Spectral Typing

Here we will take a look at the spectral typing of stars toward Cep OB3b.

```
#devtools::install_github("rstudio/reticulate")
library(reticulate)
use_python("/anaconda3/bin/python")
library(readr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
source("/Users/thomasallen/cep_ob3b/cepr/lib/helpers.R")

#library('ProjectTemplate')
#project_directory<-"/Users/thomasallen/cep_ob3b/cepr"
#setwd(project_directory)
#load.project()
```

```
import sys as sys
sys.path.append("/Users/thomasallen/Code/python_scripts/Functions")
```

```
import sys as sys
sys.path.append("/Users/thomasallen/Code/python_scripts/Functions")
from hecto_funcs import hectospec_fits_open_index
from astropy.io import ascii
import numpy as np
import matplotlib.pyplot as plt
dir_in='/Users/thomasallen/cep_ob3b/data/Spectroscopy/'
standards_fn=dir_in+'spec_standards.dat'
#print(standards_fn)
stand=ascii.read(standards_fn)
spt=stand['col1']
fn=stand['col2']
num=stand['col3']
off = np.arange(len(fn))
#print(fn)
#print('Off')
#print(off)
psize=16
fsize=22
fsize2=24
fig = plt.figure(figsize=(10,15))
plt.gcf().subplots_adjust(bottom=0.15,left=0.15)
plt.subplot(1,1,1)
```

```python
for ii in range(len(fn)):
    wav,flux = hectospec_fits_open_index(fn[ii])
    #print('')
    #print(ii)
    #print(len(wav))
    #print(len(flux))
    #plt.plot(wav,flux+off[ii])
#plt.ylim=[0,100]
#plt.xlabel(r"Wavelength ($\mu \mathrm{m}$)",fontsize=fsize2)
#
#plt.show()
```

```
## Flux
## [  944.2098   922.1142   931.3894 ... 17336.68   17145.03   15970.299 ]
## 4543
## Flux
## [21767.719 21726.934 21193.781 ... 93158.01  90285.36  85997.97 ]
## 4543
## Flux
## [ 9094.187  9162.946  8997.677 ... 63426.586 61649.402 58675.094]
## 4543
## Flux
## [ 9129.164  9265.256  9209.305 ... 45844.83  44630.625 42089.754]
## 4543
## Flux
## [ 186.06392  168.95877  183.75089 ... 4945.651   4793.325   4477.2935 ]
## 4543
## Flux
## [ 231.7342   228.94724  232.03793 ... 4453.028   4332.4634  4024.2524 ]
## 4543
## Flux
## [ 2740.4292  2652.0364  2560.6416 ... 18391.879  18125.5    17627.938 ]
## 4543
## Flux
## [ 429.02637  450.65308  384.5201  ... 8769.655   8635.592   8533.014  ]
## 4543
## Flux
## [ 252.2685   255.24324  261.85608 ... 3151.7803  3055.7188  2970.0408 ]
## 4543
## Flux
## [ 346.53986  356.17554  362.41397 ... 8550.372   8432.979   8194.168  ]
## 4543
## Flux
## [ 371.73987  336.4281   353.80847 ... 3264.0452  3141.7712  2978.4846 ]
## 4543
## Flux
## [ 2402.8838  2403.0288  2289.3652 ... 18606.92   18473.836  17583.719 ]
## 4543
## Flux
## [ 5258.8794  5133.4463  5032.046  ... 25138.998  24431.46   22627.498 ]
## 4543
## Flux
## [1521.1886 1450.7555 1350.3743 ... 9787.233  9620.314  9390.855 ]
## 4543
```

```
## Flux
## [1896.3911 1929.2911 1889.4901 ... 7461.7134 7331.8237 6998.0244]
## 4543
## Flux
## [  990.4971   1032.1786    903.86505 ... 12095.457   11529.601
##  10914.6455 ]
## 4543
## Flux
## [  507.64462   539.1483    576.3832  ... 13944.122   13760.013
##  12780.945  ]
## 4543
## Flux
## [1434.6174 1313.88   1230.8684 ... 8140.4043 7903.6055 7597.2573]
## 4543
## Flux
## [ 2873.978   2877.4749  2527.612  ... 10711.92   10905.245  10407.16  ]
## 4543
## Flux
## [2424.294  2486.9841 2326.6396 ... 9609.862  9609.04   9370.304 ]
## 4543
## Flux
## [  593.3467    563.51996   575.98236 ... 17954.213   17123.432
##  16108.525  ]
## 4543
## Flux
## [  693.69116   678.2676    628.2836  ... 21404.      20988.14
##  19976.334  ]
## 4543
## Flux
## [ 767.65295  732.82587  720.4442  ... 8505.418   8196.537   7790.302  ]
## 4543
## Flux
## [  306.44168   305.56418   289.15424 ... 18814.988   18589.727
##  17887.42   ]
## 4543
## Flux
## [ 838.5007   796.93756  755.2847  ... 7211.3003  7378.1245  7031.4087 ]
## 4543
## Flux
## [ 1353.667   1241.0208  1079.1663 ... 11081.93   10708.145  10131.838 ]
## 4543
## Flux
## [ 1099.3875  1074.6357   903.7108 ... 10336.321  10136.765   9724.313 ]
## 4543
## Flux
## [ 186.95224  179.56717  191.68216 ... 4962.128   5100.0435  4967.947  ]
## 4543
## Flux
## [ 411.58237  391.0371   363.28476 ... 8616.452   8385.051   7974.6665 ]
## 4543
## Flux
## [ 366.03314  368.53394  348.5869  ... 5733.5103  5622.2915  5296.6753 ]
## 4543
## Flux
```

```
## [ 152.90681  157.9676   153.724   ... 5121.768   4979.537   4849.9    ]
## 4543
## Flux
## [  556.7527   529.4905   510.1053 ... 14880.618  14546.254  14028.432 ]
## 4543
## Flux
## [  48.301376    6.456131   41.613266 ... 5571.078    5503.0386
##   5268.404   ]
## 4543
## Flux
## [  29.924053   54.605118   25.296505 ... 2435.7358   2275.2878
##   2049.7485  ]
## 4543
```

We read in the data set. Here we will start with the `full.df.csv` dataset. For details about this data set see the data documentation. We are interested in the spectral typing of the stars with *Hectospec* spectra.

```
data_path <- "/Users/thomasallen/cep_ob3b/cepr/data/"
data_path2 <- "/Users/thomasallen/cep_ob3b/data/"

#full.df.csv

full.df <- read_csv(paste(data_path2,"full.df.csv",sep=""))
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   X1 = col_integer(),
##   bmag = col_character(),
##   berr = col_character(),
##   vmag = col_character(),
##   verr = col_character(),
##   imag = col_character(),
##   ierr = col_character(),
##   cluster = col_character(),
##   cloud = col_character(),
##   disk = col_character(),
##   xray = col_character(),
##   acis = col_character(),
##   spec = col_character(),
##   chelle = col_character(),
##   spt = col_character(),
##   spterr = col_character(),
##   tio = col_character(),
##   tior = col_character(),
##   cah = col_character(),
##   cahr = col_character()
##   # ... with 28 more columns
## )
```

```
## See spec(...) for full column specifications.
```

```
#head(full.df)
gsdss.df <- read_csv(paste(data_path,"gsdss.csv",sep=""))
```

```
## Parsed with column specification:
## cols(
##   mag = col_double(),
##   err = col_double()
## )
```

```
rsdss.df <- read_csv(paste(data_path,"rsdss.csv",sep=""))
```

```
## Parsed with column specification:
## cols(
##   mag = col_double(),
##   err = col_double()
## )
```

```
full.df <- full.df %>%
    mutate(gmag=gsdss.df$mag,gerr=gsdss.df$err) %>%
    mutate(rmag=rsdss.df$mag,rerr=rsdss.df$err)

full.df <- full.df %>%
    mutate(bmag=as.numeric(bmag)) %>%
    mutate(berr=as.numeric(berr)) %>%
    mutate(vmag=as.numeric(vmag)) %>%
    mutate(verr=as.numeric(verr)) %>%
    mutate(imag=as.numeric(imag)) %>%
    mutate(ierr=as.numeric(ierr))
```

We want the objects that have spectral types. These will be rows where the columns **spt**, the spectral type as classified by eye, and **spt_old**, the spectral type as classified by regression.

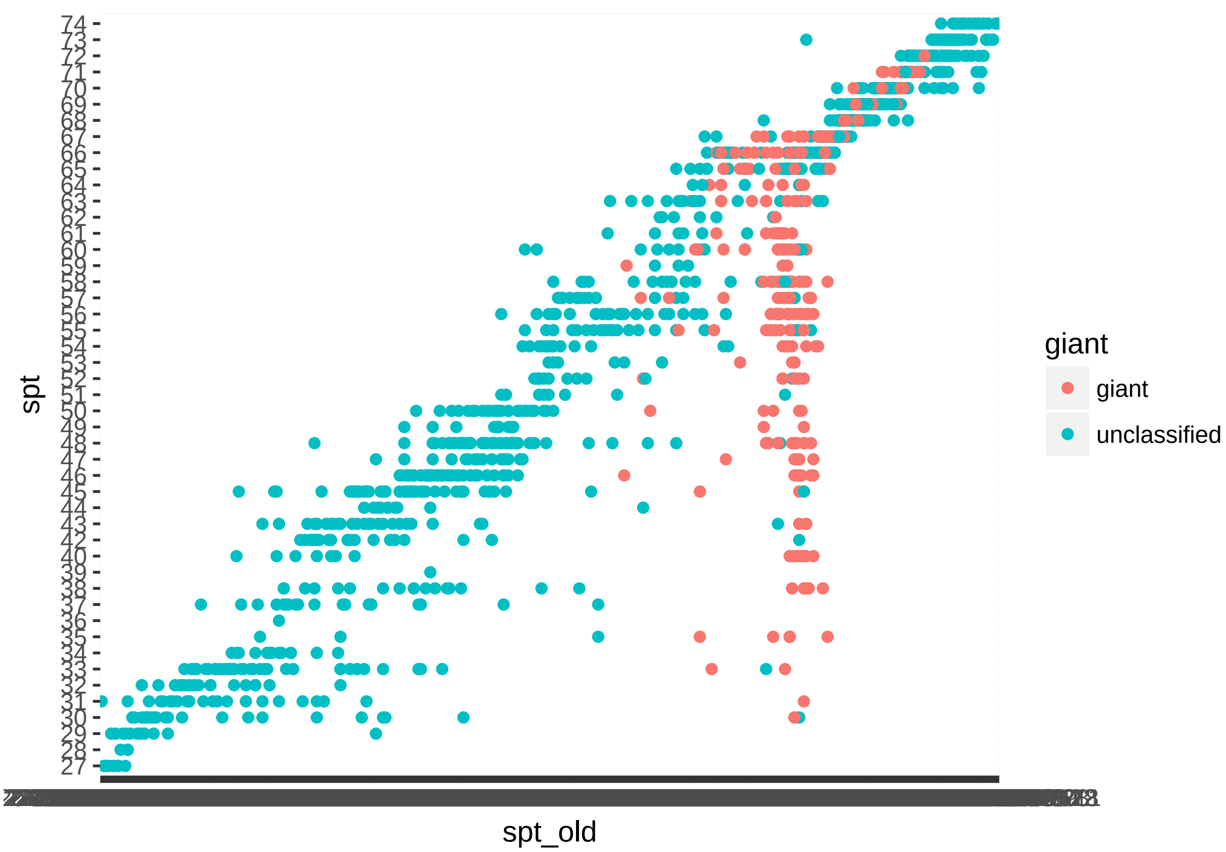
Lets make a column that tells us which stars we classified as probable background giants.

```
full.df <- full.df %>%
    mutate(giant = ifelse(cagiant == "giant" | nagiant == "giant", "giant", "unclassified"))


spt.df <- full.df %>%
    filter(is.na(spt)==FALSE & is.na(spt_old)==FALSE)


head(spt.df)
```

```
## # A tibble: 6 x 93
##      X1    ra   dec  bmag  berr  vmag  verr  imag  ierr  jmag   jerr  hmag
##   <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl>
## 1  4666  343.  62.4    NA    NA    NA    NA    NA    NA  13.8 0.025   12.4
## 2  4760  344.  62.3    NA    NA    NA    NA    NA    NA  13.8 0.025   13.1
## 3  4925  343.  62.4    NA    NA    NA    NA    NA    NA  14.8 0.046   13.7
## 4  5949  343.  62.4    NA    NA    NA    NA    NA    NA  15.3 0.0580  14.2
## 5  6017  344.  62.4    NA    NA    NA    NA    NA    NA  13.6 0.035   13.0
## 6  7049  344.  62.4    NA    NA    NA    NA    NA    NA  13.0 0.023   12.6
## # ... with 81 more variables: herr <dbl>, kmag <dbl>, kerr <dbl>,
## #   c1mag <dbl>, c1err <dbl>, c2mag <dbl>, c2err <dbl>, c3mag <dbl>,
## #   c3err <dbl>, c4mag <dbl>, c4err <dbl>, m24mag <dbl>, m24err <dbl>,
## #   cluster <chr>, cloud <chr>, disk <chr>, xray <chr>, acis <chr>,
## #   spec <chr>, chelle <chr>, spt <chr>, spterr <chr>, tio <chr>,
## #   tior <chr>, cah <chr>, cahr <chr>, spt_old <chr>, spterr_old <chr>,
## #   nagiant <chr>, cagiant <chr>, minxray.ra <int>, minxray.dec <int>,
```

```
## #   minxray.id <chr>, minxray.rcnts <int>, minxray.ncnts <int>,
## #   minxray.npflux <int>, minxray.npfluxerr <int>, minxray.nh <int>,
## #   minxray.nherr <int>, minxray.kt1 <int>, minxray.kt1err <int>,
## #   minxray.aflux <int>, minxray.uflux <int>, minxray.rchi <int>,
## #   medxray.ra <dbl>, medxray.dec <dbl>, medxray.id <chr>,
## #   medxray.rcnts <int>, medxray.ncnts <dbl>, medxray.npflux <dbl>,
## #   medxray.npfluxerr <dbl>, medxray.nh <dbl>, medxray.nherr <dbl>,
## #   medxray.kt1 <int>, medxray.kt1err <int>, medxray.aflux <dbl>,
## #   medxray.uflux <dbl>, medxray.rchi <dbl>, maxxray.ra <dbl>,
## #   maxxray.dec <dbl>, maxxray.id <chr>, maxxray.rcnts <int>,
## #   maxxray.ncnts <dbl>, maxxray.npflux <dbl>, maxxray.npfluxerr <dbl>,
## #   maxxray.nh <dbl>, maxxray.nherr <dbl>, maxxray.kt1 <dbl>,
## #   maxxray.kt1err <dbl>, maxxray.aflux <dbl>, maxxray.uflux <dbl>,
## #   maxxray.rchi <dbl>, lbol.teff.sa.lbol <chr>, lbol.teff.sa.teff <chr>,
## #   lbol.teff.sa.sa <chr>, member <chr>, gmag <dbl>, gerr <dbl>,
## #   rmag <dbl>, rerr <dbl>, giant <chr>
```

```
plot1 <- spt.df %>% ggplot(aes(x=spt_old,y=spt,color=giant)) +
    geom_point()


plot1
```

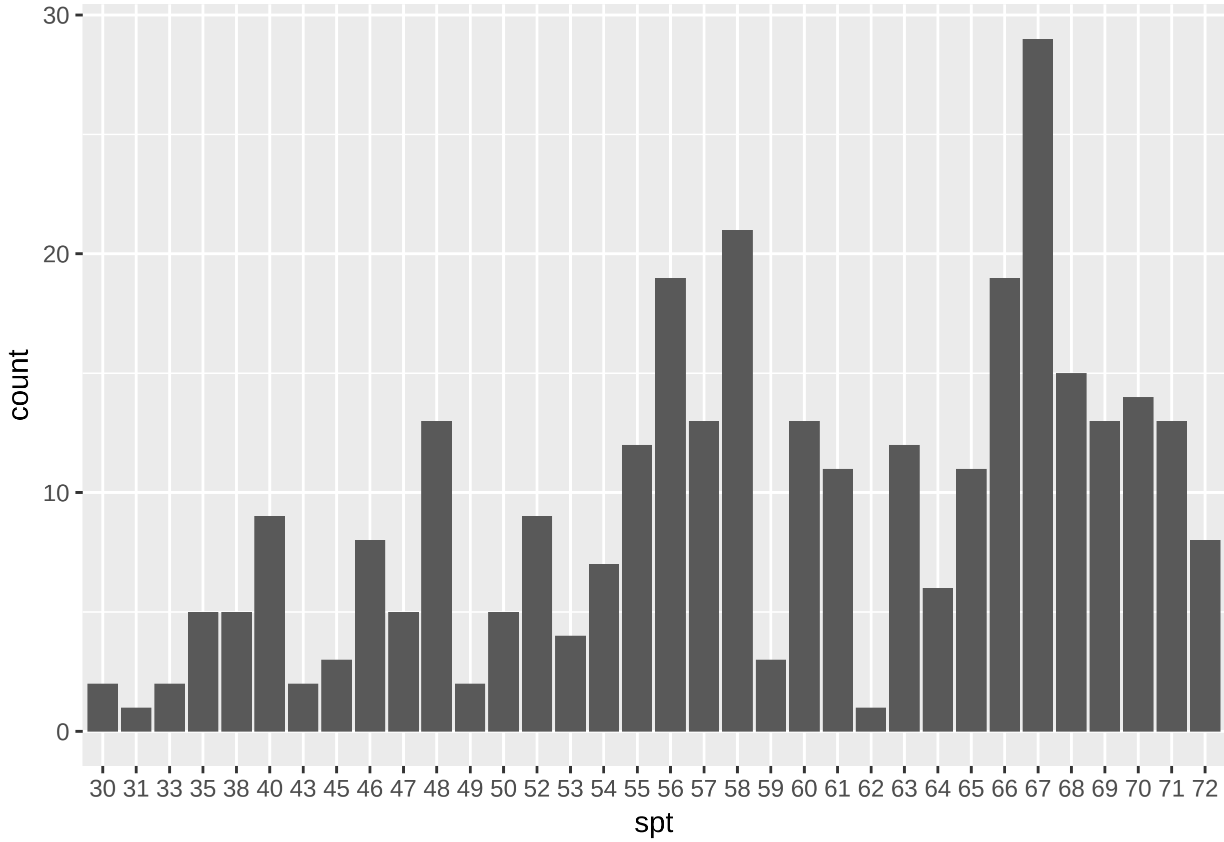


It looks like many of the stars classified as K5/K6 background giants have the wrong spectral classification.

Lets look at the distibution of “correct” spectral types of the “background giants”.

```
spt.df %>%
    filter(giant=="giant") %>%
```

```
ggplot(aes(x=spt)) + geom_bar()
```

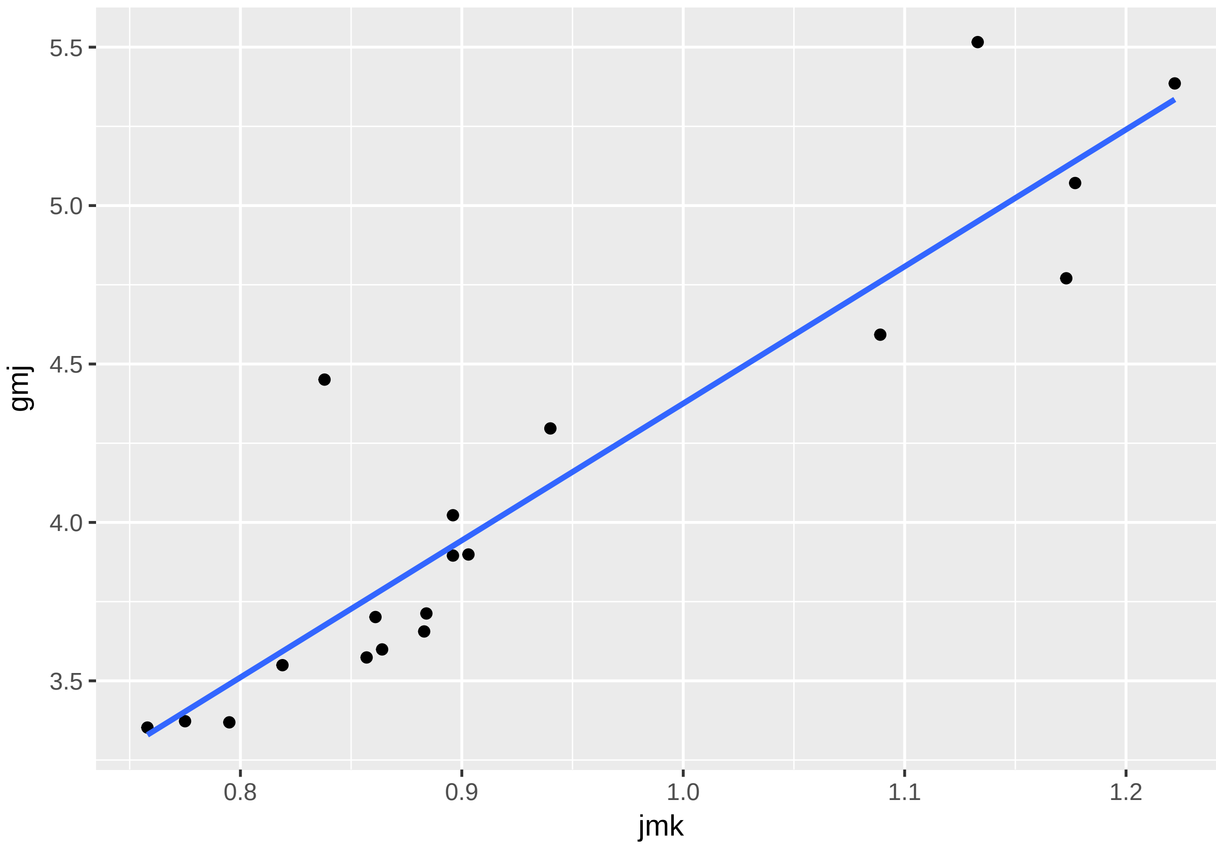


Ok, so close to 30 K5/K6 stars still remain. There is almost 30 K7s.

```
plot_fit <- function(df,c1,c2,c3)  {


    plot <- df %>%
        filter(giant=="giant") %>%
        filter(is.na(!!c2)==FALSE & is.na(!!c3)==FALSE & is.na(!!c1)==FALSE) %>%
        filter(spt==65 | spt==66) %>% # | spt==67)


        #mutate(c1=as.numeric(c1)) %>%
        #mutate(x=c2-c3,y=c1-c2)
    #filter(jmk < 1.75)


#    df %>%
        ggplot(aes(x=!!c2-!!c3, y=!!c1-!!c2)) +
        geom_point()

 #   lm(y ~ x, data = df)
    return(plot)
  #  summary(fit)
}
# g - J
```

```r
#spt.df %>% plot_fit("gmag","jamg","kmag")
```

```r
# g - J

spt.df2 <- spt.df %>%
    filter(giant=="giant") %>%
    filter(is.na(jmag)==FALSE & is.na(kmag)==FALSE & is.na(gmag)==FALSE) %>%
    filter(spt==65 | spt==66)# | spt==67)


spt.df3 <- spt.df2 %>%
    mutate(vmag=as.numeric(vmag)) %>%
    mutate(jmk=jmag-kmag,gmj=gmag-jmag)# %>%
    #filter(jmk < 1.75)

 spt.df3 %>%
    ggplot(aes(x=jmk, y=gmj)) +
    geom_point() + geom_smooth(method = "lm", se=FALSE)
```

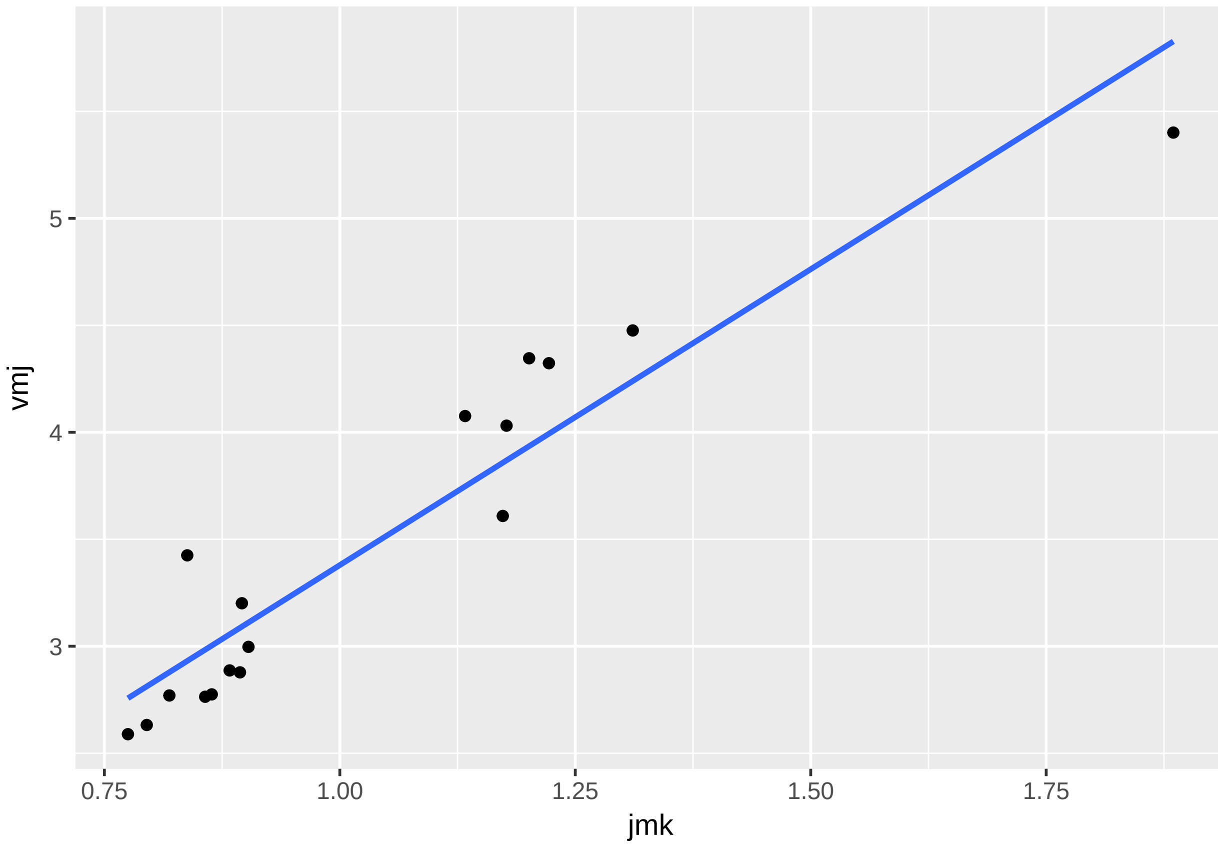



```r
fit <- lm(gmj ~ jmk, data = spt.df3)

summary(fit)
```

```
##
## Call:
## lm(formula = gmj ~ jmk, data = spt.df3)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.35270 -0.16452 -0.05736  0.03684  0.77561 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.05289    0.41965   0.126    0.901    
## jmk          4.32254    0.44374   9.741 2.27e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.2757 on 17 degrees of freedom
## Multiple R-squared:  0.8481, Adjusted R-squared:  0.8391 
## F-statistic: 94.89 on 1 and 17 DF,  p-value: 2.271e-08
```

```
# V - J

spt.df2 <- spt.df %>%
    filter(giant=="giant") %>%
    filter(is.na(jmag)==FALSE & is.na(kmag)==FALSE & is.na(vmag)==FALSE) %>%
    filter(spt==65 | spt==66)# | spt==67)


spt.df3 <- spt.df2 %>%
    mutate(vmag=as.numeric(vmag)) %>%
    mutate(jmk=jmag-kmag,vmj=vmag-jmag)# %>%
    #filter(jmk < 1.75)

 spt.df3 %>%
    ggplot(aes(x=jmk, y=vmj)) +
    geom_point() + geom_smooth(method = "lm", se=FALSE)
```

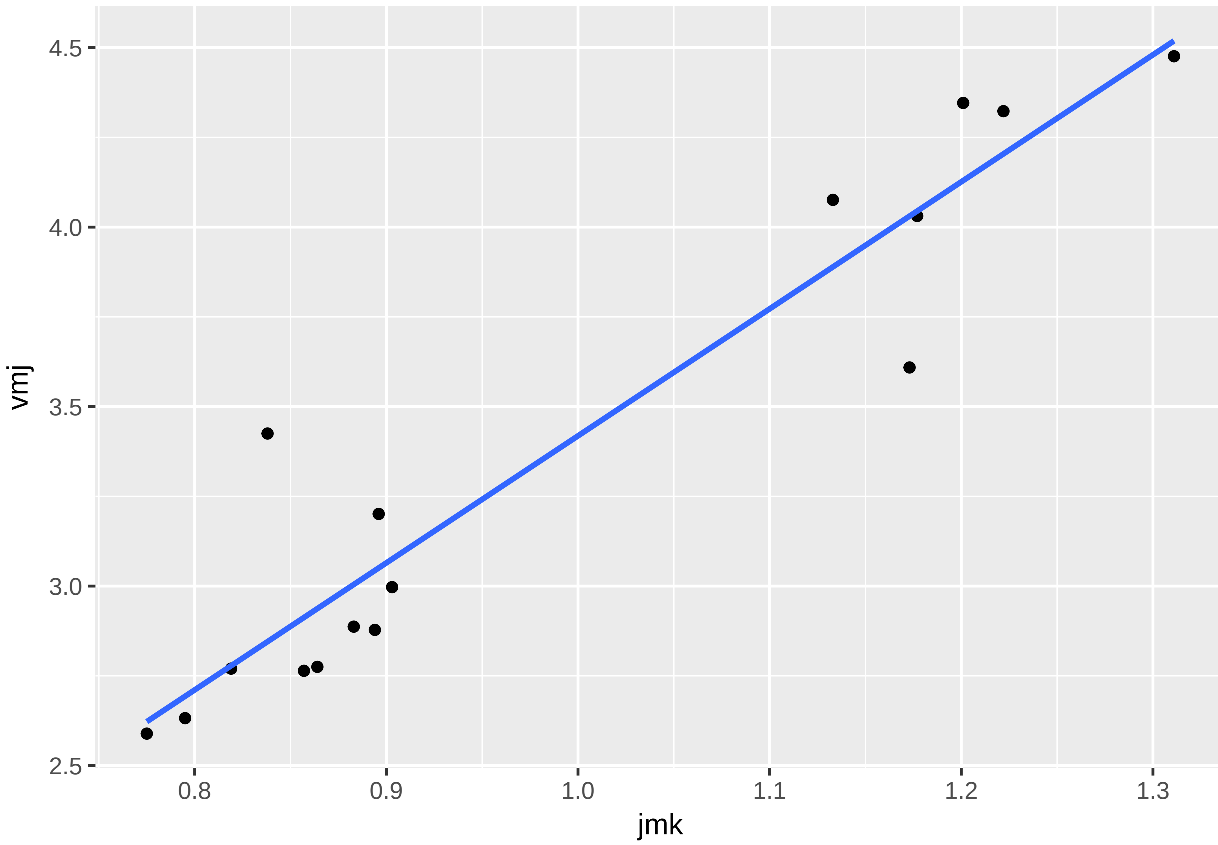



```r
fit <- lm(vmj ~ jmk, data = spt.df3)

summary(fit)
```

```
##
## Call:
## lm(formula = vmj ~ jmk, data = spt.df3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4261 -0.2082 -0.1140  0.2365  0.4937
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.6135     0.2727   2.249   0.0399 *
## jmk           2.7658     0.2544  10.871 1.65e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2857 on 15 degrees of freedom
## Multiple R-squared:  0.8874, Adjusted R-squared:  0.8799
## F-statistic: 118.2 on 1 and 15 DF,  p-value: 1.647e-08
```

```r
# V - J

spt.df2 <- spt.df %>%
    filter(giant=="giant") %>%
    filter(is.na(jmag)==FALSE & is.na(kmag)==FALSE & is.na(vmag)==FALSE) %>%
```

```
    filter(spt==65 | spt==66)# | spt==67)


spt.df3 <- spt.df2 %>%
    mutate(vmag=as.numeric(vmag)) %>%
    mutate(jmk=jmag-kmag,vmj=vmag-jmag) %>%
    filter(jmk < 1.75)

 spt.df3 %>%
    ggplot(aes(x=jmk, y=vmj)) +
    geom_point() + geom_smooth(method = "lm", se=FALSE)
```

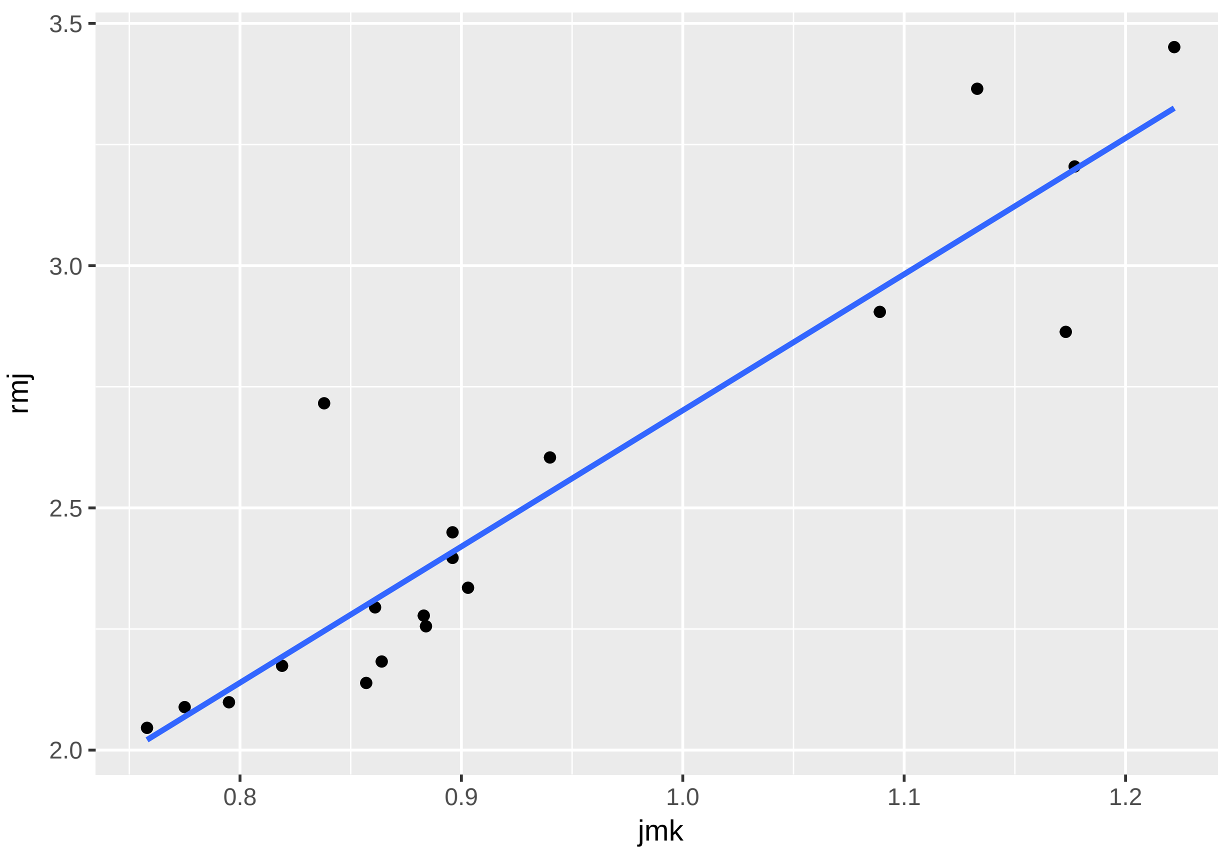



```
fit <- lm(vmj ~ jmk, data = spt.df3)

summary(fit)
```

```
##
## Call:
## lm(formula = vmj ~ jmk, data = spt.df3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42165 -0.12520 -0.03813  0.12686  0.57979
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.1201     0.3249   -0.37    0.717
```

```
## jmk           3.5386     0.3250   10.89 3.23e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2291 on 14 degrees of freedom
## Multiple R-squared:  0.8944, Adjusted R-squared:  0.8868
## F-statistic: 118.5 on 1 and 14 DF,  p-value: 3.23e-08
```

```
# r - J

spt.df2 <- spt.df %>%
    filter(giant=="giant") %>%
    filter(is.na(jmag)==FALSE & is.na(kmag)==FALSE & is.na(rmag)==FALSE) %>%
    filter(spt==65 | spt==66)# | spt==67)


spt.df3 <- spt.df2 %>%
    mutate(rmag=as.numeric(rmag)) %>%
    mutate(jmk=jmag-kmag,rmj=rmag-jmag)# %>%
    #filter(jmk < 1.75)

 spt.df3 %>%
    ggplot(aes(x=jmk, y=rmj)) +
    geom_point() + geom_smooth(method = "lm", se=FALSE)
```




```
fit <- lm(rmj ~ jmk, data = spt.df3)

summary(fit)
```

```
##
## Call:
## lm(formula = rmj ~ jmk, data = spt.df3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.32412 -0.09416 -0.01594  0.03274  0.46994
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.1092     0.2646  -0.413    0.685
## jmk           2.8104     0.2798  10.043 1.45e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1738 on 17 degrees of freedom
## Multiple R-squared:  0.8558, Adjusted R-squared:  0.8473
## F-statistic: 100.9 on 1 and 17 DF,  p-value: 1.454e-08
```

```r
alam <- function(slope) {

    alam <- slope * (1 - 0.397) + 1

    return(alam)

}

alam(3.9)
```

```
## [1] 3.3517
```

```r
alam(3.8)
```

```
## [1] 3.2914
```

```r
alam(3.5)
```

```
## [1] 3.1105
```

```r
alam(2.8)
```

```
## [1] 2.6884
```