Buan6356_Homework2_Udayakumar

Anjana

3/2/2021

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
library(tinytex)
library(ltm)
```

```
## Warning: package 'ltm' was built under R version 4.0.4
```

```
## Loading required package: MASS
```

```
## Loading required package: msm
```

```
## Warning: package 'msm' was built under R version 4.0.4
```

```
## Loading required package: polycor
```

```
## Warning: package 'polycor' was built under R version 4.0.4
```

```
library(forecast)
```

```
## Warning: package 'forecast' was built under R version 4.0.4
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.0.4
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(pivottabler)
```

```
## Warning: package 'pivottabler' was built under R version 4.0.4
```

```
library(GGally)
```

```
## Loading required package: ggplot2

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(MASS)


library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
library(ggplot2)
input <- if(file.exists("Airfares.csv")){"Airfares.csv"}
airfares <- fread(input)
airfares.dt <- airfares[,5:18]
```

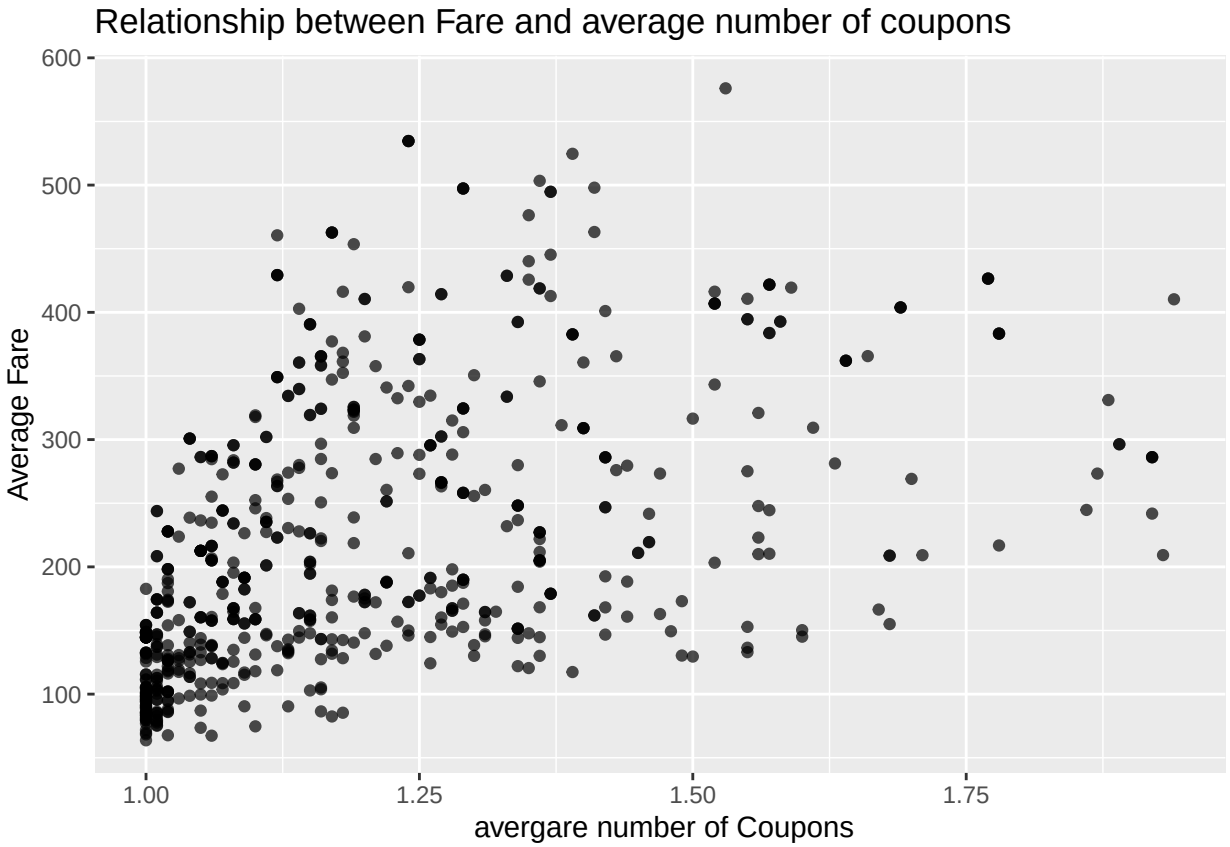
```
airfares_corr <- sapply(airfares.dt,as.numeric)
```

```
## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion

## Warning in lapply(X = X, FUN = FUN, ...): NAs introduced by coercion
```

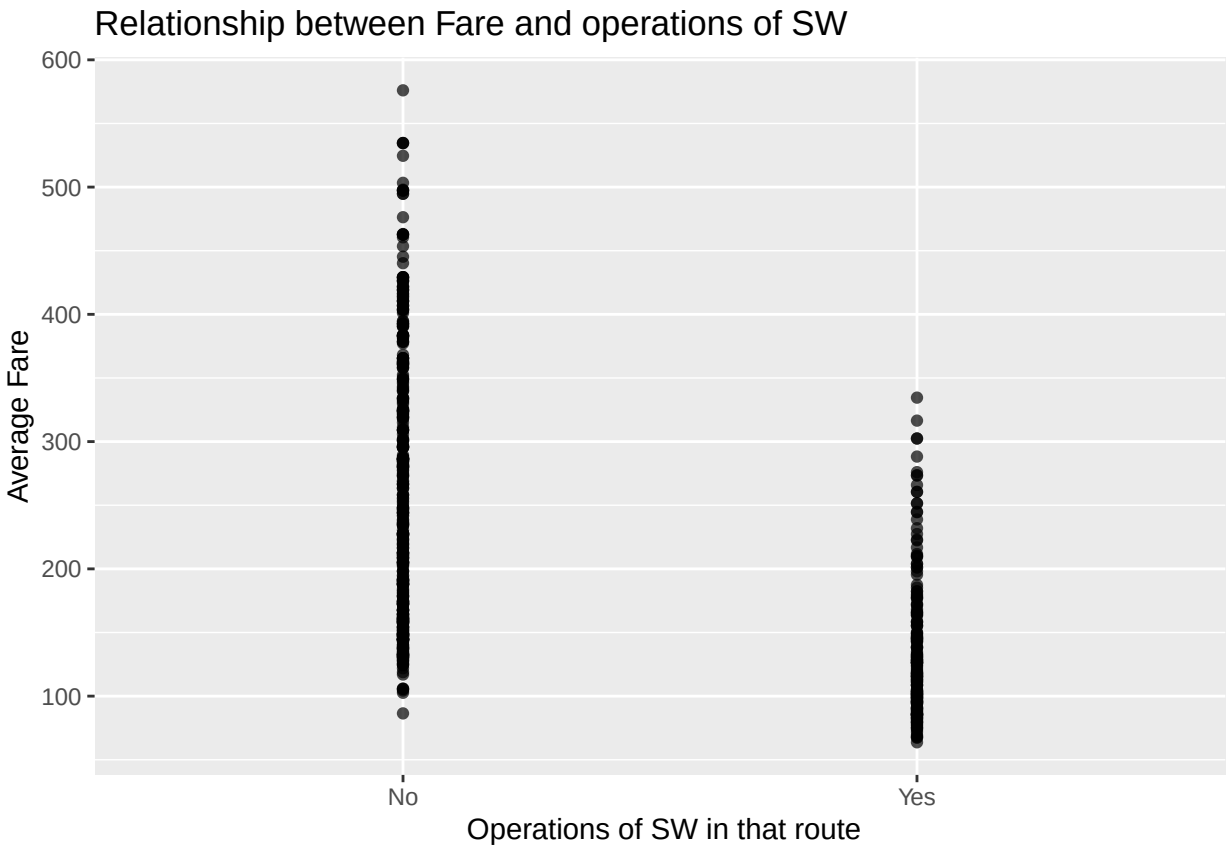
```r
correlation <- cor(airfares_corr)
options(scipen = 999)

correlation
```

```
##               COUPON         NEW VACATION SW          HI    S_INCOME   E_INCOME
## COUPON    1.00000000  0.02022307       NA NA -0.34725207 -0.08840265  0.0468892
## NEW       0.02022307  1.00000000       NA NA  0.05414685  0.02659673  0.1133766
## VACATION          NA          NA        1 NA          NA          NA         NA
## SW                NA          NA       NA  1          NA          NA         NA
## HI       -0.34725207  0.05414685       NA NA  1.00000000 -0.02738221  0.0823926
## S_INCOME -0.08840265  0.02659673       NA NA -0.02738221  1.00000000 -0.1388642
## E_INCOME  0.04688920  0.11337664       NA NA  0.08239260 -0.13886420  1.0000000
## S_POP    -0.10776336 -0.01667212       NA NA -0.17249541  0.51718718 -0.1440586
## E_POP     0.09496994  0.05856818       NA NA -0.06245600 -0.27228027  0.4584181
## SLOT              NA          NA       NA NA          NA          NA         NA
## GATE              NA          NA       NA NA          NA          NA         NA
## DISTANCE  0.74680521  0.08096520       NA NA -0.31237457  0.02815334  0.1765307
## PAX      -0.33697358  0.01049527       NA NA -0.16896078  0.13819710  0.2599611
## FARE      0.48555486  0.08709985       NA NA  0.04079123  0.20152956  0.3230925
##                S_POP       E_POP SLOT GATE    DISTANCE         PAX        FARE
## COUPON   -0.10776336  0.09496994   NA   NA  0.74680521 -0.33697358  0.48555486
## NEW      -0.01667212  0.05856818   NA   NA  0.08096520  0.01049527  0.08709985
## VACATION          NA          NA   NA   NA          NA          NA          NA
## SW                NA          NA   NA   NA          NA          NA          NA
## HI       -0.17249541 -0.06245600   NA   NA -0.31237457 -0.16896078  0.04079123
## S_INCOME  0.51718718 -0.27228027   NA   NA  0.02815334  0.13819710  0.20152956
## E_INCOME -0.14405857  0.45841806   NA   NA  0.17653074  0.25996105  0.32309251
## S_POP     1.00000000 -0.28014283   NA   NA  0.01843667  0.28461056  0.14073636
## E_POP    -0.28014283  1.00000000   NA   NA  0.11563970  0.31469750  0.27413414
## SLOT              NA          NA    1   NA          NA          NA          NA
## GATE              NA          NA   NA    1          NA          NA          NA
## DISTANCE  0.01843667  0.11563970   NA   NA  1.00000000 -0.10248160  0.66597169
## PAX       0.28461056  0.31469750   NA   NA -0.10248160  1.00000000 -0.09351465
## FARE      0.14073636  0.27413414   NA   NA  0.66597169 -0.09351465  1.00000000
```

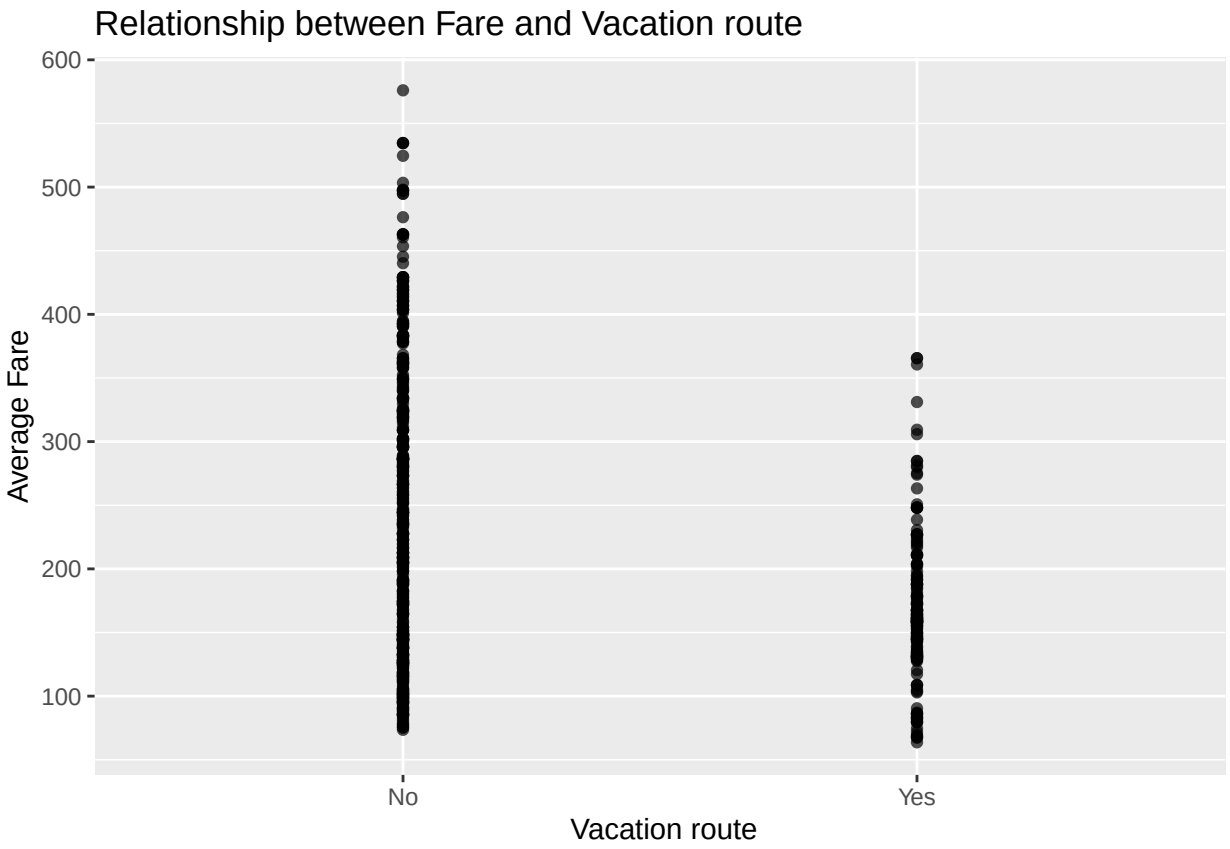
```r
ggplot(airfares.dt)+geom_point(aes(x = COUPON,y=FARE),alpha = 0.7) +
xlab(" avergare number of Coupons")+ylab("Average Fare")+
ggtitle("Relationship between Fare and average number of coupons")
```



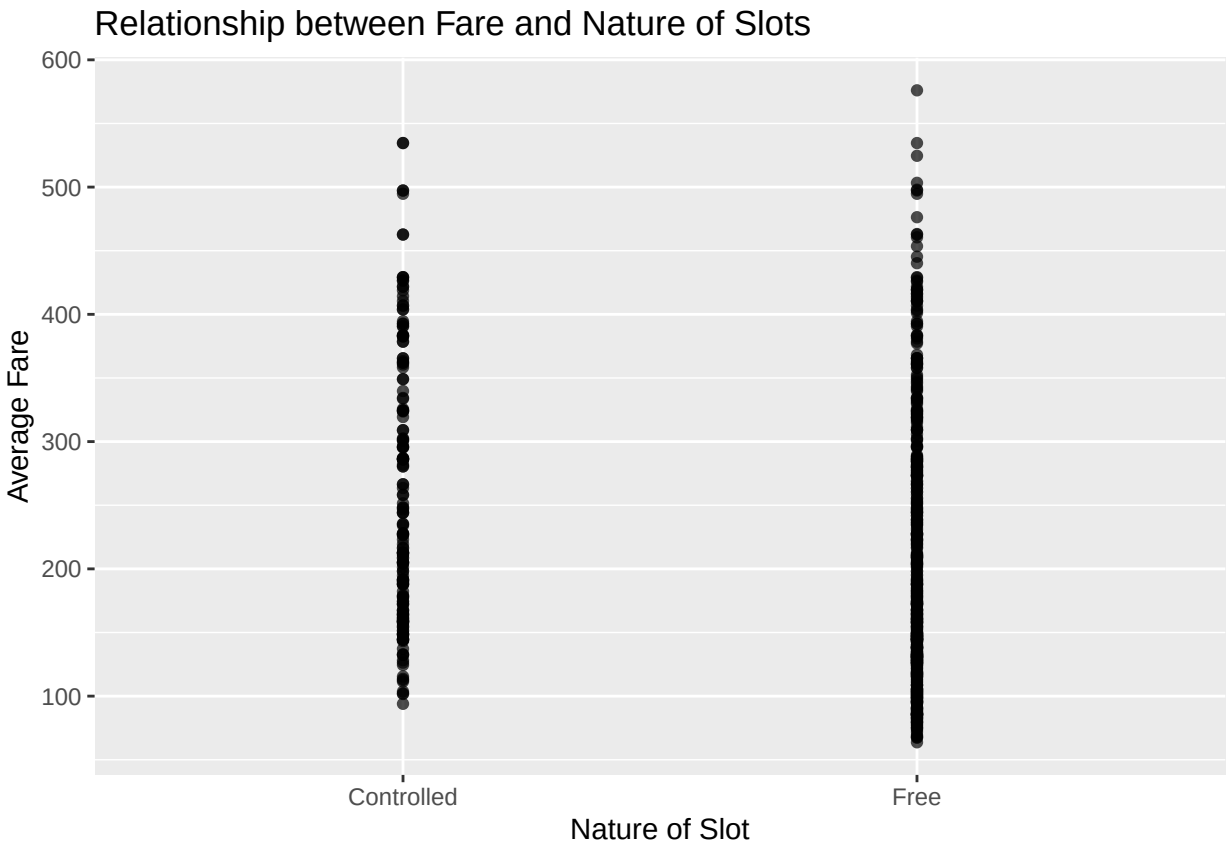

```
ggplot(airfares.dt)+geom_point(aes(x = SW,y=FARE),alpha = 0.7) +
xlab(" Operations of SW in that route")+ylab("Average Fare")+
ggtitle("Relationship between Fare and operations of SW")
```



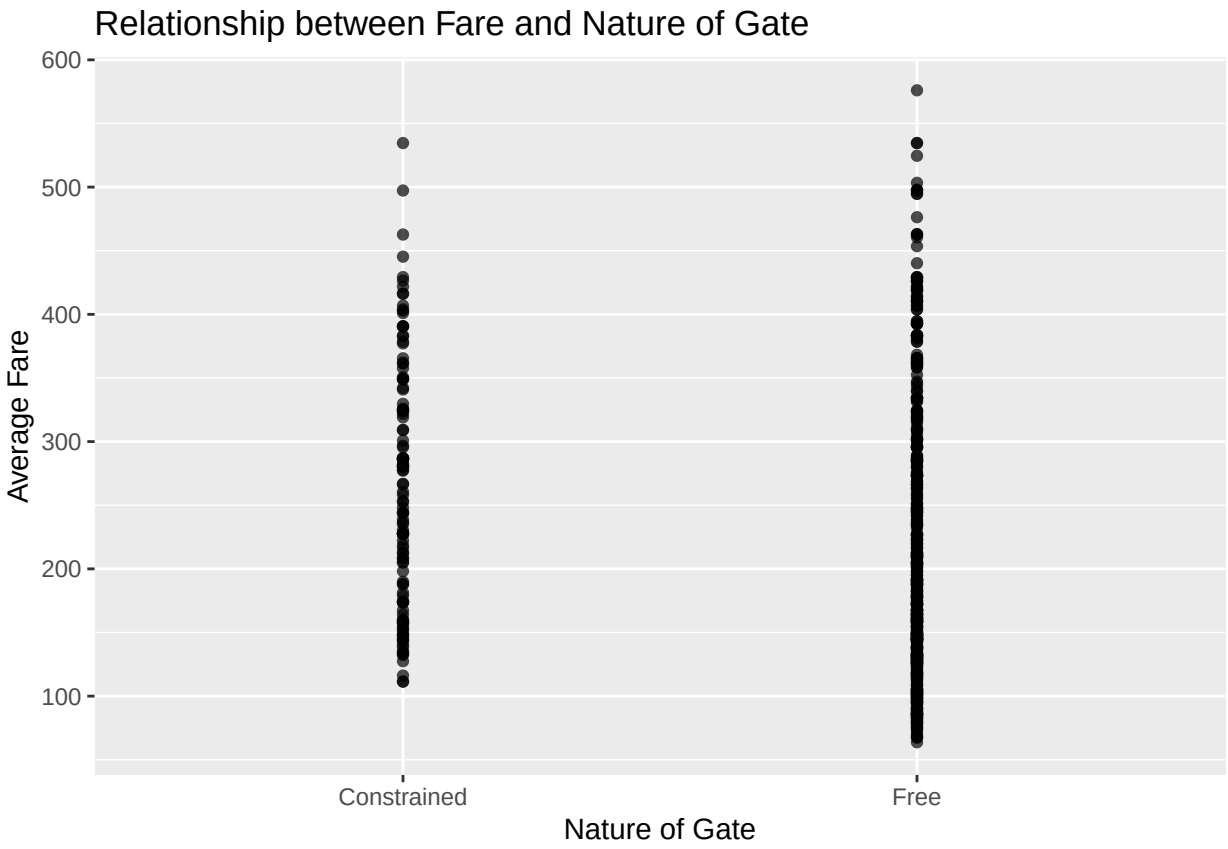

```
ggplot(airfares.dt)+geom_point(aes(x = VACATION,y=FARE),alpha = 0.7) +
xlab(" Vacation route")+ylab("Average Fare")+
ggtitle("Relationship between Fare and Vacation route")
```



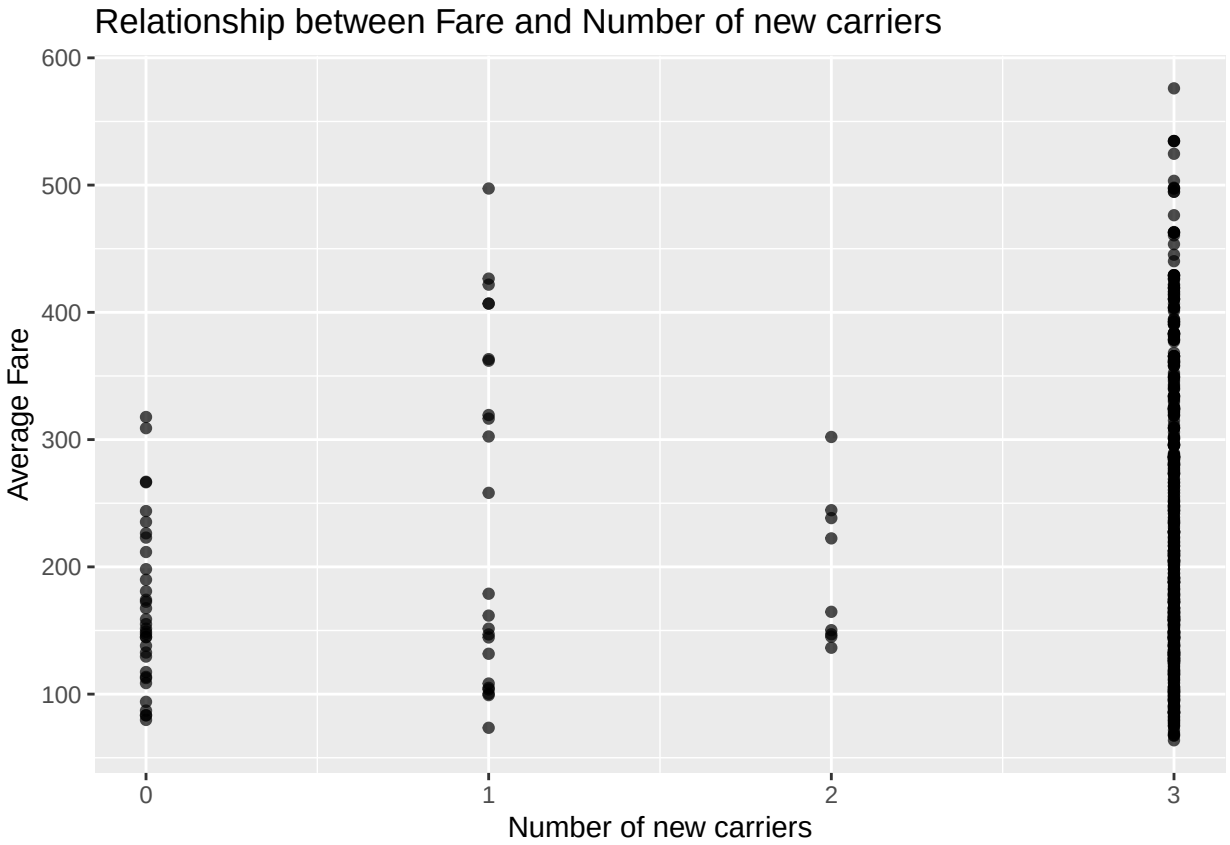

```
ggplot(airfares.dt)+geom_point(aes(x = SLOT,y=FARE),alpha = 0.7) +
xlab("Nature of Slot")+ylab("Average Fare")+
ggtitle("Relationship between Fare and Nature of Slots")
```



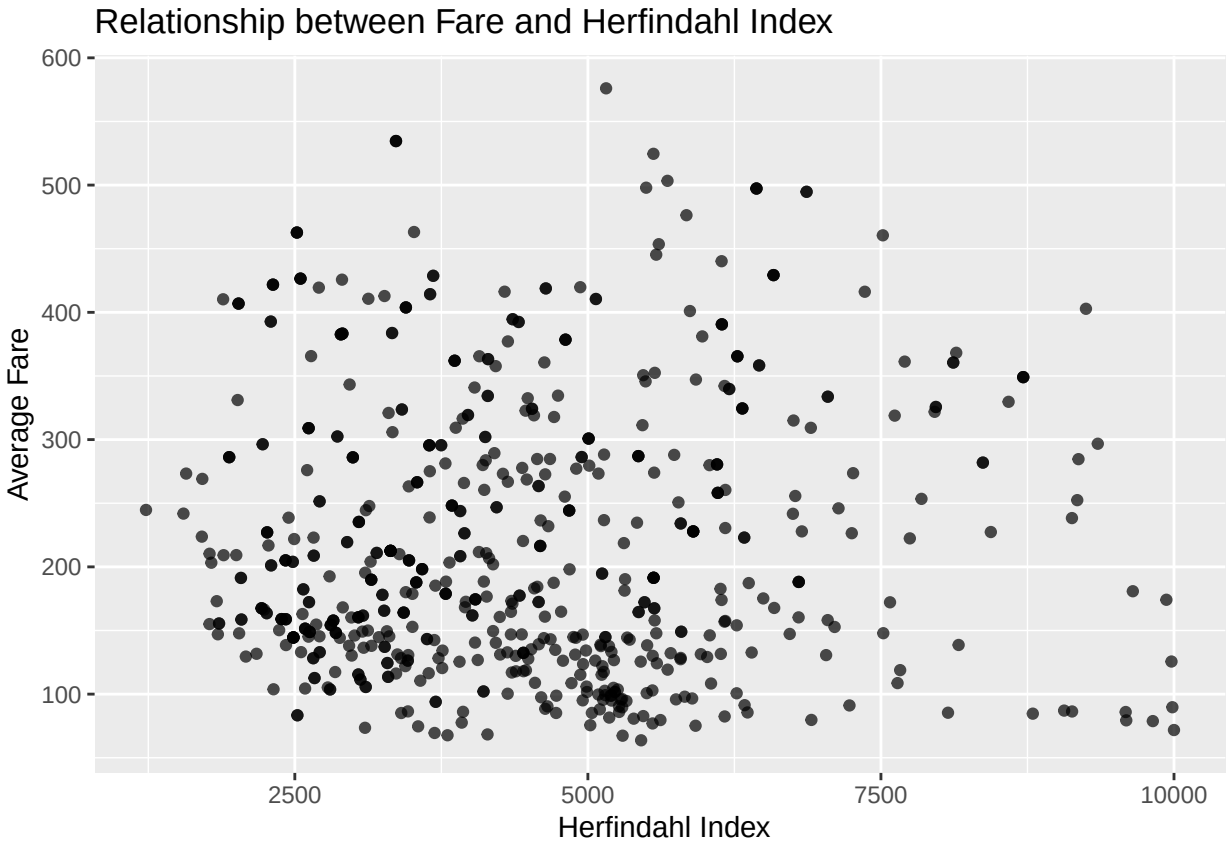

```
ggplot(airfares.dt)+geom_point(aes(x = GATE,y=FARE),alpha = 0.7) +
xlab("Nature of Gate")+ylab("Average Fare")+
ggtitle("Relationship between Fare and Nature of Gate")
```



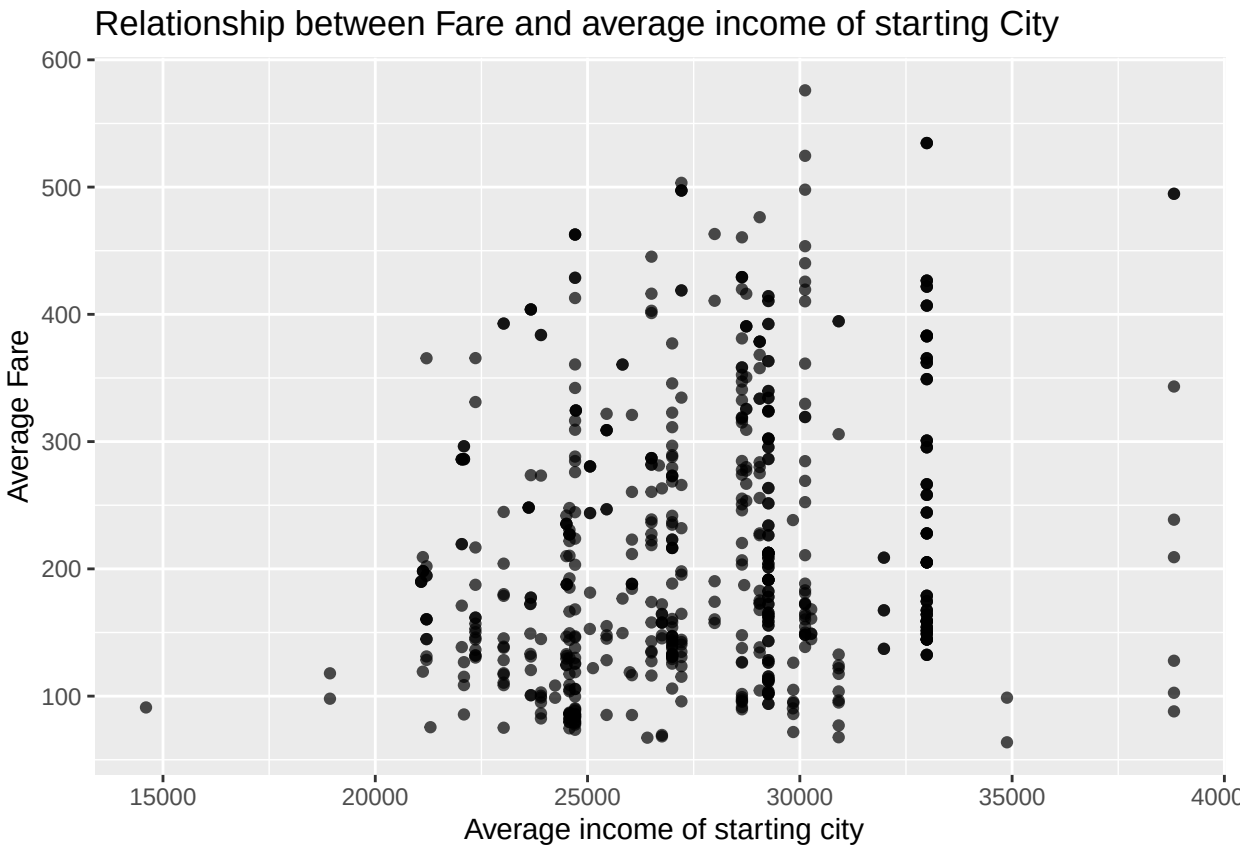

```
ggplot(airfares.dt)+geom_point(aes(x = NEW,y=FARE),alpha = 0.7) +
xlab("Number of new carriers")+ylab("Average Fare")+
ggtitle("Relationship between Fare and Number of new carriers")
```



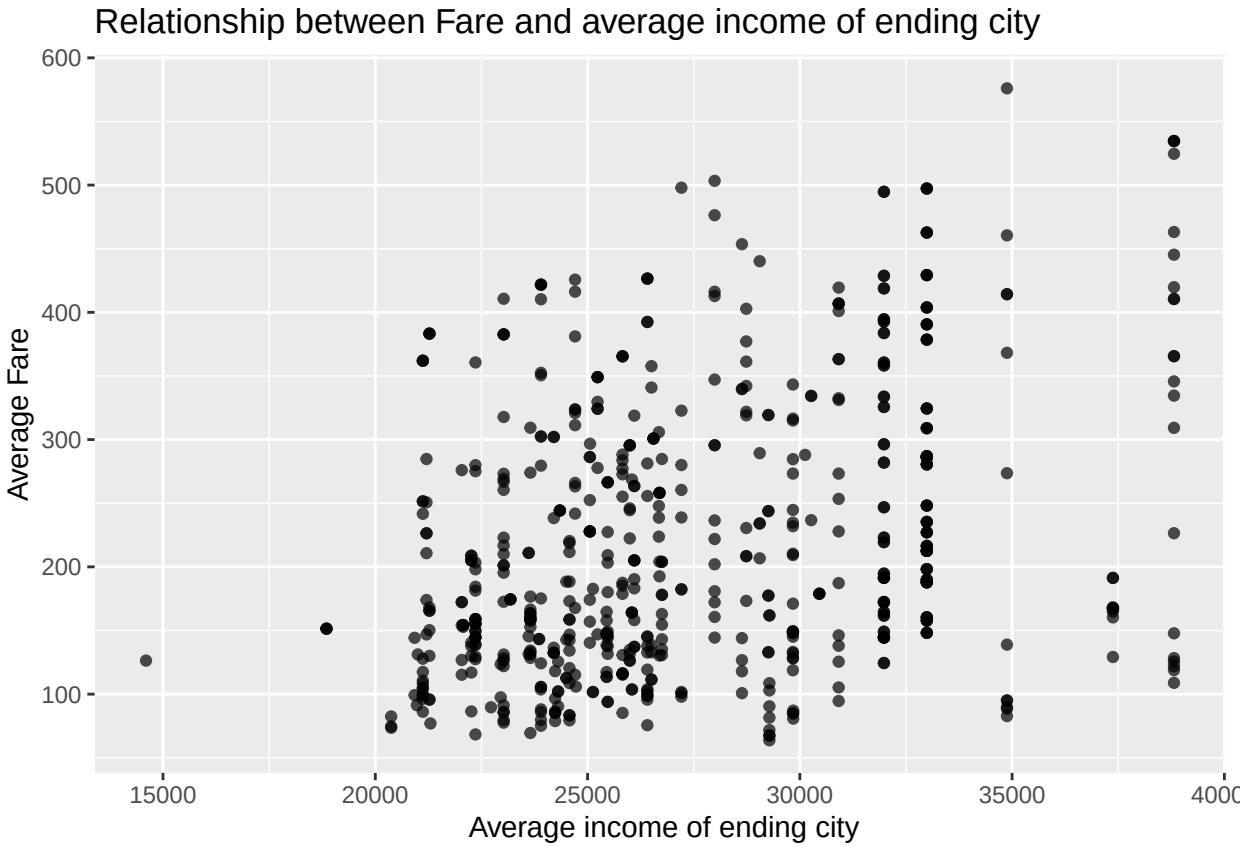

```
ggplot(airfares.dt)+geom_point(aes(x = HI,y=FARE),alpha = 0.7) +
xlab(" Herfindahl Index")+ylab("Average Fare")+
ggtitle("Relationship between Fare and Herfindahl Index")
```



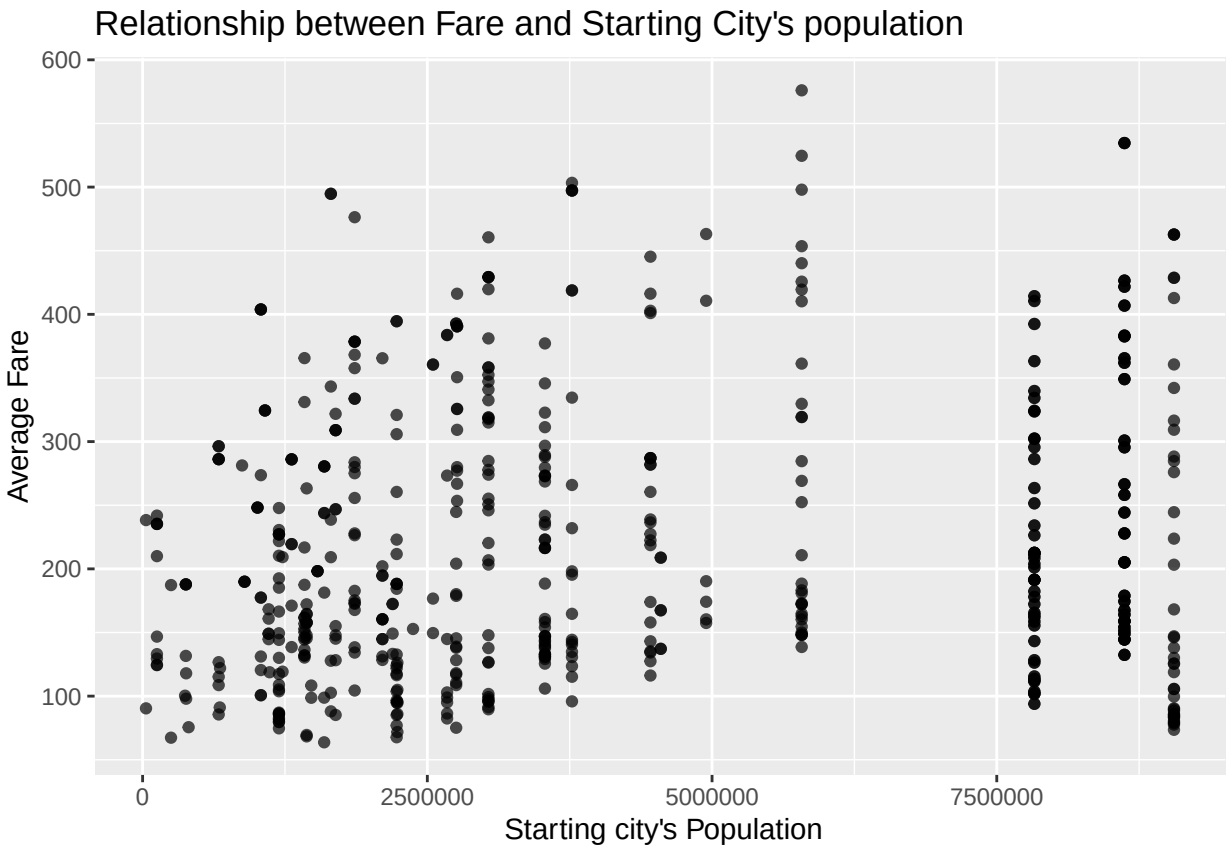

```
ggplot(airfares.dt)+geom_point(aes(x = S_INCOME,y=FARE),alpha = 0.7) +
xlab(" Average income of starting city")+ylab("Average Fare")+
ggtitle("Relationship between Fare and average income of starting City")
```



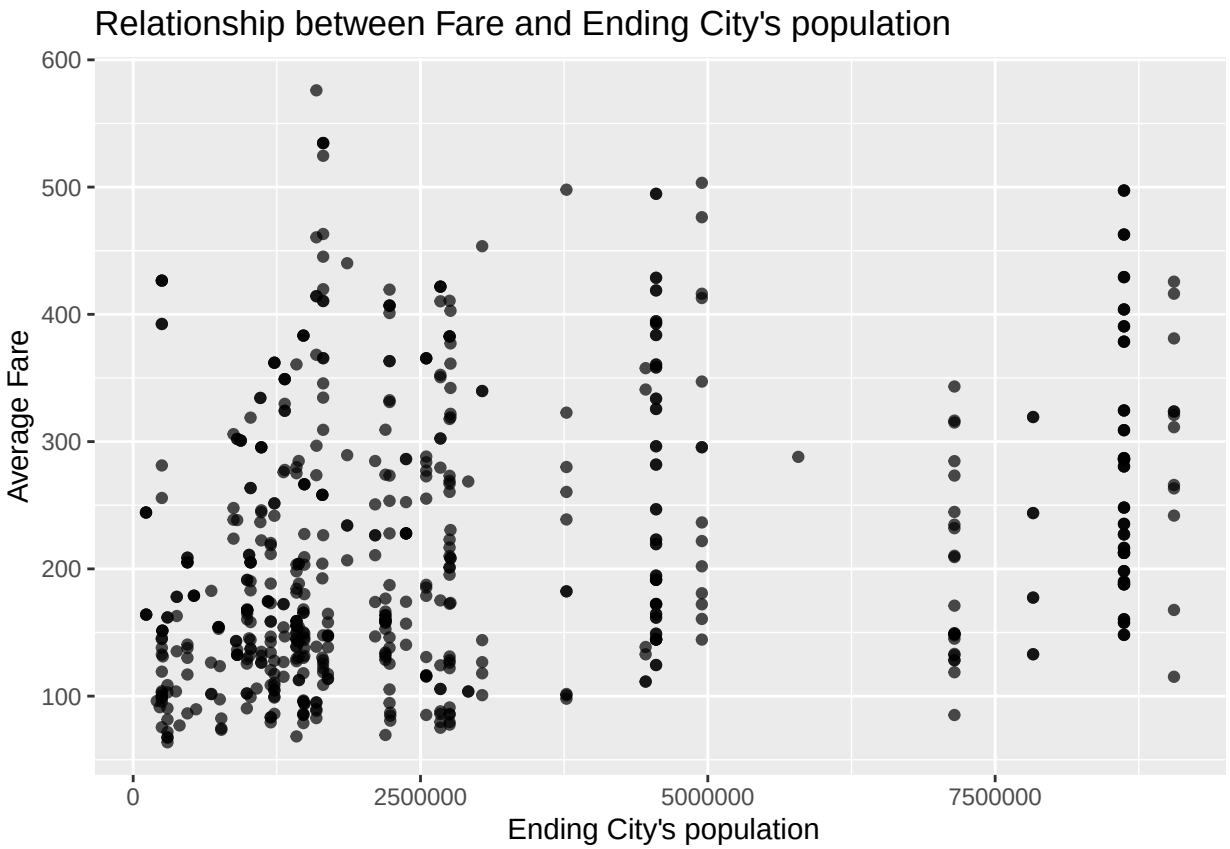

```
ggplot(airfares.dt)+geom_point(aes(x = E_INCOME,y=FARE),alpha = 0.7) +
xlab(" Average income of ending city")+ylab("Average Fare")+
ggtitle("Relationship between Fare and average income of ending city")
```



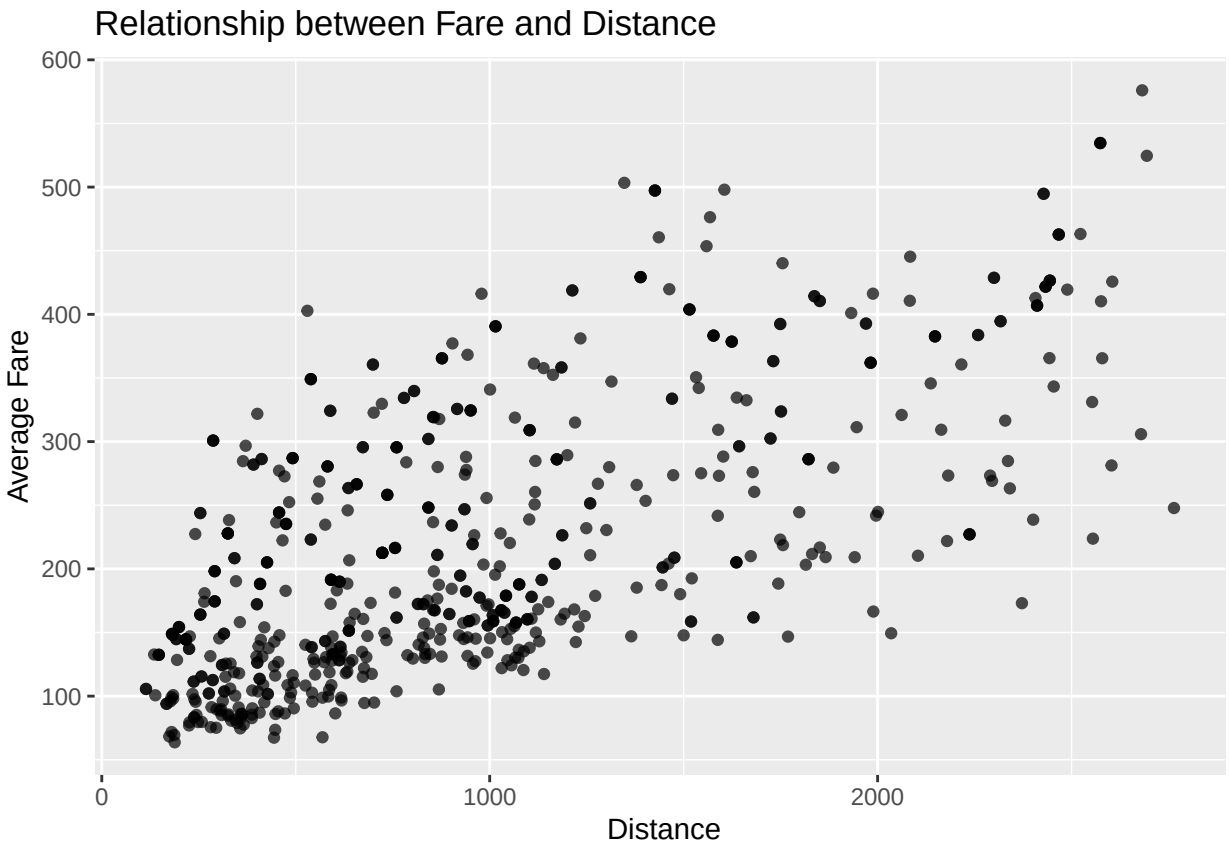

```
ggplot(airfares.dt)+geom_point(aes(x = S_POP,y=FARE),alpha = 0.7) +
xlab(" Starting city's Population")+ylab("Average Fare")+
ggtitle("Relationship between Fare and Starting City's population")
```



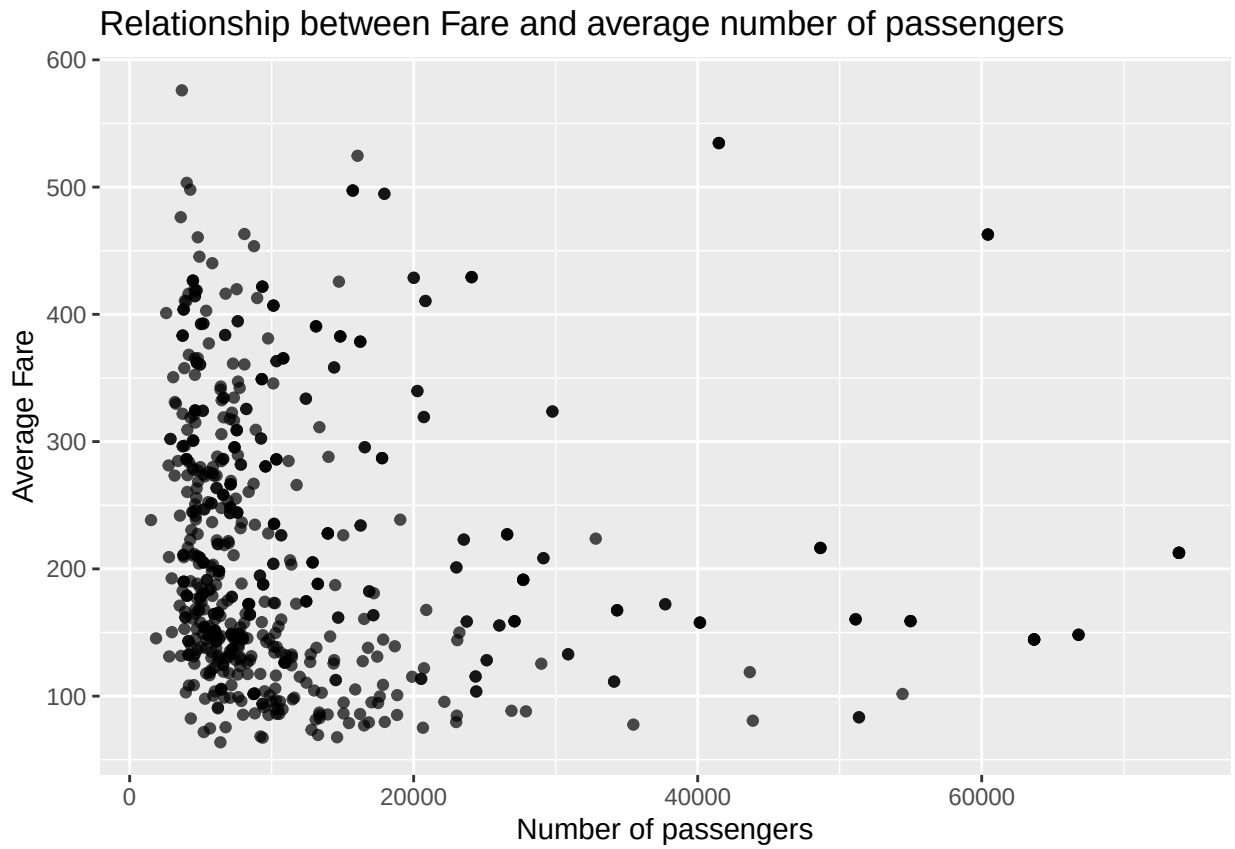

```
ggplot(airfares.dt)+geom_point(aes(x = E_POP,y=FARE),alpha = 0.7) +
xlab(" Ending City's population")+ylab("Average Fare")+
ggtitle("Relationship between Fare and Ending City's population")
```




```
ggplot(airfares.dt)+geom_point(aes(x = DISTANCE,y=FARE),alpha = 0.7) +
xlab(" Distance")+ylab("Average Fare")+
ggtitle("Relationship between Fare and Distance")
```




```
ggplot(airfares.dt)+geom_point(aes(x = PAX,y=FARE),alpha = 0.7) +
xlab("Number of passengers")+ylab("Average Fare")+
ggtitle("Relationship between Fare and average number of passengers")
```




```
airfares_categorical <- airfares.dt[,c(3,4,10,11)]
Vacation_count <- table(airfares_categorical$VACATION)
Vacation_count
```

```
##
##  No Yes
## 468 170
```

```
sum(Vacation_count)
```

```
## [1] 638
```

```
(Vacation_count/sum(Vacation_count))*100
```

```
##
##       No      Yes
## 73.35423 26.64577
```

```
sw_count <- table(airfares_categorical$SW)
sw_count
```

```
##
##  No Yes
## 444 194
```

```r
sum(sw_count)
```

```
## [1] 638
```

```r
(sw_count/sum(sw_count))*100
```

```
##
##       No      Yes
## 69.59248 30.40752
```

```r
slot_count <- table(airfares_categorical$SLOT)
slot_count
```

```
##
## Controlled       Free
##        182        456
```

```r
sum(slot_count)
```

```
## [1] 638
```

```r
slot_count/sum(slot_count)*100
```

```
##
## Controlled       Free
##   28.52665   71.47335
```

```r
Gate_count <- table(airfares_categorical$GATE)
Gate_count
```

```
##
## Constrained        Free
##         124         514
```

```r
sum(Gate_count)
```

```
## [1] 638
```

```r
Gate_count/sum(Gate_count)*100
```

```
##
## Constrained        Free
##    19.43574    80.56426
```

```r
pivot_table <-
  data.table::cube(airfares.dt,.(Average_Fare=mean(FARE)),
                   by =c("VACATION","SW","SLOT","GATE"))
pivot_table
```

```
##      VACATION   SW       SLOT        GATE Average_Fare
##  1:       No  Yes       Free        Free     136.4294
##  2:       No   No       Free        Free     270.0005
##  3:       No  Yes Controlled        Free     152.5529
##  4:      Yes  Yes       Free        Free     124.1314
##  5:       No   No Controlled        Free     285.4975
##  6:       No   No       Free Constrained     289.5674
##  7:       No   No Controlled Constrained     283.1994
##  8:      Yes   No       Free        Free     192.5948
##  9:       No  Yes Controlled Constrained     111.4200
## 10:       No  Yes       Free Constrained     190.6557
## 11:      Yes  Yes Controlled        Free     170.4100
## 12:      Yes   No       Free Constrained     178.9730
## 13:      Yes   No Controlled        Free     182.8554
## 14:       No  Yes       Free        <NA>     139.5664
## 15:       No   No       Free        <NA>     277.7478
## 16:       No  Yes Controlled        <NA>     150.2678
## 17:      Yes  Yes       Free        <NA>     124.1314
## 18:       No   No Controlled        <NA>     285.1842
## 19:      Yes   No       Free        <NA>     189.4633
## 20:      Yes  Yes Controlled        <NA>     170.4100
## 21:      Yes   No Controlled        <NA>     182.8554
## 22:       No  Yes       <NA>        Free     138.5218
## 23:       No   No       <NA>        Free     277.5827
## 24:      Yes  Yes       <NA>        Free     127.4971
## 25:       No   No       <NA> Constrained     288.3734
## 26:      Yes   No       <NA>        Free     189.7242
## 27:       No  Yes       <NA> Constrained     180.7512
## 28:      Yes   No       <NA> Constrained     178.9730
## 29:       No  Yes       <NA>        <NA>     140.9522
## 30:       No   No       <NA>        <NA>     280.7314
## 31:      Yes  Yes       <NA>        <NA>     127.4971
## 32:      Yes   No       <NA>        <NA>     187.8544
## 33:       No <NA>       Free        Free     204.6481
## 34:       No <NA> Controlled        Free     268.2452
## 35:      Yes <NA>       Free        Free     163.0047
## 36:       No <NA>       Free Constrained     281.4218
## 37:       No <NA> Controlled Constrained     274.1584
## 38:      Yes <NA> Controlled        Free     181.2997
## 39:      Yes <NA>       Free Constrained     178.9730
## 40:       No <NA>       Free        <NA>     225.1694
## 41:       No <NA> Controlled        <NA>     268.9942
## 42:      Yes <NA>       Free        <NA>     165.3189
## 43:      Yes <NA> Controlled        <NA>     181.2997
## 44:       No <NA>       <NA>        Free     227.5361
## 45:      Yes <NA>       <NA>        Free     166.9076
## 46:       No <NA>       <NA> Constrained     280.0948
## 47:      Yes <NA>       <NA> Constrained     178.9730
## 48:       No <NA>       <NA>        <NA>     239.2158
## 49:      Yes <NA>       <NA>        <NA>     168.3271
## 50:     <NA>  Yes       Free        Free     132.6282
## 51:     <NA>   No       Free        Free     242.1178
## 52:     <NA>  Yes Controlled        Free     155.9543
## 53:     <NA>   No Controlled        Free     265.2582
```

```
## 54:     <NA>   No       Free Constrained     266.9971
## 55:     <NA>   No Controlled Constrained     283.1994
## 56:     <NA>  Yes Controlled Constrained     111.4200
## 57:     <NA>  Yes       Free Constrained     190.6557
## 58:     <NA>  Yes       Free        <NA>     134.9898
## 59:     <NA>   No       Free        <NA>     250.7029
## 60:     <NA>  Yes Controlled        <NA>     153.9300
## 61:     <NA>   No Controlled        <NA>     267.2766
## 62:     <NA>  Yes       <NA>        Free     135.2618
## 63:     <NA>   No       <NA>        Free     252.1359
## 64:     <NA>   No       <NA> Constrained     269.5113
## 65:     <NA>  Yes       <NA> Constrained     180.7512
## 66:     <NA>  Yes       <NA>        <NA>     137.1376
## 67:     <NA>   No       <NA>        <NA>     256.6754
## 68:     <NA> <NA>       Free        Free     190.6483
## 69:     <NA> <NA> Controlled        Free     251.1761
## 70:     <NA> <NA>       Free Constrained     261.9077
## 71:     <NA> <NA> Controlled Constrained     274.1584
## 72:     <NA> <NA>       Free        <NA>     207.0567
## 73:     <NA> <NA> Controlled        <NA>     253.5754
## 74:     <NA> <NA>       <NA>        Free     209.8429
## 75:     <NA> <NA>       <NA> Constrained     263.7848
## 76:     <NA> <NA>       <NA>        <NA>     220.3269
##     VACATION   SW       SLOT        GATE Average_Fare
```

```
lm_cat <- lm(FARE~VACATION+SW+GATE+SLOT,data = airfares.dt)
summary(lm_cat)
```

```
##
## Call:
## lm(formula = FARE ~ VACATION + SW + GATE + SLOT, data = airfares.dt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -163.15  -58.97  -16.10   57.17  310.26
##
## Coefficients:
##             Estimate Std. Error t value             Pr(>|t|)
## (Intercept)  291.075     10.309  28.234 < 0.0000000000000002 ***
## VACATIONYes  -64.956      7.952  -8.168   0.0000000000000017 ***
## SWYes       -112.073      8.143 -13.764 < 0.0000000000000002 ***
## GATEFree     -13.992      9.347  -1.497                0.135
## SLOTFree     -11.317      8.195  -1.381                0.168
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 87.09 on 633 degrees of freedom
## Multiple R-squared:  0.3448, Adjusted R-squared:  0.3407
## F-statistic: 83.28 on 4 and 633 DF,  p-value: < 0.00000000000000022
```

```
set.seed(42)
train.index <- sample(c(1:638),510)
train.df <- airfares.dt[train.index, ]
valid.df <- airfares.dt[-train.index,]
```

```r
airfares.lm <- lm(FARE~.,data = train.df)
airfares.lm.stepwise <- step(airfares.lm,direction = "both")
```

```
## Start:  AIC=4043.05
## FARE ~ COUPON + NEW + VACATION + SW + HI + S_INCOME + E_INCOME +
##     S_POP + E_POP + SLOT + GATE + DISTANCE + PAX
##
##            Df Sum of Sq     RSS    AIC
## - COUPON    1       523 1339010 4041.2
## - S_INCOME  1      2288 1340775 4041.9
## - NEW       1      4184 1342671 4042.6
## <none>                  1338487 4043.0
## - SLOT      1     34783 1373270 4054.1
## - E_INCOME  1     36287 1374774 4054.7
## - PAX       1     44129 1382616 4057.6
## - E_POP     1     46174 1384661 4058.3
## - GATE      1     54087 1392574 4061.2
## - S_POP     1     62925 1401412 4064.5
## - SW        1    164672 1503159 4100.2
## - HI        1    173271 1511758 4103.1
## - VACATION  1    275438 1613925 4136.5
## - DISTANCE  1    891835 2230322 4301.5
##
## Step:  AIC=4041.25
## FARE ~ NEW + VACATION + SW + HI + S_INCOME + E_INCOME + S_POP +
##     E_POP + SLOT + GATE + DISTANCE + PAX
##
##            Df Sum of Sq     RSS    AIC
## - S_INCOME  1      2105 1341115 4040.0
## - NEW       1      4477 1343487 4040.9
## <none>                  1339010 4041.2
## + COUPON    1       523 1338487 4043.0
## - SLOT      1     35531 1374541 4052.6
## - E_INCOME  1     35928 1374938 4052.7
## - E_POP     1     47481 1386491 4057.0
## - GATE      1     54537 1393547 4059.6
## - PAX       1     57754 1396764 4060.8
## - S_POP     1     62447 1401457 4062.5
## - SW        1    167564 1506574 4099.4
## - HI        1    184110 1523120 4104.9
## - VACATION  1    276682 1615692 4135.0
## - DISTANCE  1   1709875 3048885 4458.9
##
## Step:  AIC=4040.05
## FARE ~ NEW + VACATION + SW + HI + E_INCOME + S_POP + E_POP +
##     SLOT + GATE + DISTANCE + PAX
##
##            Df Sum of Sq     RSS    AIC
## - NEW       1      4517 1345632 4039.8
## <none>                  1341115 4040.0
## + S_INCOME  1      2105 1339010 4041.2
## + COUPON    1       340 1340775 4041.9
```

```
## - E_INCOME  1     34205 1375320 4050.9
## - SLOT      1     38454 1379568 4052.5
## - E_POP     1     45877 1386992 4055.2
## - GATE      1     55388 1396502 4058.7
## - PAX       1     55652 1396767 4058.8
## - S_POP     1     72084 1413199 4064.7
## - HI        1    185949 1527063 4104.3
## - SW        1    197728 1538842 4108.2
## - VACATION  1    307447 1648561 4143.3
## - DISTANCE  1   1714666 3055781 4458.0
##
## Step:  AIC=4039.76
## FARE ~ VACATION + SW + HI + E_INCOME + S_POP + E_POP + SLOT +
##     GATE + DISTANCE + PAX
##
##            Df Sum of Sq     RSS    AIC
## <none>                  1345632 4039.8
## + NEW       1      4517 1341115 4040.0
## + S_INCOME  1      2145 1343487 4040.9
## + COUPON    1       581 1345050 4041.5
## - E_INCOME  1     32964 1378596 4050.1
## - SLOT      1     36615 1382246 4051.5
## - E_POP     1     46303 1391935 4055.0
## - GATE      1     54029 1399660 4057.8
## - PAX       1     55304 1400936 4058.3
## - S_POP     1     74031 1419662 4065.1
## - HI        1    184415 1530047 4103.3
## - SW        1    197192 1542824 4107.5
## - VACATION  1    304855 1650486 4141.9
## - DISTANCE  1   1711053 3056684 4456.2
```

```
options(scipen = 999)
summary(airfares.lm.stepwise)
```

```
##
## Call:
## lm(formula = FARE ~ VACATION + SW + HI + E_INCOME + S_POP + E_POP +
##     SLOT + GATE + DISTANCE + PAX, data = train.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -150.238  -33.021   -2.639   30.484  162.591
##
## Coefficients:
##                  Estimate    Std. Error t value             Pr(>|t|)
## (Intercept)  45.020461949  21.640727151   2.080             0.038002 *
## VACATIONYes -60.032628715   5.646165922 -10.632 < 0.0000000000000002 ***
## SWYes       -50.585400506   5.915521572  -8.551 < 0.0000000000000002 ***
## HI            0.012641329   0.001528646   8.270  0.00000000000000123 ***
## E_INCOME      0.002096705   0.000599695   3.496             0.000514 ***
## S_POP         0.000005514   0.000001052   5.240  0.00000023792617790 ***
## E_POP         0.000004795   0.000001157   4.144  0.00004013581083834 ***
## SLOTFree    -23.261008853   6.312675560  -3.685             0.000254 ***
## GATEFree    -28.876998830   6.451389759  -4.476  0.00000942774209450 ***
```

```
## DISTANCE      0.106035178   0.004209506  25.189 < 0.0000000000000002 ***
## PAX          -0.000990785   0.000218783  -4.529  0.00000743454723143 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 51.93 on 499 degrees of freedom
## Multiple R-squared:  0.7636, Adjusted R-squared:  0.7588
## F-statistic: 161.2 on 10 and 499 DF,  p-value: < 0.00000000000000022
```

```
search <- regsubsets(FARE~.,data = train.df,nbest =1,nvmax = dim(train.df)[2],
                     method = "exhaustive")
sum <- summary(search)
sum$which
```

```
##    (Intercept) COUPON   NEW VACATIONYes SWYes    HI S_INCOME E_INCOME S_POP
## 1         TRUE  FALSE FALSE       FALSE FALSE FALSE    FALSE    FALSE FALSE
## 2         TRUE  FALSE FALSE        TRUE FALSE FALSE    FALSE    FALSE FALSE
## 3         TRUE  FALSE FALSE        TRUE  TRUE FALSE    FALSE    FALSE FALSE
## 4         TRUE  FALSE FALSE        TRUE  TRUE  TRUE    FALSE    FALSE FALSE
## 5         TRUE  FALSE FALSE        TRUE  TRUE  TRUE    FALSE    FALSE FALSE
## 6         TRUE  FALSE FALSE        TRUE  TRUE  TRUE    FALSE    FALSE FALSE
## 7         TRUE  FALSE FALSE        TRUE  TRUE  TRUE    FALSE     TRUE FALSE
## 8         TRUE  FALSE FALSE        TRUE  TRUE  TRUE    FALSE     TRUE  TRUE
## 9         TRUE  FALSE FALSE        TRUE  TRUE  TRUE    FALSE    FALSE  TRUE
## 10        TRUE  FALSE FALSE        TRUE  TRUE  TRUE    FALSE     TRUE  TRUE
## 11        TRUE  FALSE  TRUE        TRUE  TRUE  TRUE    FALSE     TRUE  TRUE
## 12        TRUE  FALSE  TRUE        TRUE  TRUE  TRUE     TRUE     TRUE  TRUE
## 13        TRUE   TRUE  TRUE        TRUE  TRUE  TRUE     TRUE     TRUE  TRUE
##    E_POP SLOTFree GATEFree DISTANCE   PAX
## 1  FALSE    FALSE    FALSE     TRUE FALSE
## 2  FALSE    FALSE    FALSE     TRUE FALSE
## 3  FALSE    FALSE    FALSE     TRUE FALSE
## 4  FALSE    FALSE    FALSE     TRUE FALSE
## 5  FALSE     TRUE    FALSE     TRUE FALSE
## 6  FALSE     TRUE     TRUE     TRUE FALSE
## 7  FALSE     TRUE     TRUE     TRUE FALSE
## 8   TRUE    FALSE    FALSE     TRUE  TRUE
## 9   TRUE     TRUE     TRUE     TRUE  TRUE
## 10  TRUE     TRUE     TRUE     TRUE  TRUE
## 11  TRUE     TRUE     TRUE     TRUE  TRUE
## 12  TRUE     TRUE     TRUE     TRUE  TRUE
## 13  TRUE     TRUE     TRUE     TRUE  TRUE
```

```
sum$rsq
```

```
##  [1] 0.4101693 0.5583445 0.6798082 0.7119158 0.7244722 0.7425626 0.7466914
##  [8] 0.7515825 0.7577896 0.7635811 0.7643747 0.7647445 0.7648364
```

```
sum$adjr2
```

```
##  [1] 0.4090082 0.5566023 0.6779099 0.7096339 0.7217388 0.7394918 0.7431592
##  [8] 0.7476157 0.7534298 0.7588433 0.7591702 0.7590643 0.7586728
```

```
sum$cp
```

```
##  [1] 738.05319 427.52647 173.33886 107.61857  83.13499  46.97923  40.27097
##  [8]  31.95481  20.86294  10.64755  10.97371  12.19375  14.00000
```

```
airfares.lm.stepwise.best <-
  lm(FARE~VACATION+SW+HI+E_INCOME+S_POP+E_POP+SLOT+GATE+DISTANCE+PAX,
                                data = train.df)
airfares.lm.stepwise.pred <- predict(airfares.lm.stepwise.best,valid.df)
accuracy(airfares.lm.stepwise.pred,valid.df$FARE)
```

```
##                 ME     RMSE      MAE       MPE     MAPE
## Test set 5.905177 53.20389 40.55403 -4.108867 21.81686
```

```
airfares.lm.exhaustive.best <- lm(FARE~NEW+VACATION+SW+HI+E_INCOME+S_POP+E_POP+
                                    SLOT+GATE+DISTANCE+PAX,data = train.df)
airafares.lm.exhaustive.pred <- predict(airfares.lm.exhaustive.best,valid.df)
accuracy(airafares.lm.exhaustive.pred,valid.df$FARE)
```

```
##                 ME     RMSE      MAE       MPE    MAPE
## Test set 6.077915 53.11093 40.18106 -3.957714 21.5277
```

```
a <- data.frame(NEW = 3,VACATION = "No",SW = "No",HI = 4442.141,
                E_INCOME = 27664,S_POP=4557004,E_POP=3195503,SLOT = "Free",
                GATE ="Free",DISTANCE=1976,
                PAX=12782)
airfares.lm.exhaustive.best <-
  lm(FARE~NEW+VACATION+SW+HI+E_INCOME+
    S_POP+E_POP+SLOT+GATE+DISTANCE+PAX,data = train.df)
airfares.predict <- predict(airfares.lm.exhaustive.best,a,level = 0.95)
airfares.predict
```

```
##        1
## 343.5923
```

```
b <- data.frame(NEW = 3,VACATION = "No",SW = "Yes",HI = 4442.141,
                E_INCOME = 27664,S_POP=4557004,E_POP=3195503,SLOT = "Free",
                GATE ="Free",DISTANCE=1976,
                PAX=12782)
airfares.lm.exhaustive.best <-
  lm(FARE~NEW+VACATION+SW+HI+E_INCOME+S_POP+E_POP+SLOT+GATE+
       DISTANCE+PAX,data = train.df)
airfares.lm.exhaustive.best
```

```
##
## Call:
## lm(formula = FARE ~ NEW + VACATION + SW + HI + E_INCOME + S_POP +
##     E_POP + SLOT + GATE + DISTANCE + PAX, data = train.df)
##
## Coefficients:
```

```
##   (Intercept)            NEW    VACATIONYes          SWYes             HI
##  55.309712870   -3.938441937  -60.340928161  -50.656174535    0.012699197
##      E_INCOME          S_POP          E_POP       SLOTFree       GATEFree
##   0.002138982    0.000005447    0.000004773  -23.914001566  -29.270377507
##      DISTANCE            PAX
##   0.106440631   -0.000993962
```

```
airfares.predict <- predict(airfares.lm.exhaustive.best,b,level = 0.95)
airfares.predict
```

```
##        1
## 292.9361
```

```
airfares.lm.backward <- step(airfares.lm,direction = "backward")
```

```
## Start:  AIC=4043.05
## FARE ~ COUPON + NEW + VACATION + SW + HI + S_INCOME + E_INCOME +
##     S_POP + E_POP + SLOT + GATE + DISTANCE + PAX
##
##            Df Sum of Sq     RSS    AIC
## - COUPON    1       523 1339010 4041.2
## - S_INCOME  1      2288 1340775 4041.9
## - NEW       1      4184 1342671 4042.6
## <none>                  1338487 4043.0
## - SLOT      1     34783 1373270 4054.1
## - E_INCOME  1     36287 1374774 4054.7
## - PAX       1     44129 1382616 4057.6
## - E_POP     1     46174 1384661 4058.3
## - GATE      1     54087 1392574 4061.2
## - S_POP     1     62925 1401412 4064.5
## - SW        1    164672 1503159 4100.2
## - HI        1    173271 1511758 4103.1
## - VACATION  1    275438 1613925 4136.5
## - DISTANCE  1    891835 2230322 4301.5
##
## Step:  AIC=4041.25
## FARE ~ NEW + VACATION + SW + HI + S_INCOME + E_INCOME + S_POP +
##     E_POP + SLOT + GATE + DISTANCE + PAX
##
##            Df Sum of Sq     RSS    AIC
## - S_INCOME  1      2105 1341115 4040.0
## - NEW       1      4477 1343487 4040.9
## <none>                  1339010 4041.2
## - SLOT      1     35531 1374541 4052.6
## - E_INCOME  1     35928 1374938 4052.7
## - E_POP     1     47481 1386491 4057.0
## - GATE      1     54537 1393547 4059.6
## - PAX       1     57754 1396764 4060.8
## - S_POP     1     62447 1401457 4062.5
## - SW        1    167564 1506574 4099.4
## - HI        1    184110 1523120 4104.9
## - VACATION  1    276682 1615692 4135.0
## - DISTANCE  1   1709875 3048885 4458.9
```

```
##
## Step:  AIC=4040.05
## FARE ~ NEW + VACATION + SW + HI + E_INCOME + S_POP + E_POP +
##     SLOT + GATE + DISTANCE + PAX
##
##            Df Sum of Sq     RSS    AIC
## - NEW       1      4517 1345632 4039.8
## <none>                  1341115 4040.0
## - E_INCOME  1     34205 1375320 4050.9
## - SLOT      1     38454 1379568 4052.5
## - E_POP     1     45877 1386992 4055.2
## - GATE      1     55388 1396502 4058.7
## - PAX       1     55652 1396767 4058.8
## - S_POP     1     72084 1413199 4064.7
## - HI        1    185949 1527063 4104.3
## - SW        1    197728 1538842 4108.2
## - VACATION  1    307447 1648561 4143.3
## - DISTANCE  1   1714666 3055781 4458.0
##
## Step:  AIC=4039.76
## FARE ~ VACATION + SW + HI + E_INCOME + S_POP + E_POP + SLOT +
##     GATE + DISTANCE + PAX
##
##            Df Sum of Sq     RSS    AIC
## <none>                  1345632 4039.8
## - E_INCOME  1     32964 1378596 4050.1
## - SLOT      1     36615 1382246 4051.5
## - E_POP     1     46303 1391935 4055.0
## - GATE      1     54029 1399660 4057.8
## - PAX       1     55304 1400936 4058.3
## - S_POP     1     74031 1419662 4065.1
## - HI        1    184415 1530047 4103.3
## - SW        1    197192 1542824 4107.5
## - VACATION  1    304855 1650486 4141.9
## - DISTANCE  1   1711053 3056684 4456.2
```

```
summary(airfares.lm.backward)
```

```
##
## Call:
## lm(formula = FARE ~ VACATION + SW + HI + E_INCOME + S_POP + E_POP +
##     SLOT + GATE + DISTANCE + PAX, data = train.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -150.238  -33.021   -2.639   30.484  162.591
##
## Coefficients:
##                   Estimate    Std. Error t value             Pr(>|t|)
## (Intercept)  45.020461949  21.640727151   2.080             0.038002 *
## VACATIONYes -60.032628715   5.646165922 -10.632 < 0.0000000000000002 ***
## SWYes       -50.585400506   5.915521572  -8.551 < 0.0000000000000002 ***
## HI            0.012641329   0.001528646   8.270  0.00000000000000123 ***
## E_INCOME      0.002096705   0.000599695   3.496             0.000514 ***
```

```
## S_POP         0.000005514   0.000001052   5.240  0.00000023792617790 ***
## E_POP         0.000004795   0.000001157   4.144  0.00004013581083834 ***
## SLOTFree    -23.261008853   6.312675560  -3.685             0.000254 ***
## GATEFree    -28.876998830   6.451389759  -4.476  0.00000942774209450 ***
## DISTANCE      0.106035178   0.004209506  25.189 < 0.0000000000000002 ***
## PAX          -0.000990785   0.000218783  -4.529  0.00000743454723143 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 51.93 on 499 degrees of freedom
## Multiple R-squared:  0.7636, Adjusted R-squared:  0.7588
## F-statistic: 161.2 on 10 and 499 DF,  p-value: < 0.00000000000000022
```

```
airfares.lm.backward.AIC <- stepAIC(airfares.lm,direction ="backward")
```

```
## Start:  AIC=4043.05
## FARE ~ COUPON + NEW + VACATION + SW + HI + S_INCOME + E_INCOME +
##     S_POP + E_POP + SLOT + GATE + DISTANCE + PAX
##
##            Df Sum of Sq     RSS    AIC
## - COUPON    1       523 1339010 4041.2
## - S_INCOME  1      2288 1340775 4041.9
## - NEW       1      4184 1342671 4042.6
## <none>                  1338487 4043.0
## - SLOT      1     34783 1373270 4054.1
## - E_INCOME  1     36287 1374774 4054.7
## - PAX       1     44129 1382616 4057.6
## - E_POP     1     46174 1384661 4058.3
## - GATE      1     54087 1392574 4061.2
## - S_POP     1     62925 1401412 4064.5
## - SW        1    164672 1503159 4100.2
## - HI        1    173271 1511758 4103.1
## - VACATION  1    275438 1613925 4136.5
## - DISTANCE  1    891835 2230322 4301.5
##
## Step:  AIC=4041.25
## FARE ~ NEW + VACATION + SW + HI + S_INCOME + E_INCOME + S_POP +
##     E_POP + SLOT + GATE + DISTANCE + PAX
##
##            Df Sum of Sq     RSS    AIC
## - S_INCOME  1      2105 1341115 4040.0
## - NEW       1      4477 1343487 4040.9
## <none>                  1339010 4041.2
## - SLOT      1     35531 1374541 4052.6
## - E_INCOME  1     35928 1374938 4052.7
## - E_POP     1     47481 1386491 4057.0
## - GATE      1     54537 1393547 4059.6
## - PAX       1     57754 1396764 4060.8
## - S_POP     1     62447 1401457 4062.5
## - SW        1    167564 1506574 4099.4
## - HI        1    184110 1523120 4104.9
## - VACATION  1    276682 1615692 4135.0
## - DISTANCE  1   1709875 3048885 4458.9
##
```

```
## Step:  AIC=4040.05
## FARE ~ NEW + VACATION + SW + HI + E_INCOME + S_POP + E_POP +
##     SLOT + GATE + DISTANCE + PAX
##
##            Df Sum of Sq     RSS    AIC
## - NEW       1      4517 1345632 4039.8
## <none>                  1341115 4040.0
## - E_INCOME  1     34205 1375320 4050.9
## - SLOT      1     38454 1379568 4052.5
## - E_POP     1     45877 1386992 4055.2
## - GATE      1     55388 1396502 4058.7
## - PAX       1     55652 1396767 4058.8
## - S_POP     1     72084 1413199 4064.7
## - HI        1    185949 1527063 4104.3
## - SW        1    197728 1538842 4108.2
## - VACATION  1    307447 1648561 4143.3
## - DISTANCE  1   1714666 3055781 4458.0
##
## Step:  AIC=4039.76
## FARE ~ VACATION + SW + HI + E_INCOME + S_POP + E_POP + SLOT +
##     GATE + DISTANCE + PAX
##
##            Df Sum of Sq     RSS    AIC
## <none>                  1345632 4039.8
## - E_INCOME  1     32964 1378596 4050.1
## - SLOT      1     36615 1382246 4051.5
## - E_POP     1     46303 1391935 4055.0
## - GATE      1     54029 1399660 4057.8
## - PAX       1     55304 1400936 4058.3
## - S_POP     1     74031 1419662 4065.1
## - HI        1    184415 1530047 4103.3
## - SW        1    197192 1542824 4107.5
## - VACATION  1    304855 1650486 4141.9
## - DISTANCE  1   1711053 3056684 4456.2
```

```
summary(airfares.lm.backward.AIC)
```

```
##
## Call:
## lm(formula = FARE ~ VACATION + SW + HI + E_INCOME + S_POP + E_POP +
##     SLOT + GATE + DISTANCE + PAX, data = train.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -150.238  -33.021   -2.639   30.484  162.591
##
## Coefficients:
##                  Estimate    Std. Error t value             Pr(>|t|)
## (Intercept)  45.020461949  21.640727151   2.080             0.038002 *
## VACATIONYes -60.032628715   5.646165922 -10.632 < 0.0000000000000002 ***
## SWYes       -50.585400506   5.915521572  -8.551 < 0.0000000000000002 ***
## HI            0.012641329   0.001528646   8.270  0.00000000000000123 ***
## E_INCOME      0.002096705   0.000599695   3.496             0.000514 ***
## S_POP         0.000005514   0.000001052   5.240  0.00000023792617790 ***
```

```
## E_POP          0.000004795   0.000001157   4.144  0.00004013581083834 ***
## SLOTFree     -23.261008853   6.312675560  -3.685             0.000254 ***
## GATEFree     -28.876998830   6.451389759  -4.476  0.00000942774209450 ***
## DISTANCE       0.106035178   0.004209506  25.189 < 0.0000000000000002 ***
## PAX           -0.000990785   0.000218783  -4.529  0.00000743454723143 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 51.93 on 499 degrees of freedom
## Multiple R-squared:  0.7636, Adjusted R-squared:  0.7588
## F-statistic: 161.2 on 10 and 499 DF,  p-value: < 0.00000000000000022
```

Answers 1)Distance seems to be the best predictor for Fare as it as the highest value in the correlation table 2)Vacation NO - 73.35%,Yes - 26.65% SW No - 69.59%, Yes-30.41% SLOT Controlled - 28.53% Free - 71.47% GATE Constrained - 19.44% Free - 80.56% Vacation and SW seems to be the best predictors for FARE as they are the only 2 categorical variables that are statistically significant when we run the linear regression model

4)To use the best combination of variables that is to be used while running the regression, we have to choose a combination of variables where the AIC is the lowest. In this results we get the lowest AIC as 4039.76 when we use the combination of VACATION + SW + HI + E_INCOME + S_POP + E_POP + SLOT + GATE + DISTANCE + PAX From the coefficients table we see that all the values of these variables are statistically significant. Adjusted R square value is 0.7588 which tells us that this dataset accounts for 76% of the variation. The F statistic is also statistically significant which implies that atleast one variable in this combination is impacting the dependent variable significantly.

5)When we use the exhaustive search model we see that the Adjusted R square is the highest for the 11th combination. Hence when take the 11th combination of variables into account.(i.e) NEW,VACATION,SW,HI,E_INCOME,S_POP,E_POP,SLOT,GATE,DISTANCE,PAX. In comparison to the stepwise search we see that we have NEW variable added when we did exhaustive search.

6)The RMSE value of the model determined by the Exhaustive search is lesser than the model determined by the stepwise regression. So the model determined by the exhaustive search regression is better.

7)$343.59

8)The reduction should be about USD50.65 when the SW decided to cover the route.The exact price that we get when we run the model with SW variable set to yes is USD292.9361

9)We choose a model with the least AIC. The combination VACATION + SW + HI + E_INCOME + S_POP + E_POP + SLOT + GATE + DISTANCE + PAX has the least AIC. From the coefficients table we see that all the variables in this particular combination are statistically significant in terms of impacting the Fare

10)We choose a model with the least AIC. The combination VACATION + SW + HI + E_INCOME + S_POP + E_POP + SLOT + GATE + DISTANCE + PAX has the least AIC. From the coefficients table we see that all the variables in this particular combination are statistically significant in terms of impacting the Fare.

AIC gives us which combination of variables is a better fit. The lower the AIC, the better