Pearson has given an approximate formula for the skewness that is easier to calculate than the exact formula ($skew = \frac{3(mean - median)}{standard\ deviation}$). In the following report, it is in the best interest of the author to check its validity and determine if Pearson's approximation gives a reasonable estimation.

The Data:

The data file contains is an examination results for a class of 119 students pursuing a computing degree are given (www.wiley.com/go/Horgan/probabilitywithr2e) as a text file called results.txt.)

The data can be located on the website www.wiley.com/go/Horgan/probabilitywithr2e containing a results.txt. A txt file that was converted by the author to csv file (results.csv) for convenience in reading and anlyzing the said data. The csv file contains the examination results for a class of 119 pursuing a computing degree.

gender	arch1	prog1	arch2	prog2
m	99	98	83	94
m	NA	NA	86	77
m	97	97	92	93
m	99	97	95	96
m	89	92	86	94
m	91	97	91	97
m	100	88	96	85
f	86	82	89	87
m	89	88	65	84
m	85	90	83	85

The data contained 5 columns: gender, arch1, prog1, arch2, and prog2. As the name said, the gender contains the gender ('f' or 'm') of the student. The rest of the columns contains integer values ranging from 3-100 that served as the scores of the students in their a particular subject (indicated in their column name).

It is also worth noting that there appears to be NA values in some of the entry in the csv file. Such data were ommited during the calculation process. This does not however affect the results in any way.

Methodology:

The author uses RStudio and R language to calculate and graph the whole process.

The author began by calculating the mean, median, and standard deviation (sd) for each of the column.

```
results <- read.csv("results.csv", header = TRUE)
meanResults <- sapply(results, mean, na.rm = TRUE)
medianResults <- sapply(results, median, na.rm =
TRUE) |> as.numeric()
sdResults <- sapply(results, sd, na.rm = TRUE)
```

The result:

	Mean	Median	SD
arch1	63.56897	68.5	24.37469
prog1	59.01709	64.0	23.24012
arch2	51.97391	48.0	21.99061
prog2	53.78378	57.0	27.08082

Then the author wrote a function for calculating skewness using a Pearson's Approximation.

```
skewnessCalc <- function(columnNamesVar, meanDf,
medianDf, sdDf) {
        skewResults <-
                setNames(numeric(length(columnNames
                Var)), columnNamesVar)
        for (columnName in columnNamesVar) {
                skewness <- (3 * (meanDf[columnName]
                        - medianDf[columnName])) /
                        sdDf[columnName]
                skewResults[columnName] <- skewness
         }
        return(data.frame(Skewness = skewResults))
}
```

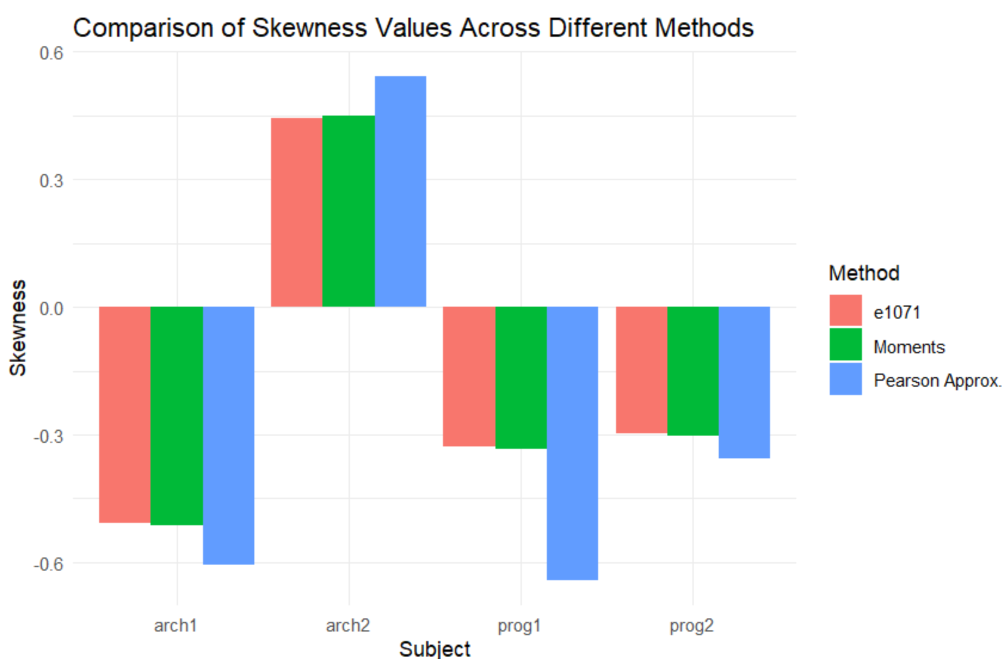
The result:

Subject	Mean	Median	SD	SkewnessPearson
arch1	63.56897	68.5	24.37469	-0.6069042
prog1	59.01709	64.0	23.24012	-0.6432290
arch2	51.97391	48.0	21.99061	0.5421286
prog2	53.78378	57.0	27.08082	-0.3562908

Then to calculate the validity of the results, the author imported and used two libraries that has a skewness calculator method. The two libraries are "e1071" and "moments."

Subject	Mean	Median	SD	SkewnessPearson	SkewnessMoments	SkewnessE1071
arch1	63.56897	68.5	24.37469	-0.6069042	-0.5129462	-0.5063276
prog1	59.01709	64.0	23.24012	-0.6432290	-0.3334265	-0.3291610
arch2	51.97391	48.0	21.99061	0.5421286	0.4481600	0.4423272
prog2	53.78378	57.0	27.08082	-0.3562908	-0.3018269	-0.2977574

Using "ggplot2" from "Tidyverse" package, the author used bar graph to visualize the discrepancies between all methods—or the lack thereof.



Interpretation:

Analysing the skewness given by different methods it could be observed that most of them gives an indication of fairly symmetrical dataset. As a rule of thumb, skewness between -0.5 to 0.5 means that the data are fairly symmetrical while between -1 and -0.5 or 1 and 0.5 gives an impression that the data are moderately skewed. It is worth noting that negative symbol in skewness means that the distribution of data is longer on the left side of its peak and vice-versa. With the exception of "arch1", when the average skewness was calculated, all columns fall under fairly skewed—and not to mention the direction of each variable are the same for all methods.

Subject	SkewnessPearson	SkewnessMoments	SkewnessE1071	Average Skewness
arch1	-0.6069042	-0.5129462	-0.5063276	-0.5420594
prog1	-0.6432290	-0.3334265	-0.3291610	-0.4352722
arch2	0.5421286	0.4481600	0.4423272	0.4775386
prog2	-0.3562908	-0.3018269	-0.2977574	-0.3186250

The Pearson skewness approximation for "arch1," "prog2," and "arch2" are relatively close to the values obtained from the "Moments" and "e1071" packages. This suggests that it is indeed reasonable when it comes to the said subjects/column.

However, the disparaty becomes apparent when it comes to the subject of "prog1" for it almost double the skewness from the imported packages. The large discrepancy suggests that Pearson's approximation might not be as reasonable for this variable. Not this begs the question as why that's the case—but that could explored further with using other statistical methods and assumptions—which are all outside of the scope of this report.

Nevertheless, in general, the magnitude and the direction of the skewness from Pearson's approximation matched those from "moments" and "e1071" library therefore indicating that it is indeed reasonable—but one should be aware of the assumptions and when to not use the said approximation due to the instances such as those data under "prog1" column.