Due: Monday, November 27 by 11:59pm Pacific (late assignments not accepted, solutions to be posted shortly thereafter for study purposes).

Instructions: Upload a well organized pdf/html document outputted from RMarkdown to Canvas. A barebones template is attached on Canvas and Github.

1. "The fire consumption data set serves as basis for predictive models of wildfire behaviour used by the Canadian Forest Service (cf. Van Wagner et al. (1992), Han and Braun (2014)). It is comprised from several documented prescribed burns and experimental fires (e.g., Muraro (1975), Stocks (1987b), Stocks (1987a), Stocks (1989), Quintilio et al. (1977)). Variables retained for this analysis are windspeed (WIND, in kilometres per hour), fine fuel moisture code (FFMC, a measure of moisture content — larger is dryer), initial spread index (ISI, a numeric representation of expected rate of spread), rate of spread (ROS, the observed rate of spread index), and buildup index (BUI, a numeric representation of the total amount of burnables available). The grouping variable of interest is the fire type, which includes two categories: surface fire and crown fire. Surface fires spread at the ground level, while crown fires spread through the top foliage layers of trees. After omitting rows with missing values, 195 observed fires remain for the aforementioned retained variables." [1]

   We now fit two classification models and report both the training classification performance and the performance as measured by LOOCV in the following tables.

k-nearest neighbours

	Full Training		LOOCV	
	Pred Crown	Pred Surface	Pred Crown	Pred Surface
Crown	75	11	66	20
Surface	4	105	20	89

Classification Tree

	Full Training		LOOCV	
	Pred Crown	Pred Surface	Pred Crown	Pred Surface
Crown	73	13	72	14
Surface	7	102	13	96

   Suppose you're part of a data science firefighting team tasked with providing predicted fire behaviour to firefighters en route to new fires. Which of these two models would you use for this problem? Why?

2. Many introductory statistics classes cover confidence intervals based on estimators such as the sample mean ($\bar{X}$) and sample variance S^2 under specific conditions (usually cases where X is normally distributed or the sample size n is sufficiently large for the Central Limit Theorem to hold). Specifically in the case of $s^2 = \frac{\sum(x_i - \bar{x})^2}{n-1}$ with normally distributed X, the confidence interval for variance σ^2 is computed as:

$$\left(\frac{(n-1)s^2}{\chi^2_{\alpha/2}}, \frac{(n-1)s^2}{\chi^2_{1-\alpha/2}} \right)$$

[1] From Andrews, J. L. (2018). 'Addressing overfitting and underfitting in Gaussian model-based clustering'. *Computational Statistics & Data Analysis*, 127, 160-171:

In the following code, we generate X from a continuous uniform with [0,1] bounds, and then compute confidence intervals for σ^2 based on an incorrect normality assumption, and bootstrap CI's via Efron intervals.

```
> set.seed(2023)
> library(bootstrap)
> norm_var_ci <- boot_var_ci <- matrix(NA, nrow=1000, ncol=2)
> for(i in 1:1000){
+   dumx <- runif(25)
+   norm_var_ci[i, 1] <- (25-1)*var(dumx)/qchisq(0.975, 25-1)
+   norm_var_ci[i, 2] <- (25-1)*var(dumx)/qchisq(0.025, 25-1)
+   dumboot <- bootstrap(dumx, 1000, var)
+   boot_var_ci[i, 1] <- quantile(dumboot$thetastar, 0.025)
+   boot_var_ci[i, 2] <- quantile(dumboot$thetastar, 0.975)
+ }
> contain_var_norm <- contain_var_boot <- rep(NA, 1000)
> for(i in 1:1000){
+   contain_var_norm[i] <- norm_var_ci[i, 1] <= 1/12 & 1/12 <= norm_var_ci[i, 2]
+   contain_var_boot[i] <- boot_var_ci[i, 1] <= 1/12 & 1/12 <= boot_var_ci[i, 2]
+ }
> sum(contain_var_norm)
[1] 997
> sum(contain_var_boot)
[1] 941
```

Explain what those last two printed numbers represent. Compute the observed confidence level of the 95% confidence interval that is based on normality. Compute the observed confidence level of the 95% bootstrap-based confidence interval. If the observed confidence level of the normality-based confidence interval is larger, should we choose that way over the bootstrap in this scenario (that is, when we have X uniformly distributed)? Explain.

3. Now, let's consider another scenario. Suppose we know our sample X arises from a continuous uniform with [0, θ] bounds, and we wish to estimate θ using the sample maximum value $X_{(n)}$ (which is the MLE). Theory suggests that a 95% CI is $(X_{(n)}, \frac{X_{(n)}}{.05^{1/n}})$. Adjust the code from the previous question to carry out an investigation for how the (correct) theory and a bootstrap CI approach using sample maximum behave when the data is generated from a continuous uniform with [0, 1] bounds. Note that the raw R code is uploaded in Canvas for ease of copy-pasting.
   Comment on the results. What nuance (briefly discussed in the final section of Lab 1) about the bootstrap are we seeing?

4. Let's shed some light on a common pitfall of modern data analysis! Suppose we have some low sample size (50), high dimensional (2000) data and expect to fit a linear model to Y...

```
set.seed(32531)
X <- matrix(rnorm(100000), nrow=50, ncol=2000)
Y <- rnorm(50)
```

Recall from DATA 570 that we have a problem fitting linear regression here since $p > n$. So, let's say we pre-screen our variables by choosing the top 10 most correlated X variables to Y, and then estimate the long-run MSE with LOOCV:

```
top10 <- which(rank(cor(cbind(Y, X))[-1,1]) %in% 1:10)
cvlm <- list()
msecvW <- NA
Xtop <- data.frame(X[, top10])
for(i in 1:nrow(X)){
  cvY <- Y[-i]
  cvlm[[i]] <- lm(cvY ~., data=Xtop[-i,])
  msecvW[i] <- (predict(cvlm[[i]], newdata=Xtop[i,]) - Y[i])^2
}
mean(msecvW)
```

(a) Run the above code and report the LOOCV estimate of the MSE.

(b) Consider the initial generation of X and Y. What is the true MSE of the generative model? Are you concerned about the LOOCV estimate? Explain.

(c) Let's do one very small change: move the variable screening INSIDE the LOOCV process.

```
cvlm <- list()
msecvR <- NA
for(i in 1:nrow(X)){
  cvtop10 <- which(rank(cor(cbind(Y[-i], X[-i,]))[-1,1]) %in% 1:10)
  cvXtop10 <- data.frame(X[, cvtop10])
  cvY <- Y[-i]
  cvlm[[i]] <- lm(cvY ~., data=cvXtop10[-i,])
  msecvR[i] <- (predict(cvlm[[i]], newdata=cvXtop10[i,]) - Y[i])^2
}
mean(msecvR)
```

Run the above code, report the LOOCV estimate of the MSE, and contrast with your previous result.