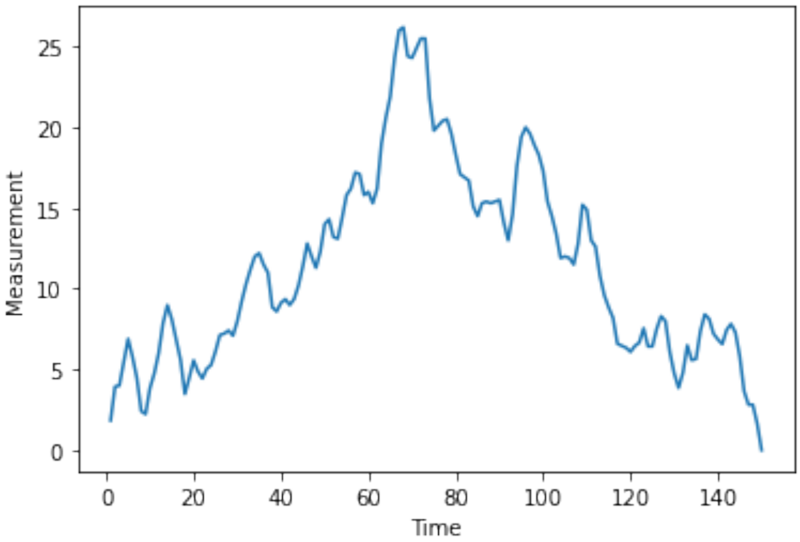
DSC 275: HW #3 Sunishka Misala

In [63]:
```
import pandas as pd
import numpy as np
import seaborn as sns
from pandas import Series
from matplotlib import pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.stattools import acf, pacf
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
import statsmodels.api as sm
```

QUESTION 1:

In [64]:
```
first = pd.read_csv('Measurement_Q1.csv')
```

In [65]:
```
x = first.Time
y = first.Measurement
plt.plot (x, y)
plt.xlabel ('Time')
plt.ylabel ('Measurement')
plt.show()
```



In [66]:
```
measurement_model = ARIMA(y, order=(0,1,1))
```

In [67]:
```
measurement_model_fit = measurement_model.fit()
```

In [68]:
```
measurement_model_fit.summary()
```

Out[68]:

SARIMAX Results

Dep. Variable:	Measurement	**No. Observations:**	150
Model:	ARIMA(0, 1, 1)	**Log Likelihood**	-202.609
Date:	Sun, 03 Oct 2021	**AIC**	409.217
Time:	01:09:19	**BIC**	415.225
Sample:	0	**HQIC**	411.658
	- 150		
Covariance Type:	opg		

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
ma.L1	0.7531	0.060	12.573	0.000	0.636	0.871
sigma2	0.8834	0.109	8.080	0.000	0.669	1.098

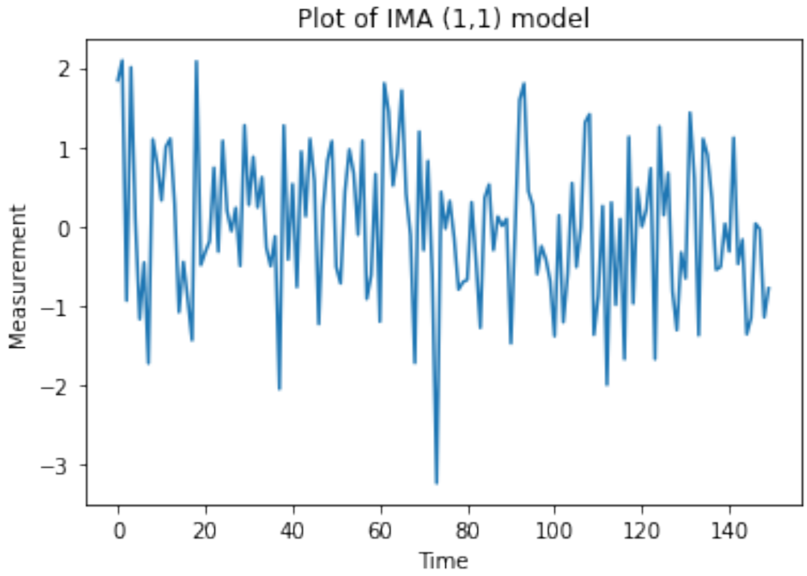
Ljung-Box (L1) (Q):	0.26	**Jarque-Bera (JB):**	1.10
Prob(Q):	0.61	**Prob(JB):**	0.58
Heteroskedasticity (H):	1.00	**Skew:**	-0.21
Prob(H) (two-sided):	0.99	**Kurtosis:**	2.98

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

In [69]:

```
from pandas import DataFrame as df
model_res = df(measurement_model_fit.resid)
plt.plot (model_res)
plt.xlabel ('Time')
plt.ylabel ('Measurement')
plt.title ('Plot of IMA (1,1) model')
```

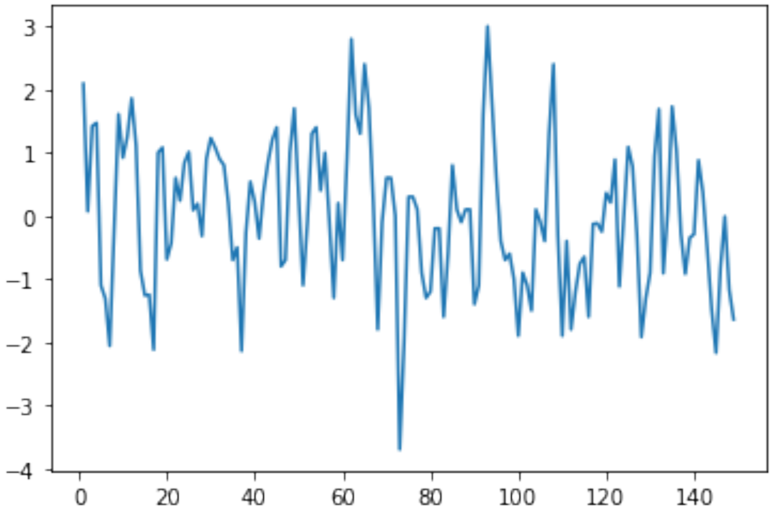
Out[69]: Text(0.5, 1.0, 'Plot of IMA (1,1) model')



b)

In [70]:
```
diff = y.diff()
```

In [71]:
```
plt.plot(diff)
plt.show()
```



In [72]:
```
diff = diff.iloc[1:]
```

In [73]:
```
diff
```

Out[73]:
```
1       2.09
2       0.07
3       1.42
4       1.47
5      -1.10
        ...
145    -2.17
146    -0.83
147    -0.01
148    -1.17
149    -1.64
Name: Measurement, Length: 149, dtype: float64
```

In [74]:
```
model2 = ARIMA(diff, order=(0,0,1))
```

In [75]:
```
model_fit2 = model2.fit()
model_fit2.summary()
```

Out[75]:

SARIMAX Results

Dep. Variable:	Measurement	**No. Observations:**	149
Model:	ARIMA(0, 0, 1)	**Log Likelihood**	-202.607
Date:	Sun, 03 Oct 2021	**AIC**	411.215
Time:	01:09:23	**BIC**	420.227
Sample:	0	**HQIC**	414.876
	- 149		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.0064	0.137	-0.047	0.962	-0.274	0.261
ma.L1	0.7531	0.061	12.392	0.000	0.634	0.872
sigma2	0.8834	0.110	8.052	0.000	0.668	1.098

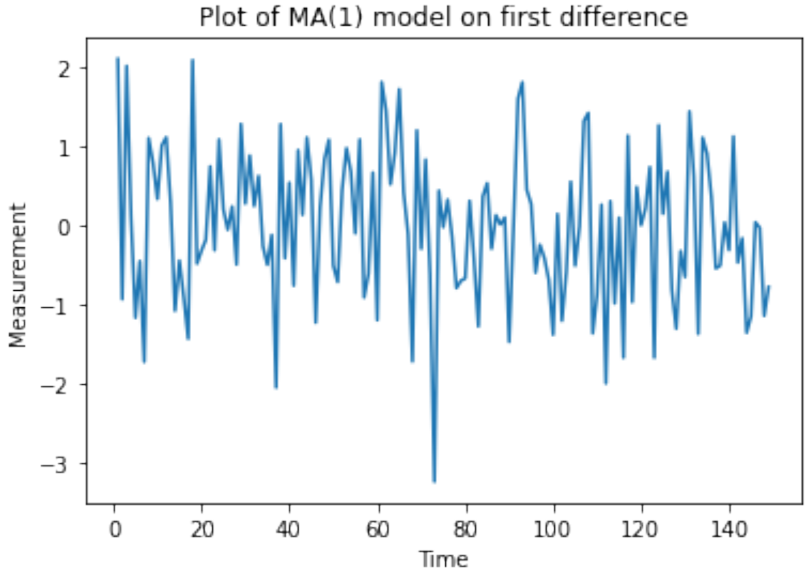
Ljung-Box (L1) (Q):	0.26	**Jarque-Bera (JB):**	1.10
Prob(Q):	0.61	**Prob(JB):**	0.58
Heteroskedasticity (H):	0.99	**Skew:**	-0.21
Prob(H) (two-sided):	0.98	**Kurtosis:**	2.98

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

In [76]:

```
model_res2 = df(model_fit2.resid)
plt.plot (model_res2)
plt.xlabel ('Time')
plt.ylabel ('Measurement')
plt.title ('Plot of MA(1) model on first difference')
```

Out[76]: Text(0.5, 1.0, 'Plot of MA(1) model on first difference')



d)a and c are the same because the result parameters are the same.

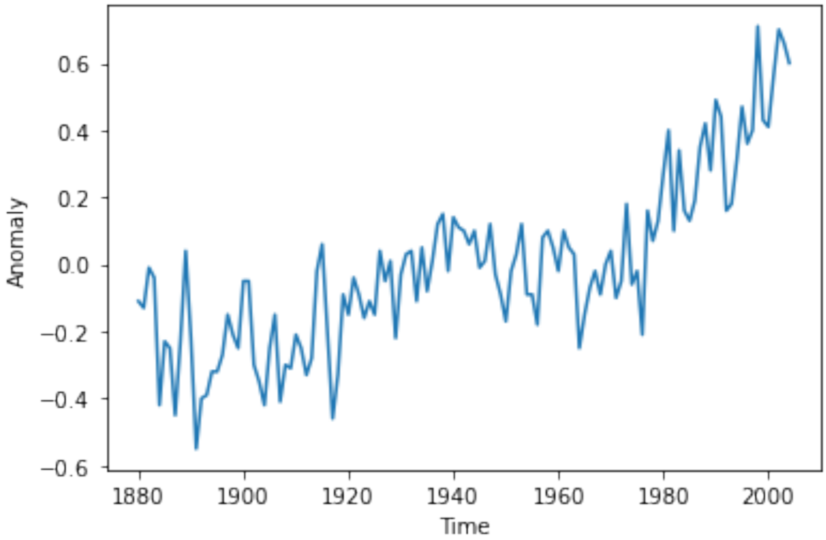
QUESTION 2:

In [77]:

```
data= pd.read_csv('GlobalAirTemperature.csv')
```

In [78]:

```
x = data.Year
y = data["Anomaly, C"]
plt.plot (x, y)
plt.xlabel ('Time')
plt.ylabel ('Anomaly')
plt.show()
```



In [79]:

```
measurement_model11 = ARIMA(y, order=(0,1,1))
measurement_model_fit11 = measurement_model11.fit()
measurement_model_fit11.summary()
```

Out[79]:

SARIMAX Results

Dep. Variable:	Anomaly, C	**No. Observations:**	125
Model:	ARIMA(0, 1, 1)	**Log Likelihood**	72.910
Date:	Sun, 03 Oct 2021	**AIC**	-141.821
Time:	01:09:28	**BIC**	-136.180
Sample:	0	**HQIC**	-139.530
	- 125		
Covariance Type:	opg		

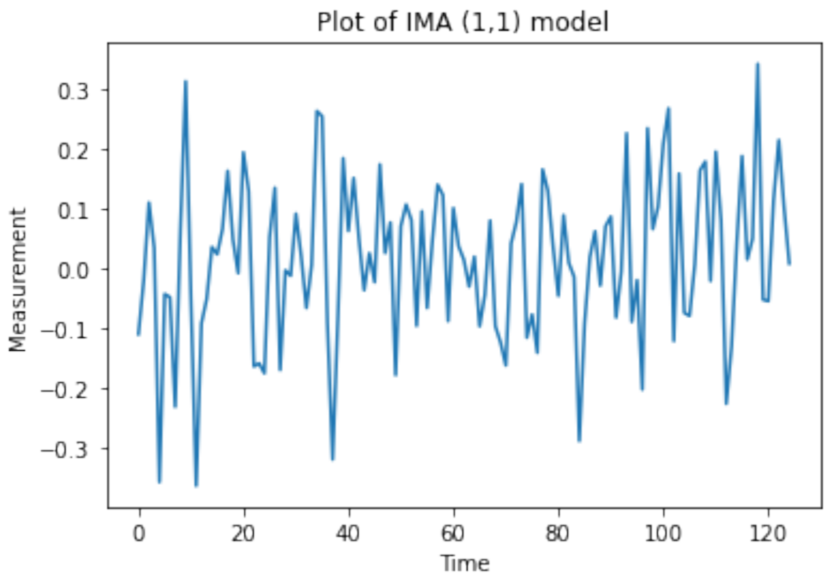
	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-0.6640	0.072	-9.243	0.000	-0.805	-0.523
sigma2	0.0180	0.002	8.182	0.000	0.014	0.022

Ljung-Box (L1) (Q):	2.36	**Jarque-Bera (JB):**	2.20
Prob(Q):	0.12	**Prob(JB):**	0.33
Heteroskedasticity (H):	0.87	**Skew:**	-0.29
Prob(H) (two-sided):	0.65	**Kurtosis:**	3.32

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

In [80]:
```
model_res11 = df(measurement_model_fit11.resid)
plt.plot (model_res11)
plt.xlabel ('Time')
plt.ylabel ('Measurement')
plt.title ('Plot of IMA (1,1) model')
```

Out[80]: Text(0.5, 1.0, 'Plot of IMA (1,1) model')



In [103…
```
measurement_model_fit11.sse
```

Out[103… 2.246084052047077

In [81]:
```
measurement_model22 = ARIMA(y, order=(0,1,2))
measurement_model_fit22 = measurement_model22.fit()
measurement_model_fit22.summary()
```

Out[81]:

SARIMAX Results

Dep. Variable:	Anomaly, C	**No. Observations:**	125
Model:	ARIMA(0, 1, 2)	**Log Likelihood**	76.508
Date:	Sun, 03 Oct 2021	**AIC**	-147.016
Time:	01:09:30	**BIC**	-138.555
Sample:	0	**HQIC**	-143.579
	- 125		
Covariance Type:	opg		

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
ma.L1	-0.4676	0.099	-4.736	0.000	-0.661	-0.274
ma.L2	-0.2296	0.090	-2.565	0.010	-0.405	-0.054
sigma2	0.0170	0.002	7.631	0.000	0.013	0.021

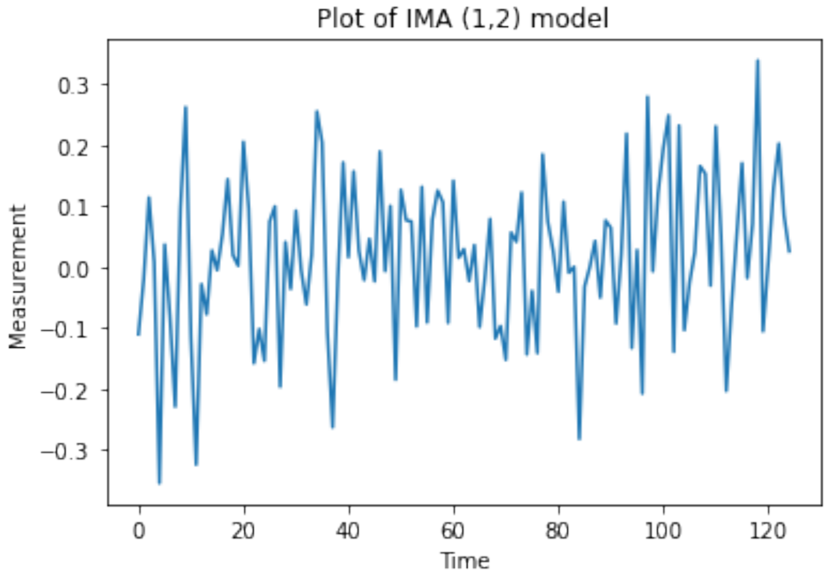
Ljung-Box (L1) (Q):	0.13	**Jarque-Bera (JB):**	1.18
Prob(Q):	0.72	**Prob(JB):**	0.55
Heteroskedasticity (H):	1.04	**Skew:**	-0.23
Prob(H) (two-sided):	0.91	**Kurtosis:**	3.12

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

In [82]:

```
model_res22 = df(measurement_model_fit22.resid)
plt.plot (model_res22)
plt.xlabel ('Time')
plt.ylabel ('Measurement')
plt.title ('Plot of IMA (1,2) model')
```

Out[82]: Text(0.5, 1.0, 'Plot of IMA (1,2) model')



In [83]:

```
measurement_model_fit22.sse
```

Out[83]: 2.124140606109833

Alternative approach for a and b

In [84]:

```
model2 = ARIMA(data['Anomaly, C'], order=(0,1,1))
model2_fit= model2.fit()
model2_fit.summary()
```

Out[84]:

SARIMAX Results

Dep. Variable:	Anomaly, C	**No. Observations:**	125
Model:	ARIMA(0, 1, 1)	**Log Likelihood**	72.910
Date:	Sun, 03 Oct 2021	**AIC**	-141.821
Time:	01:09:35	**BIC**	-136.180
Sample:	0	**HQIC**	-139.530
	- 125		
Covariance Type:	opg		

	coef	**std err**	**z**	**P>|z|**	**[0.025**	**0.975]**
ma.L1	-0.6640	0.072	-9.243	0.000	-0.805	-0.523
sigma2	0.0180	0.002	8.182	0.000	0.014	0.022

Ljung-Box (L1) (Q):	2.36	**Jarque-Bera (JB):**	2.20
Prob(Q):	0.12	**Prob(JB):**	0.33
Heteroskedasticity (H):	0.87	**Skew:**	-0.29
Prob(H) (two-sided):	0.65	**Kurtosis:**	3.32

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

In [85]:

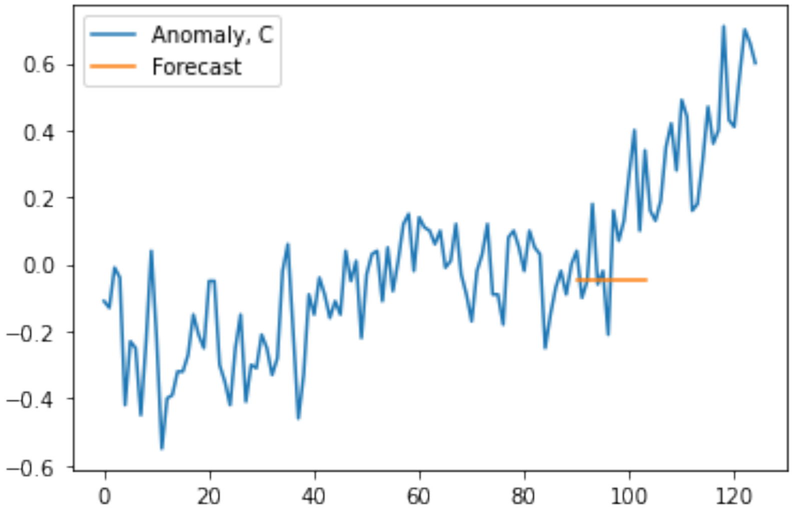
```
model2_fit.sse
```

Out[85]: 2.246084052047077

In [86]:

```
data["Forecast"]= model2_fit.predict(start=90,end=103,dynamic=True)
data[["Anomaly, C", 'Forecast']].plot()
```

Out[86]: <AxesSubplot:>



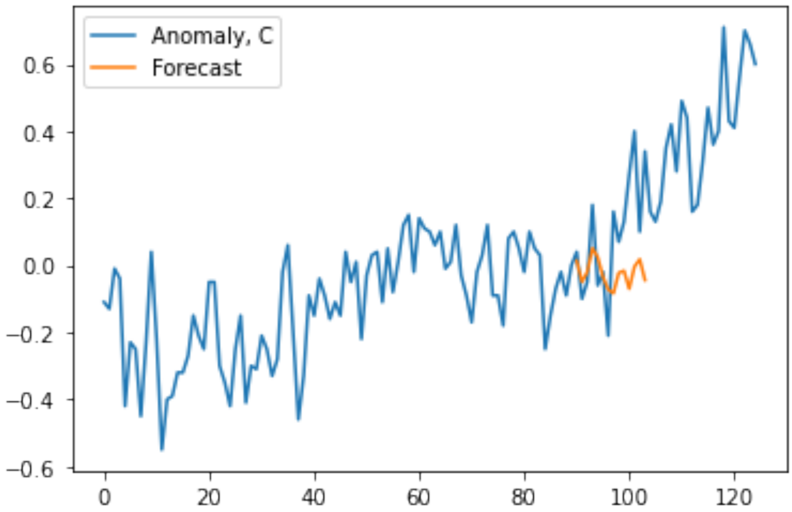
In [87]:
```
model3=sm.tsa.statespace.SARIMAX(data["Anomaly, C"], order=(1,1,1), seasona
```

In [88]:
```
results=model3.fit()
```

```
/opt/anaconda3/lib/python3.8/site-packages/statsmodels/tsa/statespace/sarim
ax.py:978: UserWarning: Non-invertible starting MA parameters found. Using
zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
/opt/anaconda3/lib/python3.8/site-packages/statsmodels/base/model.py:566: C
onvergenceWarning: Maximum Likelihood optimization failed to converge. Chec
k mle_retvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

In [89]:
```
data["Forecast"]= results.predict(start=90,end=103,dynamic=True)
data[["Anomaly, C", 'Forecast']].plot()
```

Out[89]: <AxesSubplot:>



b)

In [90]:
```
model4= ARIMA(data["Anomaly, C"], order=(0,1,2))
model4_fit=model4.fit()
model4_fit.summary()
```

Out[90]:

SARIMAX Results

Dep. Variable:	Anomaly, C	**No. Observations:**	125
Model:	ARIMA(0, 1, 2)	**Log Likelihood**	76.508
Date:	Sun, 03 Oct 2021	**AIC**	-147.016
Time:	01:09:38	**BIC**	-138.555
Sample:	0	**HQIC**	-143.579
	- 125		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ma.L1	-0.4676	0.099	-4.736	0.000	-0.661	-0.274
ma.L2	-0.2296	0.090	-2.565	0.010	-0.405	-0.054
sigma2	0.0170	0.002	7.631	0.000	0.013	0.021

Ljung-Box (L1) (Q):	0.13	**Jarque-Bera (JB):**	1.18
Prob(Q):	0.72	**Prob(JB):**	0.55
Heteroskedasticity (H):	1.04	**Skew:**	-0.23
Prob(H) (two-sided):	0.91	**Kurtosis:**	3.12

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).

In [91]:
```
model4_fit.sse
```

Out[91]:
```
2.124140606109833
```

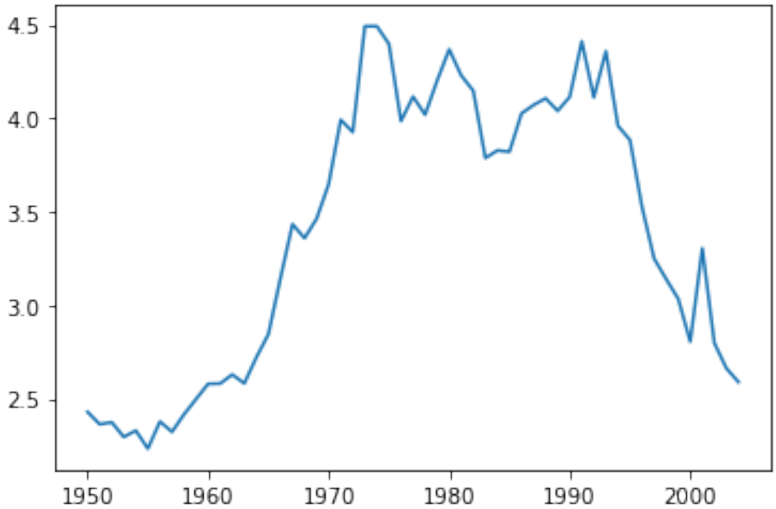
c) The SSE value of second model is lower which is why it is better.

QUESTION 3:

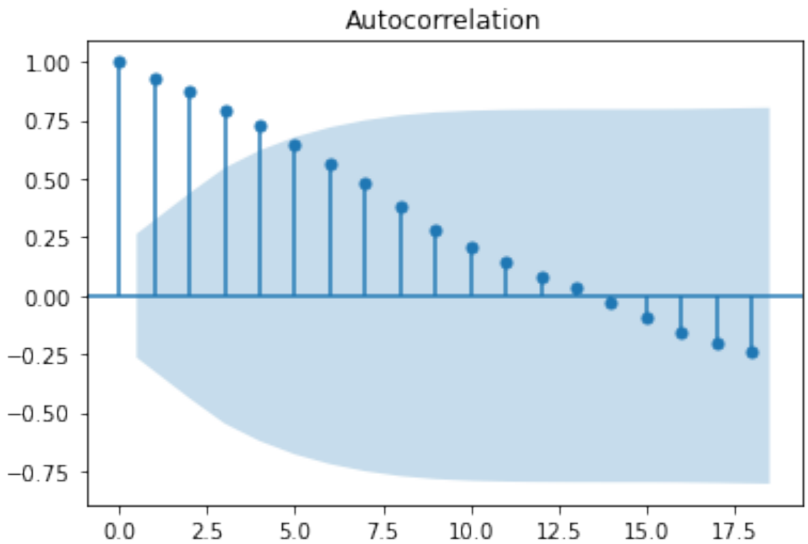
In [92]:
```
d= pd.read_csv("Measurement_Q3.csv")
```

In [93]:
```
a=d["Year"]
b=d["Measurement"]
plt.plot(a,b)
```

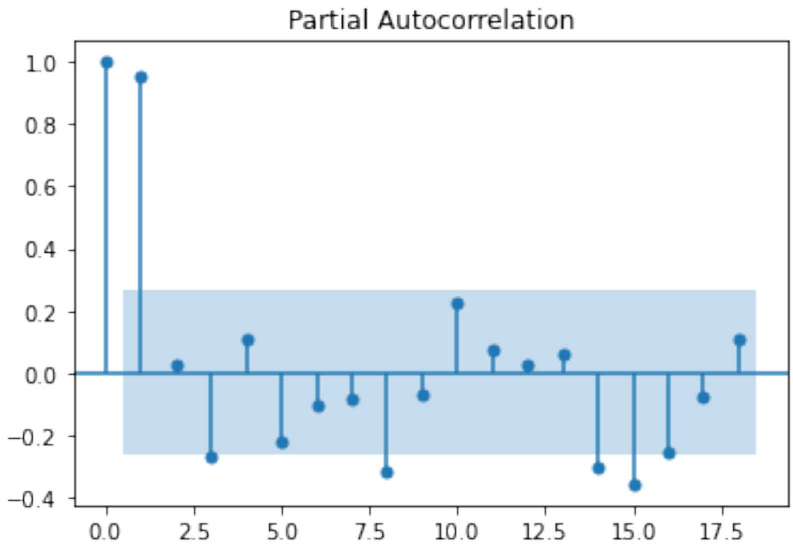
Out[93]: [<matplotlib.lines.Line2D at 0x1273ef850>]



In [94]:
```
figure= plot_acf(b)
```

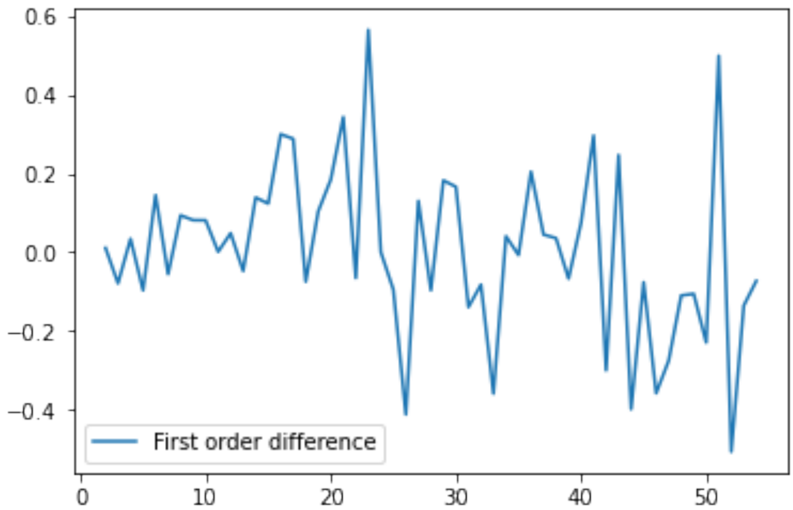


In [95]:
```
figure2= plot_pacf(b)
```



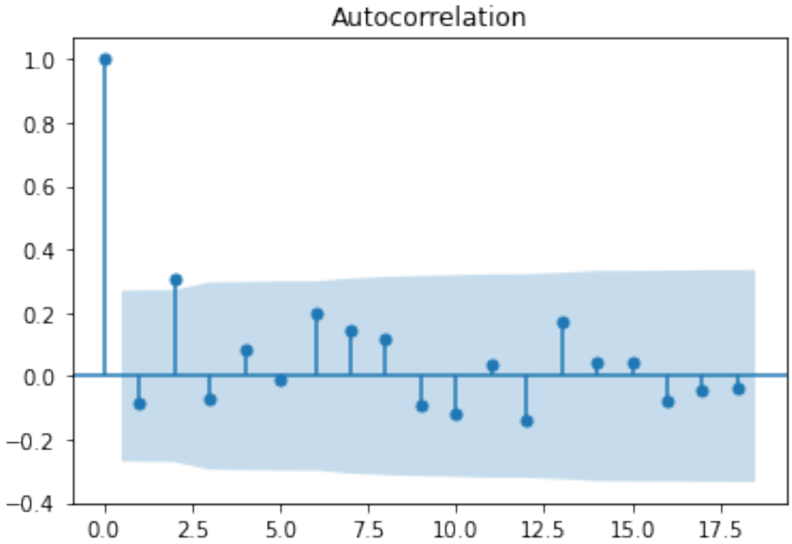
b)

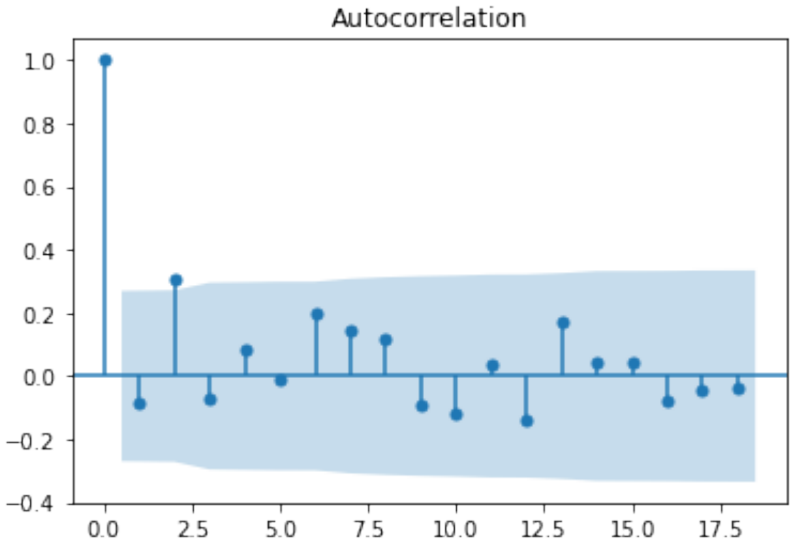
In [97]:
```
d['diff']= d['Measurement'].diff()
d=d.dropna()
plt.plot(d['diff'], label='First order difference')
plt.legend();
```



In [101…
```
plot_acf(d["diff"])
```

Out[101…

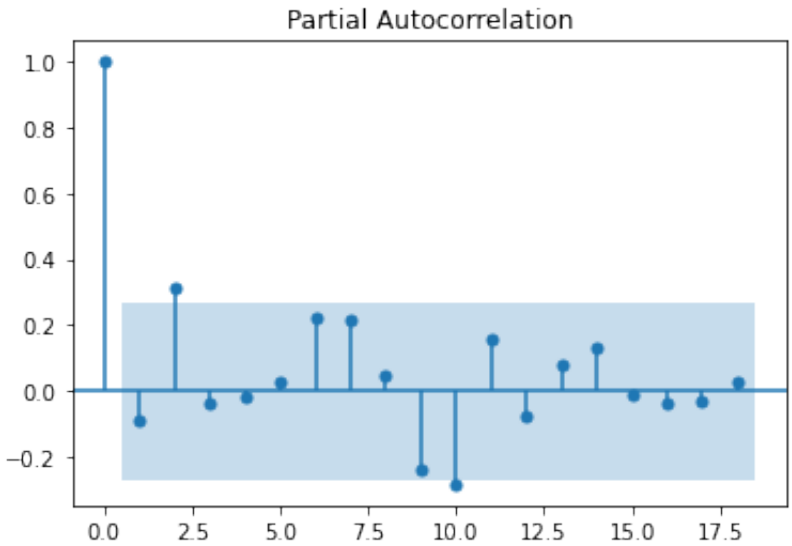


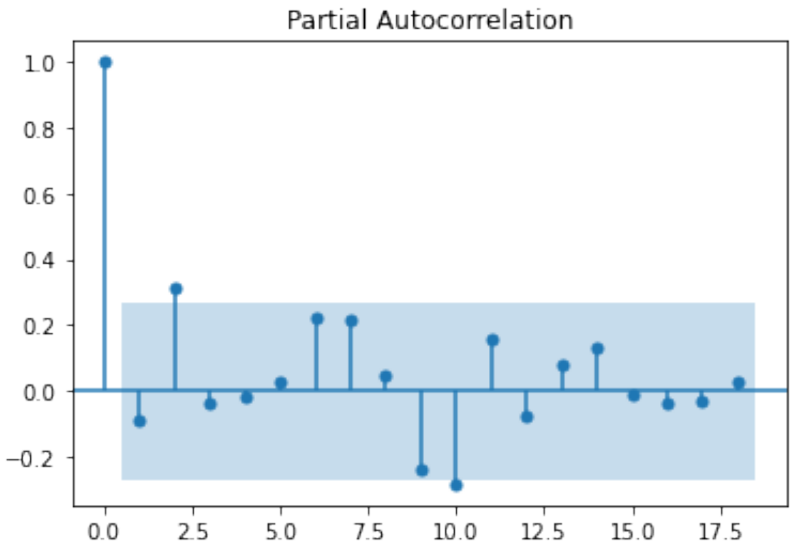


In [102…

```
plot_pacf(d["diff"])
```

Out[102…





c) ACF is significant only for 1st lag. And PACF goes down after 1st lag. So it is an AR(1) model. ARMA(p,d,q)= ARIMA(0,1,1)

In [ ]: