OLS Diagnostics

Irfan Kanat

August 13, 2017

After the first two learning activities we now know how to fit an ordinary least squares regression model with or without interaction effects in R. Next step is to verify our work and make sure the results we obtained are indeed reliable.

I will again not go into details here, but the foundation of what we do here is founded on basic assumptions of regression.

Some of these tests are more important than others (as I will note), the reason I am covering these tests is to teach you how to conduct these tests in case you need them.

Linearity

We assume there is a linear relation between independent and dependent variables. The easiest way is to investigate some scatterplots. As with any visual inspection, you will notice that this is a subjective process. It is so subjective, beyond initially exploring your data to see what kind of relations to fit, this assumption is rarely questioned. We often take it for granted that there is some linear relation.

Let me show you some examples that are in violation of the linear relationship first:

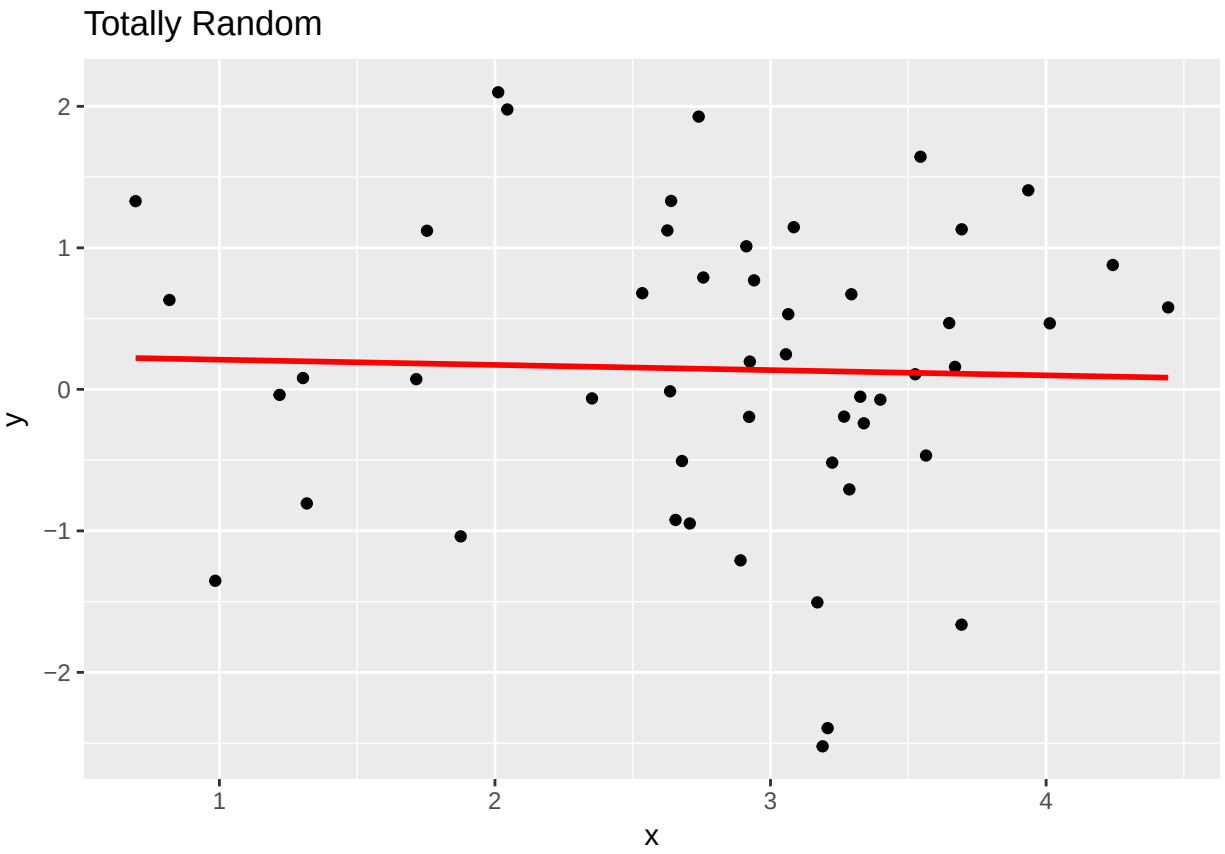
No Relationship

Sometimes there is no relation between two variables. In those cases, we would have very little correlation (about .05 is pretty small) between two variables. A regression line fit into such a relation will be almost flat. You can see an example of two random variables below.

```
set.seed(201)
# Totally Random
random <- data.frame(x = rnorm(50, mean = 3, sd = 1),
                     y = rnorm(50, mean = 0, sd = 1))
cor(random$x, random$y) # Low Correlation - Means no relation
```

```
## [1] -0.03068205
```

```
# Plot it out
qplot(data = random, x = x, y = y) + ggtitle("Totally Random") +
  geom_smooth(method = "lm", color = "red", se = F)
```



Non-Linear Relationship

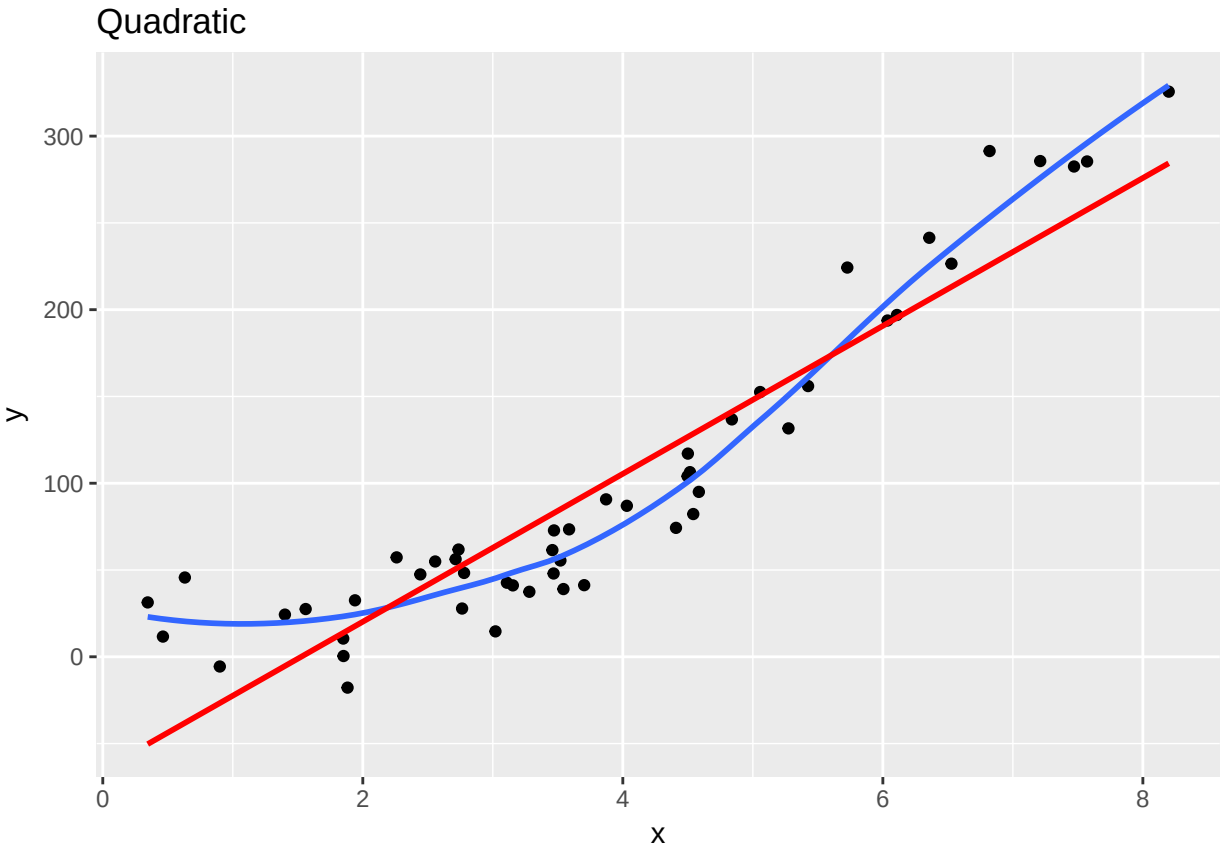
When the two variables are related to each other in non-linear ways you would find that a curve (blue) better fits the data then a line (red). Below is a quadratic relationship.

```
set.seed(2018)
# Quadratic
quadratic <- data.frame(x = rnorm(50, mean = 4, sd = 2))
quadratic$y <- quadratic$x + 5 * quadratic$x^2 + rnorm(50, mean = 0, sd = 20)
# High Correlation - Means the variables are related
cor(quadratic$x, quadratic$y)
```

```
## [1] 0.9258885
```

```
qplot(data = quadratic, x = x, y = y) + ggtitle("Quadratic") +
  geom_smooth(se = F) + geom_smooth(method = "lm", color = "red", se = F)
```

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



While it is true that non linearity can be a problem, we often can ignore small amounts of it. The above plot has a moderate ammount of non linearity in it.

To counteract this kind of relation you can add higher order variations of a variable to the model (squares, cubes, etc). In all honesty, you most probably won't need to to such a thing unless you are hypothesizing a U shaped relation where the relation first goes up and then declines (or vice versa).

The Way It Should Be: Linear

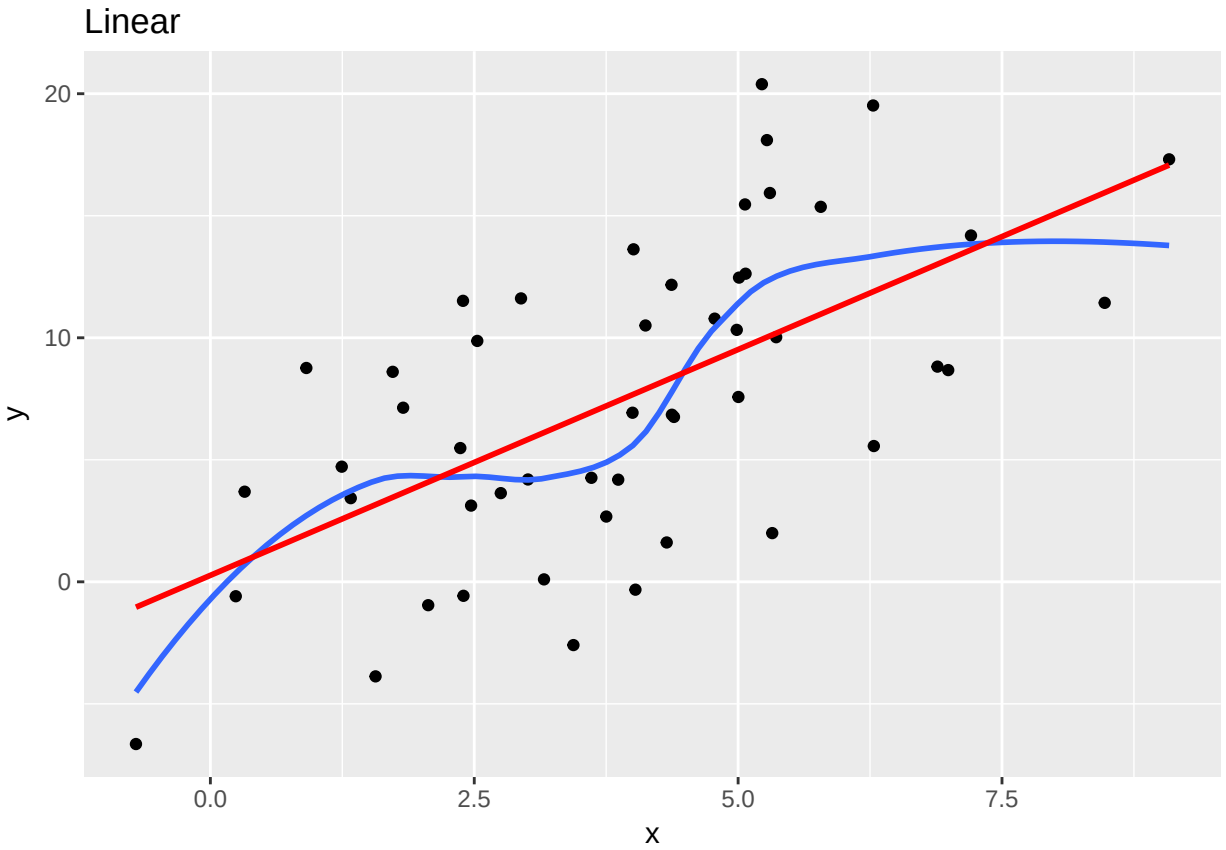
If there is a linear relationship we would expect to see the scatter plot to have slope, the points should either climb or decline with each other. If a regression line is fit on two linearly related variables, we expect the line to go through the middle of the scatter plot with dot's spread equally above and below the line.

```
set.seed(2015)
# Linear
linear <- data.frame(x = rnorm(50, mean = 4, sd = 2))
linear$y <- 2 * linear$x + rnorm(50, mean = 0, sd = 5)
cor(linear$x, linear$y) # Sizable Correlation
```

```
## [1] 0.6160474
```

```
qplot(data = linear, x = x, y = y) + ggtitle("Linear") +
  geom_smooth(se = F) + geom_smooth(method = "lm", color = "red", se = F)
```

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



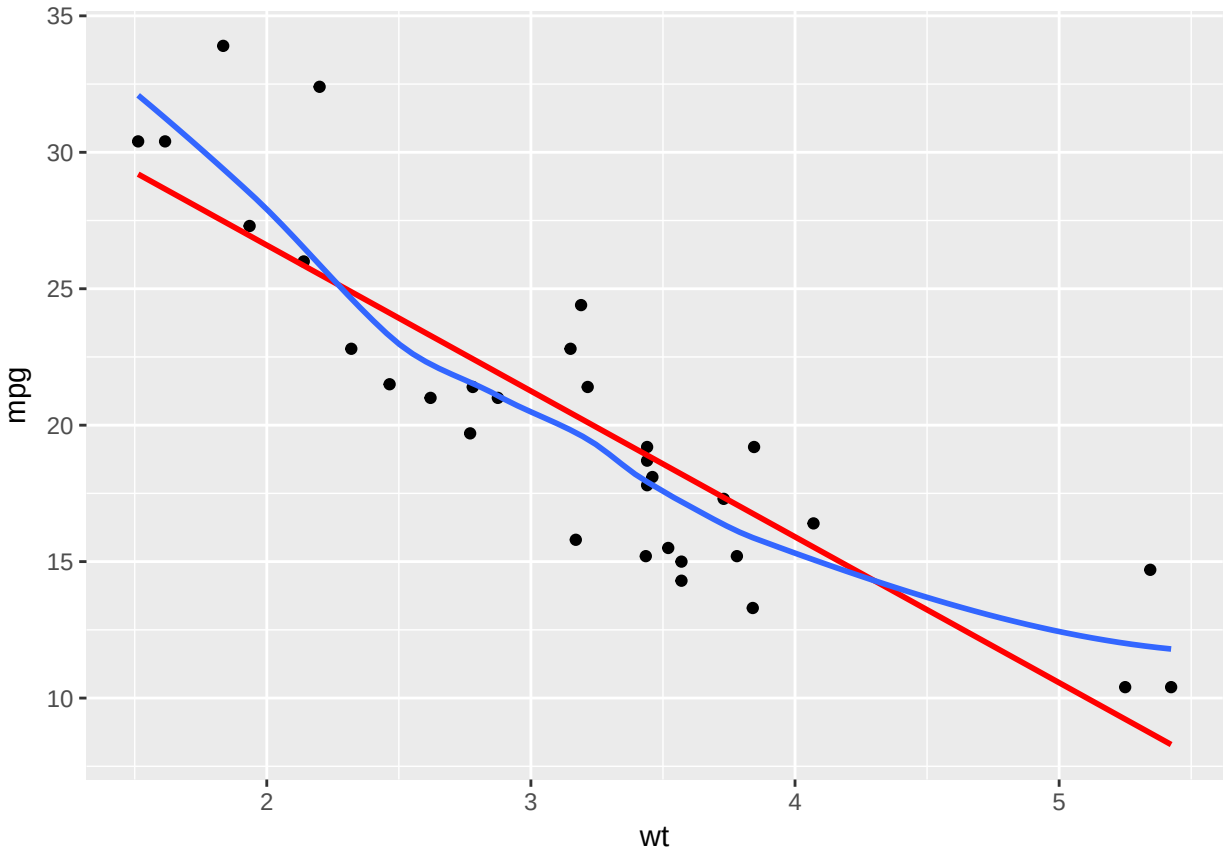
Very often you will see real data looks a bit weird. You may find it interesting to see what is deemed acceptable linear relationship.

```
cor(mtcars$mpg, mtcars$wt) # High correlation
```

```
## [1] -0.8676594
```

```
qplot(data = mtcars, x = wt, y = mpg) +
  geom_smooth(method = "lm", se = F, color = "red") + geom_smooth(se = F)
```

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



Many would look at this and say there is a linear relationship. Indeed, given the sample size, going quadratic may end up overfitting the model. So be a little open minded in your evaluation.

Normality of Residuals

A common mistake is to assume dependent or independent variables need to be normally distributed. Just think of all binary variables we use every day. There is simply no way to normally distribute a binary variable. In truth, it is the residuals of fitted models that need to be normally distributed. Remember, the goal in regression is to minimize residuals. We expect the resulting residuals to be normally distributed.

What happens if this assumption is violated? Exact result will change from case to case, but most likely, your predictions will not be reliable across the board. Still, non-normal residuals is not the end of the world. If you understand the nature of violation you can take steps to rectify the issue.

There are many visual inspection techniques. I prefer statistical tests to visual inspections as I find them more objective.

A Bad Example

I will inspect the mtcars_lm_0 for a bad example where the assumption is violated. In our case a bad example provides a great learning opportunity (so it is a great example).

Shapiro Wilk's Test for Normality

One common test for normality is Shapiro Wilk Normality Test. We conduct this test with the shapiro.test() function.

```
shapiro.test(mtcars_lm_0$residuals)
```

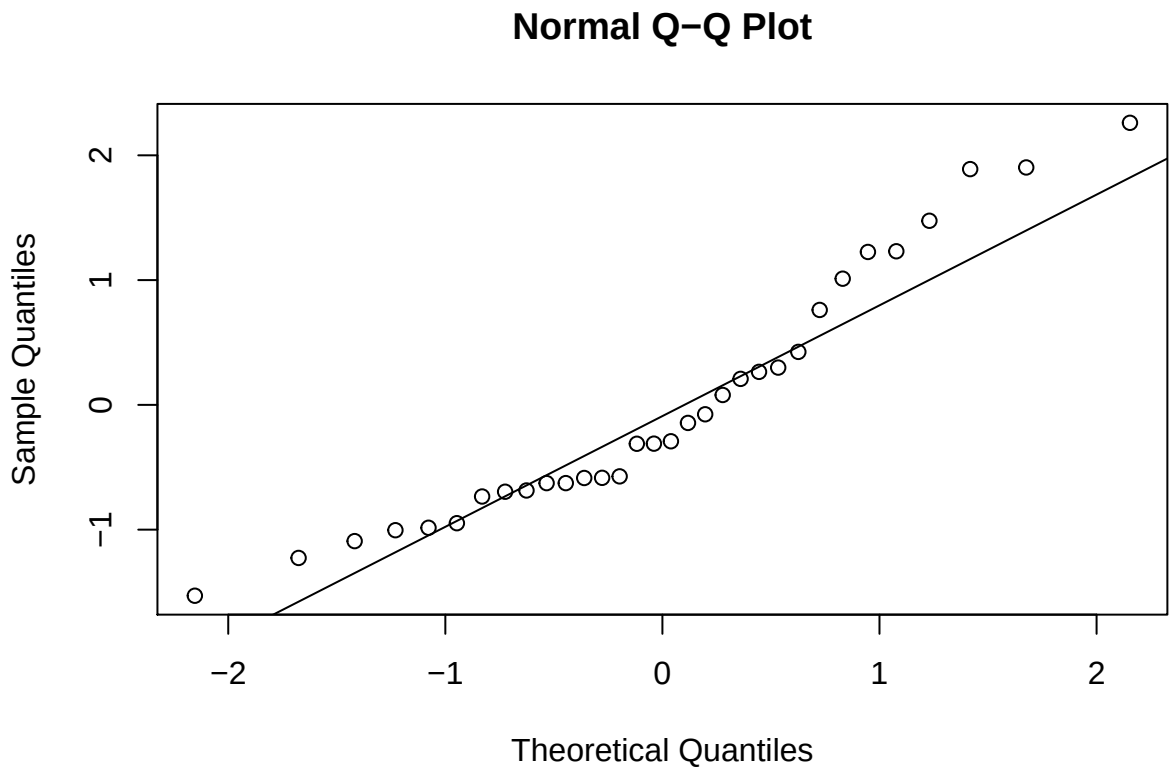
```
##
##  Shapiro-Wilk normality test
##
## data:  mtcars_lm_0$residuals
## W = 0.9271, p-value = 0.03255
```

In Shapiro-Wilk test a small p value means non-normal distribution. Shapiro-Wilk's test results indicate the residuals are not normally distributed.

Visual Inspection of Residuals

Let us see how a non-normal set of residuals look. Below you will see a qqplot. Basically this plots out theoretical (expected) quantiles and sample (actual) quantiles of residuals side by side. We want the sample quantiles to be equal (or very close) to theoretical quantiles. When the theoretical and sample quantiles are equal, you will get a line at 45 degrees.

```
# R's basic plotting functionality is really easy to use
# Remember the scale function from Module 2. We standardize residuals
qqnorm(scale(mtcars_lm_0$residuals))
qqline(scale(mtcars_lm_0$residuals)) # Adds the line
```

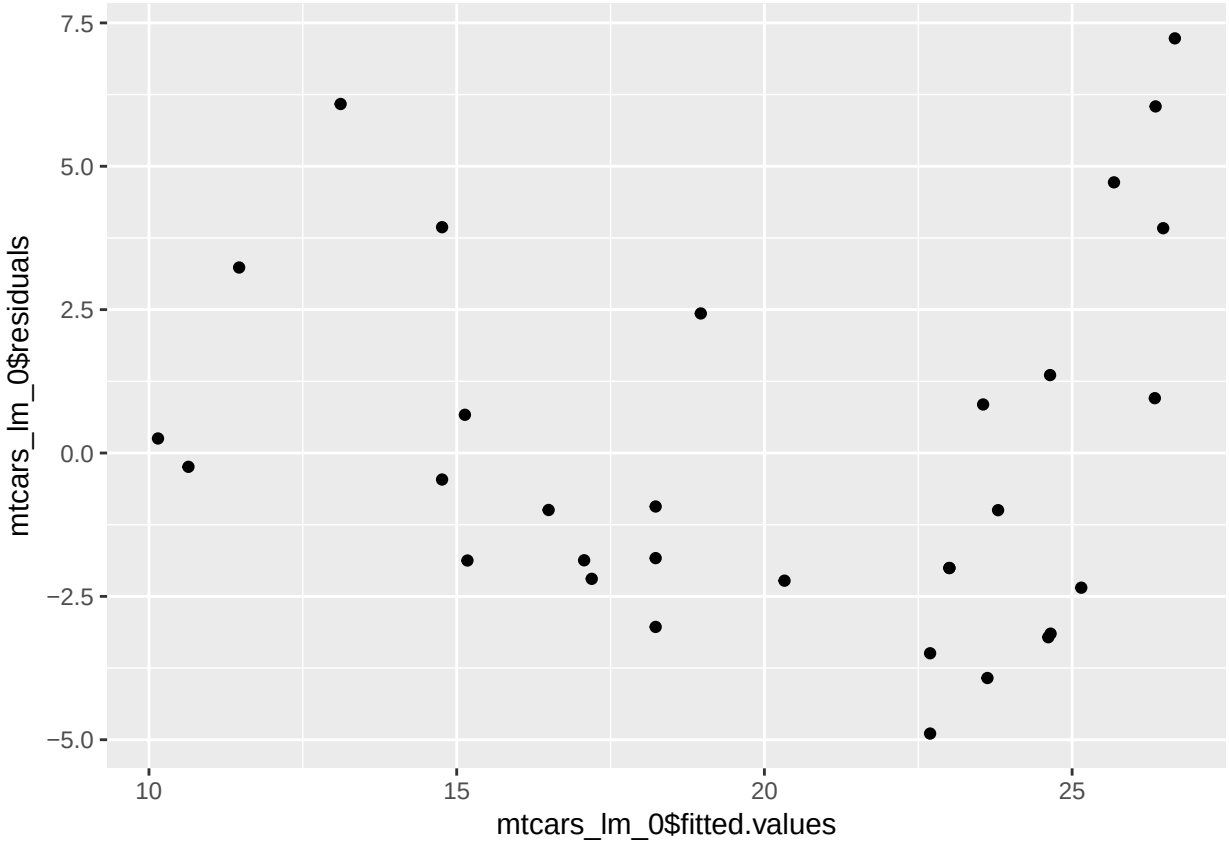



```
# With ggplot2 it is a bit harder but looks more pleasing:
# ggplot(mtcars_lm_0, aes(qqnorm(.stdresid)[[1]], .stdresid)) + geom_point()# Uses the model directly b
```

As you can see the theoretical quantiles and sample quantiles do not match. Negative second theoretical quantile corresponds to about the negative first quantile for example. Now you know how non-normal residuals look on a qq plot.

Another plot you may want to examine is fitted values against residuals.

```
# This is easy to remember
qplot(x = mtcars_lm_0$fitted.values, y = mtcars_lm_0$residuals)
```




```
# Uses model directly but requires you to remember how to call fitted vs residuals
# ggplot(mtcars_lm_0, aes(.fitted, .resid))+geom_point()
```

In this plot you want to see randomly distributed residuals. If there is a systematic pattern to the model, the non-normality assumption is violated. Looking at this plot I can tell the model systematicaly underestimates gas milage for estimates between 10 - 15 (almost all residuals are positive) and it overestimates gas milage for estimates between 15-20 (almost all residuals are negative). If the model worked, we would see residuals evenly distributed through out.

A Good Example

Now let us look at how data looks with a model fit to a simulated dataset. For this, I will use develop_lm_0.

Shapiro Wilk's Test for Normality

Take a look at the results of Shapiro Wilk's test for normality.

```
shapiro.test(develop_lm_0$residuals)
```

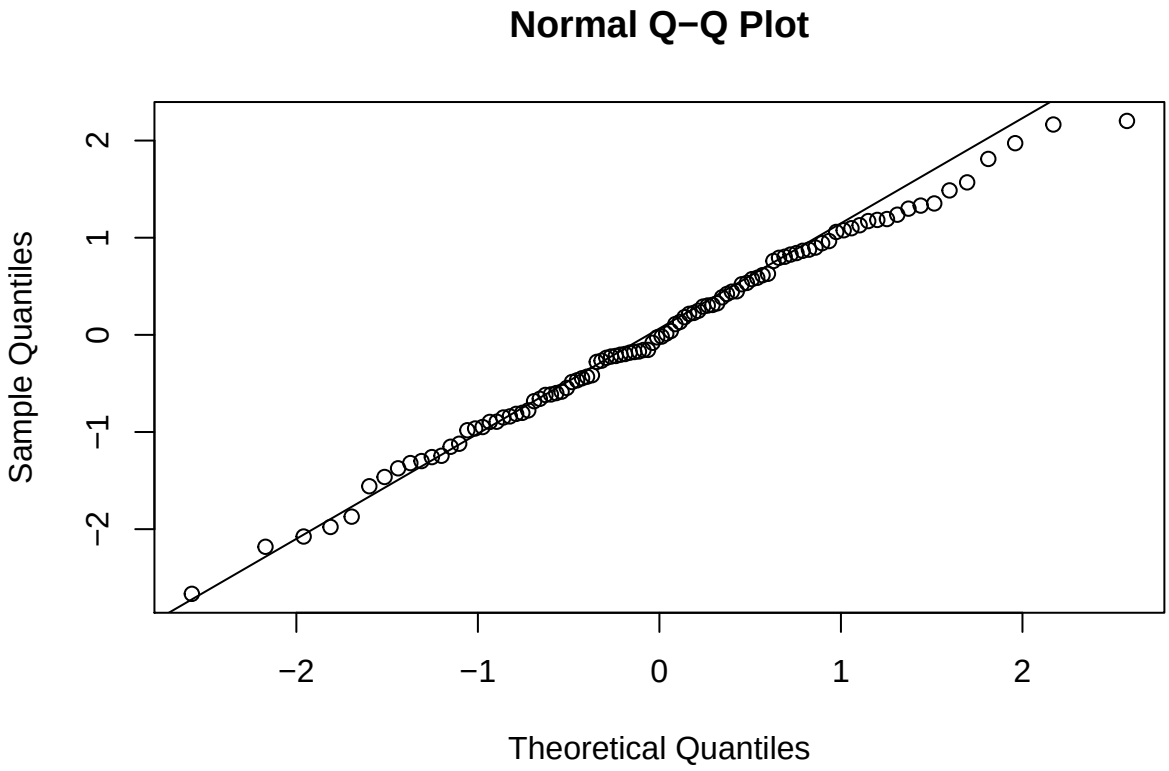
```
##
##  Shapiro-Wilk normality test
##
## data:  develop_lm_0$residuals
## W = 0.99359, p-value = 0.921
```

Just WOW! Look at how normal the test results are.

Visual Inspection of Residuals

Let us see how an almost perfectly normal set of residuals look.

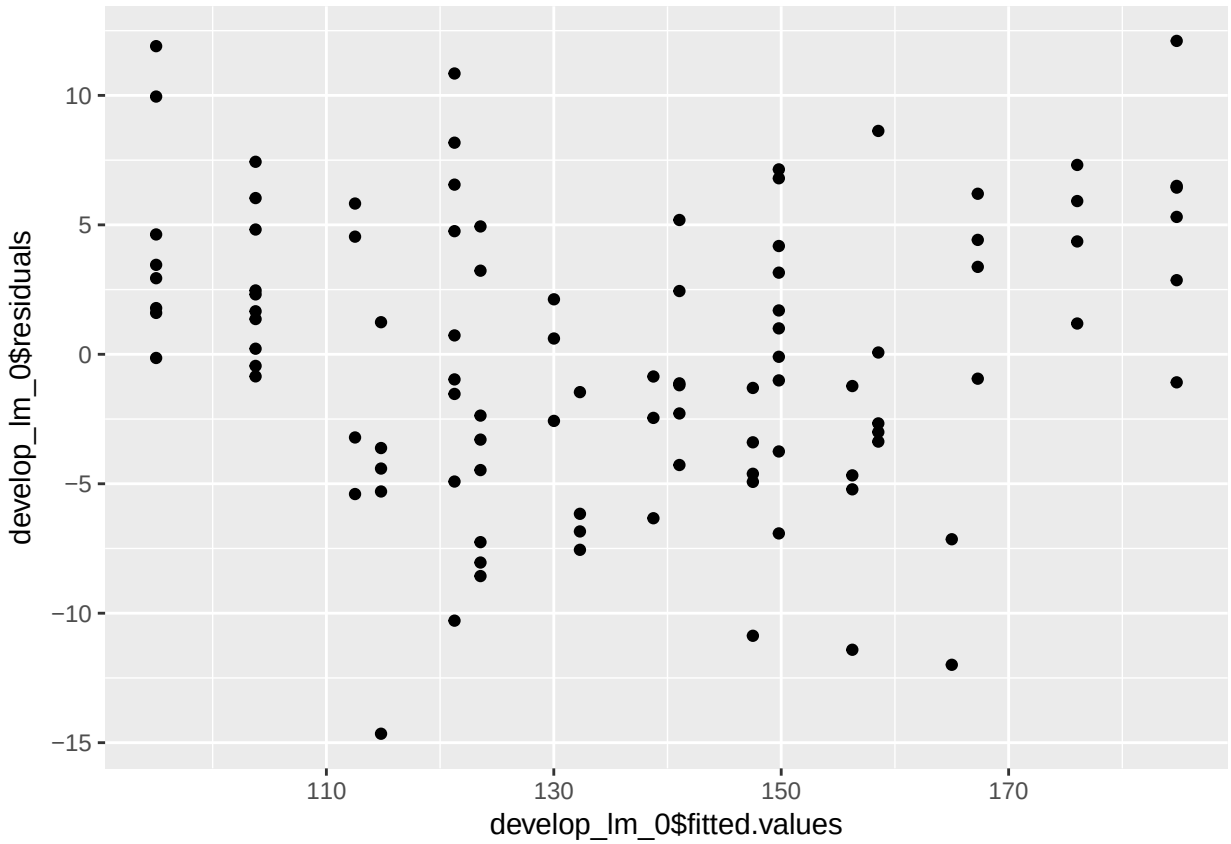
```
qqnorm(scale(develop_lm_0$residuals))
qqline(scale(develop_lm_0$residuals))
```



This is as good as it gets...

Let us look at plot of residuals against fitted values.

```
qplot(x = develop_lm_0$fitted.values, y = develop_lm_0$residuals)
```



Looks random to me...

What Is the Effect of Non-Normal Residuals?

Depends on your goals in analysis. Obviously your predictions would be off. Just look at the fitted vs residuals plot for mtcars_lm_0. There are systematic problems with our predicted values.

But if you care about statistical significance, the effects would be very small. When Regression was developed the computational power and sample sizes were quite limited. Thanks to recent advances, scientists empirically investigated effects of non-normality and found that for sample sizes above a 30 and more, the effect of non-normality is limited. The significance figures should be reliable.

What to Do In Case of Violations?

Sometimes the reason for systematic biases in estimation is an ommited variable. In those cases including a variable may reduce the problem.

Try and detect outliers. Removing outliers can improve the situation quite a bit.

Another commonly used method is to transform variables that go into the model. Very often if you deal with an eggregiously non-normal variable (wealth, income, prices...) some transformation will do a world of good.

Multicollinearity

Multicollinearity is when there are sizable correlations between multiple variables in your model. Simple correlation measures do not often capture multicollinearity. Still, if you see correlations between independent variables that is in excess of 70, you may consider dropping one of the variables.

Unlike non-normality, multicollinearity does not effect the predictions. Your predictions will be just as good, but your statistical significance estimates will be off. You may make a type 2 error in rejecting a variable that was in fact significant. So if all you care about is predictions, you can safely ignore multicollinearity.

Value Inflation Factor

Tolerance (Tol) or Value Inflation Factor (VIF) are commonly used to detect multicollinearity. Both are closely related to each other, in this case I will use VIF as the heuristic for VIF is easier to remember.

We will use vif() function from car package to detect multicollinearity (You do know how to obtain and activate packages).

```
vif(develop_lm_0)
```

```
##   gender      age
## 1.025011 1.025011
```

As long as VIF value is below 10 we need not worry about multicollinearity.

A simple exercise, run the vif() function for develop_lm_1 and speculate on why the VIF values changed the way they changed.

What is the Effect of Multicollinearity

The significance tests for coefficients will be off. You may end up rejecting a variable that was significant.

Your predictions will not be harmed.

What to do About Multicollinearity

You can drop the offending variables.

Remember our discussion of fitting models of increasing complexity? The main reason is to be able to capture change in significance of variables due to multicollinearity. In case of interactions for example, we often don't worry much. If a main effect turns insignificant, we would say the main effect was mediated by the interaction.

Autocorrelation

Another key assumption of regression is the independence of error terms (residuals). We expect there won't be a significant relation between residuals. This assumption is often violated in time series data.

A canonical example is stock prices. One day's price is correlated to the next day's price. Unless the model takes out these relations (perhaps a correlation structure in a GMM for example), the error terms will be correlated.

One way to test for this is the Durbin Watson Test.

```
durbinWatsonTest(develop_lm_0)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1      -0.1569891       2.30432   0.134
##  Alternative hypothesis: rho != 0
```

The alternative hypotheses is that there is dependency in error terms, so if p value is below 0.05 we suspect autocorrelation.

What is the Effect of Autocorrelation?

Significance tests would be impacted as the variance in error term will be higher.

What can be Done?

Fit a time series model that accomodates a correlation structure like GMM.

Heteroskedasticity - Homoskedasticity - Or in Lay Statistician's Terms: Non-Constant Variance

We assume the error terms variance will be constant (homoskedasticity) accross the board. If the residuals have greater (smaller) variance for certain estimates, this means we have heteroskedasticity.

Statistical Tests

We often use a Breusch-Pagan test for NCV.

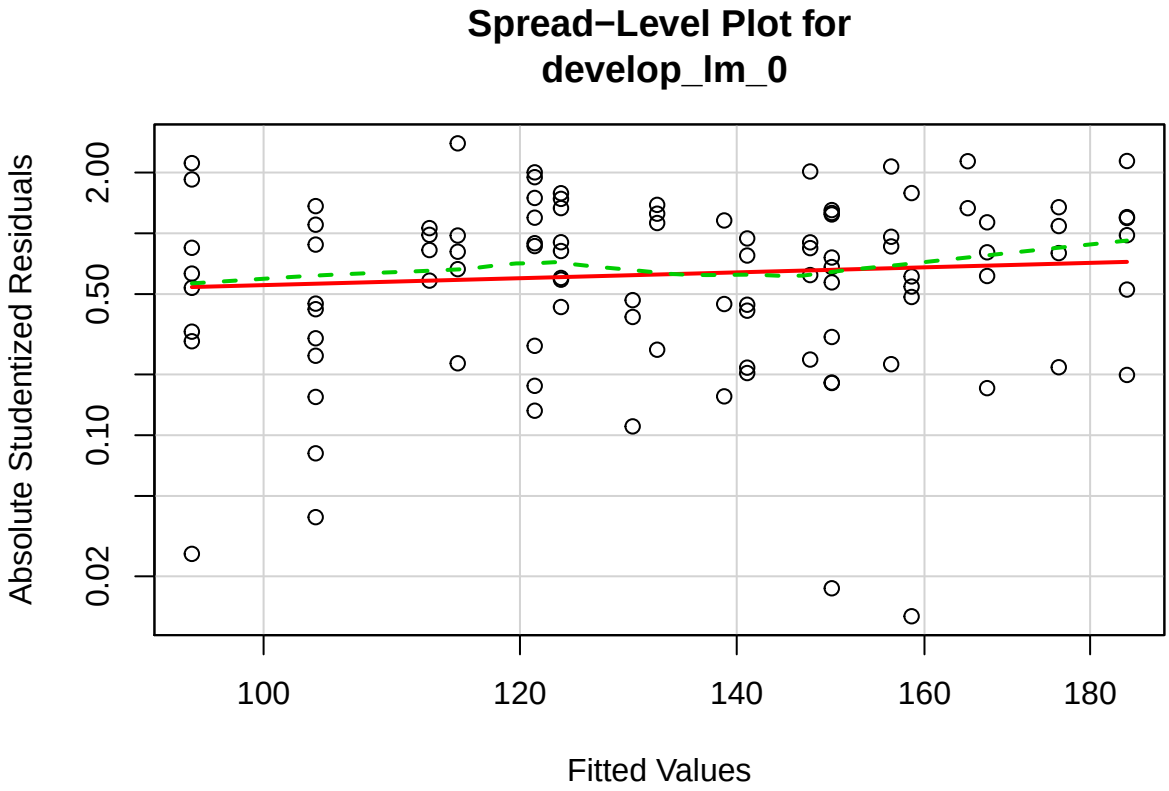
```
ncvTest(develop_lm_0)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.1521458    Df = 1     p = 0.6964931
```

Null hypothesis for BP test is homoskedasticity. In this case we fail to reject null. Meaning the model does not suffer from non-constant variance.

Visual Inspection

```
spreadLevelPlot(develop_lm_0)
```




```
##
## Suggested power transformation:  0.5684945
```

As with others, for this test we want the trend line to be as close to flat as possible. Furthermore you want the scatter plot to be randomly distributed. If you see the scatter plot getting narrower or wider going in one direction or another that would mean non constant variance.

Here is a simple exercise, conduct Breusch Pagan test for mtcars_lm_0 and report results. Compare the spread level plot between develop_lm_0 and mtcars_lm_0

What is the Effect of Heteroskedasticity

If you have heteroskedasticity in your residuals, that means your model will be more reliable for certain values of estimated values (where variance is smaller) and less reliable for other values.

What can be Done?

Fit a weighted regression.

What Next

Next we will learn about outlier detection.

Solutions to Exercises

1 - Run the vif() function for develop_lm_1 and speculate on why the VIF values changed the way they changed.

```
vif(develop_lm_1)
```

```
##     gender        age gender:age
##  30.306214   2.175625  33.245220
```

Interaction term is inevitably correlated to other two independent variables.

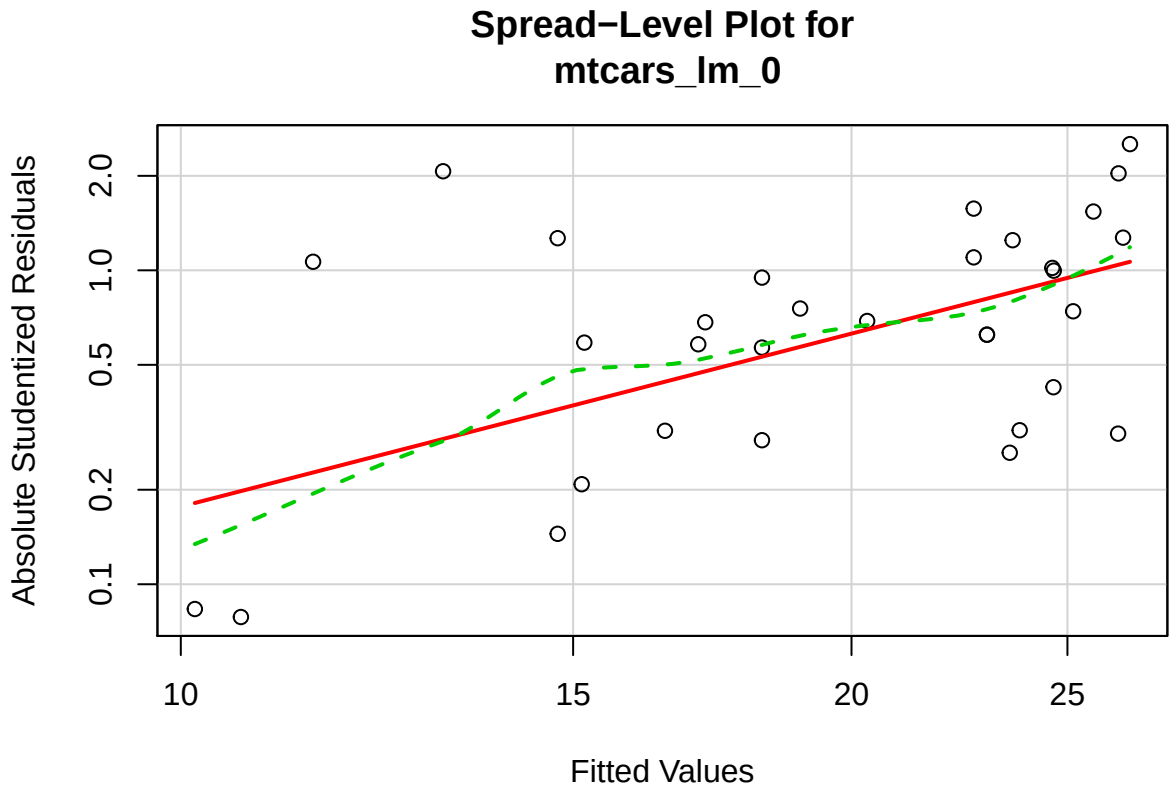
2 - Conduct Breusch Pagan test for mtcars_lm_0 and report results. Compare the spread level plot between develop_lm_0 and mtcars_lm_0

```
ncvTest(mtcars_lm_0)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 2.233958    Df = 1     p = 0.1350076
```

P value is above the cut off of 0.05 we fail to reject the homoskedasticity. Meaning there is no significant heteroskedasticity in the model.

```
spreadLevelPlot(mtcars_lm_0)
```




```
##
## Suggested power transformation:  -0.8308542
```

Compared to develop_lm_0, the mtcars_lm_0 shows more signs of heteroskedasticity (trend line is not flat and there appears to be a trend in the plot) still, we know from BP test that the heteroskedasticity is not worrisome.