Table of Contents

[References 1](#_Toc119390777)

[Conjoint Analysis 1](#_Toc119390778)

[Fractional Factorial Designs 1](#_Toc119390779)

[Level 1](#_Toc119390780)

[Part Worth 1](#_Toc119390781)

[Relative Importance 1](#_Toc119390782)

[Utility 2](#_Toc119390783)

[Survey Design 2](#_Toc119390784)

[Level Part Worth Utility 3](#_Toc119390785)

[Attribute Importance 3](#_Toc119390786)

[Predicting User Preferences 3](#_Toc119390787)

## References

<https://www.informit.com/articles/article.aspx?p=2350028> (Dr. Tom Miller)

<https://www.linkedin.com/pulse/conjoint-analysis-simple-python-implementation-prajwal-sreenivas>

## Conjoint Analysis

Conjoint analysis is a marketing technique for discovering consumer preferences. Conjoint analysis can help to predict consumer behaviour and it can help with product positioning. The product is described by a number of attributes and each **attribute** has several **levels**.

### Fractional Factorial Designs

Product designs can be created using an even matching of each attribute with all others to understand preferences (utility). In this manner, it is not necessary to study an exhaustive set of combinations.

### Level

Products or services have different attributes. Attributes may have more than one option and these options are called levels. For example, when developing a new car, an important consideration is the fuel economy attribute. **Levels** for this attributemight include **18KM/L*,* 15KM/L,** or **13KM/L**.

### Part Worth

Part worth is the consumer preference at each level for every attribute of the product.

### Relative Importance

Relative importance is how much difference an attribute can add to the utility of a product.

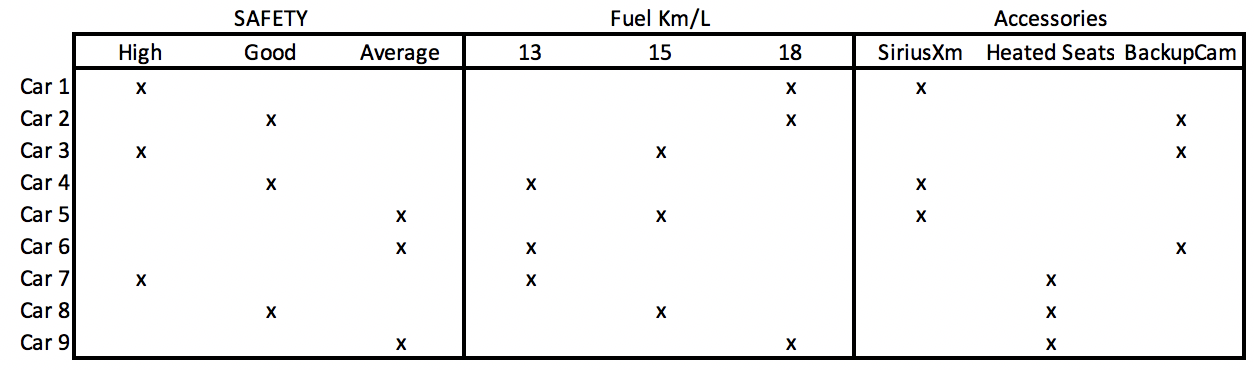
### Utility

Utility is the level of individual consumer preference.

## Survey Design

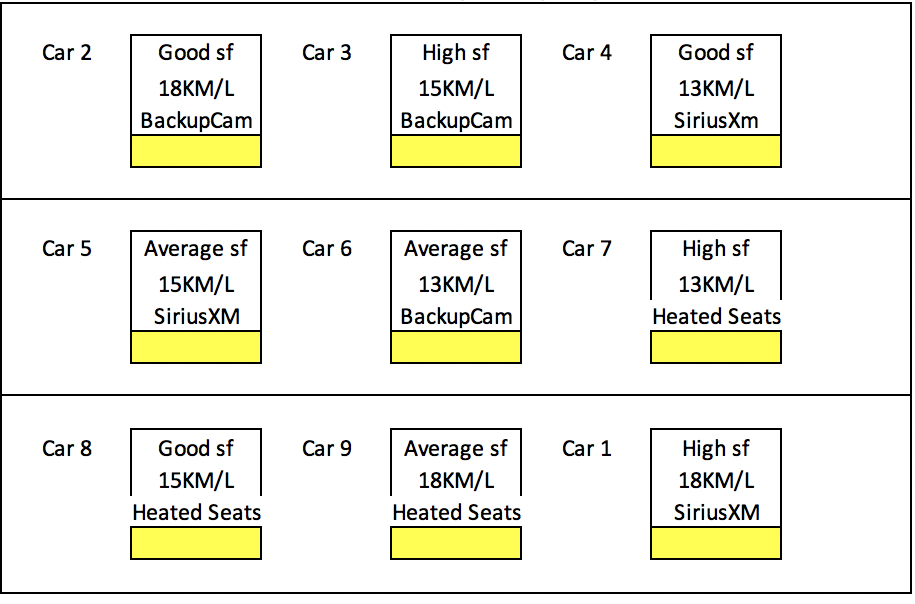
Figure 1 identifies 9 mutually exclusive attribute combinations for a car. There are actually 3x3x3=27 potential choices. However, we can design a simple survey with 9 combinations where each attribute level is represented with the same amount. In other words, each attribute appears in two of the nine survey choices. In this case, each respondent’s ranking of these 9 combinations can reveal their preferences for all 27 combinations.

Figure 1: Fractional Factorial Survey Design



Based on the combinations identified, a survey can be created. The survey in Figure 2 asks respondents to rank their preferred car options between 1 and 9 where 9 is the best.

Figure 2: Car Survey



### Level Part Worth Utility

Level part worth is a utility value that quantifies a user’s preference. For our use case, most survey respondents likely prefer higher fuel economy.

The part worth score for each level = Average of the sum of all rankings which contain the level.

(See Figure 3)

### Attribute Importance

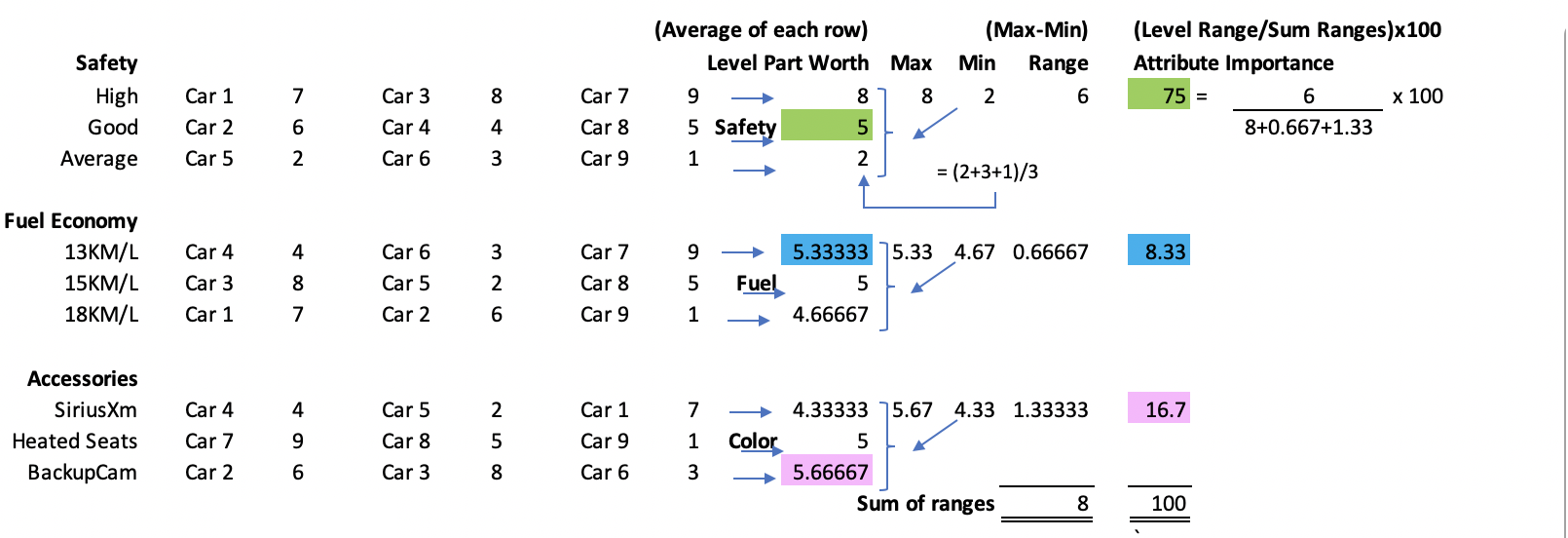
The importance score rates an attribute relative to others. In our case, it is possible that many drivers would may rate vehicle safety as a more important attribute than any car accessories.

The importance score for an attribute

= (Max(level) – Min(level))/(Sum of level ranges for each attribute)

Figure 3 shows a spreadsheet which enables the calculation of the part worth utilities and attribute importance scores.

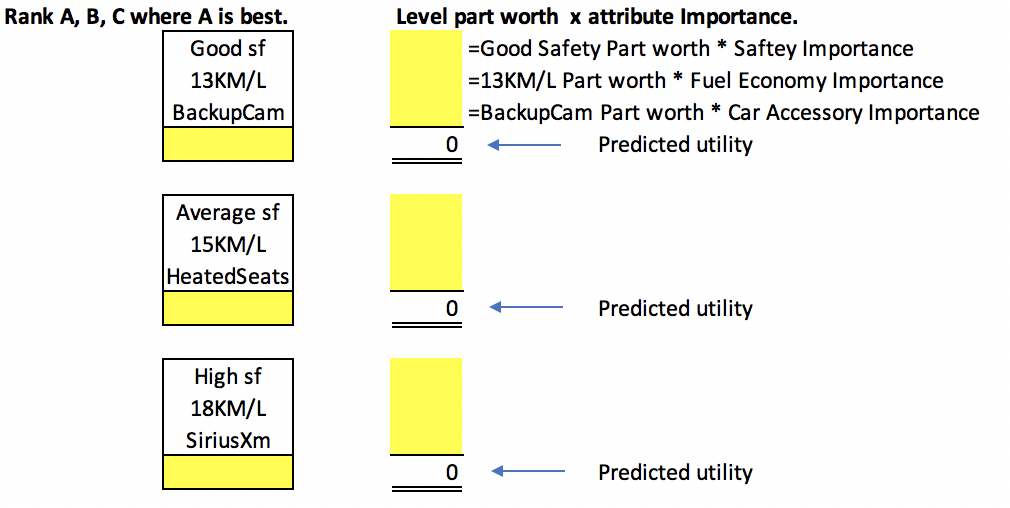
Figure 3: Calculating part worth utilities and Attribute Importance



## Predicting User Preferences

The level part worth and attribute importance scores can now be used to make predictions about a user’s preferences (see Figure 4).

Figure 4: Comparing Actual Preferences with Predicted Utility



For this next set of exercises, follow the steps in the CarUtiliityPrediction.xls file.

Exercise 1 (3 marks)

Take the survey and rank your car preferences between 1 and 9 where 9 is the best. Show a screenshot of your completed survey here.

|  |
| --- |
| Graphical user interface, application  Description automatically generated |

Exercise 2 (3 marks)

After completing the survey, show a screenshot of your level part worth and attribute importance values in your spreadsheet here:

|  |
| --- |
| Graphical user interface, application, table  Description automatically generated |

Exercise 3 (6 marks)

Next, rate the three car options presented in step 3. Then use the level part worth and attribute importance scores to calculate the predicted responses. Show a screenshot of the spreadsheet results here for your actual and predicted values.

|  |
| --- |
| Graphical user interface, application  Description automatically generated |

Exercise 4 (4 marks)

Starting with the code below, replace the part-worth values with the values that you calculated for your preferences.

|  |
| --- |
| from matplotlib import pyplot as plt  import pylab as plb  plb.rcParams['font.size'] = 16  def mk\_groups(data):  try:  newdata = data.items()  except:  return  thisgroup = []  groups = []  for key, value in newdata:  newgroups = mk\_groups(value)  if newgroups is None:  thisgroup.append((key, value))  else:  thisgroup.append((key, len(newgroups[-1])))  if groups:  groups = [g + n for n, g in zip(newgroups, groups)]  else:  groups = newgroups  return [thisgroup] + groups  def add\_line(ax, xpos, ypos):  line = plt.Line2D([xpos, xpos], [ypos + .1, ypos],  transform=ax.transAxes, color='black')  line.set\_clip\_on(False)  ax.add\_line(line)  def label\_group\_bar(ax, data):  groups = mk\_groups(data)  xy = groups.pop()  x, y = zip(\*xy)  ly = len(y)  xticks = range(1, ly + 1)  ax.bar(xticks, y, align='center')  ax.set\_xticks(xticks)  ax.set\_xticklabels(x, Rotation=0, )  ax.set\_xlim(.5, ly + .5)  ax.yaxis.grid(True)  scale = 1. / ly  for pos in range(ly + 1):  add\_line(ax, pos \* scale, -.1)  ypos = -.2 # Adjust this to shift the bottom labels  while groups:  group = groups.pop()  pos = 0  for label, rpos in group:  lxpos = (pos + .5 \* rpos) \* scale  ax.text(lxpos, ypos, label, ha='center', transform=ax.transAxes, rotation=70,  color='red')  add\_line(ax, pos \* scale, ypos)  pos += rpos  add\_line(ax, pos \* scale, ypos)  ypos -= .1  dataDict = {  # Set attributes and level part worths for the attribute.  "Car Safety": { "High":1, "Good":1, "Average":-1},  "Fuel": { "18km/l":1, "15km/l":1, "13km/l":-1},  "Accessories": { "Roadside Assist":1, "BackupCam":1, "Heated Seats":-1},  }  fig = plt.figure(figsize=(13,10))  ax = fig.add\_subplot(1, 1, 1)  label\_group\_bar(ax, dataDict)  plt.title("Part Worths")  plt.show() |

Show the resulting plot here:

|  |
| --- |
| Chart, bar chart  Description automatically generated |

Exercise 5 (4 marks)

Start with the following code, replace the importance values for the attributes with your importance values that were calculated in the spreadsheet.

|  |
| --- |
| import matplotlib.pyplot as plt  carAttributes = ['Saftey', 'Fuel Economy', 'Accessories']  importanceLevels = [2,1,0]  plt.bar(carAttributes, importanceLevels)  plt.xticks(rotation=75)  plt.title("Car Attribute Importance")  plt.show() |

Show your bar plot here:

|  |
| --- |
| Chart, bar chart  Description automatically generated |