Module 1 Assesment

Gafur Mammadov

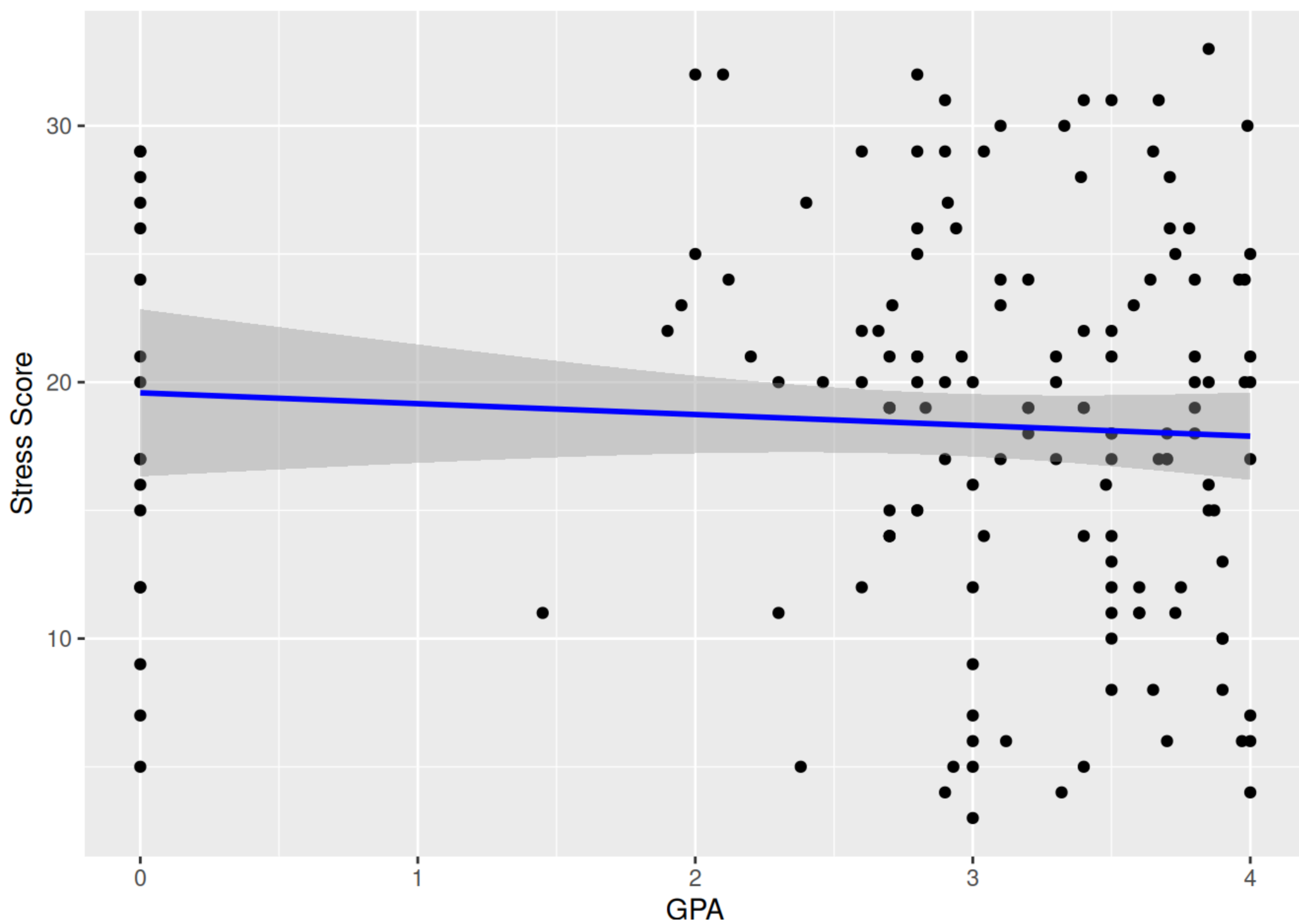
2025-01-27

```
survey_fall2023 <- read.csv("~/Documents/math133/datasets/survey_fall2023.csv")
```

Problem 1

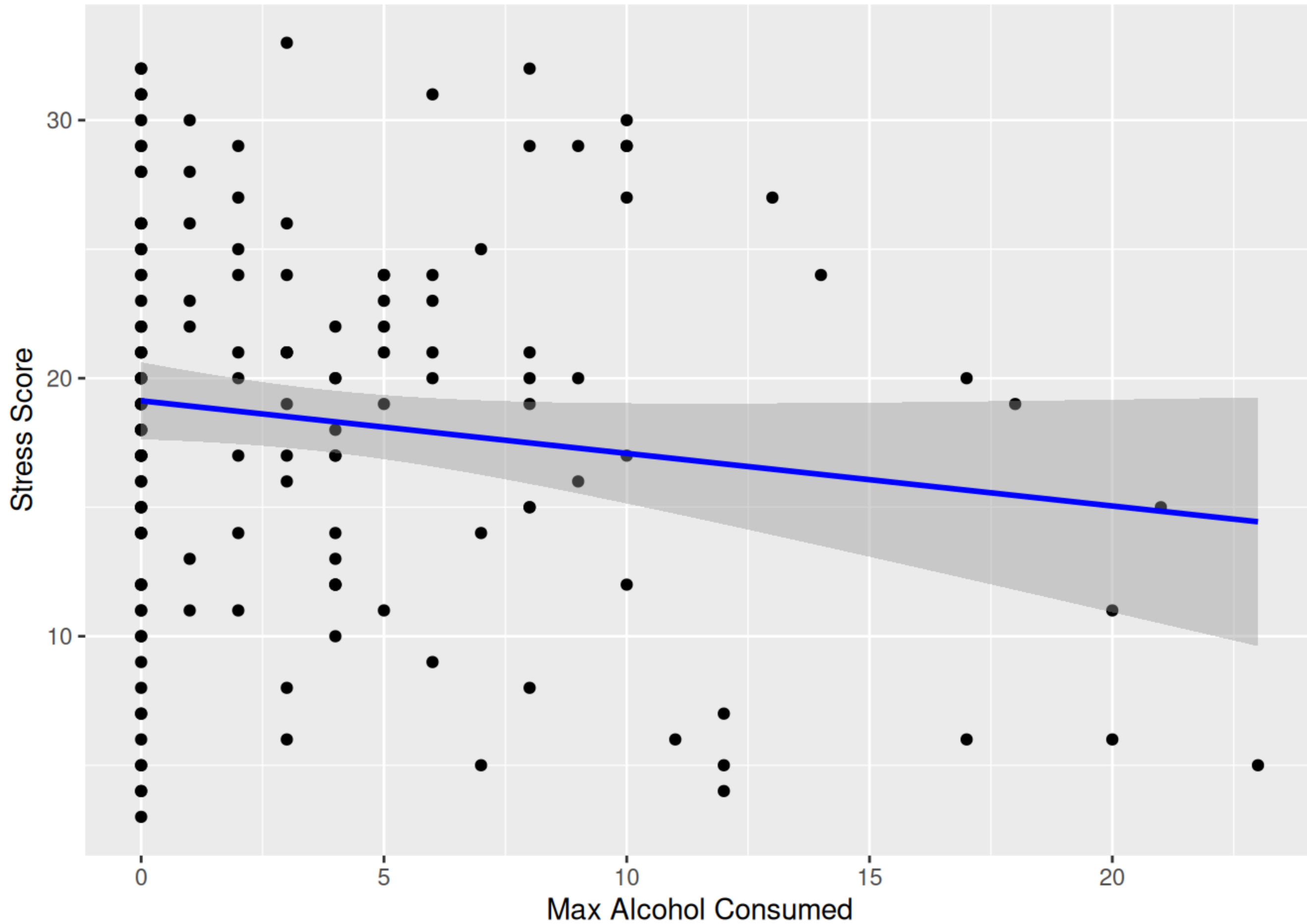
```
survey_fall2023 %>% ggplot(aes(x=gpa, y=stress_score))+
  geom_point() +
  labs(x="GPA", y="Stress Score") +
  geom_smooth(method="lm", color="blue", se=TRUE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



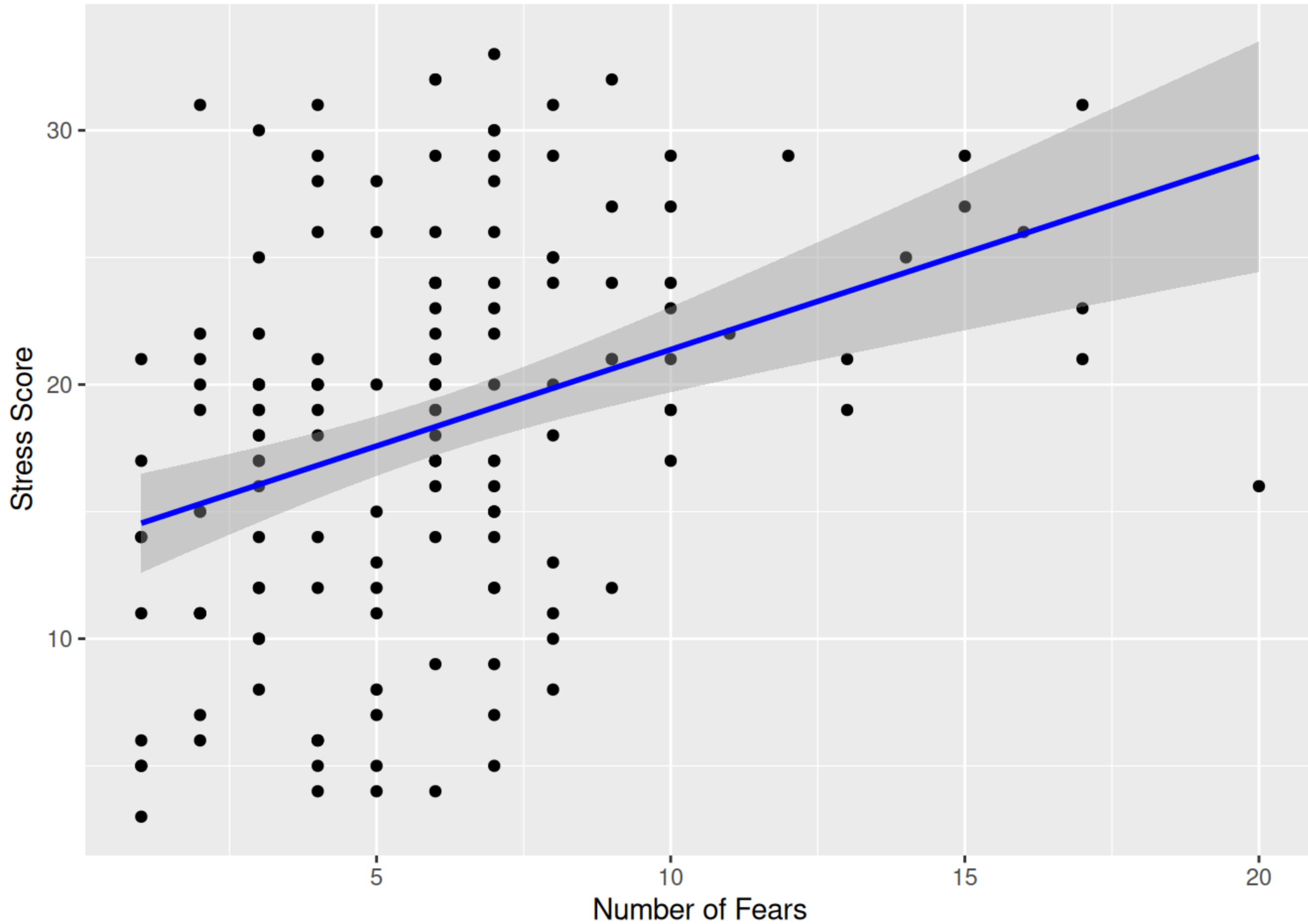

```
survey_fall2023 %>% ggplot(aes(x=maximum_alcohol_consumed, y=stress_score))+
  geom_point() +
  labs(x="Max Alcohol Consumed", y="Stress Score") +
  geom_smooth(method="lm", color="blue", se=TRUE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```
survey_fall2023 %>% ggplot(aes(x=number_of_fears, y=stress_score))+
  geom_point() +
  labs(x="Number of Fears", y="Stress Score") +
  geom_smooth(method="lm", color="blue", se=TRUE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



In between these 3 features, I think number_of_fears is a better predictor because the trend line seems stronger than the other ones.

Problem 2

GPA vs Stress Score

```
gpa_lm = lm(stress_score~gpa, data=survey_fall2023)
y = survey_fall2023$stress_score
yhat = predict(gpa_lm, survey_fall2023)
n = nrow(survey_fall2023)
SSE = sum((y - yhat)^2)
MSE = SSE/n
SST = sum((y-mean(y))^2)
R2 = 1 - SSE/SST
c(R2)
```

```
## [1] 0.004132162
```

```
summary(gpa_lm)
```

```
##
## Call:
## lm(formula = stress_score ~ gpa, data = survey_fall2023)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.3167  -5.3291   0.6896   5.7361  15.0424
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  19.5844     1.6508  11.864   <2e-16 ***
## gpa          -0.4225     0.5356  -0.789    0.431
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.551 on 150 degrees of freedom
## Multiple R-squared:  0.004132,   Adjusted R-squared:  -0.002507
## F-statistic: 0.6224 on 1 and 150 DF,  p-value: 0.4314
```

Max Alcohol Consumed vs Stress Score

```
maximum_alcohol_consumed_lm = lm(stress_score~maximum_alcohol_consumed, data=survey_fall
2023)
y = survey_fall2023$stress_score
yhat = predict(maximum_alcohol_consumed_lm, survey_fall2023)
n = nrow(survey_fall2023)
SSE = sum((y - yhat)^2)
MSE = SSE/n
SST = sum((y-mean(y))^2)
R2 = 1 - SSE/SST
c(R2)
```

```
## [1] 0.01819558
```

```
summary(maximum_alcohol_consumed_lm)
```

```
##
## Call:
## lm(formula = stress_score ~ maximum_alcohol_consumed, data = survey_fall2023)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.1246  -5.1708   0.3213   5.3340  14.5060
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               19.1246     0.7563  25.289   <2e-16 ***
## maximum_alcohol_consumed  -0.2038     0.1222  -1.667   0.0975 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.497 on 150 degrees of freedom
## Multiple R-squared:  0.0182, Adjusted R-squared:  0.01165
## F-statistic:  2.78 on 1 and 150 DF,  p-value: 0.09754
```

Number of Fears vs Stress Score

```
number_of_fears_lm = lm(stress_score~number_of_fears, data=survey_fall2023)
y = survey_fall2023$stress_score
yhat = predict(number_of_fears_lm, survey_fall2023)
n = nrow(survey_fall2023)
SSE = sum((y - yhat)^2)
MSE = SSE/n
SST = sum((y-mean(y))^2)
R2 = 1 - SSE/SST
c(R2)
```

```
## [1] 0.1313472
```

```
summary(number_of_fears_lm)
```

```
##
## Call:
## lm(formula = stress_score ~ number_of_fears, data = survey_fall2023)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.3401  -4.4269   0.3833   4.6688  15.6954
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      13.7869     1.1204  12.305  < 2e-16 ***
## number_of_fears   0.7589     0.1593   4.762 4.47e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.052 on 150 degrees of freedom
## Multiple R-squared:  0.1313, Adjusted R-squared:  0.1256
## F-statistic: 22.68 on 1 and 150 DF,  p-value: 4.469e-06
```

After calculations turns out Number of Fears has the highest R squared value among all. However, it is still very low. Here I added the summaries of lm functions and it turns out that p-value of number of fears indicates that it is statistically significant (<0.05), while max alcohol consumption and GPA are not significant.

Problem 3

GPA vs Stress Score

```
training_indices = sample(n, round(0.7*n, 0))
training_set = survey_fall2023[training_indices,]
test_set = survey_fall2023[-training_indices,]

train_lm = lm(stress_score~gpa, data = training_set)
ytest = test_set$stress_score
yhattest = predict(train_lm, test_set)

n = nrow(survey_fall2023)
SSE = sum((ytest - yhattest)^2)
MSE = SSE/n
RMSE = sqrt(MSE)
SST = sum((ytest-mean(ytest))^2)
R2 = 1 - SSE/SST
c(RMSE)
```

```
## [1] 3.947515
```

Max Alcohol Consumed vs Stress Score

```
training_indices = sample(n, round(0.7*n, 0))
training_set = survey_fall2023[training_indices,]
test_set = survey_fall2023[-training_indices,]

train_lm = lm(stress_score~maximum_alcohol_consumed, data = training_set)
ytest = test_set$stress_score
yhattest = predict(train_lm, test_set)

n = nrow(survey_fall2023)
SSE = sum((ytest - yhattest)^2)
MSE = SSE/n
RMSE = sqrt(MSE)
SST = sum((ytest-mean(ytest))^2)
R2 = 1 - SSE/SST
c(RMSE)
```

```
## [1] 4.655413
```

Number of Fears vs Stress Score

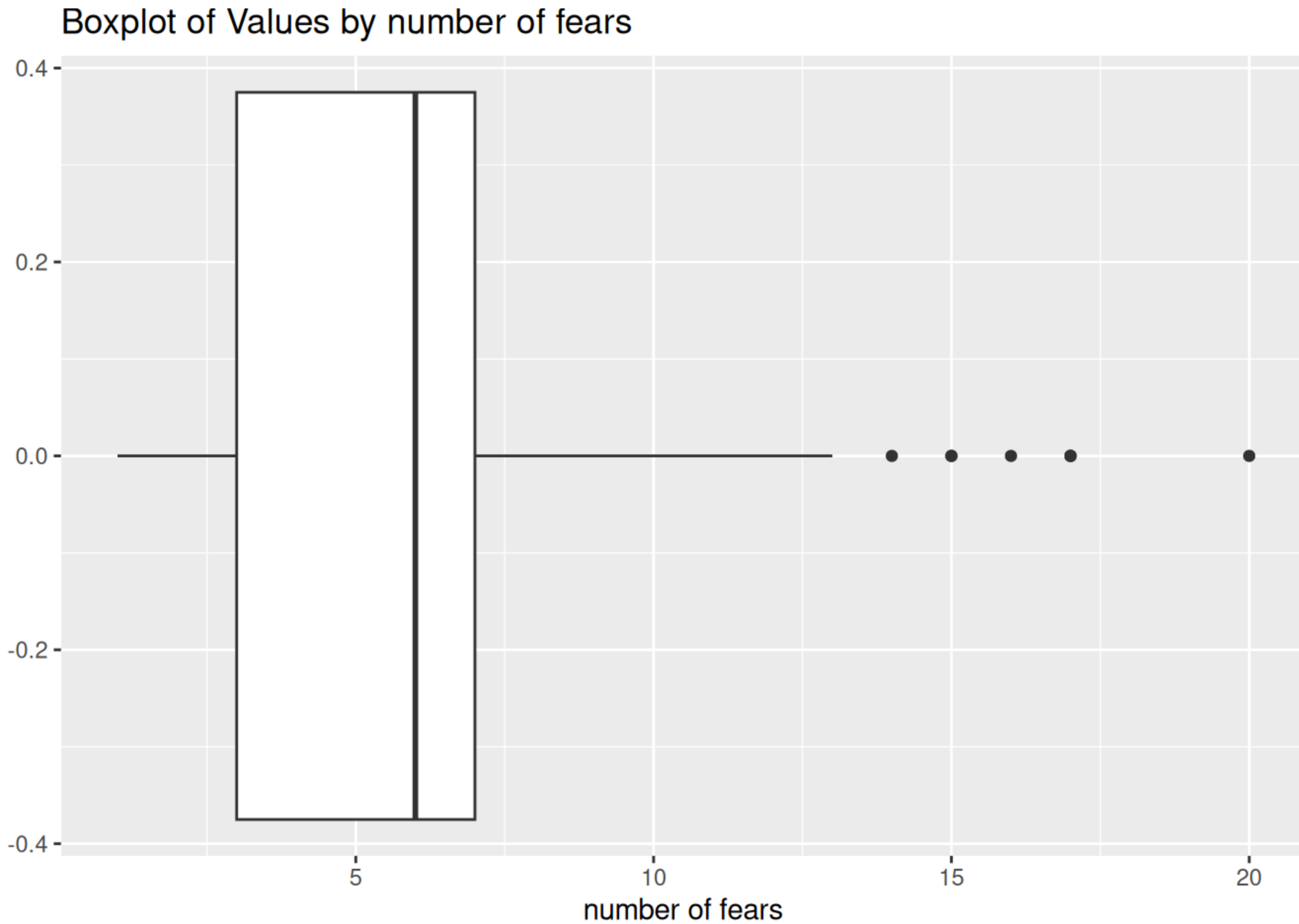
```
training_indices = sample(n, round(0.7*n, 0))
training_set = survey_fall2023[training_indices,]
test_set = survey_fall2023[-training_indices,]

train_lm = lm(stress_score~number_of_fears, data = training_set)
ytest = test_set$stress_score
yhattest = predict(train_lm, test_set)

n = nrow(survey_fall2023)
SSE = sum((ytest - yhattest)^2)
MSE = SSE/n
RMSE = sqrt(MSE)
SST = sum((ytest-mean(ytest))^2)
R2 = 1 - SSE/SST
c(RMSE)
```

```
## [1] 3.831577
```

After calculations turns out Number of Fears has the lowest RMSE value among all. That’s why Number of Fears is the best feature.

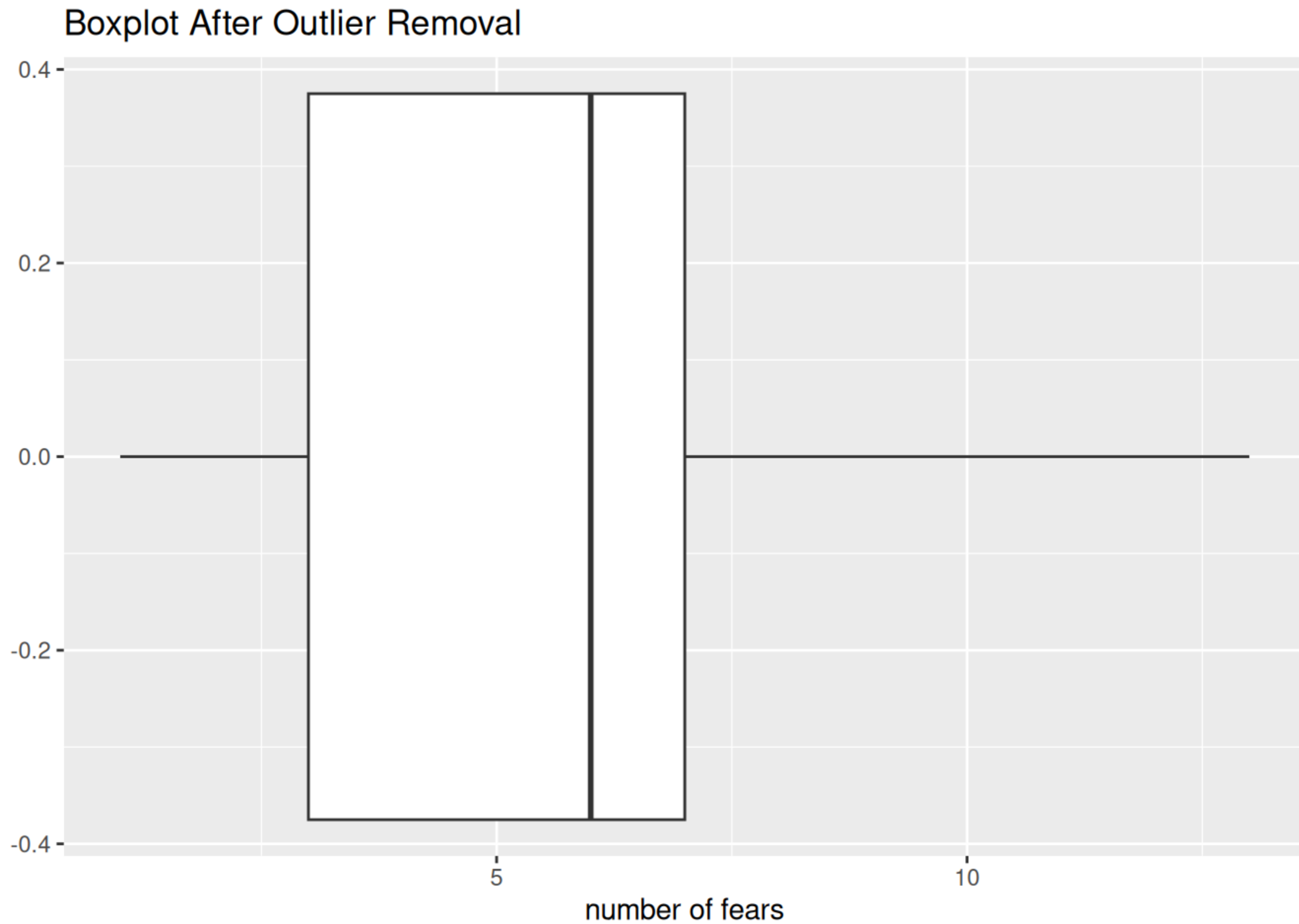
Problem 4

```
ggplot(aes(x = number_of_fears), data= survey_fall2023) +
  geom_boxplot() +
  ggtitle("Boxplot of Values by number of fears") +
  xlab("number of fears")
```



Removing outliers

```
outliers <- boxplot.stats(survey_fall2023$number_of_fears)$out

survey_fall2023_clean <- survey_fall2023[!survey_fall2023$number_of_fears %in% outliers,
]

ggplot(aes(x = number_of_fears), data= survey_fall2023_clean) +
  geom_boxplot() +
  ggtitle("Boxplot After Outlier Removal") +
  xlab("number of fears")
```



R2 after outlier removal

```
number_of_fears_clean_lm = lm(stress_score~number_of_fears, data=survey_fall2023_clean)
y = survey_fall2023_clean$stress_score
yhat = predict(number_of_fears_clean_lm, survey_fall2023_clean)
n = nrow(survey_fall2023_clean)
SSE = sum((y - yhat)^2)
MSE = SSE/n
SST = sum((y-mean(y))^2)
R2 = 1 - SSE/SST
c(R2)
```

```
## [1] 0.1129021
```

After removing the outliers, the R squared value went down which means those outliers were contributing to the linear regression model.