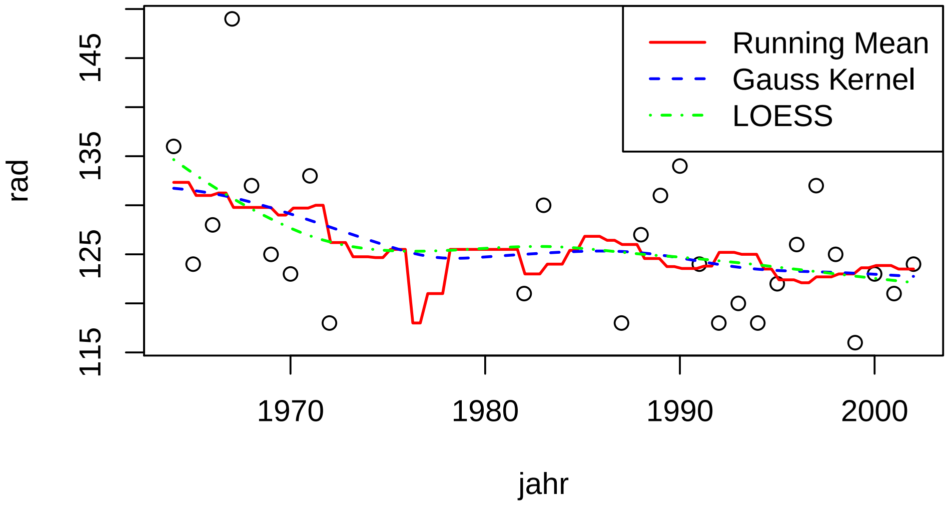
Solution to Series 2

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, the answer here is highly subjective, since is not possible to give quantitative evidence to this claim just by using non-parametric smoothing.

c)

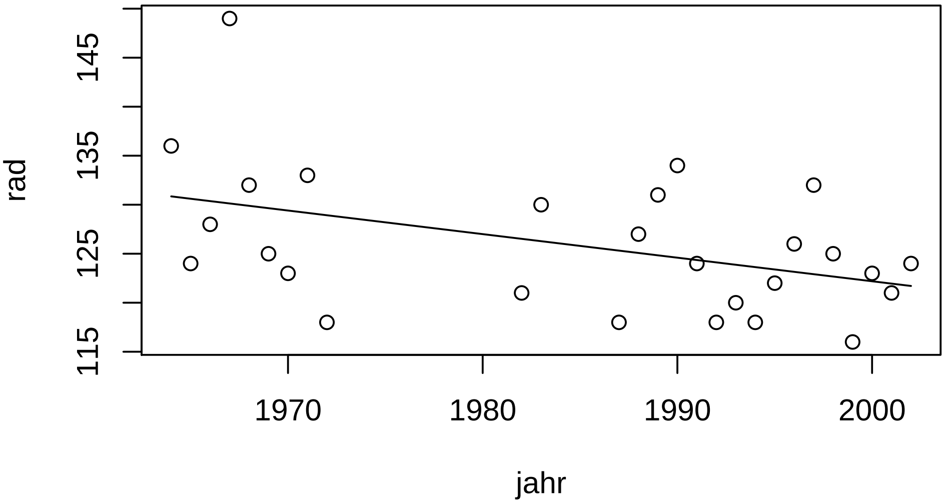
```
> # Plot scatter plot and regression line
> plot(sol.rad)
> fit.lm <- lm(rad~jahr, sol.rad)
> lines(sol.rad$jahr, fit.lm$fitted.values)
> # Print fit summary
> summary(fit.lm)
```

```
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)
```

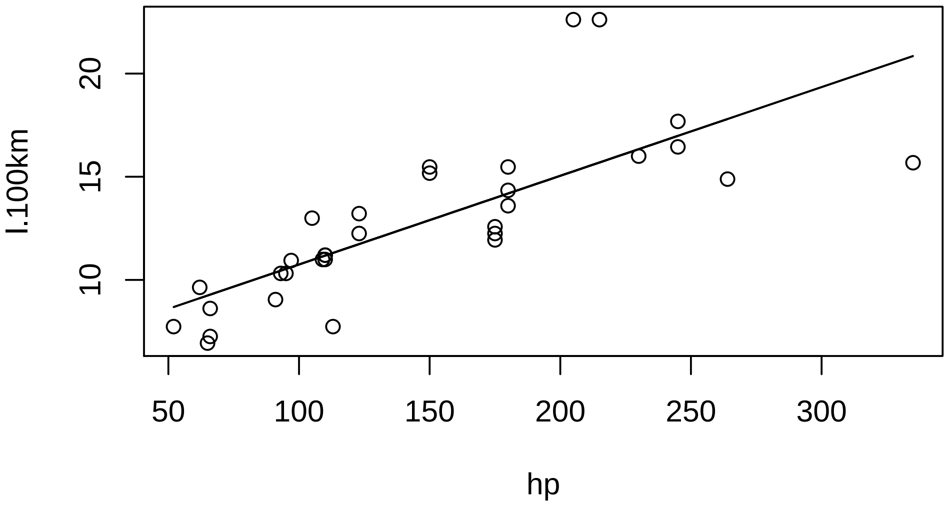
```
Call:
lm(formula = l.100km ~ hp, data = my.mtcars)

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 regression coefficient for the variable engine power is equal to 0.04299. The coefficient of a predictor variable in a linear regression model represents the amount by which the response variable changes if that predictor variable increases by one unit. Therefore, when engine power increases by $10\mathtt{hp}$ the fuel consumption will increase by $0.04299 * 10 = 0.4299$.

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.
We can also come by this result by calculating the predicted value of the fuel consumption from the fitted model equation $\hat{y} = \hat{\beta_0} + \hat{\beta_1}x$.
Here, $\hat{\beta_0} = 6.44908$, $\hat{\beta_1} = 0.04299$. Thus, when $x = 100$ the predicted value of the fuel consumption is equal to $\hat{y} = 6.44908 + .04299 * 100 = 6.44908 + 4.299 = 10.74808 \approx 10.75$
Which is the same value we get using the predict function in R.
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.

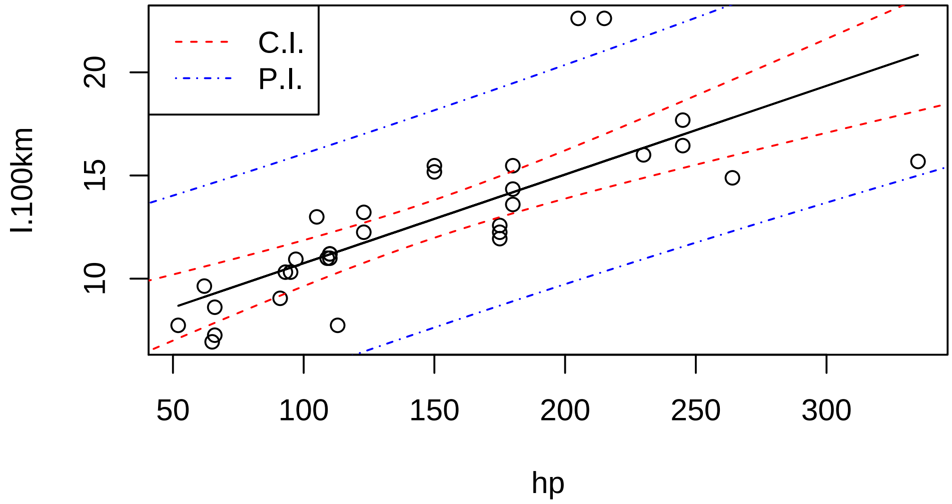
d) We can just calculate the confidence interval for the slope parameter β_1 and see whether it includes the value 0.05.

```
> 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.

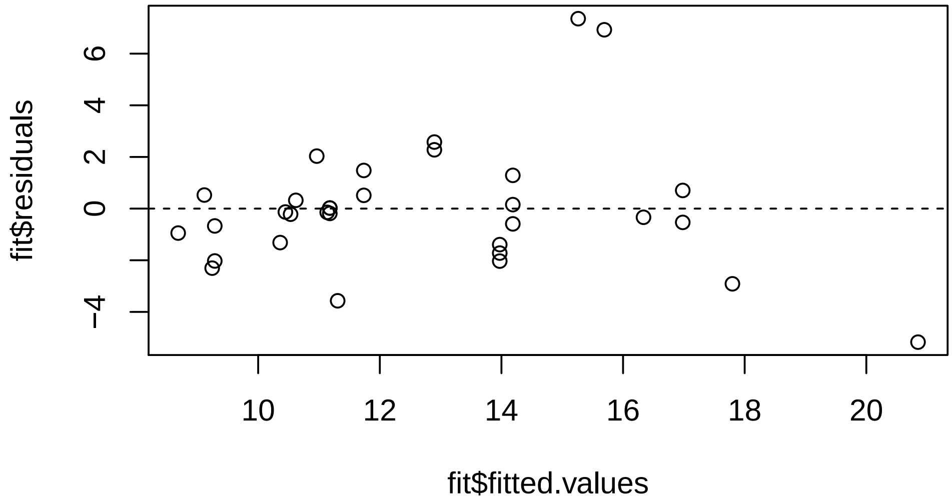
e)

```
> # 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.

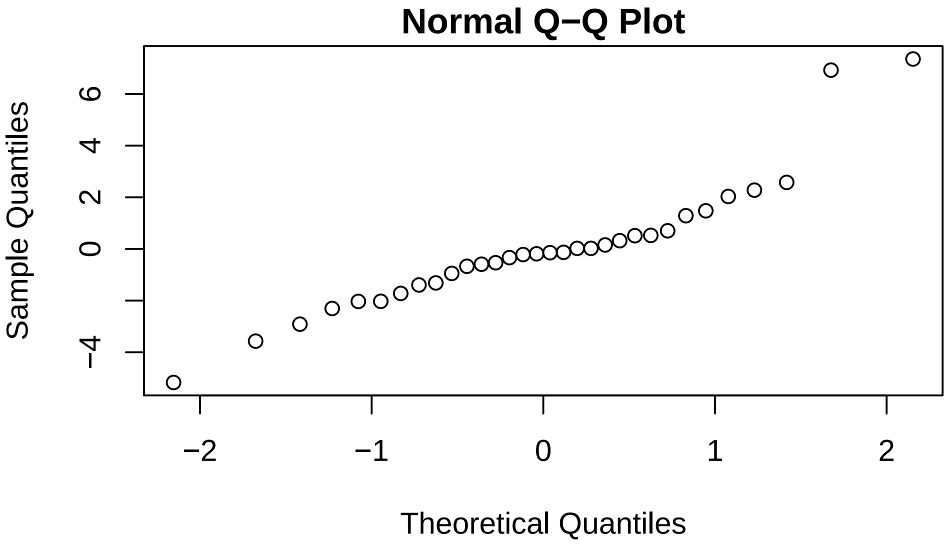
```
> plot(fit$fitted.values, fit$residuals)
> abline(0, 0, lty=2)  # Dashed line at zero
```



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. Thus, this model, and it's results which are calculated in the previous points of the exercise are not trustworthy and shouldn't be reported.

3. **a)** False. The Gaussian kernel smoother is particularly sensitive to outliers. In this situation the best smoother of the 3 to use is a LOESS smoother, because it provides the option of using a robust procedure to fit the data, which can deal with outliers.

b) True. See page 2 of the Scriptum.

c) False. A simple linear regression model should have only one predictor variable.

d) False. Smoothing methods cannot be used for extrapolation.

e) False. Firstly, the R^2 value can only give an idea of the goodness-of-fit of a model, but there is no formal criteria for which minimal value of R^2 needs to be met for a model tobe useful. More importantly, we can't use a good fit of a regression model to conclude causation.