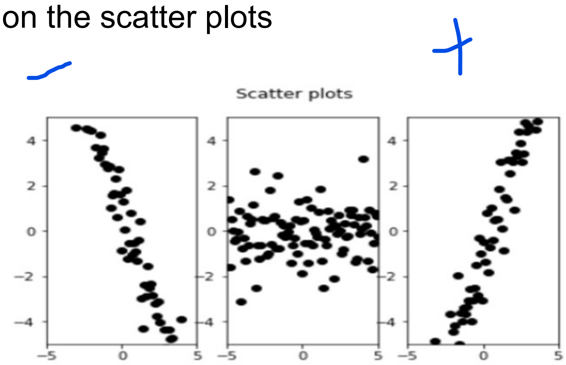
Correlation

- Scatter plots
  - We can observe the association between two variables by the shape of observations on the scatter plots



Correlation

- Correlation coefficients
  - A measurement for linear relationship between two variables
  - Correlation coefficient (r) :

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Correlation

- Test about ρ
  - The correlation coefficient can be tested about $H_0\colon\ \rho = 0$

$$T = \frac{\sqrt{n-2}\cdot r}{\sqrt{1-r^2}} \sim t_{(n-2)}$$

  - Rejection region

$$H_1\colon \rho > 0, \qquad T > t_{(\alpha;n-2)}$$
$$H_1\colon \rho < 0, \qquad T < -t_{(\alpha;n-2)}$$
$$H_1\colon \rho \neq 0, \qquad |T| > t_{(\frac{\alpha}{2};n-2)}$$

한양대학교 HANYANG UNIVERSITY

Correlation

```
sns.pairplot(data, vars, kind='scatter',
             diag_kind='auto', dropna=True )
```

-. *Plot a pair-wise relationship in a dataset*
-. *data : tidy form (variables are in columns)*
-. *vars : variables in a dataset to use*
-. *kind : kind of plot to make ('scatter', 'kde', 'hist', 'reg')*
-. *diag_kind : kind of plot for diagonal subplots*
-. *dropna=True : plot after dropping NaN*

한양대학교 HANYANG UNIVERSITY

Correlation

sns.PairGrid(data, vars, hue, palette,
hue, palette, dropna=True)

-. *Plot a pair-wise relationship in a dataset*
-. *data : tidy form (variables are in columns)*
-. *vars : variables in a dataset to use*
-. *hue: list of variable names to map the colors*
-. *palette : a set of colors for mapping the hue*
-. *diag_kind : kind of plot for diagonal subplots*
-. *dropna=True : plot after dropping NaN*

Correlation

map(kind)

-. *A kind of plot in PairGrid*
-. *map_upper() : in upper triangle area*
-. *map_lower() : in lower triangle area*
-. *map_diag() : in diagonal area*

Correlation

- Practice
  - Plot a pair-wise scatter plot after loading "penguin" data in *Seaborn*.

In [28]:
```
import seaborn as sns
penguins = sns.load_dataset("penguins")

penguins.head()
```

Out[28]:

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	Male
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	Female
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	Female
3	Adelie	Torgersen	NaN	NaN	NaN	NaN	NaN
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	Female

Correlation

- Continue..

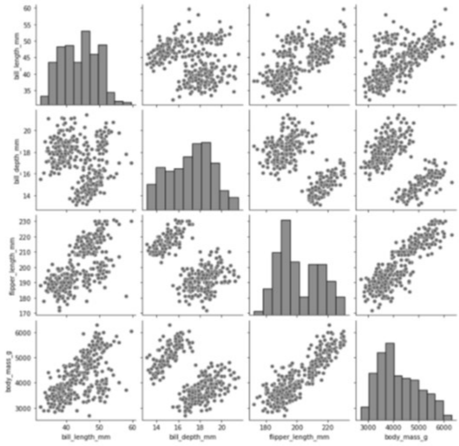
In [42]:
```
g = sns.pairplot(data = penguins
                 ,vars = ['bill_length_mm', 'bill_depth_mm', 'flipper_length_mm', 'body_mass_g']
                 , kind = 'scatter', dropna=True )
```

- Using *hue*:

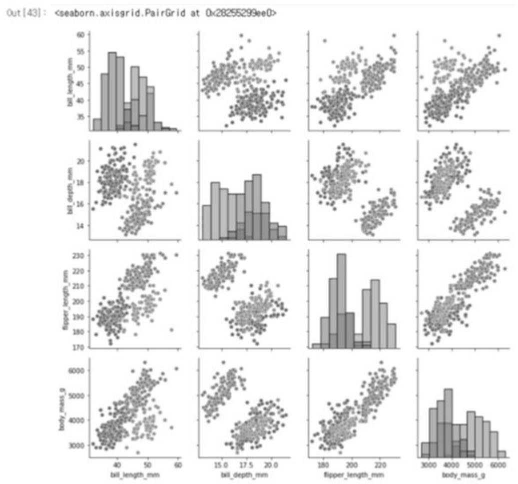
In [43]:
```
g = sns.PairGrid(data = penguins,  hue="species"
                 ,vars = ['bill_length_mm', 'bill_depth_mm', 'flipper_length_mm', 'body_mass_g']
                 , dropna=True )
g.map_upper(sns.scatterplot)
g.map_lower(sns.scatterplot)
g.map_diag(sns.histplot)
```

한양대학교 HANYANG UNIVERSITY

Correlation



vs.



Correlation

pd.corr(method='pearson')

-. *Compute Pearson's correlation coefficient*

stats.pearsonr(var1, var2)

-. *Compute Pearson's correlation coefficient and also return p-value of the correlation coefficient.*

-. var1, var2: two variables to use

Correlation

- Practice
  - Compute correlation coefficients of the variables in the scatter plot and test the correlation coefficient at α=0.05.

```
In [47]: import pandas as pd

         penguins.corr(method='pearson')
```

Out[47]:

	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g
bill_length_mm	1.000000	-0.235053	0.656181	0.595110
bill_depth_mm	-0.235053	1.000000	-0.583851	-0.471916
flipper_length_mm	0.656181	-0.583851	1.000000	0.871202
body_mass_g	0.595110	-0.471916	0.871202	1.000000

한양대학교 HANYANG UNIVERSITY

Correlation

- Practice (continue..)

```
In [53]: from scipy.stats import pearsonr

         penguins2 =penguins.dropna(axis=0, how='any', inplace=False)
         r2, pval = pearsonr(penguins2['bill_length_mm'],penguins2['bill_depth_mm'])

         print('correlation coefficient is ',r2)
         print('p-value is ',pval)

         correlation coefficient is  -0.2286256359130291
         p-value is  2.5282897209444827e-05
```

  - Then, what's your answer?

한양대학교 HANYANG UNIVERSITY

Regression

- Components
  - Independent variable (X)
  - Dependent variable (Y)
- Type
  - Simple regression
  - Multivariate or multiple regression
- Relationship type
  - linear
  - Non-linear

Regression

- Simple linear regression
  - Regression line can be expressed as

$$Y = \alpha + \beta \cdot X + \varepsilon, \qquad \varepsilon \sim N(0, \sigma^2)$$

  - Linear relationship between x and y
  - The residuals are independent
  - The residuals have constant variance
  - The residuals of the model are normally distributed

Regression

- Least Square Error (LSE)
  - To minimize the sum of errors

$$L = \min_{\alpha,\beta} \sum_{i=1}^{n} (y_i - \alpha - \beta \cdot x_i)^2$$

  - The first order partial derivatives are:

$$\frac{\partial L}{\partial \alpha} = -2 \sum_{i=1}^{n} (y_i - \alpha - \beta \cdot x_i)$$

$$\frac{\partial L}{\partial \beta} = -2 \sum_{i=1}^{n} (y_i - \alpha - \beta \cdot x_i) \cdot x_i$$

한양대학교 HANYANG UNIVERSITY

Regression

- Regression coefficients
  - Regression coefficients

$$\hat{\beta} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$$

한양대학교 HANYANG UNIVERSITY

Regression

- Total sum of squares (SST)
  - SST = SSR + SSE

$$SST = \sum_{i=1}^{n} (y_i - \overline{y_i})^2 \qquad \text{(Total)}$$

$$SSR = \sum_{i=1}^{n} (\widehat{y_i} - \overline{y_i})^2 \qquad \text{(Regression part)}$$

$$SSE = \sum_{i=1}^{n} (y_i - \widehat{y_i})^2 \qquad \text{(Error part)}$$

한양대학교 HANYANG UNIVERSITY

Regression

- Coefficients of determination (R^2)

$$R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

  - R^2 ranges between 0 and 1.
  - R^2 indicates how well terms fit a regression line.
  - If R^2 is close to 1, we can conclude that regression model explains the relationship of the data well..

한양대학교 HANYANG UNIVERSITY

Regression

- Tests about β
  - Test about the slope when H_0: $\beta = \beta_0$

$$T = \frac{\hat{\beta}-\beta_0}{\widehat{SE}\ (\hat{\beta})}, \qquad \widehat{SE}\ (\hat{\beta}) = \frac{\hat{\sigma}}{\sqrt{\sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n}}}$$

  - Rejection regions are:

$$H_1:\ \beta > \beta_0, \qquad T \geq t_{(\alpha,n-2)}$$
$$H_1:\ \beta < \beta_0, \qquad T \leq -t_{(\alpha,n-2)}$$
$$H_1:\ \beta \neq \beta_0, \qquad |T| \geq t_{(\frac{\alpha}{2},n-2)}$$

한양대학교 HANYANG UNIVERSITY

Regression

- Tests about α
  - Test about the intercept when H_0: $\alpha = \alpha_0$

$$T = \frac{\hat{\alpha}-\alpha_0}{\widehat{SE}\ (\hat{\alpha})}, \qquad \widehat{SE}\ (\hat{\alpha}) = \hat{\sigma} \cdot \sqrt{\frac{1}{n} + \frac{\bar{x}^2}{(\sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n})}}$$

  - Rejection regions are:

$$H_1:\ \alpha > \alpha_0, \qquad T \geq t_{(\alpha,n-2)}$$
$$H_1:\ \alpha < \alpha_0, \qquad T \leq -t_{(\alpha,n-2)}$$
$$H_1:\ \alpha \neq \alpha_0, \qquad |T| \geq t_{(\frac{\alpha}{2},n-2)}$$

한양대학교 HANYANG UNIVERSITY

Regression

- Mean response
  - Mean response is expressed as $\hat{\alpha} + \hat{\beta}x$
  - Standard error of mean response :

$$\widehat{SE}\ (\hat{\alpha} + \hat{\beta}x) = \hat{\sigma} \cdot \sqrt{\frac{1}{n} + \frac{(x-\bar{x})^2}{(\sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n})}}$$

  - Confidence interval of the mean response :

$$(\hat{\alpha} + \hat{\beta}x) \pm t_{(\frac{\alpha}{2}, n-2)} \cdot \widehat{SE}\ (\hat{\alpha} + \hat{\beta}x)$$

한양대학교 HANYANG UNIVERSITY

Regression

statsmodels.api

-. *Provide a interface for specifying models using formula*

statsmodels.api.ols(y, x, missing='none')
-. *y : 1-do dependent variable*
x: array of independent variables (constant needs to be added) .
missing = 'none': no NaN checking is done
= 'drop' : drop NaN
= 'raise' : error is raised for NaN

한양대학교 HANYANG UNIVERSITY

Regression

.add_constant(x)
-. *Add constant in the array of x*
.fit()
-. *Fit a regression model*
.params()
-. *Return regression parameters*
.t_test()
-. *Return test results about regression parameters*
.get_prediction(x)
-. *Return prediction value about x*

Regression

- Practice
  - Find regression coefficients about 'bill_length' with 'bill_depth' in '*penguins*'.

```
In [82]: import statsmodels.api as sm

         x = sm.add_constant(penguins2['bill_depth_mm'])
         penguins_fit1 = sm.OLS(penguins2['bill_length_mm'], x).fit()

In [83]: params1 = penguins_fit1.params
         print(params1)

         const            54.890854
         bill_depth_mm    -0.634905
         dtype: float64
```

Regression

- Practice
  - Test about α (intercept) and β at α =0.05'.

In [87]:
```
#test
print("alpha:", penguins_fit1.t_test([1,0]))
print("beta:",penguins_fit1.t_test([0,1]))
```

```
alpha:                             Test for Constraints
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
c0            54.8909      2.567     21.380      0.000      49.840      59.941
==============================================================================
beta:                              Test for Constraints
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
c0            -0.6349      0.149     -4.273      0.000      -0.927      -0.343
==============================================================================
```

Regression

- Practice
  - Make prediction with the regression line.

In [98]:
```
ypred2 = penguins_fit1.get_prediction(x)
result2 = ypred2.summary_frame(alpha=0.05).round(4)
result2.head()
```

Out[98]:

	mean	mean_se	mean_ci_lower	mean_ci_upper	obs_ci_lower	obs_ci_upper
0	43.0181	0.3707	42.2889	43.7473	32.5042	53.5321
1	43.8435	0.2943	43.2646	44.4224	33.3389	54.3481
2	43.4626	0.3174	42.8381	44.0870	32.9554	53.9697
4	42.6372	0.4313	41.7887	43.4857	32.1143	53.1601
5	41.8118	0.5882	40.6548	42.9688	31.2596	52.3640

Regression

.lmplot(x, y, data, ci)
-. *Plot regression plot*

x : independent variable name
y : dependent variable name
data : the dataset to use
ci : integer in [0, 100] or None

Regression

- Practice
  - Plot a scatter plot with the predicted values.

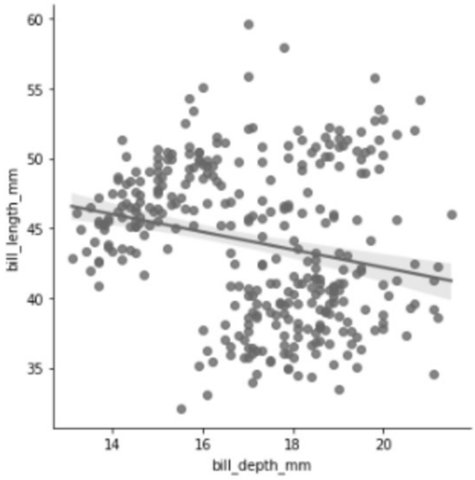
```
In [101]: import seaborn as sns

sns.lmplot(x='bill_depth_mm',y='bill_length_mm', data=penguins2, ci=95 )
```

Regression

- (continue..)

Out[101]: <seaborn.axisgrid.FacetGrid at 0x28254f31af0>



Regression

- Practice : compute R2.

In [102]:
```
penguins_fit1.summary()
```

$$Adj\ R2 = 1 - \frac{(n-1)\frac{SSE}{SST}}{n-p-1}$$

Out[102]:

OLS Regression Results

Dep. Variable:	bill_length_mm	R-squared:	0.052
Model:	OLS	Adj. R-squared:	0.049
Method:	Least Squares	F-statistic:	18.26
Date:		Prob (F-statistic):	2.53e-05
Time:		Log-Likelihood:	-1028.8
No. Observations:	333	AIC:	2062.
Df Residuals:	331	BIC:	2069.
Df Model:	1		
Covariance Type:	nonrobust		