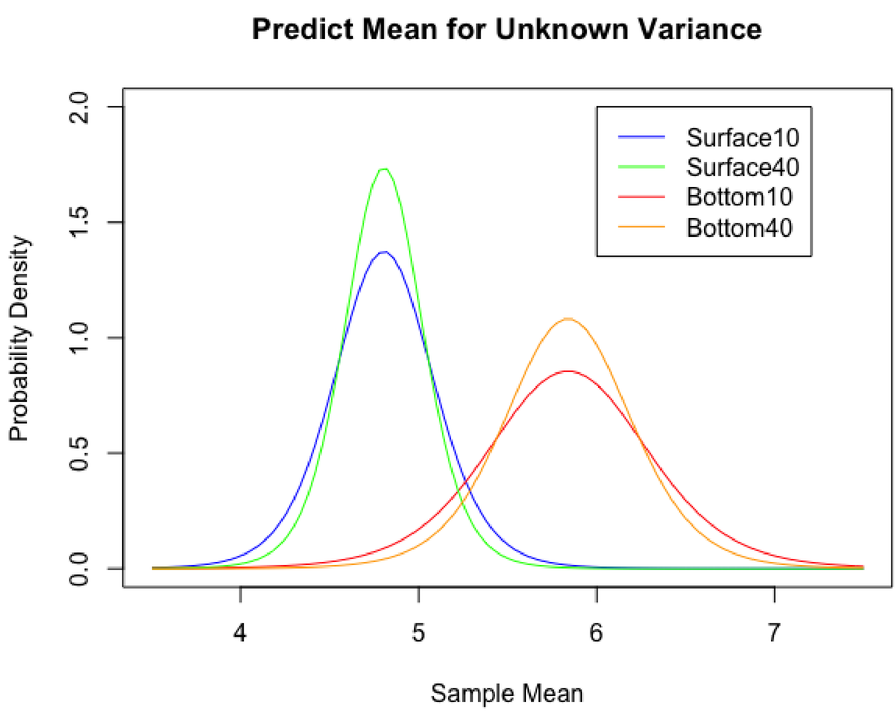
Problem 1:
The predictive distributions are T-distributions. The means are largely separated for a future sample of 10 from the surface and bottom of the river, but not entirely. Increasing the future sample size further contracts the distributions, as shown in the visualization below, and the credible intervals.

	Surface		**Bottom**	
Predictive Sample	**10**	**40**	**10**	**40**
Mean	4.804	4.804	5.839	5.839
Spread	0.282	0.223	0.453	0.358
DoF	9	9	9	9
Credible Int 0.025	4.165	4.299	4.813	5.028
Credible Int 0.975	5.443	5.309	6.865	6.650



CSI-674 Jericho McLeod
Assignment 7 G00986513

Code used for Problem 1:

```
wolf.surface=c(3.74, 4.61, 4.00, 4.67, 4.87, 5.12, 4.52, 5.29, 5.74,
5.48)
wolf.bottom=c(5.44, 6.88, 5.37, 5.44, 5.03, 6.48, 3.89, 5.85, 6.85,
7.16)

# Sufficient statistics for mean
xs=wolf.surface
sums=sum(wolf.surface)
xb=wolf.bottom
sumb=sum(wolf.bottom)
n <- length(xs)

qqnorm(wolf.surface)
qqline(wolf.surface)
qqnorm(wolf.bottom)
qqline(wolf.bottom)

mus=0          # prior mean is 0
ks=0           # prior precision multiplier is 0
alphas=-0.5    # Shape of Jeffreys prior for precision
betas=Inf      # Infinite scale for Jeffreys prior
mub=0          # prior mean is 0
kb=0           # prior precision multiplier is 0
alphab=-0.5    # Shape of Jeffreys prior for precision
betab=Inf      # Infinite scale for Jeffreys prior


# Posterior hyperparameters
mus.star=(ks*mus+sum(xs))/(ks+n)
ks.star=ks+n
alphas.star=alphas+n/2
betas.star=( 1/betas + 0.5*sum((xs-mean(xs))^2) +
               0.5*ks*n/(ks+n)*(mean(xs)-mus)^2 )^-1
mub.star=(kb*mub+sum(xb))/(kb+n)
kb.star=kb+n
alphab.star=alphab+n/2
betab.star=( 1/betab + 0.5*sum((xb-mean(xb))^2) +
               0.5*kb*n/(kb+n)*(mean(xb)-mub)^2 )^-1

# Find Marginal likelihoods for sample mean of size 10 for S and B.
n_pred   <- 10
s_center <- mus.star
s_spread <- 1 / sqrt(
((ks.star*n_pred)/(ks.star+n_pred))*alphas.star*betas.star )
s_degf   <- 2*alphas.star
b_center <- mub.star
b_spread <- 1 / sqrt(
((kb.star*n_pred)/(kb.star+n_pred))*alphab.star*betab.star)
b_degf   <- 2*alphab.star
```

```
# Plot the distribution
xbar=seq(length=101,from=3.5,to=7.5)
s_pred=dt((xbar-s_center)/s_spread,df=s_degf)/s_spread
b_pred=dt((xbar-b_center)/b_spread,df=b_degf)/b_spread
plot(xbar,s_pred,type='l',col="blue",
     main=paste("Predict Mean for 10 Obs- Unknown Variance"),
     xlab="Sample Mean",
     ylab="Probability Density")
lines(xbar,b_pred,col='green')

# Find 95% Credible Intervals
s_xbar025=mus.star+qt(0.025,2*alphas.star)/sqrt((ks.star*n_pred/(ks.st
ar+n_pred))*alphas.star*betas.star)  # 0.025 quantile for mean
s_xbar975=mus.star+qt(0.975,2*alphas.star)/sqrt((ks.star*n_pred/(ks.st
ar+n_pred))*alphas.star*betas.star)  # 0.975 quantile for mean
b_xbar025=mub.star+qt(0.025,2*alphab.star)/sqrt((kb.star*n_pred/(kb.st
ar+n_pred))*alphab.star*betab.star)  # 0.025 quantile for mean
b_xbar975=mub.star+qt(0.975,2*alphab.star)/sqrt((kb.star*n_pred/(kb.st
ar+n_pred))*alphab.star*betab.star)  # 0.975 quantile for mean

print('mean,spread,dgef: surface, then bottom')
print(c(s_center,s_spread,s_degf))
print(c(b_center,b_spread,b_degf))
print('surface CI, then bottom CI')
print(c(s_xbar025,s_xbar975))
print(c(b_xbar025,b_xbar975))

# Repeat for future sample size of 40 and compare
n_pred40   <- 40
s_center40 <- mus.star
s_spread40 <- 1 / sqrt(
((ks.star*n_pred40)/(ks.star+n_pred40))*alphas.star*betas.star )
s_degf40   <- 2*alphas.star
b_center40 <- mub.star
b_spread40 <- 1 / sqrt(
((kb.star*n_pred40)/(kb.star+n_pred40))*alphab.star*betab.star)
b_degf40   <- 2*alphab.star

# Plot the distribution
xbar=seq(length=101,from=3.5,to=7.5)
s_pred40=dt((xbar-s_center40)/s_spread40,df=s_degf40)/s_spread40
b_pred40=dt((xbar-b_center40)/b_spread40,df=b_degf40)/b_spread40
plot(xbar,s_pred40,type='l',col="blue",
     main=paste("Predict Mean for 40 Obs- Unknown Variance"),
     xlab="Sample Mean",
     ylab="Probability Density")
lines(xbar,b_pred40,col='green')

s_xbar025_40=mus.star+qt(0.025,2*alphas.star)/sqrt((ks.star*n_pred40/(
ks.star+n_pred40))*alphas.star*betas.star)  # 0.025 quantile for mean
```

```
s_xbar975_40=mus.star+qt(0.975,2*alphas.star)/sqrt((ks.star*n_pred40/(
ks.star+n_pred40))*alphas.star*betas.star)  # 0.975 quantile for mean
b_xbar025_40=mub.star+qt(0.025,2*alphab.star)/sqrt((kb.star*n_pred40/(
kb.star+n_pred40))*alphab.star*betab.star)  # 0.025 quantile for mean
b_xbar975_40=mub.star+qt(0.975,2*alphab.star)/sqrt((kb.star*n_pred40/(
kb.star+n_pred40))*alphab.star*betab.star)  # 0.975 quantile for mean


print('mean,spread,dgef: surface, then bottom')
print(c(s_center40,s_spread40,s_degf40))
print(c(b_center40,b_spread40,b_degf40))
print('surface CI, then bottom CI')
print(c(s_xbar025_40,s_xbar975_40))
print(c(b_xbar025_40,b_xbar975_40))


plot(xbar,s_pred,type='l',col="blue",
     main=paste("Predict Mean for Unknown Variance"),
     xlab="Sample Mean",
     ylab="Probability Density",
     ylim=c(0,2),
     xlim=c(3.5,7.5))
lines(xbar,b_pred,col='red')
lines(xbar,s_pred40,col="green")
lines(xbar,b_pred40,col='orange')
legend(6,2,c('Surface10','Surface40','Bottom10','Bottom40'),
       col=c("blue","green",'red','orange'),lty=c(1,1,1,1))
```

CSI-674 Jericho McLeod
Assignment 7 G00986513

Problem 2:

The resulting estimate for the predictive distribution of differences between means are for normal distributions with parameters:

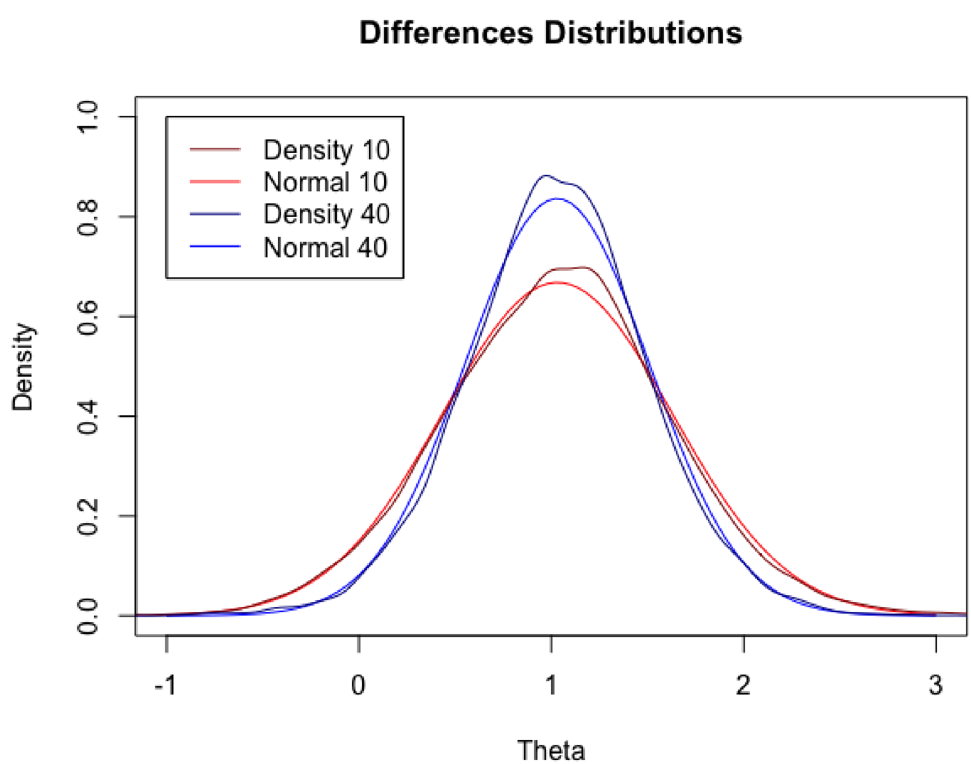
Differences	SD	Mean
10 Samples	0.597	1.032
40 Samples	0.477	1.030

The standard deviations shrink notably, while the change in mean is merely due to the small number of samples in the direct Monte Carlo; as the samples approach infinity the difference in means approaches 0.

The credible intervals are shown in the following table. Given that the standard deviation above is smaller for the larger predictive sample we would expect, and indeed see, a narrower credible interval for the larger future sample.

(Bottom – Top) Credible Intervals	0.025	0.975
Predictive Sample 10	-0.1490	2.2233
Predictive Sample 40	0.0867	1.9677

The density of the Monte Carlo simulation is visualized in the chart below for easy comparison.



Code used for Problem 2:

```
numSim <-10000

# Simulate using the random T distribution
s_pred10<-
mus.star+rt(numSim,2*alphas.star)/sqrt((ks.star*n_pred/(ks.star+n_pred
))*alphas.star*betas.star)
b_pred10<-
mub.star+rt(numSim,2*alphab.star)/sqrt((kb.star*n_pred/(kb.star+n_pred
))*alphab.star*betab.star)
s_pred40<-
mus.star+rt(numSim,2*alphas.star)/sqrt((ks.star*n_pred40/(ks.star+n_pr
ed40))*alphas.star*betas.star)
b_pred40<-
mub.star+rt(numSim,2*alphab.star)/sqrt((kb.star*n_pred40/(kb.star+n_pr
ed40))*alphab.star*betab.star)

# Find the differences
diff10 = b_pred10-s_pred10
diff40 = b_pred40-s_pred40


# Plot the density of the MC to check that it makes sense
plot(density(s_pred40),
     col='lightblue',
     xlab='Theta',
     main='Density of Estimates of Means',
     xlim=c(3,8))
lines(density(b_pred40),col='green')
lines(density(b_pred10),col='darkgreen')
lines(density(s_pred10),col='darkblue')
legend(6.5,1.7,c("Surface 40","Bottom 40", "Surface 10", "Bottom 10"),

col=c("lightblue","green","darkblue","darkgreen"),lty=c(1,1,1,1))

# Plot the density of the differences
plot(density(diff10),col='darkred',ylim=c(0,1),xlim=c(-
1,3),xlab='Theta',main='Density of Differences of Means')
lines(density(diff40),col='red')
legend(-1,1,c("Difference 10","Difference 40"),
       col=c("darkred","red"),lty=c(1,1))

# Examine the distributions of the differences
diff10_std <- sd(diff10)
diff40_std <- sd(diff40)
diff10_mean <- mean(diff10)
diff40_mean <- mean(diff40)

# plot density with normal distributions for comparison
thetas <- seq(-1,3,length=100)
```

```
diff10_dens <- dnorm(thetas,diff10_mean,diff10_std)
diff40_dens <- dnorm(thetas,diff40_mean,diff40_std)
plot(thetas,diff10_dens,type='l',col='red',ylim=c(0,1),xlab="Theta",ma
in='Differences Distributions',ylab='Density')
lines(thetas,diff40_dens,col='blue')
lines(density(diff10),col='darkred')
lines(density(diff40),col='purple')
legend(-1,1,c("Density 10","Normal 10","Density 40","Normal 40"),
       col=c("darkred","red","purple","blue"),lty=c(1,1,1,1))

# Print Results
print(c(diff10_std,diff10_mean))
print(c(diff40_std,diff40_mean))


# Find the quantiles representing the Credible Interval
quantile(diff10,c(0.025,0.975))
quantile(diff40,c(0.025,0.975))
g10 <- sum(b_pred10>s_pred10)/numSim
g40 <- sum(b_pred40>s_pred40)/numSim
print(c(g10,g40))
```

Problem 3:

Repeating Problem 1 with known population variance gives a normal predictive distribution, the parameters for which are shown in the chart below.

	Surface 10	Surface 40	Bottom 10	Bottom 40
SD	0.2823	0.2232	0.4535	0.3585
Mean	4.804	4.804	5.839	5.839

Comparing this to the case where the variance was unknown shows the effect of the additional uncertainty included in the methods for predictive distributions with unknown variance; the quantiles are expanded further from the means.

		0.025 Quantile	0.975 Quantile
Surface 10	**Unknown Var**	4.165	5.443
	Known Var	4.251	5.357
Surface 40	**Unknown Var**	4.299	5.309
	Known Var	4.366	5.242
Bottom 10	**Unknown Var**	4.813	6.865
	Known Var	4.950	6.728
Bottom 40	**Unknown Var**	5.028	6.650
	Known Var	5.136	6.542

Code used for Problem 3:

```
n = length(wolf.surface)
s_xbar = mean(wolf.surface)
b_xbar = mean(wolf.bottom)

# Prior hyperparameters - noninformative reference prior
s_mu=0          # prior mean is 0
s_tau=Inf       # prior standard deviation is infinity
s_sigma=sd(wolf.surface)  # treat standard deviation as known and
equal to sample standard deviation
b_mu=0          # prior mean is 0
b_tau=Inf       # prior standard deviation is infinity
b_sigma=sd(wolf.bottom)  # treat standard deviation as known and equal
to sample standard deviation

# Posterior distribution for reaction times for subject is normal
s_mu.star =
(s_mu/s_tau^2+sum(wolf.surface)/s_sigma^2)/(1/s_tau^2+n/s_sigma^2)  #
Posterior mean
s_tau.star = (1/(s_tau^2) + n/(s_sigma^2))^(-1/2)              #
Posterior std. dev

b_mu.star =
(b_mu/b_tau^2+sum(wolf.bottom)/b_sigma^2)/(1/b_tau^2+n/b_sigma^2)  #
Posterior mean
b_tau.star = (1/(b_tau^2) + n/(b_sigma^2))^(-1/2)              #
Posterior std. dev


# Predictive distributions
n_samples <- 10
s10_std <- sqrt((s_sigma^2/n_samples) + s_tau.star^2)
b10_std <- sqrt((b_sigma^2/n_samples) + b_tau.star^2)
n_samples <- 40
s40_std <- sqrt((s_sigma^2/n_samples) + s_tau.star^2)
b40_std <- sqrt((b_sigma^2/n_samples) + b_tau.star^2)

thetas <- seq(length=101,from=3.5,to=7.5)

s10_dens <- dnorm(thetas,s_mu.star,s10_std)
b10_dens <- dnorm(thetas,b_mu.star,b10_std)
plot(thetas,s10_dens,type='l')
lines(thetas,b10_dens)

s10_q <- qnorm(c(0.025,0.975),s_mu.star,s10_std)
s40_q <- qnorm(c(0.025,0.975),s_mu.star,s40_std)

b10_q <- qnorm(c(0.025,0.975),b_mu.star,b10_std)
b40_q <- qnorm(c(0.025,0.975),b_mu.star,b40_std)
print(b10_q)
```