

## The importance of understanding web frameworks for statisticians

R B Fajriya Hakim<sup>1\*</sup>, Sugiyarto Sugiyarto<sup>2</sup>

<sup>1</sup>Universitas Islam Indonesia, Jl. Kaliurang, Umbulmartani, Sleman, Indonesia

<sup>2</sup>Universitas Ahmad Dahlan, Jl. Jend. Ahmad Yani, Tamanan, Banguntapan, Bantul, Indonesia

\*Corresponding e-mail: [hakimf@uui.ac.id](mailto:hakimf@uui.ac.id)

### ARTICLE INFO

#### Article History

Received 22 April 2021

Revised 22 November 2021

Accepted 22 November 2021

#### Keywords

Web application;

Web frameworks;

Flask;

Python

#### How to cite this article:

Hakim, R. B. F., & Sugiyarto, S.

(2021). The important of

understanding web

frameworks for statisticians.

*Bulletin of Applied Mathematics*

*and Mathematics Education*,

1(2), 47-60.

### ABSTRACT

As the web and computational technology carry on growing and huge data are yielded on the web, these technologies are turn into important for a statistician's work. It is worthy that statistician always gain knowledge of new aspects of computation. A lack of computational reasoning skills gets it hard for statisticians to work in a team. If statistician do not take up this computations challenge more coherently, statistics will be marginalized and take away related at a time when its data science reputation grow up significantly. In addition, people rely on the information on web, for whatever their reason. Since web growth, several major transforms have evolved, from the most rudimentary concept until a new model of interaction between humans and machines. Simple interactivity denotes that users can enter data to the application on a web page, then click on button, and then appears a new web page with the results of the computations. This application has been known as web application with most are built with the utility of web frameworks which is a package of programming tasks that offering services through the Internet. Therefore, this paper gives short overview the importance of Flask web frameworks to assist the lack of computational skill of statistician over web application in the simplest possible way and how web framework is used to create a web page with application form, run the application to compute statistical calculation which has been deployed in local server, and produce a web page with the solutions.

This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## Introduction

Statistics a division of science that arrange the collection, organization, analysis, interpretation and presentation of data to draw an inference of samples to the whole population. This involves an appropriate design of the study, a proper assortment of the study sample and select of a suitable statistical test (Ali & Bhaskar, 2016). A necessary knowledge of statistics is required for proper designing of any research conducted. Inappropriate statistical methods may result in specious conclusions which may initiate wrong decision. Thereby, a statistical method comprise a study include planning, designing, collecting, analyzing, interpreting data and

visualizing the research findings. The statistical analysis delivers value to the valueless data, so alive life into a lifeless number. The findings and conclusions are clear-cut only if suitable statistical tests are utilized.

By tradition, statistics is about collecting data and calculating what the numbers can express. The earliest farmer guessing whether he had an adequate amount of grain to last the winter. Statistical instruments like the mean to quantify average summaries data, standard deviations to assess how much variation in a set of numbers and frequency distributions to show the shapes of the numbers when drawn on a graph be able to aid estimating future events. So far, people have always been getting inferences from data. Statistics has an important role as the science of learning from data. By means of its discipline which concerned with the collection, analysis, interpretation of data, and the operative communication and presentation of research findings relying on data to make a proper action for the future strategies. Statistics stay at the core of any type of quantitative way of thinking essential for making important progress in the engineering, natural, and social sciences, and for reaching valuable decisions in business, industry and public policy. At present, enormous volumes of data, everyone is currently mentioning to big data (Mauro et al., 2016), are changing the world and the manner we live in it. Statistical methods and theories are utilized everywhere, from industry, business, science, health, entertainment, controlling traffic road and exploring sustainability and climate change. This will initiate the demand for a much closer relationship between statisticians, mathematicians, computer scientists and domain scientists. Progressively, statistical assessment of databases manipulates the world, Google figure out who we are, Amazon and a lot of shopping online platform estimates what we need to buy, through our search queries. This demand pronounces the outlook that the challenges of the big data and new technologies can only be met by a new subject called data science (Yan & Davis, 2019) . The demand for a modern generation of statisticians as a data scientist performing at this line is turn into more and more obvious; there is a solid need to build data-science university curricula. The McKinsey Global Institute is assuming a lack of data scientists by future coming (Manyika & Roxburgh, 2011). Actually, this conditions as marks a call for extending competencies in statistics.

It is a certainty that almost all new technologies are not plausible without the field of statistics and its attainments. For example, web services, mobile phones and smart technologies analysis are using statistics for improvement with the familiar features broaden to massively big data-sets frequently with a spatial and/or time factor, data streams in real time, high complexity but also weakly structured or unstructured data, low dimensionality with apparently infinite observation units or high dimensionality with many more undetermined parameters than observation units. At this point, traditional statistical concepts are no longer applicable. Unquestionably, new methods for computational technologies should be understood by statisticians are necessitated. Yet supporting statistical investigation has already laid the basics upon which data sciences move toward may be established. As the web carry on growing and huge of data are yielded, these web technologies are turn into important for a statistician's work. It is worthy that statistician improve a deep basis on computing rudiments so and programming tasks and always learn new concepts of computation. The capacity to deliver computations is a critical skill for statisticians that enhance the concept of engineering data and analytical model deployment. Therefore, in this paper we made a research overview which overcome the lack of understanding from statistician perspective, how web application technology using web framework can be used to create a web page to gather data then run the application to compute simple statistical calculation which has been deployed and produce the result in a web page.

## Method

In the present day, for all news and figures people rely on the information on website, for whatever their reason. Making an online existence is one of the most significant and cleverest choice that company has occupied and gaining the outcomes. Acquiring from this, company has reached online presence to share information, to instruct on products and its uses, to seize market and to build credibility (Gandhi et al., n.d.). This mean that making a website is important and can be supported by other online presence through social media, blogs, etc. (Harris & Rae, 2009). Holding a well maintain web existence will assist persuade customer to come and visit again and again. A well-designed website with high usability (Garett et al., 2019) has been obtained to positively stimulus visitor retention (revisit rates) and buying manners (Nikolaos et al., 2003). Since its growth, several major transforms have evolved, from the most rudimentary concept, the first stage of the Web, which is characterized by static content printing, in which users only get through the content arranged by companies. At that moment, communication between the client and a company has not bidirectional side and it is known as low interactivity. Companies used the Internet as another possibility way to broadcast content and news, without upsetting much about customers' opinions, who acted as an inactive observer of information. The evolution of Web with the exponential growth of social networks, create a massive quantity of information about the users. This information is a benefit for personalization of marketing and business strategies in the most various segments of activity. Then, the growth offers a new model of interaction that is completer and more dynamic in which there is a symbiotic interaction between humans and machines (Choudhury, 2014). Interactivity on websites rises significant consumer quantities such as website liking and take in entertainment (Krishen, 2013). The important parts of how to design a web that could ask a consumer entering data in the forms on web page and processing the data in autonomous statistical calculation embedded in the web and give the results directly seems do not involving the role of statistician. As the Web carry on growing and attractive data are yielded existing in rich ways on the Web (Almeida & Monteiro, 2017), these technologies are turn into important for a statistician's work. It is worthy that statistician improve a deep basis on computing rudiments so that they can reason about computation and programming tasks and always gain knowledge of new aspects of computation. Moreover, a lack of computational reasoning skills gets it hard for statisticians to work in a team where others are computationally skilful, self-supporting and autonomous. The capacity to deliver computations is a critical skill for statisticians that improves on aspects of engineering data and analytical model deployment. Furthermore, as data and its analysis more complex, novel computational approaches are expected to create computational technologies change intractable problems be feasible. The ability to express real world problem to the computational perspective is not the finale-goal, but it is a necessary viewpoint of the whole task which involved in and strengthens all facets of a statistician's job. If statistician do not take up this computations challenge more coherently, statistics will be marginalized and take away related at a time when its reputation grow up significantly. To handle this big data, statistical challenge in big data methods and theories are utilized everywhere. The analysis of big data can be tackled in numerous ways, but the main problem is again a statistical difficulty. Become a skill in handle big data are statistical methods that take a broad view and refashion the classical techniques for the new data sets. It is worthless to seem with scepticism these huge data and the techniques that are used, because this will be the major relevance discipline of the statistical analysis in the future. There is no essential to discern big data analytics from data analytics, as data will remain expanding, and it will certainly not be small again. Commonly, the main difficulty in the statistical analysis of big data is the frequent lack of statistical

representatively of the observed data due to no involvement of statistician in component to design collecting the big data (Di Ciaccio & Giorgi, 2016). Advances in statistical theory and computational power, the collection and availability of extensive data sets, and the increasing complexity of data have generated unparalleled opportunities and provide an exciting and dynamic context for statisticians. The environment of government, academic, and industry sectors is increasingly data dependent, and the professional future of statistics is potentially unlimited asked whether statisticians are losing out to computer scientists in the big data revolution (Nair, 2013).

In the big data era, the field of data science as an arising discipline addressed on statistics, machine learning, computer science, mathematics and requires the domain sciences. The data science movement is a good motivation that researchers from different field of study come together to solve a project and able to compare, contrast and connect ideas from multiple disciplines (West, 2016). Based on research study (Cameron et al., 2017) statisticians did not seem threatened by data science and they indeed most were open to new approaches and saw value in statisticians developing data science skills. Critical thinking of statistician is about how to collect the data from web page made to meet precise business questions or to critically measure the worth of the data that is already available. Their research also found a variety of sentiments on marginalisation. Some respondents had felt marginalised by data science and associated techniques and statisticians must not be involved with big data wherever it is placed and felt difficulty in accessing a big dataset from IT specialists within their own department. Paper from (Hamed Hijazi, 2014) revealed that the data eruption is opening up new frontiers for statisticians to cope with and analyse the huge troves of data. The shortfall in statistical workforce production is a worldwide problem. Both developed and developing countries face an acute shortage of skilled statisticians and in particular official statisticians to cope with big data.

The lack of statistician's skill on how to deploy a statistical model in a web system make the design of web page to collect consumer data, analysing it in a statistical model and presenting it in a web look like a part in a black and white step. As if the involvement of statistician competence is only in the part of analysing a tidy dataset which is given by database administrator. Actually it is the pipeline of a system from beginning until data ends which denotes a user can enter data to the application on a web page, then click on button, and then appears a new web page with the results of the computations, that is pipeline is a simple algorithm that allows perform a sequence of different transformations [Ertan Mustafa Geldiev, Nayden Valkov Nenkov, Mariana Mateeva Petrova, 2018]. It shows that the role of statisticians to understand a pipeline of how data coming until computation give a result and display it to web page is a must. Recently, this application has been offered by computational scientist through a web interface and it was known as web application.

The web interface can be used as a graphics user interface and deployed to a server and ready accessible to the whole customer through Internet or could be used locally for improving the web design on the local server of scientist's computer. Web applications can be built from scratch by means of Common Gateway Interface (CGI) scripts in (e.g.) Python [Sheetal Taneja & Pratibha R. Gupta, 2014] which specifies a standard way in which information may be sent to and from the browser and server, but the code become longer and more sophisticated as the web interface grows more complex. Currently, utmost web applications are built with the utility of web frameworks (Doroodchi & Dastgheib, 2008), which are software packages that make easier the programming and without an overwhelming amount of programming tasks that offering services through the Internet (Kavya & Sarathambekai, 2019). This paper will explore a web framework which has high-level and easier-to-use framework **Flask** (Grinberg, 2014) as an important tool for

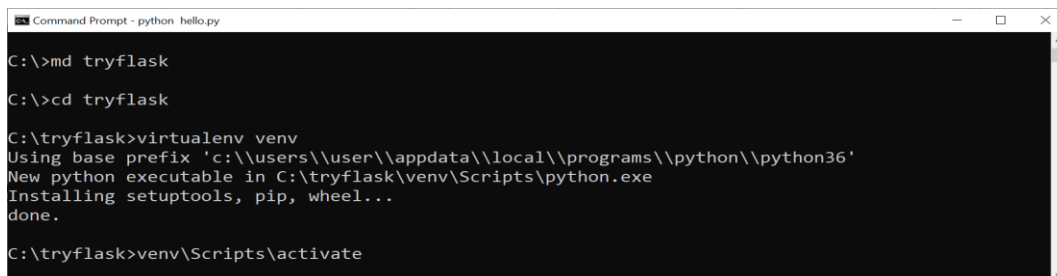
statistician to understand data interactivity between user and company through website. This paper also give simple instances on how Flask can be used to enable scientific applications for the web and this can be a challenge for a statistician who wants to enable statistical computation through the web and deploy it in a local server than utilizing web application only on counting web visitor or polls. Therefore, we made a short research overview which overcome the lack of understanding from statistician perspective, in the simplest possible style, how web frameworks can be used to create a web page with entering data into application form, run the application to compute simple statistical calculation which has been deployed it in web, and produce a web page with the solutions.

Because our aim was to investigate an important aspect of web application using web framework on the issues identified in marginalization of statistician in computation and web technologies, we prepared a step by step pipeline in the using of simple example of implementation web framework and each step identified an opportunity as a key competence topics and give with a brief summary of relevant statistical competence or skill and any initiatives that statisticians would like to address further on this issue.

## How web framework works

### *Initialization of web frameworks*

The most appropriate practice to install Flask is to use a virtual environment. A virtual environment is a private copy of the Python interpreter without disturbing other global Python interpreter installed in the system. Virtual environments are very helpful since they prevent package disarrange and version conflicts in the system's Python interpreter (Grinberg, 2014). Designing a virtual environment for every single application make sure that applications have access to only the packages used, whereas the global interpreter keep on tidy and orderly and serves only as a source from which more virtual environments can be designed. To create the Python virtual environment is using the "virtualenv" command (using Microsoft Windows). This command has a single required name of the virtual environment and all files assorted with the virtual environment will be contained by that name. A frequently used name for virtual environments is "venv" (See Figure 1).



```
Command Prompt - python hello.py
C:\>md tryflask
C:\>cd tryflask
C:\tryflask>virtualenv venv
Using base prefix 'c:\users\user\appdata\local\programs\python\python36'
New python executable in C:\tryflask\venv\Scripts\python.exe
Installing setuptools, pip, wheel...
done.
C:\tryflask>venv\Scripts\activate
```

**Figure 1.** Create the python virtual environment inside the 'tryflask' folder.

Once a virtual environment is activated, the position of its Python interpreter is added to the PATH (See Figure 2). The activation command adjusts the command prompt to add in the name of the environment.



```

Command Prompt
C:\tryflask>venv\Scripts\activate
(venv) C:\tryflask>
    
```

**Figure 2.** Environment PATH of venv.

*Installing python packages with pip*

Most Python packages are installed with the *pip* utility, since a virtual environment is activated, the location of the pip utility is added to the PATH, and use the following command to install Flask.

```
(venv) C:\tryflask> pip install flask
```

Then, Flask and its dependencies are installed in the virtual environment (See Figure 3) and could be verified that Flask was installed properly by opening the Python interpreter and attempting to import it.

```
(venv)      C:\tryflask>      python
>>>                               import      flask
>>>
```

```

Command Prompt - python hello.py
(venv) C:\tryflask>pip install flask
Collecting flask
  Downloading https://files.pythonhosted.org/packages/f2/28/2a03252dfb9ebf377f40fba6a7841b47083260bf8bd8e737b0c6952df83f/Flask-1.1.2-py2.py3-none-any.whl (94kB)
    |-----| 102kB 93kB/s
Collecting Jinja2>=2.10.1 (from flask)
  Downloading https://files.pythonhosted.org/packages/30/9e/f663a2aa66a09d838042ae1a2c5659828bb9b41ea3a6efa20a20fd92b121/Jinja2-2.11.2-py2.py3-none-any.whl (125kB)
    |-----| 133kB 32kB/s
Collecting itsdangerous>=0.24 (from flask)
  Using cached https://files.pythonhosted.org/packages/76/ae/44b03b253d6fade317f32c24d100b3b35c2239807046a4c953c7b89fa49e/itsdangerous-1.1.0-py2.py3-none-any.whl
Collecting Werkzeug>=0.15 (from flask)
  Downloading https://files.pythonhosted.org/packages/cc/94/5f7079a0e0bd6863ef8f1da638721e9da21e5bacee597595b318f71d62e/Werkzeug-1.0.1-py2.py3-none-any.whl (298kB)
    |-----| 307kB 42kB/s
Collecting click>=5.1 (from flask)
  Downloading https://files.pythonhosted.org/packages/dd/c0/4d8f43a9b16e289f36478422031b8a63b54b6ac3b1ba605d602f10dd54d6/click-7.1.1-py2.py3-none-any.whl (82kB)
    |-----| 92kB 84kB/s
Collecting MarkupSafe>=0.23 (from Jinja2>=2.10.1->flask)
  Using cached https://files.pythonhosted.org/packages/b9/82/833c7714951bff8f502ed054e6fbd8bd0e083d1fd96de6a46905cf23378/MarkupSafe-1.1.1-cp36-cp36m-win_amd64.whl
Installing collected packages: MarkupSafe, Jinja2, itsdangerous, Werkzeug, click, flask
Successfully installed Jinja2-2.11.2 MarkupSafe-1.1.1 Werkzeug-1.0.1 click-7.1.1 flask-1.1.2 itsdangerous-1.1.0
    
```

**Figure 3.** Installing flask and its dependencies.

The installation is success since there are no errors happen. Once installing Flask these distributions will be installed automatically, Werkzeug implements WSGI (Web Server Gateway Interface), the standard Python interface between applications and servers. Jinja2 is a web templating system that combines a template language with a certain data source to render dynamic web pages of application serves. MarkupSafe comes with Jinja2 which is used to escapes

untrusted input when rendering templates to avoid injection attacks. Its `Dangerous` securely signs data to ensure its integrity to protect Flask's session cookie. Click is a framework for writing command line applications and allows adding custom management commands. The entire Flask applications must set up an *application instance*. The application instance is an object of class `Flask` and built in three parts, commonly formed in the first part as an initialization is in these lines.

```
from flask import Flask
app = Flask(__name__)
```

The web server issues all requests it receives from clients to this object for running, using a protocol called Web Server Gateway Interface (WSGI). The single necessary argument to the `Flask` class constructor is the name of the main module or package of the application. Designed for a large amount application, Python's `__name__` variable is the right value.

### *Routes and view functions*

The second part is routes and view functions. Clients such as web browsers or form which users could input the data on the web, post *requests* to the web server and in sequence sends them to the Flask application instance. The application instance wants to recognize what code needs to run for each URL requested, hence it manages a mapping of URLs to Python functions. The connection linking a URL and the function that manages it is called a *route*. To decorate a route in a Flask application is through the `app.route` decorator defined by the application instance, which indices the decorated function defined as a route. The next illustrates how a route is stated by means of decorator.

```
@app.route('/')
def index():
    return '<h1>Hello World!</h1>'
```

This illustration put the function `index()` as the handler for root URL application. Function `index()` usually named as a view function (motivated by the standard `index.html` as the main page of a web app). Since this application was deployed on a server connected with the URL domain name, browsing this domain name would prompt the function `index()` to run on the server. The web browser's user will display a *response* as a return value of the function.

### *Server startup*

The third part of the web application is a running method to start up the server. A run method (`app.run()`) is a method of application instance that will launch Flask's integrated development web server.

```
if __name__ == '__main__':
    app.run(debug=True)
```

The `__name__ == '__main__'` is employed to make certain that development web server is just started up since the script is run promptly and then it turns into a loop that expect a request and services them back continuously and will stop if this application is closed. The `app.run()` method of `Flask` class runs the application on the local development server. Configuration of the mode of operation could be given into `app.run()`, enabling the mode with `debug=True` will be convenient during development where the server will be reloaded itself for each change in the code.

### A complete application

The complete hello.py application script is three chunks explained formerly (initialization, routes and view function, server startup) and joined in a single file. The application is presented in Example 3.1

#### Example 3-1. hello.py: A complete Flask application

```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def index():
    return '<h1>Hello World!</h1>'

if __name__ == '__main__':
    app.run(debug=True)
```

Here is the view of source code (See Figure 4)



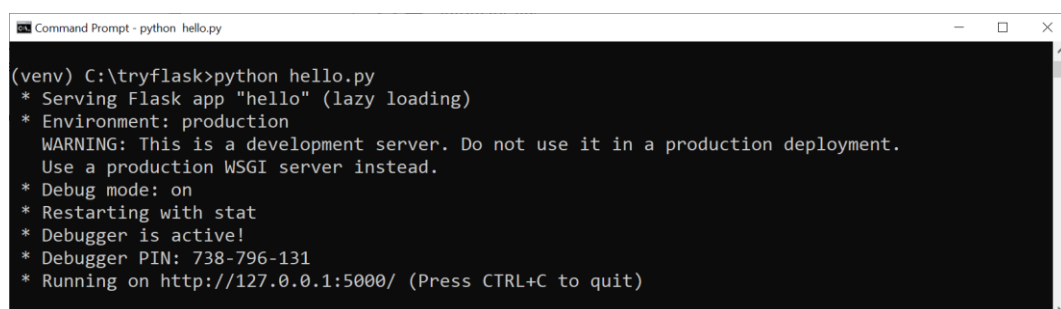
```
hello - Notepad
File Edit Format View Help
from flask import Flask
app = Flask(__name__)

@app.route('/')
def index():
    return '<h1>Hello World!</h1>'

if __name__ == '__main__':
    app.run(debug=True)
```

**Figure 4.** Source code of hello.py.

Running the application by activating virtual environment (See Figure 5) and Flask that has been already installed then open the web browser and navigate <http://127.0.0.1:5000/>.

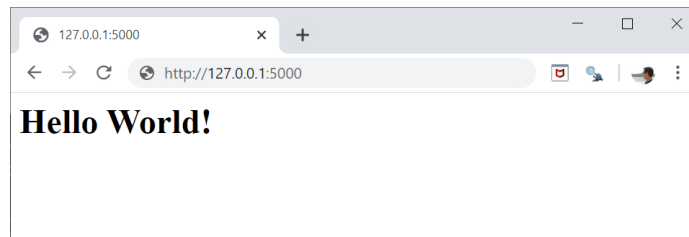


```
Command Prompt - python hello.py
(venv) C:\tryflask>python hello.py
* Serving Flask app "hello" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 738-796-131
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

**Figure 5.** Running hello.py.

Figure 6 display the web browser once connect to the application.





**Figure 6.** Display the result of hello.py

### Responses

Since Flask evokes a view function, it counts on its return value to be the response to the request. In general, the response is a plain string that is proceed back to the client as an HTML page, however, HTTP protocol expects over a string as a response to a request. Flask by default adjusts to 200 as the status code that shows the requests was executed successfully (See Figure 7).

```

Command Prompt - python hello.py
(venv) C:\tryflask>python hello.py
* Serving Flask app "hello" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 738-796-131
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [16/Apr/2020 05:55:02] "[37mGET / HTTP/1.1[0m" 200 -
  
```

**Figure 7.** Status code 200 shows executed successfully.

Since the application are done working successfully with the virtual environment, typing deactivate at the command prompt will return to the global Python interpreter (See Figure 8).

```

Command Prompt
(venv) C:\tryflask>python hello.py
* Serving Flask app "hello" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 738-796-131
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [16/Apr/2020 05:55:02] "[37mGET / HTTP/1.1[0m" 200 -

(venv) C:\tryflask>deactivate
C:\tryflask>
  
```

**Figure 8.** Deactivate python environment.

If the web page will be visually appealing and shows something more interesting, incorporating HTML files along Python file and have Python code to return it would give more fascinating HTML pages than plain strings, because returning plain strings from a function do not bring web page attractive.

The following application is to create a web interface for simple statistical program such that we can fill up data in the form and just click a compute button then obtain a new web page

with the output of a result computation. Prior to creating a web application, we first build input data module and calculate module. The `inputdata.py` is the module for entering data ( $x_1, x_2, x_3, x_4$ ) in the form of web page. The calculate module determines a function calculate ( $x_1, x_2, x_3, x_4$ ) that executes mean and variation of the data. The input data is put in file called `inputdata.py`, Statisticians must understand that this part is crucial due to define what kind of interface that will satisfy the need and services of customer and kind of data that will be generated and make any form for customer entering the data. In this part, extending the simple Flask web application with HTML form elements to gather data from a user using the WTForms library (<https://wtforms.readthedocs.org/en/latest/>), which can be installed via pip using `pip install wtforms`.

```
from wtforms import Form, FloatField, validators

class InputForm(Form):
    x1 = FloatField(label='first data', default=0,
                    validators=[validators.InputRequired()])
    x2 = FloatField(label='second data', default=0,
                    validators=[validators.InputRequired()])
    x3 = FloatField(label='third data', default=0,
                    validators=[validators.InputRequired()])
    x4 = FloatField(label='fourth data', default=0,
                    validators=[validators.InputRequired()])
```

The data entered by user in the web forms will be analysed and calculated the mean and its variance. The calculation code in this part could be extended to any machine learning programs which could handle streams data that could not be handle manually. The calculate function is put in file called `calculate.py`

```
from math import sqrt
import matplotlib.pyplot as plt
import numpy as np

def calculate(x1,x2,x3,x4):
    return calculatetestat(x1,x2,x3,x4)

def calculatetestat(x1,x2,x3,x4):
    y= (x1+x2+x3+x4)/4
    v1 = (x1-y)**2
    v2 = (x2-y)**2
    v3 = (x3-y)**2
    v4 = (x4-y)**2
    v5 = v1+v2+v3+v4
    vtot = sqrt(v5)/3
    X = ['X1', 'X2', 'X3', 'X4']
    valX=[x1, x2, x3, x4]
    XX=range(len(X))
    plt.bar(X, valX)
    plt.xticks(XX, X)
    plt.title('Mean=%g, Variance=%g' % (y, vtot))
    plt.show()
    return (y, 'and', round((vtot), 3))
```

This part is essential in the whole code for web application due to this calculation process could be included simple calculation (like mean and variation of data above, or any other descriptive statistics and show a visual of data using bar graph) to complex calculation using machine learning to process big data that could not be perform manually and would be wasting time if do not use machine learning to process the data analytics in streaming data. This step also could be included any statistical model like regression, clustering or classification techniques

which has been built from historical customer data in the company and will give the prediction result for any data inserted. Also, user would not want to spend much more time to wait the result of any data that inputted in the web page since there are no statistical calculation to give the result directly.

The whole file is brief and displayed below and save it with the name 'trycalculatestat.py'

```
from inputdata import InputForm
from flask import Flask, render_template, request
from calculate import calculate

app = Flask(__name__)
@app.route('/stat', methods=['GET', 'POST'])

def index():
    form = InputForm(request.form)
    if request.method == 'POST' and form.validate():
        result = calculate(form.x1.data, form.x2.data, form.x3.data,
                           form.x4.data)
    else:
        result = None
    return render_template('viewstat.html', form=form, result=result)

if __name__ == '__main__':
    app.run(debug=True)
```

The view that the user gets is shaped by HTML templates in a subdirectory templates and save the template files as viewstat.html. Once we are navigating <http://127.0.0.1:5000/stat> there will appear a webpage look like Figure 10. The coding similar to hello.py arrangement, with addition on index function that

```
<form method=post action="">
<table>
{% for field in form %}
<tr>
<td>{{ field.name }}</td><td>{{ field }}</td>
<td>{{ field.label }}</td>
</tr>
{% endfor %}
</table>
<p><input type=submit value=compute</form></p>
<p>
{% if result != None %}
Mean and Variance:
{{result}}
{% endif %}
</p>
```

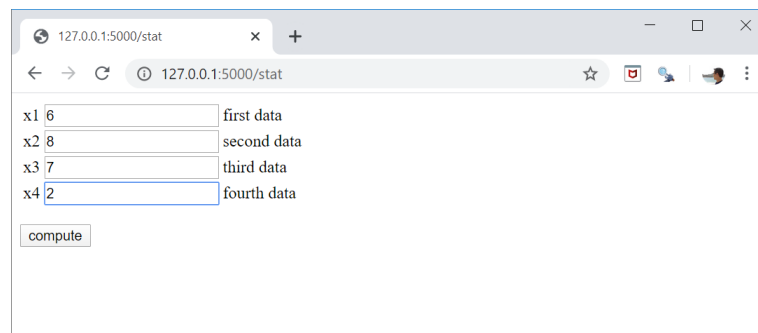
The new thing which is different from hello world application we have used before is view.html which the input data is displayed on the forms. The template language view.html supports programming Python objects inside the tag of {% and %}. Take note of contrary to plain Python, the template language needs an explicit end statement {% endif %} to finish the if-test and does not depend on indentation of blocks. Since all files are saved properly by make a new folder said tryflask2nd which inputdata.py, calculate.py, trycalculatestat.py and folder templates with viewstat.html included in templates folder then we can execute (Figure 9) by using python trycalculatestat.py

```
Command Prompt - python trycalculatestat.py
C:\tryflask2nd>venv\Scripts\activate

(venv) C:\tryflask2nd>python trycalculatestat.py
* Serving Flask app "trycalculatestat" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 613-524-673
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Figure 9. Running statistical calculation in local server.

Since the whole data already entered (Figure 10), by clicking on the compute button, the corresponding data is calculated and written out the result page seen in Figure 11 (visual bar graph) and Figure 12.



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/stat". The page contains four input fields labeled "x1", "x2", "x3", and "x4" with corresponding labels "first data", "second data", "third data", and "fourth data". The values entered in the fields are 6, 8, 7, and 2 respectively. Below the input fields is a "compute" button.

Figure 10. Web page for input forms.

The result will be shown in Figure 11.

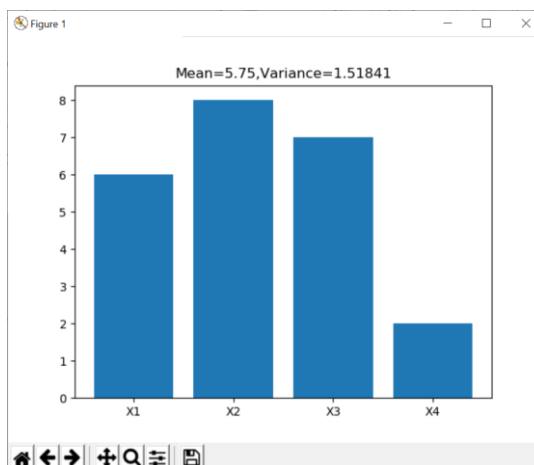
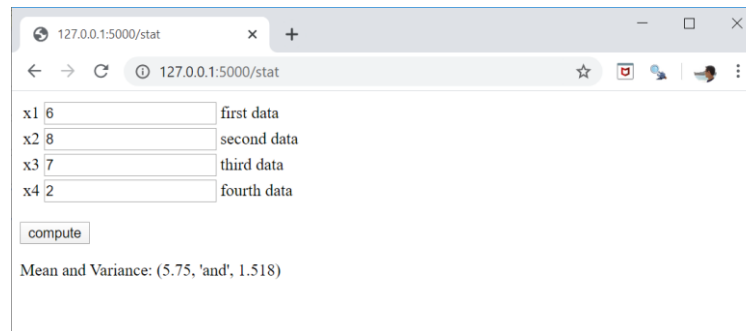


Figure 11. Data visualization using histogram from user inputting data.



**Figure 12.** The result page of computation.

This application that has been built consists of a simple web application with a form field that let user enter a data. After submitting the data to the web application, then proceed it by statistical mean and its variation and then render it back on a page with the result and visualizing data. Even though this is a very simple example of a web application, it helps encourage intuition of statisticians about how to store and pass variables and values among the different parts of code within the Flask framework.

## Conclusion

Interactivity on websites or apps between users and company using web application would increase user measurement to make right decision for advantage company business. Most web applications are built with the utility of web frameworks, which are software packages that shorten and without an overwhelming amount of programming tasks of offering services through the Internet. Users could input the data to the application on a web page, then click on some compute button and those data will be proceed by performing statistical description and analysis of data to get the best result and back comes to web page with the results of the computation are shown. The web page can be designed by data scientist to attract users to performing some activity through interface of web page and users would click any link or input data on any forms and this experiment tested as a GUI locally on the data scientist's computer before the interface would be deployed to a server and made available to the entire users on the Internet. This paper with the simple example of a web application has already shown the important to statistician to understand how web framework working as a utility for web application in how to store and pass variables and values among the different parts of code within the framework of Flask and show the result of data analysis on a web page.

In one approach we use this material in paper to anyone that just completed and step through to show how it could be carried web application out easily. This paper demonstrates how to design computing environment to do web application with statistical calculation deployed in it. This material is very new to most statisticians but it is an important aspect of modern scientific computing for many web applications. In conclusion, though this study is a simple example, it feels that the concepts will be considered with other statisticians elsewhere. We must be gaining a chance to push statistician profession, and insist that computing is an important element of their statistical skill, and it must be understood with an intellectual foundation that provides statistician with skills to reason about important computational tasks and continue to learn about new computational topics. Through these means, statistics can be seen as the wonderful and interesting career for today and the future.

## References

- Ali, Z., & Bhaskar, S. B. (2016). Basic statistical tools in research and data analysis. *Indian Journal of Anaesthesia*, 60(9), 662–669. <https://doi.org/10.4103/0019-5049.190623>
- Almeida, F., & Monteiro, J. (2017). The role of responsive design in web development. *Webology*, 14(2), 48–65.
- Cameron, C., Iosua, E., Parry, M., Richards, R., & Jaye, C. (2017). More than just numbers: Challenges for professional statisticians1. *Statistics Education Research Journal*, 16(2), 362–375.
- Choudhury, N. (2014). World Wide Web and Its Journey from Web 1.0 to Web 4.0. *IJCSIT – International Journal of Computer Science and Information Technologies*, 5(6).
- Di Ciaccio, A., & Giorgi, G. M. (2016). Statistics in the big data era. *Rivista Italiana Di Economia Demografia e Statistica Volume LXX n.4 Ottobre-Dicembre 2016*.
- Doroodchi, M., & Dastgheib, S. (2008). A framework for web application development. *Conference*.
- Gandhi, Jitendra, & Gandhi, P. (n.d.). An Importance of Web Presence for Small Business Owners. *Journal of Information, Knowledge and Research in Business Management and Administration*, 4(2).
- Garett, R., Chiu, J., Zhang, L., & Young, S. D. (2019). A Literature Review: Website Design and User Engagement. *Online Journal of Communication and Media Technologies*, 6(3), 1–10. <https://doi.org/10.29333/ojcm/2556>
- Grinberg, M. (2014). *Flask Web Development*. O'Reilly, Inc.,.
- Hamed Hijazi, R. (2014). Meeting the demand for statisticians in UAE: challenges and experiences. *Journal of Economic and Administrative Sciences*, 30(1), 45–52. <https://doi.org/10.1108/jeas-09-2013-0033>
- Harris, L., & Rae, A. (2009). Social networks: The future of marketing for small business. *Journal of Business Strategy*, 30(5), 24–31. <https://doi.org/10.1108/02756660910987581>
- Kavya, J S.L., & Sarathambekai, S. (2019). Python libraries and packages for web Development – A Survey. *International Journal of Innovative Research In Technology (IJIRT)*, 5(12).
- Krishen, A. S. (2013). First impressions count: Exploring the importance of website categorisation. *International Journal of Computer Applications in Technology*, 47(1), 32–43. <https://doi.org/10.1504/IJCAT.2013.054300>
- Manyika, J., & Roxburgh, C. (2011). The great transformer: The impact of the Internet on economic growth and prosperity. *McKinsey Global Institute*, 1–10. [http://www.iei.liu.se/facksprak/engelska/civilingenjorsutbildning/then18/kursmaterialar/kiv/lesson-twelve/1.333650/The\\_great\\_transformer\\_Impact\\_of\\_Internet\\_on\\_economic\\_growth.pdf](http://www.iei.liu.se/facksprak/engelska/civilingenjorsutbildning/then18/kursmaterialar/kiv/lesson-twelve/1.333650/The_great_transformer_Impact_of_Internet_on_economic_growth.pdf)
- Mauro, A. De, Greco, M., & Grimaldi, M. (2016). A formal definition of Big Data based on its essential features,- *Library Review*. 65(3), 122–135. <https://doi.org/10.1108/LR-06-2015-0061>
- Nair, V. (2013). What is the big deal about Big Data? *Paper Presented at the 59th ISI World Statistics Congress of the International Statistical Institute. Hong Kong, China: ISI. Abstract Retrieved from Http://2013.Isiproceedings.Org/Files/IPS106-P1-A.Pdf*, 2013.
- Nikolaos, A., Nikolaos, T., Christos, F., & Eleftherios, P. (2003). *Advances in Informatics*. Springer.
- West, J. D. (2016). , The Science of Data Science. *Journal of Integrated Creative Studies*, 2016-010e, 1689–1699.
- Yan, D., & Davis, G. E. (2019). A First Course in Data Science. *Journal of Statistics Education*, 27(2), 99–109. <https://doi.org/10.1080/10691898.2019.1623136>