PREDICTIVE PROBLEM SET 2

Deeptarka Saha 732

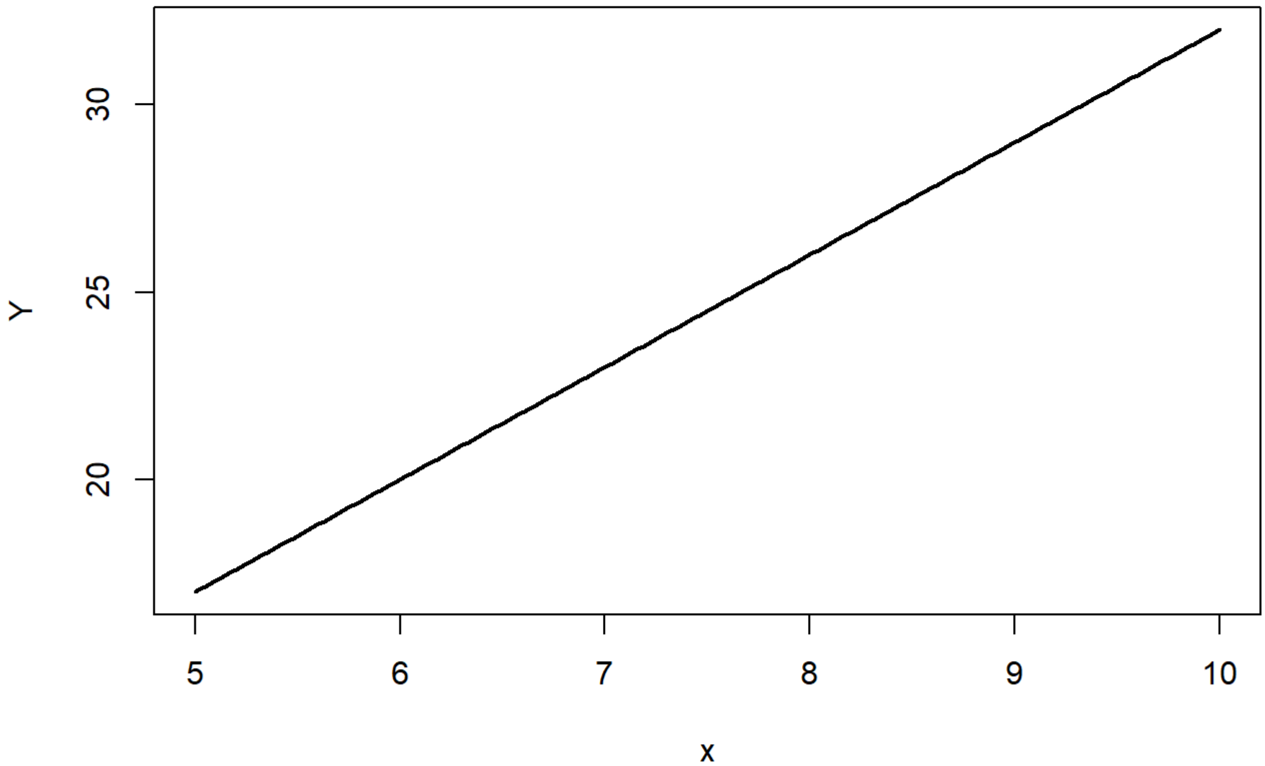
2026-01-30

Qs 1

```
rm(list=ls())


#Step 1: Graph the population regression line
x=seq(from=5,to=10,length.out=200);x
##   [1] 5.000000 5.025126 5.050251 5.075377 5.100503 5.125628 5.150754
##   [8] 5.175879 5.201005 5.226131 5.251256 5.276382 5.301508 5.326633
##  [15] 5.351759 5.376884 5.402010 5.427136 5.452261 5.477387 5.502513
##  [22] 5.527638 5.552764 5.577889 5.603015 5.628141 5.653266 5.678392
##  [29] 5.703518 5.728643 5.753769 5.778894 5.804020 5.829146 5.854271
##  [36] 5.879397 5.904523 5.929648 5.954774 5.979899 6.005025 6.030151
##  [43] 6.055276 6.080402 6.105528 6.130653 6.155779 6.180905 6.206030
##  [50] 6.231156 6.256281 6.281407 6.306533 6.331658 6.356784 6.381910
##  [57] 6.407035 6.432161 6.457286 6.482412 6.507538 6.532663 6.557789
##  [64] 6.582915 6.608040 6.633166 6.658291 6.683417 6.708543 6.733668
##  [71] 6.758794 6.783920 6.809045 6.834171 6.859296 6.884422 6.909548
##  [78] 6.934673 6.959799 6.984925 7.010050 7.035176 7.060302 7.085427
##  [85] 7.110553 7.135678 7.160804 7.185930 7.211055 7.236181 7.261307
##  [92] 7.286432 7.311558 7.336683 7.361809 7.386935 7.412060 7.437186
##  [99] 7.462312 7.487437 7.512563 7.537688 7.562814 7.587940 7.613065
## [106] 7.638191 7.663317 7.688442 7.713568 7.738693 7.763819 7.788945
## [113] 7.814070 7.839196 7.864322 7.889447 7.914573 7.939698 7.964824
## [120] 7.989950 8.015075 8.040201 8.065327 8.090452 8.115578 8.140704
## [127] 8.165829 8.190955 8.216080 8.241206 8.266332 8.291457 8.316583
## [134] 8.341709 8.366834 8.391960 8.417085 8.442211 8.467337 8.492462
## [141] 8.517588 8.542714 8.567839 8.592965 8.618090 8.643216 8.668342
## [148] 8.693467 8.718593 8.743719 8.768844 8.793970 8.819095 8.844221
## [155] 8.869347 8.894472 8.919598 8.944724 8.969849 8.994975 9.020101
## [162] 9.045226 9.070352 9.095477 9.120603 9.145729 9.170854 9.195980
## [169] 9.221106 9.246231 9.271357 9.296482 9.321608 9.346734 9.371859
## [176] 9.396985 9.422111 9.447236 9.472362 9.497487 9.522613 9.547739
## [183] 9.572864 9.597990 9.623116 9.648241 9.673367 9.698492 9.723618
```

```
## [190]  9.748744  9.773869  9.798995  9.824121  9.849246  9.874372  9.899497
## [197]  9.924623  9.949749  9.974874 10.000000
Y=2+(3*x)
plot(x,Y,type='l',lwd=2)
```

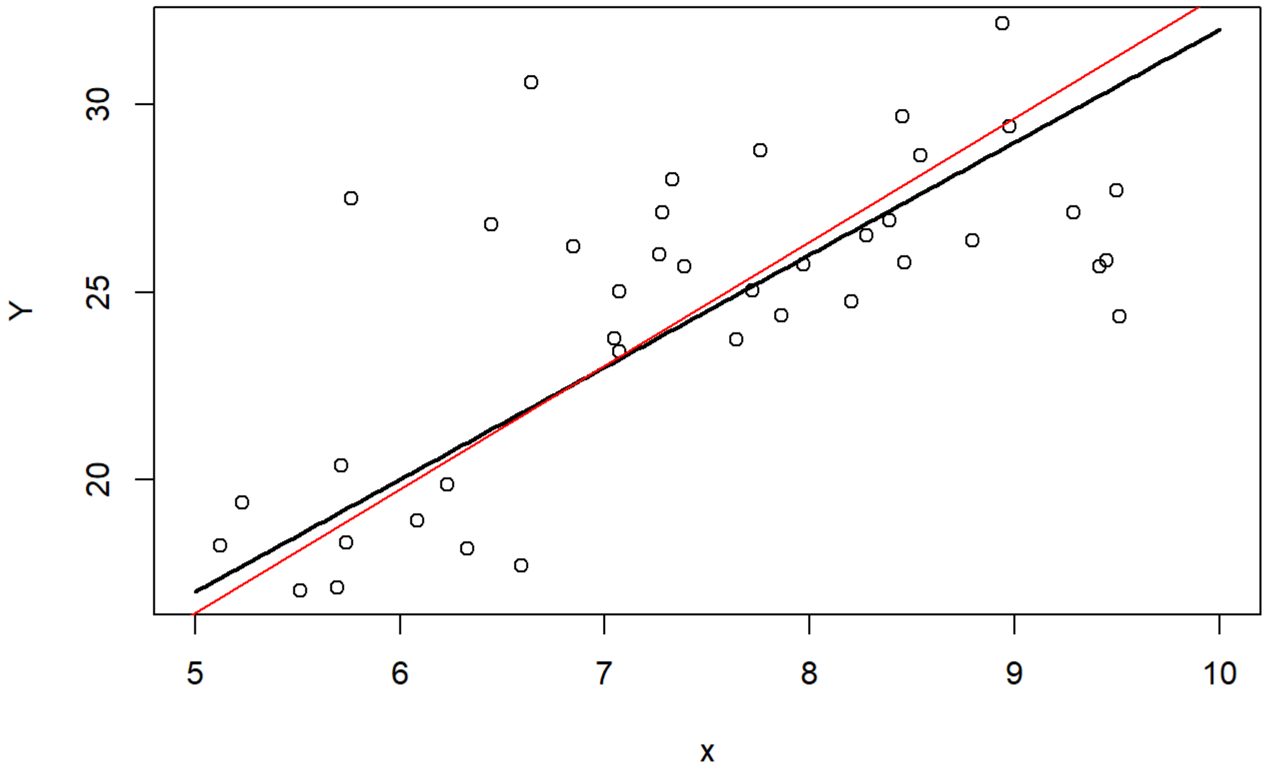



```
# Step 2
n=50
set.seed(123)
xi=runif(n,5,10);xi
##  [1] 6.437888 8.941526 7.044885 9.415087 9.702336 5.227782 7.640527 9.462095
##  [9] 7.757175 7.283074 9.784167 7.266671 8.387853 7.863167 5.514623 9.499125
## [17] 6.230439 5.210298 6.639604 9.772518 9.447697 8.464017 8.202534 9.971349
## [25] 8.278529 8.542652 7.720330 7.970710 6.445799 5.735568 9.815121 9.511495
## [33] 8.453526 8.977337 5.123068 7.388980 8.792298 6.082040 6.590905 6.158129
## [41] 5.714000 7.072732 7.068622 6.844227 5.762224 5.694030 6.165170 7.329812
## [49] 6.329863 9.289139
ei=rnorm(n,0,4);ei
##  [1] -6.7467732  3.3511482  0.6134925 -4.5525477  5.0152597  1.7058569
##  [7] -1.1802859  3.5805026  3.5125340  3.2863243  2.7545610  2.2156706
```

```
## [13] -0.2476468 -1.2238507 -1.5218840 -2.7788279 -0.8316691 -5.0615854
## [19]  8.6758239  4.8318480 -4.4924343 -1.6115393 -1.8666214  3.1198605
## [25] -0.3334763  1.0132741 -0.1141870 -0.1714818  5.4744091 -0.9030839
## [31]  6.0658824 -6.1950112  2.3384550  0.4954170  0.8637663  1.5185579
## [37] -2.0092938 -1.3328295 -4.0743015 -4.2871649  1.2141146  1.7928391
## [43]  0.2120169  3.6890699  8.2003387 -1.9641247 -9.2366755  4.0229541
## [49] -2.8368031 -2.7520345
#Step 3:
# Graph the population regression line
x=seq(from=5,to=10,length.out=200);x
##   [1]  5.000000  5.025126  5.050251  5.075377  5.100503  5.125628  5.150754
##   [8]  5.175879  5.201005  5.226131  5.251256  5.276382  5.301508  5.326633
##  [15]  5.351759  5.376884  5.402010  5.427136  5.452261  5.477387  5.502513
##  [22]  5.527638  5.552764  5.577889  5.603015  5.628141  5.653266  5.678392
##  [29]  5.703518  5.728643  5.753769  5.778894  5.804020  5.829146  5.854271
##  [36]  5.879397  5.904523  5.929648  5.954774  5.979899  6.005025  6.030151
##  [43]  6.055276  6.080402  6.105528  6.130653  6.155779  6.180905  6.206030
##  [50]  6.231156  6.256281  6.281407  6.306533  6.331658  6.356784  6.381910
##  [57]  6.407035  6.432161  6.457286  6.482412  6.507538  6.532663  6.557789
##  [64]  6.582915  6.608040  6.633166  6.658291  6.683417  6.708543  6.733668
##  [71]  6.758794  6.783920  6.809045  6.834171  6.859296  6.884422  6.909548
##  [78]  6.934673  6.959799  6.984925  7.010050  7.035176  7.060302  7.085427
##  [85]  7.110553  7.135678  7.160804  7.185930  7.211055  7.236181  7.261307
##  [92]  7.286432  7.311558  7.336683  7.361809  7.386935  7.412060  7.437186
##  [99]  7.462312  7.487437  7.512563  7.537688  7.562814  7.587940  7.613065
## [106]  7.638191  7.663317  7.688442  7.713568  7.738693  7.763819  7.788945
## [113]  7.814070  7.839196  7.864322  7.889447  7.914573  7.939698  7.964824
## [120]  7.989950  8.015075  8.040201  8.065327  8.090452  8.115578  8.140704
## [127]  8.165829  8.190955  8.216080  8.241206  8.266332  8.291457  8.316583
## [134]  8.341709  8.366834  8.391960  8.417085  8.442211  8.467337  8.492462
## [141]  8.517588  8.542714  8.567839  8.592965  8.618090  8.643216  8.668342
## [148]  8.693467  8.718593  8.743719  8.768844  8.793970  8.819095  8.844221
## [155]  8.869347  8.894472  8.919598  8.944724  8.969849  8.994975  9.020101
## [162]  9.045226  9.070352  9.095477  9.120603  9.145729  9.170854  9.195980
## [169]  9.221106  9.246231  9.271357  9.296482  9.321608  9.346734  9.371859
## [176]  9.396985  9.422111  9.447236  9.472362  9.497487  9.522613  9.547739
## [183]  9.572864  9.597990  9.623116  9.648241  9.673367  9.698492  9.723618
## [190]  9.748744  9.773869  9.798995  9.824121  9.849246  9.874372  9.899497
## [197]  9.924623  9.949749  9.974874 10.000000
```

```
Y=2+(3*x)
plot(x,Y,type='l',lwd=2)
set.seed(123)
n=50
xi=runif(n,5,10);xi
##  [1] 6.437888 8.941526 7.044885 9.415087 9.702336 5.227782 7.640527 9.462095
##  [9] 7.757175 7.283074 9.784167 7.266671 8.387853 7.863167 5.514623 9.499125
## [17] 6.230439 5.210298 6.639604 9.772518 9.447697 8.464017 8.202534 9.971349
## [25] 8.278529 8.542652 7.720330 7.970710 6.445799 5.735568 9.815121 9.511495
## [33] 8.453526 8.977337 5.123068 7.388980 8.792298 6.082040 6.590905 6.158129
## [41] 5.714000 7.072732 7.068622 6.844227 5.762224 5.694030 6.165170 7.329812
## [49] 6.329863 9.289139
ei=rnorm(n,0,4);ei
##  [1] -6.7467732  3.3511482  0.6134925 -4.5525477  5.0152597  1.7058569
##  [7] -1.1802859  3.5805026  3.5125340  3.2863243  2.7545610  2.2156706
## [13] -0.2476468 -1.2238507 -1.5218840 -2.7788279 -0.8316691 -5.0615854
## [19]  8.6758239  4.8318480 -4.4924343 -1.6115393 -1.8666214  3.1198605
## [25] -0.3334763  1.0132741 -0.1141870 -0.1714818  5.4744091 -0.9030839
## [31]  6.0658824 -6.1950112  2.3384550  0.4954170  0.8637663  1.5185579
## [37] -2.0092938 -1.3328295 -4.0743015 -4.2871649  1.2141146  1.7928391
## [43]  0.2120169  3.6890699  8.2003387 -1.9641247 -9.2366755  4.0229541
## [49] -2.8368031 -2.7520345
yi=2+(3*xi)+ei;yi
##  [1] 14.56689 32.17573 23.74815 25.69271 36.12227 19.38920 23.74130 33.96679
##  [9] 28.78406 27.13555 34.10706 26.01568 26.91591 24.36565 17.02199 27.71855
## [17] 19.85965 12.56931 30.59463 36.14940 25.85066 25.78051 24.74098 35.03391
## [25] 26.50211 28.64123 25.04680 25.74065 26.81181 18.30362 37.51125 24.33947
## [33] 29.69903 29.42743 18.23297 25.68550 26.36760 18.91329 17.69841 16.18722
## [41] 20.35611 25.01103 23.41788 26.22175 27.48701 17.11797 11.25884 28.01239
## [49] 18.15279 27.11538
points(xi,yi)
lin_reg=lm(yi~xi);lin_reg
##
## Call:
## lm(formula = yi ~ xi)
##
## Coefficients:
## (Intercept)           xi
##    -0.09639      3.30540
```

```
coef(lin_reg)
## (Intercept)          xi
## -0.09638929  3.30539569
abline(lin_reg,col="red")
```

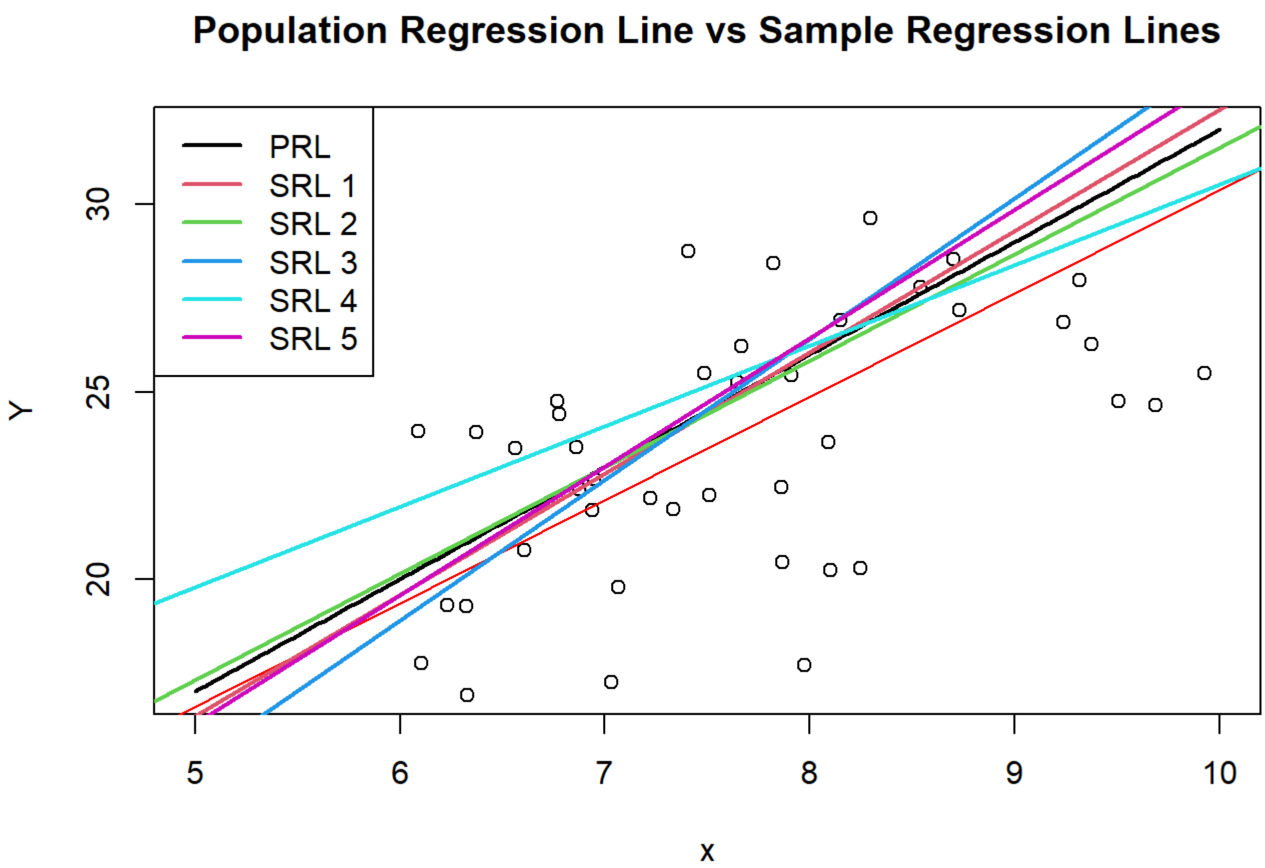



```
#Step 4:
x=seq(from=5,to=10,length.out=200);x
##   [1] 5.000000 5.025126 5.050251 5.075377 5.100503 5.125628 5.150754
##   [8] 5.175879 5.201005 5.226131 5.251256 5.276382 5.301508 5.326633
##  [15] 5.351759 5.376884 5.402010 5.427136 5.452261 5.477387 5.502513
##  [22] 5.527638 5.552764 5.577889 5.603015 5.628141 5.653266 5.678392
##  [29] 5.703518 5.728643 5.753769 5.778894 5.804020 5.829146 5.854271
##  [36] 5.879397 5.904523 5.929648 5.954774 5.979899 6.005025 6.030151
##  [43] 6.055276 6.080402 6.105528 6.130653 6.155779 6.180905 6.206030
##  [50] 6.231156 6.256281 6.281407 6.306533 6.331658 6.356784 6.381910
##  [57] 6.407035 6.432161 6.457286 6.482412 6.507538 6.532663 6.557789
##  [64] 6.582915 6.608040 6.633166 6.658291 6.683417 6.708543 6.733668
##  [71] 6.758794 6.783920 6.809045 6.834171 6.859296 6.884422 6.909548
##  [78] 6.934673 6.959799 6.984925 7.010050 7.035176 7.060302 7.085427
```

```
##  [85]  7.110553  7.135678  7.160804  7.185930  7.211055  7.236181  7.261307
##  [92]  7.286432  7.311558  7.336683  7.361809  7.386935  7.412060  7.437186
##  [99]  7.462312  7.487437  7.512563  7.537688  7.562814  7.587940  7.613065
## [106]  7.638191  7.663317  7.688442  7.713568  7.738693  7.763819  7.788945
## [113]  7.814070  7.839196  7.864322  7.889447  7.914573  7.939698  7.964824
## [120]  7.989950  8.015075  8.040201  8.065327  8.090452  8.115578  8.140704
## [127]  8.165829  8.190955  8.216080  8.241206  8.266332  8.291457  8.316583
## [134]  8.341709  8.366834  8.391960  8.417085  8.442211  8.467337  8.492462
## [141]  8.517588  8.542714  8.567839  8.592965  8.618090  8.643216  8.668342
## [148]  8.693467  8.718593  8.743719  8.768844  8.793970  8.819095  8.844221
## [155]  8.869347  8.894472  8.919598  8.944724  8.969849  8.994975  9.020101
## [162]  9.045226  9.070352  9.095477  9.120603  9.145729  9.170854  9.195980
## [169]  9.221106  9.246231  9.271357  9.296482  9.321608  9.346734  9.371859
## [176]  9.396985  9.422111  9.447236  9.472362  9.497487  9.522613  9.547739
## [183]  9.572864  9.597990  9.623116  9.648241  9.673367  9.698492  9.723618
## [190]  9.748744  9.773869  9.798995  9.824121  9.849246  9.874372  9.899497
## [197]  9.924623  9.949749  9.974874 10.000000
Y=2+(3*x)
plot(x,Y,type='l',lwd=2)
n=50
xi=runif(n,5,10);xi
##  [1] 9.237266 7.487636 6.939545 6.232245 5.555482 6.949972 7.859677 6.084464
##  [9] 7.223840 6.089953 7.511498 6.769523 8.249926 6.873570 6.777227 7.668440
## [17] 8.701672 6.105515 7.063731 6.328433 8.149865 5.919142 9.318221 8.732840
## [25] 8.341423 8.090089 6.861190 7.649178 9.373412 7.908750 9.198839 6.562241
## [33] 8.541452 6.325089 7.971716 7.406449 6.325164 7.822952 9.565941 9.509372
## [41] 6.370833 6.607414 9.928204 8.099967 9.686570 7.332664 7.034163 8.296152
## [49] 5.761733 7.864335
ei=rnorm(n,0,4);ei
##  [1] -2.8416263  1.0275348 -0.9867675 -1.3901704 -3.8064743 -0.1801109
##  [7] -3.1396179 -6.6717677 -1.5209061  3.6759864 -2.3013879  2.4318573
## [13] -6.4715308 -0.2222479  2.0776288  1.2046134  0.4227048 -2.5628240
## [19] -3.3988174 -4.0965152  0.4705864 -3.7898985 -1.9622298 -1.0243688
## [25]  7.3754480 -2.6077996  0.9415463  0.3118434 -3.8474265 -0.2852323
## [31]  5.7782034  1.8060162  0.1649317 -1.6899873 -8.2129889  4.5253489
## [37] -5.8425603  2.9597900  7.6364143 -5.7755726  2.8071373 -1.0487900
## [43] -6.2885766 -6.0586706 -6.4061447 -2.1236261 -5.8470223  2.7516671
## [49]  8.4004358 -5.1481219
yi=2+(3*xi)+ei;yi
```

```
##  [1] 26.87017 25.49044 21.83187 19.30656 14.85997 22.66981 22.43941 13.58162
##  [9] 22.15061 23.94585 22.23311 24.74043 20.27825 22.39846 24.40931 26.20993
## [17] 28.52772 17.75372 19.79237 16.88879 26.92018 15.96753 27.99243 27.17415
## [25] 34.39972 23.66247 23.52512 25.25938 26.27281 25.44102 35.37472 23.49274
## [33] 27.78929 19.28528 17.70216 28.74470 15.13293 28.42865 38.33424 24.75254
## [41] 23.91964 20.77345 25.49604 20.24123 24.65357 21.87436 17.25547 29.64012
## [49] 27.68564 20.44488
points(xi,yi)
lin_reg=lm(yi~xi);lin_reg
##
## Call:
## lm(formula = yi ~ xi)
##
## Coefficients:
## (Intercept)           xi
##       2.792        2.761
coef(lin_reg)
## (Intercept)          xi
##    2.792188    2.761042
abline(lin_reg,col="red")
set.seed(123)
xr=matrix(runif(n*5,5,10),ncol=5,nrow=n)
er=matrix(rnorm(n*5,0,4),ncol=5,nrow=n)
yr=matrix(0,ncol=5,nrow=n)
intercept=rep(0,5)
slope=rep(0,5)
for(i in 1:5)
{
  yr[,i]=2+(3*xr[,i])+er[,i]
  lin_reg=lm(yr[,i]~xr[,i])
  intercept[i]=coef(lin_reg)[1]
  slope[i]=coef(lin_reg)[2]
  abline(lin_reg,col=i+1,lwd=2)
}
Estimates=data.frame(intercept,slope);Estimates
##    intercept    slope
## 1  0.1547357 3.238639
## 2  3.1053005 2.842175
## 3 -3.6350484 3.756428
```

```
## 4  9.0589133 2.147608
## 5 -1.0229227 3.432290
legend('topleft',
       legend=c("PRL",
                "SRL 1",
                "SRL 2",
                "SRL 3",
                "SRL 4",
                "SRL 5"),
       col=1:6,
       lwd=2)
title(main="Population Regression Line vs Sample Regression Lines")
```



The differences between the 5 lines represent estimation error due to:

i. Random noise

ii)Finite sample space

OVerall the population regression line is fixed,but the least squares regression line is a random line because it depends on the random sample.

Qs 2

```
rm(list=ls())


#Step 1:
set.seed(123)
n=50
xi=runif(n,5,10)
ei=rnorm(n)
mean_xi=mean(xi)
xii=xi-mean_xi;xii
##  [1] -1.16256726  1.34107081 -0.55557025  1.81463216  2.10188156 -2.37267237
##  [7]  0.04007258  1.86164036  0.15672021 -0.31738119  2.18371186 -0.33378408
## [13]  0.78739831  0.26271215 -2.08583145  1.89866999 -1.37001619 -2.39015720
## [19] -0.96085127  2.17206338  1.84724172  0.86356217  0.60207921  2.37089402
## [25]  0.67807413  0.94219748  0.11987526  0.37025524 -1.15465618 -1.86488663
## [31]  2.21466630  1.91104036  0.85307153  1.37688223 -2.47738644 -0.21147501
## [37]  1.19184282 -1.51841518 -1.00954983 -1.44232594 -1.88645475 -0.52772318
## [43] -0.53183323 -0.75622761 -1.83823112 -1.90642455 -1.43528437 -0.27064261
## [49] -1.27059166  1.68868371
yi=2+3*xii+ei;yi
##  [1] -3.17439510  6.86099949  0.48666236  6.30575953  9.55945960 -4.69155288
##  [7]  1.82514625  8.48004674  3.34829411  1.86943752  9.23977584  1.55256541
## [13]  4.30028323  2.48217378 -4.63796535  7.00130299 -2.31796585 -6.43586794
## [19]  1.28640216  9.72415215  6.41861657  4.18780167  3.33958226  9.89264718
## [25]  3.95085333  5.07991095  2.33107903  3.06789526 -0.09536625 -3.82043087
## [31] 10.16046950  6.18436828  5.14382834  6.25450092 -5.21621775  1.74521446
## [37]  5.07320502 -2.88845294 -2.04722486 -3.39876904 -3.35583561  0.86504023
## [43]  0.45750453  0.65358464 -1.46460869 -4.21030480 -4.61502197  2.19381069
## [49] -2.52097575  6.37804252
#Step 2:
lin_reg=lm(yi~xii);lin_reg
##
## Call:
## lm(formula = yi ~ xii)
##
## Coefficients:
## (Intercept)          xii
##       2.056        3.076
```

```
coef(lin_reg)
## (Intercept)         xii
##    2.056189    3.076349
#Step 3:
b0=seq(1,3,by=0.000001)
length(b0)
## [1] 2000001
b1=seq(3,4,by=0.000001)
RSS=rep(0,2000001)
for(i in 1:2000001)
{
  RSS[i]=sum((yi-b0[i]-(b1[i]*xii))^2)
}
length(RSS)
## [1] 2000001
which(RSS==0,arr.ind=TRUE)
## integer(0)
```

Qs 3

```
#Step 1:
rm(list=ls())
n=50
x=runif(n)
x
##  [1] 0.8474532 0.4975273 0.3879090 0.2464490 0.1110965 0.3899944 0.5719353
##  [8] 0.2168928 0.4447680 0.2179907 0.5022996 0.3539046 0.6499852 0.3747140
## [15] 0.3554454 0.5336879 0.7403344 0.2211029 0.4127461 0.2656867 0.6299731
## [22] 0.1838285 0.8636441 0.7465680 0.6682846 0.6180179 0.3722381 0.5298357
## [29] 0.8746823 0.5817501 0.8397678 0.3124482 0.7082903 0.2650178 0.5943432
## [36] 0.4812898 0.2650327 0.5645904 0.9131882 0.9018744 0.2741666 0.3214828
## [43] 0.9856409 0.6199933 0.9373141 0.4665327 0.4068326 0.6592303 0.1523466
## [50] 0.5728671
e=rnorm(n)
e
##  [1] -0.71040656  0.25688371 -0.24669188 -0.34754260 -0.95161857 -0.04502772
##  [7] -0.78490447 -1.66794194 -0.38022652  0.91899661 -0.57534696  0.60796432
## [13] -1.61788271 -0.05556197  0.51940720  0.30115336  0.10567619 -0.64070601
## [19] -0.84970435 -1.02412879  0.11764660 -0.94747461 -0.49055744 -0.25609219
```

```
## [25]  1.84386201 -0.65194990  0.23538657  0.07796085 -0.96185663 -0.07130809
## [31]  1.44455086  0.45150405  0.04123292 -0.42249683 -2.05324722  1.13133721
## [37] -1.46064007  0.73994751  1.90910357 -1.44389316  0.70178434 -0.26219749
## [43] -1.57214416 -1.51466765 -1.60153617 -0.53090652 -1.46175558  0.68791677
## [49]  2.10010894 -1.28703048
b0=2
b1=3
y=b0+(b1*x)+e
y
##  [1] 3.8319529 3.7494655 2.9170352 2.3918044 1.3816708 3.1249556 2.9309015
##  [8] 0.9827364 2.9540775 3.5729686 2.9315517 3.6696780 2.3320728 3.0685799
## [15] 3.5857433 3.9022172 4.3266793 2.0226028 2.3885340 1.7729313 4.0075658
## [22] 1.6040109 4.1003749 3.9836118 5.8487160 3.2021037 3.3521008 3.6674679
## [29] 3.6621904 3.6739422 5.9638542 3.3888485 4.1661039 2.3725566 1.7297824
## [36] 4.5752066 1.3344581 4.4337188 6.6486682 3.2617300 3.5242842 2.7022508
## [43] 3.3847785 2.3453123 3.2104061 2.8686916 1.7587422 4.6656077 4.5571488
## [50] 2.4315707
#Step 2:
R=1000
lin_reg=lm(y~x)
coef(lin_reg)
## (Intercept)           x
##    1.899349    2.701302
set.seed(123)
xr=matrix(runif(1000*n),ncol=n,nrow=1000)
er=matrix(rnorm(1000*n),ncol=n,nrow=1000)
yr=matrix(0,ncol=n,nrow=1000)
b0_est=rep(0,1000)
b1_est=rep(0,1000)
for(i in 1:1000)
{
  for(j in 1:n)
  {
    yr[i,j]=b0+b1*xr[i,j]+er[i,j]
  }
  lin_reg=lm(yr[i,]~xr[i,])
  b0_est[i]=coef(lin_reg)[1]
  b1_est[i]=coef(lin_reg)[2]
}
```

```
b0_estimated=mean(b0_est);b0_estimated
## [1] 2.013712
b1_estimated=mean(b1_est);b1_estimated
## [1] 2.978454
se_b0_estmated=sd(b0_est);se_b0_estmated
## [1] 0.2940576
se_b1_estimated=sd(b1_est);se_b1_estimated
## [1] 0.5118415
```

QS 4

```
# Load MASS library
library(MASS)

# Load Boston dataset
data(Boston)

head(Boston)
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio  black lstat
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90  4.98
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90  9.14
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83  4.03
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63  2.94
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90  5.33
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12  5.21
##   medv
## 1 24.0
## 2 21.6
## 3 34.7
## 4 33.4
## 5 36.2
## 6 28.7
```

a.

```
# Model 1: medv ~ crim
m1 = lm(medv ~ crim, data = Boston)

# Model 2: medv ~ nox
```

```
m2 = lm(medv ~ nox, data = Boston)

# Model 3: medv ~ black

m3 = lm(medv ~ black, data = Boston)

# Model 4: medv ~ lstat

m4 = lm(medv ~ lstat, data = Boston)

# Presenting Output in single table

summary_table <- data.frame(
  Model = c("medv ~ crim", "medv ~ nox", "medv ~ black", "medv ~ lstat"),
  Intercept = c(coef(m1)[1], coef(m2)[1], coef(m3)[1], coef(m4)[1]),
  Slope = c(coef(m1)[2], coef(m2)[2], coef(m3)[2], coef(m4)[2]),
  R_squared = c(summary(m1)$r.squared,
                summary(m2)$r.squared,
                summary(m3)$r.squared,
                summary(m4)$r.squared),
  P_value = c(summary(m1)$coefficients[2,4],
              summary(m2)$coefficients[2,4],
              summary(m3)$coefficients[2,4],
              summary(m4)$coefficients[2,4])
)
print(summary_table)
##                Model Intercept        Slope R_squared      P_value
## crim   medv ~ crim  24.03311  -0.41519028 0.1507805 1.173987e-19
## nox     medv ~ nox  41.34587 -33.91605501 0.1826030 7.065042e-24
## black medv ~ black  10.55103   0.03359306 0.1111961 1.318113e-14
## lstat medv ~ lstat  34.55384  -0.95004935 0.5441463 5.081103e-88
```

b. The best fit is the model with the highest R^2 (explains the most variation in medv). Here medv ~ lstat gives the best fit as it usually has the largest R^2 (around 0.54)

c.

1. Crime rate (crim) Here coefficient is negative means as crime rate increases, median house value decreases. Usefulness: Significant predictor, but explains only a moderate amount of variation

2. Nitrogen oxides(nox) Coefficient is negative meaning higher pollution implies lower housing prices. Usefulness: Also significant but still not the strongest single predictor.Environmental quality clearly matters.

3. Proportion of Blacks(black) Coefficient is usually small and positive implies slight association with higher prices. Usefulness: Statistically significant but not very strong in explaining variation alone.

4. Lower stats population (lstat) Coefficient is strongly negative. Usefulness: Most useful single predictor here — highest R^2 and strong significance.