In[115]:=

```
(*Fitting-M2*)
(*Jared Frazier*)
(*09-20-20*)
(*Description:
First derive the line of best fit for a linear equation ŷ = β0 + β1X, that is
derive the coffecients β0 and β1. Once these two coefficients are derived, use the following obser
from the Cathode Ray experiment to derive the variables associated with the equation modeling stop
voltage and frequency of light:

[1] The photoelectric effect is the emission of electrons when electromagnetic radiation (e.g., li
hits a material.
[2] The expected emission of electrons was essentially continuous, meaning as the intensity (aver
one period of the wave) of light increased, it was expected that the kinetic energy of emitted ele
proportional. This expectation was refuted with experimental results.
[3] Incident light (radiation) had to have a frequency exceeding a certain threshold before elect
[4] The stopping potential V is the voltage at which no electrons that are emitted from the metal
to incident radiation are able to reach another plate (due to electron-electron repulsion). The s
was measured for different frequencies ν of the incident beam.
[5] Lastly, it is observed from Einstein and the Cathode Ray Experiment that the kinetic energy o
must be Ek = hν - ϕ = hν - hν0 where the units of Ek are the electron charge e and the voltage V.
voltage, the stopping voltage becomes a simple linear equation whose parameters can be determined
sum of squared errors w.r.t β0 and β1 such that V(ν) = (h/e)ν - (h/e)ν0 = β1ν + β0.

Refs:
https://aklectures.com/lecture/photoelectric-effect-compton-effect-wave-particle-duality/stopping
https://en.wikipedia.org/wiki/Photoelectric_effect*)

Clear["Global`"]

(*Import Notation package and define symbols*)
<<Notation`
Clear["Global`*"];
Notation[ β̂0 ⟺ betaNaught ]
Notation[ β̂1 ⟺ betaOne ]
Notation[ Ŷk ⟺ yHat ]
Notation[ Yk ⟺ yData ]
Notation[ Xk ⟺ xData ]
Notation[ Yavg ⟺ yBar ]
Notation[ Xavg ⟺ xBar ]

(*Deriving the slope (β̂0) and intercept(β̂1)*)
(*General form of linear regression equation, betaNaught and betaOne are unknown constants curren
yHat = betaNaught + betaOne * xData;

(*Define the general form for the sum of square errors*)
```

```
sse = Sum[(yData - yHat)^2, {k, 1, N}];
sseSummand = (yData - yHat)^2;

(*Display logic*)
Print["Part 1 - Derivation of slope 'a' (β̂1) and intercept 'b' (β̂0):"]
Print["(1) Linear relationship between stopping voltage and frequency of incident light
\tV(ν) = aν + b can be modeled with the same equation Ŷk = ", yHat, " where Ŷk is the
\tpredicted response variable (V(ν)) and Xk is the kth explanatory variable (ν)."]
Print["(2) The sum of squared errors, also known as the mean squared error, is
\tthe sum of the difference between all response values in the data set (Yk) and
\tpredicted response values (Ŷk) and is modeled by the equation SSE = Sum[(Yk-Ŷk)^2, {k, 1, N}]
\tfor the k^th term in a data set with N terms."]
Print["(3) Substituting the definition of Ŷk into SSE yields an equation which,
\tonce differentiated with respect to β̂1 and β̂0 and set equal to 0, may be solved
\tsimultaneously to yield the minimum values of β̂1 and β̂0 to minimize the 'distance'
\tbetween the prediction 'line' and values of Yk: SSE = Sum[(Yk-(β̂0 + β̂1Xk))^2, {k, 1, N}]"]

(*Differentiate with respect to β̂0 and β̂1, derivative of summation is sum of derivatives*)
summandDerivBetaNaught = D[sseSummand, {betaNaught, 1}];
summandDerivBetaOne = D[sseSummand, {betaOne, 1}];
Print["(4) ∂SSE/∂β̂0 Sum[(Yk-(β̂0 + β̂1Xk))^2, {k, 1, N}] = Sum[", summandDerivBetaNaught,
"\tand ∂SSE/∂β̂1 Sum[(Yk-(β̂0 + β̂1Xk))^2, {k, 1, N}] = Sum[", summandDerivBetaOne]

(*Solve partial derivative with respect to betaNaught == 0*)
soln1 = Solve[{summandDerivBetaNaught==0}, {betaNaught}];
storeSoln1 = betaNaught/.soln1[[1]];

(*Summation of a constant*)
summationRule = Sum[betaNaught, {k, 1, N}];

(*Display solution for B̂0 in terms of B̂1*)
Print["(5) The solution for ", summationRule, " = ", storeSoln1, " for N values in the
\tdata is equivalent to B̂0 = Ȳ - B̂1X̄ where Ȳ and X̄ are the average of the X and
\tY values in the data."]

(*Equivalent solution for B̂0*)
betaNaughtAvg = yBar - betaOne*xBar;

(*Solve for B̂1 by substituting B̂0 in terms of B̂1 into ∂SSE/∂β̂1*)
betaOneDerivOneVarSummand = xData * (-betaNaughtAvg - betaOne*xData + yData);

(*Display new summation*)
Print["(6) Substituting B̂0 in terms of B̂1 into ∂SSE/∂β̂1, Sum[" betaOneDerivOneVarSummand,
```

```
" which simplifies to ∑_{k=1}^{N}", FullSimplify[betaOneDerivOneVarSummand] ]
(*Self expand*)
Print["(7) Expanding the summation using summation rules,
\t∑_{k=1}^{N} X_k (Y_k-Y_avg) - B̂_1 ∑_{k=1}^{N} X_k (X_k-X_avg)=0. Mathematica cannot recognize that there are
\ttwo variables Y_k and X_k in the simplified equation. It treats all of them as
\tconstants and the syntax x[[k]] or y[[k]] cannot be used either to indicate
\tthe k^th variable of a predefined list (requires actual data)."]
Print["(8) Therefore, B̂_1 = (∑_{k=1}^{N} X_k (Y_k-Y_avg))/(∑_{k=1}^{N} X_k (X_k-X_avg)) and B̂_0 = ", betaNaughtAvg]
Print[]

(*Function to convert wavelength to frequency*)
toFrequency[wave_] := (3*10^8)/(wave*10^(-9));

(*x and y values in photoelectric data set*)
xList = {toFrequency[365], toFrequency[405], toFrequency[436],
         toFrequency[546], toFrequency[577]};
yList = {1.855, 1.366, 1.143, 0.618, 0.498};
numEle = 5;

(*Use derived equations to calc linear equation *)
Print["Part 2 - Calculate slope and intercept of the equation for stopping voltage as a function 

(*Calculate averages of x and y lists*)
xAvg = Sum[xList[[i]], {i,1,numEle}] / numEle;
yAvg = Sum[yList[[i]], {i,1,numEle}] / numEle;
calcBetaOne = Sum[xList[[i]]*(yList[[i]]-yAvg), {i, 1, numEle}] / Sum[xList[[i]]*(xList[[i]]-xAvg), {i, 1, numEle}];
calcBetaNaught = yAvg-calcBetaOne*xAvg;

(*Line of best fit*)
linEq[ν_] := calcBetaNaught + calcBetaOne*ν;
Print["(1) My line of best fit is V̂(ν) = ", linEq[ν], " and Planck's constant
\th is ", calcBetaOne, " eV*s which is close to the actual value
\t4.1357 × 10^-15 eV*s. Therefore, %error is ", 100*(Abs[calcBetaOne-4.13157*10^(-15)] / 4.13157*10^(-15)), "%"]

(*Standard Deviation and Correlation coefficient r*)
sx = (Sum[(xList[[i]] - xAvg)^2, {i,1,numEle}] / (numEle-1))^(1/2);
sy = (Sum[(yList[[i]] - yAvg)^2, {i,1,numEle}] / (numEle-1))^(1/2);
```

```
r = 1/(numEle-1)*Sum[(xList[[i]] - xAvg)/sx*(yList[[i]]-yAvg)/sy, {i,1,numEle}];

(*Coefficient of determination*)
r2 = r^2;
Print["(2) The coefficient of determination R^2 = ", r2, " from R = 1/(N-1)*Σ_{k=1}^{N}((X_k-X̄)/S_x)((Y_k-Ȳ)/S_y)"]

(*How well the line predicts values*)
eqError = Sqrt[Sum[(yList[[i]]-linEq[xList[[i]]])^2, {i, 1, numEle}]/(numEle-2)];
Print["(3) The error associated with predicted value V̂(ν) is modeled by the equation
\tS = Sqrt[(Σ_{k=1}^{N}(V(ν)_k - V̂(ν)_k)^2)/(N - 2)] = ±", eqError, " Volts" ]

(*Plot data with my line of best fit*)
Print[]
xy2dList = {{toFrequency[365], 1.855},{ toFrequency[405], 1.366},{toFrequency[436],1.143 },
         {toFrequency[546], 0.618}, {toFrequency[577], 0.498}};
Print["Part 3 - Plot the data with my line of best fit and mathematicas line of best fit:"]
Print["(1) My line of best fit is V̂(ν) = ", linEq[ν], " and the plot is shown below:"]
Show[{Plot[linEq[x], {x,toFrequency[577],toFrequency[365]}, PlotLabel→"My Line of Best Fit for Pl
AxesLabel→{"Frequency ν (s^-1)", "Stopping Voltage V"}, ImageSize→700], ListPlot[xy2dList]}]

(*Mathematica's line of best fit formula*)
mathematicaLinEq = LinearModelFit[xy2dList, ν, ν];
functionMathematicaLinEq[ν_] = Normal[mathematicaLinEq];
Print["(2) The equation mathematica finds for the line of best fit is
\tV(ν) = ", Normal[mathematicaLinEq], " which directly matches my line of best of fit, and the pl
Show[{Plot[functionMathematicaLinEq[x], {x,toFrequency[577],toFrequency[365]}, PlotLabel→"Mathem
AxesLabel→{"Frequency ν (s^-1)", "Stopping Voltage V"}, ImageSize→700], ListPlot[xy2dList]}]
```

Part 1 – Derivation of slope 'a' ($\hat{\beta}_1$) and intercept 'b' ($\hat{\beta}_0$):

(1) Linear relationship between stopping voltage and frequency of incident light $V(\nu) = a\nu + b$ can be modeled with the same equation $\hat{Y}_k = \hat{\beta}_0 + \hat{\beta}_1 X_k$ where $\hat{Y}_k$ is the predicted response variable $(V(\nu))$ and X_k is the kth explanatory variable (ν).

(2) The sum of squared errors, also known as the mean squared error, is the sum of the difference between all response values in the data set (Y_k) and predicted response values $(\hat{Y}_k)$ and is modeled by the equation $SSE = \sum_{k=1}^{N}(Y_k - \hat{Y}_k)^2$ for the k^{th} term in a data set with N terms.

(3) Substituting the definition of $\hat{Y}_k$ into SSE yields an equation which, once differentiated with respect to $\hat{\beta}_1$ and $\hat{\beta}_0$ and set equal to 0, may be solved simultaneously to yield the minimum values of $\hat{\beta}_1$ and $\hat{\beta}_0$ to minimize the 'distance' between the prediction 'line' and values of Y_k: $SSE = \sum_{k=1}^{N}(Y_k - (\hat{\beta}_0 + \hat{\beta}_1 X_k))^2$

(4) $\frac{\partial SSE}{\partial \hat{\beta}_0}\sum_{k=1}^{N}(Y_k - (\hat{\beta}_0 + \hat{\beta}_1 X_k))^2 = \sum_{k=1}^{N} -2\left(-\hat{\beta}_0 - \hat{\beta}_1 X_k + Y_k\right)$

and $\frac{\partial SSE}{\partial \hat{\beta}_1}\sum_{k=1}^{N}(Y_k - (\hat{\beta}_0 + \hat{\beta}_1 X_k))^2 = \sum_{k=1}^{N} -2 X_k\left(-\hat{\beta}_0 - \hat{\beta}_1 X_k + Y_k\right)$

(5) The solution for $\hat{\beta}_0 N = -\hat{\beta}_1 X_k + Y_k$ for N values in the data is equivalent to $\hat{B}_0 = \overline{Y} - \hat{B}_1\overline{X}$ where $\overline{Y}$ and $\overline{X}$ are the average of the X and Y values in the data.

(6) Substituting $\hat{B}_0$ in terms of $\hat{B}_1$ into $\frac{\partial SSE}{\partial \hat{\beta}_1}$, $\sum_{k=1}^{N} X_k\left(\hat{\beta}_1 X_{avg} - \hat{\beta}_1 X_k - Y_{avg} + Y_k\right)$

which simplifies to $\sum_{k=1}^{N} X_k\left(\hat{\beta}_1 (X_{avg} - X_k) - Y_{avg} + Y_k\right)$

(7) Expanding the summation using summation rules, $\sum_{k=1}^{N} X_k (Y_k - Y_{avg}) - \hat{B}_1 \sum_{k=1}^{N} X_k (X_k - X_{avg}) = 0$. Mathematica cannot recognize that there are two variables Y_k and X_k in the simplified equation. It treats all of them as constants and the syntax x[[k]] or y[[k]] cannot be used either to indicate the k^{th} variable of a predefined list (requires actual data).

(8) Therefore, $\hat{B}_1 = \frac{\sum_{k=1}^{N} X_k (Y_k - Y_{avg})}{\sum_{k=1}^{N} X_k (X_k - X_{avg})}$ and $\hat{B}_0 = -\hat{\beta}_1 X_{avg} + Y_{avg}$

Part 2 - Calculate slope and intercept of the equation for stopping voltage as a function of frequency $V(\nu)$:

(1) My line of best fit is $\hat{V}(\nu) = -1.77859 + 4.32906\times10^{-15}\,\nu$ and Planck's constant h is 4.32906×10^{-15} eV*s which is close to the actual value 4.1357×10^{-15} eV*s. Therefore, %error is 4.77992%

(2) The coefficient of determination $R^2 = 0.988843$ from $R = \frac{1}{N-1} * \sum_{k=1}^{N}\left(\frac{X_k - \overline{X}}{S_x}\right)\left(\frac{Y_k - \overline{Y}}{S_y}\right)$

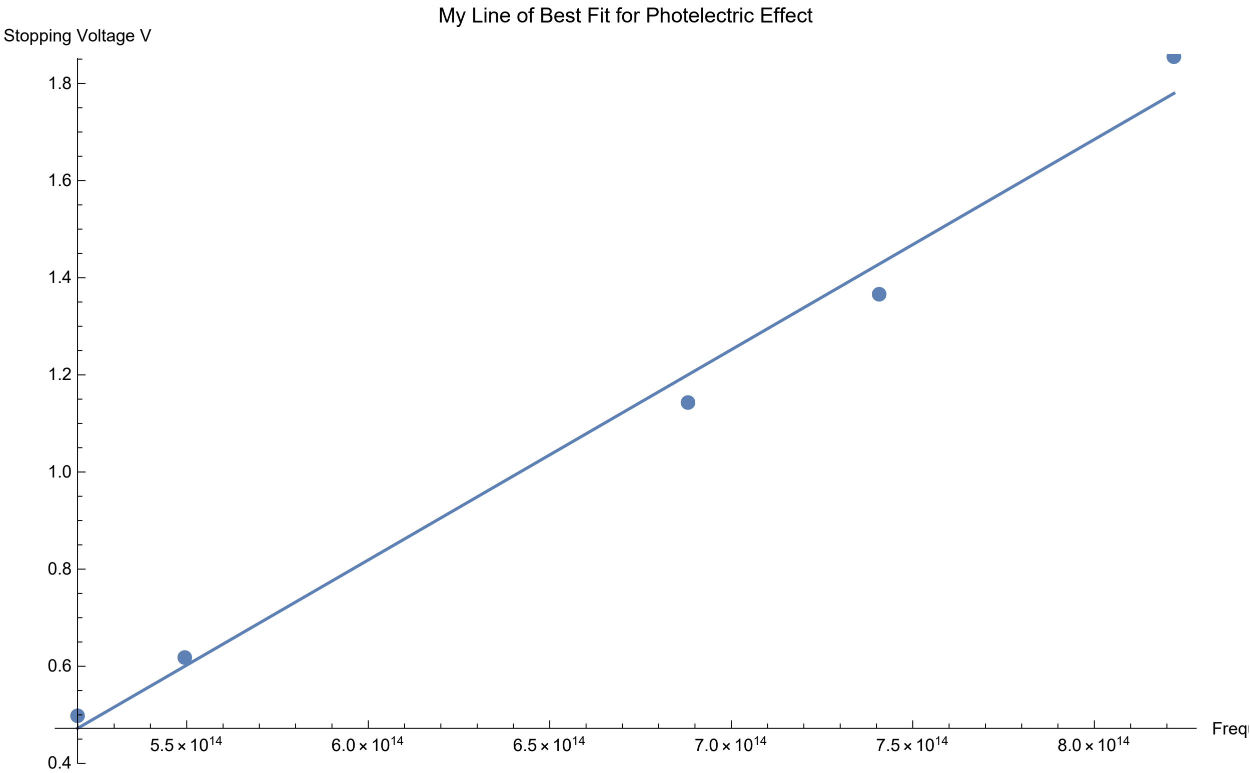
(3) The error associated with predicted value $\hat{V}(\nu)$ is modeled by the equation

$$S = \sqrt{\frac{\sum_{k=1}^{N}\left(V(\nu)_k - \hat{V}(\nu)_k\right)^2}{N-2}} = \pm 0.0678328 \text{ Volts}$$

Part 3 - Plot the data with my line of best fit and mathematicas line of best fit:

(1) My line of best fit is $\hat{V}(\nu) = -1.77859 + 4.32906\times10^{-15}\,\nu$ and the plot is shown below:

Out[167]=



(2) The equation mathematica finds for the line of best fit is

$V(\nu) = -1.77859 + 4.32906 \times 10^{-15}\,\nu$

which directly matches my line of best of fit, and the plot is shown below:

Out[171]=