Lab 7

DO NOT INCLUDE NAMES - Just add names in gradescope

# Rules

In groups of 2 or 3, complete the following.

# Modeling Snow Density

We will be using the data set from Wetlaufer, Hendrikx, and Marshall (2016) - study where they explored the relationship between snow density () or snow depth (snow, mm) or snow water equivalent (SWE, mm) with a suite of predictor variables. In this lab we will condition on there being snow present and then try to model snow density with models that will compete with what they discuss in Tables 3 and 4. We will be interested in using elev (Elevation, m), Land (forest cover with 0 = unforested and 10 = forested), rad (Potential Solar radiation, ), curvature (see <https://blogs.esri.com/esri/arcgis/2010/10/27/understanding-curvature-rasters/> for a description), aspect (orientation of slope in degrees (0 to 360 degrees)), and angle (angle of slope in degrees with 0 being flat) as predictors. Also pay attention to the strata variable (read its definition in the paper) and the role that played in the data collection and should in the analysis, as we will revisit this.

* Wetlaufer, K., Hendrikx, J., and L. Marshall (2016) Spatial Heterogeneity of Snow Density and Its Influence on Snow Water Equivalence Estimates in a Large Mountainous Basin. *Hydrology*, 3(1):3, <doi:10.3390/hydrology3010003>. Available at <http://www.mdpi.com/2306-5338/3/1/3/htm> and on D2L

Run the following code to get re-started with the data set.

data(snowdepths)  
snowdepths <- snowdepths %>%  
 mutate(AspectCat = factor(case\_when(  
 aspect %in% (0:45)~ "North",  
 aspect %in% (315:360)~ "North",  
 aspect %in% 45:(90+45) ~ "East",  
 aspect %in% (90+45):(180+45) ~ "South",  
 aspect %in% (180+45):315 ~ "West"  
 )),  
 SnowPresence = factor(case\_when(  
 snow == 0 ~ "None",  
 snow > 0 ~ "Some"  
 )),  
 Landf = factor(cover),  
 Landf = fct\_recode(Landf,  
 "Not Forested" = "0",  
 "Forested" = "10")  
 )  
  
snowdepthsR <- snowdepths %>% drop\_na(density)  
lmA <- lm(density ~ rad + snow + elev + AspectCat, data = snowdepthsR)

**Q1 (Lab 5 Q6) We should assess model assumptions before we report any inferences from a model. Discuss the potential for a violation of the independence assumption in the previous model for snow density based on the design of the study and data collection. Hint: review the paper (Sections 2 and 3) for how they decided on their sampling locations.**

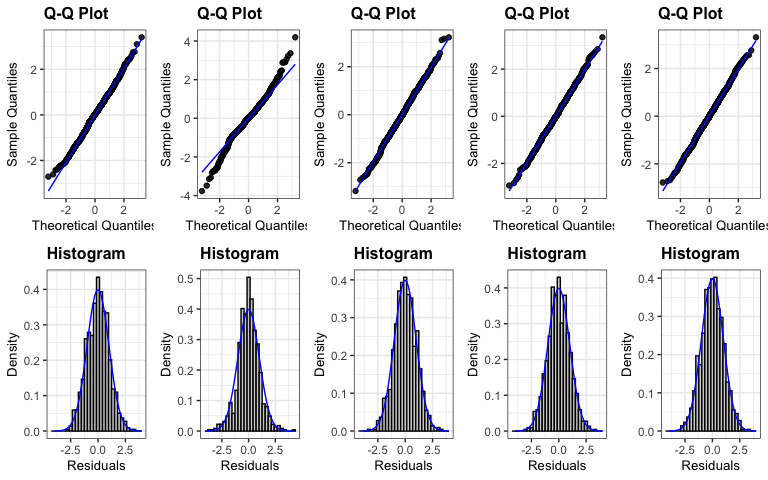
Based on their sampling and study design there are possible violations against the independence assumption. Sampling was conducted in a few distinct areas of West Fork Basin that were selected for their characteristics of different terrain types and defined as physiographic strata. Within these areas, sampling points were randomly selected at distances ranging from 30 to 400m. At each of these sampling points, three samples were taken 10m apart in triangular fashion. This cluster sampling may introduce violations to the assumption of independence as these points may show similar values and not be independent of each other.

**Q2 (Lab 5 Q7) Run the following code on your model to generate a calibration set of plots for the residuals for assessing the normality of residuals assumption. (a) Which column contained the real residuals? (b) How easy was it to identify the real residuals versus versions of the residuals simulated from normal distributions (all the others)? (c) Use how clear it was to identify the real residuals from the simulated ones to discuss the strength of evidence against the normality of residuals assumption. Make sure you discuss the shape of the residual distribution if you think it doesn’t follow a normal distribution.**

The real residuals are seen in column 2. It was somewhat easy to identify the real residuals since they display a heavy-tailed distribution that is not present in the other 4 simulations. Based on this difference we would say that there is strong evidence against the assumption of normality.

set.seed(123)  
resid\_calibrate(lmA, nsim = 4, c("qq","hist"), identify = T, shuffle = T, coordfix = F)

## [1] "Real residuals are in column 2"

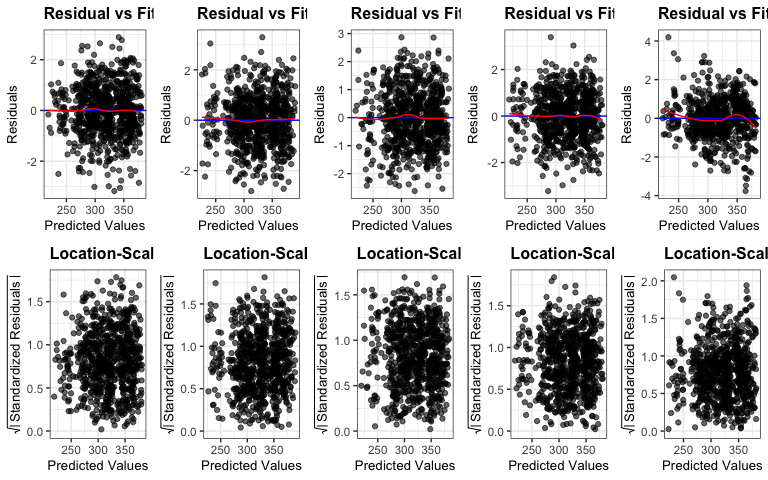


**Q3 (Lab 5 Q8) There is no clear issue with linearity in the residuals vs fitted (more later on that with partial residuals). So we are safe to consider the residuals vs fitted and location-scale plots for assessing the equal variance of residuals assumption. Use the following resid\_calibrate code on your model to explore the equal variance assumption. Write a sentence or two to discuss the strength of evidence against the assumption of equal variance based on these results. When it is safe to consider, you should always discuss the Location-Scale plot along with the Residuals vs Fitted when assessing HOV.**

We would say that there is little evidence against the assumption of equal variance as the residuals vs fitted and location-scale plots are difficult to identify from the simulated normal distributions. As the predicted values increase, the spread of the residuals seem to remain somewhat the same.

set.seed(12345)  
resid\_calibrate(lmA, c("resid", "ls"), nsim = 4, identify = T, shuffle = T)

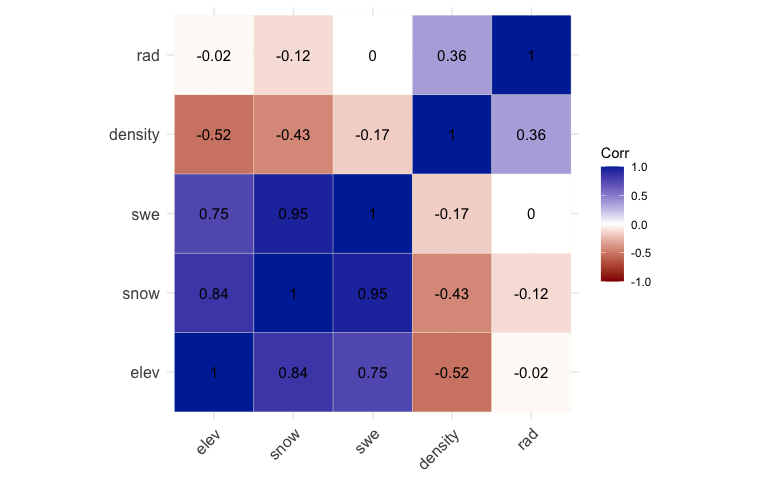
## [1] "Real residuals are in column 5"



**Q4 (Lab 5 Q9) Two more predictors are added to the model (cover (whether location had trees or not) and swe (snow water equivalent)). Check for issues with multi-collinearity in lmA using the vif function (from car) and the check\_collinearity function (from performance). Which two variables are most impacted by multi-collinearity in lmB? How can you explain that?**

Snow Water Equivalent (SWE) and Snow Depth variables are most impacted by multi-collinearity. They have a high correlation coefficient of 0.95. SWE measures the amount of water contained in the snowpack by representing the depth of water that would result if the snow were melted. It makes sense that this variable is highly correlated with snow depth as depth is used to calculate swe.

ggcorrplot(snowdepthsR %>% dplyr::select(density, rad, snow, elev, swe) %>% cor(), lab = T)



lmB <- lm(density ~ rad + snow + elev + AspectCat + cover + swe, data = snowdepthsR)  
vif(lmB)

## GVIF Df GVIF^(1/(2\*Df))  
## rad 2.164159 1 1.471108  
## snow 20.715254 1 4.551401  
## elev 4.136097 1 2.033740  
## AspectCat 2.350019 3 1.153042  
## cover 1.091445 1 1.044722  
## swe 13.970419 1 3.737702

library(performance)  
check\_collinearity(lmB)

## # Check for Multicollinearity  
##   
## Low Correlation  
##   
## Term VIF VIF 95% CI Increased SE Tolerance Tolerance 95% CI  
## rad 2.16 [ 1.96, 2.42] 1.47 0.46 [0.41, 0.51]  
## elev 4.14 [ 3.67, 4.68] 2.03 0.24 [0.21, 0.27]  
## AspectCat 2.35 [ 2.12, 2.63] 1.53 0.43 [0.38, 0.47]  
## cover 1.09 [ 1.04, 1.22] 1.04 0.92 [0.82, 0.96]  
##   
## High Correlation  
##   
## Term VIF VIF 95% CI Increased SE Tolerance Tolerance 95% CI  
## snow 20.72 [18.13, 23.69] 4.55 0.05 [0.04, 0.06]  
## swe 13.97 [12.25, 15.96] 3.74 0.07 [0.06, 0.08]

**Q5 (Lab 5 Q10) How can you explain the following change in slope coefficients for snow and swe across the following three models?**

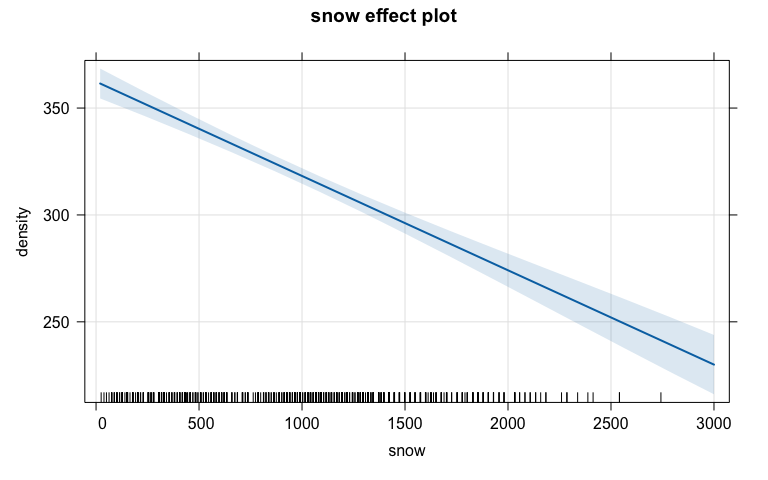
In the models with only snow depth or swe as the only predictor variables, slope coefficients were -0.044(depth) and -0.052(swe). These negative slope values suggest that as snow depth or swe increased, density decreased based on one predictor.

In the additive model, slope coefficient for depth was -0.285614 and 0.767854 for swe. The negative coefficient for snow is now larger because it reflects the independent relationship of snow with density, after removing the influence of SWE. SWE now seems to have a positive relationship with density in the presence of snow. This is due to multicollinearity and their shared variance.

lmDepth <- lm(density ~ snow, data = snowdepthsR)  
summary(lmDepth)

##   
## Call:  
## lm(formula = density ~ snow, data = snowdepthsR)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -146.886 -35.635 5.739 34.281 152.344   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 362.365906 3.618679 100.14 <2e-16  
## snow -0.044128 0.003322 -13.29 <2e-16  
##   
## Residual standard error: 51.01 on 791 degrees of freedom  
## Multiple R-squared: 0.1824, Adjusted R-squared: 0.1814   
## F-statistic: 176.5 on 1 and 791 DF, p-value: < 2.2e-16

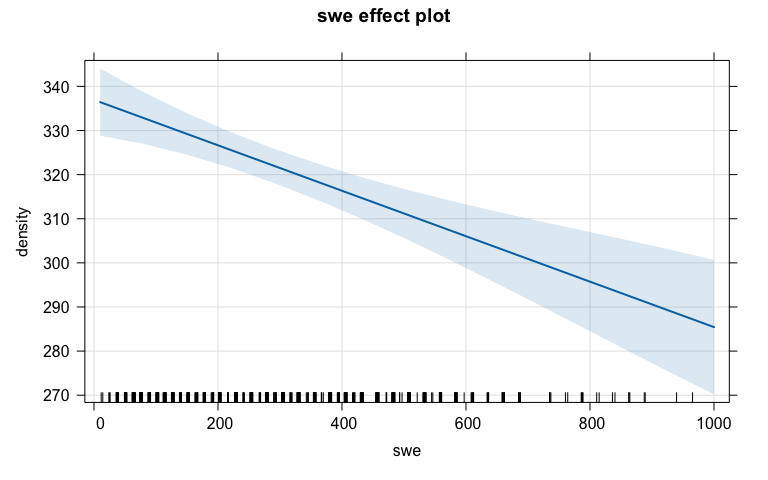
plot(allEffects(lmDepth), grid = T)



lmSWE <- lm(density ~ swe, data = snowdepthsR)  
summary(lmSWE)

##   
## Call:  
## lm(formula = density ~ swe, data = snowdepthsR)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -153.782 -40.697 4.303 43.760 141.556   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 336.95121 3.96880 84.900 < 2e-16  
## swe -0.05151 0.01094 -4.708 2.96e-06  
##   
## Residual standard error: 55.64 on 791 degrees of freedom  
## Multiple R-squared: 0.02725, Adjusted R-squared: 0.02602   
## F-statistic: 22.16 on 1 and 791 DF, p-value: 2.958e-06

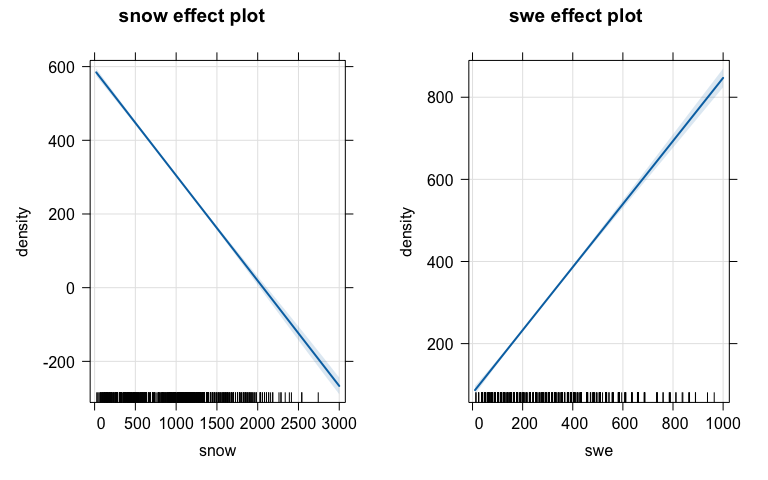
plot(allEffects(lmSWE), grid = T)



lmC <- lm(density ~ snow + swe, data = snowdepthsR)  
summary(lmC)

##   
## Call:  
## lm(formula = density ~ snow + swe, data = snowdepthsR)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -99.772 -11.555 -3.861 14.672 109.771   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 348.591457 1.925461 181.04 <2e-16  
## snow -0.285614 0.005582 -51.17 <2e-16  
## swe 0.767854 0.016858 45.55 <2e-16  
##   
## Residual standard error: 26.81 on 790 degrees of freedom  
## Multiple R-squared: 0.7745, Adjusted R-squared: 0.774   
## F-statistic: 1357 on 2 and 790 DF, p-value: < 2.2e-16

plot(allEffects(lmC), grid = T)



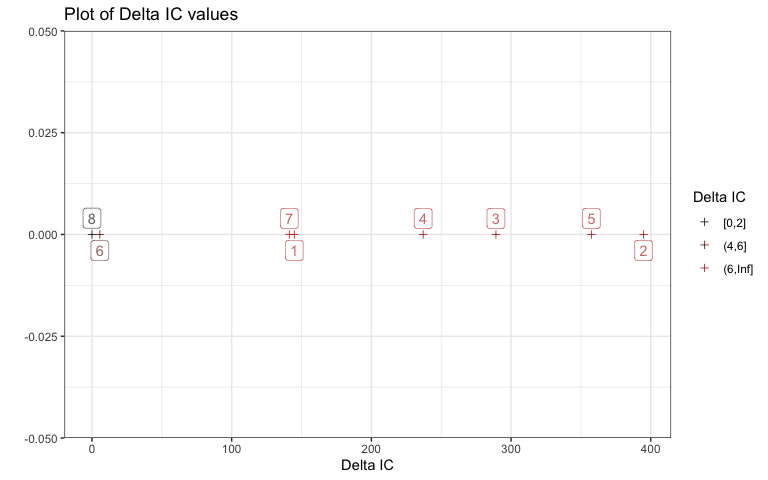
**Q6) The authors note six different models they considered in Tables 3 and 4. Fit those models and use model.sel to compare them on AIC, also including an additive model that contains elevation (elev), radiation (rad), snow depth (snow), slope (angle), and our AspectCat variable, as well as a mean-only model. Report what is in the top AIC model from this suite of eight models.**

The top AIC model is the additive model containing elevation (elev), radiation (rad), snow depth (snow), slope (angle), and our AspectCat variable.

lmElev <- lm(density ~ elev, data = snowdepthsR)  
  
lmmeanonly <- lm(density ~ 1, data = snowdepthsR)  
  
lmRad <- lm(density ~ rad, data = snowdepthsR)  
  
lmSnoDep <- lm(density ~ snow, data = snowdepthsR)  
  
lmSlopeAngle <- lm(density ~ angle, data = snowdepthsR)   
  
lmElevRad <- lm(density ~ elev + rad, data = snowdepthsR)  
  
lmSDRad <- lm(density ~ snow + rad, data = snowdepthsR)  
  
lmADD <- (lm(density ~ elev + rad + snow + angle + AspectCat,  
 data = snowdepthsR))  
  
d1 <- model.sel(list(lmElev, lmmeanonly, lmRad, lmSnoDep,  
 lmSlopeAngle, lmElevRad, lmSDRad, lmADD),  
 rank = "AIC")  
d1

## Model selection table   
## (Intrc) elev rad snow angle AspcC df logLik AIC delta  
## 8 528.5 -0.1284 0.0004326 0.01325 -0.314 + 9 -4117.800 8253.6 0.00  
## 6 479.5 -0.1104 0.0004878 4 -4125.604 8259.2 5.61  
## 7 259.7 0.0004390 -0.04034 4 -4193.436 8394.9 141.27  
## 1 593.7 -0.1121 3 -4196.215 8398.4 144.83  
## 4 362.4 -0.04413 3 -4242.349 8490.7 237.10  
## 3 206.9 0.0005043 3 -4268.379 8542.8 289.16  
## 5 342.9 -1.699 3 -4302.568 8611.1 357.54  
## 2 320.7 2 -4322.212 8648.4 394.83  
## weight  
## 8 0.943  
## 6 0.057  
## 7 0.000  
## 1 0.000  
## 4 0.000  
## 3 0.000  
## 5 0.000  
## 2 0.000  
## Models ranked by AIC(x)

ggdredgeplot(d1) ##this is in catstats2



##ggdredgeplotly(d1) #interactive!!   
##nice to share as html with people so interactive ones work :)

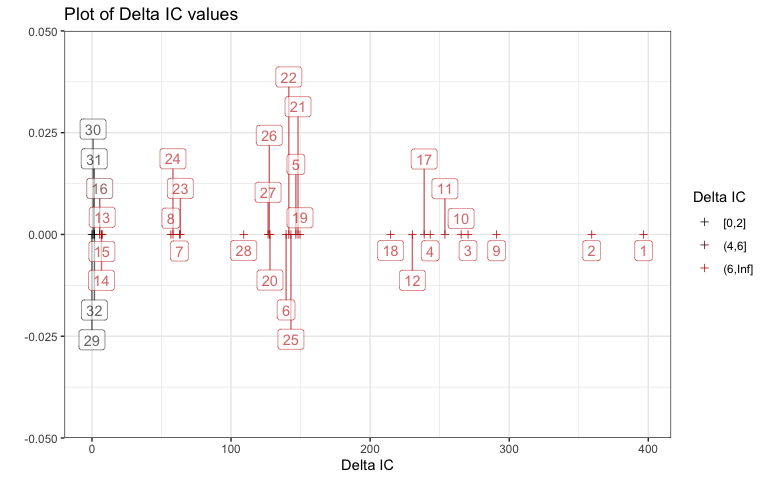
**Q7) Take the “full” model from the previous question and run dredge on it, generating results ranked based on the AIC. Make a plot of the results using ggdredgeplot. How many models were considered? What was in the top AIC model? How many models were within 2 AIC units of the top model?**

32 models were considered! There were four models within 2 AIC units of the top model. The top AIC model was density ~ elev + rad + snow.

options(na.action = "na.fail") #need to run this before dredge !   
res\_AIC <- dredge(lmADD, rank = "AIC")  
dim(res\_AIC) #the rows = number of models explored

## [1] 32 11

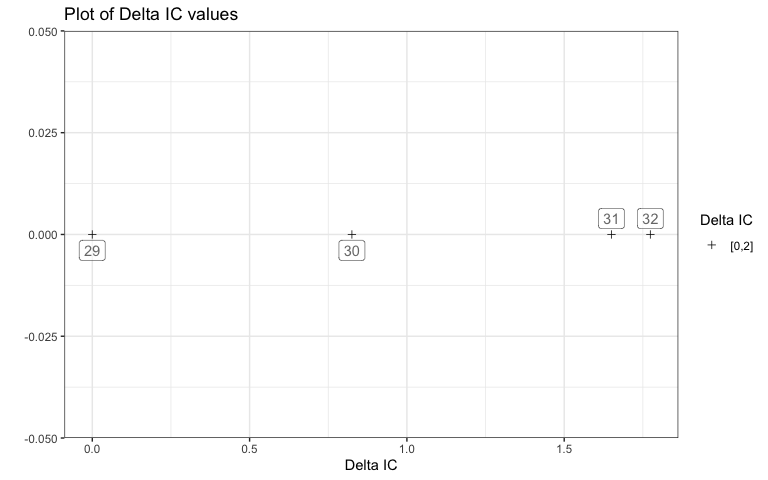
ggdredgeplot(res\_AIC)



##ggdredgeplotly(res\_AIC)  
  
AIC\_2 <- subset(res\_AIC, delta < 2)  
AIC\_2

## Global model call: lm(formula = density ~ elev + rad + snow + angle + AspectCat,   
## data = snowdepthsR)  
## ---  
## Model selection table   
## (Intrc) angle AspcC elev rad snow df logLik AIC delta  
## 29 528.5 -0.1389 0.0005100 0.01638 5 -4120.913 8251.8 0.00  
## 30 525.8 -0.2444 -0.1355 0.0005027 0.01536 6 -4120.325 8252.7 0.83  
## 31 531.8 + -0.1332 0.0004463 0.01477 8 -4118.738 8253.5 1.65  
## 32 528.5 -0.3140 + -0.1284 0.0004326 0.01325 9 -4117.800 8253.6 1.77  
## weight  
## 29 0.398  
## 30 0.264  
## 31 0.174  
## 32 0.164  
## Models ranked by AIC(x)

ggdredgeplot(res\_AIC, deltasubset = 2)



topAICmodel <- eval(attributes(res\_AIC)$model.calls$`29`)  
topAICmodel$coefficients

## (Intercept) elev rad snow   
## 5.284771e+02 -1.389295e-01 5.100051e-04 1.638175e-02

topAICmodel

##   
## Call:  
## lm(formula = density ~ elev + rad + snow + 1, data = snowdepthsR)  
##   
## Coefficients:  
## (Intercept) elev rad snow   
## 528.47706 -0.13893 0.00051 0.01638

**Q8) Report any resources outside your group and the course provided materials that you used and how they impacted your answers.**

NONE