NSDC_Spring_2025_Data_Exploration

Elias Grant & Lea Chloe

2025-04-30

Load and Clean Data

```
# Load libraries
#install.packages("tidyverse")
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.2

## Warning: package 'ggplot2' was built under R version 4.4.2

## Warning: package 'tibble' was built under R version 4.4.2

## Warning: package 'tidyr' was built under R version 4.4.2

## Warning: package 'dplyr' was built under R version 4.4.2

## Warning: package 'lubridate' was built under R version 4.4.2

## -- 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   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     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
```

```
library(ggplot2)
library(dplyr)

# Read the dataset
df <- read.csv("C:/Users/elias/Downloads/NSDC/users_data.csv", stringsAsFactors = FALSE)

# Clean currency columns
df$per_capita_income <- as.numeric(gsub("[\\$,]", "", df$per_capita_income))
df$yearly_income <- as.numeric(gsub("[\\$,]", "", df$yearly_income))
df$total_debt <- as.numeric(gsub("[\\$,]", "", df$total_debt))

#View(df)
```

Summary statistics

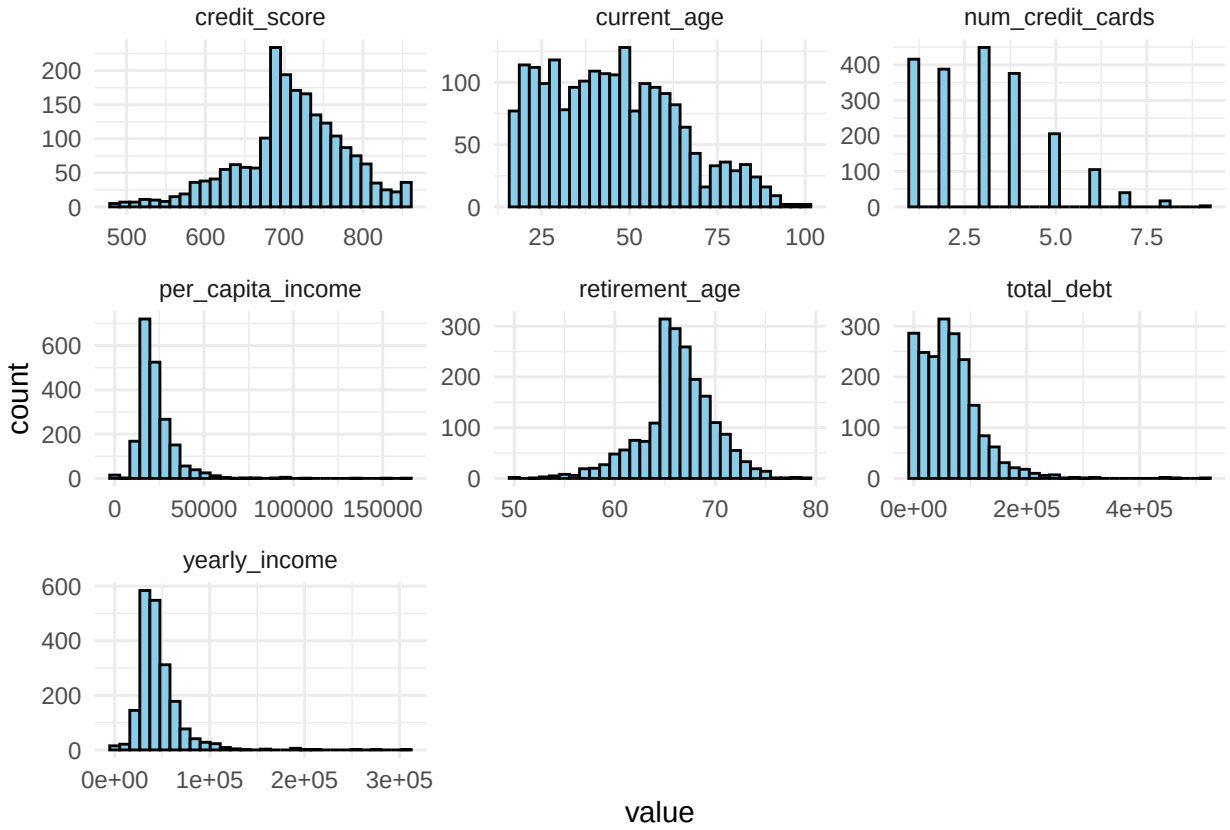
```
# Summary for numeric columns
summary(df[sapply(df, is.numeric)])
```

```
##        id          current_age     retirement_age    birth_year
##  Min.   :   0.0   Min.   : 18.00   Min.   :50.00   Min.   :1918
##  1st Qu.: 499.8   1st Qu.: 30.00   1st Qu.:65.00   1st Qu.:1961
##  Median : 999.5   Median : 44.00   Median :66.00   Median :1975
##  Mean   : 999.5   Mean   : 45.39   Mean   :66.24   Mean   :1974
##  3rd Qu.:1499.2   3rd Qu.: 58.00   3rd Qu.:68.00   3rd Qu.:1989
##  Max.   :1999.0   Max.   :101.00   Max.   :79.00   Max.   :2002
##   birth_month        latitude       longitude       per_capita_income
##  Min.   : 1.000   Min.   :20.88   Min.   :-159.41   Min.   :     0
##  1st Qu.: 3.000   1st Qu.:33.84   1st Qu.: -97.39   1st Qu.: 16825
##  Median : 7.000   Median :38.25   Median : -86.44   Median : 20581
##  Mean   : 6.439   Mean   :37.39   Mean   : -91.55   Mean   : 23142
##  3rd Qu.:10.000   3rd Qu.:41.20   3rd Qu.: -80.13   3rd Qu.: 26286
##  Max.   :12.000   Max.   :61.20   Max.   : -68.67   Max.   :163145
##  yearly_income      total_debt      credit_score   num_credit_cards
##  Min.   :     1   Min.   :     0   Min.   :480.0   Min.   :1.000
##  1st Qu.: 32819   1st Qu.: 23987   1st Qu.:681.0   1st Qu.:2.000
##  Median : 40745   Median : 58251   Median :711.5   Median :3.000
##  Mean   : 45716   Mean   : 63710   Mean   :709.7   Mean   :3.073
##  3rd Qu.: 52699   3rd Qu.: 89071   3rd Qu.:753.0   3rd Qu.:4.000
##  Max.   :307018   Max.   :516263   Max.   :850.0   Max.   :9.000
```

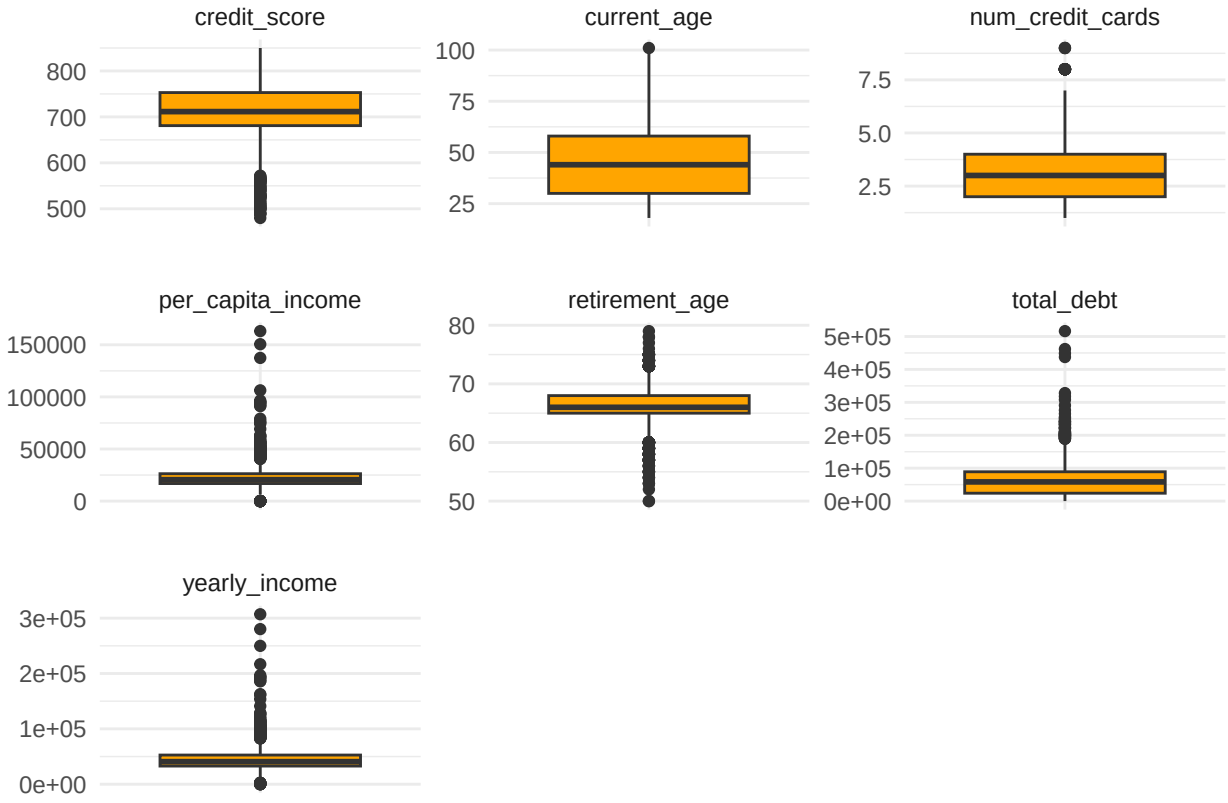
```
#View(summary(df[sapply(df, is.numeric)]))
```

Histograms and Boxplots

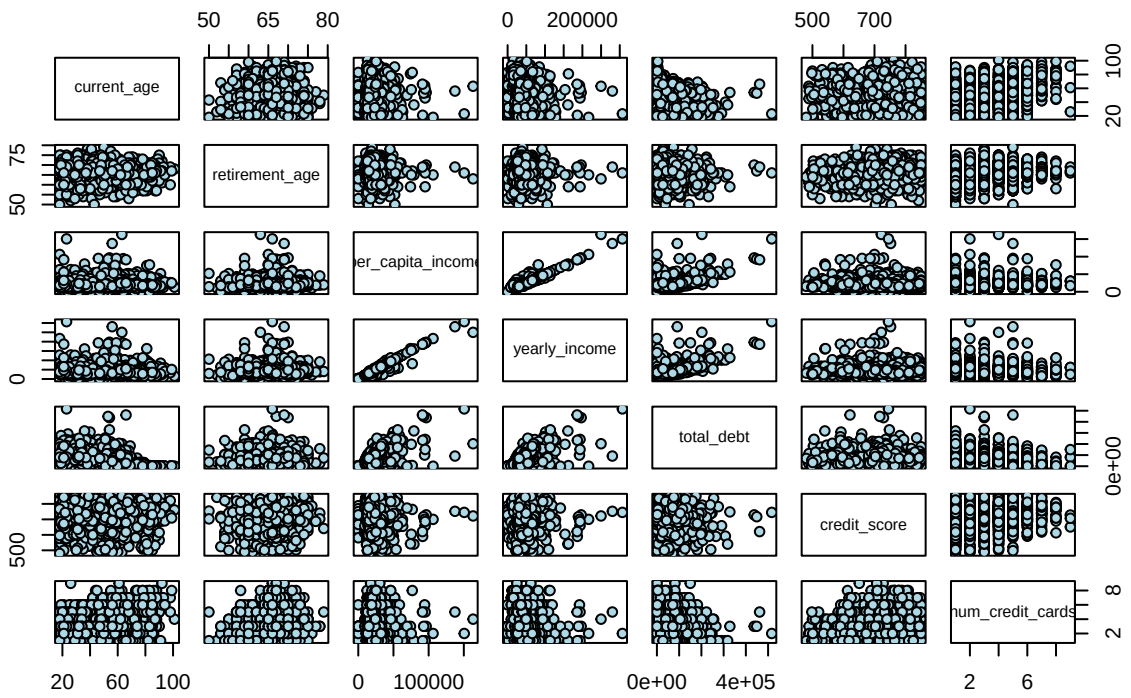
```
# Select relevant numeric columns
num_vars <- c("current_age", "retirement_age", "per_capita_income",
              "yearly_income", "total_debt", "credit_score", "num_credit_cards")

# Gather data for plotting
df_long <- df |>
  select(all_of(num_vars)) |>
  pivot_longer(everything(), names_to = "variable", values_to = "value")

# Histograms
ggplot(df_long, aes(x = value)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  facet_wrap(~ variable, scales = "free") +
  theme_minimal()
```




```
ggplot(df_long, aes(x = "", y = value)) +
  geom_boxplot(fill = "orange") +
  facet_wrap(~ variable, scales = "free") +
  labs(x = NULL, y = NULL) +
  theme_minimal()
```




```
# Basic scatterplot matrix
pairs(df[, num_vars], pch = 21, bg = "lightblue", main = "Scatterplot Matrix")
```




```
#close up version
#install.packages("GGally")
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.4.3
```

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

```
# Create the plot
plot_matrix <- ggpairs(df[, c("credit_score", "total_debt", "current_age")])

# Save with larger dimensions (in inches)
ggsave("scatterplot_matrix.png", plot_matrix, width = 12, height = 12, dpi = 300)
```

Variables by Usefulness

```
colnames(df) # all variables
```

```
##  [1] "id"                "current_age"       "retirement_age"
##  [4] "birth_year"        "birth_month"       "gender"
```

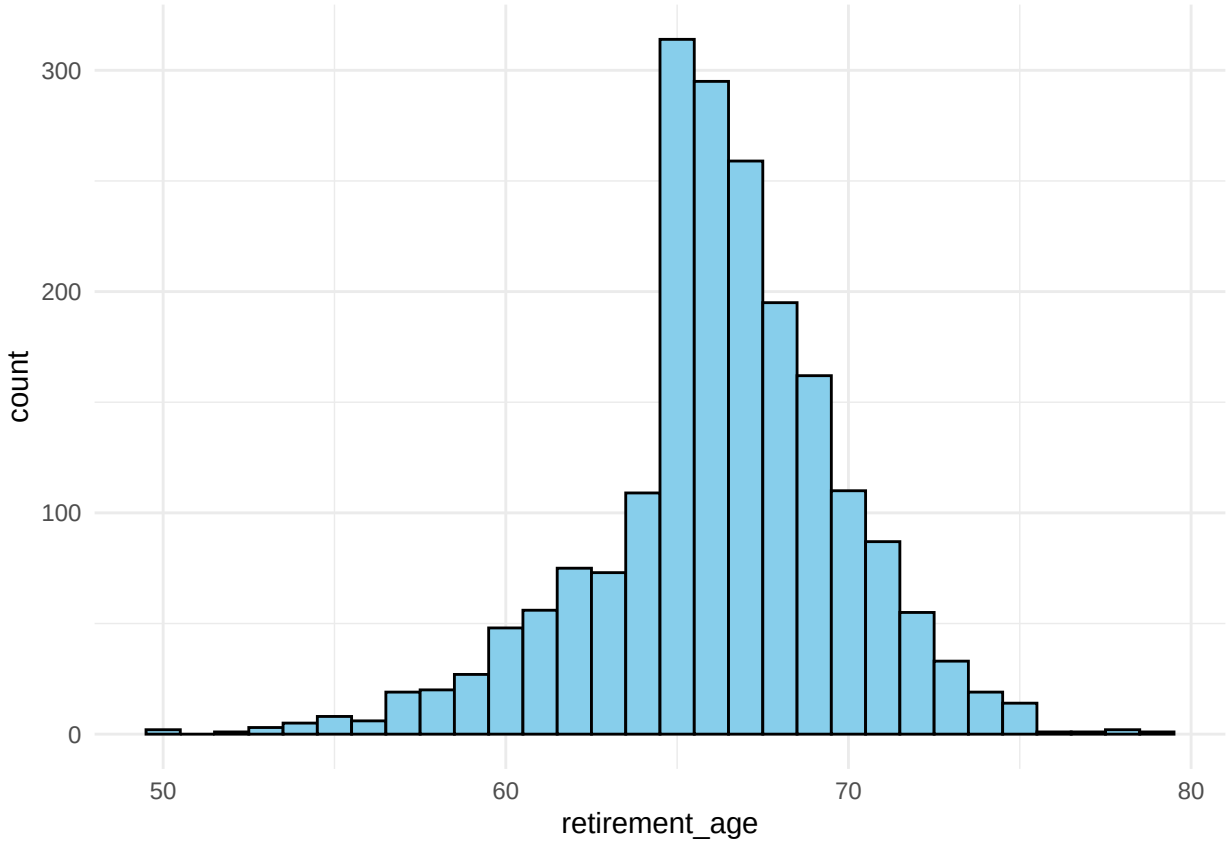
```
##  [7] "address"           "latitude"          "longitude"
## [10] "per_capita_income" "yearly_income"     "total_debt"
## [13] "credit_score"      "num_credit_cards"
```

Useful	Not sure	Useless
`current_age`	`address` -> can we get state/region from this?	`id`
`retirement_age`	`latitude` and longitude -> can we get state/region from this?	`birth_month`
`gender`	`num_credit_cards`	`birth_year` (because we already have current_age)
`per_capita_income`		
`yearly_income`		
`total_debt`		
`credit_score`		

Basic observations

- current_age
  - range: 18 to 101 years old
  - a lot between 18-50 years old, goes downward from there
  - median: 44 years old
- retirement_age
  - **Question: how is retirement_age calculated? location?**
  - range: 50 to 79 years old
  - median: 66 years old
  - huge leap between 64 (109) and 65 years old (314)
  - skewed left, meaning there's more on the right of 66 years old.

```
ggplot(df, aes(x = retirement_age)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "black") +
  theme_minimal()
```




```
df |> filter(retirement_age == 64) |> nrow()
```

```
## [1] 109
```

```
df |> filter(retirement_age == 65) |> nrow()
```

```
## [1] 314
```

```
df |> filter(retirement_age <= 64) |> nrow()
```

```
## [1] 452
```

```
df |> filter(retirement_age >= 65) |> nrow()
```

```
## [1] 1548
```

- gender
  - 984 men and 1016 women, pretty even.
  - gender diferences?

```
df |> filter(gender == "Male") |> nrow() # 984 men
```

```
## [1] 984
```

```
df |> filter(gender == "Female") |> nrow() # 1016 women
```

```
## [1] 1016
```

Gender differences are miniscule.

- not by a lot, but women have a higher median for per capita income and yearly income.
- men have a higher median for total debt, credit score

```
variables_to_compare <- c("retirement_age", "per_capita_income", "yearly_income", "total_debt", "credit_

median_table <- df %>%
  group_by(gender) %>%
  summarize(across(all_of(variables_to_compare), ~median(.x, na.rm = TRUE))) %>%
  pivot_longer(cols = -gender, names_to = "Variable", values_to = "Median") %>%
  pivot_wider(names_from = gender, values_from = Median)

# View table
print(median_table)
```

```
## # A tibble: 6 x 3
##   Variable          Female   Male
##   <chr>              <dbl>  <dbl>
## 1 retirement_age       66     66
## 2 per_capita_income 20676. 20447
## 3 yearly_income     40858. 40500
## 4 total_debt        57212. 59030.
## 5 credit_score        710    712
## 6 num_credit_cards      3      3
```

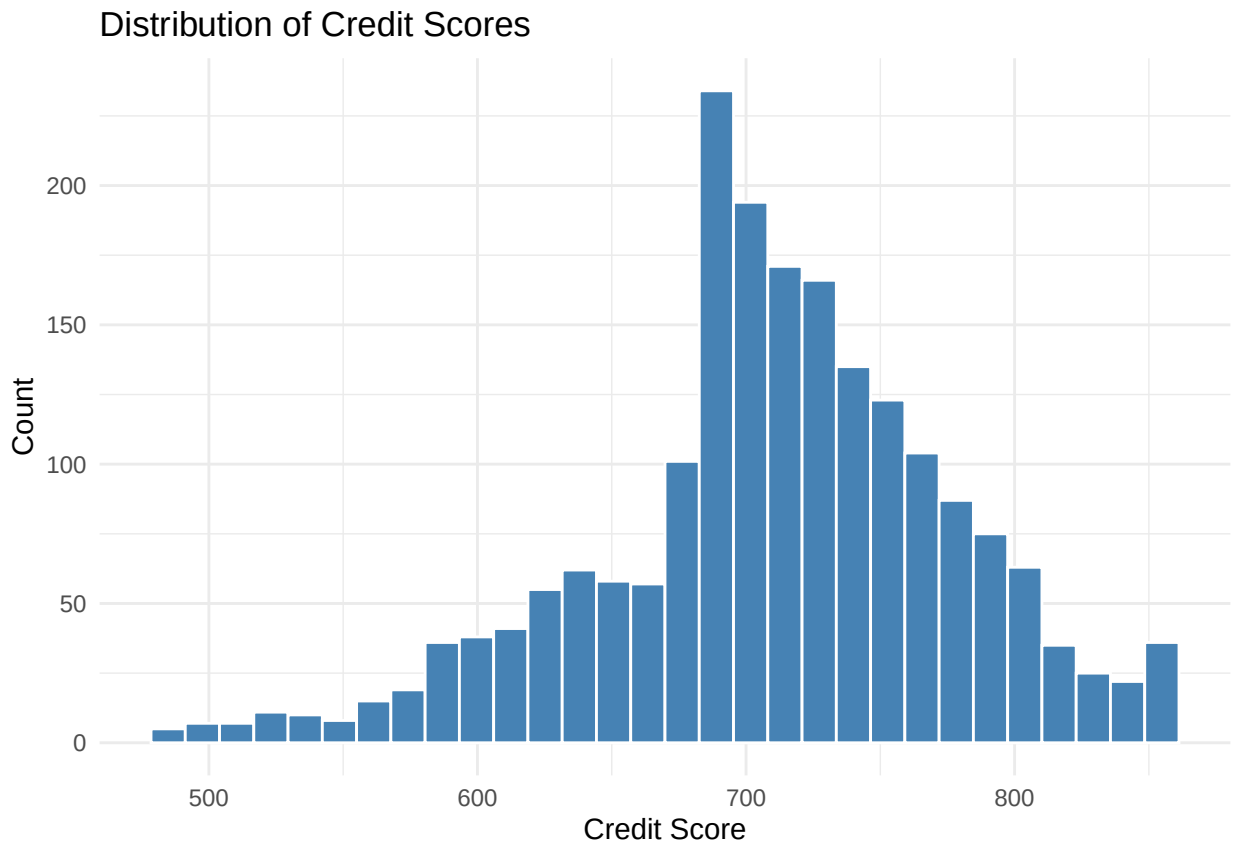
```
# Summary statistics
# View(summary(df_women[sapply(df, is.numeric)]))
# View(summary(df_men[sapply(df, is.numeric)]))
```

- per_capita_income
  - skewed right, median of $20,581
- yearly_income
  - similarly skewed right, median $40,744
- total_debt
  - median: $58,251, **much higher than median for yearly_income**

Most importantly,

- credit_score
  - *a credit score range is 300 to 850
    - range of dataset: 480 to 850
    - median = 711.5 (Good)

```
# histogram
ggplot(df, aes(x = credit_score)) +
  geom_histogram(bins = 30, fill = "steelblue", color = "white") +
  labs(title = "Distribution of Credit Scores",
       x = "Credit Score",
       y = "Count") +
  theme_minimal()
```

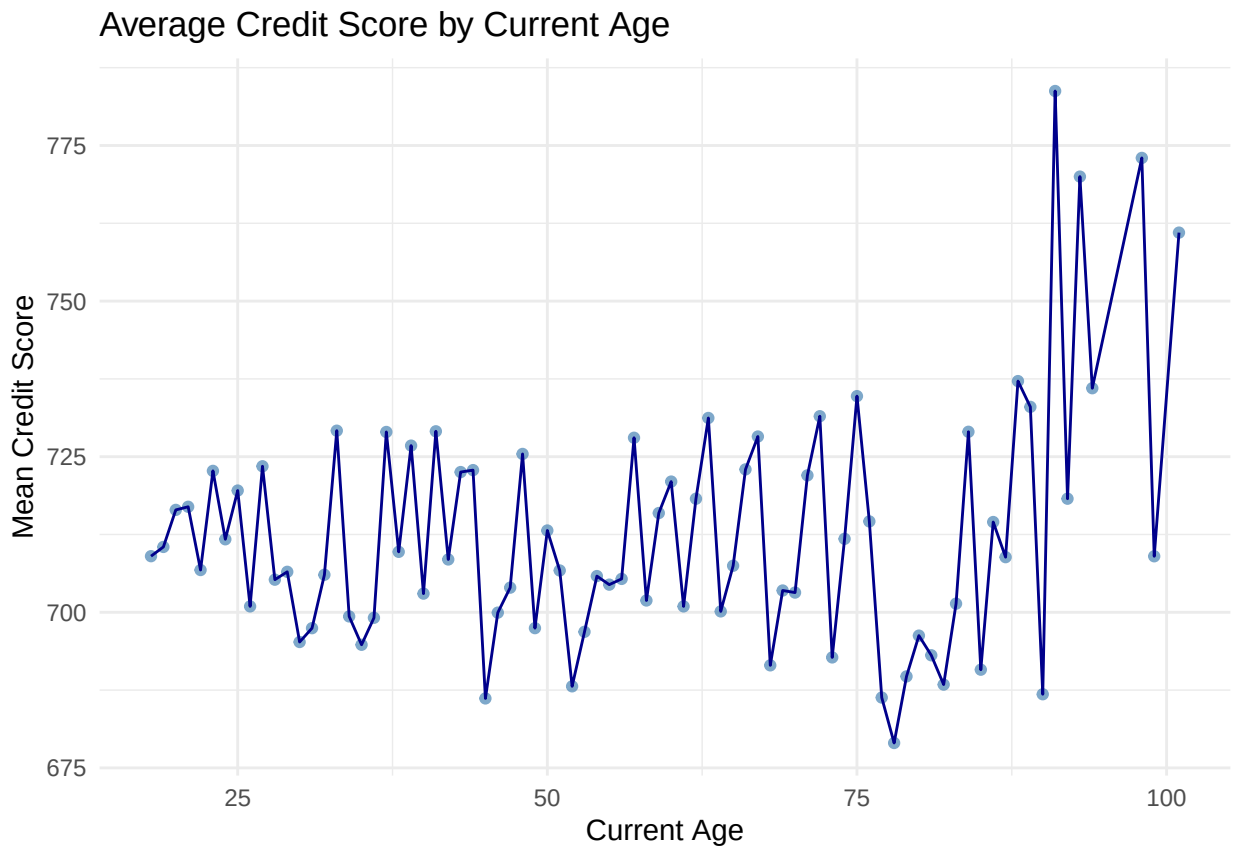



```
summary(df$credit_score)
```

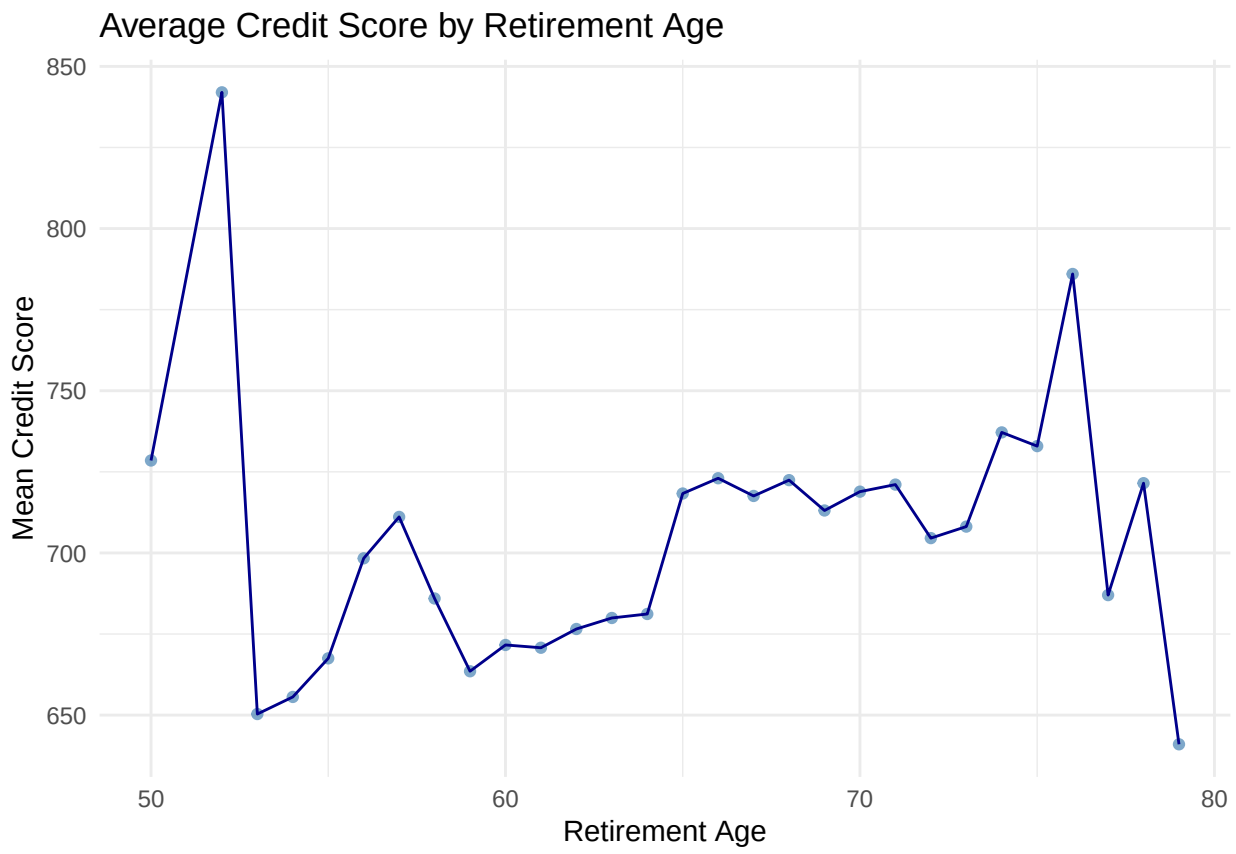
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   480.0   681.0   711.5   709.7   753.0   850.0
```

```
# CREDIT SCORE VS CURRENT AGE
# Summarize: mean credit score per current age
df_summary <- df %>%
  group_by(current_age) %>%
  summarize(mean_credit_score = mean(credit_score, na.rm = TRUE))
```

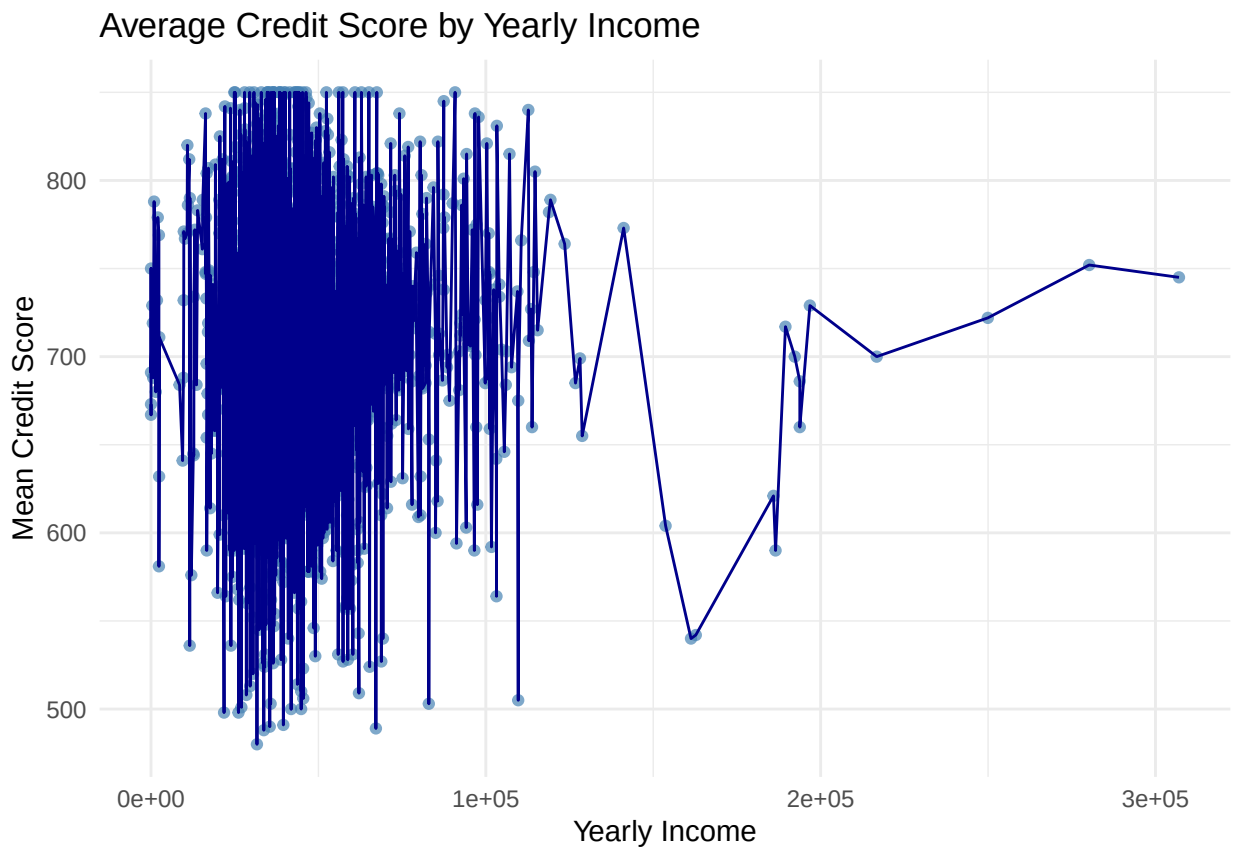
```r
# Now plot
ggplot(df_summary, aes(x = current_age, y = mean_credit_score)) +
  geom_point(alpha = 0.7, color = "steelblue") +
  geom_line(color = "darkblue") +  # optional: connects points
  labs(title = "Average Credit Score by Current Age",
       x = "Current Age",
       y = "Mean Credit Score") +
  theme_minimal()
```




```r
# CREDIT SCORE VS RETIREMENT AGE
# Summarize: mean credit score per retirement age
df_summary <- df %>%
  group_by(retirement_age) %>%
  summarize(mean_credit_score = mean(credit_score, na.rm = TRUE))

# Now plot
ggplot(df_summary, aes(x = retirement_age, y = mean_credit_score)) +
  geom_point(alpha = 0.7, color = "steelblue") +
  geom_line(color = "darkblue") +  # optional: connects points
  labs(title = "Average Credit Score by Retirement Age",
       x = "Retirement Age",
       y = "Mean Credit Score") +
  theme_minimal()
```




```r
# CREDIT SCORE VS YEARLY_INCOME
# Summarize: mean credit score per total debt
df_summary <- df %>%
  group_by(yearly_income) %>%
  summarize(mean_credit_score = mean(credit_score, na.rm = TRUE))

# Now plot
ggplot(df_summary, aes(x = yearly_income, y = mean_credit_score)) +
  geom_point(alpha = 0.7, color = "steelblue") +
  geom_line(color = "darkblue") +  # optional: connects points
  labs(title = "Average Credit Score by Yearly Income",
       x = "Yearly Income",
       y = "Mean Credit Score") +
  theme_minimal()
```



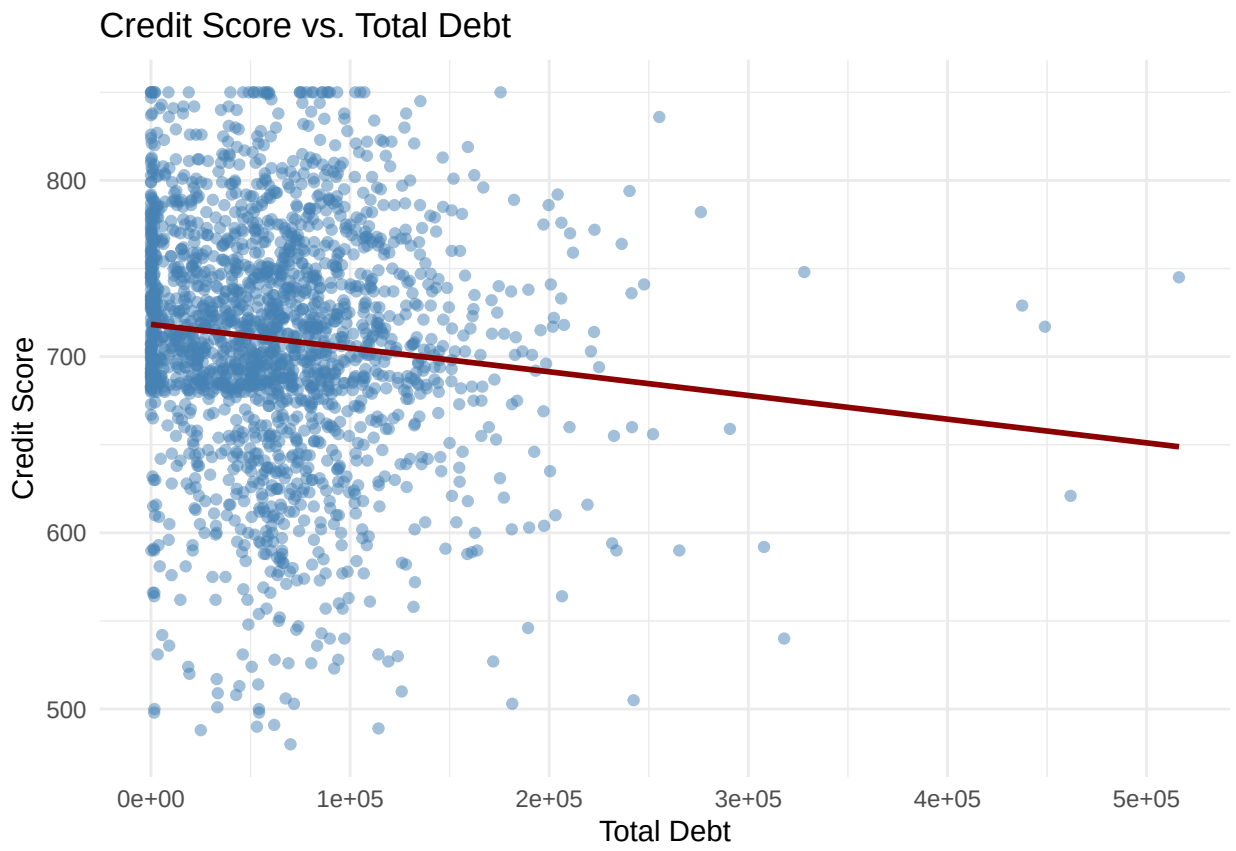
Interesting... there's no correlation between total debt and your credit score.

```
# Calculate the correlation coefficient
correlation <- cor(df$total_debt, df$credit_score, use = "complete.obs")
print(paste("Correlation between total debt and credit score:", round(correlation, 2)))
```

```
## [1] "Correlation between total debt and credit score: -0.1"
```

```
# Create a scatter plot with a trend line
library(ggplot2)
ggplot(df, aes(x = total_debt, y = credit_score)) +
  geom_point(alpha = 0.5, color = "steelblue") +
  geom_smooth(method = "lm", se = FALSE, color = "darkred") +
  labs(title = "Credit Score vs. Total Debt",
       x = "Total Debt",
       y = "Credit Score") +
  theme_minimal()
```

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


Credit Score vs. Total Debt
Credit Score
800
700
600
500
0e+00
1e+05
2e+05
3e+05
4e+05
5e+05
Total Debt