Lab 13

Henry, Gerard, Nitasha

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) that explored the relationship between snow density (kg/m^3) or snow depth (`snow`, mm) with a suite of predictor variables. To be able to measure characteristics of snow, they needed to find snow in the locations they were sampling, so the focus in this lab will be on the snow presence or absence at each location (`SnowPresence`). We will be interested in using `elev` (Elevation, m), `Land` (forest cover with 0 = unforested and 10 = forested), `rad` (Potential Solar radiation, Wh/m^2), `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)), and `angle` (angle of slope in degrees with 0 being flat) as fixed effect 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.

- 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 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)
```

```r
  )
levels(snowdepths$Landf) <- c("Not Forested", "Forested")

snowdepths <- snowdepths %>% mutate(ElMean = ave(elev, strata),
      ElevCent = elev - ElMean)

favstats(aspect ~ 1, data = snowdepths)
```

```
##   1 min Q1 median  Q3 max    mean       sd    n missing
## 1 1   0 84    153 228 359 159.061 93.78348 1017       0
```

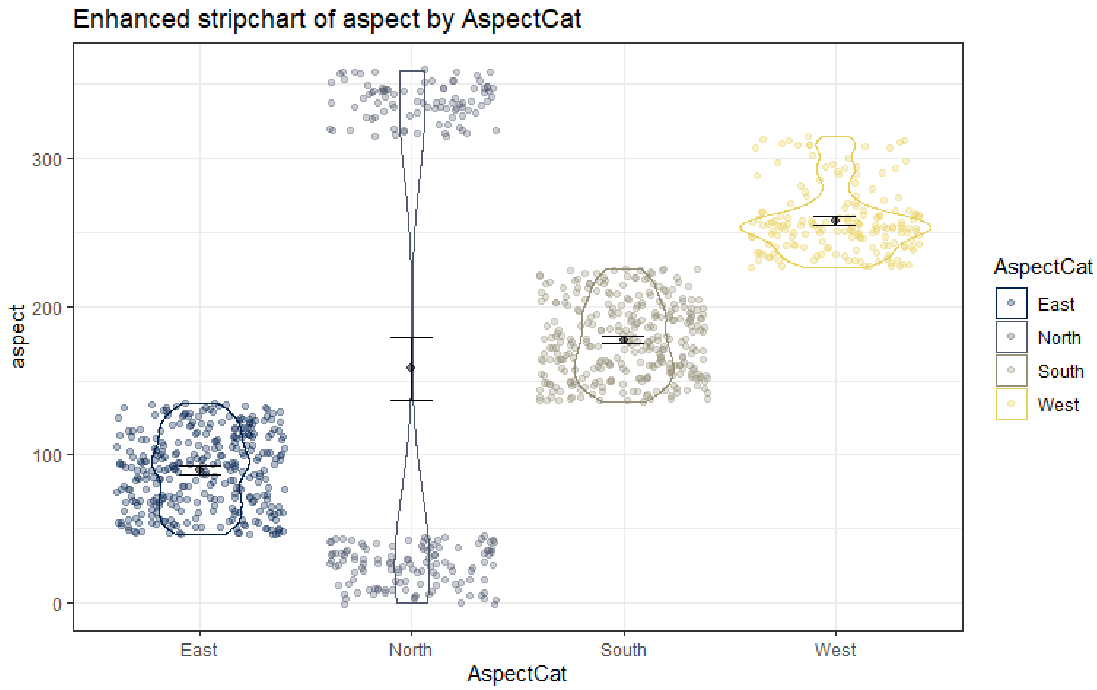
```r
favstats(aspect ~ AspectCat, data = snowdepths)
```

```
##   AspectCat min     Q1 median     Q3 max      mean        sd   n missing
## 1      East  46  68.00     91 111.25 135  89.51562  25.38308 320       0
## 2     North   0  24.25     41 335.75 359 157.92056 155.45378 214       0
## 3     South 136 157.00    176 198.00 225 177.47604  25.11551 313       0
## 4      West 226 241.00    253 263.75 314 257.50000  22.41836 170       0
```

1) Before we dig into modeling, we need to consider how we are going to handle the aspect variable. Consider the sample mean of 159 and other summary statistics for the entire data set or the similar results broken down by `AspectCat`. Make an enhanced stripchart to help you to explain the `favstats` output for the `AspectCat` variable that I created from `aspect`. Why are the conventional summary statistics misleading with a variable such as aspect?

The 'aspect' variable represents the orientation of a slope and ranges from 0 to 360 degrees, where a 360 degree rotation is equivalent to 0. The conventional summary statistics like mean or median are misleading for this data because they don't account for the cyclical nature of the data. i.e 359 degrees is almost the same as 0. The enhanced stripchart helps show the differences in spread for AspectCat and helps to understand the favstats output above. The standard deviation difference for the North AspectCat is better understood after looking at the enhanced stripchart.

```r
enhanced_stripchart(data=snowdepths, formula=aspect~AspectCat, conf.level =
0.95)
```



2) angle is also measured in degrees for the steepness of the slope. Create summary statistics for angle and then discuss how things are different for working with this variable as compared to aspect?

The standard deviation for the summary statistics for angle is much lower than that of the summary statistics for aspect. This is also true for the mean.

```
favstats(angle ~ 1, data = snowdepths)

##   1 min Q1 median Q3 max     mean       sd    n missing
## 1 1   1  8     12 19  51 14.39823 8.792653 1017       0
```

3) "strata" in this study is a grouping variable for conditions of sites that they used to develop their sampling plan. They randomly selected starting points and then sampled from within each strata (think of it as a blocking variable, but instead of random assignment, they randomly sampled locations from all the possible locations within each strata). Because the strata are based on the unique combinations of elevation, solar radiation, and forest cover, we would expect some possible systematic differences in snow characteristics across the strata. How many levels does strata have? What are the smallest and largest numbers of observations in the strata? Report this information as you might in a report in a sentence or two. Also provide the code and output you used to obtain this result.

The strata variable has levels ranging from 2 to 126 observations with 32 levels of strata.

```
strataCount <- table(snowdepths$strata)
min_obs <- min(strataCount)
max_obs <- max(strataCount)
```

```
min_obs

## [1] 2

max_obs

## [1] 126

strataCount

##
## 1100 1110 1200 1210 1300 1310 1400 1410 2100 2110 2200 2210 2300 2310 2400
2410
##    8    7  104   26   66   35   23   23    4   13   30   30   61   81   12
57
## 3100 3110 3200 3210 3300 3310 3400 3410 4100 4110 4200 4210 4300 4310 4400
5400
##   16   16  126   69   91   32   16    2    7   30    3    2    5    4   13
5
```

4) The code above created two new variables, `ElMean` and `ElevCent`. Print out all the rows of the data set for strata "1100" for just these two variables and explain what each variable contains.

'ElMean' is calculated using the ave() function in the code above, computing the mean of elevation for each level of strata. This variable contains the representation of the average elevation of all locations within a particular stratum. 'ElevCent' is the "centered" elevation, and it is calculated as the difference between the actual elevation of a specific location and the mean elevation of its stratum (elev-ElMean). This variable contains/measures how much higher or lower a particular location's elevation is compared to the mean elevation of its respective stratum.

```
snowdepths %>%
  filter(strata == "1100") %>%
  select(ElMean, ElevCent)

## # A tibble: 8 × 2
##   ElMean ElevCent
##    <dbl>    <dbl>
## 1  2020.    -39.7
## 2  2020.    -38.1
## 3  2020.    -27.6
## 4  2020.     95.7
## 5  2020.     94.6
## 6  2020.    -32.5
## 7  2020.    -25.7
## 8  2020.    -26.9
```

5) Make a contingency table that considers whether having a site that is forested/not (`Landf`) is related to encountering some measurable snow/not using `SnowPresence`.

Looking at the contingency table below, we can say that having a site that is not forested is related to encountering some measurable snow.

```
table(snowdepths$Landf, snowdepths$SnowPresence)

##
##                None  Some
##   Not Forested  115   499
##   Forested      109   294
```

6) Use the previous table results to estimate and *carefully* report the odds of encountering snow for each of the `Landf` categories. Then calculate the odds ratio for encountering snow that compares these two odds results. Do this "by hand" and show your work.

- Odds of snow for no land cover: $\frac{499}{115}$ = 4.339 to 1
- Odds of snow for land cover: $\frac{294}{109}$ = 2.697 to 1
- Odds ratio: $\frac{4.339}{2.697}$ = 1.609

7) The land cover status was calculated based on GIS work to define the strata in the sampling plan before they went to Big Sky to collect snow samples. Is this a prospective, cross-sectional, or retrospective study as it regards land cover and snow presence?

With regards to snow presence, we can say that this study is retrospective.

8) We want to confirm the previous work using a GLM. First we need to check how `glm` handled the presence of snow variable. Fit a mean-only logistic GLM and use it to check what category is being treated as a success. Explain how you checked this.

We used the ilogit below and juxtaposed this with the levels to confirm that snow present is the success.

The mean-only model provides a single coefficient that is the estimated log-odds of finding snow for all sites in the data set. Inspecting the table of counts for snow presence in the data set, we can see that there were a total number of 793 observations of snow and 224 of no snow. This results in a log-odds of seeing snow of 1.2641772, the estimate from the glm intercept only model.

Transforming this log-odds using ilogit (inverse logit) gives us the estimated probability of encountering snow. The estimated probability of encountering snow for all sites in the data set is 0.78.

```
glm0 <- glm(SnowPresence ~ 1, data = snowdepths, family = binomial)
summary(glm0)

##
## Call:
```

```
## glm(formula = SnowPresence ~ 1, family = binomial, data = snowdepths)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.26418    0.07567   16.71   <2e-16
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1072.4  on 1016  degrees of freedom
## Residual deviance: 1072.4  on 1016  degrees of freedom
## AIC: 1074.4
##
## Number of Fisher Scoring iterations: 4
```

```r
table(snowdepths$SnowPresence)
```

```
##
## None Some
##  224  793
```

```r
ilogit(glm0$coefficients)
```

```
## (Intercept)
##   0.7797443
```

9) Use a logistic GLM to get an estimate of the log-odds ratio for snow presence/absence based on forested/not that matches your previous work. Hint: you might not get the odds ratio you calculated above and might need to invert (take 1/estimated OR) the OR result to switch the baseline/deviation predictor categories. Show and explain your work.

The coefficient for land forested or not is -0.4754, which is in the log scale. We exponentiate this value to $e^{-0.4754}$ = 0.6216364. Now we can invert this value by 1/0.6216364 which is equivalent to 1.608657.

```r
glm_logistic <- glm(SnowPresence ~ Landf, family = binomial, data =
snowdepths)
summary(glm_logistic)
```

```
##
## Call:
## glm(formula = SnowPresence ~ Landf, family = binomial, data = snowdepths)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1.4677     0.1034  14.189  < 2e-16
## LandfForested  -0.4754     0.1526  -3.116  0.00183
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1072.4  on 1016  degrees of freedom
```

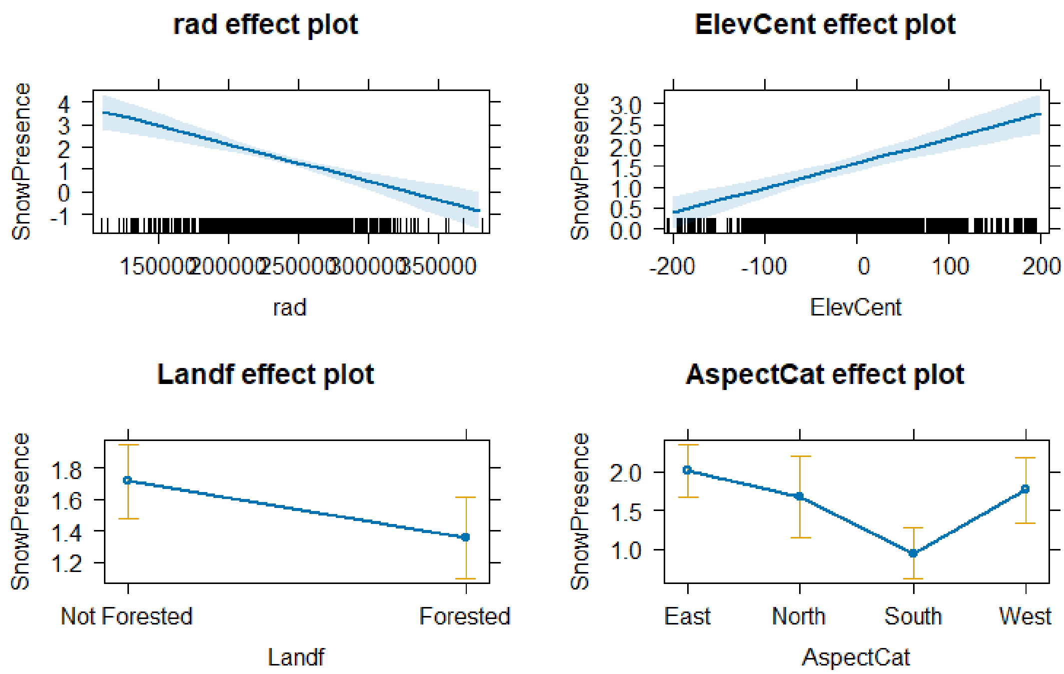
```
## Residual deviance: 1062.7  on 1015  degrees of freedom
## AIC: 1066.7
##
## Number of Fisher Scoring iterations: 4
```

10) One could imagine that they could want to also explore the relationship between presence/absence of snow based on radiation (rad), elevation (mean for strata and variation within strata), Forested/not, and Aspect (categorical version). Fit a model with these five predictors, generate a model summary, and two effects plots (one on the "link" scale and the other on the "response" scale). No discussion.

```
glm_final<- glm(SnowPresence ~ rad + ElevCent + Landf + AspectCat, family =
binomial, data = snowdepths)
summary(glm_final)

##
## Call:
## glm(formula = SnowPresence ~ rad + ElevCent + Landf + AspectCat,
##     family = binomial, data = snowdepths)
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)     5.988e+00  7.274e-01   8.232  < 2e-16
## rad            -1.639e-05  2.987e-06  -5.488 4.06e-08
## ElevCent        5.938e-03  1.014e-03   5.858 4.68e-09
## LandfForested  -3.641e-01  1.722e-01  -2.115   0.0345
## AspectCatNorth -3.382e-01  3.230e-01  -1.047   0.2951
## AspectCatSouth -1.075e+00  2.391e-01  -4.496 6.93e-06
## AspectCatWest  -2.502e-01  2.719e-01  -0.921   0.3573
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1072.39  on 1016  degrees of freedom
## Residual deviance:  885.84  on 1010  degrees of freedom
## AIC: 899.84
##
## Number of Fisher Scoring iterations: 5

# Effects plots
plot(allEffects(glm_final), type="link")
```




```
plot(allEffects(glm_final), type="response")
```