```
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



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

Choose Files CHDdata.csv

- **CHDdata.csv**(text/csv) - 21018 bytes, last modified: 10/24/2023 - 100% done

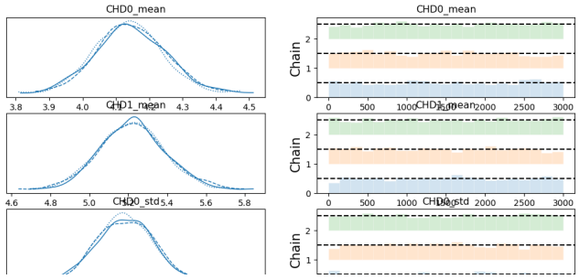
Saving CHDdata.csv to CHDdata.csv

```
chd = pd.read_csv('CHDdata.csv', header=0)
```

1. (10) Based on the analysis we did in class comparing systolic blood pres sure among men in the South African study [1], you are now asked to extend this analysis by comparing low density lipoprotein (ldl) levels be tween the same two groups in the study: those with and without coronary heart disease. Perform this analysis and comment on whether you nd a statistically signi cant di erence between the groups for this variable. To do this analysis you will use the CHD data set (CHDdata.csv).

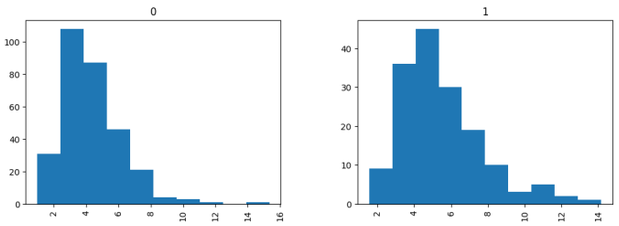
```
summary_stats = (chd.loc[:, ["chd", "ldl"]]
                    .groupby("chd")
                    .agg(["mean", "std", "count"]))
summary_stats

ldl_df = chd[["ldl", "chd"]]
ldl_df.hist("ldl", by="chd", figsize=(12, 4))
```

	mean	sd	hdi_3%	hdi_97%	mcse_mean	mcse_sd	ess_bulk	ess_tail	r_hat
CHD0_mean	4.143	0.103	3.956	4.341	0.002	0.002	2362.0	2091.0	1.0
CHD1_mean	5.220	0.166	4.916	5.542	0.003	0.002	2701.0	2168.0	1.0
CHD0_std	1.492	0.092	1.328	1.675	0.002	0.001	2345.0	1868.0	1.0
CHD1_std	1.806	0.144	1.543	2.081	0.003	0.002	2408.0	2179.0	1.0
v	6.034	1.720	3.529	9.056	0.041	0.032	2060.0	2034.0	1.0
difference of means	-1.077	0.189	-1.443	-0.726	0.004	0.003	2846.0	2413.0	1.0
difference of stds	-0.314	0.146	-0.584	-0.043	0.002	0.002	3766.0	2542.0	1.0
effect size	-0.651	0.119	-0.874	-0.427	0.002	0.002	2736.0	2171.0	1.0




```
az.plot_forest(chd_trace, var_names=["difference of means", "difference of stds", "effect size"])
```

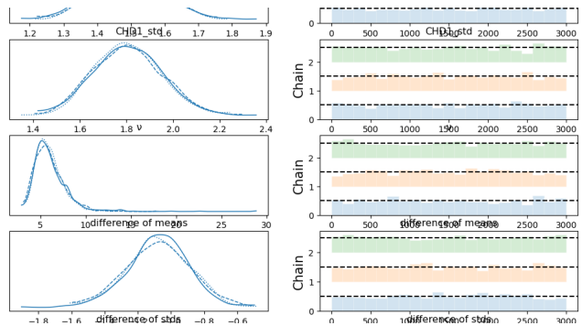
```
y0 = ldl_df.loc[ldl_df["chd"]==0]['ldl']
y1 = ldl_df.loc[ldl_df["chd"]==1]["ldl"]
```



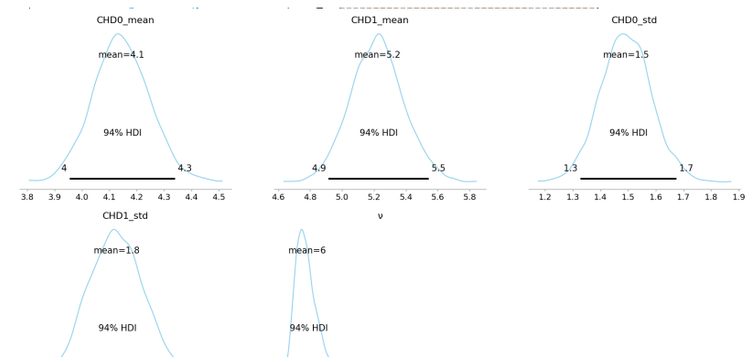

```
random_seed = 100
cores = 3

μ_prior = chd['ldl'].mean()
σ_prior = chd['ldl'].std() * 2

# prior for Std
σ_low = 1
σ_high = 100

with pm.Model() as model:
    CHD0_mean = pm.Normal("CHD0_mean", mu=μ_prior, sigma=σ_prior)
    CHD1_mean = pm.Normal("CHD1_mean", mu=μ_prior, sigma=σ_prior)
    CHD0_std = pm.Uniform("CHD0_std", lower=σ_low, upper=σ_high)
    CHD1_std = pm.Uniform("CHD1_std", lower=σ_low, upper=σ_high)
    v = pm.Exponential("v", 1/29)
    CHD0 = pm.StudentT("No_CHD", nu=v, mu=CHD0_mean, sigma=CHD0_std, observed=y0)
    CHD1 = pm.StudentT("CHD", nu=v, mu=CHD1_mean, sigma=CHD1_std, observed=y1)
    diff_of_means = pm.Deterministic("difference of means", CHD0_mean - CHD1_mean)
    diff_of_stds = pm.Deterministic("difference of stds", CHD0_std - CHD1_std)
    effect_size = pm.Deterministic(
        "effect size", diff_of_means / np.sqrt((CHD0_std ** 2 + CHD1_std ** 2) / 2)
    )

    chd_trace=pm.sample(random_seed = random_seed, cores = cores)
```

100.00% [6000/6000 00:12<00:00 Sampling 3 chains, 0 divergences]

```
az.plot_trace(chd_trace, kind = "rank_bars")
az.summary(chd_trace)
```

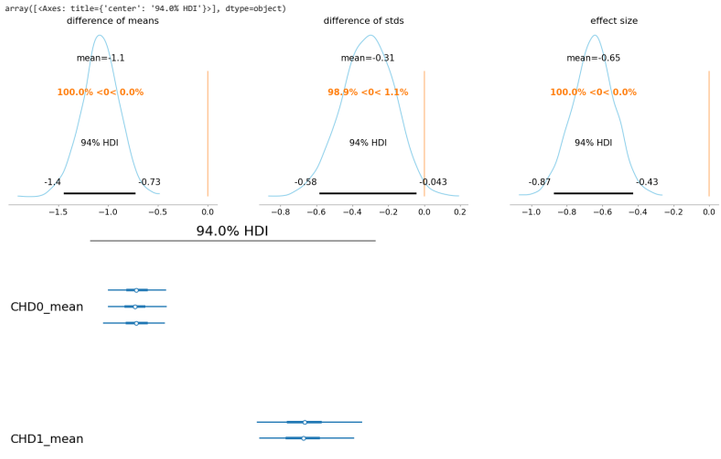



```
az.plot_posterior(
    chd_trace,
    var_names=["CHD0_mean", "CHD1_mean", "CHD0_std", "CHD1_std", "v"],
    color="#87ceeb",
);
```




```
az.plot_posterior(
    chd_trace,
    var_names=["difference of means", "difference of stds", "effect size"],
    ref_val=0,
    color="#87ceeb",
);
az.plot_forest(chd_trace, var_names = ['CHD0_mean','CHD1_mean'])
```

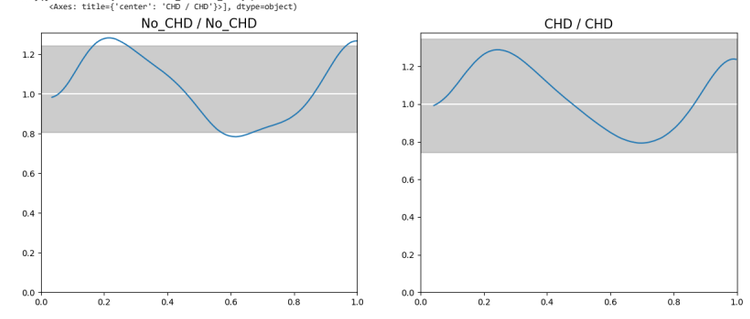
```
array([<Axes: title={'center': '94.0% HDI'}>], dtype=object)
```




```
#chd_trace
chd_pp = pm.sample_posterior_predictive(chd_trace, model = model)
az.plot_bpv(chd_pp)

# !!!!!! Note: I ran very similar code as the prof and these plots were generated once, but it seems
# that pymc was updated or something and now I can't create the plots when I reran this and I havent beeen able to resolve the issue
```

100.00% [3000/3000 00:01<00:00]

```
array([<Axes: title={'center': 'No_CHD / No_CHD'}>,
       <Axes: title={'center': 'CHD / CHD'}>], dtype=object)
```

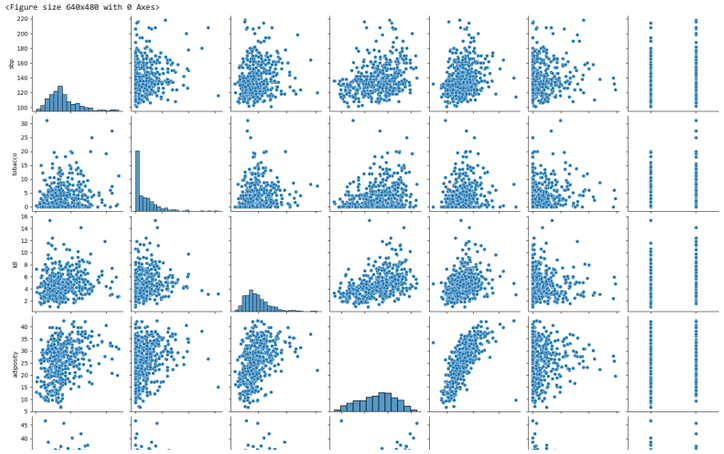


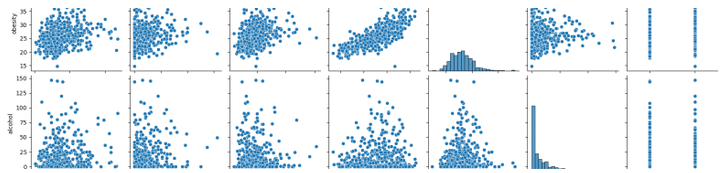
Based on the various plotes above, there is a clear difference in likelyhood to develop CHD between groups with lower and higher LDL contnet. In particular, the 94% HDI of the posterior difference of means shows that group with higher mean LDL content is more likely to have CHD. The same can be seen with the secondary 94%HDI set of graphs, and the subsequenmt difference of means graph. The effect size, in addition to the difference of means, seems to show the same general trends as well.

3. (30) The South African Health Ministry is now asking you to to extend your work from problem 1 and perform a full analysis of the data in the coronary heart disease study [1]. Their goal is to understand factors that may be associate with coronary heart disease to inform further more focused research on those factors. For this work you will again use the CHD data set (CHDdata.csv). Perform the following steps and answer the associated questions.

(a) Perform a simple exploratory analysis of the data by at least viewing summary statistics. Comment on what the EDA implies for your analysis.

```
#EDA: checking how each continious var is compared against each other continiuous var and also the response variable CHD
# even though chd is discrete I think its a good idea to see how each variable affects it
sns.pairplot(chd,vars= ['sbp' ,'tobacco' ,'ldl' , 'adiposity' ,'obesity' ,'alcohol' ,'chd'])
#Performing analysis of categorical variables againt the response
sns.boxplot(data=chd, x="famhist", y="chd")
plt.figure()
```





Based on the figures above, its clear that theres some correlations between high sbp levels, tobacco use, ldl, obesity and alcohol use. But at the same time there doesnt appear to be any colinearity so high that any predictor needs to be dropped.

(b) Preprocess the data by scaling the predictor variables using a z-transformation (subtract the mean divide by the standard deviation). Explain why this is appropriate. Code the categorical variables and explain your coding

```
preds_num = ['sbp', 'tobacco', 'ldl', 'adiposity', 'typea', 'obesity', 'alcohol', 'age']
for p in preds_num:
  chd[p] -= np.mean(chd[p])
  chd[p] /= np.std(chd[p])
#chd = pd.get_dummies(chd,columns=['famhist'])
fam_vals = {'Absent': 0, 'Present': 1}
chd['famhist'] = chd['famhist'].replace(fam_vals)


chd
```

	sbp	tobacco	ldl	adiposity	famhist	typea	obesity	alcohol	age	chd
0	1.058564	1.823073	0.478412	-0.295503	1	-0.418470	-0.176786	3.277738	0.629336	1
1	0.277089	-0.790237	-0.159680	0.412140	0	0.193344	0.671373	-0.612745	1.383115	1
2	-0.992806	-0.774980	-0.609245	0.884332	1	-0.112563	0.735519	-0.541183	0.218184	0
3	1.546985	0.842264	0.807126	1.624141	1	-0.214532	1.412621	0.295062	1.040488	1
4	-0.211332	2.171805	-0.599577	0.305351	1	0.703189	-0.012856	1.647775	0.423760	1
...	...	...	...	...	...	...	...	...	...	...
457	3.696039	-0.705234	0.599263	0.812281	0	1.111065	0.571590	-0.696983	1.040488	0
458	2.133091	0.123004	-0.159680	0.861173	0	-0.112563	0.609602	0.068519	0.629336	1
459	-1.481228	-0.138545	-1.522877	-1.309364	0	-1.336191	-1.414575	0.392385	0.834912	0

Here, scaling the data makes the distribution appear normal and thereby we can avoid the need to use the individual means for each column of the data. Encoding the categorical variables allows for numerical analysis to be run on those variables.

462 rows × 10 columns

(c) Develop and use Bayesian GLM model with all the predictor variables. Explain your choice of priors and their parameters. Show the graphical representation of your model.

```
with pm.Model() as chd_lin_model:
  beta0 = pm.Normal('beta0', mu=0, sigma=10)
  beta1 = pm.Normal('beta1', mu=0, sigma=10)
  beta2 = pm.Normal('beta2', mu=0, sigma=10)
  beta3 = pm.Normal('beta3', mu=0, sigma=10)
  beta4 = pm.Normal('beta4', mu=0, sigma=10)
  beta5 = pm.Normal('beta5', mu=0, sigma=10)
  beta6 = pm.Normal('beta6', mu=0, sigma=10)
  beta7 = pm.Normal('beta7', mu=0, sigma=10)
  beta8 = pm.Normal('beta8', mu=0, sigma=10)
  beta9 = pm.Normal('beta9', mu=0, sigma=10)
```

```
μ = pm.Deterministic("μ", beta0 + beta1*chd.iloc[:, 0] + beta2*chd.iloc[:, 1] + beta3*chd.iloc[:, 2] + beta4*chd.iloc[:, 3]
                     + beta5*chd.iloc[:, 4] + beta6*chd.iloc[:, 5] + beta7*chd.iloc[:, 6] + beta8*chd.iloc[:, 7] + beta9*chd.iloc[:, 8])

  chd_data = chd.iloc[:, 9]

  chd_pre = pm.Bernoulli('chd_pre', p = pm.math.sigmoid(μ))

  chd_post = pm.Bernoulli('chd_post', p = pm.math.sigmoid(μ), observed=chd_data)

  trace_glm = pm.sample(1000, cores=8, random_seed=1234, return_inferencedata=False)
  glm_trace = pm.to_inference_data(trace=trace_glm, log_likelihood=True)
  glm_pp = pm.sample_posterior_predictive(glm_trace,
                                          var_names=["chd_pre","chd_post"],
                                          random_seed = 1234)
g3=pm.model_to_graphviz(chd_lin_model)
graphics_path = '/content/drive/MyDrive/'
g3.render(graphics_path+"chd_lin_model", format='png', cleanup=True)
g3
```

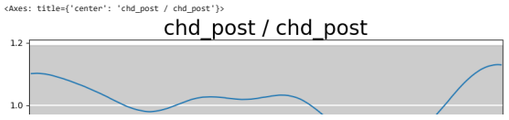
100.00% [16000/16000 18:17<00:00 Sampling 8 chains, 0 divergences]
100.00% [8000/8000 00:03<00:00]

In this linear model, I model the correlation of 9 variables with the outcome chd_data. Each variable is associated with a coefficient (beta) that describes its relationship with the outcome. A deterministic function computes the expected value (μ) based on a linear combination of these variables and their respective coefficients. The outcome is binary, and hence a Bernoulli distribution is used for both the prior prediction (chd_pre) and the posterior likelihood (chd_post), with a logistic sigmoid function linking the linear combination to a probability. Sampling provides posterior estimates for all model parameters.

(d) Evaluate both the sampling used by your model, as well as, the prior and posterior predictive results.

```
with chd_lin_model:
  chd_prior = pm.sample_prior_predictive()
az.plot_dist(chd_prior.prior['chd_pre'])
```

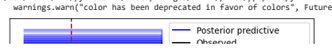

```
az.plot_bpv(glm_pp, figsize = (10,8))
```




```
with chd_lin_model:
  pm.sample_posterior_predictive(glm_trace, extend_inferencedata=True, random_seed=1234)
fig, ax = plt.subplots()
az.plot_ppc(glm_pp, color = 'b', observed_rug=True, ax=ax)
ax.axvline(chd['chd'].mean(), ls="--", color="r", label="Observed mean")
ax.legend(fontsize=10);
```

100.00% [8000/8000 00:02<00:00]

```
/usr/local/lib/python3.10/dist-packages/arviz/plots/ppcplot.py:241: FutureWarning: color has been deprecated in favor of colors
  warnings.warn("color has been deprecated in favor of colors", FutureWarning)
```

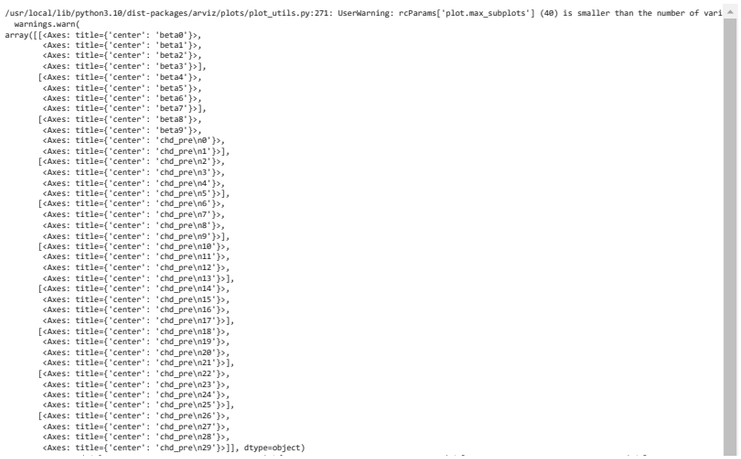


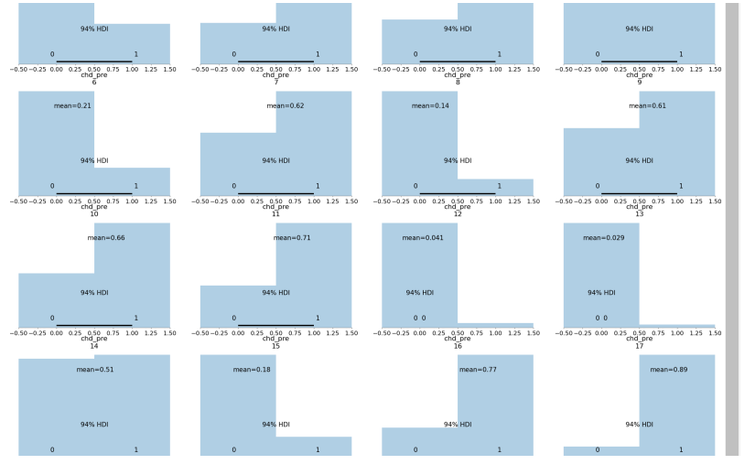
Basedon the above plots, we can see that the model has p-values that lie in the accepted range, demonstrating that the model is satisfactory. The posterior predictive sampling, with similaries between the observed and predicted parameters, supports the validity of the model.

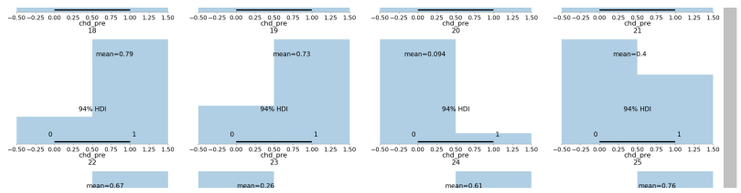
(e) Provide plots of the posterior distributions for the parameters and discuss what these results imply for the relevant predictor variables and the overall model. Show posterior analyses to include relevant odds ratio(s) and describe what this analysis means for the goal of your study.

```
az.plot_posterior(glm_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': 'beta0'}>,
        <Axes: title={'center': 'beta1'}>,
        <Axes: title={'center': 'beta2'}>,
        <Axes: title={'center': 'beta3'}>],
       [<Axes: title={'center': 'beta4'}>,
        <Axes: title={'center': 'beta5'}>,
        <Axes: title={'center': 'beta6'}>,
        <Axes: title={'center': 'beta7'}>],
       [<Axes: title={'center': 'beta8'}>,
        <Axes: title={'center': 'beta9'}>,
        <Axes: title={'center': 'chd_pre\n0'}>,
        <Axes: title={'center': 'chd_pre\n1'}>],
       [<Axes: title={'center': 'chd_pre\n2'}>,
        <Axes: title={'center': 'chd_pre\n3'}>,
        <Axes: title={'center': 'chd_pre\n4'}>,
        <Axes: title={'center': 'chd_pre\n5'}>],
       [<Axes: title={'center': 'chd_pre\n6'}>,
        <Axes: title={'center': 'chd_pre\n7'}>,
        <Axes: title={'center': 'chd_pre\n8'}>,
        <Axes: title={'center': 'chd_pre\n9'}>],
       [<Axes: title={'center': 'chd_pre\n10'}>,
        <Axes: title={'center': 'chd_pre\n11'}>,
        <Axes: title={'center': 'chd_pre\n12'}>,
        <Axes: title={'center': 'chd_pre\n13'}>],
       [<Axes: title={'center': 'chd_pre\n14'}>,
        <Axes: title={'center': 'chd_pre\n15'}>,
        <Axes: title={'center': 'chd_pre\n16'}>,
        <Axes: title={'center': 'chd_pre\n17'}>],
       [<Axes: title={'center': 'chd_pre\n18'}>,
        <Axes: title={'center': 'chd_pre\n19'}>,
        <Axes: title={'center': 'chd_pre\n20'}>,
        <Axes: title={'center': 'chd_pre\n21'}>],
       [<Axes: title={'center': 'chd_pre\n22'}>,
        <Axes: title={'center': 'chd_pre\n23'}>,
        <Axes: title={'center': 'chd_pre\n24'}>,
        <Axes: title={'center': 'chd_pre\n25'}>],
       [<Axes: title={'center': 'chd_pre\n26'}>,
        <Axes: title={'center': 'chd_pre\n27'}>,
        <Axes: title={'center': 'chd_pre\n28'}>,
        <Axes: title={'center': 'chd_pre\n29'}>]], dtype=object)
```





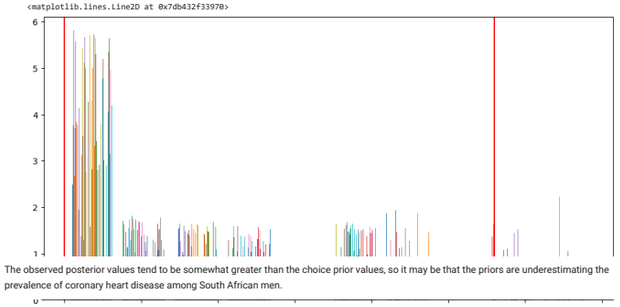


Based on the above plots, its clear that the explanatroy variables show a normal distribtution, so Bernouli distribution for the prior was a good choice. The HDI charts produced seem to indicate that the odds ratios for the parameters are acceptable. Beta are also normally distributed, showing that there shouldnt have been in issue in that area either.

In terms of the goals of the study, this helps us to see that the model being produced is accurate and should help our stakeholders achieve their goals.

```
_, ax = plt.subplots(figsize=(12, 6))
b = glm_trace.posterior["chd_pre"]
OR = np.mean(b, axis =0)
lb, ub = np.percentile(OR, 2.5), np.percentile(OR, 97.5)
plt.hist(np.exp(OR), bins=10, density=True)
plt.axvline(np.exp(lb), color = 'r')
plt.axvline(np.exp(ub), color = 'r')
```

```
<matplotlib.lines.Line2D at 0x7db432f33970>
```



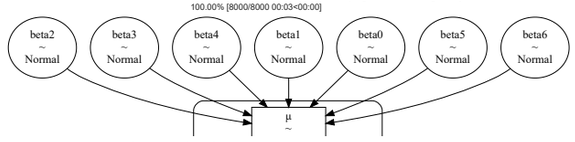
The observed posterior values tend to be somewhat greater than the choice prior values, so it may be that the priors are underestimating the prevalence of coronary heart disease among South African men.

(f) Develop a second model to help in your study of this problem. This new model can use fewer predictor variables but must contain at least one nonlinear component. Explain your choice of priors and their parameters, and show the graphical representation of your new model.

```
chd2 = chd
chd2['age'] = np.square(chd2['age'])

with pm.Model() as glm2:
  beta0 = pm.Normal('beta0', mu=0, sigma=10)
```

```
  beta1 = pm.Normal('beta1', mu=0, sigma=10)
  beta2 = pm.Normal('beta2', mu=0, sigma=10)
  beta3 = pm.Normal('beta3', mu=0, sigma=10)
  beta4 = pm.Normal('beta4', mu=0, sigma=10)
  beta5 = pm.Normal('beta5', mu=0, sigma=10)
  beta6 = pm.Normal('beta6', 0, 10)

  μ = pm.Deterministic("μ", beta0 + beta1*chd2.iloc[:, 0] + beta2*chd2.iloc[:, 1] + beta3*chd2.iloc[:, 2]
                       + beta4*chd2.iloc[:, 5] + beta5*chd2.iloc[:, 6] + beta6*chd2.iloc[:, 8])

  chd2_data = chd2.iloc[:, 9]

  chd2_pre = pm.Bernoulli('chd2_pre', p=pm.math.sigmoid(μ))

  chd2_post = pm.Bernoulli('chd2_post', p = pm.math.sigmoid(μ), observed=chd2_data)

  trace_glm2 = pm.sample(1000, cores=8, random_seed=1234, target_accept=.99, return_inferencedata=False)
  glm_trace2 = pm.to_inference_data(trace=trace_glm2, log_likelihood=True)
  glm_pp2 = pm.sample_posterior_predictive(glm_trace2,
                                           var_names=["chd2_pre","chd2_post"],
                                           random_seed = 1234)

g3f=pm.model_to_graphviz(glm2)
graphics_path = '/content/drive/MyDrive/'
g3f.render(graphics_path+"GLM_Model", format='png', cleanup=True)
g3f
```

```
100.00% [16000/16000 16:08<00:00 Sampling 8 chains, 0 divergences]
100.00% [8000/8000 00:03<00:00]
```

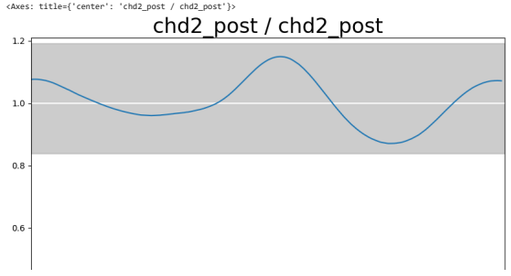


I kept the betas as normal, except for the use of a half normal on age, which was squared to become the nonlinear element. All other parameters remained the same for consistency.

(g) For the second model evaluate the sampling and choice of priors using Bayesian p-value and posterior predictive plots. Briefly explain your results.

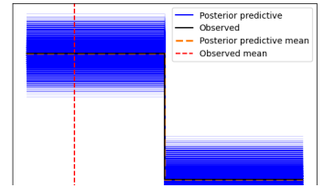
```
az.plot_bpv(glm_pp2, figsize = (10,8))
```

```
<Axes: title={'center': 'chd2_post / chd2_post'}>
```




```
with glm2:
  pm.sample_posterior_predictive(glm_trace2, extend_inferencedata=True, random_seed=1234)
fig, ax = plt.subplots()
az.plot_ppc(glm_pp2, color = 'b', observed_rug=True, ax=ax)
ax.axvline(chd2['chd'].mean(), ls="--", color="r", label="Observed mean")
ax.legend(fontsize=10);
```

```
100.00% [8000/8000 00:02<00:00]
/usr/local/lib/python3.10/dist-packages/arviz/plots/ppcplot.py:241: FutureWarning: color has been deprecated in favor of colors
  warnings.warn("color has been deprecated in favor of colors", FutureWarning)
```

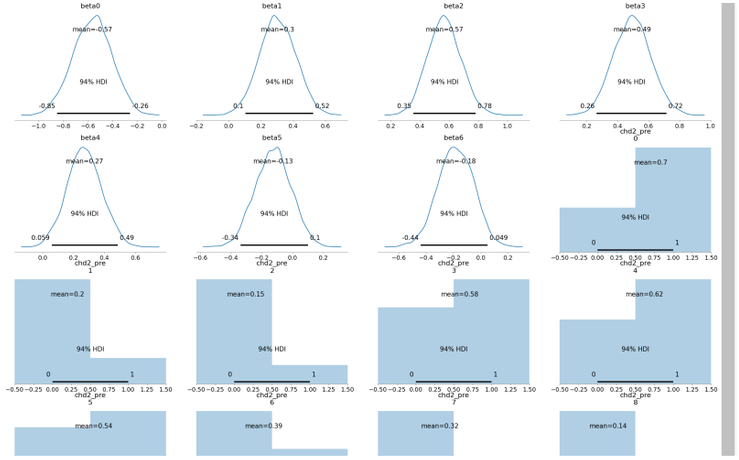


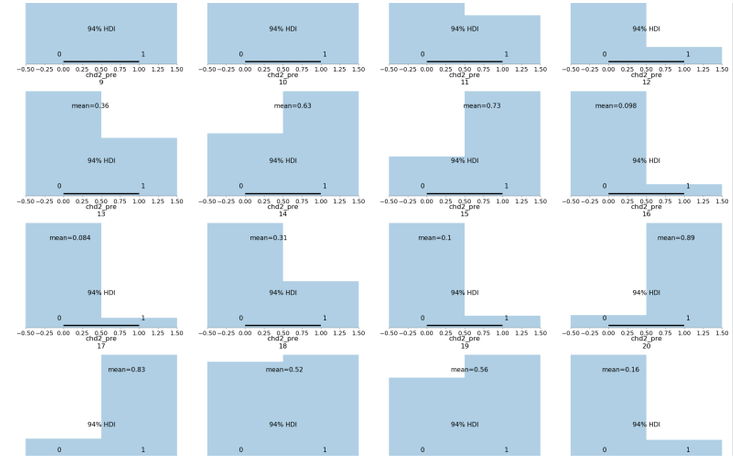
This model appears to be pretty decent. The produced Bayesian p-value is very close to 1, which appears satisfactory. Moreover, there are two clusters present within the posterior predictive plot, so it seems that the priors and parameters were well chosen.

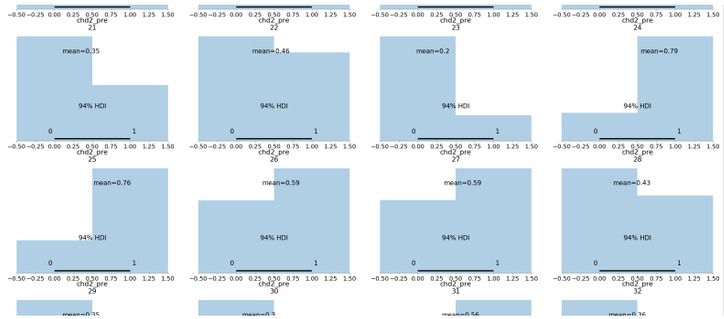
(h) For the second model provide plots of the posterior distributions for the parameters and discuss what these results imply for the relevant predictor variables and the overall model. Show posterior analyses to include relevant odds ratio(s) and describe what this analysis means for the goal of your study.

```
az.plot_posterior(glm_trace2)
```

```
/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': 'beta2'}>,
        <Axes: title={'center': 'beta3'}>],
       [<Axes: title={'center': 'beta4'}>,
        <Axes: title={'center': 'beta5'}>,
        <Axes: title={'center': 'beta6'}>,
        <Axes: title={'center': 'chd2_pre\n0'}>],
       [<Axes: title={'center': 'chd2_pre\n1'}>,
        <Axes: title={'center': 'chd2_pre\n2'}>,
        <Axes: title={'center': 'chd2_pre\n3'}>,
        <Axes: title={'center': 'chd2_pre\n4'}>],
       [<Axes: title={'center': 'chd2_pre\n5'}>,
        <Axes: title={'center': 'chd2_pre\n6'}>,
        <Axes: title={'center': 'chd2_pre\n7'}>,
        <Axes: title={'center': 'chd2_pre\n8'}>],
       [<Axes: title={'center': 'chd2_pre\n9'}>,
        <Axes: title={'center': 'chd2_pre\n10'}>,
        <Axes: title={'center': 'chd2_pre\n11'}>,
        <Axes: title={'center': 'chd2_pre\n12'}>],
       [<Axes: title={'center': 'chd2_pre\n13'}>,
        <Axes: title={'center': 'chd2_pre\n14'}>,
        <Axes: title={'center': 'chd2_pre\n15'}>,
        <Axes: title={'center': 'chd2_pre\n16'}>],
       [<Axes: title={'center': 'chd2_pre\n17'}>,
        <Axes: title={'center': 'chd2_pre\n18'}>,
        <Axes: title={'center': 'chd2_pre\n19'}>,
        <Axes: title={'center': 'chd2_pre\n20'}>],
       [<Axes: title={'center': 'chd2_pre\n21'}>,
        <Axes: title={'center': 'chd2_pre\n22'}>,
        <Axes: title={'center': 'chd2_pre\n23'}>,
        <Axes: title={'center': 'chd2_pre\n24'}>],
       [<Axes: title={'center': 'chd2_pre\n25'}>,
        <Axes: title={'center': 'chd2_pre\n26'}>,
        <Axes: title={'center': 'chd2_pre\n27'}>,
        <Axes: title={'center': 'chd2_pre\n28'}>],
       [<Axes: title={'center': 'chd2_pre\n29'}>,
        <Axes: title={'center': 'chd2_pre\n30'}>,
        <Axes: title={'center': 'chd2_pre\n31'}>,
        <Axes: title={'center': 'chd2_pre\n32'}>]], dtype=object)
```

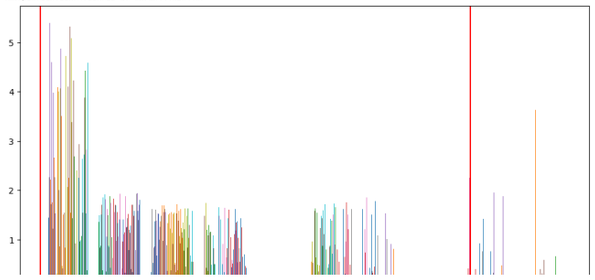







```
_, ax = plt.subplots(figsize=(12, 6))
b = glm_trace2.posterior["chd2_pre"]
OR = np.mean(b, axis =0)
lb, ub = np.percentile(OR, 2.5), np.percentile(OR, 97.5)
plt.hist(np.exp(OR), bins=10, density=True)
plt.axvline(np.exp(lb), color = 'r')
plt.axvline(np.exp(ub), color = 'r')
```

```
<matplotlib.lines.Line2D at 0x7db43482c340>
```



The ratio between the posterior and the prior are again consistently greater than one, demonstrarting that the prescence of CHD may very well be underestimated. In terms of the stakeholders, I would recommend using the other model as it is likely to be more accurate.

(i) Compare the two models using PSIS - leave-one-out cross validation and WAIC.

```
az.compare({'Linear': glm_trace, 'Nonlinear':glm_trace2})
```

	rank	elpd_loo	p_loo	elpd_diff	weight	se	dse	warning	scale

Based on this comparison between the two models, we can see that the linear model appears to havea better overall fit. This is supported by the elpd_diff score, which shows the liinear model to be better.

(j) Briefly summarize your analysis and provide an answer to the South African Health ministry to the question implied by their goal statement.

In conclusion, I have produced a couple models that predict the likelyhood of developing CHD based on a variety of factors. The most prominent risk factors of developing CHD based on these models seem to be: tobacco, LDL, age, family history, and systolic blood pressure. Dealing with these health factros through a public-health based approach may prove to lower the odds of developing chd overall.