fold cross validation
# fit models on leave-one-out samples
for (k in 1:n) {
  y <- magnetic[-k]
  x <- chemical[-k]

  J1 <- lm(y ~ x)
  yhat1 <- J1$coef[1] + J1$coef[2] * chemical[k]
  e1[k] <- magnetic[k] - yhat1

  J2 <- lm(y ~ x + I(x^2))
  yhat2 <- J2$coef[1] + J2$coef[2] * chemical[k] + J2$coef[3] * chemical[k]^2
  e2[k] <- magnetic[k] - yhat2

  J3 <- lm(log(y) ~ x)
  logyhat3 <- J3$coef[1] + J3$coef[2] * chemical[k]
  yhat3 <- exp(logyhat3)
  e3[k] <- magnetic[k] - yhat3

  J4 <- lm(y ~ x + I(x^2) + I(x^3))
  yhat4 <- J4$coef[1] + J4$coef[2] * chemical[k] + J4$coef[3] * chemical[k]^2 + J4$coef[4] * chemical[k
  e4[k] <- magnetic[k] - yhat4
}
c(mean(e1^2), mean(e2^2), mean(e3^2), mean(e4^2))
```

```
## [1] 19.55644 17.85248 18.44188 18.17756
```

```
n <- length(magnetic)
y <- magnetic
x <- chemical
J1 <- lm(y ~ x)
summary(J1)$adj.r.squared
```

```
## [1] 0.5281545
```

```
J2 <- lm(y ~ x + I(x^2))
summary(J2)$adj.r.squared
```

```
## [1] 0.5768151
```

```
J3 <- lm(log(y) ~ x)
summary(J3)$adj.r.squared
```

```
## [1] 0.5280556
```

```
J4 <- lm(y ~ x + I(x^2) + I(x^3))
summary(J4)$adj.r.squared
```

```
## [1] 0.5740396
```

The quadratic model resulted in the lowest LOOCV error and the highest r squared value and therefore we select that model in both cases.

Question 5

```
cvm1 = function (x,y){
    Fx1 = ecdf(x)
    Fx2 = ecdf(y)
    n = length(x)
    m = length(y)
    w1 = sum((Fx1(x)-Fx2(x))^2)+sum((Fx1(y)-Fx2(y))^2)
    w2 = w1*m*n/((m+n)^2)
    return(w2)
}
attach(chickwts)
x = sort(as.vector(weight[feed == "casein"]))
y = sort(as.vector(weight[feed == "horsebean"]))
ans = cvm1(x, y)
ans
```

```
## [1] 1.594697
```

```
b = 999
z = c(x,y)
n_1 = 1:length(z)
thetahat_js <- replicate(b, expr={
a1=sample(n_1, size=length(x), replace=FALSE)
x_1=z[a1]
y_1=z[-a1]
cvm1(x_1,y_1)
})
mean(abs(c(ans, thetahat_js))>=abs(ans))
```

```
## [1] 0.001
```

We observe the Cramer-von Mises Test Statistic to be = 1.594697 We also observe that p = 0.001