1. The most Nobel of Prizes

The Nobel Prize is perhaps the world's most well known scientific award. Except for the honor, prestige and substantial prize money the recipient also gets a gold medal showing Alfred Nobel (1833 - 1896) who established the prize. Every year it's given to scientists and scholars in the categories chemistry, literature, physics, physiology or medicine, economics, and peace. The first Nobel Prize was handed out in 1901, and at that time the Prize was very Eurocentric and male-focused, but nowadays it's not biased in any way whatsoever. Surely. Right?

Well, we're going to find out! The Nobel Foundation has made a dataset available of all prize winners from the start of the prize, in 1901, to 2016. Let's load it in and take a look.

In [145]:
```
# Loading in required libraries
import pandas as pd
import seaborn as sns
import numpy as np

# Reading in the Nobel Prize data
nobel = pd.read_csv('datasets/nobel.csv')

# Taking a look at the first several winners
nobel.head(6)
```

Out[145]:

	year	category	prize	motivation	prize_share	laureate_id	laureate_type	full_name	birth_date	birth_
0	1901	Chemistry	The Nobel Prize in Chemistry 1901	"in recognition of the extraordinary services ...	1/1	160	Individual	Jacobus Henricus van 't Hoff	1852-08-30	Rotterda
1	1901	Literature	The Nobel Prize in Literature 1901	"in special recognition of his poetic composit...	1/1	569	Individual	Sully Prudhomme	1839-03-16	Paris
2	1901	Medicine	The Nobel Prize in Physiology or Medicine 1901	"for his work on serum therapy, especially its...	1/1	293	Individual	Emil Adolf von Behring	1854-03-15	Hansdor (Lawice)
3	1901	Peace	The Nobel Peace Prize 1901	NaN	1/2	462	Individual	Jean Henry Dunant	1828-05-08	Geneva
4	1901	Peace	The Nobel Peace Prize 1901	NaN	1/2	463	Individual	Frédéric Passy	1822-05-20	Paris
5	1901	Physics	The Nobel Prize in Physics 1901	"in recognition of the extraordinary services ...	1/1	1	Individual	Wilhelm Conrad Röntgen	1845-03-27	Lennep (Remsch

In [146]:
```
%%nose

last_value = _

def test_pandas_loaded():
    assert pd.__name__ == 'pandas', \
        "pandas should be imported as pd"

def test_seaborn_loaded():
    assert sns.__name__ == 'seaborn', \
        "seaborn should be imported as sns"

def test_numpy_loaded():
    assert np.__name__ == 'numpy', \
        "numpy should be imported as np"

import pandas as pd

def test_nobel_correctly_loaded():
    correct_nobel = pd.read_csv('datasets/nobel.csv')
    assert correct_nobel.equals(nobel), \
        "The variable nobel should contain the data in 'datasets/nobel.csv'"

def test_Wilhelm_was_selected():
    assert "Wilhelm Conrad" in last_value.to_string(), \
        "Hmm, it seems you have not displayed at least the first six entries of nobel. A fel
low named Wilhelm Conrad Röntgen should be displayed."
```

Out[146]: 5/5 tests passed

2. So, who gets the Nobel Prize?

Just looking at the first couple of prize winners, or Nobel laureates as they are also called, we already see a celebrity: Wilhelm Conrad Röntgen, the guy who discovered X-rays. And actually, we see that all of the winners in 1901 were guys that came from Europe. But that was back in 1901, looking at all winners in the dataset, from 1901 to 2016, which sex and which country is the most commonly represented?

(For *country*, we will use the `birth_country` of the winner, as the `organization_country` is `NaN` for all shared Nobel Prizes.)

In [147]:
```
# Display the number of (possibly shared) Nobel Prizes handed
# out between 1901 and 2016
display(len(nobel))

# Display the number of prizes won by male and female recipients.
display(nobel['sex'].value_counts())

# Display the number of prizes won by the top 10 nationalities.
nobel['birth_country'].value_counts().head(10)
```

```
911

Male      836
Female     49
Name: sex, dtype: int64
```

Out[147]:
```
United States of America    259
United Kingdom               85
Germany                      61
France                       51
Sweden                       29
Japan                        24
Canada                       18
Netherlands                  18
Russia                       17
Italy                        17
Name: birth_country, dtype: int64
```

In [148]:
```
%%nose
last_value = _

correct_value = nobel['birth_country'].value_counts().head(10)

def test_last_value_correct():
    assert last_value.equals(correct_value), \
        "The number of prizes won by the top 10 nationalities doesn't seem correct... Maybe
 check the hint?"
```

Out[148]: 1/1 tests passed

3. USA dominance

Not so surprising perhaps: the most common Nobel laureate between 1901 and 2016 was a man born in the United States of America. But in 1901 all the winners were European. When did the USA start to dominate the Nobel Prize charts?

In [149]:
```
# Calculating the proportion of USA born winners per decade
nobel['usa_born_winner'] = nobel['birth_country'] == 'United States of America'
nobel['decade'] = (np.floor(nobel['year'] / 10)*10).astype(int)
prop_usa_winners = nobel.groupby('decade', as_index=False)['usa_born_winner'].mean()

# Display the proportions of USA born winners per decade
display(prop_usa_winners)
```

	decade	usa_born_winner
0	1900	0.017544
1	1910	0.075000
2	1920	0.074074
3	1930	0.250000
4	1940	0.302326
5	1950	0.291667
6	1960	0.265823
7	1970	0.317308
8	1980	0.319588
9	1990	0.403846
10	2000	0.422764
11	2010	0.292683

In [150]:
```
%%nose

def test_decade_int():
    assert nobel['decade'].dtype == "int64", \
    "Hmm, it looks like the decade column isn't calculated correctly. Did you forget to conv
ert it to an integer?"

def test_correct_prop_usa_winners():
    correct_prop_usa_winners = nobel.groupby('decade', as_index=False)['usa_born_winner'].me
an()
    assert correct_prop_usa_winners.equals(prop_usa_winners), \
        "prop_usa_winners should contain the proportion of usa_born_winner by decade. Don't
 forget to set as_index=False in the groupby() method."
```

Out[150]: 2/2 tests passed

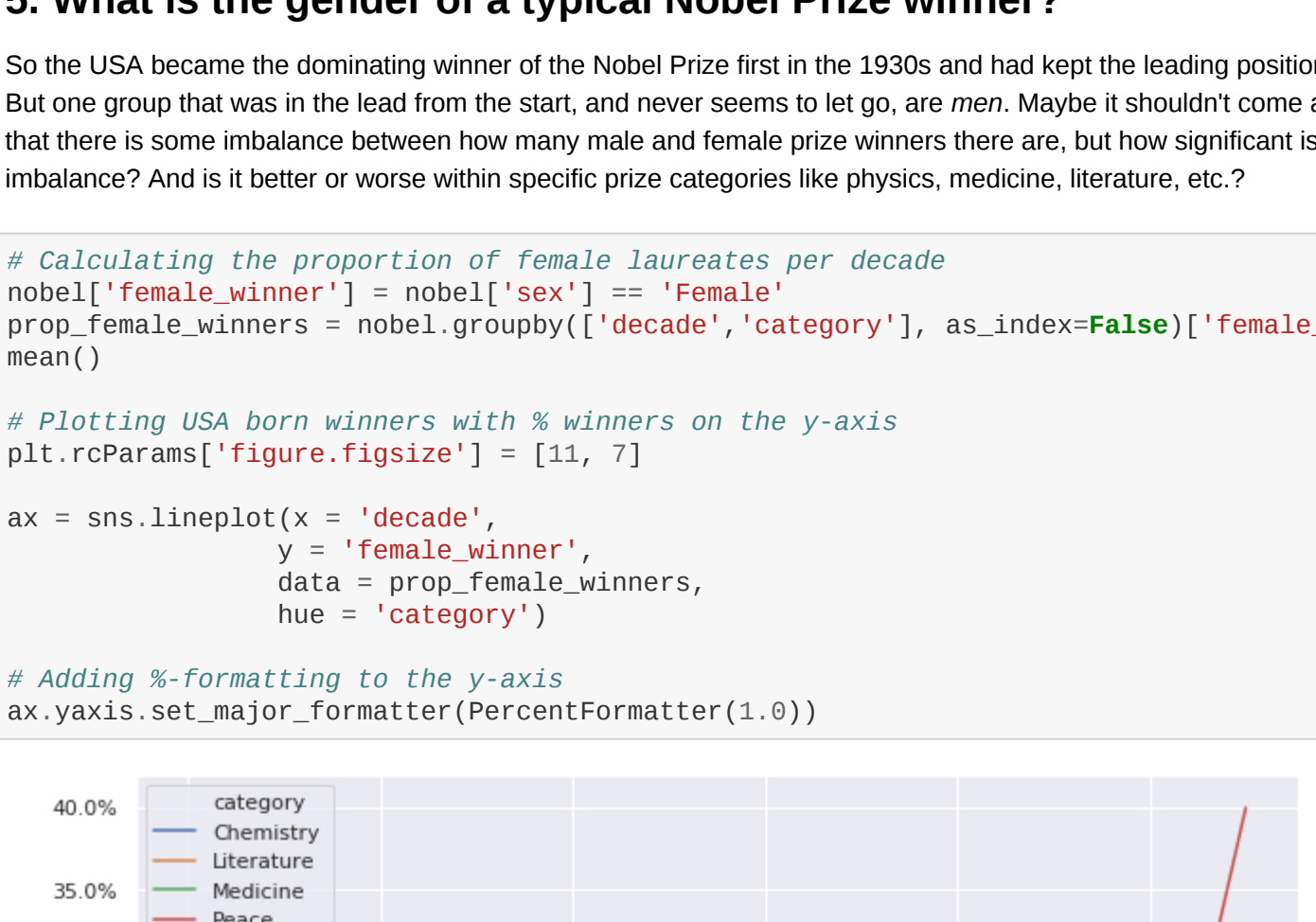
4. USA dominance, visualized

A table is OK, but to *see* when the USA started to dominate the Nobel charts we need a plot!

In [151]:
```
# Setting the plotting theme
sns.set()
# and setting the size of all plots.
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = [11, 7]

# Plotting USA born winners
ax = sns.lineplot(x = 'decade',
                  y = 'usa_born_winner',
                  data = prop_usa_winners)

# Adding %-formatting to the y-axis
from matplotlib.ticker import PercentFormatter
ax.yaxis.set_major_formatter(PercentFormatter(1.0))
```



In [152]:
```
%%nose

def test_y_axis():
    assert all(ax.get_lines()[0].get_ydata() == prop_usa_winners.usa_born_winner), \
    'The plot should be assigned to ax and have usa_born_winner on the y-axis'

def test_x_axis():
    assert all(ax.get_lines()[0].get_xdata() == prop_usa_winners.decade), \
    'The plot should be assigned to ax and have decade on the x-axis'
```

Out[152]: 2/2 tests passed

5. What is the gender of a typical Nobel Prize winner?

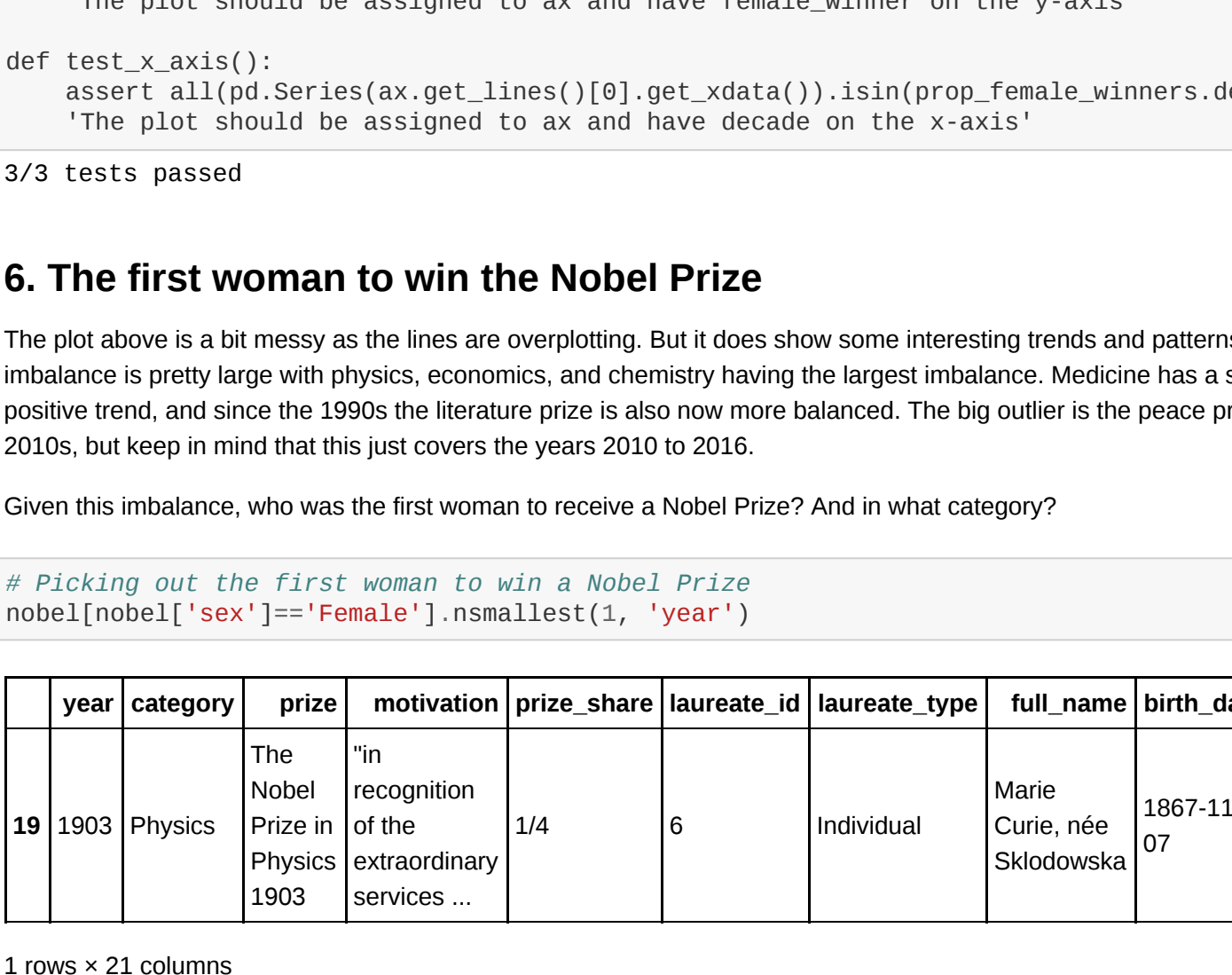
So the USA became the dominating winner of the Nobel Prize first in the 1930s and had kept the leading position ever since. But one group that was in the lead from the start, and never seems to let go, are *men*. Maybe it shouldn't come as a shock that there is some imbalance between how many male and female prize winners there are, but how significant is this imbalance? And is it better or worse within specific prize categories like physics, medicine, literature, etc.?

In [153]:
```
# Calculating the proportion of female laureates per decade
nobel['female_winner'] = nobel['sex'] == 'Female'
prop_female_winners = nobel.groupby(['decade','category'], as_index=False)['female_winner'].
mean()

# Plotting USA born winners with % winners on the y-axis
plt.rcParams['figure.figsize'] = [11, 7]

ax = sns.lineplot(x = 'decade',
                  y = 'female_winner',
                  data = prop_female_winners,
                  hue = 'category')

# Adding %-formatting to the y-axis
ax.yaxis.set_major_formatter(PercentFormatter(1.0))
```



In [154]:
```
%%nose


def test_correct_prop_usa_winners():
    correct_prop_female_winners = nobel.groupby(['decade', 'category'], as_index=False)['fem
ale_winner'].mean()
    assert correct_prop_female_winners.equals(prop_female_winners), \
        "prop_female_winners should contain the proportion of female_winner by decade. Don't
forget to set as_index=False in the groupby() method."

def test_y_axis():
    assert all(pd.Series(ax.get_lines()[0].get_ydata()).isin(prop_female_winners.female_winn
er)), \
    'The plot should be assigned to ax and have female_winner on the y-axis'

def test_x_axis():
    assert all(pd.Series(ax.get_lines()[0].get_xdata()).isin(prop_female_winners.decade)), \
    'The plot should be assigned to ax and have decade on the x-axis'
```

Out[154]: 3/3 tests passed

6. The first woman to win the Nobel Prize

The plot above is a bit messy as the lines are overplotting. But it does show some interesting trends and patterns. Overall the imbalance is pretty large with physics, economics, and chemistry having the largest imbalance. Medicine has a somewhat positive trend, and since the 1990s the literature prize is also now more balanced. The big outlier is the peace prize during the 2010s, but keep in mind that this just covers the years 2010 to 2016.

Given this imbalance, who was the first woman to receive a Nobel Prize? And in what category?

In [155]:
```
# Picking out the first woman to win a Nobel Prize
nobel[nobel['sex']=='Female'].nsmallest(1, 'year')
```

Out[155]:

	year	category	prize	motivation	prize_share	laureate_id	laureate_type	full_name	birth_date	birth_city
19	1903	Physics	The Nobel Prize in Physics 1903	"in recognition of the extraordinary services ...	1/4	6	Individual	Marie Curie, née Sklodowska	1867-11-07	Warsaw

1 rows × 21 columns

In [156]:
```
%%nose

last_value = _

def test_Marie_was_selected():
    assert "Marie Curie" in last_value.to_string(), \
        "Hmm, it seems you have not displayed the row of the first woman to win a Nobel Priz
e, her first name should be Marie."
```

Out[156]: 1/1 tests passed

7. Repeat laureates

For most scientists/writers/activists a Nobel Prize would be the crowning achievement of a long career. But for some people, one is just not enough, and few have gotten it more than once. Who are these lucky few? (Having won no Nobel Prize myself, I'll assume it's just about luck.)

In [157]:
```
# Selecting the laureates that have received 2 or more prizes.
nobel.groupby('full_name').filter(lambda group: len(group) >= 2)
```

Out[157]:

	year	category	prize	motivation	prize_share	laureate_id	laureate_type	full_name	birth_date
19	1903	Physics	The Nobel Prize in Physics 1903	"in recognition of the extraordinary services ...	1/4	6	Individual	Marie Curie, née Sklodowska	1867-11-07
62	1911	Chemistry	The Nobel Prize in Chemistry 1911	"in recognition of her services to the advance...	1/1	6	Individual	Marie Curie, née Sklodowska	1867-11-07
89	1917	Peace	The Nobel Peace Prize 1917	NaN	1/1	482	Organization	Comité international de la Croix Rouge (Intern...	NaN
215	1944	Peace	The Nobel Peace Prize 1944	NaN	1/1	482	Organization	Comité international de la Croix Rouge (Intern...	NaN
278	1954	Chemistry	The Nobel Prize in Chemistry 1954	"for his research into the nature of the chemi...	1/1	217	Individual	Linus Carl Pauling	1901-02-28
283	1954	Peace	The Nobel Peace Prize 1954	NaN	1/1	515	Organization	Office of the United Nations High Commissioner...	NaN
298	1956	Physics	The Nobel Prize in Physics 1956	"for their researches on semiconductors and th...	1/3	66	Individual	John Bardeen	1908-05-23
306	1958	Chemistry	The Nobel Prize in Chemistry 1958	"for his work on the structure of proteins, es...	1/1	222	Individual	Frederick Sanger	1918-08-13
340	1962	Peace	The Nobel Peace Prize 1962	NaN	1/1	217	Individual	Linus Carl Pauling	1901-02-28
348	1963	Peace	The Nobel Peace Prize 1963	NaN	1/2	482	Organization	Comité international de la Croix Rouge (Intern...	NaN
424	1972	Physics	The Nobel Prize in Physics 1972	"for their jointly developed theory of superco...	1/3	66	Individual	John Bardeen	1908-05-23
505	1980	Chemistry	The Nobel Prize in Chemistry 1980	"for their contributions concerning the determ...	1/4	222	Individual	Frederick Sanger	1918-08-13
523	1981	Peace	The Nobel Peace Prize 1981	NaN	1/1	515	Organization	Office of the United Nations High Commissioner...	NaN

13 rows × 21 columns

In [158]:
```
%%nose

last_value = _

def test_something():
    correct_last_value = nobel.groupby('full_name').filter(lambda group: len(group) >= 2)
    assert correct_last_value.equals(last_value), \
        "Did you use groupby followed by the filter method? Did you filter to keep only thos
e with >= 2 prises?"
```

Out[158]: 1/1 tests passed

8. How old are you when you get the prize?

The list of repeat winners contains some illustrious names! We again meet Marie Curie, who got the prize in physics for discovering radiation and in chemistry for isolating radium and polonium. John Bardeen got it twice in physics for transistors and superconductivity, Frederick Sanger got it twice in chemistry, and Linus Carl Pauling got it first in chemistry and later in peace for his work in promoting nuclear disarmament. We also learn that organizations also get the prize as both the Red Cross and the UNHCR have gotten it twice.

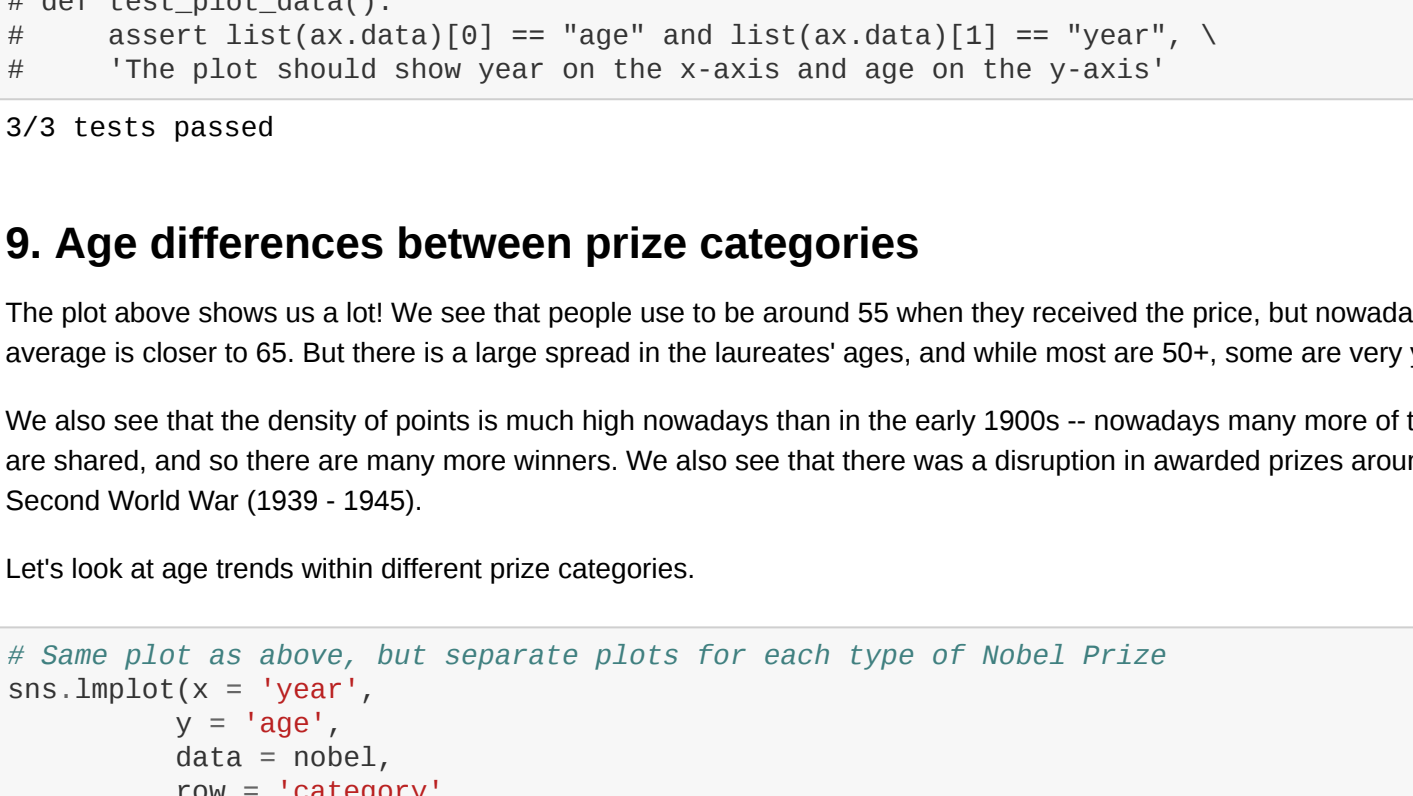
But how old are you generally when you get the prize?

In [159]:
```
# Converting birth_date from String to datetime
nobel['birth_date'] = pd.to_datetime(nobel['birth_date'])

# Calculating the age of Nobel Prize winners
nobel['age'] = nobel['year'] - nobel['birth_date'].dt.year

# Plotting the age of Nobel Prize winners
sns.lmplot(x = 'year',
          y = 'age',
          data =  nobel,
          lowess = True,
          aspect = 2,
          line_kws={'color' : 'black'})
```

Out[159]: <seaborn.axisgrid.FacetGrid at 0x7f07f2946a58>



In [160]:
```
%%nose

ax = _

def test_birth_date():
    assert pd.to_datetime(nobel['birth_date']).equals(nobel['birth_date']), \
        "Have you converted nobel['birth_date'] using to_datetime?"


def test_year():
    assert (nobel['year'] - nobel['birth_date'].dt.year).equals(nobel['age']), \
        "Have you caluclated nobel['year'] correctly?"

def test_plot_data():
    assert list(ax.data)[0] in ["age", "year"] and list(ax.data)[1] in ["age", "year"], \
    'The plot should show year on the x-axis and age on the y-axis'

# Why not this testing code?
# def test_plot_data():
#     assert list(ax.data)[0] == "age" and list(ax.data)[1] == "year", \
#     'The plot should show year on the x-axis and age on the y-axis'
```

Out[160]: 3/3 tests passed

9. Age differences between prize categories

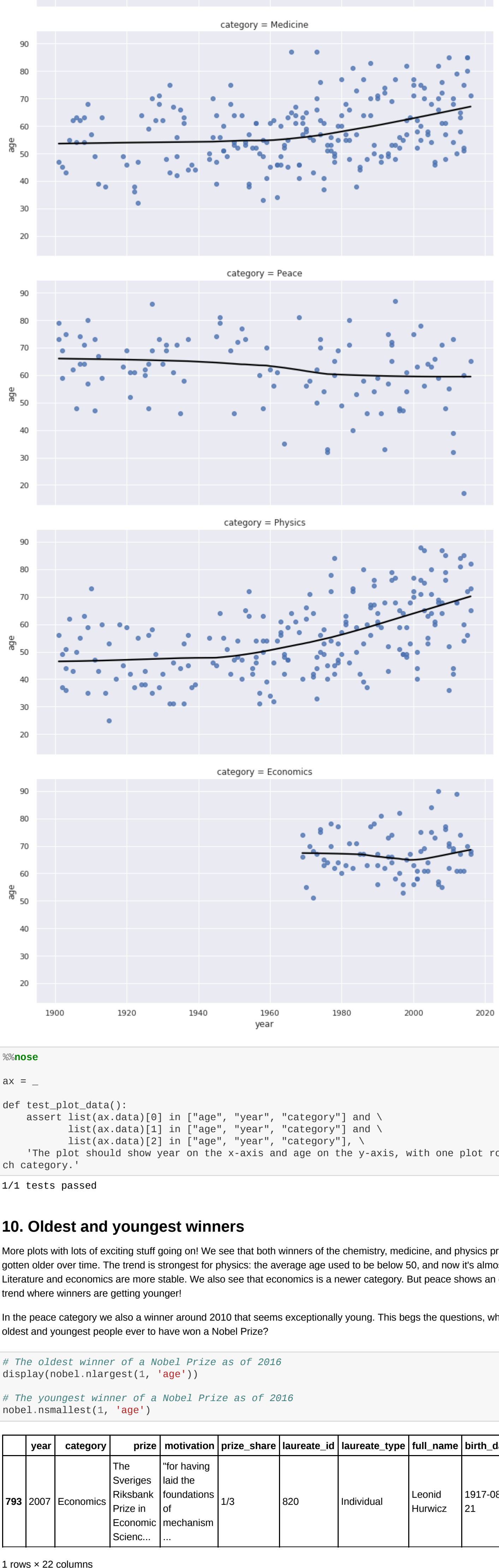
The plot above shows us a lot! We see that people use to be around 55 when they received the price, but nowadays the average is closer to 65. But there is a large spread in the laureates' ages, and while most are 50+, some are very young.

We also see that the density of points is much high nowadays than in the early 1900s -- nowadays many more of the prizes are shared, and so there are many more winners. We also see that there was a disruption in awarded prizes around the Second World War (1939 - 1945).

Let's look at age trends within different prize categories.

In [161]:
```
# Same plot as above, but separate plots for each type of Nobel Prize
sns.lmplot(x = 'year',
          y = 'age',
          data = nobel,
          row = 'category',
          lowess = True,
          aspect = 2,
          line_kws={'color' : 'black'})
```

Out[161]: <seaborn.axisgrid.FacetGrid at 0x7f07f2c84fd0>



In [162]:
```
%%nose

ax = _

def test_plot_data():
    assert list(ax.data)[0] in ["age", "year", "category"] and \
           list(ax.data)[1] in ["age", "year", "category"] and \
           list(ax.data)[2] in ["age", "year", "category"], \
    'The plot should show year on the x-axis and age on the y-axis, with one plot row for ea
ch category.'
```

Out[162]: 1/1 tests passed

10. Oldest and youngest winners

More plots with lots of exciting stuff going on! We see that both winners of the chemistry, medicine, and physics prize have gotten older over time. The trend is strongest for physics: the average age used to be below 50, and now it's almost 70. Literature and economics are more stable. We also see that economics is a newer category. But peace shows an opposite trend where winners are getting younger!

In the peace category we also a winner around 2010 that seems exceptionally young. This begs the questions, who are the oldest and youngest people ever to have won a Nobel Prize?

In [163]:
```
# The oldest winner of a Nobel Prize as of 2016
display(nobel.nlargest(1, 'age'))

# The youngest winner of a Nobel Prize as of 2016
nobel.nsmallest(1, 'age')
```

	year	category	prize	motivation	prize_share	laureate_id	laureate_type	full_name	birth_date	birth_cit
793	2007	Economics	The Sveriges Riksbank Prize in Economic Scienc...	"for having laid the foundations of mechanism ...	1/3	820	Individual	Leonid Hurwicz	1917-08-21	Moscow

1 rows × 22 columns

Out[163]:

	year	category	prize	motivation	prize_share	laureate_id	laureate_type	full_name	birth_date	birth_city	...
885	2014	Peace	The Nobel Peace Prize 2014	"for their struggle against the suppression of...	1/2	914	Individual	Malala Yousafzai	1997-07-12	Mingora	...

1 rows × 22 columns

In [164]:
```
%%nose

last_value = _

def test_oldest_or_youngest():
    assert 'Hurwicz' in last_value.to_string() or 'Yousafzai' in last_value.to_string(), \
        "Have you displayed the row of the oldest winner and the row of the youngest winne
r?"
```

Out[164]: 1/1 tests passed

11. You get a prize!

Hey! You get a prize for making it to the very end of this notebook! It might not be a Nobel Prize, but I made it myself in paint so it should count for something. But don't despair, Leonid Hurwicz was 90 years old when he got his prize, so it might not be too late for you. Who knows.

Before you leave, what was again the name of the youngest winner ever who in 2014 got the prize for "[her] struggle against the suppression of children and young people and for the right of all children to education"?

In [165]:
```
# The name of the youngest winner of the Nobel Prize as of 2016
youngest_winner = 'Malala Yousafzai'
```

In [166]:
```
%%nose

import re

def test_right_name():
    assert re.match("(malala|yousafzai)", youngest_winner.lower()), \
        "youngest_winner should be a string. Try writing only the first / given name."
```

Out[166]: 1/1 tests passed