Grupo:

Andreza Aparecida dos Santos (164213)

Leonardo Marçal (225240)

Ligia Vasconcellos (081938)

Mariana Amaral (262866)

In [1]:
```
import sklearn
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import pearsonr, spearmanr

from sklearn.linear_model import LogisticRegressionCV, LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report, confusion_matrix
```

In [2]:
```
def calculate_p_values(df, method):
    df = df.dropna()._get_numeric_data()
    df_columns = pd.DataFrame(columns=df.columns)
    p_values = df_columns.transpose().join(df_columns, how='outer')
    for r in df.columns:
        for c in df.columns:
            if method == 'pearson':
                p_values[r][c] = str(pearsonr(df[r], df[c])[1])
            else:
                p_values[r][c] = str(spearmanr(df[r], df[c])[1])
    return p_values
```

Leitura dos dados

In [3]:
```
df_zombie_survey = pd.read_csv('data/zombies-survey.csv')
df_zombie_meals = pd.read_csv('data/zombie-meals.csv')
```

Análise 1: Teste de Hipótese

Tarefa: Considerando a base Zombies Survey (zombies-survey.csv), apresente um teste de hipóteses que realize um estudo sobre a relação entre altura e gênero dos zumbis. Construa suas hipóteses, apresente a análise e as conclusões.

Hipóteses

H0 (Hipótese nula) = Não há diferença de altura (HEIGHT) entre homens (MALE) e mulheres (FEMALE)

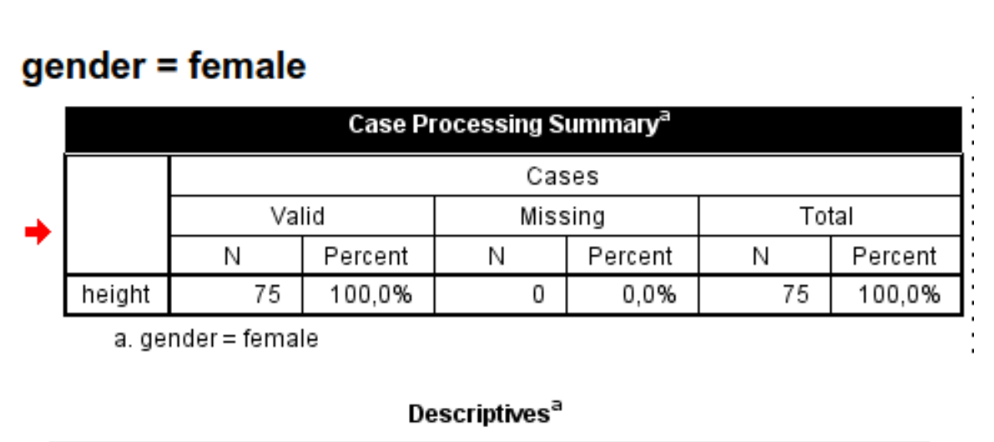
H1 (Hipótese alternativa) = Há diferença de altura entre homens e mulheres

Análises

Realizada no software estatístico **SPSS**.
O conjunto de dados foi analisado dividindo-se a variável categórica independente **GENDER**
Em seguida foi analisado a relação entre as categorias da variável independente **(FEMALE e MALE)** e a variável dependente **HEIGHT**

Analisando os dados referentes a variável GENDER = FEMALE

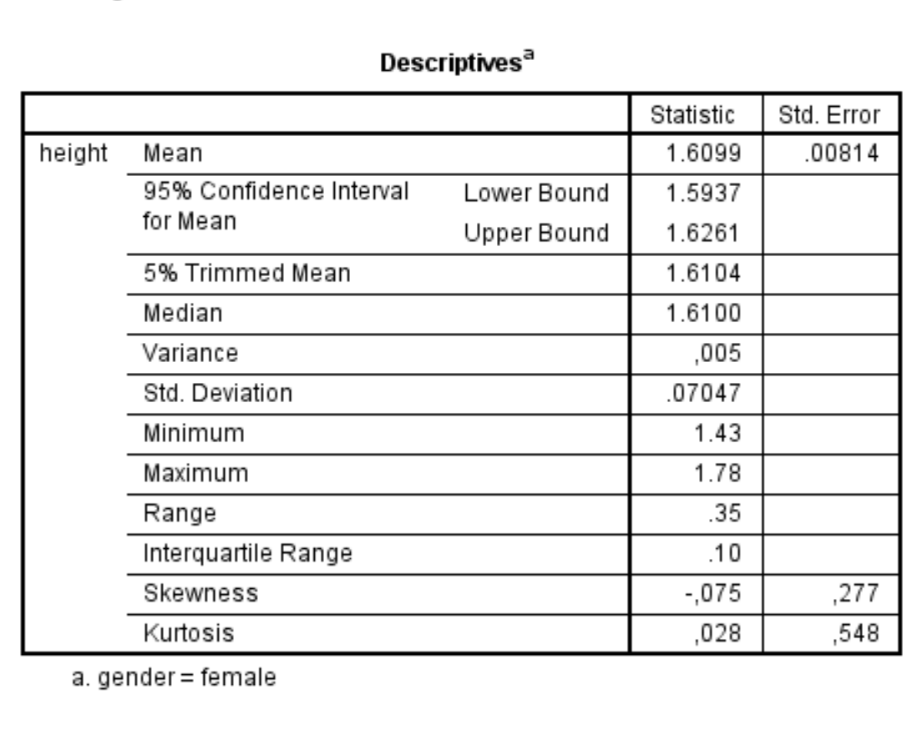
gender = female



Case Processing Summary[a]

	Cases					
	Valid		Missing		Total	
	N	Percent	N	Percent	N	Percent
height	75	100,0%	0	0,0%	75	100,0%

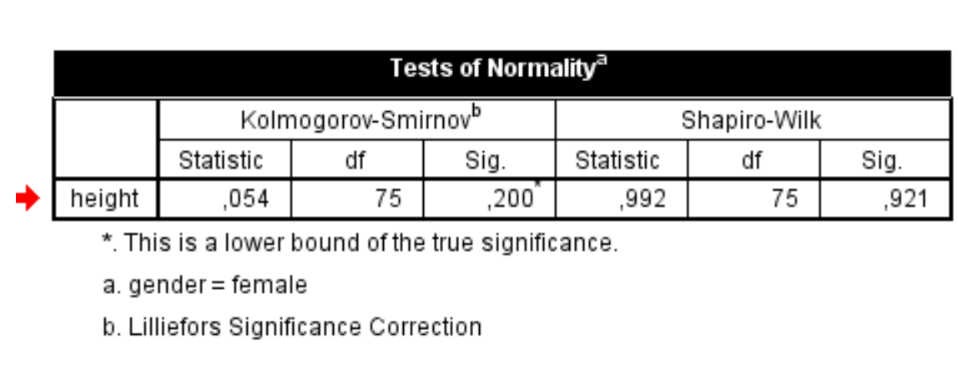
a. gender = female



Descriptives[a]

			Statistic	Std. Error
height	Mean		1.6099	.00814
	95% Confidence Interval for Mean	Lower Bound	1.5937	
		Upper Bound	1.6261	
	5% Trimmed Mean		1.6104	
	Median		1.6100	
	Variance		,005	
	Std. Deviation		.07047	
	Minimum		1.43	
	Maximum		1.78	
	Range		.35	
	Interquartile Range		.10	
	Skewness		-,075	,277
	Kurtosis		,028	,548

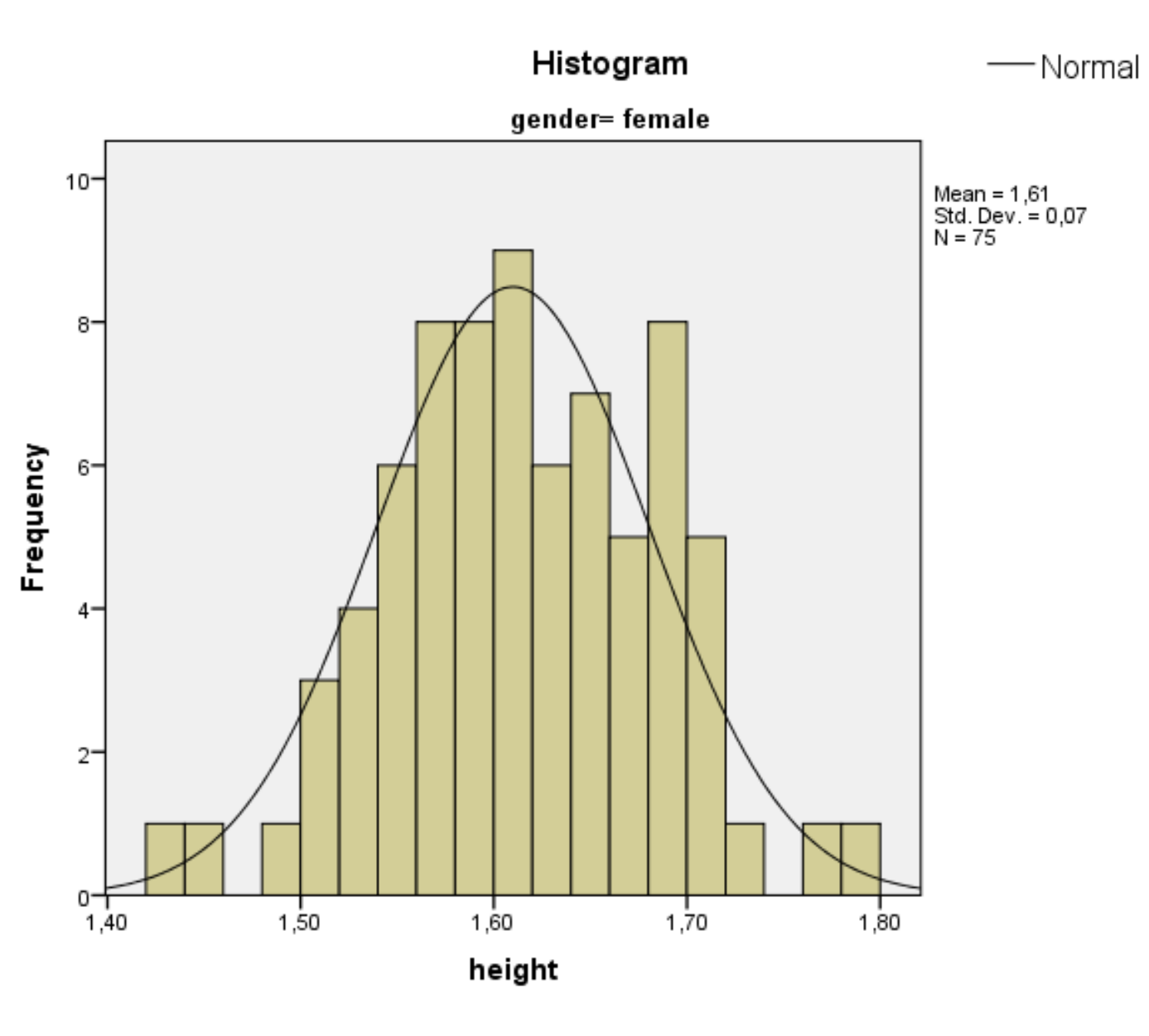
a. gender = female



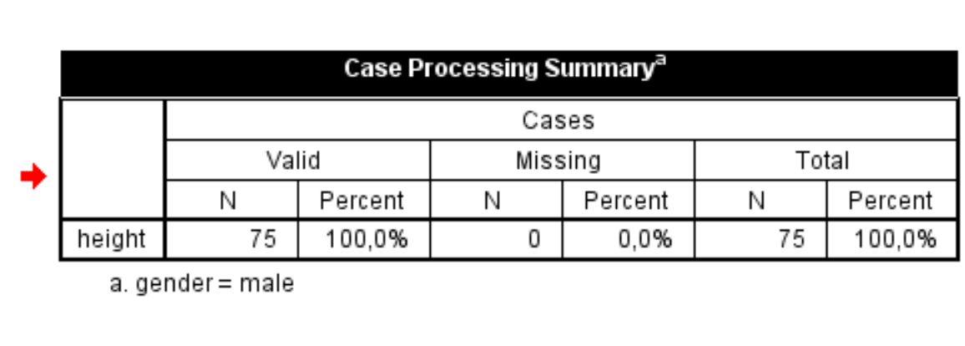
Tests of Normality[a]

	Kolmogorov-Smirnov[b]			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
height	,054	75	,200*	,992	75	,921

*. This is a lower bound of the true significance.
a. gender = female
b. Lilliefors Significance Correction



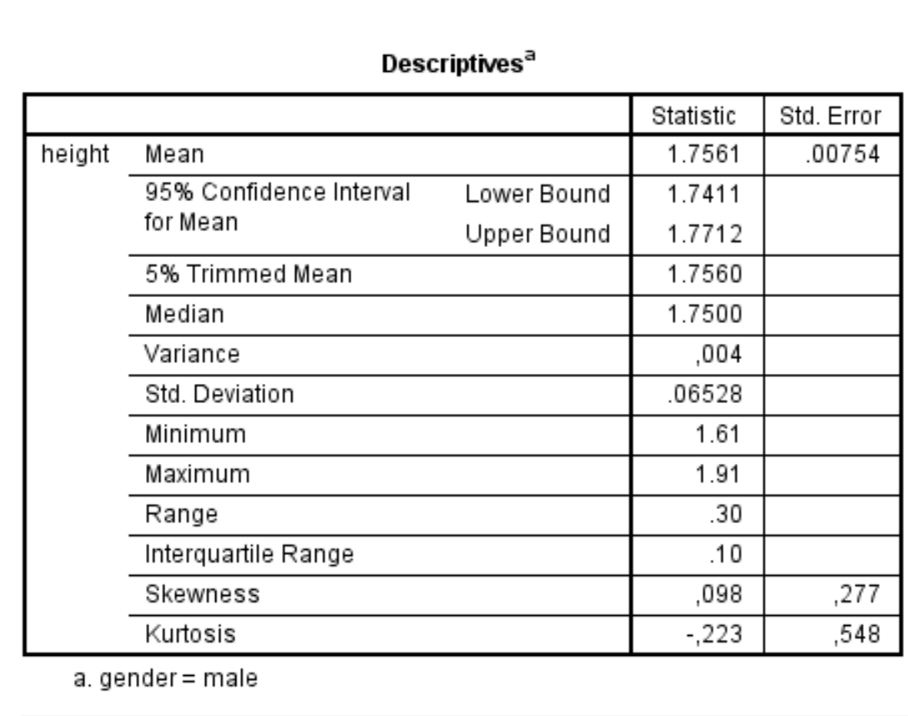
Analisando os dados referentes a variável GENDER = MALE



Case Processing Summary[a]

	Cases					
	Valid		Missing		Total	
	N	Percent	N	Percent	N	Percent
height	75	100,0%	0	0,0%	75	100,0%

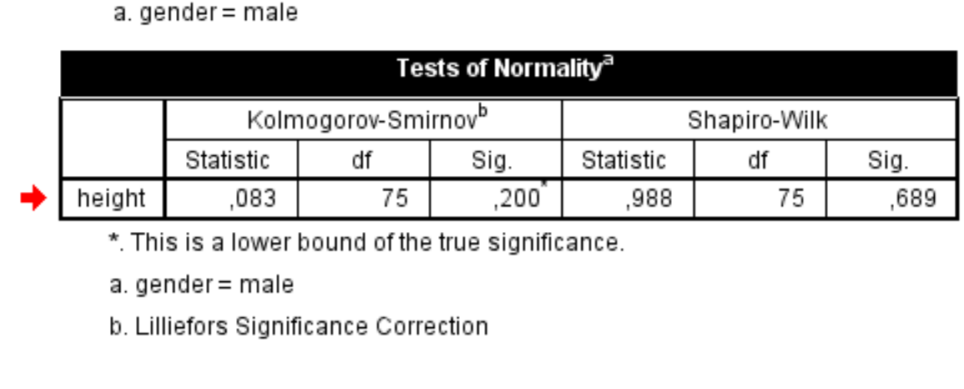
a. gender = male



Descriptives[a]

			Statistic	Std. Error
height	Mean		1.7561	.00754
	95% Confidence Interval for Mean	Lower Bound	1.7411	
		Upper Bound	1.7712	
	5% Trimmed Mean		1.7560	
	Median		1.7500	
	Variance		,004	
	Std. Deviation		.06528	
	Minimum		1.61	
	Maximum		1.91	
	Range		.30	
	Interquartile Range		.10	
	Skewness		,098	,277
	Kurtosis		-,223	,548

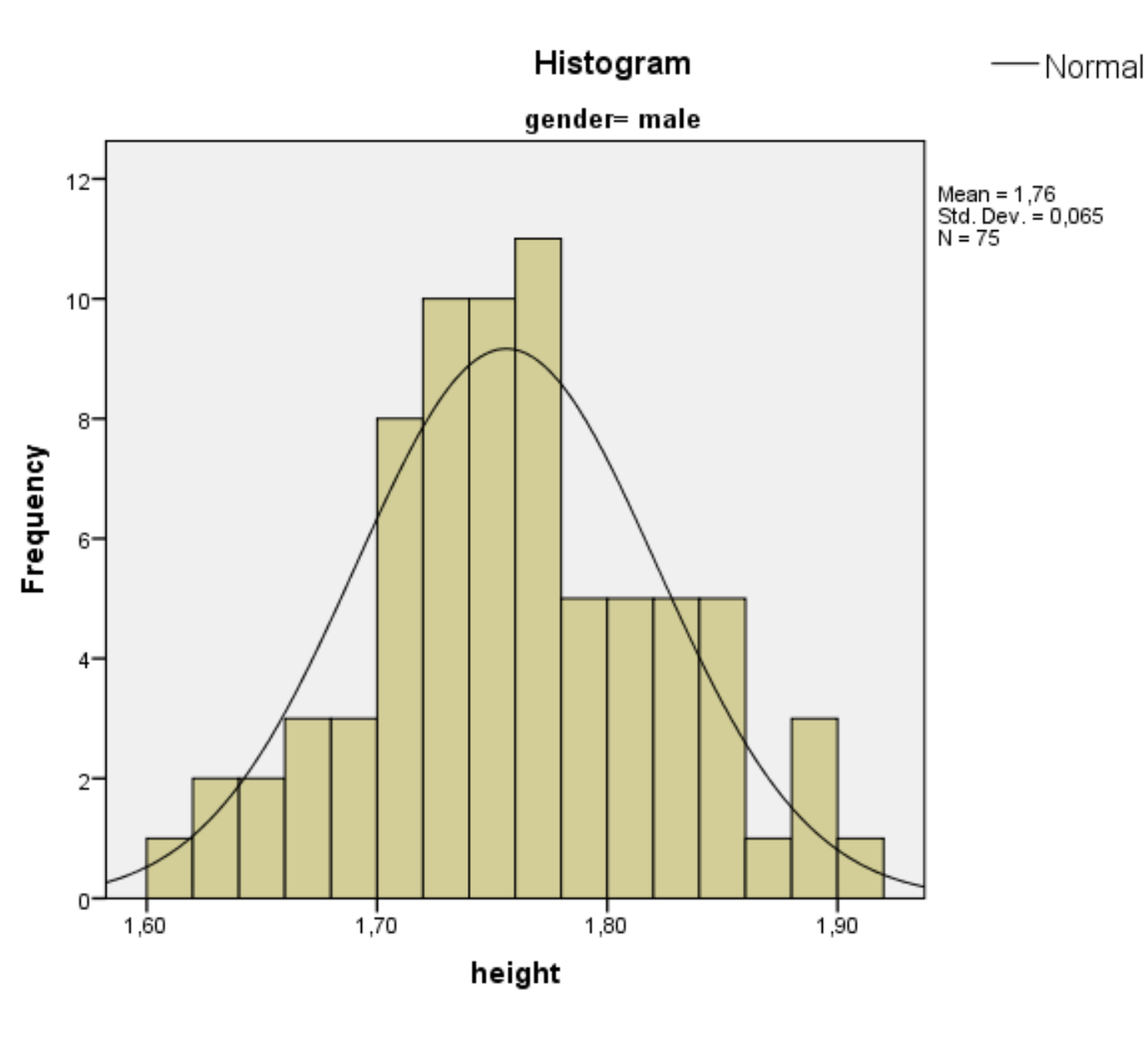
a. gender = male



Tests of Normality[a]

	Kolmogorov-Smirnov[b]			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
height	,083	75	,200*	,988	75	,689

*. This is a lower bound of the true significance.
a. gender = male
b. Lilliefors Significance Correction

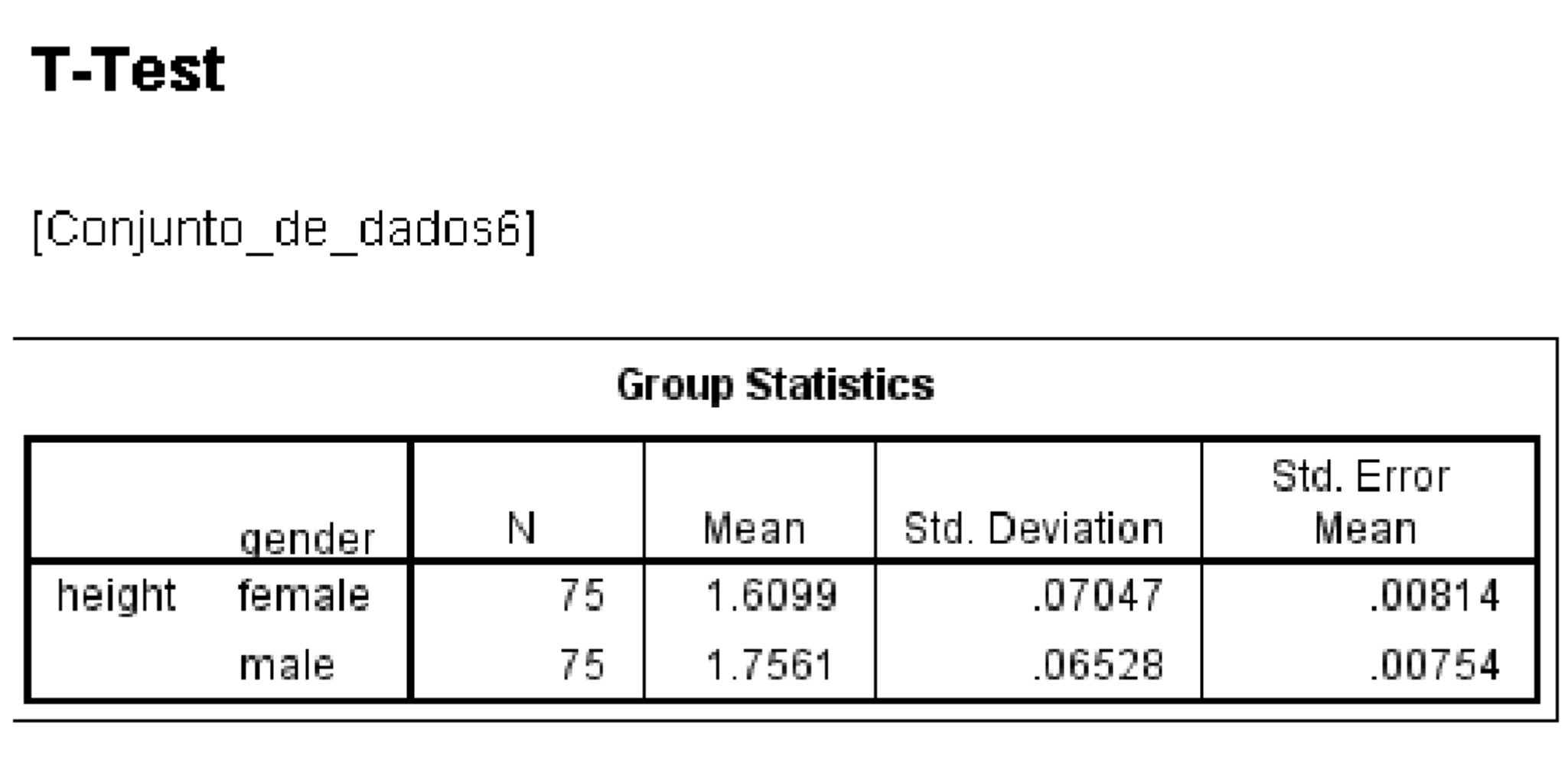


Agrupando novamente as categorias, realizamos o **teste T de amostras independentes**:

Teste T de amostras independentes

Variável(is) de teste: height

Variável de agrupamento: gender('female' 'male')

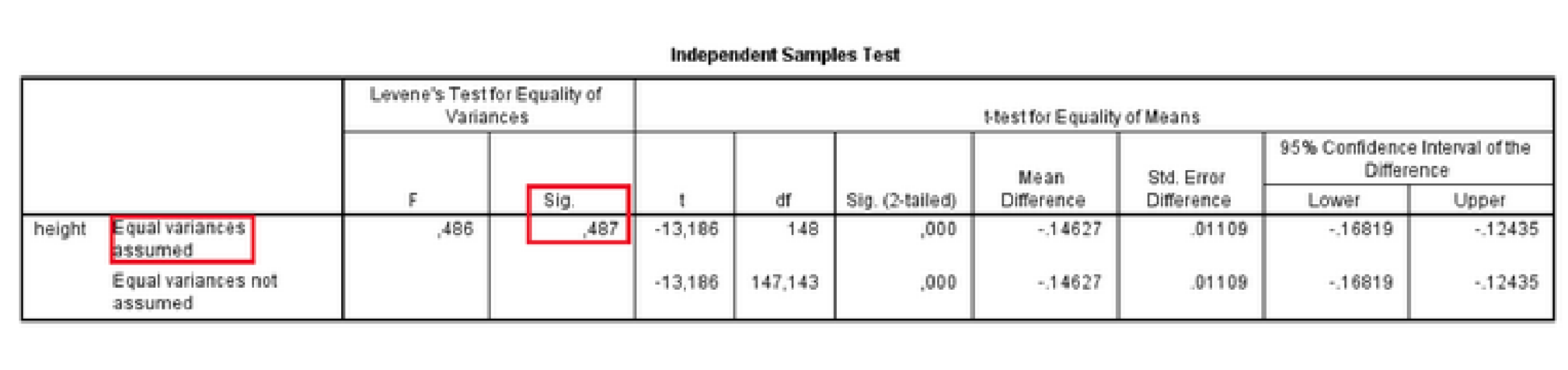


T-Test

[Conjunto_de_dados6]

Group Statistics

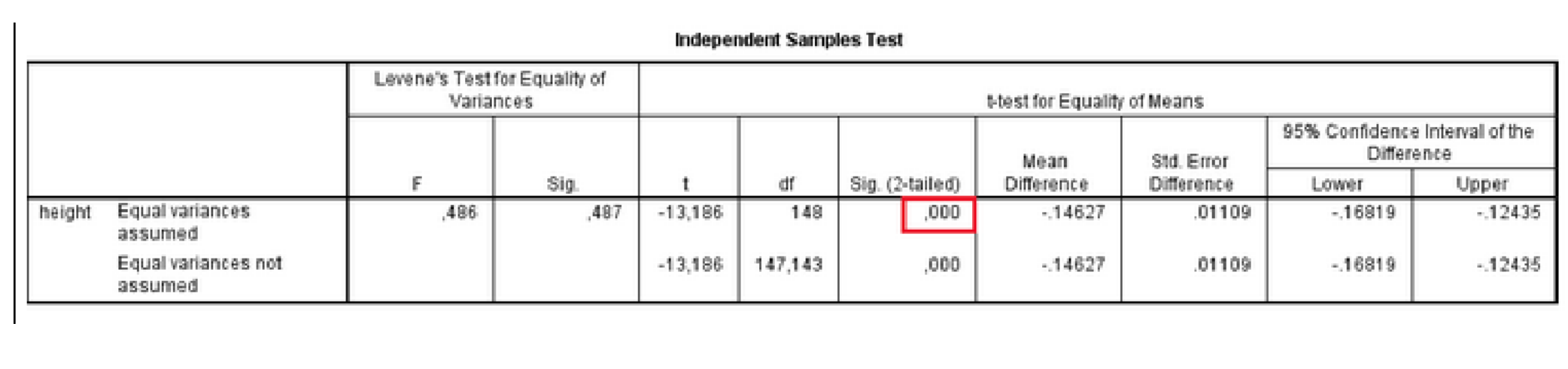
	gender	N	Mean	Std. Deviation	Std. Error Mean
height	female	75	1.6099	.07047	.00814
	male	75	1.7561	.06528	.00754



Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
height	Equal variances assumed	,486	,487	-13,186	148	,000	-,14627	,01109	-,16819	-,12435
	Equal variances not assumed			-13,186	147,143	,000	-,14627	,01109	-,16819	-,12435

O valor de sig > 0,05 mostra que as variâncias são iguais



Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
height	Equal variances assumed	,486	,487	-13,186	148	,000	-,14627	,01109	-,16819	-,12435
	Equal variances not assumed			-13,186	147,143	,000	-,14627	,01109	-,16819	-,12435

Conclusões

Valor de sig bilateral **< 0,05** = eu rejeito a hipótese nula e assumo que as médias de peso entre homens e mulheres são diferentes (hipótese alternativa).

Além disso, o intervalo de confiança é negativo e não passa por 0, sendo assim, assumi-se que 0 não está incluído na possível diferença entre as médias, rejeita-se H0 e aceita-se H1.

Análise 2: Regressão Linear

Tarefa: Considerando a base Zombie Meals (zombie-meals.csv) que lista refeições típicas de zumbis e sua caracterização em termos de proteínas, calorias, sódio, etc. A última coluna (rating) representa a nota média da avaliação feita pelos zumbis para aquela refeição.

Realize uma regressão linear e analise a influência destas características (proteínas, calorias, sódio, etc.) dos alimentos com as avaliações (ratings) feitas pelos zumbis. Analise as características que mais influenciam na avaliação (se houver) e monte um modelo de regressão (com uma ou múltiplas variáveis independentes) que você acha que melhor prediz o rating. Considere aspectos como influência dos parâmetros, correlação, R2 e overfitting.

In [4]:
```
df_zombie_meals.describe()
```

Out[4]:

	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	
count	77.000000	77.000000	77.000000	77.000000	77.000000	77.000000	77.000000	77.000000	77.000000	77.000000	77.000000	77
mean	106.883117	2.545455	1.012987	159.675325	2.151948	14.597403	6.922078	96.077922	28.246753	2.207792	1.029610	0
std	19.484119	1.094790	1.006473	83.832295	2.383364	4.278956	4.444885	71.286813	22.342523	0.832524	0.150477	0
min	50.000000	1.000000	0.000000	0.000000	0.000000	-1.000000	-1.000000	-1.000000	0.000000	1.000000	0.500000	0
25%	100.000000	2.000000	0.000000	130.000000	1.000000	12.000000	3.000000	40.000000	25.000000	1.000000	1.000000	0
50%	110.000000	3.000000	1.000000	180.000000	2.000000	14.000000	7.000000	90.000000	25.000000	2.000000	1.000000	0
75%	110.000000	3.000000	2.000000	210.000000	3.000000	17.000000	11.000000	120.000000	25.000000	3.000000	1.000000	1
max	160.000000	6.000000	5.000000	320.000000	14.000000	23.000000	15.000000	330.000000	100.000000	3.000000	1.500000	1

Devemos computar as correlações e os p-valores para cada par de variável numérica com a variável "rating" com o objetivo de verificar quais variáeis podem ser úteis para a nossa regressão linear.

In [5]:
```
R_pearson = df_zombie_meals.corr(method = 'pearson')
R_pearson
```

Out[5]:

	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cut
calories	1.000000	0.019066	0.498610	0.300649	-0.293413	0.250681	0.562340	-0.066609	0.265356	0.097234	0.696091	0.08720
protein	0.019066	1.000000	0.208431	-0.054674	0.500330	-0.130864	-0.329142	0.549407	0.007335	0.133865	0.216158	-0.24446
fat	0.498610	0.208431	1.000000	-0.005407	0.016719	-0.318043	0.270819	0.193279	-0.031156	0.263691	0.214625	-0.17589
sodium	0.300649	-0.054674	-0.005407	1.000000	-0.070675	0.355983	0.101451	-0.032603	0.361477	-0.069719	0.308576	0.11966
fiber	-0.293413	0.500330	0.016719	-0.070675	1.000000	-0.356083	-0.141205	0.903374	-0.032243	0.297539	0.247226	-0.51306
carbo	0.250681	-0.130864	-0.318043	0.355983	-0.356083	1.000000	-0.331665	-0.349685	0.258148	-0.101790	0.135136	0.36393
sugars	0.562340	-0.329142	0.270819	0.101451	-0.141205	-0.331665	1.000000	0.021696	0.125137	0.100438	0.450648	-0.03235
potass	-0.066609	0.549407	0.193279	-0.032603	0.903374	-0.349685	0.021696	1.000000	0.020699	0.360663	0.416303	-0.49519
vitamins	0.265356	0.007335	-0.031156	0.361477	-0.032243	0.258148	0.125137	0.020699	1.000000	0.299262	0.320324	0.12840
shelf	0.097234	0.133865	0.263691	-0.069719	0.297539	-0.101790	0.100438	0.360663	0.299262	1.000000	0.190762	-0.33526
weight	0.696091	0.216158	0.214625	0.308576	0.247226	0.135136	0.450648	0.416303	0.320324	0.190762	1.000000	-0.19958
cuts	0.087200	-0.244469	-0.175892	0.119665	-0.513061	0.363932	-0.032358	-0.495195	0.128405	-0.335269	-0.199583	1.00000
rating	-0.689376	0.470618	-0.409284	-0.401295	0.584160	0.052055	-0.759675	0.380165	-0.240544	0.025159	-0.298124	-0.20316

In [6]:
```
pvalores_pearson = calculate_p_values(df_zombie_meals, method='pearson')
pvalores_pearson
```

Out[6]:

	calories	protein	fat	sodium	fiber	
calories	0.0	0.8692725966354424	3.930548311924431e-06	0.007888516337690929	0.009601896675990717	0.027
protein	0.8692725966354424	0.0	0.0688970515340739	0.6367328372849335	3.5958476623668773e-06	0.25
fat	3.930548311924431e-06	0.0688970515340739	0.0	0.9627726389351892	0.8852469047311262	0.0048
sodium	0.007888516337690929	0.6367328372849335	0.9627726389351892	0.0	0.5413376454140273	0.00148
fiber	0.009601896675990717	3.5958476623668773e-06	0.8852469047311262	0.5413376454140273	0.0	0.00148
carbo	0.02787953950809777	0.2566150941216459	0.004820964962560172	0.0014858893813474415	0.0014810341035974826	
sugars	1.0249727887732055e-07	0.0034681380775225877	0.01720667114054707	0.3799765969927058	0.22059252147567893	0.0032
potass	0.5649058569748531	2.2854573005144482e-07	0.09214430487173648	0.7783348794692803	2.685747928026268e-29	0.00182
vitamins	0.019680328024492887	0.9495151700118357	0.7879384064155898	0.0012377837198298345	0.7807260000096288	0.0234
shelf	0.4001976143517816	0.24577273262611113	0.02049241098123358	0.5468352415315789	0.008588926850758344	0.3
weight	2.097529400153122e-12	0.05900622041875147	0.06087120609111044	0.006325115283939827	0.03018283108442263	0.241
cuts	0.4507918561687942	0.03213332170354314	0.12597746559561732	0.29991851835580985	1.8328178019027975e-06	0.0011
rating	4.140277400006343e-12	1.5663097849166387e-05	0.0002190278375102041	0.0002979863125537866	2.4452504430089997e-08	0.65

In [7]:
```
R_spearman = df_zombie_meals.corr(method='spearman')
R_spearman
```

Out[7]:

	calories	protein	fat	sodium	fiber	carbo	sugars	potass	vitamins	shelf	weight	cut
calories	1.000000	-0.066970	0.535217	0.290765	-0.142801	0.080772	0.596302	-0.013812	0.314042	0.152254	0.622532	0.04741
protein	-0.066970	1.000000	0.230563	-0.114397	0.680429	-0.001905	-0.285281	0.711829	-0.006971	0.283255	0.293205	-0.36255
fat	0.535217	0.230563	1.000000	0.034785	0.119626	-0.297343	0.331163	0.322380	0.116609	0.236715	0.287783	-0.24575
sodium	0.290765	-0.114397	0.034785	1.000000	-0.168594	0.376945	-0.014469	-0.115369	0.439967	-0.145355	0.263954	0.14561
fiber	-0.142801	0.680429	0.119626	-0.168594	1.000000	-0.148589	-0.109703	0.852266	-0.042424	0.316592	0.348492	-0.50759
carbo	0.080772	-0.001905	-0.297343	0.376945	-0.148589	1.000000	-0.455874	-0.197597	0.191142	-0.110154	0.119288	0.31935
sugars	0.596302	-0.285281	0.331163	-0.014469	-0.109703	-0.455874	1.000000	-0.007505	0.304366	0.073918	0.446818	-0.05666
potass	-0.013812	0.711829	0.322380	-0.115369	0.852266	-0.197597	-0.007505	1.000000	0.007495	0.360447	0.429729	-0.53534
vitamins	0.314042	-0.006971	0.116609	0.439967	-0.042424	0.191142	0.304366	0.007495	1.000000	0.269839	0.386684	0.07789
shelf	0.152254	0.283255	0.236715	-0.145355	0.316592	-0.110154	0.073918	0.360447	0.269839	1.000000	0.262832	-0.31447
weight	0.622532	0.293205	0.287783	0.263954	0.348492	0.119288	0.446818	0.429729	0.386684	0.262832	1.000000	-0.28520
cuts	0.047411	-0.362559	-0.245754	0.145612	-0.507596	0.319358	-0.056660	-0.535341	0.077897	-0.314471	-0.285207	1.00000
rating	-0.705181	0.511389	-0.443560	-0.242567	0.489379	0.241357	-0.809203	0.306618	-0.332203	0.069803	-0.278600	-0.16281

In [8]:
```
pvalores_spearman = calculate_p_values(df_zombie_meals, method = 'spearman')
pvalores_spearman
```

Out[8]:

	calories	protein	fat	sodium	fiber	
calories	0.0	0.562794427281812	5.304050527667062e-07	0.010305613803128456	0.21536788446284902	0.484
protein	0.562794427281812	0.0	0.04365768600881033	0.321847129712174	9.963020965511108e-12	0.98
fat	5.304050527667081e-07	0.04365768600881033	0.0	0.7639229432730281	0.3000737785150617	0.008
sodium	0.010305613803128456	0.32184712971217394	0.7639229432730281	0.0	0.14272386239766038	0.00072
fiber	0.21536788446284902	9.963020965511108e-12	0.3000737785150615	0.14272386239766038	0.0	0.197
carbo	0.48497690905230073	0.9868786647300243	0.00863496243496977	0.0007271203417066239	0.19715087127407424	
sugars	1.0513135932815998e-08	0.011907265197905272	0.003261999664416211	0.9006068245875288	0.3422295540445284	3.0947
potass	0.9050978900326003	3.952932552833406e-13	0.004244868132305305	0.3177261229036106	8.408679191510156e-23	0.084
vitamins	0.00541298905391098	0.9520214266679548	0.3125178454990786	6.234122442802681e-05	0.7141069947066276	0.095
shelf	0.1862032883822416	0.01255149855838096	0.038195496724085225	0.2071889856237243	0.005028679497801303	0.340
weight	1.498145295346088e-09	0.009655466517786648	0.011151557072669592	0.02036228491263532	0.0018965177351819079	0.30
cuts	0.6822036733091772	0.0011935498022749686	0.031211542483969998	0.20637873693621298	2.45586438869339e-06	0.0046
rating	8.107187106317491e-13	2.0055444688591905e-06	5.3381872906318967e-05	0.033540421351853436	6.2851600459232475e-06	0.034

Podemos concluir que as variáveis que apresentam comportamento mais relevante diante da variável "rating" são: "calories", "fiber", "protein" e "sugars".

Chegamos a essa conclusão pois, tanto para a correlação de Pearson quanto pra de Spearman, os valores de correlação obtidos para os pares `calories e rating`, `sugars e rating`, `fiber e rating` e `protein e rating`, mostraram que esses pares possuem alta correlação com p-valores bem pequenos, mostrando alto nível de certeza.

In [31]:
```
X = df_zombie_meals[['calories', 'fiber', 'sugars', 'protein']]
y = df_zombie_meals['rating']
```

In [32]:
```
linear_regression = LinearRegression(normalize=True)
X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size = 0.25,
                                                    random_state = 42,
                                                    shuffle = True)
```

In [33]:
```
print("Tamanho do conjunto de treino = {} amostras\nTamanho do conjunto de teste = {} amostras".format
(
    X_train.size,
    X_test.size))
```

```
Tamanho do conjunto de treino = 228 amostras
Tamanho do conjunto de teste = 80 amostras
```

In [34]:
```
linear_regression = linear_regression.fit(X_train, y_train)
r_square = linear_regression.score(X_test, y_test)
print("Intercept (a):", linear_regression.intercept_)
print('Coeficientes:', linear_regression.coef_)
print("R2:", r_square)
```

```
Intercept (a): 69.79838786286508
Coeficientes: [-0.21961517  2.0821638  -1.55268903  1.24401832]
R2: 0.8578989211715231
```

Conclusões

Utilizando as variáveis escolhidas a partir da análise de correlações e p-valores, podemos concluir que foram escolhas boas para a regressão linear, obtendo um modelo com R² igual a 0,85 no conjunto de teste.

Análise 3: Regressão Logística

Tarefa: Considerando a base Zombies Survey (zombies-survey.csv), realize uma regressão logistica para predizer o gênero dos zumbis a partir das outras features. Analise as features que mais influenciam na predição (se houver) e monte um modelo de regressão (com uma ou múltiplas variáveis independentes) que você acha que melhor prediz o gênero. Considere aspectos como influência dos parâmetros, correlação e overfitting.

In [13]:
```
df_zombie_survey['gender_id'] = np.where(df_zombie_survey['gender']=='male', 1, 0)
df_zombie_survey.head()
```

Out[13]:

	name	type	gender	height	weight	gbc	gs	pbc	sp	gender_id
0	Lossack Slossir	biological	male	1.88	109.72	0.8	14	5.1	3.5	1
1	Hasp Talch	biological	male	1.75	73.62	0.8	14	4.9	3.0	1
2	Lesprash Rindrad	biological	male	1.88	96.50	0.8	13	4.7	3.2	1
3	Halceff Gock	biological	male	1.82	99.81	0.8	15	4.6	3.1	1
4	Moddeche Splud	biological	male	1.77	93.60	0.8	14	5.0	3.6	1

Vamos primeiramente analisar as variáveis do dataset zombie-survey.csv. As análises visam identificar possíveis comportamentos de cada variaúel em relação às demais.

Para isso vamos computar os valores de correlação de Pearson e Spearman.

In [14]:
```
df_zombie_survey.describe()
```

Out[14]:

	height	weight	gbc	gs	pbc	sp	gender_id
count	150.000000	150.000000	150.000000	150.000000	150.000000	150.000000	150.000000
mean	1.683000	72.428400	4.794667	37.586667	5.843333	3.054000	0.500000
std	0.099837	15.001481	3.052643	17.644204	0.828066	0.433594	0.501675
min	1.430000	38.500000	0.400000	10.000000	4.300000	2.000000	0.000000
25%	1.610000	61.245000	1.200000	16.000000	5.100000	2.800000	0.000000
50%	1.685000	71.725000	5.200000	43.500000	5.800000	3.000000	0.500000
75%	1.757500	84.517500	7.200000	51.000000	6.400000	3.300000	1.000000
max	1.910000	109.720000	10.000000	69.000000	7.900000	4.400000	1.000000

In [15]:
```
df_zombie_survey_numeric_data = df_zombie_survey[['height','weight', 'gbc', 'gs', 'pbc', 'sp', 'gender
_id']]
```

In [16]:
```
# correlações de pearson
R_pearson = df_zombie_survey_numeric_data.corr(method='pearson')
R_pearson
```

Out[16]:

	height	weight	gbc	gs	pbc	sp	gender_id
height	1.000000	0.922604	0.011768	0.013053	0.011812	0.024915	0.734981
weight	0.922604	1.000000	0.050625	0.055102	0.037543	0.027108	0.811750
gbc	0.011768	0.050625	1.000000	0.962757	0.817954	-0.356544	0.017530
gs	0.013053	0.055102	0.962757	1.000000	0.871754	-0.420516	0.025779
pbc	0.011812	0.037543	0.817954	0.871754	1.000000	-0.109369	0.034735
sp	0.024915	0.027108	-0.356544	-0.420516	-0.109369	1.000000	0.016970
gender_id	0.734981	0.811750	0.017530	0.025779	0.034735	0.016970	1.000000

In [17]:
```
# P-valor para as variaves utilizando correlação de pearson
pvalores_pearson = calculate_p_values(df_zombie_survey_numeric_data, method='pearson')
pvalores_pearson
```

Out[17]:

	height	weight	gbc	gs	pbc	
height	0.0	4.202022301280149e-63	0.8863451916241203	0.8740359335284767	0.8859265689992962	0.7621682705
weight	4.202022301280149e-63	0.0	0.5384000346607991	0.5030347078447504	0.6483003076971452	0.7419344482
gbc	0.8863451916241203	0.5384000346607991	0.0	5.776660988496418e-86	2.3148491512728037e-37	7.52389095606
gs	0.8740359335284767	0.5030347078447504	5.776660988496418e-86	0.0	1.0384540627941638e-47	8.42936639295
pbc	0.8859265689992962	0.6483003076971452	2.3148491512728037e-37	1.0384540627941638e-47	0.0	0.1827652152
sp	0.7621682705239383	0.7419344482685566	7.523890956067452e-06	8.429366392950231e-08	0.1827652152713699	
gender_id	9.500026659738284e-27	2.1615141691542982e-36	0.8313934167080854	0.7541748488992304	0.6730387673403567	0.8367050517

In [18]:
```
# correlações de spearman
R_spearman = df_zombie_survey_numeric_data.corr(method='spearman')
R_spearman
```

Out[18]:

	height	weight	gbc	gs	pbc	sp	gender_id
height	1.000000	0.924758	0.035700	0.018226	0.021458	0.010272	0.760403
weight	0.924758	1.000000	0.068165	0.056425	0.042177	0.017222	0.837410
gbc	0.035700	0.068165	1.000000	0.936003	0.834421	-0.277511	0.029696
gs	0.018226	0.056425	0.936003	1.000000	0.881386	-0.303421	0.023736
pbc	0.021458	0.042177	0.834421	0.881386	1.000000	-0.159457	0.030821
sp	0.010272	0.017222	-0.277511	-0.303421	-0.159457	1.000000	0.010051
gender_id	0.760403	0.837410	0.029696	0.023736	0.030821	0.010051	1.000000

In [19]:
```
# P-valor para as variaves utilizando correlação de spearman
pvalores_spearman = calculate_p_values(df_zombie_survey_numeric_data, method='spearman')
pvalores_spearman
```

Out[19]:

	height	weight	gbc	gs	pbc	
height	0.0	5.6442008571306794e-64	0.6644975579398296	0.8248065791890679	0.7943745471372822	0.9007
weight	5.644200857130037e-64	0.0	0.4071999649180271	0.4928223586747127	0.6083214361939359	0.834
gbc	0.6644975579398296	0.4071999649180268	0.0	5.383649646073561e-69	3.9702124571318824e-40	0.0005856
gs	0.8248065791890679	0.49282235867471247	5.383649646072797e-69	0.0	4.64951031453217e-50	0.0001603
pbc	0.7943745471372822	0.6083214361939359	3.9702124571318824e-40	4.64951031453217e-50	0.0	0.05128
sp	0.9007166629314591	0.834313394477995	0.0005856929405699988	0.0001603809454660342	0.05128491483834292	
gender_id	1.5484825960886995e-29	1.1592169679368537e-40	0.7182970076128663	0.7731079571892547	0.7081015567161157	0.9028

Podemos observar que para a variável **gender_id**, que apresenta o valor 0 (zero) quando gender = female e 1 (um) quando gender = male, apresenta forte correlação com as variáveis height e weight com alto nível de certeza, ou seja, p-valor < 0.001 tanto para correlção de Pearson quanto para Spearman.

Além disso, as demais variáveis apresentam baixa correlação com a variável **gender_id**, com valores muito próximos de 0 e alto p-valor, indicando que estas variáveis não podem nos auxiliar na predição da variável gender_id.

Vamos utilizar variáveis height e weight para a regressão logística.

Construção e treinamento do modelo

In [20]:
```
regressor = LogisticRegressionCV(random_state = 42, cv = 5)
```

In [21]:
```
X = df_zombie_survey_numeric_data[['height', 'weight']]
y = df_zombie_survey_numeric_data['gender_id']
```

In [22]:
```
X_train, X_test, y_train, y_test = train_test_split(X,
                                                    y,
                                                    test_size = 0.30,
                                                    random_state = 42,
                                                    shuffle = True)
```

In [23]:
```
print("Tamanho do conjunto de treino = {} amostras\nTamanho do conjunto de teste = {} amostras".format
(
    X_train.size,
    X_test.size))
```

```
Tamanho do conjunto de treino = 210 amostras
Tamanho do conjunto de teste = 90 amostras
```

In [29]:
```
# treinamento do modelo de regressão logística
regressor = regressor.fit(X_train, y_train)
```

Avaliação do modelo de Regressão Logística

In [30]:
```
# Predição no conjunto de teste
test_predictions = regressor.predict(X_test)
```

Podemos gerar a matriz de confusão e o relatório de classificação para visualizarmos o desempenho do modelo no conjunto de teste.

In [26]:
```
report = classification_report(y_test, test_predictions, target_names=['female', 'male'], output_dict=
True)
```

In [27]:
```
report_df = pd.DataFrame(report)
report_df
```

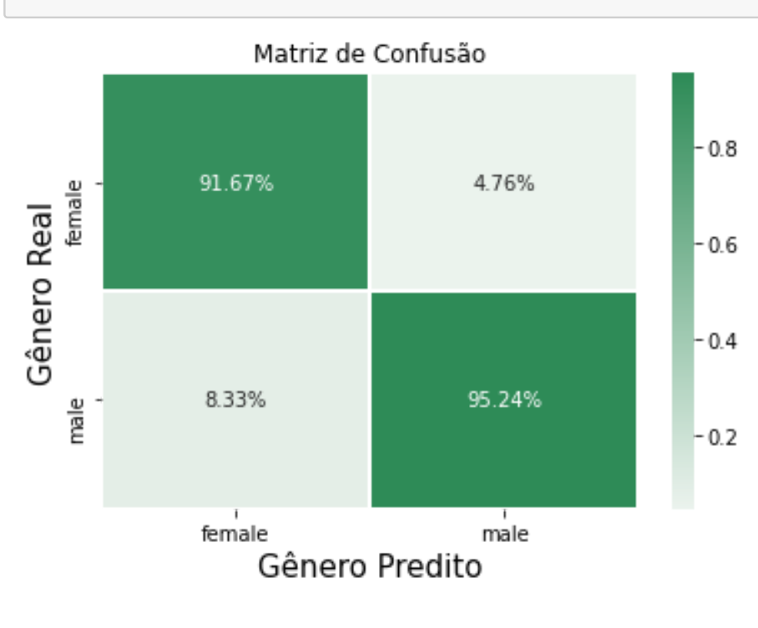
Out[27]:

	female	male	accuracy	macro avg	weighted avg
precision	0.916667	0.952381	0.933333	0.934524	0.934127
recall	0.956522	0.909091	0.933333	0.932806	0.933333
f1-score	0.936170	0.930233	0.933333	0.933201	0.933267
support	23.000000	22.000000	0.933333	45.000000	45.000000

In [28]:
```
c_matrix = confusion_matrix(y_test, test_predictions)

ticklabels = ['female', 'male']
ax = plt.axes()
sns.heatmap((c_matrix/sum(c_matrix)), annot=True,
            xticklabels=ticklabels,
            yticklabels=ticklabels,
            fmt='.02%',
            cmap=sns.light_palette("seagreen", as_cmap=True),
            linewidths=0.2, ax = ax)

ax.set_title('Matriz de Confusão')
plt.xlabel('Gênero Predito', fontsize = 15)
plt.ylabel('Gênero Real', fontsize = 15)
plt.show()
```



Conclusões

Pelo relatório de classificação conseguimos as medidas, por classe, da acurácia, precisão, recall e f1-score. Podemos ver que para as duas classes ("female" e "male"), temos mais de 90% em todas essas métricas no conjunto de teste, o que nos leva a acreditar que o desempenho do modelo está bom.

Como confirmação, podemos ver pela matriz de confusão que realmente o modelo conseguiu generalizar bem de modo a ter menos de 10% para as duas classes no conjunto de teste. Desse modo podemos assumir que não ocorreu um overfitting e que as variáveis escolhidas para realizar a regressão foram corretas.