Homework 3

2025-10-04

Priya Mantraratnam

Continuous target variable: Population, low access to store (% change), 2015 -19 [PCH_LACCESS_POP_15_19] [ACCESS]

Continuous predictor variables: nine variables with per capita alternatives
1. Grocery stores (% change), 2016-20 [PCH_GROC_16_20] [STORES] 2. Supercenters & club stores (% change), 2016-20 [PCH_SUPERC_16_20] 3. Convenience stores (% change), 2016-20 [PCH_CONVS_16_20] 4. Specialized food stores (% change), 2016-20 [PCH_SPECS_16_20] 5. SNAP-authorized stores (% change), 2017-23 [PCH_SNAPS_17_23] 6. WIC-authorized stores (% change), 2016-22 [PCH_WICS_16_22] 7. Fast-food restaurants (% change), 2016-20 [PCH_FFR_16_20] 8. Full-service restaurants (% change), 2016-20 [PCH_FSR_16_20] 9. Direct farm sales (% change), 2012 - 17 [PCH_DIRSALES_12_17] [LOCAL]

```r
install.packages("ggfortify", repos="http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/harip/AppData/Local/R/win-library/4.5'
## (as 'lib' is unspecified)

## package 'ggfortify' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\harip\AppData\Local\Temp\Rtmp6FVRQJ\downloaded_packages
```

```r
install.packages("mvnormtest", repos="http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/harip/AppData/Local/R/win-library/4.5'
## (as 'lib' is unspecified)

## package 'mvnormtest' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\harip\AppData\Local\Temp\Rtmp6FVRQJ\downloaded_packages
```

```r
install.packages("datarium", repos="http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/harip/AppData/Local/R/win-library/4.5'
## (as 'lib' is unspecified)

## package 'datarium' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\harip\AppData\Local\Temp\Rtmp6FVRQJ\downloaded_packages
```

```r
install.packages("ggplot2", repos="http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/harip/AppData/Local/R/win-library/4.5'
## (as 'lib' is unspecified)

## package 'ggplot2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
```

```
##  C:\Users\harip\AppData\Local\Temp\Rtmp6FVRQJ\downloaded_packages
```

```r
install.packages("car", repos="http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/harip/AppData/Local/R/win-library/4.5'
## (as 'lib' is unspecified)

## package 'car' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\harip\AppData\Local\Temp\Rtmp6FVRQJ\downloaded_packages
```

```r
library(MASS)
library(car)
```

```
## Loading required package: carData
```

```r
library(datarium)
library(ggplot2)
library(broom)
library(ggfortify)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.4     v tibble    3.3.0
## v purrr     1.1.0     v tidyr     1.3.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x dplyr::recode() masks car::recode()
## x dplyr::select() masks MASS::select()
## x purrr::some()   masks car::some()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(mvnormtest)
```

```r
library(readxl)
 X2025_food_environment_atlas_data <- read_excel("2025-food-environment-atlas-data.xlsx",
     sheet = "ACCESS", skip = 1)
 access <- read_excel("2025-food-environment-atlas-data.xlsx",
     sheet = "ACCESS", skip = 1)
 access
```

```
## # A tibble: 3,144 x 68
##    FIPS  State County   LACCESS_POP15 LACCESS_POP19 PCH_LACCESS_POP_15_19
##    <chr> <chr> <chr>            <dbl>         <dbl>                 <dbl>
##  1 01001 AL    Autauga         18093.        18503.                  2.27
##  2 01003 AL    Baldwin         46400.        45789.                 -1.32
##  3 01005 AL    Barbour          6684.         5634.                -15.7
##  4 01007 AL    Bibb              296.          365.                 23.5
##  5 01009 AL    Blount           5856.         3902.                -33.4
##  6 01011 AL    Bullock          6100.         7480.                 22.6
##  7 01013 AL    Butler           2478.         2508.                  1.23
##  8 01015 AL    Calhoun         34221.        42575.                 24.4
##  9 01017 AL    Chambers         6794.         6745.                 -0.720
```

```
## 10 01019 AL    Cherokee         3519.         3506.                -0.358
## # i 3,134 more rows
## # i 62 more variables: PCT_LACCESS_POP15 <dbl>, PCT_LACCESS_POP19 <dbl>,
## #   LACCESS_LOWI15 <dbl>, LACCESS_LOWI19 <dbl>, PCH_LACCESS_LOWI_15_19 <dbl>,
## #   PCT_LACCESS_LOWI15 <dbl>, PCT_LACCESS_LOWI19 <dbl>, LACCESS_HHNV15 <dbl>,
## #   LACCESS_HHNV19 <dbl>, PCH_LACCESS_HHNV_15_19 <dbl>,
## #   PCT_LACCESS_HHNV15 <dbl>, PCT_LACCESS_HHNV19 <dbl>, LACCESS_SNAP15 <dbl>,
## #   LACCESS_SNAP19 <dbl>, PCH_LACCESS_SNAP_15_19 <dbl>, ...
```

```r
library(readxl)
 X2025_food_environment_atlas_data <- read_excel("2025-food-environment-atlas-data.xlsx",
     sheet = "STORES", skip = 1)
 stores <- read_excel("2025-food-environment-atlas-data.xlsx",
     sheet = "STORES", skip = 1)
 stores
```

```
## # A tibble: 3,144 x 41
##    FIPS  State County   GROC16 GROC20 PCH_GROC_16_20 GROCPTH16  GROCPTH20
##    <chr> <chr> <chr>     <dbl>  <dbl>          <dbl>     <dbl>      <dbl>
##  1 01001 AL    Autauga       3      4           33.3    0.0542     0.0712
##  2 01003 AL    Baldwin      29     29            0      0.140      0.126
##  3 01005 AL    Barbour       4      5           25      0.155      0.203
##  4 01007 AL    Bibb          5      4          -20      0.221      0.181
##  5 01009 AL    Blount        5      4          -20      0.0870     0.0691
##  6 01011 AL    Bullock       3  -9999        -9999      0.289  -9999
##  7 01013 AL    Butler        3      3            0      0.150      0.154
##  8 01015 AL    Calhoun      27     21          -22.2    0.235      0.185
##  9 01017 AL    Chambers      7      5          -28.6    0.207      0.152
## 10 01019 AL    Cherokee      5  -9999        -9999      0.194  -9999
## # i 3,134 more rows
## # i 33 more variables: PCH_GROCPTH_16_20 <dbl>, SUPERC16 <dbl>, SUPERC20 <dbl>,
## #   PCH_SUPERC_16_20 <dbl>, SUPERCPTH16 <dbl>, SUPERCPTH20 <dbl>,
## #   PCH_SUPERCPTH_16_20 <dbl>, CONVS16 <dbl>, CONVS20 <dbl>,
## #   PCH_CONVS_16_20 <dbl>, CONVSPTH16 <dbl>, CONVSPTH20 <dbl>,
## #   PCH_CONVSPTH_16_20 <dbl>, SPECS16 <dbl>, SPECS20 <dbl>,
## #   PCH_SPECS_16_20 <dbl>, SPECSPTH16 <dbl>, SPECSPTH20 <dbl>, ...
```

```r
library(readxl)
 X2025_food_environment_atlas_data <- read_excel("2025-food-environment-atlas-data.xlsx",
     sheet = "RESTAURANTS", skip = 1)
 restaurants <- read_excel("2025-food-environment-atlas-data.xlsx",
     sheet = "RESTAURANTS", skip = 1)
 restaurants
```

```
## # A tibble: 3,144 x 15
##    FIPS  State County   FFR16 FFR20 PCH_FFR_16_20 FFRPTH16 FFRPTH20
##    <chr> <chr> <chr>    <dbl> <dbl>         <dbl>    <dbl>    <dbl>
##  1 01001 AL    Autauga     44    45          2.27    0.796    0.801
##  2 01003 AL    Baldwin    156   172         10.3     0.751    0.750
##  3 01005 AL    Barbour     23    24          4.35    0.891    0.976
##  4 01007 AL    Bibb         7     7          0       0.310    0.316
##  5 01009 AL    Blount      23    24          4.35    0.400    0.415
##  6 01011 AL    Bullock      3     3          0       0.289    0.301
##  7 01013 AL    Butler      18    21         16.7     0.898    1.08
##  8 01015 AL    Calhoun     95   104          9.47    0.826    0.917
##  9 01017 AL    Chambers    29    32         10.3     0.859    0.974
```

```
## 10 01019 AL    Cherokee    15    18         20       0.582    0.685
## # i 3,134 more rows
## # i 7 more variables: PCH_FFRPTH_16_20 <dbl>, FSR16 <dbl>, FSR20 <dbl>,
## #   PCH_FSR_16_20 <dbl>, FSRPTH16 <dbl>, FSRPTH20 <dbl>, PCH_FSRPTH_16_20 <dbl>
```

```r
library(readxl)
 X2025_food_environment_atlas_data <- read_excel("2025-food-environment-atlas-data.xlsx",
     sheet = "LOCAL", skip = 1)
 local <- read_excel("2025-food-environment-atlas-data.xlsx",
     sheet = "LOCAL", skip = 1)
 local
```

```
## # A tibble: 3,161 x 98
##    FIPS  State County   DIRSALES_FARMS12 DIRSALES_FARMS17 PCH_DIRSALES_FARMS_1~1
##    <chr> <chr> <chr>               <dbl>            <dbl>                  <dbl>
##  1 01001 AL    Autauga                51               16                  -68.6
##  2 01003 AL    Baldwin               103               78                  -24.3
##  3 01005 AL    Barbour                13                9                  -30.8
##  4 01007 AL    Bibb                   13               11                  -15.4
##  5 01009 AL    Blount                 88               40                  -54.5
##  6 01011 AL    Bullock                12                2                  -83.3
##  7 01013 AL    Butler                 31               20                  -35.5
##  8 01015 AL    Calhoun                50               52                    4
##  9 01017 AL    Chambers               22               13                  -40.9
## 10 01019 AL    Cherokee               14               14                    0
## # i 3,151 more rows
## # i abbreviated name: 1: PCH_DIRSALES_FARMS_12_17
## # i 92 more variables: PCT_LOCLFARM12 <dbl>, PCT_LOCLFARM17 <dbl>,
## #   PCT_LOCLSALE12 <dbl>, PCT_LOCLSALE17 <dbl>, DIRSALES12 <dbl>,
## #   DIRSALES17 <dbl>, PCH_DIRSALES_12_17 <dbl>, PC_DIRSALES12 <dbl>,
## #   PC_DIRSALES17 <dbl>, PCH_PC_DIRSALES_12_17 <dbl>, FMRKT13 <dbl>,
## #   FMRKT18 <dbl>, PCH_FMRKT_13_18 <dbl>, FMRKTPTH13 <dbl>, ...
```

```r
atlasog = merge(access, stores, by.x = "FIPS", by.y = "FIPS")
atlasog = merge(atlasog, restaurants, by.x = "FIPS", by.y = "FIPS")
atlasog = merge(atlasog, local, by.x = "FIPS", by.y = "FIPS")
```

```
## Warning in merge.data.frame(atlasog, local, by.x = "FIPS", by.y = "FIPS"):
## column names 'State.x', 'County.x', 'State.y', 'County.y' are duplicated in the
## result
```

Original model:

```r
atlas = select(atlasog, PCH_LACCESS_POP_15_19, PCH_GROC_16_20, PCH_SUPERC_16_20, PCH_CONVS_16_20, PCH_SP
atlas = filter(atlas, PCH_LACCESS_POP_15_19 != -9999, PCH_GROC_16_20 != -9999, PCH_SUPERC_16_20 != -9999
atlas = filter(atlas, PCH_LACCESS_POP_15_19 != -8888, PCH_GROC_16_20 != -8888, PCH_SUPERC_16_20 != -8888
atlas2 = filter(atlas, PCH_LACCESS_POP_15_19 <5000, PCH_GROC_16_20 <5000, PCH_SUPERC_16_20 <5000, PCH_CO

colMeans(atlas2)
```

```
## PCH_LACCESS_POP_15_19        PCH_GROC_16_20      PCH_SUPERC_16_20
##             0.2320956            -0.7079341            68.9385445
##       PCH_CONVS_16_20       PCH_SPECS_16_20        PCH_SNAPS_17_23
##             2.6007777            -0.6579917            11.6952787
##        PCH_WICS_16_22         PCH_FFR_16_20         PCH_FSR_16_20
##            -5.6402305             6.6306411             1.8834212
##    PCH_DIRSALES_12_17
```

```
##          106.6679969
```

1. Variable Selection (R) Consider a dataset with a large number of predictor variables. Perform the following variable selection methods: Best Subset Selection, Sequential Selection Methods, Ridge, Lasso, Principal Components Regression, Partial Least Squares. Compare and contrast them for variable selection on the same dataset. Discuss the selected variables and their performance.

Best Subset Selection:

The subset of size 1 that gives smallest residual sum of squares includes only fast food stores. For a subset of size 2, we add grocery stores. For size 3, full-service restaurants, then convenience stores, then SNAPS-accepting stores, then WICS-accepting, then direct farm sales, and finally specialized stores. The RSS is relatively similar across subsets, but it is worth noting that the WICS data does not appear to have a high correlation with any other variables on the scatterplot.

```r
install.packages("leaps",repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/harip/AppData/Local/R/win-library/4.5'
## (as 'lib' is unspecified)

## package 'leaps' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'leaps'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\harip\AppData\Local\R\win-library\4.5\00LOCK\leaps\libs\x64\leaps.dll
## to C:\Users\harip\AppData\Local\R\win-library\4.5\leaps\libs\x64\leaps.dll:
## Permission denied

## Warning: restored 'leaps'

##
## The downloaded binary packages are in
##  C:\Users\harip\AppData\Local\Temp\Rtmp6FVRQJ\downloaded_packages
```

```r
install.packages("glmnet",repos = "http://cran.us.r-project.org")
```

```
## Installing package into 'C:/Users/harip/AppData/Local/R/win-library/4.5'
## (as 'lib' is unspecified)

## package 'glmnet' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'glmnet'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\harip\AppData\Local\R\win-library\4.5\00LOCK\glmnet\libs\x64\glmnet.dll
## to C:\Users\harip\AppData\Local\R\win-library\4.5\glmnet\libs\x64\glmnet.dll:
## Permission denied

## Warning: restored 'glmnet'

##
## The downloaded binary packages are in
##  C:\Users\harip\AppData\Local\Temp\Rtmp6FVRQJ\downloaded_packages
```

```r
library(MASS)
library(leaps)
library(glmnet)
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
## Loaded glmnet 4.1-10
```

```r
Y <- as.numeric(atlas2[,1])
X <- as.matrix(atlas2[,-1])

Y <- Y- mean(Y)
Y
```

```
##   [1]  -1.54955140  -8.42069617  -0.31177433  10.07001313  -3.96253697
##   [6]  -4.63784639  -0.02331065 -19.01504890  22.85461625  11.07688626
##  [11] -10.53748599  -0.56003359   6.07112130   5.04149684   0.35615298
##  [16]   9.49653157   1.97278293  -3.97360746   1.88305720   6.25111350
##  [21]   2.13741454  11.01591596  -2.74741522  13.26399430  42.89198884
##  [26]   4.41279515   5.87820824  -0.19596140  -3.41769472  -1.41945890
##  [31] -16.79631606   2.28695258   7.07438144   7.46880397  -6.24996129
##  [36]  -8.73679343 -16.55705634  -1.58497504  14.25566682  -1.50904289
##  [41]  -2.00114325  -8.59137431   3.31988462   1.34578499  -4.17591563
##  [46]  -0.54371822   6.46370562  12.26215371   1.93530377   8.84354695
##  [51]  -4.65690604   6.90383777 -15.56608573 -19.32853690 -25.47346870
##  [56] -10.47300616  -7.58239308   0.37886706 -19.05877486  -6.05604402
##  [61] -11.97629348  -9.22649089 -10.46015826  -0.20760735   3.74072369
##  [66]   7.47980985  -5.40384475  -4.79450313   3.45786532  -3.54116288
##  [71]  -2.00949863   1.86794671   3.52407774  -6.75043574  -0.72298277
##  [76]  -1.10681507  14.01112470  -3.88822404 116.63695344   4.27314433
##  [81]  12.93036660  -2.90132037  -5.78159228   1.81493148   1.00338873
##  [86] -14.84337035 -15.89561359   3.70482596   0.86282846  -4.46914378
##  [91]  -2.78720490   5.39839085  -3.23085967   0.97212097   4.83406409
##  [96]   6.42107304 -10.72987071   2.43071469  -0.80083612   1.51884493
## [101] -15.08095447  -5.72086803   2.26877054  10.04851350  -3.92543307
## [106]   2.43061575  -7.38814202  -8.51007072  -9.83180514 -16.85493079
## [111]   0.97421321   3.26750382  -3.53996531  -3.26686088  -1.42556134
## [116]  -3.36134020   3.05429467   5.15452489  -3.73525969  18.03687676
## [121]   0.32666018  -2.72431770 -17.94923965   4.36290273 -18.17609397
## [126]   0.03645372 -21.86427108 -15.14158908   0.15794247  13.83246716
## [131]   4.23857030  -9.48423282   1.20300826 -16.11195174  10.48587521
## [136]   4.90376052  -2.84615532 -27.38164130  -5.68699876   1.48640272
## [141]  -4.13112394  -2.07373623  -0.66742614  -0.50564343  37.58550462
## [146]  13.44430932  -3.17268506  11.24370393   1.42393991  -6.63918964
## [151]   9.98732480  11.46509274  36.45784959  30.67425355  16.34264764
## [156]  18.77668198  -4.14752070   8.68066510  -2.39690343   8.86146363
## [161]   7.10744628 -17.07832900  22.50458154   4.14260062  -3.50864473
## [166]   9.64985761  13.17238339  30.23374375  30.21066865   9.83492288
## [171] -18.74587051  20.93605622  18.18497666  -0.03438764  -1.86255840
## [176]  47.00863084  33.52925309 -13.17245856  -8.08730975 -32.34263793
## [181]  -9.08797351   1.79182943  -7.85287324   2.70580825  -3.89236251
## [186] -29.71892921  -0.23209563 -10.74391261 -11.33706561  -0.99475185
## [191]  -1.25733021  41.49313172  10.48341283   4.84253320   5.65083608
## [196]  -4.27124826  -6.43146077  18.07390984  21.04268845   5.60056409
## [201]  -5.86247197  24.98252114  -2.70318905  -6.17452136   1.54212233
## [206]  -6.30469409  -2.62831560   3.99684247  -2.28159205   5.19331797
## [211]  19.73551949   3.42283305  -5.81308356  -9.12596313  -7.95384637
```

```
## [216]  -4.94127551  15.63862237   1.57324346 -11.48762218  22.21607980
## [221]  -1.63353912 -12.61595145   1.22568628   6.80469521   2.94190511
## [226]   5.75197085  -2.63461534  -2.66750208  -2.69036690  13.15810498
## [231] -18.42299453   2.29692301 104.46146973   2.53805884  -1.00861213
## [236]  -1.21594146  -3.34197584  -4.71548740  -0.95021078  -3.31259290
## [241]   0.78328689  -0.34271289  10.98403939   1.38646420  -5.35296098
## [246]   4.97937640  11.99350556   0.37435743  -0.25070663  -1.52267555
## [251]  -7.35531655  10.69061765  20.56641587  -9.51251689  -0.22489200
## [256]  -6.59277192 -15.55067245  -3.64592591  13.90726575   6.25516137
## [261]   4.02297982  -1.61295489  -2.56740681  12.76264008  -0.28235161
## [266]   9.02572640 140.58162507   2.19562610  -0.48302073   2.01098927
## [271]  -0.18641880   3.87363967  12.53957375  -6.63154546  -9.11359683
## [276]   3.26718768  -6.29280749  -6.66860334  -1.15818003  -0.68760217
## [281]   3.48517379  -9.85498992  10.24212941  -8.72534553 -13.39809981
## [286]  37.31947526   2.64532622  -3.27864782   1.19126769 -17.78131858
## [291]   5.12548598   5.74516162  -8.22423640  -0.35657939 -13.30803577
## [296]  -4.55630437  -8.91800300 -15.94113341 -17.20382110 -10.21511260
## [301]   5.02028617  -6.52220193 -27.09105292  37.32330140  12.77078923
## [306]   1.94744738 -11.85329524 -10.67189113  -0.76454822   3.82929715
## [311] -11.17945090   5.71314486  -0.69660575  -3.76367251   0.20860677
## [316]  -2.27654329  -3.16603318  -6.46731797  12.33593282 -16.17960158
## [321]   9.44767293  -6.29586021  -2.58474151  -2.91238943  -0.21068321
## [326]   3.82855138   7.47281464   9.17081364  -9.59036246  -0.07357489
## [331] -11.48701087  20.56862267  -6.37878362  -4.37653199   6.32566317
## [336] -11.99015609  22.90294274  -2.25161139 -63.36934844  -0.40052188
## [341]   2.26524385  -0.38219709  -4.32259456  -0.23209563   7.51370868
## [346]  -0.23209563  17.71827515   6.45703038  16.92393693 -10.27622691
## [351]  -0.50449238 -12.11335555  -0.32807889   5.53381642  24.75442895
## [356] -17.46907798  -0.78492901 -22.11463538   4.81697997  -7.64600268
## [361]  -7.30424539   4.07063826  -1.67890683   4.64539584   9.14020356
## [366]  -5.28906051  13.71781548  -2.62922565   7.51463517  -1.23456672
## [371]  -6.93676320  -7.91100541 -13.12242786  -8.50043670 -12.63381377
## [376]   5.70410165   7.21418580  -0.65310580  -4.86486474   2.45927152
## [381]   1.56376346  -0.24266475  13.99254617 -33.32357207   3.53983482
## [386]   6.70313701   7.83973607 -17.24345580  -0.69699461   0.64619216
## [391] -18.27722350   8.70542916 -10.36892501  -5.27675668   2.60017380
## [396]   2.10753783 -11.48915092  -3.95127216   3.62798079 -19.09945480
## [401]  16.31523332  19.67610749 -14.22137538   9.34257325  -0.55067701
## [406]  20.35429009  -4.10603682   9.81895932   5.19690713  -2.27632919
## [411]  30.33983048  -7.10416833   7.49666318 -16.10685245 -12.09821883
## [416]  -2.82770387  -3.08485309 -10.77156821  24.07968530 -12.57123176
## [421] -11.01321880  17.67212685   6.21099433   4.96295079 -11.71806136
## [426]   4.52594336  -6.27925960  -4.65404407 -22.34643165   8.68512639
## [431]   2.38191136 -21.22278205  -0.71790189  14.15624436  -5.94324103
## [436]   0.69451013 -16.12832252   0.45653954 -34.18491164  11.65935811
## [441]  13.42434033  -7.57146922  -5.98938361   0.13188013   0.08421844
## [446]  -6.40509645  -4.92631093  23.81460389  -6.54357043   5.29483231
## [451] -57.46896926   3.57145699   7.33366021   9.95165166  -6.37980882
## [456]   4.67221984  25.49358186  22.00375756  -2.44427887   1.04753932
## [461]  -2.51468149 -12.69204036  22.71488961   2.84761914   0.61881873
## [466]  16.85270509   1.69908294   4.17731818  -2.12667230  -9.84707633
## [471] -15.28257743 -15.88211528   3.23586496  -2.64141623  -0.44919093
## [476]  21.36890992   0.28122434  -5.66956226  -1.53124229 -17.08127967
## [481]  25.79296311  -4.39741985 -11.54728023  12.07805070  -4.64399425
```

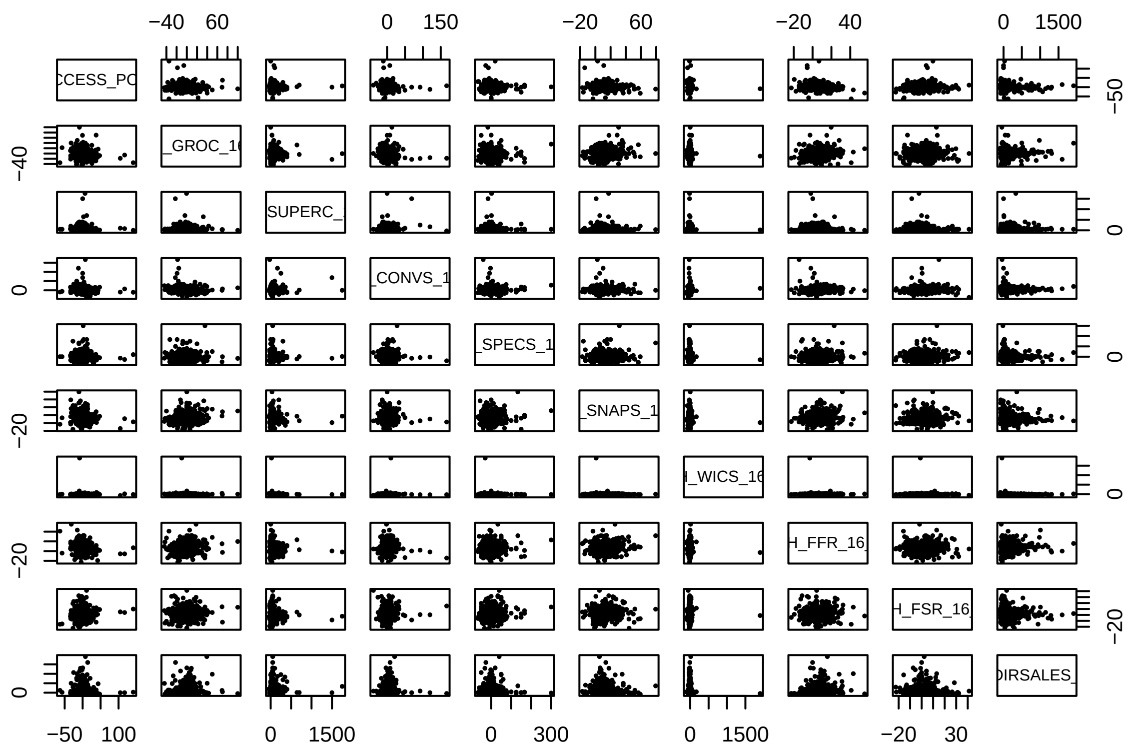
```
## [486]   4.28692731   1.09807404  -2.59531442  -7.45187799  -3.17637221
## [491] -22.35397330  -3.20806948  -5.29680577  -1.61590627   3.08164724
## [496]   9.23127278  10.10591897 -27.53666106 -13.35114947  13.81649217
## [501] -12.63875762  -4.17324511 -12.42715541  -8.94462386   2.81432518
## [506] -16.47215644  -4.19690267   1.32880279  -2.85365907 -25.11243430
## [511] -13.01414767 -22.82818786  -9.67447940 -15.20773402 -10.89677707
## [516]   3.12274703   1.00602421 -31.83578101  -8.89415542 -16.50019637
## [521]  -4.58781854 -10.54717818  -6.53425113   2.56547030  -9.71062556
## [526]  38.34921464 -20.82062522  -9.20293990  -2.36076370  -2.62551538
## [531]  -4.76356259 -18.09618751 -25.90262023 -16.77194396  -3.03036777
## [536] -33.63630477   2.44652399  -2.79529945  -5.08324186 -15.40027896
## [541] -11.53592578  -4.21706787 -19.92559425  -2.56557528 -14.48502341
## [546]  35.58666047   4.18550929  19.47993668  -6.95899573  -8.20762626
## [551]  -1.56706384  -0.46068603  -1.67821566 -27.27751342 -18.51480476
## [556]  -1.03106484  -0.25513626  -1.62324217  -5.77376643   1.16532441
## [561]  -0.07948170   8.13243016  -3.02140013  -5.28878013  -5.22633639
## [566]  -3.79084746   7.24946317   0.17408412   5.14595279  -3.16138998
## [571]  -0.48855874   5.36254414  -0.86312047   1.31964787  -0.23209563
## [576]   0.49766126  11.18550500   1.94988307  -5.16826335 -13.50440208
## [581]   6.85215434  -1.90402917  -6.66172639  11.14138707  29.71829423
## [586]  -5.49116985  -1.28135208  -9.58393470  16.38340958   1.84773072
## [591] -33.41437331   8.14972695   6.75352963  -5.30637113  11.67494305
## [596]   1.40130051   2.92557796  25.33718690   1.49714562  -0.23209563
```

```r
X <- t(t(X) -colMeans(X))

cor(atlas2)
```

```
##                         PCH_LACCESS_POP_15_19 PCH_GROC_16_20 PCH_SUPERC_16_20
## PCH_LACCESS_POP_15_19              1.00000000    -0.12825303       0.01054011
## PCH_GROC_16_20                    -0.12825303     1.00000000      -0.03476868
## PCH_SUPERC_16_20                   0.01054011    -0.03476868       1.00000000
## PCH_CONVS_16_20                   -0.04837386    -0.10194873       0.08313836
## PCH_SPECS_16_20                   -0.01327074    -0.03113158      -0.03446943
## PCH_SNAPS_17_23                   -0.07170763     0.14143289      -0.01624929
## PCH_WICS_16_22                    -0.04365784    -0.00269269      -0.03008571
## PCH_FFR_16_20                     -0.17928865     0.14941098      -0.09015193
## PCH_FSR_16_20                      0.04399629     0.08591949      -0.11639025
## PCH_DIRSALES_12_17                -0.04141853     0.05904253       0.06667660
##                         PCH_CONVS_16_20 PCH_SPECS_16_20 PCH_SNAPS_17_23
## PCH_LACCESS_POP_15_19      -0.048373860     -0.01327074    -0.071707633
## PCH_GROC_16_20             -0.101948734     -0.03113158     0.141432885
## PCH_SUPERC_16_20            0.083138361     -0.03446943    -0.016249294
## PCH_CONVS_16_20             1.000000000      0.03017534     0.001181517
## PCH_SPECS_16_20             0.030175344      1.00000000     0.078258999
## PCH_SNAPS_17_23             0.001181517      0.07825900     1.000000000
## PCH_WICS_16_22              0.018046555     -0.03976880    -0.031334023
## PCH_FFR_16_20              -0.012903865      0.06940831     0.095304847
## PCH_FSR_16_20               0.095275731      0.06975211     0.090492833
## PCH_DIRSALES_12_17          0.026334746     -0.01929969    -0.036825560
##                         PCH_WICS_16_22 PCH_FFR_16_20 PCH_FSR_16_20
## PCH_LACCESS_POP_15_19     -0.043657843   -0.17928865   0.043996289
## PCH_GROC_16_20            -0.002692690    0.14941098   0.085919490
## PCH_SUPERC_16_20          -0.030085712   -0.09015193  -0.116390245
## PCH_CONVS_16_20            0.018046555   -0.01290386   0.095275731
```

```
## PCH_SPECS_16_20         -0.039768796    0.06940831    0.069752112
## PCH_SNAPS_17_23         -0.031334023    0.09530485    0.090492833
## PCH_WICS_16_22           1.000000000   -0.01633849    0.005264114
## PCH_FFR_16_20           -0.016338491    1.00000000    0.063304297
## PCH_FSR_16_20            0.005264114    0.06330430    1.000000000
## PCH_DIRSALES_12_17      -0.029665757    0.06431184   -0.024182203
##                       PCH_DIRSALES_12_17
## PCH_LACCESS_POP_15_19         -0.04141853
## PCH_GROC_16_20                 0.05904253
## PCH_SUPERC_16_20               0.06667660
## PCH_CONVS_16_20                0.02633475
## PCH_SPECS_16_20               -0.01929969
## PCH_SNAPS_17_23               -0.03682556
## PCH_WICS_16_22                -0.02966576
## PCH_FFR_16_20                  0.06431184
## PCH_FSR_16_20                 -0.02418220
## PCH_DIRSALES_12_17             1.00000000
```

```r
plot(atlas2,pch=16,cex=.5)
```




```r
fit <- lm(PCH_LACCESS_POP_15_19~.,data=atlas2)
summary(fit)
```

```
##
## Call:
## lm(formula = PCH_LACCESS_POP_15_19 ~ ., data = atlas2)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -57.382  -6.965  -0.849   5.095 134.086 
## 
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)    
## (Intercept)          2.7315104  1.0976782   2.488  0.01311 *  
## PCH_GROC_16_20      -0.1089864  0.0412257  -2.644  0.00842 ** 
## PCH_SUPERC_16_20     0.0007048  0.0050674   0.139  0.88944    
## PCH_CONVS_16_20     -0.0718614  0.0433000  -1.660  0.09752 .  
## PCH_SPECS_16_20     -0.0032387  0.0187694  -0.173  0.86306    
## PCH_SNAPS_17_23     -0.0692429  0.0572575  -1.209  0.22702    
## PCH_WICS_16_22      -0.0089636  0.0074607  -1.201  0.23006    
## PCH_FFR_16_20       -0.2667556  0.0673227  -3.962 8.33e-05 ***
## PCH_FSR_16_20        0.1400873  0.0760650   1.842  0.06602 .  
## PCH_DIRSALES_12_17 -0.0016520  0.0026964  -0.613  0.54033    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 14.65 on 590 degrees of freedom
## Multiple R-squared:  0.05645,    Adjusted R-squared:  0.04206 
## F-statistic: 3.922 on 9 and 590 DF,  p-value: 7.339e-05
```

```r
fit0 <- lm(PCH_LACCESS_POP_15_19~1,data=atlas2)
summary(fit0)
```

```
## 
## Call:
## lm(formula = PCH_LACCESS_POP_15_19 ~ 1, data = atlas2)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -63.369  -6.525  -0.707   5.026 140.582 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.2321     0.6111    0.38    0.704
## 
## Residual standard error: 14.97 on 599 degrees of freedom
```

```r
fit_bsub <- regsubsets(x=atlas2[,2:10],y=atlas2[,1])
summary(fit_bsub)
```

```
## Subset selection object
## 9 Variables  (and intercept)
##                    Forced in Forced out
## PCH_GROC_16_20         FALSE      FALSE
## PCH_SUPERC_16_20       FALSE      FALSE
## PCH_CONVS_16_20        FALSE      FALSE
## PCH_SPECS_16_20        FALSE      FALSE
## PCH_SNAPS_17_23        FALSE      FALSE
## PCH_WICS_16_22         FALSE      FALSE
## PCH_FFR_16_20          FALSE      FALSE
## PCH_FSR_16_20          FALSE      FALSE
## PCH_DIRSALES_12_17     FALSE      FALSE
```

```
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          PCH_GROC_16_20 PCH_SUPERC_16_20 PCH_CONVS_16_20 PCH_SPECS_16_20
## 1  ( 1 ) " "            " "              " "             " "
## 2  ( 1 ) "*"            " "              " "             " "
## 3  ( 1 ) "*"            " "              " "             " "
## 4  ( 1 ) "*"            " "              "*"             " "
## 5  ( 1 ) "*"            " "              "*"             " "
## 6  ( 1 ) "*"            " "              "*"             " "
## 7  ( 1 ) "*"            " "              "*"             " "
## 8  ( 1 ) "*"            " "              "*"             "*"
##          PCH_SNAPS_17_23 PCH_WICS_16_22 PCH_FFR_16_20 PCH_FSR_16_20
## 1  ( 1 ) " "             " "            "*"           " "
## 2  ( 1 ) " "             " "            "*"           " "
## 3  ( 1 ) " "             " "            "*"           "*"
## 4  ( 1 ) " "             " "            "*"           "*"
## 5  ( 1 ) "*"             " "            "*"           "*"
## 6  ( 1 ) "*"             "*"            "*"           "*"
## 7  ( 1 ) "*"             "*"            "*"           "*"
## 8  ( 1 ) "*"             "*"            "*"           "*"
##          PCH_DIRSALES_12_17
## 1  ( 1 ) " "
## 2  ( 1 ) " "
## 3  ( 1 ) " "
## 4  ( 1 ) " "
## 5  ( 1 ) " "
## 6  ( 1 ) " "
## 7  ( 1 ) "*"
## 8  ( 1 ) "*"
```

```
fit_bsub$rss
```

```
##  [1] 134203.8 133948.0 133917.7 133907.5 133615.0 129175.8 129055.9 128653.4
##  [9] 128127.6 126627.6
```

Sequential Selection Methods

Both forward and backward step-wise selection via BIC chose fast food restaurants (coefficient of -0.26948 with a p-value of 5.94e-05) and grocery stores (coefficient of -0.10368 with a p-value of 0.01063).

```
# forward step-wise via BIC
fit_forw <- stepAIC(fit0,scope=PCH_LACCESS_POP_15_19~PCH_GROC_16_20+PCH_SUPERC_16_20+PCH_CONVS_16_20+PCH
                    direction="forward",data=atlas2,k=log(nrow(atlas2)))
```

```
## Start:  AIC=3252.51
## PCH_LACCESS_POP_15_19 ~ 1
##
##                      Df Sum of Sq    RSS    AIC
## + PCH_FFR_16_20       1    4313.9 129890 3239.3
## + PCH_GROC_16_20      1    2207.5 131996 3249.0
## <none>                            134204 3252.5
## + PCH_SNAPS_17_23     1     690.1 133514 3255.8
## + PCH_CONVS_16_20     1     314.0 133890 3257.5
## + PCH_FSR_16_20       1     259.8 133944 3257.7
## + PCH_WICS_16_22      1     255.8 133948 3257.8
## + PCH_DIRSALES_12_17  1     230.2 133974 3257.9
```

```
## + PCH_SPECS_16_20      1      23.6 134180 3258.8
## + PCH_SUPERC_16_20     1      14.9 134189 3258.8
##
## Step:  AIC=3239.3
## PCH_LACCESS_POP_15_19 ~ PCH_FFR_16_20
##
##                       Df Sum of Sq    RSS    AIC
## + PCH_GROC_16_20       1   1413.20 128477 3239.1
## <none>                             129890 3239.3
## + PCH_FSR_16_20        1    412.74 129477 3243.8
## + PCH_SNAPS_17_23      1    404.05 129486 3243.8
## + PCH_CONVS_16_20      1    344.86 129545 3244.1
## + PCH_WICS_16_22       1    291.35 129599 3244.4
## + PCH_DIRSALES_12_17   1    120.38 129769 3245.1
## + PCH_SUPERC_16_20     1      4.28 129886 3245.7
## + PCH_SPECS_16_20      1      0.09 129890 3245.7
##
## Step:  AIC=3239.13
## PCH_LACCESS_POP_15_19 ~ PCH_FFR_16_20 + PCH_GROC_16_20
##
##                       Df Sum of Sq    RSS    AIC
## <none>                             128477 3239.1
## + PCH_FSR_16_20        1    542.84 127934 3243.0
## + PCH_CONVS_16_20      1    505.74 127971 3243.2
## + PCH_WICS_16_22       1    291.68 128185 3244.2
## + PCH_SNAPS_17_23      1    236.30 128240 3244.4
## + PCH_DIRSALES_12_17   1     82.81 128394 3245.1
## + PCH_SUPERC_16_20     1      8.31 128468 3245.5
## + PCH_SPECS_16_20      1      3.56 128473 3245.5
```

```r
summary(fit_forw)
```

```
##
## Call:
## lm(formula = PCH_LACCESS_POP_15_19 ~ PCH_FFR_16_20 + PCH_GROC_16_20,
##     data = atlas2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -58.001  -6.651  -1.035   4.928 137.077
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.94549    0.74732   2.603  0.00946 **
## PCH_FFR_16_20  -0.26948    0.06663  -4.044 5.94e-05 ***
## PCH_GROC_16_20 -0.10368    0.04046  -2.563  0.01063 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.67 on 597 degrees of freedom
## Multiple R-squared:  0.04267,    Adjusted R-squared:  0.03947
## F-statistic: 13.31 on 2 and 597 DF,  p-value: 2.22e-06
```

```r
# backward step-wise via BIC
fit_back <- stepAIC(fit,direction="backward",data=atlas2,k=log(nrow(atlas2)))
```

```
## Start:  AIC=3275.22
## PCH_LACCESS_POP_15_19 ~ PCH_GROC_16_20 + PCH_SUPERC_16_20 + PCH_CONVS_16_20 +
##     PCH_SPECS_16_20 + PCH_SNAPS_17_23 + PCH_WICS_16_22 + PCH_FFR_16_20 +
##     PCH_FSR_16_20 + PCH_DIRSALES_12_17
##
##                      Df Sum of Sq    RSS    AIC
## - PCH_SUPERC_16_20    1       4.2 126632 3268.8
## - PCH_SPECS_16_20     1       6.4 126634 3268.8
## - PCH_DIRSALES_12_17  1      80.6 126708 3269.2
## - PCH_WICS_16_22      1     309.8 126937 3270.3
## - PCH_SNAPS_17_23     1     313.9 126942 3270.3
## - PCH_CONVS_16_20     1     591.1 127219 3271.6
## - PCH_FSR_16_20       1     728.0 127356 3272.3
## <none>                            126628 3275.2
## - PCH_GROC_16_20      1    1500.0 128128 3275.9
## - PCH_FFR_16_20       1    3369.6 129997 3284.6
##
## Step:  AIC=3268.84
## PCH_LACCESS_POP_15_19 ~ PCH_GROC_16_20 + PCH_CONVS_16_20 + PCH_SPECS_16_20 +
##     PCH_SNAPS_17_23 + PCH_WICS_16_22 + PCH_FFR_16_20 + PCH_FSR_16_20 +
##     PCH_DIRSALES_12_17
##
##                      Df Sum of Sq    RSS    AIC
## - PCH_SPECS_16_20     1       6.6 126638 3262.5
## - PCH_DIRSALES_12_17  1      78.5 126710 3262.8
## - PCH_WICS_16_22      1     312.4 126944 3263.9
## - PCH_SNAPS_17_23     1     313.4 126945 3263.9
## - PCH_CONVS_16_20     1     587.0 127219 3265.2
## - PCH_FSR_16_20       1     725.0 127357 3265.9
## <none>                            126632 3268.8
## - PCH_GROC_16_20      1    1501.4 128133 3269.5
## - PCH_FFR_16_20       1    3413.3 130045 3278.4
##
## Step:  AIC=3262.47
## PCH_LACCESS_POP_15_19 ~ PCH_GROC_16_20 + PCH_CONVS_16_20 + PCH_SNAPS_17_23 +
##     PCH_WICS_16_22 + PCH_FFR_16_20 + PCH_FSR_16_20 + PCH_DIRSALES_12_17
##
##                      Df Sum of Sq    RSS    AIC
## - PCH_DIRSALES_12_17  1      77.7 126716 3256.4
## - PCH_WICS_16_22      1     309.4 126948 3257.5
## - PCH_SNAPS_17_23     1     321.6 126960 3257.6
## - PCH_CONVS_16_20     1     589.9 127228 3258.9
## - PCH_FSR_16_20       1     719.2 127358 3259.5
## <none>                            126638 3262.5
## - PCH_GROC_16_20      1    1495.1 128133 3263.1
## - PCH_FFR_16_20       1    3448.9 130087 3272.2
##
## Step:  AIC=3256.44
## PCH_LACCESS_POP_15_19 ~ PCH_GROC_16_20 + PCH_CONVS_16_20 + PCH_SNAPS_17_23 +
##     PCH_WICS_16_22 + PCH_FFR_16_20 + PCH_FSR_16_20
##
##                   Df Sum of Sq    RSS    AIC
## - PCH_WICS_16_22   1     300.2 127016 3251.5
## - PCH_SNAPS_17_23  1     307.0 127023 3251.5
```

```
## - PCH_CONVS_16_20  1     606.7 127323 3252.9
## - PCH_FSR_16_20    1     735.3 127451 3253.5
## <none>                         126716 3256.4
## - PCH_GROC_16_20   1    1543.5 128260 3257.3
## - PCH_FFR_16_20    1    3525.3 130241 3266.5
##
## Step:  AIC=3251.47
## PCH_LACCESS_POP_15_19 ~ PCH_GROC_16_20 + PCH_CONVS_16_20 + PCH_SNAPS_17_23 +
##     PCH_FFR_16_20 + PCH_FSR_16_20
##
##                   Df Sum of Sq    RSS    AIC
## - PCH_SNAPS_17_23  1     288.8 127305 3246.4
## - PCH_CONVS_16_20  1     622.0 127638 3248.0
## - PCH_FSR_16_20    1     728.9 127745 3248.5
## <none>                         127016 3251.5
## - PCH_GROC_16_20   1    1550.3 128567 3252.3
## - PCH_FFR_16_20    1    3497.1 130513 3261.4
##
## Step:  AIC=3246.43
## PCH_LACCESS_POP_15_19 ~ PCH_GROC_16_20 + PCH_CONVS_16_20 + PCH_FFR_16_20 +
##     PCH_FSR_16_20
##
##                   Df Sum of Sq    RSS    AIC
## - PCH_CONVS_16_20  1     628.7 127934 3243.0
## - PCH_FSR_16_20    1     665.8 127971 3243.2
## <none>                         127305 3246.4
## - PCH_GROC_16_20   1    1746.7 129052 3248.2
## - PCH_FFR_16_20    1    3662.4 130968 3257.1
##
## Step:  AIC=3242.99
## PCH_LACCESS_POP_15_19 ~ PCH_GROC_16_20 + PCH_FFR_16_20 + PCH_FSR_16_20
##
##                  Df Sum of Sq    RSS    AIC
## - PCH_FSR_16_20   1     542.8 128477 3239.1
## <none>                        127934 3243.0
## - PCH_GROC_16_20  1    1543.3 129477 3243.8
## - PCH_FFR_16_20   1    3653.2 131587 3253.5
##
## Step:  AIC=3239.13
## PCH_LACCESS_POP_15_19 ~ PCH_GROC_16_20 + PCH_FFR_16_20
##
##                  Df Sum of Sq    RSS    AIC
## <none>                        128477 3239.1
## - PCH_GROC_16_20  1    1413.2 129890 3239.3
## - PCH_FFR_16_20   1    3519.6 131996 3249.0
```

```r
summary(fit_back)
```

```
##
## Call:
## lm(formula = PCH_LACCESS_POP_15_19 ~ PCH_GROC_16_20 + PCH_FFR_16_20,
##     data = atlas2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -58.001  -6.651  -1.035   4.928 137.077
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.94549    0.74732   2.603  0.00946 **
## PCH_GROC_16_20 -0.10368    0.04046  -2.563  0.01063 *
## PCH_FFR_16_20  -0.26948    0.06663  -4.044 5.94e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.67 on 597 degrees of freedom
## Multiple R-squared:  0.04267,    Adjusted R-squared:  0.03947
## F-statistic: 13.31 on 2 and 597 DF,  p-value: 2.22e-06
```

Ridge:

The coefficients for least squares are very similar to the coefficients for ridge. The orders of the coefficients does not vary with lambda. We see the strongest effects from fast food restaurants and full service restaurants, followed by grocery stores, then convenience stores and SNAP-accepting stores.

```r
lam <- 1*nrow(atlas2)

beta_ls <- solve(t(X)%*%X)%*%t(X)%*%Y
beta_r <- solve(t(X)%*%X + diag(rep(lam,9)))%*%t(X)%*%Y
cbind(beta_ls,beta_r)
```

```
##                              [,1]          [,2]
## PCH_GROC_16_20     -0.1089863758 -0.1086595767
## PCH_SUPERC_16_20    0.0007047547  0.0007047661
## PCH_CONVS_16_20    -0.0718614173 -0.0712912167
## PCH_SPECS_16_20    -0.0032387185 -0.0032698322
## PCH_SNAPS_17_23    -0.0692429375 -0.0687920794
## PCH_WICS_16_22     -0.0089636422 -0.0089561724
## PCH_FFR_16_20      -0.2667555927 -0.2635144639
## PCH_FSR_16_20       0.1400872946  0.1375277231
## PCH_DIRSALES_12_17 -0.0016519904 -0.0016641008
```

```r
# coefficient paths

lambdas <- exp(seq(log(.01),log(100*nrow(atlas2)),l=100))
betasr <- matrix(0,length(lambdas),9)
for(i in 1:length(lambdas))
{
  betasr[i,] = solve(t(X)%*%X + diag(rep(lambdas[i],9)))%*%t(X)%*%Y
}

betasr
```

```
##               [,1]         [,2]        [,3]         [,4]        [,5]
##   [1,] -0.10898637 0.0007047547 -0.07186141 -0.003238719 -0.06924293
##   [2,] -0.10898637 0.0007047547 -0.07186141 -0.003238719 -0.06924293
##   [3,] -0.10898637 0.0007047547 -0.07186140 -0.003238719 -0.06924293
##   [4,] -0.10898637 0.0007047547 -0.07186140 -0.003238719 -0.06924293
##   [5,] -0.10898637 0.0007047547 -0.07186140 -0.003238720 -0.06924292
##   [6,] -0.10898636 0.0007047547 -0.07186140 -0.003238720 -0.06924292
##   [7,] -0.10898636 0.0007047547 -0.07186139 -0.003238720 -0.06924292
##   [8,] -0.10898636 0.0007047547 -0.07186139 -0.003238720 -0.06924291
```

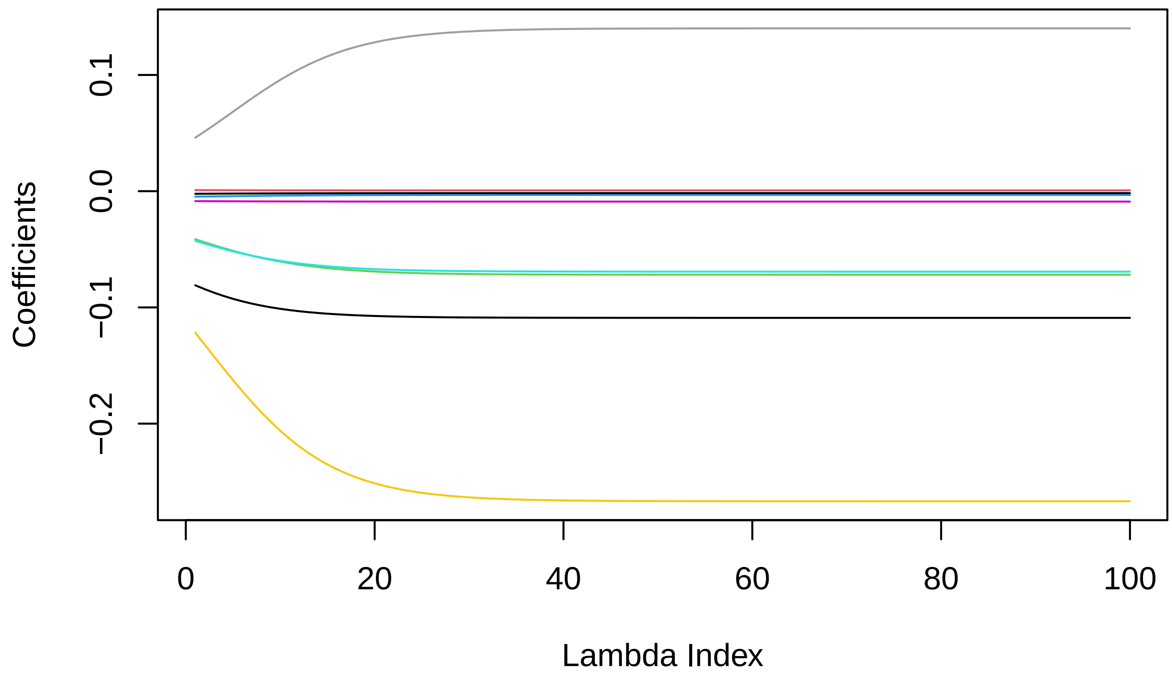
```
##    [9,] -0.10898636 0.0007047547 -0.07186138 -0.003238720 -0.06924291
##   [10,] -0.10898635 0.0007047547 -0.07186138 -0.003238721 -0.06924291
##   [11,] -0.10898635 0.0007047547 -0.07186137 -0.003238721 -0.06924290
##   [12,] -0.10898634 0.0007047547 -0.07186136 -0.003238722 -0.06924289
##   [13,] -0.10898634 0.0007047547 -0.07186135 -0.003238722 -0.06924289
##   [14,] -0.10898633 0.0007047547 -0.07186134 -0.003238723 -0.06924288
##   [15,] -0.10898633 0.0007047547 -0.07186133 -0.003238723 -0.06924287
##   [16,] -0.10898632 0.0007047547 -0.07186132 -0.003238724 -0.06924286
##   [17,] -0.10898631 0.0007047547 -0.07186130 -0.003238725 -0.06924284
##   [18,] -0.10898630 0.0007047547 -0.07186128 -0.003238726 -0.06924283
##   [19,] -0.10898628 0.0007047546 -0.07186125 -0.003238727 -0.06924281
##   [20,] -0.10898627 0.0007047546 -0.07186123 -0.003238729 -0.06924279
##   [21,] -0.10898625 0.0007047546 -0.07186119 -0.003238731 -0.06924276
##   [22,] -0.10898623 0.0007047546 -0.07186115 -0.003238733 -0.06924273
##   [23,] -0.10898620 0.0007047546 -0.07186111 -0.003238735 -0.06924269
##   [24,] -0.10898617 0.0007047546 -0.07186106 -0.003238738 -0.06924265
##   [25,] -0.10898614 0.0007047546 -0.07186100 -0.003238742 -0.06924260
##   [26,] -0.10898610 0.0007047546 -0.07186092 -0.003238745 -0.06924255
##   [27,] -0.10898605 0.0007047546 -0.07186084 -0.003238750 -0.06924248
##   [28,] -0.10898599 0.0007047546 -0.07186074 -0.003238755 -0.06924240
##   [29,] -0.10898593 0.0007047546 -0.07186062 -0.003238762 -0.06924231
##   [30,] -0.10898585 0.0007047545 -0.07186049 -0.003238769 -0.06924221
##   [31,] -0.10898576 0.0007047545 -0.07186033 -0.003238778 -0.06924208
##   [32,] -0.10898565 0.0007047545 -0.07186014 -0.003238788 -0.06924193
##   [33,] -0.10898553 0.0007047545 -0.07185993 -0.003238800 -0.06924176
##   [34,] -0.10898539 0.0007047544 -0.07185967 -0.003238814 -0.06924156
##   [35,] -0.10898522 0.0007047544 -0.07185938 -0.003238830 -0.06924133
##   [36,] -0.10898502 0.0007047543 -0.07185903 -0.003238849 -0.06924105
##   [37,] -0.10898479 0.0007047543 -0.07185862 -0.003238871 -0.06924073
##   [38,] -0.10898452 0.0007047542 -0.07185814 -0.003238897 -0.06924035
##   [39,] -0.10898420 0.0007047541 -0.07185758 -0.003238928 -0.06923991
##   [40,] -0.10898383 0.0007047540 -0.07185693 -0.003238963 -0.06923939
##   [41,] -0.10898339 0.0007047539 -0.07185616 -0.003239005 -0.06923879
##   [42,] -0.10898288 0.0007047538 -0.07185526 -0.003239054 -0.06923808
##   [43,] -0.10898229 0.0007047537 -0.07185421 -0.003239111 -0.06923725
##   [44,] -0.10898159 0.0007047535 -0.07185298 -0.003239178 -0.06923628
##   [45,] -0.10898077 0.0007047533 -0.07185154 -0.003239257 -0.06923514
##   [46,] -0.10897981 0.0007047531 -0.07184985 -0.003239349 -0.06923381
##   [47,] -0.10897869 0.0007047528 -0.07184788 -0.003239456 -0.06923226
##   [48,] -0.10897738 0.0007047525 -0.07184557 -0.003239582 -0.06923043
##   [49,] -0.10897584 0.0007047521 -0.07184286 -0.003239730 -0.06922830
##   [50,] -0.10897405 0.0007047517 -0.07183969 -0.003239902 -0.06922580
##   [51,] -0.10897194 0.0007047512 -0.07183599 -0.003240104 -0.06922287
##   [52,] -0.10896947 0.0007047507 -0.07183165 -0.003240341 -0.06921945
##   [53,] -0.10896659 0.0007047500 -0.07182656 -0.003240618 -0.06921544
##   [54,] -0.10896321 0.0007047493 -0.07182062 -0.003240942 -0.06921075
##   [55,] -0.10895925 0.0007047485 -0.07181366 -0.003241321 -0.06920525
##   [56,] -0.10895462 0.0007047476 -0.07180551 -0.003241765 -0.06919882
##   [57,] -0.10894920 0.0007047466 -0.07179597 -0.003242285 -0.06919130
##   [58,] -0.10894285 0.0007047454 -0.07178481 -0.003242893 -0.06918249
##   [59,] -0.10893542 0.0007047442 -0.07177175 -0.003243605 -0.06917218
##   [60,] -0.10892672 0.0007047428 -0.07175647 -0.003244438 -0.06916011
##   [61,] -0.10891654 0.0007047415 -0.07173859 -0.003245413 -0.06914599
##   [62,] -0.10890461 0.0007047401 -0.07171767 -0.003246554 -0.06912947
```

```
##  [63,] -0.10889065 0.0007047388 -0.07169320 -0.003247888 -0.06911014
##  [64,] -0.10887430 0.0007047377 -0.07166457 -0.003249449 -0.06908753
##  [65,] -0.10885516 0.0007047370 -0.07163110 -0.003251275 -0.06906107
##  [66,] -0.10883276 0.0007047370 -0.07159196 -0.003253411 -0.06903013
##  [67,] -0.10880653 0.0007047381 -0.07154620 -0.003255907 -0.06899395
##  [68,] -0.10877582 0.0007047409 -0.07149273 -0.003258826 -0.06895165
##  [69,] -0.10873987 0.0007047463 -0.07143025 -0.003262237 -0.06890220
##  [70,] -0.10869777 0.0007047553 -0.07135728 -0.003266223 -0.06884442
##  [71,] -0.10864850 0.0007047697 -0.07127208 -0.003270878 -0.06877692
##  [72,] -0.10859081 0.0007047916 -0.07117267 -0.003276312 -0.06869809
##  [73,] -0.10852327 0.0007048242 -0.07105672 -0.003282653 -0.06860607
##  [74,] -0.10844420 0.0007048717 -0.07092158 -0.003290047 -0.06849871
##  [75,] -0.10835164 0.0007049398 -0.07076417 -0.003298665 -0.06837353
##  [76,] -0.10824328 0.0007050363 -0.07058100 -0.003308700 -0.06822765
##  [77,] -0.10811644 0.0007051716 -0.07036805 -0.003320375 -0.06805780
##  [78,] -0.10796797 0.0007053598 -0.07012077 -0.003333943 -0.06786021
##  [79,] -0.10779419 0.0007056194 -0.06983400 -0.003349690 -0.06763058
##  [80,] -0.10759082 0.0007059749 -0.06950192 -0.003367938 -0.06736405
##  [81,] -0.10735284 0.0007064584 -0.06911807 -0.003389048 -0.06705509
##  [82,] -0.10707442 0.0007071114 -0.06867524 -0.003413415 -0.06669754
##  [83,] -0.10674878 0.0007079870 -0.06816553 -0.003441473 -0.06628450
##  [84,] -0.10636802 0.0007091529 -0.06758035 -0.003473688 -0.06580834
##  [85,] -0.10592304 0.0007106934 -0.06691047 -0.003510547 -0.06526071
##  [86,] -0.10540330 0.0007127122 -0.06614616 -0.003552552 -0.06463260
##  [87,] -0.10479674 0.0007153349 -0.06527731 -0.003600194 -0.06391435
##  [88,] -0.10408959 0.0007187100 -0.06429365 -0.003653932 -0.06309586
##  [89,] -0.10326632 0.0007230091 -0.06318507 -0.003714153 -0.06216675
##  [90,] -0.10230953 0.0007284251 -0.06194195 -0.003781135 -0.06111665
##  [91,] -0.10120007 0.0007351676 -0.06055560 -0.003854983 -0.05993556
##  [92,] -0.09991718 0.0007434555 -0.05901876 -0.003935579 -0.05861431
##  [93,] -0.09843895 0.0007535065 -0.05732613 -0.004022504 -0.05714510
##  [94,] -0.09674290 0.0007655237 -0.05547498 -0.004114985 -0.05552215
##  [95,] -0.09480697 0.0007796803 -0.05346570 -0.004211825 -0.05374234
##  [96,] -0.09261070 0.0007961034 -0.05130236 -0.004311371 -0.05180594
##  [97,] -0.09013681 0.0008148593 -0.04899320 -0.004411486 -0.04971720
##  [98,] -0.08737295 0.0008359417 -0.04655096 -0.004509571 -0.04748486
##  [99,] -0.08431353 0.0008592638 -0.04399300 -0.004602611 -0.04512243
## [100,] -0.08096150 0.0008846558 -0.04134123 -0.004687262 -0.04264821
##                [,6]       [,7]       [,8]         [,9]
##   [1,] -0.008963642 -0.2667555 0.14008725 -0.001651991
##   [2,] -0.008963642 -0.2667555 0.14008724 -0.001651991
##   [3,] -0.008963642 -0.2667555 0.14008724 -0.001651991
##   [4,] -0.008963642 -0.2667555 0.14008722 -0.001651991
##   [5,] -0.008963642 -0.2667555 0.14008721 -0.001651991
##   [6,] -0.008963642 -0.2667555 0.14008720 -0.001651991
##   [7,] -0.008963642 -0.2667555 0.14008718 -0.001651991
##   [8,] -0.008963642 -0.2667554 0.14008716 -0.001651991
##   [9,] -0.008963642 -0.2667554 0.14008714 -0.001651991
##  [10,] -0.008963642 -0.2667554 0.14008712 -0.001651991
##  [11,] -0.008963642 -0.2667553 0.14008708 -0.001651991
##  [12,] -0.008963641 -0.2667553 0.14008705 -0.001651992
##  [13,] -0.008963641 -0.2667552 0.14008701 -0.001651992
##  [14,] -0.008963641 -0.2667552 0.14008696 -0.001651992
##  [15,] -0.008963641 -0.2667551 0.14008690 -0.001651992
```

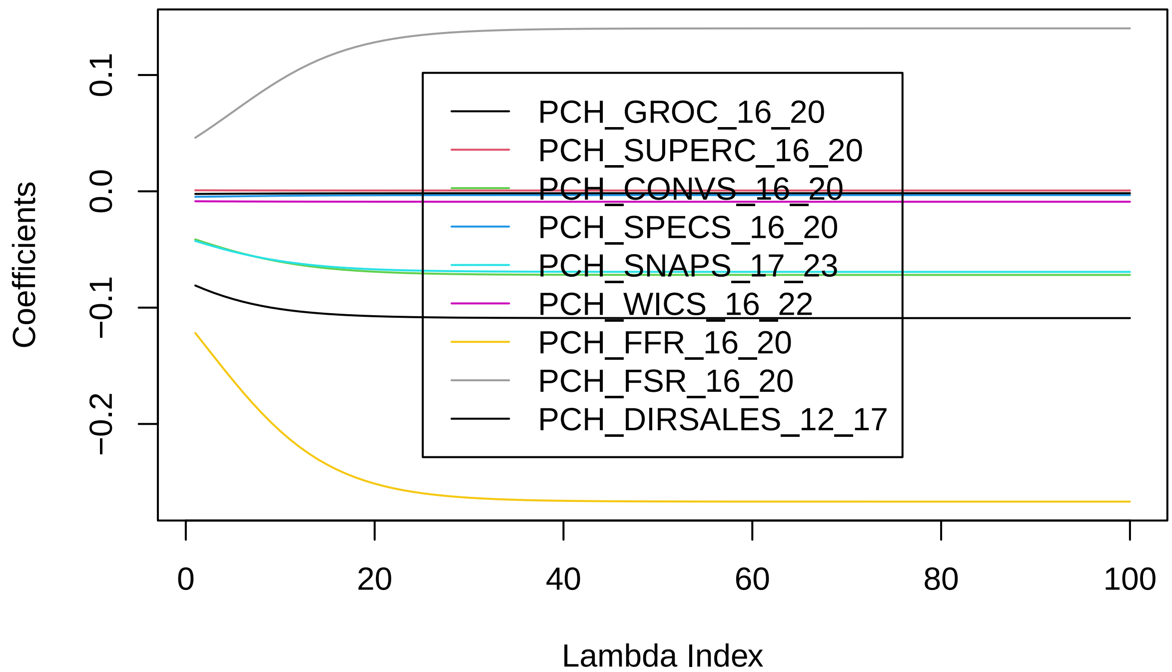
```
##  [16,] -0.008963641 -0.2667550 0.14008683 -0.001651993
##  [17,] -0.008963641 -0.2667549 0.14008675 -0.001651993
##  [18,] -0.008963640 -0.2667548 0.14008666 -0.001651993
##  [19,] -0.008963640 -0.2667547 0.14008655 -0.001651994
##  [20,] -0.008963640 -0.2667545 0.14008643 -0.001651994
##  [21,] -0.008963639 -0.2667543 0.14008628 -0.001651995
##  [22,] -0.008963639 -0.2667541 0.14008611 -0.001651996
##  [23,] -0.008963638 -0.2667538 0.14008590 -0.001651997
##  [24,] -0.008963637 -0.2667535 0.14008566 -0.001651998
##  [25,] -0.008963637 -0.2667532 0.14008539 -0.001651999
##  [26,] -0.008963636 -0.2667528 0.14008506 -0.001652001
##  [27,] -0.008963635 -0.2667523 0.14008468 -0.001652003
##  [28,] -0.008963633 -0.2667517 0.14008423 -0.001652005
##  [29,] -0.008963632 -0.2667511 0.14008371 -0.001652007
##  [30,] -0.008963630 -0.2667503 0.14008310 -0.001652010
##  [31,] -0.008963628 -0.2667494 0.14008238 -0.001652013
##  [32,] -0.008963626 -0.2667483 0.14008154 -0.001652017
##  [33,] -0.008963623 -0.2667471 0.14008056 -0.001652022
##  [34,] -0.008963619 -0.2667457 0.14007941 -0.001652027
##  [35,] -0.008963615 -0.2667440 0.14007806 -0.001652034
##  [36,] -0.008963611 -0.2667420 0.14007649 -0.001652041
##  [37,] -0.008963605 -0.2667396 0.14007464 -0.001652050
##  [38,] -0.008963599 -0.2667369 0.14007248 -0.001652060
##  [39,] -0.008963592 -0.2667337 0.14006995 -0.001652072
##  [40,] -0.008963583 -0.2667300 0.14006699 -0.001652086
##  [41,] -0.008963573 -0.2667256 0.14006353 -0.001652102
##  [42,] -0.008963562 -0.2667205 0.14005947 -0.001652121
##  [43,] -0.008963548 -0.2667145 0.14005472 -0.001652144
##  [44,] -0.008963532 -0.2667075 0.14004915 -0.001652170
##  [45,] -0.008963513 -0.2666993 0.14004264 -0.001652201
##  [46,] -0.008963491 -0.2666897 0.14003502 -0.001652236
##  [47,] -0.008963465 -0.2666784 0.14002610 -0.001652278
##  [48,] -0.008963434 -0.2666653 0.14001566 -0.001652328
##  [49,] -0.008963399 -0.2666498 0.14000343 -0.001652385
##  [50,] -0.008963358 -0.2666318 0.13998912 -0.001652453
##  [51,] -0.008963309 -0.2666107 0.13997237 -0.001652532
##  [52,] -0.008963252 -0.2665859 0.13995276 -0.001652624
##  [53,] -0.008963186 -0.2665570 0.13992981 -0.001652732
##  [54,] -0.008963108 -0.2665231 0.13990296 -0.001652858
##  [55,] -0.008963016 -0.2664835 0.13987152 -0.001653007
##  [56,] -0.008962910 -0.2664370 0.13983474 -0.001653180
##  [57,] -0.008962785 -0.2663827 0.13979170 -0.001653383
##  [58,] -0.008962638 -0.2663192 0.13974134 -0.001653620
##  [59,] -0.008962467 -0.2662448 0.13968243 -0.001653898
##  [60,] -0.008962267 -0.2661578 0.13961351 -0.001654223
##  [61,] -0.008962033 -0.2660560 0.13953291 -0.001654603
##  [62,] -0.008961759 -0.2659369 0.13943865 -0.001655048
##  [63,] -0.008961438 -0.2657976 0.13932844 -0.001655568
##  [64,] -0.008961063 -0.2656347 0.13919963 -0.001656177
##  [65,] -0.008960625 -0.2654443 0.13904910 -0.001656888
##  [66,] -0.008960112 -0.2652218 0.13887324 -0.001657719
##  [67,] -0.008959512 -0.2649617 0.13866789 -0.001658691
##  [68,] -0.008958812 -0.2646579 0.13842818 -0.001659826
##  [69,] -0.008957993 -0.2643031 0.13814851 -0.001661152
```

```
##  [70,] -0.008957038 -0.2638891 0.13782242 -0.001662700
##  [71,] -0.008955922 -0.2634060 0.13744246 -0.001664506
##  [72,] -0.008954620 -0.2628428 0.13700008 -0.001666613
##  [73,] -0.008953101 -0.2621866 0.13648550 -0.001669067
##  [74,] -0.008951331 -0.2614227 0.13588760 -0.001671926
##  [75,] -0.008949270 -0.2605342 0.13519374 -0.001675251
##  [76,] -0.008946872 -0.2595020 0.13438971 -0.001679117
##  [77,] -0.008944084 -0.2583043 0.13345959 -0.001683604
##  [78,] -0.008940847 -0.2569167 0.13238571 -0.001688807
##  [79,] -0.008937093 -0.2553117 0.13114865 -0.001694828
##  [80,] -0.008932748 -0.2534591 0.12972734 -0.001701785
##  [81,] -0.008927725 -0.2513253 0.12809923 -0.001709805
##  [82,] -0.008921932 -0.2488744 0.12624065 -0.001719028
##  [83,] -0.008915265 -0.2460674 0.12412731 -0.001729606
##  [84,] -0.008907612 -0.2428638 0.12173506 -0.001741700
##  [85,] -0.008898853 -0.2392218 0.11904085 -0.001755477
##  [86,] -0.008888859 -0.2350997 0.11602397 -0.001771108
##  [87,] -0.008877498 -0.2304576 0.11266752 -0.001788762
##  [88,] -0.008864632 -0.2252594 0.10896014 -0.001808599
##  [89,] -0.008850124 -0.2194748 0.10489779 -0.001830766
##  [90,] -0.008833838 -0.2130826 0.10048555 -0.001855382
##  [91,] -0.008815645 -0.2060730 0.09573923 -0.001882535
##  [92,] -0.008795422 -0.1984507 0.09068652 -0.001912268
##  [93,] -0.008773056 -0.1902371 0.08536755 -0.001944570
##  [94,] -0.008748444 -0.1814722 0.07983448 -0.001979373
##  [95,] -0.008721491 -0.1722153 0.07415025 -0.002016541
##  [96,] -0.008692107 -0.1625446 0.06838615 -0.002055873
##  [97,] -0.008660197 -0.1525549 0.06261864 -0.002097100
##  [98,] -0.008625652 -0.1423548 0.05692553 -0.002139892
##  [99,] -0.008588340 -0.1320616 0.05138187 -0.002183865
## [100,] -0.008548087 -0.1217963 0.04605610 -0.002228592
```

```
plot(c(1,length(lambdas)),range(betasr),type="n",ylab="Coefficients",xlab="Lambda Index")
for(j in 1:9)
{
  lines(betasr[length(lambdas):1,j],col=j)
}
legend(0,20,legend=names(atlas2)[2:10],col=1:9,lty=rep(1,9))
```




```
plot(c(1,length(lambdas)),range(betasr),type="n",ylab="Coefficients",xlab="Lambda Index")
for(j in 1:9)
{
  lines(betasr[length(lambdas):1,j],col=j)
}
legend("center",legend=names(atlas2)[2:10],col=1:9,lty=rep(1,9))
```



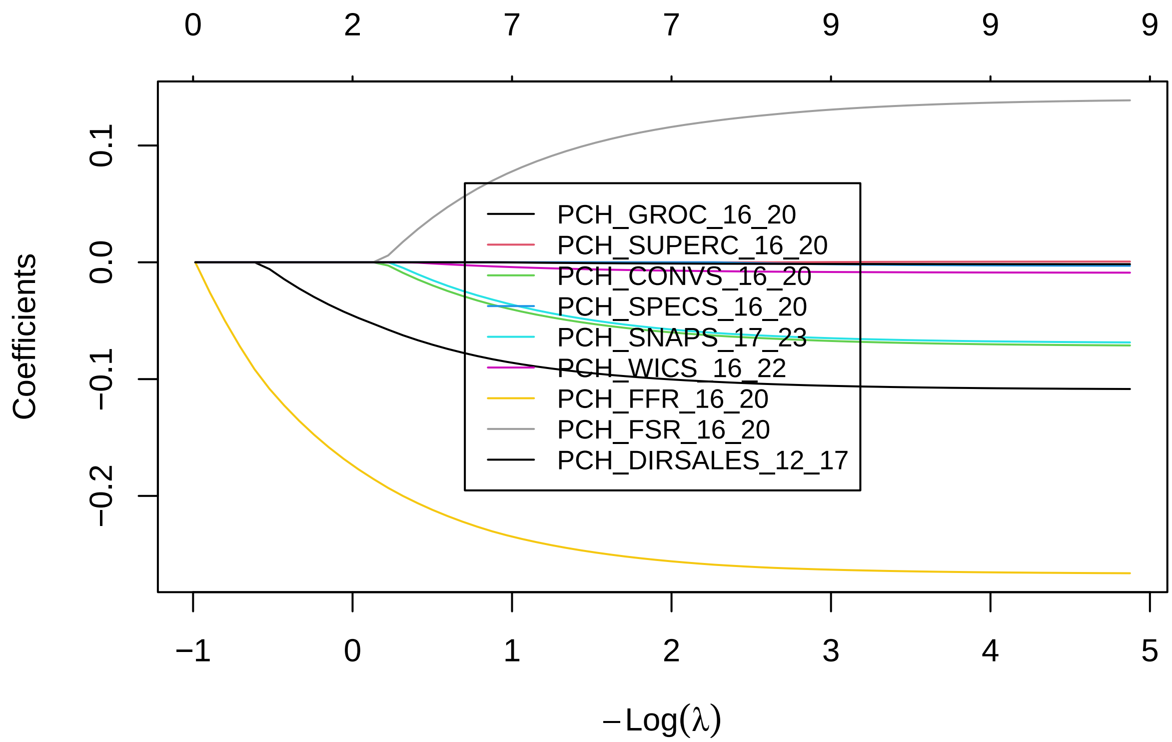
Lasso:

Here we select grocery stores and fast food restaurants with minimum lambdas of -.02964961 and -.1475496 respectively. For cross-validation, the minimum lambda was 0.6052 with a least squares estimate of 2.6814. The mean squared error did not significantly vary across lambdas, and the cross-validation suggested to include four more variables, which indicates that lasso is not the best choice for this dataset.

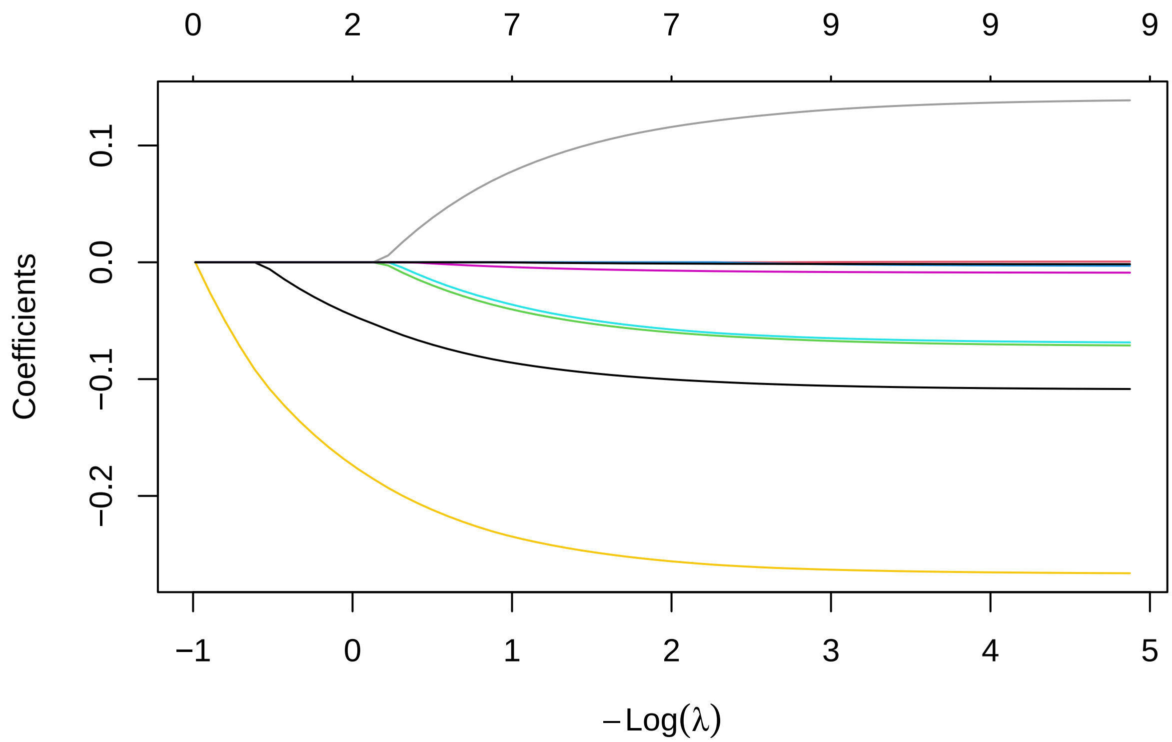
```
install.packages("glmnet",repos = "http://cran.us.r-project.org")
```

```
## Warning: package 'glmnet' is in use and will not be installed
```

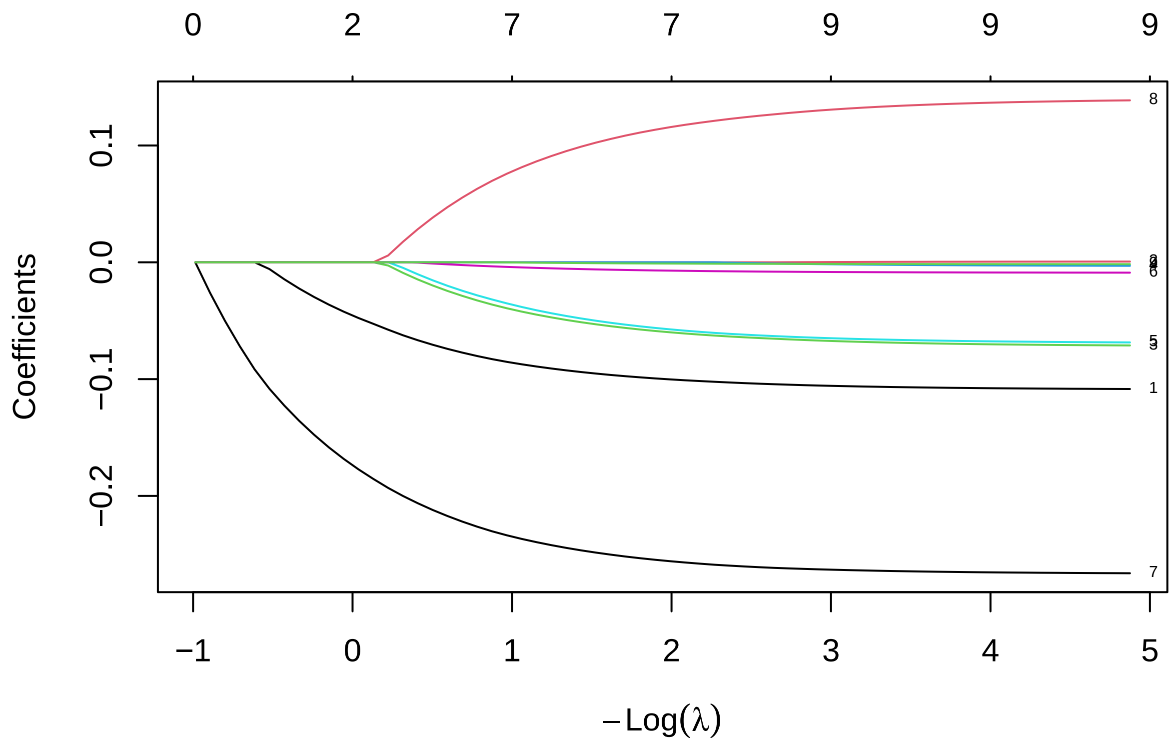
```
library(glmnet)

fitl <- glmnet(x=X,y=Y,family="gaussian",alpha=1)
plot(fitl,col=1:9)
legend("center",legend=names(atlas2)[2:10],col=1:9,lty=rep(1,9),cex=.8)
```



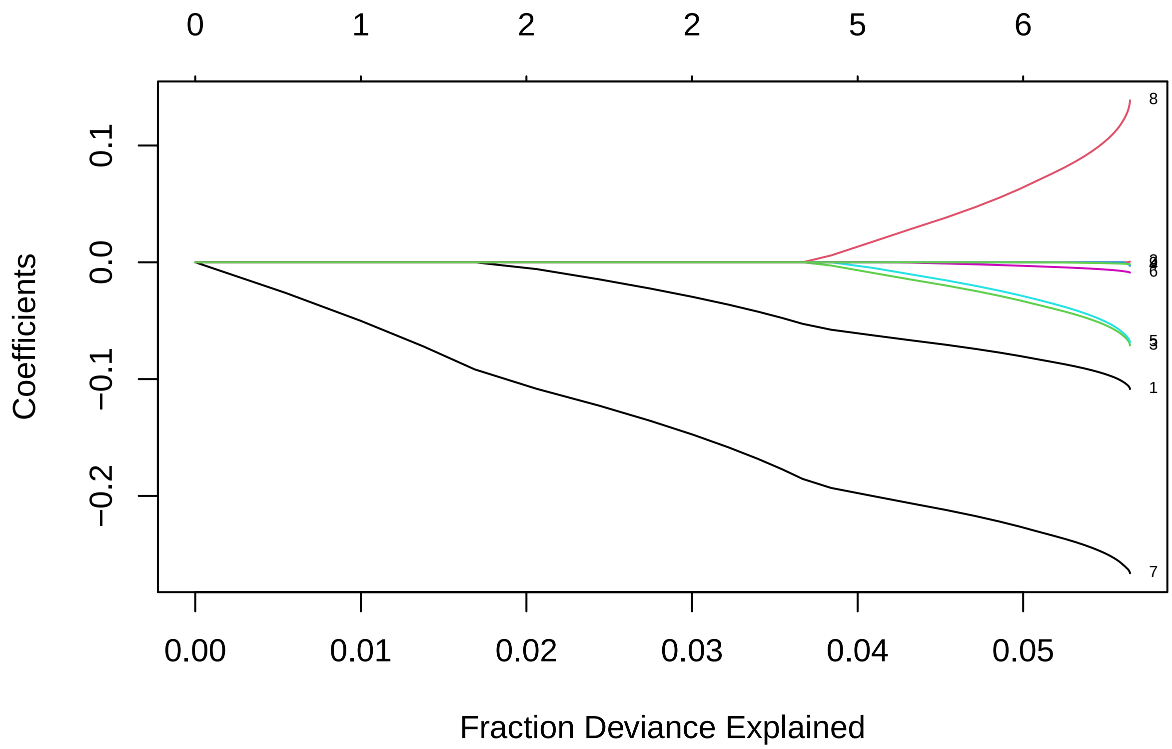

```
fitl <- glmnet(x=X,y=Y,family="gaussian",alpha=1)
plot(fitl,col=1:9)
```




```
plot(fitl, xvar = "lambda", label = TRUE)
```



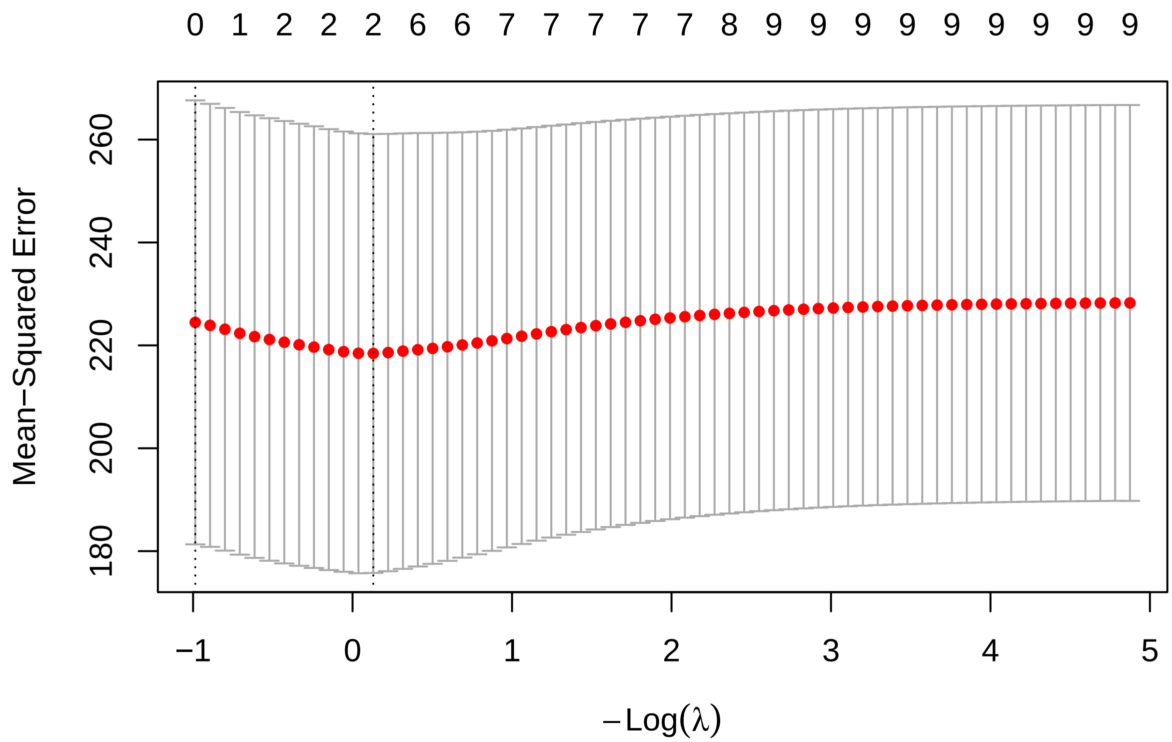

```r
plot(fitl, xvar = "dev", label = TRUE)
```




```r
cvfit <- cv.glmnet(X, Y)
print(cvfit)
```

```
##
## Call:  cv.glmnet(x = X, y = Y)
##
## Measure: Mean-Squared Error
##
##     Lambda Index Measure    SE Nonzero
## min  0.878    13   218.4 42.66       2
## 1se  2.681     1   224.5 43.16       0
```

```r
plot(cvfit)
```




```
cvfit$lambda.min
```

```
## [1] 0.8780333
```

```
coef(cvfit, s = "lambda.min")
```

```
## 10 x 1 sparse Matrix of class "dgCMatrix"
##                       lambda.min
## (Intercept)        -4.011030e-16
## PCH_GROC_16_20     -5.265520e-02
## PCH_SUPERC_16_20     .
## PCH_CONVS_16_20      .
## PCH_SPECS_16_20      .
## PCH_SNAPS_17_23      .
## PCH_WICS_16_22       .
## PCH_FFR_16_20      -1.854368e-01
## PCH_FSR_16_20        .
## PCH_DIRSALES_12_17   .
```

```
predict(cvfit, newx = X[1:5,], s = "lambda.min")
```

```
##       lambda.min
## [1,] -0.7096276
## [2,]  0.3929920
## [3,]  1.6032029
## [4,] -0.3747832
## [5,] -0.3638456
```

```r
Y[1:5]
```

```
## [1] -1.5495514 -8.4206962 -0.3117743 10.0700131 -3.9625370
```

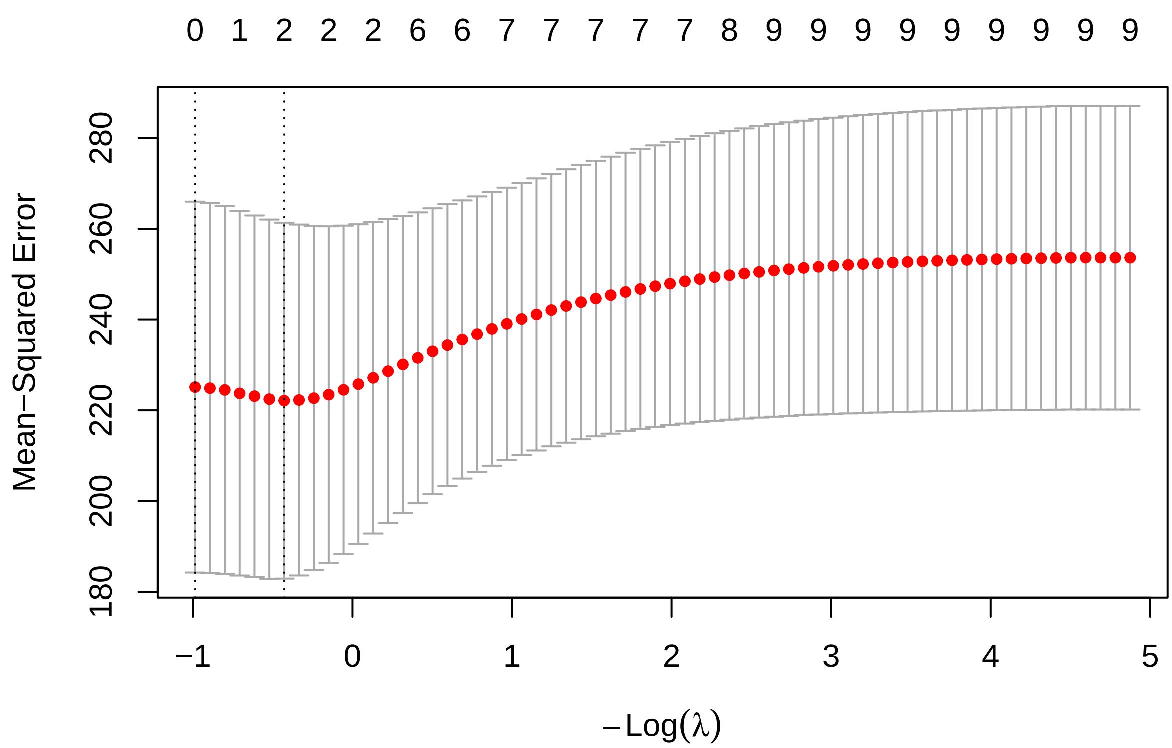
```r
cvfit2 <- cv.glmnet(X, Y, type.measure = "mse", nfolds = 5)
print(cvfit2)
```

```
##
## Call:  cv.glmnet(x = X, y = Y, type.measure = "mse", nfolds = 5)
##
## Measure: Mean-Squared Error
##
##     Lambda Index Measure    SE Nonzero
## min  1.534     7   222.1 39.21       2
## 1se  2.681     1   225.1 40.86       0
```

```r
cvfit2$lambda.min
```

```
## [1] 1.534388
```

```r
plot(cvfit2)
```



Principal Components Regression:

Immediately from looking at the scatter plots, supercenters and convenience stores do not appear to be good candidates for a principal components regression. The bar graphs suggest that the first 3 or 4 variables is enough to explain most of the variance. The mean squared error is only slightly higher than the original model.

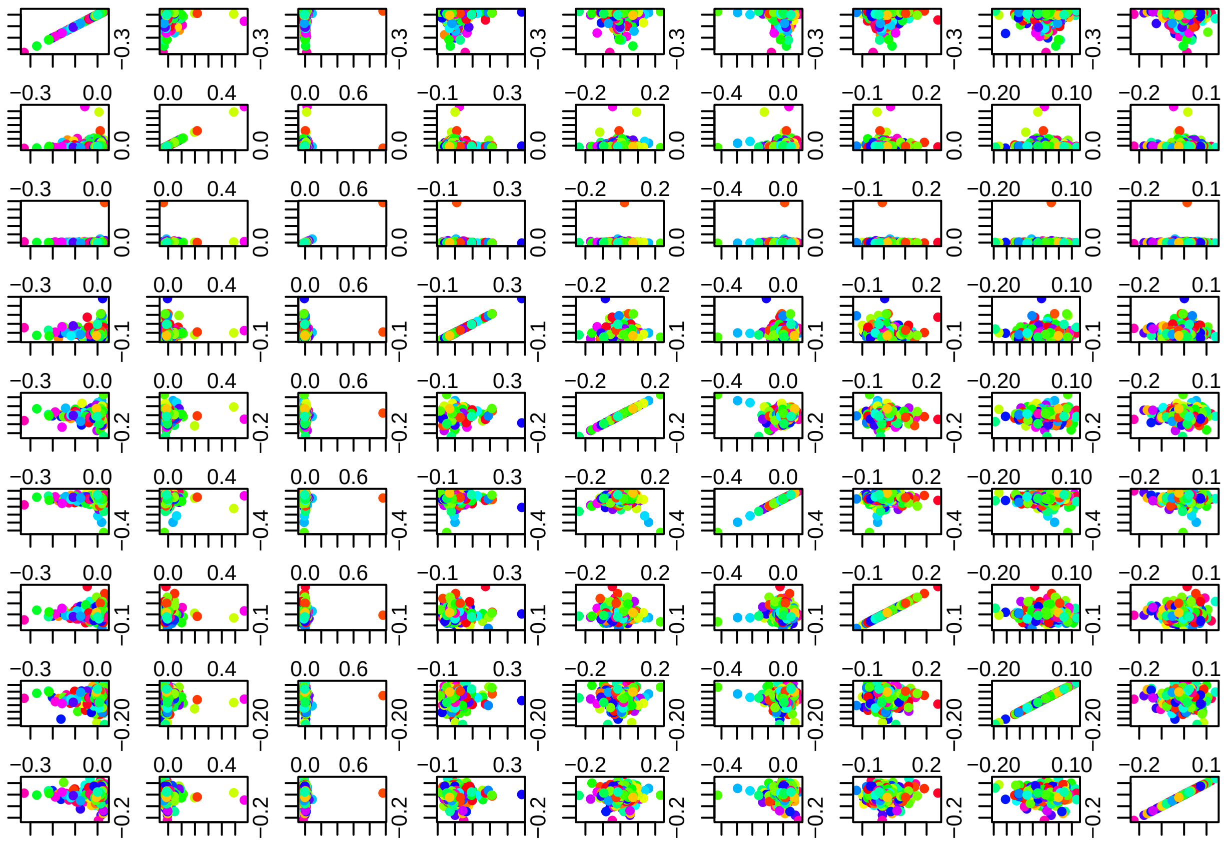
```r
svdx = svd(X)
svdx$d
```

```
## [1] 5485.2502 2935.3203 1966.5020  787.5126  379.3500  335.9742  255.3708
## [8]  215.6719  191.0439
```

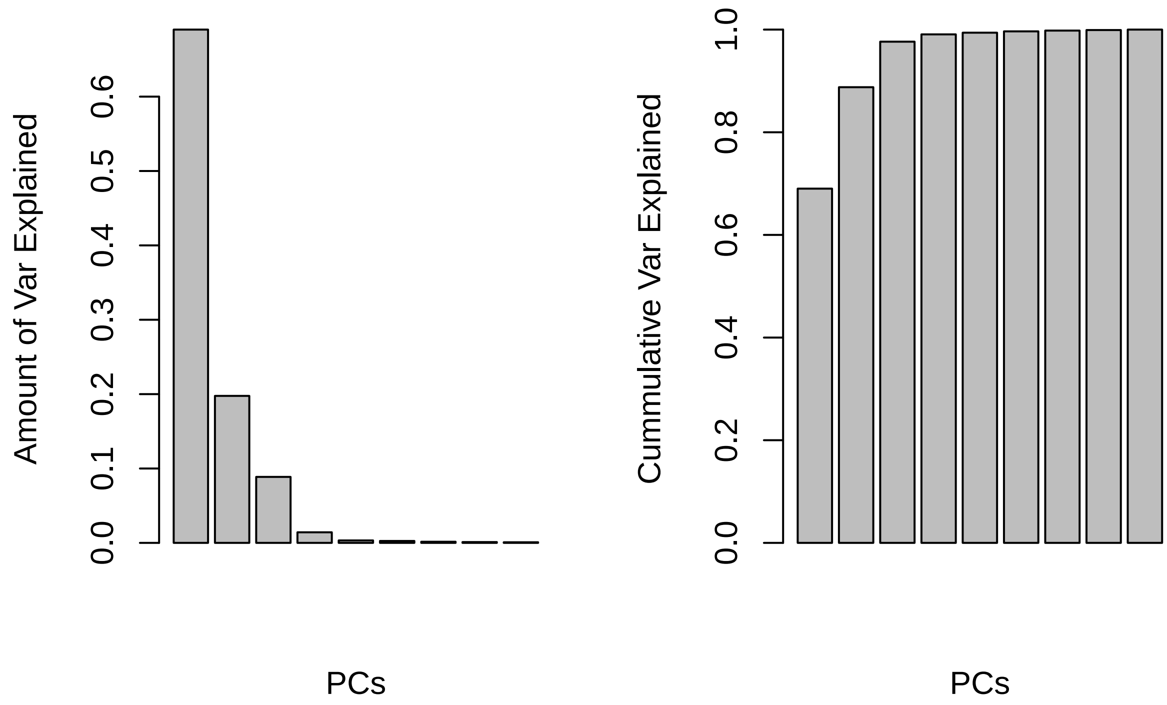
```r
svdx$v
```

```
##                [,1]         [,2]          [,3]         [,4]         [,5]
##  [1,] -0.0038933753 -0.005138437 -0.0005040145 -0.017682233 -0.887783723
##  [2,] -0.0502320909  0.998009582  0.0334506377  0.010050905 -0.009911546
##  [3,] -0.0017865017  0.009531256  0.0041066886  0.019352641  0.406713802
##  [4,]  0.0029360868 -0.009206351 -0.0201141715  0.998648747 -0.015494105
##  [5,]  0.0017662523 -0.001117619 -0.0045850525  0.027989848 -0.172380265
##  [6,]  0.0125176939 -0.033116383  0.9991582325  0.019827022 -0.003207317
##  [7,] -0.0025127925 -0.007343337 -0.0022440839  0.020350290 -0.123154390
##  [8,]  0.0009810497 -0.007716170 -0.0001883104  0.017729878 -0.034404540
##  [9,] -0.9986404146 -0.050630630  0.0107744348  0.002729105  0.003117598
##               [,6]         [,7]          [,8]          [,9]
##  [1,] -0.373714107 -0.239768946  0.1090362430  0.0494499727
##  [2,]  0.005467566  0.002183143 -0.0067812568 -0.0074000194
##  [3,] -0.905703613 -0.069369938  0.0158804068  0.0934614689
##  [4,]  0.018361259 -0.034239598  0.0175445251  0.0120564970
##  [5,] -0.136562202  0.952681809  0.1837840548  0.0971996844
##  [6,]  0.002875377  0.004091646 -0.0008159195  0.0006051756
##  [7,] -0.073831261  0.145281372 -0.9744817620  0.0904497138
##  [8,] -0.124722801  0.088262018 -0.0640794816 -0.9853878937
##  [9,]  0.002713956  0.002305803  0.0026430539 -0.0009684370
```

```r
par(mar=c(1,1,1,1))
layout(matrix(1:81,9,9))
mycols = rainbow(length(Y))
orY = order(Y)
for(i in 1:9)
{
  for(j in 1:9)
    {
      plot(svdx$u[,i],svdx$u[,j],type="p",pch=16,col=mycols[orY])
    }
}
```



```r
#amount of variance explained
varex = 0; cumvarex = 0;
for(i in 1:9)
{
  varex[i] = svdx$d[i]^2/sum(svdx$d^2)
  cumvarex[i] = sum(varex)
}
par(mfrow=c(1,2))
par(mar=c(5,4,4,2))
barplot(varex,ylab="Amount of Var Explained",xlab="PCs")
barplot(cumvarex,ylab="Cummulative Var Explained",xlab="PCs")
```




```r
# ridge paths again

plot(c(1,length(lambdas)),range(betasr),type="n",ylab="Coefficients",xlab="Lambda Index")
for(j in 1:9)
{
  lines(betasr[length(lambdas):1,j],col=j)
}
legend(0,20,legend=names(atlas2)[2:10],col=1:10,lty=rep(1,9))

# principal components regression

betapcr <- diag(svdx$d)%*%t(svdx$u)%*%Y
ypcr <- svdx$u[,1:2]%*%t(svdx$u[,1:2])%*%Y
mean((predict(fit)-atlas2[,1])^2)
```

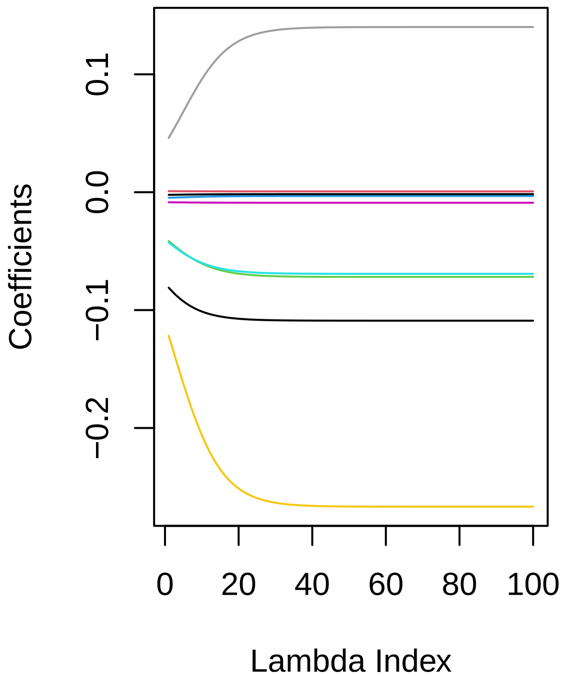
```
## [1] 211.046
```

```r
mean((ypcr-Y)^2)
```

```
## [1] 223.2447
```

```r
mse_pcr <- NULL
for(i in 1:9){
    ypcr <- svdx$u[,1:i]%*%t(svdx$u[,1:i])%*%Y
    mse_pcr <- c(mse_pcr, mean((ypcr-Y)^2))
}
mse_pcr
```

```
## [1] 223.2990 223.2447 222.8075 222.7454 219.9703 217.3770 216.9704 213.0400
## [9] 211.0460
```

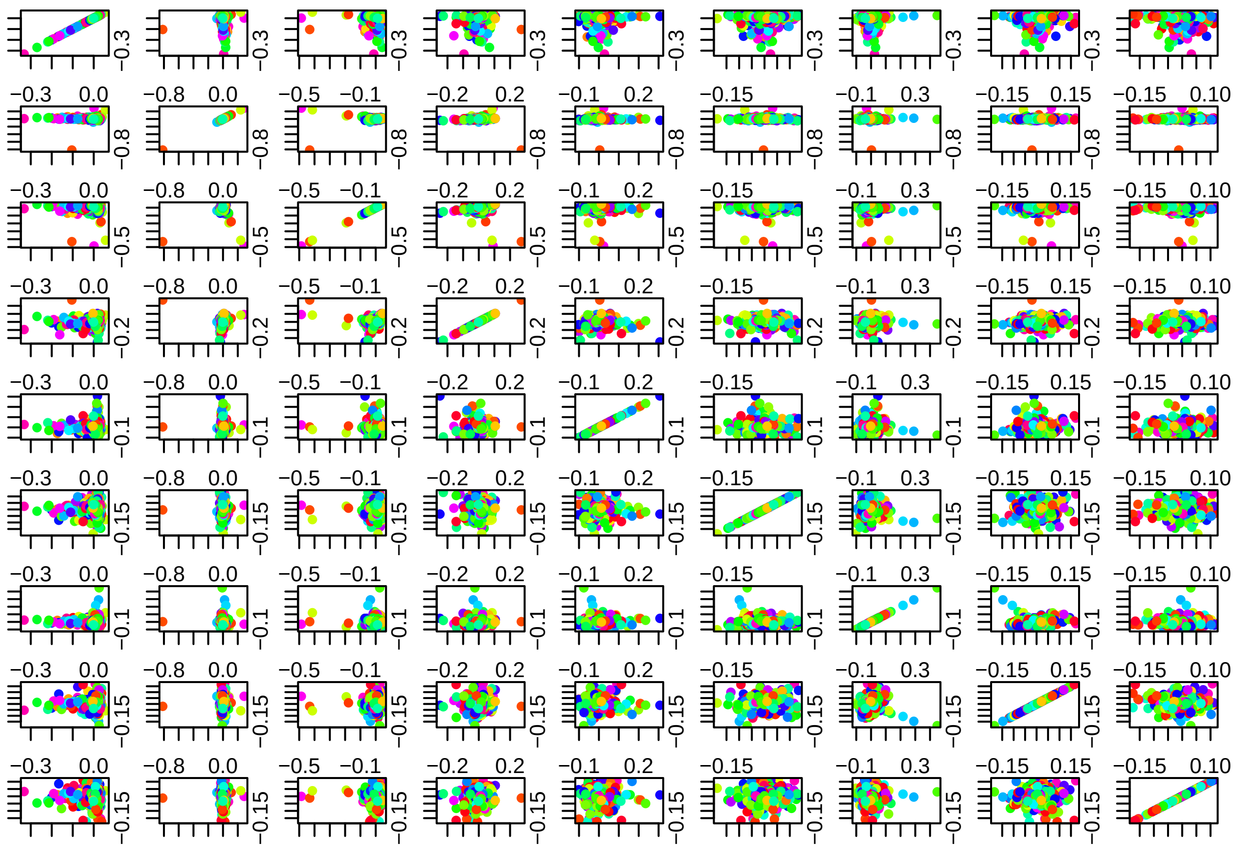


Partial Least Squares:

From the scatter plots, there are high correlations for all variables except supercenters and convenience stores. This model chooses all other variables, which is three more than the four included in the original regression. The partial least squares values look extremely different from the principal components regression.

```
plsfunc <- function(x,y)
{
    p <- ncol(x)
    n <- nrow(x)
    M <- t(x)%*%y
    Z <- NULL; V <- NULL; P <- NULL;
    for(k in 1:p){
      svdm <- svd(M)
      z <- x%*%svdm$u
      z <- z*as.numeric(1/sqrt(t(z)%*%z))
      V <- cbind(V,svdm$u)
      p <- t(x)%*%z/as.numeric(t(z)%*%z)
      P <- cbind(P,p);
      Z <- cbind(Z,z);
      M <- M - P%*%solve(t(P)%*%P)%*%t(P)%*%M;
    }
  return(list(Z=Z,V=V))
}
```

```r
plsx <- plsfunc(X,Y)

# scatterplots of PLS components

par(mar=c(1,1,1,1))
layout(matrix(1:81,9,9))
mycols <- rainbow(length(Y))
orY <- order(Y)
for(i in 1:9)
{
  for(j in 1:9)
    {
      plot(plsx$Z[,i],plsx$Z[,j],type="p",pch=16,col=mycols[orY])
    }
}
```



```r
betapls = t(plsx$Z)%*%Y

cbind(betapcr,betapls)
```

```
##              [,1]       [,2]
## [1,]  82163.287 18.9041081
## [2,]  16750.307 22.8618721
## [3,] -31848.270 24.1716572
## [4,]  -4810.084 64.8951570
## [5,]  15479.250 27.4537198
```

```
## [6,]  13252.838 32.3496279
## [7,]  -3989.039  9.4936885
## [8,]  10473.297  3.0996293
## [9,]  -6608.012  0.7131445
```

2. Model Assessment (R) Choose a dataset and split into training/validation sets. Implement linear regression models (degrees 1 to 10) in R. You have developed multiple linear regression models for a dataset, but you are unsure which model to choose. Use the following model assessment techniques: A. Bias-Variance Tradeoff: Explain the concept of bias-variance tradeoff in the context of model selection. Evaluate the bias and variance of your models and discuss how they relate to model complexity. B. Information Criteria: Calculate and compare the Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), and F statistic for your models. Interpret the results and recommend the best model based on these criteria.

Degree - MSE - AIC - BIC - F statistic

1 - 282.4778 - 3914.723 - 3960.634 - 2.559, p-value: 0.007037

2 - 271.488 - 3922.186 - 4005.662 - 1.851, p-value: 0.01802

3 - 339.6882 - 3918.573 - 4039.613 - 2.037, p-value: 0.001847

4 - 562.2987 - 3910.399 - 4069.003 - 2.271, p-value: 6.541e-05

5 - 2829.748 - 3915.607 - 4111.775 - 2.088, p-value: 9.737e-05

6 - 3307.753 - 3923.67 - 4157.402 - 1.904, p-value: 0.0002601

7 - 16504.31 - 3927.542 - 4190.491 - 1.84, p-value: 0.0003018

8 - 95229.63 - 3935.103 - 4231.442 - 1.729, p-value: 0.0006672

9 - 2511899 - 3927.937 - 4257.666 - 1.852, p-value: 7.813e-05

10 - 23562511 - 3934.527 - 4293.472 - 1.767, p-value: 0.0001673

As the model becomes more complex (higher degree), the MSE increases, meaning that the sum of bias and variance increases. The degree 2 model has the lowest MSE. The lowest AIC is for the degree 4 model, the lowest BIC is for the degree 1 model, and the highest F-statistic is for the degree 1. I recommend the degree 1 model based on these results; this model is also the second lowest AIC.

```r
label.splitting <- sample(1:5, nrow(atlas2), replace = TRUE)
label.splitting
```

```
##   [1] 2 3 2 2 4 3 4 5 1 2 1 4 3 4 4 3 4 5 5 2 4 3 2 5 3 2 4 1 5 4 3 4 2 2 4 2 4
##  [38] 5 1 1 2 4 4 1 3 5 2 3 2 3 2 2 2 1 1 1 2 3 1 4 4 4 5 2 5 1 5 2 3 2 5 4 5 3
##  [75] 4 1 3 1 3 2 1 4 4 3 2 3 1 5 5 1 1 2 4 2 1 5 2 1 2 4 4 3 1 1 1 3 4 4 3 1 4
## [112] 4 4 3 3 2 2 3 4 1 5 4 4 2 4 3 1 3 1 4 4 1 2 4 4 5 3 4 2 2 2 1 3 3 2 4 2 3
## [149] 3 1 1 5 4 1 1 1 1 1 2 5 5 2 1 1 3 5 1 1 5 5 2 4 3 4 2 3 2 3 1 3 1 1 2 5 2
## [186] 4 2 2 3 3 4 3 1 4 1 1 5 1 5 4 3 5 1 3 5 4 3 2 3 1 3 3 2 1 1 3 4 3 5 3 1 5
## [223] 2 4 2 3 3 4 4 2 3 5 3 2 4 5 4 3 2 5 2 2 2 2 4 1 5 1 1 4 5 4 4 1 2 5 5 1 1
## [260] 5 5 4 2 3 3 5 1 1 1 2 2 3 2 5 3 4 1 3 3 2 4 1 2 5 1 1 4 3 4 4 5 2 1 3 1 1
## [297] 5 3 2 2 3 2 4 4 5 5 1 2 4 4 4 5 3 4 2 2 2 1 5 5 1 3 2 3 1 3 5 1 4 3 3 2 5
## [334] 1 3 3 2 4 3 5 3 5 2 5 1 1 5 4 1 1 2 2 3 4 4 4 5 2 5 1 5 1 1 1 5 3 4 3 1 3
## [371] 4 2 4 3 2 3 2 3 2 5 1 1 4 3 2 2 3 3 5 4 4 2 3 5 2 2 5 3 1 3 4 3 5 3 2 5 2
## [408] 3 4 2 1 3 5 1 1 3 4 5 3 2 3 3 4 5 5 1 5 1 4 3 3 5 4 4 1 3 1 2 5 1 1 1 2 4
## [445] 4 2 4 3 5 2 3 1 5 1 1 1 3 1 1 2 3 2 4 1 5 2 1 3 3 3 4 3 5 5 4 5 5 5 2 4 5
## [482] 1 4 3 2 1 3 4 2 4 3 4 5 2 3 5 3 2 1 1 5 4 5 2 5 4 4 2 3 4 3 2 4 4 4 2 2 4
## [519] 1 3 1 5 4 4 4 1 2 3 3 3 4 4 4 4 2 5 2 1 2 1 4 5 4 5 4 1 5 2 5 1 5 2 4 4 5
## [556] 4 5 4 4 5 1 4 1 5 1 1 2 4 5 4 2 1 3 3 5 2 5 5 3 3 2 2 2 1 1 3 2 4 1 5 5 1
## [593] 2 2 1 5 4 3 5 4
```

```r
for (k in 1:5) {
  dat.valid <- atlas2[label.splitting == k, ]
  dat.train <- atlas2[label.splitting != k, ]
}

fitdeg_1 <- lm(PCH_LACCESS_POP_15_19 ~ ., data = dat.train)
predictions_1 <- predict(fitdeg_1, newdata = dat.valid)
mean((dat.valid$PCH_LACCESS_POP_15_19 - predictions_1)^2)
```

```
## [1] 229.0101
```

```r
AIC(fitdeg_1)
```

```
## [1] 4121.316
```

```r
BIC(fitdeg_1)
```

```
## [1] 4167.566
```

```r
summary(fitdeg_1)
```

```
##
## Call:
## lm(formula = PCH_LACCESS_POP_15_19 ~ ., data = dat.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -58.218  -7.056  -1.234   5.200 133.775
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           2.3474857  1.3179834   1.781  0.07552 .
## PCH_GROC_16_20       -0.0915038  0.0472312  -1.937  0.05328 .
## PCH_SUPERC_16_20      0.0005455  0.0054787   0.100  0.92073
## PCH_CONVS_16_20      -0.0783678  0.0489727  -1.600  0.11020
## PCH_SPECS_16_20      -0.0035107  0.0212771  -0.165  0.86901
## PCH_SNAPS_17_23      -0.0876681  0.0681724  -1.286  0.19906
## PCH_WICS_16_22       -0.0641108  0.0354497  -1.809  0.07115 .
## PCH_FFR_16_20        -0.2399929  0.0771652  -3.110  0.00198 **
## PCH_FSR_16_20         0.1476720  0.0867343   1.703  0.08929 .
## PCH_DIRSALES_12_17 -0.0020260  0.0030366  -0.667  0.50496
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.36 on 485 degrees of freedom
## Multiple R-squared:  0.05464,    Adjusted R-squared:  0.0371
## F-statistic: 3.115 on 9 and 485 DF,  p-value: 0.001178
```

```r
fitdeg_2 <- lm(PCH_LACCESS_POP_15_19 ~ poly(PCH_GROC_16_20, 2, raw=TRUE) + poly(PCH_SUPERC_16_20, 2, ra
predictions_2 <- predict(fitdeg_2, newdata = dat.valid)
mean((dat.valid$PCH_LACCESS_POP_15_19 - predictions_2)^2)
```

```
## [1] 120731
```

```r
AIC(fitdeg_2)
```

```
## [1] 4115.176
```

```r
BIC(fitdeg_2)
```

```
## [1] 4199.267
```

```r
summary(fitdeg_2)
```

```
##
## Call:
## lm(formula = PCH_LACCESS_POP_15_19 ~ poly(PCH_GROC_16_20, 2,
##     raw = TRUE) + poly(PCH_SUPERC_16_20, 2, raw = TRUE) + poly(PCH_CONVS_16_20,
##     2, raw = TRUE) + poly(PCH_SPECS_16_20, 2, raw = TRUE) + poly(PCH_SNAPS_17_23,
##     2, raw = TRUE) + poly(PCH_WICS_16_22, 2, raw = TRUE) + poly(PCH_FFR_16_20,
##     2, raw = TRUE) + poly(PCH_FSR_16_20, 2, raw = TRUE) + poly(PCH_DIRSALES_12_17,
##     2, raw = TRUE), data = dat.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -49.701  -7.396  -1.370   5.782 127.629
##
## Coefficients:
##                                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                               2.068e+00  1.574e+00   1.314  0.18955
## poly(PCH_GROC_16_20, 2, raw = TRUE)1     -1.345e-01  4.965e-02  -2.709  0.00699
## poly(PCH_GROC_16_20, 2, raw = TRUE)2      2.293e-03  1.192e-03   1.924  0.05499
## poly(PCH_SUPERC_16_20, 2, raw = TRUE)1   -5.188e-03  1.089e-02  -0.476  0.63394
## poly(PCH_SUPERC_16_20, 2, raw = TRUE)2    5.117e-06  7.985e-06   0.641  0.52189
## poly(PCH_CONVS_16_20, 2, raw = TRUE)1    -1.375e-01  6.851e-02  -2.007  0.04531
## poly(PCH_CONVS_16_20, 2, raw = TRUE)2     5.415e-04  6.838e-04   0.792  0.42883
## poly(PCH_SPECS_16_20, 2, raw = TRUE)1    -2.953e-03  2.807e-02  -0.105  0.91626
## poly(PCH_SPECS_16_20, 2, raw = TRUE)2     3.484e-05  1.916e-04   0.182  0.85577
## poly(PCH_SNAPS_17_23, 2, raw = TRUE)1     4.697e-02  1.325e-01   0.354  0.72319
## poly(PCH_SNAPS_17_23, 2, raw = TRUE)2    -3.973e-03  3.542e-03  -1.122  0.26255
## poly(PCH_WICS_16_22, 2, raw = TRUE)1     -8.362e-02  3.557e-02  -2.351  0.01912
## poly(PCH_WICS_16_22, 2, raw = TRUE)2      1.027e-03  4.620e-04   2.222  0.02673
## poly(PCH_FFR_16_20, 2, raw = TRUE)1      -9.710e-02  1.119e-01  -0.868  0.38608
## poly(PCH_FFR_16_20, 2, raw = TRUE)2      -7.215e-03  4.169e-03  -1.731  0.08414
## poly(PCH_FSR_16_20, 2, raw = TRUE)1       2.974e-01  1.017e-01   2.926  0.00360
## poly(PCH_FSR_16_20, 2, raw = TRUE)2      -1.645e-02  5.536e-03  -2.972  0.00311
## poly(PCH_DIRSALES_12_17, 2, raw = TRUE)1 -1.028e-02  6.358e-03  -1.618  0.10640
## poly(PCH_DIRSALES_12_17, 2, raw = TRUE)2  8.091e-06  5.430e-06   1.490  0.13692
##
## (Intercept)
## poly(PCH_GROC_16_20, 2, raw = TRUE)1     **
## poly(PCH_GROC_16_20, 2, raw = TRUE)2     .
## poly(PCH_SUPERC_16_20, 2, raw = TRUE)1
## poly(PCH_SUPERC_16_20, 2, raw = TRUE)2
## poly(PCH_CONVS_16_20, 2, raw = TRUE)1    *
## poly(PCH_CONVS_16_20, 2, raw = TRUE)2
## poly(PCH_SPECS_16_20, 2, raw = TRUE)1
## poly(PCH_SPECS_16_20, 2, raw = TRUE)2
## poly(PCH_SNAPS_17_23, 2, raw = TRUE)1
## poly(PCH_SNAPS_17_23, 2, raw = TRUE)2
## poly(PCH_WICS_16_22, 2, raw = TRUE)1     *
## poly(PCH_WICS_16_22, 2, raw = TRUE)2     *
```

```
## poly(PCH_FFR_16_20, 2, raw = TRUE)1
## poly(PCH_FFR_16_20, 2, raw = TRUE)2      .
## poly(PCH_FSR_16_20, 2, raw = TRUE)1      **
## poly(PCH_FSR_16_20, 2, raw = TRUE)2      **
## poly(PCH_DIRSALES_12_17, 2, raw = TRUE)1
## poly(PCH_DIRSALES_12_17, 2, raw = TRUE)2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.13 on 476 degrees of freedom
## Multiple R-squared:  0.09964,    Adjusted R-squared:  0.06559
## F-statistic: 2.927 on 18 and 476 DF,  p-value: 5.727e-05
```

```r
fitdeg_3 <- lm(PCH_LACCESS_POP_15_19 ~ poly(PCH_GROC_16_20, 3, raw=TRUE) + poly(PCH_SUPERC_16_20, 3, ra
predictions_3 <- predict(fitdeg_3, newdata = dat.valid)
mean((dat.valid$PCH_LACCESS_POP_15_19 - predictions_3)^2)
```

```
## [1] 81592331
```

```r
AIC(fitdeg_3)
```

```
## [1] 4119.613
```

```r
BIC(fitdeg_3)
```

```
## [1] 4241.545
```

```r
summary(fitdeg_3)
```

```
##
## Call:
## lm(formula = PCH_LACCESS_POP_15_19 ~ poly(PCH_GROC_16_20, 3,
##     raw = TRUE) + poly(PCH_SUPERC_16_20, 3, raw = TRUE) + poly(PCH_CONVS_16_20,
##     3, raw = TRUE) + poly(PCH_SPECS_16_20, 3, raw = TRUE) + poly(PCH_SNAPS_17_23,
##     3, raw = TRUE) + poly(PCH_WICS_16_22, 3, raw = TRUE) + poly(PCH_FFR_16_20,
##     3, raw = TRUE) + poly(PCH_FSR_16_20, 3, raw = TRUE) + poly(PCH_DIRSALES_12_17,
##     3, raw = TRUE), data = dat.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.484  -7.203  -1.067   5.013 130.393
##
## Coefficients:
##                                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              2.640e+00  1.859e+00   1.420  0.15641
## poly(PCH_GROC_16_20, 3, raw = TRUE)1    -1.491e-01  5.892e-02  -2.530  0.01172
## poly(PCH_GROC_16_20, 3, raw = TRUE)2     2.034e-03  2.081e-03   0.977  0.32889
## poly(PCH_GROC_16_20, 3, raw = TRUE)3     1.952e-06  2.812e-05   0.069  0.94468
## poly(PCH_SUPERC_16_20, 3, raw = TRUE)1  -1.056e-02  1.925e-02  -0.548  0.58370
## poly(PCH_SUPERC_16_20, 3, raw = TRUE)2   2.602e-05  5.527e-05   0.471  0.63799
## poly(PCH_SUPERC_16_20, 3, raw = TRUE)3  -1.057e-08  2.740e-08  -0.386  0.70004
## poly(PCH_CONVS_16_20, 3, raw = TRUE)1   -9.864e-02  7.389e-02  -1.335  0.18253
## poly(PCH_CONVS_16_20, 3, raw = TRUE)2   -1.297e-03  2.966e-03  -0.437  0.66213
## poly(PCH_CONVS_16_20, 3, raw = TRUE)3    1.056e-05  1.784e-05   0.592  0.55405
## poly(PCH_SPECS_16_20, 3, raw = TRUE)1    3.836e-04  2.880e-02   0.013  0.98938
## poly(PCH_SPECS_16_20, 3, raw = TRUE)2   -5.758e-04  5.396e-04  -1.067  0.28645
## poly(PCH_SPECS_16_20, 3, raw = TRUE)3    2.248e-06  1.874e-06   1.199  0.23103
```

```
## poly(PCH_SNAPS_17_23, 3, raw = TRUE)1    -1.090e-01  1.618e-01  -0.674  0.50075
## poly(PCH_SNAPS_17_23, 3, raw = TRUE)2     1.382e-02  1.026e-02   1.346  0.17883
## poly(PCH_SNAPS_17_23, 3, raw = TRUE)3    -3.178e-04  1.711e-04  -1.857  0.06398
## poly(PCH_WICS_16_22, 3, raw = TRUE)1      1.386e-03  4.834e-02   0.029  0.97714
## poly(PCH_WICS_16_22, 3, raw = TRUE)2      2.381e-03  7.321e-04   3.252  0.00123
## poly(PCH_WICS_16_22, 3, raw = TRUE)3     -1.475e-05  6.000e-06  -2.458  0.01433
## poly(PCH_FFR_16_20, 3, raw = TRUE)1      -1.143e-01  1.127e-01  -1.015  0.31083
## poly(PCH_FFR_16_20, 3, raw = TRUE)2      -2.819e-03  8.263e-03  -0.341  0.73319
## poly(PCH_FFR_16_20, 3, raw = TRUE)3      -1.152e-04  1.764e-04  -0.653  0.51406
## poly(PCH_FSR_16_20, 3, raw = TRUE)1       1.872e-01  1.229e-01   1.523  0.12841
## poly(PCH_FSR_16_20, 3, raw = TRUE)2      -2.367e-02  8.941e-03  -2.647  0.00839
## poly(PCH_FSR_16_20, 3, raw = TRUE)3       3.851e-04  3.415e-04   1.128  0.26009
## poly(PCH_DIRSALES_12_17, 3, raw = TRUE)1 -8.415e-03  1.018e-02  -0.826  0.40900
## poly(PCH_DIRSALES_12_17, 3, raw = TRUE)2  4.520e-06  2.037e-05   0.222  0.82446
## poly(PCH_DIRSALES_12_17, 3, raw = TRUE)3  1.628e-09  9.326e-09   0.175  0.86149
##
## (Intercept)
## poly(PCH_GROC_16_20, 3, raw = TRUE)1     *
## poly(PCH_GROC_16_20, 3, raw = TRUE)2
## poly(PCH_GROC_16_20, 3, raw = TRUE)3
## poly(PCH_SUPERC_16_20, 3, raw = TRUE)1
## poly(PCH_SUPERC_16_20, 3, raw = TRUE)2
## poly(PCH_SUPERC_16_20, 3, raw = TRUE)3
## poly(PCH_CONVS_16_20, 3, raw = TRUE)1
## poly(PCH_CONVS_16_20, 3, raw = TRUE)2
## poly(PCH_CONVS_16_20, 3, raw = TRUE)3
## poly(PCH_SPECS_16_20, 3, raw = TRUE)1
## poly(PCH_SPECS_16_20, 3, raw = TRUE)2
## poly(PCH_SPECS_16_20, 3, raw = TRUE)3
## poly(PCH_SNAPS_17_23, 3, raw = TRUE)1
## poly(PCH_SNAPS_17_23, 3, raw = TRUE)2
## poly(PCH_SNAPS_17_23, 3, raw = TRUE)3    .
## poly(PCH_WICS_16_22, 3, raw = TRUE)1
## poly(PCH_WICS_16_22, 3, raw = TRUE)2     **
## poly(PCH_WICS_16_22, 3, raw = TRUE)3     *
## poly(PCH_FFR_16_20, 3, raw = TRUE)1
## poly(PCH_FFR_16_20, 3, raw = TRUE)2
## poly(PCH_FFR_16_20, 3, raw = TRUE)3
## poly(PCH_FSR_16_20, 3, raw = TRUE)1
## poly(PCH_FSR_16_20, 3, raw = TRUE)2      **
## poly(PCH_FSR_16_20, 3, raw = TRUE)3
## poly(PCH_DIRSALES_12_17, 3, raw = TRUE)1
## poly(PCH_DIRSALES_12_17, 3, raw = TRUE)2
## poly(PCH_DIRSALES_12_17, 3, raw = TRUE)3
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.07 on 467 degrees of freedom
## Multiple R-squared:  0.124,  Adjusted R-squared:  0.07333
## F-statistic: 2.448 on 27 and 467 DF,  p-value: 9.176e-05
```

```r
fitdeg_4 <- lm(PCH_LACCESS_POP_15_19 ~ poly(PCH_GROC_16_20, 4, raw=TRUE) + poly(PCH_SUPERC_16_20, 4, ra
predictions_4 <- predict(fitdeg_4, newdata = dat.valid)
mean((dat.valid$PCH_LACCESS_POP_15_19 - predictions_4)^2)
```

```
## [1] 42719881974
```

```r
AIC(fitdeg_4)
```

```
## [1] 4127.585
```

```r
BIC(fitdeg_4)
```

```
## [1] 4287.358
```

```r
summary(fitdeg_4)
```

```
##
## Call:
## lm(formula = PCH_LACCESS_POP_15_19 ~ poly(PCH_GROC_16_20, 4,
##     raw = TRUE) + poly(PCH_SUPERC_16_20, 4, raw = TRUE) + poly(PCH_CONVS_16_20,
##     4, raw = TRUE) + poly(PCH_SPECS_16_20, 4, raw = TRUE) + poly(PCH_SNAPS_17_23,
##     4, raw = TRUE) + poly(PCH_WICS_16_22, 4, raw = TRUE) + poly(PCH_FFR_16_20,
##     4, raw = TRUE) + poly(PCH_FSR_16_20, 4, raw = TRUE) + poly(PCH_DIRSALES_12_17,
##     4, raw = TRUE), data = dat.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.685  -6.683  -0.969   5.604 132.070
##
## Coefficients:
##                                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              1.948e+00  2.203e+00   0.884  0.37700
## poly(PCH_GROC_16_20, 4, raw = TRUE)1    -2.384e-01  7.815e-02  -3.051  0.00242
## poly(PCH_GROC_16_20, 4, raw = TRUE)2     3.422e-03  2.354e-03   1.454  0.14669
## poly(PCH_GROC_16_20, 4, raw = TRUE)3     1.149e-04  7.278e-05   1.579  0.11498
## poly(PCH_GROC_16_20, 4, raw = TRUE)4    -1.276e-06  7.815e-07  -1.633  0.10317
## poly(PCH_SUPERC_16_20, 4, raw = TRUE)1  -8.580e-03  2.857e-02  -0.300  0.76407
## poly(PCH_SUPERC_16_20, 4, raw = TRUE)2   1.345e-05  1.236e-04   0.109  0.91340
## poly(PCH_SUPERC_16_20, 4, raw = TRUE)3   2.687e-09  1.454e-07   0.018  0.98527
## poly(PCH_SUPERC_16_20, 4, raw = TRUE)4  -3.950e-12  4.993e-11  -0.079  0.93698
## poly(PCH_CONVS_16_20, 4, raw = TRUE)1   -7.537e-02  7.540e-02  -1.000  0.31803
## poly(PCH_CONVS_16_20, 4, raw = TRUE)2    3.831e-03  4.721e-03   0.811  0.41753
## poly(PCH_CONVS_16_20, 4, raw = TRUE)3   -1.002e-04  7.681e-05  -1.304  0.19279
## poly(PCH_CONVS_16_20, 4, raw = TRUE)4    4.799e-07  3.220e-07   1.490  0.13684
## poly(PCH_SPECS_16_20, 4, raw = TRUE)1    1.941e-02  3.560e-02   0.545  0.58584
## poly(PCH_SPECS_16_20, 4, raw = TRUE)2   -4.590e-04  7.968e-04  -0.576  0.56491
## poly(PCH_SPECS_16_20, 4, raw = TRUE)3   -2.173e-06  9.531e-06  -0.228  0.81974
## poly(PCH_SPECS_16_20, 4, raw = TRUE)4    1.240e-08  2.434e-08   0.510  0.61056
## poly(PCH_SNAPS_17_23, 4, raw = TRUE)1   -1.119e-01  1.671e-01  -0.670  0.50336
## poly(PCH_SNAPS_17_23, 4, raw = TRUE)2    1.404e-02  1.406e-02   0.999  0.31852
## poly(PCH_SNAPS_17_23, 4, raw = TRUE)3   -3.282e-04  6.378e-04  -0.515  0.60710
## poly(PCH_SNAPS_17_23, 4, raw = TRUE)4   -3.598e-08  8.127e-06  -0.004  0.99647
## poly(PCH_WICS_16_22, 4, raw = TRUE)1    -2.237e-02  5.002e-02  -0.447  0.65487
## poly(PCH_WICS_16_22, 4, raw = TRUE)2     4.072e-03  1.412e-03   2.883  0.00413
## poly(PCH_WICS_16_22, 4, raw = TRUE)3     2.887e-06  1.292e-05   0.224  0.82320
## poly(PCH_WICS_16_22, 4, raw = TRUE)4    -1.652e-07  1.127e-07  -1.466  0.14332
## poly(PCH_FFR_16_20, 4, raw = TRUE)1     -2.048e-01  1.512e-01  -1.355  0.17619
## poly(PCH_FFR_16_20, 4, raw = TRUE)2     -6.649e-03  8.766e-03  -0.758  0.44856
## poly(PCH_FFR_16_20, 4, raw = TRUE)3      4.624e-04  5.439e-04   0.850  0.39574
## poly(PCH_FFR_16_20, 4, raw = TRUE)4     -9.261e-06  8.593e-06  -1.078  0.28170
```

```
## poly(PCH_FSR_16_20, 4, raw = TRUE)1       1.939e-01  1.590e-01   1.219  0.22336
## poly(PCH_FSR_16_20, 4, raw = TRUE)2      -2.288e-02  1.042e-02  -2.195  0.02866
## poly(PCH_FSR_16_20, 4, raw = TRUE)3       3.338e-04  6.354e-04   0.525  0.59964
## poly(PCH_FSR_16_20, 4, raw = TRUE)4      -3.616e-06  1.799e-05  -0.201  0.84081
## poly(PCH_DIRSALES_12_17, 4, raw = TRUE)1 -1.918e-02  1.391e-02  -1.379  0.16852
## poly(PCH_DIRSALES_12_17, 4, raw = TRUE)2  5.663e-05  5.343e-05   1.060  0.28975
## poly(PCH_DIRSALES_12_17, 4, raw = TRUE)3 -5.949e-08  6.078e-08  -0.979  0.32825
## poly(PCH_DIRSALES_12_17, 4, raw = TRUE)4  1.938e-11  1.970e-11   0.984  0.32560
##
## (Intercept)
## poly(PCH_GROC_16_20, 4, raw = TRUE)1     **
## poly(PCH_GROC_16_20, 4, raw = TRUE)2
## poly(PCH_GROC_16_20, 4, raw = TRUE)3
## poly(PCH_GROC_16_20, 4, raw = TRUE)4
## poly(PCH_SUPERC_16_20, 4, raw = TRUE)1
## poly(PCH_SUPERC_16_20, 4, raw = TRUE)2
## poly(PCH_SUPERC_16_20, 4, raw = TRUE)3
## poly(PCH_SUPERC_16_20, 4, raw = TRUE)4
## poly(PCH_CONVS_16_20, 4, raw = TRUE)1
## poly(PCH_CONVS_16_20, 4, raw = TRUE)2
## poly(PCH_CONVS_16_20, 4, raw = TRUE)3
## poly(PCH_CONVS_16_20, 4, raw = TRUE)4
## poly(PCH_SPECS_16_20, 4, raw = TRUE)1
## poly(PCH_SPECS_16_20, 4, raw = TRUE)2
## poly(PCH_SPECS_16_20, 4, raw = TRUE)3
## poly(PCH_SPECS_16_20, 4, raw = TRUE)4
## poly(PCH_SNAPS_17_23, 4, raw = TRUE)1
## poly(PCH_SNAPS_17_23, 4, raw = TRUE)2
## poly(PCH_SNAPS_17_23, 4, raw = TRUE)3
## poly(PCH_SNAPS_17_23, 4, raw = TRUE)4
## poly(PCH_WICS_16_22, 4, raw = TRUE)1
## poly(PCH_WICS_16_22, 4, raw = TRUE)2     **
## poly(PCH_WICS_16_22, 4, raw = TRUE)3
## poly(PCH_WICS_16_22, 4, raw = TRUE)4
## poly(PCH_FFR_16_20, 4, raw = TRUE)1
## poly(PCH_FFR_16_20, 4, raw = TRUE)2
## poly(PCH_FFR_16_20, 4, raw = TRUE)3
## poly(PCH_FFR_16_20, 4, raw = TRUE)4
## poly(PCH_FSR_16_20, 4, raw = TRUE)1
## poly(PCH_FSR_16_20, 4, raw = TRUE)2      *
## poly(PCH_FSR_16_20, 4, raw = TRUE)3
## poly(PCH_FSR_16_20, 4, raw = TRUE)4
## poly(PCH_DIRSALES_12_17, 4, raw = TRUE)1
## poly(PCH_DIRSALES_12_17, 4, raw = TRUE)2
## poly(PCH_DIRSALES_12_17, 4, raw = TRUE)3
## poly(PCH_DIRSALES_12_17, 4, raw = TRUE)4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.07 on 458 degrees of freedom
## Multiple R-squared:  0.1415, Adjusted R-squared:  0.07407
## F-statistic: 2.098 on 36 and 458 DF,  p-value: 0.0003012
```

```r
fitdeg_5 <- lm(PCH_LACCESS_POP_15_19 ~ poly(PCH_GROC_16_20, 5, raw=TRUE) + poly(PCH_SUPERC_16_20, 5, ra
predictions_5 <- predict(fitdeg_5, newdata = dat.valid)
mean((dat.valid$PCH_LACCESS_POP_15_19 - predictions_5)^2)
```

```
## [1] 6.60578e+13
```

```r
AIC(fitdeg_5)
```

```
## [1] 4135.997
```

```r
BIC(fitdeg_5)
```

```
## [1] 4333.611
```

```r
summary(fitdeg_5)
```

```
##
## Call:
## lm(formula = PCH_LACCESS_POP_15_19 ~ poly(PCH_GROC_16_20, 5,
##     raw = TRUE) + poly(PCH_SUPERC_16_20, 5, raw = TRUE) + poly(PCH_CONVS_16_20,
##     5, raw = TRUE) + poly(PCH_SPECS_16_20, 5, raw = TRUE) + poly(PCH_SNAPS_17_23,
##     5, raw = TRUE) + poly(PCH_WICS_16_22, 5, raw = TRUE) + poly(PCH_FFR_16_20,
##     5, raw = TRUE) + poly(PCH_FSR_16_20, 5, raw = TRUE) + poly(PCH_DIRSALES_12_17,
##     5, raw = TRUE), data = dat.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -47.667  -6.587  -0.977   5.295 133.000
##
## Coefficients:
##                                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              6.138e-01  2.471e+00   0.248  0.80389
## poly(PCH_GROC_16_20, 5, raw = TRUE)1    -2.495e-01  7.871e-02  -3.170  0.00163
## poly(PCH_GROC_16_20, 5, raw = TRUE)2     2.104e-03  4.761e-03   0.442  0.65875
## poly(PCH_GROC_16_20, 5, raw = TRUE)3     1.254e-04  7.361e-05   1.703  0.08920
## poly(PCH_GROC_16_20, 5, raw = TRUE)4    -3.830e-07  2.630e-06  -0.146  0.88431
## poly(PCH_GROC_16_20, 5, raw = TRUE)5    -8.825e-09  2.204e-08  -0.400  0.68911
## poly(PCH_SUPERC_16_20, 5, raw = TRUE)1   2.030e-02  4.233e-02   0.480  0.63179
## poly(PCH_SUPERC_16_20, 5, raw = TRUE)2  -1.880e-04  2.839e-04  -0.662  0.50831
## poly(PCH_SUPERC_16_20, 5, raw = TRUE)3   4.236e-07  6.283e-07   0.674  0.50057
## poly(PCH_SUPERC_16_20, 5, raw = TRUE)4  -2.937e-10  5.024e-10  -0.585  0.55906
## poly(PCH_SUPERC_16_20, 5, raw = TRUE)5   6.269e-14  1.309e-13   0.479  0.63217
## poly(PCH_CONVS_16_20, 5, raw = TRUE)1    4.488e-02  1.083e-01   0.414  0.67874
## poly(PCH_CONVS_16_20, 5, raw = TRUE)2    5.029e-03  4.940e-03   1.018  0.30916
## poly(PCH_CONVS_16_20, 5, raw = TRUE)3   -4.829e-04  2.604e-04  -1.854  0.06439
## poly(PCH_CONVS_16_20, 5, raw = TRUE)4    5.711e-06  3.374e-06   1.693  0.09123
## poly(PCH_CONVS_16_20, 5, raw = TRUE)5   -1.803e-08  1.151e-08  -1.567  0.11784
## poly(PCH_SPECS_16_20, 5, raw = TRUE)1    3.810e-02  4.912e-02   0.776  0.43832
## poly(PCH_SPECS_16_20, 5, raw = TRUE)2   -7.805e-04  9.236e-04  -0.845  0.39856
## poly(PCH_SPECS_16_20, 5, raw = TRUE)3   -1.297e-05  1.956e-05  -0.663  0.50772
## poly(PCH_SPECS_16_20, 5, raw = TRUE)4    1.316e-07  1.900e-07   0.693  0.48893
## poly(PCH_SPECS_16_20, 5, raw = TRUE)5   -2.677e-10  4.258e-10  -0.629  0.52976
## poly(PCH_SNAPS_17_23, 5, raw = TRUE)1    2.544e-02  2.367e-01   0.107  0.91448
## poly(PCH_SNAPS_17_23, 5, raw = TRUE)2    1.400e-02  1.438e-02   0.973  0.33086
## poly(PCH_SNAPS_17_23, 5, raw = TRUE)3   -1.345e-03  1.146e-03  -1.173  0.24136
## poly(PCH_SNAPS_17_23, 5, raw = TRUE)4    4.148e-05  4.026e-05   1.030  0.30346
```

```
## poly(PCH_SNAPS_17_23, 5, raw = TRUE)5    -4.318e-07  4.091e-07  -1.056  0.29176
## poly(PCH_WICS_16_22, 5, raw = TRUE)1     -7.756e-02  6.227e-02  -1.245  0.21364
## poly(PCH_WICS_16_22, 5, raw = TRUE)2      2.156e-03  1.845e-03   1.168  0.24333
## poly(PCH_WICS_16_22, 5, raw = TRUE)3      4.513e-05  2.887e-05   1.563  0.11877
## poly(PCH_WICS_16_22, 5, raw = TRUE)4      2.429e-07  2.765e-07   0.878  0.38015
## poly(PCH_WICS_16_22, 5, raw = TRUE)5     -3.504e-09  2.149e-09  -1.631  0.10367
## poly(PCH_FFR_16_20, 5, raw = TRUE)1      -2.999e-01  1.855e-01  -1.617  0.10655
## poly(PCH_FFR_16_20, 5, raw = TRUE)2       5.326e-03  1.446e-02   0.368  0.71282
## poly(PCH_FFR_16_20, 5, raw = TRUE)3       6.549e-04  6.253e-04   1.047  0.29544
## poly(PCH_FFR_16_20, 5, raw = TRUE)4      -3.799e-05  3.408e-05  -1.115  0.26557
## poly(PCH_FFR_16_20, 5, raw = TRUE)5       4.082e-07  4.493e-07   0.909  0.36407
## poly(PCH_FSR_16_20, 5, raw = TRUE)1       1.475e-01  1.623e-01   0.909  0.36396
## poly(PCH_FSR_16_20, 5, raw = TRUE)2      -1.572e-02  1.909e-02  -0.823  0.41071
## poly(PCH_FSR_16_20, 5, raw = TRUE)3       6.073e-04  7.558e-04   0.804  0.42207
## poly(PCH_FSR_16_20, 5, raw = TRUE)4      -2.245e-05  4.526e-05  -0.496  0.62014
## poly(PCH_FSR_16_20, 5, raw = TRUE)5       1.553e-09  1.090e-06   0.001  0.99886
## poly(PCH_DIRSALES_12_17, 5, raw = TRUE)1 -2.105e-02  1.681e-02  -1.252  0.21126
## poly(PCH_DIRSALES_12_17, 5, raw = TRUE)2  9.444e-05  1.042e-04   0.906  0.36517
## poly(PCH_DIRSALES_12_17, 5, raw = TRUE)3 -1.558e-07  2.093e-07  -0.744  0.45701
## poly(PCH_DIRSALES_12_17, 5, raw = TRUE)4  9.903e-11  1.591e-10   0.623  0.53392
## poly(PCH_DIRSALES_12_17, 5, raw = TRUE)5 -2.067e-14  4.003e-14  -0.516  0.60591
##
## (Intercept)
## poly(PCH_GROC_16_20, 5, raw = TRUE)1     **
## poly(PCH_GROC_16_20, 5, raw = TRUE)2
## poly(PCH_GROC_16_20, 5, raw = TRUE)3     .
## poly(PCH_GROC_16_20, 5, raw = TRUE)4
## poly(PCH_GROC_16_20, 5, raw = TRUE)5
## poly(PCH_SUPERC_16_20, 5, raw = TRUE)1
## poly(PCH_SUPERC_16_20, 5, raw = TRUE)2
## poly(PCH_SUPERC_16_20, 5, raw = TRUE)3
## poly(PCH_SUPERC_16_20, 5, raw = TRUE)4
## poly(PCH_SUPERC_16_20, 5, raw = TRUE)5
## poly(PCH_CONVS_16_20, 5, raw = TRUE)1
## poly(PCH_CONVS_16_20, 5, raw = TRUE)2
## poly(PCH_CONVS_16_20, 5, raw = TRUE)3    .
## poly(PCH_CONVS_16_20, 5, raw = TRUE)4    .
## poly(PCH_CONVS_16_20, 5, raw = TRUE)5
## poly(PCH_SPECS_16_20, 5, raw = TRUE)1
## poly(PCH_SPECS_16_20, 5, raw = TRUE)2
## poly(PCH_SPECS_16_20, 5, raw = TRUE)3
## poly(PCH_SPECS_16_20, 5, raw = TRUE)4
## poly(PCH_SPECS_16_20, 5, raw = TRUE)5
## poly(PCH_SNAPS_17_23, 5, raw = TRUE)1
## poly(PCH_SNAPS_17_23, 5, raw = TRUE)2
## poly(PCH_SNAPS_17_23, 5, raw = TRUE)3
## poly(PCH_SNAPS_17_23, 5, raw = TRUE)4
## poly(PCH_SNAPS_17_23, 5, raw = TRUE)5
## poly(PCH_WICS_16_22, 5, raw = TRUE)1
## poly(PCH_WICS_16_22, 5, raw = TRUE)2
## poly(PCH_WICS_16_22, 5, raw = TRUE)3
## poly(PCH_WICS_16_22, 5, raw = TRUE)4
## poly(PCH_WICS_16_22, 5, raw = TRUE)5
## poly(PCH_FFR_16_20, 5, raw = TRUE)1
```

```
## poly(PCH_FFR_16_20, 5, raw = TRUE)2
## poly(PCH_FFR_16_20, 5, raw = TRUE)3
## poly(PCH_FFR_16_20, 5, raw = TRUE)4
## poly(PCH_FFR_16_20, 5, raw = TRUE)5
## poly(PCH_FSR_16_20, 5, raw = TRUE)1
## poly(PCH_FSR_16_20, 5, raw = TRUE)2
## poly(PCH_FSR_16_20, 5, raw = TRUE)3
## poly(PCH_FSR_16_20, 5, raw = TRUE)4
## poly(PCH_FSR_16_20, 5, raw = TRUE)5
## poly(PCH_DIRSALES_12_17, 5, raw = TRUE)1
## poly(PCH_DIRSALES_12_17, 5, raw = TRUE)2
## poly(PCH_DIRSALES_12_17, 5, raw = TRUE)3
## poly(PCH_DIRSALES_12_17, 5, raw = TRUE)4
## poly(PCH_DIRSALES_12_17, 5, raw = TRUE)5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.07 on 449 degrees of freedom
## Multiple R-squared:  0.158,  Adjusted R-squared:  0.07362
## F-statistic: 1.872 on 45 and 449 DF,  p-value: 0.0008303
```

```r
fitdeg_6 <- lm(PCH_LACCESS_POP_15_19 ~ poly(PCH_GROC_16_20, 6, raw=TRUE) + poly(PCH_SUPERC_16_20, 6, ra
predictions_6 <- predict(fitdeg_6, newdata = dat.valid)
mean((dat.valid$PCH_LACCESS_POP_15_19 - predictions_6)^2)
```

```
## [1] 8.941643e+15
```

```r
AIC(fitdeg_6)
```

```
## [1] 4141.521
```

```r
BIC(fitdeg_6)
```

```
## [1] 4376.976
```

```r
summary(fitdeg_6)
```

```
##
## Call:
## lm(formula = PCH_LACCESS_POP_15_19 ~ poly(PCH_GROC_16_20, 6,
##     raw = TRUE) + poly(PCH_SUPERC_16_20, 6, raw = TRUE) + poly(PCH_CONVS_16_20,
##     6, raw = TRUE) + poly(PCH_SPECS_16_20, 6, raw = TRUE) + poly(PCH_SNAPS_17_23,
##     6, raw = TRUE) + poly(PCH_WICS_16_22, 6, raw = TRUE) + poly(PCH_FFR_16_20,
##     6, raw = TRUE) + poly(PCH_FSR_16_20, 6, raw = TRUE) + poly(PCH_DIRSALES_12_17,
##     6, raw = TRUE), data = dat.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -48.052  -6.644  -0.595   5.523 129.379
##
## Coefficients:
##                                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              6.283e-01  2.675e+00   0.235   0.8144
## poly(PCH_GROC_16_20, 6, raw = TRUE)1    -1.083e-01  1.074e-01  -1.008   0.3139
## poly(PCH_GROC_16_20, 6, raw = TRUE)2     5.362e-03  5.030e-03   1.066   0.2870
## poly(PCH_GROC_16_20, 6, raw = TRUE)3    -3.418e-04  2.424e-04  -1.410   0.1593
## poly(PCH_GROC_16_20, 6, raw = TRUE)4    -1.212e-06  2.660e-06  -0.456   0.6489
```

```
## poly(PCH_GROC_16_20, 6, raw = TRUE)5      1.966e-07  1.040e-07   1.890   0.0594
## poly(PCH_GROC_16_20, 6, raw = TRUE)6     -1.553e-09  7.698e-10  -2.017   0.0443
## poly(PCH_SUPERC_16_20, 6, raw = TRUE)1    5.957e-03  6.195e-02   0.096   0.9234
## poly(PCH_SUPERC_16_20, 6, raw = TRUE)2    1.607e-04  7.514e-04   0.214   0.8307
## poly(PCH_SUPERC_16_20, 6, raw = TRUE)3   -1.308e-06  3.272e-06  -0.400   0.6894
## poly(PCH_SUPERC_16_20, 6, raw = TRUE)4    2.735e-09  5.555e-09   0.492   0.6228
## poly(PCH_SUPERC_16_20, 6, raw = TRUE)5   -2.000e-12  3.783e-12  -0.529   0.5974
## poly(PCH_SUPERC_16_20, 6, raw = TRUE)6    4.765e-16  8.819e-16   0.540   0.5893
## poly(PCH_CONVS_16_20, 6, raw = TRUE)1    -2.458e-02  1.230e-01  -0.200   0.8416
## poly(PCH_CONVS_16_20, 6, raw = TRUE)2     1.056e-02  8.833e-03   1.196   0.2324
## poly(PCH_CONVS_16_20, 6, raw = TRUE)3    -3.287e-04  3.127e-04  -1.051   0.2937
## poly(PCH_CONVS_16_20, 6, raw = TRUE)4    -4.331e-06  1.393e-05  -0.311   0.7561
## poly(PCH_CONVS_16_20, 6, raw = TRUE)5     9.429e-08  1.582e-07   0.596   0.5515
## poly(PCH_CONVS_16_20, 6, raw = TRUE)6    -3.522e-10  5.053e-10  -0.697   0.4861
## poly(PCH_SPECS_16_20, 6, raw = TRUE)1     2.612e-02  4.981e-02   0.524   0.6002
## poly(PCH_SPECS_16_20, 6, raw = TRUE)2    -1.752e-03  1.682e-03  -1.041   0.2983
## poly(PCH_SPECS_16_20, 6, raw = TRUE)3    -1.328e-06  2.061e-05  -0.064   0.9487
## poly(PCH_SPECS_16_20, 6, raw = TRUE)4     3.328e-07  4.209e-07   0.791   0.4296
## poly(PCH_SPECS_16_20, 6, raw = TRUE)5    -2.558e-09  3.342e-09  -0.765   0.4444
## poly(PCH_SPECS_16_20, 6, raw = TRUE)6     5.096e-12  6.867e-12   0.742   0.4584
## poly(PCH_SNAPS_17_23, 6, raw = TRUE)1    -2.676e-01  3.183e-01  -0.841   0.4009
## poly(PCH_SNAPS_17_23, 6, raw = TRUE)2     4.432e-02  2.628e-02   1.687   0.0924
## poly(PCH_SNAPS_17_23, 6, raw = TRUE)3    -6.389e-04  1.256e-03  -0.509   0.6112
## poly(PCH_SNAPS_17_23, 6, raw = TRUE)4    -8.808e-05  1.019e-04  -0.864   0.3879
## poly(PCH_SNAPS_17_23, 6, raw = TRUE)5     3.216e-06  2.663e-06   1.208   0.2278
## poly(PCH_SNAPS_17_23, 6, raw = TRUE)6    -3.031e-08  2.174e-08  -1.394   0.1641
## poly(PCH_WICS_16_22, 6, raw = TRUE)1     -9.719e-02  6.758e-02  -1.438   0.1511
## poly(PCH_WICS_16_22, 6, raw = TRUE)2      3.016e-03  2.637e-03   1.144   0.2533
## poly(PCH_WICS_16_22, 6, raw = TRUE)3      7.325e-05  5.333e-05   1.373   0.1703
## poly(PCH_WICS_16_22, 6, raw = TRUE)4     -7.039e-08  6.759e-07  -0.104   0.9171
## poly(PCH_WICS_16_22, 6, raw = TRUE)5     -6.841e-09  6.345e-09  -1.078   0.2816
## poly(PCH_WICS_16_22, 6, raw = TRUE)6      2.420e-11  4.613e-11   0.525   0.6001
## poly(PCH_FFR_16_20, 6, raw = TRUE)1      -4.122e-01  1.933e-01  -2.132   0.0335
## poly(PCH_FFR_16_20, 6, raw = TRUE)2      -1.426e-02  2.156e-02  -0.661   0.5089
## poly(PCH_FFR_16_20, 6, raw = TRUE)3       2.226e-03  1.118e-03   1.992   0.0470
## poly(PCH_FFR_16_20, 6, raw = TRUE)4      -1.965e-07  4.538e-05  -0.004   0.9965
## poly(PCH_FFR_16_20, 6, raw = TRUE)5      -2.825e-06  2.103e-06  -1.343   0.1799
## poly(PCH_FFR_16_20, 6, raw = TRUE)6       3.928e-08  2.388e-08   1.645   0.1007
## poly(PCH_FSR_16_20, 6, raw = TRUE)1       1.021e-01  2.189e-01   0.466   0.6412
## poly(PCH_FSR_16_20, 6, raw = TRUE)2      -1.328e-02  1.948e-02  -0.681   0.4959
## poly(PCH_FSR_16_20, 6, raw = TRUE)3       1.303e-03  2.036e-03   0.640   0.5225
## poly(PCH_FSR_16_20, 6, raw = TRUE)4      -4.953e-05  5.292e-05  -0.936   0.3498
## poly(PCH_FSR_16_20, 6, raw = TRUE)5      -1.393e-06  3.587e-06  -0.388   0.6979
## poly(PCH_FSR_16_20, 6, raw = TRUE)6       4.658e-08  7.599e-08   0.613   0.5403
## poly(PCH_DIRSALES_12_17, 6, raw = TRUE)1 -1.778e-02  1.849e-02  -0.962   0.3366
## poly(PCH_DIRSALES_12_17, 6, raw = TRUE)2  1.753e-05  1.724e-04   0.102   0.9190
## poly(PCH_DIRSALES_12_17, 6, raw = TRUE)3  1.250e-07  5.517e-07   0.227   0.8209
## poly(PCH_DIRSALES_12_17, 6, raw = TRUE)4 -2.635e-10  7.330e-10  -0.360   0.7194
## poly(PCH_DIRSALES_12_17, 6, raw = TRUE)5  1.735e-13  4.230e-13   0.410   0.6819
## poly(PCH_DIRSALES_12_17, 6, raw = TRUE)6 -3.675e-17  8.759e-17  -0.420   0.6750
##
## (Intercept)
## poly(PCH_GROC_16_20, 6, raw = TRUE)1
## poly(PCH_GROC_16_20, 6, raw = TRUE)2
```

```
## poly(PCH_GROC_16_20, 6, raw = TRUE)3
## poly(PCH_GROC_16_20, 6, raw = TRUE)4
## poly(PCH_GROC_16_20, 6, raw = TRUE)5     .
## poly(PCH_GROC_16_20, 6, raw = TRUE)6     *
## poly(PCH_SUPERC_16_20, 6, raw = TRUE)1
## poly(PCH_SUPERC_16_20, 6, raw = TRUE)2
## poly(PCH_SUPERC_16_20, 6, raw = TRUE)3
## poly(PCH_SUPERC_16_20, 6, raw = TRUE)4
## poly(PCH_SUPERC_16_20, 6, raw = TRUE)5
## poly(PCH_SUPERC_16_20, 6, raw = TRUE)6
## poly(PCH_CONVS_16_20, 6, raw = TRUE)1
## poly(PCH_CONVS_16_20, 6, raw = TRUE)2
## poly(PCH_CONVS_16_20, 6, raw = TRUE)3
## poly(PCH_CONVS_16_20, 6, raw = TRUE)4
## poly(PCH_CONVS_16_20, 6, raw = TRUE)5
## poly(PCH_CONVS_16_20, 6, raw = TRUE)6
## poly(PCH_SPECS_16_20, 6, raw = TRUE)1
## poly(PCH_SPECS_16_20, 6, raw = TRUE)2
## poly(PCH_SPECS_16_20, 6, raw = TRUE)3
## poly(PCH_SPECS_16_20, 6, raw = TRUE)4
## poly(PCH_SPECS_16_20, 6, raw = TRUE)5
## poly(PCH_SPECS_16_20, 6, raw = TRUE)6
## poly(PCH_SNAPS_17_23, 6, raw = TRUE)1
## poly(PCH_SNAPS_17_23, 6, raw = TRUE)2    .
## poly(PCH_SNAPS_17_23, 6, raw = TRUE)3
## poly(PCH_SNAPS_17_23, 6, raw = TRUE)4
## poly(PCH_SNAPS_17_23, 6, raw = TRUE)5
## poly(PCH_SNAPS_17_23, 6, raw = TRUE)6
## poly(PCH_WICS_16_22, 6, raw = TRUE)1
## poly(PCH_WICS_16_22, 6, raw = TRUE)2
## poly(PCH_WICS_16_22, 6, raw = TRUE)3
## poly(PCH_WICS_16_22, 6, raw = TRUE)4
## poly(PCH_WICS_16_22, 6, raw = TRUE)5
## poly(PCH_WICS_16_22, 6, raw = TRUE)6
## poly(PCH_FFR_16_20, 6, raw = TRUE)1      *
## poly(PCH_FFR_16_20, 6, raw = TRUE)2
## poly(PCH_FFR_16_20, 6, raw = TRUE)3      *
## poly(PCH_FFR_16_20, 6, raw = TRUE)4
## poly(PCH_FFR_16_20, 6, raw = TRUE)5
## poly(PCH_FFR_16_20, 6, raw = TRUE)6
## poly(PCH_FSR_16_20, 6, raw = TRUE)1
## poly(PCH_FSR_16_20, 6, raw = TRUE)2
## poly(PCH_FSR_16_20, 6, raw = TRUE)3
## poly(PCH_FSR_16_20, 6, raw = TRUE)4
## poly(PCH_FSR_16_20, 6, raw = TRUE)5
## poly(PCH_FSR_16_20, 6, raw = TRUE)6
## poly(PCH_DIRSALES_12_17, 6, raw = TRUE)1
## poly(PCH_DIRSALES_12_17, 6, raw = TRUE)2
## poly(PCH_DIRSALES_12_17, 6, raw = TRUE)3
## poly(PCH_DIRSALES_12_17, 6, raw = TRUE)4
## poly(PCH_DIRSALES_12_17, 6, raw = TRUE)5
## poly(PCH_DIRSALES_12_17, 6, raw = TRUE)6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 15.03 on 440 degrees of freedom
## Multiple R-squared:  0.179,  Adjusted R-squared:  0.0782
## F-statistic: 1.776 on 54 and 440 DF,  p-value: 0.001027
```

```r
fitdeg_7 <- lm(PCH_LACCESS_POP_15_19 ~ poly(PCH_GROC_16_20, 7, raw=TRUE) + poly(PCH_SUPERC_16_20, 7, ra
predictions_7 <- predict(fitdeg_7, newdata = dat.valid)
mean((dat.valid$PCH_LACCESS_POP_15_19 - predictions_7)^2)
```

```
## [1] 1.346487e+18
```

```r
AIC(fitdeg_7)
```

```
## [1] 4154.316
```

```r
BIC(fitdeg_7)
```

```
## [1] 4427.612
```

```r
summary(fitdeg_7)
```

```
##
## Call:
## lm(formula = PCH_LACCESS_POP_15_19 ~ poly(PCH_GROC_16_20, 7,
##     raw = TRUE) + poly(PCH_SUPERC_16_20, 7, raw = TRUE) + poly(PCH_CONVS_16_20,
##     7, raw = TRUE) + poly(PCH_SPECS_16_20, 7, raw = TRUE) + poly(PCH_SNAPS_17_23,
##     7, raw = TRUE) + poly(PCH_WICS_16_22, 7, raw = TRUE) + poly(PCH_FFR_16_20,
##     7, raw = TRUE) + poly(PCH_FSR_16_20, 7, raw = TRUE) + poly(PCH_DIRSALES_12_17,
##     7, raw = TRUE), data = dat.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -45.452  -6.439  -0.533   5.117 130.239
##
## Coefficients:
##                                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              -2.090e-01  2.849e+00  -0.073   0.9415
## poly(PCH_GROC_16_20, 7, raw = TRUE)1     -1.108e-01  1.135e-01  -0.976   0.3297
## poly(PCH_GROC_16_20, 7, raw = TRUE)2      7.139e-03  8.897e-03   0.802   0.4227
## poly(PCH_GROC_16_20, 7, raw = TRUE)3     -3.046e-04  2.776e-04  -1.097   0.2731
## poly(PCH_GROC_16_20, 7, raw = TRUE)4     -4.391e-06  1.225e-05  -0.359   0.7201
## poly(PCH_GROC_16_20, 7, raw = TRUE)5      1.849e-07  1.086e-07   1.703   0.0894
## poly(PCH_GROC_16_20, 7, raw = TRUE)6     -2.845e-10  4.393e-09  -0.065   0.9484
## poly(PCH_GROC_16_20, 7, raw = TRUE)7     -8.897e-12  3.063e-11  -0.290   0.7716
## poly(PCH_SUPERC_16_20, 7, raw = TRUE)1   -8.406e-03  7.366e-02  -0.114   0.9092
## poly(PCH_SUPERC_16_20, 7, raw = TRUE)2    5.352e-04  1.274e-03   0.420   0.6746
## poly(PCH_SUPERC_16_20, 7, raw = TRUE)3   -4.109e-06  8.741e-06  -0.470   0.6386
## poly(PCH_SUPERC_16_20, 7, raw = TRUE)4    1.142e-08  2.683e-08   0.426   0.6707
## poly(PCH_SUPERC_16_20, 7, raw = TRUE)5   -1.408e-11  3.790e-11  -0.372   0.7104
## poly(PCH_SUPERC_16_20, 7, raw = TRUE)6    7.795e-15  2.346e-14   0.332   0.7399
## poly(PCH_SUPERC_16_20, 7, raw = TRUE)7   -1.581e-18  5.181e-18  -0.305   0.7604
## poly(PCH_CONVS_16_20, 7, raw = TRUE)1     5.284e-02  1.441e-01   0.367   0.7141
## poly(PCH_CONVS_16_20, 7, raw = TRUE)2     1.444e-02  1.262e-02   1.144   0.2532
## poly(PCH_CONVS_16_20, 7, raw = TRUE)3    -8.347e-04  6.595e-04  -1.266   0.2063
## poly(PCH_CONVS_16_20, 7, raw = TRUE)4    -9.719e-06  1.970e-05  -0.493   0.6219
## poly(PCH_CONVS_16_20, 7, raw = TRUE)5     6.565e-07  8.254e-07   0.795   0.4269
## poly(PCH_CONVS_16_20, 7, raw = TRUE)6    -6.874e-09  8.797e-09  -0.781   0.4350
```

```
## poly(PCH_CONVS_16_20, 7, raw = TRUE)7     2.068e-11  2.713e-11   0.762   0.4464
## poly(PCH_SPECS_16_20, 7, raw = TRUE)1    -3.335e-02  6.556e-02  -0.509   0.6112
## poly(PCH_SPECS_16_20, 7, raw = TRUE)2    -3.109e-03  1.899e-03  -1.637   0.1023
## poly(PCH_SPECS_16_20, 7, raw = TRUE)3     6.333e-05  5.002e-05   1.266   0.2061
## poly(PCH_SPECS_16_20, 7, raw = TRUE)4     4.750e-07  4.334e-07   1.096   0.2737
## poly(PCH_SPECS_16_20, 7, raw = TRUE)5    -1.549e-08  9.619e-09  -1.611   0.1079
## poly(PCH_SPECS_16_20, 7, raw = TRUE)6     9.313e-11  6.214e-11   1.499   0.1347
## poly(PCH_SPECS_16_20, 7, raw = TRUE)7    -1.623e-13  1.141e-13  -1.422   0.1557
## poly(PCH_SNAPS_17_23, 7, raw = TRUE)1    -3.035e-01  3.294e-01  -0.921   0.3574
## poly(PCH_SNAPS_17_23, 7, raw = TRUE)2     5.783e-02  4.526e-02   1.278   0.2021
## poly(PCH_SNAPS_17_23, 7, raw = TRUE)3    -1.465e-03  2.155e-03  -0.680   0.4971
## poly(PCH_SNAPS_17_23, 7, raw = TRUE)4    -1.124e-04  1.413e-04  -0.796   0.4267
## poly(PCH_SNAPS_17_23, 7, raw = TRUE)5     6.079e-06  8.548e-06   0.711   0.4774
## poly(PCH_SNAPS_17_23, 7, raw = TRUE)6    -9.789e-08  1.795e-07  -0.545   0.5858
## poly(PCH_SNAPS_17_23, 7, raw = TRUE)7     4.911e-10  1.255e-09   0.391   0.6957
## poly(PCH_WICS_16_22, 7, raw = TRUE)1     -8.770e-02  9.535e-02  -0.920   0.3582
## poly(PCH_WICS_16_22, 7, raw = TRUE)2      3.270e-03  3.030e-03   1.079   0.2810
## poly(PCH_WICS_16_22, 7, raw = TRUE)3      6.347e-05  1.117e-04   0.568   0.5701
## poly(PCH_WICS_16_22, 7, raw = TRUE)4     -2.563e-07  1.437e-06  -0.178   0.8585
## poly(PCH_WICS_16_22, 7, raw = TRUE)5     -4.251e-09  2.192e-08  -0.194   0.8463
## poly(PCH_WICS_16_22, 7, raw = TRUE)6      4.277e-11  1.467e-10   0.292   0.7708
## poly(PCH_WICS_16_22, 7, raw = TRUE)7     -1.543e-13  1.203e-12  -0.128   0.8980
## poly(PCH_FFR_16_20, 7, raw = TRUE)1      -3.063e-01  2.676e-01  -1.144   0.2531
## poly(PCH_FFR_16_20, 7, raw = TRUE)2      -1.584e-02  2.204e-02  -0.719   0.4728
## poly(PCH_FFR_16_20, 7, raw = TRUE)3       1.394e-03  2.419e-03   0.576   0.5647
## poly(PCH_FFR_16_20, 7, raw = TRUE)4       2.692e-05  8.099e-05   0.332   0.7397
## poly(PCH_FFR_16_20, 7, raw = TRUE)5      -1.643e-06  3.801e-06  -0.432   0.6658
## poly(PCH_FFR_16_20, 7, raw = TRUE)6      -1.727e-08  1.483e-07  -0.116   0.9073
## poly(PCH_FFR_16_20, 7, raw = TRUE)7       5.652e-10  1.449e-09   0.390   0.6966
## poly(PCH_FSR_16_20, 7, raw = TRUE)1      -5.161e-03  2.368e-01  -0.022   0.9826
## poly(PCH_FSR_16_20, 7, raw = TRUE)2       1.845e-02  3.586e-02   0.514   0.6073
## poly(PCH_FSR_16_20, 7, raw = TRUE)3       2.372e-03  2.221e-03   1.068   0.2862
## poly(PCH_FSR_16_20, 7, raw = TRUE)4      -2.576e-04  2.134e-04  -1.207   0.2279
## poly(PCH_FSR_16_20, 7, raw = TRUE)5      -1.221e-06  4.247e-06  -0.288   0.7738
## poly(PCH_FSR_16_20, 7, raw = TRUE)6       3.363e-07  3.071e-07   1.095   0.2740
## poly(PCH_FSR_16_20, 7, raw = TRUE)7      -4.519e-09  5.919e-09  -0.763   0.4456
## poly(PCH_DIRSALES_12_17, 7, raw = TRUE)1 -1.865e-02  1.900e-02  -0.982   0.3267
## poly(PCH_DIRSALES_12_17, 7, raw = TRUE)2  5.988e-05  2.529e-04   0.237   0.8130
## poly(PCH_DIRSALES_12_17, 7, raw = TRUE)3 -1.005e-07  1.218e-06  -0.083   0.9342
## poly(PCH_DIRSALES_12_17, 7, raw = TRUE)4  1.814e-10  2.502e-09   0.073   0.9422
## poly(PCH_DIRSALES_12_17, 7, raw = TRUE)5 -2.319e-13  2.462e-12  -0.094   0.9250
## poly(PCH_DIRSALES_12_17, 7, raw = TRUE)6  1.366e-16  1.146e-15   0.119   0.9052
## poly(PCH_DIRSALES_12_17, 7, raw = TRUE)7 -2.818e-20  2.025e-19  -0.139   0.8894
##
## (Intercept)
## poly(PCH_GROC_16_20, 7, raw = TRUE)1
## poly(PCH_GROC_16_20, 7, raw = TRUE)2
## poly(PCH_GROC_16_20, 7, raw = TRUE)3
## poly(PCH_GROC_16_20, 7, raw = TRUE)4
## poly(PCH_GROC_16_20, 7, raw = TRUE)5     .
## poly(PCH_GROC_16_20, 7, raw = TRUE)6
## poly(PCH_GROC_16_20, 7, raw = TRUE)7
## poly(PCH_SUPERC_16_20, 7, raw = TRUE)1
## poly(PCH_SUPERC_16_20, 7, raw = TRUE)2
```

```
## poly(PCH_SUPERC_16_20, 7, raw = TRUE)3
## poly(PCH_SUPERC_16_20, 7, raw = TRUE)4
## poly(PCH_SUPERC_16_20, 7, raw = TRUE)5
## poly(PCH_SUPERC_16_20, 7, raw = TRUE)6
## poly(PCH_SUPERC_16_20, 7, raw = TRUE)7
## poly(PCH_CONVS_16_20, 7, raw = TRUE)1
## poly(PCH_CONVS_16_20, 7, raw = TRUE)2
## poly(PCH_CONVS_16_20, 7, raw = TRUE)3
## poly(PCH_CONVS_16_20, 7, raw = TRUE)4
## poly(PCH_CONVS_16_20, 7, raw = TRUE)5
## poly(PCH_CONVS_16_20, 7, raw = TRUE)6
## poly(PCH_CONVS_16_20, 7, raw = TRUE)7
## poly(PCH_SPECS_16_20, 7, raw = TRUE)1
## poly(PCH_SPECS_16_20, 7, raw = TRUE)2
## poly(PCH_SPECS_16_20, 7, raw = TRUE)3
## poly(PCH_SPECS_16_20, 7, raw = TRUE)4
## poly(PCH_SPECS_16_20, 7, raw = TRUE)5
## poly(PCH_SPECS_16_20, 7, raw = TRUE)6
## poly(PCH_SPECS_16_20, 7, raw = TRUE)7
## poly(PCH_SNAPS_17_23, 7, raw = TRUE)1
## poly(PCH_SNAPS_17_23, 7, raw = TRUE)2
## poly(PCH_SNAPS_17_23, 7, raw = TRUE)3
## poly(PCH_SNAPS_17_23, 7, raw = TRUE)4
## poly(PCH_SNAPS_17_23, 7, raw = TRUE)5
## poly(PCH_SNAPS_17_23, 7, raw = TRUE)6
## poly(PCH_SNAPS_17_23, 7, raw = TRUE)7
## poly(PCH_WICS_16_22, 7, raw = TRUE)1
## poly(PCH_WICS_16_22, 7, raw = TRUE)2
## poly(PCH_WICS_16_22, 7, raw = TRUE)3
## poly(PCH_WICS_16_22, 7, raw = TRUE)4
## poly(PCH_WICS_16_22, 7, raw = TRUE)5
## poly(PCH_WICS_16_22, 7, raw = TRUE)6
## poly(PCH_WICS_16_22, 7, raw = TRUE)7
## poly(PCH_FFR_16_20, 7, raw = TRUE)1
## poly(PCH_FFR_16_20, 7, raw = TRUE)2
## poly(PCH_FFR_16_20, 7, raw = TRUE)3
## poly(PCH_FFR_16_20, 7, raw = TRUE)4
## poly(PCH_FFR_16_20, 7, raw = TRUE)5
## poly(PCH_FFR_16_20, 7, raw = TRUE)6
## poly(PCH_FFR_16_20, 7, raw = TRUE)7
## poly(PCH_FSR_16_20, 7, raw = TRUE)1
## poly(PCH_FSR_16_20, 7, raw = TRUE)2
## poly(PCH_FSR_16_20, 7, raw = TRUE)3
## poly(PCH_FSR_16_20, 7, raw = TRUE)4
## poly(PCH_FSR_16_20, 7, raw = TRUE)5
## poly(PCH_FSR_16_20, 7, raw = TRUE)6
## poly(PCH_FSR_16_20, 7, raw = TRUE)7
## poly(PCH_DIRSALES_12_17, 7, raw = TRUE)1
## poly(PCH_DIRSALES_12_17, 7, raw = TRUE)2
## poly(PCH_DIRSALES_12_17, 7, raw = TRUE)3
## poly(PCH_DIRSALES_12_17, 7, raw = TRUE)4
## poly(PCH_DIRSALES_12_17, 7, raw = TRUE)5
## poly(PCH_DIRSALES_12_17, 7, raw = TRUE)6
## poly(PCH_DIRSALES_12_17, 7, raw = TRUE)7
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.11 on 431 degrees of freedom
## Multiple R-squared:  0.1876, Adjusted R-squared:  0.0688
## F-statistic: 1.579 on 63 and 431 DF,  p-value: 0.004977
```

```
fitdeg_8 <- lm(PCH_LACCESS_POP_15_19 ~ poly(PCH_GROC_16_20, 8, raw=TRUE) + poly(PCH_SUPERC_16_20, 8, ra
predictions_8 <- predict(fitdeg_8, newdata = dat.valid)
mean((dat.valid$PCH_LACCESS_POP_15_19 - predictions_8)^2)
```

```
## [1] 8.562843e+23
```

```
AIC(fitdeg_8)
```

```
## [1] 4155.66
```

```
BIC(fitdeg_8)
```

```
## [1] 4466.797
```

```
summary(fitdeg_8)
```

```
##
## Call:
## lm(formula = PCH_LACCESS_POP_15_19 ~ poly(PCH_GROC_16_20, 8,
##     raw = TRUE) + poly(PCH_SUPERC_16_20, 8, raw = TRUE) + poly(PCH_CONVS_16_20,
##     8, raw = TRUE) + poly(PCH_SPECS_16_20, 8, raw = TRUE) + poly(PCH_SNAPS_17_23,
##     8, raw = TRUE) + poly(PCH_WICS_16_22, 8, raw = TRUE) + poly(PCH_FFR_16_20,
##     8, raw = TRUE) + poly(PCH_FSR_16_20, 8, raw = TRUE) + poly(PCH_DIRSALES_12_17,
##     8, raw = TRUE), data = dat.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -41.318  -6.742  -0.623   5.385 123.907
##
## Coefficients:
##                                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                              -2.616e+00  3.021e+00  -0.866   0.3870
## poly(PCH_GROC_16_20, 8, raw = TRUE)1     -1.961e-01  1.471e-01  -1.333   0.1833
## poly(PCH_GROC_16_20, 8, raw = TRUE)2      1.418e-03  9.778e-03   0.145   0.8848
## poly(PCH_GROC_16_20, 8, raw = TRUE)3      3.268e-04  5.924e-04   0.552   0.5814
## poly(PCH_GROC_16_20, 8, raw = TRUE)4      7.635e-06  1.452e-05   0.526   0.5993
## poly(PCH_GROC_16_20, 8, raw = TRUE)5     -5.996e-07  6.084e-07  -0.986   0.3249
## poly(PCH_GROC_16_20, 8, raw = TRUE)6     -2.410e-09  4.514e-09  -0.534   0.5936
## poly(PCH_GROC_16_20, 8, raw = TRUE)7      2.457e-10  1.875e-10   1.310   0.1908
## poly(PCH_GROC_16_20, 8, raw = TRUE)8     -1.726e-12  1.273e-12  -1.355   0.1761
## poly(PCH_SUPERC_16_20, 8, raw = TRUE)1   -2.745e-02  8.456e-02  -0.325   0.7457
## poly(PCH_SUPERC_16_20, 8, raw = TRUE)2    1.381e-03  2.006e-03   0.688   0.4917
## poly(PCH_SUPERC_16_20, 8, raw = TRUE)3   -1.363e-05  1.916e-05  -0.711   0.4772
## poly(PCH_SUPERC_16_20, 8, raw = TRUE)4    5.617e-08  8.435e-08   0.666   0.5058
## poly(PCH_SUPERC_16_20, 8, raw = TRUE)5   -1.162e-10  1.864e-10  -0.623   0.5333
## poly(PCH_SUPERC_16_20, 8, raw = TRUE)6    1.247e-13  2.102e-13   0.593   0.5535
## poly(PCH_SUPERC_16_20, 8, raw = TRUE)7   -6.477e-17  1.131e-16  -0.573   0.5671
## poly(PCH_SUPERC_16_20, 8, raw = TRUE)8    1.278e-20  2.284e-20   0.560   0.5761
## poly(PCH_CONVS_16_20, 8, raw = TRUE)1     5.892e-02  1.904e-01   0.310   0.7571
## poly(PCH_CONVS_16_20, 8, raw = TRUE)2     1.961e-02  1.450e-02   1.353   0.1768
```

```
## poly(PCH_CONVS_16_20, 8, raw = TRUE)3    -1.120e-03  1.176e-03  -0.952   0.3417
## poly(PCH_CONVS_16_20, 8, raw = TRUE)4    -1.695e-05  4.273e-05  -0.397   0.6917
## poly(PCH_CONVS_16_20, 8, raw = TRUE)5     1.243e-06  1.309e-06   0.950   0.3428
## poly(PCH_CONVS_16_20, 8, raw = TRUE)6    -1.727e-08  4.827e-08  -0.358   0.7207
## poly(PCH_CONVS_16_20, 8, raw = TRUE)7     9.278e-11  4.826e-10   0.192   0.8477
## poly(PCH_CONVS_16_20, 8, raw = TRUE)8    -1.737e-13  1.441e-12  -0.121   0.9041
## poly(PCH_SPECS_16_20, 8, raw = TRUE)1    -3.956e-02  7.513e-02  -0.527   0.5988
## poly(PCH_SPECS_16_20, 8, raw = TRUE)2    -2.605e-03  2.515e-03  -1.036   0.3010
## poly(PCH_SPECS_16_20, 8, raw = TRUE)3     8.108e-05  6.900e-05   1.175   0.2406
## poly(PCH_SPECS_16_20, 8, raw = TRUE)4    -7.932e-08  1.277e-06  -0.062   0.9505
## poly(PCH_SPECS_16_20, 8, raw = TRUE)5    -1.765e-08  1.089e-08  -1.620   0.1060
## poly(PCH_SPECS_16_20, 8, raw = TRUE)6     1.980e-10  2.181e-10   0.908   0.3643
## poly(PCH_SPECS_16_20, 8, raw = TRUE)7    -8.119e-13  1.242e-12  -0.654   0.5135
## poly(PCH_SPECS_16_20, 8, raw = TRUE)8     1.140e-15  2.114e-15   0.539   0.5901
## poly(PCH_SNAPS_17_23, 8, raw = TRUE)1    -1.445e-02  3.931e-01  -0.037   0.9707
## poly(PCH_SNAPS_17_23, 8, raw = TRUE)2     8.830e-02  4.663e-02   1.894   0.0589
## poly(PCH_SNAPS_17_23, 8, raw = TRUE)3    -9.124e-03  4.577e-03  -1.993   0.0469
## poly(PCH_SNAPS_17_23, 8, raw = TRUE)4     7.100e-05  1.804e-04   0.393   0.6942
## poly(PCH_SNAPS_17_23, 8, raw = TRUE)5     2.626e-05  1.286e-05   2.043   0.0417
## poly(PCH_SNAPS_17_23, 8, raw = TRUE)6    -1.271e-06  6.093e-07  -2.086   0.0376
## poly(PCH_SNAPS_17_23, 8, raw = TRUE)7     2.251e-08  1.106e-08   2.035   0.0425
## poly(PCH_SNAPS_17_23, 8, raw = TRUE)8    -1.397e-10  6.978e-11  -2.003   0.0459
## poly(PCH_WICS_16_22, 8, raw = TRUE)1     -8.698e-02  9.680e-02  -0.899   0.3694
## poly(PCH_WICS_16_22, 8, raw = TRUE)2      1.096e-02  5.750e-03   1.906   0.0574
## poly(PCH_WICS_16_22, 8, raw = TRUE)3      1.412e-04  1.155e-04   1.222   0.2222
## poly(PCH_WICS_16_22, 8, raw = TRUE)4     -7.295e-06  4.465e-06  -1.634   0.1031
## poly(PCH_WICS_16_22, 8, raw = TRUE)5     -5.956e-08  3.617e-08  -1.647   0.1004
## poly(PCH_WICS_16_22, 8, raw = TRUE)6      1.287e-09  7.422e-10   1.734   0.0836
## poly(PCH_WICS_16_22, 8, raw = TRUE)7      5.585e-12  3.361e-12   1.661   0.0974
## poly(PCH_WICS_16_22, 8, raw = TRUE)8     -5.912e-14  3.378e-14  -1.750   0.0808
## poly(PCH_FFR_16_20, 8, raw = TRUE)1      -3.110e-01  2.817e-01  -1.104   0.2703
## poly(PCH_FFR_16_20, 8, raw = TRUE)2       1.835e-03  3.853e-02   0.048   0.9620
## poly(PCH_FFR_16_20, 8, raw = TRUE)3       1.231e-03  2.431e-03   0.506   0.6128
## poly(PCH_FFR_16_20, 8, raw = TRUE)4      -8.410e-05  2.366e-04  -0.355   0.7224
## poly(PCH_FFR_16_20, 8, raw = TRUE)5       1.197e-06  6.241e-06   0.192   0.8480
## poly(PCH_FFR_16_20, 8, raw = TRUE)6       1.160e-07  3.150e-07   0.368   0.7128
## poly(PCH_FFR_16_20, 8, raw = TRUE)7      -4.914e-09  1.094e-08  -0.449   0.6534
## poly(PCH_FFR_16_20, 8, raw = TRUE)8       5.029e-11  9.828e-11   0.512   0.6091
## poly(PCH_FSR_16_20, 8, raw = TRUE)1       2.907e-01  2.875e-01   1.011   0.3125
## poly(PCH_FSR_16_20, 8, raw = TRUE)2       7.404e-02  4.413e-02   1.678   0.0941
## poly(PCH_FSR_16_20, 8, raw = TRUE)3      -5.305e-03  4.743e-03  -1.118   0.2640
## poly(PCH_FSR_16_20, 8, raw = TRUE)4      -6.175e-04  2.685e-04  -2.300   0.0220
## poly(PCH_FSR_16_20, 8, raw = TRUE)5       4.122e-05  2.252e-05   1.830   0.0679
## poly(PCH_FSR_16_20, 8, raw = TRUE)6       4.187e-07  3.324e-07   1.260   0.2085
## poly(PCH_FSR_16_20, 8, raw = TRUE)7      -6.220e-08  2.955e-08  -2.105   0.0359
## poly(PCH_FSR_16_20, 8, raw = TRUE)8       9.327e-10  4.967e-10   1.878   0.0611
## poly(PCH_DIRSALES_12_17, 8, raw = TRUE)1 -2.254e-02  1.957e-02  -1.152   0.2500
## poly(PCH_DIRSALES_12_17, 8, raw = TRUE)2 -9.935e-05  3.206e-04  -0.310   0.7568
## poly(PCH_DIRSALES_12_17, 8, raw = TRUE)3  1.647e-06  2.304e-06   0.715   0.4752
## poly(PCH_DIRSALES_12_17, 8, raw = TRUE)4 -6.100e-09  7.119e-09  -0.857   0.3920
## poly(PCH_DIRSALES_12_17, 8, raw = TRUE)5  1.022e-11  1.096e-11   0.933   0.3514
## poly(PCH_DIRSALES_12_17, 8, raw = TRUE)6 -8.663e-15  8.810e-15  -0.983   0.3260
## poly(PCH_DIRSALES_12_17, 8, raw = TRUE)7  3.604e-18  3.533e-18   1.020   0.3083
## poly(PCH_DIRSALES_12_17, 8, raw = TRUE)8 -5.831e-22  5.567e-22  -1.047   0.2955
```

```
##
## (Intercept)
## poly(PCH_GROC_16_20, 8, raw = TRUE)1
## poly(PCH_GROC_16_20, 8, raw = TRUE)2
## poly(PCH_GROC_16_20, 8, raw = TRUE)3
## poly(PCH_GROC_16_20, 8, raw = TRUE)4
## poly(PCH_GROC_16_20, 8, raw = TRUE)5
## poly(PCH_GROC_16_20, 8, raw = TRUE)6
## poly(PCH_GROC_16_20, 8, raw = TRUE)7
## poly(PCH_GROC_16_20, 8, raw = TRUE)8
## poly(PCH_SUPERC_16_20, 8, raw = TRUE)1
## poly(PCH_SUPERC_16_20, 8, raw = TRUE)2
## poly(PCH_SUPERC_16_20, 8, raw = TRUE)3
## poly(PCH_SUPERC_16_20, 8, raw = TRUE)4
## poly(PCH_SUPERC_16_20, 8, raw = TRUE)5
## poly(PCH_SUPERC_16_20, 8, raw = TRUE)6
## poly(PCH_SUPERC_16_20, 8, raw = TRUE)7
## poly(PCH_SUPERC_16_20, 8, raw = TRUE)8
## poly(PCH_CONVS_16_20, 8, raw = TRUE)1
## poly(PCH_CONVS_16_20, 8, raw = TRUE)2
## poly(PCH_CONVS_16_20, 8, raw = TRUE)3
## poly(PCH_CONVS_16_20, 8, raw = TRUE)4
## poly(PCH_CONVS_16_20, 8, raw = TRUE)5
## poly(PCH_CONVS_16_20, 8, raw = TRUE)6
## poly(PCH_CONVS_16_20, 8, raw = TRUE)7
## poly(PCH_CONVS_16_20, 8, raw = TRUE)8
## poly(PCH_SPECS_16_20, 8, raw = TRUE)1
## poly(PCH_SPECS_16_20, 8, raw = TRUE)2
## poly(PCH_SPECS_16_20, 8, raw = TRUE)3
## poly(PCH_SPECS_16_20, 8, raw = TRUE)4
## poly(PCH_SPECS_16_20, 8, raw = TRUE)5
## poly(PCH_SPECS_16_20, 8, raw = TRUE)6
## poly(PCH_SPECS_16_20, 8, raw = TRUE)7
## poly(PCH_SPECS_16_20, 8, raw = TRUE)8
## poly(PCH_SNAPS_17_23, 8, raw = TRUE)1
## poly(PCH_SNAPS_17_23, 8, raw = TRUE)2    .
## poly(PCH_SNAPS_17_23, 8, raw = TRUE)3    *
## poly(PCH_SNAPS_17_23, 8, raw = TRUE)4
## poly(PCH_SNAPS_17_23, 8, raw = TRUE)5    *
## poly(PCH_SNAPS_17_23, 8, raw = TRUE)6    *
## poly(PCH_SNAPS_17_23, 8, raw = TRUE)7    *
## poly(PCH_SNAPS_17_23, 8, raw = TRUE)8    *
## poly(PCH_WICS_16_22, 8, raw = TRUE)1
## poly(PCH_WICS_16_22, 8, raw = TRUE)2     .
## poly(PCH_WICS_16_22, 8, raw = TRUE)3
## poly(PCH_WICS_16_22, 8, raw = TRUE)4
## poly(PCH_WICS_16_22, 8, raw = TRUE)5
## poly(PCH_WICS_16_22, 8, raw = TRUE)6     .
## poly(PCH_WICS_16_22, 8, raw = TRUE)7     .
## poly(PCH_WICS_16_22, 8, raw = TRUE)8     .
## poly(PCH_FFR_16_20, 8, raw = TRUE)1
## poly(PCH_FFR_16_20, 8, raw = TRUE)2
## poly(PCH_FFR_16_20, 8, raw = TRUE)3
## poly(PCH_FFR_16_20, 8, raw = TRUE)4
```

```
## poly(PCH_FFR_16_20, 8, raw = TRUE)5
## poly(PCH_FFR_16_20, 8, raw = TRUE)6
## poly(PCH_FFR_16_20, 8, raw = TRUE)7
## poly(PCH_FFR_16_20, 8, raw = TRUE)8
## poly(PCH_FSR_16_20, 8, raw = TRUE)1
## poly(PCH_FSR_16_20, 8, raw = TRUE)2      .
## poly(PCH_FSR_16_20, 8, raw = TRUE)3
## poly(PCH_FSR_16_20, 8, raw = TRUE)4      *
## poly(PCH_FSR_16_20, 8, raw = TRUE)5      .
## poly(PCH_FSR_16_20, 8, raw = TRUE)6
## poly(PCH_FSR_16_20, 8, raw = TRUE)7      *
## poly(PCH_FSR_16_20, 8, raw = TRUE)8      .
## poly(PCH_DIRSALES_12_17, 8, raw = TRUE)1
## poly(PCH_DIRSALES_12_17, 8, raw = TRUE)2
## poly(PCH_DIRSALES_12_17, 8, raw = TRUE)3
## poly(PCH_DIRSALES_12_17, 8, raw = TRUE)4
## poly(PCH_DIRSALES_12_17, 8, raw = TRUE)5
## poly(PCH_DIRSALES_12_17, 8, raw = TRUE)6
## poly(PCH_DIRSALES_12_17, 8, raw = TRUE)7
## poly(PCH_DIRSALES_12_17, 8, raw = TRUE)8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.01 on 422 degrees of freedom
## Multiple R-squared:  0.2144, Adjusted R-squared:  0.08041
## F-statistic:   1.6 on 72 and 422 DF,  p-value: 0.002632
```

```
fitdeg_9 <- lm(PCH_LACCESS_POP_15_19 ~ poly(PCH_GROC_16_20, 9, raw=TRUE) + poly(PCH_SUPERC_16_20, 9, ra
predictions_9 <- predict(fitdeg_9, newdata = dat.valid)
mean((dat.valid$PCH_LACCESS_POP_15_19 - predictions_9)^2)
```

```
## [1] 1.408751e+26
```

```
AIC(fitdeg_9)
```

```
## [1] 4156.01
```

```
BIC(fitdeg_9)
```

```
## [1] 4504.988
```

```
summary(fitdeg_9)
```

```
##
## Call:
## lm(formula = PCH_LACCESS_POP_15_19 ~ poly(PCH_GROC_16_20, 9,
##     raw = TRUE) + poly(PCH_SUPERC_16_20, 9, raw = TRUE) + poly(PCH_CONVS_16_20,
##     9, raw = TRUE) + poly(PCH_SPECS_16_20, 9, raw = TRUE) + poly(PCH_SNAPS_17_23,
##     9, raw = TRUE) + poly(PCH_WICS_16_22, 9, raw = TRUE) + poly(PCH_FFR_16_20,
##     9, raw = TRUE) + poly(PCH_FSR_16_20, 9, raw = TRUE) + poly(PCH_DIRSALES_12_17,
##     9, raw = TRUE), data = dat.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.964  -6.435  -0.421   5.293 120.344
##
## Coefficients:
```

```
##                                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                             -4.437e+00  3.089e+00  -1.436  0.15162
## poly(PCH_GROC_16_20, 9, raw = TRUE)1    -2.558e-01  1.577e-01  -1.622  0.10549
## poly(PCH_GROC_16_20, 9, raw = TRUE)2     2.342e-03  1.319e-02   0.177  0.85921
## poly(PCH_GROC_16_20, 9, raw = TRUE)3     5.874e-04  7.174e-04   0.819  0.41336
## poly(PCH_GROC_16_20, 9, raw = TRUE)4    -2.773e-06  3.145e-05  -0.088  0.92978
## poly(PCH_GROC_16_20, 9, raw = TRUE)5    -8.641e-07  7.744e-07  -1.116  0.26513
## poly(PCH_GROC_16_20, 9, raw = TRUE)6     9.450e-09  2.616e-08   0.361  0.71807
## poly(PCH_GROC_16_20, 9, raw = TRUE)7     2.801e-10  1.982e-10   1.413  0.15835
## poly(PCH_GROC_16_20, 9, raw = TRUE)8    -5.151e-12  7.324e-12  -0.703  0.48220
## poly(PCH_GROC_16_20, 9, raw = TRUE)9     2.237e-14  4.720e-14   0.474  0.63572
## poly(PCH_SUPERC_16_20, 9, raw = TRUE)1  -2.879e-02  8.621e-02  -0.334  0.73861
## poly(PCH_SUPERC_16_20, 9, raw = TRUE)2   2.694e-03  2.641e-03   1.020  0.30829
## poly(PCH_SUPERC_16_20, 9, raw = TRUE)3  -3.814e-05  3.663e-05  -1.041  0.29836
## poly(PCH_SUPERC_16_20, 9, raw = TRUE)4   2.360e-07  2.433e-07   0.970  0.33265
## poly(PCH_SUPERC_16_20, 9, raw = TRUE)5  -7.641e-10  8.422e-10  -0.907  0.36477
## poly(PCH_SUPERC_16_20, 9, raw = TRUE)6   1.356e-12  1.575e-12   0.861  0.38963
## poly(PCH_SUPERC_16_20, 9, raw = TRUE)7  -1.304e-15  1.575e-15  -0.828  0.40804
## poly(PCH_SUPERC_16_20, 9, raw = TRUE)8   6.271e-19  7.784e-19   0.806  0.42092
## poly(PCH_SUPERC_16_20, 9, raw = TRUE)9  -1.172e-22  1.482e-22  -0.790  0.42970
## poly(PCH_CONVS_16_20, 9, raw = TRUE)1    7.758e-02  1.921e-01   0.404  0.68661
## poly(PCH_CONVS_16_20, 9, raw = TRUE)2    6.398e-02  2.344e-02   2.729  0.00662
## poly(PCH_CONVS_16_20, 9, raw = TRUE)3   -1.676e-03  1.262e-03  -1.329  0.18463
## poly(PCH_CONVS_16_20, 9, raw = TRUE)4   -2.221e-04  9.541e-05  -2.328  0.02042
## poly(PCH_CONVS_16_20, 9, raw = TRUE)5    6.107e-06  2.602e-06   2.347  0.01938
## poly(PCH_CONVS_16_20, 9, raw = TRUE)6    1.616e-07  8.779e-08   1.841  0.06639
## poly(PCH_CONVS_16_20, 9, raw = TRUE)7   -6.553e-09  2.852e-09  -2.298  0.02208
## poly(PCH_CONVS_16_20, 9, raw = TRUE)8    6.282e-11  2.691e-11   2.334  0.02007
## poly(PCH_CONVS_16_20, 9, raw = TRUE)9   -1.820e-13  7.795e-14  -2.335  0.02003
## poly(PCH_SPECS_16_20, 9, raw = TRUE)1    1.093e-02  7.642e-02   0.143  0.88629
## poly(PCH_SPECS_16_20, 9, raw = TRUE)2    4.026e-03  3.717e-03   1.083  0.27940
## poly(PCH_SPECS_16_20, 9, raw = TRUE)3   -1.256e-05  7.538e-05  -0.167  0.86776
## poly(PCH_SPECS_16_20, 9, raw = TRUE)4   -4.729e-06  2.238e-06  -2.113  0.03523
## poly(PCH_SPECS_16_20, 9, raw = TRUE)5    4.763e-08  2.628e-08   1.812  0.07069
## poly(PCH_SPECS_16_20, 9, raw = TRUE)6    6.981e-10  2.960e-10   2.359  0.01880
## poly(PCH_SPECS_16_20, 9, raw = TRUE)7   -1.244e-11  4.564e-12  -2.725  0.00670
## poly(PCH_SPECS_16_20, 9, raw = TRUE)8    6.111e-14  2.242e-14   2.725  0.00670
## poly(PCH_SPECS_16_20, 9, raw = TRUE)9   -9.530e-17  3.521e-17  -2.706  0.00708
## poly(PCH_SNAPS_17_23, 9, raw = TRUE)1    1.978e-01  4.683e-01   0.422  0.67293
## poly(PCH_SNAPS_17_23, 9, raw = TRUE)2    5.103e-02  5.816e-02   0.877  0.38078
## poly(PCH_SNAPS_17_23, 9, raw = TRUE)3   -1.191e-02  5.356e-03  -2.223  0.02673
## poly(PCH_SNAPS_17_23, 9, raw = TRUE)4    5.342e-04  4.623e-04   1.155  0.24856
## poly(PCH_SNAPS_17_23, 9, raw = TRUE)5    1.887e-05  1.397e-05   1.351  0.17752
## poly(PCH_SNAPS_17_23, 9, raw = TRUE)6   -2.178e-06  1.104e-06  -1.972  0.04927
## poly(PCH_SNAPS_17_23, 9, raw = TRUE)7    6.690e-08  4.378e-08   1.528  0.12725
## poly(PCH_SNAPS_17_23, 9, raw = TRUE)8   -8.889e-10  6.983e-10  -1.273  0.20377
## poly(PCH_SNAPS_17_23, 9, raw = TRUE)9    4.402e-12  4.001e-12   1.100  0.27190
## poly(PCH_WICS_16_22, 9, raw = TRUE)1    -8.326e-02  1.199e-01  -0.695  0.48765
## poly(PCH_WICS_16_22, 9, raw = TRUE)2     1.243e-02  6.015e-03   2.067  0.03936
## poly(PCH_WICS_16_22, 9, raw = TRUE)3     2.035e-04  2.799e-04   0.727  0.46770
## poly(PCH_WICS_16_22, 9, raw = TRUE)4    -8.238e-06  4.501e-06  -1.830  0.06796
## poly(PCH_WICS_16_22, 9, raw = TRUE)5    -1.114e-07  1.636e-07  -0.681  0.49640
## poly(PCH_WICS_16_22, 9, raw = TRUE)6     1.186e-09  9.496e-10   1.249  0.21245
## poly(PCH_WICS_16_22, 9, raw = TRUE)7     1.404e-11  2.431e-11   0.578  0.56375
```

```
## poly(PCH_WICS_16_22, 9, raw = TRUE)8     -3.384e-14  8.052e-14  -0.420  0.67454
## poly(PCH_WICS_16_22, 9, raw = TRUE)9     -3.631e-16  9.822e-16  -0.370  0.71177
## poly(PCH_FFR_16_20, 9, raw = TRUE)1      -2.560e-01  3.277e-01  -0.781  0.43519
## poly(PCH_FFR_16_20, 9, raw = TRUE)2       1.695e-02  4.161e-02   0.407  0.68393
## poly(PCH_FFR_16_20, 9, raw = TRUE)3      -6.317e-04  4.670e-03  -0.135  0.89247
## poly(PCH_FFR_16_20, 9, raw = TRUE)4      -8.181e-05  2.373e-04  -0.345  0.73041
## poly(PCH_FFR_16_20, 9, raw = TRUE)5       6.681e-06  1.987e-05   0.336  0.73694
## poly(PCH_FFR_16_20, 9, raw = TRUE)6      -1.659e-08  4.764e-07  -0.035  0.97224
## poly(PCH_FFR_16_20, 9, raw = TRUE)7      -8.318e-09  2.286e-08  -0.364  0.71617
## poly(PCH_FFR_16_20, 9, raw = TRUE)8       1.985e-10  7.322e-10   0.271  0.78642
## poly(PCH_FFR_16_20, 9, raw = TRUE)9      -1.319e-12  6.271e-12  -0.210  0.83346
## poly(PCH_FSR_16_20, 9, raw = TRUE)1       3.636e-01  3.417e-01   1.064  0.28790
## poly(PCH_FSR_16_20, 9, raw = TRUE)2       4.332e-02  5.271e-02   0.822  0.41164
## poly(PCH_FSR_16_20, 9, raw = TRUE)3      -6.636e-03  6.804e-03  -0.975  0.32991
## poly(PCH_FSR_16_20, 9, raw = TRUE)4      -1.970e-04  5.366e-04  -0.367  0.71365
## poly(PCH_FSR_16_20, 9, raw = TRUE)5       4.272e-05  3.226e-05   1.324  0.18618
## poly(PCH_FSR_16_20, 9, raw = TRUE)6      -1.227e-06  2.126e-06  -0.577  0.56409
## poly(PCH_FSR_16_20, 9, raw = TRUE)7      -4.005e-08  3.254e-08  -1.231  0.21904
## poly(PCH_FSR_16_20, 9, raw = TRUE)8       2.739e-09  2.645e-09   1.036  0.30101
## poly(PCH_FSR_16_20, 9, raw = TRUE)9      -3.990e-11  4.019e-11  -0.993  0.32142
## poly(PCH_DIRSALES_12_17, 9, raw = TRUE)1 -2.784e-02  2.262e-02  -1.231  0.21903
## poly(PCH_DIRSALES_12_17, 9, raw = TRUE)2 -2.769e-04  3.471e-04  -0.798  0.42554
## poly(PCH_DIRSALES_12_17, 9, raw = TRUE)3  4.294e-06  3.672e-06   1.169  0.24293
## poly(PCH_DIRSALES_12_17, 9, raw = TRUE)4 -1.864e-08  1.652e-08  -1.129  0.25960
## poly(PCH_DIRSALES_12_17, 9, raw = TRUE)5  3.869e-11  3.720e-11   1.040  0.29887
## poly(PCH_DIRSALES_12_17, 9, raw = TRUE)6 -4.355e-14  4.583e-14  -0.950  0.34251
## poly(PCH_DIRSALES_12_17, 9, raw = TRUE)7  2.721e-17  3.132e-17   0.869  0.38560
## poly(PCH_DIRSALES_12_17, 9, raw = TRUE)8 -8.863e-21  1.112e-20  -0.797  0.42591
## poly(PCH_DIRSALES_12_17, 9, raw = TRUE)9  1.174e-24  1.596e-24   0.736  0.46233
##
## (Intercept)
## poly(PCH_GROC_16_20, 9, raw = TRUE)1
## poly(PCH_GROC_16_20, 9, raw = TRUE)2
## poly(PCH_GROC_16_20, 9, raw = TRUE)3
## poly(PCH_GROC_16_20, 9, raw = TRUE)4
## poly(PCH_GROC_16_20, 9, raw = TRUE)5
## poly(PCH_GROC_16_20, 9, raw = TRUE)6
## poly(PCH_GROC_16_20, 9, raw = TRUE)7
## poly(PCH_GROC_16_20, 9, raw = TRUE)8
## poly(PCH_GROC_16_20, 9, raw = TRUE)9
## poly(PCH_SUPERC_16_20, 9, raw = TRUE)1
## poly(PCH_SUPERC_16_20, 9, raw = TRUE)2
## poly(PCH_SUPERC_16_20, 9, raw = TRUE)3
## poly(PCH_SUPERC_16_20, 9, raw = TRUE)4
## poly(PCH_SUPERC_16_20, 9, raw = TRUE)5
## poly(PCH_SUPERC_16_20, 9, raw = TRUE)6
## poly(PCH_SUPERC_16_20, 9, raw = TRUE)7
## poly(PCH_SUPERC_16_20, 9, raw = TRUE)8
## poly(PCH_SUPERC_16_20, 9, raw = TRUE)9
## poly(PCH_CONVS_16_20, 9, raw = TRUE)1
## poly(PCH_CONVS_16_20, 9, raw = TRUE)2    **
## poly(PCH_CONVS_16_20, 9, raw = TRUE)3
## poly(PCH_CONVS_16_20, 9, raw = TRUE)4    *
## poly(PCH_CONVS_16_20, 9, raw = TRUE)5    *
```

```
## poly(PCH_CONVS_16_20, 9, raw = TRUE)6    .
## poly(PCH_CONVS_16_20, 9, raw = TRUE)7    *
## poly(PCH_CONVS_16_20, 9, raw = TRUE)8    *
## poly(PCH_CONVS_16_20, 9, raw = TRUE)9    *
## poly(PCH_SPECS_16_20, 9, raw = TRUE)1
## poly(PCH_SPECS_16_20, 9, raw = TRUE)2
## poly(PCH_SPECS_16_20, 9, raw = TRUE)3
## poly(PCH_SPECS_16_20, 9, raw = TRUE)4    *
## poly(PCH_SPECS_16_20, 9, raw = TRUE)5    .
## poly(PCH_SPECS_16_20, 9, raw = TRUE)6    *
## poly(PCH_SPECS_16_20, 9, raw = TRUE)7    **
## poly(PCH_SPECS_16_20, 9, raw = TRUE)8    **
## poly(PCH_SPECS_16_20, 9, raw = TRUE)9    **
## poly(PCH_SNAPS_17_23, 9, raw = TRUE)1
## poly(PCH_SNAPS_17_23, 9, raw = TRUE)2
## poly(PCH_SNAPS_17_23, 9, raw = TRUE)3    *
## poly(PCH_SNAPS_17_23, 9, raw = TRUE)4
## poly(PCH_SNAPS_17_23, 9, raw = TRUE)5
## poly(PCH_SNAPS_17_23, 9, raw = TRUE)6    *
## poly(PCH_SNAPS_17_23, 9, raw = TRUE)7
## poly(PCH_SNAPS_17_23, 9, raw = TRUE)8
## poly(PCH_SNAPS_17_23, 9, raw = TRUE)9
## poly(PCH_WICS_16_22, 9, raw = TRUE)1
## poly(PCH_WICS_16_22, 9, raw = TRUE)2     *
## poly(PCH_WICS_16_22, 9, raw = TRUE)3
## poly(PCH_WICS_16_22, 9, raw = TRUE)4     .
## poly(PCH_WICS_16_22, 9, raw = TRUE)5
## poly(PCH_WICS_16_22, 9, raw = TRUE)6
## poly(PCH_WICS_16_22, 9, raw = TRUE)7
## poly(PCH_WICS_16_22, 9, raw = TRUE)8
## poly(PCH_WICS_16_22, 9, raw = TRUE)9
## poly(PCH_FFR_16_20, 9, raw = TRUE)1
## poly(PCH_FFR_16_20, 9, raw = TRUE)2
## poly(PCH_FFR_16_20, 9, raw = TRUE)3
## poly(PCH_FFR_16_20, 9, raw = TRUE)4
## poly(PCH_FFR_16_20, 9, raw = TRUE)5
## poly(PCH_FFR_16_20, 9, raw = TRUE)6
## poly(PCH_FFR_16_20, 9, raw = TRUE)7
## poly(PCH_FFR_16_20, 9, raw = TRUE)8
## poly(PCH_FFR_16_20, 9, raw = TRUE)9
## poly(PCH_FSR_16_20, 9, raw = TRUE)1
## poly(PCH_FSR_16_20, 9, raw = TRUE)2
## poly(PCH_FSR_16_20, 9, raw = TRUE)3
## poly(PCH_FSR_16_20, 9, raw = TRUE)4
## poly(PCH_FSR_16_20, 9, raw = TRUE)5
## poly(PCH_FSR_16_20, 9, raw = TRUE)6
## poly(PCH_FSR_16_20, 9, raw = TRUE)7
## poly(PCH_FSR_16_20, 9, raw = TRUE)8
## poly(PCH_FSR_16_20, 9, raw = TRUE)9
## poly(PCH_DIRSALES_12_17, 9, raw = TRUE)1
## poly(PCH_DIRSALES_12_17, 9, raw = TRUE)2
## poly(PCH_DIRSALES_12_17, 9, raw = TRUE)3
## poly(PCH_DIRSALES_12_17, 9, raw = TRUE)4
## poly(PCH_DIRSALES_12_17, 9, raw = TRUE)5
```

```
## poly(PCH_DIRSALES_12_17, 9, raw = TRUE)6
## poly(PCH_DIRSALES_12_17, 9, raw = TRUE)7
## poly(PCH_DIRSALES_12_17, 9, raw = TRUE)8
## poly(PCH_DIRSALES_12_17, 9, raw = TRUE)9
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.91 on 413 degrees of freedom
## Multiple R-squared:  0.242,  Adjusted R-squared:  0.09328
## F-statistic: 1.627 on 81 and 413 DF,  p-value: 0.001255
```

```
fitdeg_10 <- lm(PCH_LACCESS_POP_15_19 ~ poly(PCH_GROC_16_20, 10, raw=TRUE) + poly(PCH_SUPERC_16_20, 10,
predictions_10 <- predict(fitdeg_10, newdata = dat.valid)
mean((dat.valid$PCH_LACCESS_POP_15_19 - predictions_10)^2)
```

```
## [1] 1.800208e+31
```

```
AIC(fitdeg_10)
```

```
## [1] 4140.072
```

```
BIC(fitdeg_10)
```

```
## [1] 4522.687
```

```
summary(fitdeg_10)
```

```
##
## Call:
## lm(formula = PCH_LACCESS_POP_15_19 ~ poly(PCH_GROC_16_20, 10,
##     raw = TRUE) + poly(PCH_SUPERC_16_20, 10, raw = TRUE) + poly(PCH_CONVS_16_20,
##     10, raw = TRUE) + poly(PCH_SPECS_16_20, 10, raw = TRUE) +
##     poly(PCH_SNAPS_17_23, 10, raw = TRUE) + poly(PCH_WICS_16_22,
##     10, raw = TRUE) + poly(PCH_FFR_16_20, 10, raw = TRUE) + poly(PCH_FSR_16_20,
##     10, raw = TRUE) + poly(PCH_DIRSALES_12_17, 10, raw = TRUE),
##     data = dat.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -43.265  -6.260  -0.185   5.254 112.995
##
## Coefficients: (1 not defined because of singularities)
##                                              Estimate Std. Error t value
## (Intercept)                                -4.092e+00  3.198e+00  -1.280
## poly(PCH_GROC_16_20, 10, raw = TRUE)1      -2.568e-01  1.763e-01  -1.457
## poly(PCH_GROC_16_20, 10, raw = TRUE)2       6.537e-03  1.575e-02   0.415
## poly(PCH_GROC_16_20, 10, raw = TRUE)3       4.674e-04  1.024e-03   0.457
## poly(PCH_GROC_16_20, 10, raw = TRUE)4      -1.909e-05  4.398e-05  -0.434
## poly(PCH_GROC_16_20, 10, raw = TRUE)5      -4.265e-07  1.705e-06  -0.250
## poly(PCH_GROC_16_20, 10, raw = TRUE)6       2.713e-08  3.707e-08   0.732
## poly(PCH_GROC_16_20, 10, raw = TRUE)7      -2.211e-10  1.164e-09  -0.190
## poly(PCH_GROC_16_20, 10, raw = TRUE)8      -8.829e-12  7.924e-12  -1.114
## poly(PCH_GROC_16_20, 10, raw = TRUE)9       1.849e-13  2.914e-13   0.635
## poly(PCH_GROC_16_20, 10, raw = TRUE)10     -9.600e-16  1.796e-15  -0.534
## poly(PCH_SUPERC_16_20, 10, raw = TRUE)1    -6.544e-02  8.540e-02  -0.766
## poly(PCH_SUPERC_16_20, 10, raw = TRUE)2     3.964e-03  2.625e-03   1.510
## poly(PCH_SUPERC_16_20, 10, raw = TRUE)3    -5.613e-05  3.648e-05  -1.539
```

```
## poly(PCH_SUPERC_16_20, 10, raw = TRUE)4     3.553e-07  2.425e-07   1.465
## poly(PCH_SUPERC_16_20, 10, raw = TRUE)5    -1.172e-09  8.391e-10  -1.397
## poly(PCH_SUPERC_16_20, 10, raw = TRUE)6     2.107e-12  1.568e-12   1.343
## poly(PCH_SUPERC_16_20, 10, raw = TRUE)7    -2.044e-15  1.568e-15  -1.303
## poly(PCH_SUPERC_16_20, 10, raw = TRUE)8     9.881e-19  7.751e-19   1.275
## poly(PCH_SUPERC_16_20, 10, raw = TRUE)9    -1.851e-22  1.476e-22  -1.254
## poly(PCH_SUPERC_16_20, 10, raw = TRUE)10           NA         NA      NA
## poly(PCH_CONVS_16_20, 10, raw = TRUE)1      1.083e-01  2.344e-01   0.462
## poly(PCH_CONVS_16_20, 10, raw = TRUE)2      6.481e-02  2.456e-02   2.639
## poly(PCH_CONVS_16_20, 10, raw = TRUE)3     -2.479e-03  2.387e-03  -1.039
## poly(PCH_CONVS_16_20, 10, raw = TRUE)4     -2.260e-04  9.821e-05  -2.302
## poly(PCH_CONVS_16_20, 10, raw = TRUE)5      9.084e-06  7.190e-06   1.263
## poly(PCH_CONVS_16_20, 10, raw = TRUE)6      1.134e-07  1.302e-07   0.871
## poly(PCH_CONVS_16_20, 10, raw = TRUE)7     -8.952e-09  5.977e-09  -1.498
## poly(PCH_CONVS_16_20, 10, raw = TRUE)8      1.391e-10  1.562e-10   0.890
## poly(PCH_CONVS_16_20, 10, raw = TRUE)9     -8.693e-13  1.345e-12  -0.646
## poly(PCH_CONVS_16_20, 10, raw = TRUE)10     1.936e-15  3.717e-15   0.521
## poly(PCH_SPECS_16_20, 10, raw = TRUE)1      5.384e-02  9.667e-02   0.557
## poly(PCH_SPECS_16_20, 10, raw = TRUE)2      3.144e-03  3.704e-03   0.849
## poly(PCH_SPECS_16_20, 10, raw = TRUE)3     -9.396e-05  1.492e-04  -0.630
## poly(PCH_SPECS_16_20, 10, raw = TRUE)4     -3.957e-06  2.202e-06  -1.797
## poly(PCH_SPECS_16_20, 10, raw = TRUE)5      8.014e-08  6.691e-08   1.198
## poly(PCH_SPECS_16_20, 10, raw = TRUE)6      3.164e-10  5.438e-10   0.582
## poly(PCH_SPECS_16_20, 10, raw = TRUE)7     -1.491e-11  7.911e-12  -1.885
## poly(PCH_SPECS_16_20, 10, raw = TRUE)8      1.142e-13  1.001e-13   1.140
## poly(PCH_SPECS_16_20, 10, raw = TRUE)9     -3.509e-16  4.327e-16  -0.811
## poly(PCH_SPECS_16_20, 10, raw = TRUE)10     3.869e-19  6.256e-19   0.618
## poly(PCH_SNAPS_17_23, 10, raw = TRUE)1      3.743e-01  4.694e-01   0.797
## poly(PCH_SNAPS_17_23, 10, raw = TRUE)2     -1.109e-01  8.668e-02  -1.280
## poly(PCH_SNAPS_17_23, 10, raw = TRUE)3     -3.175e-03  6.277e-03  -0.506
## poly(PCH_SNAPS_17_23, 10, raw = TRUE)4      1.608e-03  6.758e-04   2.379
## poly(PCH_SNAPS_17_23, 10, raw = TRUE)5     -7.000e-05  4.314e-05  -1.622
## poly(PCH_SNAPS_17_23, 10, raw = TRUE)6     -2.250e-06  1.090e-06  -2.064
## poly(PCH_SNAPS_17_23, 10, raw = TRUE)7      2.480e-07  9.815e-08   2.527
## poly(PCH_SNAPS_17_23, 10, raw = TRUE)8     -7.339e-09  3.240e-09  -2.265
## poly(PCH_SNAPS_17_23, 10, raw = TRUE)9      9.542e-11  4.532e-11   2.105
## poly(PCH_SNAPS_17_23, 10, raw = TRUE)10    -4.692e-13  2.351e-13  -1.996
## poly(PCH_WICS_16_22, 10, raw = TRUE)1       8.152e-02  1.335e-01   0.611
## poly(PCH_WICS_16_22, 10, raw = TRUE)2       2.942e-02  8.597e-03   3.422
## poly(PCH_WICS_16_22, 10, raw = TRUE)3      -2.174e-04  3.130e-04  -0.695
## poly(PCH_WICS_16_22, 10, raw = TRUE)4      -3.904e-05  1.277e-05  -3.058
## poly(PCH_WICS_16_22, 10, raw = TRUE)5       5.505e-08  1.697e-07   0.324
## poly(PCH_WICS_16_22, 10, raw = TRUE)6       1.604e-08  5.931e-09   2.704
## poly(PCH_WICS_16_22, 10, raw = TRUE)7       3.164e-11  2.551e-11   1.241
## poly(PCH_WICS_16_22, 10, raw = TRUE)8      -2.040e-12  8.013e-13  -2.546
## poly(PCH_WICS_16_22, 10, raw = TRUE)9      -4.427e-15  1.970e-15  -2.247
## poly(PCH_WICS_16_22, 10, raw = TRUE)10      7.380e-17  2.964e-17   2.490
## poly(PCH_FFR_16_20, 10, raw = TRUE)1       -8.179e-01  3.723e-01  -2.197
## poly(PCH_FFR_16_20, 10, raw = TRUE)2        1.145e-01  5.042e-02   2.271
## poly(PCH_FFR_16_20, 10, raw = TRUE)3        1.037e-02  5.659e-03   1.832
## poly(PCH_FFR_16_20, 10, raw = TRUE)4       -1.450e-03  4.738e-04  -3.061
## poly(PCH_FFR_16_20, 10, raw = TRUE)5       -2.109e-05  2.102e-05  -1.003
## poly(PCH_FFR_16_20, 10, raw = TRUE)6        5.173e-06  1.611e-06   3.212
## poly(PCH_FFR_16_20, 10, raw = TRUE)7       -6.922e-08  2.913e-08  -2.377
```

```
## poly(PCH_FFR_16_20, 10, raw = TRUE)8       -4.896e-09  1.655e-09  -2.958
## poly(PCH_FFR_16_20, 10, raw = TRUE)9        1.528e-10  4.586e-11   3.331
## poly(PCH_FFR_16_20, 10, raw = TRUE)10      -1.222e-12  3.613e-13  -3.382
## poly(PCH_FSR_16_20, 10, raw = TRUE)1        3.082e-01  3.436e-01   0.897
## poly(PCH_FSR_16_20, 10, raw = TRUE)2        2.845e-02  7.156e-02   0.398
## poly(PCH_FSR_16_20, 10, raw = TRUE)3       -5.338e-03  7.437e-03  -0.718
## poly(PCH_FSR_16_20, 10, raw = TRUE)4       -8.648e-05  8.926e-04  -0.097
## poly(PCH_FSR_16_20, 10, raw = TRUE)5        3.565e-05  5.116e-05   0.697
## poly(PCH_FSR_16_20, 10, raw = TRUE)6       -1.342e-06  3.435e-06  -0.391
## poly(PCH_FSR_16_20, 10, raw = TRUE)7       -3.534e-08  1.721e-07  -0.205
## poly(PCH_FSR_16_20, 10, raw = TRUE)8        2.766e-09  3.155e-09   0.877
## poly(PCH_FSR_16_20, 10, raw = TRUE)9       -3.045e-11  2.085e-10  -0.146
## poly(PCH_FSR_16_20, 10, raw = TRUE)10      -2.714e-13  2.862e-12  -0.095
## poly(PCH_DIRSALES_12_17, 10, raw = TRUE)1  -1.220e-02  2.723e-02  -0.448
## poly(PCH_DIRSALES_12_17, 10, raw = TRUE)2  -2.327e-04  3.414e-04  -0.681
## poly(PCH_DIRSALES_12_17, 10, raw = TRUE)3   3.349e-07  4.842e-06   0.069
## poly(PCH_DIRSALES_12_17, 10, raw = TRUE)4   1.411e-08  3.136e-08   0.450
## poly(PCH_DIRSALES_12_17, 10, raw = TRUE)5  -7.813e-11  1.004e-10  -0.778
## poly(PCH_DIRSALES_12_17, 10, raw = TRUE)6   1.780e-13  1.789e-13   0.995
## poly(PCH_DIRSALES_12_17, 10, raw = TRUE)7  -2.125e-16  1.859e-16  -1.143
## poly(PCH_DIRSALES_12_17, 10, raw = TRUE)8   1.392e-19  1.117e-19   1.247
## poly(PCH_DIRSALES_12_17, 10, raw = TRUE)9  -4.738e-23  3.589e-23  -1.320
## poly(PCH_DIRSALES_12_17, 10, raw = TRUE)10  6.537e-27  4.763e-27   1.373
##                                            Pr(>|t|)
## (Intercept)                                0.201434
## poly(PCH_GROC_16_20, 10, raw = TRUE)1      0.145956
## poly(PCH_GROC_16_20, 10, raw = TRUE)2      0.678434
## poly(PCH_GROC_16_20, 10, raw = TRUE)3      0.648193
## poly(PCH_GROC_16_20, 10, raw = TRUE)4      0.664580
## poly(PCH_GROC_16_20, 10, raw = TRUE)5      0.802591
## poly(PCH_GROC_16_20, 10, raw = TRUE)6      0.464702
## poly(PCH_GROC_16_20, 10, raw = TRUE)7      0.849412
## poly(PCH_GROC_16_20, 10, raw = TRUE)8      0.265885
## poly(PCH_GROC_16_20, 10, raw = TRUE)9      0.526048
## poly(PCH_GROC_16_20, 10, raw = TRUE)10     0.593315
## poly(PCH_SUPERC_16_20, 10, raw = TRUE)1    0.443967
## poly(PCH_SUPERC_16_20, 10, raw = TRUE)2    0.131771
## poly(PCH_SUPERC_16_20, 10, raw = TRUE)3    0.124691
## poly(PCH_SUPERC_16_20, 10, raw = TRUE)4    0.143583
## poly(PCH_SUPERC_16_20, 10, raw = TRUE)5    0.163220
## poly(PCH_SUPERC_16_20, 10, raw = TRUE)6    0.179870
## poly(PCH_SUPERC_16_20, 10, raw = TRUE)7    0.193165
## poly(PCH_SUPERC_16_20, 10, raw = TRUE)8    0.203132
## poly(PCH_SUPERC_16_20, 10, raw = TRUE)9    0.210395
## poly(PCH_SUPERC_16_20, 10, raw = TRUE)10         NA
## poly(PCH_CONVS_16_20, 10, raw = TRUE)1     0.644395
## poly(PCH_CONVS_16_20, 10, raw = TRUE)2     0.008638 **
## poly(PCH_CONVS_16_20, 10, raw = TRUE)3     0.299508
## poly(PCH_CONVS_16_20, 10, raw = TRUE)4     0.021865 *
## poly(PCH_CONVS_16_20, 10, raw = TRUE)5     0.207154
## poly(PCH_CONVS_16_20, 10, raw = TRUE)6     0.384536
## poly(PCH_CONVS_16_20, 10, raw = TRUE)7     0.134989
## poly(PCH_CONVS_16_20, 10, raw = TRUE)8     0.373751
## poly(PCH_CONVS_16_20, 10, raw = TRUE)9     0.518490
```

```
## poly(PCH_CONVS_16_20, 10, raw = TRUE)10    0.602677
## poly(PCH_SPECS_16_20, 10, raw = TRUE)1     0.577857
## poly(PCH_SPECS_16_20, 10, raw = TRUE)2     0.396387
## poly(PCH_SPECS_16_20, 10, raw = TRUE)3     0.529112
## poly(PCH_SPECS_16_20, 10, raw = TRUE)4     0.073038 .
## poly(PCH_SPECS_16_20, 10, raw = TRUE)5     0.231709
## poly(PCH_SPECS_16_20, 10, raw = TRUE)6     0.560987
## poly(PCH_SPECS_16_20, 10, raw = TRUE)7     0.060146 .
## poly(PCH_SPECS_16_20, 10, raw = TRUE)8     0.254764
## poly(PCH_SPECS_16_20, 10, raw = TRUE)9     0.417922
## poly(PCH_SPECS_16_20, 10, raw = TRUE)10    0.536630
## poly(PCH_SNAPS_17_23, 10, raw = TRUE)1     0.425703
## poly(PCH_SNAPS_17_23, 10, raw = TRUE)2     0.201413
## poly(PCH_SNAPS_17_23, 10, raw = TRUE)3     0.613326
## poly(PCH_SNAPS_17_23, 10, raw = TRUE)4     0.017801 *
## poly(PCH_SNAPS_17_23, 10, raw = TRUE)5     0.105476
## poly(PCH_SNAPS_17_23, 10, raw = TRUE)6     0.039666 *
## poly(PCH_SNAPS_17_23, 10, raw = TRUE)7     0.011888 *
## poly(PCH_SNAPS_17_23, 10, raw = TRUE)8     0.024044 *
## poly(PCH_SNAPS_17_23, 10, raw = TRUE)9     0.035891 *
## poly(PCH_SNAPS_17_23, 10, raw = TRUE)10    0.046589 *
## poly(PCH_WICS_16_22, 10, raw = TRUE)1      0.541641
## poly(PCH_WICS_16_22, 10, raw = TRUE)2      0.000684 ***
## poly(PCH_WICS_16_22, 10, raw = TRUE)3      0.487707
## poly(PCH_WICS_16_22, 10, raw = TRUE)4      0.002376 **
## poly(PCH_WICS_16_22, 10, raw = TRUE)5      0.745794
## poly(PCH_WICS_16_22, 10, raw = TRUE)6      0.007139 **
## poly(PCH_WICS_16_22, 10, raw = TRUE)7      0.215461
## poly(PCH_WICS_16_22, 10, raw = TRUE)8      0.011252 *
## poly(PCH_WICS_16_22, 10, raw = TRUE)9      0.025155 *
## poly(PCH_WICS_16_22, 10, raw = TRUE)10     0.013168 *
## poly(PCH_FFR_16_20, 10, raw = TRUE)1       0.028573 *
## poly(PCH_FFR_16_20, 10, raw = TRUE)2       0.023686 *
## poly(PCH_FFR_16_20, 10, raw = TRUE)3       0.067693 .
## poly(PCH_FFR_16_20, 10, raw = TRUE)4       0.002355 **
## poly(PCH_FFR_16_20, 10, raw = TRUE)5       0.316291
## poly(PCH_FFR_16_20, 10, raw = TRUE)6       0.001424 **
## poly(PCH_FFR_16_20, 10, raw = TRUE)7       0.017934 *
## poly(PCH_FFR_16_20, 10, raw = TRUE)8       0.003274 **
## poly(PCH_FFR_16_20, 10, raw = TRUE)9       0.000944 ***
## poly(PCH_FFR_16_20, 10, raw = TRUE)10      0.000788 ***
## poly(PCH_FSR_16_20, 10, raw = TRUE)1       0.370346
## poly(PCH_FSR_16_20, 10, raw = TRUE)2       0.691205
## poly(PCH_FSR_16_20, 10, raw = TRUE)3       0.473333
## poly(PCH_FSR_16_20, 10, raw = TRUE)4       0.922868
## poly(PCH_FSR_16_20, 10, raw = TRUE)5       0.486302
## poly(PCH_FSR_16_20, 10, raw = TRUE)6       0.696286
## poly(PCH_FSR_16_20, 10, raw = TRUE)7       0.837393
## poly(PCH_FSR_16_20, 10, raw = TRUE)8       0.381083
## poly(PCH_FSR_16_20, 10, raw = TRUE)9       0.883944
## poly(PCH_FSR_16_20, 10, raw = TRUE)10      0.924503
## poly(PCH_DIRSALES_12_17, 10, raw = TRUE)1  0.654237
## poly(PCH_DIRSALES_12_17, 10, raw = TRUE)2  0.495955
## poly(PCH_DIRSALES_12_17, 10, raw = TRUE)3  0.944886
```

```
## poly(PCH_DIRSALES_12_17, 10, raw = TRUE)4  0.652923
## poly(PCH_DIRSALES_12_17, 10, raw = TRUE)5  0.437060
## poly(PCH_DIRSALES_12_17, 10, raw = TRUE)6  0.320404
## poly(PCH_DIRSALES_12_17, 10, raw = TRUE)7  0.253631
## poly(PCH_DIRSALES_12_17, 10, raw = TRUE)8  0.213201
## poly(PCH_DIRSALES_12_17, 10, raw = TRUE)9  0.187553
## poly(PCH_DIRSALES_12_17, 10, raw = TRUE)10 0.170661
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.58 on 405 degrees of freedom
## Multiple R-squared:  0.2893, Adjusted R-squared:  0.1331
## F-statistic: 1.853 on 89 and 405 DF,  p-value: 3.252e-05
```

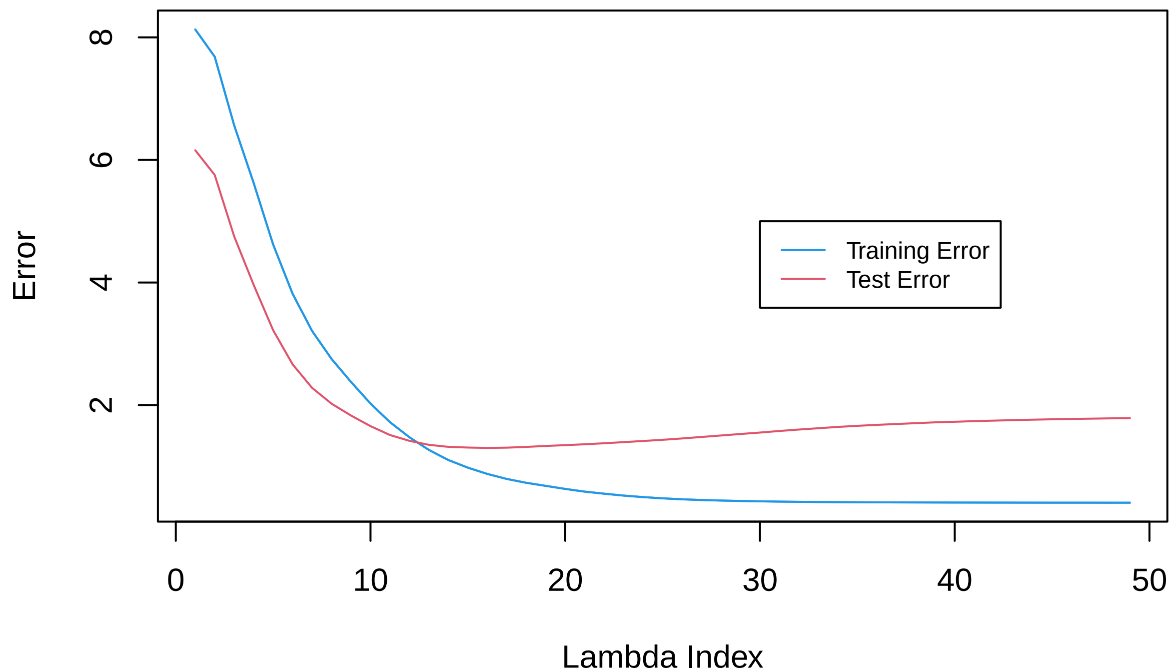
The concept of bias-variance trade-off is that when you choose between estimators, you are choosing between reducing bias or reducing variance. The expected prediction error and therefore the mean squared error both include terms for squared bias and variance. Regularizing the model through different methods of shrinking, dampening, or controlling the estimates will reduce variance while increasing bias. As long as the increase in bias is smaller than the decrease in variance, this is a helpful tradeoff to make.

```
# L1 regularization

fit1 <- lm(Y~X-1,data=atlas2)
lam <- .01
fitl <- glmnet(x=X,y=Y,family="gaussian",lambda=lam,alpha=1)
cbind(fit1$coef,as.matrix(fitl$beta))
```

```
##                                              s0
## XPCH_GROC_16_20     -0.1089863758 -0.1083396753
## XPCH_SUPERC_16_20    0.0007047547  0.0006070183
## XPCH_CONVS_16_20    -0.0718614173 -0.0709547145
## XPCH_SPECS_16_20    -0.0032387185 -0.0029274202
## XPCH_SNAPS_17_23    -0.0692429375 -0.0683904389
## XPCH_WICS_16_22     -0.0089636422 -0.0088316445
## XPCH_FFR_16_20      -0.2667555927 -0.2660301396
## XPCH_FSR_16_20       0.1400872946  0.1382014447
## XPCH_DIRSALES_12_17 -0.0016519904 -0.0016076121
```

```
# model complexity and prediction error

n = 100
p = 50
Btrue = matrix(0,p,1)
Btrue[1:10] = rnorm(10)*.75
Btrue[11:20] = rnorm(10)*.5

Xtr = scale(matrix(rnorm(n*p),n,p))
Ytr = Xtr%*%Btrue + matrix(rnorm(n),n,1)
Xts = scale(matrix(rnorm(n*p),n,p))
Yts = Xts%*%Btrue + matrix(rnorm(n),n,1)

fit = glmnet(x=Xtr,y=Ytr,family="gaussian",standardize=FALSE,nlambda=50,lambda.min.ratio=.001)
Yhtr = predict(fit,newx=Xtr)
MSEtr = apply((Yhtr-Ytr%*%matrix(1,1,length(fit$lambda)))^2,2,mean)
Yhts = predict(fit,newx=Xts)
MSEts = apply((Yhts-Yts%*%matrix(1,1,length(fit$lambda)))^2,2,mean)
```

```r
plot(1:length(fit$lambda),MSEtr,type="l",col=4,xlab="Lambda Index",ylab="Error")
lines(1:length(fit$lambda),MSEtr,col=4)
lines(1:length(fit$lambda),MSEts,col=2)
legend(30,5,legend=c("Training Error","Test Error"),col=c(4,2),lty=c(1,1),cex=.75)
```



3. Cross-Validation (R) Explain the concept of cross-validation and its importance in model assessment. Compare k-fold cross-validation and leave-one-out cross-validation. Perform cross-validation on your models to assess their predictive performance: A. K-Fold Cross-Validation: Implement K-fold cross-validation (choose an appropriate value of K) on your models and calculate the mean squared error (MSE) for each fold. Report the average MSE and discuss its significance. B. Leave-One-Out Cross-Validation (LOOCV): Apply LOOCV to evaluate the performance of your models. Compare the results with K-fold cross-validation and discuss the pros and cons of each method.

   Cross-validation is a nonparametric method to validate the data by reusing the data efficiently. Ideally there would be enough data to set aside a training set without affecting estimation accuracy, but we use cross-validation when there is not enough data. K-fold cross-validation randomly splits the data into K parts of about the same size and fits a model for each part using all the other parts. For each model, we find the prediction error on the kth part. Then we average all of the errors to find an estimate of the prediction error. This is a direct estimate of the error outside the sample. Leave-One-Out cross-validation uses all observations but one and repeats the process n times so that each observation was excluded one time. This method gives an approximately unbiased prediction error, but the variance can be high because the training sets are alike. It can also take a long time to run. In comparison, k-fold cross-validation usually has a low variance but could have a large bias. If the error is much higher for smaller training sets, then 5-fold or 10-fold cross-validation could give too-high prediction errors.

   The average training error for 5-fold CV is 1.044289 compared to the testing error of 1.40464. These values are similar, indicating that the model is a good fit. LOO cross-validation gives a training error of 0.8705967 and a testing error of 1.253238. Here it appears that 5-fold is better because the difference between the training and testing errors is smaller.

```
install.packages("leaps",repos = "http://cran.us.r-project.org")
```

```
## Warning: package 'leaps' is in use and will not be installed
```

```
install.packages("glmnet",repos = "http://cran.us.r-project.org")
```

```
## Warning: package 'glmnet' is in use and will not be installed
```

```
library(MASS)
library(leaps)
library(glmnet)

fold = 5
sam = sample(1:n,n)
CVerrs = NULL
for(i in 1:fold)
{
  ind = sam[((i-1)*n/fold + 1):(i*n/fold)]
  Xin = Xtr[-ind,]; Yin = Ytr[-ind]
  Xout = Xtr[ind,]; Yout = Ytr[ind]
  fit = glmnet(x=Xin,y=Yin,family="gaussian",standardize=FALSE,nlambda=50,lambda.min.ratio=.001)
  Yh = predict(fit,newx=Xout)
  CVerrs = cbind(CVerrs,apply((Yh-Yout%*%matrix(1,1,length(fit$lambda)))^2,2,mean))
}
CVerr = apply(CVerrs,1,mean)

#getting an error message here for xycoords not matching when I try to knit to pdf, but the code runs i
#plot(1:length(fit$lambda),MSEtr,type="l",col=4,xlab="Lambda Index",ylab="Error")
#lines(1:length(fit$lambda),MSEtr,col=4)
#lines(1:length(fit$lambda),MSEts,col=2)
#legend(30,5,legend=c("Training Error","Test Error"),col=c(4,2),lty=c(1,1),cex=.75)
#lines(1:length(fit$lambda),CVerr,col=1)

# test error
fit = glmnet(x=Xtr,y=Ytr,family="gaussian",standardize=FALSE,nlambda=50,lambda.min.ratio=.001)
optlam = fit$lambda[which.min(CVerr)]
optlam
```

```
## [1] 0.0404109
```

```
fit = glmnet(x=Xtr,y=Ytr,family="gaussian",standardize=FALSE,lambda=optlam)
Yhtr = predict(fit,newx=Xtr)
TRerr = mean( (Yhtr - Ytr)^2 )
Yhts = predict(fit,newx=Xts)
TSerr = mean( (Yhts - Yts)^2 )
sum(fit$beta!=0)
```

```
## [1] 36
```

```
TRerr
```

```
## [1] 0.4785626
```

```r
TSerr
```

```
## [1] 1.433029
```

```r
#now LOO
fold = n
sam = sample(1:n,n)
CVerrs = NULL
for(i in 1:fold)
{
  ind = sam[((i-1)*n/fold + 1):(i*n/fold)]
  Xin = Xtr[-ind,]; Yin = Ytr[-ind]
  Xout = Xtr[ind,]; Yout = Ytr[ind]
  fit = glmnet(x=Xin,y=Yin,family="gaussian",standardize=FALSE,nlambda=50,lambda.min.ratio=.001)
  Yh = predict(fit,newx=Xout)
  CVerrs = cbind(CVerrs,apply((Yh-Yout%*%matrix(1,1,length(fit$lambda)))^2,2,mean))
}
```

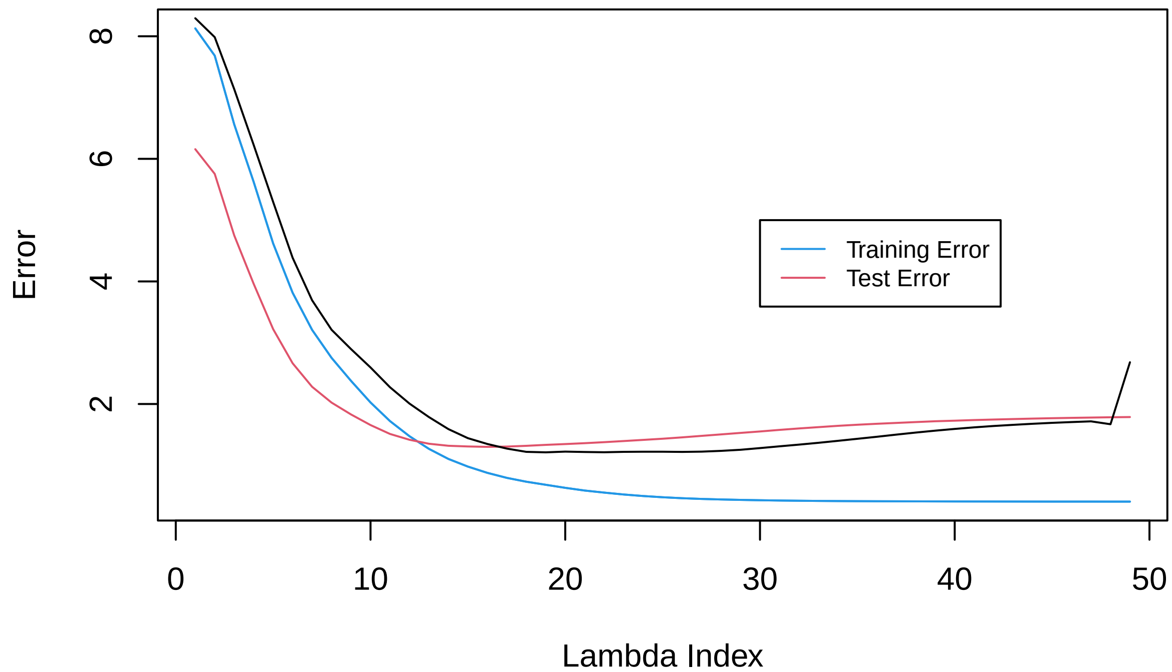
```
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
```

```
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
```

```
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
```

```
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
## Warning in cbind(CVerrs, apply((Yh - Yout %*% matrix(1, 1,
## length(fit$lambda)))^2, : number of rows of result is not a multiple of vector
## length (arg 2)
```

```r
CVerr = apply(CVerrs,1,mean)

plot(1:length(fit$lambda),MSEtr,type="l",col=4,xlab="Lambda Index",ylab="Error")
lines(1:length(fit$lambda),MSEtr,col=4)
lines(1:length(fit$lambda),MSEts,col=2)
legend(30,5,legend=c("Training Error","Test Error"),col=c(4,2),lty=c(1,1),cex=.75)
lines(1:length(fit$lambda),CVerr,col=1)
```




```
fit = glmnet(x=Xtr,y=Ytr,family="gaussian",standardize=FALSE,nlambda=50,lambda.min.ratio=.001)
optlam = fit$lambda[which.min(CVerr)]
optlam
```

```
## [1] 0.09415545
```

```
fit = glmnet(x=Xtr,y=Ytr,family="gaussian",standardize=FALSE,lambda=optlam)
Yhtr = predict(fit,newx=Xtr)
TRerr = mean( (Yhtr - Ytr)^2 )
Yhts = predict(fit,newx=Xts)
TSerr = mean( (Yhts - Yts)^2 )
sum(fit$beta!=0)
```

```
## [1] 25
```

```
TRerr
```

```
## [1] 0.682088
```

```
TSerr
```

```
## [1] 1.333516
```