Τύποι, Ποιότητα και Προεπεξεργασία Δεδομένων

Δρ. Σωτήριος Δ. Νικολόπουλος

Big Data & Analytics

Πανεπιστήμιο Πελοποννήσου

Τμήμα Λογιστικής & Χρηματοοικονομικών

s.nikolopoulos@go.uop.gr

Ενότητες

Ενότητες Παρουσίασης

Σε αυτό το μάθημα θα γίνει κατανοητό ότι τα δεδομένα, οι τύποι και η ποιότητά τους συνιστούν ένα αναπόσπαστο κομμάτι στη διαδικασία της εξόρυξης δεδομένων.

Η ποιότητα των δεδομένων καθορίζει σε μεγάλο βαθμό και την ποιότητα των αποτελεσμάτων της εξόρυξης δεδομένων αλλά και κάθε διαδικασίας ανάπτυξης μοντέλων.

Οι παράμετροι εκείνοι των δεδομένων που επηρεάζουν την ποιότητά τους πρέπει να είναι σαφείς, έτσι ώστε να είναι σε θέση κάποιος να τις αξιολογήσει και να τις βελτιώσει.

Η προεπεξεργασία των δεδομένων αποτελεί το πιο επίπονο και χρονοβόρο κομμάτι στη διαδικασία της ανακάλυψης γνώσης από τα δεδομένα

Στόχος, του τρέχοντος του μαθήματος είναι να εξοικειωθείτε με όλες τις διαφορετικές μορφές προεπεξεργασίας των δεδομένων και να είστε σε θέση να τις εφαρμόσετε.

Παράλληλα, να είστε σε θέση να εφαρμόζετε τις τεχνικές αυτές μέσω ενός εργαλείου, όπως η γλώσσα προγραμματισμού R.

Η προεπεξεργασία των δεδομένων αποτελεί ένα από τα πιο σημαντικά βήματα της **Ανακάλυψης Γνώσης σε Βάσεις Δεδομένων (ΑΓΒΔ)**, το οποίο μπορεί να απαιτήσει έως και το 60% - 80% της συνολικής προσπάθειας.

Αυτό συμβαίνει διότι, αν τα ίδια τα δεδομένα δεν είναι «καθαρά» και στην κατάλληλη μορφή, δεν έχει νόημα να μιλάμε για ποιότητα αποτελεσμάτων.

Οι δυο βασικές κατηγορίες μεταβλητών είναι οι

- ποιοτικές και
- ποσοτικές

Οι ποιοτικές μεταβλητές αναφέρονται σε μεταβλητές, όπως για παράδειγμα

- το φύλο
- το επίπεδο μόρφωσης
- η περιοχή καταγωγής κ.ο.κ.

Περαιτέρω διαχωρίζονται σε

- ονομαστικές (nominal) και
- σε διατακτικές ή τακτικές

Οι **ονομαστικές (nominal) μεταβλητές** αναπαριστούν κατηγορίες, που η σειρά τους δεν έχει σημασία, π.χ. χρώμα, μέσο μεταφοράς.

Αντίθετα, **οι διατακτικές ή τακτικές μεταβλητές** αναπαριστούν κατηγορίες, των οποίων η διάταξη έχει σημασία, π.χ. σοβαρότητα ασθένειας, γνώμη.

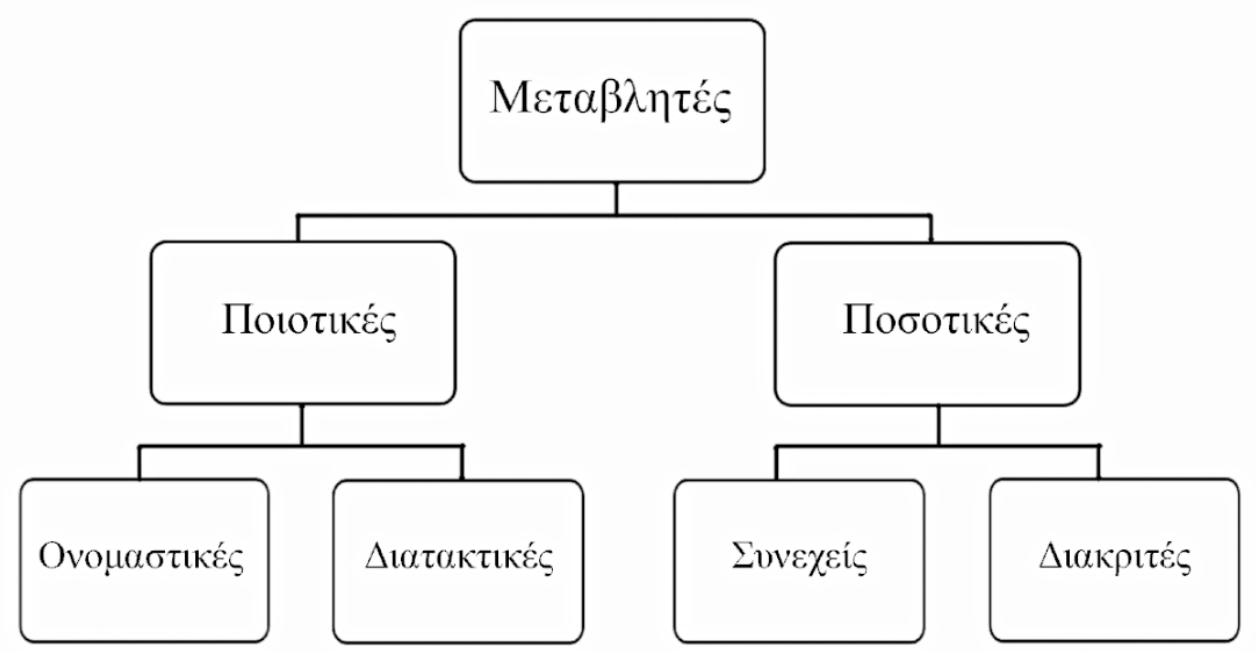
Οι ποσοτικές μεταβλητές είναι αριθμητικές τιμές, οι οποίες εκφράζονται σε μια μονάδα μέτρησης, π.χ. ηλικία.

Διαχωρίζονται

- σε ασυνεχείς ή διακριτές και
- σε συνεχείς.

Τα δεδομένα ανάλογα με την κλίμακα μέτρησης τους χαρακτηρίζονται ως

- κατηγορικά, δηλαδή σε επίπεδα ή κατηγορίες, και
- σε μετρήσεις.



Σχήμα: Κατηγορίες και τύποι μεταβλητών

Προεπεξεργασία

Όπως αναφέρθηκε ήδη, η προεπεξεργασία των δεδομένων είναι ίσως το σημαντικότερο βήμα για την ΑΓΒΔ.

Γι' αυτό και θα πρέπει να έχει προηγηθεί κάποια προεπεξεργασία των δεδομένων που θα χρησιμοποιηθούν, ώστε να εξασφαλιστεί η ποιότητά τους.

Στην συνέχεια θα παρουσιάζουμε τις βασικότερες διεργασίες που λαμβάνουν χώρα κατά το στάδιο της προεπεξεργασίας των δεδομένων.

Καθαρισμός Δεδομένων

Οι βασικότερες δραστηριότητες του καθαρισμού δεδομένων είναι:

- συμπλήρωση ελλιπών τιμών,
- αναγνώριση ακραίων τιμών (outliers) και εξομάλυνση, εφόσον περιέχουν θόρυβο, και τέλος
- διόρθωση τυχόν ασυνεπειών στα δεδομένα.

Τα δεδομένα δεν είναι πάντα διαθέσιμα.

Πιο συγκεκριμένα, σε αρκετές πλειάδες (γραμμές) του συνόλου δεδομένων δεν υπάρχουν καταγεγραμμένες τιμές για ορισμένα από τα χαρακτηριστικά.

Αυτό το φαινόμενο είναι γνωστό ως ελλιπείς τιμές.

Μπορεί να οφείλεται σε διάφορους παράγοντες, όπως για παράδειγμα σε δυσλειτουργία του εξοπλισμού, σε ασυνέπειες με άλλα καταγεγραμμένα δεδομένα, που οδήγησε στη διαγραφή τους ή ακόμα και στη μη καταχώρηση τους.

Σε κάθε περίπτωση, τα ελλιπή δεδομένα ενδεχομένως να πρέπει να τα συμπεράνουμε και να τα συμπληρώσουμε.

Το πρώτο βήμα στον χειρισμό των ελλιπών δεδομένων είναι η αναγνώριση των πλειάδων με ελλιπείς τιμές.

Στη συνέχεια γίνεται η συμπλήρωση τους. Προφανώς, αν το σύνολο δεδομένων είναι μεγάλο, αυτό δεν μπορεί να γίνει χειρωνακτικά.

Η πιο εύκολη λύση είναι να αγνοήσουμε τη συγκεκριμένη πλειάδα-γραμμή.

Ωστόσο, αν έχουμε μεγάλο αριθμό ελλιπών τιμών αυτό δεν αποτελεί αποτελεσματική λύση.

Μερικές από τις πιο αποτελεσματικές, αυτοματοποιημένες μεθόδους συμπλήρωσης ελλιπών τιμών είναι οι ακόλουθες:

- Χρήση καθολικής σταθεράς για τη συμπλήρωση των χαμένων τιμών, π.χ. -1, “unknown”, νέα κλάση.
- Χρήση της μέσης τιμής του χαρακτηριστικού για τη συμπλήρωση των χαμένων τιμών.
- Χρήση της μέσης τιμής των δειγμάτων της ίδια κλάσης για τη συμπλήρωση των χαμένων τιμών.
- Χρήση της πιο πιθανής τιμής για τη συμπλήρωση των χαμένων τιμών, η οποία παράγεται από κάποια τεχνική, όπως η παλινδρόμηση, τα δένδρα απόφασης κ.ά.

Τα δεδομένα ενδεχομένως να είναι διαθέσιμα, αλλά να υπάρχει θόρυβος ή ακραίες τιμές σε αυτά.

Για παράδειγμα, μπορεί να έχουμε λανθασμένες τιμές χαρακτηριστικών λόγω προβληματικού εξοπλισμού λήψης μετρήσεων ή λόγω κάποιου προβλήματος κατά την εγγραφή των δεδομένων.

Υπάρχουν αρκετές μέθοδοι για τον χειρισμό δεδομένων με θόρυβο.

Θα εστιάσουμε στις μεθόδους **ενδοχείασης (binning) και στη συσταδοποίηση**.

Οι μέθοδοι ενδοχείασης έχουν όλες ως πρώτο βήμα την ταξινόμηση των δεδομένων, έτσι ώστε στη συνέχεια να διαχωριστούν σε δοχεία (bins).

Διακρίνονται με βάση τον διαμερισμό σε δοχεία, σε μεθόδους διαμερισμού ίσου πλάτους (απόσταση) και σε μεθόδους διαμερισμού ίσου βάθους (συχνότητα).

Κατά τον διαμερισμό ίσου πλάτους, το εύρος διαιρείται σε Ν διαστήματα ίσου μεγέθους.

Ωστόσο, αυτός ο διαμερισμός είναι επιρρεπής σε ακραίες τιμές, καθώς τα ασύμμετρα δεδομένα δεν διαχειρίζονται σωστά.

Κατά τον διαμερισμό ίσου βάθους, το εύρος διαιρείται σε N διαστήματα, τα οποία περιέχουν τον ίδιο αριθμό δειγμάτων. Σε αυτή την περίπτωση έχουμε καλύτερη κλιμάκωση των δεδομένων.

Οι μέθοδοι ενδοχείασης χρησιμοποιούνται και για διακριτοποίηση.

Οι πιο γνωστές είναι οι ακόλουθες:

- Ομαλοποίηση με βάση τη μέση τιμή του κάθε δοχείου: οι τιμές αντικαθίστανται με τη μέση τιμή κάθε δοχείου.
- Ομαλοποίηση με χρήση του μεσαίου (median) του κάθε δοχείου: οι τιμές αντικαθίστανται με τη μεσαία (median) τιμή κάθε δοχείου.
- Ομαλοποίηση με χρήση των ορίων του κάθε δοχείο

Παράδειγμα - Εξομάλυνση δεδομένων με μεθόδους ενδοχείασης

Έστω ότι μας δίνονται κάποιες θερμοκρασίες (σε $^{\circ}$C) ταξινομημένες σε αύξουσα σειρά:
4, 9, 11, 16, 21, 23, 24, 24, 27, 30, 32, 35.

Με διαμερισμό ίσου βάθους έχουμε τα εξής δοχεία:

- Δοχείο 1: 4, 9, 11, 16
- Δοχείο 2: 21, 23, 24, 24
- Δοχείο 3: 27, 30, 32, 35

4, 9, 11, 16, 21, 23, 24, 24, 27, 30, 32, 35.

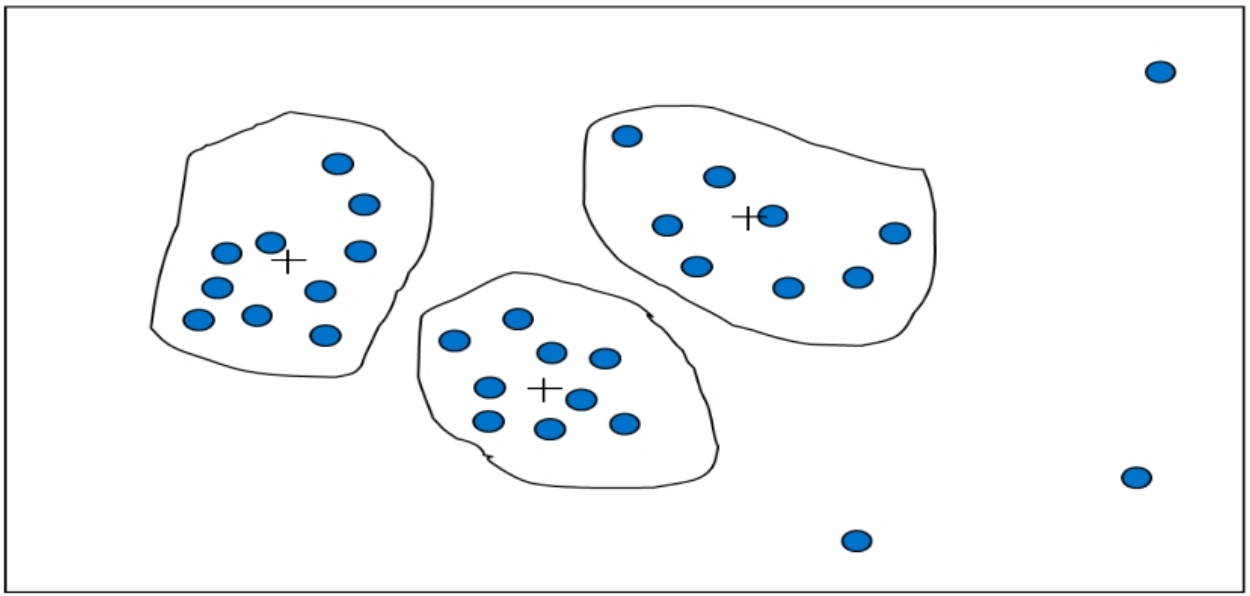
Χρησιμοποιώντας ομαλοποίηση με βάση τη μέση τιμή του κάθε δοχείου, έχουμε:

- Δοχείο 1: 10, 10, 10, 10
- Δοχείο 2: 23, 23, 23, 23
- Δοχείο 3: 31, 31, 31, 31

Χρησιμοποιώντας ομαλοποίηση με χρήση των ορίων του κάθε δοχείου, έχουμε:

- Δοχείο 1: 4, 4, 16, 16
- Δοχείο 2: 21, 24, 24, 24
- Δοχείο 3: 27, 27, 35, 35

Η χρήση συσταδοποίησης έχει ως στόχο την ομαδοποίηση των δεδομένων σε συστάδες (clusters), έτσι ώστε τα δεδομένα με θόρυβο να διαχωριστούν από τα καθαρά δεδομένα.
Για παράδειγμα, στο παρακάτω σχήμα βλέπουμε ότι έχουν δημιουργηθεί 3 συστάδες και ότι οι ακραίες τιμές δεν ανήκουν σε καμία από αυτές.



Σχήμα: Εφαρμογή συσταδοποίησης για ανίχνευση ακραίων τιμών.

Ασυνέπεια στα δεδομένα έχουμε, όταν δυο ή περισσότερες διαφορετικές πηγές ή αρχεία έχουν διαφορετικές εκδόσεις αποθηκευμένων δεδομένων, τα οποία θα έπρεπε να είναι ίδια.

Ασυνέπεια έχουμε, δηλαδή, όταν για την ίδια πραγματική οντότητα οι τιμές των χαρακτηριστικών από διαφορετικές πηγές διαφέρουν.

Συνήθως, αυτό συμβαίνει όταν έχουμε πλεονασμό δεδομένων και χρειασθεί να γίνει κάποια αλλαγή.

Τότε είναι πολύ πιθανό να γίνει διόρθωση μόνο σε κάποιο ή κάποια αρχεία και όχι σε όλα.

Άλλη πιθανή αιτία είναι ο διαφορετικός τρόπος αναπαράστασης ή και η χρήση διαφορετικών κλιμάκων, π.χ. μονάδες μέτρησης, διαφορετικό νόμισμα.

Η ενοποίηση των δεδομένων έχει ως στόχο τον συνδυασμό δεδομένων από πολλαπλές πηγές σε μια συνεκτική έκδοση.

Όταν τα δεδομένα είναι αποθηκευμένα σε βάσεις δεδομένων, πρέπει να υλοποιηθεί ενοποίηση σχήματος (schema integration) με χρήση των μεταδεδομένων που υπάρχουν από τις διάφορες πηγές.

Κατά τη διαδικασία της ενοποίησης πρέπει να ανιχνευτούν και να αναλυθούν πιθανές συγκρούσεις ή ασυνέπειες μεταξύ των τιμών των δεδομένων.

Τα πλεονάζοντα δεδομένα εμφανίζονται συχνά, όταν συνενώνονται πολλαπλές βάσεις δεδομένων.

Πιθανά προβλήματα, που μπορούν να προκύψουν στην προσπάθεια συνένωσης, είναι η χρήση διαφορετικού ονόματος σε διαφορετικές βάσεις δεδομένων ή όταν ένα γνώρισμα είναι παραγόμενο γνώρισμα σε άλλο πίνακα.

Για να εντοπίσουμε τα πλεονάζοντα δεδομένα χρησιμοποιείται ανάλυση συσχετίσεων.

Τέλος, αξίζει να αναφέρουμε ότι με προσεκτική ενοποίηση μπορούν να αφαιρεθούν περιττές πληροφορίες και να αποφευχθούν ασυνέπειες, να βελτιωθεί σημαντικά η ταχύτητα της διαδικασίας εξόρυξης δεδομένων και να αυξηθεί η ποιότητα των αποτελεσμάτων της.

Ο μετασχηματισμός των δεδομένων έχει ως βασικό στόχο τη δημιουργία συγκρίσιμων δεδομένων, τα οποία αρχικά είναι μη συγκρίσιμα.

Με τον μετασχηματισμό των δεδομένων μπορούμε να πετύχουμε και άλλα θετικά αποτελέσματα, όπως μείωση του όγκου των δεδομένων, π.χ. με το μετασχηματισμό των τιμών ενός χαρακτηριστικού σε κάποιο υποσύνολο τους (πλήρης ημερομηνία-/μόνο έτος), και μεγαλύτερη ακρίβεια των αποτελεσμάτων των αλγορίθμων εξόρυξης, π.χ. με το μετασχηματισμό των τιμών των χαρακτηριστικών σε κοινό εύρος (εύρος [0, 1]).

Η διακριτοποίηση μπορεί να θεωρηθεί ως μια ειδική μορφή μετασχηματισμού δεδομένων.

Η βασική ιδέα είναι η μετατροπή ενός συνεχούς εύρους τιμών σε διακριτές τιμές ή ετικέτες.

Θα δούμε παρακάτω ότι σε κάποιες περιπτώσεις, η διακριτοποίηση είναι αναγκαία για την εκτέλεση κάποιων τεχνικών εξόρυξης δεδομένων.

Ο μετασχηματισμός των δεδομένων χρησιμοποιείται κυρίως:

- για την εξομάλυνση των δεδομένων και την απομάκρυνση θορύβου,
- για τη συνάθροιση των δεδομένων, δηλαδή παραγωγή σύνοψης τους,
- για την κανονικοποίησή τους, δηλαδή την κλιμάκωση των χαρακτηριστικών του συνόλου δεδομένων σε ένα συγκεκριμένο και περιορισμένο εύρος τιμών, και
- για τη δημιουργία νέων χαρακτηριστικών από τα ήδη υπάρχοντα.

Η πιο συχνή εφαρμογή του μετασχηματισμού δεδομένων είναι η **κανονικοποίηση** και η δημιουργία νέων χαρακτηριστικών από τα ήδη υπάρχοντα.

Η κανονικοποίηση είναι ιδιαίτερα χρήσιμη σε προβλήματα κατηγοριοποίησης, καθώς και όταν τα δεδομένα έχουν τελείως διαφορετικές κλίμακες και μονάδες μέτρησης.

Υπάρχουν διαφορετικοί τρόποι κανονικοποίησης των δεδομένων. Οι πιο βασικοί είναι οι ακόλουθοι:

- **Κανονικοποίηση** min-max: οι τιμές κανονικοποιούνται, ώστε το εύρος τους να ανήκει σε ένα νέο, περιορισμένο εύρος, π.χ. [-1, 1], [0, 1] κοκ.

  Η νέα τιμή του χαρακτηριστικού σε κάθε πλειάδα υπολογίζεται χρησιμοποιώντας τον τύπο:

$$v_{new} = \frac{v - min}{max - min}(max_{new} - min_{new}) + min_{new}$$

Κανονικοποίηση με δεκαδική κλίμακα: οι τιμές κανονικοποιούνται με τάξεις μεγέθους του 10.

Η κανονικοποίηση γίνεται με τον τύπο:

$$v_{new} = \frac{v}{10^j}$$

όπου j είναι ο μικρότερος ακέραιος τέτοιος ώστε: $max(v_{new}) < 1$

Καθαρισμός Δεδομένων

Παράδειγμα - Κανονικοποίηση Δεδομένων

Έστω ότι δίνεται ένα πλαίσιο δεδομένων με ηλικίες και ύψη μαθητών. Θέλουμε να κανονικοποιήσουμε και τα δυο χαρακτηριστικά στο διάστημα [0, 1].

Ο παρακάτω Κώδικας υλοποιεί σε R την παραπάνω διαδικασία.

```
> age <- c(15,23,12,32)
> height <- c(172,185,130,178)
> mydf <- data.frame(age, height)
> mydf
age height
1   15     172
2   23     185
3   12     130
4   32     178
}
```

```
# [0,1] Normalization
> M <- sapply(mydf, max)
> M
age height
32    185

# Find min values
m <- sapply(mydf, min)
> m
age height
12    130
```

```
> mydf$age <-((mydf$age - m[1])/(M[1] - m[1]))* (1 - 0) + 0
> mydf$height <-((mydf$height - m[2])/(M[2] - m[2])) * (1 - 0) + 0
> mydf
age      height
1 0.15 0.7636364
2 0.55 1.0000000
3 0.00 0.0000000
4 1.00 0.8727273
```

Καθαρισμός Δεδομένων

Διακριτοποίηση Δεδομένων

Η διακριτοποίηση σχετίζεται με 3 τύπους χαρακτηριστικών:

- Ονομαστικά χαρακτηριστικά, όπου οι τιμές είναι ένα μη διατεταγμένο σύνολο.
- Διατακτικά χαρακτηριστικά, όπου οι τιμές είναι ένα διατεταγμένο σύνολο.
- Συνεχή χαρακτηριστικά, όπου οι τιμές είναι πραγματικοί αριθμοί.

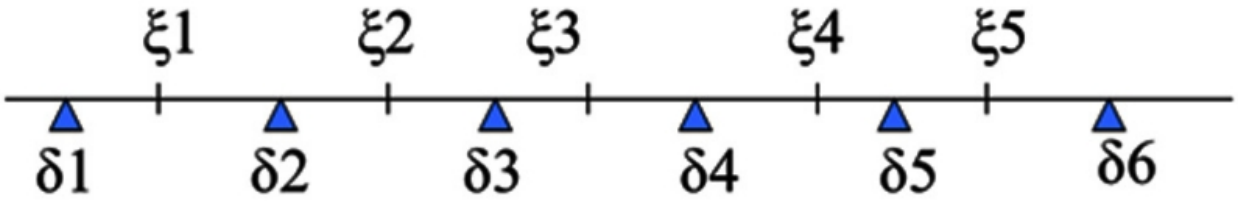
Ένα παράδειγμα διακριτοποίησης είναι η δειγματοληψία από το εύρος ενός συνεχούς χαρακτηριστικού.

Ο κύριος λόγος ύπαρξης της διακριτοποίησης είναι ότι κάποιοι αλγόριθμοι κατηγοριοποίησης δέχονται μόνο κατηγορικά χαρακτηριστικά.

Ακόμα μπορεί να συντελέσει στη μείωση του αριθμού και συνεπώς του μεγέθους των δεδομένων.

Για ένα δεδομένο συνεχές χαρακτηριστικό μπορούμε να διαχωρίσουμε το εύρος του σε διαστήματα και να αναθέσουμε ετικέτες στο κάθε διάστημα (βλέπε παρακάτω εικόνα).

Για παράδειγμα, μια τιμή που ανήκει στο διάστημα [ξ1, ξ2) θα αντικατασταθεί από την ετικέτα δ2.



Σχήμα: Παράδειγμα διακριτοποίησης

Όπως ήδη αναφέραμε, η ενδοχείαση (binning) μπορεί να χρησιμοποιηθεί για διακριτοποίηση.

Μια ακόμα τεχνική διακριτοποίησης είναι η διακριτοποίηση με χρήση της εντροπίας.

Έστω ένα σύνολο δειγμάτων S. Αν το S διαχωρίζεται σε δυο διαστήματα S_1 και S_2, χρησιμοποιώντας ένα κατώφλι T για τις τιμές του χαρακτηριστικού A, τότε το κέρδος πληροφορίας που προκύπτει από τον διαχωρισμό είναι:

$$I(S,T) = \frac{|S_1|}{|s|}E(S_1) + \frac{|S_2|}{|s|}E(S_2)$$

όπου η συνάρτηση εντροπίας E για ένα δεδομένο σύνολο υπολογίζεται με βάση την κατανομή της κλάσης των δειγμάτων στο σύνολο.

Αν έχουμε m κλάσεις, η εντροπία για το διάστημα S_1 είναι:

$$E(S_1) = -\sum_{i=1}^{m} p_i log_2(p_i)$$

όπου p_i είναι η πιθανότητα της κλάσης i στο S_1.

Η διαδικασία εφαρμόζεται αναδρομικά σε διαχωρισμούς, μέχρι να ικανοποιηθεί κάποιο κριτήριο τερματισμού, π.χ.

$$G(S,T) = E(S) - I(S,T) \leq \delta$$

όπου δ είναι ένας αρκετά μικρός αριθμός. Με άλλα λόγια, η διαδικασία εφαρμόζεται αναδρομικά μέχρι να μην έχουμε ουσιαστικό κέρδος από περαιτέρω διαχωρισμούς.

Πειράματα έχουν δείξει ότι η διακριτοποίηση μπορεί να μειώσει το μέγεθος των δεδομένων, βελτιώνοντας την ακρίβεια της κατηγοριοποίησης.

Παρακάτω παρουσιάζουμε ένα σύνολο δεδομένων με τις ώρες μελέτης για την εξέταση ενός μαθήματος και το αν τελικά οι μαθητές πέτυχαν στην αντίστοιχη εξέταση του μαθήματος, δηλαδή αν πέρασαν το σχετικό μάθημα ή όχι (N = ΝΑΙ, O = ΟΧΙ).

Ώρες Μελέτης	Επιτυχία Εξέτασης Μαθήματος
4	O
5	N
8	O
12	N
15	N

Οι ώρες μελέτης είναι η συνεχής μεταβλητή.

Θέλουμε να διακριτοποιήσουμε τα δεδομένα.

Ξεκινάμε υπολογίζοντας την εντροπία του συνόλου δεδομένων.

Για την επιτυχία εξέτασης μαθήματος έχουμε τρία N (NAI), και δυο O (OXI).

Συνεπώς,

$$E(S) = -\left(\frac{3}{5}log_2\left(\frac{3}{5}\right) + \frac{2}{5}log_2\left(\frac{2}{5}\right)\right) = 0.529 + 0.442 = 0.971$$

Στη συνέχεια, θα πρέπει να βρούμε ποιος διαχωρισμός θα μας δώσει το μέγιστο κέρδος.

Για να βρούμε έναν διαχωρισμό, υπολογίζουμε το ημιάθροισμα δυο γειτονικών τιμών.

Για παράδειγμα, από τις δυο πρώτες τιμές έχουμε 5+4 = 9 και T= 9/2 = 4.5.

Επομένως, ο πρώτος πιθανός διαχωρισμός είναι στο T = 4.5.

Με βάση αυτό τον διαχωρισμό προκύπτουν οι τιμές που φαίνονται παρακάτω πίνακα

	Επιτυχία Εξέτασης	Αποτυχία Εξέτασης
≤ 4.5	0	1
> 4.5	3	1

Υπολογίζουμε την εντροπία για κάθε περίπτωση και το κέρδος του συγκεκριμένου διαχωρισμού:

$$E(S_{\leqslant 4.5}) = -\left(\frac{1}{1} log_2(1) + 0 log_2(0)\right) = 0 + 0 = 0$$

$$E(S_{>4.5}) = -\left(\frac{3}{4} log_2\left(\frac{3}{4}\right) + \frac{1}{4} log_2\left(\frac{1}{4}\right)\right) = 0.311 + 0.5 = 0.811$$

Επομένως, τώρα έχουμε:

$$I(S, 4.5) = \frac{1}{5}(0) + \frac{4}{5}(0.811) = 0.6488$$

και το καθαρό κέρδος από τον διαχωρισμό είναι:

$$G(S, 4.5) = E(S) - I(S, 4.5) = 0.971 - 0.6488 = 0.322$$

Παίρνοντας τις δυο επόμενες διαδοχικές τιμές, έχουμε 6 + 8 = 13, και T = 13/2 = 6.5.

Επομένως, ο δεύτερος πιθανός διαχωρισμός είναι στο T = 6.5.

Με βάση αυτό τον διαχωρισμό προκύπτουν οι τιμές που φαίνονται παρακάτω πίνακα

	Επιτυχία Εξέτασης	Αποτυχία Εξέτασης
≤ 6.5	1	1
> 6.5	2	1

Υπολογίζουμε την εντροπία για κάθε περίπτωση και το κέρδος του συγκεκριμένου διαχωρισμού:

$$E(S_{\leq 6.5}) = -\left(\frac{1}{2}log_2\left(\frac{1}{2}\right) + \frac{1}{2}log_2\left(\frac{1}{2}\right)\right) = 0.5 + 0.5 = 1$$

$$E(S_{>6.5}) = -\left(\frac{2}{3}log_2\left(\frac{2}{3}\right) + \frac{1}{3}log_2\left(\frac{1}{3}\right)\right) = 0.389 + 0.528 = 0.917$$

Επομένως, τώρα έχουμε:

$$I(S, 6.5) = \frac{2}{5}(1) + \frac{3}{5}(0.917) = 0.95$$

και το καθαρό κέρδος από τον διαχωρισμό είναι:

$$G(S, 6.5) = E(S) - I(S, 6.5) = 0.971 - 0.95 = 0.021$$

Συνεχίζοντας το παράδειγμα, παίρνουμε τις δυο επόμενες διαδοχικές τιμές.

Έχουμε 8 + 12 = 20, και T = 20/2 = 10.

Επομένως, ο τρίτος πιθανός διαχωρισμός είναι στο T = 10. Με βάση αυτό τον διαχωρισμό προκύπτουν οι τιμές που φαίνονται παρακάτω πίνακα

	Επιτυχία Εξέτασης	Αποτυχία Εξέτασης
≤ 10	1	2
> 10	2	0

Υπολογίζουμε την εντροπία για κάθε περίπτωση και το κέρδος του συγκεκριμένου διαχωρισμού:

$$E(S_{\leq 10}) = -\left(\frac{1}{3}log_2\left(\frac{1}{3}\right) + \frac{2}{3}log_2\left(\frac{2}{3}\right)\right) = 0.528 + 0.389 = 0.917$$

$$E(S_{>10}) = -\left(\frac{2}{2}log_2\left(\frac{2}{2}\right) + 0log_2(0)\right) = 0.0 + 0.0 = 0.0$$

Επομένως, τώρα έχουμε:

$$I(S,10) = \frac{2}{5}(0) + \frac{3}{5}(0.917) = 0.55$$

και το καθαρό κέρδος από τον διαχωρισμό είναι:

$$G(S,10) = E(S) - I(S,10) = 0.971 - 0.55 = 0.421$$

Τέλος, παίρνουμε τις δυο τελευταίες διαδοχικές τιμές.

Έχουμε 12 + 15 = 27, και T = 27/2 = 13.5. Επομένως, ο τέταρτος και τελευταίος πιθανός διαχωρισμός είναι στο T = 13.5.

Με βάση αυτό τον διαχωρισμό προκύπτουν οι τιμές που φαίνονται παρακάτω πίνακα

	Επιτυχία Εξέτασης	Αποτυχία Εξέτασης
≤ 13.5	2	2
> 13.5	1	0

Υπολογίζουμε την εντροπία για κάθε περίπτωση και το κέρδος του συγκεκριμένου διαχωρισμού:

$$E(S_{\leq 13.5}) = -\left(\frac{2}{4}log_2\left(\frac{2}{4}\right) + \frac{2}{4}log_2\left(\frac{2}{4}\right)\right) = 0.5 + 0.5 = 1$$

$$E(S_{>13.5}) = -\left(\frac{1}{1}log_2(1) + 0log_2(0)\right) = 0.0 + 0.0 = 0.0$$

Επομένως, τώρα έχουμε:

$$I(S, 13.5) = \frac{1}{5}(0) + \frac{4}{5}(0.917) = 0.8$$

και το καθαρό κέρδος από τον διαχωρισμό είναι:

$$G(S, 13.5) = E(S) - I(S, 13.5) = 0.971 - 0.8 = 0.2$$

Από τα παραπάνω συμπεραίνουμε ότι ο τρίτος πιθανός διαχωρισμός, στο T = 10, είναι καλύτερος με το μέγιστο κέρδος (0.421).

Μετά τον διαχωρισμό μπορούμε να συνεχίσουμε, εξετάζοντας νέες περιπτώσεις διαχωρισμού και επιλέγοντας πάλι τον καλύτερο.

Η διαδικασία μπορεί να συνεχιστεί μέχρι να μην έχουμε κέρδος από περαιτέρω διαχωρισμούς, με βάση κάποια μικρή τιμή για το δ.

Ενότητες Παρουσίασης

Μείωση Δεδομένων

Το πρόβλημα που προσπαθεί να αντιμετωπίσει η μείωση των δεδομένων είναι ο τεράστιος όγκος δεδομένων προς επεξεργασία, καθώς η ανάλυση σύνθετων δεδομένων ενδεχομένως να απαιτεί απαγορευτικά πολύ χρόνο για να εκτελεστεί σε ολόκληρο το σύνολο δεδομένων.

Η διαδικασία της μείωσης των δεδομένων έχει ως στόχο την παραγωγή μιας μειωμένης αναπαράστασης του συνόλου δεδομένων, η οποία είναι αρκετά μικρότερη σε μέγεθος, αλλά που να μπορεί να παράγει ίδια ή παραπλήσια αποτελέσματα.

Μείωση Δεδομένων

Μείωση Διαστάσεων

Όσο περισσότερες διαστάσεις έχουμε, τόσο πιο δύσκολη είναι η διαχείριση των δεδομένων και τόσο πιο αραιά (sparse) είναι τα δεδομένα μας.

Το τελευταίο φαινόμενο είναι γνωστό στη βιβλιογραφία ως η κατάρα της διαστατικότητας (curse of dimensionality). Η μείωση των διαστάσεων έχει ως στόχο την ευκολότερη διαχείριση, κατανόηση και οπτικοποίηση των δεδομένων, ενώ ταυτόχρονα μειώνει τις απαιτήσεις σε χώρο μνήμης και σε χρόνο εκτέλεσης των αλγορίθμων εξόρυξης δεδομένων και μηχανικής μάθησης.

Δυο βασικές προσεγγίσεις, με τις οποίες μπορεί να επιτευχθεί μείωση των διαστάσεων, είναι

1. η επιλογή χαρακτηριστικών και
2. ο μετασχηματισμός των δεδομένων.

Μείωση Δεδομένων

Μείωση Διαστάσεων

Με βάση την προσέγγιση επιλογής χαρακτηριστικών επιλέγουμε το ελάχιστο πλήθος χαρακτηριστικών, με τα οποία είναι εφικτό να παραχθούν ισοδύναμα ή όσο το δυνατόν κοντινότερα αποτελέσματα με αυτά που θα παίρναμε, αν χρησιμοποιούσαμε όλα τα χαρακτηριστικά για ανάλυση.

Ιδανικά, ο αριθμός των χαρακτηριστικών που επιλέγονται είναι πολύ μικρότερος από τον αριθμό των αρχικών χαρακτηριστικών.

Ο πιο γνωστός μετασχηματισμός χαρακτηριστικών για μείωση των διαστάσεων είναι η **Ανάλυση Βασικών Συνιστωσών** (Principal Component Analysis, PCA).

Ο μετασχηματισμός των χαρακτηριστικών δημιουργεί ένα νέο σύνολο χαρακτηριστικών, λιγότερων διαστάσεων από το αρχικό, αλλά χωρίς μείωση των βασικών διαστάσεων.

Συχνά η PCA χρησιμοποιείται και για την οπτικοποίηση των δεδομένων.

Η **Ανάλυση Βασικών Συνιστωσών** λειτουργεί ως εξής:

Έχοντας N διανύσματα k-διαστάσεων βρίσκει $m \leqslant k$ ορθογώνια διανύσματα(*), τα οποία μπορούν να χρησιμοποιηθούν για τη βέλτιστη αναπαράσταση των δεδομένων.

Έτσι, το αρχικό σύνολο δεδομένων μειώνεται, ουσιαστικά προβάλλεται, σε ένα νέο, το οποίο αποτελείται από N διανύσματα δεδομένων πάνω σε m βασικές συνιστώσες.

Κάθε διάνυσμα δεδομένων είναι γραμμικός συνδυασμός των m διανυσμάτων βασικών συνιστωσών. Αυτή η τεχνική μπορεί να χρησιμοποιηθεί και με διατεταγμένα και με μη διατεταγμένα χαρακτηριστικά, ενώ χρησιμοποιείται κυρίως, όταν ο αριθμός των διαστάσεων είναι μεγάλος.

(*) Δύο διανύσματα x, y καλούνται ορθογώνια όταν το εσωτερικό τους γινόμενο είναι ίσο με 0, δηλαδή όταν $x^T y = 0$.

Μείωση Δεδομένων

Συμπίεση Δεδομένων

Μια ακόμα επιλογή για τη μείωση των δεδομένων είναι η συμπίεση τους.

Συμπίεση μπορούμε να κάνουμε σε διάφορες μορφές δεδομένων, όπως για παράδειγμα, σε αλφαριθμητικά. Εδώ υπάρχουν εκτενείς θεωρίες και αλγόριθμοι, ενώ συνήθως δεν έχουμε απώλεια πληροφορίας. Ωστόσο, εισάγονται περιορισμοί ως προς τη διαχείριση.

Επίσης σε βίντεο, ήχο και εικόνα, όπου στις περισσότερες περιπτώσεις έχουμε απώλεια πλη ροφορίας κατά τη συμπίεση.

Τέλος, σε χρονικές ακολουθίες, που δεν είναι ήχος, έχουν σύντομη διάρκεια και μεταβάλλονται αργά στον χρόνο.

Στόχος είναι η μείωση των δεδομένων και η χρήση μιας προσέγγισης των αρχικών δεδομένων, που όμως θα δώσει όσο το δυνατόν πλησιέστερα αποτελέσματα με αυτά που θα παίρναμε, αν χρησιμοποιούσαμε τα αρχικά δεδομένα.

Ενότητες Παρουσίασης

Το πακέτο dplyr χρησιμοποιείται για τον εύκολο χειρισμό των δεδομένων. Αναπτύχθηκε από τους Hadley Wickham και Roman Francois και παρέχει έτοιμες συναρτήσεις για συνεπή και περιεκτική διαχείριση δεδομένων σε μορφή πινάκων.

Η εγκατάσταση του πακέτου γίνεται με την εντολή install.packages(”d-plyr”), ενώ η φόρτωση του με την εντολή library(”dplyr”).

Το πρώτο βήμα για τη χρήση του πακέτου dplyr είναι η μετατροπή των δεδομένων σε συμβατή μορφή με το πακέτο.

Αυτό γίνεται εύκολα, καλώντας τη συνάρτηση tbl_df και δίνοντας ως όρισμα το αντικείμενο.

Το βασικό πλεονέκτημα χρήσης της tbl_df είναι ότι κάνει την αναπαράσταση κατά την εκτύπωση πιο συμπαγή και ευανάγνωστη.

Δραστηριότητα: Εκτελέστε το κομμάτι κώδικα που δίνεται πιο κάτω. Εκτυπώστε το περιεχόμενο του αρχικού πλαισίου δεδομένων airquality, το οποίο είναι ένα από τα έτοιμα σύνολα δεδομένων που παρέχονται από την R.

Τι διαφορές παρατηρείτε, ως προς τον τρόπο εκτύπωσης, σε σχέση με το νέο, tbl_df, αντικείμενο;

```
> library(dplyr)
> data(airquality)
> class(airquality)
[1] "data.frame"
> airquality <- as_tibble(airquality)
> class(airquality)
[1] "tbl_df"     "tbl"        "data.frame"
```

```
> airquality
# A tibble: 153 x 6
Ozone  Solar.R  Wind  Temp  Month  Day
<int>  <int>  <dbl>  <int>  <int>  <int>
1      41      190    7.4    67     5      1
2      36      118    8      72     5      2
3      12      149   12.6    74     5      3
4      18      313   11.5    62     5      4
5      NA       NA   14.3    56     5      5
6      28       NA   14.9    66     5      6
7      23      299    8.6    65     5      7
8      19       99   13.8    59     5      8
9       8       19   20.1    61     5      9
10     NA      194    8.6    69     5      10
```

Το πακέτο dplyr παρέχει 5 συναρτήσεις, οι οποίες καλύπτουν τις θεμελιώδεις εργασίες διαχείρισης δεδομένων. Αυτές είναι οι:

1. **select**, για επιλογή-φιλτράρισμα στηλών του συνόλου δεδομένων,
2. **filter**, για επιλογή-φιλτράρισμα γραμμών του συνόλου δεδομένων,
3. **arrange**, για ταξινόμηση των γραμμών βάσει τιμών συγκεκριμένων στηλών,
4. **mutate**, για δημιουργία νέων μεταβλητών από τις ήδη υπάρχουσες,
5. **summarize**, για συνάθροιση δεδομένων – ιδιαίτερα χρήσιμη σε συνδυασμό με ομαδοποιημένα δεδομένα.

Σε αρκετές περιπτώσεις, κυρίως όταν το σύνολο δεδομένων είναι μεγάλο σε μέγεθος, μας ενδιαφέρει μόνο ένα υποσύνολο των χαρακτηριστικών (features) του συνόλου δεδομένων.

Η συνάρτηση **select** μας επιτρέπει να επιλέξουμε συγκεκριμένες στήλες του συνόλου δεδομένων.

Αρκεί να δώσουμε τα ονόματα των στηλών και η **select** θα μας επιστρέψει τις στήλες με τη σειρά που ορίσαμε.

```
> select(airquality, Ozone, Solar.R, Day)
# A tibble: 153 x 3
Ozone  Solar.R    Day
<int>    <int> <int>
1      41      190      1
2      36      118      2
3      12      149      3
4      18      313      4
5      NA       NA      5
6      28       NA      6
7      23      299      7
8      19       99      8
9       8       19      9
10     NA      194     10
```

Επιπλέον, μπορούμε να επιλέξουμε πολλαπλές στήλες με χρήση του τελεστή ”:”, να επιλέξουμε ποια στήλη θέλουμε να παραλείψουμε με χρήση του ”-” μπροστά από τα ονόματα στηλών ή να παραλείψουμε πολλαπλές στήλες με συνδυασμό των προηγούμενων τελεστών.

```
> select(airquality, -(Wind:Month))
# A tibble: 153 x 3
Ozone  Solar.R    Day
<int>    <int> <int>
1      41      190      1
2      36      118      2
3      12      149      3
4      18      313      4
5      NA       NA      5
6      28       NA      6
7      23      299      7
8      19       99      8
```

```
> filter(airquality, Month > 5, Month < 9, Day < 3)
# A tibble: 6 x 6
Ozone Solar.R  Wind  Temp Month   Day
<int>   <int> <dbl> <int> <int> <int>
1    NA     286   8.6    78     6     1
2    NA     287   9.7    74     6     2
3   135     269   4.1    84     7     1
4    49     248   9.2    85     7     2
5    39      83   6.9    81     8     1
6     9      24  13.8    81     8     2
```

Σε περίπτωση που μας αρκεί οι γραμμές να ικανοποιούν μία από δύο συνθήκες, μπορούμε να χρησιμοποιήσουμε τον τελεστή **“|” (λογικό OR)**.

Γενικά, μπορούμε να χρησιμοποιήσουμε όλους του συγκριτικούς τελεστές με αριθμητικά δεδομένα.

```
> filter(airquality, Day == 1 | Day == 2)
# A tibble: 10 x 6
Ozone  Solar.R  Wind  Temp  Month  Day
<int>  <int>  <dbl>  <int>  <int>  <int>
1      41      190    7.4    67     5      1
2      36      118    8      72     5      2
3      NA      286    8.6    78     6      1
4      NA      287    9.7    74     6      2
5     135      269    4.1    84     7      1
6      49      248    9.2    85     7      2
7      39       83    6.9    81     8      1
```

Για την ταξινόμηση των γραμμών με βάση την τιμή συγκεκριμένων στηλών χρησιμοποιούμε τη συνάρτη- ση arrange. Η συνάρτηση ταξινομεί τις γραμμές με βάση τη σειρά που θα δώσουμε τα ονόματα των στηλών ως ορίσματα.

Η προεπιλεγμένη ταξινόμηση είναι κατά αύξουσα σειρά. Αν θέλουμε η ταξινόμηση να γίνει κατά φθίνουσα σειρά, θα πρέπει να το ορίσουμε ρητά, δίνοντας το όνομα της αντίστοιχης στήλης στη συνάρτηση **desc**.

```
> arrange(airquality, Ozone, desc(Solar.R))
# A tibble: 153 x 6
Ozone Solar.R  Wind  Temp Month   Day
<int>   <int> <dbl> <int> <int> <int>
1       1       8   9.7    59     5    21
2       4      25   9.7    61     5    23
3       6      78  18.4    57     5    18
4       7      49  10.3    69     9    24
5       7      48  14.3    80     7    15
```

Χρησιμοποιώντας τη συνάρτηση mutate, μπορούμε να δημιουργήσουμε νέες μεταβλητές-χαρακτηριστικά από τις ήδη υπάρχουσες.

Η συγκεκριμένη συνάρτηση είναι ιδιαίτερα χρήσιμη, για παράδειγμα, όταν θέλουμε να κάνουμε μετατροπή μονάδων μέτρησης.

Μπορούμε να δημιουργήσουμε πολλές νέες μεταβλητές με μόνο μια κλήση της συνάρτησης.

Ένα πολύ χρήσιμο χαρακτηριστικό της συγκεκριμένης συνάρτησης είναι ότι μπορούμε να δώσουμε δικά μας ονόματα στις νέες μεταβλητές και να τα χρησιμοποιήσουμε άμεσα στην ίδια κλήση της συνάρτησης, για δημιουργία κι άλλων μεταβλητών.

Ουσιαστικά, το νέο χαρακτηριστικό δεν είναι παρά η μετατροπή του χαρακτηριστικού θερμοκρασίας Temp από Fahrenheit σε βαθμούς Κελσίου.

```
> mutate(airquality, Temp.C = round((Temp - 32)*5/9))
# A tibble: 153 x 7
Ozone  Solar.R  Wind  Temp  Month   Day  Temp.C
<int>   <int> <dbl> <int> <int> <int>  <dbl>
1     41      190   7.4     67     5     1     19
2     36      118   8       72     5     2     22
3     12      149  12.6     74     5     3     23
4     18      313  11.5     62     5     4     17
5     NA       NA  14.3     56     5     5     13
6     28       NA  14.9     66     5     6     19
7     23      299   8.6     65     5     7     18
8     19       99  13.8     59     5     8     15
9      8       19  20.1     61     5     9     16
10    NA      194   8.6     69     5    10     21
```

Αφαίρεση γραμμών με ελλιπείς τιμές στο χαρακτηριστικό Ozone

```
> airquality <- filter(airquality, !is.na(Ozone))
```

Ομαδοποίηση κατά μήνα

```
> by_month <- group_by(airquality, Month)
```

Εύρεση ελάχιστης, μέσης και μέγιστης τιμής κατά μήνα

```
> summarize(by_month, min(Ozone), mean(Ozone), max(Ozone))
```

Το πακέτο **tidyr** αναπτύχθηκε από τον Hadley Wickham και χρησιμοποιείται για την εύκολη διαχείριση κατά τον καθαρισμό των δεδομένων, δηλαδή τον μετασχηματισμό των δεδομένων σε κατάλληλη μορφή ώστε να είναι κατάλληλα προς χρήση.

Η εγκατάσταση του πακέτου γίνεται με την εντολή **install.packages(tidyr)**, ενώ η φόρτωση του με την εντολή **library(tidyr)**.

Τα «καθαρά» δεδομένα πρέπει να πληρούν κάποιες συνθήκες, οι οποίες διευκολύνουν την εξερεύνηση και ανάλυση τους.

Οι 3 θεμελιώδεις συνθήκες, οι οποίες πρέπει να ικανοποιούνται, είναι:

1. Κάθε μεταβλητή σχηματίζει μια στήλη στο σύνολο δεδομένων.
2. Κάθε παρατήρηση-μέτρηση σχηματίζει μια γραμμή στο σύνολο δεδομένων.
3. Κάθε μονάδα μέτρησης, που προκύπτει από τις εγγραφές των δεδομένων, σχηματίζει έναν ξεχωριστό πίνακα.

Η πρώτη προβληματική περίπτωση είναι όταν τα ονόματα στηλών είναι τιμές και όχι ονόματα μεταβλητών.

Για την αντιμετώπιση αυτού του προβλήματος χρησιμοποιούμε τη συνάρτηση gather.

Η συνάρτηση παίρνει ως ορίσματα ονόματα στηλών και τα συγκεντρώνει σε ζεύγη κλειδί-τιμή.

Το αρχικό σύνολο δεδομένων έχει ουσιαστικά 3 μεταβλητές: τον βαθμό, το φύλο και το πλήθος μαθητών.

Οι τιμές για το χαρακτηριστικό του φύλου των μαθητών εμφανίζονται ως ονόματα της δεύτερης και της τρίτης στήλης του συνόλου δεδομένων. Η τρίτη μεταβλητή είναι ο αριθμός των μαθητών για κάθε συνδυασμό βαθμού-φύλου.

```
> grade <- c("A", "B", "C")
> male <- c(9, 20, 16)
> female <- c(15, 23, 14)
> dat <- data.frame(grade, male, female)
> print (dat)
grade male female
1     A    9     15
2     B   20     23
3     C   16     14
```

Για να καθαρίσουμε τα δεδομένα πρέπει κάθε μεταβλητή να βρίσκεται σε ξεχωριστή στήλη.

Για τον σκοπό αυτό χρησιμοποιούμε τη συνάρτηση gather. Θέλουμε να ενοποιήσουμε τα δεδομένα ως προς το φύλο και το πλήθος, αφήνοντας τη στήλη του βαθμού απείραχτη.

Για τον λόγο αυτό χρησιμοποιούμε τον τελεστή "**–**" μπροστά από τη μεταβλητή grade

```
> gather(dat, sex, count, -grade)
grade      sex count
1         A    male     9
2         B    male    20
3         C    male    16
4         A  female    15
5         B  female    23
6         C  female    14
```

Η δεύτερη προβληματική περίπτωση που μπορεί να συναντήσουμε είναι όταν πολλαπλές μεταβλητές αποθηκεύονται σε μια στήλη.

Σε αυτή την περίπτωση θα πρέπει να συνδυάσουμε τις συναρτήσεις **gather** και **separate**.

Η συνάρτηση separate μετατρέπει μια στήλη σε πολλές βάσει ενός μοτίβου.

```
> grade <- c("A", "B", "C")
> male_i <- c(16, 13, 8)
> male_ii <- c(3, 7, 8)
> female_i <- c(8, 16, 9)
> female_ii <- c(7, 7, 5)
> dat <- data.frame(grade, male_i, male_ii,  female_i, female_ii)
> dat
grade male_i male_ii female_i female_ii
1     A     16       3        8         7
2     B     13       7       16         7
3     C      8       8        9         5
```

Το σύνολο δεδομένων είναι παρόμοιο με αυτό που είδαμε προηγουμένως.

Σε αυτή την περίπτωση έχουμε 2 διαφορετικά τμήματα μαθητών, i και ii, με το πλήθος των μαθητών για κάθε φύλο σε καθένα από τα τμήματα.

Παρατηρούμε ότι έχουμε πολλαπλές μεταβλητές σε κάθε στήλη.

Για τον καθαρισμό του συνόλου δεδομένων σε αυτή την περίπτωση πρέπει να γίνουν δυο ενέργειες.

Αρχικά με χρήση της gather συγκεντρώνουμε τα δεδομένα ως προς τη μεταβλητή που δηλώνει το φύλο και το τμήμα, και ως προς το πλήθος μαθητών.

```
> dat <- gather(dat, sex_class, count, -grade)
> dat
grade sex_class count
1         A     male_i    16
2         B     male_i    13
3         C     male_i     8
4         A    male_ii     3
5         B    male_ii     7
6         C    male_ii     8
7         A   female_i     8
8         B   female_i    16
9         C   female_i     9
10        A  female_ii     7
11        B  female_ii     7
```

Στη συνέχεια, με χρήση της separate διαχωρίζουμε τη στήλη sex_class σε δυο διαφορετικές.

Για τη συγκεκριμένη περίπτωση, η συνάρτηση μπόρεσε από μόνη της να εντοπίσει τον χαρακτήρα διαχωρισμού.

```
> separate(dat, sex_class, c("sex", "class"))
grade    sex class count
1     A   male     i    16
2     B   male     i    13
3     C   male     i     8
4     A   male    ii     3
5     B   male    ii     7
6     C   male    ii     8
7     A female     i     8
8     B female     i    16
9     C female     i     9
10    A female    ii     7
11    B female    ii     7
12    C female    ii     5
```

Μια τρίτη περίπτωση προβληματικών δεδομένων είναι όταν μεταβλητές αποθηκεύονται και σε γραμμές και σε στήλες.

Σε αυτή την περίπτωση θα πρέπει να συνδυάσουμε τις συναρτήσεις **gather** και **spread**.

Η συνάρτηση spread υλοποιεί την αντίθετη λειτουργία της **gather**.

Μετατρέπει ζεύγη κλειδιού-τιμής σε πολλαπλές στήλες.

Έστω ένα σύνολο δεδομένων, στο οποίο μεταβλητές αποθηκεύονται τόσο σε γραμμές όσο και σε στήλες.

Πιο συγκεκριμένα, η πρώτη μεταβλητή είναι τα ονόματα (name) των μαθητών.

Τα ονόματα των τεσσάρων τελευταίων στηλών αποτελούν τιμές για τη μεταβλητή μάθημα (lesson).

Οι τιμές της μεταβλητής τριμήνου (quarter) θα πρέπει να καταχωρηθούν σε διαφορετικές μεταβλητές με τον αντίστοιχο βαθμό για κάθε μαθητή.

```
> name <- cΚώστας(”” , Κώστας”” , Κώστας”” , Μαρία”” , Μαρία”” , Μαρία”” ,
> quarter <- c(1,2,3,1,2,3,1,2,3)
> lesson1 <- c(”A”,”A”,”A”,”B”,”C”,”A”,”C”,”B”,”A”)
> lesson2 <- c(”NA”,”NA”,”NA”,”B”,”A”,”A”,”B”,”B”,”A”)
> lesson3 <- c(”C”,”B”,”B”,”NA”,”NA”,”NA”,”A”,”B”,”B”)
> lesson4 <- c(”C”,”B”,”B”,”NA”,”NA”,”NA”,”NA”,”NA”,”NA”)
```

```
> dat
name quarter   lesson  grade
1  Κώστας        1  lesson1      A
2  Κώστας        2  lesson1      A
3  Κώστας        3  lesson1      A
4  Μαρία         1  lesson1      B
5  Μαρία         2  lesson1      C
6  Μαρία         3  lesson1      A
7  Παναγιώτης    1  lesson1      C
8  Παναγιώτης    2  lesson1      B
9  Παναγιώτης    3  lesson1      A
10  Κώστας       1  lesson2     NA
11  Κώστας       2  lesson2     NA
12  Κώστας       3  lesson2     NA
```

```
> dat <- spread(dat, quarter, grade)
> dat
name   lesson   1   2   3
1  Κώστας       lesson1   A   A   A
2  Κώστας       lesson2  NA  NA  NA
3  Κώστας       lesson3   C   B   B
4  Κώστας       lesson4   C   B   B
5  Μαρία        lesson1   B   C   A
6  Μαρία        lesson2   B   A   A
7  Μαρία        lesson3  NA  NA  NA
8  Μαρία        lesson4  NA  NA  NA
9  Παναγιώτης   lesson1   C   B   A
10 Παναγιώτης   lesson2   B   B   A
11 Παναγιώτης   lesson3   A   B   B
```

```
> mutate(dat, lesson = extract_numeric(lesson))
extract_numeric() is deprecated: please use readr::parse_number() in
name lesson  1  2  3
1  Κώστας      1  A  A  A
2  Κώστας      2 NA NA NA
3  Κώστας      3  C  B  B
4  Κώστας      4  C  B  B
5  Μαρία       1  B  C  A
6  Μαρία       2  B  A  A
7  Μαρία       3 NA NA NA
8  Μαρία       4 NA NA NA
9  Παναγιώτης  1  C  B  A
10 Παναγιώτης  2  B  B  A
11 Παναγιώτης  3  A  B  B
```

```
> mutate(dat, lesson = readr::parse_number(lesson))
name lesson  1  2  3
1  Κώστας       1  A  A  A
2  Κώστας       2 NA NA NA
3  Κώστας       3  C  B  B
4  Κώστας       4  C  B  B
5  Μαρία        1  B  C  A
6  Μαρία        2  B  A  A
7  Μαρία        3 NA NA NA
8  Μαρία        4 NA NA NA
9  Παναγιώτης   1  C  B  A
10 Παναγιώτης   2  B  B  A
11 Παναγιώτης   3  A  B  B
12 Παναγιώτης   4 NA NA NA
```