Linear Regression of the Pressure Dataset

Jon Kinsey

Tue Dec 30 13:49:10 2014

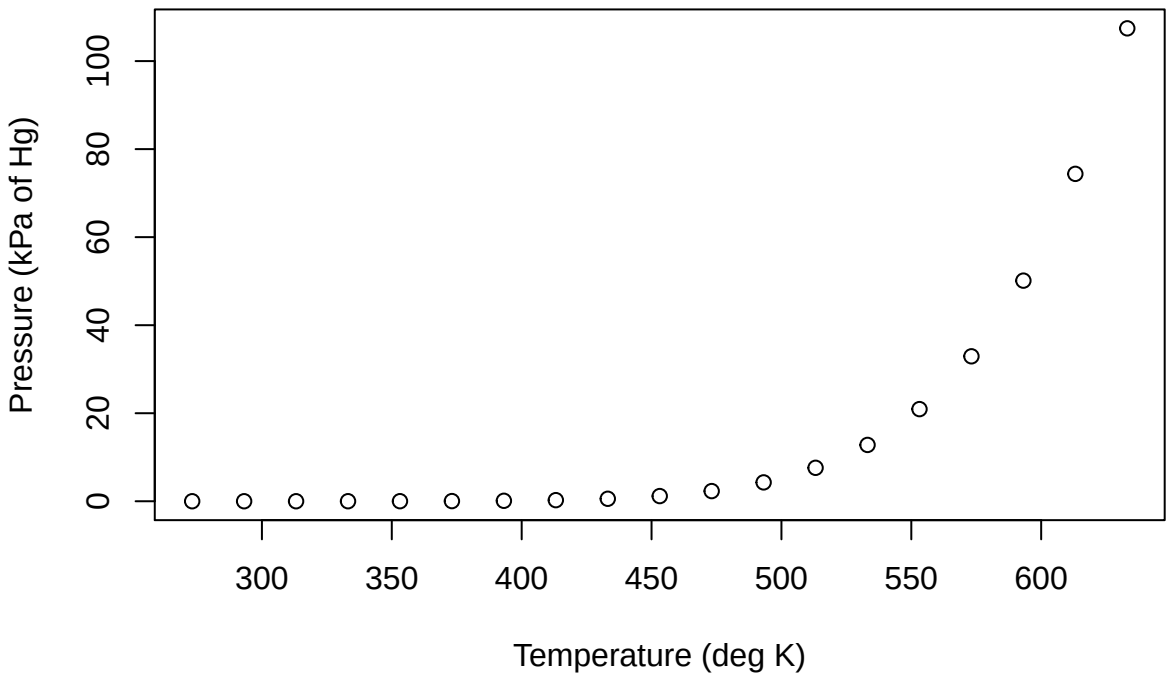
```
#
# Here we examine the the dependence of mercury vapor pressure on temperature.
# The vapor pressure is the pressure exerted by the vapor on the liquid mercury.
# The data frame has 19 observations on 2 variables.
# [, 1]  temperature    numeric temperature (deg C)
# [, 2] pressure    numeric pressure (mm)
# Source: Handbook of Chemistry and Physics, CRC Press (1973)
#
data(pressure)
head(pressure)
```

```
##   temperature pressure
## 1           0   0.0002
## 2          20   0.0012
## 3          40   0.0060
## 4          60   0.0300
## 5          80   0.0900
## 6         100   0.2700
```

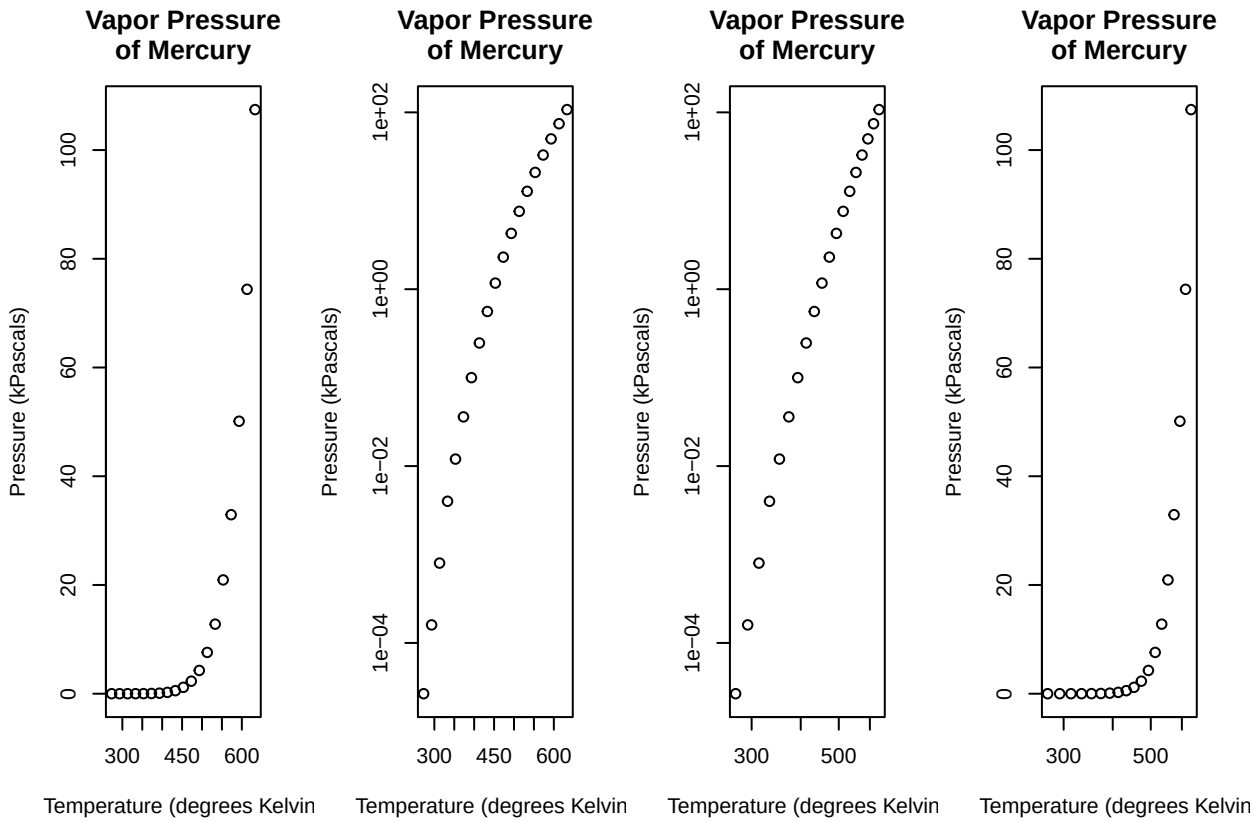
```
#
summary(pressure)
```

```
##   temperature     pressure
##  Min.   :  0   Min.   :  0.0002
##  1st Qu.: 90   1st Qu.:  0.1800
##  Median :180   Median :  8.8000
##  Mean   :180   Mean   :124.3367
##  3rd Qu.:270   3rd Qu.:126.5000
##  Max.   :360   Max.   :806.0000
```

```
# Convert temperature to Kelvin
pressure$temperature = pressure$temperature + 273.15
# Convert pressure to kiloPascals
pressure$pressure = pressure$pressure * .1333
#
par(mfrow=c(1,1))    # reset the graphics defaults
plot(pressure, xlab = "Temperature (deg K)",
     ylab = "Pressure (kPa of Hg)",
     main = "Vapor Pressure of Mercury")
```



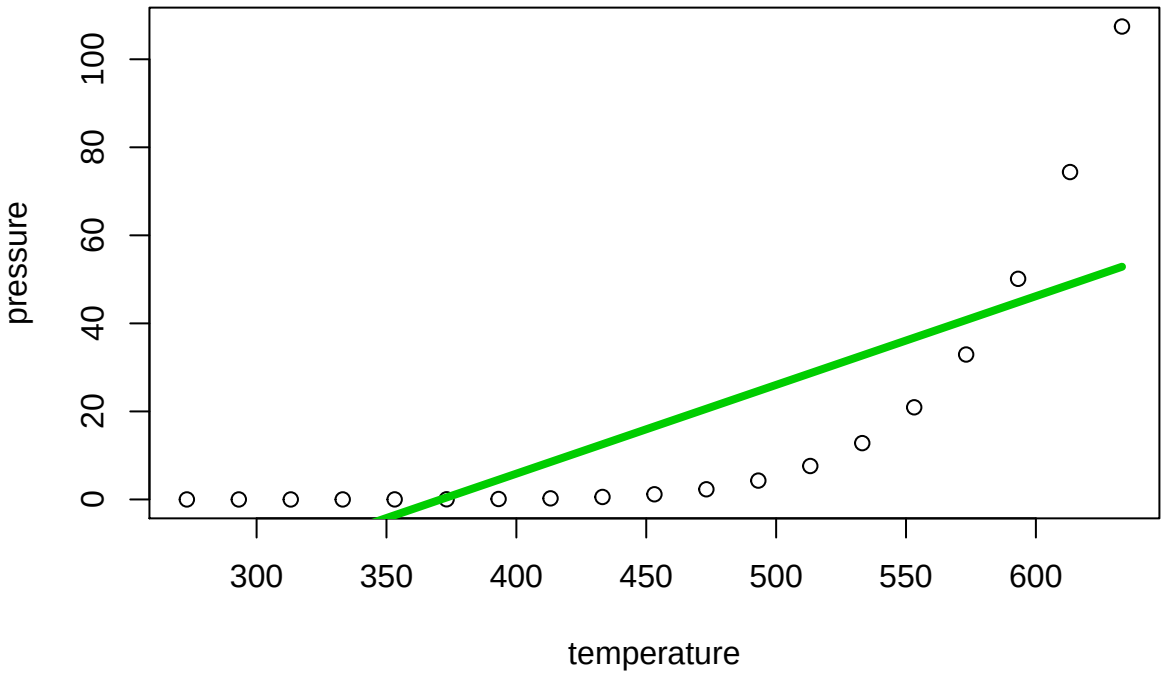

```
#
# Plot various log transformations
## Identity, log(y), power, log(x)
## Identity, exponential, power, log

par(mfrow=c(1,4))    # one row of four graphs
plot(pressure ~ temperature, data=pressure,
     main="Vapor Pressure\nof Mercury",
     xlab="Temperature (degrees Kelvin)", ylab="Pressure (kPascals)")
plot(pressure ~ temperature, data=pressure,
     main="Vapor Pressure\nof Mercury",
     xlab="Temperature (degrees Kelvin)", ylab="Pressure (kPascals)", log="y")
plot(pressure ~ temperature, data=pressure,
     main="Vapor Pressure\nof Mercury",
     xlab="Temperature (degrees Kelvin)", ylab="Pressure (kPascals)", log="xy")
plot(pressure ~ temperature, data=pressure,
     main="Vapor Pressure\nof Mercury",
     xlab="Temperature (degrees Kelvin)", ylab="Pressure (kPascals)", log="x")
```




```
#
par(mfrow=c(1,1))    # reset the graphics defaults
#
# Start with a linear model
linear.model <- lm(pressure ~ temperature, data=pressure)
summary(linear.model)
```

```
##
## Call:
## lm(formula = pressure ~ temperature, data = pressure)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -21.072 -15.604  -4.378   9.637  54.577
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -74.7835    19.6282  -3.810 0.001400 **
## temperature   0.2016     0.0421   4.788 0.000171 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.1 on 17 degrees of freedom
## Multiple R-squared:  0.5742, Adjusted R-squared:  0.5492
## F-statistic: 22.93 on 1 and 17 DF,  p-value: 0.000171
```

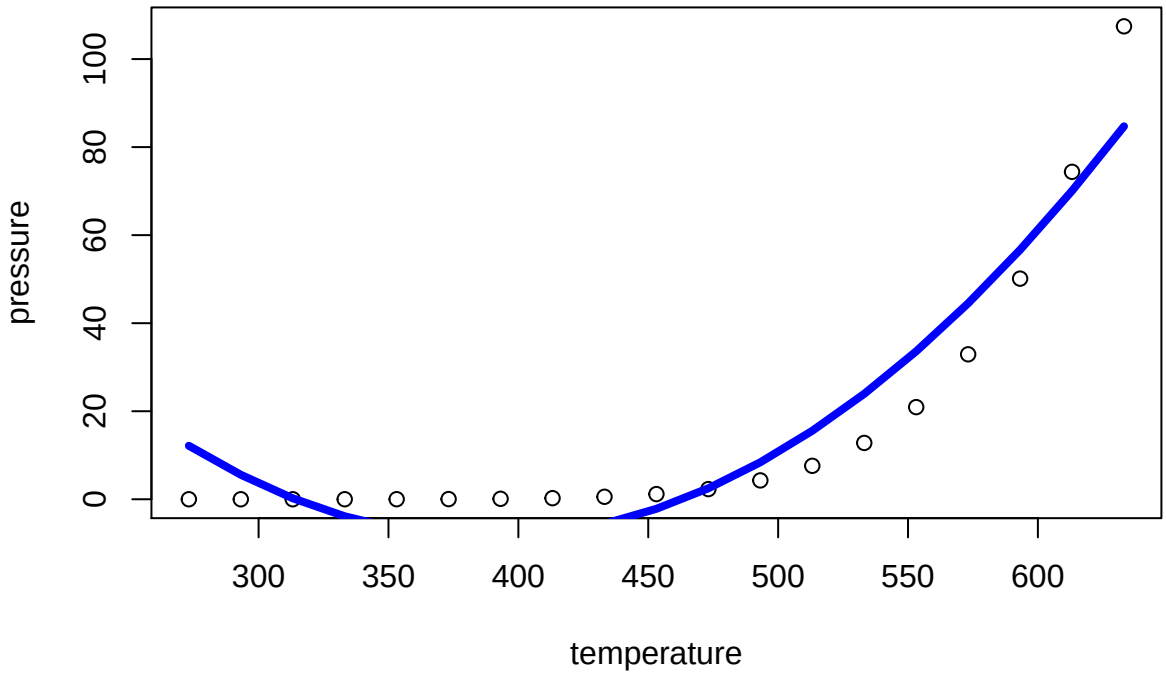
```
newdf <- data.frame(temperature=seq(min(pressure$temperature),
                                    max(pressure$temperature), len=19))
plot(pressure ~ temperature, data=pressure)
lines(newdf$temperature, predict(linear.model, newdf), col=3, lw=4)
```




```
#
# This has an R-squared of 0.574. Not a good fit to the data.
#
# Lets try a quadratic fit now.
quad.model <- lm(pressure ~ temperature + I(temperature^2), data=pressure)
summary(quad.model)
```

```
##
## Call:
## lm(formula = pressure ~ temperature + I(temperature^2), data = pressure)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.6824  -7.2504  -0.1803   6.4301  22.7109
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.272e+02  4.229e+01   5.373 6.22e-05 ***
## temperature      -1.214e+00  1.941e-01  -6.255 1.15e-05 ***
## I(temperature^2)  1.562e-03  2.129e-04   7.336 1.67e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.92 on 16 degrees of freedom
## Multiple R-squared:  0.9024, Adjusted R-squared:  0.8902
## F-statistic:    74 on 2 and 16 DF,  p-value: 8.209e-09
```

```
plot(pressure ~ temperature, data=pressure)
lines(newdf$temperature, predict(quad.model, newdf), col=4, lw=4)
```




```
#
# This has an R-squared of 0.902. This is better but there is room
# for improvement.
#
power.model <- lm(log(pressure) ~ log(temperature), data=pressure)
summary(power.model)
```

```
##
## Call:
## lm(formula = log(pressure) ~ log(temperature), data = pressure)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.2390 -0.3594  0.2142  0.4625  0.6045
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -108.1138     3.1744  -34.06   <2e-16 ***
## log(temperature)   17.6150     0.5212   33.80   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5736 on 17 degrees of freedom
## Multiple R-squared:  0.9853, Adjusted R-squared:  0.9845
## F-statistic:  1142 on 1 and 17 DF,  p-value: < 2.2e-16
```

```
#
# This has an R-squared of 0.985
#
# Now, lets plot the 3 models together against the data
plot(pressure ~ temperature, data=pressure)
lines(newdf$temperature, predict(linear.model, newdf), col=3, lw=4)      # green
lines(newdf$temperature, predict(quad.model, newdf), col=4, lw=4)        # blue
lines(newdf$temperature, exp(predict(power.model, newdf)), col=2, lw=4) # red
text(585, 7.0, substitute(plain("Power model")),col=2)
text(585, .75, substitute(plain("R-squared: ") * r2,
                          list(r2=summary(power.model)$r.squared)),col=2)
```