```
#R script for the bachelors thesis data analysis


# Define the cutoff point
cutoff <- 0


#implementing regression discontinuity (at least X nominations vs. distance)

#1) the code

# Perform the RDD analysis with additional arguments
rd_result <- rdrobust(y = Dataset$awarded_or_nominated_1, x = Dataset$distance, covs =
covariates, cluster = Dataset$clustering_ID, c = cutoff, p = 1, kernel = "triangular", level = 90)

# Print the summary of the results
summary(rd_result)



#2) the plot

# plot the RDD
rdplot(y = Dataset$awarded_or_nominated_1, x = Dataset$distance, covs = covariates, x.label =
"Distance from the cut-off", y.label = "probability of getting at least one award or nomination", title
= "",  c = cutoff, p = 1, h = 25 , kernel = "triangular", nbins = 30, x.lim = c(-25, 25))







#implementing regression discontinuity (nominated best film vs. distance)

#1) the code

# Perform the RDD analysis with additional arguments
rd_result <- rdrobust(y = Dataset$nominated_best_film, x = Dataset$distance, covs = covariates,
cluster = Dataset$clustering_ID,  c = cutoff, p = 1, kernel = "triangular", level = 90)

# Print the summary of the results
summary(rd_result)


#2) the plot

# plot the RDD
rdplot(y = Dataset$nominated_best_film, x = Dataset$distance, covs = covariates, x.label =
"Distance from the cut-off", y.label = "probability of being nominated for a best film",c = cutoff, p
= 1, h = 8.8, kernel = "triangular", nbins = 30, x.lim = c(-8.8, 8.8) )
```

#implementing regression discontinuity (number of nominations vs. distance)

#1) the code

```
# Perform the RDD analysis with additional arguments
rd_result <- rdrobust(y = Dataset$nominations, x = Dataset$distance, h =50, c = cutoff, p = 1,
kernel = "triangular", level = 90)

# Print the summary of the results
summary(rd_result)
```

#2) the plot

```
# plot the RDD
rdplot(y = Dataset$nominations, x = Dataset$distance, x.label = "Distance from the cut-off", y.label
= "number of nominations",c = cutoff, h =50, p = 1, kernel = "triangular", nbins = 30)
```

#implementing regression discontinuity (ratings vs. distance)

#1) the code

```
# Perform the RDD analysis with additional arguments
rd_result <- rdrobust(y = Dataset$viewer_ratings, x = Dataset$distance, covs = covariates, cluster =
Dataset$clustering_ID,  c = cutoff, h = 50, p = 1, kernel = "triangular", level = 90)

# Print the summary of the results
summary(rd_result)
```

#2) the plot

```
# plot the RDD
```

```
rdplot(y = Dataset$viewer_ratings, x = Dataset$distance, covs = covariates, x.label = "Distance from the cut-off", y.label = "viewer ratings", title = "", c = cutoff, h =50, p = 1, kernel = "triangular", nbins = 100, x.lim = c(-50, 50))
```

```
#implementing regression discontinuity (box_office vs. distance)

#1) the code

# Perform the RDD analysis with additional arguments
rd_result <- rdrobust(y = Dataset$log_cpi_box_office, x = Dataset$distance, covs = covariates, cluster = Dataset$clustering_ID, c = cutoff, p = 1, kernel = "triangular", level = 90)

# Print the summary of the results
summary(rd_result)

#2) the plot

# plot the RDD
rdplot(y = Dataset$log_cpi_box_office, x = Dataset$distance, c = cutoff, covs = covariates, x.label = "Distance from the cut-off", y.label = "logarithm of inflation ajusted box office", title = "" , p = 1, h = 25, kernel = "triangular", nbins = 50, x.lim = c(-25, 25))
```

```
#implementing regression discontinuity (shows vs. distance)

#1) the code


# Perform the RDD analysis with additional arguments
rd_result <- rdrobust(y =  log(Dataset$shows), x = Dataset$distance, c = cutoff, p = 1, kernel = "triangular", level = 90)

# Print the summary of the results
summary(rd_result)

#2) the plot
```

```
# plot the RDD
rdplot(y = log(Dataset$shows), x = Dataset$distance, c = cutoff, p = 1, kernel = "triangular", nbins
= 1000)
```

```
#implementing regression discontinuity (viewer_count vs. distance)

#1) the code

# Perform the RDD analysis with additional arguments
rd_result <- rdrobust(y =  log(Dataset$viewer_count), x = Dataset$distance, c = cutoff, p = 1, kernel
= "triangular", level = 90)

# Print the summary of the results
summary(rd_result)

#2) the plot

# plot the RDD
rdplot(y = log(Dataset$viewer_count), x = Dataset$distance, c = cutoff, p = 1, kernel = "triangular",
nbins = 1000)
```

```
#implementing regression discontinuity ( completion_3Y vs. distance)

#selecting observations up to 2021

Completion_data <- filter(Dataset, Dataset$year < 2021 )


#1) the code

# Perform the RDD analysis with additional arguments
rd_result <- rdrobust(y =  Completion_data$completion_3Y, x = Completion_data$distance, covs =
covariates_short, cluster = Completion_data$clustering_ID, c = cutoff, h = 50,  p = 1, kernel =
"triangular", level = 90)
```

```
# Print the summary of the results
summary(rd_result)

#2) the plot

# plot the RDD
rdplot(y =  Completion_data$completion_3Y, x = Completion_data$distance, covs =
covariates_short, x.label = "distance from the cut-off", y.label = "probability of the film being made
in 3 years from the subsidy request", title = "", h = 25, p = 1, kernel = "triangular", nbins = 30, x.lim
= c(-25, 25))




##############################################

#implementing liner regression of logarithm of inflation adjusted box office on score

model_result <- lm(Dataset$log_cpi_box_office ~ Dataset$score + covariates)

# Clustered standard errors at the level of 'cluster'
clustered_vcov <- vcovCL(model_result, cluster = ~ Dataset$clustering_ID)

# Summary of the model with clustered standard errors
coeftest(model_result, vcov = clustered_vcov)

ggplot(Dataset, aes(x = score, y = log_cpi_box_office)) +
  geom_point() +                        # Plot the data points
  geom_smooth(method = "lm", se = FALSE) +   # Add the regression line
  labs(title = "Linear Regression of viewer ratings on score",
       x = "score",
       y = "box office")




#implementing liner regression of viewer ratings on score

model_result <- lm(Dataset$viewer_ratings ~ Dataset$score + covariates)

# Clustered standard errors at the level of 'cluster'
clustered_vcov <- vcovCL(model_result, cluster = ~ Dataset$clustering_ID)

# Summary of the model with clustered standard errors
coeftest(model_result, vcov = clustered_vcov)

ggplot(Dataset, aes(x = score, y = viewer_ratings)) +
  geom_point() +                        # Plot the data points
```

```
  geom_smooth(method = "lm", se = FALSE) +   # Add the regression line
  labs(title = "Linear Regression of viewer ratings on score",
       x = "score",
       y = "viewer ratings")



#implementing liner regression of awards and nominations on score

model_result <- lm(Dataset$awarded_or_nominated_1 ~ Dataset$score + covariates)

# Clustered standard errors at the level of 'cluster'
clustered_vcov <- vcovCL(model_result, cluster = ~ Dataset$clustering_ID)

# Summary of the model with clustered standard errors
coeftest(model_result, vcov = clustered_vcov)

ggplot(Dataset, aes(x = score, y = awarded_or_nominated_1)) +
  geom_point() +                          # Plot the data points
  geom_smooth(method = "lm", se = FALSE) +   # Add the regression line
  labs(title = "Linear Regression of viewer ratings on score",
       x = "score",
       y = "awarded_or_nominated_1")



#density of the score variable (continuity at the cut-off)

density_test <- rddensity(Dataset$distance, c = 0)
summary(density_test)

vector <- seq(from = -30, to = 30, by = 3)

rdplotdensity(density_test, Dataset$distance, xlabel = "distance", ylabel ="estimated density of the
distance" , plotN = 100, histBreaks = vector)


################################################################################


#Covariate balance test (aka covariate continuity at the cut-off) - budget

balance_test <- rdrobust(y = Dataset$log_cpi_budget, x = Dataset$distance, p = 1)

summary(balance_test)

rdplot(y = Dataset$log_cpi_budget, x = Dataset$distance, p = 1, title = "", x.label = "distance",
y.label = "logarithm of inflation adjusted budget" )
```