Everest

Victor M. Uribe

2022-09-19

```
**READ ME**
```

To do List: (01/16/23 - Last Worked On)

- Basically done, just figure out what "Linear relationship of independent variables to log odds" means

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidymodels)
```

```
## -- Attaching packages -------------------------------------- tidymodels 1.0.0 --
## v broom        1.0.2     v rsample      1.1.1
## v dials        1.1.0     v tune         1.0.1
## v infer        1.0.4     v workflows    1.1.2
## v modeldata    1.0.1     v workflowsets 1.0.0
## v parsnip      1.0.3     v yardstick    1.1.0
## v recipes      1.0.3
## -- Conflicts ----------------------------------------- tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Use tidymodels_prefer() to resolve common conflicts.
```

```
library(rsample)
ggpairs <- GGally::ggpairs
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
select <- dplyr::select
```

```
members <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/2
```

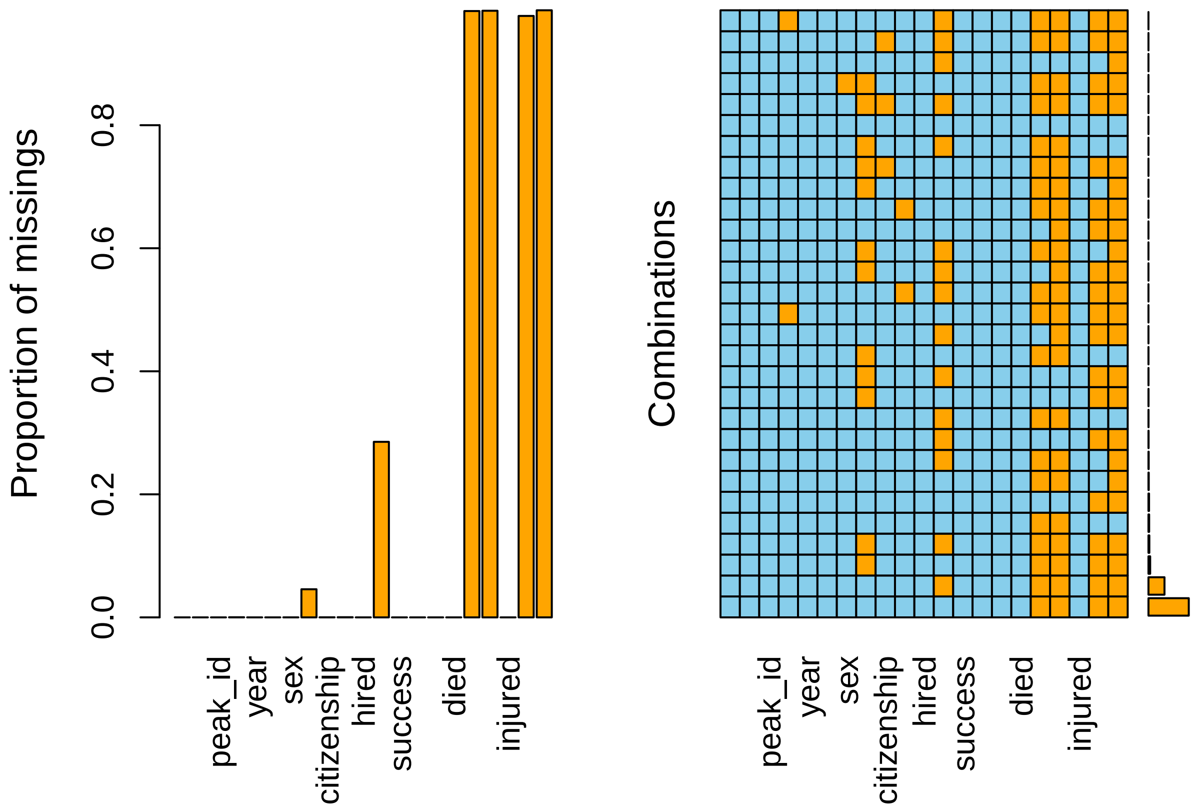
```
## Rows: 76519 Columns: 21
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (10): expedition_id, member_id, peak_id, peak_name, season, sex, citizen...
## dbl  (5): year, age, highpoint_metres, death_height_metres, injury_height_me...
## lgl  (6): hired, success, solo, oxygen_used, died, injured
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
glimpse(members)
```

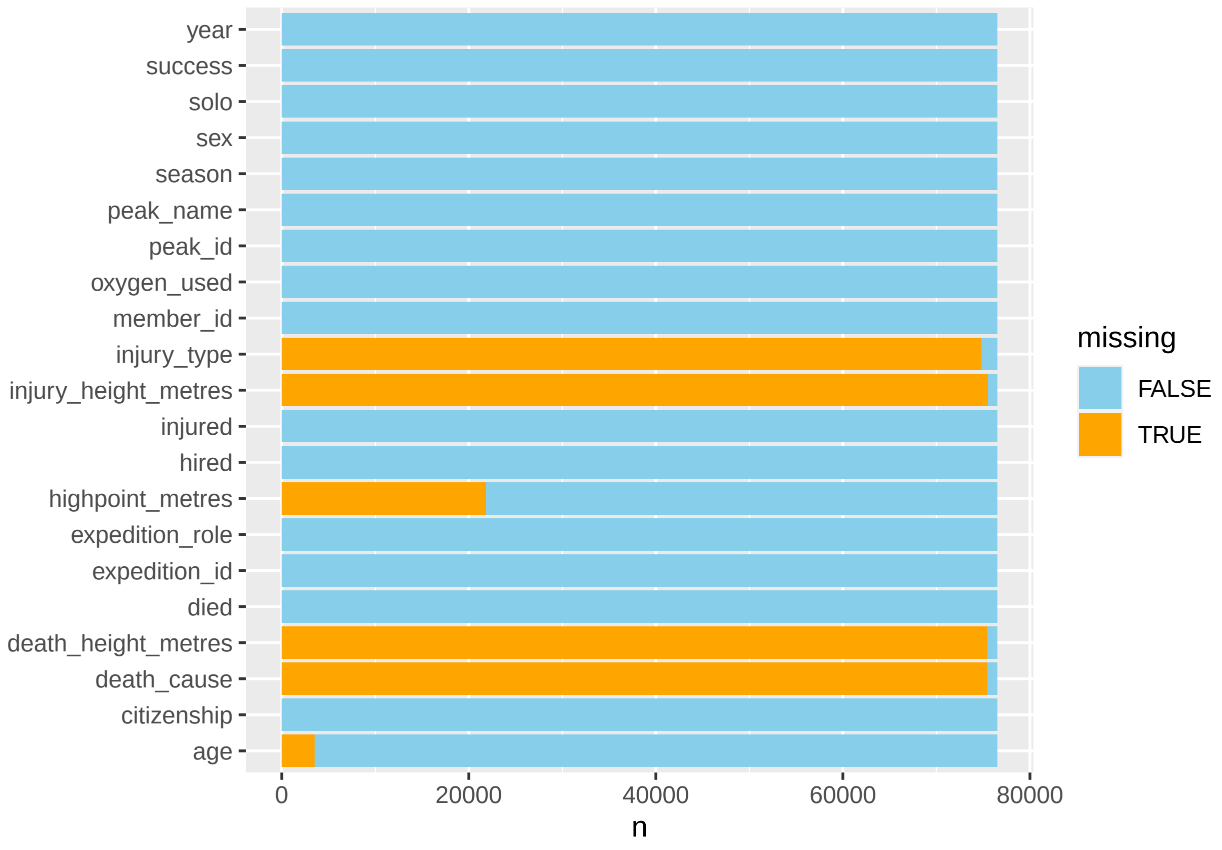
```
## Rows: 76,519
## Columns: 21
## $ expedition_id        <chr> "AMAD78301", "AMAD78301", "AMAD78301", "AMAD78301~
## $ member_id            <chr> "AMAD78301-01", "AMAD78301-02", "AMAD78301-03", "~
## $ peak_id              <chr> "AMAD", "AMAD", "AMAD", "AMAD", "AMAD", "AMAD", "~
## $ peak_name            <chr> "Ama Dablam", "Ama Dablam", "Ama Dablam", "Ama Da~
## $ year                 <dbl> 1978, 1978, 1978, 1978, 1978, 1978, 1978, 1978, 1~
## $ season               <chr> "Autumn", "Autumn", "Autumn", "Autumn", "Autumn",~
## $ sex                  <chr> "M", "M", "M", "M", "M", "M", "M", "M", "M", "M",~
## $ age                  <dbl> 40, 41, 27, 40, 34, 25, 41, 29, 35, 37, 23, 44, 2~
## $ citizenship          <chr> "France", "France", "France", "France", "France",~
## $ expedition_role      <chr> "Leader", "Deputy Leader", "Climber", "Exp Doctor~
## $ hired                <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, ~
## $ highpoint_metres     <dbl> NA, 6000, NA, 6000, NA, 6000, 6000, 6000, NA, 681~
## $ success              <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, ~
## $ solo                 <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, ~
## $ oxygen_used          <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, ~
## $ died                 <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, ~
## $ death_cause          <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ death_height_metres  <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ injured              <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, ~
## $ injury_type          <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
## $ injury_height_metres <dbl> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, N~
```

```
attach(members)
```

```
VIM::aggr(members, col = c("skyblue","orange"))
```



```
missvalues_visual <-
    members  %>%
      summarise_all(list(~is.na(.)))%>%
      pivot_longer(everything(),
                   names_to = "variables", values_to="missing") %>%
      count(variables, missing) %>%
      ggplot(aes(y=variables,x=n,fill=missing))+
      geom_col()+
      scale_fill_manual(values=c("skyblue","orange"))+
      theme(axis.title.y=element_blank())
missvalues_visual
```



So most of the data that is giving issues is composed of mostly na

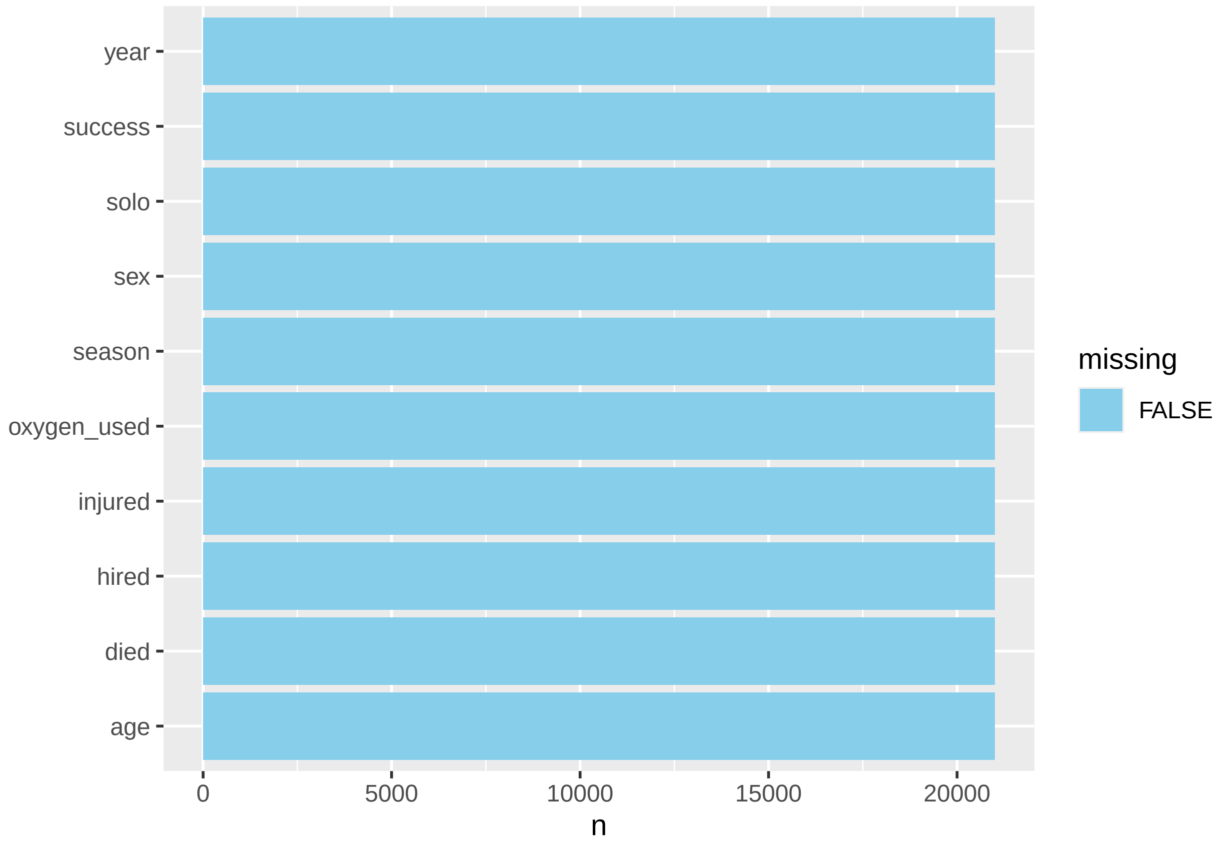
Data cleaning

```
everest <- members %>% # should have an age
  filter(age != 'NA' & peak_name == "Everest") %>%
  select(-c(peak_id,peak_name,expedition_id,member_id, death_height_metres,
            injury_height_metres, highpoint_metres, death_cause,
            citizenship, expedition_role, injury_type)) %>%
  mutate(
    #death_height_metres = ifelse(death_height_metres == NA, 0, death_height_metres), # issue wiith lev
    #injury_height_metres = ifelse(injury_height_metres == NA, 0, injury_height_metres), # gives leveli
    #highpoint_metres = ifelse(highpoint_metres == 'NA', 0, highpoint_metres), # thows off the p values
    season = factor(season),
    sex = factor(sex),
    #citizenship = factor(citizenship), # not significant after testing
    #expedition_role = factor(expedition_role), # not significant after testing
    hired = factor(hired),
    success = factor(success), # value being predicted
    solo = factor(solo),
    oxygen_used = factor(oxygen_used),
    died = factor(died),
    #death_cause = factor(death_cause), # issue
    injured = factor(injured)#,
    #injury_type = factor(injury_type) # issue with levels
  )
```

```r
glimpse(everest)
```

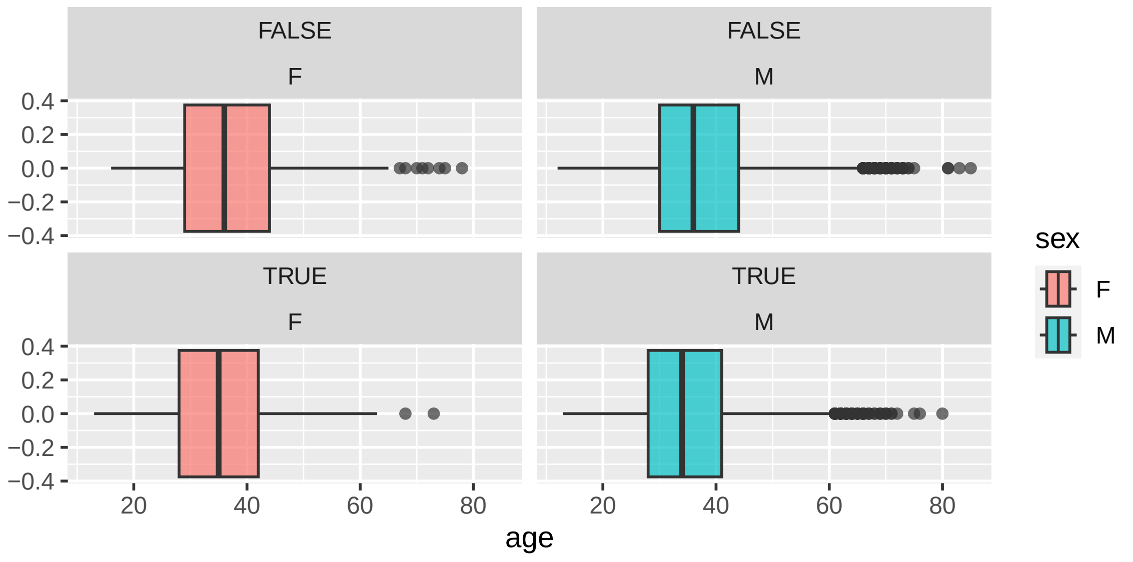
```
## Rows: 20,997
## Columns: 10
## $ year        <dbl> 1963, 1963, 1963, 1963, 1963, 1963, 1963, 1963, 1963, 1963~
## $ season      <fct> Spring, Spring, Spring, Spring, Spring, Spring, Spring, Sp~
## $ sex         <fct> M, M, M, M, M, M, M, M, M, M, M, M, M, M, M, M, M, M, M, M~
## $ age         <dbl> 36, 31, 27, 26, 26, 29, 44, 37, 32, 26, 34, 42, 35, 23, 27~
## $ hired       <fct> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FA~
## $ success     <fct> FALSE, TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, TRU~
## $ solo        <fct> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FA~
## $ oxygen_used <fct> TRUE, TRUE, FALSE, TRUE, TRUE, FALSE, TRUE, TRUE, TRUE, TR~
## $ died        <fct> FALSE, FALSE, TRUE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL~
## $ injured     <fct> FALSE, FALSE, FALSE, FALSE, FALSE, TRUE, FALSE, FALSE, FAL~
```

```r
missvalues_visual2 <-
    everest  %>%
      summarise_all(list(~is.na(.)))%>%
      pivot_longer(everything(),
                   names_to = "variables", values_to="missing") %>%
      count(variables, missing) %>%
      ggplot(aes(y=variables,x=n,fill=missing))+
      geom_col()+
      scale_fill_manual(values=c("skyblue","orange"))+
      theme(axis.title.y=element_blank())
missvalues_visual2
```



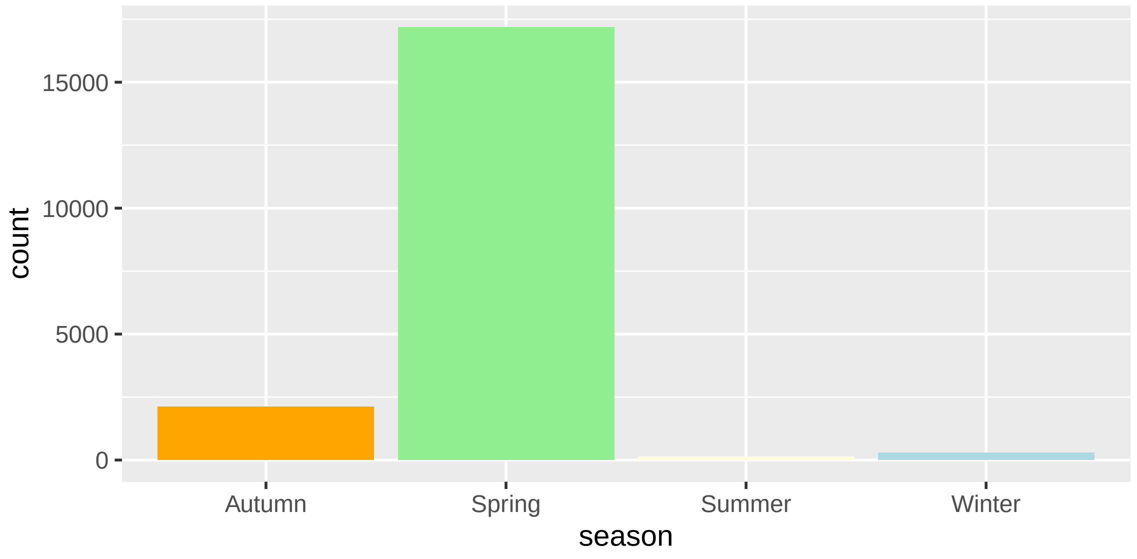
EDA

```
ggplot(everest_clean, aes(x = age, fill = sex)) + geom_boxplot(alpha = .7) +
  facet_wrap(~ success + sex )
```



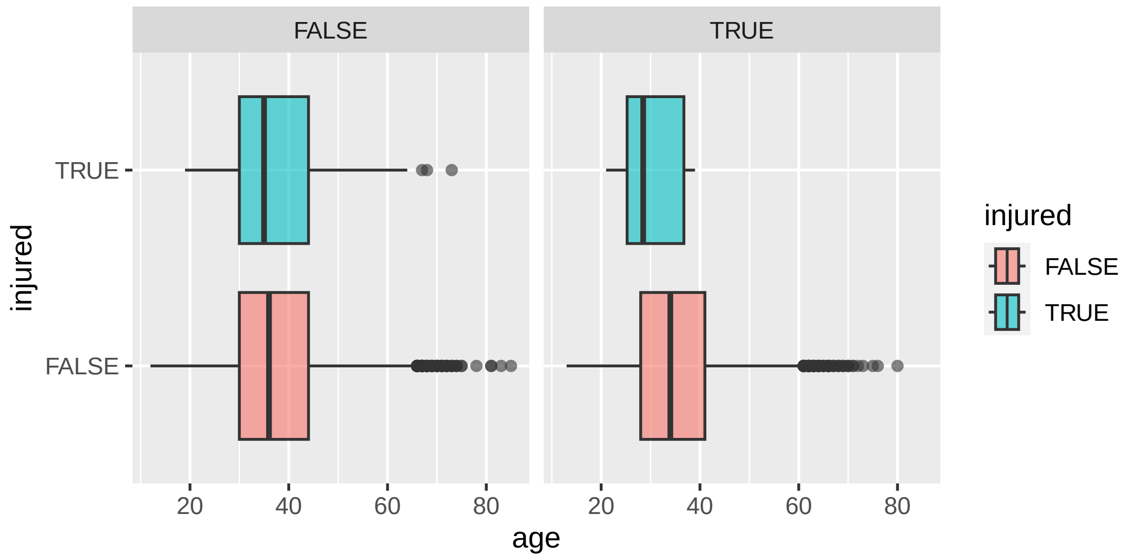
The figure above shows the success of each sex plotted on age. For it we see the median for males shift left meaning that the older you are the less likely you are to succeed. On the other hand we see that the median of the females plot tends to stay stationary, meaning that there aren't many attempts after the age of 60 years of age but also of those that do attempt much of them don't succeed.

```
(p_season <- ggplot(everest_clean, aes(x = season)) +
   geom_bar( fill = c("orange","light green","light yellow","light blue")))
```



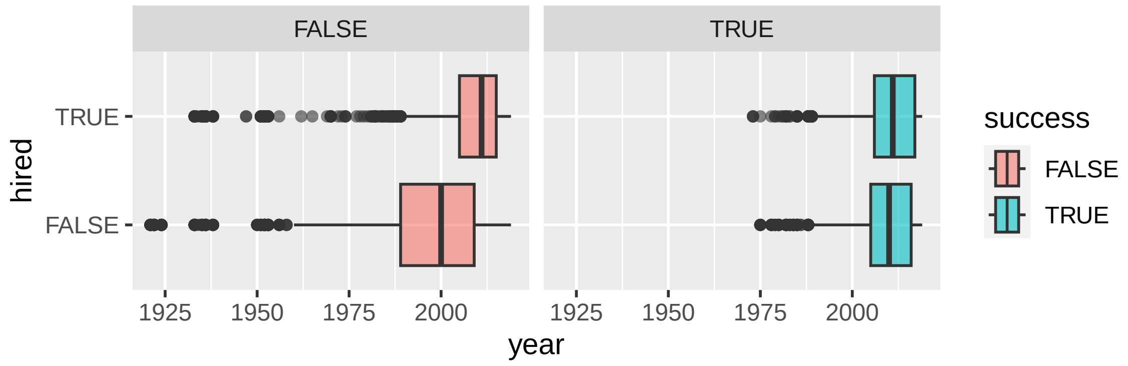
As seen the number of climbers whose season of expedition is spring or autumn is significantly larger than that of winter and summer. In order to understand the reason for it, I conducted some research, and found out that some environmental conditions, including barometric pressure (BP), perceived altitude (Alt), and climbing speed (Speed), are worse in the midwinter season than in May (Szymczak).

```r
ggplot(everest_clean) + geom_boxplot(aes(x = age, y = injured, fill = injured),
                                     alpha = .6) + facet_wrap(~success)
```



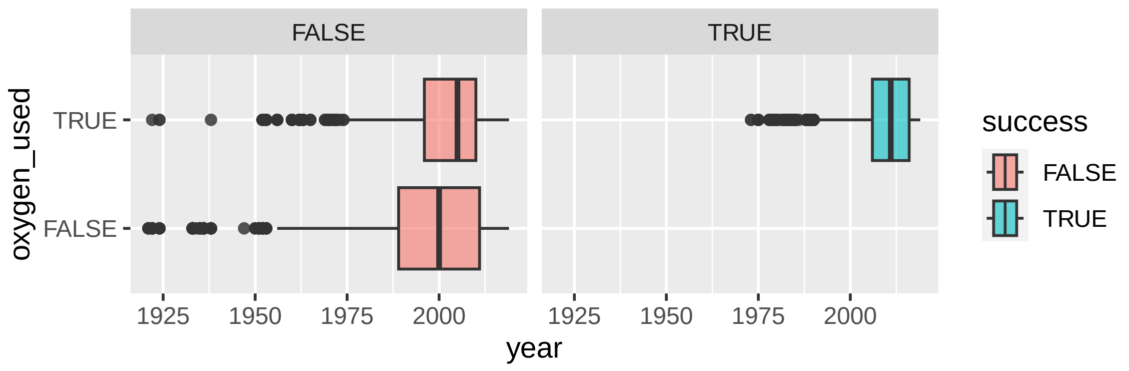
The plot above shows us the age and if they were injured or not for both distributions of success and failure. Which as you can see folks older than 40 years of age tend not to succeed when injured. Even then looking at the median it seems that most that get injured and succeed are under 30 years of age.

```r
ggplot(everest_clean) + geom_boxplot(aes(x = year, y = hired, fill = success),
                                     alpha = .6) + facet_wrap(~success)
```



For the figure above we see that most of the people hired was in the 2000's. Meaning that people more people were hired as the demand increase with the years. We also see that individuals who where hired tend to succeed a bit more often.

```r
ggplot(everest_clean) + geom_boxplot(aes(x = year, y = oxygen_used, fill = success)
                                     , alpha = .6) + facet_wrap(~success)
```



As seen above there was absolutely no one that successfully climbed Everest without using oxygen. Which makes sense since Everest stands at a staggering 29,032 feet.

Training and Testing data

```
set.seed(5302)
split <- initial_split(everest_clean, prop = .70)

train_data <- training(split)
test_data <- testing(split)
```

fiting models

```
sapply(lapply(everest, unique), length)
```

```
##        year      season         sex         age       hired     success
##          72           4           2          70           2           2
##        solo oxygen_used        died     injured
##           2           2           2           2
```

```
sapply(lapply(everest_clean, unique), length)
```

```
##        year      season         sex         age       hired     success
##          72           4           2          69           2           2
##        solo oxygen_used        died     injured
##           1           2           2           2
```

solo needs to be taken out since the outliers that were removed were all the climbers that attempted to climb solo.

Lasso Regression

```
lasso_spec <- logistic_reg(penalty = 0.1, mixture = 1) %>%
  set_engine("glmnet")

wf <- workflow() %>%
  add_formula(success ~.)

lasso_fit <- wf %>%
  add_model(lasso_spec) %>%
  fit(train_data)

lasso_fit %>%
  pull_workflow_fit() %>%
  tidy()
```

```
## Warning: `pull_workflow_fit()` was deprecated in workflows 0.2.3.
## i Please use `extract_fit_parsnip()` instead.

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Loaded glmnet 4.1-6

## # A tibble: 12 x 3
##    term             estimate penalty
##    <chr>               <dbl>   <dbl>
##  1 (Intercept)         -2.04     0.1
##  2 year                 0        0.1
##  3 seasonSpring         0        0.1
##  4 seasonSummer         0        0.1
##  5 seasonWinter         0        0.1
##  6 sexM                 0        0.1
##  7 age                  0        0.1
##  8 hiredTRUE            0        0.1
##  9 soloTRUE             0        0.1
## 10 oxygen_usedTRUE      3.07     0.1
## 11 diedTRUE             0        0.1
## 12 injuredTRUE          0        0.1
```

```
set.seed(5302)
boot <- bootstraps(train_data, strata = success)

tune_spec <- logistic_reg(penalty = tune(), mixture = 1) %>%
  set_engine("glmnet")

lambda_grid <- grid_regular(penalty(), levels = 50)
```

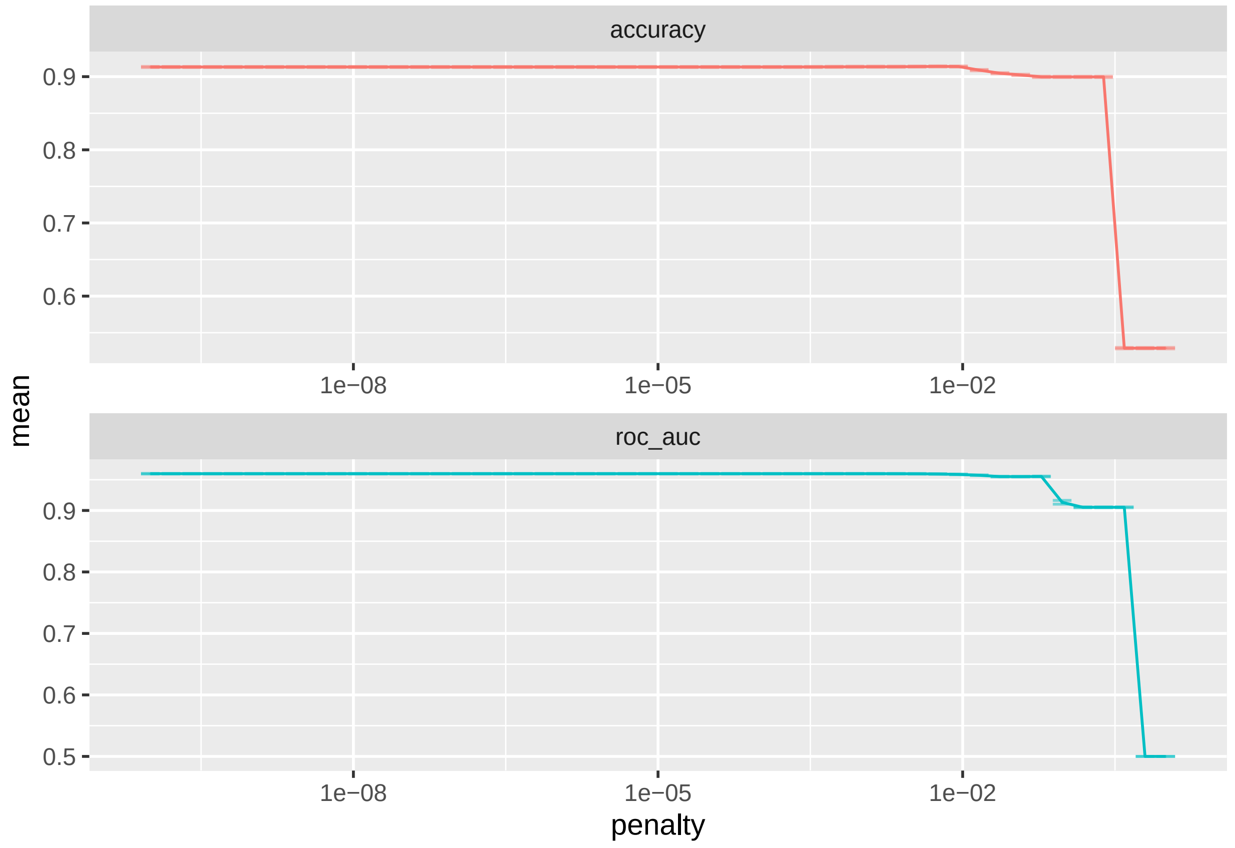
```
doParallel::registerDoParallel()
set.seed(5302)
```

```r
lasso_grid <- tune_grid(
  wf %>% add_model(tune_spec),
  resamples = boot,
  grid = lambda_grid
)
```

```r
lasso_grid %>%
  collect_metrics()
```

```
## # A tibble: 100 x 7
##     penalty .metric  .estimator  mean     n  std_err .config
##       <dbl> <chr>    <chr>      <dbl> <int>    <dbl> <chr>
##  1 1   e-10 accuracy binary     0.913    25 0.000764 Preprocessor1_Model01
##  2 1   e-10 roc_auc  binary     0.960    25 0.000489 Preprocessor1_Model01
##  3 1.60e-10 accuracy binary     0.913    25 0.000764 Preprocessor1_Model02
##  4 1.60e-10 roc_auc  binary     0.960    25 0.000489 Preprocessor1_Model02
##  5 2.56e-10 accuracy binary     0.913    25 0.000764 Preprocessor1_Model03
##  6 2.56e-10 roc_auc  binary     0.960    25 0.000489 Preprocessor1_Model03
##  7 4.09e-10 accuracy binary     0.913    25 0.000764 Preprocessor1_Model04
##  8 4.09e-10 roc_auc  binary     0.960    25 0.000489 Preprocessor1_Model04
##  9 6.55e-10 accuracy binary     0.913    25 0.000764 Preprocessor1_Model05
## 10 6.55e-10 roc_auc  binary     0.960    25 0.000489 Preprocessor1_Model05
## # ... with 90 more rows
```

```r
lasso_grid %>%
  collect_metrics() %>%
  ggplot(aes(penalty, mean, color = .metric)) +
  geom_errorbar(aes(
    ymin = mean - std_err,
    ymax = mean + std_err
  ),
  alpha = 0.5
  ) +
  geom_line() +
  facet_wrap(~.metric, scales = "free", nrow = 2) +
  scale_x_log10() +
  theme(legend.position = "none")
```

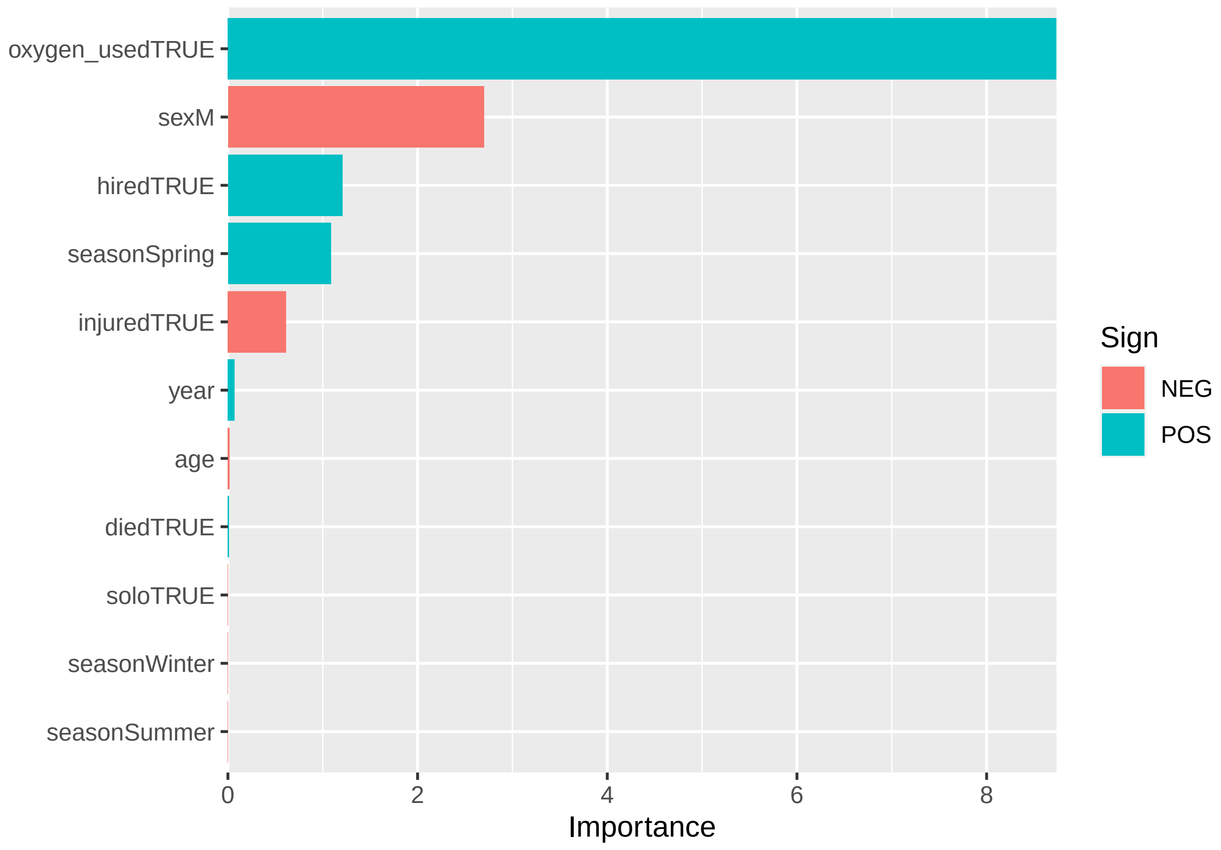


```r
lowest_auc <- lasso_grid %>%
  select_best("roc_auc", max = FALSE)

final_lasso <- finalize_workflow(
  wf %>% add_model(tune_spec),
  lowest_auc
)
```

```r
library(vip)
```

```
##
## Attaching package: 'vip'

## The following object is masked from 'package:utils':
##
##     vi
```

```r
final_lasso %>%
  fit(train_data) %>%
  pull_workflow_fit() %>%
  vi(lambda = lowest_auc$penalty) %>%
  mutate(
    Importance = abs(Importance),
    Variable = fct_reorder(Variable, Importance)
  ) %>%
  ggplot(aes(x = Importance, y = Variable, fill = Sign)) +
  geom_col() +
  scale_x_continuous(expand = c(0, 0)) +
  labs(y = NULL)
```



```
last_fit(
  final_lasso,
  split
) %>%
  collect_metrics()
```

```
## # A tibble: 2 x 4
##   .metric  .estimator .estimate .config
##   <chr>    <chr>          <dbl> <chr>
## 1 accuracy binary         0.914 Preprocessor1_Model1
## 2 roc_auc  binary         0.958 Preprocessor1_Model1
```

```
fin_mod <- final_lasso %>%
  fit(train_data)

prob <- fin_mod %>%
  predict(new_data = test_data)

tmp <- factor(ifelse(prob == TRUE, TRUE, FALSE))
caret::confusionMatrix(tmp, test_data[["success"]])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  2647   44
##      TRUE    468 2762
##
##                Accuracy : 0.9135
##                  95% CI : (0.9061, 0.9206)
##     No Information Rate : 0.5261
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8279
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8498
##             Specificity : 0.9843
##          Pos Pred Value : 0.9836
##          Neg Pred Value : 0.8551
##              Prevalence : 0.5261
##          Detection Rate : 0.4471
##    Detection Prevalence : 0.4545
##       Balanced Accuracy : 0.9170
##
##        'Positive' Class : FALSE
##
```

```
fin_mod %>%
  tidy() %>%
  filter(estimate != 0)
```

```
## # A tibble: 9 x 3
##   term             estimate  penalty
##   <chr>               <dbl>    <dbl>
## 1 (Intercept)     -148.     0.000869
## 2 year               0.0708 0.000869
## 3 seasonSpring       1.09   0.000869
## 4 sexM              -2.70   0.000869
## 5 age               -0.0165 0.000869
## 6 hiredTRUE          1.21   0.000869
## 7 oxygen_usedTRUE    8.73   0.000869
## 8 diedTRUE           0.0127 0.000869
## 9 injuredTRUE       -0.613  0.000869
```

Note that we need to run a group lasso to account for seasons properly since it has more that two levels. But this still tells us the best variables.

Classic Logistic Regression

```
sig_mod_aic <- glm(success ~ year + age + sex + hired + oxygen_used + died + injured,
               data = train_data, family = "binomial") %>% MASS::stepAIC(trace = FALSE)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
sig_mod_aic %>% summary()
```

```
##
## Call:
## glm(formula = success ~ year + age + sex + hired + oxygen_used +
##     died + injured, family = "binomial", data = train_data)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.69231  -0.00002  -0.00001   0.43524   1.80797
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -1.825e+02  1.910e+02  -0.955  0.33939
## year             8.225e-02  3.749e-03  21.941  < 2e-16 ***
## age             -1.742e-02  3.701e-03  -4.708 2.50e-06 ***
## sexM            -4.207e+00  7.123e-01  -5.905 3.52e-09 ***
## hiredTRUE        1.217e+00  8.119e-02  14.983  < 2e-16 ***
## oxygen_usedTRUE  2.333e+01  1.908e+02   0.122  0.90269
## diedTRUE         2.626e+00  2.305e+00   1.139  0.25456
## injuredTRUE     -1.468e+00  5.368e-01  -2.735  0.00623 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 19107.5  on 13814  degrees of freedom
## Residual deviance:  5938.5  on 13807  degrees of freedom
## AIC: 5954.5
##
## Number of Fisher Scoring iterations: 19
```

```r
# Predict on test
p <- predict(sig_mod_aic, newdata = test_data, type = "response")

# If p exceeds threshold of 0.5, 1 else 0
yes_no <- ifelse(p > 0.5, 1, 0)

# Convert to factor: p_class
p_class <- factor(ifelse(yes_no == 1, TRUE, FALSE))

# Create confusion matrix
caret::confusionMatrix(p_class, test_data[["success"]])
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE  2661   59
##      TRUE    454 2747
##
##                Accuracy : 0.9134
##                  95% CI : (0.9059, 0.9204)
##     No Information Rate : 0.5261
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.8274
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8543
##             Specificity : 0.9790
##          Pos Pred Value : 0.9783
##          Neg Pred Value : 0.8582
##              Prevalence : 0.5261
##          Detection Rate : 0.4494
##    Detection Prevalence : 0.4594
##       Balanced Accuracy : 0.9166
##
##        'Positive' Class : FALSE
##
```

	Accuracy	Sensitivity	Specificity
Lasso	0.9135	0.8498	0.9843
AIC	0.9134	0.8515	0.9817

Validating Assumptions

Logistic Regression Assumptions:

- Response variable is binary or dichotomous

```
levels(everest_clean$success)
```

```
## [1] "FALSE" "TRUE"
```

- Linear relationship of independent variables to log odds
- large sample size

```
dim(everest_clean)
```

```
## [1] 19736    10
```

Which is 39x the minimum sample size needed (500 is the minimum)

- Problem with extreme outliers

We use cooks distance to eliminate the outliers

- independent observations

As each row represents a unique climber we can assume they are independent from one another.

- No multicollinearity among the predictor variables

df	Weak	Medium	Strong
1	0.10	0.30	0.50
2	0.07	0.21	0.35
3	0.06	0.17	0.29
4	0.05	0.15	0.25
5	0.04	0.13	0.22

using Cramer's V we conclude that there is no violation of multicollinearity between the predictor variables. Note I used Cramer's V and the table provided above to find strong associations between the categorical predictors and numerical.

```
#vcd::assocstats( xtabs(~ sex + season, everest_clean) ) #not correlated
#vcd::assocstats( xtabs(~ sex + oxygen_used, everest_clean) )#not correlated
#vcd::assocstats( xtabs(~ sex + injured, everest_clean) )#not correlated
#vcd::assocstats( xtabs(~ sex + hired, everest_clean) )#not correlated
#vcd::assocstats( xtabs(~ sex + died, everest_clean) )#not correlated

#vcd::assocstats( xtabs(~ season + oxygen_used, everest_clean) )#not correlated
#vcd::assocstats( xtabs(~ season + injured, everest_clean) )#not correlated
#vcd::assocstats( xtabs(~ season + hired, everest_clean) )#not correlated
#vcd::assocstats( xtabs(~ season + died, everest_clean) )#not correlated

#vcd::assocstats( xtabs(~ oxygen_used + injured, everest_clean) )#not correlated
#vcd::assocstats( xtabs(~ oxygen_used + hired, everest_clean) )#not correlated
#vcd::assocstats( xtabs(~ oxygen_used + died, everest_clean) )#not correlated

#vcd::assocstats( xtabs(~ injured + hired, everest_clean) )#not correlated
#vcd::assocstats( xtabs(~ injured + died, everest_clean) )#not correlated

#vcd::assocstats( xtabs(~ hired + died, everest_clean) )#not correlated
```

```
#vcd::assocstats( xtabs(~ year + solo, everest_clean) )#not correlated
#vcd::assocstats( xtabs(~ year + sex, everest_clean) )#not correlated
#vcd::assocstats( xtabs(~ year + season, everest_clean) )#not correlated
#vcd::assocstats( xtabs(~ year + oxygen_used, everest_clean) )#            correlated
#vcd::assocstats( xtabs(~ year + injured, everest_clean) )#not correlated
#vcd::assocstats( xtabs(~ year + hired, everest_clean) )#not correlated
#vcd::assocstats( xtabs(~ year + died, everest_clean) )#not correlated
```

```
#vcd::assocstats( xtabs(~ age + solo, everest_clean) )#not correlated
#vcd::assocstats( xtabs(~ age + sex, everest_clean) )#not correlated
#vcd::assocstats( xtabs(~ age + season, everest_clean) )#not correlated
#vcd::assocstats( xtabs(~ age + oxygen_used, everest_clean) )#not correlated
#vcd::assocstats( xtabs(~ age + injured, everest_clean) )#not correlated
#vcd::assocstats( xtabs(~ age + hired, everest_clean) )#not correlated
#vcd::assocstats( xtabs(~ age + died, everest_clean) )#not correlated
```