Project 3: Heat on Rice

SDS348 Spring 2021 ¶

Emily Qiu eq887

In [39]:
```
# Import packages
import pandas as pd
```

In [13]:
```
#import dataset
data=pd.read_csv("Raw_Data.csv")
print(data)
```

	Year	Cultivars	Tem	RH	Days	R1	R2	R3
0	2017	16343	33	75.0	8.0	20.20	20.06	20.38
1	2017	16343	35	75.0	8.0	17.32	17.98	18.40
2	2017	16343	37	75.0	8.0	17.46	17.44	17.18
3	2017	16343	33	75.0	15.0	19.40	19.30	19.24
4	2017	16343	35	75.0	15.0	17.44	17.46	17.52
5	2017	16343	37	75.0	15.0	17.01	17.22	17.00
6	2017	16343	33	85.0	8.0	20.52	20.62	20.82
7	2017	16343	35	85.0	8.0	18.94	19.04	18.96
8	2017	16343	37	85.0	8.0	18.40	18.30	18.32
9	2017	16343	33	85.0	15.0	19.88	20.26	20.00
10	2017	16343	35	85.0	15.0	18.96	19.00	18.92
11	2017	16343	37	85.0	15.0	18.18	18.04	18.42
12	2017	16343	CK	NaN	NaN	20.02	20.40	20.04
13	2017	IR64	33	75.0	8.0	22.56	22.44	22.60
14	2017	IR64	35	75.0	8.0	21.58	21.56	21.70
15	2017	IR64	37	75.0	8.0	21.66	21.64	21.68
16	2017	IR64	33	75.0	15.0	23.00	23.38	22.88
17	2017	IR64	35	75.0	15.0	22.10	22.12	21.94
18	2017	IR64	37	75.0	15.0	21.68	21.06	20.86
19	2017	IR64	33	85.0	8.0	23.70	23.54	23.54
20	2017	IR64	35	85.0	8.0	22.82	23.00	23.00
21	2017	IR64	37	85.0	8.0	22.18	22.12	22.08
22	2017	IR64	33	85.0	15.0	23.36	23.26	23.28
23	2017	IR64	35	85.0	15.0	22.26	22.12	21.96
24	2017	IR64	37	85.0	15.0	21.70	21.64	21.72
25	2017	IR64	CK	NaN	NaN	23.54	23.60	22.72
26	2017	LY27	33	75.0	8.0	23.80	24.20	24.00
27	2017	LY27	35	75.0	8.0	23.00	22.60	22.20
28	2017	LY27	37	75.0	8.0	22.20	23.40	22.80
29	2017	LY27	33	75.0	15.0	23.60	23.60	22.90
..	...	...	..	...	...	...	...	...
48	2017	LY6	33	85.0	15.0	23.00	23.80	23.60
49	2017	LY6	35	85.0	15.0	22.80	22.60	23.40
50	2017	LY6	37	85.0	15.0	22.80	22.40	22.80
51	2017	LY6	CK	NaN	NaN	24.80	24.20	24.60
52	2017	R168	33	75.0	8.0	21.18	21.30	21.16
53	2017	R168	35	75.0	8.0	20.72	21.34	21.08
54	2017	R168	37	75.0	8.0	21.46	21.26	21.56
55	2017	R168	33	75.0	15.0	20.44	20.78	21.06
56	2017	R168	35	75.0	15.0	20.78	20.42	20.60
57	2017	R168	37	75.0	15.0	19.78	19.58	19.60
58	2017	R168	33	85.0	8.0	20.92	20.96	20.66
59	2017	R168	35	85.0	8.0	21.84	22.00	21.62
60	2017	R168	37	85.0	8.0	20.84	21.12	20.80
61	2017	R168	33	85.0	15.0	21.12	21.24	21.40
62	2017	R168	35	85.0	15.0	20.60	20.78	20.90
63	2017	R168	37	85.0	15.0	20.10	20.32	20.50
64	2017	R168	CK	NaN	NaN	22.00	22.16	22.36
65	2017	ZLY47	33	75.0	8.0	25.98	25.52	25.48
66	2017	ZLY47	35	75.0	8.0	23.96	23.90	24.00
67	2017	ZLY47	37	75.0	8.0	23.88	23.26	23.24
68	2017	ZLY47	33	75.0	15.0	25.24	24.90	24.46
69	2017	ZLY47	35	75.0	15.0	23.54	23.90	23.98
70	2017	ZLY47	37	75.0	15.0	22.40	22.74	22.46
71	2017	ZLY47	33	85.0	8.0	26.08	24.98	25.82
72	2017	ZLY47	35	85.0	8.0	25.10	24.56	22.86

```
73   2017     ZLY47  37   85.0    8.0  24.48  24.06  24.26
74   2017     ZLY47  33   85.0   15.0  26.62  26.48  26.82
75   2017     ZLY47  35   85.0   15.0  24.32  24.30  24.40
76   2017     ZLY47  37   85.0   15.0  23.32  23.70  23.36
77   2017     ZLY47  CK    NaN    NaN  25.84  26.06  26.24

[78 rows x 8 columns]
```

This dataset had 78 observations and 8 variables including: Year, Cultivars, Tem which is Temperature/°C, RH which is relative humidity in %, Days, R1, R2 and R3.

In [14]:
```
#summary statistic of numerical variables
data.describe()
```

Out[14]:

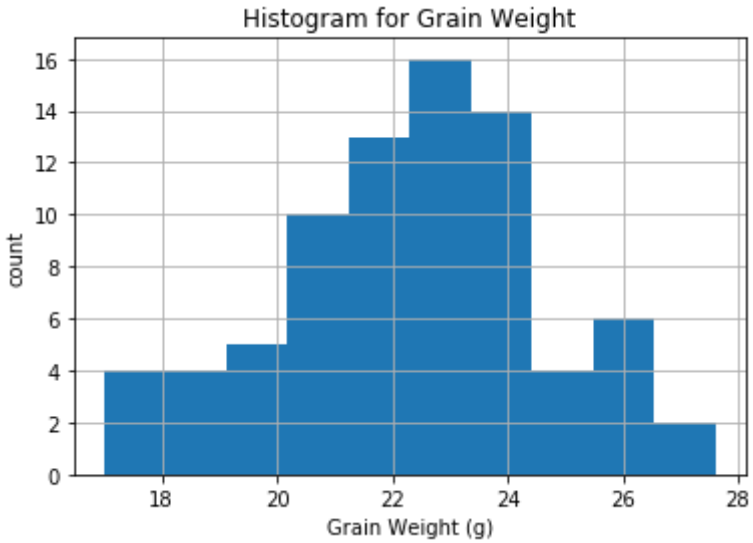
	Year	RH	Days	R1	R2	R3
count	78.0	72.000000	72.000000	78.000000	78.000000	78.000000
mean	2017.0	80.000000	11.500000	22.355684	22.347265	22.322308
std	0.0	5.035088	3.524562	2.294811	2.193554	2.259374
min	2017.0	75.000000	8.000000	17.010000	17.220000	17.000000
25%	2017.0	75.000000	8.000000	20.860000	21.075000	20.940000
50%	2017.0	80.000000	11.500000	22.600000	22.600000	22.430000
75%	2017.0	85.000000	15.000000	23.675000	23.800000	23.585000
max	2017.0	85.000000	15.000000	27.600000	27.000000	28.000000

The mean for R1 = 22.355684 g.

In [33]:
```
import matplotlib.pyplot as plt
```

In [34]:
```
data.R1.hist()
# Add title and axis names
plt.title('Histogram for Grain Weight')
plt.xlabel('Grain Weight (g)')
plt.ylabel('count')
```

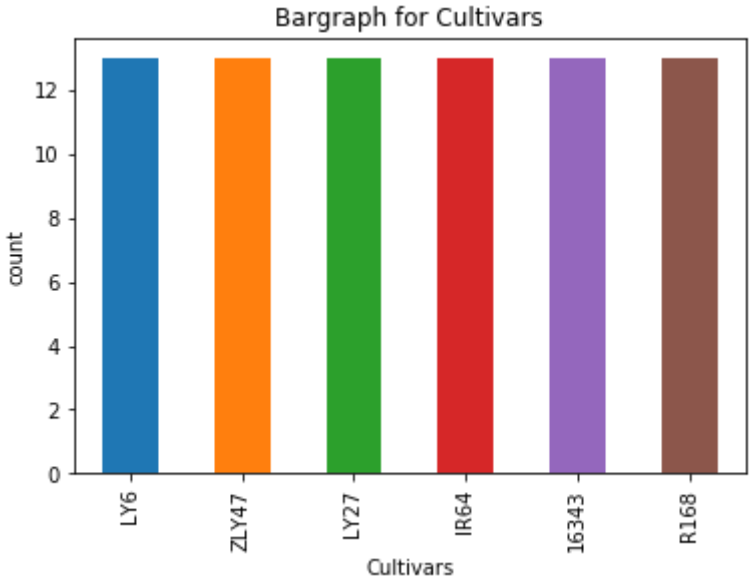
Out[34]: Text(0,0.5,'count')



The distribution for R1 grain weight is normal.

In [37]:
```
#summary statistic of Cultivars
counts = data['Cultivars'].value_counts()
print(counts)
#bargraph
counts.plot(kind='bar')
plt.title('Bargraph for Cultivars')
plt.xlabel('Cultivars')
plt.ylabel('count')
```

```
LY6      13
ZLY47    13
LY27     13
IR64     13
16343    13
R168     13
Name: Cultivars, dtype: int64
```



There were 13 counts of each type of Cultivars which can also be seen in the bar graph where the counts are all even/the same.

In [ ]: