Regression Analysis

Project

2022-10-04

Introduction

This project seeks to perform a descriptive and later a regression analysis

```
#load data
pension_data <- readxl::read_excel("C:/Users/USER/Downloads/sample_data_regression.xlsx")


#clean the column names using janitor
pension_data<- janitor::clean_names(pension_data)

#renaming some of the columns
pension_data <- dplyr::rename(pension_data, contributions = "density_of_contributions", net_investment_


head(pension_data)
```

```
## # A tibble: 6 x 9
##   ref   mode        fund_value contributions administration_f~ net_investment_i~
##   <chr> <chr>            <dbl>         <dbl>             <dbl>             <dbl>
## 1 208   Defined C~ 18734901000     154379000           4012000        1946250000
## 2 1790  Defined C~ 15557691468    1260511196          32431942        1528160685
## 3 1760  Defined C~  9581585400     683544820           4289026         970010962
## 4 1676  Defined C~  6516360000     591333000           7205000         749025000
## 5 1886  Defined C~  5813138385     507520341           2660076         776058999
## 6 103   Defined C~  3989324593     177451512           1748010         572271357
## # ... with 3 more variables: investment_fees <dbl>, custodian_fees <dbl>,
## #   year <dbl>
```

```
#the data was precleaned in google sheets. Therefore, we proceed directly to visualization
```

We get a summary of the data

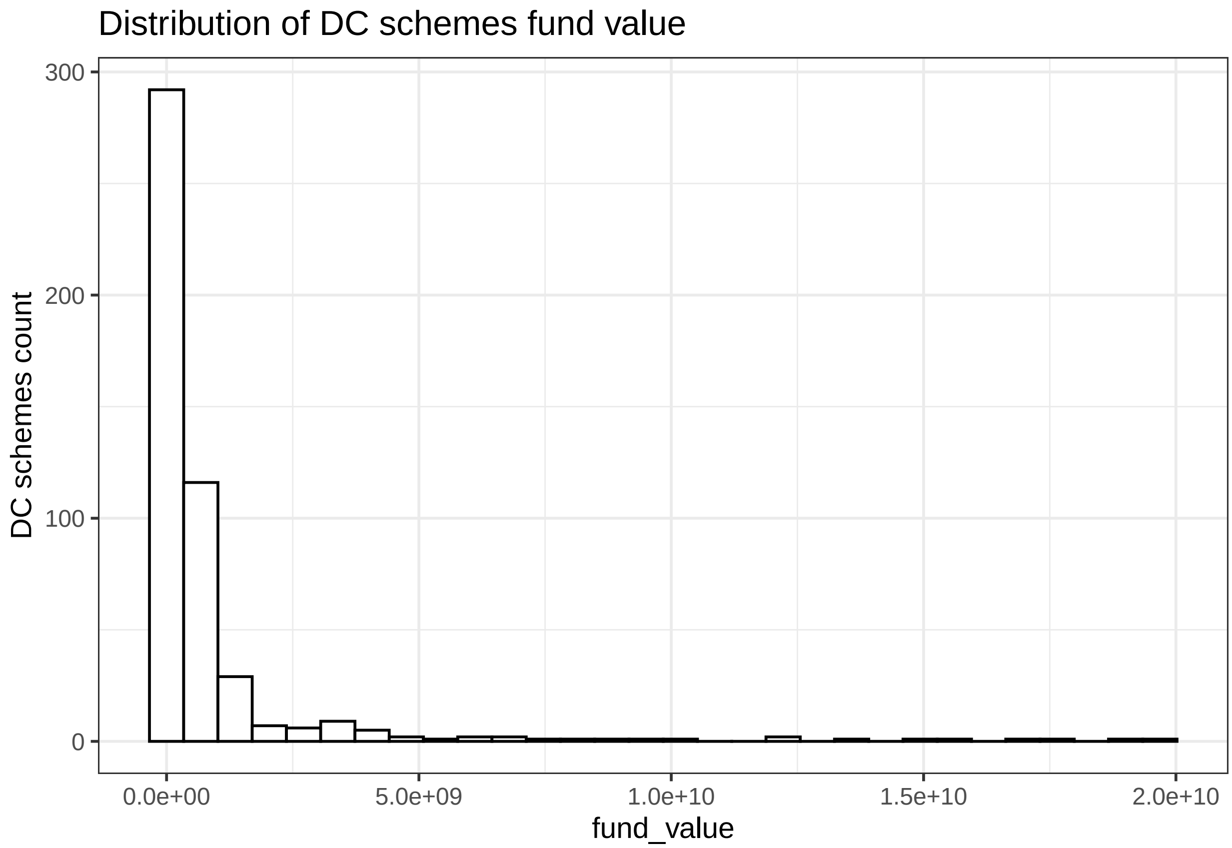
```
#create vectors to make a table for summary statistics

variables <- c("fund_value", "contributions", "adminstration_fees", "net_investment_income", "investmen
mean <-c( mean(pension_data$fund_value), mean(pension_data$contributions), mean(pension_data$administra

standard_deviation <-c( sd(pension_data$fund_value), sd(pension_data$contributions), sd(pension_data$adm
```

```
minimum <-c( min(pension_data$fund_value), min(pension_data$contributions), min(pension_data$administrat

first_quartile <-c( quantile(pension_data$fund_value, 0.25), quantile(pension_data$contributions,0.25),

median <-c( median(pension_data$fund_value), median(pension_data$contributions), median(pension_data$adm

third_quartile <-c( quantile(pension_data$fund_value, 0.75), quantile(pension_data$contributions,0.75),


max <-c( max(pension_data$fund_value), max(pension_data$contributions), max(pension_data$administration_

summary_statistics <- data.frame(variables, mean, standard_deviation, minimum, first_quartile, median, 

View(summary_statistics)
```

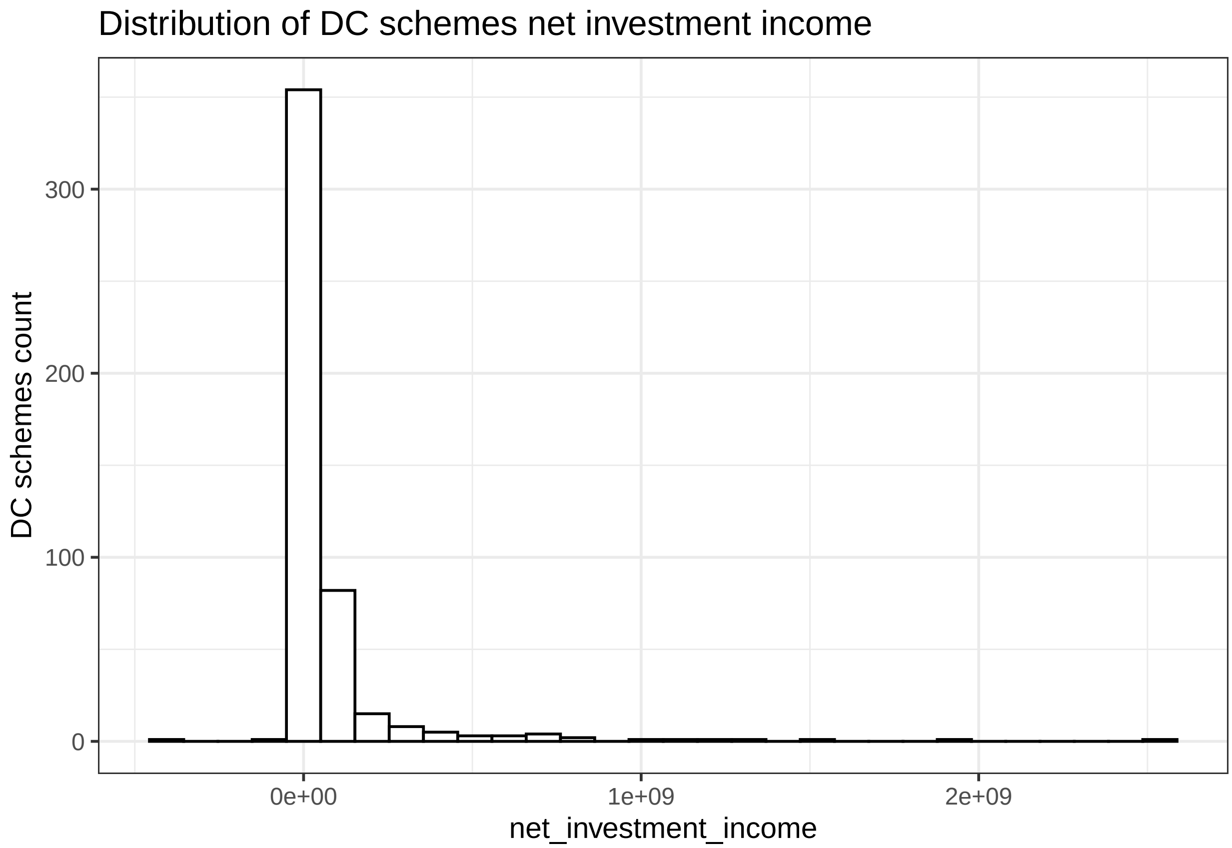
The next section will test for normality.

From the median above, it is evident that the data is positively skewed

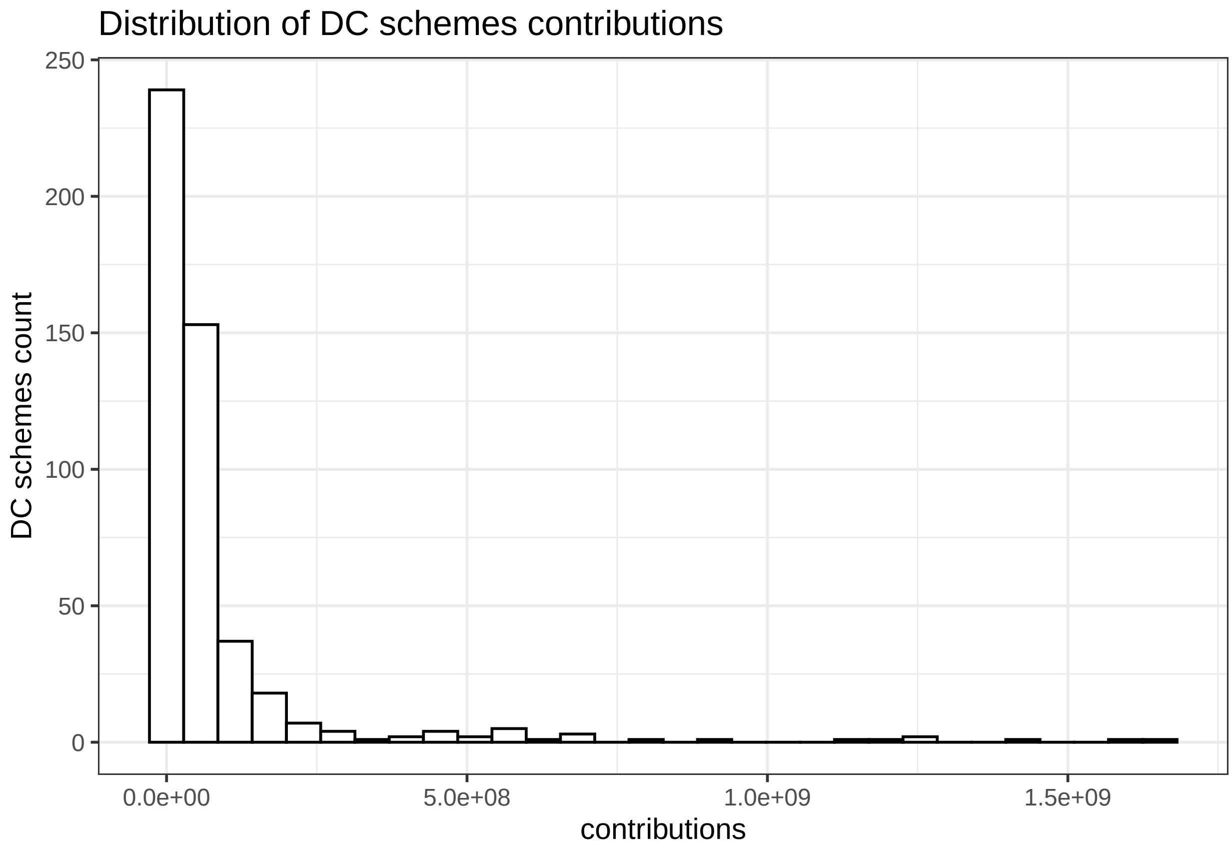
```
#Histogram

# A histogram for fund_value
ggplot2::ggplot(data = pension_data, ggplot2::aes(x = fund_value)) +
ggplot2::geom_histogram(fill = "white", color = "black", bins = 30) +
#ggplot2::scale_x_continuous(breaks = seq(0,3000000000, 5000000)) +
ggplot2::ylab("DC schemes count") +
ggplot2::theme(axis.text.x = ggplot2::element_text(angle = 90, hjust = 1)) +
ggplot2::theme_bw() +
ggplot2::labs(title = "Distribution of DC schemes fund value")
```



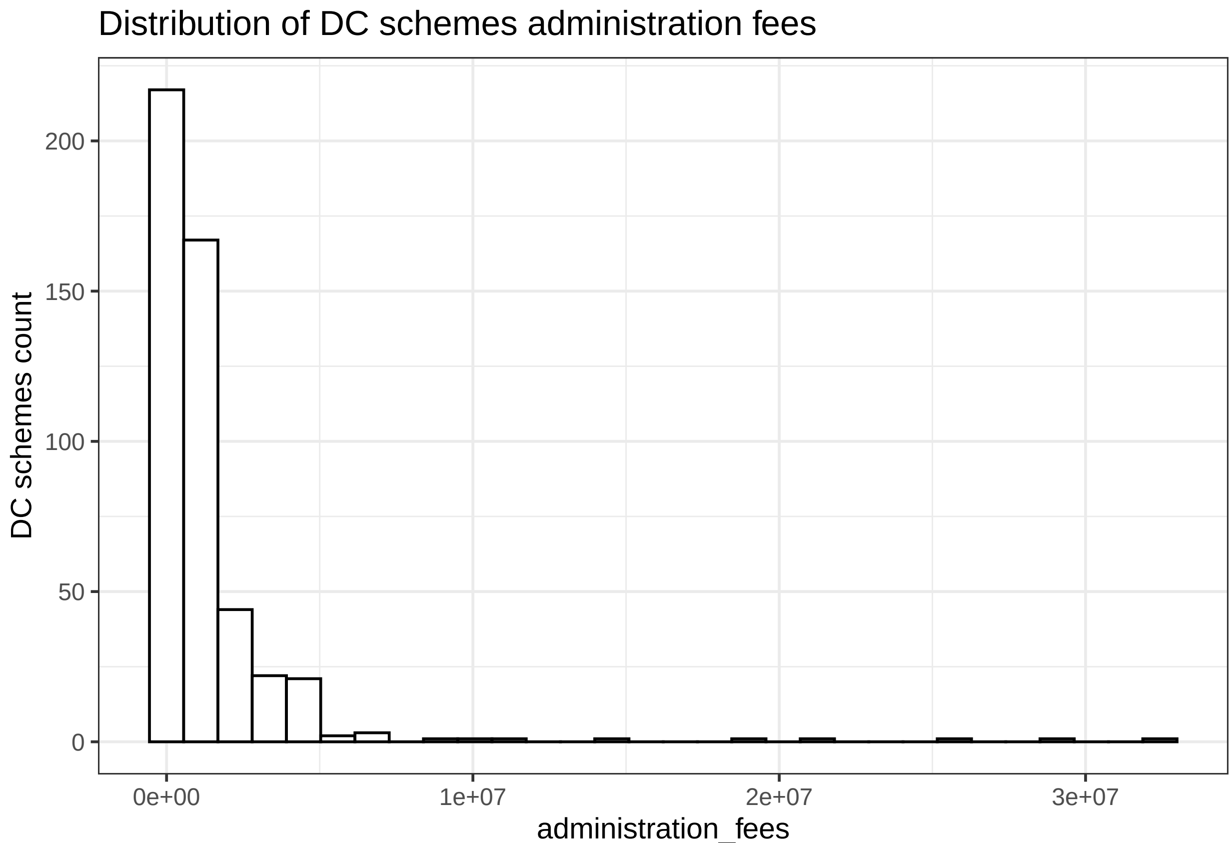

```
ggplot2::ggplot(data = pension_data, ggplot2::aes(x = net_investment_income)) +
ggplot2::geom_histogram(fill = "white", color = "black", bins = 30) +
#ggplot2::scale_x_continuous(breaks = seq(0,3000000000, 5000000)) +
ggplot2::ylab("DC schemes count") +
ggplot2::theme(axis.text.x = ggplot2::element_text(angle = 90, hjust = 1)) +
ggplot2::theme_bw() +
ggplot2::labs(title = "Distribution of DC schemes net investment income")
```



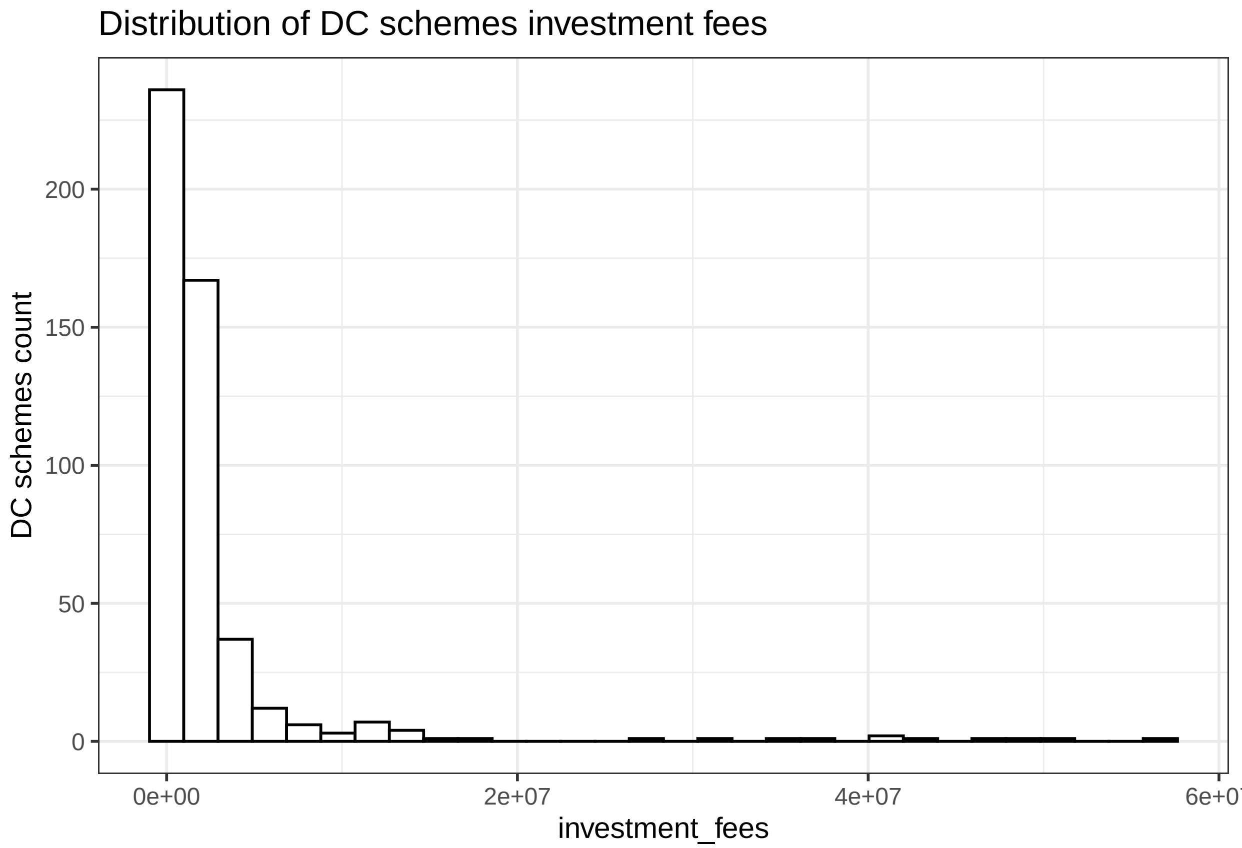

```
ggplot2::ggplot(data = pension_data, ggplot2::aes(x = contributions)) +
ggplot2::geom_histogram(fill = "white", color = "black", bins = 30) +
#ggplot2::scale_x_continuous(breaks = seq(0,3000000000, 5000000)) +
ggplot2::ylab("DC schemes count") +
ggplot2::theme(axis.text.x = ggplot2::element_text(angle = 90, hjust = 1)) +
ggplot2::theme_bw() +
ggplot2::labs(title = "Distribution of DC schemes contributions")
```



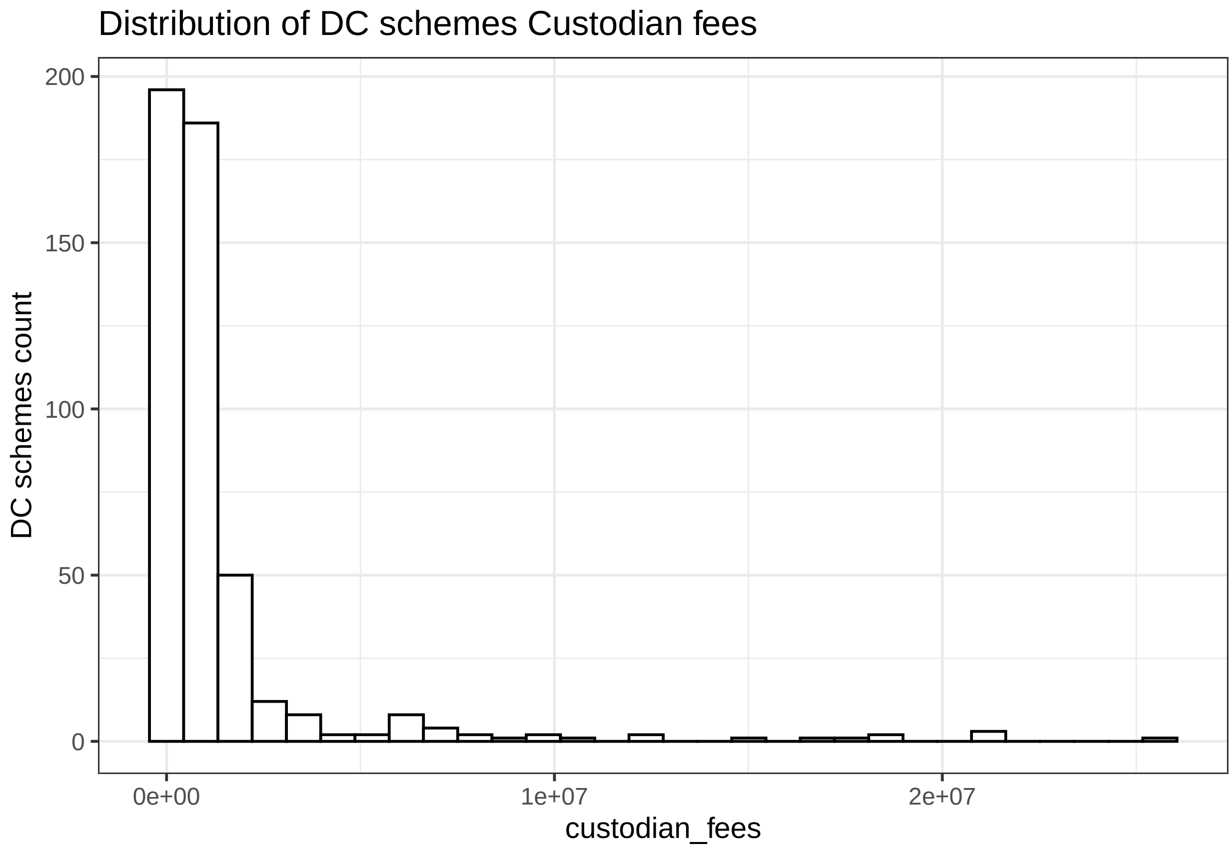

```r
ggplot2::ggplot(data = pension_data, ggplot2::aes(x = administration_fees)) +
ggplot2::geom_histogram(fill = "white", color = "black", bins = 30) +
#ggplot2::scale_x_continuous(breaks = seq(0,3000000000, 5000000)) +
ggplot2::ylab("DC schemes count") +
ggplot2::theme(axis.text.x = ggplot2::element_text(angle = 90, hjust = 1)) +
ggplot2::theme_bw() +
ggplot2::labs(title = "Distribution of DC schemes administration fees")
```




```
ggplot2::ggplot(data = pension_data, ggplot2::aes(x = investment_fees)) +
ggplot2::geom_histogram(fill = "white", color = "black", bins = 30) +
#ggplot2::scale_x_continuous(breaks = seq(0,3000000000, 5000000)) +
ggplot2::ylab("DC schemes count") +
ggplot2::theme(axis.text.x = ggplot2::element_text(angle = 90,                          hjust = 1)) +
ggplot2::theme_bw() +
ggplot2::labs(title = "Distribution of DC schemes investment fees")
```



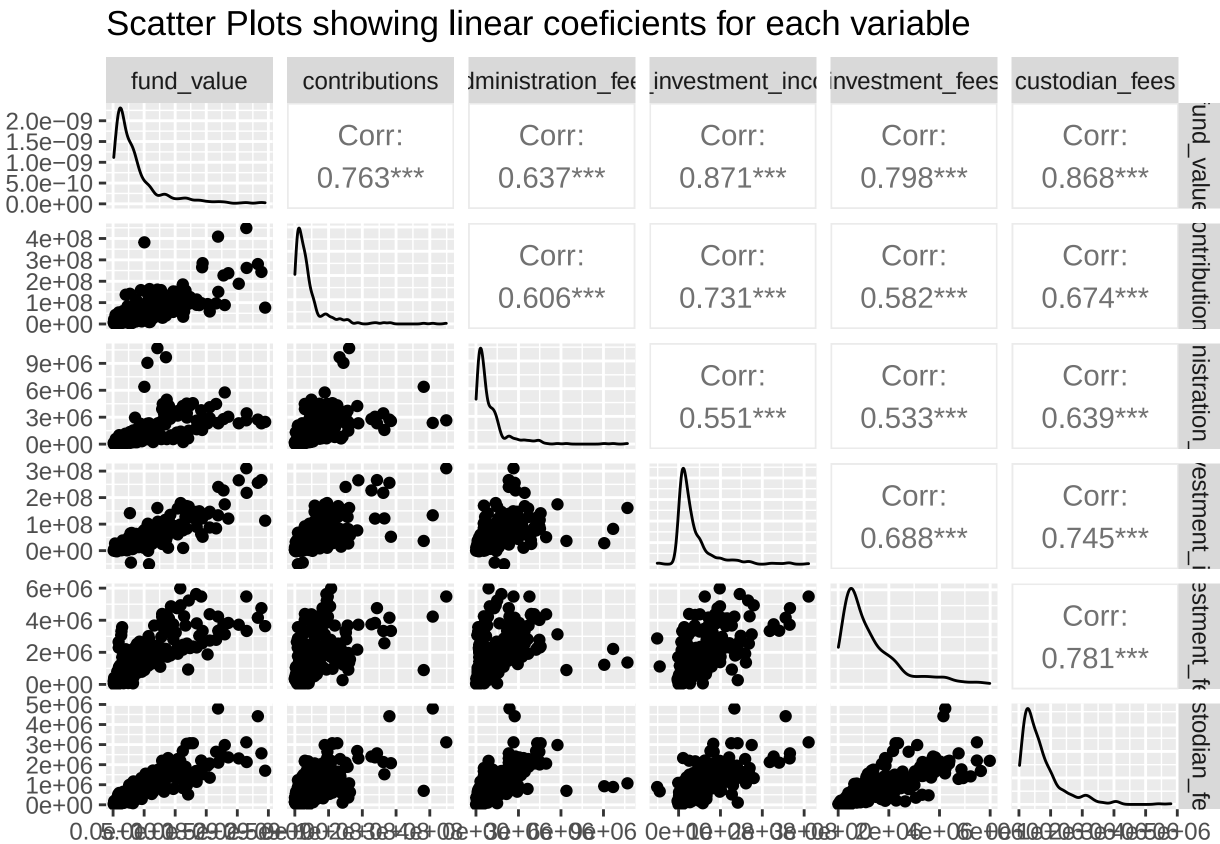

```
ggplot2::ggplot(data = pension_data, ggplot2::aes(x = custodian_fees)) +
ggplot2::geom_histogram(fill = "white", color = "black", bins = 30) +
#ggplot2::scale_x_continuous(breaks = seq(0,3000000000, 5000000)) +
ggplot2::ylab("DC schemes count") +
ggplot2::theme(axis.text.x = ggplot2::element_text(angle = 90, hjust = 1)) +
ggplot2::theme_bw() +
ggplot2::labs(title = "Distribution of DC schemes Custodian fees")
```



We fit the regression model

```
#head(pension_data)

#create variable with only the variabes we need

regression_data <- pension_data[, c("fund_value", "contributions", "administration_fees", "net_investme

regression_data<- regression_data[regression_data$fund_value<= 2500000000,]

#head(regression_data)

#create a scatter plot of each possible value

#pairs(regression_data, pch = 18, col = "steelblue")

GGally::ggpairs(regression_data, title = "Scatter Plots showing linear coeficients for each variable")
```

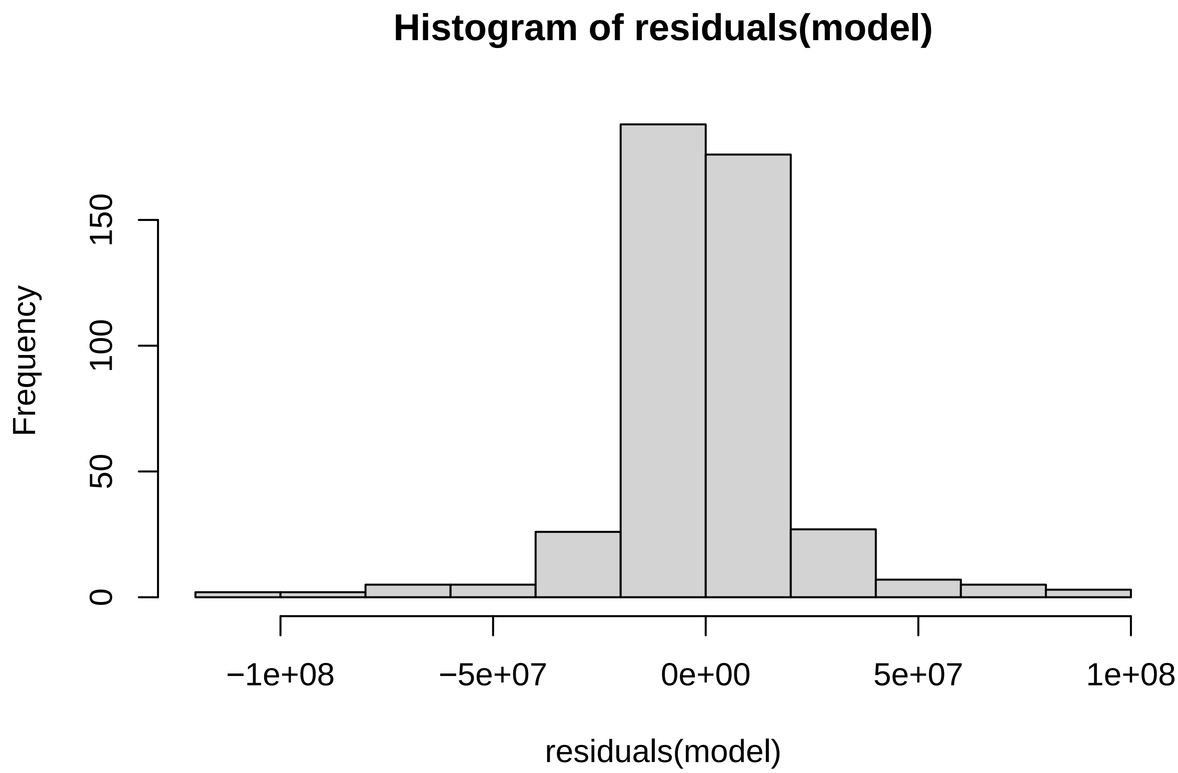
```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```



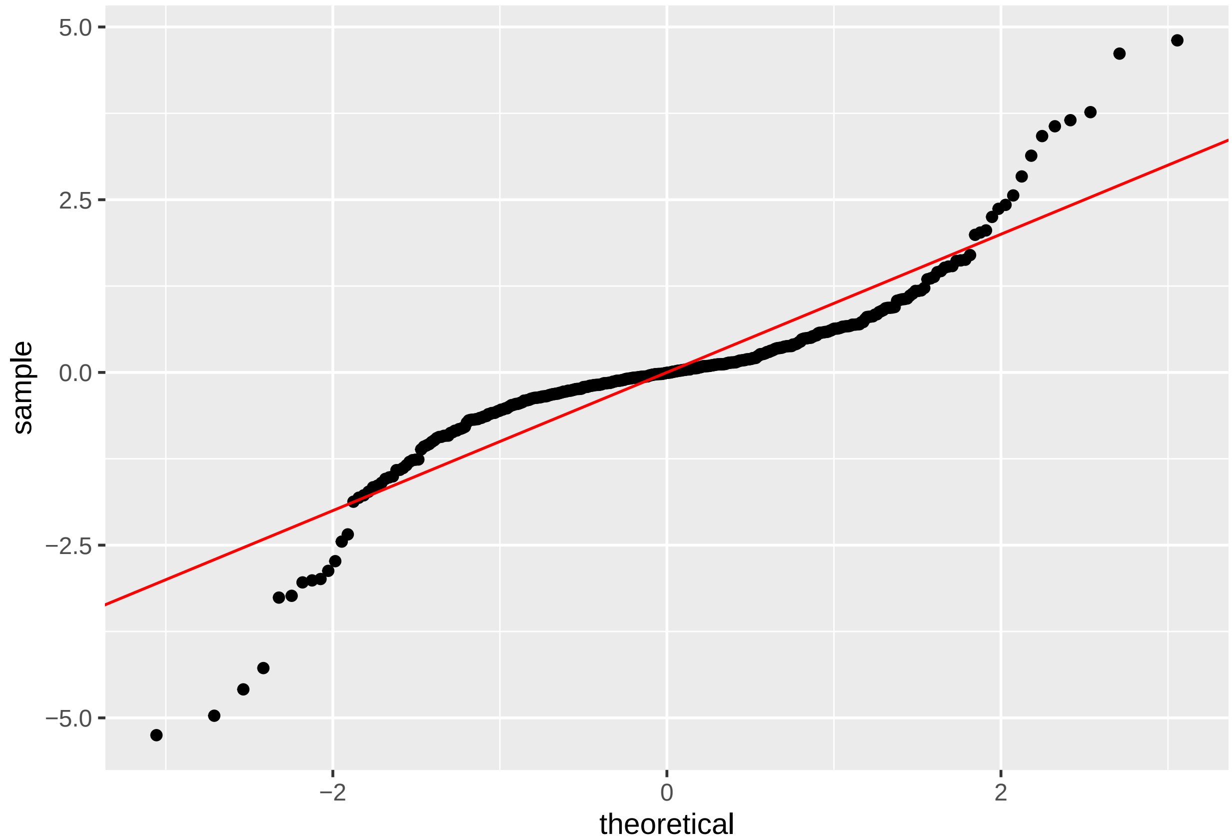

```
ggplot2::ggsave("linear_coeficients.png", width = 2606, height = 1608, units ="px")
```

Fitting the values in a linear model and testing for normality

```
# fitting our data into the model
#family = binomial(link = "logit")
model <- lm(net_investment_income ~ contributions + administration_fees + fund_value + investment_fees +

#We proceed to check for model assumptions

#The first assumption is for the normal distribution of the residuals

#To achieve this, we create a simple histogram for the residuals

hist(residuals(model))
```




```
ggplot2::ggsave("residuals_distribution.png", width = 1600, height = 800, units ="px")
#despite the model being slightly skewed to the right, the skewness is not abnormal enough to cause any

# We can also create a Q-Q plot

ggplot2::ggplot() +
  ggplot2::geom_qq(ggplot2::aes(sample = rstandard(model)))+
  ggplot2::geom_abline(color = "red")
```




```
#ggplot2::coord_fixed()
```

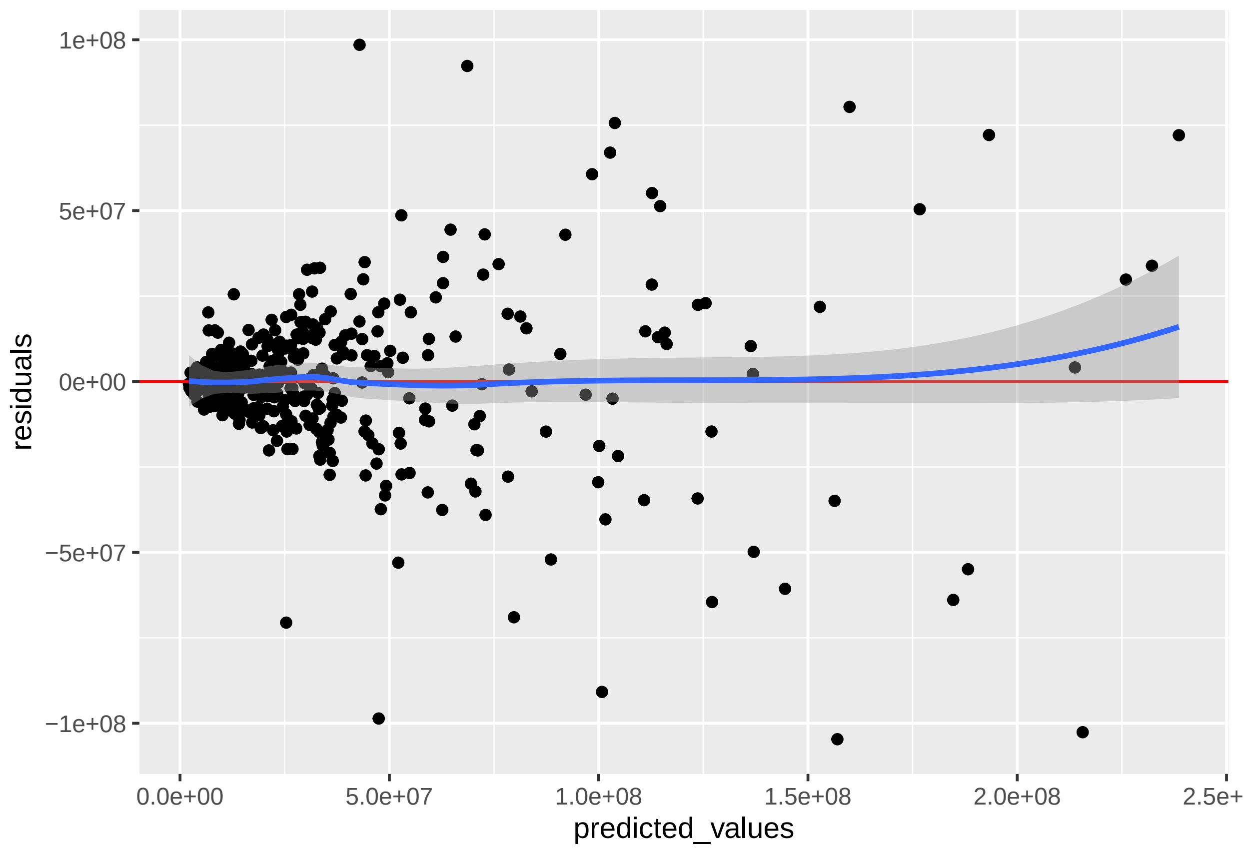
Next we test for linearity of the model

The assumption is that the residues have mean zero for every value of the fitted values and of the predictors.

This means that relevant variables and interactions are included in the model, and the functional form of the relationship between the predictors and the outcome is correct.

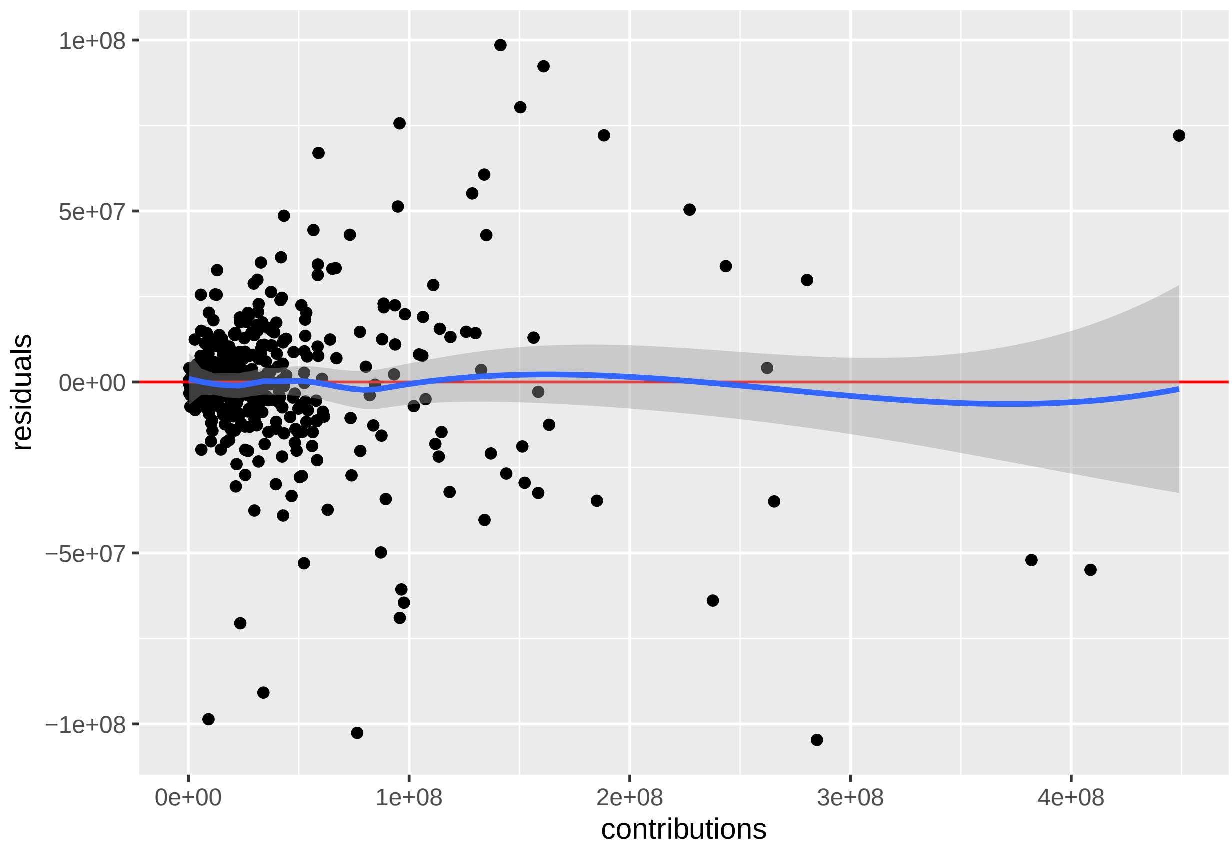
```
#we create a variables for predicted values and residuals

linearity_test <- dplyr::mutate(regression_data, predicted_values = fitted(model), residuals = residuals

ggplot2::ggplot(linearity_test, ggplot2::aes(predicted_values, residuals))+
  ggplot2::geom_point()+
  ggplot2::geom_hline(yintercept = 0, color = "red")+
  ggplot2::geom_smooth()
```

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



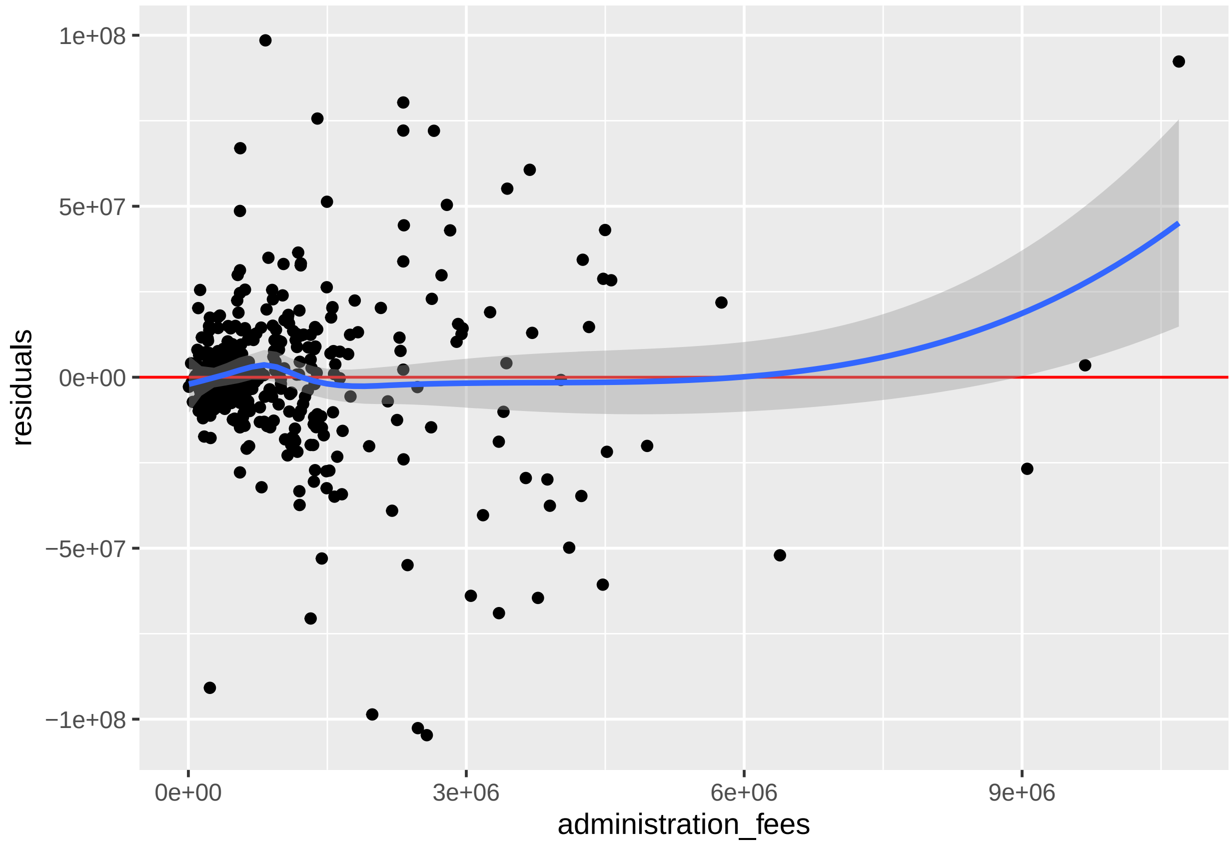

```
#check each of the variables against the residuals

ggplot2::ggplot(linearity_test, ggplot2::aes(contributions, residuals))+
  ggplot2::geom_point()+
  ggplot2::geom_hline(yintercept = 0, color = "red")+
  ggplot2::geom_smooth()
```

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



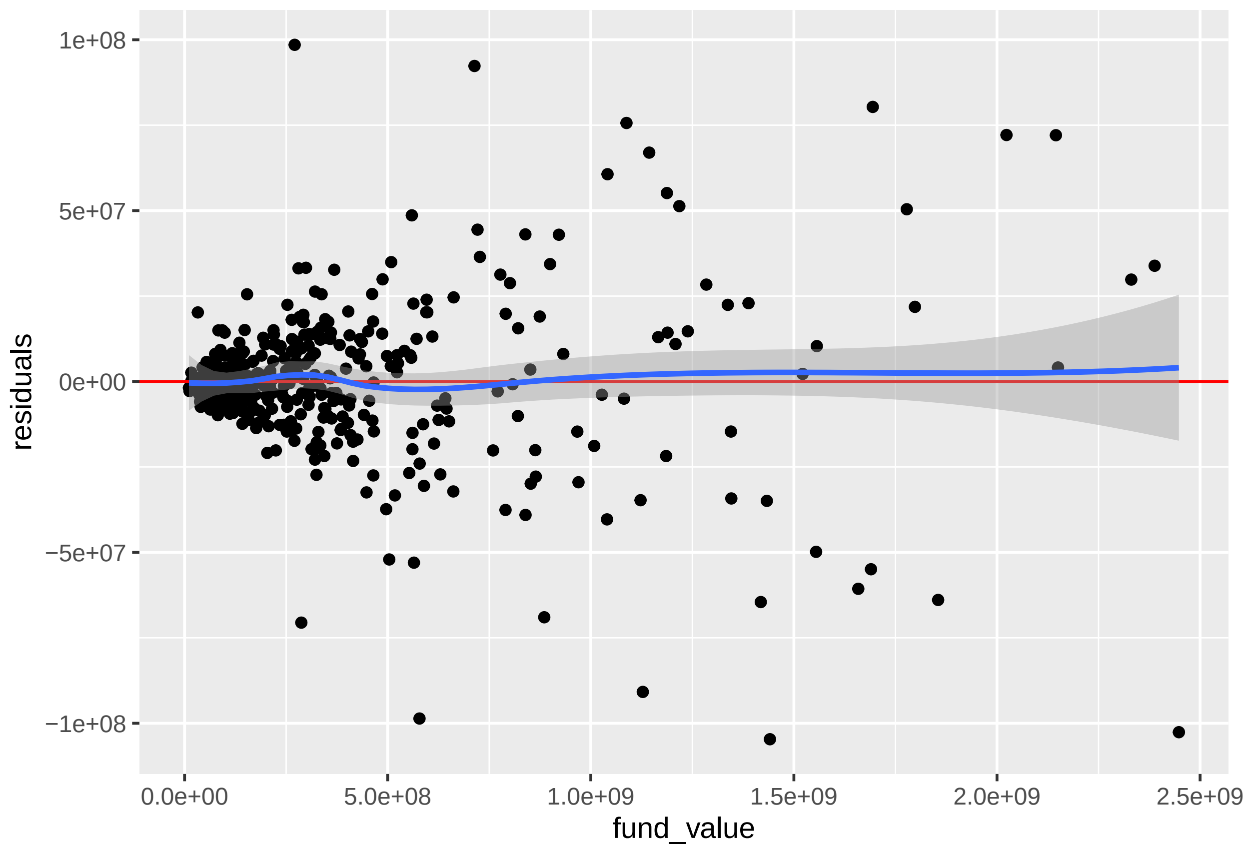

```
ggplot2::ggplot(linearity_test, ggplot2::aes(administration_fees, residuals))+
  ggplot2::geom_point()+
  ggplot2::geom_hline(yintercept = 0, color = "red")+
  ggplot2::geom_smooth()
```

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



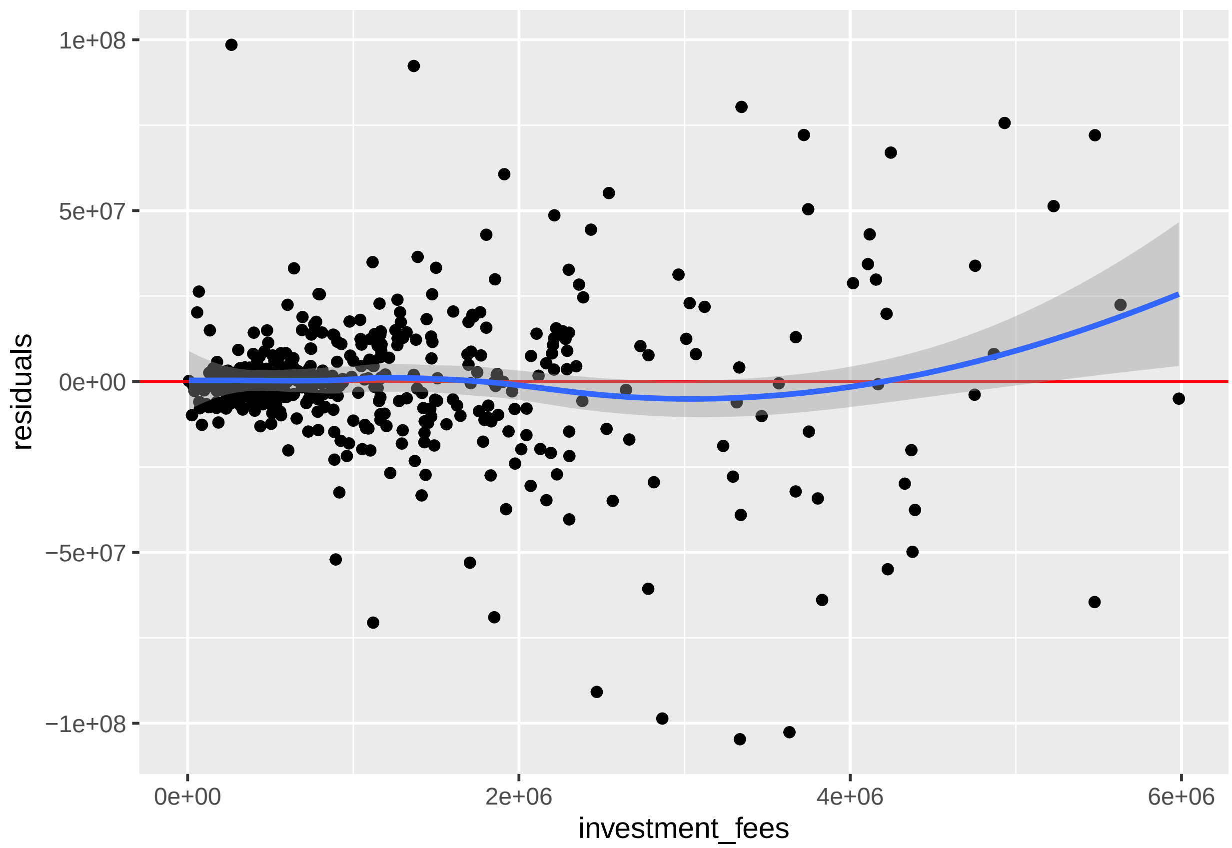

```
ggplot2::ggplot(linearity_test, ggplot2::aes(fund_value, residuals))+
  ggplot2::geom_point()+
  ggplot2::geom_hline(yintercept = 0, color = "red")+
  ggplot2::geom_smooth()
```

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



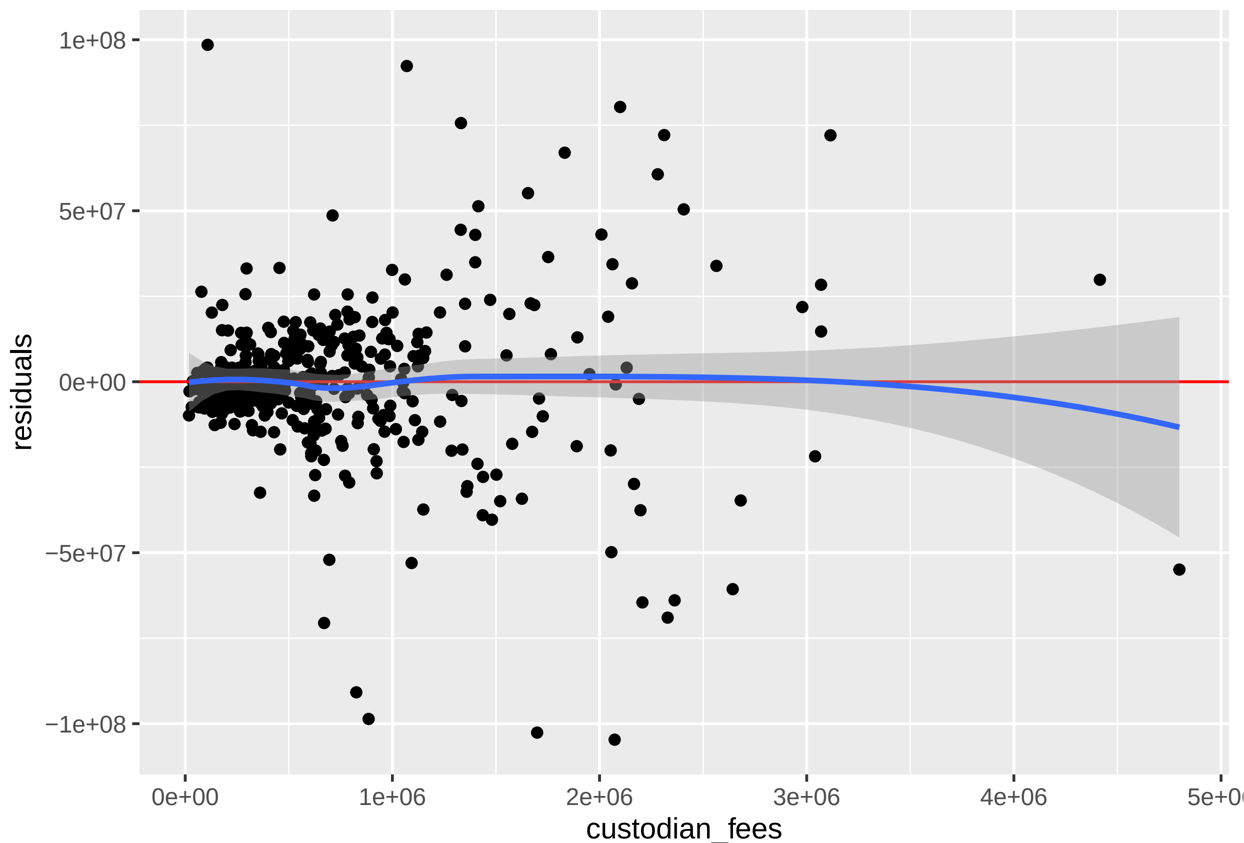

```
ggplot2::ggplot(linearity_test, ggplot2::aes(investment_fees, residuals))+
  ggplot2::geom_point()+
  ggplot2::geom_hline(yintercept = 0, color = "red")+
  ggplot2::geom_smooth()
```

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




```
ggplot2::ggplot(linearity_test, ggplot2::aes(custodian_fees, residuals))+
  ggplot2::geom_point()+
  ggplot2::geom_hline(yintercept = 0, color = "red")+
  ggplot2::geom_smooth()
```

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



Next we test for multicolinearity

```
car::vif(model)
```

```
##       contributions administration_fees          fund_value     investment_fees
##            2.555560            1.878762            6.190210            3.035224
##      custodian_fees
##            4.664067
```

Then since the model passes all the diagnostic tests we can get a summary of the model

```
summary(model)
```

```
##
## Call:
## lm(formula = net_investment_income ~ contributions + administration_fees +
##     fund_value + investment_fees + custodian_fees, data = regression_data)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -104668652   -6764151    -128257    7636054   98510249
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             6.057e+05  1.616e+06   0.375    0.708
```

```
## contributions        1.428e-01  3.029e-02   4.715 3.25e-06 ***
## administration_fees -1.283e+00  1.126e+00  -1.140    0.255    
## fund_value           8.638e-02  6.158e-03  14.027  < 2e-16 ***
## investment_fees      3.215e-01  1.582e+00   0.203    0.839    
## custodian_fees      -3.132e+00  3.395e+00  -0.923    0.357    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 21680000 on 440 degrees of freedom
## Multiple R-squared:  0.7712, Adjusted R-squared:  0.7686 
## F-statistic: 296.6 on 5 and 440 DF,  p-value: < 2.2e-16
```

From that we conduct the analysis of variance

```
anova(model)
```

```
## Analysis of Variance Table
##
## Response: net_investment_income
##                      Df     Sum Sq    Mean Sq  F value    Pr(>F)    
## contributions         1 4.8370e+17 4.8370e+17 1028.635 < 2.2e-16 ***
## administration_fees   1 1.6538e+16 1.6538e+16   35.169 6.119e-09 ***
## fund_value            1 1.9671e+17 1.9671e+17  418.326 < 2.2e-16 ***
## investment_fees       1 2.3473e+12 2.3473e+12    0.005    0.9437    
## custodian_fees        1 4.0019e+14 4.0019e+14    0.851    0.3568    
## Residuals           440 2.0690e+17 4.7024e+14                       
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```