

kmmd

April 21, 2021

1 KMMD metrics computation (R kernel)

1.1 Utils

```
[2]: # install.packages("OpenImageR")  
# install.packages("kernlab")  
# install.packages("repr")  
install.packages("BiocManager")  
BiocManager::install("EBImage")
```

Installing package into 'C:/Users/peter/Documents/R/win-library/4.0'
(as 'lib' is unspecified)

package 'BiocManager' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
C:\Users\peter\AppData\Local\Temp\RtmpOcaAQU\downloaded_packages

'getOption("repos")' replaces Bioconductor standard repositories, see
'?repositories' for details

replacement repositories:
CRAN: <https://cran.r-project.org>

Bioconductor version 3.12 (BiocManager 1.30.12), R 4.0.5 (2021-03-31)

Installing package(s) 'BiocVersion', 'EBImage'

also installing the dependencies 'bitops', 'BiocGenerics', 'abind', 'locfit',
'fftwtools', 'RCurl'

package 'bitops' successfully unpacked and MD5 sums checked
package 'BiocGenerics' successfully unpacked and MD5 sums checked
package 'abind' successfully unpacked and MD5 sums checked
package 'locfit' successfully unpacked and MD5 sums checked

```
package 'fftwtools' successfully unpacked and MD5 sums checked
package 'RCurl' successfully unpacked and MD5 sums checked
package 'BiocVersion' successfully unpacked and MD5 sums checked
package 'EBImage' successfully unpacked and MD5 sums checked
```

The downloaded binary packages are in

```
C:\Users\peter\AppData\Local\Temp\RtmpOcaAQU\downloaded_packages
```

Installation paths not writeable, unable to update packages

```
path: C:/Program Files/R/R-4.0.5/library
```

```
packages:
```

```
cluster, mgcv
```

```
Old packages: 'callr', 'DT', 'haven', 'highr', 'knitr', 'parsedate',
'RcppArmadillo', 'tibble', 'tidyverse', 'withr'
```

```
[3]: library(OpenImageR)
library(kernlab)
library(tidyverse)
library(ggplot2)
library(EBImage)
```

Attaching package: 'EBImage'

The following object is masked from 'package:purrr':

```
transpose
```

The following objects are masked from 'package:OpenImageR':

```
readImage, writeImage
```

1.2 Computation of KMMD

```
[4]: compute_kmmd <- function(input_loc, recon_loc) {
  inputs = list.files(input_loc)
  recons = list.files(recon_loc)
  H0.reject = c()
  low_order_mmd_stats = c()
  high_order_mmd_stats = c()
}
```

```

for (i in 1:length(inputs)) {
  input_img = readImage(paste(input_loc, "/", inputs[i], sep=""))
  recon_img = readImage(paste(recon_loc, "/", recons[i], sep=""))

  input_img = rgb_2gray(input_img)
  recon_img = rgb_2gray(recon_img)

  w_input = dim(input_img)[1]
  h_input = dim(input_img)[2]
  w_recon = dim(recon_img)[1]
  h_recon = dim(recon_img)[2]
  if ((w_input != w_recon) || (h_input != h_recon)) {
    recon_img = resize(recon_img, w_input, h_input)
  }

  result = kmmd(input_img, recon_img)
  H0.reject = append(H0.reject, result@H0)
  low_order_mmd_stats = append(low_order_mmd_stats, result@mmdstats[1])
  high_order_mmd_stats = append(high_order_mmd_stats, result@mmdstats[2])
}

result <- list("H0.reject" = H0.reject, "low_order_mmd_stats" =
→low_order_mmd_stats, "high_order_mmd_stats" = high_order_mmd_stats)
return(result)
}

```

1.3 Result of Flowers dataset for BigbiGAN

```

[56]: result = compute_kmmd("./bigbigan_flower_results/inputs", "./
→bigbigan_flower_results/recons")

```

```

Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel

```

```

Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel

```

```

[57]: H0.reject = result$H0.reject
df_H0.reject = as.data.frame(H0.reject) %>% group_by(H0.reject) %>% count() %>%
  ↪rename("inference_count"=n)
low_order_mmd_stats = result$low_order_mmd_stats
df_low_order_mmd_stats = as.data.frame(low_order_mmd_stats)
high_order_mmd_stats = result$high_order_mmd_stats
df_high_order_mmd_stats = as.data.frame(high_order_mmd_stats)

```

1.3.1 Hypothesis test result

H_0 : two images have the same distribution.

H_a : two images do not have the same distribution.

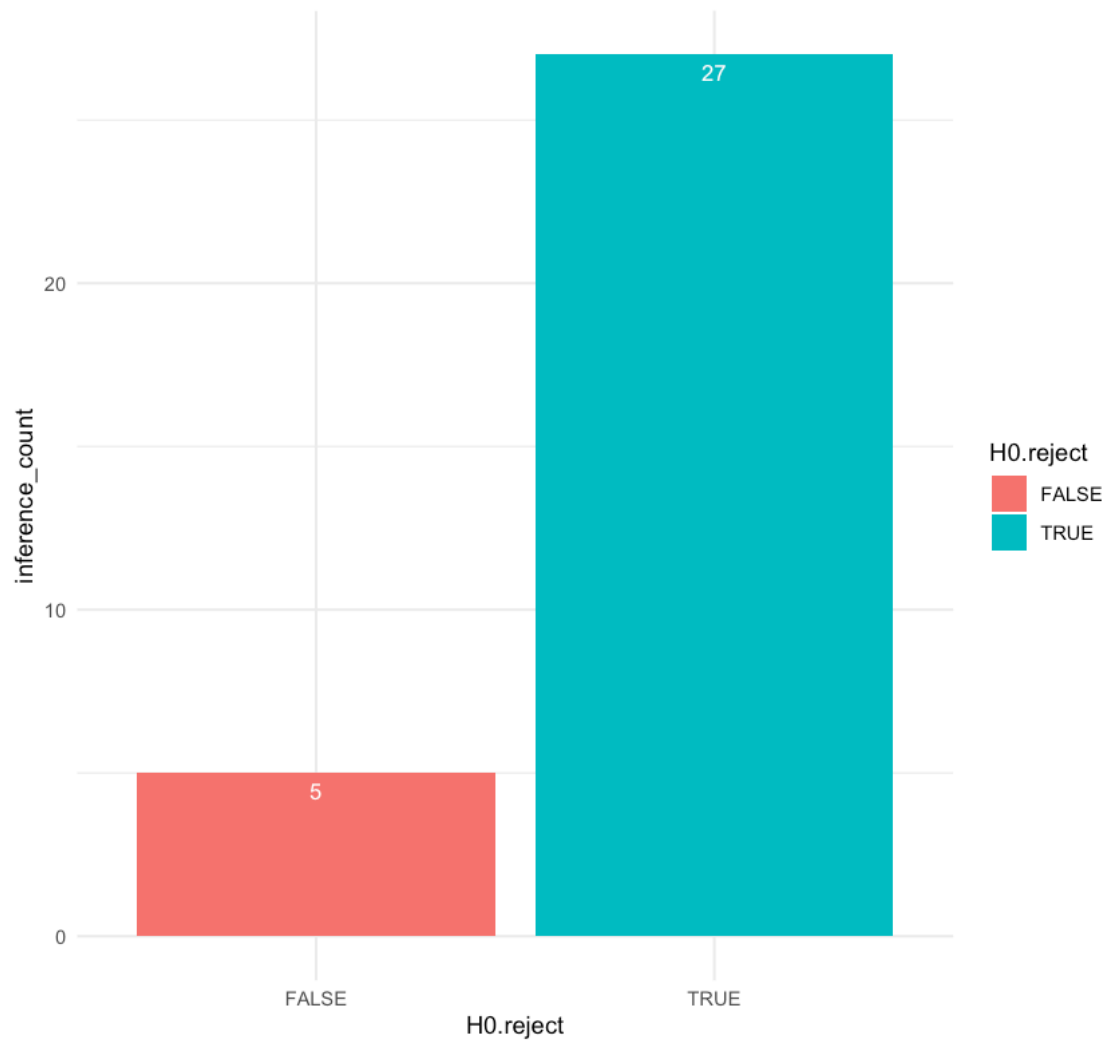
Reference: <http://www.gatsby.ucl.ac.uk/~gretton/mmd/mmd.htm>

```

[58]: ggplot(data=df_H0.reject, aes(x=H0.reject, y=inference_count, fill=H0.reject)) +
  labs(title="H0 hypothesis: two images are from the same distribution for_
  ↪BigbiGAN\n (alpha=0.05)") +
  geom_bar(stat="identity")+
  geom_text(aes(label=inference_count), vjust=1.6, color="white", size=3.5)+
  theme_minimal()

```

H0 hypothesis: two images are from the same distribution for BigbiGAN
(alpha=0.05)



1.3.2 MMD statistics

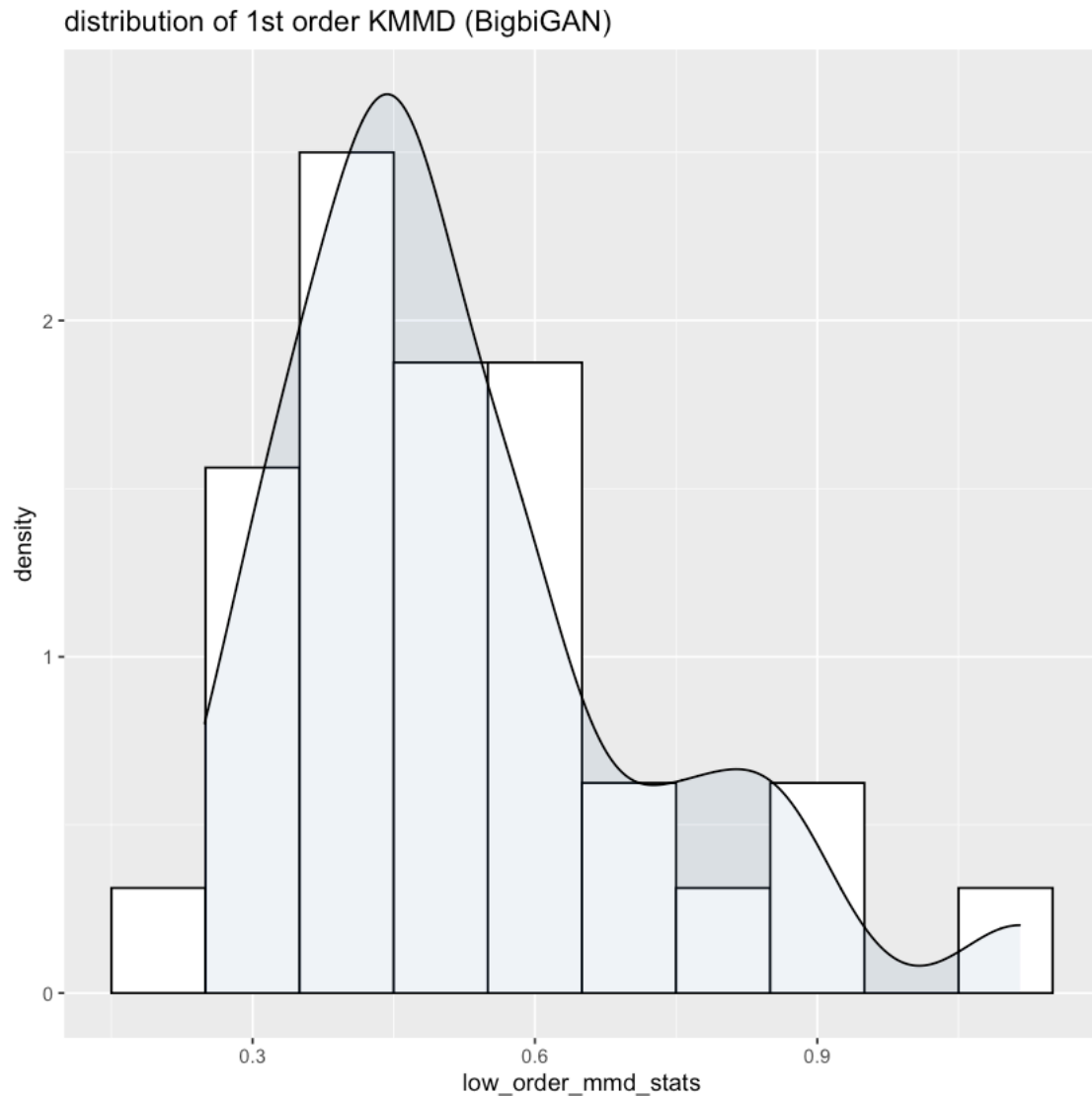
1. Density distribution for 1st order MMD statistics
2. Density distribution for 3rd order MMD statistics
3. Mean and other statistics for 1st order MMD statistics
4. Mean and other statistics for 3rd order MMD statistics

```
[59]: ggplot(df_low_order_mmd_stats, aes(x=low_order_mmd_stats)) +
  labs(title="distribution of 1st order KMMD (BigbiGAN)") +
  geom_histogram(aes(y=..density..),      # Histogram with density instead of
  ↪ count on y-axis
  binwidth=.1,
```

```

    colour="black", fill="white") +
    geom_density(alpha=.1, fill="steelblue") # Overlay with transparent
    ↪ density plot

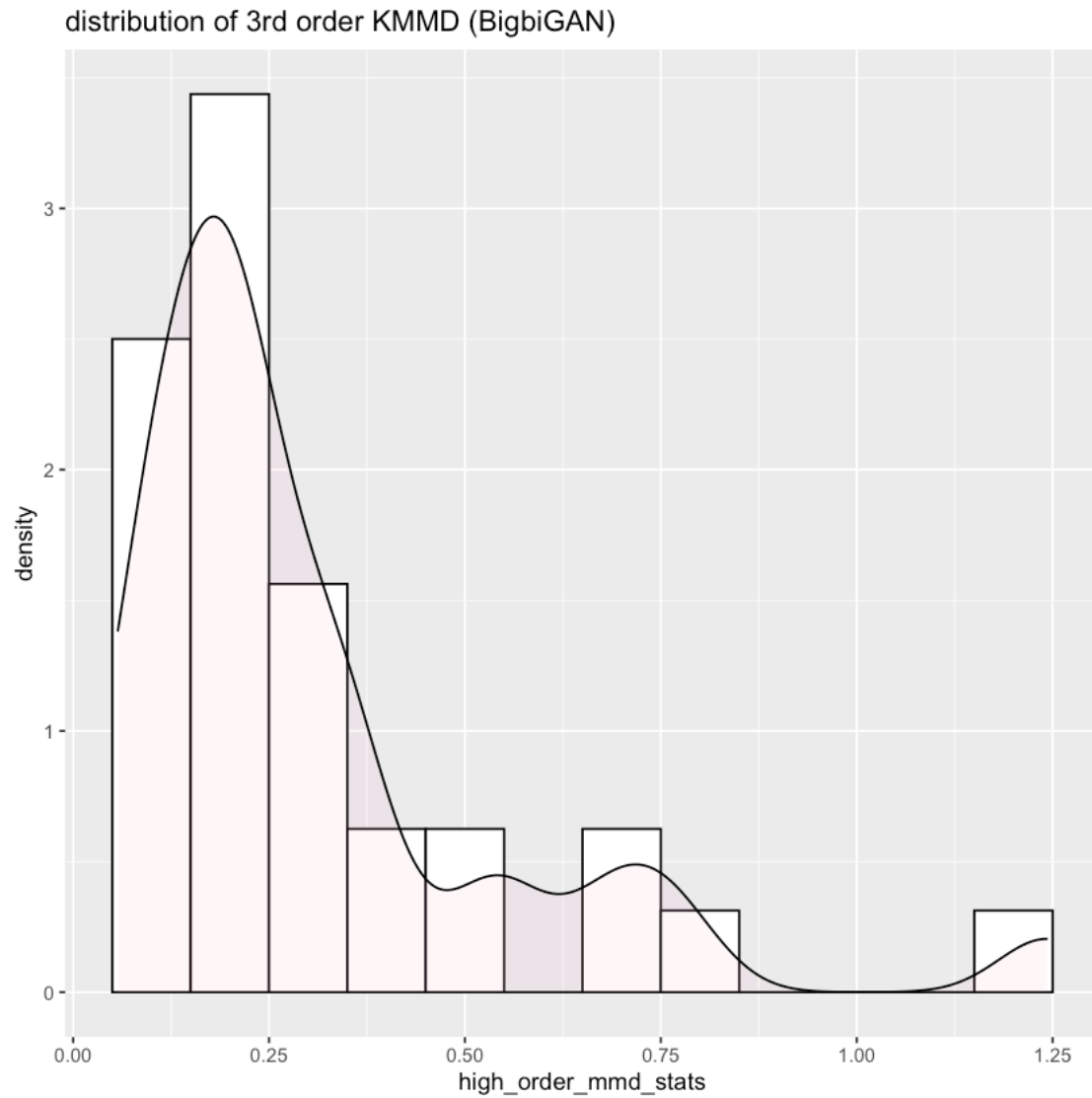
```



```

[60]: ggplot(df_high_order_mmd_stats, aes(x=high_order_mmd_stats)) +
    labs(title="distribution of 3rd order KMMD (BigbiGAN)") +
    geom_histogram(aes(y=..density..), # Histogram with density instead of
    ↪ count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
    geom_density(alpha=.1, fill="pink") # Overlay with transparent density plot

```



```
[113]: summary(df_low_order_mmd_stats)
```

```
high_order_mmd_stats
Min.   :0.06683
1st Qu.:0.12282
Median :0.17194
Mean   :0.23591
3rd Qu.:0.32188
Max.   :0.79631
```

```
[114]: summary(df_high_order_mmd_stats)
```

```
high_order_mmd_stats
```

```
Min.      :0.06683
1st Qu.   :0.12282
Median    :0.17194
Mean      :0.23591
3rd Qu.   :0.32188
Max.      :0.79631
```

1.4 Result of Flowers dataset for SNGan (iteration 500)

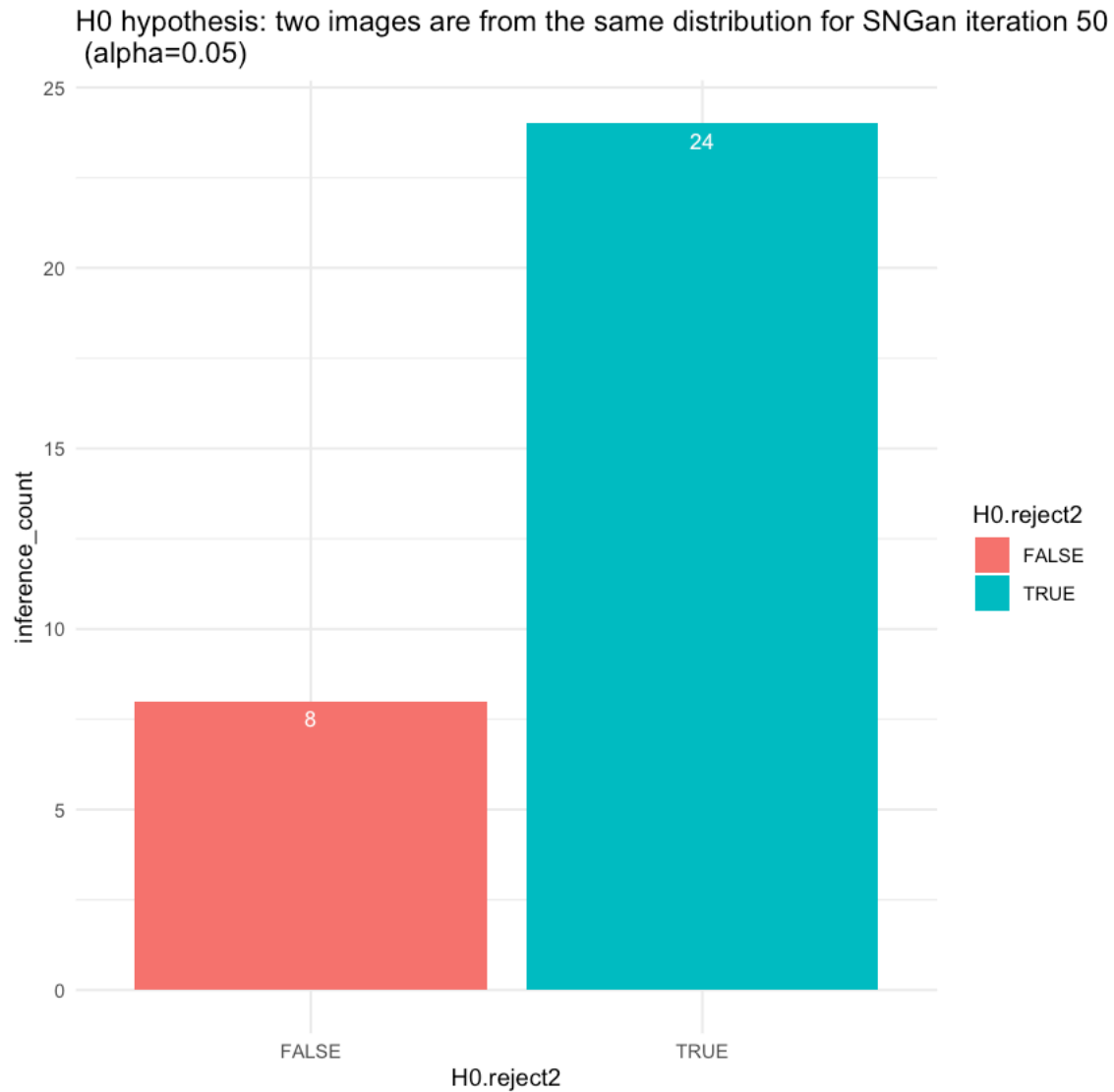
```
[3]: result2 = compute_kmmd("./sngan_flower_results/input", "./sngan_flower_results/
    ↪iter_500")
```

[illegible]


```
[4]: H0.reject2 = result2$H0.reject
df_H0.reject2 = as.data.frame(H0.reject2) %>% group_by(H0.reject2) %>% count()
  ↳ %>% rename("inference_count"=n)
low_order_mmd_stats2 = result2$low_order_mmd_stats
df_low_order_mmd_stats2 = as.data.frame(low_order_mmd_stats2)
high_order_mmd_stats2 = result2$high_order_mmd_stats
df_high_order_mmd_stats2 = as.data.frame(high_order_mmd_stats2)
```

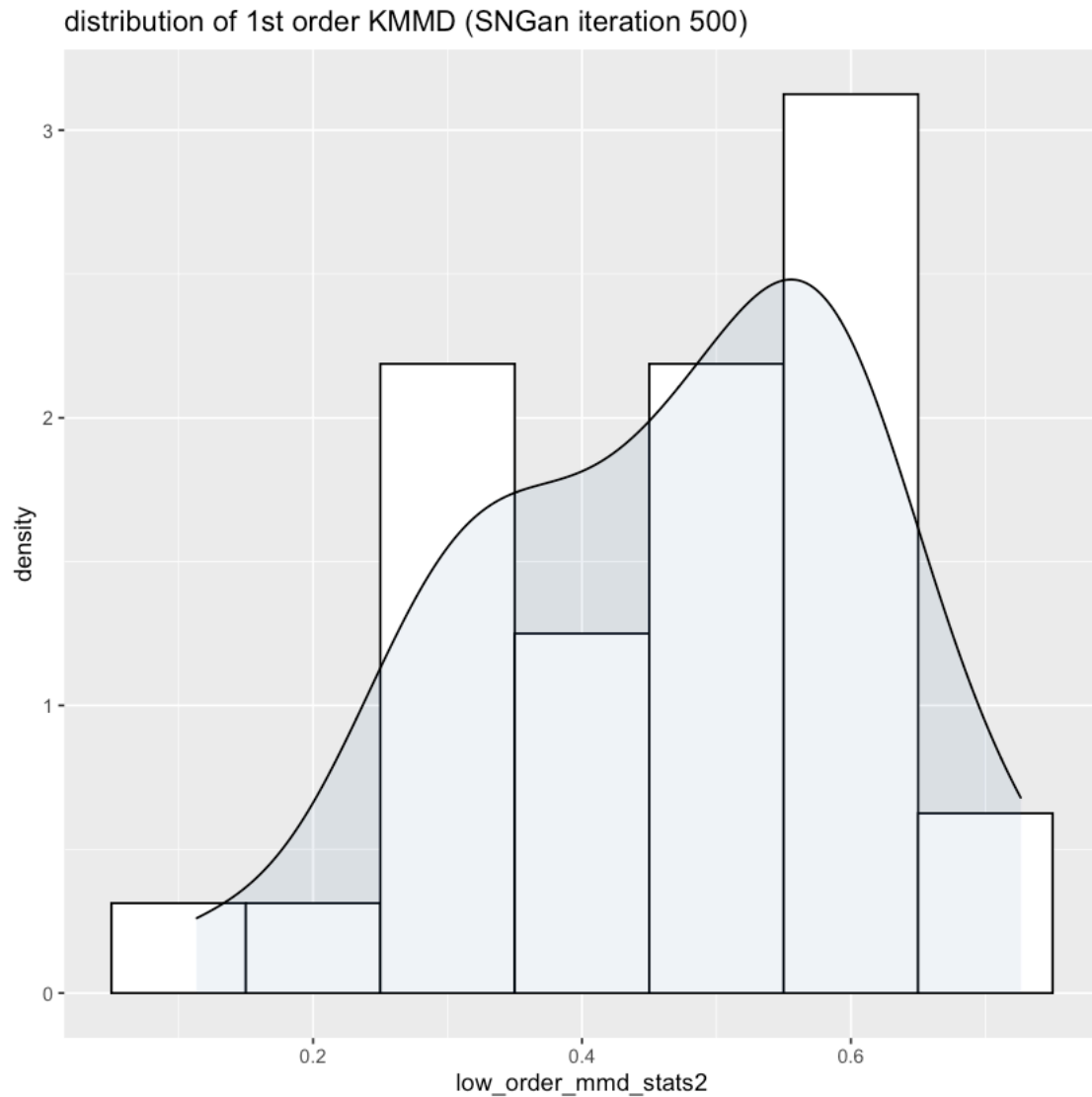
1.4.1 Hypothesis test result

```
[5]: ggplot(data=df_H0.reject2, aes(x=H0.reject2, y=inference_count, fill=H0.
  ↳ reject2)) +
  labs(title="H0 hypothesis: two images are from the same distribution for
  ↳ SNGan iteration 500\n (alpha=0.05)") +
  geom_bar(stat="identity")+
  geom_text(aes(label=inference_count), vjust=1.6, color="white", size=3.5)+
  theme_minimal()
```

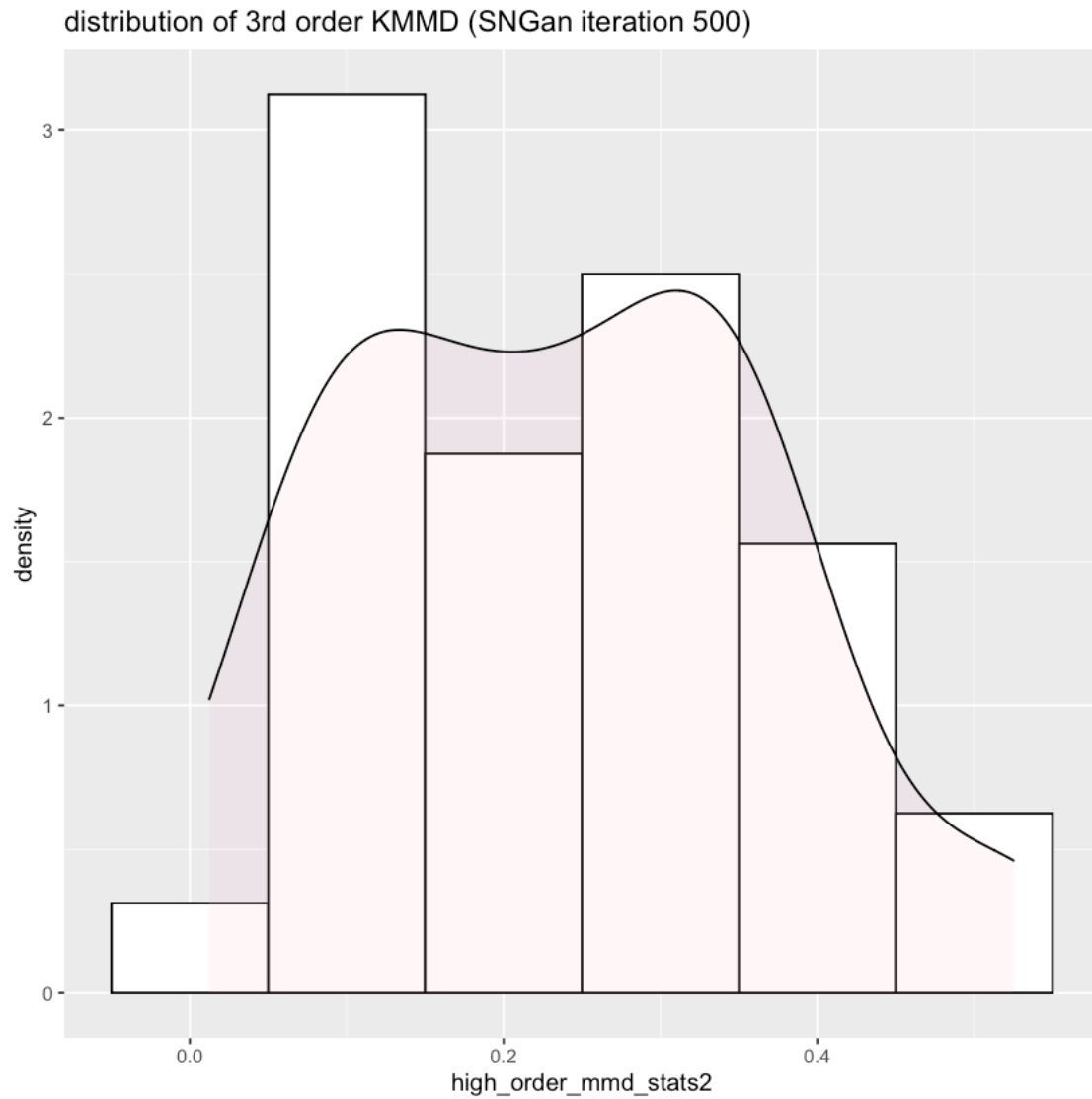


1.4.2 MMD statistics

```
[6]: ggplot(df_low_order_mmd_stats2, aes(x=low_order_mmd_stats2)) +
      labs(title="distribution of 1st order KMMD (SNGan iteration 500)") +
      geom_histogram(aes(y=..density..),          # Histogram with density instead of
      ↪count on y-axis
                     binwidth=.1,
                     colour="black", fill="white") +
      geom_density(alpha=.1, fill="steelblue")    # Overlay with transparent
      ↪density plot
```



```
[7]: ggplot(df_high_order_mmd_stats2, aes(x=high_order_mmd_stats2)) +
  labs(title="distribution of 3rd order KMMD (SNGan iteration 500)") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↪count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="pink") # Overlay with transparent density plot
```



```
[8]: summary(df_low_order_mmd_stats2)
```

```
low_order_mmd_stats2
Min.   :0.1134
1st Qu.:0.3446
Median :0.4767
Mean   :0.4678
3rd Qu.:0.5790
Max.   :0.7262
```

```
[9]: summary(df_high_order_mmd_stats2)
```

```
high_order_mmd_stats2
```

```
Min.      :0.01248
1st Qu.   :0.11638
Median    :0.22464
Mean      :0.23761
3rd Qu.   :0.33200
Max.      :0.52510
```

1.5 Result of Face dataset for SNGan (iteration 500)

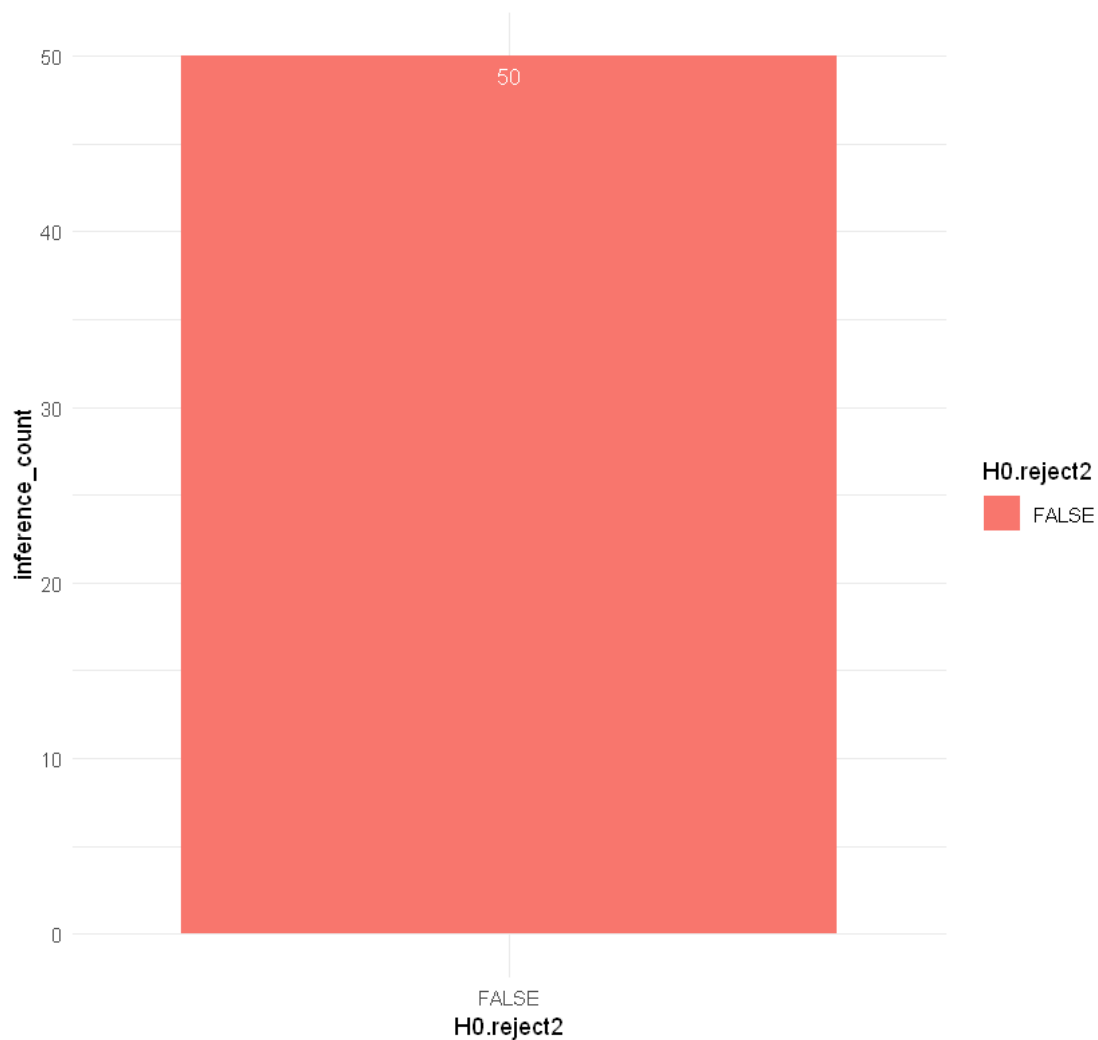
```
[6]: result2 = compute_kmmd("./sngan_face_results/input", "./sngan_face_results/
    ↪iter_500")

H0.reject2 = result2$H0.reject
df_H0.reject2 = as.data.frame(H0.reject2) %>% group_by(H0.reject2) %>% count()
    ↪%>% rename("inference_count"=n)

low_order_mmd_stats2 = result2$low_order_mmd_stats
df_low_order_mmd_stats2 = as.data.frame(low_order_mmd_stats2)
high_order_mmd_stats2 = result2$high_order_mmd_stats
df_high_order_mmd_stats2 = as.data.frame(high_order_mmd_stats2)
```

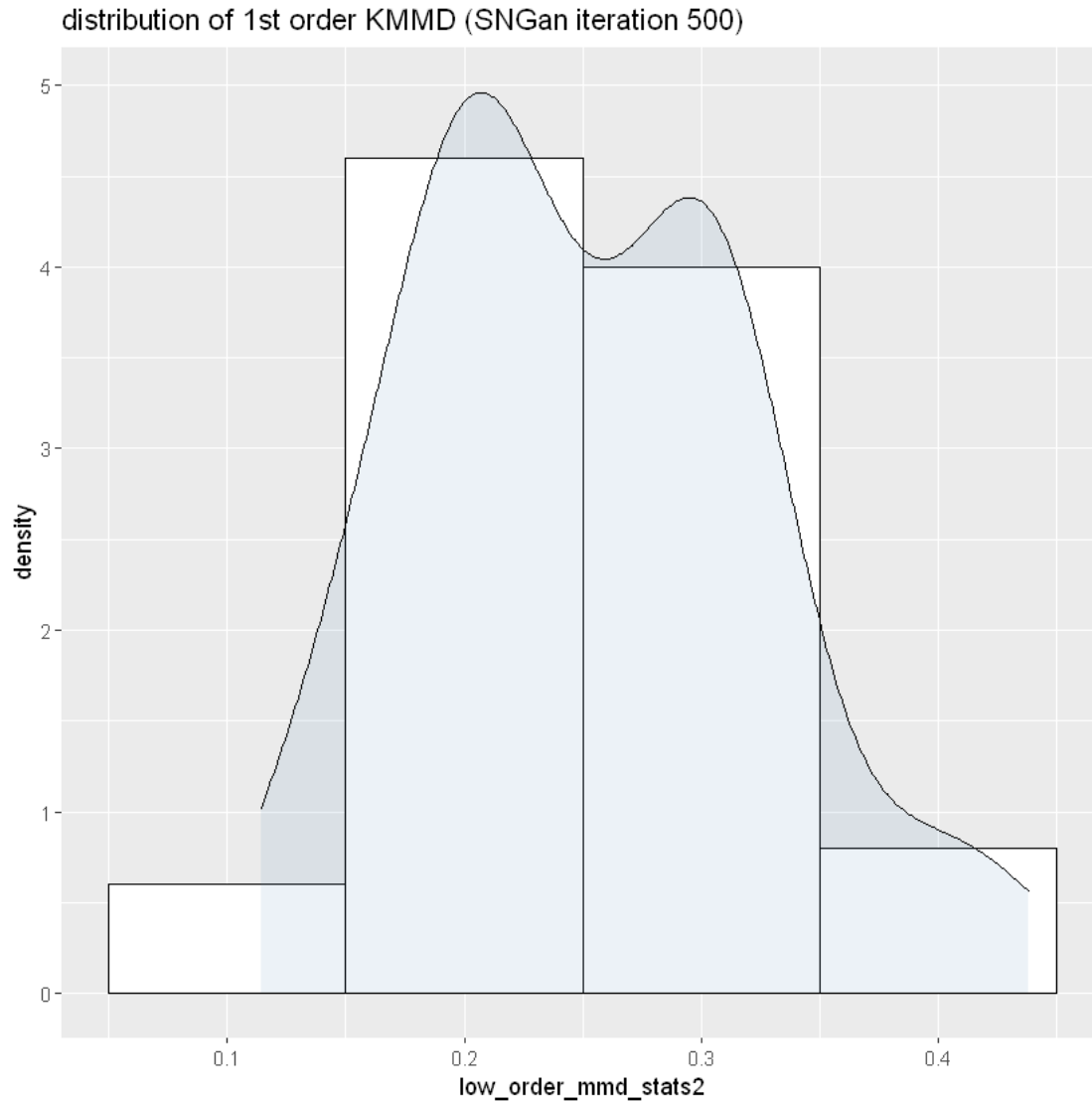
[illegible]

H0 hypothesis: two images are from the same distribution for SNGan iteration 500 (alpha=0.05)

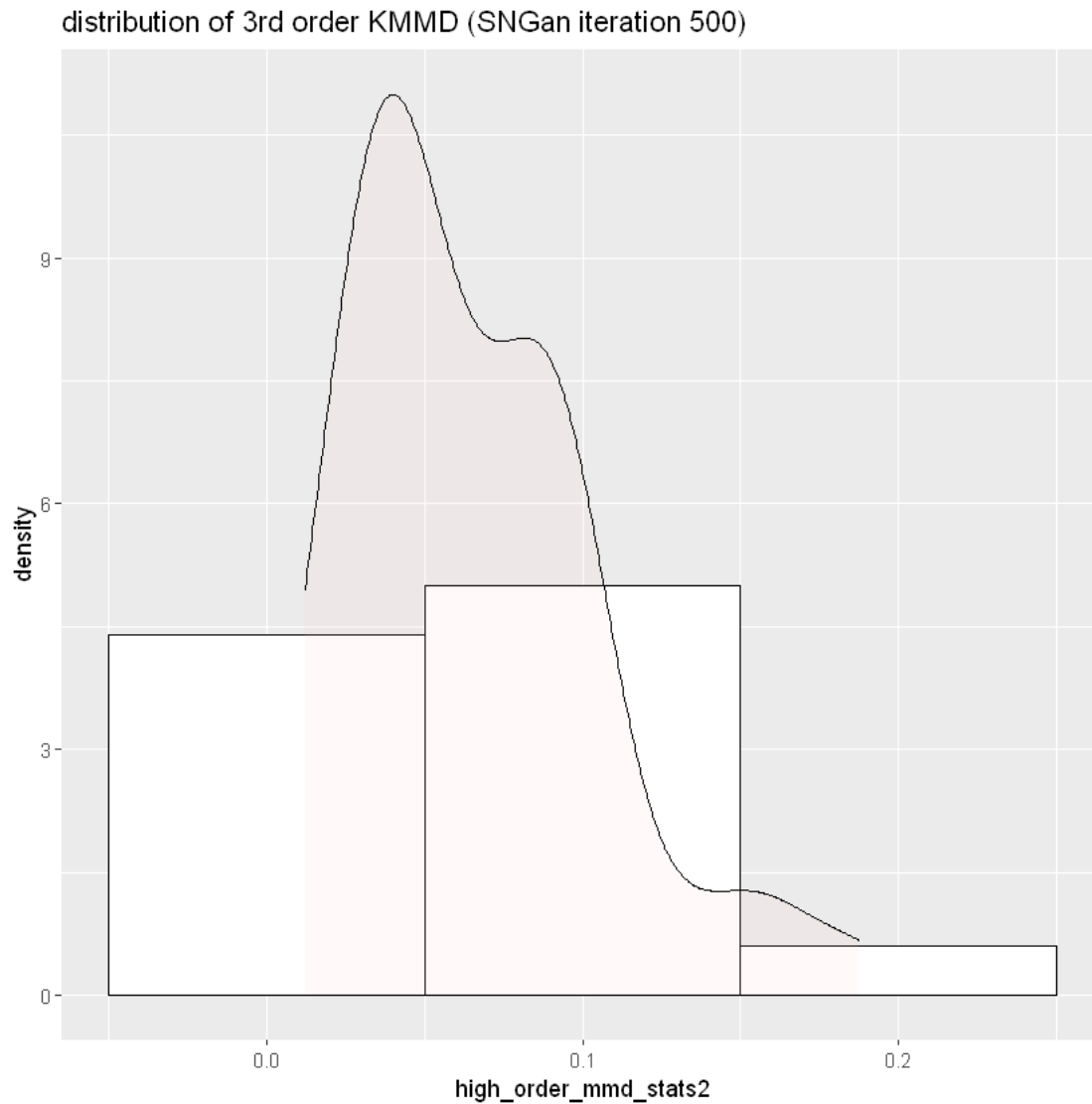


1.5.2 MMD statistics

```
[8]: ggplot(df_low_order_mmd_stats2, aes(x=low_order_mmd_stats2)) +
      labs(title="distribution of 1st order KMMD (SNGan iteration 500)") +
      geom_histogram(aes(y=..density..),          # Histogram with density instead of
      ↪ count on y-axis
                     binwidth=.1,
                     colour="black", fill="white") +
      geom_density(alpha=.1, fill="steelblue") # Overlay with transparent
      ↪ density plot
```



```
[9]: ggplot(df_high_order_mmd_stats2, aes(x=high_order_mmd_stats2)) +
  labs(title="distribution of 3rd order KMMD (SNGan iteration 500)") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↪count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="pink") # Overlay with transparent density plot
```

```
[10]: summary(df_low_order_mmd_stats2)
```

```
low_order_mmd_stats2  
Min.   :0.1141  
1st Qu.:0.1986  
Median :0.2429  
Mean   :0.2525  
3rd Qu.:0.3082  
Max.   :0.4382
```

```
[11]: summary(df_high_order_mmd_stats2)
```

```
high_order_mmd_stats2
```

```
Min.      :0.01207
1st Qu.   :0.03759
Median    :0.05575
Mean      :0.06602
3rd Qu.   :0.09131
Max.      :0.18750
```

1.6 Result of Anime dataset for SNGan (iteration 500)

```
[12]: result2 = compute_kmmd("./sngan_anime_results/input", "./sngan_anime_results/
    ↪iter_500")
```

[illegible]

```

Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel

```

```

[13]: H0.reject2 = result2$H0.reject
df_H0.reject2 = as.data.frame(H0.reject2) %>% group_by(H0.reject2) %>% count()
  ↳ %>% rename("inference_count"=n)
low_order_mmd_stats2 = result2$low_order_mmd_stats
df_low_order_mmd_stats2 = as.data.frame(low_order_mmd_stats2)
high_order_mmd_stats2 = result2$high_order_mmd_stats
df_high_order_mmd_stats2 = as.data.frame(high_order_mmd_stats2)

```

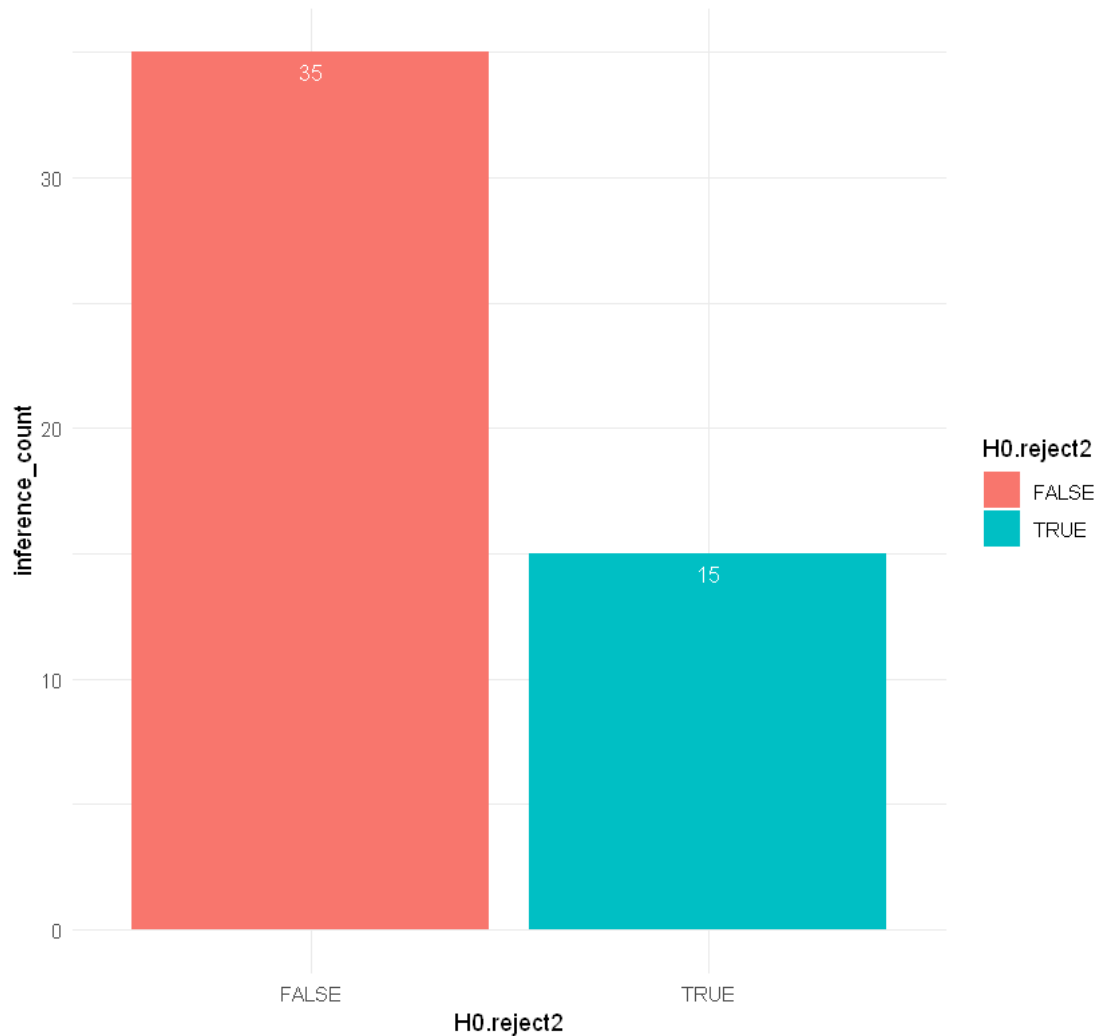
1.6.1 Hypothesis test result

```

[15]: ggplot(data=df_H0.reject2, aes(x=H0.reject2, y=inference_count, fill=H0.
  ↳ reject2)) +
  labs(title="H0 hypothesis: two images are from the same distribution for_
  ↳ SNGan iteration 500\n (alpha=0.05)") +
  geom_bar(stat="identity")+
  geom_text(aes(label=inference_count), vjust=1.6, color="white", size=3.5)+
  theme_minimal()

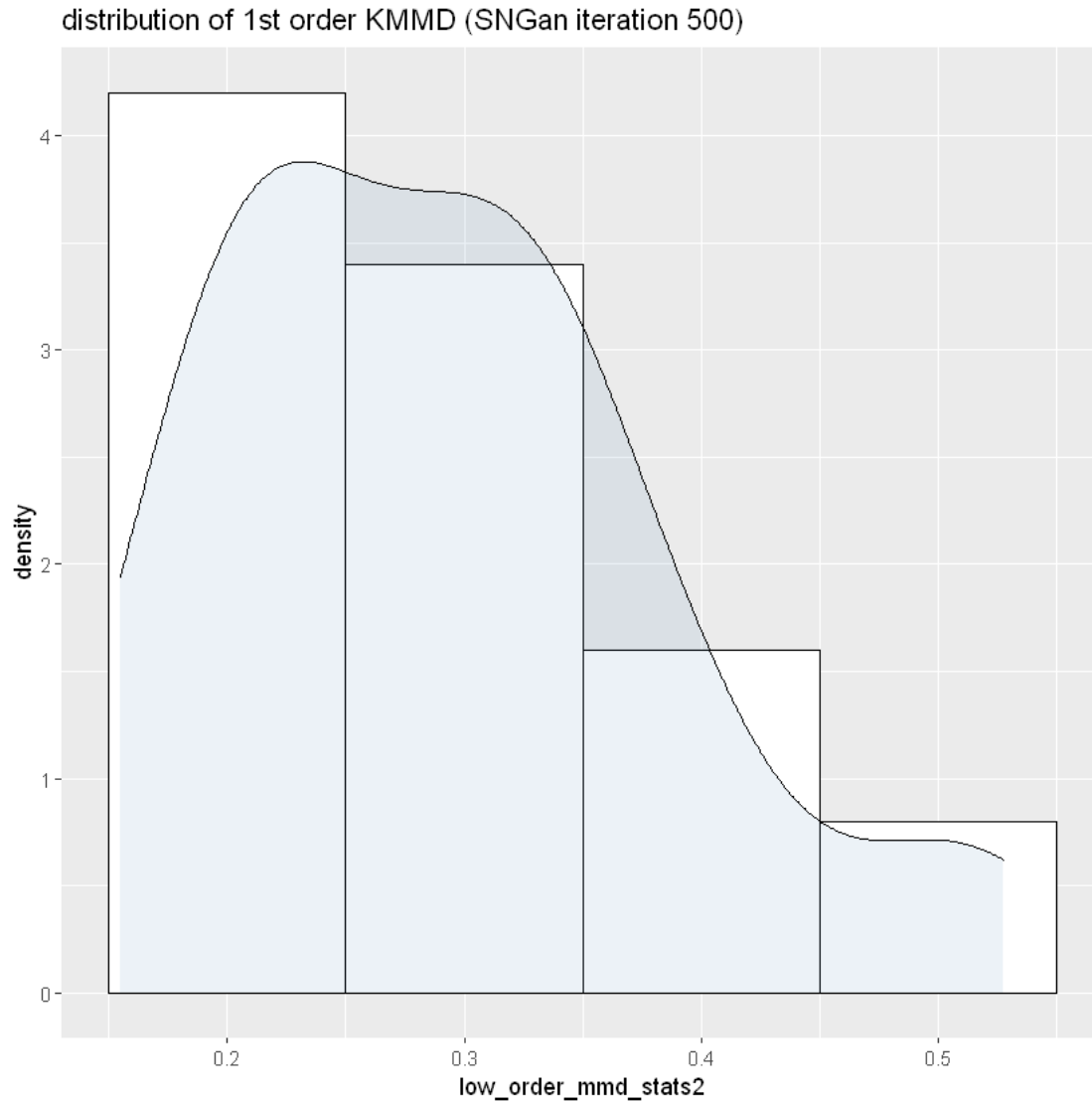
```

H0 hypothesis: two images are from the same distribution for SNGan iteration 500 (alpha=0.05)

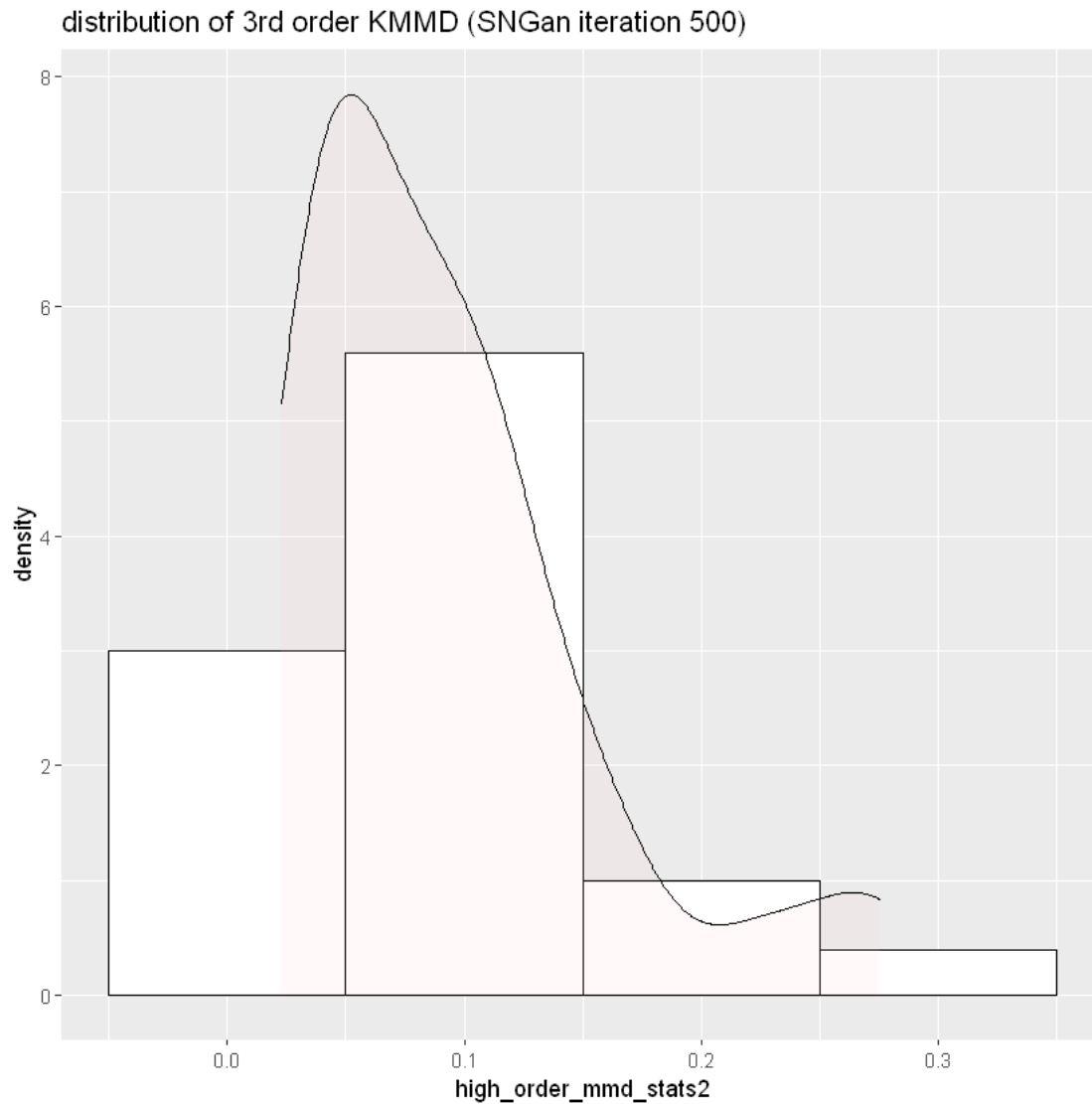


1.6.2 MMD statistics

```
[16]: ggplot(df_low_order_mmd_stats2, aes(x=low_order_mmd_stats2)) +
  labs(title="distribution of 1st order KMMD (SNGan iteration 500)") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↪count on y-axis
              binwidth=.1,
              colour="black", fill="white") +
  geom_density(alpha=.1, fill="steelblue")    # Overlay with transparent
  ↪density plot
```



```
[17]: ggplot(df_high_order_mmd_stats2, aes(x=high_order_mmd_stats2)) +
  labs(title="distribution of 3rd order KMMD (SNGan iteration 500)") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↪count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="pink") # Overlay with transparent density plot
```



```
[18]: summary(df_low_order_mmd_stats2)
```

```
low_order_mmd_stats2
Min.   :0.1547
1st Qu.:0.2143
Median :0.2832
Mean   :0.2923
3rd Qu.:0.3451
Max.   :0.5278
```

```
[19]: summary(df_high_order_mmd_stats2)
```

```
high_order_mmd_stats2
```

```
Min.      :0.02294
1st Qu.   :0.04413
Median    :0.07810
Mean      :0.09198
3rd Qu.   :0.11654
Max.      :0.27561
```

1.7 Result of Flowers dataset for SNGan (iteration 2000)

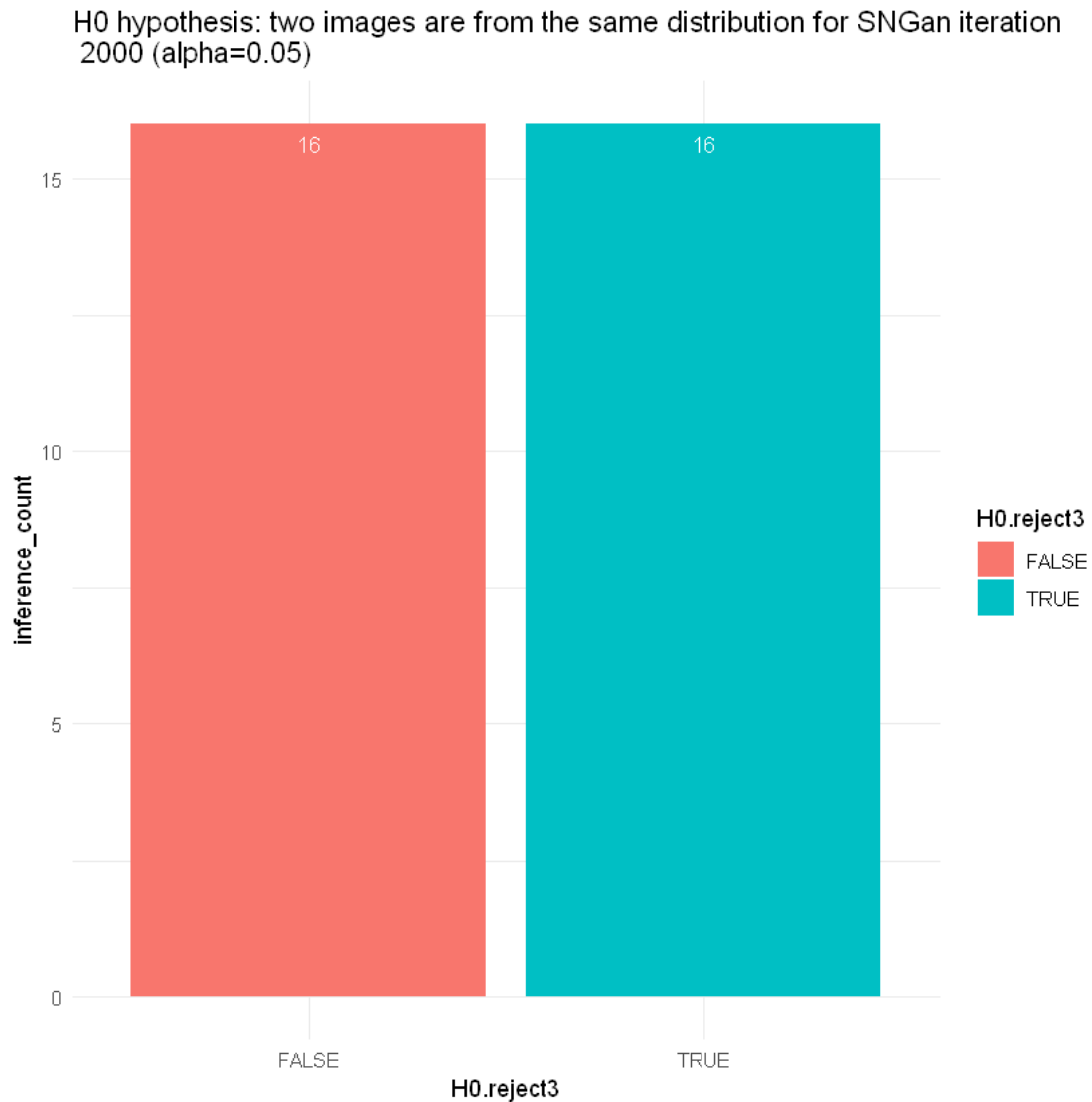
```
[20]: result3 = compute_kmmd("./sngan_flower_results/input", "./sngan_flower_results/
    ↪iter 2000")
```

[illegible]

```
[21]: H0.reject3 = result3$H0.reject
df_H0.reject3 = as.data.frame(H0.reject3) %>% group_by(H0.reject3) %>% count()
↳ %>% rename("inference_count"=n)
low_order_mmd_stats3 = result3$low_order_mmd_stats
df_low_order_mmd_stats3 = as.data.frame(low_order_mmd_stats3)
high_order_mmd_stats3 = result3$high_order_mmd_stats
df_high_order_mmd_stats3 = as.data.frame(high_order_mmd_stats3)
```

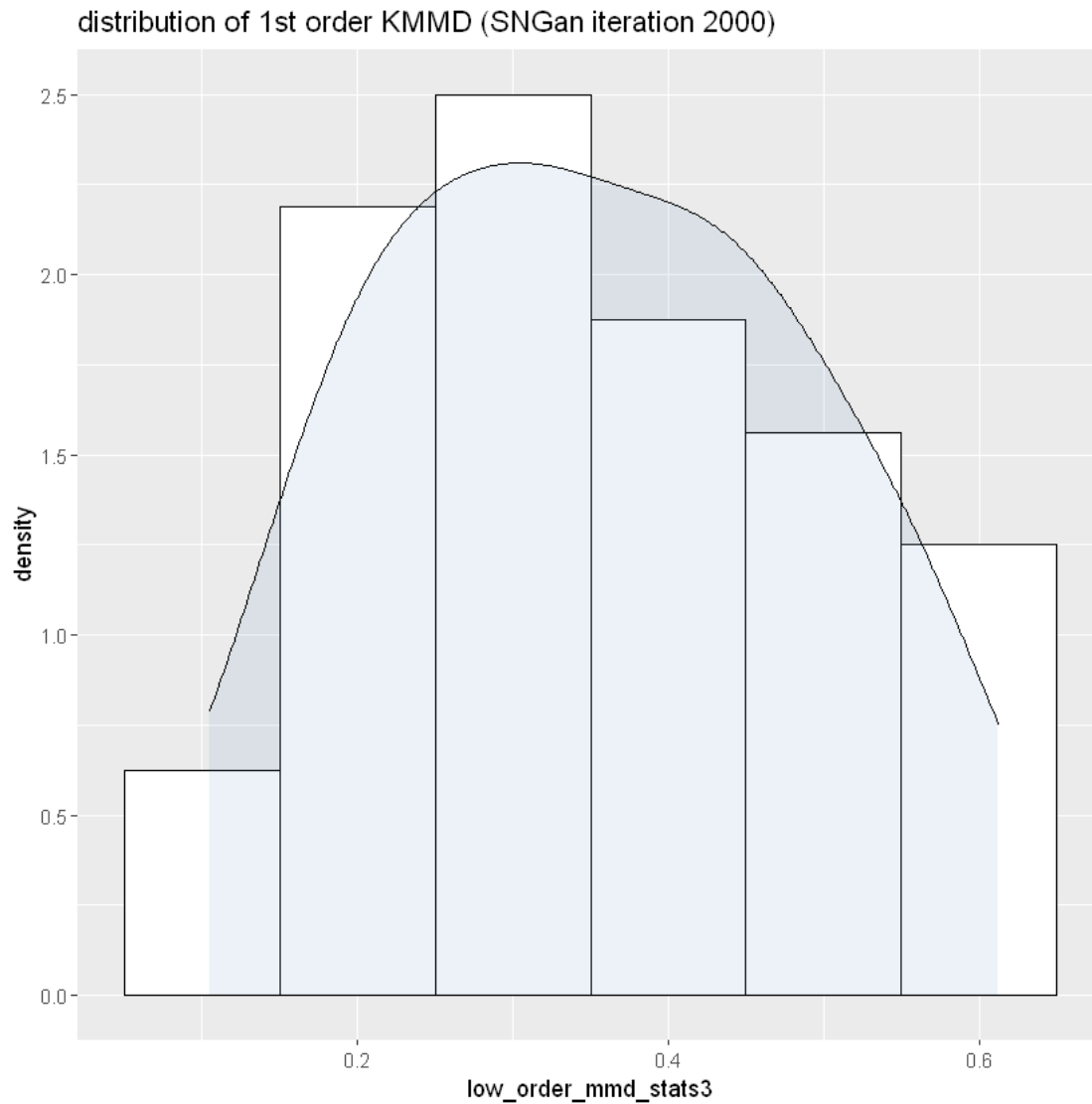
1.7.1 Hypothesis test result

```
[22]: ggplot(data=df_H0.reject3, aes(x=H0.reject3, y=inference_count, fill=H0.
↳ reject3)) +
  labs(title="H0 hypothesis: two images are from the same distribution for
↳ SNGan iteration\n 2000 (alpha=0.05)") +
  geom_bar(stat="identity")+
  geom_text(aes(label=inference_count), vjust=1.6, color="white", size=3.5)+
  theme_minimal()
```

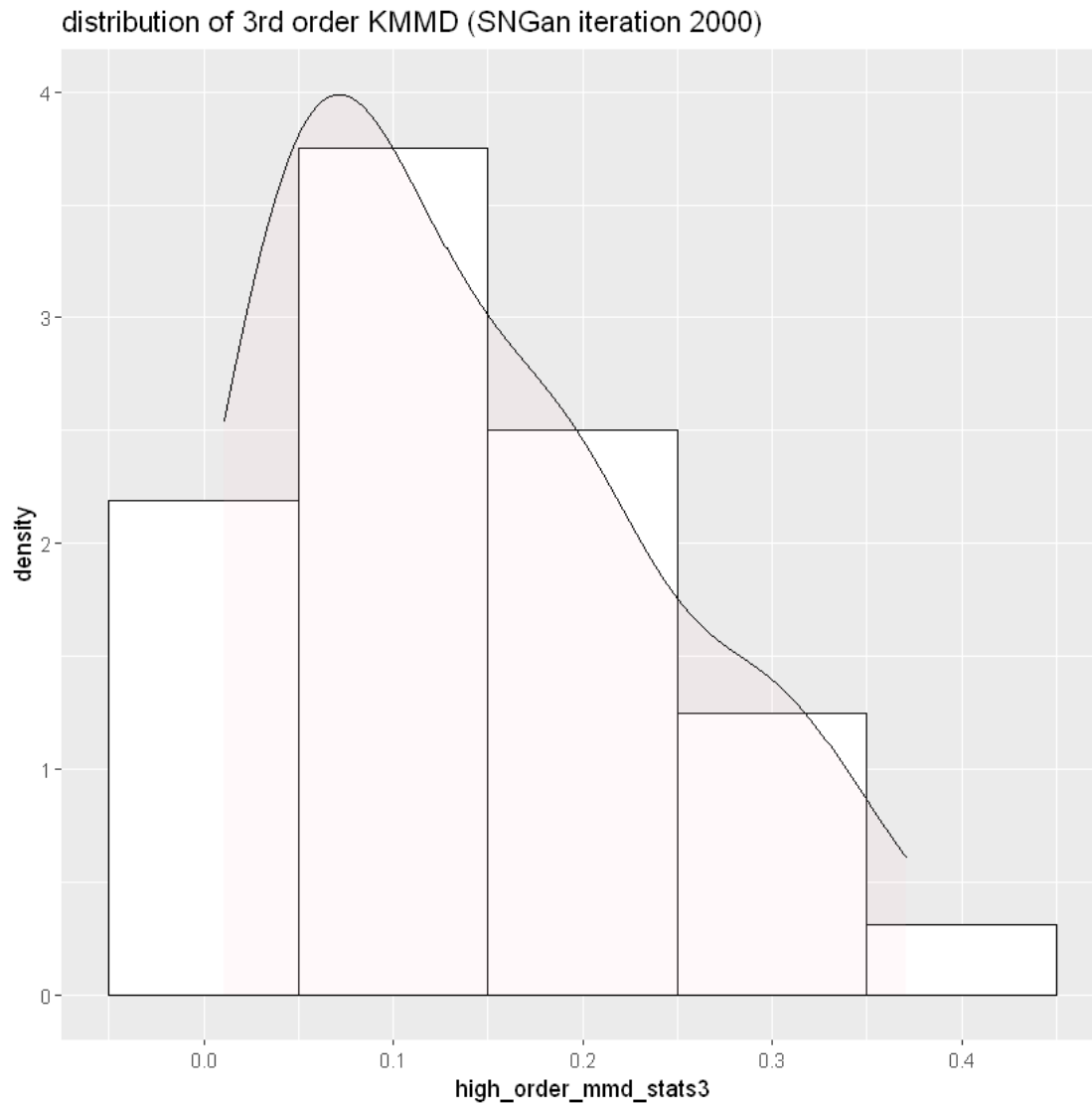



1.7.2 MMD statistics

```
[23]: ggplot(df_low_order_mmd_stats3, aes(x=low_order_mmd_stats3)) +
  labs(title="distribution of 1st order KMMD (SNGan iteration 2000)") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↪ count on y-axis
               binwidth=.1,
               colour="black", fill="white") +
  geom_density(alpha=.1, fill="steelblue")    # Overlay with transparent
  ↪ density plot
```



```
[24]: ggplot(df_high_order_mmd_stats3, aes(x=high_order_mmd_stats3)) +
  labs(title="distribution of 3rd order KMMD (SNGan iteration 2000)") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↪count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="pink") # Overlay with transparent density plot
```



```
[25]: summary(df_low_order_mmd_stats3)
```

```
low_order_mmd_stats3  
Min.   :0.1050  
1st Qu.:0.2437  
Median :0.3428  
Mean   :0.3525  
3rd Qu.:0.4604  
Max.   :0.6123
```

```
[26]: summary(df_high_order_mmd_stats3)
```

```
high_order_mmd_stats3
```

```
Min.      :0.01075
1st Qu.   :0.05812
Median    :0.11494
Mean      :0.14001
3rd Qu.   :0.20875
Max.      :0.37065
```

1.8 Result of Face dataset for SNGan (iteration 2000)

```
[27]: result3 = compute_kmmd("./sngan_face_results/input", "./sngan_face_results/
↳ iter_2000")
```

[illegible]

```

Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel

```

```

[28]: H0.reject3 = result3$H0.reject
df_H0.reject3 = as.data.frame(H0.reject3) %>% group_by(H0.reject3) %>% count()
  ↪ %>% rename("inference_count"=n)
low_order_mmd_stats3 = result3$low_order_mmd_stats
df_low_order_mmd_stats3 = as.data.frame(low_order_mmd_stats3)
high_order_mmd_stats3 = result3$high_order_mmd_stats
df_high_order_mmd_stats3 = as.data.frame(high_order_mmd_stats3)

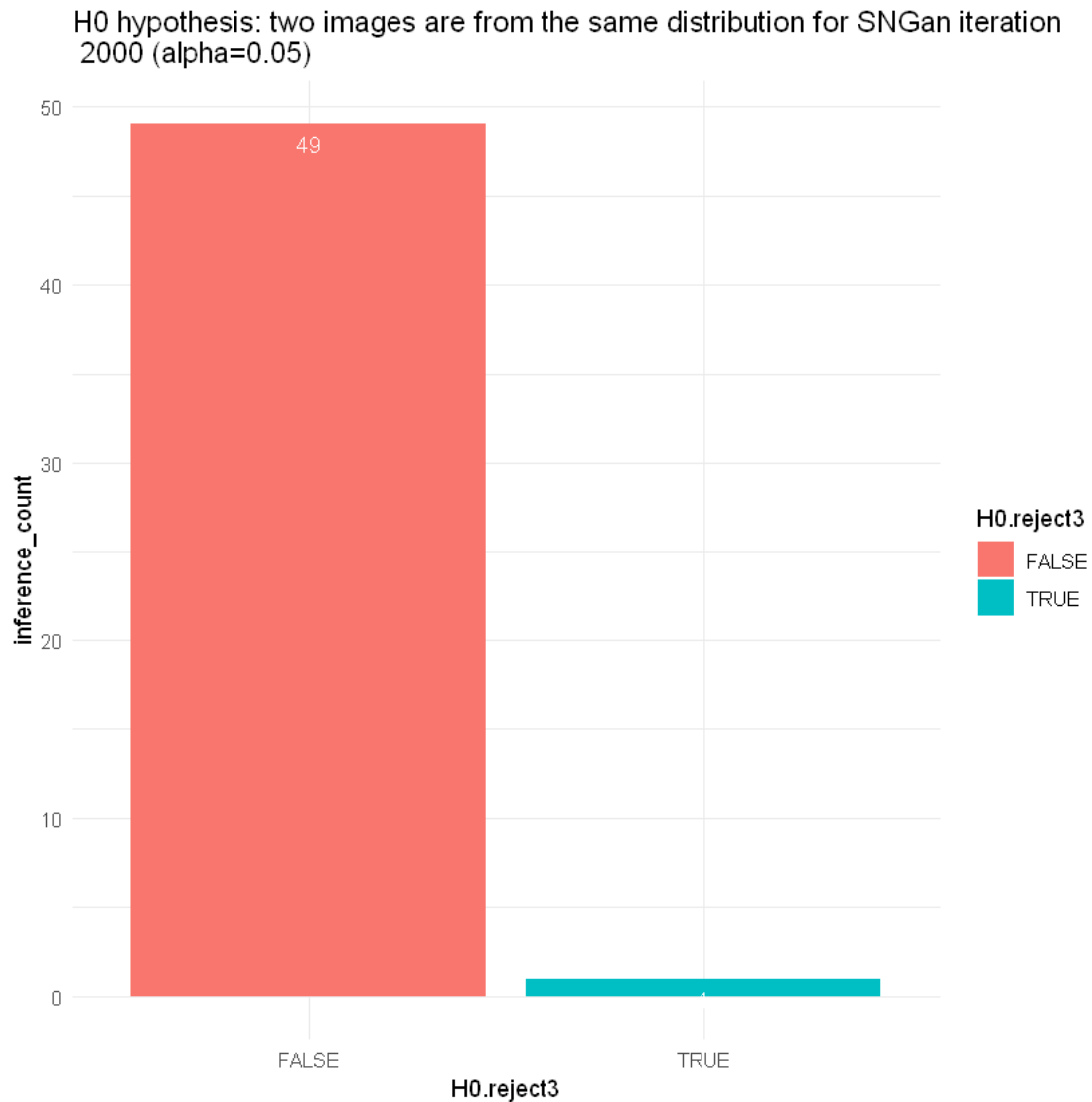
```

1.8.1 Hypothesis test result

```

[29]: ggplot(data=df_H0.reject3, aes(x=H0.reject3, y=inference_count, fill=H0.
  ↪ reject3)) +
  labs(title="H0 hypothesis: two images are from the same distribution for
  ↪ SNGan iteration\n 2000 (alpha=0.05)") +
  geom_bar(stat="identity")+
  geom_text(aes(label=inference_count), vjust=1.6, color="white", size=3.5)+
  theme_minimal()

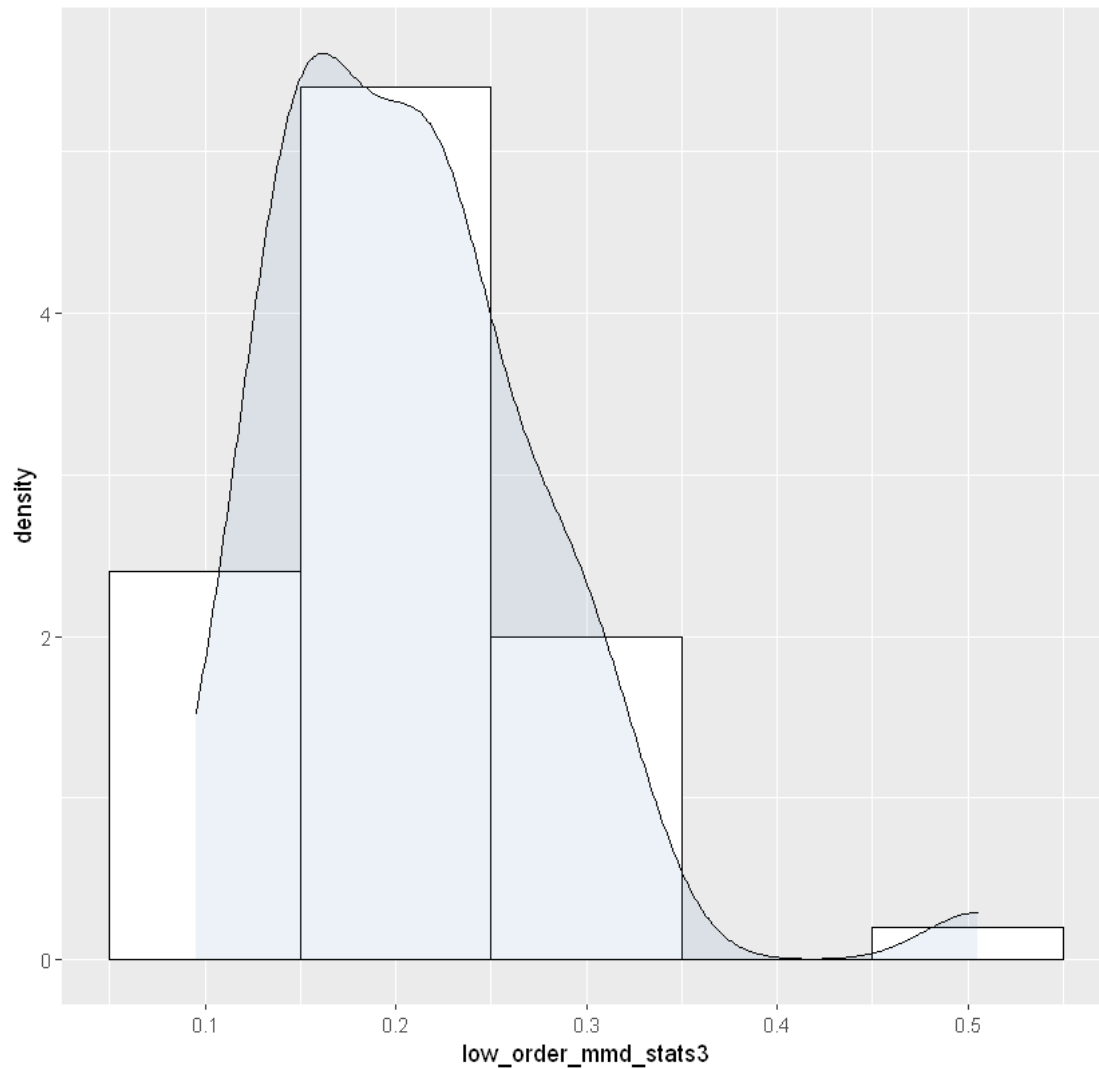
```



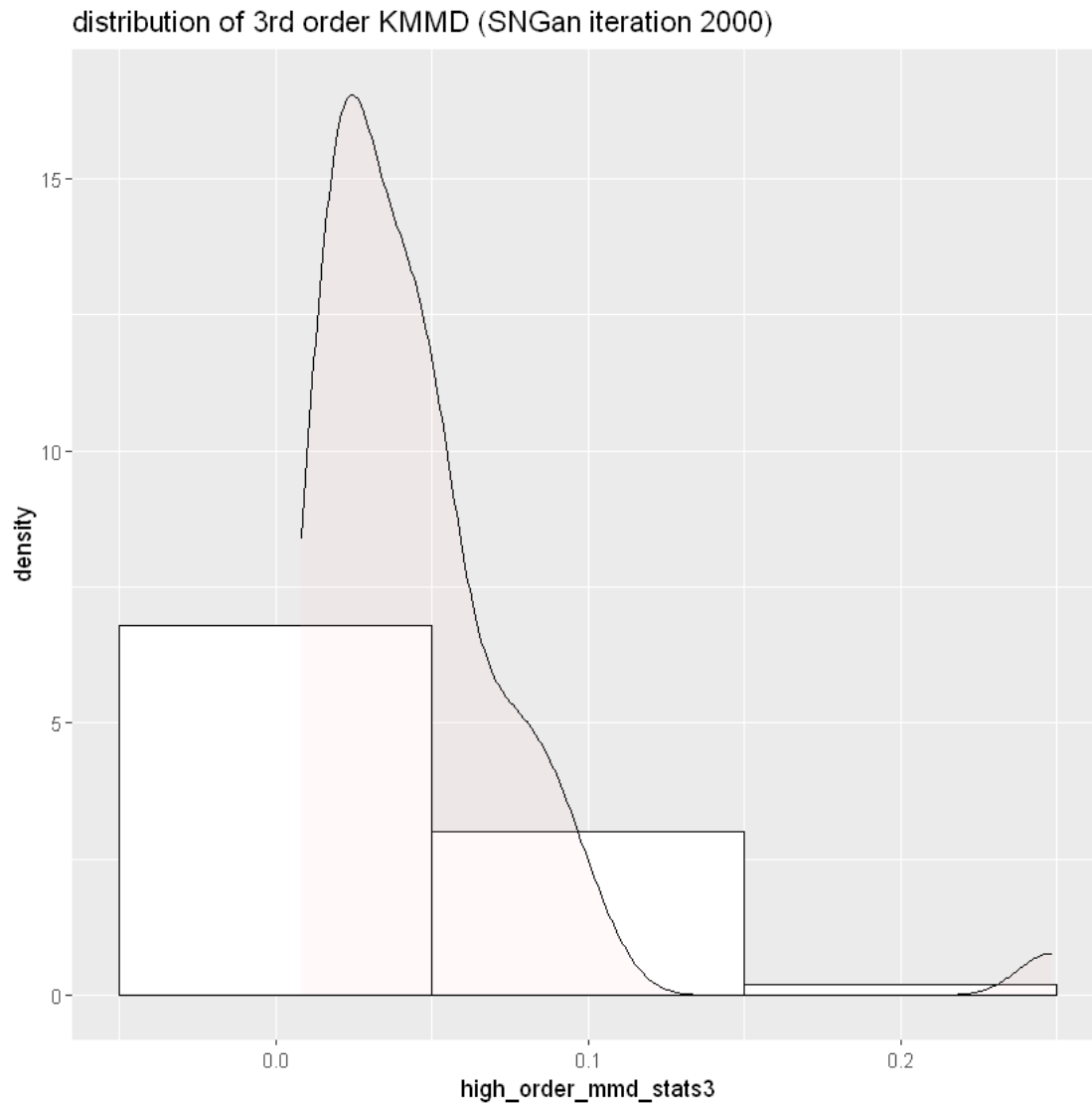
1.8.2 MMD statistics

```
[30]: ggplot(df_low_order_mmd_stats3, aes(x=low_order_mmd_stats3)) +
  labs(title="distribution of 1st order KMMD (SNGan iteration 2000)") +
  geom_histogram(aes(y=..density..),      # Histogram with density instead of
  ↪count on y-axis
               binwidth=.1,
               colour="black", fill="white") +
  geom_density(alpha=.1, fill="steelblue") # Overlay with transparent
  ↪density plot
```

distribution of 1st order KMMD (SNGan iteration 2000)



```
[31]: ggplot(df_high_order_mmd_stats3, aes(x=high_order_mmd_stats3)) +
  labs(title="distribution of 3rd order KMMD (SNGan iteration 2000)") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↪count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="pink") # Overlay with transparent density plot
```



```
[32]: summary(df_low_order_mmd_stats3)
```

```
low_order_mmd_stats3
Min.   :0.09513
1st Qu.:0.15050
Median :0.20370
Mean   :0.20815
3rd Qu.:0.24009
Max.   :0.50502
```

```
[33]: summary(df_high_order_mmd_stats3)
```

```
high_order_mmd_stats3
```



```
Min.      :0.008101
1st Qu.   :0.020882
Median    :0.039048
Mean      :0.045939
3rd Qu.   :0.054713
Max.      :0.248085
```

1.9 Result of Anime dataset for SNGan (iteration 2000)

```
[34]: result3 = compute_kmmd("./sngan_anime_results/input", "./sngan_anime_results/
↳ iter_2000")
```

[illegible]

```

Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel

```

```

[39]: H0.reject3 = result3$H0.reject
df_H0.reject3 = as.data.frame(H0.reject3) %>% group_by(H0.reject3) %>% count()
  ↳ %>% rename("inference_count"=n)
low_order_mmd_stats3 = result3$low_order_mmd_stats
df_low_order_mmd_stats3 = as.data.frame(low_order_mmd_stats3)
high_order_mmd_stats3 = result3$high_order_mmd_stats
df_high_order_mmd_stats3 = as.data.frame(high_order_mmd_stats3)

```

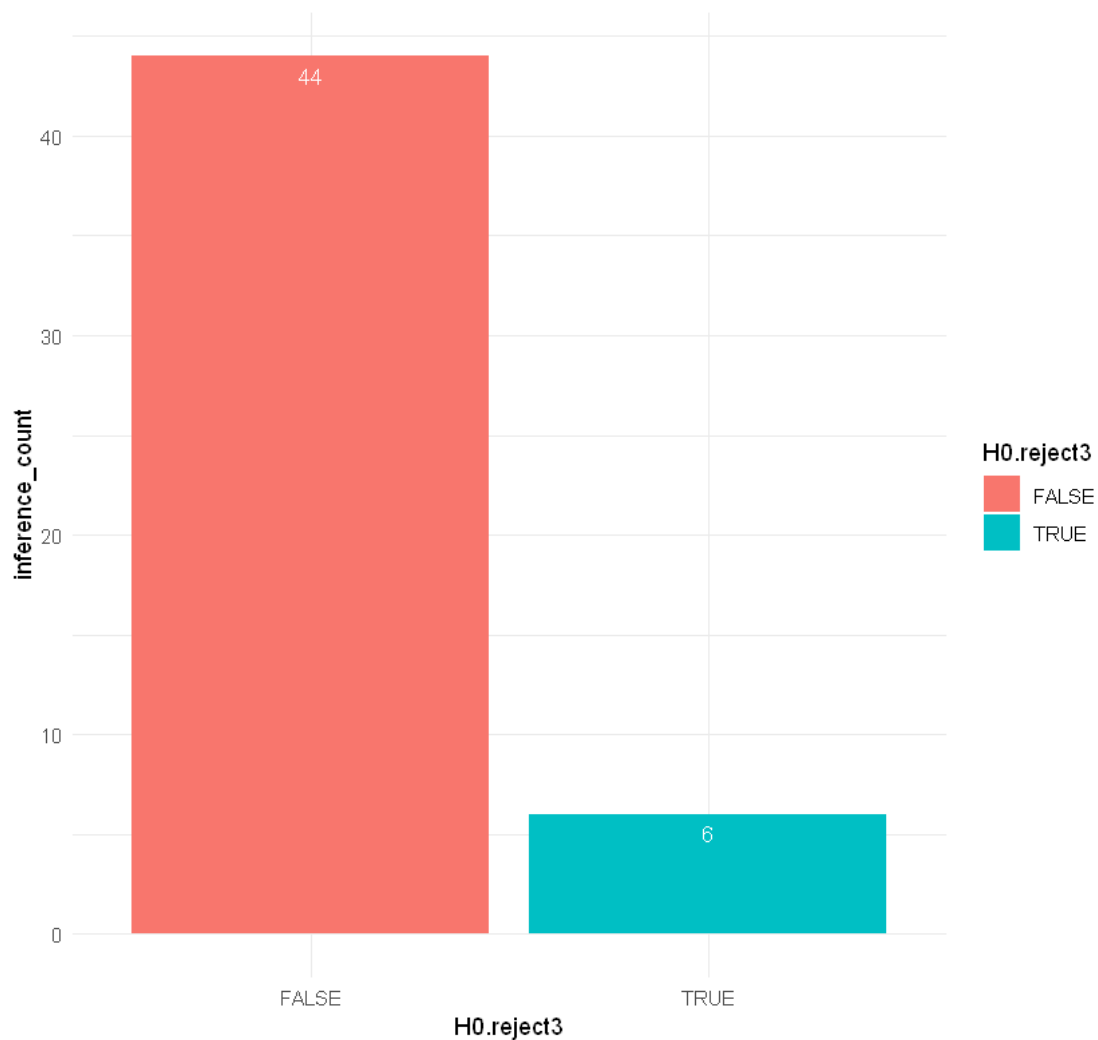
1.9.1 Hypothesis test result

```

[40]: ggplot(data=df_H0.reject3, aes(x=H0.reject3, y=inference_count, fill=H0.
  ↳ reject3)) +
  labs(title="H0 hypothesis: two images are from the same distribution for
  ↳ SNGan iteration\n 2000 (alpha=0.05)") +
  geom_bar(stat="identity")+
  geom_text(aes(label=inference_count), vjust=1.6, color="white", size=3.5)+
  theme_minimal()

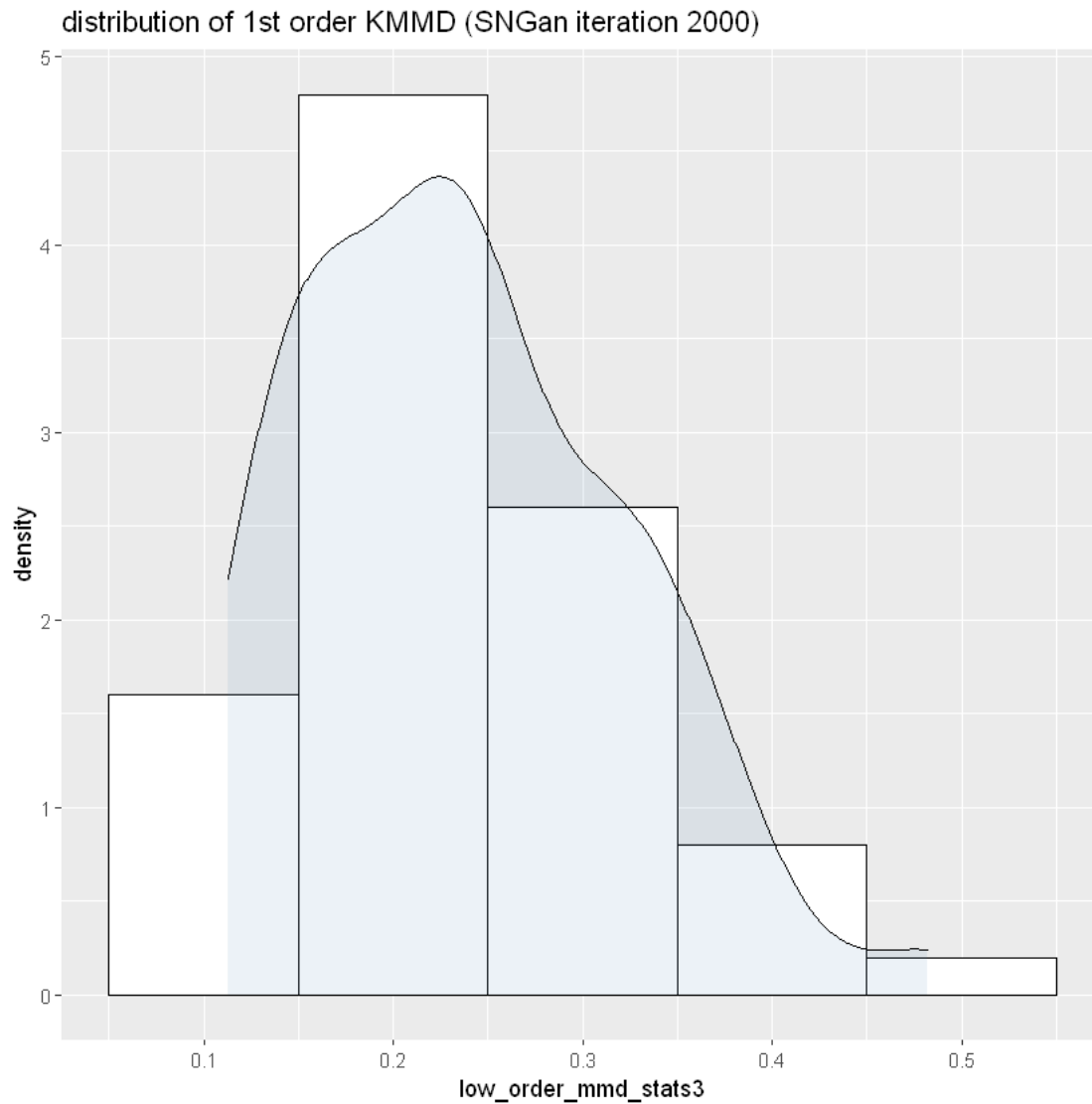
```

H0 hypothesis: two images are from the same distribution for SNGan iteration 2000 (alpha=0.05)

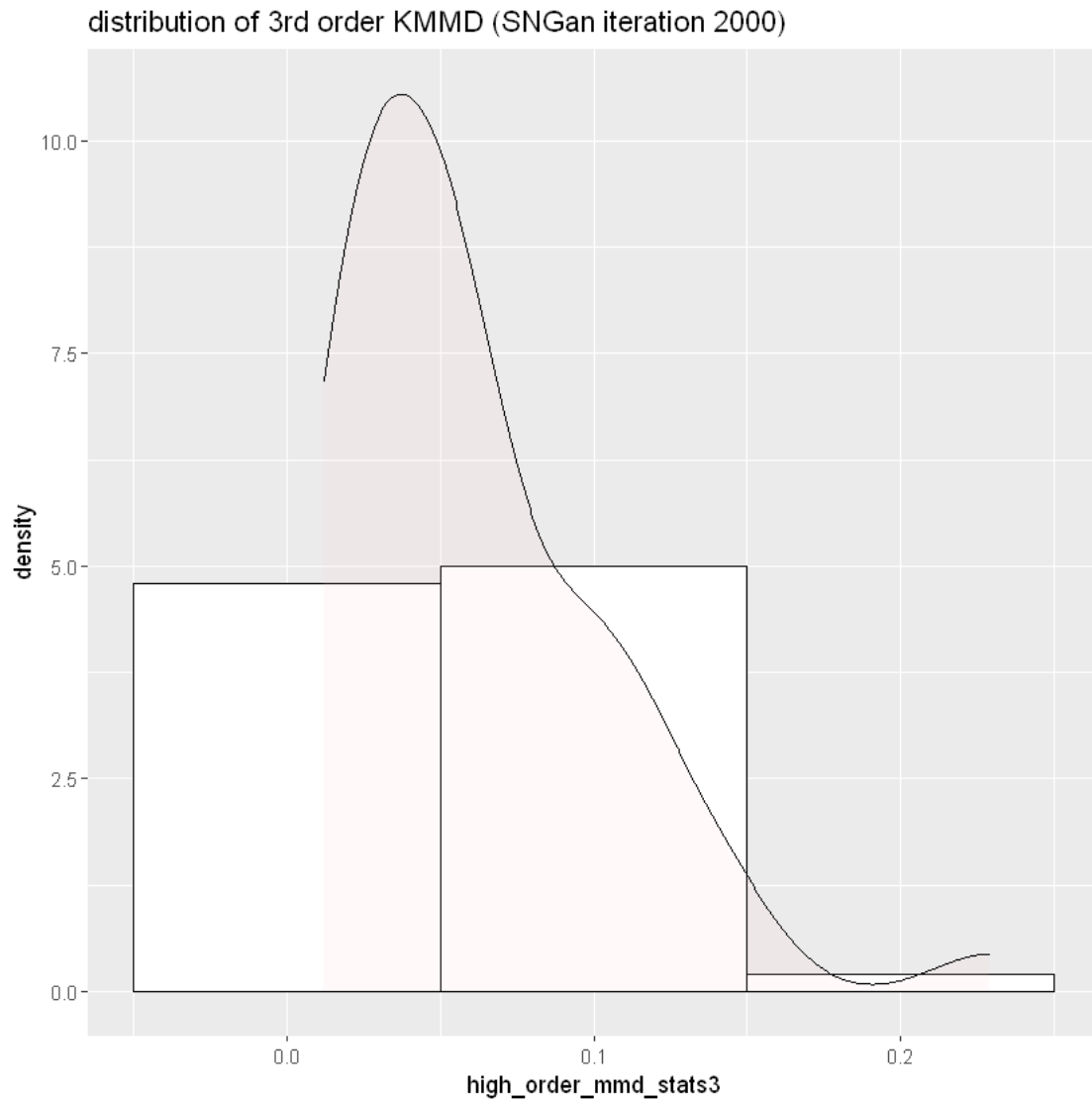


1.9.2 MMD statistics

```
[41]: ggplot(df_low_order_mmd_stats3, aes(x=low_order_mmd_stats3)) +
  labs(title="distribution of 1st order KMMD (SNGan iteration 2000)") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↪count on y-axis
               binwidth=.1,
               colour="black", fill="white") +
  geom_density(alpha=.1, fill="steelblue")    # Overlay with transparent
  ↪density plot
```



```
[42]: ggplot(df_high_order_mmd_stats3, aes(x=high_order_mmd_stats3)) +
  labs(title="distribution of 3rd order KMMD (SNGan iteration 2000)") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↪count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="pink") # Overlay with transparent density plot
```



```
[43]: summary(df_low_order_mmd_stats3)
```

```
low_order_mmd_stats3  
Min.   :0.1128  
1st Qu.:0.1638  
Median :0.2326  
Mean   :0.2372  
3rd Qu.:0.2961  
Max.   :0.4818
```

```
[44]: summary(df_high_order_mmd_stats3)
```

```
high_order_mmd_stats3
```

```
Min.      :0.01215
1st Qu.   :0.02558
Median    :0.05234
Mean      :0.06125
3rd Qu.   :0.08578
Max.      :0.22912
```

1.10 Result of Flowers dataset for SNGan (iteration 5000)

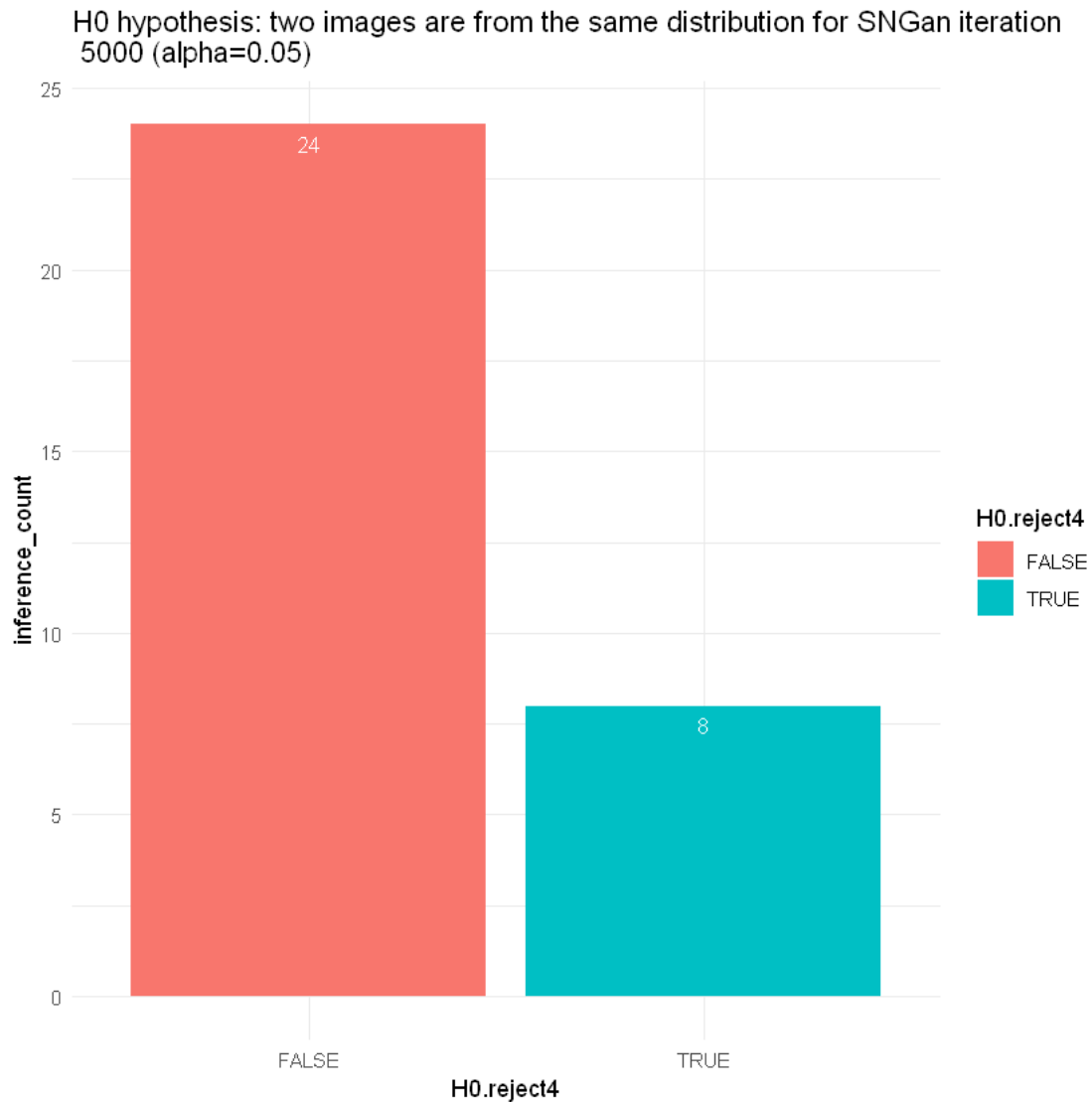
```
[45]: result4 = compute_kmmd("./sngan_flower_results/input", "./sngan_flower_results/
↳ iter_5000")
```

[illegible]

```
[46]: H0.reject4 = result4$H0.reject
df_H0.reject4 = as.data.frame(H0.reject4) %>% group_by(H0.reject4) %>% count()
↳ %>% rename("inference_count"=n)
low_order_mmd_stats4 = result4$low_order_mmd_stats
df_low_order_mmd_stats4 = as.data.frame(low_order_mmd_stats4)
high_order_mmd_stats4 = result4$high_order_mmd_stats
df_high_order_mmd_stats4 = as.data.frame(high_order_mmd_stats4)
```

1.10.1 Hypothesis test result

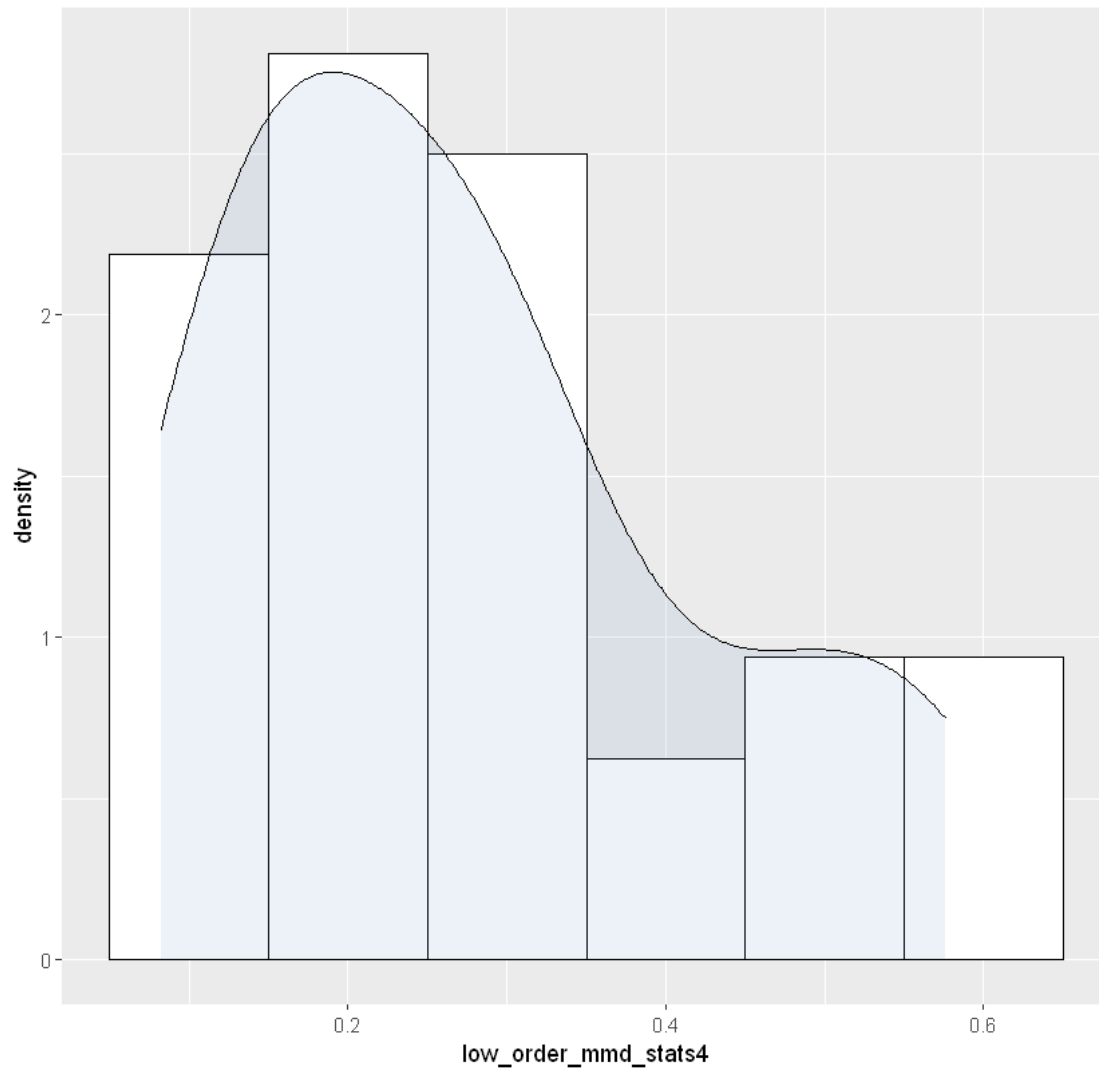
```
[47]: ggplot(data=df_H0.reject4, aes(x=H0.reject4, y=inference_count, fill=H0.
↳ reject4)) +
  labs(title="H0 hypothesis: two images are from the same distribution for
↳ SNGan iteration\n 5000 (alpha=0.05)") +
  geom_bar(stat="identity")+
  geom_text(aes(label=inference_count), vjust=1.6, color="white", size=3.5)+
  theme_minimal()
```



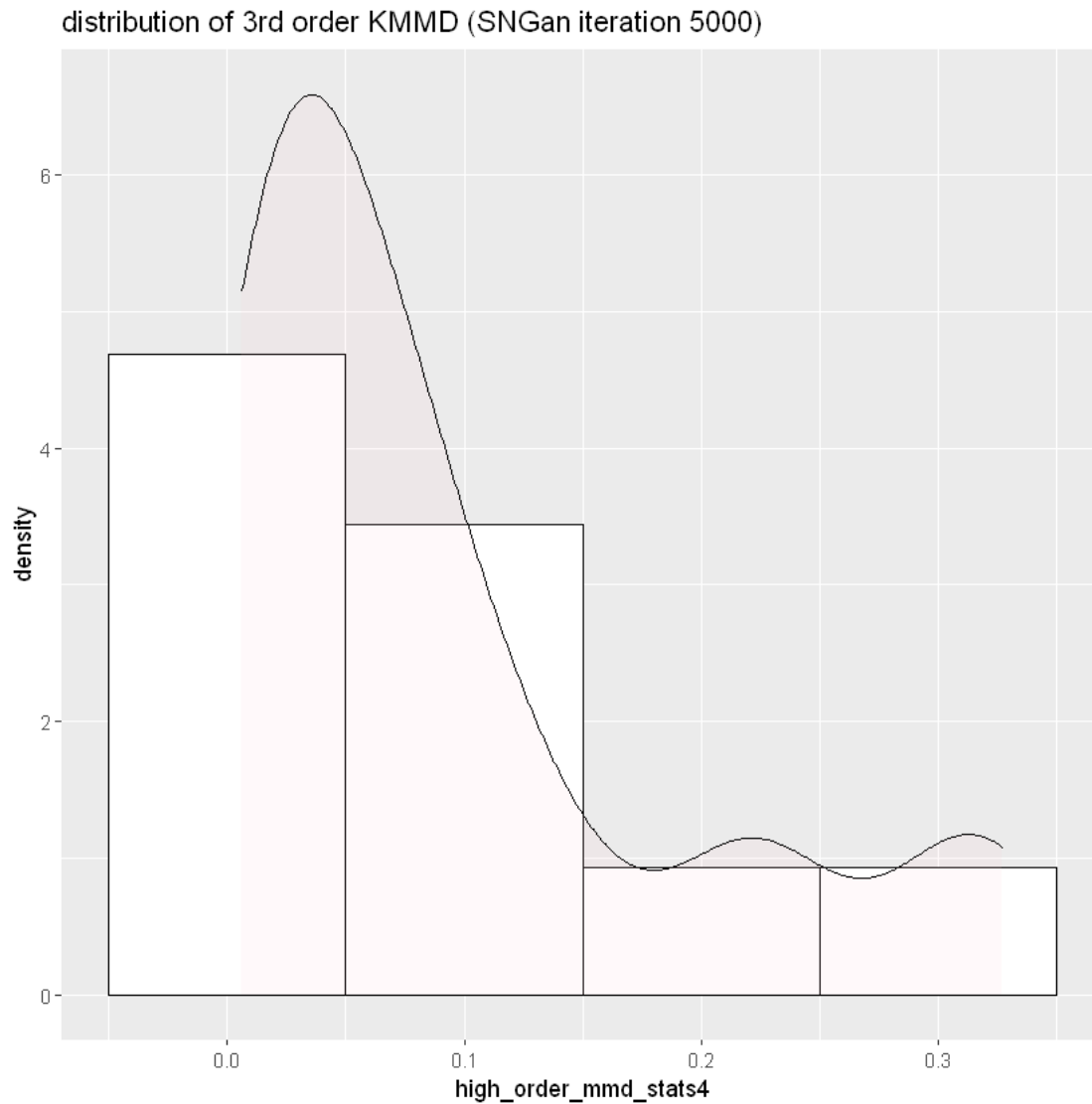
1.10.2 MMD statistics

```
[48]: ggplot(df_low_order_mmd_stats4, aes(x=low_order_mmd_stats4)) +
  labs(title="distribution of 1st order KMMD (SNGan iteration 5000)") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↪count on y-axis
               binwidth=.1,
               colour="black", fill="white") +
  geom_density(alpha=.1, fill="steelblue")    # Overlay with transparent
  ↪density plot
```


distribution of 1st order KMMD (SNGan iteration 5000)



```
[49]: ggplot(df_high_order_mmd_stats4, aes(x=high_order_mmd_stats4)) +
  labs(title="distribution of 3rd order KMMD (SNGan iteration 5000)") +
  geom_histogram(aes(y=..density..),      # Histogram with density instead of
  ↪count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="pink") # Overlay with transparent density plot
```



```
[50]: summary(df_low_order_mmd_stats4)
```

```
low_order_mmd_stats4
Min.   :0.08221
1st Qu.:0.15313
Median :0.24500
Mean   :0.27322
3rd Qu.:0.34510
Max.   :0.57584
```

```
[51]: summary(df_high_order_mmd_stats4)
```

```
high_order_mmd_stats4
```

```
Min.      :0.006196
1st Qu.   :0.022311
Median    :0.058128
Mean      :0.093146
3rd Qu.   :0.116364
Max.      :0.327304
```

1.11 Result of Face dataset for SNGan (iteration 5000)

```
[52]: result4 = compute_kmmd("./sngan_face_results/input", "./sngan_face_results/
↳ iter_5000")
```

[illegible]

```

Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel

```

```

[53]: H0.reject4 = result4$H0.reject
df_H0.reject4 = as.data.frame(H0.reject4) %>% group_by(H0.reject4) %>% count()
  ↪ %>% rename("inference_count"=n)
low_order_mmd_stats4 = result4$low_order_mmd_stats
df_low_order_mmd_stats4 = as.data.frame(low_order_mmd_stats4)
high_order_mmd_stats4 = result4$high_order_mmd_stats
df_high_order_mmd_stats4 = as.data.frame(high_order_mmd_stats4)

```

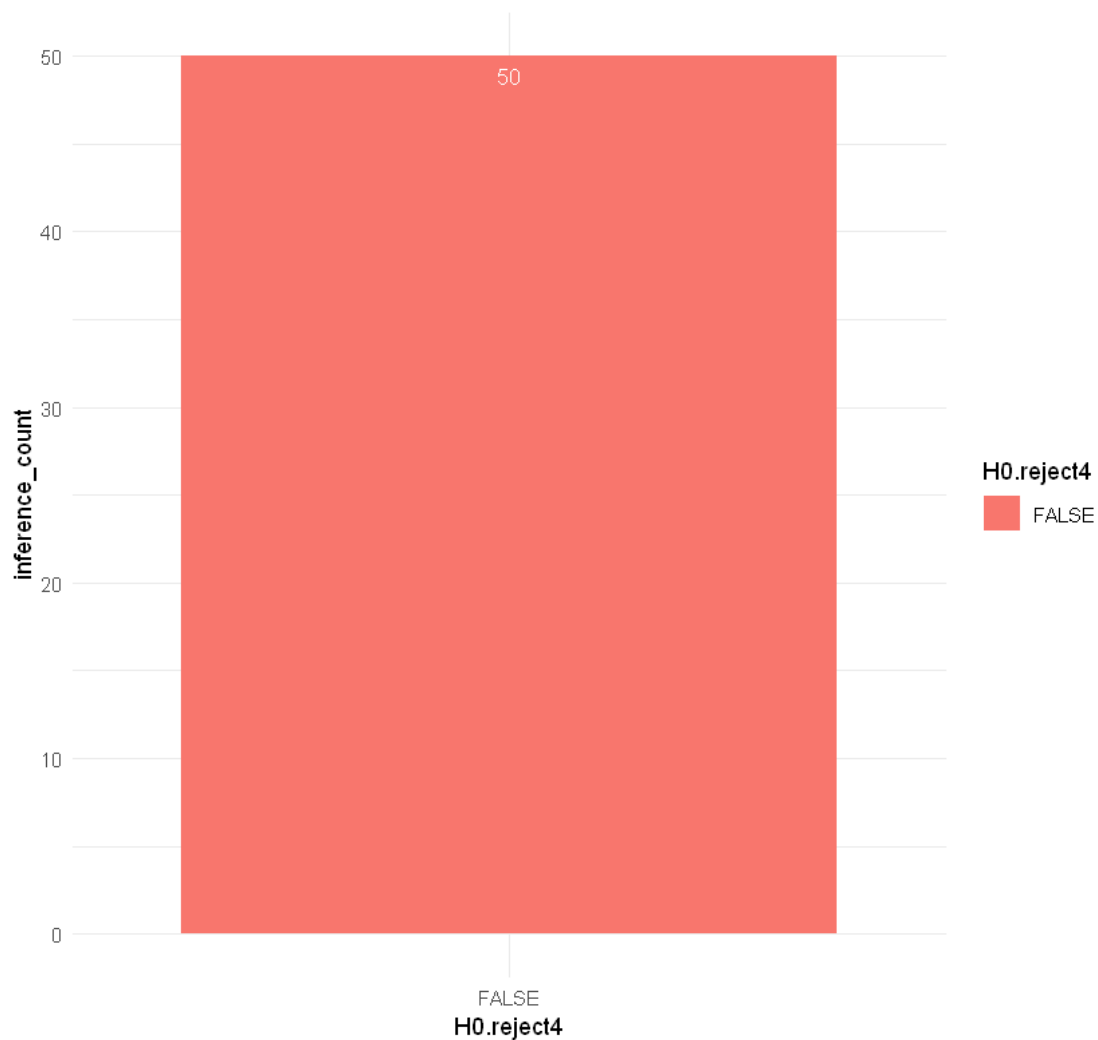
1.11.1 Hypothesis test result

```

[54]: ggplot(data=df_H0.reject4, aes(x=H0.reject4, y=inference_count, fill=H0.
  ↪ reject4)) +
  labs(title="H0 hypothesis: two images are from the same distribution for_
  ↪ SNGan iteration\n 5000 (alpha=0.05)") +
  geom_bar(stat="identity")+
  geom_text(aes(label=inference_count), vjust=1.6, color="white", size=3.5)+
  theme_minimal()

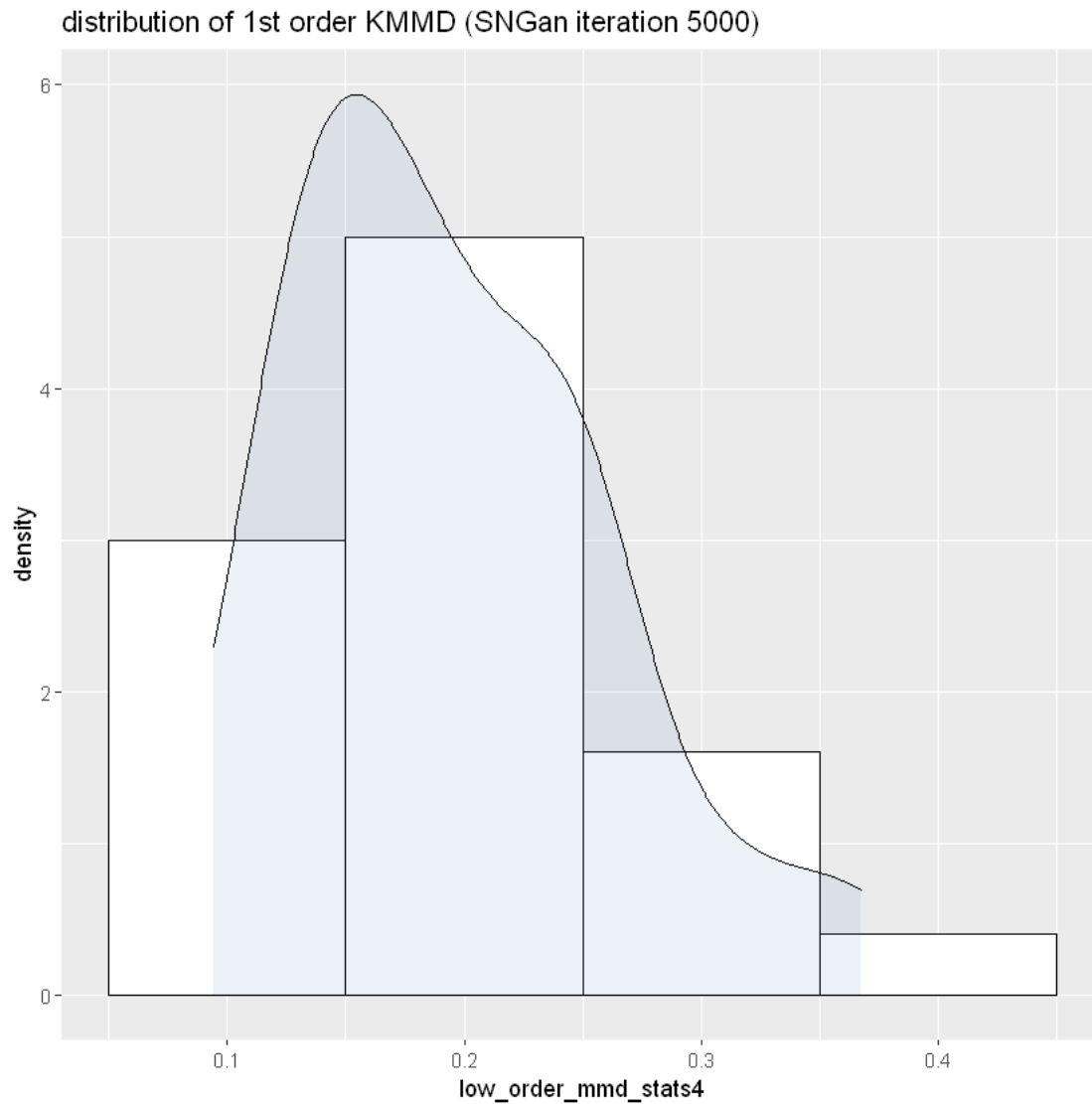
```

H0 hypothesis: two images are from the same distribution for SNGan iteration 5000 (alpha=0.05)

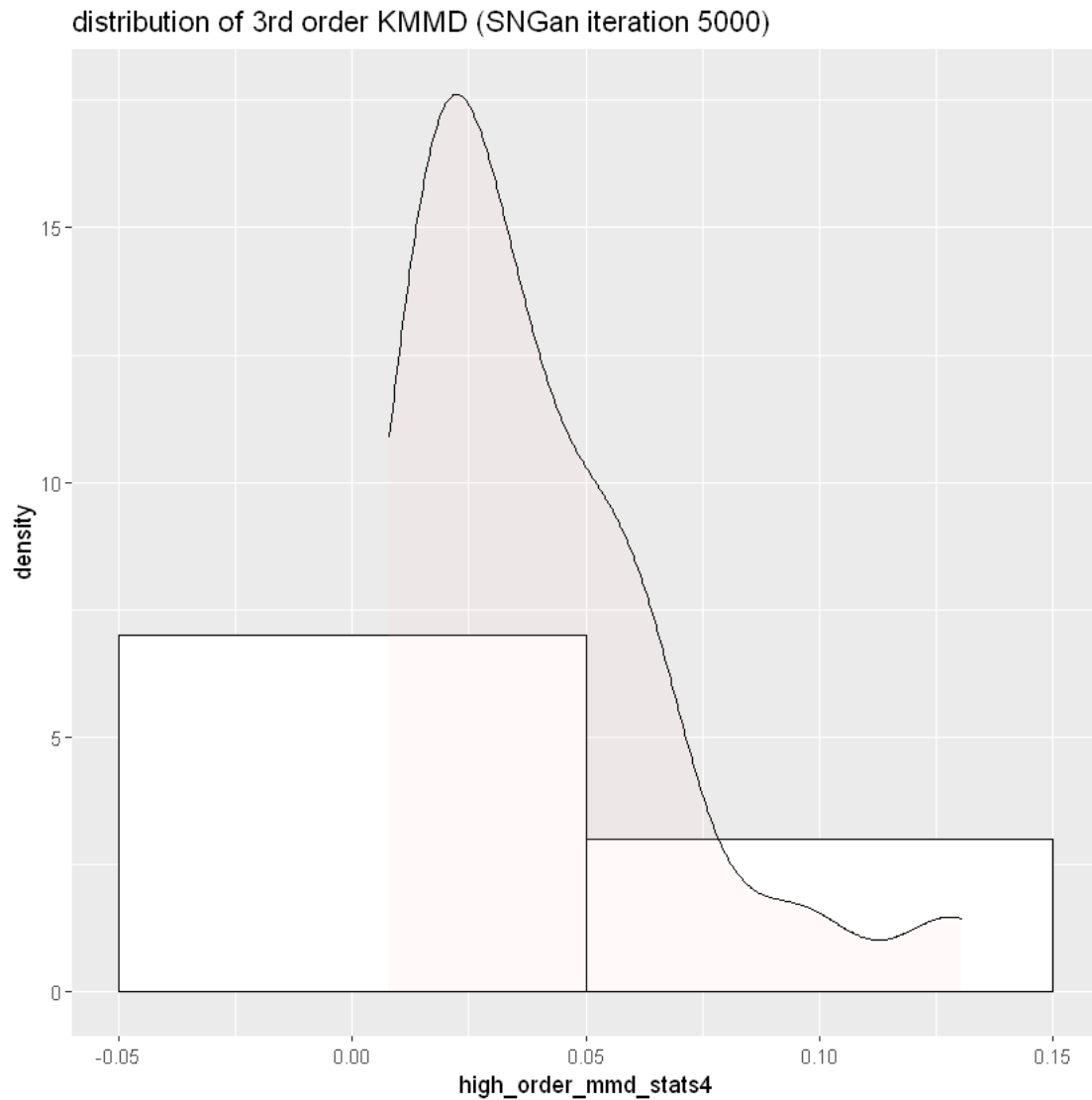


1.11.2 MMD statistics

```
[55]: ggplot(df_low_order_mmd_stats4, aes(x=low_order_mmd_stats4)) +
  labs(title="distribution of 1st order KMMMD (SNGan iteration 5000)") +
  geom_histogram(aes(y=..density..),      # Histogram with density instead of
  ↪count on y-axis
               binwidth=.1,
               colour="black", fill="white") +
  geom_density(alpha=.1, fill="steelblue") # Overlay with transparent
  ↪density plot
```



```
[56]: ggplot(df_high_order_mmd_stats4, aes(x=high_order_mmd_stats4)) +
  labs(title="distribution of 3rd order KMMD (SNGan iteration 5000)") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↪count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="pink") # Overlay with transparent density plot
```



```
[57]: summary(df_low_order_mmd_stats4)
```

```
low_order_mmd_stats4
Min.   :0.09433
1st Qu.:0.14044
Median :0.18524
Mean   :0.19491
3rd Qu.:0.23649
Max.   :0.36736
```

```
[58]: summary(df_high_order_mmd_stats4)
```

```
high_order_mmd_stats4
```

```
Min.      :0.007909
1st Qu.   :0.018253
Median    :0.032325
Mean      :0.039961
3rd Qu.   :0.053585
Max.      :0.130525
```

1.12 Result of anime dataset for SNGan (iteration 5000)

```
[59]: result4 = compute_kmmd("./sngan_anime_results/input", "./sngan_anime_results/
↳ iter_5000")
```

[illegible]


```

Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel

```

```

[60]: H0.reject4 = result4$H0.reject
df_H0.reject4 = as.data.frame(H0.reject4) %>% group_by(H0.reject4) %>% count()
  ↪ %>% rename("inference_count"=n)
low_order_mmd_stats4 = result4$low_order_mmd_stats
df_low_order_mmd_stats4 = as.data.frame(low_order_mmd_stats4)
high_order_mmd_stats4 = result4$high_order_mmd_stats
df_high_order_mmd_stats4 = as.data.frame(high_order_mmd_stats4)

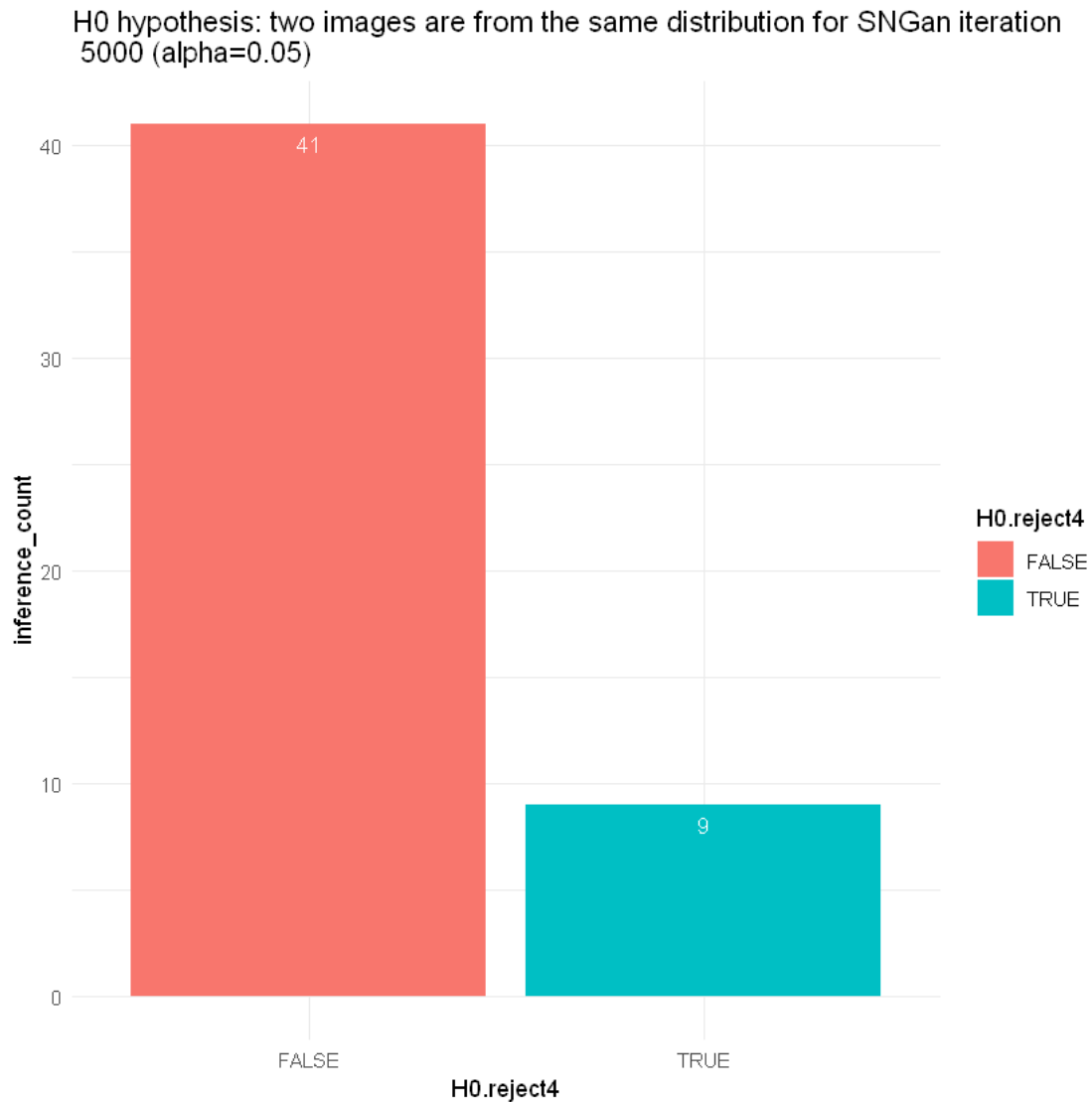
```

1.12.1 Hypothesis test result

```

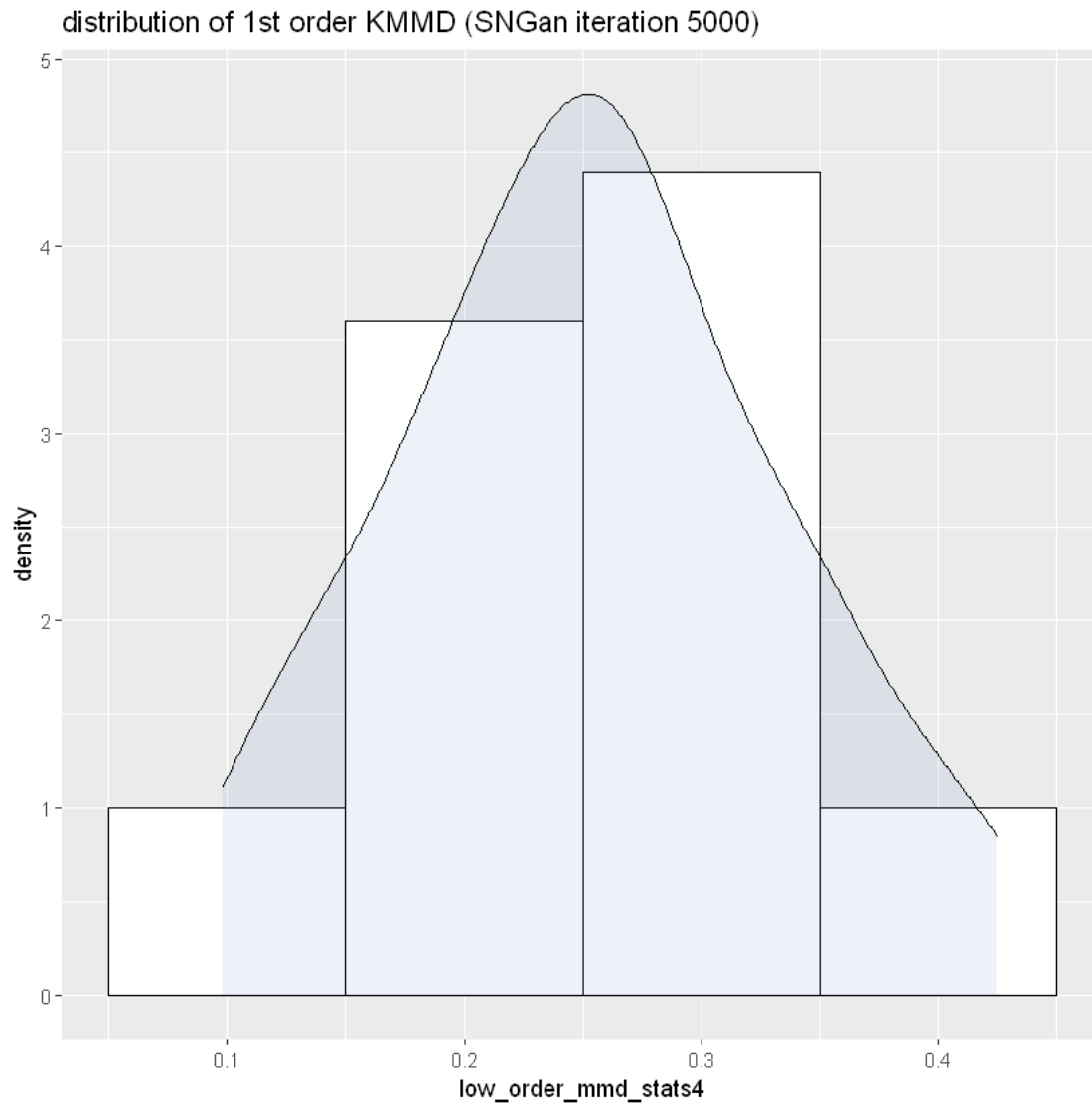
[61]: ggplot(data=df_H0.reject4, aes(x=H0.reject4, y=inference_count, fill=H0.
  ↪ reject4)) +
  labs(title="H0 hypothesis: two images are from the same distribution for_
  ↪ SNGan iteration\n 5000 (alpha=0.05)") +
  geom_bar(stat="identity")+
  geom_text(aes(label=inference_count), vjust=1.6, color="white", size=3.5)+
  theme_minimal()

```

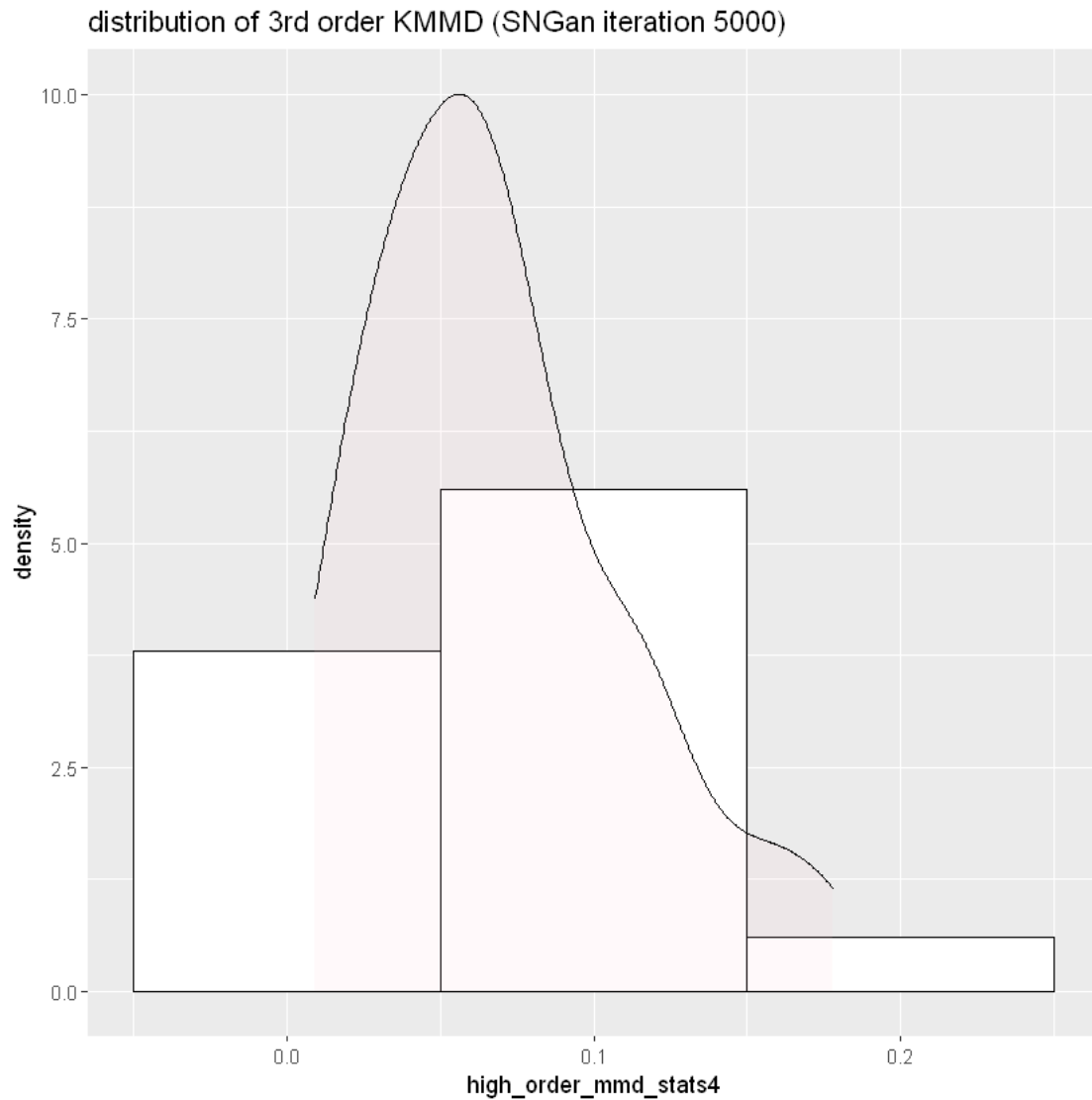


```
[62]: ### MMD statistics
```

```
[63]: ggplot(df_low_order_mmd_stats4, aes(x=low_order_mmd_stats4)) +
  labs(title="distribution of 1st order KMMD (SNGan iteration 5000)") +
  geom_histogram(aes(y=..density..),      # Histogram with density instead of count
    ↪ count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="steelblue") # Overlay with transparent density plot
    ↪ density plot
```



```
[64]: ggplot(df_high_order_mmd_stats4, aes(x=high_order_mmd_stats4)) +
  labs(title="distribution of 3rd order KMMD (SNGan iteration 5000)") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↪count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="pink") # Overlay with transparent density plot
```



```
[65]: summary(df_low_order_mmd_stats4)
```

```
low_order_mmd_stats4
Min.   :0.09808
1st Qu.:0.19764
Median :0.25452
Mean   :0.25366
3rd Qu.:0.30486
Max.   :0.42495
```

```
[66]: summary(df_high_order_mmd_stats4)
```

```
high_order_mmd_stats4
```

```
Min.      :0.009135
1st Qu.   :0.037675
Median    :0.062997
Mean      :0.068799
3rd Qu.   :0.090967
Max.      :0.177984
```

1.13 Result of datasets for BigGan (iteration 500)

1.13.1 flowers

```
[47]: result_flowerns = compute_kmmd("./biggan_flowern_results/input", "./
      ↪biggan_flowern_results/iter_500")
```

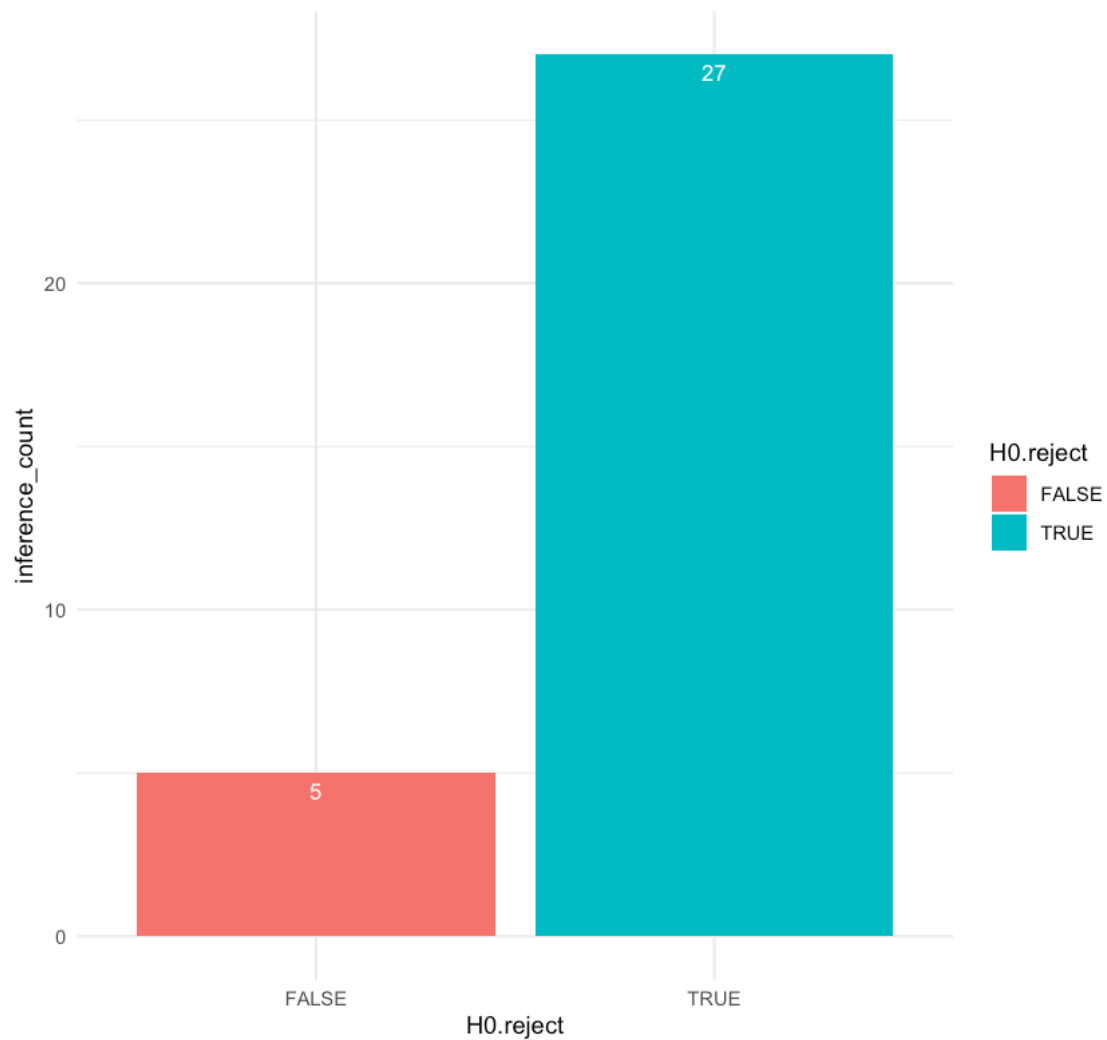
[illegible]

```
[48]: H0.reject = result_flowers$H0.reject
df_H0.reject = as.data.frame(H0.reject) %>% group_by(H0.reject) %>% count() %>%
  ↪rename("inference_count"=n)
low_order_mmd_stats = result_flowers$low_order_mmd_stats
df_low_order_mmd_stats = as.data.frame(low_order_mmd_stats)
high_order_mmd_stats = result_flowers$high_order_mmd_stats
df_high_order_mmd_stats = as.data.frame(high_order_mmd_stats)
```

Hypothesis test result

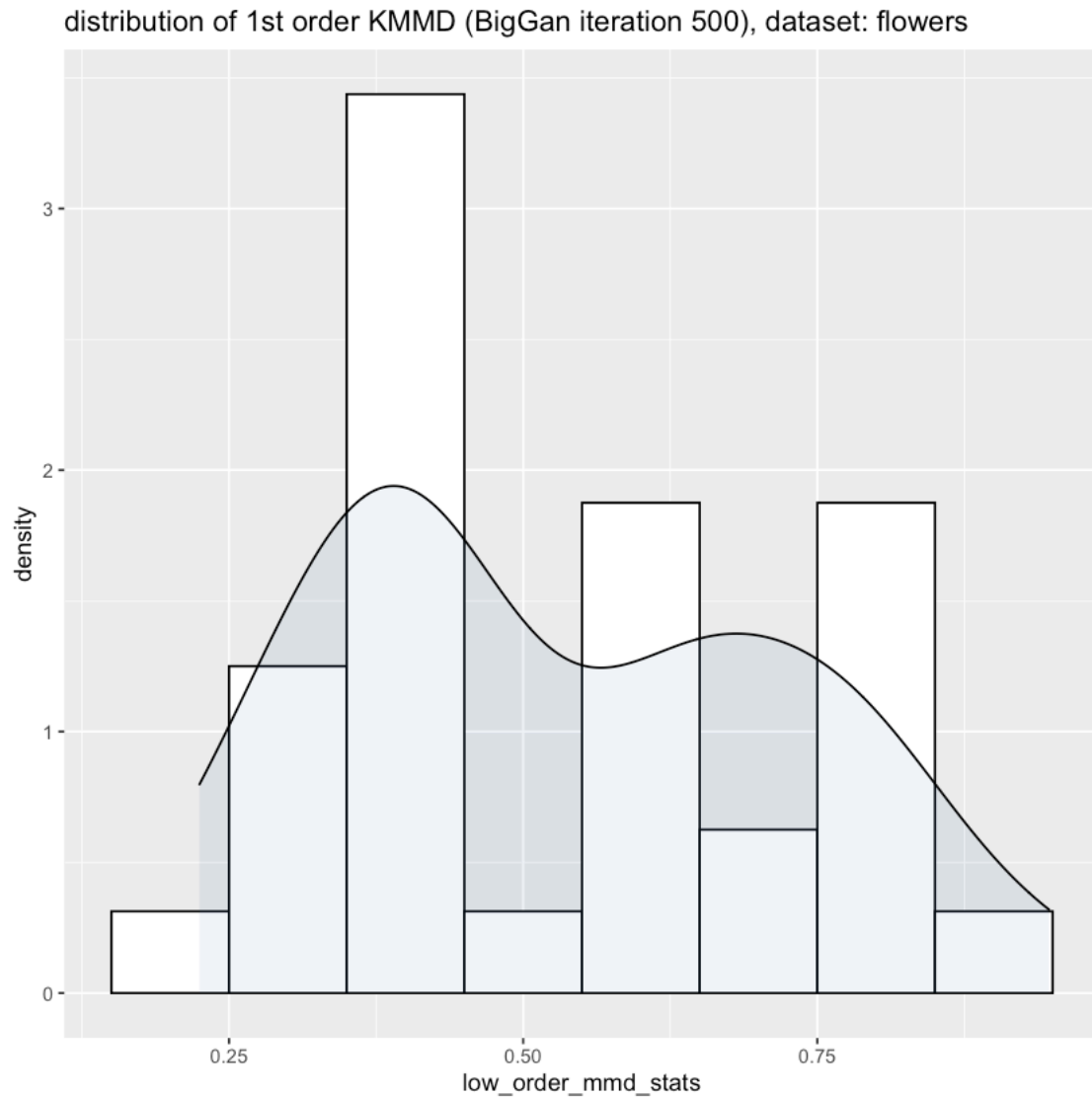
```
[49]: ggplot(data=df_H0.reject, aes(x=H0.reject, y=inference_count, fill=H0.reject)) +
  labs(title="H0 hypothesis: two images are from the same distribution for
  ↪BigGan iteration\n 500 (alpha=0.05), dataset: flowers") +
  geom_bar(stat="identity")+
  geom_text(aes(label=inference_count), vjust=1.6, color="white", size=3.5)+
  theme_minimal()
```

H0 hypothesis: two images are from the same distribution for BigGan iteration 500 (alpha=0.05), dataset: flowers

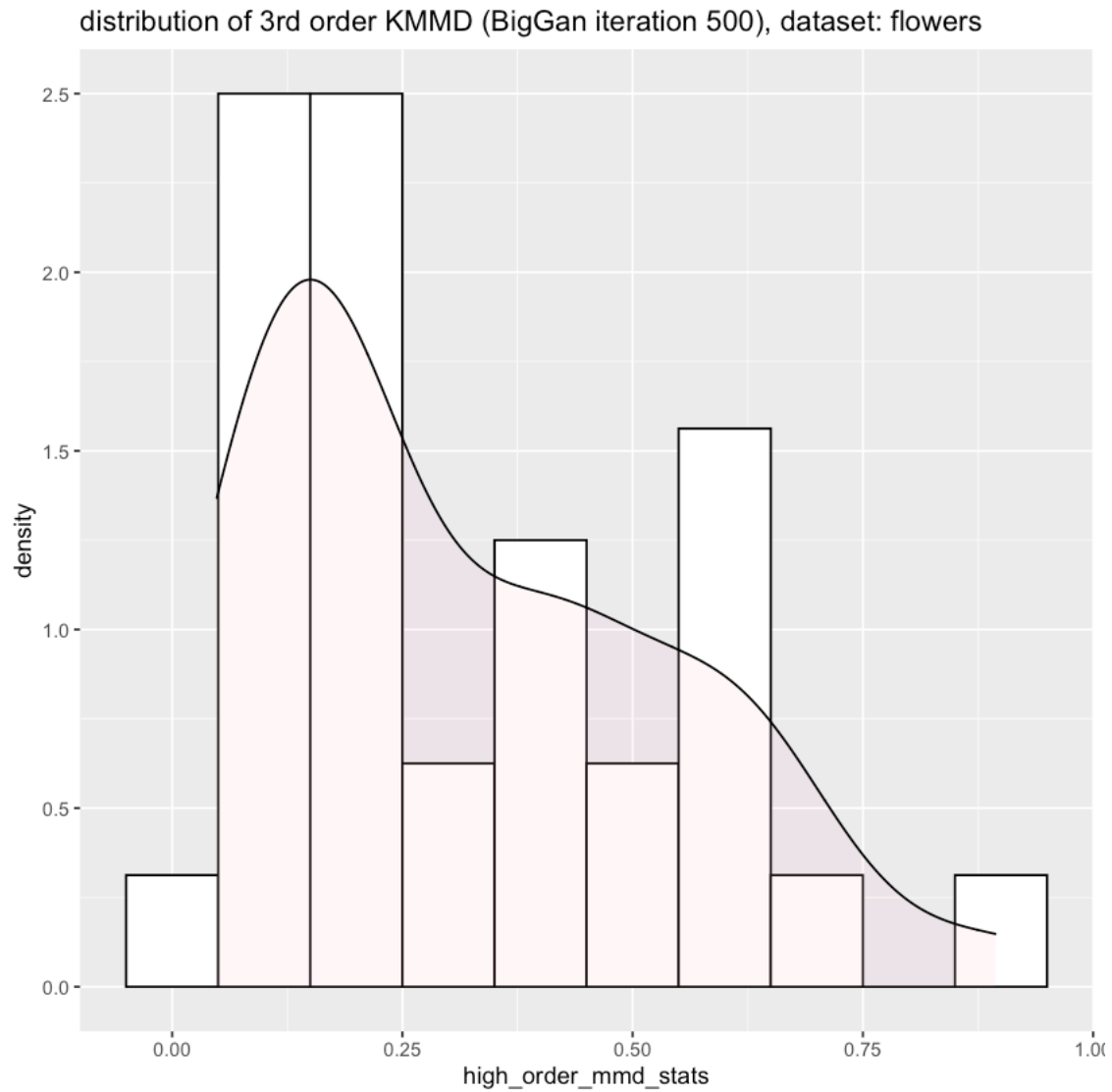


MMD statistics

```
[50]: ggplot(df_low_order_mmd_stats, aes(x=low_order_mmd_stats)) +
  labs(title="distribution of 1st order KMMD (BigGan iteration 500), dataset:
  ↳flowers") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↳count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="steelblue") # Overlay with transparent
  ↳density plot
```



```
[51]: ggplot(df_high_order_mmd_stats, aes(x=high_order_mmd_stats)) +
  labs(title="distribution of 3rd order KMMD (BigGan iteration 500), dataset:
  →flowers") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  →count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="pink") # Overlay with transparent density plot
```

```
[52]: summary(df_low_order_mmd_stats)
```

```
low_order_mmd_stats
Min.   :0.2248
1st Qu.:0.3795
Median :0.4479
Mean   :0.5321
3rd Qu.:0.6763
Max.   :0.9469
```

```
[53]: summary(df_high_order_mmd_stats)
```

```
high_order_mmd_stats
```

```
Min.      :0.04844
1st Qu.   :0.14069
Median    :0.19715
Mean      :0.31712
3rd Qu.   :0.45411
Max.      :0.89385
```

1.13.2 face

```
[54]: result_face = compute_kmmd("./biggan_face_results/input", "biggan_face_results/iter_500")
```

[illegible]

```

Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel

```

```

[55]: H0.reject = result_face$H0.reject
df_H0.reject = as.data.frame(H0.reject) %>% group_by(H0.reject) %>% count() %>%
  ↪rename("inference_count"=n)
low_order_mmd_stats = result_face$low_order_mmd_stats
df_low_order_mmd_stats = as.data.frame(low_order_mmd_stats)
high_order_mmd_stats = result_face$high_order_mmd_stats
df_high_order_mmd_stats = as.data.frame(high_order_mmd_stats)

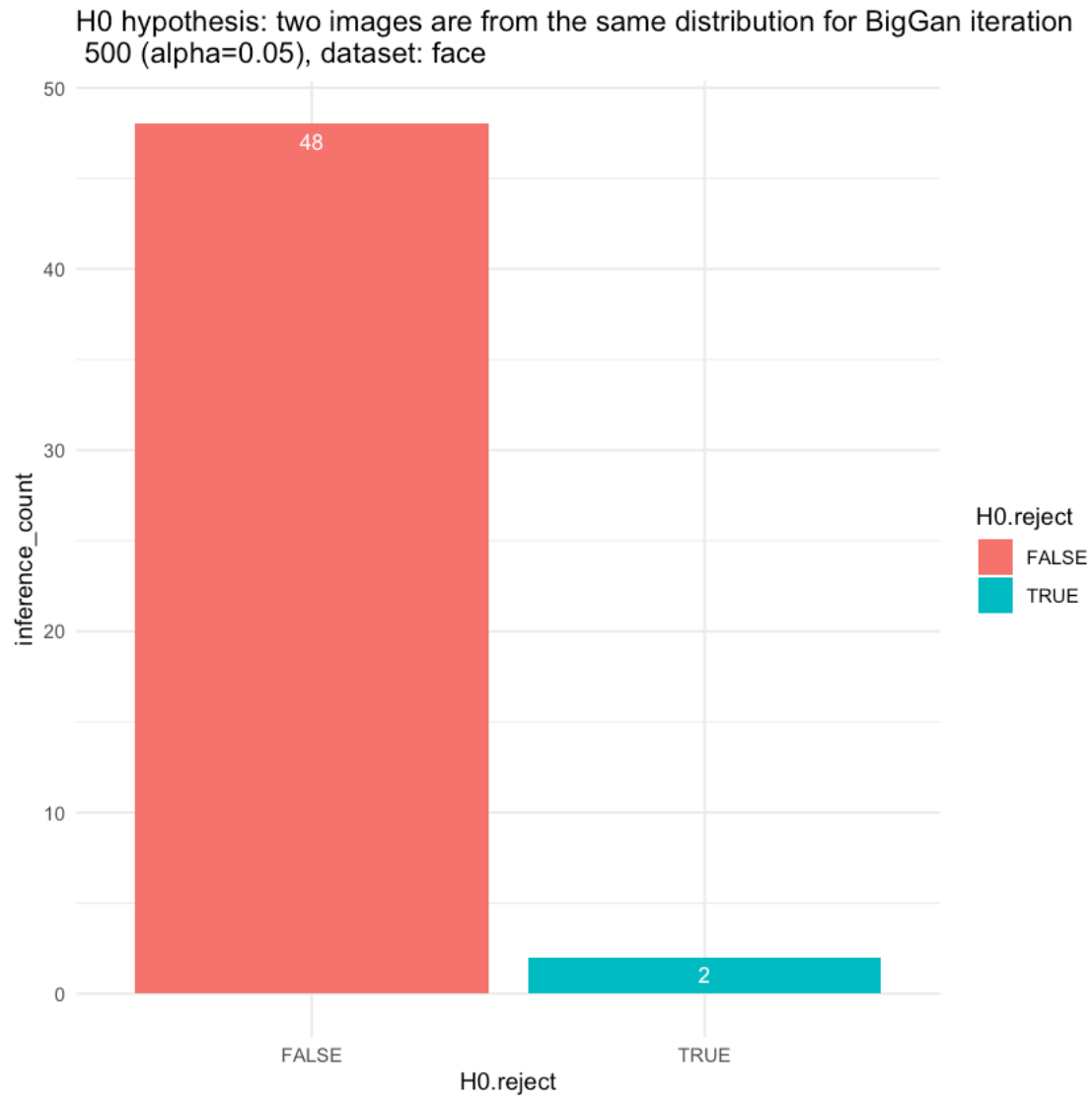
```

Hypothesis test result

```

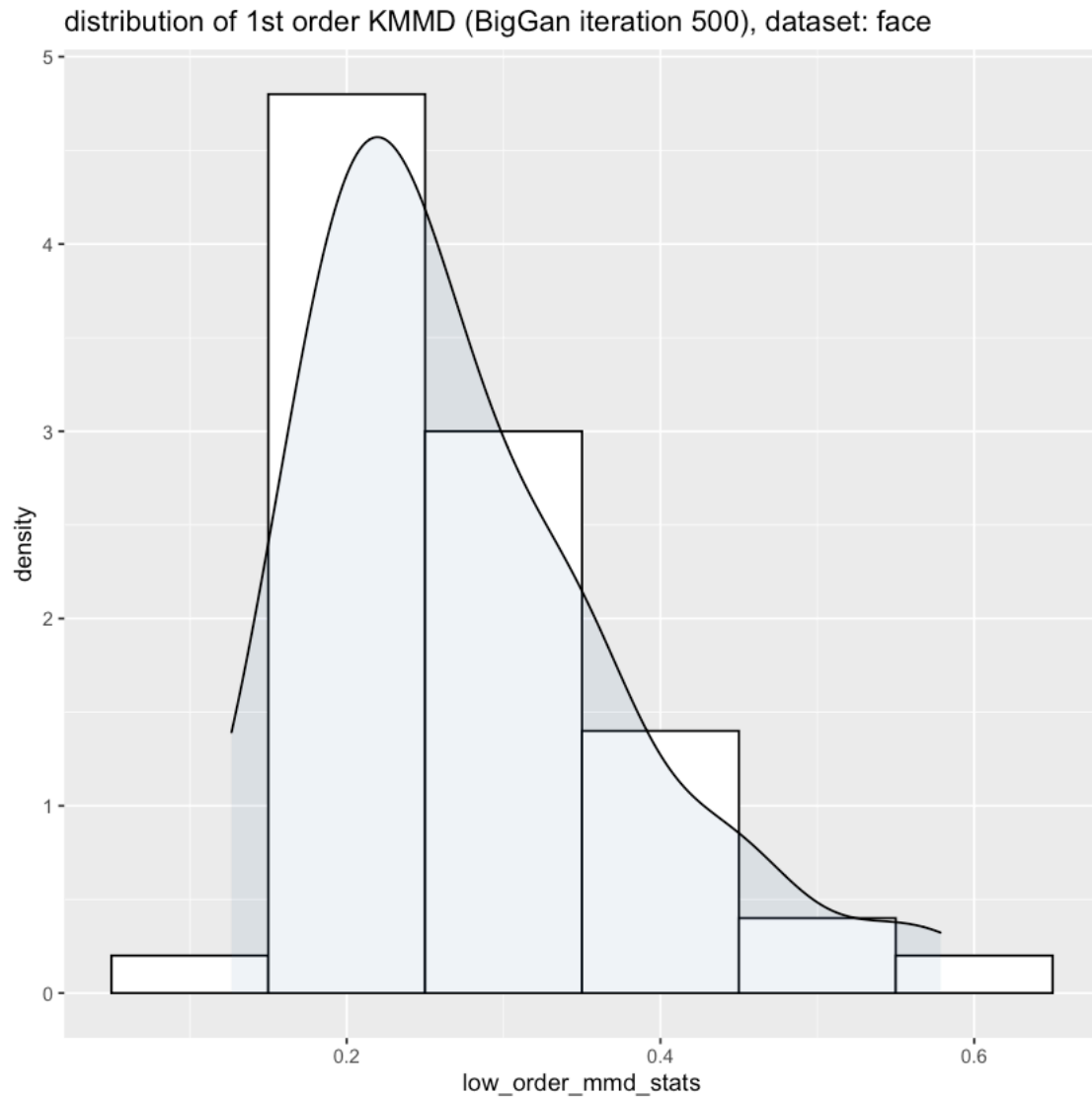
[56]: ggplot(data=df_H0.reject, aes(x=H0.reject, y=inference_count, fill=H0.reject)) +
  labs(title="H0 hypothesis: two images are from the same distribution for_
  ↪BigGan iteration\n 500 (alpha=0.05), dataset: face") +
  geom_bar(stat="identity")+
  geom_text(aes(label=inference_count), vjust=1.6, color="white", size=3.5)+
  theme_minimal()

```

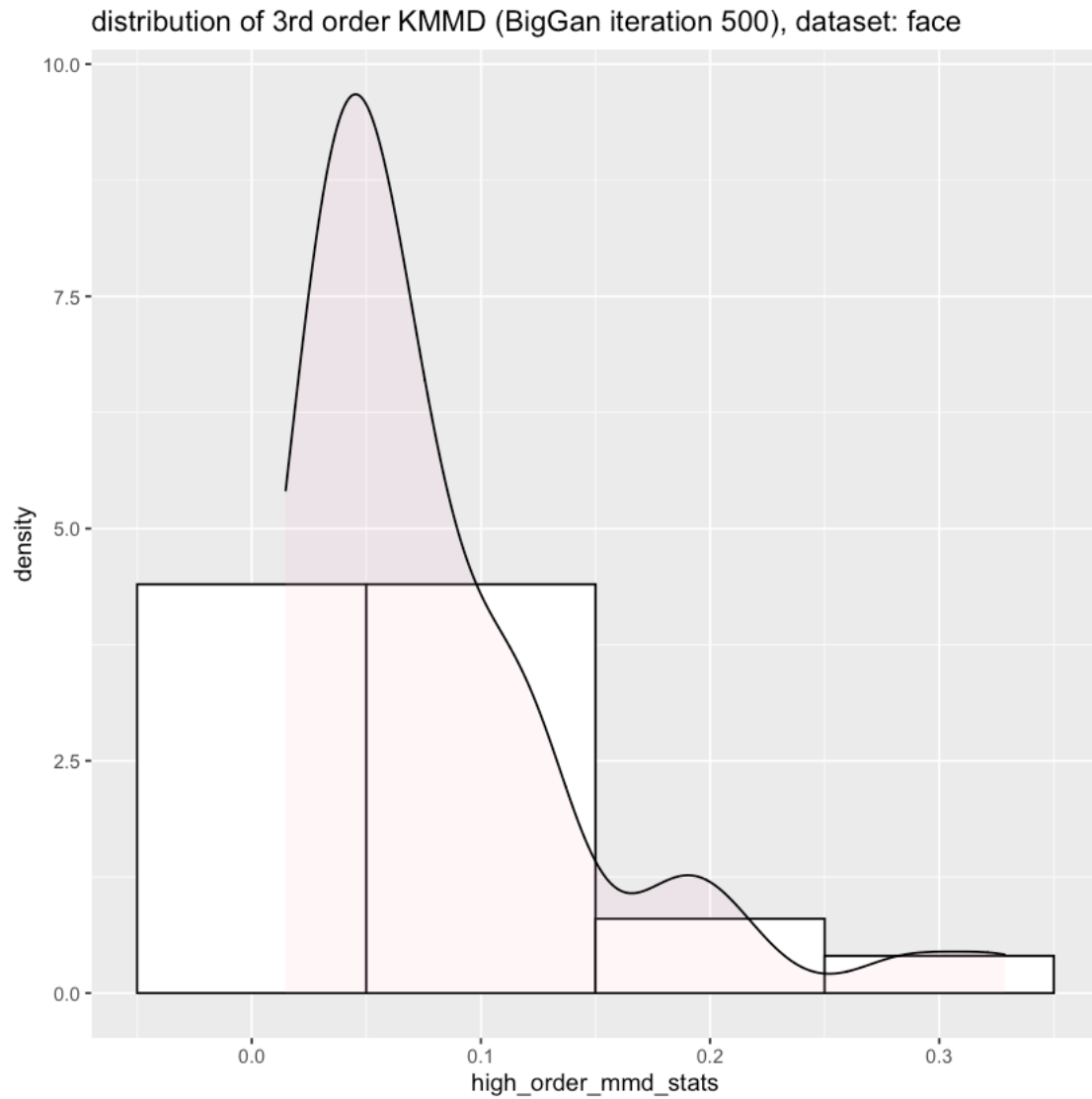


MMD statistics

```
[57]: ggplot(df_low_order_mmd_stats, aes(x=low_order_mmd_stats)) +
  labs(title="distribution of 1st order KMMD (BigGan iteration 500), dataset:
  ↳face") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↳count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="steelblue") # Overlay with transparent
  ↳density plot
```



```
[58]: ggplot(df_high_order_mmd_stats, aes(x=high_order_mmd_stats)) +
  labs(title="distribution of 3rd order KMMD (BigGan iteration 500), dataset:
  ↳face") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↳count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="pink") # Overlay with transparent density plot
```



```
[59]: summary(df_low_order_mmd_stats)
```

```
low_order_mmd_stats
Min.   :0.1266
1st Qu.:0.2016
Median :0.2501
Mean   :0.2749
3rd Qu.:0.3377
Max.   :0.5785
```

```
[60]: summary(df_high_order_mmd_stats)
```

```
high_order_mmd_stats
```

```
Min.      :0.01481
1st Qu.   :0.03831
Median    :0.05865
Mean      :0.08169
3rd Qu.   :0.11010
Max.      :0.32857
```

1.13.3 anime

```
[63]: result_anime = compute_kmmd("./biggan_anime_results/input", "./
      ↪biggan_anime_results/iter_500")
```

[illegible]

```

Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel

```

```

[64]: H0.reject = result_anime$H0.reject
df_H0.reject = as.data.frame(H0.reject) %>% group_by(H0.reject) %>% count() %>%
  ↪rename("inference_count"=n)
low_order_mmd_stats = result_anime$low_order_mmd_stats
df_low_order_mmd_stats = as.data.frame(low_order_mmd_stats)
high_order_mmd_stats = result_anime$high_order_mmd_stats
df_high_order_mmd_stats = as.data.frame(high_order_mmd_stats)

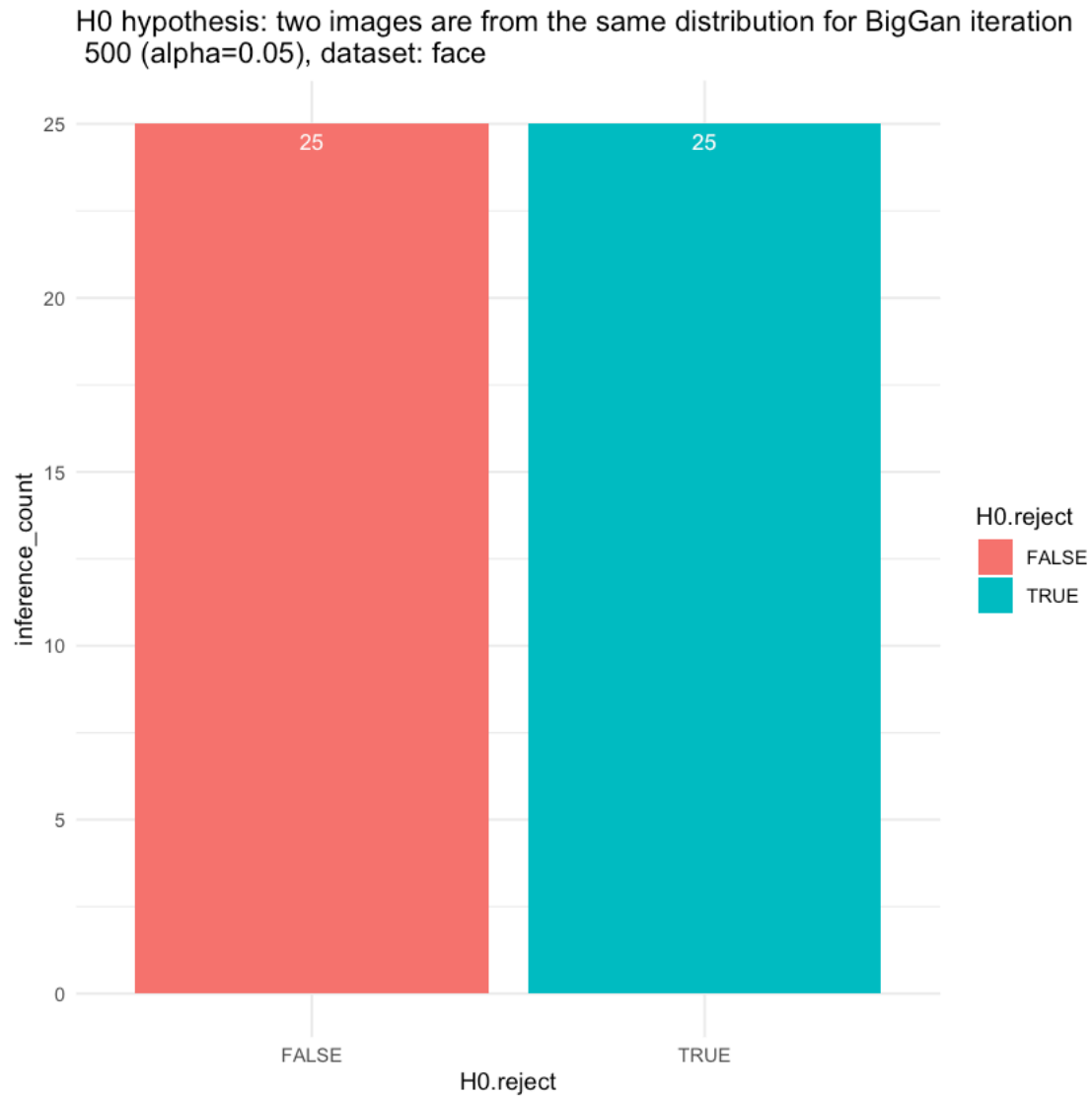
```

Hypothesis test result

```

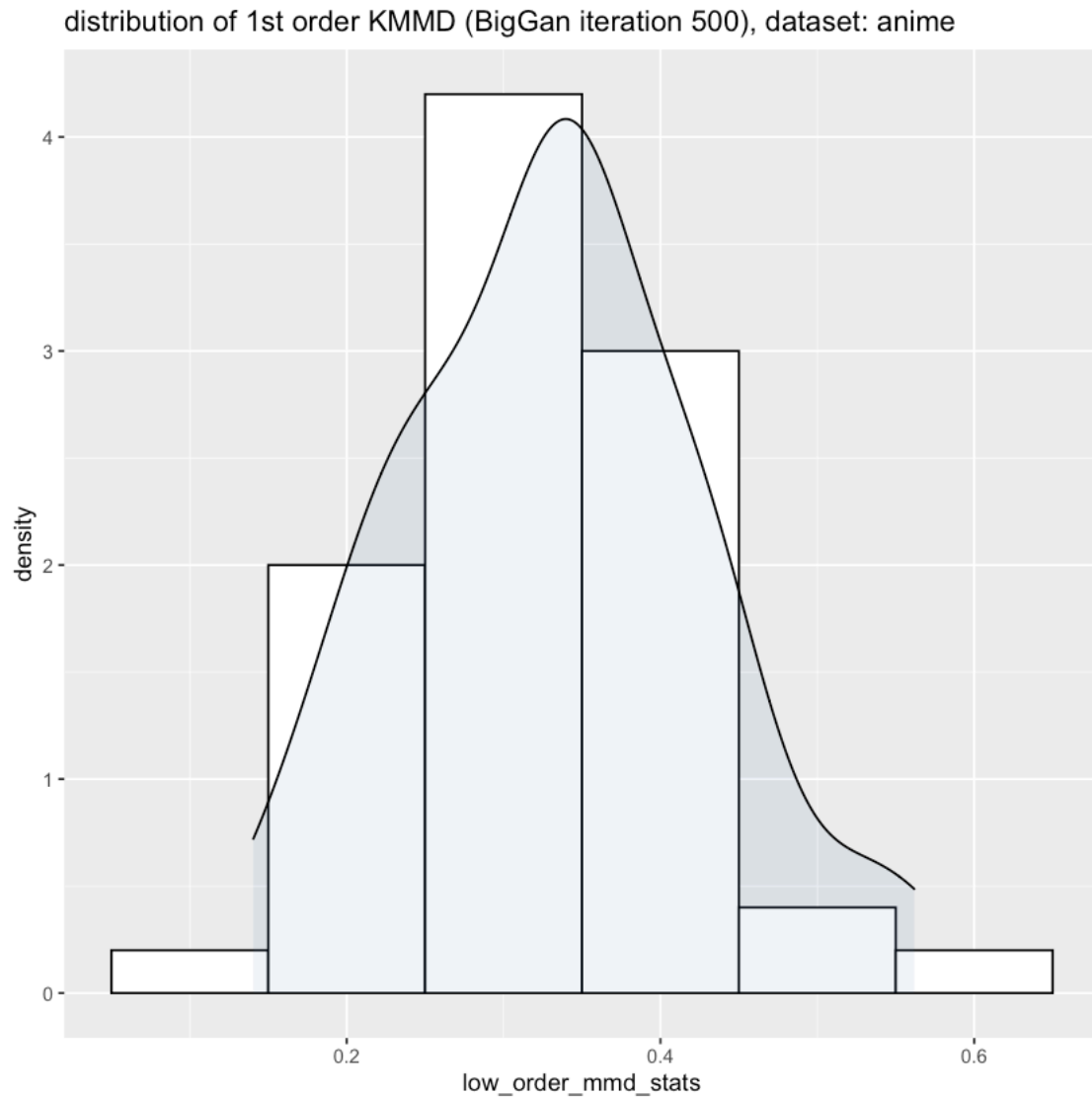
[65]: ggplot(data=df_H0.reject, aes(x=H0.reject, y=inference_count, fill=H0.reject)) +
  labs(title="H0 hypothesis: two images are from the same distribution for_
  ↪BigGan iteration\n 500 (alpha=0.05), dataset: face") +
  geom_bar(stat="identity")+
  geom_text(aes(label=inference_count), vjust=1.6, color="white", size=3.5)+
  theme_minimal()

```

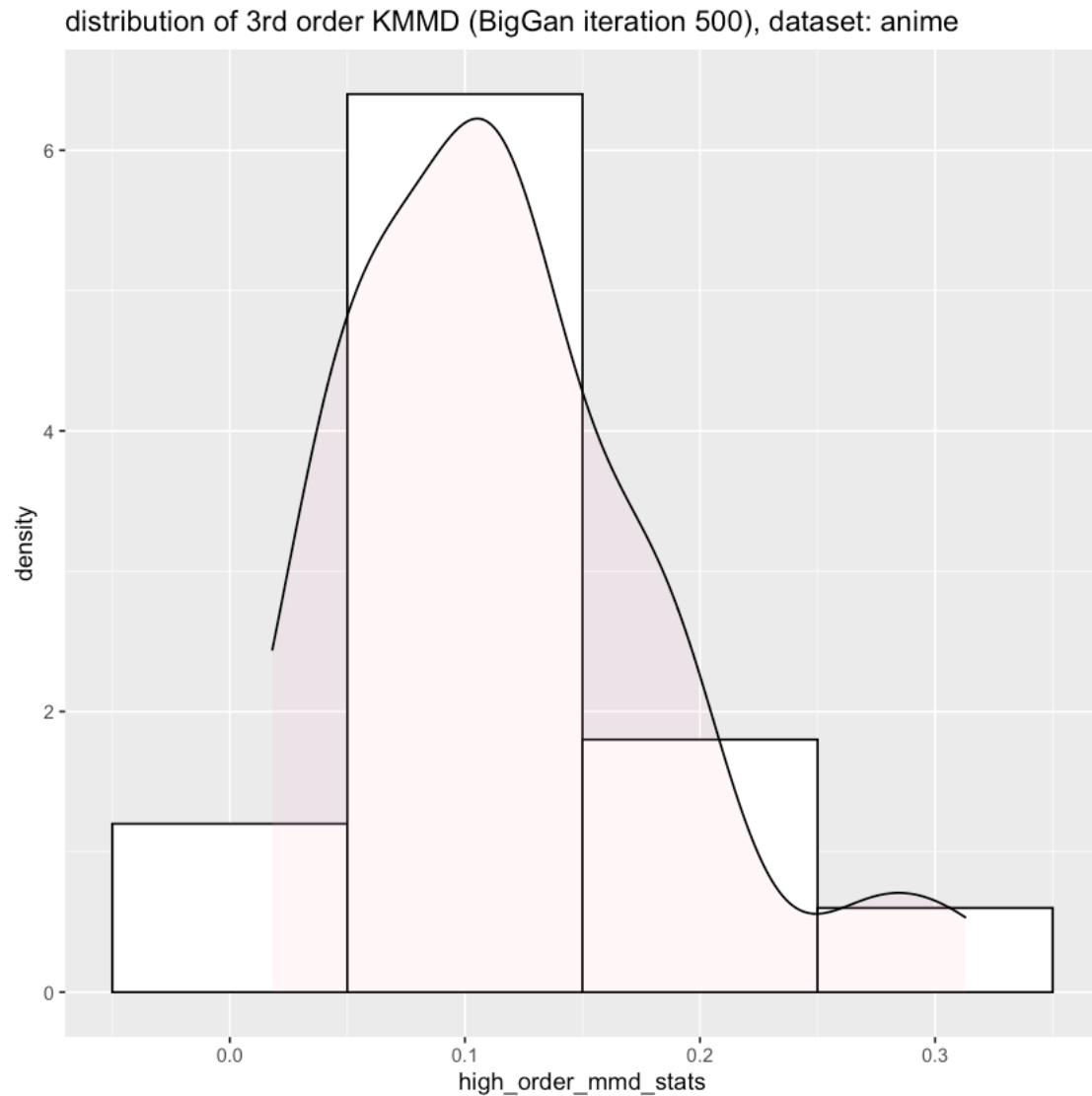



MMD statistics

```
[66]: ggplot(df_low_order_mmd_stats, aes(x=low_order_mmd_stats)) +
  labs(title="distribution of 1st order KMMD (BigGan iteration 500), dataset:␣
  ↪anime") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of␣
  ↪count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="steelblue") # Overlay with transparent␣
  ↪density plot
```



```
[67]: ggplot(df_high_order_mmd_stats, aes(x=high_order_mmd_stats)) +
  labs(title="distribution of 3rd order KMMD (BigGan iteration 500), dataset: anime") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="pink") # Overlay with transparent density plot
```



```
[68]: summary(df_low_order_mmd_stats)
```

```
low_order_mmd_stats
Min.   :0.1405
1st Qu.:0.2549
Median :0.3414
Mean   :0.3319
3rd Qu.:0.3872
Max.   :0.5618
```

```
[69]: summary(df_high_order_mmd_stats)
```

```
high_order_mmd_stats
```

```
Min.      :0.01810
1st Qu.   :0.06274
Median    :0.11402
Mean      :0.11651
3rd Qu.   :0.14738
Max.      :0.31285
```

1.14 Result of datasets for BigGan (iteration 2000)

1.14.1 flowers

```
[70]: result_flowers = compute_kmmnd("./biggan_flower_results/input", "./
      ↪biggan_flower_results/iter_2000")
```

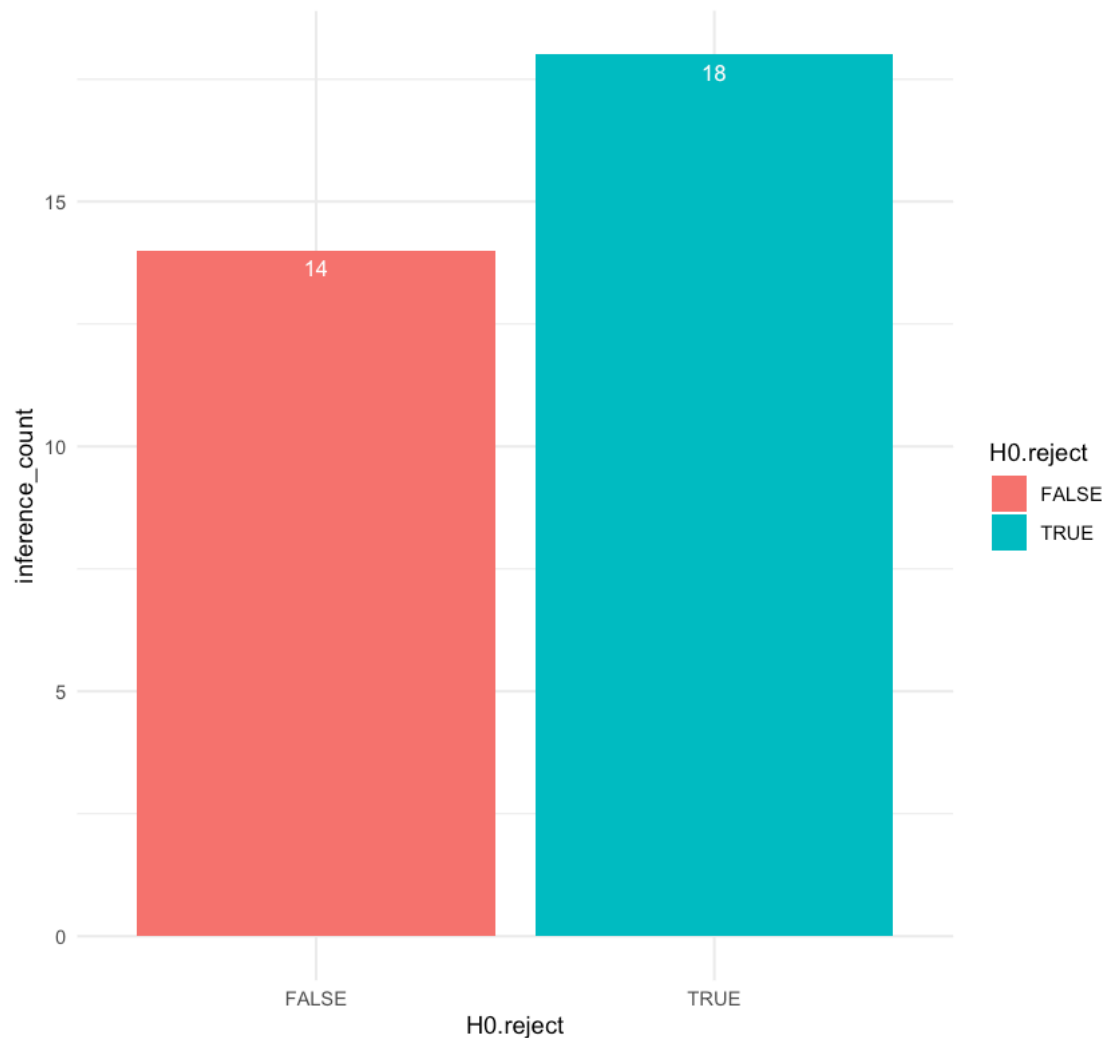
[illegible]

```
[71]: H0.reject = result_flowers$H0.reject
df_H0.reject = as.data.frame(H0.reject) %>% group_by(H0.reject) %>% count() %>%
  ↪rename("inference_count"=n)
low_order_mmd_stats = result_flowers$low_order_mmd_stats
df_low_order_mmd_stats = as.data.frame(low_order_mmd_stats)
high_order_mmd_stats = result_flowers$high_order_mmd_stats
df_high_order_mmd_stats = as.data.frame(high_order_mmd_stats)
```

Hypothesis test result

```
[72]: ggplot(data=df_H0.reject, aes(x=H0.reject, y=inference_count, fill=H0.reject)) +
  labs(title="H0 hypothesis: two images are from the same distribution for
  ↪BigGan iteration\n 2000 (alpha=0.05), dataset: flowers") +
  geom_bar(stat="identity")+
  geom_text(aes(label=inference_count), vjust=1.6, color="white", size=3.5)+
  theme_minimal()
```

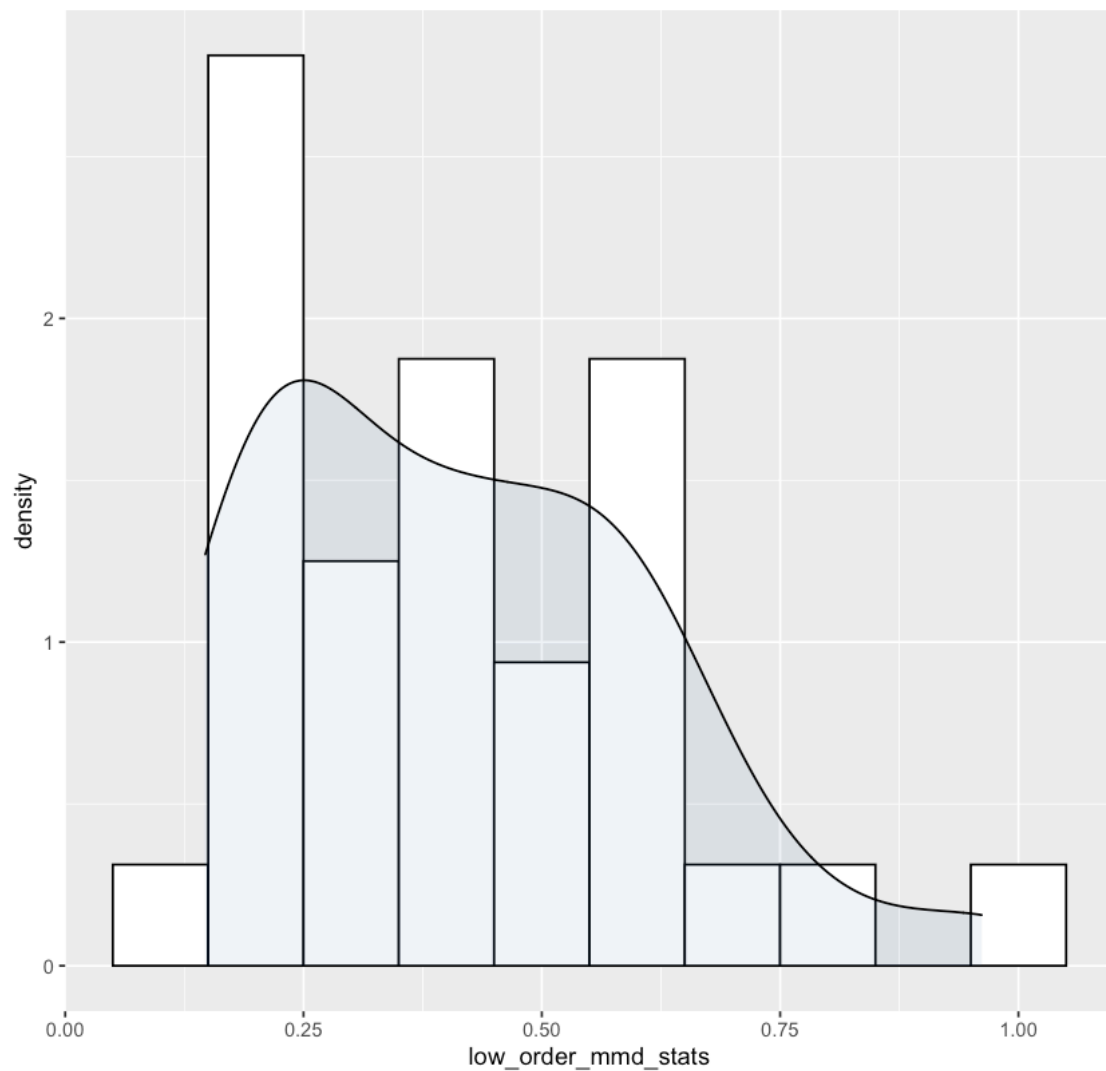
H0 hypothesis: two images are from the same distribution for BigGan iteration 2000 (alpha=0.05), dataset: flowers



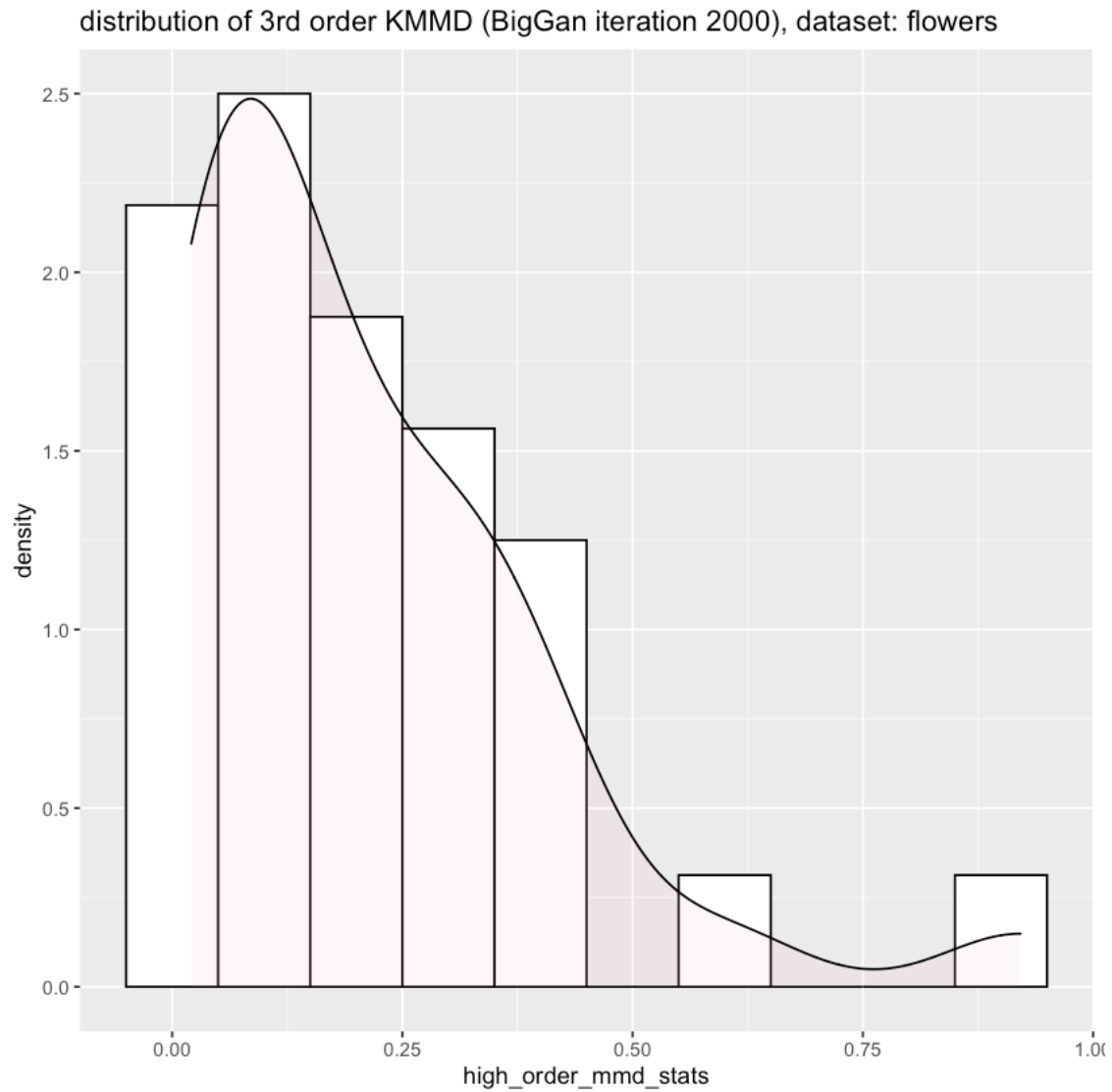
MMD statistics

```
[73]: ggplot(df_low_order_mmd_stats, aes(x=low_order_mmd_stats)) +
  labs(title="distribution of 1st order KMMD (BigGan iteration 2000), dataset:
  ↳ flowers") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↳ count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="steelblue") # Overlay with transparent
  ↳ density plot
```

distribution of 1st order KMMMD (BigGan iteration 2000), dataset: flowers



```
[74]: ggplot(df_high_order_mmd_stats, aes(x=high_order_mmd_stats)) +
  labs(title="distribution of 3rd order KMMMD (BigGan iteration 2000), dataset:
  ↪ flowers") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↪ count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="pink") # Overlay with transparent density plot
```



```
[75]: summary(df_low_order_mmd_stats)
```

```
low_order_mmd_stats
Min.   :0.1470
1st Qu.:0.2361
Median :0.4007
Mean   :0.4120
3rd Qu.:0.5547
Max.   :0.9613
```

```
[76]: summary(df_high_order_mmd_stats)
```

```
high_order_mmd_stats
```



```
Min.      :0.02075
1st Qu.   :0.05430
Median    :0.15750
Mean      :0.20625
3rd Qu.   :0.30452
Max.      :0.92120
```

1.14.2 face

```
[202]: result_face = compute_kmmd("./biggan_face_results/input", "./
      ↪biggan_face_results/iter_2000")
```

[illegible]

```

Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel

```

```

[203]: H0.reject = result_face$H0.reject
df_H0.reject = as.data.frame(H0.reject) %>% group_by(H0.reject) %>% count() %>%
  ↪rename("inference_count"=n)
low_order_mmd_stats = result_face$low_order_mmd_stats
df_low_order_mmd_stats = as.data.frame(low_order_mmd_stats)
high_order_mmd_stats = result_face$high_order_mmd_stats
df_high_order_mmd_stats = as.data.frame(high_order_mmd_stats)

```

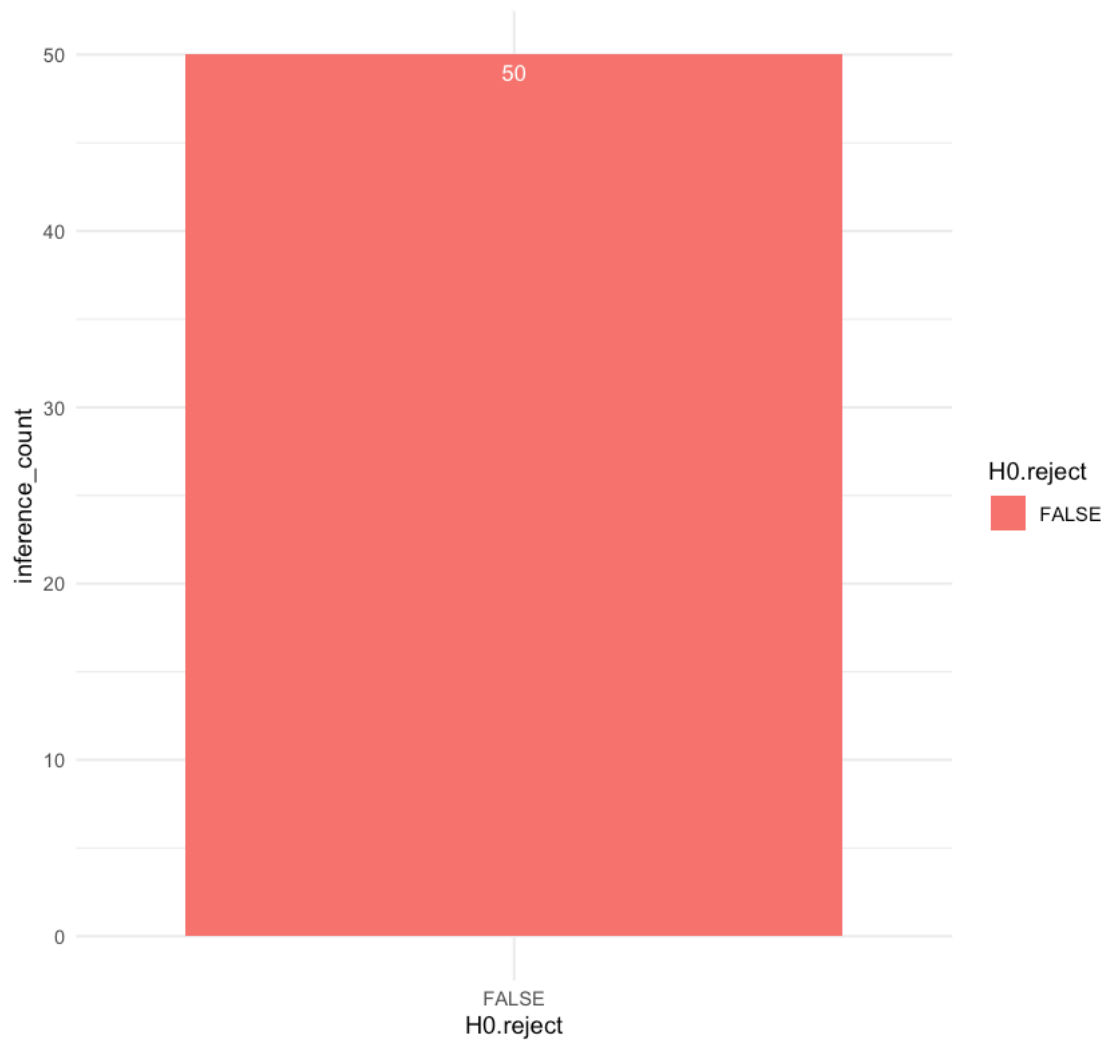
Hypothesis test result

```

[204]: ggplot(data=df_H0.reject, aes(x=H0.reject, y=inference_count, fill=H0.reject)) +
  labs(title="H0 hypothesis: two images are from the same distribution for_
  ↪BigGan iteration\n 2000 (alpha=0.05), dataset: face") +
  geom_bar(stat="identity")+
  geom_text(aes(label=inference_count), vjust=1.6, color="white", size=3.5)+
  theme_minimal()

```

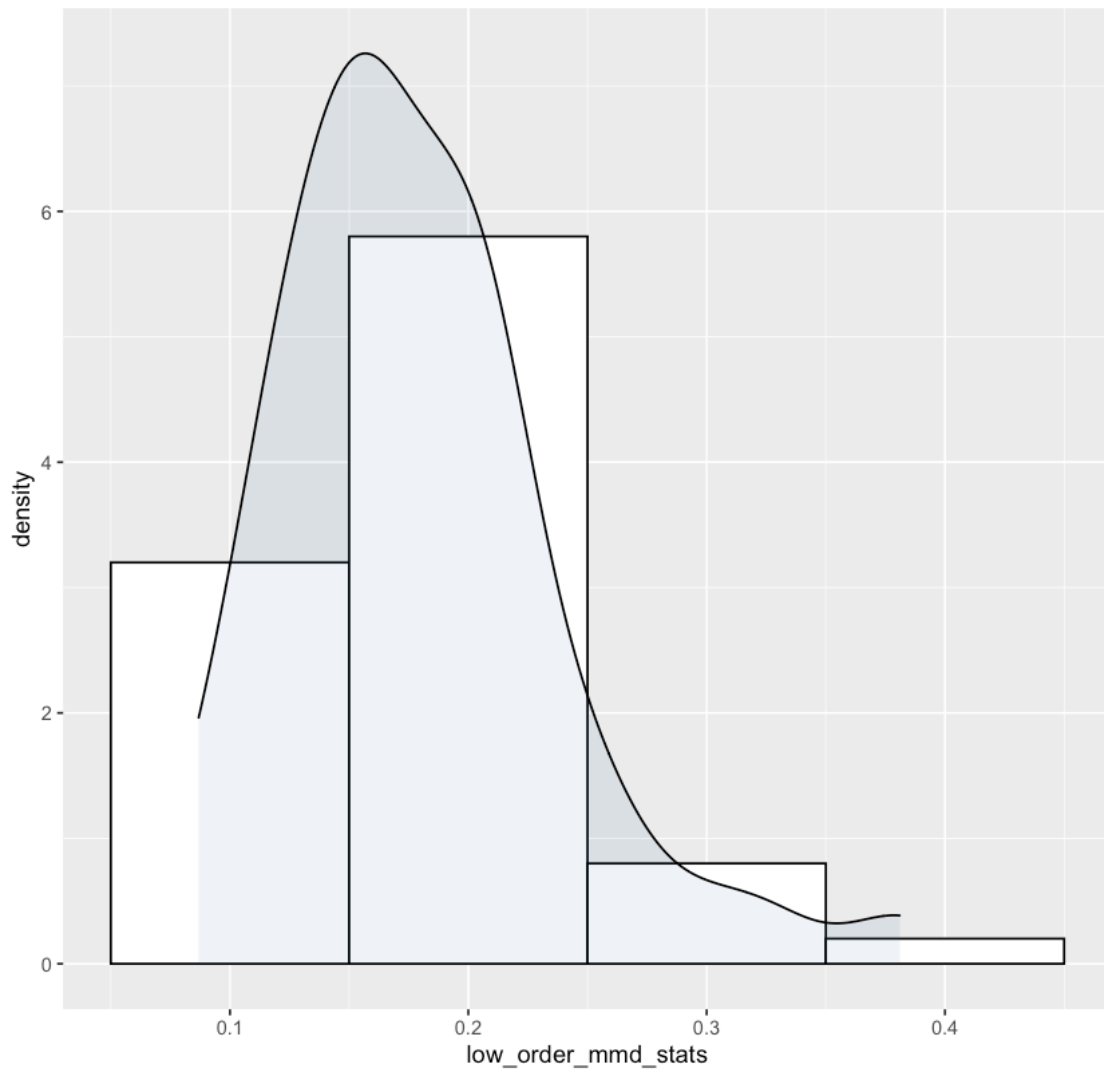
H0 hypothesis: two images are from the same distribution for BigGan iteration 2000 (alpha=0.05), dataset: face



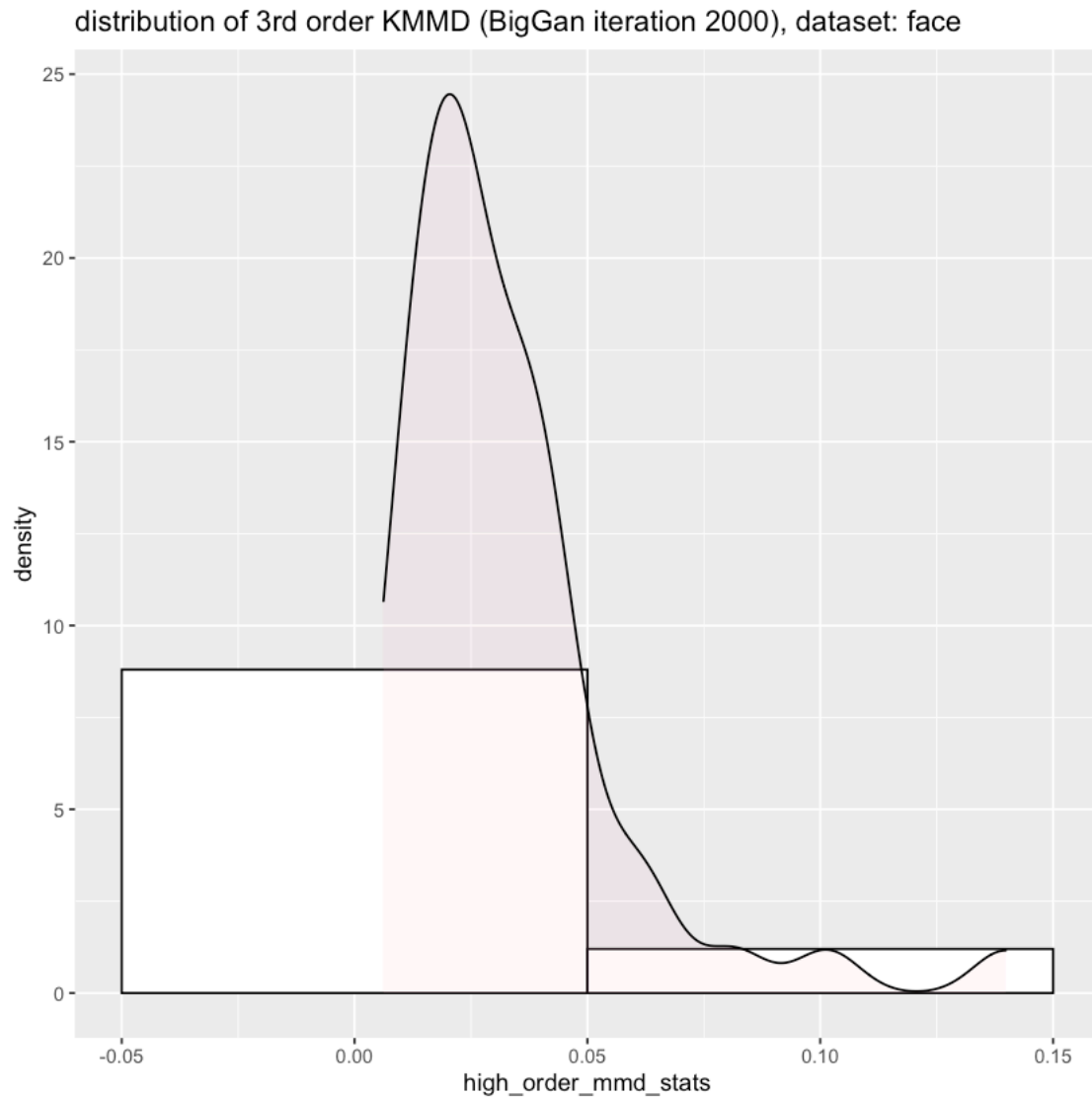
MMD statistics

```
[205]: ggplot(df_low_order_mmd_stats, aes(x=low_order_mmd_stats)) +
  labs(title="distribution of 1st order KMMD (BigGan iteration 2000), dataset:
  ↳ face") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↳ count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="steelblue") # Overlay with transparent
  ↳ density plot
```

distribution of 1st order KMMD (BigGan iteration 2000), dataset: face



```
[206]: ggplot(df_high_order_mmd_stats, aes(x=high_order_mmd_stats)) +
  labs(title="distribution of 3rd order KMMD (BigGan iteration 2000), dataset:
  ↳ face") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↳ count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="pink") # Overlay with transparent density plot
```



```
[207]: summary(df_low_order_mmd_stats)
```

```
low_order_mmd_stats
Min.   :0.08686
1st Qu.:0.13603
Median :0.16570
Mean   :0.17887
3rd Qu.:0.20574
Max.   :0.38098
```

```
[208]: summary(df_high_order_mmd_stats)
```

```
high_order_mmd_stats
```

```
Min.      :0.006209
1st Qu.:0.017121
Median   :0.025473
Mean     :0.032912
3rd Qu.:0.039507
Max.     :0.139827
```

1.14.3 anime

```
[217]: result_anime = compute_kmmd("./biggan_anime_results/input", "↪biggan_anime_results/iter_2000")
```

[illegible]

```

Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel

```

```

[218]: H0.reject = result_anime$H0.reject
df_H0.reject = as.data.frame(H0.reject) %>% group_by(H0.reject) %>% count() %>%
  ↪rename("inference_count"=n)
low_order_mmd_stats = result_anime$low_order_mmd_stats
df_low_order_mmd_stats = as.data.frame(low_order_mmd_stats)
high_order_mmd_stats = result_anime$high_order_mmd_stats
df_high_order_mmd_stats = as.data.frame(high_order_mmd_stats)

```

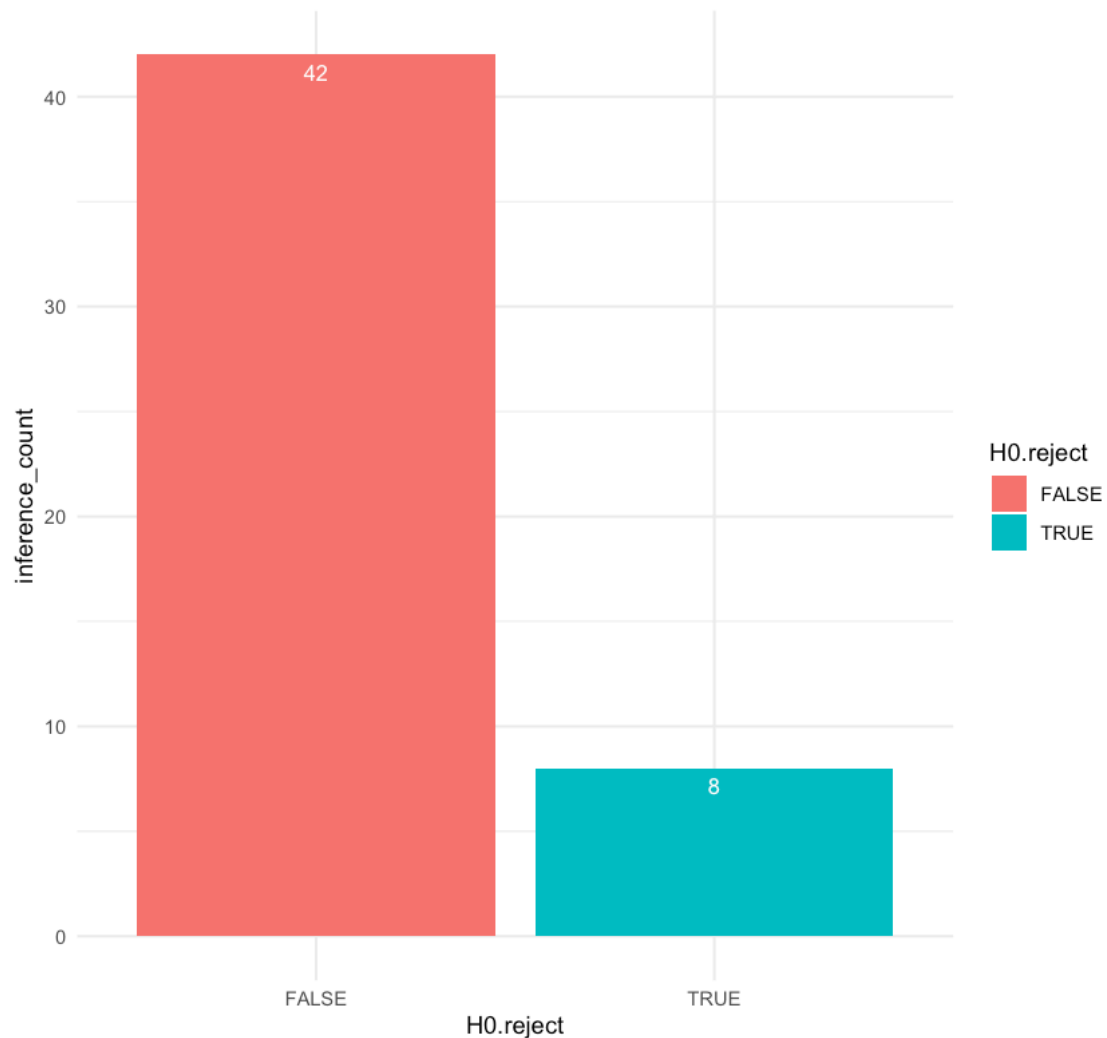
Hypothesis test result

```

[219]: ggplot(data=df_H0.reject, aes(x=H0.reject, y=inference_count, fill=H0.reject)) +
  labs(title="H0 hypothesis: two images are from the same distribution for_
  ↪BigGan iteration\n 2000 (alpha=0.05), dataset: anime") +
  geom_bar(stat="identity")+
  geom_text(aes(label=inference_count), vjust=1.6, color="white", size=3.5)+
  theme_minimal()

```

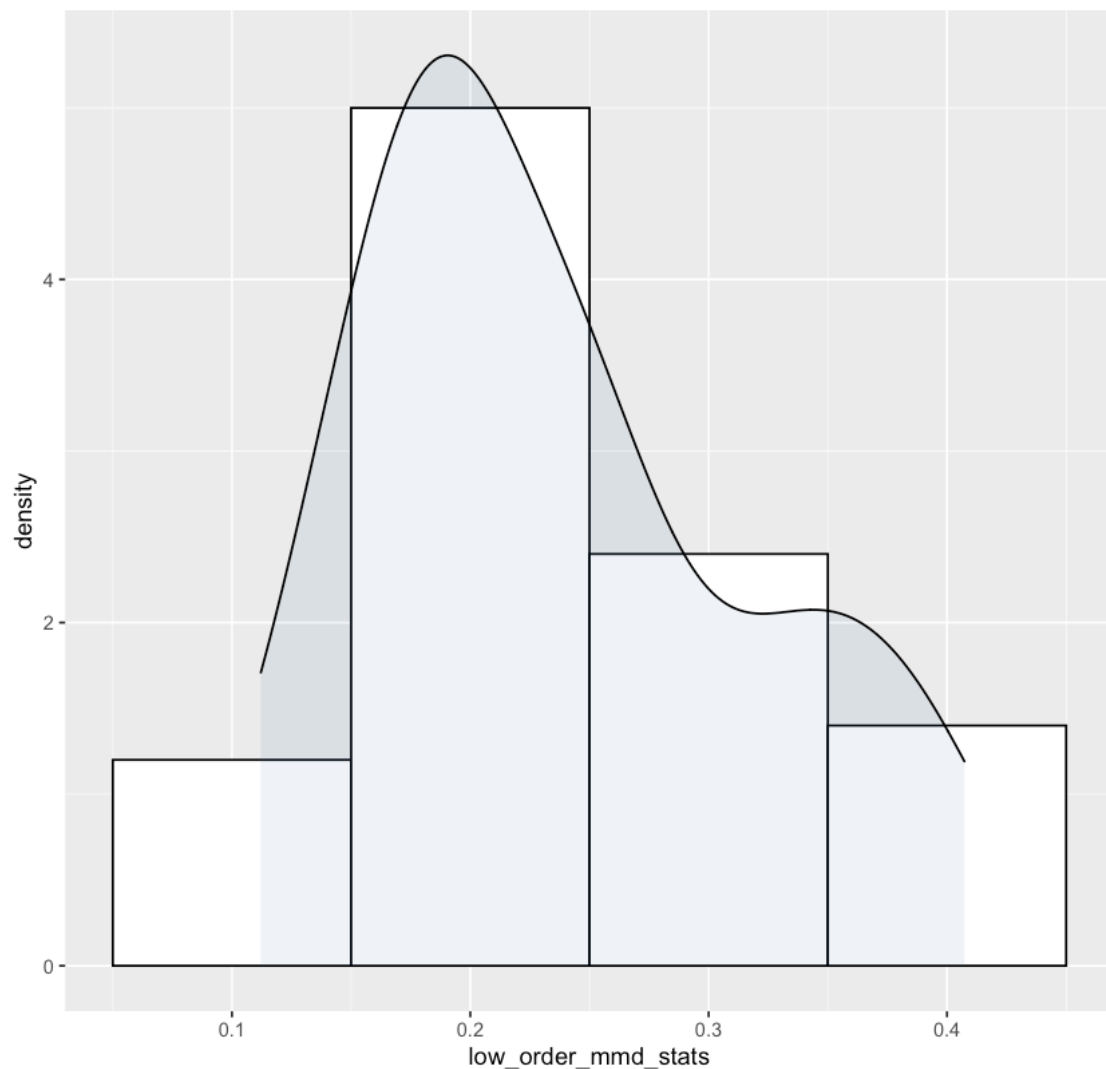
H0 hypothesis: two images are from the same distribution for BigGan iteration 2000 (alpha=0.05), dataset: anime



MMD statistics

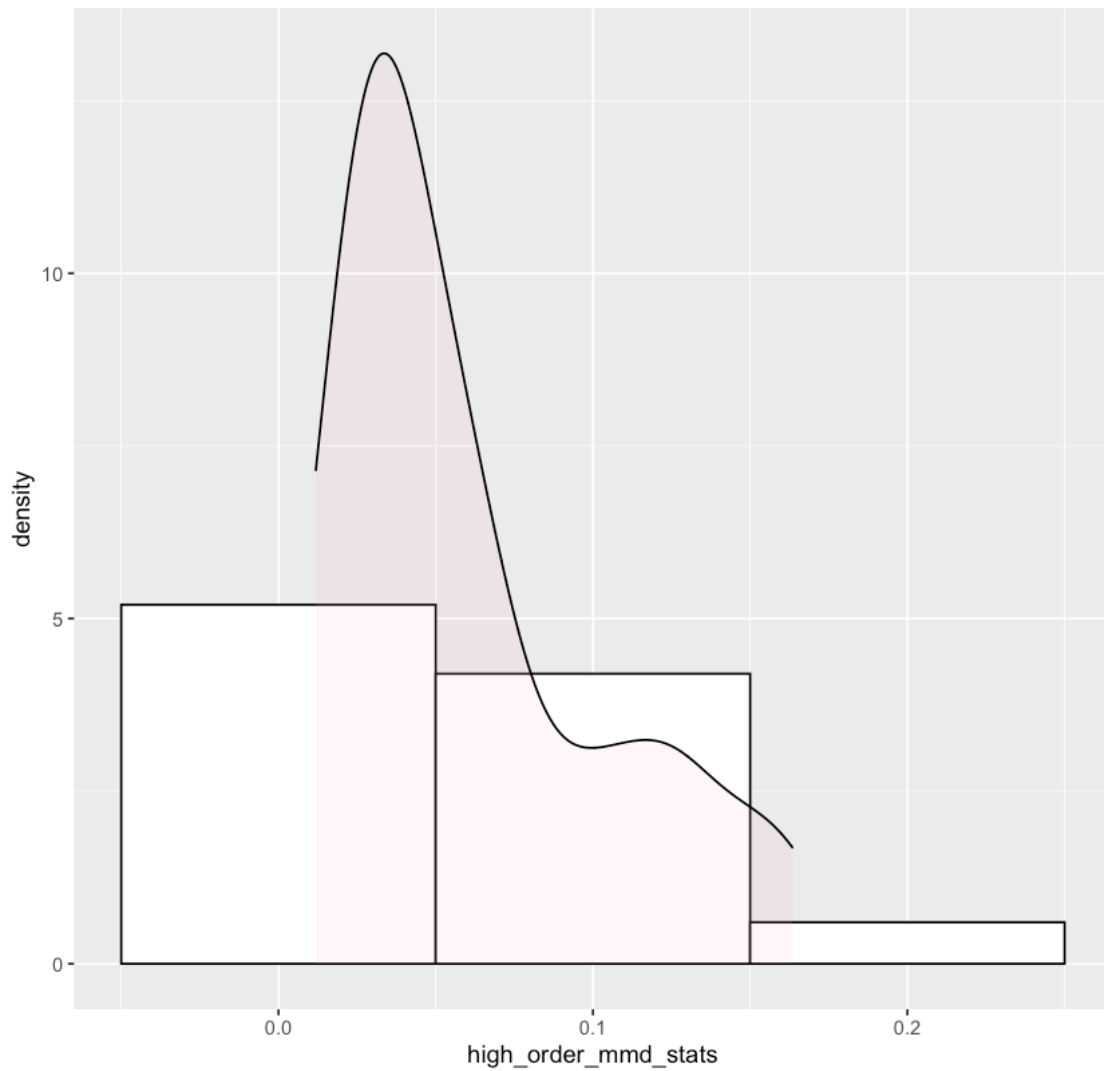
```
[220]: ggplot(df_low_order_mmd_stats, aes(x=low_order_mmd_stats)) +
  labs(title="distribution of 1st order KMMD (BigGan iteration 2000), dataset:
  → anime") +
  geom_histogram(aes(y=..density..),      # Histogram with density instead of
  → count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="steelblue") # Overlay with transparent
  → density plot
```


distribution of 1st order KMMD (BigGan iteration 2000), dataset: anime



```
[221]: ggplot(df_high_order_mmd_stats, aes(x=high_order_mmd_stats)) +
  labs(title="distribution of 3rd order KMMD (BigGan iteration 2000), dataset:
  ↪ anime") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↪ count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="pink") # Overlay with transparent density plot
```

distribution of 3rd order KMMD (BigGan iteration 2000), dataset: anime



```
[222]: summary(df_low_order_mmd_stats)
```

```
low_order_mmd_stats
Min.   :0.1121
1st Qu.:0.1814
Median :0.2156
Mean   :0.2367
3rd Qu.:0.2823
Max.   :0.4073
```

```
[223]: summary(df_high_order_mmd_stats)
```

```
high_order_mmd_stats
```

```
Min.      :0.01187
1st Qu.   :0.03117
Median    :0.04444
Mean      :0.06039
3rd Qu.   :0.07756
Max.      :0.16341
```

1.15 Result of datasets for BigGan (iteration 5000)

1.15.1 flowers

```
[231]: result_flowerns = compute_kmmd("./biggan_flower_results/input", "./
      ↪biggan_flower_results/iter_5000")
```

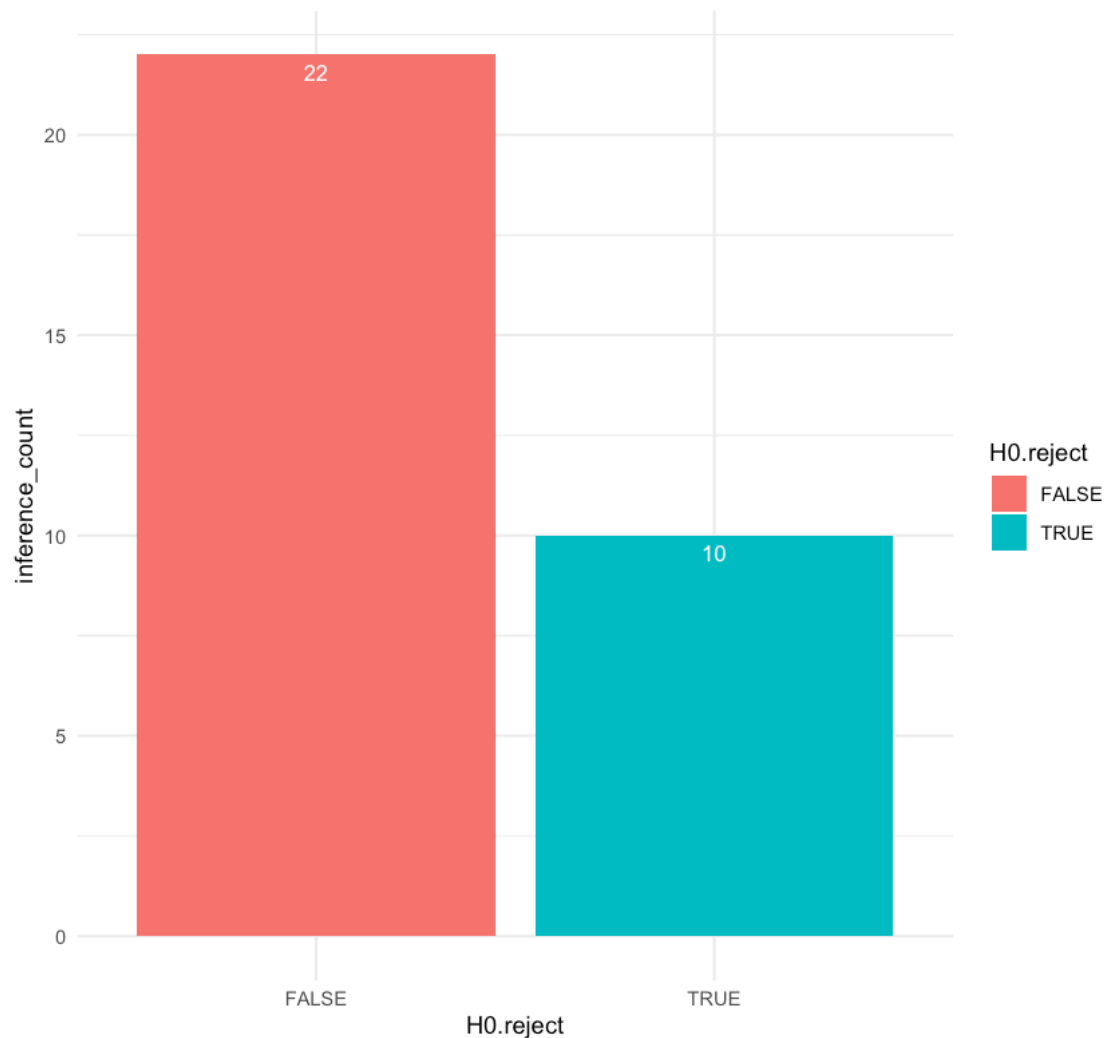
[illegible]

```
[232]: H0.reject = result_flowers$H0.reject
df_H0.reject = as.data.frame(H0.reject) %>% group_by(H0.reject) %>% count() %>%
  ↪rename("inference_count"=n)
low_order_mmd_stats = result_flowers$low_order_mmd_stats
df_low_order_mmd_stats = as.data.frame(low_order_mmd_stats)
high_order_mmd_stats = result_flowers$high_order_mmd_stats
df_high_order_mmd_stats = as.data.frame(high_order_mmd_stats)
```

Hypothesis test result

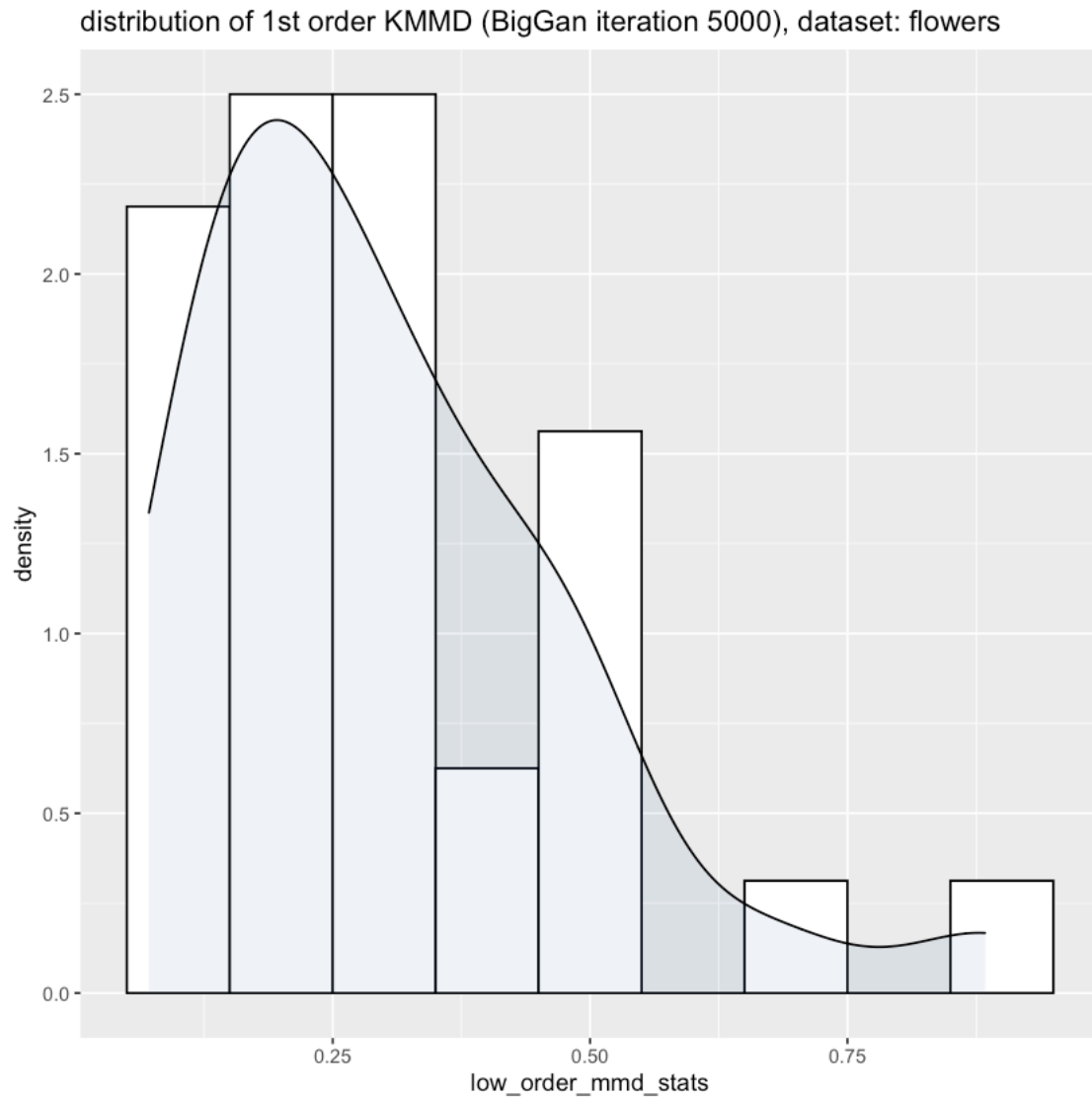
```
[233]: ggplot(data=df_H0.reject, aes(x=H0.reject, y=inference_count, fill=H0.reject)) +
  labs(title="H0 hypothesis: two images are from the same distribution for
  ↪BigGan iteration\n 5000 (alpha=0.05), dataset: flowers") +
  geom_bar(stat="identity")+
  geom_text(aes(label=inference_count), vjust=1.6, color="white", size=3.5)+
  theme_minimal()
```

H0 hypothesis: two images are from the same distribution for BigGan iteration 5000 (alpha=0.05), dataset: flowers

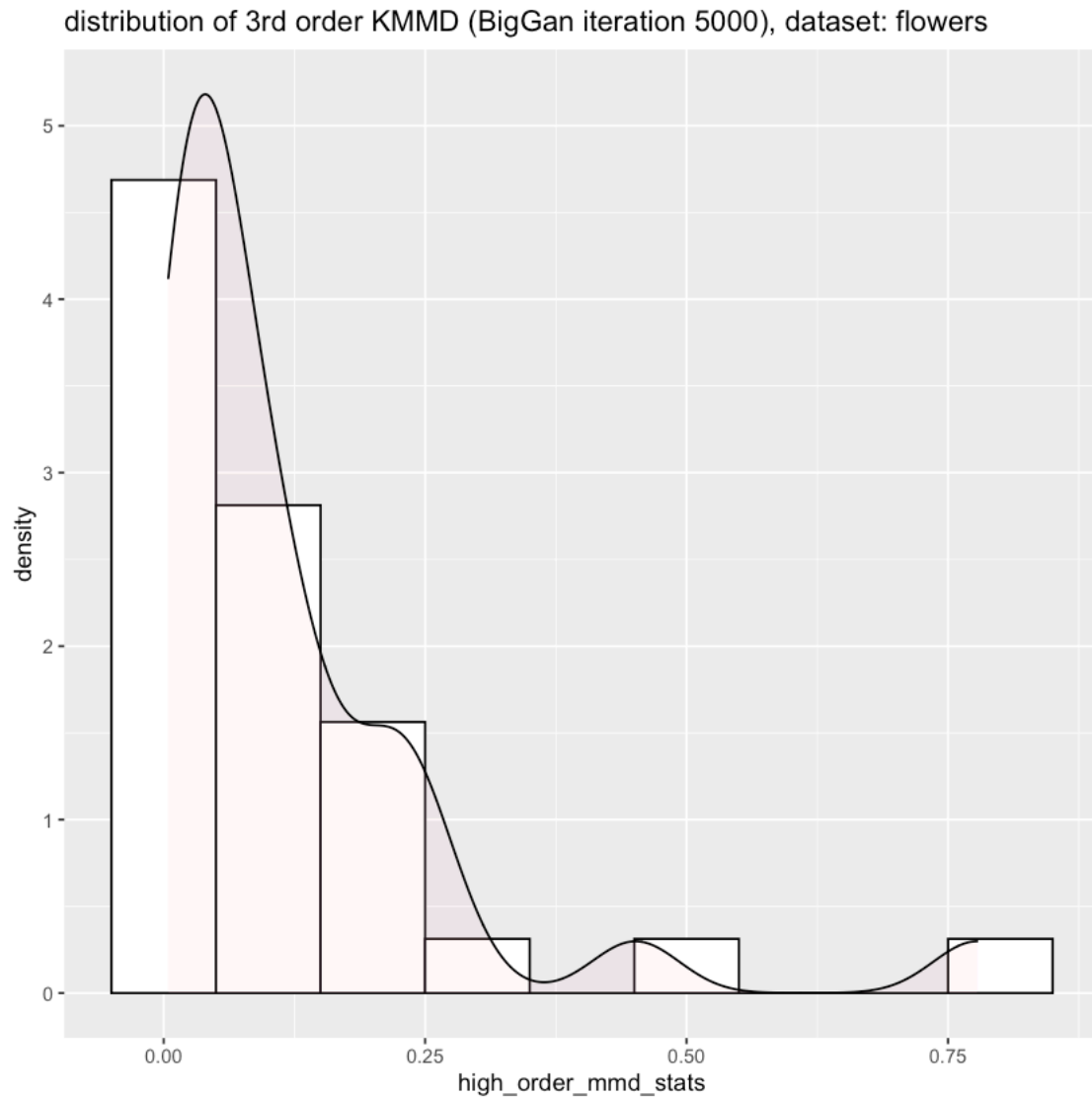


MMD statistics

```
[234]: ggplot(df_low_order_mmd_stats, aes(x=low_order_mmd_stats)) +
  labs(title="distribution of 1st order KMMD (BigGan iteration 5000), dataset:
  ↳ flowers") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↳ count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="steelblue") # Overlay with transparent
  ↳ density plot
```



```
[235]: ggplot(df_high_order_mmd_stats, aes(x=high_order_mmd_stats)) +
  labs(title="distribution of 3rd order KMMD (BigGan iteration 5000), dataset:
  → flowers") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  → count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="pink") # Overlay with transparent density plot
```



```
[238]: summary(df_low_order_mmd_stats)
```

```
low_order_mmd_stats
Min.   :0.07145
1st Qu.:0.16366
Median :0.25251
Mean   :0.29985
3rd Qu.:0.39105
Max.   :0.88358
```

```
[239]: summary(df_high_order_mmd_stats)
```

```
high_order_mmd_stats
```

```
Min.      :0.004349
1st Qu.   :0.025602
Median    :0.061978
Mean      :0.119626
3rd Qu.   :0.150000
Max.      :0.777871
```

1.15.2 face

```
[240]: result_face = compute_kmmd("./biggan_face_results/input", "./
      ↪biggan_face_results/iter_5000")
```

[illegible]


```

Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel

```

```

[241]: H0.reject = result_face$H0.reject
df_H0.reject = as.data.frame(H0.reject) %>% group_by(H0.reject) %>% count() %>%
  ↪rename("inference_count"=n)
low_order_mmd_stats = result_face$low_order_mmd_stats
df_low_order_mmd_stats = as.data.frame(low_order_mmd_stats)
high_order_mmd_stats = result_face$high_order_mmd_stats
df_high_order_mmd_stats = as.data.frame(high_order_mmd_stats)

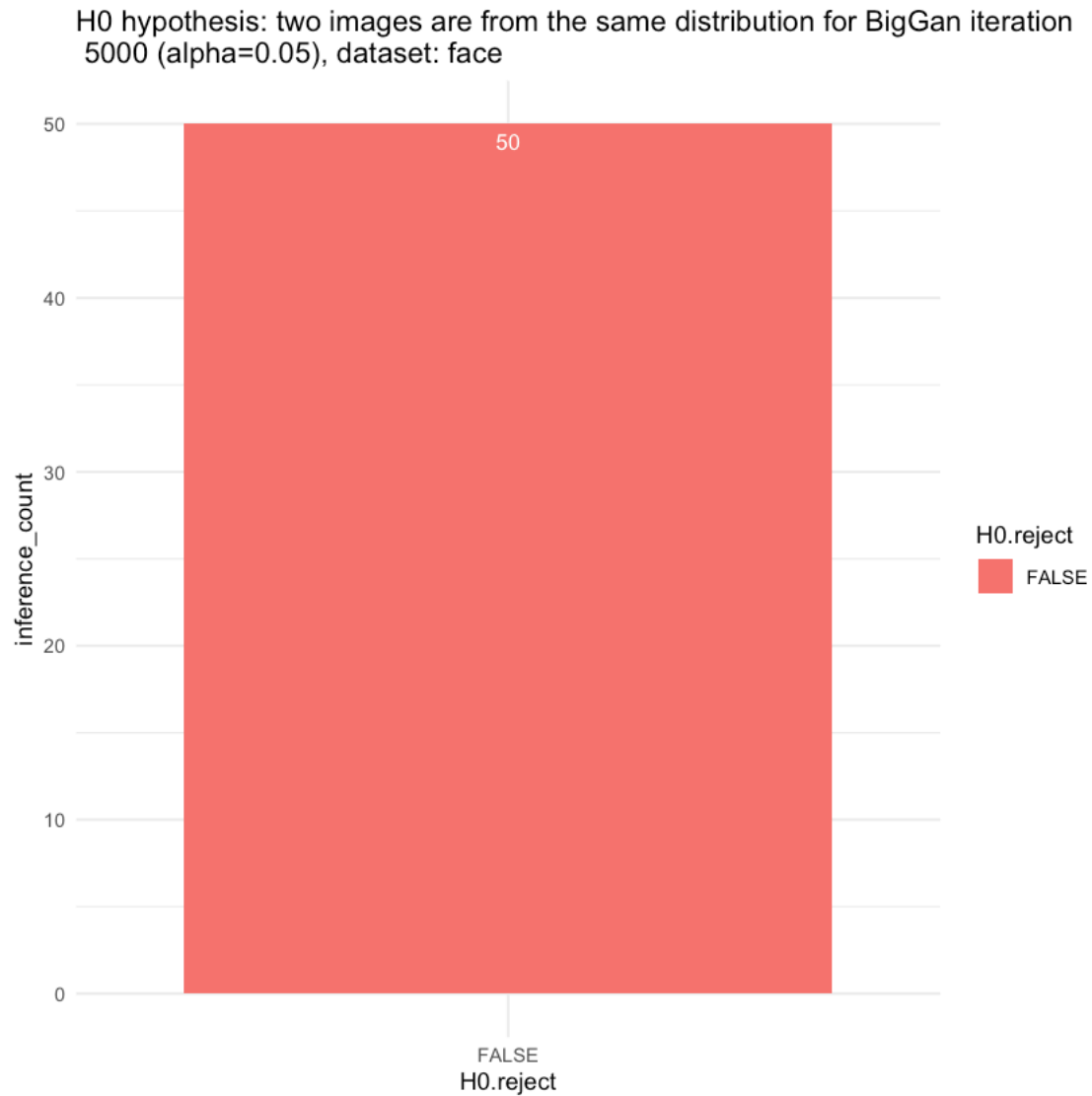
```

Hypothesis test result

```

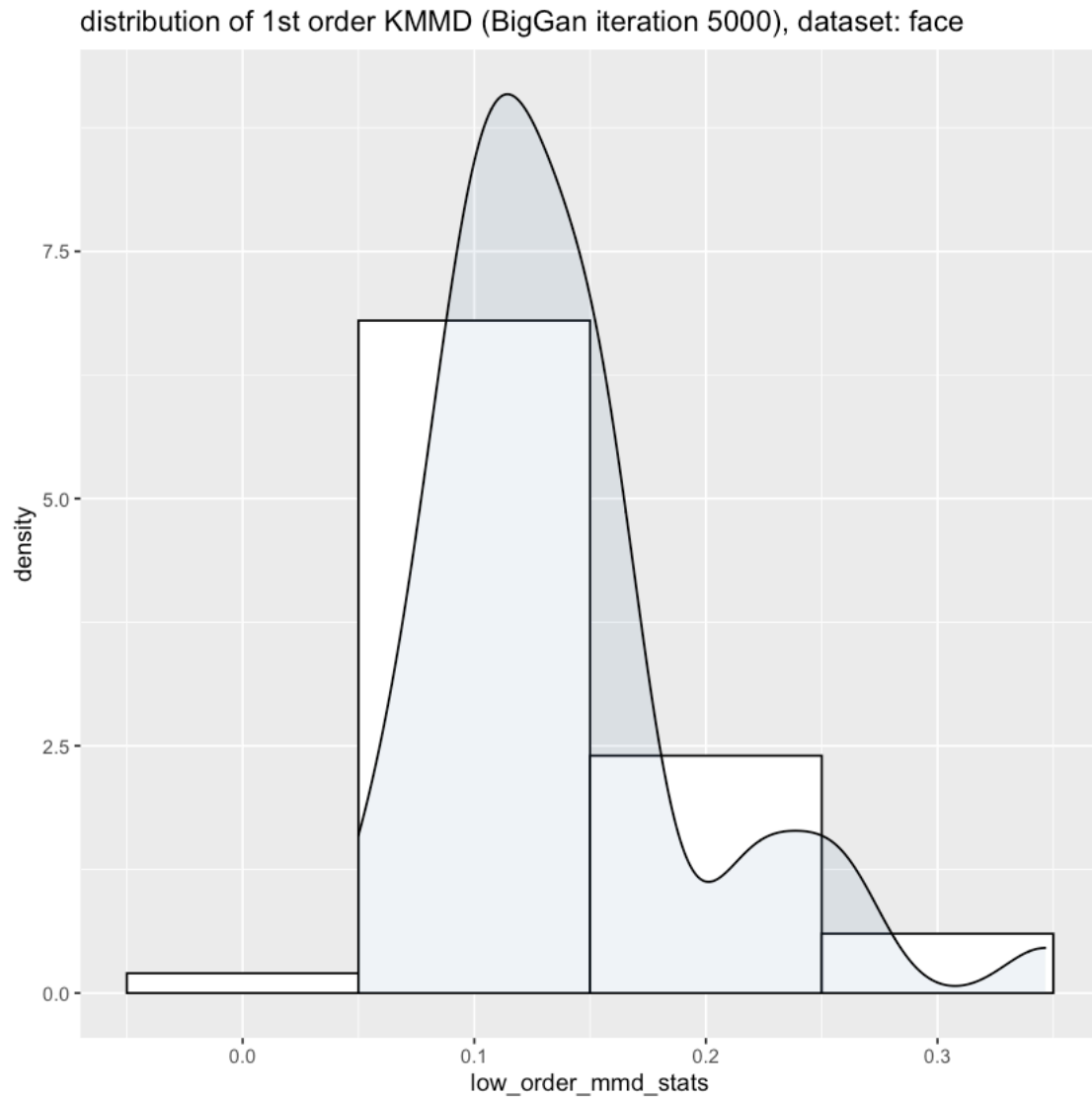
[247]: ggplot(data=df_H0.reject, aes(x=H0.reject, y=inference_count, fill=H0.reject)) +
  labs(title="H0 hypothesis: two images are from the same distribution for_
  ↪BigGan iteration\n 5000 (alpha=0.05), dataset: face") +
  geom_bar(stat="identity")+
  geom_text(aes(label=inference_count), vjust=1.6, color="white", size=3.5)+
  theme_minimal()

```

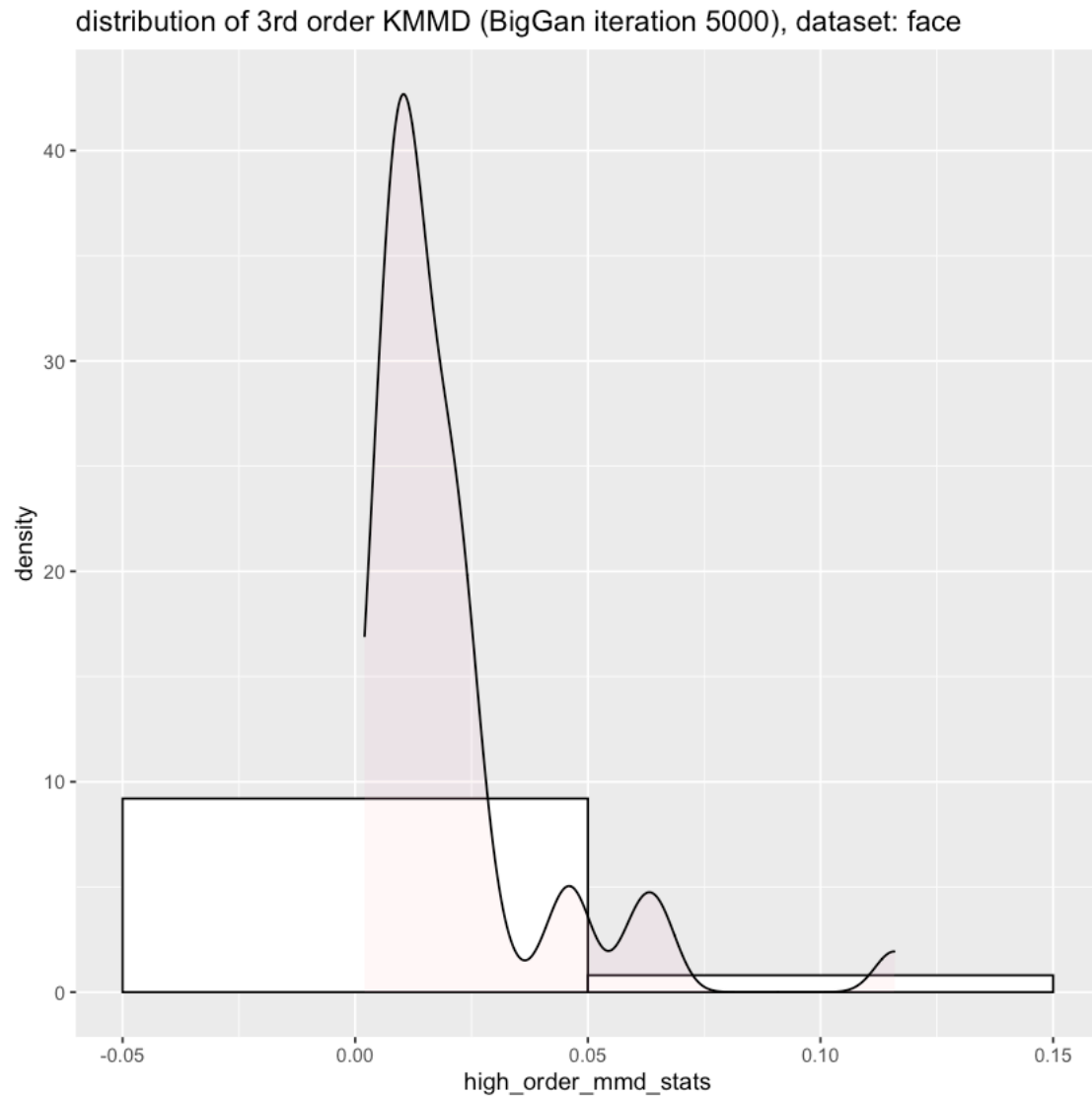


MMD statistics

```
[248]: ggplot(df_low_order_mmd_stats, aes(x=low_order_mmd_stats)) +
  labs(title="distribution of 1st order KMMD (BigGan iteration 5000), dataset:
  ↳ face") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↳ count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="steelblue") # Overlay with transparent
  ↳ density plot
```



```
[249]: ggplot(df_high_order_mmd_stats, aes(x=high_order_mmd_stats)) +
  labs(title="distribution of 3rd order KMMD (BigGan iteration 5000), dataset:
  ↳ face") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  ↳ count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="pink") # Overlay with transparent density plot
```



```
[250]: summary(df_low_order_mmd_stats)
```

```
low_order_mmd_stats
Min.   :0.04999
1st Qu.:0.09949
Median :0.12403
Mean   :0.13767
3rd Qu.:0.15642
Max.   :0.34648
```

```
[251]: summary(df_high_order_mmd_stats)
```

```
high_order_mmd_stats
```

```
Min.      :0.002009
1st Qu.   :0.008974
Median    :0.013758
Mean      :0.020511
3rd Qu.   :0.022401
Max.      :0.115854
```

1.15.3 anime

```
[266]: result_anime = compute_kmmd("./biggan_face_results/input", "\n      ↪biggan_anime_results/iter_5000")
```

[illegible]

```

Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel
Using automatic sigma estimation (sigest) for RBF or laplace kernel

```

```

[267]: H0.reject = result_anime$H0.reject
df_H0.reject = as.data.frame(H0.reject) %>% group_by(H0.reject) %>% count() %>%
  ↪rename("inference_count"=n)
low_order_mmd_stats = result_anime$low_order_mmd_stats
df_low_order_mmd_stats = as.data.frame(low_order_mmd_stats)
high_order_mmd_stats = result_anime$high_order_mmd_stats
df_high_order_mmd_stats = as.data.frame(high_order_mmd_stats)

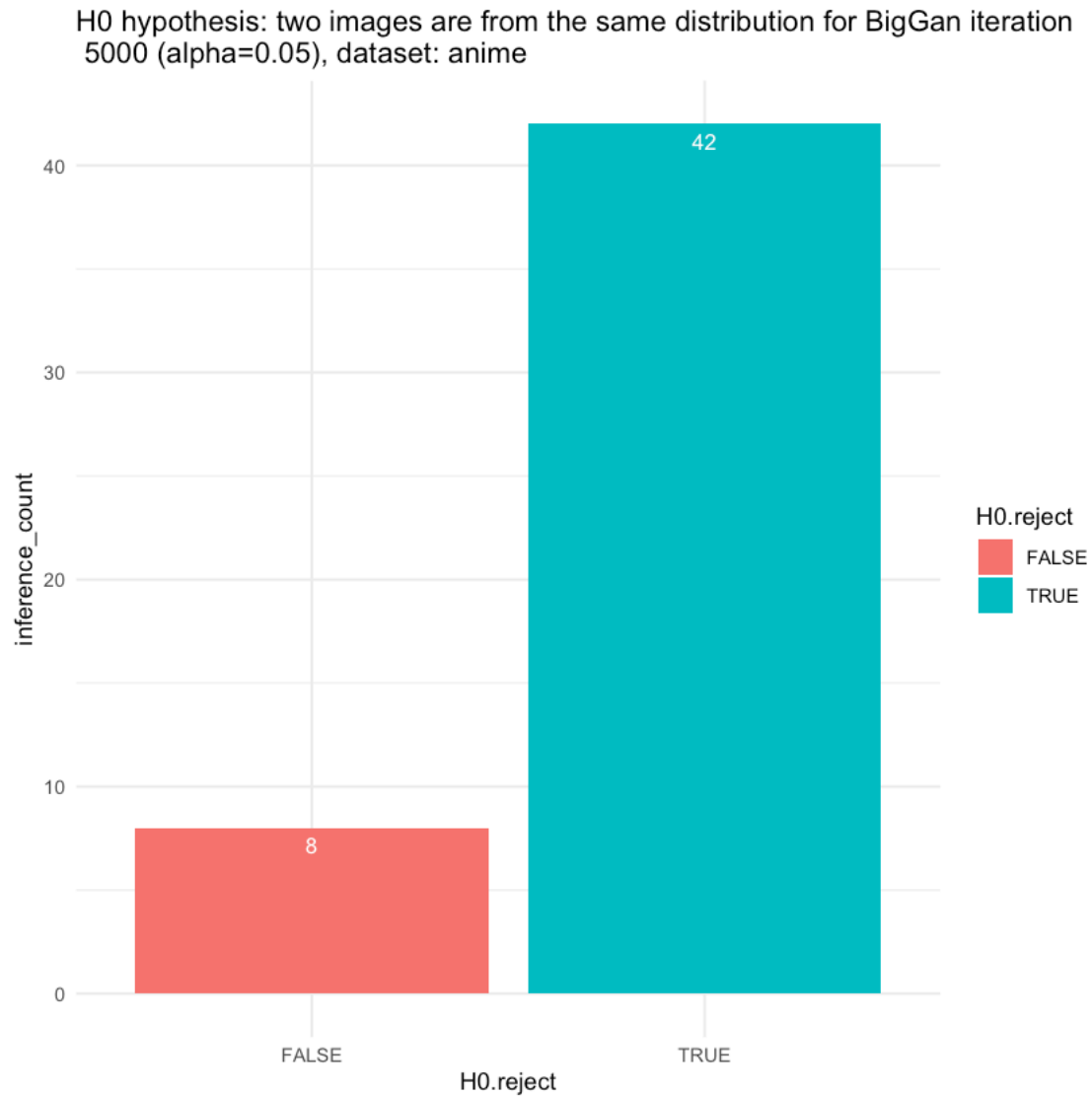
```

Hypothesis test result

```

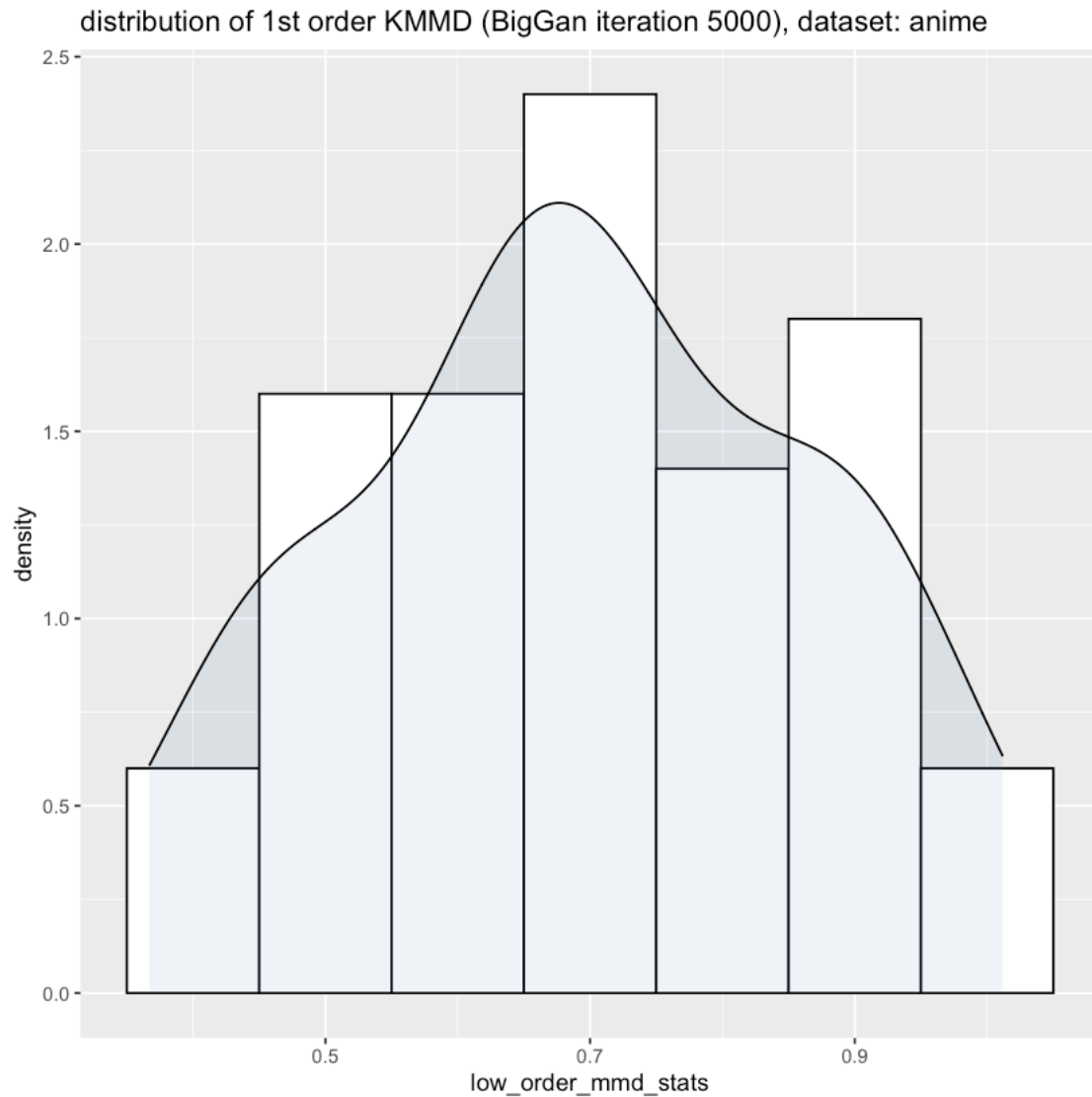
[268]: ggplot(data=df_H0.reject, aes(x=H0.reject, y=inference_count, fill=H0.reject)) +
  labs(title="H0 hypothesis: two images are from the same distribution for_
  ↪BigGan iteration\n 5000 (alpha=0.05), dataset: anime") +
  geom_bar(stat="identity")+
  geom_text(aes(label=inference_count), vjust=1.6, color="white", size=3.5)+
  theme_minimal()

```

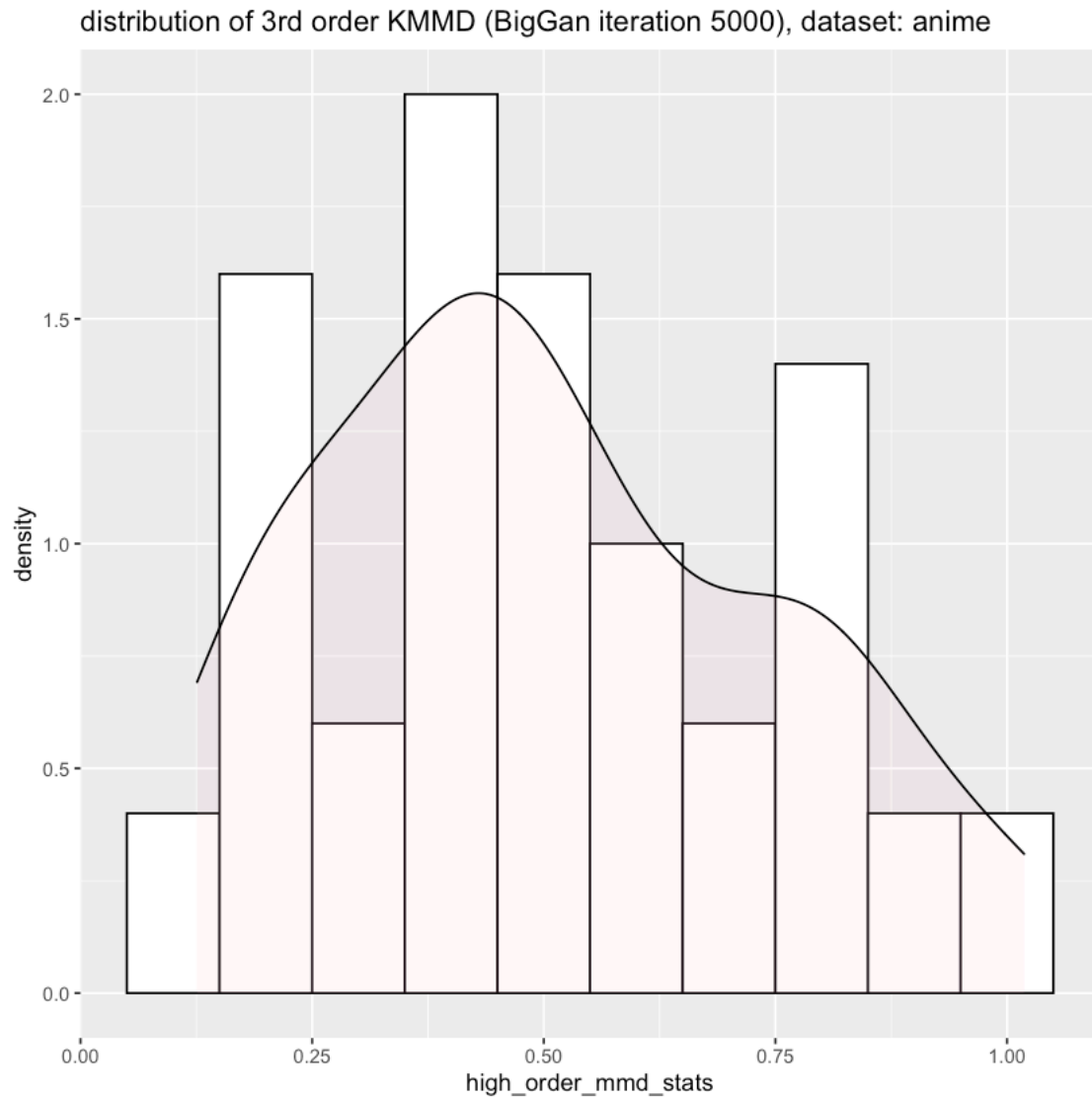


MMD statistics

```
[269]: ggplot(df_low_order_mmd_stats, aes(x=low_order_mmd_stats)) +
  labs(title="distribution of 1st order KMMD (BigGan iteration 5000), dataset:
  → anime") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  → count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="steelblue") # Overlay with transparent
  → density plot
```



```
[270]: ggplot(df_high_order_mmd_stats, aes(x=high_order_mmd_stats)) +
  labs(title="distribution of 3rd order KMMD (BigGan iteration 5000), dataset:
  → anime") +
  geom_histogram(aes(y=..density..),          # Histogram with density instead of
  → count on y-axis
    binwidth=.1,
    colour="black", fill="white") +
  geom_density(alpha=.1, fill="pink") # Overlay with transparent density plot
```

```
[271]: summary(df_low_order_mmd_stats)
```

```
low_order_mmd_stats
Min.   :0.3673
1st Qu.:0.5990
Median :0.6815
Mean   :0.6939
3rd Qu.:0.8323
Max.   :1.0115
```

```
[272]: summary(df_high_order_mmd_stats)
```

```
high_order_mmd_stats
```

```
Min.    :0.1258
1st Qu.:0.3501
Median  :0.4570
Mean    :0.5035
3rd Qu.:0.6853
Max.    :1.0185
```

[]: