Midterm Economic Modelling and Simulation

Instructions

- There is one and only one correct answer per question
- Fill out the answers sheet: only that sheet will be graded
- You will need to return the question sheet, but whatever you write on it will be disregarded
- All questions are worth the same
- Correct answers are worth +1
- Wrong answers are worth -1
- If you are unsure about an answer, leave it blank and it will be worth zero
- For the sake of readability, the semicolon `;` has been used to combine several lines of code into a single one, when appropriate

Questions

1. How would you measure the accuracy of a classification problem (e.g. diagnosing a disease) when using logistic regression?

- a) Dividing the true positives by the true negatives
- b) Dividing the true positives by the false positives
- c) Dividing the sum of the true positives and the true negatives by the total
- d) Dividing the sum of the false positives and the true negatives by the sum of the true negatives

2. Choose the right import statement of the 'car_crashes' dataset from the `seaborn` package:

- a) `import seaborn as sns; df = sns.load_dataset('car_crashes')`
- b) `from seaborn import car_crashes`
- c) `import seaborn as pd; car_crashes = sns.get_dataset_names('pandas')`
- d) `import diamonds from seaborn`

We will now work with a dataset called `diamonds`. It contains properties of diamonds such as the `depth`, the `price`, the `color`, etc. of a collection of diamonds.

3. In order to get the column names of the 'diamonds' dataset, the proper command is:

- a) `diamonds.columns`
- b) `diamonds.names`
- c) `diamonds.values`

- d) `diamonds[columns]`

Below you can find a sample of the `diamonds` dataset:

```
       carat        cut color clarity  depth  table  price     x     y     z
0       0.23      Ideal     E     SI2   61.5   55.0    326  3.95  3.98  2.43
1       0.21    Premium     E     SI1   59.8   61.0    326  3.89  3.84  2.31
2       0.23       Good     E     VS1   56.9   65.0    327  4.05  4.07  2.31
3       0.29    Premium     I     VS2   62.4   58.0    334  4.20  4.23  2.63
4       0.31       Good     J     SI2   63.3   58.0    335  4.34  4.35  2.75
...      ...        ...   ...     ...    ...    ...    ...   ...   ...   ...
53935   0.72      Ideal     D     SI1   60.8   57.0   2757  5.75  5.76  3.50
53936   0.72       Good     D     SI1   63.1   55.0   2757  5.69  5.75  3.61
53937   0.70  Very Good     D     SI1   62.8   60.0   2757  5.66  5.68  3.56
53938   0.86    Premium     H     SI2   61.0   58.0   2757  6.15  6.12  3.74
53939   0.75      Ideal     D     SI2   62.2   55.0   2757  5.83  5.87  3.64

[53940 rows x 10 columns]

Index(['carat', 'cut', 'color', 'clarity', 'depth', 'table', 'price', 'x', 'y',
       'z'],
      dtype='object')
```

Figure 1: Excerpt from `diamonds` dataset

4. Which command would subset the dataset 'diamonds' when the price is lower than 300:

- a) `diamonds.query("price > 300")`
- b) `diamonds.query("cut == Ideal")`
- c) `diamonds.query("price == 300")`
- d) `diamonds.query("price < 300")`

5. Now we decided to look at the price statistics when the `cut` variable equals `Ideal` using the following filter: `diamonds.query("cut == 'Ideal'")['price'].describe()`. Based on the output, which is the median price when cut is 'Ideal'?

```
count    21551.000000
mean      3457.541970
std       3808.401172
min        326.000000
25%        878.000000
50%       1810.000000
75%       4678.500000
max      18806.000000
Name: price, dtype: float64
```

Figure 2: Output of the `describe()` command

- a) `3456.5`
- b) `326.0`
- c) `1810.0`
- d) `4678.5`

6. We want to understand how many diamonds have color `E`, how many have color `J`, and so on. Which command achieves this?

- a) `diamonds.color.value_counts()`
- b) `diamonds.value_counts().color`
- c) `diamonds.color_values()`
- d) `diamonds.value_counts(color)`

7. Based on the summary statistic of Question 5, what would be the value corresponding to the 30th percentile?

- a) Less than `326`
- b) Between `326.0` and `878.0`
- c) Between `4678.5` and `18806.0`
- d) Between `878.0` and `1810.0`

8. Now we want to subset the dataset by the 2 first indexes. The command would be:

- a) `diamonds.loc[0:2]`
- b) `diamonds.query("0:2")`
- c) `diamonds.iloc[0:2]`
- d) `diamonds - 0:2`

9. Create a column named `volume` based on the dimensions `x`, `y` and `z` of the diamonds:

- a) `diamonds["volume"] = diamonds["x"] ** diamonds["y"] ** diamonds['z']`
- b) `diamonds["volume"] = product(diamonds["x"] * diamonds["y"] * diamonds['z'])`
- c) `diamonds.volume = diamonds["x"] + diamonds["y"] + diamonds['z']`
- d) `diamonds = diamonds.assign(volume = lambda df: df.x * df.y * df.z)`

10. Obtain a descriptive analysis of the `price` column that includes the number of elements, quantiles, etc:

a. `diamonds.price.stats()`
b. `diamonds.price.describe()`
c. `diamonds[‘price’].analyse()`
d. `diamonds.price`

11. The following is one row extracted from the `diamonds` dataset. What would be the result of the sum of columns `x` and `y` for this row?

- a) `NaN`
- b) `3.95`
- c) `0`
- d) `6.38`

12. Which of the following charts display a histogram of the `price` variable?

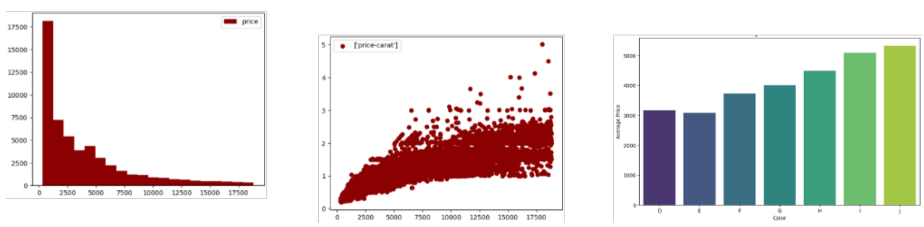


Figure 3: Output of the `describe()` command

- a) the left one
- b) the one in the center
- c) the right one
- d) none

13. Based on the above histogram, which is the most frequent price range?

- a) Between 7500 and 10000
- b) Between 17500 and 20000
- c) Between 5000 and 7500
- d) Between 0 and 2500

14. Which of the following best defines the term “Least Squares” in the context of statistical modeling?

- a) A method for minimizing the sum of squared differences between observed and predicted values
- b) A technique used to maximize the variance between data points

- c) A statistical approach focused on maximizing the absolute differences between observed and predicted values
- d) A method that prioritizes minimizing the maximum deviation between observed and predicted values

15. Choose the right definition of the the LeastSquares function that returns the intercept and slope after getting as arguments an `x` and `y` vectors:

- a)

```
def LeastSquares(xs, ys):
      mean_x = np.mean(xs)
      mean_y = np.mean(ys)
      cov = np.dot(xs - mean_x, ys - mean_y) / len(xs)
      slope = cov
      inter = mean_y - slope * mean_x
      return inter, slope
```

- b)

```
def LeastSquares(xs, ys):
      mean_x = np.mean(xs)
      var_x = np.var(xs)
      mean_y = np.mean(ys)
      inter = mean_x / mean_y
      slope = mean_x - var_x
      return inter, slope
```

- c)

```
def LeastSquares(xs, ys):
      mean_x = np.mean(xs)
      var_x = np.var(xs)
      mean_y = np.mean(ys)
      cov = np.dot(xs - mean_x, ys - mean_y) / len(xs)
      slope = cov / var_x
      inter = mean_y - slope * mean_x
      return inter, slope
```

- d) None are correct

16. The above function is applied to the 'carat' and 'price' columns in order to find a correlation between them: `inter, slope = LeastSquares(diamonds.carat, diamonds.price)`. Based on the above estimated intercept & slope for the columns `carat` and `price`, let's build a column called `fit_carat` that contains the model fit for `price` for the datapoints in the dataset:

- a) `diamonds["fit_carat"] = inter * diamonds['carat'] * slope`

- b) `diamonds["fit_carat"] = inter * diamonds['carat'] + slope`
- c) `diamonds["fit_carat"] = inter + slope * diamonds['carat']`
- d) `diamonds["fit_carat"] = inter + slope + diamonds['carat']`

17. Which of the following is correct:

- a) The residuals are the differences between the observed values and the mean of the dependent variable.
- b) The residuals are the sum of the observed values and the predicted values
- c) The residuals of a linear model are the differences between the observed values and the values predicted
- d) The residuals are the differences between the predicted values and the mean of the independent variable.

18. Now we want to define a function that is capable to estimate the residuals of the linear fit. Select the correct definition:

- a)
```
def Residuals(xs, ys, inter, slope):
    xs = np.asarray(xs)
    ys = np.asarray(ys)
    res = ys - (inter + slope * xs)
```
- b)
```
def Residuals(xs, inter, slope):
    xs = np.asarray(xs)
    res = (inter + slope * xs)
    return res
```
- c)
```
def Residuals(xs, inter):
    xs = np.asarray(xs)
    res = xs - inter
    return res
```
- d) None is correct

19. Select the correct code-snippet that would allow to draw a scatterplot that includes `price` in the y-axis, and `carat` in the x-axis; plotting both the actual dataset and the fits:

- a)
```
plt.scatter(x="carat", y="price", data=diamonds, color="blue", label="carat")
plt.scatter(x="carat", y="fit_carat", data=diamonds, color="cyan", marker="s")
plt.xlabel("carat")
plt.ylabel("price")
plt.legend()
```
- b)
```
plt.scatter(x="price", y="carat", data=diamonds, color="blue", label="carat")
plt.scatter(x="fit_carat", y="carat", data=diamonds, color="cyan", marker="s")
```

```
plt.xlabel("carat")
plt.ylabel("price")
plt.legend()
```

- c)
```
plt.line(x="price", y="cut", data=diamonds, color="blue", label="carat")
plt.line(x="fit_carat", y="cut", data=diamonds, color="cyan", marker="s")
plt.xlabel("carat")
plt.ylabel("price")
plt.legend()
```
- d) None is correct

20. What function would you use to store the above `plt` object as a .png figure?

- a) `plt.storefig(f"{directory}/scatter.png")`
- b) `df.savefig(f"{directory}/scatter.png")`
- c) `plt.storefig()`
- d) `plt.savefig(f"{directory}/scatter.png")`

21. Choose which is the correct definition of an outlier in the data: (B)

- a) An outlier is any value in a dataset that is larger than the mean, indicating a superior significance in the analysis.
- b) An outlier is an observation that significantly deviates from the overall pattern of a dataset, often falling far outside the expected range of values.
- c) An outlier is an observation that perfectly fits the trend of a dataset, contributing to the overall consistency of the data.
- d) An outlier is the most common value in a dataset, representing the typical or average observation.

22. The following command obtains the 10 largest price numbers of the diamonds dataset when `cut` variable equals to `Ideal`: `diamonds.query("cut == ‘Ideal’").nlargest(10, ‘price’).loc[:, ‘price’])`. Would you say, based on the output, that there is an outlier?

- a) No, because all the values are above 18000, meaning there is no deviation from the reference
- b) No, because the values are in the 18000-20000 range
- c) Yes, because the first value is one magnitude order larger than the others
- d) Yes, because the last value is significantly lower than the first one

23. The variance is a summary statistic used to:

```
0        199999
27747     18806
27746     18804
27741     18791
27738     18787
27735     18780
27734     18779
27732     18768
27730     18760
27728     18757
```

Figure 4: Output

- a) Describe the central tendency of the distribution.
- b) Describe the median of the distribution.
- c) Describe the errors of a distribution.
- d) Describe the variability of a distribution.

24. What's the correct formula of the mean statistic?

- a) $\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$
- b) $\bar{x} = \frac{1}{n-1} \sum_{i=1}^{n} x_i$
- c) $\bar{x} = \frac{1}{n} \sum_{i=1}^{n-1} x_i$
- d) $\bar{x} = \frac{1}{n} \sum_{i=0}^{n} y_i$

25. The standard deviation can be conceived as: (B)

- a) The mean of the absolute differences from the mean.
- b) The squared root of the variance.
- c) The triple power of the variance.
- d) The mean divided by the population size.

In the following questions we will be using the dataset `mpg` from the seaborn package. Here is a sample:

26. We want to create a binary variable where 1 represents to USA as 'origin' while 0 represents any other possibility. This variable will be encoded under the name 'is_usa'. The correct prompt is:

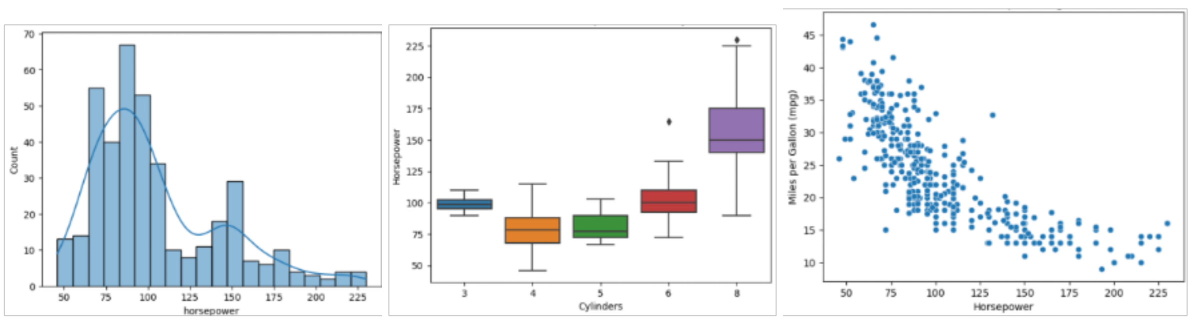
- a) `mpg['is_usa'] = (mpg.origin == 'usa') * 1`
- b) `mpg['is_usa'] = 'usa' * 1`
- c) `mpg['is_usa'] = 'usa' * 0`

mpg	cylinders	displacement	horsepower	weight	acceleration	model_year	origin	name
18.0	8	307.0	130.0	3504	12.0	70	usa	chevrolet chevelle malibu
15.0	8	350.0	165.0	3693	11.5	70	usa	buick skylark 320
18.0	8	318.0	150.0	3436	11.0	70	usa	plymouth satellite
16.0	8	304.0	150.0	3433	12.0	70	usa	amc rebel sst
17.0	8	302.0	140.0	3449	10.5	70	usa	ford torino
...	...	...	...	...	...	...	...	...
27.0	4	140.0	86.0	2790	15.6	82	usa	ford mustang gl
44.0	4	97.0	52.0	2130	24.6	82	europe	vw pickup
32.0	4	135.0	84.0	2295	11.6	82	usa	dodge rampage
28.0	4	120.0	79.0	2625	18.6	82	usa	ford ranger
31.0	4	119.0	82.0	2720	19.4	82	usa	chevy s-10

Figure 5: mpg dataset

- d) `mpg[‘is_usa’] = (mpg.origin = ‘usa’) not in 'country'`

27. Select the plot that shows the distribution of ‘horsepower’ for each ‘cylinder’:



- a) Left
- b) Center
- c) Right
- d) None is correct

28. Based on the above chart, what type of statistical relationship appears to be the one between the variable `horsepower` and `cylinder`?

- a) Non-linearly growing: horsepower increases with the number of cylinders.
- b) Linearly decreasing: horsepower decreases with the numbers of cylinders since they are less efficient in distributing the power.
- c) Normally distributed: horsepower reaches its peak at 80 cylinders.
- d) Non-linearly growing: horsepower decreases from 50 to 225 as the number of cylinder increases.

29. Is there a linear relationship between ‘mpg’ & ‘horsepower’ based on the charts ?

- a) Yes, data follows a perfect linear trend.
- b) No, data seems to follow a non-linear declining trend.
- c) No, data follows an exponential uptrend.
- d) Yes, since there is a constant decline on the miles per gallon as the horsepower increases.

30. What distinguishes multiple linear regression from simple linear regression?

- a) Multiple linear regression involves more than one independent variable, while simple linear regression involves only one.
- b) Multiple linear regression is always more accurate than simple linear regression.
- c) Simple linear regression can handle categorical variables, while multiple linear regression cannot.
- d) Multiple linear regression does not work with small datasets.