Project 3: Regression analysis of Gapminder data

CMSC320

Mukun Guo, 18 April 2020

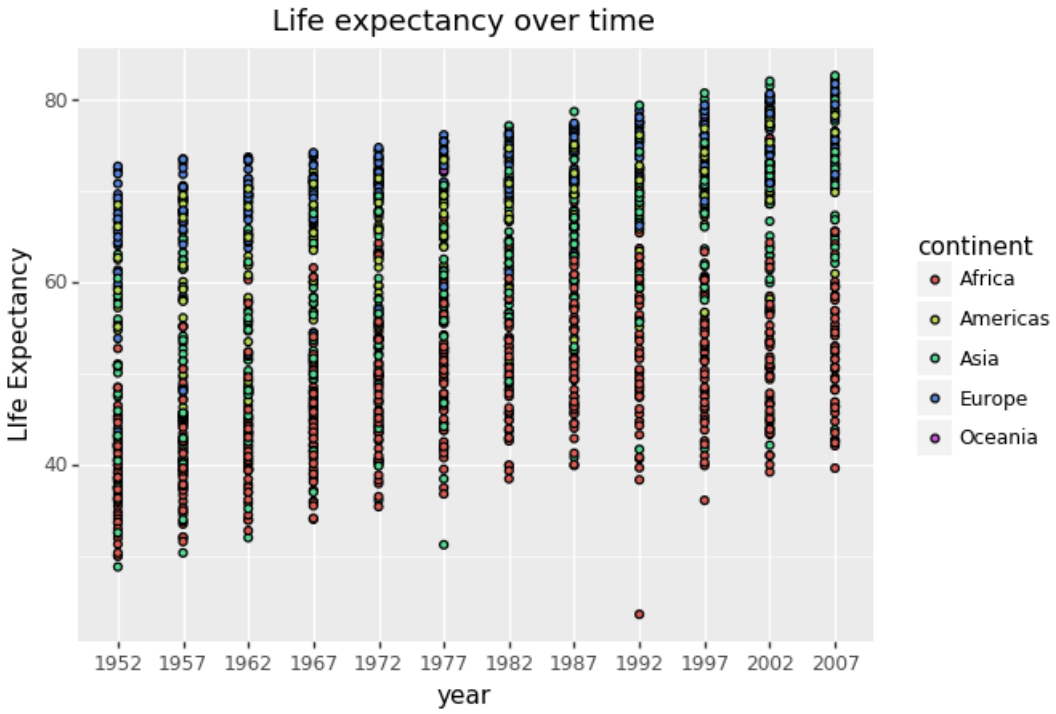
In [1]:
```
import pandas as pd
from plotnine import *
from matplotlib import pyplot as plt

df = pd.read_csv('./data/gapminder.csv')
df.head(3)
```

Out[1]:

	country	continent	year	lifeExp	pop	gdpPercap
0	Afghanistan	Asia	1952	28.801	8425333	779.445314
1	Afghanistan	Asia	1957	30.332	9240934	820.853030
2	Afghanistan	Asia	1962	31.997	10267083	853.100710

Exercise 1

In [2]:
```
p = (ggplot(df, aes(x='factor(year)', y='lifeExp', fill='continent'))
     + geom_point()
     + labs(y="Llfe Expectancy", x = "year", title='Life expectancy over time')
    )

p
```



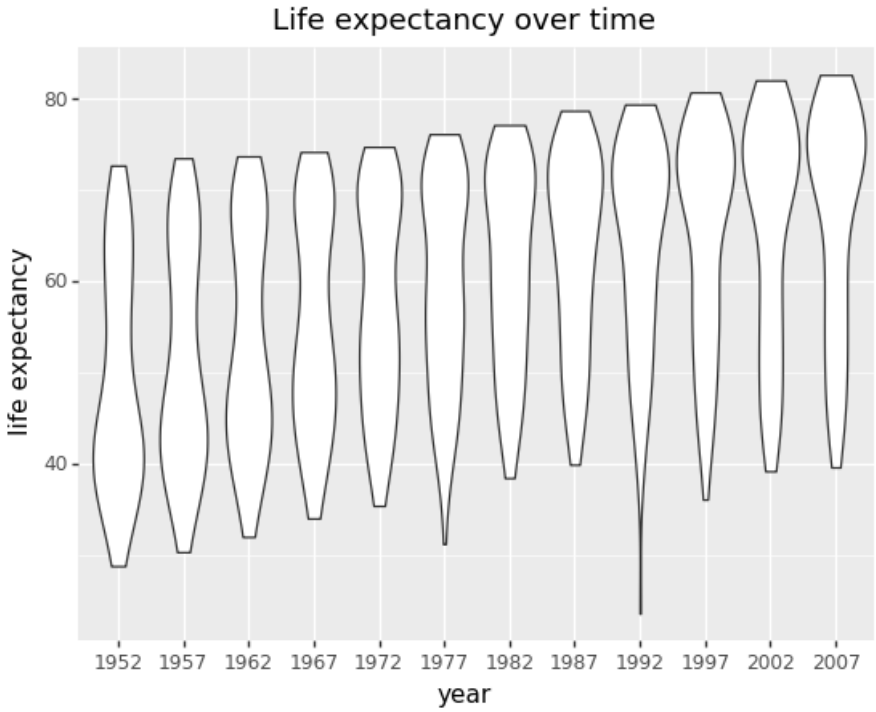
Out[2]: <ggplot: (7550661165)>

Question 1

from the plot, we can see that life expectancy is positively correlated to year, that is, life expectancy gets higher and over the years. The trend looks linear

Question 2

In [3]:
```
p = (ggplot(df, aes(x='factor(year)', y='lifeExp'))
     + geom_violin()
     + labs(y="life expectancy", x = "year", title='Life expectancy over time')
    )

p
```



Out[3]: <ggplot: (7551348509)>

Intuitvely, I would say that life expectancy differs a lot for different countries. It is skewed, especially after 1980s. It is bimodal in the 1960s and 1950s but unimodal after 1970s. It's not symmetry around its center. It is bimodal in some years (1960s), but gradually becomes unimodal after 1980s.

Question 3

I would reject the null hopythesis of no relationship

Question 4

Basically, the violin plot of residuals will look similar to the above one but shift downward as a whole. To be more precise, each "violin" will have the same shape but it will be (roughly) centered around 0. Aditionally, I'm expecting a larger shift for data after 1980s as they have a larger expected life expectancy

Question 5

The residuals should be normally distributed, which means that the (half) violin of each year should be centered around 0 and have the shape of a normal distribution.

Exercise 2

In [4]:
```
p = (ggplot(df, aes(x='year', y='lifeExp'))
     + geom_point()
     + geom_smooth(method='lm')
     + labs(y="Llfe Expectancy", x = "year", title='Life expectancy over time')
    )

p
```

```
/Users/guomukun/opt/anaconda3/lib/python3.7/site-packages/numpy/core/fromnumeric.py:2495: FutureWarni
ng: Method .ptp is deprecated and will be removed in a future version. Use numpy.ptp instead.
  return ptp(axis=axis, out=out, **kwargs)
```

Out[4]:
```
<ggplot: (279240361)>
```

In [5]:
```
import statsmodels.formula.api as sm
simple_res = sm.ols('lifeExp~year', data=df).fit()
simple_res.params
```

Out[5]:
```
Intercept   -585.652187
year           0.325904
dtype: float64
```

Question 6

In [6]:
```
print("The slope of the regression line is {:.4f}".format(simple_res.params['year']))
```

```
The slope of the regression line is 0.3259
```

we can see that on average, life expectancy increase about 0.3259 every year around the world

Question 7

The null hypothesis is there is no relationship between year and life expectancy, in other words, lets say $y = m \times x + b$, then $m = 0$ means that there is no relationship between year and life expectancy. Then we can perform the hypothesis test to determine whether we can reject the hypothesis.

In [10]:
```
from scipy.stats import norm

se = simple_res.bse['year']
p_value = 1 - norm.cdf(simple_res.params['year'], loc=0, scale=se ** .5)


if p_value < 0.05:
    print('We reject the null hypothesis as p-value={:.4f} < 0.05'.format(p_value))
else:
    print('We accept the null hypothesis as p-avlue={:.4f} > 0.05'.format(p_value))
```

```
We reject the null hypothesis as p-value=0.0054 < 0.05
```

Hence we should reject the null hypothesis

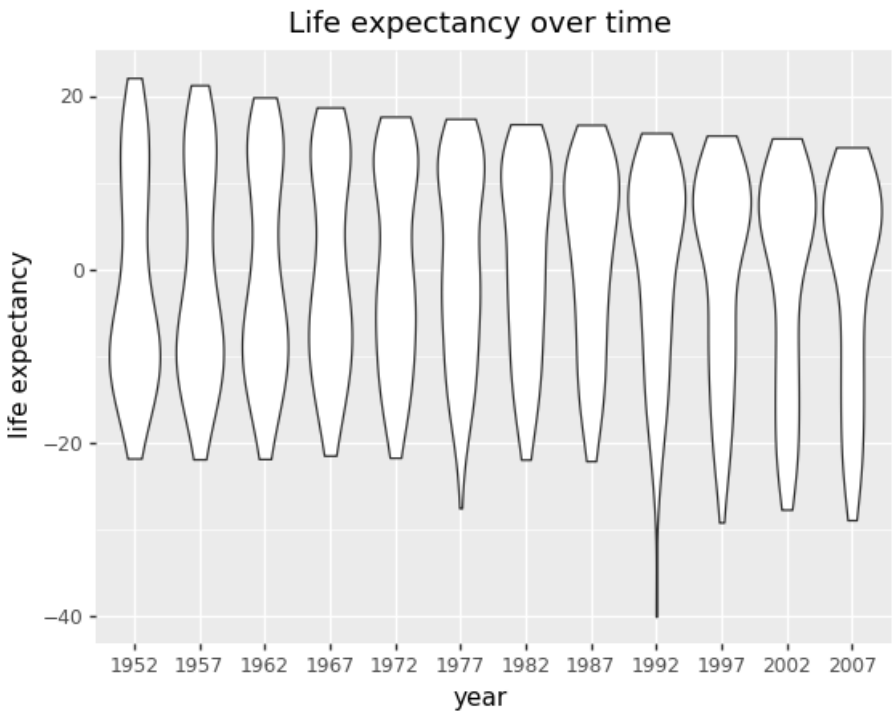
Exercise 3

In [11]:
```
lifeExp = df.copy()
lifeExp['fitted'] = simple_res.fittedvalues
lifeExp['resid'] = simple_res.resid

lifeExp.head(3)
```

Out[11]:

	country	continent	year	lifeExp	pop	gdpPercap	fitted	resid
0	Afghanistan	Asia	1952	28.801	8425333	779.445314	50.512084	-21.711084
1	Afghanistan	Asia	1957	30.332	9240934	820.853030	52.141603	-21.809603
2	Afghanistan	Asia	1962	31.997	10267083	853.100710	53.771122	-21.774122

In [12]:
```
p = (ggplot(lifeExp, aes(x='factor(year)', y='resid'))
     + geom_violin()
     + labs(y="life expectancy", x = "year", title='Life expectancy over time')
    )

p
```



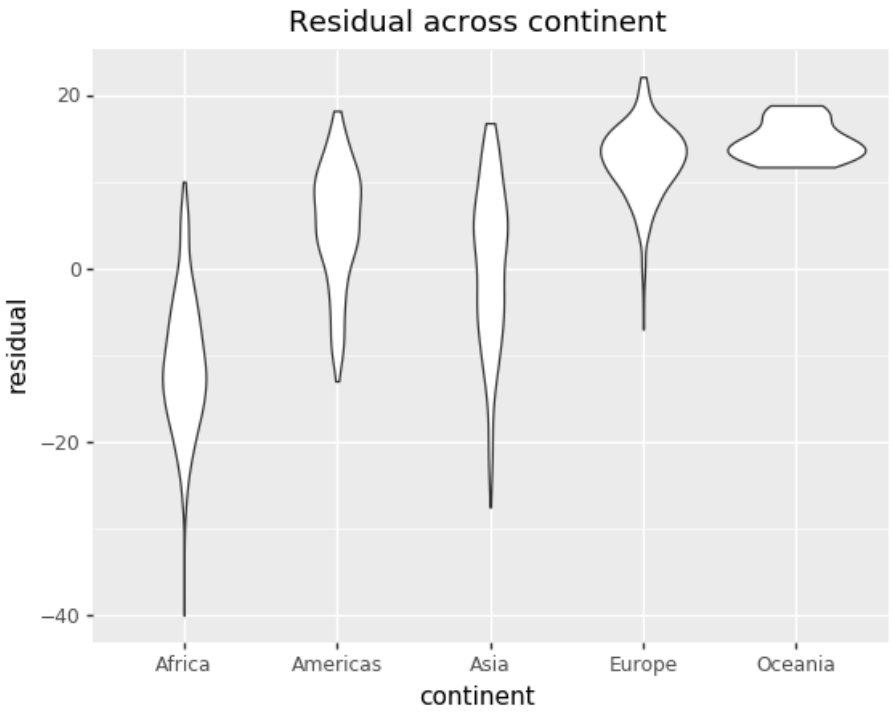
Out[12]:
```
<ggplot: (7551530049)>
```

Question 8

Yes. It matches my expectation.

Exercise 4

In [13]:
```
p = (ggplot(lifeExp, aes(x='factor(continent)', y='resid'))
     + geom_violin()
     + labs(y="residual", x = "continent", title='Residual across continent')
    )

p
```



Out[13]: <ggplot: (7551609393)>

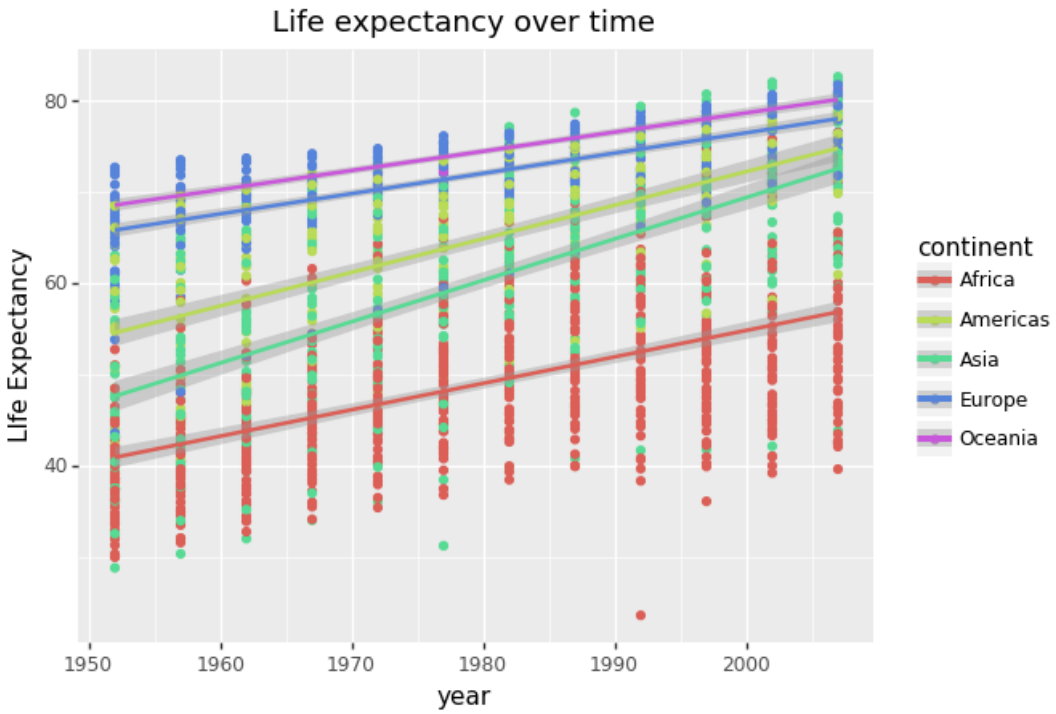
Question 9

Yes. There is an obvious dependence between residuals and continent. The prediction is generally higher than the actual life expectancy in Africa, but less than the actual life expectancy in Europe, Americas, and Oceania. My suggestion for performing a regression analysis would be to group the data by continent first, then perform regression analysis on data sampled from population of each continent.

Exercise 5

In [14]:
```
p = (ggplot(df, aes(x='year', y='lifeExp', color='continent'))
     + geom_point()
     + geom_smooth(method='lm')
     + labs(y="Llfe Expectancy", x = "year", title='Life expectancy over time')
    )

p
```

```
/Users/guomukun/opt/anaconda3/lib/python3.7/site-packages/numpy/core/fromnumeric.py:2495: FutureWarni
ng: Method .ptp is deprecated and will be removed in a future version. Use numpy.ptp instead.
  return ptp(axis=axis, out=out, **kwargs)
```



Out[14]: <ggplot: (7552313577)>

Question 10

Yes. There should be an interaction term for continent and year. As we can tell from the plot, the regression line is quite different for each continent, which means that continent could be another predictor for life expectancy. By including an interaction term we could better capture the effect of how the combination of continent and year affect the life expectancy (i.e. We could answer the question such as will year have a positive effect for life expectancy in Aisa? what about in Europe?)

Exercise 6

In [17]:
```
interact_res = sm.ols('lifeExp~year+continent+year*continent', data=df).fit()
interact_res.summary()
```

Out[17]:

OLS Regression Results

Dep. Variable:	lifeExp	**R-squared:**	0.693
Model:	OLS	**Adj. R-squared:**	0.691
Method:	Least Squares	**F-statistic:**	424.3
Date:	Tue, 21 Apr 2020	**Prob (F-statistic):**	0.00
Time:	08:48:00	**Log-Likelihood:**	-5771.9
No. Observations:	1704	**AIC:**	1.156e+04
Df Residuals:	1694	**BIC:**	1.162e+04
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-524.2578	32.963	-15.904	0.000	-588.911	-459.605
continent[T.Americas]	-138.8484	57.851	-2.400	0.016	-252.315	-25.382
continent[T.Asia]	-312.6330	52.904	-5.909	0.000	-416.396	-208.870
continent[T.Europe]	156.8469	54.498	2.878	0.004	49.957	263.737
continent[T.Oceania]	182.3499	171.283	1.065	0.287	-153.599	518.298
year	0.2895	0.017	17.387	0.000	0.257	0.322
year:continent[T.Americas]	0.0781	0.029	2.673	0.008	0.021	0.135
year:continent[T.Asia]	0.1636	0.027	6.121	0.000	0.111	0.216
year:continent[T.Europe]	-0.0676	0.028	-2.455	0.014	-0.122	-0.014
year:continent[T.Oceania]	-0.0793	0.087	-0.916	0.360	-0.249	0.090

Omnibus:	27.121	**Durbin-Watson:**	0.242
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	44.106
Skew:	-0.121	**Prob(JB):**	2.65e-10
Kurtosis:	3.750	**Cond. No.**	2.09e+06

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.09e+06. This might indicate that there are
strong multicollinearity or other numerical problems.

In [18]:
```
interact_res.params
```

Out[18]:
```
Intercept                   -524.257846
continent[T.Americas]       -138.848447
continent[T.Asia]           -312.633049
continent[T.Europe]          156.846852
continent[T.Oceania]         182.349883
year                           0.289529
year:continent[T.Americas]     0.078122
year:continent[T.Asia]         0.163593
year:continent[T.Europe]      -0.067597
year:continent[T.Oceania]     -0.079257
dtype: float64
```

Question 11:

We will determine whether the parameters is significantly different from zero by performing hypothesis test on the parameters. The null assumption is the parameters are zero.

In [19]:
```
interact_res.bse
for param in ['continent[T.Americas]', 'continent[T.Asia]', 'continent[T.Europe]', 'continent[T.Oceania]', 'year', 'year:continent[T.Americas]', 'year:continent[T.Asia]', 'year:continent[T.Europe]', 'year:continent[T.Oceania]']:
    p = 1 - norm.cdf(abs(interact_res.params[param]), loc=0, scale=interact_res.bse[param] ** .5)
#     print(p)
    if p < 0.05:
        print('parameter "{}" is significantly different from zero because we can reject the null hypothesis for this parameter (p-value={:.3f} < 0.05)'.format(param, p))
    else:
        print('parameter "{}" is not significantly different from zero because we accept the null hypothesis for this parameter (p-value={:.3f} > 0.05)'.format(param, p))
```

```
parameter "continent[T.Americas]" is significantly different from zero because we can reject the null hypothesis for this parameter (p-value=0.000 < 0.05)
parameter "continent[T.Asia]" is significantly different from zero because we can reject the null hypothesis for this parameter (p-value=0.000 < 0.05)
parameter "continent[T.Europe]" is significantly different from zero because we can reject the null hypothesis for this parameter (p-value=0.000 < 0.05)
parameter "continent[T.Oceania]" is significantly different from zero because we can reject the null hypothesis for this parameter (p-value=0.000 < 0.05)
parameter "year" is significantly different from zero because we can reject the null hypothesis for this parameter (p-value=0.012 < 0.05)
parameter "year:continent[T.Americas]" is not significantly different from zero because we accept the null hypothesis for this parameter (p-value=0.324 > 0.05)
parameter "year:continent[T.Asia]" is not significantly different from zero because we accept the null hypothesis for this parameter (p-value=0.158 > 0.05)
parameter "year:continent[T.Europe]" is not significantly different from zero because we accept the null hypothesis for this parameter (p-value=0.342 > 0.05)
parameter "year:continent[T.Oceania]" is not significantly different from zero because we accept the null hypothesis for this parameter (p-value=0.394 > 0.05)
```

To summarize, we conclude that parameters **"year:continent[T.Americas]", "year:continent[T.Asia]", "year:continent[T.Europe]", "year:continent[T.Europe]"** are not significantly different from zero because their p-value > 0.05

Question 12

In [20]:
```
slopes = interact_res.params

print('Life expectancy increases by {:.4f} for America'.format(interact_res.params['year'] + interact_res.params['year:continent[T.Americas]']))
print('Life expectancy increases by {:.4f} for Europe'.format(interact_res.params['year'] + interact_res.params['year:continent[T.Europe]']))
print('Life expectancy increases by {:.4f} for Asia'.format(interact_res.params['year'] + interact_res.params['year:continent[T.Asia]']))
print('Life expectancy increases by {:.4f} for Oceania'.format(interact_res.params['year'] + interact_res.params['year:continent[T.Oceania]']))
print('Life expectancy increases by {:.4f} for Africa'.format(interact_res.params['year']))
```

```
Life expectancy increases by 0.3677 for America
Life expectancy increases by 0.2219 for Europe
Life expectancy increases by 0.4531 for Asia
Life expectancy increases by 0.2103 for Oceania
Life expectancy increases by 0.2895 for Africa
```

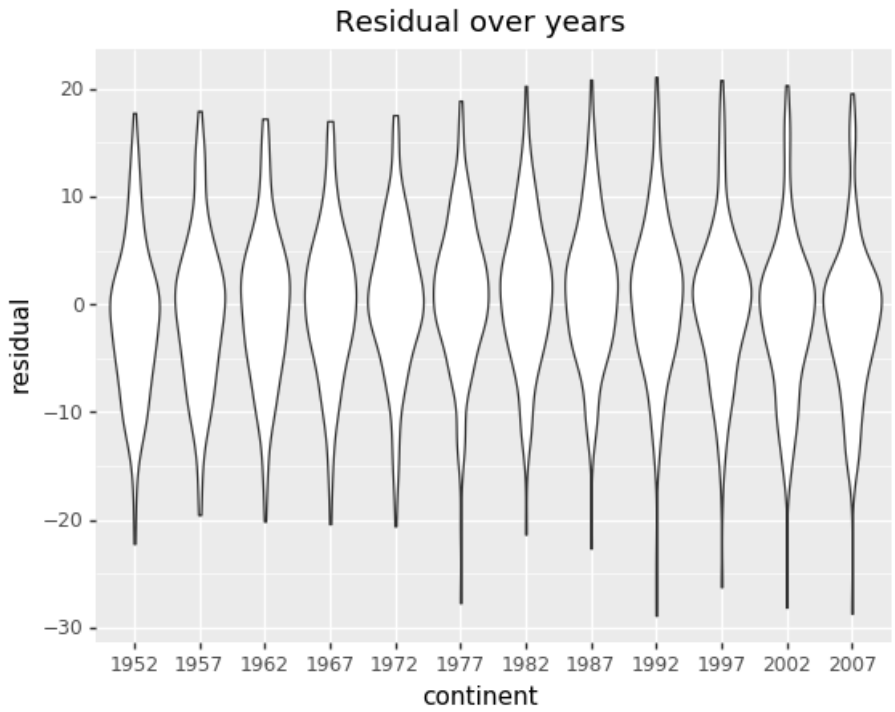
Exercise 7

In [21]:
```
lifeExp['fitted_interact'] = interact_res.fittedvalues
lifeExp['resid_interact'] = interact_res.resid
lifeExp.head(3)
```

Out[21]:

	country	continent	year	lifeExp	pop	gdpPercap	fitted	resid	fitted_interact	resid_interact
0	Afghanistan	Asia	1952	28.801	8425333	779.445314	50.512084	-21.711084	47.604037	-18.803037
1	Afghanistan	Asia	1957	30.332	9240934	820.853030	52.141603	-21.809603	49.869649	-19.537649
2	Afghanistan	Asia	1962	31.997	10267083	853.100710	53.771122	-21.774122	52.135261	-20.138261

In [22]:
```
p = (ggplot(lifeExp, aes(x='factor(year)', y='resid_interact'))
     + geom_violin()
     + labs(y="residual", x = "continent", title='Residual over years')
    )

p
```

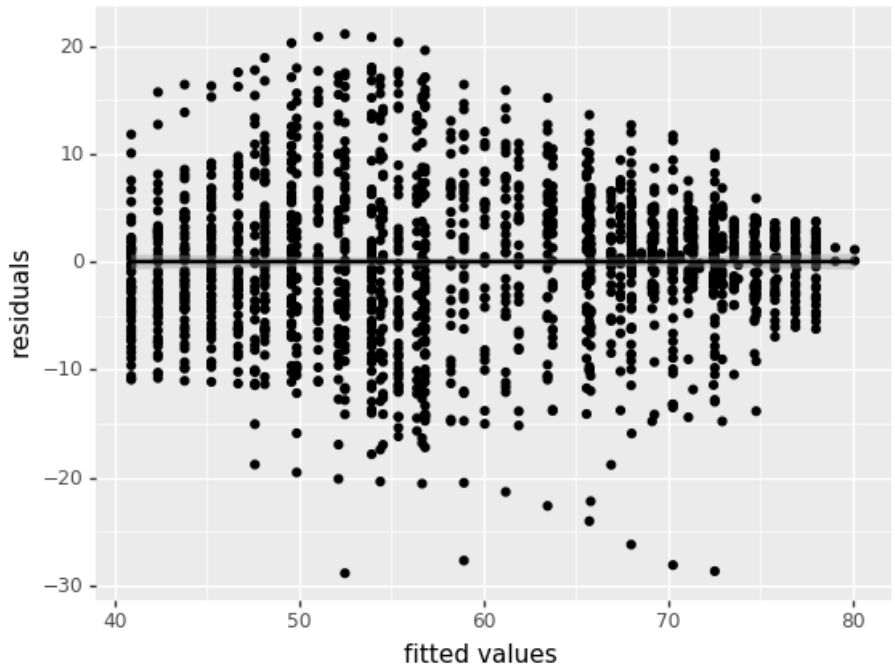


Out[22]: <ggplot: (7551682725)>

From the residuals vs. year violin plot of the interaction model, we can see that the assumptions we made for linear regression are generally well testified. Specifically, we can see that each (half) violin is **close to** a normal distribution with residual=0 as center.

In [23]:
```
p = (ggplot(lifeExp, aes(x='fitted_interact', y='resid_interact'))
     + geom_point()
     + geom_smooth()
     + labs(y="residuals", x = "fitted values", title='')
    )

p
```

```
/Users/guomukun/opt/anaconda3/lib/python3.7/site-packages/numpy/core/fromnumeric.py:2495: FutureWarni
ng: Method .ptp is deprecated and will be removed in a future version. Use numpy.ptp instead.
  return ptp(axis=axis, out=out, **kwargs)
```



Out[23]: <ggplot: (7552315033)>

By our assumption, the average of residuals should be zero. From the plot we can see that the distribution of residuals attest our assumption quite well. This can also be told by observing the line in the above graph (fitted using `geom_smooth()`). The spread (variance), however, is not completely constant, which shows that our regression may not be perfect for the case.