a) Make a scatterplot of the data.

![Scatterplot](image)

b) Find the least squares line through the data, the sample correlation between x and y, and the coefficient of determination for the SLR fit.

<table>
<thead>
<tr>
<th>Coefficients</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>66.4177</td>
</tr>
<tr>
<td>stress</td>
<td>-0.9009</td>
</tr>
</tbody>
</table>

- coefficient of determination: 0.6325
- sample correlation: -0.7953102

(c) What are values of for this data set \( \bar{x} \) and \( \sum_{i=1}^{n}(x_i - \bar{x})^2 \)?

\[
\text{mean(stress)} = 20
\]
\[
\text{var(stress)}*(\text{length(stress)}-1) = 1412.5
\]

(d) What is the value of \( S_{LF} \) in this data set? Use this and make 95% two-sided confidence limits for the standard deviation of time to fracture for any fixed stress (according to the SLR model).

\( S_{LF} = 9.124307 \)
chiup<-sqrt(qchisq(.975,length(stress)-2))
chilw<-sqrt(qchisq(.025,length(stress)-2))

summary(timefit1)$sigma*sqrt(length(stress)-2)/c(chiup,chilw)

95% two-sided confidence limits : (6.163078 17.480083)

d) Give 95% confidence limits for slope and intercept in the SLR model. Is it possible that time to fracture doesn't change with stress? Explain.

No, since the slope is negative, which indicating that as the stress increases, the fracture time will decrease.

The 95% confidence limits for slope and intercept in the SLR model can be calculated by the equation $b_1 \pm t*0.2428$ $b_0 \pm t*5.6481$ or as follows:

```
confint(timefit1, level=.95)
```


e) Give an ANOVA table for SLR in the format presented in class.

```
aov(timefit1)
```  

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>DF</th>
<th>MS</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression</td>
<td>1146.38</td>
<td>1</td>
<td>1146.38</td>
<td>13.77</td>
</tr>
<tr>
<td>Error</td>
<td>666.02</td>
<td>8</td>
<td>83.25</td>
<td></td>
</tr>
<tr>
<td>C.T.</td>
<td>1812.4</td>
<td>9</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

f) Plot the data, the least squares line, 95% confidence limits for the mean time to fracture as a function of stress, and 95% prediction limits for the next time to fracture one stress level at a time.

```
predict.lm(timefit1, se.fit=TRUE, interval="prediction", level=.95)
```  

```
## $fit
##     fit lwr   upr
## 1 64.16549 40.020763 88.31021
## 2 61.91327 38.301794 85.52475
## 3 57.40885 34.642118 80.17558
## 4 52.90442 30.659935 75.14891
## 5 50.65221 28.540210 72.76421
## 6 48.40000 26.332337 70.46766
## 7 43.89558 21.651086 66.14006
## 8 39.39115 16.624419 62.15788
## 9 34.88673 11.275245 58.49821
```
#This has narrower confidence
out1c<-predict.lm(timefit1, se.fit=TRUE, interval="confidence", level=.95)
out1p<-predict.lm(timefit1, se.fit=TRUE, interval="prediction", level=.95)

# Predictions with new dataset, you can try this choice
predict(lm(failuretime~stress))

new <- data.frame(stress = seq(3, 50, 5))
predict(lm(failuretime~stress), new, se.fit = TRUE)
2 a)

```
hardnessfit1$coef
## (Intercept)        pctcu        temp
##   161.33646    32.96875 -0.08550
```

b)

```
chiup <- sqrt(qchisq(.975, length(hardness) - 3))
chilw <- sqrt(qchisq(.025, length(hardness) - 3))
summary(hardnessfit1)$sigma * sqrt(length(hardness) - 3) / c(chiup, chilw)
```

```
.95 CI : (2.607536 6.920762)
```

c) 

```
> confint(hardnessfit1, level = .95)
     2.5 %       97.5 %
(Intercept) 135.4735441 187.19937258
pctcu        -4.9307663  70.86826626
temp        -0.1076423 -0.06335769
```

d) 

```
new <- data.frame(pctcu=c(.1), temp=c(1100))
predict(lm(hardness ~ pctcu + temp), new, se.fit = TRUE, interval = "confidence", level = 0.95)
$fit
  fit      lwr     upr
1 70.58333 67.87146 73.2952
predict(lm(hardness ~ pctcu + temp), new, se.fit = TRUE, interval = "prediction", level = 0.95)
$fit
  fit      lwr      upr
1 70.58333 61.58908 79.57758
```

The result of the prediction limits is much wider than those of confidence limits.

**Confidence intervals** tell you about how well you have determined the mean. Assume that the data really are randomly sampled from a Gaussian distribution. If you do this many times, and calculate a confidence interval of the mean from each sample, you'd expect about 95% of those intervals to include the true value of the population mean. The key point is that the confidence interval tells you about the likely location of the true population parameter.

**Prediction intervals** tell you where you can expect to see the next data point sampled. Assume that the data really are randomly sampled from a Gaussian distribution. Collect a sample of data and calculate a prediction interval. Then sample one more value from the population. If you do this many times, you'd expect that next value to lie within that prediction interval in 95% of the samples. The key point is that the prediction interval tells you about the distribution of values, not the uncertainty in determining the population mean. Prediction intervals must account for both the uncertainty in knowing the value of the population mean, plus data scatter. So a prediction interval is always wider than a confidence interval.

e) 

```
new <- data.frame(pctcu=c(.02), temp=c(1000))
```
predict(lm(hardness~pctcu+temp), new, se.fit = TRUE, interval="confidence", level=0.95)
$fit
      fit     lwr      upr
1 76.49583 71.36235 81.62932
predict(lm(hardness~pctcu+temp), new, se.fit = TRUE, interval="prediction", level=0.95)
$fit
      fit     lwr      upr
1 76.49583 66.50109 86.49058

f) The prediction by using data in part d) is much accurate in perdition acco
rding to the narrower CI created in part d). This is because point (0.1,1100)
is in the middle of the data set.

g) plot(hardnessfit1)

hardnessfit2<-lm(hardness~pctcu+temp+tempsq)
plot(hardnessfit2)

From the plot above, it is obvious that residual plot in hardnessfit1 has a
quadratic pattern , while in the right one no such pattern exists, and it is
more randomly bounded around the zero line. Therefore, we can say it is a goo
d idea to fit model with quadratic term.

h) summary(hardnessfit2)$r.squared
R^2=0.9828802 which is greater than those in hardnessfit1 which is 0.8990734.

i) confint(hardnessfit2, level=.95)

  2.5 %       97.5 %
(Intercept) 4.083721e+02  6.981175e+02
pctcu       1.609179e+01  4.984571e+01
temp       -1.027327e+00 -5.198400e-01
tempsq     -1.889268e-04  4.094065e-04
The .95 CI for tempsq is (1.889268e-04, 4.094065e-04).

```
summary(hardnessfit2)
```

Coefficients:

| Estimate  | Std. Error | t value | Pr(>|t|) |
|-----------|------------|---------|----------|
| (Intercept) | 5.532e+02  | 8.806   | 2.17e-05 *** |
| pctcu      | 3.297e+01  | 4.505   | 0.001990 ** |
| temp       | -7.736e-01 | -7.030  | 0.000109 *** |
| tempsq     | 2.992e-04  | 6.258   | 0.000244 *** |

And the p-value for tempsq is 0.000244, which is small enough to reject the hypothesis that the coefficient of tempsq=0, indicating that the quadratic term is important in the model. Hence, it is a good idea to employ the quadratic term in modeling.