Clustering with Credit Card Customers

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.0     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(tidymodels)
```

```
## -- Attaching packages -------------------------------------- tidymodels 1.2.0 --
## v broom        1.0.5      v rsample      1.2.1
## v dials        1.2.1      v tune         1.2.0
## v infer        1.0.7      v workflows    1.1.4
## v modeldata    1.3.0      v workflowsets 1.1.0
## v parsnip      1.2.1      v yardstick    1.3.1
## v recipes      1.0.10
## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Search for functions across packages at https://www.tidymodels.org/find/
```

```r
library(DataExplorer)
library(tidyclust)
```

```
##
## Attaching package: 'tidyclust'
##
## The following objects are masked from 'package:parsnip':
##
##     knit_engine_docs, list_md_problems
```

```r
library(dbscan)
```

```
##
## Attaching package: 'dbscan'
##
## The following object is masked from 'package:stats':
##
##     as.dendrogram
```

```r
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

Load Data

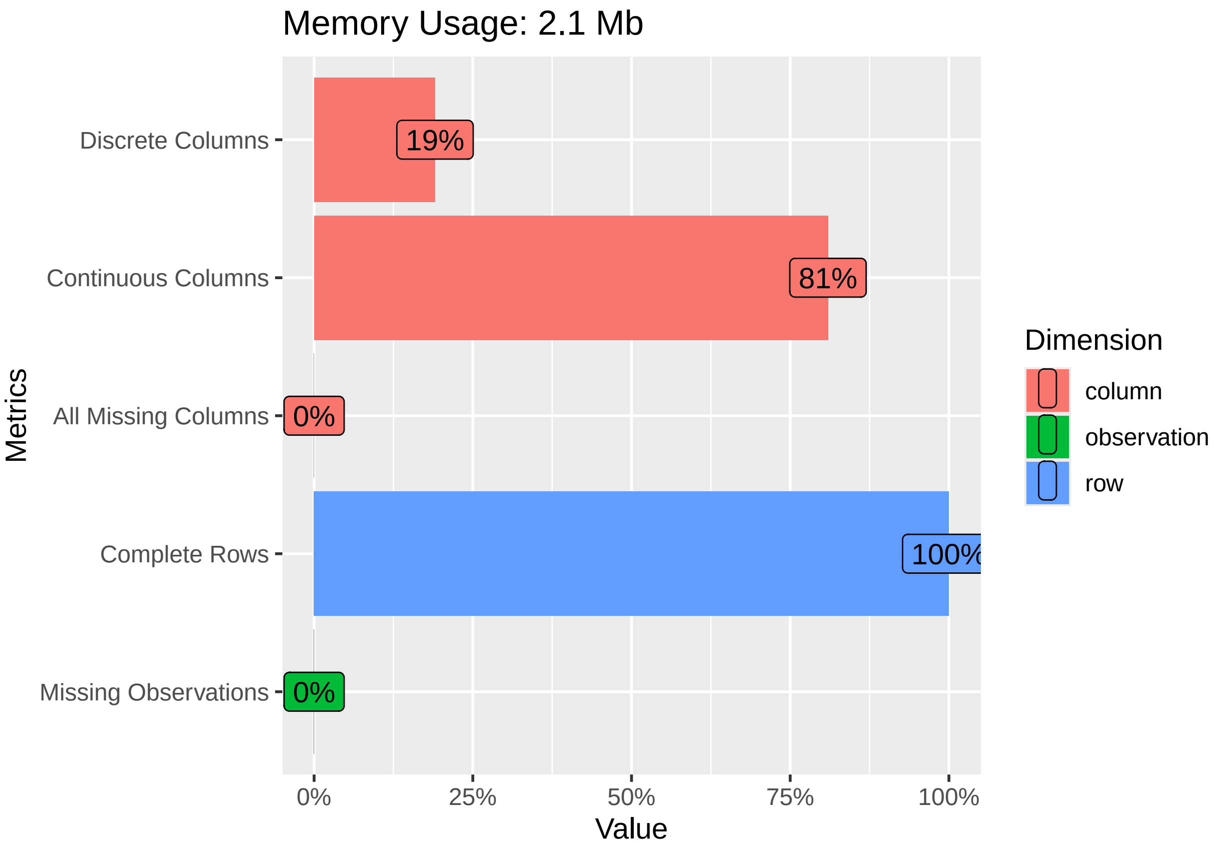
```r
df <- read_csv("credit_card_customers.csv", show_col_types = FALSE)
```

Data Cleaning

```r
df <- df %>%
  mutate(
    CLIENTNUM = as.character(CLIENTNUM),
    Attrition_Flag = recode(Attrition_Flag,
                            "Existing Customer" = "Retained",
                            "Attrited Customer" = "Churned"),
    Gender = factor(Gender, ordered = FALSE),
    Marital_Status = factor(Marital_Status, ordered = FALSE),
    Education_Level = as.integer(factor(Education_Level, levels = c("Unknown", "Uneducated", "High Scho
                                                         "College", "Graduate", "Post-Graduate",
                                                         "Doctorate"), ordered = TRUE)),
    Income_Category = as.integer(factor(Income_Category, levels = c("Unknown", "Less than $40K",
                                                          "$40K - $60K", "$60K - $80K",
                                                          "$80K - $120K", "$120K +"), ordered = TRUE)),
    Card_Category = as.integer(factor(Card_Category, c("Blue", "Silver", "Gold", "Platinum"), ordered =
  )
```

Data Exploration

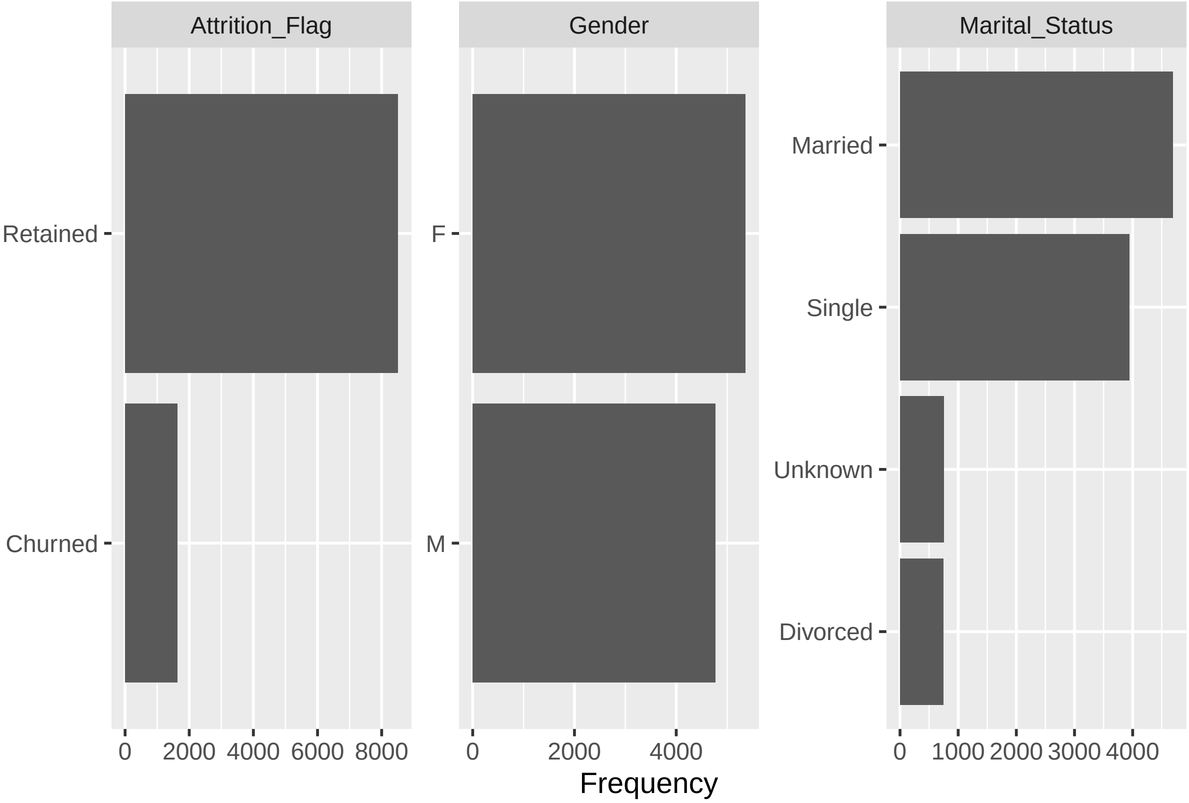
```r
plot_intro(df)
```



Bar plots of categorical variables

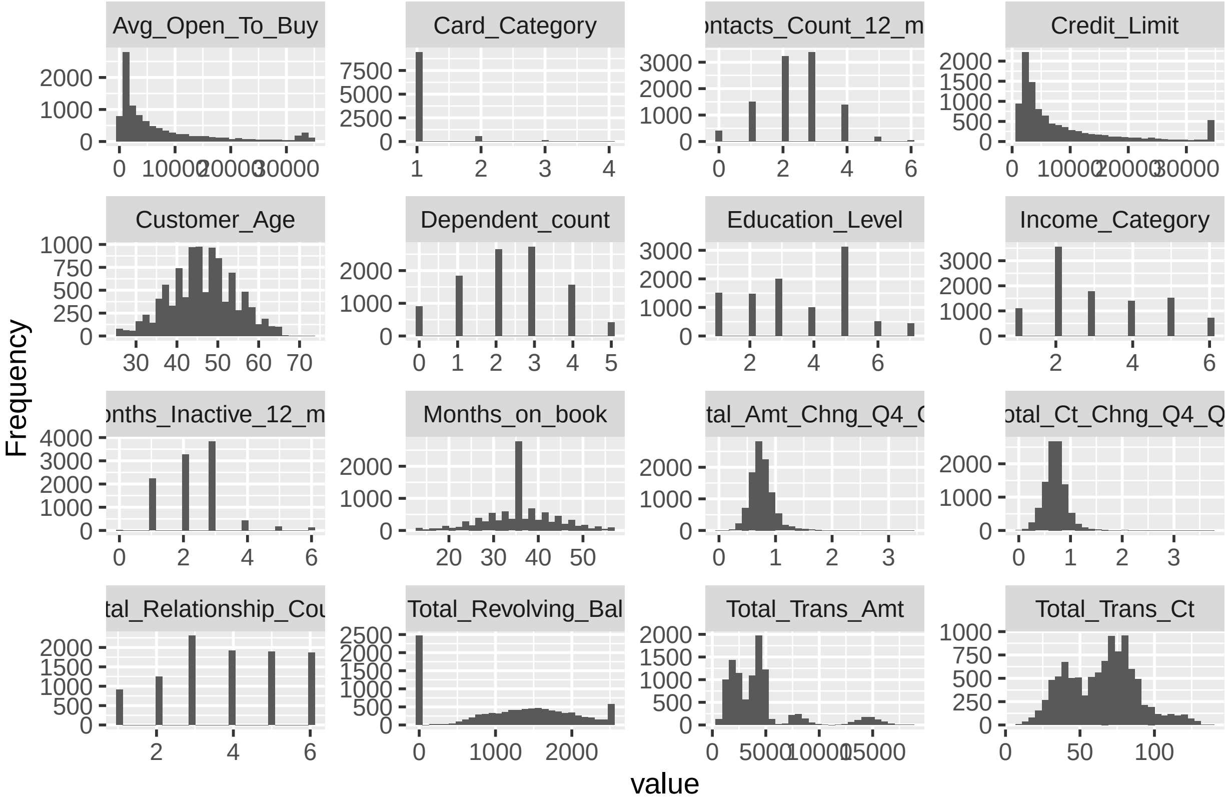
```
plot_bar(df)
```

```
## 1 columns ignored with more than 50 categories.
## CLIENTNUM: 10127 categories
```

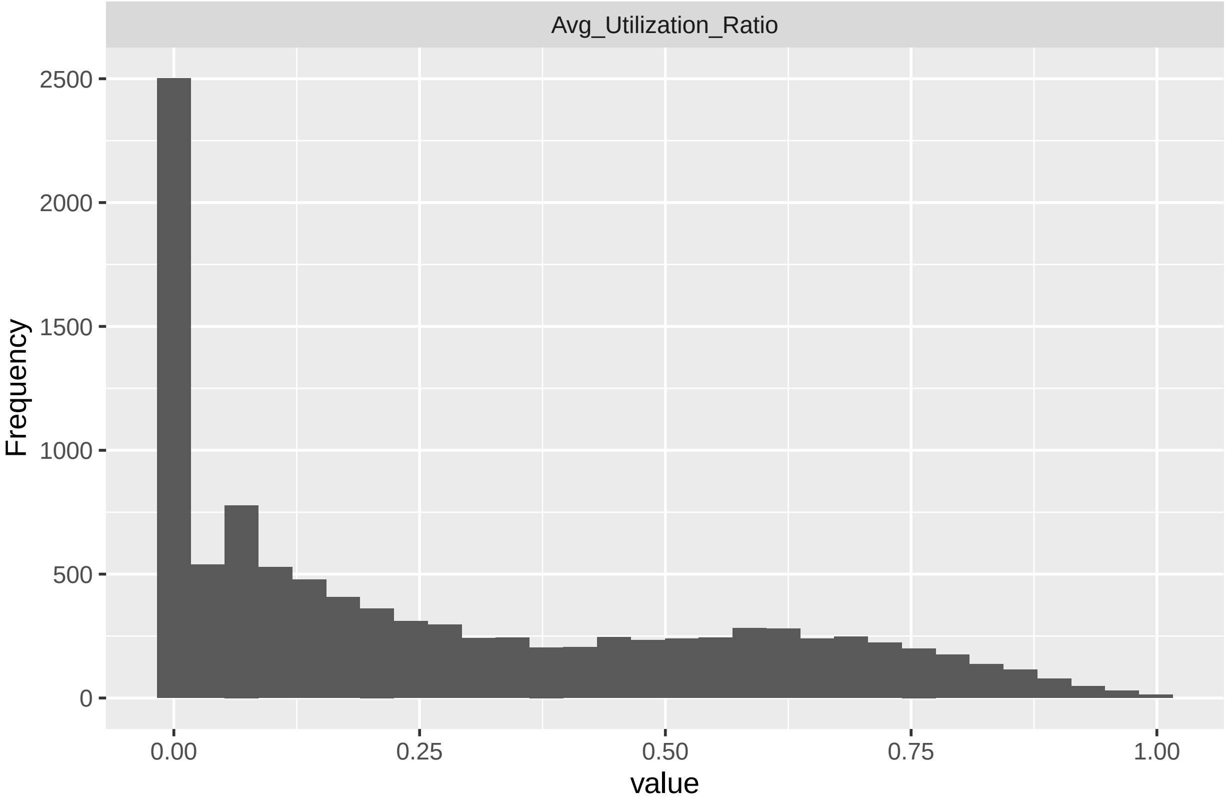


Histogram of numeric variables

```
plot_histogram(df)
```


Avg_Open_To_Buy
Card_Category
ontacts_Count_12_m
Credit_Limit
Customer_Age
Dependent_count
Education_Level
Income_Category
onths_Inactive_12_m
Months_on_book
tal_Amt_Chng_Q4_(
otal_Ct_Chng_Q4_Q
tal_Relationship_Cou
Total_Revolving_Bal
Total_Trans_Amt
Total_Trans_Ct
Frequency
value

Page 1

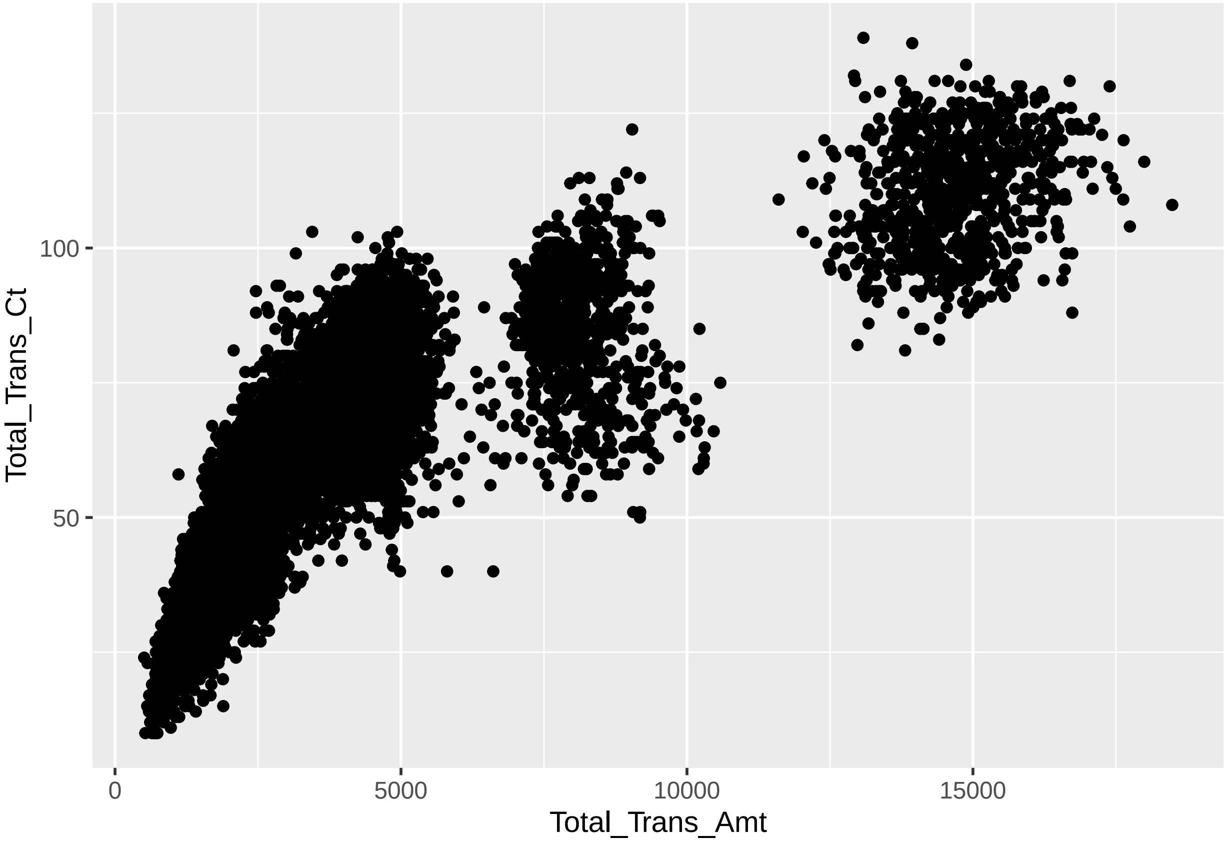

Avg_Utilization_Ratio
Frequency
2500
2000
1500
1000
500
0
0.00
0.25
0.50
0.75
1.00
value

Page 2

Data Exploration

Total Transaction Count vs. Total Transaction Amount

```
ggplot(df, aes(Total_Trans_Amt,Total_Trans_Ct)) + geom_point()
```



K-Means

Recipe

```
kmeans_recipe <- recipe(~Income_Category + Education_Level + Total_Trans_Amt + Total_Trans_Ct, data=df)
   step_normalize(all_numeric_predictors())
```

Model and Workflow

Model Creation

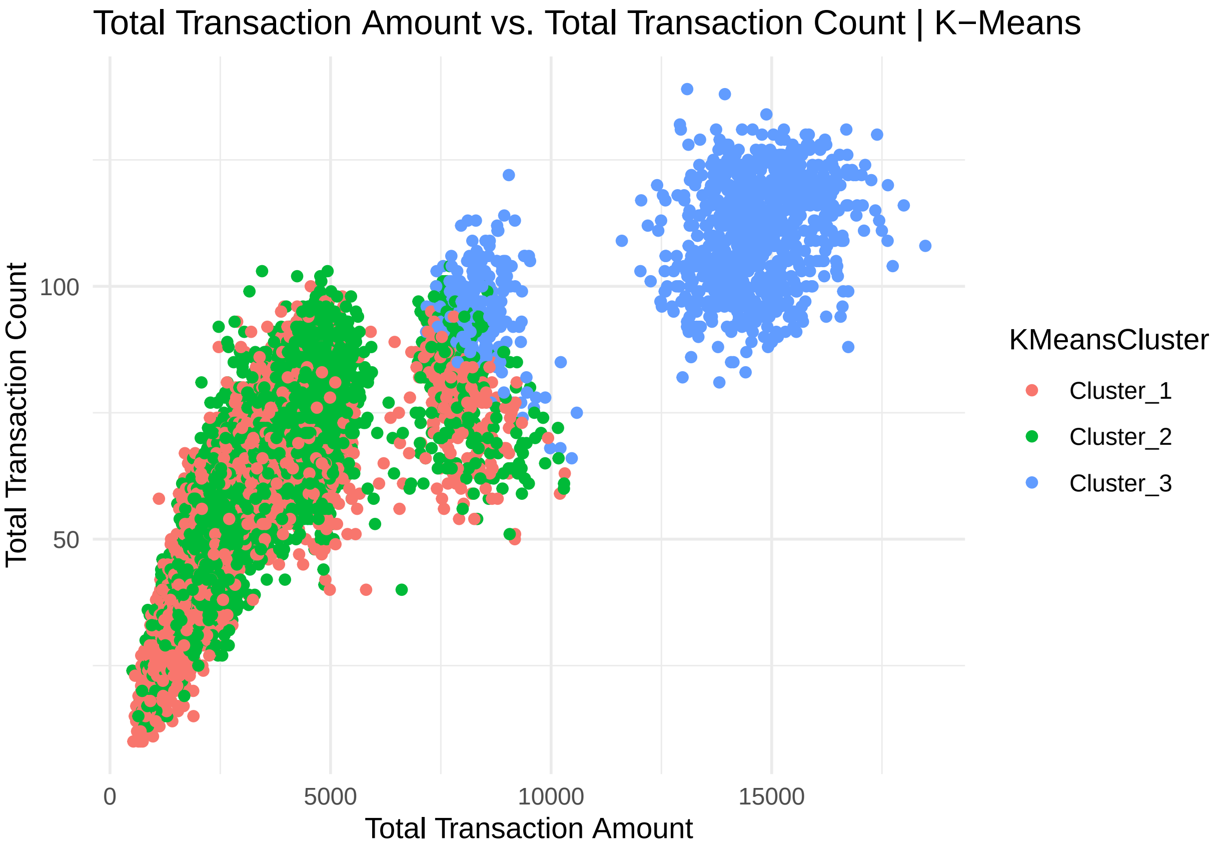
```
kmeans_model <- k_means(num_clusters = 3) |>
  set_engine("stats")
```

Workflow Creation and Data Fitting

```
set.seed(11)

kmeans_model <- k_means(num_clusters = 3) %>%
  set_engine("stats")

kmeans_workflow <- workflow() %>%
    add_recipe(kmeans_recipe) %>%
    add_model(kmeans_model) %>%
    fit(data = df)

kmeans_summary <- kmeans_workflow %>%
  extract_fit_summary()

df <- df |>
  mutate(KMeansCluster=kmeans_summary$cluster_assignments)
```

K-Means Clusters Visualization

```
ggplot(df, aes(x = Total_Trans_Amt, y = Total_Trans_Ct, color = KMeansCluster)) +
  geom_point() +
  labs(title = "Total Transaction Amount vs. Total Transaction Count | K-Means",
       x = "Total Transaction Amount",
       y = "Total Transaction Count") +
  theme_minimal()
```



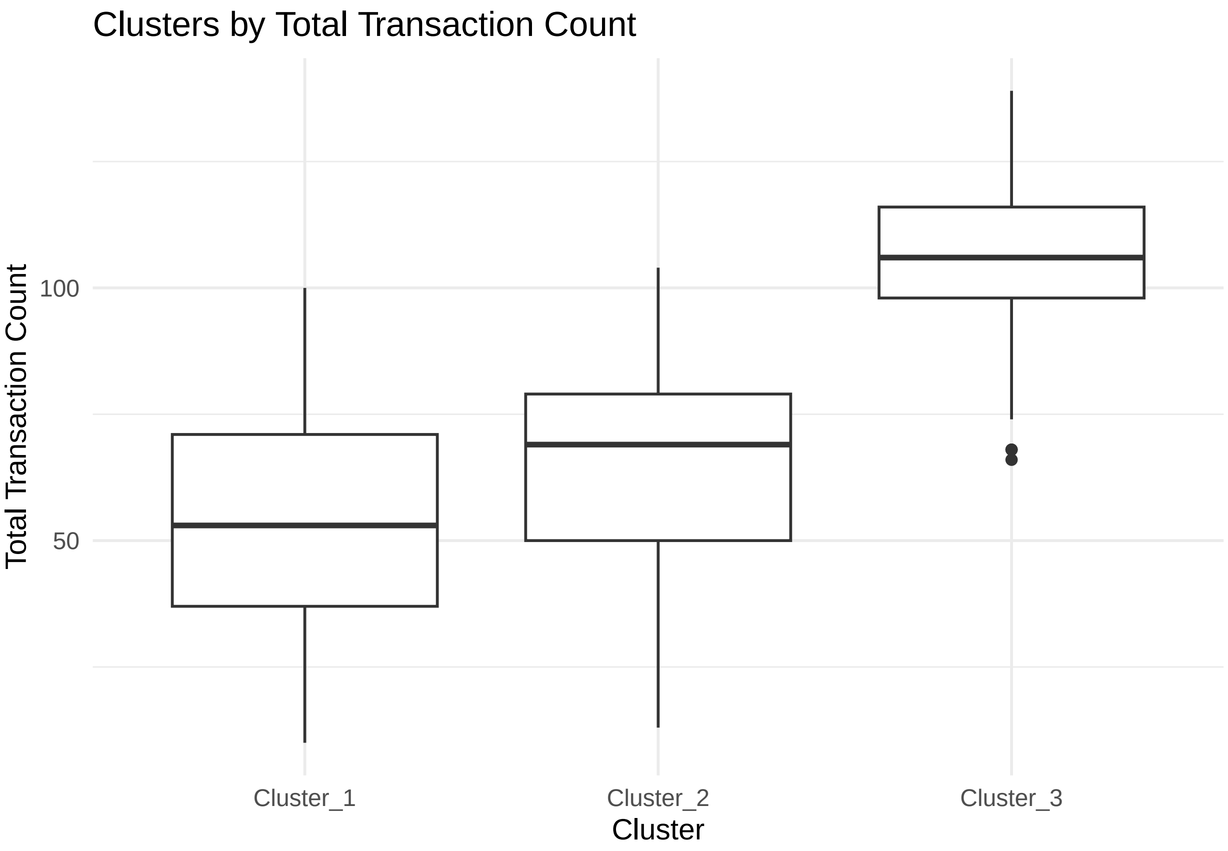
Summary Statistics by Cluster

```
df |>
  group_by(KMeansCluster) |>
  summarise(count=n(), across(where(is.numeric), mean)) |>
  select(KMeansCluster, Total_Trans_Ct, Total_Trans_Amt, Total_Revolving_Bal, count)
```

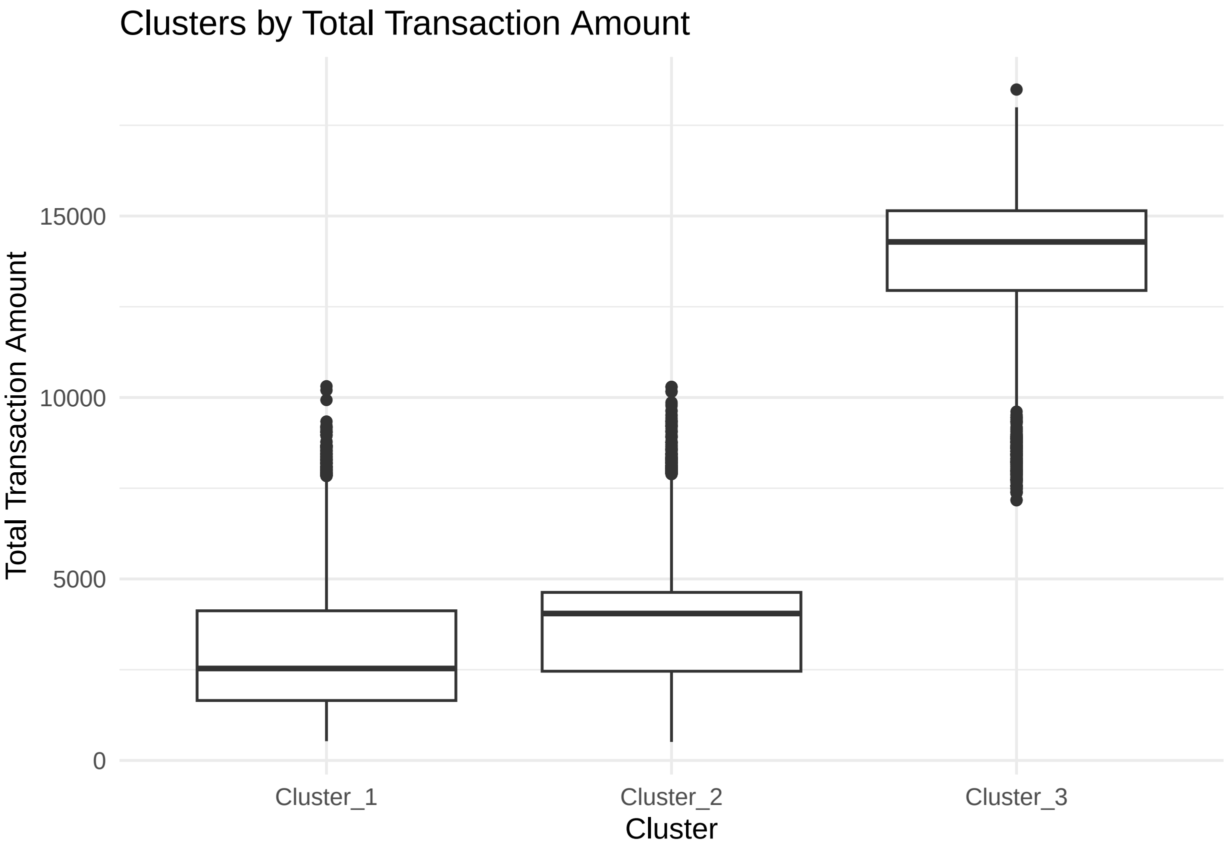
```
## # A tibble: 3 x 5
##   KMeansCluster Total_Trans_Ct Total_Trans_Amt Total_Revolving_Bal count
##   <fct>                  <dbl>           <dbl>               <dbl> <dbl>
## 1 Cluster_1               54.0           3042.               1159.  3604
## 2 Cluster_2               64.6           3754.               1130.  5568
## 3 Cluster_3              107.           13333.               1369.   955
```

Explore metrics by cluster

```
ggplot(df, aes(x = KMeansCluster, y = Total_Trans_Ct)) +
  geom_boxplot() +
  labs(title = "Clusters by Total Transaction Count", x = "Cluster", y = "Total Transaction Count") +
  theme_minimal()
```



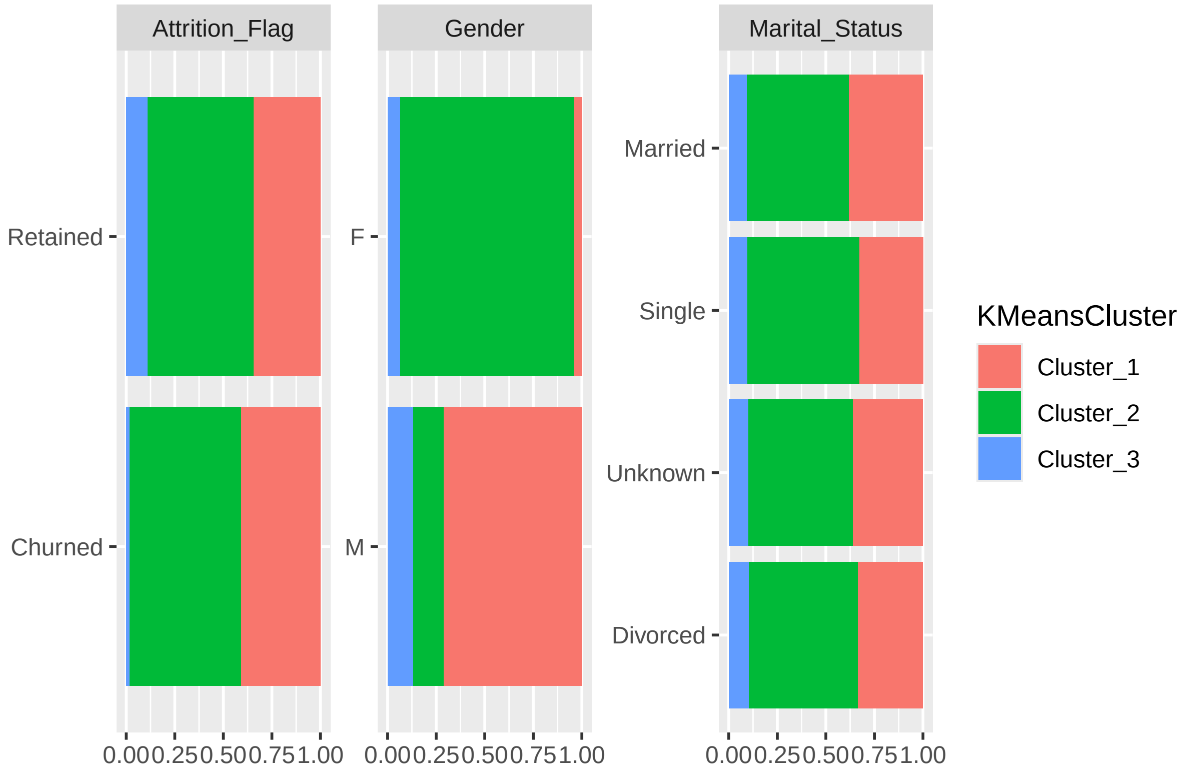
```
ggplot(df, aes(x = KMeansCluster, y = Total_Trans_Amt)) +
  geom_boxplot() +
  labs(title = "Clusters by Total Transaction Amount", x = "Cluster", y = "Total Transaction Amount") +
  theme_minimal()
```



Cluster Distribution by Categorical Variables

```r
plot_bar(df, by = "KMeansCluster")
```

```
## 1 columns ignored with more than 50 categories.
## CLIENTNUM: 10127 categories
```

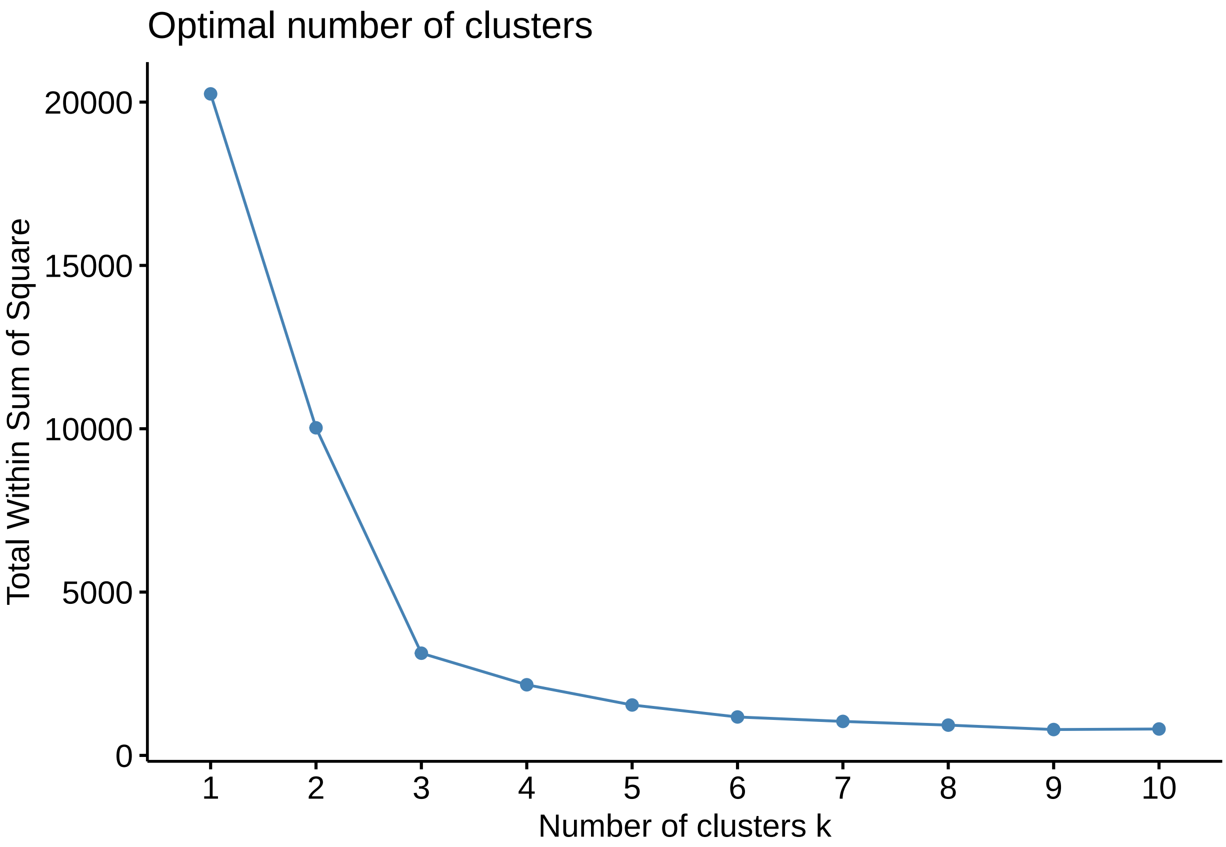


Elbow Method

Data Preparation

```r
mini_df <- df |>
  select(Total_Trans_Amt, Total_Trans_Ct) |>
  mutate(across(where(is.numeric), scale))
```

```r
fviz_nbclust(mini_df, kmeans, method = "wss") # weighted sum of squares / elbow
```



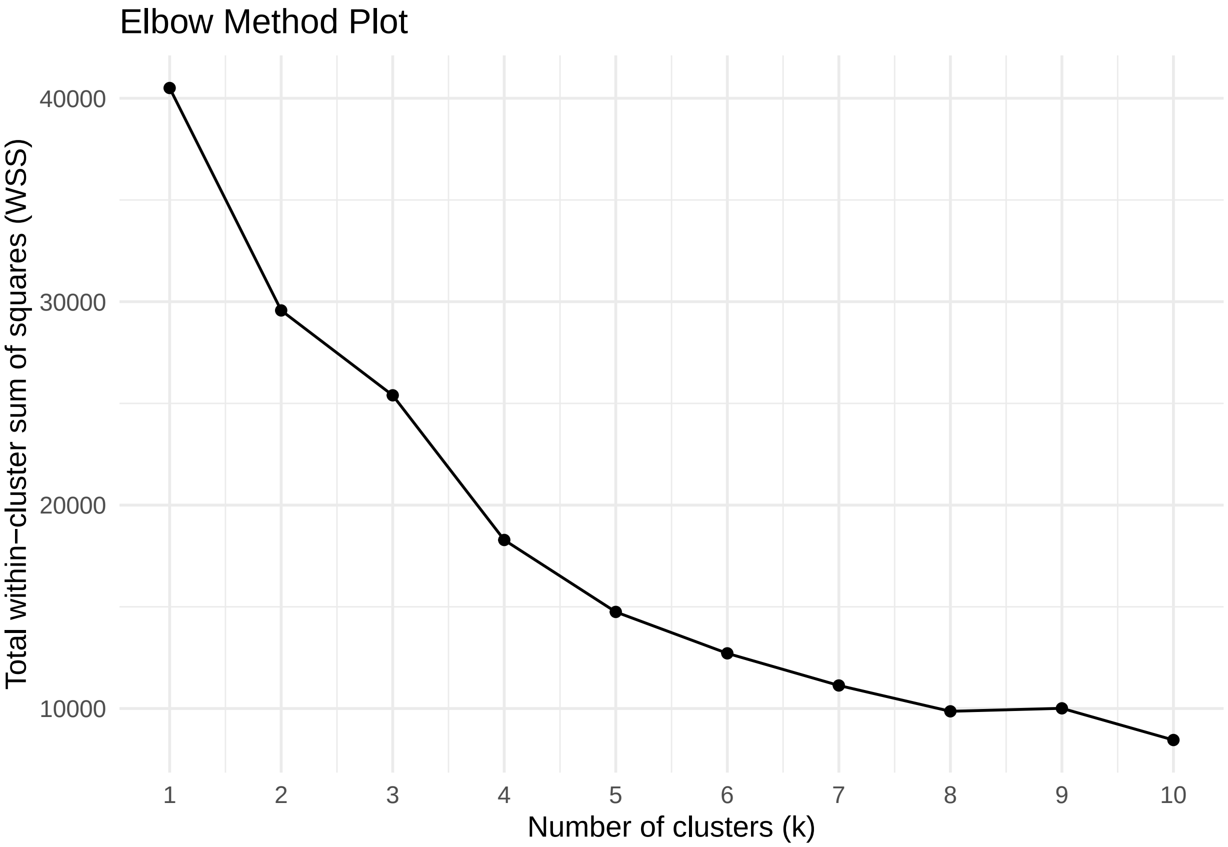
Elbow Method (Manual)

```
max_k = 10
wss = numeric(max_k)

for(k in 1:max_k) {
  kmeans_model <- k_means(num_clusters = k) %>%
  set_engine("stats")

  fit_workflow <- workflow() %>%
      add_recipe(kmeans_recipe) %>%
      add_model(kmeans_model) %>%
      fit(data = df) %>%
      extract_fit_parsnip()

  wss[k] <- fit_workflow$fit$tot.withinss
}
```

Elbow Plot

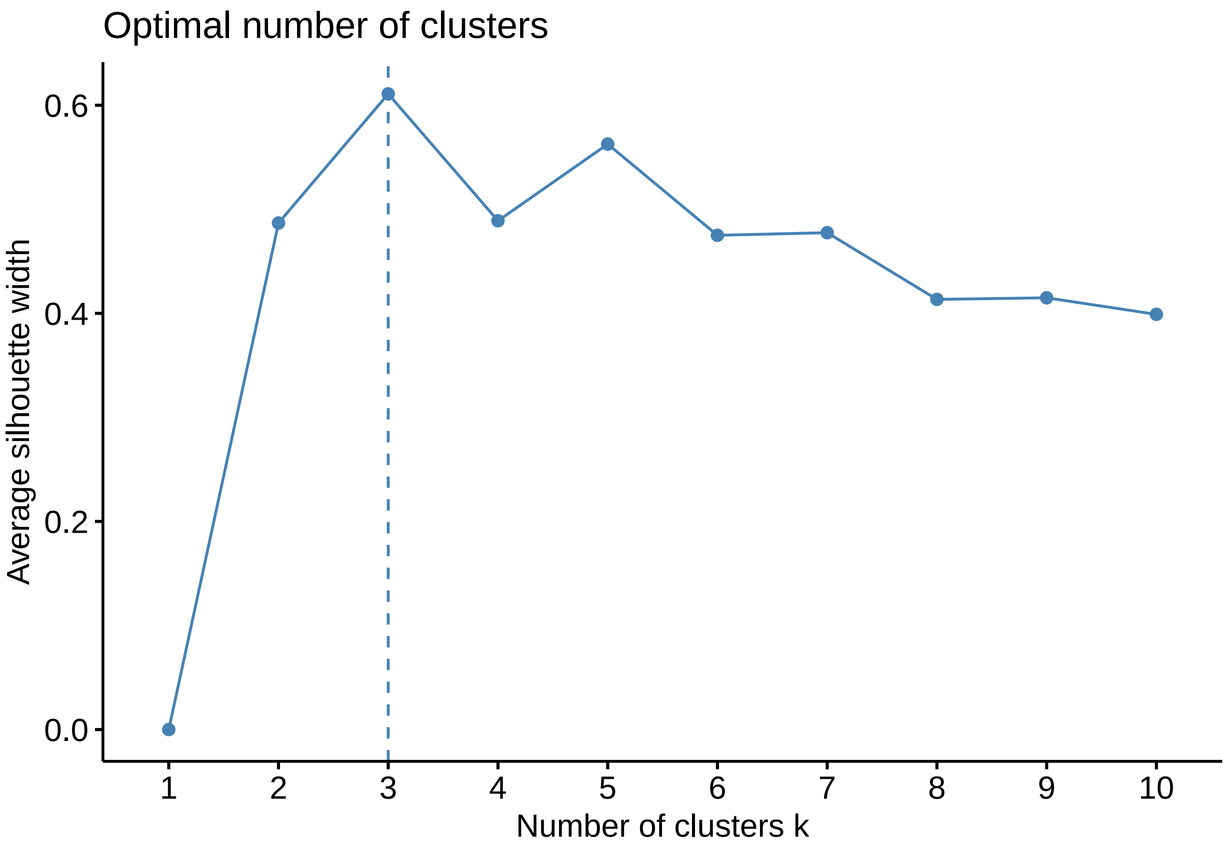
```
elbow_data <- data.frame(
  k = 1:length(wss),
  wss = wss
)
```

```
ggplot(elbow_data, aes(x = k, y = wss)) +
  geom_line() +
  geom_point() +
  scale_x_continuous(breaks = seq(1, max_k, 1)) +
  labs(title = 'Elbow Method Plot',
       x = 'Number of clusters (k)',
       y = 'Total within-cluster sum of squares (WSS)') +
  theme_minimal()
```



Method 2: Silhouette Method

```
fviz_nbclust(mini_df, kmeans, method='silhouette')
```



Save clusters to CSV file

```
write_csv(df, "credit_card_customers_kmeans.csv")
```

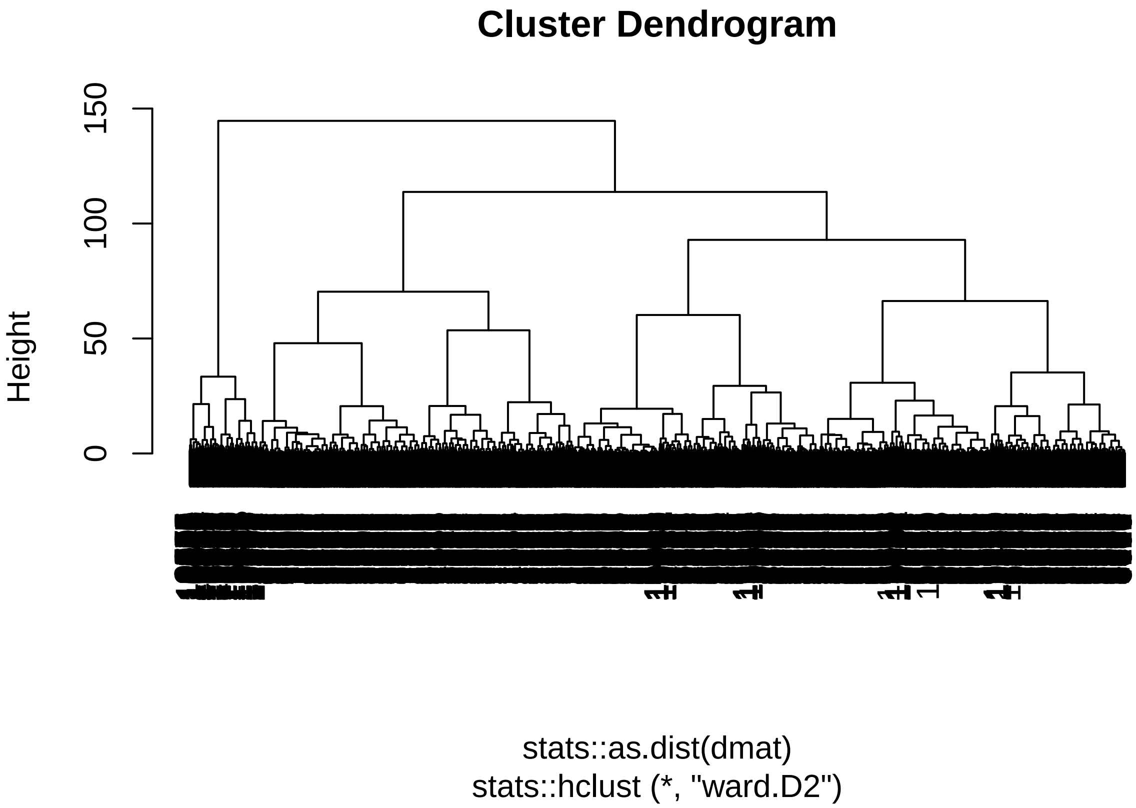
```
cluster_recipe <- recipe(~Income_Category + Education_Level + Total_Trans_Amt + Total_Trans_Ct, data=df
  step_normalize(all_numeric_predictors())
```

Hierarchical Clustering

```
hc_model <- hier_clust(linkage_method = "ward.D2")

hc_workflow <- workflow() |>
    add_recipe(cluster_recipe) |>
    add_model(hc_model) |>
    fit(data = df)
```

Plot the dendrogram

```
hc_workflow |> extract_fit_engine() |> plot()
```



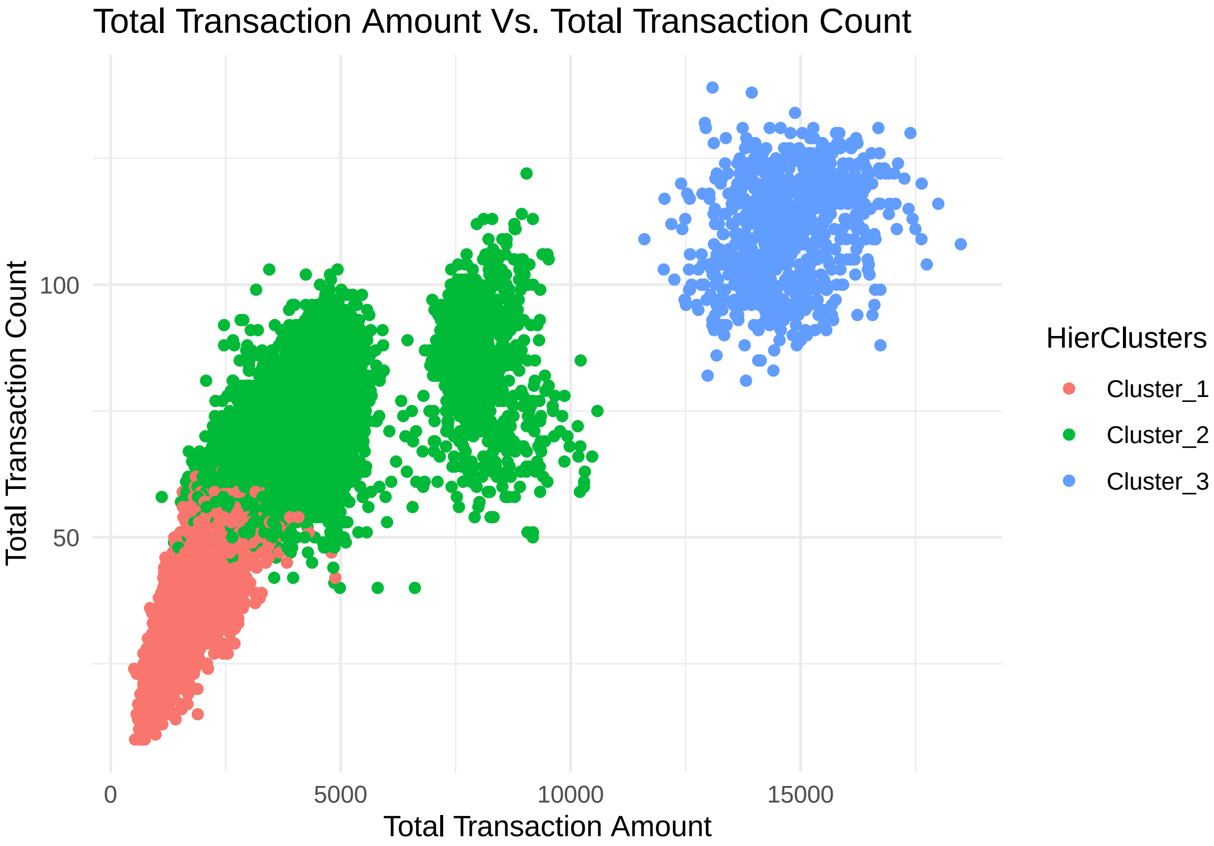
```
hc_summary <- hc_workflow |>
  extract_fit_summary(num_clusters=3)

hc_summary |> str()
```

```
## List of 7
##  $ cluster_names          : Factor w/ 3 levels "Cluster_1","Cluster_2",..: 1 2 3
##  $ centroids              : tibble [3 x 4] (S3: tbl_df/tbl/data.frame)
##   ..$ Income_Category: num [1:3] 0.0869 -0.0735 0.1897
##   ..$ Education_Level: num [1:3] 0.00492 -0.00137 -0.01154
##   ..$ Total_Trans_Amt: num [1:3] -0.7523 0.0509 3.0303
##   ..$ Total_Trans_Ct : num [1:3] -1.121 0.399 1.937
##  $ n_members              : int [1:3] 3413 5967 747
##  $ sse_within_total_total: num [1:3] 4803 8699 1084
##  $ sse_total              : num 18643
##  $ orig_labels            : NULL
##  $ cluster_assignments    : Factor w/ 3 levels "Cluster_1","Cluster_2",..: 1 1 1 1 1 1 1 1 1 1 ...
```

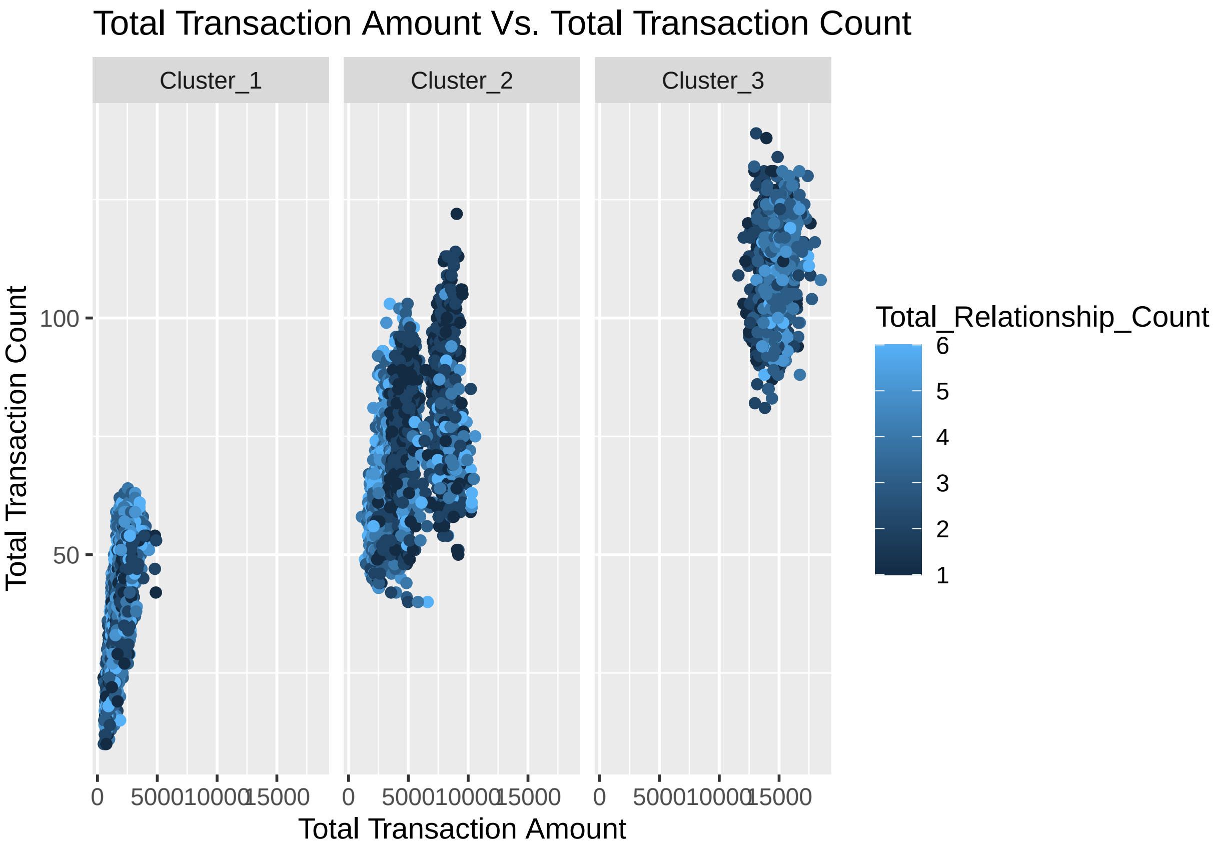
```
fit_hc <- hc_workflow |> extract_fit_engine()

df$HierClusters <- cutree(fit_hc, k = 3)
df$HierClusters <- factor(paste("Cluster_", df$HierClusters, sep = ""), ordered = FALSE)
```

Plot the Hierarchical Clusters

```
ggplot(df, aes(x = Total_Trans_Amt, y = Total_Trans_Ct, color = HierClusters)) +
  geom_point() +
  labs(title = "Total Transaction Amount Vs. Total Transaction Count",
       x = "Total Transaction Amount",
       y = "Total Transaction Count") +
  theme_minimal()
```



```
ggplot(df, aes(x = Total_Trans_Amt, y = Total_Trans_Ct, color=Total_Relationship_Count)) +
  geom_point() +
  facet_wrap(~ HierClusters) +
  labs(title = "Total Transaction Amount Vs. Total Transaction Count",
       x = "Total Transaction Amount",
       y = "Total Transaction Count")
```



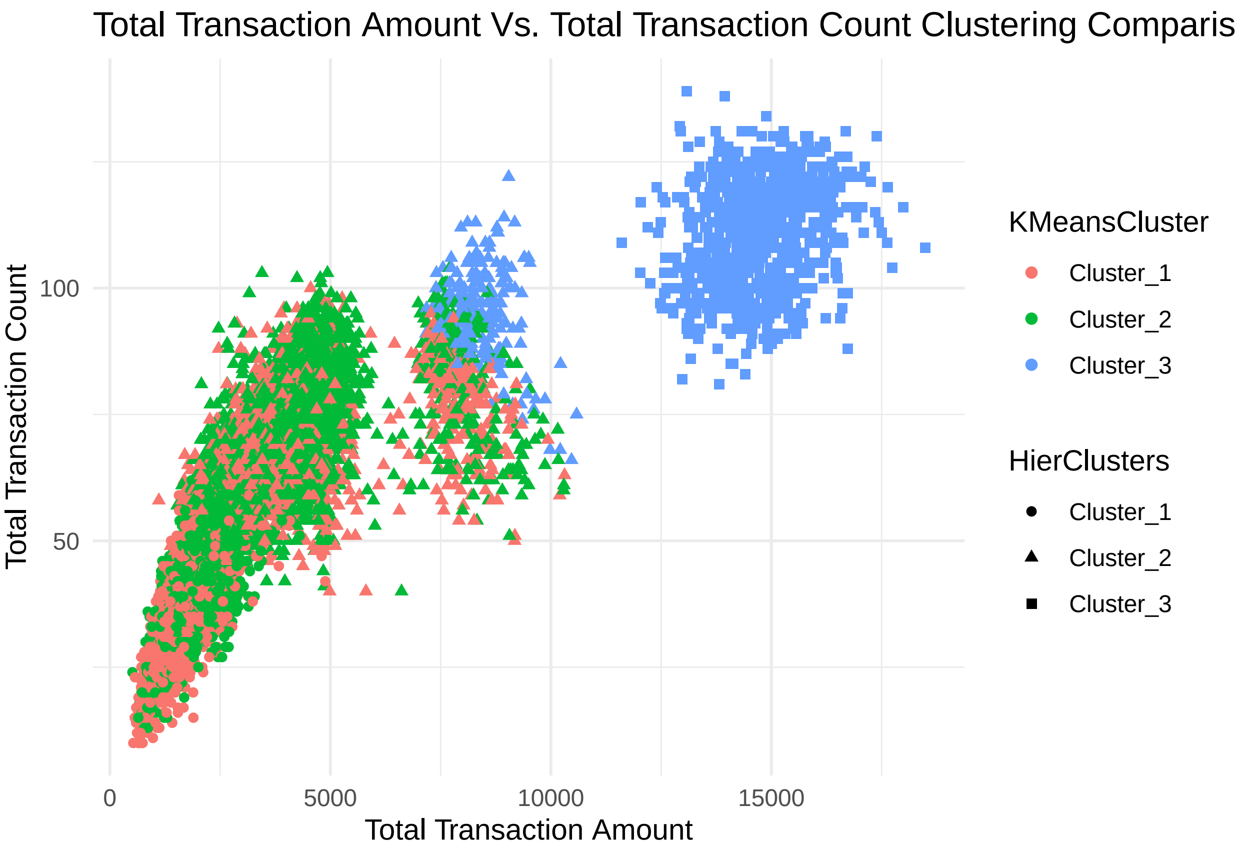
Comparing KMeans and Hierarchical Clusters

```
conf_mat(df, truth = KMeansCluster,
         estimate = HierClusters)
```

```
##             Truth
## Prediction  Cluster_1 Cluster_2 Cluster_3
##   Cluster_1      1800      1613         0
##   Cluster_2      1804      3955       208
##   Cluster_3         0         0       747
```

Plot Hierarchical Clustering vs. K-Means

```
ggplot(df, aes(x = Total_Trans_Amt, y = Total_Trans_Ct, color = KMeansCluster, shape=HierClusters)) +
  geom_point() +
  labs(title = "Total Transaction Amount Vs. Total Transaction Count Clustering Comparison | K-Means vs
       x = "Total Transaction Amount",
       y = "Total Transaction Count") +
  theme_minimal()
```


Total Transaction Amount Vs. Total Transaction Count Clustering Comparis
Total Transaction Count
100
50
0
5000
10000
15000
Total Transaction Amount
KMeansCluster
Cluster_1
Cluster_2
Cluster_3
HierClusters
● Cluster_1
▲ Cluster_2
■ Cluster_3