Compare two vowel based features

based on the paper *Speech Rhythm Variation in Early-Stage Parkinson's Disease: A Study on Different Speaking Tasks*,

compute vtv and pv for speech samples (using the allosaurus package)

requires a file called *ground_truth.csv* with audformat groundturh for wav files based in *HC_speakers*

note: all original audio files had to be converted

Here's the code for VtV and pV calculation

```
# remove all rows where cv is not V or C
phonemes = phonemes[phonemes['cv'].isin(['V', 'C'])]

# Filter for vowels
vowels_df = phonemes[phonemes['cv'] == 'V']
# Calculate vtv (time between vowels)
#  VtV, the mean interval between two consecutive vowel onset points
vtv = 0.0
startCount = False
for i in range(len(phonemes) - 1):
    if startCount:
        vtv += phonemes.iloc[i]['dur']
    if phonemes.iloc[i]['cv'] == 'V':
        vtv += phonemes.iloc[i]['dur']
        startCount = True
# now count backwards to find the last vowel
rem_vtv = 0.0
for i in range(len(phonemes) - 1, -1, -1):
    if phonemes.iloc[i]['cv'] == 'V':
        rem_vtv += phonemes.iloc[i]['dur']
        break
    else:
        rem_vtv += phonemes.iloc[i]['dur']
vtv -= rem_vtv
# Average time between vowel onset times
if len(vowels_df) > 1:
    vtv /= (len(vowels_df) - 1)

# Calculate pv (percentage of vowel duration)
total_vowel_duration = vowels_df['dur'].sum()
total_duration = phonemes['dur'].sum()
pv = (total_vowel_duration / total_duration) * \
    100 if total_duration > 0 else 0.0

return vtv, pv
```