

Severity of adverse reactions in children

May 31, 2020

1 Severity of adverse reactions in pediatric patients

1.1 Data collection and cleaning

We will first collect the data. We will initially collect only data for 'children' i.e. category 3 as reported in the database. This does not include neonates, infants or adolescents.

Given the total number of children in the database (35,000), we will initially restrict our analyses to the first 5000. The sample can be expanded or reduced by changing `n_reports`.

```
[1]: # import standard modules necessary for data processing and visualisation
      # set style for plotting
      import pandas as pd
      import seaborn as sns
      import matplotlib.pyplot as plt
      import numpy as np
      from matplotlib.colors import LogNorm
      import scipy.stats as st
      plt.style.use('seaborn')
      plt.rcParams['figure.figsize'] = (10.0, 6.0)

      # import my modules
      import collect_data
      import clean_data
      import process_data

      # define the base of the URL used to access the database
      # search terms will be appended to this to obtain specific recordsets
      url_base = "https://api.fda.gov/drug/event.json?search=receivedate:
      ↪ [20030101+T0+20200528]"
```

1.1.1 Collecting the pediatric dataset

```
[2]: n_reports = 5000

      # collect pediatric data from database and flatten
      raw_pediatric = collect_data.collect_pediatric_data(n_reports)
```

```
raw_pediatric.reset_index(drop=True, inplace=True)
flat_pediatric = collect_data.flatten_dataframe(raw_pediatric)
```

```
HBox(children=(FloatProgress(value=0.0, description='Progress', max=50.0, style=ProgressStyle(
```

```
[3]: flat_pediatric.head(5)
```

```
[3]:  reporttype receiptdateformat      companynumb occurcountry \
0          1             102  US-GILEAD-2012-0061944        US
1          1             102  US-GILEAD-2012-0059314        US
2          1             102  US-GILEAD-2012-0060537        US
3          1             102  US-GILEAD-2012-0060700        US
4          1             102  US-GILEAD-2012-0061671        US

      safetyreportversion receivedateformat duplicate transmissiondateformat \
0                        3             102          1             102
1                        3             102          1             102
2                        3             102          1             102
3                        2             102          1             102
4                        3             102          1             102

      fulfilllexpeditecriteria safetyreportid ... \
0                        2      10003430 ...
1                        2      10003517 ...
2                        2      10004354 ...
3                        2      10004368 ...
4                        2      10004919 ...

      spl_id substance_name \
0  acf09b42-e9a4-4aee-82f7-75d413d06ec5  AMBRISANTAN
1  acf09b42-e9a4-4aee-82f7-75d413d06ec5  AMBRISANTAN
2  acf09b42-e9a4-4aee-82f7-75d413d06ec5  AMBRISANTAN
3  acf09b42-e9a4-4aee-82f7-75d413d06ec5  AMBRISANTAN
4  acf09b42-e9a4-4aee-82f7-75d413d06ec5  AMBRISANTAN

      product_type route application_number  nui pharm_class_cs \
0  HUMAN PRESCRIPTION DRUG  ORAL      NDA022081  NaN      NaN
1  HUMAN PRESCRIPTION DRUG  ORAL      NDA022081  NaN      NaN
2  HUMAN PRESCRIPTION DRUG  ORAL      NDA022081  NaN      NaN
3  HUMAN PRESCRIPTION DRUG  ORAL      NDA022081  NaN      NaN
4  HUMAN PRESCRIPTION DRUG  ORAL      NDA022081  NaN      NaN

      pharm_class_epc pharm_class_moa pharm_class_pe
0                NaN                NaN                NaN
1                NaN                NaN                NaN
```

2	NaN	NaN	NaN
3	NaN	NaN	NaN
4	NaN	NaN	NaN

[5 rows x 87 columns]

We can see that we have a flattened dataframe, with no nested dictionaries or lists (in any desired columns). However, it is also clear that there are numerous missing values and a large number of columns (87). We must reduce this featureset and clean the data before analysis.

1.1.2 Cleaning the data

We clean the data to remove unnecessary/undesired columns, impute missing values (where appropriate), remove columns with a high fraction of missing values, format types appropriately, and filter any outliers.

```
[4]: # remove unnecessary columns
clean_pediatric = clean_data.drop_unnecessary_columns(flat_pediatric)

# replace missing values in seriousness outcome columns with 0
# assume that no entry means this outcome did not take place
clean_pediatric = clean_data.fill_seriousness_nan(clean_pediatric)

# remove any columns which have more than 40% missing values
clean_pediatric = clean_data.remove_nan_columns(clean_pediatric, 40)

# fix data types and formatting
clean_pediatric = clean_data.fix_data_types(clean_pediatric)
clean_pediatric = clean_data.reformat_onsetage(clean_pediatric)

# remove outliers in patient age category
clean_pediatric = clean_data.remove_outliers(clean_pediatric)

# remap seriousness from {1, 2} to {1, 0}
clean_pediatric.serious = clean_pediatric.serious.map({2:0, 1:1}) #map ↪
↪serious_to_boolean

pediatric_data = clean_pediatric.reset_index()
```

Examine the cleaned and reduced dataset by eye to look for anomalies and problems.

```
[5]: # preview data
pediatric_data.head(10)
```

```
[5]:   index  reporttype  occurcountry  serious  primarysource.qualification \
0      0          1.0            US         0                      5.0
1      1          1.0            US         0                      1.0
```

2	2	1.0	US	0	1.0
3	3	1.0	US	0	5.0
4	4	1.0	US	0	5.0
5	5	1.0	US	1	5.0
6	6	1.0	US	0	5.0
7	7	1.0	US	0	1.0
8	8	1.0	US	0	1.0
9	9	1.0	US	0	5.0

	patient.patientsex	seriousnessother	seriousnesshospitalization	\
0	1.0	0.0	0.0	
1	1.0	0.0	0.0	
2	1.0	0.0	0.0	
3	2.0	0.0	0.0	
4	2.0	0.0	0.0	
5	2.0	1.0	0.0	
6	1.0	0.0	0.0	
7	2.0	0.0	0.0	
8	1.0	0.0	0.0	
9	2.0	0.0	0.0	

	seriousnesslifethreatening	seriousnessdeath	seriousnessdisabling	\
0	0.0	0.0	0.0	
1	0.0	0.0	0.0	
2	0.0	0.0	0.0	
3	0.0	0.0	0.0	
4	0.0	0.0	0.0	
5	0.0	0.0	0.0	
6	0.0	0.0	0.0	
7	0.0	0.0	0.0	
8	0.0	0.0	0.0	
9	0.0	0.0	0.0	

	seriousnesscongenitalanomaly	reactionoutcome	reactionmeddrapt	route	\
0	0.0	6.0	Sinusitis	ORAL	
1	0.0	6.0	Hiccups	ORAL	
2	0.0	6.0	Hordeolum	ORAL	
3	0.0	6.0	Oral candidiasis	ORAL	
4	0.0	6.0	Nasopharyngitis	ORAL	
5	0.0	6.0	Coeliac disease	ORAL	
6	0.0	6.0	Heart rate increased	ORAL	
7	0.0	6.0	Dyspnoea	ORAL	
8	0.0	6.0	Sinusitis	ORAL	
9	0.0	6.0	Sinusitis	ORAL	

	patient.patientonsetageyear
0	10.0

1	4.0
2	4.0
3	2.0
4	5.0
5	13.0
6	7.0
7	6.0
8	6.0
9	8.0

1.2 Analysis of factors influencing severity of adverse response in children

1.2.1 Descriptive statistics

We initially calculate some basic descriptive statistics on the numerical data in the dataframe.

```
[6]: pediatric_data.describe()
```

```
[6]:
```

	index	reporttype	serious	primarysource.qualification \
count	4991.000000	4991.000000	4991.000000	4974.000000
mean	2498.518734	1.257864	0.535965	3.488741
std	1443.076841	0.450591	0.498755	1.703362
min	0.000000	1.000000	0.000000	1.000000
25%	1249.500000	1.000000	0.000000	1.000000
50%	2497.000000	1.000000	1.000000	5.000000
75%	3748.500000	2.000000	1.000000	5.000000
max	4999.000000	4.000000	1.000000	5.000000

	patient.patientsex	seriousnessother	seriousnesshospitalization \
count	4030.000000	4991.000000	4991.000000
mean	1.331266	0.364857	0.166299
std	0.470727	0.481438	0.372386
min	1.000000	0.000000	0.000000
25%	1.000000	0.000000	0.000000
50%	1.000000	0.000000	0.000000
75%	2.000000	1.000000	0.000000
max	2.000000	1.000000	1.000000

	seriousnesslifethreatening	seriousnessdeath	seriousnessdisabling \
count	4991.000000	4991.000000	4991.000000
mean	0.017832	0.048487	0.006211
std	0.132354	0.214815	0.078574
min	0.000000	0.000000	0.000000
25%	0.000000	0.000000	0.000000
50%	0.000000	0.000000	0.000000
75%	0.000000	0.000000	0.000000

max	1.000000	1.000000	1.000000
-----	----------	----------	----------

	seriousnesscongenitalanomaly	reactionoutcome	\
count	4991.000000	4870.000000	
mean	0.003206	4.596099	
std	0.056534	2.000540	
min	0.000000	1.000000	
25%	0.000000	3.000000	
50%	0.000000	6.000000	
75%	0.000000	6.000000	
max	1.000000	6.000000	

	patient.patientonsetageyear
count	2586.000000
mean	7.881574
std	3.320164
min	0.666667
25%	5.000000
50%	8.000000
75%	11.000000
max	17.000000

Not all of this data is meaningful but we can still see some useful properties of the data from these statistics.

Basic checks of the boolean and categorical data have maxima and minima as expected. The means of boolean data give an indication of the skew, e.g. 54% of the reports are 'serious', defined as an adverse event resulting in death, a life threatening condition, hospitalization, disability, congenital anomaly, or other serious condition. We can also quickly see that the data are skewed towards male patients and that a serious outcome of 'other' is reported for almost 50% of the dataset.

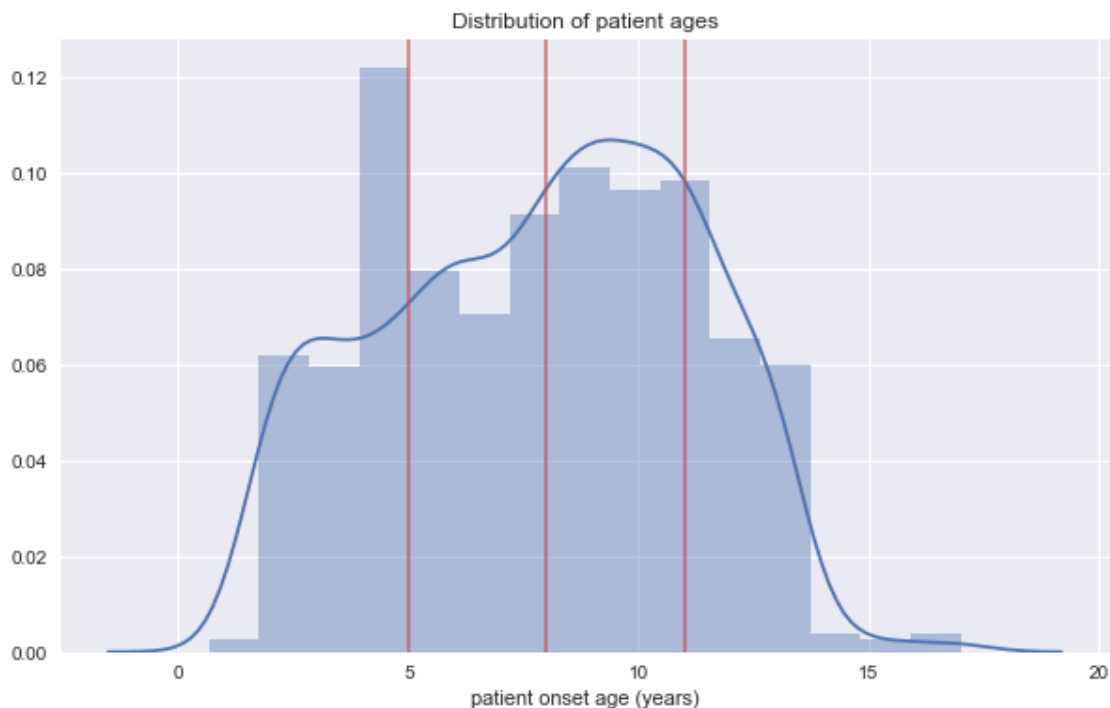
The only continuous numerical data in this dataset is the age of the patient at the onset of the adverse response. The oldest patient in our sample is 17 (we removed any outliers above this threshold) whilst the youngest is 0.6yr. The 'children' category on which the data was selected encompasses a wide range of (pediatric) ages. More detailed future analysis should consider selecting data across all pediatric categories rather than just 'child' since it seems the categories are subject to some interpretation.

Within this data, the standard deviation of patient age is 3.3 years with a mean of 7.8 years. The ages seem fairly symmetrically distributed. We plot this distribution below and mark the quartiles in red.

```
[7]: # plot distribution of patients ages
data_age = pediatric_data['patient.patientonsetageyear']
ax = sns.distplot(data_age, bins=15, kde=True);
ax.set_xlabel('patient onset age (years)');
ax.set_title('Distribution of patient ages');

# plot the quartiles
```

```
for q in [0.25,0.5,0.75]:
    ax.axvline(data_age.quantile(q), 0, 1, color='C2', alpha=0.7)
```



1.2.2 Exploring types of adverse response outcomes

Similar to our analysis for the full database (presented in the [Exploring OpenFDA Adverse Reactions notebook](#)), we can now perform some basic analysis and exploration of the cleaned pediatric dataset before building a more sophisticated model. We are primarily interested in the severity of the adverse response and so we start by exploring this feature.

We begin by considering the relative proportions of seriousness outcomes in the data.

```
[8]: # create pie chart for serious vs not serious
fig1, (ax1, ax2) = plt.subplots(nrows=1, ncols=2, figsize=(18,5))
pediatric_data.serious.value_counts().plot.pie(labels=['serious', 'not_
→serious'], ax=ax1, autopct='%1.0f%%');
ax1.set_ylabel('severity of adverse response');

# create pie chart showing all seriousness outcomes
seriousness_col_names = ['seriousnesslifethreatening',
                        'seriousnesscongenitalanomaly',
                        'seriousnesshospitalization',
                        'seriousnessother',
```

```

        'seriousnessdeath' ,
        'seriousnessdisabling'
    ]
labels = [item.replace('seriousness','') for item in seriousness_col_names]
df_serious = pediatric_data[seriousness_col_names]
df_serious.sum().plot.pie(ax=ax2, autopct='%1.0f%%', labels=labels,
    ↪startangle=-20)
ax2.set_ylabel('nature of serious response');

```



We can see that the fraction of reported serious adverse responses is similar to the entire population calculated in our [population study](#) (54% compared to 59%). Of those where the nature of the serious response was reported, the majority (60%) reported a response that did not result in death, disability, congenital anomaly, a threat to life or hospitalisation.

Two different but comparable metrics exist within the database that give an indication of severity. We can gain some understanding of the accuracy of the data by comparing these metrics. We know whether the outcome was classified as 'serious' or 'not serious'. We also have the nature of the outcome as falling into one of these categories:

- 1 = Recovered/resolved
- 2 = Recovering/resolving
- 3 = Not recovered/not resolved
- 4 = Recovered/resolved with sequelae (consequent health issues)
- 5 = Fatal
- 6 = Unknown

We can examine the relative frequencies of these outcomes in our data (ignoring any blanks or unknowns).

```

[9]: # count and plot frequencies of patient outcomes
outcome_df = pediatric_data.reactionoutcome.value_counts()
outcome_df.sort_index(inplace=True)
labels_dict = {1 : 'Recovered/resolved',
                2 : 'Recovering/resolving',
                3 : 'Not recovered/not resolved',

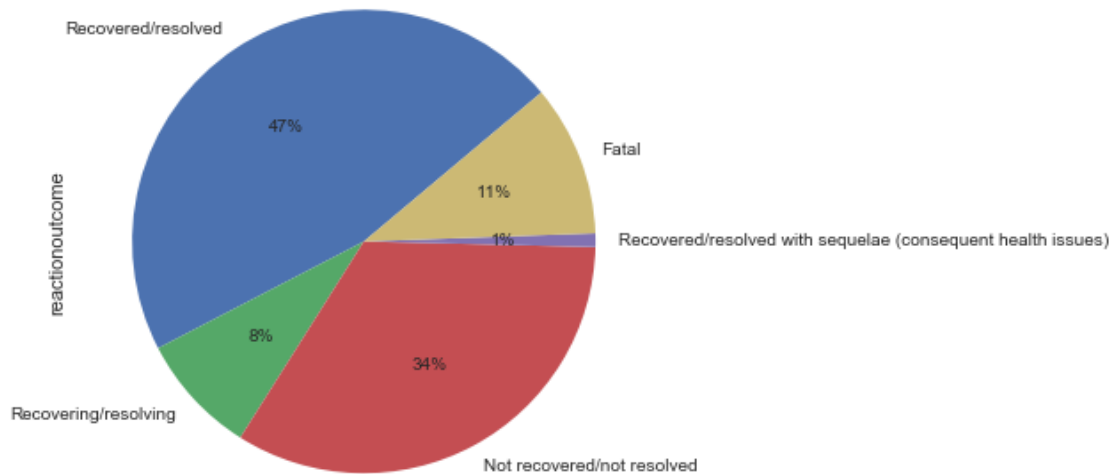
```



```

        4 : 'Recovered/resolved with sequelae (consequent health_
→issues)',
        5 : 'Fatal',
        6 : 'Unknown'
    }
ax1 = outcome_df[outcome_df.index != 6].plot.pie(autopct='%1.0f%%',
→labels=labels_dict.values(), startangle=40);

```



Some categories can be compared directly. For example the number of reports listing the outcome as 'death' is 11% by this metric compared to 8% by the feature 'seriousness outcome'. When combined with the relative sparsity of entries in the seriousness outcome fields, we might infer that this metric may be more reliable.

It is noteworthy that many reports indicate multiple serious outcomes, for example hospitalization and death are not mutually exclusive and consequently they are not mutually exclusive.

1.2.3 Exploring the impact of age on adverse response outcomes

```
[10]: pediatric_data.groupby('serious').mean()
```

```

[10]:
          index  reporttype  primarysource.qualification \
serious
0      2894.278929      1.145509                      4.163424
1      2155.871776      1.355140                      2.902292

          patient.patientsex  seriousnessother  seriousnesshospitalization \
serious

```

0	1.310559	0.000000	0.000000
1	1.350334	0.680748	0.31028

	seriousnesslifethreatening	seriousnessdeath	seriousnessdisabling \
serious			
0	0.000000	0.000000	0.000000
1	0.033271	0.090467	0.011589

	seriousnesscongenitalanomaly	reactionoutcome \
serious		
0	0.000000	4.856007
1	0.005981	4.370299

	patient.patientonsetageyear
serious	
0	8.076423
1	7.711292

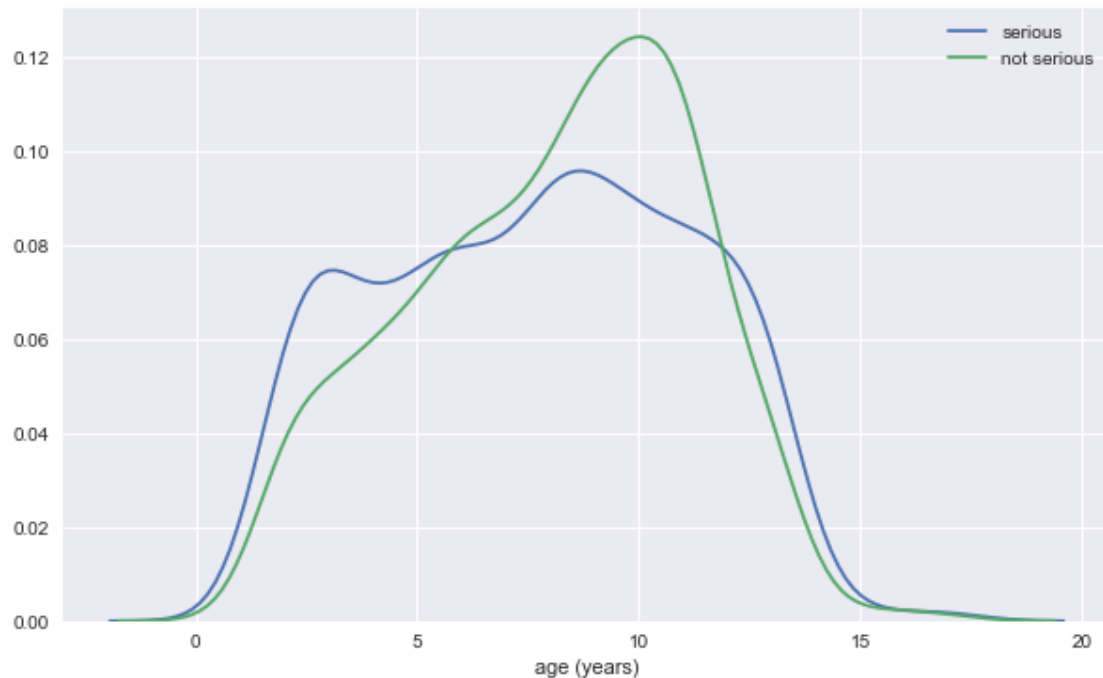
It seems that reports of serious adverse effects are more likely to be seen in slightly younger patients. We consider their relative distributions and compare them at a 95% significance level using a z-test.

```
[11]: # plot two distribution plots comparing serious and not serious distributions
fig = plt.figure()
serious_1 = pediatric_data.loc[pediatric_data.serious==1, 'patient.
    ↳patientonsetageyear']
serious_0 = pediatric_data.loc[pediatric_data.serious==0, 'patient.
    ↳patientonsetageyear']
sns.kdeplot(serious_1, label='serious');
sns.kdeplot(serious_0, label='not serious');
plt.xlabel('age (years)');

# perform z test to compare means of distributions
# at 95% significant level
z = process_data.z_test(serious_0,serious_1, 0.95)
```

z = 0.0781 critical value = 1.64

Do not reject null hypothesis. Insufficiest evidence for dependence.



Whilst the 'not serious' responses are more skewed to older children, there is not enough evidence here to reject the null hypothesis that age is a significant factor in the severity of adverse reports.

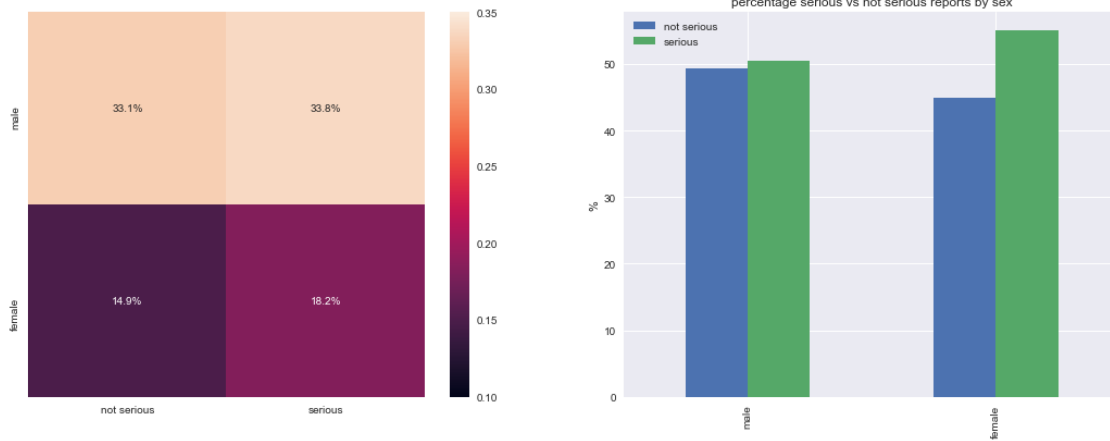
1.2.4 Exploring the impact of sex on adverse response outcomes

We can also consider the correlation with sex using a contingency table.

```
[12]: # create a pivot table on sex and serious
pivot_tb = process_data.calculate_serious_pivot(pediatric_data, 'patient.
↳patientsex')
tb = pivot_tb.drop(columns=['total', 'not serious %', 'serious %'])
tb = tb.divide(tb.sum().sum())

# plot heatmap of contingency table
fig, (ax1, ax2) = plt.subplots(nrows=1, ncols=2, figsize=(18,6.5))
ax1 = sns.heatmap(tb, vmin=0.10, vmax=0.35,
                  #norm=LogNorm(vmin=sub_data.min().min(), vmax=sub_data.max().
↳max()),
                  annot=True,
                  fmt='.1%', ax=ax1);
ax1.set_xlabel('');
ax1.set_ylabel('');
ax1.set_yticklabels(['male', 'female']);
```

```
# plot bar chart showing relative frequencies of serious/not serious by sex
process_data.plot_serious_pivot(pivot_tb, 'route', ax2)
ax2.set_title('percentage serious vs not serious reports by sex');
ax2.set_xticklabels(['male', 'female']);
ax2.set_xlabel('');
ax2.set_ylabel('%');
```



We can see that there are significantly more reports of adverse reponses in male patients in the database. This might suggest that male patients are more likely to suffer adverse responses or that they are more likely to report adverse responses than female patients (or both).

One indication that suggests male patients might simply be more likely to report adverse effects (either directly or through a clinician) is that, of female reports, 55% of reports were for serious adverse effects compared to 45% for not serious. The number of serious and non-serious adverse reports for male patients in the data is 50:50.

We can calculate pearson's χ^2 correlation coefficient to understand whether these results are significant (i.e. whether, for a given confidence level, there is a correlation between sex and severity of response). We will choose a 99% confidence interval.

```
[13]: process_data.significance_test(pivot_tb[['not serious', 'serious']], 0.99)
```

```
dof = 1  probability = 0.990 | critical = 6.635  chi2 = 7.004
```

Reject null hypothesis. Variables dependent at the 99% confidence level.

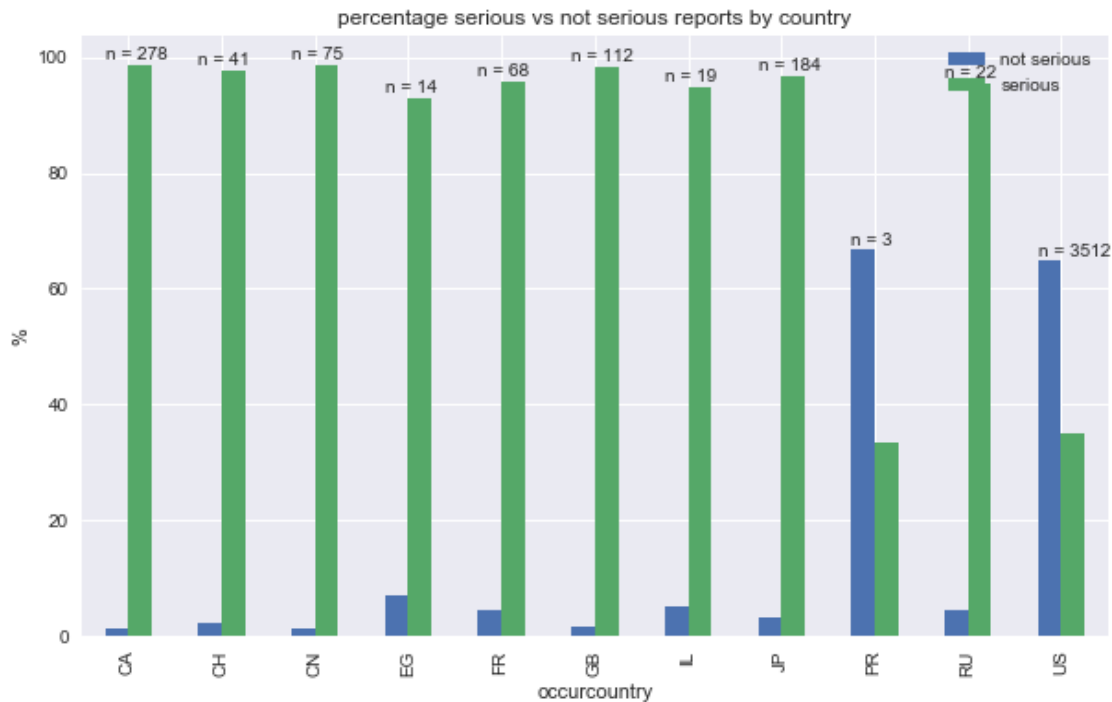
We find that there is sufficient evidence at the 99% significance level to reject the null hypothesis that there is no dependency between gender and the serious status of the report. This feature may be useful in a predictive model.

1.2.5 Exploring the impact of country on adverse response outcomes

A similar analysis is performed on the country in which the adverse response took place.

```
[14]: pivot_tb = process_data.calculate_serious_pivot(pediatric_data, 'occurcountry')
ax = process_data.plot_serious_pivot(pivot_tb, 'occurcountry', annotate=True)
ax.set_title('percentage serious vs not serious reports by country');
plt.show()

# calculate significance
process_data.significance_test(pivot_tb[['not serious','serious']], 0.99)
→#consider appropriateness of pearson's chi2 here (US dominates)
```



dof = 10 probability = 0.990 | critical = 23.209 chi2 = 1038.201

Reject null hypothesis. Variables dependent at the 99% confidence level.

We find in this case also that, at the 99% significance level, there is sufficient evidence to reject the null hypothesis (i.e. that there is no correlation). This feature may be useful in a predictive model.

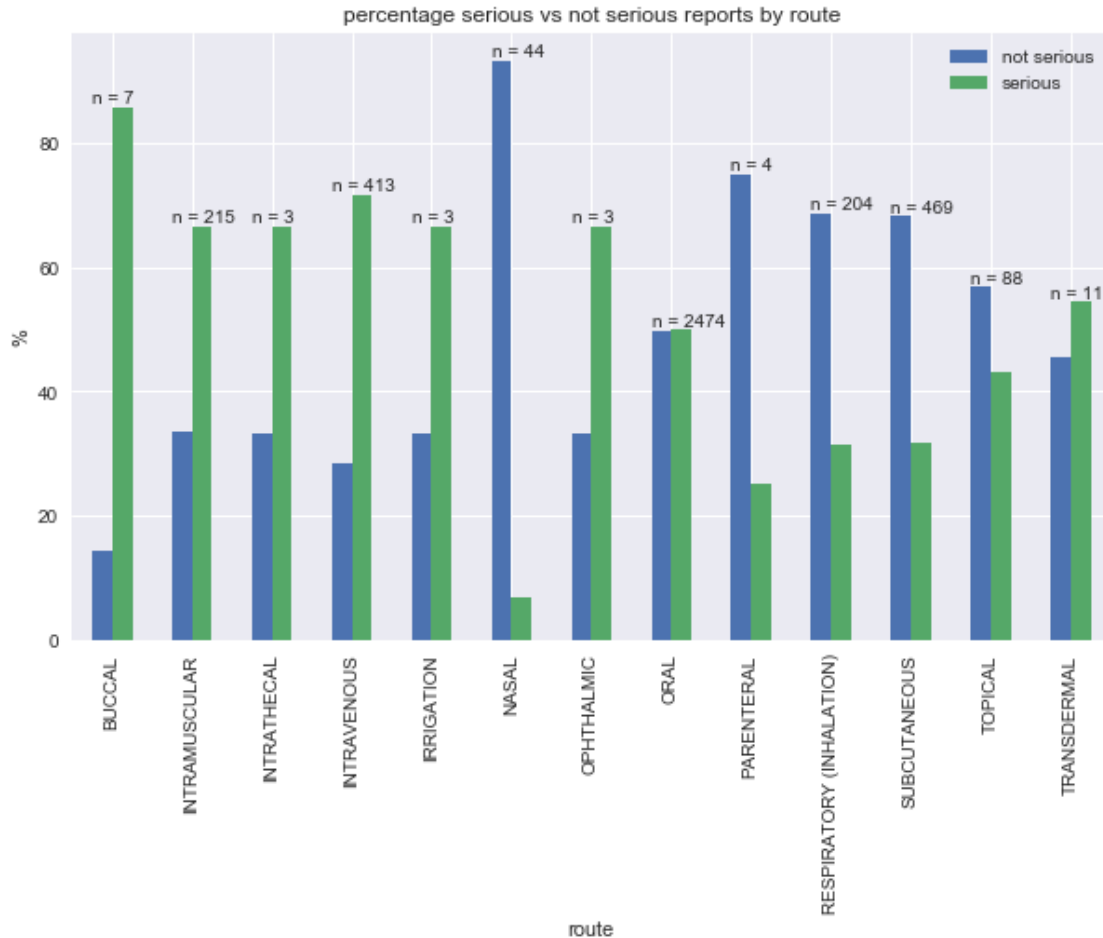
1.2.6 Exploring the impact of drug administration route on adverse response outcomes

We repeat the process again exploring the possibility that adverse response severity may be influenced by the administration route.

```
[15]: # calculate pivot table for severity against drug administration route
pivot_tb = process_data.calculate_serious_pivot(pediatric_data, 'route')
```

```
ax = process_data.plot_serious_pivot(pivot_tb, 'route', annotate=True)
ax.set_title('percentage serious vs not serious reports by route');
plt.show()

# calculate significance
process_data.significance_test(pivot_tb[['not serious','serious']], 0.99)
```



dof = 12 probability = 0.990 | critical = 26.217 chi2 = 231.544

Reject null hypothesis. Variables dependent at the 99% confidence level.

We find that, again, there is enough evidence at a 99% significance level to reject the null hypothesis.

We have discovered at least three factors that are correlated with the severity of the adverse response reports. With further time, interpretation of these correlations would benefit from comparison to general population data, a larger dataset and a more detailed understanding how and why adverse reports are made to the database.

1.3 Logistic Regression model

I am interested to construct a mode to predict the severity of adverse responses given certain information about a patient and the drug(s) that they are taking. The model would require significant fine tuning taking longer than the time available to complete this task. However, I include a representative approach using logistic regression to classify severity as serious or not serious. This simple model performs poorly but a variety of approaches might improve the accuracy of the model. I discuss these below.

```
[16]: from sklearn.impute import SimpleImputer
      from sklearn.linear_model import LogisticRegression
      from sklearn import metrics
      from sklearn.model_selection import train_test_split
      from sklearn.preprocessing import LabelEncoder
      import process_data as process
```

I will establish a simple model based only on gender, age of patient and drug administration route. Initially the data needs further cleaning and preprocessing.

Further cleaning of the data

```
[17]: # Shuffle the data in case it is sorted
      data = pediatric_data.sample(frac=1).reset_index(drop=True)

      # Remove all rows with no gender specified
      # (could have alternatively mapped to additional category)
      print(f"{data['patient.patientsex'].isna().sum()} of {data['index'].count()} "
            f"rows with missing sex removed "
            f"({data['patient.patientsex'].isna().sum()/data['index'].count():.2%})")
      data = data[data['patient.patientsex'].notna()]

      # Due to small numbers, group least prevalent routes into 'other'
      # Kepp only most prevalent routes (>10% of data)
      data = process.map_routes(data, 0.1)

      # Map routes with missing values to new category 'unknown'
      data.route_summary = data.route_summary.fillna('UNKNOWN')
```

961 of 4991 rows with missing sex removed (19.25%)

Impute missing values and construct logistic regressor

```
[18]: # Select features to model
      X = data[['patient.patientsex', 'patient.patientonsetageyear', 'route_summary']]
      y = data['serious']

      # Split into training data and validation data (consider cross-validation later)
```

```

X_train, X_valid, y_train, y_valid = train_test_split(X, y, test_size=0.3,
↳random_state=0)

# Imputation: list categorical and numerical columns
cols_cat = ['route_summary']
patient_age = 'patient.patientonsetageyear'
imputed_X_train = X_train.copy()
imputed_X_valid = X_valid.copy()

# Apply label encoder to each column with categorical data
label_encoder = LabelEncoder()
for col in cols_cat:
    imputed_X_train[col] = label_encoder.fit_transform(X_train[col].astype(str))
    imputed_X_valid[col] = label_encoder.transform(X_valid[col].astype(str))

# Imputation: replace missing numerical data with mean value
process.impute_on_mean(imputed_X_train, 'patient.patientonsetageyear')
process.impute_on_mean(imputed_X_valid, 'patient.patientonsetageyear')

# Imputation removed column names; put them back
imputed_X_train.columns = X_train.columns
imputed_X_valid.columns = X_valid.columns

# define logistic regression model
log_reg = LogisticRegression();
log_reg.fit(imputed_X_train, y_train);

```

Determine accuracy of model

```

[19]: # predict
y_pred = log_reg.predict(imputed_X_valid)
print('Accuracy of logistic regression classifier on validation set: {:.2f}'.
↳format(log_reg.score(imputed_X_valid, y_valid)))
confusion_matrix = metrics.confusion_matrix(y_valid, y_pred)
fig, (ax1, ax2) = plt.subplots(nrows=1,ncols=2,figsize=(15,5))
sns.heatmap(confusion_matrix/np.sum(confusion_matrix),annot=True,fmt='
↳1%',ax=ax1);
ax1.set_xlabel('predicted');
ax1.set_ylabel('true');
ax1.xaxis.tick_top()
ax1.xaxis.set_label_position('top');

#plot ROC curve
fpr, tpr, _ = metrics.roc_curve(y_valid, y_pred)
ax2.plot(fpr, tpr);
ax2.set_xlabel('false positive rate')

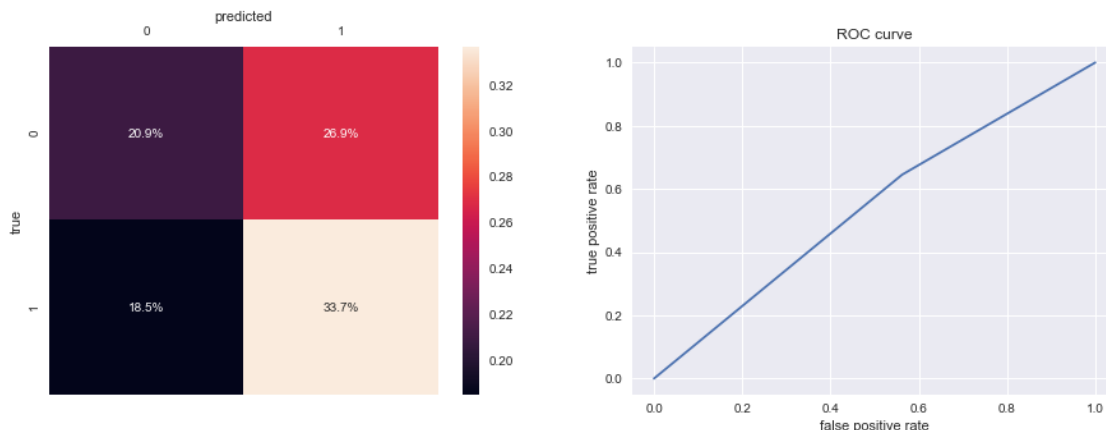
```



```
ax2.set_ylabel('true positive rate')
ax2.set_title('ROC curve')
plt.show()

print(metrics.classification_report(y_valid, y_pred))
```

Accuracy of logistic regression classifier on validation set: 0.55



	precision	recall	f1-score	support
0	0.53	0.44	0.48	578
1	0.56	0.65	0.60	631
accuracy			0.55	1209
macro avg	0.54	0.54	0.54	1209
weighted avg	0.54	0.55	0.54	1209

The accuracy is extremely poor for this model (little better than random). There are likely many reasons for this and I list possible next steps to improve the model below.

Next steps to improve the model

- Increase the volume of data
- Increase the number of features (e.g. country, drug type, pediatric indication, prescription/OTC)
- Balance the dataset
- Reintroduce less common administration routes
- Alter encoding (use one-hot encoding)
- Cross-validation
- Alternative approaches e.g random forest or XGBoost

Additionally, I would build the code into a pipeline rather than having each step as laid out above.

1.4 Conclusions

I have explored the FDA Adverse Reaction System and summarized the data it contains in [the first notebook](#). Simple frequency plots and calculations are performed to understand where reports comes from, the number of reports over time, the gender breakdown of reports and the severity of the adverse responses reported.

In a second stage, I have pulled records directly from the data to create a dataset of 5000 reports of adverse responses in the patient age group category 'children'. Using these data, I have examined the correlation of severity of response with various other reported features including age, gender, drug administration route and country. In all cases except age, a statistically significant correlation is observed. The analysis could be improved by ensuring balanced samples and accounting for variations within the population at large. A larger dataset would also produce more reliable results. Given the age range found in the 'child' category, it would likely be sensible to include all pediatric patient age ranges.

In a final stage, I have constructed a illustrative logistic regression model to predict a response as 'serious' or 'not serious' as defined by the FDA. This model is overly simplistic and required further refinement both for reliability and for accuracy.