MATH - 4360: Linear Statistical Models

Chapter 2: Simple Linear Regression

Suthakaran Ratnasingam

Simple Linear Regression

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \quad i = 1, 2, \ldots, n$$

1. y is called as the dependent or study variable
2. X is called as the independent or explanatory variable
3. he terms β_0 and β_1 are the parameters of the model. The parameter β_0 is called as an intercept term, and the parameter β_1 is called as the slope parameter. **These parameters are usually called as regression coefficients.**

Example 2.1 The Rocket Propellant Data

```
rm(list = ls())
# It is assumed that you already have installed the
# following R packages. If not, please install them using
# the R function: install.packages('package_name')
library(olsrr)
library(ggfortify)
library(ggplot2)
library(tidyverse)
library(car)
library(Rcpp)
library(GGally)
library(leaps)
library(matlib)
# y = Shear Strength x = Age of Propellant
data1 = read.table("ex21.txt", header = TRUE)
names(data1)
```

```
## [1] "y" "x"
```

```
head(data1)
```

```
##         y     x
## 1 2158.70 15.50
## 2 1678.15 23.75
## 3 2316.00  8.00
## 4 2061.30 17.00
## 5 2207.50  5.50
## 6 1708.30 19.00
```

```
y = data1$y
x = data1$x
str(data1)
```

```
## 'data.frame':    20 obs. of  2 variables:
##  $ y: num  2159 1678 2316 2061 2208 ...
##  $ x: num  15.5 23.8 8 17 5.5 ...
```

```
# Mean of Age_of_Propellant
mean(x)
```

Summary Statistics

```
## [1] 13.3625
```

```
# Mean of Shear_Strength
mean(y)
```

```
## [1] 2131.358
```

```
# Sum of (Age_of_Propellant)^2
sum(x^2)
```

```
## [1] 4677.688
```

```
# Sum of (Shear_Strength)^2
sum(y^2)
```

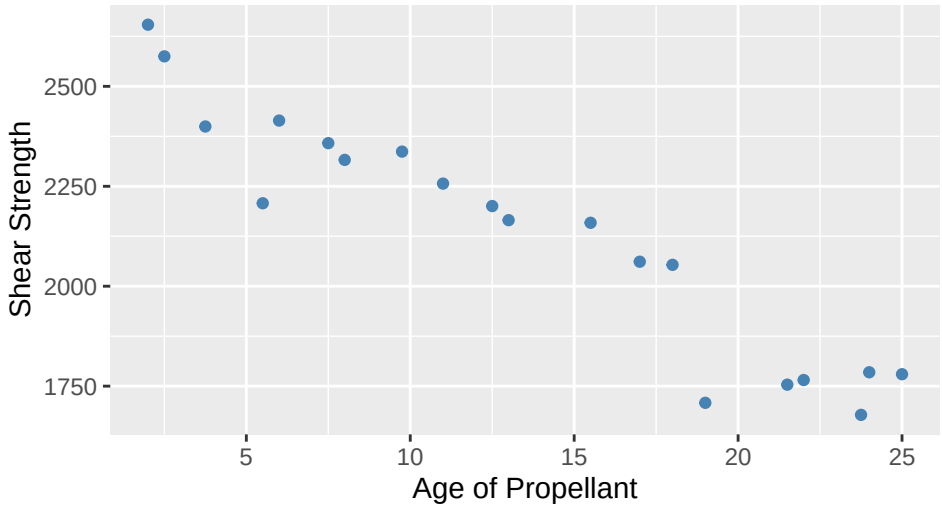
```
## [1] 92547433
```

```
# Sum of (Age_of_Propellant)*(Shear_Strength)
sum((x) * (y))
```

```
## [1] 528492.6
```

```
ggplot(data1, aes(x = x, y = y)) + geom_point(color = "steelblue") +
    labs(x = "Age of Propellant", y = "Shear Strength", title = "")
```

Scatter plot



Least - Squares Estimation of the Parameters

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, \quad i = 1, 2, \ldots, n$$

Thus, the least - squares criterion is

$$S(\beta_0, \beta_1) = \sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n} \left(y_i - \beta_0 - \beta_1 x_i\right)^2$$

The partial derivatives of $S(\beta_0, \beta_1)$ with respect to β_0 is

$$\frac{\partial S(\beta_0, \beta_1)}{\partial \beta_0} = -2 \sum_{i=1}^{n} \left(y_i - \beta_0 - \beta_1 x_i\right)$$

and the partial derivative of $S(\beta_0, \beta_1)$ with respect to β_1 is

$$\frac{\partial S(\beta_0, \beta_1)}{\partial \beta_1} = -2 \sum_{i=1}^{n} \left(y_i - \beta_0 - \beta_1 x_i\right) x_i$$

The solutions of β_0 and β_1 are obtained setting

$$\frac{\partial S(\beta_0, \beta_1)}{\partial \beta_0} = 0$$
$$\frac{\partial S(\beta_0, \beta_1)}{\partial \beta_1} = 0$$

Therefore,

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}}, \quad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

We then calculate the three sums of squares defined above.

```
Sxy = sum((x - mean(x)) * (y - mean(y)))
Sxx = sum((x - mean(x))^2)
Syy = sum((y - mean(y))^2)
c(Sxy, Sxx, Syy)
```

```
## [1]  -41112.654    1106.559 1693737.601
```

We can find $\hat{\beta}_0$ and $\hat{\beta}_1$

```
beta_1_hat = Sxy/Sxx
beta_0_hat = mean(y) - beta_1_hat * mean(x)
c(beta_0_hat, beta_1_hat)
```

```
## [1] 2627.82236  -37.15359
```

```
# Using the lm() function in R
Fit1 = lm(y ~ x, data = data1)
summary(Fit1)
```

```
##
## Call:
## lm(formula = y ~ x, data = data1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -215.98  -50.68   28.74   66.61  106.76
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2627.822     44.184   59.48  < 2e-16 ***
## x            -37.154      2.889  -12.86 1.64e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 96.11 on 18 degrees of freedom
## Multiple R-squared:  0.9018, Adjusted R-squared:  0.8964
## F-statistic: 165.4 on 1 and 18 DF,  p-value: 1.643e-10
```

Some of the components can be extracted using a function.

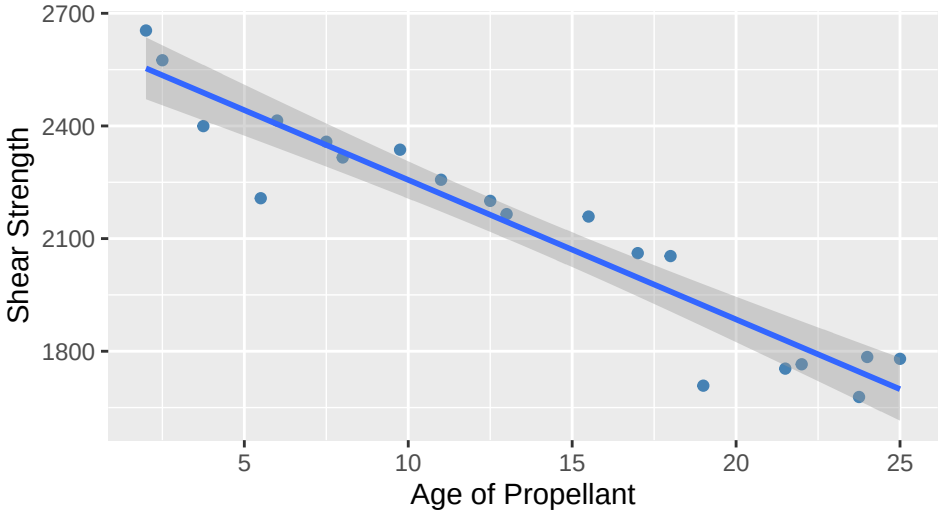
```
coefficients(Fit1)
```

```
## (Intercept)           x
##  2627.82236   -37.15359
```

We see that our model takes the form of `Shear Strength = 2627.822  - 37.154 * (Age of Propellant)`

Adding best fit lines It is often useful to summarize the relationship displayed in the scatterplot, using a best fit line. Many types of lines are supported, including linear, polynomial, and nonparametric (loess). By default, 95% confidence limits for these lines are displayed

```
ggplot(data1, aes(x = x, y = y)) + geom_point(color = "steelblue") +
    geom_smooth(method = "lm") + labs(x = "Age of Propellant",
    y = "Shear Strength", title = "")
```



simple linear regression estimate in matrix form

```
n = nrow(data1)
X = cbind(rep(1, n), data1$x)
Y = data1$y

Xt_X = t(X) %*% X  # X'X matrix
Xt_X_inv = solve(Xt_X)  # (X'X)^(-1)
Xt_y = t(X) %*% y  # X'Y
beta_hat = Xt_X_inv %*% Xt_y
beta_hat
```

```
##             [,1]
## [1,] 2627.82236
## [2,]  -37.15359
```

2.2.3: Estimation of σ^2

$$\hat{\sigma}^2 = \frac{SS_{Res}}{(n-2)} = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{(n-2)} = MS_{Res}$$

$$\hat{\sigma} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{(n-2)}}$$

```
n = length(y)
y_hat = Fit1$fitted.values
sqrt(sum((y - y_hat)^2)/(n - 2))
```

```
## [1] 96.10609
```

```
# Method 2
summary(Fit1)$sigma
```

```
## [1] 96.10609
```

It is also important to understand if the these coefficients are statistically significant. In other words, can we state these coefficients are statistically different then 0? To do that we can start by assessing the `standard error (SE)`. The SE for β_0 and β_1 are computed with:

Using the assumption that y_i's are independently distributed, the variance of β_1 is

$$
\begin{aligned}
Var(\hat{\beta}_1) &= Var\Big(\sum_{i=1}^{n} c_i y_i\Big) \\
&= \sum_{i=1}^{n} c_i^2 Var(y_i) + \sum_i \sum_{i \neq j} c_i c_j Cov(y_i, y_j) \\
&= \sigma^2 \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{S_{xx}^2} \\
&= \sigma^2 \frac{S_{xx}}{S_{xx}^2} \\
&= \frac{\sigma^2}{S_{xx}}
\end{aligned}
$$

where $c_i = (x_i - \bar{x})/S_{xx},\ i = 1, 2, \ldots, n$. Note that

$$
\begin{aligned}
\sum_{i=1}^{n} c_i &= 0, \quad \text{and} \\
\sum_{i=1}^{n} c_i x_i &= 1
\end{aligned}
$$

The variance of $\hat{\beta}_0$ is

$$
\begin{aligned}
Var(\hat{\beta}_0) &= Var(\bar{y} - \hat{\beta}_1 \bar{x}) \\
&= Var(\bar{y}) + \bar{x}^2 Var(\hat{\beta}_1) - 2\bar{x} Cov(\bar{y}, \hat{\beta}_1)
\end{aligned}
$$

First, we will find that

$$
\begin{aligned}
Cov(\bar{y}, \hat{\beta}_1) &= E\big[(\bar{y} - E(\bar{y}))(\hat{\beta}_1 - E(\hat{\beta}_1)\big] \\
&= E\big[\bar{\epsilon}\big(\sum_{i=1}^{n} c_i y_i - \beta_1\big)\big] \\
&= \frac{1}{n}(0 + 0 + 0 + 0) = 0
\end{aligned}
$$

Thus,

$$
Var(\hat{\beta}_0) = \sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}}\right)
$$

Covariance: The covariance between $\hat{\beta}_0$ and $\hat{\beta}_1$ is

$$
\begin{aligned}
Cov(\hat{\beta}_0, \hat{\beta}_1) &= Cov(\bar{y} - \hat{\beta}_1 \bar{x},\ \hat{\beta}_1) \\
&= Cov(\bar{y},\ \hat{\beta}_1) - \bar{x} Cov(\hat{\beta}_1, \hat{\beta}_1) \\
&= Cov(\bar{y},\ \hat{\beta}_1) - \bar{x} Var(\hat{\beta}_1) \\
&= -\frac{\bar{x}}{S_{xx}} \sigma^2
\end{aligned}
$$

```
vcov(Fit1)
```

```
##             (Intercept)           x
## (Intercept)   1952.2181 -111.535941
## x             -111.5359    8.346937
```

```
coef(Fit1)
```

```
## (Intercept)           x
##  2627.82236   -37.15359
```

Estimate of variances of $\hat{\beta}_0$ and $\hat{\beta}_1$ and Test of hypothesis on individual regression coefficients

2.3.2: Testing Significance of Regression

A very important special case of the hypotheses

$$H_0 : \beta_1 = 0 \quad \text{vs} \quad H_1 : \beta_1 \neq 0$$

These hypotheses relate to the significance of regression.

Testing of hypotheses for intercept term

Suppose that we wish to test the hypothesis that the intercept equals a constant, say β_{00}. The appropriate hypotheses are

$$H_0 : \beta_0 = \beta_{00} \quad \text{vs} \quad H_1 : \beta_0 \neq \beta_{00}$$

where β_{00} is some given constant.

```
summary(Fit1)$coef
```

```
##               Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 2627.82236  44.183912  59.47464 4.063559e-22
## x            -37.15359   2.889107 -12.85989 1.643344e-10
```

Confidence Intervals on β_0 and β_1, and σ^2

Case 1: When σ^2 is known:

If the errors are normally and independently distributed, then the sampling distribution of both $\dfrac{(\hat{\beta}_1 - \beta_1)}{SE(\hat{\beta}_1)}$ and $\dfrac{(\hat{\beta}_0 - \beta_0)}{SE(\hat{\beta}_0)}$ follows $N(0, 1)$.

- Therefore, a $100(1-\alpha)\%$ confidence interval (CI) on the slope β_1 is given by

$$\hat{\beta}_1 - Z_{\alpha/2}\, SE(\hat{\beta}_1) \leq \hat{\beta}_1 \leq \hat{\beta}_1 + Z_{\alpha/2}\, SE(\hat{\beta}_1)$$

$$\left(\hat{\beta}_1 - Z_{\alpha/2} \sqrt{\frac{\sigma^2}{S_{xx}}}, \quad \hat{\beta}_1 + Z_{\alpha/2} \sqrt{\frac{\sigma^2}{S_{xx}}} \quad \right)$$

- A $100(1-\alpha)\%$ confidence interval (CI) on the slope β_0 is given by

$$\hat{\beta}_0 - Z_{\alpha/2}\, SE(\hat{\beta}_0) \leq \hat{\beta}_0 \leq \hat{\beta}_0 + Z_{\alpha/2}\, SE(\hat{\beta}_0)$$

$$\left(\hat{\beta}_1 - Z_{\alpha/2} \sqrt{\sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}} \right)}, \quad \hat{\beta}_1 + Z_{\alpha/2} \sqrt{\sigma^2 \left(\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}} \right)} \right)$$

Case 1: When σ^2 is unknown:

If the errors are normally and independently distributed, then the sampling distribution of both $\dfrac{(\hat{\beta}_1 - \beta_1)}{SE(\hat{\beta}_1)}$ and $\dfrac{(\hat{\beta}_0 - \beta_0)}{SE(\hat{\beta}_1)}$ is t with $(n-2)$ degrees of freedom.

- Therefore, a $100(1-\alpha)\%$ confidence interval (CI) on the slope β_1 is given by

$$\hat{\beta}_1 - t_{\alpha/2,n-2}\, SE(\hat{\beta}_1) \leq \hat{\beta}_1 \leq \hat{\beta}_1 + t_{\alpha/2,n-2}\, SE(\hat{\beta}_1)$$

$$\left(\hat{\beta}_1 - t_{\alpha/2,n-2} \sqrt{\frac{SS_{Res}}{(n-2)S_{xx}}}, \quad \hat{\beta}_1 + t_{\alpha/2,n-2} \sqrt{\frac{SS_{Res}}{(n-2)S_{xx}}} \right)$$

- A $100(1-\alpha)\%$ confidence interval (CI) on the slope β_0 is given by

$$\hat{\beta}_0 - t_{\alpha/2,n-2}\, SE(\hat{\beta}_0) \leq \hat{\beta}_0 \leq \hat{\beta}_0 + t_{\alpha/2,n-2}\, SE(\hat{\beta}_0)$$

$$\left(\hat{\beta}_0 - t_{\alpha/2,n-2} \sqrt{\frac{SS_{Res}}{(n-2)} \left(\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}} \right)}, \quad \hat{\beta}_0 + t_{\alpha/2,n-2} \sqrt{\frac{SS_{Res}}{(n-2)} \left(\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}} \right)} \right)$$

We see that our model results provide the SE (noted as `std.error`). We can use the `SE` to compute the 95% confidence interval for the coefficients:

To get this information in R we can simply use:

```
confint(Fit1)
```

```
##                  2.5 %    97.5 %
## (Intercept) 2534.99540 2720.6493
## x            -43.22338  -31.0838
```

Confidence interval for σ^2 $SS_{Res}/\sigma^2 = (n-2)MS_{Res}/\sigma^2$ is chi square with $(n-2)$ degrees of freedom. Thus,

$$P\left\{\chi^2_{1-\alpha/2,n-2} \leq \frac{(n-2)MS_{Res}}{\sigma^2} \leq \chi^2_{\alpha/2,n-2}\right\} = 1-\alpha$$

The corresponding $100(1-\alpha)\%$ CI on σ^2 is

$$\left(\frac{(n-2)MS_{Res}}{\chi^2_{\alpha/2,n-2}}, \quad \frac{(n-2)MS_{Res}}{\chi^2_{1-\alpha/2,n-2}}\right)$$

```
sigma = sigma(Fit1)
sigma
```

```
## [1] 96.10609
```

```
n = length(data1$y)
k = 1
alpha = 0.05

lower = (n - (k + 1)) * sigma^2/qchisq(alpha/2, df = n - (k +
    1), lower.tail = FALSE)
upper = (n - (k + 1)) * sigma^2/qchisq(1 - alpha/2, df = n -
    (k + 1), lower.tail = FALSE)
c(lower, upper)
```

```
## [1]  5273.516 20199.245
```

Analysis of Variance (ANOVA)

$$\sum_{i=1}^{n}(y_i - \bar{y})^2 = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2 + \sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$SS_T = SS_R + SS_{Res}$$

We can use function `anova()` to observe sum of squares of the model

```
anova(Fit1)
```

```
## Analysis of Variance Table
##
## Response: y
##           Df  Sum Sq Mean Sq F value    Pr(>F)
## x          1 1527483 1527483  165.38 1.643e-10 ***
## Residuals 18  166255    9236
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova_fit = anova(Fit1)
as.table(cbind(SS = c(Regression = anova_fit[1, 2], Residual = anova_fit[2,
    2], Total = sum(anova_fit[1:2, 2])), Df = c(anova_fit[1,
    1], anova_fit[2, 1], sum(anova_fit[1:2, 1])), MS = c(anova_fit[1,
    3], anova_fit[2, 2]/anova_fit[2, 1], NA), `F-Test` = c(anova_fit[1,
    3]/anova_fit[2, 3], NA, NA)))
```

```
##                        SS           Df           MS       F-Test
## Regression 1527482.7433       1.0000 1527482.7433     165.3768
## Residual    166254.8581      18.0000    9236.3810
## Total      1693737.6014      19.0000
```

2.4.2: Interval Estimation of the Mean Response

We assume that x_0 is any value of the regressor variable within the range of the original data on x used to fit the model. An unbiased point estimator of $E(y|x_0)$ is found from the fitted model as

$$\widehat{E(y|x_0)} = \hat{\mu}_{y|x_0} = \hat{\beta}_0 + \hat{\beta}_1 x_0$$

The variance of $\hat{\mu}_{y|x_0}$ is

$$\begin{aligned} Var(\hat{\mu}_{y|x_0}) &= Var(\hat{\beta}_0 + \hat{\beta}_1 x_0) \\ &= Var\left[\bar{y} + \hat{\beta}_1(x_0 - \bar{x})\right] \\ &= \frac{\sigma^2}{n} + \frac{\sigma^2(x_0 - \bar{x})^2}{S_{xx}} \\ &= \sigma^2\left(\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}}\right), \quad \text{where } Cov(\bar{y}, \hat{\beta}_1) = 0 \end{aligned}$$

A $100(1-\alpha)\%$ confidence interval on the mean response at the point $x = x_0$ is

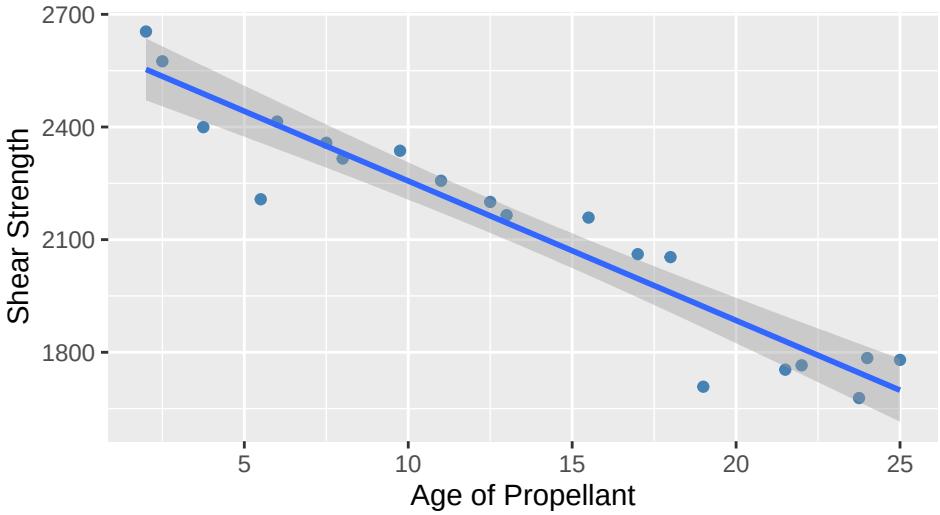
$$\hat{\mu}_{y|x_0} - t_{\alpha/2,n-2}\sqrt{MS_{Res}\left(\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}}\right)} \leq E(y|x_0) \leq \hat{\mu}_{y|x_0} + t_{\alpha/2,n-2}\sqrt{MS_{Res}\left(\frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}}\right)}$$

we will obtain the 95% CI on the mean response at x = x0 = 13.3625.

```
new_obs = data.frame(x = c(13.3625))
predict(Fit1, newdata = new_obs, interval = c("confidence"),
    level = 0.95)
```

```
##        fit      lwr      upr
## 1 2131.357 2086.209 2176.506
```

```
ggplot(data1, aes(x = x, y = y)) + geom_point(color = "steelblue") +
    geom_smooth(method = "lm", se = TRUE) + labs(x = "Age of Propellant",
    y = "Shear Strength", title = "")
```



Prediction interval Estimation

Because the future observation y_0 is independent of $\hat{y}_0$, the predictive variance of $\hat{y}_0$ is

$$Var(\psi) = Var(y_0 - \hat{y}_0) = \sigma^2\left(1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}}\right)$$

The estimate of predictive variance can be obtained by replacing σ^2 by its estimate $\hat{\sigma}^2 = MS_{Res}$ as

$$\begin{aligned} Var(\psi) &= \hat{\sigma}^2\left(1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}}\right) \\ &= MS_{Res}\left(1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}}\right) \end{aligned}$$

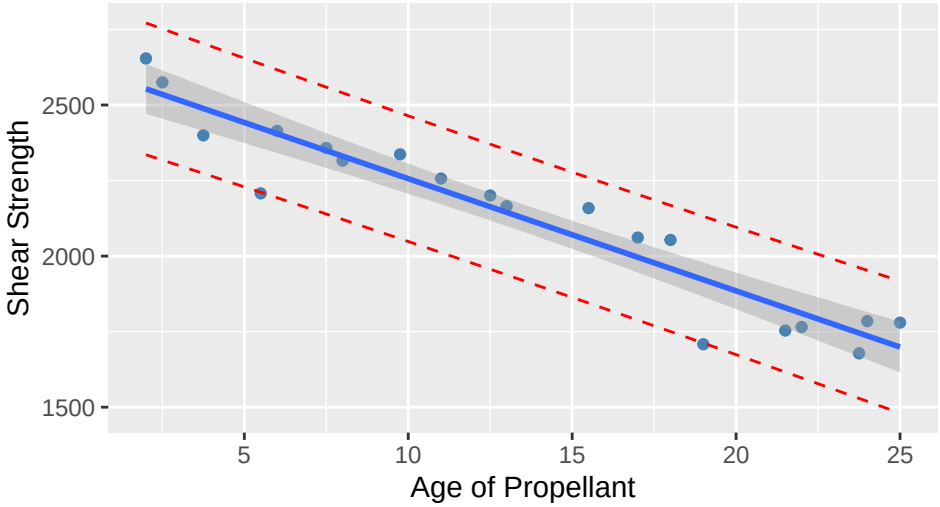
A $100(1-\alpha)\%$ prediction interval on a future observation at x_0 is

$$\hat{y}_0 - t_{\alpha/2,n-2}\sqrt{MS_{Res}\left(1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}}\right)} \leq y_0 \leq \hat{y}_0 + t_{\alpha/2,n-2}\sqrt{MS_{Res}\left(1 + \frac{1}{n} + \frac{(x_0 - \bar{x})^2}{S_{xx}}\right)}$$

```
new_obs = data.frame(x = c(13.3625))
predict(Fit1, newdata = new_obs, interval = c("prediction"),
    level = 0.95)
```

```
##        fit     lwr      upr
## 1 2131.357 1924.46 2338.255
```

```
temp_var = predict(Fit1, interval = c("prediction"), level = 0.95)
new_df = cbind(data1, temp_var)
ggplot(new_df, aes(x = x, y = y)) + geom_point(color = "steelblue") +
    geom_smooth(method = "lm", se = TRUE) + geom_line(aes(y = lwr),
    color = "red", linetype = "dashed") + geom_line(aes(y = upr),
    color = "red", linetype = "dashed") + labs(x = "Age of Propellant",
    y = "Shear Strength", title = "")
```



Assessing Model Accuracy

Next, we want to understand the extent to which the model fits the data. This is typically referred to as the `goodness-of-fit`. We can measure this quantitatively by assessing three things:

- Residual standard error
- R squared (R^2)
- F-statistic

The RSE is an estimate of the standard deviation of ϵ. Roughly speaking, it is the average amount that the response will deviate from the true regression line. We get the RSE at the bottom of `summary(Fit1)`, we can also get it directly with

```
sigma(Fit1)
```

```
## [1] 96.10609
```

Similar to RSE, the R^2 can be found at the bottom of `summary(Fit1)` but we can also get it directly with r-square. The result suggests that TV advertising budget can explain 65% of the variability in our sales data.

$$R^2 = \frac{SS_R}{SS_T} = 1 - \frac{SS_{Res}}{SS_T}$$

```
summary(Fit1)$r.squared
```

```
## [1] 0.9018414
```

As a side note, in a simple linear regression model the R^2 value will equal the squared correlation between X and Y:

```
cor(x, y)^2
```

```
## [1] 0.9018414
```

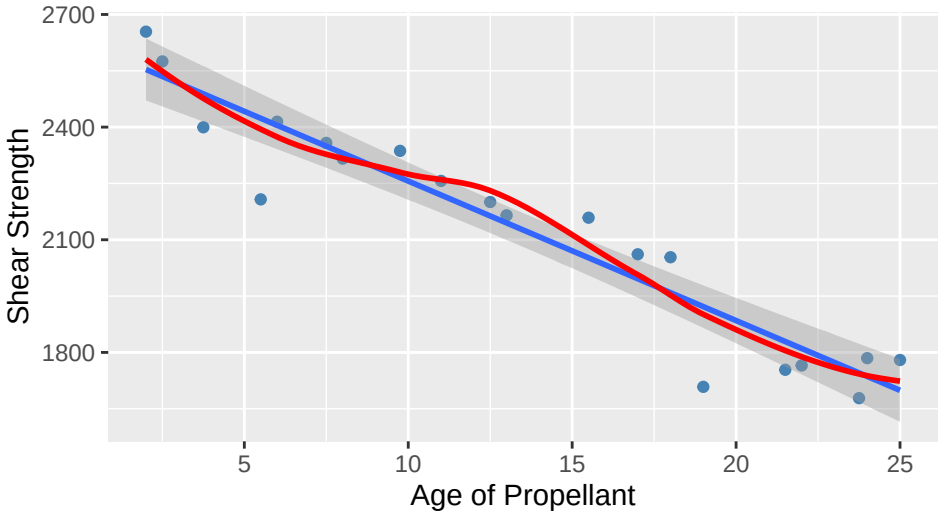
Lastly, the F-statistic tests to see if at least one predictor variable has a non-zero coefficient. This becomes more important once we start using multiple predictors as in multiple linear regression; however, we will introduce it here.

Assessing Our Model Visually

The major assumptions that we have made thus far in our study of regression analysis are as follows: * The relationship between the response y and the regressors is linear, at least approximately.

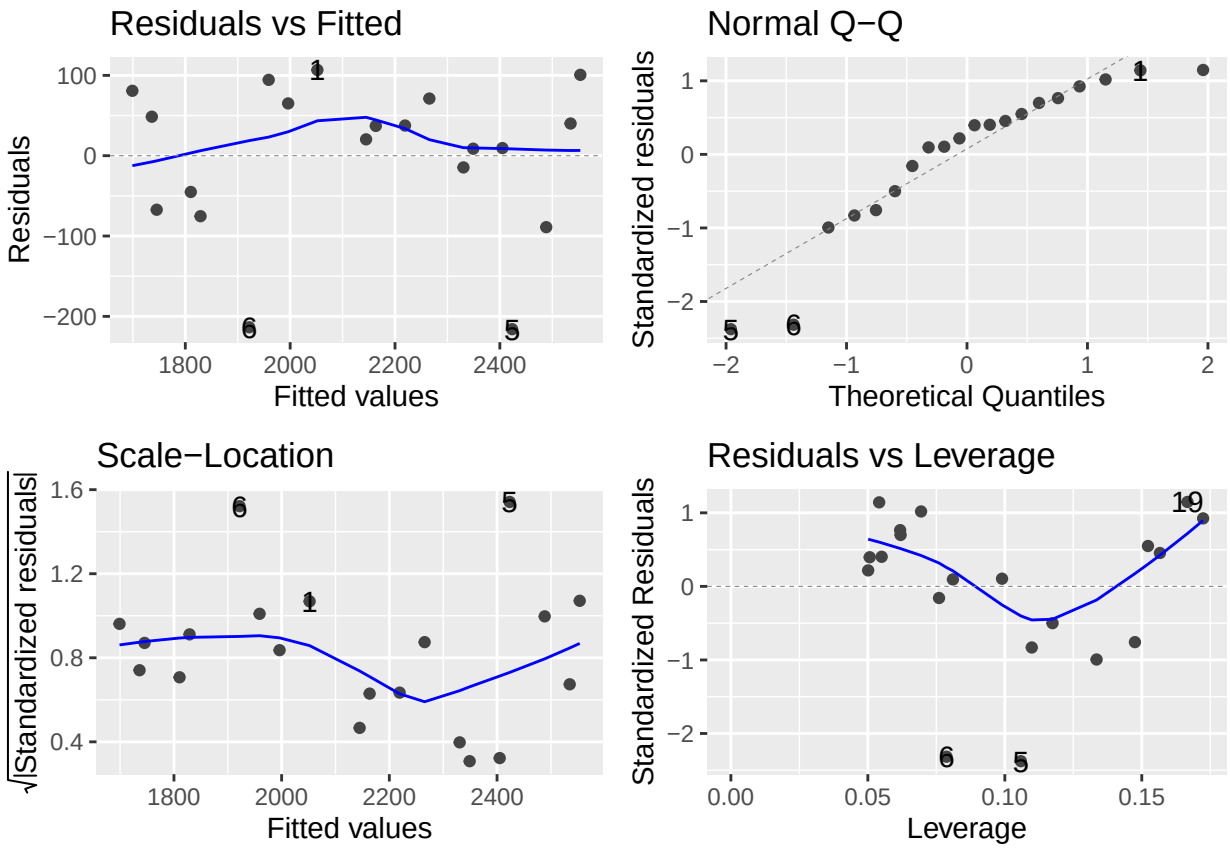
- The error term ϵ has zero mean.
- The error term ϵ has constant variance σ^2.
- The errors are uncorrelated.
- The errors are normally distributed.

```
ggplot(data1, aes(x = x, y = y)) + geom_point(color = "steelblue") +
    geom_smooth(method = "lm") + geom_smooth(se = FALSE, color = "red") +
    labs(x = "Age of Propellant", y = "Shear Strength", title = "")
```



An important part of assessing regression models is visualizing residuals. If you use `autoplot(Fit1)` in the `ggfortify` package, you will get four residual plots. We will learn more about model adequacy checking in later chapters.

```
# ols_plot_diagnostics(Fit1) # olsrr r package
autoplot(Fit1)  # R package ggfortify
```



References

- Introduction to Linear Regression Analysis, 5th Edition, by Douglas C. Montgomery, Elizabeth A. Peck, G. Geoffrey Vining (Wiley), ISBN: 978-0-470-54281-1.
- R Core Team (2020). R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing.
- RStudio Team (2020). RStudio: Integrated Development Environment for R. Boston, MA: RStudio, PBC.