

Dataset AMP

Dataset de medição do produto AMP (adoro meu pet). O produto é constituído de 05 módulos identificados de M01 a M05.

O produto foi construído em 80 sprints numeradas de 1 a 80. Para cada sprint foi feita uma estimativa em use case points das quais se derivaram estimativas em horas. Para cada sprint foram medias as horas realizadas.

Os defeitos são classificados em estáticos e dinâmicos. Durante a construção do produto foram identificados 103 defeitos numerados de 5961 a 6063.

O dataset é formado por 5 arquivos no formato csv:

- AMP_modulo_sprint
- AMP_esforco_sprint
- AMP_modulo defeito
- AMP_defeitos_dinamicos
- AMP_defeitos_estaticos

AMP_modulo_sprint.csv

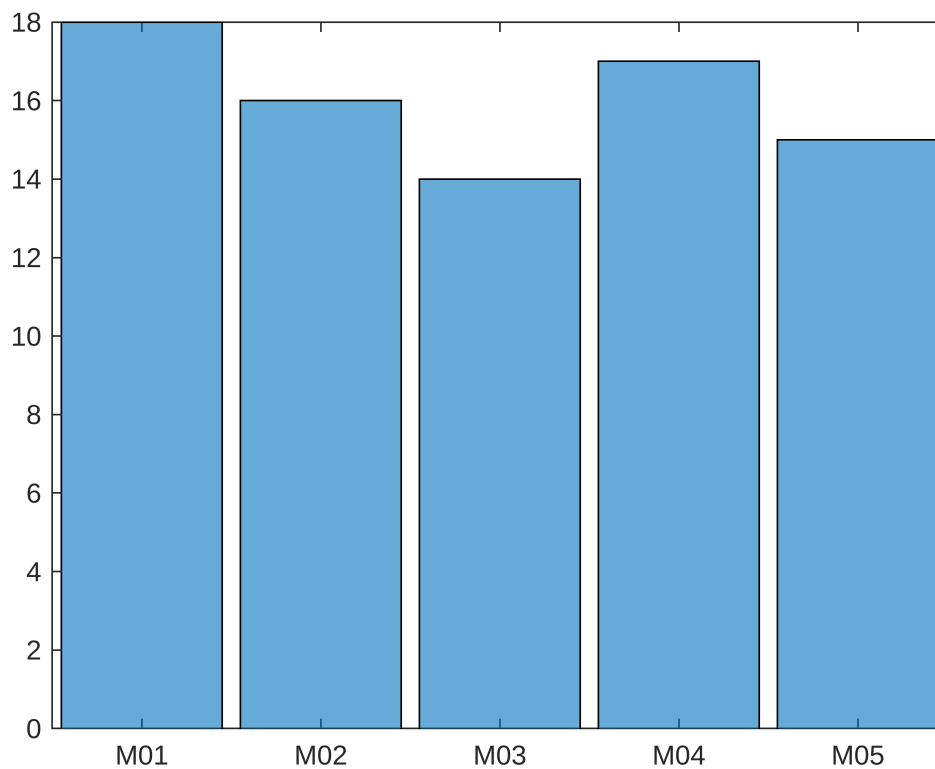
O produto tem cinco módulos identificados de M01 a M05.

Durante a construção do produto foram realizadas 80 sprints numeradas de 1 a 80.

O arquivo **AMP_modulo_sprint.csv** relaciona as sprints com os módulos.

O histograma a seguir mostra a quantidade de sprints realizadas para construção de cada módulo.

```
MS = readtable('AMP_modulo_sprint.csv');  
modulo = categorical(MS.ID_MODULO);  
histogram(modulo)
```



AMP_esforco_sprint.csv

Durante a construção do produto foram realizadas 80 sprints numeradas de 1 a 80.

Para cada sprint o arquivo armazena as seguintes informações:

- ID_SPRINT: identificador da sprint
- Estimativa_UCP: estimativa de esforço em use case points
- Estimativa_Horas: estimativa do esforço em horas
- Realizado_Horas: esforço realizado em horas

```
ES = readtable('AMP_esforco_sprint.csv');  
fprintf("Média do esforço realizado nas sprints = %.2f horas\n", mean(ES.Realizado_Horas));
```

```
Média do esforço realizado nas sprints = 114.60 horas
```

```
fprintf("Desvio padrão do esforço realizado nas sprints = %.2f horas\n", std(ES.Realizado_Horas));
```

```
Desvio padrão do esforço realizado nas sprints = 74.83 horas
```

```
fprintf("Qualtil 0.95 do esforço realizado nas sprints = %.2f horas\n", quantile(ES.Realizado_Horas, 0.95));
```

```
Qualtil 0.95 do esforço realizado nas sprints = 245.38 horas
```

AMP_modulo_defeito.csv

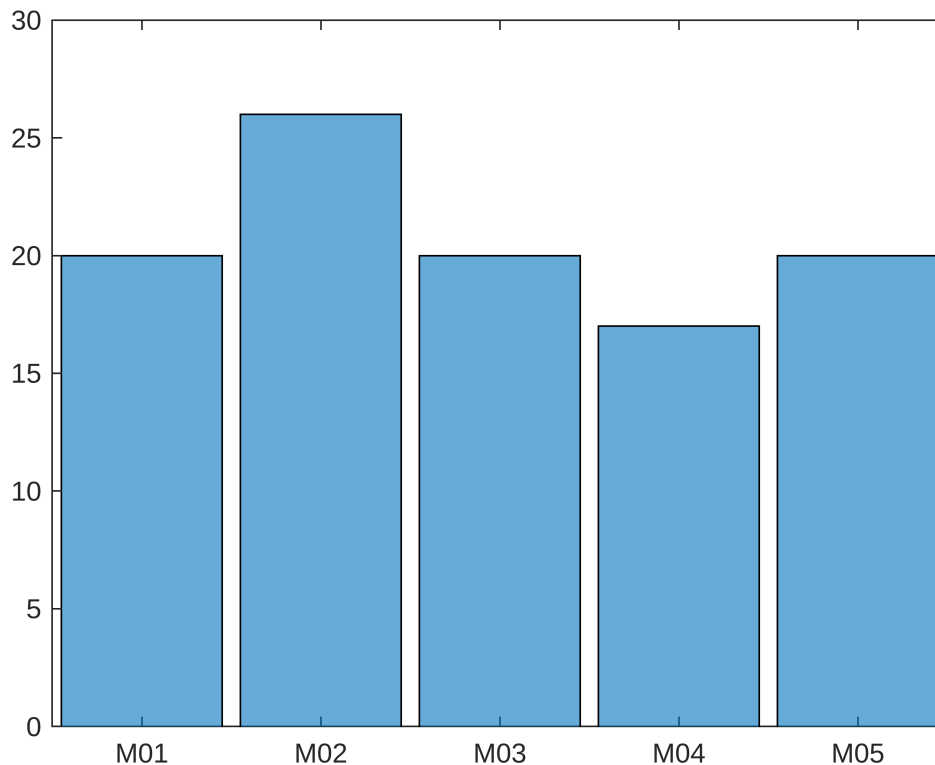
O produto tem cinco módulos identificados de M01 a M05.

Durante a construção do produto foram identificados 103 defeitos numerados de 5961 a 6063.

O arquivo CEP_modulo_defeito relaciona os defeitos com os módulos.

O histograma a seguir mostra a quantidade de defeitos em cada módulo.

```
MD = readtable('AMP_modulo_defeito.csv');  
modulo = categorical(MD.ID_MODULO);  
histogram(modulo)
```



AMP_defeitos_dinamicos.csv

O arquivo defeitos_dinamicos contém a informação de evolução no tempo dos defeitos, com as seguintes variáveis:

- ID_DEFEITO: identificador do defeito de 5961 a 6063
- TIPO_REGISTRO: ABERTURA, ALTERACAO ESTADO, FECHAMENTO
- DATA_REGISTRO: data de abertura (formato dia-mês-ano hora:minuto:segundo)
- SEVERIDADE_REGISTRO: severidade atribuída ao tratamento do registro dinâmico (1 a 5)
- DADOS_REGISTRO: Descrição do registro

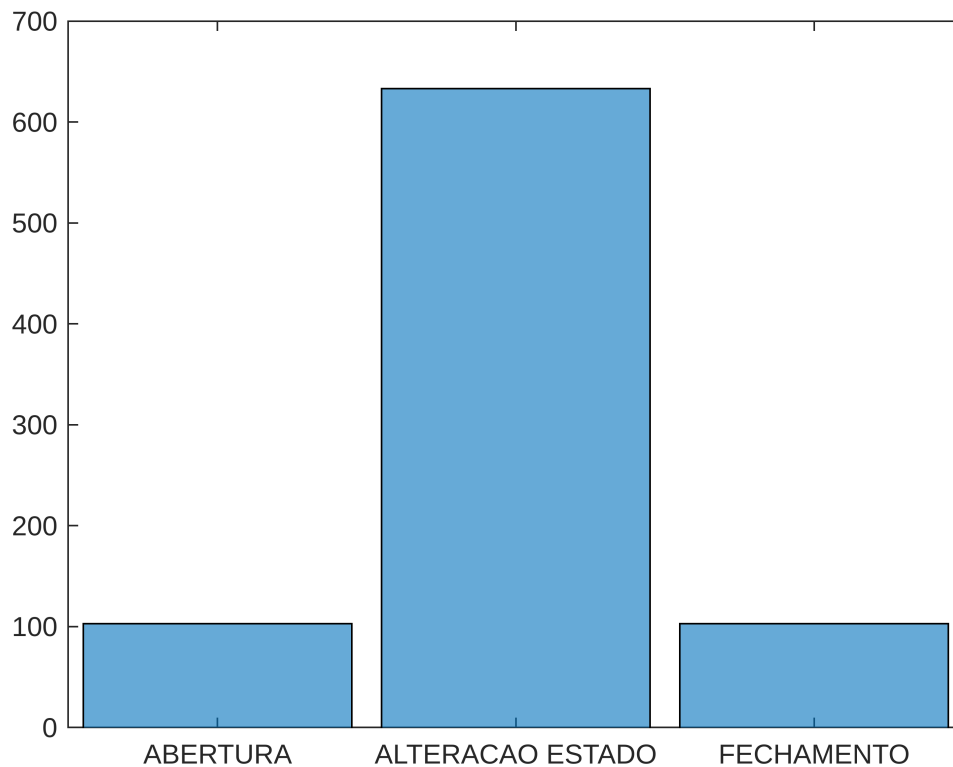
Carregar o arquivo na tabela DefeitosDinamicos

```
DefeitosDinamicos = readtable('AMP_defeitos_dinamicos.csv');
```

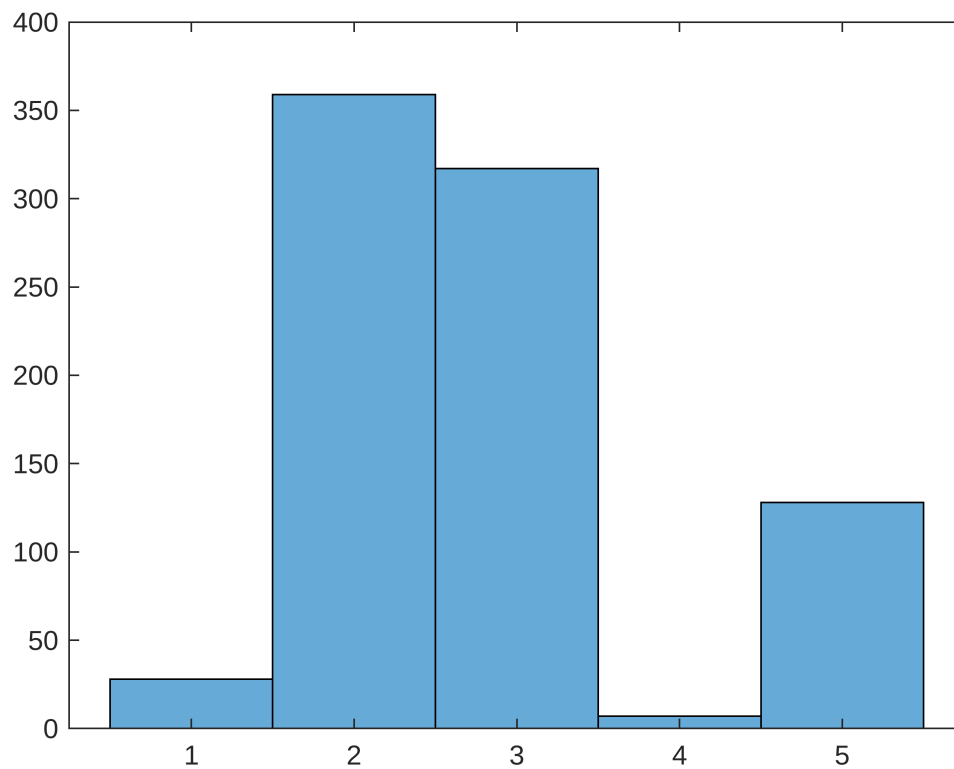
Construir o histograma das variáveis TIPO_REGISTRO e SEVERIDADE_REGISTRO

Converter o tipo de dados para categorial, quando necessário.

```
DefeitosDinamicos.TIPO_REGISTRO = categorical(DefeitosDinamicos.TIPO_REGISTRO);  
h1 = histogram(DefeitosDinamicos.TIPO_REGISTRO);
```



```
h2 = histogram(DefeitosDinamicos.SEVERIDADE_REGISTRO);
```

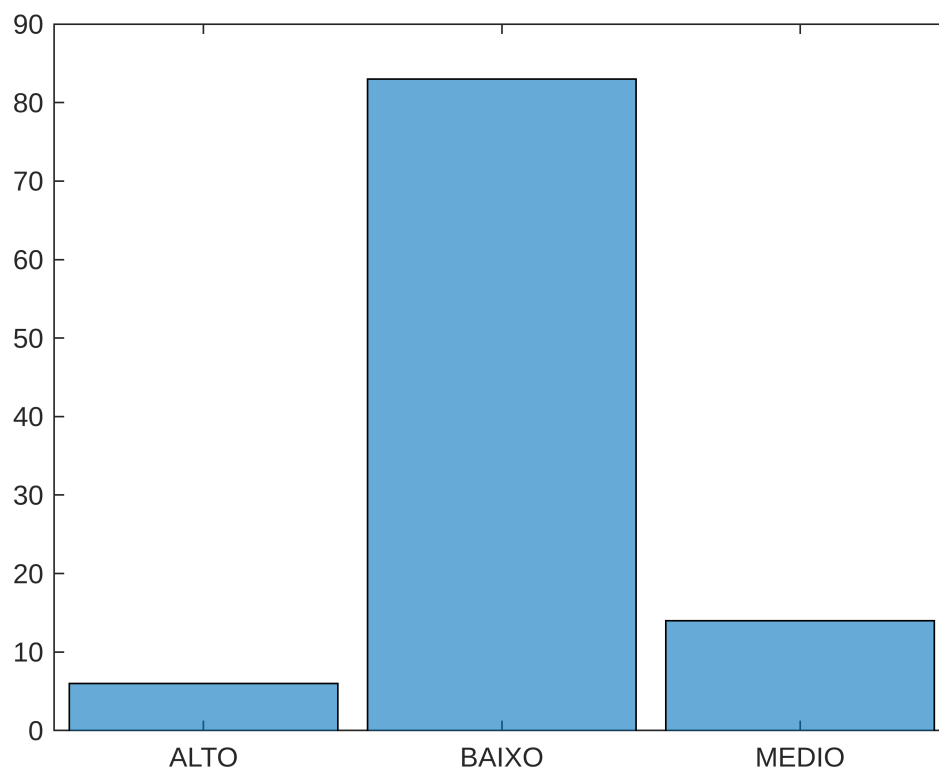


AMP_defeitos_estaticos.csv

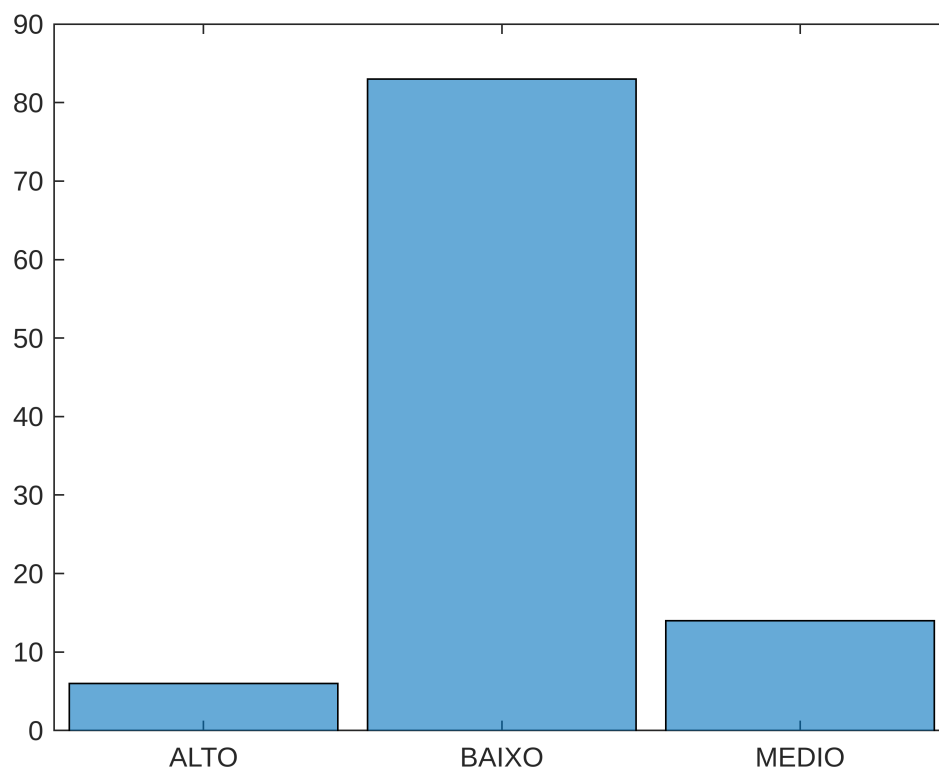
O arquivo defeitos_estáticos contém a informação dos defeitos que não se modificam no tempo dos defeitos, com as seguintes variáveis:

- ID_DEFEITO: identificador do defeito de 5961 a 6063
- SEVERIDADE_REGISTRO: severidade atribuída ao tratamento do defeito (estático, 1 a 5)
- PRIORIDADE: prioridade atribuída ao defeito
- CUSTO: custo de solução: BAIXO, MEDIO, ALTO
- EST_HORAS_REPARO: Estimativa de horas para o reparo
- HORAS_REPARO: Horas realizadas para o reparo
- COMO_ENCONTRADO: Como o defeito foi encontrado
- TIPO_PROBLEMA: Tipo do problema
- SLOC_COUNT: Quantidade de linhas de código (source lines of code)

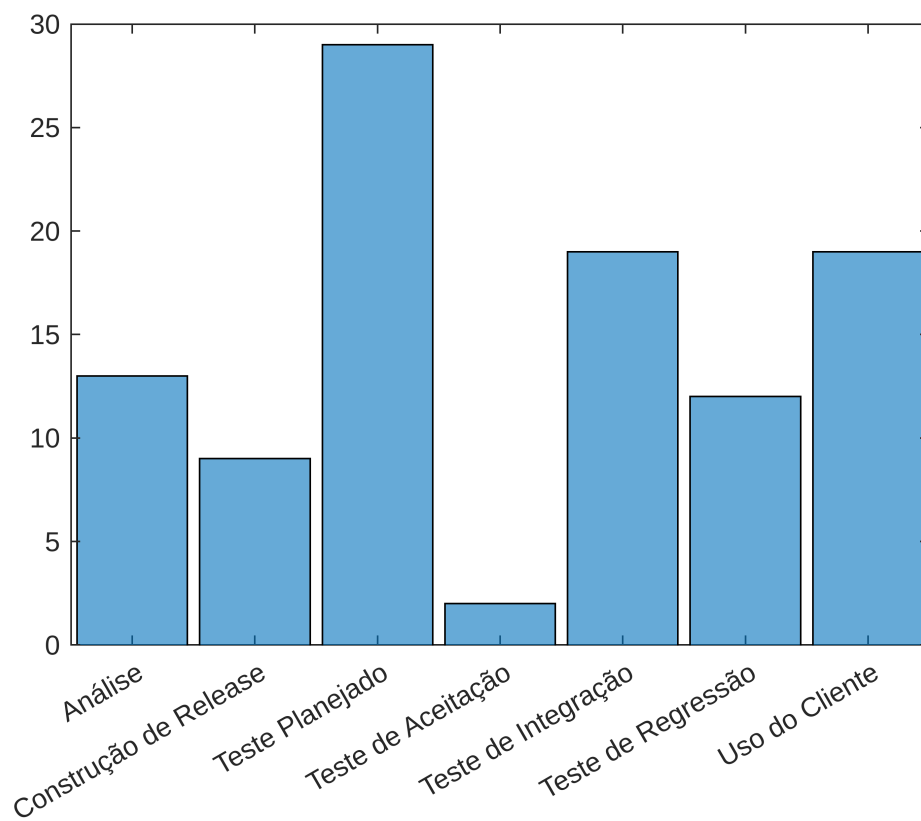
```
DefeitosEstaticos = readtable('AMP_defeitos_estaticos.csv');  
DefeitosEstaticos.CUSTO = categorical(DefeitosEstaticos.CUSTO);  
h = histogram(DefeitosEstaticos.CUSTO);
```



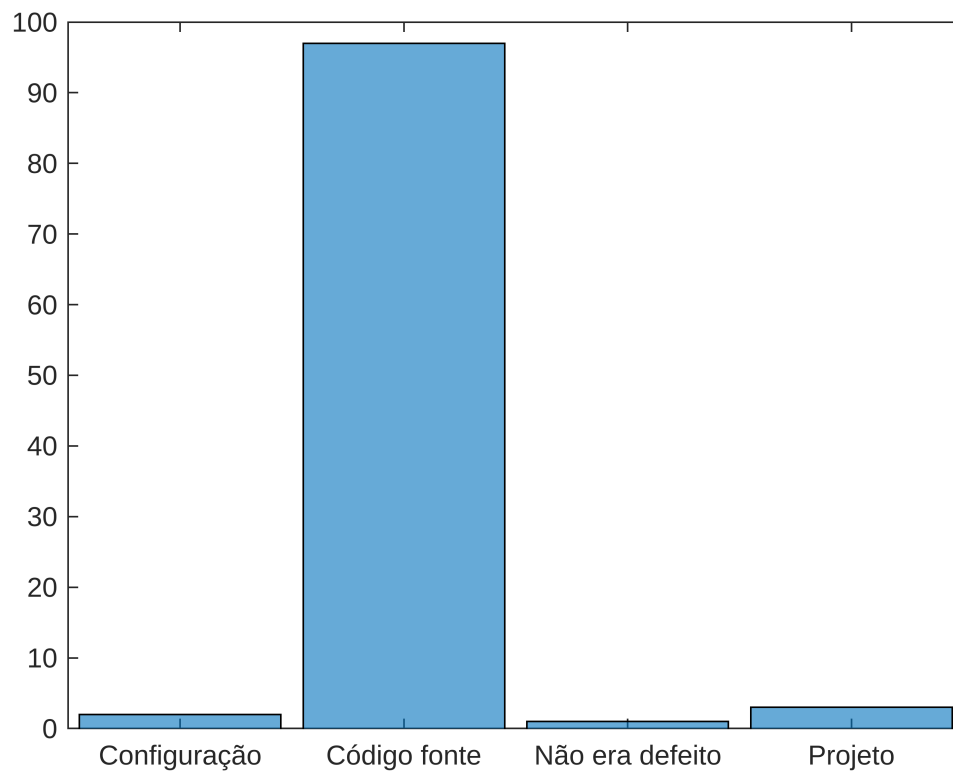
```
DefeitosEstaticos.CUSTO = categorical(DefeitosEstaticos.CUSTO);  
h = histogram(DefeitosEstaticos.CUSTO);
```



```
DefeitosEstaticos.COMO_ENCONTRADO = categorical(DefeitosEstaticos.COMO_ENCONTRADO);  
h = histogram(DefeitosEstaticos.COMO_ENCONTRADO);
```



```
DefeitosEstaticos.TIPO_PROBLEMA = categorical(DefeitosEstaticos.TIPO_PROBLEMA);  
h = histogram(DefeitosEstaticos.TIPO_PROBLEMA);
```

Combinar dados através de colunas em comum

- `innerjoin(tabela_esquerda, tabela_direita, 'Keys',{'Chave'})`
- `outerjoin(tabela_esquerda, tabela_direita, 'Keys',{'Chave'})`

Exemplo: Calcular a média de estimativas de use case points nas sprints do módulo 01

Os dados de módulo estão na tabela MS (módulo sprint) e os dados de UCP estão na tabela ES (esforço sprint)

- Fazer o outerjoin das tabelas usando como chave a coluna ID_SPURT
- Selecionar as linhas do Módulo 01
- Calcular a média

```
% Outer Join
Sprints = outerjoin(MS,ES,'Keys',{'ID_SPURT'}, 'MergeKeys',true);
% Transformar ID_MODULO para o tipo categorical para permitir filtragem por
% linha
Sprints.ID_MODULO = categorical(Sprints.ID_MODULO);
```

```
% Selecionar as linhas do módulo 01
%indice = (Sprints.ID_MODULO == 'M01');
Sprints_M01 = Sprints(Sprints.ID_MODULO == 'M01', [1:6])
```

Sprints_M01 = 18x6 table

...

	ID_MODULO	ID_SPRINT	Estimativa_UCP	Estimativa_Horas
1	M01	1	16.7437	91.8900
2	M01	2	41.3325	225.7400
3	M01	5	44.3278	261.5300
4	M01	6	49.2950	283.8600
5	M01	10	17.1645	130.8000
6	M01	16	3.7973	16.3200
7	M01	19	25.6640	110.3200
8	M01	21	8.3113	135.7300
9	M01	23	57.0634	288.2900
10	M01	27	15.0453	84.6800
11	M01	36	12.0159	51.6500
12	M01	38	8.5145	43.6000
13	M01	41	26.8078	115.2400
14	M01	47	50.3940	173.6400
15	M01	49	45.2938	214.7100
16	M01	62	45.9896	154.7100
17	M01	72	7.2325	31.0900
18	M01	75	15.1255	65.0200

```
% Calcular a média das estimativas de use case points
mean(Sprints_M01.Estimativa_UCP)
```

ans = 27.2288

Formativa

Calcular as estatísticas (média, desvio padrão, quantil) das horas realizadas nas sprints do módulo 5 (utilizar describe).

```
% Coloque seu código aqui
```

```
Sprints_M05 = Sprints(Sprints.ID_MODULO == 'M05', 1:6)
```

Sprints_M05 = 15x6 table

...

	ID_MODULO	ID_SPRINT	Estimativa_UCP	Estimativa_Horas
1	M05	12	66.5514	284.0800
2	M05	15	61.5668	264.6600
3	M05	22	25.1214	107.9900
4	M05	26	26.6305	114.4800
5	M05	31	43.9759	189.0400
6	M05	35	6.8153	29.3000
7	M05	37	21.0622	90.5400
8	M05	40	26.9838	116
9	M05	51	64.6212	277.7900
10	M05	54	17.4482	75.0100
11	M05	56	26.0273	111.8900
12	M05	66	56.4599	242.7100
13	M05	70	41.1542	176.9100
14	M05	79	30.4373	130.8400
15	M05	80	11.2890	48.5300

```
mean(Sprints_M05.Realizado_Horas)
```

```
ans = 145.6160
```

```
std(Sprints_M05.Realizado_Horas)
```

```
ans = 73.0660
```

```
quantile(Sprints_M05.Realizado_Horas, 0.5)
```

```
ans = 138.4000
```

Entrega

Completar o código da formativa.

Executar o LiveScript com opção VIEW Output Inline.

Exportar para pdf.

Fazer o upload no AVA.