Too Much Time and Too Little Time Linked to Lower Subjective Well Being

If R^2 is low, then you can suspect mean model error.

Table 2
Regression Results of Study 2: The Influence of Discretionary Time on Subjective Well-Being

Variable	(1)	(2)
Hours of discretionary time	0.027*	.035
	(0.011)	(0.012)
Hours of discretionary time squared	−0.004***	−0.003***
	(0.001)	(0.001)
Male		−0.189***
		(0.029)
White		−0.051
		(0.035)
Age		0.006***
		(0.001)
Married		0.577***
		(0.031)
		0.076*
Children		(0.036)
		0.117***
		(0.032)
4-year college		0.030***
		(0.008)
Natural log-transformed earnings		−0.279**
Employed		(0.087)
Constant	7.134***	7.070***
	(0.034)	(0.037)
R^2	.003	.032

Note. (1) The coefficients reported above are the unstandardized coefficients. (2) All predictor variables, except Hours of Discretionary time and Hours of Discretionary time Squared, are mean-centered.
$^{+}p < .10$. $^{*}p < .05$. $^{**}p < .01$. $^{***}p < .001$.

Use the coefficients.

What does /. mean?

It means replace values.

Note that Wolfram Engine is stringent to *

Claim is made for coefficient to be robustly is zero.

In[∘]:=
```
? /.
```
Out[∘]=

Symbol

expr /. *rules* or ReplaceAll[*expr*, *rules*] applies a rule or

list of rules in an attempt to transform each subpart of an expression *expr*.

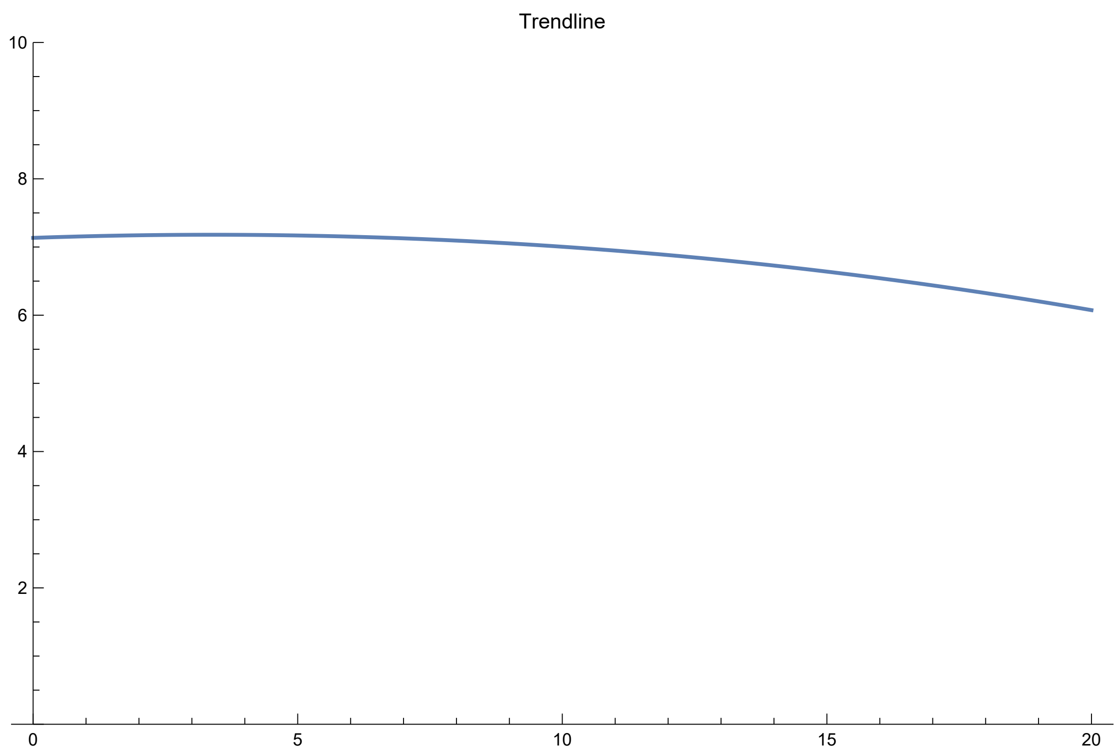
ReplaceAll[*rules*] represents an operator form of ReplaceAll that can be applied to an expression.

In[∘]:=
```
f = a * x + b * x^2 + c /. {a → .027, b → -.004, c → 7.134}
```
Out[∘]=

$7.134 + 0.027\,x - 0.004\,x^2$

Plot the trend line for the plot.

In[∘]:=
```
Plot[f, {x, 0, 20}, PlotRange → {0, 10}, PlotLabel → "Trendline", ImageSize → Large]
```
Out[∘]=



Replicate the table, how is the table replicated?

In[ ]:=
```
X = Table[i, {i, 1, 24, 1 / 2}]
```
Out[ ]=

$$\left\{1, \frac{3}{2}, 2, \frac{5}{2}, 3, \frac{7}{2}, 4, \frac{9}{2}, 5, \frac{11}{2}, 6, \frac{13}{2}, 7, \frac{15}{2}, 8, \frac{17}{2}, 9, \frac{19}{2}, 10, \frac{21}{2}, 11, \frac{23}{2}, 12, \frac{25}{2}, 13, \frac{27}{2}, 14, \frac{29}{2}, 15, \frac{31}{2}, 16, \frac{33}{2}, 17, \frac{35}{2}, 18, \frac{37}{2}, 19, \frac{39}{2}, 20, \frac{41}{2}, 21, \frac{43}{2}, 22, \frac{45}{2}, 23, \frac{47}{2}, 24\right\}$$

Nassim Taleb had the following.
He had the figure of the number of observations to hours.
This was obtained from another version of the paper by Adam Grant.

In[ ]:=
```
Y = {838, 828, 1181, 1136, 1306, 1167, 1222, 1113, 1073, 1006,
   1002, 953, 848, 779, 729, 688, 585, 546, 514, 447, 412, 350, 319, 246, 227,
   204, 161, 120, 84, 63, 41, 21, 35, 15, 21, 6, 8, 2, 9, 2, 1, 5, 1, 1, 1, 1, 1}
```
Out[ ]=
```
{838, 828, 1181, 1136, 1306, 1167, 1222, 1113, 1073, 1006, 1002,
 953, 848, 779, 729, 688, 585, 546, 514, 447, 412, 350, 319, 246, 227, 204,
 161, 120, 84, 63, 41, 21, 35, 15, 21, 6, 8, 2, 9, 2, 1, 5, 1, 1, 1, 1, 1}
```

In[ ]:=
```
ta = Table[Table[X[[i]], {Y[[i]]}], {i, 1, Length[X]}] // Flatten;
```

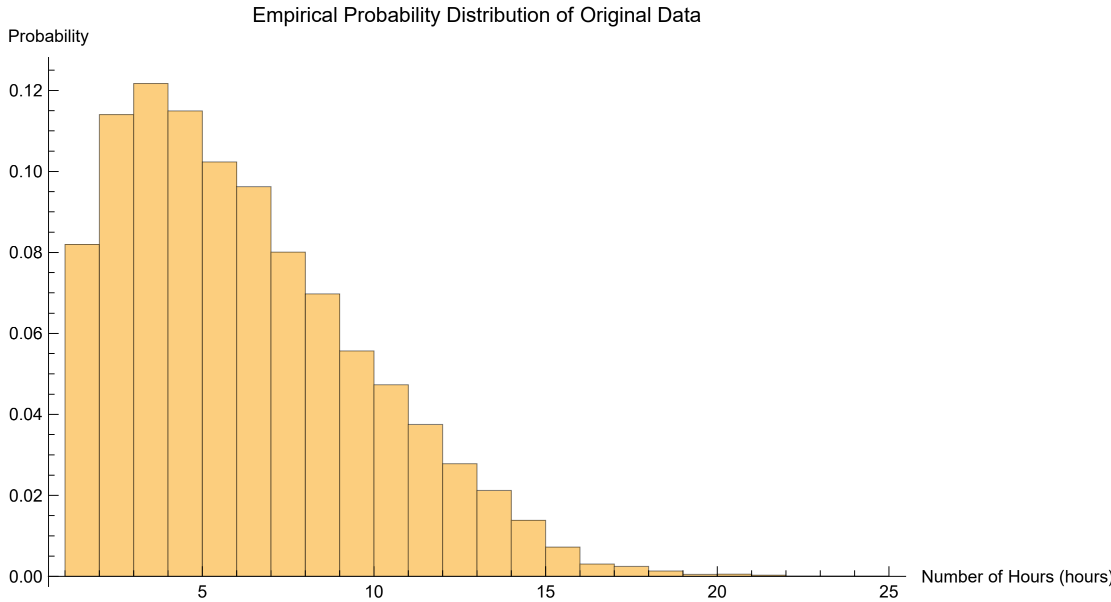
Makes the distribution into an empirical distribution.

In[ ]:=
```
dist = EmpiricalDistribution[ta]
```
Out[ ]=
```
DataDistribution[ Type: Empirical
                   Data points: 20318 ]
```

From a power law discussion, the square of a variable has half the tail exponent.

In[∘]:=
```
Histogram[ta, 30, Probability,
 PlotLabel → "Empirical Probability Distribution of Original Data",
 AxesLabel → {"Number of Hours (hours)", "Probability"}, ImageSize → Large]
```

Out[∘]=

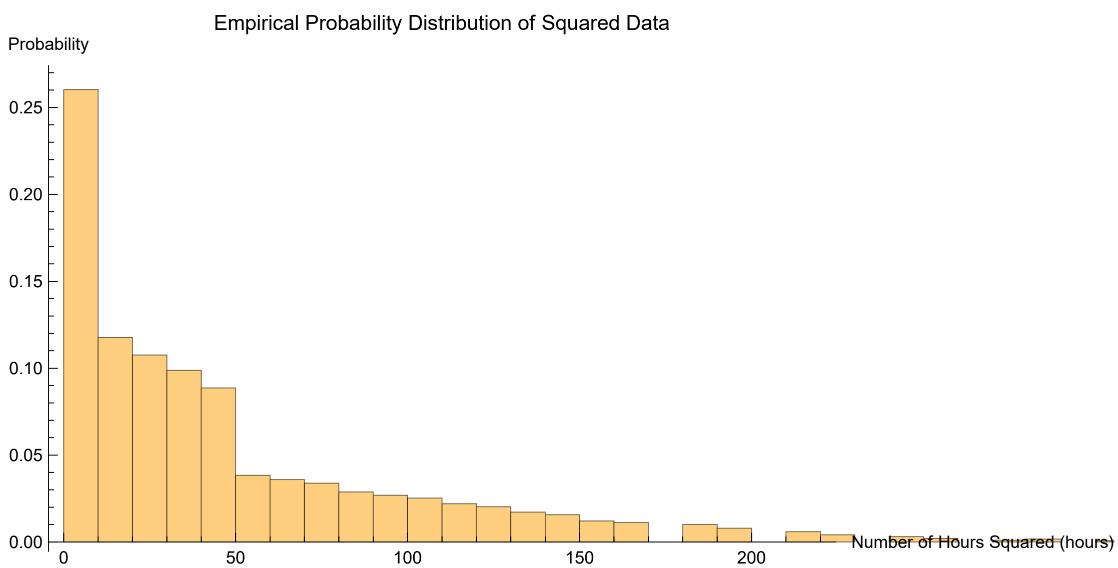


We take the square of the number of hours given the claims of a quadratic fit. The squared variable has half the tail exponent.

In[∘]:=
```
tasquare = Table[Table[X[[i]]^2, {Y[[i]]}], {i, 1, Length[X]}] // Flatten;
```

In[∘]:=
```
Histogram[tasquare, 50, Probability,
 PlotLabel → "Empirical Probability Distribution of Squared Data",
 AxesLabel → {"Number of Hours Squared (hours)", "Probability"}, ImageSize → Large]
```

Out[∘]=



This is the key step in replicating the paper. Take the smooth kernel distribution. It makes the distribution of the bins smoother.

In[ ]:=
```
dist2 = SmoothKernelDistribution[ta];
```

Apply N to all these values which makes them real and not fractions.

In[ ]:=
```
{Mean[dist2], Kurtosis[dist2], Skewness[dist2]} // N
```

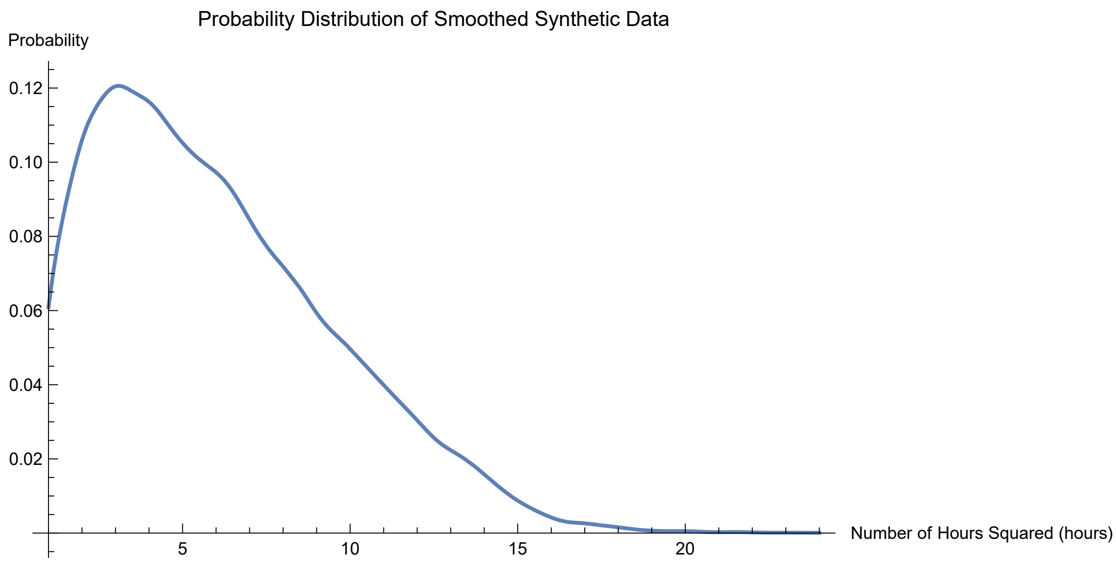
Out[ ]=
```
{6.02567, 3.17826, 0.750064}
```

In[ ]:=
```
{Skewness[ta], Skewness[dist2]} // N
```

Out[ ]=
```
{0.767495, 0.750064}
```

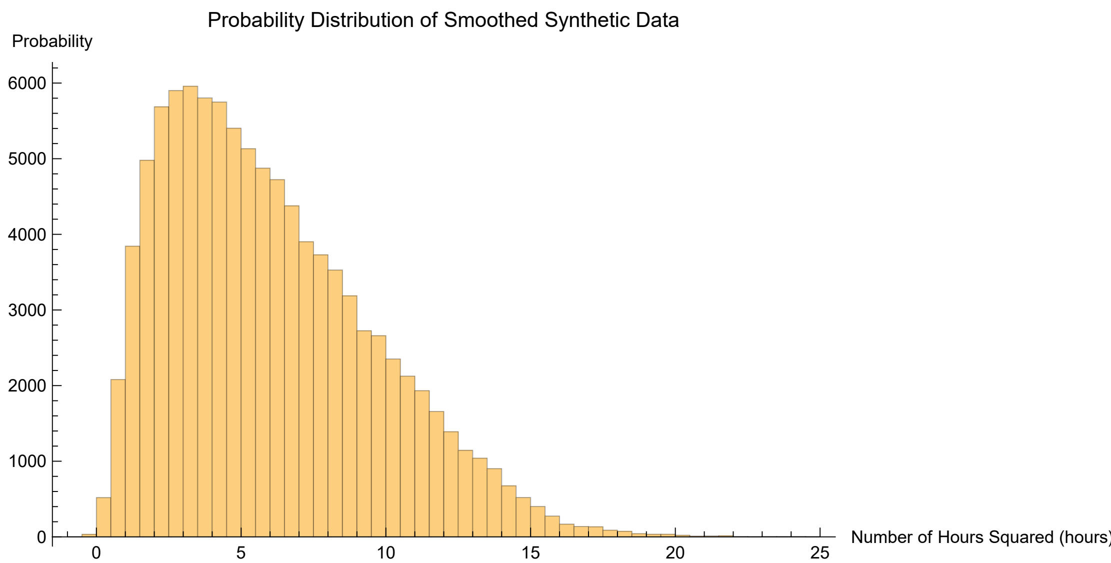
In[ ]:=
```
Plot[PDF[dist2, x], {x, 1, 24},
 PlotLabel → "Probability Distribution of Smoothed Synthetic Data",
 AxesLabel → {"Number of Hours Squared (hours)", "Probability"}, ImageSize → Large]
```

Out[ ]=



In[ ]:=
```
Histogram[RandomVariate[dist2, 10^5], 50,
 PlotLabel → "Probability Distribution of Smoothed Synthetic Data",
 AxesLabel → {"Number of Hours Squared (hours)", "Probability"}, ImageSize → Large]
```

Out[ ]=



Testing with No Noise

In[ ]:=
```
f = a * x + b * x^2 + c /. {a → .044, b → -.003, c → 3.180};
f = a * x + b * x^2 + c /. {a → .027, b → -.004, c → 7.134};
```

Apply the regression line to the original function.

In[ ]:=
```
f1 = f /. x → ta;
```

In[ ]:=
```
reg = Transpose[{ta, f1}];
```

In[ ]:=
```
lm = LinearModelFit[reg, {1, x, x^2}, x];
```

Recall the table above was to get this linear regression fit of $R^2 = 1$ given the coefficients.

In[ ]:=
```
lm
```

Out[ ]=

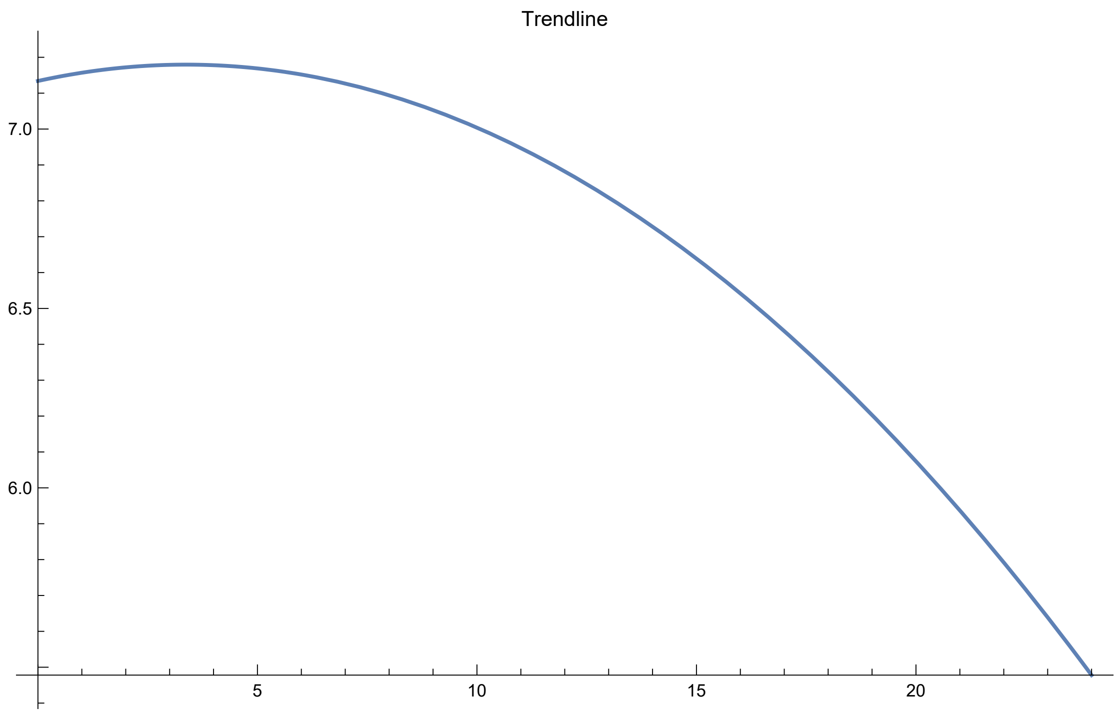
```
FittedModel[ 7.134 + 0.027 x - 0.004 x^2 ]
```

In[ ]:=
```
lm["RSquared"]
```

Out[ ]=

```
1.
```

In[∘]:=
```
Plot[lm[x], {x, 0, 24}, PlotLabel → "Trendline", ImageSize → Large]
```
Out[∘]=



Obtain the length of f1 based on the length of samples in ta.

In[∘]:=
```
f1 // Length
```
Out[∘]=
```
20 318
```

Apply some Gaussian noise to the function. Then perform the transpose to make it suitable for regression again.

In[∘]:=
```
reg1[sigma_] :=
 Transpose[{ta, f1 + RandomVariate[NormalDistribution[0, sigma], Length[f1]]}]
```

Then, apply a linear model fit with the model. Check the R^2 as usual, run about the model 10^4 times so as to get a distribution of possible R^2 values. This is the Monte Carlo simulation for possible R^2 and eats up a lot of time.

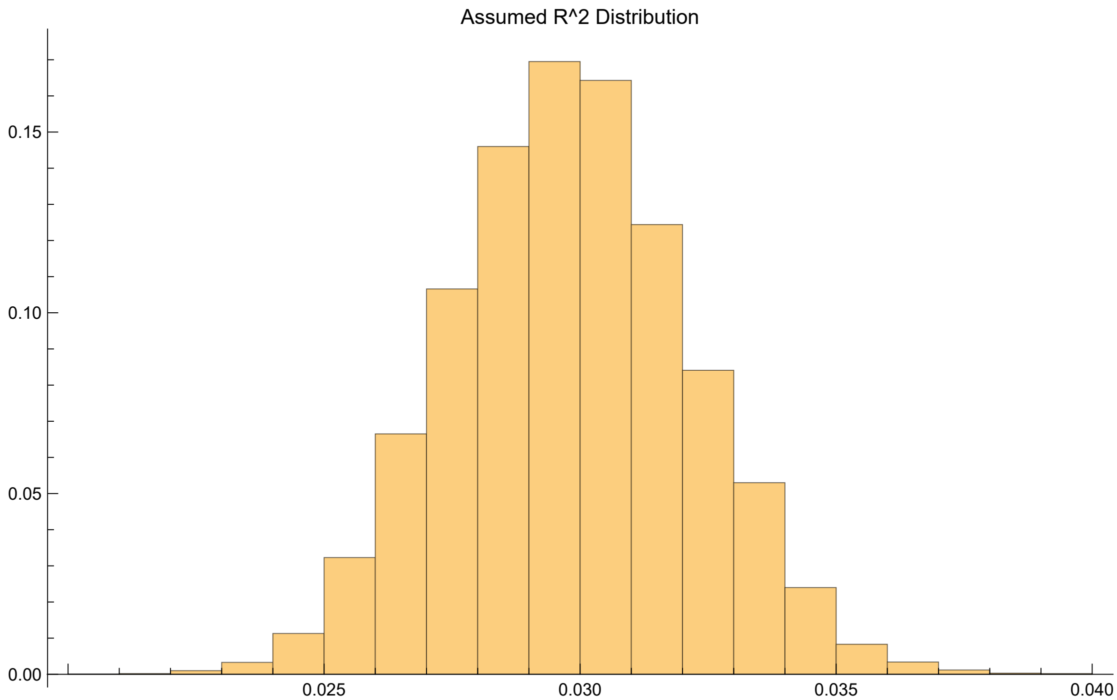
In[∘]:=
```
res = Table[lm = LinearModelFit[reg1[.75], {1, x, x^2}, x]; lm["RSquared"], {z, 1, 10^4}];
```

Get the assumed R^2 distribution and parameters. It is almost Gaussian.

In[∘]:=
```
{Mean[res], StandardDeviation[res], Skewness[res], Kurtosis[res]}
```
Out[∘]=
```
{0.0298131, 0.00233106, 0.0892445, 3.02466}
```

In[ ]:=
```
pl0 = Histogram[res, Automatic, "Probability",
  PlotLabel → "Assumed R^2 Distribution", ImageSize → Large]
```

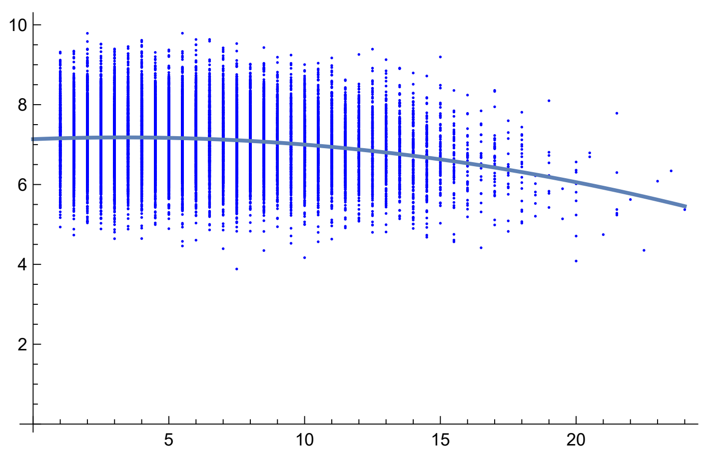
Out[ ]=



Replicate the graph that slopes down because of the lack of observations. This is why their slope is quadratic.

In[ ]:=
```
Show[ListPlot[reg1[.75], PlotStyle → Blue], Plot[lm[x], {x, 0, 24}]]
```

Out[ ]=



Replicating Their Quadratic Term

Apply the same technique, tabulate the values of the second coefficient, run the Monte Carlo simulation 10^4 times.

In[◦]:=
```
ta2 = Table[lm2 = LinearModelFit[reg1[.75], {1, x, x^2}, x]; lm2[x][[3]][[1]], {z, 1, 10^4}]
```

Out[◦]=
```
{-0.00395854, -0.00323263, -0.0040112, -0.00360207, -0.00449473, -0.00384216, -0.00375709,
 -0.00427511, -0.00375041, -0.00373552, … 9980 …, -0.00364796, -0.00378253, -0.00388888,
 -0.00452171, -0.0035481, -0.00418265, -0.00413494, -0.00412322, -0.00438103, -0.00401041}
```

Full expression not available (original memory size: 240.1 kB)

Get the distribution parameters of the quadratic term.

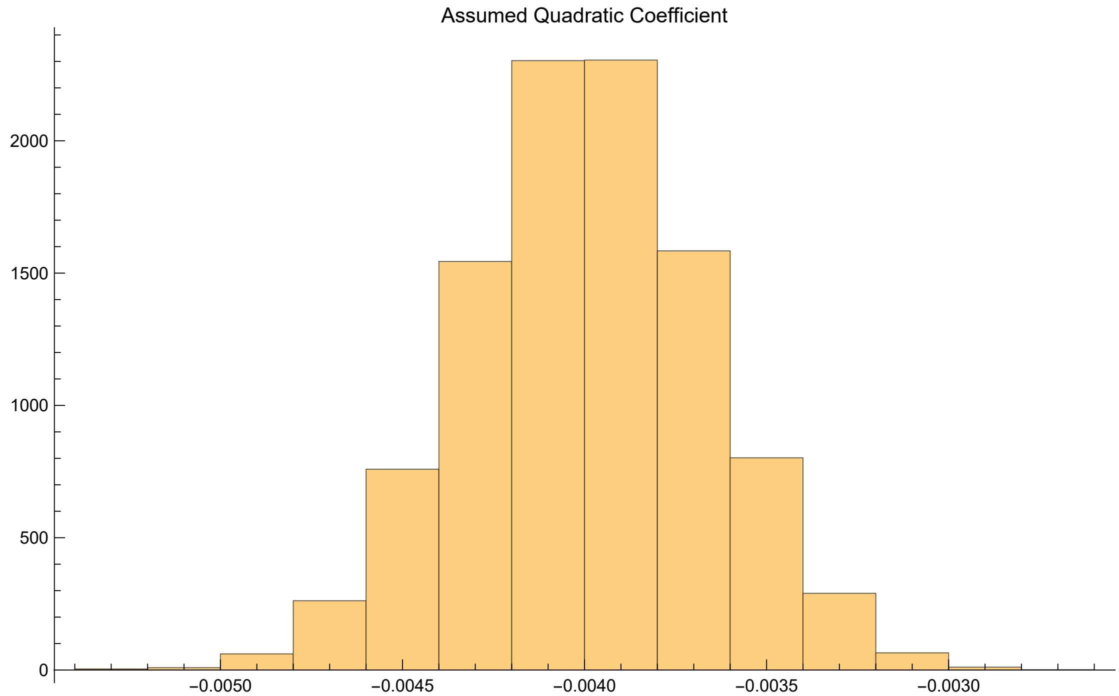
In[◦]:=
```
{Mean[ta2], StandardDeviation[ta2], Skewness[ta2], Kurtosis[ta2]}
```

Out[◦]=
```
{-0.00399399, 0.000330347, -0.00206064, 3.00813}
```

Plot the histogram of the assumed quadratic coefficient.

In[◦]:=
```
pl1 = Histogram[ta2, PlotLabel → "Assumed Quadratic Coefficient", ImageSize → Large]
```

Out[◦]=

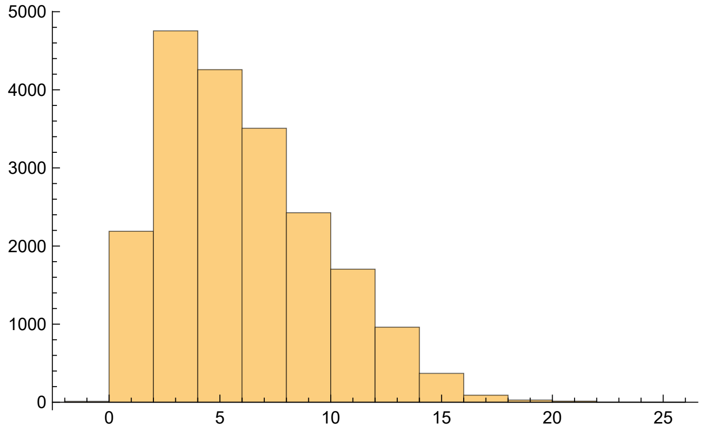


Monte-Carlo of both the distribution and the quadratic term.

```
ta := RandomVariate[dist2, Length[f1]]
```

In[∘]:= Histogram[ta]

Out[∘]=



In[∘]:= ta2 = Table[lm2 = LinearModelFit[reg1[.75], {1, x, x^2}, x]; lm2[x][[3]][[1]], {z, 1, 10^4}]

Out[∘]=

{0.0000542908, -0.000133462, 0.000665807, -0.000122389, -0.000680994, -0.000398676, 0.000109784, 0.000197352, 0.000044398, -0.000343765, … 9980 …, 1.37934×10^{-6}, -0.000639528, 0.000405537, -0.000397033, 0.000220323, -0.00014021, -0.000247262, 0.000207362, 0.000217003, 0.000450744}

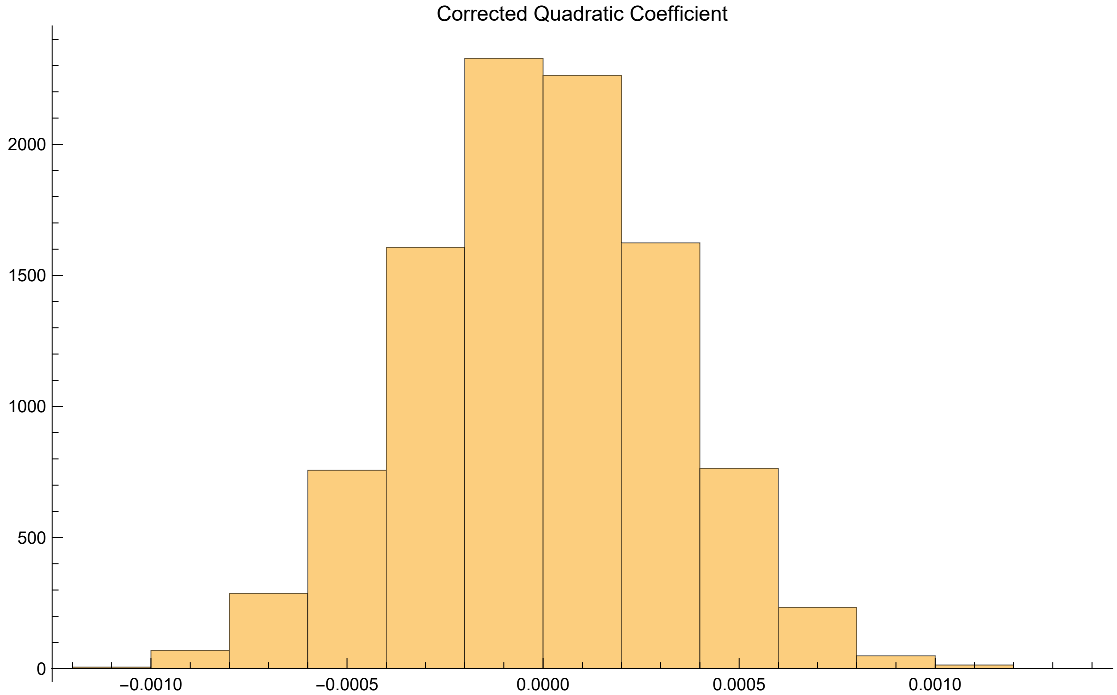
Full expression not available (original memory size: 240.1 kB)

In[∘]:= {Mean[ta2], StandardDeviation[ta2], Kurtosis[ta2]}

Out[∘]=

$\{-4.14713 \times 10^{-6}, 0.000325102, 2.98905\}$

In[◦]:=
```
pl2 = Histogram[ta2, PlotLabel → "Corrected Quadratic Coefficient", ImageSize → Large]
```
Out[◦]=



Examining R^2

Do we need to define *n*?

In[◦]:=
```
Mean[FRatioDistribution[n, n]]
```
Out[◦]=

$$\begin{cases} \frac{n}{-2+n} & n > 2 \\ \text{Indeterminate} & \text{True} \end{cases}$$

Why did this transformation fail?

In[◦]:=
```
EstimatedDistribution[ta2, FRatioDistribution[n, n]];
```

••• EstimatedDistribution: One or more data points are not in support of the process or distribution FRatioDistribution[n, n].

In[◦]:=
```
TransformedDistribution[x / y, {x ≈ ChiSquareDistribution[n], y ≈ ChiSquareDistribution[n]}]
```
Out[◦]=
```
FRatioDistribution[n, n]
```

In[◦]:=
```
LinearModelFit[reg1[.75], {1, x, x^2}, x]["RSquared"];
```

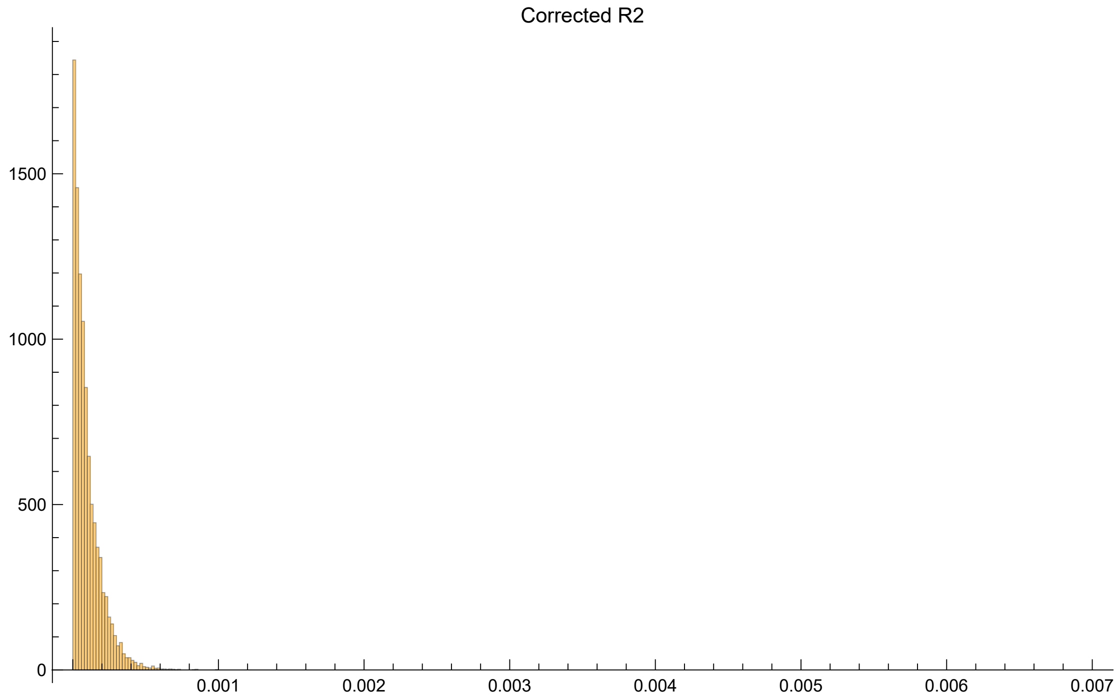
Verify the ta3 is legitimate.

In[◦]:=
```
ta3 = Table[lm2 = LinearModelFit[reg1[.88], {1, x, x^2}, x]["RSquared"], {z, 1, 10^4}];
```

In[◦]:=
```
{Mean[ta3], StandardDeviation[ta3], Kurtosis[ta3]}
```
Out[◦]=
```
{0.0000979671, 0.000096557, 8.98911}
```

Plot the last histogram which shows that it is all in the left.

```
In[*]:= pl3 = Histogram[ta3, PlotLabel → "Corrected R2",
          PlotRange → {{0, 0.007}, Automatic}, ImageSize → Large]
Out[*]=
```



Plot the comparisons between the various grids.

In[]:=

```
GraphicsGrid[{{pl3, pl2}, {pl0, pl1}}]
```

Out[]=