2022 - Data Analytics for Immersive Environments - CA4 - RDBMS & Linear Regression Project

CA4 Part B - Linear Regression Analysis

Joe O'Regan

2023-01-16

Repo Link

https://github.com/joeaoregan/2022_DAIE_CA4_JOR1

Statement of Assumptions

Variables to be tested should ideally be numeric for plotting graphs etc. Average monthly hours gaming (**avg_monthly_hrs_gaming**) would have a positive effect on average monthly expenditure downloadable content (DLC) (**avg_monthly_expenditure_dlc**).

The more hours a player plays games, the more inclined they would be to spend money on DLC. As the number of hours grows the expenditure should also grow.

I would assume there is linearity between the chosen variables and constant variance.

Testing of Assumptions

Assumptions for Linear Regression

1. Independence of observation
2. Normality
3. Linearity
4. Homoscedasticity

Independence of observation (No autocorrellation)

No need to test for hidden relationships between variables when there is only one independent and one dependent variable. Find the R value or correlation between variables using cor().The variables age and avg_years_playing_games don't have floating values so are more likely to repeat.

Normality (Histograms, Shapiro-Wilk Significance Test

Visually inspect normality with histograms. If the histogram is symmetrical/unimodal, then the data is assumed to be normally distributed.

Shapiro-Wilk Significance test. Visual inspection isn't always reliable. Widely recommended for normality test and more powerful than Kolmogorov-Smirnov (K-S) nomality test.

Need to combine visual inspection and significance test to get good results, as normality test can be sensitive to sample size. Small samples can pass normality tests.

Linearity

Any relationship between the independent and dependent variable is linear: the line of best fit through the data points is a straight line and not a curve or grouping factor.

The statistical method for fitting a line to data where the relationship between two variables, x and y, can be modeled by a straight line with some error (M., D., D., C. and Çetinkaya-Rundel, M., 2019):

$Y = \beta 0 + \beta 1 x + \epsilon$

$\beta 0$: intercept, predicted value of y when x is 0
$\beta 1$: regression coefficient - how much y changes as x increases
ϵ: Error of estimate. How much variation exists in estimate of regression coefficient
x: Explanatory variable (independent), influences y
y: Response variable (dependent)

Homoscedasticity

Homogeneity of variance. The size of the error in our prediction doesn't change significantly across the values of the independent variable.

Transforming the data had no impact with the p-value remaining low for all 3 transformations, log, square root and cube root.

Analysis conducted and results obtained

Correlation (R Value)

Correlation between avg_monthly_hrs_gaming and avg_monthly_expenditure_dlc is smallest. There is no apparent linear relationship between the variables.

Correlation between age and avg_yers_playing_games is largest but it is still not close to 1 or -1.

Normality

Visual: Inspecting the histograms, data is not normally distributed for both variables. For avg_monthly_hrs_gaming the histogram is skewed to the right. The histogram for avg_monthly_expenditure_dlc is roughly bell-shaped, but the number of breaks increases it appears multimodal.

Significance: Null hypothesis for Shapiro-Wilk's normality test rejected for all variables before sampling. The p-value is less than 0.05 and the distribution of the data is significantly different from normal distribution.

Linearity

After checking data meets assumptions, check the relationship between independent and dependent variables using linear regression.

Box plot: Outliers in the prediction can negatively affect predictions as they may affect the direction/slope of the best fit line.

Homoscedasticity

Plot the linear model results to check whether the observed data meets our model assumptions.

Normal Q-Qplot doesn't create a perfect one-to-one line with the theoretical residuals.

The red lines representing the mean of the residuals are not entirely horizontal.

Plot

From the scatterplot the variables appear to have a weak relationship, and trying a linear fit would be reasonable.

Line

When the least squares line is added there is a very weak downward trend in the data.

R and R squared

Coefficient of determination (R2) tests how good the model is. The total variability explained by the regression model. Low r squared value means less variability is explained by the model.

High R squared isn't necessary in every situation.

Residuals

Residuals appear to be still random when plotting the linear model residuals.

Data transformation might be an option.

Boxplots

No substantial outliers detected.

R Code

Load and Randomly Sample Data

```r
# Load and Randomly Sample Data
# use readr::read_csv() to load data from csv file

data <- read_csv("amalgamated_game_survey_250_2022.csv") # read data from csv
```

```
## Rows: 250 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): gender, top_reason_gaming, gaming_platform, favourite_game, ethnici...
## dbl (4): age, avg_monthly_hrs_gaming, avg_years_playing_games, avg_monthly_e...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
summary(data) # Check data has been read in correctly
```

```
##     gender               age        top_reason_gaming  gaming_platform
##  Length:250         Min.   :20.00   Length:250         Length:250
##  Class :character   1st Qu.:22.00   Class :character   Class :character
##  Mode  :character   Median :23.00   Mode  :character   Mode  :character
##                     Mean   :23.16
##                     3rd Qu.:24.00
##                     Max.   :33.00
##  favourite_game     avg_monthly_hrs_gaming avg_years_playing_games
##  Length:250         Min.   : 8.70          Min.   : 6.00
##  Class :character   1st Qu.:17.23          1st Qu.:10.00
##  Mode  :character   Median :19.80          Median :12.00
##                     Mean   :19.98          Mean   :11.78
##                     3rd Qu.:22.80          3rd Qu.:14.00
##                     Max.   :27.20          Max.   :17.00
##  avg_monthly_expenditure_dlc  ethnicity          play_roblox
##  Min.   :38.93               Length:250         Length:250
##  1st Qu.:47.47               Class :character   Class :character
##  Median :55.28               Mode  :character   Mode  :character
##  Mean   :55.48
##  3rd Qu.:63.48
##  Max.   :72.78
##   use_steam
##  Length:250
##  Class :character
##  Mode  :character
##
##
##
```

```r
# set a seed to reproduce random values
set.seed(1234)

# randomly sample 200 of the 250 rows
sample_data <- sample_n(data, 200) # returns tibble 200 x 11
```

Calculate Linear Regression for Data

1. Independence of observation

Correlation / R Value

```r
# check the correlation between the chosen variables
cor(sample_data$avg_monthly_hrs_gaming, sample_data$avg_monthly_expenditure_dlc)
```

```
## [1] -0.01702819
```

```r
# perform correlation test on chosen variables
cor.test(sample_data$avg_monthly_hrs_gaming, sample_data$avg_monthly_expenditure_dlc)
```

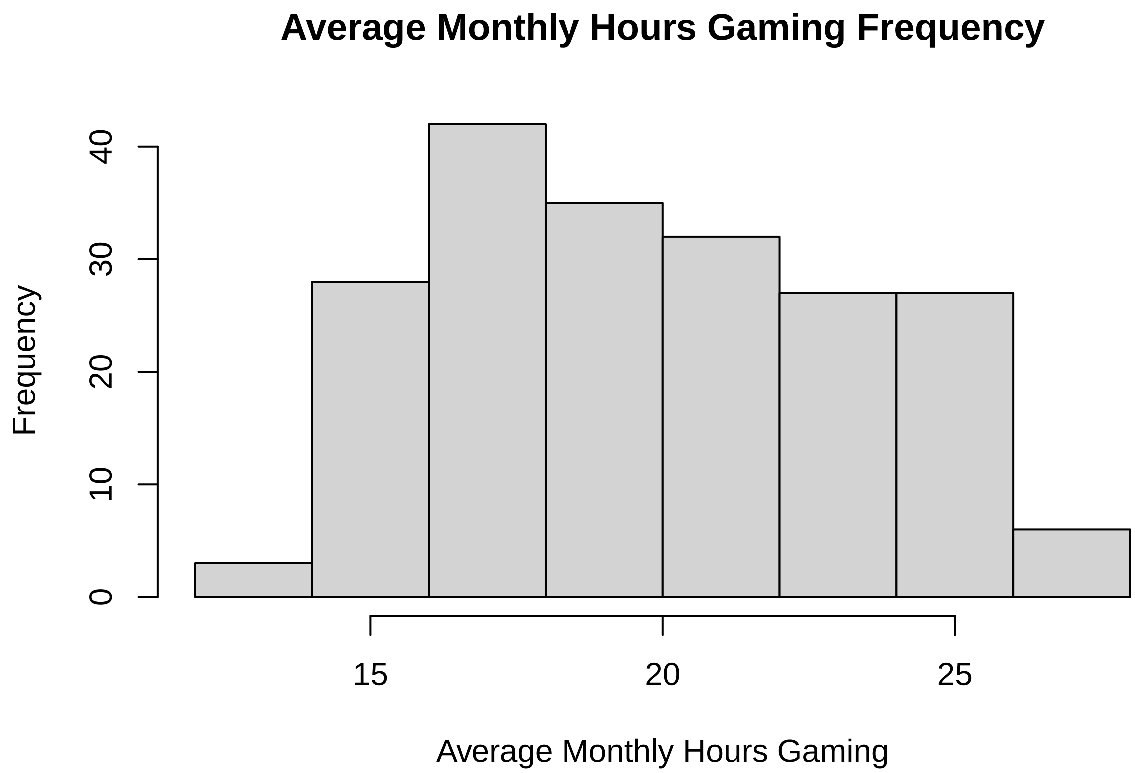
```
##
##  Pearson's product-moment correlation
##
## data:  sample_data$avg_monthly_hrs_gaming and sample_data$avg_monthly_expenditure_dlc
## t = -0.23964, df = 198, p-value = 0.8109
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1554021  0.1220011
## sample estimates:
##         cor
## -0.01702819
```

2. Normality

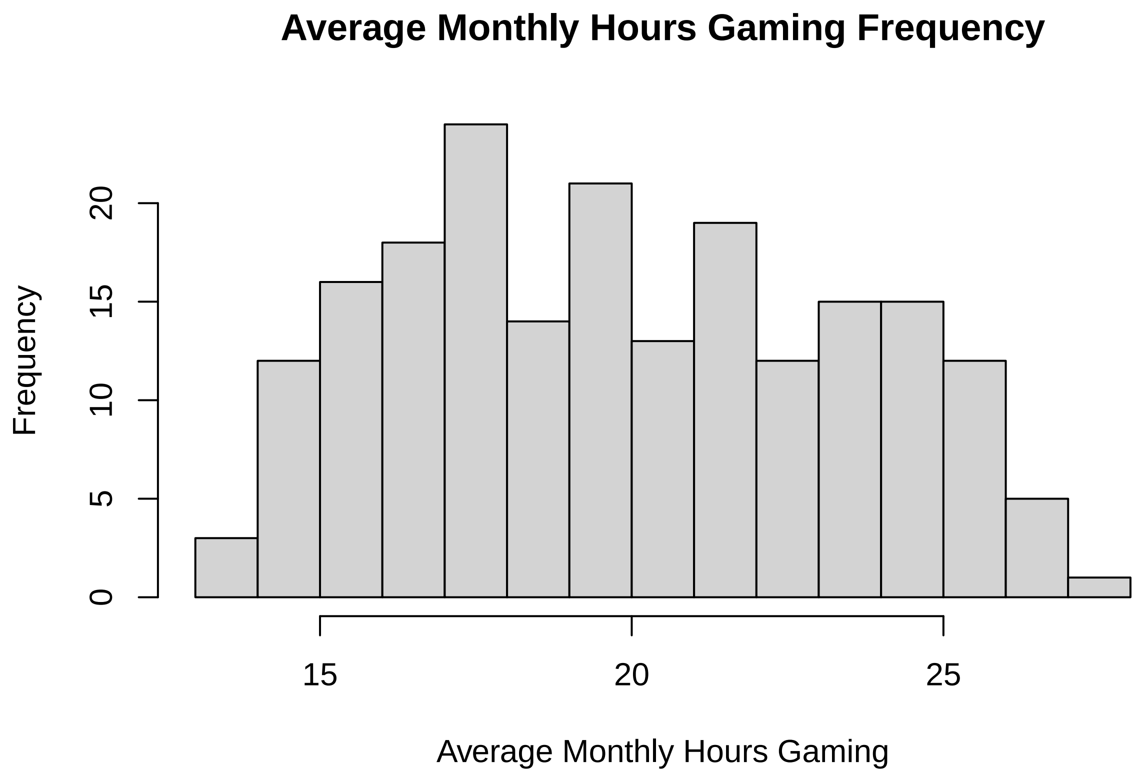
Histograms

Check data visually with histograms.

```r
hist(sample_data$avg_monthly_hrs_gaming,
     main="Average Monthly Hours Gaming Frequency",
     xlab="Average Monthly Hours Gaming")
```

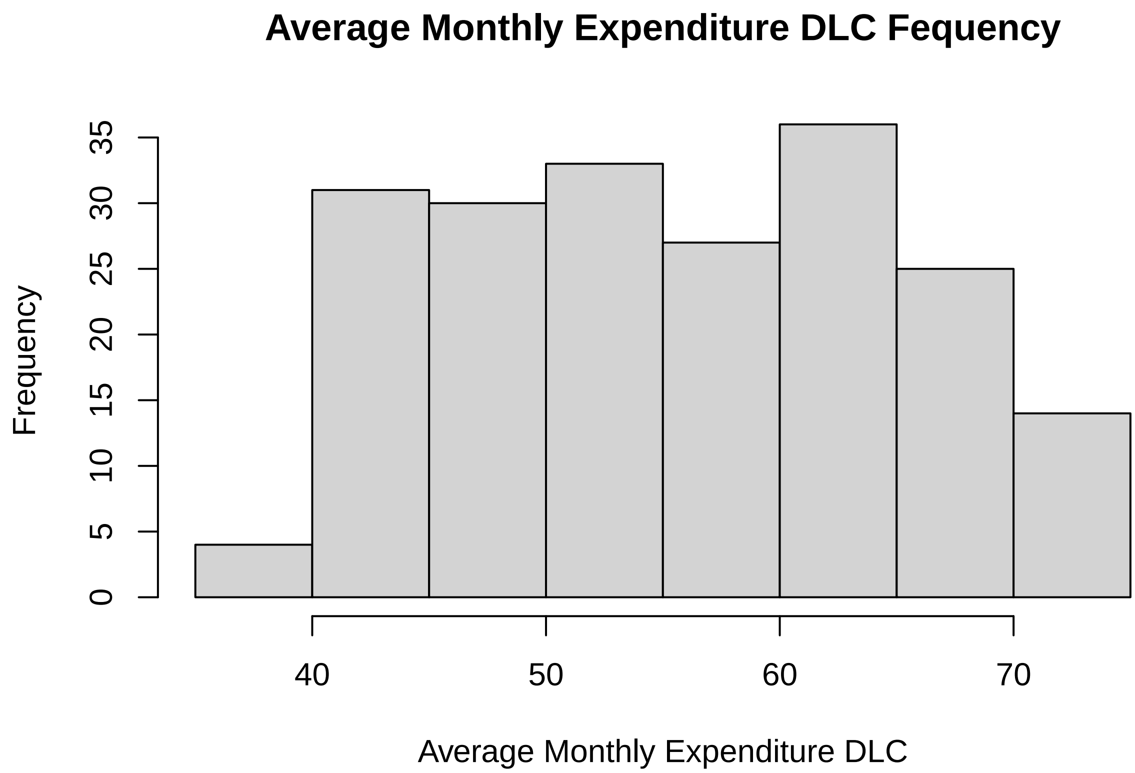



```
hist(sample_data$avg_monthly_hrs_gaming,
     main="Average Monthly Hours Gaming Frequency",
     xlab="Average Monthly Hours Gaming",
     breaks=12)
```

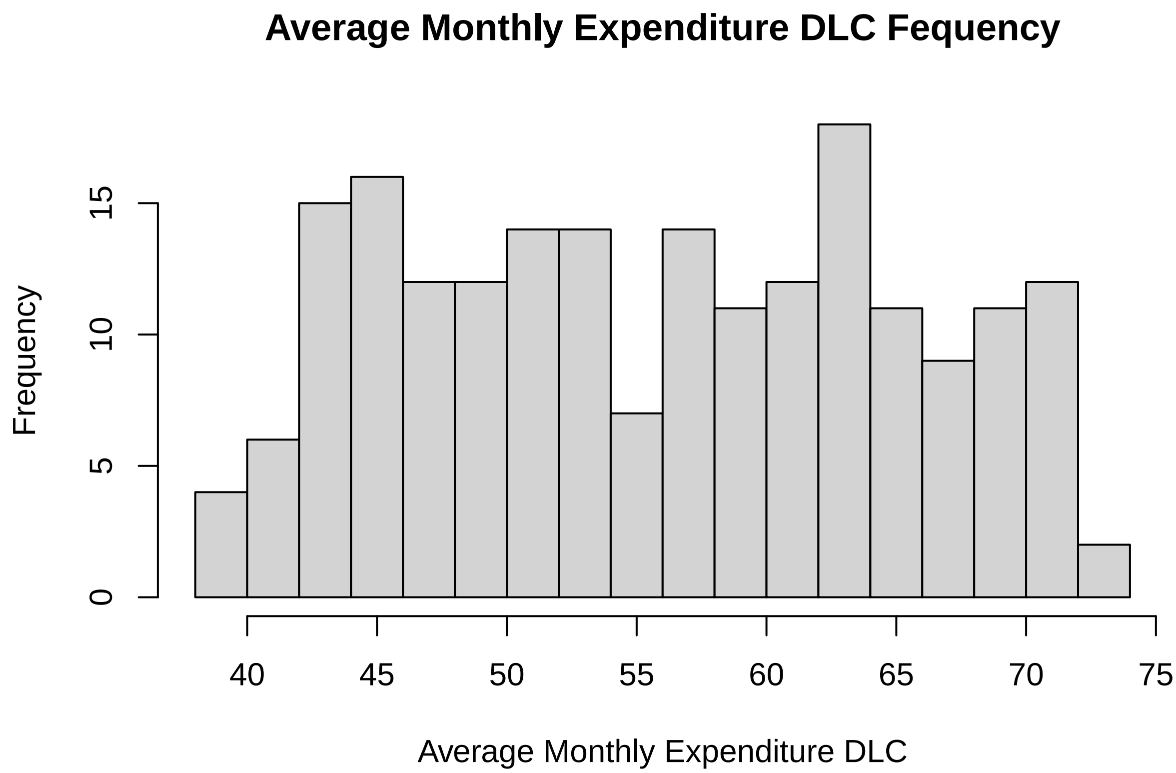


Average Monthly Hours Gaming histogram skewed to the left slightly.

```
hist(sample_data$avg_monthly_expenditure_dlc,
     main="Average Monthly Expenditure DLC Fequency",
     xlab = "Average Monthly Expenditure DLC")
```




```
hist(sample_data$avg_monthly_expenditure_dlc,
     main="Average Monthly Expenditure DLC Fequency",
     xlab = "Average Monthly Expenditure DLC",
     breaks=12)
```



Roughly bell-shaped. Increasing the breaks makes it appear multimodal.

Shapiro-Wilk's Method (Significance test)

null hypothesis: the data are sampled from a Gaussian distribution.

```
# Shapiro-Wilk's method for normality test

# If the P value is greater than 0.05 accept null hypothesis
# If the P value is less than or equal to 0.05 reject null hypothesis

significance <- 0.05

# perform shapiro test on avg_monthly_hrs_gaming
st_hours <- shapiro.test(sample_data$avg_monthly_hrs_gaming)

# if shapiro test result is too low reject the null hypothesis
if(st_hours$p.value < significance) {
  print("reject") } else {
  print("accept")
}
```

```
## [1] "reject"
```

```
# perform shapiro test on avg_monthly_expenditure_dlc
st_bucks <- shapiro.test(sample_data$avg_monthly_expenditure_dlc)
```

```r
# use ifelse() to perform similar check as above
print(ifelse(st_bucks$p.value < significance, "reject", "accept"))
```

```
## [1] "reject"
```

Null hypothesis rejected for all variables before sampling.

Data is not normally distributed for either variable.

3. Linear Regression Analysis

```r
# simple linear regression

# calculate effect of independent on dependent variable
mod <- lm(avg_monthly_expenditure_dlc ~ avg_monthly_hrs_gaming,
          data = sample_data)

# summarise the results of the model
summary(mod)
```

```
##
## Call:
## lm(formula = avg_monthly_expenditure_dlc ~ avg_monthly_hrs_gaming,
##     data = sample_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.2939  -8.2364   0.2041   7.8369  17.2410
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)            56.61685    3.83003   14.78   <2e-16 ***
## avg_monthly_hrs_gaming -0.04529    0.18899   -0.24    0.811
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.321 on 198 degrees of freedom
## Multiple R-squared:  0.00029,    Adjusted R-squared:  -0.004759
## F-statistic: 0.05743 on 1 and 198 DF,  p-value: 0.8109
```

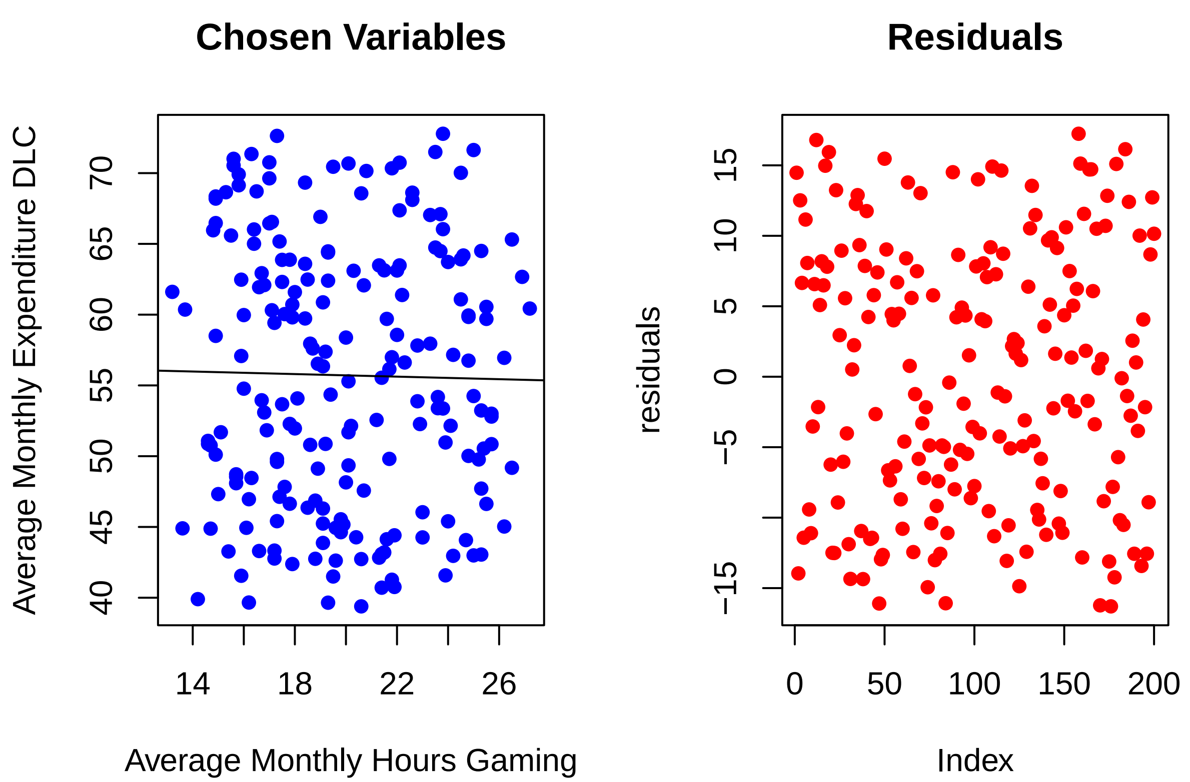
Not a Significant positive relationship between avg_monthly_hrs_gaming and avg_monthly_expenditure_dlc (p value > 0.05)

Equation for least-squares regression line: avg_monthly_expenditure_dlc = 56.61685 -0.04529*avg_monthly_hrs_gaming (When seed is set 1234 above set.seed(1234))

```r
#plot(sample_data$avg_monthly_hrs_gaming, mod$residuals)

df <- data.frame(sample_data$avg_monthly_hrs_gaming, sample_data$avg_monthly_expenditure_dlc)

par(mfrow=c(1,2), main="test") # 2 rows and 2 columns
```

```
## Warning in par(mfrow = c(1, 2), main = "test"): "main" is not a graphical
## parameter
```

```
plot(df, pch=16, col="blue",
     xlab="Average Monthly Hours Gaming",
     ylab="Average Monthly Expenditure DLC",
     main="Chosen Variables")
abline(mod)
plot(mod$residuals, pch=16, col="red", ylab="residuals", main="Residuals")
```




```
par(mfrow=c(1,1)) # Reset to 1 row and 1 column
```

```
# r squared value, percent of variation
r_squared <- summary(mod)$r.squared
r_squared
```

```
## [1] 0.0002899591
```

```
# r value, derived from r squared
sqrt(r_squared)
```

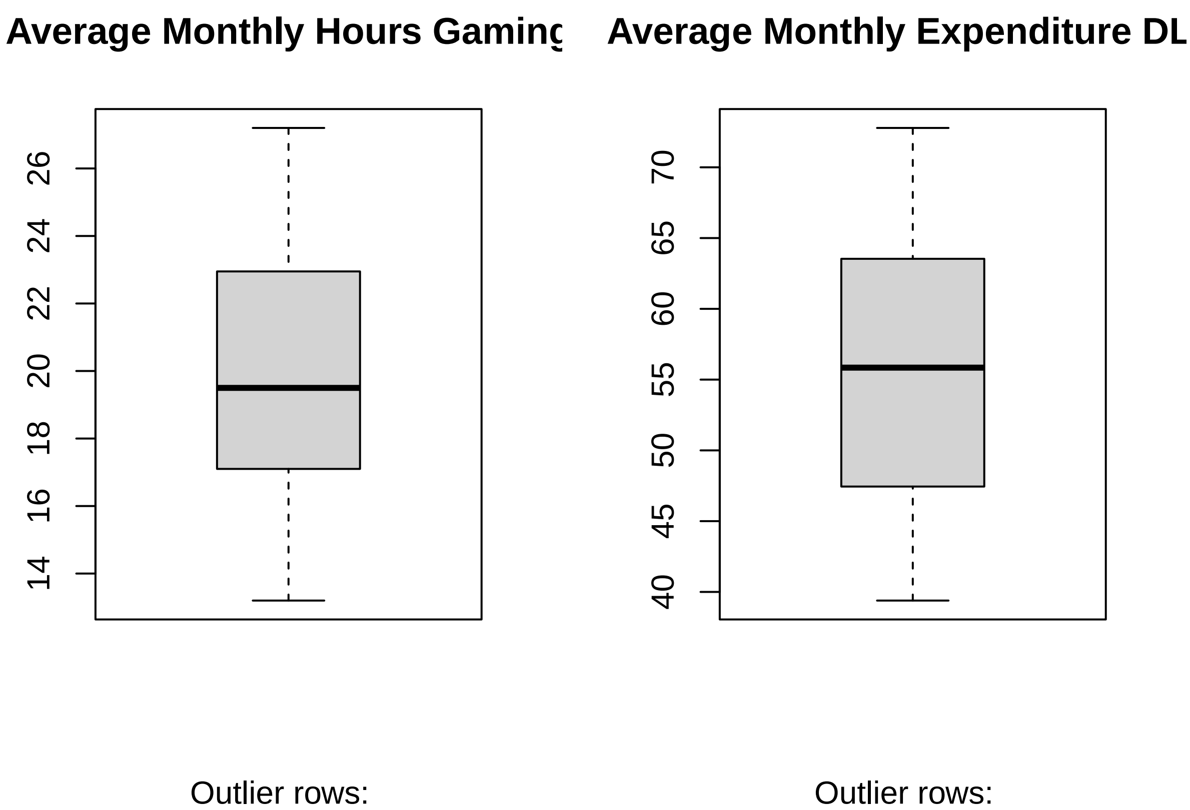
```
## [1] 0.01702819
```

```r
# r value, using correlation function
cor(sample_data$avg_monthly_hrs_gaming, sample_data$avg_monthly_expenditure_dlc)
```

```
## [1] -0.01702819
```

Check for outliers

```r
par(mfrow=c(1, 2))  # divide graph area in 2 columns

boxplot(sample_data$avg_monthly_hrs_gaming,
        main="Average Monthly Hours Gaming",
        sub=paste("Outlier rows: ",
         boxplot.stats(sample_data$avg_monthly_hrs_gaming)$out))

boxplot(sample_data$avg_monthly_expenditure_dlc,
        main="Average Monthly Expenditure DLC",
        sub=paste("Outlier rows: ",
         boxplot.stats(sample_data$avg_monthly_expenditure_dlc)$out))
```



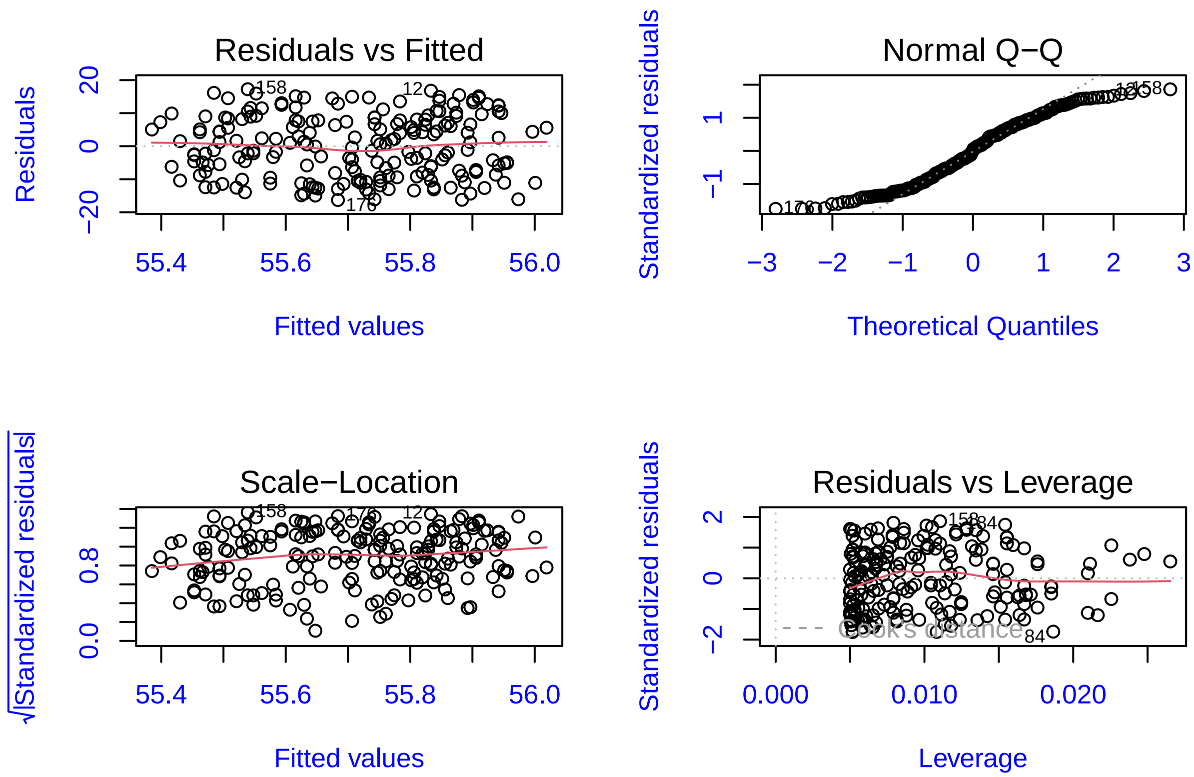
no outliers detected

4. Check for homoscedasticity

```r
par(mfrow=c(2,2), main="test") # 2 rows and 2 columns
```

```
## Warning in par(mfrow = c(2, 2), main = "test"): "main" is not a graphical
## parameter
```

```r
plot(mod, col.lab="blue", col.axis="blue") # plot the model
```




```r
par(mfrow=c(1,1)) # Reset to 1 row and 1 column
```

Normal Q-Qplot doesn't a perfect one-to-one line with the theoretical residuals.

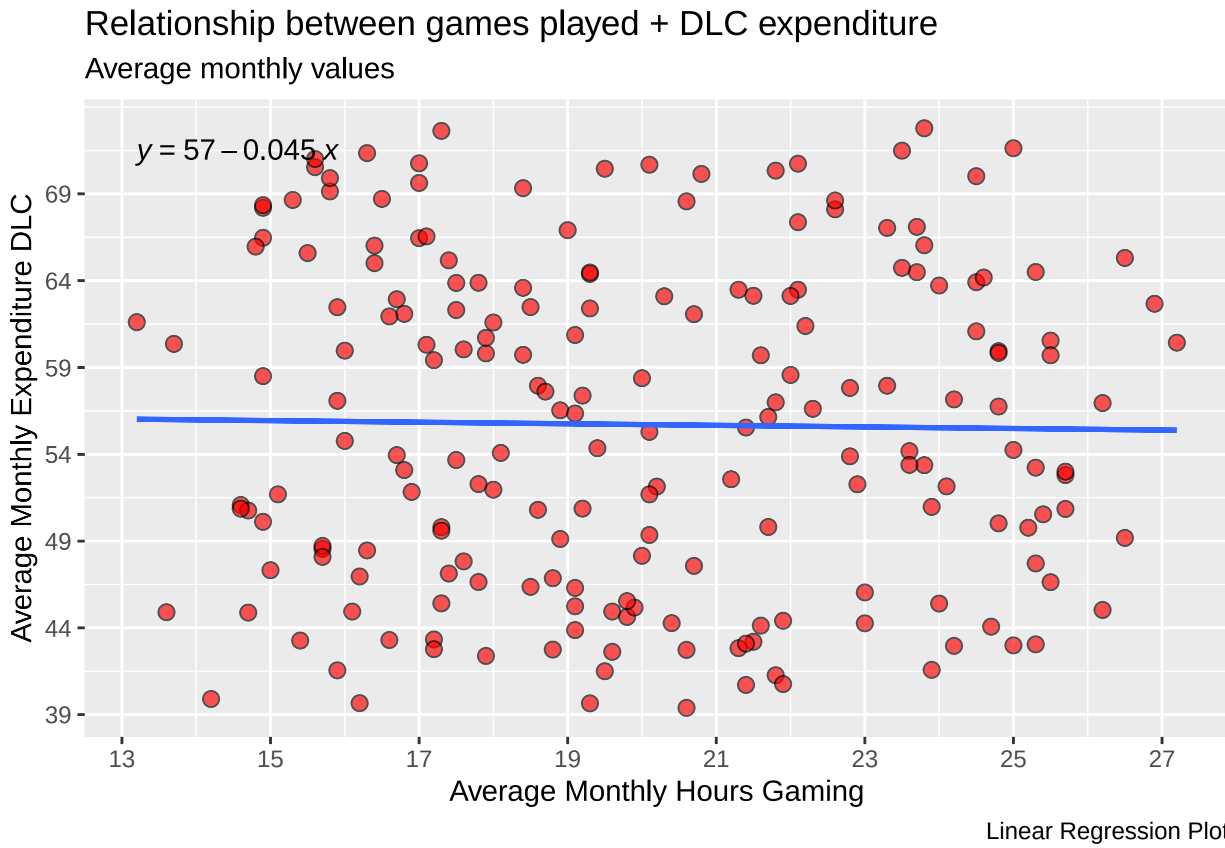
Linear Regression Plot(s)

The plot is created using the linear model data to map the avg_monthly_hrs_gaming and avg_montly_expenditure_dlc variables as points in the plot.

The scale of the x and y axiis are set using the rounded down min value for the variable and the max value rounded up. With just rounded values they were showing with a decimal place and didn't look right.

```
# plot dataset in a scatter plot, add colours for points
plot <- ggplot(data = mod, mapping = aes(x = avg_monthly_hrs_gaming,
                                     y = avg_monthly_expenditure_dlc)) +
  geom_point(alpha = 0.66, # transparcency, lets stacked points show darker
             shape=21,# round
             fill="red", # inner colour
             color="black", # outline colour
             size=2.5) + # size (3 too big, 1 too small) +

  labs(title = "Relationship between games played + DLC expenditure",
       subtitle = "Average monthly values",
       caption = "Linear Regression Plot")

# Calculate x and y tick spacing and frequency
scale_x = scale_x_continuous(breaks = seq(
  floor(min(sample_data$avg_monthly_hrs_gaming)), # round down lowest value
  ceiling(max(sample_data$avg_monthly_hrs_gaming)), # round up highest value
  by = 2), # frequency
  name = "Average Monthly Hours Gaming") # x label
scale_y = scale_y_continuous(breaks = seq(
  floor(min(sample_data$avg_monthly_expenditure_dlc)),
  ceiling(max(sample_data$avg_monthly_expenditure_dlc)),
  by = 5),
  name = "Average Monthly Expenditure DLC")

# Get intercept and slope for regresion line
coeff <- coefficients(mod) # get coefficients returned from linear model
intercept <- coeff[1] # avg_monthly_hrs_gaming intercept
slope <- coeff[[2]] # slope of line, double square brackets = just the number

# Add x and y labels and geometry line to plot
plot + scale_x + scale_y +
  # geom_abline(intercept = intercept, slope = slope, color="red") + # regression line
  #stat_smooth(method = "lm", formula = y ~ x, geom = "smooth")
  geom_smooth(method="lm", se=F) +
  stat_regline_equation() # add equation to regression line
```

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


Relationship between games played + DLC expenditure
Average monthly values
y = 57 – 0.045 x
Average Monthly Expenditure DLC
39
44
49
54
59
64
69
13
15
17
19
21
23
25
27
Average Monthly Hours Gaming
Linear Regression Plot

Appendices

Transform Data

Perform transformation: log, square root, or cube root. To see can data become more normally distributed.

```
log_hours <- log10(sample_data$avg_monthly_hrs_gaming)
log_bucks <- log10(sample_data$avg_monthly_expenditure_dlc)

df_log <- data.frame(log_hours, log_bucks)
mod_log <- lm(log_bucks ~ log_hours, data=df_log)
summary(mod_log)
```

```
##
## Call:
## lm(formula = log_bucks ~ log_hours, data = df_log)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.14413 -0.06323  0.00774  0.06338  0.12339
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.75827    0.08854  19.859   <2e-16 ***
## log_hours   -0.01427    0.06833  -0.209    0.835
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07388 on 198 degrees of freedom
## Multiple R-squared:  0.0002204,  Adjusted R-squared:  -0.004829
## F-statistic: 0.04365 on 1 and 198 DF,  p-value: 0.8347
```

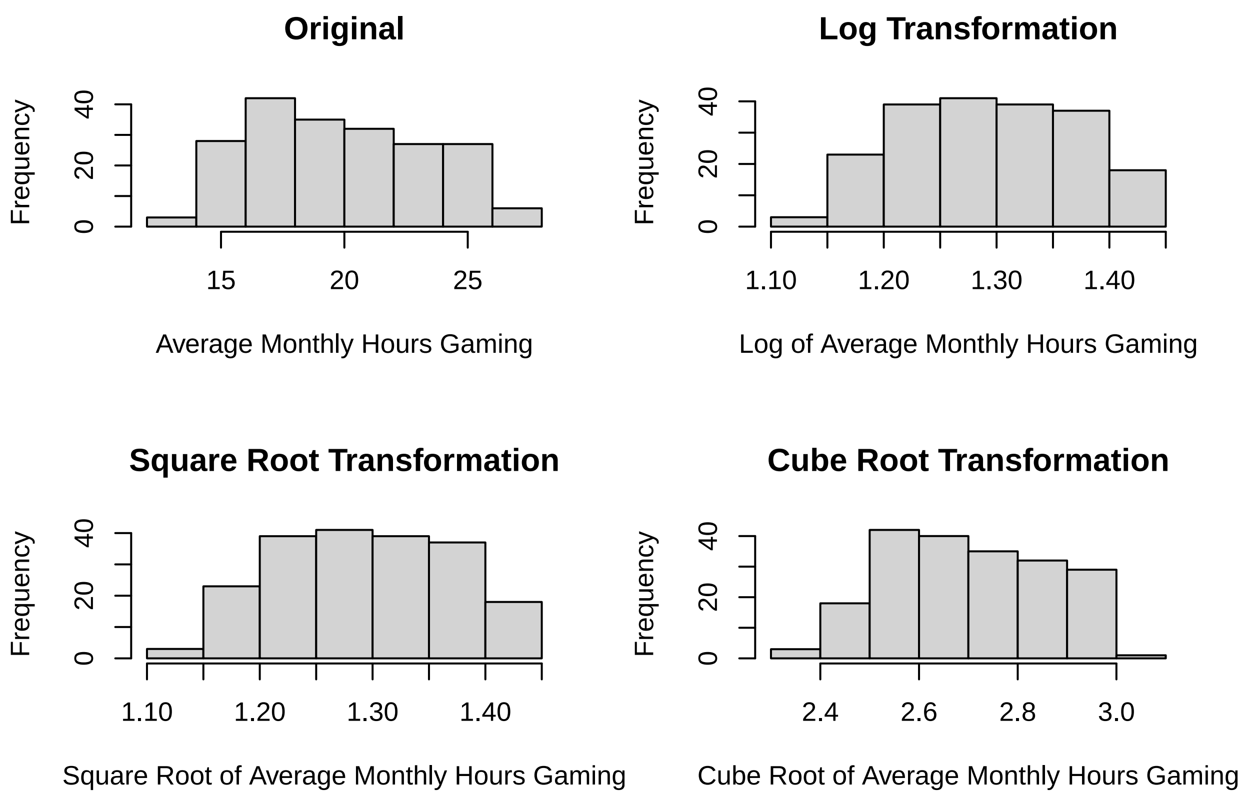
```r
sqrt_hours <- log10(sample_data$avg_monthly_hrs_gaming)
sqrt_bucks <- log10(sample_data$avg_monthly_expenditure_dlc)

df_sqrt <- data.frame(sqrt_hours, sqrt_bucks)
mod_sqrt <- lm(sqrt_bucks ~ sqrt_hours, data=df_sqrt)
summary(mod_sqrt)
```

```
##
## Call:
## lm(formula = sqrt_bucks ~ sqrt_hours, data = df_sqrt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.14413 -0.06323  0.00774  0.06338  0.12339
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.75827    0.08854  19.859   <2e-16 ***
## sqrt_hours  -0.01427    0.06833  -0.209    0.835
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07388 on 198 degrees of freedom
## Multiple R-squared:  0.0002204,  Adjusted R-squared:  -0.004829
## F-statistic: 0.04365 on 1 and 198 DF,  p-value: 0.8347
```

```r
cube_hours <- sample_data$avg_monthly_hrs_gaming^(1/3)
cube_bucks <- sample_data$avg_monthly_expenditure_dlc^(1/3)

df_cube <- data.frame(cube_hours, cube_bucks)
mod_cube <- lm(cube_bucks ~ cube_hours, data=df_cube)
summary(mod_cube)
```

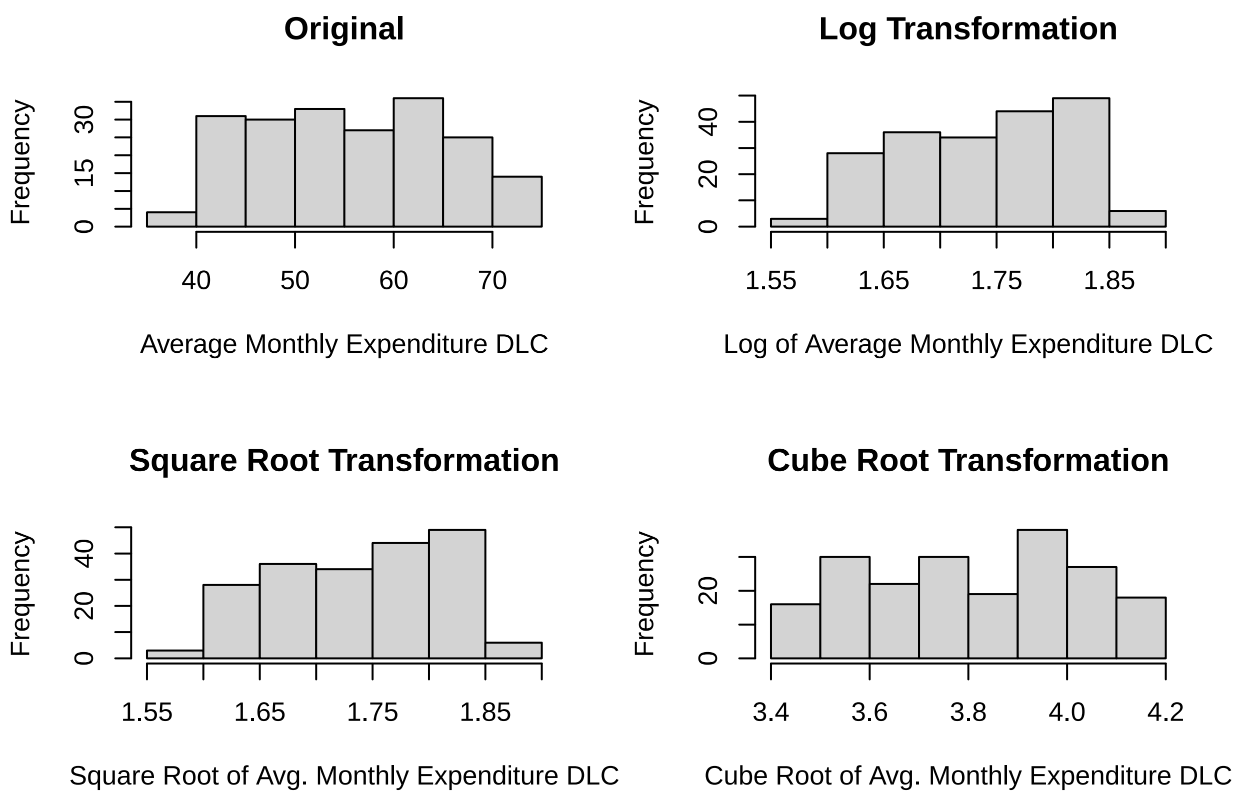
```
##
## Call:
## lm(formula = cube_bucks ~ cube_hours, data = df_cube)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40410 -0.18618  0.01661  0.18339  0.37140
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.86404    0.25986  14.869   <2e-16 ***
## cube_hours  -0.02096    0.09596  -0.218    0.827
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2148 on 198 degrees of freedom
## Multiple R-squared:  0.000241,   Adjusted R-squared:  -0.004808
## F-statistic: 0.04772 on 1 and 198 DF,  p-value: 0.8273
```

```r
par(mfrow=c(2,2)) # 2 rows and 2 columns

# avg_monthly_hrs_gaming transforms

hist(sample_data$avg_monthly_hrs_gaming, main='Original',
     xlab="Average Monthly Hours Gaming") # original
hist(log_hours, main='Log Transformation',
     xlab="Log of Average Monthly Hours Gaming") # log
hist(sqrt_hours, main='Square Root Transformation',
     xlab="Square Root of Average Monthly Hours Gaming") # square root
hist(cube_hours, main='Cube Root Transformation',
     xlab="Cube Root of Average Monthly Hours Gaming") # cube root
```




```r
par(mfrow=c(1,1)) # Reset to 1 row and 1 column
```

```r
par(mfrow=c(2,2)) # 2 rows and 2 columns

# avg_monthly_expenditure_dlc transforms

hist(sample_data$avg_monthly_expenditure_dlc, main='Original',
     xlab="Average Monthly Expenditure DLC") # original
hist(log_bucks, main='Log Transformation',
     xlab="Log of Average Monthly Expenditure DLC") # log
hist(sqrt_bucks, main='Square Root Transformation',
     xlab="Square Root of Avg. Monthly Expenditure DLC") # square root
hist(cube_bucks, main='Cube Root Transformation',
     xlab="Cube Root of Avg. Monthly Expenditure DLC") # cube root
```




```r
par(mfrow=c(1,1)) # Reset to 1 row and 1 column
```

```r
st1 <- shapiro.test(sample_data$avg_monthly_hrs_gaming)
st1
```

```
##
##  Shapiro-Wilk normality test
##
## data:  sample_data$avg_monthly_hrs_gaming
## W = 0.96331, p-value = 4.589e-05
```

```r
print(ifelse(st1$p.value < significance, "reject", "accept"))
```

```
## [1] "reject"
```

```r
st2 <- shapiro.test(log_hours)
print(ifelse(st2$p.value < significance, "reject", "accept"))
```

```
## [1] "reject"
```

```r
st3 <- shapiro.test(sqrt_hours)
print(ifelse(st3$p.value < significance, "reject", "accept"))
```

```
## [1] "reject"
```

```r
st4 <- shapiro.test(cube_hours)
print(ifelse(st4$p.value < significance, "reject", "accept"))
```

```
## [1] "reject"
```

```r
st5 <- shapiro.test(sample_data$avg_monthly_expenditure_dlc)
print(ifelse(st5$p.value < significance, "reject", "accept"))
```

```
## [1] "reject"
```

```r
st6 <- shapiro.test(log_bucks)
print(ifelse(st6$p.value < significance, "reject", "accept"))
```

```
## [1] "reject"
```

```r
st7 <- shapiro.test(sqrt_bucks)
print(ifelse(st7$p.value < significance, "reject", "accept"))
```

```
## [1] "reject"
```

```r
st8 <- shapiro.test(cube_bucks)
print(ifelse(st8$p.value < significance, "reject", "accept"))
```

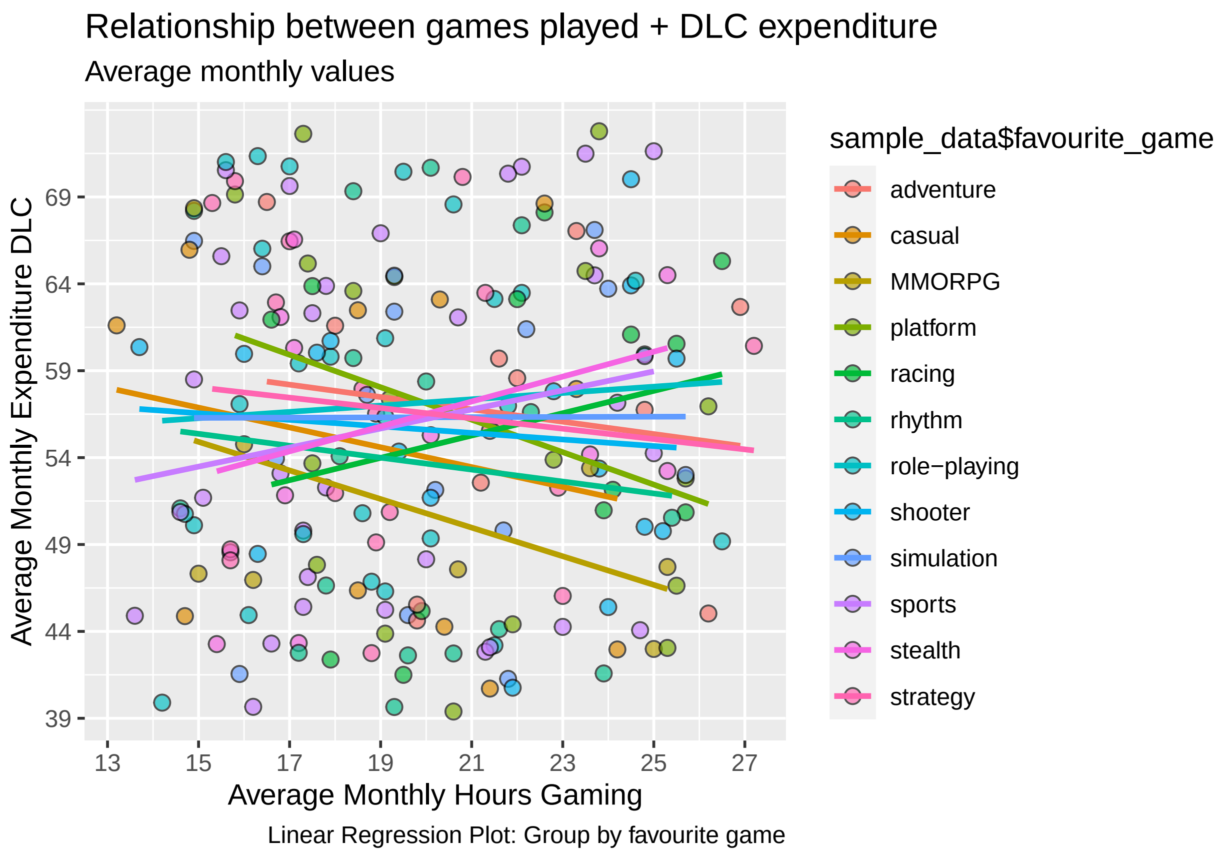
```
## [1] "reject"
```

Well, that was a waste of time.

Misc plots to try and make sense of the data (and failing)

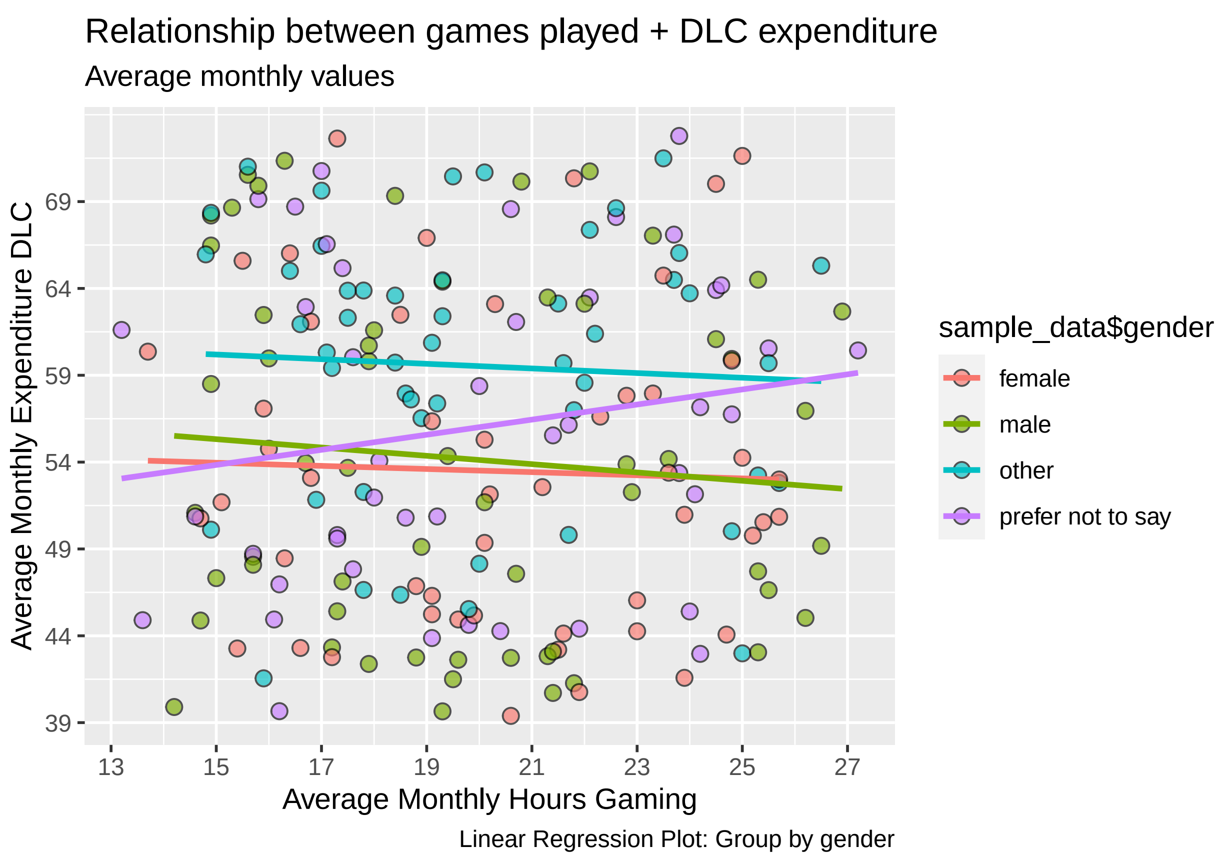
```
plot <- ggplot(data = mod, mapping = aes(x = avg_monthly_hrs_gaming,
                                  y = avg_monthly_expenditure_dlc,
                                  fill=sample_data$favourite_game,
                                  color=sample_data$favourite_game)) +
  geom_point(alpha = 0.66, # transparcency, lets stacked points show darker
             shape=21,# round
             color="black", # outline colour
             size=2.5) + # size (3 too big, 1 too small) +

  labs(title = "Relationship between games played + DLC expenditure",
       subtitle = "Average monthly values",
       caption = "Linear Regression Plot: Group by favourite game")

# Add x and y labels and geometry line to plot
plot + scale_x + scale_y +
  geom_smooth(method="lm", se=F)
```

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



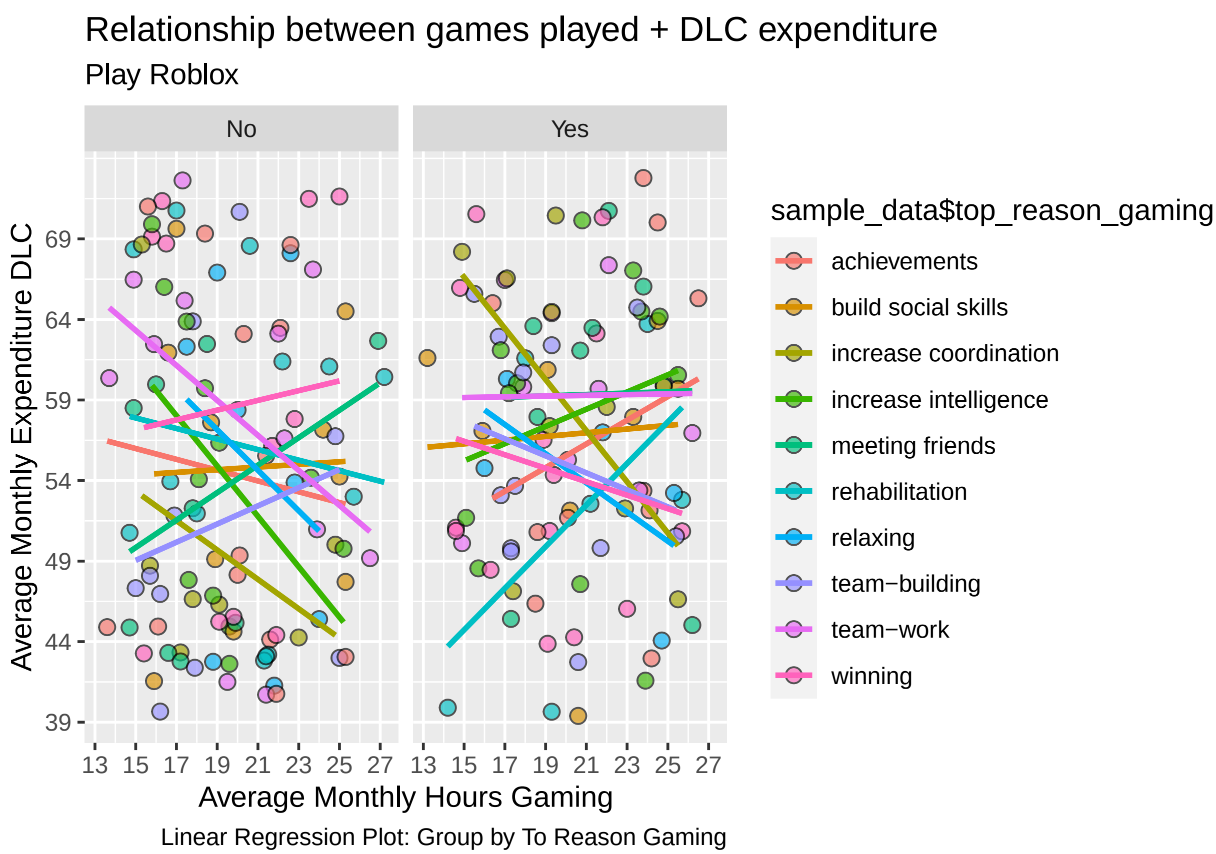
```r
plot <- ggplot(data = mod, mapping = aes(x = avg_monthly_hrs_gaming,
                                         y = avg_monthly_expenditure_dlc,
                                         fill=sample_data$gender,
                                         color=sample_data$gender)) +
  geom_point(alpha = 0.66, # transparcency, lets stacked points show darker
             shape=21,# round
             color="black", # outline colour
             size=2.5) + # size (3 too big, 1 too small) +

  labs(title = "Relationship between games played + DLC expenditure",
       subtitle = "Average monthly values",
       caption = "Linear Regression Plot: Group by gender")

# Add x and y labels and geometry line to plot
plot + scale_x + scale_y +
  geom_smooth(method="lm", se=F)
```

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



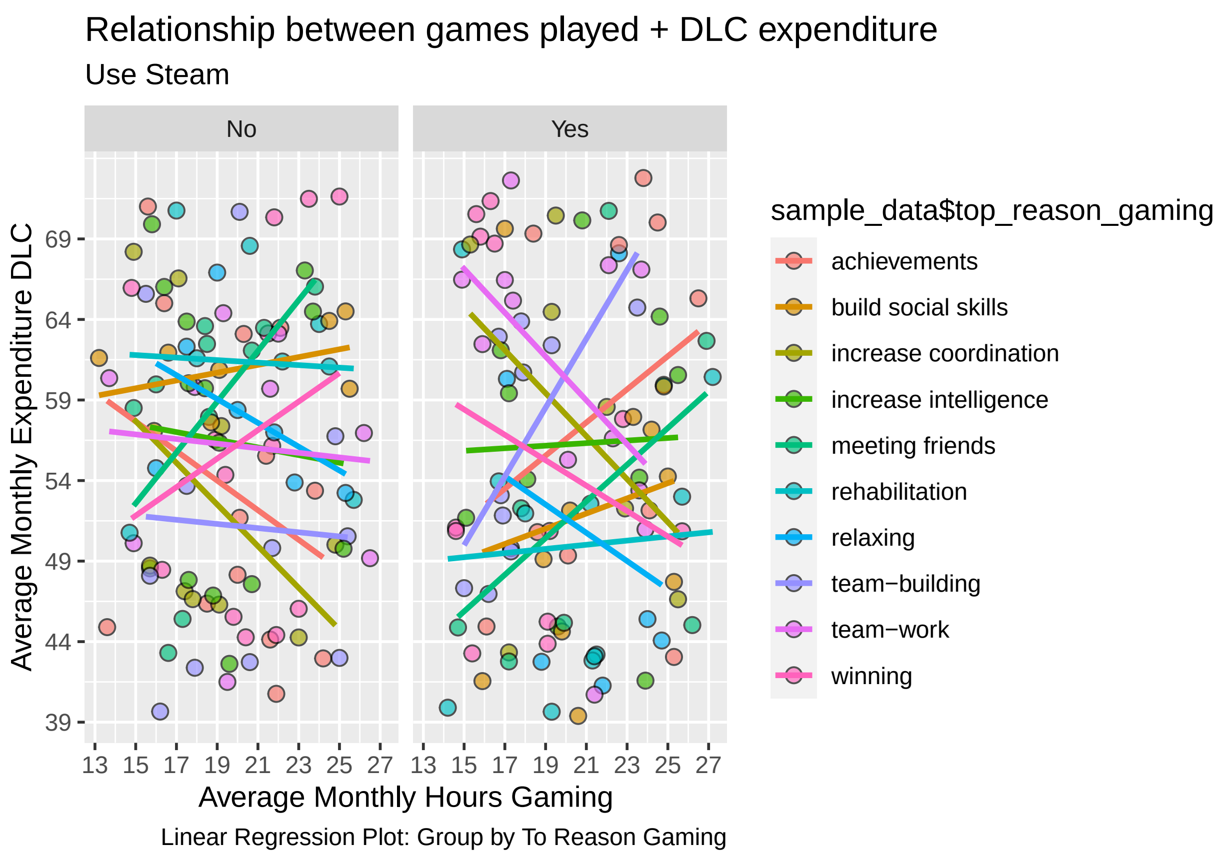
```r
plot <- ggplot(data = mod, mapping = aes(x = avg_monthly_hrs_gaming,
                                         y = avg_monthly_expenditure_dlc,
                                         fill=sample_data$top_reason_gaming,
                                         color=sample_data$top_reason_gaming)) +
  geom_point(alpha = 0.66, # transparcency, lets stacked points show darker
             shape=21,# round
             color="black", # outline colour
             size=2.5) + # size (3 too big, 1 too small) +

  labs(title = "Relationship between games played + DLC expenditure",
       subtitle = "Play Roblox",
       caption = "Linear Regression Plot: Group by To Reason Gaming")

# Add x and y labels and geometry line to plot
plot + scale_x + scale_y +
  geom_smooth(method="lm", se=F) +
  facet_grid(~sample_data$play_roblox)
```

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



```
plot <- ggplot(data = mod, mapping = aes(x = avg_monthly_hrs_gaming,
                                y = avg_monthly_expenditure_dlc,
                                fill=sample_data$top_reason_gaming,
                                color=sample_data$top_reason_gaming)) +
  geom_point(alpha = 0.66, # transparcency, lets stacked points show darker
             shape=21,# round
             color="black", # outline colour
             size=2.5) + # size (3 too big, 1 too small) +

  labs(title = "Relationship between games played + DLC expenditure",
       subtitle = "Use Steam",
       caption = "Linear Regression Plot: Group by To Reason Gaming")

# Add x and y labels and geometry line to plot
plot + scale_x + scale_y +
  geom_smooth(method="lm", se=F) +
  facet_grid(~sample_data$use_steam)
```

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



References

OpenIntro Statistics:

M., D., D., C. and Çetinkaya-Rundel, M., 2019. OpenIntro Statistics. OpenIntro, Incorporated.

OpenIntro Statistics. 2022. OpenIntro Statistics. [ONLINE] Available at: https://www.openintro.org/book/os/. [Accessed 18 January 2023].

Linear Regression:

Rebecca Bevans. 2023. Linear Regression in R | A Step-by-Step Guide & Examples. [ONLINE] Available at: https://www.scribbr.com/statistics/linear-regression-in-r/. [Accessed 18 January 2023].

Rebecca Bevans. 2023. Simple Linear Regression | An Easy Introduction & Examples. [ONLINE] Available at: https://www.scribbr.com/statistics/simple-linear-regression/. [Accessed 19 January 2023].

R Programming 101 (YouTube). 2023. Linear regression using R programming - YouTube. [ONLINE] Available at: https://www.youtube.com/watch?v=-mGXnm0fHtI. [Accessed 19 January 2023].

datacamp. 2023. Linear Regression in R Tutorial. [ONLINE] Available at: https://www.datacamp.com/tutorial/linear-regression-R. [Accessed 19 January 2023].

Linear Regression With R. 2023. Linear Regression With R. [ONLINE] Available at: http://r-statistics.co/Linear-Regression.html#Linear%20Regression%20Diagnostics. [Accessed 19 January 2023].

GGPlot

R Programming 101 (YouTube) 2023. ggplot for plots and graphs. An introduction to data visualization using R programming - YouTube. [ONLINE] Available at: https://www.youtube.com/watch?v=HPJn1CMvtmI. [Accessed 19 January 2023].

Data Carpentry contributors. 2023. Data visualization with ggplot2. [ONLINE] Available at: https://datacarpentry.org/R-ecology-lesson/04-visualization-ggplot2.html. [Accessed 19 January 2023].

ggplot2 point shapes - Easy Guides - Wiki - STHDA. 2023. ggplot2 point shapes - Easy Guides - Wiki - STHDA. [ONLINE] Available at: http://www.sthda.com/english/wiki/ggplot2-point-shapes. [Accessed 19 January 2023].

Least-Squares Regression

The Least-Square Regression Line and Equation. 2023. The Least-Square Regression Line and Equation. [ONLINE] Available at: https://rstudio-pubs-static.s3.amazonaws.com/199692_d02c8f7b352e4ec1b85544432ac28896.html. [Accessed 19 January 2023].

Linear Least Squares Regression — R Tutorial. 2023. 8. Linear Least Squares Regression — R Tutorial. [ONLINE] Available at: https://www.cyclismo.org/tutorial/R/linearLeastSquares.html. [Accessed 19 January 2023].

Normality Tests

Normality Test in R - Easy Guides - Wiki - STHDA. 2023. Normality Test in R - Easy Guides - Wiki - STHDA. [ONLINE] Available at: http://www.sthda.com/english/wiki/normality-test-in-r. [Accessed 19 January 2023].

Zach. 2023. How to Test for Normality in R (4 Methods) - Statology. [ONLINE] Available at: https://www.statology.org/test-for-normality-in-r/. [Accessed 19 January 2023].

Transform Data

Zach. 2023. How to Transform Data in R (Log, Square Root, Cube Root). [ONLINE] Available at: https://www.statology.org/transform-data-in-r/. [Accessed 19 January 2023].