Comparing the Frequency and Severity of Adverse Events among Adult Vaccinations in 2019-2021

Andrew Huang, Emma McMillan, Ben Mosong, Yanyu Tao

Brown University, Providence, RI, USA

BROWN Center for Biomedical Informatics

ABSTRACT

The Vaccine Adverse Event Reporting System (VAERS) is a national vaccine post-marketing surveillance program designed to detect adverse events that were not detected in smaller, pre-marketing studies. Using a dictionary for symptoms, we coded events as serious or non-serious and used frequency tables and proportional reporting ratios (PRRs) to compare reports among most frequently reported vaccines and adverse events between Jan 1, 2019 and March 26, 2021. RESULTS. DISCUSSION

INTRODUCTION

- Vaccines are highly successful means of preventing illness, but pose risks of adverse events and complications
- Pre-marketing clinical trials are limited in ability to detect adverse events due to small cohort sizes, short durations, and lack of enrolled patient diversity (Sultuna, 2013)
- Vaccine Adverse Event Reporting System (VAERS) is post-marketing surveillance system co-sponsored by FDA & CDC used to detect safety issues in USA authorized vaccines
- Reports entered by patients, vaccine manufacturers, physicians
- VAERS data cannot be used to determine causality, since events reported with specific vaccinations may have occurred coincidentally and are not related to vaccination itself
- Symptom data coded using Medical Dictionary for Regulatory Activities (MedDRA), but VAERS does not provide serious vs. non-serious event breakdown, which we created for this project

RESEARCH QUESTIONS

1. Among reports of all vaccine recipients from 2019-2021, which vaccines are most frequently reported in the VAERS dataset among adult patients? What are the symptoms/adverse events most commonly reported for this vaccine? (Aim 1)
2. Among these vaccines, what are the proportional reporting ratios for serious vs. non-serious events? (Aim 2)
3. What are the relative frequencies of serious events (and general, all AEs) for groups of vaccine distributors? (Aim 3)
4. What are the proportional reporting ratio of serious AEs between age groups across the emergency use authorized COVID-19 vaccines, including Pfizer/BioNTech, Moderna, and Janssen)? (Aim 4)

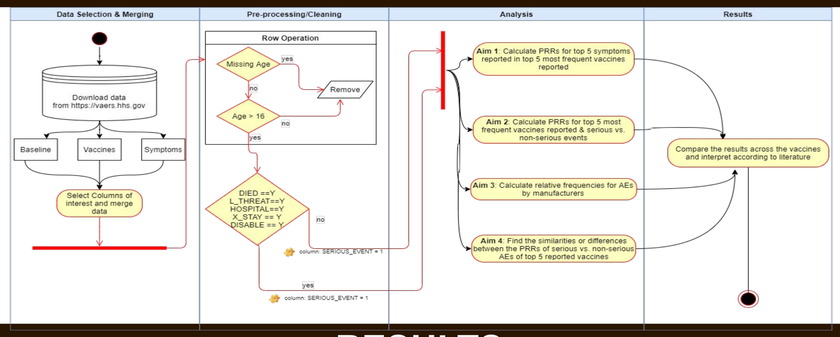
ANALYSIS METHODS

1. Export data from VAERS site: 3 files from VAERS data (data, symptoms, vaccine), from Jan 1, 2019 - Mar 26, 2021
2. Pre-process and conduct exploratory data analysis in Julia
   a. Packages used: CSV, Dataframes, Plots, StatsPlots, StatsBase, FreqTables, Missing, Query, HypothesisTests
   b. Included rows of px ≥16 yrs old and w/o missing age data
3. Codebook to dichotomize events as serious or non-serious based on FDA Code of Federal Regulations ("CFR"))
4. Tabulate frequencies of events by vaccine, top adverse events, serious vs. non serious events, vaccine manufacturer
5. Calculate Proportional Reporting Ratios (PRRs) for vaccines/adverse event pairs specified using above Aims using contingency tables and formula in Julia (modeled from below)

	Reported VAE "Y"	Other reported VAE
Reports following vaccine "X"	a	b
Reports following other vaccines	c	d

$$PRR = \frac{a / (a + b)}{c / (c + d)}$$

METHODS OVERVIEW



RESULTS

- Identified a total of 103,050 reported adult vaccinations from 1/1/2019 to 3/26/2021 submitted to VAERS, and includes 112 vaccines and 5,565 MedDRA symptoms.
- 9.6% (n=10,054) of all reports contained reports of serious events
- Among the 94,840 reporting adults, 27% were 16 to 39 years old, 41% age 40 to 64, and 32% were age 65 and older. 72% identified Female.

Figure 1. Frequency of VAERS report by date

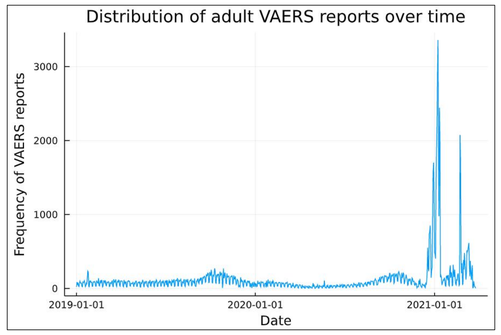


Figure 2. PRRs of symptoms that raises potential signals

PNEUMO (PNEUMOVAX)		COVID19 (JANSSEN)	
Erythema	4.02	Headache	2.28
Injection site erythema	2.54	Chills	2.71
Pain in extremity	1.89	Pyrexia	2.29
Injection site swelling	3.16	Pain	1.77
Injection site pain	1.77	Fatigue	2.0

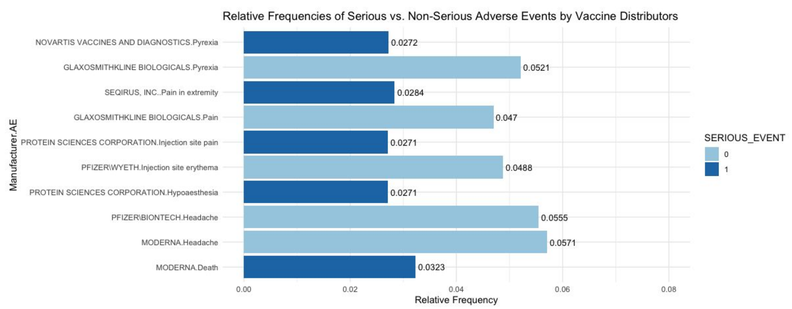
Table 1. Number of serious reports & PRR by vaccine (Aim 2)

Vaccine	# of reports	# of serious events for specific vaccine	PRR serious vs. non-serious
COVID-19 Pfizer	23,630	3,621	1.89
Zoster (Shingles)	21,666	668	0.27
COVID-19 Moderna	20,815	3,611	2.21
Pneumovax	4,810	215	0.45
COVID-19 Janssen	2,774	154	0.56

Table 2. PRR of Serious Event by COVID-19 Vaccine by Age Group (Aim 4)

Age Group	COVID-19 Pfizer	COVID-19 Moderna	COVID-19 Janssen
16 to 39	1.07	1.05	0.49
40 to 49	1.02	1.19	0.62
50 to 59	1.62	2.16	0.59
60 to 64	2.16	2.69	0.88
65 to 74	4.29	4.49	0.97
75+	2.91	3.08	1.38
All (16 to 75+)	1.89	2.22	0.56

Figure 3. Relative frequencies of top 5 serious vs. top 5 non-serious AEs for different vaccine distributors (Aim 3)



DISCUSSION

- Summary Reporting Ratios
  - Aim 1:COVID19 (JANSSEN) and PNEUMO (PNEUMOVAX) raised 3 potential signals each for possible investigation.
  - Aim 2: larger PRRs associated with 3 COVID-19 vaccines and serious events
  - Aim 4: similar PRRs between Pfizer and Moderna, within strata of age; smaller PRRs for Janssen.
- Summary Frequency by Distributor
  - In Aim 3, found relative frequencies of serious AEs are approximately half of the frequencies of non-serious AEs, across all distributors
  - No difference in the means of number of reported AEs between Moderna and Pfizer
  - Difference in the means of number of reported AEs between Moderna and Seqirus
- Limitations of approach
  - VAERS dataset is a passive surveillance system that relies on unverified reports of health events occurring after vaccination
  - Cannot determine whether patient, provider, parent or manufacturer submitted report
  - PRRs cannot be used to assume causal relationship b/t vaccines and adverse events
  - Codebook for serious did not code office visits or emergency department visits as serious, likely underestimating total # of serious events
  - 3 of 5 of most frequently reported vaccines are COVID-19 vaccines, and studied since end 2020
- Strength(s)
  - In analysis, we were able to classify events using simple, novel codebook, as serious vs. non serious

CONCLUSION

Post-marketing surveillance of safety data from vaccines is critical to ensure that benefits of vaccines continue to outweigh their risks. Applying analytical tools and statistical methods (i.e. contingency tables, PRRs)t, we were able to identify differences among vaccine reports. At the same time, we developed a novel system to simplify characterization of reported events as being serious or non-serious. It is imperative that this post-marketing safety data is collected and made accessible to everyone in a responsible manner. Further analysis and publication of such analysis can help ensure appropriate transmittal of vaccine safety data to the public, ultimately supporting public health.

References

1. "CFR - Code of Federal Regulations Title 21." *Accessdata.fda.gov*, www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/cfrsearch.cfm?fr=312.32.
2. Sultana J, Cutroneo P, Trifirò G. Clinical and economic burden of adverse drug reactions. J Pharmacol Pharmacother. 2013 Dec;4(Suppl 1):S73-7. doi: 10.4103/0976-500X.120957. PMID: 24347988; PMCID: PMC3853675.

Acknowledgments

Thank you to Dr. Chen, Dr. Sarkar, Dilum, Michelle for support on our project.

Create dict of VAERS reports and symptoms

Original VAERS data structure:

	VAERS_ID	SYMPTOMS
	Int64	Array…
1	794156	["Injected limb mobility decreased", "Injection site joint pain"]
2	794157	["Apathy", "Arthralgia", "Asthenia", "Injection site erythema", "Injection site pain"]
3	794157	["Injection site pruritus", "Injection site swelling", "Injection site warmth", "Listless", "Night sweats"]
4	794158	["Chills", "Headache", "Nausea", "Pain", "Pyrexia"]
5	794159	["Injection site erythema", "Injection site swelling", "Injection site warmth", "Pain"]
6	794160	["Injection site swelling", "Lip blister", "Lip swelling", "Pain"]
7	794160	["Asthenia", "Chills", "Fatigue", "Influenza like illness", "Injection site erythema"]
8	794161	["Pyrexia"]

New VAERS data structure:

	VAERS_ID	SYMPTOMS
	Int64	Array…
1	794156	["Injection site joint pain", "Injected limb mobility decreased"]
2	794157	["Apathy", "Injection site pain", "Injection site pruritus", "Asthenia", "Arthralgia", "Injection site erythema", "Injection site warmth", "Injection site swelling", "Night sweats", "Listless"]
3	794158	["Pain", "Headache", "Nausea", "Pyrexia", "Chills"]
4	794160	["Lip swelling", "Lip blister", "Pain", "Influenza like illness", "Asthenia", "Injection site erythema", "Fatigue", "Injection site swelling", "Chills"]
5	794161	["Pyrexia"]

Code (https://github.com/methods2021/8_v-team):

```
# Collapse the rows so that we get a dict of symptoms for each VAERS_ID
# Step 1: Create the dict
vaers_id_to_symptoms_dict = Dict{Int, Set{String}}()
# Step 2: Populate the keys (VAERS_ID) of the dict
for rownumber in 1:size(test, 1)
    vaers_id = test[rownumber, :VAERS_ID]
    if !haskey(vaers_id_to_symptoms_dict, vaers_id)
        # this is the set where we will store all of the symptoms for this VAERS ID
        vaers_id_to_symptoms_dict[vaers_id] = Set{String}()
    end
end
# Step 3: Populate the values (SYMPTOMS) of the dict
for rownumber in 1:size(test, 1)
    vaers_id = test[rownumber, :VAERS_ID]
    symptoms = test[rownumber, :SYMPTOMS]
    for symptom in symptoms
        push!(vaers_id_to_symptoms_dict[vaers_id], symptom)
    end
end
# View dict
vaers_id_to_symptoms_dict
```

```
Dict{Int64, Set{String}} with 135054 entries:
  870067  => Set(["Platelet count decreased"])
  1043880 => Set(["Aspiration", "Death", "Insomnia", "Seizure"])
  900301  => Set(["Coronavirus infection", "Cough", "Dyspnoea"])
  905057  => Set(["Influenza"])
  818452  => Set(["Abdominal distension", "Flatulence", "Abdominal pain upper",…
  870391  => Set(["Product storage error"])
  1051588 => Set(["Paraesthesia"])
  1048923 => Set(["Pain", "Pyrexia", "Pain in extremity"])
```

Special thanks to Dilum for helping write the code in Office Hours!

Label reports as serious or non-serious

Per FDA CFR (intended for randomized trials), **a serious adverse event** = "if in the view of either the investigator or sponsor, it results in any of the following outcomes: **death, life-threatening adverse event, inpatient hospitalization or prolongation of existing hospitalization, persistent or significant incapacity or substantial disruption of the ability to conduct normal life functions, or a congenital anomaly/birth defect**" ("CFR")

In [48]:
```
#labeling of events as serious or non-serious
a = (df.DIED .== "Y") .| (df.L_THREAT .== "Y") .| (df.HOSPITAL .== "Y") .| (df.X_STAY .== "Y") .|
(df.DISABLE .== "Y") .| (df.BIRTH_DEFECT .== "Y")

df.SERIOUS_EVENT = replace(a, missing => 0)

show(names(df))
```

```
["VAERS_ID", "YEAR", "RECVDATE", "STATE", "AGE_YRS", "CAGE_YR", "CAGE_MO", "SEX", "RPT_DATE", "SYMPTOM_TEXT", "DIED",
"DATEDIED", "L_THREAT", "ER_VISIT", "HOSPITAL", "HOSPDAYS", "X_STAY", "DISABLE", "RECOVD", "VAX_DATE", "ONSET_DATE",
"NUMDAYS", "LAB_DATA", "V_ADMINBY", "V_FUNDBY", "OTHER_MEDS", "CUR_ILL", "HISTORY", "PRIOR_VAX", "SPLTTYPE", "FORM_VE
RS", "TODAYS_DATE", "BIRTH_DEFECT", "OFC_VISIT", "ER_ED_VISIT", "ALLERGIES", "VAX_TYPE", "VAX_MANU", "VAX_LOT", "VAX_
DOSE_SERIES", "VAX_ROUTE", "VAX_SITE", "VAX_NAME", "SYMPTOMS", "SERIOUS_EVENT"]
```

In [49]:
```
#count # of serious event
sum(df.SERIOUS_EVENT)
```

Out[49]: 10099

In [50]:
```
#SERIOUS_EVENT =1 is serious and SERIOUS_EVENT=0 is non-serious adverse event
#find most frequent serious adverse event - pull from SYMPTOM1, SYMPTOM2, SYMPTOM3, SYMPTOM4, SYMPTOM5
#find most frequent non-serious adverse event - pull from SYMPTOM1, SYMPTOM2, SYMPTOM3, SYMPTOM4, SYMPTOM5
df_nonserious = df[df.SERIOUS_EVENT .== 0, :]
df_serious = df[df.SERIOUS_EVENT .== 1, :]
```

Out[50]: 10,099 rows × 45 columns (omitted printing of 36 columns)

	VAERS_ID	YEAR	RECVDATE	STATE	AGE_YRS	CAGE_YR	CAGE_MO	SEX	RPT_DATE
	Int64	Int64	String	String?	Float64?	Int64?	Float64?	String	String?
1	794197	2019	1/2/19	MN	56.0	56	*missing*	F	*missing*
2	794197	2019	1/2/19	MN	56.0	56	*missing*	F	*missing*
3	794210	2019	1/2/19	AZ	52.0	52	*missing*	M	*missing*
4	794271	2019	1/2/19	NJ	43.0	43	*missing*	F	*missing*
5	794307	2019	1/2/19	CA	40.0	40	*missing*	F	*missing*
6	794307	2019	1/2/19	CA	40.0	40	*missing*	F	*missing*
7	794354	2019	1/2/19	CA	66.0	66	*missing*	F	*missing*
8	794403	2019	1/3/19	OR	53.0	53	*missing*	F	*missing*
9	794461	2019	1/3/19	WA	79.0	79	*missing*	M	12/28/18

In [51]:
```
df_SERIOUS = combine(groupby(combine(groupby(df, :VAERS_ID),
             :SERIOUS_EVENT=>:SERIOUS_EVENT), :SERIOUS_EVENT), nrow =>
    :COUNT);
df_SERIOUS
```

Out[51]: 2 rows × 2 columns

	SERIOUS_EVENT	COUNT
	Int64	Int64
1	0	92951
2	1	10099

Github: https://github.com/methods2021/8_v-team/blob/master/serious%20vs%20nonserious%20labeling.jl

1. "CFR - Code of Federal Regulations Title 21." *Accessdata.fda.gov*, www.accessdata.fda.gov/scripts/cdrh/cfdocs/cfcfr/cfrsearch.cfm?fr=312.32.

Calculate PRRs for top 5 symptoms reported in top 5 most frequent vaccines reported (Aim 1)

ZOSTER (SHINGRIX)		COVID19 (PFIZER-BIONTECH)		PNEUMO (PNEUMOVAX)		COVID19 (MODERNA)		COVID19 (JANSSEN)	
Pyrexia	1.59	Headache	1.40	Erythema	4.02	Headache	1.2	Headache	2.28
Chills	1.68	Fatigue	1.35	Injection site erythema	2.54	Pyrexia	0.95	Chills	2.71
Pain	1.46	Chills	1.11	Pain in extremity	1.89	Chills	1.04	Pyrexia	2.29
Headache	1.24	Pyrexia	0.92	Injection site swelling	3.16	Fatigue	1.16	Pain	1.77
Injection site pain	1.66	Pain	0.92	Injection site pain	1.77	Pain	0.86	Fatigue	2.0

```
list_symptoms_mostlikely = Dict{String, Vector{String}}()

for vax in ["COVID19 (COVID19 (PFIZER-BIONTECH))",
            "ZOSTER (SHINGRIX)",
            "COVID19 (COVID19 (MODERNA))",
            "PNEUMO (PNEUMOVAX)",
            "COVID19 (COVID19 (JANSSEN))"]
    # Create dict of most reported symptoms for each vaccine
    df_vax = filter(row -> row.VAX_NAME == vax, df)

    symptoms_all_dupes = reduce(vcat, df_vax.SYMPTOMS)
    symptoms_freq_dict = StatsBase.countmap(symptoms_all_dupes)
    list_symptoms = unique(reduce(vcat, df_vax.SYMPTOMS))
    freq_per_symptom = [symptoms_freq_dict[val] for val in list_symptoms]
    perm = sortperm(freq_per_symptom; rev=true)
    # only pick the top 5 symptoms for each of the vaccine
    list_symptoms_mostlikely[vax] = list_symptoms[perm[1:5]]
end

prr_dict = Dict()

for (vax, symptoms) in list_symptoms_mostlikely
    symp_prr = []
    for symptom in symptoms
        push!(symp_prr, string(symptom, "(PRR: ",
                round(get_PRR(get_freqtable(df, vax, symptom)); digits=2), ")"))
    end
    prr_dict[vax] = symp_prr
end
prr_dict
```

When PRR > 2, it's a potential signal for investigation

COVID19 (JANSSEN) and PNEUMO (PNEUMOVAX) raised 3 potential signals each

Headache, Pyrexia and Chills are the common symptoms among the top 5 vaccines. Their PRRs ranks high too.

https://github.com/methods2021/8_v-team/blob/master/ben_aim1.jl

Calculate PRRs for top 5 most frequent vaccines reported & serious vs. non-serious events (Aim 2)

PRR using Julia (via Jupyter notebooks) sample:

In [62]:
```julia
#function for aims analysis, to calc PRR from contingency tables created below

function get_freqtable(df, vax_name_str, symptom_str)
    test = select(df, ["VAX_NAME", "SYMPTOMS"])
    # Create dummy variable for vax_name
    test.VAX_IND = (test.VAX_NAME .== vax_name_str)
    # Create dummy variable for symptom
    test.SYMPTOM_IND = zeros(size(test, 1))
    for rownumber in 1:size(test, 1)
        if symptom_str in test[rownumber,:SYMPTOMS]
            test[rownumber,:SYMPTOM_IND] = 1
        end
    end
    # Create frequency table
    tbl = freqtable(test, :VAX_IND, :SYMPTOM_IND)
    return tbl
end;
function get_PRR(tbl)
    # Calculate PRR
    a = tbl[2,2]
    b = tbl[2,1]
    c = tbl[1,2]
    d = tbl[1,1]
    PRR = (a/(a+b))/(c/(c+d))
    return PRR
end;
```

In [63]:
```julia
# PRR for #1 freq VAX = "COVID19 (COVID19 (PFIZER-BIONTECH))" and serious report
df.COVID19_PFIZER = (df.VAX_NAME .== "COVID19 (COVID19 (PFIZER-BIONTECH))");
tbl = freqtable(df, :COVID19_PFIZER, :SERIOUS_EVENT)
```

Out[63]:
```
2×2 Named Array{Int64,2}
COVID19_PFIZER \ SERIOUS_EVENT │     0      1
──────────────────────────────┼──────────────
false                         │ 72960   6460
true                          │ 19991   3639
```

D → false/0 (72960); C → false/1 (6460); B → true/0 (19991); A → true/1 (3639)

In [64]:
```julia
#PRR for Pfizer COVID-19
get_PRR(tbl)
```

Out[64]:
```
1.8932837121306416
```

PRR by hand sample:

$$PRR = \frac{A/(A+B)}{C/(C+D)}$$

	Reported adverse event of interest (serious AE)	Reported all other adverse events (non-serious AE)	Total
Reports for COVID-19 Pfizer	3,621 (a)	20,009 (b)	23,630 (a+b)
Reports for all other vaccines	6,433 (c)	72,987 (d)	79,420 (c+d)
Total	10,554	92, 996	103,050

a/(a+b)/[c/(c+d)] =
(3,621/(23,630)/[(6,433/79,420)]
= PRR = **1.89**

PRR Interpretation: There is 1.89 times (a much greater) probability that there were serious events detected in the Pfizer COVID-19 vaccine group vs. all other reports of all other vaccines and non-serious events in the dataset.

GitHub: https://github.com/methods2021/8_v-team/blob/master/emcmill2_aim2_analysis_v2.jl

Construct function of calculating counts for AEs by groups (i.e. vaccine types, distributors); Calculate relative frequencies for AEs by manufacturers (Aim 3)

```
# this function takes in VAERS dataframe, one column name(default), or two column names
# and calculate the overall frequency/counts for col-AE (or col1-col2-AE) pairs
function get_freq(df, col1, col2 = nothing)

    # when only one column name specified
    if isnothing(col2)

        # one observation is associated with 5 symptom columns,
        # calculate the frequency/counts for col1-AE pairs separately by symptom column
        count1 = dropmissing(rename!(combine(groupby(df, [Symbol(col1), :SYMPTOM1]),
                    nrow => :COUNT), :SYMPTOM1 => :SYMPTOM), :SYMPTOM);
        count2 = dropmissing(rename!(combine(groupby(df, [Symbol(col1), :SYMPTOM2]),
                    nrow => :COUNT), :SYMPTOM2 => :SYMPTOM), :SYMPTOM);
        count3 = dropmissing(rename!(combine(groupby(df, [Symbol(col1), :SYMPTOM3]),
                    nrow => :COUNT), :SYMPTOM3 => :SYMPTOM), :SYMPTOM);
        count4 = dropmissing(rename!(combine(groupby(df, [Symbol(col1), :SYMPTOM4]),
                    nrow => :COUNT), :SYMPTOM4 => :SYMPTOM), :SYMPTOM);
        count5 = dropmissing(rename!(combine(groupby(df, [Symbol(col1), :SYMPTOM5]),
                    nrow => :COUNT), :SYMPTOM5 => :SYMPTOM), :SYMPTOM);

        # concatenating, note that col1-AE pairs may not be unique at this step
        count_raw = reduce(vcat, [count1, count2, count3, count4, count5]);

        # recalculating frequency/counts by summing up the counts for non-unique col1-AE pairs
        count = combine(groupby(count_raw, [Symbol(col1), :SYMPTOM]), :COUNT => sum)

    # when two column names specified
    else
        count1 = dropmissing(rename!(combine(groupby(df, [Symbol(col1), Symbol(col2), :SYMPTOM1]),
                    nrow => :COUNT), :SYMPTOM1 => :SYMPTOM), :SYMPTOM);
        count2 = dropmissing(rename!(combine(groupby(df, [Symbol(col1), Symbol(col2), :SYMPTOM2]),
                    nrow => :COUNT), :SYMPTOM2 => :SYMPTOM), :SYMPTOM);
        count3 = dropmissing(rename!(combine(groupby(df, [Symbol(col1), Symbol(col2), :SYMPTOM3]),
                    nrow => :COUNT), :SYMPTOM3 => :SYMPTOM), :SYMPTOM);
        count4 = dropmissing(rename!(combine(groupby(df, [Symbol(col1), Symbol(col2), :SYMPTOM4]),
                    nrow => :COUNT), :SYMPTOM4 => :SYMPTOM), :SYMPTOM);
        count5 = dropmissing(rename!(combine(groupby(df, [Symbol(col1), Symbol(col2), :SYMPTOM5]),
                    nrow => :COUNT), :SYMPTOM5 => :SYMPTOM), :SYMPTOM);

        count_raw = reduce(vcat, [count1, count2, count3, count4, count5]);

        count = combine(groupby(count_raw, [Symbol(col1), Symbol(col2), :SYMPTOM]), :COUNT => sum)
    end
    return count
end
```

Calculate overall counts for MANU-SYPM pair:

```
#get the manufacturer w/ the most frequent aderse event overall
tb1 = sort!(get_freq(df, "VAX_MANU"), :COUNT_sum, rev = true)
```

19,054 rows × 3 columns

	VAX_MANU	SYMPTOM	COUNT_sum
	String	String	Int64
1	GLAXOSMITHKLINE BIOLOGICALS	Pyrexia	5848
2	GLAXOSMITHKLINE BIOLOGICALS	Pain	5300
3	GLAXOSMITHKLINE BIOLOGICALS	Chills	5151

```
#get the manufacturer with most frequent non-serious adverse event overall
tb2 = sort!(filter(x -> x.SERIOUS_EVENT == 0,
        get_freq(df, "VAX_MANU", "SERIOUS_EVENT"))
    :COUNT_sum, rev = true)
```

13,357 rows × 4 columns

	VAX_MANU	SERIOUS_EVENT	SYMPTOM	COUNT_sum
	String	Int64	String	Int64
1	GLAXOSMITHKLINE BIOLOGICALS	0	Pyrexia	5700
2	GLAXOSMITHKLINE BIOLOGICALS	0	Pain	5143
3	GLAXOSMITHKLINE BIOLOGICALS	0	Chills	5066

Calculate relative frequencies by manufacturers:

```
filtered_non_serious = filter(row -> row.VAX_MANU ∈ manu10, tb2);
rf_non_serious = sort(transform(groupby(filtered_non_serious, :VAX_MANU),
                                :COUNT_sum => (x -> x / sum(x)) => :freq),
                      :freq, rev=true)
show(rf_non_serious, allcols = true)
```

Top 3 Manufacturers with the highest frequencies of non-serious AEs:
Moderna (headache), Pfizer (headache), GSK (pyrexia)
Top 3 Manufacturers with the highest frequencies of serious AEs:
Moderna (death), Seqirus (pain in extreme), Novartis (pyrexia)

11377×5 DataFrame

Row	VAX_MANU	SERIOUS_EVENT	SYMPTOM	COUNT_sum	freq
	String	Int64	String	Int64	Float64
1	MODERNA	0	Headache	2526	0.0570667
2	PFIZER\\BIONTECH	0	Headache	3136	0.0554828
3	GLAXOSMITHKLINE BIOLOGICALS	0	Pyrexia	5700	0.0520976
4	PFIZER\\WYETH	0	Injection site erythema	362	0.0487739

Github: https://github.com/methods2021/8_v-team/blob/master/ytao_aim3_analysis.jl