Overview

The project proposal states it aims to use LIDAR and other remote sensing to produce the following key deliverables:

> maps of the following metrics for all formations with tree or shrub cover:
>
> - Near surface fuel height and cover (%)
> - Elevated fuel height and cover (%)
> - Canopy height and cover (%)

The pilot is being run over three areas: Pilliga, Central Coast and Snowy-Monaro

datasets:

For each area the following raster layers hve been produced:

- `X-fuels-classes-2021-density.tif` : fuel loads (t/ha) for litter, surface, elevated and bark.
- `X-vegetation-canopy_base_height-2021.tif` : base of the canopy (m)
- `X-vegetation-canopy_cover-2021.tif` : canopy cover (%)
- `X-vegetation-canopy_height-2021.tif` : canopy height (m)
- `X-vegetation-ladder_fuel_density-2021.tif` : ladder fuel load, assumed to be (t/ha)

note:

it is not clear how these layers can be used to produce near surface or elevated fuel height or cover.

In [2]:
```
import rioxarray as rxr
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import rasterio
from rasterio.plot import show, show_hist
```

Fuel Loads

In [32]:
```
## Fuel Loads
fuel_load_paths = {
    'pilliga': 'analyses/Spatial_data/Pilliga-fuels-classes-2021-density.tif',
    'central': 'analyses/Spatial_data/Centralcoast-fuels-classes-2021-density.tif',
    'snowy': 'analyses/Spatial_data/Southmnts-fuels-classes-2021-density.tif',
}

fuel_loads = {}

strata = ['litter', 'surface', 'elevated', 'bark']

for loc, path in fuel_load_paths.items():
```

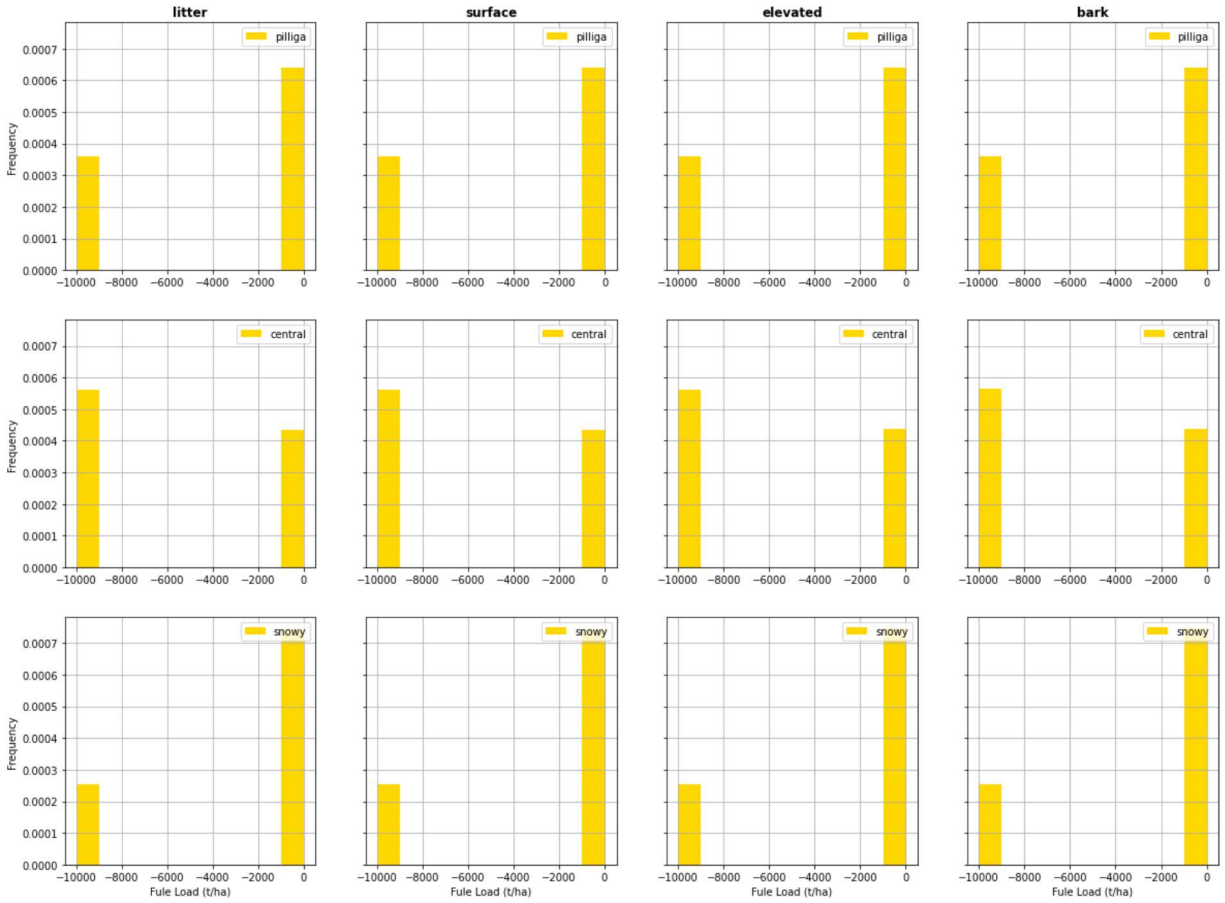
```
    with rasterio.open(path, 'r') as src:
        load_dic = {}
        for i, stratum in enumerate(strata):
            load_dic[stratum] = src.read(i+1) #GDAL indexed from 1
    fuel_loads[loc] = load_dic
```

Basic description statistics

Best overall appreciation from histograms

In [53]:

```
fig, axes = plt.subplots(len(fuel_loads),len(strata), sharey=True, figsize=(20,15))
for loc_id, (loc, strata) in enumerate(fuel_loads.items()):
    for stratum_id, (stratum, data) in enumerate(strata.items()):
        ax = axes[loc_id,stratum_id]
        show_hist(
            data,
            label= loc,
            title= stratum,
            ax = ax,
            density=True,
        )
        ax.set_xlabel('Fule Load (t/ha)')
        if loc_id: ax.title.set_visible(False)
        if loc_id < (len(fuel_loads)-1): ax.xaxis.label.set_visible(False)
        if stratum_id: ax.yaxis.label.set_visible(False)


# for ax in axes: ax.set_xlabel('Fuel load (t/ha)')
plt.show()
```



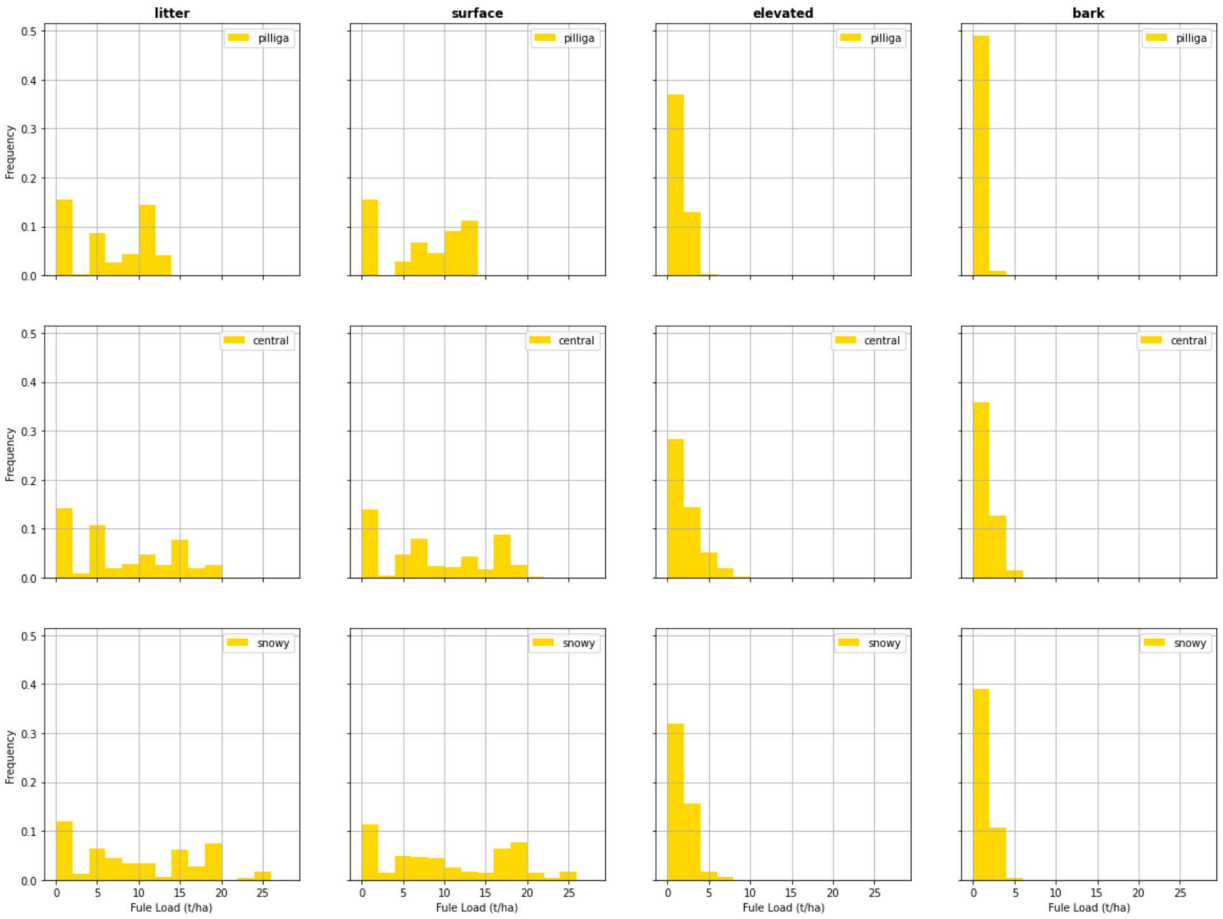
It is obvious from the binomial distribution in the plots that the data need to be cleaned. It

seems that a value of -9999 was used ti indicate no data. All values less than 0 were removed.

In [56]:
```
for loc, strata in fuel_loads.items():
    for stratum, data in strata.items():
        data = np.where(data < 0, np.nan, data)
        fuel_loads[loc][stratum]= data
```

histograms cleaned

In [64]:
```
fig, axes = plt.subplots(len(fuel_loads),len(strata), sharey=True, sharex=True, figs
bins=[x for x in range(0,30,2)]
for loc_id, (loc, strata) in enumerate(fuel_loads.items()):
    for stratum_id, (stratum, data) in enumerate(strata.items()):
        ax = axes[loc_id,stratum_id]
        show_hist(
            data,
            label= loc,
            title= stratum,
            ax = ax,
            density=True,
            bins=bins,
        )
        ax.set_xlabel('Fuel Load (t/ha)')
        if loc_id: ax.title.set_visible(False)
        if loc_id < (len(fuel_loads)-1): ax.xaxis.label.set_visible(False)
        if stratum_id: ax.yaxis.label.set_visible(False)


# for ax in axes: ax.set_xlabel('Fuel Load (t/ha)')
plt.show()
```

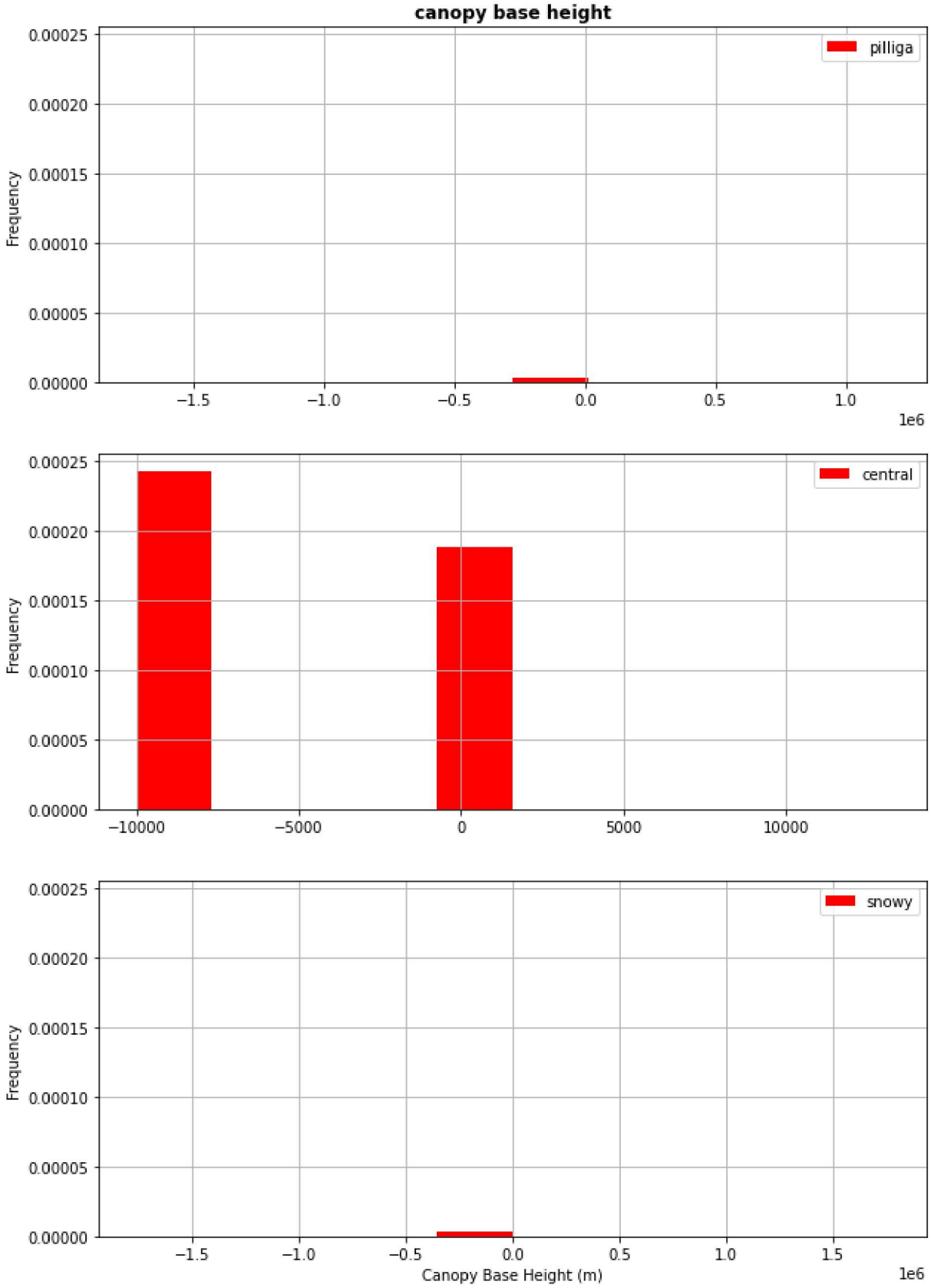


Canopy Base Height (m)

In [66]:
```
## Canopy Base Height
cbh_paths = {
    'pilliga': 'analyses/Spatial_data/Pilliga-vegetation-canopy_base_height-2021.tif
    'central': 'analyses/Spatial_data/Centralcoast-vegetation-canopy_base_height-202
    'snowy': 'analyses/Spatial_data/Southmnts-vegetation-canopy_base_height-2021.tif
}

canopy_bh = {}

for loc, path in cbh_paths.items():
    with rasterio.open(path, 'r') as src:
        cbh = src.read() #GDAL indexed from 1
    canopy_bh[loc] = cbh
```

Histograms - raw data

In [68]:
```
fig, axes = plt.subplots(len(canopy_bh),1, sharey=True, figsize=(10,15))
for loc_id, (loc, cbh) in enumerate(canopy_bh.items()):

    ax = axes[loc_id]
    show_hist(
        cbh,
        label= loc,
        title= 'canopy base height',
        ax = ax,
        density=True,
    )
    ax.set_xlabel('Canopy Base Height (m)')
    if loc_id: ax.title.set_visible(False)
    if loc_id < (len(canopy_bh)-1): ax.xaxis.label.set_visible(False)
    # if stratum_id: ax.yaxis.label.set_visible(False)

plt.show()
```



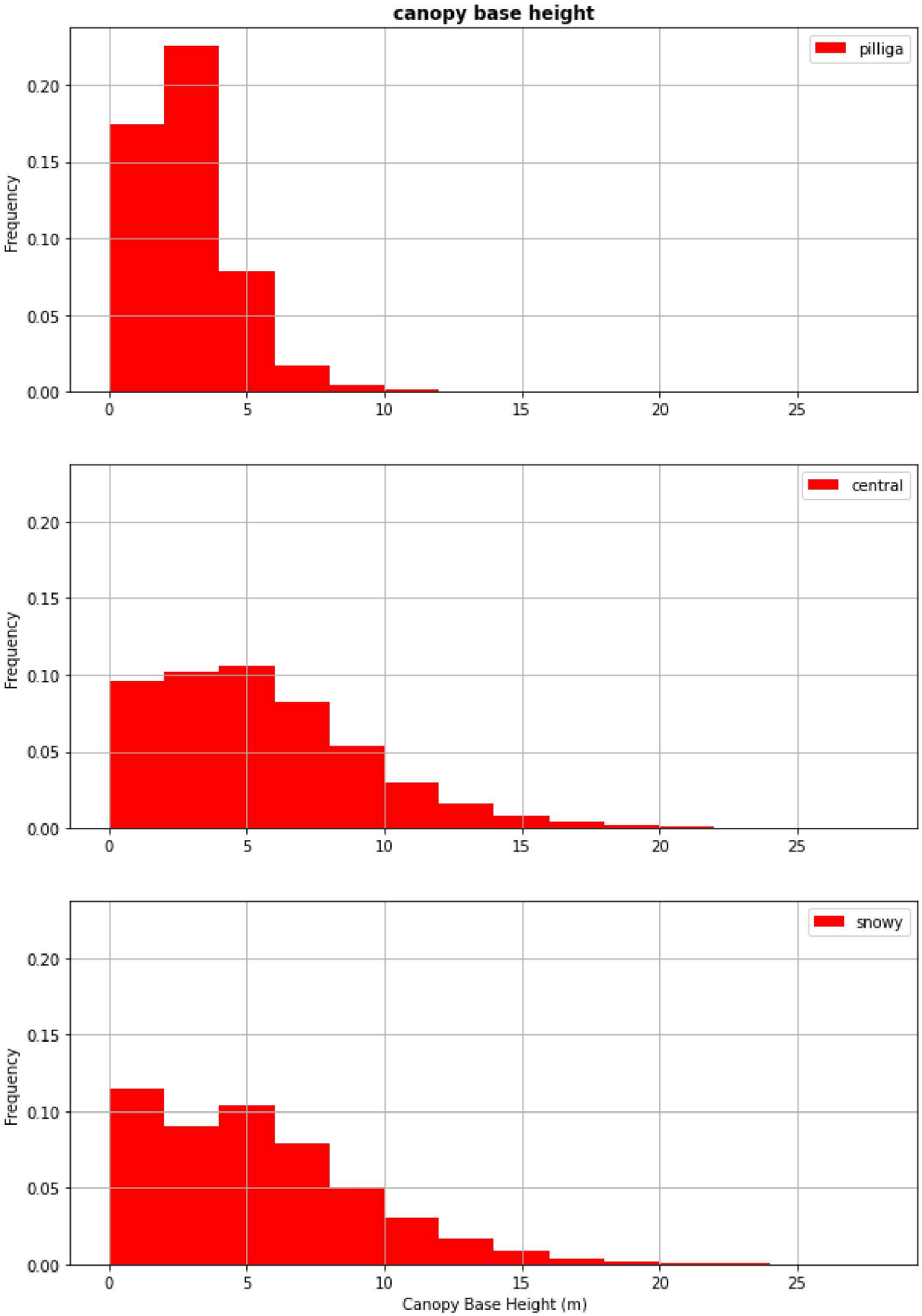
Shows -9999 used as flag but also indicates some large values wich need to be cleaned. Restric data to values between 0 and 30 m

In [71]:
```
for loc, cbh in canopy_bh.items():
    cbh = np.where(cbh < 0, np.nan, cbh)
    cbh = np.where(cbh > 30, np.nan, cbh)
    canopy_bh[loc]= cbh
```

Histograms cleaned

In [73]:

```
fig, axes = plt.subplots(len(canopy_bh),1, sharey=True, figsize=(10,15))
bins=[x for x in range(0,30,2)]
for loc_id, (loc, cbh) in enumerate(canopy_bh.items()):

    ax = axes[loc_id]
    show_hist(
        cbh,
        label= loc,
        title= 'canopy base height (m)',
        ax = ax,
        density=True,
        bins=bins
    )
    ax.set_xlabel('Canopy Base Height (m)')
    if loc_id: ax.title.set_visible(False)
    if loc_id < (len(canopy_bh)-1): ax.xaxis.label.set_visible(False)
    # if stratum_id: ax.yaxis.label.set_visible(False)

plt.show()
```



Canopy Cover (%)

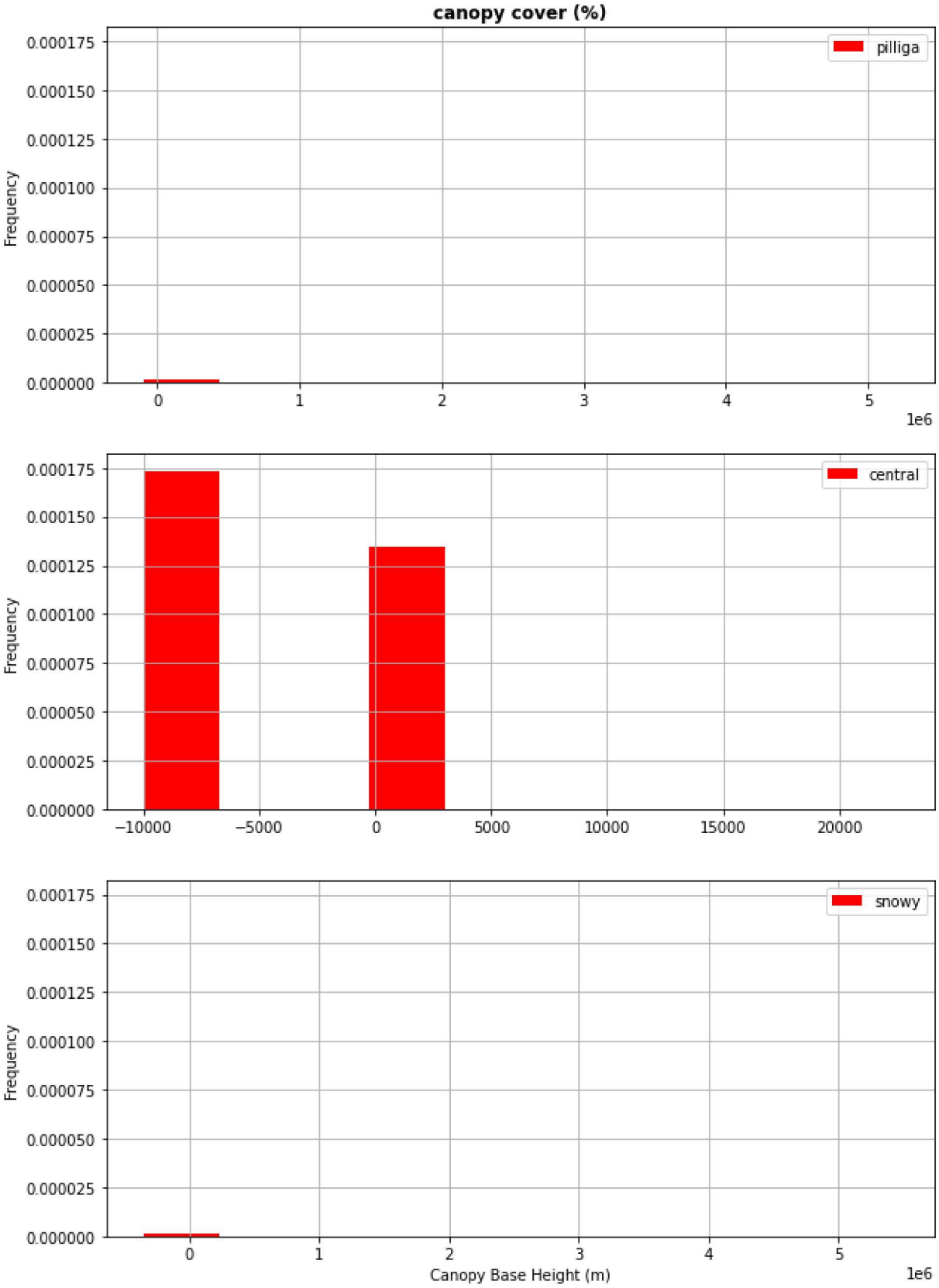
In [3]:
```
cc_paths = {
    'pilliga': 'analyses/Spatial_data/Pilliga-vegetation-canopy_cover-2021.tif',
    'central': 'analyses/Spatial_data/Centralcoast-vegetation-canopy_cover-2021.tif'
    'snowy': 'analyses/Spatial_data/Southmnts-vegetation-canopy_cover-2021.tif',
}
```

```
canopy_cov = {}

for loc, path in cc_paths.items():
    with rasterio.open(path, 'r') as src:
        cc = src.read() #GDAL indexed from 1
    canopy_cov[loc] = cc
```

Histograms raw

In [5]:

```
fig, axes = plt.subplots(len(canopy_cov),1, sharey=True, figsize=(10,15))
for loc_id, (loc, cc) in enumerate(canopy_cov.items()):

    ax = axes[loc_id]
    show_hist(
        cc,
        label= loc,
        title= 'canopy cover (%)',
        ax = ax,
        density=True,
    )
    ax.set_xlabel('Canopy Cover (%)')
    if loc_id: ax.title.set_visible(False)
    if loc_id < (len(canopy_cov)-1): ax.xaxis.label.set_visible(False)
    # if stratum_id: ax.yaxis.label.set_visible(False)

plt.show()
```



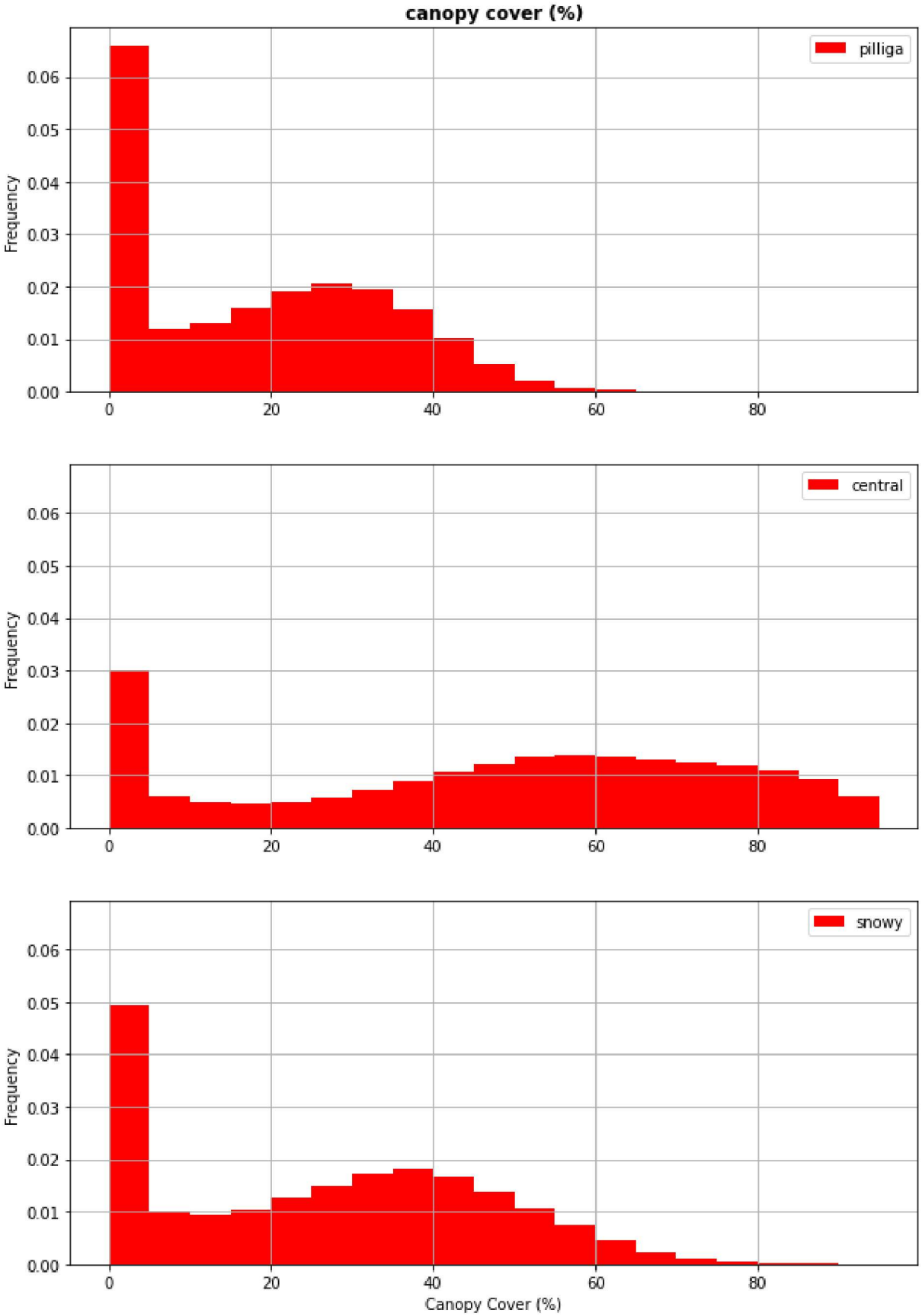
clean to keep values between 0 and 100

In [6]:
```
for loc, cc in canopy_cov.items():
    cc = np.where(cc < 0, np.nan, cc)
    cc = np.where(cc > 100, np.nan, cc)
    canopy_cov[loc]= cc
```

Histogram cleaned

In [7]:

```
fig, axes = plt.subplots(len(canopy_cov),1, sharey=True, figsize=(10,15))
bins=[x for x in range(0,100,5)]
for loc_id, (loc, cc) in enumerate(canopy_cov.items()):

    ax = axes[loc_id]
    show_hist(
        cc,
        label= loc,
        title= 'canopy cover (%)',
        ax = ax,
        density=True,
        bins=bins
    )
    ax.set_xlabel('Canopy Cover (%)')
    if loc_id: ax.title.set_visible(False)
    if loc_id < (len(canopy_cov)-1): ax.xaxis.label.set_visible(False)

plt.show()
```



Canopy Height (m)

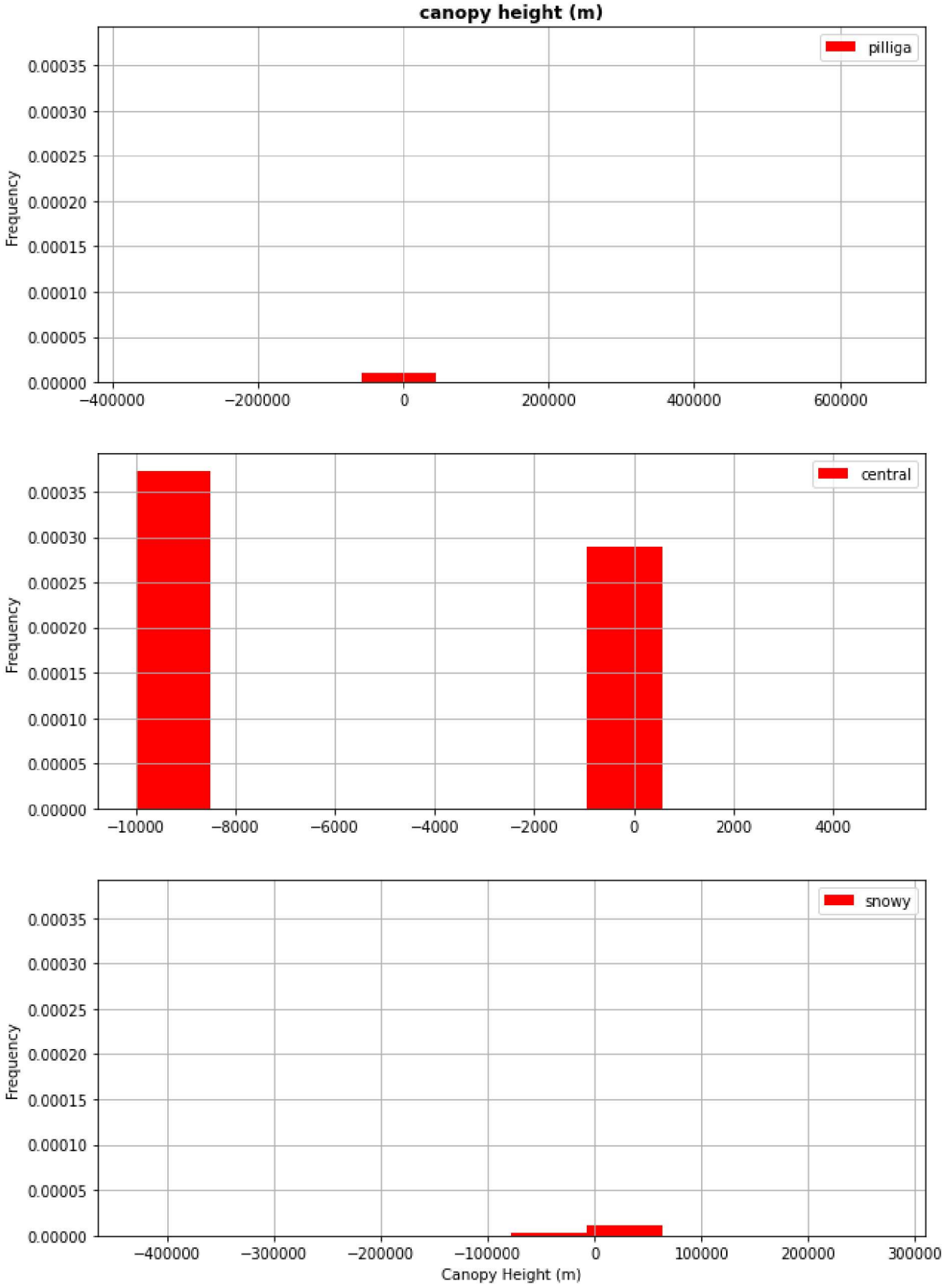
In [9]:
```
ch_paths = {
    'pilliga': 'analyses/Spatial_data/Pilliga-vegetation-canopy_height-2021.tif',
    'central': 'analyses/Spatial_data/Centralcoast-vegetation-canopy_height-2021.tif
    'snowy': 'analyses/Spatial_data/Southmnts-vegetation-canopy_height-2021.tif',
}
```

```
canopy_h = {}

for loc, path in ch_paths.items():
    with rasterio.open(path, 'r') as src:
        ch = src.read() #GDAL indexed from 1
    canopy_h[loc] = ch
```

Histograms Raw - Canopy Height

In [10]:

```
fig, axes = plt.subplots(len(canopy_h),1, sharey=True, figsize=(10,15))
for loc_id, (loc, ch) in enumerate(canopy_h.items()):

    ax = axes[loc_id]
    show_hist(
        ch,
        label= loc,
        title= 'canopy height (m)',
        ax = ax,
        density=True,
    )
    ax.set_xlabel('Canopy Height (m)')
    if loc_id: ax.title.set_visible(False)
    if loc_id < (len(canopy_h)-1): ax.xaxis.label.set_visible(False)

plt.show()
```



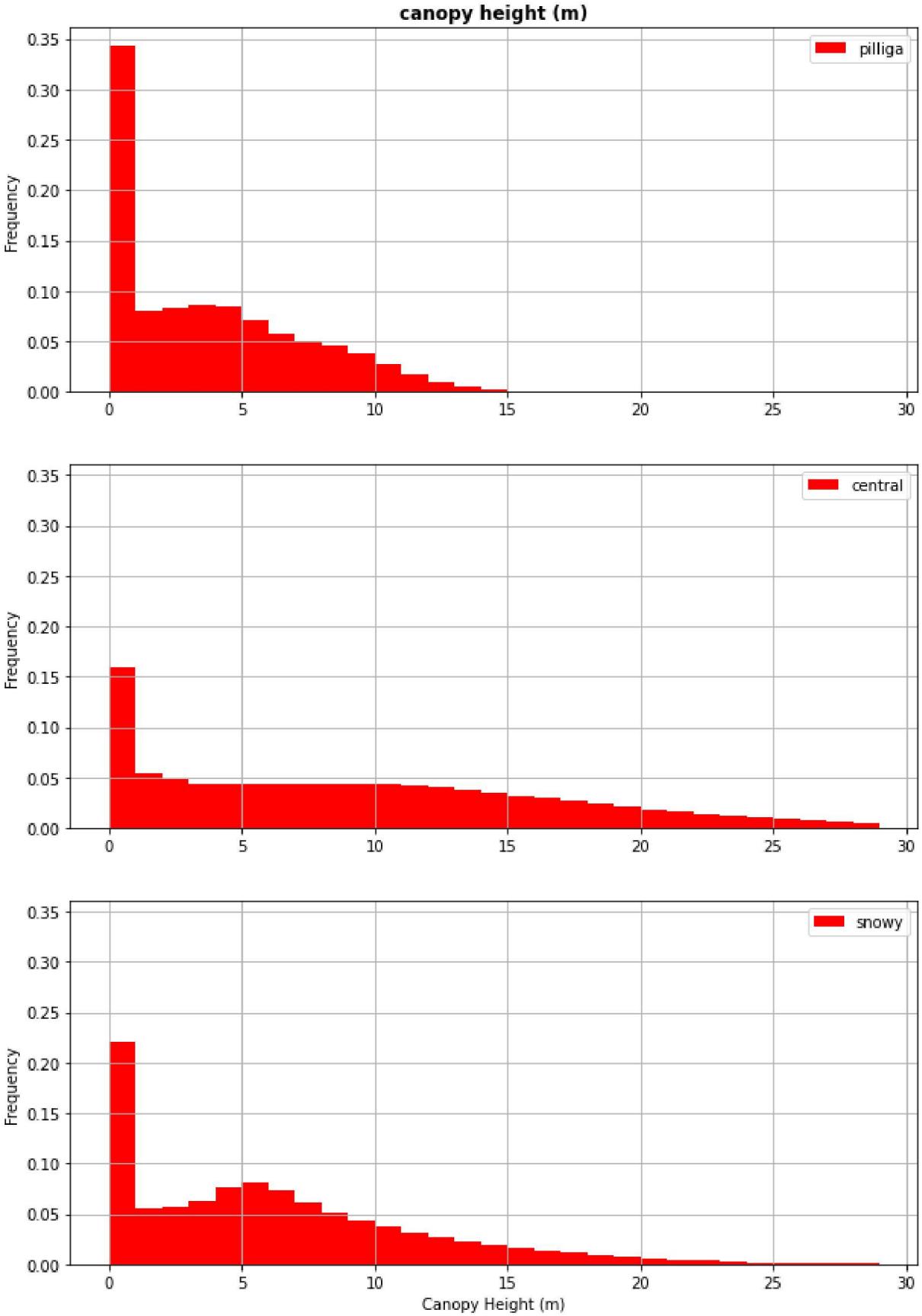
clean to keep values between 0 and 30 m

In [11]:
```
for loc, ch in canopy_h.items():
    ch = np.where(ch < 0, np.nan, ch)
    ch = np.where(ch > 30, np.nan, ch)
    canopy_h[loc]= ch
```

Histogram cleaned

In [13]:
```
fig, axes = plt.subplots(len(canopy_h),1, sharey=True, figsize=(10,15))
```

```
bins=[x for x in range(0,30,1)]
for loc_id, (loc, ch) in enumerate(canopy_h.items()):

    ax = axes[loc_id]
    show_hist(
        ch,
        label= loc,
        title= 'canopy height (m)',
        ax = ax,
        density=True,
        bins=bins
    )
    ax.set_xlabel('Canopy Height (m)')
    if loc_id: ax.title.set_visible(False)
    if loc_id < (len(canopy_h)-1): ax.xaxis.label.set_visible(False)

plt.show()
```



Ladder Fuel Density (t/ha)

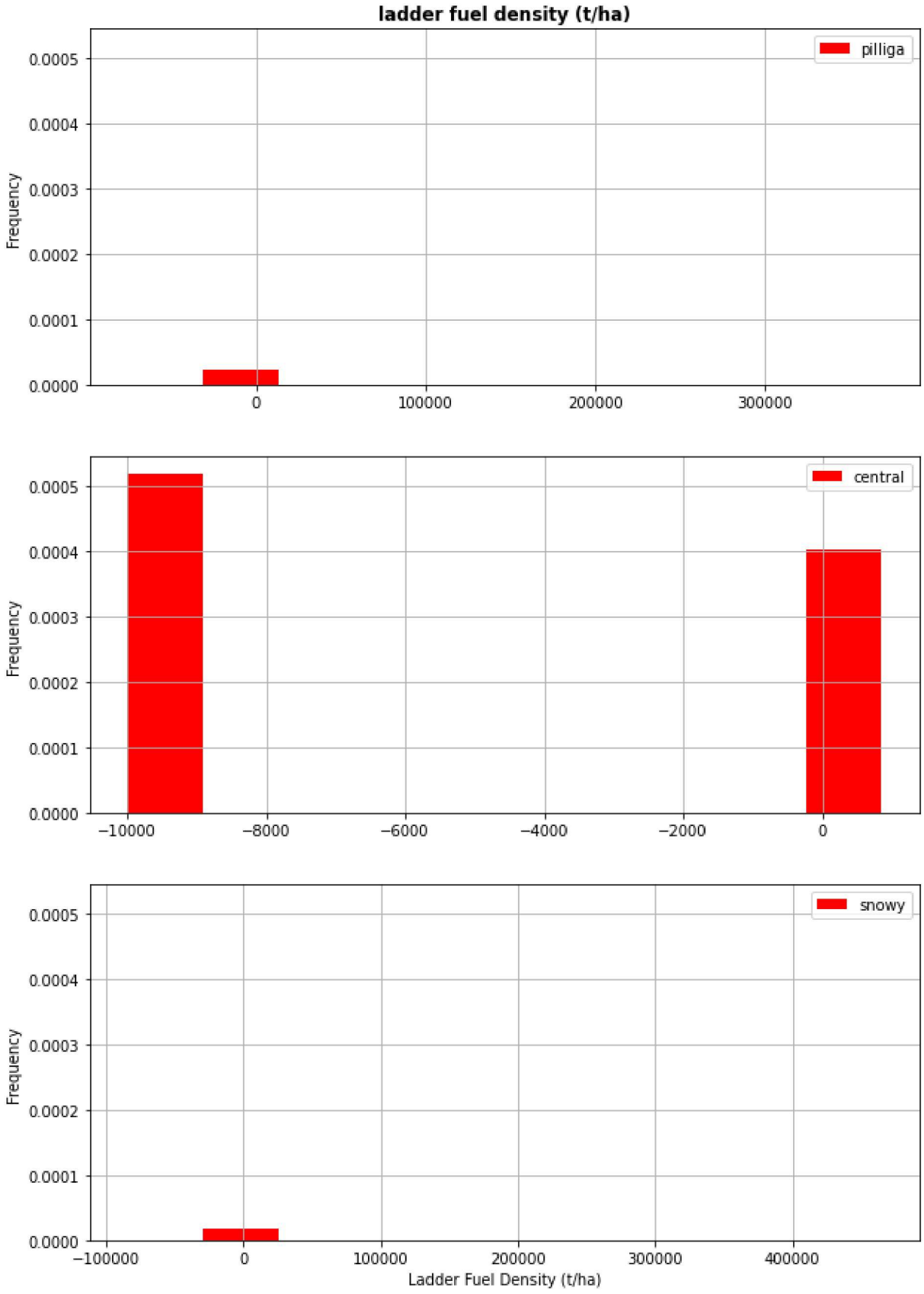
In [18]:
```
lfd_paths = {
    'pilliga': 'analyses/Spatial_data/Pilliga-vegetation-ladder_fuel_density-2021.ti
    'central': 'analyses/Spatial_data/Centralcoast-vegetation-ladder_fuel_density-20
    'snowy': 'analyses/Spatial_data/Southmnts-vegetation-ladder_fuel_density-2021.ti
}
```

```
lfd = {}

for loc, path in lfd_paths.items():
    with rasterio.open(path, 'r') as src:
        lf = src.read()
    lfd[loc] = lf
```

Histograms Raw - Ladder Fuel Density

In [15]:

```
fig, axes = plt.subplots(len(lfd),1, sharey=True, figsize=(10,15))
for loc_id, (loc, lf) in enumerate(lfd.items()):

    ax = axes[loc_id]
    show_hist(
        lf,
        label= loc,
        title= 'ladder fuel density (t/ha)',
        ax = ax,
        density=True,
    )
    ax.set_xlabel('Ladder Fuel Density (t/ha)')
    if loc_id: ax.title.set_visible(False)
    if loc_id < (len(lfd)-1): ax.xaxis.label.set_visible(False)

plt.show()
```

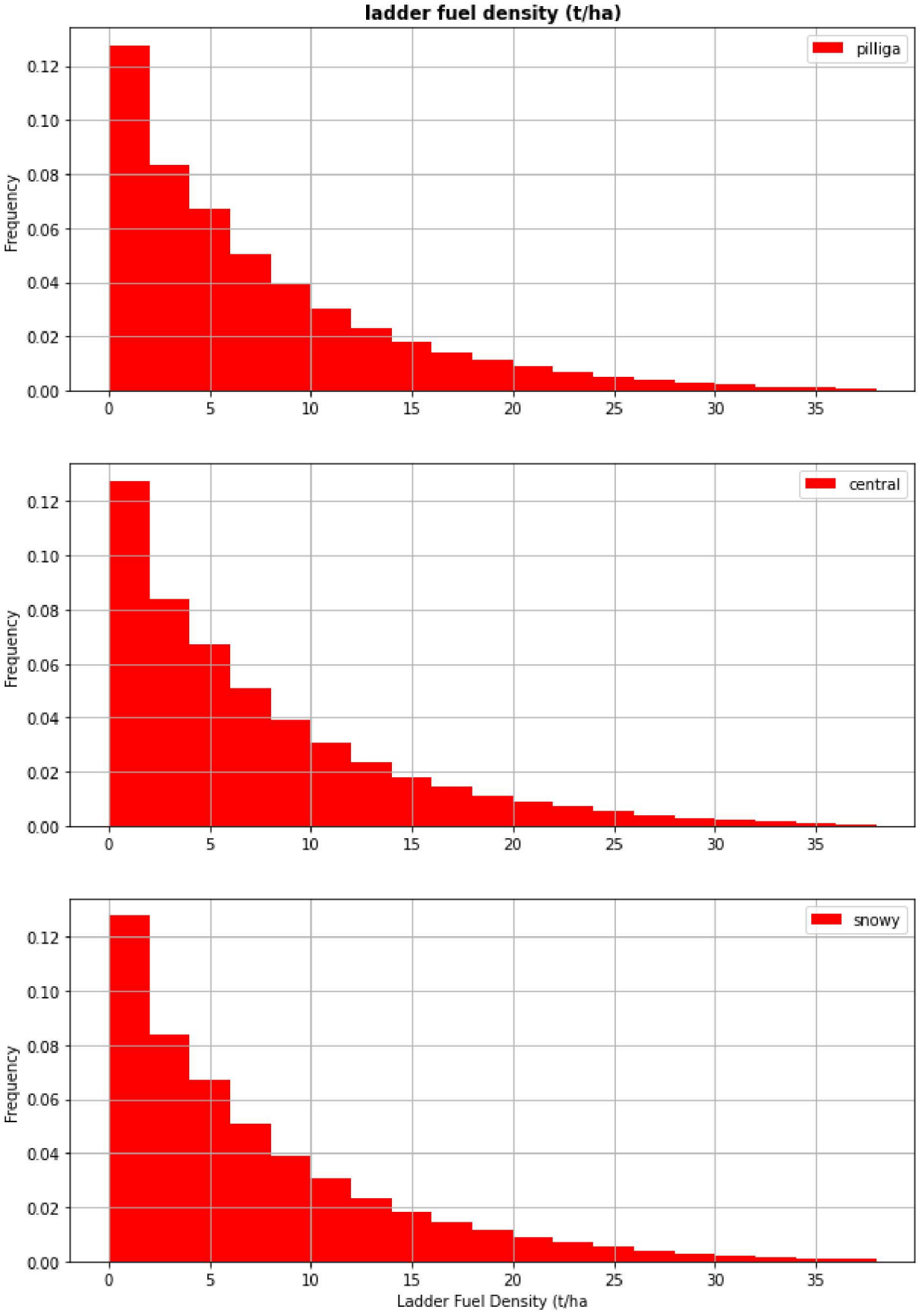


clean to keep values between 0 and 40 t/ha

In [19]:
```
for loc, lf in lfd.items():
    lf = np.where(lf < 0, np.nan, lf)
    lf = np.where(lf > 40, np.nan, lf)
    lfd[loc]= lf
```

Histogram cleaned

In [20]:
```
fig, axes = plt.subplots(len(lfd),1, sharey=True, figsize=(10,15))
bins=[x for x in range(0,40,2)]
for loc_id, (loc, lfch) in enumerate(lfd.items()):

    ax = axes[loc_id]
    show_hist(
        lf,
        label= loc,
        title= 'ladder fuel density (t/ha)',
        ax = ax,
        density=True,
        bins=bins
    )
    ax.set_xlabel('Ladder Fuel Density (t/ha)')
    if loc_id: ax.title.set_visible(False)
    if loc_id < (len(canopy_h)-1): ax.xaxis.label.set_visible(False)

plt.show()
```


ladder fuel density (t/ha)
pilliga
central
snowy
Frequency
Ladder Fuel Density (t/ha
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0
5
10
15
20
25
30
35