```
#assignment part 2

# create the new model
model2b = lm(pain ~ age + sex + STAI_trait + pain_cat + mindfulness + cortisol_serum, data
= data_sample_2)
model2b

model3b = lm(pain ~ age + sex + STAI_trait + pain_cat + mindfulness + cortisol_serum +
weight + IQ + household_income, data = data_sample_2)
model3b
#the initial model


#re-run data and model diagnostics
# recheck the assumption of normality of residuals
describe(residuals(model2b))
residuals_model2b = enframe(residuals(model2b))
residuals_model2b %>%
  ggplot() + aes(x = value) + geom_histogram()

describe(residuals(model3b))
residuals_model3b = enframe(residuals(model3b))
residuals_model3b %>%
  ggplot() + aes(x = value) + geom_histogram()


#residuals vs Leverage
model2b %>%
  plot(which = 5)
model3b %>%
  plot(which = 5)

#cook´s distance
model2b %>%
  plot(which = 4)
model3b %>%
  plot(which = 4)

# QQ plot
model2b %>%
  plot(which = 2)
model3b %>%
  plot(which = 2)

# histogram
residuals_model2b = enframe(residuals(model2b))
residuals_model2b %>%
  ggplot() + aes(x = value) + geom_histogram()
residuals_model3b = enframe(residuals(model3b))
residuals_model3b %>%
  ggplot() + aes(x = value) + geom_histogram()

# skew and kurtosis
describe(residuals(model2b))
describe(residuals(model3b))


shapiro.test(residuals(model2b))
shapiro.test(residuals(model3b))


#Linearity
model2b %>%
  residualPlots()
```

```
model3b %>%
  residualPlots()


#Homoscedasticty
model2b %>%
  plot(which = 3)
model2b %>%
  ncvTest() # NCV test

model3b %>%
  plot(which = 3)
model3b %>%
  ncvTest()

model2b %>%
  vif()
model3b %>%
  vif()


# comparing the models on data with and without the outliers
summary(model2b)
summary(model3b)


#backward regression
model3c = step(model3b, direction = "backward")
# mindfullness
# age
# cortisol_serum
# pain_cat

#new model with the new predictors
model_backward = lm(pain ~ age + pain_cat + mindfulness + cortisol_serum, data =
data_sample_2) #is it the right dataset??
model_backward
#run the assignment 1 model (theorybased model)
model_theorybased = lm(pain ~ age + sex + STAI_trait + pain_cat + mindfulness +
cortisol_serum, data = data_sample_2)
model_theorybased


#compare the models
AIC(model_backward)
AIC(model_theorybased)

anova(model_backward, model_theorybased)


#New dataset
data_sample_3 = read_csv("https://tinyurl.com/87v6emky")
View(data_sample_3)

#test the models
summary(model_backward)
summary(model_theorybased)


pred_backward = predict(model_backward, data_sample_3)
pred_theorybased = predict(model_theorybased, data_sample_3)
RSS_backward = sum((data_sample_3[, "pain"] - pred_backward)^2)
RSS_theorybased = sum((data_sample_3[, "pain"] - pred_theorybased)^2)
RSS_backward
# 248.1698
```

```
RSS_theorybased
# 241.9065


summary(model_backward)


#run summary function and 95% confidence intervals
coef_table = function(model){
  require(lm.beta)
  mod_sum = summary(model)
  mod_sum_p_values = as.character(round(mod_sum$coefficients[,4], 3))
  mod_sum_p_values[mod_sum_p_values != "0" & mod_sum_p_values != "1"] =
substr(mod_sum_p_values[mod_sum_p_values != "0" & mod_sum_p_values != "1"], 2,
nchar(mod_sum_p_values[mod_sum_p_values != "0" & mod_sum_p_values != "1"]))
  mod_sum_p_values[mod_sum_p_values == "0"] = "<.001"


  mod_sum_table = cbind(as.data.frame(round(cbind(coef(model), confint(model), c(0,
lm.beta(model)$standardized.coefficients[c(2:length(model$coefficients))])), 2)),
mod_sum_p_values)
  names(mod_sum_table) = c("b", "95%CI lb", "95%CI ub", "Std.Beta", "p-value")
  mod_sum_table["(Intercept)","Std.Beta"] = "0"
  return(mod_sum_table)
}
coef_table (model_backward) #raport in table and put in appendix
coef_table (model_theorybased)



AIC(model3b)
AIC(model_backward)

anova(model3b, model_backward)
```