LinearRegression

Raw data

In other to model linear regression, data for 6 bills in restaurant and their tips have been used.

```
##   bill tip
## 1   34   5
## 2  108  17
## 3   64  11
## 4   88   8
## 5   99  14
## 6   51   5
```

Correlation

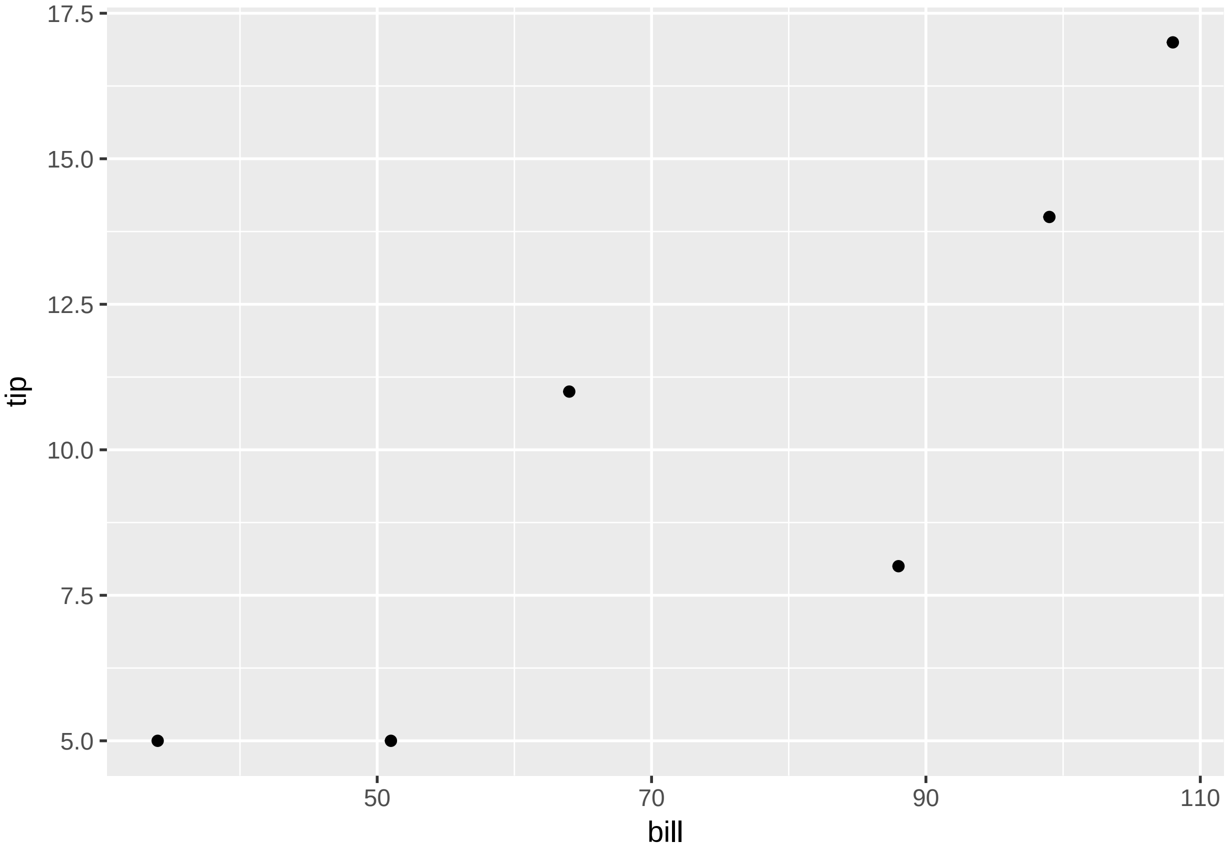
Are variables dependent on each other?

```
my_data %>% summarize(N = n(), r = cor(tip, bill))
```

```
##   N        r
## 1 6 0.865665
```

Correlation coefficient = 0.87; variables are dependent on each other

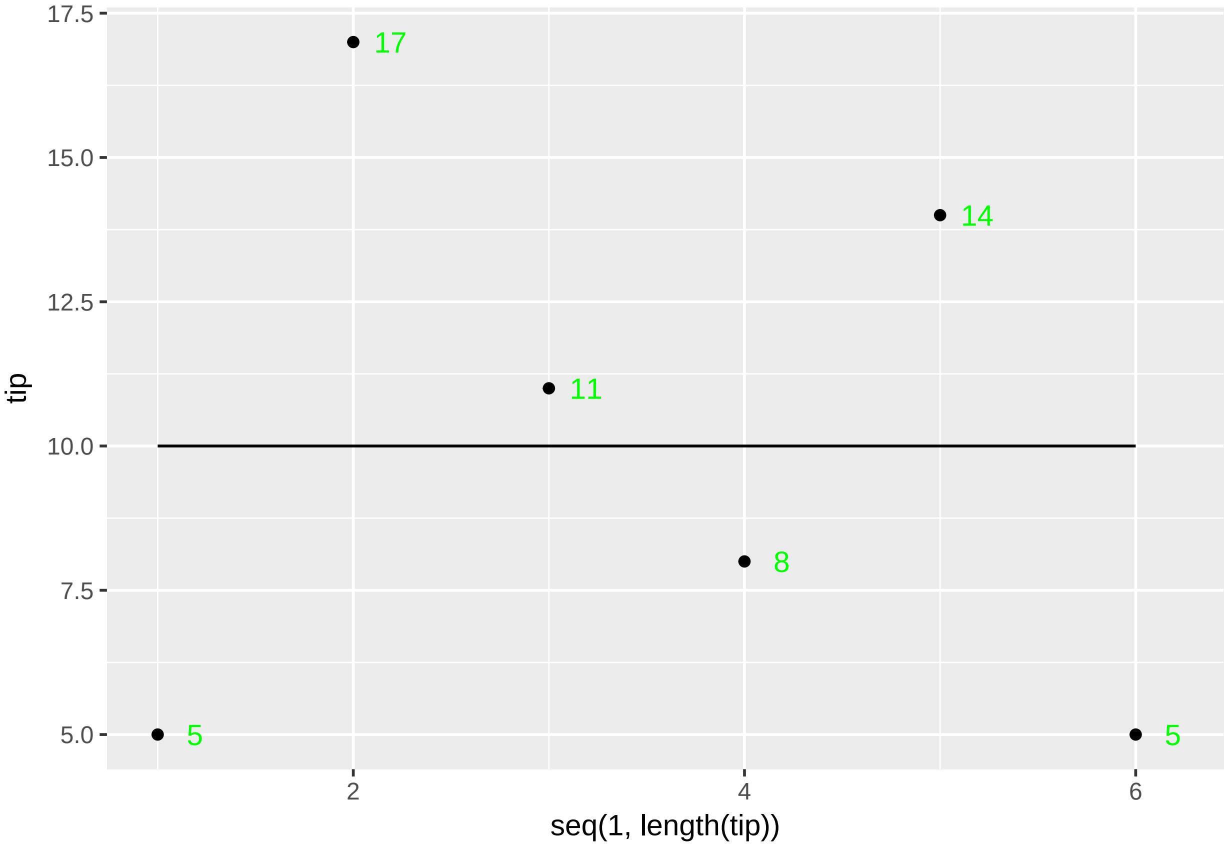
```
ggplot(data = my_data, aes(x = bill, y = tip)) +
  geom_point()
```



Only one variable available

If there is only one variable, the best prediction for other values is the **mean** of the dependant variable.

```
ggplot(my_data, aes(y = tip, x = seq(1, length(tip))))+
  geom_point() +
  geom_line(aes(y=mean(tip)))+
  geom_text(label = my_data$tip, check_overlap=TRUE, nudge_x = 0.19,color="green")
```



SSE of one variable

Simple linear regression is designed to find the best fit line through the data that minimises the SSE (Sum of squared errors).

```
mean_tip <- mean(my_data$tip)
```

```
one_variable_residual <- my_data %>% mutate(residual = tip-mean_tip, residual_squared=(tip-mean_tip)^2)

one_variable_residual
```

```
##   bill tip residual residual_squared
## 1   34   5       -5               25
## 2  108  17        7               49
## 3   64  11        1                1
## 4   88   8       -2                4
## 5   99  14        4               16
## 6   51   5       -5               25
```

```
SSE_one_variable <- sum(one_variable_residual$residual_squared)

SSE_one_variable
```

```
## [1] 120
```

Analysing independent and dependent variable

Centroid

Best fit-line has to go through the centroid of dependent and independent variable.

```
centroid <- c(mean(my_data$bill),mean(my_data$tip))
```

Simple linear regression

Fitting the regression line

$\hat{y} = b_0 + b1_x$

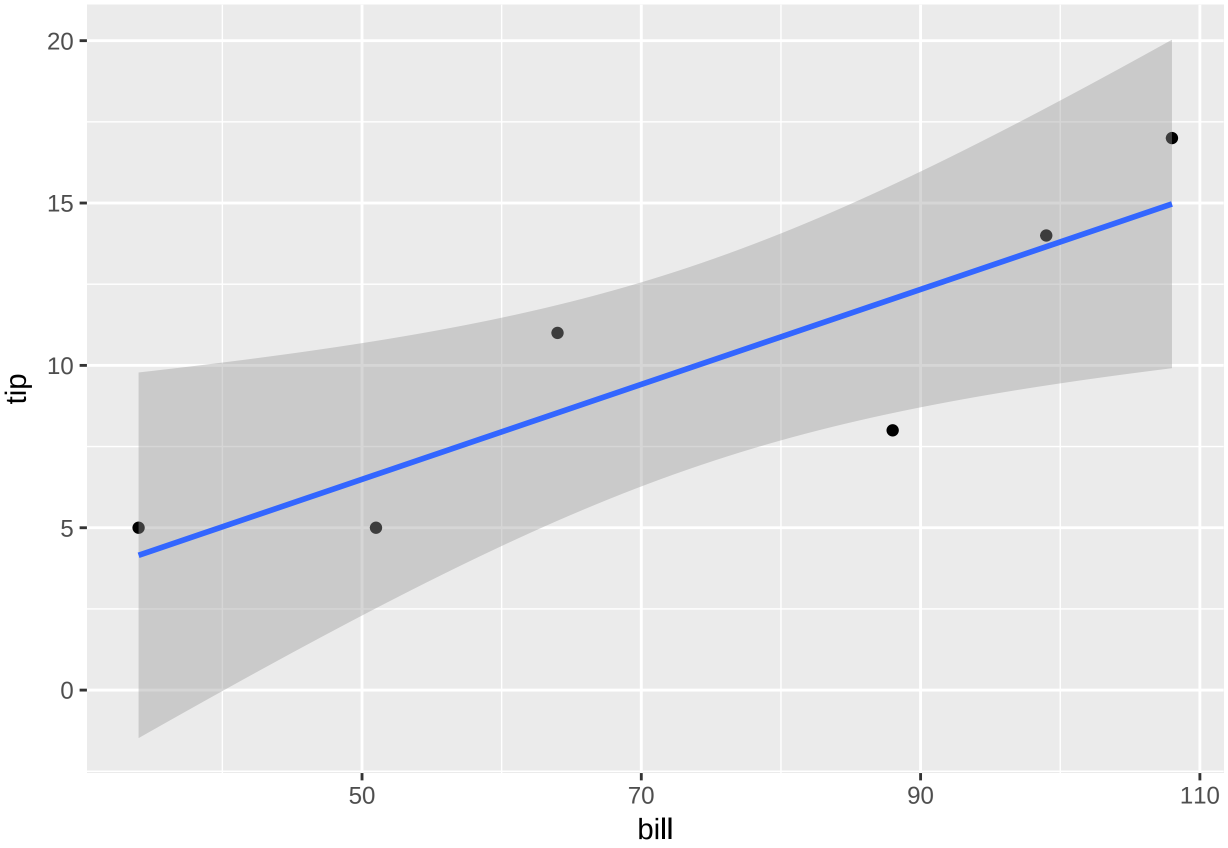
y = dependent variable $\hat{y}$ (y-hat) is point estimator of E(y), it is mean value of y for a given value of X b_0 = y-intercept b_1 = slope of the line X = independent variable

The least square method

min $\sum$ (y_i - $\hat{y}$)^2

Observed value of dependant variable (tip amount) - estimated (predicted) value of the dependant variable (predictd tip amount))

```
ggplot(my_data, aes(x=bill, y=tip))+
  geom_point()+
  geom_smooth(method=lm)
```



```
#blue line is the best-fit line, with 95% confidence intervals in grey indicating a range of values so
```

```
linear_model<-lm(tip~bill, data=my_data)
linear_model
```

```
##
## Call:
## lm(formula = tip ~ bill, data = my_data)
##
## Coefficients:
## (Intercept)         bill
##     -0.8203       0.1462
```

Best-Fit Line: $\hat{y}$ = 0.1462X - 0.8303

Result: For every $1 the bill amount(X) increases, we would expect the tip amount to increase by 0.1462 (about 15 cents).

If the bill amount is 0(x), then the expected/predicted tip amount is -$0.8303 or negative 83 cents. Intercept doesn't have to always make sense.

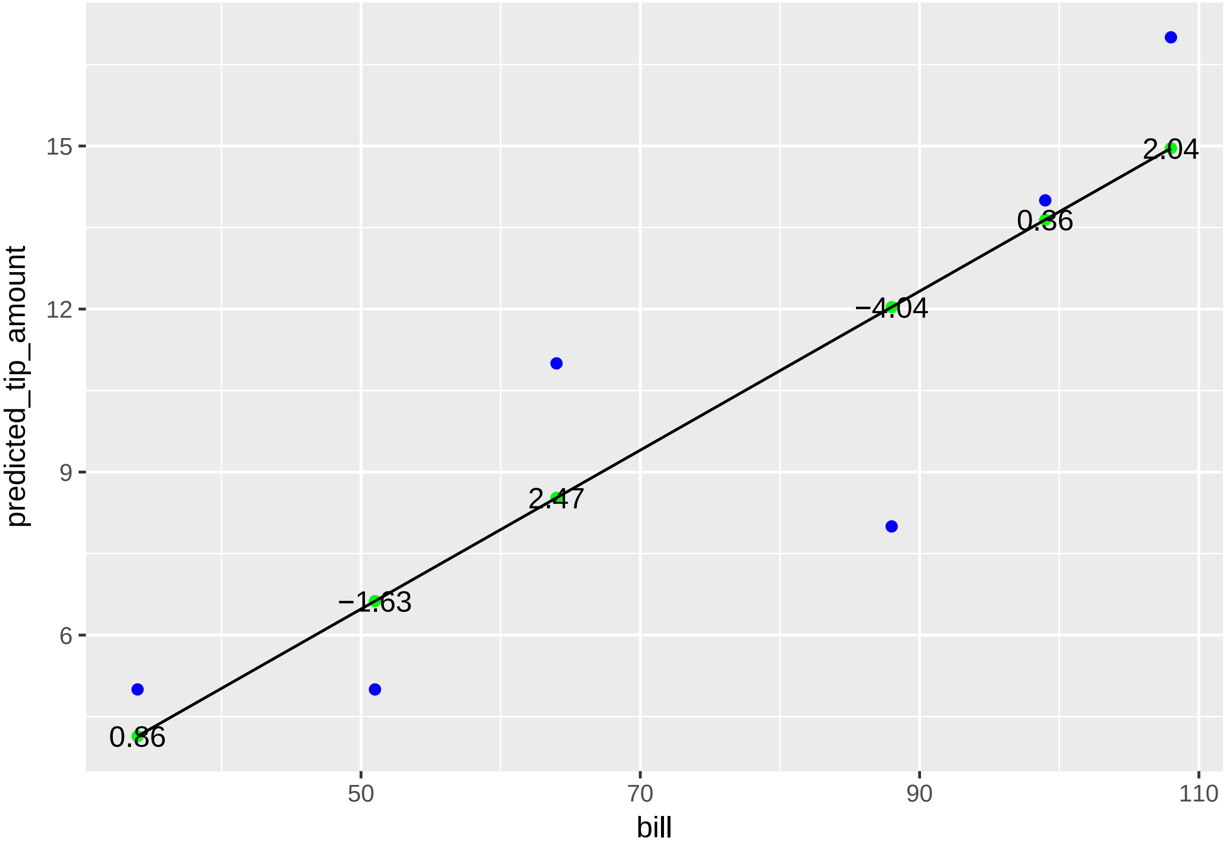
Predicted values of dependent variable

```
my_data_predicted <- my_data%>%mutate(predicted_tip_amount=(0.1462*bill)-0.8303)
```

Plotting difference between observed and predicted values of dependent variable

```
ggplot(my_data_predicted, aes(x=bill, y=predicted_tip_amount))+
  geom_point(aes(y=tip), color='blue',label=my_data_predicted$tip)+
  geom_point(aes(y=predicted_tip_amount), color='green')+
  geom_line(aes(y=predicted_tip_amount))+
  geom_text(label= residual_label)
```

```
## Warning: Ignoring unknown parameters: label
```



### SSE of two variables

```
squared_residual <- residual_label^2

SSE_two_variables <- sum(squared_residual)
SSE_two_variables
```

```
## [1] 30.1102
```

When conducted regression, the SSE decresed from 120 to 30.075. That is, 30.075 of the sum of squares was explained or allocated to error.

SST = SSR + SSE 120 = SSR + 30.075 SSR = 89.925

(Note: SSR = Squared residuals; SST = Total sum of squares)

If SSR is large, it uses up more of SST and therefore SSE is smaller relative to SST. The coefficient of detertmination quantifies this ration as percentage.

Coefficient of determination = Rˆ2 = SSR/SST = 0.7493 = 74.93%

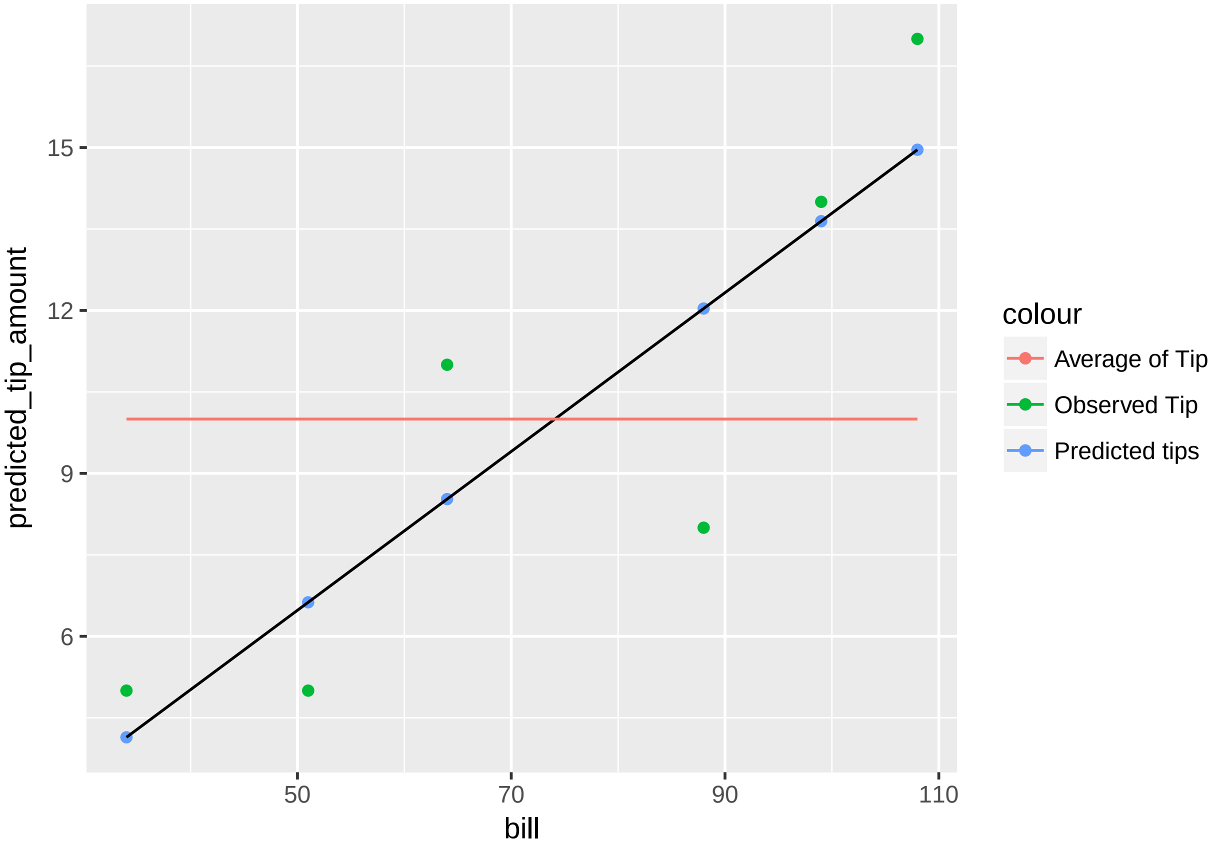
Coefficient of Determination in R

```
linear_statistics<-lm(tip~bill, data=my_data)

coeff_of_determination <- summary(linear_statistics)$r.squared

# The coefficient of determination of the simple linear regression model for the data set is 0.749
```

We can conclude that 74.93% of the total sum of squares can be explained by using the estimated regression equation to predict the tip amount. The remainder is error.

Difference between SSE, SST, SSR

```
colors <- c("Average of Tip" = 'blue', 'Predicted tips' = 'green', 'Observed tips' = 'red')
three_squared_determination <- ggplot(my_data_predicted, aes(x=bill, y=predicted_tip_amount))+
  geom_point(aes(y=tip, color='Observed Tip'))+
  geom_point(aes(y=predicted_tip_amount, color='Predicted tips'))+
  geom_line(aes(y=predicted_tip_amount))+
  geom_line(aes(y=mean(tip), color='Average of Tip'))

three_squared_determination
```



SSE = $\sum$(Observed value - Predicted value)^2 SST = $\sum$(Observed value - Average value)^2 SSR = $\sum$(Predicted value - Average value)^2

Test and confidence interval for the slope

We need to answer 2 questions: a) How much variance in the dependant variable is explained by the model/independent variable? (For this we look at the value of R^2 or adjusted R^2)

b) Does a statistically significant linear relationship exist between the independent and dependant variables?

- Is the overall F-test or t-test (in simple regression these are actually the same thing) significant?
- Can we reject the null hypothesis that the slope b1 of the regression line is zero?

Confidence interval of the slope in R

```r
confint(linear_statistics, level=0.95)
```

```
##                    2.5 %    97.5 %
## (Intercept) -10.04629623 8.4057827
## bill          0.02883093 0.2636084
```

Confidence interval of the slope manually

```r
summary(linear_model)
```

```
##
## Call:
## lm(formula = tip ~ bill, data = my_data)
##
## Residuals:
##       1       2       3       4       5       6
##  0.8488  2.0285  2.4622 -4.0471  0.3445 -1.6369
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.82026    3.32297  -0.247   0.8172
## bill         0.14622    0.04228   3.458   0.0259 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.742 on 4 degrees of freedom
## Multiple R-squared:  0.7494, Adjusted R-squared:  0.6867
## F-statistic: 11.96 on 1 and 4 DF,  p-value: 0.02586
```

```r
# Standard deviation of the slope (Sb1, point estimator for the slope,t-value)
```

a) Confidence interval of the slope:

$b_1 \ \pm T_{\alpha/2} * S_{b1}$

b_1=*point estimator for the slope (0.1462197)*

$T_{\alpha/2}$ = *t-value = 2.776*

S_{b1} = *Standard deviation of the slope*

b) Standard deviation of the slope:

$S_{b1} = \frac{s}{\sum \sqrt{(X_i - \bar{x})^2}}$

s *(standar error)* $= \sqrt{MSE}$

$MSE = \frac{SSE}{(n-2)}$

c) Result:

$S_{b1} = \frac{2.742}{\sqrt{4{,}206}} = 0.04228$

0.1462197 ± 2.776] x 0.04228 (standard deviation of the slope)

(0.02885 ± 0.2636) => confidence interval of the slope

We are 95% confident that the interval (0.02855, 0.2636) contains the true slope of the regression line.

Does the interval contain zero? No!

Rejecting the null hypothesis that the slope is zero.

H0: $\beta_1 = 0$

No significant relationship exists between the two variables.

H1: $\beta_1 \neq 0$

Based on the 95% confidence interval, the slope of confidence interval is not zero.

t= $\frac{b_1}{s_{b1}} = \frac{0.1462197}{0.04228}$ => 3.4584>2.776 => IS SIGNIFICANT

p-value < 5%

Example of confidence interval in R

What is the **average** tip for a meal of $64?

```r
new.dat <- data.frame(bill=64)

predict(linear_statistics, newdata = new.dat, interval = 'confidence')
```

```
##        fit      lwr      upr
## 1 8.537803 5.215473 11.86013
```

```r
#The 95% confidence interval of the mean bill of 64 dollars is between 5.215473 and 11.86013.
```

We expect a tip about 8$ and 54 cents. However, regression is not deterministic! That's why we need to calculate confidence interval for 64$ bill.

With 95% of confidence we can expect a mean tip for $64 bill to be between 5.215473 and 11.86013 interval.

Example of confidence interval - manually

$\hat{y} \pm T_{\alpha/2} S_{\hat{y}} *$

$T_{\alpha/2} = 2.776445$

$\hat{y} = 8.537803$

n = number of observations

$S_{\hat{y}} *$ => Standard deviation of $\hat{y}$ ($\hat{y} = b_0 + b_1 x$)

Standard deviation of y-hat*: $\hat{y} = s(\sqrt{(\frac{1}{n} + \frac{(x*-x)^2}{(\sum(X_i-\bar{x})^2}})$

$\hat{y}$ = 2.742029 x ($\sqrt{\frac{1}{6} + \frac{(64-74)^2)}{4206}}$)

$\hat{y} = 1.196613$

Confidence interval = 8.537803 ± 2.776445 x 1.196613 = (5.215473, 11.8560133)

Example of confidence interval - manually in R

```r
# writting a function:

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

  # 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
  upper <- data.frame(rbind(round(mean.conf.upper, 2), round(pred.conf.upper, 2), round(b1.conf.upper,
  lower <- data.frame(rbind(round(mean.conf.lower, 2), round(pred.conf.lower, 2), round(b1.conf.lower,
  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', 'Coefficient'))
  colnames(results) <- c('Lower', 'Upper', 'Fit')

  return(results)
}
```

```r
mean.pred.intervals(my_data$bill, my_data$tip, new.dat)
```

```
##               Lower Upper  Fit
## Mean           5.22 11.86 8.54
## Prediction   0.23 16.84 8.54
## Coefficient  0.03  0.26 0.15
```

Prediction Interval

Prediction Interval Example

```
new.dat <- data.frame(bill=64)
predict(linear_statistics, newdata = new.dat, interval = 'prediction')
```

```
##        fit       lwr      upr
## 1 8.537803 0.2313561 16.84425
```

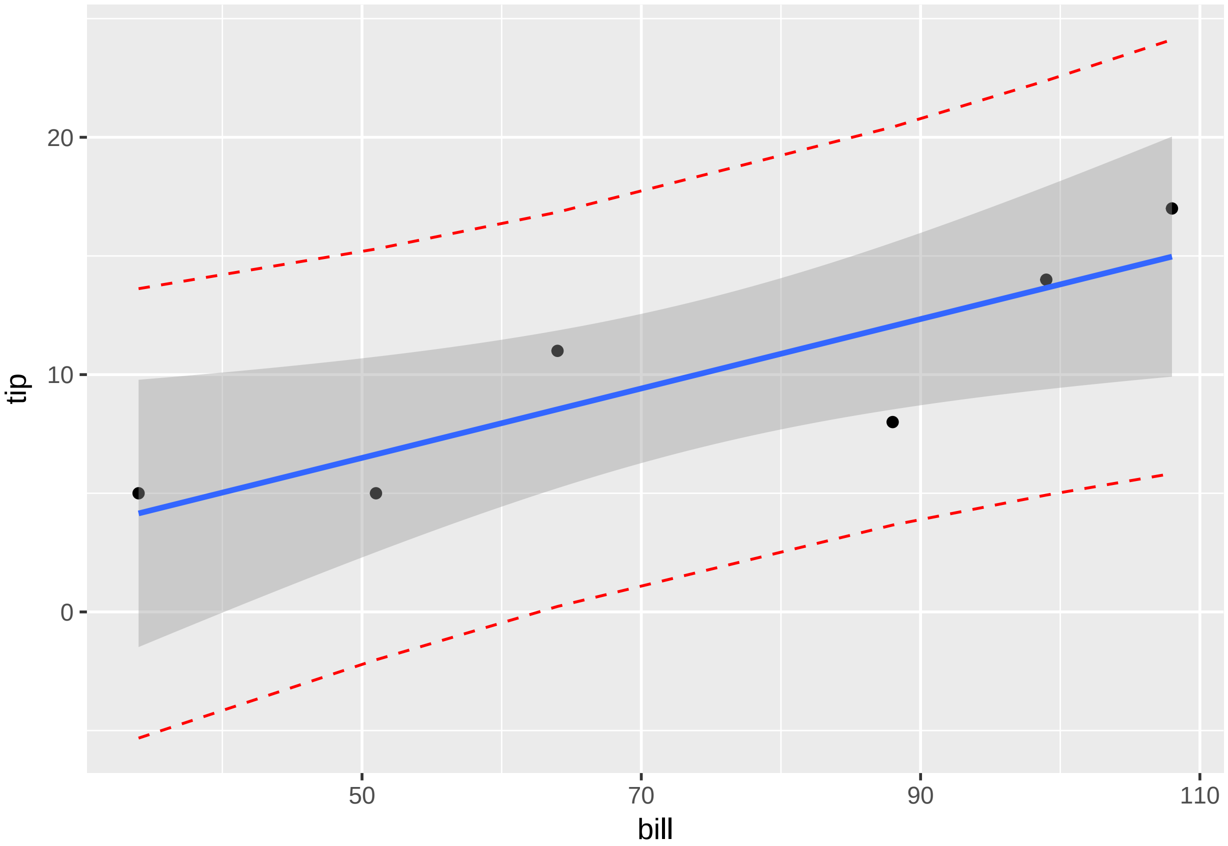
What is the predicted tip for $64 meal? The 95% confidence interval of the tip of bill of 64 dollars is between 0.2313561 and 16.84425.

Prediction interval is for individual value while confidence interval is for the mean value.

Ploting Prediction Interval for Slope

```
slope_prediction_interval <- predict(linear_statistics, interval="prediction")
```

```
plotting_prediction <- cbind(my_data, slope_prediction_interval)

ggplot(plotting_prediction, aes(bill, tip))+
    geom_point() +
    geom_line(aes(y=lwr), color = "red", linetype = "dashed")+
    geom_line(aes(y=upr), color = "red", linetype = "dashed")+
    geom_smooth(method=lm, se=TRUE)
```



Prediction interval will always be wider than confidence interval. The estimated variance (standard deviation) is at its minimum at the mean of the independent variable (x-$\bar{x}$=0), that's why prediction interval is wider than confidence interval. As x gets further from $\bar{x}$, the confidence interval will become wider. This implies to prediction interval as well.

Assumptions

Normal distribution of residuals - QQ Plot or Shapiro-Wilk Test Homoscedasticity - Levene's Test or graphically (residua vs X; residua vs predicted values) Correct Regression Function

Assumption 1

Mean of residuals is close to 0.

Check the mean of the residuals. If it zero (or very close), then this assumption is held true for that model.

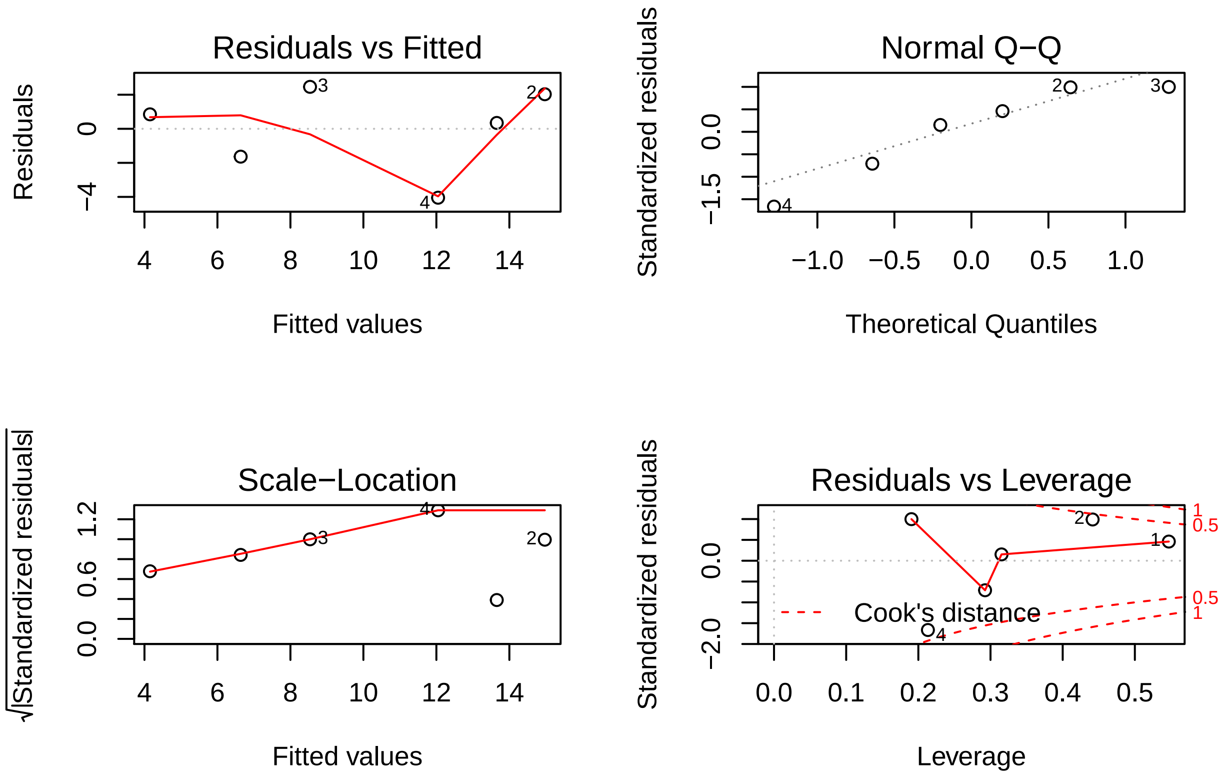
```
mean(linear_model$residuals)
```

```
## [1] -1.846758e-17
```

Assumption 2

Homoscedasticity of residuals or equal variance

```
par(mfrow=c(2,2))  # set 2 rows and 2 column plot layout
mod <- linear_model<-lm(tip~bill, data=my_data)  # linear model
plot(mod)
```



1. Residual vs Fitted Plot known as Residual Plot. Here the X-axis is the predicted or fitted Y values: the Y hats. On Y-axis are residuals or errors. If the linearity assumption is met, we should see no pattern here. The red line should be fairly flat. If the variation is constant here, we should see no pattern.
2. QQ Plot or quantile quantile plot, the Y-axis is the ordered, observed standardized residuals. X-axis is the ordered theoretical residuals. That is what we would expect the residuals to be if the errors or the residuals are truly normally distributed. These points should follow roughly on a diagonal line.

Levene's Test to check heteroscedasticity/homoscedasticity:

```
# median of x
median(my_data$bill)
```

```
## [1] 76
```

```
my_data %>% mutate(group=case_when(bill > 76 ~ "above",bill <= 75 ~ "below")) -> my_data_category

# Levene Test
leveneTest(linear_model$residuals ~ group, my_data_category)
```

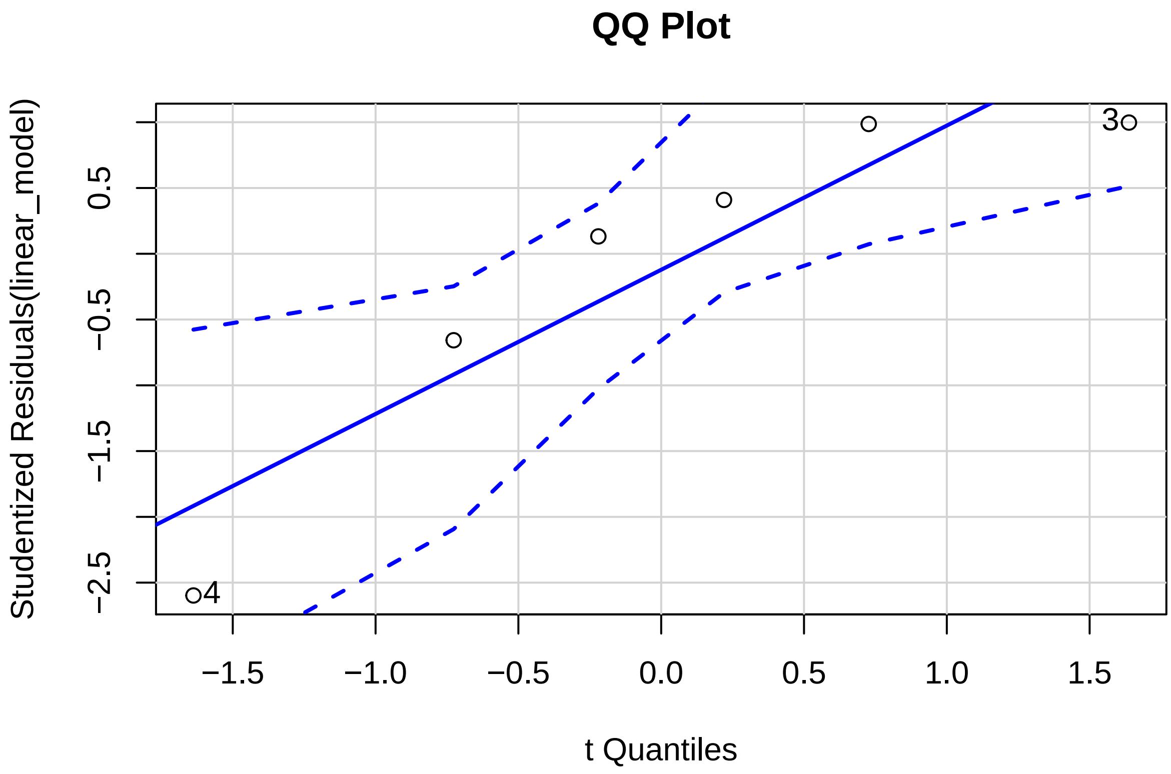
```
## Warning in leveneTest.default(y = y, group = group, ...): group coerced to
## factor.
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  1  0.2004 0.6776
##        4
```

Assumption 3

Normal distribution of residuals

```
qqPlot(linear_model, main="QQ Plot")
```



```
## [1] 3 4
```

```
#Shapiro-Wilk test normality
shapiro.test(linear_model$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  linear_model$residuals
## W = 0.91951, p-value = 0.5018
```

```
#From the output, the p-value > 0.05 implying that the distribution of the data are not significantly d

#The central limit theorem tells us that no matter what distribution things have, the sampling distribu
```