```
from google.colab import files
uploaded = files.upload()
```

```
import pymc as pm
import matplotlib.pyplot as plt
import arviz as az
import pandas as pd
from scipy import special, stats
import numpy as np
import seaborn as sns
```

4. (30) You have been asked by the government of Bangladesh to determine whether the use of contraceptives by women in Bangladesh varies by district. The data come from surveys conducted in Bangladesh [2] (a) Develop three models for these data: pooled, unpooled, and hierarchical for all districts to predict usage of contraceptives. Use only district and age.centered as predictor variables. For each model briefly explain your choice of priors. (b) For each model evaluate the sampling performance and discuss your findings. (c) Plot the posterior distributions for the parameters for each model and discuss what they show regarding the question posed by WHO. (d) Plot each of the predictions with age.centered on the x-axis and the expected proportion of women using contraception on the y-Axis with overlaid plots for each of the districts, as appropriate. Briefly explain these results as you will report them to the government of Bangladesh.

```
bang = pd.read_csv('bangladesh.csv', header=0)
```

```
bang
```

	woman	district	use.contraception	living.children	age.centered	urban
0	1	1	0	4	18.4400	1
1	2	1	0	1	-5.5599	1
2	3	1	0	3	1.4400	1
3	4	1	0	4	8.4400	1
4	5	1	0	1	-13.5590	1
...	...	...	...	...	...	...
1929	1930	61	0	4	14.4400	0
1930	1931	61	0	3	-4.5599	0
1931	1932	61	0	4	14.4400	0
1932	1933	61	0	1	-13.5600	0

(a) Develop three models for these data: pooled, unpooled, and hierarchical for all districts to predict usage of contraceptives. Use only district and age.centered as predictor variables. For each model briefly explain your choice of priors.

```
# Pooled model

use = pd.Categorical(bang['use.contraception'])
#age = pd.Categorical(bang['age.centered'])
age = bang.loc[:, 'age.centered'].values
district = pd.Categorical(bang['district'])
#coords = {"district": district.unique(), "age": age.codes, "obs_id": np.arange(age.size)}
coords = {"district": district.unique(), "obs_id": np.arange(age.size)}

with pm.Model(coords=coords) as pooled:
  #age_idx = pm.Data("age_idx", age.codes, dims="obs_id", mutable = True)

  beta0 = pm.Normal("beta0", 0.0, sigma=10.0)
  beta1 = pm.Normal("beta1", 0.0, sigma=10.0)

  μ = beta0 + pm.math.dot(age, beta1)
```

```
  #μ = beta[age_idx]
  θ = pm.Deterministic("θ", pm.math.sigmoid(μ))

  y = pm.Bernoulli("y", p=θ, observed=use.codes, dims="obs_id")
  #y = pm.Bernoulli("y", p=θ, observed=use, dims="obs_id")

  trace_pooled = pm.sample(1000, cores=4, random_seed=1234, tune=1000)
```

In the pooled model I developed for contraceptive use based on age, I chose relatively uninformative priors to let the data primarily inform the results. I set normal distributions with a mean of 0 for both the intercept and slope, indicating no initial bias towards any particular age effect. The standard deviation of 10 gives a wide possible range, showcasing my uncertainty about the true parameter values. This approach ensures that my model's conclusions rely heavily on the data and not on prior assumptions.

```
# un-pooled model
age_centered = bang['age.centered'].values
use_contraception = bang['use.contraception'].values
districts = pd.Categorical(bang['district']).codes  # Convert district to categorical codes

coords = {
    "district": np.arange(len(np.unique(districts))),  # Number of unique districts
    "obs_id": np.arange(len(use_contraception))  # Number of observations
}

with pm.Model(coords=coords) as unpooled_model:
    # Mutable data container for district
    district_idx = pm.Data("district_idx", districts, dims="obs_id")

    # Normal priors for the district effect
    beta_district = pm.Normal("beta_district", mu=0.0, sigma=10.0, dims="district")

    # Normal prior for the effect of age
    beta_age = pm.Normal("beta_age", mu=0.0, sigma=10.0)

    # Model equation
    theta = beta_district[district_idx] + beta_age * age_centered
```

```
    # HalfStudentT distribution for the error term provides robustness
    sigma = pm.HalfStudentT('sigma', nu=2, sigma=8)

    # Likelihood function assuming Bernoulli distribution for the binary outcome
    y = pm.Bernoulli("y", p=pm.math.sigmoid(theta), observed=use_contraception, dims="obs_id")

# Sampling from the model
with unpooled_model:
    unpooled_trace = pm.sample(1000, tune=1000, target_accept=0.95)
```

In the un-pooled model, I assigned individual intercepts to each district with a broad normal prior, reflecting my neutral expectation and allowing for variability. The age effect also has a neutral prior. To handle outliers, I used a HalfStudentT distribution for the error term. This model captures both the unique effects of each district and the influence of age on contraceptive use.

```
#heirarchical model

age_centered = bang['age.centered'].values
use_contraception = bang['use.contraception'].values
districts = pd.Categorical(bang['district']).codes

# Define coordinates
coords = {
    "district": np.unique(districts),
    "obs_id": np.arange(len(use_contraception))
}

with pm.Model(coords=coords) as hierarchical_model:
    district_idx = pm.Data("district_idx", districts, dims="obs_id")

    # Hyperpriors for the group means
    a = pm.Normal("a", mu=0.0, sigma=5.0)
    b = pm.Normal("b", mu=0.0, sigma=1.0)

    # Hyperpriors for the group standard deviations
```

```
    sigma_a = pm.HalfStudentT('sigma_a', nu=2, sigma=9)
    sigma_b = pm.HalfStudentT('sigma_b', nu=2, sigma=9)

    # Varying intercepts and slopes for each district
    a_district = pm.Normal("a_district", mu=a, sigma=sigma_a, dims="district")
    b_district = pm.Normal("b_district", mu=b, sigma=sigma_b, dims="district")

    # Model equation
    theta = a_district[district_idx] + b_district[district_idx] * age_centered

    # Model error
    sigma_y = pm.HalfStudentT("sigma_y", nu=2, sigma=8)

    # Likelihood
    y = pm.Bernoulli("y", p=pm.math.sigmoid(theta), observed=use_contraception, dims="obs_id")

# Sampling
with hierarchical_model:
    hierarchical_trace = pm.sample(1000, tune=1000, target_accept=0.95)
```

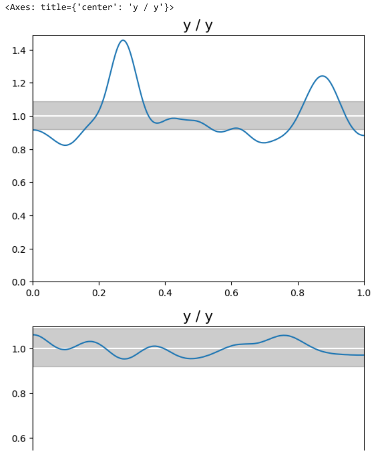
In the hierarchical model, I used varying intercepts and slopes for each district, guided by hyperpriors. This captures both district-specific effects and the overarching trend of age on contraceptive use. The HalfStudentT distributions provide robustness against outliers.

(b) For each model evaluate the sampling performance and discuss your findings.

```
#pooled
pooled_pp = pm.sample_posterior_predictive(trace_pooled, model = pooled)
az.plot_bpv(pooled_pp)

#unpooled model
unpooled_pp = pm.sample_posterior_predictive(unpooled_trace, model = unpooled_model)
az.plot_bpv(unpooled_pp)
```

```
#heirarchical
unpooled_pp = pm.sample_posterior_predictive(hierarchical_trace, model = hierarchical_model)
az.plot_bpv(unpooled_pp)
```

```
100.00% [4000/4000 00:01<00:00]
100.00% [2000/2000 00:00<00:00]
100.00% [2000/2000 00:01<00:00]
<Axes: title={'center': 'y / y'}>
```



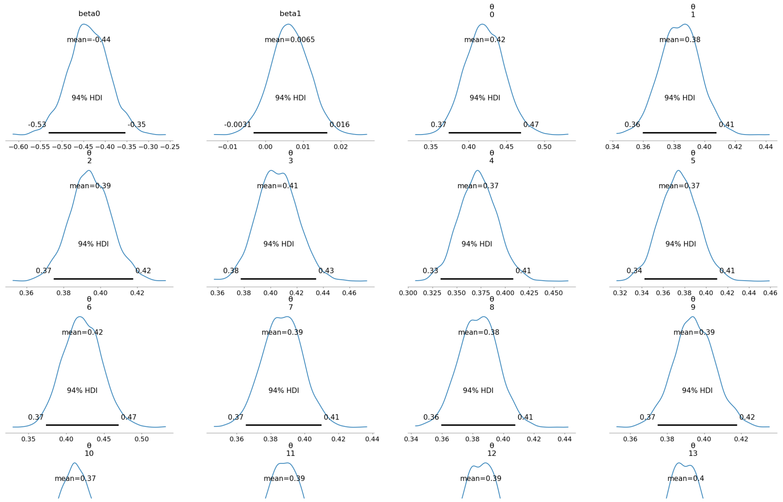
As expected, without the use of district as a predictroy variable, the unpooled model is not nearly as accurate as the others. We see the best performance within the pooled model with p-values that lie entriely in an acceptable range. Additionally, the heirarchical model seems to have a degree of accuracy that sits between the two. While the heirarchical may hypothetically have the best performance in some circumstances, the accuracy of this model might suggest suboptimal choice of priors or tuning for that model.

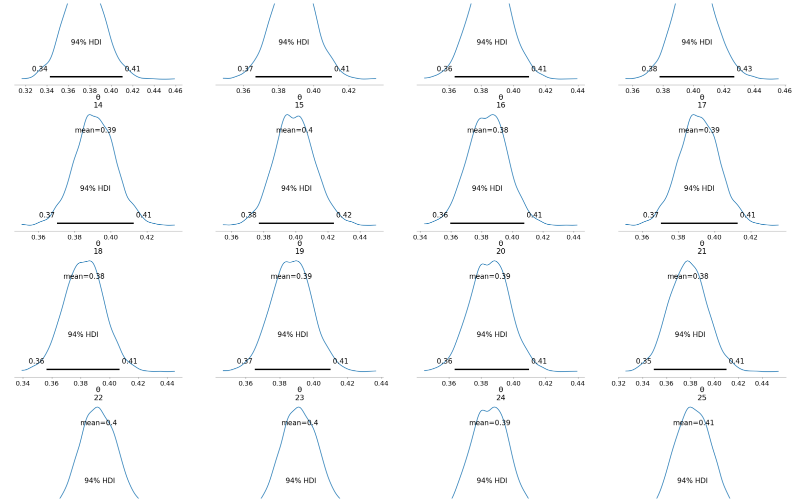
(c) Plot the posterior distributions for the parameters for each model and discuss what they show regarding the question posed by WHO.

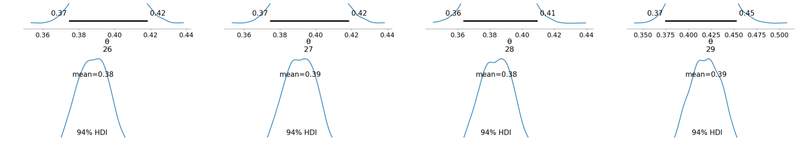
0.0

```
az.plot_posterior(trace_pooled)
```

```
/usr/local/lib/python3.10/dist-packages/arviz/plots/plot_utils.py:271: UserWarning: rcParams['plot.max_subplots'] (40) is smaller than the number of vari
  warnings.warn(
array([[<Axes: title={'center': 'beta0'}>,
        <Axes: title={'center': 'beta1'}>,
        <Axes: title={'center': 'θ\n0'}>,
        <Axes: title={'center': 'θ\n1'}>],
       [<Axes: title={'center': 'θ\n2'}>,
        <Axes: title={'center': 'θ\n3'}>,
        <Axes: title={'center': 'θ\n4'}>,
        <Axes: title={'center': 'θ\n5'}>],
       [<Axes: title={'center': 'θ\n6'}>,
        <Axes: title={'center': 'θ\n7'}>,
        <Axes: title={'center': 'θ\n8'}>,
        <Axes: title={'center': 'θ\n9'}>],
       [<Axes: title={'center': 'θ\n10'}>,
        <Axes: title={'center': 'θ\n11'}>,
        <Axes: title={'center': 'θ\n12'}>,
        <Axes: title={'center': 'θ\n13'}>],
       [<Axes: title={'center': 'θ\n14'}>,
        <Axes: title={'center': 'θ\n15'}>,
        <Axes: title={'center': 'θ\n16'}>,
        <Axes: title={'center': 'θ\n17'}>],
       [<Axes: title={'center': 'θ\n18'}>,
        <Axes: title={'center': 'θ\n19'}>,
        <Axes: title={'center': 'θ\n20'}>,
        <Axes: title={'center': 'θ\n21'}>],
       [<Axes: title={'center': 'θ\n22'}>,
        <Axes: title={'center': 'θ\n23'}>,
        <Axes: title={'center': 'θ\n24'}>,
        <Axes: title={'center': 'θ\n25'}>],
       [<Axes: title={'center': 'θ\n26'}>,
        <Axes: title={'center': 'θ\n27'}>,
        <Axes: title={'center': 'θ\n28'}>,
        <Axes: title={'center': 'θ\n29'}>],
       [<Axes: title={'center': 'θ\n30'}>,
        <Axes: title={'center': 'θ\n31'}>,
        <Axes: title={'center': 'θ\n32'}>,
        <Axes: title={'center': 'θ\n33'}>],
       [<Axes: title={'center': 'θ\n34'}>,
        <Axes: title={'center': 'θ\n35'}>,
        <Axes: title={'center': 'θ\n36'}>,
        <Axes: title={'center': 'θ\n37'}>]], dtype=object)
```

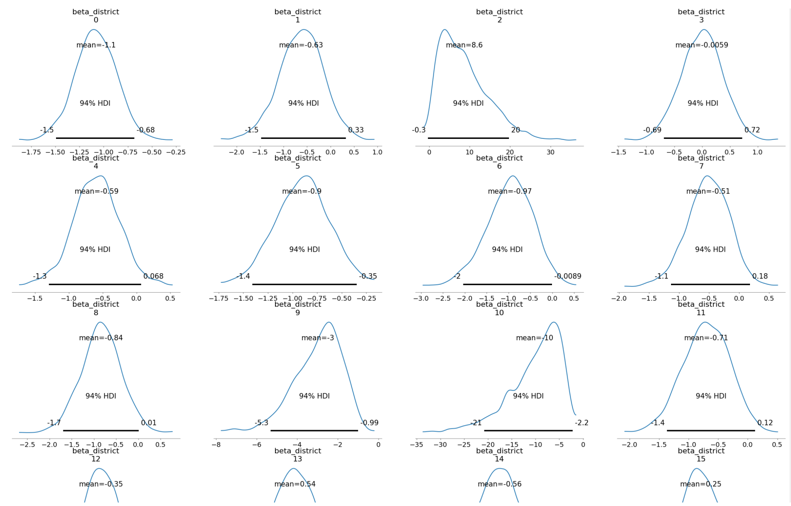


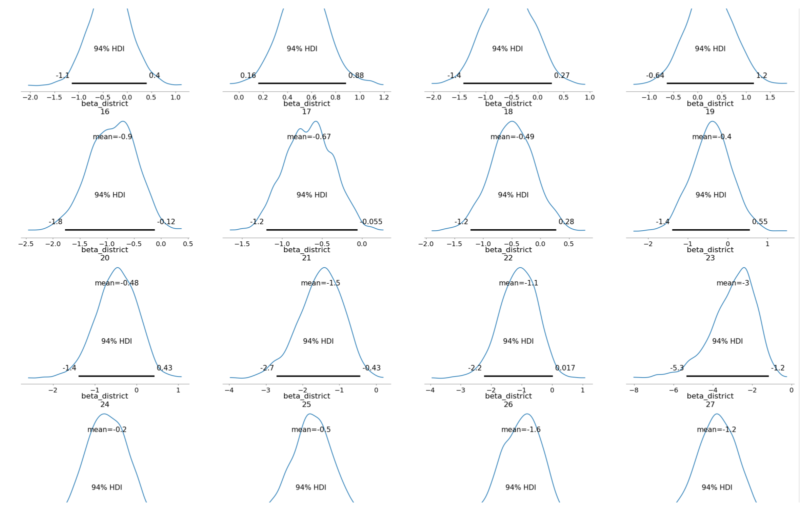


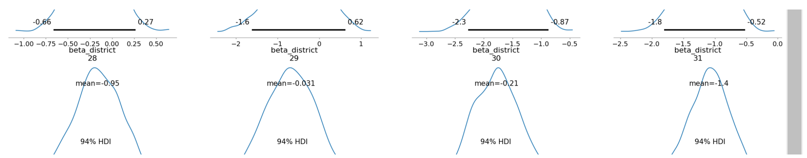


```
az.plot_posterior(unpooled_trace)
```

```
/usr/local/lib/python3.10/dist-packages/arviz/plots/plot_utils.py:271: UserWarning: rcParams['plot.max_subplots'] (40) is smaller than the number of vari
  warnings.warn(
array([[<Axes: title={'center': 'beta_district\n0'}>,
        <Axes: title={'center': 'beta_district\n1'}>,
        <Axes: title={'center': 'beta_district\n2'}>,
        <Axes: title={'center': 'beta_district\n3'}>],
       [<Axes: title={'center': 'beta_district\n4'}>,
        <Axes: title={'center': 'beta_district\n5'}>,
        <Axes: title={'center': 'beta_district\n6'}>,
        <Axes: title={'center': 'beta_district\n7'}>],
       [<Axes: title={'center': 'beta_district\n8'}>,
        <Axes: title={'center': 'beta_district\n9'}>,
        <Axes: title={'center': 'beta_district\n10'}>,
        <Axes: title={'center': 'beta_district\n11'}>],
       [<Axes: title={'center': 'beta_district\n12'}>,
        <Axes: title={'center': 'beta_district\n13'}>,
        <Axes: title={'center': 'beta_district\n14'}>,
        <Axes: title={'center': 'beta_district\n15'}>],
       [<Axes: title={'center': 'beta_district\n16'}>,
        <Axes: title={'center': 'beta_district\n17'}>,
        <Axes: title={'center': 'beta_district\n18'}>,
        <Axes: title={'center': 'beta_district\n19'}>],
       [<Axes: title={'center': 'beta_district\n20'}>,
        <Axes: title={'center': 'beta_district\n21'}>,
        <Axes: title={'center': 'beta_district\n22'}>,
        <Axes: title={'center': 'beta_district\n23'}>],
       [<Axes: title={'center': 'beta_district\n24'}>,
        <Axes: title={'center': 'beta_district\n25'}>,
        <Axes: title={'center': 'beta_district\n26'}>,
        <Axes: title={'center': 'beta_district\n27'}>],
       [<Axes: title={'center': 'beta_district\n28'}>,
        <Axes: title={'center': 'beta_district\n29'}>,
        <Axes: title={'center': 'beta_district\n30'}>,
        <Axes: title={'center': 'beta_district\n31'}>],
       [<Axes: title={'center': 'beta_district\n32'}>,
        <Axes: title={'center': 'beta_district\n33'}>,
        <Axes: title={'center': 'beta_district\n34'}>,
        <Axes: title={'center': 'beta_district\n35'}>],
       [<Axes: title={'center': 'beta_district\n36'}>,
        <Axes: title={'center': 'beta_district\n37'}>,
        <Axes: title={'center': 'beta_district\n38'}>,
        <Axes: title={'center': 'beta_district\n39'}>]], dtype=object)
```







```
az.plot_posterior(hierarchical_trace)
```

These plots display the distribution for each of the betas. They demonstrate that the distributions chosen are good fits to measure the contraception use of women in Bangladesh.

(d) Plot each of the predictions with age.centered on the x-axis and the expected proportion of women using contraception on the y-Axis with overlaid plots for each of the districts, as appropriate. Briefly explain these results as you will report them to the government of Bangladesh.

```
#  parameters from the model trace

beta0_pooled_samples = trace_pooled.posterior["beta0"].values
beta1_pooled_samples = trace_pooled.posterior["beta1"].values

beta_district_samples = unpooled_trace.posterior["beta_district"].values
beta_age_samples = unpooled_trace.posterior["beta_age"].values

a_district_samples = hierarchical_trace.posterior["a_district"].values
b_district_samples = hierarchical_trace.posterior["b_district"].values
```

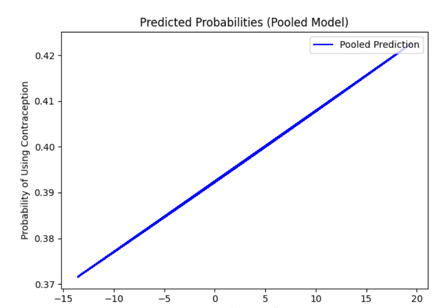
```
beta0_pooled_mean = np.mean(beta0_pooled_samples)
beta1_pooled_mean = np.mean(beta1_pooled_samples)
```

```
# making preds for pooled
prediction_pooled = beta0_pooled_mean + beta1_pooled_mean * age_centered

#Transforming the prediction to probability using logistic function
probability_pooled = 1 / (1 + np.exp(-prediction_pooled))
```

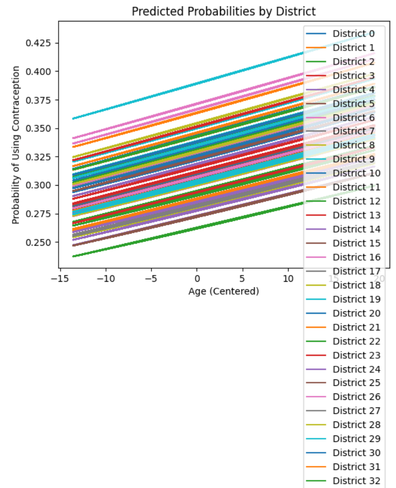
```
predictions = []

# Compute Predictions for Each District
for d in np.unique(districts):
    district_mean_beta = np.mean(beta_district_samples[:, d])
    age_mean_beta = np.mean(beta_age_samples)

    prediction = district_mean_beta + age_mean_beta * age_centered

    # Transform prediction to probability using logistic function
    probability = 1 / (1 + np.exp(-prediction))

    predictions.append(probability)
```

```
hierarchical_predictions = []

for d in np.unique(districts):
    a_district_mean = np.mean(a_district_samples[:, d])
    b_district_mean = np.mean(b_district_samples[:, d])

    prediction_hierarchical = a_district_mean + b_district_mean * age_centered

    probability_hierarchical = 1 / (1 + np.exp(-prediction_hierarchical))

    hierarchical_predictions.append(probability_hierarchical)
```

```
# plotting preds for pooled
plt.plot(age_centered, probability_pooled, label='Pooled Prediction', color='blue')

plt.xlabel('Age (Centered)')
plt.ylabel('Probability of Using Contraception')
plt.title('Predicted Probabilities (Pooled Model)')
plt.legend(loc='upper right')
plt.tight_layout()
plt.show()
```

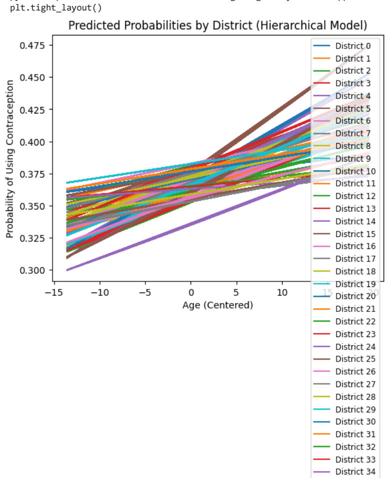


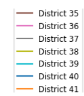
```python
# Un-pooled: predictions for conctreception based on age.centred
for idx, district_probs in enumerate(predictions):
    plt.plot(age_centered, district_probs, label=f"District {idx}")

plt.xlabel('Age (Centered)')
plt.ylabel('Probability of Using Contraception')
plt.title('Predicted Probabilities by District')
plt.legend()
plt.show()
```




```python
# Plot the hierarchical predictions
for idx, district_probs in enumerate(hierarchical_predictions):
    plt.plot(age_centered, district_probs, label=f"District {idx}")

plt.xlabel('Age (Centered)')
plt.ylabel('Probability of Using Contraception')
plt.title('Predicted Probabilities by District (Hierarchical Model)')
plt.legend(loc='upper right', fontsize='small')
plt.tight_layout()
plt.show()
```

```
<ipython-input-21-5b58207d13f4>:9: UserWarning: Tight layout not applied. The bottom and top margins cannot be made large enough to accommodate all axes
  plt.tight_layout()
```





Based on each of these above graphs, we can see how each model predicts the probability of conctraception use in Bangaledeshi women. In the unpooled model, we can see that the model, since it's pooled, doesn't differentiate between districts in predicting the odds of using contraception. On the other hand, the heirarchial and pooled models show the difference between districts when predicting the probability of using contraception. The heirarchical model shows different interecpts and slopes between dirsticts, while the pooled model has different interecpts but the same slope for its logistic regression.