STAT ASSIGNMENT

**Question

Yes,there is a relationship between student math test scores and socioeconomic variables.

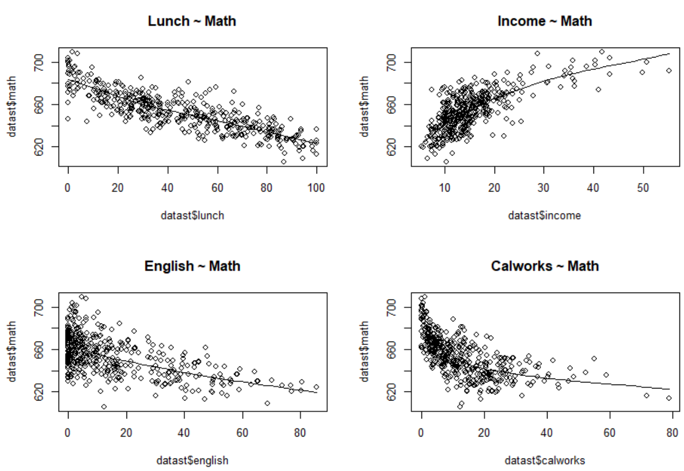
A)

```
#linear regression model
model = lm(math ~ ., data =cas)
summary(model)

dim(cas)
```

OUTPUT:

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.561e+02  4.487e+00 146.227  < 2e-16 ***
students    -6.938e-04  1.735e-03  -0.400  0.68941
teachers     5.077e-03  3.817e-02   0.133  0.89426
calworks    -1.330e-01  6.834e-02  -1.947  0.05226 .
lunch       -3.306e-01  4.273e-02  -7.735 8.05e-14 ***
computer     4.541e-03  3.266e-03   1.390  0.16519
expenditure  9.776e-04  8.645e-04   1.131  0.25877
income       6.953e-01  1.057e-01   6.576 1.47e-10 ***
english     -1.471e-01  4.097e-02  -3.591  0.00037 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.93 on 411 degrees of freedom
Multiple R-squared:  0.725,     Adjusted R-squared:  0.7197
F-statistic: 135.5 on 8 and 411 DF,  p-value: < 2.2e-16
```

1)Overall the relationship between the predictors and math test scores are linear.



2)Yes there are insignificant values namely students,teachers,calworks,computer,expenditure.

3)Income predictor is the best compared to others as the t-value is highest compared to others.

4)From the above plot we can see that the relationship between lunch-math is linear.

B)The new model is built using three variable namely math vs lunch,English,income.As we can see that rse is higher for the new model compared to older model.

D)The comparison between the two models for R2 and RSE

```
> #RSE calculation
> #model
> sqrt(deviance(model)/df.residual(model))
[1] 9.929613
> #newmodel
> sqrt(deviance(newmodel)/df.residual(newmodel))
[1] 31.49345
> #R2 calculation
> #model
> rsquare(model, data = cas)
[1] 0.7250237
> #newmodel
> rsquare(newmodel, data = cas)
[1] 0.3648683
>
```

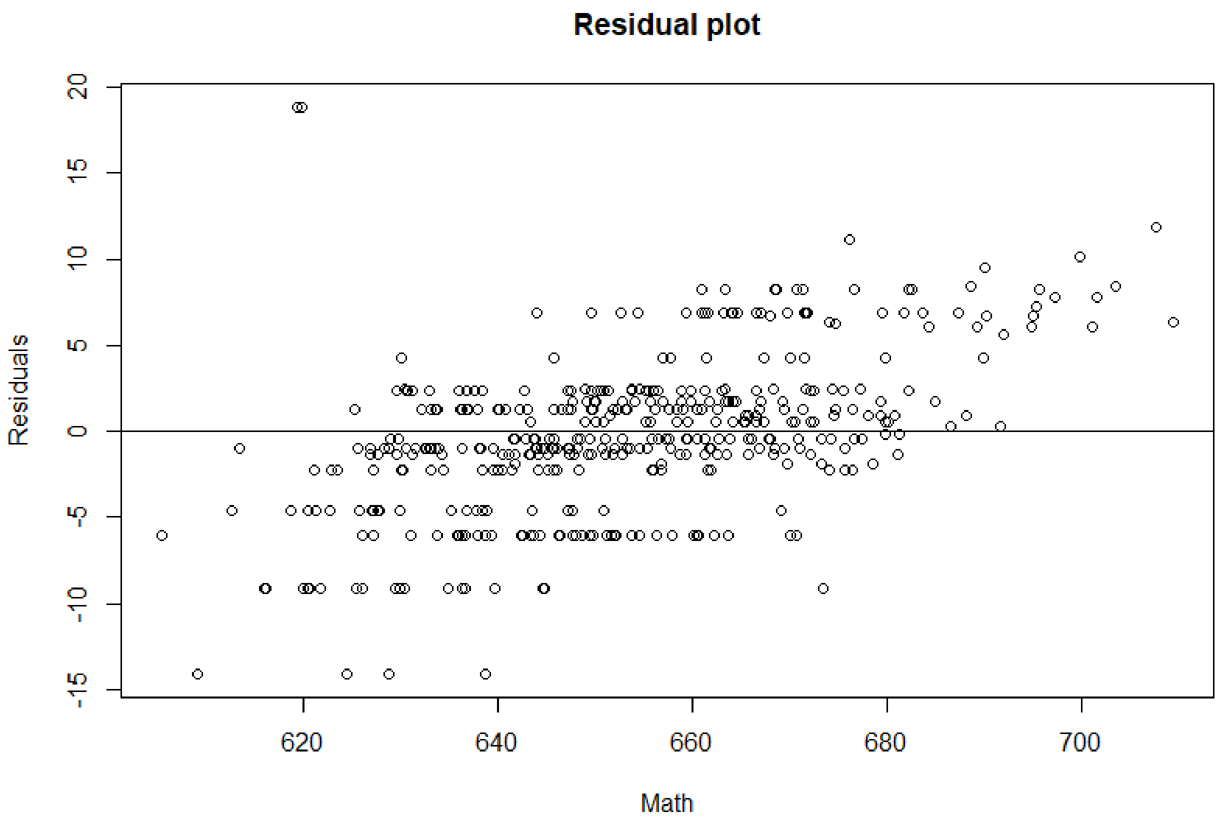
E)Residual plot for the two models

#FOR MODEL

```
> #for model
> res = resid(model)
> plot(datast$math, res,ylab="Residuals", xlab="Math",main="Residual plot")
> abline(0, 0)
>
```

#FOR NEWMODEL

```
> #for newmodel
> res2 = resid(newmodel)
> plot(datast$math, res2,ylab="Residuals", xlab="Math",main="Residual plot")
> abline(0, 0)
>
```



The residual plot shows the residuals on the vertical axis and independent variable on horizontal axis.A linear regression is used to know how far are the residuals away from the line.

F)

```
Call:
lm(formula = math ~ ., data = cas)

Residuals:
     Min       1Q   Median       3Q      Max
-31.2893  -6.9982   0.2331   5.9637  30.3968

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.561e+02  4.487e+00 146.227  < 2e-16 ***
students    -6.938e-04  1.735e-03  -0.400  0.68941
teachers     5.077e-03  3.817e-02   0.133  0.89426
calworks    -1.330e-01  6.834e-02  -1.947  0.05226 .
lunch       -3.306e-01  4.273e-02  -7.735 8.05e-14 ***
computer     4.541e-03  3.266e-03   1.390  0.16519
expenditure  9.776e-04  8.645e-04   1.131  0.25877
income       6.953e-01  1.057e-01   6.576 1.47e-10 ***
english     -1.471e-01  4.097e-02  -3.591  0.00037 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.93 on 411 degrees of freedom
Multiple R-squared:  0.725,	Adjusted R-squared:  0.7197
F-statistic: 135.5 on 8 and 411 DF,  p-value: < 2.2e-16

> data.frame(
+   R2 = rsquare(mt1, data = cas),
+   RMSE = rmse(mt1, data = cas),
+   MAE = mae(mt1, data = cas)
+ )
         R2     RMSE      MAE
1 0.7250237 9.822648 7.689363
>
```

G)Ridge Regression

```
> set.seed(1)
> cv.out <- cv.glmnet(x[train, ], y[train], alpha = 0)
> plot(cv.out)
>
> bestlam <- cv.out$lambda.min
> bestlam
[1] 1.555159
>
> ridge.pred <- predict(ridge.mod, s = bestlam,
+                       newx = x[test, ])
> mean((ridge.pred - y.test)^2)
[1] 107.8739
>
>
> out <- glmnet(x, y, alpha = 0)
> predict(out, type = "coefficients", s = bestlam)[1:9, ]
  (Intercept)      students      teachers      calworks         lunch      computer   expenditure        income
 6.540762e+02 -1.608066e-04 -1.871845e-03 -1.984999e-01 -2.736509e-01  2.903502e-03  1.086860e-03  7.125792e-01
      english
-1.760037e-01
>
```

H)Lasso Regression

```
> set.seed(1)
> cv.out <- cv.glmnet(x[train, ], y[train], alpha = 1)
> plot(cv.out)
>
> bestlam <- cv.out$lambda.min
> lasso.pred <- predict(lasso.mod, s = bestlam,
+                       newx = x[test, ])
> mean((lasso.pred - y.test)^2)
[1] 107.9297
>
> out <- glmnet(x, y, alpha = 1, lambda = grid)
> lasso.coef <- predict(out, type = "coefficients",
+                       s = bestlam)[1:9, ]
> lasso.coef
  (Intercept)      students      teachers      calworks         lunch      computer   expenditure        income
 6.586702e+02  0.000000e+00  0.000000e+00 -8.817604e-02 -3.457893e-01  0.000000e+00  4.944489e-04  6.960045e-01
      english
-1.260826e-01
>
> lasso.coef[lasso.coef != 0]
  (Intercept)      calworks         lunch   expenditure        income       english
 6.586702e+02 -8.817604e-02 -3.457893e-01  4.944489e-04  6.960045e-01 -1.260826e-01
>
```