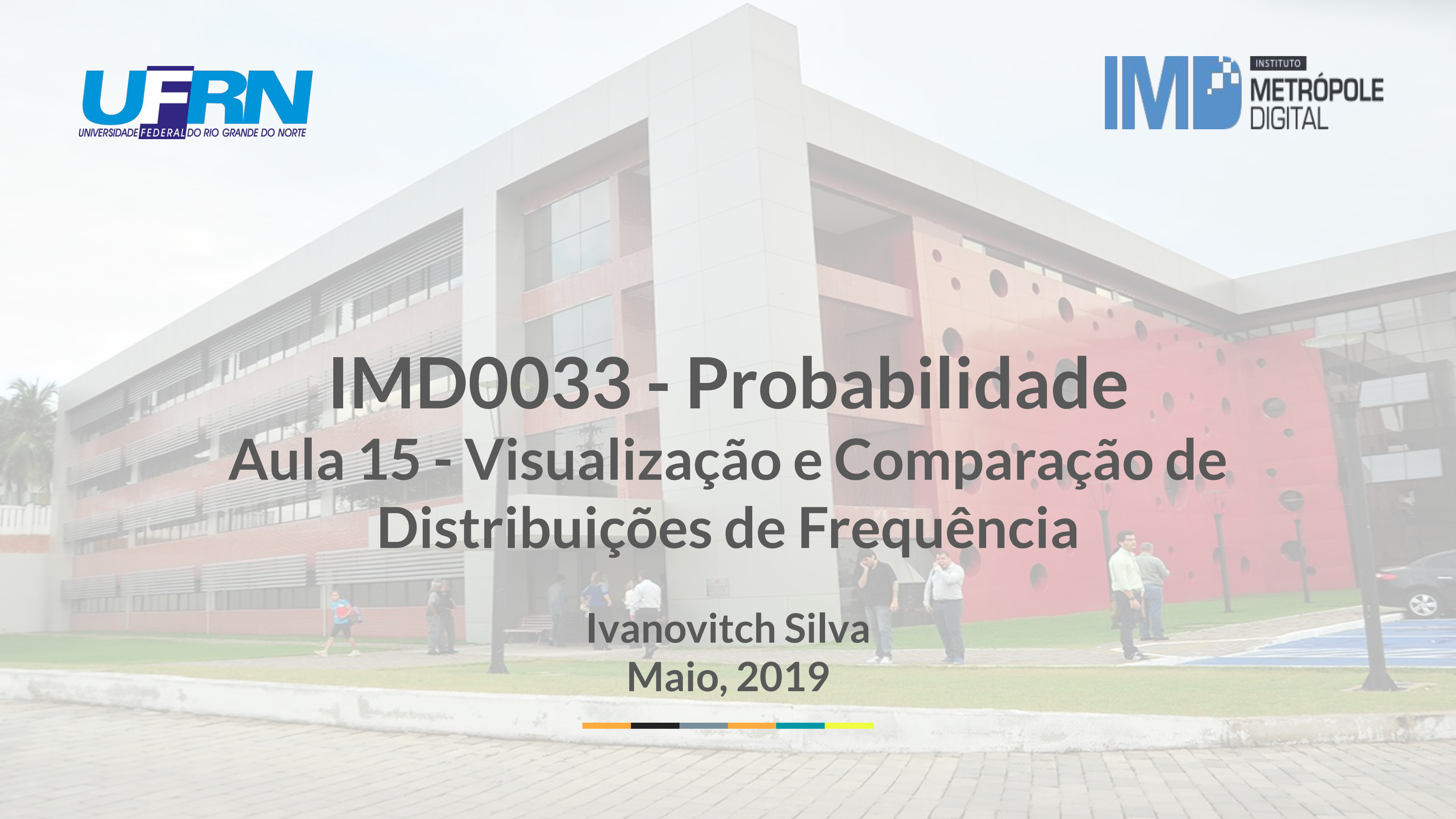


IMD0033 - Probabilidade

Aula 15 - Visualização e Comparação de Distribuições de Frequência

Ivanovitch Silva
Maio, 2019

Agenda (Parte I)

- Visualizando distribuições
- Gráficos de barra, pizza e histogramas
- Assimetria
- Distribuições simétricas

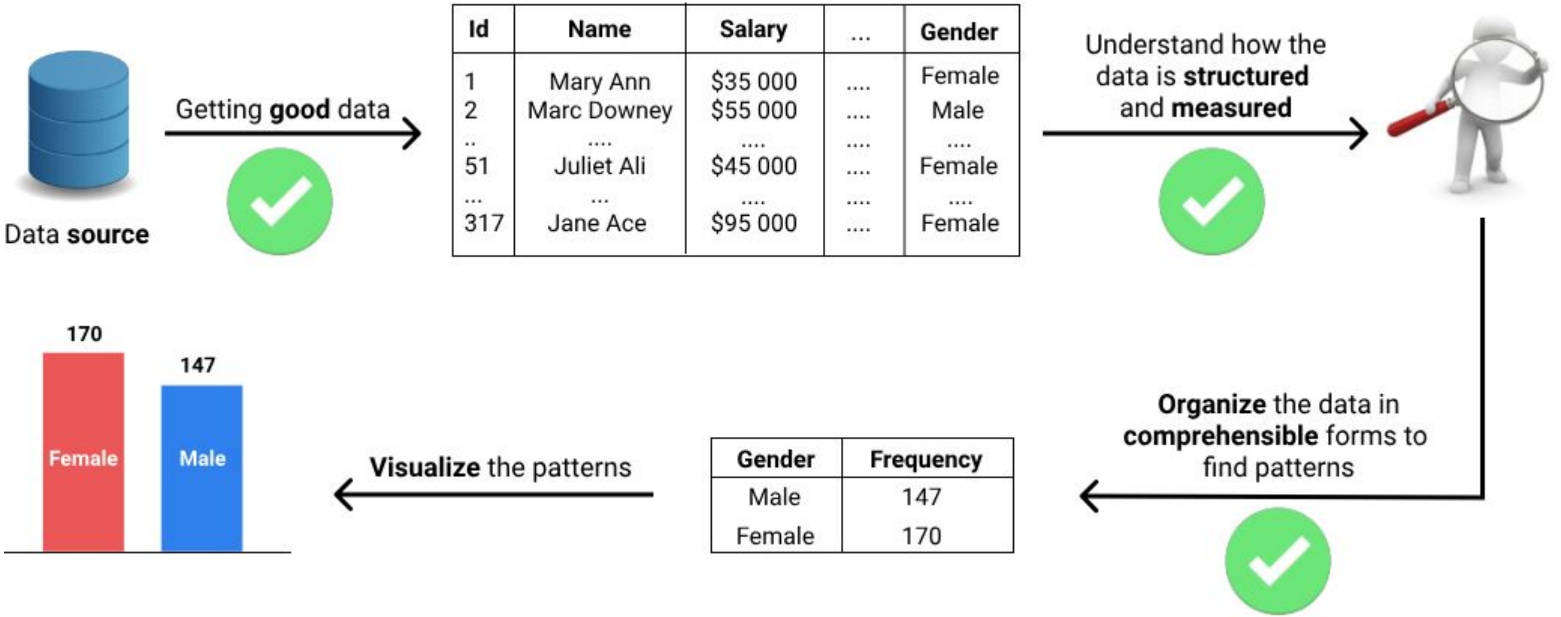
Atualizar o repositório

```
git clone https://github.com/ivanovitchm/imd0033_2019_1.git
```

Ou ....

```
git pull
```

PREVIOUSLY ON...



Data **source**

Getting **good** data

Id	Name	Salary	...	Gender
1	Mary Ann	$35 000	....	Female
2	Marc Downey	$55 000	....	Male
..	....	....	....	....
51	Juliet Ali	$45 000	....	Female
...	...	....	....	....
317	Jane Ace	$95 000	....	Female

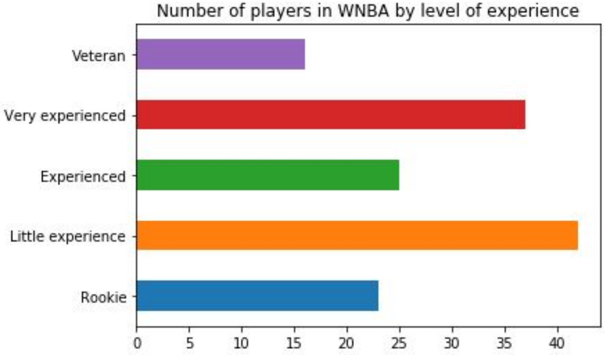
Understand how the data is **structured** and **measured**

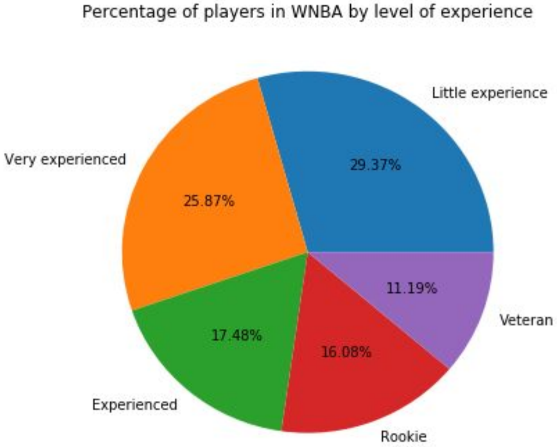
Organize the data in **comprehensible** forms to find patterns

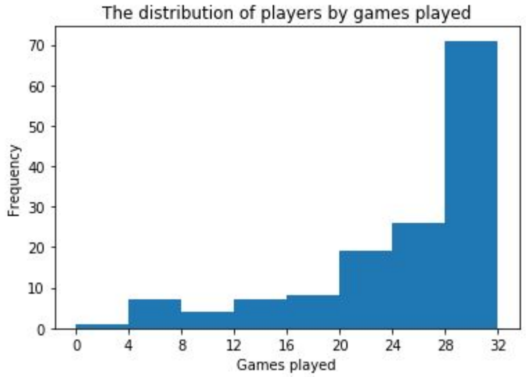
Gender	Frequency
Male	147
Female	170

Visualize the patterns

Visualizing Distributions



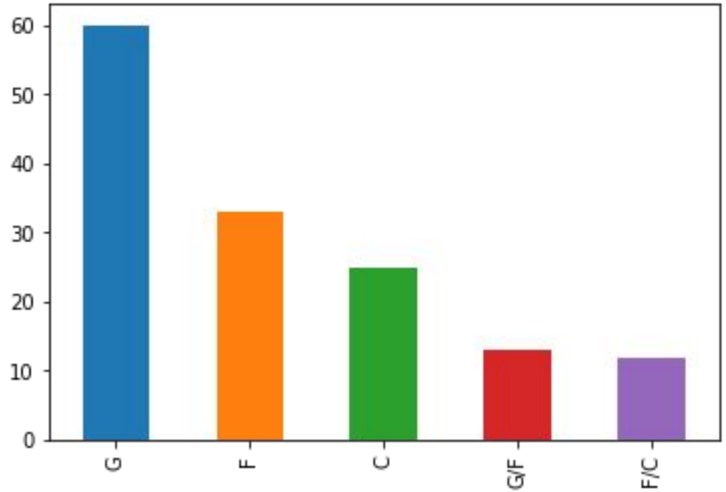




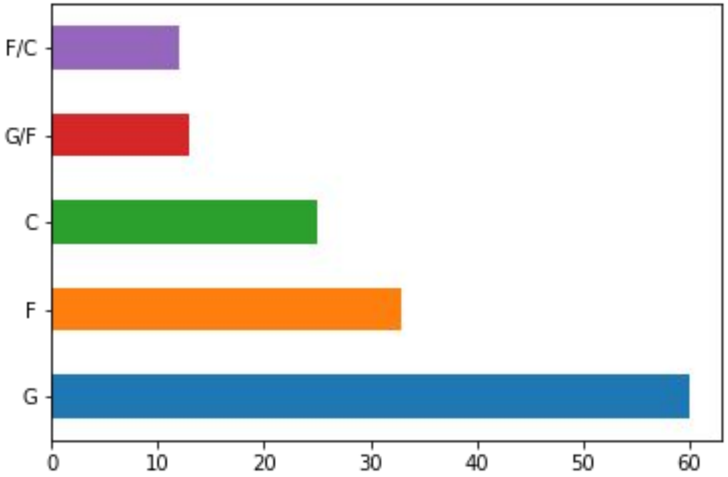
Graphs make easy to scan and compare frequencies, providing us with a single picture of the entire distribution of a variable (**nominal** or **ordinal scale**)

Bar Plots

horizontal bar plots are ideal to use when the labels of the unique values are long

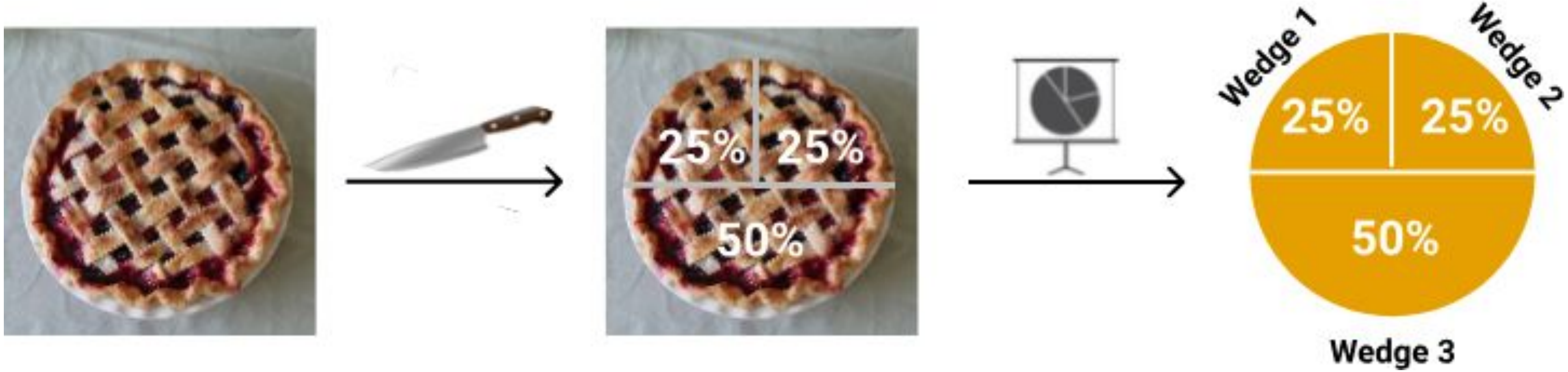


```
wnba['Pos'].value_counts().plot.bar()
```

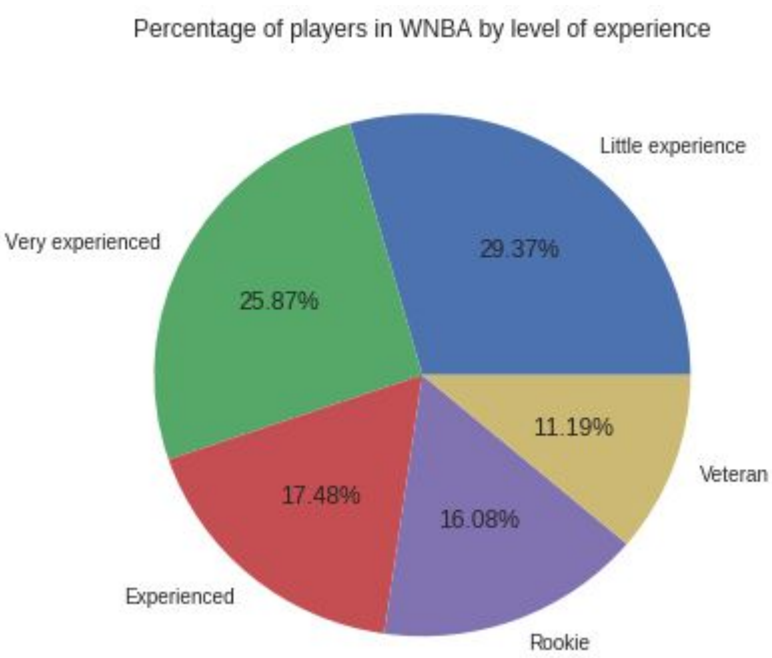


```
wnba['Pos'].value_counts().plot.barh()
```

Pie Charts

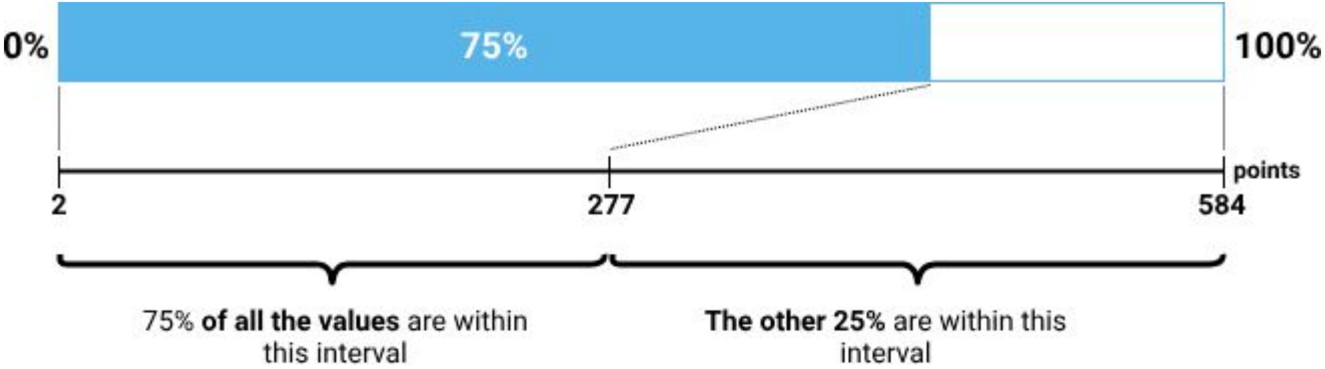


Pie Charts



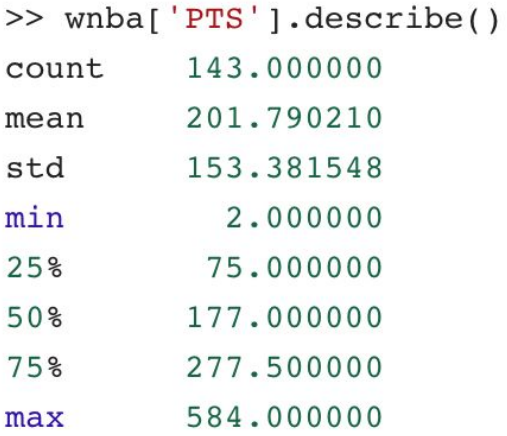
```
wnba['Exp_ordinal'].value_counts().\
plot.pie(figsize = (6,6),
         autopct = '%.2f%%',
         title = 'Percentage of players in \
         WNBA by level of experience')
plt.ylabel('')
```

Histograms



75% **of all the values** are within this interval

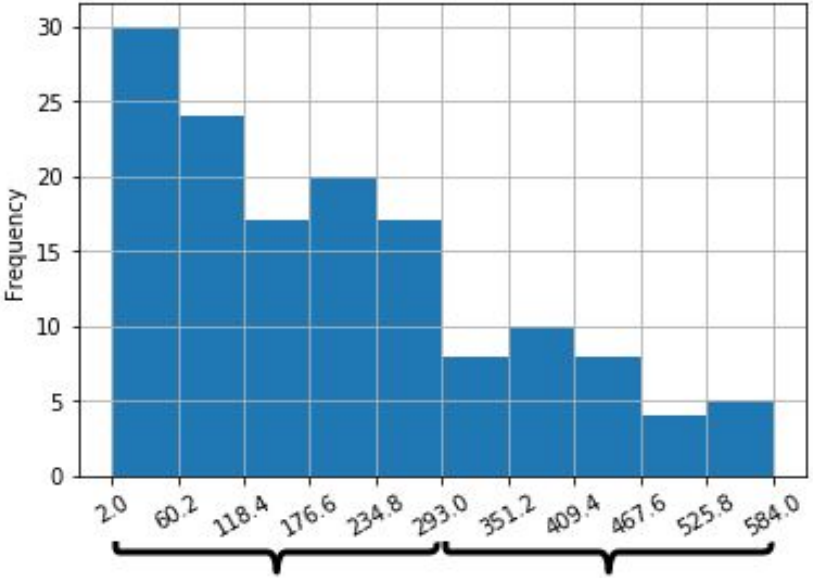
The other 25% are within this interval



```
>> wnba['PTS'].describe()
count    143.000000
mean     201.790210
std      153.381548
min        2.000000
25%       75.000000
50%      177.000000
75%      277.500000
max      584.000000
```

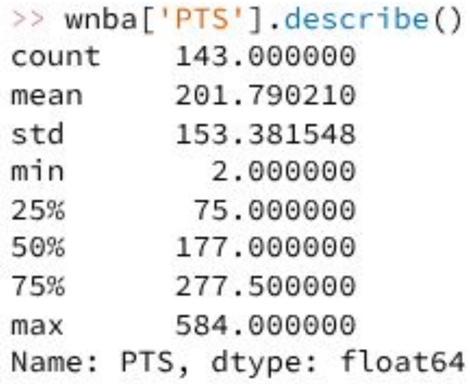
We can see that 75% of the values are distributed within a relatively narrow interval (between 2 and 277), while the remaining 25% are distributed in an interval that's slightly larger.

The Statistics Behind Histograms



We can **see immediately** that roughly **three quarters** **(75%)** of the values are within this interval

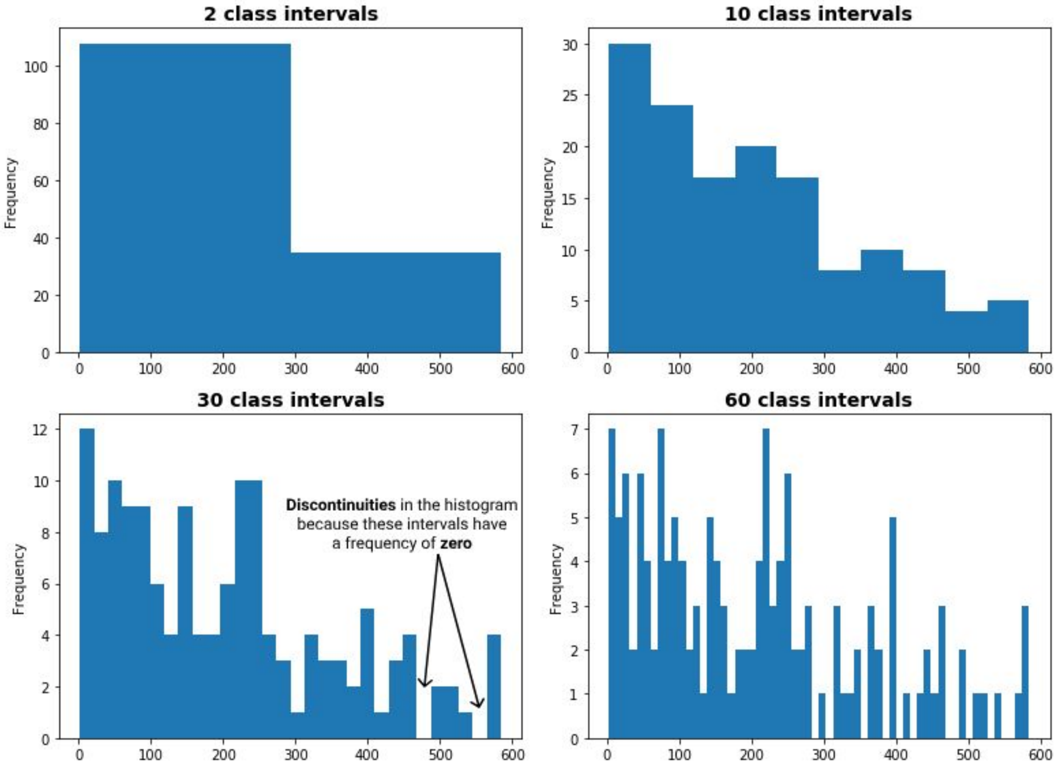
The **remaining quarter** is within this interval

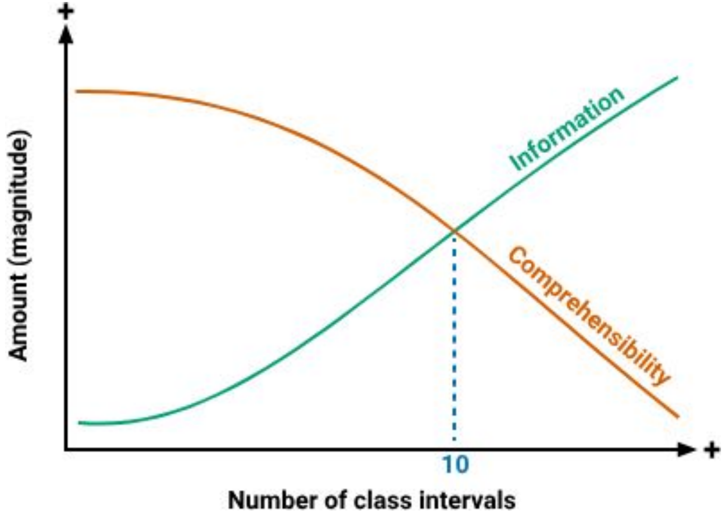


```
>> wnba['PTS'].describe()
count     143.000000
mean      201.790210
std       153.381548
min         2.000000
25%        75.000000
50%       177.000000
75%       277.500000
max       584.000000
Name: PTS, dtype: float64
```

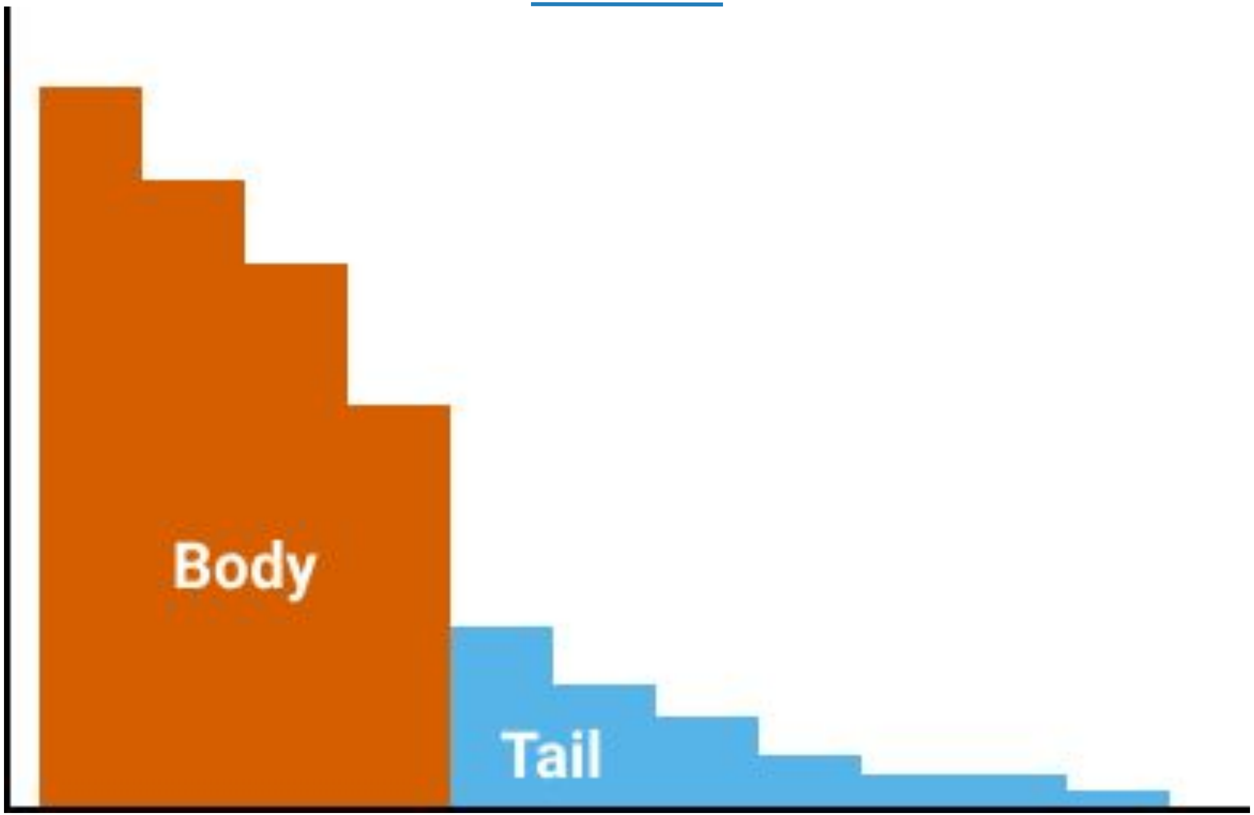
```
>> wnba['PTS'].plot.hist()
```

Binning for Histograms

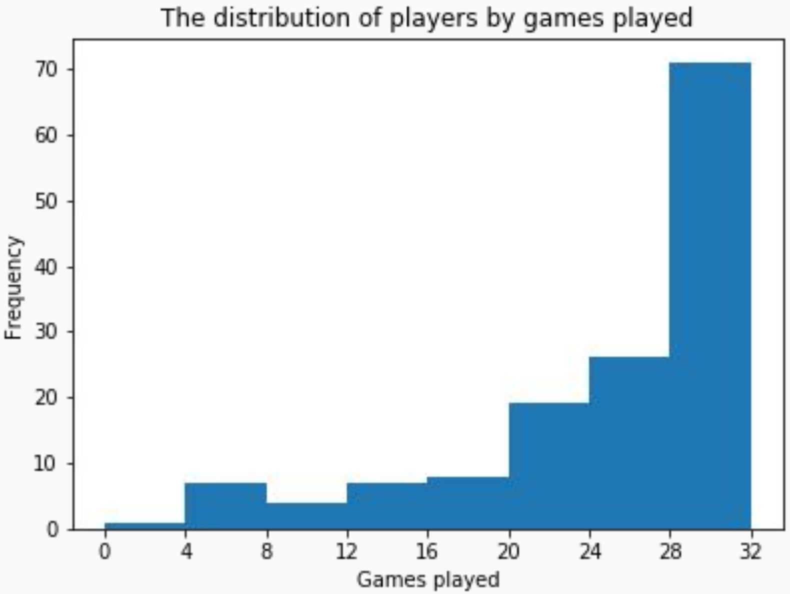


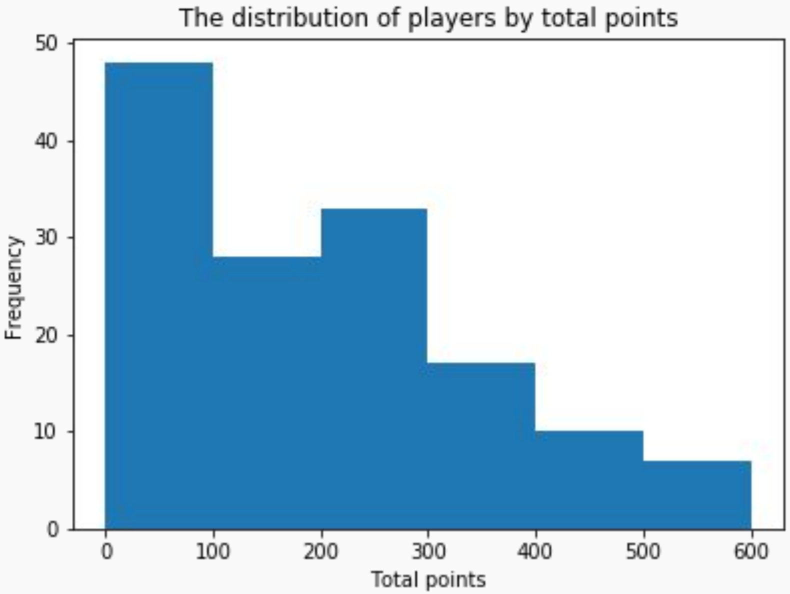


Skewed Distributions

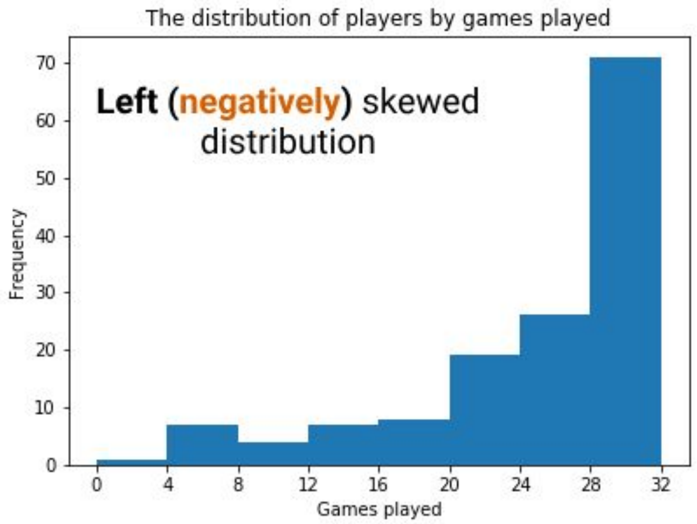


Skewed Distributions

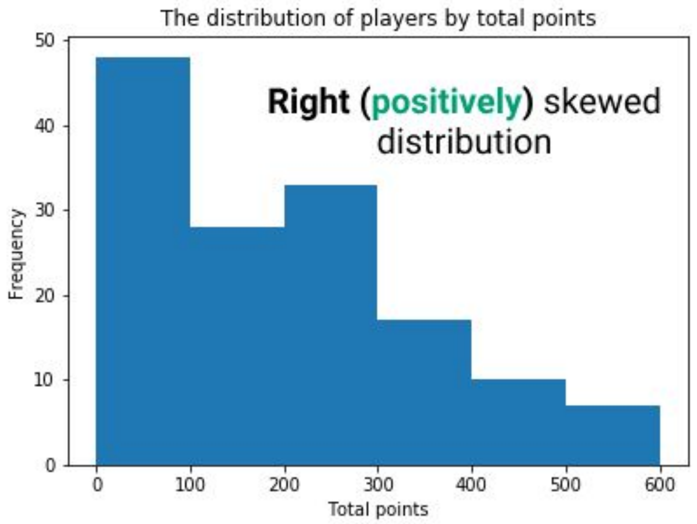




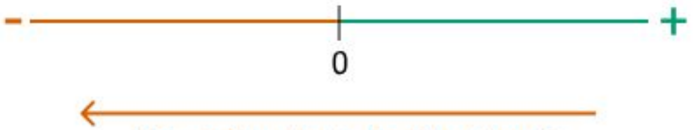
Skewed Distributions



Left (negatively) skewed distribution



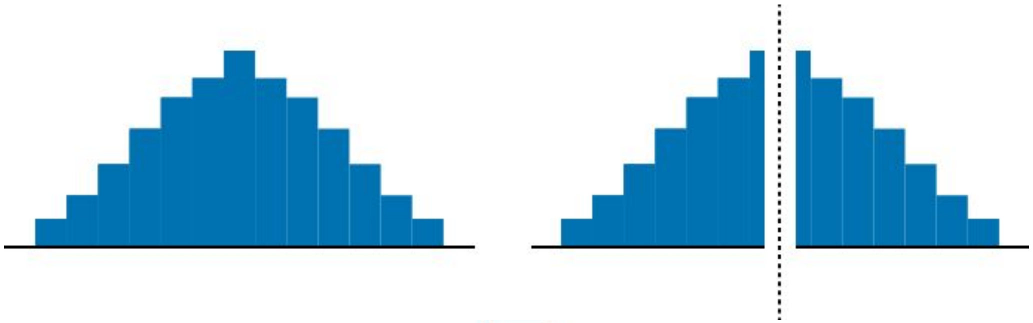
Right (positively) skewed distribution

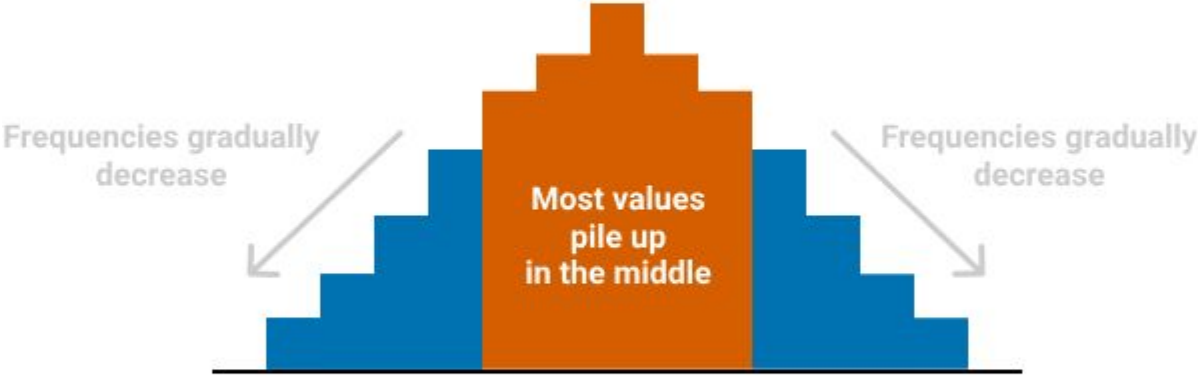


If the **tail** points in the direction of **negative numbers** then the distribution is **negatively skewed**

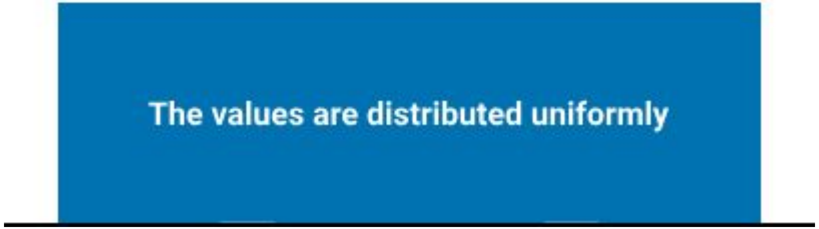
If the **tail** points in the direction of **positive numbers** then the distribution is **positevely skewed**

Symmetrical Distributions

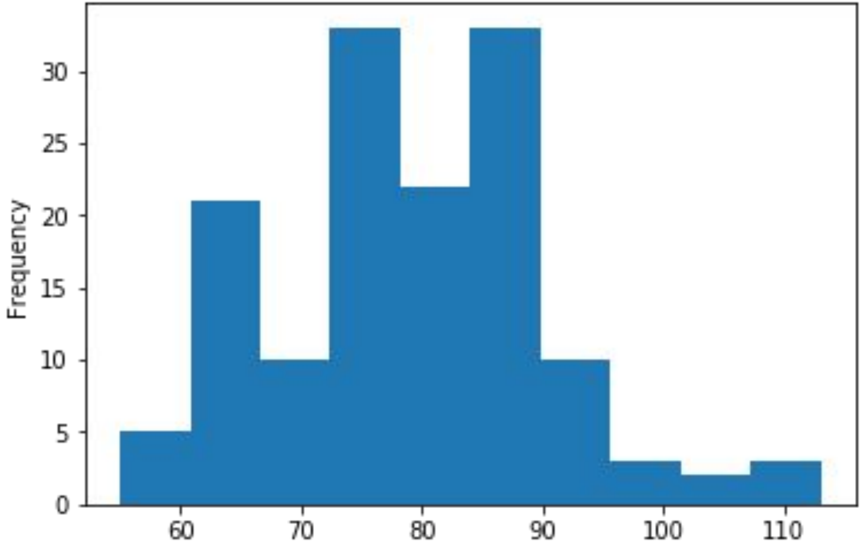


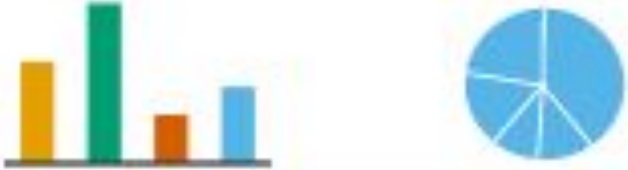
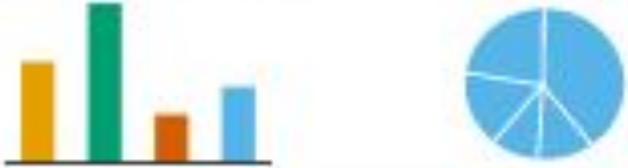
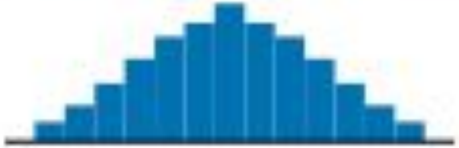
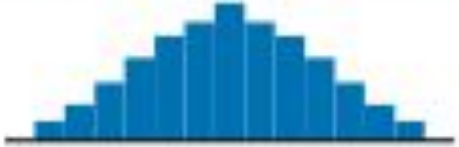


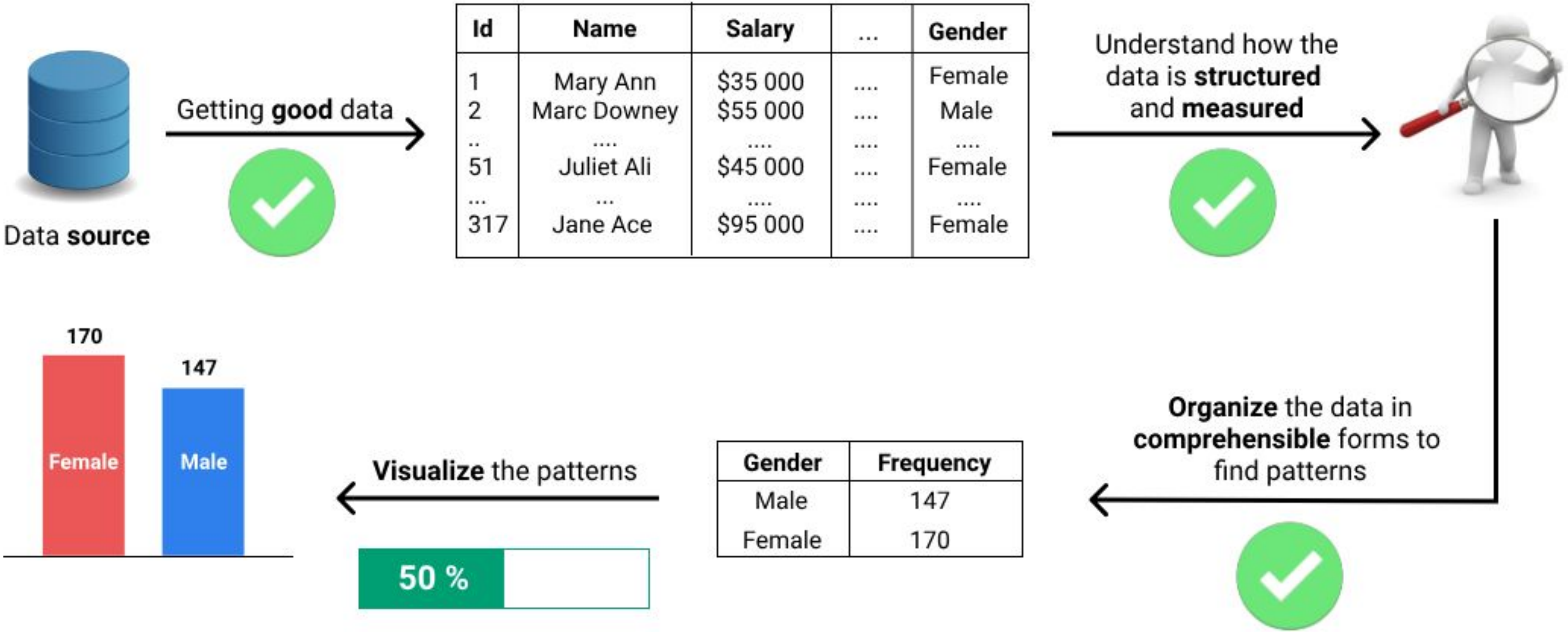
Symmetrical Distribution (uniform)



The values are distributed uniformly



Scale of measurement	Graphs we can use to show the distribution
Nominal	
Ordinal	
Interval	
Ratio	



Data **source**

Getting **good** data

Id	Name	Salary	...	Gender
1	Mary Ann	$35 000	....	Female
2	Marc Downey	$55 000	....	Male
..	....	....	....	....
51	Juliet Ali	$45 000	....	Female
...	...	....	....	....
317	Jane Ace	$95 000	....	Female

Understand how the data is **structured** and **measured**

Organize the data in **comprehensible** forms to find patterns

Gender	Frequency
Male	147
Female	170

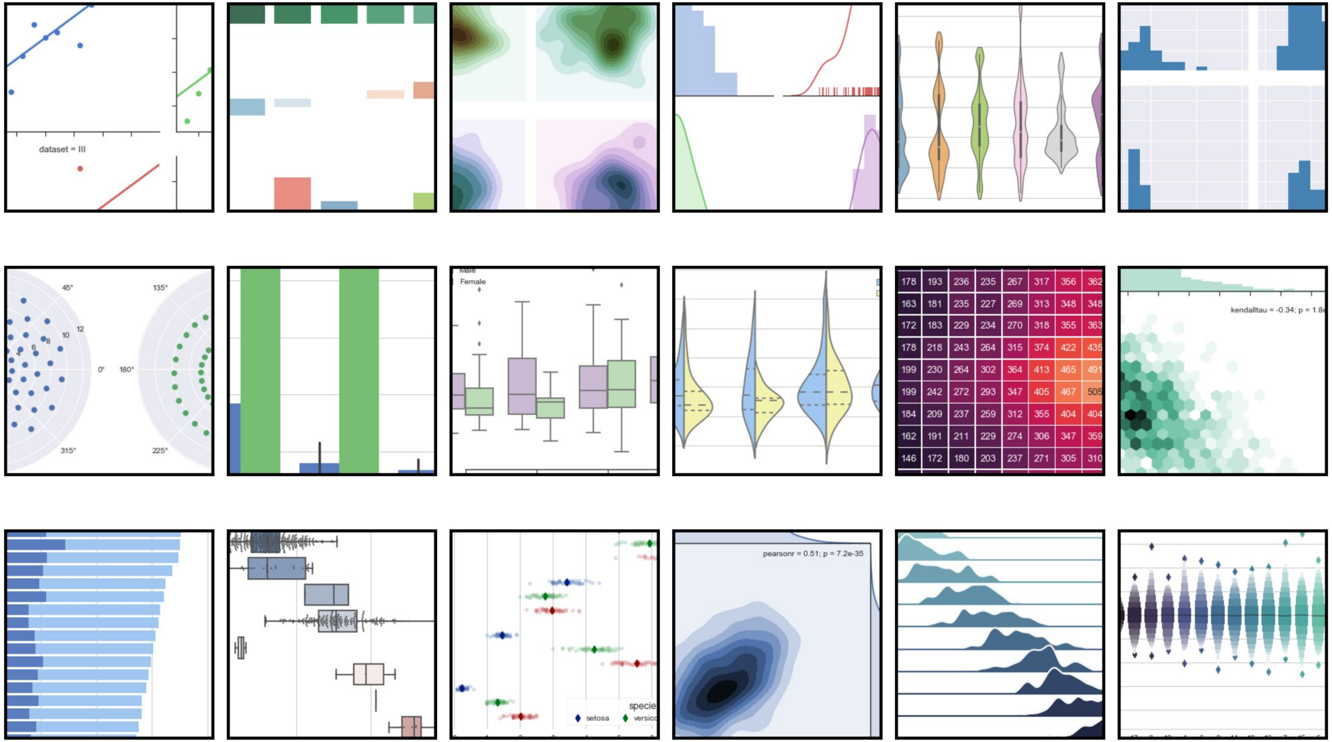
Visualize the patterns

50 %

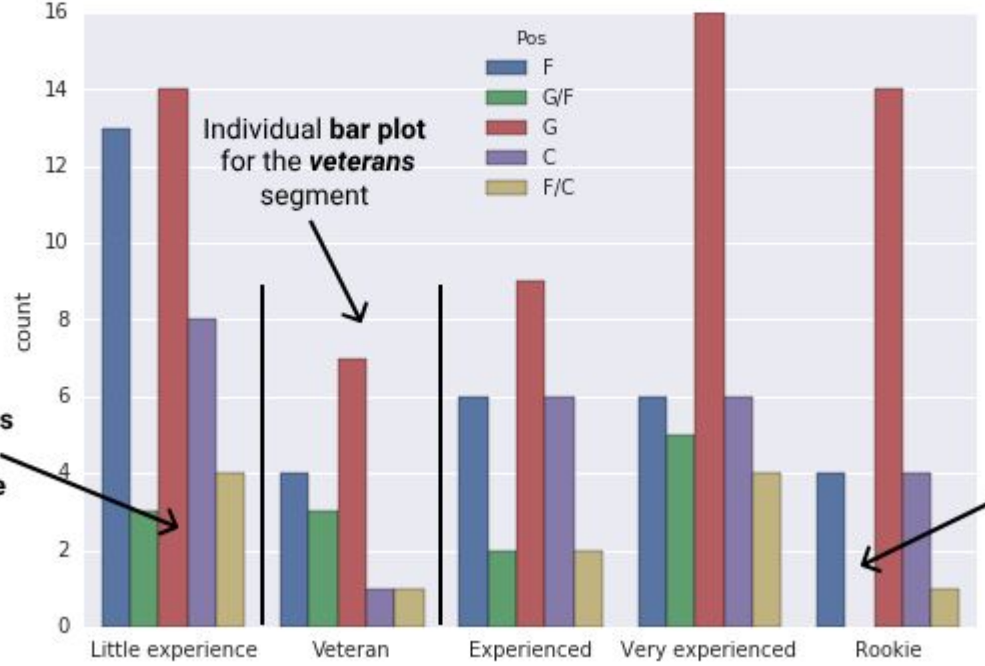
Agenda (Parte II)

- Agrupamentos de gráficos de barras
- Comparando histogramas
- Estimativa de densidade kernel
- Gráficos de faixa e caixa
- Pontos fora da curva

Seaborn


dataset = III
45°
135°
0°
180°
315°
225°
12
10
8
6
4
Male
Female
178 193 236 235 267 317 356 362
163 181 235 227 269 313 348 348
172 183 229 234 270 318 355 363
178 218 243 264 315 374 422 435
199 230 264 302 364 413 465 491
199 242 272 293 347 405 467 505
184 209 237 259 312 355 404 404
162 191 211 229 274 306 347 359
146 172 180 203 237 271 305 310
kendalltau = -0.34; p = 1.8
pearsonr = 0.51; p = 7.2e-35
specie
setosa
versico

Comparing Frequency Distribution



Individual **bar plot** for the ***veterans*** segment

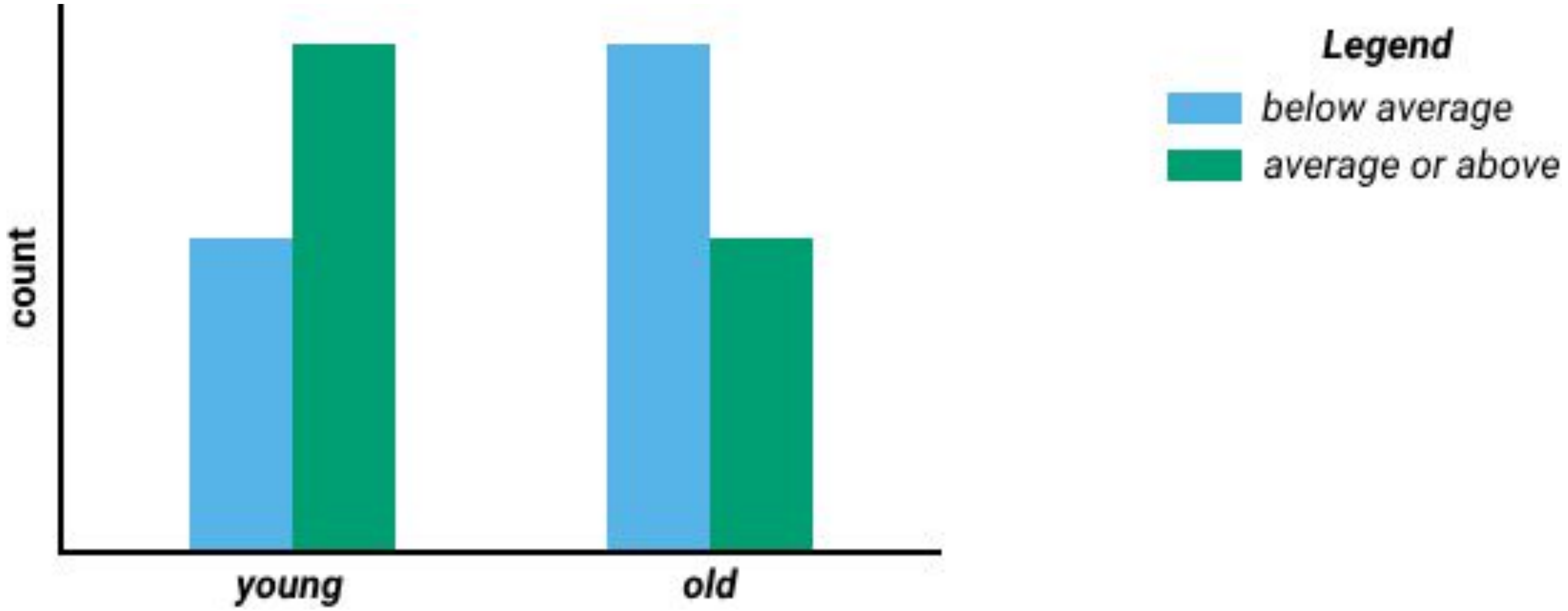
We **don't leave gaps** between bars to **win some space**

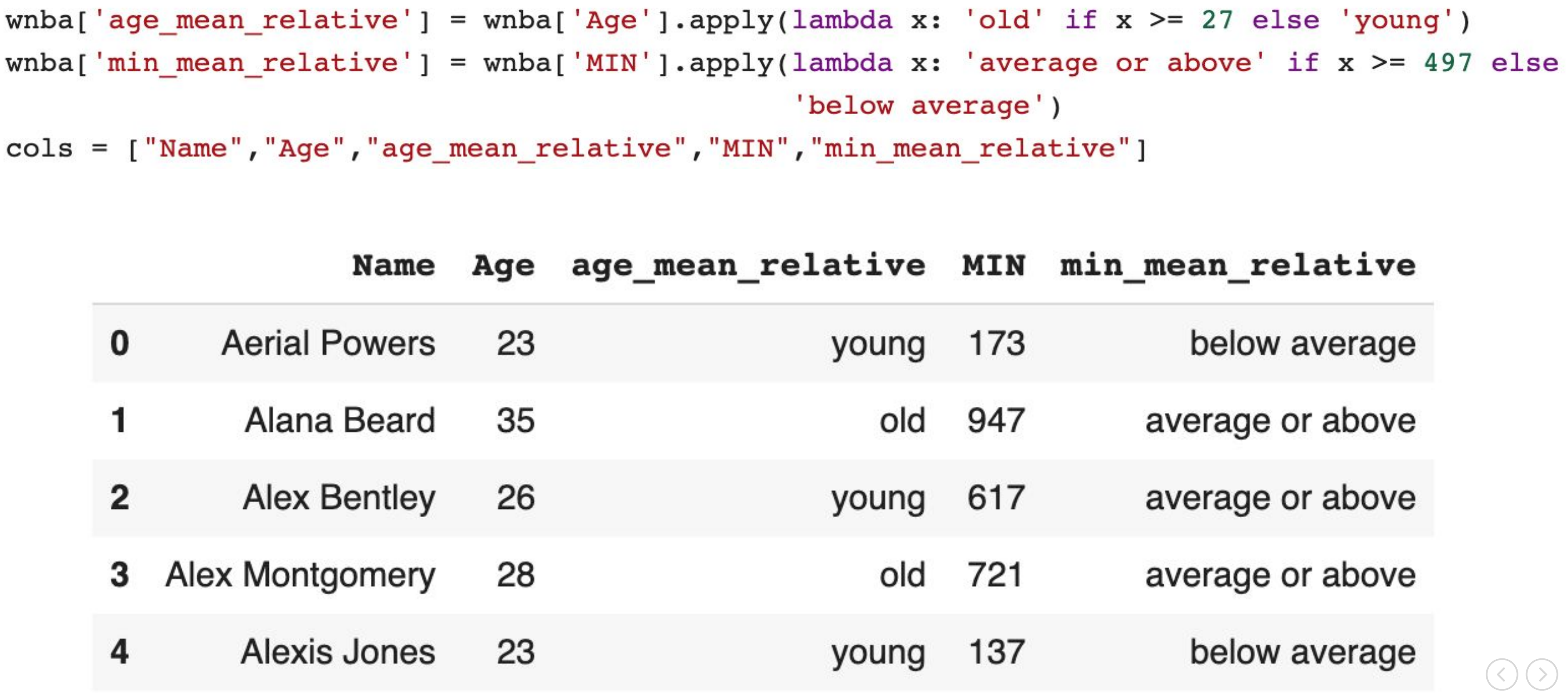
The gap here indicates **zero** frequency

Years in WNBA	Label
0	Rookie
1-3	Little experience
4-5	Experienced
5-10	Very experienced
>10	Veteran

```
sns.countplot(x = 'Exp_ordinal', hue = 'Pos', data = wnba)
```

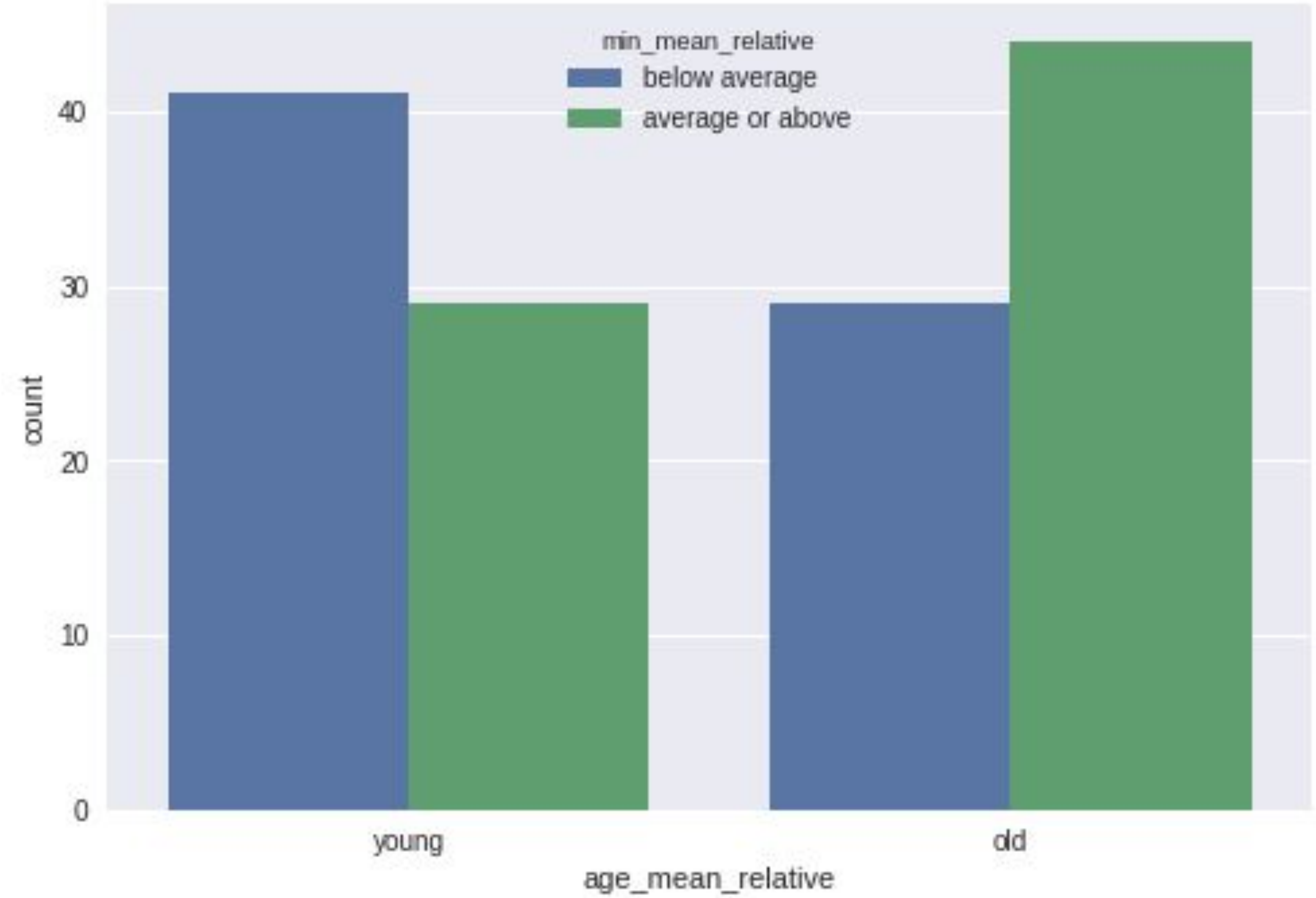
Challenge: Do older players play less?





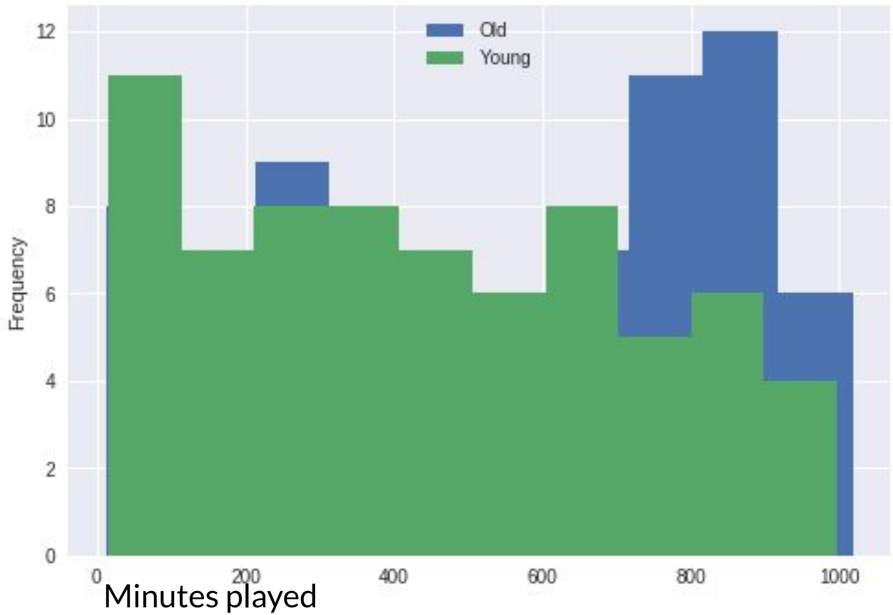
```
wnba['age_mean_relative'] = wnba['Age'].apply(lambda x: 'old' if x >= 27 else 'young')
wnba['min_mean_relative'] = wnba['MIN'].apply(lambda x: 'average or above' if x >= 497 else
                                               'below average')
cols = ["Name","Age","age_mean_relative","MIN","min_mean_relative"]
```

	Name	Age	age_mean_relative	MIN	min_mean_relative
0	Aerial Powers	23	young	173	below average
1	Alana Beard	35	old	947	average or above
2	Alex Bentley	26	young	617	average or above
3	Alex Montgomery	28	old	721	average or above
4	Alexis Jones	23	young	137	below average

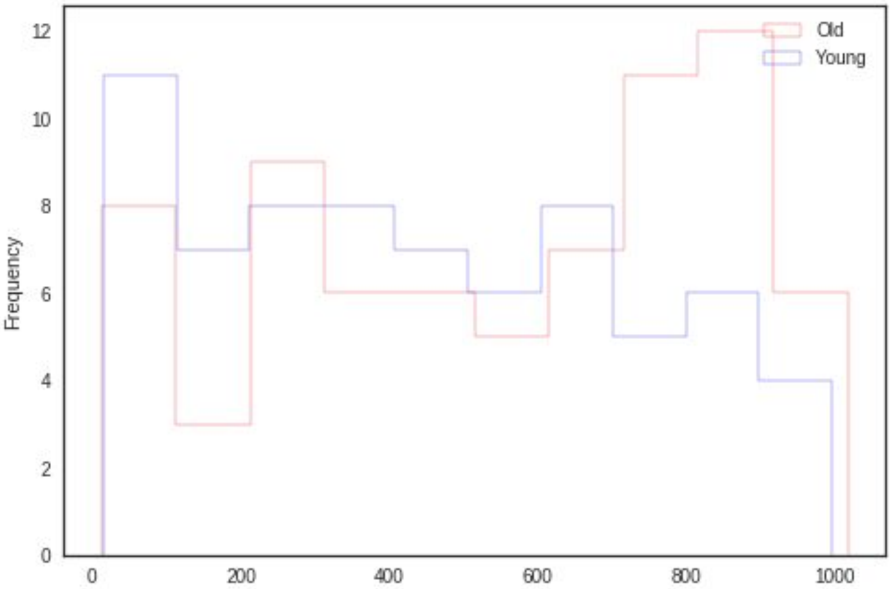


```
sns.countplot(x = 'age_mean_relative', hue = 'min_mean_relative', data = wnba)
```

Comparing Histograms

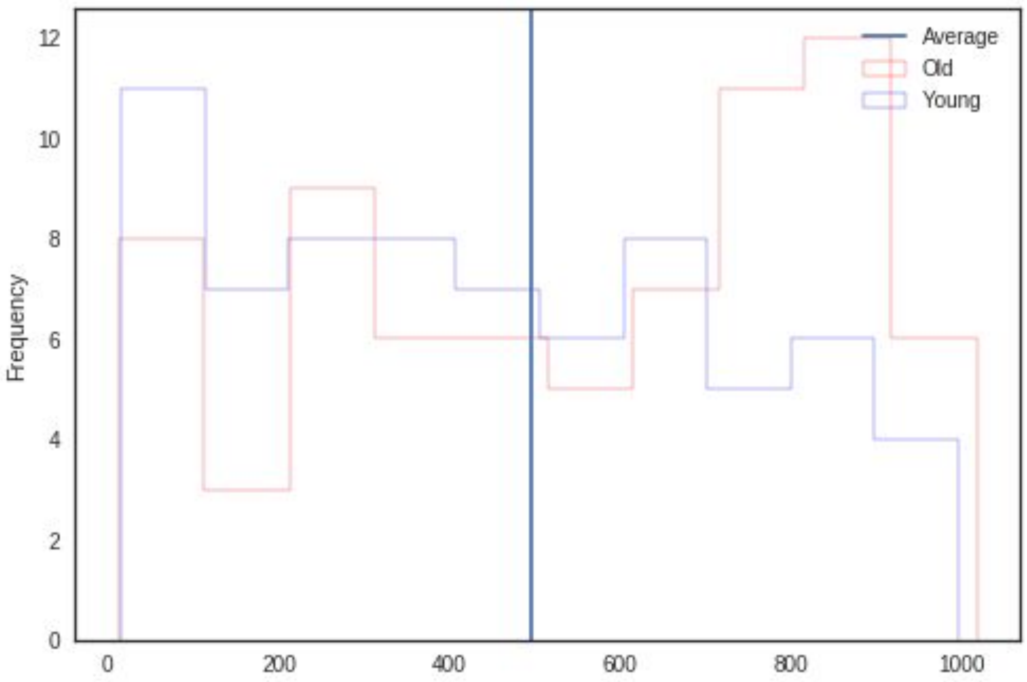


```
wnba[wnba.Age >= 27]['MIN'].plot.hist(label = 'Old', legend = True)
wnba[wnba.Age < 27]['MIN'].plot.hist(label = 'Young', legend = True)
```



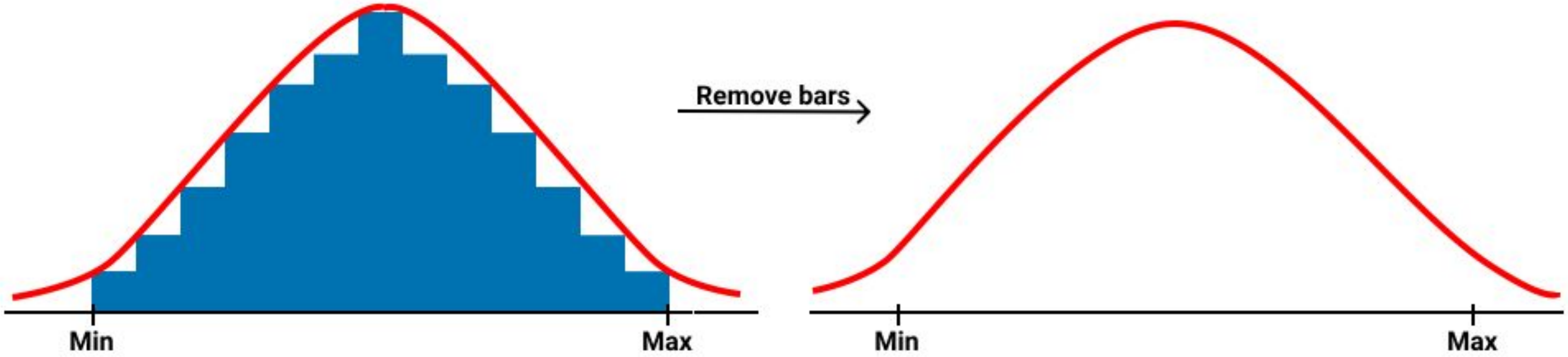
```
sns.set_style("white")
wnba[wnba.Age >= 27]['MIN'].plot.hist(histtype = 'step',
                                       label = 'Old',
                                       legend = True,color="red")
```

Comparing Histograms

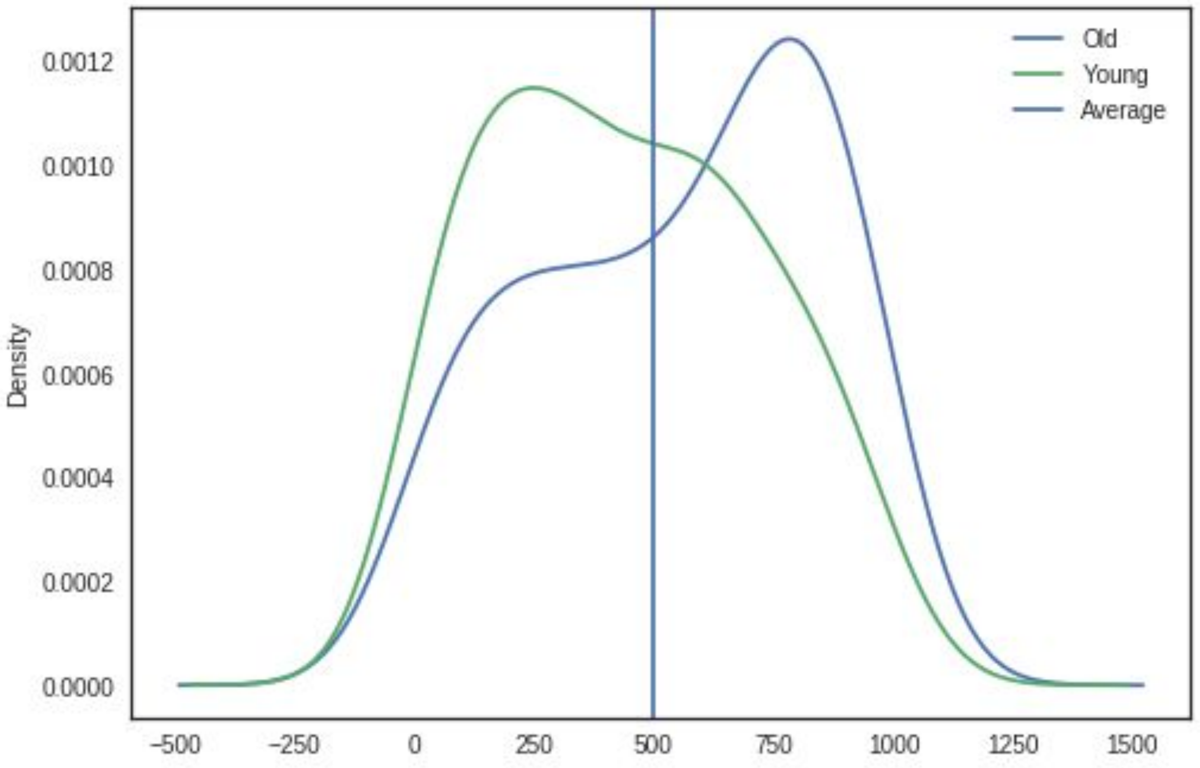


```
plt.axvline(497, label = 'Average')
```

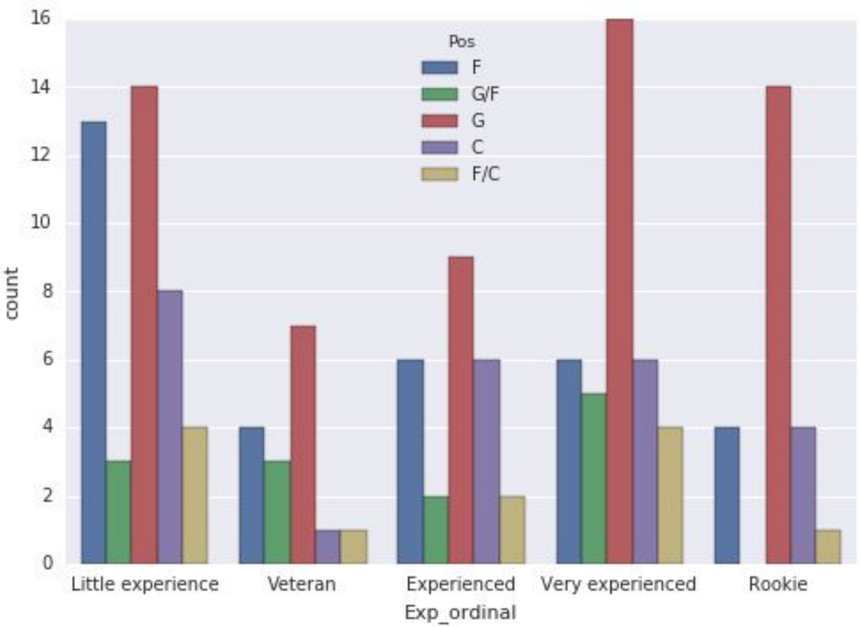
Kernel Density Estimate (KDE) Plots

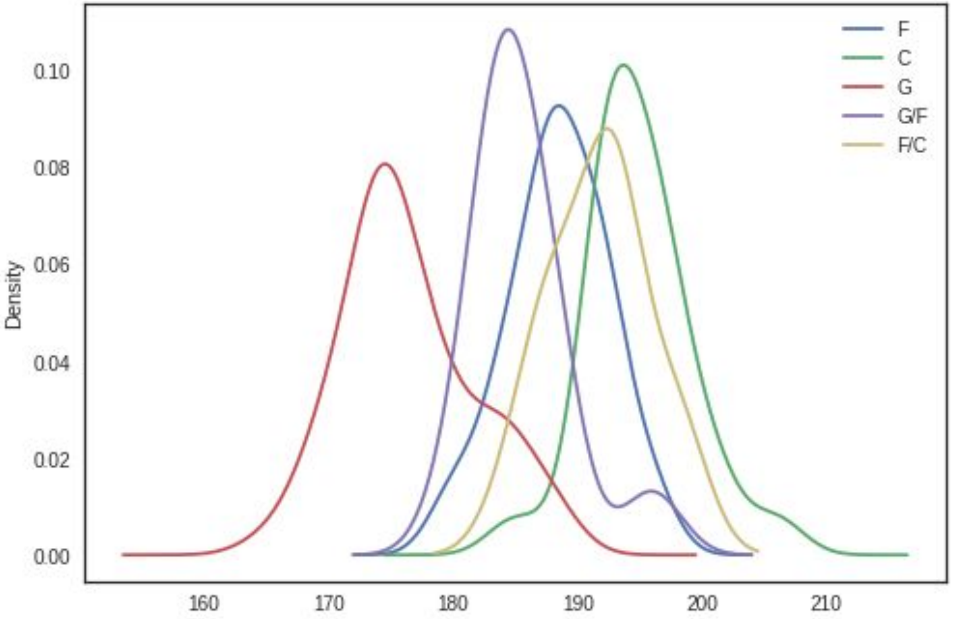


Kernel Density Estimate Plots

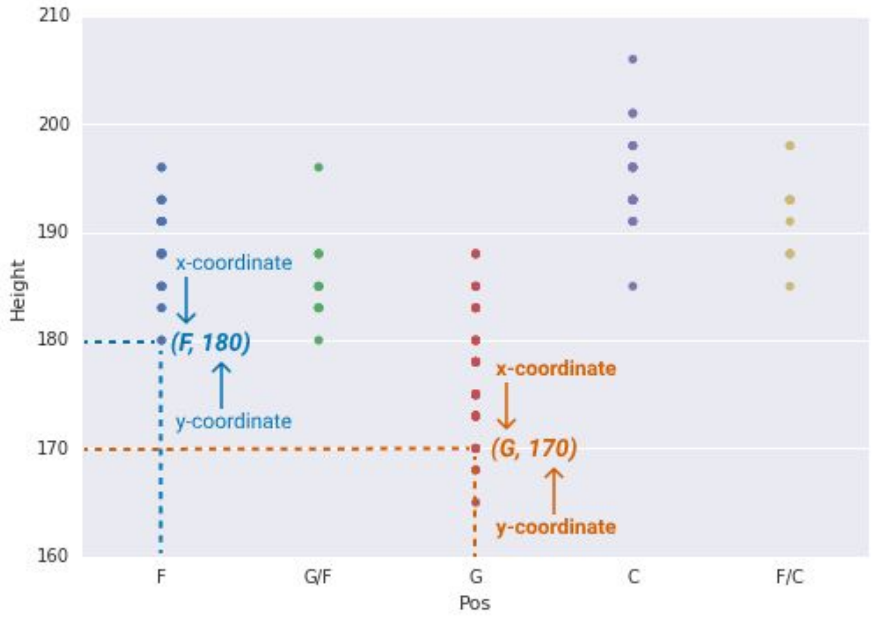


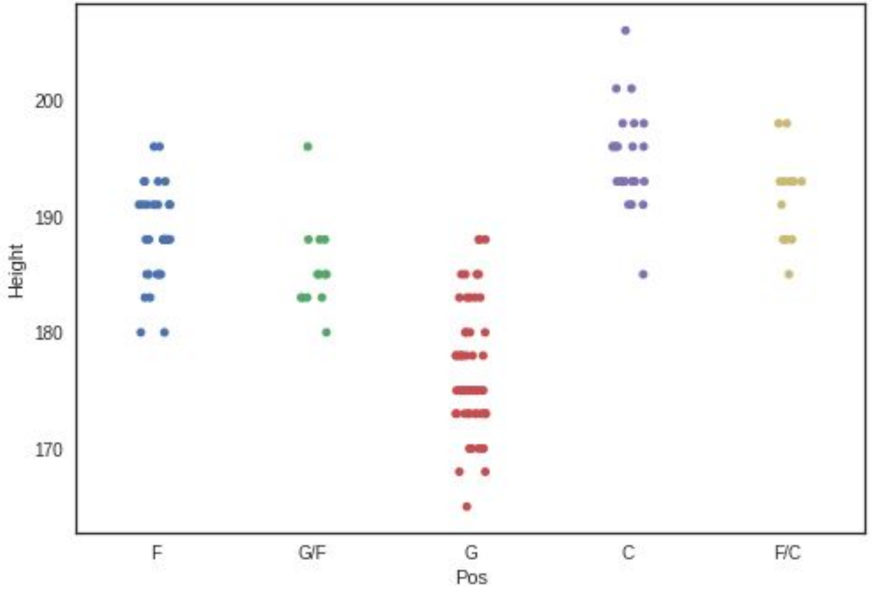
Drawbacks of Kernel Density Plots



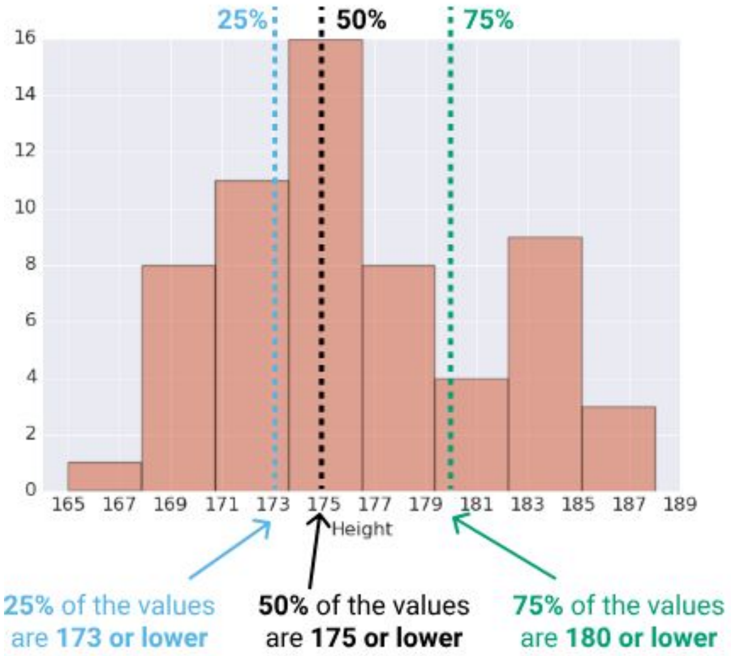


Strip Plots


210
200
190
180
170
160
Height
F
G/F
G
C
F/C
Pos
x-coordinate
(F, 180)
y-coordinate
x-coordinate
(G, 170)
y-coordinate


200
190
180
170
Height
F
G/F
G
C
F/C
Pos

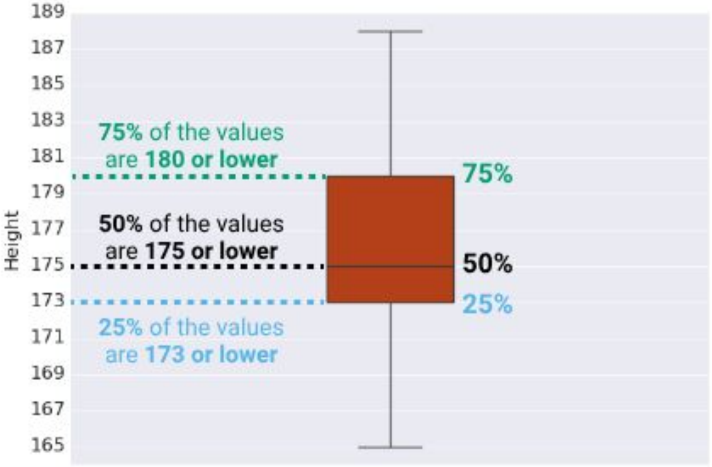
Box Plots



25% of the values are 173 or lower

50% of the values are 175 or lower

75% of the values are 180 or lower

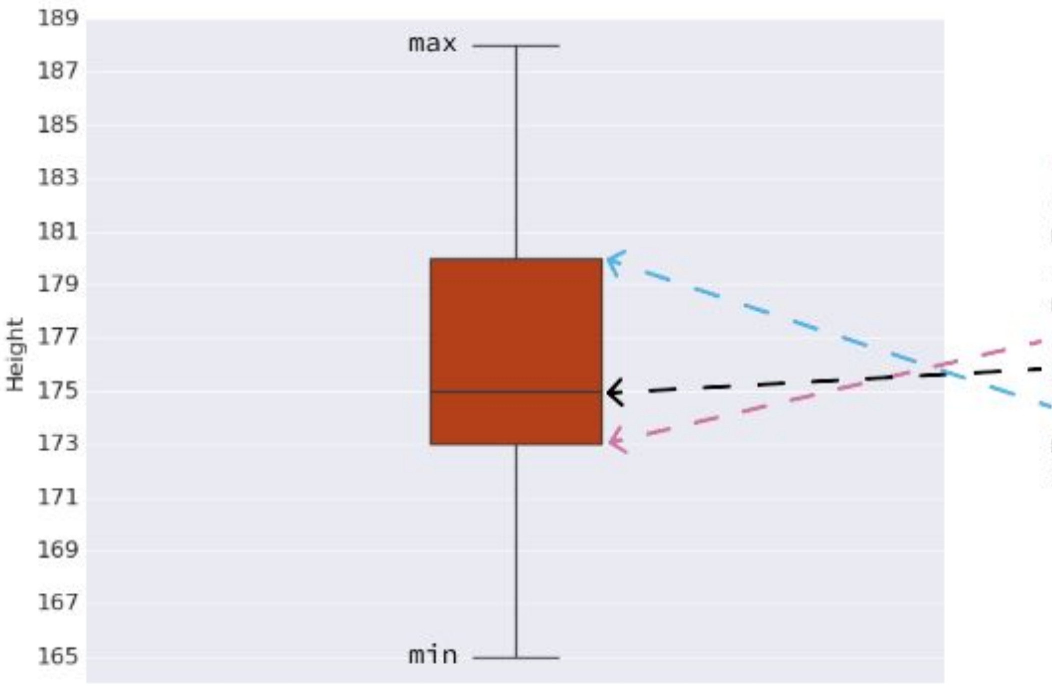


75% of the values are 180 or lower

50% of the values are 175 or lower

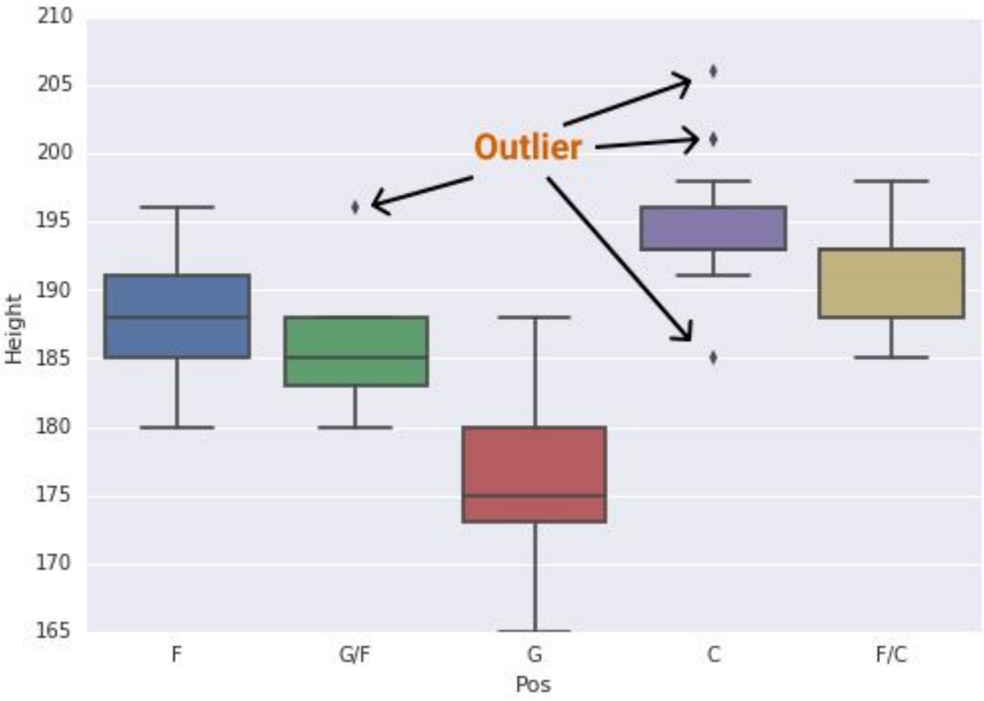
25% of the values are 173 or lower

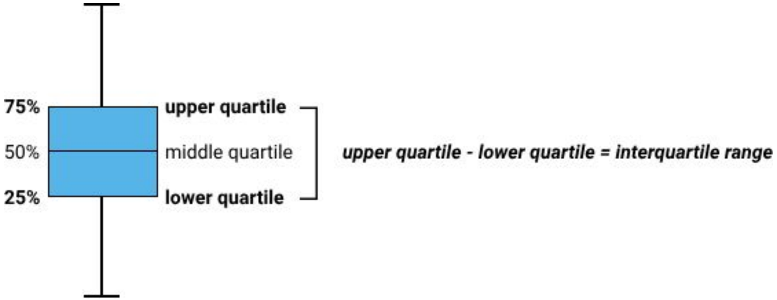
Box Plots



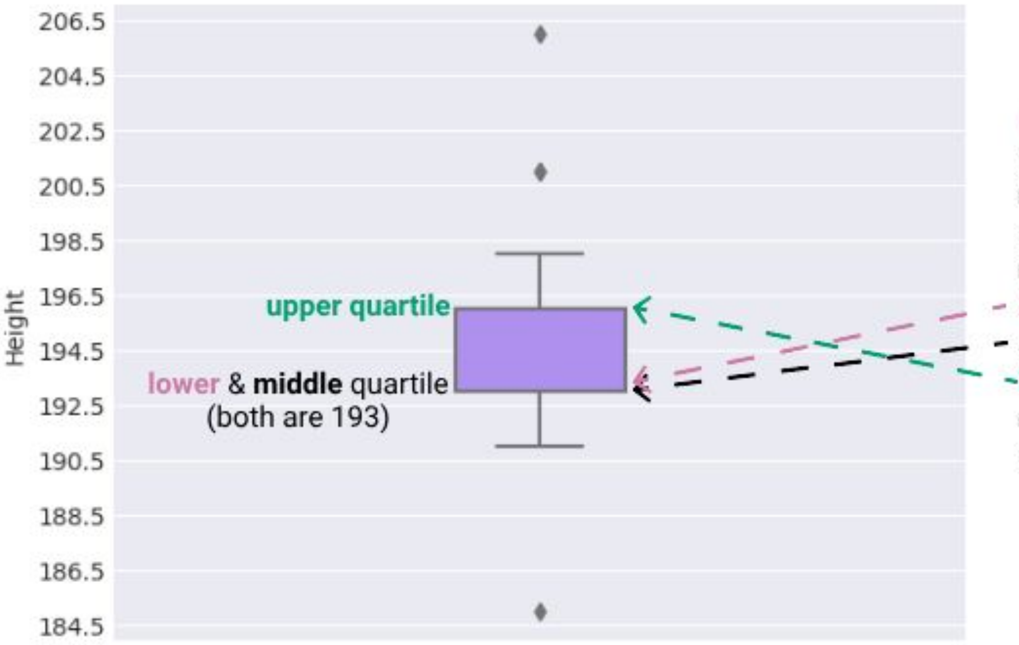
```
>> wnba[wnba['Pos'] == 'G']['Height'].describe()
count        60.000000
mean        176.450000
std           5.312043
min         165.000000
25%         173.000000
50%         175.000000
75%         180.000000
max         188.000000
Name: Height, dtype: float64
```

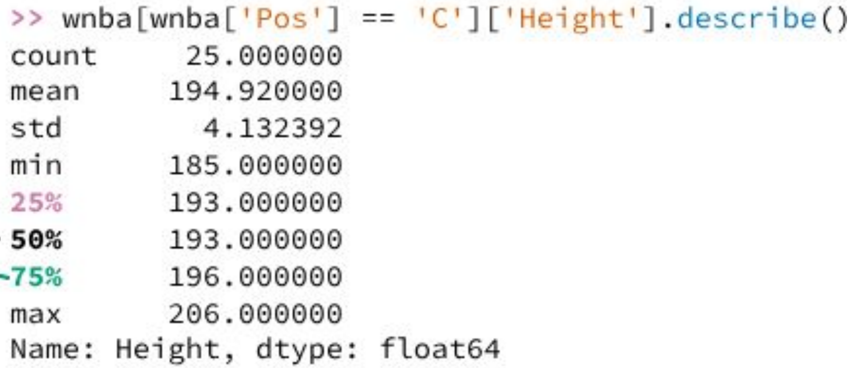
Outliers


Height
210
205
200
195
190
185
180
175
170
165
Outlier
F
G/F
G
C
F/C
Pos


75%
50%
25%
upper quartile
middle quartile
lower quartile
upper quartile - lower quartile = interquartile range

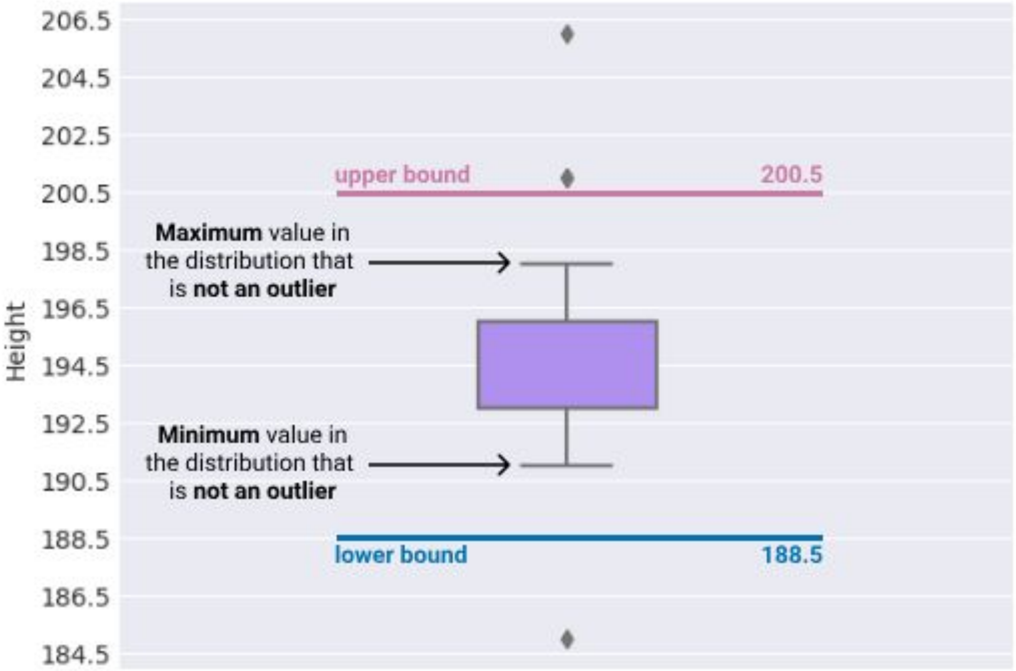
Outliers



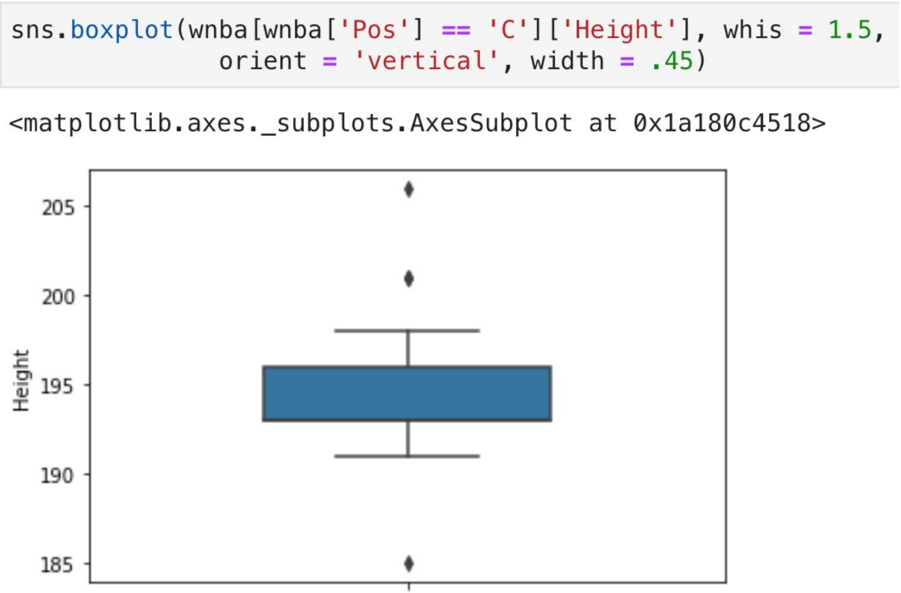


```
>> wnba[wnba['Pos'] == 'C']['Height'].describe()
count     25.000000
mean     194.920000
std        4.132392
min      185.000000
25%      193.000000
50%      193.000000
75%      196.000000
max      206.000000
Name: Height, dtype: float64
```

Outliers

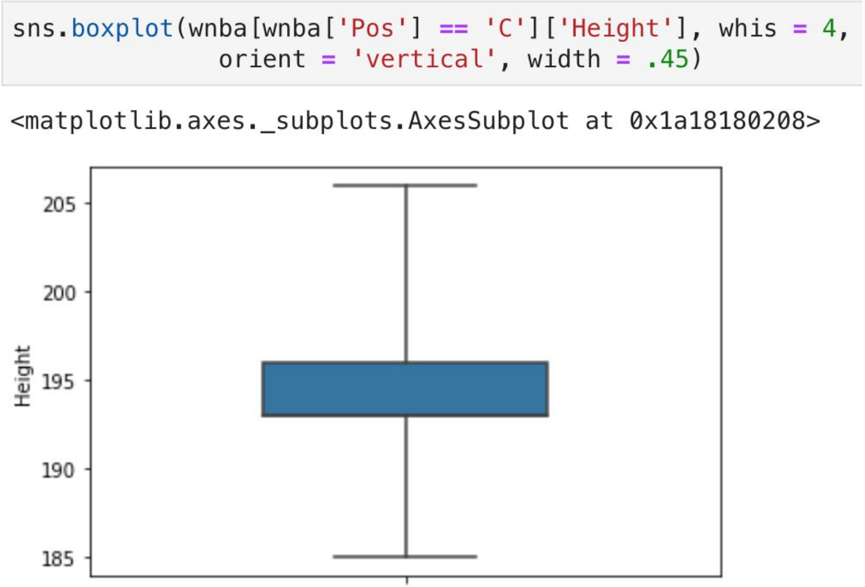

206.5
204.5
202.5
200.5
198.5
196.5
194.5
192.5
190.5
188.5
186.5
184.5
Height
upper bound
200.5
Maximum value in the distribution that is **not an outlier**
Minimum value in the distribution that is **not an outlier**
lower bound
188.5

Outliers



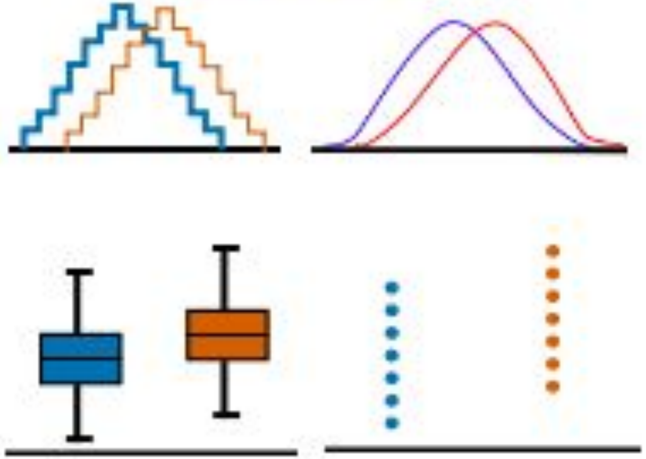
```
sns.boxplot(wnba[wnba['Pos'] == 'C']['Height'], whis = 1.5,
              orient = 'vertical', width = .45)
```

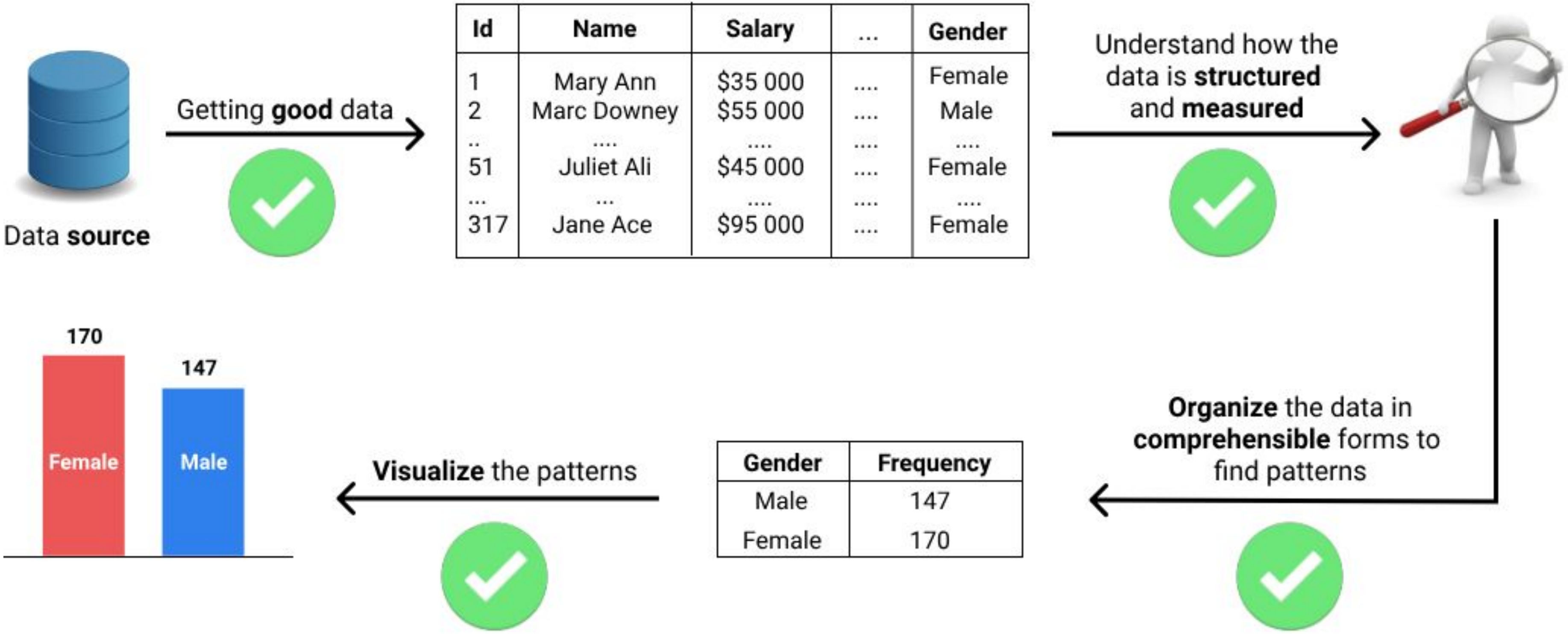
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a180c4518>
```



```
sns.boxplot(wnba[wnba['Pos'] == 'C']['Height'], whis = 4,
              orient = 'vertical', width = .45)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x1a18180208>
```

Scale of measurement	Graphs we can use to compare distributions
Nominal	
Ordinal	
Interval & Ratio	



Data **source**

Getting **good** data

Id	Name	Salary	...	Gender
1	Mary Ann	$35 000	....	Female
2	Marc Downey	$55 000	....	Male
..	....	....	....	....
51	Juliet Ali	$45 000	....	Female
...	...	....	....	....
317	Jane Ace	$95 000	....	Female

Understand how the data is **structured** and **measured**

Organize the data in **comprehensible** forms to find patterns

Gender	Frequency
Male	147
Female	170

Visualize the patterns