Lab # 10: Regression Analysis

Objectives:

1. Scatterplot with Regression Line
2. Linear Regression
3. Saving outputs and further analysis

Scatterplot

As mentioned in the lecture, the first thing one should try when analyzing two numerical variables trying to identify some correlation between them is a simple scatterplot. We mentioned this process in lab 2 but now it's a good time to review and expand on the ways of creating scatterplots.

The example we will use today is the dataset Iris.csv which you can find on our MOODLE page and contains the measurements of the sepal length and width of the beautiful wildflower **Iris Setosa** which you can see on the left. This is a circum-arctic wildflower found in Alaska, Northern USA, Russia, Northeastern Asia, China, Korea and Japan and it is considered an endangered species (critically threatened).

First let's import our dataset by using the by now well-known proc import command as follows:

```
PROC IMPORT OUT= Iris
            DATAFILE= "put the complete path here…\Iris.csv"
            DBMS=CSV REPLACE;
     GETNAMES=YES;
     DATAROW=2;
RUN;
```

Using proc print we can view the dataset and notice that it contains the length and the width of the sepals measured in centimeters.

```
PROC PRINT DATA = Iris;
RUN;
```

Recall, that creating a scatterplot can be done with the PROC GPLOT which we learned in lab 2 as follows:

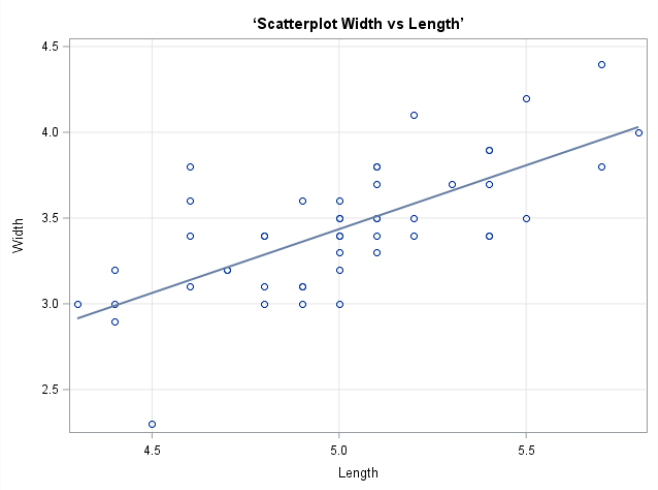
```
SYMBOL V=circle C=Blue;
Title "Scatterplot Width vs Length";
PROC GPLOT Data=Iris;
Plot Width*Length;
RUN;
```

The **SYMBOL** command instructs SAS to use blue circles to represent the points. The line **Plot Width*Length** says that the Width variable will be represented on the y axis and the variable Length on the x axis.

So Gplot creates beautiful graphs and it is very versatile, but the best way to create scatterplots is using the procedure **SGSCATTER**. This procedure also fits and outputs various lines to our scatterplot including a linear regression. The code is as follows:

```
Title "Scatterplot with regression line Width vs Length";
PROC SGSCATTER Data=Iris;
Plot= Width*Length/ Reg= (nogroup degree=1) grid;
RUN;
```

The command responsible for creating the regression is the: **Reg = (nogroup degree=1).** This instructs SAS to add a regression line of degree 1 (a straight line) and to not group the variables using some other variable. If for example you had two groups of observations (2 different species) SAS would fit a regression line to each group and present all of them in a nice graph. The command **grid** creates a grid of lines on the graph so that you can easily see the values of the points.



Linear Regression (Simple)

We will now try to create a simple linear regression model and analyze all the results. The procedure here is **PROC REG** used as follows:

```
Title "Simple Linear Regression";
PROC REG Data=Iris;
Model Width=Length;
RUN;
```

Just like in the PROC GLM the connection between the variables is denoted by the line

Model Width=Length; If no other variables are present, SAS will automatically try a linear regression between the two variables presented in the model. Let's now try to explain all the outputs starting by the analysis table:

Analysis of Variance					
Source	**DF**	**Sum of Squares**	**Mean Square**	**F Value**	**Pr > F**
Model	1	3.34457	3.34457	44.57	<.0001
Error	46	3.45210	0.07505		
Corrected Total	47	6.79667			

Notice that just like in class, the first table presented is an F statistic with the same set up as ANOVA. The hypotheses here are:

$$\boldsymbol{H_0: \beta_1 = 0}\ , \boldsymbol{H_A: \beta_1 \neq 0}$$

In other words we assume that the slope $\boldsymbol{\beta_1}$ in the formula

$$\boldsymbol{Width = \beta_0 + \beta_1 * Length + \varepsilon}$$

is zero, which would imply that Length does not help predict the Width. Since our p value is extremely small we reject the null hypothesis and confirm that length CAN be used to predict width and that the regression model might be a good predictor model for the relationship between the two.

The second table:

Root MSE	0.27394	**R-Square**	0.4921
Dependent Mean	3.44167	**Adj R-Sq**	0.4810
Coeff Var	7.95965		

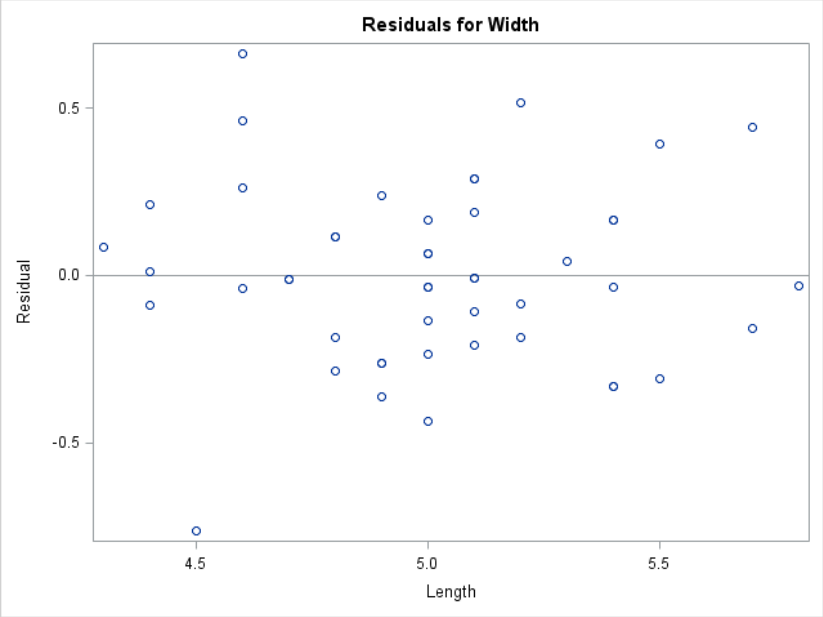
contains the Root MSE which is the square root of the Mean Square error we talked about in class. The parameter estimates can be found in the table:

Parameter Estimates					
Variable	**DF**	**Parameter Estimate**	**Standard Error**	**t Value**	**Pr > |t|**
Intercept	**1**	-0.28624	0.55981	-0.51	0.6116
Length	**1**	0.74434	0.11150	6.68	<.0001

Based on those results our model becomes:

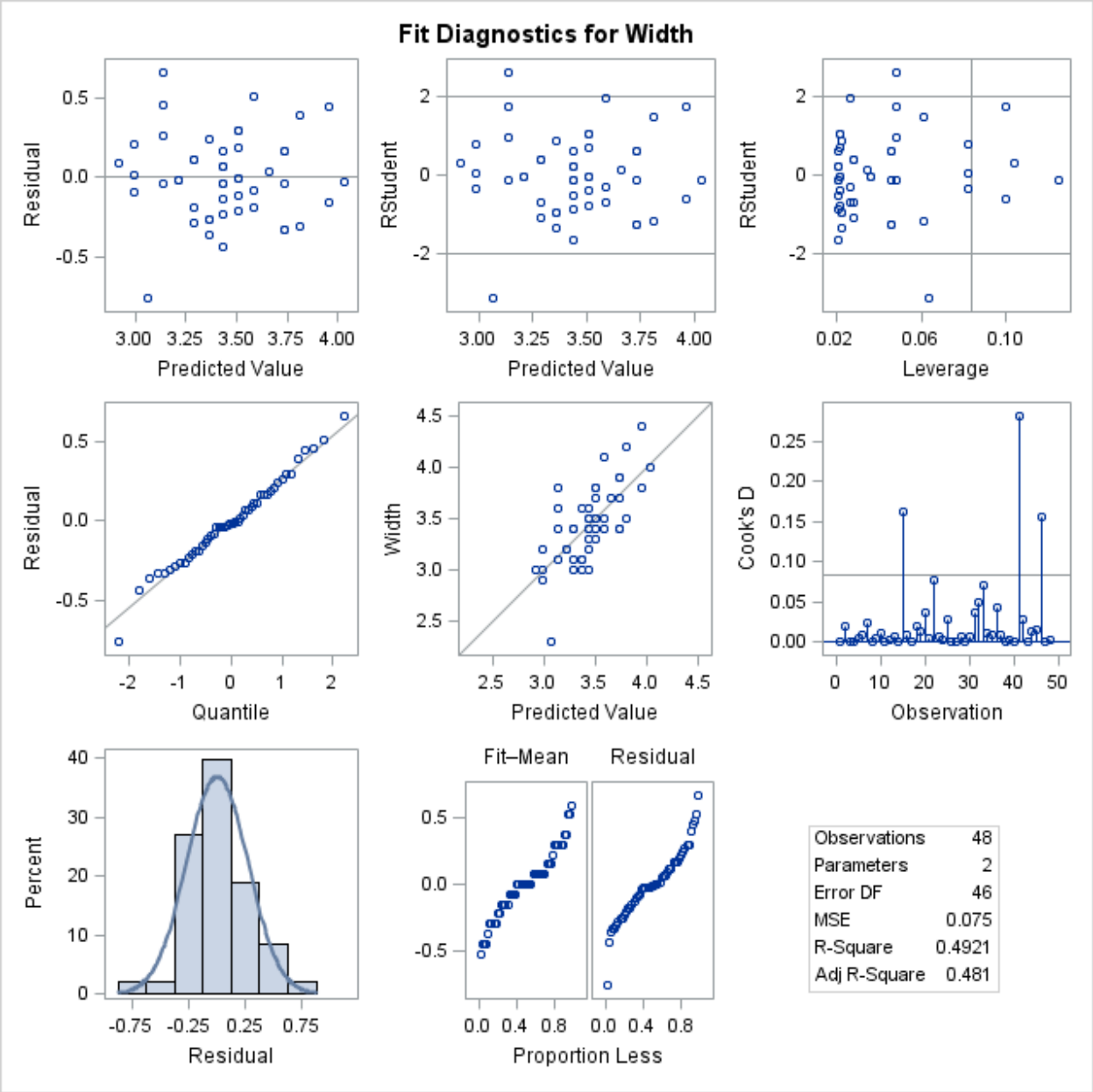
$$\boldsymbol{Width} = -0.28624 + 0.74434 * \boldsymbol{Length} + \boldsymbol{\varepsilon}$$

The program also gives us the residuals for each observation pair in a scatterplot.

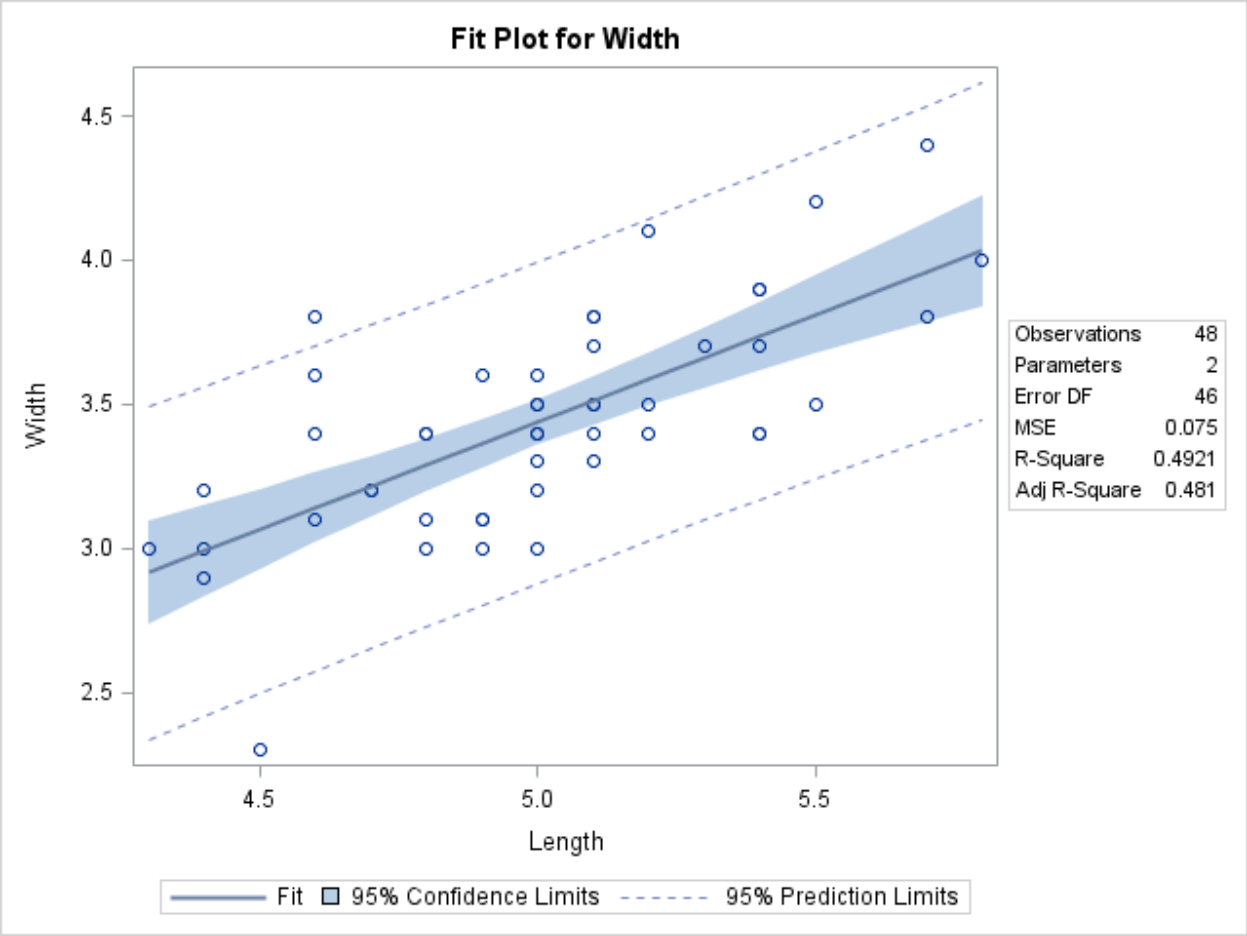


This basically shows how far apart from zero are all the residuals for each point. As we can see there are somewhat evenly distributed around zero, with a moderate spread. So it seems like our model is a good fit.

In the fit diagnostics panel we can get a lot more information about the residuals and if we focus on their histogram (bottom left corner) we can safely say that the residuals DO follow a normal distribution around zero. (One of the requirements of regression). Further evidence for that is the QQ plot of the residuals (middle left graph). On the bottom right corner we see a summary of the regression analysis. Another interesting graph is the middle one. To create it, SAS is using the model and the Length values and predicts the Width values. Then it plots the predicted values vs the real ones. It would be ideal if all the plotted points were on the y=x axis, but still our result is not that bad.



Finally we get a much better scatterplot with the 95% confidence interval lines and the 95% Prediction limits (which we will talk about next week). This is the plot that I prefer putting in my results since it also has a summary on the side.



Saving outputs and Further Analysis:

So far we have been interested in statistical outputs and graphs coming from SAS. But what if we wanted to get the output from a tedious computation for further analysis. For example, in the computation of linear regression we discussed that computing the residuals is one of the most important aspects. Furthermore, testing the normality of their distribution is crucial. Earlier in the semester we learned a test for normality with the Kolmogorov-Smirnov test.

The following code will allow us to save the extracted residuals in a new dataset and then apply any analyses from the ones that we have already learned. The command is **output** and it can be used in other SAS procedures not just PROC REG.

```
Title "Saving the residuals";
PROC REG Data=Iris;
Model Width=Length;
Output out=Results r=res;
RUN;
```

The extra line **Output out=Results r=res;** creates a new dataset utilizing the output of the process it is attached to. The name of the dataset is Results and it is given by the command **out=Results**. Since we want to keep the residuals in the output dataset we need to give the residuals a name in the output dataset. The way to signify residuals is simply **r** and so I gave them the name res with the command **r=res**.

Let's go ahead and PROC PRINT the dataset results and see what it looks like:

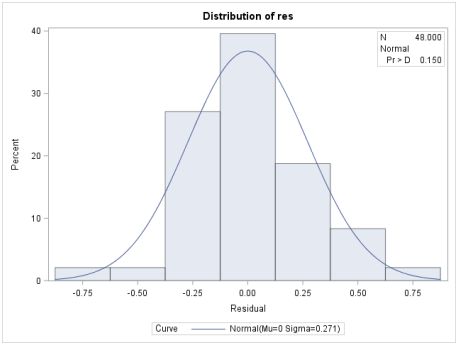
```
PROC PRINT DATA = Results;
RUN;
```

The dataset contains the columns of width and length but also the column of the residuals as a new variable res. We can now apply all the tools we know and especially the normality test we learned in lab 7 namely:

```
TITLE "How Normal are the residuals?";
Ods select Histogram ParameterEstimates GoodnessOfFit FitQuantiles Bins;
PROC UNIVARIATE DATA = Results;
HISTOGRAM res/ normal(percents=20 40 60 80 midpercents);
INSET n normal(ksdpval) / pos = ne format =6.3;
RUN;
```

Since the KS statistic is greater than 0.05 we can safely say that our residuals probably follow a normal distribution and everything is fine.

Goodness-of-Fit Tests for Normal Distribution				
Test	**Statistic**		**p Value**	
Kolmogorov-Smirnov	**D**	0.07706671	**Pr > D**	>0.150
Cramer-von Mises	**W-Sq**	0.03377812	**Pr > W-Sq**	>0.250
Anderson-Darling	**A-Sq**	0.22661831	**Pr > A-Sq**	>0.250



For more information on the output functionality for regression in SAS check here:

https://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm#statug_reg_sect015.htm