Relationships Between Continuous Variables

Luis Francisco Gomez Lopez

FAEDIS

2023-06-23

Contents

Please Read Me

- This presentation is based on (Chapman and Feit 2019, chap. 4)

Purpose

- Understand the relationships between pairs of variables in multivariate data and examine how to visualize the relationships and compute statistics that describe their associations

CRM system data

- **cust.id**: customer identifier
- **age**: decimal age in years
- **credit.score**: 3-digit number in [300, 900], representing the credit risk
- **email**: whether or not there is information about the customer email
- **distance.to.store**: distance in kilometers to the nearest physical store
- **online.visits**: yearly visits to the online store
- **online.trans**: yearly online orders
- **online.spend**: yearly spending in those online orders
- **store.trans**: yearly orders in physical stores
- **store.spend**: yearly spending in those physical store orders

CRM system data

- **sat.service**: satisfaction with service using an ordinal 5 point scale and collected using a survey
- **sat.selection**: satisfaction with product selection using an ordinal 5 point scale and collected using a survey
  - **Ordinal 5 point scale used and possible values in the survey**:
    - Extremely satisfied: 5
    - Very satisfied: 4
    - Moderately satisfied: 3
    - Very satisfied: 2
    - Extremely satisfied: 1
    - NA: customer did not response the survey

CRM system data

- **Import data**

```
customer <- read_csv(file = "http://goo.gl/PmPkaG")
customer |> head(n=5)
```

```
# A tibble: 5 x 12
  cust.id   age credit.score email distance.to.store online.visits online.trans
    <dbl> <dbl>        <dbl> <chr>             <dbl>         <dbl>        <dbl>
1       1  22.9         631. yes                2.58            20            3
2       2  28.0         749. yes               48.2            121           39
3       3  35.9         733. yes                1.29            39           14
4       4  30.5         830. yes                5.25             1            0
5       5  38.7         734. no                25.0             35           11
# i 5 more variables: online.spend <dbl>, store.trans <dbl>, store.spend <dbl>,
#   sat.service <dbl>, sat.selection <dbl>
```

CRM system data

- **Transform data**

```
customer <- customer |>
  mutate(cust.id = factor(x = cust.id, ordered = FALSE),
         email = factor(x = email, ordered = FALSE),
         online.visits = as.integer(x = online.visits),
         online.trans = as.integer(x = online.trans),
         store.trans = as.integer(x = store.trans),
         sat.service = factor(x = sat.service, ordered = TRUE),
         sat.selection = factor(x = sat.selection, ordered = TRUE))
customer |> head(n=5)
```

```
# A tibble: 5 x 12
  cust.id   age credit.score email distance.to.store online.visits online.trans
  <fct>   <dbl>        <dbl> <fct>             <dbl>         <int>        <int>
1 1        22.9         631. yes                2.58            20            3
2 2        28.0         749. yes               48.2            121           39
3 3        35.9         733. yes                1.29            39           14
4 4        30.5         830. yes                5.25             1            0
5 5        38.7         734. no                25.0             35           11
# i 5 more variables: online.spend <dbl>, store.trans <int>, store.spend <dbl>,
#   sat.service <ord>, sat.selection <ord>
```

CRM system data

- **Inspect data**

```
customer |> glimpse()
```

```
Rows: 1,000
Columns: 12
$ cust.id           <fct> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1~
$ age               <dbl> 22.89437, 28.04994, 35.87942, 30.52740, 38.73575, 42~
$ credit.score      <dbl> 630.6089, 748.5746, 732.5459, 829.5889, 733.7968, 68~
$ email             <fct> yes, yes, yes, yes, no, yes, yes, yes, no, no, no, y~
$ distance.to.store <dbl> 2.582494, 48.175989, 1.285712, 5.253992, 25.044693, ~
$ online.visits     <int> 20, 121, 39, 1, 35, 1, 1, 48, 0, 14, 2, 0, 0, 108, 0~
$ online.trans      <int> 3, 39, 14, 0, 11, 1, 1, 13, 0, 6, 1, 0, 0, 26, 0, 0,~
$ online.spend      <dbl> 58.42999, 756.88008, 250.32801, 0.00000, 204.69331, ~
$ store.trans       <int> 4, 0, 0, 2, 0, 0, 2, 4, 0, 3, 0, 9, 0, 3, 0, 2, 0, 2~
$ store.spend       <dbl> 140.32321, 0.00000, 0.00000, 95.91194, 0.00000, 0.00~
$ sat.service       <ord> 3, 3, NA, 4, 1, NA, 3, 2, 4, 3, 3, NA, NA, 1, NA, 3,~
$ sat.selection     <ord> 3, 3, NA, 2, 1, NA, 3, 3, 2, 2, 2, NA, NA, 2, NA, 3,~
```

CRM system data

- **Summarize data**
  - Ups the table is really big!!! Try it in your console to see the complete table

```
customer |> skim()
```

Table 1: Data summary

Name	customer
Number of rows	1000
Number of columns	12
_______________________	
Column type frequency:	
factor	4
numeric	8
________________________	
Group variables	None

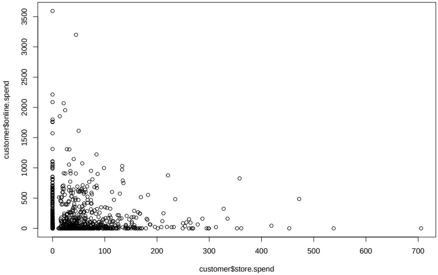
Variable type: factor

skim_variable	n_missing	complete_rate	ordered	n_unique	top_counts
cust.id	0	1.00	FALSE	1000	1: 1, 2: 1, 3: 1, 4: 1
email	0	1.00	FALSE	2	yes: 814, no: 186
sat.service	341	0.66	TRUE	5	3: 309, 4: 167, 2: 133, 5: 28

CRM system data

- **Scatterplots: the base R way**

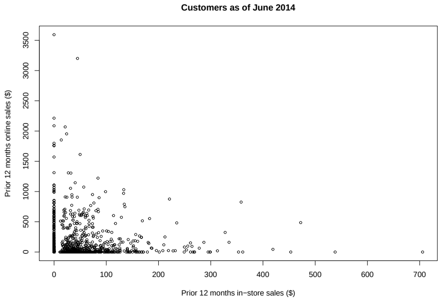
```
plot(x = customer$store.spend, y = customer$online.spend)
```



CRM system data

- **Scatterplots: the base R way**

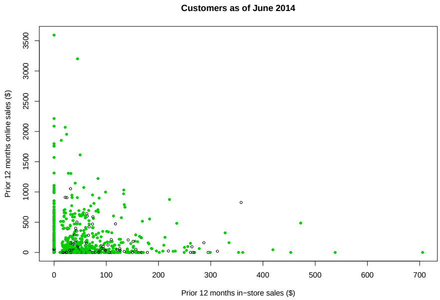
```r
plot(x = customer$store.spend, y = customer$online.spend,
     cex=0.7,
     main="Customers as of June 2014",
     xlab="Prior 12 months in-store sales ($)",
     ylab="Prior 12 months online sales ($)")
```



CRM system data

- **Scatterplots: the base R way**

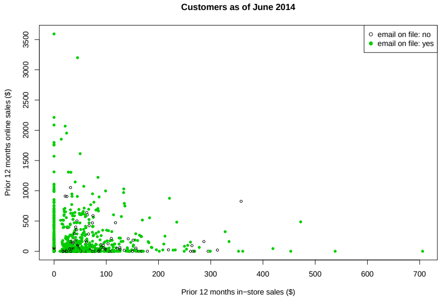
```
my.col <- c("black", "green3")
my.pch <- c(1, 19)
plot(x = customer$store.spend, y = customer$online.spend,
     cex=0.7, col=my.col[customer$email], pch=my.pch[customer$email],
     main="Customers as of June 2014",
     xlab="Prior 12 months in-store sales ($)",
     ylab="Prior 12 months online sales ($)")
```



CRM system data

- **Scatterplots: the base R way**

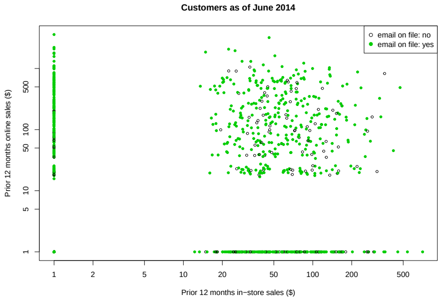
```
my.col <- c("black", "green3")
my.pch <- c(1, 19)
plot(x = customer$store.spend, y = customer$online.spend,
     cex=0.7, col=my.col[customer$email], pch=my.pch[customer$email],
     main="Customers as of June 2014",
     xlab="Prior 12 months in-store sales ($)",
     ylab="Prior 12 months online sales ($)")
legend(x="topright", legend=paste("email on file:", levels(customer$email)), col=my.col, pch=my.pch)
```



CRM system data

- **Scatterplots: the base R way**

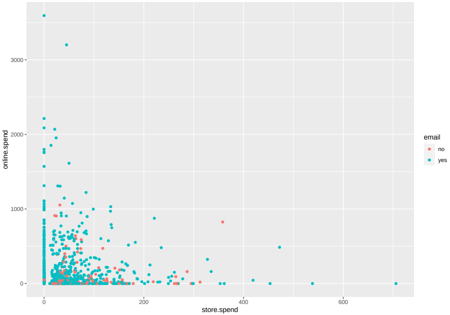
```
my.col <- c("black", "green3")
my.pch <- c(1, 19)
plot(x = customer$store.spend + 1, y = customer$online.spend + 1,
     cex=0.7, col=my.col[customer$email], pch=my.pch[customer$email],
     log ="xy",
     main="Customers as of June 2014",
     xlab="Prior 12 months in-store sales ($)",
     ylab="Prior 12 months online sales ($)")
legend(x="topright", legend=paste("email on file:", levels(customer$email)), col=my.col, pch=my.pch)
```



CRM system data

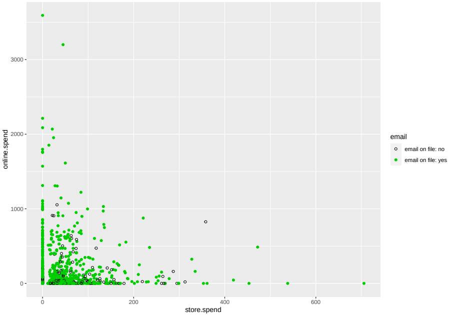
- **Scatterplots: the tidyverse way**

```
customer |> ggplot() +
  geom_point(aes(x = store.spend, y = online.spend, color = email))
```



- **Scatterplots: the tidyverse way**

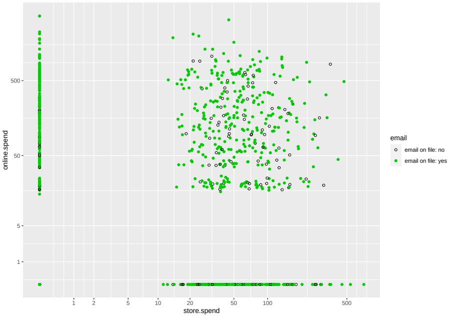
```
customer |> ggplot() +
  geom_point(aes(x = store.spend, y = online.spend, color = email, shape = email)) +
  scale_color_manual(values = c("black", "green3"), labels = c("email on file: no", "email on file: yes")) +
  scale_shape_manual(values = c(1, 19), labels = c("email on file: no", "email on file: yes"))
```



CRM system data

- **Scatterplots: the tidyverse way**

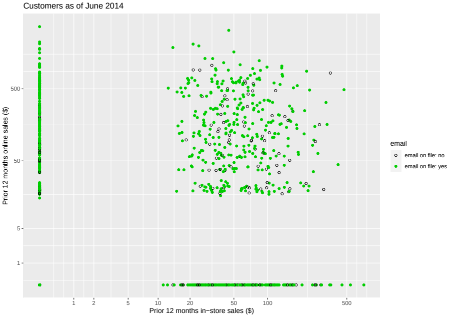
```
customer |> ggplot() +
  geom_point(aes(x = store.spend, y = online.spend, color = email, shape = email)) +
  scale_color_manual(values = c("black", "green3"), labels = c("email on file: no", "email on file: yes")) +
  scale_shape_manual(values = c(1, 19), labels = c("email on file: no", "email on file: yes")) +
  scale_x_continuous(trans = "log1p", breaks = c(1, 2, 5, 10, 20, 50, 100, 500)) +
  scale_y_continuous(trans = "log1p", breaks = c(1, 5, 50, 500))
```



CRM system data

- **Scatterplots: the tidyverse way**

```
customer |> ggplot() +
  geom_point(aes(x = store.spend, y = online.spend, color = email, shape = email)) +
  scale_color_manual(values = c("black", "green3"), labels = c("email on file: no", "email on file: yes")) +
  scale_shape_manual(values = c(1, 19), labels = c("email on file: no", "email on file: yes")) +
  scale_x_continuous(trans = "log1p", breaks = c(1, 2, 5, 10, 20, 50, 100, 500)) +
  scale_y_continuous(trans = "log1p", breaks = c(1, 5, 50, 500)) +
  labs(x = "Prior 12 months in-store sales ($)", y = "Prior 12 months online sales ($)",
       title = "Customers as of June 2014")
```



CRM system data

- **Correlation Coefficients**
  - Pearson correlation coefficient for a sample

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Where n is the sample size, we must have paired numeric data $\{(x_1, y_1), ..., (x_n, y_n)\}$, $\bar{x} = \sum_{i=1}^{n} x_i$ and $\bar{y} = \sum_{i=1}^{n} y_i$

- This is a "nasty" formula but we can brake it down in smaller chunks

CRM system data

- **Correlation Coefficients**
  - Pearson correlation coefficient for a sample

```
age_mean <- mean(customer$age)
age_credit.score <- mean(customer$credit.score)
numerator <- sum((customer$age - age_mean) * (customer$credit.score - age_credit.score))
denominator <- sqrt(sum((customer$age - age_mean)^2)) * sqrt(sum(((customer$credit.score - age_credit.score)^2)
pearson_corr <- numerator / denominator
pearson_corr
```

```
[1] 0.2545045
```

- **But don't worry be happy!!!: Use `cor`**

```
cor(customer$age, customer$credit.score, method = 'pearson')
```

```
[1] 0.2545045
```

CRM system data

- **Correlation matrices**
  - Pearson correlation coefficients for samples in a tibble

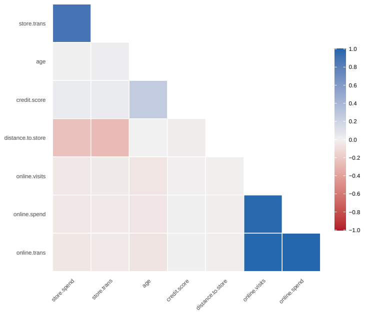
```
library(corrr) # Remember to install the package if it is not installed
correlation_matrix <- customer |>
  select(where(is.numeric)) |>
  correlate(use = "pairwise.complete.obs", # There are NA values
            method = "pearson",
            diagonal = NA)
correlation_matrix # Ups!!! The tibble is wide. Check out the tibble in your console
```

```
# A tibble: 8 x 9
  term             age credit.score distance.to.store online.visits online.trans
  <chr>          <dbl>        <dbl>             <dbl>         <dbl>        <dbl>
1 age         NA            0.255             0.00199       -0.0614     -0.0630
2 credit.sco~  0.255       NA                -0.0233        -0.0108     -0.00502
3 distance.t~  0.00199     -0.0233           NA             -0.0146     -0.0196
4 online.vis~ -0.0614      -0.0108           -0.0146        NA           0.987
5 online.tra~ -0.0630      -0.00502          -0.0196         0.987      NA
6 online.spe~ -0.0607      -0.00608          -0.0204         0.982       0.993
7 store.trans  0.0242       0.0404           -0.277         -0.0367     -0.0402
8 store.spend  0.00384      0.0423           -0.241         -0.0507     -0.0522
# i 3 more variables: online.spend <dbl>, store.trans <dbl>, store.spend <dbl>
```

CRM system data

- **Correlation matrices**
  - Pearson correlation coefficients for samples in a tibble

```
correlation_matrix |> autoplot(triangular = "lower")
```



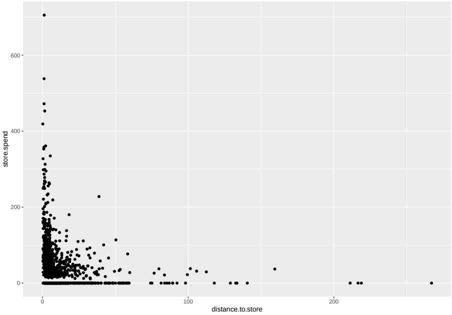
CRM system data

- **Transforming variables**

```
cor(customer$store.spend, customer$distance.to.store)
```

```
[1] -0.2414949
```

```
customer |> ggplot() +
  geom_point(aes(x = distance.to.store, y = store.spend))
```



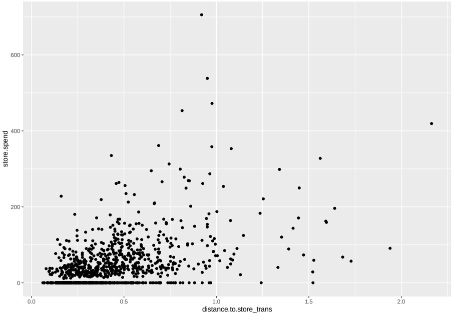
CRM system data

- **Transforming variables**

```
cor(customer$store.spend, 1 / sqrt(customer$distance.to.store))
```

```
[1] 0.4843334
```

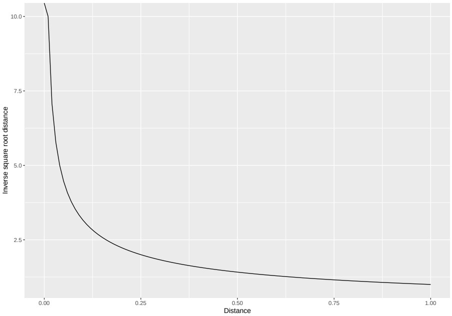
```
customer |>
  mutate(distance.to.store_trans = 1 / sqrt(distance.to.store)) |>
  ggplot() +
  geom_point(aes(x = distance.to.store_trans, y = store.spend))
```



CRM system data

- **Transforming variables**
  - Understanding the logic behind inverse square root distance

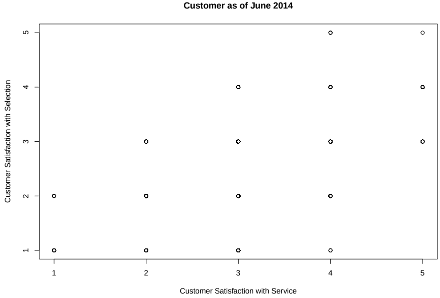
```
ggplot() +
  geom_function(fun = function(x) {1 / sqrt(x)}) +
  labs(x = "Distance",
       y = "Inverse square root distance")
```



CRM system data

- **Visualizing categorical variables**
  - **Scatterplots: the base R way**

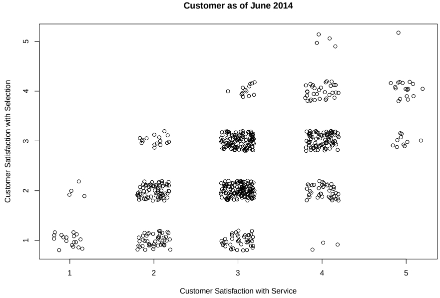
```r
plot(as.integer(customer$sat.service), as.integer(customer$sat.selection),
     xlab = "Customer Satisfaction with Service",
     ylab = "Customer Satisfaction with Selection",
     main = "Customer as of June 2014")
```



CRM system data

- **Visualizing categorical variables**
  - **Scatterplots: the base R way**

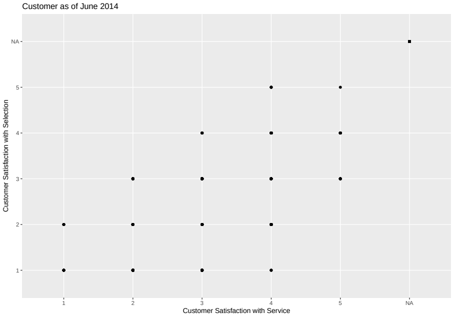
```
plot(jitter(as.integer(customer$sat.service)), jitter(as.integer(customer$sat.selection)),
     xlab = "Customer Satisfaction with Service",
     ylab = "Customer Satisfaction with Selection",
     main = "Customer as of June 2014")
```



CRM system data

- **Visualizing categorical variables**
  - **Scatterplots: the tidyverse way**

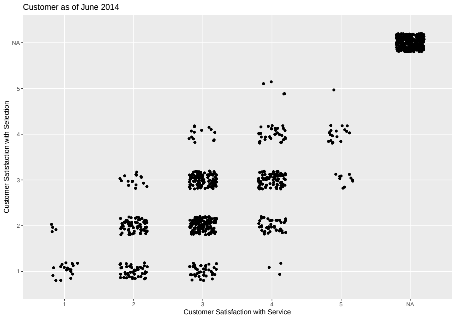
```
customer |>
  ggplot() +
  geom_point(aes(x = sat.service, y = sat.selection)) +
  labs(x = "Customer Satisfaction with Service",
       y = "Customer Satisfaction with Selection",
       title = "Customer as of June 2014")
```



CRM system data

- **Visualizing categorical variables**
  - **Scatterplots: the tidyverse way**

```
customer |>
  ggplot() +
  geom_point(aes(x = sat.service, y = sat.selection),
             position = position_jitter(width = 0.2, height = 0.2)) +
  labs(x = "Customer Satisfaction with Service",
       y = "Customer Satisfaction with Selection",
       title = "Customer as of June 2014")
```



Acknowledgments

- To my family that supports me
- To the taxpayers of Colombia and the **UMNG students** who pay my salary
- To the **Business Science** and **R4DS Online Learning** communities where I learn **R**
- To the **R Core Team**, the creators of **RStudio IDE**, **Quarto** and the authors and maintainers of the packages **tidyverse**, **skimr**, **corrr** and **tinytex** for allowing me to access these tools without paying for a license
- To the **Linux kernel community** for allowing me the possibility to use some **Linux distributions** as my main **OS** without paying for a license

References

Chapman, Chris, and Elea McDonnell Feit. 2019. *R For Marketing Research and Analytics*. 2nd ed. 2019. Use R! Cham: Springer International Publishing : Imprint: Springer. https://doi.org/10.1007/978-3-030-14316-9.