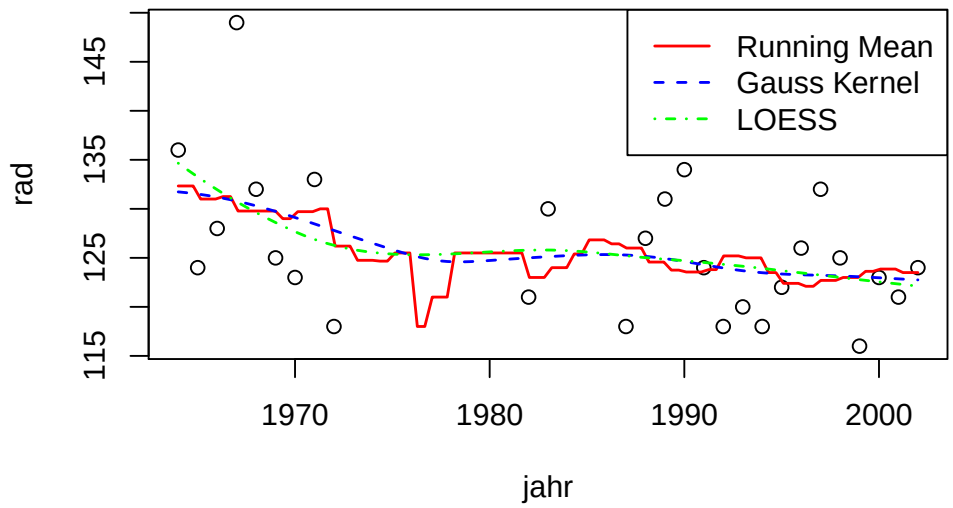
Solution to Series 2

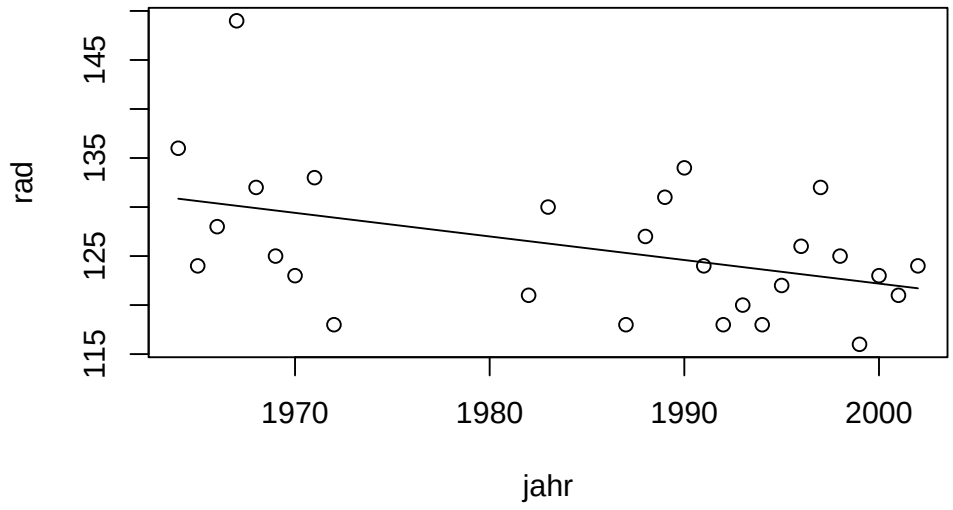
1. **a)**
```
> # Load data
> load("solar.radiation.rda")
> # Ignore corrupted data points
> sol.rad[sol.rad==99999] <- NA
> sol.rad <- na.omit(sol.rad)
> # Scatter plot
> plot(sol.rad)
> # Running Mean
> lines(ksmooth(sol.rad$jahr, sol.rad$rad, kernel="box", bandwidth=10), lwd=1.5,
        col="red")
> # Gaussian Kernel Smoother
> lines(ksmooth(sol.rad$jahr, sol.rad$rad, kernel="normal", bandwidth=10), lty=2,
        lwd=1.5, col="blue")
> # LOESS
> fit <- loess(rad~jahr, sol.rad)
> x <- seq(1964, 2002, length.out=100)
> y <- predict(fit, newdata=data.frame(jahr=x))
> lines(x, y, lty=4, lwd=1.5, col="green")
> # Add legend
> legend("topright", lwd=1.5, lty=c(1,2,4), col=c("red", "blue", "green"),
         legend=c("Running Mean", "Gauss Kernel", "LOESS"))
```



b) Visually, it seems there is a slight decrease in both clusters of the data (60s/70s and after 1980). However, it is not possible to give quantitative evidence to this claim just by using non-parametric smoothing.

c)
```
> # Plot scatter plot and regression line
> plot(sol.rad)
> fit.lm <- lm(rad~jahr, sol.rad)
> lines(sol.rad$jahr, fit.lm$fitted.values)
> # Print fit summary
> summary(fit.lm)
```

```
Call:
lm(formula = rad ~ jahr, data = sol.rad)

Residuals:
     Min       1Q   Median       3Q      Max
-10.9251  -5.5769  -0.3553   3.2839  18.8724

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 603.21788  196.46192    3.07   0.0051 **
jahr         -0.24051    0.09898   -2.43   0.0226 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.608 on 25 degrees of freedom
Multiple R-squared: 0.191,        Adjusted R-squared: 0.1587
F-statistic: 5.904 on 1 and 25 DF,  p-value: 0.02262
```

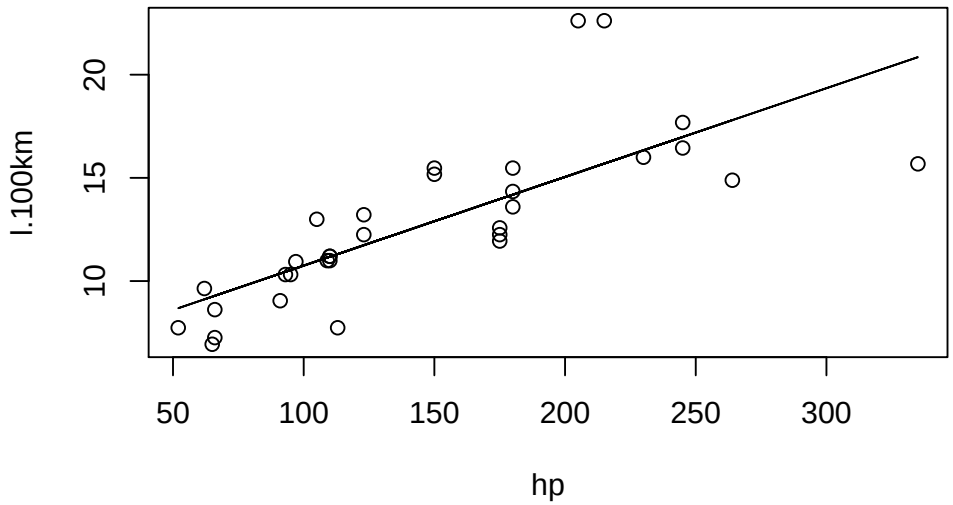


Assuming all the conditions of the OLS regression are correct here, there is considerable quantitative evidence for the claim. The slope parameter indicates a negative trend and is significant on the 5% level.

2. **a)**

```
> # Load data and create scatter plot
> load("my.mtcars.rda")
> plot(l.100km ~ hp, my.mtcars)
> # Fit linear regression and plot
> fit <- lm(l.100km~ hp, my.mtcars)
> lines(my.mtcars$hp, fit$fitted.values)
> # Print fit summary
> summary(fit)

Call:
lm(formula = l.100km ~ hp, data = my.mtcars)

Residuals:
    Min      1Q  Median      3Q     Max
-5.1694 -1.3342 -0.1650  0.5701  7.3550

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.44908    1.07380   6.006 1.37e-06 ***
hp           0.04299    0.00665   6.464 3.84e-07 ***
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.539 on 30 degrees of freedom
Multiple R-squared: 0.5821,        Adjusted R-squared: 0.5682
F-statistic: 41.79 on 1 and 30 DF,  p-value: 3.839e-07
```



b) The residual standard error is 2.54 (from summary output).

c) For the first question we can just use the `predict` function:

```
> # Predict
> predicted.consumption <- predict(fit, newdata=data.frame(hp=100))
> # Print
> names(predicted.consumption) <- NULL
> print(predicted.consumption)
[1] 10.74799
```

So the predicted fuel consumption is 10.75.
For the second part we need to invert the model equation and plug in the values from the summary output ourselves.

```
> # Store regression coefficients
> beta0 <- fit$coefficients[1]
> beta1 <- fit$coefficients[2]
> # Calculate predicted value
> predicted.hp <- (15 - beta0)/beta1
> # Print
> names(predicted.hp) <- NULL
> print(predicted.hp)
[1] 198.9092
```

So the predicted engine power is 198.91.

d) We can just calculate the confidence interval for the slope parameter β_1 and see whether it includes the value 0.05.

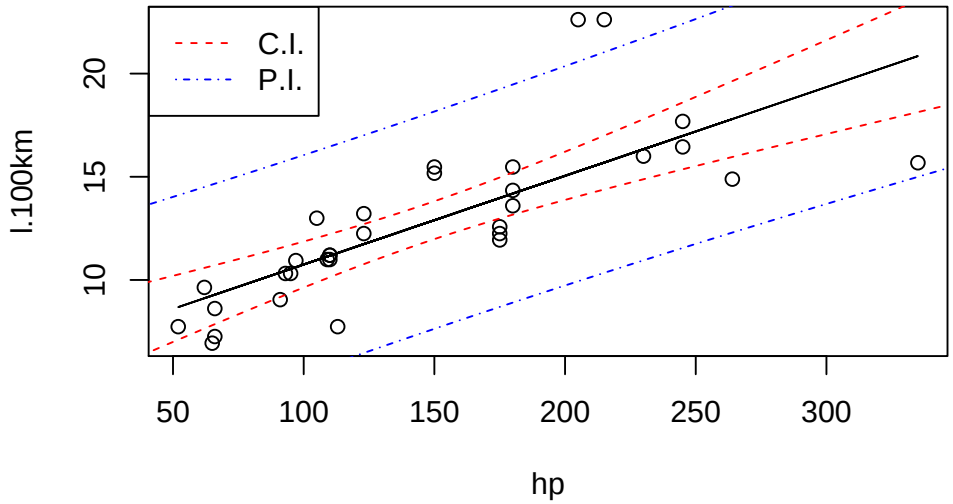
```
> confint(fit, "hp")
        2.5 %     97.5 %
hp 0.02940723 0.05657087
```

It does include 0.05, so the claim cannot be disproved by the data.

e)

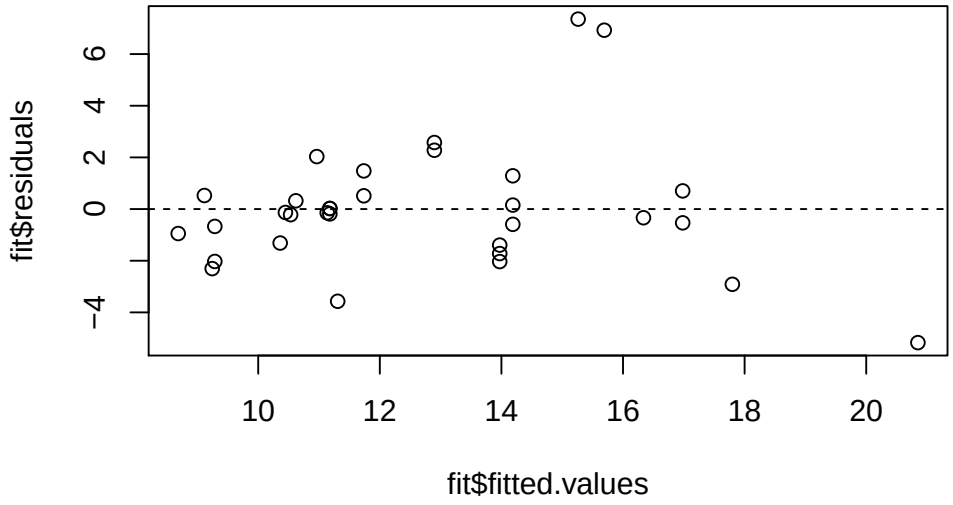
```
> # Draw scatter plot and regression line
> plot(l.100km ~ hp, my.mtcars)
> lines(my.mtcars$hp, fit$fitted.values)
> # Grid with x-values
> newdata <- data.frame(hp=0:400)
```

```
> # Generate and plot confidence interval
> ci <- predict(fit, newdata=newdata, interval="confidence")
> lines(newdata$hp, ci[,2], col="red", lty=2)
> lines(newdata$hp, ci[,3], col="red", lty=2)
> # Generate and plot confidence interval
> pi <- predict(fit, newdata=newdata, interval="prediction")
> lines(newdata$hp, pi[,2], col="blue", lty=4)
> lines(newdata$hp, pi[,3], col="blue", lty=4)
> legend("topleft", lty=c(2, 4), col=c("red", "blue"), legend=c("C.I.", "P.I."))
```



f) We first check for constant variance by plotting the residuals against fitted values (Tukey-Anscombe plot). In this plot we can also see whether the zero expectation assumption is valid.

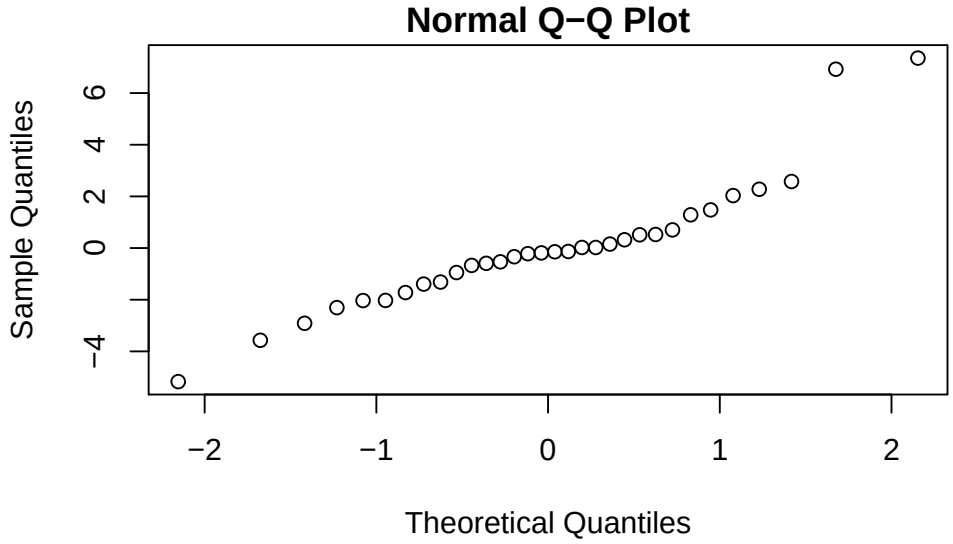
```
> plot(fit$fitted.values, fit$residuals)
> abline(0, 0, lty=2)  # Dashed line at zero
```



The first thing to note is that there seem to be two outliers with very large residuals. Disregarding these, the mean of the residuals is negative, so the zero expectation assumption seems to be violated. The constant variance assumption seems to be fine (without the outliers).

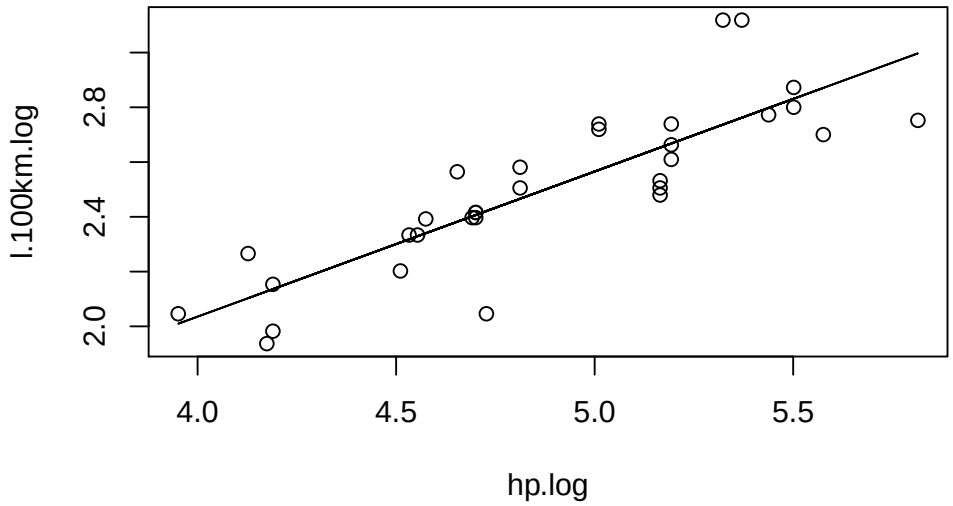
We now look at a QQ-Plot to check for normality or the errors.

```
> qqnorm(fit$residuals)
```



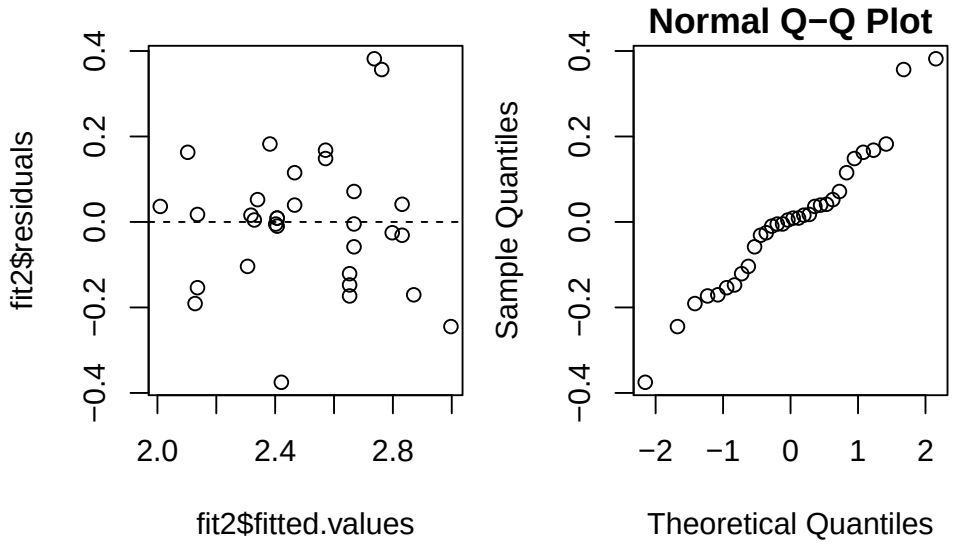
Again, we see the two outliers, which heavily distort the QQ plot. In summary, the model does not seem appropriate for the data. To the very least, the zero expectation and normality assumptions are violated.

g)
```
> # Transform data
> my.mtcars.log <- data.frame(hp.log=log(my.mtcars$hp),
                              l.100km.log=log(my.mtcars$l.100km))
> # Fit linear regression and plot
> fit2 <- lm(l.100km.log ~ hp.log, my.mtcars.log)
> plot(l.100km.log ~ hp.log, my.mtcars.log)
> lines(my.mtcars.log$hp.log, fit2$fitted.values)
> # Print fit summary
> summary(fit2)

Call:
lm(formula = l.100km.log ~ hp.log, data = my.mtcars.log)

Residuals:
     Min       1Q   Median       3Q      Max
-0.37501 -0.10815  0.00691  0.05707  0.38189

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.08488    0.29913  -0.284    0.779
hp.log       0.53009    0.06099   8.691 1.08e-09 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1614 on 30 degrees of freedom
Multiple R-squared: 0.7157,        Adjusted R-squared: 0.7062
F-statistic: 75.53 on 1 and 30 DF,  p-value: 1.08e-09
```



We see immediately from the plot that the model fits the data better. Looking at the residuals confirms this first impression:

```
> par(mfrow=c(1,2))
> plot(fit2$fitted.values, fit2$residuals)
> abline(0, 0, lty=2)
> qqnorm(fit2$residuals)
```

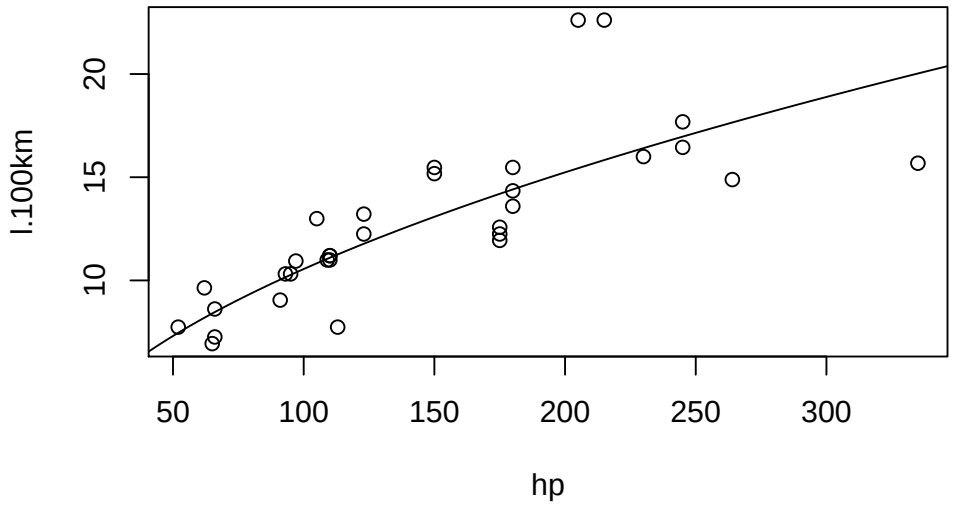


h) Exponentiating yields:

$$\texttt{l.100km} = \exp(\beta_0) \cdot \texttt{hp}^{\beta_1} \cdot \exp(\epsilon)$$

I.e. the relation is not linear any more, it is a power law in hp. Also, the error now is multiplicative and follows a log-Normal distribution.

i)
```
> # Scatter plot
> plot(l.100km ~ hp, my.mtcars)
> # Log-model curve
> newdata.log <- data.frame(hp.log=seq(3,6,length.out=200))
> y.pred <- predict(fit2, newdata=newdata.log)
> lines(exp(newdata.log$hp.log), exp(y.pred))
```



3. **a)** The scatterplot shows a curved relation.

b) N_t is the number of surviving bacteria upt to the time point t, hence N_0 is the starting population. In each interval only a constant proportion b of bacteria survives, where $0 < b < 1$.
Therefore it follows that

at time point $t = 1$	$N_1 = b \cdot N_0$ bacteria
at time point $t = 2$	$N_2 = b \cdot N_1 = b^2 \cdot N_0$ bacteria
$\vdots$	$\vdots$
at time point $t = i$	$N_i = b \cdot N_{i-1} = \ldots = b^i \cdot N_0$ bacteria

$$N_i = b^i \cdot N_0 \Longleftrightarrow \log(N_i) = i \cdot \log(b) + \log(N_0)$$

$$\Longleftrightarrow \underbrace{\log(N_i)}_{y} = \underbrace{\log(N_0)}_{\beta_0} + \underbrace{\log(b)}_{\beta_1} \cdot \underbrace{i}_{x}$$

The scatterplot of $\log(N_t)$ versus t exhibits a tolerably linear relation.

c) Regression equation $\hat{y} = 5.973 - 0.218x$
Starting population: $\hat{N_0} = e^{5.97316} = 393$
Percentaged decrease: $1 - \hat{b} = 1 - e^{-0.218} = 0.20$

4. **a)** The gas consumption is quite constant if the temperature difference is smaller than 14 ℃, only if it gets larger the consumption increases. The spread is rather large, which is not surprising since the measurements were performed on different houses.

b)
```
> mod1 <- lm(verbrauch~temp,data=gas)
> mod1

Call:
lm(formula = verbrauch ~ temp, data = gas)

Coefficients:
(Intercept)         temp
     36.894        3.413

> summary(mod1)

Call:
lm(formula = verbrauch ~ temp, data = gas)

Residuals:
    Min      1Q  Median      3Q     Max
```

```
-13.497  -7.391  -2.235   6.280  17.367

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   36.894     16.961   2.175   0.0487 *
temp           3.413      1.177   2.900   0.0124 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.601 on 13 degrees of freedom
Multiple R-squared: 0.3929,        Adjusted R-squared: 0.3462
F-statistic: 8.413 on 1 and 13 DF,  p-value: 0.0124
```

c) The residual plots do not look satisfying, but transformation (log, $\sqrt{}$) or a quadratic term seem not to be helpful either.

d) $\hat{y} = 36.8937 + 3.4127 \cdot 14 = 84.67$

```
> new.x <- data.frame(temp=14)
> predict(mod1,new.x)
       1
84.67202
> predict(mod1,new.x,interval="confidence")
       fit      lwr      upr
1 84.67202 79.27618 90.06787
```