Residual Plots in R

You should have watched the video on checking regression assumptions. In the video you saw how residual plots are useful in verifying several of the requirements for a regression analysis. Here we will practice using R to generate those plots.

A Clock Example

Remember our clock dealer from last week? Imagine an antique clock dealer has collected data on recent auctions of grandfather clocks. The variables are `Price`: final selling price, `Bidders`: the number of bidders, `Age`: the age of the clock, `Temp`: the outside temperature the day of the auction, and `Condition`: the condition of the clock.

```
clocks <- read_csv("https://raw.githubusercontent.com/moravian-mspa/MGMT555/master/m6_Tutorial_Residual_
```

```
Rows: 32 Columns: 5
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): Condition
dbl (4): Age, Bidders, Price, Temp

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Generate the model

Recall from last time we decided the appropriate variables were `Age`, `Bidders`, and `Condition`.

```
clockModel <- lm(Price ~ Age + Bidders + Condition, data=clocks)
summary(clockModel)
```

```
Call:
lm(formula = Price ~ Age + Bidders + Condition, data = clocks)

Residuals:
    Min      1Q  Median      3Q     Max
-186.87  -69.19  -17.17   70.14  227.44

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    -401.09     403.27  -0.995  0.32877
Age               8.94       1.69   5.290 1.40e-05 ***
Bidders          63.66      13.00   4.898 4.01e-05 ***
ConditionFair  -295.63     135.83  -2.176  0.03844 *
ConditionGood  -253.31      75.87  -3.339  0.00247 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 114.3 on 27 degrees of freedom
Multiple R-squared:  0.9263,    Adjusted R-squared:  0.9154
```

```
F-statistic: 84.86 on 4 and 27 DF,  p-value: 6.914e-15
```

Checking assumptions

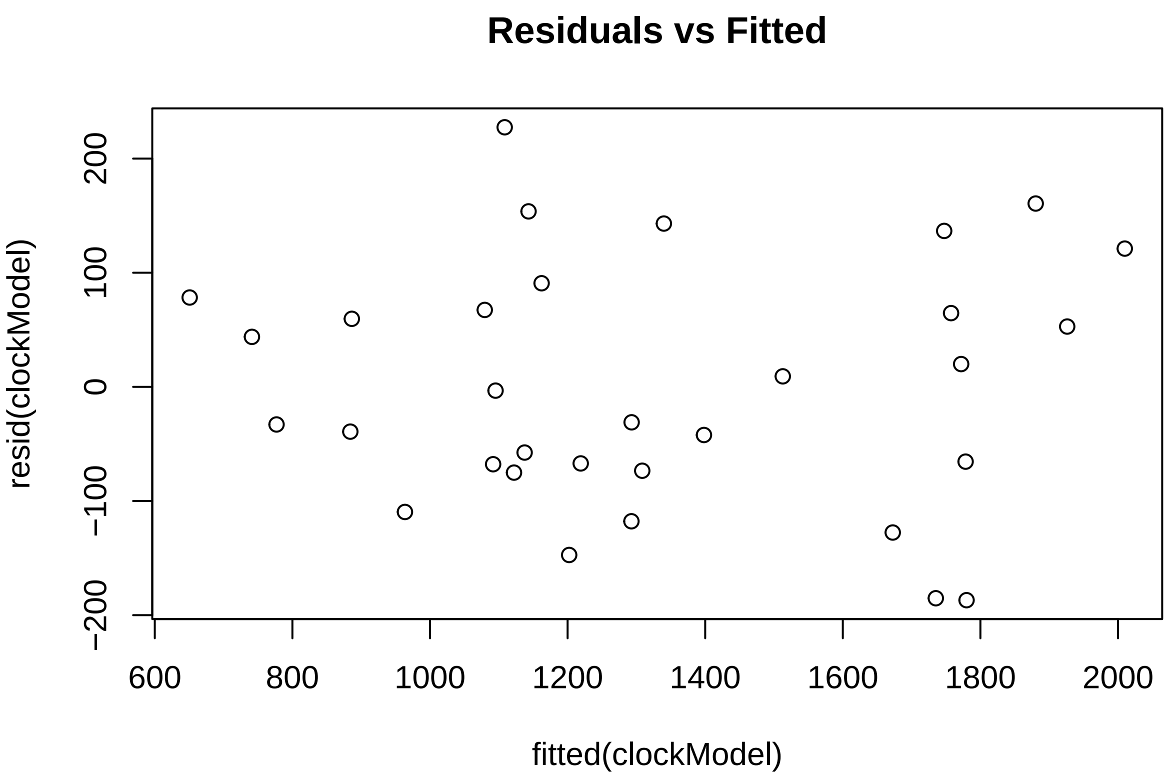
Recall the assumptions for a multiple linear regression are:

1. The relationship between the dependent variable and each predictor variable is linear
2. The errors (residuals) are independent
3. The errors (residuals) at each predictor value are normally distributed
4. The errors (residuals) have equal variance across predictors (homoscedasticity)

Residuals vs Fitted values for #1 and #4

We can check assumption #1 and #4 using a scatterplot with the fitted (ie predicted) values for each data point on the horizontal axis and the residual (ie error) associated with that data point on the vertical axis.

```
plot(resid(clockModel) ~ fitted(clockModel), main="Residuals vs Fitted")
```



If #1 is true, then we should see no clear pattern in this plot. That seems to be the case here.

If #4 is true, then we should see a roughly constant spread as we move left to right across the plot. That seems to be the case here.

Recall from the video, in this first residual plot we want to see no systematic pattern (this would indicate a non-linear relationship in the data). Also, we want a scatter plot with even variation throughout (to ensure homoscedacticity).

Checking independence of redsiduals for #2

Since we are not given any indication of what order this data was collected in, we cannot do a plot of residuals vs order. It seems plausible that these auctions were independent though, so we will proceed with the model.

Remember, the type of plot discussed in the video only makes sense if your data was collected in some sort of clear order. If there is no order then simply consider how the data was collected and ask if it seems plausible that the values are independent.

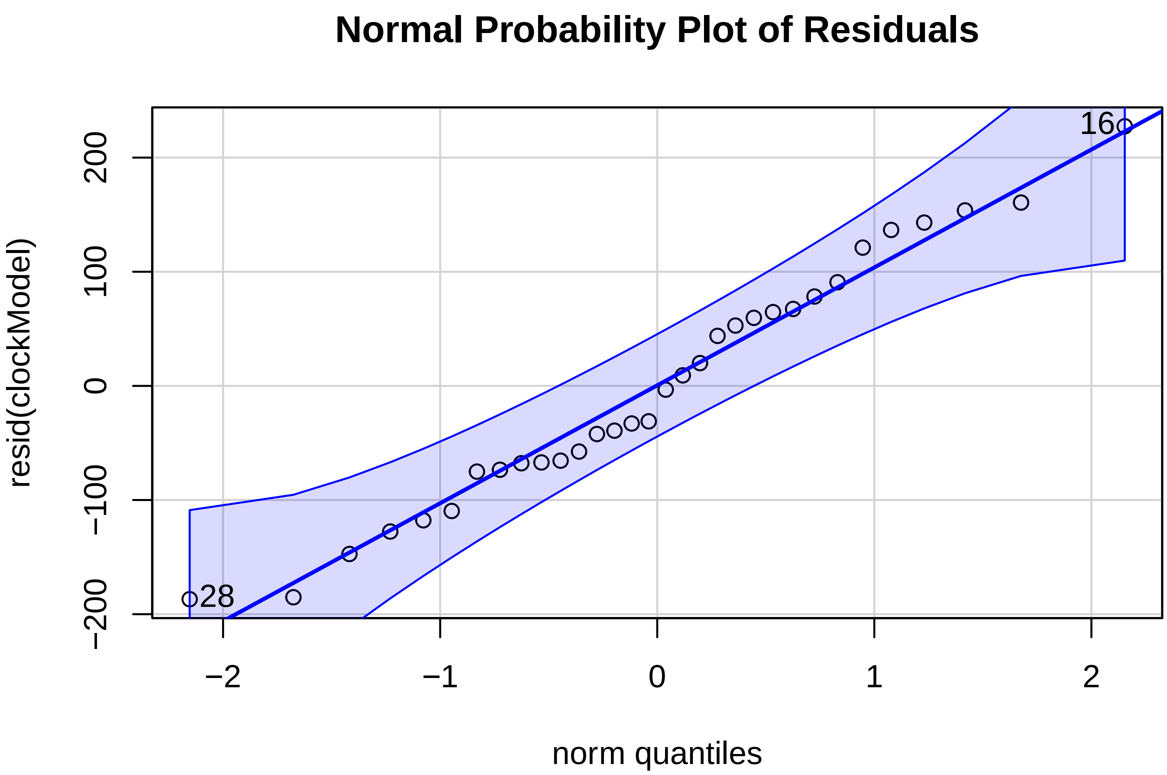
Normally distributed errors for #3

To test the assumption that the residuals are normally distributed, we can check a normal probability plot. Recall we did this earlier with simple linear regression and we loaded the `car` package.

```
library(car)
```

Once you have the car package loaded successfully, the command below produces the normal quantile plot for the residuals along with a reference line and confidence bands. We hope the points are close to the reference line and within the dashed lines. (note the capital P in the command name)

```
qqPlot(resid(clockModel), main="Normal Probability Plot of Residuals")
```




```
[1] 16 28
```

This is a good normal probability plot. All the points lie within the dashed confidence band. We can assume the residuals are normally distributed.

Summary

This model appears to satisfy all the assumptions of linear regression. I would be confident in using this model.

EPA Data

In a Week 5 tutorial we used the file "EPA gasoline rating 2019.csv"[1] contains data about 2019 model year vehicles collected by the Enivronmental Protection Agency along with the EPA miles per gallon fuel efficiency

[1] EPA (2019). Fuel Economy Data Set [Data File]. Accessed at https://www.fueleconomy.gov/feg/download.shtml

ration. This data set includes gasoline powered vehicles only.

```r
epa <- read_csv("https://raw.githubusercontent.com/moravian-mspa/MGMT555/master/m6_Tutorial_Residual_Pl
```

```
Rows: 2415 Columns: 17
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (9): Model, Trans, Drive, Cert Region, Stnd, Stnd Description, Underhood...
dbl (8): Displ, Cyl, Air Pollution Score, City MPG, Hwy MPG, Cmb MPG, Greenh...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

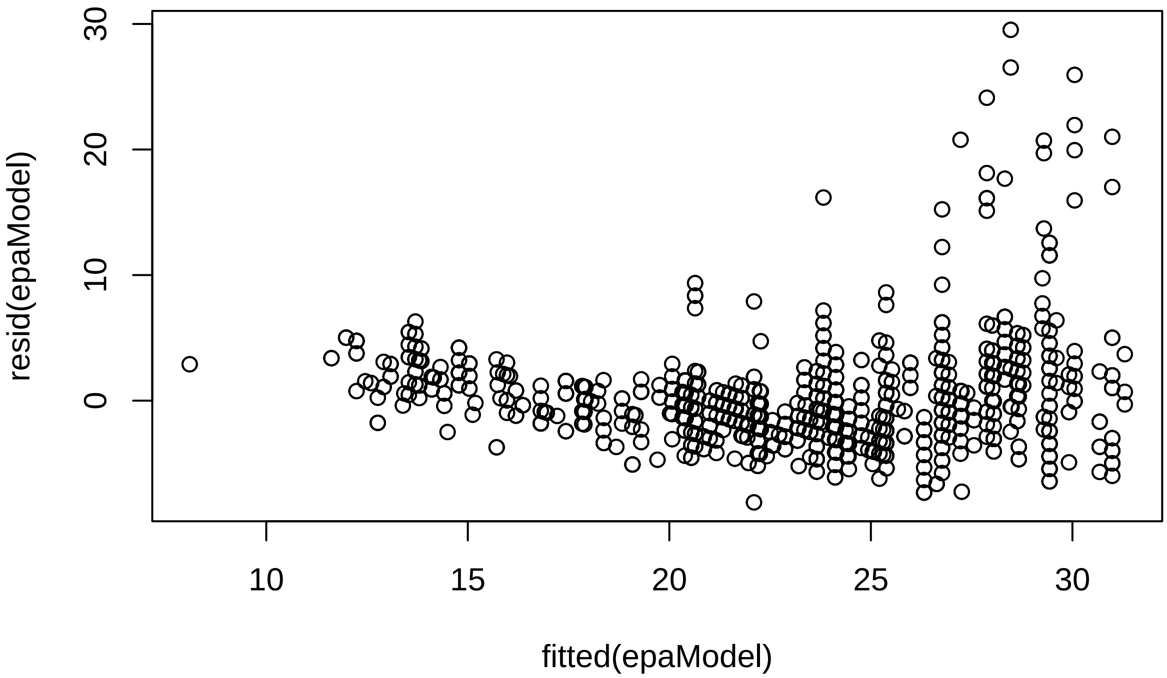
Let's try a relatively simple model using `Displ` and `Veh Class` to predict `Cmb MPG`.

```r
epaModel <- lm(`Cmb MPG` ~ Displ + `Veh Class`, data=epa)
summary(epaModel)
```

```
Call:
lm(formula = `Cmb MPG` ~ Displ + `Veh Class`, data = epa)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.1004 -2.3791 -0.4322  1.2878 29.5314 

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)    
(Intercept)                33.44658    0.35288  94.781  < 2e-16 ***
Displ                      -3.11126    0.07015 -44.354  < 2e-16 ***
`Veh Class`midsize car      2.20816    0.33655   6.561 6.52e-11 ***
`Veh Class`minivan         -1.06978    1.04282  -1.026 0.305066    
`Veh Class`pickup          -2.17479    0.39483  -5.508 4.01e-08 ***
`Veh Class`small car       -0.45678    0.30189  -1.513 0.130387    
`Veh Class`small SUV       -1.84497    0.33102  -5.574 2.77e-08 ***
`Veh Class`special purpose -3.87295    0.59900  -6.466 1.22e-10 ***
`Veh Class`standard SUV    -1.91775    0.37379  -5.131 3.12e-07 ***
`Veh Class`station wagon    0.82363    0.48700   1.691 0.090923 .  
`Veh Class`van             -6.74606    1.97023  -3.424 0.000627 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.904 on 2404 degrees of freedom
Multiple R-squared:  0.5748,	Adjusted R-squared:  0.5731 
F-statistic:   325 on 10 and 2404 DF,  p-value: < 2.2e-16
```

First a residual vs fitted plot.

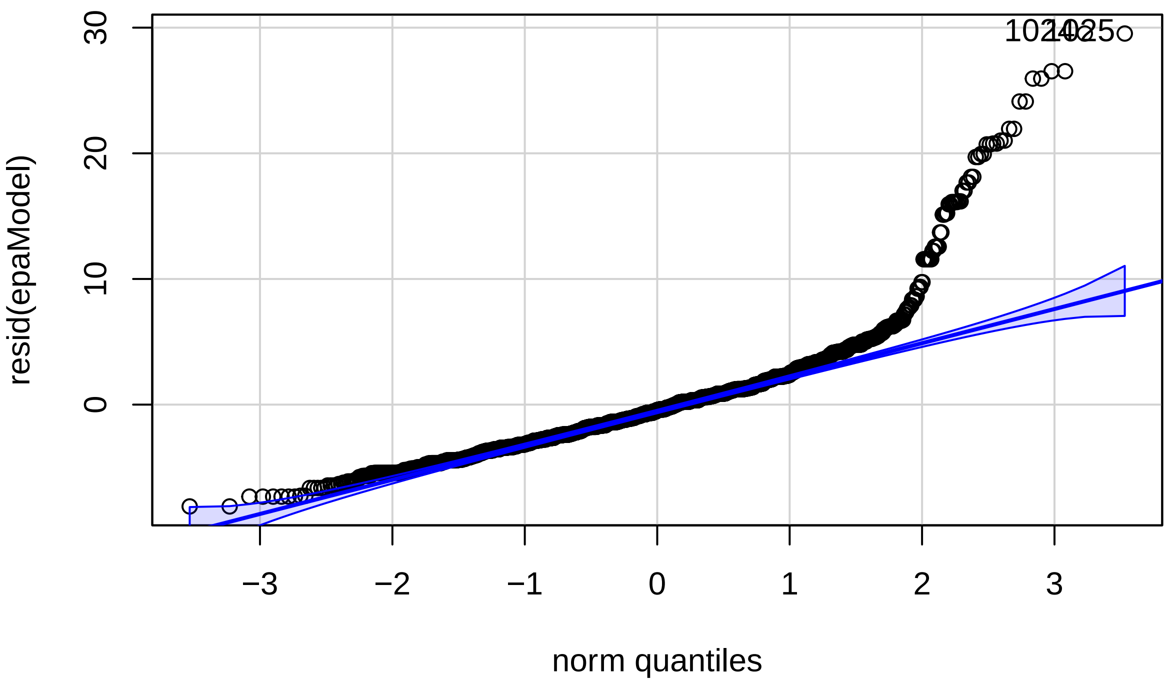
```r
plot(resid(epaModel) ~ fitted(epaModel))
```



That does not look very good. The variation in the residiuals changes quite a bit (heteroscedasticity) and we see a clear funnel effect in the plot. Assumption #4 is not met.

Now a normal probability plot.

```
qqPlot(resid(epaModel))
```




```
[1] 1024 1025
```

Yikes. This is not a good plot at all. A significant portion of the data on the right leaves the confidence band. The errors are not normally distributed. Assumption #3 is not met.

Note I would not be overly concerned about the left side of the plot, but the right side is really just terrible.

Overall this model has too many problems. I would not recommend using this model as it is. It will likely not accurately predict our dependent variable.