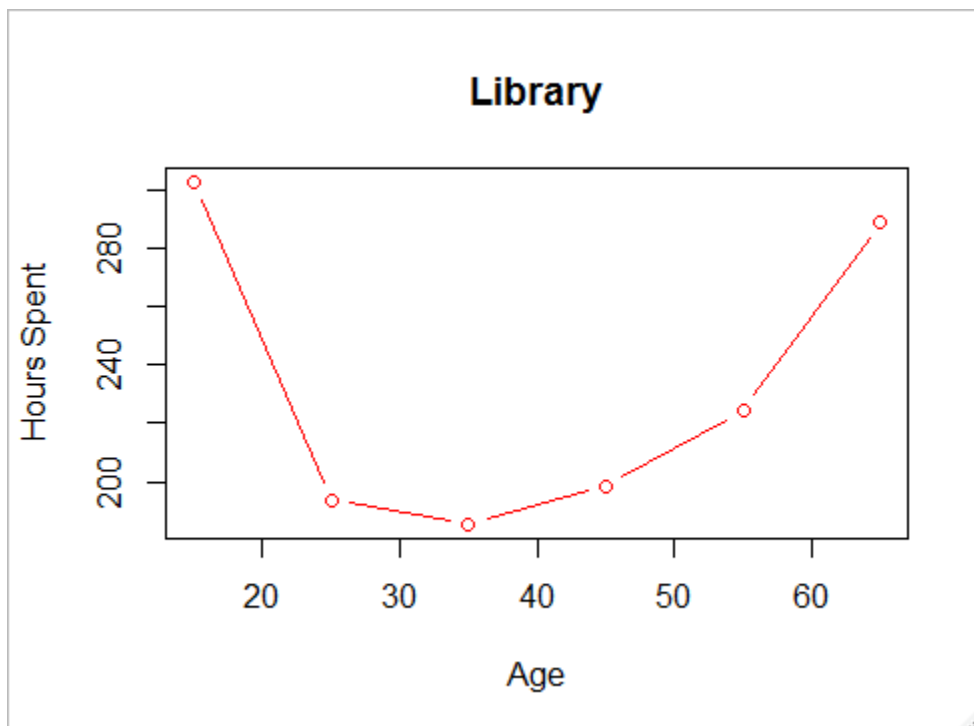


Correlation and regression

Q1)

```
> age_rep = c(15, 25, 35, 45, 55, 65)
> hours_spent = c(302.38, 193.63, 185.46, 198.49, 224.30, 288.71)
> df = data.frame(age_rep, hours_spent)
> df
  age_rep hours_spent
1     15    302.38
2     25    193.63
3     35    185.46
4     45    198.49
5     55    224.30
6     65    288.71
> plot(age_rep, hours_spent, type='b', col='red', main='Library', xlab='Age', ylab='Hours Spent')
```

>



Q2)

```
> x = c(23, 27, 28, 28, 29, 30, 31, 33, 35, 36)
> y = c(18, 20, 22, 27, 21, 29, 27, 29, 28, 29)
> df2 = data.frame(x, y)
> cov(x, y)
[1] 13.66667
> var(x, y)
[1] 13.66667
> sd(x)
[1] 3.91578
> sd(y)
[1] 4.268749
> corelation_coeff = cov(x, y)/(sd(x)*sd(y))
> corelation_coeff
[1] 0.8176052
> corelation_coeff1 = var(x, y)/(sd(x)*sd(y))
> corelation_coeff1
[1] 0.8176052
> corelation_coeff2 = cor(x, y)
> corelation_coeff2
[1] 0.8176052
> corelation_coeff3 = cor.test(x, y, method = 'pearson')
> corelation_coeff3
```

Pearson's product-moment correlation

data: x and y
t = 4.0164, df = 8, p-value = 0.003861
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
0.3874142 0.9554034
sample estimates:
cor
0.8176052

Spearman Rank Correlation

Q3)

```
> selection_score = c(44, 49, 52, 54, 47, 76, 65, 60, 63, 58, 50, 67)
> proficiency_score = c(48, 55, 45, 60, 43, 80, 58, 50, 77, 46, 47, 65)
> cor.test(selection_score, proficiency_score, method = 'spearman')
```

Spearman's rank correlation rho

data: selection_score and proficiency_score
 S = 80, p-value = 0.01102
 alternative hypothesis: true rho is not equal to 0
 sample estimates:

```
rho
0.7202797
> df3 = data.frame(selection_score, proficiency_score)
> df3
```

	selection_score	proficiency_score
1	44	48
2	49	55
3	52	45
4	54	60
5	47	43
6	76	80
7	65	58
8	60	50
9	63	77
10	58	46
11	50	47
12	67	65

Q4)

```
weight=c(15, 26, 27, 25, 25.5, 27, 32, 18, 22, 20,
26, 24);
bmi=c(13.35, 16.12, 16.74, 16.00, 13.59, 15.73, 15.65, 13.85, 16.07,
12.8, 13.65, 14.42);
```

```
model<-lm(bmi~weight);
model
```

Call:
 lm(formula = bmi ~ weight)

Coefficients:

(Intercept)	weight
10.735	0.171