UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
School of Professional & Executive Development

Tools for Artificial Intelligence with MATLAB, initiation (TAIM)

José Antonio Lázaro

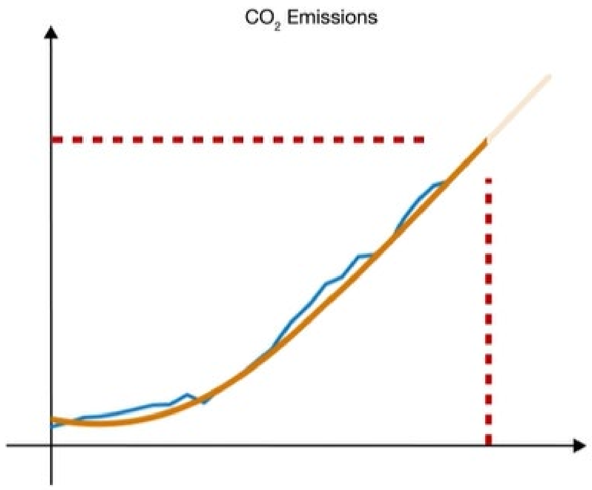
Data analysis: 2 dimensions (Curve fitting)

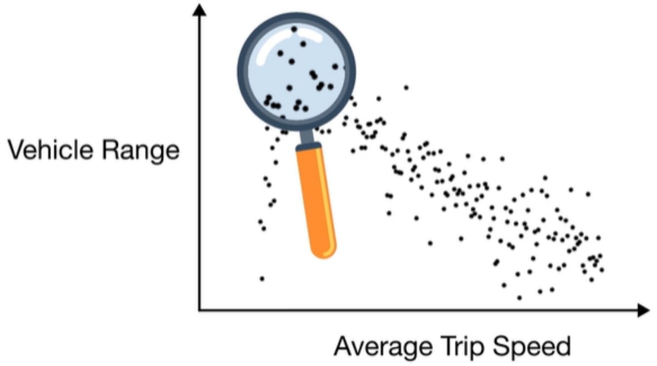
Barcelona, 3, February 2025

“Nous seminaris transversals per a estudiantat de grau i màster universitari de la UPC”

What for?

- E.g. Future predictions and estimations

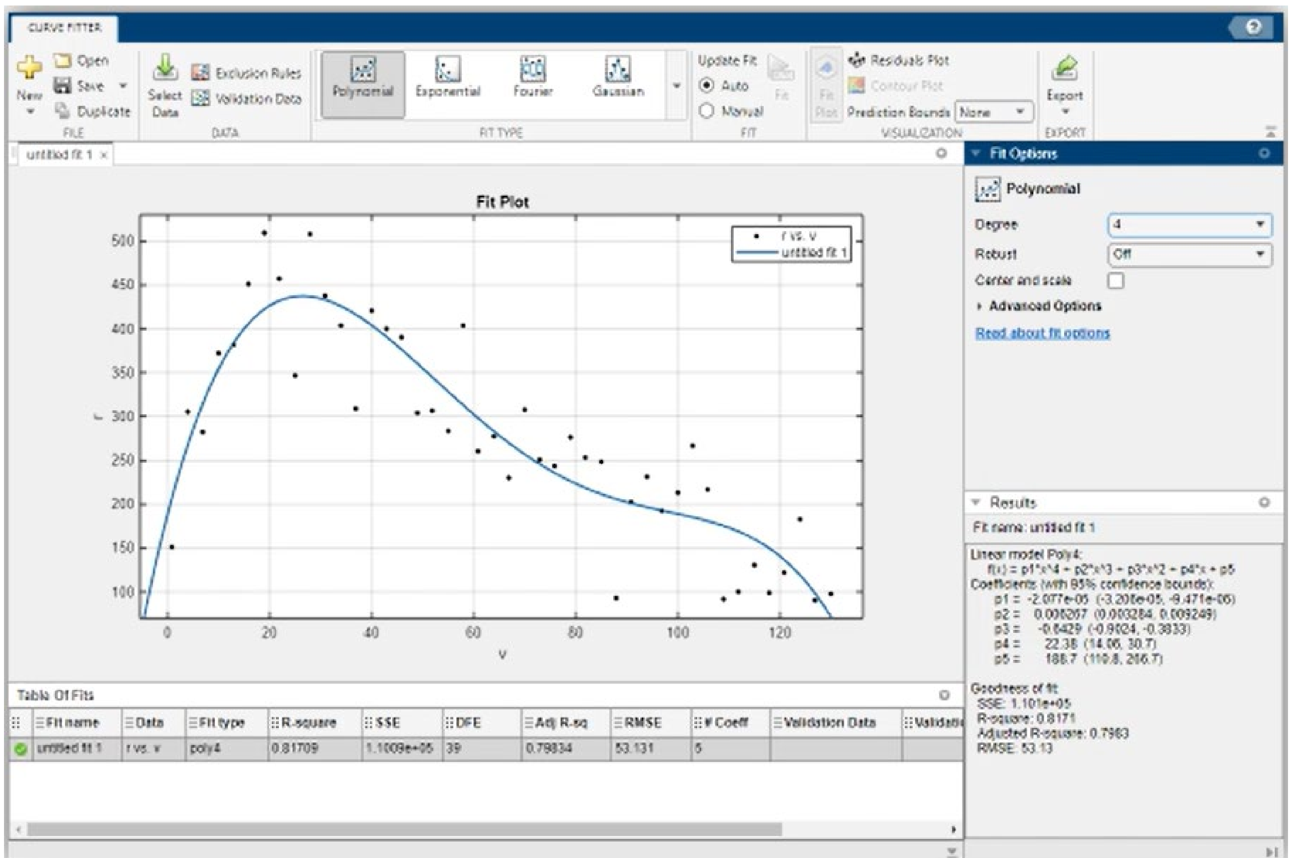




- Optimization: e.g. the best speed of a vehicle to reach a maximum distance?

What for?

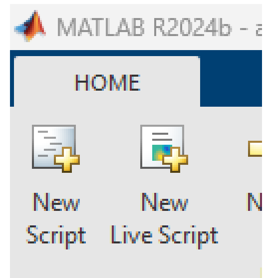
- Optimization: e.g. the best speed of a vehicle to reach a maximum distance for an electric vehicle



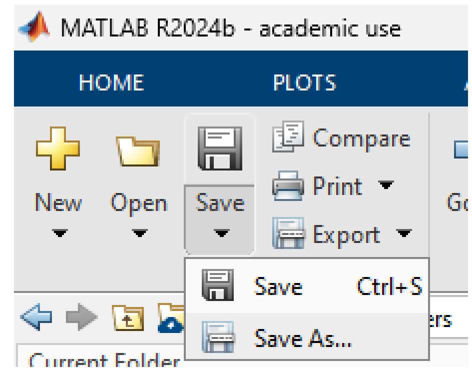
Let’s go

Open Matlab

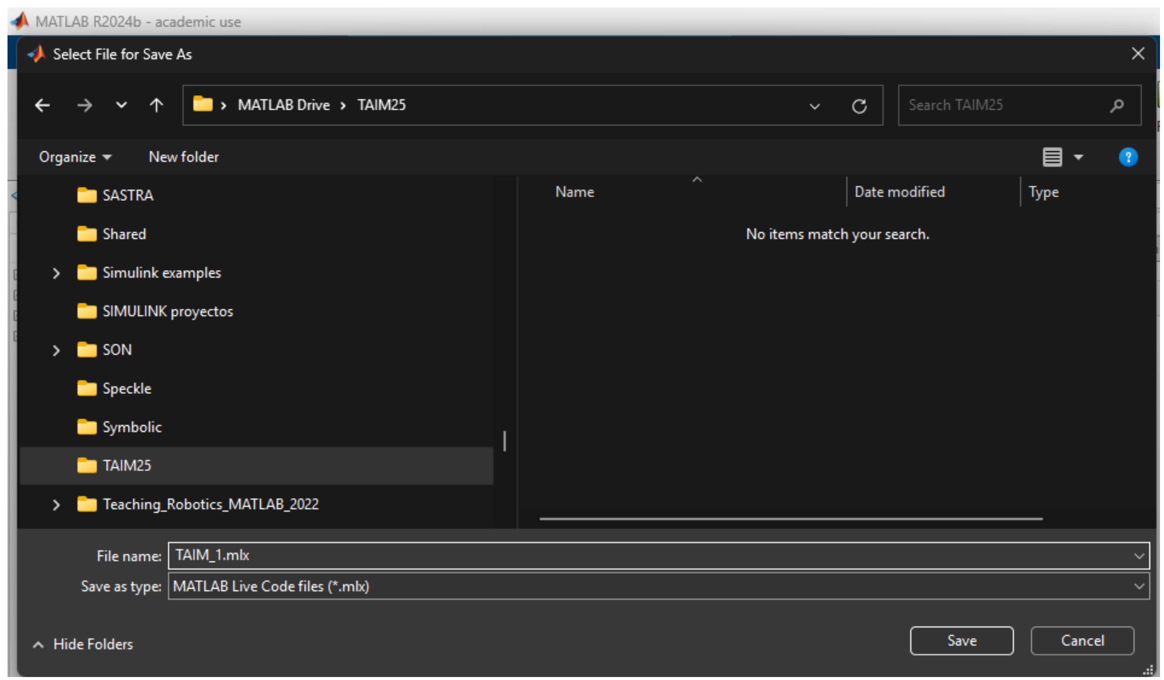
- Do “New Live Script”



- Go “Live Editor”



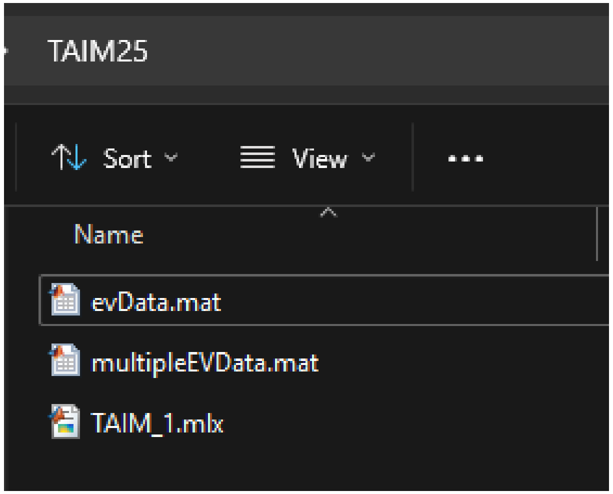
- And “Save as”
- Create a Folder for your course “TAIM25”
- Select a Name and save it. (E.g. “TAIM_1”)
- NO SPACES at the NAME



Let's go

Go to "My_TECH_SPACE"

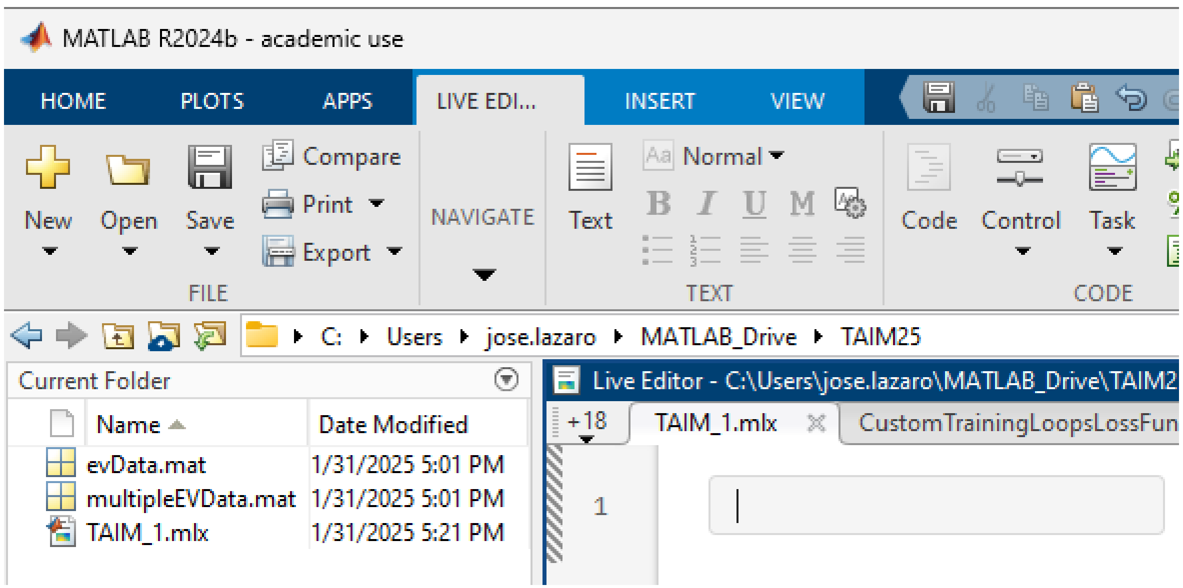
- Download the files: "evData.mat" and "multipleEVData.mat"
- Copy them at your "TAIM_1" folder.
- It should look like this



Let’s go

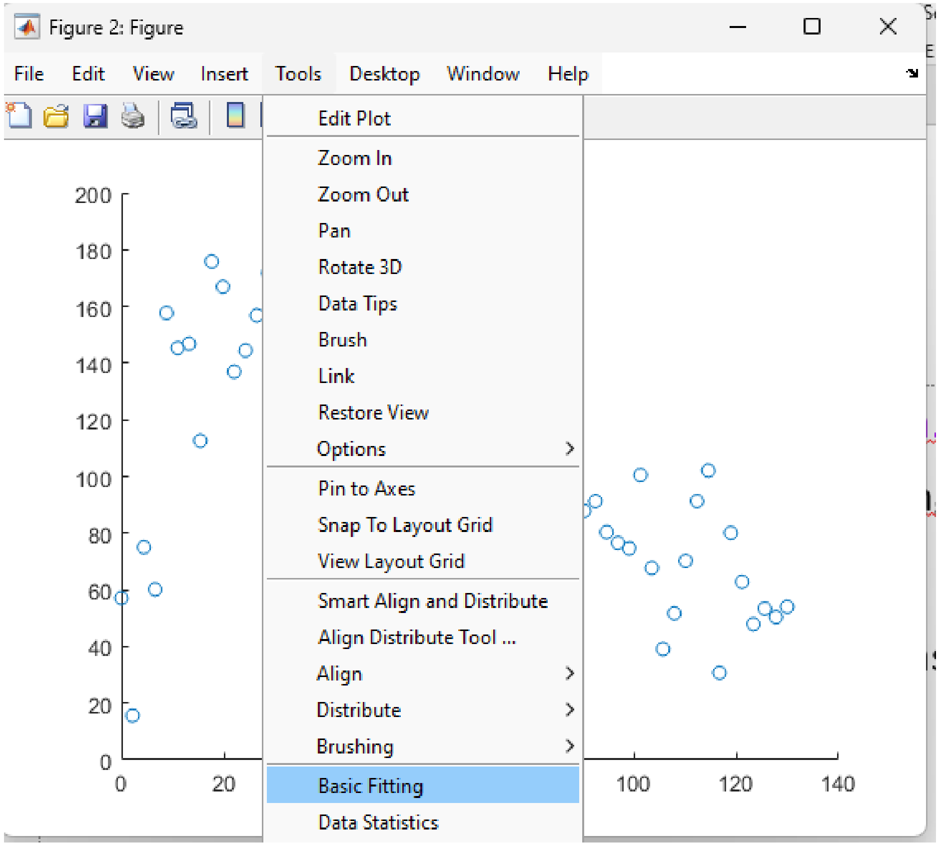
Go back to Matlab

Your file should look like this:



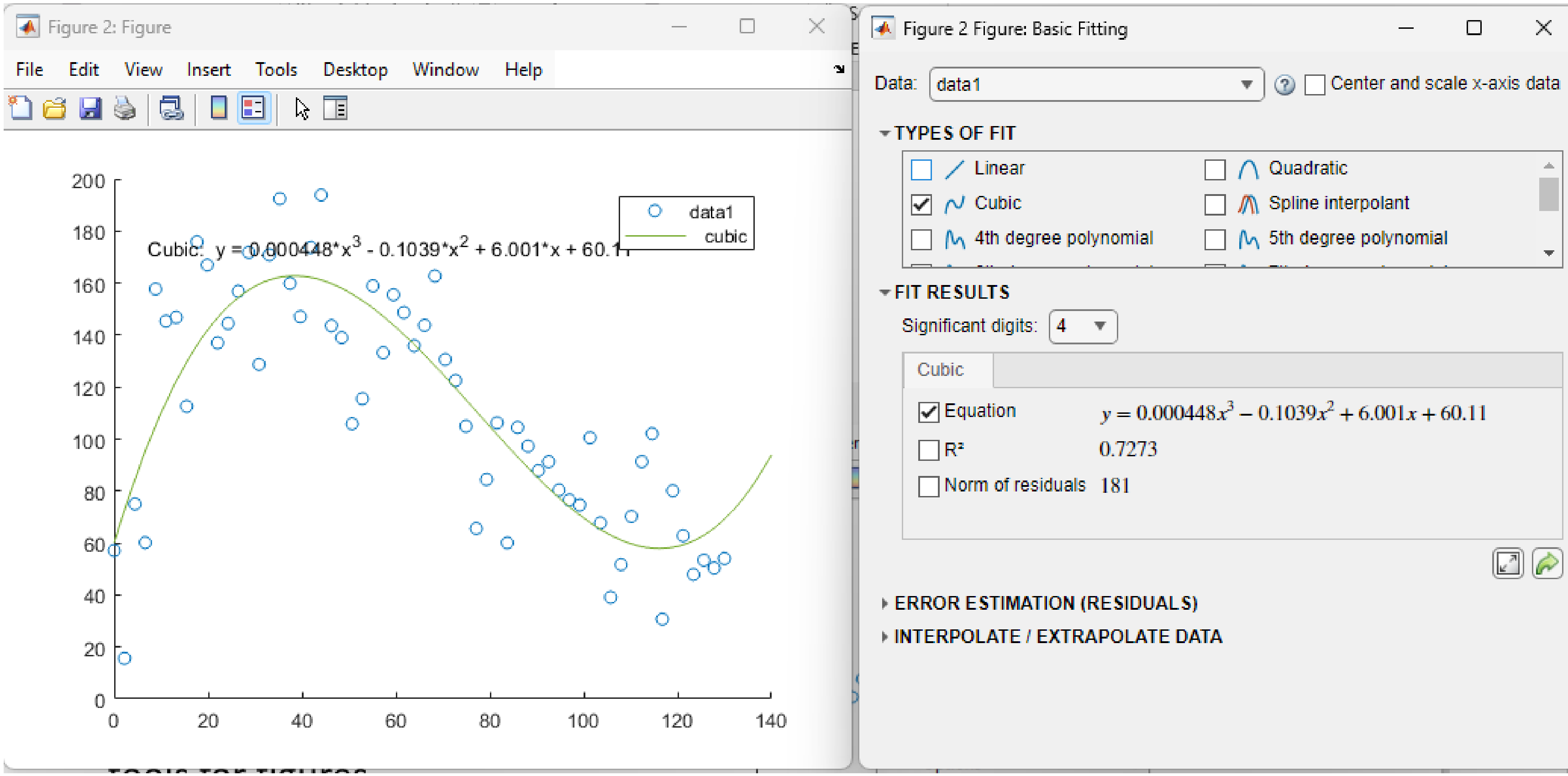
Let’s go

- Start loading the working data: “load evData.mat”
- Now let’s draw the data: “scatter(speed,range)”
- Now we can do a 1st basic fitting suing the basic fitting tools for figures



Let’s go

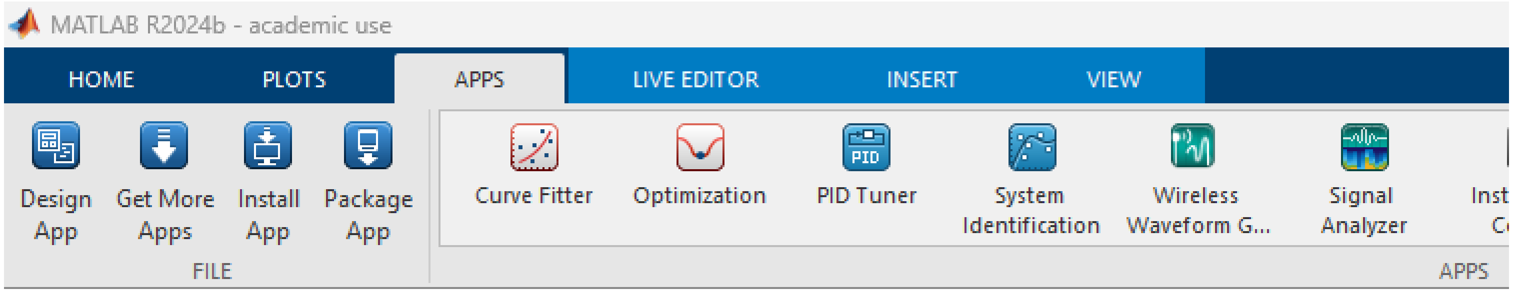
- You can do a 1st fitting, though maybe too basic



Let’s do it better

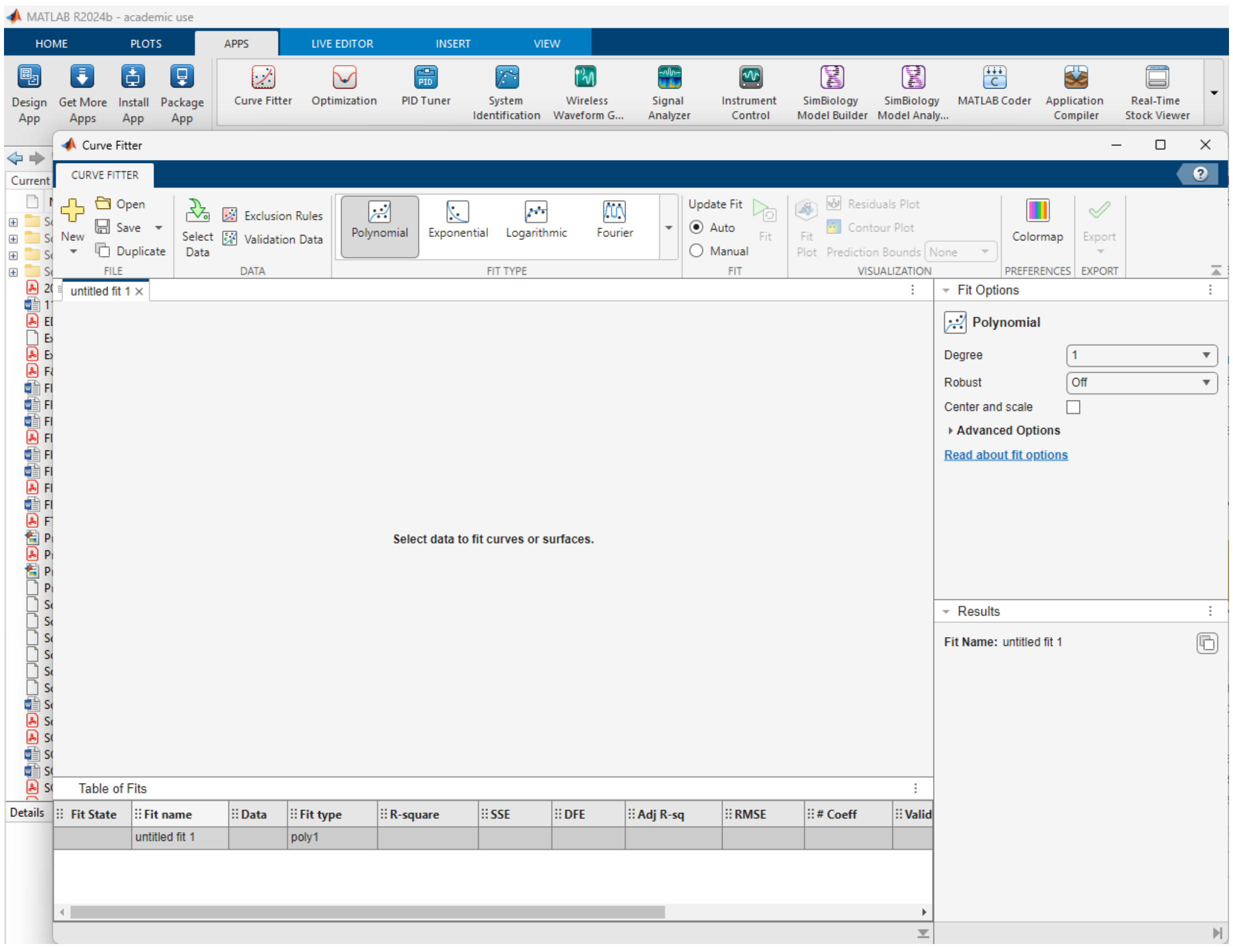
- Go to “APPS”

- Open “Curve Fitter” Application



Curve Fitter App

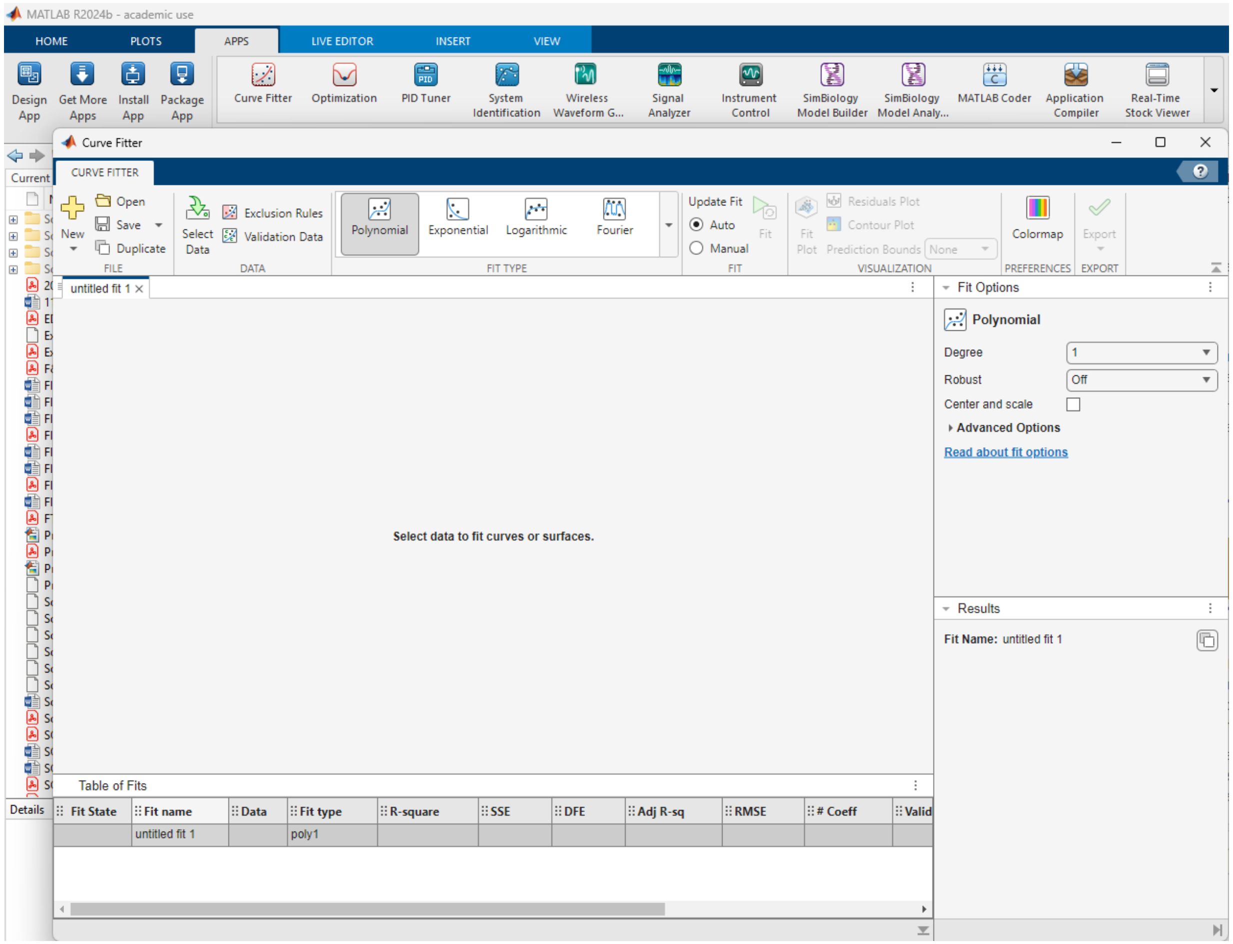
- You get this

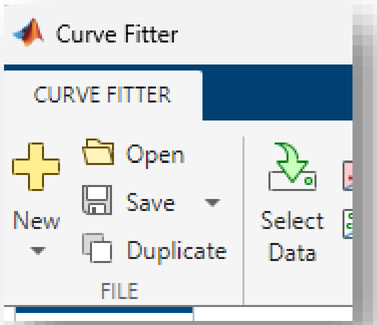


UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
School of Professional & Executive Development

Curve Fitter App

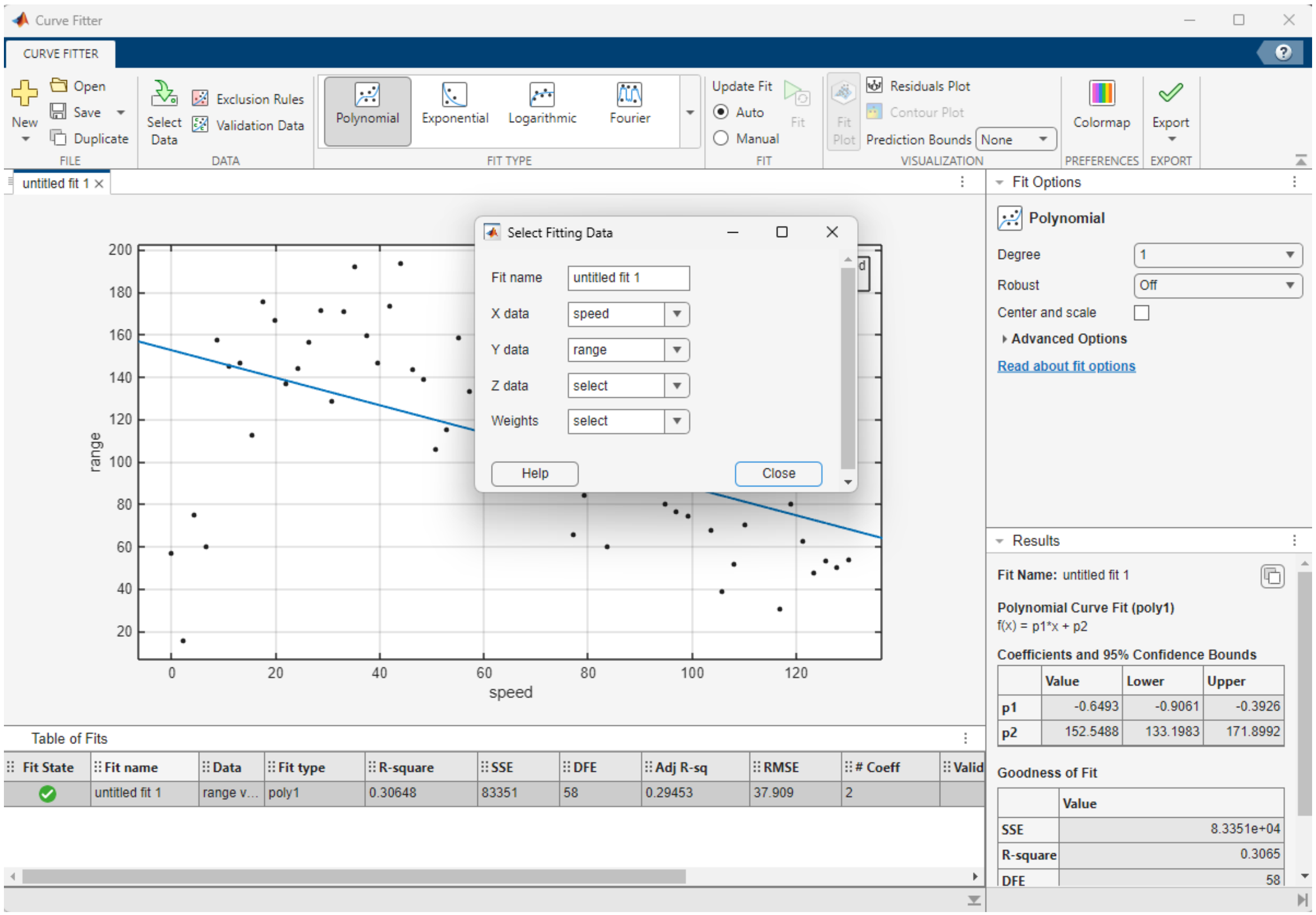
- Import the Data at the new tool





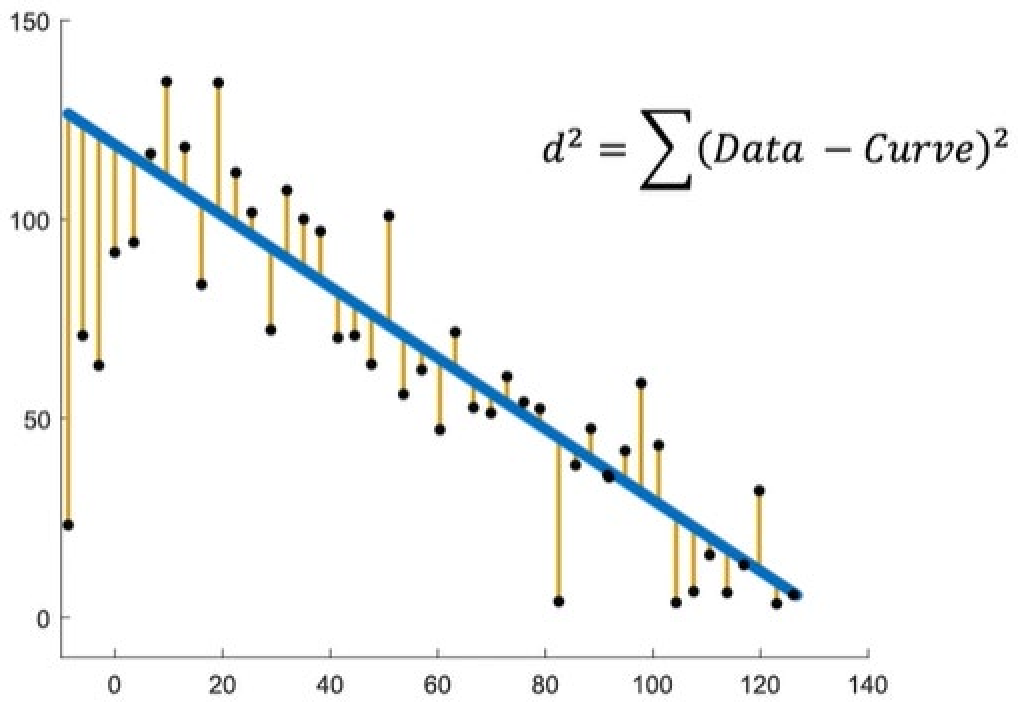
Curve Fitter App

- Import the Data at the new tool, selecting **X = Speed** & **Y = Range**



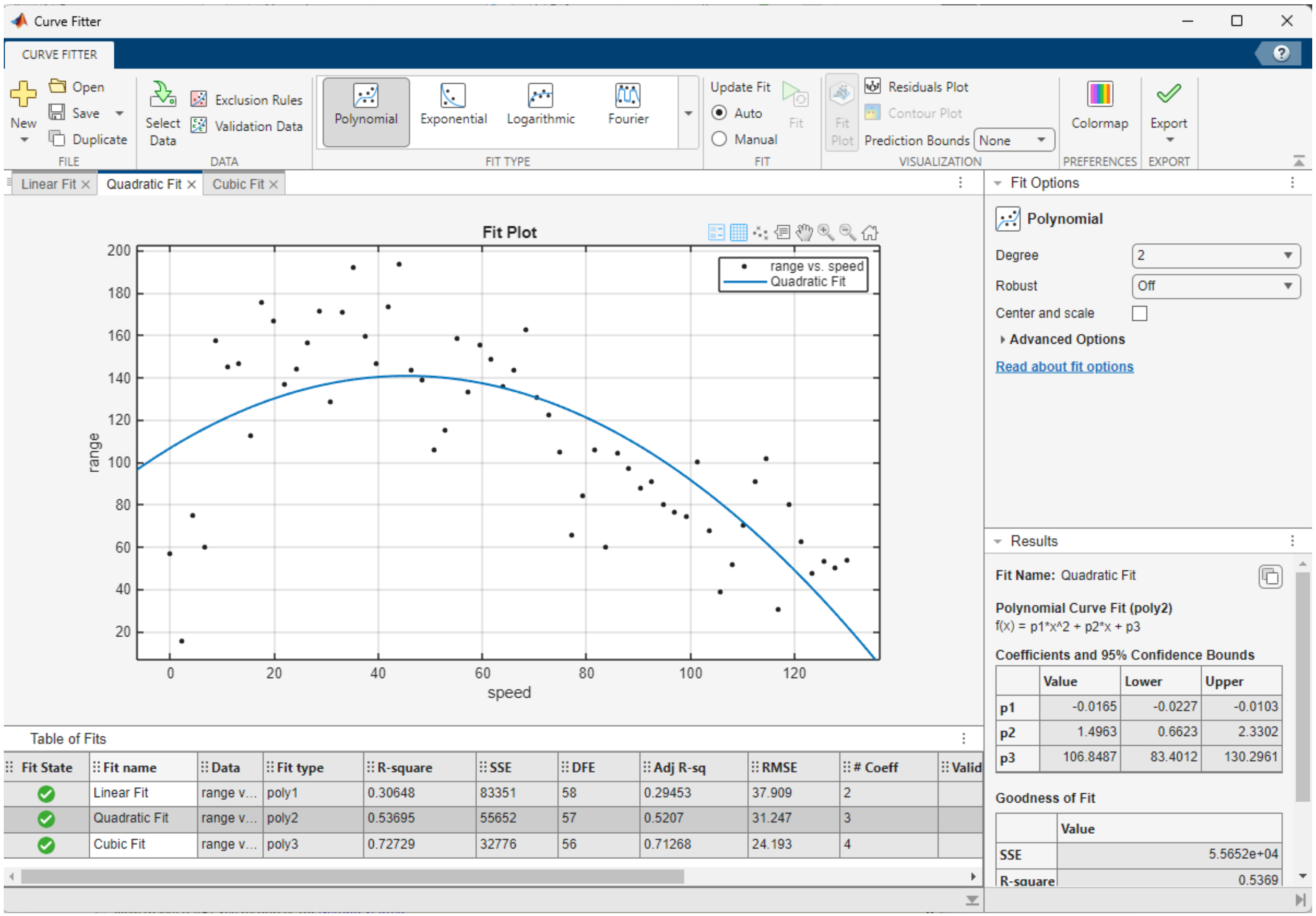
Curve Fitter App

- A 1st curve is done, but: Is this the best mathematical model of the data?
- The linear curve has been calculated, automatically to minimize the "d^2"



Curve Fitter App

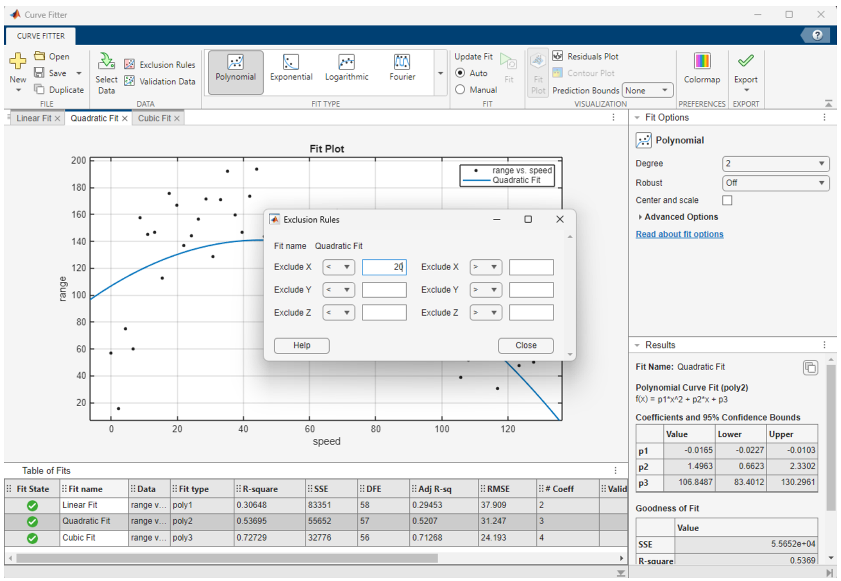
- You can also try a quadratic fit

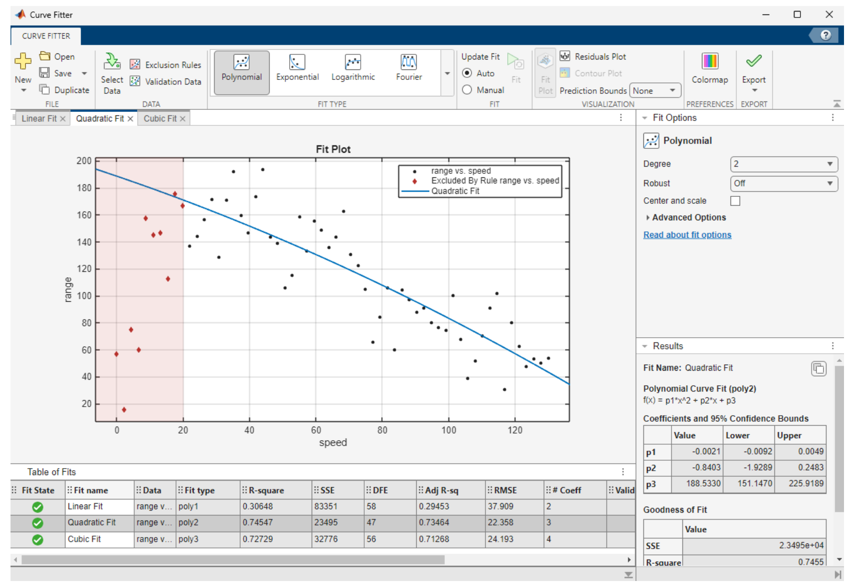


UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
School of Professional & Executive Development

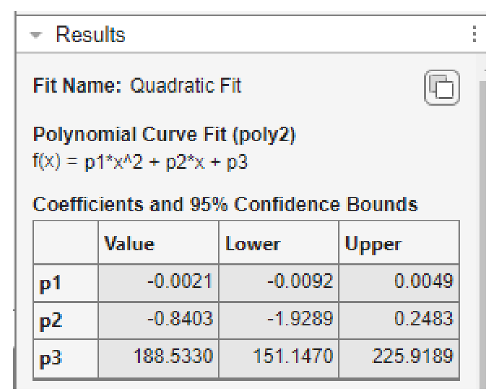
Curve Fitter App

- You can also exclude some data, if it is considered not relevant





- But, then, the coefficient for X^2 is nearly zero -> This means that without this data, the rest of the data fits better to a linear curve.


Results

Fit Name: Quadratic Fit

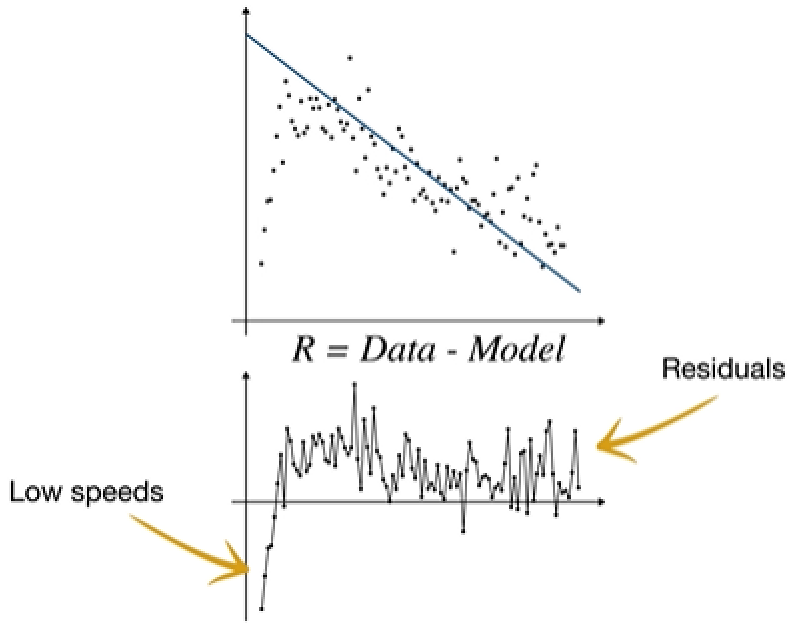
Polynomial Curve Fit (poly2)
f(x) = p1*x^2 + p2*x + p3

Coefficients and 95% Confidence Bounds

	Value	Lower	Upper
p1	-0.0021	-0.0092	0.0049
p2	-0.8403	-1.9289	0.2483
p3	188.5330	151.1470	225.9189

Curve Fitter App

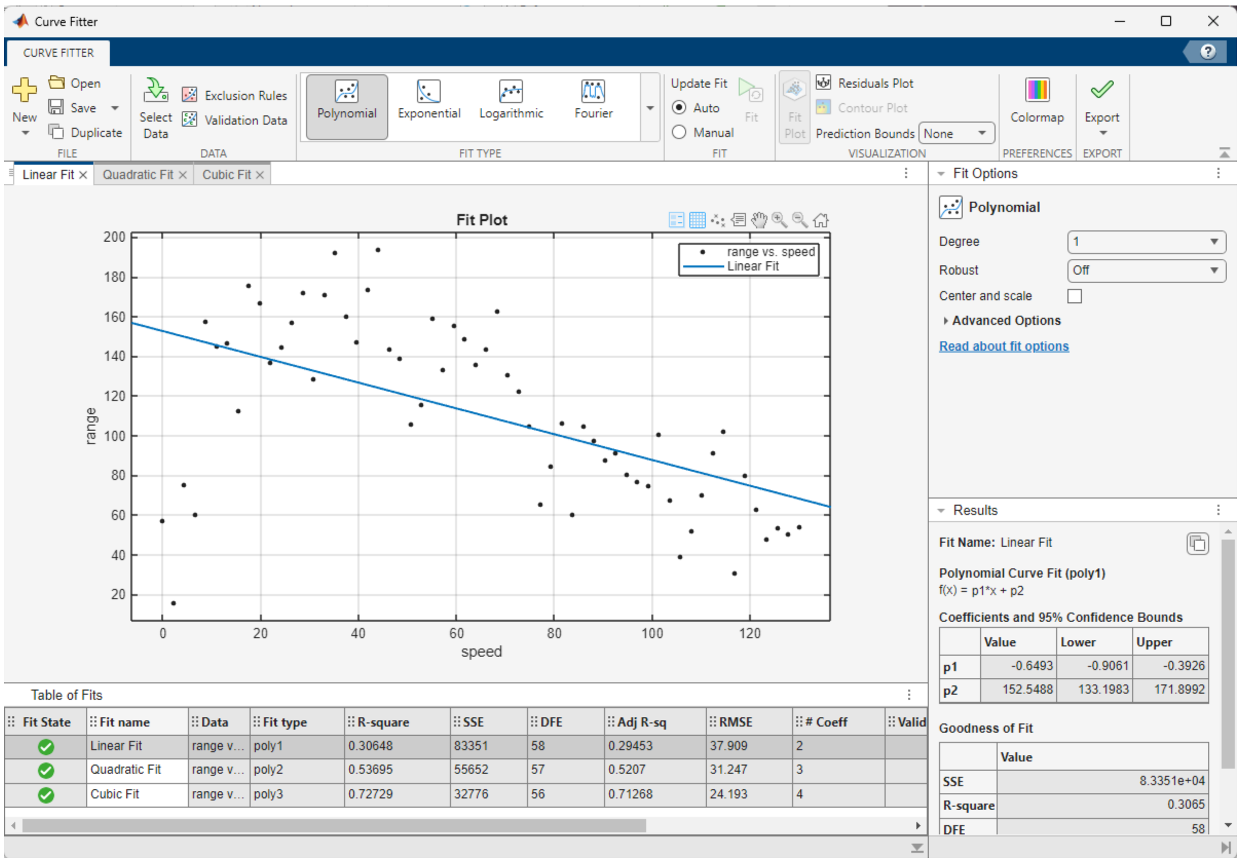
- A more detailed analysis can be done looking to the “residuals”



- For a linear curve, it says that the low speeds are “badly” represented.

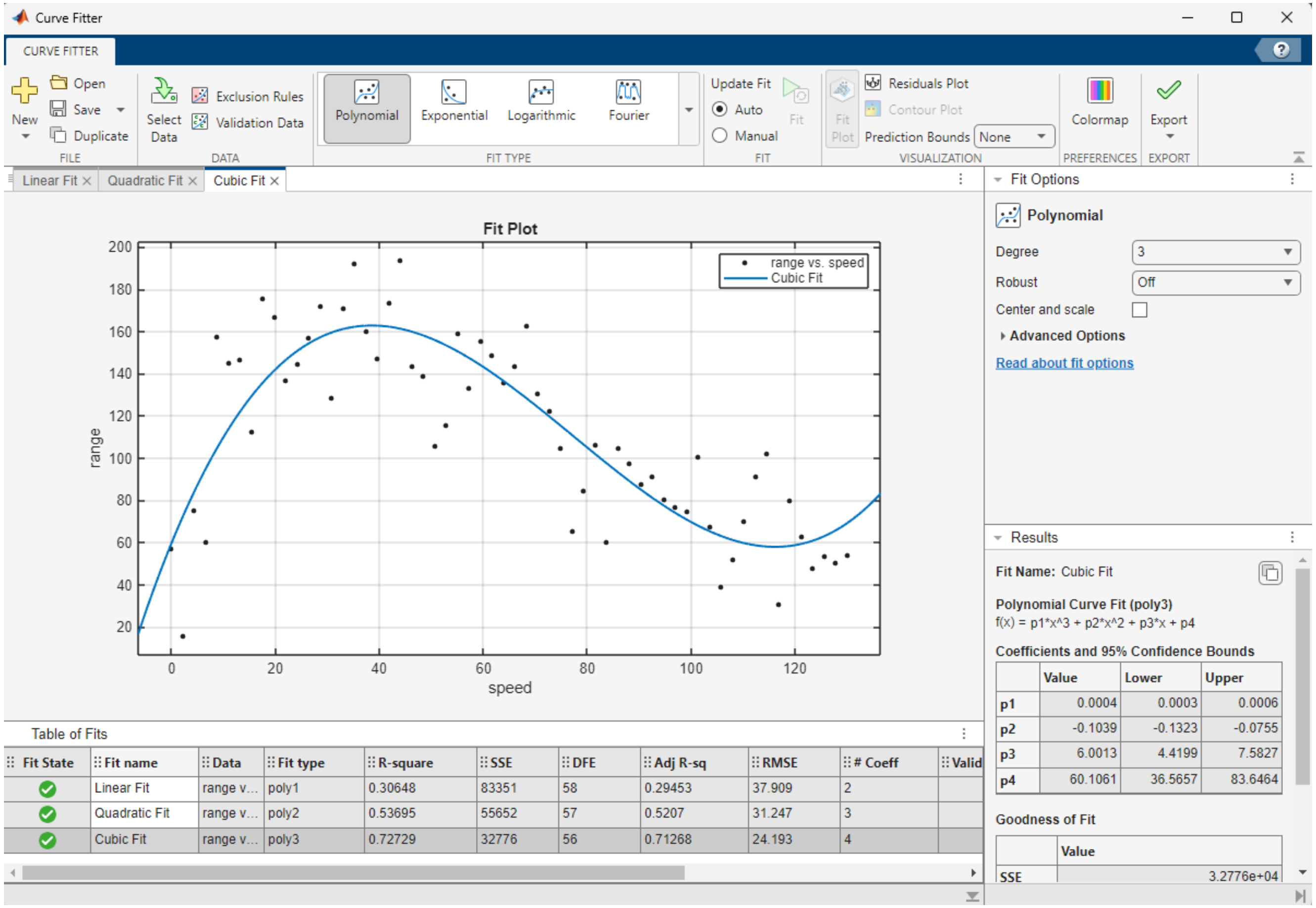
Curve Fitter App

- Let's tray different curves and let's see to the "residuals"
- "Duplicate" at "Fit State" and create to Windows for Linear and Quadratic fit



Curve Fitter App

- And "Cubic" one



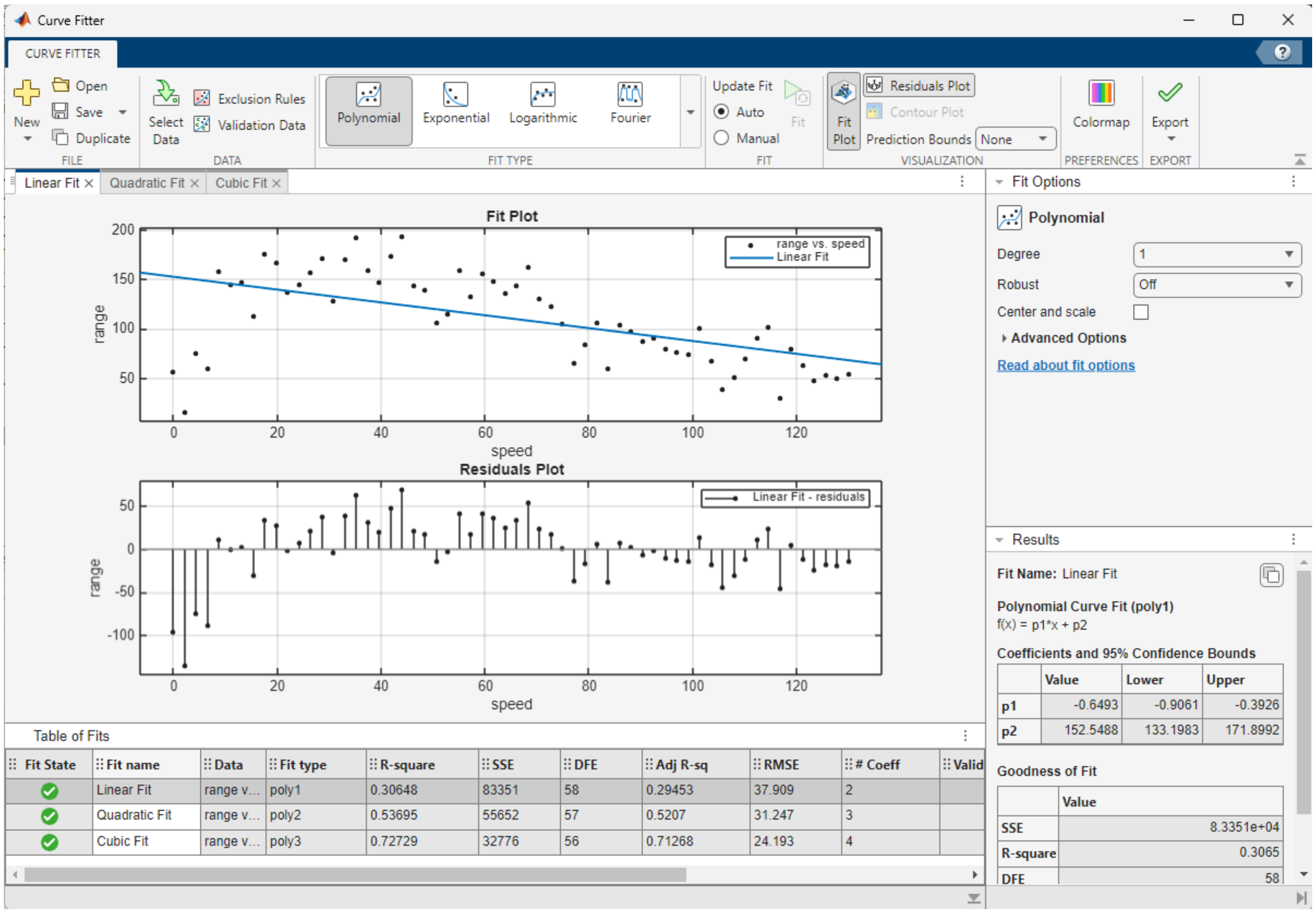
	Value	Lower	Upper
p1	0.0004	0.0003	0.0006
p2	-0.1039	-0.1323	-0.0755
p3	6.0013	4.4199	7.5827
p4	60.1061	36.5657	83.6464

	Value
SSE	3.2776e+04

Fit State	Fit name	Data	Fit type	R-square	SSE	DFE	Adj R-sq	RMSE	# Coeff	Valid
	Linear Fit	range v...	poly1	0.30648	83351	58	0.29453	37.909	2	
	Quadratic Fit	range v...	poly2	0.53695	55652	57	0.5207	31.247	3	
	Cubic Fit	range v...	poly3	0.72729	32776	56	0.71268	24.193	4	

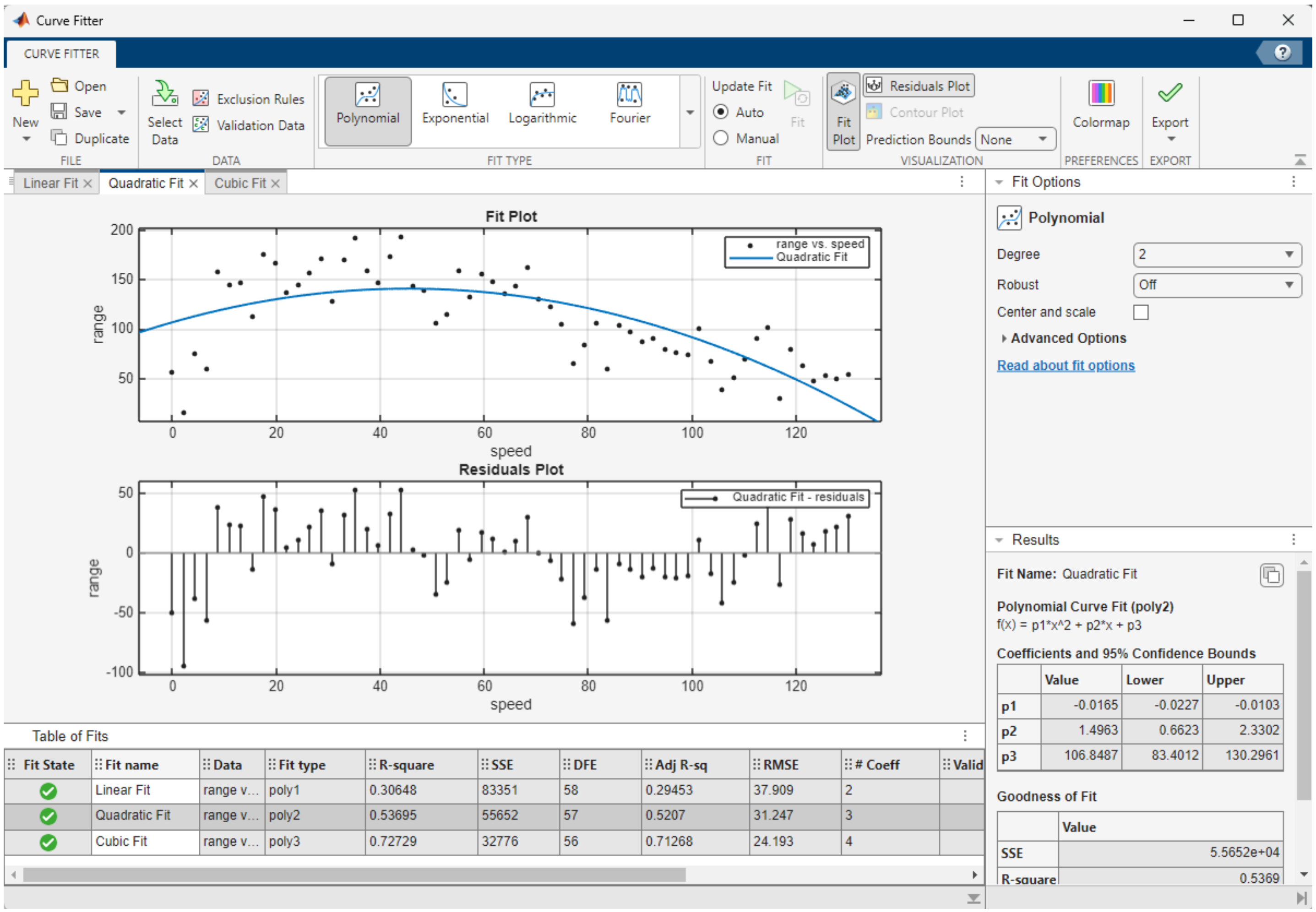
Curve Fitter App

- Now we can see the residuals for “Linear Fit”



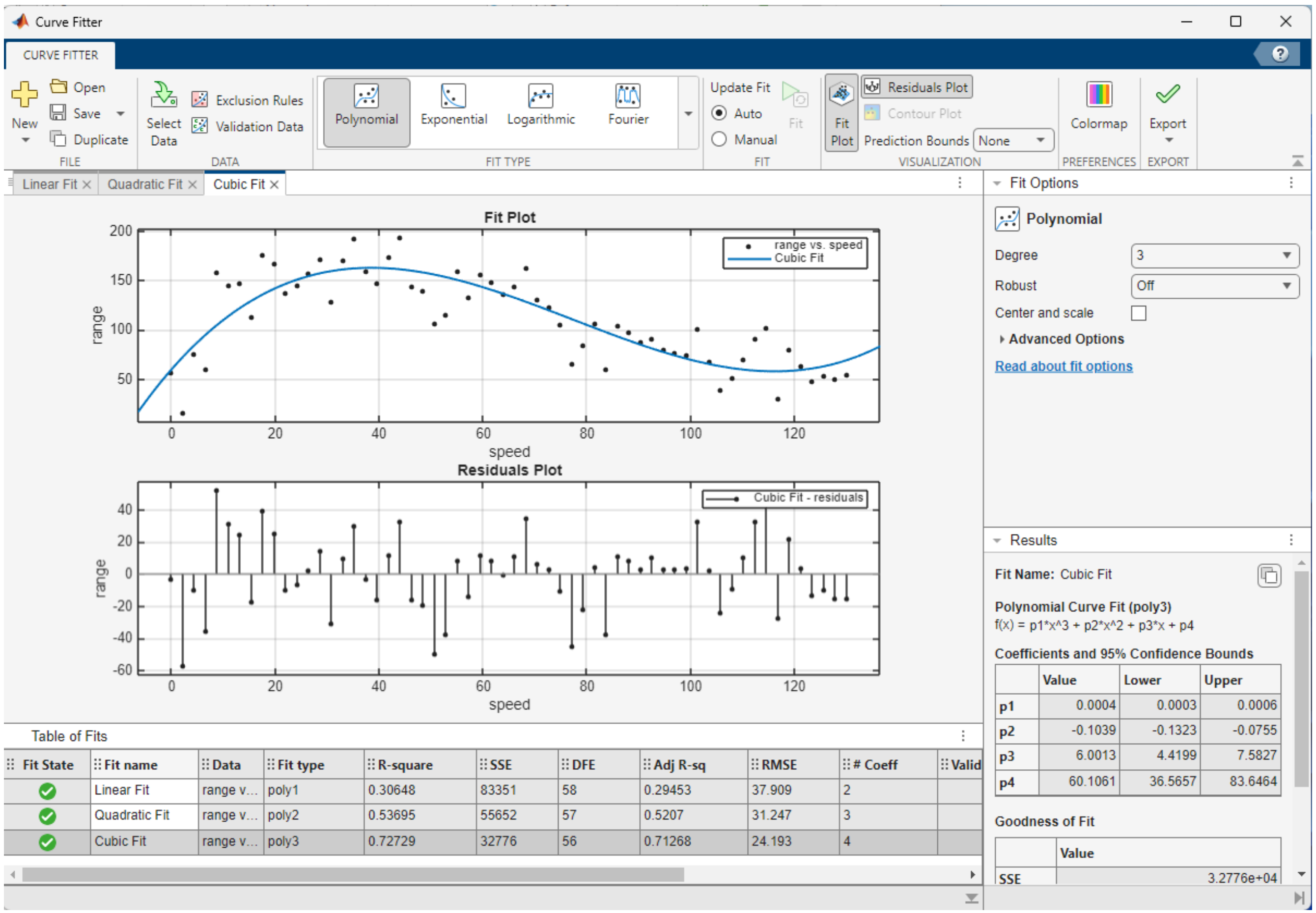
Curve Fitter App

- Now we can see the residuals for "Quadratic Fit"



Curve Fitter App

- And for “Cubic Fit”



	Value	Lower	Upper
p1	0.0004	0.0003	0.0006
p2	-0.1039	-0.1323	-0.0755
p3	6.0013	4.4199	7.5827
p4	60.1061	36.5657	83.6464

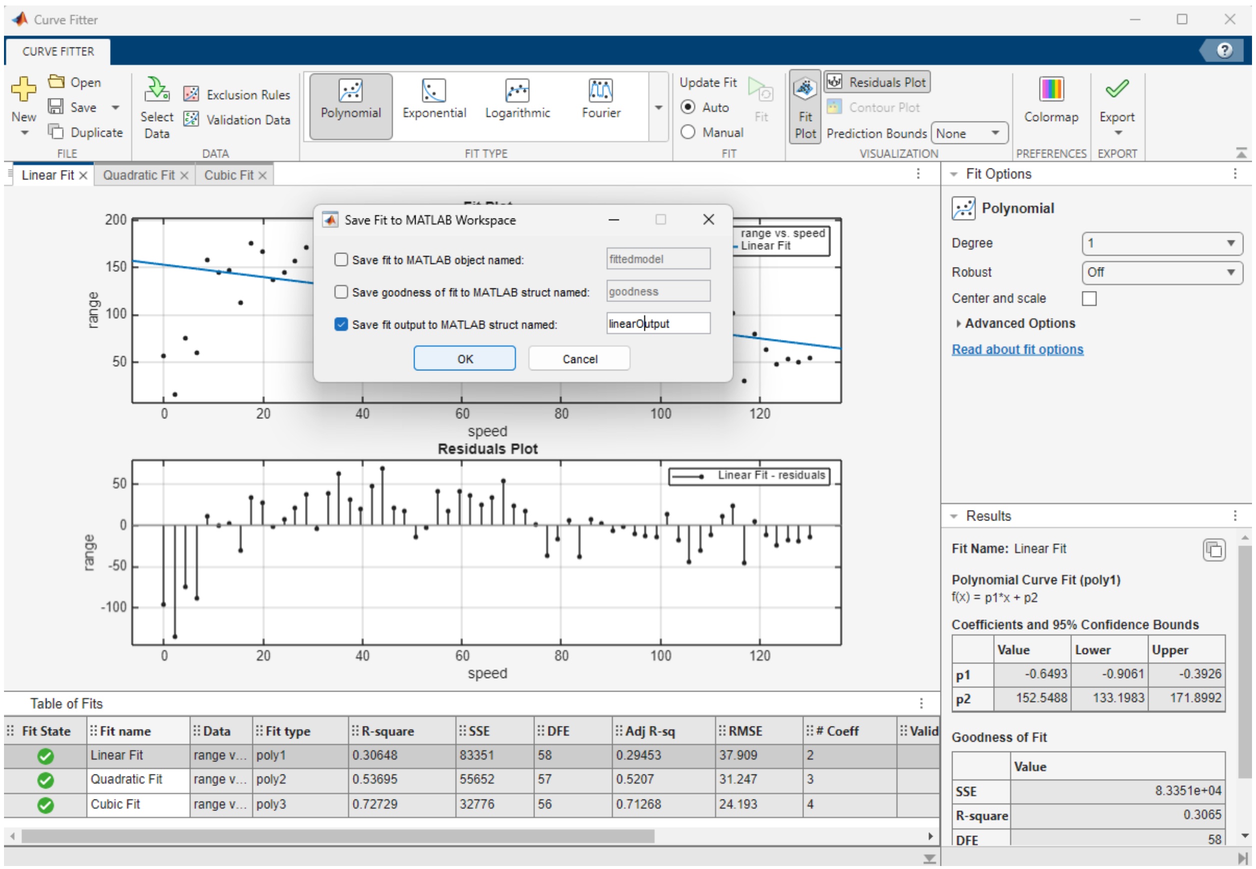
	Value
SSE	3.2776e+04

Fit State	Fit name	Data	Fit type	R-square	SSE	DFE	Adj R-sq	RMSE	# Coeff	Valid
	Linear Fit	range v...	poly1	0.30648	83351	58	0.29453	37.909	2	
	Quadratic Fit	range v...	poly2	0.53695	55652	57	0.5207	31.247	3	
	Cubic Fit	range v...	poly3	0.72729	32776	56	0.71268	24.193	4	

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
School of Professional & Executive Development

Curve Fitter App

- You can also “save” the Linear results to your Matlab “Workspace” as “linearOutput”.



Curve Fitter App

- You can also “save” the Linear results to your Matlab “Workspace” as “linearOutput”.

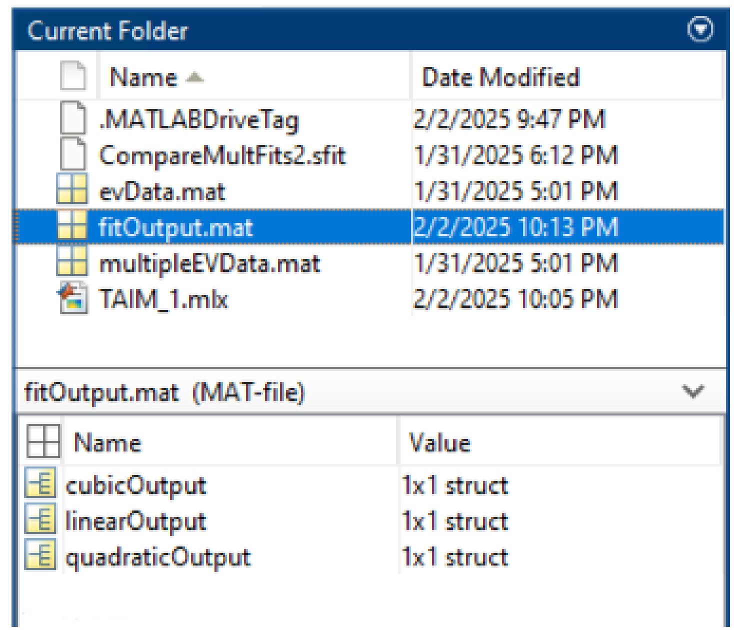
- Repeat “save” the “quadraticOutput”

- Same ”save” the “cubicOutput”

- Save the 3 fits in a file using:

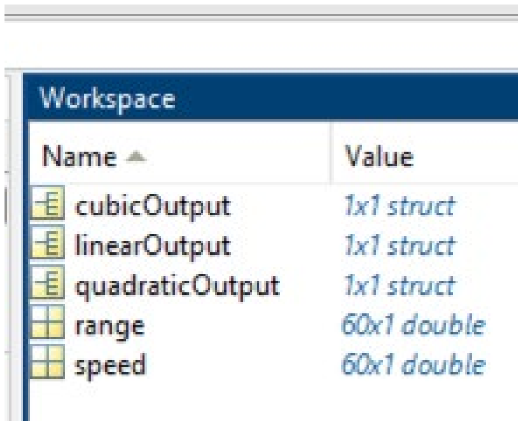
“save fitOutput.mat cubicOutput linearOutput quadraticOutput”

- Now you should have:

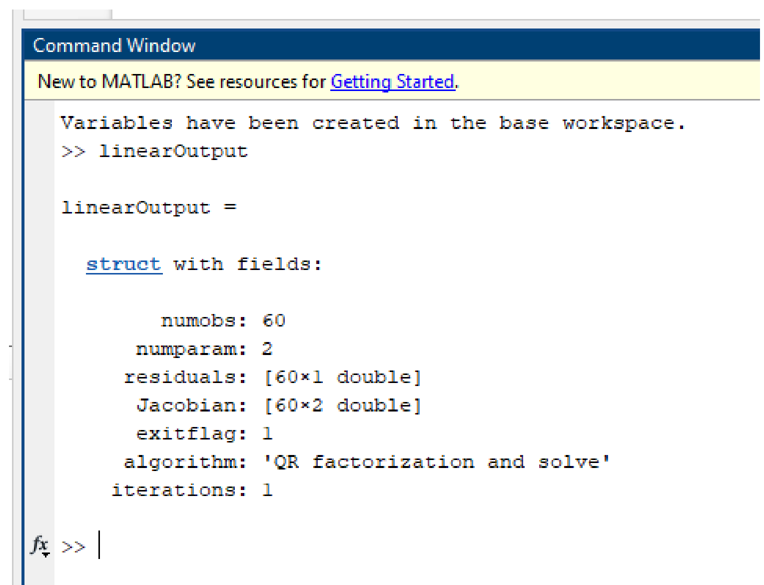


Curve Fitter App

- You can also “save” the Linear results to your Matlab “Workspace”

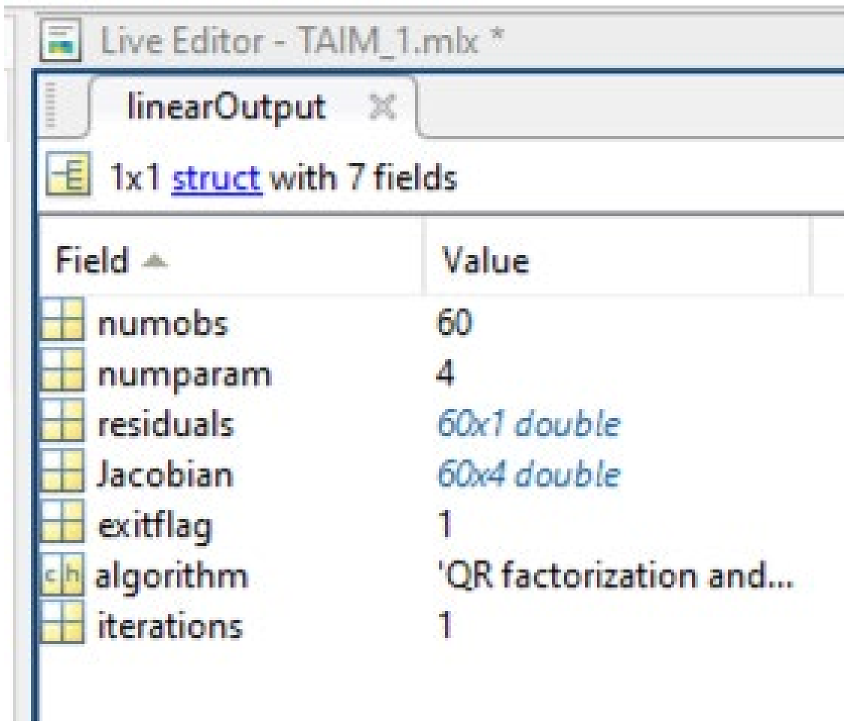


- You can use the “Command” Window.



Curve Fitter App

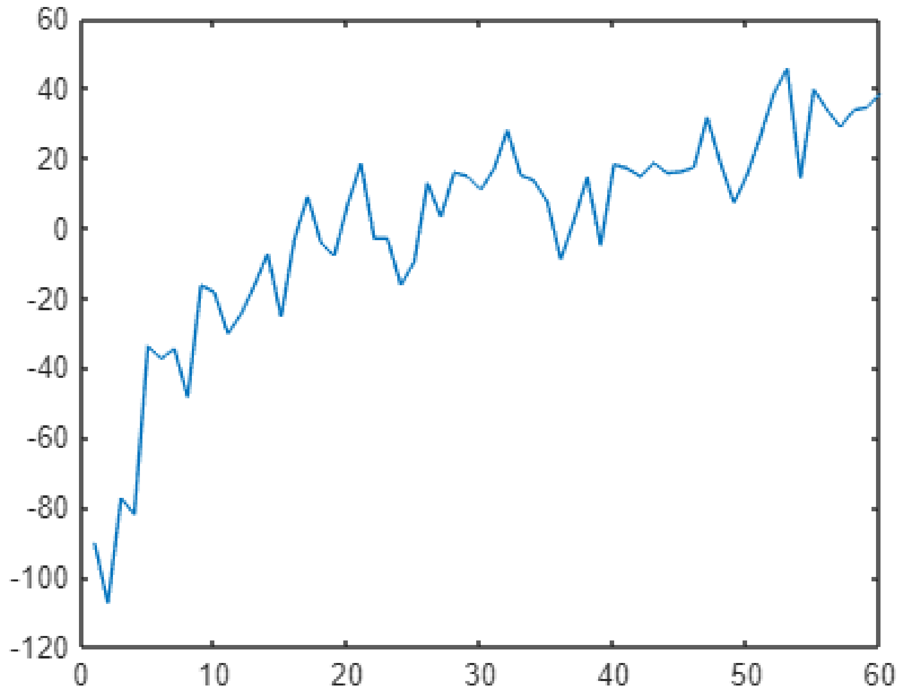
- Output is organized as a Matlab structure variable



Curve Fitter App

- Access to the residuals of the linear fit and make a plot of the residuals…

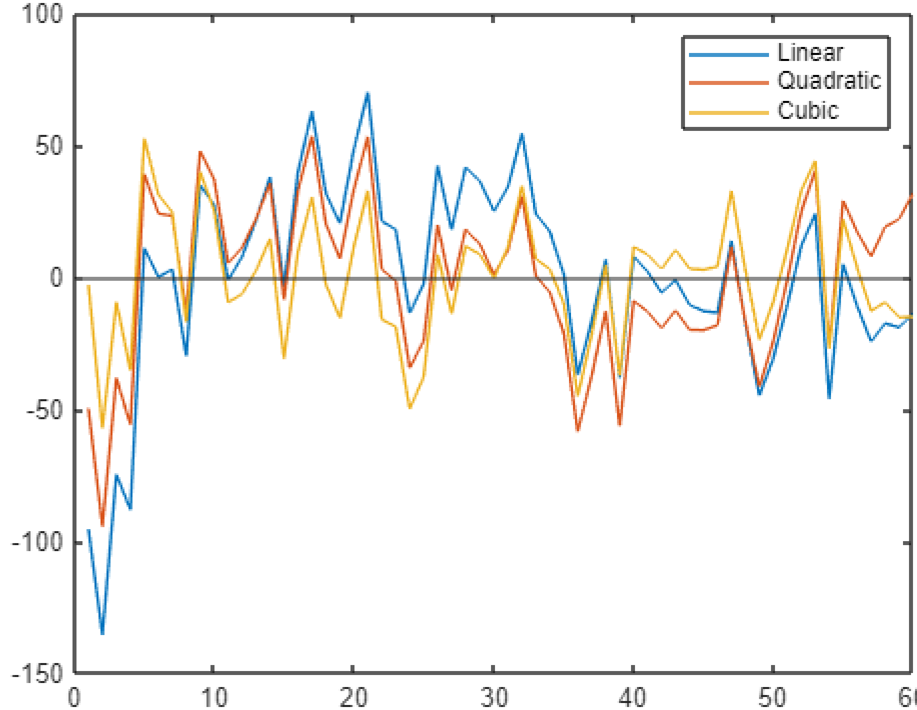
- You should get:



Curve Fitter App

- Using
hold on
plot(x)
hold off

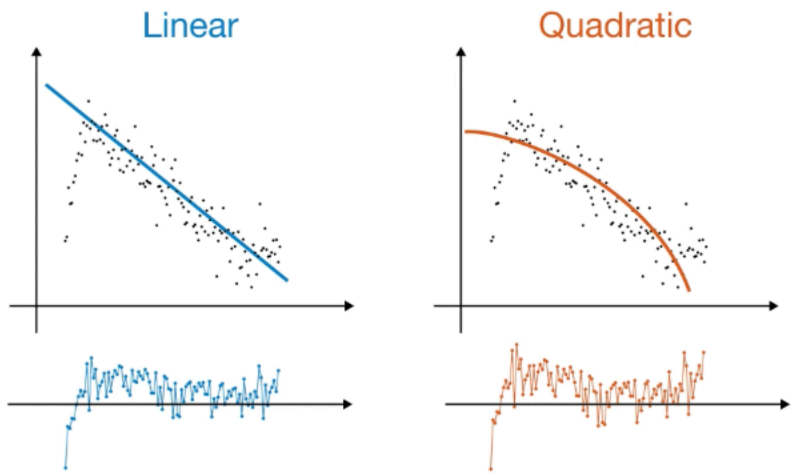
➔ Add plots of the residuals for the quadratic and cubic fits to the existing plot. Adc a legend that labels the plots "Linear", "Quadratic", and "Cubic".
➔ You can also add a line at "0": "yline(0)"



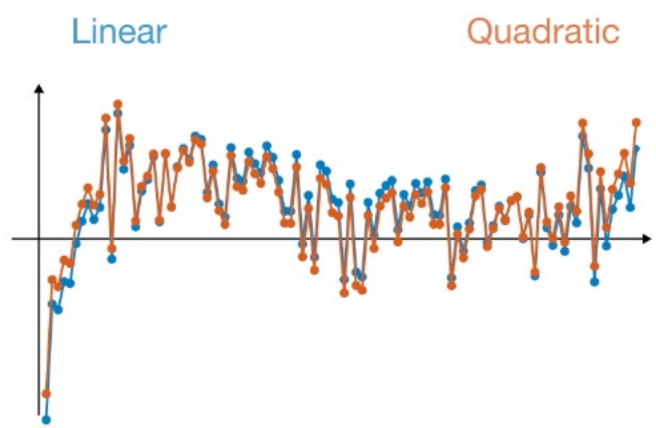
- You should get:

Notice that the fits improve as the residuals move closer and closer to zero. However, this method for comparing different models remains subjective.

Which is the best fit?

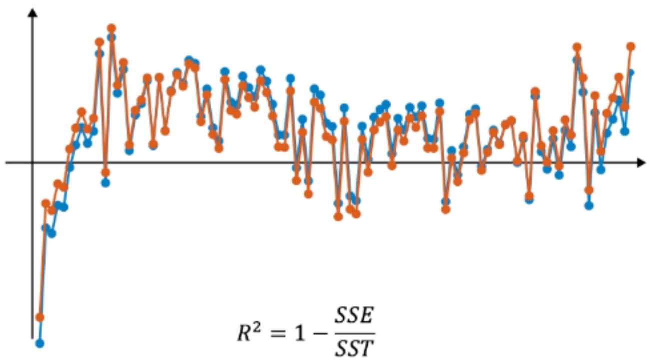


Notice that the fits improve as the residuals move closer and closer to zero. However, this method for comparing different models remains subjective.



Which is the best fit?

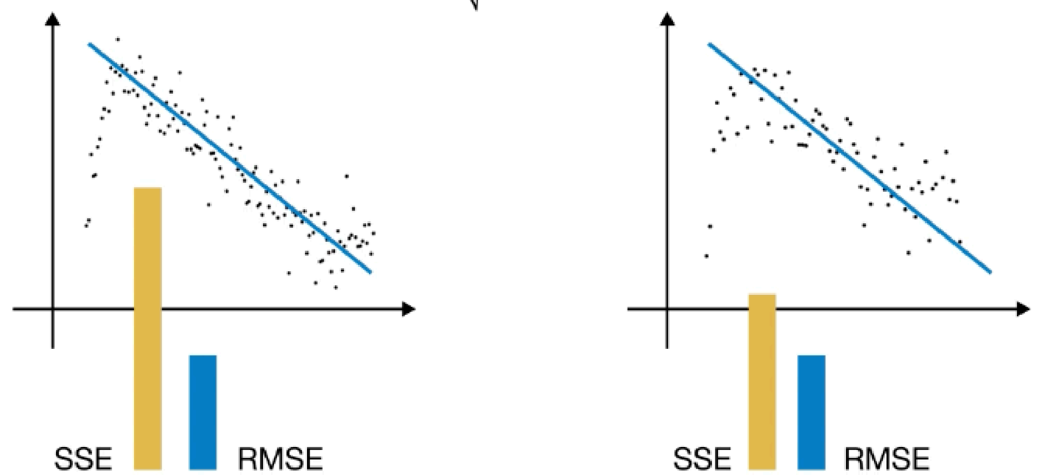
$$SSE = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$



$$R^2 = 1 - \frac{SSE}{SST}$$

Mathematical formulas to quantitatively evaluate the goodness of fit.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N}}$$



Sum Squares Error (SSE) does not counts on that 2 curves may have different number of points.

Root Mean Square Error (RMSE) is better.

Which is the best fit?

All this data automatically calculated by the App

Results

Fit Name: Cubic Fit

Polynomial Curve Fit (poly3)

f(x) = p1*x^3 + p2*x^2 + p3*x + p4

Coefficients and 95% Confidence Bounds

	Value	Lower	Upper
p1	0.0004	0.0003	0.0006
p2	-0.1039	-0.1323	-0.0755
p3	6.0013	4.4199	7.5827
p4	60.1061	36.5657	83.6464

Goodness of Fit

	Value
SSE	3.2776e+04
R-square	0.7273
DFE	56
Adj R-sq	0.7127
RMSE	24.1926

Which is the best fit?

All this data automatically calculated by the App

Table of Fits

Fit State	Fit name	Data	Fit type	R-square	SSE	DFE	Adj R-sq	RMSE	# Coeff
✓	Linear Fit	range v...	poly1	0.30648	83351	58	0.29453	37.909	2
✓	Quadratic Fit	range v...	poly2	0.53695	55652	57	0.5207	31.247	3
✓	Cubic Fit	range v...	poly3	0.72729	32776	56	0.71268	24.193	4

$$R^2 = 1 - \frac{SSE}{SST}$$

$$SSE = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$$

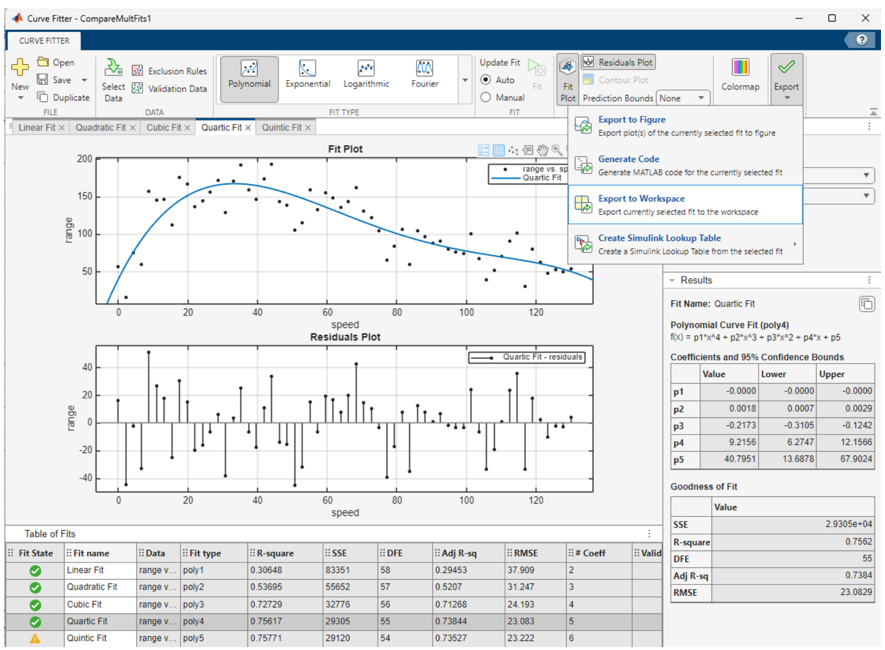
$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N}}$$

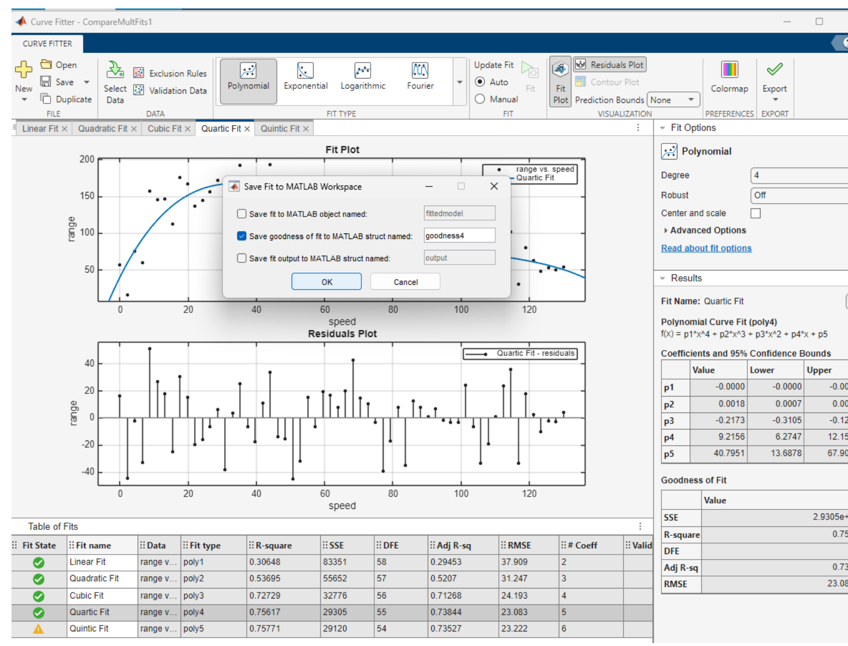
Which is the best fit?

Task:

1 Generate a fourth-order (quartic) fit and a fifth-order (quintic) fit.

2 Export the goodness-of-fit statistics for the quartic fit to a variable named “goodness4”.





Which is the best fit?

See: SSE and R-square of Table of Fits.

- SSE decreases while R-square increases -> indicating that the fits keep improving.
- Adding more terms to the polynomial results in better fits.
- So, should you continue fitting more complicated models with more and more terms?
- It might start fitting random noise in the data instead of just the general trend.
- This result is called **overfitting**.

Table of Fits

Fit State	Fit name	Data	Fit type	R-square	SSE	DFE	Adj R-sq	RMSE	# Coeff
✓	Linear Fit	range v...	poly1	0.30648	83351	58	0.29453	37.909	2
✓	Quadratic Fit	range v...	poly2	0.53695	55652	57	0.5207	31.247	3
✓	Cubic Fit	range v...	poly3	0.72729	32776	56	0.71268	24.193	4
✓	Quartic Fit	range v...	poly4	0.75617	29305	55	0.73844	23.083	5
⚠	Quintic Fit	range v...	poly5	0.75771	29120	54	0.73527	23.222	6

Which is the best fit?

- This result is called **<u>overfitting</u>**.

- One easy way to see if you're overfitting the data is to look at the best fit parameter values.

- Notice that the two lowest order parameters in the fifth-order model are essentially zero. This result can be an indicator that you're overfitting the data.

Task:
<u>Export the goodness-of-fit</u> statistics for ALL fit as “goodness1” to “goodnessX”.

Results

Fit Name: Quintic Fit

⚠ Equation is badly conditioned. Try centering and scaling, or add points at non-repeated x values.

Polynomial Curve Fit (poly5)
f(x) = p1*x^5 + p2*x^4 + p3*x^3 + p4*x^2 + p5*x + p6

Coefficients and 95% Confidence Bounds

	Value	Lower	Upper
p1	0.0000	-0.0000	0.0000
p2	-0.0000	-0.0001	0.0000
p3	0.0032	-0.0016	0.0080
p4	-0.2833	-0.5278	-0.0388
p5	10.4007	5.3790	15.4224
p6	36.2613	4.8738	67.6487

Goodness of Fit

	Value
SSE	2.9120e+04
R-square	0.7577
DFE	54
Adj R-sq	0.7353
RMSE	23.2220

Which is the best fit?

- Once you have saved all the ”goodness1” to “goodness5”

- Generate an overall “goodness” array by:

“A = [array1 array2 array3]”

(“goodness = [goodness1 goodness2 goodness3 goodness4 goodness5]”)

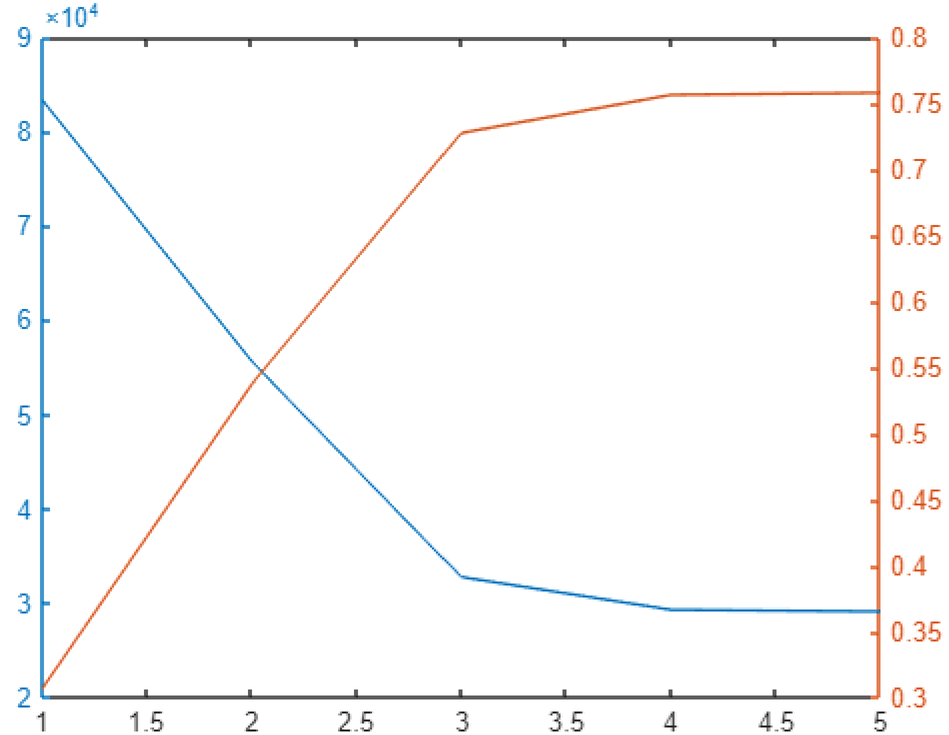
- Typically, handling data is easier if you use tables instead of structures, because you can access using just the names of the columns.

- Convert a struct variable to a table by: “table = struct2table(structure)” -> “goodness = struct2table(goodness)”

Which is the best fit?

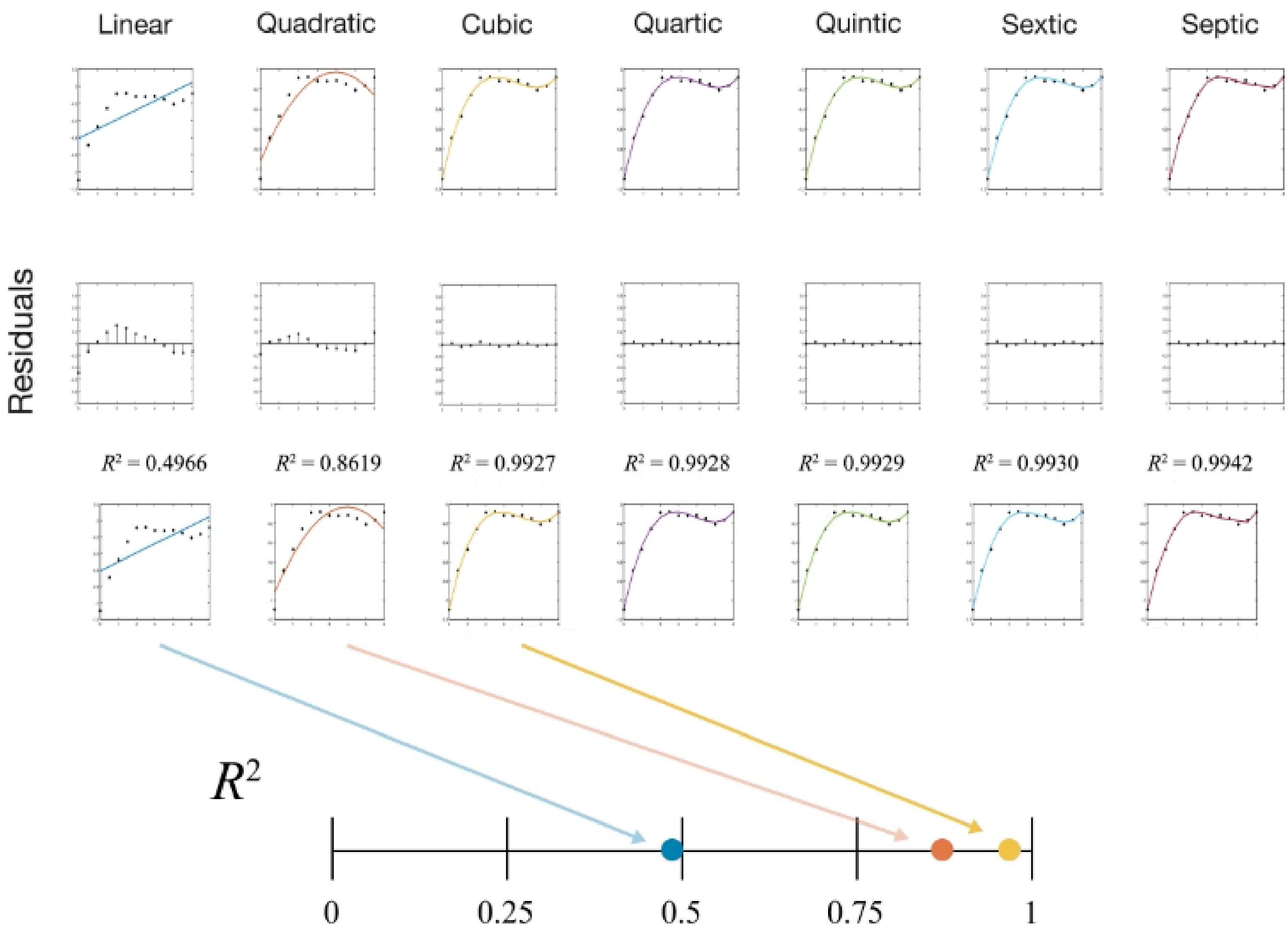
- Using:
"yyaxis command.
yyaxis left
plot(x)
yyaxis right
plot(y)"

- Plot the SSE and R-squared values on the same figure using the yyaxis command. Plot the SSE on the left and the R-squared on the right.

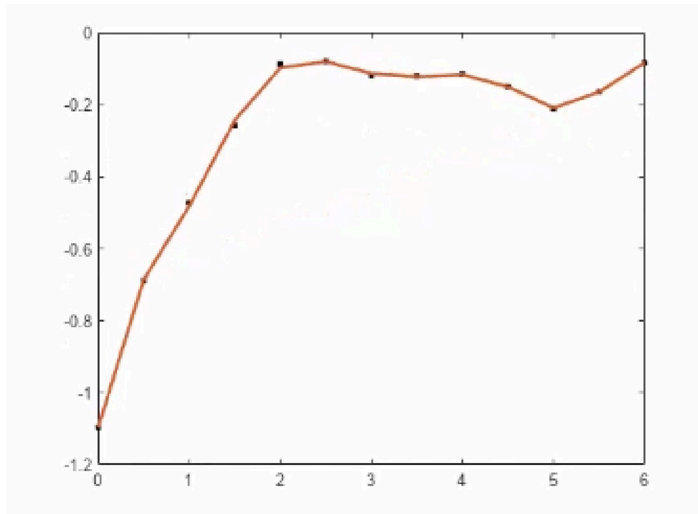


Both the SSE and R-squared look opposite. However, both seem to follow the same pattern. The goodness-of-fit improves significantly until the cubic model and then begins to plateau. The quintic model has the best SSE and R-squared.

Which is the best fit?



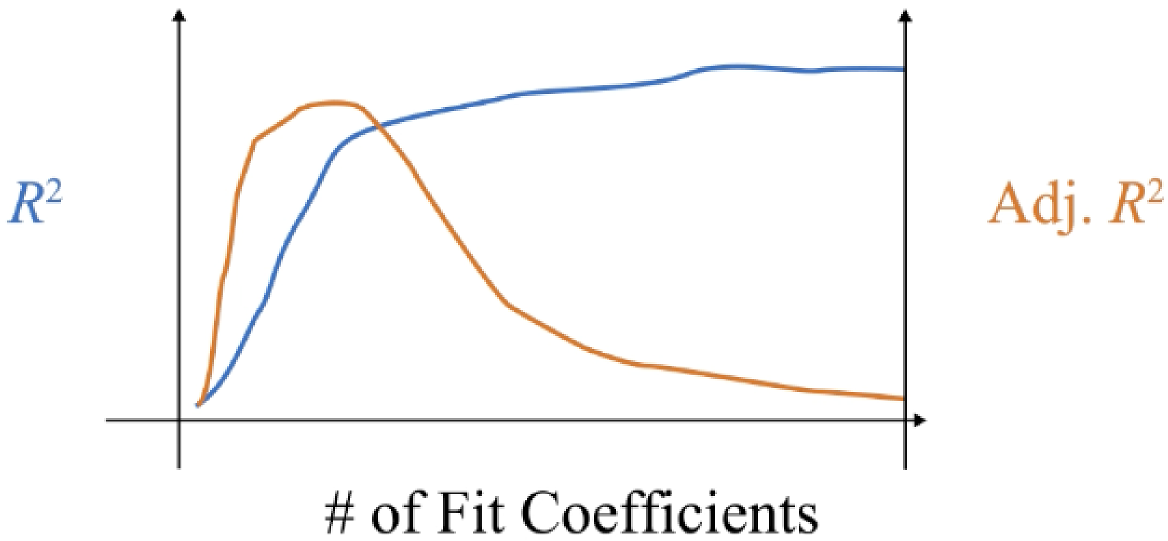
Which is the best fit?



$$y = c_1 + c_2x + c_3x^2 + c_4x^3 + c_5x^4 + c_6x^5 + c_7x^6 + c_8x^7 + c_9x^8 + c_{10}x^9 + c_{11}x^{10} + c_{12}x^{11} + c_{13}x^{12}$$

- Just with the R-squared values, you might be fitting not only the underlying trend, but also the noise in the data.
- This is called **overfitting the data**,
- It can be avoided by using a modified version of the R-squared called the **adjusted R-squared**.

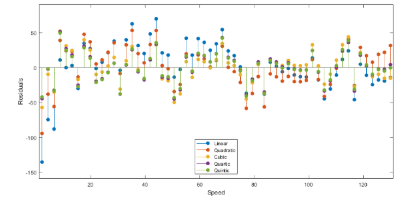
Which is the best fit?

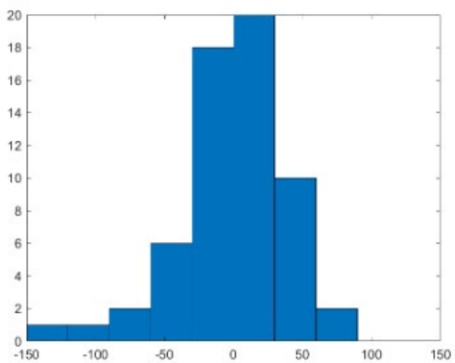


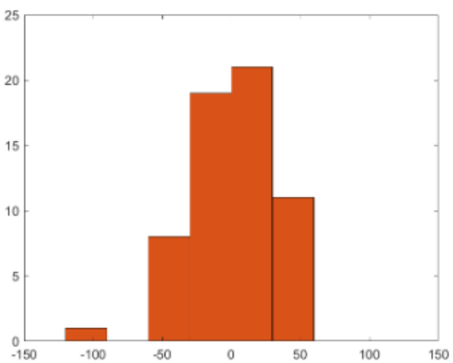
- R-squared increases with the addition of more fit coefficients
- Adjusted R-squared only increases if the more complicated model results in a sufficiently better fit
- The idea is to use a simpler model if possible, or a more complex one that justifies the use of an additional coefficient.

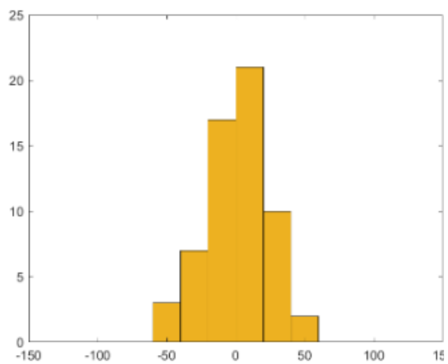
Hints to select the best fit

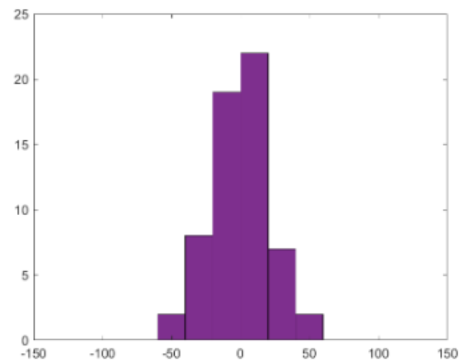
1. Visual inspection of the fits











Linear — and quadratic Residuals
Some significant outliers @ -150.
-> to conclude that these models are not ideal for this particular data set.

Cubic Residuals
No significant outliers -> better fitting.

Quartic Residuals
The residuals are only marginally better compared to cubic model.

Hints to select the best fit

2. Look at the Goodness-of-Fit Statistics

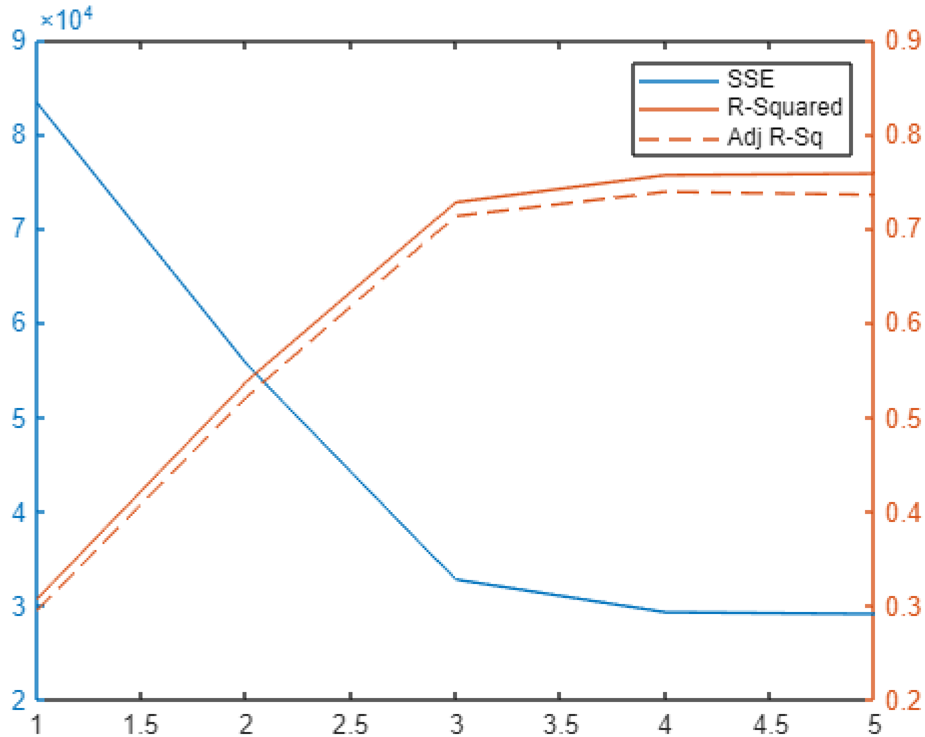
Table of Fits

Fit State	Fit name	Data	Fit type	R-square	SSE	DFE	Adj R-sq	RMSE	# Coeff
✓	Linear Fit	range v...	poly1	0.30648	83351	58	0.29453	37.909	2
✓	Quadratic Fit	range v...	poly2	0.53695	55652	57	0.5207	31.247	3
✓	Cubic Fit	range v...	poly3	0.72729	32776	56	0.71268	24.193	4
✓	Quartic Fit	range v...	poly4	0.75617	29305	55	0.73844	23.083	5
⚠	Quintic Fit	range v...	poly5	0.75771	29120	54	0.73527	23.222	6

- SSE and R-squared almost always increase continually for models with an increasing number of fitting coefficients.
- How much they improve? -> This can indicate if starting to overfit data

Hints to select the best fit

2. Look at the Goodness-of-Fit Statistics



- Adjusted R^2 does not continually increase because it penalizes models for their number of fit coefficients.

Hints to select the best fit

3. Look at the Best Fit Coefficients and Confidence Bounds

If a model contains any unnecessary fit coefficients, → the best fit values of those coefficients will likely be very close to zero (if the 95% confidence bounds contain a negative lower bound and a positive upper bound).

(The confidence bounds indicate that the best fit value for a fit coefficient has a 95% chance of lying inside that range.)

Results

Fit Name: Cubic Fit

Polynomial Curve Fit (poly3)
f(x) = p1*x^3 + p2*x^2 + p3*x + p4

Coefficients and 95% Confidence Bounds

	Value	Lower	Upper
p1	0.0004	0.0003	0.0006
p2	-0.1039	-0.1323	-0.0755
p3	6.0013	4.4199	7.5827
p4	60.1061	36.5657	83.6464

Goodness of Fit

	Value
SSE	3.2776e+04
R-square	0.7273
DFE	56
Adj R-sq	0.7127
RMSE	24.1926

Results

Fit Name: Quartic Fit

Polynomial Curve Fit (poly4)
f(x) = p1*x^4 + p2*x^3 + p3*x^2 + p4*x + p5

Coefficients and 95% Confidence Bounds

	Value	Lower	Upper
p1	-0.0000	-0.0000	-0.0000
p2	0.0018	0.0007	0.0029
p3	-0.2173	-0.3105	-0.1242
p4	9.2156	6.2747	12.1566
p5	40.7951	13.6878	67.9024

Results

Fit Name: Quintic Fit

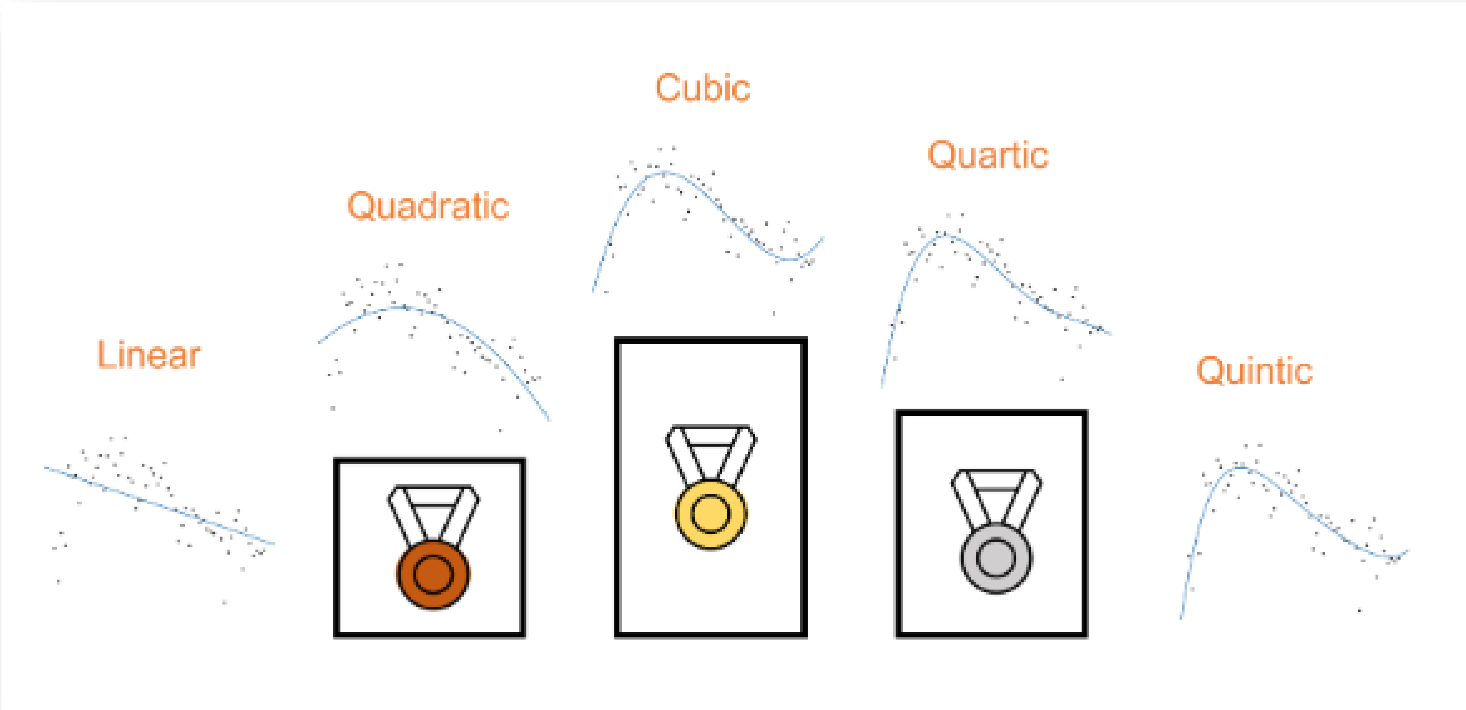
⚠ Equation is badly conditioned. Try centering and scaling, or add points at non-repeated x values.

Polynomial Curve Fit (poly5)
f(x) = p1*x^5 + p2*x^4 + p3*x^3 + p4*x^2 + p5*x + p6

Coefficients and 95% Confidence Bounds

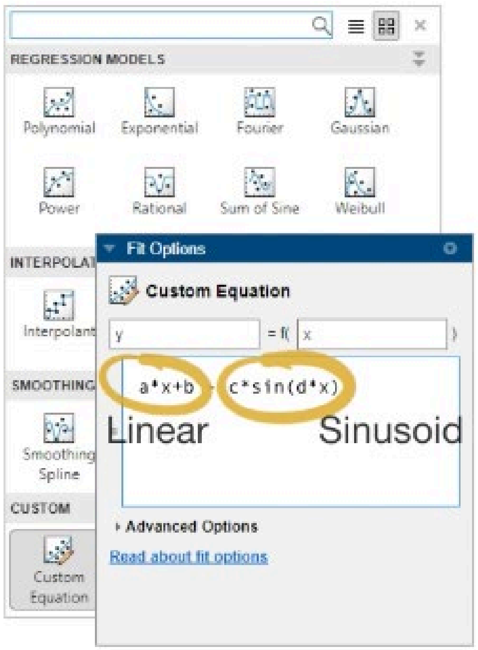
	Value	Lower	Upper
p1	0.0000	-0.0000	0.0000
p2	-0.0000	-0.0001	0.0000
p3	0.0032	-0.0016	0.0080
p4	-0.2833	-0.5278	-0.0388
p5	10.4007	5.3790	15.4224
p6	36.2613	4.8738	67.6487

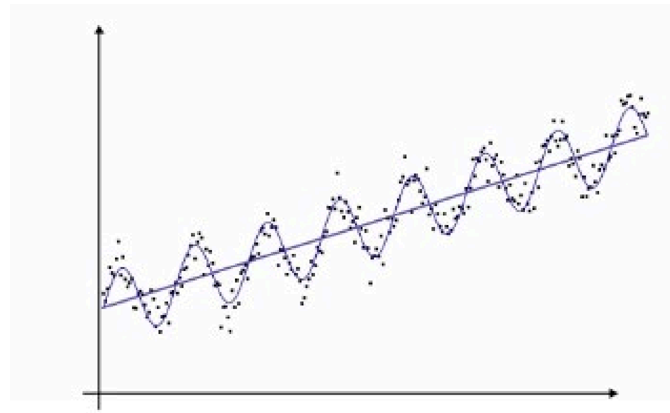
Which model do you rank as the best?

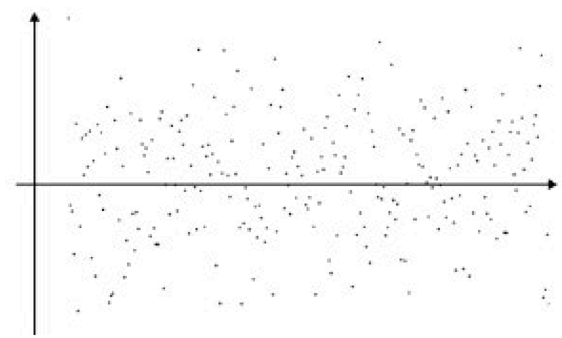


- The cubic model is the first to have best fit coefficients that are not microscopically small.
- It has goodness-of-fit statistics that are significantly better than those of the quadratic model and very close to those of the more complicated quartic and quintic models.
- In doubt, choose the simpler model

Fitting your own custom model?

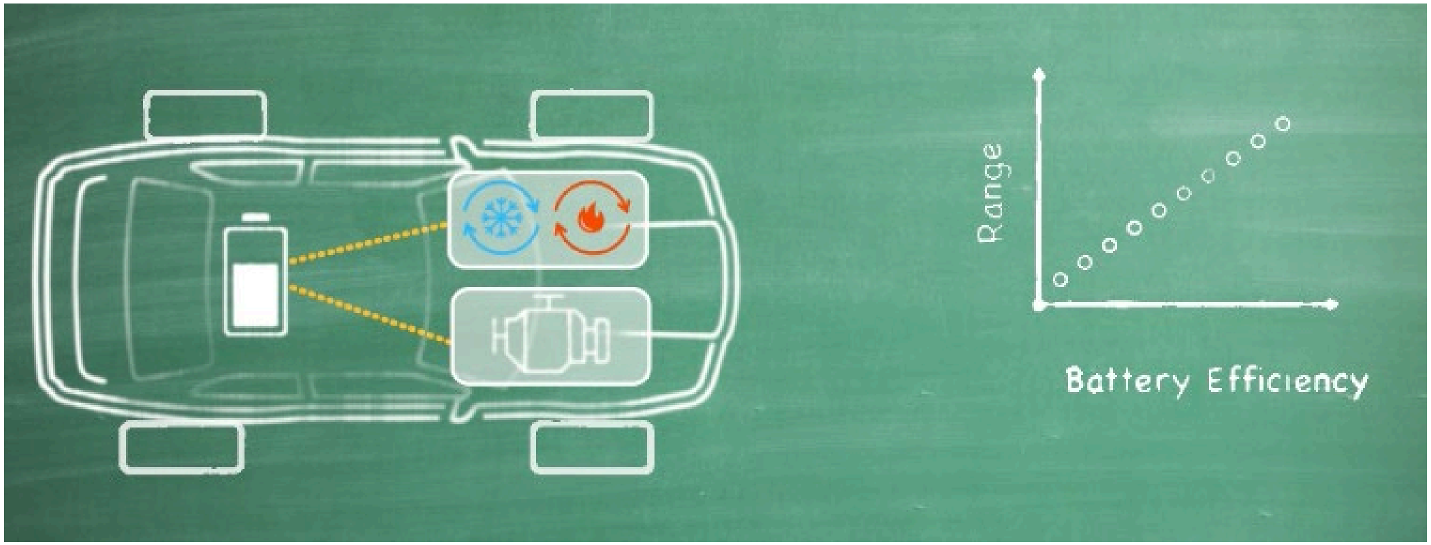






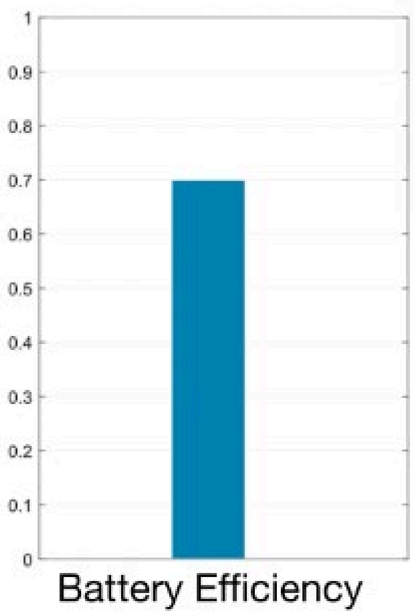
- Imagine you have a complex data that is not one of the common curves provided by Matlab.

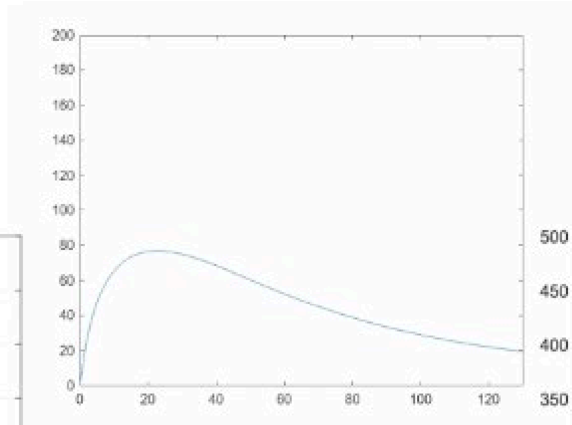
Fitting your own custom model?

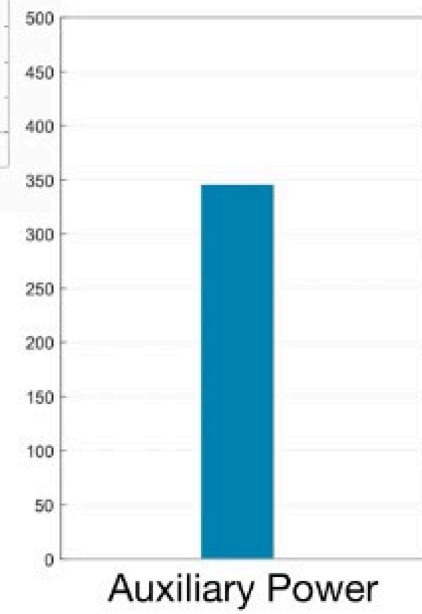


- Or you have a better physical model of the data, e.g. for an electrical vehicle, it is not only the consumption of the motor to move, but also,…

Fitting your own custom model?



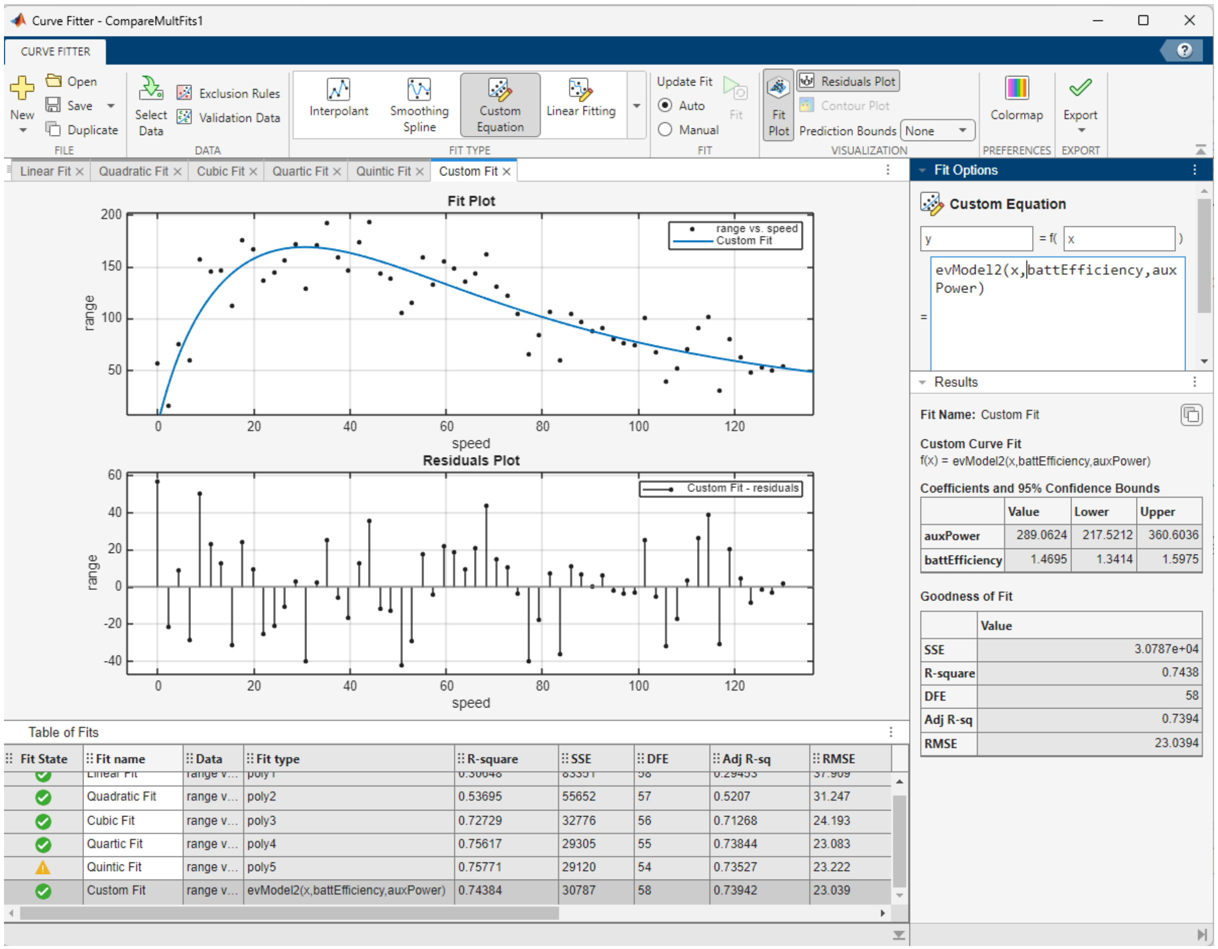




- Or you have a better physical model of the data, e.g. for an electrical vehicle, it is not only the consumption of the motor to move, but also,…
  on the “Auxiliary power” consumption for air-conditioned, heating, music,…

Fitting your own custom model?

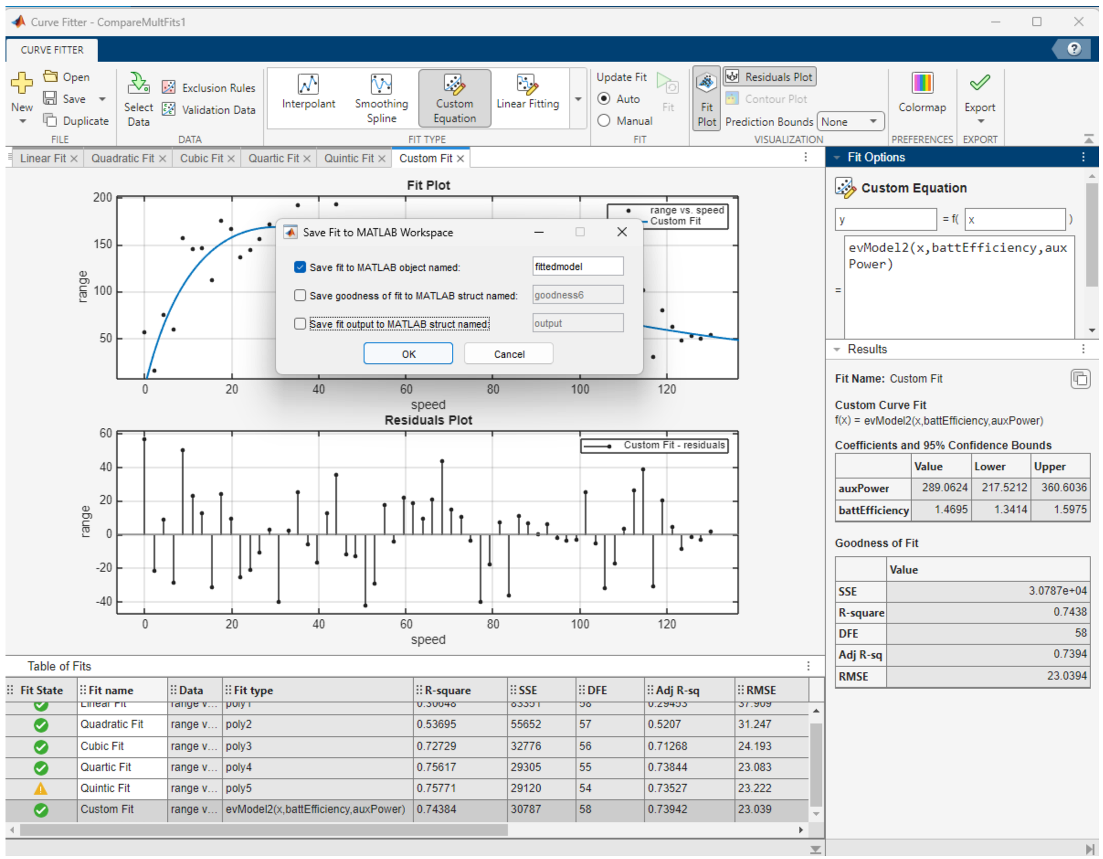
- Follow the instructions of the Prof to build a function-based “interactive” model in your Matlab. -> You fill be able to implement the custom fit, as:



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
School of Professional & Executive Development

Fitting your own custom model?

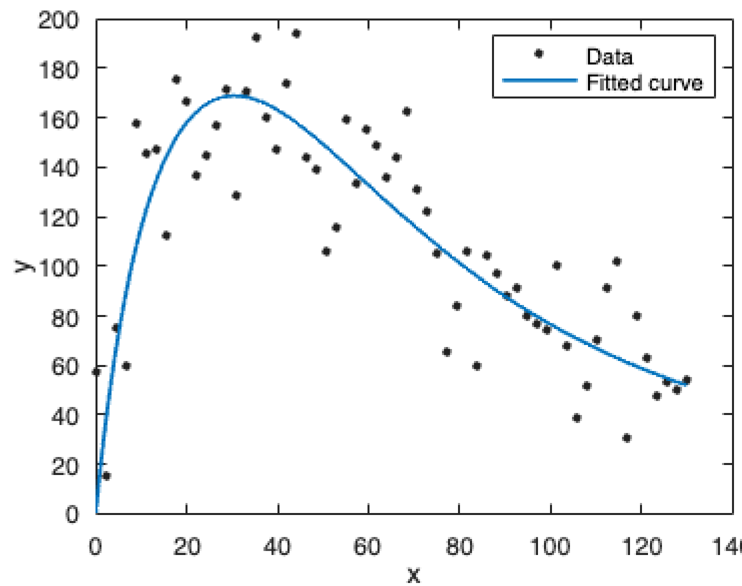
- Once achieved, save the “fittedmodel” at the MATLAB Workspace
- Save the fittedmodel to the HD by: “save fittedmodel.mat fittedmodel”



Uses of a fit

- Now “fittedmodel” is a more complex variable of type “object”
- You can just run “fittedmodel” in Command Window to see what’s that.
- And you can “plotplot(fittedmodel,speed,range)” to see the fit.

```
fittedmodel =
     General model:
     fittedmodel(x) = evModel2(x,battEfficiency,auxPower)
     Coefficients (with 95% confidence bounds):
       auxPower =       289.1  (217.5, 360.6)
       battEfficiency =       1.469  (1.341, 1.598)
```



Uses of a fit

1. You can get values of “fittedmodel” as a function!

    1. “newPoint = feval(fittedmodel,150)”
    2. You can get the coefficients of the function
    3. You can see all “methods” of the “fittedmodel” by: “methods(fittedmodel)”

```
>> methods(fittedmodel)

Methods for class cfit:

argnames      coeffnames    dependnames    fitoptions    integrate    numcoeffs    probnames    type
category      coeffvalues   differentiate  formula       islinear     plot         probvalues
cfit          confint       feval          indepnames    numargs      predint      setoptions
```

Uses of a fit

- Now you can use the fit e.g. for finding the optimal speed:

  Option 1: a) find the index of the maximum of the fitted model, b) index into the speed variable.

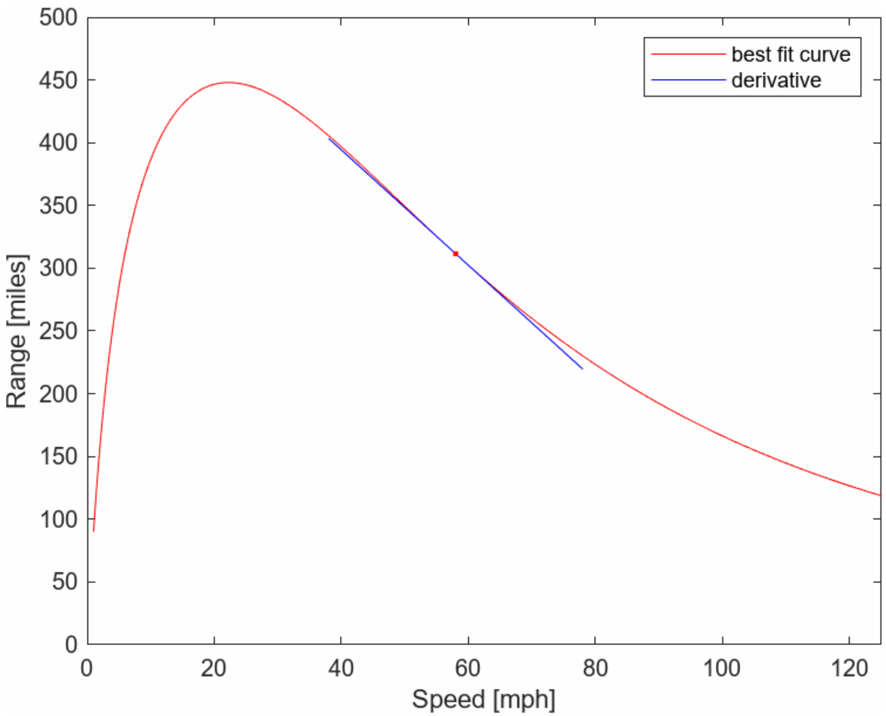
  [maxVal,idx] = max(fittedmodel)
  speedBest = speed(idx)

  Only a certain number of functions are compatible with fit objects, so you will need specifically:

  “ValsTest = feval(fittedmodel,speed);
  [maxVal,idx] = max(ValsTest)
  speedBest = speed(idx)”

Uses of a fit

- Now you can use the fit e.g. for finding the optimal speed:



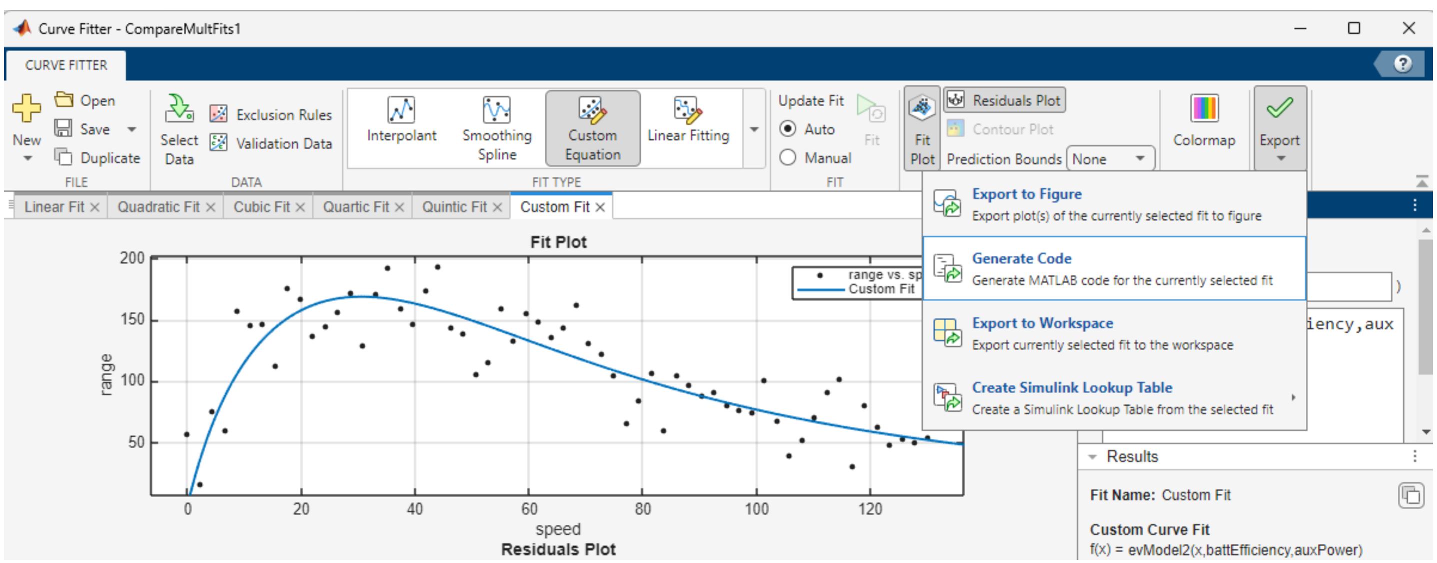
Option 2:

Alternate approach -> calculating slope = 0 -> find the corresponding speed.

“d = differentiate(fittedmodel,speed)
plot(speed,abs(d))
[minVal, idx] = min(abs(d));
speedBest = speed(idx)
rMax =
feval(fittedmodel,speedBest)”

Uses of a fit

- Now let's analyse MULTIPLE Vehicles in an automatic way
- This is said a "Curve Fitting programmatically"
- TASK:
  - "Export" to "Generate Code" and save it as the generated function (I will explain)

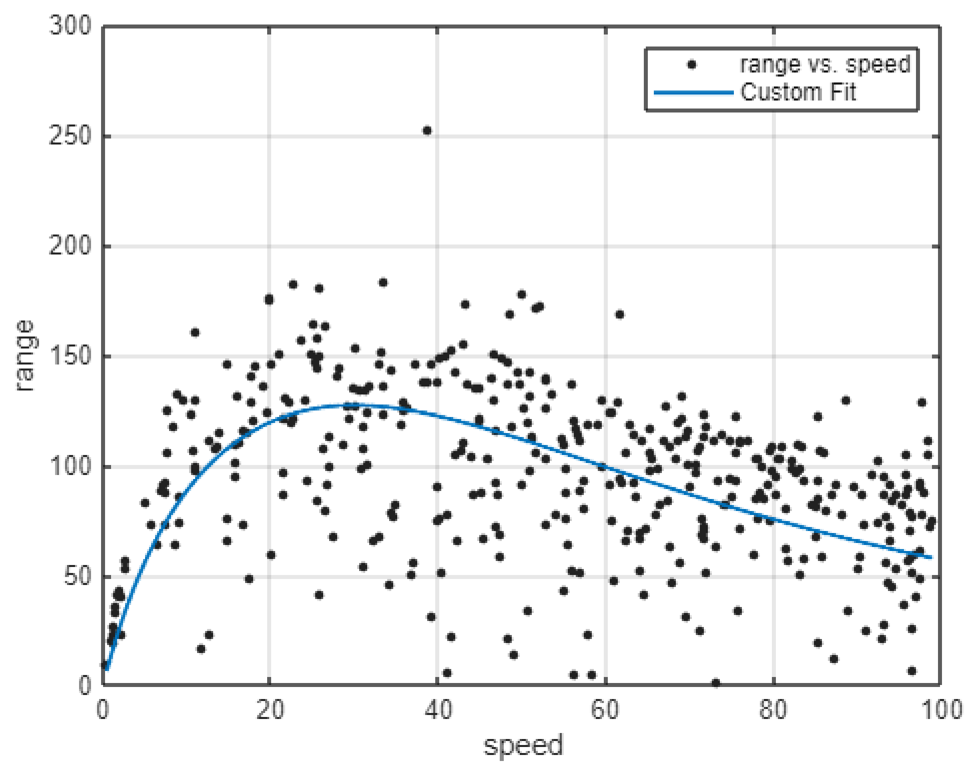


Uses of a fit

- TASK:
  1. “Export” to “Generate Code” and save it as the generated function (I will explain)
  2. “load multipleEVData”
  3. TEST it with for some of the vehicles

  “carNumber = SLIDER (1 to 50)
  createFit(speedData{carNumber},rangeData{carNumber})
  coeff = coeffvalues(fittedmodel)”

```
carNumber = 20
```




```
fittedmodel =

     General model:
     fittedmodel(x) = evModel2(x,battEfficiency,auxPower)
     Coefficients (with 95% confidence bounds):
       auxPower =       370.9  (302.1, 439.7)
```

Uses of a fit

- TASK:

  4. Modify in a “2” version and: a) remove the part of “% Create a figure for the plots.” till the end (We do not want now to see 1 figure for each vehicle); b) Add “speedBest, maxRange” to the output variables of the function.

```
function [fitresult, gof, speedBest, maxRange] = createFit2(speed, range)
%CREATEFIT(SPEED,RANGE)
%  Create a fit.
%
%  Data for 'Custom Fit' fit:
%      X Input: speed
%      Y Output: range
%  Output:
%      fitresult : a fit object representing the fit.
%      gof : structure with goodness-of-fit info.
```

Uses of a fit

- Finally:

```
“nCars = 100;
coeffs = zeros(nCars,2);
speedBest = zeros(nCars,1);
maxRange = zeros(nCars,1);

for carNumber = 1:nCars
[fittedmodel, ~, speedBest(carNumber), maxRange(carNumber)] =
createFit2(speedData{carNumber},rangeData{carNumber});
coeffs(carNumber,:) = coeffvalues(fittedmodel);
end

...
```

Uses of a fit

- Finally:

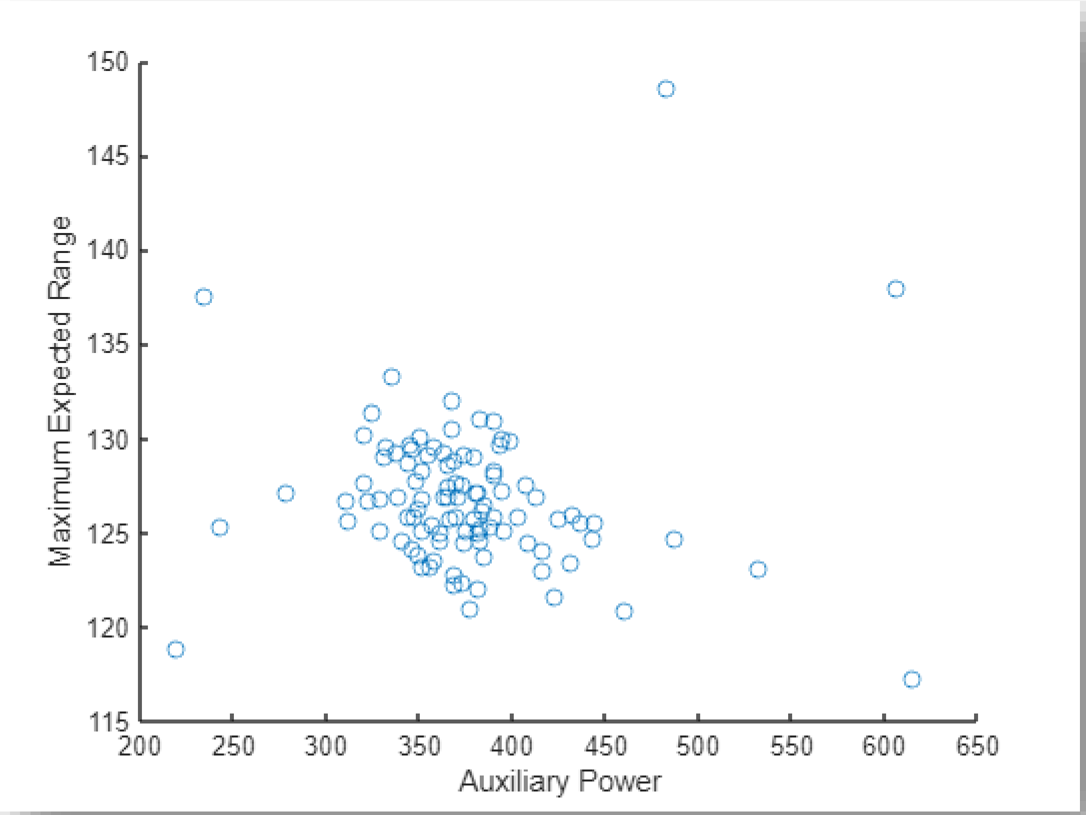
"…

```
figure
scatter(coeffs(:,1),maxRange)
xlabel("Auxiliary Power")
ylabel("Maximum Expected Range")
figure
scatter(coeffs(:,2),maxRange)
xlabel("Battery Efficiency")
ylabel("Maximum Expected Range")
figure
histogram(speedBest)
xlabel("Most Efficient Driving Speed")"
```

Uses of a fit

- **We process the data of 100 cars in seconds!!**

- OK, most of the users take about 350-400 W for auxiliary uses as air-conditioning, music, etc.

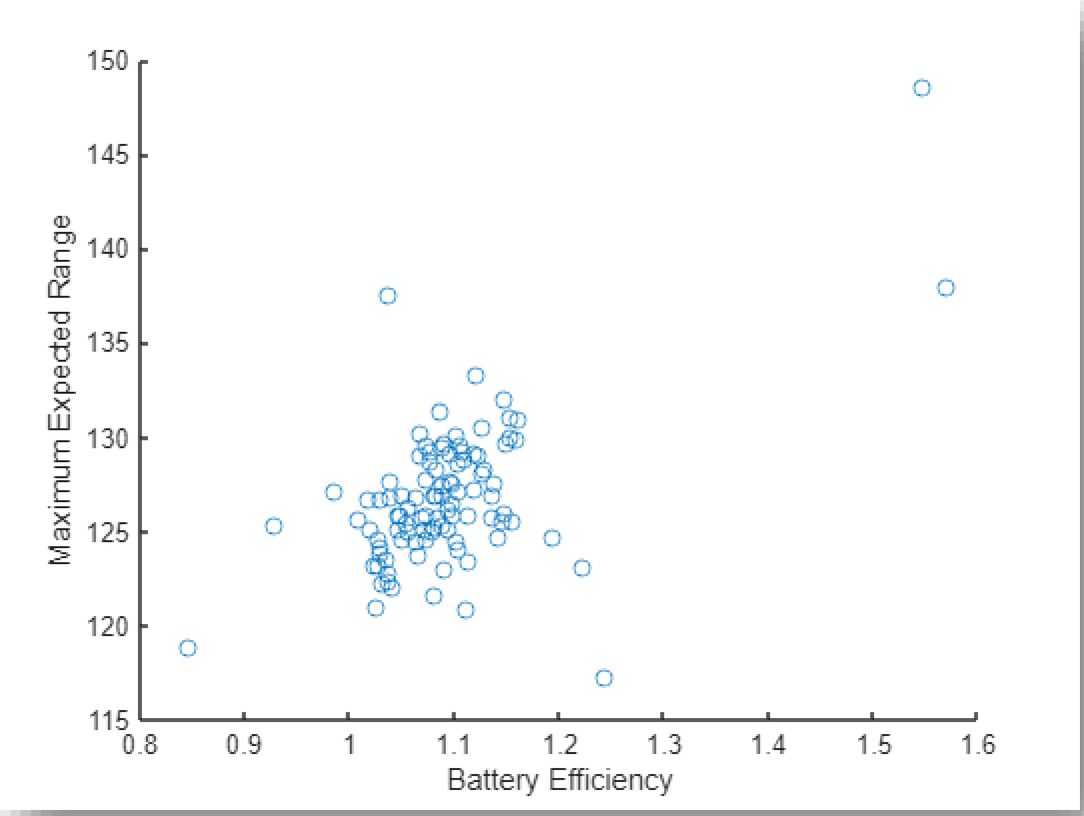


Uses of a fit

- **We process the data of 100 cars in seconds!!**
- Ups!! There is an electrical vehicle with a really poor Battery Efficiency of only aprox. 0.85
- Who's that?

"[minVal, idx] = min(coeffs(:,2))" →

minVal = 0.8473
idx = 32)



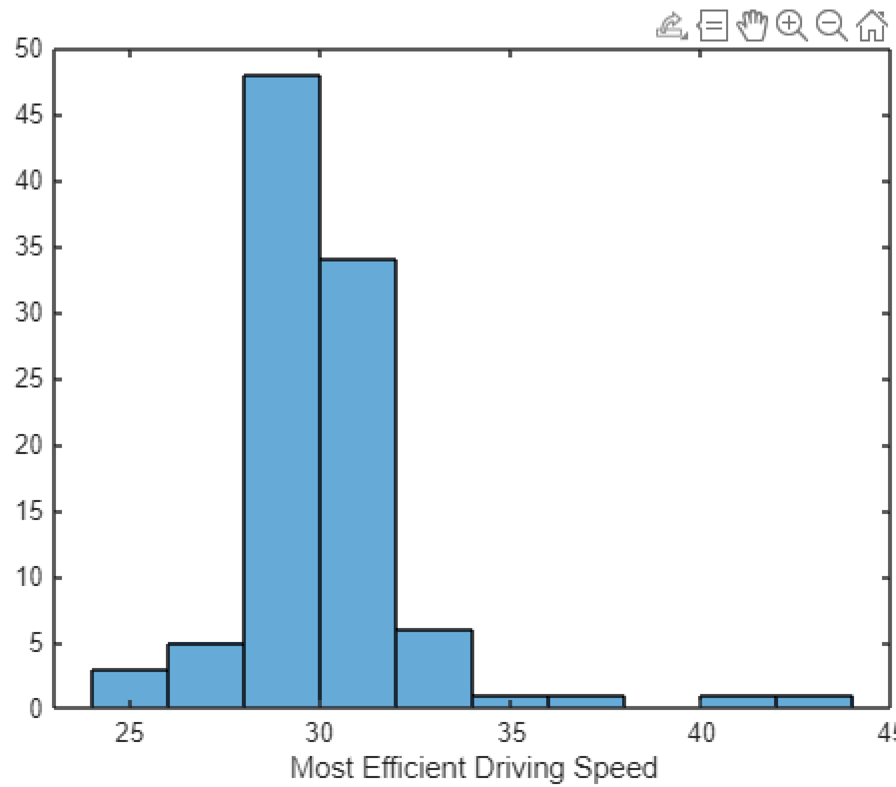
We should call Proprietary of Car Num.: "32" to propose him a Revision!

Uses of a fit

- **We process the data of 100 cars in seconds!!**
- Also very interesting:

✓ Most efficient Driving Speed at about 30 mph or a bit lower. (About 48,3 km/h)

✓ Ideal for city traffic.



We should call Proprietary of Car Num.: “32” to propose him a Revision!

Now is your turn

- Now follow the next steps indicated by the Prof for your Task on this topic.

- Choose a data set that you want to analyse and develop your own analisys.

- You have also examples at:

- https://es.mathworks.com/help/curvefit/getting-started-with-curve-fitting-toolbox.html?s_tid=CRUX_lftnav
- https://es.mathworks.com/matlabcentral/fileexchange/93435-regression-basics?s_tid=ma_spoc_edu_orcf