# Linear Regression Confidence Intervals

In a previous example, linear regression was examined through the simple regression (https://rpubs.com/aaronsc32/simple-linear-regression) setting, i.e., one independent variable. Fitting a linear model allows one to answer questions such as:

- What is the mean response for a particular value of x?
- What value will the response be assuming a particular value of x?

In the case of the `cars` dataset, which was examined in the previous example and will be used again here, these questions can be more precise:

- What is the mean stopping distance of the cars given a particular speed?
- What is the stopping distance at the speed of x?

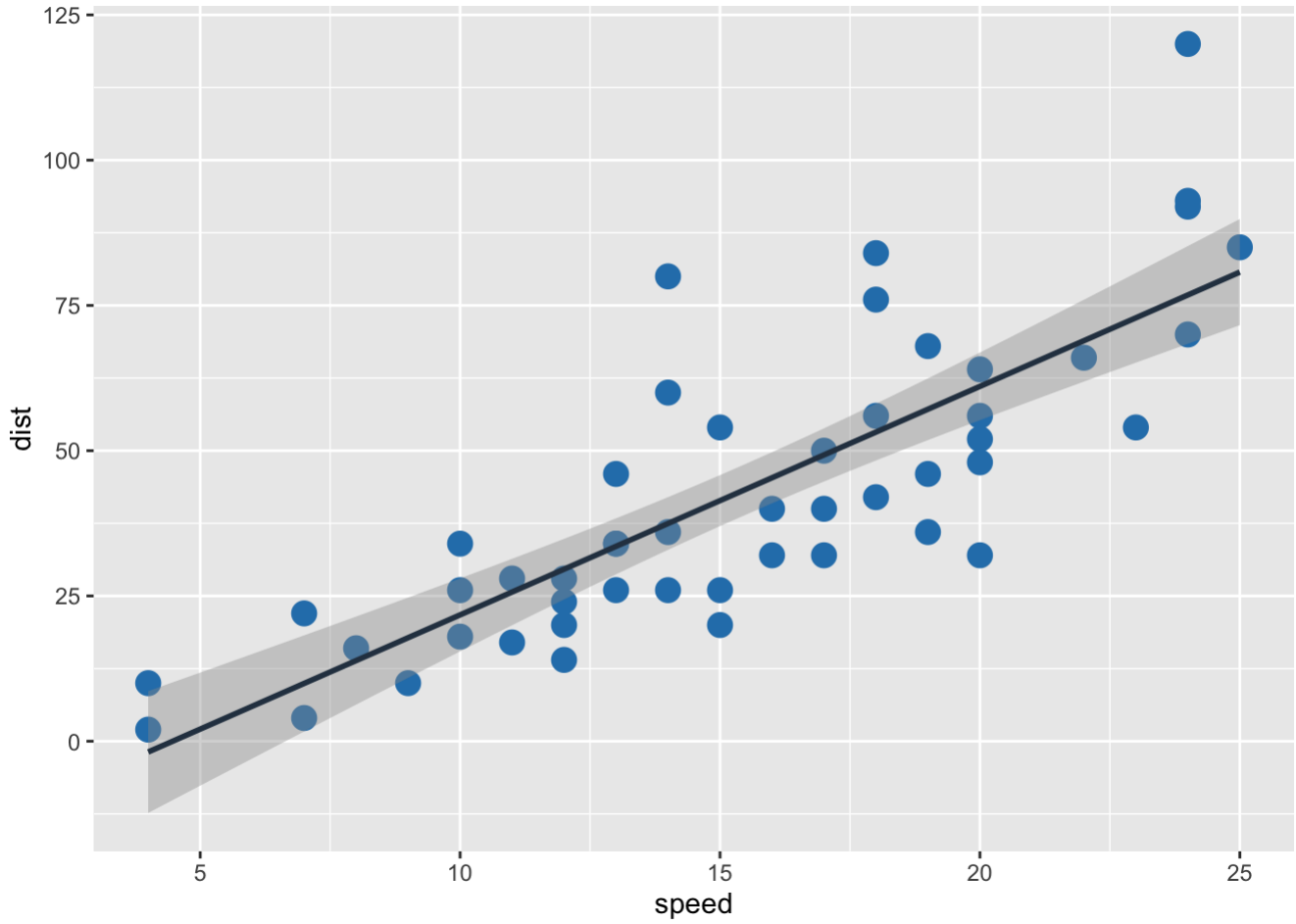
Confidence and prediction intervals are often formed to answer questions such as the above. Intervals allow one to estimate a range of values that can be said with reasonable confidence (typically 95%) contains the true population parameter. It should be noted that although the questions above sound similar, there is a difference in estimating a mean response and predicting a new value. This difference will be seen in the interval equations. This post will explore confidence and prediction intervals as well as the confidence 'bands' around a linear regression line.

Begin by loading the `cars` dataset and the `ggplot2` package.

```
library(ggplot2)
data("cars")
```

Once again, plot the two variables as a scatterplot and draw a linear regression line through the points. The gray bands around the line represent the standard error of the regression line. This plot will be recreated using base R graphics in a function below.

```
ggplot(cars, aes(x=speed, y=dist)) +
  geom_point(color='#2980B9', size = 4) +
  geom_smooth(method=lm, color='#2C3E50')
```




Fit a linear regression model and print a summary.

```
cars.lm <- lm(dist ~ speed, data = cars)
summary(cars.lm)
```

```
##
## Call:
## lm(formula = dist ~ speed, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## speed         3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

As before, the speed of the car has a relatively high and significant relationship with stopping distance. Knowing this relationship, one can then form and answer questions such as those above. A confidence interval is created to respond to the first question, “What is the mean stopping distance of the cars given a particular speed?”.

# Confidence (Mean) Intervals

When estimating the confidence interval (also called the mean interval), the question one is trying to answer is typically as mentioned above: What is the mean stopping distance of the car at a certain speed?

The confidence interval around the mean response, denoted $\mu_y$, when the predictor value is $x_k$ is defined as:

$$\hat{y}_h \pm t_{\alpha/2,n-2}\sqrt{MSE\left(\frac{1}{n}+\frac{(x_k-\bar{x})^2}{\sum(x_i-\bar{x}^2)}\right)}$$

Where $\hat{y}_h$ is the fitted response for predictor value $x_h$, $t_{\alpha/2,n-2}$ is the t-value with $n-2$ degrees of freedom, while $\sqrt{MSE\left(\frac{1}{n}+\frac{(x_k-\bar{x})^2}{\sum(x_i-\bar{x}^2)}\right)}$ is the standard error of the fit.

To find the confidence interval in R, create a new `data.frame` with the desired value to predict. The prediction is made with the `predict()` function. The `interval` argument is set to ‘confidence’ to output the mean interval.

```
new.dat <- data.frame(speed=30)
predict(cars.lm, newdata = new.dat, interval = 'confidence')
```

```
##        fit      lwr      upr
## 1 100.3932 87.43543 113.3509
```

From the output, the fitted stopping distance at a speed of 30 mph is just above 100 feet. The confidence interval of (87.44, 113.35) signifies the range in which the true population parameter lies at a 95% level of confidence.

# Prediction Intervals

The prediction interval is rather similar to the confidence interval in calculation, but as mentioned earlier, there are significant differences. The prediction interval equation is defined as:

$$\hat{y}_h \pm t_{\alpha/2,n-2}\sqrt{MSE\left(1+\frac{1}{n}+\frac{(x_k-\bar{x})^2}{\sum(x_i-\bar{x})^2}\right)}$$

Where $\hat{y}_h$ is the fitted reponse at predictor value $x_k$ and the critical t-value is $t_{\alpha/2,n-2}$ with $n-2$ degrees of freedom. $\sqrt{MSE\left(1+\frac{1}{n}+\frac{(x_k-\bar{x})^2}{\sum(x_i-\bar{x})^2}\right)}$ is the standard error of prediction.

To find the prediction interval in R, the `predict()` function is utilized once again, but this time, the interval argument is given 'prediction.'

```
predict(cars.lm, newdata = new.dat, interval = 'prediction')
```

```
##        fit      lwr     upr
## 1 100.3932 66.86529 133.921
```

Notice the fitted value is the same as before, but the interval is wider. This is due to the additional term in the standard error of prediction. It should be noted prediction and confidence intervals are similar in that they are both predicting a response, however, they differ in what is being represented and interpreted. The best predictor of a random variable (assuming the variable is normally distributed) is the mean $\mu$. The best predictor for an observation from a sample of $x$ data points, $x_1, x_2, \cdots, x_n$ and error $\sigma$ is $\bar{x}$. Since the prediction interval must take into account the variability of the estimators for $\mu$ and $\sigma$, the interval will be wider.

# Confidence Interval of the Estimated $\beta_1$

Confidence intervals can also be formed around the estimated $\beta_1$. The equation for the interval for $\beta_1$ is defined as:

$$\beta_1 \pm t_{\alpha/2,n-2}\left(\frac{\sqrt{\overline{MSE}}}{\sqrt{\sum(x_i-\bar{x})^2}}\right)$$

If the confidence interval for $\beta_1$ contains 0, it can be concluded there is no significant evidence of a linear relationship between predictor $x$ and response $y$ in the population.

The $\beta_1$ interval can be found with the `confint()` function.

```
confint(cars.lm)
```

```
##                 2.5 %    97.5 %
## (Intercept) -31.167850 -3.990340
## speed         3.096964  4.767853
```

The fitted $\beta_1$ is 3.93 with an interval of (3.10, 4.77)

# Calculating Confidence Intervals

The equations written above can be verified with a custom function. The following is a function that replicates the mean, prediction and coefficient intervals found earlier.

```
mean.pred.intervals <- function(x, y, pred.x) {
  n <- length(y) # Find sample size
  lm.model <- lm(y ~ x) # Fit linear model
  y.fitted <- lm.model$fitted.values # Extract the fitted values of y

  # Coefficients of the linear model, beta0 and beta1
  b0 <- lm.model$coefficients[1]
  b1 <- lm.model$coefficients[2]

  pred.y <- b1 * pred.x + b0 # Predict y at the given value of x (argument pred.x)

  # Find SSE and MSE
  sse <- sum((y - y.fitted)^2)
  mse <- sse / (n - 2)

  t.val <- qt(0.975, n - 2) # Critical value of t

  mean.se.fit <- (1 / n + (pred.x - mean(x))^2 / (sum((x - mean(x))^2))) # Standard error
of the mean estimate
  pred.se.fit <- (1 + (1 / n) + (pred.x - mean(x))^2 / (sum((x - mean(x))^2))) # Standard
error of the prediction

  # Mean Estimate Upper and Lower Confidence limits at 95% Confidence
  mean.conf.upper <- pred.y + t.val * sqrt(mse * mean.se.fit)
  mean.conf.lower <- pred.y - t.val * sqrt(mse * mean.se.fit)

  # Prediction Upper and Lower Confidence limits at 95% Confidence
  pred.conf.upper <- pred.y + t.val * sqrt(mse * pred.se.fit)
  pred.conf.lower <- pred.y - t.val * sqrt(mse * pred.se.fit)

  # Beta 1 Upper and Lower Confidence limits at 95% Confidence
  b1.conf.upper <- b1 + t.val * sqrt(mse) / sqrt(sum((x - mean(x))^2))
  b1.conf.lower <- b1 - t.val * sqrt(mse) / sqrt(sum((x - mean(x))^2))

  # Build data.frame of upper and lower limits calculated above, as well as the predicted
y and beta 1 values
  upper <- data.frame(rbind(round(mean.conf.upper, 2), round(pred.conf.upper, 2), round(b
1.conf.upper, 2)))
  lower <- data.frame(rbind(round(mean.conf.lower, 2), round(pred.conf.lower, 2), round(b
1.conf.lower, 2)))
  fit <- data.frame(rbind(round(pred.y, 2), round(pred.y, 2), round(b1, 2)))

  # Collect all into data.frame and rename columns
  results <- data.frame(cbind(lower, upper, fit), row.names = c('Mean', 'Prediction', 'Co
efficient'))
  colnames(results) <- c('Lower', 'Upper', 'Fit')

  return(results)
```

```
}

mean.pred.intervals(cars$speed, cars$dist, new.dat)
```

```
##             Lower  Upper    Fit
## Mean        87.44 113.35 100.39
## Prediction  66.87 133.92 100.39
## Coefficient  3.10   4.77   3.93
```

The output confirms the equations above return the same results as the built-in R functions.

# Confidence Interval around a Linear Regression Line

The gray 'bands' around the regression line in the plot above represent the range in which the true regression line lies at a certain level of confidence (95% in the plot). The bands visualize all intervals for every possible x and are tightest where the data is grouped more densely.

The confidence interval around the regression line is calculated similarly to the other intervals examined above.

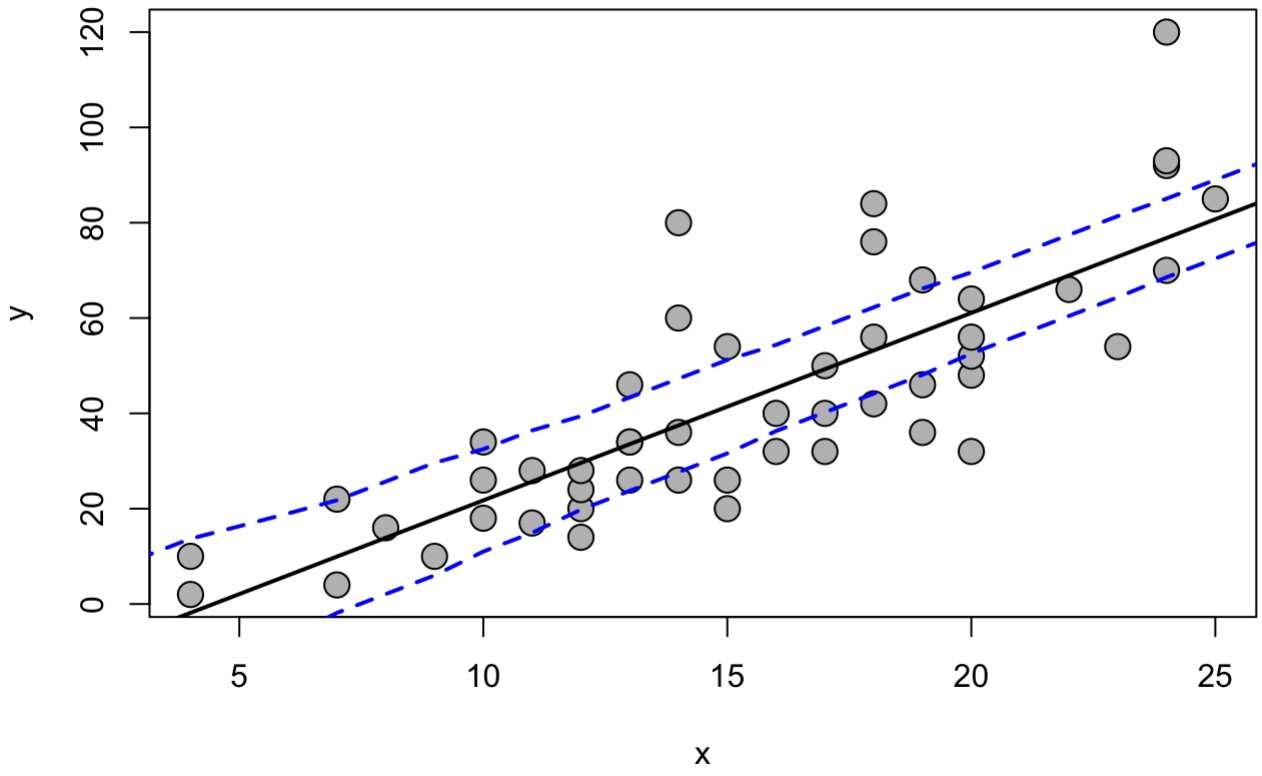
$$\hat{y}_h \pm t_{\alpha/2,n-2}(s.e.)_y$$

Where the $(s.e.)$ is defined as standard error of the regression line multipled by the standard error of the estimate at $x_k$:

$$(s.e.)_y = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y})^2}{n-2}} \sqrt{\frac{1}{n} + \frac{(x-\bar{x})^2}{\sum_{i=1}^{n}(x_i - \bar{x})^2}}$$

The below function fits a linear regression line on one independent variable and finds the confidence interval around the line. The function also outputs a graph with the computed confidence intervals using base R graphics. Obviously, it's not meant to replace the much nicer plot above, but rather to demonstrate an example of how the intervals are calculated and plotted.

```r
reg.conf.intervals <- function(x, y) {
  n <- length(y) # Find length of y to use as sample size
  lm.model <- lm(y ~ x) # Fit linear model

  # Extract fitted coefficients from model object
  b0 <- lm.model$coefficients[1]
  b1 <- lm.model$coefficients[2]

  # Find SSE and MSE
  sse <- sum((y - lm.model$fitted.values)^2)
  mse <- sse / (n - 2)

  t.val <- qt(0.975, n - 2) # Calculate critical t-value

  # Fit linear model with extracted coefficients
  x_new <- 1:max(x)
  y.fit <- b1 * x_new + b0

  # Find the standard error of the regression line
  se <- sqrt(sum((y - y.fit)^2) / (n - 2)) * sqrt(1 / n + (x - mean(x))^2 / sum((x - mean
(x))^2))

  # Fit a new linear model that extends past the given data points (for plotting)
  x_new2 <- 1:max(x + 100)
  y.fit2 <- b1 * x_new2 + b0

  # Warnings of mismatched lengths are suppressed
  slope.upper <- suppressWarnings(y.fit2 + t.val * se)
  slope.lower <- suppressWarnings(y.fit2 - t.val * se)

  # Collect the computed confidence bands into a data.frame and name the colums
  bands <- data.frame(cbind(slope.lower, slope.upper))
  colnames(bands) <- c('Lower Confidence Band', 'Upper Confidence Band')

  # Plot the fitted linear regression line and the computed confidence bands
  plot(x, y, cex = 1.75, pch = 21, bg = 'gray')
  lines(y.fit2, col = 'black', lwd = 2)
  lines(bands[1], col = 'blue', lty = 2, lwd = 2)
  lines(bands[2], col = 'blue', lty = 2, lwd = 2)

  return(bands)
}

conf.intervals <- reg.conf.intervals(cars$speed, cars$dist)
```




With the blue dashed lines representing the confidence interval.

# Summary

This post examined confidence and prediction intervals of predicted values from a linear regression model as well as the model's coefficients. Confidence intervals are often a much more useful representation of the data than a point estimate (such as a prediction) as intervals given more information about the population.

The functions above can also be found as a Gist (https://gist.github.com/aschleg/39ed2e1bba6a4501edca3c7792bd46c8).

# References

3.3 - prediction interval for a new response. (2016). Retrieved July 12, 2016, from Penn State Eberly College of Science, https://onlinecourses.science.psu.edu/stat501/node/274 (https://onlinecourses.science.psu.edu/stat501/node/274)

Understanding shape and calculation of confidence bands in linear regression. (2016). From http://stats.stackexchange.com/questions/101318/understanding-shape-and-calculation-of-confidence-bands-in-linear-regression (http://stats.stackexchange.com/questions/101318/understanding-shape-and-calculation-of-confidence-bands-in-linear-regression)

Confidence interval (2016). In Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Confidence_interval#Statistical_theory (https://en.wikipedia.org/wiki/Confidence_interval#Statistical_theory)