Stat 302
Statistical Software and Its Applications
SAS: Simple Linear Regression

Yen-Chi Chen

Department of Statistics, University of Washington

Autumn 2016

Outline

- SAS procedures for simple linear regression.
  - `proc sgplot` + `reg`
  - `proc corr`
  - `proc reg`
- Log transform and simple linear regression.
- Multiple linear regression.
- Examples of fitting linear regression.
  - `student` dataset.
  - A simulated dataset.

Setup – 1

- Go to Canvas and download the dataset `SpiritStLouis.csv`.
- It is a dataset about airplane takeoff distance from `http://www.charleslindbergh.com/hall/spirit.pdf`.
- Do the following to import the dataset into SAS:

```
data spirit;
   infile "U:\data\SpiritStLouis.csv" dsd;
   input gas weight headwind TO_distance; run;
title "Spirit of St. Louis Takeoff Distance";
   proc print data = spirit; run;
```

Setup (Plot)

Spirit of St. Louis Takeoff Distance

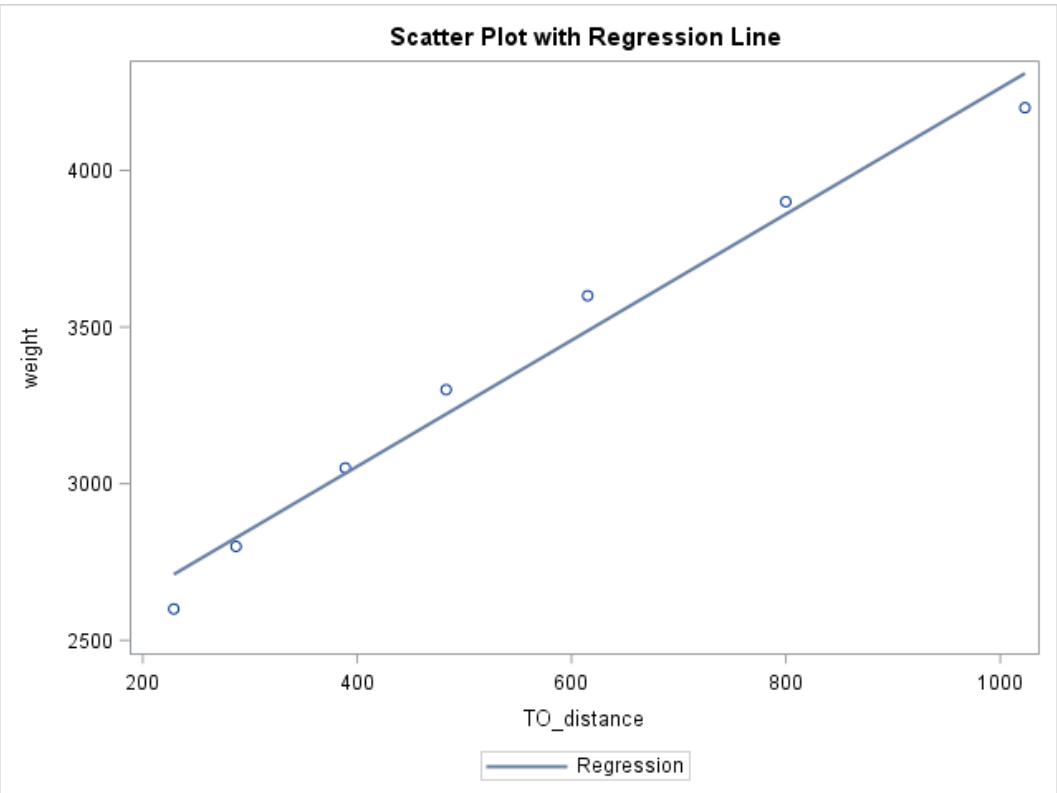
Obs	gas	weight	headwind	TO_distance
1	36	2600	7	229
2	71	2800	9	287
3	111	3050	9	389
4	151	3300	6	483
5	201	3600	4	615
6	251	3900	2	800
7	301	4200	0	1023

- In today's analysis, we will focus on variable `weight` and `TO_distance`.
- We will treat `weight` as the response variable (Y) and `TO_distance` as the covariate (X).

proc sgplot – 1

- We can show the scatter plot with fitted linear regression using `proc sgplot`.

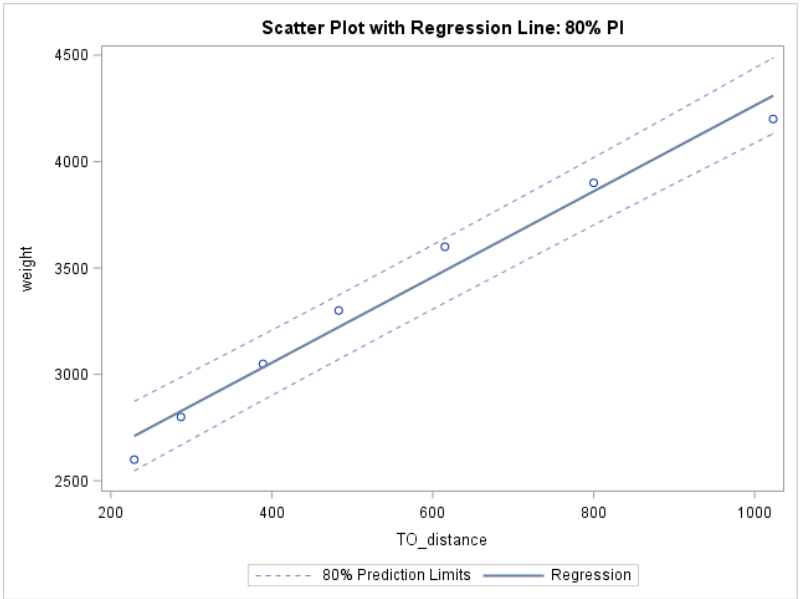
```
title "Scatter Plot with Regression Line";
   proc sgplot data=spirit;
      reg y = weight x=TO_distance; run;
```



proc sgplot – 2

- To add a prediction interval, use `/CLI`.
- The command `alpha=0.2` means that we are constructing a `1-alpha` prediction interval.

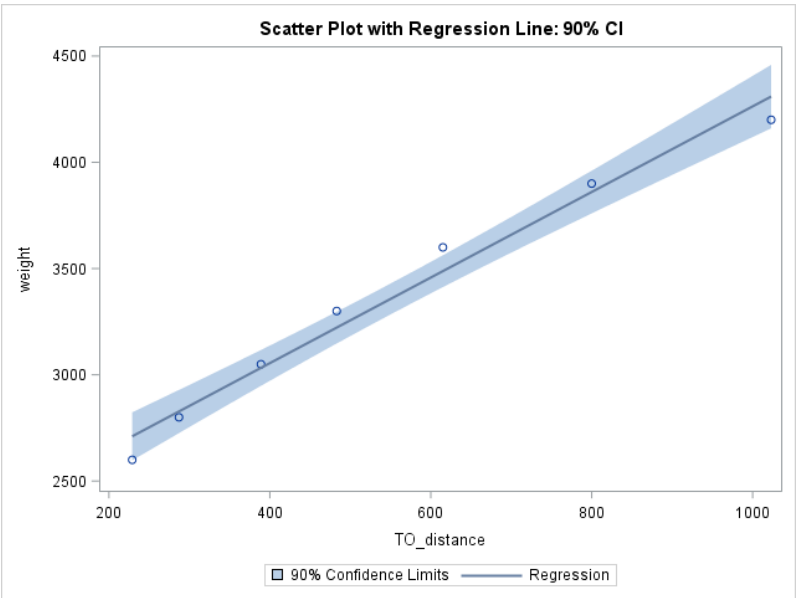
```
title "Scatter Plot with Regression Line: 80% PI";
   proc sgplot data=spirit;
      reg y = weight x=TO_distance/
         CLI alpha=0.2;
run;
```



proc sgplot – 3

- To show a confidence interval, use `/CLM`.
- The quantity `alpha` controls the confidence level.

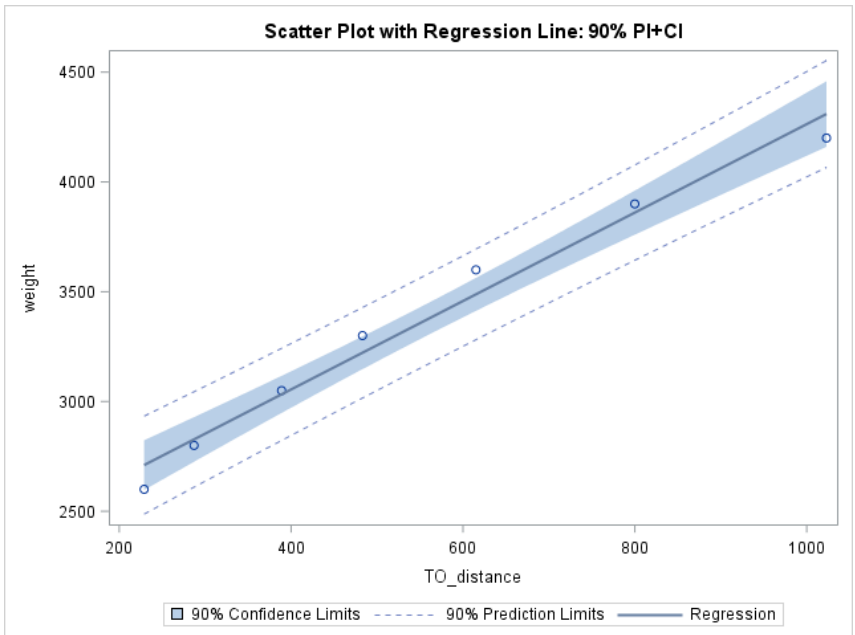
```
title "Scatter Plot with Regression Line: 90% CI";
   proc sgplot data=spirit;
      reg y = weight x=TO_distance/
         CLM alpha=.1;
run;
```



proc sgplot – 4

- We can show both prediction interval and confidence interval at the same time.

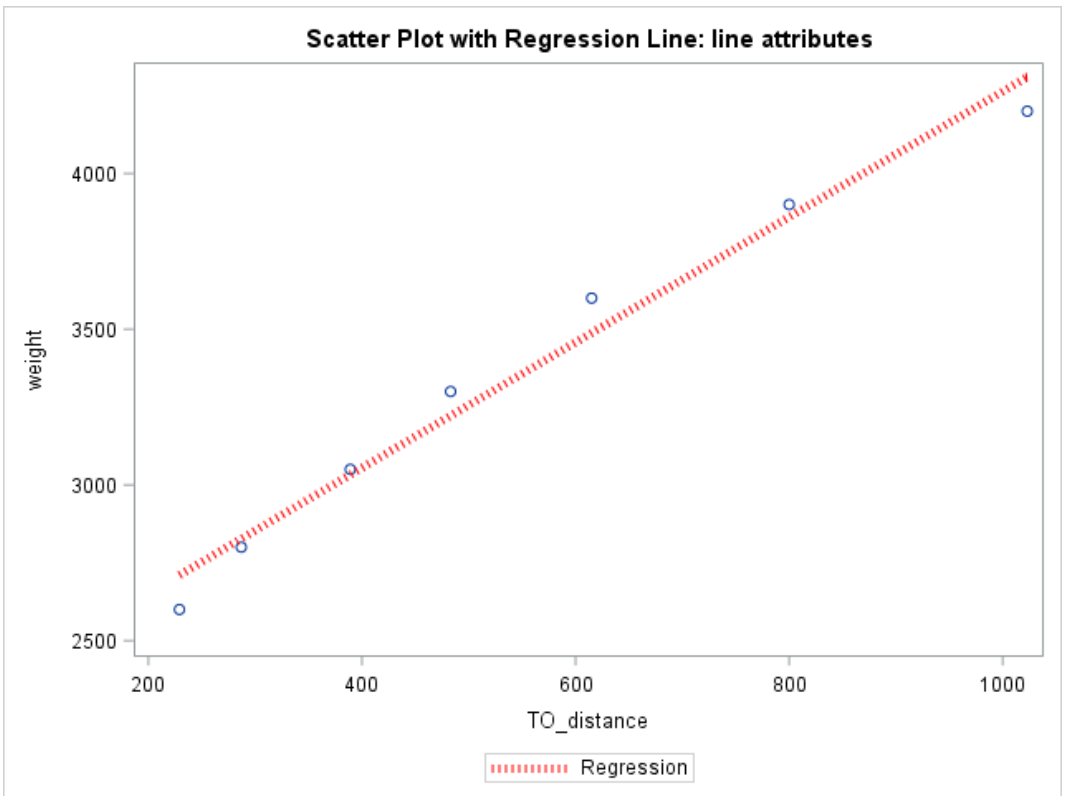
```
title "Scatter Plot with Regression Line: 90% PI+CI";
   proc sgplot data=spirit;
      reg y = weight x=TO_distance/
         CLI CLM alpha=0.1;
run;
```



proc sgplot – 5

- To adjust the line attributes, use `/ lineattrs = ...`

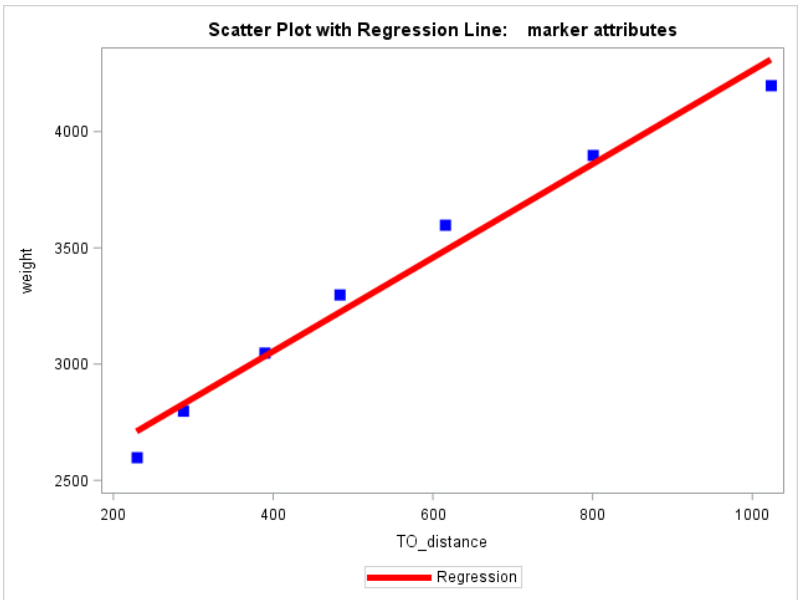
```
title "Scatter Plot with Regression Line: line attributes";
   proc sgplot data=spirit;
     reg y = weight x=TO_distance/
        lineattrs=(color=red thickness=5 pattern=dot);
run;
```



proc sgplot – 6

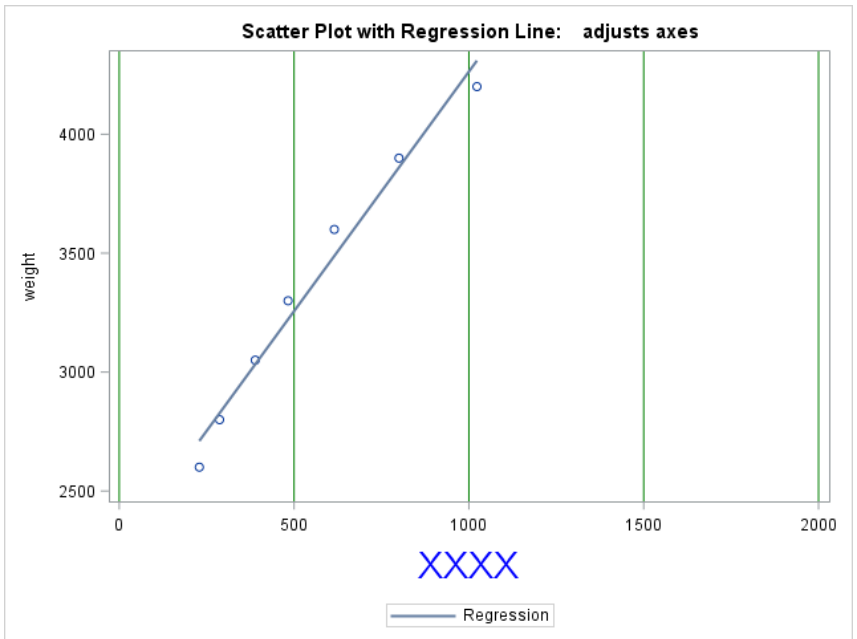
- To adjust the attributes of data points, use `/ markerattrs = ...`

```
title "Scatter Plot with Regression Line:
        marker attributes";
   proc sgplot data=spirit;
     reg y = weight x=TO_distance/
        lineattrs=(color=red thickness=5)
        markerattrs=(color=blue size=10
                symbol=squarefilled);
run;
```



- To adjust the axes, use `xaxis` and `yaxis`.

```
title "Scatter Plot with Regression Line:
        adjusts axes";
   proc sgplot data=spirit;
     reg y = weight x=TO_distance;
        xaxis label="XXXX" min = 0 max = 2000
                labelattrs=(size=20 color=blue)
                grid gridattrs=(color=green) ;
run;
```



proc corr

- The `corr` procedure is a method to obtain a table of the correlation analysis.

```
title "Correlation";
   proc corr data = spirit;
     var weight TO_distance; run;
```

proc corr (Plot)

Correlation

The CORR Procedure

2 Variables:	weight TO_distance

Simple Statistics

Variable	N	Mean	Std Dev	Sum	Minimum	Maximum
weight	7	3350	583.80933	23450	2600	4200
TO_distance	7	546.57143	286.64488	3826	229.00000	1023

Pearson Correlation Coefficients, N = 7

Prob > |r| under H0: Rho=0

	weight	TO_distance
weight	1.00000	0.98882 <.0001
TO_distance	0.98882 <.0001	1.00000

proc reg – 1

- The `reg` procedure is a power tool for regression analysis.
- It performs a comprehensive analysis for linear regression.
- First it generates a summary table.
- Then it shows diagnostic plots, a residual plot, and the scatter plot with fitted regression line.

```
title "Simple Linear regression";
proc reg data = spirit;
model weight = TO_distance;
run;
```

proc ref – 1 (Plot – 1)

Simple Linear regression

The REG Procedure
Model: MODEL1
Dependent Variable: weight

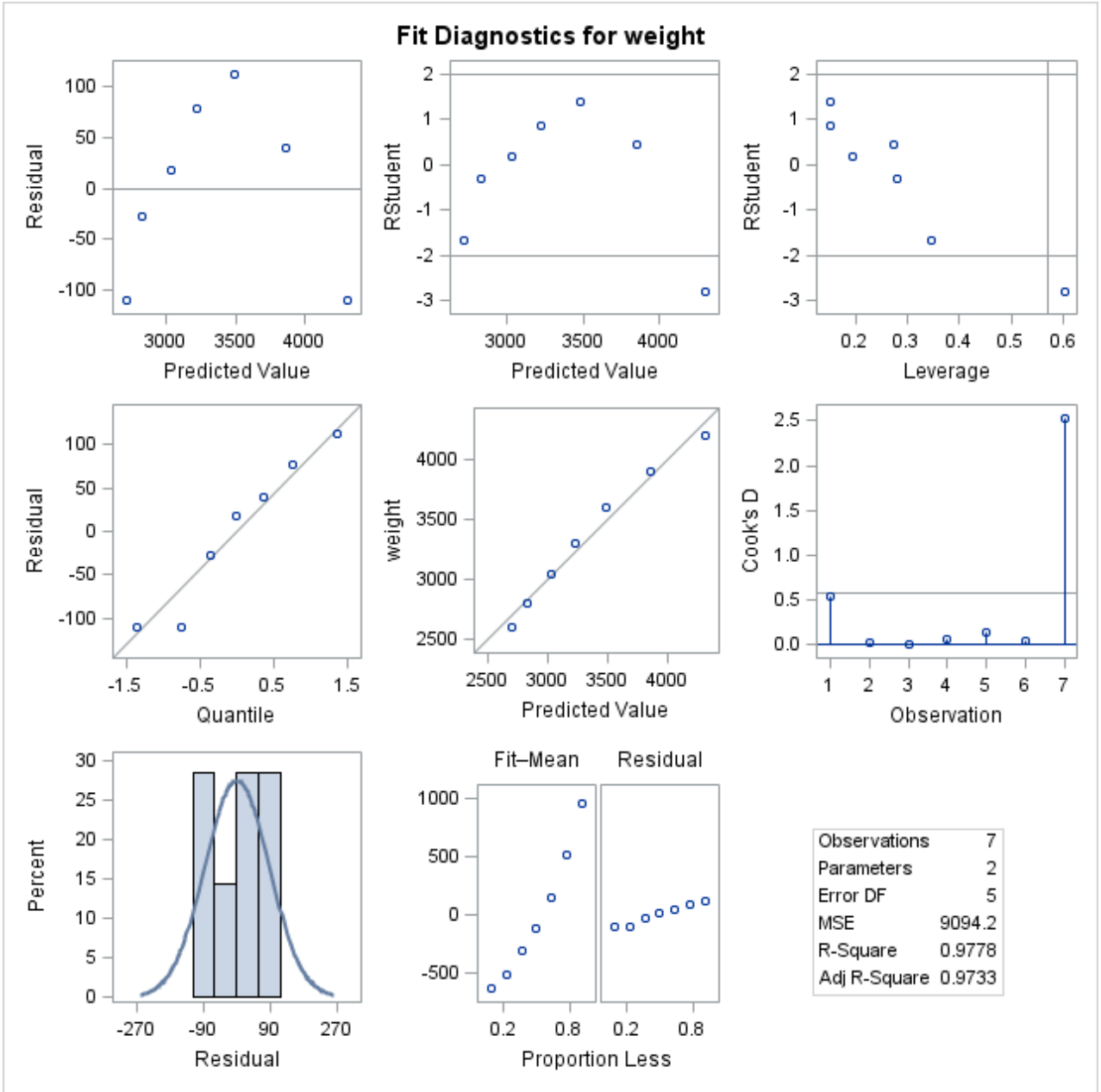
Number of Observations Read	7
Number of Observations Used	7

Analysis of Variance					
Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	1999529	1999529	219.87	<.0001
Error	5	45471	9094.24340		
Corrected Total	6	2045000			

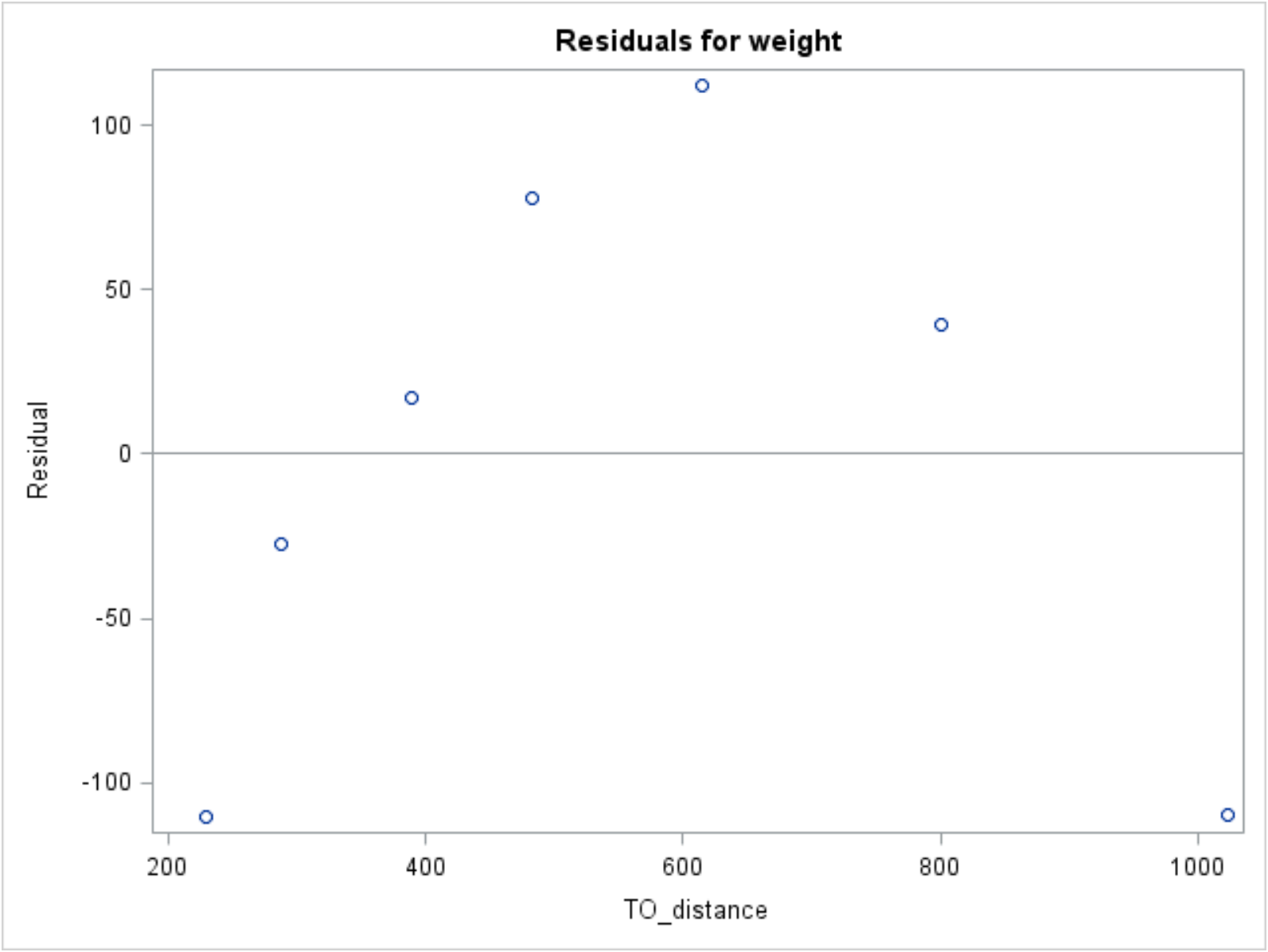
Root MSE	95.36374	R-Square	0.9778
Dependent Mean	3350.00000	Adj R-Sq	0.9733
Coeff Var	2.84668		

Parameter Estimates					
Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	2249.24429	82.52306	27.26	<.0001
TO_distance	1	2.01393	0.13582	14.83	<.0001

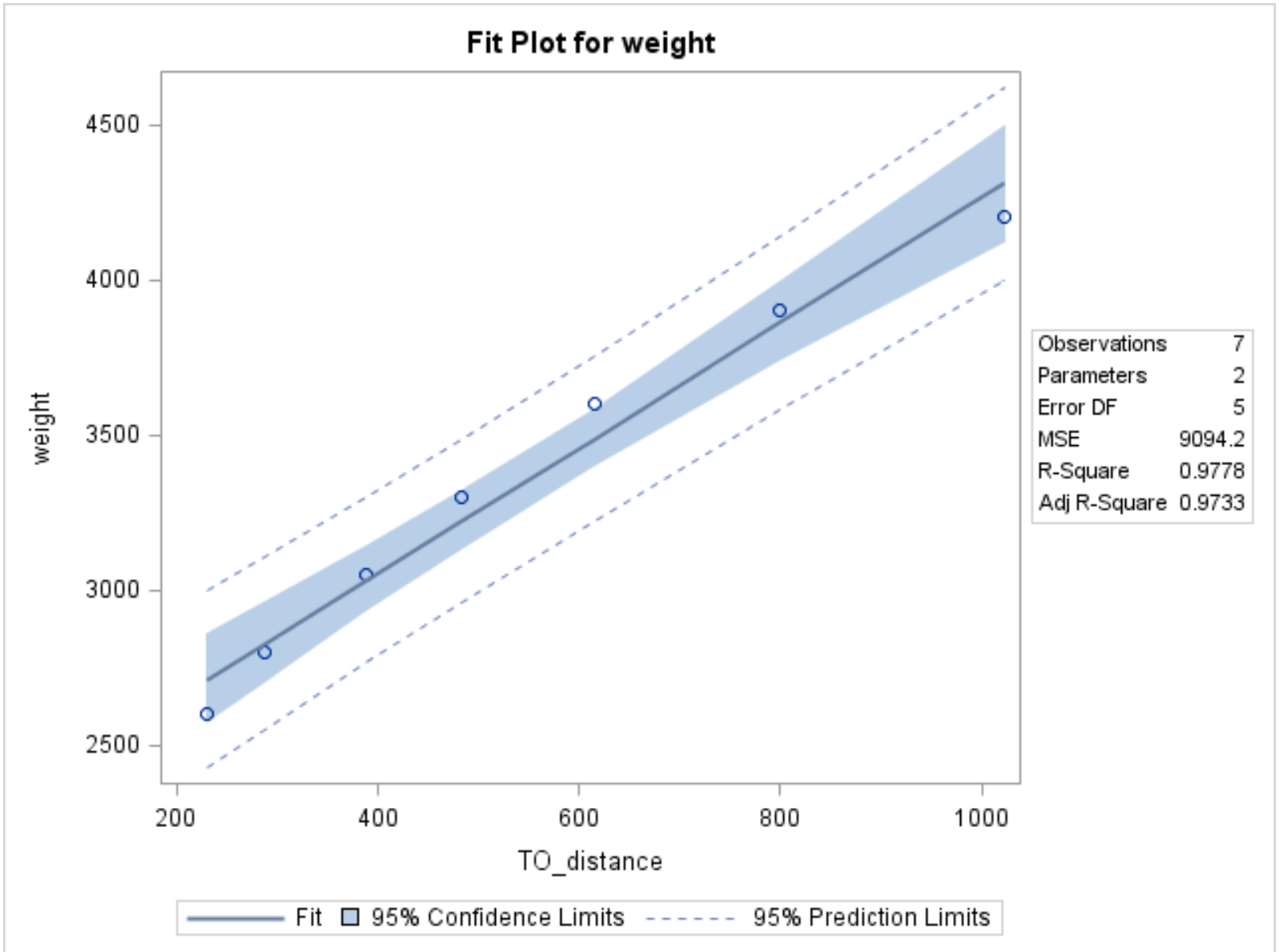
proc ref – 1 (Plot – 2)



proc ref – 1 (Plot – 3)



proc ref – 1 (Plot – 4)



proc reg – 2

- `plots = diagnostics(unpack)`: this unpacks the diagnostic plot.
- It will be useful if you want to use an individual figure in the diagnostic plot.

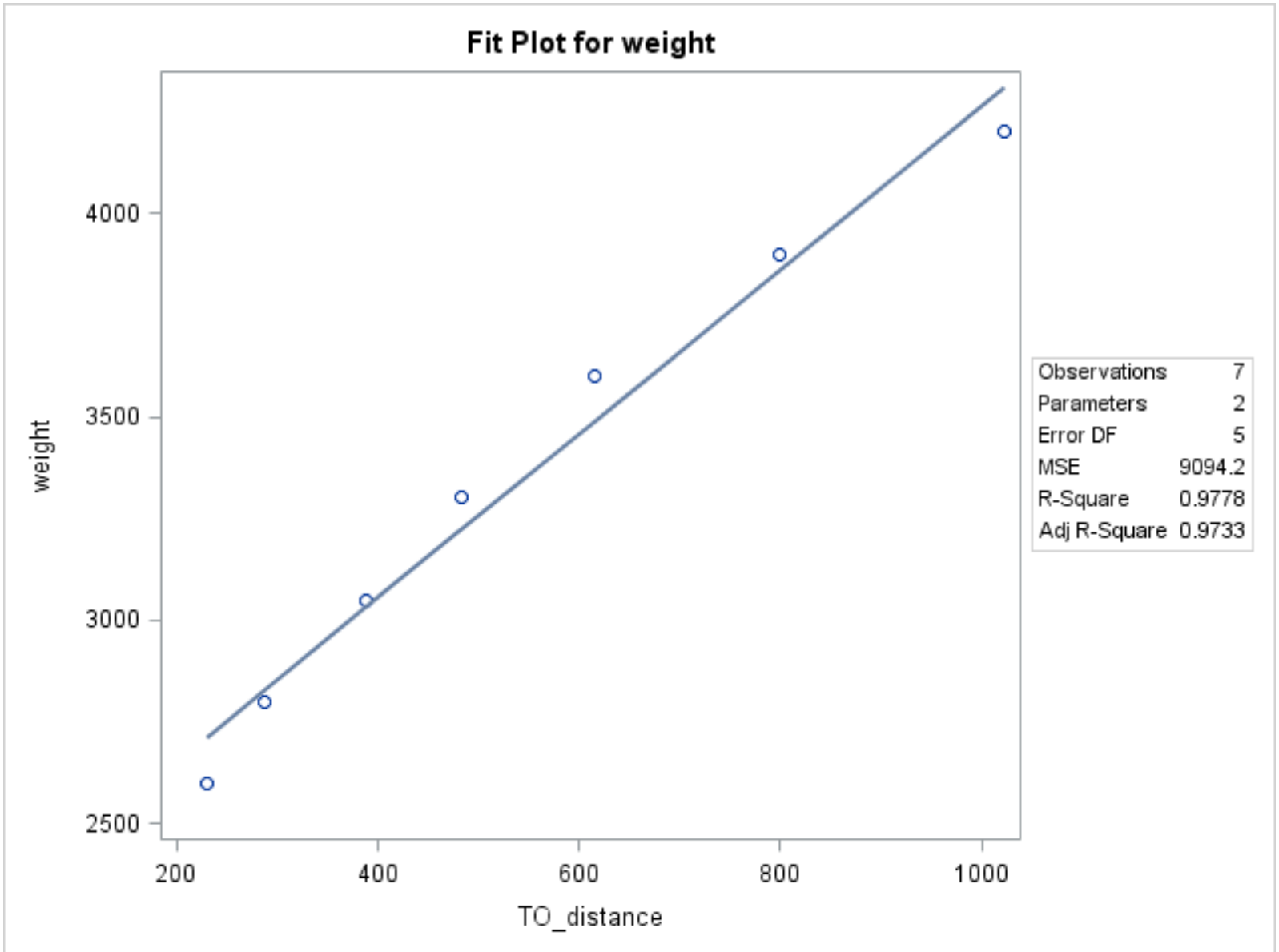
```
title "Simple Linear regression: unpack plots";
proc reg data = spirit
         plots = diagnostics(unpack);
model weight = TO_distance;
run;
```

proc reg – 3

- `plots = FITPLOT(nolimits)`: this removes the prediction interval and the confidence interval.
- `plots = FITPLOT(nocli)`: no prediction interval.
- `plots = FITPLOT(noclm)`: no confidence interval.
- It will be useful if you want to use an individual figure in the diagnostic plot.

```
title "Simple Linear regression: no PI/&CI";
proc reg data = spirit
         plots = FITPLOT(nolimits);
model weight = TO_distance;
run;
```

proc ref – 3 (Plot)



proc reg – 4

- `alpha = ...`: this specifies the prediction level/confidence level.
- `corr`: this adds the correlation table from `proc corr`.

```
title "Simple Linear regression: others";
proc reg data = spirit
         plots = FITPLOT(nocli)
         alpha = 0.2 corr;
model weight = TO_distance;
run;
```

Log transform – 1

- To use log transform, we first create a new data object.

```
data spirit;
   infile "U:\data\SpiritStLouis.csv" dsd;
   input gas weight headwind TO_distance;
        TO_DistL10 = log10(TO_Distance);
        weightL10 = log10(weight); run;
title "Spirit of St. Louis Takeoff Distance L10";
   proc print data = spirit; run;
```

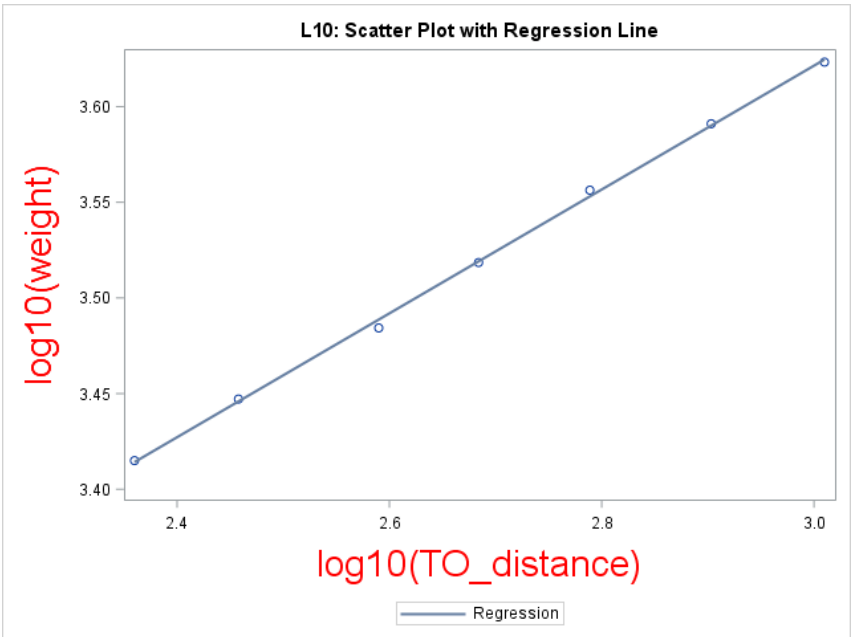
Log transform – 1 (Plot)

Spirit of St. Louis Takeoff Distance L10

Obs	gas	weight	headwind	TO_distance	TO_DistL10	weightL10
1	36	2600	7	229	2.35984	3.41497
2	71	2800	9	287	2.45788	3.44716
3	111	3050	9	389	2.58995	3.48430
4	151	3300	6	483	2.68395	3.51851
5	201	3600	4	615	2.78888	3.55630
6	251	3900	2	800	2.90309	3.59106
7	301	4200	0	1023	3.00988	3.62325

Log transform – 2

```
title "L10: Scatter Plot with Regression Line";
proc sgplot data=spirit;
xaxis label="log10(TO_distance)"
        labelattrs=(size=20 color=red);
yaxis label="log10(weight)"
        labelattrs=(size=20 color=red);
reg y = weightL10 x=TO_distL10;
run;
```



Log transform – 3

Get the correlation table:

```
title "L10: Correlation";
proc corr data = spirit;
var weightL10 TO_distL10; run;
```

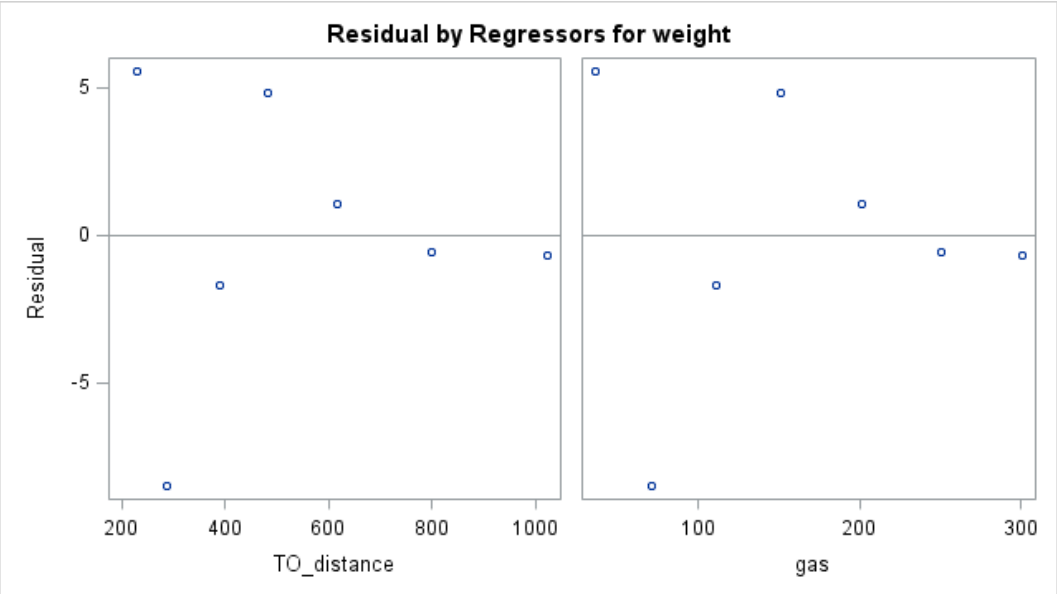
Do the full linear regression analysis:

```
title "Simple Linear regression L10";
proc reg data = spirit;
model weightL10 = TO_distL10;
run;
```

Multiple Linear Regression – 1

- It is very easy to fit a multiple linear regression.
- `model weight = TO_distance gas`: the covariates will be in the right part of the 'equality'.
- In R, we use `lm(weight~ TO_distance+gas)`.

```
title "LR: weight ~ TO_distance, gas";
proc reg data = spirit;
model weight = TO_distance gas;
run;
```



Multiple Linear Regression – 2

- To fit more variables, just add them in the right part of the equality.

```
title "LR: weight ~ TO_distance, gas, headwind";
proc reg data = spirit;
model TO_distance = weight gas headwind;
run;
```

Data analysis example: student data – 1

1. First we import the dataset and print out to have an overview about this data.

```
data student;
infile "U:\data\student.txt";
input Age Major $ GPA;
run;
title "Student DATA";
proc print data= student;
run;
```

Data analysis example: student data – 2

- Assume our target is to analyze variable `Age` and `GPA`.

2. So the next step is to analyze these variables individually.

```
title "Student DATA: GPA";
proc univariate data= student;
histogram GPA/normal;
run;

title "Student DATA: age";
proc univariate data= student;
histogram age/normal;
run;
```

Data analysis example: student data – 3

3. Then we examine the scatter plot along with a simple linear fit.

```
title "Student DATA: GPA vs Age";
   proc sgplot data=student;
     reg y = GPA x=age;
run;
```

- It seems that there is a negative trend.

4. To see if this trend is significant, we use `proc reg` to perform a full analysis for these two variables.

```
title "Student DATA: GPA vs Age";
   proc reg data=student;
     model GPA=age;
run;
```

Data analysis example: simulated data – 1

1. First we generate a random dataset:

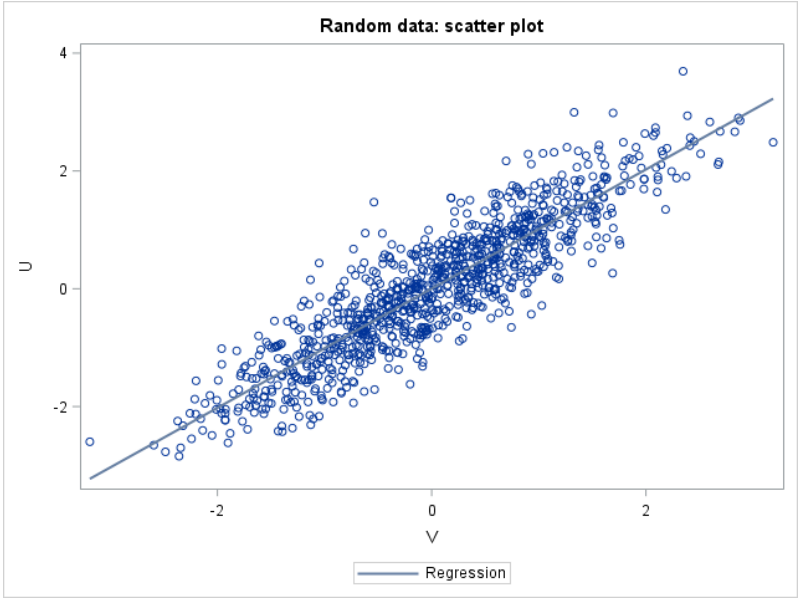
```
data randdata;
do i = 1 to 1000;
        V = rand('normal');
        U = V + rand('normal', 0,0.5);
        output;
end;
run;
title "Random data";
proc print data=randdata noobs;
run;
```

Data analysis example: simulated data – 2

- Now we test how linear regression works when we treat variable `U` as the response and variable `V` as the covariate.

2. Use `proc sgplot` to show the scatter plot along with a simple linear fit:

```
title "Random data: scatter plot";
   proc sgplot data=randdata;
      reg y = U x=V;
run;
```



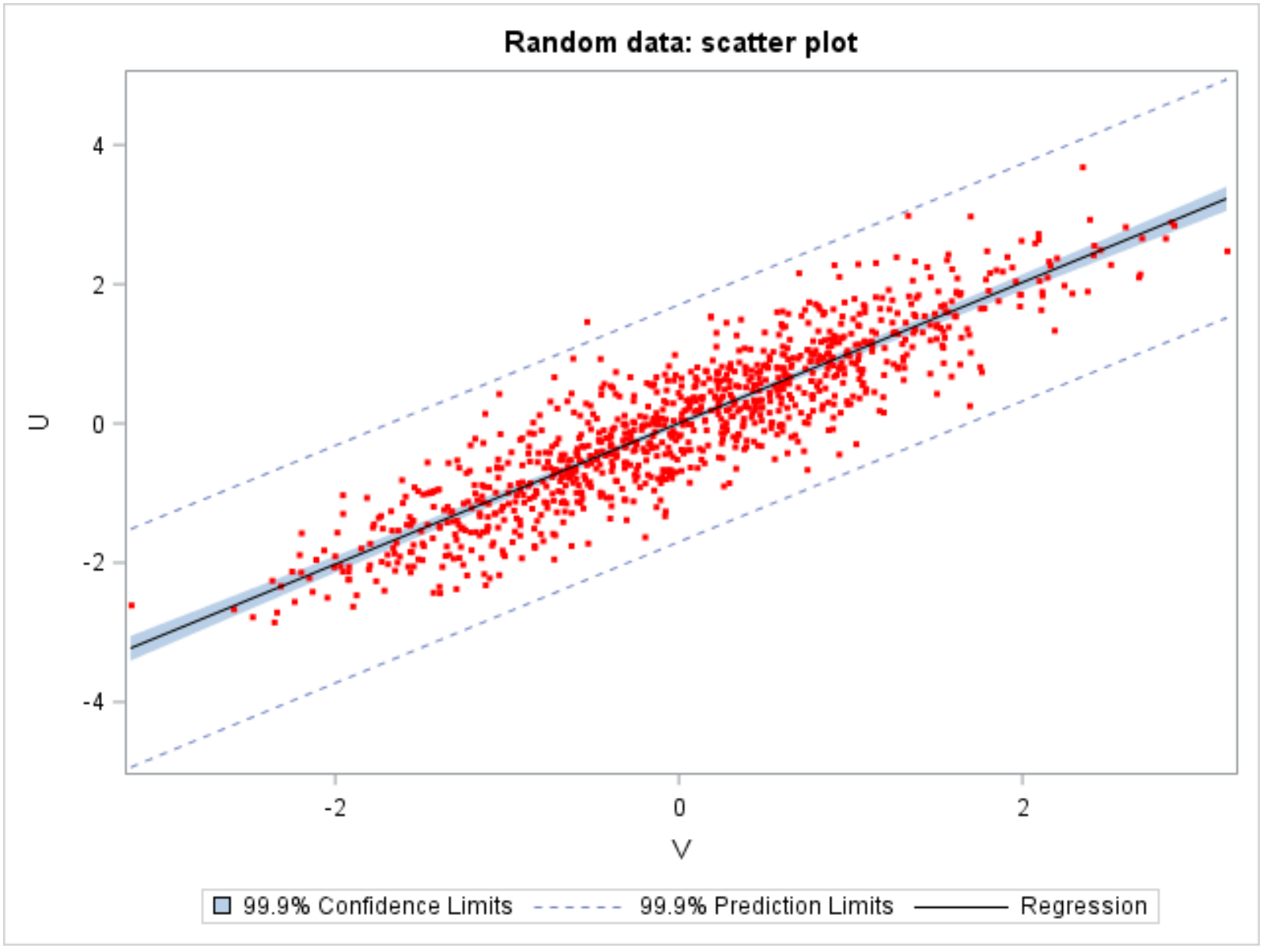
Data analysis example: simulated data – 3

- It seems that there is a strong trend.

3. Now we add the prediction interval and confidence interval to the linear fit:

```
title "Random data: scatter plot";
   proc sgplot data=randdata;
     reg y = U x=V/CLI CLM alpha=0.001
        lineattrs=(color=black thickness=1)
        markerattrs=(color=red size=3
                symbol=squarefilled);
run;
```

Data analysis example: simulated data – 3 (Plot)



Data analysis example: simulated data – 4

4. Finally, we apply simple linear regression to perform a detailed analysis.

```
title "Random data: regression analysis";
   proc reg data=randdata;
     model U=V;
run;
```

In-class Exercise

1. Generate `X=0.1, 0.2, ..., 5` and $Y = X + N(0, 1)$. Namely, the value of `Y` is the value of `X` plus a standard normal noise.
2. Use `proc sgplot` to show the scatter plot along with a regression line.
3. Based on the previous result, add a 90% confidence interval to the regression line and change the color of the regression line into `red`.
4. Use `proc corr` to show the correlation table. What is the correlation between variable `X` and `Y`?
5. Use `proc reg` to perform a comprehensive linear regression for variable `X` and `Y`. What are the estimated intercept and slope? What are the errors of the estimation?
6. Based on the previous result, does the residual looks like a normal?