In [1]:
```
# Loading the libraries
library(tidyverse)
library(repr)
library(tidymodels)
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✓ dplyr     1.1.4     ✓ readr     2.1.5
✓ forcats   1.0.0     ✓ stringr   1.5.1
✓ ggplot2   3.5.1     ✓ tibble    3.2.1
✓ lubridate 1.9.3     ✓ tidyr     1.3.1
✓ purrr     1.0.2
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all confl
icts to become errors
── Attaching packages ─────────────────────────────────────── tidymodels 1.1.1 ──

✓ broom        1.0.6     ✓ rsample      1.2.1
✓ dials        1.3.0     ✓ tune         1.1.2
✓ infer        1.0.7     ✓ workflows    1.1.4
✓ modeldata    1.4.0     ✓ workflowsets 1.0.1
✓ parsnip      1.2.1     ✓ yardstick    1.3.1
✓ recipes      1.1.0

── Conflicts ───────────────────────────────────────── tidymodels_conflicts() ──
✖ scales::discard() masks purrr::discard()
✖ dplyr::filter()   masks stats::filter()
✖ recipes::fixed()  masks stringr::fixed()
✖ dplyr::lag()      masks stats::lag()
✖ yardstick::spec() masks readr::spec()
✖ recipes::step()   masks stats::step()
• Use suppressPackageStartupMessages() to eliminate package startup messages
```

K-NN Classification of PLAIcraft subscribers Based on Age and Played Hours

Introduction

Popular online kids game, Minecraft, encourages kids to be creative by allowing them to build, craft, and explore in an expansive cube sandbox world. We are exploring the data collected by a research group led by Frank Wood in Computer Science at UBC based in a Minecraft server to target recruitment

Plaicraft is a version of Minecraft by UBC used to conduct AI research based on player movement, in-game speech, and other information. Through this research, information was gathered about each player and was compiled into a CSV (comma separated variables) dataset called "player" with a total of 7 variables and 195

observations for each, creating a 196 x 7 table. Each row contains information about one user who logged into Plaicraft.

The 7 variables include character, logical, and double variables:

- `experience` : character variable, player's familiarity with Minecraft (e.g. Beginner, Amateur, Regular, Pro, Veteran)
- `subscribe` : logical variable, player's newsletter subscription status
- `hashedEmail` : character variable, player's encoded email
- `played_hours` : double variable, number of hours spent on Plaicraft
- `name` : character variable, player's name
- `gender` : character variable, player's gender
- `Age` : double variable, player's age (ranging from 8-50 years old)

Data Quality and Potential Issues

1. If data was collected from a particular demographic, for example, UBC students, this could lead to potential skew in data.

2. In the played_hours dataset, there could be observations from AFK (away from keyboard) players who logged into the server but remained idle, artificially inflating their recorded playtime. This could infer misleading conclusions.

3. Variables like experience and gender are stored as a free-text character which could have potential typos or multiple forms of the same category. Converting them into a factor could help prevent these inconsistencies in spelling.

Summary Statistics

Below is the summary statistics of all 7 variables and the overview of the players dataset.

- Number of Observations : 196
- Number of Variables : 7

In [2]:
```
players <- read_csv("https://raw.githubusercontent.com/jseo07/dsci100-data/refs/
```

```
Rows: 196 Columns: 7
── Column specification ────────────────────────────────────────────────────────
Delimiter: ","
chr (4): experience, hashedEmail, name, gender
dbl (2): played_hours, Age
lgl (1): subscribe

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 messag
e.
```

In [3]:
```
summary(players)
```

```
 experience        subscribe       hashedEmail         played_hours
Length:196         Mode :logical   Length:196         Min.   :  0.000
Class :character   FALSE:52        Class :character   1st Qu.:  0.000
Mode  :character   TRUE :144       Mode  :character   Median :  0.100
                                                      Mean   :  5.846
                                                      3rd Qu.:  0.600
                                                      Max.   :223.100

    name              gender              Age        
Length:196         Length:196         Min.   : 8.00  
Class :character   Class :character   1st Qu.:17.00  
Mode  :character   Mode  :character   Median :19.00  
                                      Mean   :20.52  
                                      3rd Qu.:22.00  
                                      Max.   :50.00  
                                      NA's   :2      
```

Research Question

Our predictive question is whether we can use played minutes and age to predict player subscription?

In [4]:
```
# players_split <- initial_split(players, prop = 0.75, strata = subscribe)
# players_training <- training(players_split)
# players_testing <- testing(players_split)
# head(players_training)
# head(players_testing)
```

Methods & Results

To predict subscription of players, we are going to use K-nearest neighbors classification.

The players dataset is loaded above in the introduction.

Wrangling and Cleaning the Data

The variables we are interested in are:

- Predictors: played_hours, Age
- Response Variable: subscribe

  The relevant columns are selected below.

In [5]:
```
players_select <- players |>
select(subscribe, played_hours, Age)
head(players_select)
```

A tibble: 6 × 3

subscribe	played_hours	Age
<lgl>	<dbl>	<dbl>
TRUE	30.3	9
TRUE	3.8	17
FALSE	0.0	17
TRUE	0.7	21
TRUE	0.1	21
TRUE	0.0	17

Table 1. The head of table with relevant variables selected.

The players dataset consists of many observations with less than one hour of played time saved as 0, which neglects the sub-one hour observations in our exploration. Hence, played_hours was converted to played_minutes.

In [45]:
```
players_minutes_na <- players_select |>
mutate(played_minutes = played_hours * 60) |>
mutate(subscribe = as.factor(subscribe)) |>
select(-played_hours)

players_minutes <- drop_na(players_minutes_na)
head(players_minutes)
```

A tibble: 6 × 3

subscribe	Age	played_minutes
<fct>	<dbl>	<dbl>
TRUE	9	1818
TRUE	17	228
FALSE	17	0
TRUE	21	42
TRUE	21	6
TRUE	17	0

Table 2. The head of table with players_hours converted into players_minutes.

Summary of Cleaned Dataset

The cleaned dataset have the following summary statistics:

In [7]:
```
summary(players_minutes)
players_average <- players_minutes |>
select(played_minutes, Age) |>
```

```
map_dfr(mean, na.rm = TRUE)
players_average
```

```
subscribe        Age         played_minutes
FALSE: 52   Min.   : 8.00   Min.   :    0.0
TRUE :142   1st Qu.:17.00   1st Qu.:    0.0
            Median :19.00   Median :    6.0
            Mean   :20.52   Mean   :  354.3
            3rd Qu.:22.00   3rd Qu.:   36.0
            Max.   :50.00   Max.   :13386.0
```

A tibble: 1 × 2

played_minutes	Age
<dbl>	<dbl>
354.2784	20.52062

Table 3. The means of played_minutes and Age.

The Relationship Between minutes played and age to subscription status

In [8]:
```
options(repr.plot.width = 25, repr.plot.height = 9)

players_plot <- players_minutes |>
ggplot(aes(x = played_minutes, y = Age)) +
geom_point(aes(colour = subscribe)) +
labs(x = "Minutes Played", y = "Age of Player (Years)", colour = "Subscribed Or
theme(text = element_text(size = 14))
players_plot
```

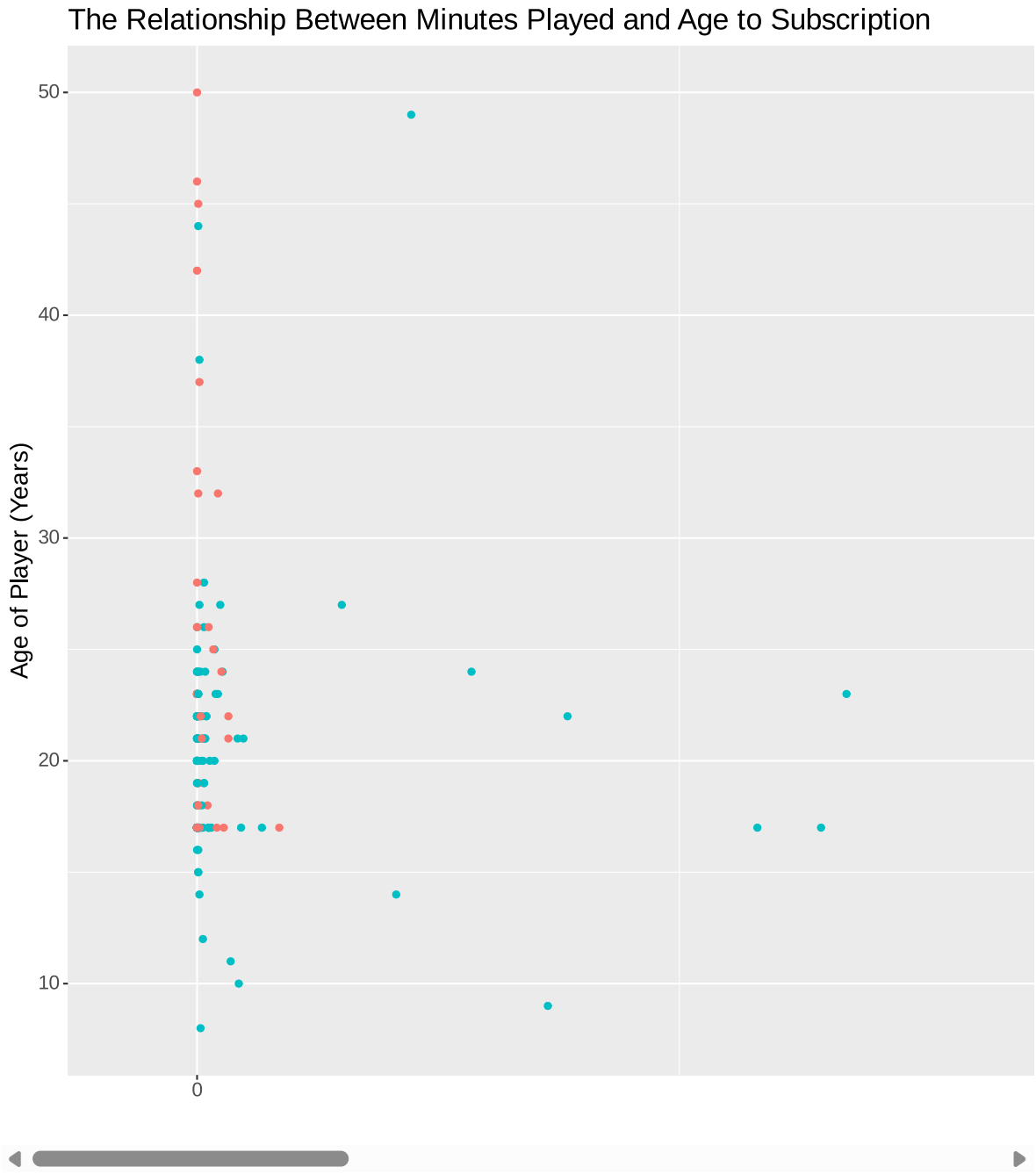

Figure 1. Scatterplot of the relationship between minutes played and age to subscription.

The scatterplot above shows that the majority of players have a low (close to 0) played_minutes, but the players who did play a significant number of minutes are mostly subscribed. Hence there is a clear sign that higher number of minutes lead to subscription being true. However, there is no clear pattern shown for the relationship between age of players and subscription.

Performing K-nearest algorithm

Splitting the Data

The dataset is split into training and testing datsets. 75% of the dataset was used for training dataset and the rest was used as the testing set.

In [9]:
```
players_split <- initial_split(players_minutes, prop = 0.75, strata = subscribe)
players_training <- training(players_split)
players_testing <- testing(players_split)
head(players_training)
head(players_testing)
```

A tibble: 6 × 3

subscribe	Age	played_minutes
<fct>	<dbl>	<dbl>
FALSE	17	0
FALSE	21	0
FALSE	22	0
FALSE	18	6
FALSE	25	84
FALSE	24	0

A tibble: 6 × 3

subscribe	Age	played_minutes
<fct>	<dbl>	<dbl>
TRUE	17	228
TRUE	21	6
TRUE	17	0
TRUE	19	0
TRUE	17	6
TRUE	22	0

Table 4. The head of training and testing dataset.

In [15]:
```
players_recipe <- recipe(subscribe ~ Age + played_minutes, data = players_traini
        step_scale(all_predictors()) |>
    step_center(all_predictors())

knn_spec <- nearest_neighbor(weight_func = "rectangular", neighbors = tune()) |>
        set_engine("kknn") |>
        set_mode("classification")

players_vfold <- vfold_cv(players_training, v = 5, strata = subscribe)


k_vals <- tibble(neighbors = seq(from = 1, to = 25, by = 1))

knn_results <- workflow() |>
        add_recipe(players_recipe) |>
        add_model(knn_spec) |>
        tune_grid(resamples = players_vfold, grid = k_vals) |>
        collect_metrics()
```

```
accuracies <- knn_results |>
        filter(.metric == "accuracy")
head(accuracies)
```

A tibble: 6 × 7

neighbors	.metric	.estimator	mean	n	std_err	.config
<dbl>	<chr>	<chr>	<dbl>	<int>	<dbl>	<chr>
1	accuracy	binary	0.4837274	5	0.05010412	Preprocessor1_Model01
2	accuracy	binary	0.4908703	5	0.05479245	Preprocessor1_Model02
3	accuracy	binary	0.4904269	5	0.06307613	Preprocessor1_Model03
4	accuracy	binary	0.4763875	5	0.05836992	Preprocessor1_Model04
5	accuracy	binary	0.5463054	5	0.05361543	Preprocessor1_Model05
6	accuracy	binary	0.5600985	5	0.05007699	Preprocessor1_Model06

Table 5. Tabulated results of K values against accuracy.

In [16]:
```
options(repr.plot.width = 25, repr.plot.height = 9)

accuracy_vs_k <- ggplot(accuracies, aes(x = neighbors, y = mean)) +
    geom_point() +
    geom_line() +
    labs(x = "Neighbors", y = "Accuracy Estimate", colour = "Subscribed Or Not",
    theme(text = element_text(size = 12))

accuracy_vs_k
```

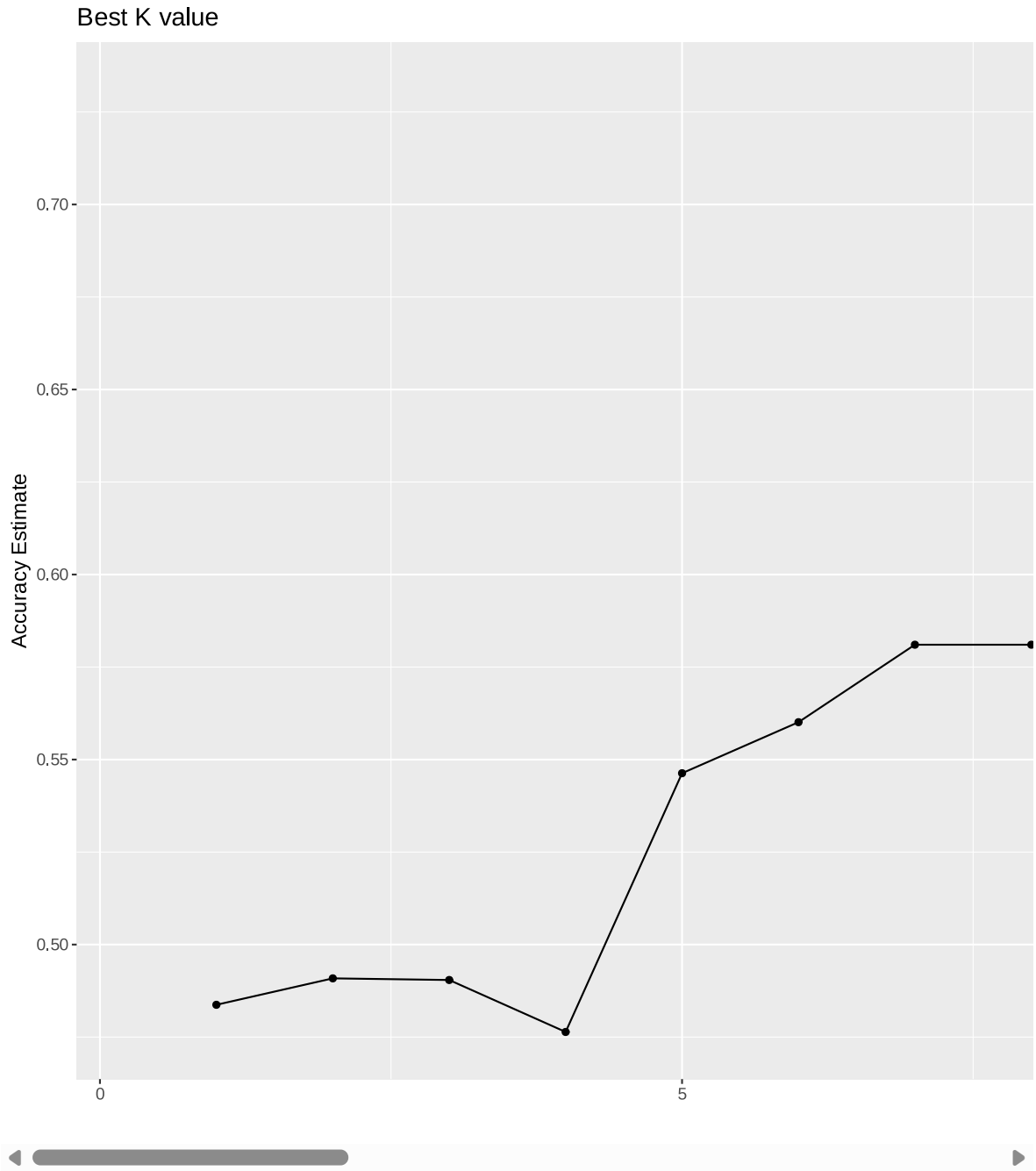

Figure 2. Visualization of Accuracy VS K value to find the best K.

The "elbow", or where the accuracy stops increasing dramatically and levels off or begins to decrease is at K = 21. Hence, we chose K value to be 21 for our model.

Creating The Model: Using the Best K Value

The model is created with the best value of K chosen from above:

In [21]:
```
players_knn <- nearest_neighbor(weight_func = "rectangular", neighbors = 21) |>
        set_engine("kknn") |>
        set_mode("classification")

knn_fit <- workflow() |>
        add_recipe(players_recipe) |>
        add_model(players_knn) |>
        fit(data = players_training)
```

With the model created, we test and predict the subscription status of players. The model is evaluated based on accuracy, precision and recall.

In [44]:
```
players_predictions <- predict(knn_fit, players_testing) |>
bind_cols(players_testing)

players_accuracy  <- players_predictions |>
metrics(truth = subscribe, estimate = .pred_class) |>
filter(.metric == "accuracy")

players_accuracy

players_precision <- players_predictions |>
precision(truth = subscribe, estimate = .pred_class, event_level="first")
players_precision

players_recall <- players_predictions |>
recall(truth = subscribe, estimate = .pred_class, event_level="first")
players_recall
```

A tibble: 1 × 3

.metric	.estimator	.estimate
<chr>	<chr>	<dbl>
accuracy	binary	0.755102

A tibble: 1 × 3

.metric	.estimator	.estimate
<chr>	<chr>	<dbl>
precision	binary	1

A tibble: 1 × 3

.metric	.estimator	.estimate
<chr>	<chr>	<dbl>
recall	binary	0.07692308

Table 6. Accuracy, precision, and recall of the model.

In [31]:
```
players_confusion <- players_predictions |>
             conf_mat(truth = subscribe, estimate = .pred_class)
players_confusion
```

```
          Truth
Prediction FALSE TRUE
     FALSE     1    0
     TRUE     12   36
```

Table 7. The confusion matrix of the model

The implications of this confusion matrix and evaulation data will be discussed further in the discussion section.

Discussion

Result

Best K Value:

K = 21 was found to be the best k value.

Based on the size of the players.csv dataset, it would be expected that the assigned k value would be on the smaller side. The impact of this finding is that this model would be expressing overfitting tendencies where it is too sensitive to the training data which decreases the reliability of its prediction.

Reflecting on the initial dataset:

Reflecting on the two predictors that we chose, played_minutes and age, we have come to realise that they do not strongly correlate with subscription status, and thus do not offer a great prediction of subscription as seen within our visualisation.

Due to the small sample size of the player.csv data set, the efficacy of our classification model is limited. However, we also have to consider that KNN-classification models do not function as efficiently when dealing with larger data sets.

Evaluation of the Model:

- Accuracy: Our results show that our classification model is adequately accurate (75%) but is not a reliable metric for more high stakes operations. There is a clear bias toward TRUE predictions because in the original dataset, there were more TRUE subscription observations compared to FALSE, when we used a high k value (21) the majority was skewed toward the larger quantity observation.
- Precision: The precision of the classifier was 100% which is not representative of the actual precision because there was only 1 correctly predicted FALSE value and no TRUE predictions. While, the other 48 predictions were TRUE.
- Recall: Similarly, for recall, its value is unacceptable for a classification model. A value of 8% means that our model is not able to find the TRUE predictions within testing set to a proper degree. We can interpret that the reason for these metric values is that our classification model is that the predictor variables we chose do not have a strong correlation to the response variable as also seen within Table 6.

Conclusion

Through this journey of a group project, we have come to appreciate the importance of using wrangled data and visualisation to then utilise a classification model for predictions of a new observation.

For future projects, the following questions would be beneficial to consider: What variables would be better predictors of subscription status? What variable should be predict using the classification model?

In [ ]: