

# Agregacje

Wszystkie agregacje znajdują się w osobnych plikach i są gotowe do uruchomienia, schemat nazewnictwa:

aggregate{nr agregacji w tym pliku}.js

## 1. Regiony świata, w których najczęściej dochodzi do zamachów (w ostatniej dekadzie).

Usuujemy “nieznane” regiony:

```
var match = {
  $match: {
    region: {
      $ne: "Unknown"
    }
  }
};
```

Grupujemy po id regionu i zapamiętujemy jego nazwę, oraz zliczamy zamachy:

```
var group = {
  $group: {
    _id: "$region",
    region: {
      $first: "$region_txt"
    },
    count: {
      $sum: 1
    }
  }
};
```

Sortujemy po ilości zamachów:

```
var sort = {
  $sort: {
    count: -1
  }
};
```

Usuujemy z wyniku niepotrzebne nam pole “\_id”:

```
var project = {
  $project: {
    _id: 0
  }
};
```

Składamy zapytanie w całość:

```
var aggregate1a = db.terrorism.aggregate([match, group, sort, project]);
```

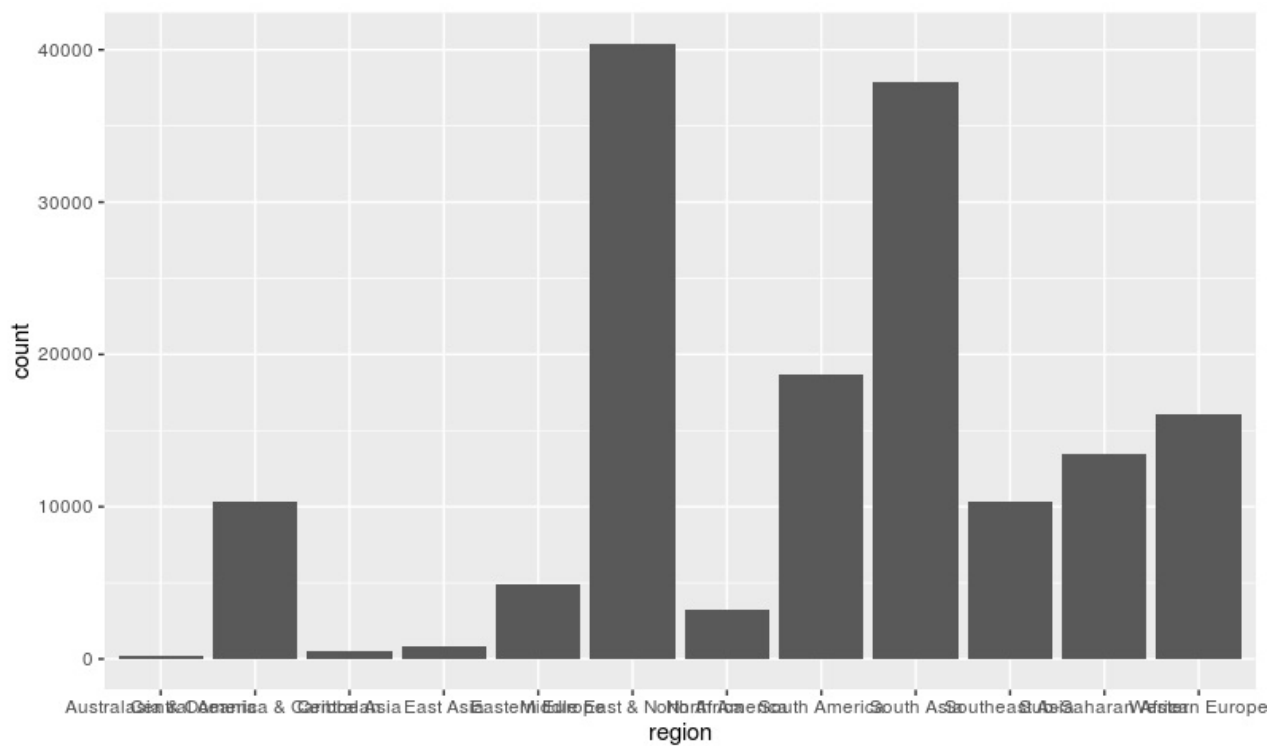
Otrzymaliśmy ranking regionów świata najbardziej dotkniętych zamachami terrorystycznymi od roku 1970. Aby otrzymać dane z ostatniej dekady musimy zrobić małą zmianę (zaznaczam, że dane sięgają początku roku 2016):

```
var match2 = {
  $match: {
    region: {
      $ne: "Unknown"
    },
    iyear: {
      $gte: 2005
    }
  }
};

var aggregate1b = db.terrorism.aggregate([match2, group, sort, project], {
  explain: false
}).pretty();
```

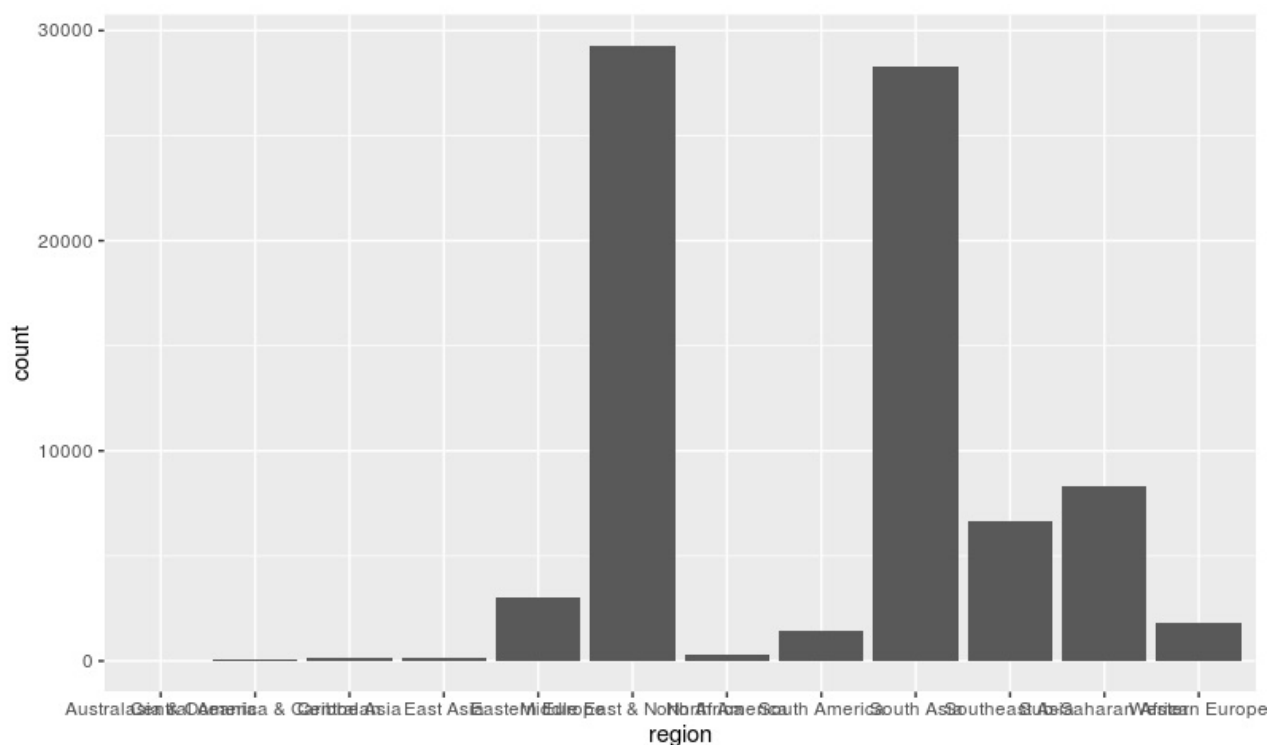
Wynik pierwszego zapytania:

```
{ "region" : "Middle East & North Africa", "count" : 40422 }
{ "region" : "South Asia", "count" : 37841 }
{ "region" : "South America", "count" : 18628 }
{ "region" : "Western Europe", "count" : 16020 }
{ "region" : "Sub-Saharan Africa", "count" : 13434 }
{ "region" : "Southeast Asia", "count" : 10360 }
{ "region" : "Central America & Caribbean", "count" : 10337 }
{ "region" : "Eastern Europe", "count" : 4892 }
{ "region" : "North America", "count" : 3268 }
{ "region" : "East Asia", "count" : 786 }
{ "region" : "Central Asia", "count" : 538 }
{ "region" : "Australasia & Oceania", "count" : 246 }
```



Oraz drugiego:

```
{ "region" : "Middle East & North Africa", "count" : 29297 }
{ "region" : "South Asia", "count" : 28288 }
{ "region" : "Sub-Saharan Africa", "count" : 8348 }
{ "region" : "Southeast Asia", "count" : 6649 }
{ "region" : "Eastern Europe", "count" : 3019 }
{ "region" : "Western Europe", "count" : 1816 }
{ "region" : "South America", "count" : 1471 }
{ "region" : "North America", "count" : 299 }
{ "region" : "Central Asia", "count" : 143 }
{ "region" : "East Asia", "count" : 129 }
{ "region" : "Central America & Caribbean", "count" : 44 }
{ "region" : "Australasia & Oceania", "count" : 30 }
```



Wnioski:

Wciąż najwięcej aktów terroryzmu ma miejsce na Bliskim Wschodzie, w Północnej Afryce i Południowej Azji. Niepokojąco wzrasta również pozycja Europy.

## 2. Średnia długość przetrzymywania zakładników (w dniach).

Oczywiście chodzi tutaj głównie o porwania.

Najpierw wybieramy te zdarzenia, w których doszło do porwania. Tutaj mamy dwie możliwości: możemy uwzględnić te, których długość trwała 0 dni lub nie. Oczywiście zmieni nam to odrobinę średnią.

Porwania, których długość to 0 dni to porwania krótkie, których czas liczony jest raczej w godzinach. Dla uwzględniania tego faktu postanowiłem ująć takie porwania w średniej:

```
var match = { $match: {
  ishostkid: {
    $eq: 1
  },
  ndays: {
    $gte: 0
  }
}
};
```

Nasze grupowanie sprowadza się do wyliczenia średniej ze wszystkich pasujących rekordów:

```
var group = {
  $group: {
    _id: null,
```

```
    avg: {
      $avg: "$ndays"
    }
  }
};
```

Dla czytelności możemy pominąć “\_id”:

```
var project = {
  $project: {
    _id: 0
  }
};
```

Składamy agregację w całość:

```
var aggregate2 = db.terrorism.aggregate([match, group, project], {
  explain: false
}).pretty();
```

Wynik:

```
{ "avg" : 35.87450980392157 }
```

Wnioski:

Porwania trwają średnio około miesiąca. Inną sprawą jest to czy porwani po miesiącu giną czy wracają do domów. Magia statystyki.

### 3. Polska.

Czy w Polsce zdarzały się zamachy terrorystyczne?  
Jeśli tak, to czy któryś się powiódł? Czy były ofiary?  
Sprawdźmy to!

161 to ID naszego kraju:

```
var match = {
  $match: {
    country: {
      $eq: 161
    }
  }
};
```

Grupujemy i sumujemy poszczególne wartości:

```
var group = {
  $group: {
    _id: "$country",
    "liczba zamachow": {
      $sum: 1
    }
  }
};
```

```

    },
    "udanych": {
      $sum: "$success"
    },
    "liczba ofiar smiertelnych": {
      $sum: "$nkill"
    },
    "liczba rannych": {
      $sum: "$nwound"
    }
  }
}
};

```

Składamy wszystko w całość:

```

var aggregate3 = db.terrorism.aggregate([match, group], {
  explain: false
}).pretty();

```

Wynik (widać dlaczego w poprzednich zapytaniach pomijaliśmy ID):

```

{
  "_id" : 161,
  "liczba zamachow" : 34,
  "udanych" : 30,
  "liczba ofiar smiertelnych" : 7,
  "liczba rannych" : 30
}

```

Wnioski:

Tutaj pojawia się miejsce na dyskusję p.t.: “Czym właściwie jest zamach terrorystyczny?” Odsyłam do dołączonego [PDF](#).

## 4. W jakich miastach jest najwięcej zamachów i co jest narzędziem terroru?

Narzędzie terroru nie jest może idealnym określeniem, ale czy, na przykład, samochód można nazwać “bronią”?

Na początek usuwamy “śmieci”, czyli zamachy, gdzie miasto jest nieznane:

```

var match = {
  $match: {
    city: {
      $ne: "Unknown"
    }
  }
}
};

```

Grupujemy po mieście:

```

var group = {

```

```
$group: {
  _id: "$city",
  count: {
    $sum: 1
  }
};
```

Sortujemy malejąco:

```
var sort = {
  $sort: {
    count: -1
  }
};
```

I wybieramy pierwszą, niechlubną, dziesiątkę:

```
var limit = {
  $limit: 10
};
```

Łączymy wszystko w całość:

```
var aggregate4a = db.terrorism.aggregate([match, group, sort, limit], {
  explain: false
}).pretty();
```

Wynik:

```
{ "_id" : "Baghdad", "count" : 6237 }
{ "_id" : "Karachi", "count" : 2530 }
{ "_id" : "Lima", "count" : 2358 }
{ "_id" : "Belfast", "count" : 2102 }
{ "_id" : "Santiago", "count" : 1614 }
{ "_id" : "Mosul", "count" : 1553 }
{ "_id" : "San Salvador", "count" : 1547 }
{ "_id" : "Mogadishu", "count" : 1169 }
{ "_id" : "Istanbul", "count" : 999 }
{ "_id" : "Bogota", "count" : 974 }
```

Wnioski:

Większość miast na tej liście nie powinno być zaskoczeniem. Dziwić może jedynie tak wysoka pozycja Belfastu. IRA przez lata, niestety, nie próżnowała...

Ciekawostka:

W Mongo 3.4 to zapytanie może uprościć. Zamiast osobno grupować i sortować wystarczy użyć `$sortByCount`:

```
var sortByCount = {
  $sortByCount: "$city"
};
```

```
var aggregate4b = db.terrorism.aggregate([match, sortByCount, limit], {
  explain: false
}).pretty();
```

Oczywiście wynik takiego zapytania jest dokładnie taki sam. Również szybkość działania się nie zmienia. Dlaczego? Wystarczy zmienić *explain* na *true* i okaże się, że plan wykonania jest taki sam jak w poprzednim przypadku. Tak więc *sortByCount* to jedynie tzw. “cukier syntaktyczny”.

A teraz sprawdźmy jakie “narzędzia terroru” najczęściej wykorzystywano w tych miastach:

```
var groupWithWeapons = {
  $group: {
    _id: {
      city: "$city",
      weapon: "$weaptype1_txt"
    },
    count: {
      $sum: 1
    }
  }
};

var aggregate4c = db.terrorism.aggregate([match, groupWithWeapons, sort, limit], {
  explain: false
}).pretty();
```

Wynik:

```
{
  "_id" : {
    "city" : "Baghdad",
    "weapon" : "Explosives/Bombs/Dynamite"
  },
  "count" : 5219
}
{
  "_id" : {
    "city" : "Lima",
    "weapon" : "Explosives/Bombs/Dynamite"
  },
  "count" : 1690
}
{
  "_id" : {
    "city" : "Karachi",
    "weapon" : "Firearms"
  },
  "count" : 1417
}
{
  "_id" : {
    "city" : "Santiago",
    "weapon" : "Explosives/Bombs/Dynamite"
  },
  "count" : 1170
}
```



```

{
  "_id" : {
    "city" : "Belfast",
    "weapon" : "Firearms"
  },
  "count" : 1081
}
{ "_id" : { "city" : "Baghdad", "weapon" : "Firearms" }, "count" : 963 }
{
  "_id" : {
    "city" : "Mosul",
    "weapon" : "Explosives/Bombs/Dynamite"
  },
  "count" : 930
}
{
  "_id" : {
    "city" : "San Salvador",
    "weapon" : "Explosives/Bombs/Dynamite"
  },
  "count" : 735
}
{
  "_id" : {
    "city" : "Belfast",
    "weapon" : "Explosives/Bombs/Dynamite"
  },
  "count" : 673
}
{
  "_id" : {
    "city" : "Mogadishu",
    "weapon" : "Explosives/Bombs/Dynamite"
  },
  "count" : 630
}
}

```

Wnioski:

Zdecydowanie najpopularniejsze wśród terrorystów są materiały wybuchowe.

## 5. Ile osób zginęło od danego rodzaju broni?

Poprzednia agregacja daje nam pewną odpowiedź. Ale po kolei:

Wybieramy te zamachy, w których ktoś zginął:

```

var match = {
  $match: {
    nkill: {
      $gte: 1
    }
  }
};

```

Grupujemy po typie broni i usuwamy to co po przecinku (to nie tak, że można kogoś trochę zabić, to po prostu statystyka):

```

var group = {
  $group: {
    _id: {
      bron: "$weaptype1_txt"
    },
    zabitych: {
      $sum: {
        $trunc: "$nkill"
      }
    }
  }
}
};

```

Sortujemy malejąco po liczbie ofiar i wybieramy 10 pierwszych wyników:

```

var sort = {
  $sort: {
    zabitych: -1
  }
};

var limit = {
  $limit: 10
};

```

Składamy zapytanie w całość:

```

var aggregate5 = db.terrorism.aggregate([match, group, sort, limit], {
  explain: false
}).pretty();

```

Wynik:

```

{ "_id" : { "bron" : "Firearms" }, "zabitych" : 156350 }
{ "_id" : { "bron" : "Explosives/Bombs/Dynamite" }, "zabitych" : 141581 }
{ "_id" : { "bron" : "Unknown" }, "zabitych" : 31840 }
{ "_id" : { "bron" : "Melee" }, "zabitych" : 9800 }
{ "_id" : { "bron" : "Incendiary" }, "zabitych" : 5008 }
{
  "_id" : {
    "bron" : "Vehicle (not to include vehicle-borne explosives, i.e., car or truck bombs)"
  },
  "zabitych" : 3101
}
{ "_id" : { "bron" : "Chemical" }, "zabitych" : 430 }
{ "_id" : { "bron" : "Sabotage Equipment" }, "zabitych" : 49 }
{ "_id" : { "bron" : "Other" }, "zabitych" : 46 }
{ "_id" : { "bron" : "Biological" }, "zabitych" : 9 }

```

Wnioski:

A jednak nie materiały wybuchowe. O włos wyprzedziła je broń palna. "Melee" to bijatyka.

## 6. Jakiego typu ataków było najwięcej w danym regionie

## świata.

Może da się to zrobić prościej. Ja męczyłem się długo.

Grupujemy po regionie i typie ataku:

```
var group = {
  $group: {
    "_id": {
      "region": "$region_txt",
      "typ": "$attacktype1_txt"
    },
    "countTypes": {
      "$sum": 1
    }
  }
};
```

Sortujemy po raz pierwszy:

```
var sort = {
  $sort: {
    "countTypes": -1
  }
};
```

Grupujemy kolejny raz ale teraz tworząc podobiekty odpowiadające za typy ataku:

```
var secondGroup = {
  $group: {
    "_id": "$_id.region",
    "typy": {
      "$push": {
        "typ": "$_id.typ",
        "liczba": "$countTypes"
      }
    },
    "liczba": {
      "$sum": "$countTypes"
    }
  }
};
```

Sortujemy kolejny raz:

```
var secondSort = {
  $sort: {
    "liczba": -1
  }
};
```

Chcemy tylko pierwszy obiekt z tablicy typów (już posortowanej) i "liczba":

```
var project = {
  $project: {
    "typy": {
      $slice: ["$typy", 1]
    },
    "liczba": 1
  }
};
```

Składamy agregację w całość:

```
var aggregate6 = db.terrorism.aggregate([group, sort, secondGroup, secondSort, project], {
  explain: false
}).pretty();
```

Wynik:

```
{
  "_id" : "Middle East & North Africa",
  "liczba" : 40422,
  "typy" : [
    {
      "typ" : "Bombing/Explosion",
      "liczba" : 24053
    }
  ]
}
{
  "_id" : "South Asia",
  "liczba" : 37841,
  "typy" : [
    {
      "typ" : "Bombing/Explosion",
      "liczba" : 18247
    }
  ]
}
{
  "_id" : "South America",
  "liczba" : 18628,
  "typy" : [
    {
      "typ" : "Bombing/Explosion",
      "liczba" : 8931
    }
  ]
}
{
  "_id" : "Western Europe",
  "liczba" : 16020,
  "typy" : [
    {
      "typ" : "Bombing/Explosion",
      "liczba" : 8354
    }
  ]
}
```

```
}
{
  "_id" : "Sub-Saharan Africa",
  "liczba" : 13434,
  "typy" : [
    {
      "typ" : "Armed Assault",
      "liczba" : 4671
    }
  ]
}
{
  "_id" : "Southeast Asia",
  "liczba" : 10360,
  "typy" : [
    {
      "typ" : "Bombing/Explosion",
      "liczba" : 4019
    }
  ]
}
{
  "_id" : "Central America & Caribbean",
  "liczba" : 10337,
  "typy" : [
    {
      "typ" : "Armed Assault",
      "liczba" : 4358
    }
  ]
}
{
  "_id" : "Eastern Europe",
  "liczba" : 4892,
  "typy" : [
    {
      "typ" : "Bombing/Explosion",
      "liczba" : 2665
    }
  ]
}
{
  "_id" : "North America",
  "liczba" : 3268,
  "typy" : [
    {
      "typ" : "Bombing/Explosion",
      "liczba" : 1518
    }
  ]
}
{
  "_id" : "East Asia",
  "liczba" : 786,
  "typy" : [
    {
      "typ" : "Bombing/Explosion",
      "liczba" : 325
    }
  ]
}
```

```

}
{
  "_id" : "Central Asia",
  "liczba" : 538,
  "typy" : [
    {
      "typ" : "Bombing/Explosion",
      "liczba" : 224
    }
  ]
}
{
  "_id" : "Australasia & Oceania",
  "liczba" : 246,
  "typy" : [
    {
      "typ" : "Bombing/Explosion",
      "liczba" : 72
    }
  ]
}
}

```

Wnioski:

Prawie wszędzie materiały wybuchowe wygrywają.

## 7. Atak w którym było najwięcej rannych i zabitych.

Wybieramy te ataki, które mają jakichś rannych i zabitych:

```

var match = {
  $match: {
    nkill: {
      $gte: 0
    },
    nwound: {
      $gte: 0
    }
  }
}
};

```

Dodajemy liczbę rannych i zabitych:

```

var project = {
  $project: {
    "liczba ofiar": {
      $add: ["$nkill", "$nwound"]
    }
  }
}
};

```

Wybieramy największą wartość:

```

var sort = {
  $sort: {

```

```

    "liczba ofiar": -1
  }
};

var limit = {
  $limit: 1
};

```

Składamy wszystko w całość:

```

var aggregate7a = db.terrorism.aggregate([match, project, sort, limit], {
  explain: false
})

```

Wynik:

```

{ "_id" : ObjectId("591ead90043ea5be6ffcd6f3"), "liczba ofiar" : 5513 }

```

Wnioski:

Wynik jest dobry, ale zapytanie złe. W pierwszej fazie użyliśmy logicznego AND zamiast OR. Tego drugiego nie mogliśmy użyć, bo kolumna przyjmuje również wartości nieliczbowe i w kolejnych etapach agregacja by się “wykrzaczyła”. Możemy to obejść używając na przykład instrukcji *\$switch*:

```

var match2 = {
  $match: {
    $or: [{
      nkill: {
        $gte: 0
      }
    }, {
      nwound: {
        $gte: 0
      }
    }]
  }
};

var projectSwitch = {
  $project: {
    "liczba ofiar": {
      $switch: {
        branches: [{
          case: {
            $and: [{
              $ne: ["$nkill", ""]
            },
            {
              $ne: ["$nwound", ""]
            }
          ]
        },
        then: {
          $add: ["$nkill", "$nwound"]
        }
      }
    }
  }
};

```

```

    }
  },
  {
    case: {
      $and: [{
        $ne: ["$nkill", ""]
      },
      {
        $eq: ["$nwound", ""]
      }
    ]
  },
  then: "$nkill"
},
{
  case: {
    $and: [{
      $eq: ["$nkill", ""]
    },
    {
      $ne: ["$nwound", ""]
    }
  ]
},
  then: "$nwound"
}
],
default: 0
}
}
}
};

var aggregate7b = db.terrorism.aggregate([match2, projectSwitch, sort, limit], {
  explain: false
}).pretty();

```

Wynik może i ten sam, ale zapytanie tym razem poprawne.

## 8. Atak, w którym uczestniczyło najwięcej terrorystów.

Wybieramy te ataki, w których jakiś terrorysta na pewno uczestniczył:

```

var match = {
  $match: {
    nperps: {
      $ne: ""
    }
  }
}
};

```

Wybieramy ten największy atak:

```

var sort = {
  $sort: {
    nperps: -1
  }
}

```



```
};

var limit = {
  $limit: 1
};
```

Łączymy agregację w całość:

```
var aggregate8 = db.terrorism.aggregate([match, sort, limit], {
  explain: false
}).pretty();
```

Wynik (dla pełnego obrazu wynikiem jest cały obiekt):

```
{
  "_id" : ObjectId("591ead90043ea5be6ffcd0a5"),
  "eventid" : NumberLong("199409100003"),
  "iyear" : 1994,
  "imonth" : 9,
  "iday" : 10,
  "approxdate" : "",
  "extended" : 0,
  "resolution" : "",
  "country" : 19,
  "country_txt" : "Bangladesh",
  "region" : 6,
  "region_txt" : "South Asia",
  "provstate" : "Dhaka",
  "city" : "Dhaka",
  "latitude" : 23.811388,
  "longitude" : 90.412106,
  "specificity" : 1,
  "vicinity" : 0,
  "location" : "",
  "summary" : "",
  "crit1" : 1,
  "crit2" : 1,
  "crit3" : 1,
  "doubtterr" : 0,
  "alternative" : "",
  "alternative_txt" : ".",
  "multiple" : 0,
  "success" : 1,
  "suicide" : 0,
  "attacktype1" : 9,
  "attacktype1_txt" : "Unknown",
  "attacktype2" : "",
  "attacktype2_txt" : ".",
  "attacktype3" : "",
  "attacktype3_txt" : ".",
  "targtype1" : 3,
  "targtype1_txt" : "Police",
  "targsubtype1" : 24,
  "targsubtype1_txt" : "Police Checkpoint",
  "corp1" : "Police",
  "target1" : "Police Barracades",
  "natlty1" : 19,
```

```
"natlty1_txt" : "Bangladesh",
"targtype2" : "",
"targtype2_txt" : ".",
"targsubtype2" : "",
"targsubtype2_txt" : ".",
"corp2" : "",
"target2" : "",
"natlty2" : "",
"natlty2_txt" : ".",
"targtype3" : "",
"targtype3_txt" : ".",
"targsubtype3" : "",
"targsubtype3_txt" : ".",
"corp3" : "",
"target3" : "",
"natlty3" : "",
"natlty3_txt" : ".",
"gname" : "Islamist Extremists",
"gsubname" : "Fundamentalists",
"gname2" : "",
"gsubname2" : "",
"gname3" : "",
"ingroup" : 421,
"ingroup2" : "",
"ingroup3" : "",
"gsubname3" : "",
"motive" : "",
"guncertain1" : 0,
"guncertain2" : "",
"guncertain3" : "",
"nperps" : 25000,
"nperpcap" : "",
"claimed" : "",
"claimmode" : "",
"claimmode_txt" : ".",
"claim2" : "",
"claimmode2" : "",
"claimmode2_txt" : ".",
"claim3" : "",
"claimmode3" : "",
"claimmode3_txt" : ".",
"compclaim" : "",
"weaptype1" : 13,
"weaptype1_txt" : "Unknown",
"weapsubtype1" : "",
"weapsubtype1_txt" : ".",
"weaptype2" : "",
"weaptype2_txt" : ".",
"weapsubtype2" : "",
"weapsubtype2_txt" : ".",
"weaptype3" : "",
"weaptype3_txt" : ".",
"weapsubtype3" : "",
"weapsubtype3_txt" : ".",
"weaptype4" : "",
"weaptype4_txt" : ".",
"weapsubtype4" : "",
"weapsubtype4_txt" : ".",
"weapdetail" : "",
"nkill" : "",
```

```

"nkillus" : "",
"nkillter" : "",
"nwound" : 200,
"nwoundus" : "",
"nwoundte" : "",
"property" : 1,
"propextent" : "",
"propextent_txt" : ".",
"propvalue" : "",
"propcomment" : "",
"ishostkid" : 0,
"nhostkid" : "",
"nhostkidus" : "",
"nhours" : "",
"ndays" : "",
"divert" : "",
"kidhijcountry" : "",
"ransom" : 0,
"ransomamt" : "",
"ransomamtus" : "",
"ransompaid" : "",
"ransompaidus" : "",
"ransomnote" : "",
"hostkidoutcome" : "",
"hostkidoutcome_txt" : ".",
"released" : "",
"addnotes" : "",
"scite1" : "",
"scite2" : "",
"scite3" : "",
"dbsource" : "PGIS",
"INT_LOG" : -9,
"INT_IDEO" : -9,
"INT_MISC" : 0,
"INT_ANY" : -9,
"related" : ""
}

```

Wnioski:

25 tysięcy islamskich terrorystów w 1994 roku walczyło z policją w Bangladeszu.

## Java

Dodatkowo zainstalowałem sterownik MongoDB do języka Java i wykonałem w nim przykładowe zapytanie.

Sterownik można zainstalować poprzez narzędzie Maven. Należy w projekcie umieścić plik [pom.xml](#) o takiej treści i go zsynchronizować:

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-
4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>Mongo</groupId>
  <artifactId>Mongo-test</artifactId>

```

```

<version>1.0-SNAPSHOT</version>
<dependencies>
  <dependency>
    <groupId>org.mongodb</groupId>
    <artifactId>mongo-java-driver</artifactId>
    <version>2.13.3</version>
  </dependency>
  <!-- https://mvnrepository.com/artifact/com.google.code.gson/gson -->
  <dependency>
    <groupId>com.google.code.gson</groupId>
    <artifactId>gson</artifactId>
    <version>2.8.0</version>
  </dependency>
</dependencies>

</project>

```

Niestety budowanie zapytań Mongo w Java jest dość uciążliwe, a składnia niezbyt przyjemna. Dość proste zapytanie wygląda [tak](#) tak:

```

import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import com.mongodb.*;

import java.util.ArrayList;
import java.util.List;

public class Main {

    public static void main( String args[] ) {

        try{

            MongoClient mongoClient = new MongoClient( "localhost" , 27017 );

            DB db = mongoClient.getDB( "nosql" );
            DBCollection coll = db.getCollection("terrorism");
            DBObject myDoc = coll.findOne();
            System.out.println(myDoc);

            DBObject match = new BasicDBObject("$match", new BasicDBObject("region", new BasicDBObject("$ne", "Unknown")));

            DBObject groupFields = new BasicDBObject( "_id", "$region");
            groupFields.put("region", new BasicDBObject("$first", "$region_txt"));
            groupFields.put("count", new BasicDBObject( "$sum", 1));
            DBObject group = new BasicDBObject("$group", groupFields );

            DBObject sortFields = new BasicDBObject("count", -1);
            DBObject sort = new BasicDBObject("$sort", sortFields );

            DBObject fields = new BasicDBObject("_id", 0);
            DBObject project = new BasicDBObject("$project", fields );

            List<DBObject> pipeline = new ArrayList();
            pipeline.add(match);

```

```
pipeline.add(group);
pipeline.add((sort));
pipeline.add(project);

AggregationOutput output = coll.aggregate(pipeline);

Gson gson = new GsonBuilder().setPrettyPrinting().create();
String json = gson.toJson(output.results());
System.out.println(json);
} catch (Exception e) {
    System.err.print( e.getClass().getName() + ": " + e.getMessage() );
}
}
```