

فاز دوم پژوهی استنباط آماری

سحر رجبی - ۸۱۰۱۹۹۱۶۵ (دیتاست StudentPerformance)

• سوال اول:

برای این سوال، از دو متغیر Mjob و romantics استفاده کردیم.

- بخش A :

در این بخش می‌خواهیم بررسی کنیم که آیا سهم یک شغل به عنوان شغل مادران، در بین دانشآموزانی که رماناتیک هستند و آن‌هایی که نیستند، یکسان است یا تفاوت دارد.
برای این کار باید هر بار یکی از شغل‌ها را به عنوان success و مابقی را fail در نظر بگیریم و سهم این شغل را در دو دسته از دانشآموزان romantic و non-romantic محاسبه کنیم.
بعد از محاسبه‌ی تفاضل دو p در دو گروه، SE را برای بدستآوردن بازه‌ی اطمینان محاسبه می‌کنیم و سپس با استفاده از point estimate و se بدستآمد، این بازه را می‌سازیم.
برای محاسبه‌ی SE تفاضل p در دو گروه، ازتابع زیر را تعریف کردیم.

```
se_two_prop <- function(p1, p2, n1, n2){  
  return (((p1*(1-p1)/n1) + (p2*(1-p2)/n2))**0.5)  
}
```

همچنین برای محاسبه‌ی بازه‌ی اطمینان، تابع زیر تعریف شده‌است که برای هر دو توزیع Normal و t-student قابل استفاده است.

```
conf_interval <- function(pe, s, percent, distr="Normal", df=0) {  
  if (distr == "Normal"){  
    z = qnorm((1-percent)/2, lower.tail=FALSE)  
    return (c(pe - z*s, pe + z*s))  
  }  
  else if (distr == 't') {  
    t = qt((1-percent)/2, df=df, lower.tail=FALSE)  
    return (c(pe - t*s, pe + t*s))  
  }  
}
```

با استفاده از این دو تابع و طبق توضیحات قبلی، هر بار یکی از شغل‌ها را در نظر گرفتیم و سایر آن‌ها را صفر که کد این بخش هم در تصویر زیر آورده شده است.

```
# ----- part A
data = students[c('romantic', 'Mjob')]
romantics = data[data['romantic'] == 'yes', ]
non_romantics = data[data['romantic'] == 'no', ]

for (job in unique(data$Mjob)){
  print(job)

  rpos_count = nrow(romantics[romantics$Mjob == job, ])
  rtotal_count = nrow(romantics)

  nrpos_count = nrow(non_romantics[non_romantics$Mjob == job, ])
  nrtotal_count = nrow(non_romantics)

  r_prop = rpos_count / rtotal_count
  nr_prop = nrpos_count / nrtotal_count

  d = r_prop - nr_prop
  print("proportion difference:")
  print(d)
  s = se_two_prop(r_prop, nr_prop, rtotal_count, nrtotal_count)

  ci = conf_interval(d, s, 0.95)
  print('95% confidence interval for proportion(romantics) - proportion(non-romantics) people for this job:')
  print(ci)
}
```

در جدول زیر می‌توانیم تفاوت proportion برای هر شغل و بازه‌ی اطمینان ۹۵ درصدی متناظر آن را ببینیم.

	$P_{romantic} - P_{\sim romantic}$	95% Confidence Interval
Teacher	-0.0385	(-0.1096, 0.0326)
Health	0.0186	(-0.0418, 0.0791)
Services	-0.0389	(-0.1291, 0.0513)
At Home	0.0146	(-0.0609, 0.0901)
Other	0.0442	(-0.0568, 0.01451)

تفسیر بازه‌های به دست آمده این است که ما ۹۵ درصد اطمینان داریم که تفاوت درصد سهم شغل مورد نظر در مادران دانشآموزان احساسی و غیر احساسی، در بازه‌های گزارش شده قرار دارد.

با توجه به اینکه فرض صفر ما در اینجا این است که این تفاوت سهم برابر صفر می‌باشد (و فرض جایگزین صفر نبودن آن)، با توجه به بازه‌های اطمینان بدستآمده و اینکه عدد صفر در تمامی این بازه‌ها موجود است، نمی‌توانیم فرض صفر را رد کنیم و به نظر می‌رسد که رابطه‌ای بین شغل مادران و احساسی بودن دانش‌آموزان وجود ندارد.

- بخش B :

در این بخش با استفاده از تست استقلال با توزیع chi-square بررسی کردیم که آیا این دو متغیر مستقل از یکدیگر هستند یا خیر.

در واقع فرض صفر ما این است که متغیرها مستقل از هم هستند و فرض جایگزین اینکه مستقل نیستند.

برای انجام این آزمون فرض از کدی که در ادامه قابل بررسی است استفاده شده است.

```
# ----- part B
t = ftable(students$romantic ~ students$Mjob)
test_result = chisq.test(t)
```

در اینجا اول یک frequency table ساخته‌ایم که تعداد نفرات دسته‌های ایجاد شده از شغل‌های مختلف مادران و احساسی بودن یا نبودن دانش‌آموزان را استخراج کند و بعد با استفاده از توزیع chi-square آزمون فرض را بررسی کردیم.
نتیجه‌ی این آزمون به این صورت بود:

```
Pearson's Chi-squared test
data: t
X-squared = 2.3571, df = 4, p-value = 0.6704
```

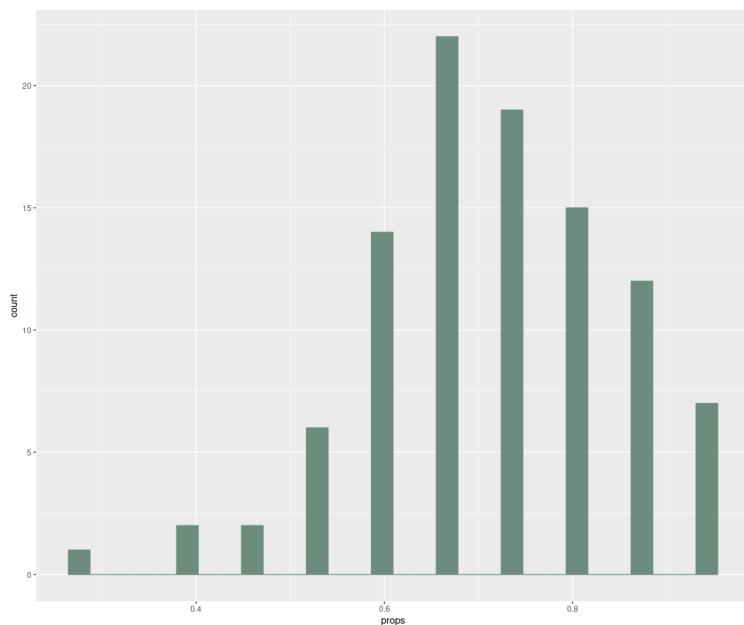
همانطور که مشخص است، p-value این تست مقدار ۰.۶۷۰۴ است که عدد بسیار بزرگی است و ما نمی‌توانیم فرض صفر را -که استقلال این متغیرها از یکدیگر است- رد کنیم.

• سوال دوم:

در این بخش سوال ما از متغیر کتگوریکال باینتری internet استفاده کردیم و می‌خواهیم بررسی کنیم که آیا بیش از ۷۰ درصد دانشآموزان دسترسی به اینترنت دارند یا خیر. برای این کار یک نمونه‌ی ۱۵اتایی از داده‌ها به صورت رندم انتخاب می‌کنیم. و درصد دانشآموزانی که دسترسی به اینترنت دارند را بدست می‌آوریم. سپس ۱۰۰ بار، هر بار یک سمپل ۱۵اتایی از اعداد ۰ و ۱ تولید می‌کنیم که احتمال دیدن عدد ۱، ۰.۷ باشد. نهایتاً می‌توانیم p-value مشاهده‌ی خود از سمپل اصلی را با توجه به نتیجه‌ی این ۱۰۰ سیمولیشن حساب کنیم. کد این بخش به این صورت است:

```
sampled = students[sample(nrow(students), size=15), ]  
sample_prop = length(which(sampled$internet == "yes")) / nrow(sampled)  
  
p = 0.7  
props = c()  
for (i in 1:100){  
  sim_sample = sample(c(0, 1), 15, prob=c(0.3, 0.7), replace=TRUE)  
  pos_prop = length(which(sim_sample == 1)) / length(sim_sample)  
  props = append(props, pos_prop)  
}  
  
p_value = length(which(props >= sample_prop)) / length(props)  
  
ggplot(data.frame(props), aes(x=props)) + geom_histogram(color="#6c8c7e", fill="#6c8c7e")
```

مقدار sample proportion در نمونه‌ی ۱۵اتایی اولیه برابر با ۰.۸۶۷ است و بعد از اجرای سیمولیشن، مقدار گزارش شده برای p-value برابر با ۰.۱۹ است که یعنی نمی‌توانیم فرض صفر را رد کنیم. هیستوگرام proportion‌های مختلف بدست‌آمده در زیر قابل مشاهده است.



• سوال سوم:

برای سوالات این بخش از متغیر شغل مادران استفاده شده است.

- بخش A :

برای بدست آوردن سهم هر یک از این شغل‌ها در این متغیر، از کد زیر استفاده کردیم:

```
probs = c()
for (job in unique(students$Mjob)){
  probs = append(probs, nrow(students[students$Mjob == job, ]) / nrow(students))
}
```

که تعداد هر کدام از شغل‌ها را حساب می‌کند و به کمک آن درصد هر کدام بدست می‌آید.

درصدها در جدول زیر گزارش شده‌اند:

At Home	Health	Services	Teacher	Other
0.1494	0.0861	0.2608	0.1468	0.3570

بعد از این، یک نمونه‌ی ۱۰۰ اتایی به صورت کاملاً رندم از داده‌ها انتخاب می‌کنیم و درصد دسته‌های مختلف این متغیر را در آن‌ها بررسی می‌کنیم. نهایتاً با استفاده از تابع chisq.test آزمون goodness of fit را بر روی این درصدهای جدید اعمال می‌کنیم تا بررسی کنیم توزیع متغیرها در این نمونه، با داده‌ی اصلی برابر است یا خیر. کدهای این قسمت در عکس بعدی آورده شده است.

```
sampled = students[sample(nrow(students), size=100), ]
fs_probs = c()
for (job in unique(students$Mjob)){
  fs_probs = append(fs_probs, nrow(sampled[sampled$Mjob == job, ]) / nrow(sampled))
}
fs_probs_t = tabel_from_vector(fs_probs, unique(students$Mjob))
comparision = chisq.test(fs_probs, p=probs)
```

مقدار p-value بدست‌آمده در این تست برابر ۰.۹ است که مقدار بسیار زیادی است. در نتیجه نمی‌توانیم فرض صفر را که یکسان بودن توزیع این نمونه و داده‌ی اصلی است را نتیجه بگیریم. درصد شغل‌های مختلف در این نمونه، در جدول زیر گزارش شده است.

At Home	Health	Services	Teacher	Other
0.14	0.06	0.26	0.12	0.42

همانطور که از مقایسه‌ی این جدول و جدول قبلی مشخص است، توزیع تا حد بسیار زیادی شبیه به نظر می‌رسد.

برای بایاس سمپل، ما دو روش را انجام دادیم. در روش اول، شانس انتخاب یکی از گروه‌ها را بالا بردیم. فرض کنید در زمان اداری به خانه‌ی دانشآموزان زنگ زده می‌شود و اگر مادر تلفن منزل را پاسخ بدهد از او شغلش را می‌پرسند. تحت این شرایط، شانس انتخاب زنان خانه‌دار بسیار افزایش می‌یابد. با استفاده از کد زیر، ما با ایجاد این بایاس، ۱۰۰ نمونه انتخاب کردیم.

```
selection_prob = transform(students, selection_prob=ifelse(Mjob == 'at_home', 20, 1))$selection_prob
selection_prob = selection_prob / sum(selection_prob)
sampled = students[sample(nrow(students), size=100, prob=selection_prob), ]
ps_probs = c()
for (job in unique(students$Mjob)){
  ps_probs = append(ps_probs, nrow(sampled[sampled$Mjob == job, ]) / nrow(sampled))
}
ps_probs_t = tabel_from_vector(ps_probs, unique(students$Mjob))
comparision = chisq.test(ps_probs, p=probs)
```

توزیع شغل‌ها در نمونه‌ی انتخاب شده به این صورت بود.

At Home	Health	Services	Teacher	Other
0.55	0.04	0.14	0.07	0.20

به صورت چشمی، توزیع این بار با نمونه‌ی قبلی تفاوت زیادی دارد. اگر از تست goodness of fit استفاده کنیم، مقدار p-value برابر ۰.۸۶۷۴ می‌آید که نسبت به دفعه‌ی قبل کاهش داشته، اما باز هم نمی‌توانیم فرض صفر را رد کنیم.

روش دومی که برای بایاس انتخاب کردیم، ایجاد بایاس با استفاده از شغل پدر بود و خواستیم ببرسی کنیم آیا توزیع شغل مادران زمانی که شغل پدران services باشد با توزیع جمعیت یکسان است یا خیر.

با کد زیر این بایاس را ایجاد، و پس از محاسبه‌ی توزیع، تست مورد نظر را اجرا کردیم.

```
filtered_data = students[students$Fjob == 'services', ]
sampled = filtered_data[sample(nrow(filtered_data), size=100), ]
ps_probs = c()
for (job in unique(students$Mjob)){
  ps_probs = append(ps_probs, nrow(sampled[sampled$Mjob == job, ]) / nrow(sampled))
}
ps_probs_t = tabel_from_vector(ps_probs, unique(students$Mjob))
comparision = chisq.test(ps_probs, p=probs)
```

توزیع شغل‌ها در این نمونه هم در جدول بعدی گزارش شده است.

At Home	Health	Services	Teacher	Other
0.13	0.09	0.38	0.18	0.22

این توزیع خیلی متفاوت از توزیع اولیه نیست و نتیجه‌ی آزمون goodness of fit هم بر روی آن به p-value ۰.۹ می‌رسد.

بخش B -

در این بخش ارتباط دسترسی داشتن و یا نداشتن به اینترنت را با شغل مادران مقایسه کردیم. به این صورت که در ابتدای frequency table نشانگر این است که در هر دسته چه تعداد از دانشآموزان قرار می‌گیرند ساختیم و سپس با استفاده از independent test توزیع chi-square استقلال این دو متغیر را چک کردیم.

در اینجا فرض صفر این است که این دو متغیر کاملاً مستقل از یکدیگر هستند و فرض جایگزین اینکه رابطه‌ی بین این دو متغیر کتگوریکال، استقلال نیست و به هم وابسته‌اند. کد زیر نحوه‌ی انجام این کار را نشان می‌دهد.

```
t = ftable(students$internet ~ students$Mjob)
test_result = chisq.test(t)
```

نتایج این تست در زیر خلاصه شده است.

```
Pearson's Chi-squared test
data: t
X-squared = 28.861, df = 4, p-value = 8.341e-06
```

همانطور که مشخص است مقدار گزارش شده برای p-value عدد بسیار کوچکی است. پس ما می‌توانیم فرض صفر را رد کنیم و ادعا کنیم که این متغیرها به یکدیگر وابستگی دارند.

• سوال چهارم:

برای سوالات این بخش، ما نمره‌ی G2 را می‌خواهیم با استفاده از مدل‌ها پیش‌بینی کنیم.

- بخش A:

در فاز اول تمرین، بیشترین correlation با متغیر G2 را نمرات G1 و G3 داشتند و بعد از آن‌ها تعداد failureها، correlation بالایی با این متغیر داشت. از طرف دیگر این نتایج منطقی هم به نظر می‌رسد. چرا که طبق تجربه، دانش‌آموزان عملکرد‌های مشابهی در دروس مختلف دارند و کسی که تعداد افتادن‌های زیادی دارد احتمالاً نمرات کمتری هم در دروس خواهد داشت. همانطور که گفته شد دو متغیر G1 و G3 بیشترین correlation را با نمره‌ی G2 داشتند؛ اما از آنجایی که بررسی این رابطه‌ها احتمالاً بسیار مشابه می‌شد، ما از متغیر سوم failures هم استفاده کردیم و در هر بخش در واقع ۳ مدل فیت کردیم.

- بخش B:

قبل از ذکر گزارش نتایج این بخش، کدهای مربوط به هرکدام را در اینجا می‌آوریم. برای فیت‌کردن سه مدل گفته شده، کد زیر نوشته شده است.

```
fit_g1 = lm(students$G2 ~ students$G1)
summary(fit_g1)
fit_g3 = lm(students$G2 ~ students$G3)
summary(fit_g3)
fit_failures = lm(students$G2 ~ students$failures)
summary(fit_failures)
```

و برای رسم نقاط به همراه خط فیت شده بر آن‌ها، از کدهای آورده شده در زیر استفاده شده.

```
ggplot(students, aes(x=G1, y=G2)) +
  geom_point(color='#6c8c7e') +
  geom_smooth(method='lm', formula= y~x, linetype = "dashed")

ggplot(students, aes(x=G3, y=G2)) +
  geom_point(color='#6c8c7e') +
  geom_smooth(method='lm', formula= y~x, linetype = "dashed")

ggplot(students, aes(x=failures, y=G2)) +
  geom_point(color='#6c8c7e') +
  geom_smooth(method='lm', formula= y~x, linetype = "dashed")
```

به ترتیب سه بخش خواسته شده را برای هر مدل بررسی می‌کنیم.

- مدل پیش‌بینی نمره‌ی G2 بر حسب نمره‌ی G1:

نتایج مدل فیت شده در زیر خلاصه شده است:

```
Call:
lm(formula = students$G2 ~ students$G1)

Residuals:
    Min      1Q  Median      3Q     Max 
-13.4756 -0.7993  0.3754  1.2947  4.5889 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1.62360   0.34880   4.655 4.44e-06 ***  
students$G1  0.98767   0.03075  32.115 < 2e-16 ***  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 2.149 on 393 degrees of freedom
Multiple R-squared:  0.7241, Adjusted R-squared:  0.7234 
F-statistic: 1031 on 1 and 393 DF,  p-value: < 2.2e-16
```

همانطور که مشخص است، به نظر این متغیر می‌تواند تخمین خوبی از ما response variable

بزند و مقدار p-value عدد بسیار کوچکی است در حالی که عدد R² هم قابل قبول است.

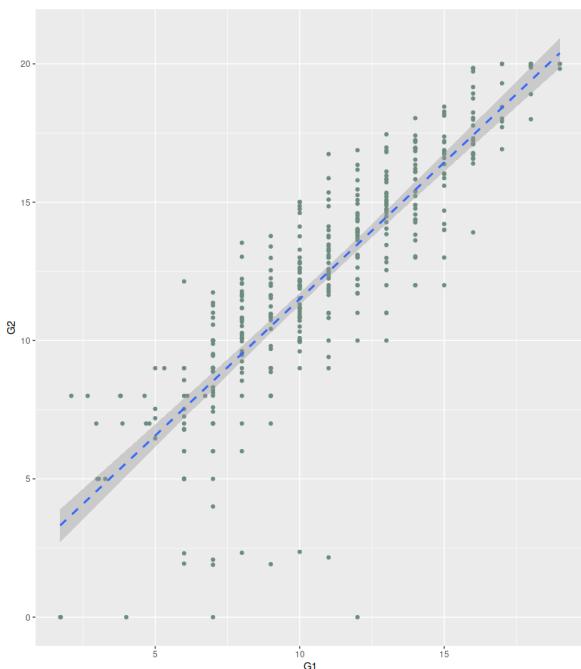
معادله‌ی خط به دست آمده از این مدل، $y = 0.98767*x + 1.6236$ است. معنای slope در اینجا

این است که به ازای گرفتن یک نمره بالاتر در G1، مدل ما پیش‌بینی می‌کند که نمره‌ی G2 به اندازه‌ی

۰.۹۸۷۶۷ افزایش پیدا کند. همچنین عرض از مبدا هم به این معناست که اگر نمره‌ی کسی در G1 صفر

شد، احتمالاً در G2 نمره‌ی ۱.۶۲۳۶ را کسب می‌کند!

رابطه‌ی این دو متغیر در scatter plot زیر مشخص است.



رابطه‌ی آن‌ها هم تقریباً خطی است. هرچند در نمرات بسیار پایین outlierهایی دیده می‌شود.

- مدل پیش‌بینی نمره‌ی G2 بر حسب نمره‌ی G3:

نتایج این مدل به این صورت است:

```

Call:
lm(formula = students$G2 ~ students$G3)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.7383 -0.9028 -0.0993  0.7570  6.9151 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 3.08494   0.22049  13.99 <2e-16 ***
students$G3  0.72690   0.01616  44.98 <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.65 on 393 degrees of freedom
Multiple R-squared:  0.8374, Adjusted R-squared:  0.837 
F-statistic: 2024 on 1 and 393 DF,  p-value: < 2.2e-16

```

p-value در این مدل هم عدد بسیار کوچکی است و مقدار R² از مدل قبلی بیشتر است.

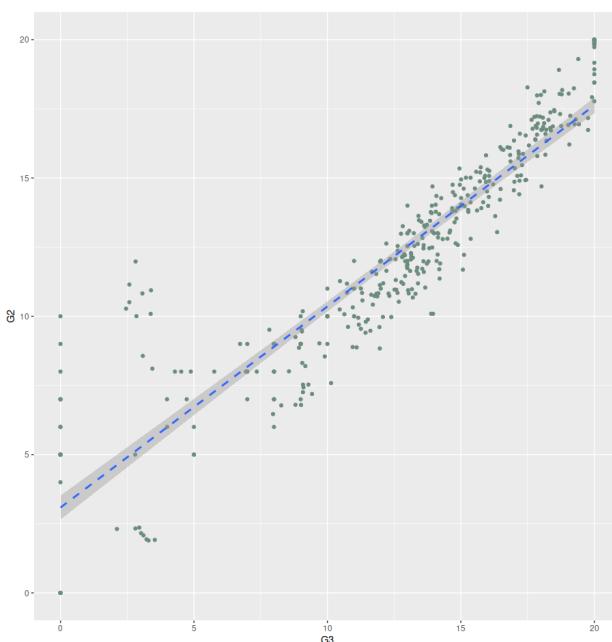
معادله‌ی خط بدست‌آمده از این مدل $y = 0.7269x + 3.08494$ است. معنای شیب خط این مدل این

است که به ازای یک نمره‌ی بیشتر در G3 کسب‌کردن، ۰.۷۲۶۹ افزایش در G2 خواهیم داشت. همچنین

مشابه قبل، معنای عرض از مبدا مدل این است که چنانکه کسی نمره‌ی ۰ در G3 کسب کرد، احتمالاً

نمره‌ی ۳.۰۸۴۹۴ در G2 کسب خواهد کرد.

رابطه‌ی این دو متغیر به همراه خط فیت‌شده در مدل، در شکل زیر مشخص است.



رابطه‌ی خطی این متغیرها قابل مشاهده است؛ اما این بار هم در نمرات پایین outlierهای زیادی داریم که شبیه خط را هم تحت تاثیر قرار داده‌اند. اما همبستگی نقاط و پراکندگی آن‌ها دور خط از متغیر قبلی منسجم‌تر است و پراکندگی کمتری دارد.

- مدل پیش‌بینی نمره‌ی G2 بر حسب تعداد :failure

نتایج مدل را در زیر مشاهده می‌کنید:

```

Call:
lm(formula = students$G2 ~ students$failures)

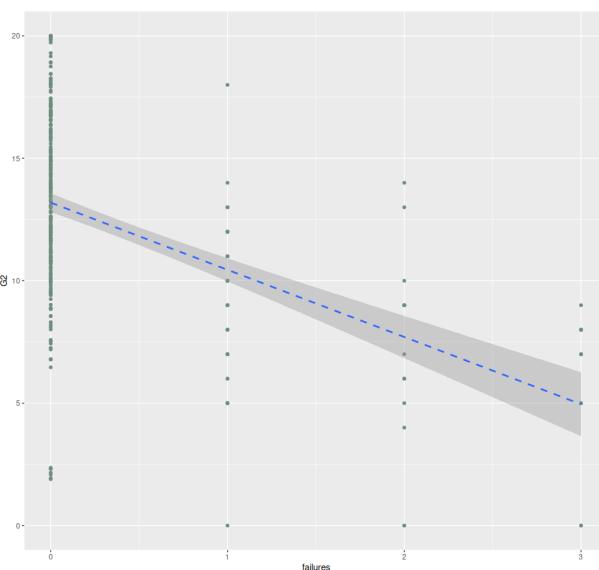
Residuals:
    Min      1Q  Median      3Q     Max 
-11.2962 -2.2293  0.0441  2.5542  7.5542 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 13.1908    0.1956   67.44 <2e-16 ***
students$failures -2.7450    0.2402  -11.43 <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.545 on 393 degrees of freedom
Multiple R-squared:  0.2495,    Adjusted R-squared:  0.2476 
F-statistic: 130.6 on 1 and 393 DF,  p-value: < 2.2e-16

```

مقدار p-value برای این مدل هم بسیار عدد کوچکی است، اما مقدار R² به مرتب کمتر از مدل‌های قبلی است و این متغیر توانایی بسیار کمتری در پیش‌بینی این نقاط نشان داده است. معادله‌ی این خط $y = -2.7450*x + 13.1908$ است و slope در آن به معنای این است که کسی که یک بار بیشتر افتاده باشد، احتمالاً ۰.۷۴۵ نمره کمتر در G2 کسب می‌کند. همچنین عرض از مبدا به این معناست که مدل پیش‌بینی می‌کند که کسی که هیچ failure‌ای ندارد، در این درس نمره‌ی ۱۳.۱۹۰۸ کسب می‌کند.



رابطه‌ی این متغیرها به این صورت است.

به ازای هر failure یک روند کاهشی کلی در نمرات را شاهد هستیم اما رنج تغییرات نمرات حول هر کدام از این نقاط بسیار زیاد است و منطقی است که این مدل R² کوچکی دارد و سهم زیادی از داده‌ها را نتوانسته توجیه کند.

- بخش C :

در قسمت‌های قبل، p-value هر سه متغیر مقدار بسیار کمی بود که با توجه به *t*-value می‌توانیم متوجه شویم که مقدار آن در مدل پیش‌بینی G2 برحسب G3 کمتر از سایرین است. همچنین از نظر معیار R² که بیان می‌کند مدل چه درصدی از variability را توجیه می‌کند، مدل ساخته شده بر اساس G3 موفقیت بیشتری کسب کرده است. در مجموع به نظر می‌رسد که از بین این سه متغیر، بهترین آن‌ها همان G3 است و بهترین توانایی را در پیش‌بینی مدل داشته. جدول زیر نتایج این سه مدل را خلاصه کرده است.

explanatory	R2	T-Value	P-Value
G1	0.7241	32.115	<2e-16
G3	0.8374	44.98	<2.2e-16
Failures	0.2495	-11.43	<2.2e-16

- بخش D :

معیار adjusted-R² بیشترین مقدار را در مدل بر حسب G3 داشت و بعد از آن مدل 1 و کمترین مقدار آن هم مربوط به مدل فیت شده با متغیر failures بود. مقدار آن‌ها به ترتیب نزولی ۰.۷۲۳۴، ۰.۷۴۷۶ و ۰.۸۳۷ است و تفاوت متغیر failures با دو متغیر دیگر بسیار زیاد است. برای مقایسه با جدول ANOVA هم در ابتدا ANOVA table مربوط به هر مدل را که با دستور anova(model) بدست می‌آید خواهیم دید.

Analysis of Variance Table						
Response: students\$G2	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
students\$G1	1	4765.0	4765.0	1031.4	< 2.2e-16	***
Residuals	393	1815.7	4.6			

Signif. codes:	0	***	0.001	**	0.01	*
					0.05	.
					0.1	'
						1

جدول مربوط به G2 ~ G1 :

:G2 ~ G3 جدول مربوط به

```
Analysis of Variance Table

Response: students$G2
           Df Sum Sq Mean Sq F value    Pr(>F)
students$G3     1 5510.5 5510.5 2023.6 < 2.2e-16 ***
Residuals   393 1070.2    2.7
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

:G2 ~ failures جدول مربوط به

```
Analysis of Variance Table

Response: students$G2
           Df Sum Sq Mean Sq F value    Pr(>F)
students$failures  1 1641.8 1641.76 130.64 < 2.2e-16 ***
Residuals       393 4938.9   12.57
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

همانطور که در این تصاویر مشخص است، مقدار p-value گزارش شده توسط این مدل‌ها، به علت اینکه همگی بسیار کوچک هستند، قابل مقایسه نیست. اما بر حسب مقدار F-value می‌توانیم بررسی کنیم که واقعاً p-value کدام یک کوچک‌تر است. طبق این مدل‌ها مقدار p-value برای مدل .failures به علت بزرگ‌تر بودن F-value کوچک‌تر از دو مدل دیگر است، بعد از آن G1 و بعد G3 residual sum of squares هم در مدل ساخته‌شده بر حسب G3 به مراتب کمتر از failures و به طور قابل توجهی کمتر از G1 است.

با توجه به تمامی این مقایسه‌ها، به نظر من رسد که مدل ساخته شده با G3 به عنوان explanatory variable به مراتب بهتر از مدل ساخته‌شده بر حسب failures و بهتر از مدل ساخته‌شده بر حسب G1 است و در این بین مدل failures ضعیفترین عملکرد را داشته است. که یک علت آن این است که این متغیر، یک متغیر گسسته است که باید رنج زیادی از تغییرات را تنها با یک مقدار پوشش‌دهی کند.

- بخش E:

برای اینکه مدل رگرسیون خوبی داشته باشیم، باید ۱- رابطه‌ی متغیرهای explanatory با متغیر response یک رابطه‌ی خطی باشد ۲- توزیع residual‌ها در مدل فیت شده تقریباً نرمال باشد ۳-

در نواحی مختلف تقریباً ثابت باشد ۴- نقاط مختلف داده و *residual variability* آنها مستقل از یکدیگر باشند. اگر ما یک متغیر مناسب داشته باشیم، باید به *R²* و *adjusted R²* بالایی بررسیم و این مدل بتواند بخش زیادی از پراکندگی داده‌ها را جبران کند. برای مثال، متغیر G3 در اینجا مناسب است. همچنین *p-value* یک predictor مناسب، مقدار کوچک‌تری خواهد بود که باز هم این ویژگی را دارد.

- بخش F:

• قسمت a:

در این قسمت ما کد انتخاب نمونه با سایز ۱۰۰، تقسیم داده‌ها به *train* و *test* و نهایتاً فیت کردن سه مدل با سه متغیری که در بخش‌های قبل گفته شد را آورده‌ایم.

```
sampled = students[sample(nrow(students), size=100), ]
train_idx = sample(nrow(sampled), size=90)
train = sampled[train_idx, ]
test = sampled[-train_idx, ]

fit_g1 = lm(train$G2 ~ train$G1)
summary(fit_g1)
fit_g3 = lm(train$G2 ~ train$G3)
summary(fit_g3)
fit_failures = lm(train$G2 ~ train$failures)
summary(fit_failures)
```

فرض صفر برای هر یک از این مدل‌ها این است که ضریب متغیر explanatory برابر صفر می‌باشد و فرض جایگزین، صفر نبودن آن است. در زیر نتیجه‌ی هر کدام از این مدل‌ها را در یک جدول خلاصه، و بررسی می‌کنیم.

explanatory	Coef	T-Value	P-Value
G1	1.0022	19.839	<2e-16
G3	0.65903	18.377	<2e-16
Failures	-3.0824	-6.474	<5.26e-9

هر سه‌ی این متغیرها بیان می‌کند که ما می‌توانیم فرض صفر را رد کنیم! چرا که مقدار آنها

بسیار کوچک است. و نتیجه بگیریم که این متغیرها از نظر آماری significant هستند برای پیش‌بینی

.G2

• قسمت b :

برای محاسبه بازه‌ی اطمینان این ضرایب، از تابعی که در قسمت‌های قبل هم اشاره کردیم، استفاده خواهیم کرد. این تابع مجدداً در اینجا هم آورده شده است.

```
conf_interval <- function(pe, s, percent, distr="Normal", df=0) {  
  if (distr == "Normal"){  
    z = qnorm((1-percent)/2, lower.tail=FALSE)  
    return (c(pe - z*s, pe + z*s))  
  }  
  else if (distr == 't') {  
    t = qt((1-percent)/2, df=df, lower.tail=FALSE)  
    return (c(pe - t*s, pe + t*s))  
  }  
}
```

و کد این قسمت برای محاسبه بازه‌ها هم به این صورت است:

```
g1_ci = conf_interval(coef(summary(fit_g1))["train$G1", "Estimate"],  
                      coef(summary(fit_g1))["train$G1", "Std. Error"],  
                      0.95, distr='t', df=88)  
g3_ci = conf_interval(coef(summary(fit_g3))["train$G3", "Estimate"],  
                      coef(summary(fit_g3))["train$G3", "Std. Error"],  
                      0.95, distr='t', df=88)  
failures_ci = conf_interval(coef(summary(fit_failures))["train$failures", "Estimate"],  
                           coef(summary(fit_failures))["train$failures", "Std. Error"],  
                           0.95, distr='t', df=88)
```

مقدار اصلی بدست‌آمده برای ضریب‌ها و بازه‌ی اطمینان متناظر آن‌ها در جدول زیر خلاصه شده است.

explanatory	coef.	95% Confidence Interval
G1	1.0022	(0.9018, 1.1026)
G3	0.65903	(0.5878, 0.7303)
failures	-3.0824	(-4.0287, -2.1362)

بازه‌های اطمینان بدست‌آمده برای هر ضریب، به این معناست که ما ۹۵ درصد اطمینان داریم که در صورت افزایش یک واحد متغیر explanatory، مقدار تغییر متغیر response در بازه‌ی گزارش شده قرار دارد.

• قسمت C :

توسط تابع predict می‌توانیم مقدار متغیر G2 را با هرکدام از این مدل‌ها، برای داده‌ی test تخمین بزنیم.

```
train = sampled[-train_idx, ]
predicted_g1 = predict(fit_g1, newdata=test)
predicted_g3 = predict(fit_g3, newdata=test)
predicted_failures = predict(fit_failures, newdata=test)
```

کیفیت این تخمین‌ها را در قسمت بعدی بررسی خواهیم کرد.

• قسمت d :

در این بخش برای بررسی دقیق مدل‌ها، ابتدا مقادیر واقعی و مقادیر پیش‌بینی شده توسط مدل‌ها را گزارش می‌کنیم. سپس برای محاسبه‌ی دقیق، در صورتی که نمره‌ی اعلام شده و نمره‌ی اصلی کمتر از یک نمره اختلاف داشتند، پیش‌بینی را درست در نظر می‌گیریم. چرا که در سیستم نمره‌دهی، دو نمره با اختلاف کمتر از یک تفاوت چندانی با یکدیگر ندارد و ما اگر بتوانیم با این دقیق مدل هم پیش‌بینی کنیم، از عملکرد مدل راضی خواهیم بود.

برای مدل G2 ~ G1 :

	predicted_g1	test.G2	true_pred
395	9.456576	10.732965	0
362	14.818177	12.000000	0
14	11.601216	11.908667	1
95	12.673537	15.353065	0
5	7.311935	12.138542	0
351	6.022913	7.000000	1
280	11.601216	12.454330	1
387	7.311935	7.256775	1
53	12.673537	11.000000	0
300	18.035139	17.115479	1

برای مدل G2 ~ G3 :

	predicted_g3	test.G2	true_pred
395	11.726597	10.732965	1
362	11.892533	12.000000	1
14	13.478959	11.908667	0
95	15.470936	15.353065	1
5	12.462960	12.138542	1
351	6.708245	7.000000	1
280	13.033182	12.454330	1
387	9.813298	7.256775	0
53	10.466508	11.000000	1
300	17.099771	17.115479	1

برای مدل G2 ~ failures

	predicted_failures	test\$G2	true_pred
395	13.097692	10.732965	0
362	10.236604	12.000000	0
14	13.097692	11.908667	0
95	13.097692	15.353065	0
5	13.097692	12.138542	1
351	4.514427	7.000000	0
280	13.097692	12.454330	1
387	13.097692	7.256775	0
53	10.236604	11.000000	1
300	13.097692	17.115479	0

همانطور که به نظر می‌رسد دقت مدل ارائه شده بر حسب failures بسیار پایین‌تر از بقیه بوده و علت آن است که این متغیر یک متغیر گسسته است. در نتیجه رنج زیادی از بازه‌ی ما باید توسط یک مقدار پوشش داده شوند.

جدول زیر دقت هر یک از این مدل‌ها بر روی دادگان تست را گزارش کرده است.

explanatory	success rate
G1	0.5
G3	0.8
failures	0.3

جدول بالا هم موید تمامی ادعاهای قبلی مبنی بر برتر بودن G3 و ضعیفتر بودن failures برای پیش‌بینی نمرات G2 است.

کد زده شده برای این بخش هم در تصویر زیر قابل ملاحظه است.

```
df1 = data.frame(predicted_g1, test$G2)
df1 = transform(df1, true_pred=ifelse(abs(predicted_g1-test$G2) < 1, 1, 0))
df3 = data.frame(predicted_g3, test$G2)
df3 = transform(df3, true_pred=ifelse(abs(predicted_g3-test$G2) < 1, 1, 0))
dff = data.frame(predicted_failures, test$G2)
dff = transform(dff, true_pred=ifelse(abs(predicted_failures-test$G2) < 1, 1, 0))

success_rate1 = nrow(df1[df1$true_pred == 1, ])/10
success_rate3 = nrow(df3[df3$true_pred == 1, ])/10
success_ratef = nrow(dff[dff$true_pred == 1, ])/10
```

• سوال پنجم:

متغیر response در این سوال و سوال قبلی نمره‌ی G2 است.

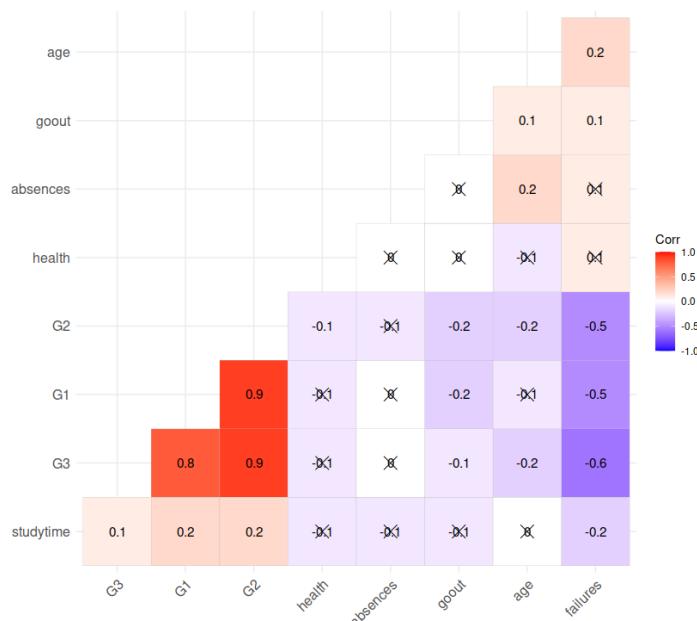
:A - بخش

با استفاده از کد زیر، نمودار correlogram را برای متغیرهای numerical این دیتاست رسم می‌کنیم.

```
library(ggcorrplot)

nomeric_feature = students[, sapply(students, is.numeric)]
nomeric_feature = nomeric_feature[-1]
p.mat <- cor_pmat(nomeric_feature)
corr <- round(corr(nomeric_feature), 1)
ggcorrplot(
  corr, hc.order = TRUE, type = "lower",
  lab = TRUE, p.mat=p.mat
)
```

که این نمودار را در تصویر زیر مشاهده می‌کنید.



همانطور که می‌بینید، بیشترین correlation در بین سه نمره‌ی G1، G2 و G3 قرار دارد و بعد از آن متغیر failures بیشترین correlation را با این سه نمره دارد. مقدار بقیه‌ی predictorها بسیار ناچیز است. در نتیجه به نظر می‌رسد که بهترین predictor برای نمره‌ی G2 سه متغیر G1، G3 و G2 است.

هستند. اما آنها با یکدیگر هم دارای correlation هستند و استفاده‌ی همزمان آنها می‌تواند ایجاد مشکل کند.

بخش B -

با قطعه کد زیر، یک مدل برای پیش‌بینی نمره‌ی G2 توسط سه متغیر G1، G3 و failures طراحی می‌کنیم.

```
mlm = lm(G2 ~ G1 + G3 + failures, data=students)
summary(mlm)
```

که عملکرد آن در زیر به‌طور خلاصه قابل بررسی است.

```
Call:
lm(formula = G2 ~ G1 + G3 + failures, data = students)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.3552 -0.6949 -0.0047  0.6979  5.7639 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1.25454   0.28561   4.392 1.45e-05 ***
G1          0.38290   0.03419  11.198 < 2e-16 ***
G3          0.53839   0.02545  21.154 < 2e-16 ***
failures    0.25293   0.11914   2.123   0.0344 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.433 on 391 degrees of freedom
Multiple R-squared:  0.8779, Adjusted R-squared:  0.877 
F-statistic: 937.5 on 3 and 391 DF,  p-value: < 2.2e-16
```

مقدار p-value‌ی تمامی متغیرها مقداری significant است، اما متغیر failures به مراتب ضعیفتر عمل کرده است.

عملکرد این مدل در بخش‌های بعدی بیشتر بررسی خواهد شد.

بخش C -

مقدار پراکندگی متغیر response که قابل توجیه با مدل باشد توسط مقدار R2 قابل اندازه‌گیری است که در این مدل برابر ۰.۸۷۷۹ است. یعنی این مدل تقریباً ۰.۸۸ پراکندگی را توجیه می‌کند.

- بخش D :

مدل ما موفق شده است که بخش خیلی خوبی از پراکندگی را توجیه کند و مقدار p-value تمامی متغیرهای آن هم significant است. البته شاید به علت correlation بین متغیرهای explanatory این مدل مدل بهینه‌ی ما نباشد، اما عملکرد آن، تا به اینجا مورد قبول است.

- بخش E :

در این بخش باید از دو روش backward elimination و forward selection برای بدست آوردن بهترین مدل استفاده کرد و هم چنین در هر دوی این روش‌ها، با هر دو متریک adj-R2 و p-value انتخاب مدل را صورت داد.

در هر بخش ما بعد از توضیح کد مربوطه، استپ‌های طی شده برای رسیدن به مدل را گزارش می‌کنیم و نهایتاً مدل نهایی را اعلام می‌کنیم.

• روش adj-R2 با متریک forward selection

کد این بخش به صورت زیر است.

```
# adj-R2, forward selection
selected_cols = c()
best_model = NULL
max_adjR2 = 0
formula_ = 'G2 ~ '
available_cols = setdiff(col_names, selected_cols)
while (length(available_cols) > 0){
  pre_adjR2 = max_adjR2
  selected_var = ''
  selected_formula = ''
  for (col in available_cols){
    if (length(selected_cols) > 0){
      added_str = paste('+', col)
      new_formula = paste(formula_, added_str)
    } else{
      new_formula = paste(formula_, col)
    }
    model = lm(new_formula, data=students)
    print(paste('by adding', col, 'adj-R2 is', summary(model)$adj.r.squared))
    if (summary(model)$adj.r.squared > max_adjR2){
      max_adjR2 = summary(model)$adj.r.squared
      selected_var = col
      selected_formula = new_formula
    }
  }
  if (max_adjR2 > pre_adjR2){
    selected_cols = append(selected_cols, selected_var)
    formula_ = selected_formula
    best_model = lm(formula_, data=students)
    print(paste('selected variable in this step is', selected_var))
    print(paste('the new formula is', formula_))
    print('-----')
  } else{
    print(paste('***** no increase in adj-R2, the final formula is', formula_, '*****'))
    break
  }
  available_cols = setdiff(col_names, selected_cols)
}
fs_adjR2_model = best_model
summary(fs_adjR2_model)
```

اول مدل را خالی در نظر می‌گیریم، سپس هر بار یک متغیر را که بیش از سایر متغیرها باعث افزایش adjusted R2 شده است به لیست متغیرهای مدل اضافه می‌کنیم. زمانی که هیچ متغیری از متغیرهای باقی‌مانده موفق به افزایش آن نباشد روند الگوریتم متوقف می‌شود.

در زیر می‌توانیم مراحل انجام این الگوریتم را ببینیم. در هر بار adj-R2-ای که با اضافه کردن هر متغیر بدست می‌آید را چاپ می‌کند، انتخاب نهایی و فرمول مدل نهایی را اعلام می‌کند.

```
by adding age adj-R2 is 0.0334305928740296
by adding goout adj-R2 is 0.0277357696629541
by adding studytime adj-R2 is 0.0214574686235082
by adding failures adj-R2 is 0.247573014457772
by adding health adj-R2 is 0.0076300195552006
by adding absences adj-R2 is 0.000767562652848275
by adding G1 adj-R2 is 0.723390952689711
by adding G3 adj-R2 is 0.836962836516546
selected variable in this step is G3
the new formula is G2 ~ G3
```

1

```
by adding age adj-R2 is 0.875829906406066
by adding goout adj-R2 is 0.876172559371396
by adding studytime adj-R2 is 0.875742543658399
by adding failures adj-R2 is 0.877011250503318
by adding health adj-R2 is 0.876652778658784
by adding absences adj-R2 is 0.877996453968559
selected variable in this step is absences
the new formula is G2 ~ G3 + G1 + absences
```

3

```
by adding age adj-R2 is 0.879247330833652
by adding goout adj-R2 is 0.879679449472182
by adding studytime adj-R2 is 0.879312344706837
by adding health adj-R2 is 0.880374385311923
selected variable in this step is health
the new formula is
G2 ~ G3 + G1 + absences + failures + health
```

5

```
by adding age adj-R2 is 0.836823079458681
by adding goout adj-R2 is 0.838020049776618
by adding studytime adj-R2 is 0.837875937205102
by adding failures adj-R2 is 0.83798282389614
by adding health adj-R2 is 0.837981096637005
by adding absences adj-R2 is 0.839688908927708
by adding G1 adj-R2 is 0.875910916776897
selected variable in this step is G1
the new formula is G2 ~ G3 + G1
```

2

```
by adding age adj-R2 is 0.877723203353878
by adding goout adj-R2 is 0.878169738010515
by adding studytime adj-R2 is 0.877770824785037
by adding failures adj-R2 is 0.879414809662745
by adding health adj-R2 is 0.878848617807876
selected variable in this step is failures
the new formula is
G2 ~ G3 + G1 + absences + failures
```

4

```
by adding age adj-R2 is 0.880283829972921
by adding goout adj-R2 is 0.880681526141755
by adding studytime adj-R2 is 0.880214955955422
selected variable in this step is goout
the new formula is
G2 ~ G3+G1+absences+failures+health+goout
```

6

```
by adding age adj-R2 is 0.88052692114214
by adding studytime adj-R2 is 0.880504398801303
no increase in adj-R2, the final formula is
G2 ~ G3 + G1 + absences + failures + health + goout
```

7

همانطور که مشخص است، مدل نهایی در این روش بر حسب متغیرهای G1، G3، absenses و goout ساخته شده و دارای بیشترین adj-R2 بوده است.

• روشن p-value با متريک forward selection

کد اين روش در زير آورده شده است.

```
# p-value, forward selection
selected_cols = c()
best_model = NULL
formula_ = 'G2 ~'
available_cols = setdiff(col_names, selected_cols)
while (length(available_cols) > 0){
  min_pvalue = 1
  selected_var = ''
  selected_formula = ''
  for (col in available_cols){
    if (length(selected_cols) > 0){
      added_str = paste('+', col)
      new_formula = paste(formula_, added_str)
    } else{
      new_formula = paste(formula_, col)
    }
    model = lm(new_formula, data=students)
    print(paste('p-value of added variable', col, 'is', coef(summary(model))[col, 'Pr(>|t|)']))
    if (coef(summary(model))[col, 'Pr(>|t|)'] < min_pvalue){
      min_pvalue = coef(summary(model))[col, 'Pr(>|t|)']
      selected_var = col
      selected_formula = new_formula
    }
  }
  if (min_pvalue < 0.05){
    selected_cols = append(selected_cols, selected_var)
    formula_ = selected_formula
    best_model = lm(formula_, data=students)
    print(paste('selected variable in this step is', selected_var))
    print(paste('the new formula is', formula_))
    print('-----')
  } else{
    print(paste('***** no more significant p-value, the final formula is', formula_, '*****'))
    break
  }
  available_cols = setdiff(col_names, selected_cols)
}
fs_pvalue_model = best_model
summary(best_model)
```

در اين روش هم در ابتدا يك مدل خالي داريم و در هر بار متغيری که در صورت اضافه شدن به مدل کمترین مقدار p-value را دارد، به مدل اضافه می‌کنیم. زمانی که هیچ p-value significant نتوانیم اضافه کنیم، الگوریتم متوقف می‌شود.

مراحل کار اين الگوریتم در زير گزارش شده است.

```
p-value of variable age is 0.000152314459040324
p-value of variable goout is 0.000521375410536271
p-value of variable studytime is 0.00204219387616633
p-value of variable failures is 2.57677198675204e-26
p-value of variable health is 0.0453996051720094
p-value of variable absences is 0.254424347044156
p-value of variable G1 is 6.08169729954587e-112
p-value of variable G3 is 4.36758566267675e-157
selected variable in this step is G3
the new formula is G2 ~ G3
```

1

```
p-value of variable age is 0.415854917055342
p-value of variable goout is 0.0597466042243362
p-value of variable studytime is 0.0738076221433051
p-value of variable failures is 0.0630820593929061
p-value of variable health is 0.0632415409290773
p-value of variable absences is 0.00584033300454316
p-value of variable G1 is 2.86743277247649e-25
selected variable in this step is G1
the new formula is G2 ~ G3 + G1
```

2

```

p-value of variable age is 0.388830549350702
p-value of variable goout is 0.177112293254907
p-value of variable studytime is 0.49393317428649
p-value of variable failures is 0.0343826257099999
p-value of variable health is 0.0676537386915824
p-value of variable absences is 0.005784332389537
selected variable in this step is absences
the new formula is G2 ~ G3 + G1 + absences

```

3

```

p-value of variable age is 0.722558507978771
p-value of variable goout is 0.212980582384185
p-value of variable studytime is 0.598161795230673
p-value of variable failures is 0.0184582864874115
p-value of variable health is 0.0535218113067928
selected variable in this step is failures
the new formula is G2 ~ G3 + G1 + absences + failures

```

4

```

p-value of variable age is 0.498453961693976
p-value of variable goout is 0.173668745299899
p-value of variable studytime is 0.413940818581565
p-value of variable health is 0.0428487955424145
selected variable in this step is health
the new formula is
G2 ~ G3 + G1 + absences + failures + health

```

5

```

p-value of variable age is 0.401374419048328
p-value of variable goout is 0.15796271701341
p-value of variable studytime is 0.487817937262923
no more significant p-value, the final formula is
G2 ~ G3 + G1 + absences + failures + health

```

6

مدل نهایی بدستآمده با این روش شامل متغیرهای G1، G3، health و absenses است و

نسبت به متغیر قبلی، یک متغیر کمتر دارد.

روش backward elimination با adj-R2 :

کد این قسمت در تصویر زیر قابل بررسی است.

```

selected_cols = c()
best_model = NULL
formula_ = 'G2 ~ age + goout + studytime + failures + health + absences + G1 + G3'
max_adjR2 = summary(lm(formula_, data=students))$adj.r.squared
print(paste('full model adj-R2 is', max_adjR2))
available_cols = col_names
while (length(available_cols) > 0){
  pre_adjR2 = max_adjR2
  selected_var = ''
  selected_formula = ''
  for (col in available_cols){
    new_formula = make_formula(setdiff(available_cols, c(col)))
    model = lm(new_formula, data=students)
    print(paste('by removing', col, 'adj-R2 is', summary(model)$adj.r.squared))
    if (summary(model)$adj.r.squared > max_adjR2){
      max_adjR2 = summary(model)$adj.r.squared
      selected_var = col
      selected_formula = new_formula
    }
  }
  if (max_adjR2 > pre_adjR2){
    available_cols = setdiff(available_cols, c(selected_var))
    formula_ = selected_formula
    best_model = lm(formula_, data=students)
    print(paste('removed variable in this step is', selected_var))
    print(paste('the new formula is', formula_))
    print('-----')
  } else{
    print(paste('***** no increase in adj-R2, the final formula is', formula_, '*****'))
    break
  }
}
be_adjR2_model = best_model
summary(best_model)

```

در این روش، ما از مدلی شامل تمامی متغیرها شروع می‌کنیم و هر بار سعی می‌کنیم متغیری که حذف آن باعث بیشترین افزایش adj-R2 می‌شود را پیدا کنیم و حذف کنیم. زمانی که دیگر این متغیر وجود نداشت، الگوریتم به اتمام رسیده است.

برای ساخت فرمول هر مدل، از تابع زیر استفاده می‌کنیم که لیستی از نام متغیرهای لازم را می‌گیرد و فرمول معادل آن را برミگرداند.

```
make_formula = function(columns){
  base = 'G2 ~ '
  for (i in 1:length(columns)){
    if (i == length(columns)){
      base = paste(base, columns[i])
    } else{
      base = paste(base, columns[i], '+')
    }
  }
  return (base)
}
```

نتیجه‌ی اجرای این الگوریتم، در زیر قابل مشاهده است.

```
by removing age adj-R2 is 0.880504398801303
by removing goout adj-R2 is 0.880134690305068
by removing studytime adj-R2 is 0.88052692114214
by removing failures adj-R2 is 0.878514628098547
by removing health adj-R2 is 0.879349304930627
by removing absences adj-R2 is 0.878282378990325
by removing G1 adj-R2 is 0.844240717194549
by removing G3 adj-R2 is 0.744760386149535
removed variable in this step is studytime
the new formula is
G2 ~ age + goout + failures + health +
absences + G1 + G3
```

1

```
by removing age adj-R2 is 0.737612001385686
by removing goout adj-R2 is 0.744974620746261
by removing studytime adj-R2 is 0.745333748465889
by removing failures adj-R2 is 0.73690469125719
by removing health adj-R2 is 0.743428158372841
by removing absences adj-R2 is 0.745122840871797
by removing G1 adj-R2 is 0.263990403468682
no increase in adj-R2, the final formula is
G2 ~ age + goout + failures + health + absences + G1
+ G3
```

2

این الگوریتم تنها یک متغیر-study time را حذف می‌کند! و بعد از آن هیچ متغیری را از مدل حذف نمی‌کند.

- روش p-value با متريک back elimination

در تصویر بعدی می‌توانيم کد استفاده شده برای اين قسمت را بررسی کنيد.

```

# p-value, back elimination
selected_cols = c()
best_model = NULL
formula_ = 'G2 ~ age + goout + studytime + failures + health + absences + G1 + G3'
available_cols = col.names
while (length(available_cols) > 0){
  max_pvalue = 0
  selected_var = ''
  model = lm(formula_, data=students)
  for (col in available_cols){
    print(paste('variable', col, 'has p-value', coef(summary(model))[col, 'Pr(>|t|)']))
    if (coef(summary(model))[col, 'Pr(>|t|)'] > max_pvalue){
      max_pvalue = coef(summary(model))[col, 'Pr(>|t|)']
      selected_var = col
    }
  }
  if (max_pvalue > 0.05){
    available_cols = setdiff(available_cols, c(selected_var))
    formula_ = make_formula(available_cols)
    best_model = lm(formula_, data=students)
    print(paste('removed variable in this step is', selected_var))
    print(paste('the new formula is', formula_))
    print('-----')
  } else{
    print(paste('***** no more non-significant variables, the final formula is', formula_, '*****'))
    break
  }
}
be_pvalue_model = best_model
summary(best_model)

```

در اینجا هر بار متغیر با بیشترین p-value که هم نباشد حذف می‌کنیم تا دیگر متغیر non-significant در مدل باقی نماند.

روند انجام این الگوریتم در باکس‌های زیر گزارش شده است.

variable age has p-value 0.46782134272235
 variable goout has p-value 0.189953319036621
 variable studytime has p-value 0.500228353758538
 variable failures has p-value 0.00865146507669938
 variable health has p-value 0.0395838733093082
 variable absences has p-value 0.0057415613319855
 variable G1 has p-value 3.89930700747348e-24
 variable G3 has p-value 1.04128855555366e-65
removed variable in this step is studytime
the new formula is G2 ~ age + goout + failures + health + absences + G1 + G3

1

variable age has p-value 0.480846259152316
 variable goout has p-value 0.181777382236851
 variable failures has p-value 0.0102789977021952
 variable health has p-value 0.035351863883167
 variable absences has p-value 0.00501947534694885
 variable G1 has p-value 1.08276502970888e-24
 variable G3 has p-value 8.38041345973717e-66
removed variable in this step is age
the new formula is G2 ~ goout + failures + health + absences + G1 + G3

2

variable goout has p-value 0.157962717013411
 variable failures has p-value 0.0124995589959419
 variable health has p-value 0.039524030784949
 variable absences has p-value 0.00294504062915305
 variable G1 has p-value 6.66254579296827e-25
 variable G3 has p-value 1.94214043088336e-68
removed variable in this step is goout
the new formula is G2 ~ failures + health + absences + G1 + G3

3

variable failures has p-value 0.0149593807563569
 variable health has p-value 0.0428487955424133
 variable absences has p-value 0.00249268567708328
 variable G1 has p-value 2.4285913044675e-25
 variable G3 has p-value 1.85613805808044e-68
no more non-significant variables, the final formula is
G2 ~ failures + health + absences + G1 + G3

4

مدل بدست آمده با این روش، دارای چهار متغیر $G1$ ، $G3$ ، $health$ و $absenses$ است و مابقی متغیرها را به علت $p\text{-value}$ ناکافی، حذف کرده است.

برای مقایسه این مدل‌ها با یکدیگر، اطلاعات آن‌ها را در کنار هم بررسی می‌کنیم.

Coefficients:						
	Estimate	Std. Error	t value	Pr(> t)		
(Intercept)	2.10795	0.41576	5.070	6.17e-07 ***		
$G3$	0.54193	0.02511	21.582	< 2e-16 ***		
$G1$	0.37387	0.03380	11.060	< 2e-16 ***		
$absences$	-0.02672	0.00893	-2.992	0.00295 **		
$failures$	0.29584	0.11789	2.509	0.01250 *		
$health$	-0.10610	0.05136	-2.066	0.03952 *		
$goout$	-0.09183	0.06491	-1.415	0.15796		

Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'	0.1 ' '	1
Residual standard error:	1.412	on 388 degrees of freedom				
Multiple R-squared:	0.8825		Adjusted R-squared:	0.8807		
F-statistic:	485.7	on 6 and 388 DF,	p-value:	< 2.2e-16		

روش adj-R2 با forward selection

Coefficients:						
	Estimate	Std. Error	t value	Pr(> t)		
(Intercept)	1.779037	0.345116	5.155	4.05e-07 ***		
$G3$	0.542386	0.025140	21.575	< 2e-16 ***		
$G1$	0.377318	0.033758	11.177	< 2e-16 ***		
$absences$	-0.027200	0.008935	-3.044	0.00249 **		
$failures$	0.288225	0.117920	2.444	0.01496 *		
$health$	-0.104470	0.051416	-2.032	0.04285 *		

Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'	0.1 ' '	1
Residual standard error:	1.414	on 389 degrees of freedom				
Multiple R-squared:	0.8819		Adjusted R-squared:	0.8804		
F-statistic:	580.9	on 5 and 389 DF,	p-value:	< 2.2e-16		

روش p-value با forward selection

Coefficients:						
	Estimate	Std. Error	t value	Pr(> t)		
(Intercept)	2.801234	1.066959	2.625	0.00900 **		
age	-0.042357	0.060028	-0.706	0.48085		
$goout$	-0.087309	0.065268	-1.338	0.18178		
$failures$	0.306911	0.119007	2.579	0.01028 *		
$health$	-0.108845	0.051544	-2.112	0.03535 *		
$absences$	-0.025608	0.009075	-2.822	0.00502 **		
$G1$	0.378136	0.034360	11.005	< 2e-16 ***		
$G3$	0.538259	0.025659	20.977	< 2e-16 ***		

Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'	0.1 ' '	1
Residual standard error:	1.413	on 387 degrees of freedom				
Multiple R-squared:	0.8826		Adjusted R-squared:	0.8805		
F-statistic:	415.8	on 7 and 387 DF,	p-value:	< 2.2e-16		

روش adj-R2 با backward elimination

Coefficients:						
	Estimate	Std. Error	t value	Pr(> t)		
(Intercept)	1.779037	0.345116	5.155	4.05e-07 ***		
$failures$	0.288225	0.117920	2.444	0.01496 *		
$health$	-0.104470	0.051416	-2.032	0.04285 *		
$absences$	-0.027200	0.008935	-3.044	0.00249 **		
$G1$	0.377318	0.033758	11.177	< 2e-16 ***		
$G3$	0.542386	0.025140	21.575	< 2e-16 ***		

Signif. codes:	0 '***'	0.001 '**'	0.01 '*'	0.05 '.'	0.1 ' '	1
Residual standard error:	1.414	on 389 degrees of freedom				
Multiple R-squared:	0.8819		Adjusted R-squared:	0.8804		
F-statistic:	580.9	on 5 and 389 DF,	p-value:	< 2.2e-16		

روش p-value با backward elimination

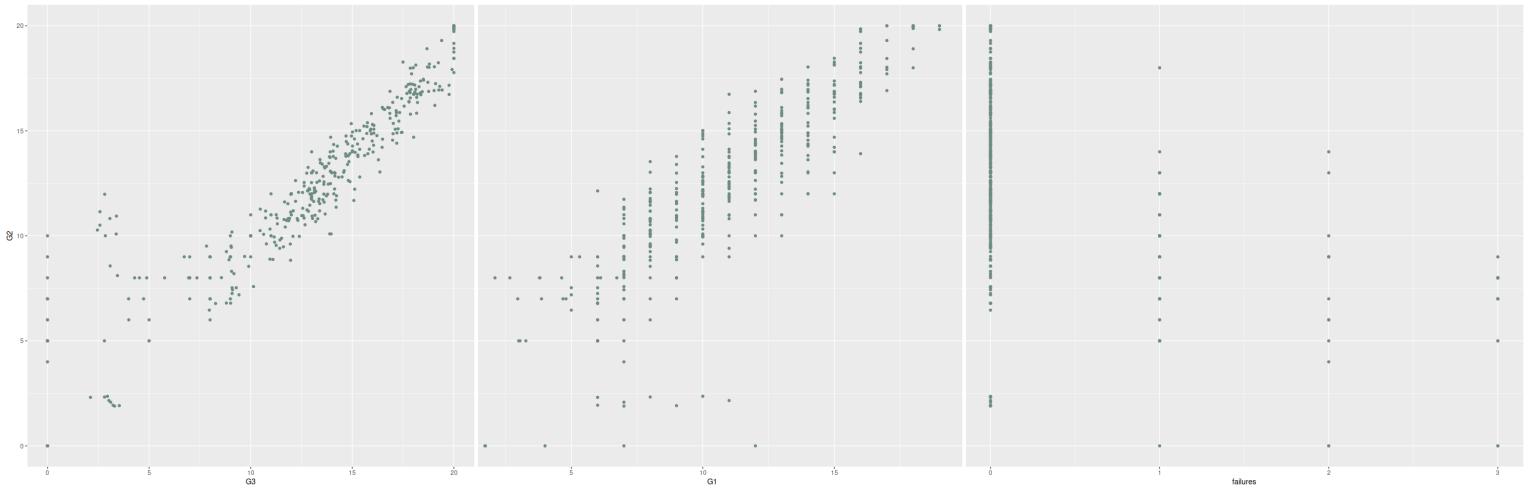
در تمامی این مدل‌ها، مقدار adjusted R2 بسیار نزدیک به یکدیگر است و می‌توانیم تصور کنیم که تفاوتی نداریم. از انجایی که هرچه متغیرهای استفاده شده در یک مدل کمتر و ساده‌تر باشد، مدل مطلوب‌تر است، ما مدلی که متغیرهای $health$ ، $failures$ ، $absenses$ و $G1$ دارد را به عنوان بهترین مدل در نظر می‌گیریم. (پ‌های متغیرهای این مدل هم تماماً significant هستند.)

- بخش F :

ابتدا این شرایط را برای مدل به دست آمده در بخش B بررسی می‌کنیم.

- خطی بودن:

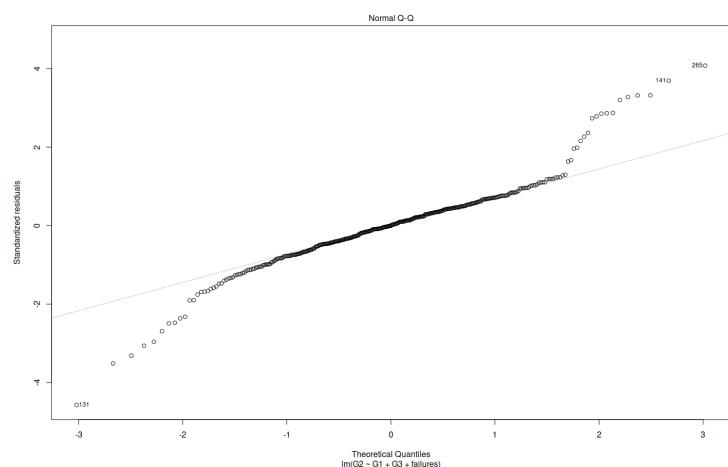
باید رابطه‌ی predictorها و متغیر response، رابطه‌ای نسبتاً خطی باشد. در بخش B از سه متغیر G1، G2 و failures استفاده شده است، با رسم متغیر G2 بر حسب هر کدام، خطی بودن رابطه‌ی آن‌ها را بررسی می‌کنیم.



رابطه‌ی این سه متغیر با متغیر G2 نسبتاً خطی است. حتی در متغیر failure که پراکندگی زیاد است، اما میانگین این نمره با افزایش تعداد failure به صورت خطی کاهش یافته است.

- شرط nearly normal residuals

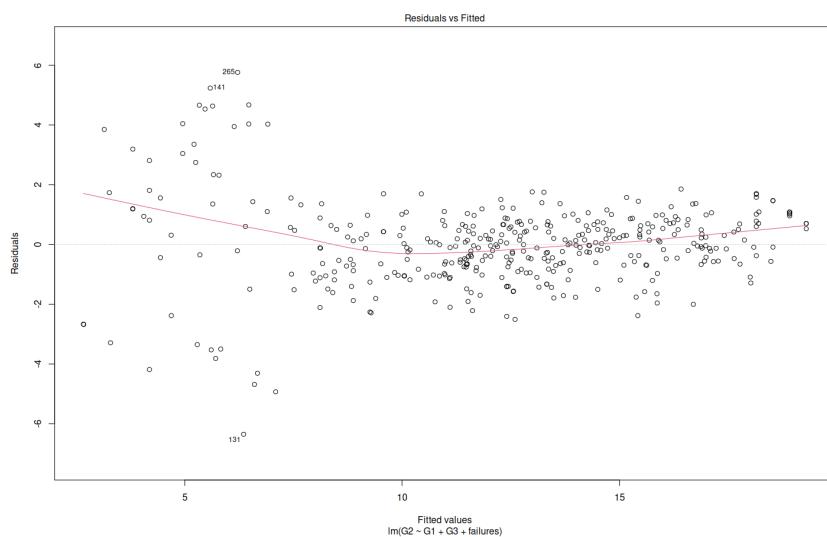
با رسم نمودار Q-Q برای residual‌ها، شرط nearly normal بودن آن‌ها را بررسی می‌کنیم.



با توجه به این نمودار، به نظر می‌رسد توزیع residual‌ها یک توزیع long tail است که در ابتدا نقاط زیر خط قرار دارند؛ در نقاط میانی منطبق است و در انتهای بالای خط.

- شرط constant variability

با رسم نمودار residual‌ها بر حسب مقادیر پیش‌بینی شده، شرط constant variability را چک می‌کنیم.



به‌وضوح، در نمرات residual variability اکثر outlier‌ها در آن قسمت وجود دارند) مقدار بیشتری دارد و residual‌ها رنج بیشتری را نسبت به نمرات بالاتر پوشش داده‌اند. کد استفاده شده برای رسم این پلات‌ها را در ادامه آورده‌ایم.

```
#model in part B
ggplot(students, aes(x=G3, y=G2)) +
  geom_point(color='#6c8c7e')

ggplot(students, aes(x=G1, y=G2)) +
  geom_point(color='#6c8c7e')

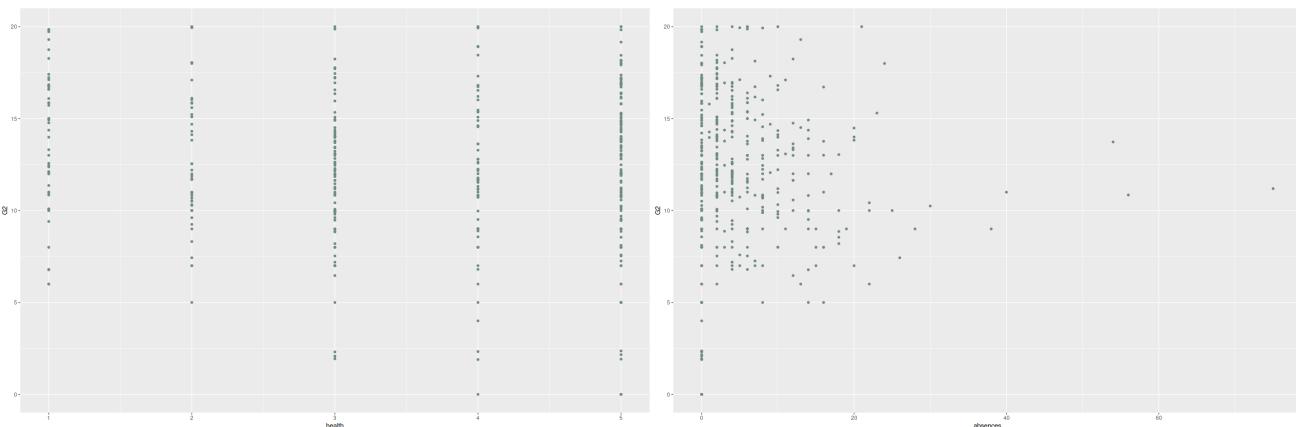
ggplot(students, aes(x=failures, y=G2)) +
  geom_point(color='#6c8c7e')

plot(mlm)
```

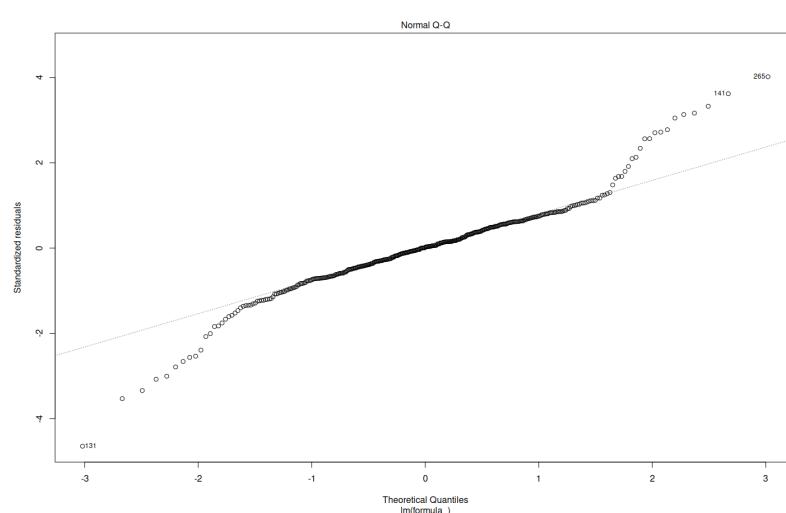
در ادامه همین شرط را برای مدل ارائه شده در بخش E بررسی می‌کنیم و نهایتاً این دو مدل را با هم مقایسه خواهیم کرد.

- خطی بودن:

در این مدل سه متغیر مدل قبل حضور دارند، به اضافه‌ی متغیرهای health و absenses. خطی بودن رابطه G2 و سه متغیر G1، G3 و failures را که در بخش قبلی بررسی کردیم؛ این دو متغیر هم در ادامه می‌توانید مشاهده کنید.

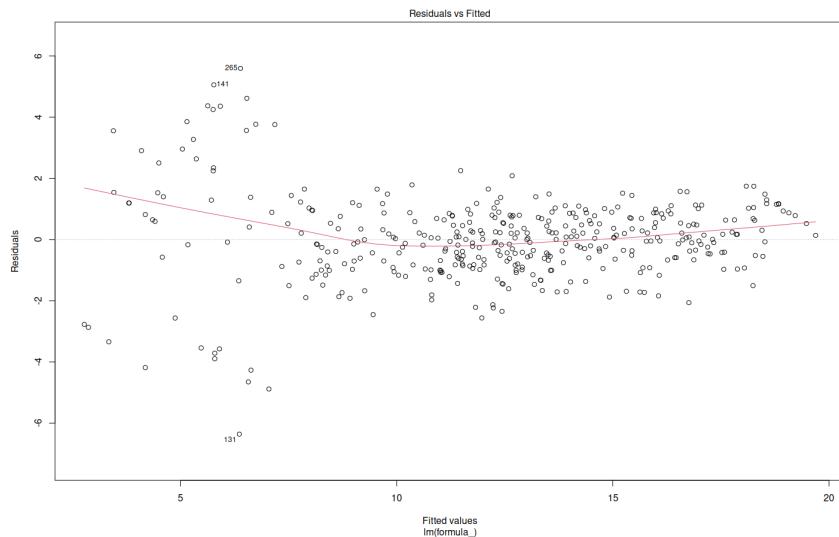


هیچ‌یک از این دو متغیر به نظر نمی‌رسد که رابطه‌ی خطی قوی‌ای با G2 داشته باشند، و در واقع اصلاً بالایی با correlation variable response شرط :nearly normal residuals •



این بار هم مانند مدل B، این توزیع یک توزیع heavy-tail از نرمال را نشان می‌دهد.

• شرط constant variability



مجدداً مثل قسمت قبل، پراکندگی residual‌ها در پیش‌بینی نمرات کمتر، به مراتب بیشتر از سایر نمرات است.

کد رسم این پلات‌ها را هم در ادامه می‌بینید.

```
#model in part E
ggplot(students, aes(x=health, y=G2)) +
  geom_point(color='#6c8c7e')

ggplot(students, aes(x=absences, y=G2)) +
  geom_point(color='#6c8c7e')

plot(be_pvalue_model)
```

برای مقایسه‌ی این دو روش، باید یادآوری کنم که adjusted R-squared در مدل قسمت B برابر ۰.۸۷۷ است و در قسمت E برابر با ۰.۸۸۵۴ است. این اعداد نشان می‌دهند که بهبود حاصل از دو متغیر اضافه‌تر - که بر خلاف متغیرهای قبلی، شرط خطی بودن رابطه را هم ندارند - بسیار بسیار جزئی بوده و بدیهی است که اشکالات مدل را در دو شرط nearly normal residuals و constant variability مرتفع نکرده است.

هر دوی مدل‌ها می‌توانند بخش خوبی از پراکندگی داده‌ها را توجیه کنند اما شرایط linear regression را به طور کامل برآورده نکرده‌اند و احتمالاً در صورتی که از مدل قسمت B استفاده کنیم، به دلیل نبود متفاوتی general response، مدل non-correlated خواهیم داشت. و مدل قسمت E یک مدل reliable نیست.

- بخش G :

با استفاده از کد زیر، مقدار RMSE میانگین را برای هر یک از دو مدل بخش B و E و با روش k-fold cross validation محاسبه کردیم.

```
train.control <- trainControl(method = "cv", number=5)
fit = train(G2 ~ G3 + G1 + failures, data=students, method="lm", trControl=train.control)
print(fit)

fit2 = train(G2 ~ G3 + G1 + failures + health + absences, data=students, method='lm', trControl=train.control)
print(fit2)
```

نتیجه‌ی cross-validation برای مدل قسمت B به این صورت است:

```
Linear Regression

395 samples
 3 predictor

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 316, 316, 316, 317, 315
Resampling results:

RMSE     Rsquared      MAE
1.443747  0.8775615  1.011196
```

و برای مدل قسمت E:

```
Linear Regression

395 samples
 5 predictor

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 316, 315, 315, 317, 317
Resampling results:

RMSE     Rsquared      MAE
1.428285  0.8803372  1.006095
```

از مقایسه‌ی این دو نتیجه متوجه می‌شویم که مقدار RMSE در مدل قسمت E با اختلاف بسیار کمی کمتر است و مقدار Rsquared آن هم بیشتر است. مقدار RMSE در واقع میزان پراکندگی residual‌ها را نشان می‌دهد و در قسمت E با کم شدن این مقدار، در واقع ما به مدل دقیق‌تری برای پیش‌بینی دست یافته‌ایم.

• سوال ششم:

برای حل این سوال، متغیر romantic به عنوان response variable انتخاب شده و متغیرهای age، studytime، internet، sex و health برای پیش‌بینی آن انتخاب شده‌اند.

- بخش A:

برای این بخش، ابتدا باید متغیرهای sex، romantic و internet را که همگی categorical هستند به numerical تبدیل کنیم (برای جنسیت، زن بودن معادل ۱ در نظر گرفته شده و برعکس). بعد از آن مدل خودمان را فیت می‌کنیم. کد این بخش را می‌توانید در ادامه ببینید.

```
data = students

data$romantic[data$romantic == 'yes'] = 1
data$romantic[data$romantic == 'no'] = 0
data$romantic = as.numeric(data$romantic)

data$sex[data$sex == 'F'] = 1
data$sex[data$sex == 'M'] = 0
data$sex = as.numeric(data$sex)

data$internet[data$internet == 'yes'] = 1
data$internet[data$internet == 'no'] = 0
data$internet = as.numeric(data$internet)

lr = glm(romantic ~ sex + internet + studytime + age + health, binomial, data=data)
summary(lr)
```

ضرایب بدست‌آمده برای مدل فوق، در خلاصه‌ی زیر قابل بررسی است.

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.28518   1.64918  -4.417 9.99e-06 ***
sex          0.48260   0.23694   2.037 0.041670 *
internet    0.74337   0.32384   2.295 0.021706 *
studytime   0.04743   0.13902   0.341 0.732972
age          0.31159   0.08791   3.545 0.000393 ***
health       0.10675   0.08188   1.304 0.192308
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 503.31 on 394 degrees of freedom
Residual deviance: 481.63 on 389 degrees of freedom
AIC: 493.63
```

مقدار بدست‌آمده برای intercept به این معناست که اگر تمامی متغیرهای در نظر گرفته شده، برابر با صفر باشند، لگاریتم شانس اینکه یک دانش‌آموز، دانش‌آموز احساسی‌ای باشد، برابر با ۷.۲۹- است.

ضریب متغیر sex بیان می‌کند که لگاریتم نسبت شانس احساسی بودن یک دانشآموز دختر به یک دانشآموز پسر، برابر با $e^{0.48}$ است؛ اگر تمامی متغیرهای دیگر را ثابت در نظر بگیریم. در واقع با ثابت بودن سایر شرایط، شانس احساسی بودن یک دختر $e^{0.48}$ برابر یک پسر است.

ضریب متغیر internet که عدد $e^{0.74}$ بدهست آمده، در واقع بیان می‌کند که با ثابت درنظر گرفته شدن تمامی شرایط، لگاریتم نسبت شانس احساسی بودن کسی که به اینترنت دسترسی دارد به کسی که ندارد، $e^{0.74}$ است و این یعنی شانس احساسی بودن با دسترسی به اینترنت، $e^{0.74}$ برابر شانس احساسی بودن، بدون دسترسی به اینترنت است.

همچنین ضریب studytime نشان می‌دهد که لگاریتم نسبت شانس کسی که یک ساعت در هفته بیشتر از فرد دیگری درس می‌خواند به فرد دوم، برابر با $e^{0.47}$ است که یعنی شانس احساسی بودن نفر اول، $e^{0.47}$ برابر نفر دوم است.

در ادامه، با بررسی مقدار به دست آمده برای متغیر age به این نتیجه میرسیم که لگاریتم نسبت شانس احساسی بودن فرد A که همه‌ی شرایطش با فرد B مساوی است و تنها یک سال بزرگتر از فرد B است، به فرد B، $e^{0.31}$ برابر است که به عبارت دیگر، شانس احساسی بودن فرد A، $e^{0.31}$ برابر شانس احساسی بودن فرد B است.

در نهایت و برای معیار سلامتی، ضریب محاسبه شده برابر با $e^{0.11}$ است که به این معناست که در صورت ثابت بودن تمامی شرایط، لگاریتم نسبت شانس کسی که یک واحد سلامت‌تر است، نسبت به شانس کسی که یک واحد کمتر است، برابر با $e^{0.11}$ است و یا شانس احساسی بودن فرد اول، $e^{0.11}$ برابر فرد دوم است.

- بخش B:

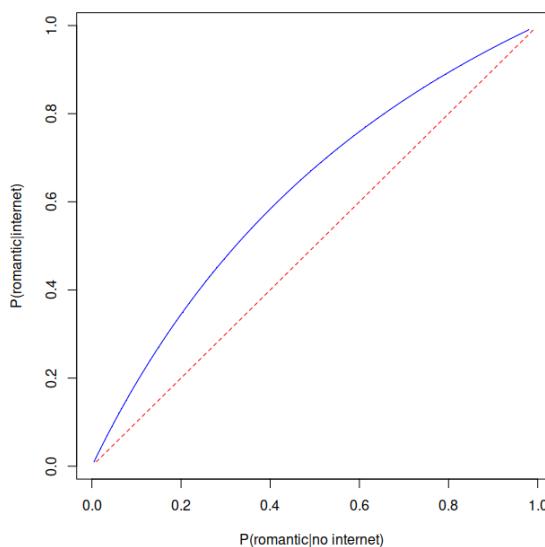
متغیر internet را برای رسم نمودار OR انتخاب کردیم. ما با توجه به ضریب این متغیر در مدل بالا، می‌دانیم که شانس احساسی بودن فردی که به اینترنت دسترسی دارد، $e^{0.74}$ برابر شانسی کسی است که دسترسی ندارد. برای رسم این نمودار، هر بار یک مقدار برای احتمال احساسی بودن به شرط دسترسی داشتن به اینترنت در نظر می‌گیریم، و با کمک آن احتمال احساسی بودن به شرط دسترسی نداشتن به اینترنت را محاسبه می‌کنیم.

کد این بخش، به صورت زیر است:

```
internet_OR = exp(coef(summary(lr))['internet', 'Estimate'])
pps = c()
ps = seq(from=0.01, to=0.99, by=0.01)
for (p in ps){
  odd = (p/(1-p))/internet_OR
  pp = odd/(1+odd)
  pps = append(pps, pp)
}

plot(pps, ps, col="blue", pch=".",
      xlab='P(romantic|no internet)', ylab='P(romantic|internet)')
lines(pps, ps, col="blue")
lines(ps, ps, col='red', type="l", lty=2)
```

و نمودار رسم شده که نتیجه‌ی آن است، در زیر آورده شده است.



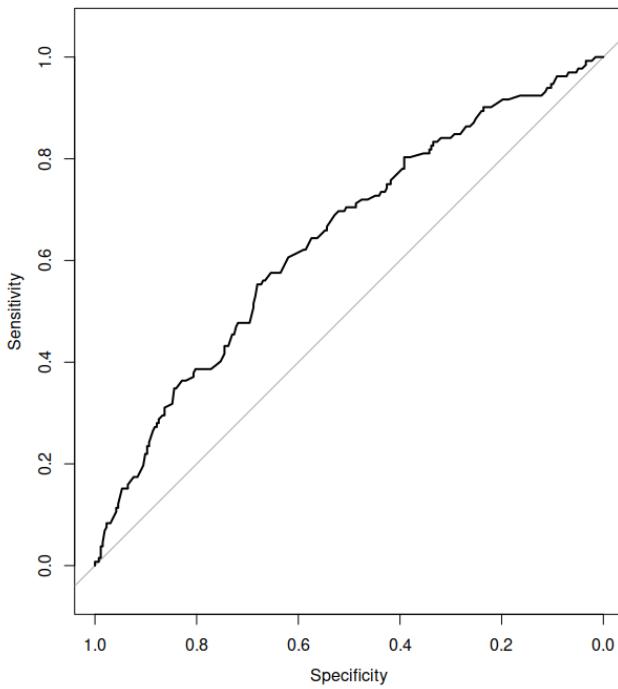
نمودار رسم شده رابطه‌ی احتمال احساسی بودن به شرط دسترسی داشتن به اینترنت و احساسی بودن به شرط دسترسی نداشتن را نشان می‌دهد که با توجه به ضریب بدستآمده در مدل محاسبه شده است. طبق این نمودار، شانس احساسی بودن در حالتی که دانشآموزان به اینترنت دسترسی دارند بیشتر است و به طبع آن احتمال محور ع هم همواره بیشتر است.

- بخش C :

برای رسم نمودار ROC و بدست آوردن سطح زیر نمودار آن (AUC) از کد زیر استفاده می‌کنیم.

```
library(pROC)
preds = predict(lr, data, type="response")
curve = roc(data$romantic, preds, print.auc=TRUE, show.thres=TRUE)
plot(curve)
auc(curve)
```

نمودار ROC رسم شده به این شکل است.



این نمودار بررسی می‌کند که در هر لحظه، با تغییر specificity چه تغییری در sensitivity در مدل رخ می‌دهد. از آنجایی که بین این دو تقریباً یک tradeoff داریم، با توجه به این نمودار، می‌توانیم سعی کنیم نقطه‌ی بهینه برای application خود را بیابیم و تصمیم بگیریم که در مدل خود، می‌خواهیم چه سهمی از هر یک از این دو معیار ببریم.

هرچقدر نمودار ما از خط $y=x$ فاصله‌ی بیشتری داشته باشد، مدل بهتری داریم اما این نمودار بیشتر به این خط نزدیک است، تا دور.

همچنین سطح زیر نمودار این پلات، نشان‌دهنده‌ی AUC است که معیاری برای سنجیدن خوب یا بد بودن مدل ماست. مقدار AUC در این مدل، برابر با 0.6422 است که مقدار خیلی بالایی نیست و نمی‌توانیم ادعا کنیم که مدل ما کارآیی مناسب را دارد.

- بخش D :

بار دیگر نتیجه‌ی مدل را در شکل زیر بررسی می‌کنیم.

```

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.28518   1.64918  -4.417 9.99e-06 ***
sex          0.48260   0.23694   2.037 0.041670 *
internet    0.74337   0.32384   2.295 0.021706 *
studytime   0.04743   0.13902   0.341 0.732972
age          0.31159   0.08791   3.545 0.000393 ***
health       0.10675   0.08188   1.304 0.192308
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 503.31 on 394 degrees of freedom
Residual deviance: 481.63 on 389 degrees of freedom
AIC: 493.63

```

در این مدل، متغیر age کمترین مقدار p-value را دارد و این نشان از این دارد که در این مدل بهترین پر迪کتور ماست. چرا که ضریب آن از نظر آماری کاملا significant است و سایر ضرایب از آن فاصله‌ی قابل توجهی دارند. البته شاید در صورتی که مدلی با متغیرهای دیگری مدل می‌کردیم، ویژگی دیگری عملکرد بهتری نشان می‌داد. به طور کلی این p-value‌ها در کنار سایر متغیرها و در همان مدل معنا دارند.

- بخش E -

در بخش قبل دیدیم که متغیرهای sex، age و internet در مدل p-value قابل قبول و از نظر آماری significant داشته‌اند. در این بخش، با استفاده از این سه متغیر، romantic را مدل کردیم.

```

lr = glm(romantic ~ sex + internet + age, binomial, data=data)
summary(lr)

```

نتیجه‌ی مدل جدید، به این صورت است:

```

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.58410   1.54932  -4.250 2.14e-05 ***
sex          0.46160   0.22110   2.088 0.036825 *
internet    0.70542   0.32002   2.204 0.027501 *
age          0.30093   0.08733   3.446 0.000569 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 503.31 on 394 degrees of freedom
Residual deviance: 483.46 on 391 degrees of freedom
AIC: 491.46

```

در این مدل باز هم این سه متغیر significant هستند و همچنین این مدل معیار AIC کوچکتری نسبت به مدل پیشین دارد. هرچند این اختلاف ناچیز است، اما عملکرد بهتری از خودش نشان داده است.

مقدار بدستآمده برای intercept به این معناست که اگر تمامی متغیرهای درنظر گرفته شده، برابر با صفر باشند، لگاریتم شانس اینکه یک دانشآموز، دانشآموز احساسی‌ای باشد، برابر با ۶.۵۸ است.

ضریب متغیر sex بیان می‌کند که لگاریتم نسبت شانس احساسی بودن یک دانشآموز دختر به یک دانشآموز پسر، برابر با ۰.۴۶ است؛ اگر تمامی متغیرهای دیگر را ثابت در نظر بگیریم. در واقع با ثابت بودن سایر شرایط، شانس احساسی بودن یک دختر $e^{0.46}$ برابر یک پسر است.

ضریب متغیر internet که عدد ۰.۷۱ بودست آمده، در واقع بیان می‌کند که با ثابت درنظر گرفته شدن تمامی شرایط، لگاریتم نسبت شانس احساسی بودن کسی که به اینترنت دسترسی دارد به کسی که ندارد، ۰.۷۱ است و این یعنی شانس احساسی بودن با دسترسی به اینترنت، $e^{0.71}$ برابر شانس احساسی بودن، بدون دسترسی به اینترنت است.

در ادامه، با بررسی مقدار بهدست آمده برای متغیر age به این نتیجه میرسیم که لگاریتم نسبت شانس احساسی بودن فرد A که همه‌ی شرایطش با فرد B مساوی است و تنها یک سال بزرگتر از فرد B است، به فرد B، ۰.۳ برابر است که به عبارت دیگر، شانس احساسی بودن فرد A، $e^{0.3}$ برابر شانس احساسی بودن فرد B است.

- بخش F :

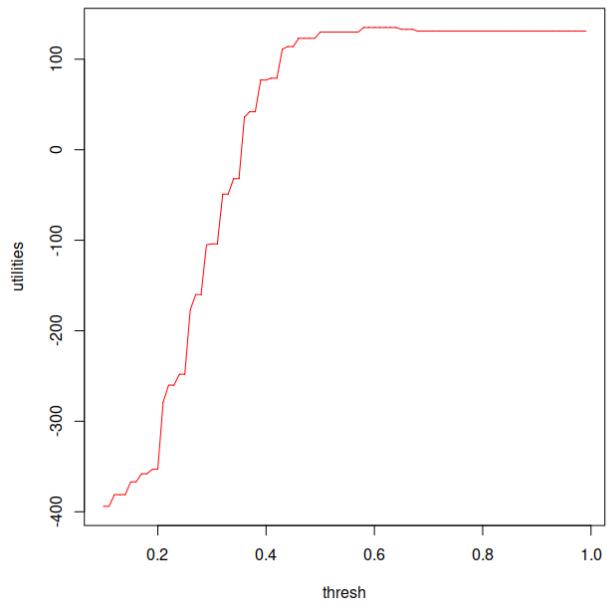
برای محاسبه‌ی utility function، از آنجایی که به نظر می‌رسد تشخیص احساسی بودن یک دانشآموز برای ما اهمیت بیشتر دارد، برای false negative ضریب ۲ در نظر گرفتیم و اندازه‌ی بقیه‌ی ضرایب، برابر ۱ است. با جابجا کردن مقدار threshold، utility function را محاسبه می‌کنیم. کد این

```
utilities = c()
ps = seq(from=0.1, to=0.99, by=0.01)
for (p in ps){
  result = coords(curve, x=p, ret=c('tp', 'fp', 'tn', 'fn'))
  utilities = append(utilities, result$tp + result$tn - 2*result$fp - result$fn)
}

plot(ps, utilities, col="red", pch=".", xlab='thresh')
lines(ps, utilities, col="red")
```

بخش در زیر آورده شده است.

نمودار رسم شده به ازای وزن‌های بالا، به این شکل است.



اگر بخواهیم بیشترین مقدار utility را با توجه به اهمیت‌های گفته‌شده داشته باشیم، باید مقدار threshold را تقریباً برابر با 0.6 قرار دهیم.

• سوال هفتم:

در این بخش، ابتدا باید تمامی متغیرهای کتگوریکال را به متغیرهای numerical تبدیل کنیم. این کار برای متغیرهای باینری نکته‌ای ندارد. اما برای دو متغیر شغل مادران و پدران، باید به تعداد ۴ متغیر باینری تعریف کنیم (برای ۵ دسته شغل) که مدل ما بتواند به درستی تخمین بزند. برای مثلاً یک متغیر Mjob-teacher برای شغل معلم مادران داریم که تنها در صورت معلم بودن آنها ۱ است و به همین ترتیب برای سایر شغل‌ها. مقدار other هم با صفر بودن همه‌ی این ۴ متغیر بررسی می‌شود. همچنین در این پیش‌بینی، از نمرات G1، G2 و G3 استفاده نکردیم. چرا که متغیر response دقیقاً از روی همان‌ها ساخته شده است.

در کد زیر هم اضافه کردن academic_probation را داریم و هم تبدیل متغیرها به متغیرهای مناسب برای مدل. در نهایت هم یک مدل logistic regression را فیت کرده‌ایم.

```
data = students
data = transform(data, academic_probation=ifelse(G1+G2+G3 < 25, 1, 0))

data$school[data$school == 'GP'] = 1
data$school[data$school == 'MS'] = 0
data$school = as.numeric(data$school)

data$sex[data$sex == 'F'] = 1
data$sex[data$sex == 'M'] = 0
data$sex = as.numeric(data$sex)

data$internet[data$internet == 'yes'] = 1
data$internet[data$internet == 'no'] = 0
data$internet = as.numeric(data$internet)

data$romantic[data$romantic == 'yes'] = 1
data$romantic[data$romantic == 'no'] = 0
data$romantic = as.numeric(data$romantic)

data = transform(data, Fjob_teacher=ifelse(Fjob == 'teacher', 1, 0))
data = transform(data, Fjob_services=ifelse(Fjob == 'services', 1, 0))
data = transform(data, Fjob_health=ifelse(Fjob == 'health', 1, 0))
data = transform(data, Fjob_at_home=ifelse(Fjob == 'at_home', 1, 0))

data = transform(data, Mjob_teacher=ifelse(Mjob == 'teacher', 1, 0))
data = transform(data, Mjob_services=ifelse(Mjob == 'services', 1, 0))
data = transform(data, Mjob_health=ifelse(Mjob == 'health', 1, 0))
data = transform(data, Mjob_at_home=ifelse(Mjob == 'at_home', 1, 0))

model = glm(academic_probation ~ school + sex + age + goout + internet + romantic + studytime + failures + health + absences +
            Fjob_teacher + Fjob_services + Fjob_health + Fjob_at_home + Mjob_teacher + Mjob_services + Mjob_health + Mjob_at_home, data=data)
summary(model)
```

و نتیجه‌ی مدل به دست آمده به این صورت است:

```

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.204124  0.286358  -0.713   0.4764    
school       0.006508  0.058321   0.112   0.9112    
sex          0.081016  0.036690   2.208   0.0278 *  
age          0.016013  0.015398   1.040   0.2991    
goout        0.039572  0.015398   2.570   0.0106 *  
internet    0.022123  0.048393   0.457   0.6478    
romantic    0.021889  0.036953   0.592   0.5540    
studytime   -0.049004  0.021778  -2.250   0.0250 *  
failures    0.279257  0.024267  11.508 <2e-16 *** 
health       0.006817  0.012410   0.549   0.5831    
absences    -0.007470  0.002210  -3.380   0.0008 *** 
Fjob_teacher 0.050673  0.068306   0.742   0.4586    
Fjob_services -0.033484 0.040812  -0.820   0.4125    
Fjob_health  -0.093153 0.085207  -1.093   0.2750    
Fjob_at_home -0.071726 0.079294  -0.905   0.3663    
Mjob_teacher -0.041723 0.055512  -0.752   0.4528    
Mjob_services -0.022039 0.045132  -0.488   0.6256    
Mjob_health  -0.116115 0.066043  -1.758   0.0795 .  
Mjob_at_home  0.033027 0.053304   0.620   0.5359    
...
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Dispersion parameter for gaussian family taken to be 0.1098093)

Null deviance: 64.390  on 394  degrees of freedom
Residual deviance: 41.288  on 376  degrees of freedom
AIC: 268.93

```

متغیرهایی که از نظر آماری در این مدل significant بوده‌اند، متغیرهای failures، absenses و sex است. این متغیرهای به ترتیب صعودی p-value نامبرده شده‌اند و متغیری که کوچک‌ترین p-value را دارد همان تعداد failures است.