Final Project - 412

Andrew Sang & Jason Yi

11/30/2019

Dataset

Dataset: https://archive.ics.uci.edu/ml/datasets/bank+marketing (https://archive.ics.uci.edu/ml/datasets/bank+marketing)

The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

Term Deposit is like a CD: https://en.wikipedia.org/wiki/Time_deposit (https://en.wikipedia.org/wiki/Time_deposit)

Data Description

Attribute Information:

bank client data: 1 - age (numeric) 2 - job : type of job (categorical: "admin.","unknown","unemployed","management","housemaid","entrepreneur","student", "blue-collar","self-employed","retired","technician","services") 3 - marital : marital status (categorical: "married","divorced","single"; note: "divorced" means divorced or widowed) 4 - education (categorical: "unknown","secondary","primary","tertiary") 5 - default: has credit in default? (binary: "yes","no") 6 - balance: average yearly balance, in euros (numeric) 7 - housing: has housing loan? (binary: "yes","no") 8 - loan: has personal loan? (binary: "yes","no")

related with the last contact of the current campaign: 9 - contact: contact communication type (categorical: "unknown","telephone","cellular") 10 - day: last contact day of the month (numeric) 11 - month: last contact month of year (categorical: "jan", "feb", "mar", …, "nov", "dec") 12 - duration: last contact duration, in seconds (numeric)

other attributes: 13 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact) 14 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric, -1 means client was not previously contacted) 15 - previous: number of contacts performed before this campaign and for this client (numeric) 16 - poutcome: outcome of the previous marketing campaign (categorical: "unknown","other","failure","success")

Output variable (desired target): 17 - y - has the client subscribed a term deposit? (binary: "yes","no")

Missing Attribute Values: None

Data Import

```
path = 'data/bank/'
bank = read.csv(paste(path,'bank-additional-full.csv',sep=''), sep=";")

# duration should be removed: cheating column. From the docs: Important note: this attri
bute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the durati
on is not known before a call is performed. Also, after the end of the call y is obvious
ly known. Thus, this input should only be included for benchmark purposes and should be
 discarded if the intention is to have a realistic predictive model.
bank = bank %>% dplyr::select(-'duration')

bank$y = ifelse(bank$y == 'yes', 1, 0) # recode as 1/0

# check for complete cases
print(complete.cases(bank) %>% sum() == (bank %>% dim())[[1]])
```

```
## [1] TRUE
```

```
print(bank$y %>% mean())
```

```
## [1] 0.1126542
```

```
# feature change: scale was affected so recoded 999 to -1
bank$pdays_code = ifelse(bank$pdays == 999, -1, bank$pdays)
bank = bank %>% dplyr::select(-'pdays')
```

Train/Test Split

```
set.seed(1234)
train.list <- sample(1:nrow(bank), 0.8*nrow(bank), replace = F)
bank_train <- bank[train.list,]
bank_test <- bank[-train.list,]
```

Random Forest to figure out important variables

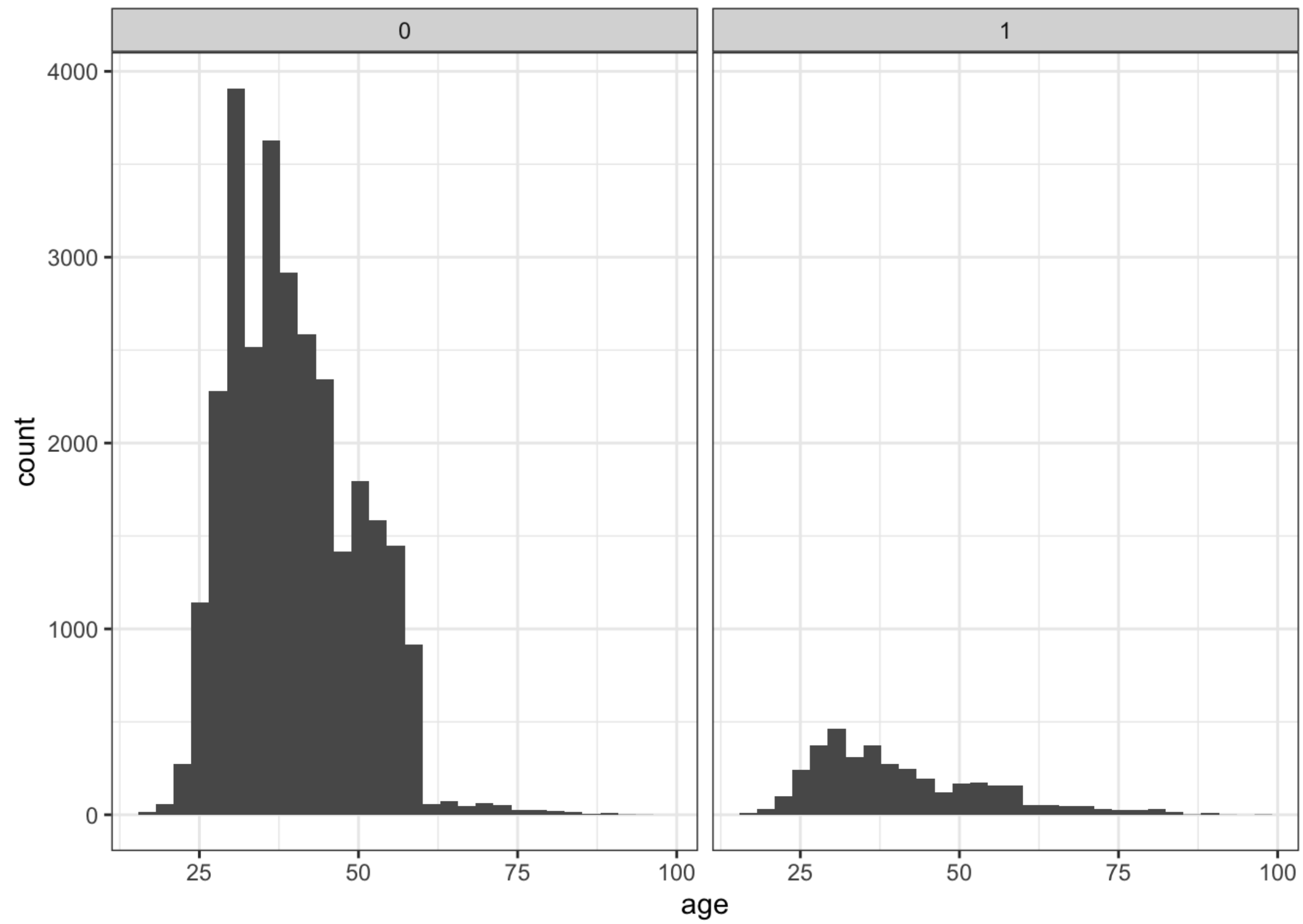
```
h2o.varimp(rf)
```

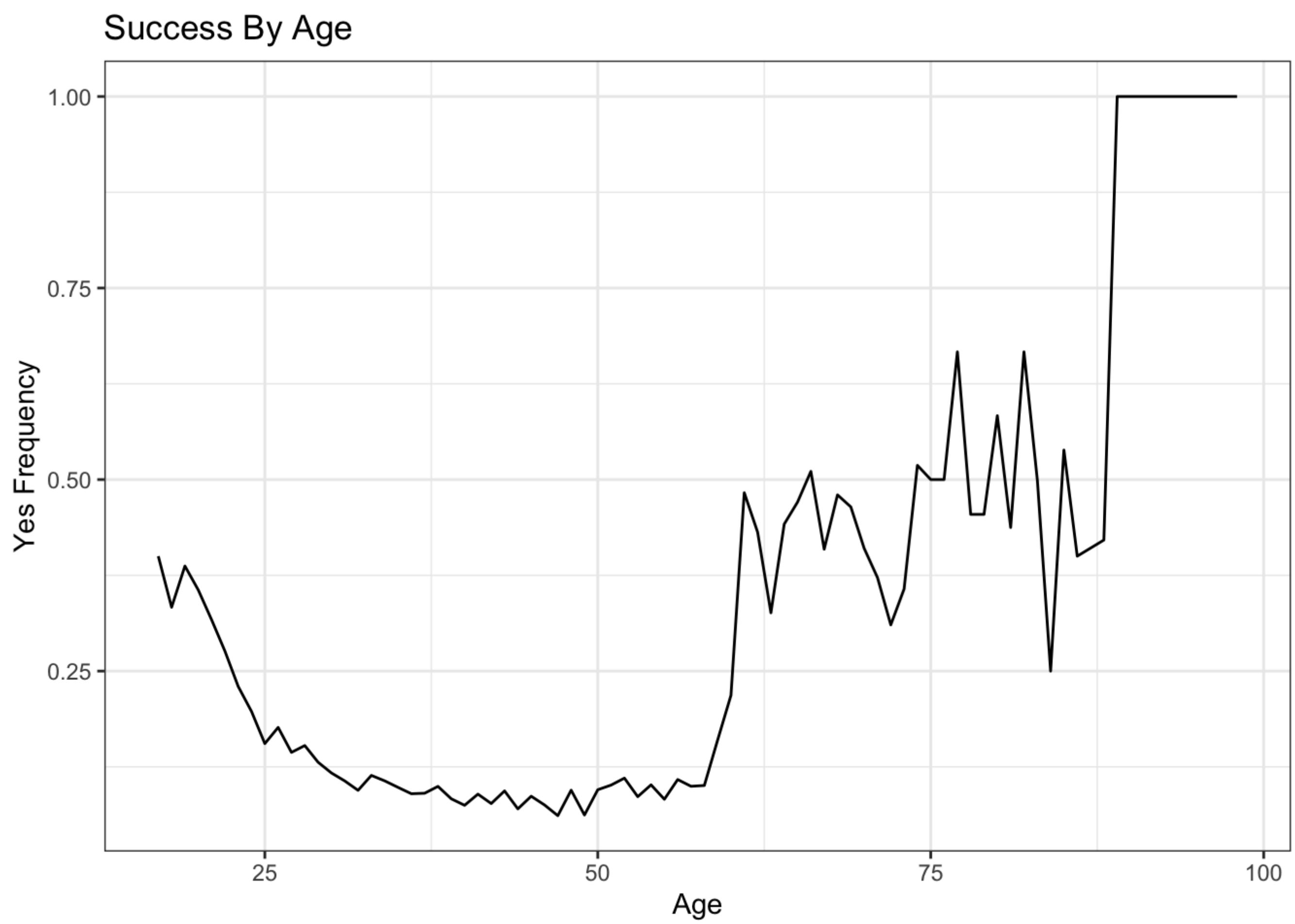
```
## Variable Importances:
##           variable relative_importance scaled_importance percentage
## 1              age         6718.672852          1.000000   0.123365
## 2      nr.employed         6162.181641          0.917172   0.113147
## 3        euribor3m         5441.937012          0.809972   0.099922
## 4              job         5253.984375          0.781997   0.096471
## 5        education         3963.116455          0.589866   0.072769
## 6      day_of_week         3641.172363          0.541948   0.066857
## 7       pdays_code         3342.767090          0.497534   0.061378
## 8         campaign         2977.885498          0.443225   0.054679
## 9         poutcome         2602.421875          0.387342   0.047784
## 10   cons.conf.idx         2073.428223          0.308607   0.038071
## 11           month         1996.680908          0.297184   0.036662
## 12         marital         1962.256348          0.292060   0.036030
## 13         housing         1762.748291          0.262366   0.032367
## 14    emp.var.rate         1558.517578          0.231968   0.028617
## 15            loan         1308.133301          0.194701   0.024019
## 16  cons.price.idx         1124.484619          0.167367   0.020647
## 17        previous         1057.870850          0.157452   0.019424
## 18         contact          880.120728          0.130996   0.016160
## 19         default          633.345459          0.094266   0.011629
```

Data Exploration

y

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


0
1
4000
3000
2000
1000
0
count
25
50
75
100
age

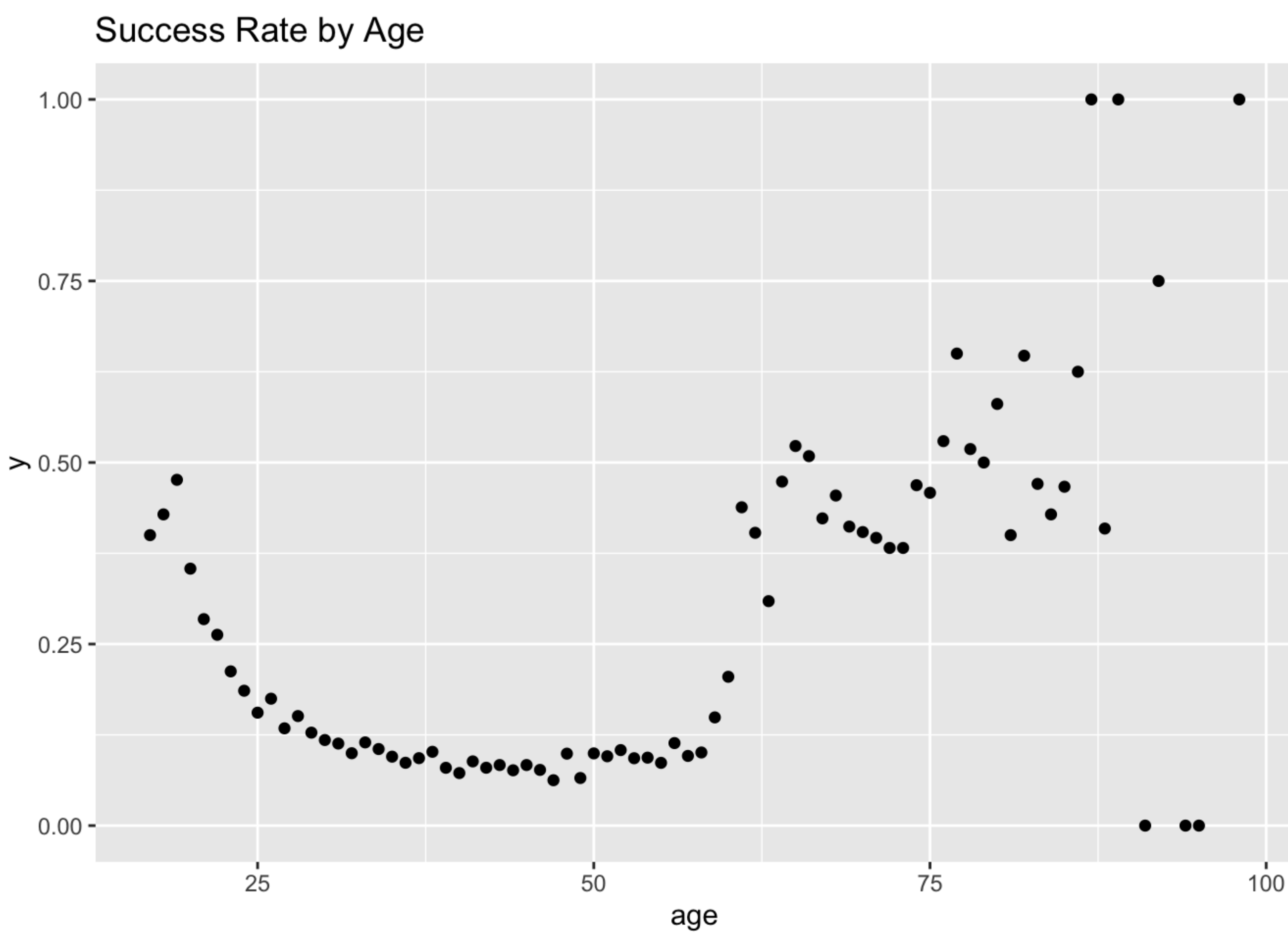

Success By Age
1.00
0.75
0.50
0.25
Yes Frequency
25
50
75
100
Age

```
## # A tibble: 2 x 3
## # Groups:   y [2]
##       y     n  freq
##   <dbl> <int> <dbl>
## 1     0 36548     1
## 2     1  4640     1
```

age

First, we look at age.

```
## # A tibble: 78 x 3
##      age `mean(y)` `n()`
##    <int>     <dbl> <int>
##  1    17     0.4       5
##  2    18     0.429    28
##  3    19     0.476    42
##  4    20     0.354    65
##  5    21     0.284   102
##  6    22     0.263   137
##  7    23     0.212   226
##  8    24     0.186   463
##  9    25     0.156   598
## 10    26     0.175   698
## # … with 68 more rows
```



####

euribor3m Then we look at the euribor3m

job

```
bank %>% group_by(job) %>% summarise(mean(y), cnt= n()) %>% arrange(desc(cnt))
```

```
## # A tibble: 12 x 3
##    job           `mean(y)`   cnt
##    <fct>             <dbl> <int>
##  1 admin.           0.130  10422
##  2 blue-collar      0.0689  9254
##  3 technician       0.108   6743
##  4 services         0.0814  3969
##  5 management       0.112   2924
##  6 retired          0.252   1720
##  7 entrepreneur     0.0852  1456
##  8 self-employed    0.105   1421
##  9 housemaid        0.1     1060
## 10 unemployed       0.142   1014
## 11 student          0.314    875
## 12 unknown          0.112    330
```

campaign

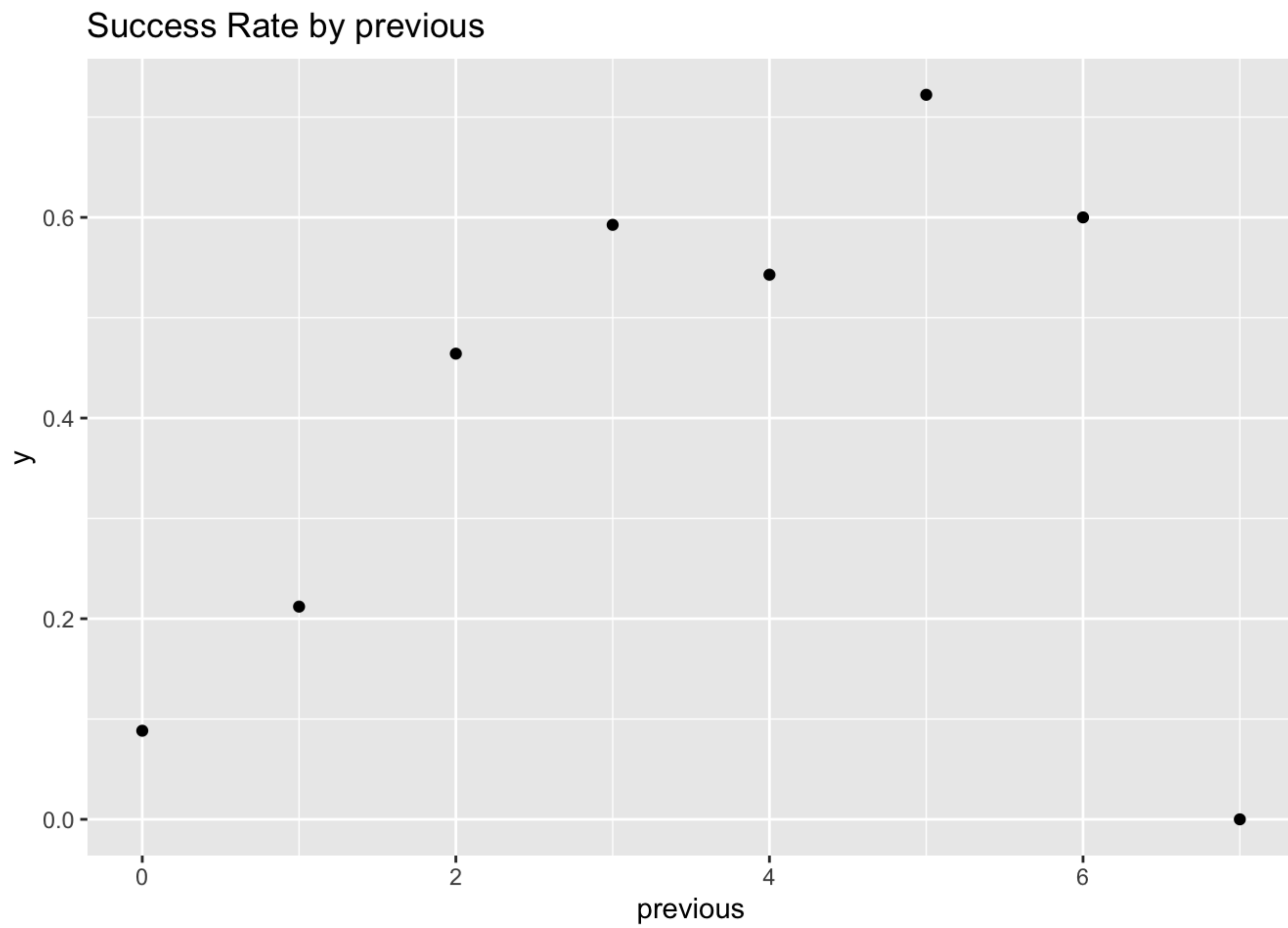
pdays

poutcome

```
bank %>% group_by(poutcome) %>% summarise(mean(y), n())
```

```
## # A tibble: 3 x 3
##   poutcome    `mean(y)` `n()`
##   <fct>           <dbl> <int>
## 1 failure        0.142   4252
## 2 nonexistent    0.0883 35563
## 3 success        0.651   1373
```

```
ggplot(data=bank, aes(x=previous, y=y)) +
  stat_summary(fun.y="mean", geom="point") +
  ggtitle('Success Rate by previous')
```




```
bank %>% group_by(previous) %>% summarise(mean(y), n())
```

```
## # A tibble: 8 x 3
##   previous `mean(y)` `n()`
##      <int>     <dbl> <int>
## 1        0    0.0883 35563
## 2        1    0.212   4561
## 3        2    0.464    754
## 4        3    0.593    216
## 5        4    0.543     70
## 6        5    0.722     18
## 7        6    0.6        5
## 8        7    0          1
```

loans

```
## # A tibble: 6 x 4
## # Groups:   loan [3]
##   loan        y     n  freq
##   <fct>   <dbl> <int> <dbl>
## 1 no          0 24088 0.886
## 2 no          1  3109 0.114
## 3 unknown     0   711 0.895
## 4 unknown     1    83 0.105
## 5 yes         0  4422 0.892
## 6 yes         1   537 0.108
```

Model Fitting and Selection

```
# everything
fit1 = glm(y ~ ., data=bank_train, family = binomial(link = "logit"))

# probit
fit2 = glm(y ~ ., data=bank_train, family = binomial(link = "probit"))

# Stepwise regression model
fit3 <- stepAIC(fit1, direction = "both",
                      trace = FALSE)
fit4 <- step(glm(y ~., data = bank_train, family=binomial),trace=0,steps=100)

# Try a model with fewer variables
drop_obj = drop1(fit1, test="Chisq")
fit5<- glm(y ~ age + job + marital + education + default + housing +
    loan + contact + month + day_of_week + campaign + # previous +
      previous + poutcome + emp.var.rate + cons.price.idx + euribor3m + nr.employed + pd
ays_code,
    data=bank_train, family = binomial(link = "logit"))

# Stepwise for interaction terms too
add_one_mdl = add1(fit1, ~.^2, test="Chisq") # job:education, age:marital look like good
candidates to add
fit8<- glm(y ~ age:marital + job:education + job + default + contact + month + day_of_we
ek +
    campaign + poutcome + emp.var.rate + cons.price.idx + cons.conf.idx +
    euribor3m + nr.employed + pdays_code,
    data=bank_train,
    family = "binomial")
summary(fit8)
print(1-pchisq(23269-deviance(fit8), 32949-df.residual(fit8)))
```

Model Metrics

```
Model_<- c(1,2,3,8)
Formula_<- c(summary(fit1)$call,summary(fit2)$call,summary(fit3)$call,summary(fit8)$cal
l)
Formula_<- format(Formula_)

# Metrics
Deviance_<- c(summary.glm(fit1)$deviance,summary.glm(fit2)$deviance,summary.glm(fit3)$de
viance,summary.glm(fit8)$deviance)
AIC_<- c(AIC(fit1),AIC(fit2),AIC(fit3),AIC(fit8))
MSE_<- c(mean(fit1$residuals^2), mean(fit2$residuals^2),mean(fit3$residuals^2),mean(fit8
$residuals^2))
PR_AUC_ = c(PRAUC(fitted(fit1), bank_train$y), PRAUC(fitted(fit2), bank_train$y), PRAUC
(fitted(fit3), bank_train$y),PRAUC(fitted(fit8), bank_train$y))
AUC_ = c(AUC(fitted(fit1), bank_train$y), AUC(fitted(fit2), bank_train$y), AUC(fitted(fi
t3), bank_train$y),AUC(fitted(fit8), bank_train$y))

# Metrics on Test
TEST_PR_AUC = c(PRAUC(predict(fit1, newdata=bank_test),bank_test$y),PRAUC(predict(fit2,
 newdata=bank_test),bank_test$y),PRAUC(predict(fit3, newdata=bank_test),bank_test$y),PRA
UC(predict(fit8, newdata=bank_test),bank_test$y) )
TEST_AUC = c(AUC(predict(fit1, newdata=bank_test),bank_test$y),AUC(predict(fit2, newdata
=bank_test),bank_test$y),AUC(predict(fit3, newdata=bank_test),bank_test$y),AUC(predict(f
it8, newdata=bank_test),bank_test$y) )

# Set up Output Table
together <- data.frame(Model_,Formula_,Deviance_,MSE_, AIC_,  AUC_, PR_AUC_, TEST_AUC, T
EST_PR_AUC)
kable(together, caption= "Model Summaries", digits = 2) %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed", "responsive")) %
>%
  row_spec(3:3, bold = T, color = "white", background = "#D7261E")
```

Model Summaries

Model_	Formula_	Deviance_	MSE_	AIC_	AUC_	PR_AUC_	TEST_AUC	TEST_PR_AUC
1	glm(formula = y ~ ., family = binomial(link = "logit"), data = bank_train)	18196.98	20.49	18300.98	0.79	0.46	0.79	0.44
2	glm(formula = y ~ ., family = binomial(link = "probit"), data = bank_train)	18201.23	4.37	18305.23	0.80	0.46	0.79	0.44

Model_	Formula_	Deviance_	MSE_	AIC_	AUC_	PR_AUC_	TEST_AUC	TEST_PR_AUC
3	glm(formula = y ~ job + default + contact + month + day_of_week + , campaign + poutcome + emp.var.rate + cons.price.idx + cons.conf.idx + , euribor3m + nr.employed + pdays_code, family = binomial(link = "logit"), , data = bank_train)	18205.91	20.48	18279.91	0.79	0.46	0.79	0.44
8	glm(formula = y ~ age:marital + job:education + job + default + , contact + month + day_of_week + campaign + poutcome + emp.var.rate + , cons.price.idx + cons.conf.idx + euribor3m + nr.employed + , pdays_code, family = "binomial", data = bank_train)	18132.25	21.26	18370.25	0.80	0.46	0.79	0.44

Model Diagnostics

```
summary(fit3)
```

```
##
## Call:
## glm(formula = y ~ job + default + contact + month + day_of_week +
##     campaign + poutcome + emp.var.rate + cons.price.idx + cons.conf.idx +
##     euribor3m + nr.employed + pdays_code, family = binomial(link = "logit"),
##     data = bank_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1610  -0.3898  -0.3236  -0.2620   2.9636
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -2.234e+02  3.698e+01  -6.040 1.54e-09 ***
## jobblue-collar     -1.943e-01  6.309e-02  -3.080 0.002070 **
## jobentrepreneur    -4.945e-02  1.159e-01  -0.427 0.669699
## jobhousemaid       -5.409e-02  1.359e-01  -0.398 0.690596
## jobmanagement      -4.017e-02  8.143e-02  -0.493 0.621796
## jobretired          2.071e-01  8.350e-02   2.481 0.013104 *
## jobself-employed   -8.804e-02  1.144e-01  -0.770 0.441580
## jobservices        -1.506e-01  7.995e-02  -1.883 0.059663 .
## jobstudent          2.181e-01  1.019e-01   2.140 0.032385 *
## jobtechnician      -5.355e-02  6.228e-02  -0.860 0.389920
## jobunemployed      -1.246e-01  1.257e-01  -0.991 0.321635
## jobunknown         -1.344e-01  2.318e-01  -0.580 0.562224
## defaultunknown     -2.215e-01  6.292e-02  -3.520 0.000432 ***
## defaultyes         -7.606e+00  8.423e+01  -0.090 0.928047
## contacttelephone   -7.242e-01  7.505e-02  -9.649  < 2e-16 ***
## monthaug            4.981e-01  1.203e-01   4.140 3.47e-05 ***
## monthdec            5.515e-01  2.154e-01   2.560 0.010466 *
## monthjul            2.713e-02  9.316e-02   0.291 0.770896
## monthjun           -6.100e-01  1.235e-01  -4.939 7.85e-07 ***
## monthmar            1.528e+00  1.445e-01  10.574  < 2e-16 ***
## monthmay           -4.548e-01  8.048e-02  -5.652 1.59e-08 ***
## monthnov           -4.294e-01  1.172e-01  -3.664 0.000248 ***
## monthoct           -1.425e-02  1.514e-01  -0.094 0.925011
## monthsep            2.194e-01  1.770e-01   1.240 0.215146
## day_of_weekmon     -2.163e-01  6.445e-02  -3.356 0.000790 ***
## day_of_weekthu      6.151e-02  6.210e-02   0.991 0.321916
## day_of_weektue      4.880e-02  6.384e-02   0.764 0.444619
## day_of_weekwed      1.211e-01  6.365e-02   1.903 0.057030 .
## campaign           -4.262e-02  1.032e-02  -4.128 3.65e-05 ***
## poutcomenonexistent  4.456e-01  6.168e-02   7.225 5.00e-13 ***
## poutcomesuccess     1.631e+00  1.145e-01  14.243  < 2e-16 ***
## emp.var.rate       -1.441e+00  1.383e-01 -10.416  < 2e-16 ***
## cons.price.idx      2.026e+00  2.438e-01   8.310  < 2e-16 ***
## cons.conf.idx       2.915e-02  7.742e-03   3.764 0.000167 ***
## euribor3m           1.939e-01  1.279e-01   1.516 0.129468
## nr.employed         6.219e-03  3.011e-03   2.066 0.038852 *
## pdays_code          2.736e-02  1.314e-02   2.082 0.037317 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 23269  on 32949  degrees of freedom
## Residual deviance: 18206  on 32913  degrees of freedom
## AIC: 18280
##
## Number of Fisher Scoring iterations: 9
```

```
# interpretation odds
round(exp(coef(fit3)) - 1,2)
```

```
##         (Intercept)      jobblue-collar     jobentrepreneur        jobhousemaid
##               -1.00               -0.18               -0.05               -0.05
##       jobmanagement          jobretired    jobself-employed         jobservices
##               -0.04                0.23               -0.08               -0.14
##          jobstudent       jobtechnician       jobunemployed          jobunknown
##                0.24               -0.05               -0.12               -0.13
##      defaultunknown          defaultyes    contacttelephone            monthaug
##               -0.20               -1.00               -0.52                0.65
##            monthdec            monthjul            monthjun            monthmar
##                0.74                0.03               -0.46                3.61
##            monthmay            monthnov            monthoct            monthsep
##               -0.37               -0.35               -0.01                0.25
##      day_of_weekmon      day_of_weekthu      day_of_weektue      day_of_weekwed
##               -0.19                0.06                0.05                0.13
##            campaign poutcomenonexistent     poutcomesuccess        emp.var.rate
##               -0.04                0.56                4.11               -0.76
##      cons.price.idx       cons.conf.idx           euribor3m         nr.employed
##                6.58                0.03                0.21                0.01
##          pdays_code
##                0.03
```

Deviance Residuals vs Binned Linear Predictor

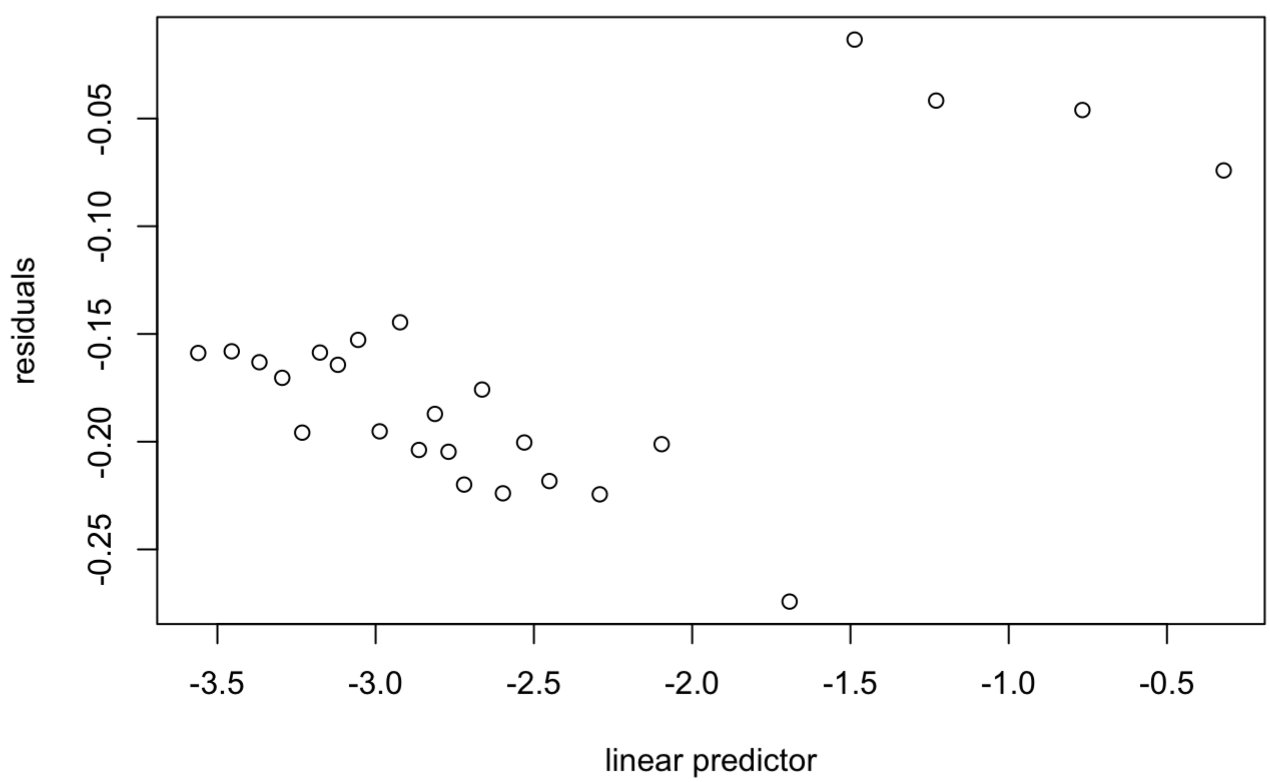
```
# farway example
linpred = predict(fit3)
bank_train = bank_train %>% mutate(residuals = residuals(fit3), linpred=predict(fit3))

gdf = bank_train %>% group_by(cut(linpred, breaks=unique(quantile(linpred,(1:25)/26))))
```

```
## Warning: Factor `cut(linpred, breaks = unique(quantile(linpred, (1:25)/26)))`
## contains implicit NA, consider using `forcats::fct_explicit_na`
```

```
diagdf <- summarise(gdf, residuals=mean(residuals), linpred=mean(linpred), cnt=n())
diagdf %>% head()
```

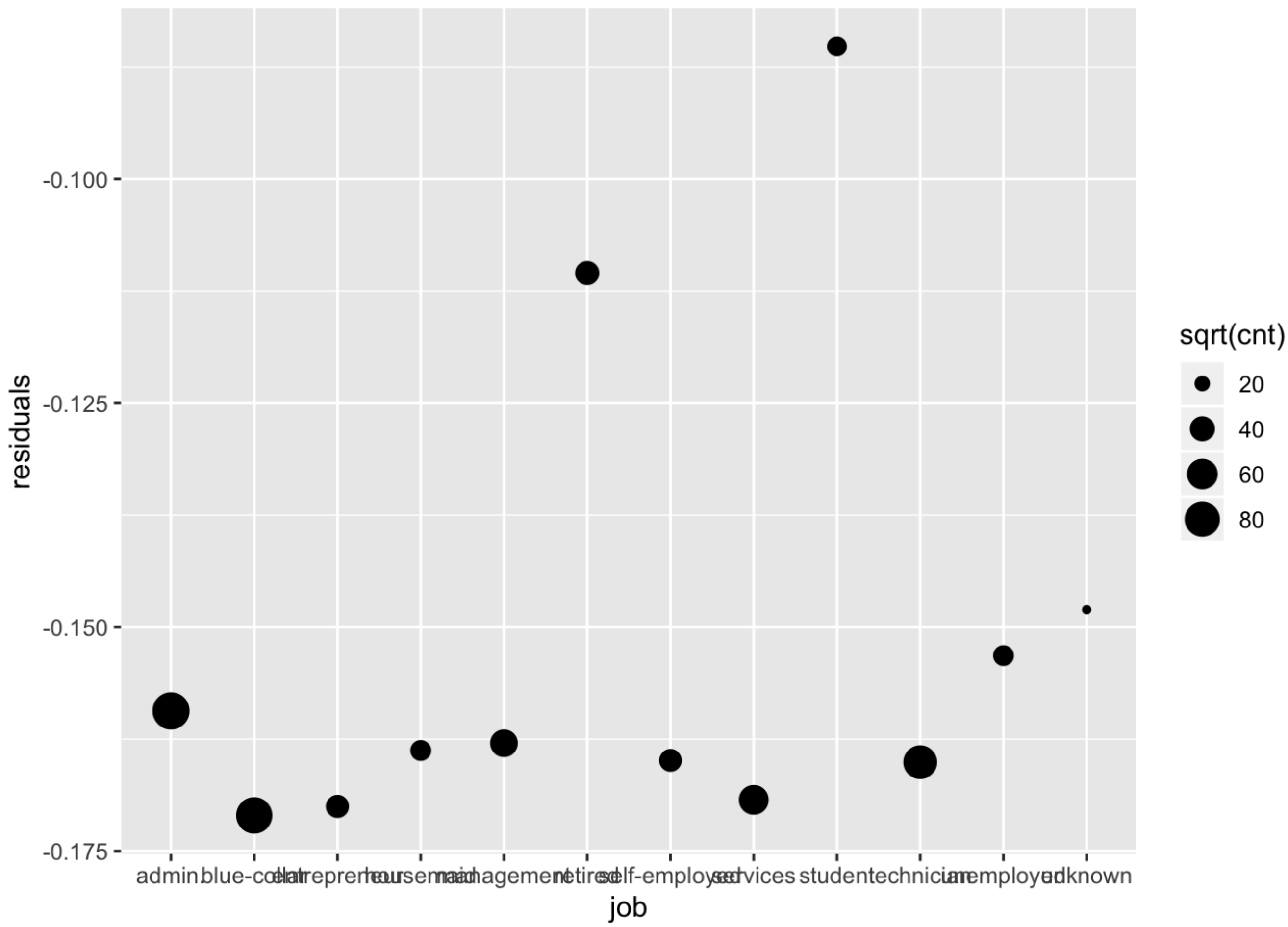
```
## # A tibble: 6 x 4
##   `cut(linpred, breaks = unique(quantile(linpred, (1:25… residuals linpred   cnt
##   <fct>                                                      <dbl>   <dbl> <int>
## 1 (-3.62,-3.5]                                              -0.159   -3.56  1273
## 2 (-3.5,-3.41]                                              -0.158   -3.45  1265
## 3 (-3.41,-3.33]                                             -0.163   -3.37  1297
## 4 (-3.33,-3.26]                                             -0.170   -3.30  1233
## 5 (-3.26,-3.2]                                              -0.196   -3.23  1272
## 6 (-3.2,-3.15]                                              -0.159   -3.18  1263
```

```
plot(residuals ~ linpred, diagdf, xlab="linear predictor")
```



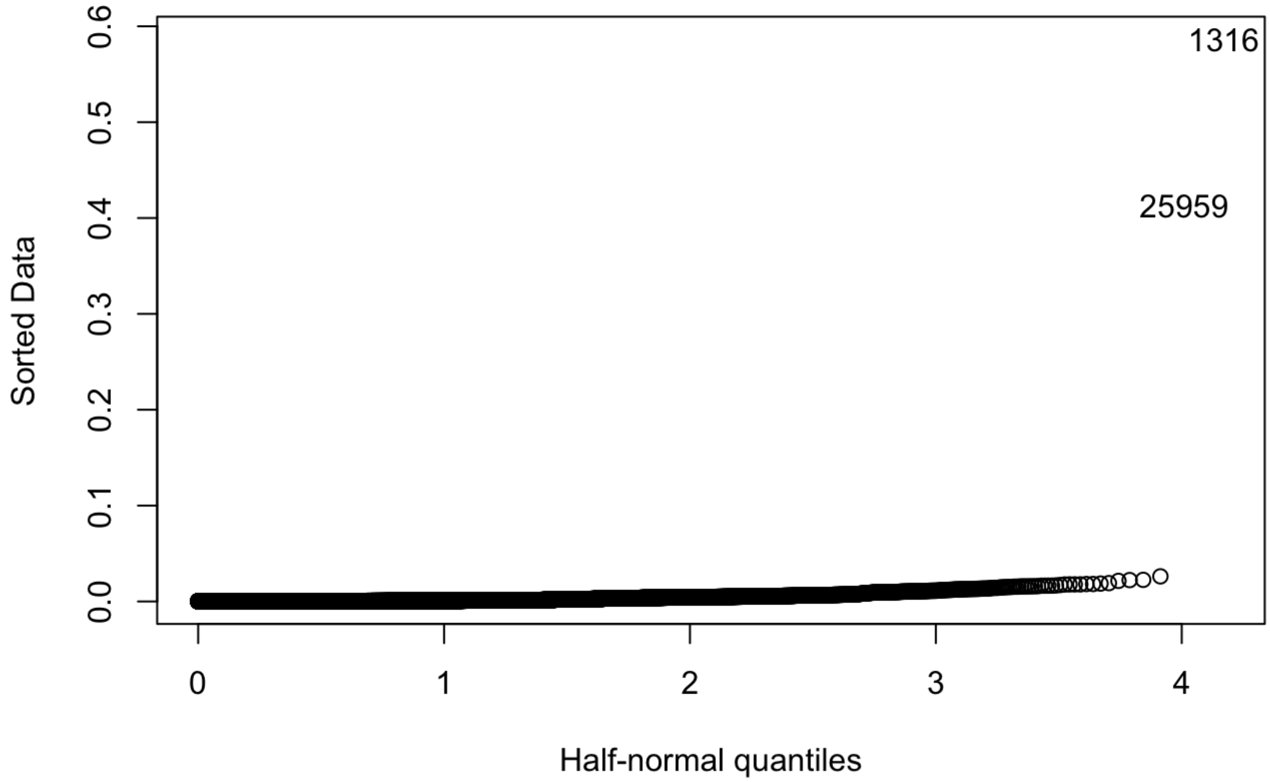
Deviance Residuals vs Predictor Values

```
gdf <- group_by(bank_train, job)
diagdf <- summarise(gdf, residuals=mean(residuals), cnt=n())
ggplot(diagdf, aes(x=job,y=residuals, size=sqrt(cnt))) + geom_point()
```



Leverage Analysis

```
halfnorm(hatvalues(fit3))
```

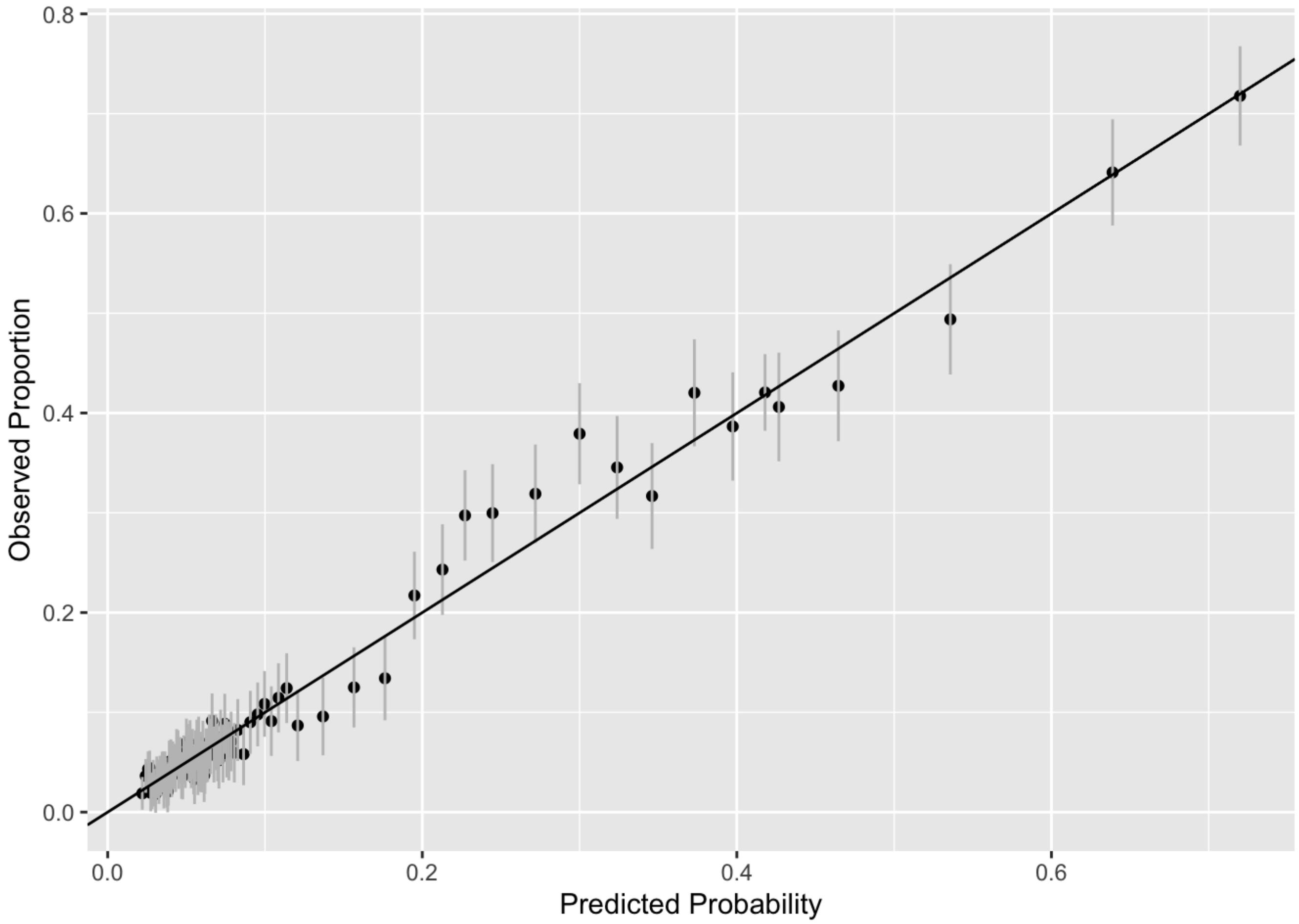



```
bank_train %>% filter(hatvalues(fit3) > 0.3)
```

```
##   age        job marital           education default housing loan  contact
## 1  48 technician married professional.course     yes     yes   no cellular
## 2  31 unemployed married         high.school     yes      no   no cellular
##   month day_of_week campaign previous    poutcome emp.var.rate cons.price.idx
## 1   aug         tue        1        0 nonexistent          1.4         93.444
## 2   nov         tue        2        1     failure         -0.1         93.200
##   cons.conf.idx euribor3m nr.employed y pdays_code    residuals   linpred
## 1         -36.1     4.963      5228.1 0         -1 -0.007799366 -10.40056
## 2         -42.0     4.153      5195.8 0         -1 -0.006547746 -10.75041
```

Comparing Observed and Predicted Proportions

```
bank_trainm <- na.omit(bank_train)
bank_trainm <- mutate(bank_trainm, predprob=predict(fit3,type="response"))
gdf <- group_by(bank_trainm, cut(linpred, breaks=unique(quantile(linpred,(1:100)/101))))
hldf <- summarise(gdf, y=sum(y), ppred=mean(predprob), count=n())

hldf <- mutate(hldf, se.fit=sqrt(ppred*(1-ppred)/count))
ggplot(hldf,aes(x=ppred,y=y/count,ymin=y/count-2*se.fit,ymax=y/count+2*se.fit))+
  geom_point()+geom_linerange(color=grey(0.75))+
  geom_abline(intercept=0,slope=1)+
  xlab("Predicted Probability")+
  ylab("Observed Proportion")
```



Observed Proportion
0.0
0.2
0.4
0.6
0.8
Predicted Probability
0.0
0.2
0.4
0.6