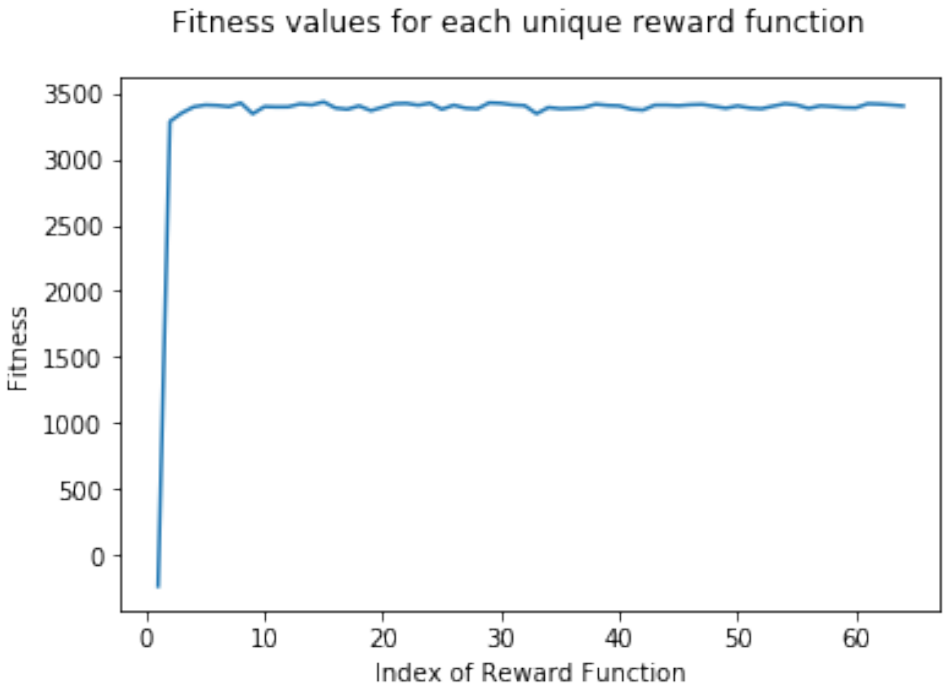
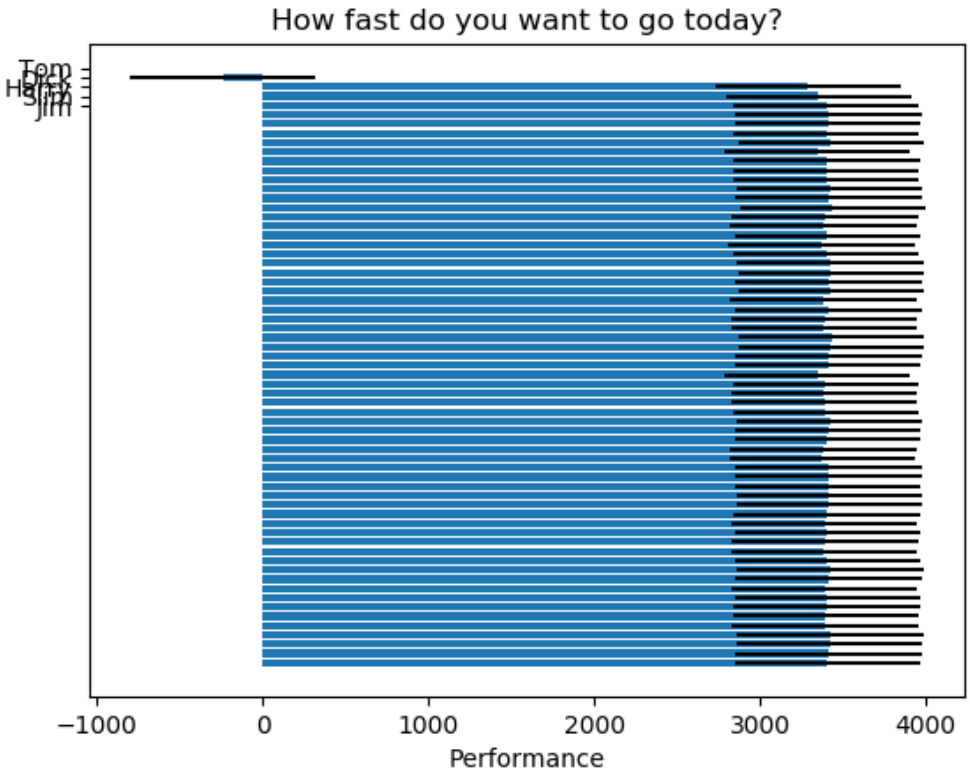
[19]:
```
fig, axis = plt.subplots()

axis.plot(np.arange(1,65),fitness_vals_thousand)

fig.suptitle("Fitness values for each unique reward function", fontsize=12)
axis.set_xlabel('Index of Reward Function', fontsize=10)
axis.set_ylabel('Fitness', fontsize=10)

mean_center_init = np.mean(fitness_vals_thousand)
variance_center_init = np.var(fitness_vals_thousand)
cov_center_init = (variance_center_init/mean_center_init)*100

print("Mean Fitness over all reward functions = ", mean_center_init)
print("Variance of Fitness over all reward functions = ", variance_center_init)
print("Coefficient of Variation over all reward functions = ", cov_center_init)

mean_center_init = np.mean(fitness_vals_thousand[1:])
variance_center_init = np.var(fitness_vals_thousand[1:])
cov_center_init = (variance_center_init/mean_center_init)*100

print("Mean Fitness over all reward functions except case 1 with all rewards set␣
 ↪0 = ", mean_center_init)
print("Variance of Fitness over all reward functions except case 1 with all␣
 ↪rewards set 0 = ", variance_center_init)
print("Coefficient of variation over all reward functions except case 1 with all␣
 ↪rewards set 0 = ", cov_center_init)
```

```
Mean Fitness over all reward functions =  3343.0987187503274
Variance of Fitness over all reward functions =  204300.80709689224
Coefficient of Variation over all reward functions =  6111.120977404497
Mean Fitness over all reward functions except case 1 with all rewards set 0 =
3399.9680476193794
Variance of Fitness over all reward functions except case 1 with all rewards set
0 =  559.9608257597987
Coefficient of variation over all reward functions except case 1 with all
rewards set 0 =  16.469590829004325
```



[23]:
```
# Fixing random state for reproducibility
plt.rcdefaults()
fig, ax = plt.subplots()

# Example data
reward_functions = []
error = variance_center_init = np.var(fitness_vals_thousand[1:])


ax.barh(np.arange(1,65), fitness_vals_thousand, xerr=error, align='center')
ax.set_yticks(65)
ax.invert_yaxis()  # labels read top-to-bottom
ax.set_ylabel('Reward Function')
ax.set_xlabel('Fitness')

plt.show()
```



[24]:
```
max_fitness = np.max(fitness_vals_thousand)
max_index = np.where(fitness_vals_thousand == max_fitness)

max_x = x_vals_thousand[max_index[0][0]]
max_y = y_vals_thousand[max_index[0][0]]
max_z = z_vals_thousand[max_index[0][0]]
max_q = q_vals_thousand[max_index[0][0]]
max_w = w_vals_thousand[max_index[0][0]]
max_e = e_vals_thousand[max_index[0][0]]

#for i in range(len(max_index)):
    #max_y_e_tuples.append((y_vals_thousand[max_index[i]],␣
 ↪e_vals_thousand[max_index[i]]))


print(max_fitness)
print(type(max_index[0][0]))
print("Max x = ", max_x, "Max y = ", max_y, "Max z = ", max_z, "Max q = ",␣
 ↪max_q, "Max w = ", max_w, "Max e = ", max_e)
#print(max_y_e_tuples)
```

```
3436.784000000338
<class 'numpy.int64'>
Max x =  0.0 Max y =  0.0 Max z =  1.0 Max q =  1.0 Max w =  1.0 Max e =  0.0
```

[ ]: