MAT215: Project 4: Snow melting exp't - nonlinear models

Zaigham Abbas Randhawa

Fri, 4/7/17

Data

Consider the data from the improved experiment that you used for your Project 3.

For Project 3, the consensus seems to have been that, even though models $melting\ time = \beta_0 + \beta_1 temperature$ and $melting\ time = \beta_0 + \beta_1 temperature + \beta_2 temperature^2$ had reasonable fits, we did not like these models due to the fact that they would eventually break the laws of physics by either predicting negative melting times or predicting high melting times for very high temperatures.

For simplicity, let Y be the melting time for 100 g of snow and let X be the temperature in degrees Fahrenheits.

```
library(mosaic)
f1<-read.table(file="experiment2.txt", header=TRUE)
f1$temp = as.character(f1$temp)
time=(f1$time)
weight=(f1$wght)
time100=time*(100/weight)
fx=cbind.data.frame(f1, time100)
```

Questions

Question 1

A usual statistical report contains numerical data summaries before analyses results.

a. Using lm and $confint$ functions, fill in the following table. $\bar{x}_{130}$ is the average observed melting time for 130°F, and LB and UB are lower and upper bounds of the 95% confidence interval for the true average melting time at 130°F. (6 pts)

```
mod=lm(fx$time100~fx$temp-1)
Intervals=confint(mod)
```

Temperature (°F)	Average melting time (s)	95% CI
130	38.75	(32.46, 45.04)
140	21.31	(15.02, 27.61)
150	15.36	(9.06, 21.65)

Table 1. Temperatures with the average melting times for 100 grams of snow, and the upper and lower bounds for the respective 95% confidence intervals of these times

b. Using *aggregate* function calculate means and std. deviations of the melting times for each observed temperature and reconstruct the table above. Remember, the approxmate 95% CI is given by $\bar{x}\pm$ 2*std. error for each temperature. Std. error = std. deviation/$\sqrt{k}$ where k represents the number of observations from which each $\bar{x}$ was calculated. (6 pts)

```
means=aggregate(fx$time100, by = list(temperature=fx$tempN), mean)
sds=aggregate(fx$time100, by = list(temperature=fx$tempN), sd)
fb=cbind.data.frame(means, 2*(means-sds)/(4^.5), 2*(means+sds)/(4^.5))
```

Temperature (°F)	Average melting time (s)	95% CI
130	38.75	(33.65, 43.85)
140	21.31	(15.63, 27.00)
150	15.36	(9.49, 21.23)

Table 2. Same as Table 1 except that here the confidence intervals are estimated

c. Make a plot that gives the same information as the tables above. Your x-axis is temperature and y-axis is melting time. The plot should look similar to the plot obtained by running the following code

```
x=unique(fx$tempN)
y=means$x
sd=sds[,2]/2
plot(x, y, ylim=c(10, 50), xlab=expression(paste("Temperature (", degree,"F)")),
     ylab="Melting time (s)",las=1, col="red1")
segments(x, y - 2*sd, x , y + 2*sd, col = "blue1")
epsilon <- 0.02
segments(x - epsilon, y - 2*sd, x + epsilon, y - 2*sd, col = "blue1")
segments(x - epsilon, y + 2*sd, x + epsilon, y + 2*sd, col = "blue1")
legend('topright', legend = c("Mean Melting Time", "95% Confidence Interval"),
       col = c("Red1","blue1"), pch=c(1,-1), lty = c(-1, 1), cex=.8)
```

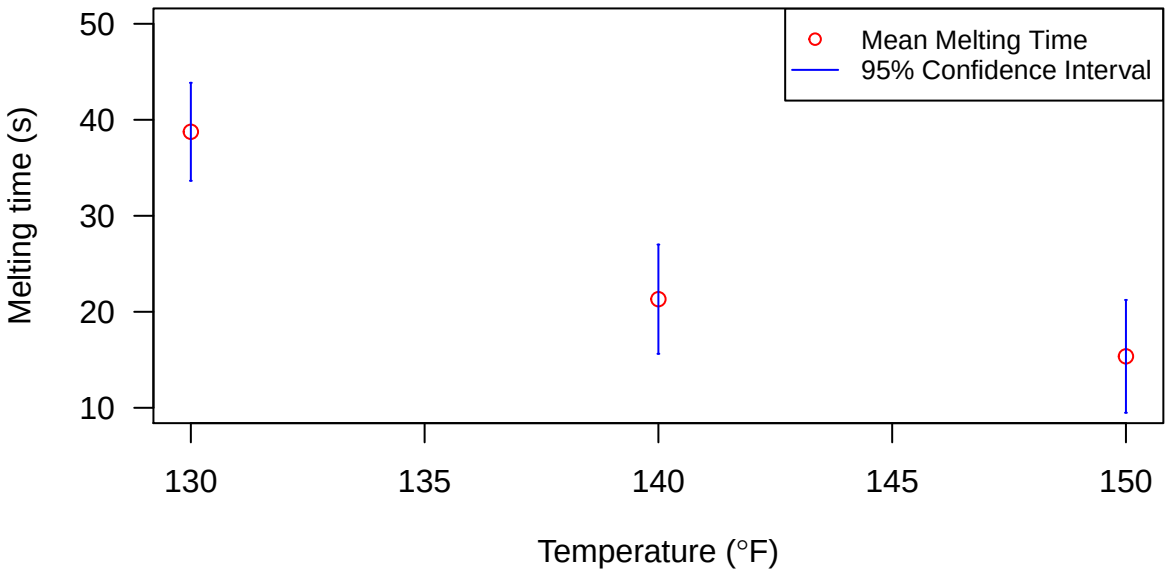

Figure 1. Melting times at given temperatures along with the respective 95% confidence intervals

Question 2

Since we weren't the greatest fans of the straight line or the quadratic model, we would like to fit a model of the form

$$Y = e^{\beta_0+\beta_1\times X} \tag{1}$$

to our data.

a. Show mathematically that this is a nonlinear model. Use LaTex's *eqnarray* environment introduced in the last project (and used above) to type up your work. (3 pts)

$$\frac{\mathrm{d}}{\mathrm{d}\beta_1}Y = X \times e^{\beta_0+\beta_1\times X}$$

Y's derivative with respect to β_1 still has β_1 in it. This implies that Y is not linear.

b. Fit the model shown in Equation 3. Use *R*'s *nls* function to fit the model. Write down your estimated model. (5 pts)

```
fit_something=nls(time100~(exp(a*tempN+b)),start = c(a=0, b=0),data=fx)
```

$$Y = e^{-.05\times Temperature+10.21} \tag{2}$$

c. Predict the melting times for $x = seq(125, 155, 0.01)$. Plot your observed data and add the predicted values (line) to the plot. (5 pts)

```
X_prediction_set=data.frame(tempN = seq(125, 155, 0.01))
X_prediction_set$Y_prediction_set=predict(fit_something,X_prediction_set)
plot(time100~tempN, data=fx, ylab="Melting time (s)",
col="blue",las=1,xlab=expression(paste("Temperature (", degree,"F)")),
xlim=c(125,155), ylim=c(0,50))
lines(X_prediction_set$tempN,X_prediction_set$Y_prediction_set, col="red")
legend('topright', legend=c("Observed Values", "Fitted Values"),
       col=c("blue","red"),pch=c(1,-1), lty=c(-1,1), cex=.9)
```

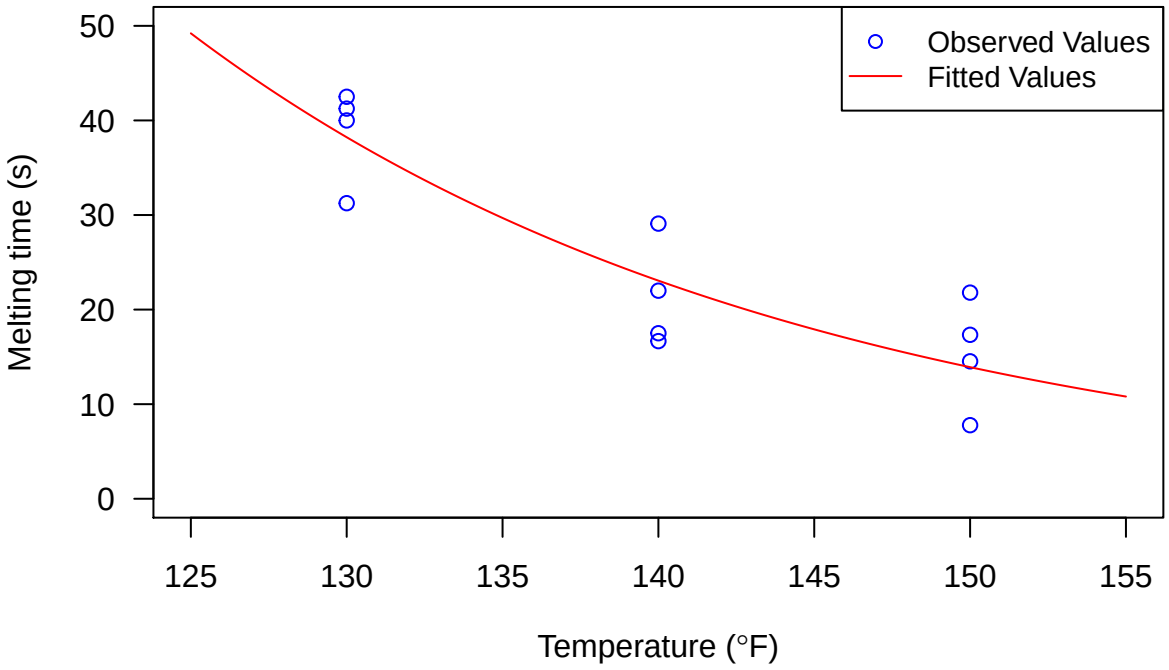

Figure 2. Scatter plot for the melting times of 100 grams of snow at various temperatures along with an exponential fit

d. Calculate R^2 value for this model and compare it to the straight line and quadratic models from Project 3. Describe the fit of this model. (8 pts)

The R^2 value for this model is 0.89. It means that this exponential model roughly explains about 90% percent of the variation in melting times for a 100 grams of snow. The predictions by this model are thus pretty accurate. The R^2 for the linear and quadratic models are 0.75 and 0.81. As far as the linear model is concerned, the difference (1.4) is quiet significant and the exponential model is definitely much better. And for the quadratic model, the difference is only 0.8-almost half the previous difference. Though this value is small, it is still quiet significant. Again, we can conclude that the exponential model is better than the quadratic model.

The linear and quadratic model also went against as to what we were expecting. The linear one predicted negative melting times at high temperatures and the exponential one predicted that after a certain temperature the melting time should increase with an increase in temperature. However, the predictions from this new exponential model align with what we were expecting: As the temperature increases, the melting time should decrease; this predicted melting time should also reach an asymptote at 0s, i.e. tend to reach but never actually touch 0s-which is exactly what happens. Thus the exponential model fits well and is the best model.

e. This questions concerns $\frac{\mathrm{d}}{\mathrm{d}X}Y$ where Y is given in Equation 3.

i. Calculate the instantenous rate of change in melting time as a function of the temperature. (5 pts)

$$\frac{\mathrm{d}}{\mathrm{d}Temperature}MeltingTime = -0.05 \times e^{-0.05 \times Temperature + 10.21}$$

ii. What is the instantenous rate of change in melting time when temperatures is 146°F? (2 pts)

$$\frac{\mathrm{d}}{\mathrm{d}Temperature}MeltingTime_{Temperature=146}$$
$$= -0.05 \times e^{-.05 \times 146 + 10.21} \simeq -.03 \ \frac{\mathrm{s}}{^\circ F}$$

f. Recently we had a big snow storm. Suppose that after the storm, the temperature is 40°F. Estimate the time necessary for 1 kg of snow to melt at this temperature? Do you have any concerns about your estimate? (5 pts)

$$\begin{aligned} &Time\ taken\ to\ melt\ 1kg\ of\ snow\ at\ 40^\circ F \\ &= 10 \times Time\ taken\ to\ melt\ 100\ grams\ of\ snow\ at\ 40^\circ F \\ &= 10 \times (e^{-.05 \times 40 + 10.21}) \\ &= 10 \times (367.754) \\ &= 36775.4\ s \end{aligned}$$

Extrapolation is defined as trying to make guesses for response variables at values of explanatory variables which lie way out of the given data range. It is something which should be avoided. In this scenario, we are extrapolating on two explanatory variables: the mass of ice and the temperature. Our model's data only had an ice mass of a 100 grams, and the temperatures for the experiments ranged from 130°F to 160°F. Here we are trying to predict for a mass which is ten times larger at a temperature which is well below the aforementioned temperature range. Therefore, the ability of our models to predict accurately at this mass and temperature comes into question.

Question 3

a. The model $Y = e^{\beta_0 + \beta_1 X}$ can also be fitted using the *lm* function. Use an appropriate transformation of Y here and then fit the model using the *lm* function and write down your estimated model. (5 pts)

```
ln_time<-log(time100)
fx1=cbind.data.frame(fx, ln_time)
fit_stuff <- lm(ln_time ~ tempN, data=fx1)
```

$$ln(Melting\ Time) = 10.01 - .05 * Temperature \quad (3)$$

b. From the model in part (a), what is your best guess f or μ_{146}, the true average melting time of 100 g of snow at 146°F? Construct a 95% confidence interval for μ_{146}. (6 pts)

```
myfun=makeFun(fit_stuff)
At146=data.frame(tempN=146)
prediction=exp(predict(fit_stuff, At146, interval="confidence"))
```

At 146°F, the true average melting time (i.e. μ_{146}) is 16.80s. The 95% confidence interval for this temperature has a lower bound at 13.25s and an upper bound at 21.29s.

c. From the model in part (a), what is your best guess for Y at 146°F. Construct a 95% prediction interval for Y at 146°F. (6 pts)

```
prediction1=exp(predict(fit_stuff, At146, interval="prediction"))
```

At 146°F, the predicted melting time (i.e. Y_{146}) is also 16.80s. However, the prediction interval is much wider than the confidence interval. The lower bound for the 95% prediction interval at this temperature is 8.32s and the upper bound is 33.90s.

d. Briefly explain the difference between the intervals from parts (b) and (c). (5 pts)

For a confidence interval we are given a certain set of data, and we calculate a specific statistic along with a given level of confidence as a range, e.g. calculating the average melting time for a given mass of snow with only just a sample of temperature-melting time pairs from our data set. The only problem that we deal with here is the sampling uncertainty.

A prediction interval on the other hand is used to do predictions at very specific values of explanatory variables, e.g. estimating the melting time for a specific temperature. Making such a prediction accurately is very difficult. Here we are not just dealing with the possible sampling bias, but also with the variation of the individual values around our predicted value-who knows what we may predict at a certain instance and what might be the actual value at that particular instance?

In simpler terms, the prediction interval has to deal with a far more risky situation than the confidence interval and therefore needs to be wider.

e. Construct a plot that contains your observed data and predicted melting times for $x = seq(125, 155, 0.01)$ with 95% confidence and prediction intervals for each one of these x's. Plot the predicted means with the solid line, CI's with the dashed lines, and PI's with the dotted lines (there should be 5 lines on your plot). (8 pts) Hint: You can get CI's and PI's from the *predict* function. Look up the help manual of *predict* by typing *?predict*. In particular, focus on the *interval* option. You can also get the std. errors if you set $se.fit$ to $TRUE$.

```
X_prediction_set$Y_prediction_set1=exp(predict(fit_stuff,X_prediction_set))
```

```r
plot(time100~tempN, data=fx, ylab="Melting time (s)",
col="blue",las=1,xlab=expression(paste("Temperature (", degree,"F)")),
xlim=c(125,155),ylim=c(0,120), pch=1)

lines(X_prediction_set$tempN,X_prediction_set$Y_prediction_set1, lty=1,
      col="black")


prediction_confidence=exp(predict(fit_stuff, X_prediction_set,
                                  interval="confidence"))
prediction_prediction=exp(predict(fit_stuff, X_prediction_set,
                                  interval="prediction"))

lines(X_prediction_set$tempN,prediction_confidence[,2], lty=2, col="red")
lines(X_prediction_set$tempN,prediction_confidence[,3], lty=2, col="red")

lines(X_prediction_set$tempN,prediction_prediction[,2], lty=3, col="blue")
lines(X_prediction_set$tempN,prediction_prediction[,3], lty=3, col="blue")
legend('topright', legend = c("Observed Values", "Predicted Values",
"Confidence Interval","Predicton Interval"), col = c("blue","Black", "Red",
"blue"), pch=c(1,-1,-1,-1), lty = c(-1, 1,2,3), cex=.8)
```

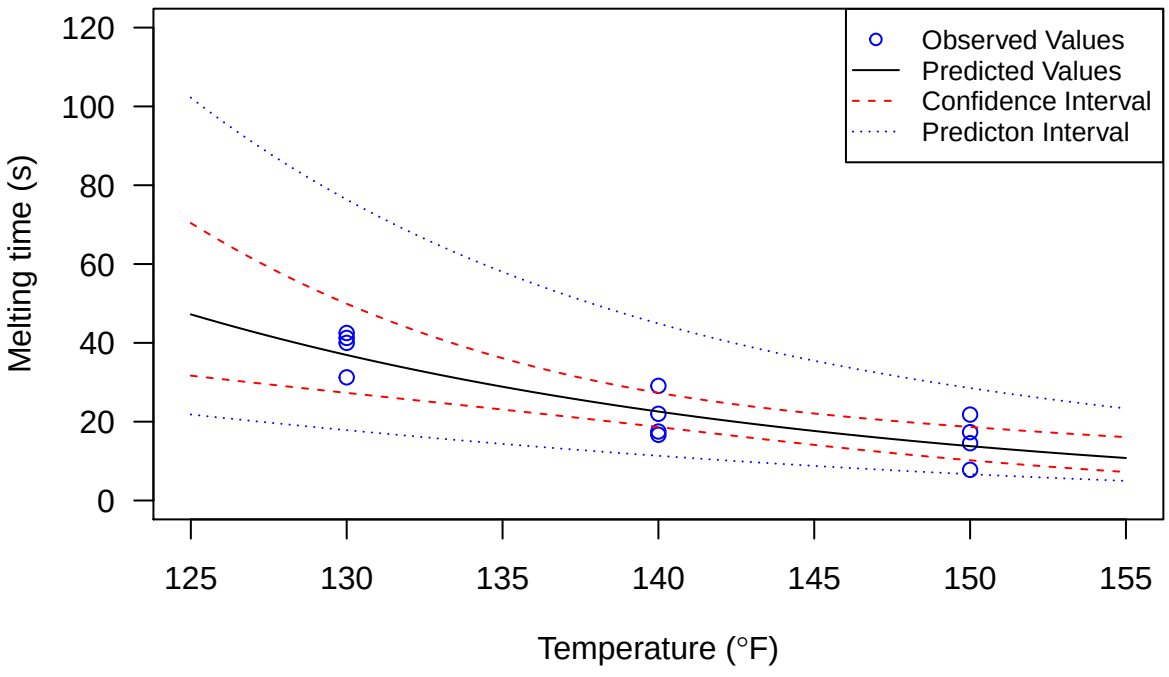

Figure 3. Scatter plot for the melting times of 100 grams of snow at various temperatures along with predicted times and 95% confidence and prediction intervals for these times