PROJECT 1, GALAXY-QUASAR CLASSIFICATION

Farah Najla - IYC18 IVC on Astrostatistics and Machine Learning Day 9 Hands-on

Import data

In [16]:
```
# Import extensions
import numpy as np
import matplotlib.pyplot as plt
import astropy.io.fits as fits

# Import data from website
data_url = 'https://anirut.space/data/sdss_galaxy_qso.fits'
hdul = fits.open(data_url)
data = hdul[1].data
```

Create parameters

In [31]:
```
ra = data["ra"]
dec = data["dec"]
u = data["u"]
g = data["g"]
r = data["r"]
M_i = data["i"]
z = data["z"]
obj_class = data["class"]
n = len(u)
print("Number of objects = ", n)
```

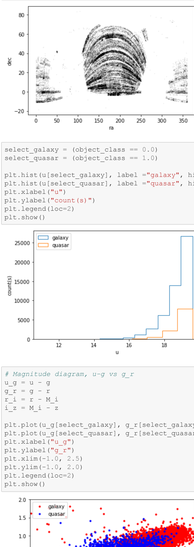
```
Number of objects =  61843
```

Classification

In [23]:
```
object_class = np.empty(n)
for i in range(0, n):
    if obj_class[i] == "GALAXY":
        object_class[i] = 0.0
    elif obj_class[i] == "QSO":
        object_class[i] = 1.0
    else:
        print("error", j)
```

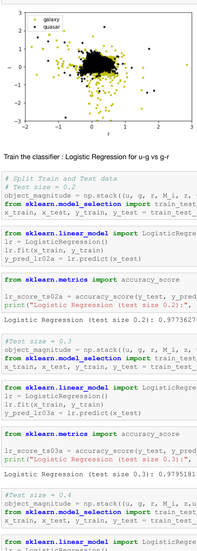
Plotting

In [24]:
```
# RA vs DEC
plt.plot(ra, dec, 'k.', markersize = 0.1)
plt.xlabel("ra")
plt.ylabel("dec")
plt.savefig("figure1.png", dpi=300)
plt.show()
```



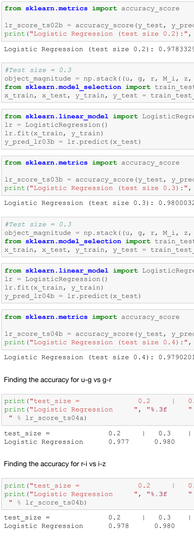
In [25]:
```
select_galaxy = (object_class == 0.0)
select_quasar = (object_class == 1.0)

plt.hist(u[select_galaxy], label ="galaxy", histtype='step')
plt.hist(u[select_quasar], label ="quasar", histtype='step')
plt.xlabel("u")
plt.ylabel("count(s)")
plt.legend(loc=2)
plt.show()
```



In [49]:
```
# Magnitude diagram, u-g vs g_r
u_g = u - g
g_r = g - r
r_i = r - M_i
i_z = M_i - z

plt.plot(u_g[select_galaxy], g_r[select_galaxy], 'r.', label="galaxy")
plt.plot(u_g[select_quasar], g_r[select_quasar], 'b.', label="quasar")
plt.xlabel("u_g")
plt.ylabel("g_r")
plt.xlim(-1.0, 2.5)
plt.ylim(-1.0, 2.0)
plt.legend(loc=2)
plt.show()
```



In [84]:
```
# Magnitude diagram, r-i
plt.plot(r_i[select_galaxy], i_z[select_galaxy], 'y.', label="galaxy")
plt.plot(r_i[select_quasar], i_z[select_quasar], 'k.', label="quasar")
plt.xlabel("r-i")
plt.ylabel("i-z")
plt.xlim(-2.0, 3.0)
plt.ylim(-3.0, 3.0)
plt.legend(loc=2)
plt.show()
```


Train the classifier : Logistic Regression for u-g vs g-r

In [65]:
```
# Split Train and Test data
# Test size = 0.2
object_magnitude = np.stack((u, g, r, M_i, z, u_g, g_r), axis=-1)
from sklearn.model_selection import train_test_split
X_train, x_test, y_train, y_test = train_test_split(object_magnitude, object_class, test_size=0.2)
```

In [66]:
```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(x_train, y_train)
y_pred_lr02a = lr.predict(x_test)
```

In [67]:
```
from sklearn.metrics import accuracy_score

lr_score_ts02a = accuracy_score(y_test, y_pred_lr02a)
print("Logistic Regression (test size 0.2):", lr_score_ts02a)
```

```
Logistic Regression (test size 0.2): 0.9773627617430674
```

In [68]:
```
#Test size = 0.3
object_magnitude = np.stack((u, g, r, M_i, z, u_g, g_r), axis=-1)
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(object_magnitude, object_class, test_size=0.3)
```

In [69]:
```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(x_train, y_train)
y_pred_lr03a = lr.predict(x_test)
```

In [71]:
```
from sklearn.metrics import accuracy_score

lr_score_ts03a = accuracy_score(y_test, y_pred_lr03a)
print("Logistic Regression (test size 0.3):", lr_score_ts03a)
```

```
Logistic Regression (test size 0.3): 0.9795181372284806
```

In [72]:
```
#Test size = 0.4
object_magnitude = np.stack((u, g, r, M_i, z,u_g, g_r), axis=-1)
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(object_magnitude, object_class, test_size=0.4)
```

In [73]:
```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(x_train, y_train)
y_pred_lr04a = lr.predict(x_test)
```

In [74]:
```
from sklearn.metrics import accuracy_score

lr_score_ts04a = accuracy_score(y_test, y_pred_lr04a)
print("Logistic Regression (test size 0.4):", lr_score_ts04a)
```

```
Logistic Regression (test size 0.4): 0.9791818255315708
```

Train the classifier : Logistic Regression for r-i vs i-z

In [75]:
```
# Split Train and Test data
# Test size = 0.2
object_magnitude = np.stack((u, g, r, M_i, z, r_i, i_z), axis=-1)
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(object_magnitude, object_class, test_size=0.2)
```

In [76]:
```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(x_train, y_train)
y_pred_lr02b = lr.predict(x_test)
```

In [77]:
```
from sklearn.metrics import accuracy_score

lr_score_ts02b = accuracy_score(y_test, y_pred_lr02b)
print("Logistic Regression (test size 0.2):", lr_score_ts02b)
```

```
Logistic Regression (test size 0.2): 0.9783329290969359
```

In [78]:
```
#Test size = 0.3
object_magnitude = np.stack((u, g, r, M_i, z, r_i, i_z), axis=-1)
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(object_magnitude, object_class, test_size=0.3)
```

In [79]:
```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(x_train, y_train)
y_pred_lr03b = lr.predict(x_test)
```

In [80]:
```
from sklearn.metrics import accuracy_score

lr_score_ts03b = accuracy_score(y_test, y_pred_lr03b)
print("Logistic Regression (test size 0.3):", lr_score_ts03b)
```

```
Logistic Regression (test size 0.3): 0.9800032339783323
```

In [81]:
```
#Test size = 0.3
object_magnitude = np.stack((u, g, r, M_i, z, r_i, i_z), axis=-1)
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(object_magnitude, object_class, test_size=0.4)
```

In [82]:
```
from sklearn.linear_model import LogisticRegression
lr = LogisticRegression()
lr.fit(x_train, y_train)
y_pred_lr04b = lr.predict(x_test)
```

In [83]:
```
from sklearn.metrics import accuracy_score

lr_score_ts04b = accuracy_score(y_test, y_pred_lr04b)
print("Logistic Regression (test size 0.4):", lr_score_ts04b)
```

```
Logistic Regression (test size 0.4): 0.9790201309725928
```

Finding the accuracy for u-g vs g-r

In [86]:
```
print("test_size =              0.2     |   0.3    |   0.4    |")
print("Logistic Regression     ", "%.3f" % lr_score_ts02a, "%.3f" % lr_score_ts03a,  "%.3f" % lr_score_ts04a)
```

```
test_size =              0.2     |   0.3    |   0.4    |
Logistic Regression      0.977       0.980      0.979
```

Finding the accuracy for r-i vs i-z

In [87]:
```
print("test_size =              0.2     |   0.3    |   0.4    |")
print("Logistic Regression     ", "%.3f" % lr_score_ts02b, "%.3f" % lr_score_ts03b,  "%.3f" % lr_score_ts04b)
```

```
test_size =              0.2     |   0.3    |   0.4    |
Logistic Regression      0.978       0.980      0.979
```