1)

1.

Abs.	Good	Medium	Strong delay	
Without alcohol	120	60	20	200 (1)
With alcohol	60	100	40	200 (2)
	180 (3)	160 (4)	60 (5)	400 (6)

(1) 200 subjects where in the group that recieved no alcohol.
(2) 200 subjects where in the group that recieved a standardized amount of alcohol.
(3) 180 subjects had a good reaction time.
(4) 160 subjects had a medium reaction time.
(5) 60 subjects had a strong reaction delay.
(6) 400 subjects were tested.

2.

$f(a_i \mid b_j)$	Good	Medium	Strong delay	
Without alcohol	2/3 (1)	3/8	1/3	
With alcohol	1/3	5/8	2/3	
	1	1	1	

(1) 2/3 of subjects who had a good reaction time didn't recieve alcohol.

$f(b_j \mid a_i)$	Good	Medium	Strong delay	
Without alcohol	3/5 (1)	3/10	1/10	1
With alcohol	3/10	1/2	2/20	1

(1) 3/5 of subjects who didn't recieve alcohol had a good reaction time.

3.

H_0: Reaction time doesn't correlate with alcohol intake

H_1: Reaction time correlates with alcohol intake

$$x^2 = \frac{\left(120-\frac{180*200}{400}\right)^2}{\frac{180*200}{400}} + \frac{\left(60-\frac{160*200}{400}\right)^2}{\frac{160*200}{400}} + \frac{\left(20-\frac{60*200}{400}\right)^2}{\frac{60*200}{400}} + \frac{\left(60-\frac{180*200}{400}\right)^2}{\frac{180*200}{400}} + \frac{\left(100-\frac{160*200}{400}\right)^2}{\frac{160*200}{400}} + \frac{\left(40-\frac{60*200}{400}\right)^2}{\frac{60*200}{400}}$$

$\approx 36.78 > 10.597$

→ i.e. we reject H_0 at significance level $\alpha = 0.005$.

2)

1.

Temperature (F): Intervall-scale

Socio-Economic Status: Ordinal

Temperature (K): Ratio-scale

Distance: Intervall-scale

Grades in School: Ordinal

Description: Nominal

Date: Intervall-scale

2.

Intervall-scale: temperature (Celsius), time (e.g. 00:00)

Ratio-scale: number of sales, weight

3)

1. **Original L1 and L∞ Distances:**
   - $\sum|A_i - B_i|$ for each time point i.
     $L1=|-1.66-0.29|+|0.30-0.89|+|-0.08-0.82|+|0.10-0.97|+|-1.17-0.53|+|-0.05-0.83|+|0.84-1.06|+|-0.66-0.67|+|0.42-0.86|+|-0.99-0.51|$

     - $|-1.66-0.29|=|-1.95|=1.95|-1.66-0.29|=|-1.95|=1.95$
     - $|0.30-0.89|=|-0.59|=0.59|0.30-0.89|=|-0.59|=0.59$
     - $|-0.08-0.82|=|-0.90|=0.90|-0.08-0.82|=|-0.90|=0.90$
     - $|0.10-0.97|=|-0.87|=0.87|0.10-0.97|=|-0.87|=0.87$
     - $|-1.17-0.53|=|-1.70|=1.70|-1.17-0.53|=|-1.70|=1.70$
     - $|-0.05-0.83|=|-0.88|=0.88|-0.05-0.83|=|-0.88|=0.88$
     - $|0.84-1.06|=|-0.22|=0.22|0.84-1.06|=|-0.22|=0.22$
     - $|-0.66-0.67|=|-1.33|=1.33|-0.66-0.67|=|-1.33|=1.33$
     - $|0.42-0.86|=|-0.44|=0.44|0.42-0.86|=|-0.44|=0.44$
     - $|-0.99-0.51|=|-1.50|=1.50|-0.99-0.51|=|-1.50|=1.50$

     $\boldsymbol{L1}=1.95+0.59+0.90+0.87+1.70+0.88+0.22+1.33+0.44+1.50=10.38$

   - $\boldsymbol{L\infty}=\max(1.95,0.59,0.90,0.87,1.70,0.88,0.22,1.33,0.44,1.50)=1.95$

1. **Offset Translation (Mean Subtraction):**

   - $|A1'-B1'|=0.912$
   - $|A2'-B2'|=0.448$
   - $|A3'-B3'|=0.138$
   - $|A4'-B4'|=0.168$
   - $|A5'-B5'|=0.662$
   - $|A6'-B6'|=0.158$
   - $|A7'-B7'|=0.818$
   - $|A8'-B8'|=0.292$
   - $|A9'-B9'|=0.598$
   - $|A10'-B10'|=0.462$

   $\boldsymbol{L1}$**offset** $=0.912+0.448+0.138+0.168+0.662+0.158+0.818+0.292+0.598+0.462=4.656$

   $\boldsymbol{L\infty}$**offset** $=\max(0.912,0.448,0.138,0.168,0.662,0.158,0.818,0.292,0.598,0.462)=0.912$

2. **Amplitude Scaling (Standard Deviation Normalization):**

   - $|A1''-B1''|=0.1897$
   - $|A2''-B2''|=0.1405$
   - $|A3''-B3''|=0.0549$
   - $|A4''-B4''|=0.4792$
   - $|A5''-B5''|=0.2207$
   - $|A6''-B6''|=0.0592$
   - $|A7''-B7''|=0.1062$
   - $|A8''-B8''|=0.1622$
   - $|A9''-B9''|=0.4327$
   - $|A10''-B10''|=0.1070$

   $\boldsymbol{L1}$**scaled** $= 0.1897+0.1405+0.0549+0.4792+0.2207+0.0592+0.1062+0.1622+0.4327+0.1070$

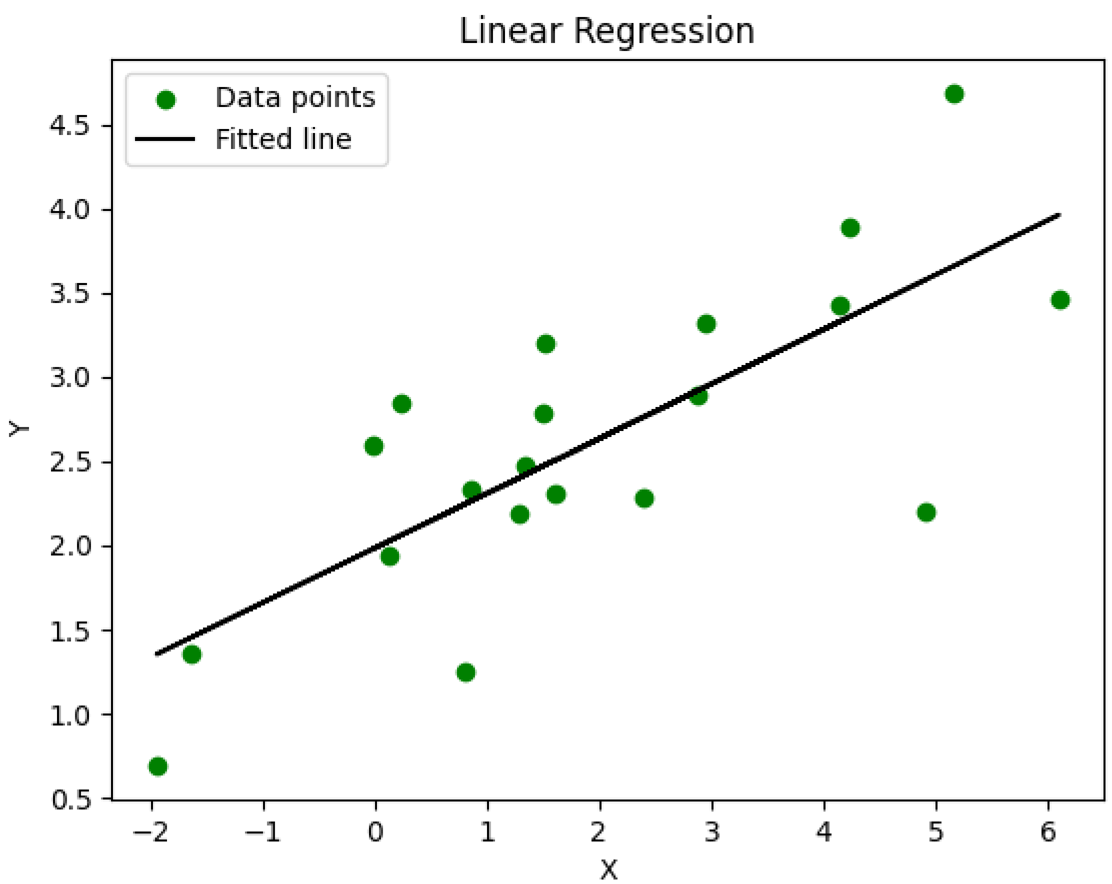
   $=1.9524$

   $\boldsymbol{L\infty}$**scaled** $=\max(0.1897,0.1405,0.0549,0.4792,0.2207,0.0592,0.1062,0.1622,0.4327,0.1070)$

   $=0.4792$

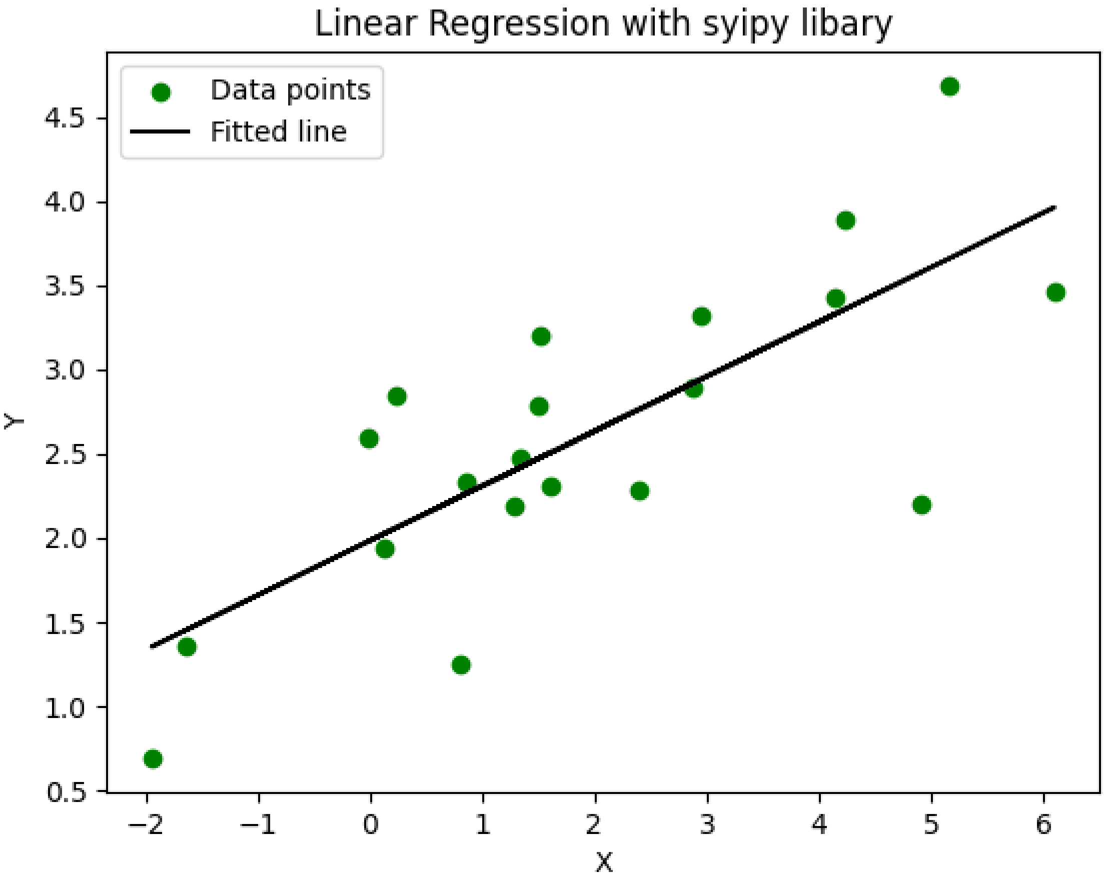
4.1

```
In [ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

data = pd.read_csv('linreg_data.csv')

x = data['X']
y = data['Y']

x_mean = np.mean(x)
y_mean = np.mean(y)

data['xy_cov'] = (data['X'] - x_mean) * (data['Y'] - y_mean)
data['x_var'] = (data['X'] - x_mean)**2

beta = data['xy_cov'].sum() / data['x_var'].sum()
alpha = y_mean - (beta * x_mean)

def predict(x):
    return alpha + beta * x

fit_line = predict(x)

plt.scatter(x, y, color='green', label='Data points')
plt.plot(x, fit_line, color='black', label='Fitted line')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Linear Regression')
plt.legend()
plt.show()
```



4.2

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import linregress

data = pd.read_csv('linreg_data.csv')

x = data['X']
y = data['Y']

regression_result = linregress(x, y)

slope = regression_result.slope
intercept = regression_result.intercept

def predict(x):
    return intercept + slope * x

fit_line = predict(x)

plt.scatter(x, y, color='green', label='Data points')
plt.plot(x, fit_line, color='black', label='Fitted line')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Linear Regression with syipy libary')
plt.legend()
plt.show()
```



4.3

In [ ]:
```
import pandas as pd

data = pd.read_csv('linreg_data.csv')

x = data['X']
```

```python
y = data['Y']

x_mean = x.mean()
y_mean = y.mean()

numerator = sum((x - x_mean) * (y - y_mean))

denominator = sum((x - x_mean)**2)

slope = numerator / denominator

intercept = y_mean - slope * x_mean

def predict(x):
    return intercept + slope * x

y_predicted_for_x_2 = predict(2)
print("Predicted Y value for X = 2:", y_predicted_for_x_2)
```

```
Predicted Y value for X = 2: 2.629747164010065
```

4.4

In order to smoothen the original data one can use the linear regression model to replace the former Y values with the predictions of the regression. This way its easier to see trends in the data.