Statistical Computing Homework 5

110024516 邱繼賢

Problem 2.

```
library(ISLR2)
library(latex2exp)
library(fields)
library(GpGp)
library(knitr)
```

資料前處理：

1. 將所有的 count variables 視為 approximately continuous variables，並去除三個 categorical variables *League, Division, NewLeague*

2. 將變數 *Salary* 中有缺失值的資料刪除，僅剩下 263 筆資料

3. 將資料以 8:2 的比例隨機區分為 training data 和 testing data

```
data("Hitters")
hit_data = Hitters[,-c(14,15,20)]
idx.na = which(is.na(hit_data$Salary))
hit_data = hit_data[-idx.na,]
set.seed(1239)
idx = sample(1:263,210)
hit_training = hit_data[idx,]
hit_testing = hit_data[-idx,]
```

以變數 *Salary* 為反應變數 Y，其餘變數為解釋變數 X，然後對 training data fit linear model

$$Y = X\beta + \epsilon$$

```r
model_lm = lm(Salary ~ ., hit_training)
summary(model_lm)
```

```
##
## Call:
## lm(formula = Salary ~ ., data = hit_training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -856.47 -180.22  -41.06  124.41 1900.67
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 199.88332  100.70688   1.985  0.04858 *
## AtBat        -2.47310    0.75429  -3.279  0.00124 **
## Hits          8.07398    2.85378   2.829  0.00516 **
## HmRun         1.11320    7.29746   0.153  0.87892
## Runs         -1.81743    3.61622  -0.503  0.61583
## RBI           0.41774    3.07001   0.136  0.89191
## Walks         6.23485    2.14208   2.911  0.00403 **
## Years       -10.78019   14.76273  -0.730  0.46614
## CAtBat       -0.09523    0.15547  -0.613  0.54092
## CHits        -0.31098    0.82309  -0.378  0.70598
## CHmRun       -1.30655    1.89373  -0.690  0.49106
## CRuns         1.86603    0.92501   2.017  0.04505 *
## CRBI          1.22291    0.80034   1.528  0.12815
## CWalks       -1.01345    0.38907  -2.605  0.00991 **
## PutOuts       0.28897    0.08707   3.319  0.00108 **
## Assists       0.35583    0.25313   1.406  0.16141
## Errors       -2.91376    5.03581  -0.579  0.56353
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 330.7 on 193 degrees of freedom
## Multiple R-squared:  0.502,  Adjusted R-squared:  0.4607
## F-statistic: 12.16 on 16 and 193 DF,  p-value: < 2.2e-16
```

Fit training data by GP model with Matern covariance family

```
model_gp <- fit_model(y=hit_training$Salary, locs=hit_training[,-17],
                      covfun_name="matern15_isotropic", m_seq=c(10,30))
```

```
## Design matrix not specified, using constant mean
## Reordering...Done
## Finding nearest neighbors...Done
## Iter 0:
## pars =  202726.0916 673.434 0.1
## loglik = -1526.074773
## grad = 102.086 -108.29 41.292
##
## Iter 1:
## pars =  344298.9699 216.0025 0.0598
## loglik = -1507.305579
## grad = 54.2641 -63.8604 14.4418
## step dot grad =  -156.0051
##
## Iter 2:
## pars =  292233.0589 267.6008 0.0275
## loglik = -1479.576425
## grad = 26.7153 -43.3927 8.1353
## step dot grad =  -33.80771
##
## Iter 3:
## pars =  330723.3115 317.4482 0.0089
## loglik = -1469.778714
## grad = 11.4588 -23.7809 3.2999
## step dot grad =  -13.32636
##
## Iter 4:
## pars =  374591.4918 345.0651 0.0026
## loglik = -1466.712598
## grad = 5.9373 -13.9144 1.1949
## step dot grad =  -4.622058
##
```

```
## Iter 5:
## pars =  399653.7755 359.3586 7e-04
## loglik = -1465.575894
## grad = 3.1941 -8.1896 0.3897
## step dot grad =  -1.800357
##
## Iter 6:
## pars =  414621.8386 369.7337 2e-04
## loglik = -1465.21309
## grad = 1.1408 -3.6625 0.0771
## step dot grad =  -0.621103
##
## Iter 7:
## pars =  426549.387 377.54 1e-04
## loglik = -1465.159968
## grad = 0.1269 -0.9943 0.0137
## step dot grad =  -0.0839667
##
## Iter 8:
## pars =  434013.9095 381.427 1e-04
## loglik = -1465.153424
## grad = 0.0142 -0.3395 0.0047
## step dot grad =  -0.009390819
##
## Iter 9:
## pars =  437288.7048 383.0402 1e-04
## loglik = -1465.152376
## grad = 0.0044 -0.1356 0.0018
## step dot grad =  -0.001494075
##
## Iter 10:
## pars =  438637.761 383.6998 1e-04
## loglik = -1465.152203
## grad = 0.0016 -0.0549 7e-04
## step dot grad =  -0.0002450838
##
```

```
## Iter 11:
## pars =  439189.2543 383.9687 1e-04
## loglik = -1465.152175
## grad = 6e-04 -0.0223 3e-04
## step dot grad =  -4.05192e-05
##
## Iter 0:
## pars =  439189.2543 383.9687 1e-04
## loglik = -1463.932781
## grad = 3.36 -6.751 -0.005
##
## Iter 1:
## pars =  448269.7588 381.5312 1e-04
## loglik = -1463.877618
## grad = 0.07 0.0193 -0.0014
## step dot grad =  -0.1124554
##
## Iter 2:
## pars =  446423.8111 380.8281 1e-04
## loglik = -1463.877388
## grad = -0.0014 0.0628 -7e-04
## step dot grad =  -0.0003407898
##
## Iter 3:
## pars =  445809.5079 380.5458 1e-04
## loglik = -1463.877355
## grad = -6e-04 0.0254 -3e-04
## step dot grad =  -4.797848e-05
```

分別計算兩模型對 testing data 的測值

```
hit_testing$y.hat_lm = predict(model_lm, hit_testing[,1:16])
hit_testing$y.hat_gp = predictions(model_gp, locs_pred = as.matrix(hit_testing[,1:16]),X_pred = rep(1,53
kable(hit_testing[,17:19], col.names = c("Y", "$\\hat{Y}_{LM}$", "$\\hat{Y}_{GP}$"))
```

	Y	$\hat{Y}_{LM}$	$\hat{Y}_{GP}$
-Al Newman	70.000	167.7968	459.62051
-Alan Wiggins	700.000	331.1380	486.79891
-Buddy Bell	775.000	1076.6208	547.48363
-Bill Buckner	776.667	1311.2487	1197.55227
-Bo Diaz	750.000	558.8017	691.38021
-Chris Brown	215.000	439.2963	240.05942
-Curt Ford	70.000	234.4679	41.62539
-Carney Lansford	1200.000	774.4431	436.26937
-Chet Lemon	675.000	688.6940	533.68109
-Carmelo Martinez	340.000	283.2126	217.89501
-Don Baylor	950.000	1071.1114	1094.31344
-Dale Murphy	1900.000	963.9780	830.31644
-Dan Pasqua	110.000	462.9875	57.13934
-Dale Sveum	70.000	185.9696	48.88973
-Gary Carter	1925.571	1086.7192	1688.81120
-Garth Iorg	362.500	302.1291	270.20923
-Hubie Brooks	750.000	627.7705	262.93707
-John Cangelosi	100.000	338.5416	94.54587
-Jack Howell	95.000	278.4053	246.12449
-John Kruk	110.000	465.9348	-29.67965
-Jerry Mumphrey	600.000	625.2825	334.70321
-John Russell	155.000	301.6383	211.53796
-Joel Skinner	110.000	122.8173	81.55392
-Kevin Bass	630.000	595.8458	418.09830
-Kirk Gibson	1300.000	715.2145	709.96196
-Ken Landreaux	737.500	503.5638	478.63490
-Kevin Mitchell	125.000	323.8986	23.52764
-Ken Oberkfell	725.000	522.4469	717.44279
-Kurt Stillwell	75.000	227.4226	195.44914
-Len Dykstra	202.500	537.0587	165.97863
-Lee Lacy	525.000	575.9275	653.75303
-Lloyd Moseby	787.500	676.0962	1009.63196
-Lance Parrish	800.000	852.8698	658.28957
-Mark Salas	137.000	258.1642	163.60521

	Y	$\hat{Y}_{LM}$	$\hat{Y}_{GP}$
-Mike Scioscia	875.000	399.9872	437.94294
-Milt Thompson	140.000	211.5667	52.23525
-Marvell Wynne	240.000	261.4628	383.11853
-Ozzie Guillen	175.000	224.7722	209.90809
-Rafael Belliard	130.000	202.8450	80.89582
-Ron Hassey	560.000	536.0629	323.06171
-Ron Oester	750.000	476.9300	519.13732
-Robin Yount	1000.000	1371.1452	864.95953
-Scott Bradley	90.000	312.3073	47.90010
-Scott Fletcher	475.000	538.1461	465.87844
-Terry Harper	425.000	239.6747	267.14048
-Tommy Herr	925.000	742.7886	512.34666
-Tito Landrum	286.667	172.0393	228.09549
-Tom Paciorek	235.000	452.1622	515.41264
-Terry Pendleton	160.000	233.8078	235.82093
-Tony Phillips	425.000	565.5673	468.87993
-Wally Backman	550.000	520.5808	456.44321
-Wade Boggs	1600.000	1201.4436	706.60862
-Willie Wilson	1000.000	756.4029	533.51173

計算各別的 standard error

$$\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(\hat{Y}_i - Y_i\right)^2}$$

For LM model

```
se_lm = sqrt(1/52*sum((hit_testing$y.hat_lm-hit_testing$Salary)^2))
se_lm
```

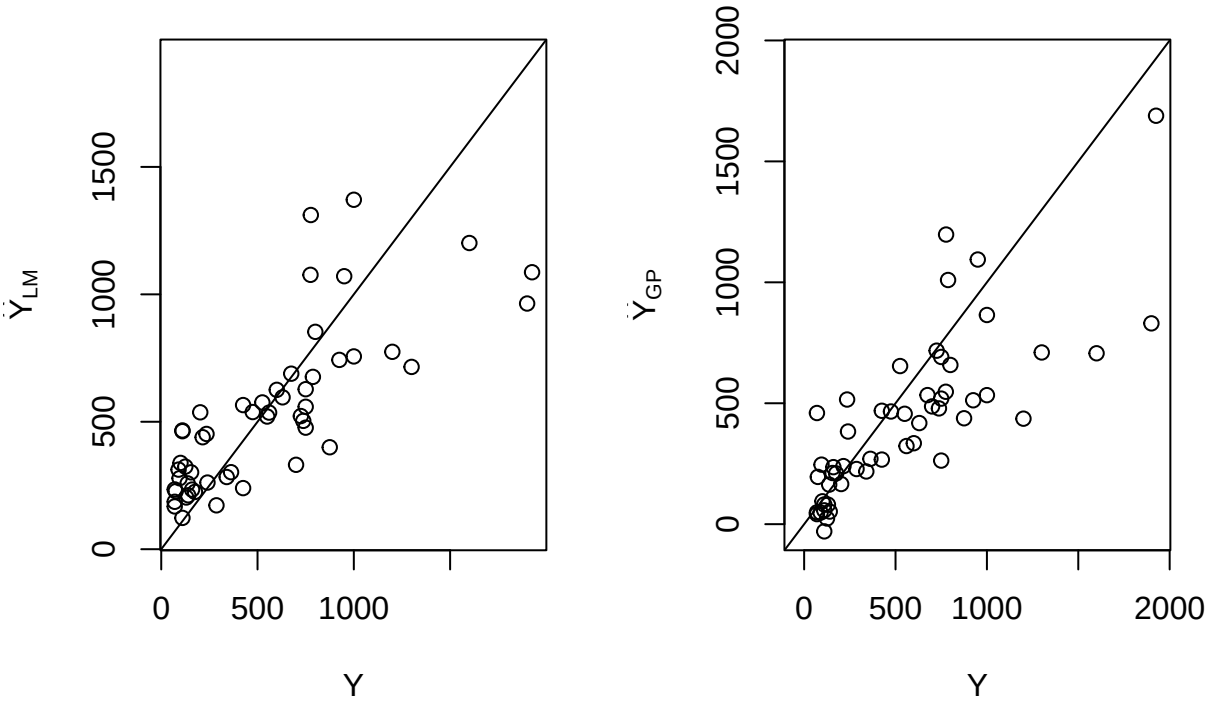
```
## [1] 288.729
```

For GP model

```
se_gp = sqrt(1/52*sum((hit_testing$y.hat_gp-hit_testing$Salary)^2))
se_gp
```

```
## [1] 306.0555
```

```r
par(mfrow = c(1,2))
plot(hit_testing$Salary,hit_testing$y.hat_lm, xlab = "Y", ylab = TeX("$\\hat{Y}_{LM}$"),
     xlim = c(min(hit_testing$y.hat_lm,hit_testing$Salary),max(hit_testing$y.hat_lm,hit_testing$Salary))
     ylim = c(min(hit_testing$y.hat_lm,hit_testing$Salary),max(hit_testing$y.hat_lm,hit_testing$Salary))
abline(0,1)
plot(hit_testing$Salary,hit_testing$y.hat_gp, xlab = "Y", ylab = TeX("$\\hat{Y}_{GP}$"),
     xlim = c(min(hit_testing$y.hat_gp,hit_testing$Salary),max(hit_testing$y.hat_gp,hit_testing$Salary))
     ylim = c(min(hit_testing$y.hat_gp,hit_testing$Salary),max(hit_testing$y.hat_gp,hit_testing$Salary))
abline(0,1)
```



我們可以看到兩模型都可以大致捕捉到變數 Y 的趨勢，但都不是 fit 的非常好，也許可以嘗試看看更為複雜的模型