In [1]:
```python
#importing packages
import numpy as np
import pandas as pd
import yfinance as yf
import scipy.optimize as spop

# Plotting library
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
plt.style.use("seaborn-darkgrid")
matplotlib.rcParams['figure.figsize']=[8,4]
```

In [2]:
```python
#importing data
tickers = ['CVX', 'GS', 'ORCL', 'PEP', 'PFE']
start = '2015-12-31'
end = '2020-12-31'
prices_df = yf.download(' '.join(tickers), start, end)['Adj Close']
returns_df = prices_df.pct_change()[1:]
```

```
[*********************100%***********************]  5 of 5 completed
```

In [3]:
```python
#calculating the return vector and the covariance matrix
cov = returns_df.cov()*252
r = np.array(((1+returns_df).prod())**(252/len(returns_df)) - 1)
e = np.ones(len(r))
rets = returns_df.values
mu = rets.mean(0)
sigma = rets.std(0)
```

In [4]:
```python
#defining the investable universe
icov = np.linalg.inv(cov)
h = np.matmul(e, icov)
g = np.matmul(r, icov)
a = np.sum(e*h)
b = np.sum(e*g)
c = np.sum(r*g)
d = a*c - b**2
```

In [5]:
```python
#minimum variance and tangency portfolio
mvp = h/a
mvp_return = b/a
mvp_risk = 1/a**(1/2)
tangency = g/b
tangency_return = c/b
tangency_risk = c**(1/2)/b
```

In [6]:
```python
#plotting the efficient portfolio frontier
exp_returns = np.arange(0.05, 0.2001, 0.001)
risk = ((a*exp_returns**2 - 2*b*exp_returns + c)/d)**(1/2)
plt.plot(mvp_risk, mvp_return, 'o', label='GVMP', color='orange')
plt.plot(risk, exp_returns, '-', label='Minimum Variance Frontier', color='blue')
plt.plot(sigma*np.sqrt(252), mu*252, 'x', label='Individual Assets', color='red')
plt.xlabel('Annualised Std Dev')
plt.ylabel('Annualised Expected Return')
plt.title('The Gains from Diversification - Illustrative Example')
plt.legend()
plt.show()
```