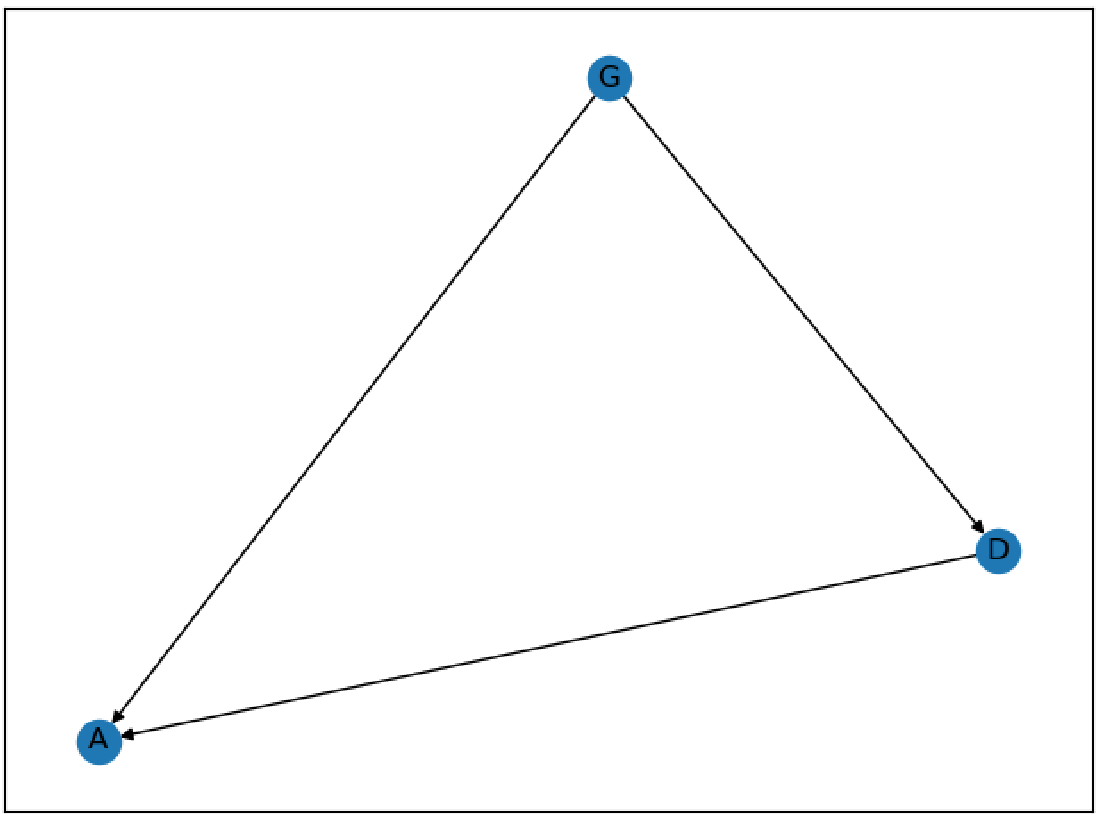
Question 1. **(20 points)** The data in *Data/NWOGrants.csv* are outcomes for scientific funding applications for the Netherlands Organization for Scientific Research (NWO) from 2010–2012 (see van der Lee and Ellemers).

These data have a very similar structure to the UC Berkeley Admissions data discussed in lecture.

- Include an image of a directed acyclic graph (DAG) for this data that represents the relationship between the variables. **Clearly describe what the nodes in your graph represent**. **Do not include the number of applications** in your DAG or as a predictor in your linear model.

  Note : *Refer to **dag_tutorial.ipynb** for instructions on how to include a DAG in your Jupyter notebook using Python.*

In [193]:
```
graph = nx.DiGraph()
graph.add_edges_from([("G", "D"), ("D", "A"), ("G", "A")])
nx.draw_networkx(graph, arrows=True)
plt.tight_layout()
```



There are 3 nodes, "G" for the gender of an individual, "A" for the number of awards an individual gets, and "D" for the discipline the individual is in. G influences A because it is possible that one gender was awarded more than the other based on historical practices. G influences D because it is possible that certain genders tend to favor specific disciplines. D influences A because it is likely that certain disciplines provide varying amounts of opportunity and potential to win awards.

Use a Binomial GLM to estimate the TOTAL causal effect of reported gender on grant awards.

Notes:
- You **must provide justification** for the prior distributions used in your model to receive full credit. **Limit your justification to no more than 5 sentences.** - You can also use logistic regression for your solution but this will require disaggregating the data.

In [194]:
```
df = pd.read_csv("Data/NWOGrants.csv", header=0)
A = df.awards
N = df.applications
G = np.where(df.gender == "f", 0, 1) # gender
D = pd.Categorical(df.discipline).codes

with pm.Model() as m_GA_total:
    alpha = pm.Normal("alpha", 0, 1, shape=2)
    p = pm.Deterministic("p", pm.math.invlogit(alpha[G]))
    modA = pm.Binomial("A", n=N, p=p, observed=A)
    idata_GA_total = pm.sample()
az.summary(idata_GA_total, var_names=["~p"], kind="all")
```

```
Auto-assigning NUTS sampler…
Initializing NUTS using jitter+adapt_diag…
Sequential sampling (2 chains in 1 job)
NUTS: [alpha]
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
Sampling 2 chains for 1_000 tune and 1_000 draw iterations (2_000 + 2_000 draws total) took 6 seconds.
We recommend running at least 4 chains for robust computation of convergence diagnostics
```

```
Out[194]:             mean     sd  hdi_5.5%  hdi_94.5%  mcse_mean  mcse_sd  ess_bulk  \
          alpha[0] -1.732  0.081    -1.851     -1.593      0.002    0.001    1904.0
          alpha[1] -1.527  0.065    -1.633     -1.423      0.002    0.001    1694.0

                    ess_tail  r_hat
          alpha[0]    1462.0    1.0
          alpha[1]    1507.0    1.0
```

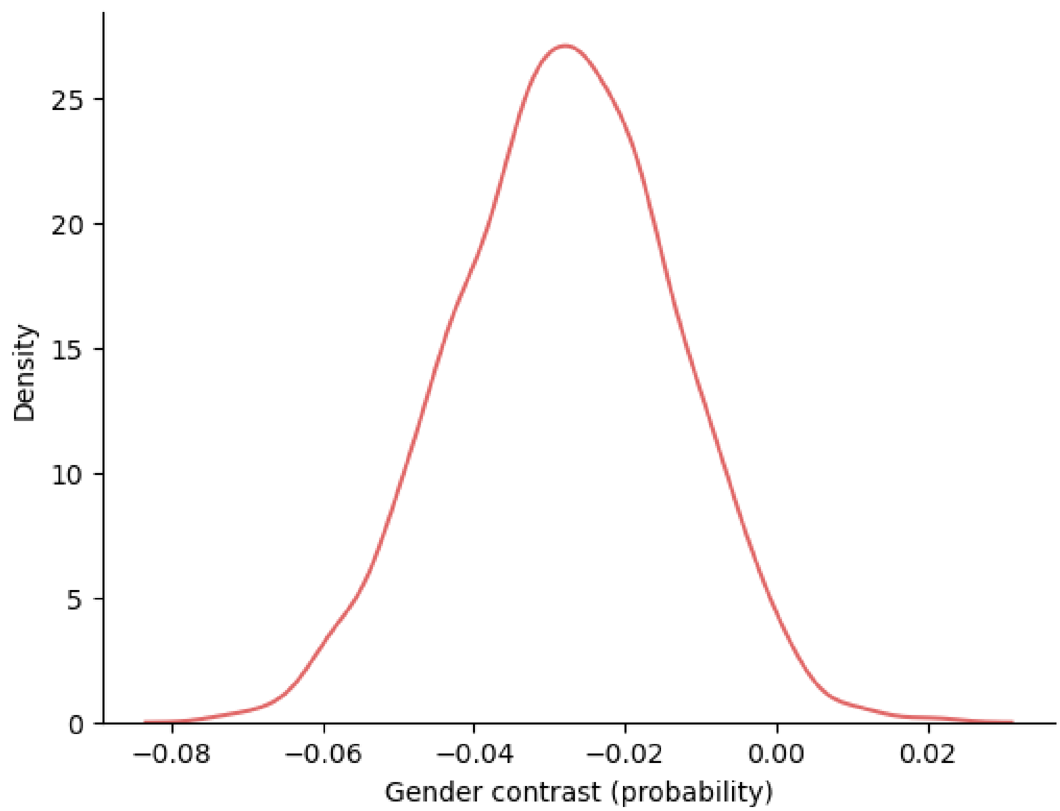
"alpha" has a mean of 0 and standard deviation of 1 to reflect that there are no major initial effects from G on A. The gender value is transformed into a probability using logit.

Provide a gender contrast plot and a written interpretation of the plot. **Limit your interpretation of the plot to no more than 3 sentences.**

```
In [195]: post1 = az.extract_dataset(idata_GA_total)
          probA_G0 = logistic(post1.alpha[0]) # female
          probA_G1 = logistic(post1.alpha[1]) # male
          diff_prob = probA_G0 - probA_G1 # female - male

          sns.kdeplot(diff_prob, color="#e06666")
          plt.xlabel("Gender contrast (probability)")
          sns.despine()
```

```
/tmp/ipykernel_1140/2360435162.py:1: FutureWarning: extract_dataset has been deprecated, please use ext
  post1 = az.extract_dataset(idata_GA_total)
```



Males are favored for awards according to this contrast model, since the peak of this curve lies at approx. -0.01. This indicates that the probability for males was larger than females if the difference between (female

- male) is negative. The area under the curve from probability 0.00 to 0.02 demonstrate that there are much smaller percentage of differences where female probabilty is greater than male probability, such that the difference is positive. Values of probability beneath the curve from -0.08 to 0.00 have negative differences, where males have a higher probability.

Question 2. **(22 points)** Now estimate the AVERAGE DIRECT causal effect of gender on grant awards using a counterfactual simulation as demonstrated in lecture.

Note : *Refer to the counterfactual simulation for UC Berkeley admission outcomes from the lecture on Bernoulli/Binomial Generalized Linear Models for guidance. Its ok to just write code and generate plots for this question. Question 3 focuses on interpreting your results.*

There seems to be an effect of gender on grant awards, based on the summarized data. Each of the averages fall between the HPDI interval. r-hat values are extremely close to 1 suggesting reliable estimates.

In [196]:
```
# model for counterfactual simulation

with pm.Model() as m_GA_cf:
    # pm.MutableData allow for simulated interventions on variable
    d_G = pm.MutableData("gender", G.astype("int64"))
    d_D = pm.MutableData("discipline", D.astype("int64"))
    d_A = pm.MutableData("awards", A.astype("int64"))
    d_N = pm.MutableData("num", N.astype("int64")) # number of applications

    alpha = pm.Normal("alpha", 0, 1, shape=(2, len(np.unique(A))))
    p = pm.Deterministic("p", pm.math.invlogit(alpha[d_G, d_D]))
    mod_A = pm.Binomial("A", d_N, p, observed=A, shape=d_D.shape)

    idata_GA_cf = pm.sample()

az.summary(idata_GA_cf, var_names="~p", kind="all")
```

```
Auto-assigning NUTS sampler…
Initializing NUTS using jitter+adapt_diag…
Sequential sampling (2 chains in 1 job)
NUTS: [alpha]



<IPython.core.display.HTML object>


<IPython.core.display.HTML object>


<IPython.core.display.HTML object>


<IPython.core.display.HTML object>


Sampling 2 chains for 1_000 tune and 1_000 draw iterations (2_000 + 2_000 draws total) took 7 seconds.
We recommend running at least 4 chains for robust computation of convergence diagnostics
```

Out[196]:

	mean	sd	hdi_5.5%	hdi_94.5%	mcse_mean	mcse_sd	ess_bulk	\
alpha[0, 0]	-0.966	0.327	-1.536	-0.506	0.005	0.004	5263.0	
alpha[0, 1]	-1.709	0.241	-2.083	-1.328	0.003	0.003	5747.0	
alpha[0, 2]	-1.392	0.191	-1.697	-1.090	0.003	0.002	5609.0	
alpha[0, 3]	-1.209	0.256	-1.619	-0.815	0.004	0.003	5327.0	
alpha[0, 4]	-2.012	0.199	-2.290	-1.651	0.003	0.002	5993.0	
alpha[0, 5]	-1.086	0.343	-1.635	-0.537	0.005	0.004	4860.0	
alpha[0, 6]	-0.825	0.593	-1.744	0.121	0.008	0.009	6185.0	
alpha[0, 7]	-2.004	0.144	-2.220	-1.755	0.002	0.001	5703.0	
alpha[0, 8]	-1.235	0.286	-1.714	-0.800	0.004	0.003	5069.0	
alpha[0, 9]	0.019	0.969	-1.579	1.528	0.013	0.025	5137.0	
alpha[0, 10]	0.000	1.019	-1.656	1.607	0.013	0.026	5901.0	
alpha[0, 11]	0.017	1.002	-1.525	1.718	0.014	0.026	5043.0	
alpha[0, 12]	0.009	1.018	-1.675	1.559	0.013	0.026	5965.0	
alpha[0, 13]	0.009	0.977	-1.499	1.637	0.014	0.026	4546.0	
alpha[0, 14]	-0.004	0.996	-1.533	1.672	0.015	0.025	4219.0	
alpha[0, 15]	-0.021	0.965	-1.554	1.502	0.013	0.024	5664.0	
alpha[0, 16]	-0.005	0.966	-1.503	1.550	0.015	0.024	4199.0	
alpha[1, 0]	-0.972	0.244	-1.331	-0.543	0.003	0.003	5575.0	
alpha[1, 1]	-1.101	0.176	-1.401	-0.840	0.002	0.002	6028.0	
alpha[1, 2]	-1.737	0.180	-2.017	-1.451	0.002	0.002	5953.0	
alpha[1, 3]	-1.902	0.275	-2.346	-1.475	0.004	0.003	4924.0	
alpha[1, 4]	-1.435	0.161	-1.697	-1.188	0.002	0.002	5893.0	
alpha[1, 5]	-1.387	0.203	-1.705	-1.076	0.003	0.002	5370.0	
alpha[1, 6]	-0.943	0.254	-1.326	-0.523	0.004	0.003	4017.0	
alpha[1, 7]	-1.687	0.132	-1.893	-1.475	0.002	0.001	5897.0	
alpha[1, 8]	-1.621	0.203	-1.951	-1.306	0.003	0.002	6066.0	
alpha[1, 9]	-0.010	1.013	-1.657	1.531	0.013	0.032	5925.0	
alpha[1, 10]	-0.025	1.014	-1.682	1.525	0.015	0.029	4860.0	
alpha[1, 11]	0.009	0.992	-1.535	1.602	0.014	0.028	4827.0	
alpha[1, 12]	0.028	1.037	-1.559	1.712	0.014	0.026	5654.0	
alpha[1, 13]	-0.020	1.052	-1.751	1.590	0.013	0.029	6180.0	
alpha[1, 14]	0.018	1.003	-1.535	1.658	0.015	0.026	4391.0	
alpha[1, 15]	0.001	1.024	-1.592	1.622	0.015	0.025	4825.0	
alpha[1, 16]	-0.034	1.018	-1.612	1.563	0.015	0.029	4590.0	

	ess_tail	r_hat
alpha[0, 0]	1372.0	1.00
alpha[0, 1]	1336.0	1.00
alpha[0, 2]	1202.0	1.00
alpha[0, 3]	1409.0	1.00
alpha[0, 4]	1442.0	1.00
alpha[0, 5]	1636.0	1.00
alpha[0, 6]	1393.0	1.00
alpha[0, 7]	1260.0	1.00
alpha[0, 8]	1545.0	1.00
alpha[0, 9]	1315.0	1.00
alpha[0, 10]	1515.0	1.00
alpha[0, 11]	1307.0	1.00
alpha[0, 12]	1420.0	1.00
alpha[0, 13]	1345.0	1.00
alpha[0, 14]	1419.0	1.00
alpha[0, 15]	1524.0	1.00
alpha[0, 16]	1254.0	1.00

```
alpha[1, 0]     1294.0   1.00
alpha[1, 1]     1373.0   1.00
alpha[1, 2]     1448.0   1.01
alpha[1, 3]     1542.0   1.00
alpha[1, 4]     1592.0   1.00
alpha[1, 5]     1238.0   1.00
alpha[1, 6]     1246.0   1.00
alpha[1, 7]     1485.0   1.00
alpha[1, 8]     1607.0   1.00
alpha[1, 9]     1168.0   1.00
alpha[1, 10]    1300.0   1.00
alpha[1, 11]    1397.0   1.00
alpha[1, 12]    1555.0   1.00
alpha[1, 13]    1269.0   1.00
alpha[1, 14]    1280.0   1.00
alpha[1, 15]    1414.0   1.00
alpha[1, 16]    1275.0   1.00
```

After estimating the average direct causal effect of gender on grant awards: 1. Plot the average direct causal effect as a contrast plot from the results of the counterfactual simulation.

In [197]:
```python
# model for counterfactual simulation
with pm.Model() as m_GA_cf:
# pm.MutableData allow for simulated interventions on variable
    d_D = pm.MutableData("discipline", D.astype("int64"))
    d_G = pm.MutableData("gender", G.astype("int64"))
    d_N = pm.MutableData("num", N.astype("int64"))
    alpha = pm.Normal("alpha", 0, 1, shape=(2, len(np.unique(D)))) # unique disciplines
    p = pm.Deterministic("p", pm.math.invlogit(alpha[d_G, d_D]))
    mod_A = pm.Binomial("A", d_N, p, observed=A, shape=d_D.shape)
    idata_GA_cf = pm.sample()

total_apps = df.applications.sum()
apps_per_discipline = df.groupby("discipline")["applications"].sum()

with m_GA_cf: # Female
# simulate as if all apps from women
    pm.set_data({
         "discipline": np.repeat(range(len(apps_per_discipline)), apps_per_discipline),
         "num": np.repeat(1, total_apps),
         "gender": np.repeat(0, total_apps)
    })
    cf_A_G0 = pm.sample_posterior_predictive(idata_GA_cf)

with m_GA_cf:
    pm.set_data({
         "discipline": np.repeat(range(len(apps_per_discipline)), apps_per_discipline),
         "num": np.repeat(1, total_apps),
         "gender": np.repeat(1, total_apps)
    })
    cf_A_G1 = pm.sample_posterior_predictive(idata_GA_cf)
```

```python
# extract the posterior_predictive samples from the Inference Data object to simplify access
admitted_G0 = az.extract(cf_A_G0, group="posterior_predictive")
admitted_G1 = az.extract(cf_A_G1, group="posterior_predictive")

# compute admit rate for each simulation by gender
num_samples = admitted_G0.A.values.shape[1]
admit_rate_G0 = admitted_G0.A.values.sum(axis=1) / num_samples
admit_rate_G1 = admitted_G1.A.values.sum(axis=1) / num_samples


# plot contrast
sns.kdeplot(admit_rate_G0 - admit_rate_G1, color="#e06666")
plt.xlabel("effect of gender perception")
plt.xlim(left=-0.2)
plt.axvline(0, linestyle="--")
sns.despine();
```

```
Auto-assigning NUTS sampler…
Initializing NUTS using jitter+adapt_diag…
Sequential sampling (2 chains in 1 job)
NUTS: [alpha]
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
Sampling 2 chains for 1_000 tune and 1_000 draw iterations (2_000 + 2_000 draws total) took 6 seconds.
We recommend running at least 4 chains for robust computation of convergence diagnostics
Sampling: [A]
```

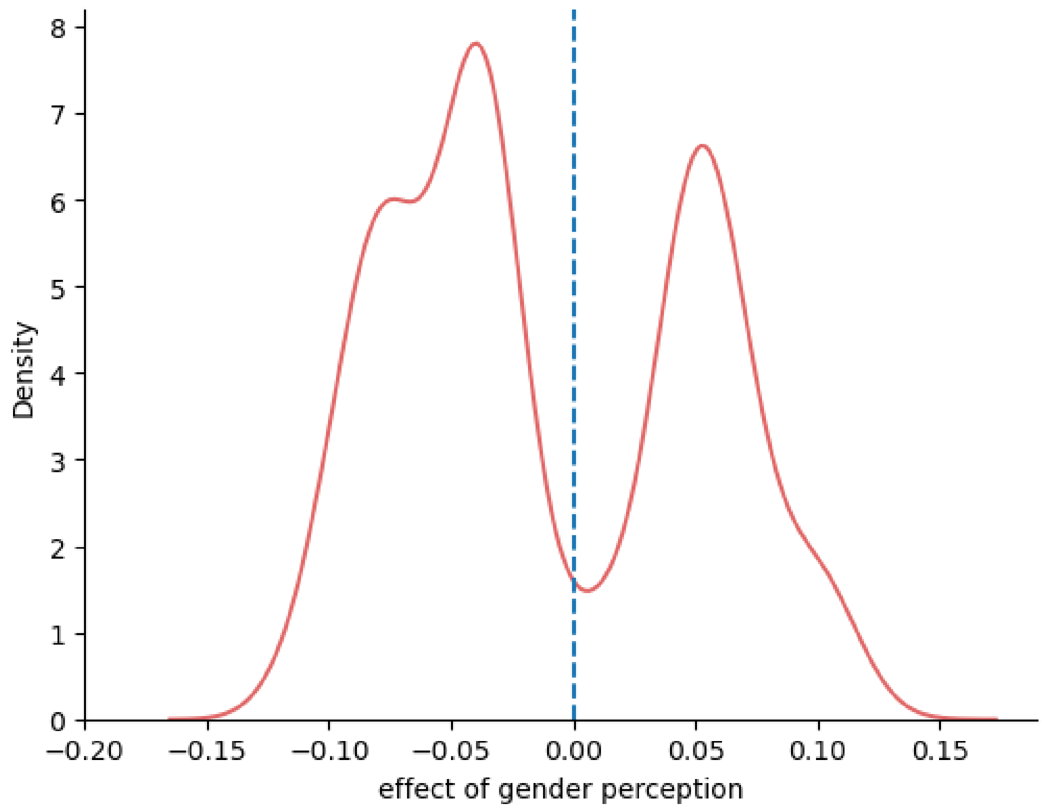
```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
Sampling: [A]
```

```
&lt;IPython.core.display.HTML object&gt;
```

```
&lt;IPython.core.display.HTML object&gt;
```



2. Plot the DIRECT causal effect of reported gender by discipline, **weighting each discipline** in proportion to the number of applications in the observed data. *Refer to the lecture code for an example of how the thickness of the distribution curves can be determined in proportion to the number of applications*

- **Add a legend** to your plot to label the discipline distributions. You can find documentation for adding a legend to your plot here. Specify a `label` each time you plot a contrast distribution and call the `matplotlib.pyplot.legend()` function before displaying the full plot to display the legend.

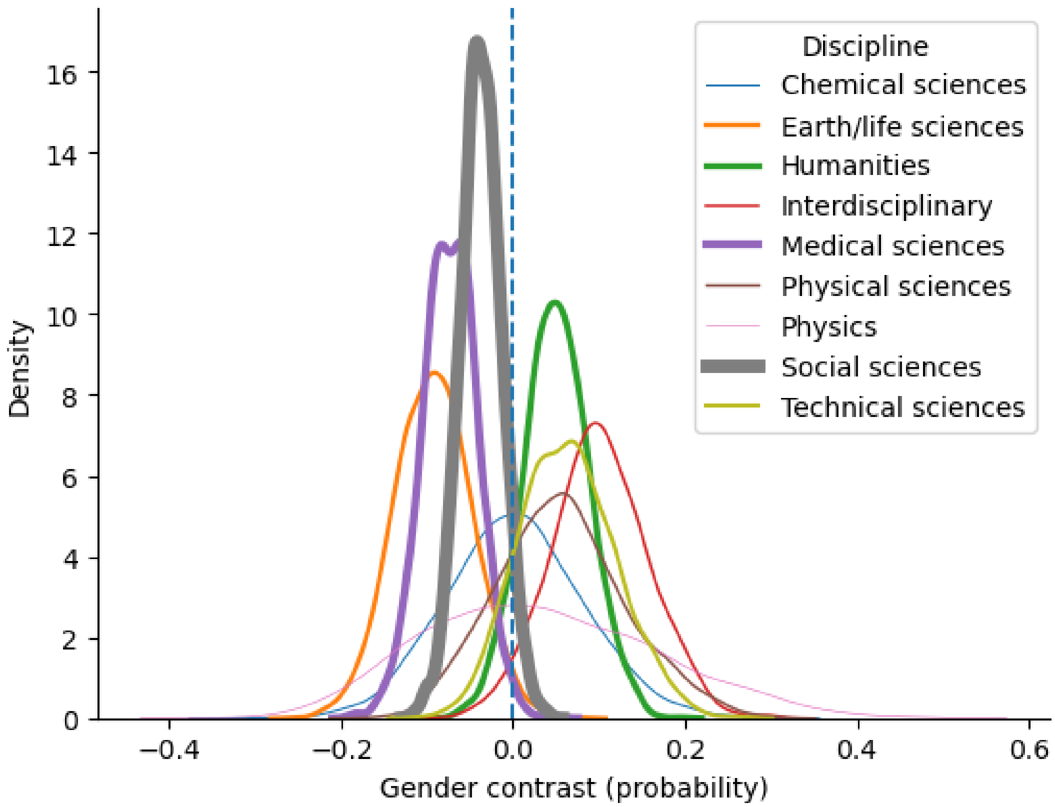
In [198]:
```
w = apps_per_dept / total_apps
w = w / max(w)
my_data = az.extract_dataset(idata_GA_cf)
```

```python
probA = logistic(my_data.alpha)
labels = df.groupby("discipline").sum().index.tolist() # store labels

for i in range(len(apps_per_discipline)):
    # gender contrasts by department
    probA_G0_D = probA[0][i]
    probA_G1_D = probA[1][i]
    diff_prob_D = probA_G0_D - probA_G1_D
    _ = sns.kdeplot(diff_prob_D, linewidth=5 * w[i], label=labels[i])

plt.axvline(x=0, linestyle="--")
plt.xlabel("Gender contrast (probability)")
plt.legend(title="Discipline",loc='upper right') # Show labels
sns.despine()
```

```
/tmp/ipykernel_1140/3669619632.py:3: FutureWarning: extract_dataset has been deprecated, please use extr
  my_data = az.extract_dataset(idata_GA_cf)
/tmp/ipykernel_1140/3669619632.py:12: FutureWarning: Series.__getitem__ treating keys as positions is de
  _ = sns.kdeplot(diff_prob_D, linewidth=5 * w[i], label=labels[i])
/tmp/ipykernel_1140/3669619632.py:12: FutureWarning: Series.__getitem__ treating keys as positions is de
  _ = sns.kdeplot(diff_prob_D, linewidth=5 * w[i], label=labels[i])
/tmp/ipykernel_1140/3669619632.py:12: FutureWarning: Series.__getitem__ treating keys as positions is de
  _ = sns.kdeplot(diff_prob_D, linewidth=5 * w[i], label=labels[i])
/tmp/ipykernel_1140/3669619632.py:12: FutureWarning: Series.__getitem__ treating keys as positions is de
  _ = sns.kdeplot(diff_prob_D, linewidth=5 * w[i], label=labels[i])
/tmp/ipykernel_1140/3669619632.py:12: FutureWarning: Series.__getitem__ treating keys as positions is de
  _ = sns.kdeplot(diff_prob_D, linewidth=5 * w[i], label=labels[i])
/tmp/ipykernel_1140/3669619632.py:12: FutureWarning: Series.__getitem__ treating keys as positions is de
  _ = sns.kdeplot(diff_prob_D, linewidth=5 * w[i], label=labels[i])
/tmp/ipykernel_1140/3669619632.py:12: FutureWarning: Series.__getitem__ treating keys as positions is de
  _ = sns.kdeplot(diff_prob_D, linewidth=5 * w[i], label=labels[i])
/tmp/ipykernel_1140/3669619632.py:12: FutureWarning: Series.__getitem__ treating keys as positions is de
  _ = sns.kdeplot(diff_prob_D, linewidth=5 * w[i], label=labels[i])
/tmp/ipykernel_1140/3669619632.py:12: FutureWarning: Series.__getitem__ treating keys as positions is de
  _ = sns.kdeplot(diff_prob_D, linewidth=5 * w[i], label=labels[i])
```



Discipline
Chemical sciences
Earth/life sciences
Humanities
Interdisciplinary
Medical sciences
Physical sciences
Physics
Social sciences
Technical sciences
Density
Gender contrast (probability)
0
2
4
6
8
10
12
14
16
−0.4
−0.2
0.0
0.2
0.4
0.6

Question 3. **(8 points)** Considering the total effect (Question 1) and the direct effects (Question 2) of reported gender, what causes seem to contribute to the differences between women and men in award rate in this dataset?

It is not necessary to say whether or not there is evidence of discrimination. Simply explain how the direct effects you have estimated make sense (or not) of the total effect. **Limit your response to no more than 10 sentences.**

There are causes (that aren't specifically known) that can favor females to males and vice versa. The data from total effect reports that more men are favored for awards since the difference in proability (female - male < 0). The direct effects suggest that there is favorability evident for both female and males. In some cases, (female - male > 0) suggesting favorability for females, and others (female - male < 0) suggesting favorability for males. Therefore, we have evidence of bias being in favor of both genders (from the direct effect model) instead of only male as suggested in the total effect model. This makes sense because when you are just analyzing direct effect, it would appear that males are awarded more. However, when analyzing the total effect, which takes any other factors into consideration, the observation that men are awarded more becomes less obvious.

What explanations might be unaccounted for regarding differences in award outcomes observed in this dataset? As part of your explanation, **provide at least one potential unobserved confound** that might impact the statistical inference performed in this problem. **Limit your response to no more than 3 sentences.**

One confound could be the network that is available to each gender. It is possible men may have more connections in research during the time this data was observed (around 2012).

In [199]:
```
grader.check("q3.1")
```

Out[199]:
```
q3.1 results: All test cases passed!
```