Workshop Week 12, Assumptions and Diagnostic Plots

Johnathan Marshall, Nick Knowlton

2024-10-10

Table of contents

Exercise 1: Visualising the model with a prediction band

`visreg` provides us an efficient routine to visualise our linear model. However, it only produces the line with a confidence band. We need to do a bit more work to visualise our model with a prediction band.

Question 1

Load the package `datarium` and the data `marketing`. Fit the linear model, and save the result in the object `lm.youtube` again. Then, sweep your fitted result by the package `broom` and get the tidy version `lm.youtube`.

```
library(datarium)
library(broom)
data(`marketing`)
lm.youtube <- lm(sales ~ youtube, data=marketing)
lm.youtube.fit <- augment(lm.youtube)
```

Question 2

Get all prediction intervals for each observed `youtube` by calling `predict()` without supplying `new.data`, and save the result in `lm.youtube.pred`. We can then bind `lm.youtube.pred` with `lm.youtube.fit` by column with the R function `cbind()`, aka column bind.

```
lm.youtube.pred <- predict(lm.youtube, interval='prediction')
```

```
Warning in predict.lm(lm.youtube, interval = "prediction"): predictions on current data refe
```

```
lm.youtube.fit.pred <- lm.youtube.fit |> cbind(lm.youtube.pred) |> select(-fit) |> tibble()
lm.youtube.fit.pred
```

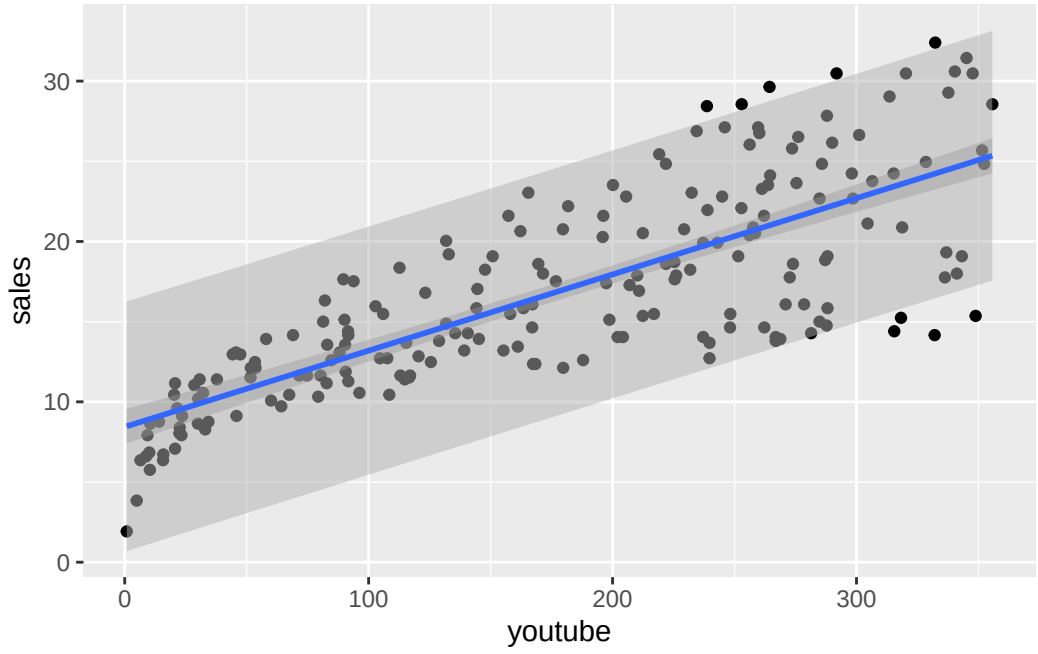
```
# A tibble: 200 x 10
   sales youtube .fitted .resid    .hat .sigma   .cooksd .std.resid   lwr   upr
   <dbl>   <dbl>   <dbl>  <dbl>   <dbl>  <dbl>     <dbl>      <dbl> <dbl> <dbl>
 1 26.5    276.    21.6   4.96  0.00970   3.90 0.00794       1.27   13.8   29.3
 2 12.5     53.4   11.0   1.50  0.0122    3.92 0.000920      0.387   3.22  18.7
 3 11.2     20.6    9.42  1.74  0.0165    3.92 0.00169       0.449   1.65  17.2
 4 22.2    182.    17.1   5.12  0.00501   3.90 0.00434       1.31    9.35  24.8
 5 15.5    217.    18.8  -3.27  0.00578   3.91 0.00205      -0.839  11.0   26.5
 6  8.64    10.4    8.94 -0.295 0.0180    3.92 0.0000534    -0.0762  1.15  16.7
 7 14.2     69     11.7   2.44  0.0105    3.92 0.00208       0.627   3.97  19.5
 8 15.8    144.    15.3   0.544 0.00549   3.92 0.0000538     0.140   7.56  23.0
 9  5.76    10.3    8.93 -3.17  0.0181    3.91 0.00616      -0.818   1.15  16.7
10 12.7    240.    19.8  -7.12  0.00690   3.89 0.0116       -1.83   12.1   27.6
# i 190 more rows
```

Here we further remove the duplicate column ***fit****.*

Question 3

Visualise your linear model with the fitted line and the scatter plot. You can then add the prediction band by the R function `geom_ribbon()` with `aes(ymin = lwr, ymax = upr)` which defines a shaded region bounded by the lower limits and upper limits of the prediction intervals in your plot. You may need to adjust `fill` and `alpha` in `geom_ribbon()` to get a nicer plot.

```
lm.youtube.fit.pred |> ggplot(aes(x = youtube ,y=sales)) +
    geom_point() +
    geom_ribbon(aes(ymin = lwr, ymax = upr), fill = "grey70", alpha=0.5) +
    geom_smooth(method='lm', formula=y~x)
```



Question 4

The default prediction interval in R is a 95% prediction interval which means around 95% of observations should fall within the bands. Check the plot in Step 3 and count the points falling outside the prediction band. Add one or two comments on your findings.

```
points_in_band <- lm.youtube.fit.pred |>
    summarise(count_in_band = sum(sales >= lwr & sales <= upr),
              total_points = n(),
              percentage_in_band = (count_in_band / total_points) * 100)
```

```
points_in_band
```

```
# A tibble: 1 x 3
  count_in_band total_points percentage_in_band
          <int>        <int>              <dbl>
1           191          200               95.5
```

```
# Answer
# Upon checking the plot, we can see that there are a few points falling outside the predicti
# consistent with the 5% expected due to the 95% prediction interval. This observation matche
# concept that approximately 95% of data points should fall within the bounds, while about 5%
```

Question 5

Actually, the prediction band we plotted is a **pointwise** 95% prediction interval. The term **pointwise** means that each individual observation falls within the corresponding prediction interval with a probability of 95%. Try to split `youtube` into two groups `youtube < 200` and `youtube > 200` and count the points falling in the prediction band for each group. Add one or two comments on your findings.

```
lm.youtube.fit.pred <- lm.youtube.fit.pred |>
    mutate(group = ifelse(youtube < 200, "< 200", "> 200"))

points_in_band <- lm.youtube.fit.pred |>
    group_by(group) |>
    summarise(count_in_band = sum(sales >= lwr & sales <= upr),
              total_points = n(),
              percentage_in_band = (count_in_band / total_points) * 100)

points_in_band
```

```
# A tibble: 2 x 4
  group count_in_band total_points percentage_in_band
  <chr>         <int>        <int>              <dbl>
1 < 200           107          107              100
2 > 200            84           93               90.3
```

```
# Answer
# After splitting the data into two groups (`youtube < 200` and `youtube > 200`), we find tha
# of points falling within the prediction band is too high in the <200 and too low in the > 2
# minor variations due to the sample sizes, but the overall trend confirms the discrepency in
```

Exercise 2: Model diagnostics

As we can see from Exercise 1, the prediction interval tends to be too wide when `youtube` is large and too narrow when `youtube` is small. A similar issue also arises when we try to predict the sales given zero budget in Exercise 3 of Lab C2. The prediction at zero budget may significantly overestimate the actual sales.

These issues allude to a critical point: does our linear model provide a good fit to our data?

Of course, we can always get some clues from the scatter plot and smoothed curve for `sales` against `youtube`. We can even conclude that the fitted line fails to capture a significant portion of uncertainties in our data. So a linear model may not be an adequate model for the relationship between `sales` and `youtube`.

The next question is how to identify and quantify the inadequacy of our linear model?

The visualisation of model identifies some ill-posed patterns in our model but it fails to identify the root cause of the problem. R^2 is a simple indicator but it won't capture the issues on prediction discussed above.

What we need are some diagnostic tools to check the fitness of our linear model on the data set and identify the crux in our fitted model. However, before we carry out the examination on our fitted linear model, we need to know when a fitted linear model will behave well.

To ensure the overall good performance of a fitted linear model, we need our data to satisfy a few conditions, i.e. four assumptions as follows:

- **L**inearity

  Residuals don't depend on x. The trend is correctly modelled as a line. A line won't fit an exponential trend.

- **I**ndependence

  Residuals don't depend on each other. If residuals are dependent, we can expect that one residual may contribute some information to another residual and vice versa.

- **N**ormality

  Residuals are distributed normally. Normality ensures that the least square estimation will catch the best possible line.

- **E**qual variance

  Residuals have constant variance. The variation doesn't change as we move along the trend.

with Linearity being the most important.

Question 1

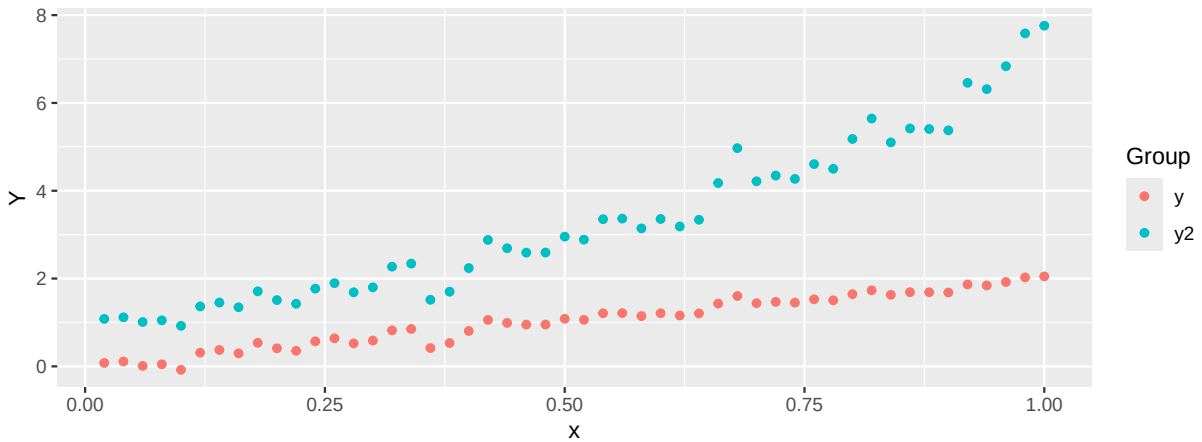
Firstly we'll be looking at linearity and equal variance, both of which can be assessed using a plot of **the residuals versus the fitted value**.

If linearity holds, we'd expect a plot of residuals vs fitted value to show no trend - the points should be scattered fairly constantly above and below the line - in particular we don't want to see a curve.

If equal variance holds, we'd expect the scatter of points around the trend to be constant as the fitted value changes. You want it to be relatively even, and in particular not increasing from left to right (i.e. not spreading out).

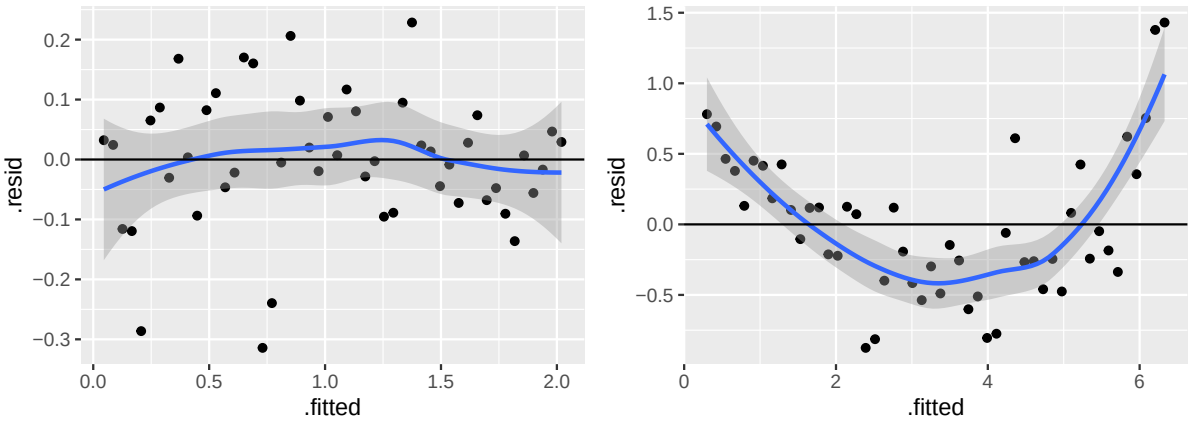
We can demonstrate this idea by using the stochastic simulation. An example of a good plot (left) and bad plot (right) is shown below for two artificial data sets, i.e., linear and exponential.

```
set.seed(2020)
n <- 50
ab <- c(0,2)
demo <- tibble(x=(1:n)/n, e= rnorm(n, sd=0.1)) |> mutate (y=ab[1]+ab[2]*x+e) |>
  mutate (y2=exp(ab[1]+ab[2]*x+e))
demo |> select(x, y, y2) |> gather(key = "Group", value = "Y", -x) |>
  ggplot(aes(x=x,y=Y,col=Group)) + geom_point()
```



```
g1 <- lm(y~x,data=demo) |> augment() |>
  ggplot(aes(x=.fitted,y=.resid)) + geom_point() +geom_smooth(formula=y~x)+geom_hline(yintere
g2 <- lm(y2~x,data=demo) |> augment() |>
  ggplot(aes(x=.fitted,y=.resid)) + geom_point() +geom_smooth(formula=y~x)+geom_hline(yintere
library(patchwork)
g1+g2
```

```
`geom_smooth()` using method = 'loess'
`geom_smooth()` using method = 'loess'
```



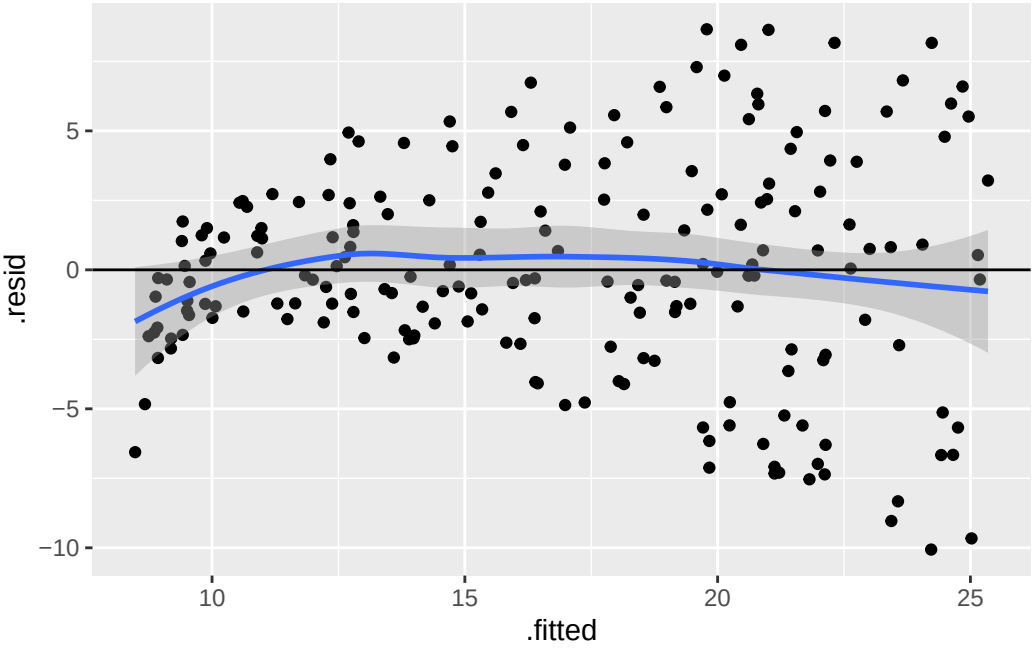
Compare the scatter plots and the plots of residuals versus fitted values. Think about the reason why we include horizontal lines at $y = 0$ and smoothed curves with confidence bands in the residual plots.

Question 2

Let's see how well our model for sales does by producing the diagnostic plot for the linear model you fit above using the following code.

```
lm.youtube.fit |> ggplot(aes(x=.fitted,y=.resid)) +
  geom_point() + geom_smooth(formula=y~x) + geom_hline(yintercept=0)
```

```
`geom_smooth()` using method = 'loess'
```



Take a good look at the plot. Do you think linearity and equal variance hold? Add some notes about each assumption to your notebook.

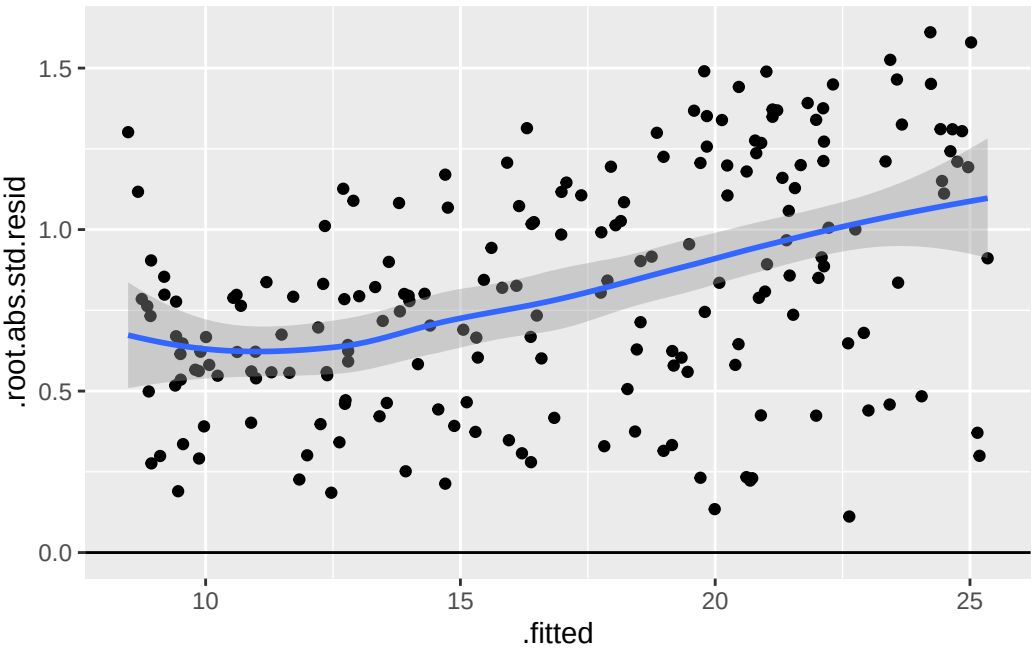
```
# Answer
# The plot shows that there is a slight trend in the residuals, suggesting that linearity may
# Additionally, the spread of residuals appears to increase slightly as the fitted values inc
```

Question 3

In addition to the residuals vs fits plot, we have another tool, i.e. **the scale-location plot**, to check linearity and equal variance. The plot can be generated by using the following R code chunk:

```
lm.youtube.fit |> mutate(.root.abs.std.resid=sqrt(abs(.std.resid))) |>
  ggplot(aes(x=.fitted,y=.root.abs.std.resid)) + geom_point() + geom_smooth(formula=y~x) + ge
```

```
`geom_smooth()` using method = 'loess'
```



Similar to the residual versus fits plot, the scale-Location plot shows whether residuals are spread equally along the ranges of input variables (predictor). The assumption of equal variance (**homoscedasticity**) could also be checked with this plot. If we see a horizontal line with randomly spread points, it means that the model is good.

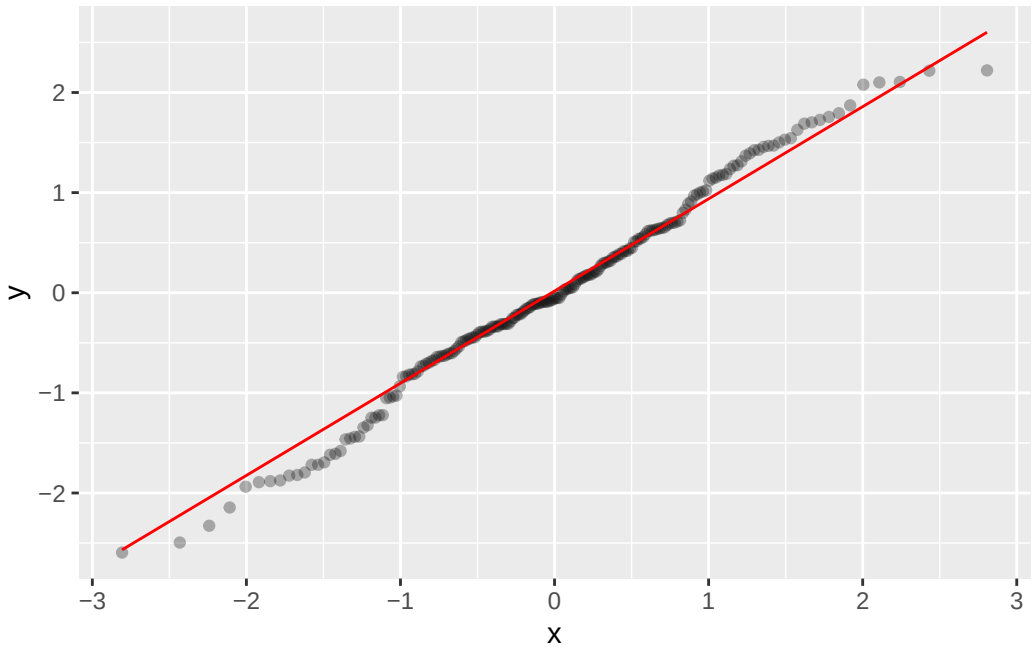
Take a good look at the plot. Do you think linearity and equal variance hold? Add some notes about each assumption to your notebook.

```
# Answer
# The scale-location plot shows a slight upward trend, suggesting that the residuals' varianc
# This means that the equal variance assumption might be violated.
```

Question 4

The residuals vs fits plot and scale-location plot are usually sufficient for us to tell the issues on linearity and equal variance. Our next task is to verify the normality. Recall the simulation study in Lab 2 where we make a scatter plot of the residuals versus true random errors which can be approximated by the line $y = x$. Even if we can't observe the true random error in a real problem, we can still make a similar plot called **quantile-quantile plot**, aka **Q-Q plot**, via `geom_qq()` and `geom_qq_line` as follows.

```
lm.youtube.fit |> ggplot(aes(sample=.std.resid)) + geom_qq(alpha=0.3) +  geom_qq_line(color=
```



Take a good look at the above two plots. Do you think the normality hold? Add some notes about the assumption to your notebook.

```
# Answer
# The Q-Q plot shows that most points fall along the red line, which suggests that the residu
# However, there are deviations at quartiles plus and minus 1, indicating possible non-normal
```

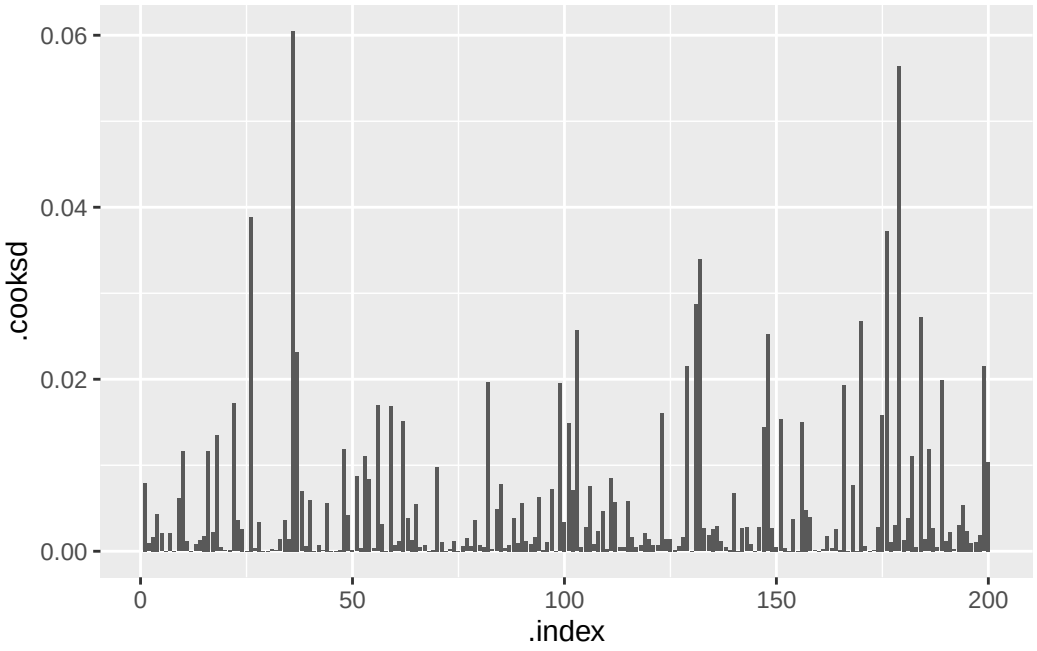
Question 5

The last thing to check is slightly different from the above assumptions. We will look at the **outliers** in linear model. Just like we observe some distant points in a boxplot for one variable, we can have some outliers for x and y. The trick thing is that, even both x and y look good in their own boxplots, their joint effort may push the point away from the main body in a scatter plot. A possible pitfall in fitting a linear model is that the linear model found by least square method is not very robust against outliers. Only a few outliers can distort the point estimates significantly. So it is essential to spot those bad guys hidden in our data.**

Of course, even a scatter plot can help us identify the outliers. If we want some more quantitative measurements, there is a specific measure called **Cook's distance**, or just Cook's D, for each data point to measure its influence on the regression line. A large Cook's distance suggests that this point may be an outlier which has a big influence on the whole regression line.

This piece of information is readily collected by `augment()` in the column `.cooksd`. The following R code chunk produces a plot of Cook's distance against the index.

```
lm.youtube.fit |> mutate(.index=1:n()) |> ggplot(aes(y=.cooksd,x=.index)) + geom_col()
```

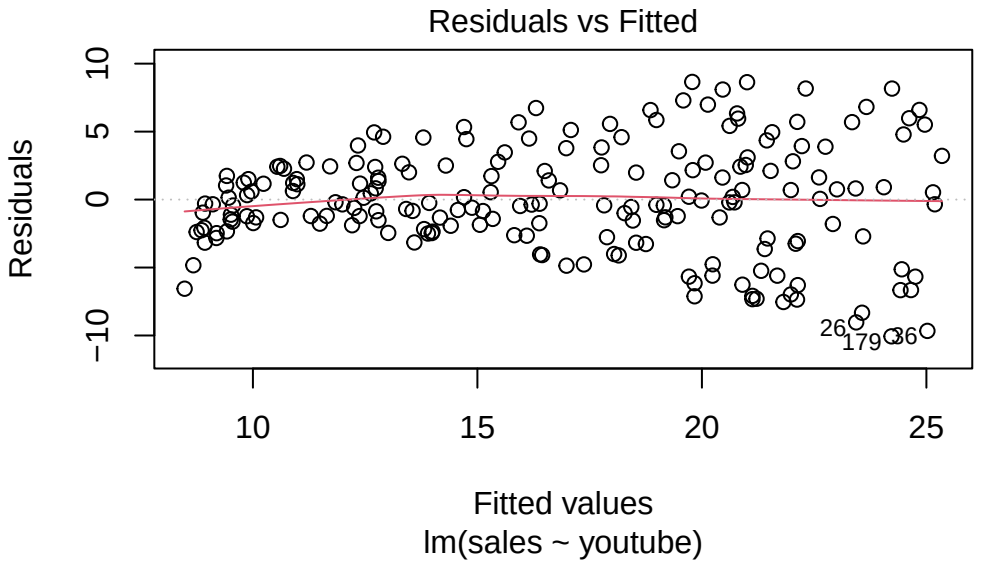


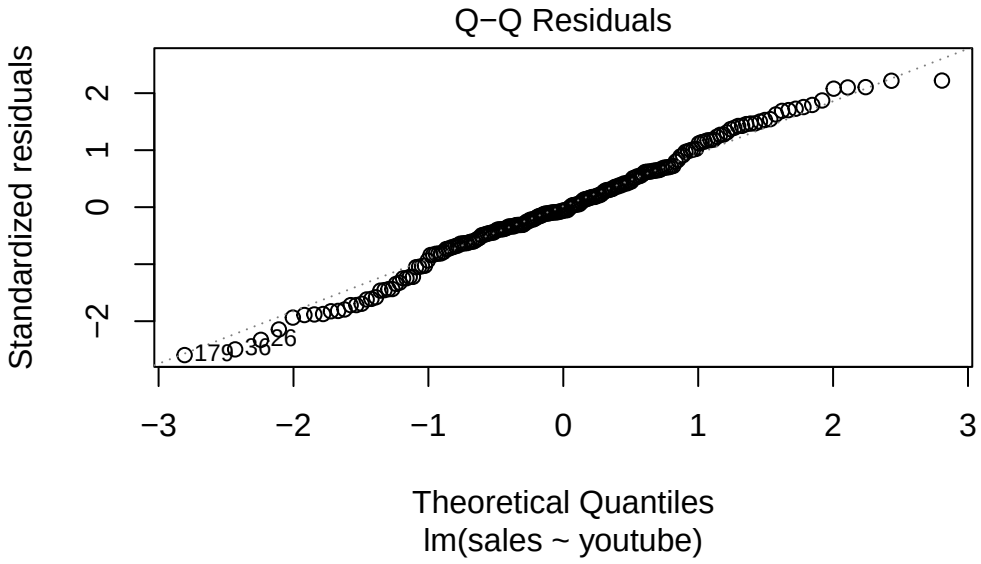
Those spikes in the above plot suggest potential outliers. However, one needs a cut-off point to distinguish the outliers from those tamed observations. Unfortunately, there is no golden rule to split these two groups of points for arbitrary data sets. Some people use 0.5 or 1 as a cut-off point but our Cook's distances here are much smaller. We will see a refined diagnostic plot for outlier detection in the next step.

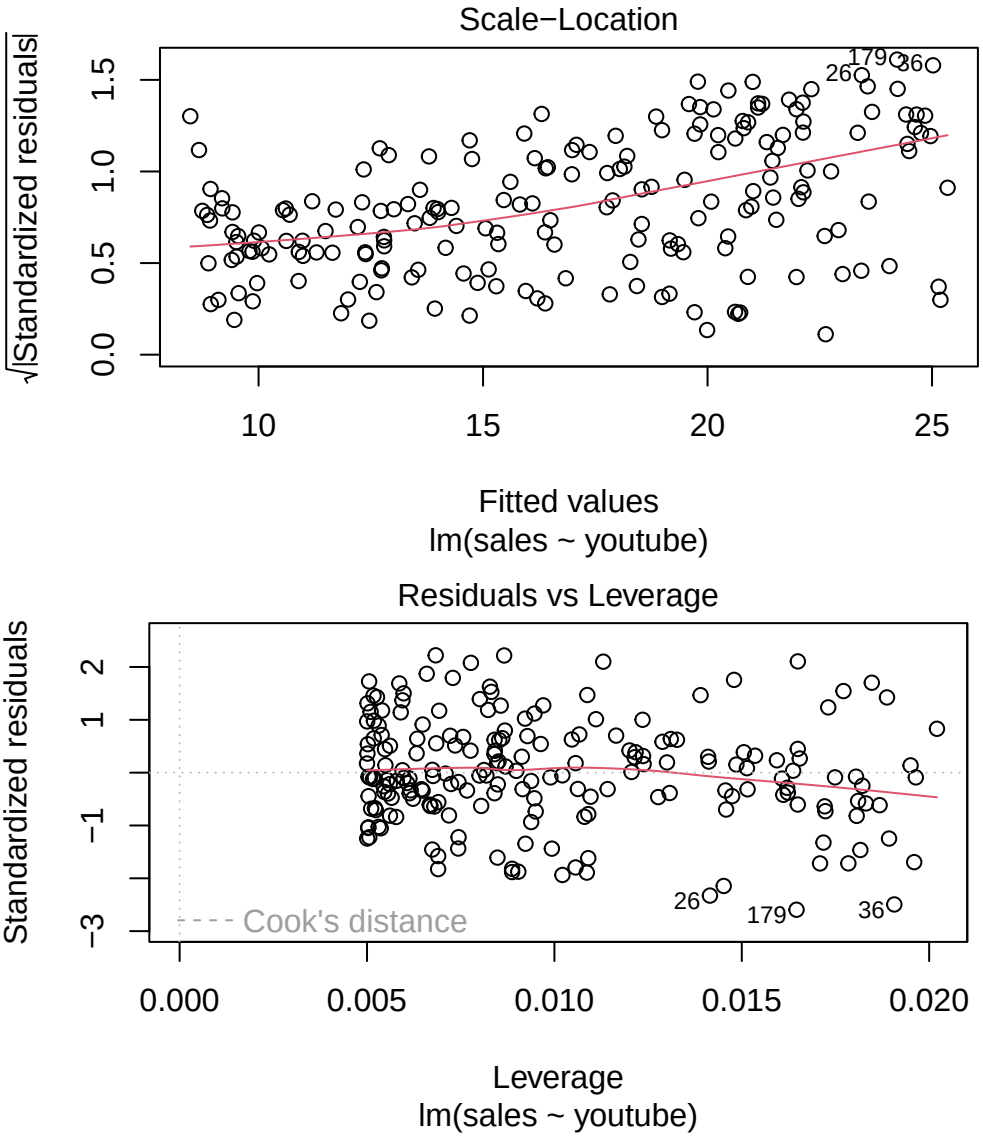
Question 6

One can easily generate a set of residuals diagnostic plots via `plot()`, i.e the default plot function in R, as follows.

```
plot(lm.youtube)
```


Residuals vs Fitted
Residuals
10
5
0
−10
26
179
36
10
15
20
25
Fitted values
lm(sales ~ youtube)


Q−Q Residuals
Standardized residuals
2
1
0
−2
179
36
26
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(sales ~ youtube)



The last one of the above plots is called **the residuals versus leverage plot**. The **leverage** is another measure for identifying outliers and you shall notice that there are several points being flagged already. This plot will also contain a cut-off curve in the red dash line for Cook's distances if there are any points with Cook's $D > 0.5$.

Besides the convenience, another advantage of `plot()` is that the indices of potential outliers will be flagged in the plot. You can use the information in these diagnostic plots to locate the untamed outliers in your data.

However, one shall notice that we can't customise those plots by following the same procedures in `ggplot()`.

In fact, `plot()` can produce six different diagnostic plots. You can make the individual plot by specifying `which` in `plot()` like `plot(lm.youtube,which=1)` or `plot(lm.youtube,which=4)`. Have a try with `which=1,2,3,4,5,6` and see which number gives you the desired plot.

Exercise 3: Log Model

We have to acknowledge that the residuals plots of `lm.youtube` suggest that the linear model is not a great one for describing the relationship between `sales` and `youtube`. A transformation is one way to deal with the non-linearity and unequal variance of the data. We will try the log transformation in this exercise.

Question 1

Instead of modelling `sales` in terms of `youtube`, we could instead take log transforms of `sales` and `youtube` to see if it is possible to get rid of the curvature in the relationship.

```
lm.youtube.log <- lm(log(sales) ~ log(youtube), data=marketing)
summary(lm.youtube.log)
```

```
Call:
lm(formula = log(sales) ~ log(youtube), data = marketing)

Residuals:
     Min       1Q   Median       3Q      Max
-0.43349 -0.15917  0.01696  0.16910  0.39399

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.02284    0.07372   13.87   <2e-16 ***
log(youtube)  0.35504    0.01487   23.87   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2109 on 198 degrees of freedom
Multiple R-squared:  0.7421,    Adjusted R-squared:  0.7408
F-statistic: 569.8 on 1 and 198 DF,  p-value: < 2.2e-16
```

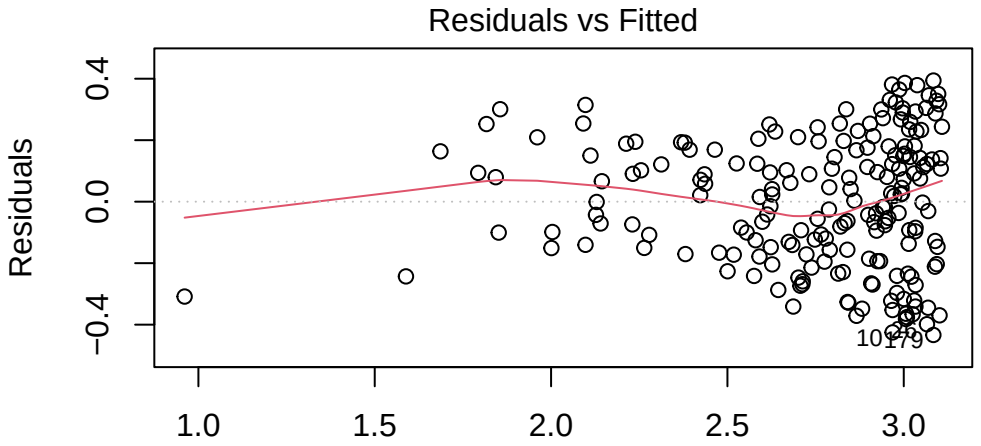
Compare the summary output of the two models you have. Which do you think is better? Why?

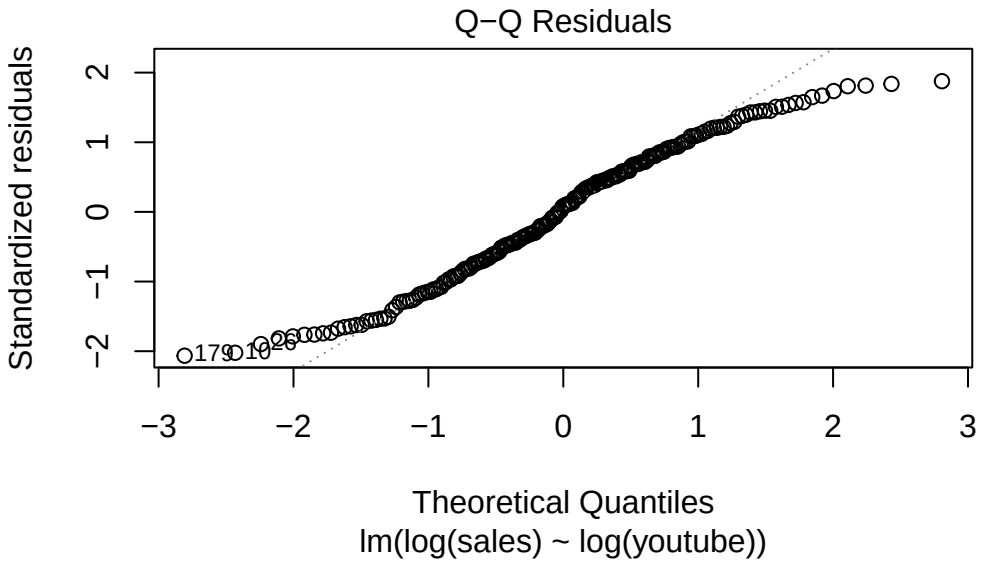
```
# Answer
# The summary of the log-transformed model shows an increase in the R-squared value, suggesti
```

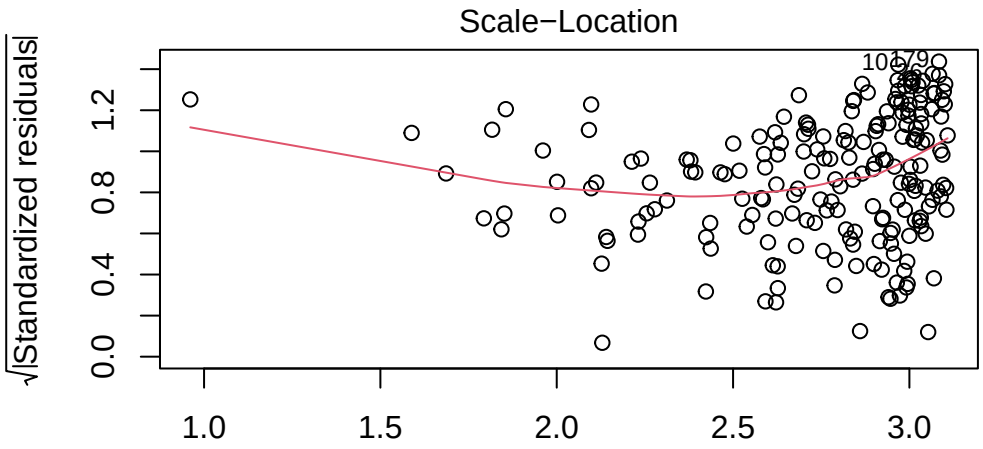
Question 2

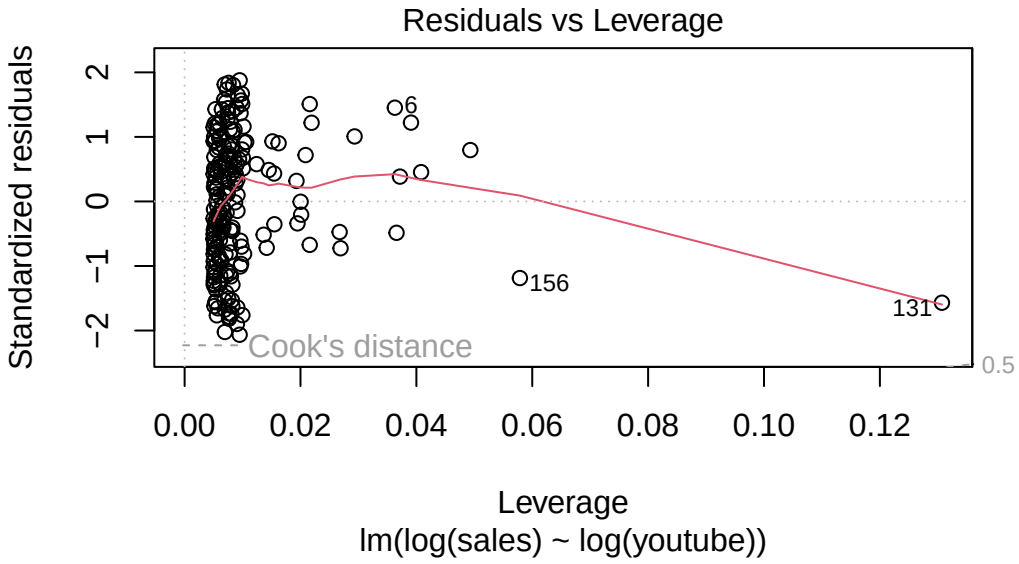
Run the diagnostic plots for the log model.

```
plot(lm.youtube.log)
```









What do you think of the diagnostic plots for the log model? Are there any violations of the log model? Why?

```
# Answer
# The diagnostic plots for the log model show an improvement in the linearity and equal varia
# However, there are still some deviations in the Q-Q plot, indicating slight non-normality.
# Overall, the log transformation appears to have improved the model fit but some minor issue
```