Solution 1:

a) Derivation of PD function for $S = \{1\}$ (with $C = \{2\}$) given

$$\hat{f}(\mathbf{x}) = \hat{f}(x_1, x_2) = \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_1 x_2 + \hat{\beta}_0$$

$$\begin{aligned}
f_{1,PD}(x_1) = \mathbb{E}_{x_2}\left(\hat{f}(x_1, x_2)\right) &= \int_{-\infty}^{\infty} \left(\hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \hat{\beta}_3 x_1 x_2 + \hat{\beta}_0\right) \, d\mathbb{P}(x_2) \\
&= \hat{\beta}_1 x_1 + \int_{-\infty}^{\infty} \hat{\beta}_2 x_2 + \hat{\beta}_3 x_1 x_2 \, d\mathbb{P}(x_2) + \hat{\beta}_0 \\
&= \hat{\beta}_1 x_1 + \int_{-\infty}^{\infty} (\hat{\beta}_2 + \hat{\beta}_3 x_1) x_2 \, d\mathbb{P}(x_2) + \hat{\beta}_0 \\
&= \hat{\beta}_1 x_1 + (\hat{\beta}_2 + \hat{\beta}_3 x_1) \cdot \int_{-\infty}^{\infty} x_2 \, d\mathbb{P}(x_2) + \hat{\beta}_0 \\
&= \hat{\beta}_1 x_1 + (\hat{\beta}_2 + \hat{\beta}_3 x_1) \cdot \mathbb{E}_{x_2}(x_2) + \hat{\beta}_0
\end{aligned}$$

b) PD function for $\hat{\beta}_0 = 0$, $\hat{\beta}_1 = -8$, $\hat{\beta}_2 = 0.2$, $\hat{\beta}_3 = 16$, $X_1 \sim Unif(-1, 1)$ and $X_2 \sim B(1, 0.5)$.

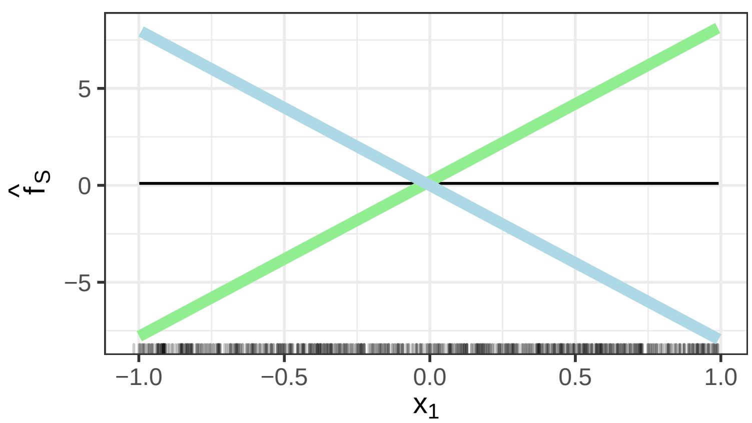
$$\begin{aligned}
f_{1,PD}(x_1) = \hat{\beta}_1 x_1 + (\hat{\beta}_2 + \hat{\beta}_3 x_1) \cdot \mathbb{E}_{x_2}(x_2) + \hat{\beta}_0 &= -8x_1 + (0.2 + 16x_1) \cdot \mathbb{E}_{x_2}(x_2) + 0 \\
&= -8x_1 + (0.2 + 16x_1) \cdot 0.5 \\
&= -8x_1 + 0.1 + 8x_1 \\
&= 0.1 + 0x_1
\end{aligned}$$

c) ICE functions for group $X_2 = 1$ and for group $X_2 = 0$:

$$f_1(x_1) = \begin{cases} -8x_1 + (0.2 + 16x_1) \cdot 1 = 8x_1 + 0.2 & x_2 = 1 \\ -8x_1 + (0.2 + 16x_1) \cdot 0 = -8x_1 & x_2 = 0 \end{cases}$$

The light green dots correspond to group $X_2 = 1$, the light blue dots to group $X_2 = 0$.

d) The example illustrates that by averaging of ICE curves for a PD plot, we might obfuscate heterogeneous effects and interactions. Although ICE curves showed a strong effect of X_1 on Y, the effect was not apparent in PDPs. Therefore, it is highly recommended to plot PD plots and ICE curves together.



Solution 2:

a) Pseudocode of `get_bounds()`

Algorithm 1 `get_bounds()`

```
Require: X: input data
Require: s: index of features for calculating ALE
Require: n_intervals: number of intervals
x_s ← s-th column of X
x_s_min ← min value of x_s
x_s_max ← max value of x_s
return equidistant sequence of n_intervals + 1 grid points between x_s_min and x_s_max
```

b) Pseudocode of `calculate_ale()`

Algorithm 2 `calculate_ale()`

```
Require: model : Classifier
Require: X: input data
Require: s: index of feature for calculating ALE
Require: n_intervals: number of intervals
Require: centered: whether to return centered or uncentered ALE
bounds ← get_bounds(X, s, n_intervals)
lowerbound ← lower interval bounds
upperbound ← upper interval bounds
result ← {}
for i1 in lowerbound & i2 in upperbound do
    idx ← ids of datapoints of X ∈ (i1, i2]          ▷ for first interval [i1, i2]
    if idx = ∅ then diff ← 0
    else if idx ≠ ∅ then
        X_min ← X[idx,] with feature values of s replaced by i1
        X_max ← X[idx,] with feature values of s replaced by i2
        y_min ← model predictions of X_min
        y_max ← model predictions of X_max
        diff ← y_max - y_min
    end if
    results ← {results, mean of diff}
end for
uncentered_ale ← cumulative sum of result
if centered then centered_ale ← uncentered_ale - (mean of uncentered_ale)
end if
return bounds and either uncentered_ale or centered_ale
```

c) Pseudocode of `prepare_ale()`

Algorithm 3 `prepare_ale()`

Require: `model` : Classifier
Require: `X`: input data
Require: `s`: index of feature for calculating ALE
Require: `n_intervals`: number of intervals
Require: `centered`: whether to return centered or uncentered ALE

`bounds, y` ← `prepare_ale`(`X`, `s`, `n_intervals`, `centered`)
`lowerbound` ← lower interval bounds
`upperbound` ← upper interval bounds
`x` ← center of each interval (middle of `lowerbound` and `upperbound`)
return `x` and `y`

Solution 3:

a) Overall, all customers, regardless of their personal status and gender, have on average a high probability of being a low (good) risk for the bank. The average marginal prediction for divorced or separated male customers reveals a slightly higher risk for this group.

b) The ALE is faster to compute and unbiased. Unbiasedness means that it does not suffer from the extrapolation problem which is especially apparent in PDPs when features are correlated.

c) The following pseudocode computes the pairwise sum of the absolute differences of relative frequencies in the categories of a categorical feature x_j based on a feature x_k

Algorithm 4 `get_diff_cat()`

Require: `feature.k`: values of categorical feature for which relative frequencies per class are calculated
Require: `feature.j`: values of categorical feature for which similarity based on `feature.k` should be assessed

`dists` ← unique class combinations of `feature.j`
`x.count` ← number of observations per class of `feature.j`
`A` ← relative cross table of `feature.j` and `feature.k` weighted by `x.count` (per class of `feature.j` relative frequencies of classes of `feature.k` should sum up to 1)
`dist` ← sum up distances of probability distributions per unique class combination specified in `dists`
return `dist`, `dists`

For our task at hand, we obtain the following distances

```
                                 class1                                  class2      dist
1                 male : married/widowed                  male : married/widowed 0.0000000
2  female : non-single or male : single                  male : married/widowed 0.4482929
3              male : divorced/separated                  male : married/widowed 0.3747445
4                        female : single                  male : married/widowed 0.4976198
5                 male : married/widowed female : non-single or male : single 0.4482929
6  female : non-single or male : single female : non-single or male : single 0.0000000
7              male : divorced/separated female : non-single or male : single 0.2864516
8                        female : single female : non-single or male : single 0.1586255
9                 male : married/widowed               male : divorced/separated 0.3747445
10 female : non-single or male : single               male : divorced/separated 0.2864516
11             male : divorced/separated               male : divorced/separated 0.0000000
12                       female : single               male : divorced/separated 0.2921739
13                male : married/widowed                         female : single 0.4976198
14 female : non-single or male : single                         female : single 0.1586255
15             male : divorced/separated                         female : single 0.2921739
16                       female : single                         female : single 0.0000000
```

Overall, the following ordering of `personal_status_sex` was returned by the method:

```
[1] "female : single"                "female : non-single or male : single"
[3] "male : married/widowed"         "male : divorced/separated"
```

The ordering seems to be feasible, since categories including females are close to each other and also categories with males. Also the ordering of males according to their relationship status seems to make sense, since typically the process is: single, then married and then divorced :-).

d) **Bonus:** ALE and PDP are global interpretation tools which base their insights on averages (of predictions or prediction differences) over whole test sets. Indeed vulnerable groups are typically not the majority of a population but have a low proportion, and biases might be overlooked. Therefore, local explanation tools should be consulted, in addition to these methods in order to identify pointwise biases or discriminatory behavior.