TREBALL DE FI DE MÀSTER

Màster Interuniversitari en Estadística i Investigació Operativa UPC-UB

Títol: Clustering d'agències asseguradores: descobrint sinèrgies i dissimilaritats entre elles

Autor: Pere Barber Lloréns

Director: Dr. Daniel Fernández Martínez

Departament: Estadística i Investigació Operativa

Universitat: Universitat Politècnica de Catalunya

Convocatòria: Setembre 2023

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

UNIVERSITAT DE BARCELONA

Clustering d'agències asseguradores:

descobrint sinèrgies i dissimilaritats entre elles

Pere Barber Lloréns

Treball de fi de màster
Màster en Estadística i Investigació Operativa

Tutor: Dr. Daniel Fernández Martínez

Setembre 2023

Resum

El sector assegurador té la funció principal de protegir els clients dels diferents riscos a què s'exposen diàriament. En particular, són les agències asseguradores les encarregades de fer les diferents gestions sobre els seus clients. En aquest treball es porta a terme una segmentació de les agències de l'empresa Zurich en funció de la seva composició. Es plantegen tres algoritmes de clustering no jeràrquic: $k-$means, $k-$medoids i *kamila*, destacant similituds i diferències entre ells. Es comparen i contrasten els diferents grups formats mitjançant test estadístics de Shapiro-Wilks i Kruskal-Wallis, així com les seves característiques més rellevants. I per últim, se n'extreuen diverses conclusions rellevants de cara a futurs passos amb les segmentacions obtingudes.

Índex

1 Introducció

1.1 Context

El sector assegurador és una de les indústries més importants d'Espanya, que genera any rere any prop del 5.5% del PIB i que, en tot el seu conjunt, conforma una de les plataformes de servei més potent de l'estat, capaç de resoldre més de 5.500 problemes, cada 60 minuts, cada dia de l'any. Són 4 els grans reptes que se li presenten al sector: la situació macroeconòmica, la tendència demogràfica amb l'envelliment poblacional, la digitalització i l'aparició de nous riscos.

El Grup Zurich és una asseguradora multiram líder, que atén els seus clients en mercats locals i globals. Amb més de 55.000 empleats, ofereix una àmplia gama de productes i serveis d'assegurances de béns i de vida, en més de 215 països i territoris. Dintre dels clients de Grup Zurich s'hi troben particulars, pimes, així com grans companyies i corporacions multinacionals. Amb més de 150 anys d'història, l'organització ha anat més enllà per protegir les persones que hi confien, reconeixent els canvis dràstics que s'estan produint a les societats, ja siguin impulsats pel canvi climàtic o pels efectes de la tecnologia a la feina i en la forma en què es viu. L'estratègia del Grup Zurich posiciona l'organització per l'èxit a llarg plaç, basant-se en la seva sòlida posició financera, la seva cartera equilibrada i la seva marca de confiança, així com en les habilitats, punts forts i experiència personal.

Grup Zurich té el seu origen a Suïssa a l'any 1872. Dotze anys més tard, al 1884 va obrir la seva primera oficina a Barcelona. Zurich Insurance PLC España (Zurich, d'ara en endavant), tal i com se la coneix actualment, és el resultat de 22 adquisicions, entre les quals s'hi troben les companyies Hispania, Vita o Eagle Star, que són les que l'han acabant convertint en la companyia asseguradora que és avui dia. Durant els darrers 5 anys, Zurich ha entrat tant en l'era de les assegurances digitals (amb Zurich Klinc, amb l'objectiu d'oferir assegurances de dispositius electrònics, patinets i bicicletes al públic millenial) com a nous sectors (amb Orange o MediaMarkt, aliança estratègica com a mostra de transformació del sector, amb l'objectiu de fer noves propostes al client, amb una major component de digitalització i personalització). La Figura 1 presenta de forma esquemàtica aquesta evolució.

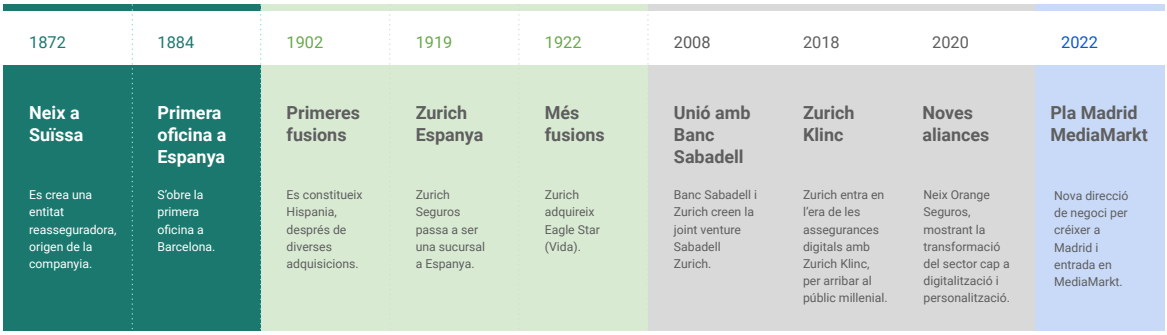


Figura 1: Evolució del negoci Zurich.

Zurich compta actualment amb més de 2.000 empleats i cobreix les necessitats de més de 1.1M de clients, que estan gestionats a través de la distribució o intermediaris, que són aquelles persones (naturals o jurídiques) que tenen el deure fiducidiari amb els clients durant el seu cicle de vida a la companyia (tenint la responsabilitat legal d'actuar exclusivament amb el millor interés del client, oferint tant la venda de pòlisses com un servei continu de sòlida relació amb els assegurats, estant disponibles per respondre les seves consultes i sol·licituds de servei). Zurich compta amb més de 3.800 intermediaris, dintre dels quals s'hi troben els

+750 agents (representants exclusius de la companyia), els +2.900 corredors (representants independents i autònoms, que treballen tant per Zurich com per altres companyies asseguradores) i els col·lectius (representants de col·legis professionals i associacions).

1.2 Objectius

Un dels principals focus de l'empresa és impulsar l'evolució del seu canal de distribució, principalment, amb els intermediaris exclusius (agents). En un context altament competitiu, amb +200 empreses asseguradores al mercat espanyol (veure Figura 2) cada vegada és més important la diferenciació respecte la competència, ja sigui a través de la personalització de pòlisses, amb tecnologia o plataformes digitals avançades, oferint un servei excepcional al client o a través de programes d'incentius o recompenses. Per aquest motiu, es requereix tenir un coneixement clar de com és el canal agencial per poder-lo impulsar, detectar sinèrgies i dur a terme accions de màrketing més eficients. És per això que es proposa generar una segmentació que tingui en compte el comportament individual de cada agència.

D'aquesta manera, els objectius són:

- Realitzar una segmentació d'agències fent servir tres mètodes de clustering diferents, valorant la consistència dels resultats de cadascun.
- Caracteritzar els grups obtinguts amb cada metodologia.
- Comprovar que els grups són efectivament diferents i estadísticament significatius.
- Valorar la implementació dels grups obtinguts a nivell negoci.

Figura 2: Representació d'algunes de les companyies d'assegurances amb major presència al mercat espanyol. Imatge extreta de [1].

2 Metodologia

Amb l'objectiu de fer una segmentació de les agències de Zurich, s'ha procedit a confeccionar una base de dades amb informació per cada agència individual, principalment relativa a la seva cartera (és a dir, la tipologia de clients que porta i la quantitat de pòlisses i primes que gestiona i sobre quins productes).

2.1 BBDD

El dataset utilizat per portar a terme l'anàlisi ens permet conèixer la composició d'una agència a través de com està distribuïda la seva cartera (clients, pòlisses, primes i productes), tant amb nova producció, renovació/retenció de cartera com pèrdua de la mateixa. Conté tant variables numèriques com categòriques. Per la seva obtenció, s'ha hagut de portar a terme una consolidació de dades a nivell agència, amb un únic registre (fila) per cadascuna d'ella. S'ha partit de la composició de l'agència per una banda a nivell client (compteig de clients únics per trams d'edat, per vinculació (com és el client segons els diferents rams de productes contractats) i per antiguitat), i per una altra banda a nivell producció (pòlisses i primes) i producte. Per últim, s'han fusionat els datasets de clients i de producció a nivell agència (codi d'identificació fiscal) fins obtenir les 43 variables numèriques que configuren la BBDD, que estan definides a la Taula 1. Respecte les altres dues variables categòriques (`dt` i `digi`): la primera és intrínseca de la pròpia agència (es tracta de la regió on s'hi troba localitzada), mentre que la segona s'ha obtingut mitjançant una enquesta llençada a les agències on indicaven en una escala de l'1 al 5 el seu nivell de digitalització.

2.2 Clustering

L'anàlisi *clustering* és un mètode d'anàlisi de dades multivariant (MDA) que té com a principal objectiu proporcionar per un conjunt d'individus una representació sintètica dels mateixos en diferents grups: els individus queden separats en grups, mutuament exclusius (per norma general), de tal manera que cada individu és més similar a la resta d'individus del seu grup que a altres individus fora del grup. Dintre del *Machine Learning* (o aprenentatge automàtic) és considerat (veure Figura 3) un algoritme de tipus *no supervisat*, és a dir, pertany al conjunt d'algoritmes que busquen patrons a les dades sense cap coneixement previ dels mateixos, a diferència dels algoritmes de tipus *supervisat*, on es fan servir els resultats ja coneguts d'un conjunt d'inputs per predir els resultats d'altres inputs. Per tant, el nombre de grups o clusters que existeixen entre els individus és desconegut abans d'executar el model. Al contrari que la majoria de mètodes estadístics, s'utilitza típicament quan no hi ha cap assumpció envers les relacions probables entre les dades. Proporciona informació sobre les associacions i patrons presents a les dades, però no què signifiquen.

Les aplicacions de clustering abasten un gran nombre d'àrees, des de marketing (per identificar patrons de comportament de diferents grups de clients), passant per web (per agrupar pàgines web basades en el seu contingut o grups d'usuaris segons els patrons d'accés a les mateixes) fins la bioinformàtica (per agrupar grups de proteïnes similars respecte la seva estructura química i/o funcionalitat).

Generalment, tot anàlisi clustering passa per respondre preguntes de l'estil *quants grups tenen les dades?*, o *què és una bona partició dels objectes?*, o *quants mètodes hi ha per fer clustering i quin és el millor?*, fins obtenir els resultats i preguntar-se *són aquests grups 'reals' o es tracta d'una agrupació artificial sense cap sentit per l'àrea on s'estudien?*.

Variable	Tipus	Descripció
cif	id	Codi d'identificació fiscal de l'agència.
polisses_cartera	num	Nombre de pòlisses en cartera.
primes_cartera	num	Nombre de primes en cartera.
primes_np	num	Nombre de primes de nova producció.
polisses_np	num	Nombre de pòlisses de nova producció.
polisses_retingudes	num	Nombre de pòlisses retingudes.
polisses_inici	num	Nombre de pòlisses a l'inici.
clients_menys_1a	num	Nombre de clients en cartera amb menys d'1 any d'antiguitat.
clients_1a_2a	num	Nombre de clients en cartera amb entre 1 i 2 anys d'antiguitat.
clients_2a_3a	num	Nombre de clients en cartera amb entre 2 i 3 anys d'antiguitat.
clients_3a_6a	num	Nombre de clients en cartera amb entre 3 i 6 anys d'antiguitat.
clients_6a_10a	num	Nombre de clients en cartera amb entre 6 i 10 anys d'antiguitat.
clients_10a_20a	num	Nombre de clients en cartera amb entre 10 i 20 anys d'antiguitat.
clients_mes_20a	num	Nombre de clients en cartera amb més de 20 anys d'antiguitat.
clients_edat_menys_25	num	Nombre de clients en cartera amb menys de 25 anys d'edat.
clients_edat_25_34	num	Nombre de clients en cartera amb entre 25 i 34 anys d'edat.
clients_edat_35_44	num	Nombre de clients en cartera amb entre 35 i 44 anys d'edat.
clients_edat_45_54	num	Nombre de clients en cartera amb entre 45 i 54 anys d'edat.
clients_edat_55_64	num	Nombre de clients en cartera amb entre 55 i 64 anys d'edat.
clients_edat_65_74	num	Nombre de clients en cartera amb entre 65 i 74 anys d'edat.
clients_edat_mes_74a	num	Nombre de clients en cartera amb més de 74 anys d'edat.
clients_edat_fisica_nd	num	Nombre de clients en cartera sense edat informada.
clients_edat_juridica_nd	num	Nombre de clients en cartera de personalitat jurídica.
clients_monoram_mono	num	Nombre de clients en cartera amb una única pòlissa.
clients_monoram_multi	num	Nombre de clients en cartera amb més d'una pòlissa, però del mateix ram.
clients_multiram_multi	num	Nombre de clients en cartera amb més d'una pòlissa de més d'un ram.
clients_inici	num	Nombre de clients en cartera a l'inici del periode.
clients_nous	num	Nombre de clients nous.
clients_cartera	num	Nombre de clients en cartera.
clients_perduts	num	Nombre de clients perduts al periode.
polisses_cartera_AUTO	num	Nombre de pòlisses del ram auto en cartera.
polisses_cartera_HOGAR	num	Nombre de pòlisses del ram hogar en cartera.
polisses_cartera_COMERÇOS	num	Nombre de pòlisses del ram comerços en cartera.
polisses_cartera_PYMES	num	Nombre de pòlisses del pimes en cartera.
polisses_cartera_SPECIALTIES	num	Nombre de pòlisses de productes especialistes en cartera.
polisses_cartera_VIDA	num	Nombre de pòlisses de vida en cartera.
polisses_cartera_REST	num	Nombre de pòlisses d'altres rams en cartera.
primes_cartera_AUTO	num	Nombre de primes del ram auto en cartera.
primes_cartera_HOGAR	num	Nombre de primes del ram hogar en cartera.
primes_cartera_COMERÇOS	num	Nombre de primes del ram comerços en cartera.
primes_cartera_PYMES	num	Nombre de primes del ram pimes en cartera.
primes_cartera_SPECIALTIES	num	Nombre de primes de productes especialistes en cartera.
primes_cartera_VIDA	num	Nombre de primes de vida en cartera.
primes_cartera_REST	num	Nombre de primes d'altres rams en cartera.
dt	cat	Distribució territorial a què pertany l'agència.
digi	cat	Nivell de digitalització que presenta l'agència.

Taula 1: Descripció i tipus de variables presents a la BBDD. `num` fa referència a variable numèrica, `cat` fa referència a variable categòrica i `id` fa referència a variable indicador.

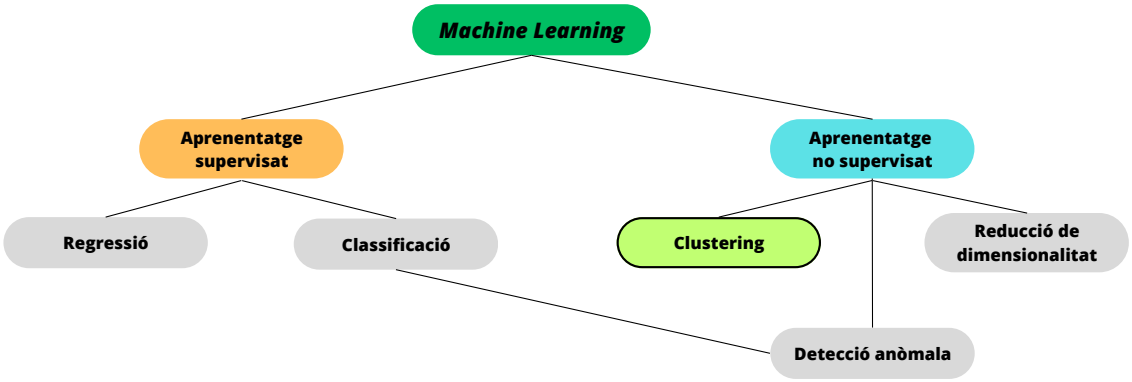


Figura 3: Representació esquemàtica dels principals algoritmes de Machine Learning.

El dataset bàsic per la majoria d'aplicacions de clustering es tracta d'una matriu multivariant $\boldsymbol{X}$ de dimensions $n \times p$, $\boldsymbol{X} = (x_{ij})$, on cada entrada x_{ij} d'$\boldsymbol{X}$ dóna el valor de la $j-$èssima variable de l'individu i.

$$\boldsymbol{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & \cdots & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ x_{n1} & \cdots & \cdots & x_{np} \end{pmatrix}$$

Les variables a $\boldsymbol{X}$ poden ser tant contínues, ordinals com categòriques, on algunes de les seves entrades poden estar buides. És justament el fet de tenir entrades buides i variables mixtes el que pot complicar l'anàlisi clustering, com és el cas que ens pertoca.

Sovint s'acostuma a representar gràficament les dades multidimensionals, ja que generalment, poden ser d'utilitat per suggerir l'existència de clusters o grups entre els individus o relacions entre les variables. Aquest gràfic es tracta típicament d'un núvol de punts que evoluciona dintre d'un espai Euclidià (que pot ser reduït a una representació en dues dimensions), i les distàncies entre els punts s'interpreten en termes de similaritat pels individus (agrupacions no jeràrquiques). No obstant, altres visualitzacions (com les agrupacions jeràrquiques d'arbre) són també habituals a la literatura per representar similaritats o correlacions entre un grup d'individus, comunament utilitzats a les companyies i administracions per què cadascú conegui on s'hi troba (nivell) respecte l'organització o un arbre genealògic per descriure les relacions entre avant-passats.

No obstant, cal tenir en compte que previ a l'aplicació de qualsevol de les diferents metodologies de clustering és molt convenient avaluar si el conjunt de dades és clusteritzable, és a dir, provar la hipòtesi de l'existència de patrons a les dades respecte un conjunt de dades distribuïdes uniformement. Aquesta verificació pot portar-se a terme estadísticament, mitjançant l'*estadístic Hopkins H* [4] que permet avaluar la tendència del clustering d'un conjunt de dades mitjançant el càlcul de probabilitat de que aquestes dades procedeixin d'una distribució uniforme. El procediment per obtenir-lo segueix els següent passos:

1. S'extreu una mostra d'n observacions $p_1, \ldots, p_n$ de la base de dades proporcionades. I per cada observació p_i, es troba quina és l'observació p_j més propera, i es calcula la distància entre les dues, $x_i = d(p_i, p_j)$.

2. Es simula una mostra d'una distribució uniforme, també d'n observacions $q_1, \ldots, q_n$ amb la mateixa variació que les dades originals. De nou, per cada observació q_i, es troba quina és l'observació q_j més propera, i es calcula la distància entre les dues, $y_i = d(p_i, p_j)$.

3. Es calcula l'estadístic Hopkins H seguint l'expressió següent:

$$H = \frac{\sum_{i=1}^{n} y_i}{\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} y_i}. \tag{1}$$

La interpretació d'H segueix el següent racional: si les dades del dataset original estiguessin uniformement distribuïdes, aleshores $\sum_{i=1}^{n} x_i \simeq \sum_{i=1}^{n} y_i$, i per tant, $H \simeq 0.5$. No obstant, si efectivament hi ha clusters al dataset original, aleshores $\sum_{i=1}^{n} x_i \ll \sum_{i=1}^{n} y_i$, i per tant $H \gg 0.5$. Valors d'$H > 0.75$ indiquen una tendència de clustering al 90% CI.

Existeixen, essencialment, dues tipologies de clustering: els algoritmes jeràrquics i els no jeràrquics.

2.2.1 Clustering jeràrquic

Els algoritmes *jeràrquics* permeten organitzar els individus de forma jeràrquica. Els individus s'acostumen a representar gràficament mitjançant una estructura d'arbre o dendograma: cada node representa un cluster d'objectes, i els nodes es van fusionant recursivament fins formar clusters més grans a mesura que es "puja" en l'arbre. D'aquesta manera, un dendograma s'encarrega d'il·lustrar les fusions o divisions que es fan a cada etapa de l'anàlisi.

Dintre dels algoritmes jeràrquics hi trobem els mètodes aglomeratius (partint de cada individu com un únic cluster, aquests es van fusionant successivament per similaritat fins compondre el conjunt complet d'individus) i els mètodes divisius (a partir d'un cluster inicial amb tot el conjunt d'individus, aquest es va dividint successivament en clusters més petits, fins tenir un cluster per cada individu). S'ha representat esquemàticament la diferència a la Figura 4. Amb els mètodes jeràrquics, tan bon punt les divisions/fusions es produeixen, aquestes són irrevocables, amb la qual cosa quan l'algoritme aglomeratiu ha unit dos individus aquesta assignació és immutable i ja no es podran separar, i quan l'algoritme divisiu els ha separat, ja no es podran unir.

El procediment per portar a terme els algoritmes jeràrquics passa per computar i emmagatzemar la matriu d'interdistàncies entre els diferents individus. Aquest càlcul suposa un alt cost computacional, éssent recomanable la seva utilització quan la base de dades amb què es treballa contingui un nombre reduït de registres. És important tenir en compte el tipus distància, ja que és el propi usuari qui l'ha d'escollir, tenint present que ha de ser aquella millor hi caracteritzi les similiaritats i dissimilaritats entre els individus (tant pot ser distàncies de la família d'Euclídies, o de perfils (com per exemple casos on el valor 0 sigui important a nivell qualitatiu) o distàncies estadístiques (com per exemple amb el cas de taules de contingència o anàlisis de correspodències)). Un cop escollida la mètrica de distància, s'ha de també escollir la distància inter-cluster de cara a anar fent les sucessives agrupacions: tenint dos clusters C_i i C_j, es pot tenir en compte la mínima distància entre qualsevol dels membres dels clusters (*single-link distance*), o la màxima (*complete-link distance*), o el promig entre tots els elements de cada cluster (*group average distance*), o la distància *Ward*, que contempla només aquelles agrupacions tals que es tingui el menor increment al valor

total de la suma de quadrats de les diferències dintre de cada cluster i la mitjana del cluster on s'integra. Escollir una o altra distància per agrupar comporta resultats molt diferents, i aquesta subjectivitat és única d'aquest tipus d'algoritmes.

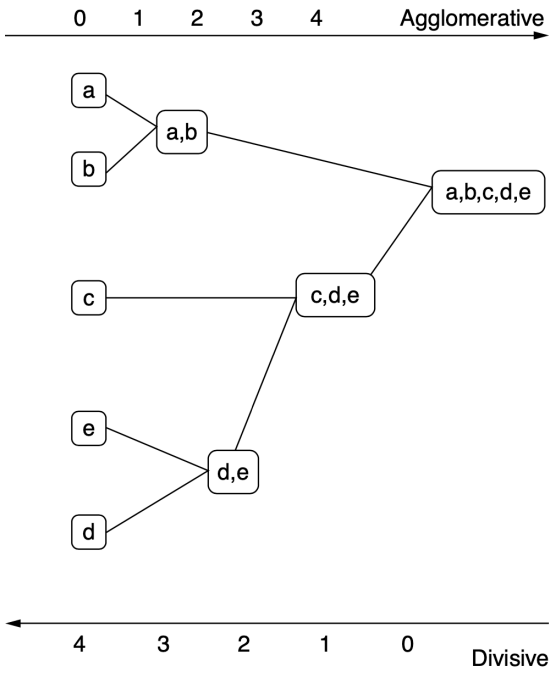


Figura 4: Representació esquemàtica dels mètodes aglomeratius i divisius als algoritmes jeràrquics. Extreta de [4].

Atès que tots els mètodes aglomeratius tenen com últim pas compondre un únic cluster amb tots els individus i els mètodes divisius n clusters, cadascun amb un invidividu, la solució del nombre "òptim" de clusters vindrà donada pel punt/alçada on es decideixi fer el tall al dendograma, i no existeixen mètodes generalitzats per trobar aquest tall, ja que depenen implícitament de les formes que tinguin els clusters formats.

Donada la subjectivitat del clustering jeràrquic i les dimensions del dataset proporcionat per elaborar el present treball i estudi estadístic, s'ha decidit no utilitzar aquesta metodologia.

2.2.2 Clustering no jeràrquic: $k-$means

Els algoritmes de clustering no jeràrquic es basen en construir una partició d'una base de dades d'n objectes en k clusters $(\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k)$ de tal manera que minimitzen la distància total intra-cluster o el TESS (Total Error Sum of Squares, suma de l'error quadrat):

$$\text{TESS} = \sum_{i=1}^{k} \sum_{j \in \mathcal{C}_j} d^2(x_{ij} - c_j), \tag{2}$$

on d^2 és la mètrica de distància euclídia, $\mathcal{C}_j$ indica el conjunt de punts al cluster j i c_j és el seu centroide/medoide. Per poder trobar la solució exacta caldria tenir en compte el nombre total de possibles combinacions que hi ha de que n elements puguin pertànyer a k, és a dir, $C(n,k) = \frac{1}{k!} \sum_{j=1}^{k} (-1)^{k-j} \binom{k}{j} j^n \approx \frac{k^n}{k!}$. Per exemple, tenint en compte un dataset petit de $n = 25$ individus amb $k = 8$ grups, $C(25, 8) \simeq 6.9 \cdot 10^{17}$, cosa que comporta un gran cost computacional i requereix portar a terme mètodes de resolució heurística.

Un avantatge, doncs, dels mètodes no jeràrquics respecte els jeràrquics és el fet que els darrers són computacionalment més eficients, atès que no requereixen emmagatzemar les matrius de distàncies entre elements que sí requereixen els darrers.

$k-$means és, per excel·lència, l'heurística més utilitzada per fer clustering no jeràrquic quan es tracta amb un dataset conformat per variables quantitatives, on cada cluster o grup ve representat pel seu centre. Consisteix en els següents passos (veure Figura 5):

i. S'escull el valor de k (nombre de clusters).

ii. S'inicialitzen de forma aleatòria els k centres de cluster (Figura 5b).

iii. S'associa un cluster per cadascun dels n individus a aquell que s'hi trobi més a prop (amb una mitjana aritmètica) (Figura 5c).

iv. Es re-estimen els centres del clusters, assumint que la pertinença dels individus als mateixos és correcta (Figura 5d).

v. Si cap dels n objectes canvia de grup en la darrera iteració, la classificació quedaria acabada. (Figura 5f). En cas contrari, es torna a iii.

Generalment s'acostuma a computar $\sim 10^4$ simulacions i és la més probable la que s'escull.

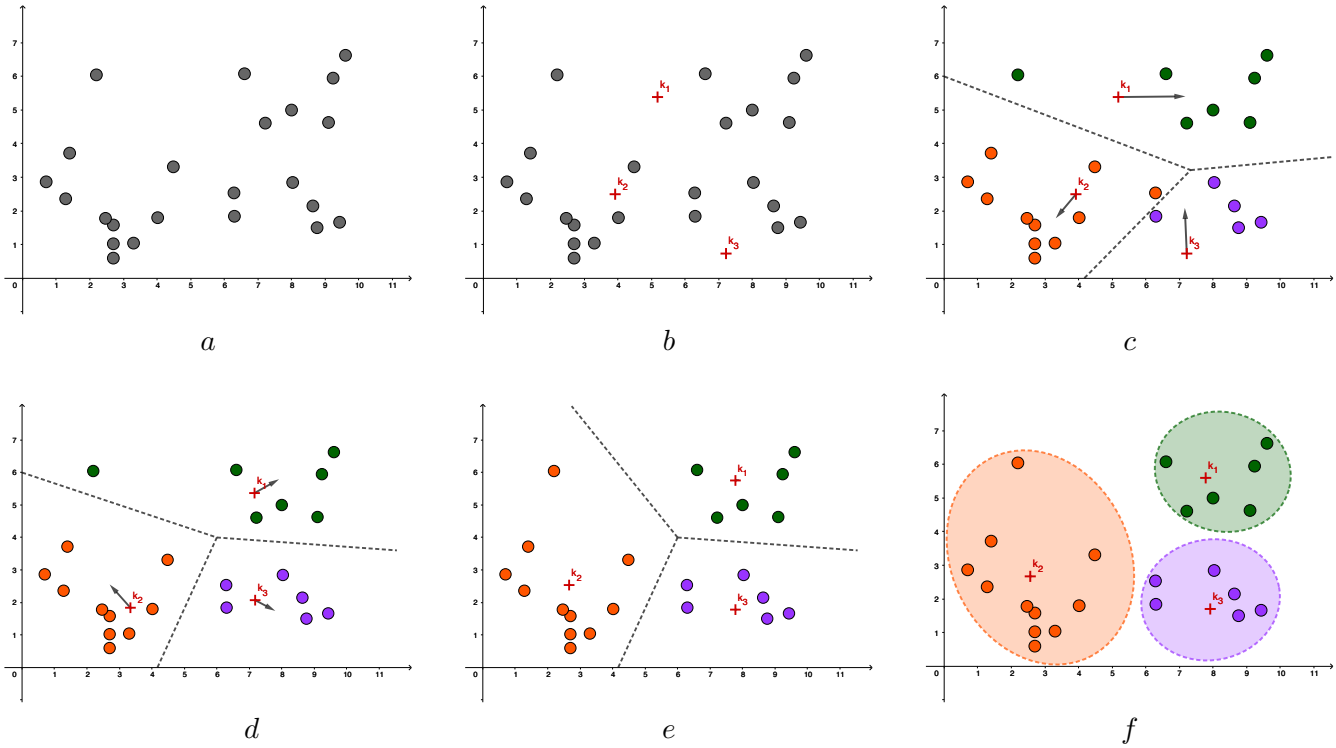


Figura 5: Representació gràfica de l'evolució dels passos de l'algoritme $k-$means.
a: representació dels individus. *b*: inicialització aleatòria dels clusters. *c*: associació dels individus al cluster més proper. *d*: re-estimació dels clusters. *e*: re-associació dels individus al cluster més proper. *f*: representació definitiva dels parells "individu-cluster".

Un cop feta la segmentació, diferents criteris/mètriques es fan servir per avaluar la validesa dels grups formats:

- **Gradent TESS**: Es basa en escollir el valor k de clusters on hi hagi una maximització del gradent TESS:

$$\Delta(k-1,k) = \frac{\text{TESS}(k-1) - \text{TESS}(k)}{\text{TESS}(k-1)}, \tag{3}$$

  on la representació de l'evolució d'aquest gradent respecte al nombre de clusters és monòtonament decreixent. Per escollir el valor k òptim s'acostuma a fer servir l'*Elbow rule* o regla del colze, prenent per k aquell valor que marqui gràficament un "canvi" significatiu com si d'un colze es tractés.

- **Silhouettes**: L'estadístic de *Silhouettes* $\bar{s}$ proporciona quant semblant un individu és respecte al seu grup en comparació als altres grups. Es basa en la diferència d'inter-distàncies de cada individu ω_i a la seva partició (C) i a la partició del cluster més proper. El nombre de clusters k que s'escull és aquell que maximitza $\bar{s}$:

$$\bar{s} = \frac{1}{n}\sum_{i=1}^{n} s(\omega_i) = \frac{1}{n}\sum_{i=1}^{n} \frac{b(\omega_i) - a(\omega_i)}{\max\{a(\omega_i), b(\omega_i)\}}, \tag{4}$$

  on

$$a(\omega_i|\omega_i \in C_i) = \frac{1}{|C_i|}\sum_{\omega_i \in C_i} d^2(\omega_i, \omega_j), \qquad b(\omega_i|\omega_i \in C_i) = \min_{s \neq i}\left\{\frac{1}{|C_i|}\sum_{\omega_s \in C_s} d^2(\omega_i, \omega_s)\right\}.$$

- **Estadístic Pseudo*F***: L'*estadístic PseudoF* és el ratio entre la suma d'errors entre grups i la suma d'errors intragrup. El nombre de clusters k que s'escull és aquell que maximitza $F(k)$:

$$F(k) = \frac{\sum_{i=1}^{k} d^2(c(i) - \bar{c})/(k-1)}{\sum_{i=1}^{k}\sum_{j=1}^{|C_j|} d^2(\omega_{ij} - c(i))/(n-c)}. \tag{5}$$

Per últim, esmentar que existeixen altres criteris per determinar el nombre òptim de clusters que no s'han utilitzat en el present treball, com és el cas del *GAP estadístic*, que en síntesi compara la dispersió entre les dades originals (variació intragrup) respecte les que es podria esperar d'un conjunt de dades generades aleatòriament. Per les dades observades i les dades de referència, la variació intragrup es calcula amb diferents valors de k, fent servir l'expressió següent,

$$\text{GAP}_n(k) = \mathbb{E}_n^*(\log(W_k)) - \log(W_k), \tag{6}$$

on $\mathbb{E}_n^*$ denota l'esperança sota una mostra de mida n de la distribució de referència. Aquest estadístic mesura la desviació del valor W_k observat respecte el valor esperat de W_k sota la hipòtesi nul·la. D'aquesta manera, el valor k escollit seria aquell en què es maximitzi aquest gap, significant que l'estructura de l'agrupació amb k grups està molt allunyada de la distribució uniforme i aleatòria de punts.

2.2.3 Clustering no jeràrquic: $k-$medoids

Donada la sensibilitat que $k-$means té amb els outliers (valors inusuals, que es desvien significativament de la resta) es va evolucionar cap a l'algoritme $k-$medoids: en comptes de calcular la mitjana per cadascun dels clusters per determinar el seu centroide, es calcula el seu medoide (veure Figura 6), que és l'individu la distància mitjana del qual respecte la resta dels punts del cluster és mínima. D'aquesta manera és un element del propi cluster

el que reuneix les característiques idònies del grup a què pertany. La diferència primordial amb $k-$means es basa amb la distància a minimitzar: es tracta de la distància Manhattan en comptes de l'Euclídia.

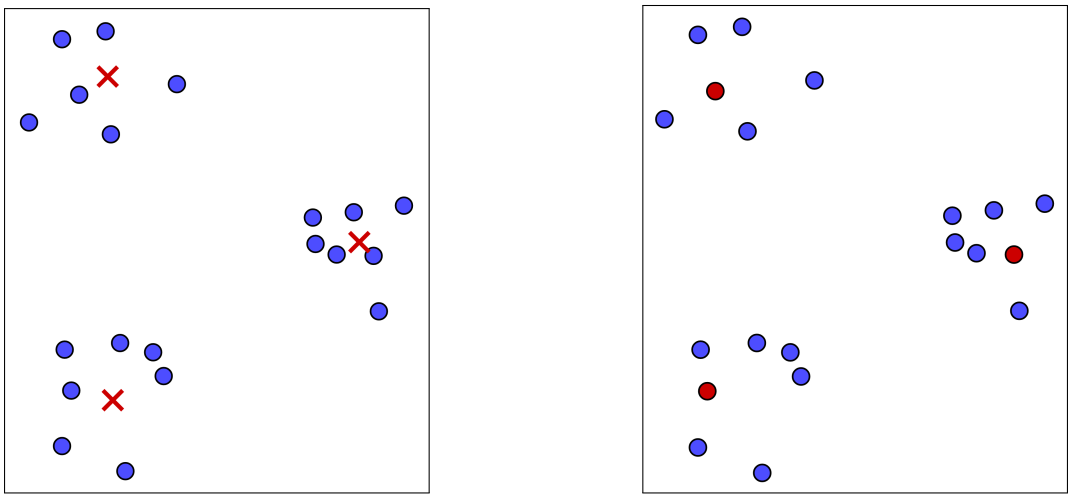


Figura 6: Representació gràfica dels algoritmes $k-$means i $k-$medoids. En vermell, es representen, respectivament, els centroides (esquerra) i medoides (dreta).

2.2.4 Clustering amb dades mixtes: *kamila*

Tot i l'existència d'una àmplia varietat d'algoritmes de clustering per dades numèriques, són menys freqüents aquells que involucrin dades mixtes, és a dir, variables tant numèriques com categòriques. En la majoria d'aquests casos s'acostuma a abordar les variables categòriques aplicant una codificació *dummy* per convertir-les en numèriques i així poder aplicar posteriorment qualsevol dels algoritmes explicats a les seccions anteriors. Aquesta codificació funciona de la manera següent: donada una variable categòrica X_j amb ℓ categories diferents, es creen ℓ diferents noves variables indicador $0-c$, una per cada categoria, on 0 indica absència i $c \in \mathbb{R}$ indica la presència d'aquell nivell o categoria a X_j. La dificultat apareix en la selecció de c, ja que valors elevats (petits) sobreponderaran (infravaloraran) les variables categòriques sobre les numèriques.

Altres estratègies involucren l'ús de *dissimilaritats* específiques compatibles amb dades mixtes, com és el cas de la distància de Gower d_G [7]: considerem dos vectors $p-$dimensionals $\boldsymbol{x}, \boldsymbol{y}$ amb variables mixtes.

$$\text{Definim: } f_j(x_j, y_j) = \begin{cases} |x_j - y_j|/r_j & \text{si} \quad x_j, y_j \text{ són numèriques} \\ 0 \quad \text{si} \quad x_j = y_j & \\ 1 \quad \text{si} \quad x_j \neq y_j & \text{si} \quad x_j, y_j \text{ són categòriques} \end{cases}$$

Aleshores, la distància de Gower es defineix com

$$d_G = \frac{\sum_{j=1}^{p} w_j f_j(x_j, y_j)}{\sum_{j=1}^{p} w_j}, \tag{7}$$

on r_j és el rang de la variable $j-$èssima i w_j és un pes definit per l'usuari per aplicar a cada element j. Però de nou, impliquen l'assignació d'un pes per la contribució específica de

cada variable a la distància, problemàtica equiparable a l'elecció de c en la codificació *dummy*.

Per aquest motiu, es proposa la utilització de la metodologia *kamila* (KAy-means for MIxed LArge data sets) [8], que és capaç d'aconseguir un balanç favorable entre variables numèriques i categòriques sense la necessitat d'atorgar pesos: les variables numèriques es modelen utilitzant distribucions el·líptiques, mentre que les categòriques es modelen com mixtures de variables aleatòries multinomials.

kamila és un algoritme de clustering que presenta un conjunt d'avantatges notables respecte a metodologies ja existents:

1. Les variables (siguin de la tipologia que siguin) s'utilitzen en la seva escala de mesura original (per tant, no calen transformacions de les mateixes, evitant així pèrdua d'informació).
2. Assegura un balanceig equitatiu entre les variables numèriques i categòriques.
3. Evita l'assumpció de restriccions paramètriques, generalitzant la forma dels clusters a distribucions el·líptiques.
4. Evita, per últim, l'especificació de pesos o codificacions per part de l'usuari.

La seva metologia es presenta a continuació: considerem un dataset amb N observacions independents i idènticament distribuïdes amb $P+Q$ atributs que segueixen una distribució de tipus mixtura amb G components, on $\boldsymbol{V}$ és un vector $P-$dimensional de variables numèriques i $\boldsymbol{W}$ un vector de Q variables categòriques, on l'element $q-$èssim de $\boldsymbol{W}$ té L_q nivells o categories, denotats per $1, 2, \ldots, L_q$, per $q \in \{1, 2, \ldots Q\}$.

Donada la pertinença al cluster g, per una banda $\boldsymbol{V}$ es modela com una mixtura de distribucions el·líptiques, on cada component individual té com a funció densitat $f_{\boldsymbol{V},g}(\boldsymbol{v}; \mu_g, \Sigma_g)$, on g denota l'índex de pertinença al cluster, μ_g el $g-$èssim centroide i Σ_g és la $g-$èssima matriu d'escalat. Per altra banda, $\boldsymbol{W}$ es modela com una mixtura de multinomials, on cada component individual té com a funció densitat $f_{\boldsymbol{W},g}(\boldsymbol{w}) = \prod_{q=1}^{Q} m(w_q; \theta_{gq})$, éssent $m(\cdot;\cdot)$ la funció multinomial de probabilitat i θ_{qq} el paràmetre multinomial per la $g-$èssima component de la $q-$èssima variable categòrica. D'aquesta manera, sota l'assumpció d'independència local, la funció de densitat conjunta de $(\boldsymbol{V}^\top, \boldsymbol{W}^\top)^\top$ té l'expressió següent:

$$f_{\boldsymbol{V},\boldsymbol{W}}(\boldsymbol{v}, \boldsymbol{w}) = \sum_{g=1}^{G} \pi_g f_{\boldsymbol{V},\boldsymbol{W},g}(\boldsymbol{v}, \boldsymbol{w}; \mu_g, \Sigma_g, \theta_{gq}), \tag{8}$$

on $f_{\boldsymbol{V},\boldsymbol{W},g}(\boldsymbol{v}, \boldsymbol{w}; \mu_g, \Sigma_g, \theta_{gq}) = f_{\boldsymbol{V},g}(\boldsymbol{v}; \mu_g, \Sigma_g) \cdot \prod_{q=1}^{Q} m(w_q; \theta_{gq})$ i π_g denota la probabilitat a priori d'observar el $g-$èssim cluster.

Els paràmetres de (8) s'estimen iterativament. Siguin $\hat{\boldsymbol{\mu}}_g^{(t)}$ i $\hat{\boldsymbol{\theta}}_{gq}^{(t)}$ els estimadors de μ_g i θ_{gq} respectivament a la iteració t. S'inicialitza $\hat{\boldsymbol{\mu}}_g^{(0)}$ per cada $g \in 1, 2, \ldots, G$ amb un sampleig de les observacions de les variables contínues i $\hat{\boldsymbol{\theta}}_{gq}^{(0)}$ per cada g i $q \in \{1, 2, \ldots, L_q\}$ amb un sampleig d'una distribució uniforme a $\mathbb{R}^{L_q}$. L'estimació es produeix de forma iterativa, amb primer un pas de partició (on s'assigna cada observació un cluster) seguit d'un pas d'estimació (es reestimen els paràmetres a partir de la pertinença al nou cluster).

Donats $\hat{\boldsymbol{\mu}}_g^{(t)}$ i $\hat{\boldsymbol{\theta}}_{gq}^{(t)}$ a la iteració t, es calcula la distància de l'observació i a cadascun dels $\hat{\boldsymbol{\mu}}_g^{(t)}$ seguint la norma Euclídia, és a dir $d_{ig}^{(t)} = \sqrt{\sum_{p=1}^{P} \left(v_{ip} - \hat{\mu}_{gp}^{(t)}\right)^2}$ i la mínima distància

$r_i^{(t)} = \min_g d_{ig}^{(t)}$ es calcula amb la funció de densitat $\hat{f}_R^{(t)}(r) = \frac{1}{Nh^{(t)}} \sum_{\ell=1}^{N} k\left(\frac{r - r_\ell^{(t)}}{h^{(t)}}\right)$, on $k(\cdot)$ és una funció tipus kernel i $h^{(t)}$ és el corresponent ample de banda a la iteració t (es fa servir un kernel Gaussià, amb $h = 0.9An^{-1/5}$, on $A = \min(\hat{\sigma}, \hat{q}/1.34)$, éssent $\hat{\sigma}$ la desviació estàndard mostral i $\hat{q}$ el rang interquartílic).
Assumint independència local a les Q variables categòriques, es calcula la probabilitat d'observar l'$i-$èssim vector de categories donada la pertinença a un grup g com $c_{ig}^{(t)} = \prod_{q=1}^{Q} m(w_{iq}; \hat{\boldsymbol{\theta}}_{gq}^{(t)})$, amb $m(\cdot;\cdot)$, de nou, la funció multinomial de probabilitat.

Aleshores, assignem l'$i-$èssim objecte al grup g que maximitzi la funció

$$H_{ig}^{(t)} = \log\left(\hat{f}_{\boldsymbol{V}}^{(t)}\left(d_{ig}^{(t)}\right)\right) + \log\left(c_{ig}^{(t)}\right). \tag{9}$$

La implementació `kamila` [9] a R atura l'algoritme quan la pertinença a un grup es manté immutable d'una iteració a la següent. En particular, fa servir les quantitats $\epsilon_{\text{con}}, \epsilon_{\text{cat}}$, generalment $\epsilon_{\text{con}}, \epsilon_{\text{cat}} \sim 0.01$ com a thresholds per aturar l'algoritme:

$$\epsilon_{\text{con}} = \sum_{g=1}^{G}\sum_{p=1}^{P}\left|\hat{\mu}_{gp}^{(t)} - \hat{\mu}_{gp}^{(t-1)}\right|, \qquad \epsilon_{\text{cat}} = \sum_{g=1}^{G}\sum_{q=1}^{Q}\sum_{\ell=1}^{L_q}\left|\hat{\theta}_{gq\ell}^{(t)} - \hat{\theta}_{gq\ell}^{(t-1)}\right|.$$

Per últim, un cop s'aturen les iteracions, s'escull la millor de les particions en funció d'una funció objectiu. En el cas de la implementació `kamila` a R es basa en la minimització de $Q_{\text{con}} \times \left(-\log c_{ig}^{(t)}\right)$, on $Q_{\text{con}} = W_{\text{con}}/B_{\text{con}}$ és el ratio de distàncies intra i inter cluster.

3 Resultats i discussió

3.1 BBDD + EDA

Les Taules 3 i 2a presenten el resum estadístic de les variables tant numèriques com categòriques, respectivament, de les 781 agències. En el cas de les categòriques podem apreciar que tenim 6 categories a DT (Catalunya, Este Baleares, Madrid, Sur Canarias, Norte i Centralizados), mentre que al nivell de digitalització en tenim 5, ordenats d'1 a 5 en ordre creixent (aquelles agències més digitals presenten un 5 i aquelles que menys presenten un 1). Podem apreciar, també, que 170 agències no disposen d'aquest valor, i totes pertanyen a la DT Centralizados.

Cal tenir en compte que, per construcció, algunes de les variables són combinació lineals d'altres, és a dir:

$$\begin{aligned}\text{Clients cartera} &= \sum_{i=1}^{9} \text{Clients segons el seu tram d'edat} \\ &= \sum_{i=1}^{7} \text{Clients segons el seu tram d'antiguitat} \\ &= \sum_{i=1}^{3} \text{Clients segons la seva vinculació}\end{aligned}$$

$$\text{Pòlisses (primes) cartera} = \sum_{i=1}^{7} \text{Pòlisses (primes) segons la línia de negoci,}$$

i que per simplicitat s'ha evitat generar tots els productes cartesians entre les diferents variables (és a dir, compteig de pòlisses d'una certa línia de negoci pels clients d'una certa edat, certa antiguitat i certa vinculació).

Variable	Categories	N	Percent
dt	Catalunya	113	14.5%
	Este Baleares	218	27.9%
	Madrid	31	4.0%
	Sur Canarias	145	18.6%
	Norte	99	12.7%
	Centralizados	175	22.4%
digi	1	233	29.8%
	2	69	8.8%
	3	257	32.9%
	4	50	6.4%
	5	2	0.3%
	<NA>	170	21.8%

	1	2	3	4	5	<NA>
Catalunya	52	10	45	6	0	0
Este Baleares	94	26	81	16	1	0
Madrid	11	3	14	3	0	0
Sur Canarias	44	17	70	13	1	0
Norte	27	13	47	12	0	0
Centralizados	5	0	0	0	0	170

Taula 2: Taula resum de les variables categòriques. A l'esquerra (a), resum estadístic, a la dreta (b), taula de contingència (en files les DT, en columnes el nivell de digitalització).

Variable	$\bar{x}$	σ	Min	Q1	Q2	Q3	Max
polisses_cartera	839	1.134	1	122	463	1.073	9.708
primes_cartera	330.528	470.411	0	45.832	162.774	396.581	3.948.468
primes_np	51.064	159.315	0	2.718	16.536	47.750	3.595.319
polisses_np	116	177	0	9	51	141	1.549
polisses_retingudes	722	981	0	103	394	938	8.342
polisses_inici	834	1.126	0	123	445	1.065	9.611
clients_menys_1a	38	73	0	3	17	43	1.147
clients_1a_2a	37	61	0	3	17	46	728
clients_2a_3a	30	47	0	2	14	38	504
clients_3a_6a	76	126	0	6	36	94	1.747
clients_6a_10a	70	109	0	5	32	89	1.277
clients_10a_20a	116	165	0	11	57	158	1.660
clients_mes_20a	130	187	0	9	59	182	1.439
clients_edat_menys_25	3	8	0	0	1	3	97
clients_edat_25_34	20	37	0	1	7	22	387
clients_edat_35_44	55	88	0	5	25	68	792
clients_edat_45_54	94	134	0	11	47	112	1.240
clients_edat_55_64	99	138	0	15	53	125	1.637
clients_edat_65_74	68	95	0	10	39	87	1.316
clients_edat_mes_74a	54	75	0	6	30	72	926
clients_edat_fisica_nd	49	73	0	6	23	60	652
clients_edat_juridica_nd	54	76	0	7	26	68	542
clients_monoram_mono	315	404	0	59	178	410	4.310
clients_monoram_multi	67	100	0	7	35	84	1.155
clients_multiram_multi	116	167	0	11	56	142	1.489
clients_inici	498	648	0	87	270	663	6.629
clients_nous	47	87	0	4	22	55	1.271
clients_cartera	498	648	1	85	269	654	6.633
clients_perduts	48	67	0	8	27	61	732
polisses_cartera_AUTO	417	675	0	29	193	516	7.637
polisses_cartera_HOGAR	251	335	0	30	135	316	2.786
polisses_cartera_COMERÇOS	33	52	0	2	14	39	375
polisses_cartera_PYMES	14	27	0	1	5	15	250
polisses_cartera_REST	21	39	0	1	7	24	559
polisses_cartera_SPECIALTIES	42	68	0	3	17	47	508
polisses_cartera_VIDA	60	102	0	3	22	76	1.500
primes_cartera_AUTO	136.740	203.668	0	10.251	65.490	164.838	1.755.831
primes_cartera_HOGAR	64.298	90.315	0	7.094	33.450	82.004	644.591
primes_cartera_COMERÇOS	16.081	25.566	0	902	6.418	19.946	204.693
primes_cartera_PYMES	22.040	54.732	0	294	4.655	19.335	687.948
primes_cartera_REST	18.375	31.468	0	435	5.431	21.318	253.832
primes_cartera_SPECIALTIES	20.911	45.291	0	615	4.995	18.181	518.713
primes_cartera_VIDA	52.082	171.660	0	1.959	11.549	45.091	3.707.564

Taula 3: Resum estadístic de les 43 variables numèriques.

Seguidament s'ha pogut comprovar que les variables presenten rangs amb diferents ordres de magnituds: pòlisses i clients totals en cartera estan al mateix ordre de magnitud, però aquestes tenen unes primes que poden moure's en els rangs dels centenars, mil·lers i fins i tot desenes i centenars de mil·lers (l'assegurança d'un producte de motor no és de l'ordre de l'assegurança del conglomerat d'una empresa), raó per la qual s'ha decidit escalar les variables per evitar que aquelles amb magnituds superiors (primes) sobreponderin sobre d'altres amb ordres de magnitud inferior (nombre de clients). La Figura 7 justament exemplifica el que s'acaba de comentar: s'ha representat la distribució de clients, pòlisses i primes en cartera per cadascuna de les agències segons la DT a què pertanyen.

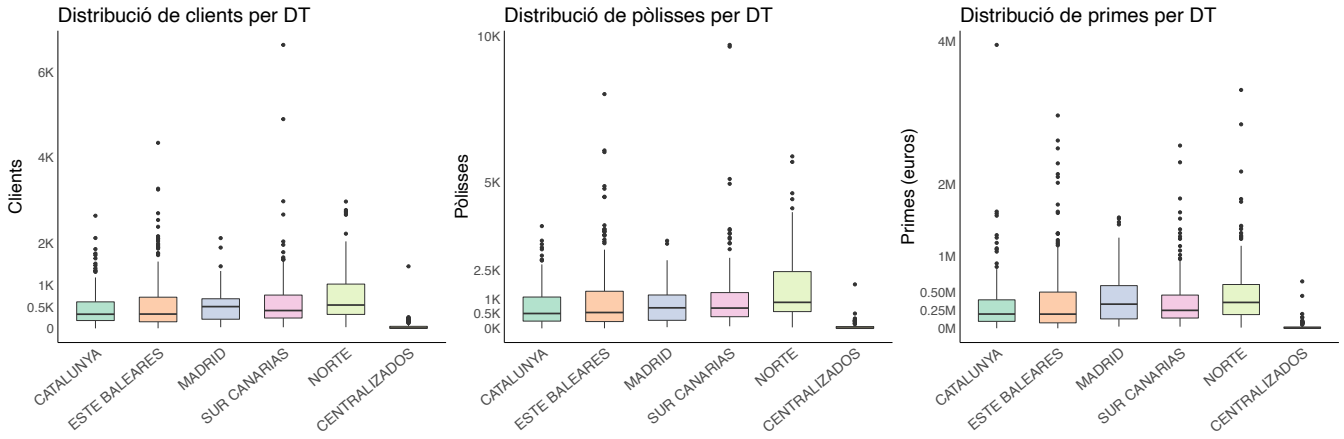


Figura 7: Distribució per DT de clients, pòlisses i primes.

Podem apreciar (detall a la Taula 3) que mentre que la mediana de les agències es mou amb carteres de l'ordre de 269 clients, amb 463 pòlisses i facturacions de 163K€, trobem outliers en cadascuna de les regions, amb valors extrems d'una agència de la regió Sur Canarias, amb més de 6K clients i 10K pòlisses, tot i que no és la que més factura (l'agència amb major facturació s'hi troba a la zona de Catalunya, amb un volum de primes rondant els 4M€). Per una altra banda, les agències de la DT Centralizados presenten en les tres variables valors significativament més petits que la resta de les DT i que és la DT Norte la que té, en promig, la mediana dels seus valors de clients, pòlisses i primes en cartera per agència superior a la resta de les DT.

El nombre total d'agències s'ha representat a la Figura 8, així com el total de clients i pòlisses que totes elles en conjunt gestionen. Es pot apreciar clarament la corroboració que hem exemplificat amb la Figura 7: les agències de la DT Centralizados, tot i suposar el 22% del total del canal agencial gestionen poc més del 2% de clients i quasi el 2% de pòlisses.

A continuació, es realitza una representació de la matriu de correlacions de Spearman entre les 43 variables numèriques, present a la Figura 9. S'ha fet servir Spearman atès que la correlació de Pearson requereix dades normals i amb l'aplicació del test Shapiro-Wilks s'ha conclós que les dades no són normals. Aquesta s'ha ordenat (fent servir un algoritme de clustering jeràrquic) per presentar les correlacions més elevades al principi i les menys elevades a continuació. Com era d'esperar, totes les correacions són positives, al tractar-se de variables definides positives. Pot apreciar-se, tot i que lleugerament, que les variables relacionades amb caràcter jurídic (Pymes, Specialties, Comerços), així com les de vida tenen una correlació inferior que la resta, de igual manera que ocorre amb clients nous o d'edats inferiors als 45 anys.

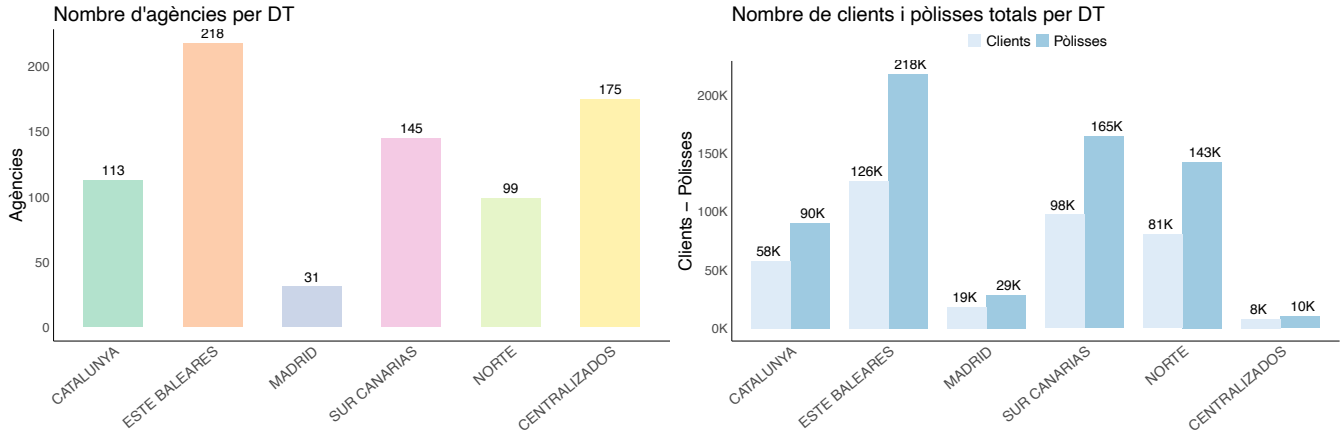


Figura 8: Total d'agències, clients i pòlisses per DT.

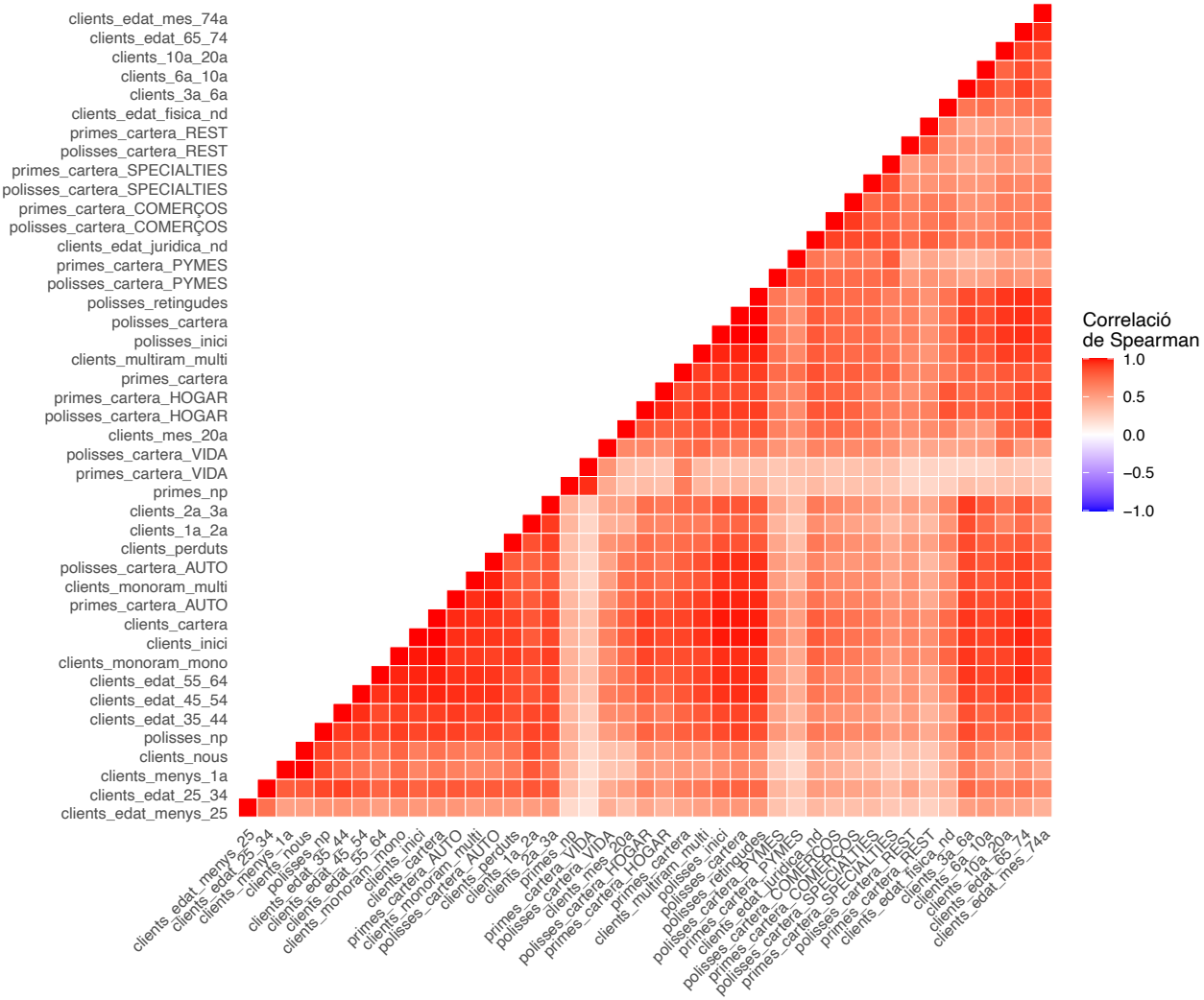
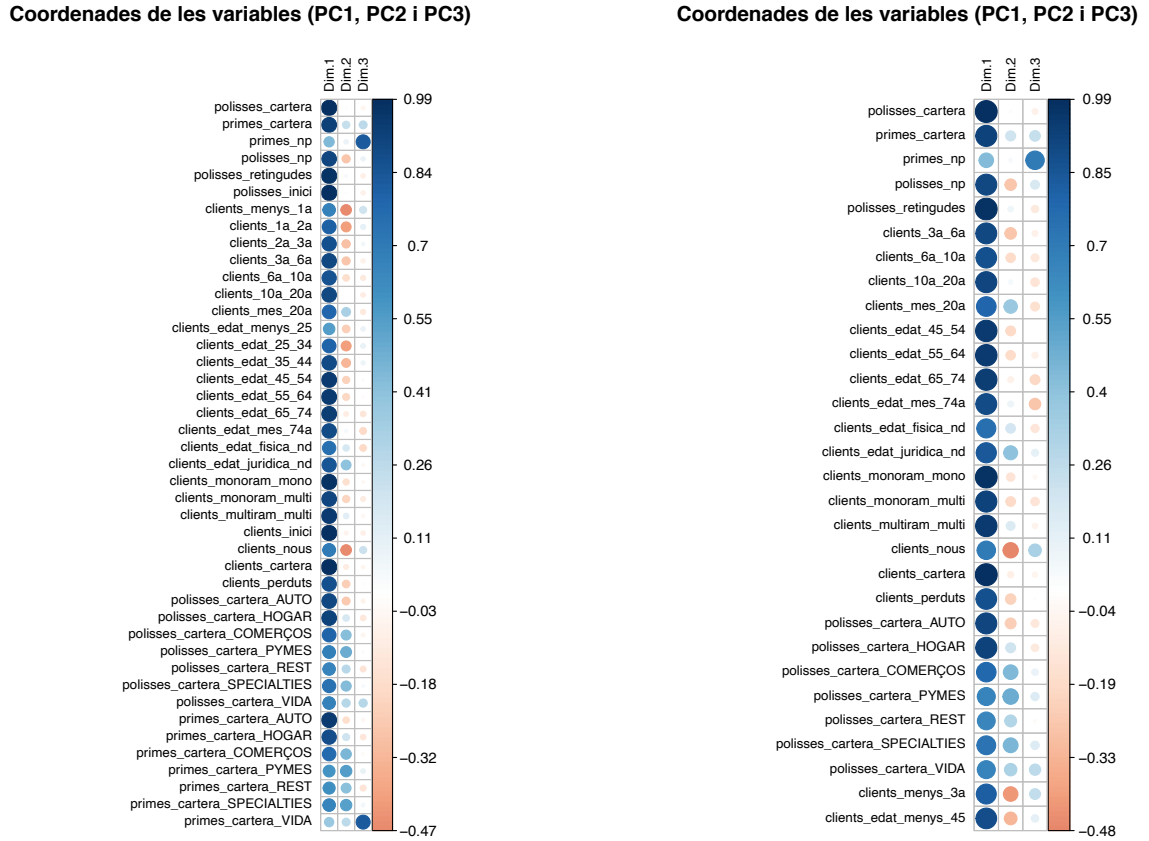


Figura 9: Representació gràfica de la matriu de correlació de Spearman entre les variables.

Per entendre millor aquestes correlacions, s'ha procedit a fer una anàlisi de components principals (PCA) amb l'objectiu triple: trobar aquelles components que expliquin la major quantitat de variança, capturar la senyal en les dades i ometre el soroll i per últim, reduir la dimensionalitat del dataset.



(a) Dataset original (43 variables)
PC1: 68.7%, PC2: 8.8%, PC3: 4.4%

(b) Dataset reduït (30 variables)
PC1: 73.4%, PC2: 7.5%, PC3: 3.9%

Figura 10: Representació gràfica de les coordenades de les variables respecte les tres primeres components principals. A l'esquerra (a), amb les 43 variables numèriques. A la dreta (b), amb les 30 variables numèriques finals.

En primer lloc, s'han escalat les variables i seguidament s'ha utilitzat l'algoritme `PCA` implementat a R al paquet `FactorMineR` [10]. A la Figura 10a trobem la representació gràfica de les 43 variables numèriques respecte les tres primeres components principals. Obtenim un percentatge de variança explicada del 68.7% només amb la primera component. Es tracta d'una component purament de volum, on totes les variables hi tenen una component estrictament positiva. La 2a component és capaç d'explicar un 8.8% de variança, on hi trobem positivament correlades aquelles components més relacionades amb productes de negocis (tant amb pòlisses com amb primes com clients de personalitat jurídica). Per altra banda, negativament correlades hi trobem components més relacionades amb clients més joves (amb antiguitats inferiors a 3 anys) o edats inferiors a 45 anys i clients i pòlisses noves. Per últim, caldria anar a una 3a component, amb una explicabilitat de la variança del 4.4% per trobar una contribució significativa amb vida.

Amb els resultats obtinguts amb el PCA, s'ha simplificat el dataset original, eliminant aquelles variables que afegeixen redundància (com és el cas de les primes de cada línia de negoci o els clients a l'inici del periode amb els clients cartera) i creant-ne de noves (agrupant els

clients amb una antiguitat inferior a 3 anys o amb edat inferior a 45 anys). D'aquesta manera, reduïm el dataset a 30 variables numèriques. I reapliquem el PCA. Els resultats els presentem a la Figura 10b. Amb aquesta reducció de dimensionalitat aconseguim augmentar la variança explicada per la PC1 (arribant al 73.4%) i assolint gairebé un 81% de variança explicada entre les dues primeres PC (vs 77.5% del dataset original). Podem apreciar que les conclusions extretes al dataset original es mantenen en el dataset reduït: PC1 és una component de volum, PC2 és una component d'especialització (en positiu, component jurídica, en negatiu, component de clients recents i clients amb menys de 45 anys).

3.2 Clustering

Partint del dataset reduït, amb 30 variables, s'ha procedit a aplicar amb el mateix conjunt de dades els tres algoritmes de clustering explicats a les seccions anteriors: $k-$means, $k-$medoids i *kamila*. Prèviament, s'ha calculat l'estadístic de Hopkins amb la funció `hopkins` del paquet d'R del mateix nom [11], on s'ha trobat que $H = 0.98$, concloent que la BBDD és clusteritzable. Seguidament, s'ha decidit el nombre de clusters. Per fer-ho, s'han aplicat els diferents mètodes comentats en la secció 2.2 i s'han representat gràficament a la Figura 11.

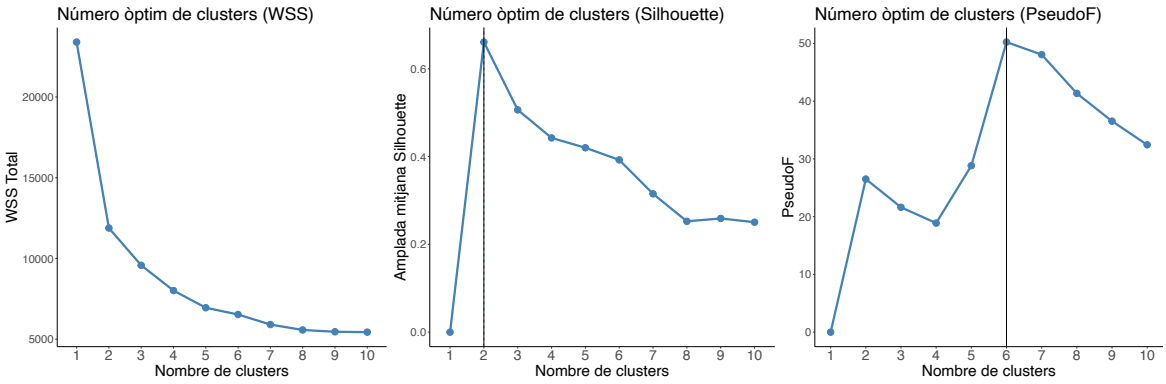


Figura 11: Representació gràfica dels diferents criteris per la selecció del número òptim de clusters.

Podem apreciar que el criteri construït a partir del gradent TESS (primera representació de la Figura 11) el canvi del *colze* es produeix al valor $k = 2$, tot i que també, però de forma més succinta, al valor $k = 5$. En el cas del criteri de Silhouette obtenim que el nombre òptim de clusters és $k = 2$, i per últim, el criteri Pseudo$F$ aposta per un valor $k = 6$. Tot i que $k = 2$ és el nombre de clusters que s'hauria d'escollir segons els resultats de la majoria de criteris, per tal de poder dur a terme campanyes de marketing més focalitzades aquest resultat es considera inviable a nivell negoci. És per això que negoci ha demanat una segmentació amb un mínim de 5 grups, raó per la qual s'ha apostat per $k = 6$ grups, tal i com ho exemplifica el criteri Pseudo$F$.

En les properes subseccions s'ha procedit a fer una representació gràfica dels clusters obtinguts per cada algoritme amb la funció `fviz_cluster` del paquet `factoextra` [12], així com una taula resumint el comportament de cada cluster segons les variables on hi ha una component de significació major comparat entre els altres clusters. Cadascun dels clusters en cada metodologia ha rebut un "bateig", on se'ls hi ha atorgat un nom simbòlic per ajudar el lector a poder tenir una noció de comparabilitat/diferenciabilitat entre els mateixos. Pel que fa referència a les taules resum, s'ha marcat amb color **blau** aquells valors que suposen una diferència respecte la resta de clusters. S'ha inclós també una columna ($p-$valor) on

s'ha computat el $p-$value del test d'hipòtesis següent:

$$\begin{aligned} H_0: &\quad \mu_1 = \mu_2 = \cdots = \mu_k \\ H_1: &\quad \mu_i \neq \mu_j,\ i \neq j,\ i,j \in \{1,\ldots,k\}, \end{aligned}$$

on s'ha fet servir el paquet `compareGroups` [13], que, amb la funció del mateix nom, primer avalua la normalitat per cadascuna de les variables amb tests estadístics Shapiro-Wilks, i en el cas que no ho siguin (com es el cas que s'escau), procedeix amb tests Kruskal-Wallis. Per altra banda, en cadascuna de les figures de la representació dels algoritmes de clustering s'ha incorporat el temps de computació, obtingut amb la funció `system.time` del paquet `base` [5] d'R.

Finalment, cal destacar, però, que els valors presentats a les taules són, en gran mesura, proporcions de les variables originals respecte al total de clients o primes. S'han construït també noves variables per facilitar el seguiment, com es el cas de l'agrupació dels productes de Negocis (com la suma de Pymes, Specialties i Comerços), la densitat de pòlissa (com el quocient entre el nombre total de pòlisses en cartera respecte al nombre total de clients en cartera) i la prima promig (com el quocient entre el nombre total de primes en cartera respecte al nombre total de de pòlisses en cartera).

3.2.1 $k-$means

La Figura 12 representa gràficament les 781 agències en els 6 grups que s'han trobat, a partir de la implementació de l'algoritme `kmeans` en R del paquet `stats` [5], amb $k = 6$. Apreciem que l'algoritme té un temps de computació de 0.077s. La Taula 4 resumeix el comportament de cada cluster segons l'algoritme `kmeans`. Podem apreciar que la majoria de variables presenten significació estadística.

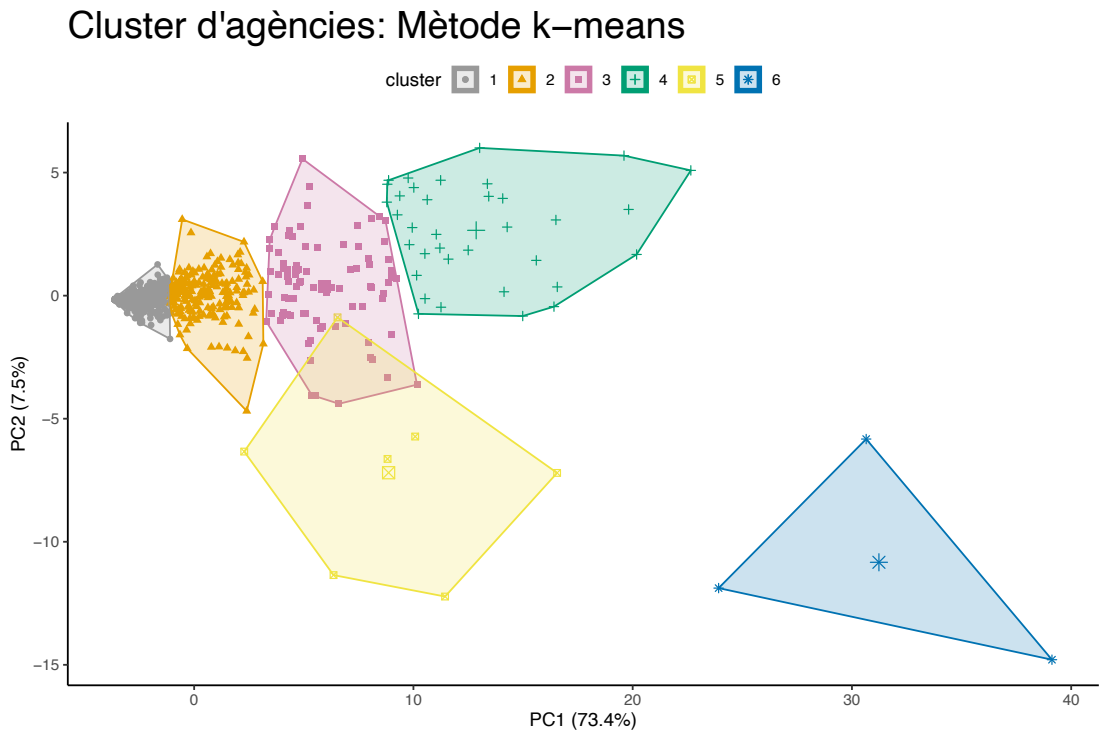


Figura 12: Representació gràfica del clustering $k-$means. Temps de computació: 0.077s.

Variable *Nom del cluster*	**Cluster 1** *Bronze*	**Cluster 2** *Plata*	**Cluster 3** *Or*	**Cluster 4** *Safir*	**Cluster 5** *Rubí*	**Cluster 6** *Diamant*	$p-$valor
n	442	208	86	35	7	3	–
$\%n$	56.6%	26.6%	11.0%	4.5%	0.9%	0.3%	–
primes_cartera	0.08M	0.35M	0.88M	1.72M	1.24M	2.30M	<0.001
%clients_edat_menys_45	**11%**	14%	16%	15%	**41%**	16%	<0.001
%clients_menys_3a	19%	20%	20%	19%	**61%**	25%	<0.001
%primes_cartera_AUTO	35%	44%	45%	36%	**63%**	**70%**	<0.001
%primes_cartera_HOGAR	24%	22%	19%	19%	18%	19%	0.057
%primes_cartera_VIDA	**18%**	12%	14%	16%	16%	2%	0.031
%primes_NEGOCIS	15%	16%	17%	**24%**	**2%**	**6%**	0.002
%clients_monoram_mono	**74%**	64%	61%	59%	**81%**	64%	<0.001
%clients_monoram_multi	11%	13%	14%	12%	12%	18%	<0.001
%clients_multiram_multi	15%	23%	25%	28%	7%	18%	<0.001
%primes_np	12%	13%	14%	14%	**48%**	15%	<0.001
%polisses_retingudes	85%	87%	86%	87%	**74%**	87%	0.019
densitat_polissa	1.40	1.66	1.76	1.81	**1.28**	1.61	<0.001
%clients_nous	10%	9%	9%	8%	**39%**	11%	<0.001
%clients_perduts	**14%**	10%	9%	8%	**15%**	9%	0.007
prima_promig	360€	380€	399€	450€	320€	**279€**	0.026

Taula 4: Resum dels clusters: clustering $k-$means.

El primer que podem veure és que la representació bidimensional és consistent amb el què hem trobat a l'anàlisi PCA: els grups formats es diferencien principalment per una component de volum (en aquest cas, hem decidit presentar-ho amb primes): els grups estan molt ben diferenciats i a continuació presentem els trets més diferenciadors respecte la resta de clusters:

- El Cluster 1 (*Cluster bronze: agències petites amb carteres senior*) conté el major nombre d'agències (442 agències, 56.6%). Es tracta d'agències amb una baixa facturació (0.08M€), on el 89% dels clients presenta edats de 45 anys o superiors. Es tracta d'agències on quasi el 75% dels seus clients només té una única pòlissa i on la nova producció només suposa el 12% del total cartera. Per últim, és el grup d'agències on els productes de vida hi tenen una major presencialitat (el 18% de la facturació) i un alt percentatge de pèrdua de clients (14%).

- Els Cluster 2 (*Cluster plata: agències mitjanes*) i 3 (*Cluster or: agències grans*) són els següents grups que contenen més agències (en conjunt, 296 agències, 37.6%). Es tracta d'agències on la principal diferència entre elles és la facturació: les agències *or* facturen fins a 2.5 vegades el que facturen les agències *plata*. Les agències *or* tenen, en general, major densitat (és a dir, pòlisses per client).

- El Cluster 4 (*Cluster safir: agències especialistes en productes jurídics*) conté un grup reduït d'agències (35 agències, 4.5%). El tret diferencial d'aquest grup és la component de facturació que presenta en productes de naturalesa jurídica (tot i no arribar a un $p-$value significatiu inferior al 0.001), on arriben al 24% del total facturació i on la densitat és major (1.81 pòlisses per client).

- El Cluster 5 (*Cluster rubí: agències joves productores amb rotació*), amb només 7 agències (0.9%) recull un comportament molt concret: es tracta d'agències amb un alt pes de clients nous i joves, amb una forta component d'auto i de nova producció en la seva facturació, però també el grup amb el major percentatge de clients amb una única pòlissa (80%). És també el grup d'agències on hi ha menor retenció de pòlisses (74%) i major pèrdua de clients (15%).

- El Cluster 6 (*Cluster diamant: agències gegants*), finalment, amb només 3 agències (0.3%) és el grup amb major facturació: 2.3M€, focalitzat principalment amb productes d'auto (70%) i amb primes baixes (279€).

La Figura 13 presenta la distribució de clusters respecte la DT i el nivell de digitalització. Podem apreciar que el gran gruix d'agències del cluster *bronze* són agències de la DT Centralizados, i que, per conseqüència, presenten o bé absència de nivell de digitalització o el nivell més baix.

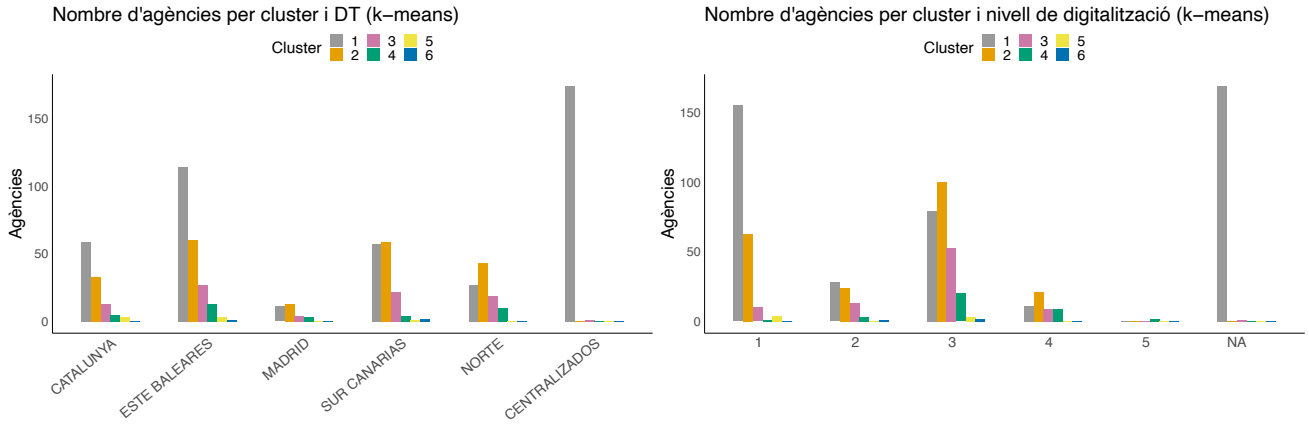


Figura 13: Representació gràfica del clustering $k-$means respecte DT i nivell de digitalització.

Atès que el Cluster *diamant* és molt diferent a la resta, podria considerar-se com un outlier tot ell en conjunt. S'ha decidit realitzar de forma excepcional el clustering $k-$means obviant les tres agències d'aquest grup i s'ha reexecutat l'algoritme fent servir 5 clusters. La Taula 5 presenta la taula de contingència amb la comparació entre el algoritme original obviant el cluster *diamant* i la reexecució del nou amb 5 clusters: només 27/778 agències (les que queden a fora de la diagonal) passarien a formar part d'un cluster diferent, sempre anant un enrere i que l'estructura general dels clusters es manté invariant.

		$k-$means (5 clusters)					
		1	**2**	**3**	**4**	**5**	
$k-$means (6)	**1**	442	0	0	0	0	*442*
	2	0	208	0	0	0	*208*
	3	0	7	79	0	0	*86*
	4	0	0	19	16	0	*35*
	5	0	0	0	1	6	*7*
		442	*215*	*98*	*17*	*6*	*778*

Taula 5: Taula de contingència de les agències: clustering $k-$means original amb 6 clusters (files) vs clustering $k-$means amb 5 clusters (columnes).

3.2.2 $k-$medoids

La Figura 14 representa gràficament les 781 agències en els 6 grups que s'han trobat, a partir de la implementació de l'algoritme $k-$medoids, amb $k = 6$, amb la funció `pam` del paquet `cluster` [6]. Comparativament amb el mètode $k-$means, l'algoritme $k-$medoids té un temps de computació superior: en concret, 0.139s, gairebé 2 cops més que l'execució de `kmeans`. La Taula 6 resumeix el comportament de cada cluster segons l'algoritme $k-$medoids. Per últim, s'inclou la Taula 7 amb la taula de contingència de l'assignació de clusters segons $k-$means (en files) i $k-$medoids (en columnes).

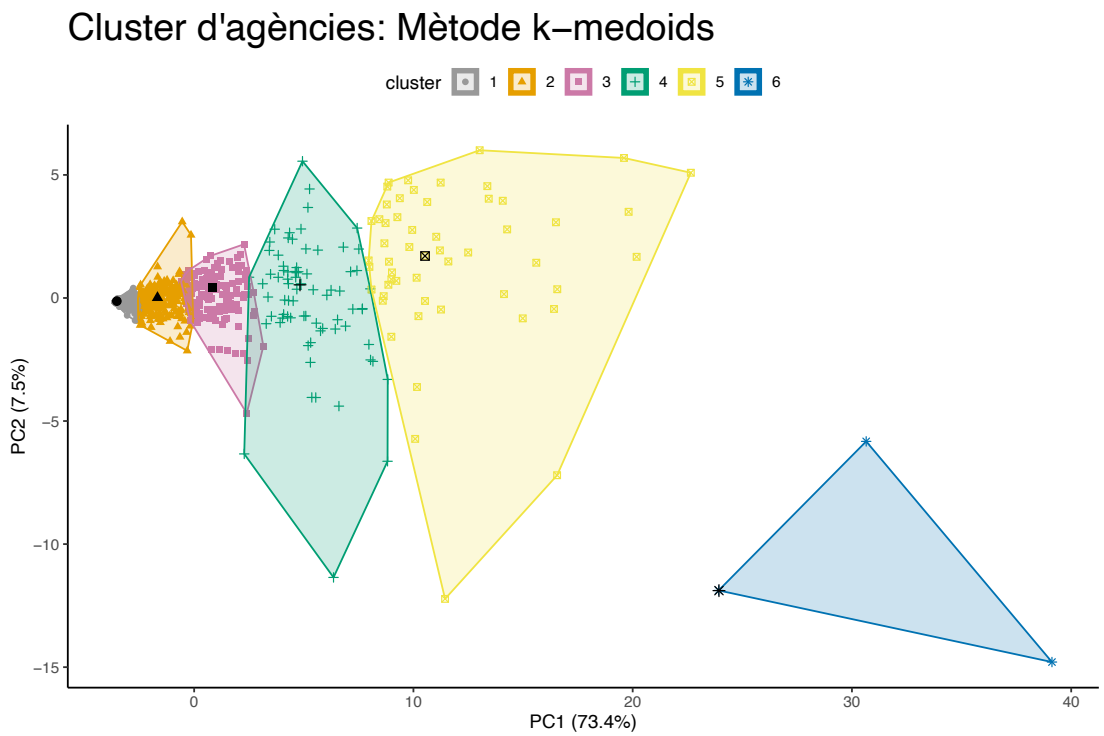


Figura 14: Representació gràfica del clustering $k-$medoids. En negre, l'element (medoide) representatiu de cada cluster. Temps de computació: 0.139s.

Variable *Nom del cluster*	**Cluster 1** *Bronze*$^-$	**Cluster 2** *Bronze*$^+$	**Cluster 3** *Plata*	**Cluster 4** *Or*	**Cluster 5** *Safir*	**Cluster 6** *Diamant*	$p-$valor
n	267	243	138	76	54	3	–
$\%n$	34.2%	31.1%	17.7%	9.7%	6.9%	0.3%	–
primes_cartera	0.03M	0.18M	0.40M	0.85M	1.50M	2.30M	<0.001
%clients_edat_menys_45	**9%**	13%	15%	18%	17%	16%	0.023
%clients_menys_3a	17%	22%	19%	23%	20%	25%	<0.001
%primes_cartera_AUTO	28%	44%	44%	45%	40%	**70%**	<0.001
%primes_cartera_HOGAR	26%	22%	22%	20%	18%	19%	0.004
%primes_cartera_VIDA	**22%**	12%	11%	14%	15%	2%	<0.001
%primes_NEGOCIS	15%	16%	16%	16%	**21%**	**6%**	0.174
%clients_monoram_mono	**79%**	66%	63%	63%	61%	64%	<0.001
%clients_monoram_multi	9%	14%	13%	13%	13%	18%	<0.001
%clients_multiram_multi	13%	20%	24%	24%	**26%**	18%	<0.001
%primes_np	11%	14%	12%	17%	15%	15%	0.037
%polisses_retingudes	84%	87%	86%	85%	87%	87%	0.069
densitat_polissa	**1.26**	1.62	1.67	1.72	1.77	1.61	<0.001
%clients_nous	10%	10%	8%	11%	8%	11%	0.650
%clients_perduts	**17%**	10%	10%	9%	9%	9%	<0.001
prima_promig	367€	385€	379€	439€	422€	**279€**	0.001

Taula 6: Resum dels clusters: clustering $k-$medoids.

- El Cluster 1 (*Cluster bronze*$^-$*: agències molt petites amb carteres molt seniors*) conté, de nou, el major nombre d'agències (267 agències, 34.2%). Es tracta d'agències encara amb una menor facturació que les trobades amb el clustering $k-$means (0.03M€) (en particular, es tracta d'un subgrup de 267 agències presents al cluster *bronze* de $k-$means). En aquest cas, s'arriba fins al 91% dels clients amb edats de 45 anys o superiors. Són agències on quasi el 80% dels seus clients només té una única pòlissa i on la nova producció només suposa el 11% del total cartera. Per últim, és el grup d'agències on els productes de vida hi tenen una major presència (el 22% de la facturació) i un alt percentatge de pèrdua de clients (17%). Comparativament amb $k-$means,

$k-$medoids es capaç de distingir agències encara més petites, més envellides i amb més pèrdua de clients.

- Els Cluster 2 (*Cluster bronze$^+$: agències petites*) i 3 (*Cluster plata: agències mitjanes*) són, tal i com ocorre amb $k-$means, els següents grups que contenen més agències (en conjunt, 381 agències, 48.8%). De nou, es tracta d'agències on la principal diferència entre elles és la facturació: les agències *plata* facturen fins a 2.2 vegades el que facturen les agències *bronze*$^+$. Les agències *plata* tenen, en general, major densitat (és a dir, pòlisses per client). Comparativament amb $k-$means, el cluster *plata* està format íntegrament per agències del cluster *plata* de $k-$means, mentre que el cluster *bronze*$^+$ suposa una mescla entre agències del clusters *bronze* i *plata* de $k-$means.

- El Cluster 4 (*Cluster or: agències grans*) conté un grup reduït d'agències (76 agències, 9.7%) i estan conformades principalment per gran part de les agències que composen el cluster *or* a $k-$means, raó per la qual les seves característiques són molt semblants. Addicionalment, de cluster *bronze*$^+$ a cluster *or* tenim l'evolució ja comentada anteriorment: es tracta d'agències que a mesura que avancen en els clusters, van reduint el pes de clients monopòlissa i van, conseqüentment, augmentant la densitat de pòlissa.

- El Cluster 5 (*Cluster safir: agències especialistes en productes jurídics*) conté 54 agències (6.9%) i recull un comportament molt semblant al cluster *safir* de $k-$means: es tracta d'agències amb un alt pes de facturació en productes de negocis (21%).

- El Cluster 6 (*Cluster diamant: agències gegants*) recull exactament les mateixes agències que el cluster *diamant* de $k-$means.

		$k-$medoids						
		1	**2**	**3**	**4**	**5**	**6**	Σ
$k-$means	**1**	267	175	0	0	0	0	*442*
	2	0	68	138	2	0	0	*208*
	3	0	0	0	70	16	0	*86*
	4	0	0	0	0	35	0	*35*
	5	0	0	0	4	3	0	*7*
	6	0	0	0	0	0	3	*3*
	Σ	*267*	*243*	*138*	*76*	*54*	*3*	*781*

Taula 7: Taula de contingència: clustering $k-$means vs clustering $k-$medoids. $RI = 0.77$.

En resum, el tret més diferenciador de $k-$medoids respecte $k-$means és el fet de la partició dels dos primers clusters de $k-$means en 3. La Figura 15 presenta la distribució de clusters respecte la DT i el nivell de digitalització. Podem apreciar que es repeteix el comportament comentat pel clustering $k-$means. S'ha afegit a la Taula 7 el valor RI, corresponent a l'índex de Rand (*Rank Index*) [14], computat amb el paquet `fossil` [15] a R, com a mesura de semblança entre dues metodologies de clustering. En síntesi, si tenim un conjunt S amb n elements $S = \{o_1, \ldots, o_n\}$ i dues particions d'S a comparar $X = \{X_1, \ldots, X_r\}$, $Y = \{Y_1, \ldots, Y_s\}$, l'índex de Rand es defineix com

$$RI = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}}, \tag{10}$$

on per $1 \leq i, j \leq n$, $i \neq j$, $1 \leq k, k_1, k_2 \leq r$, $k_1 \neq k_2$, $1 \leq \ell, \ell_1, \ell_2 \leq s$, $\ell_1 \neq k_2$:

- $a = |S^*|$, on $S^* = \{(o_i, o_j) \,|\, o_i, o_j \in X_k, o_i, o_j \in Y_\ell\}$ (i.e. nombre de parell d'elements en S que coincideixen tant a X com a Y),

- $b = |S^*|$, on $S^* = \{(o_i, o_j) \,|\, o_i \in X_{k_1}, o_j \in X_{k_2}, o_i \in Y_{\ell_1}, o_j \in Y_{\ell_2}\}$ (i.e. nombre de parell d'elements en S que difereixen tant a X com a Y),
- $c = |S^*|$, on $S^* = \{(o_i, o_j) \,|\, o_i, o_j \in X_k, o_i \in Y_{\ell_1}, o_j \in Y_{\ell_2}\}$ (i.e. nombre de parell d'elements en S que coincideixen en X, però difereixen a Y), i
- $d = |S^*|$, on $S^* = \{(o_i, o_j) \,|\, o_i \in X_{k_1}, o_j \in X_{k_2}, o_i, o_j \in Y_\ell\}$ (i.e. nombre de parell d'elements en S que difereixen a X, però coincideixen en Y).

Per tant, la mètrica RI representa la freqüència d'ocurrència de parells coincidents sobre el total de parells. Comparant el resultat de $k-$means respecte $k-$medoids obtenim un $RI = 0.77$, comportant una bona semblança entre ambdós mètodes.

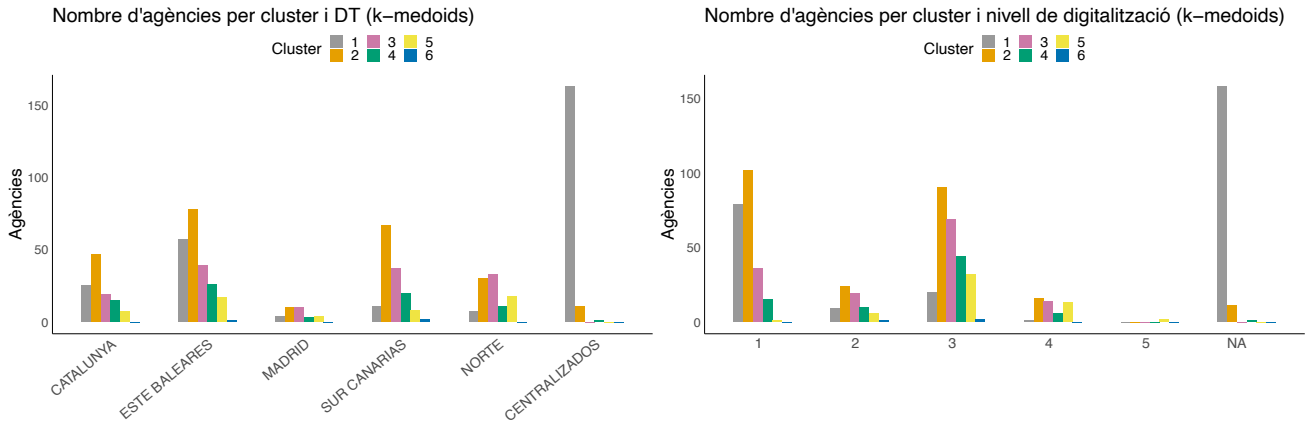


Figura 15: Representació gràfica del clustering $k-$medoids respecte DT i nivell de digitalització.

3.2.3 *kamila*

Finalment, s'ha executat l'algoritme *kamila*. Prèviament, però, s'ha requerit la imputació dels valors buits del nivell de digitalització ja presentats a la Taula 2b. Aquesta s'ha portat a terme fent servir una imputació no paramètrica amb l'algoritme Random Forest, amb la funció `missForest` [16] del paquet amb el mateix nom d'R. L'avantatge d'aquest algoritme és que es pot aplicar a datasets amb variables mixtes, com és el cas que ens pertoca. El resultat queda exemplificat a la Taula 8, on, comparant amb la Taula 2b, totes les agències que no tenien nivell de digitalització han estat associades a un nivell de digitalització 1, i així totes les agències de la DT Centralizados han passat a tenir aquest nivell de digitalització.

	1	2	3	4	5
Catalunya	52	10	45	6	0
Este Baleares	94	26	81	16	1
Madrid	11	3	14	3	0
Sur Canarias	44	17	70	13	1
Norte	27	13	47	12	0
Centralizados	175	0	0	0	0

Taula 8: Taula de contingència de les variables categòriques després d'aplicar l'algoritme `missForest`. En files, les DT. En columnes, el nivell de digitalització.

Per altra banda, en el cas de *kamila* el criteri de selecció de clusters va diferent als casos de $k-$means i $k-$medoids, i es fa servir la implementació del mètode *prediction strength* [17] de Tibshirani & Walter. No obstant, aquest mètode tendeix a afavorir un nombre més petit de clusters, com ocorre al nostre cas (veure Figura 16), on la recomanació del nombre de

clusters seria 2 per aquest mètode amb el dataset proporcionat:

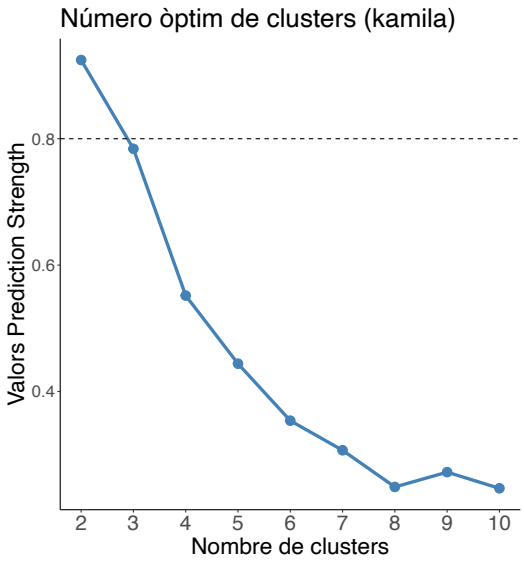


Figura 16: Representació gràfica dels valors de prediction strength per *kamila*.

La línia horitzontal de la Figura 16 a $y = 0.8$ denota el threshold per defecte, segons [17] per determinar una 'bona' separació en les dades. Qualsevol dels valors de pS per sobre d'aquest nivell seria un bon candidat per k, i en aquest cas seria $k = 2$. No obstant, per poder portar a terme comparacions entre els dos mètodes anteriors, continuarem amb $k = 6$.

La Figura 17 representa gràficament les 781 agències en els 6 grups que s'han trobat, a partir de la utilització de l'algoritme *kamila*, amb $k = 6$, amb la implementació `kamila` [9] a R. Destacar que aquest és l'algoritme amb major cost computacional: ha trigat gairebé 7 segons, 50 vegades més que la implementació `pam` pel $k-$medoids i 90 vegades més que `kmeans`. La Taula 9 resumeix el comportament de cada cluster segons l'algoritme *kamila*. Per últim, s'inclouen les taules 10 i 11 amb la taula de contingència de l'assignació de clusters segons *kamila* (en columnes) i $k-$means/$k-$medoids (en files), respectivament.

Variable *Nom del cluster*	**Cluster 1** *Bronze*$^-$	**Cluster 2** *Bronze*$^+$	**Cluster 3** *Plata*	**Cluster 4** *Or*	**Cluster 5** *Safir*	**Cluster 6** *Diamant*	$p-$valor
n	332	211	119	72	40	7	–
$\%n$	42.5%	27.0%	15.2%	9.2%	5.1%	0.9%	–
`primes_cartera`	0.04M	0.22M	0.46M	0.90M	1.57M	2.24M	<0.001
`%clients_edat_menys_45`	10%	13%	16%	18%	15%	19%	0.471
`%clients_menys_3a`	19%	21%	20%	23%	19%	22%	<0.001
`%primes_cartera_AUTO`	32%	43%	44%	49%	34%	**61%**	<0.001
`%primes_cartera_HOGAR`	25%	22%	22%	18%	19%	18%	0.014
`%primes_cartera_VIDA`	**20%**	11%	12%	14%	18%	5%	<0.001
`%primes_NEGOCIS`	15%	16%	17%	15%	**23%**	12%	0.019
`%clients_monoram_mono`	**77%**	65%	63%	63%	60%	60%	<0.001
`%clients_monoram_multi`	9%	14%	13%	14%	12%	18%	<0.001
`%clients_multiram_multi`	14%	22%	24%	23%	28%	23%	<0.001
`%primes_np`	12%	13%	13%	17%	15%	13%	0.145
`%polisses_retingudes`	84%	87%	86%	85%	87%	87%	0.036
`densitat_polissa`	**1.31**	1.66	1.68	1.71	1.78	1.83	<0.001
`%clients_nous`	10%	9%	9%	11%	8%	10%	0.749
`%clients_perduts`	**16%**	10%	10%	10%	9%	8%	<0.001
`prima_promig`	360€	385€	381€	430€	452€	**319€**	0.018

Taula 9: Resum dels clusters: clustering *kamila*.

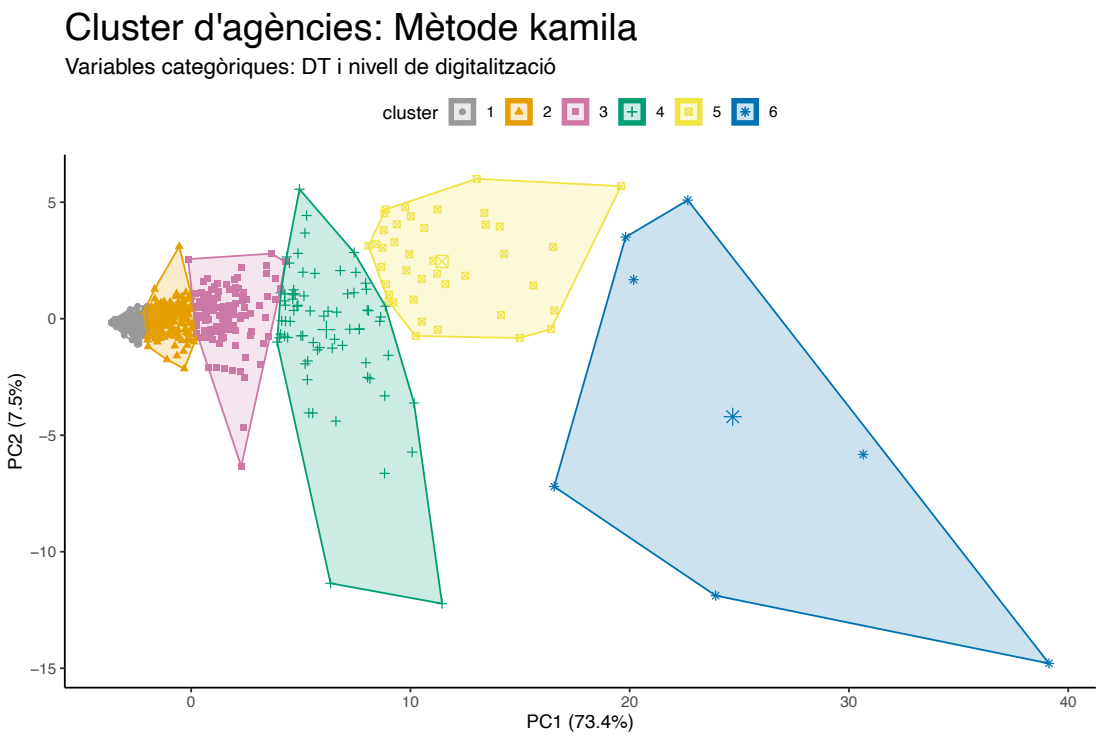


Figura 17: Representació gràfica del clustering *kamila*. Temps de computació: 6.994s.

- El Cluster 1 (*Cluster bronze*$^-$*: agències molt petites amb carteres seniors*) torna a contenir, de nou, el major nombre d'agències (332 agències, 42.5%). Es tracta, en aquest cas, d'agències amb una facturació intermitja entre el cluster *bronze* de $k-$means i el *bronze*$^-$ de $k-$medoids (0.04M€). De nou hi destaca l'elevat nombre de clients amb edats de 45 anys o superiors (90%), amb un 77% dels seus clients amb una única pòlissa i on la nova producció només suposa el 12% del total cartera. Per últim, és el grup d'agències on els productes de vida hi tenen una major presencialitat (el 20% de la facturació) i un alt percentatge de pèrdua de clients (16%).

- Els Cluster 2 (*Cluster bronze*$^+$*: agències petites*), 3 (*Cluster plata: agències mitjanes*) i 4 (*Cluster or: agències grans*) recullen 402 agències (51.5%) i, de nou, es tracta d'agències on la principal diferència entre elles és la facturació: cada grup factura en promig el doble que l'anterior. Es torna, de nou, a produir l'efecte de l'increment de densitat, i per tant, la reducció del pes de clients monopòlissa.

- El Cluster 5 (*Cluster safir: agències especialistes en productes jurídics*) conté 40 agències (5.1%) i es tracta d'un subgrup de les agències del cluster *safir* en $k-$medoids: són agències amb un alt pes de facturació en productes de negocis (23%).

- El Cluster 6 (*Cluster diamant: agències gegants*), a diferència dels algoritmes de $k-$means i $k-$medoids, ara està format per 7 agències (0.9%). Aquestes, però, tenen també una forta component de facturació (2.2M€), sobretot amb auto (61%), però és el nivell de digitalització i la DT les que fan que apareguin 4 noves agències que no estaven abans.

A grans trets, les agències més especialitzades (*safir*) presenten uns nivells de digitalització major (entre 3 i 5), amb carteres amb major facturació i pel contrari, les agències més petites presenten nivells de digitalització menor (entre 1 i 2), concentrades, la majoria d'elles, a la DT Centralizados. Aquesta conclusió pot veure's més clarament a la Figura 18.

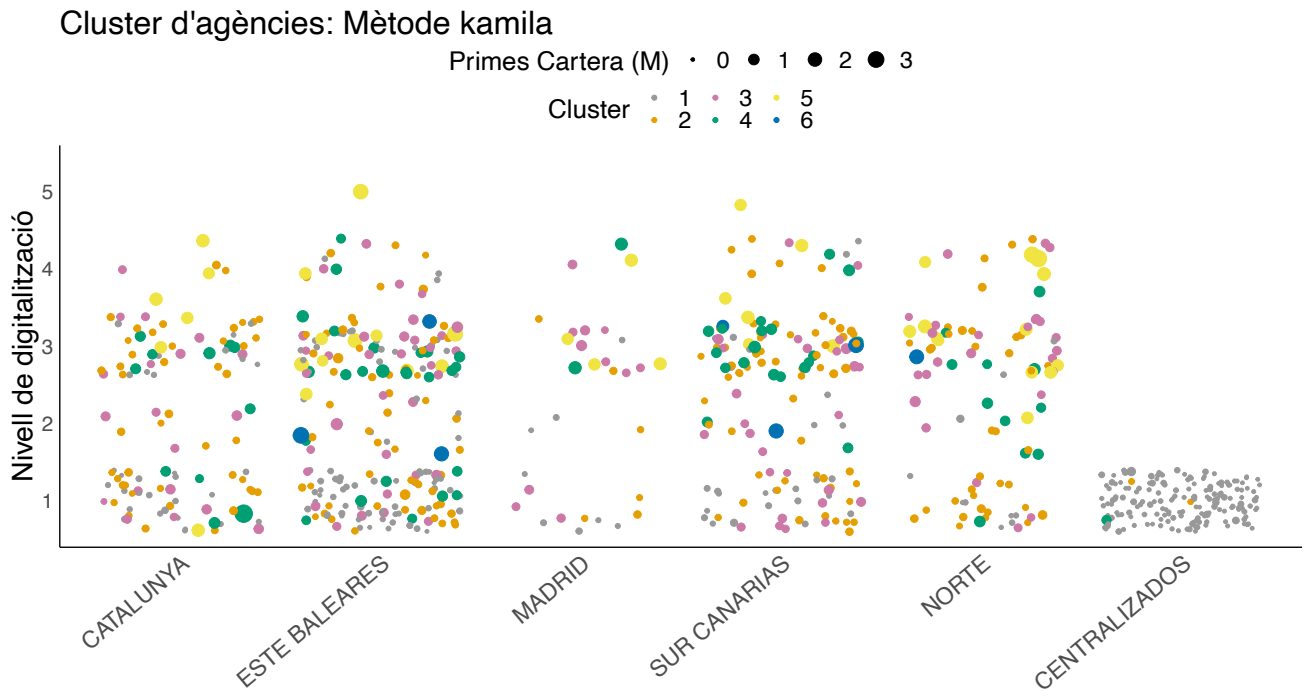


Figura 18: Representació gràfica del clustering *kamila* respecte DT i nivell de digitalització.

Per últim a les taules 10 i 11 hi trobem els índex de Rand, respectivament, entre $k-$means i *kamila* i entre $k-$medoids i *kamila*, amb $RI = 0.80$ i $RI = 0.86$ sent, aquest darrer el més elevant d'entre els tres.

		kamila						
		1	**2**	**3**	**4**	**5**	**6**	Σ
$k-$means	**1**	332	110	0	0	0	0	*442*
	2	0	101	107	0	0	0	*208*
	3	0	0	11	67	8	0	*86*
	4	0	0	0	0	32	3	*35*
	5	0	0	1	5	0	1	*7*
	6	0	0	0	0	0	3	*3*
	Σ	*332*	*211*	*119*	*72*	*40*	*7*	*781*

Taula 10: Taula de contingència: clustering $k-$means vs clustering *kamila*. $RI = 0.80$.

		kamila						
		1	**2**	**3**	**4**	**5**	**6**	Σ
$k-$medoids	**1**	267	0	0	0	0	0	*267*
	2	65	177	1	0	0	0	*243*
	3	0	34	104	0	0	0	*138*
	4	0	0	14	62	0	0	*76*
	5	0	0	0	10	40	4	*54*
	6	0	0	0	0	0	3	*3*
	Σ	*332*	*211*	*119*	*72*	*40*	*7*	*781*

Taula 11: Taula de contingència: clustering $k-$medoids vs clustering *kamila*. $RI = 0.86$.

La Figura 19 presenta la distribució de clusters respecte la DT i el nivell de digitalització. Podem apreciar que es repeteix el comportament comentat pel clustering $k-$means i $k-$medoids.

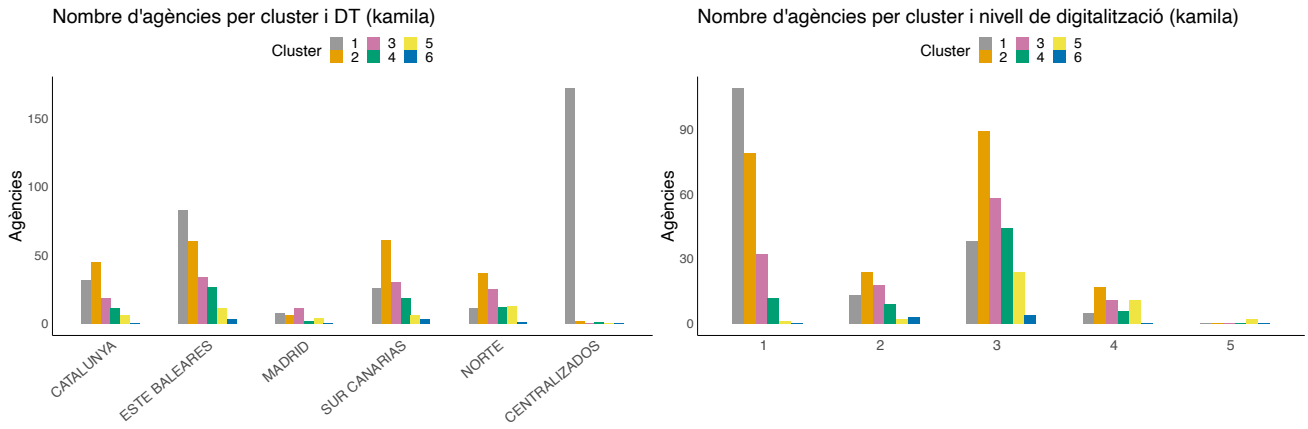


Figura 19: Representació gràfica del clustering *kamila* respecte DT i nivell de digitalització.

4 Conclusions i futurs passos

En aquest treball s'ha aprofundit i s'han avaluat diferents tècniques de clustering dintre de l'anàlisi multivariant de dades. En primer lloc, s'han exposat i explicat les dues grans metodologies de clustering, els models jeràrquics i els models no jeràrquics, on s'ha decidit apostar pels algoritmes no jeràrquics o de partició, per termes de computació i eficiència pel tipus de base de dades utilitzades. Seguidament, s'ha aprofundit en tres metodologies diferents per portar a terme clustering no jeràrquic: $k-$means, $k-$medoids i *kamila*, sent aquesta darrera una recent implementació (2019) i un nou mètode que s'ha après gràcies a aquest treball, que, respecte metodologies anteriors, permet portar a terme clustering de dades de tipologia mixta sense cap pèrdua de generalitat i sense la introducció de pesos i subjectivitat per part del usuari.

En segon lloc, s'ha realitzat una exploració completa de la base da dades proporcionada que s'ha complementat amb un anàlisi de components principals que, a més d'identificar aquelles variables més rellevants dintre del conjunt de dades, ens ha permés reduir de 43 a 30 variables numèriques el dataset facilitat.

A continuació, s'ha comprovat que el conjunt de dades proporcionades és efectivament clusteritzable amb l'estadístic de Hopkins i s'ha procedit a trobar el nombre òptim de clusters, mitjant tres metodologies diferents: criteri del colze, criteri de Silhouette i estadístic Pseudo$F$, a partir dels quals i seguint les recomanacions de negoci, s'ha continuat tot l'anàlisi posterior amb 6 clusters.

Per últim, s'han implementat les tres metodologies de clustering no jeràrquic ja esmentades i s'han exposat les similituds i diferències entre elles, destacant la consistent semblança entre els resultats de les mateixes gràcies a l'índex de Rand, i que és la metodologia que té en compte dades mixtes (*kamila*) la que genera una classificació intermitja entre els mètodes ja existents. Per altra banda, això no comporta que aquesta segmentació sigui millor o pitjor. Hem pogut apreciar que $k-$means és capaç de trobar un gran grup d'agències petites, amb població de major edat, poca captació i baixa retenció. Ha estat $k-$medoids, més robust i amb major resistència als outliers qui ha estat capaç de discernir, d'entre aquestes, aquelles encara més petites, amb facturacions inferiors, carteres més envellides, menor captació i menor retenció. Per últim, *kamila* ha permés poder integrar dues variables categòriques a l'anàlisi, sense la necessitat d'haver d'aplicar pesos i així poder portar a terme una segmentació amb un dataset amb dades mixtes. Aquest darrer algoritme ha resultat ser computacionalment més costós, suposant 90 cops més temps de computació que la implementació de `kmeans`.

Amb futures accions, es pretén integrar més informació, que als moments de produir aquest document no estaven disponibles, com és el cas de la pròpia organització de l'agència (amb respostes a una enquesta segons dedicacions a la gestió de cartera/sinistres, composició pròpiament a nivell estructural (nombre d'empleats en plantilla)) així com dades relatives a nivells sociodemogràfics segons la ubicació de l'agència, per entendre oportunitats de mercat segons s'hi trobi en àrees rurals o urbanes i amb altres agències d'altres assegurances al seu voltant.

Referències

[1] https://netgroup.edu.vn/aseguradoras-logos-09mtucge/

[2] W. K. Härdle and L. Simar. (2019). *Applied Multivariate Statistical Analysis*, 5th Edition. Springer Series.

[3] Hastie T., Tibshirani R., and Friedman J. (2009). *The Elements of Statistical Learning*. Springer Series in Statistics.

[4] Brian S. Everitt, Sabine Landau, Morven Leese, Daniel Stahl (2011). *Cluster Analysis*, 5th Edition. Wiley series.

[5] R Core Team (2023). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

[6] Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K. (2022). *cluster: Cluster Analysis Basics and Extensions*. R package version 2.1.4.

[7] Gower, J.C. (1971). *A General Coefficient of Similarity and Some of Its Properties*. Biometrics, 27, 857-871.

[8] Foss, A. H., Markatou, M., Ray, B. (2019). *Distance metrics and clustering methods for mixed-type data*. International Statistical Review, 87(1), 80-109.

[9] Foss, A. H., Markatou, M. (2018). *kamila: clustering mixed-type data in R and Hadoop. Journal of Statistical Software*, 83 (13), 1-44.

[10] Sebastien Le, Julie Josse, Francois Husson (2008). *FactoMineR: An R Package for Multivariate Analysis*. Journal of Statistical Software, 25(1), 1-18.

[11] Wright K (2023). *hopkins: Calculate Hopkins Statistic for Clustering*. R package version 1.1.

[12] Kassambara A, Mundt F (2020). *factoextra: Extract and Visualize the Results of Multivariate Data Analyses*. R package version 1.0.7.

[13] Isaac Subirana, Hector Sanz, Joan Vila (2014). *Building Bivariate Tables: The compareGroups Package for R*. Journal of Statistical Software, 57(12), 1-16. URL https://www.jstatsoft.org/v57/i12/.

[14] W. M. Rand (1971). *Objective criteria for the evaluation of clustering methods*. Journal of the American Statistical Association. American Statistical Association. 66 (336): 846–850.

[15] Vavrek, Matthew J. 2011. *fossil: palaeoecological and palaeogeographical analysis tools*. Palaeontologia Electronica, 14:1T. http://palaeo-electronica.org/2011_1/238/index.html

[16] Daniel J. Stekhoven (2022). *missForest: Nonparametric Missing Value Imputation using Random Forest*. R package version 1.5.

[17] Tibshirani R, Walther G (2005). *Cluster Validation by Prediction Strength*. Journal of Computational and Graphical Statistics, 14(3), 511–528.

A Codi d'R

```
## Paquets
library(tidyverse)
library(fossil)
library(readxl)
library(openxlsx)
library(reshape2)
library(factoextra)
library(FactoMineR)
library(clusterSim)
library(NbClust)
library(cluster)
library(StatMatch)
library(scales)
library(corrplot)
library(kamila)
library(missForest)
library(psych)
library(vtable)
library(cowplot)
library(compareGroups)
require(gridExtra)

## Input dades
df_agents <- read.csv("Agents_20221110.csv")
df_agents$dt <- factor(substring(df_agents$dt, 7,),
                       levels=c("CATALUNYA","ESTE BALEARES","MADRID",
                                "SUR CANARIAS", "NORTE", "CENTRALIZADOS"))

## Matriu de correlacions
cormat <- round(cor(df_agents[,-c(1, 45)], method = 'spearman'),2)
melted_cormat <- melt(cormat)

# Part superior de la matriu de correlacions
get_upper_tri <- function(cormat){
  cormat[lower.tri(cormat)]<- NA
  return(cormat)
}

# Reordenem amb un hclust
reorder_cormat <- function(cormat){
  dd <- as.dist((1-cormat)/2)
  hc <- hclust(dd)
  cormat <-cormat[hc$order, hc$order]
}

# Aplicació de l'ordre
cormat <- reorder_cormat(cormat)
upper_tri <- get_upper_tri(cormat)
melted_cormat <- melt(upper_tri, na.rm = TRUE)

# Chart - Correlació de Spearman
ggheatmap <- ggplot(melted_cormat, aes(Var2, Var1, fill = value))+
  geom_tile(color = "white")+
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(-1,1), space = "Lab",
                       name="Correlació\nde Spearman") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 9, hjust = 1),
        axis.text.y = element_text(vjust = 1,
                                   size = 9, hjust = 1),
        axis.title.x = element_blank(),
        axis.title.y = element_blank(),
```

```
        panel.grid.major = element_blank(),
        panel.border = element_blank(),
        panel.background = element_blank(),
        axis.ticks = element_blank())+
  coord_fixed()

## Plots EDA
# Establim temes genèrics pels plots
# General
theme_plots <- theme(plot.title = element_text(size=25),
                     axis.title.x = element_blank(),
                     axis.text.x = element_text(angle = 40, vjust = 1,
                                                size = 18, hjust = 1),
                     axis.text.y = element_text(size = 16),
                     axis.title.y = element_text(size = 22),
                     panel.grid.major = element_blank(),
                     panel.border = element_blank(),
                     panel.background = element_blank(),
                     legend.title = element_text(size = 20),
                     legend.text = element_text(size = 18),
                     axis.ticks = element_blank())

# Digitalització
theme_plots_digi <- theme(plot.title = element_text(size=25),
                          axis.title.x = element_blank(),
                          axis.text.x = element_text(vjust = 1,
                                                     size = 18, hjust = 1),
                          axis.text.y = element_text(size = 16),
                          axis.title.y = element_text(size = 22),
                          legend.title = element_text(size = 20),
                          legend.text = element_text(size = 18),
                          panel.border = element_blank(),
                          panel.background = element_blank(),
                          axis.ticks = element_blank())

# Criteris nombre de clusters
theme_criteris <- theme(plot.title = element_text(size=25),
                        axis.title.x = element_text(size=22),
                        axis.text.y = element_text(size=16),
                        axis.text.x = element_text(size=20),
                        axis.title.y = element_text(size=22))

# Clients
plot_clients <- ggplot(df_agents, aes(dt, clients_cartera, fill = dt))+
  labs( title ="Distribució de clients per DT",
        y="Clients")+
  geom_boxplot(width = .6, show.legend = FALSE)+
  theme_classic()+
  theme_plots+
  scale_fill_brewer(palette = "Pastel2")+
  scale_y_continuous(labels = c("0", "0.5K", "1K", "2K", "4K", "6K"),
                     breaks = c(0,500,1000,2000,4000,6000))

# Pòlisses
plot_polisses <- ggplot(df_agents, aes(dt, polisses_cartera, fill = dt))+
  labs( title ="Distribució de pòlisses per DT",
        y = "Pòlisses")+
  geom_boxplot(width = .6, show.legend = FALSE)+
  theme_classic()+
  theme_plots+
  scale_fill_brewer(palette = "Pastel2")+
  scale_y_continuous(labels = c("0K", "0.5K", "1K", "2.5K", "5K", "10K"),
                     breaks = c(0,500,1000,2000,5000,10000))


```

```r
# Primes
plot_primes <- ggplot(df_agents, aes(dt, primes_cartera, fill = dt))+
  labs( title ="Distribució de primes per DT",
        y = "Primes (euros)")+
  geom_boxplot(width = .6, show.legend = FALSE)+
  theme_classic()+
  theme_plots+
  scale_fill_brewer(palette = "Pastel2")+
  scale_y_continuous(labels = c("0M", "0.25M", "0.50M", "1M", "2M", "4M"),
                     breaks = c(0,250000,500000,1000000,2000000,4000000))

# Clients + pòlisses + primes
grid.arrange(plot_clients, plot_polisses, plot_primes, ncol=3)

# df aux per agències
df_agg <- df_agents %>% group_by(dt) %>%
  summarise(n_agencies = n(), n_polisses = sum(polisses_cartera),
            n_clients = sum(clients_cartera)) %>% ungroup()

df_agg_pivot <- df_agg %>% rename(Pòlisses = n_polisses,
                                  Clients = n_clients) %>%
  pivot_longer(c(n_agencies, Pòlisses, Clients),
               names_to = "var", values_to = "value") %>%
  filter(var != "n_agencies")

# Agències
plot_agencies <- ggplot(df_agg, aes(dt, n_agencies, fill = dt))+
  labs( title ="Nombre d'agències per DT",
        y = "Agències")+
  geom_col(width = .6, show.legend = FALSE)+
  theme_classic()+
  theme_plots+
  scale_fill_brewer(palette = "Pastel2")+
  geom_text(aes(label = n_agencies), vjust = -0.5, size=6)

# Clients+pòlisses
plot_clients_polisses <- ggplot(df_agg_pivot,
                                aes(dt, value, fill = var,
                                    label=scales::label_number_si()(value)))+
  labs( title ="Nombre de clients i pòlisses totals per DT",
        y = "Clients - Pòlisses")+
  geom_col(position = "dodge", width = 0.8)+
  theme_classic()+
  theme_plots+
  theme(legend.title=element_blank(),
        legend.text = element_text(size=18),
        legend.position = "top")+
  geom_text(position = position_dodge(width = .9),
            vjust = -0.5,
            size = 6)+
  scale_fill_brewer(c("#66A61E", "#009E73"))+
  scale_y_continuous(labels = c("0K", "50K", "100K", "150K", "200K", "250K"),
                     breaks = c(0,50000,100000,150000,200000,250000))

# Agències + clients+pòlisses
grid.arrange(plot_agencies, plot_clients_polisses, ncol=2)


## PCA
df_agents_norm <- df_agents
df_agents_norm[,-c(1, 45)] <- scale(df_agents_norm[,-c(1, 45)])

res_PCA <- PCA(df_agents_norm[,-c(1, 45)], graph = FALSE,
               scale.unit = FALSE, ncp=5)
# View(res_PCA$var$cor[, c(1,2,3)])
```

```r

# Eliminem les variables molt correlacionades entre elles
df_agents_new <- df_agents
df_agents_new$clients_menys_3a <- df_agents_new$clients_menys_1a+
                                  df_agents_new$clients_1a_2a+
                                  df_agents_new$clients_2a_3a
df_agents_new$clients_edat_menys_45 <- df_agents_new$clients_edat_menys_25+
                                       df_agents_new$clients_edat_25_34+
                                      df_agents_new$clients_edat_35_44
df_agents_sense_redun <- df_agents_new[, -c(27,7,38,39,40,41,
                                            42,43,44,45,8,9,10,15,16,17)]

res_PCA2 <- PCA(df_agents_sense_redun[,-c(1)], graph = FALSE,
                scale.unit = TRUE, ncp=5)

# Chart PCA Variables originals
corrplot(res_PCA$var$coord[, c(1,2,3)], is.corr = FALSE,
         title = "Coordenades de les variables (PC1, PC2 i PC3)",
         mar = c(0,0,2,0), cl.ratio = 2, tl.col = "black", tl.cex = 0.75)

# Chart PCA Variables reduïdes
corrplot(res_PCA2$var$coord[, c(1,2,3)], is.corr = FALSE,
         title = "Coordenades de les variables (PC1, PC2 i PC3)",
         mar = c(0,0,2,0), cl.ratio = 1.5, tl.col = "black", tl.cex = 0.75)


## Criteris del nombre de grups
# 1. Colze (WSS - TESS)
plot_wss <- fviz_nbclust(scale(df_agents_sense_redun[, -c(1)]),
                         kmeans, method = "wss", iter.max = 50)+
  theme_classic()+
  labs(title= "Número òptim de clusters (WSS)") +
  theme_criteris+
  geom_line(aes(group = 1), color = "#4381B6", size = 1.5) +
  geom_point(group = 1, size = 4, color = "#4381B6")+
  xlab("Nombre de clusters") +
  ylab("WSS Total")

# 2. Silhouette
plot_slihouette <- fviz_nbclust(scale(df_agents_sense_redun[, -c(1)]),
                                kmeans, method = "silhouette", iter.max = 50)
    +
  theme_classic()+
  labs(title= "Número òptim de clusters (Silhouette)") +
  theme_criteris+
  geom_line(aes(group = 1), color = "#4381B6", size = 1.5) +
  geom_point(group = 1, size = 4, color = "#4381B6")+
  geom_vline(xintercept = 2, linetype = 1,
             color = "black", size = 0.5)+
  xlab("Nombre de clusters") +
  ylab("Amplada mitjana Silhouette")

# 3. PseudoF
D.mah<-as.dist(mahalanobis.dist(df_agents_sense_redun[, -c(1)]))
dades <- df_agents_sense_redun[, -c(1)]
PseudoF<-data.frame()
for(centers in 2:10){
  km.mah <- pam(D.mah, centers, diss=T)  # Partitioning Around Medoids (PAM)
  PseudoF.mah <- index.G1(x=dades, cl=km.mah$cluster,
                          d=D.mah, centrotypes = "medoids")
  #càlcul de la metrica PseudoF
  PseudoF[centers-1, 1]<-centers
  PseudoF[centers-1, 2]<-PseudoF.mah
}
PseudoF # maxim local en  5->6 50.73213
```

```
plot_PseudoF <- ggplot(data = rbind(PseudoF,
                                    c(1,0)), aes(V1, V2))+
  geom_line(color = "#4381B6",  size = 1.5)+
  geom_point(color = "#4381B6", size = 4)+
  theme_classic()+
  theme_criteris+
  geom_vline(xintercept = 6, linetype = 1,
             color = "black", size = 0.5)+
  labs(title = "Número òptim de clusters (PseudoF)")+
  xlab("Nombre de clusters")+
  ylab("PseudoF")+
  scale_x_continuous(breaks=c(1,2,3,4,5,6,7,8,9,10))

## Chart - 3 criteris
grid.arrange(plot_wss, plot_slihouette, plot_PseudoF, ncol=3)


### Aplicació dels mètodes de clustering
## K-MEANS
set.seed(456)
ks <-kmeans(scale(df_agents_sense_redun[,-c(1)]), 6, nstart = 25)
# Chart: K-Means
fviz_cluster(ks, data = scale(df_agents_sense_redun[,-c(1)]),
             geom = "point",
             pointsize = 1.5) +
  theme_classic()+
  labs(title ="Cluster d'agències: Mètode k-means")+
  xlab("PC1 (73.4%)") +
  ylab("PC2 (7.5%)") +
  theme(plot.title = element_text(size=22))+
  theme(legend.position = "top")+
  guides(colour = guide_legend(override.aes = list(size=1.5),
                               nrow = 1))+
  scale_colour_manual(values = c("#999999", "#E69F00", "#CC79A7",
                                 "#009E73", "#F0E442", "#0072B2")) +
  scale_fill_manual(values = c("#999999", "#E69F00", "#CC79A7",
                               "#009E73", "#F0E442", "#0072B2"))


## K-MEDOIDS
kmed <- pam(scale(df_agents_sense_redun[,-c(1)]), 6)
# Guardem els punts dels medoids
medoids_coord <- as.data.frame(res_PCA2$ind$coord[kmed$id.med, c(1,2)])
# Chart: K-Medoids
fviz_cluster(kmed, data = scale(df_agents_sense_redun[,-c(1)]),
             geom = "point",
             pointsize = 1.5) +
  theme_classic()+
  labs(title ="Cluster d'agències: Mètode k-medoids")+
  xlab("PC1 (73.4%)") +
  ylab("PC2 (7.5%)") +
  theme(plot.title = element_text(size=22))+
  theme(legend.position = "top")+
  guides(colour = guide_legend(override.aes = list(size=1.5),
                               nrow = 1))+
  scale_colour_manual(values = c("#999999", "#E69F00", "#CC79A7",
                                 "#009E73", "#F0E442", "#0072B2")) +
  scale_fill_manual(values = c("#999999", "#E69F00", "#CC79A7",
                               "#009E73", "#F0E442", "#0072B2"))+
  geom_point(data = medoids_coord[2,],
             aes(x=Dim.1, y=Dim.2), shape = 19, size = 2)+
  geom_point(data = medoids_coord[1,],
             aes(x=Dim.1, y=Dim.2), shape = 17, size = 2)+
  geom_point(data = medoids_coord[4,],
             aes(x=Dim.1, y=Dim.2), shape = 15, size = 2)+
```

```
  geom_point(data = medoids_coord[3,],
             aes(x=Dim.1, y=Dim.2), shape = 3, size = 2)+
  geom_point(data = medoids_coord[5,],
             aes(x=Dim.1, y=Dim.2), shape = 7, size = 2)+
  geom_point(data = medoids_coord[6,],
             aes(x=Dim.1, y=Dim.2), shape = 8, size = 2)


## KAMILA
# Afegim la digitalització
digi <- read.xlsx("20221129 Resumen Nivel Digitalizacion.xlsx",
                  sheet = "Resumen",startRow = 3)
digi <- digi[, c(2,5)]
df_agents_new <- merge(df_agents_new, digi,
                       by.x = "cif", by.y = "CIF", all.x = T)
df_agents_new$Nivel.Digital <- as.factor(df_agents_new$Nivel.Digital)
# Imputem els NA
df_agents_imp <- missForest(df_agents_new[,-c(1)], verbose = TRUE)
df_agents_imp <- df_agents_imp$ximp
df_agents_imp$cif <- df_agents_new$cif

# Separem en dos dataframes numèriques i categòriques
conVars <- as.data.frame(scale(df_agents_sense_redun[, -c(1)]))
catVars_digi <- df_agents_imp[,c(44, 47)]

# Calculem Precision Strength
kmRes_digi_2 <- kamila(conVars, catVars_digi, numClust = 2:10, numInit = 100,
                       calcNumClust = "ps")
ps_kamila <- data.frame(ks = 2:10, ps = kmRes_digi_2$nClust$psValues)
# Chart - Ps Kamila
plot_precision_strength <- ggplot(data = ps_kamila, aes(ks, ps))+
  geom_line(color = "#4381B6",  size = 1.5)+
  geom_point(color = "#4381B6", size = 4)+
  theme_classic()+
  theme_criteris+
  geom_hline(yintercept = 0.8, linetype = 2,
             color = "black", size = 0.5)+
  labs(title = "Número òptim de clusters (kamila)")+
  xlab("Nombre de clusters")+
  ylab("Valors Prediction Strength")+
  scale_x_continuous(breaks=c(1,2,3,4,5,6,7,8,9,10))

# Apliquem kamila
kmRes_digi <- kamila(conVars, catVars_digi, numClust = 6, numInit = 1000)
# Chart: kamila
fviz_cluster(object = list(data = df_agents_sense_redun[, -c(1)],
                           cluster = kmRes_digi$finalMemb),
             geom = "point",
             pointsize = 1.5) +
  theme_classic()+
  labs( title ="Cluster d'agències: Mètode kamila",
        subtitle = "Variables categòriques: DT i nivell de digitalització")+
  xlab("PC1 (73.4%)") +
  ylab("PC2 (7.5%)") +
  theme(plot.title = element_text(size=22),
        plot.subtitle = element_text(size=13))+
  theme(legend.position = "top")+
  guides(colour = guide_legend(override.aes = list(size=1.5),
                               nrow = 1))+
  scale_colour_manual(values = c("#999999", "#E69F00", "#CC79A7",
                                 "#009E73", "#F0E442", "#0072B2")) +
  scale_fill_manual(values = c("#999999", "#E69F00", "#CC79A7",
                               "#009E73", "#F0E442", "#0072B2"))

# Taules de contingència entre mètodes
```

```r
df_agents_new$cluster_kmeans <- ks$cluster
df_agents_new$cluster_kmedoids <- kmed$clustering
df_agents_new$cluster_kamila <- kmRes_digi$finalMemb
# K-Means vs K-Medoids
table(df_agents_new$cluster_kmeans, df_agents_new$cluster_kmedoids)
# K-Means vs kamila
table(df_agents_new$cluster_kmeans, df_agents_new$cluster_kamilab)
# K-Medoids vs kamila
table(df_agents_new$cluster_kmedoids, df_agents_new$cluster_kamila)

## Anàlisi x clusters
# Creem les variables de ratios
df_agents2 <- df_agents_new
df_agents2[,8:14] <- df_agents2[,8:14]/df_agents2$clients_cartera
df_agents2[,8:14][is.na(df_agents2[,8:14])] <- 0
df_agents2[,46:47] <- df_agents2[,46:47]/df_agents2$clients_cartera
df_agents2[,46:47][is.na(df_agents2[,46:47])] <- 0
df_agents2[,15:21] <- df_agents2[,15:21]/rowSums(df_agents2[,15:21])
df_agents2[,15:21][is.na(df_agents2[,15:21])] <- 0
df_agents2[,23] <- df_agents2[,23]/df_agents2$clients_cartera
df_agents2[,23][is.na(df_agents2[,23])] <- 0
df_agents2[,24:26] <- df_agents2[,24:26]/df_agents2$clients_cartera
df_agents2[,24:26][is.na(df_agents2[,24:26])] <- 0
df_agents2[,28] <- df_agents2[,28]/df_agents2$clients_cartera
df_agents2[,28][is.na(df_agents2[,28])] <- 0
df_agents2[,31:37]<- df_agents2[,31:37]/rowSums(df_agents2[,31:37])
df_agents2[,31:37][is.na(df_agents2[,31:37])] <- 0
df_agents2[,38:44]<- df_agents2[,38:44]/rowSums(df_agents2[,38:44])
df_agents2[,38:44][is.na(df_agents2[,38:44])] <- 0
# Fem una dimensió de NEGOCIS
df_agents2$polisses_NEGOCIS <- rowSums(df_agents2[,c(33,34,36)])/
                               rowSums(df_agents2[,31:37])
df_agents2$primes_NEGOCIS <- rowSums(df_agents2[,c(40,41,43)])/
                             rowSums(df_agents2[,38:44])
df_agents2$polisses_NEGOCIS[is.na(df_agents2$polisses_NEGOCIS)] <- 0
df_agents2$primes_NEGOCIS[is.na(df_agents2$primes_NEGOCIS)] <- 0
# NP
df_agents2$perc_pol_np <- df_agents2$polisses_np/
                          df_agents2$polisses_cartera
df_agents2$perc_pol_np[is.na(df_agents2$perc_pol_np)] <- 0
df_agents2$perc_prim_np <- df_agents2$primes_np/
                           df_agents2$primes_cartera
df_agents2$perc_prim_np[is.na(df_agents2$perc_prim_np)] <- 0
# Reten
df_agents2$rati_retencio_pol <- df_agents2$polisses_retingudes/
                                df_agents2$polisses_inici
df_agents2$rati_retencio_pol[is.na(df_agents2$rati_retencio_pol)] <- 0
# Densitat
df_agents2$dens <- df_agents2$polisses_cartera/
                   df_agents2$clients_cartera
df_agents2$clients_nous_perc <- df_agents2$clients_nous
df_agents2$clients_perduts_perc <- df_agents2$clients_perduts/
                                   df_agents2$clients_cartera
df_agents2$prima_promig <- df_agents2$primes_cartera/
                           df_agents2$polisses_cartera


# K-Means
clusters_resum_kmeans <- df_agents2[,-c(1, 45, 51)] %>%
  mutate(CLuster = cluster_kmeans) %>%
  group_by(CLuster) %>%
  summarise_all("mean")
resum_kmeans <- t(round(clusters_resum_kmeans[, c(1,3,46,45, 38,39,44,
                                                  52,24,25,26,54,
                                                  55,56, 57, 58, 59)],2))
```

```r
resum_kmeans[-c(1,2, 14, 17),] <- percent(resum_kmeans[-c(1,2,14, 17),])
#write.csv(resum_kmeans,file='resum_kmeans.csv', row.names=TRUE)

# K-Medoids
clusters_resum_kmedoids <- df_agents2[kmed$id.med,-c(1, 45, 51)] %>%
  mutate(CLuster = cluster_kmedoids) %>%
  group_by(CLuster) %>%
  summarise_all("mean")
resum_kmedoids <- t(round(clusters_resum_kmedoids[, c(1,3,46,45, 38,39,44,
                                                      52,24,25,26,54,
                                                      55,56, 57, 58, 59)],2))
resum_kmedoids[-c(1,2, 14, 17),] <- percent(resum_kmedoids[-c(1,2,14,17),])
#write.csv(resum_kmedoids,file='resum_kmedoids.csv', row.names=TRUE)

# kamila
clusters_resum_kamila <- df_agents2[,-c(1, 45, 51)] %>%
  mutate(CLuster = cluster_kamila) %>%
  group_by(CLuster) %>%
  summarise_all("mean")
resum_kamila <- t(round(clusters_resum_kamila[, c(1,3,46,45, 38,39,44,
                                                  52,24,25,26,54,
                                                  55,56, 57, 58, 59)],2))
resum_kamila[-c(1,2, 14, 17),] <- percent(resum_kamila[-c(1,2,14, 17),])
resum_kamila
#write.csv(resum_kamila,file='resum_kamila.csv', row.names=TRUE)

## Taules EDA
df_summary_num <- st(df_agents, add.median = TRUE)
## Digitalització i dt
table(df_agents_new$dt, df_agents_new$Nivel.Digital, useNA = "ifany")

# Charts per tipus de cluster, DT i Digitalització
df_agents_new$digi <- as.numeric(df_agents_new$Nivel.Digital)
df_agents_new[is.na(df_agents_new$digi),"digi"] <- "NA"
df_agents_new$digi <- factor(df_agents_new$digi)
# Df auxiliars: Cluster i DT
df_aux <- tidyr::crossing(unique(df_agents_new$cluster_kmeans),
                          unique(df_agents_new$dt))
names(df_aux) <- c("cluster_kmeans", "dt")
df_aux$cluster_kmedoids <- df_aux$cluster_kmeans
df_aux$cluster_kamila <- df_aux$cluster_kmeans
# Df auxiliars: Cluster i Digitalització
df_aux_digi <- tidyr::crossing(unique(df_agents_new$cluster_kmeans),
                               unique(df_agents_new$digi))
names(df_aux_digi) <- c("cluster_kmeans", "digi")
df_aux_digi$cluster_kmedoids <- df_aux_digi$cluster_kmeans
df_aux_digi$cluster_kamila <- df_aux_digi$cluster_kmeans

# K-Means
kmeans_dt <- df_agents_new %>% group_by(cluster_kmeans, dt) %>%
  summarise(n_agencies = n()) %>% merge(df_aux, all.y = T) %>%
  replace_na(list("n_agencies" = 0)) %>%
  ggplot(aes(dt, n_agencies, fill = factor(cluster_kmeans)))+
  labs( title ="Nombre d'agències per cluster i DT (k-means)",
        y = "Agències",
        fill = "Cluster")+
  geom_col(width = .6,position = "dodge")+
  theme_classic()+
  theme(legend.position = "top")+
  theme_plots+
  scale_colour_manual(values = c("#999999", "#E69F00", "#CC79A7",
                                 "#009E73", "#F0E442", "#0072B2")) +
  scale_fill_manual(values = c("#999999", "#E69F00", "#CC79A7",
                               "#009E73", "#F0E442", "#0072B2"))

```

```r
kmeans_digi <- df_agents_new %>% group_by(cluster_kmeans, digi) %>%
  summarise(n_agencies = n()) %>% merge(df_aux_digi, all.y = T) %>%
  replace_na(list("n_agencies" = 0)) %>%
  ggplot(aes(digi, n_agencies, fill = factor(cluster_kmeans)))+
  labs(title ="Nombre d'agències per cluster i
       nivell de digitalització (k-means)",
        y = "Agències",
        fill = "Cluster")+
  geom_col(width = .6,position = "dodge")+
  theme_classic()+
  theme(legend.position = "top")+
  theme_plots_digi+
  scale_colour_manual(values = c("#999999", "#E69F00", "#CC79A7",
                                 "#009E73", "#F0E442", "#0072B2")) +
  scale_fill_manual(values = c("#999999", "#E69F00", "#CC79A7",
                               "#009E73", "#F0E442", "#0072B2"))

# K-Medoids
kmedoids_dt <- df_agents_new %>% group_by(cluster_kmedoids, dt) %>%
  summarise(n_agencies = n()) %>% merge(df_aux, all.y = T) %>%
  replace_na(list("n_agencies" = 0)) %>%
  ggplot(aes(dt, n_agencies, fill = factor(cluster_kmedoids)))+
  labs( title ="Nombre d'agències per cluster i DT (k-medoids)",
        y = "Agències",
        fill = "Cluster")+
  geom_col(width = .6,position = "dodge")+
  theme_classic()+
  theme(legend.position = "top")+
  theme_plots+
  scale_colour_manual(values = c("#999999", "#E69F00", "#CC79A7",
                                 "#009E73", "#F0E442", "#0072B2")) +
  scale_fill_manual(values = c("#999999", "#E69F00", "#CC79A7",
                               "#009E73", "#F0E442", "#0072B2"))

kmedoids_digi <- df_agents_new %>% group_by(cluster_kmedoids, digi) %>%
  summarise(n_agencies = n()) %>% merge(df_aux_digi, all.y = T) %>%
  replace_na(list("n_agencies" = 0)) %>%
  ggplot(aes(digi, n_agencies, fill = factor(cluster_kmedoids)))+
  labs( title ="Nombre d'agències per cluster i
        nivell de digitalització (k-medoids)",
        y = "Agències",
        fill = "Cluster")+
  geom_col(width = .6,position = "dodge")+
  theme_classic()+
  theme(legend.position = "top")+
  theme_plots_digi+
  scale_colour_manual(values = c("#999999", "#E69F00", "#CC79A7",
                                 "#009E73", "#F0E442", "#0072B2")) +
  scale_fill_manual(values = c("#999999", "#E69F00", "#CC79A7",
                               "#009E73", "#F0E442", "#0072B2"))

# Kamila
kamila_dt <- df_agents_new %>% group_by(cluster_kamila, dt) %>%
  summarise(n_agencies = n()) %>% merge(df_aux, all.y = T) %>%
  replace_na(list("n_agencies" = 0)) %>%
  ggplot(aes(dt, n_agencies, fill = factor(cluster_kamila)))+
  labs( title ="Nombre d'agències per cluster i DT (kamila)",
        y = "Agències",
        fill = "Cluster")+
  geom_col(width = .6,position = "dodge")+
  theme_classic()+
  theme(legend.position = "top")+
  theme_plots+
  scale_colour_manual(values = c("#999999", "#E69F00", "#CC79A7",
                                 "#009E73", "#F0E442", "#0072B2")) +
```

```
  scale_fill_manual(values = c("#999999", "#E69F00", "#CC79A7",
                               "#009E73", "#F0E442", "#0072B2"))

kamila_digi <- df_agents_new %>% group_by(cluster_kamila, digi) %>%
  summarise(n_agencies = n()) %>% merge(df_aux_digi, all.y = T) %>%
  replace_na(list("n_agencies" = 0)) %>%
  ggplot(aes(digi, n_agencies, fill = factor(cluster_kamila)))+
  labs( title ="Nombre d'agències per cluster i
        nivell de digitalització (kamila)",
        y = "Agències",
        fill = "Cluster")+
  geom_col(width = .6,position = "dodge")+
  theme_classic()+
  theme(legend.position = "top")+
  theme_plots_digi+
  scale_colour_manual(values = c("#999999", "#E69F00", "#CC79A7",
                                 "#009E73", "#F0E442", "#0072B2")) +
  scale_fill_manual(values = c("#999999", "#E69F00", "#CC79A7",
                               "#009E73", "#F0E442", "#0072B2"))

# Charts 2 a 2
plot_grid(kmeans_dt, kmeans_digi, align = "h", nrow = 1)
plot_grid(kmedoids_dt, kmedoids_digi, align = "h", nrow = 1)
plot_grid(kamila_dt, kamila_digi, align = "h", nrow = 1)


## Chart complet de kamila
df_agents_new %>% ggplot(aes(y = digi_kamila, x = dt,
                             color = factor(cluster_kamila),
                             size = primes_cartera/1000000))+
  theme_classic()+
  geom_point(position = "jitter")+
  labs( title ="Cluster d'agències: Mètode kamila",
        color = "Cluster",
        size = "Primes Cartera (M)")+
  theme(plot.title = element_text(size=22),
        plot.subtitle = element_text(size=13))+
  ylab("Nivell de digitalització")+
  theme(legend.position = "top", legend.box="vertical",
        legend.margin=margin())+
  theme_plots+
  scale_colour_manual(values = c("#999999", "#E69F00", "#CC79A7",
                                 "#009E73", "#F0E442", "#0072B2")) +
  scale_fill_manual(values = c("#999999", "#E69F00", "#CC79A7",
                               "#009E73", "#F0E442", "#0072B2"))

## Rand Index
rand.index(df_agents_new$cluster_kmeans, df_agents_new$cluster_kmedoids)
rand.index(df_agents_new$cluster_kmeans, df_agents_new$cluster_kamila)
rand.index(df_agents_new$cluster_kamila, df_agents_new$cluster_kmedoids)

## Temps de computació
system.time({kmeans(scale(df_agents_sense_redun[,-c(1)]), 6,
                    nstart = 25)) # 0.077s
system.time({pam(scale(df_agents_sense_redun[,-c(1)]), 6)}) # 0.139s
system.time({kamila(conVars, catVars_digi, numClust = 6,
                    numInit = 1000)}) #6.994s

# Compare groups
clusters_resum <- df_agents2[,-c(1, 45, 51)]
resum_global <- df_agents2[, c(47,48,49,2,44,45,37,38,43,53,
                               23,24,25,55,56,57,58,59,60)]
resu_kmeans<-compareGroups(cluster_kmeans ~ .,
                           data=resum_global[resum_global$cluster_kmeans,
                                             max.ylev = 6, -c(2,3)])
```

```
resu_kmedoids<-compareGroups(cluster_kmedoids ~ .,
                             data=resum_global[resum_global$cluster_kmedoids,
                                               max.ylev = 6, -c(1,3)])
resu_kamila<-compareGroups(cluster_kamila2 ~ .,
                           data=resum_global[resum_global$cluster_kamila,
                                             max.ylev = 6, -c(1,2)])
#createTable(resu_kmeans)
#createTable(resu_kmedoids)
#createTable(resu_kamila)
```