Suggested Solution for Final lab – Check-point1

extarct1.sql

```
DROP TABLE padat;
CREATE TABLE
     padat(
       Loan_Type VARCHAR(20),
       Loan_Purpose VARCHAR(25),
       Loan_Amount_inK INTEGER,
       Preapproval VARCHAR(25),
       Action_Type VARCHAR(25),
       County_Name VARCHAR(50),
       Applicant_Ethnicity VARCHAR(25),
       Applicant_Race_1 VARCHAR(25),
       Applicant_Sex VARCHAR(25),
       Applicant_Income_inK VARCHAR(4),
       Rate_Spread VARCHAR(5),
       HOEPA_Status VARCHAR(1),
       Lien_Status VARCHAR(25),
       Minority_Population_pct VARCHAR(6),
       HUD_Median_Family_Income VARCHAR(8),
       Tract_To_MSAMD_Income_pct VARCHAR(6),
       Number_of_Owner_occupied_units VARCHAR(8)
     );
INSERT INTO padat
SELECT
       a.value,
       b.value,
       l.loan_amount_ink ,
       p.value ,
       ac.value ,
       c.county_name ,
       e.value ,
       r.value ,
       s.value ,
       l.applicant_income_ink ,
       l.rate_spread ,
       l.hoepa_status ,
       ls.value ,
       l.minority_population_pct ,
       l.hud_median_family_income ,
       l.tract_to_msamd_income_pct ,
       l.number_of_owner_occupied_units
FROM
larDB3 l
JOIN loantype a ON (a.code = l.Loan_Type)
JOIN loanpurpose b ON (b.code = l.Loan_Purpose)
JOIN preapproval p ON (p.code = l.Preapproval)
JOIN action ac ON (ac.code = l.Action_Type)
JOIN lienstatus ls ON (ls.code = l.Lien_Status)
JOIN ethnicity e ON (e.code = l.Applicant_Ethnicity)
```

```
JOIN race r ON (r.code = l.Applicant_Race_1)
JOIN sex s ON (s.code = l.Applicant_Sex)
JOIN counties c ON (c.State_Code = l.state_code AND c.County_Code =
l.County_Code AND l.Loan_Type = a.code)

WHERE
  (l.state_code = 42 AND
   l.Property_Type = 1 AND
   l.Occupancy = 1 AND
   l.Action_Type <= 4)
;
```

Suggested Solution for Final lab – Check-point2 and Check-point3

Finallabpart1.R

```
setwd("~/finallab")

#
#  Data Processing
#

#
#


#
#
# My query already restricted to:
# Property_Type = 1 : 1-4 family
# Occupancy = 1 : owner-occupied
# Action_Type <= 4: (1) loan originated (2) application approved but not
accepted
#                   (3) application denied (4) application withdrawn by
applicant

#
library('RODBC')

ch <- odbcConnect("Greenplum",uid="gpadmin",
      case="postgresql",pwd="changeme")
paRaw <- (sqlFetch(ch,"padat",as.is=T))
odbcClose(ch)


#
# Let's turn some of the codes into factors with names
#
#
#
paRaw$loan_type = factor(paRaw$loan_type)
paRaw$loan_purpose = factor(paRaw$loan_purpose)
paRaw$preapproval = factor(paRaw$preapproval)
paRaw$action_type = factor(paRaw$action_type)
paRaw$county_name=factor(paRaw$county_name)
paRaw$lien_status = factor(paRaw$lien_status)
paRaw$applicant_ethnicity = factor(paRaw$applicant_ethnicity)
paRaw$applicant_race_1 = factor(paRaw$applicant_race_1)
paRaw$applicant_sex = factor(paRaw$applicant_sex)
#
# an example of how to check quickly. the diagonal of the table will
# show how the codes match to the factor names
#
with(paRaw, table(loan_type, applicant_ethnicity))


#
# convert income and rate spread to numeric. Check the NAs.
```

```
#
paRaw$applicant_income_ink = as.numeric(paRaw$applicant_income_ink)
paRaw$rate_spread = as.numeric(paRaw$rate_spread)
paRaw$tract_to_msamd_income_pct = as.numeric(paRaw$tract_to_msamd_income_pct)
paRaw$number_of_owner_occupied_units =
as.numeric(paRaw$number_of_owner_occupied_units)
paRaw$minority_population_pct = as.numeric(paRaw$minority_population_pct)

with(paRaw, {
  print(paste("no income:",
sum(is.na(applicant_income_ink))/length(applicant_income_ink)))
  print(paste("low rate spread:",
sum(is.na(rate_spread))/length(rate_spread)))
  print(paste("tract to MSA income",
sum(is.na(tract_to_msamd_income_pct))/length(tract_to_msamd_income_pct)))
  print(paste("owner occupied units",
sum(is.na(number_of_owner_occupied_units)/length(number_of_owner_occupied_uni
ts))))
  print(paste("minority population",
sum(is.na(minority_population_pct)/length(minority_population_pct))))
})

# most people are NA on the rate spread, small minorities everywhere else
# 1] "no income: 0.0392906709656305"
# [1] "low rate spread: 0.983923329757597"
# [1] "tract to MSA income 0.00585855035222009"
# [1] "owner occupied units 0.00587269168065648"
# [1] "minority population 0.00579794465892126"
with(paRaw, table(highrate = !is.na(rate_spread), action_type))
#
#  of loans either originated/denied -- 3:1 ratio
#  2% of originated loans at reportable rate spread
#

# with(paRaw, table(highrate = !is.na(Rate_Spread), action_type)
# + )
#         actiontype
# highrate Originated Approved Not Accepted Denied Withdrawn
#    FALSE     303028                24835 100775     58407
#    TRUE        7958                    0      0         0
# > 7958/393928

# save it
save(paRaw, file="paRaw.RData")

#---------------------------------------------
#
# Data analysis.
#

load("paRaw.RData")  # if they have come from another session..

# library(ggplot2)  # for my own plotting purposes. not part of the official
lab

cnames = c("action_type", "loan_type", "loan_purpose", "loan_amount_ink",
"preapproval",
```

```
           "county_name", "applicant_income_ink", "lien_status",
"applicant_race_1", "applicant_ethnicity", "applicant_sex",
           "tract_to_msamd_income_pct", "number_of_owner_occupied_units",
"minority_population_pct",
           "hud_median_family_income", "hoepa_status", "rate_spread")
prob1data = paRaw[, cnames]

# let's just get rid of the people without income info
prob1data = subset(prob1data, !is.na(prob1data$applicant_income_ink))

#
# relevel county, race and ethnicity -- this sets Allegheny County (where
Pittsburgh is),
# white and non hispanic latino to be the first category
# value in their respective lists, hence they will be folded into the
"reference situation" for the logistic model.
# Only necessary if they are doing a logistic model, and honestly, not
strictly necessary even then.
# The model will work fine without the releveling, but this makes some of the
explanation easier, perhaps.
#
prob1data$county_name = relevel(prob1data$county, "Allegheny")
prob1data$applicant_race_1 = relevel(prob1data$applicant_race_1, "White")
prob1data$applicant_ethnicity = relevel(prob1data$applicant_ethnicity, "Non
Hispanic.Latino")

# make hud median family income (MSA level) numeric. some nulls
prob1data$hud_median_family_income =
as.numeric(prob1data$hud_median_family_income)

tmp = sum(with(prob1data, is.na(tract_to_msamd_income_pct) |
is.na(minority_population_pct) | is.na(hud_median_family_income)))
tmp/dim(prob1data)[1]  # 0.006. nuke them

# remove rows with nulls in msa income, minority pop, tract to msa pct
tmp = subset(prob1data, !(is.na(tract_to_msamd_income_pct) |
is.na(minority_population_pct) | is.na(hud_median_family_income)))
prob1data=tmp

# make the estimate of tract median income -- this is a driver that I thought
of; it turns out not to be significant.
# not strictly necessary by the students, but if we can breadcrumb them to
it, it is a good exercise in
# being creative about variable creation/selection
tract_median_income = with(prob1data,
hud_median_family_income*(tract_to_msamd_income_pct/100))
prob1data$tract_median_income_ink = round(tract_median_income/1000)  # useful
to have it in the same units as loan amount and income

#ggplot(prob1data) + geom_density(aes(x=tract_median_income_inK)) +
scale_x_log10()  # reasonably normalish
# one of the following 2
den = density(prob1data$tract_median_income_ink)
plot(den, log="x")
# or
den = density(log10(prob1data$tract_median_income_inK))
plot(den)
```

```
#
# visualize the variables. They should visualize all (or many, at least) of
them.
# I'm skipping the details here, and cutting to the chase.
# We observe that loan amount has a very odd, multi-modal distribution. this
suggests
# that we have multiple borrower populations. This suggests to us that we
might
# want to build separate models for the different loan purposes. Let's check.
#


# ggplot(prob1data, aes(x=Loan_Amount_inK)) + geom_density(adjust=0.5) +
scale_x_log10()
plot(density(log10(prob1data$loan_amount_ink)))

# look how each loan amount is distributed.
# ggplot(prob1data, aes(x=Loan_Amount_inK, colour=loanpurpose)) +
geom_density(adjust=0.5) + scale_x_log10()
with(prob1data,  {
 homepurchase = density(log10(subset(loan_amount_ink, loan_purpose=="Home
purchase")))
 homeimprovement = density(log10(subset(loan_amount_ink, loan_purpose=="Home
improvement")))
 refinance = density(log10(subset(loan_amount_ink,
loan_purpose=="Refinancing")))

 plot(homepurchase, col="red")
 lines(homeimprovement, col="blue")
 lines(refinance, col="green")
})


# so let's drop home improvements, just to make the experiment cleaner
prob1data = subset(prob1data, !(prob1data$loan_purpose %in% "Home
improvement"))

# what is that spike at about 400K? Smooth up until then. That and past it is
one or two other populations.
# Let's try a model for below that spike. In principle, we can develop a
separate model beyond that spike
# (Note, they might also want to eliminate some of the very small loans that
trail out on the right.
# I didn't do that here, but it would be a fair thing for them to do)
filter = prob1data$loan_amount_ink <= 400
# ggplot(prob1data[filter,], aes(x=Loan_Amount_inK, colour=loanpurpose)) +
geom_density(adjust=0.5) + scale_x_log10()
plot( density(log10(prob1data[filter,c("loan_amount_ink")])) )

sum(filter)/length(filter) # we lose about 4% of the loan data.

prob1data = prob1data[filter,]

#
# subset to only originated and denied: reasoning -- borrowers will take most
advantageous loan they can.
```

```
# This might be part of the starting scenario
#

filter = with(prob1data, action_type %in% c("Originated", "Denied"))
prob1data = prob1data[filter,]
prob1data$approved = prob1data$action_type=="Originated"
table(prob1data$approved)/dim(prob1data)[1]
#     FALSE      TRUE
# 0.2217928 0.7782072
```

Finallabpart2.R

```
#
# Full model. User supplies purpose, amount, income, loantype (default to
conventional), lienstatus of house (defaults to first), zip of property
#             From zip code we would discover county, minority population and
tract median income
#             The question is -- do we ask race/sex/ethnicity, and does it
improve the prediction appreciably?
#             Not to mention -- can we get decent probabilities out of this
at all?
#
dependentVar = "approved"
driversFull = c("loan_type", # this is assuming the user would know what type
of loan they were applying for...
                "loan_purpose",
                "log10(loan_amount_ink)",
                "log10(applicant_income_ink)",
                "lien_status",
                "applicant_race_1",
                "applicant_ethnicity",
                "applicant_sex",
                "county_name",
                "minority_population_pct",
                "log10(tract_median_income_ink)")
fmlaFull = paste(dependentVar, "~", paste(driversFull, collapse=" + "))

prob1data$gp = runif(dim(prob1data)[1])
smallset = subset(prob1data, prob1data$gp < 0.1) # 10% of data

fullmodel = glm(fmlaFull, data=smallset, family=binomial(link="logit"),
na.action=na.exclude)
summary(fullmodel)
#
# Only including the most significant counties (plus Philadelphia) for
clarity.
#
# Call:
# glm(formula = fmlaFull, family = binomial(link = "logit"), data = smallset,
#     na.action = na.exclude)
#
# Deviance Residuals:
#     Min       1Q   Median       3Q      Max
# -2.9050   0.3025   0.5531   0.7198   2.8346
#
# Coefficients:
#                                  Estimate Std. Error z value Pr(>|z|)
# (Intercept)                    -1.121e+00  2.927e-01  -3.830 0.000128 ***
# loantypeFHA                    -5.566e-01  3.704e-02 -15.024  < 2e-16 ***
# loantypeVA                     -7.727e-01  9.846e-02  -7.848 4.24e-15 ***
# loantypeFSA.RHS                -6.403e-01  1.596e-01  -4.013 6.00e-05 ***
# loanpurposeRefinancing         -1.325e+00  3.982e-02 -33.280  < 2e-16 ***
# log10(Loan_Amount_inK)         -1.149e+00  7.089e-02 -16.204  < 2e-16 ***
# log10(Applicant_Income_inK)     1.938e+00  6.785e-02  28.558  < 2e-16 ***
# lienstatusSubordinate lien     -9.685e-01  6.387e-02 -15.164  < 2e-16 ***
```

```
# raceAmerInd.AlaskaNat         -9.843e-01  2.940e-01  -3.348 0.000814 ***
# raceAsian                     -2.448e-01  9.233e-02  -2.651 0.008022 **
# raceBlack.AfroAmer            -5.542e-01  7.270e-02  -7.624 2.46e-14 ***
# raceHawaiian.PacificIs        -4.007e-01  2.966e-01  -1.351 0.176716
# raceNo Info                   -3.088e-01  9.297e-02  -3.322 0.000895 ***
# raceNot Applicable            -1.053e+01  2.541e+02  -0.041 0.966947
# ethnicityHispanic.Latino      -4.403e-01  8.824e-02  -4.990 6.05e-07 ***
# ethnicityNo Info              -3.752e-01  9.271e-02  -4.047 5.19e-05 ***
# ethnicityNot Applicable        9.846e+00  1.970e+02   0.050 0.960133
# sexFemale                     -1.372e-01  3.192e-02  -4.299 1.71e-05 ***
# sexNo Info                    -1.279e-02  7.667e-02  -0.167 0.867518
# sexNot Applicable              1.015e+01  1.360e+02   0.075 0.940510
# county027                      4.916e-01  1.494e-01   3.291 0.000998 ***
Centre County (small, center of state (duh). Rural?)
# county041                      4.077e-01  1.032e-01   3.952 7.74e-05 ***
Cumberland County (~ quarter Allegheny. Harrisburg environs)
# county043                      3.695e-01  1.059e-01   3.489 0.000485 ***
Dauphin County  (~ quarter Allegheny. Harrisburg)
# county071                      4.354e-01  8.387e-02   5.191 2.09e-07 ***
Lancaster County (~ half Allegheny. Southeast corner, btwn Harrisburg &
Philly)
# county089                     -4.852e-01  1.120e-01  -4.333 1.47e-05 ***
Monroe County (Along Northeast boundary of state)
# county091                      2.415e-01  7.093e-02   3.405 0.000661 ***
Montgomery County (The size of Allegheny. Philly environs)
# county101                      2.199e-01  7.247e-02   3.034 0.002411 **
Philadelphia County (Larger than Allegheny. Philadelphia)
# county103                     -6.062e-01  1.569e-01  -3.863 0.000112 ***
Pike County (Northeast corner, above Monroe)
# Minority_Population_pct       -2.521e-03  1.153e-03  -2.187 0.028710 *
# log10(tract_median_income_inK)  1.381e+00  1.571e-01   8.791  < 2e-16 ***
# ---
# Signif. codes:  0 â€˜***â€™ 0.001 â€˜**â€™ 0.01 â€˜*â€™ 0.05 â€˜.â€™ 0.1
â€˜ â€™ 1
#
# (Dispersion parameter for binomial family taken to be 1)
#
#     Null deviance: 35640  on 33437  degrees of freedom
# Residual deviance: 32100  on 33350  degrees of freedom
# AIC: 32276

attributes(fullmodel)  # get me the names of the 'class members'

# pseudo R2. Explains 10% of deviance. Not great, but it's explaining
something ... I should look at ROC and calculate AUC here too...
1- with(fullmodel, deviance/null.deviance)
# [1] 0.09932954

# Let's build an indicator vector for the variables of interest. Makes my
life easier
coefnames = names(fullmodel$coefficients)
countiesIX = grep("^county_name", coefnames) # all the counties
notcounties = coefnames[-countiesIX] # all the coefnames that are not
counties
countiesOfInterest = c("county_nameCarbon", "county_nameCentre",
"county_nameColumbia", "county_nameCumberland",
                       "county_nameDauphin",
```

```
                        "county_nameLancaster", "county_nameLehigh",
"county_nameMonroe","county_nameVenango")

driversIwant = c(notcounties, countiesOfInterest)

# ok, let's look at the driver impacts. (via the exp of the coefficients)
exp(fullmodel$coefficients[driversIwant])
#                  (Intercept)                   loantypeFHA
loantypeVA
#                3.259644e-01                   5.731804e-01
4.617593e-01
#              loantypeFSA.RHS         loanpurposeRefinancing
log10(Loan_Amount_inK)
#                5.271425e-01                   2.657513e-01
3.170294e-01
#    log10(Applicant_Income_inK)     lienstatusSubordinate lien
raceAmerInd.AlaskaNat
#                6.942041e+00                   3.796461e-01
3.737038e-01
#                    raceAsian             raceBlack.AfroAmer
raceHawaiian.PacificIs
#                7.828837e-01                   5.745092e-01
6.698795e-01
#                  raceNo Info             raceNot Applicable
ethnicityHispanic.Latino
#                7.343172e-01                   2.669520e-05
6.438321e-01
#             ethnicityNo Info        ethnicityNot Applicable
sexFemale
#                6.871751e-01                   1.887957e+04
8.717528e-01
#                   sexNo Info              sexNot Applicable
Minority_Population_pct
#                9.872915e-01                   2.555030e+04
9.974819e-01
# log10(tract_median_income_inK)                     county027
county041
#                3.978892e+00                   1.634930e+00
1.503369e+00
#                    county043                      county071
county089
#                1.447017e+00                   1.545528e+00
6.155632e-01
#                    county091                      county101
county103
#                1.273185e+00                   1.245936e+00
5.454279e-01

#
#  Note: the analysis below is not strictly needed for the lab, but it is a
good example
#  of how one might analyze the coefficients. They need to analyze the value
of certain
#  coeffcients, to generate suggestions for the FPC advice page.
#
```

```
# Reference situation: White, non-latino male, with no income and a conventional loan of zero dollars for home purchase,
# living in Allegheny County in a tract with no minorities and no tract income.
# (I'm not going to pretend that makes sense... This is one of the probematic aspects of model interpretation).
# Baseline odds (that is : P(accept)/P(deny)) of loan acceptance for him is the intercept: 3:10 (meaning 3 out of every 13 loans are accepted)
#
# Consider effect sizes. General rule of thumb: magnitude near 1 means not much effect,
#  large magnitude above 1 increases odds of acceptance, small magnitude below 1 increases odds of denial (by 1/coef).
#
# Taking a loan other than conventional seems to cut the odds of getting
# the loan approved by around a half.
# A refinancing is 5 times the odds of being denied (1/0.2), all other things being equal.
#
# Being African American increases odds of denial by 10/6 = 1 5/6, (relative to White) not providing the info reduces the odds somwhat.
# Asians seem to have slightly higher odds of denial, relative to White, as well.
# For other races, there is either less impact, or the coefficients don't meet the significance test, or the population
# in this data set is vanishingly small.
#
# Latino ethnicity has a mild negative impact, being female a smaller impact (relative to Male).
# sex "Not Applicable" (whatever that means) has a ridiculous coefficient. Overfitting. Only 46 of them in the
# data set anyway. (I probably should have removed those rows before fitting, but I shall carry on...)
#
# Minority population pct close to 1: no impact
#
# Every increase of 10K in loan amount increases odds of denial by 10/3 = 3.33; every increase in 10K of income increases
# odds of acceptance by almost 7 (!). Every increase in tract income of 10K increases odds by almost 4.
#
# All the counties of interest have somewhat higher odds of loan acceptance than Allegheny, except Monroe and Pike
#
# All of this suggests that there is a correlation with race/ethnicity/sex, even when accounting for income and tract affluence.
# Since there are probably correlations between race/ethnicity, minority population, tract income, county, it is possible
# the other locale related variables can pick up the slack even if personal demographic information is removed.
#

#
# try a model without the personal demographics
#
driversNoPersonal = c("loan_type", # this is assuming the user would know what type of loan they were applying for...
```

```
              "loan_purpose",
              "log10(loan_amount_ink)",
              "log10(applicant_income_ink)",
              "lien_status",
              # "race",
              # "ethnicity",
              # "sex",
              "county_name",
              "minority_population_pct",
              "log10(tract_median_income_ink)")
fmlaNoPersonal = paste(dependentVar, "~", paste(driversNoPersonal, collapse="
+ "))

modelNoPersonal = glm(fmlaNoPersonal, data=smallset,
family=binomial(link="logit"), na.action=na.exclude)
summary(modelNoPersonal)
#
# Call:
# glm(formula = fmlaNoPersonal, family = binomial(link = "logit"),
#     data = smallset, na.action = na.exclude)
#
# Deviance Residuals:
#     Min       1Q   Median       3Q      Max
# -2.9162   0.3097   0.5673   0.7241   2.7360
#
# Coefficients:
#                                 Estimate Std. Error z value Pr(>|z|)
# (Intercept)                   -1.211e+00  2.890e-01  -4.191 2.77e-05 ***
# loantypeFHA                   -5.998e-01  3.654e-02 -16.415  < 2e-16 ***
# loantypeVA                    -8.348e-01  9.696e-02  -8.610  < 2e-16 ***
# loantypeFSA.RHS               -6.168e-01  1.592e-01  -3.873 0.000107 ***
# loanpurposeRefinancing        -1.344e+00  3.946e-02 -34.069  < 2e-16 ***
# log10(Loan_Amount_inK)        -1.117e+00  7.040e-02 -15.868  < 2e-16 ***
# log10(Applicant_Income_inK)    1.981e+00  6.699e-02  29.574  < 2e-16 ***
# lienstatusSubordinate lien    -9.723e-01  6.345e-02 -15.324  < 2e-16 ***
# [... counties omitted ...]
# Minority_Population_pct       -6.455e-03  1.067e-03  -6.050 1.45e-09 ***
# log10(tract_median_income_inK)  1.296e+00  1.558e-01   8.319  < 2e-16 ***
# ---
# Signif. codes:  0 â€˜***â€™ 0.001 â€˜**â€™ 0.01 â€˜*â€™ 0.05 â€˜.â€™ 0.1
â€˜ â€™ 1
#
# (Dispersion parameter for binomial family taken to be 1)
#
#     Null deviance: 35640  on 33437  degrees of freedom
# Residual deviance: 32412  on 33362  degrees of freedom
# AIC: 32564

# pseudo-R2
1- with(modelNoPersonal, deviance/null.deviance)
# [1] 0.09055168 -- from 10% to 9%

removed = c(grep("^applicant_race", driversIwant), grep("^applicant_sex",
driversIwant), grep("^applicant_ethnicity", driversIwant))
driversIwant = driversIwant[-removed]
exp(modelNoPersonal$coefficients[driversIwant])
```

```
#                  (Intercept)                         loantypeFHA
loantypeVA
#                  0.2977522                           0.5489482
0.4339603
#               loantypeFSA.RHS         loanpurposeRefinancing
log10(Loan_Amount_inK)
#                    0.5396797                           0.2607444
0.3272130
#    log10(Applicant_Income_inK)     lienstatusSubordinate lien
Minority_Population_pct
#                    7.2508144                           0.3782146
0.9935663
# log10(tract_median_income_inK)                         county027
county041
#                    3.6544696                           1.7079857
1.5343414
#                    county043                           county071
county089
#                    1.4424620                           1.5949323
0.5869427
#                    county091                           county101
county103
#                    1.2677475                           1.1696274
0.5446403
#
# Intercept and applicant income coefficient is the only one that moved much
(both by about 0.3), which suggests
# a correlation between personal demographics and income can pick up much of
the lost information.
# Minority Population pct has even less impact in this model (which is rather
a relief).
#

#
# Ok. I'll look at the ROC curves ...
#
library(ROCR)
# the full model
# make the prediction object required by ROCR
predFull = prediction(predict(fullmodel, type="response"), smallset$approved)
rocFull = performance(predFull, "tpr", x.measure="fpr")  # fpr on x-axis, tpr
on y-axis
aucFull = performance(predFull, "auc")

# the model without personal demographics
predNP = prediction(predict(modelNoPersonal, type="response"),
smallset$approved)
rocNP = performance(predNP, "tpr", x.measure="fpr")  # fpr on x-axis, tpr on
y-axis
aucNP = performance(predNP, "auc")

# the aucs
aucFull@y.values[[1]]  # the auc value -  0.7117478
aucNP@y.values[[1]] # 0.703762. Not a huge difference

# plot the ROC curve. Base graphics
```

```
plot(rocFull@x.values[[1]], rocFull@x.values[[1]], type="l", col='gray',
xlab="fpr", ylab="tpr" )
    # x=y line for reference
plot(rocFull, text.col="green", col="green", add=T)
    # add=T adds it to existing plot, rather than making a new one
    # cutoffs are the points on the ROC curve corresponding to using 0.5 or
0.75 as score thresholds
    # for approval. not strictly needed here, but nice to see.
plot(rocNP, text.col="blue", col="blue", add=T)

# The modelNoPersonal curve lies just inside the full model curve --
essentially they are the same model

# ----------------------------------------------------------------------------
------
# FYI, The following creates the graph I actually look at.  It's gonna look
bad....
# (We don't need to add this to the lab -- ROC is enough)

predFull = predict(fullmodel, type="response")
predNP = predict(modelNoPersonal, type="response")

pframe = data.frame(approved=smallset$approved, fullmodel=predFull,
noPersonal=predNP)
pframelong = melt(pframe, measure.var=c("fullmodel", "noPersonal"))
colnames(pframelong) = c("approved", "model", "score")
# compare the score densities for approved and denied for each model
ggplot(pframelong) + geom_histogram(aes(x=score, fill=approved),
position="identity", alpha=0.5) + facet_wrap(~model)

# no separation at all. More evidence that we don't have the data that really
predicts
# loan disposition (FICA, existing debt, etc). But we knew that.
# ----------------------------------------------------------------------------
-------

#
# Back to the lab. To answer the question -- do we *need* to query for
personal demographic info?
# clearly, the answer is no. we can do just as good (or as bad) a job of
predicting outcome
# with less controversial data: income, loan size, zip code
#

#
# Let's examine the probability thresholds suggested by marketing.
#
 # preset the probability scores.
hithresh = 0.75
lothresh = 0.5

# do the evaluation on a holdout set
holdout = subset(prob1data, gp > 0.75) # about 25% of the data
predHO = predict(modelNoPersonal, newdata=holdout, type="response")

binsHO = cut(predHO, breaks = c(0, lothresh, hithresh, 1.0), labels = c("low
prob", "med prob", "hi prob"), include.lowest=T)
```

```
# confusion matrix
tab = table(outcome = holdout$approved, scorebin = binsHO)
tab
#        scorebin
# outcome low prob med prob hi prob
#   FALSE     2061     8467    8157
#   TRUE      1154    16380   48035

# sum up how many people ended up in each bin
binSums = colSums(tab)
# the proportions
binSums/sum(binSums)
#   low prob   med prob    hi prob
# 0.03815843 0.29490588 0.66693569

# the second row of the confusion matrix, divided by the number of
# people in each bin. This gives us the probability of approval in each bin
# As we desire, the probability of getting a loan in the high prob bin is
# indeed high (higher than 75%, at least). And the probability of getting
# a loan in the low prob bin is low.

tab[2,]/binSums
#  low prob  med prob   hi prob
# 0.3589425 0.6592345 0.8548370
```