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

Assumptions (Linear Regression)

1. **Homogeneity of variance (homoscedasticity):** The size of the error in our prediction doesn't change significantly across the values of the independent variable.
2. **Independence of observations:** the observations in the dataset were collected using statistically valid sampling methods, and there are no hidden relationships among observations.
3. **Normality:** The data follows a normal distribution.
4. **The relationship between the independent and dependent variable is linear:** the line of best fit through the data points is a straight line (rather than a curve or some sort of grouping factor).

Read 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.
```

Find usable columns

```
# assumption here is it would be very hard to plot a graph on anything else
# colnames(data) # list of column names
#sapply(data, class)
# str(data) # show column properties, find numeric columns
numeric_cols <- unlist(lapply(data, is.numeric))
numeric_data <- data[ , numeric_cols]
colnames(numeric_data)
```

```
## [1] "age"                          "avg_monthly_hrs_gaming"
## [3] "avg_years_playing_games"      "avg_monthly_expenditure_dlc"
```

Variables

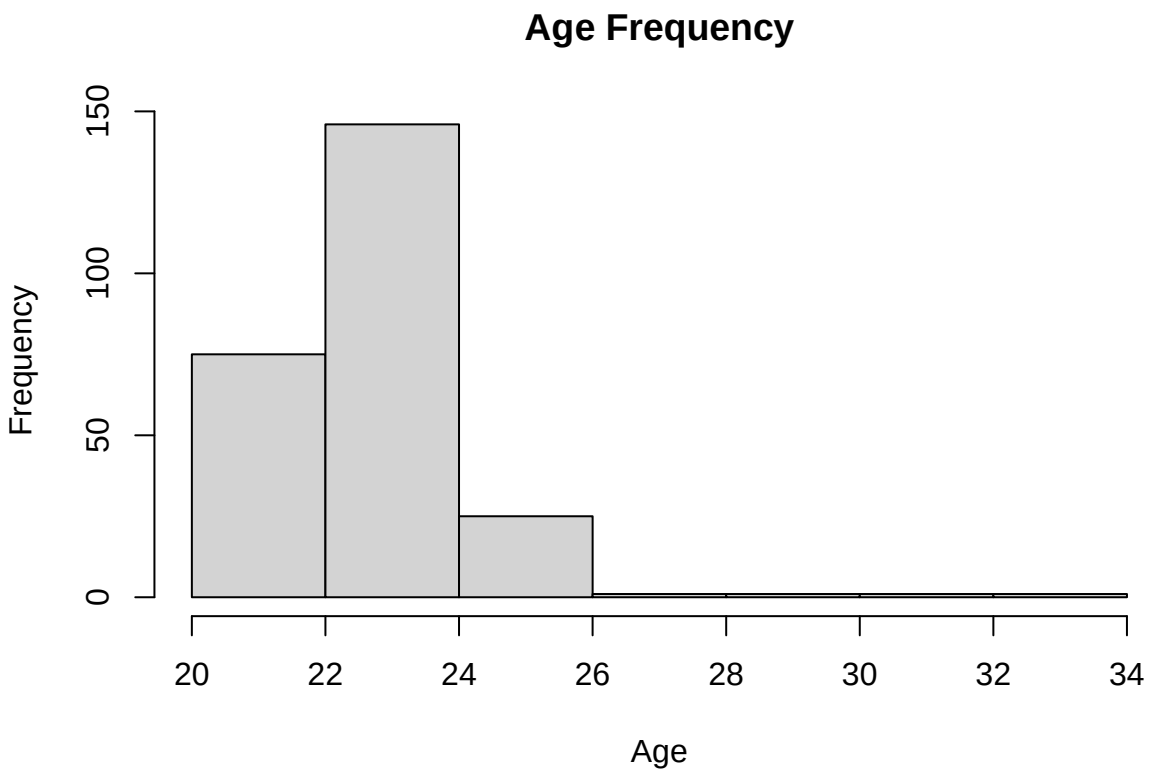
age, avg_monthly_hrs_gaming, avg_years_playing_games, avg_monthly_expenditure_dlc are all numeric fields.

Normality

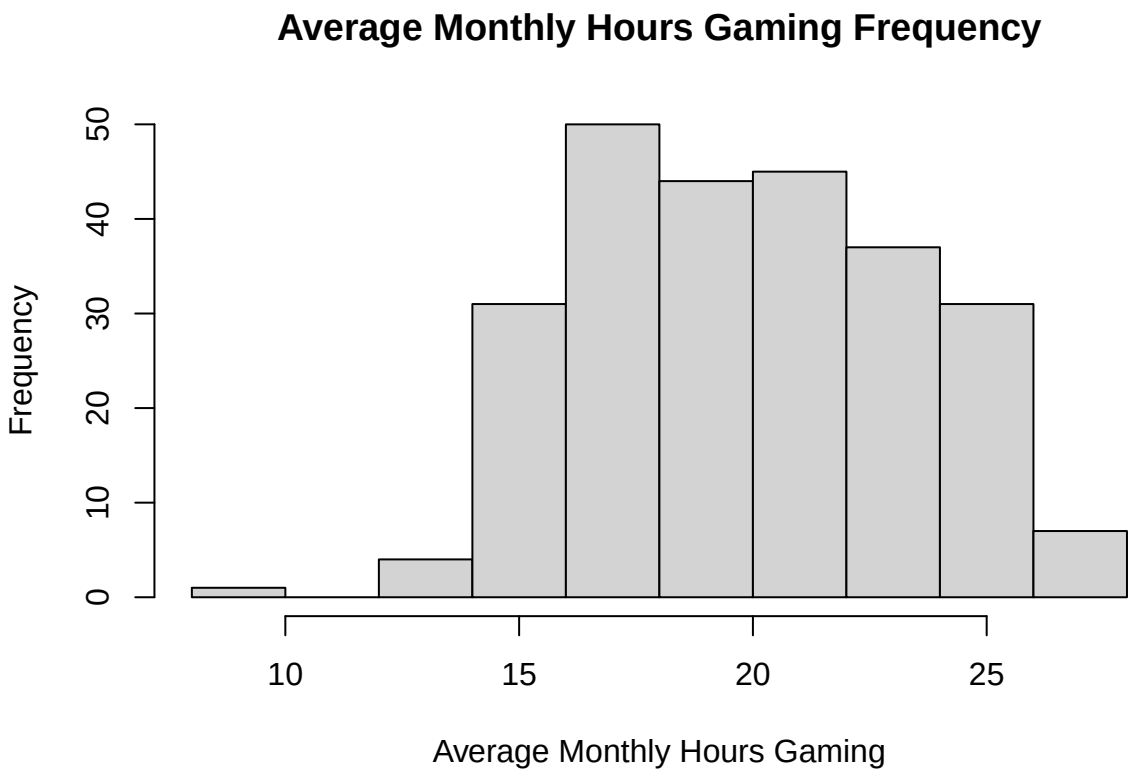
Histogram (Visual check)

Use histograms to visually check for normality. If the histogram is symmetrical/unimodal, then the data is assumed to be normally distributed.

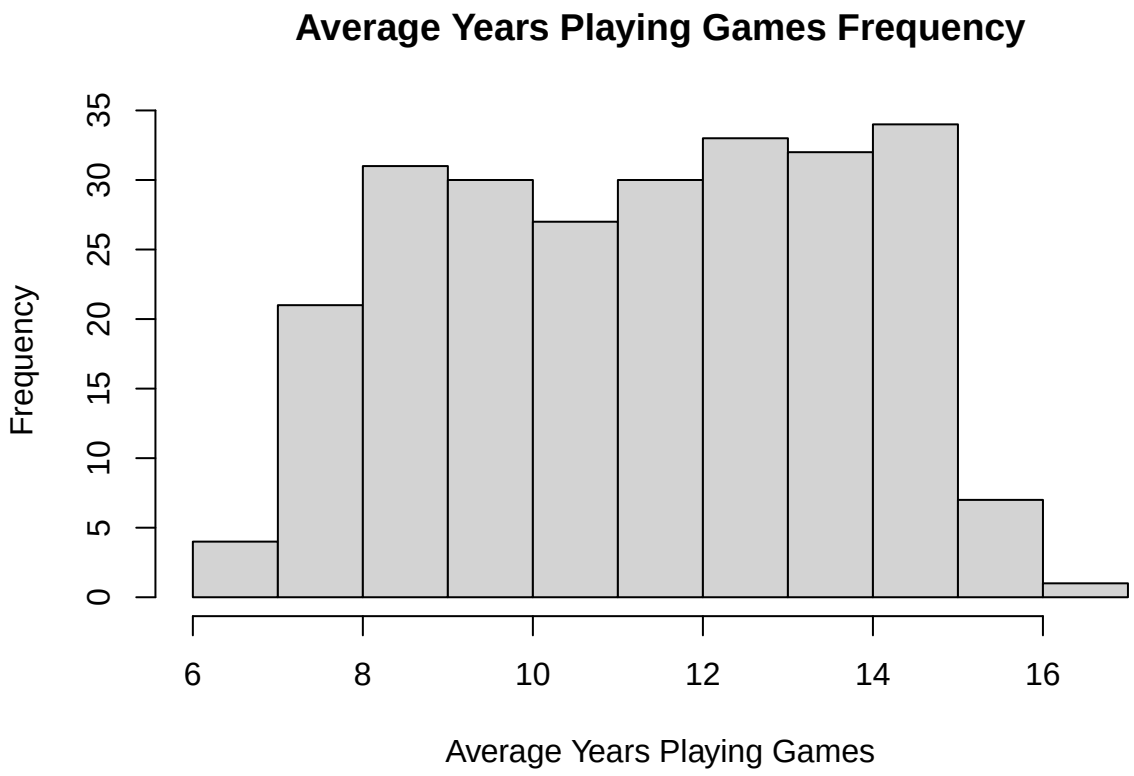
```
hist(data$age,
     main="Age Frequency",
     xlab = "Age")
```



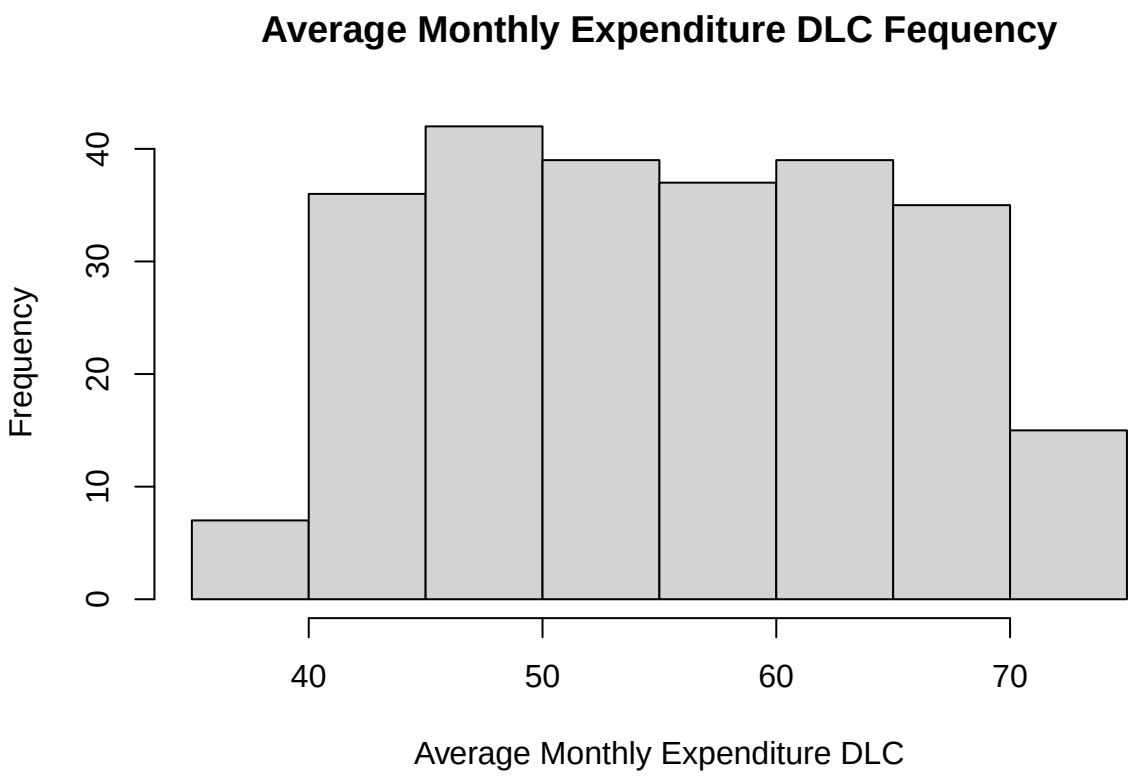

```
hist(data$avg_monthly_hrs_gaming,
     main="Average Monthly Hours Gaming Frequency",
     xlab="Average Monthly Hours Gaming")
```




```
hist(data$avg_years_playing_games,
     main="Average Years Playing Games Frequency",
     xlab = "Average Years Playing Games")
```




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



Shapiro-Wilk's test

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

If the P value is greater than 0.05 the answer is yes.

If the P value is less than or equal to 0.05 the answer is no.

```
st_age <- shapiro.test(data$age)
if(st_age$p.value < 0.05) print("nope") else print("yep")
```

```
## [1] "nope"
```

```
st_hours <- shapiro.test(data$avg_monthly_hrs_gaming)
if(st_hours$p.value < 0.05) print("nope") else print("yep")
```

```
## [1] "nope"
```

```
st_years <- shapiro.test(data$avg_years_playing_games)
if(st_years$p.value < 0.05) print("nope") else print("yep")
```

```
## [1] "nope"
```

```r
st_bucks <- shapiro.test(data$avg_monthly_expenditure_dlc)
if(st_bucks$p.value < 0.05) print("nope") else print("yep")
```

```
## [1] "nope"
```

Dependent Variable: avg_monthly_hrs_gaming

Independent Variable: avg_monthly_expenditure_dlc

```r
sample_data <- sample_n(data, 200) # tibble 200 x 11

# lm() -
# dependent var. ~ independent var.
mod <- lm(avg_monthly_expenditure_dlc ~ avg_monthly_hrs_gaming,
          data = sample_data)
summary(mod)
```

```
##
## Call:
## lm(formula = avg_monthly_expenditure_dlc ~ avg_monthly_hrs_gaming,
##     data = sample_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.4920  -8.2308   0.0123   8.0807  17.0935
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              56.34882    3.94919  14.268   <2e-16 ***
## avg_monthly_hrs_gaming -0.02783    0.19508  -0.143    0.887
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.27 on 198 degrees of freedom
## Multiple R-squared:  0.0001028,  Adjusted R-squared:  -0.004947
## F-statistic: 0.02035 on 1 and 198 DF,  p-value: 0.8867
```

```r
#attributes(mod)
#mod$residuals
# hist(mod$residuals)

plot <- ggplot(data = mod, mapping = aes(x = avg_monthly_hrs_gaming,
                                         y = avg_monthly_expenditure_dlc)) +
  # geom_point(alpha = 0.1, color = "blue") # add colours for points
  geom_point() + # plot dataset in a scatter plot
  labs(title = "Relationship between games monthly hours played + DLC expenditure",
       x = "Average Monthly Hours Gaming",
       y = "Average Monthly Expenditure DLC")

# plot + geom_smooth(method = lm, se = FALSE, formula=y~x) # probably this one
# plot + stat_smooth(method = lm, formula = y ~ x, geom = "smooth") # ok
# plot + geom_smooth(method = "loess", se = FALSE, formula=y~x) # curved line
```

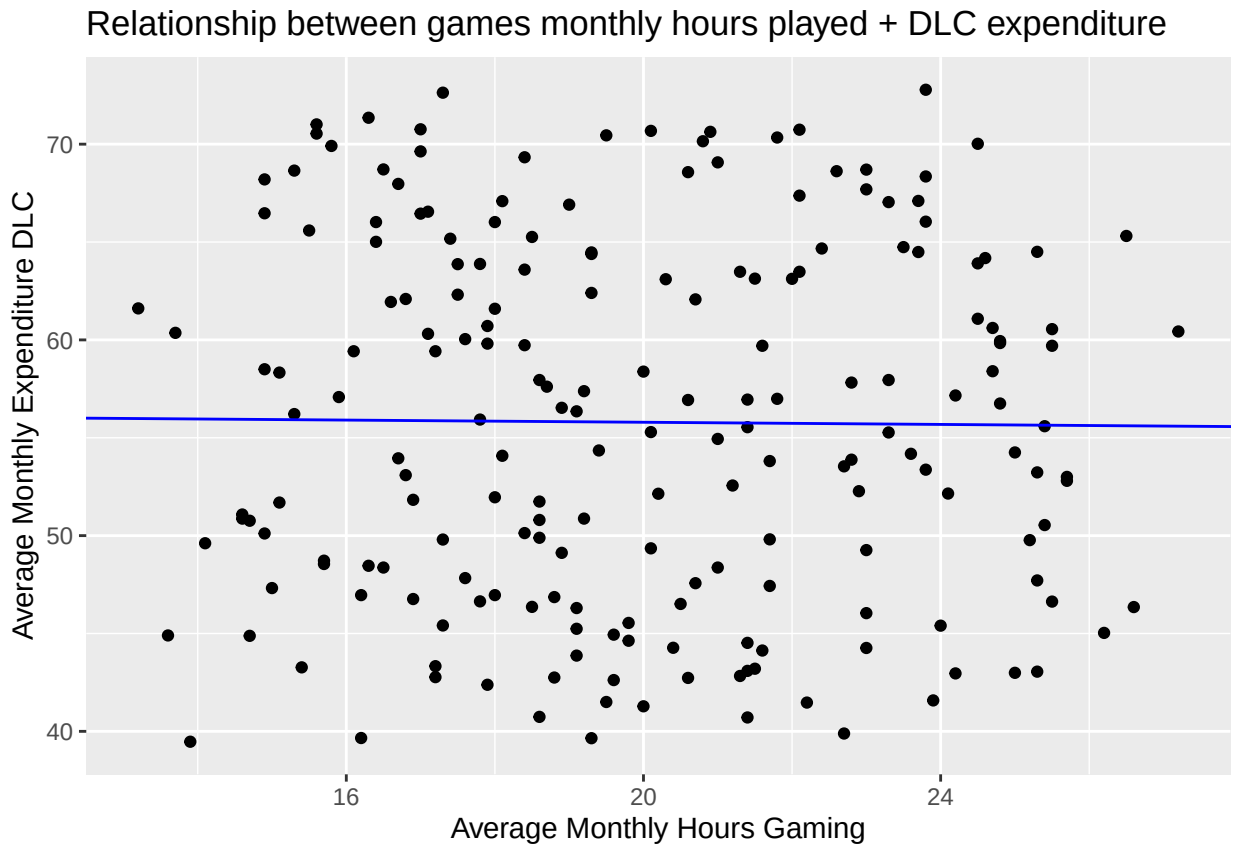
```
coeff <- coefficients(mod)
coeff
```

```
##            (Intercept) avg_monthly_hrs_gaming
##            56.34882376            -0.02782948
```

```
intercept <- coeff[1]
slope <- coeff[2]
slope
```

```
## avg_monthly_hrs_gaming
##            -0.02782948
```

```
plot +
  geom_abline(intercept = intercept, slope = slope, color="blue") # regression line
```




```
# + geom_abline(mapping = aes(x = avg_monthly_hrs_gaming, y = avg_monthly_expenditure_dlc), data = mod)
```