

STAT 631 Homework 3

Jack Cunningham (jgavc@tamu.edu)

9/16/24

1)

```
library(MASS) # need for mvrnorm
par(mfrow=c(2,2))
N = 2500
nu = 3
set.seed(5640)
cov=matrix(c(1, 0.8, 0.8, 1), nrow = 2)
x= mvrnorm(N, mu = c(0, 0), Sigma = cov)
w = sqrt(nu / rchisq(N, df = nu))
x=x*cbind(w, w)
plot(x, main = "(a)")

set.seed(5640)
cov=matrix(c(1, 0.8, 0.8, 1),nrow = 2)
x= mvrnorm(N, mu = c(0, 0), Sigma = cov)
w1 = sqrt(nu / rchisq(N, df = nu))
w2 = sqrt(nu / rchisq(N, df = nu))
x=x*cbind(w1, w2)
plot(x, main = "(b)")

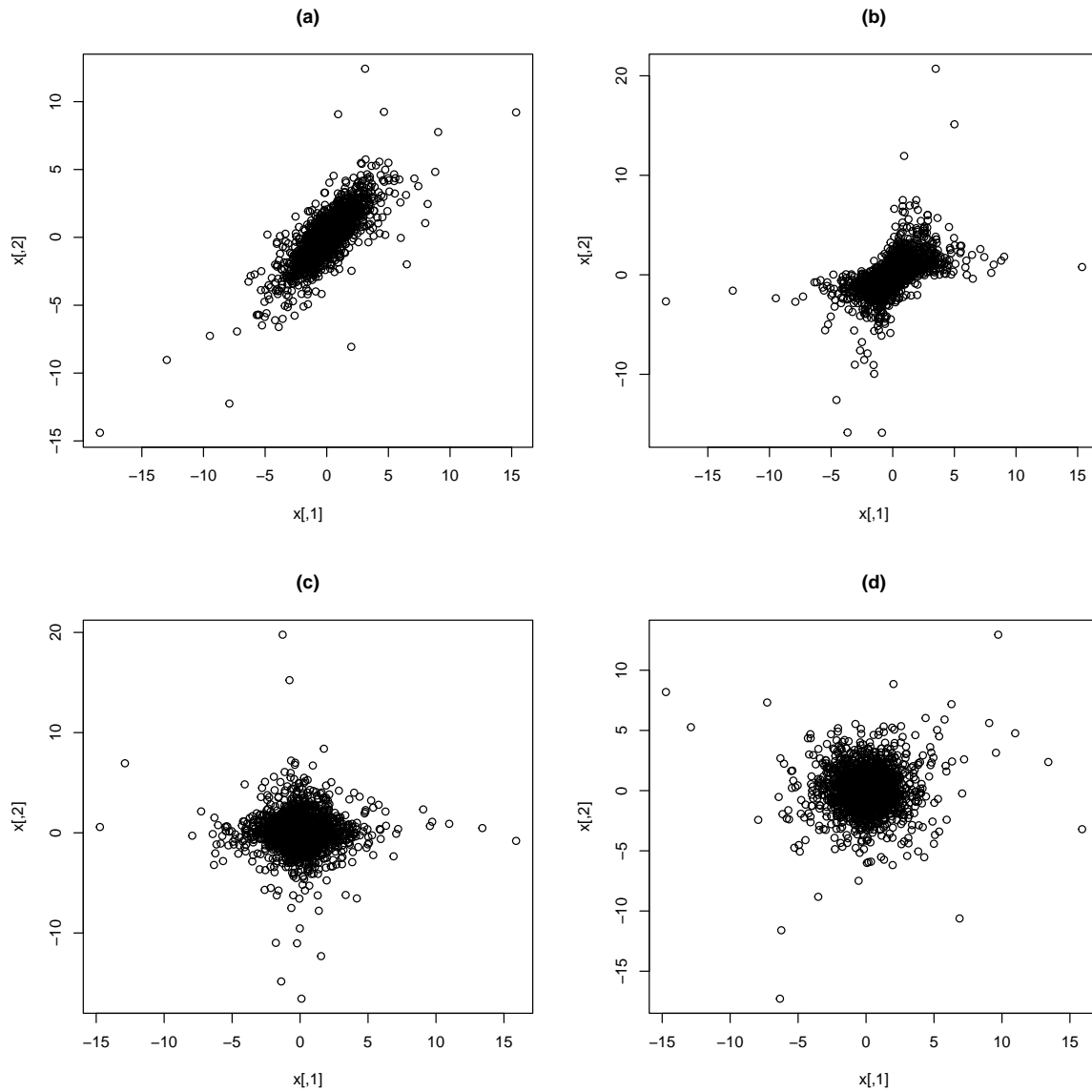
set.seed(5640)
cov=matrix(c(1, 0, 0, 1), nrow = 2)
x= mvrnorm(N, mu = c(0, 0), Sigma = cov)
w1 = sqrt(nu / rchisq(N, df = nu))
w2 = sqrt(nu / rchisq(N, df = nu))
x=x*cbind(w1, w2)
plot(x, main = "(c)")

set.seed(5640)
```

```

cov=matrix(c(1, 0, 0, 1), nrow = 2)
x= mvrnorm(N, mu = c(0, 0), Sigma = cov)
w = sqrt(nu / rchisq(N, df = nu))
x=x*cbind(w, w)
plot(x, main = "(d)")

```



Problem 3)

The sample with independent variates is c), we can see the density is circular. We can also

see that there is no tail dependence, extreme values in $x[, 1]$ are not correlated with extreme values in $x[, 2]$ and vice versa.

Also if we look at the code used, the following covariance matrix Σ is used in the simulation of the multivariate normal data.

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Since Σ is diagonal we know components x_1, x_2 are independent.

We know there is no tail dependence because when we generate the marginal t-distributions using x_1, x_2 the random chi-square component is different for w_1, w_2 . There is no shared component in the marginal t-distributions.

Problem 4)

The sample with correlated variates that do not have tail dependence is b). We can see the correlation from the shape in the center of the density, it is diagonal stretching both to the top right and the bottom left.

If we look at the code we can see the following Σ matrix when generating the multivariate normal data:

$$\Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}$$

We can see the covariance in the non diagonal entries.

We know there is no tail dependence because we generate different chi-square components for each marginal t distribution.

Problem 5)

The sample with uncorrelated variates with tail dependence is d. We can see the density shape in the middle is circular but we also see that extreme values in $x[, 1]$ are also often extreme in $x[, 2]$ and vice versa.

If we look at the code we can see the following Σ matrix when generating the multivariate normal data:

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Since Σ is diagonal, the variates are independent.

We know there is tail dependence because each marginal t distribution in the multivariate t-distribution share the same chi-square component.

Problem 6)

a)

We know the below:

$$\nu = 5, \mu = \begin{bmatrix} 0.001 \\ 0.002 \end{bmatrix}, \Sigma = \begin{bmatrix} 0.10 & 0.03 \\ 0.03 & 0.15 \end{bmatrix}$$

We use the relationship $\Sigma = \frac{\nu}{\nu-2}\Lambda$. Then:

$$\Lambda = \frac{3}{5} \begin{bmatrix} 0.1 & 0.03 \\ 0.03 & 0.15 \end{bmatrix}$$

So then the multivariate t-distribution of (X, Y) , which we can label as S is:

$$S \sim t_5(\mu, \Lambda)$$

Since R is a portfolio of the two stocks and they are evenly weighted we can have weight vector $w = [0.5, 0.5]^T$. Then, since S is a multivariate t-distribution we can say:

$$R = w^T S \sim t_\nu(w^T \mu, w^T \Lambda w)$$

```
w <- c(.5, .5)
Lambda <- matrix(c(.1, 0.03, 0.03, 0.15), nrow = 2, ncol = 2)
mu <- c(0.001, 0.002)
R_mu <- t(w) %*% mu
R_lambda <- t(w) %*% Lambda %*% w

c(R_mu, R_lambda)
```

```
[1] 0.0015 0.0775
```

$$R \sim t_5(0.0015, 0.0775)$$

b)

```

set.seed(200128)
x <- 0.0015 + rt(10000, df = 5)*sqrt(0.0775)
quantile_99 <- quantile(x, probs = .99)
avg_greater <- mean(x[which(x > quantile_99)])
results <- c(quantile_99, avg_greater)
names(results) <- c("99th Quantile", "Sample Avg > 99th q")
results

```

```

      99th Quantile Sample Avg > 99th q
      0.9178628      1.2027965

```

2)

```

library(quantmod)
syb = c("GIS", "KDP", "KO", "PG")
d = length(syb)
rt = c()

for(i in 1:d){
  getSymbols(syb[i], from = "2011-01-01", to = "2024-08-31")
  rt = cbind(rt, weeklyReturn(Ad(get(syb[i])), type = "log")[-1])
}
colnames(rt) = syb
rt = rt*100

```

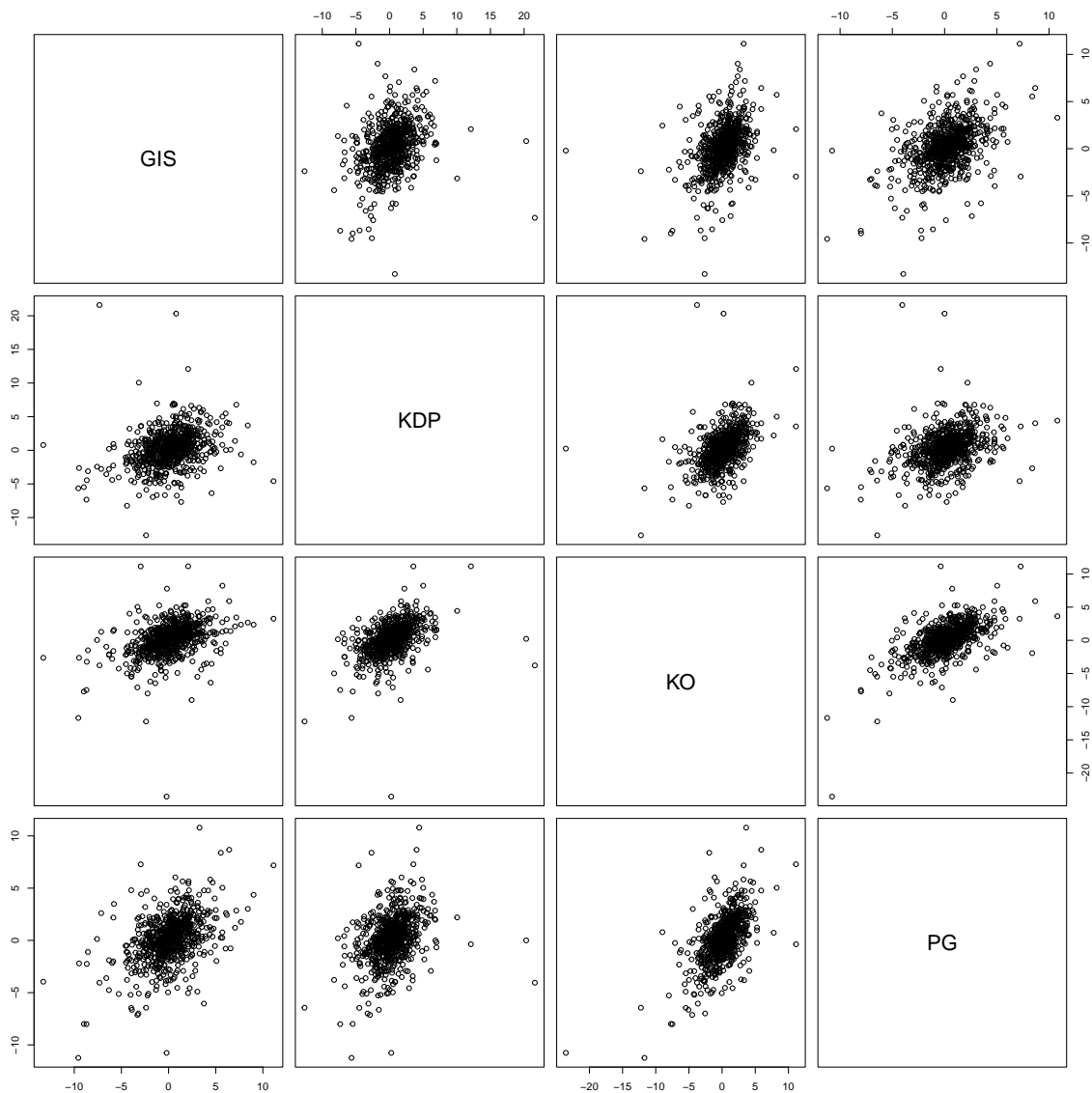
a)

We can create a scatter-plot matrix for these returns.

```

par(mfrow = c(4,4))
pairs(~GIS + KDP + KO + PG, data = rt)

```



The first step of estimating the variance of $.5X_1 + .3X_2 + .2X_3$ is getting the sample covariance matrix

```
Sigma <- cov(rt[,1:3])
Sigma
```

```
      GIS      KDP      KO
GIS 6.345148 2.059470 2.477712
KDP 2.059470 7.658112 3.014969
```

KD 2.477712 3.014969 6.255656

Then we will use the fact that $Var(w^T Y) = w^T Cov(Y)w$, where w is the vector of weights $w = [.5, .3, .2]^T$.

```
w = c(.5,.3,.2)
variance_est <- t(w) %*% Sigma %*% w
variance_est
```

```
      [,1]
[1,] 4.000923
```

The estimated variance is 4.0009231.

b)

The multivariate t distribution requires all marginal distributions to have the same degrees of freedom. We must check that assumption.

```
library(MASS)
nu = c(); se = c();
for(i in 1:4){
  start = list(m = mean(rt[,i]), s = sd(rt[,i]), df = 4)
  fit = fitdistr(rt[,i], "t", start, lower = c(-100, 0.0001, 0.1))
  nu[i] = fit$est["df"]
  se[i] = fit$sd["df"]
}
stat = cbind(nu, se, nu - qnorm(.975)*se, nu + qnorm(.975)*se)
rownames(stat) = syb
colnames(stat) = c("nu", "std err", "lower 95%", "upper 95%")
stat
```

	nu	std err	lower 95%	upper 95%
GIS	4.285302	0.7073373	2.898947	5.671658
KDP	4.223986	0.6811152	2.889025	5.558947
KD	4.087235	0.6134598	2.884875	5.289594
PG	4.100585	0.6627888	2.801543	5.399627

Since each fitted marginal t distribution appears to have similar degrees of freedom the multivariate t distribution is a suitable candidate model for rt . Some other reasons we would use this model are as follows. The tails of return data are often longer and fatter than the normal distribution. Additionally in our scatter plot we can see the tail dependence in the

pairs of return data. Using the multivariate t distribution also has convenient properties for analyzing a portfolio, it can be shown that given a weight w and a multivariate t distribution of returns that the distribution of the portfolio's return is $t_\nu(w^T \mu, w^T \Lambda w)$. Where μ and Λ are the location vector and scale matrix respectively.

c)

```
library(mnormt)
df = seq(2.7, 5.7, 0.01)
n = length(df)
loglik_profile = rep(0,n)
for(i in 1:n){
  fit = cov.trob(rt, nu = df[i])
  mu = as.vector(fit$center)
  sigma = matrix(fit$cov, nrow = 4)
  loglik_profile[i] = sum(log(dmt(rt, mean = fit$center,
                                S = fit$cov, df = df[i])))
}
```

To find the MLE of ν we find where profile likelihood is maximized.

```
df_mle <- df[which.max(loglik_profile)]
df_mle
```

```
[1] 4.24
```

The MLE of ν is 4.24.

We accept the null hypothesis $H_0 : \theta_1 = \theta_{0,1}$ if $L_P(\theta_{0,1}) > L_P(\hat{\theta}_1) - \frac{1}{2}\chi_{\alpha,1}^2$. Where $\hat{\theta}_1$ is the MLE of ν that we previously computed. Then our 90% confidence interval is the range of ν that fail to reject the null hypothesis for each hypothesized value of ν .

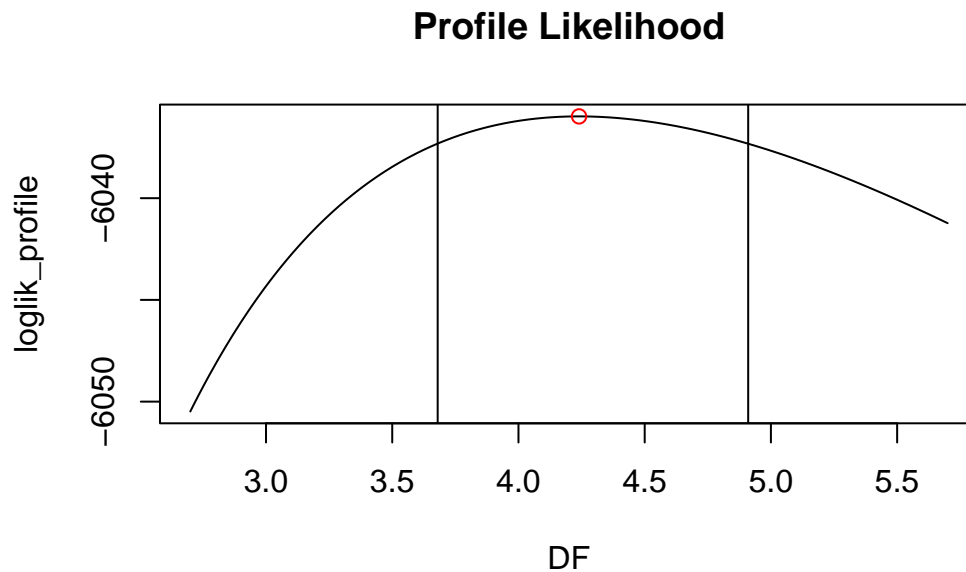
```
critical_value <- max(loglik_profile) - 1/2*qchisq(.90,1)
lower <- min(df[which(loglik_profile > critical_value)])
upper <- max(df[which(loglik_profile > critical_value)])

conf_interval <- c(lower,upper)
names(conf_interval) <- c("lower.90%","upper.90%")
conf_interval
```

```
lower.90% upper.90%
      3.68      4.91
```


The plot of the log-likelihood marking its MLE and confidence interval is below:

```
plot(x = df, y = loglik_profile, main = "Profile Likelihood", xlab = "DF", type = "l")
abline(v = conf_interval[1])
abline(v = conf_interval[2])
points(x = df_mle, y = max(loglik_profile), col = "red")
```



d)

AIC and BIC are defined as:

$$\text{AIC} = -2\log(\hat{\nu}_{ML}) + 2p \quad \text{BIC} = -2\log(\hat{\nu}_{ML}) + p\log(n)$$

```
d = dim(rt)[2]; n = dim(rt)[1]
p = d*(d+1)/2 + d + 1

AIC <- -2*max(loglik_profile) + 2*p
BIC <- -2*max(loglik_profile) + p*log(length(n))
information_criteria <- c(AIC,BIC)
names(information_criteria) <- c("AIC","BIC")
information_criteria
```

	AIC	BIC
	12101.96	12071.96

e)

```
library(sn)
```

```
fit_st = mst.mple(y = rt)
estimated_nu <- fit_st$dp$nu
estimated_nu
```

```
[1] 4.278654
```

```
dp2cp(fit_st$dp, "st")
```

```
$beta
      GIS      KDP      KO      PG
[1,] 0.1576347 0.2721444 0.1728321 0.1760816
```

```
$var.cov
      GIS      KDP      KO      PG
GIS 6.966534 2.826171 2.920285 2.806366
KDP 2.826171 7.702953 3.496993 2.568127
KO  2.920285 3.496993 5.903324 3.376390
PG  2.806366 2.568127 3.376390 5.606491
```

```
$gamma1
      GIS      KDP      KO      PG
-0.3516716 -0.3037892 -1.0232148 -0.7583804
```

```
$gamma2M
gamma2M
185.815
```

Our estimated ν is greater than 4, so we use the `dp2cp()` function to obtain the mean vector, covariance matrix and skewness.

f)

The addition of the skewness parameter increases our p from 15 to 19.

```

p_st = d*(d+1)/2 + d + 1 + d

aic_st = -2*fit_st$logL + 2*p_st
bic_st = -2*fit_st$logL + p_st*log(n)

ic_st <- c(aic_st, bic_st)
names(ic_st) <- c("AIC","BIC")
ic_st

```

	AIC	BIC
	12099.27	12186.06

We can compare the information criteria for the multivariate t distribution and the skewed multivariate t distribution.

```

ic_compare <- rbind(information_criteria, ic_st)
rownames(ic_compare) = c("Multivariate T","Multivariate Skewed T")
ic_compare

```

	AIC	BIC
Multivariate T	12101.96	12071.96
Multivariate Skewed T	12099.27	12186.06

The AIC would choose the multivariate skewed T distribution and BIC would choose the multivariate T distribution.

3)

```
load("HW03.Rdata")
```

```
head(xt,3)
```

	ACM	AXP	CVX	JNJ
2009-04-02	4.285842	3.671404	2.9004135	-0.13208106
2009-04-03	3.308144	2.309577	0.2415228	-1.56010997
2009-04-06	-2.336389	-1.115153	-0.8406822	0.09577359

a)

We want to check that the degrees of freedom for each fitted marginal t distribution are similar to each other.

```
univ.t$mle[, "df"]
```

	ACM	AXP	CVX	JNJ
	4.021471	2.639951	4.090161	4.012284

```
univ.t$se[, "df"]
```

	ACM	AXP	CVX	JNJ
	0.3199585	0.1612412	0.3585920	0.3383137

We can create a confidence interval for each fitted marginal t distributions' degree of freedom.

```
upper_df <- univ.t$mle[, "df"] + 1.96*univ.t$se[, "df"]
lower_df <- univ.t$mle[, "df"] - 1.96*univ.t$se[, "df"]
conf_dfs <- rbind(lower_df, upper_df)
rownames(conf_dfs) <- c("Lower.95%", "Upper.95%")
conf_dfs
```

	ACM	AXP	CVX	JNJ
Lower.95%	3.394352	2.323918	3.387321	3.349189
Upper.95%	4.648590	2.955984	4.793002	4.675379

A multivariate-t model is inappropriate for xt, the degrees of freedom for each marginal t distribution are not identical. Specifically AXP returns' fitted degrees of freedom is far smaller than the other three returns.

b)

We can model ACM, CVX and JNJ with a multivariate-t model since their marginal t distributions have similar degrees of freedom.

```
xt_mvt <- xt[, c(1,3,4)]
```

```
df = seq(3.2, 4.9, 0.01)
n = length(df)
loglik_profile = rep(0, n)
for(i in 1:n){
  fit = cov.trob(xt_mvt, nu = df[i])
  mu = as.vector(fit$center)
  sigma = matrix(fit$cov, nrow = 3)
```

```
loglik_profile[i] = sum(log(dmt(xt_mvt, mean = fit$center,
                               S = fit$cov, df = df[i])))
}
```

```
df_xt_mvt <- df[which.max(loglik_profile)]
df_xt_mvt
```

```
[1] 4.37
```

The degrees of freedom that maximizes the likelihood is 4.37. Now we can fit the model.

```
est = cov.trob(xt_mvt, nu = df_xt_mvt, cor = T)
names(est)
```

```
[1] "cov"      "center" "n.obs"  "cor"     "call"    "iter"
```

c)

```
est$center
```

```
      ACM      CVX      JNJ
0.04700037 0.05358264 0.06739520
```

```
diag(est$cov)
```

```
      ACM      CVX      JNJ
1.9843306 1.0402260 0.4834828
```

The marginal distribution for each return is as follows: ACM is $t_5(0.04700037, 1.9843306)$, CVX is $t_5(0.05358264, 1.0402260)$ and JNJ is $t_5(0.06739520, 0.4834828)$.

d)

The hypothesis we are testing is $H_0 : \nu = 4, H_a : \nu \neq 4$. Our test statistic is $2(\log L(\hat{\nu}_{ML}) - \log L(\hat{\nu}_{0,ML})) \geq \chi_{\alpha,1}$.

```

log_lik_at_mle <- max(loglik_profile)
log_lik_at_4 <- loglik_profile[which(df==4)]
test_statistic <- 2*(log_lik_at_mle - log_lik_at_4)
p_value <- 1 - pchisq(test_statistic, df = 1)
results <- c(test_statistic, p_value)
names(results) <- c("Test Statistic", "P-Value")
results

```

Test Statistic	P-Value
2.85799171	0.09092095

With a P-value of 0.091 at a significance level of .05 we cannot reject the null hypothesis of ν being equal to 4.