Stat 6021 R Tutorial: Assessing Regression Assumptions

The simple linear regression model involves several assumptions. Among them are:

1. That $E(y|x)$, the mean value of y, is a straight-line function of x.
2. That the errors, ϵ_i, have constant variance. That is, the variation in the errors is theoretically the same regardless of the value of x or $\hat{y}$.
3. The errors have mean 0.
4. The errors are independent.
5. The errors have a normal distribution.

To assess assumptions 1 to 3, we can examine scatterplots of

- y versus x.
- residuals versus fitted values, $\hat{y}$, or x.

Assumption 4 is assessed with an autocorrelation (ACF) plot of the residuals. Assumption 5 is assessed with a normal probability plot, and is considered the least crucial of the assumptions. We will see how to generate the relevant graphical displays to help us assess whether the assumptions are met.

For this example, we will use the dataset "windmill.txt". This is Example 5.2 from your textbook. The researcher is investigating the use of a windmill to generate electricity. He has collected data on the DC output from his windmill and the corresponding wind velocity.

```
data <- read.table("windmill.txt", header=TRUE ,sep="")
attach(data)
```

Next, regress *output* against *wind*. We should always check if the regression assumptions are met.

1. One plot we could use is a scatterplot of *output* against *wind*. A useful tool would be to overlay the estimated regression line to the scatterplot.

```
result<-lm(output~wind)
plot(wind, output, main="Plot of DC Output against Wind Velocity")
abline(result,col="red")
```

The function `abline()` overlays a line to the plot. In this case, it overlays the estimated regression line from `result`. What are the features to look out for in a scatterplot of the response against the predictor?

2. It is usually easier to assess the regression assumptions using a residual plot (residuals plotted against either the fitted values or the predictor).

```
plot(result$fitted.values,result$residuals, main="Plot of Residuals
against Fitted Values")
abline(h=0,col="red")
```

As a visual aid, I recommend overlaying a horizontal line where the residual is 0. What are the features to look out for in a residual plot? Based on this plot, what will you say about the regression assumptions?

3. To assess independence of errors, we examine an autocorrelation (ACF) plot of the residuals.

```
acf(result$residuals, main="ACF of Residuals")
```

What does this plot tell us?

4. The last thing we need to check is the normality assumption. We use a QQ plot for this.

```
qqnorm(result$residuals)
qqline(result$residuals, col="red")
```

Based on this plot, what will you say about assumption regarding normality of error terms?

5. When the assumptions are not met, some remedial measures we could consider using would be to transform the response variable and/or transform the predictor variable. From the scatterplot and residual plot in parts 1 and 2: the linearity assumption is not met, while the constant variance assumption appears to be met. Thus, we would consider transforming the predictor variable. Based on the scatterplot, an inverse transformation appears to be reasonable.

```
inv.wind<-1/wind
result.inv<-lm(output ~ inv.wind)
```

6. After transforming the predictor and fitting the linear regression using the transformed predictor and the response variable, we need to recheck all the regression assumptions. Generate the scatterplot, residual plot, ACF plot of the residuals, and the QQ plot.

```
par(mfrow=c(2,2))
plot(inv.wind,output, main="Scatterplot after Transforming Predictor")
abline(result.inv,col="red")
plot(result.inv$fitted.values,result.inv$residuals, main="Plot of
residuals against fits")
abline(h=0,col="red")
acf(result.inv$residuals, main="ACF of Residuals")
qqnorm(result.inv$residuals)
qqline(result.inv$residuals, col="red")
```

What do these plots tell us about the assumptions after transforming the predictor variable?

7. When the variance is not constant, we consider transforming the response variable. An analytical method to see if the response variable needs to be transformed is to produce a plot of the profile log-likelihoods for the parameter, λ, of the Box-Cox power transformation, type

```
library(MASS)
boxcox(result.inv)
```

The `boxcox()` function is stored in the `MASS` library. You need to load this library to use this function. What do you notice? For the `boxcox()` function, there is an optional argument called `lambda` that allows us to change the range of λ for the Box-Cox transform, for example

```
boxcox(result.inv, lambda = seq(0.6, 1.6, 0.01))
```

Should we transform the response variable?