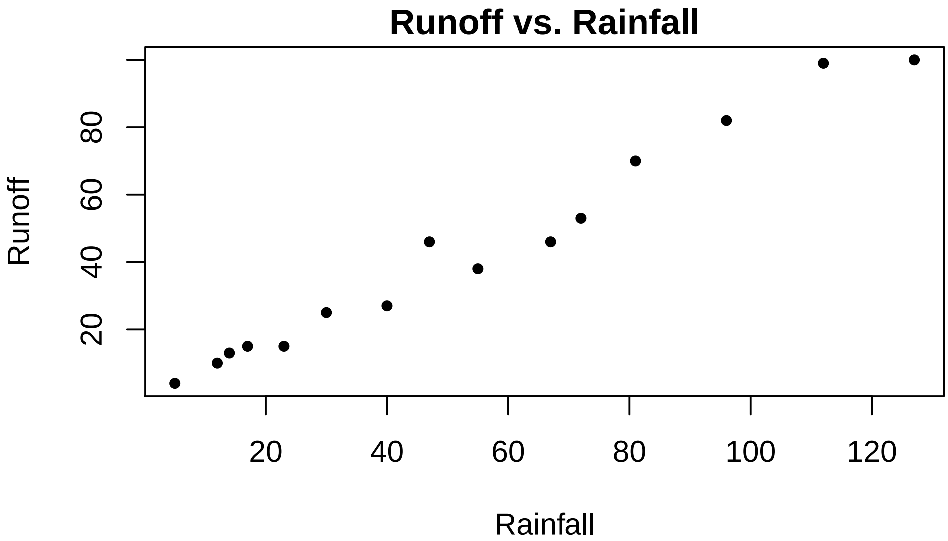
# Solution to Series 3

**1. a)** First we type in the data. The scatterplot of `runoff` versus `rainfall` suggests that a linear relationship holds. Therefore, one would guess that the $R^2$ should be large, i.e. close to 1.

```
> rainfall <- c(5, 12, 14, 17, 23, 30, 40, 47, 55, 67, 72, 81, 96, 112, 127)
> runoff <- c(4, 10, 13, 15, 15, 25, 27, 46, 38, 46, 53, 70, 82,  99, 100)
> data <- data.frame(rainfall=rainfall, runoff=runoff)
> plot(data$runoff ~ data$rainfall, pch=20, xlab="Rainfall", ylab="Runoff",
     main="Runoff vs. Rainfall")
```

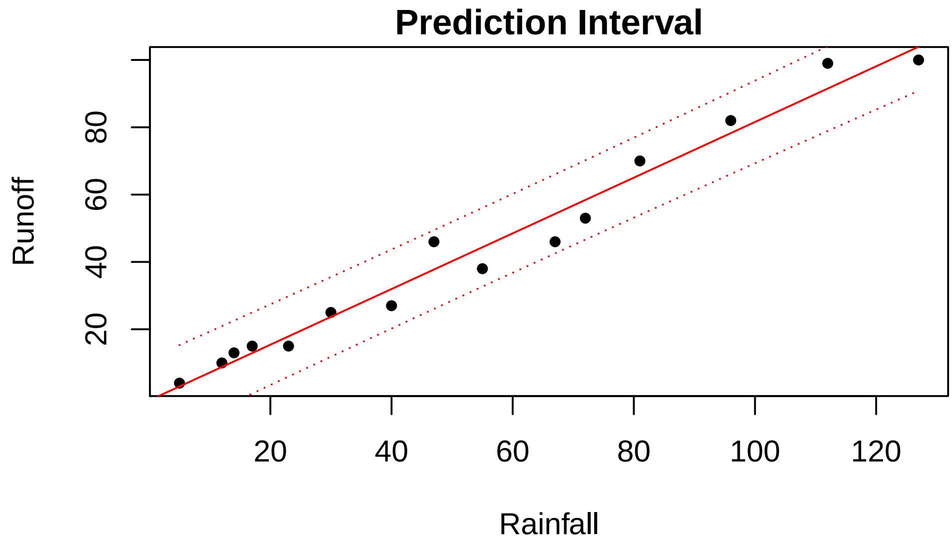



**b)** We fit a linear model with `runoff` as response and `rainfall` as predictor. We are then able to use this model for prediction.

```
> fit <- lm(runoff ~ rainfall, data=data)
> pred <- predict(fit, newdata=data.frame(rainfall=50), interval="prediction")
```

If the rainfall volume takes a value of 50 we find a runoff volume of 40.22 with a 95% prediction interval of [28.53,51.92].
We can also draw the regression line and the 95% prediction interval to the data.

```
> plot(data$runoff ~ data$rainfall, pch=20, xlab="Rainfall", ylab="Runoff",
     main="Prediction Interval")
> abline(fit, col="red")
> interval <- predict(fit, interval="prediction")
> lines(data$rainfall, interval[,2], lty=3, col="red")
> lines(data$rainfall, interval[,3], lty=3, col="red")
```

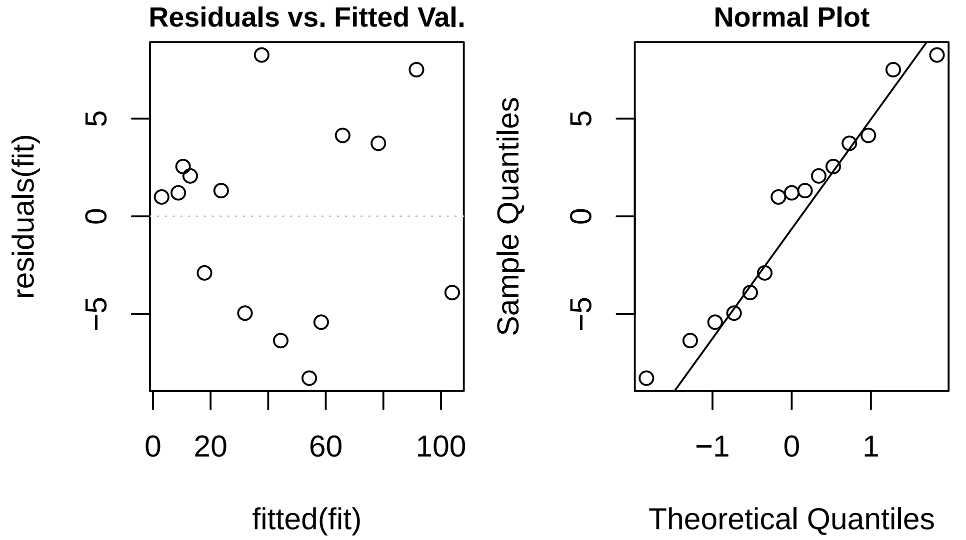



**c)** An $R^2$ of 0.98 is extremely high, i.e. a huge part of the variation in the data can be attributed to the linear association between runoff and rainfall volume.

**d)**
```
> summary(fit)

Call:
lm(formula = runoff ~ rainfall, data = data)

Residuals:
   Min     1Q Median     3Q    Max
-8.279 -4.424  1.205  3.145  8.261

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.12830    2.36778  -0.477    0.642
rainfall     0.82697    0.03652  22.642  7.9e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.24 on 13 degrees of freedom
Multiple R-squared:  0.9753,	Adjusted R-squared:  0.9734
F-statistic: 512.7 on 1 and 13 DF,  p-value: 7.896e-12

> ## Confidence intervals for the coefficients
> confint(fit)

                 2.5 %    97.5 %
(Intercept) -6.2435879 3.9869783
rainfall     0.7480677 0.9058786
```

There is a significant linear association between runoff and rainfall volume, since the null hypothesis $\beta_1 = 0$ is clearly rejected. However, the confidence interval for $\beta_1$ does not contain $\beta_1 = 1$, i.e. a null hypothesis of $\beta_1 = 1$ would be rejected, too. Therefore, we conclude that no $1:1$ relation between rainfall and runoff holds. We suspect that part of the rain evaporates or trickles away.

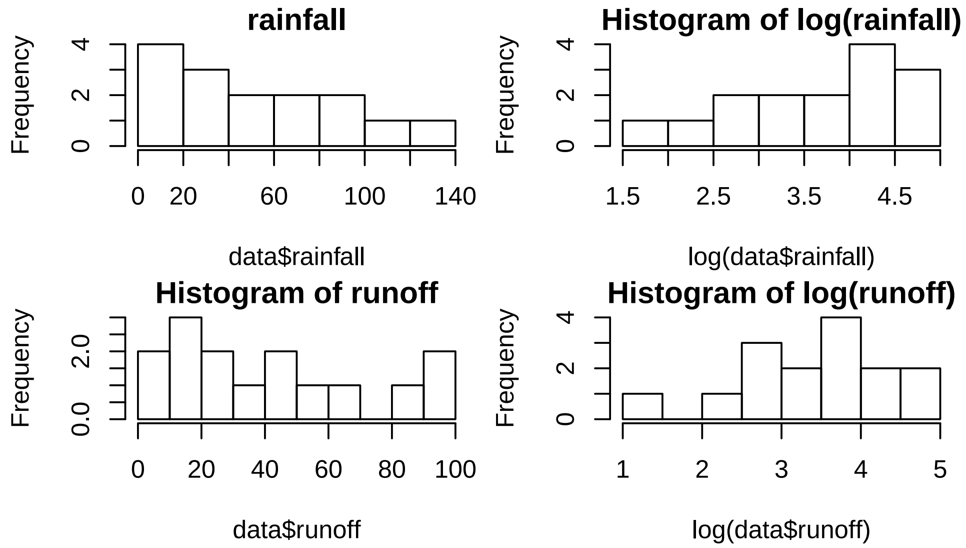
**e)**
```
> par(mfrow=c(1,2))
> plot(fitted(fit), residuals(fit), main="Residuals vs. Fitted Val.", cex.main=0.9)
> abline(h=0, col="grey", lty=3)
> qqnorm(residuals(fit), main="Normal Plot", cex.main=0.9)
> qqline(residuals(fit))
```




From the Tukey-Anscombe plot (residuals vs. fitted values) we observe a non-constant variance of the residuals. With increasing runoff the residuals increase.

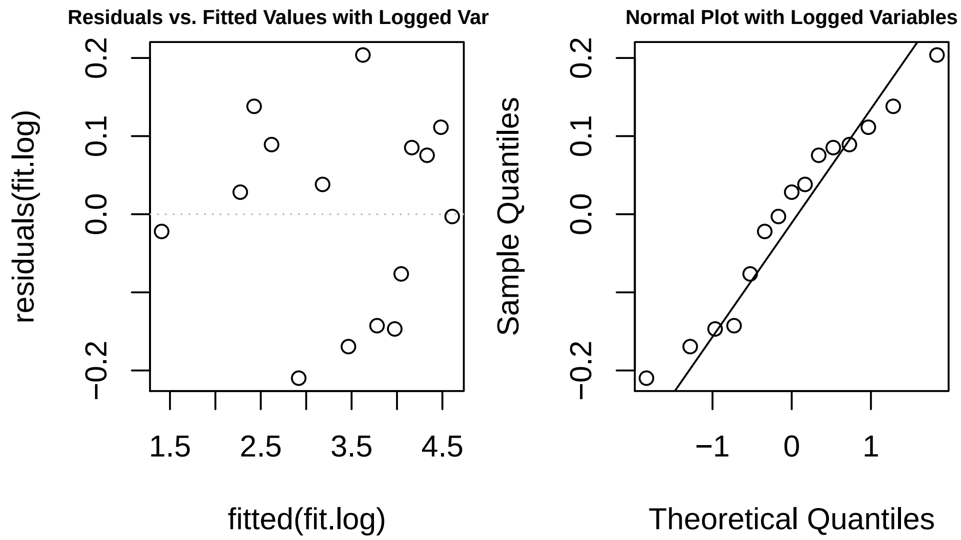
**f)** Although the histograms of the original data do not strongly point to a log-transformation, we try it and will see that it turns out to be useful.

```
> par(mfrow=c(2,2))
> hist(data$rainfall, 8, main="rainfall")
> hist(log(data$rainfall), 8, main="Histogram of log(rainfall)")
> hist(data$runoff, 8, main="Histogram of runoff")
> hist(log(data$runoff), 8, main="Histogram of log(runoff)")
```



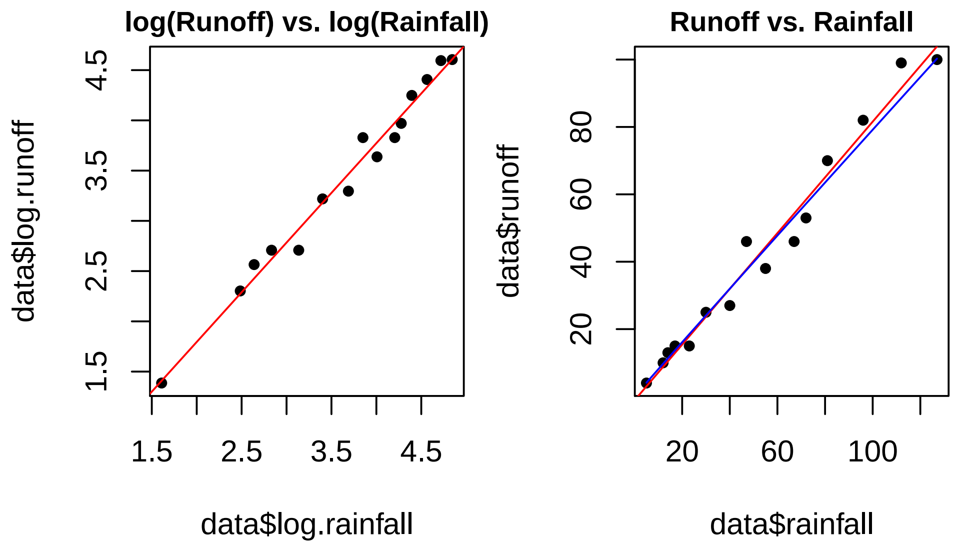

From the diagnostic plots we can see that the model on the transformed scale performs better, and the constant variance assumption seems more justified.

```
> data$log.runoff <- log(data$runoff)
> data$log.rainfall <- log(data$rainfall)
> fit.log <- lm(log.runoff ~ log.rainfall, data=data)
> par(mfrow = c(1,2))
> plot(fitted(fit.log), residuals(fit.log),
     main="Residuals vs. Fitted Values with Logged Variables",
     cex.main=0.7)
> abline(h=0, col="grey", lty=3)
> qqnorm(residuals(fit.log), main="Normal Plot with Logged Variables", cex.main=0.7)
> qqline(residuals(fit.log))
```




However, differences between the two models are small.

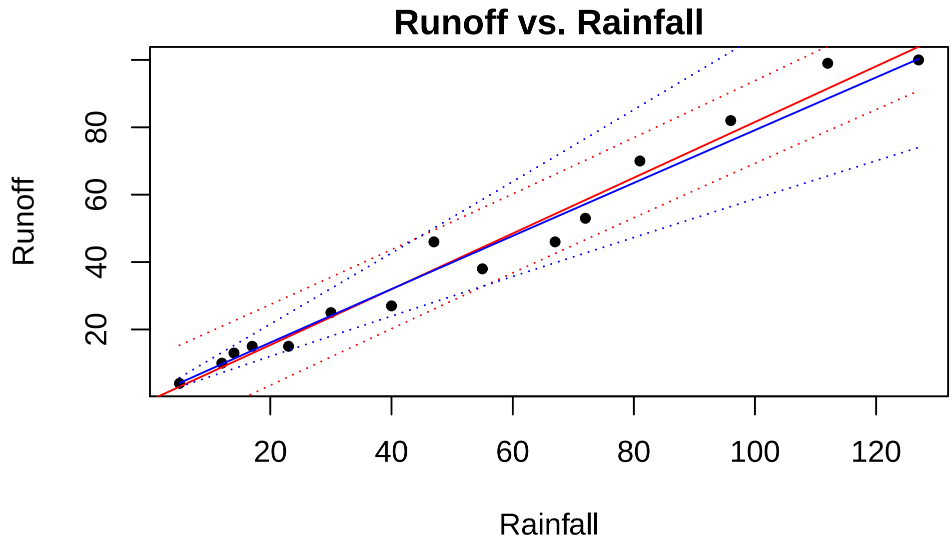
```
> par(mfrow=c(1,2))
> ## Scatterplot on the log scale
> plot(data$log.rainfall, data$log.runoff,
      main="log(Runoff) vs. log(Rainfall)", cex.main=0.9,
      pch=20)
> abline(fit.log, col="red")
> ## Scatterplot on original scale
> plot(data$rainfall, data$runoff, main = "Runoff vs. Rainfall", cex.main=0.9,
      pch=20)
> abline(fit, col="red")
> lines(rainfall, exp(predict(fit.log)), col="blue")
```




g) On the original scale the prediction interval of the log-transformed model is of the form of a trumpet (blue dot lines). This is more realistic, especially since fitted values and the prediction interval of the log-transformed model have positive values. Negative runoff values, as seen on the original scale, are impossible that is why the log-transformed model is superior to the original one. Although differences in the diagnostic plots seem small and problems appear to be more academic than fundamental, the log-transformed model resulting from a thorough statistical analysis pays off.

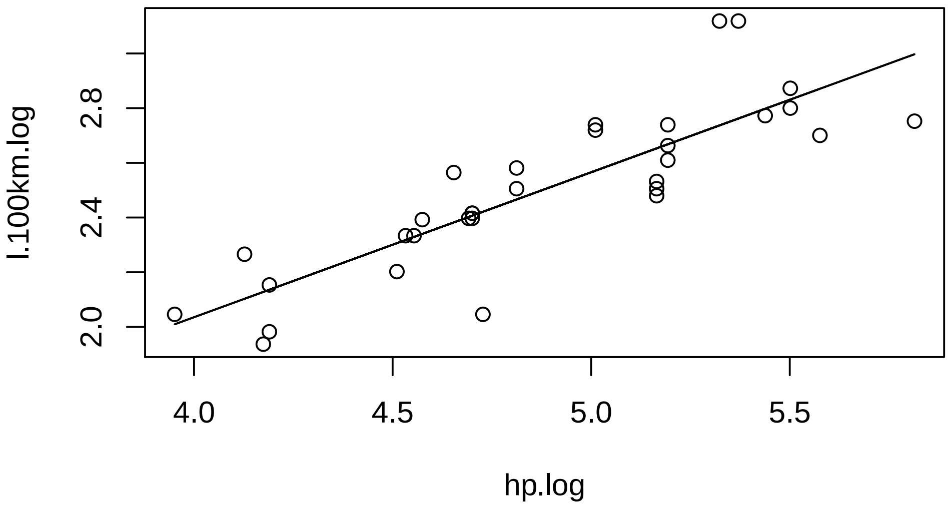
```
> ## Prediction intervals of the transformed model on the original scale
> plot(data$rainfall, data$runoff, pch=20, xlab="Rainfall", ylab="Runoff",
      main="Runoff vs. Rainfall")
> abline(fit, col="red")
> lines(data$rainfall, exp(predict(fit.log)), col="blue")
```

```
> interval <- predict(fit, interval="prediction")
> lines(data$rainfall, interval[,2], lty=3, col="red")
> lines(data$rainfall, interval[,3], lty=3, col="red")
> interval.log <- predict(fit.log, interval="prediction")
> lines(data$rainfall, exp(interval.log[,2]), lty=3, col="blue")
> lines(data$rainfall, exp(interval.log[,3]), lty=3, col="blue")
```



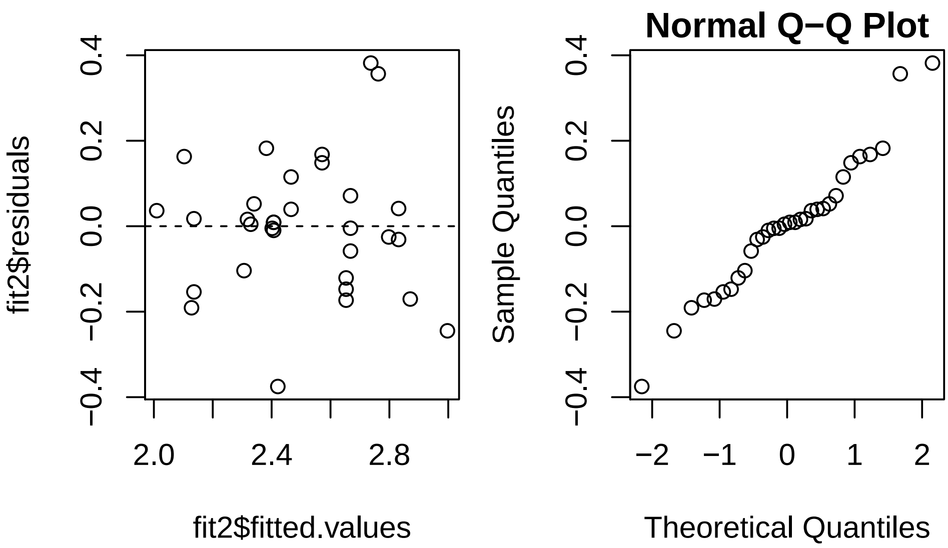

**2.** a)

```
> # 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)
```

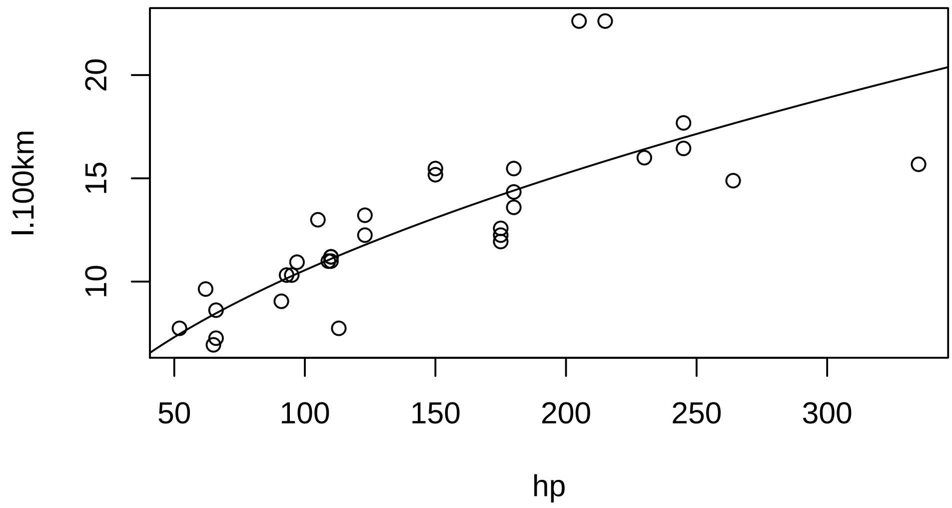



**b)** Exponentiating yields:

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

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.

**c)**
```
> # 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 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
```