In [2]:
```
import numpy as np
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.pipeline import Pipeline
import pandas as pd

%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('ggplot')

from basis_expansions import (Binner, Polynomial,
                              LinearSpline, CubicSpline,
                              NaturalCubicSpline)
```

```
/Users/matthewdrury/anaconda/envs/py36/lib/python3.6/site-packages/ma
tplotlib/__init__.py:913: UserWarning: axes.color_cycle is deprecated
and replaced with axes.prop_cycle; please use the latter.
  warnings.warn(self.msg_depr % (key, alt_key))
```

Basis Expansions Vignette

Introduction

Installation

To install, clone the repository and run `setup.py` :

```
git clone https://github.com/madrury/basis-expansions
cd basis-expansions
pip install .
```

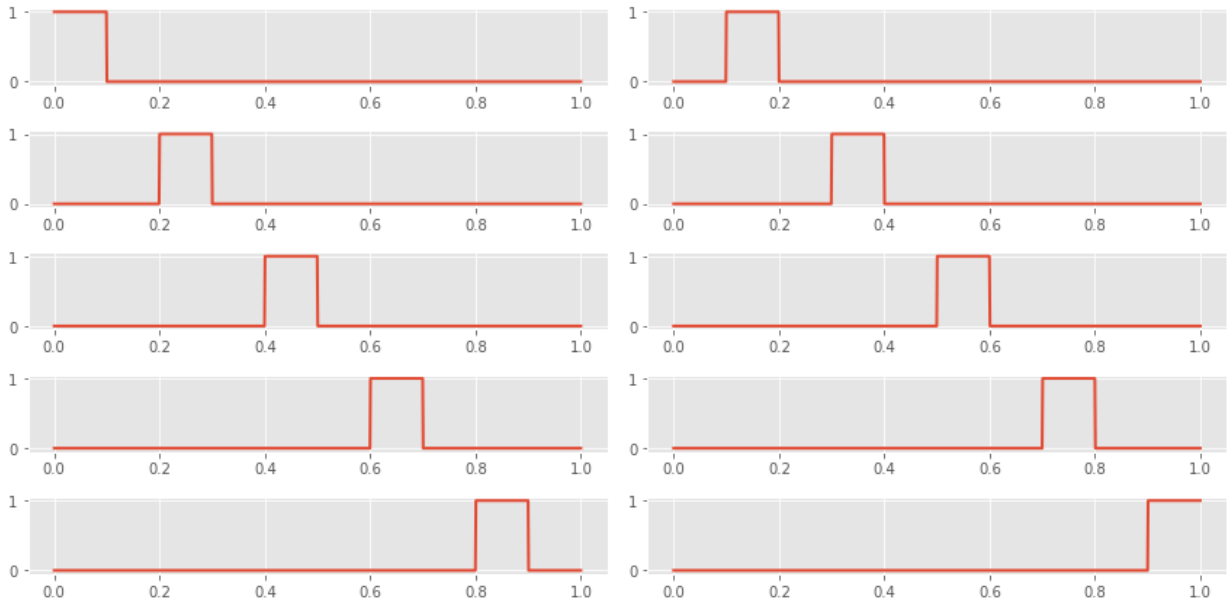
Quick Start

Basis Expansions

In [11]:
```
def plot_basis(axs, basis_expansion, t):
    basis = basis_expansion.fit_transform(t)
    for idx, ax in enumerate(axs.flatten()):
        ax.plot(t, basis[:, idx])
```

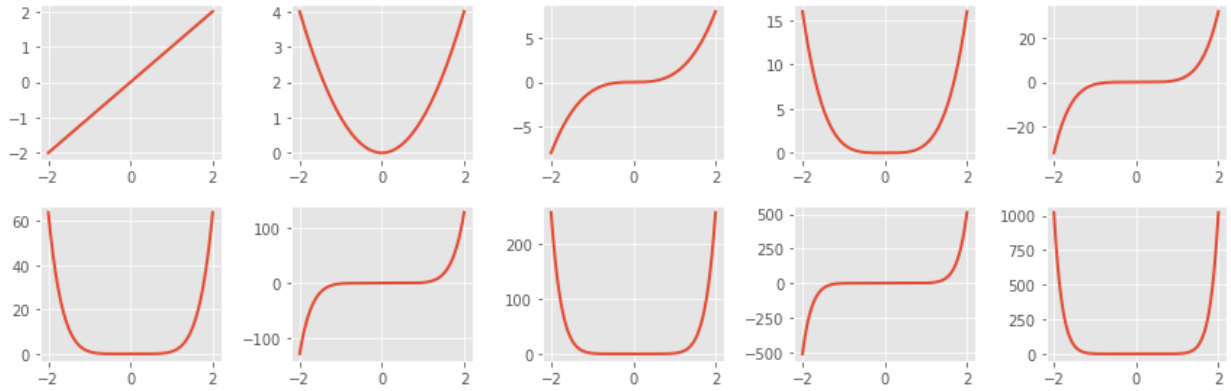
Binning Expansion

In [18]:
```
binner = Binner(min=0, max=1, n_cuts=9)

fig, ax = plt.subplots(5, 2, figsize=(12, 6))
plot_basis(ax, binner, np.linspace(0, 1, num=1000))
fig.tight_layout()
```



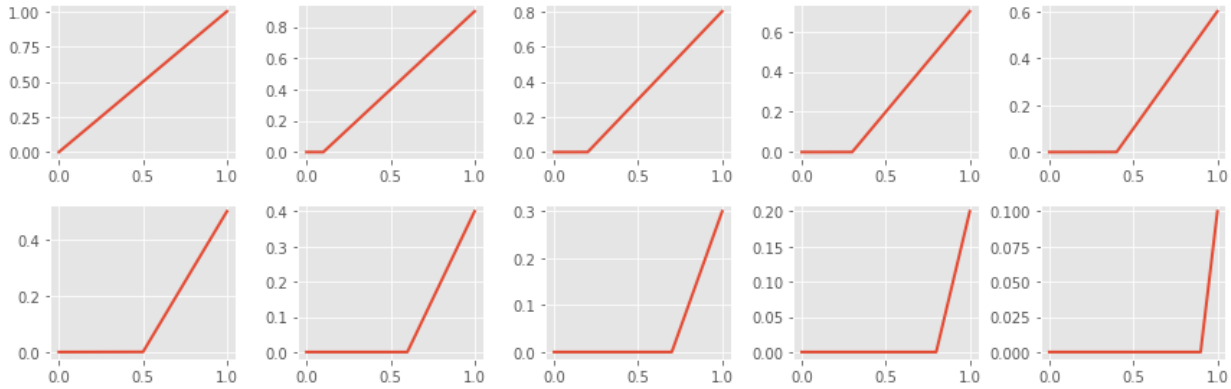
Polynomial Expansions

In [22]:
```
poly = Polynomial(degree=10)

fig, ax = plt.subplots(2, 5, figsize=(12, 4))
plot_basis(ax, poly, np.linspace(-2, 2, num=1000))
fig.tight_layout()
```



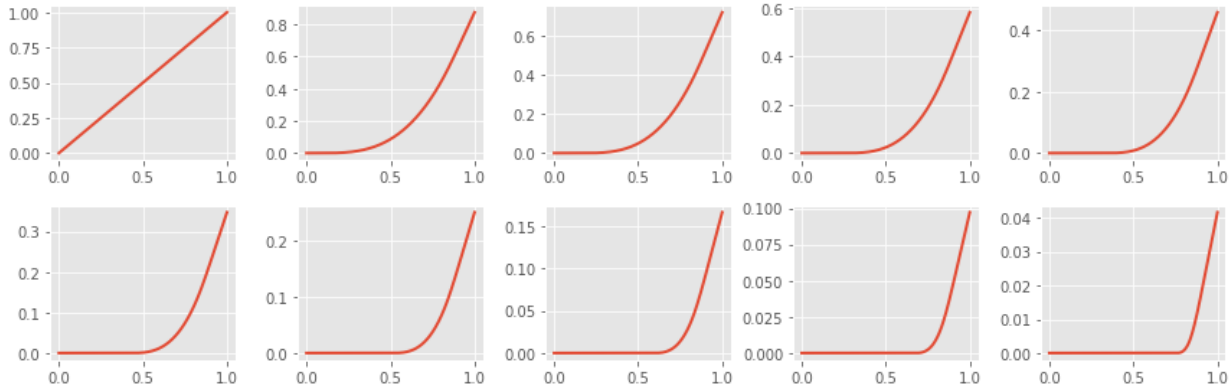
Piecewise Linear Expansions

In [27]:
```
lin = LinearSpline(min=0, max=1, n_knots=9)

fig, ax = plt.subplots(2, 5, figsize=(12, 4))
plot_basis(ax, lin, np.linspace(0, 1, num=1000))
fig.tight_layout()
```



Natural Cubic Spline Expansions

In [30]:
```
cub = NaturalCubicSpline(min=0, max=1, n_knots=11)

fig, ax = plt.subplots(2, 5, figsize=(12, 4))
plot_basis(ax, cub, np.linspace(0, 1, num=1000))
fig.tight_layout()
```



In [ ]:
```

```