```
# Part C

source("scripts/partitions.R")

# c.i.

D <- bondEnergy$disEnergy
disDiox <- D[5] + D[6]
disHyd <- 436
disWat <- D[8] + D[9]
disMon <- D[5]

deltaDRxn2 <- disMon + disWat - disDiox - disHyd

# c.ii.

N <- 6.022e23
mDiox <- 44.1 / N / 1000 # g/mol to kg
mHyd <- 1.008 / N / 1000 # g/mol to kg
mWat <- 18.015 / N / 1000 # g/mol to kg
mMon <- 28.01 / N / 1000 # g/mol to kg

vibDiox <- data$vibration[1] # K
vibHyd <- data$vibration[2] # K
vibWat <- data$vibration[4] # K
vibMon <-  data$vibration[5] # K

iDiox <- data$inertia[1] * 1e-7 # g*cm^2 to kg*m^2
iHyd <- data$inertia[2] * 1e-7 # g*cm^2 to kg*m^2
iWat <- data$inertia[4] * 1e-21 # g^3*cm^6 to kg^3*m^6
iMon <- data$inertia[5] * 1e-7 # g*cm^2 to kg*m^2

sigmaDiox <- data$sigma[1]
sigmaHyd <- data$sigma[2]
sigmaWat <- data$sigma[4]
sigmaMon <- data$sigma[5]

# q translational

qTransRxn2 <- function(temp) {
  (qTrans(temp, mMon) * qTrans(temp, mWat)) /
    (qTrans(temp, mDiox) * qTrans(temp, mHyd))
}

# q vibrational

qVibRxn2 <- function(temp) {
  (qVib(temp, vibMon) * qVib(temp, vibWat)) /
    (qVib(temp, vibDiox) * qVib(temp, vibHyd))
}

#q nonlinear rotation

qRotRxn2 <- function(temp) {
```

```
  (qRotlin(temp, iMon, sigmaMon) * qRotNonlin(temp, iWat, sigmaWat)) /
    (qRotlin(temp, iDiox, sigmaDiox) * qRotlin(temp, iHyd, sigmaHyd))
}

# q electronic
# theta_electronic is large and T is relativly small so qElec = g0
# g0 = 1 for all the molecules we are dealing with

qEleRxn2 <- 1

# Kp for Reaction 1

R <- 0.00814 # kJ/mol
k2 <- (8.206e-5) / (6.022e23) # R/N Units are m^3 atm / K
KpRxn2 <- function(temp) {
  (k2 * temp)^0 * qTransRxn2(temp) * qVibRxn2(temp) * qRotRxn2(temp) *
    qEleRxn2 * exp(deltaDRxn2 / (R * temp))
}

# c.iii.

tempC <- seq(from = 50, to = 1500, by = 1)
tempK <- tempC + 273
inverseTempK <- 1 / tempK

KpReaction2 <- sapply(tempK, KpRxn2)
log10KpRxn2 <- log10(KpReaction2)
lnKpRxn2 <- log(KpReaction2)

ReactionTwo <- data.frame(
  "temp" = tempC,
  "inverseTemp" = inverseTempK,
  "Kp" = KpReaction2,
  "log10Kp" = log10KpRxn2,
  "lnKp" = lnKpRxn2
)

KpPlotRxn2 <- ggplot(data = ReactionTwo,
                     mapping = aes(
                       x = temp,
                       y = log10Kp)) +
  geom_line() +
  labs(title = "Figure 3",
       subtitle = "Reaction Two Equilibrium Constant") +
  xlab("Temperature (Celcius)") +
  ylab("Kp (atm^-2) Log10 Scale")

ggsave("pics/Figure3.pdf")

# c.iv.

HoffPlotRxn2 <- ggplot(data = ReactionTwo,
                   mapping = aes(
                     x = inverseTemp,
```

```
                    y = lnKp)) +
  geom_line() +
  labs(title = "Figure 4",
       subtitle = "Reaction Two van't Hoff Plot") +
  xlab("Temperature (Celcius^-1)") +
  ylab("Kp (atm^-2) Natural Log Scale")

ggsave("pics/Figure4.pdf")

deltaHRxn2 <- -R *(ReactionTwo$lnKp[300] - ReactionTwo$lnKp[1100]) /
  (ReactionTwo$inverseTemp[300] - ReactionTwo$inverseTemp[1100])

# c.v.

deltaHDiox <- data$heatFormation[1]
deltaHHyd <- data$heatFormation[2]
deltaHWat <- data$heatFormation[4]
deltaHMon <- data$heatFormation[5]

deltaHPrimeRxn2 <- deltaHMon + deltaHWat - deltaHDiox - deltaHHyd

# c.vi.

# deltaH

deltaH2Rxn2 <- 1:length(lnKpRxn2)

for (i in 1:length(lnKpRxn2)) {
  deltaH2Rxn2[i] <- -R * ((lnKpRxn2[i+1] - lnKpRxn2[i]) / (inverseTempK[i+1] -
inverseTempK[i]))
}

ReactionTwo <- mutate(ReactionTwo, "deltaH" = deltaH2Rxn2)

hPlotRxn2 <- ggplot(data = ReactionTwo,
                    mapping = aes(
                      x = temp,
                      y = deltaH)) +
  geom_line() +
  labs(title = "Figure 5",
       subtitle = "Delta H for Reaction Two") +
  xlab("Temperature (Celcius)") +
  ylab("Delta H (kJ/mol)")

ggsave("pics/Figure5.pdf")

# deltaU

deltaURxn2 <- -R * tempK * lnKpRxn2

ReactionTwo <- mutate(ReactionTwo, "deltaU" = deltaURxn2)

uPlotRxn2 <- ggplot(data = ReactionTwo,
                    mapping = aes(
```

```
                      x = temp,
                      y = deltaU)) +
  geom_line() +
  labs(title = "Figure 6",
       subtitle = "Delta u for Reaction Two") +
  xlab("Temperature (Celcius)") +
  ylab("Delta u (kJ/mol)")

ggsave("pics/Figure6.pdf")

# deltaS

deltaSRxn2 <- (deltaH2Rxn2 - deltaURxn2) / tempK

ReactionTwo <- mutate(ReactionTwo, "deltaS" = deltaSRxn2)

sPlotRxn2 <- ggplot(data = ReactionTwo,
                    mapping = aes(
                      x = temp,
                      y = deltaS)) +
  geom_line() +
  labs(title = "Figure 7",
       subtitle = "Delta S for Reaction Two") +
  xlab("Temperature (Celcius)") +
  ylab("Delta S (kJ/mol)")

ggsave("pics/Figure7.pdf")
```