Instructions after running the installation script -

For first time users, after running the installation script -

```
cd ArduCopter
. ~/.profile
export PATH=$PATH:$HOME/ardupilot/Tools/autotest
export PATH=/usr/lib/ccache:$PATH
. ~/.bashrc
sim_vehicle.py -w
```

copy paste this in the terminal to compile arducopter, you will see about 950-ish components to be compiled, wait for 5-10 minutes.

```
sudo apt-get install python3-dev python3-opencv python3-wxgtk4.0 python3-pip python3-matplotlib python3-lxml python3-pygame
pip3 install PyYAML mavproxy --user
echo 'export PATH="$PATH:$HOME/.local/bin"' >> ~/.bashrc
. ~/.bashrc
```

if you get a mavproxy error, then copy paste the above chunk of code.

Now, follow the below to set up the environment, make sure to note the directories and move into them before running the stuff after the $ symbol.

You have to get 4 terminal windows and run one command on each, sometimes when a terminal is started, you will need to source the bash scripts required for the ardupilot tools aka autotest to be in the environment, if you get an error like sim_vehicle.py not found etc. then, rerun . ~/.bashrc and . ~/.profile and try executing sim_vehicle.py again.

abdul@ubuntu:~/Downloads/MissionPlanner-latest$ mono MissionPlanner.exe

runs mission planner

abdul@ubuntu:~/Documents/Programming/ardupilot$ ./Tools/autotest/fg_quad_view.sh

runs flightgear

abdul@ubuntu:~/Documents/Programming/ardupilot/ArduCopter$ sim_vehicle.py --console --map

runs mavlink in 2D with console

abdul@ubuntu:~/Documents/Programming/ardupilot/ArduCopter$ sim_vehicle.py -L KSFO

runs mavlink in 3D without console

Once all of this is done, start executing the programs. Even if the program shows some minor errors like datatype conversion errors, it doesn't really matter cause the instructions will be sent to the drone already and the dude will take care of the flight paths on its own.