

R-Kielen Projekti

ETL



Janne Bragge

Sisällysluettelo

[Datan luominen 3](#_Toc184394682)

[Tiedoston käyttö 3](#_Toc184394683)

[Vaihe 1: Extract - Datan lataus 4](#_Toc184394684)

[Koodi 4](#_Toc184394685)

[Selitys 4](#_Toc184394686)

[Odotettu tulos 5](#_Toc184394687)

[Vaihe 2: Transform - Datan puhdistus ja rikastus 5](#_Toc184394688)

[Koodi 5](#_Toc184394689)

[Selitys 6](#_Toc184394690)

[Odotettu tulos 6](#_Toc184394691)

[Vaihe 3: Yhteenvetojen luominen 8](#_Toc184394692)

[Koodi 8](#_Toc184394693)

[Selitys 9](#_Toc184394694)

[Odotettu tulos 10](#_Toc184394695)

[Vaihe 4: Load - Datan tallennus 11](#_Toc184394696)

[Koodi 11](#_Toc184394697)

[Selitys 11](#_Toc184394698)

[Odotetut tiedostot 11](#_Toc184394699)

[Vaihe 5: Datan tallentaminen tietokantaan 12](#_Toc184394700)

[Askel 1: Asenna ja lataa tarvittavat paketit 12](#_Toc184394701)

[Askel 2: Luo SQLite-tietokanta ja yhdistä 12](#_Toc184394702)

[Askel 3: Tallenna data tietokantaan 12](#_Toc184394703)

[Askel 4: Tarkista tietokannan sisältö 12](#_Toc184394704)

[Askel 5: Sulje yhteys 13](#_Toc184394705)

[Vaihe 6. Datavisualisointi: Myyntidatan analyysi 15](#_Toc184394706)

[Lataa ggplot2-paketti 15](#_Toc184394707)

[Visualisointien luominen 15](#_Toc184394708)

[1. Kokonaismyynnit kategorioittain 15](#_Toc184394709)

[2. Tuloluokkien jakautuminen kategorioittain 16](#_Toc184394710)

[3. Myynnin kehitys ajan kuluessa 16](#_Toc184394711)

[4. Hajontakaavio tuotteiden määrän ja hinnan suhteesta 17](#_Toc184394712)

[Lisävisualisoinnit 18](#_Toc184394713)

[Historiallinen jakauma kokonaishinnan mukaan 18](#_Toc184394714)

[Termien selitys 18](#_Toc184394715)

[Odotetut tulokset 19](#_Toc184394716)

# Datan luominen

Tässä luomme esimerkkimyyntejä sisältävän datan.

# Luo myyntidata

set.seed(123) # Jotta tulokset ovat toistettavia

sales\_data <- data.frame(

Date = sample(seq(as.Date("2023-01-01"), as.Date("2023-12-31"), by = "day"), 100, replace = TRUE),

Product = sample(c("Product A", "Product B", "Product C"), 100, replace = TRUE),

Category = sample(c("Electronics", "Groceries", "Clothing"), 100, replace = TRUE),

Quantity = sample(1:20, 100, replace = TRUE),

Price = round(runif(100, 5, 100), 2)

)

# Tarkista data

head(sales\_data)

# Tallenna data CSV-tiedostoon, jotta voimme käyttää sitä myöhemmin

write.csv(sales\_data, "sales\_data.csv", row.names = FALSE)

## Tiedoston käyttö

Kun suoritat tämän koodin, se luo sales\_data.csv-tiedoston, jonka voimme ladata ja käyttää seuraavissa vaiheissa.

**Nyt kun tiedosto on luotu:**

1. Varmista, että se on tallennettu samaan kansioon, josta suoritat R-koodia.
2. Voimme jatkaa **Extract**-vaiheeseen ja ladata tämän tiedoston käsittelyyn.

# Vaihe 1: Extract - Datan lataus

Nyt ladataan juuri luotu CSV-tiedosto sales\_data.csv R:ään ja tarkistetaan sen sisältö.

## Koodi

# Lataa data CSV-tiedostosta

sales\_data <- read.csv("sales\_data.csv", stringsAsFactors = FALSE)

# Tarkista datan ensimmäiset rivit

head(sales\_data)

A black text on a white background

Description automatically generated

# Tarkista datan rakenne

str(sales\_data)

A black text on a white background

Description automatically generated

# Tarkista, onko tiedostossa puuttuvia arvoja

sum(is.na(sales\_data))

## Selitys

1. **read.csv()**: Lataa CSV-tiedoston dataan. Käytämme stringsAsFactors = FALSE, jotta tekstimuotoiset sarakkeet pysyvät merkkijonoina, eivätkä muutu automaattisesti faktoreiksi.
2. **head()**: Näyttää datan ensimmäiset rivit, jotta voimme tarkistaa sen rakenteen.
3. **str()**: Näyttää datan rakenteen, kuten sarakkeiden nimet, tietotyypit ja muut tiedot.
4. **sum(is.na())**: Tarkistaa, onko datassa puuttuvia arvoja.

## Odotettu tulos

* Näet myyntidatan, jossa on seuraavat sarakkeet:
  + **Date**: Myyntipäivämäärä
  + **Product**: Tuotteen nimi
  + **Category**: Tuotteen kategoria
  + **Quantity**: Myyty määrä
  + **Price**: Tuotteen yksikköhinta
* Varmistamme, että datassa ei ole puuttuvia arvoja.

# Vaihe 2: Transform - Datan puhdistus ja rikastus

Nyt puhdistamme ja rikastamme dataa seuraavilla tavoilla:

1. Poistetaan mahdolliset puuttuvat arvot.
2. Muutetaan päivämäärät oikeaan muotoon.
3. Lasketaan kokonaishinta (määrä \* yksikköhinta).
4. Luodaan uusi sarake, joka luokittelee tuotteet tuloluokkiin.

## Koodi

# Lataa tarvittavat paketit

library(dplyr)

library(lubridate)

# 1. Poistetaan puuttuvat arvot

cleaned\_data <- sales\_data %>%

drop\_na()

# 2. Muutetaan päivämäärät oikeaan muotoon

cleaned\_data <- cleaned\_data %>%

mutate(Date = as.Date(Date, format = "%Y-%m-%d"))

# 3. Lasketaan kokonaishinta

cleaned\_data <- cleaned\_data %>%

mutate(TotalPrice = Quantity \* Price)

# 4. Luodaan tuloluokkien sarake

cleaned\_data <- cleaned\_data %>%

mutate(IncomeCategory = case\_when(

TotalPrice < 100 ~ "Low",

TotalPrice >= 100 & TotalPrice < 500 ~ "Medium",

TRUE ~ "High"

))

# Tarkista puhdistettu ja rikastettu data

head(cleaned\_data)

A table with numbers and letters

Description automatically generated

## Selitys

1. **drop\_na()**: Poistaa rivit, joissa on puuttuvia arvoja.
2. **mutate()**:
   * Muuttaa Date-sarakkeen päivämäärämuotoon.
   * Laskee uuden sarakkeen TotalPrice, joka on myyntimäärä kerrottuna hinnalla.
   * Luo IncomeCategory-sarake, joka luokittelee tuotteet myynnin mukaan kolmeen tuloluokkaan:
     + **Low**: Kokonaishinta alle 100.
     + **Medium**: Kokonaishinta 100–500.
     + **High**: Kokonaishinta yli 500.

## Odotettu tulos

* Data on nyt puhdistettu ja rikastettu seuraavilla sarakkeilla:
  + **TotalPrice**: Lasketaan myynnin kokonaisarvo (määrä \* hinta).
  + **IncomeCategory**: Tuotteet jaoteltuna Low, Medium ja High -tuloluokkiin.

# Vaihe 3: Yhteenvetojen luominen

Nyt luomme yhteenvetoja puhdistetusta ja rikastetusta datasta. Tavoitteena on:

1. Laskea kategorioiden kokonaissummat, keskihinnat ja kokonaismyyntimäärät.
2. Tarkastella, kuinka eri tuloluokat (Low, Medium, High) jakautuvat kategorioittain.

## Koodi

# Lataa dplyr-paketti (jos ei jo ladattu)

library(dplyr)

# 1. Yhteenveto kategorioittain

category\_summary <- cleaned\_data %>%

group\_by(Category) %>%

summarise(

TotalSales = sum(TotalPrice),

AveragePrice = mean(Price),

TotalQuantity = sum(Quantity)

)

# Näytä kategorioiden yhteenveto

print(category\_summary)

A screenshot of a computer code

Description automatically generated

# 2. Tuloluokkien jakautuminen kategorioittain

income\_summary <- cleaned\_data %>%

group\_by(Category, IncomeCategory) %>%

summarise(

Count = n(),

TotalSales = sum(TotalPrice)

)

# Näytä tuloluokkien jakautuminen

print(income\_summary)

A screenshot of a computer program

Description automatically generated

## Selitys

1. **group\_by(Category)**: Ryhmittelee datan kategorian mukaan.
2. **summarise()**: Laskee yhteenvedot:
   * **TotalSales**: Kokonaissumma kategorian myynneistä.
   * **AveragePrice**: Keskihinta kategorian tuotteille.
   * **TotalQuantity**: Myytyjen tuotteiden kokonaismäärä.
3. **Tuloluokkien jakautuminen**:
   * **group\_by(Category, IncomeCategory)**: Ryhmittelee datan sekä kategorian että tuloluokan mukaan.
   * **Count**: Laskee, kuinka monta myyntiä kuuluu kuhunkin ryhmään.
   * **TotalSales**: Laskee kokonaissumman kullekin tuloluokalle kategorioittain.

## Odotettu tulos

**Kategorioiden yhteenveto** näyttää esimerkiksi:

**A screenshot of a computer

Description automatically generated**

**Tuloluokkien jakautuminen kategorioittain** näyttää esimerkiksi:

A screenshot of a computer

Description automatically generated

# Vaihe 4: Load - Datan tallennus

Tässä vaiheessa tallennamme:

1. **Puhdistetun ja rikastetun datan** uuteen CSV-tiedostoon.
2. **Kategorioiden yhteenvedon** ja **tuloluokkien jakautumisen** erillisiin tiedostoihin.

## Koodi

# 1. Tallennetaan puhdistettu data

write.csv(cleaned\_data, "cleaned\_sales\_data.csv", row.names = FALSE)

# 2. Tallennetaan kategorioiden yhteenveto

write.csv(category\_summary, "category\_summary.csv", row.names = FALSE)

# 3. Tallennetaan tuloluokkien jakautuminen kategorioittain

write.csv(income\_summary, "income\_summary.csv", row.names = FALSE)

# Vahvistusviesti

cat("Data on tallennettu seuraaviin tiedostoihin:\n",

"- cleaned\_sales\_data.csv\n",

"- category\_summary.csv\n",

"- income\_summary.csv\n")

## Selitys

1. **write.csv()**:
   * Tallentaa datan CSV-tiedostoon.
   * **row.names = FALSE** estää rivinumeroiden lisäämisen tiedostoon.
2. **cat()**: Tulostaa viestin, joka vahvistaa tallennuksen onnistuneen.

## Odotetut tiedostot

1. **cleaned\_sales\_data.csv**: Puhdistettu ja rikastettu raakadata.
2. **category\_summary.csv**: Kategorioiden kokonaissummat, keskihinnat ja myyntimäärät.
3. **income\_summary.csv**: Tuloluokkien jakautuminen kategorioittain.

# Vaihe 5: Datan tallentaminen tietokantaan

Voimme tallentaa puhdistetun ja käsitellyn datan SQLite-tietokantaan käyttäen R:n **DBI**- ja **RSQLite**-paketteja. Tämä prosessi luo tietokannan ja taulut, joissa data säilytetään.

## Askel 1: Asenna ja lataa tarvittavat paketit

Jos paketteja ei ole vielä asennettu, asenna ne seuraavasti:

install.packages("DBI")

install.packages("RSQLite")

Lataa paketit:

library(DBI)

library(RSQLite)

## Askel 2: Luo SQLite-tietokanta ja yhdistä

# Luo tietokanta ja muodosta yhteys

con <- dbConnect(RSQLite::SQLite(), "sales\_data.db")

## Askel 3: Tallenna data tietokantaan

**Tallennetaan puhdistettu data:**

# Tallenna cleaned\_data-taulukko tietokantaan

dbWriteTable(con, "cleaned\_sales\_data", cleaned\_data, overwrite = TRUE)

**Tallennetaan yhteenvetotaulukot:**

# Tallenna category\_summary-taulukko

dbWriteTable(con, "category\_summary", category\_summary, overwrite = TRUE)

# Tallenna income\_summary-taulukko

dbWriteTable(con, "income\_summary", income\_summary, overwrite = TRUE)

## Askel 4: Tarkista tietokannan sisältö

# Listaa kaikki taulut tietokannassa

dbListTables(con)

# Esikatsele taulua

dbReadTable(con, "cleaned\_sales\_data") %>% head()

A number of numbers on a white background

Description automatically generated

## Askel 5: Sulje yhteys

Kun kaikki on tehty, muista sulkea yhteys tietokantaan:

dbDisconnect(con)

**Tietokantatiedoston sijainti**

Tämä luo **sales\_data.db**-tiedoston samaan hakemistoon, jossa suoritat R-skriptisi. SQLite-tietokanta sisältää kolme taulua:

1. **cleaned\_sales\_data**
2. **category\_summary**
3. **income\_summary**

**Jatko-optiot**

Voit käyttää tätä tietokantaa myöhemmin:

* **Datan noutamiseen ja analyysiin R:llä**:
* con <- dbConnect(RSQLite::SQLite(), "sales\_data.db")
* data <- dbReadTable(con, "cleaned\_sales\_data")
* dbDisconnect(con)
* **Tietokannan käyttö SQL-komentojen avulla**:
* result <- dbGetQuery(con, "SELECT \* FROM category\_summary WHERE TotalSales > 5000")
* print(result)

# Vaihe 6. Datavisualisointi: Myyntidatan analyysi

Tässä vaiheessa visualisoimme tietoja, jotka ovat syntyneet ETL-prosessin aikana. Käytämme **ggplot2**-kirjastoa, joka on tehokas ja monipuolinen työkalu datan visualisointiin.

## Lataa ggplot2-paketti

Asenna ja lataa ggplot2, jos se ei ole jo käytössä:

install.packages("ggplot2")

library(ggplot2)

## Visualisointien luominen

### 1. Kokonaismyynnit kategorioittain

ggplot(category\_summary, aes(x = Category, y = TotalSales, fill = Category)) +

geom\_bar(stat = "identity") +

labs(title = "Kategorioiden kokonaismyynnit", x = "Kategoria", y = "Kokonaismyynti") +

theme\_minimal()

A screenshot of a graph

Description automatically generated

### 2. Tuloluokkien jakautuminen kategorioittain

ggplot(income\_summary, aes(x = Category, y = TotalSales, fill = IncomeCategory)) +

geom\_bar(stat = "identity", position = "dodge") +

labs(title = "Tuloluokkien jakautuminen kategorioittain",

x = "Kategoria", y = "Kokonaismyynti") +

theme\_minimal()

A screenshot of a graph

Description automatically generated

### 3. Myynnin kehitys ajan kuluessa

Jos haluat tarkastella myyntien kehitystä ajan kuluessa, voimme käyttää **puhdistettua dataa**:

# Päivämääräkohtainen kokonaismyynti

daily\_sales <- cleaned\_data %>%

group\_by(Date) %>%

summarise(DailyTotal = sum(TotalPrice))

# Viivakaavio myynnin kehityksestä

ggplot(daily\_sales, aes(x = Date, y = DailyTotal)) +

geom\_line(color = "blue") +

labs(title = "Myynnin kehitys ajan kuluessa", x = "Päivämäärä", y = "Kokonaismyynti") +

theme\_minimal()

A screen shot of a graph

Description automatically generated

### 4. Hajontakaavio tuotteiden määrän ja hinnan suhteesta

ggplot(cleaned\_data, aes(x = Quantity, y = Price, color = Category)) +

geom\_point(size = 2, alpha = 0.7) +

labs(title = "Tuotteiden määrä vs. hinta", x = "Määrä", y = "Hinta") +

theme\_minimal()

A screen shot of a graph

Description automatically generated

## Lisävisualisoinnit

### Historiallinen jakauma kokonaishinnan mukaan

ggplot(cleaned\_data, aes(x = TotalPrice, fill = IncomeCategory)) +

geom\_histogram(bins = 20, alpha = 0.7) +

labs(title = "Kokonaishintojen jakauma", x = "Kokonaishinta", y = "Frekvenssi") +

theme\_minimal()

A graph with red and blue bars

Description automatically generated

### Termien selitys

* **geom\_bar(stat = "identity")**: Käytetään pylväsdiagrammien piirtämiseen, kun arvot ovat jo laskettuja.
* **geom\_line()**: Piirtää viivakaavion.
* **geom\_point()**: Luo hajontakaavion.
* **geom\_histogram()**: Näyttää arvon jakauman histogrammina.
* **labs()**: Lisää otsikoita ja akseleiden selitteitä.
* **theme\_minimal()**: Käyttää minimalistista teemaa, joka tekee kaaviosta modernin ja selkeän.

### Odotetut tulokset

* Selkeitä ja visuaalisesti miellyttäviä kaavioita, jotka havainnollistavat datan piirteitä ja trendejä:
  + Kokonaismyyntien vertailu kategorioittain.
  + Tuloluokkien jakautuminen.
  + Myynnin kehitys ajan kuluessa.
  + Tuotteiden hinnan ja määrän välinen suhde.