Solution to Series 2

1. 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: $1 - \hat{b} = 1 - e^{-0.218} = 0.20$

2. a) R code:

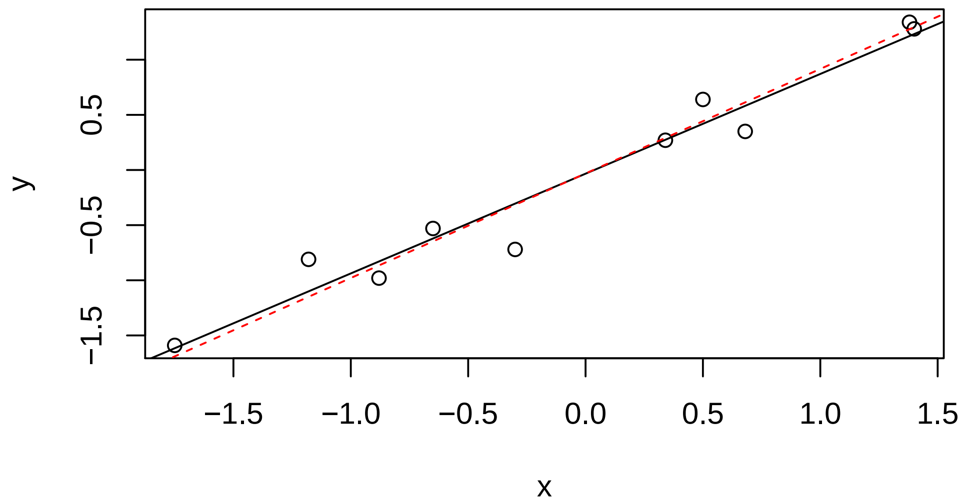
```
> x<-c(0.34,1.38,-0.65,0.68,1.40,-0.88,-0.30,-1.18,0.50,-1.75)
> y<-c(0.27,1.34,-0.53,0.35,1.28,-0.98,-0.72,-0.81,0.64,-1.59)
> plot(x,y)
```

b) R code:

```
> mod1 <- lm(y~x)
> abline(mod1)
```

c) The command `abline(mod)` draws a straight line with the slope and the axis intercept defined in the object `mod`. Thus a line with slope c and axis intercept d is drawn in this subtask. Which means the line drawn is described by the equation $x = cy + d$, i.e. plotting x versus y. However we are interested in plotting y versus x. Therefore we first have to solve the equation for y, which yields $y = \frac{x}{c} - \frac{d}{c}$ and then use this equation for drawing a line with slope $\frac{1}{c}$ and axis intercept $\frac{1}{c}$:
R code:

```
> mod2 <- lm(x~y)
> c <- mod2$coefficients[2]
> d <- mod2$coefficients[1]
> abline(a=-d/c,b=1/c,col=2, lty = 2)
```



d) No, the straigth lines do not match. This would only be the case if all points lie exactly on the line. This can be understood as follows:
We want to estimate the regression line $y = \beta_0 + \beta_1 x$. As pointed out in the script (paragraph 2.3) it holds that

$$\begin{aligned} \hat{\beta}_0 &= \overline{y} - \hat{\beta}_1 \overline{x} \\ \hat{\beta}_1 &= r \tfrac{s_y}{s_x} \end{aligned}$$

where r is the estimated correlation between x and y (cf. paragraph 2.1 in the script). Thus the regression equation is given by:

$$y = (\overline{y} - r\frac{s_y}{s_x}\overline{x}) + r\frac{s_y}{s_x}x \Leftrightarrow y - \overline{y} = r\frac{s_y}{s_x}(x - \overline{x})$$

The regression from x to y follows similarly:

$$x = (\overline{x} - r\frac{s_x}{s_y}\overline{y}) + r\frac{s_x}{s_y}y \Leftrightarrow x - \overline{x} = r\frac{s_x}{s_y}(y - \overline{y})$$

These two lines match each other, if $r = \frac{1}{r}$, i.e. only if the correlation between x and y is equal to 1. That is equivalent to the graphical case that x and y lie on the diagonal line.

3. 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
```