

Tutorial 7 Solutions

Question 1

(11.15) Derive the following identity:

$$\text{SSE} = \sum_{i=1}^n (y_i - \hat{y}_i)^2 = S_{yy} - \hat{\beta}_1 S_{xy}$$

$$\begin{aligned} \text{SSE} &= \sum_{i=1}^n (y_i - \hat{y}_i)^2 \\ &= \sum_{i=1}^n (y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i))^2 \\ &= \sum_{i=1}^n (y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2 \\ &= \sum_{i=1}^n (y_i - (\bar{y} - \hat{\beta}_1 \bar{x}) - \hat{\beta}_1 x_i)^2 \\ &= \sum_{i=1}^n (y_i - \bar{y} + \hat{\beta}_1 \bar{x} - \hat{\beta}_1 x_i)^2 \\ &= \sum_{i=1}^n (y_i - \bar{y} - \hat{\beta}_1 (x_i - \bar{x}))^2 \\ &= \sum_{i=1}^n (y_i - \bar{y})^2 - 2\hat{\beta}_1 \sum_{i=1}^n (y_i - \bar{y})(x_i - \bar{x}) + \hat{\beta}_1^2 \sum_{i=1}^n (x_i - \bar{x})^2 \\ &= S_{yy} - 2 \frac{S_{xy}}{S_{xx}} S_{xy} + \frac{S_{xy}^2}{S_{xx}^2} S_{xx} \\ &= S_{yy} - 2 \frac{S_{xy}^2}{S_{xx}} + \frac{S_{xy}^2}{S_{xx}} \\ &= S_{yy} - \frac{S_{xy}^2}{S_{xx}} \\ &= S_{yy} - \hat{\beta}_1 S_{xy} \end{aligned}$$

For questions 2 and 3, we will be making plots via `ggplot2`. In preparation for this, I will load the package here.

```
library(ggplot2)
```

Question 2

(11.5/11.17a/11.32) The median sale prices for new single family houses are given in the table below for the years 1972 through 1979.

Year	Median Sale Price (Thousands, y)	Year Index (x)
1972	27.6	1
1973	32.5	2
1974	35.9	3
1975	39.3	4
1976	44.2	5
1977	48.8	6
1978	55.7	7
1979	62.9	8

- (a) Letting Y denote the median sales price and x the year (using integers 1, 2, ..., 8), fit the model $Y = \beta_0 + \beta_1 x + \varepsilon$.

We begin by calculating some summary quantities.

$$\bar{x} = 4.5, \quad \sum x_i^2 = 204, \quad \bar{y} = 43.3625, \quad \sum y_i^2 = 16045.29, \quad \sum x_i y_i = 1764.4, \quad n = 8$$

Our least squares estimates are found as:

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}} = \frac{\sum x_i y_i - n\bar{x}\bar{y}}{\sum x_i^2 - n\bar{x}^2} = \frac{1764.4 - 8(4.5)(43.3625)}{204 - 8(4.5)^2} = 4.8417$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} = 43.3625 - 4.84167(4.5) = 21.5750$$

The equation of our fitted line is given by:

$$\widehat{\text{Median Sales Price}} = 21.5750 + 4.8417 * \text{Year Index}$$

- (b) Calculate SSE and S^2 .

The calculation of all the \hat{y}_i values are obtained by plugging in each x_i value into the fitted equation obtained in (a). However, this is a tedious task and is something that we don't want to do if we are approaching this problem by hand. To calculate the SSE, we can use the result obtained in Question 1.

$$\text{SSE} = S_{yy} - \hat{\beta}_1 S_{xy}$$

$$\begin{aligned}
&= \left(\sum y_i^2 - n\bar{y}^2 \right) - \hat{\beta}_1 \frac{\sum x_i y_i - n\bar{x}\bar{y}}{\sum x_i^2 - n\bar{x}^2} \\
&= (16045.29 - 8(43.3625)^2) - 4.8417 \frac{1764.4 - 8(4.5)(43.3625)}{204 - 8(4.5)^2} \\
&= 18.2858
\end{aligned}$$

To calculate S^2 , we use the following formula:

$$S^2 = \frac{\text{SSE}}{n - (p + 1)},$$

where $p + 1$ is the number of coefficients in the model (including the intercept). In the case of simple linear regression with an intercept, $p = 1$, and so $p + 1$ is always equal to 2. As such, we have

$$S^2 = \frac{\text{SSE}}{n - 2}.$$

Plugging in our value obtained for SSE, we have

$$S^2 = \frac{18.2858}{8 - 2} = 3.0476$$

- (c) Is there sufficient evidence to indicate that the median sales price for new single family houses increased over the period from 1972 through 1979? Use $\alpha = 0.01$.

The hypotheses we wish to test are:

$$H_0 : \beta_1 \leq 0 \quad \text{vs} \quad H_1 : \beta_1 > 0.$$

Our test statistic is of the form

$$T = \frac{\hat{\beta}_1 - \beta_{1,0}}{\text{se}(\hat{\beta}_1)} = \frac{\hat{\beta}_1 - 0}{\sqrt{\frac{S^2}{S_{xx}}}} \sim t_{n-(p+1)}.$$

As in (b), in the case of simple linear regression with an intercept, $p = 1$, so that $p + 1 = 2$, so the above statistic has a t -distribution on $n - 2$ degrees of freedom. Plugging in the required values:

$$t = \frac{4.8417 - 0}{\sqrt{\frac{3.0476}{204 - 8(4.5)^2}}} = 17.97.$$

The p -value of this upper-tailed test is:

```
pt(17.97, df=8-2, lower.tail=FALSE)
```

```
## [1] 9.549767e-07
```

Since the p -value is less than $\alpha = 0.01$, we reject the null hypothesis. We conclude that the median sales price of new single family houses has increased over the period of 1972 to 1979.


- (d) Estimate the expected yearly increase in median sale price by constructing a 99% confidence interval.

The 99% confidence interval for β_1 is given by:

$$\hat{\beta}_1 \pm t_{n-(p+1), 0.01/2} \sqrt{\frac{S^2}{S_{xx}}},$$

again with $p + 1 = 2$. Plugging in the required values, we obtain the interval [3.8430, 5.8404].

We conclude with 99% confidence that the expected yearly increase in median sales price of new single family houses over the period of 1972 and 1979 is between 3.8430 and 5.8404 ($\times 1000$) dollars.

- (e)  Repeat parts (a) - (d) using **R**. After fitting the model in (a), create a plot of the residuals against the year index.

We begin by reading the data into **R**.

```
houses <- data.frame(  
  sale_price = c(27.6, 32.5, 35.9, 39.3, 44.2, 48.8, 55.7, 62.9),  
  year_index = 1:8  
)
```

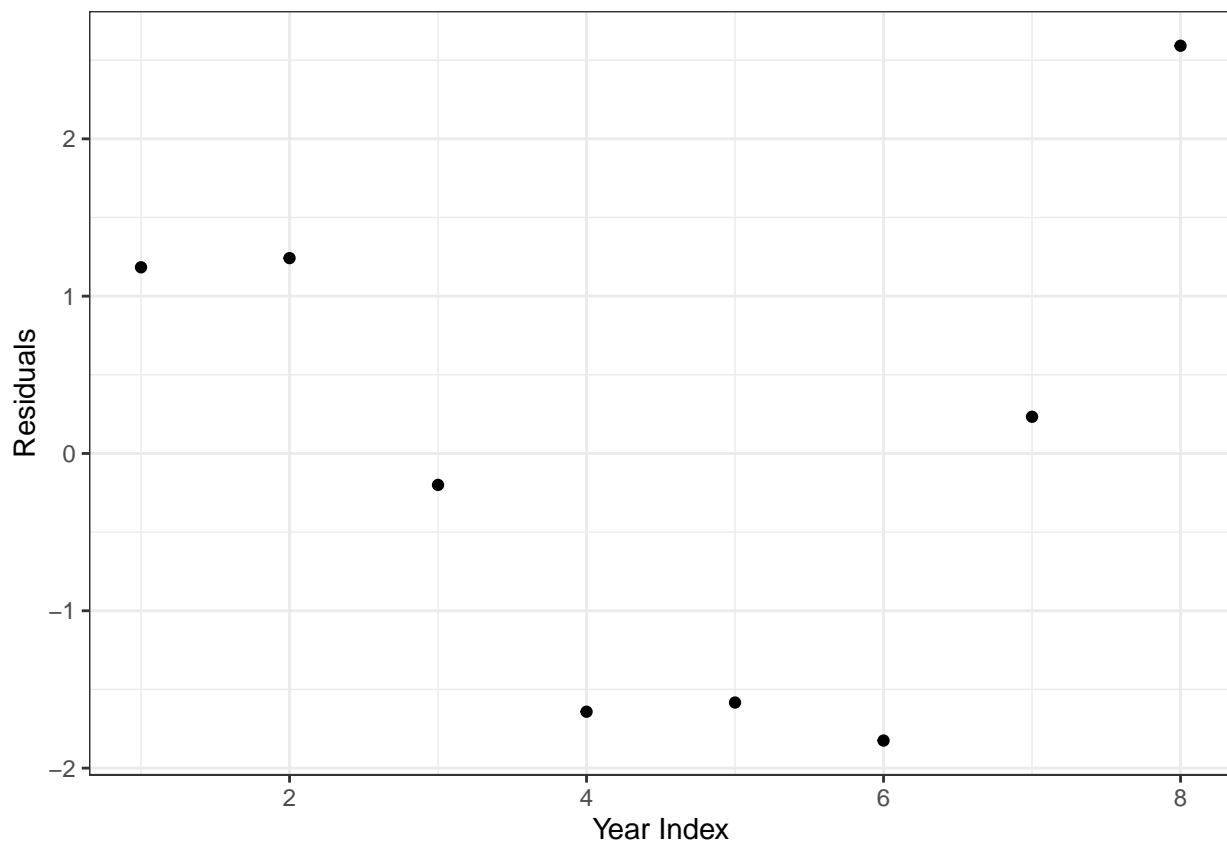
The linear model is fitted as usual.

```
houses_lm <- lm(sale_price ~ year_index, data=houses)  
  
coef(houses_lm)
```

```
## (Intercept)  year_index  
##    21.575000    4.841667
```

In last week's tutorial, I briefly mentioned that while `ggplot()` always requires a data frame to be supplied to its first argument, if a linear model is supplied, it will call `ggplot2::fortify.lm()` under the hood to convert the linear model into a data frame. This data frame that is constructed behind the scenes contains a column called `.resid` which contains the residuals. The plot of the residuals against the year index can be constructed using the following code:

```
ggplot(houses_lm, aes(x=year_index, y=.resid)) +  
  geom_point() +  
  labs(x="Year Index", y="Residuals")
```



We can calculate the SSE using the direct method since we can easily obtain all the fitted values of the model using software.

```
SSE <- sum((houses$sale_price - predict(houses_lm))^2)
```

```
SSE
```

```
## [1] 18.28583
```

S^2 can be obtained by dividing the SSE by its degrees of freedom, $n - (p + 1) = n - 2 = 6$.

```
S_sq <- SSE / 6
```

```
S_sq
```

```
## [1] 3.047639
```

Note that if we called the model summary, near the bottom, a value called **Residual standard error** is given with its degrees of freedom.

```
summary(houses_lm)
```

```
##
```

```
## Call:
```

```
## lm(formula = sale_price ~ year_index, data = houses)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
```

```
## -1.82500 -1.59792  0.01667  1.19792  2.59167
```

```
##
```

```
## Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  21.5750     1.3603   15.86 3.99e-06 ***
## year_index    4.8417     0.2694   17.97 1.91e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.746 on 6 degrees of freedom
## Multiple R-squared:  0.9818, Adjusted R-squared:  0.9787
## F-statistic: 323.1 on 1 and 6 DF, p-value: 1.908e-06
```

This residual standard error is in fact S . As such, if we take the square root of the value we obtained for S^2 , we should get the same value displayed for the residual standard error. However, I don't recommend reading this value off the summary and using it for further calculations as there is a bit of rounding so that the numbers print nicely. As such, your calculations will be more precise if you calculate S^2 yourself. In fact, under the hood, **R** calculates this value of S^2 as I have shown above, before square rooting it to return the value of S .

To perform the hypothesis test of (c), we can return to the model summary again and look under the **t value** column of the coefficient table for the row **year_index**. Recall that the value of the test statistic is the same whether it is an upper-tailed, lower-tailed, or two-tailed hypothesis test. However, we cannot use the p -value included in the table as this corresponds to the two-tailed test of

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

As such, for our upper tailed test, we must find the p -value manually by taking the given value of the test statistic and the degrees of freedom of the residual sum of squares (which is the same as the degrees of freedom of our test statistic).

```
pt(17.97, df=6, lower.tail=FALSE)
```

```
## [1] 9.549767e-07
```

As the p -value of this test is less than $\alpha = 0.01$, we reject the null hypothesis in favour of the alternative and make the same concluding remarks as before.

To obtain confidence intervals for the model coefficients, we can simply wrap our model in the **confint()** function. Note that by default, **confint()** returns 95% confidence intervals. If we seek a 99% confidence interval, we also need to specify **level=0.99**.

```
confint(houses_lm, level=0.99)
```

```
##           0.5 %      99.5 %
## (Intercept) 16.531873 26.618127
## year_index   3.842979  5.840355
```

Our interval is given by [3.8430, 5.8404] and we make the appropriate concluding remarks as above.

Question 3

(11.16/11.39/11.46) An experiment was conducted to observe the effect of an increase in temperature on the potency of an antibiotic. Three 1-ounce portions of the antibiotic were stored for equal lengths of time at various Fahrenheit temperatures.

Potency Reading (y)	Temperature (x)
38	30
43	30
29	30
32	50
26	50
33	50
19	70
27	70
23	70
14	90
19	90
21	90

(a) Find the least-squares line appropriate for this data.

We begin by calculating some summary quantities.

$$\bar{x} = 60, \quad \sum x_i^2 = 49200, \quad \bar{y} = 27, \quad \sum y_i^2 = 9540, \quad \sum x_i y_i = 17540, \quad n = 12$$

Our least squares estimates are found as:

$$\hat{\beta}_1 = \frac{S_{xy}}{S_{xx}} = \frac{\sum x_i y_i - n\bar{x}\bar{y}}{\sum x_i^2 - n\bar{x}^2} = \frac{17540 - 12(60)(27)}{49200 - 12(60)^2} = -0.3167$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x} = 27 - (-0.3167)(60) = 46.0000$$

The equation of our fitted line is given by:

$$\widehat{\text{Potency}} = 46.0000 - 0.3167 * \text{Temperature}$$

(b) Calculate S^2 .

We start by calculating the SSE using the result from Question 1.

$$\begin{aligned}
\text{SSE} &= S_{yy} - \hat{\beta}_1 S_{xy} \\
&= \left(\sum y_i^2 - n\bar{y}^2 \right) - \hat{\beta}_1 \frac{\sum x_i y_i - n\bar{x}\bar{y}}{\sum x_i^2 - n\bar{x}^2} \\
&= (9540 - 12(27)^2) - (-0.3167) \frac{17540 - 12(60)(27)}{49200 - 12(60)^2} \\
&= 190.3333
\end{aligned}$$

As before, in the case of simple linear regression with an intercept, the formula for S^2 reduces to:

$$S^2 = \frac{\text{SSE}}{n - 2}.$$

Plugging in our value obtained for SSE, we have

$$S^2 = \frac{190.3333}{12 - 2} = 19.0333$$

- (c) Find a 95% confidence interval for the mean potency of a 1-ounce portion of antibiotic stored at 65° F.

In the setting of a simple linear regression model with an intercept, the formula for a 95% confidence interval for the mean response at a particular value of the predictor is given by:

$$\hat{y}_* \pm t_{n-2, 0.05/2} \sqrt{S^2 \left(\frac{1}{n} + \frac{(x_* - \bar{x})^2}{S_{xx}} \right)},$$

where $\hat{y}_* = \hat{\beta}_0 + \hat{\beta}_1 x_*$, and the degrees of freedom of the critical value arises from the degrees of freedom of the SSE. Plugging in the required values, we obtain the interval [22.5412, 28.2921].

We conclude with 95% confidence that for a 1-ounce portion of antibiotic stored at 65° F, the mean potency is between 22.5412 and 28.2921.

- (d) Find a 95% prediction interval for the potency of a 1-ounce portion of antibiotic stored at 65° F. How does this interval compare to the interval found in (c)?


In the setting of a simple linear regression model with an intercept, the formula for a 95% prediction interval for the response at a particular value of the predictor is given by:

$$\hat{y}_* \pm t_{n-2, 0.05/2} \sqrt{S^2 \left(1 + \frac{1}{n} + \frac{(x_* - \bar{x})^2}{S_{xx}} \right)},$$

where $\hat{y}_* = \hat{\beta}_0 + \hat{\beta}_1 x_*$, and the degrees of freedom of the critical value arises from the degrees of freedom of the SSE. Plugging in the required values, we obtain the interval [15.2796, 35.5538].

We conclude with 95% confidence that for a 1-ounce portion of antibiotic stored at 65° F, the potency is between 15.2796 and 35.5538.

Comparing this interval to the interval found in (c), we notice that it is much wider. This is due to the presence of the extra term under the square root of the standard error of our prediction.

- (e)  Repeat parts (a) - (d) using **R**. After fitting the model in (a), create a plot of the data points and the fitted line to verify that the fit is appropriate.

We begin by reading the data into **R**.

```
antibiotic <- data.frame(
  potency = c(38, 43, 29, 32, 26, 33, 19, 27, 23, 14, 19, 21),
  temperature = rep(c(30, 50, 70, 90), each=3)
)
```

The linear model is fitted as usual.

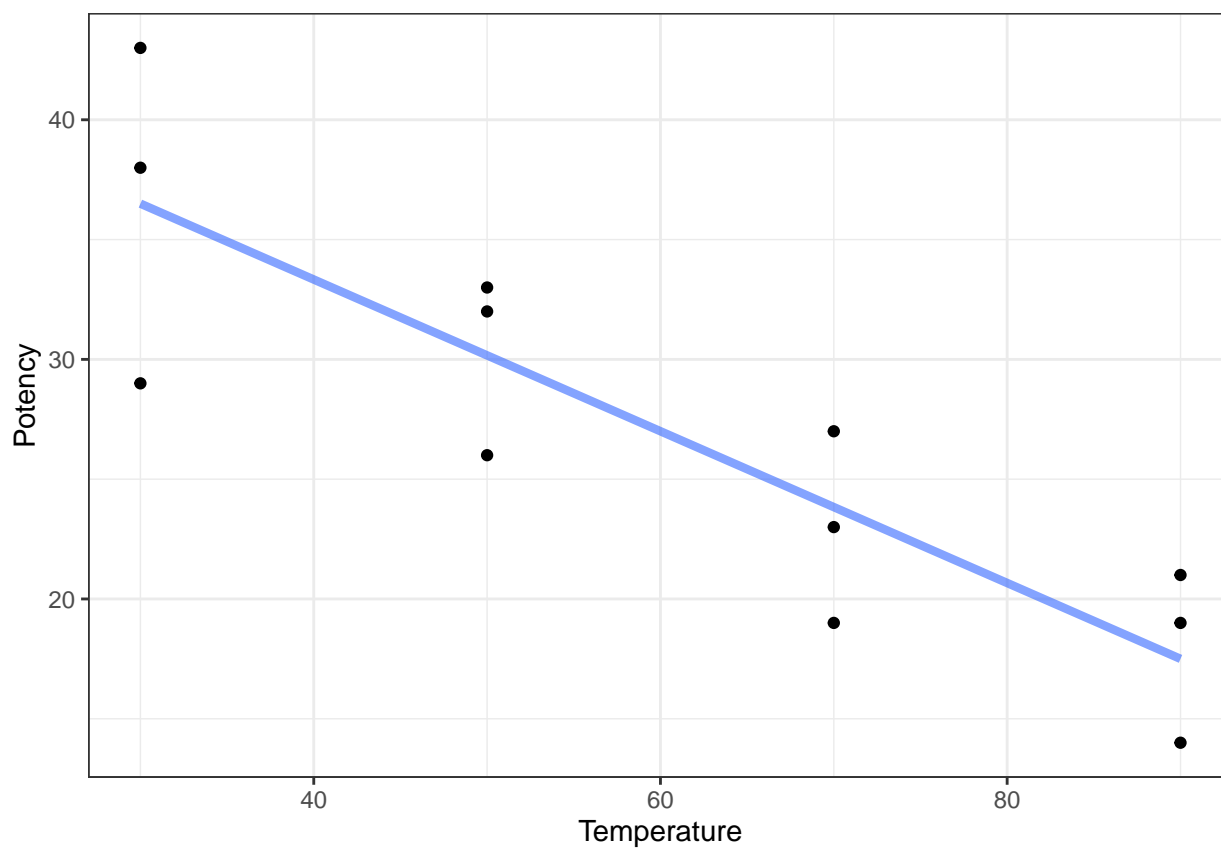

```
antibiotic_lm <- lm(potency ~ temperature, data=antibiotic)

coef(antibiotic_lm)
```

```
## (Intercept) temperature
## 46.0000000 -0.3166667
```

We can use the code below to create a plot of the data points with the fitted line.

```
ggplot(antibiotic_lm, aes(x=temperature)) +
  geom_point(aes(y=potency)) +
  geom_line(aes(y=.fitted), colour="#3366FF", size=1.5, alpha=0.6) +
  labs(x="Temperature", y="Potency")
```



The fit seems appropriate as it is able to capture the overall trend of our data.

To calculate S^2 , we begin by calculating the SSE.

```
SSE <- sum((antibiotic$potency - predict(antibiotic_lm))^2)
```

```
SSE
```

```
## [1] 190.3333
```

To obtain S^2 , we divide the SSE by its degrees of freedom. In the setting of simple linear regression with an intercept, this is $n - 2 = 12 - 2 = 10$.

```
S_sq <- SSE / 10
```

```
S_sq
```

```
## [1] 19.03333
```

We can double check our calculation by taking the square root and comparing it to the value shown in the model summary next to **Residual standard error**.

```
summary(antibiotic_lm)
```

```
##  
## Call:  
## lm(formula = potency ~ temperature, data = antibiotic)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max   
## -7.500 -3.667  1.500   2.917   6.500     
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)      
## (Intercept)  46.00000     3.60640   12.755 1.64e-07 ***  
## temperature  -0.31667     0.05632   -5.622 0.000221 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 4.363 on 10 degrees of freedom  
## Multiple R-squared:  0.7597, Adjusted R-squared:  0.7356   
## F-statistic: 31.61 on 1 and 10 DF,  p-value: 0.0002207
```

```
sqrt(S_sq)
```

```
## [1] 4.362721
```

In addition to returning single point estimates from our model, the `predict()` function is also able to produce confidence intervals and prediction intervals. To see the documentation for predictions with linear models, call `?predict.lm` in the console.

To compute a 95% confidence interval for the mean potency when temperature is 65° F, we predict at a point as usual by supplying a data frame that contains columns with the same names as the predictors in the model, while also specifying `interval="confidence"`.

```
predict(antibiotic_lm, newdata=data.frame(temperature = 65), interval="confidence")
```

```
##      fit      lwr      upr  
## 1 25.41667 22.54123 28.2921
```

Unsurprisingly, to make a *prediction* interval, we instead specify `interval="prediction"`.

```
predict(antibiotic_lm, newdata=data.frame(temperature = 65), interval="prediction")
```

```
##      fit      lwr      upr  
## 1 25.41667 15.27955 35.55378
```

From the documentation, we should note that the confidence level has a default value of 0.95. As such, if we required a 99% confidence/prediction interval, we would also pass in `level=0.99`, as demonstrated below.

```
predict(  
  antibiotic_lm,  
  newdata = data.frame(temperature = 65),  
  level = 0.99)
```

```
interval = "prediction",  
level = 0.99  
)
```

```
##          fit      lwr      upr  
## 1 25.41667 10.99778 39.83555
```