NAME = "Xudong Guo" Purpose: This visualization is used to answer question 3: Do frequency of use, number of strokes, and grade levels have some relationships with the Wanikani levels?

Input: cleaned_link.csv

In [4]:
```
import pandas as pd
import ast
import math
import matplotlib.pylab as plt
import numpy as np
```

In [5]:
```
#word_table = pd.read_csv('wikiword_table.csv')
kanji_table = pd.read_csv("../Question1/cleaned_link.csv")
```

In [6]:
```
#word_table
kanji_table
```

Out[6]:

	kanji	strokes	frequency	grade	jlpt	parts	radicals	on_readings	kun_readings	on_readings_compou
0	亜	7.0	1509.0	junior high	N1	['一', '｜', '口']	{'二': 'two'}	['ア']	['つ.ぐ']	['亜 【ア】 sub-, (indicating a low oxid
1	哀	9.0	1715.0	junior high	N1	['亠', '口', '衣']	{'口': 'mouth, opening'}	['アイ']	['あわ.れ', 'あわ.れむ', 'かな.しい']	['哀悼 【アイト condolence, regret, trib
2	挨	10.0	2258.0	junior high	NaN	['厶', '扎', '矢', '乞']	{'手 (扌, 龵)': 'hand'}	['アイ']	['ひら.く']	['挨拶 【アイサ greeting, greeti salutatio
3	愛	13.0	640.0	grade 4	N3	['冖', '夂', '心', '爪']	{'心 (忄, 㣺)': 'heart'}	['アイ']	['いと.しい', 'かな.しい', 'め.でる', 'お.しむ', 'まな']	['愛 【アイ】 affection, care, attachn
4	曖	17.0	NaN	junior high	NaN	['冖', '夂', '心', '日', '爪']	{'日': 'sun, day'}	['アイ']	['くら.い']	['曖昧 【アイマイ】 va ambiguous, unc fuz
...	...	...	...	...	...	...	...	...	...	
2131	脇	10.0	1806.0	junior high	NaN	['力', '月']	{'肉 (月)': 'meat'}	['キョウ']	['わき', 'わけ']	['脇侍 【ワキジ】 flar image (e.g. in a Buddh
2132	惑	12.0	777.0	junior high	N1	['口', '心', '戈']	{'心 (忄, 㣺)': 'heart'}	['ワク']	['まど.う']	['惑星 【ワクセ planet', '惑星科学 【ワ イカガク】 pla
2133	枠	8.0	922.0	junior high	N1	['十', '九', '木']	{'木': 'tree'}	NaN	['わく']	['枠 【わく】 fra framework, border, li
2134	湾	12.0	545.0	junior high	N2	['亠', '弓', '汁']	{'水 (氵, 氺)': 'water'}	['ワン']	['いりえ']	['湾 【ワン】 bay, inlet', '湾岸 【ワンガ
2135	腕	12.0	1163.0	junior high	N2	['卩', '夕', '宀', '月']	{'肉 (月)': 'meat'}	['ワン']	['うで']	['腕力 【ワンリョ physical strength, b stre

2136 rows × 26 columns

In [7]:
```
row = kanji_table['radicals'][0]
list(ast.literal_eval(row).keys())
```

Out[7]: ['二']

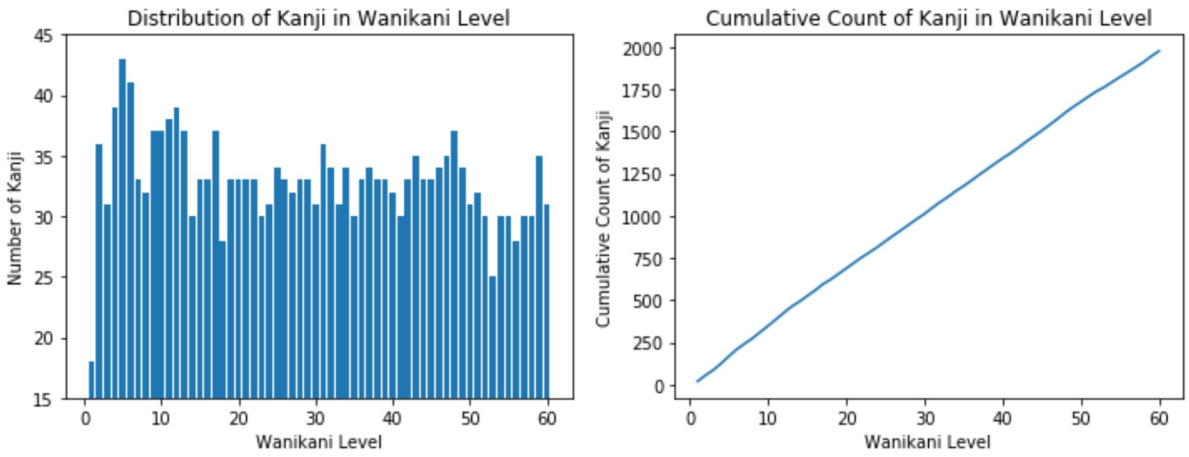
In [8]:
```
# Wanikani Level Viz Get Data
kanjis = kanji_table['kanji']
levels = kanji_table['wanikani_level']
levels = levels.unique()
count = {}
for i in range(0,len(kanjis)):
    l = kanji_table.iloc[i]['wanikani_level']
    if math.isnan(l):
        continue
    if l in count:
        count[l] = count[l] + 1
    else:
        count[l] = 1
```

In [ ]:
```
```

Number of Kanji in Each Wanikani Level

In [9]:
```
lists = sorted(count.items())
x, y = zip(*lists)

fig = plt.figure(figsize = (12,4))
ax1 = fig.add_subplot(1,2,1)
ax1.set_title("Distribution of Kanji in Wanikani Level")
ax1.set_xlabel("Wanikani Level")
ax1.set_ylabel("Number of Kanji")
ax2 = fig.add_subplot(1,2,2)
ax2.set_title("Cumulative Count of Kanji in Wanikani Level")
ax2.set_xlabel("Wanikani Level")
ax2.set_ylabel("Cumulative Count of Kanji")
ax1.bar(x,y)
ax1.set_ylim(15, 45)
ax2.plot(x,np.array(y).cumsum())
```

Out[9]:
```
[<matplotlib.lines.Line2D at 0xcecc5f0>]
```

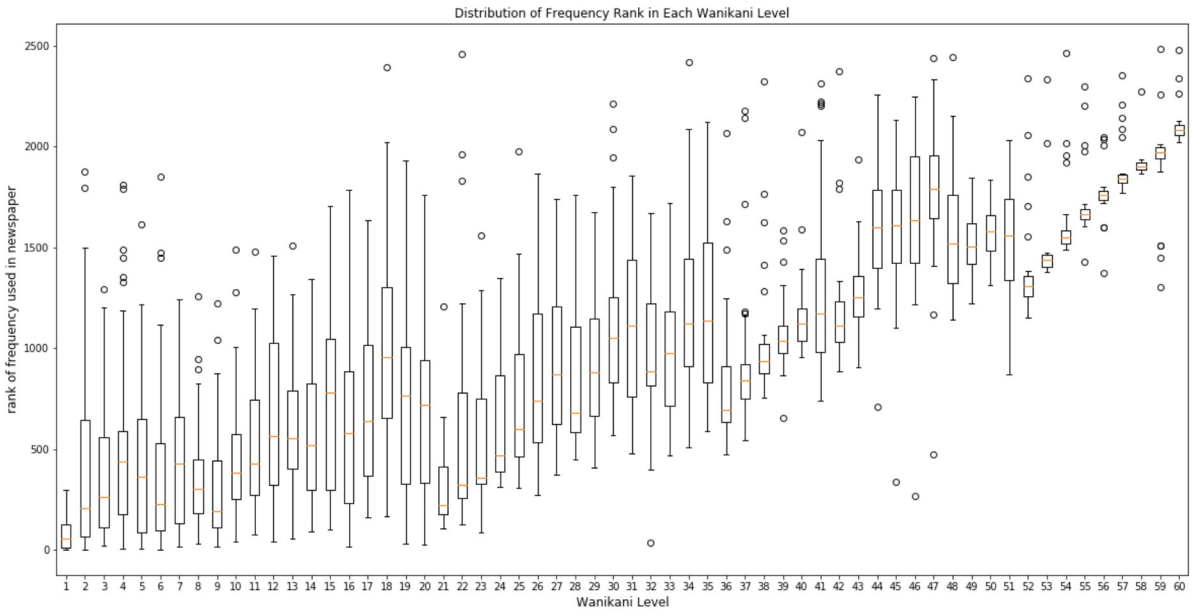


Conclusion

Wanikani distributed kanjis in a quite uniform way. Each level contains approximately 45 kanjis. We could see in the cumulative curve, the 'pace' is quite smooth.

In [102]:
```
container2 = []

grade_char_count = []

for gradenum in np.arange(len(x)):

    raw_in_gradenum = kanji_table[kanji_table["wanikani_level"] == x[gradenum]]

    grade_char_count.append(raw_in_gradenum["frequency"].count())

    #raw_grade_freq_avr = np.mean(raw_in_gradenum["frequency"])

    #container2[gradenum] = raw_grade_freq_avr
    container2.append(list(raw_in_gradenum['frequency'].dropna()))
```

In [103]:
```
fig2 = plt.figure(figsize=(20,10))
axis21 = fig2.add_subplot(1,1,1)
#axis21.bar(np.arange(len(grades)),container2, alpha=0.5, color="green")
axis21.boxplot(container2)
#axis21.plot(np.array(grade_char_count).cumsum(),color="green")
axis21.set_title("Distribution of Frequency Rank in Each Wanikani Level", fontsiz
e="large")
axis21.set_xlabel("Wanikani Level", fontsize="large")
axis21.set_ylabel("rank of frequency used in newspaper", fontsize="large")

xticks21 = []
for i in np.arange(len(x)):
    xticks21.append("{}".format(int(x[i])))

#axis21.set_xticks(np.arange(len(x)))
axis21.set_xticklabels(xticks21)
print("")
```

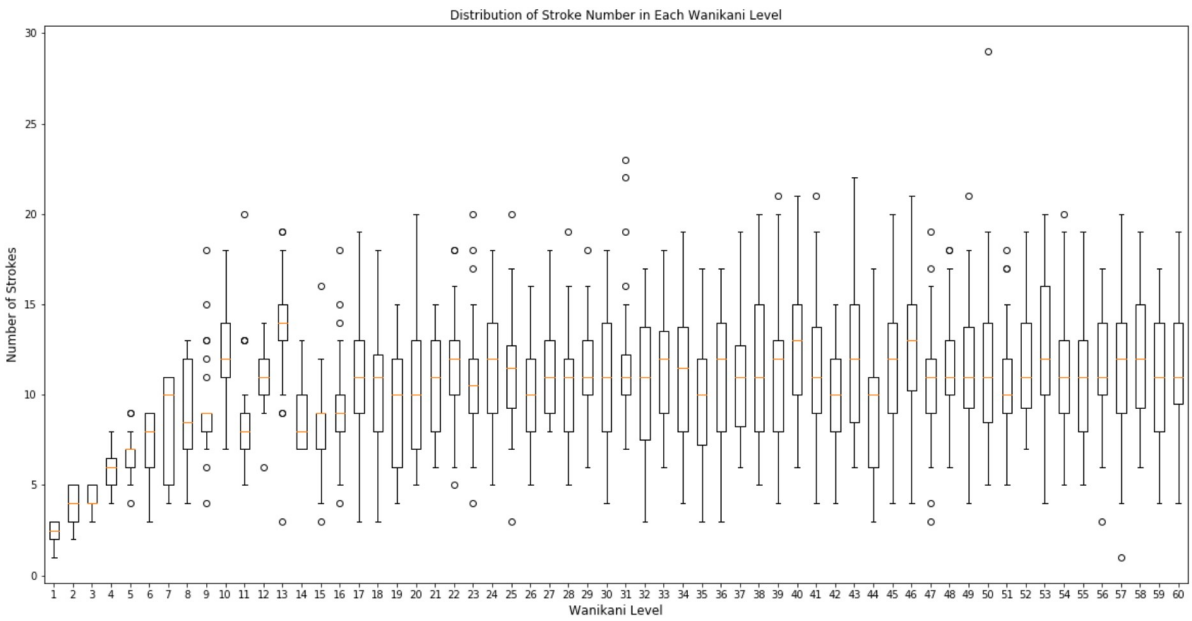


Conclusion

Wanikani tends to introduce the most frequently used kanjis first. The box plot shows Q1 and Q3 boxes are at the bottom (meaning these kanjis are more frequently used) for lower level kanjis. As the kanjis' level goes higher, their frequency of use is becoming less.

In [104]:
```
container3 = []

stroke_char_count = []

for gradenum in np.arange(len(x)):

    raw_in_gradenum = kanji_table[kanji_table["wanikani_level"] == x[gradenum]]

    stroke_char_count.append(raw_in_gradenum["strokes"].count())

    container3.append(list(raw_in_gradenum['strokes'].dropna()))
```

In [105]:
```
fig3 = plt.figure(figsize=(20,10))
axis3 = fig3.add_subplot(1,1,1)
#axis21.bar(np.arange(len(grades)),container2, alpha=0.5, color="green")
axis3.boxplot(container3)
#axis21.plot(np.array(grade_char_count).cumsum(),color="green")
axis3.set_title("Distribution of Stroke Number in Each Wanikani Level", fontsize="
large")
axis3.set_xlabel("Wanikani Level", fontsize="large")
axis3.set_ylabel("Number of Strokes", fontsize="large")

xticks3 = []
for i in np.arange(len(x)):
    xticks3.append("{}".format(int(x[i])))

#axis21.set_xticks(np.arange(len(x)))
axis3.set_xticklabels(xticks3)
print("")
```



Conclusion

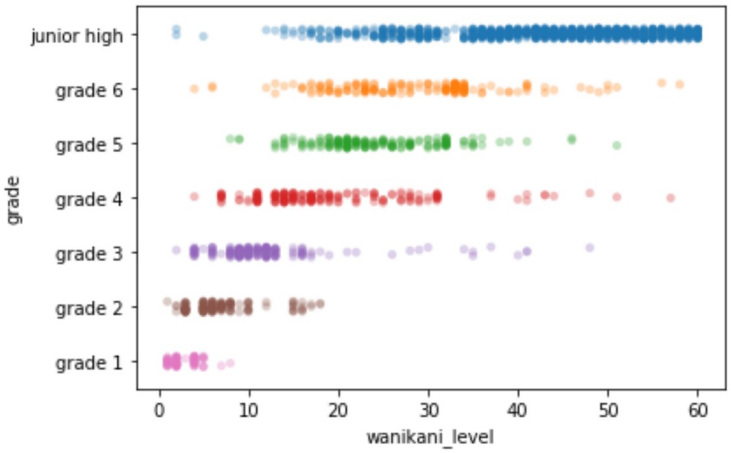
Although in the first 10-ish levels, the number of strokes of those kanjis are less, it doesn't have much difference for the latter levels. So, after level 10, the number of strokes may not be a categorizing standard.

In [108]:
```
level = kanji_table['wanikani_level']
grade = kanji_table['grade']
newtable = pd.DataFrame([level,grade])
fig4 = plt.figure()
newtable = newtable.transpose()
newtable = newtable.dropna()
#axis4 = fig4.add_subplot(1,1,1)
newtable = newtable.sort_values(by = ['grade'], ascending = False)
#axis4.scatter(newtable['wanikani_level'], newtable['grade'])
#axis4.set_yticks(['grade 1', 'grade 2', 'grade 3', 'grade 4', 'grade 5', 'grade 6
', 'junior high'])
```

```
<Figure size 432x288 with 0 Axes>
```

In [107]:
```
import seaborn as sns
sns.stripplot(x='wanikani_level', y='grade', data=newtable, jitter=True, edgecolor
= 'none', alpha = 0.3)
```

Out[107]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1a24d5d898>
```



Conclusion

This plot shows WaniKani tends to use the similar sets as the categorization made by Japanese Ministry of Education. Lower WaniKani level kanjis are taught in grade 1 or 2, while the higher level ones mostly taught in junior high. They have somewhat a correlation.

In [ ]: