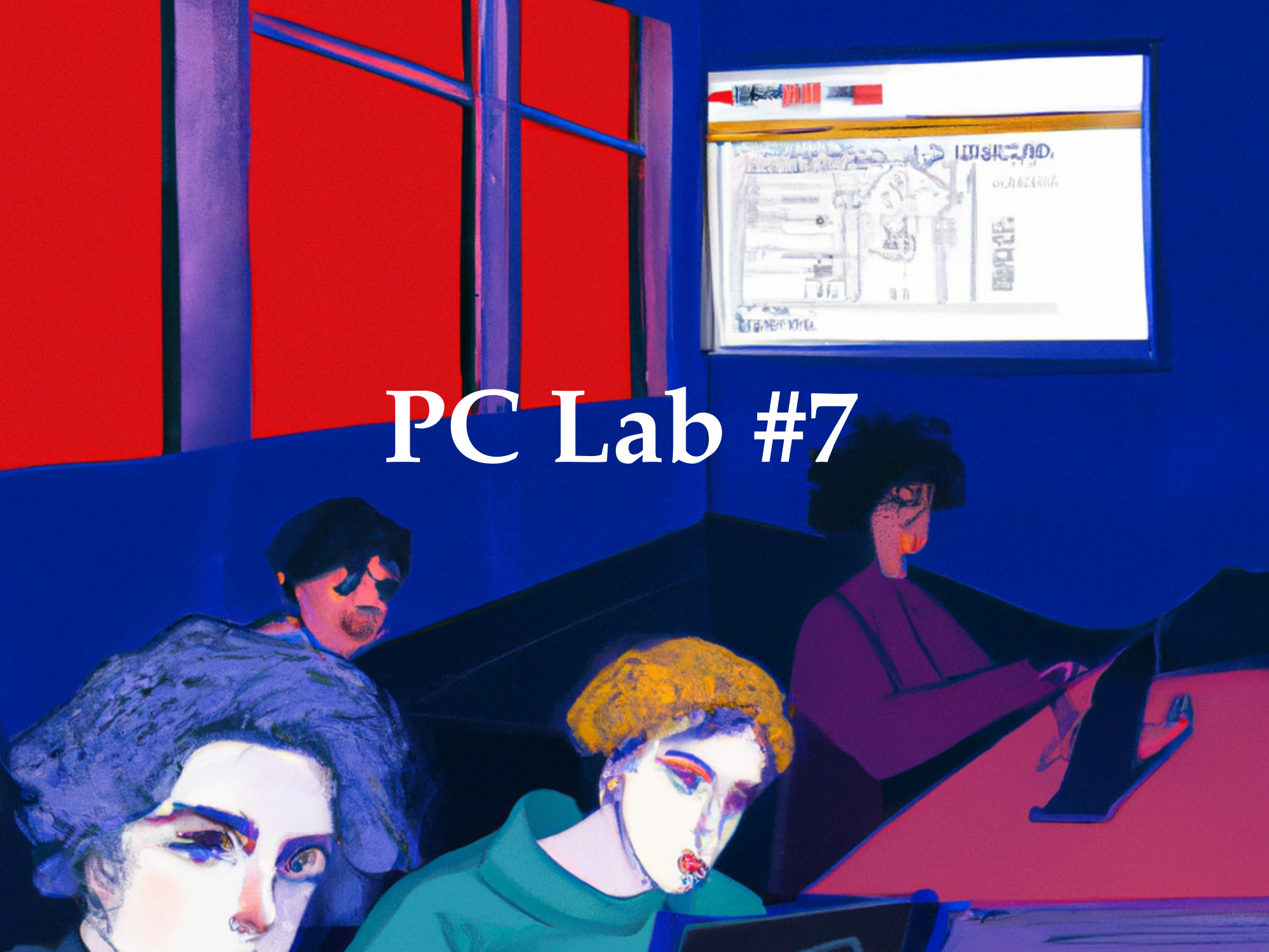
# 20598 – Finance with Big Data

## PC Lab #7: Bank Customers Segmentation (Week 8)

Clément Mazet-Sonilhac
clement.mazetsonilhac@unibocconi.it

Department of Finance, Bocconi University


PC Lab #7

# PC Labs Grading

- PC Labs solutions are submitted as Jupyter Notebooks, via email
  - Email title : PCLab#7 - Group X - Name1 Name2 Name3
  - Your Jupyter Notebook starts in the same way (same .ipynb name)
  - Tell me (again) how long did it take

**Reminder**: Andrea Andolfatto (andrea.andolfatto@phd.unibocconi.it) should be cc'ed to your email

# PC Labs Grading

- PC Labs grade will depend on :
  - Your ability to submit it before the deadline
  - The quality of your code (comments, readability, use of functions, etc.)
  - The structure of the Jupyter Notebook : well organized, explain what you are doing and why
  - Your ability to complete the tasks and innovate
  - You should maybe produce less, but more *useful* outputs

# Goals

- Manipulate and visualize bank customers data
- Perform segmentation : Find the optimal number of clusters
- Compare K-means, PCA and Autoencoder methods

# Big picture context

- You've been hired as a consultant for a bank that suffers from FinTech competition
- The bank has extensive data on its customers over the past 6 months
- Wants to launch a marketing campaign, need to divide customers in 3 groups (at least)

# Task #1 : Basic manipulation and descriptive statistics

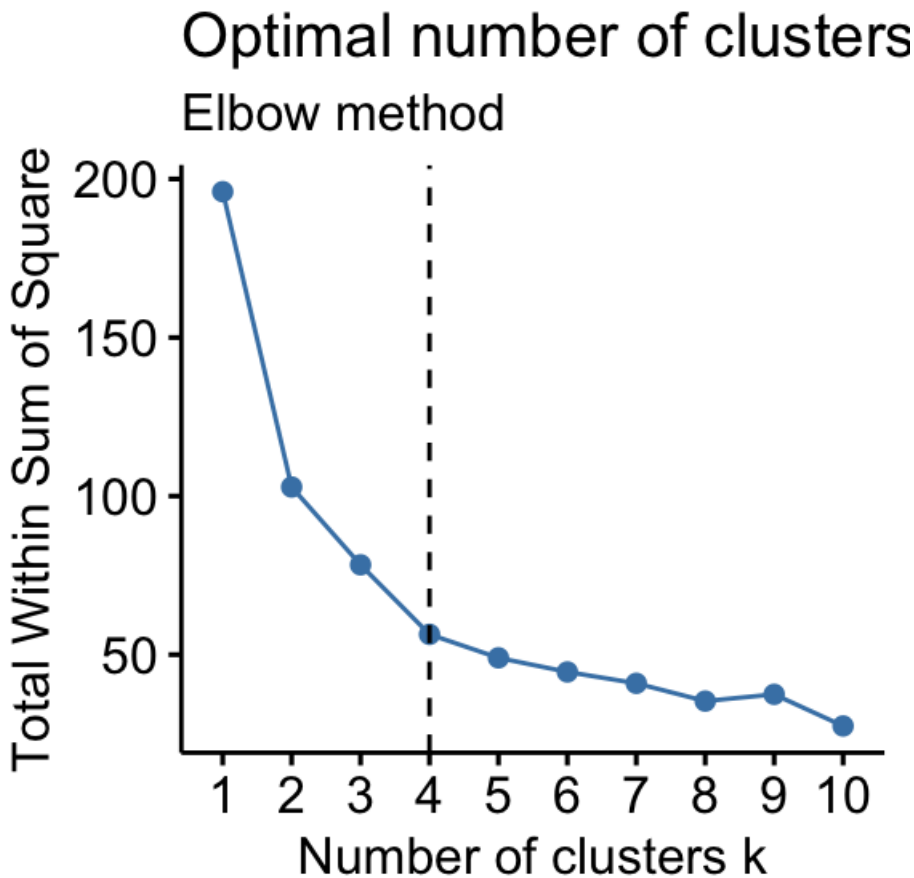
- Import the `BankCustomersData_PCLab7.csv` data
- Describe the sample :
  - Let's see if we have any missing data
  - Fill up the missing elements with mean of the `'MINIMUM_PAYMENT'`
  - Fill up the missing elements with mean of the `'CREDIT_LIMIT'`
  - Let's see if we have duplicated entries in the data
  - What is the average credit limit ?
- Usual correlation and distribution of key variables (show only if interesting)

# Task #1 : Glossary

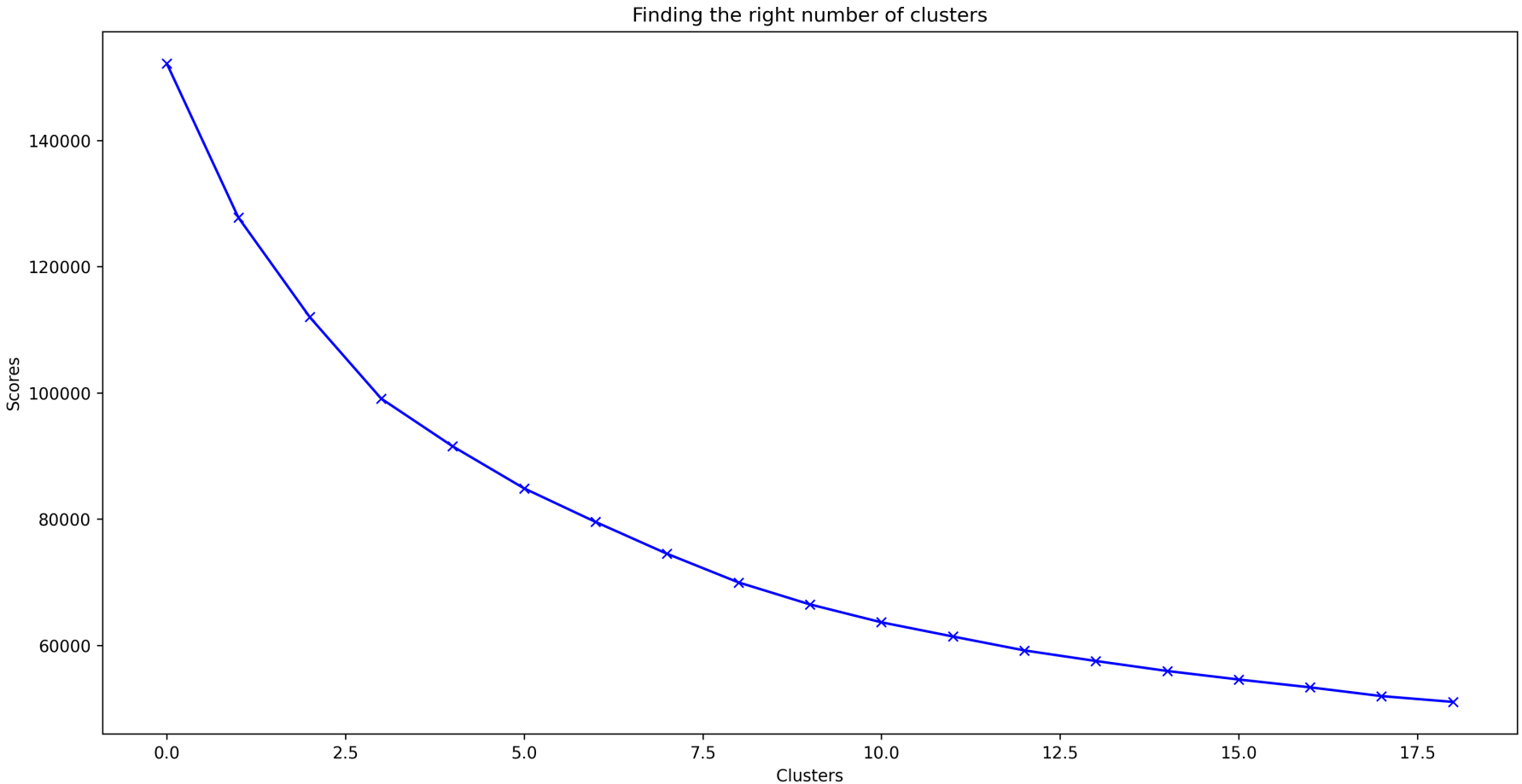
```
# CUSTID: Identification of Credit Card holder
# BALANCE: Balance amount left in customer's account to make purchases
# BALANCE_FREQUENCY: How frequently the Balance is updated, score between 0 and
1 (1 = frequently updated, 0 = not frequently updated)
# PURCHASES: Amount of purchases made from account
# ONEOFFPURCHASES: Maximum purchase amount done in one-go
# INSTALLMENTS_PURCHASES: Amount of purchase done in installment
# CASH_ADVANCE: Cash in advance given by the user
# PURCHASES_FREQUENCY: How frequently the Purchases are being made, score
between 0 and 1 (1 = frequently purchased, 0 = not frequently purchased)
# ONEOFF_PURCHASES_FREQUENCY: How frequently Purchases are happening in one-go
(1 = frequently purchased, 0 = not frequently purchased)
# PURCHASES_INSTALLMENTS_FREQUENCY: How frequently purchases in installments
are being done (1 = frequently done, 0 = not frequently done)
# CASH_ADVANCE_FREQUENCY: How frequently the cash in advance being paid
# CASH_ADVANCE_TRX: Number of Transactions made with "Cash in Advance"
# PURCHASES_TRX: Number of purchase transactions made
# CREDIT_LIMIT: Limit of Credit Card for user
# PAYMENTS: Amount of Payment done by user
# MINIMUM_PAYMENTS: Minimum amount of payments made by user
# PRC_FULL_PAYMENT: Percent of full payment paid by user
# TENURE: Tenure of credit card service for user
```

# Task #2 : Optimal nb. of clusters using the Elbow method

- Let's scale the data first (`scaler = StandardScaler(); data_df_scaled = scaler.fit_transform(data_df)`)
- Draw the *Elbow* curve to find the optimal number of clusters



# Task #2 : Optimal number of clusters using the Elbow method



# Task #2 : Apply K-means method

- Apply the K-means method (w. optimal K found above) and describe your clusters
- Does it make sense ? How can you interpret those customers groups ? If the bank wants to specialize on high-spending individuals (who use credit card as a loan), on what cluster should we concentrate the marketing efforts ?
- Find a smart way to summarize and plot your results

# Optional Task #3 : Perform dimensionality reduction

- Use PCA (using nb. of comp. = 2) or Auto-encoders (ANN) to perform dimensionality reduction
- Does it help you to think about the right number of clusters you should use ? If yes, how are you able to improve the segmentation ?