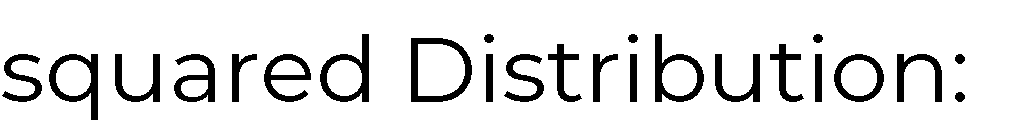
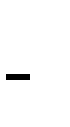
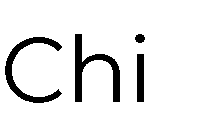
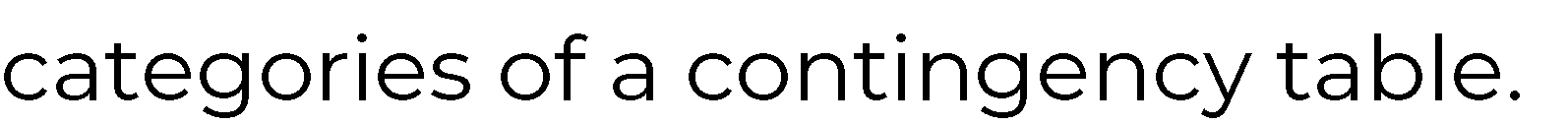
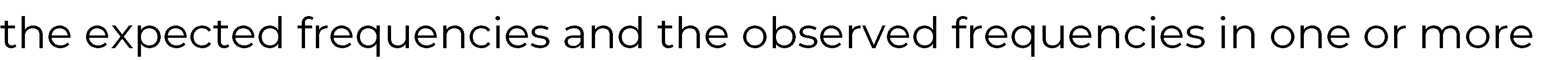
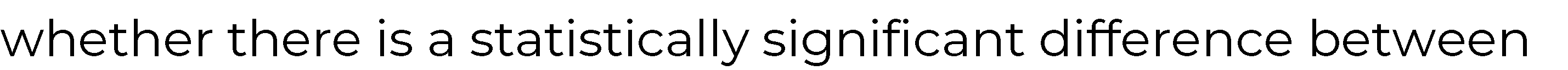
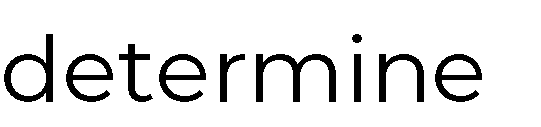
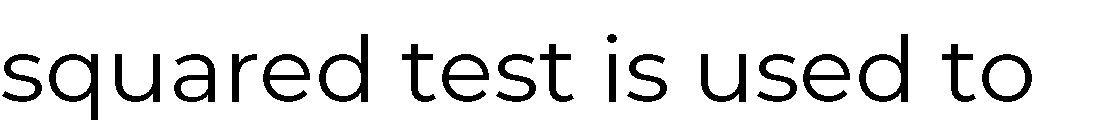
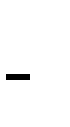
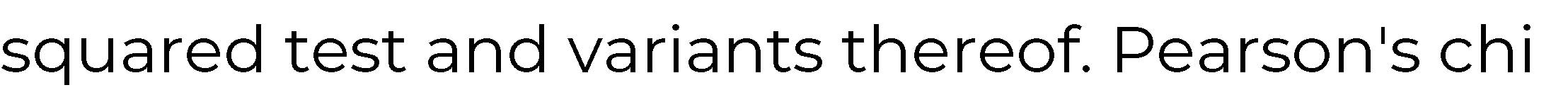
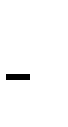
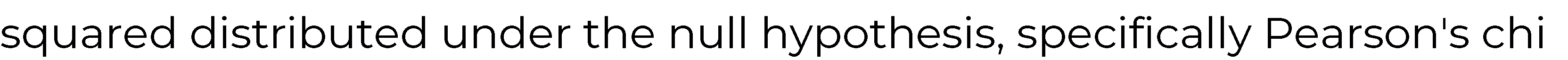
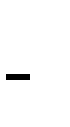
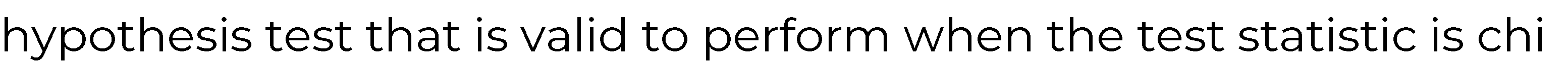
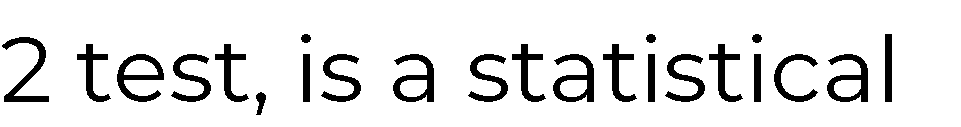
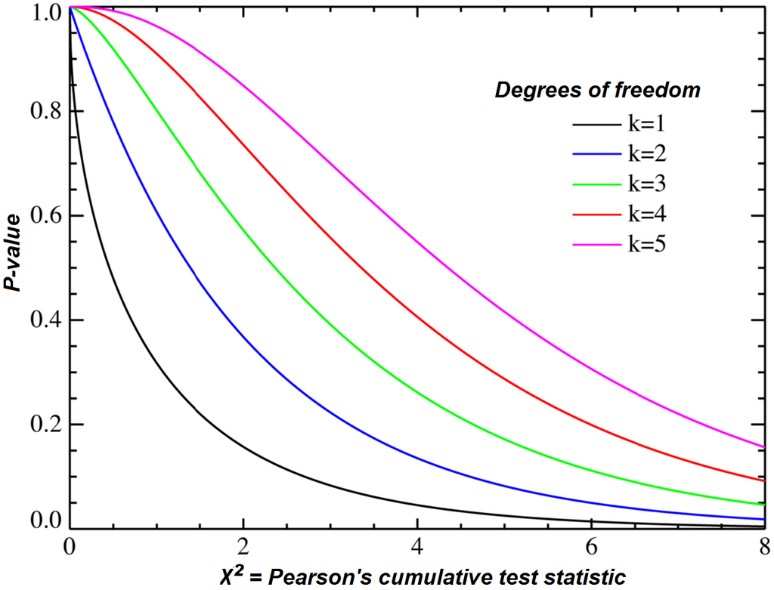
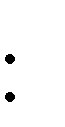
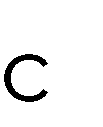
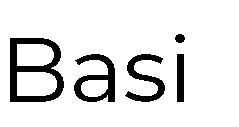
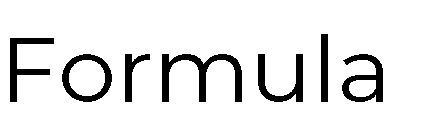
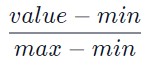
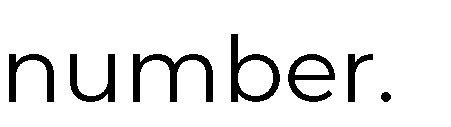
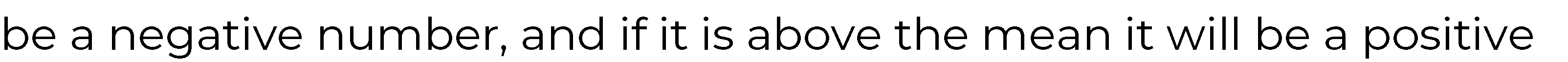
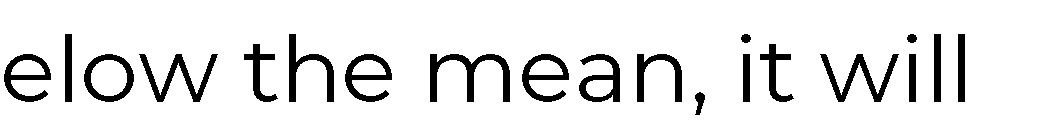
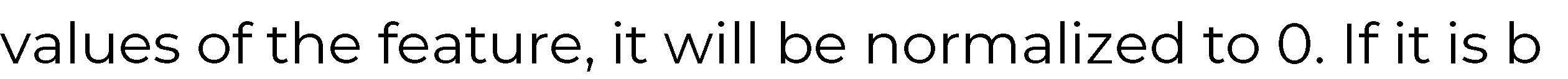
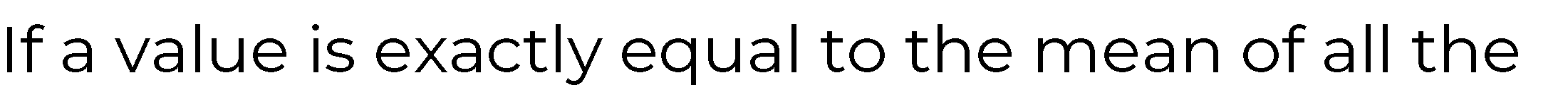
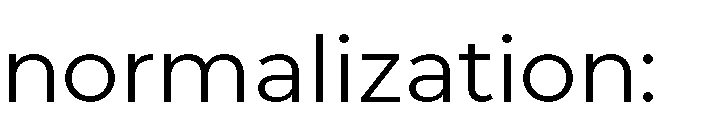
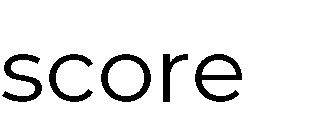
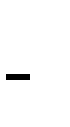
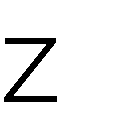
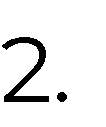
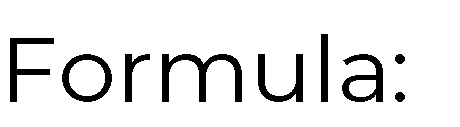
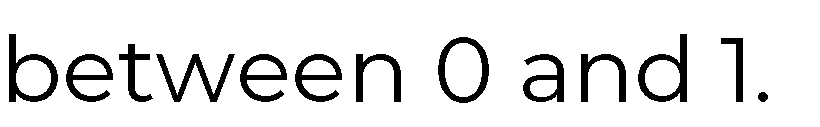
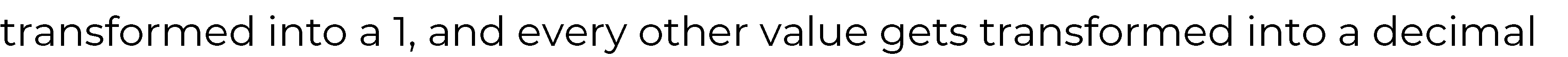
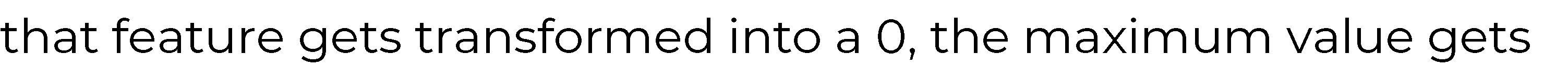
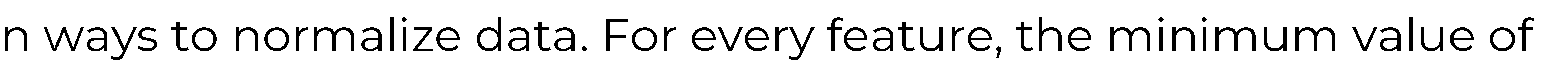
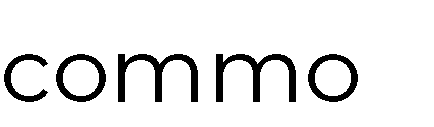
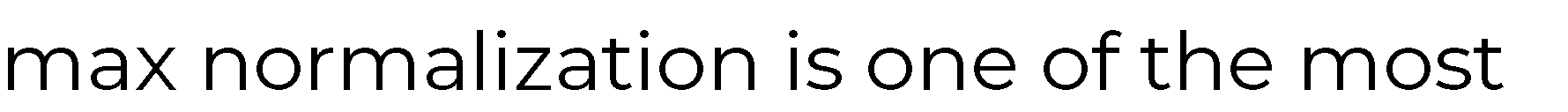
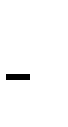
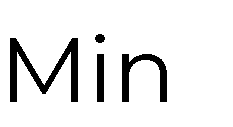
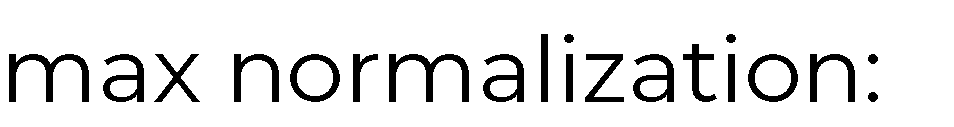
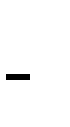
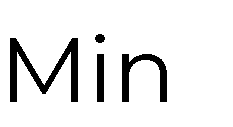
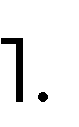
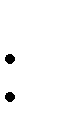
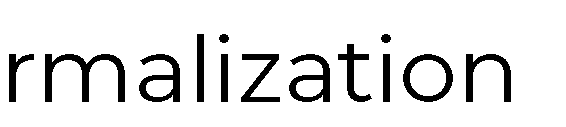
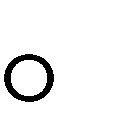
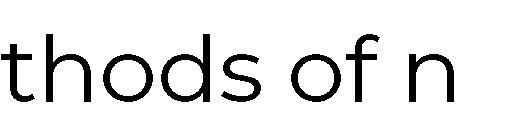
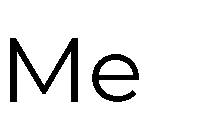
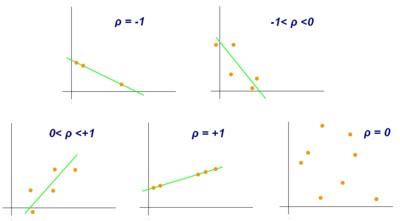
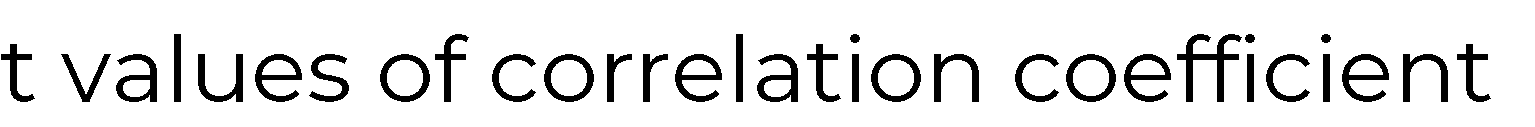
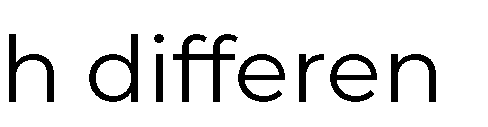
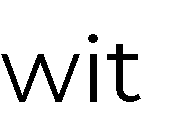
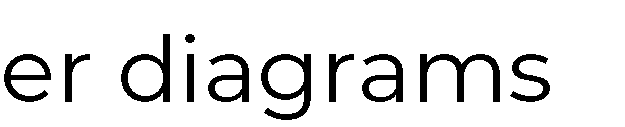
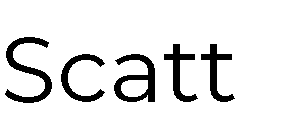
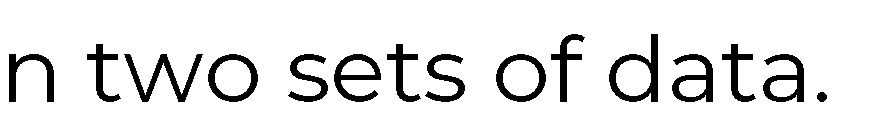
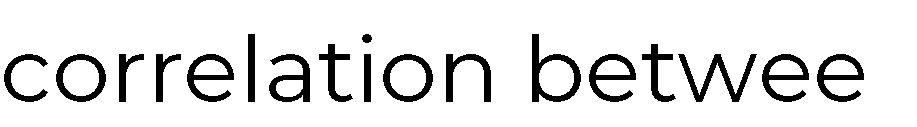
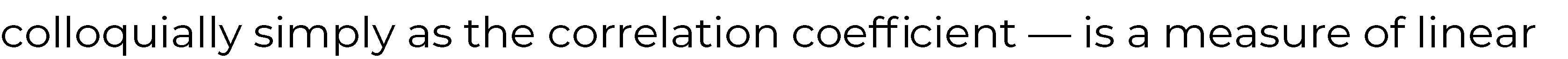
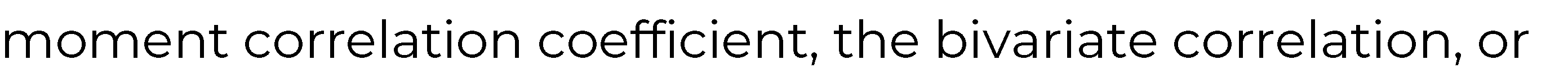
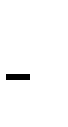
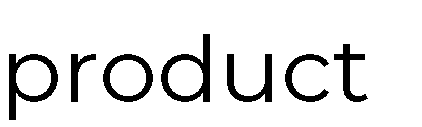
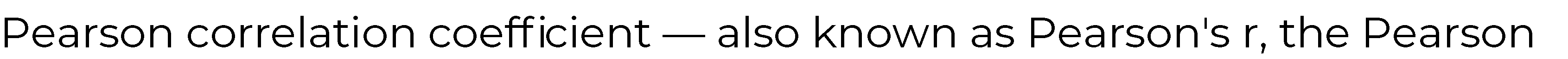
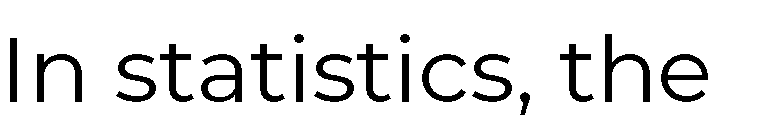
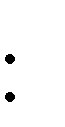
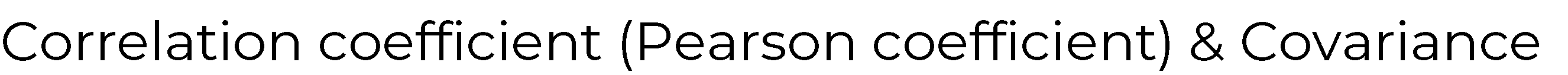
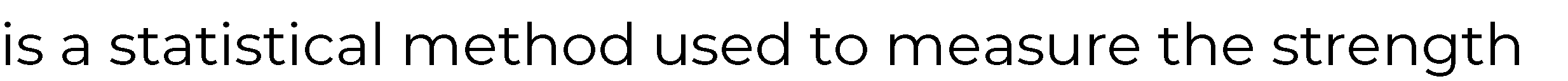
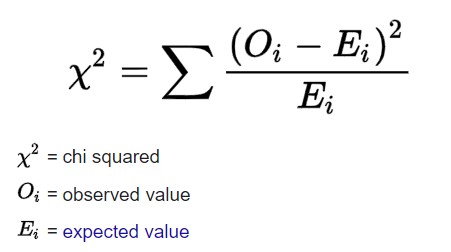
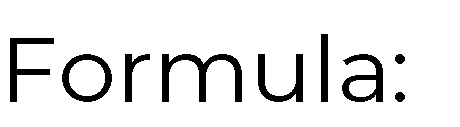


χ







**z**

**=**

**(**

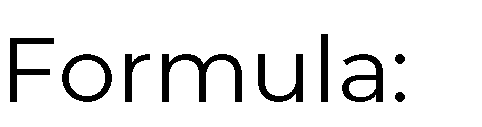
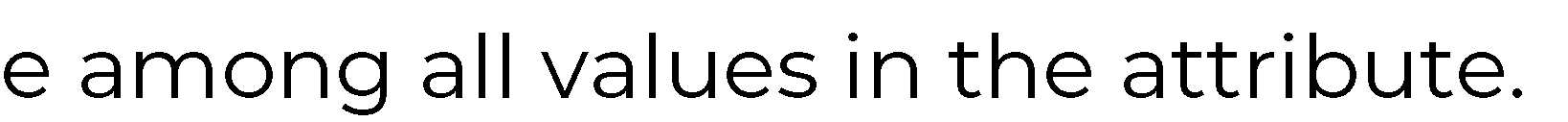
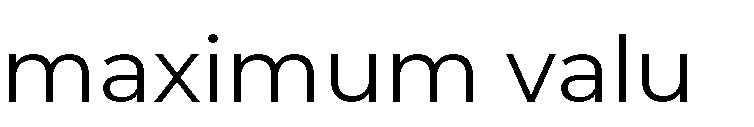
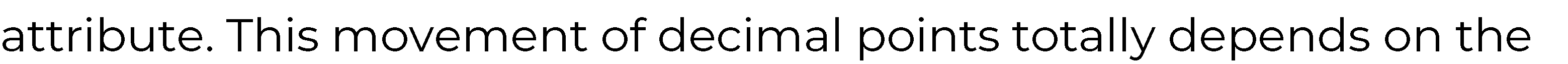
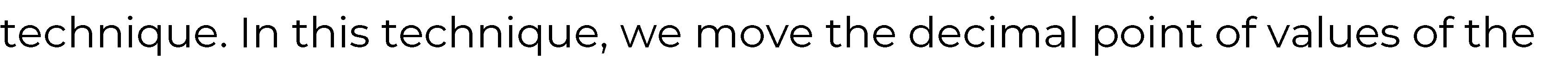
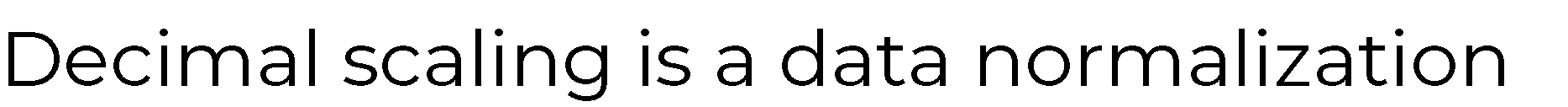
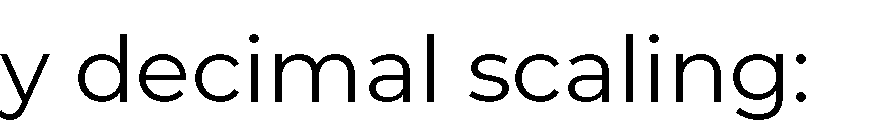
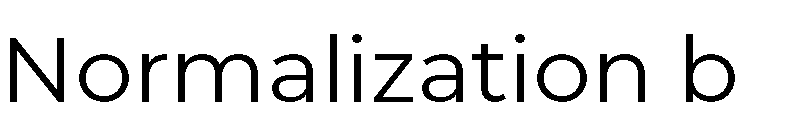
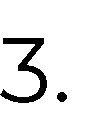
**x**

**–**

**μ)**

**/**

**σ**



Normalized value of attribute

=

**(**

**v**

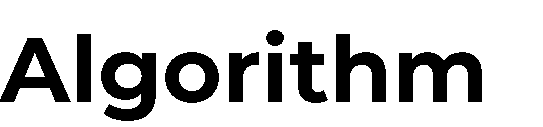
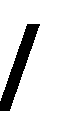
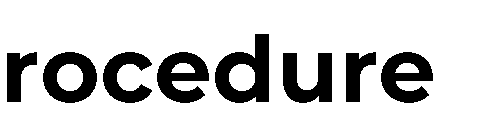
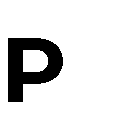
**i**

**/ 1**

**0**

**j**

**)**



|  |
| --- |
| **Chi-square Test:**  **def** chi2test\_with\_x\_y(): table = [ x,y ] stat, p, dof, expected = chi2\_contingency(table) prob = 0.95 critical = chi2.ppf(prob, dof) if abs(stat) >= critical:  print("Dependent (reject H0)") else:  print("Independent (fail to reject H0)") print(str(expected))    **Pearson Coefficient:**  **def** pearsoncoef\_with\_x\_y(): corr, \_ = pearsonr(x, y) str=""  str+='Pearsons correlation: %.3f' % corr if(corr<0):  str+="\nNegative correlation exists" if(corr>0):  str+="\nPositive correlation exists" |

|  |
| --- |
| if(corr==0): str+="\nNo correlation exists" print(str)    **Covariance:**  **def** covariance\_with\_x\_y():  cov\_mat = np.stack((x, y), axis = 0) print(str(cov\_mat)) cov\_mat = np.stack((x, y), axis = 1) print(str(cov\_mat))    **Normalization:**  **1.Min-Max:**  **def** min\_max\_norm\_x():  x\_min\_max\_scaled = x.copy() x\_min\_max\_scaled = (x\_min\_max\_scaled[5] - x\_min\_max\_scaled[0]) / (x\_min\_ma x\_scaled[5] - x\_min\_max\_scaled[0]) plt.scatter(x\_min\_max\_scaled, x\_min\_max\_scaled) plt.show() print(str(x\_min\_max\_scaled))    **2.Z-score:**  **def** z\_score\_norm\_x():  x\_z\_scaled = np.array(x) x\_z\_scaled = (x\_z\_scaled[0] - x\_z\_scaled[3]) / x\_z\_scaled.std() plt.scatter(x\_z\_scaled, x\_z\_scaled) plt.show() print(str(x\_z\_scaled))    **3.Decimal Scaling:**  **def** dec\_scale\_norm\_x(): p = x[5]  q = len(str(abs(p))) x\_des\_scaled = x[5]/10\*\*q  plt.scatter(x\_des\_scaled, x\_des\_scaled) plt.show() print(str(x\_des\_scaled)) |

