Ps2

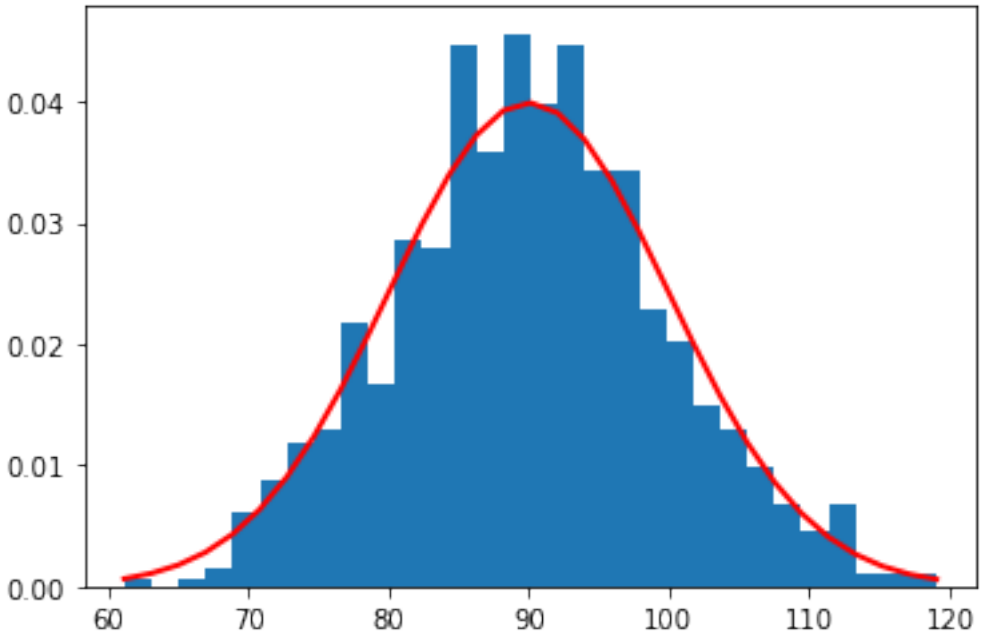
November 1, 2020

[393]:
```
import pandas as pd
import matplotlib.pyplot as plt
import math
import numpy as np
from tabulate import tabulate
import seaborn as sns
```

0.1 Próbki oznaczają wagę mężczyzn

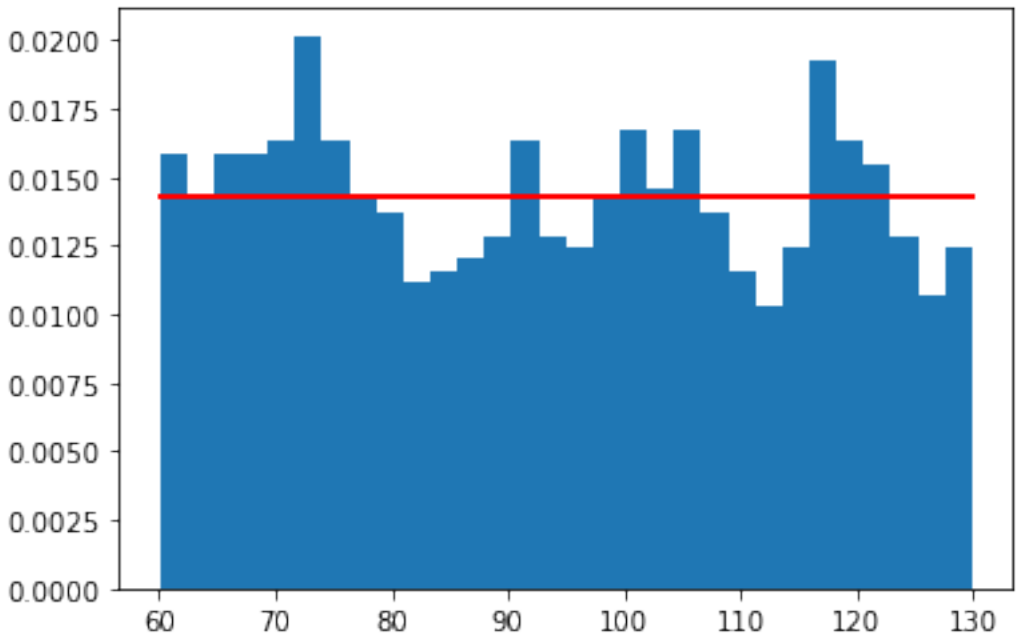
[394]:
```
mu, sigma = 90, 10 # mean and standard deviation
s = np.random.normal(mu, sigma, 1000)
```

[395]:
```
count, bins, ignored = plt.hist(s, 30, density=True)
plt.plot(bins, 1/(sigma * np.sqrt(2 * np.pi)) *
               np.exp( - (bins - mu)**2 / (2 * sigma**2) ),
         linewidth=2, color='r')
plt.show()
```



[396]:
```
s2 = np.random.uniform(60,130,1000)
```

[397]:
```
x = np.arange(31, dtype=float)
count, bins, ignored = plt.hist(s2, 30, density=True)
plt.plot(bins, np.full_like(x, 1/(130-60)), linewidth=2, color='r')
plt.show()
```



[398]:
```
df = pd.DataFrame(s, columns = ["Rozkład normalny"])
df2 = pd.DataFrame(s2, columns= ["Rozkład jednostajny"])
```

[399]:
```
df_both = pd.concat([df,df2], axis=1)
df_working_copy = df_both.copy()
df_both.describe()
```

[399]:

	Rozkład normalny	Rozkład jednostajny
count	1000.000000	1000.000000
mean	90.129857	94.062175
std	9.551618	20.381951
min	61.177262	60.024951
25%	83.861243	75.483133
50%	89.958795	94.166920
75%	96.344635	112.117586
max	119.076596	129.931428

[400]:
```
sample_to_drop = df_working_copy.sample(frac=0.1)
df90 = df_working_copy.drop(sample_to_drop.index)
```

[401]:
```
def print_statistics(data_, label):
    i = 100
    for d in data_:
        i = i-10
        print()
        print('Tabela ' + str(i) + label)
        print(tabulate(d.describe(), headers = ['Nazwa', 'Rozkład normalny',␣
↪'Rozkład jednostajny'], tablefmt = 'fancy_grid'))
```

[402]:
```
df80 = df90.drop(df90.sample(100).index)
df70 = df80.drop(df80.sample(100).index)
df60 = df70.drop(df70.sample(100).index)
df50 = df60.drop(df60.sample(100).index)
df40 = df50.drop(df50.sample(100).index)
df30 = df40.drop(df40.sample(100).index)
df20 = df30.drop(df30.sample(100).index)
df_dropped = [df90, df80, df70, df60, df50, df40, df30, df20]
print_statistics(df_dropped, "%")
```

```
Tabela 90%

 Nazwa       Rozkład normalny     Rozkład jednostajny

 count               900                      900

 mean                 90.149                   94.1443

 std                   9.42822                 20.4599

 min                  61.1773                  60.025

 25%                  83.9154                  75.4831

 50%                  89.968                   94.4012

 75%                  96.3703                 112.465

 max                 119.077                  129.895


Tabela 80%

 Nazwa       Rozkład normalny     Rozkład jednostajny
```

count	800	800
mean	90.2034	94.1179
std	9.49898	20.3534
min	61.1773	60.025
25%	83.8612	75.4831
50%	90.0065	94.3977
75%	96.5275	112.162
max	119.077	129.895

Tabela 70%

Nazwa	Rozkład normalny	Rozkład jednostajny
count	700	700
mean	90.287	94.2151
std	9.51858	20.267
min	61.1773	60.0708
25%	83.8612	75.782
50%	89.9649	94.568
75%	96.6371	112.162
max	119.077	129.895

Tabela 60%

Nazwa	Rozkład normalny	Rozkład jednostajny
count	600	600
mean	90.2423	94.1355
std	9.45922	20.3045

min	61.1773	60.0708
25%	83.8612	75.3149
50%	90.0945	94.3745
75%	96.6322	111.991
max	118.345	129.895

Tabela 50%

Nazwa	Rozkład normalny	Rozkład jednostajny
count	500	500
mean	90.2364	94.05
std	9.34925	20.2475
min	61.1773	60.1403
25%	83.9005	75.8325
50%	90.157	94.568
75%	96.6322	111.673
max	115.858	129.895

Tabela 40%

Nazwa	Rozkład normalny	Rozkład jednostajny
count	400	400
mean	90.0108	93.0797
std	9.36228	19.8405
min	61.1773	60.1403
25%	83.9827	75.3016
50%	89.9379	93.1232

75%	96.1721	109.842
max	115.858	129.895

Tabela 30%

Nazwa	Rozkład normalny	Rozkład jednostajny
count	300	300
mean	89.7493	93.3391
std	9.46076	19.8776
min	61.1773	60.1403
25%	84.0801	75.614
50%	89.7696	93.8413
75%	96.018	110.104
max	115.858	129.895

Tabela 20%

Nazwa	Rozkład normalny	Rozkład jednostajny
count	200	200
mean	90.0214	92.9171
std	9.78956	20.2395
min	66.4787	60.147
25%	83.4279	73.4537
50%	89.7761	92.4303
75%	96.8272	110.104
max	115.858	129.895

[403]:
```
def fill_df_with_mean(data_):
    return data_.append(pd.DataFrame({
                                        'Rozkład normalny': np.
 ↪full(1000-len(data_),data_['Rozkład normalny'].mean()),
                                        'Rozkład jednostajny': np.
 ↪full(1000-len(data_),data_['Rozkład jednostajny'].mean())
                                       })
                        )
```

[404]:
```
df90_mean = fill_df_with_mean(df90)
df80_mean = fill_df_with_mean(df80)
df70_mean = fill_df_with_mean(df70)
df60_mean = fill_df_with_mean(df60)
df50_mean = fill_df_with_mean(df50)
df40_mean = fill_df_with_mean(df40)
df30_mean = fill_df_with_mean(df30)
df20_mean = fill_df_with_mean(df20)
df_mean = [df90_mean, df80_mean, df70_mean, df60_mean, df50_mean, df40_mean,␣
 ↪df30_mean, df20_mean]
print_statistics(df_mean, "% + uzupełnienie średnią")
```

Tabela 90% + uzupełnienie średnią

Nazwa	Rozkład normalny	Rozkład jednostajny
count	1000	1000
mean	90.149	94.1443
std	8.9439	19.4089
min	61.1773	60.025
25%	84.6498	76.9915
50%	90.149	94.1443
75%	95.6111	109.621
max	119.077	129.895

Tabela 80% + uzupełnienie średnią

Nazwa	Rozkład normalny	Rozkład jednostajny
count	1000	1000

mean	90.2034	94.1179
std	8.49508	18.2024
min	61.1773	60.025
25%	85.4171	79.2909
50%	90.2034	94.1179
75%	94.7729	107.338
max	119.077	129.895

Tabela 70% + uzupełnienie średnią

Nazwa	Rozkład normalny	Rozkład jednostajny
count	1000	1000
mean	90.287	94.2151
std	7.9621	16.953
min	61.1773	60.0708
25%	86.2778	83.6834
50%	90.287	94.2151
75%	93.7767	104.326
max	119.077	129.895

Tabela 60% + uzupełnienie średnią

Nazwa	Rozkład normalny	Rozkład jednostajny
count	1000	1000
mean	90.2423	94.1355
std	7.32463	15.7225
min	61.1773	60.0708

25%	88.0621	88.1449
50%	90.2423	94.1355
75%	92.518	100.066
max	118.345	129.895

Tabela 50% + uzupełnienie średnią

Nazwa	Rozkład normalny	Rozkład jednostajny
count	1000	1000
mean	90.2364	94.05
std	6.60761	14.31
min	61.1773	60.1403
25%	90.185	94.05
50%	90.2364	94.05
75%	90.2364	94.5533
max	115.858	129.895

Tabela 40% + uzupełnienie średnią

Nazwa	Rozkład normalny	Rozkład jednostajny
count	1000	1000
mean	90.0108	93.0797
std	5.91678	12.5388
min	61.1773	60.1403
25%	90.0108	93.0797
50%	90.0108	93.0797
75%	90.0108	93.0797

max	115.858	129.895

Tabela 30% + uzupełnienie średnią

Nazwa	Rozkład normalny	Rozkład jednostajny
count	1000	1000
mean	89.7493	93.3391
std	5.17582	10.8747
min	61.1773	60.1403
25%	89.7493	93.3391
50%	89.7493	93.3391
75%	89.7493	93.3391
max	115.858	129.895

Tabela 20% + uzupełnienie średnią

Nazwa	Rozkład normalny	Rozkład jednostajny
count	1000	1000
mean	90.0214	92.9171
std	4.36925	9.03322
min	66.4787	60.147
25%	90.0214	92.9171
50%	90.0214	92.9171
75%	90.0214	92.9171
max	115.858	129.895

[405]:
```
def fill_df_with_median(data_):
    return data_.append(pd.DataFrame({
                                        'Rozkład normalny': np.
 ↪full(1000-len(data_),data_['Rozkład normalny'].median()),
                                        'Rozkład jednostajny': np.
 ↪full(1000-len(data_),data_['Rozkład jednostajny'].median())
                                      })
                        )
```

[406]:
```
df90_median = fill_df_with_median(df90)
df80_median = fill_df_with_median(df80)
df70_median = fill_df_with_median(df70)
df60_median = fill_df_with_median(df60)
df50_median = fill_df_with_median(df50)
df40_median = fill_df_with_median(df40)
df30_median = fill_df_with_median(df30)
df20_median = fill_df_with_median(df20)
df_median = [df90_median, df80_median, df70_median, df60_median, df50_median,␣
 ↪df40_median, df30_median, df20_median]
print_statistics(df_median, "% + uzupełnienie medianą")
```

```
Tabela 90% + uzupełnienie medianą

 Nazwa       Rozkład normalny     Rozkład jednostajny

 count              1000                      1000

 mean               90.1309                    94.17

 std                 8.94407                   19.409

 min                61.1773                    60.025

 25%                84.6498                    76.9915

 50%                89.968                     94.4012

 75%                95.6111                   109.621

 max               119.077                     129.895


Tabela 80% + uzupełnienie medianą

 Nazwa       Rozkład normalny     Rozkład jednostajny

 count              1000                      1000
```

mean	90.164	94.1738
std	8.49545	18.2027
min	61.1773	60.025
25%	85.4171	79.2909
50%	90.0065	94.3977
75%	94.7729	107.338
max	119.077	129.895

Tabela 70% + uzupełnienie medianą

Nazwa	Rozkład normalny	Rozkład jednostajny
count	1000	1000
mean	90.1904	94.321
std	7.96347	16.9538
min	61.1773	60.0708
25%	86.2778	83.6834
50%	89.9649	94.568
75%	93.7767	104.326
max	119.077	129.895

Tabela 60% + uzupełnienie medianą

Nazwa	Rozkład normalny	Rozkład jednostajny
count	1000	1000
mean	90.1832	94.2311
std	7.32499	15.723
min	61.1773	60.0708

25%	88.0621	88.1449
50%	90.0945	94.3745
75%	92.518	100.066
max	118.345	129.895

Tabela 50% + uzupełnienie medianą

Nazwa	Rozkład normalny	Rozkład jednostajny
count	1000	1000
mean	90.1967	94.309
std	6.60773	14.3123
min	61.1773	60.1403
25%	90.143	94.5607
50%	90.157	94.568
75%	90.171	94.5754
max	115.858	129.895

Tabela 40% + uzupełnienie medianą

Nazwa	Rozkład normalny	Rozkład jednostajny
count	1000	1000
mean	89.967	93.1058
std	5.91688	12.5388
min	61.1773	60.1403
25%	89.9379	93.1232
50%	89.9379	93.1232
75%	89.9379	93.1232

max	115.858	129.895

Tabela 30% + uzupełnienie medianą

Nazwa	Rozkład normalny	Rozkład jednostajny
count	1000	1000
mean	89.7635	93.6907
std	5.17583	10.8771
min	61.1773	60.1403
25%	89.7696	93.8413
50%	89.7696	93.8413
75%	89.7696	93.8413
max	115.858	129.895

Tabela 20% + uzupełnienie medianą

Nazwa	Rozkład normalny	Rozkład jednostajny
count	1000	1000
mean	89.8251	92.5276
std	4.37035	9.03532
min	66.4787	60.147
25%	89.7761	92.4303
50%	89.7761	92.4303
75%	89.7761	92.4303
max	115.858	129.895

[407]:
```
def fill_df_with_random_numbers(data_):
    return data_.append(pd.DataFrame({
                                        'Rozkład normalny': np.random.normal(mu,␣
↪sigma, 1000-len(data_)),
                                        'Rozkład jednostajny': np.random.
↪uniform(data_['Rozkład jednostajny'].min(),data_['Rozkład jednostajny'].
↪max(),1000-len(data_))
                                    })
                        )
```

[408]:
```
df90_random = fill_df_with_random_numbers(df90)
df80_random = fill_df_with_random_numbers(df80)
df70_random = fill_df_with_random_numbers(df70)
df60_random = fill_df_with_random_numbers(df60)
df50_random = fill_df_with_random_numbers(df50)
df40_random = fill_df_with_random_numbers(df40)
df30_random = fill_df_with_random_numbers(df30)
df20_random = fill_df_with_random_numbers(df20)
df_random = [df90_random, df80_random, df70_random, df60_random, df50_random,␣
↪df40_random, df30_random, df20_random]
print_statistics(df_random, "% + uzupełnienie losowymi wartościami")
```

Tabela 90% + uzupełnienie losowymi wartościami

Nazwa	Rozkład normalny	Rozkład jednostajny
count	1000	1000
mean	90.144	94.2502
std	9.43295	20.415
min	61.1773	60.025
25%	83.8612	75.782
50%	89.9588	94.5104
75%	96.3703	112.568
max	119.077	129.895

Tabela 80% + uzupełnienie losowymi wartościami

Nazwa	Rozkład normalny	Rozkład jednostajny

count	1000	1000
mean	89.8279	94.6734
std	9.56964	20.3044
min	60.0623	60.025
25%	83.3336	76.4037
50%	89.8035	94.5104
75%	96.0135	112.987
max	119.077	129.895

Tabela 70% + uzupełnienie losowymi wartościami

Nazwa	Rozkład normalny	Rozkład jednostajny
count	1000	1000
mean	89.8521	94.1369
std	9.64192	20.0285
min	54.6641	60.0708
25%	83.4674	76.3716
50%	89.7696	94.3463
75%	96.302	111.717
max	119.077	129.895

Tabela 60% + uzupełnienie losowymi wartościami

Nazwa	Rozkład normalny	Rozkład jednostajny
count	1000	1000
mean	90.1514	94.2461
std	9.82329	20.0651

min	61.1773	60.0708
25%	83.38	76.0811
50%	89.988	94.3745
75%	96.9272	111.673
max	118.345	129.895

Tabela 50% + uzupełnienie losowymi wartościami

Nazwa	Rozkład normalny	Rozkład jednostajny
count	1000	1000
mean	90.5105	94.5575
std	9.54393	20.1117
min	61.1773	60.1403
25%	84.2703	76.6918
50%	90.7994	95.1509
75%	96.8764	111.78
max	115.858	129.895

Tabela 40% + uzupełnienie losowymi wartościami

Nazwa	Rozkład normalny	Rozkład jednostajny
count	1000	1000
mean	89.972	94.5548
std	9.87245	19.8432
min	51.5431	60.1403
25%	83.3113	77.3566
50%	89.7037	94.5919

75%	96.8475	112.157
max	121.239	129.895

Tabela 30% + uzupełnienie losowymi wartościami

Nazwa	Rozkład normalny	Rozkład jednostajny
count	1000	1000
mean	90.1489	94.0726
std	9.72587	20.0437
min	61.0138	60.1403
25%	83.8517	76.1032
50%	90.677	93.8025
75%	96.6004	112.128
max	120.156	129.895

Tabela 20% + uzupełnienie losowymi wartościami

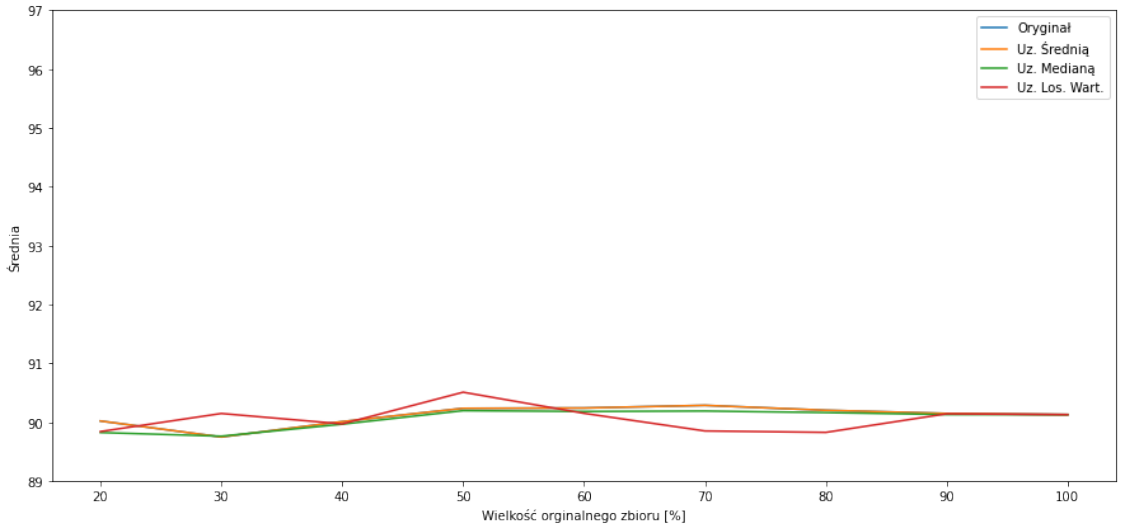
Nazwa	Rozkład normalny	Rozkład jednostajny
count	1000	1000
mean	89.8422	94.9707
std	9.88008	19.9937
min	57.224	60.147
25%	83.8334	76.8567
50%	90.0865	95.6692
75%	96.3935	112.356
max	122.969	129.895

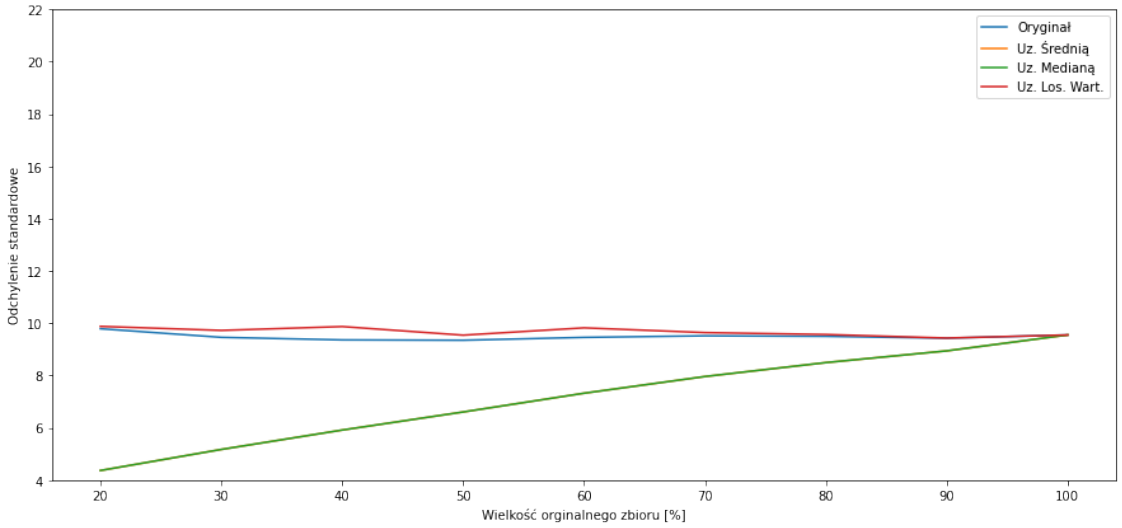
[409]:
```
def get_mean_std_normal(data_):
    data_mean = [d['Rozkład normalny'].mean() for d in data_]
    data_mean.reverse()
    data_mean.append(df_both['Rozkład normalny'].mean())
    data_std = [d['Rozkład normalny'].std() for d in data_]
    data_std.reverse()
    data_std.append(df_both['Rozkład normalny'].std())
    return data_mean, data_std
```

[410]:
```
def get_mean_std_uniform(data_):
    data_mean = [d['Rozkład jednostajny'].mean() for d in data_]
    data_mean.reverse()
    data_mean.append(df_both['Rozkład jednostajny'].mean())
    data_std = [d['Rozkład jednostajny'].std() for d in data_]
    data_std.reverse()
    data_std.append(df_both['Rozkład jednostajny'].std())
    return data_mean, data_std
```

[411]:
```
x = ['20', '30', '40','50','60','70','80','90','100']
df_dropped_mean, df_dropped_std = get_mean_std_normal(df_dropped)
df_mean_mean, df_mean_std = get_mean_std_normal(df_mean)
df_median_mean, df_median_std = get_mean_std_normal(df_median)
df_random_mean, df_random_std = get_mean_std_normal(df_random)

f = plt.figure(figsize=(15,15))
ax = f.add_subplot(211)
ax2 = f.add_subplot(212)

ax.plot(x, df_dropped_mean, label = "Oryginał")
ax.plot(x, df_mean_mean, label = "Uz. Średnią")
ax.plot(x, df_median_mean, label = "Uz. Medianą")
ax.plot(x, df_random_mean, label = "Uz. Los. Wart.")
ax.set_xlabel("Wielkość orginalnego zbioru [%]")
ax.set_ylabel("Średnia")
ax.set_ylim([89,97])
ax.legend()

ax2.plot(x, df_dropped_std, label = "Oryginał")
ax2.plot(x, df_mean_std, label = "Uz. Średnią")
ax2.plot(x, df_median_std, label = "Uz. Medianą")
ax2.plot(x, df_random_std, label = "Uz. Los. Wart.")
ax2.set_xlabel("Wielkość orginalnego zbioru [%]")
ax2.set_ylabel("Odchylenie standardowe")
ax2.set_ylim([4,22])
ax2.legend()
```

[411]:
```
<matplotlib.legend.Legend at 0x2de5c27ffa0>
```

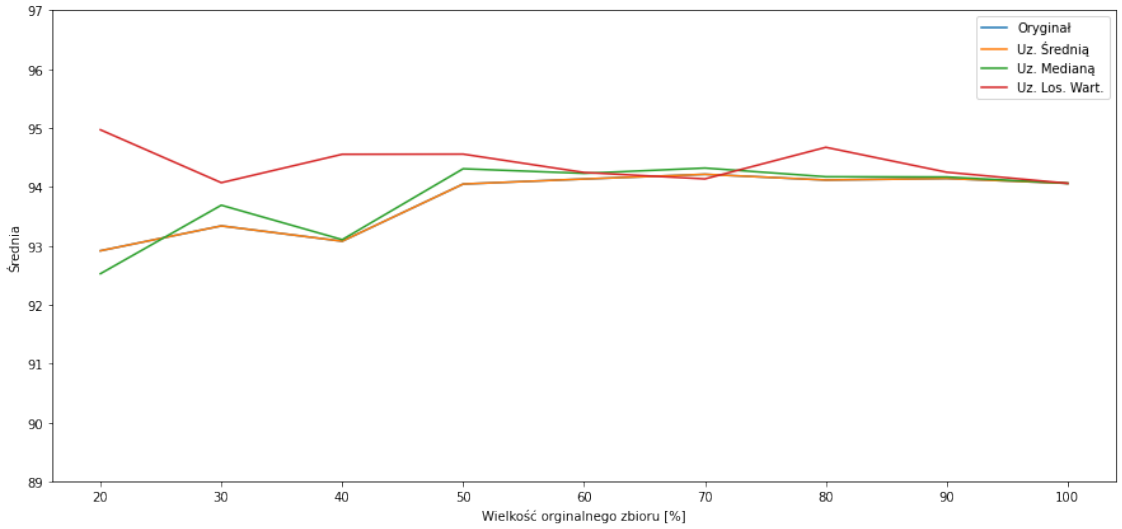


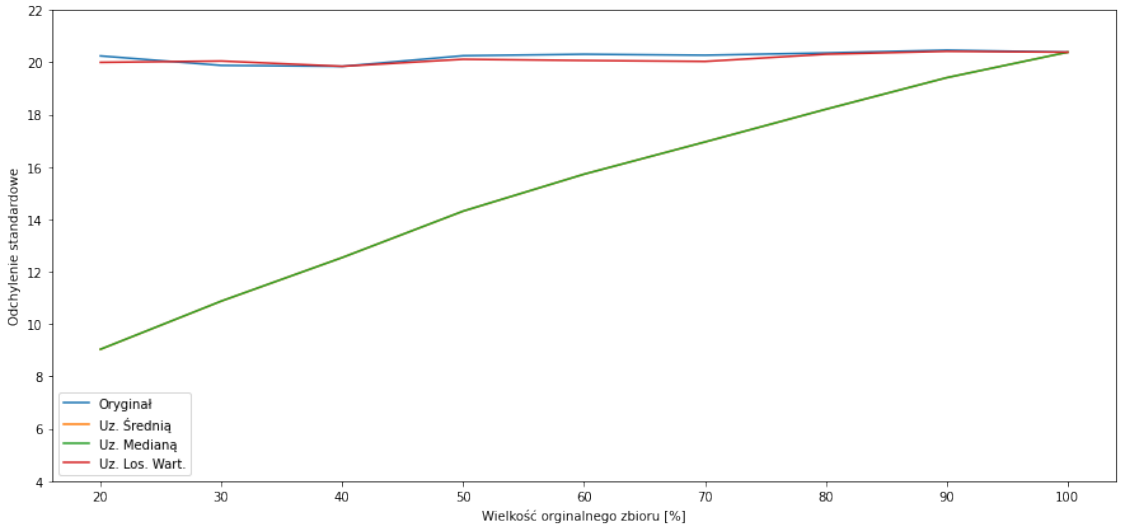


[412]:
```
x = ['20', '30', '40','50','60','70','80','90','100']
df_dropped_mean, df_dropped_std = get_mean_std_uniform(df_dropped)
df_mean_mean, df_mean_std = get_mean_std_uniform(df_mean)
df_median_mean, df_median_std = get_mean_std_uniform(df_median)
df_random_mean, df_random_std = get_mean_std_uniform(df_random)

f = plt.figure(figsize=(15,15))
ax = f.add_subplot(211)
ax2 = f.add_subplot(212)

ax.plot(x, df_dropped_mean, label = "Oryginał")
ax.plot(x, df_mean_mean, label = "Uz. Średnią")
ax.plot(x, df_median_mean, label = "Uz. Medianą")
```

```
ax.plot(x, df_random_mean, label = "Uz. Los. Wart.")
ax.set_xlabel("Wielkość orginalnego zbioru [%]")
ax.set_ylabel("Średnia")
ax.set_ylim([89,97])
ax.legend()

ax2.plot(x, df_dropped_std, label = "Oryginał")
ax2.plot(x, df_mean_std, label = "Uz. Średnią")
ax2.plot(x, df_median_std, label = "Uz. Medianą")
ax2.plot(x, df_random_std, label = "Uz. Los. Wart.")
ax2.set_xlabel("Wielkość orginalnego zbioru [%]")
ax2.set_ylabel("Odchylenie standardowe")
ax2.set_ylim([4,22])
ax2.legend()
```

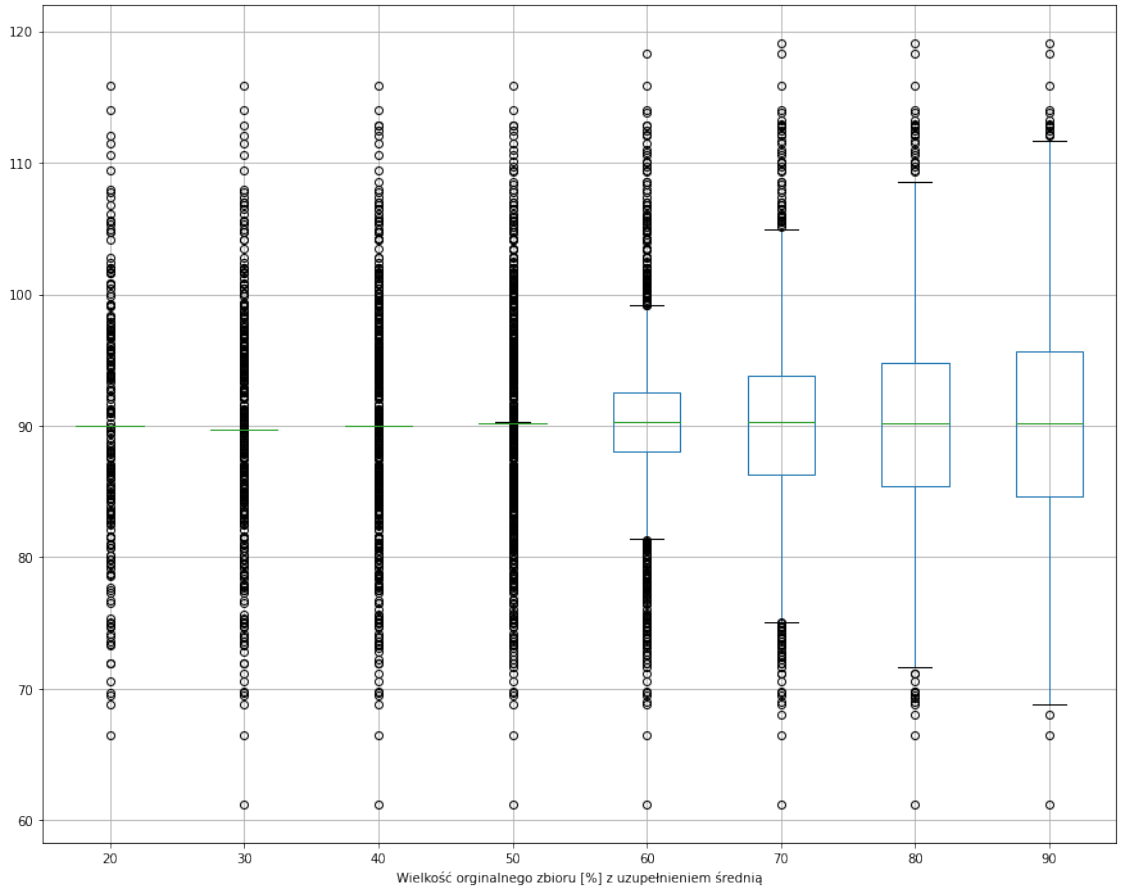
[412]: <matplotlib.legend.Legend at 0x2de5c36c850>





[413]:
```
df_mean_normal = [df_['Rozkład normalny'].copy() for df_ in df_mean]
for df_ in df_mean_normal:
    df_.index = range(1, len(df_) + 1)

df_mean_normal.reverse()
df_mean_normal_merged = pd.concat(df_mean_normal, axis = 1)
df_mean_normal_merged.columns = ['20', '30', '40','50','60','70','80','90']
ax = df_mean_normal_merged.boxplot(column = ['20', '30',␣
 ↪'40','50','60','70','80','90'], figsize = [15,12])
ax.set_xlabel("Wielkość orginalnego zbioru [%] z uzupełnieniem średnią")
```

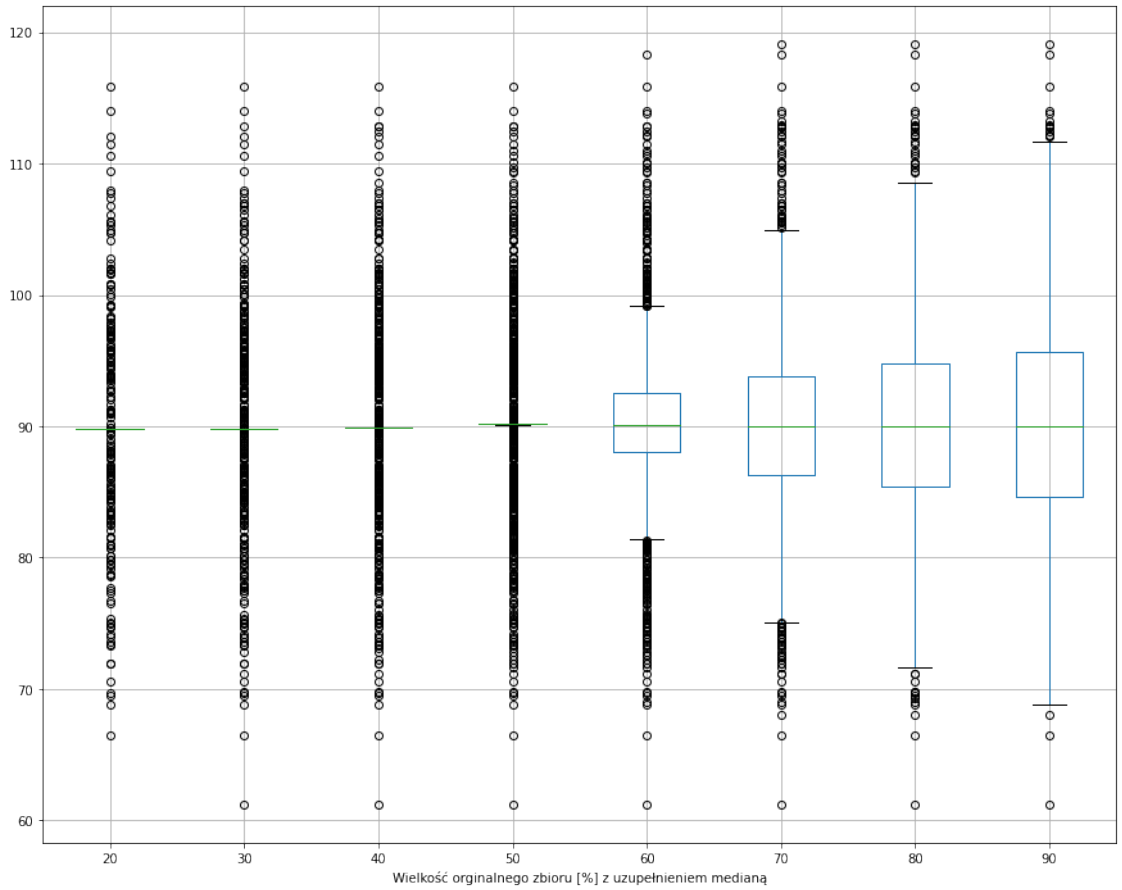
[413]:
```
Text(0.5, 0, 'Wielkość orginalnego zbioru [%] z uzupełnieniem średnią')
```



[414]:
```
df_median_normal = [df_['Rozkład normalny'].copy() for df_ in df_median]
for df_ in df_median_normal:
    df_.index = range(1, len(df_) + 1)
```

```
df_median_normal.reverse()
df_median_normal_merged = pd.concat(df_median_normal, axis = 1)
df_median_normal_merged.columns = ['20', '30', '40','50','60','70','80','90']
ax = df_median_normal_merged.boxplot(column = ['20', '30',␣
↪'40','50','60','70','80','90'], figsize = [15,12])
ax.set_xlabel("Wielkość orginalnego zbioru [%] z uzupełnieniem medianą")
```

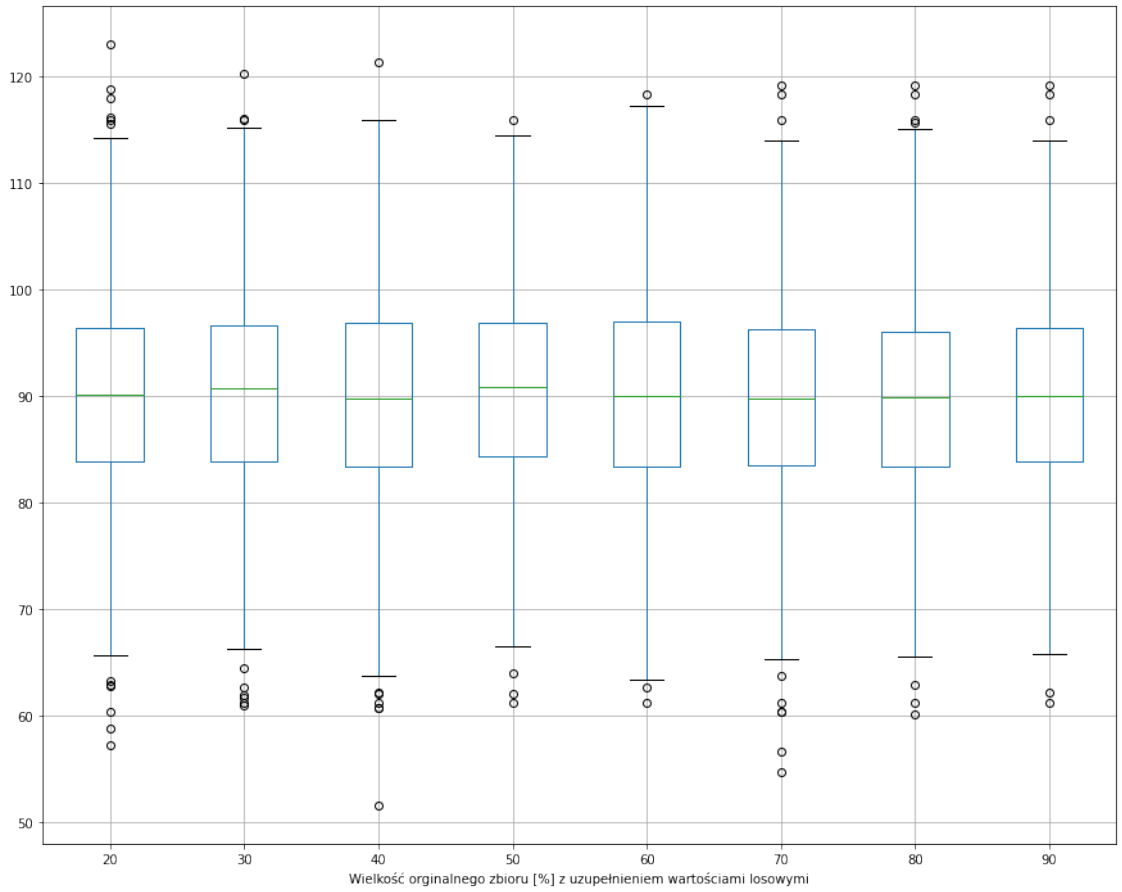
[414]: Text(0.5, 0, 'Wielkość orginalnego zbioru [%] z uzupełnieniem medianą')



[415]:
```
df_random_normal = [df_['Rozkład normalny'].copy() for df_ in df_random]
for df_ in df_random_normal:
    df_.index = range(1, len(df_) + 1)

df_random_normal.reverse()
df_random_normal_merged = pd.concat(df_random_normal, axis = 1)
df_random_normal_merged.columns = ['20', '30', '40','50','60','70','80','90']
ax = df_random_normal_merged.boxplot(column = ['20', '30',␣
↪'40','50','60','70','80','90'], figsize = [15,12])
```

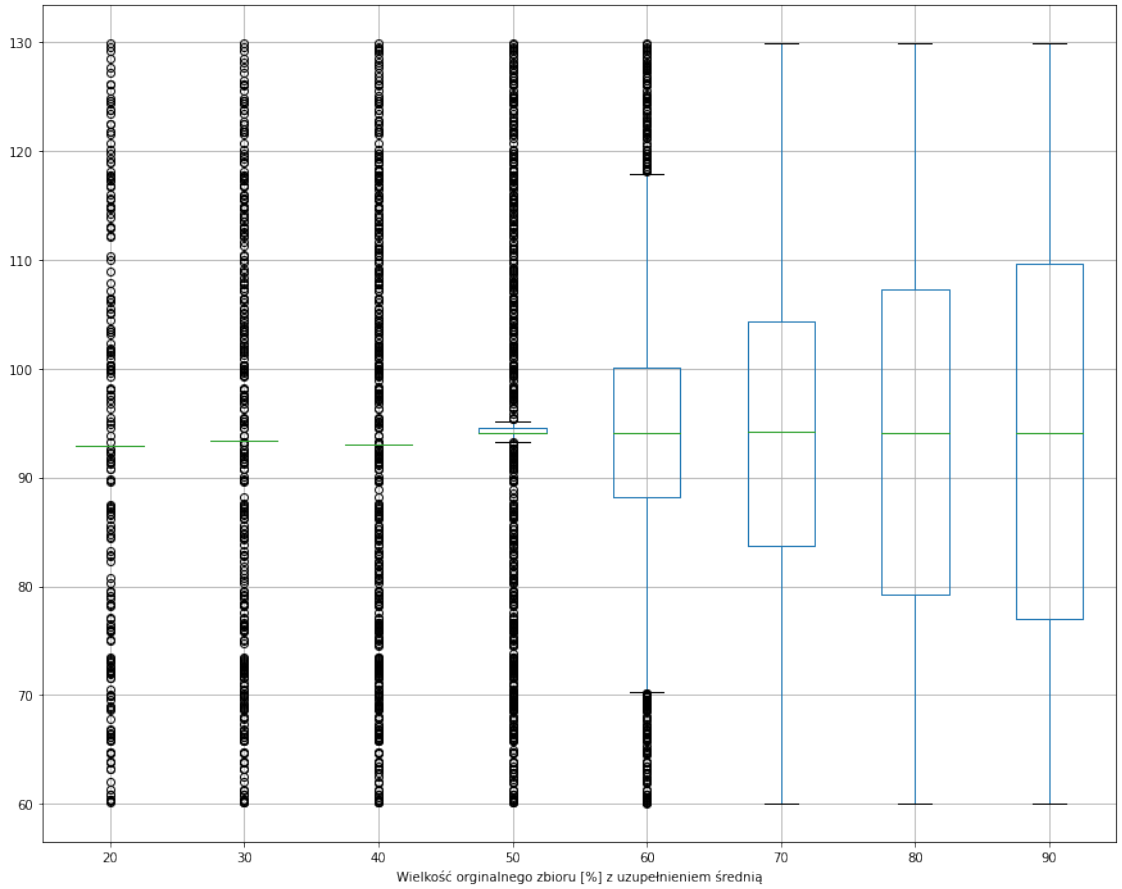
```
ax.set_xlabel("Wielkość orginalnego zbioru [%] z uzupełnieniem wartościami␣
↪losowymi")
```

[415]:
```
Text(0.5, 0, 'Wielkość orginalnego zbioru [%] z uzupełnieniem wartościami
losowymi')
```



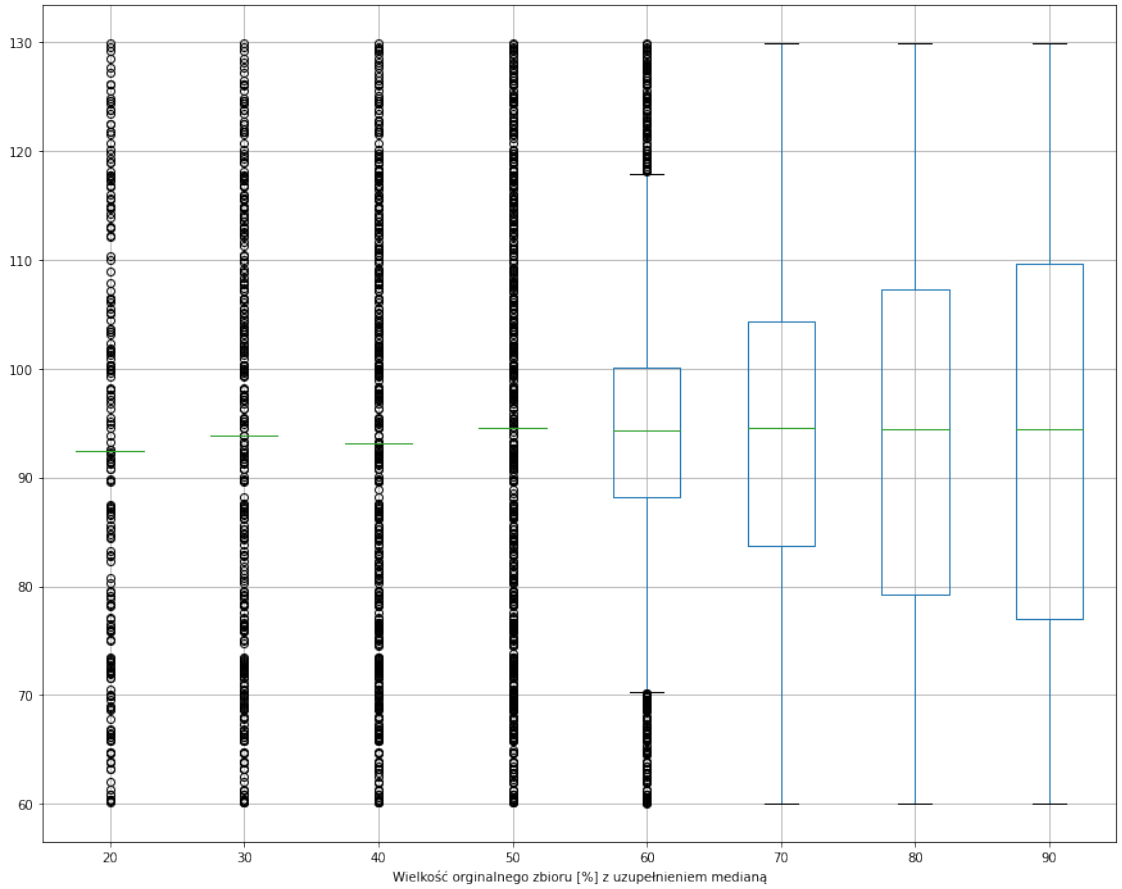
[416]:
```
df_mean_uniform = [df_['Rozkład jednostajny'].copy() for df_ in df_mean]
for df_ in df_mean_uniform:
    df_.index = range(1, len(df_) + 1)

df_mean_uniform.reverse()
df_mean_uniform_merged = pd.concat(df_mean_uniform, axis = 1)
df_mean_uniform_merged.columns = ['20', '30', '40','50','60','70','80','90']
ax = df_mean_uniform_merged.boxplot(column = ['20', '30',␣
↪'40','50','60','70','80','90'], figsize = [15,12])
ax.set_xlabel("Wielkość orginalnego zbioru [%] z uzupełnieniem średnią")
```

[416]:
```
Text(0.5, 0, 'Wielkość orginalnego zbioru [%] z uzupełnieniem średnią')
```



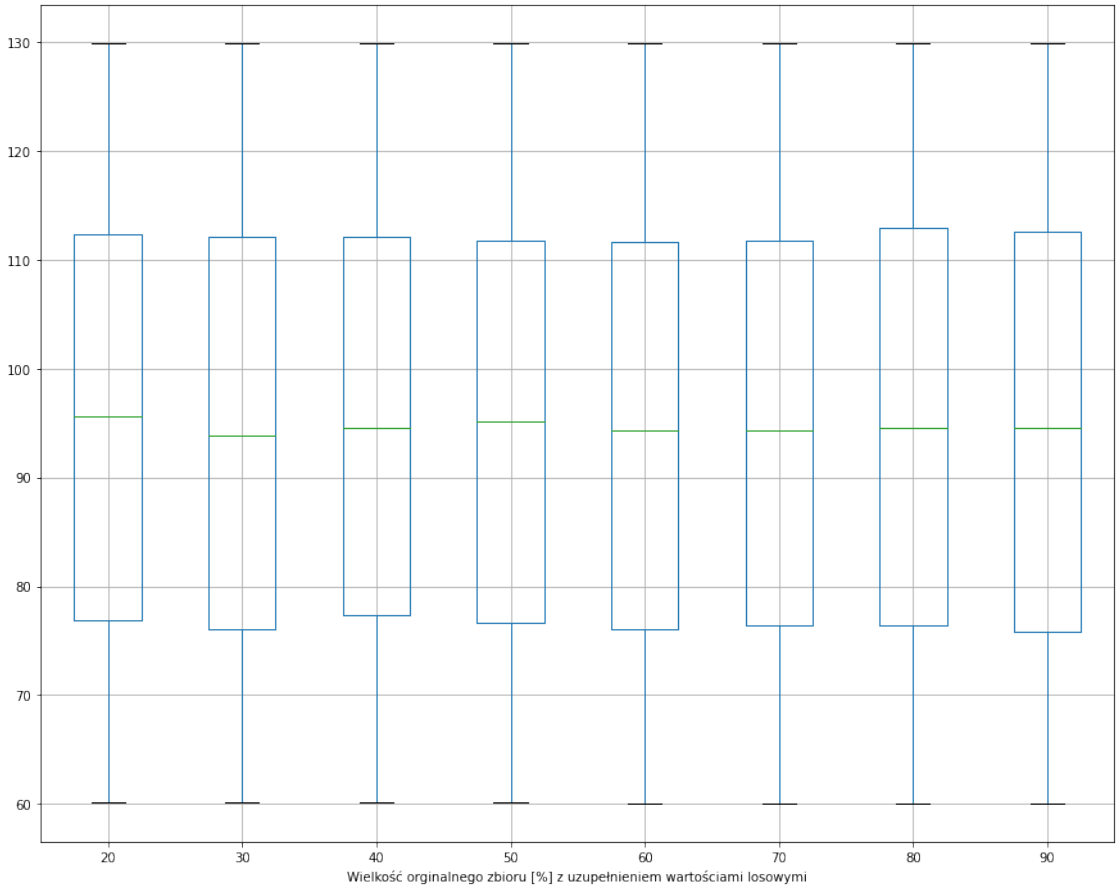
[417]:
```
df_median_uniform = [df_['Rozkład jednostajny'].copy() for df_ in df_median]
for df_ in df_median_uniform:
    df_.index = range(1, len(df_) + 1)

df_median_uniform.reverse()
df_median_uniform_merged = pd.concat(df_median_uniform, axis = 1)
df_median_uniform_merged.columns = ['20', '30', '40','50','60','70','80','90']
ax = df_median_uniform_merged.boxplot(column = ['20', '30',␣
 ↪'40','50','60','70','80','90'], figsize = [15,12])
ax.set_xlabel("Wielkość orginalnego zbioru [%] z uzupełnieniem medianą")
```

[417]:
```
Text(0.5, 0, 'Wielkość orginalnego zbioru [%] z uzupełnieniem medianą')
```



[418]:
```
df_random_uniform = [df_['Rozkład jednostajny'].copy() for df_ in df_random]
for df_ in df_random_uniform:
    df_.index = range(1, len(df_) + 1)

df_random_uniform.reverse()
df_random_uniform_merged = pd.concat(df_random_uniform, axis = 1)
df_random_uniform_merged.columns = ['20', '30', '40','50','60','70','80','90']
ax = df_random_uniform_merged.boxplot(column = ['20', '30',␣
 ↪'40','50','60','70','80','90'], figsize = [15,12])
ax.set_xlabel("Wielkość orginalnego zbioru [%] z uzupełnieniem wartościami␣
 ↪losowymi")
```

[418]:
```
Text(0.5, 0, 'Wielkość orginalnego zbioru [%] z uzupełnieniem wartościami
losowymi')
```


130
120
110
100
90
80
70
60
20
30
40
50
60
70
80
90
Wielkość orginalnego zbioru [%] z uzupełnieniem wartościami losowymi