

**Wprowadzenie do obsługi programu QDA Miner 4.1.
i do kodowania dokumentów samorządowych
w ramach badania „Mapa wiedzy i niewiedzy”**

dr Paweł Tomanek
Obserwatorium Żywej Kultury
Warszawa 2013

Spis treści

1. Wprowadzenie	3
2. Słownik najważniejszych terminów	3
3. Tworzenie nowego pliku projektu.....	5
3.1. Tworzenie „gołej” struktury kodów	5
3.2. Importowanie z plików tekstowych.....	6
3.3. Importowanie z baz danych.....	7
4. Operacje na przypadkach.....	8
4.1. Dodawanie przypadków	8
4.2. Usuwanie przypadków	8
4.3. Filtrowanie przypadków	8
4.4. Grupowanie przypadków	9
5. Operacje na zmiennych.....	9
5.1. Określanie wartości zmiennych.....	9
5.2. Dodawanie zmiennych	10
5.3. Usuwanie zmiennych	10
5.4. Porządkowanie zmiennych.....	10
5.5. Edytowanie zmiennych	10
6. Operacje na kodach.....	11
6.1. Kodowanie materiału	11
6.2. Przekodowywanie	11
6.3. Dodawanie kodów	12
6.4. Usuwanie kodów	12
6.5. Edytowanie kodów	12
6.6. Przesuwanie kodów w strukturze	12
6.7. Łączenie kodów.....	13
6.8. Dzielenie kodów.....	13
7. Wybrane możliwości analizy zakodowanego materiału.....	13
7.1. Przeglądanie zakodowanych fragmentów	13
7.2. Analiza częstości występowania kodów	14
7.3. Analiza częstości występowania kodów według zmiennych	14
8. Omówienie listy zmiennych w badaniu „Mapa wiedzy i niewiedzy”	15
9. Omówienie struktury kodów w badaniu „Mapa wiedzy i niewiedzy”	15
10. Kilka praktycznych porad dotyczących kodowania.....	19

1. Wprowadzenie

Ten przewodnik powstał pod kątem konkretnego badania – „Mapy wiedzy i niewiedzy” – dlatego nie jest wyczerpującym wprowadzeniem do obsługi programu QDA Miner 4.1. Skupiłem się przede wszystkim na tych opcjach i narzędziach, które są konieczne do płynnego kodowania danych, a mniejszą uwagę poświęciłem opcjom analizy – poza trzema wybranymi, które okazały się najprzydatniejsze w badaniach na Warmii i Mazurach.

Również wprowadzenie do struktury kodów w tym badaniu jest z konieczności pobieżne (to ponad 300 kodów), dlatego w razie wątpliwości co do znaczenia poszczególnych kodów proszę o kontakt przez naszą grupę facebookową albo mailowo: pawel.tomanek@uw.edu.pl.

2. Słownik najważniejszych terminów

PLIK PROJEKTU (*project file*) – plik programu QDA Miner zawierający partię **kodowanego materiału** tekstowego lub graficznego podzieloną na **przypadki** (minimum jeden) i opisaną za pomocą **zmiennych** i **kodów**. Plik projektu można utworzyć od zera albo na podstawie już istniejących plików tekstowych lub baz danych, z których importuje się materiał (zob 2.2., 2.3.). Na jednym pliku może pracować tylko jeden użytkownik, ale pliki można łączyć lub porównywać je pod względem zgodności kodowania (jeżeli zawierają ten sam materiał).

KODOWANY MATERIAŁ – tekst lub grafika, które importuje się do **pliku projektu**, dzieli na **przypadki** i opisuje za pomocą **zmiennych** i **kodów**. Kodowany materiał wyświetla się w **oknie głównym programu (DOCUMENTS)**.

PRZYPADEK (*case*) – podstawowa jednostka organizacji **kodowanego materiału**. Najprościej mówiąc, przypadek to ta partia materiału (tekstu, obrazów itp.), którą możemy przewinąć w głównym oknie programu i opisać jako całość za pomocą **zmiennych**. W naszych badaniach pojedynczym przypadkiem jest **zawsze pojedynczy dokument strategiczny JST** (jednostki samorządu terytorialnego), a właściwie jego wcześniej wybrane fragmenty dotyczące kultury. To oznacza, że tekst z jednego dokumentu nie może się znaleźć w dwóch różnych przypadkach, ani tekst z dwóch różnych dokumentów – w jednym przypadku.

Lista (albo drzewo) przypadków wyświetla się w **lewym górnym oknie programu (CASES)**.

ZMIENNA (*variable*) – cecha lub wartość, którą możemy przypisać wyłącznie **przypadkowi jako całości**. Jeden przypadek można opisać za pomocą **wielu zmiennych**. Celem zmiennych jest po pierwsze identyfikacja poszczególnych przypadków (w naszych badaniach – określeniu powiatu, gminy lub miejscowości, której dotyczy dany dokument), a po drugie –

umożliwienie porównań między przypadkami pod względem cech istotnych badawczo (w naszych badaniach np. porównanie dokumentów różnych szczebli samorządu terytorialnego). Dlatego wartość danej zmiennej teoretycznie powinno się dać określić dla **każdego przypadku**. Wyjątkiem są **zmienne opcjonalne**, które z zasady dotyczą tylko **części przypadków** (np. zmienna „TYPGMINY” będzie dotyczyła tylko dokumentów gminnych, w pozostałych przypadkach jej wartość pozostanie nieokreślona). Szczegółowy opis zmiennych stosowanych w projekcie, zob. 8).

Zmienne w programie QDA Miner 4.1 mają kilka typów:

- **Nominal/Ordinal** – zmienna nominalna/porządkowa;
- **Numeric** – zmienna numeryczna;
- **Date** – zmienna daty;
- **Boolean** – zmienna logiczna typu „prawda/fałsz”;
- **Short string** – łańcuch dowolnych znaków;
- **Document** – zmienna tekstowa;
- **Image** – obraz.

W naszym badaniu używamy tylko dwóch: **zmiennych nominalnych** oraz **zmiennej tekstowej**, której program automatycznie nadaje nazwę DOCUMENT (można ją zmienić). Ta zmienna ma szczególne znaczenie – jeżeli nie ma jej na liście zmiennych, do pliku nie można importować ani wklejać żadnego materiału.

Lista zmiennych wyświetla się w **lewym środkowym oknie programu (VARIABLES)**.

KOD (code) – cecha lub wartość, którą możemy przypisać konkretnemu fragmentowi **kodowanego materiału**. Tego samego kodu można używać **wielokrotnie w obrębie jednego przypadku**, a ten sam fragment materiału można zakodować przy użyciu **różnych kodów**. Każdy kod jest przypisany do pewnej **kategorii**.

Nazwy kodów, za pomocą których zakodowano poszczególne fragmenty, wyświetlają się w **prawym oknie programu**. Oprócz tego można wybrać opcję podświetlania zakodowanych fragmentów kolorami przypisanymi poszczególnym kodom (zob. 6.1.)

KATEGORIA (category) – jednostka **struktury kodów** obejmująca jeden lub więcej kodów. Kategorie służą tylko do porządkowania struktury kodów – za ich pomocą nie można **kodować** materiału (można to robić tylko przy użyciu samych kodów). Każdy kod należy **tylko do jednej kategorii**, ale same kategorie mogą tworzyć wielopiętrową hierarchię (zob. struktura kodów).

STRUKTURA KODÓW (codebook) – uporządkowany układ wszystkich kodów i kategorii utworzonych w jednym **pliku projektu**. Struktura kodów w programie QDA Miner jest zawsze co najmniej dwustopniowa – musi istnieć przynajmniej jedna kategoria, do której należą utworzone kody. Najczęściej takich kategorii jest więcej i mogą one tworzyć

wielopiętrową hierarchię, w której główne kategorie dzielą się na podkategorie, podkategorie itd.

Struktura kodów wyświetla się w **lewym dolnym oknie programu (CODES)**.

KODOWANIE (*coding*) – zaznaczanie wybranego fragmentu **kodowanego materiału** i przypisywanie mu któregoś z wcześniej utworzonych **kodów** (lub kodu nowo utworzonego). Ten sam fragment można zakodować wielokrotnie przy użyciu różnych kodów.

3. Tworzenie nowego pliku projektu

W programie QDA Miner 4.1. istnieje kilka sposobów utworzenia nowego pliku. Można to zrobić po uruchomieniu programu, kiedy w pierwszy oknie wyświetla się opcja **Create a new project** (pozostałe dwie służą do otwarcia już istniejących projektów). To samo można zrobić z już działającego programu, wybierając z górnego menu **Project** → **New**.

3.1. Tworzenie „golej” struktury kodów

Minimalna opcja to druga od góry, **Create a blank project (design structure)** – po jej wybraniu dostajemy możliwość utworzenia projektu „od zera”, czyli od określenia zmiennych. Wyznaczamy nazwę zmiennej (**Variable name**), jej opis (**Description** – opcjonalnie) i typ.

UWAGA: jeżeli nie wybierzemy zmiennej typu **Document**, do projektu nie będzie można dołączać ani kopiować żadnych tekstów!

Ale ponieważ w „Mapie wiedzy i niewiedzy” mamy już gotową listę zmiennych i wstępną strukturę kodów, **lepiej zacząć inaczej**, w jeden z trzech polecanych przez nas sposobów:

- 1) Otworzyć przygotowany przez nas plik z „gołą” listą zmiennych i strukturą kodów, a potem dodawać do niego nowe przypadki i załączać lub przeklejać teksty strategii (zob. 4.1.).
- 2) Skorzystać z opcji **Document Conversion Wizard** (zob. 3.2). To chyba najlepszy sposób, bo umożliwia od razu określenie wartości zmiennych. Trzeba się tylko trochę napracować nad wcześniejszym przygotowaniem plików tekstowych, ale ten czas zwraca się później, podczas pracy w samym QDA Minerze.
- 3) Utworzyć nowy plik projektu na podstawie arkusza bazy danych (otwieramy program → **Create a new project** → **Create a project from a datafile**). Problem w tym, że w Excelu albo innym arkuszu kalkulacyjnym teksty strategii mogą stracić formatowanie, np. tabelki.

3.2. Importowanie z plików tekstowych

Po otwarciu programu wybieramy **Create a new document** → **Run Document Conversion Wizard**. W oknie **FILE SELECTION** dodajemy jeden albo więcej plików tekstowych, które mają stanowić materiał do kodowania. Żeby sprawdzić, czy program na pewno je wczyta, możemy je podejrzeć (**View documents**) – jeżeli w oknie podglądu pojawia się tekst, to znaczy, że wszystko w porządku. Jeżeli nie, to być może konieczne będzie przekonwertowanie pliku do któregoś z odczytywanych formatów (w mojej wersji najlepiej sprawdza się .doc albo .docx).

Potem wybieramy **Next** i przechodzimy do okna **TRANSFER OPTIONS**. Tu określamy zasadę podziału na przypadki. W zależności od tego, jak będziecie Państwo gromadzić dokumenty, przydatna będzie jedna z trzech opcji.

File – jeżeli każdy dokument będziecie Państwo mieli zapisaną w osobnym pliku (wtedy program zrobi z nich osobne przypadki).

Page – jeżeli wszystkie dokumenty będziecie Państwo mieli w jednym albo niewielu plikach. Wtedy program podzieli przypadki, zaczynając nowy za każdym razem od **znaku podziału strony** (W Wordzie można go zrobić tak: **Wstawianie** → **Podział strony**). To oznacza, że plik (pliki) z dokumentami trzeba wcześniej przejrzeć, wstawić znaki podziału strony między poszczególnymi dokumentami i upewnić się, że nie ma tych znaków w obrębie dokumentów.

Section – rozwiązanie podobne do poprzedniego, tylko że tutaj koniec przypadku i początek następnego wyznacza **wybrany ciąg znaków** (np. <koniec>). Musi mieć taką formę, żeby nie miał szans pojawić się jako zwykłe słowo w obrębie dokumentów. Tu też musimy wcześniej przygotować dokumenty i powstawiać ten ciąg między przypadkami; korzyść w porównaniu z poprzednią opcją jest taka, że nie musimy się upewniać, czy nie ma go w obrębie poszczególnych przypadków.

Czwarta opcja (trzecia w kolejności – **Line/Paragraph**) nam się nie przyda, bo z każdej linijki tekstu robi osobny przypadek.

Znowu **Next** i przechodzimy do okna **VARIABLES EXTRACTION**. Tu określamy, czy chcemy od razu odczytać z dokumentów wartości zmiennych. Można to pominąć i określić wartości zmiennych później, już w trakcie kodowania przypadków, ale jeżeli chcemy mieć to z głowy, to robi się to tak:

Po prawej stronie klikamy **Add** i otwiera się okno **Extract**. Wpisujemy nazwę zmiennej, jej typ i **Start delimiter** - ciąg znaków, po którym program rozpoznaje, że w tym miejscu dokumentu jest określona wartość zmiennej. Powiedzmy, że chcę odczytać z dokumentu jego nazwę, żeby program sam ją wstawił jako wartość zmiennej NAZWA. Wtedy na samym początku dokumentu wpisuję określony ciąg znaków, np. <nazwa_dokumentu>, a zaraz po

nim nazwę strategii. Żeby program wiedział, gdzie kończy się ta wartość, określamy **End delimiter** – najlepiej zostawić go tak, jak jest, z wartością **End of line**. Wybieramy jeszcze opcję **Delete strings and delimiters after succesful extraction** – wtedy te wpisane znaczniki i wartości zmiennych nie znajdą się w materiale pobranym do okna głównego. Klikamy **OK** i dodajemy kolejne zmienne (**Add**), aż wyczerpiemy listę. Wtedy klikamy **Next**, a w następnym oknie – **Create Project File**. Nowy plik projektu otworzy się automatycznie.

Oczywiście żeby to zadziałało, trzeba najpierw odpowiednio przygotować dokumenty. Na początku każdego dokumentu wstawiamy ciągi znaków wprowadzające wartości kolejnych zmiennych, odpowiadające tym, które określiliśmy w oknie **Extract**. Jeżeli znacznik końcowy chcemy ustawić na **End of line**, to kolejne ciągi wstawiamy w „słupku” w osobnych liniach. W dokumencie wygląda to np. tak:

```
<nazwa>Strategia rozwoju miasta i gminy Olecko  
<poziom>gminy  
<powiat>olecki  
<gmina>Olecko  
<typ_gminy>miejska
```

W ostatnim oknie warto zapisać wprowadzone ustawienia do pliku (**Save options in a file**), bo gdybyśmy mieli kiedyś tworzyć nowy projekt na bazie tych samych zmiennych, to nie trzeba będzie wpisywać ustawień ich wczytywania od nowa.

3.3. Importowanie z baz danych

Ta opcja może być mniej przydatna niż Document Conversion Wizard, bo uniemożliwia zachowanie formatowania dokumentów, ale jeżeli mamy teksty, które można bez większych strat przekopiować do Excela, to dzięki tej opcji łatwiej jest odczytać wartości zmiennych. Nie robiłem tego na innych bazach danych, więc moje wskazówki odnoszą się bezpośrednio do Excela.

Wybieramy **Create a new document** → **Import from an existing data file**, wybieramy plik Excela lub innego programu typu baza danych, nie zmieniamy niczego w ustawieniach importowania (jeżeli chcemy importować cały zakres bazy) i klikamy **Import**. Każdy wiersz Excela zostanie odczytany jako osobny przypadek, a wartości komórek w poszczególnych kolumnach – jako wartości kolejnych zmiennych. Dlatego przed importowaniem warto przygotować plik Excela – w pierwszym wierszu dać nagłówki kolumn i nazwać je nazwami zmiennych, które chcemy mieć w projekcie, a kolumnie z tekstami dokumentów obowiązkowo dać nagłówek DOCUMENT (bez tego zostanie potraktowana jak inne zmienne). Tu też nowy plik otworzy się automatycznie.

4. Operacje na przypadkach

4.1. Dodawanie przypadków

Kiedy mamy już utworzony plik projektu, dodawanie kolejnych przypadków jest banalnie łatwe. Klikamy zakładkę **Cases** w górnym pasku programu i wybieramy jedną z trzech opcji (pierwsze dwie można też wybrać, klikając prawym klawiszem myszy na okno **CASES**):

Add – otwiera się okno **Add new case**. Możemy od razu określić w nim wartości zmiennych dla tworzonego przypadku.

Append documents/images – otwiera się okno **Import Documents and Images**. Wybieramy dokumenty albo obrazy, z których chcemy zrobić nowe przypadki. Każdy dokument będzie potraktowany jako osobny przypadek. W oknie **Variable selection** najlepiej niczego nie ruszać – najważniejsze, żeby w punkcie **Store document in** była wybrana zmienna typu tekstowego, czyli **DOCUMENT** (jeżeli nie zmieniliśmy nazwy).

Append from a Datafile – w ten sposób można dodać do naszego obecnego pliku projektu przypadki z innego pliku projektu. Warunek: zmienne w obu plikach muszą się przynajmniej częściowo pokrywać.

4.2. Usuwanie przypadków

Klikamy prawym klawiszem myszy na okno **CASES** i wybieramy **Delete cases** (albo z górnego menu **Cases** → **Delete cases**), zaznaczamy na liście, które przypadki chcemy usunąć, i klikamy **OK**.

4.3. Filtrowanie przypadków

Jeżeli chcemy, aby widoczne były tylko przypadki o określonej wartości lub wartościach jednej lub kilku zmiennych, klikamy prawym klawiszem myszy na okno **CASES** i wybieramy **Filter cases** (albo z górnego menu **Cases** → **Filter**), wybieramy zmienną, według której chcemy filtrować, rodzaj operatora (**contains** = przypadki zawierające daną wartość; **does not contain** = przypadki nie zawierające danej wartości; **is empty** = przypadki, w których dana zmienna nie ma określonej wartości; **is not empty** = przypadki w których dana zmienna ma określoną wartość), a potem wartości zmiennej, które chcemy włączyć albo wykluczyć. Jeżeli chcemy filtrować jednocześnie według kilku zmiennych, ponawiamy całą operację w następujących wierszach. Aby zastosować filtrowanie, klikamy **Apply**.

4.4. Grupowanie przypadków

Jeżeli chcemy uporządkować przypadki nie według kolejności utworzenia (domyślna opcja), tylko wartości którejś ze zmiennych, to klikamy prawym klawiszem myszy na okno **CASES** i wybieramy **Case descriptor** (albo z górnego menu **Cases** → **Grouping / Case Descriptor**). Najpierw wybieramy, czy chcemy mieć przypadki uporządkowane w postaci drzewa (**Tree**), czy listy (**List**). Poniżej określamy, według wartości której zmiennej lub których zmiennych zostaną posortowane. Sortowanie będzie dwustopniowe, jeżeli określimy zarówno główną zmienną sortującą (**Main**), jak i zmienną drugorzędą (**Secondary**).

Przykład: wybieram jako główną zmienną POWIAT, a jako drugorzędą zmienną GMINA. Wtedy program posortuje mi przypadki tak, że te z jednego powiatu znajdą się obok siebie, a w obrębie tej grupy obok siebie znajdą się te z jednej gminy (jeżeli jest więcej niż jeden dokument dla danej gminy). Najlepiej widać to w formie drzewa.

Poniżej, w oknie **Case Description**, możemy zmienić sposób opisywania przypadków. Domyślnie są to same numery przypadków („Case #1”, „Case #2” itd.), ale możemy to zmodyfikować i dodać do tego wartość jednej albo kilku zmiennych. Wtedy w wierszu **Description String** powinien znaleźć się np. taki ciąg: „Przypadek {CASENUM}, {POWIAT}, {GMINA}”. Nazwy zmiennych w nawiasach można wpisać ręcznie albo dodać z wyświetlonej listy – tylko trzeba umieścić kursor po nawiasie z nazwą poprzedniej i dodać spację i przecinek, żeby wartości się nie zlewały.

Jeżeli tak zrobimy, to w oknie z listą przypadków, które widzimy podczas pracy w programie, będzie to wyglądać np. tak: „Przypadek 1, olecki, Olecko”.

Aby pogrupować przypadki według wybranej reguły, klikamy **Apply**.

5. Operacje na zmiennych

UWAGA: wszystkie omówione działania (oprócz określania wartości zmiennej) można też wykonać z górnego menu, z zakładki **Variables**.

5.1. Określanie wartości zmiennych

Lista zmiennych razem z wartościami dla bieżącego przypadku wyświetla się w **górnym lewym oknie programu (VARIABLES)**. Jeżeli chcemy określić wartość którejś z nich, klikamy w biały pasek na prawo od nazwy zmiennej. Jeżeli jest to zmienna typu nominalnego (jak większość naszych zmiennych) i utworzyliśmy już dla niej wcześniej jakieś wartości, to rozwinie się ich lista i wystarczy kliknąć na którąś z nich. Jeżeli chcemy utworzyć nową wartość, na tej samej liście klikamy **<new value>** i w okienku, które wyskoczy, wpisujemy wartość, a potem klikamy **OK**.

Wartości zmiennych dla danego przypadku można określić już podczas jego dodawania (zob. 4.1.).

5.2. Dodawanie zmiennych

Klikamy prawym klawiszem myszy na okno **VARIABLES** i wybieramy **Add**. Określamy obowiązkowo nazwę zmiennej (**Name** – duże litery bez polskich znaków) i jej typ (**Data type** – zob. Słowniczek), a ewentualnie dodajemy jej opis na własny użytek (**Description**). Jeżeli zmienna jest nominalna/porządkowa, musimy od razu określić przynajmniej jedną z jej wartości. Klikamy **Edit** obok **Values** i wpisujemy ręcznie wartości, każdą w osobnej linii. Polecenie **Link to values in** powoduje, że lista wartości tej zmiennej staje się taka sama jak lista wartości innej zmiennej (ale tylko lista – w konkretnych przypadkach same wartości są odrębne, tzn. określenie wartości dla jednej zmiennej nie pociąga za sobą jej określenia dla drugiej). Żeby utworzyć zmienną, klikamy **Create**.

5.3. Usuwanie zmiennych

Klikamy prawym klawiszem na nazwę dowolnej zmiennej w środkowym lewym oknie i wybieramy **Delete**. W oknie **Delete variables** przenosimy zmienne do wykasowania z lewego okna **Existing variables** do prawego **Variables to delete** (klikamy na nazwę zmiennej, a potem na strzałkę w prawo; analogicznie można to odwrócić). Żeby skasować wyznaczone zmienne, klikamy **OK**.

5.4. Porządkowanie zmiennych

Jeżeli chcemy zmienić kolejność, w której zmienne wyświetlają się w górnym lewym oknie, klikamy prawym klawiszem na nazwę dowolnej zmiennej i wybieramy **Reorder**. W oknie, które się otworzy, klikamy na nazwę zmiennej, którą chcemy przesunąć, i za pomocą strzałek przesuwamy ją w górę albo w dół. Kiedy osiągniemy taki porządek, o jaki nam chodziło, klikamy **OK**.

5.5. Edytowanie zmiennych

Żeby edytować zmienną, klikamy prawym klawiszem myszy na jej nazwę w lewym środkowym oknie programu, a potem wybieramy:

Edit properties – tu można wprowadzić nowy opis i zmienić nazwę zmiennej (**Rename**).

Transform {nazwa zmiennej} – tu można zmienić typ zmiennej (program podpowiada, jakie są możliwości) albo przekodować jej wartości (**Recode**), jeżeli np. stwierdzimy, że jakaś wartość powinna się inaczej nazywać albo że dwie wartości właściwie sprowadzają się do jednej. W górnym oknie (**Existing values**) klikamy na wartość do przekodowania, a w pasku **Recode as** wpisujemy wartość, którą ma zostać zastąpiona, albo wybieramy ją z listy (jeżeli to jedna z już istniejących wartości), i klikamy **Add**. W dolnym oknie powinna się pokazać

formuła transformacji, np. **aaa** → **bbb** (wszystkie wartości „aaa” zostaną przekodowane na „bbb”). Żeby potwierdzić, klikamy **OK**.

6. Operacje na kodach

UWAGA: wszystkie omówione działania (oprócz samego kodowania) można też wykonać z górnego menu, z zakładki **Codes**.

6.1. Kodowanie materiału

Żeby zakodować fragment materiału, zaznaczamy go myszą, touchpadem albo klawiszami kierunku z wciśniętym SHIFT, a potem w **dolnym lewym oknie (CODES)** znajdujemy odpowiedni kod i klikamy na niego dwukrotnie. Są inne sposoby, ale ten jest najprostszy i najszybszy.

Nazwa kodu, którym zakodowaliśmy ten fragment, powinna się wyświetlić w **prawym oknie** na tej samej wysokości, po nawiasie. Po najejchaniu na nią kursorem wyświetli nam się informacja o miejscu kodu w strukturze kodów, dacie kodowania i koderze.

Możemy też wyróżnić zakodowane fragmenty, wybierając z górnego menu **Document** → **Coded tekst** i którąś z opcji wyróżniania – np. **Code colors** wyróżnia te fragmenty tłem w kolorze wybranym dla danego kodu.

Żeby cofnąć operację kodowania, klikamy prawym klawiszem myszy na nazwie kodu w **prawym oknie** i wybieramy **Remove coding**. W tym samym menu jest opcja **Resize**, którą możemy zmienić zakres zakodowanego tym konkretnym kodem materiału, ale przy większych zmianach często szybciej wychodzi cofnięcie kodowania i zakodowanie od nowa.

6.2. Przekodowywanie

Jeżeli chcemy zmienić kod, którym zakodowaliśmy dany fragment materiału, możemy po prostu cofnąć kodowanie i zakodować ten fragment jeszcze raz albo kliknąć prawym klawiszem myszy na nazwę kodu w prawym oknie i wybrać **Recode to**. Pokaże się drzewo kodów, z którego wybieramy nowy, właściwy kod.

Natomiast jeżeli chcemy przekodować od razu więcej fragmentów, bo np. stwierdziliśmy, że część fragmentów zakodowanych jednym kodem bardziej pasuje do innego, to służy do tego opcja **Search & Replace** rozwijająca się po naciśnięciu nazwy kodu prawym klawiszem myszy. W pasku **Find code** wybieramy kod wyjściowy i klikamy **Find next**. W głównym oknie programu podświetli się pierwszy fragment zakodowany tym kodem – jeżeli nie chcemy zmieniać jego zakodowania, to klikamy znowu **Find next**, a jeżeli chcemy, to w pasku **Replace with** wybieramy nowy kod i klikamy **Replace**. Kliknięcie **Replace all**

spowoduje, że wszystkie fragmenty zakodowane pierwszym kodem zostaną automatycznie przekodowane na drugi.

UWAGA: tej operacji nie można cofnąć – można tylko wrócić do ostatniej zapisanej wersji pliku.

6.3. Dodawanie kodów

Klikamy prawym klawiszem myszy w oknie **CODES** na nazwę kategorii, w której chcemy utworzyć kod, i wybieramy **Add code**. Wpisujemy nazwę kodu (**Code name**), a w drzewie rozwijanym pod **Under** sprawdzamy, czy jest w odpowiednim miejscu w strukturze kodów. Możemy też określić jego kolor, a niżej podać opis (**Description**) i powiązane z nim słowa kluczowe. Aby utworzyć kod, klikamy **Add**. Potem możemy w tym samym oknie utworzyć nowy kod (domyślnie będzie miał przypisaną tę samą kategorię, co poprzedni).

6.4. Usuwanie kodów

Klikamy prawym klawiszem myszy na nazwę kodu w oknie **CODES** i wybieramy **Delete code**, a potem potwierdzamy **Yes**.

UWAGA: tej operacji nie można cofnąć – można tylko wrócić do ostatniej zapisanej wersji pliku.

6.5. Edytowanie kodów

Klikamy prawym klawiszem myszy na nazwę kodu w oknie **CODES** i wybieramy **Edit code**. Otworzy się okno takie samo jak przy dodawaniu – można w nim zmienić nazwę, miejsce w strukturze kodów, opis i kolor.

6.6. Przesuwanie kodów w strukturze

Jeżeli chcemy przenieść kod z jednej kategorii do innej albo zmienić jego miejsce w obrębie kategorii, to najłatwiej po prostu **przeciągnąć go myszą albo touchpadem**. Jeżeli przeciągniemy go (tzn. niebieski pasek, który go zaznacza) na inny kod, to pojawi się przed nim, a jeżeli na nazwę kategorii, to zostanie do niej przeniesiony i automatycznie umieszczony na ostatnim miejscu wśród kodów tej kategorii.

Trochę inaczej jest z kategoriami: przeciągnięcie A na B **nie spowoduje**, że A stanie się podkategorią B, tylko że zostanie jej równorzędną kategorią.

Jeżeli mamy rozbudowaną strukturę kodów (jak w naszym projekcie), a miejsce, w które chcemy przenieść dany kod, nie jest „tuż obok”, to lepiej skorzystać z opcji **Edit** (zob. 5.4.) i zmienić przyporządkowanie kodu do kategorii w oknie edycji.

6.7. Łączenie kodów

Jeżeli chcemy połączyć kod A z kodem B, klikamy prawym klawiszem myszy na nazwę kodu w oknie **CODES** i wybieramy **Merge code**. Rozwija się okno ze strukturą kodów i z niej wybieramy kod B, czyli ten, w który chcemy „wtopić” kod A. W następnym oknie klikamy **OK**. Wszystkie fragmenty zakodowane kodem A zostają podpięte pod kod B, a kod A znika ze struktury kodów.

6.8. Dzielenie kodów

Jeżeli stwierdzimy, że kod A jest zbyt zróżnicowany wewnątrz, i chcemy podzielić go na nowe kody B,C i D, klikamy prawym klawiszem na nazwę kodu A w oknie **CODES** i wybieramy **Split code**. W oknie **Code splitter** wpisujemy nazwy nowych kodów (każdy w osobnej linii), a pod **Store new codes** określamy, czy kod A ma się zmienić w kategorię, do której wejdą B,C i D (**Under current code...**), czy też B,C i D wejdą bezpośrednio do kategorii, do której wcześniej należał A (**Under same parent**). Potem wybieramy metodę zastępowania (**Replacement method**). Domyślna to **Search and ask for replacement**: program będzie wyświetlał kolejne fragmenty zakodowane do tej pory kodem A i pytał nas, którym z nowych kodów chcemy je zakodować. Jeżeli wybierzemy **Assign all to first code**, program automatycznie przypisze wszystkie te fragmenty do pierwszego z nowych kodów.

7. Wybrane możliwości analizy zakodowanego materiału

Przedstawiam tu tylko trzy narzędzia, które okazały się najbardziej przydatne w naszych warmińsko-mazurskich badaniach dokumentów samorządowych. Ale tych możliwości jest o wiele więcej i zachęcam Państwa do ich odkrywania na własną rękę albo z pomocą przewodnika producenta programu.

7.1. Przeglądanie zakodowanych fragmentów

To właściwie nie opcja analizy, tylko narzędzie ułatwiające wyselekcjonowanie materiału do analizy. Jeżeli chcemy przejrzeć wszystkie fragmenty zakodowane jednym albo kilkoma wybranymi kodami, w górnym menu wybieramy **Retrieval** → **Coding Retrieval**, a potem w pasku **Codes** rozwijamy hierarchię kodów (najlepiej ikonką z „trzema kulkami”) i wybieramy z niej kody, które chcemy przejrzeć (jeżeli zaznaczymy kwadracik danej kategorii, zaznaczą się automatycznie wszystkie kody w tej kategorii).

Jeżeli zaznaczymy kwadracik pod **If** w **Conditions**, będziemy mogli zawęzić wyszukiwanie tylko do tych przypadków, w których wyszukiwane przez nas fragmenty spełniają albo nie (**is/is not**) któryś z warunków: pokrywają się z fragmentami zakodowanymi innym, wybranym kodem (**equal to**), zawierają je w sobie (**enclosing**), zawierają się w nich (**included in**), zachodzą na nie (**overlapping**), poprzedzają je (**followed by**), występują po nich (**preceded by**) albo pojawiają się blisko nich (**near**).

Jeżeli dodatkowo chcemy, żeby w przeglądzie obok tekstu fragmentów wyświetlały się wartości zmiennych dla tego przypadku, to wybieramy zmienne na dole, w pasku **Add Variables**.

Po kliknięciu **Search** otwiera się okno **Search Hits** z listą zakodowanych fragmentów opisanych kategorią, kodem, numerem przypadku itd. Lista domyślnie jest pogrupowana według przypadków, ale można to zmienić w lewym górnym rogu poleceniem **Group by**, na szeregowanie według kategorii, kodów albo koderów. Jeżeli chcemy zobaczyć dany fragment w kontekście, to **klikamy na linijkę z nim i minimalizujemy okno Search Hits** – program automatycznie przejdzie do odpowiedniego przypadku i podświetli w głównym oknie wyszukiwany fragment.

7.2. Analiza częstości występowania kodów

Najbardziej podstawowe narzędzie ilościowe pozwalające nam zorientować się w częstości występowania poszczególnych kodów. W górnym menu klikamy **Analyze** → **Coding Frequency**. Zaznaczamy, czy chcemy dostać dane dotyczące wszystkich kodów (**All**), czy tylko wybranych (**Selected**) – jeżeli to drugie, to wybieramy je z „drzewka” („trzy kulki”). Ikona obok „trzech kulek” pozwala nam ograniczyć zliczanie do częściej występujących kodów (takich, które występują co najmniej n razy albo co najmniej w n przypadkach). Po kliknięciu **Search** dostajemy tabelę z zestawieniem wybranych kodów, ich licznosci bezwzględnej i procentowej oraz liczby i odsetka przypadków, w których występują.

Jeżeli chcemy zobaczyć te wyniki w formie graficznej, to musimy zaznaczyć wybrane wiersze tabeli wyników (przejechać je myszą – wybieranie punktowe ze wciśniętym SHIFT niestety nie działa) i kliknąć w **ikonę wykresu z prawej górnej strony okna**. Pokażą się różne opcje formatowania wykresu, ale osobiście radzę wyeksportować wyniki z tabeli do Excela czy SPSS (**ikona dyskietki** w tym samym pasku, co ikona wykresu) i zrobić wykresy tam albo w innym programie, bo te z QDA Minera są po prostu brzydkie.

7.3. Analiza częstości występowania kodów według zmiennych

To najczęściej używane przez nas narzędzie, które dało chyba najlepsze i najciekawsze wyniki. Pozwala ono zliczyć częstość występowania danego kodu lub kodów w powiązaniu z wartościami pewnej zmiennej. Możemy na przykład chcieć sprawdzić, czy w strategiach gmin miejskich planuje się modernizację infrastruktury częściej niż w strategiach gmin miejsko-wiejskich i wiejskich.

Klikamy **Analyze** → **Coding by Variables**, wybieramy kody do analizy, a pod spodem (**Tabulate with**) zmienną, z którą chcemy je skrzyżować. W ramce **Table content** wybieramy ustawienia tabeli – czy wyniki będą dotyczyły licznosci kodu (**Code frequency**), liczby przypadków, w których występuje (**Code occurrence (cases)**), liczby zakodowanych słów (**Word count**) albo ich odsetka w dokumentach o danej wartości zmiennej (**Percentage**

of words). W pasku **Display** wybieramy, czy wyniki będą podawane w liczbach bezwzględnych (**Count**), czy też będą stanowić procent wyników w wierszu (**Row percent**), kolumnie (**Column percent**) albo całej tabeli (**Total percent**). Oprócz tego możemy dodać do tabeli wyniki analiz statystycznych. Żeby wygenerować tabelę, klikamy **Search**.

Podobnie jak w wypadku samej analizy częstości (zob. 6.2.), możemy zobaczyć wyniki w postaci graficznej jako jeden z kilku typów wykresu (grupa **ikon wykresu z prawej górnej strony**) albo wyeksportować je do Excela czy SPSS (**ikona dyskietki**).

8. Omówienie listy zmiennych w badaniu „Mapa wiedzy i niewiedzy”

W „Mapie wiedzy i niewiedzy” używamy stałej listy zmiennych opisujących kodowane dokumenty jednostek samorządu terytorialnego (JST). Ale gdyby ktoś z Państwa wpadł na pomysł nowych zmiennych, to podzielcie się nimi Państwo na grupie facebookowej.

Nasze zmienne są następujące:

STRATEGIA – nazwa kodowanego dokumentu JST.

POZIOM_STR – poziom JST, której dotyczy dokument (**powiat – gmina – miejscowość**).

POWIAT – nazwa powiatu, do którego należy dana JST (albo powiatu, którego dotyczy dokument).

GMINA – nazwa gminy, której dotyczy dokument (albo – w przypadku strategii miejscowości na poziomie sołectw – do której należy dana miejscowość).

TYPGMINY – typ gminy, której dotyczy dokument (**miejska – miejsko/wiejska – wiejska**).

MIEJSCOWOSC – nazwa miejscowości na poziomie sołectwa, której dotyczy dokument.

DOKUMENT – zmienna tekstowa, konieczna do załączania dokumentów.

Wszystkie zmienne będą określone tylko w wypadku dokumentów miejscowości na poziomie sołectw (jeżeli takie będą). W wypadku strategii gminnych pusta zostaje zmienna MIEJSCOWOSC, a strategii powiatowych – zmienne GMINA, TYPGMINY i MIEJSCOWOSC.

UWAGA: niektóre gminy miejskie i wiejskie nazywają się tak samo, trzeba sprawdzać, czy na pewno kodujemy dobry typ gminy.

9. Omówienie struktury kodów w badaniu „Mapa wiedzy i niewiedzy”

W projekcie mamy dokładnie 328 kodów i niestety trzeba brać wszystkie pod uwagę, chociaż nie wszystkie mają jednakową wagę badawczą. Nie mogę tu oczywiście omówić wszystkich,

dlatego przybliżę zakres poszczególnych kategorii i podkategorii, a w ich ramach niektóre ważne albo problematyczne kody.

INFRASTRUKTURA – tu kodujemy informacje o materialnej i instytucjonalnej infrastrukturze kultury. Ta kategoria obejmuje dwie podkategorie:

INSTYTUCJE I PLACÓWKI KULTURALNE – kodujemy **występowanie** określonych instytucji kultury (domów kultury, bibliotek, świetlic itd.) – jedną instytucję wystarczy zakodować **raz na przypadek**, później osobno możemy już kodować jej cele i aktywność (zob. dalej i 11. – czwarta porada).

MATERIALNA INFRASTRUKTURA KULTURY – informacje o zabytkach oraz wielkości, stanie materialnym i wyposażeniu placówek kulturalnych.

KADRY KULTURY – informacje o aktorach w „polu kulturalnym” – najczęściej będą to pracownicy instytucji kultury. Kodujemy zarówno wzmianki o ogólnej sytuacji kadrowej („Dom kultury zatrudnia cztery osoby”), jak i o konkretnych osobach. Kody z tej kategorii mogą się nakładać na kody z kategorii **AKTYWNOŚĆ** (np. kiedy mowa o tym, co robi lokalny twórca).

FINANSE – wszystkie wzmianki o budżetach, dotacjach i wydatkach na działalność kulturalną (często pojawiają się w ramach tabelek opisujących dotowane projekty). Tu kodujemy tylko konkretne dane – ogólne oceny nakładów na kulturę („wysokie”, „za niskie”) podpadają pod **ANALIZĘ MOCNYCH I SŁABYCH STRON**.

UCZESTNICZY KULTURY – informacje o tym, kto uczestniczył w działaniach kulturalnych lub do kogo mają one zostać skierowane w przyszłości. Kodujemy zarówno ogólne informacje („Świetlica będzie przeznaczona dla dzieci”), jak i szczegółowe zestawienia, np. liczbę czytelników biblioteki albo uczestników zajęć w domu kultury.

CELE INSTYTUCJI – KOGO? – kiedy mowa o planach działalności kulturalnej, realizacji projektów kulturalnych czy modernizacji infrastruktury kulturalnej, kodujemy, kto jest podmiotem stawiającym sobie te cele. Kiedy celów jest więcej i nie zostały wymienione obok siebie, ten sam podmiot kodujemy kilka razy – ale jeżeli są obok siebie, obejmujemy je jednym kodem podmiotowym.

CELE INSTYTUCJI – JAKIE? – tu kodujemy treść planów działalności kulturalnej. Kody należące do tej kategorii pojawiają się **najczęściej ze wszystkich**, więc warto zwrócić uwagę na ich precyzyjne rozróżnienie. Do tej kategorii należą następujące podkategorie:

OGÓLNE – najszersze możliwe cele działalności kulturalnej, często tak szerokie, że trudno powiedzieć, co właściwie oznaczają. Zazwyczaj pojawiają się pod postacią gotowych formułek typu **organizacja czasu wolnego, stymulacja uczestnictwa w kulturze** itp.

ORGANIZACYJNE I MATERIALNE – cele „wewnętrzne” instytucji kultury, związane z ich bazą materialną, strukturą i relacjami z tzw. otoczeniem (ale nie uczestnikami kultury, tylko innymi aktorami z pola kultury, najczęściej instytucjami).

EDUKACYJNE I INFORMACYJNE – nazwa mówi wszystko. Kody w tej kategorii są dosyć precyzyjne, warto tylko zwrócić uwagę na różnicę między **edukacją** a **edukacją kulturową** – ta pierwsza to np. szkolenia zawodowe organizowane w domach kultury, a ta druga – np. lekcje biblioteczne czy muzealne.

PROGRAMOWE I KOMERCYJNE – obejmuje większość działań zwróconych bezpośrednio do uczestników kultury. Najważniejsze jest tu rozróżnienie **organizacji imprez kulturalnych**, planów działań typu „eventowego” czy „punktowego”, takich jak koncert czy festyn, od **tworzenia usług kulturalnych**, czyli stałych albo przynajmniej okresowych zajęć, warsztatów itp. Pamiętajmy, że tutaj kodujemy tylko działania **planowane** – już (z)realizowane podpadają pod **AKTYWNOŚĆ KULTURALNĄ**.

DŁUGOFALOWE CELE KULTURY – ta kategoria okazała się bardzo płodna w badaniach na Warmii i Mazurach. Obejmuje ona wszystkie sytuacje, kiedy w strategiach pisze się o działalności kulturalnej jako o narzędziu osiągnięcia celów sytuujących się już poza dziedziną samej kultury. Ta kategoria obejmuje dwie podkategorie:

EKONOMICZNE – kultura i działalność kulturalna jako dźwignia handlu, magnes dla turystów itp.

SPOŁECZNE – kultura i działalność kulturowa jako czynnik współtworzący tożsamość, poprawiający jakość życia i relacje międzyludzkie itp.

AKTYWNOŚĆ KULTURALNA – KOGO? – kodami z tej kategorii kodujemy informacje o tym, jaki podmiot **zrealizował albo właśnie realizuje** dane działanie kulturalne. Trzeba tu pamiętać o odróżnieniu celów czy planów od rzeczywistej działalności. W tej kategorii mieści się wydzielona podkategoria:

AKTYWNOŚĆ INSTYTUCJI KULTURY – działania podejmowane przez określone typy placówek kulturalnych.

AKTYWNOŚĆ KULTURALNA – JAKA? - kody identyfikujące rodzaje podejmowanych działań kulturalnych, często analogiczne do kodów dotyczących celów. Do tej kategorii należą następujące podkategorie:

WYDARZENIA – wszystkie działania typu „improwego” czy „eventowego”. Kody są tu bardzo szczegółowe, ale czasem trudno stwierdzić na podstawie dokumentu, do którego pasuje dane wydarzenie – wtedy warto skorzystać z kodu „inne”. Kod

„impresy ogółem” dotyczy tylko sytuacji, kiedy w strategii jest mowa o zupełnie niesprecyzowanych imprezach kulturalnych („Dom kultury organizuje wiele imprez”).

USŁUGI KULTURALNE – wszystkie formy działalności, które mają stały, okresowy albo długoterminowy charakter – zajęcia, warsztaty itp. Kod „działalność komercyjna” dotyczy np. organizowania dyskotek w domu kultury.

DZIAŁANIA MEDIALNE I INFORMACYJNE – przede wszystkim różne formy informowania o działaniach danego podmiotu za pośrednictwem mediów (też prowadzenie strony internetowej).

DZIAŁANIA WEWNĘTRZNE – odpowiednik CELÓW ORGANIZACYJNYCH I MATERIALNYCH, tylko że po stronie „zrealizowanej”.

NGO – wszystkie wzmianki o organizacjach pozarządowych, nie tylko tych lokalnych. UWAGA: jeżeli w dokumencie jest mowa nie tylko o istnieniu danej organizacji, ale też o jej działalności, kodujemy ją dodatkowo kodem AKTYWNOŚĆ – KOGO? → aktywność NGO.

LIDERZY – wszystkie wzmianki o konkretnych ludziach, którzy wybijają się jako znaczący aktorzy działalności kulturalnej. Może to być np. operatywny wójt (jeżeli jego operatywność dotyczy działalności kulturowej), czyli **lider samorządowy**, albo zasłużona bibliotekarka, czyli **lokalny lider** (różnica jest bliżej wyjaśniona w opisie kodów).

POTRZEBY – wszystkie wzmianki o tym, że jakaś grupa osób, instytucja albo budynek „czegoś potrzebuje” albo „czegoś mu/jej brakuje”, oczywiście w kontekście kultury („Niepełnosprawni potrzebują aktywizacji kulturalnej”, „Świetlica wymaga remontu” itp.).

ANALIZA MOCNYCH I SŁABYCH STRON – wszystkie fragmenty, w których autorzy dokumentu przedstawiają zasoby, infrastrukturę albo ofertę kulturalną jako mocny lub słaby punkt JST albo coś, co może mieć wpływ na jej rozwój. Te oceny najczęściej pojawiają się w ramach analizy SWAT, ale używamy tych kodów również wtedy, kiedy pojawiają się poza tym kontekstem. Ta kategoria obejmuje cztery podkategorie:

MOCNE STRONY – konkretne aspekty kultury wskazuje się jako mocną stronę JST.

SŁABE STRONY – konkretne aspekty kultury wskazuje się jako słabą stronę JST.

SZANSE – konkretne aspekty kultury wskazuje się jako coś, co może sprzyjać rozwojowi JST.

ZAGROŻENIA – konkretne aspekty kultury wskazuje się jako coś, co może zagrozić rozwojowi JST.

SZERSZE UWARUNKOWANIA DZIAŁALNOŚCI KULTURALNEJ – wszystkie wzmianki o tym, że działania kulturalne danego podmiotu są uzależnione od czynników o charakterze ponadlokalnym – ram prawnych, strategii wojewódzkich i państwowych, relacji międzynarodowych itp.

BADANIA – wszystkie miejsca, w których autorzy strategii przytaczają badania, ekspertyzy, diagnozy lub konsultacje społeczne dotyczące lokalnej działalności kulturalnej.

WSKAŹNIKI – wszystkie miejsca, gdzie autorzy strategii wskazują, na podstawie czego zostanie oceniony stopień realizacji celów działalności kulturalnej. Takie wskaźniki najczęściej pojawiają się w tabelach opisujących dotowane projekty, jako „Wskaźniki realizacji celu”.

10. Kilka praktycznych porad dotyczących kodowania

Porada pierwsza, praktyczna – przed przystąpieniem do kodowania warto **zwinąć strukturę kodów**, tak żeby w lewym oknie były widoczne tylko główne kategorie, a potem w miarę potrzeb rozwijać kategorie, z których potrzebujemy kodów. Inaczej samo przewijanie kodów zajmuje więcej czasu niż kodowanie. Zwinąć strukturę kodów można tak: prawym klawiszem myszy klikamy na dowolny kod (albo w górnym pasku zakładka **Codes**) i z rozwiniętego menu wybieramy **Collapse All**. Jak łatwo się domyślić, **Expand All** rozwija strukturę do pełnej postaci.

UWAGA: lepiej nie dotykać opcji **Sort Codebook** – porządkuje ona strukturę kodów w porządku alfabetycznym, a to wbrew pozorom utrudnia ich odnajdywanie.

Porada druga, też praktyczna – okna CASES, VARIABLES i CODES na początku mają jednakową wysokość, ale warto zmienić ich rozmiar, najeżdżając myszą na granice między nimi i przeciągając je. Okno CODES powinno być jak najwyższe, żebyśmy mieli dostęp do jak największej liczby kodów, dlatego okno CASES warto zmniejszyć do dwóch-trzech przypadków, a okno VARIABLES zwinąć do minimum i rozwijać tylko podczas określania zmiennych (albo sprawdzania, czy są prawidłowo określone).

Trzecia porada dotyczy problemu zakresu kodowania – czy lepiej zakodować to, do czego ściśle odnosi się dany kod, czy może całe zdanie? W dużej mierze zależy to od kodu – tymi z kategorii INFRASTRUKTURA można spokojnie zakodować tylko nazwę instytucji („Gminny Ośrodek Kultury”), nie trzeba obejmować kodem całego zdania („W gminie mamy sławny na całe województwo Gminny Ośrodek Kultury”). Podobnie z nazwami imprez z kategorii AKTYWNOŚĆ KULTURALNA – JAKA? → WYDARZENIA. Ale już np. kodami z kategorii AKTYWNOŚĆ KULTURALNA – KOGO? często można kodować cały akapit (albo i kilka), jeżeli mowa w nim cały czas o działalności jednej placówki czy podmiotu. Większość kodów będzie mieścić się gdzieś pomiędzy tymi biegunami.

W razie wątpliwości zawsze lepiej zakodować trochę więcej, bo to nie wpływa na większość analiz ilościowych (oprócz tych dotyczących liczby słów), a w niektórych wypadkach może ułatwić analizę jakościową. W większości wypadków wystarczy pomyśleć, co powiedziałby nam zakodowany fragment, gdybyśmy prowadzili analizę jakościową za pomocą opcji **Coding Retrieval** (zob. 6.1.), a nie chcielibyśmy za każdym razem sprawdzać kontekstu w głównym oknie. Na przykład kodując wydatki na kulturę, oprócz samej kwoty powinniśmy spróbować objąć kodem również przedmiot wydatków (jeżeli to możliwe), bo w przeciwnym razie zostaniemy z gołą kwotą, która niewiele nam nie będzie mówić.

Czwarta porada dotyczy kwestii jednokrotnego lub wielokrotnego kodowania tych samych elementów (w jednym przypadku). Zasadniczo większość analiz prowadziliśmy do tej pory na poziomie występowania/niewystępowania kodu w przypadku (**Code occurrence (cases)** w opcji **Coding by Variables**), a wtedy po pierwszym wystąpieniu danego elementu następne już nie są brane pod uwagę. Ale tutaj znowu wszystko zależy od kategorii: w niektórych (np. INFRASTRUKTURA), jeżeli dany element pojawi się raz, to potem już go nie kodujemy (przynajmniej nie tym samym kodem). Czyli: we fragmencie poświęconym gminnemu ośrodkowi kultury kodujemy raz jego nazwę kodem INFRASTRUKTURA → INSTYTUCJE I PLACÓWKI KULTURALNE → domy/centra kultury, a potem ewentualnie używamy innych kodów – dotyczących celów tej instytucji, jej działalności, wydatków itp.

Ogólna zasada jest taka: jeżeli w przypadku powtarza się **dokładnie ten sam element (czyli konkretna instytucja, podmiot, wydarzenie itd.)**, to nie kodujemy go powtórnie, chyba że pojawia się w innym kontekście, który coś wnosi w porównaniu z poprzednim wystąpieniem. Na przykład jeżeli w dokumencie jest mowa o konkretnym festiwalu organizowanym (nie tylko planowanym) przez gminny ośrodek kultury, to kodujemy go jako:

AKTYWNOŚĆ KULTURALNA – KOGO? → AKTYWNOŚĆ INSTYTUCJI KULTURY
→ aktywność domów kultury
AKTYWNOŚĆ KULTURALNA – JAKA? → WYDARZENIA → festiwale,

ale jeżeli w innym miejscu strategii jest mowa o tym samym festiwalu, to nie kodujemy go już tymi kodami. Możemy za to innymi, jeżeli pojawiają się nowe informacje – np. dotyczące wydatków czy uczestników.

Jeżeli chodzi o same kody, to oczywiście **mogą się powtarzać**, nawet w połączeniu z tym samym elementem. Na przykład jeżeli ten sam podmiot ma kilka celów działalności kulturalnej i nie są one wymienione obok siebie (a więc nie można ich zakodować łącznie), to każdy z nich kodujemy osobno tym samym kodem, np. CELE INSTYTUCJI – KOGO? → cele bibliotek.

W wypadku **niektórych kategorii i kodów** przyjęliśmy, że **można zakodować łącznie kilka równorzędnych elementów**. Najważniejsze i już wymienione wypadki to kategorie CELE INSTYTUCJI – KOGO? i AKTYWNOŚĆ KULTURALNA – KOGO?, w których jednym

kodek możemy objąć kilka celów albo działań tego samego podmiotu. Podobnie **spis zabytków** w gminie obejmujemy jednym kodek INFRASTRUKTURA → zabytki, nie kodujemy każdego zabytku osobno. Ale już **wydarzenia kulturalne kodujemy każde z osobna**, nawet jeżeli podpadają pod ten sam kod, a wzmianki o nich są bezpośrednio obok siebie. A więc trzy różne festiwale wymienione w jednym zdaniu zakodujemy trzy razy tym samym kodek. Podobnie w kategorii FINANSE – jeżeli jest mowa o trzech różnych projektach kulturalnych finansowanych przez gminę, to każdy zakodujemy osobno kodek FINANSE → wydatki gminy na kulturę. W pozostałych wypadkach trzymamy się raczej zasady ekonomii i kodujemy równorzędne elementy łącznie, zwłaszcza jeżeli występują obok siebie.