cycleparking

Yuhei Ito

2020/10/7

Read me

input data

census.rda: A census data of 2011 subsetting by Greater London. route.rda: route data obtained from PCT. parking.rda: parking data merged from CID and OSM. lsoa_msoa.rda: LSOA and MSOA link.

Load data

```
# Load census data.
census = paste(getwd(),"/data/census.rda", sep = "")
load(census)

# Load route data.
route = paste(getwd(),"/data/route.rda", sep = "")
load(route)

# Load parking data.
parking = paste(getwd(),"/data/parking.rda", sep = "")
load(parking)

# Load area data.
area = paste(getwd(),"/data/area.rda", sep = "")
load(area)

# Load lsoa_msoa data.
lsoa_msoa = paste(getwd(),"/data/lsoa_msoa.rda", sep = "")
load(lsoa_msoa)

# MSOA shapefile.
msoa = st_read("C:\\Users\\yuhei\\Documents\\R\\cycleparking\\data\\msoa.shp")
```

```
## Reading layer ‘msoa’ from data source ‘C:\Users\yuhei\Documents\R\cycleparking\data\msoa.shp’ using d
## Simple feature collection with 983 features and 1 field
## geometry type:  MULTIPOLYGON
## dimension:      XY
## bbox:           xmin: 503574.2 ymin: 155850.8 xmax: 561956.7 ymax: 200933.6
## epsg (SRID):    NA
## proj4string:    +proj=tmerc +lat_0=49 +lon_0=-2 +k=0.9996012717 +x_0=400000 +y_0=-100000 +datum=OSGB3
```

Join MSOA code to census

```r
# Join Origin MSOA
census = left_join(census, lsoa_msoa, by = c("olsoa" = "lsoa"))
any(is.na(census$msoa)) # False, join successful.
```

```
## [1] FALSE
```

```r
census$omsoa = census$msoa
census = census[,c("olsoa","omsoa","dlsoa","all","bicycle")]

# Join Destination MSOA
census = left_join(census, lsoa_msoa, by = c("dlsoa" = "lsoa"))
any(is.na(census$msoa)) # False, join successful.
```

```
## [1] FALSE
```

```r
census$dmsoa = census$msoa
census = census[,c("olsoa","omsoa","dlsoa","dmsoa","all","bicycle")]
```

Join MSOA code to parking

```r
parking = left_join(parking, lsoa_msoa, by = c("lsoa" = "lsoa"))
any(is.na(parking$msoa)) # False, join successful.
```

```
## [1] FALSE
```

```r
parking = parking[,c("lsoa","msoa","space")]
```

Aggregate census' all and bicycle by MSOA.

```r
census2 =
  census %>%
  group_by(omsoa,dmsoa)%>%
  summarise(all = sum(all), bicycle = sum(bicycle))
```

```
## `summarise()` regrouping output by 'omsoa' (override with `.groups` argument)
```

```r
sum(census2$all) == sum(census$all) #TRUE, successful.
```

```
## [1] TRUE
```

Add distance and hilliness to census2.

```
# Delete rows with rf_dist_km and rf_avslope_perc are NA. These NA rows have Other or OD0000003 written
route =
  route %>%
  filter(!is.na(rf_dist_km))
any(is.na(route)) #FALSE, successful.
```

```
## [1] FALSE
```

```
# CREATE JOIN ID.
# 1. Create join ID for Origin = geo_code1, Destination = geo_code2, ID = OD
route12 = route
route12$id = paste(route$geo_code1, route$geo_code2)
nrow(route12) #308782
```

```
## [1] 308782
```

```
# 2. Create join ID for Origin = geo_code2, Destination = geo_code1, ID = OD
route21 = route
route21$id = paste(route$geo_code2, route$geo_code1)
nrow(route21) #308782
```

```
## [1] 308782
```

```
# 3. Combine two dfs.
route = rbind(route12,route21)
nrow(route) #617564
```

```
## [1] 617564
```

```
# Remove duplication of ID which are intra-zonal OD pairs. Avoid adding duplicated rows when joining to
route =
  route %>% distinct(id, .keep_all = TRUE)

# Delete unnecessary column
route = route[,c("id", "rf_dist_km", "rf_avslope_perc")]
```

Join distance and hilliness to census2.

```
# Create join id for Census data.
census2$id = paste(census2$omsoa, census2$dmsoa)

# Join route data.
census2 = left_join(census2, route, by = "id")

# Delete id column
census2 =
  census2 %>%
  select(-id)

any(is.na(census2)) #TRUE
```

```
## [1] TRUE
```

```
  census2 %>%
  filter(is.na(rf_dist_km))
```

```
## # A tibble: 219,452 x 6
## # Groups:   omsoa [7,147]
##    omsoa     dmsoa       all bicycle rf_dist_km rf_avslope_perc
##    <chr>     <chr>     <int>   <int>      <dbl>           <dbl>
##  1 E02000001 E02000509     1       0         NA              NA
##  2 E02000002 E02000053     2       0         NA              NA
##  3 E02000002 E02000093     3       0         NA              NA
##  4 E02000002 E02000106     2       0         NA              NA
##  5 E02000002 E02000116     1       0         NA              NA
##  6 E02000002 E02000119     2       0         NA              NA
##  7 E02000002 E02000150     1       0         NA              NA
##  8 E02000002 E02000220     1       0         NA              NA
##  9 E02000002 E02000225     1       0         NA              NA
## 10 E02000002 E02000244     1       0         NA              NA
## # ... with 219,442 more rows
```

Check census data.

```
# Check if there is any No fixed place data on Census. -> No.
nrow(census2) #601902 rows.
```

```
## [1] 601902
```

```
census2 %>%
  mutate(First3D = substring(dmsoa,1,3)) %>% # Obtain first three letters of destination MSOA.
  filter(First3D == "E02") # Check if all rows are E02. 601902 rows Correct.
```

```
## # A tibble: 601,902 x 7
## # Groups:   omsoa [7,189]
##    omsoa     dmsoa       all bicycle rf_dist_km rf_avslope_perc First3D
##    <chr>     <chr>     <int>   <int>      <dbl>           <dbl> <chr>
##  1 E02000001 E02000001  1506      33       1.05            1.64 E02
##  2 E02000001 E02000014     2       0      19.1             0.77 E02
##  3 E02000001 E02000016     3       0      13.9             0.87 E02
##  4 E02000001 E02000025     1       0      19.4             1.66 E02
##  5 E02000001 E02000028     1       0      18.5             1.58 E02
##  6 E02000001 E02000051     1       0      13               1.87 E02
##  7 E02000001 E02000053     2       0      15.8             2.34 E02
##  8 E02000001 E02000057     1       0      12.8             2.28 E02
##  9 E02000001 E02000058     1       0      12               2.31 E02
## 10 E02000001 E02000059     1       0      14.8             1.6  E02
## # ... with 601,892 more rows
```

```
census2 %>%
  mutate(First3O = substring(omsoa,1,3)) %>% # Obtain first three letters of origin MSOA.
  group_by(First3O)%>%
  summarise(n(),sum(all),sum(bicycle)) # There are 4570 (bicycle n=97) commuters come from Wales.
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 4
##   First3O  `n()` `sum(all)` `sum(bicycle)`
##   <chr>    <int>      <int>          <int>
## 1 E02     598276    3711412         147102
## 2 W02       3626       4570             97
```

```
# Identify intra-zonal trip.
census2$intra = ifelse(census2$omsoa == census2$dmsoa, TRUE, FALSE)
census2 %>%
  filter(intra == TRUE) %>%
  nrow() #983 rows
```

```
## [1] 983
```

```
# Excluded rows?
census2$excluded = ifelse(is.na(census2$rf_dist_km), TRUE, FALSE)
```

Check how many trips are excluded and how many are intra/inter zonal trips. (Table 6. Number of Commuters in/to Greater London)

```
# Total amount of travel to GL before excluding any trips.
sum(census2$all) # 3715982
```

```
## [1] 3715982
```

```
# Intra and Inter zonal trips
census2 %>%
  group_by(intra) %>%
  summarise(n(),sum(all), sum(bicycle))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 2 x 4
##   intra  `n()` `sum(all)` `sum(bicycle)`
##   <lgl>  <int>      <int>          <int>
## 1 FALSE 600919    3589916         143899
## 2 TRUE     983     126066           3300
```

```
# Excluded and Included to the study.
census2 %>%
  group_by(excluded) %>%
  summarise(n(),sum(all), sum(bicycle))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 2 x 4
##   excluded  `n()` `sum(all)` `sum(bicycle)`
##   <lgl>     <int>      <int>          <int>
## 1 FALSE    382450    3145728         143182
## 2 TRUE     219452     570254           4017
```

```
# Intra/Inter and Exclude/Include.
census2 %>%
  group_by(excluded, intra) %>%
  summarise(n(),sum(all), sum(bicycle))
```

```
## `summarise()` regrouping output by 'excluded' (override with `.groups` argument)

## # A tibble: 3 x 5
## # Groups:   excluded [2]
##   excluded intra  `n()` `sum(all)` `sum(bicycle)`
##   <lgl>    <lgl>  <int>      <int>          <int>
## 1 FALSE    FALSE 381467    3019662         139882
## 2 FALSE    TRUE     983     126066           3300
## 3 TRUE     FALSE 219452     570254           4017
```

```
219452/nrow(census2)*100      # 36% OD pairs
```

```
## [1] 36.45976
```

```
570254/sum(census2$all)*100   # 15% all commuters
```

```
## [1] 15.34598
```

```
4017/sum(census2$bicycle)*100 # 3%  cycle commuters
```

```
## [1] 2.728959
```

BCG construction Step 1: Eliminating >30km od and CalculateGoDutch propensity

```
# Eliminate >30km OD pairs
census2 =
  census2 %>% #601902 od pairs
  filter(excluded == FALSE) %>% #382450 od pairs
  select(-c("excluded"))

# Calculate parameter for Go Dutch.
census2 =
  census2 %>%
  mutate(logit_gd =
```

```
            -3.959 +
            (-0.5963 * rf_dist_km) +
            (1.866 * sqrt(rf_dist_km)) +
            (0.008050 * rf_dist_km^2) +
            (-0.2710 * rf_avslope_perc) +
            (0.009394 * rf_dist_km*rf_avslope_perc) +
            (-0.05135 * sqrt(rf_dist_km) *rf_avslope_perc) +
            2.523 +
            (-0.07626*rf_dist_km)
          )

# Calculate GD propensity.
census2$GDpct = exp(census2$logit_gd) / (1 + exp(census2$logit_gd))
```

BCG construction Step 2: Calculate GoDutch number of cycle commuters for each OD pair.

```
# Calculate future cycle commuter no. based on Go Dutch scenario.
census2 =
  census2%>%
  mutate(bicycle_gd = all*GDpct)

# Check
census2 %>%
  mutate(check = bicycle_gd > bicycle) %>%
  group_by(check) %>%
  summarise(n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 2 x 2
##   check  `n()`
##   <lgl>  <int>
## 1 FALSE  22467
## 2 TRUE  359983
```

```
#FALSE  22467   OD pairs have already achieved Dutch propensity.
#TRUE   359983

census2 %>%
  filter(bicycle > bicycle_gd) %>%
  arrange(desc(bicycle)) %>%
  nrow() #22467
```

```
## [1] 22467
```

```
nrow(census2) #382450
```

```
## [1] 382450
```

```
22467/382450 #0.05874493
```

```
## [1] 0.05874493
```

```
census2 %>%
  mutate(OverDutch = ifelse(bicycle > bicycle_gd,TRUE, FALSE)) %>%
  group_by(OverDutch)%>%
  summarise(bicycle = sum(bicycle), bicycle_gd = sum(bicycle_gd))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 2 x 3
##   OverDutch bicycle bicycle_gd
##   <lgl>       <int>      <dbl>
## 1 FALSE      101615    572379.
## 2 TRUE        41567     21290.
```

```
41567- 21290 #20277
```

```
## [1] 20277
```

```
# Actual future share is assumed to be unchanged for OD pairs which already achieved the Dutch propensi
census2$bicycle_future =
  ifelse(census2$bicycle_gd > census2$bicycle,
         census2$bicycle_gd,
         census2$bicycle)

# Rearrange columns
names(census2)
```

```
##  [1] "omsoa"           "dmsoa"           "all"
##  [4] "bicycle"         "rf_dist_km"      "rf_avslope_perc"
##  [7] "intra"           "logit_gd"        "GDpct"
## [10] "bicycle_gd"      "bicycle_future"
```

```
census2 =
  census2[,c("omsoa","dmsoa","all","bicycle","bicycle_future","bicycle_gd","GDpct","logit_gd","rf_dist_
```

BCG construction Step 3: Aggregate by destination MSOA.

```
# Aggregate based on destination MSOA.
census3 =
  census2 %>%
  select(c(1:7))%>%
  group_by(dmsoa)%>%
  summarise(all = sum(all),
            bicycle = sum(bicycle),
            bicycle_future = sum(bicycle_future),
            bicycle_gd = sum(bicycle_gd))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
# How many MSOAs in GL?
length(unique(census2$dmsoa)) #983
```

```
## [1] 983
```

```
# Check
sum(census2$all) == sum(census3$all) # TRUE
```

```
## [1] TRUE
```

```
sum(census2$bicycle) == sum(census3$bicycle) # TRUE
```

```
## [1] TRUE
```

```
sum(census2$bicycle_future) == sum(census3$bicycle_future) # TRUE
```

```
## [1] TRUE
```

```
sum(census2$bicycle_gd) == sum(census3$bicycle_gd) # TRUE
```

```
## [1] TRUE
```

BCG construction Step 4: Construct BCG by mean values - Growth Share version

```
# Calculate current share for each MSOA
census3$CurrentShare = census3$bicycle/census3$all

# Calculate future share for each LSOA.
census3$FutureShare = census3$bicycle_future/census3$all

# Calculate growth share and add a new column.
census3$Growth = census3$FutureShare - census3$CurrentShare

# Horizontal axis (Current AT share)
# If the share is lower than the mean, 1 otherwise 2.

BCG_x = ifelse(
  census3$CurrentShare < mean(census3$CurrentShare),
  1,2)

# Vertical axis (Growth AT share)
# If the share is lower than the mean, 1 otherwise 2.

BCG_y = ifelse(
  census3$Growth < mean(census3$Growth),
```

```
  1,2)

# Combine the above two.
BCG = paste(BCG_x,BCG_y,sep="")

# Rename the codes to BCG terms.
BCG = recode(BCG,
             '11' = "Problematic",
             '12' = "Promising",
             '21' = "Achieved",
             '22' = "Maturing")

# Add a column.
census3$BCG = BCG

census3$BCG = factor(census3$BCG,
               levels = c("Achieved","Maturing","Promising","Problematic"))

# Current, Future Share and growth max.
max(census3$CurrentShare) #0.148398
```

```
## [1] 0.148398
```

```
max(census3$FutureShare) #0.3400569
```

```
## [1] 0.3400569
```

```
max(census3$Growth) #0.3242483
```

```
## [1] 0.3242483
```

BCG construction Step 5: Construct BCG by mean values - Growth amount version

```
# Calculate growth amount and add a new column.
census3$GrowthNos = census3$bicycle_future - census3$bicycle

# Horizontal axis (Current AT share)
# If the share is lower than the mean, 1 otherwise 2.

BCG_x = ifelse(
  census3$CurrentShare < mean(census3$CurrentShare),
  1,2)

# Vertical axis (Growth AT share)
# If the share is lower than the mean, 1 otherwise 2.

BCG_y = ifelse(
  census3$GrowthNos < mean(census3$GrowthNos),
```

```
  1,2)

# Combine the above two.
BCG = paste(BCG_x,BCG_y,sep="")

# Rename the codes to BCG terms.
BCG = recode(BCG,
             '11' = "Problematic",
             '12' = "Promising",
             '21' = "Achieved",
             '22' = "Maturing")

# Add a column.
census3$BCGNos = BCG

census3$BCGNos = factor(census3$BCGNos,
                levels = c("Achieved","Maturing","Promising","Problematic"))

# Current, Future Share and growth max.
max(census3$CurrentShare) #0.148398
```

```
## [1] 0.148398
```

```
max(census3$FutureShare) #0.3400569
```

```
## [1] 0.3400569
```

```
max(census3$GrowthNos) #0.3242483
```

```
## [1] 21981.31
```

```
census3[,c("GrowthNos", "BCGNos")]
```

```
## # A tibble: 983 x 2
##    GrowthNos BCGNos
##        <dbl> <fct>
##  1    21981. Maturing
##  2      146. Problematic
##  3      628. Promising
##  4      240. Problematic
##  5      217. Problematic
##  6      773. Promising
##  7      408. Problematic
##  8      129. Problematic
##  9      232. Problematic
## 10      444. Problematic
## # ... with 973 more rows
```

```
tbd= census3[,c("GrowthNos", "BCGNos")]
tbd$BCG_y = BCG_y
rm(tbd)
```

Parking 1: Aggregate parking space for each MSOA

```
# the updated parking data
sum(parking$space) #140788
```

```
## [1] 140788
```

```
# Aggregate parking capacity based on MSOA
parking2 =
  parking %>%
  select(msoa, space) %>%
  group_by(msoa) %>%
  summarise(space = sum(space))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
sum(parking2$space) == sum(parking$space) #140788 TRUE
```

```
## [1] TRUE
```

Parking 2: Join parking space to census

```
census3 = left_join(census3, parking2, by = c("dmsoa" = "msoa"))
sum(census3$space) == sum(parking2$space) #TRUE Successful!
```

```
## [1] TRUE
```

Parking Occupancy Rate: por

4 Categories: Surplus : Less than 50% occupancy rate Ideal : 50% and more and 80% and less occupancy rate Threshold: More than 80% and 100% and less occupancy rate Shortage : More than 100% occupancy rate

Examples: Surplus : bicycle= 30, parking=100 - Less than 50% occupancy rate Ideal : bicycle= 70, parking=100 - 50% and more and 80% and less occupancy rate Threshold: bicycle= 90, parking=100 - More than 80% and 100% and less occupancy rate Shortage : bicycle=110, parking=100 - More than 100% occupancy rate

Exceptions: Case 1 : bicycle= 0, parking=100 -> Surplus (n = 22 current) Case 2 : bicycle=10, parking=0 -> Shortage (n = 1 current) Case 3 : bicycle= 0, parking=0 -> Shortage (n = 0 observation)

Current Occupancy Rate and Category

```
# Current Occupancy Rate
census3$porCurrent = census3$bicycle / census3$space

census3%>%
  filter(bicycle == 0)
```

```
## # A tibble: 1 x 13
##   dmsoa   all bicycle bicycle_future bicycle_gd CurrentShare FutureShare
##   <chr> <int>   <int>          <dbl>      <dbl>        <dbl>       <dbl>
## 1 E020~   464       0           92.2       92.2            0       0.199
## # ... with 6 more variables: Growth <dbl>, BCG <fct>, GrowthNos <dbl>,
## #   BCGNos <fct>, space <int>, porCurrent <dbl>
```

```
census3%>%
  filter(space == 0)
```

```
## # A tibble: 22 x 13
##    dmsoa   all bicycle bicycle_future bicycle_gd CurrentShare FutureShare
##    <chr> <int>   <int>          <dbl>      <dbl>        <dbl>       <dbl>
##  1 E020~   811      12          244.       243.       0.0148       0.300
##  2 E020~   441      11           78.3       73.8      0.0249       0.178
##  3 E020~  1259      28          191.       173.       0.0222       0.151
##  4 E020~   819      14          172.       167.       0.0171       0.210
##  5 E020~   715      14          152.       148.       0.0196       0.213
##  6 E020~   669      13          184.       183.       0.0194       0.274
##  7 E020~  1186      25          253.       252.       0.0211       0.213
##  8 E020~   467      12           98.2       92.8      0.0257       0.210
##  9 E020~   362       3           61.0       60.1      0.00829      0.168
## 10 E020~  1584      35          314.       305.       0.0221       0.198
## # ... with 12 more rows, and 6 more variables: Growth <dbl>, BCG <fct>,
## #   GrowthNos <dbl>, BCGNos <fct>, space <int>, porCurrent <dbl>
```

```
# Categorise into 4.
## 2011 Occupancy Rate
census3 =
  census3 %>%
    mutate(porCurrentCat = ifelse(porCurrent > 1, "Shortage",
                           ifelse(0.8 < porCurrent & porCurrent <= 1, "Threshold",
                                  ifelse(0.5 <= porCurrent & porCurrent <= 0.8, "Ideal",
                                         ifelse(0.5 > porCurrent, "Surplus",NA))))
           )
census3
```

```
## # A tibble: 983 x 14
##    dmsoa    all bicycle bicycle_future bicycle_gd CurrentShare FutureShare
##    <chr>  <int>   <int>          <dbl>      <dbl>        <dbl>       <dbl>
##  1 E020~ 258804   13053         35034.     34712.       0.0504       0.135
##  2 E020~    585       7           153.       151.       0.0120       0.261
##  3 E020~   2473      49           677.       674.       0.0198       0.274
##  4 E020~    926      19           259.       257.       0.0205       0.279
##  5 E020~    809      18           235.       234.       0.0222       0.290
##  6 E020~   2826      51           824.       822.       0.0180       0.292
##  7 E020~   1421      49           457.       453.       0.0345       0.321
##  8 E020~    435      11           140.       138.       0.0253       0.322
##  9 E020~    811      12           244.       243.       0.0148       0.300
## 10 E020~   1948      62           506.       501.       0.0318       0.260
## # ... with 973 more rows, and 7 more variables: Growth <dbl>, BCG <fct>,
## #   GrowthNos <dbl>, BCGNos <fct>, space <int>, porCurrent <dbl>,
## #   porCurrentCat <chr>
```

Future Occupancy Rate and Category

```
# Future Occupancy Rate
census3$porFuture = census3$bicycle_future / census3$space

# Categorise into 4.
## 2011 Occupancy Rate
census3 =
  census3 %>%
    mutate(porFutureCat = ifelse(porFuture > 1, "Shortage",
                           ifelse(0.8 < porFuture & porFuture <= 1, "Threshold",
                                  ifelse(0.5 <= porFuture & porFuture <= 0.8, "Ideal",
                                         ifelse(0.5 > porFuture, "Surplus",NA))))
           )

census3
```

```
## # A tibble: 983 x 16
##    dmsoa    all bicycle bicycle_future bicycle_gd CurrentShare FutureShare
##    <chr>  <int>   <int>          <dbl>      <dbl>        <dbl>       <dbl>
##  1 E020~ 258804   13053         35034.     34712.       0.0504       0.135
##  2 E020~    585       7           153.       151.       0.0120       0.261
##  3 E020~   2473      49           677.       674.       0.0198       0.274
##  4 E020~    926      19           259.       257.       0.0205       0.279
##  5 E020~    809      18           235.       234.       0.0222       0.290
##  6 E020~   2826      51           824.       822.       0.0180       0.292
##  7 E020~   1421      49           457.       453.       0.0345       0.321
##  8 E020~    435      11           140.       138.       0.0253       0.322
##  9 E020~    811      12           244.       243.       0.0148       0.300
## 10 E020~   1948      62           506.       501.       0.0318       0.260
## # ... with 973 more rows, and 9 more variables: Growth <dbl>, BCG <fct>,
## #   GrowthNos <dbl>, BCGNos <fct>, space <int>, porCurrent <dbl>,
## #   porCurrentCat <chr>, porFuture <dbl>, porFutureCat <chr>
```

por checking

```
### Check data
census3 %>%
  group_by(porCurrentCat)%>%
  summarise(n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 4 x 2
##   porCurrentCat `n()`
##   <chr>         <int>
## 1 Ideal           197
## 2 Shortage        290
## 3 Surplus         416
## 4 Threshold        80
```

```
census3 %>%
  group_by(porFutureCat)%>%
  summarise(n())
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 4 x 2
##   porFutureCat `n()`
##   <chr>        <int>
## 1 Ideal            5
## 2 Shortage       958
## 3 Surplus          4
## 4 Threshold       16
```

```
nrow(census3) #983
```

```
## [1] 983
```

```
census3 %>%
  group_by(porCurrentCat)%>%
  summarise(n = n())%>%
  select(n)%>%
  sum() #983
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## [1] 983
```

```
census3 %>%
  group_by(porFutureCat)%>%
  summarise(n = n())%>%
  select(n)%>%
  sum() #983
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## [1] 983
```

Reorder factor of por categorisation

```
# Convert to factor.
census3$porCurrentCat = as.factor(census3$porCurrentCat)
census3$porFutureCat  = as.factor(census3$porFutureCat)

# Check levels
levels(census3$porCurrentCat)
```

```
## [1] "Ideal"     "Shortage"  "Surplus"   "Threshold"
```

```
levels(census3$porFutureCat)
```

```
## [1] "Ideal"     "Shortage"  "Surplus"   "Threshold"
```

```
#"Ideal"     "Shortage"  "Surplus"   "Threshold"

# Reorder factor levels
census3$porCurrentCat = factor(census3$porCurrentCat,
                levels = c("Surplus","Ideal","Threshold","Shortage"))
census3$porFutureCat = factor(census3$porFutureCat,
                levels = c("Surplus","Ideal","Threshold","Shortage"))


# Check levels
levels(census3$porCurrentCat)
```

```
## [1] "Surplus"   "Ideal"     "Threshold" "Shortage"
```

```
levels(census3$porFutureCat)
```

```
## [1] "Surplus"   "Ideal"     "Threshold" "Shortage"
```

```
#"Surplus"   "Ideal"     "Threshold" "Shortage"
```

Join census 3 dataframe to shapefile msoa

```
msoa = left_join(msoa,census3, by = c("msoa" = "dmsoa"))
```

Set Up

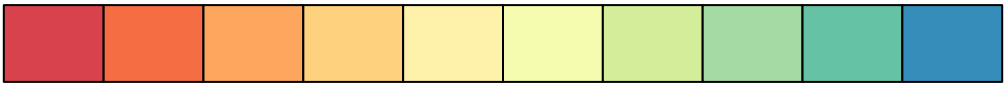
```
BCGcol = c( "#73d87d", "#368cd3", "#ffff01", "#ff3b40")
col_pocu = c("#368cd3", "#73d87d", "#ffff01" ,"#ff3b40")
dev.list()
```

```
## pdf
##   2
```

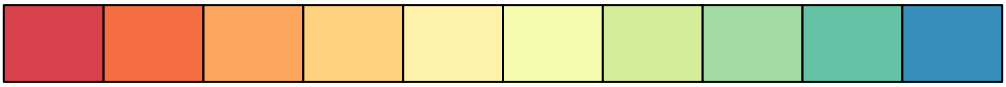
Figures

Figure 10 Destination Based Spatial Distribution of Current Cycle Commuter Share

```
get_brewer_pal("Spectral", n = 10, contrast = c(0, 0.75))
```




```
tm_shape(msoa) +
  tm_polygons(col = "CurrentShare",
              style = "fixed",
              breaks = c(0, 0.02,0.04,0.06,0.09,0.15,0.19,0.22,0.24,0.27,0.35),
              palette = get_brewer_pal("Spectral", n = 10, contrast = c(0, 0.75)),
              title = "Current Share \n of Cycle Commuter")+
  tm_layout(legend.outside = TRUE)+
  tm_compass(position = c("left", "bottom"))+
  tm_scale_bar(position = c("left", "bottom"))
```



#D8424D #F46D43 #FCA75D #FDD17E #FEF2AA #F5FBAF #D4ED9B #A4DAA4 #66C2A5 #378DBA

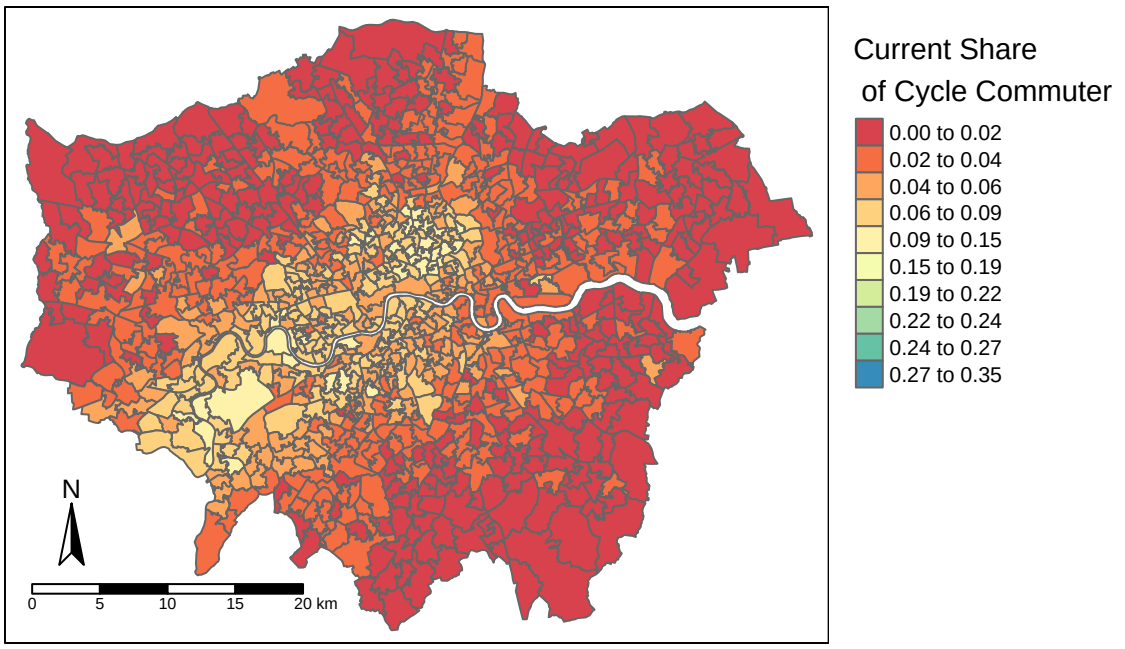
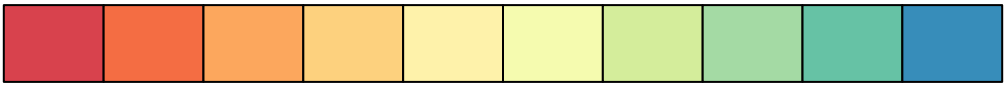


Figure 11 Destination Based Spatial Distribution of Future Cycle Commuter Share

```
tm_shape(msoa) +
  tm_polygons(col = "FutureShare",
              style = "fixed",
              breaks = c(0, 0.02,0.04,0.06,0.09,0.15,0.19,0.22,0.24,0.27,0.35),
              legend.hist = TRUE,
              palette = get_brewer_pal("Spectral", n = 10, contrast = c(0, 0.75)),
              title = "Future Share \n of Cycle Commuter")+
  tm_layout(legend.outside = TRUE) +
  tm_compass(position = c("left", "bottom"))+
  tm_scale_bar(position = c("left", "bottom"))
```



#D8424D #F46D43 #FCA75D #FDD17E #FEF2AA #F5FBAF #D4ED9B #A4DAA4 #66C2A5 #378DBA

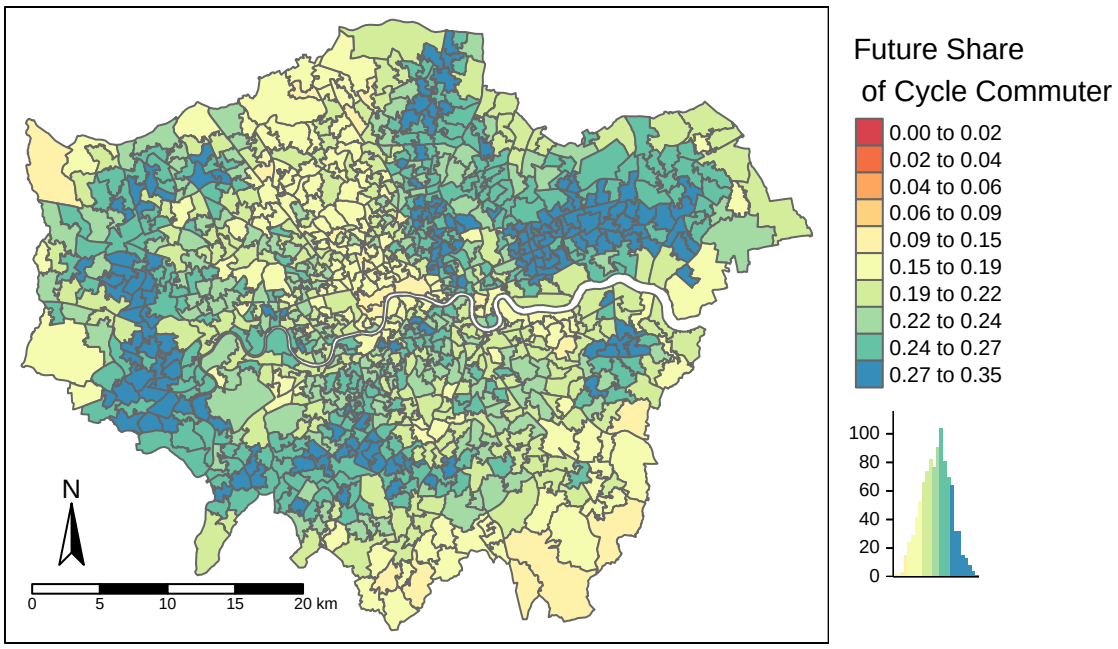
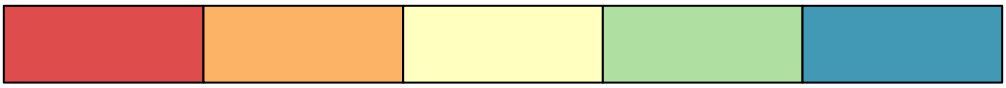


Figure 15 Destination Based Spatial Distribution of Growth Cycle Commuter Share

```
tm_shape(msoa) +
  tm_polygons(col = "Growth",
              style = "fixed",
              breaks = c(0.06, 0.16, 0.18, 0.20, 0.24, 0.33),
              legend.hist = TRUE,
              palette = get_brewer_pal("Spectral", n = 5, contrast = c(0, 0.75)),
              title = "Growth Share \n of Cycle Commuter")+
  tm_layout(legend.outside = TRUE) +
  tm_compass(position = c("left", "bottom"))+
  tm_scale_bar(position = c("left", "bottom"))
```


#DE4C4B
#FDB365
#FFFFBF
#B0DFA2
#4199B5

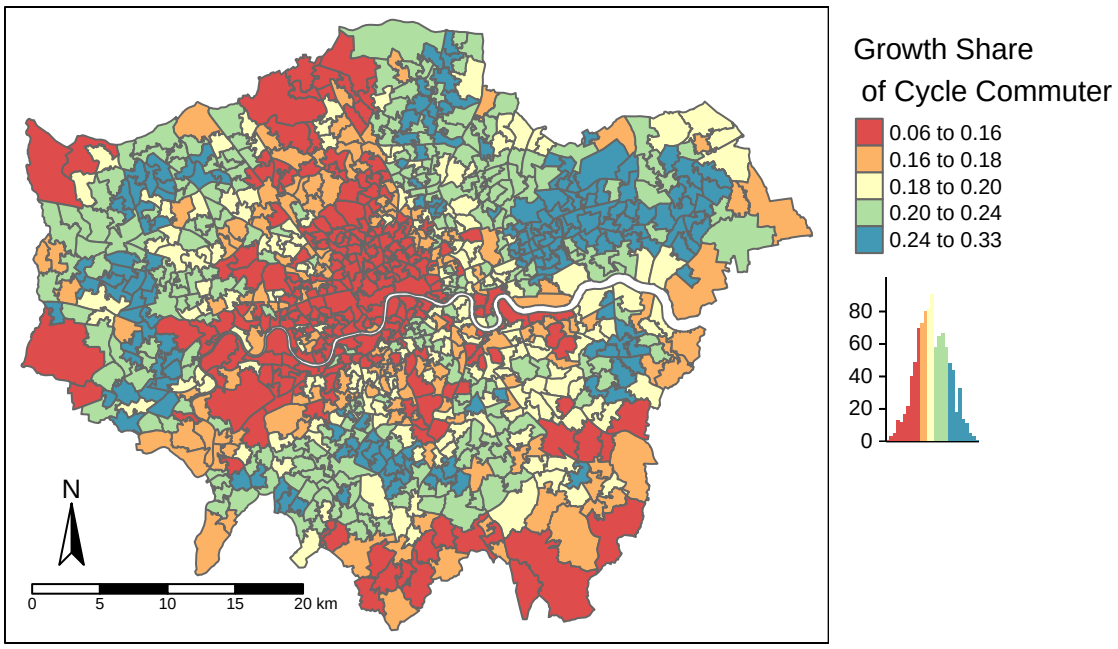


Figure 15-1 Destination Based Spatial Distribution of Growth Nos Cycle Commuter Share

```
tm_shape(msoa) +
  tm_polygons(col = "GrowthNos",
              title = "Growth Number \n of Cycle Commuter")+
  tm_layout(legend.outside = TRUE) +
  tm_compass(position = c("left", "bottom"))+
  tm_scale_bar(position = c("left", "bottom"))
```

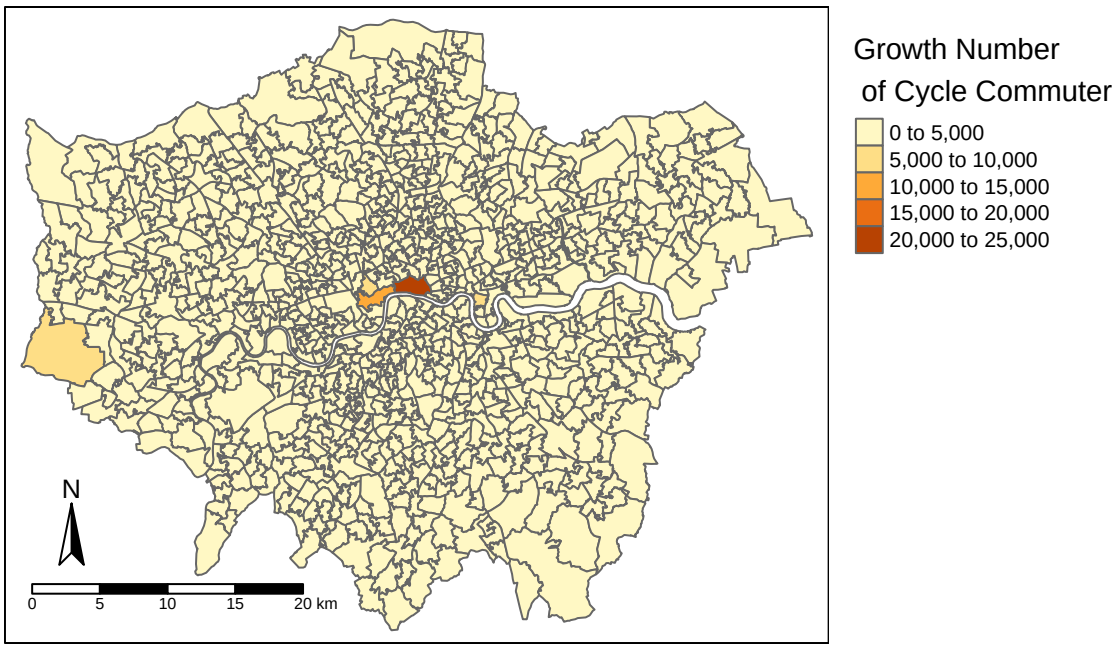
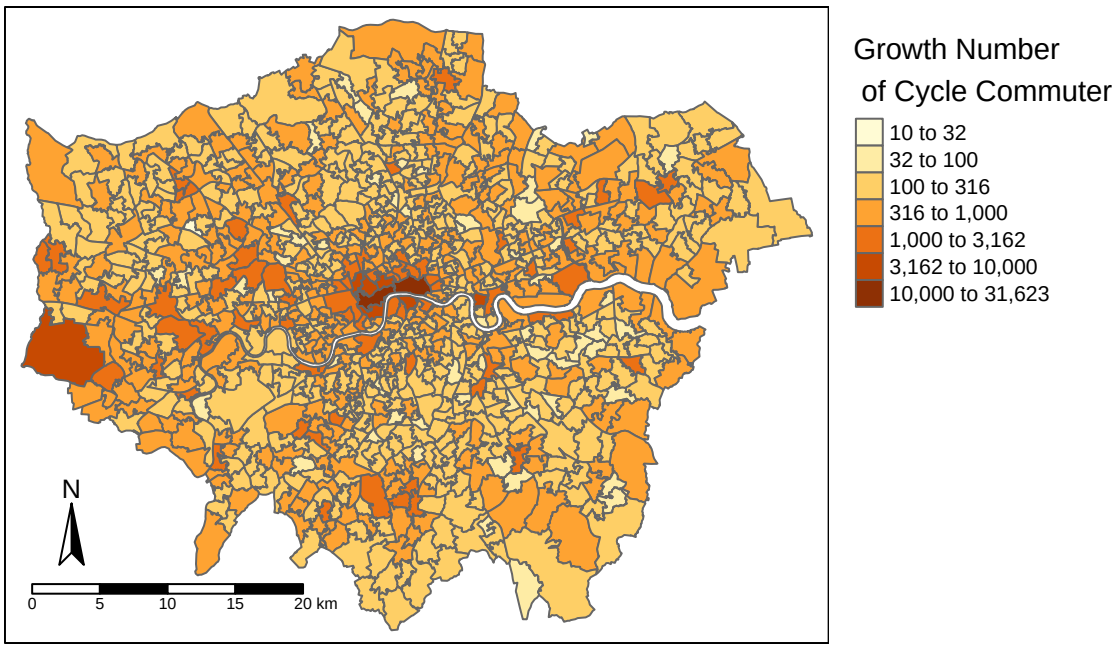


Figure 15-1-1 Destination Based Spatial Distribution of Growth Nos Cycle Commuter Share LOG SCALE!

```
tm_shape(msoa) +
  tm_polygons(col = "GrowthNos",
              style = "log10_pretty",
              title = "Growth Number \n of Cycle Commuter")+
  tm_layout(legend.outside = TRUE) +
  tm_compass(position = c("left", "bottom"))+
  tm_scale_bar(position = c("left", "bottom"))
```




```
tm_shape(msoa) +
  tm_polygons(col = "GrowthNos",
              style = "log10",
              title = "Growth Number \n of Cycle Commuter")+
  tm_layout(legend.outside = TRUE) +
  tm_compass(position = c("left", "bottom"))+
  tm_scale_bar(position = c("left", "bottom"))
```

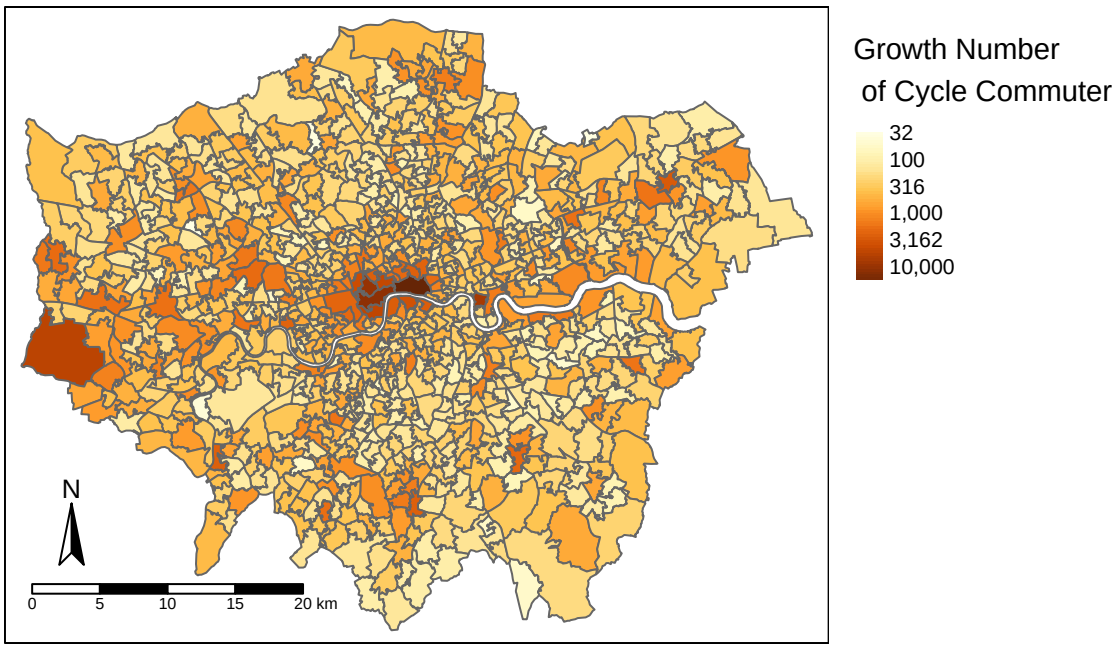
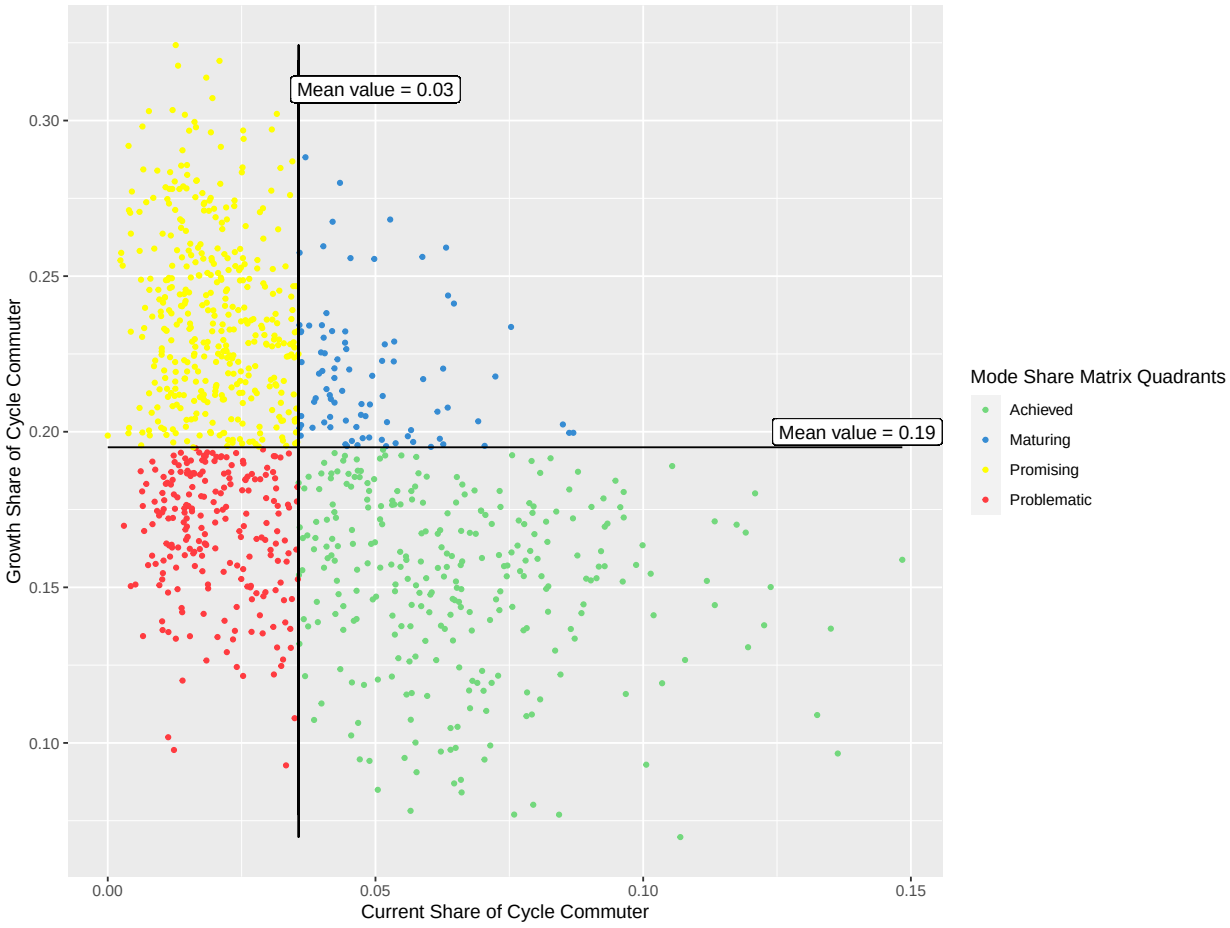


Figure 16 Scatter Plot of Growth and Current Share of Cycle Commuter

```
ggplot(census3, aes(x = CurrentShare, y = Growth)) +
  geom_point(aes(color = BCG), size = 1) +
  scale_colour_manual(values=BCGcol) +
  geom_line(aes(x = mean(CurrentShare))) +
  geom_line(aes(y = mean(Growth))) +
  labs(colour = "Mode Share Matrix Quadrants")+
  labs(y="Growth Share of Cycle Commuter", x = "Current Share of Cycle Commuter")+
  geom_label(label = "Mean value = 0.19",
             x = 0.14,
             y = 0.2
             )+
    geom_label(label = "Mean value = 0.03",
             x = 0.05,
             y = 0.31
             )
```




```
mean(census3$CurrentShare)*100 #3.563695
```

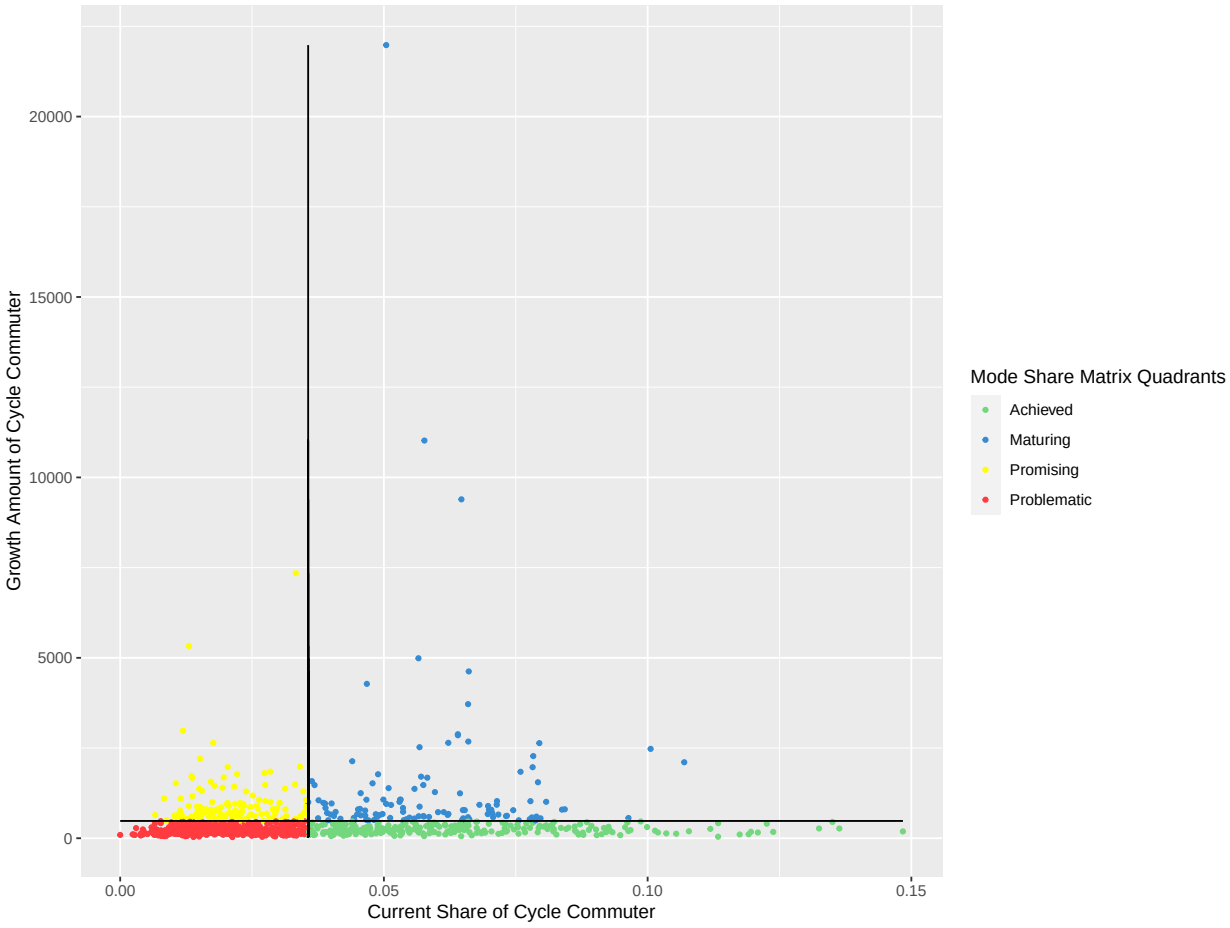
```
## [1] 3.563695
```

```
mean(census3$Growth)*100 #19.50014
```

```
## [1] 19.50014
```

Figure 16-1 Scatter Plot of Growth and Current Share of Cycle Commuter - Growth Amount Vesion

```
ggplot(census3, aes(x = CurrentShare, y = GrowthNos)) +
  geom_point(aes(color = BCGNos), size = 1) +
  scale_colour_manual(values=BCGcol) +
  geom_line(aes(x = mean(CurrentShare))) +
  geom_line(aes(y = mean(GrowthNos))) +
  labs(colour = "Mode Share Matrix Quadrants")+
  labs(y="Growth Amount of Cycle Commuter", x = "Current Share of Cycle Commuter")
```




```
mean(census3$CurrentShare)*100 #3.563695
```

```
## [1] 3.563695
```

```
mean(census3$GrowthNos) #478.905
```

```
## [1] 478.905
```

Figure 16-1-1 Scatter Plot of Growth and Current Share of Cycle Commuter - Growth Amount Vesion Log scale

```
test = ggplot(census3, aes(x = CurrentShare, y = GrowthNos)) +
  geom_point(aes(color = BCGNos), size = 1) +
  scale_colour_manual(values=BCGcol) +
  geom_line(aes(x = mean(CurrentShare))) +
  geom_line(aes(y = mean(GrowthNos))) +
  labs(colour = "Mode Share Matrix Quadrants")+
  labs(y="Growth Amount of Cycle Commuter", x = "Current Share of Cycle Commuter")+
  geom_label(label = "Mean value = 478",
             x = 0.14,
```

```
              y = 450
              )

mean(census3$CurrentShare) #0.03563695
```

```
## [1] 0.03563695
```

```
mean(census3$GrowthNos) #478.905
```

```
## [1] 478.905
```

```
test + scale_y_continuous(trans = 'log10')
```

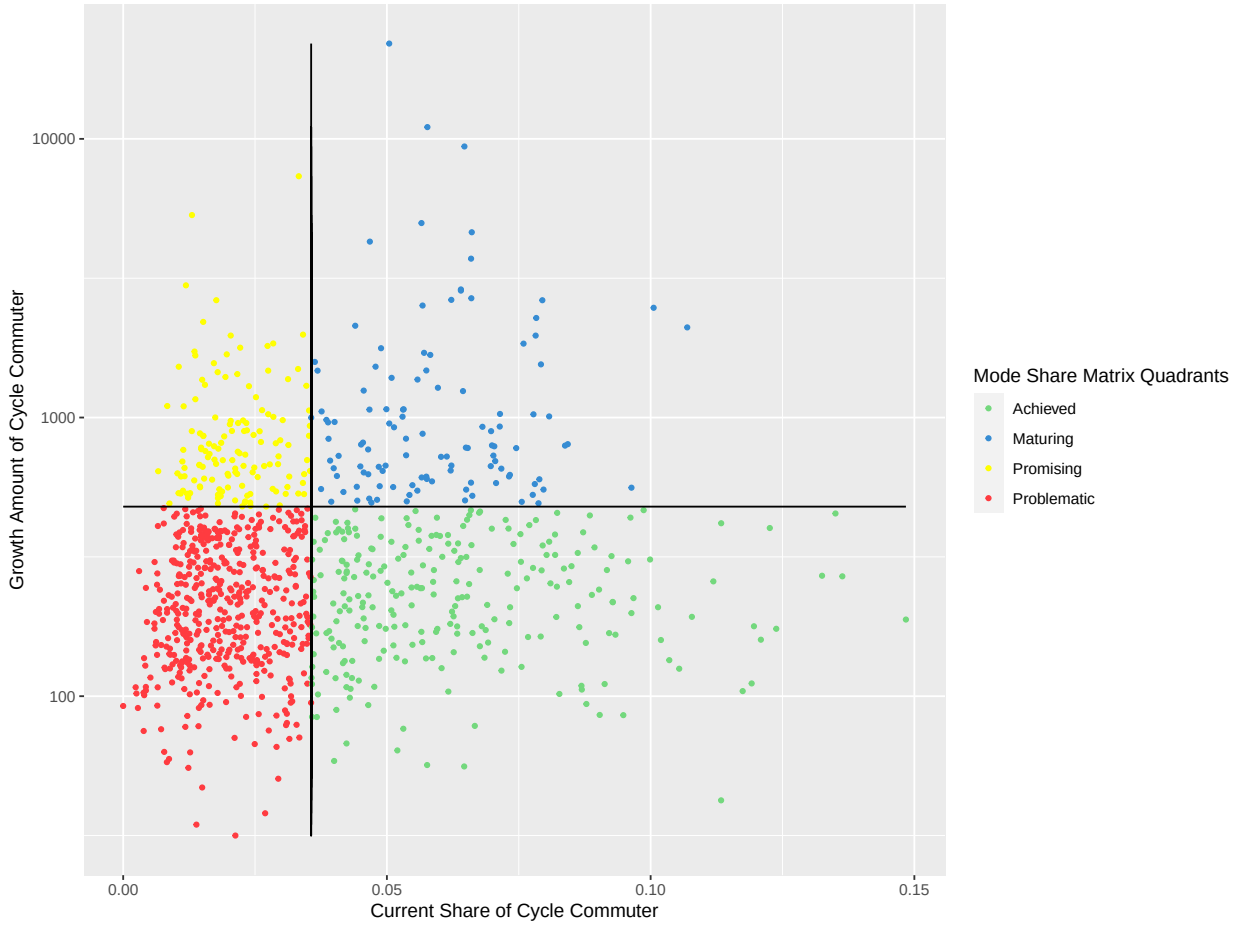


Figure 18 Spatial Distribution of BCG Categories

```
tm_shape(msoa) +
  tm_polygons(col = "BCG", palette = BCGcol,
              title = "Mode Share\nMatrix Quadrants")+
  tm_compass(position = c("left", "bottom"))+
  tm_scale_bar(position = c("left", "bottom"))
```

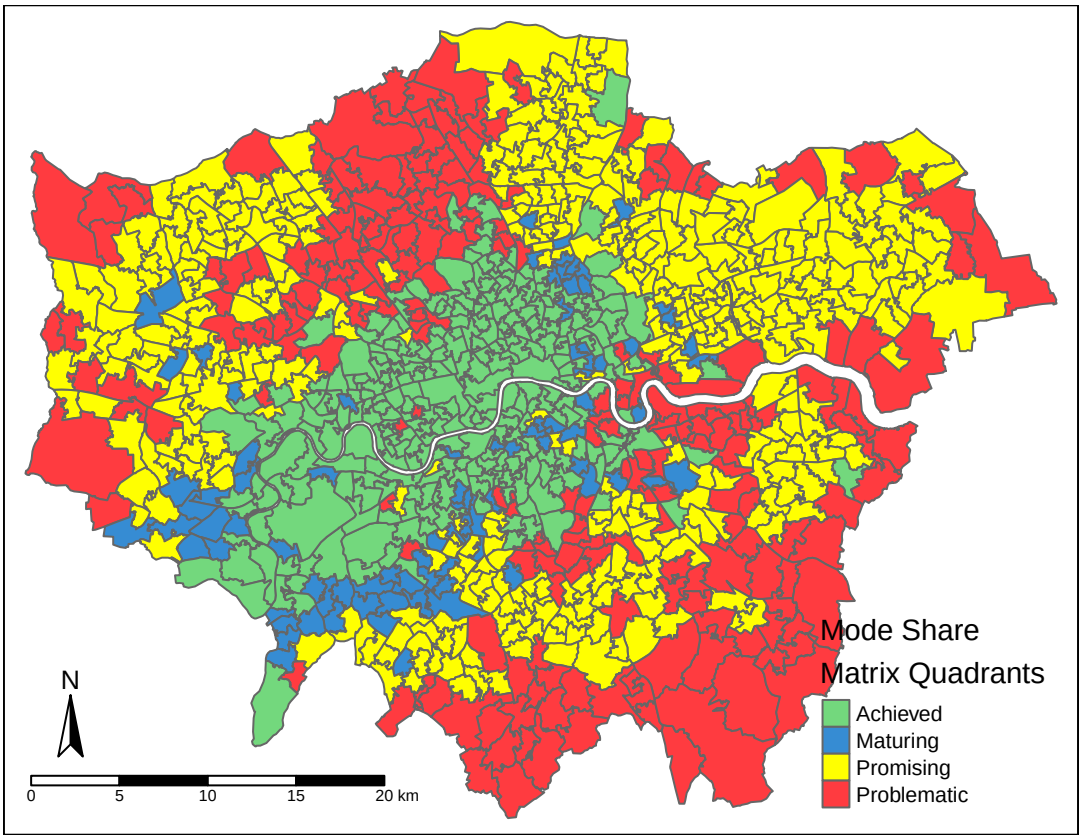


Figure 18-1 Spatial Distribution of BCG Categories

```
tm_shape(msoa) +
  tm_polygons(col = "BCGNos", palette = BCGcol,
              title = "Mode Share\nMatrix Quadrants")+
  tm_compass(position = c("left", "bottom"))+
  tm_scale_bar(position = c("left", "bottom"))
```

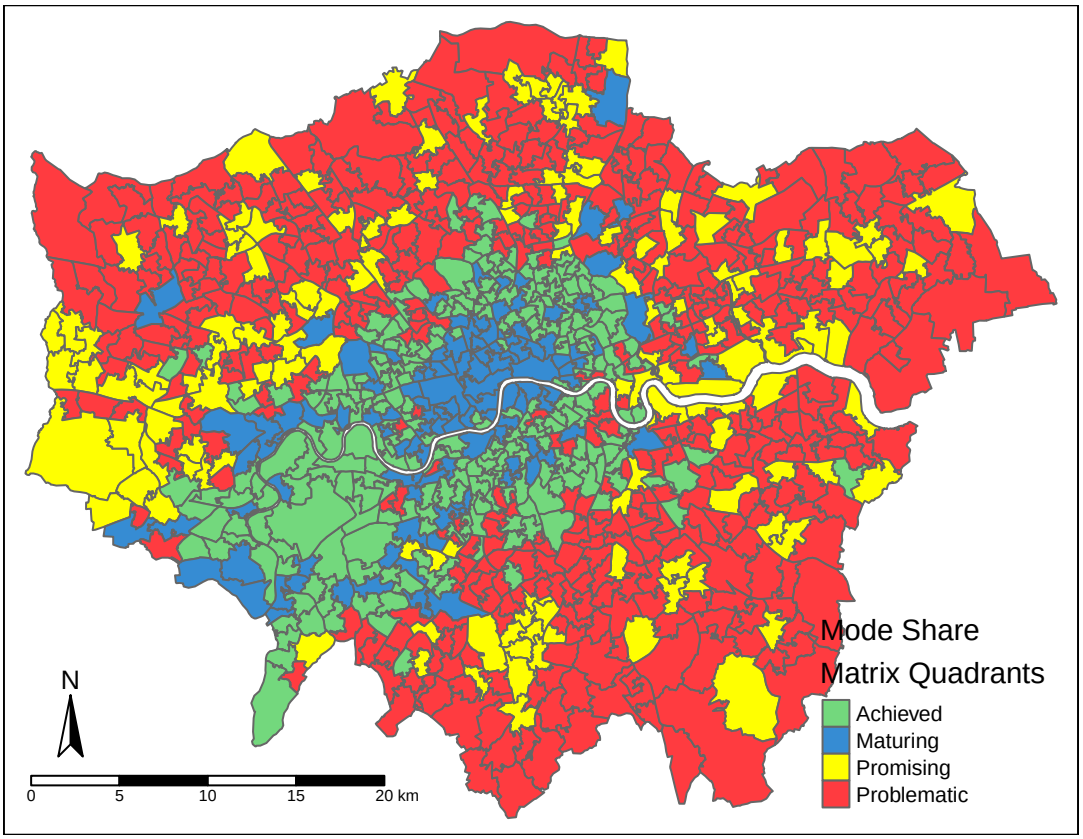


Figure 21 Breakdown of Parking Space by No. of Cycle Commuter (left) and Area (right)

```
# Join area to census 3.
census3 = left_join(census3, area, by = c("dmsoa" = "msoa"))
```

```
# Categorise into 6 ranges.
tbd=
  census3 %>%
  mutate(space2 = ifelse(space == 0, "None",
                         ifelse(1 <= space & space <= 100, "1-100",
                                ifelse(100 < space & space <= 300, "100-300",
                                       ifelse(300 < space & space <= 500, "300-500",
                                              ifelse(500 < space & space <= 1000, "500-1000",
                                                     ifelse(1000 < space,"1000-",NA))))))
           )

tbd2= tbd %>%
  group_by(space2)%>%
  summarise(all = sum(all), bicycle = sum(bicycle),area = sum(area_sqm))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```r
tbd2$space2 = as.factor(tbd2$space2)
levels(tbd2$space2)
```

```
## [1] "1-100"    "100-300"  "1000-"    "300-500"  "500-1000" "None"
```

```r
tbd2$space2 = factor(tbd2$space2,
                levels = c("None","1-100","100-300","300-500","500-1000","1000-"))

tbd2$area_kmsq = tbd2$area/1000^2
tbd2$area_kmsq = as.numeric(tbd2$area_kmsq)

# Size in terms of area
(1002575971+ 65723076 )/ 1573508480  #0.6789281 by small numbers
```

```
## [1] 0.6789281
```

```r
31713098 / 1573508480 #0.02015439
```

```
## [1] 0.02015439
```

```r
# Size in terms of cyclist
(511+ 21666 )/ 143182  #0.1548868 by small numbers
```

```
## [1] 0.1548868
```

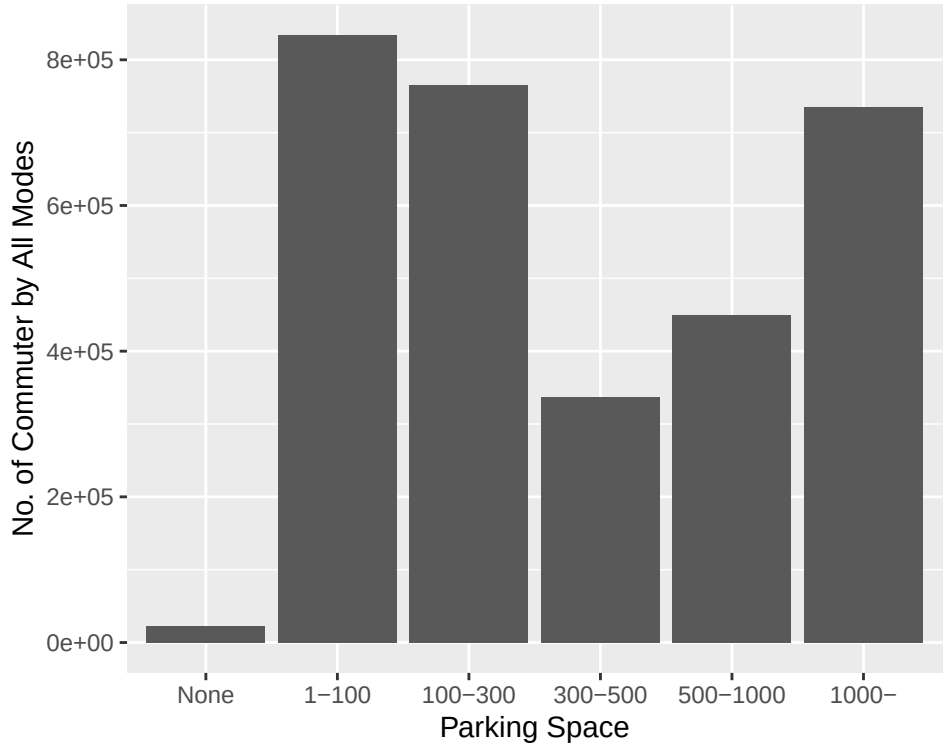
```r
sum(tbd2$bicycle)
```

```
## [1] 143182
```

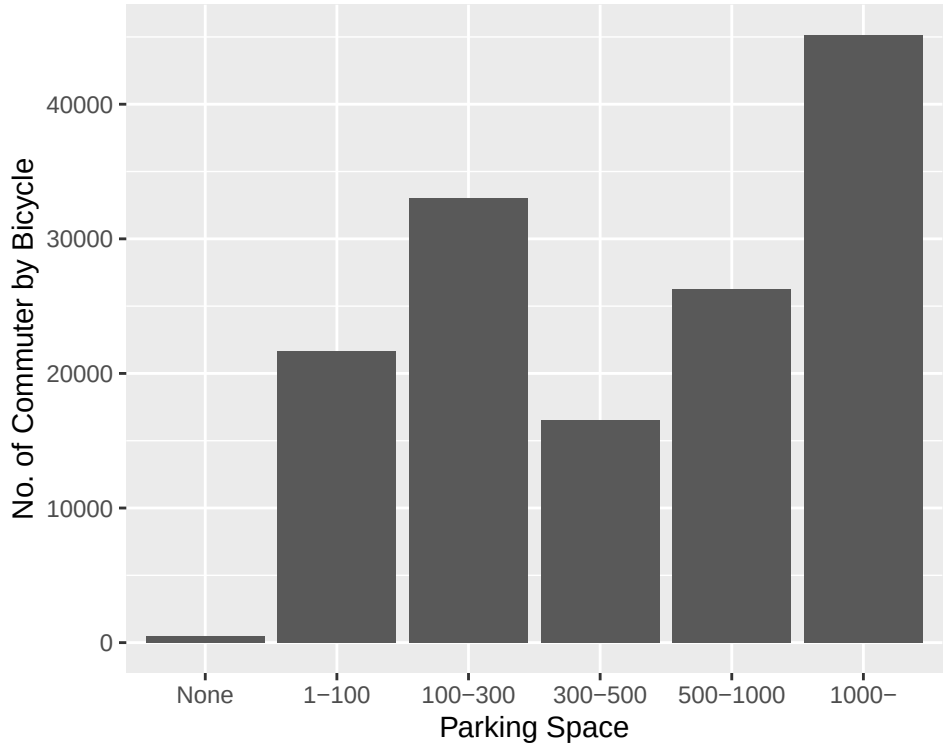
```r
45138 / 143182 #0.3152491
```

```
## [1] 0.3152491
```

```r
ggplot(tbd2,aes(x=space2, y = all)) +
  geom_bar( stat="identity")+
  labs(y="No. of Commuter by All Modes", x = "Parking Space")
```




```
ggplot(tbd2,aes(x=space2, y = bicycle)) +
  geom_bar( stat="identity")+
  labs(y="No. of Commuter by Bicycle", x = "Parking Space")
```



```r
ggplot(tbd2,aes(x=space2, y = area_kmsq)) +
  geom_bar( stat="identity")+
  labs(y="Area (sq.km)", x = "Parking Space")
```

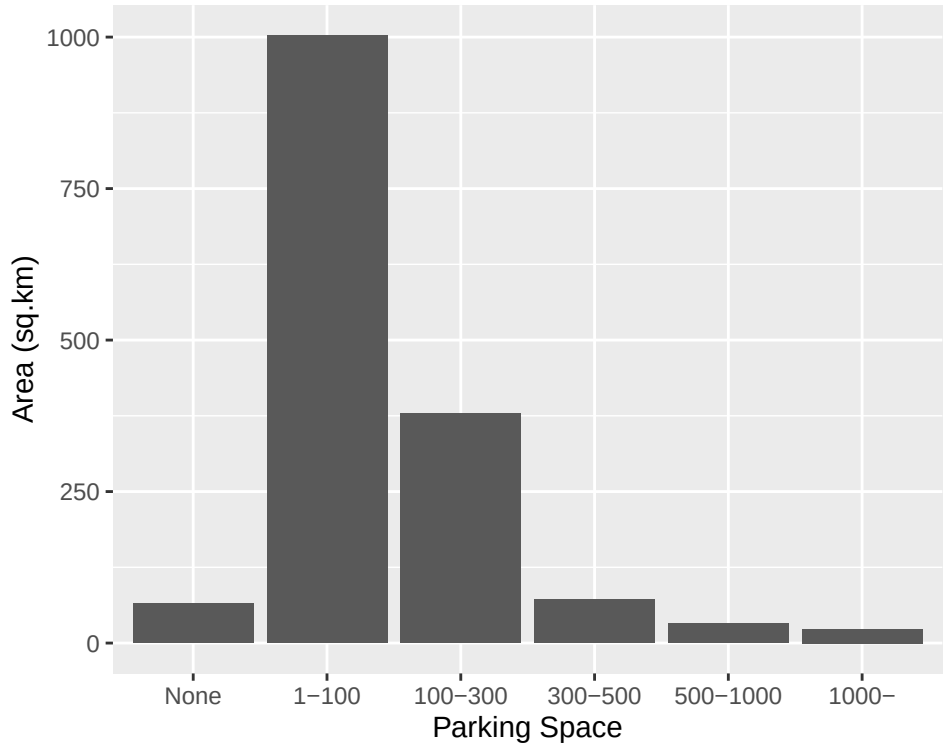
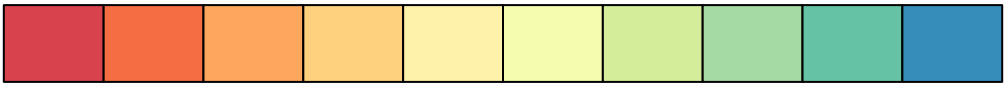


Figure 22 Spatial Distribution of Cycle Parking Supply

```r
tm_shape(msoa) +
  tm_polygons(col = "space",
              style = "fixed",
              breaks = c(0, 134, 176, 217, 270, 336, 420, 508, 718, 1074, 3010),
              palette = get_brewer_pal("Spectral", n = 10, contrast = c(0, 0.75)),
              legend.hist = TRUE,
              title = "Parking Supply")+
  tm_layout(legend.outside = TRUE) +
  tm_compass(position = c("left", "bottom"))+
  tm_scale_bar(position = c("left", "bottom"))
```



#D8424D #F46D43 #FCA75D #FDD17E #FEF2AA #F5FBAF #D4ED9B #A4DAA4 #66C2A5 #378DBA

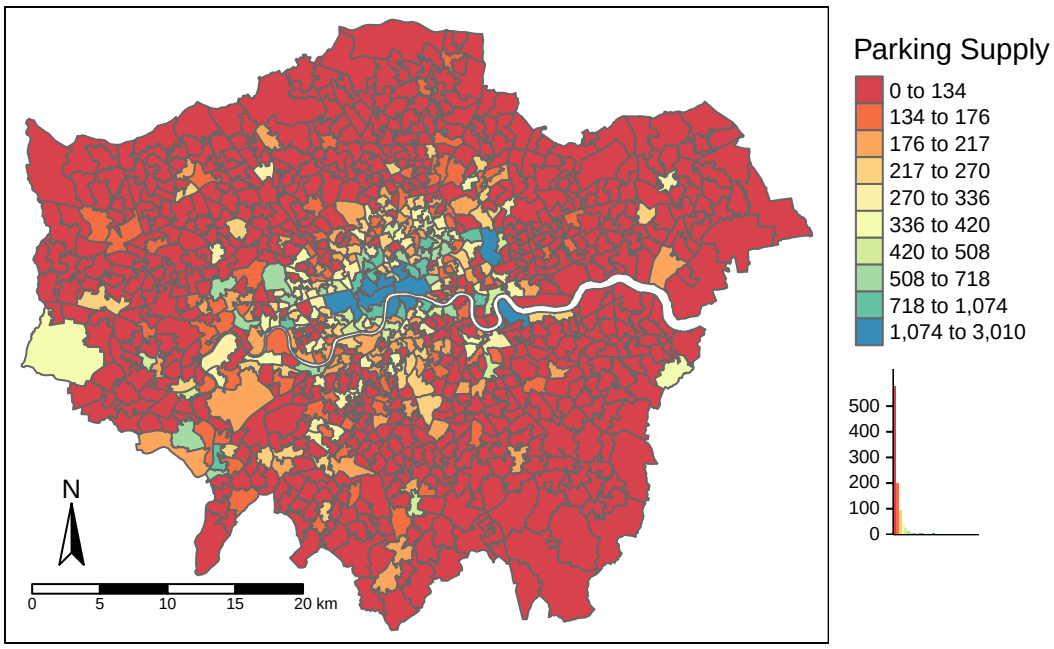
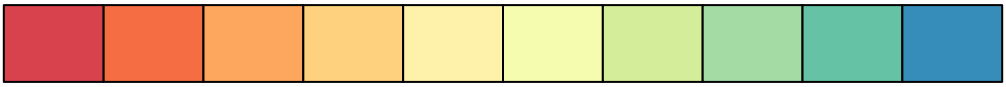


Figure 25 Spatial Distribution of Current Parking Space Demand

```
tm_shape(msoa) +
  tm_polygons(col = "bicycle",
              style = "fixed",
              breaks = c(0, 134, 176, 217, 270, 336, 420, 508, 718, 1074, 13053),
              legend.hist = TRUE,
              palette = get_brewer_pal("Spectral", n = 10, contrast = c(0, 0.75)),
              title = "Current Parking \nDemand")+
  tm_layout(legend.outside = TRUE) +
  tm_compass(position = c("left", "bottom"))+
  tm_scale_bar(position = c("left", "bottom"))
```



#D8424D #F46D43 #FCA75D #FDD17E #FEF2AA #F5FBAF #D4ED9B #A4DAA4 #66C2A5 #378DBA

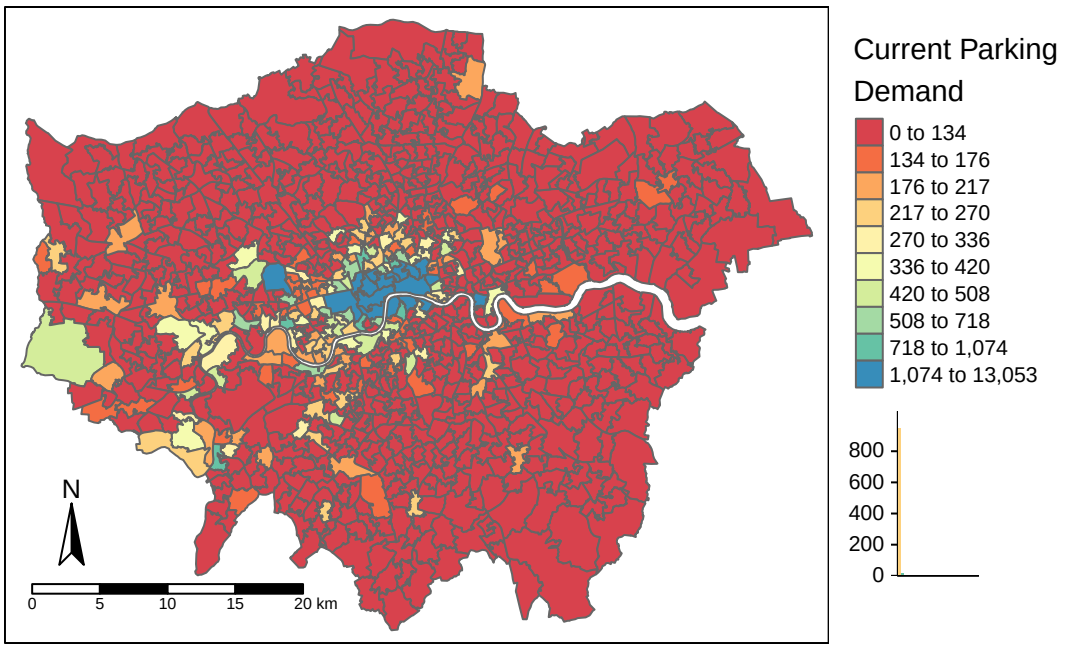
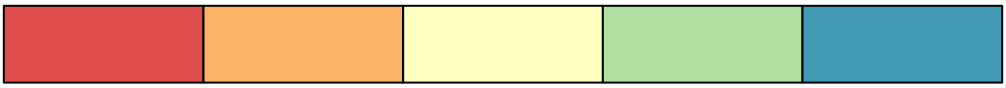
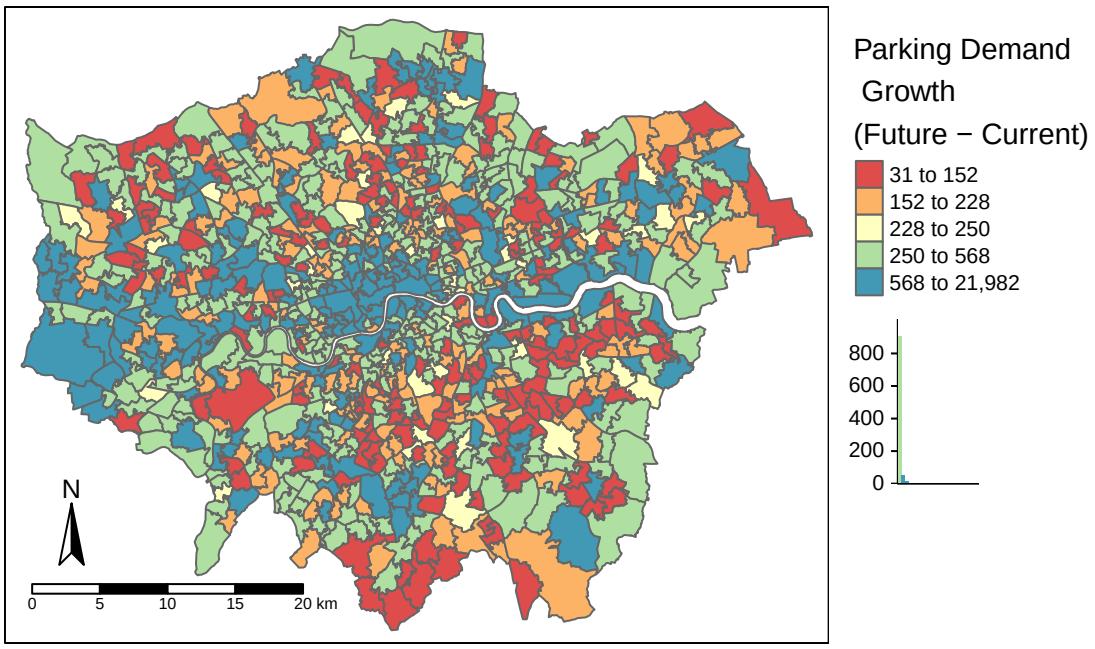


Figure 26 Increased Amount (left) and Increased Rate (right) of Parking Demand

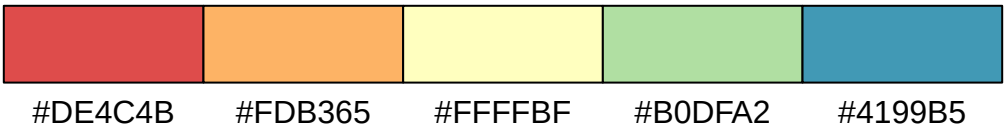
```
tbd = msoa[,c("bicycle", "bicycle_future")]
tbd$increasedAmount = tbd$bicycle_future - tbd$bicycle
tbd$increasedRate = tbd$bicycle_future/tbd$bicycle

tm_shape(tbd) +
  tm_polygons(col = "increasedAmount",
              style = "fixed",
              breaks = c(31, 152, 228, 250, 568, 21982),
              legend.hist = TRUE,
              palette = get_brewer_pal("Spectral", n = 5, contrast = c(0, 0.75)),
              title = "Parking Demand\n Growth\n(Future - Current)")+
  tm_layout(legend.outside = TRUE) +
  tm_compass(position = c("left", "bottom"))+
  tm_scale_bar(position = c("left", "bottom"))
```


#DE4C4B
#FDB365
#FFFFBF
#B0DFA2
#4199B5




```
tm_shape(tbd) +
  tm_polygons(col = "increasedRate",
              style = "fixed",
              breaks = c(1.6, 4, 7, 10, 15,109),
              legend.hist = TRUE,
              palette = get_brewer_pal("Spectral", n = 5, contrast = c(0, 0.75)),
              title = "Parking Demand\n Growth\n(Future/Current)")+
  tm_layout(legend.outside = TRUE) +
  tm_compass(position = c("left", "bottom"))+
  tm_scale_bar(position = c("left", "bottom"))
```




```
## Warning: Values have found that are higher than the highest break
```

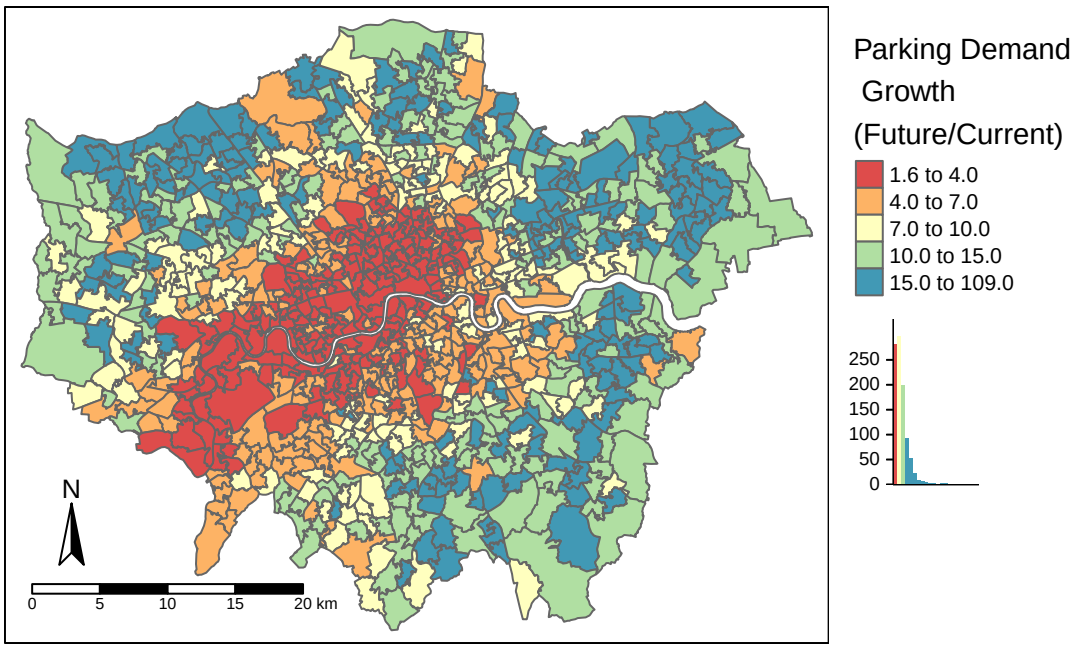
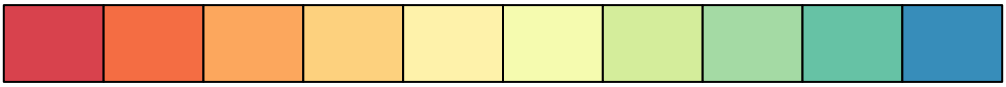


Figure 28 Spatial Distribution of Future Parking Capacity Demand

```
tm_shape(msoa) +
  tm_polygons(col = "bicycle_future",
              style = "fixed",
              breaks = c(34, 134, 176, 217, 270, 336, 420, 508, 718, 1074, 35035),
              legend.hist = TRUE,
              palette = get_brewer_pal("Spectral", n = 10, contrast = c(0, 0.75)),
              title = "Future Parking\n Demand")+
  tm_layout(legend.outside = TRUE)
```



#D8424D #F46D43 #FCA75D #FDD17E #FEF2AA #F5FBAF #D4ED9B #A4DAA4 #66C2A5 #378DBA

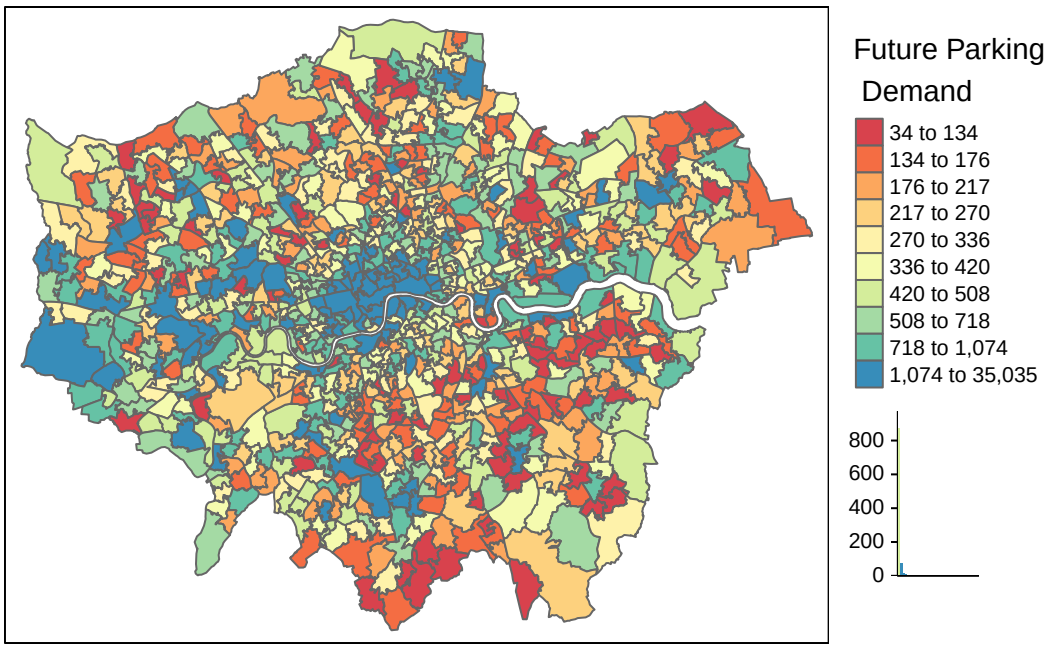


Figure 29 Spatial Distribution of Parking Occupancy Category for Current Scenario

```
tm_shape(msoa) +
  tm_polygons(col = "porCurrentCat",
              palette = col_pocu,
              title = "Current \n Parking Occupancy")+
  tm_compass(position = c("left", "bottom"))+
  tm_scale_bar(position = c("left", "bottom"))
```

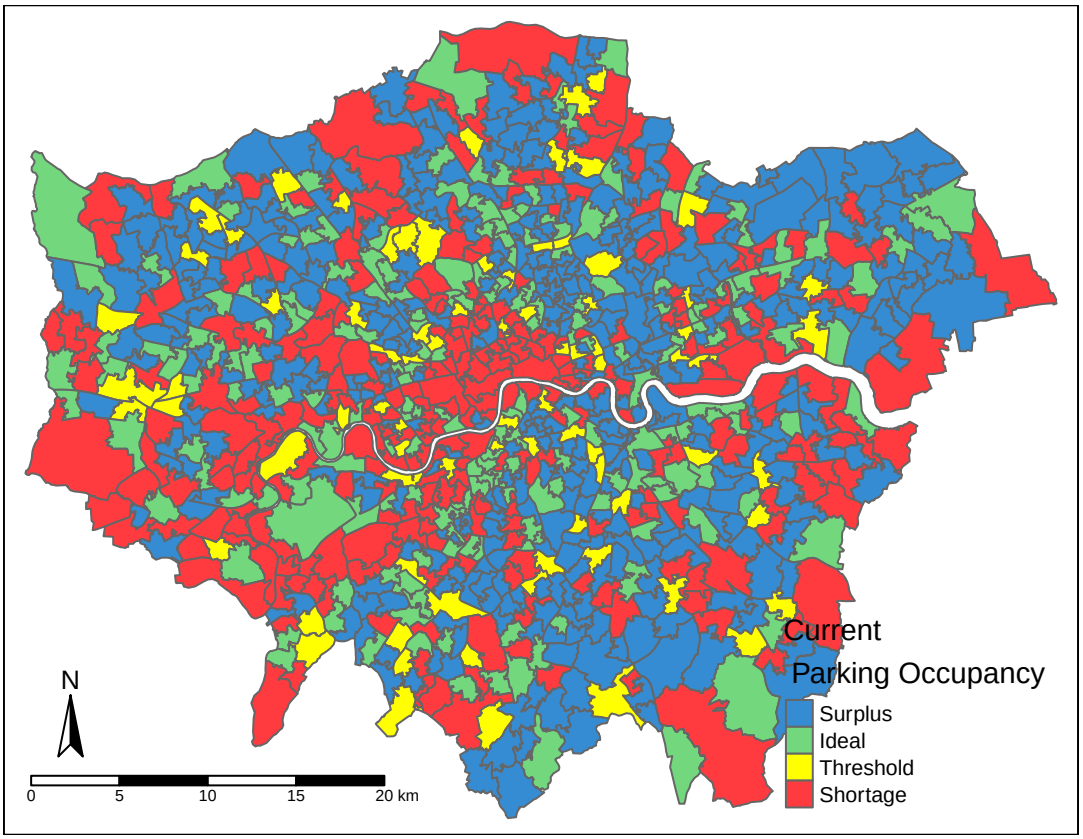
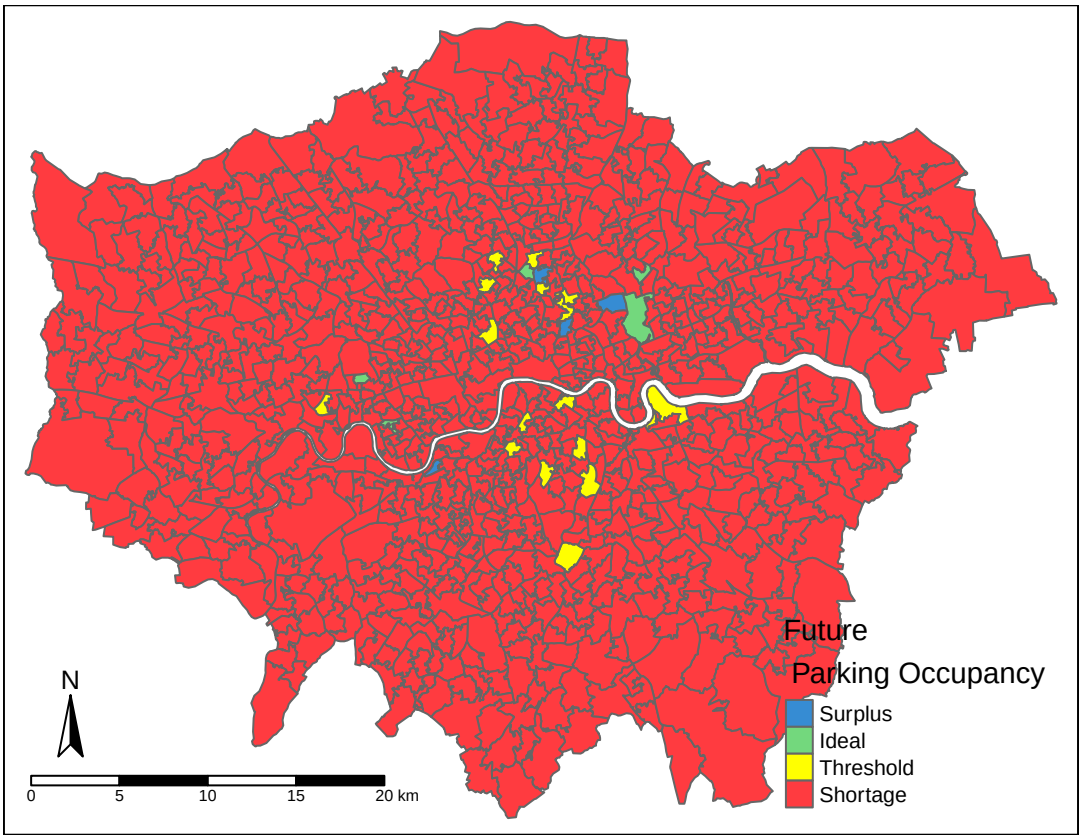


Figure 30 Spatial Distribution of Parking Occupancy Category for Future Scenario

```
tm_shape(msoa) +
  tm_polygons(col = "porFutureCat",
              palette = col_pocu,
              title = "Future \n Parking Occupancy")+
  tm_compass(position = c("left", "bottom"))+
  tm_scale_bar(position = c("left", "bottom"))
```



BCG Growth share version

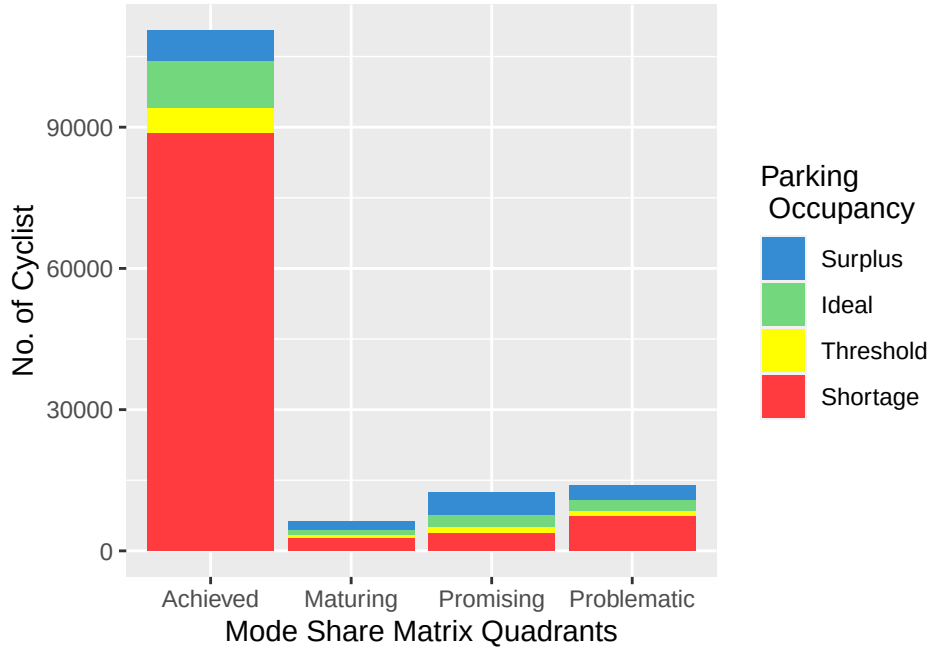
Figure 31 Breakdown of Parking Occupancy Category based on No. of Cyclist for Current Scenario (left) and Future Scenario (right)

Figure 32 Percentage Breakdown of Parking Occupancy Category based on No. of Cyclist for Current Scenario (left) and Future Scenario (right)

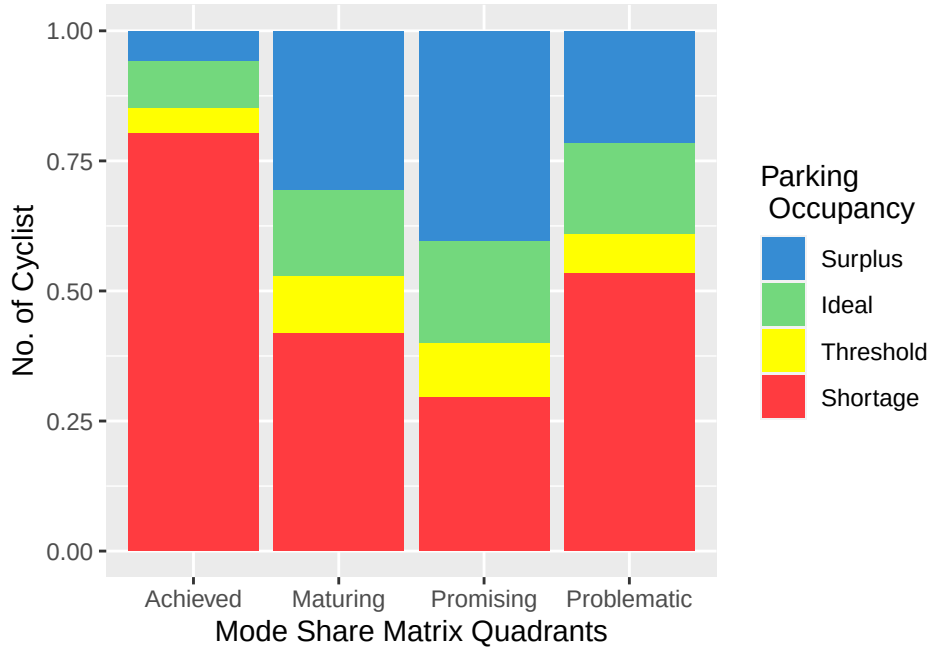
```
# Current scenario
tbd =
  census3 %>%
  group_by(BCG, porCurrentCat) %>%
    summarise(count = n(), cyclists = sum(bicycle))
```

```
## `summarise()` regrouping output by 'BCG' (override with `.groups` argument)
```

```
# Count Stacked barchart
ggplot(tbd, aes(fill=porCurrentCat, y=cyclists, x=BCG))  +
    geom_bar(position="stack", stat="identity")+
  scale_fill_manual(values=col_pocu)+
  labs(fill = "Parking\n Occupancy")+
  labs(y="No. of Cyclist", x = "Mode Share Matrix Quadrants")
```




```
# Percent stacked barchart
ggplot(tbd, aes(fill=porCurrentCat, y=cyclists, x=BCG))  +
    geom_bar(position="fill",  stat="identity", ) +
  scale_fill_manual(values=col_pocu)+
  labs(fill = "Parking\n Occupancy")+
  labs(y="No. of Cyclist", x = "Mode Share Matrix Quadrants")
```

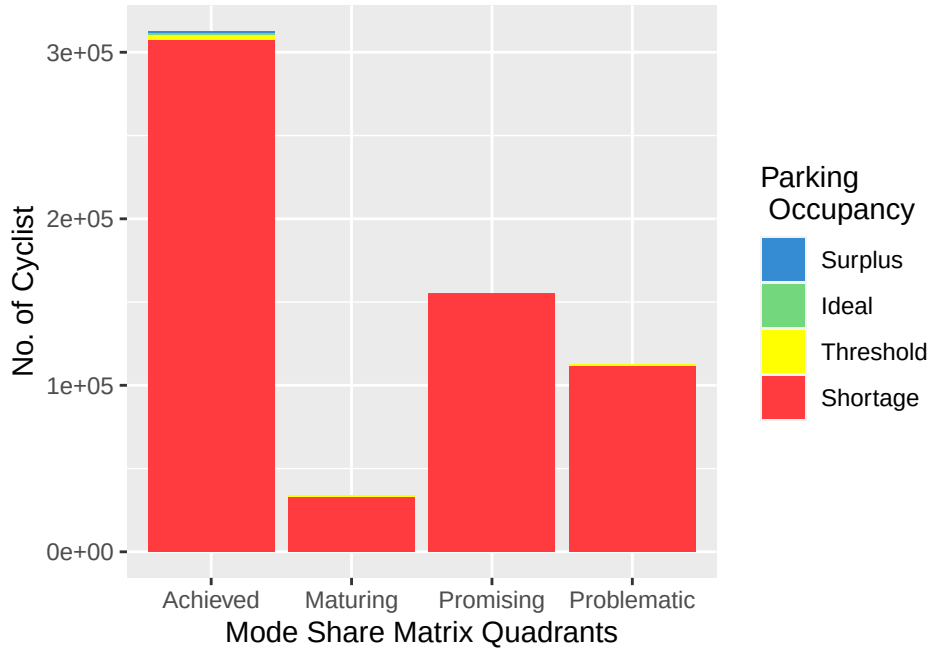



```
# Future scenario
# Group by BCG and parking availability category based on GoDutch
```

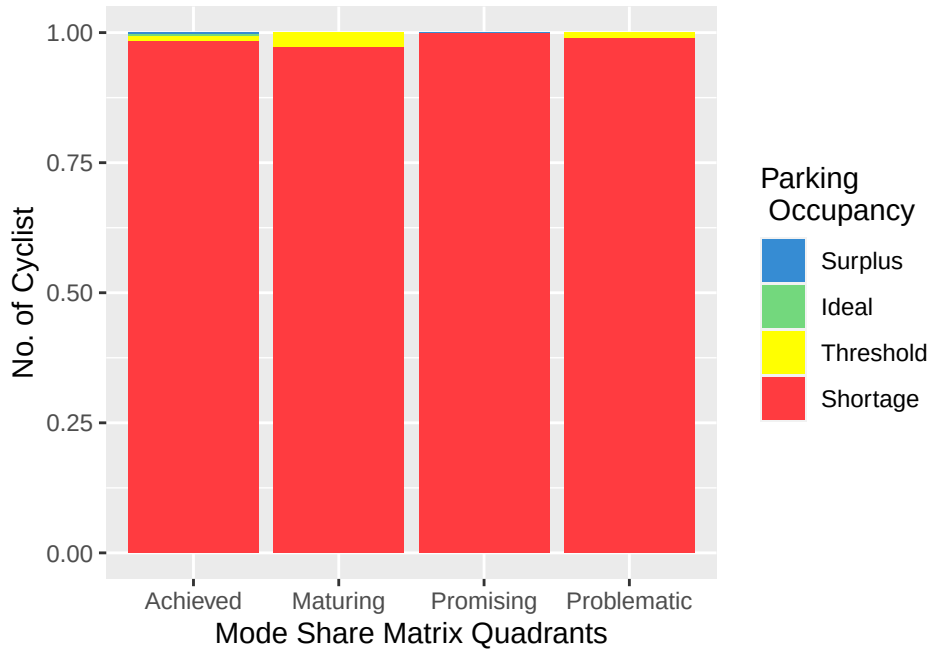
```
tbd =
  census3 %>%
  group_by(BCG, porFutureCat) %>%
    summarise(count = n(), cyclists = sum(bicycle_future))
```

```
## `summarise()` regrouping output by 'BCG' (override with `.groups` argument)
```

```
# Count Stacked barchart
ggplot(tbd, aes(fill=porFutureCat, y=cyclists, x=BCG))  +
    geom_bar(position="stack", stat="identity")+
  scale_fill_manual(values=col_pocu)+
  labs(fill = "Parking\n Occupancy")+
  labs(y="No. of Cyclist", x = "Mode Share Matrix Quadrants")
```




```
# Percent stacked barchart
ggplot(tbd, aes(fill=porFutureCat, y=cyclists, x=BCG))  +
    geom_bar(position="fill",  stat="identity", ) +
  scale_fill_manual(values=col_pocu)+
  labs(fill = "Parking\n Occupancy")+
  labs(y="No. of Cyclist", x = "Mode Share Matrix Quadrants")
```

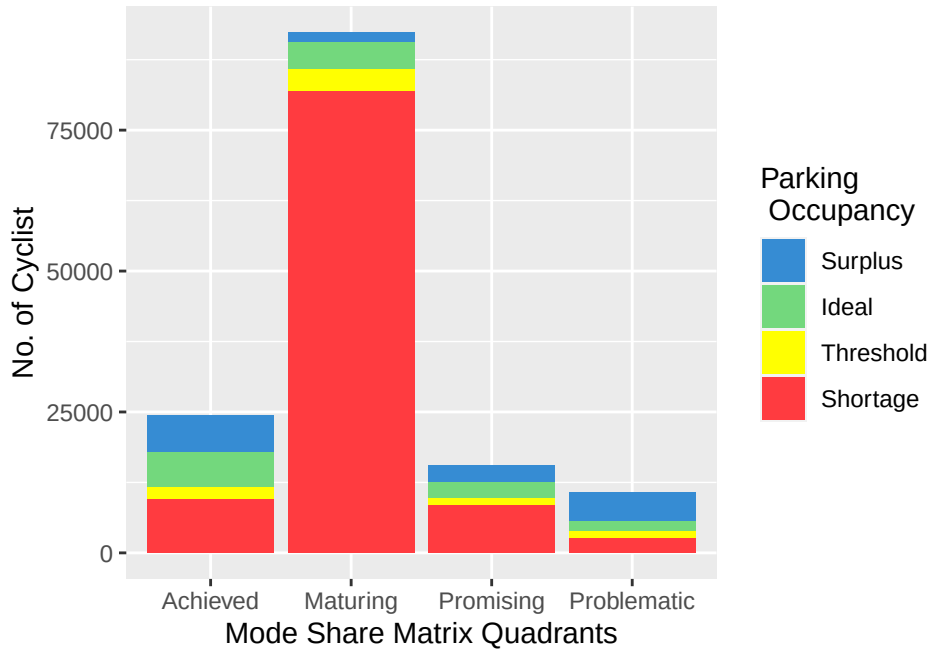


BCG Growth Amount version ### Figure 31-1 Breakdown of Parking Occupancy Category based on No. of Cyclist for Current Scenario (left) and Future Scenario (right) ### Figure 32-1 Percentage Breakdown of Parking Occupancy Category based on No. of Cyclist for Current Scenario (left) and Future Scenario (right)

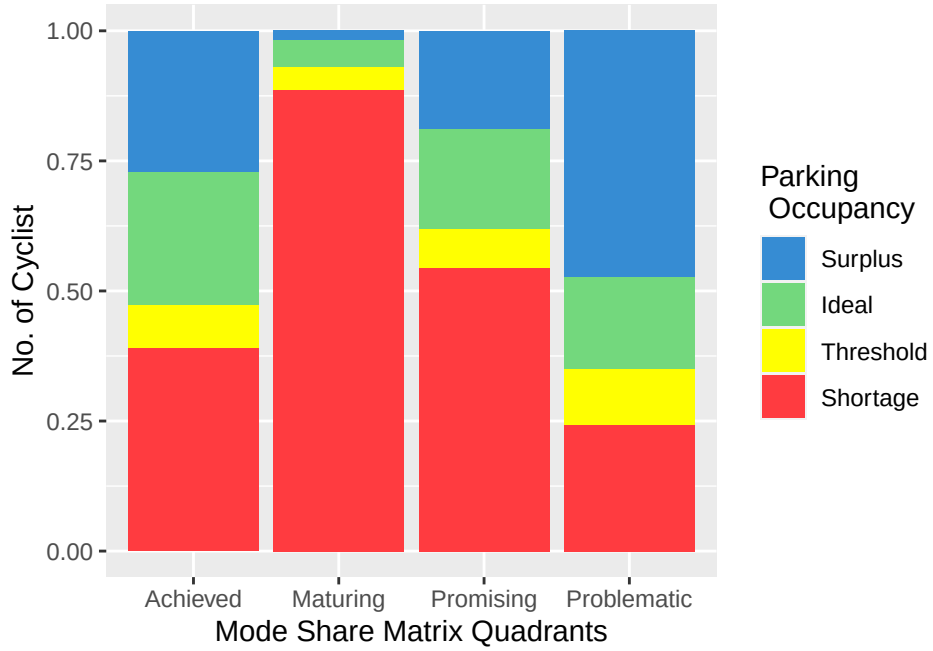
```
# Current scenario
tbd =
  census3 %>%
  group_by(BCGNos, porCurrentCat) %>%
    summarise(count = n(), cyclists = sum(bicycle))
```

```
## `summarise()` regrouping output by 'BCGNos' (override with `.groups` argument)
```

```
# Count Stacked barchart
ggplot(tbd, aes(fill=porCurrentCat, y=cyclists, x=BCGNos))  +
    geom_bar(position="stack", stat="identity")+
  scale_fill_manual(values=col_pocu)+
  labs(fill = "Parking\n Occupancy")+
  labs(y="No. of Cyclist", x = "Mode Share Matrix Quadrants")
```




```
# Percent stacked barchart
ggplot(tbd, aes(fill=porCurrentCat, y=cyclists, x=BCGNos))  +
    geom_bar(position="fill",  stat="identity", ) +
  scale_fill_manual(values=col_pocu)+
  labs(fill = "Parking\n Occupancy")+
  labs(y="No. of Cyclist", x = "Mode Share Matrix Quadrants")
```

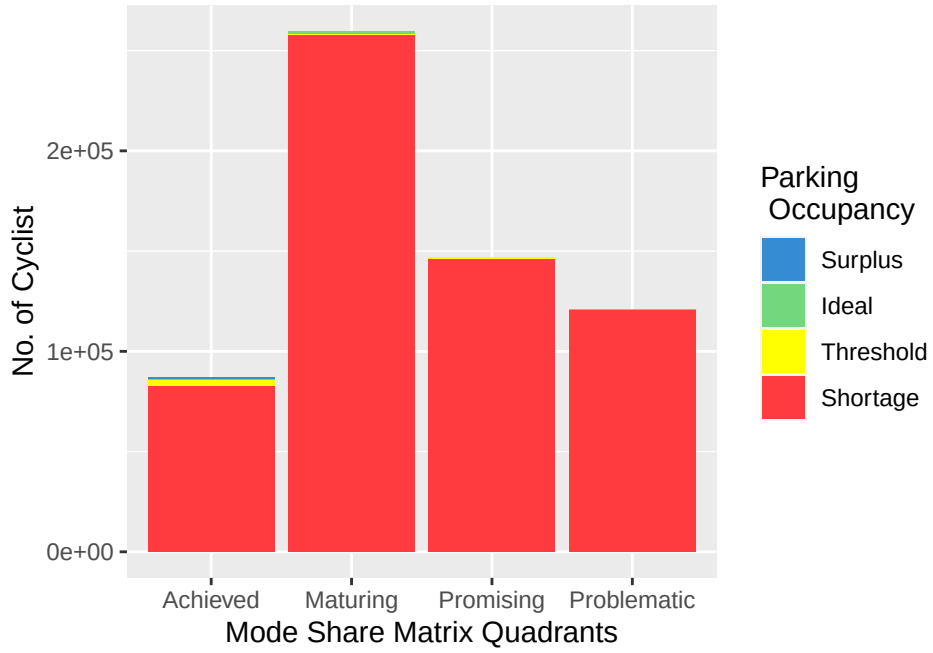



```
# Future scenario
# Group by BCG and parking availability category based on GoDutch
```

```r
tbd =
  census3 %>%
  group_by(BCGNos, porFutureCat) %>%
    summarise(count = n(), cyclists = sum(bicycle_future))
```

```
## `summarise()` regrouping output by 'BCGNos' (override with `.groups` argument)
```

```r
# Count Stacked barchart
ggplot(tbd, aes(fill=porFutureCat, y=cyclists, x=BCGNos))  +
    geom_bar(position="stack", stat="identity")+
  scale_fill_manual(values=col_pocu)+
  labs(fill = "Parking\n Occupancy")+
  labs(y="No. of Cyclist", x = "Mode Share Matrix Quadrants")
```




```r
# Percent stacked barchart
ggplot(tbd, aes(fill=porFutureCat, y=cyclists, x=BCGNos))  +
    geom_bar(position="fill",  stat="identity", ) +
  scale_fill_manual(values=col_pocu)+
  labs(fill = "Parking\n Occupancy")+
  labs(y="No. of Cyclist", x = "Mode Share Matrix Quadrants")
```

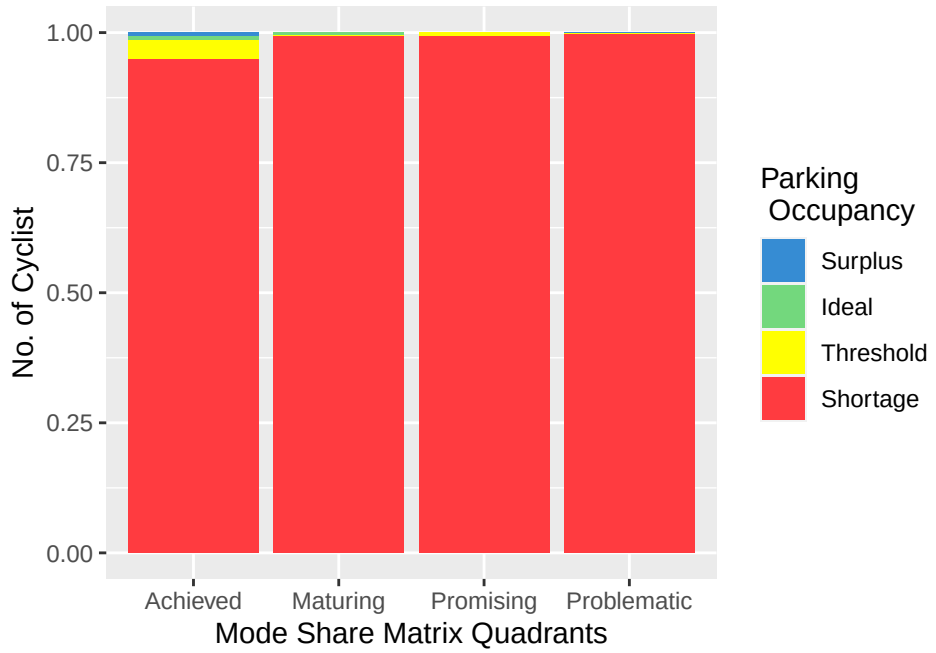
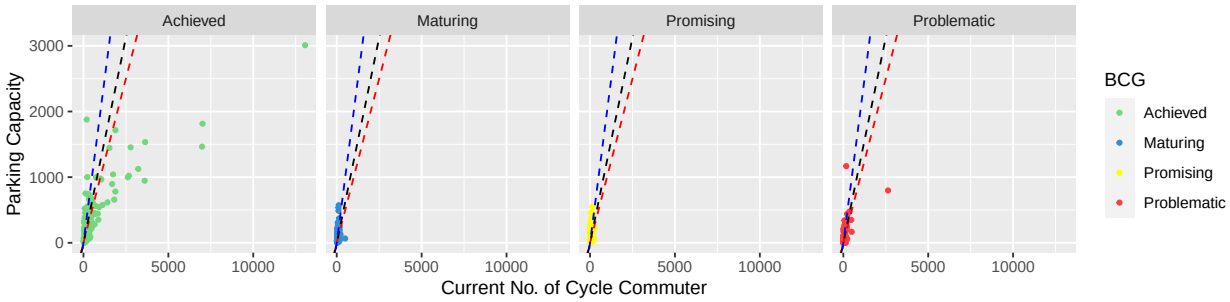


Figure 33 Scatter Plot of Parking Space and No. of Cyclist by MSOA for Current Scenario

```
ggplot(census3,
       aes(x = bicycle,
           y = space,
           color = BCG)
       )+
  geom_point(size = 1) +
  scale_colour_manual(values=BCGcol) +
  geom_abline(intercept = 0,
              slope = 1,
              color="red",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:1
  geom_abline(intercept = 0,
              slope = 1.25,
              color="black",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:1.2
  geom_abline(intercept = 0,
              slope = 2,
              color="blue",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:2
  xlim(0,NA)+
  ylim(0,NA)+
  facet_wrap( ~ BCG, ncol=4)+
  labs(y="Parking Capacity", x = "Current No. of Cycle Commuter")
```




```
ggplot(census3,
       aes(x = bicycle,
           y = space,
           color = BCG)
       )+
  geom_point(size = 1) +
  scale_colour_manual(values=BCGcol) +
  geom_abline(intercept = 0,
              slope = 1,
              color="red",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:1
  geom_abline(intercept = 0,
              slope = 1.25,
              color="black",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:1.2
  geom_abline(intercept = 0,
              slope = 2,
              color="blue",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:2
  xlim(0,1000)+
  ylim(0,500)+
  facet_wrap( ~ BCG, ncol=4)+
  labs(y="Parking Capacity", x = "Current No. of Cycle Commuter")
```

```
## Warning: Removed 42 rows containing missing values (geom_point).
```

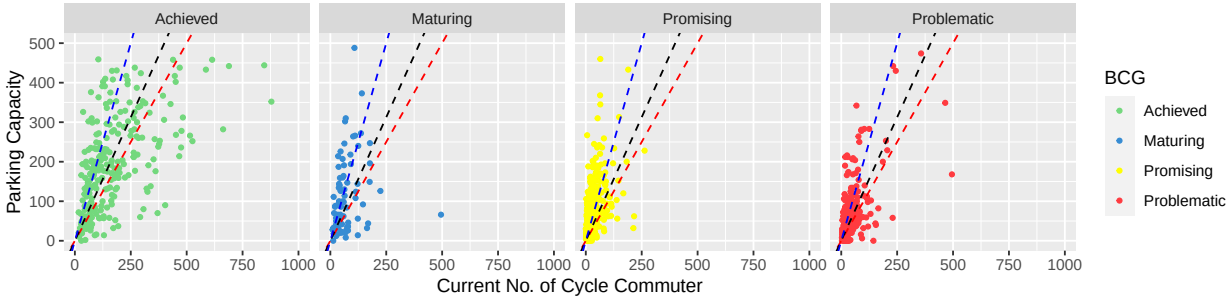


Figure 33-1 Scatter Plot of Parking Space and No. of Cyclist by MSOA for Current Scenario LOG SCALE! - Growth Share Version

```
test = ggplot(census3,
       aes(x = bicycle,
           y = space,
           color = BCG)
       )+
  geom_point(size = 1) +
  scale_colour_manual(values=BCGcol) +
  geom_abline(intercept = 0,
              slope = 1,
              color="red",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:1
  geom_abline(intercept = 0,
              slope = 1.25,
              color="black",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:1.2
  geom_abline(intercept = 0,
              slope = 2,
              color="blue",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:2
  xlim(0,NA)+
  ylim(0,NA)+
  facet_wrap( ~ BCG, ncol=4)+
  labs(y="Parking Capacity", x = "Current No. of Cycle Commuter")

test + scale_x_continuous(trans = 'log10') +  scale_y_continuous(trans = 'log10')
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which
## will replace the existing scale.

## Scale for 'y' is already present. Adding another scale for 'y', which
## will replace the existing scale.

## Warning: Transformation introduced infinite values in continuous x-axis

## Warning: Transformation introduced infinite values in continuous y-axis
```

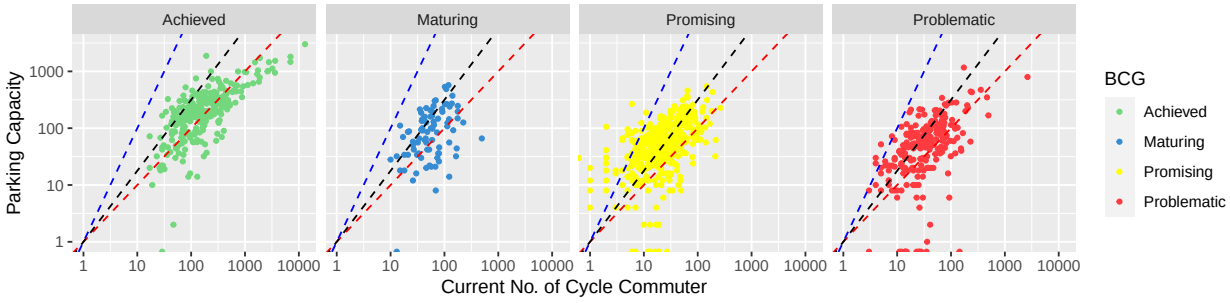


Figure 33-1-1 Scatter Plot of Parking Space and No. of Cyclist by MSOA for Current Scenario LOG SCALE! - Growth Number Version

```
test = ggplot(census3,
       aes(x = bicycle,
           y = space,
           color = BCGNos)
       )+
  geom_point(size = 1) +
  scale_colour_manual(values=BCGcol) +
  geom_abline(intercept = 0,
              slope = 1,
              color="red",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:1
  geom_abline(intercept = 0,
              slope = 1.25,
              color="black",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:1.2
  geom_abline(intercept = 0,
              slope = 2,
              color="blue",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:2
  xlim(0,1000)+
  ylim(0,500)+
  facet_wrap( ~ BCGNos, ncol=4)+
  labs(y="Parking Capacity", x = "Current No. of Cycle Commuter")

test + scale_x_continuous(trans = 'log10') +  scale_y_continuous(trans = 'log10')
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which
## will replace the existing scale.
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which
## will replace the existing scale.
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

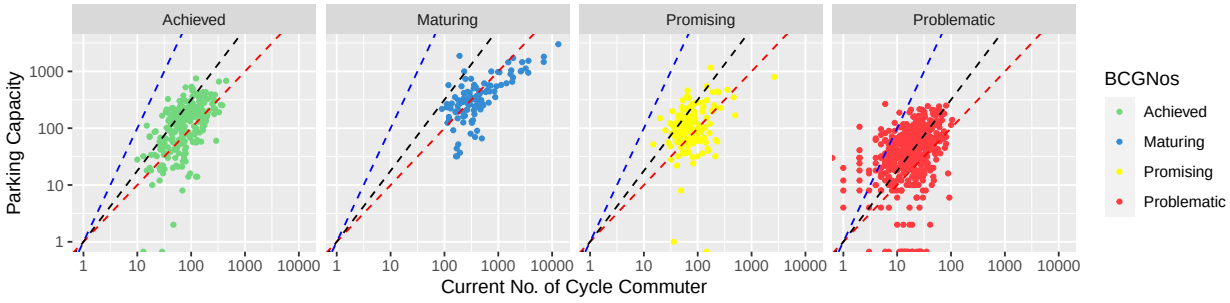
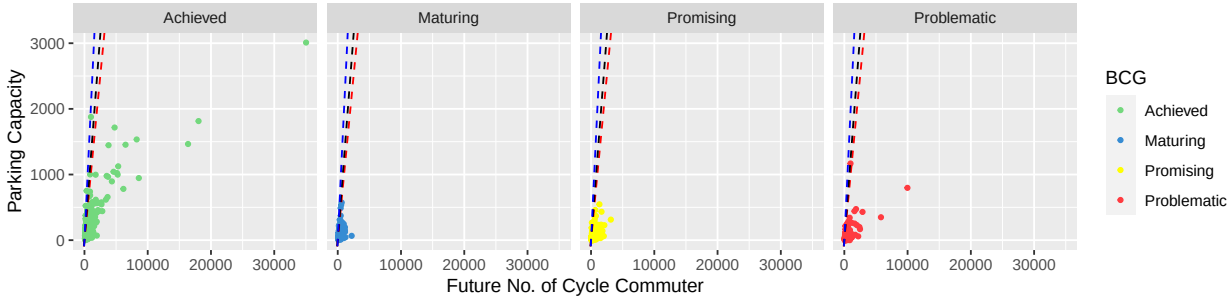


Figure 34 Scatter Plot of Parking Space and No. of Cyclist by MSOA for Future Scenario

```
# GoDutch - Plot provision of parking per BCG.
ggplot(census3,
       aes(x = bicycle_future,
           y = space,
           color = BCG)
       )+
  geom_point(size = 1) +
  scale_colour_manual(values=BCGcol) +
  geom_abline(intercept = 0,
              slope = 1,
              color="red",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:1
  geom_abline(intercept = 0,
              slope = 1.25,
              color="black",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:1.2
  geom_abline(intercept = 0,
              slope = 2,
              color="blue",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:2
  xlim(0,NA)+
  ylim(0,NA)+
  facet_wrap( ~ BCG, ncol=4)+
  labs(y="Parking Capacity", x = "Future No. of Cycle Commuter")
```




```
# GoDutch - Plot provision of parking per BCG.
ggplot(census3,
       aes(x = bicycle_future,
           y = space,
           color = BCG)
       )+
  geom_point(size = 1) +
  scale_colour_manual(values=BCGcol) +
  geom_abline(intercept = 0,
              slope = 1,
              color="red",
              linetype="dashed",
```

```
              size=0.5) + # Cyclist:Parking=1:1
  geom_abline(intercept = 0,
              slope = 1.25,
              color="black",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:1.2
  geom_abline(intercept = 0,
              slope = 2,
              color="blue",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:2
  xlim(0,1000)+
  ylim(0,500)+
  facet_wrap( ~ BCG, ncol=4)+
  labs(y="Parking Capacity", x = "Future No. of Cycle Commuter")
```

```
## Warning: Removed 120 rows containing missing values (geom_point).
```

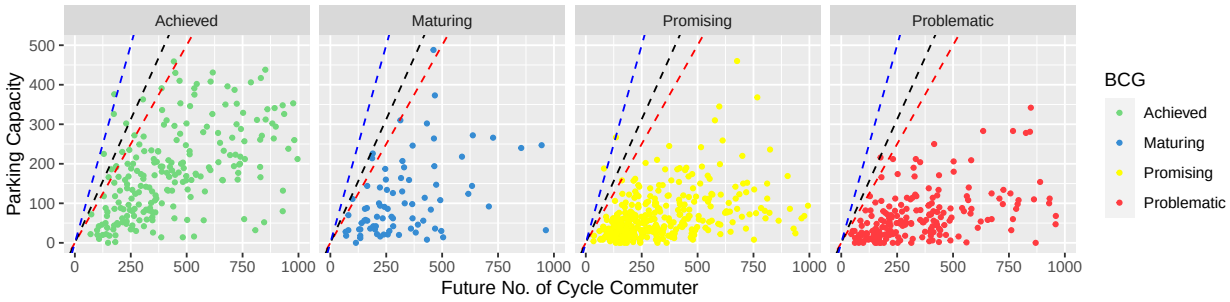


Figure 34-1 Scatter Plot of Parking Space and No. of Cyclist by MSOA for Future Scenario - Growth Share Version

```
# GoDutch - Plot provision of parking per BCG.
test = ggplot(census3,
       aes(x = bicycle_future,
           y = space,
           color = BCG)
       )+
  geom_point(size = 1) +
  scale_colour_manual(values=BCGcol) +
  geom_abline(intercept = 0,
              slope = 1,
              color="red",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:1
  geom_abline(intercept = 0,
              slope = 1.25,
              color="black",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:1.2
  geom_abline(intercept = 0,
```

```
              slope = 2,
              color="blue",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:2
  xlim(0,NA)+
  ylim(0,NA)+
  facet_wrap( ~ BCG, ncol=4)+
  labs(y="Parking Capacity", x = "Future No. of Cycle Commuter")

test + scale_x_continuous(trans = 'log10') +  scale_y_continuous(trans = 'log10')
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which
## will replace the existing scale.
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which
## will replace the existing scale.
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

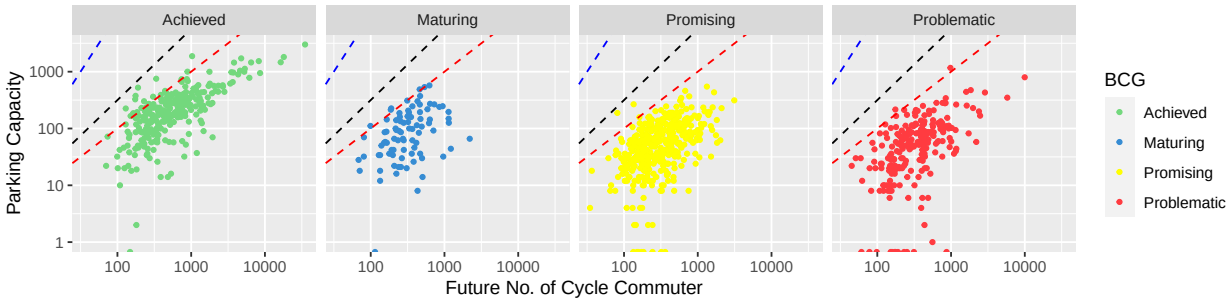


Figure 34-1-1 Scatter Plot of Parking Space and No. of Cyclist by MSOA for Future Scenario - Growth Number Version

```
# GoDutch - Plot provision of parking per BCG.
test = ggplot(census3,
       aes(x = bicycle_future,
           y = space,
           color = BCGNos)
       )+
  geom_point(size = 1) +
  scale_colour_manual(values=BCGcol) +
  geom_abline(intercept = 0,
              slope = 1,
              color="red",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:1
  geom_abline(intercept = 0,
              slope = 1.25,
              color="black",
              linetype="dashed",
```

```
              size=0.5) + # Cyclist:Parking=1:1.2
  geom_abline(intercept = 0,
              slope = 2,
              color="blue",
              linetype="dashed",
              size=0.5) + # Cyclist:Parking=1:2
  xlim(0,1000)+
  ylim(0,500)+
  facet_wrap( ~ BCGNos, ncol=4)+
  labs(y="Parking Capacity", x = "Future No. of Cycle Commuter")

test + scale_x_continuous(trans = 'log10') +  scale_y_continuous(trans = 'log10')
```

```
## Scale for 'x' is already present. Adding another scale for 'x', which
## will replace the existing scale.
```

```
## Scale for 'y' is already present. Adding another scale for 'y', which
## will replace the existing scale.
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```