Solution 1:

The code for this programming exercise can be found in the `hw_shapley_py_sol.ipynb` or `hw_shapley_vanilla_py_sol.ipynb` files for Python and in the `hw_shapley_R_sol.Rmd` or `hw_shapley_vanilla_R_sol.Rmd` files for R.

a)

$$\text{payoff}(S) = 10t + 10m + 10s + 2j + 20(t \wedge m) + 20(t \wedge m \wedge s) - 30((t \vee m \vee s) \wedge j)$$

$$\text{payoff}(\{\texttt{t,m}\}) = 10 + 10 + 20 = 40$$

$$\text{payoff}(\{\texttt{t,j,s}\}) = 10 + 10 + 2 - 30 = -8$$

b) Pseudocode of `payoff(coalition)`:

Algorithm 1 `payoff()`

Require: `coalition`: Coalition vector (`set` or `list` or ...)
```
t ← boolean if 't' is in coalition
s ← boolean if 's' is in coalition
m ← boolean if 'm' is in coalition
j ← boolean if 'j' is in coalition
l ← boolean if 'l' is in coalition
return 10 * t + 10 * m + 2 * j + 20 * (t and m) + 20 * (t and m and s) - 30 * ((t or m or s) and j)
```

$$\text{payoff}(\{\emptyset\}) = 0$$

$$\text{payoff}(\{\texttt{t,m,s,j,l}\}) = 10 + 10 + 10 + 2 + 20 + 20 - 30 = 42$$

Concerning `all_unique_subsets(population)`, both R and Python provide built-in functions that return the power set, i.e. the set of all subsets of a given set. Alternatively, one possible pseudocode of `all_unique_subsets(population)` would be:

Algorithm 2 `all_unique_subsets()`

Require: `population`: Vector / list / set containing all available players
```
if population = ∅ then subsets ← ∅
else if population ≠ ∅ then
    first_member ← pick any element from population
    population_wo_member ← population \ first_member
    subsets_wo_member ← all_unique_subsets(population_wo_member)
    subsets_w_member ← list of all sets in subsets_wo_member each with first_member added to it
    subsets ← Union(subsets_w_member, subsets_wo_member)
end if
return subsets
```

Pseudocode of shapley_set(member, population, v_function):

Algorithm 3 shapley_set()

```
Require: member: individual player, or feature of interest
Require: population: vector containing all available players
Require: v_function: Some value function
remainder ← everyone from the population but member (or: population \ member)
all_sets_wo_member ← all_unique_subsets(remainder)
F ← length of population
result ← 0
for coalition in all_sets_wo_member do
    S ← length of coalition
    diff ← v_function(Union(coalition, member)) - v_function(coalition)
    factor ← S! * (F - S - 1)!  / F!
    result ← result + factor * diff
end for
return result
```

c) A function to generate all permutations can be implemented similar to the function all_unique_subsets(), also by first generating the first element and then calling the function on the set of remaining elements.

Pseudocode of shapley_perm(member, population, v_function):

Algorithm 4 shapley_perm()

```
Require: member: individual player, or feature of interest
Require: population: vector containing all available players
Require: v_function: Some value function
all_perms ← All permutations of population
F ← length of population
result ← 0
for perm in all_perms do
    member_ix ← index of member in perm
    coalition ← coalition of perm before member_ix
    diff ← v_function(Union(coalition, member)) - v_function(coalition)
    result ← result + diff
end for
return result / F!
```

d) Pseudocode of shapley_perm_approx(member, population, v_function, num_iter):

Algorithm 5 shapley_perm_approx()

```
Require: member: individual player, or feature of interest
Require: population: vector containing all available players
Require: v_function: value function
Require: num_iter: number of iterations
vals ← Empty vector or list
for i in num_iter do
    perm ← draw a random permutation of population
    member_ix ← index of member in perm
    coalition ← coalition of perm before member_ix
    vals[i] ← v_function(Union(coalition, member)) - v_function(coalition)
end for
return average of vals
```

e) (i) Pseudocode of symmetry_check():

Algorithm 6 `symmetry_check()`

```
Require: j: first feature index
Require: k: second feature index
Require: population: vector containing all available players
Require: v_function: value function
Require: shapley_func: function for computing Shapley values
remainder ← everyone from the population but j, k
all_S ← all_unique_subsets(remainder)
for S in all_S do
    surplus_j ← v_function(Union(coalition, j)) - v_function(coalition)
    surplus_k ← v_function(Union(coalition, k)) - v_function(coalition)
    save surplus_j and surplus_k in vectors surpluss_j and surpluss_k, respectively, for every iteration
end for
if surpluss_j equal surpluss_k then
    print equal surplus
    val_j ← shapley_func(j, population, v_function)
    val_k ← shapley_func(k, population, v_function)
    return val_j == val_k
end if
return TRUE
```

(ii) Pseudocode of `dummy_check()`:

Algorithm 7 `dummy_check()`

```
Require: j: feature index
Require: population: vector containing all available players
Require: v_function: value function
Require: shapley_func: function for computing Shapley values
remainder ← everyone from the population but j
all_S ← all_unique_subsets(remainder)
for S in all_S do
    surplus_j ← difference of v_function of S with j minus v_function of S
    save surplus_j in vector surpluss_j for every iteration
end for
if sum of |surpluss_j| == 0 then
    print has contribution
    val_j ← shapley_func(j, population, v_function)
    return val_j == 0
end if
return TRUE
```

(iii) Pseudocode of `additivity_check()`:

Algorithm 8 `additivity_check()`

```
Require: population: vector containing all available players
Require: v_function1: value function 1
Require: v_function2: value function 2
Require: shapley_func: function for computing Shapley values
combined ← addition of v_function1 and v_function2
vals1 ← Shapley values for all features using v_function1
vals2 ← Shapley values for all features using v_function2
vals_comb ← Shapley values for all features using combined
vals_additive ← vals1 + vals2
return vals_comb == vals_additive
```

(iv) Pseudocode of `efficiency_check()`:

Algorithm 9 `efficiency_check()`

Require: `population`: vector containing all available players
Require: `v_function`: value function
Require: `shapley_func`: function for computing Shapley values
`payoff_total` ← `v_function` of `population`
`shapley_vals` ← Shapley values for all features using `v_function`
`total_shapley_vals` ← sum of `shapley_vals`
return `payoff_total` == `total_shapley_vals`

Solution 2:

Setup. Let $P = \{1, \ldots, p\}$ be the set of $p := |P|$ players, v the value function and ϕ_j the Shapley value of player j.

Recall the two equivalent definitions of Shapley values:

Permutation-Based Definition of Shapley Values. Let $\mathrm{Pred}_\pi(j)$ be the set of players that appear before player j in the permutation π. Then for any j, the Shapley value of player j is

$$\phi_j \;=\; \frac{1}{p!} \sum_{\pi \in \mathfrak{S}_P} \Big(v\big(\mathrm{Pred}_\pi(j) \cup \{\, j\}\big) \;-\; v\big(\mathrm{Pred}_\pi(j)\big)\Big),$$

where $\mathfrak{S}_P$ is the set of all $p!$ permutations of P. In words, we look at the "marginal contribution" of j each time it arrives in a permutation (depending on whichever players arrived before it), and then average over all permutations.

Set-Based Definition of Shapley Values. For any j, we consider all subsets S of P which do not contain j, then the Shapley value of player j is

$$\phi_j \;=\; \sum_{S \subseteq P \setminus \{j\}} \frac{|S|!\,\big(p - |S| - 1\big)!}{p!} \Big(v(S \cup \{j\}) - v(S)\Big).$$

In essence, both definitions are almost the same, just that those terms from the first definition that yield the same subsets S (although from different permutations), are summed up and counted in the second definition, so that the second definition has less terms in the sum.

(a) **Proof of the first 3 Axioms:** The Dummy and Additivity properties are relatively easy to proof.

Theorem 1 (Null Player (Dummy))**.** *Let $\{\phi_j\}_{j \in P}$ be the Shapley values induced by v. For any player $j \in P$ it holds:*

$$\textit{If } v(S \cup \{j\}) = v(S) \textit{ for all } S \subseteq P \setminus \{j\}, \textit{ then } \phi_j = 0.$$

Proof. Assume $p \geq 1$, $j \in P$ and $v(S \cup \{j\}) = v(S)$ for all subsets $S \subseteq P \setminus \{j\}$. We need to prove that $\phi_j = 0$.

This follows directly from either of the two definitions: The assumption states that the difference $v(S \cup \{j\}) - v(S)$ inside the sum is 0, for every $S \subseteq P \setminus \{j\}$, in other words for every summand. Therefore, also the whole sum is 0, so $\phi_j = 0$.

□

Theorem 2 (Additivity)**.** *Let $v, v_1, v_2 : 2^P \to \mathbb{R}$ be value functions with $v = v_1 + v_2$, which means that for every subset $S \subseteq P$ we have $v(S) = v_1(S) + v_2(S)$. Let $\{\phi_{j,v}\}_{j \in P}$ be the Shapley values induced by v and $\{\phi_{j,v_1}\}_{j \in P}$ and $\{\phi_{j,v_2}\}_{j \in P}$ those for v_1 and v_2 respectively. Then for any player $j \in P$ it holds:*

$$\phi_{j,v} = \phi_{j,v_1+v_2} = \phi_{j,v_1} + \phi_{j,v_2}.$$

Proof. Assume $p \geq 1$, $j \in P$ and that v_1, v_2 are two value functions. We need to prove that $\phi_{j,v_1+v_2} = \phi_{j,v_1} + \phi_{j,v_2}$.

For the marginal contributions of player j on any subset $S \subseteq P$ we have: $v(S \cup \{j\}) - v(S) = v_1(S \cup \{j\}) - v_1(S) + v_2(S \cup \{j\}) - v_2(S)$. Plugging this into either of the two definitions yields the result:

$$\begin{aligned}
\phi_{j,v} &= \phi_{j,v_1+v_2} = \sum_{S \subseteq P \setminus \{j\}} \frac{|S|!\,\big(p-|S|-1\big)!}{p!} \Big(v(S \cup \{j\}) - v(S)\Big) \\
&= \sum_{S \subseteq P \setminus \{j,k\}} \frac{|S|!\,\big(p-|S|-1\big)!}{p!} \Big(v_1(S \cup \{j\}) - v_1(S) + v_2(S \cup \{j\}) - v_2(S)\Big) \\
&= \sum_{S \subseteq P \setminus \{j,k\}} \frac{|S|!\,\big(p-|S|-1\big)!}{p!} \Big(v_1(S \cup \{j\}) - v_1(S)\Big) + \sum_{S \subseteq P \setminus \{j,k\}} \frac{|S|!\,\big(p-|S|-1\big)!}{p!} \Big(v_2(S \cup \{j\}) - v_2(S)\Big) \\
&= \phi_{j,v_1} + \phi_{j,v_2}.
\end{aligned}$$

□

The symmetry axiom requires a little more calculation.

Theorem 3 (Symmetry). *Let $\{\phi_j\}_{j \in P}$ be the Shapley values induced by v. For any two players $j, k \in P$ it holds:*

$$\textit{If } v(S \cup \{j\}) = v(S \cup \{k\}) \textit{ for all } S \subseteq P \setminus \{j,k\}, \textit{ then } \phi_j = \phi_k.$$

Proof. Assume $p \geq 2$ and $j, k \in P$ and $v(S \cup \{j\}) = v(S \cup \{k\})$ for all subsets $S \subseteq P \setminus \{j,k\}$. We need to prove that $\phi_j = \phi_k$.

We will use the set-based definition of Shapley values, and will show the claim by straightforward calculation and rearranging the sum. We first split the sum, which is over all sets $S \subseteq P \setminus \{j\}$, into two parts, depending on whether player k is in S or not:

$$\begin{aligned}
\phi_j &= \sum_{S \subseteq P \setminus \{j\}} \frac{|S|!\,\big(p-|S|-1\big)!}{p!} \Big(v(S \cup \{j\}) - v(S)\Big) \\
&= \sum_{S \subseteq P \setminus \{j,k\}} \frac{|S|!\,\big(p-|S|-1\big)!}{p!} \Big(v(S \cup \{j\}) - v(S)\Big) + \sum_{\substack{S \subseteq P \setminus \{j\},\\ k \in S}} \frac{|S|!\,\big(p-|S|-1\big)!}{p!} \Big(v(S \cup \{j\}) - v(S)\Big).
\end{aligned}$$

Now, in the first part of the sum we can directly use our assumption:

$$\sum_{S \subseteq P \setminus \{j,k\}} \frac{|S|!\,\big(p-|S|-1\big)!}{p!} \Big(v(S \cup \{j\}) - v(S)\Big) = \sum_{S \subseteq P \setminus \{j,k\}} \frac{|S|!\,\big(p-|S|-1\big)!}{p!} \Big(v(S \cup \{k\}) - v(S)\Big)$$

For the second part, first note that both sums have the same number of summands, since there is a one-to-one correspondence (a bijection) between the subsets of $P \setminus \{j,k\}$ and those subsets of $P \setminus \{j\}$ that contain k, by either adding k to a given subset $S \subseteq P \setminus \{j,k\}$ or not. We can use this bijection to define an "index transformation" for the second sum: We replace the set S, which we know must contain k, with $S = \tilde{S} \cup \{k\}$, in other words we define $\tilde{S} := S \setminus \{k\}$. We can then use our assumption on the set $\tilde{S}$ as well. This yields:

$$\begin{aligned}
&\sum_{\substack{S \subseteq P \setminus \{j\},\\ k \in S}} \frac{|S|!\,\big(p-|S|-1\big)!}{p!} \Big(v(S \cup \{j\}) - v(S)\Big) \\
&= \sum_{\tilde{S} \subseteq P \setminus \{j,k\}} \frac{(|\tilde{S}|+1)!\,\big(p-(|\tilde{S}|+1)-1\big)!}{p!} \Big(v(\tilde{S} \cup \{j,k\}) - v(\tilde{S} \cup \{k\})\Big) \\
&= \sum_{\tilde{S} \subseteq P \setminus \{j,k\}} \frac{(|\tilde{S}|+1)!\,\big(p-|\tilde{S}|-2\big)!}{p!} \Big(v(\tilde{S} \cup \{j,k\}) - v(\tilde{S} \cup \{j\})\Big) \\
&= \sum_{\substack{S \subseteq P \setminus \{k\},\\ j \in S}} \frac{|S|!\,\big(p-|S|-1\big)!}{p!} \Big(v(S \cup \{k\}) - v(S)\Big).
\end{aligned}$$

In the last step, we used this transformation backwards, so we add / remove j instead of k this time, in other words $S = \tilde{S} \cup \{j\}$ and $\tilde{S} := S \setminus \{j\}$. This is possible since $\tilde{S}$ contains neither j nor k.

We now use the first step from the beginning backwards and arrive at the desired result:

$$\begin{aligned}\phi_j \;&=\; \sum_{S \subseteq P\setminus\{j,k\}} \frac{|S|!\,\big(p-|S|-1\big)!}{p!} \Big(v(S\cup\{k\}) - v(S)\Big) \;+\; \sum_{\substack{S \subseteq P\setminus\{k\},\\ j\in S}} \frac{|S|!\,\big(p-|S|-1\big)!}{p!} \Big(v(S\cup\{k\}) - v(S)\Big)\\ &=\; \sum_{S \subseteq P\setminus\{k\}} \frac{|S|!\,\big(p-|S|-1\big)!}{p!} \Big(v(S\cup\{k\}) - v(S)\Big) \;=\; \phi_k.\end{aligned}$$

□

(b) **Bonus: Proof of the Efficiency Axiom**

Efficiency requires a little more effort than the others:

Theorem 4 (Efficiency)**.** *Let $\{\phi_j\}_{j\in P}$ be the Shapley values induced by v. Then*

$$\sum_{j=1}^{p} \phi_j \;=\; v(P).$$

The proof idea is as follows: Because we sum up the Shapley values over all players, the values $v(S)$ of each coalition not equal to P, so $S \subsetneq P$, appear with both minus and plus signs that exactly cancel each other out. Apart from that, the term $v(P)$ occurs for every player, and one only needs to calculate that its weights sum up to 1.

We will use the order-based definition of Shapley values here, since with it the proof is simpler.

Proof. We first plug in the definition and swap the two sums:

$$\begin{aligned}\sum_{j=1}^{p} \phi_j \;&=\; \sum_{j=1}^{p} \frac{1}{p!} \sum_{\pi\in\mathfrak{S}_P} \Big(v\big(\mathrm{Pred}_\pi(j)\cup\{j\}\big) - v\big(\mathrm{Pred}_\pi(j)\big)\Big)\\ &= \frac{1}{p!} \sum_{\pi\in\mathfrak{S}_P} \sum_{j=1}^{p} \Big(v\big(\mathrm{Pred}_\pi(j)\cup\{j\}\big) - v\big(\mathrm{Pred}_\pi(j)\big)\Big).\end{aligned}$$

Now the inner sum corresponds to a single permutation, and sums over all players in this permutation.

Fix a particular permutation π. We consider the order of the players given by this permutation: $\big(\pi(1), \pi(2), \ldots, \pi(p)\big)$. Each marginal contribution inside the inner sum corresponds to one step in this ordering. We reorder the inner sum according to this ordering given by the permutation:

$$\begin{aligned}&\sum_{j=1}^{p}\Big(v\big(\mathrm{Pred}_\pi(j)\cup\{j\}\big) - v\big(\mathrm{Pred}_\pi(j)\big)\Big) = \sum_{i=1}^{p}\Big(v\big(\{\pi(1),\ldots\pi(i)\}\big) - v\big(\{\pi(1),\ldots\pi(i-1)\}\big)\Big)\\ &\quad = \underbrace{v\big(\{\pi(1)\}\big) - v(\varnothing)}_{i=1 \text{ or } j=\pi(1)} \;+\; \underbrace{v\big(\{\pi(1),\pi(2)\}\big) - v\big(\{\pi(1)\}\big)}_{i=2 \text{ or } j=\pi(2)} \;+\; \underbrace{v\big(\{\pi(1),\pi(2),\pi(3)\}\big) - v\big(\{\pi(1),\pi(2)\}\big)}_{i=3 \text{ or } j=\pi(3)}\\ &\qquad +\;\ldots\;+\; \underbrace{v\big(\{\pi(1),\ldots,\pi(p)\}\big) - v\big(\{\pi(1),\ldots,\pi(p-1)\}\big)}_{i=p \text{ or } j=\pi(p)} = v\big(\pi(1),\ldots,\pi(p)\}\big) - v(\varnothing) = v(P).\end{aligned}$$

In other words, the inner sum is a telescope sum, where all the terms except the first and the last one cancel. Plugging this into the outer sum yields the desired result:

$$\begin{aligned}\sum_{j=1}^{p} \phi_j \;&=\; \frac{1}{p!} \sum_{\pi\in\mathfrak{S}_P} \left(\sum_{j=1}^{p}\Big(v\big(\mathrm{Pred}_\pi(j)\cup\{j\}\big) - v\big(\mathrm{Pred}_\pi(j)\big)\Big)\right)\\ &=\; \frac{1}{p!} \sum_{\pi\in\mathfrak{S}_P} v(P) \;=\; \frac{1}{p!}\cdot\big(p!\cdot v(P)\big) \;=\; v(P).\end{aligned}$$

□

When using the set-based definition, one can show that the term $v(P)$ appears exactly once per player (hence p times in total) for the coalition $S = P \setminus \{j\}$ with the weight $\frac{|P-1|!\,\big(|P|-|P-1|-1\big)!}{|P|!} = \frac{1}{p}$ each, and then one just has to show that for all the other terms, their number of appearance together with their weights cause them to cancel out. See also here.

(c) **Bonus: Proof of Linearity**

This is as straightforward as the additivity:

Theorem 5 (Linearity)**.** *Let $\alpha, \beta \in \mathbb{R}$ and $v, v_1, v_2 : 2^P \to \mathbb{R}$ be value functions with $v = \alpha v_1 + \beta v_2$, which means that for every subset $S \subseteq P$ we have $v(S) = \alpha \cdot v_1(S) + \beta \cdot v_2(S)$. Let $\{\phi_{j,v}\}_{j\in P}$ be the Shapley values induced by v and $\{\phi_{j,v_1}\}_{j\in P}$ and $\{\phi_{j,v_2}\}_{j\in P}$ those for v_1 and v_2 respectively. Then for any player $j \in P$ it holds:*

$$\phi_{j,v} = \phi_{j,\alpha v_1 + \beta v_2} = \alpha \phi_{j,v_1} + \beta \phi_{j,v_2}.$$

Proof. Since additivity was already proven in part (a), we only have to prove homogeneity, that means that $\phi_{j,\alpha v_1} = \alpha \phi_{j,v_1}$ for any $\alpha \in \mathbb{R}$.

So let $p \geq 1$, $j \in P$, $\alpha \in \mathbb{R}$ and v_1 be any value function. Denote $v = \alpha v_1$. For the marginal contributions of player j on any subset $S \subseteq P$ we have: $v(S \cup \{j\}) - v(S) = \alpha v_1(S \cup \{j\}) - \alpha v_1(S) = \alpha\Big(v_1(S \cup \{j\}) - v_1(S)\Big)$. As for the additivity, plugging this into either of the two definitions yields the results:

$$\begin{aligned}
\phi_{j,v} &= \sum_{S \subseteq P \setminus \{j\}} \frac{|S|!\,\big(p - |S| - 1\big)!}{p!} \Big(v(S \cup \{j\}) - v(S)\Big) = \sum_{S \subseteq P \setminus \{j\}} \frac{|S|!\,\big(p - |S| - 1\big)!}{p!}\, \alpha\Big(v_1(S \cup \{j\}) - v_1(S)\Big) \\
&= \alpha \sum_{S \subseteq P \setminus \{j\}} \frac{|S|!\,\big(p - |S| - 1\big)!}{p!} \Big(v_1(S \cup \{j\}) - v_1(S)\Big) = \alpha \phi_{j,v_1}.
\end{aligned}$$

$\square$